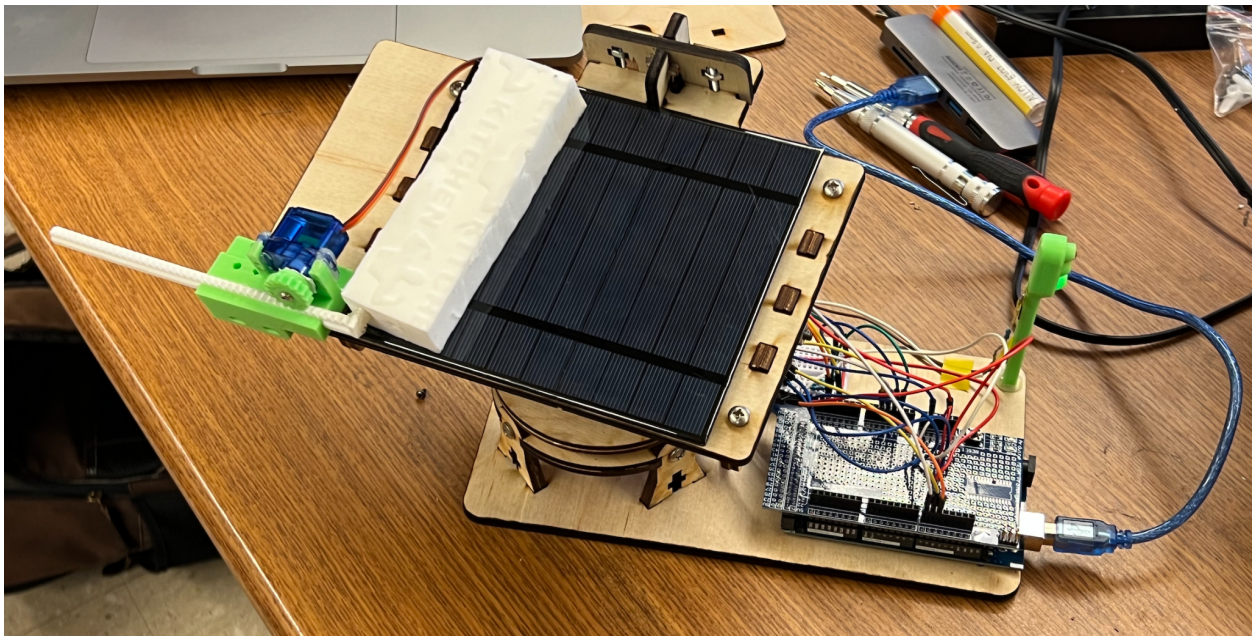
CS 3651 Final Project Documentation

Brooke Christiansen and Ish Mehta

Watch our demo video here.



Hardware

Solar Tracking

We have 4 photoresistors positioned in opposite corners of a rectangle and use the readings from each corner to determine which direction is receiving the most light. Each photoresistor is accompanied by a 10,000 ohms resistor.

We have 2 servos positioned such that they can move the solar panel to follow the direction with the most light.

Self-cleaning Mechanism

We used a fixed-rotation servo from our parts kit that was modified for continuous rotation with the help of Peter. We wired this to run back-and-forth over the solar panel if a “clean” command was entered.

Traffic Light

We used LEDs from our parts kit and connected one of them to the output of our solar panel. This was simply a proof-of-concept for a solar-powered traffic light that would have the logic to cycle through all 3 colors, which we couldn't do because we didn't have something that the output of the solar panel could go into that could handle such logic.

Libraries/Software

We used starter code from a similar solar tracking project, and combined that with logic for the self-cleaning mechanism. No additional code was needed for the traffic light since its voltage came right from the solar panel.

We used the Arduino Servo library to handle attaching and writing to servos. With the modified continuous rotation servo for the self-cleaning mechanism, we simply needed to write a value of 90 to get the servo to be still, with anything greater than 90 and less than or equal to 180 making it retract and anything less than 90 and greater than or equal to 0 making it extend. However, there was a small error (we believe this was a result of the modifications we made to the servo) such that the middle value ended up being 87, not 90.

We used the Serial.available and Serial.readStringUntil functions to take user input for the "clean" command, which simply writes values above and then below 87 to the servo to make it move back and forth for specified amounts of time and writes a value of 87 to make it stop moving.

