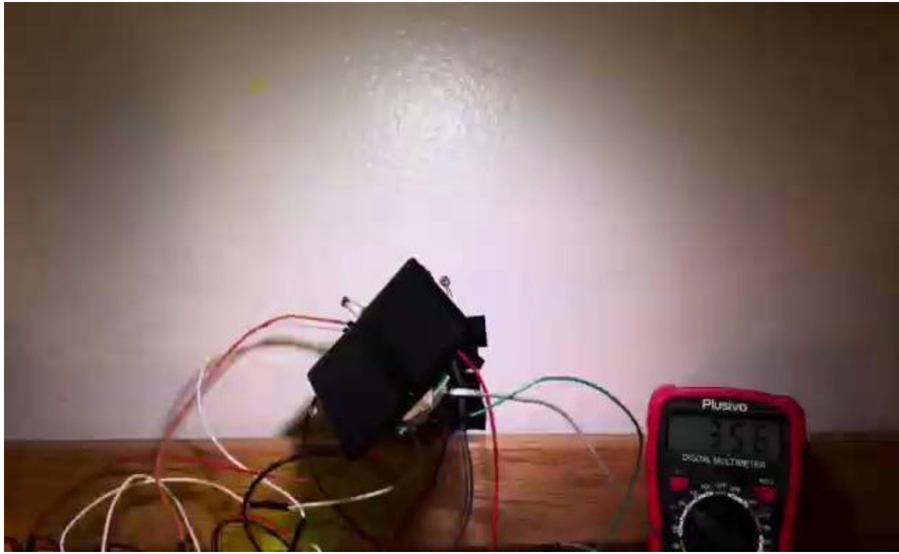
Dual Axis Solar Tracking System

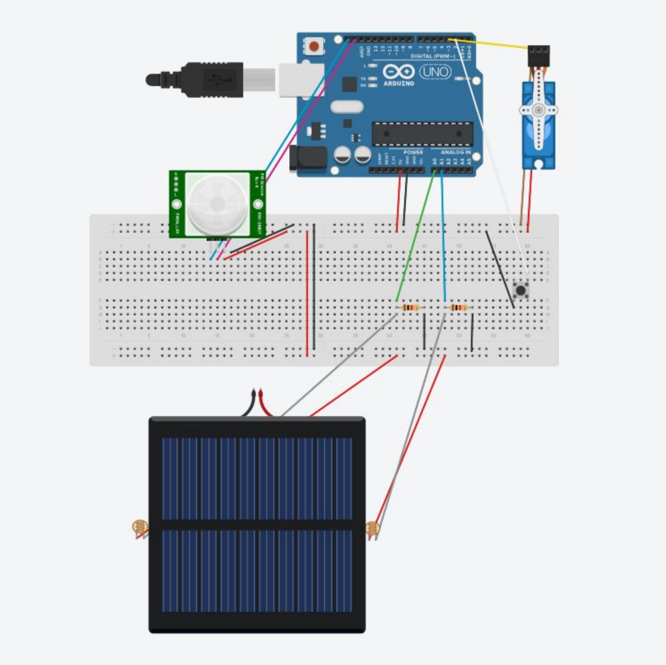
Evie, Feiyang, Bobby

Demo Video

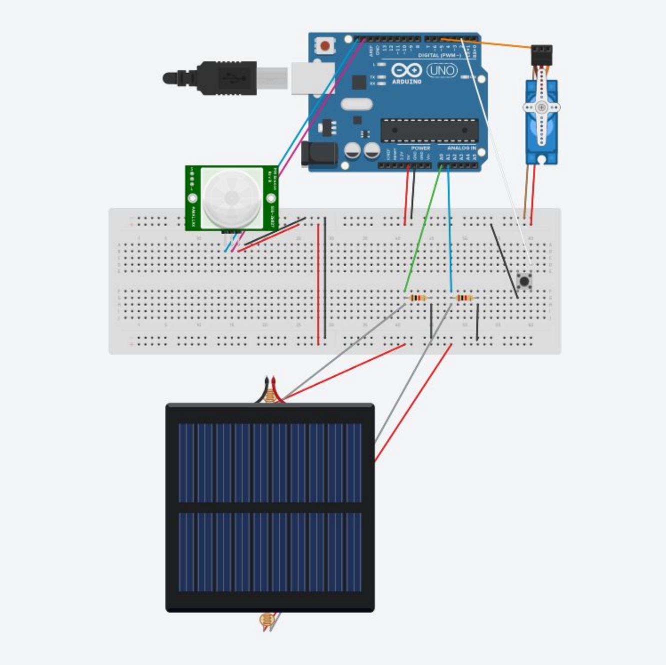


Sensor-Actuator-Microcontroller Integration

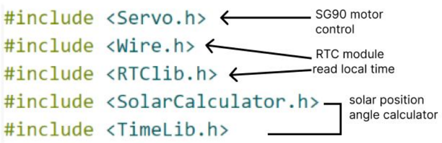
Top motor for elevation tracking:



Bottom motor for azimuth tracking:



Algorithm



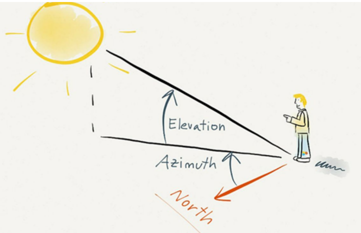
```
#include <Servo.h>
#include <Wire.h>
#include <RTClib.h>
#include <SolarCalculator.h>
#include <TimeLib.h>
```

