DUAL AXIS SOLAR TRACKER

Ishneet Singh[1], Samuel Leong[1] and Denise Mok[1]

Mentor(s): Mr Ganesh Jaga Daseen[2]

[1] Temasek Junior College, 22 Bedok South Road, 469278
[2] CRADLE, Science Centre, 15 Science Centre Rd, 609081

INTRODUCTION

Due to climate change, burning fossil fuels for energy is no longer a feasible long-term method as the greenhouse gases emitted during the process would highly contribute to global warming. Hence, new renewable and more clean sources of energy are recommended, one being solar energy. For solar panels to be the most effective in solar-system energy production, the solar panel would have to always be oriented to the sun no matter the time. The dual axis solar tracker would allow for that and hence maximize the amount of clean energy produced from solar energy and would make solar panels a more promising alternative to non-renewable sources of energy. The basis of a dual axis solar tracker is to have two rotating motors, one moving the panel's orientation and one moving the panel's angle. We were given the option to choose between using coordinates to find the sun's position and move the tracker according to that, or to use LDRs and using the difference in light reading to move the tracker. We realised that the former option had limitations, for example, the tracker would still face the same direction despite the presence of obstructions between the sun and the tracker. Hence, we decided to use LDRs in our tracker instead.

MATERIALS AND METHODS

In order to construct the solar tracker, we would be using an Arduino which would send instructions to the Servo motors which would move a specific angle so that the solar panel would move in the direction of the Sun. We would also be using libreCAD to design the prototype and then printing it using laser cutting. After doing the literature review for the research problem, we decided that there are 2 ways to tackle this research problem. In the first approach, a set of longitude and latitude in the geographic coordinate system and local time would be required to calculate the solar position specified by the Sun's azimuth and altitude angle. This would be done by the code written in Arduino language and the code will send instructions to the Arduino to move the servo motors according to the azimuth and altitude angle. In the second approach, we would have to make an electrical circuit on a PCB/ breadboard which would allow the light dependent resistors (LDRs) to be connected to the Arduino. These LDRs then would measure the light intensity of the surroundings and send this information back to the Arduino which would then instruct the Servo motors to move in the direction with the greatest light intensity and hence allow the dual axis solar tracker to move to the solar position and hence allowing maximum energy output for the solar panel. In the end after discussions, we decided to pursue the second approach as it has lesser limitations than the first one. In the first approach, obstructions in way between the solar panel and the sun would not be taken into account as the code would just instruct the motor to move in the direction of the Sun and hence on a very cloudy day or if some object is hindering the path for the solar panel to the Sun, the solar panel would still be there and the maximum output for the solar panel wouldn't be achieved whereas in the second approach, the solar tracker system would ensure the solar panel is always in the direction of greatest light intensity due to the LDRs giving input to the Arduino and this would allow for greatest electrical energy to be harnessed from the solar panel. To construct the LDR-based dual axis solar tracker, we would need to have 2 servos, one responsible for the vertical motion of the tracker and the other responsible for the horizontal motion of the tracker. We would also need 4 of 220Ω resistors and 4 LDRs and M-M jumper wires to connect the LDRs to the analog pins in the Arduino and to complete the electrical circuit on the breadboard. We decided to use a breadboard instead of a PCB as it would be easier to troubleshoot where the errors in the circuitry are and fix them and if any electrical components are damaged it is easier to replace them whereas in a PCB all the electrical components are soldered onto the circuit board and it is harder to replace them and in the midst of removing the solder and replacing the parts, the circuit board could also be damaged or the electrical components could be damaged. Hence taking all of this into consideration we decided to prefer breadboards over PCB. We also needed a shield for the Arduino on top of which the breadboard would be mounted.

Code for the Arduino

For the code we took reference from an online source and changed some parts in the code to fit our design for the tracker [1].

