Nuevo Foundation Workshop

# Sun follower project

**Difficulty: Intermediate - Advanced**

**Pre-requisites: Arduino Light Game, basic electronics understanding**

Contents

[Sun follower project 1](#_Toc188233047)

[Introduction 2](#_Toc188233048)

[Materials needed 2](#_Toc188233049)

[Hardware 2](#_Toc188233050)

[Software 3](#_Toc188233051)

[Setting up the PC 3](#_Toc188233052)

[Building the project 4](#_Toc188233053)

[Building and assembling the cardboard structure 5](#_Toc188233054)

[Connecting the Arduino to the model 8](#_Toc188233055)

[Programming our Sun Follower 9](#_Toc188233056)

[Appendix 9](#_Toc188233057)

[Understanding the Components 9](#_Toc188233058)

## Introduction

A sun follower, or solar tracker, is a device that moves solar panels to follow the sun throughout the day. This helps the panels get more sunlight and produce more energy compared to panels that stay in one place. It's like how a sunflower turns to face the sun!

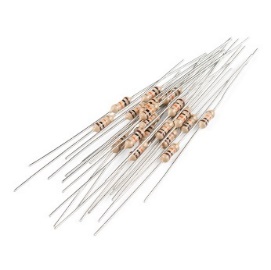
William Kamkwamba, a young innovator from Malawi, built wind turbines and solar trackers from scrap materials to generate electricity and improve life in his village. Inspired by his story, the Nuevo Foundation designed this workshop to teach students about using programming on renewable energy.



