

Solar Tracker Turret

Marisa Marzolini & Nicolino Primavera

Dr. Robert Duffin

Fordham University

PHYS 3013 R01 “Experimental Techniques in Engineering and
Physics”

11 May 2022

Table of Contents

I.	Abstract	p3
II.	Introduction	p3
	a. Figure I	p3
III.	Mechanical Design	p5
	a. Figure II	p6
	b. Figure III	p6
	c. Figure IV	p7
	d. Figure V	p7
	e. Figure VI	p7
	f. Figure VII	p8
	g. Figure VIII	p8
IV.	Arduino Programming	p8
V.	Optical and Electrical Systems	p9
	a. Figure IX	p10
VI.	Test and Assembly	p11
	a. Figure X	p11
	b. Figure XI	p11
VII.	Conclusion	p13
VIII.	Appendix	p14
IX.	References	p18

Abstract

Sun-seeking instruments on space crafts are used to tell the position of the sun. The primary purpose of this project was to design a solar tracker that would detect light from a given light source in a dark room. Only given solar panels, an Arduino Uno, servo motors, and jumper wires, our task was to design a 3D model using Solidworks to serve as the foundation for our device. After constructing the model, we were tasked with coding a program that would instruct the solar tracker to find a given light source. The device was designed to rotate automatically, detecting the light source with the help of two servo motors. By creating, designing, and testing the solar tracker, we were able to understand control theory while striving for a degree of accuracy, stability, and optimality.

Introduction

Without the sun, life ceases to exist. The sun's gravitational force keeps planets in orbit all while radiating heat and light to them. In order to survive and grow, plants depend on the sun. Animals, including humans, depend on plants for survival; the sun is essential in all life forms. The sun produces a certain amount of power that remains constant, known as radiation intensity. "The average radiation intensity that hits the edge of the Earth's atmosphere is

known as the solar constant, which is about 1,361 watts per square meter. Although this value is called a constant, it varies by about 7% between January 4th (perihelion), when the Earth is closest to the sun, and July 4th (aphelion), when the Earth is furthest away." (Sarjana, *Solar Energy to the Earth*). Solar power is the energy from the sun that is harnessed into electrical or thermal energy through various technologies. Specifically, energy from the sun is converted into electrical energy through photovoltaic (PV)

panels, typically known as solar panels. Once transformed into electrical energy, the energy can then be used to generate electricity.



Figure I: Image of solar panels absorbing light.

Photovoltaic panels offer many advantages to the environment, given that they are low in maintenance, easy to install, and offer renewable energy from any location. A single PV device is also known as a cell, the specific material used in today's project. These cells can produce a range of 1-2 Watts of power. The photovoltaic cells are in the form of photoelectric cells, given that they vary in current, voltage, and resistance when exposed to a source of light. The source of light does not always have to be from the sun; some photovoltaic cells, like the ones

used in this project, can also convert artificial light into electricity. In order to convert the energy into electricity, the cell must first be able to track and detect the source of light.

The solar cells are only able to successfully harness energy through significant exposure to a light source. In a situation where solar panels are exposed to direct sunlight, "the angle at which the sun's rays meet the surface of the solar panel (known as the "angle of incidence") determines how well the panel can convert the incoming light into electricity. The narrower the angle of incidence, the more energy a photovoltaic panel can produce." (Marsh, Jacob). The purpose of the solar trackers is to orient panels in an attempt to minimize the angle of incidence. Once the angle is minimized, the light will perpendicularly strike the surface. Eventually, the electricity can move throughout the device once the light is

absorbed through the photovoltaic cell, creating an electric flow.

While this project demonstrated the process of light photons activating electrons in a photovoltaic cell, the main focus was for the solar tracker to successfully detect the light source in a dark room. By learning the Solidworks methodology, crafting a unique code for the Arduino Uno, and assembling the design through 3D printing, the device will automatically rotate in a dark room to catch the light source.