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

Servo horizontal; // creates horizontal servo
int servoh = 180;// set initial angle for horizontal servo
int servohLimitHigh = 180;//limits for servo horizontal angle
int servohLimitLow = 65;



Servo vertical; // creates vertical servo
int servov = 45;//set initial angle for vertical servo
int servovLimitHigh = 80;//limits for servo vertical angle
int servovLimitLow = 15;
int analogTopLeft = 0; // These will be used in analogRead
int analogTopRight = 1;
int analogBottomLeft = 2;
int analogBottomRight = 3;


void setup() {
    Serial.begin(9600);
    horizontal.attach(9); // attach horizontal motor to pin 9
    vertical.attach(10); // attach vertical motor to pin 10
    horizontal.write(180);
    vertical.write(45);
    delay(2000);
}


void loop() {
    int topLeft = analogRead(analogTopLeft); // reads analog A0
    int topRight = analogRead(analogTopRight); // reads analog A1
```

```
    int bottomLeft = analogRead(analogBottomLeft); // reads analog A2
    int bottomRight = analogRead(analogBottomRight); // reads analog A3
    int delayTime = 10;// time delay between each set of servo movement
    int tolerance = 30;
    int averageTop = (topLeft + topRight) / 2; // top average value
    int averageBottom = (bottomLeft + bottomRight) / 2; // bottom average
value
    int averageleft = (topLeft + bottomLeft) / 2; // left average value
    int averageRight = (topRight + topBottom) / 2; // right average value
    int differenceVertical = averageTop - averageBottom; // check the
difference of up and down
    int differenceHorizontal = averageleft - averageRight; // check the
difference of left and right
    Serial.print(averageTop);
    Serial.print(" ");
    Serial.print(averageBottom);
    Serial.print(" ");
    Serial.print(averageleft);
    Serial.print(" ");
    Serial.print(averageRight);
    Serial.print(" ");
    Serial.print(delayTime);
    Serial.print(" ");
    Serial.print(tolerance);
    Serial.println(" ");
    if (-1*tolerance > differenceVertical || differenceVertical > tolerance){
// check if the difference is in the tolerance range else change vertical
angle
        if (averageTop > averageBottom) {
            servov = ++servov;//increase vertical angle
            if (servov > servovLimitHigh) {
                servov = servovLimitHigh;//ensure vertical angle does not
cross limit
            }
        }
        else if (averageTop < averageBottom) {
            servov= --servov;
            if (servov < servovLimitLow) {
                servov = servovLimitLow;
            }
        }
        vertical.write(servov);//tells vertical servo to move accordingly
    }
//same code for the horizontal motor
    if (-1*tolerance > differenceHorizontal || differenceHorizontal >
tolerance) {
        if (averageleft > averageRight){
            servoh = --servoh;
            if (servoh < servohLimitLow) {
                servoh = servohLimitLow;
            }
        }
        else if (averageleft < averageRight) {
            servoh = ++servoh;
            if (servoh > servohLimitHigh) {
                servoh = servohLimitHigh;
            }
```

