LAB REPORT

Embedded Systems

Pushkar Kumar
2017UGEC032R

Akash Kumar
2017UGEC033R

Saket Tiwari
2017UGEC035R

Manu Kumari
2017UGEC034R

Vikas Kumar
2017UGEC037R

VITH SEMSESTER
INDIAN INSTITUTE OF INFORMATION TECHONOLGY
Ranchi

PREFACE

This project has been designed keeping this in view to make the harnessing of solar energy more efficient on tinkercad.

We learned and performed this experiment under the constant guidance of Shashikant Sir.

Throughout performing this experiment, we learned about Microcontrollers and Embedded System and Embedded C programming.

We would like to thank Shashikant sir and Sukanta Sir who gave us this opportunity to learn and work under his mentorship and we believe Learning never stops.

Group 5

Pushkar Kumar (2017UGEC032R)

Akash Kumar (2017UGEC033R)

Manu Kumari (2017UGEC034R)

Saket Tiwari (2017UGEC035R)

Vikas Kumar (2017UGEC037R)

INDEX

TITLE

Electricity Harvest

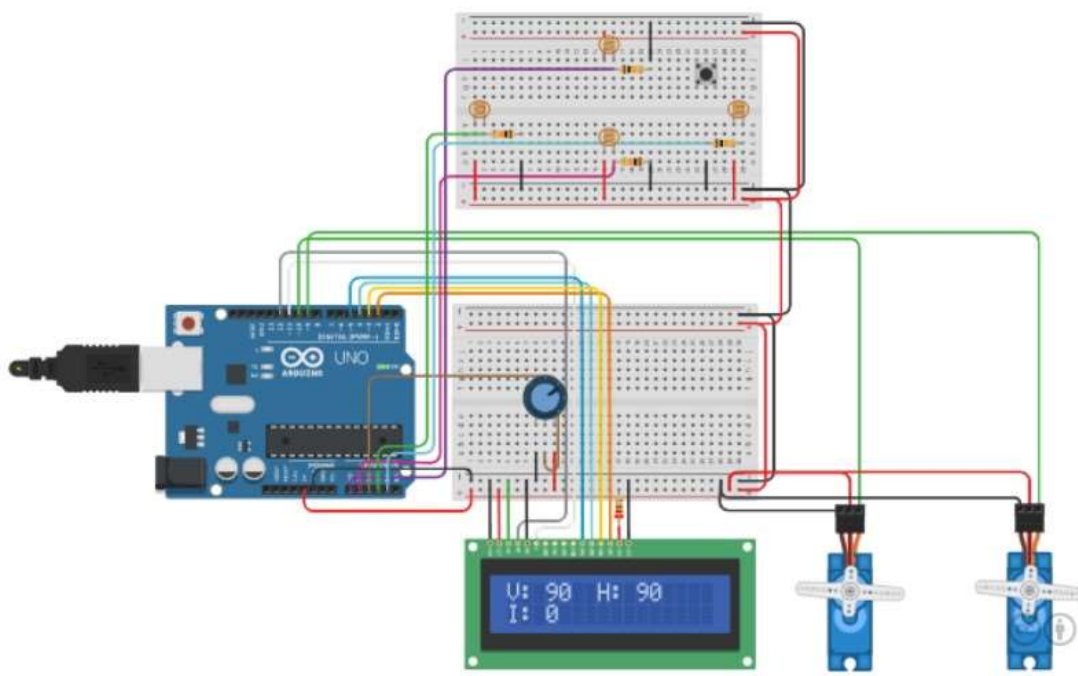
Using

Solar Plate

OBJECTIVE

- AIM OF THE EXPERIMENT

The aim of this project is to consume the maximum solar energy through solar panel.