Custom Parts

Solar Tracking

We modified a file from a similar project for laser cutting the structure for the solar tracker. We made the part where the solar panel sat wider and longer to accommodate the size of our solar panel cleaning mechanism, and we removed some holes that didn't make sense for our design.

Since there wasn't wood of the right dimensions available, two copies of the structure were cut from 0.125 inch wood and glued to create a structure with 0.25 inch thickness. This was done for two main reasons. First, so that the design files we adapted would be easier to use and second to add structural integrity. We also had some issues screwing in the servo horns to the holes that we had laser cut, so we had to add more holes to accommodate this. Pictures of the laser-cut files can be seen in the Appendix below.

Self-cleaning Mechanism

The linear servo actuator was modified in Tinkercad to include a long pointy part at the end of the pusher which could pierce the sponge and move it along the solar panel.

Traffic Light

We used this 3D print file for the traffic light (modified to add a star for the 3D printing part of skill demo 8). After printing, we glued it onto the base of the solar tracker.

New Skills

Ish

Throughout this project, I learned several skills across a range of topics from mechanical engineering to electrical engineering. We learned how to put the structure together using the resources we had, which aren't always the ones we need. Moreover, we learned a bit more about how to use multiple photoresistors together to achieve our desired result. This took a little bit of experimentation and trial and error to tune our hyper-parameters so that the solar tracker moves smoothly.

Brooke

For this project, the skills I used outside of what we had learned in skill demos included using Tinkercad to modify/combine 3D shapes, modifying a servo, gluing and screwing together laser cut parts, and troubleshooting issues with gears and screws.

Process

Solar Tracking

1. Played around with IR sensors and photoresistors on breadboard
2. Modified laser cut file from a similar project for structure, holes, etc., before cutting and assembling
3. Added servos and photoresistors wired to breadboards
    a. Had issues with servo horns that required using a hammer (with the help of Peter) to hand make some holes in our structure
    b. Got photoresistor-triggered movement to work
4. Added solar panel
5. Soldered and continued testing
6. Re-laser cut to accommodate self-cleaning mechanism and make space for Arduino at base

Self-cleaning Mechanism

1. Decided on mechanism and cleaning tool
    a. Settled on this linear servo actuator 3D print file after Googling
    b. Decided sponge would be easier to find and work with than a brush or other cleaning tool
    c. Modified actuator pusher to add long pointy part to stick in sponge
2. 3D printed and assembled actuator with sponge
3. Wired and coded
    a. Modified (for continuous rotation) and attached servo
    b. Used servo library and added conditional logic for extending and retracting actuator
    c. Tested but had issues with servo-rotated gear staying in place/moving pusher
        i. Resolved by hot gluing gear onto servo with a piece of paper in between
4. Glued to solar tracker

Traffic Light

1. Tested wiring of LEDs on breadboard
    a. Realized that we wouldn't be able to cycle through different traffic light colors with the solar panel as the voltage source
        i. We later realized that the Arduino would not have been able to handle the traffic light logic while also taking readings from the photoresistors on the solar tracker anyways, so it was fine that we just had 1 color working
2. Tested powering LED with solar panel
3. 3D printed and assembled traffic light
4. Wired/soldered and tested
5. Glued to solar tracker base and taped down loose wires

Appendix A - Code

Our latest code can be found in this Github Repo. It is also pasted below for reference:

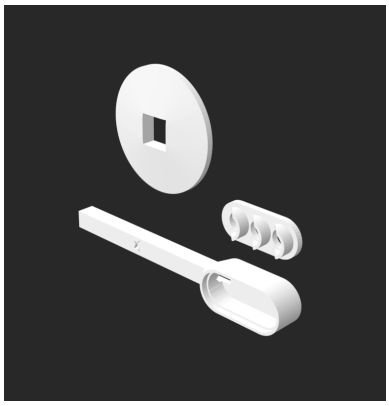
```
#include <Servo.h> // include Servo library

// 180 horizontal MAX
Servo horizontal; // horizontal servo
int servoh = 180;   // 90;     // stand horizontal servo

int servohLimitHigh = 180;
int servohLimitLow = 65;

// 65 degrees MAX
Servo vertical;   // vertical servo
int servov = 50;    //   90;     // stand vertical servo

int servovLimitHigh = 80;
int servovLimitLow = 15;

// LDR pin connections
// define name analog_pin
#define ldrlt A3 //LDR top left
#define ldrrt A2 //LDR top right
#define ldrld A4 //LDR down left
#define ldrrd A5 //LDR down right

#define CLEANER_SERVO_PIN 8
#define HORIZONTAL_SERVO_PIN 9
#define VERTICAL_SERVO_PIN 10

Servo cleaner_servo; // servo controller (multiple can exist)
String cleaner_input_command;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  // servo connections
  horizontal.attach(HORIZONTAL_SERVO_PIN);
```

```
  vertical.attach(VERTICAL_SERVO_PIN);

  // set default servos to initial position
  horizontal.write(90);
  vertical.write(45);
  delay(3000);

  cleaner_servo.attach(CLEANER_SERVO_PIN); // start servo control
  cleaner_servo.write(87); // start servo at 0

  // pinMode(RED_LIGHT_PIN, OUTPUT);
  // pinMode(YELLOW_LIGHT_PIN, OUTPUT);
  // pinMode(GREEN_LIGHT_PIN, OUTPUT);
}

void loop() {
  // Read all photo resistors
  int lt = analogRead(ldrlt); // top left
  int rt = analogRead(ldrrt); // top right
  int ld = analogRead(ldrld); // down left
  int rd = analogRead(ldrrd); // down rigt

  int dtime = 10;
  int tol = 40;

  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 (-1*tol > dvert || dvert > tol) // check if the diffirence is in
the tolerance else change vertical angle
  {
  if (avt > avd)
  {
    servov = ++servov;
```

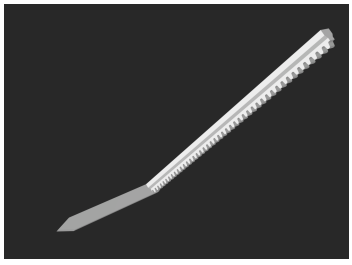
```
     if (servov > servovLimitHigh)
     {
      servov = servovLimitHigh;
     }
  }
  else if (avt < avd)
  {
    servov= --servov;
    if (servov < servovLimitLow)
  {
    servov = servovLimitLow;
  }
  }
  vertical.write(servov);
  }

  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)
    {
      // nothing
    }
    horizontal.write(servoh);
```

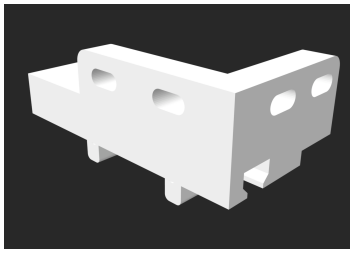
```
  }

  if (Serial.available()) {
    cleaner_input_command = Serial.readStringUntil('\n');
    // Serial.println(cleaner_input_command + "input");
    if (cleaner_input_command == "clean") {
        cleaner_servo.write(120);
        delay(5000);

        cleaner_servo.write(60);
        delay(5000);

        cleaner_servo.write(87);
    }
  }

  delay(dtime);
}
```