```
        }
        else if (averageleft = averageRight) {
            // nothing
        }
        horizontal.write(servoh);
    }
    delay(dtime);//delay time between each set of combined movement by both
horizontal and vertical servo
}
```

RESULTS AND DISCUSSION

Design errors and improvements

We went through many versions of our prototype which were not that compact and aesthetically pleasing before coming up with the finalised prototype. Our initial design was to have a base printed via laser cutting with a hole in the middle so that the horizontal motor could fit in the middle and 3cmx5cm rectangular holes on the side so the walls could fit two on each side. These walls have rectangular holes in them to store the vertical motor. We also had another baseboard without any holes to store the horizontal motor. We laser cutted the trial design, but unfortunately made some mistakes with the sizing and how the parts/motors would fit into the base. The side pieces were unstable when they were attached to the base and the horizontal motor could not fit tightly into the center of the base.

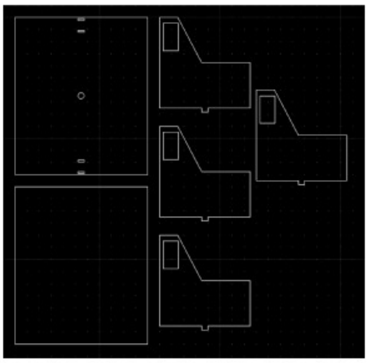

Figure 1: LibreCAD of first trial design

Figure 2: First trial design

From there, we fixed our measurement problems by decreasing the dimensions of the holes so that the pieces can attach tightly. We managed to create our second trial design (Figure 2.1) with the breadboards and Arduino located outside the tracker. The second trial design was able to function as a solar tracker. However, as the sensors were located in the breadboards outside the tracker, the direction was not very accurate. The wiring of the second trial design was also messy hence we decided to change our design again.

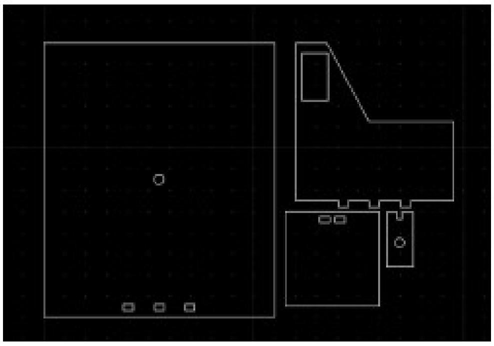

Figure 3: LibreCAD of second trial design

Figure 4: Second trial design

In our third trial design, we placed the breadboards on the horizontal board, and placed the LDRs on the panel separated by four partition walls. Although it also worked as a tracker, the vertical movement of the panel was restricted by the dangling wires.

Figure 5: Third trial design

We then tried a new design and did so by re-soldering wires, using shorter wires and printing a new base that allows walls to close off the circuitry of the tracker. We also made the walls between LDRs higher to further prevent light from entering multiple sensors and hence preventing the system from working. Instead of using 2 breadboards, we decided to make the system more compact by using an Arduino shield, in which all the wires would be soldered to. Even though we encountered some circuitry problems, we eventually managed to create a working model. However, the same problem of restricting movement occurred.

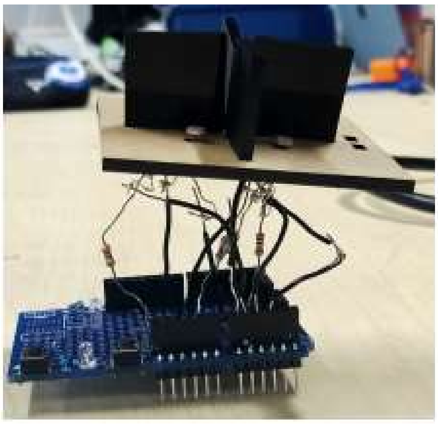

Figure 6: Panel of fourth trial design

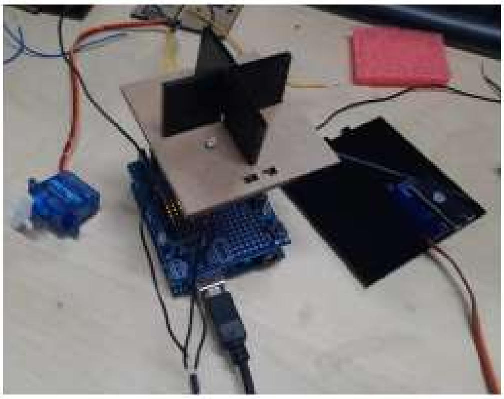

Figure 7: Fourth trial design

After identifying that the dangling wires from the solar panel restricts the movement of the panel, we decided to use a circuit board in our next design, as the circuit board allows the components of the circuit to be secured flatly on the circuit board. It also reduces the amount and length of wires that we have to use, which would solve our problem of messy wires and restricted movement. We planned to connect the circuit board directly below the panel so it rotates along with it, but when we tested the circuit board to see if the LDRs work, no results were coming out from the LDRs, meaning that something was wrong with the circuit board.

We suspected that it was because when soldering the wires to the circuit board, some of the wires which are not meant to touch were connected through the solder, or that we might have melted the circuit board while soldering. Although we knew how to make the circuit board function, we realized it would take a lot of time and trial and error due to the fact that our soldering skills are basic. We weighed the pros and cons of going with a circuit board and decided that it was not worth it, especially considering the duration of this project.