## Materials needed

### Hardware

4 Light Dependent Resistors (LDR)

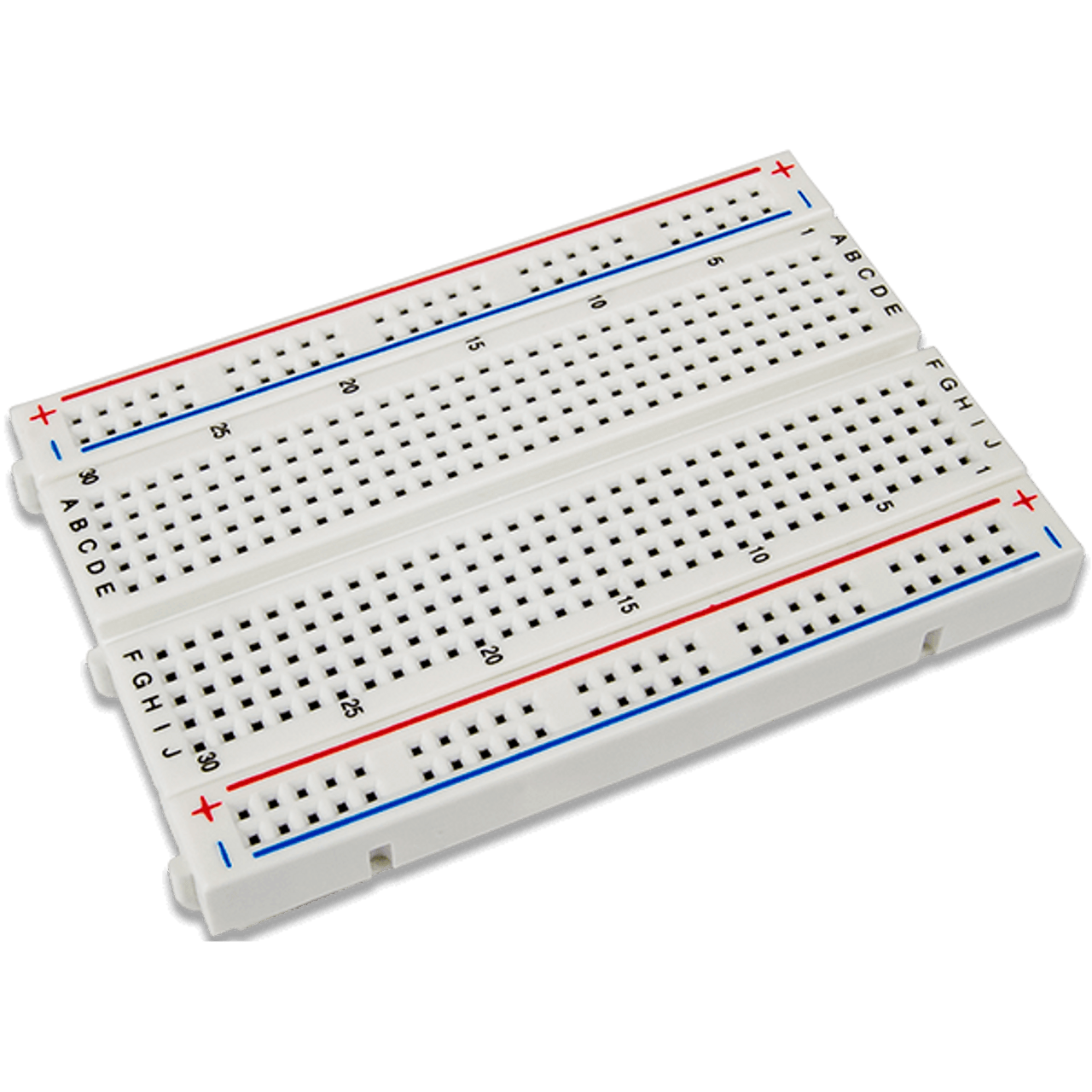


4 Resistors 10kΩ



2 Servo Motors 180° (SG90, MG996R, or any 180° servo)

Arduino Nano Board (or any other Arduino board)



Breadboard and Jumper Wires

Power Supply (will use USB-C in this project, check Voltage requirements)



Base and Frame for Mounting Components (cardboard)



Silicone glue gun



Solar cells (optional)

### Software

Arduino IDE

## Setting up the PC

To set up the PC and the Arduino IDE software, please refer to the “Arduino Light Game” workshop and follow the steps to configure the components and board.

In this case, we are going to be using an Arduino Nano (any other Arduino board can be used). Make sure to select the correct board on the Arduino IDE software.

Try setting up something simple like some LED to turn on and off to test the board is configured correctly.

Additionally to setting up the board, we will need to install a library (“Servo”) to be able to move the servo motors following the next steps:

1. Select the Tools menu, and go to Manage Libraries…

A screenshot of a computer program

Description automatically generated

1. Search for the Servo Library, and select the one for Arduino. Make sure to install the latest version.

A screenshot of a computer

Description automatically generated

Good job setting everything up, now lets get to the project!