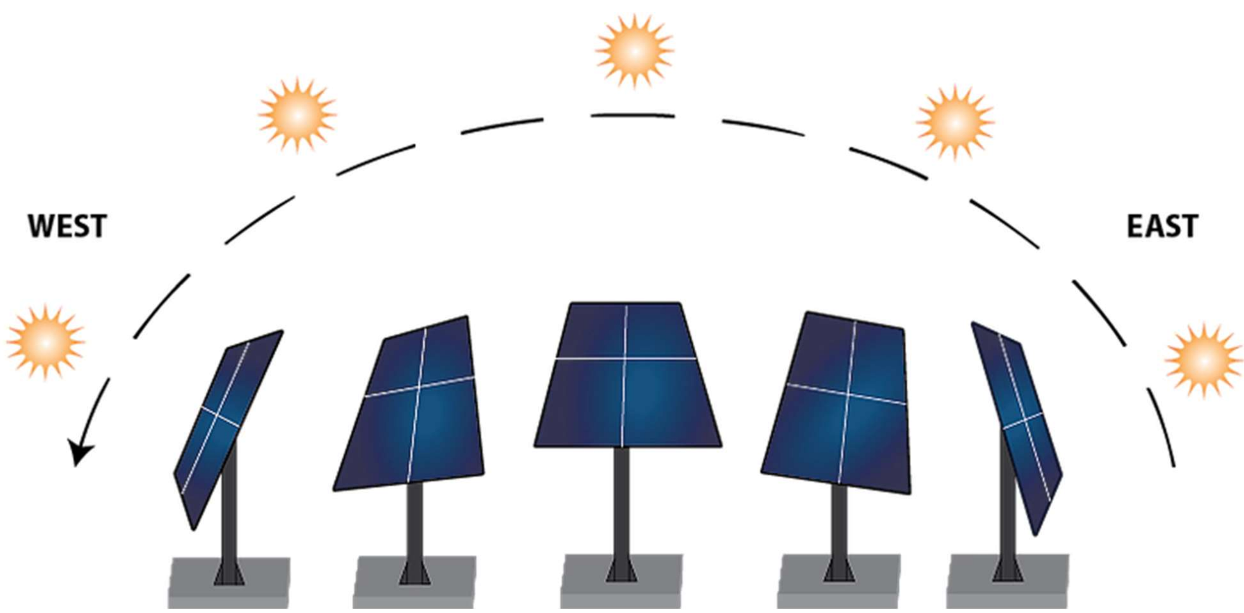


THEORY

Solar energy is an unlimited source of energy which if harnessed properly will get the mankind devoid of using the conventional sources of energy he has been long using -

This project has been designed keeping this in view to make the harnessing of solar energy more efficient.

In this experiment we aim to build and setup a Solar Tracker. A solar tracker is a device which tracks the motion of the sun ensuring maximum amount of sunlight striking the panels.



THEORY

Purpose of this device.

A typical solar panel converts only 30 to 40 percent of the incident solar irradiation into electrical energy. Thus, to get a constant output, an automated system is required which should be capable to constantly rotate the solar panel.

The prototype, which we have designed is capable of solving the problem, mentioned above. The device is completely automatic and keeps the panel in front of sun until that is visible.

Scope of this device.

- It can be used for small and medium scale power generations.
- It can be used for power generation at remote places where power lines are not accessible.
- It can be used for domestic and industrial power backup system.

Definition.

A Solar tracker is an automated solar panel which actually follows the sun to get maximum power. 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.

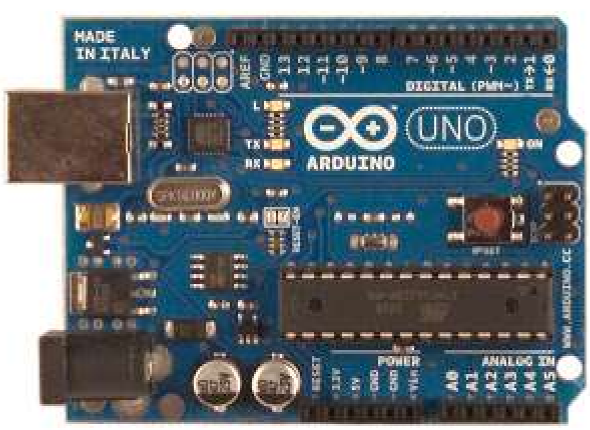
THEORY

Components required.

• Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz



THEORY

• Servo Motor

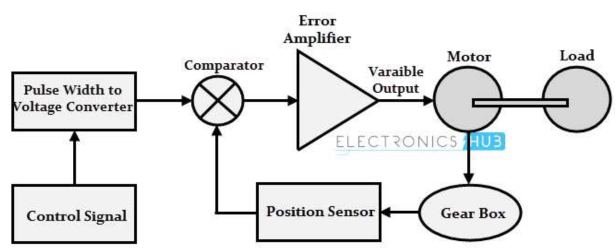
They are actuation devices for the precise control of speed, torque and position. They have a better performance and precision when compared to actuations based on frequency converters, since these do not offer position control and have low effectiveness at low speeds.

A servo motor is a device that contains an encoder which converts the mechanical motion (turns of the shaft) into **digital pulses** interpreted by a **motion controller**. It also contains a driver; and in conjunction, they make up a circuit that governs the position, torque and speed.

Their main specifications are torque and speed.

Servomotors can be found in camera zooms, lift doors, or tools we may have at home.