Appendix B - Design Files

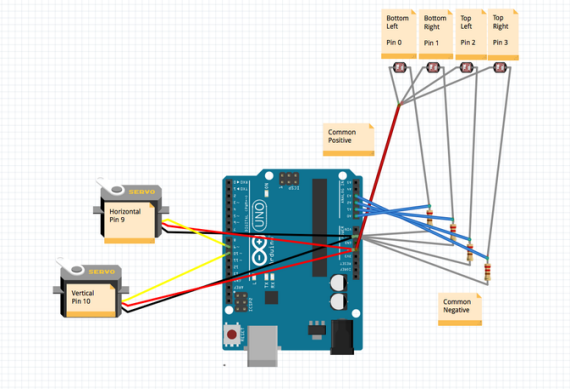


Traffic light design modified from this 3D print file

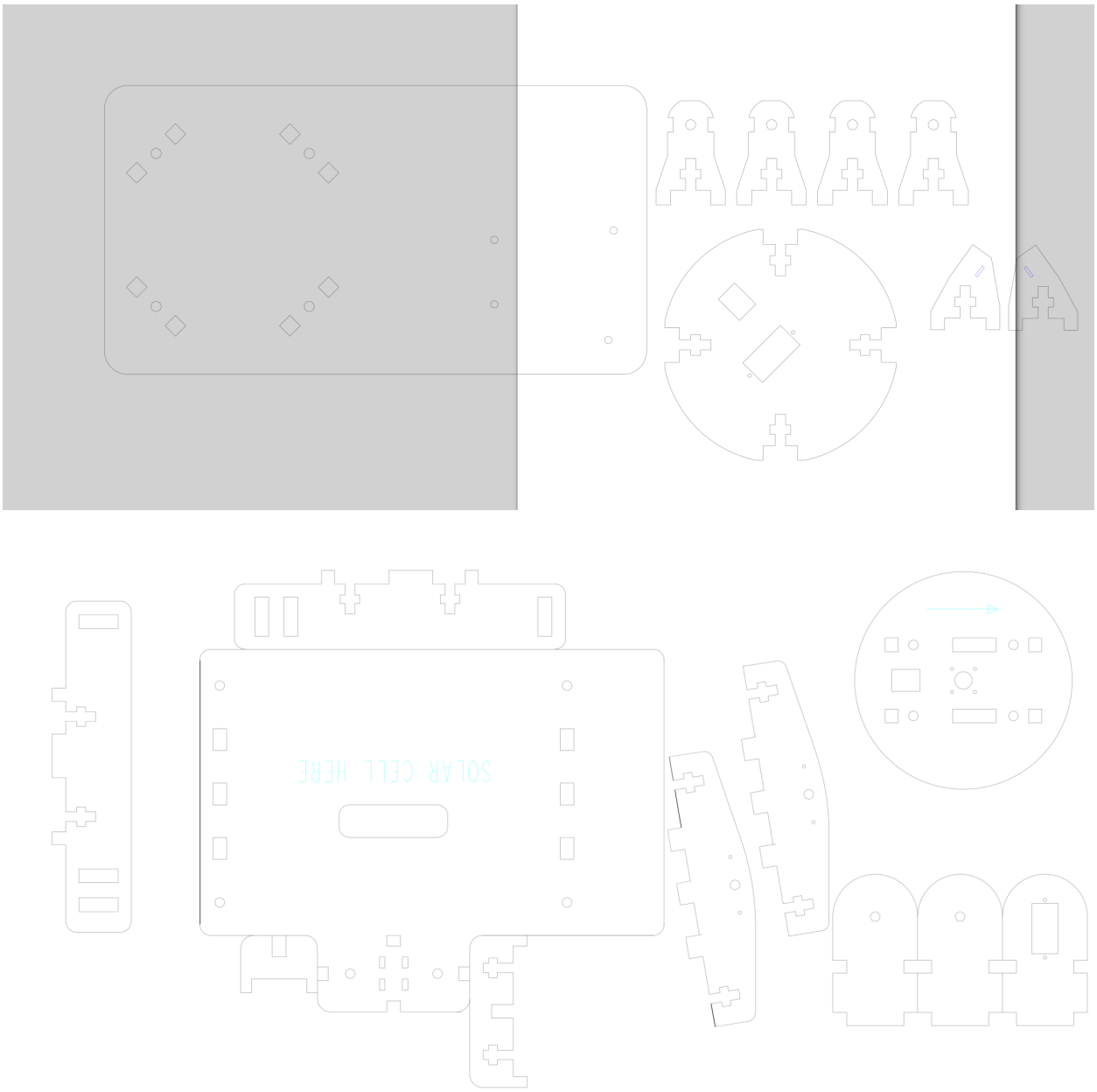




Linear servo actuator design modified from this 3D print file



Solar tracking wiring schematic from this tutorial



Solar tracker structure designs modified from [this file]

References

https://github.com/BrownDogGadgets/SolarTracker/blob/master/Dual%20Axis%20Tracker/Dual_Axis_Tracker_V2.ino

https://www.instructables.com/Simple-Dual-Axis-Solar-Tracker/

https://www.arduino.cc/reference/en/libraries/servo/

Free STL file Linear Servo Actuators・3D printing template to download・Cults

https://projecthub.arduino.cc/123samridhgarg/6ab53511-48b2-4798-82dc-14c69e9b571b