## Building the project

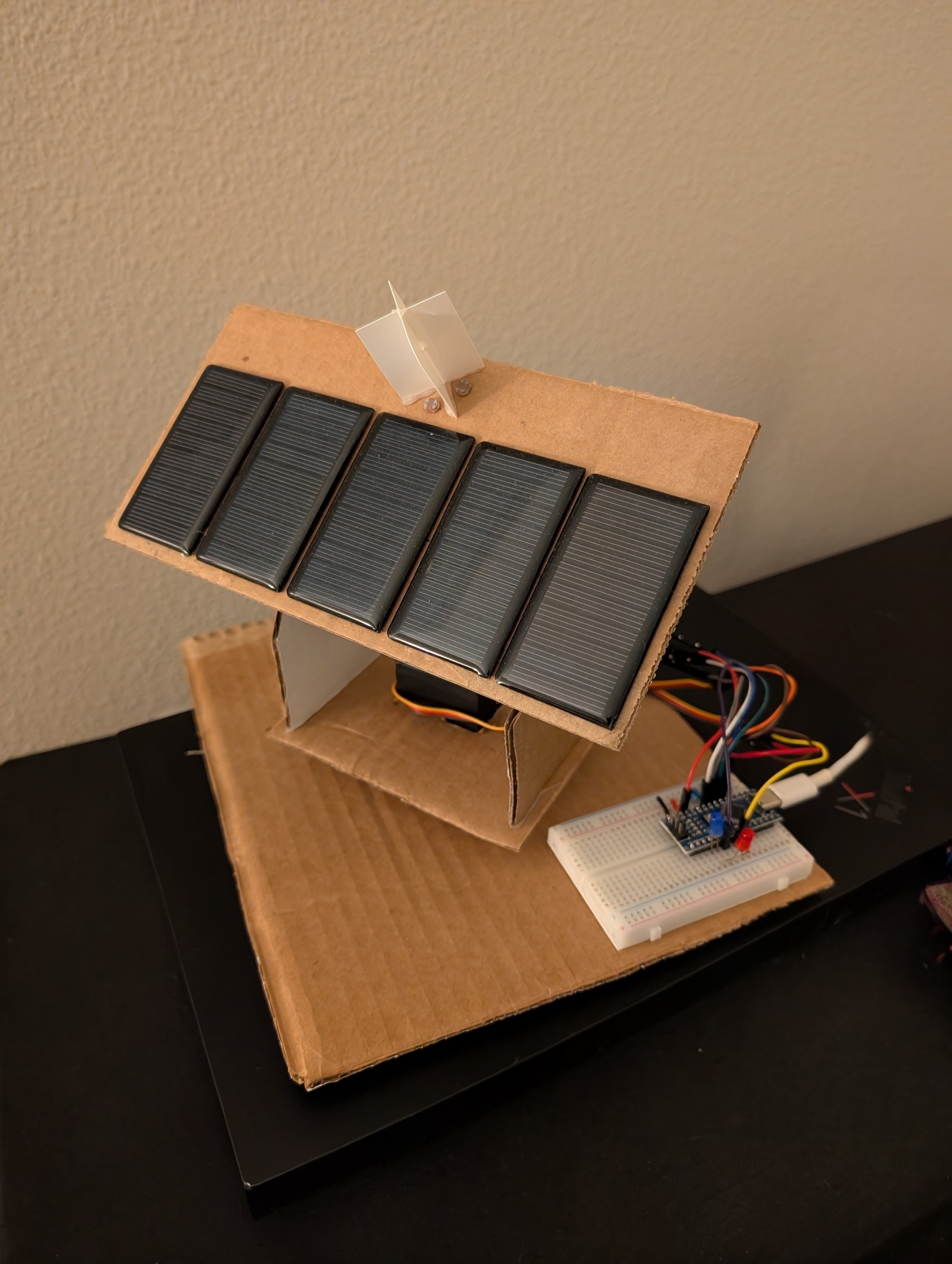
A solar panel on a cardboard box

Description automatically generatedA cardboard box with wires and a solar panel

Description automatically generatedA cardboard box with wires

Description automatically generated

First, we need to understand the basic behavior of our project. We are going to use 4 LDRs positioned in an array to compare the incident light from a source. As we can see in the images, the screen we are using will shade the LDRs depending on its position, changing its resistance. Those changes in resistance will give us the opportunity to program the movement of our servo motors.



### Building and assembling the cardboard structure

To mount the hardware, we will need to build a structure that is able to hold the servo motors and at the same time move our LDR array (or the solar panels) to the source of light.