SERVO MOTOR

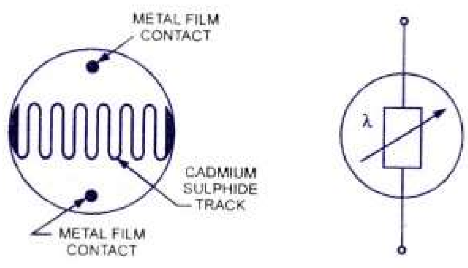


SERVO MOTOR – BLOCK DIAGRAM

THEORY

• LDR

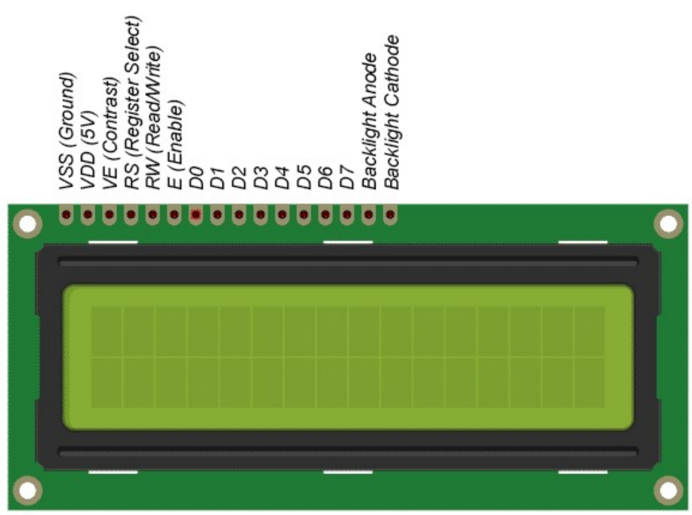
A Light Dependent Resistor (LDR) is also called a photoresistor or a cadmium sulfide (CdS) cell. It is also called a photoconductor. It is basically a photocell that works on the principle of photoconductivity. The passive component is basically a resistor whose resistance value decreases when the intensity of light decreases. This **optoelectronic device** is mostly used in light varying sensor circuit, and light and dark activated switching circuits. Some of its applications include camera light meters, street lights, clock radios, light beam alarms, reflective smoke alarms, and outdoor clocks.



(a) Basic Structure *(b) Symbol*

LDR

• LCD



Data & Observation

BLOCK DIAGRAM.

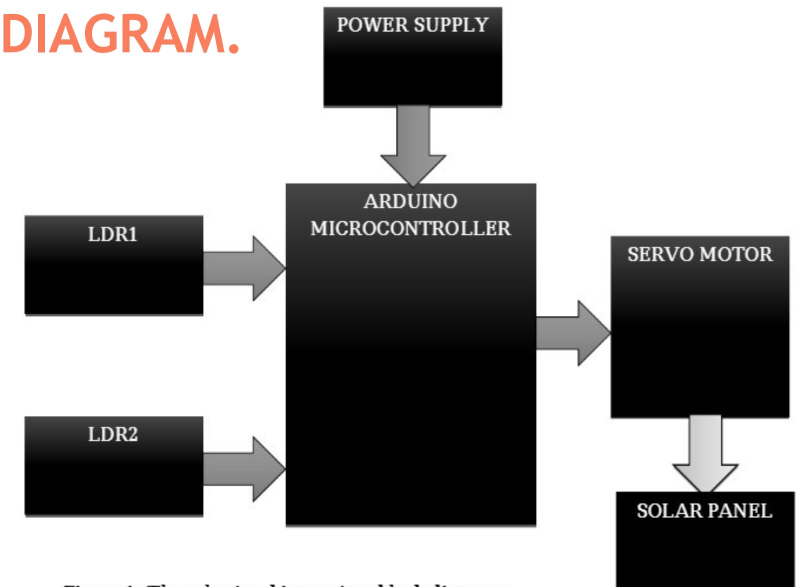


Figure 1: The solar tracking system block diagram.

BLOCK DIAGRAM.

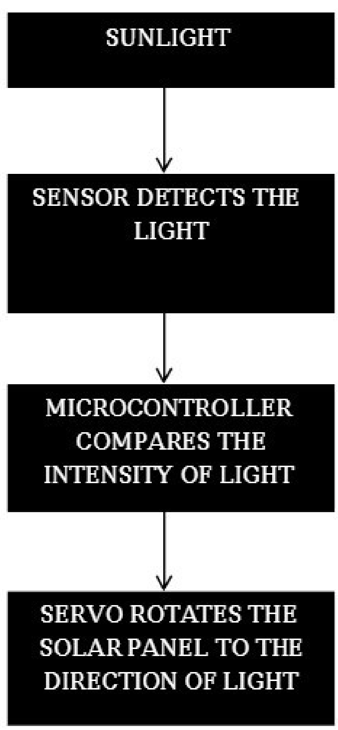
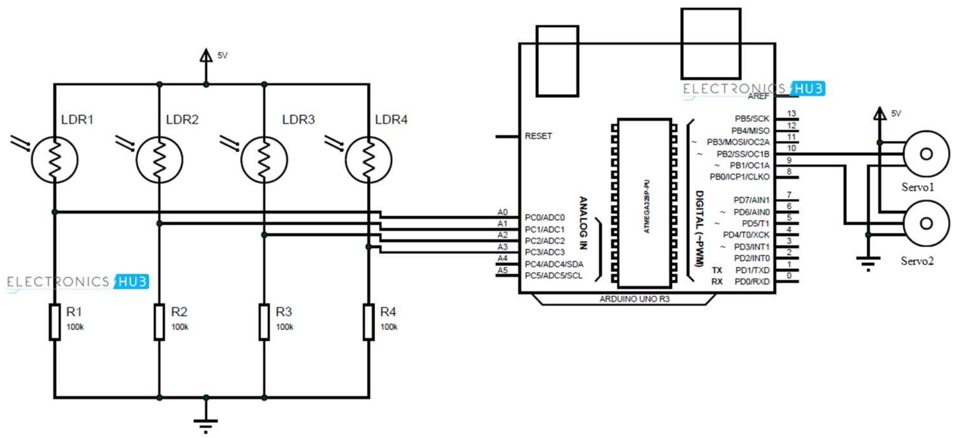


Figure 2: Flow chart diagram of the solar tracking system

Data & Observation

CIRCUIT DIAGRAM.