Mechanical Design

We began the experiment by learning how to use Solidworks through the tutorial. Through knowledge obtained from the tutorial, we were able to start designing the unique parts needed for the project. In order to determine which parts were needed, we researched different solar tracker projects. With the preliminary researched information, we determined that we needed an Arduino

UNO, two servo motors, solar cells, jumper wires and a breadboard.

When reflecting on how to make the design possible, we thought of regular household objects that rotate, to draw inspiration. Our first thought was in regard to an ordinary object in our dining room table; a lazy susan. This object spins 360 degrees to rotate food dishes to serve a table. Seeing the functionality of a design as simple as a lazy susan gave us the idea to include a circular, rotating piece in our project that would rotate the solar panel to maximize the potential of locating the light source. Our only complication was figuring out how to incorporate the motor to create a design that simulated a lazy susan. A lazy susan uses a turntable mounted between the foundation and the circular rotating piece in order to get the object to spin, which was not something we were looking for.

With further research and collaboration, the mechanical design for the

project was produced. The design for the final model consisted of a foundation, 2 servo motors, a circular base, two columns, a solar cell, an Arduino UNO, a breadboard, and a circular rod. The solar tracker was designed to operate in three dimensions while the foundation remained stationary.

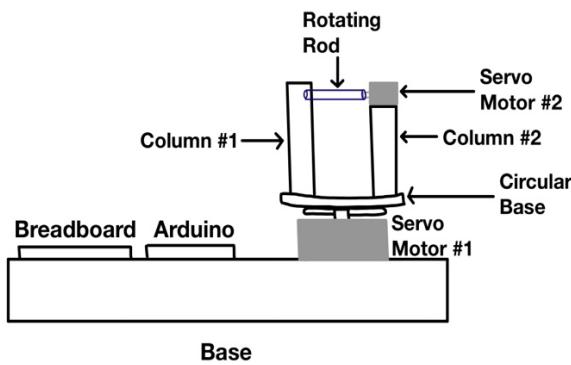


Figure II: Image of the final prototype that consisted of each individual part.

The first step of the design was creating a foundation in SolidWorks. The model included placing the Arduino UNO and breadboard on the foundation while inserting one servo motor into the base. We concluded that the easiest way to do this was by making an extruded cut in the foundation that the motor could fit into. Using Solidworks, we were able to precisely cut the

dimensions of the servo motor into the foundation, so the motor fit perfectly by measuring pieces and determining the final specifications. The base is pictured below.

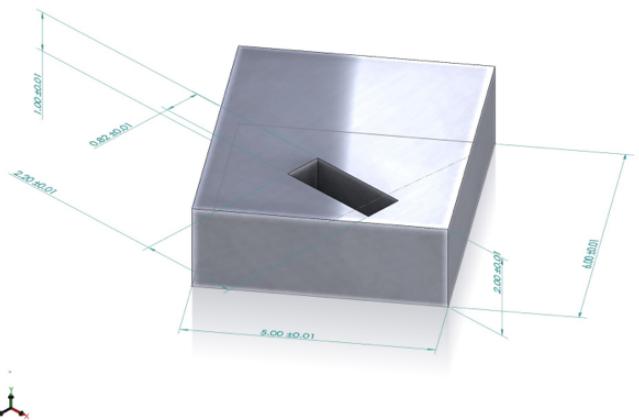


Figure III: Image of the solar tracker's foundation. Its dimensions are $b \cdot w \cdot h = 6 \text{ in} \cdot 5 \text{ in} \cdot 2 \text{ in}$, and the dimensions of the extruded cut (where one servo motor goes) are $b \cdot w \cdot h = 2.20 \text{ in} \cdot 0.82 \text{ in} \cdot 1 \text{ in}$. The dimensions of the servo motor are $b \cdot w \cdot h = 2.125 \text{ in} \cdot 0.791 \text{ in} \cdot 1.5 \text{ in}$.