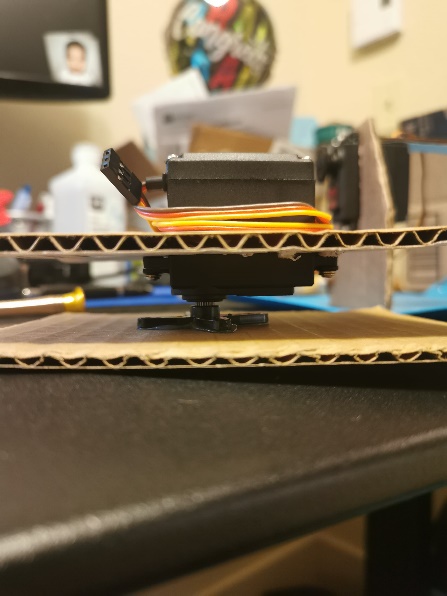
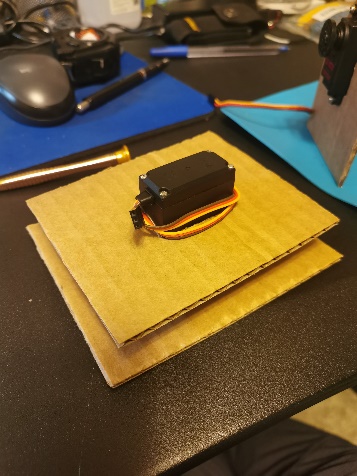
In this project, we used the following cardboard parts to assemble the structure:

A drawing of different shapes

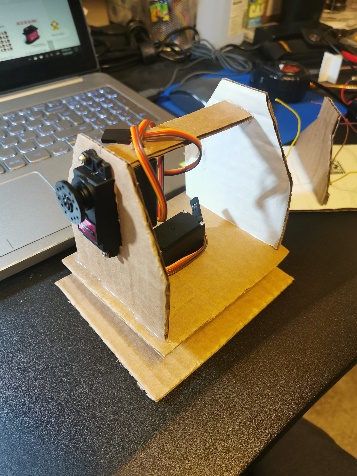
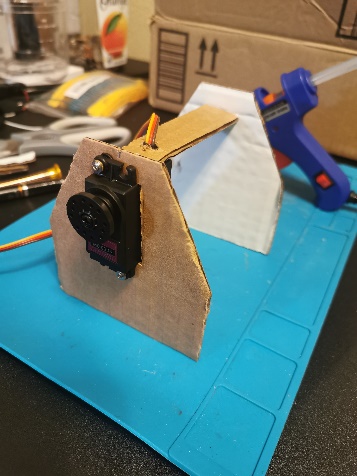
Description automatically generated

We must mention that you can be creative with the design of the structure and that the measurements are made to house 2 MG996R servo motors.

We will start by positioning our first servo motor in the perforated cardboard. We can use the little screws that come with the set or hot glue to hold it in place. Once in place we will add hot glue to the cross like adaptor to hold it to the main base where all the structure will rest.

Next step is to put the other servo motor in the other perforated cardboard, that will serve as our second axis. Once there, we can use some hot glue to stick the long thin cardboard to hold together the two vertical supports of our structure. Once the hot glue dries, we can get it glued over the first cardboard. Almost there!



Last but not least, and a bit tricky to get done as everything else is already assembled, we need to glue our last pieces to the cardboard that will serve as support for the LDR array and the solar panels (optional). I strongly recommend to add the LDRs to the cardboard BEFORE mounting the last parts of cardboard to the main structure, just as in the last couple of images. Preferably use a thin and strong plastic or cardboard to build the screen like LDR divider.

A solar panel on a desk

Description automatically generatedA hand holding a cardboard model of a solar panel

Description automatically generated

A solar panel on a cardboard box

Description automatically generatedA row of solar panels

Description automatically generated

And the structural part is complete! Now we just need to make the respective connections and programming to get the project working!

## Connecting the Arduino to the model

Next we will be doing the appropriate connections between the Arduino board and the model we just created.

The following steps will explain how the project technically works. If most of this doesn´t make sense, just try to connect correctly each of the cables and ports following the next diagram.

A computer diagram of a circuit board

Description automatically generated

1. First of all we get our controller powered by a source (normally a USB source, aka computer), which then will have the power to energize all the other components.
2. We then proceed to power the LDRs and measure the voltage drop across each LDR and its corresponding 10k resistor. The voltage drop will vary depending on the amount of light each LDR receives, as their resistance changes in response to the incident light.
3. Now we compare the values received on each of the measured inputs (A1-A4), and we can give instructions to the servo motors after comparing these values.