The circuit design of solar tracker is simple but setting up the system must be done carefully. Four LDRs and Four 10KΩ resistors are connected in a voltage divider fashion and the output is given to 4 Analog input pins of Arduino. The PWM inputs of two servos are given from digital pins 9 and 10 of Arduino.

Working.

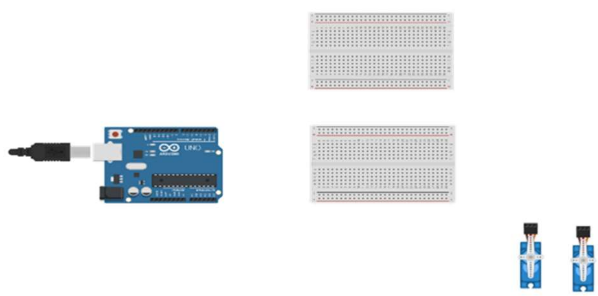
- LDRs are used as the main light sensors. Two servo motors are fixed to the structure that holds the solar panel. The program for Arduino is uploaded to the microcontroller. The working of the project is as follows.
- LDRs sense the amount of sunlight falling on them. Four LDRs are divided into top, bottom, left and right.
- For east – west tracking, the analog values from two top LDRs and two bottom LDRs are compared and if the top set of LDRs receive more light, the vertical servo will move in that direction.
- If the bottom LDRs receive more light, the servo moves in that direction.
- For angular deflection of the solar panel, the analog values from two left LDRs and two right LDRs are compared. If the left set of LDRs receive more light than the right set, the horizontal servo will move in that direction.
- If the right set of LDRs receive more light, the servo moves in that direction.

Data & Observation

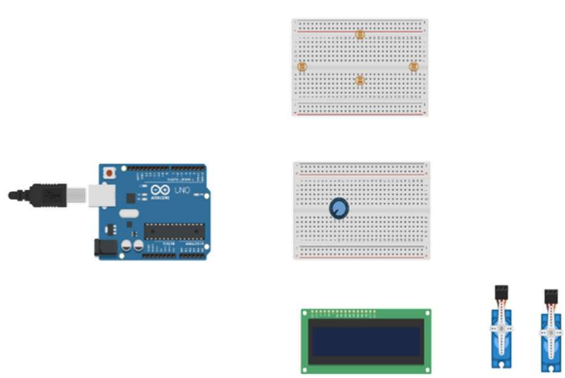
Setup.

To make the circuit, here we are using tinkercad circuits. Open the tinkercad and login in your account. Go to circuits and create new circuits.

STEP 1: In the circuit, click on the left side, component basic and select an Arduino uno. Place two breadboards for the connection of servo motors, 16 *2 LCD and photo resisters. Place two servo motors from the basic components to rotate the solar panel and measure the angle.

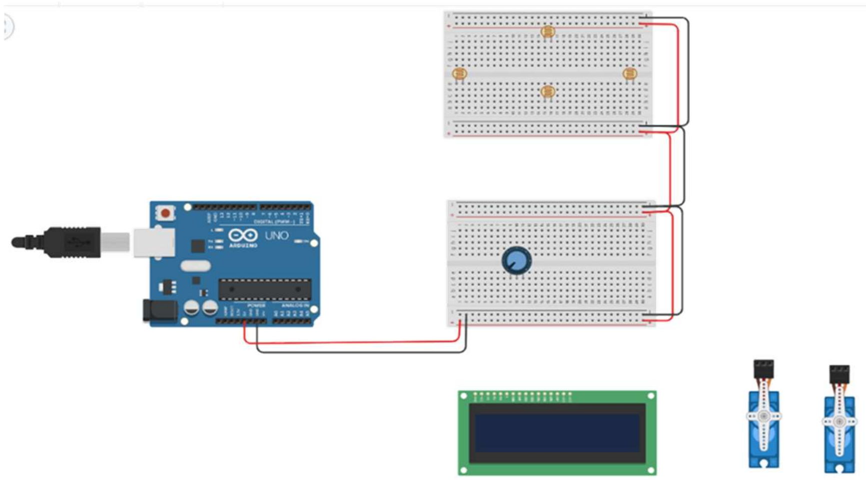


STEP 2: Place four photoresistors from basic components for measuring the intensity of sunlight and give it as analog input into the Arduino board. Place a potentiometer to provide a threshold value for comparison purposes and give analog input to the Arduino board. Place a 16 *2 LCD from All components to display the intensity of light and the horizontal and vertical angle of the servo motor.