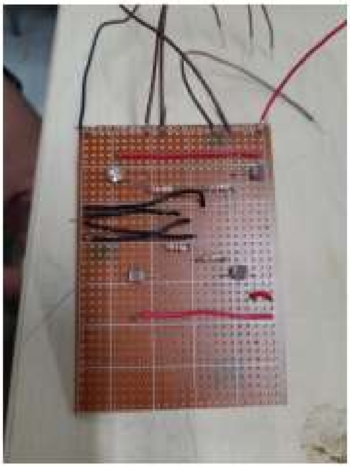

Figure 7: Circuit board for fifth trial design

Final Prototype

Finally, after lots of thinking, reviewing through our past designs, and with the advice from our mentor, we came up with our final design for the solar tracker. We decided to use the Arduino board as the solar panel, with a small breadboard placed on top of it which fit perfectly into the Arduino shield. Then we cut down the resistors and the M-M jumper wires so that the circuit is more compact and the electrical components are flat on the breadboard so that there are no dangling wires. Hence this would solve the previous problems we faced of the dangling wires that restrict the panel movement, and this design is also more compact. However, we did not cut the LDRs down as they needed to be tall to read the light intensity. We were going to have four walls surrounding the Arduino to support the wall holding the vertical motor and Arduino but we realized that it was unnecessary as the wall was stable enough on its own. We had the LDRs separated from each other as shown in Fig 6.1 so that when a light source is shone on them, they all do not read higher light intensity and only one is affected by the increase in light intensity so that the difference in vertical and horizontal LDRs is greater than the minimum tolerance value and hence instruct the servos to move accordingly.

Figure 8: Arduino as a solar panel

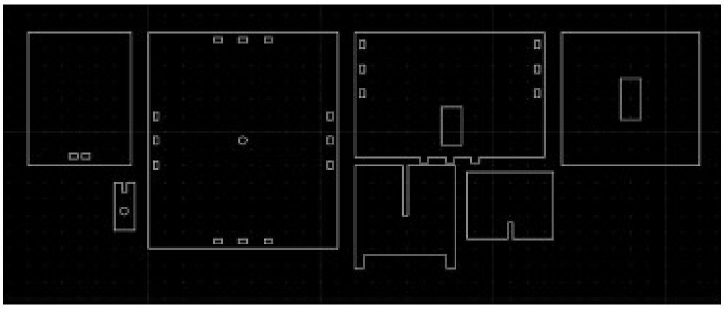

Figure 9: LibreCAD of final prototype

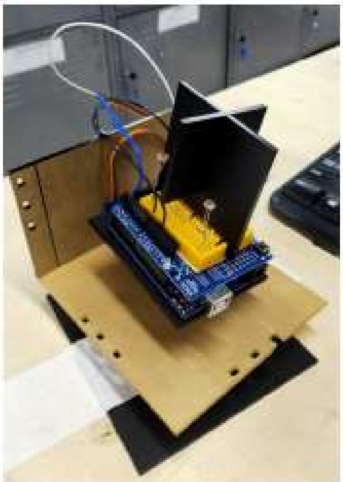

Figure 10: Final prototype

Electrical Circuit for final prototype

By having a breadboard attached to an Arduino shield, it allowed the wires used to be flat and neat. There are 4 yellow wires which serve to connect the LDRs to the analog pins so that the computer can read the surrounding light intensity via the LDRs. the red wire supplies 5V from the Arduino to the breadboard. The black wires provide the ground state for the breadboard so that the circuit is a closed circuit. There are also 4 of 220 ohms resistors to ensure the LDRs do not damage due to excessive current flowing through them and then this would cause our system to fail. There are other wires at the end of the Arduino shield connecting the servo motors to the Arduino.

Figure 10: Circuitry

CONCLUSION

Currently, our solar tracker works and is able to track a light source, but it has room for more improvement. If we could continue working on our solar tracker again, we would have designed it such that it is more compact, and that the circuitry is concealed to look more aesthetically pleasing. Adding on, when using LibreCAD to make the design for the joints of the solar tracker, we would have measured the length needed using the vernier calliper, then subtracted it by around 0.1mm, so the joints can attach tightly. We also would improve on our soldering skills such that we can solder circuits more neatly and properly. We realized that using breadboards would be easier and more time saving compared to circuit boards, especially if our soldering skills are not the best. Although circuit boards will allow things to look neater and more compact, it's not much of a difference from breadboards and there's much more room for error with soldering. Hence, we think that if it isn't necessary, and that if your soldering skills aren't great, you should choose breadboards over circuit boards to save time and materials. Overall, this R&DEP experience was quite thrilling and exciting, and we managed to pick up new skills like soldering and arranging circuits which we do not have the chance to do in our school. The mentorship experience was another thing we took away from this programme as it helped us grow in our communication, teamwork, and coordination skills. This research programme was very fun and we hope to participate in more research programmes like this offered by CRADLE.

REFERENCES

[1] BrownDogGadgets. n.d. “Simple Dual Axis Solar Tracker.”