LDR1

LDR4

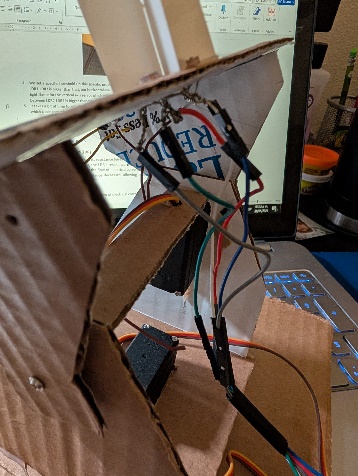
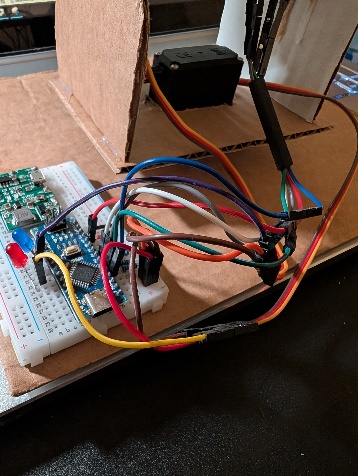
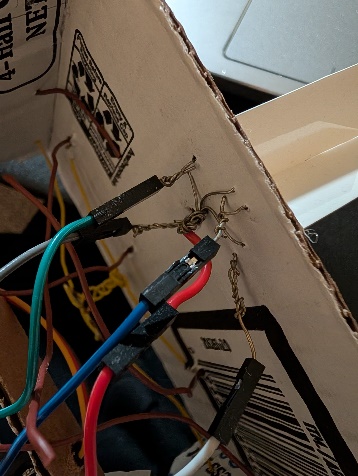
LDR2

LDR3

A solar panel on a cardboard box

Description automatically generated

1. We set a specific threshold (in this specific project 100) and when the difference between LDR1-LDR3 is bigger than that, our horizontal axis servo will move, directing our array to the light. Same for the vertical axis servo, which will be instructed to move when the difference between LDR2-LDR4 is bigger than the threshold.
2. It takes a bit of time to get all the connections properly done, just make sure you identify which cable goes where, as it is easy to loose track of each of the connectors.



## Programming our Sun Follower

For the final step of our project, we will proceed to program our model. We can proceed to copy and paste the following code and our project will be complete!

// J. Ruben Garcia Villalobos

// Solar Tracker Project

// Libraries required

#include <Servo.h>

#include <math.h>

// Define variables and objects

// Servo related variables

Servo ServoH;         // Define Horizontal axis servo (180°)

Servo ServoV;         // Define Vertical axis servo (180°)

int servoAngleH = 0;  // Angle of Horizontal axis Servo

int servoAngleV = 0;  // Angle of Vertical axis Servo

// Declaring LDR related variables

int LDR1 = 0;

int LDR1pin = 1;  // Light Dependent Resistor 1 and Pin

int LDR2 = 0;

int LDR2pin = 2;  // Light Dependent Resistor 2 and Pin

int LDR3 = 0;

int LDR3pin = 3;  // Light Dependent Resistor 3 and Pin

int LDR4 = 0;

int LDR4pin = 4;          // Light Dependent Resistor 4 and Pin

int LDRH = 0;             // LDR 2 & 4 comparative

int LDRV = 0;             // LDR 1 & 3 comparative

int LDR\_threshold = 100;  // Value difference when start to move

int Mov\_Speed = 1;        // Speed at which the servos move (recommended 1)

void setup() {

  //Define Servo pins

  ServoH.attach(3);  // Digital pin 3 on board

  ServoV.attach(5);  // Digital pin 5 on board

  // Define if pins are Inputs or Outputs

  pinMode(LDR1pin, INPUT);

  pinMode(LDR2pin, INPUT);

  pinMode(LDR3pin, INPUT);

  pinMode(LDR4pin, INPUT);

  pinMode(LED\_pin, OUTPUT);

  pinMode(LED\_H, OUTPUT);

  pinMode(LED\_V, OUTPUT);