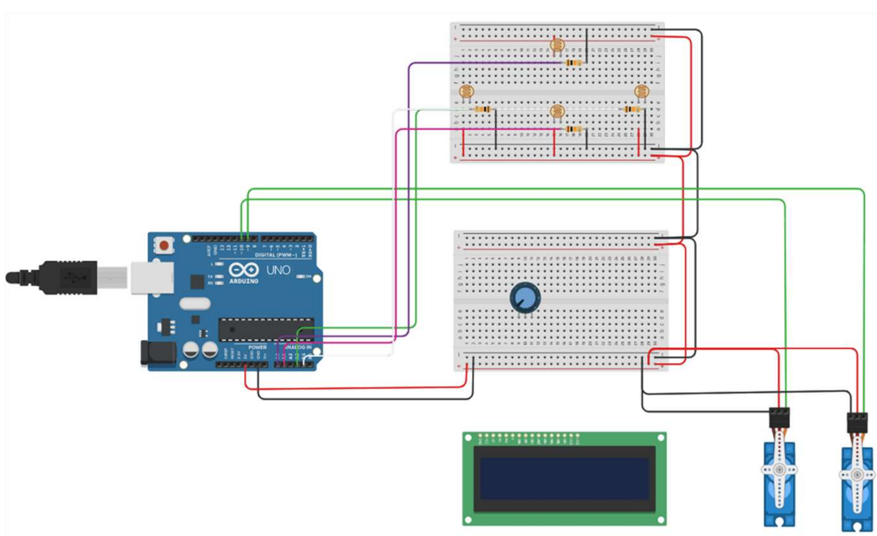


Data & Observation

STEP 3: Start the connection, use ground pin and 5v to provide power and ground to the breadboard. Similarly make connections on another breadboard.

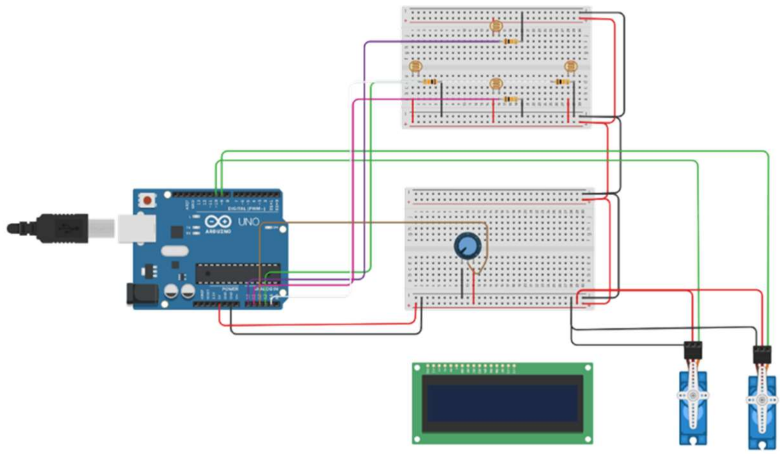


STEP 4: Make the connection of the servo motor to the breadboard and connect the pwm pins of Arduino uno (9,10). This will provide analog output to the servo motors. connect the LDRs to the analog input of Arduino uno (A0, A1, A3, A4).

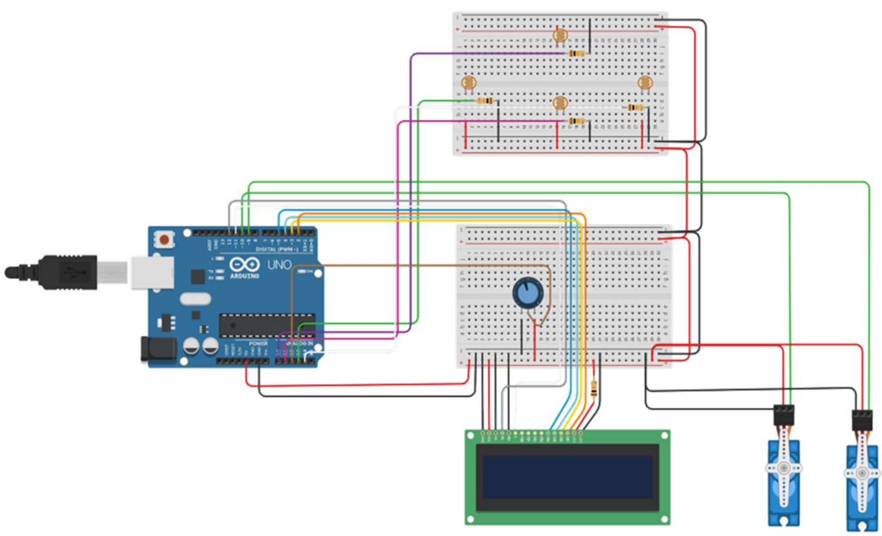


Data & Observation

STEP 5: Connect the potentiometer to give the threshold value and it is connected to the analog input A2.