Next, we began to design the pieces of the model that would be placed on top of the foundation. On top of the servo motor that rested in the extruded cut of the foundation was a circular disk. The disk rotated, with the help of the servo motor, while acting as a foundation for two columns.



Figure IV: This is the circular base with dimensions $r = 2$ in, $w = 0.5$ in.

The two columns were designed to rest on top of the circular base, and later provide support for the rod and solar panel. The second servo motor was mounted on top of column two.

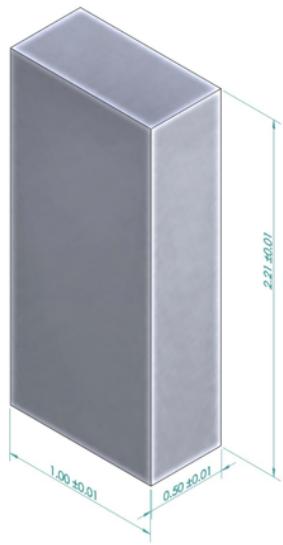


Figure V: Column two holds the second servo motor on top of it. Its dimensions are $b \bullet w \bullet h = 1.0$ in $\bullet 0.5$ in $\bullet 2.205$ in.

This servo motor accounted for the rotation of the rod which held the solar cell. The other column had a circular extruded cut within it to act as a rest for the rod and to serve as a safe place for the rod to spin without friction.



Figure VI: Column one provides stability and support to the rotating rod holding the solar cell. Its dimensions are $b \bullet w \bullet h = 1.0$ in $\bullet 0.5$ in $\bullet 4.0$ in.

The rod was then attached to the servo motor on top of column two, which rotated within the circular extruded cut in the first column. The solar panel was hot glued to the rod with the intention of the rod rotating the solar panel to detect light. With the columns resting on top of the circular base, the model was designed so that the circular foundation would spin while the rod rotated the solar

panel to detect any light in the room. The rod was designed to have part of its side flat so that it was easier to assemble (glue) the solar cell onto it. This small feature played a vital role in ensuring the solar cell's stability as well as the projects.

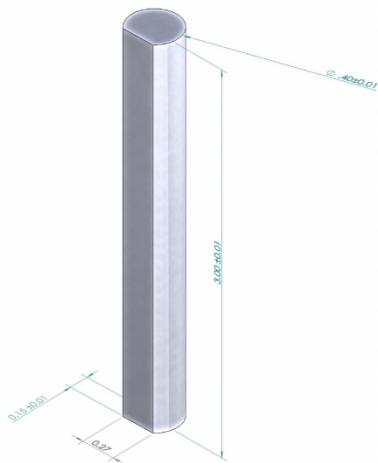


Figure VII: The rod the solar cell is mounted to that will rotate because it is attached to the second servo motor on top of column 2. Its dimensions are $l = 3$ in, $r = 0.20$ in. Its flat extruded cut dimensions are $l = 3$ in, $w = 0.27$ in. After modeling all the design's parts and successfully printing them out using the MakerBot printers in the lab, a preimlinary final copy was put together with all the parts balanced on top of each other.



Figure VIII: The final preliminary design after everything was printed.

Arduino Programming

Designing the prototype for the project consisted of researching and crafting code that would tell the servo motors to carry out specific functions. The particular codes associated with each servo motor were almost identical to each other, with both codes telling the motors to scan and pick the angle that detected maximum light. We began the code by telling the servo motor that rests in the model's foundation to rotate 90

degrees in increments of 10 degrees. This caused the columns and rod above to rotate in a circular manner, which aided in finding and absorbing the maximum light. Additionally, another similar code was needed that would tell the servo motor supporting the rod to rotate 90 degrees. With the solar cell resting on top of the rod, rotating the rod caused the cell to detect light from different angles starting at 0 degrees and ending at 90 degrees. Code 1 was the first code applied, and it was tested and successfully caused both servo motors to rotate 90 degrees at different times.

(See appendix A).