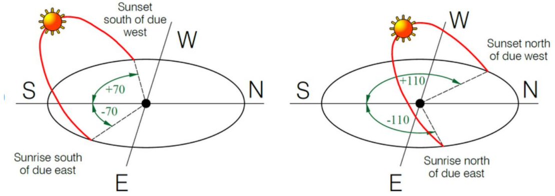
SG90 motor control

RTC module read local time

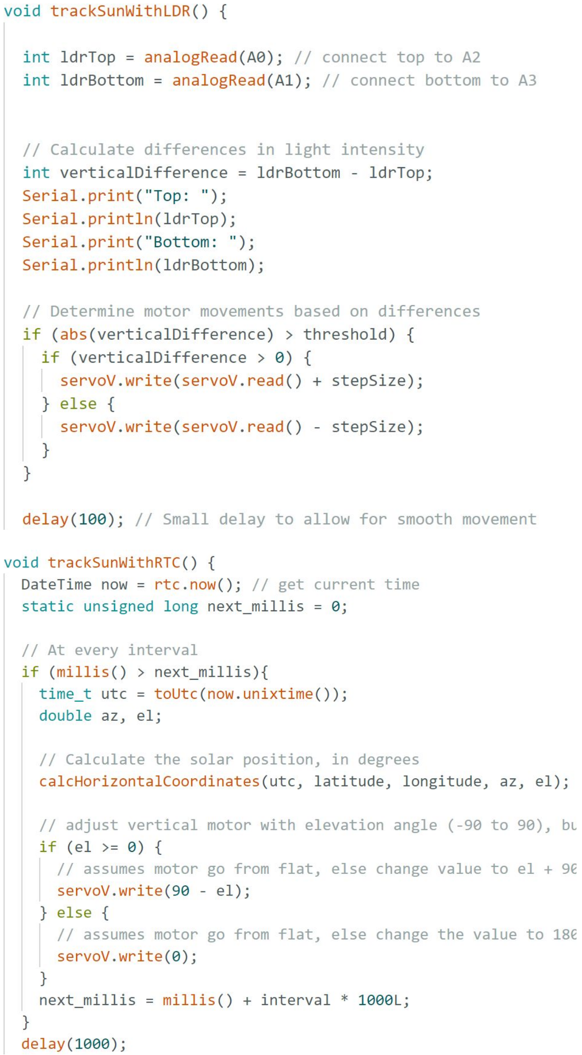
solar position angle calculator

- A button to switch between 2 modes in loop (RTC mode and LDR mode)
- Two arduino sketches (1 for each motor, enable dual-axis tracking)
- RTC Mode
    - Use library to calculate elevation and azimuth angle of sun at predefined location (longitude and latitude) and current time (from RTC)
    - Adjust angle according to available range of our motor and physical setup
- LDR Mode
    - 4 LDRs for different directions (2 to control each motor, each pair controls one axis)





(a) Sun route during winter, (b) Sun route during summer.



```
void trackSunWithLDR() {

  int ldrTop = analogRead(A0); // connect top to A2
  int ldrBottom = analogRead(A1); // connect bottom to A3


  // Calculate differences in light intensity
  int verticalDifference = ldrBottom - ldrTop;
  Serial.print("Top: ");
  Serial.println(ldrTop);
  Serial.print("Bottom: ");
  Serial.println(ldrBottom);

  // Determine motor movements based on differences
  if (abs(verticalDifference) > threshold) {
    if (verticalDifference > 0) {
      servoV.write(servoV.read() + stepSize);
    } else {
      servoV.write(servoV.read() - stepSize);
    }
  }

  delay(100); // Small delay to allow for smooth movement

void trackSunWithRTC() {
  DateTime now = rtc.now(); // get current time
  static unsigned long next_millis = 0;

  // At every interval
  if (millis() > next_millis){
    time_t utc = toUtc(now.unixtime());
    double az, el;

    // Calculate the solar position, in degrees
    calcHorizontalCoordinates(utc, latitude, longitude, az, el);

    // adjust vertical motor with elevation angle (-90 to 90), bu
    if (el >= 0) {
      // assumes motor go from flat, else change value to el + 90
      servoV.write(90 - el);
    } else {
      // assumes motor go from flat, else change the value to 180
      servoV.write(0);
    }
    next_millis = millis() + interval * 1000L;
  }
  delay(1000);
```

Challenges

Power Supply:

- Problem: 9V battery didn't power a 5V servo due to insufficient current.
- Solution: Used 2 separate Arduinos and RTC modules, one of each per motor, and adjusted our code.

360° Mount:

- Problem: The mount didn't move 360 degrees as expected.
- Solution: RTC mode turns halfway at the sun's peak and LDR mode is best when mounted sideways.
- Problem: Screws loosen as servo moves.
- Solution: Frequently secure screws more firmy.

Solar panels:

- Problem: Solar panels didn't come with pre-soldered wires.
- Solution: Took time but figured out the correct way for max voltage output (though Avanti burnt herself multiple times).

Energy Output Storage:

- Problem: Significant voltage output difference in artificial and natural light & insufficient current to power up rechargeable batteries.
- Solution: Used a multimeter to display voltage output and a white LED to show functionality.

Presentation Video