STEP 6: Connect the LCD to display the intensity of light and the angle of horizontal and vertical movement of servo motors. Connect the LCD using pins (12,11,5,4 ,3,2).

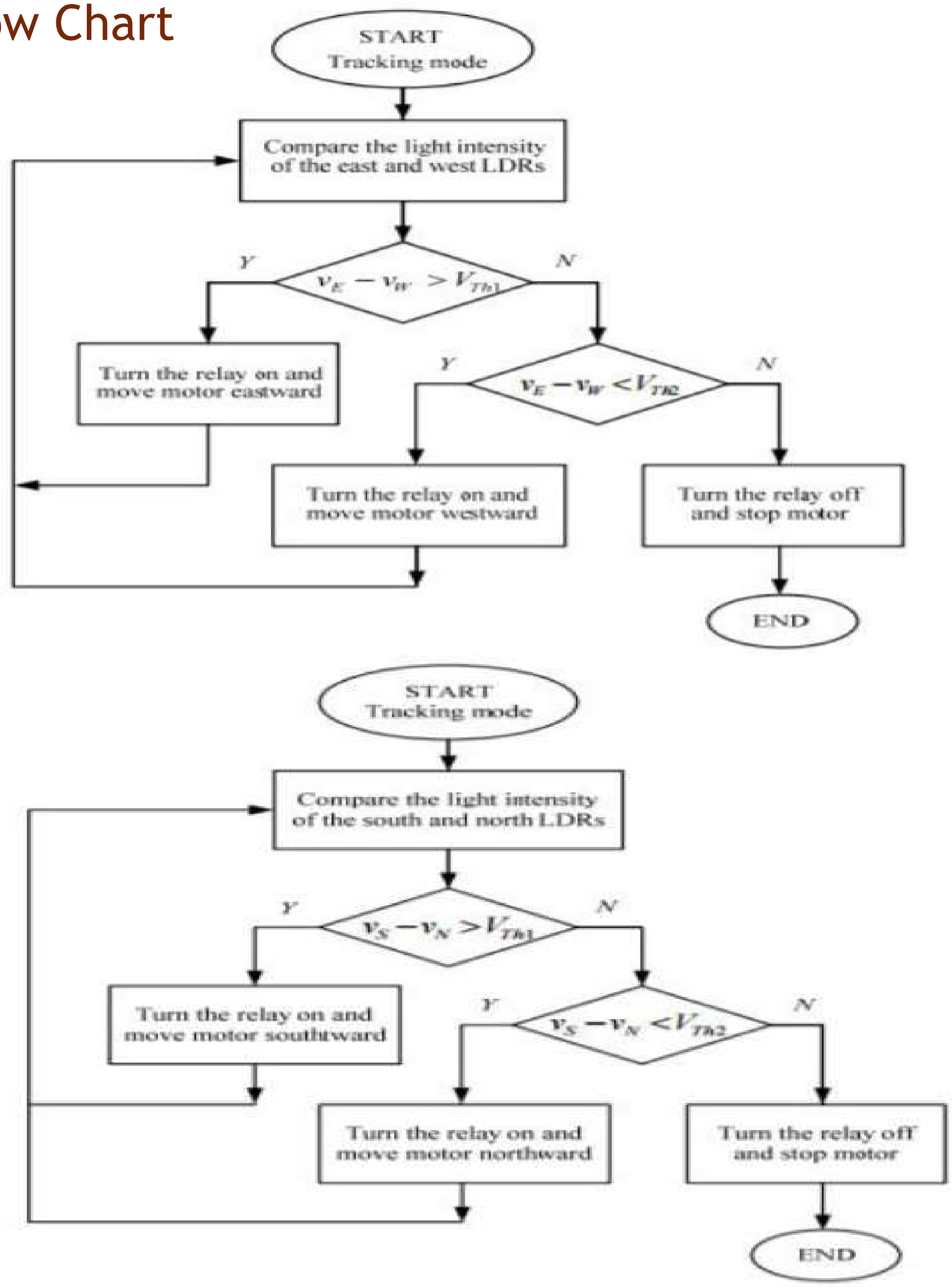


STEP 6: Click on the code and get in the text, write the code and then start simulation.

Data & Observation

Project Code.

- Flow Chart



Flowchart of tracking algorithm for azimuth control and elevation control.

Data & Observation

• Algorithm

Step1: Start

Step2: Initialize all necessary inputs and outputs to zero.

Step3: Assign analog LDR outputs and PWM servomotor inputs to Arduino Uno.

Step4: If center LDR = 0, then delay (longer).

Step5: Check alignment (Simultaneously for north south and east-west)

Step6: If up (LDR) greater than center and down (LDR) lesser than center, then increase position of servomotor1 by 1 unit. Give delay.

Step7: Else if up (LDR) lesser than center and down (LDR) greater than center, then decrease position of servomotor1 by 1 unit. Give delay.