  // Define default Servo's Angles

  servoAngleH = 60;

  servoAngleV = 90;

  ServoH.write(servoAngleH);

  ServoV.write(servoAngleV);

}

void loop() {

  // Read values from LDRs  0-4095

  LDR1 = analogRead(LDR1pin);

  LDR2 = analogRead(LDR2pin);

  LDR3 = analogRead(LDR3pin);

  LDR4 = analogRead(LDR4pin);

  // LDR array comparative to define movement in H horizontal and V vertical axis

  LDRH = abs(LDR1 - LDR3);

  LDRV = abs(LDR2 - LDR4);

  // Checking if comparatives are bigger than threshold

  if (LDRV > LDR\_threshold || LDRH > LDR\_threshold) {

    // When one or both of the array comparatives is bigger than the threshold, start the servos movement

    // The LDR with the biggest difference starts to move first

    if (LDRV > LDRH) {

      if (LDR2 > LDR4 & servoAngleH < 90 & servoAngleV < 180) {

        servoAngleV = servoAngleV + Mov\_Speed;

        ServoV.write(servoAngleV);

      }

      if (LDR2 < LDR4 & servoAngleH< 90 & servoAngleV > 0) {

        servoAngleV = servoAngleV - Mov\_Speed;

        ServoV.write(servoAngleV);

      }

      // When Horizontal axis angle is more than 90, Vertical axis rotation is inverted

      if (LDR2 > LDR4 & servoAngleH >= 90 & servoAngleV > 0) {

        servoAngleV = servoAngleV - Mov\_Speed;

        ServoV.write(servoAngleV);

      }

      if (LDR2 < LDR4 & servoAngleH >= 90 & servoAngleV < 180) {

        servoAngleV = servoAngleV + Mov\_Speed;

        ServoV.write(servoAngleV);

      }

    }

    if (LDRV < LDRH) {

      if (LDR1 > LDR3 & servoAngleH < 180) {

        servoAngleH = servoAngleH + Mov\_Speed;

        ServoH.write(servoAngleH);

      }

      if (LDR1< LDR3 & servoAngleH > 0) {

        servoAngleH = servoAngleH - Mov\_Speed;

        ServoH.write(servoAngleH);

      }

    }

  }

  delay(30);

}

You did it! You built your own light follower. You can test changing values, speed of movement of the servos or even add some LED lights that activate when each of the servo motors are working. Try to expand and improve the project. There is even several ways to put solar cells connected to a battery and make the follower auto sufficient, the possibilities are endless!

## Appendix

### Understanding the Components

* **Light Dependent Resistors (LDR)**
  + Component that changes its resistance based on the light intensity it detects. In low light or darkness, the LDR's resistance is high, often in the range of megaohms, which limits the flow of electrical current. As the light intensity increases, the LDR's resistance decreases, allowing more current to pass through.
* **Resistors**
  + Resistors work by limiting the flow of electrical current in a circuit.
* **Servo Motors**
  + Servo motors work by receiving a control signal that determines their position. Inside the motor, a small circuit interprets this signal and adjusts the motor's angle accordingly, usually between 0° and 180°.
* **Arduino Nano**
  + Compact, microcontroller board. It is used for building and programming electronic projects.
* **Breadboard**
  + Reusable platforms for building and testing electronic circuits without soldering. They consist of a grid of interconnected holes where you can insert electronic components and wires.
* **Jumper wires**
  + Short, insulated wires with connectors at each end, used to make temporary electrical connections.
* **Power Supply**
  + Device that provides electrical power to a circuit or electronic device.
* **Solar Cells**
  + Devices that convert sunlight directly into electrical energy.