Step8: (Simultaneously along with step6) If right (LDR) greater than center and left (LDR) lesser than center then increase the position of servomotor2 by 1 unit. Give delay.

Step9: Else if right (LDR) is lesser than center and left (LDR) greater than center then decrease position of servomotor2 by 1 unit. Give delay.

Step10: Go to Step 5.

Step11: End.

Data & Observation

- Code

Initialization

```
#include <LiquidCrystal.h>
#include <Servo.h>

// initialize the functon to move the
// sensor up and down
void UpDown();

// initialize the functon to move the
// sensor up and down
void LeftRight();

Servo servo1;
Servo servo2;

// Initialize the library with the numbers
// of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

Setup Function

```
// set up loop
void setup() {

  // Initialize the LCD with 16 characters
  // long on each line and 2 lines.
  lcd.begin(16, 2);

  // conect servo 1 to interface pin 9
  servo1.attach(9);
  // conect servo 2 to interface pin 10
  servo2.attach(10);

  // sets the angle of servo 1 to 90 degrees
  servo1.write(90);
  // sets the angle of servo 2 to 90 degrees
  servo2.write(90);
}
```

Data & Observation

Loop Function

Initialization of sensors.

```
void loop() {

  int Tolerence = 0;// set tolerence to 0

  // set the top LDR input to analog pin 0
  int sensorTop = analogRead(A0);
  // set the bottom LDR input to analog pin 1
  int sensorBottom = analogRead(A1);
  // set the left LDR input to analog pin 3
  int sensorLeft = analogRead(A3);
  // set the right LDR input to analog pin 4
  int sensorRight = analogRead(A4);

  // Compares the difference in top and bottom
  int sensorTest1 = sensorTop - sensorBottom;
  // Compares the difference in top and bottom
  int sensorTest2 = sensorBottom - sensorTop;
  // Compares the difference in left and right
  int sensorTest3 = sensorLeft - sensorRight;
  // Compares the difference in left and right
  int sensorTest4 = sensorRight - sensorLeft;

  // Reads the variable resistor value to get tolerence
  int TolerenceValue = analogRead(A2);
  // sets tolerence value and changes value between 1 and 10
  Tolerence = (TolerenceValue * (5.0 / 1023.0)) * 2;
```

Data & Observation

Logic

```
int avgIntensity = (sensorTop + sensorBottom + sensorLeft + sensorRight) / 4 - 54;
lcd.clear();

// checks to see if sensor test 1 is greater then or equal to the
// tolerance and that sensor test 1 is a positive number
if ((sensorTest1 >= 0) && (sensorTest1 >= Tolerence)) {
  // calls function to move the solar panel up or down towards the sun.
  // Also pass the values of top and bottom sensors
  UpDown(sensorTop, sensorBottom);
}

// checks to see if sensor test 2 is greater then or equal to the
// tolerance and that sensor test 2 is a positive number
if ((sensorTest2 >= 0) && (sensorTest2 >= Tolerence)) {
  // calls function to move the solar panel up or down towards the sun.
  // Also pass the values of top and bottom sensors
  UpDown(sensorTop, sensorBottom);
}

// checks to see if sensor test 3 is greater then or equal to the
// tolerance and that sensor test 3 is a positive number
if ((sensorTest3 >= 0) && (sensorTest3 >= Tolerence)) {
  // calls function to move the solar panel left or right towards the sun.
  //Also pass the values of left and right sensors
  LeftRight(sensorLeft, sensorRight);
}

// checks to see if sensor test 4 is greater then or equal to the
// tolerance and that sensor test 4 is a positive number
if ((sensorTest4 >= 0) && (sensorTest4 >= Tolerence)) {
  // calls function to move the solar panel Left or right towards the sun.
  // Also pass the values of left and right sensors
  LeftRight(sensorLeft, sensorRight);
}

```

```
 int pos1 = servo1.read();
 int pos2 = servo2.read();
 lcd.setCursor(0, 0);
 lcd.print("V: ");
 lcd.print(pos1);

 lcd.setCursor(7, 0);
 lcd.print("H: ");
 lcd.print(pos2);

 lcd.setCursor(0, 1);
 lcd.print("I: ");
 lcd.print(avgIntensity);

 delay(100);
}
```

Data & Observation

Up-Down Movement Function

```
// function to check what direction to move up or down and turn towards the sun.
// Also take the value of top and bottom from the main loop and return nothing.
void UpDown(int sensorTop, int sensorBottom) {

  // reads the current angle of servo 1
  int pos1 = servo1.read();

  // checks to see if the bottom sensor is grater than the top tehn enter the loop
  if (sensorTop < sensorBottom) {
    pos1 = --pos1; // change the angle by -1
  }
  // if the bottom is not greater than the top then enter this loop
  else{
    pos1 = ++pos1;// change the angle by +1
  }

  servo1.write(pos1); // write the new angle to servo 1
}
```

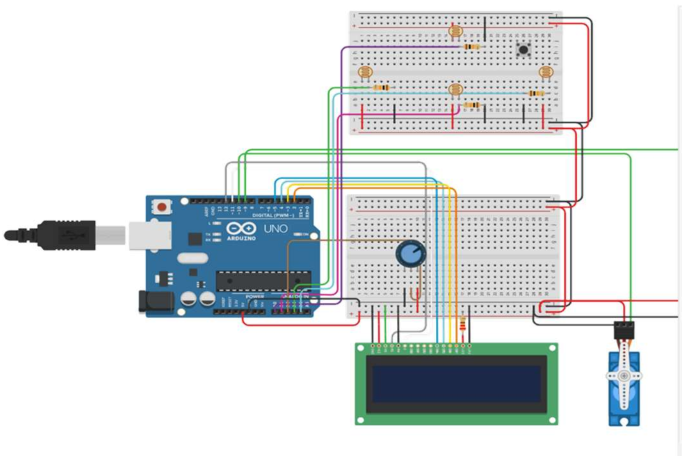
Right-Left Function

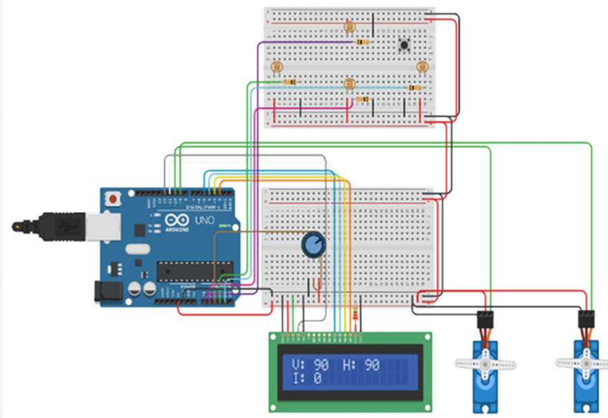
```
// function to check what direction to move left or right and turn towards the sun.
// Also take the value of left and right from the main loop and return nothing.
void LeftRight(int sensorLeft, int sensorRight) {

  int pos2 = servo2.read(); // reads the current angle of servo 1

  // checks to see if the left sensor is grater than the right tehn enter the loop.
  if (sensorLeft < sensorRight) {
    pos2 = --pos2;// change the angle by -1
  }
  // if the left is not greater than the right then enter this loop
  else{
    pos2 = pos2 + 1;// change the angle by +1
  }
  servo2.write(pos2); // write the new angle to servo 1
}
```

Data & Observation

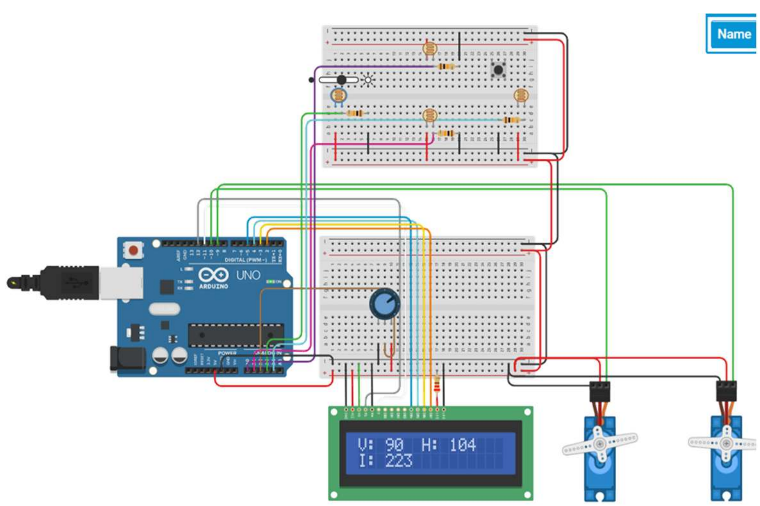
• Output

Finally, after starting the simulation.





Changing the Intensity, the angle will also change automatically and get displayed on the LCD display.



Discussion & Conclusion

Final Discussion

Result of this project is, when light falls on the LDR, its resistance varies and a potential divider circuit is used to obtain corresponding voltage value (5v) from the resistance of LDR. The voltage signal is sent to the Arduino microcontroller. Established on the voltage signal, a corresponding PWM signal is sent to the servo motor which causes it to rotate and to end with attains a position where intensity of light falls on the solar panel is maximum.

Final Conclusion

An Arduino solar tracker was designed and constructed in the current work. LDR light sensors were used to sense the intensity of the solar light occurrence on the photo-voltaic cells panel. Conclusions of this project is summarized as, the existing tracking system successfully sketched the light source even it is a small torch light, in a dark room, or it is the sun light rays. The Arduino solar tracker with servo motor is employed by means of Arduino ATmega328 microcontroller. The essential software is developed via Arduino Uno. The cost and reliability of this solar tracker creates it suitable for the rural usage. The purpose of renewable energy from this paper offered new and advanced idea to help the people.

Thank You

-------- End --------