To improve the code, a “function” and “for loop” were created for each servo motor and an equal sign (=) was added to instruct each servo motor proper directions. (See appendix B). These codes worked similarly to a checklist. First, the servo motor was tasked with rotating and comparing the light intake with the previous location. As the motor was rotating, if the light intake

went to a maximum level, the motor would reverse and go back to the previous direction. This method was used on both servo motors at the same time to gain maximum light intake. The “for loop” and “function” codes allowed this to happen, but only while they were receiving information from the solar cell. The solar cell provided information to the Arduino through the analog function (A0). Once the Arduino received the light intensity input, the servo motors knew in which direction to move.

Optical and Electrical Systems

In order for the code to execute properly, correctly wiring the two servo motors to the Arduino was a crucial step in the optical and electrical systems phase. Attached below is the figure depicting the two servo motors attached to the Arduino with the assistance of a breadboard.

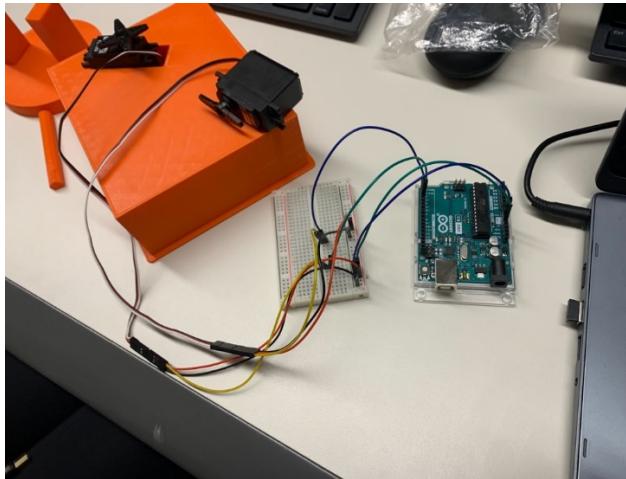


Figure IX: Shows both servo motors attached to the breadboard with the help of jumper wires. Then with another set of jumper wires, the two servomotors were connected to the Arduino using the 5V, GND, pin input 2, and pin input 3.

Given that there are insufficient inputs on the Arduino, it was essential to attach jumper wires to the wires of the servo motors, which would allow the placement of the wires on the breadboard. As noted in the figure above, the 5V red wires from the two servo motors were placed on the positive column of the breadboard. The jumper wires connected this column to the 5V input on the Arduino. The same process was completed with the black ground wires from the two servo motors; the only difference was that these wires were inserted in the negative column instead of the positive column. Finally, the jumper wires

connected the ground wires to the Arduino. The final step was correctly assigning each servo motor to the correct pin number, which would ensure the correct code would be carried out. The code was uploaded after the servo motors were correctly wired and assembled to the breadboard and the Arduino. The code ran smoothly just as we had hoped, with each servo motor moving milliseconds after the other, scanning for maximum light intensity.

The next step for wiring and assembly was soldering the solar panel wires and connecting them to the Arduino. This was an essential step in the wiring process, as the solar panels provide all of the information that causes the system to function. The servo motors act solely based on the input received from the solar cells. One challenge faced with wiring the solar panels was that the wires connected to the solar panel fell off. We trimmed new wires and soldered them to the solar panel to correct this issue. The process

of cutting and stripping the wires, followed by using the soldering machine, was a new and valuable experience. With collaboration and research, we were able to properly solder new wires onto the solar panel to send input to the Arduino. The final step was to connect the solar cell to the Arduino using the analog input A0 and the grounded state.

Test and Assembly

After soldering the wires, the final step was to connect the wires and cables to their appropriate spot on the breadboard and Arduino. Following that, the assembly of the project was the next step. After debating various methods, we concluded that using a strong adhesive, such as super glue, would be the best method to assemble the project and put the pieces together. Given that the pieces of the model were 3D printed using plastic, each individual piece was light enough to be glued to other pieces. Attached below are two

figures, depicting the final project glued together from different angles.

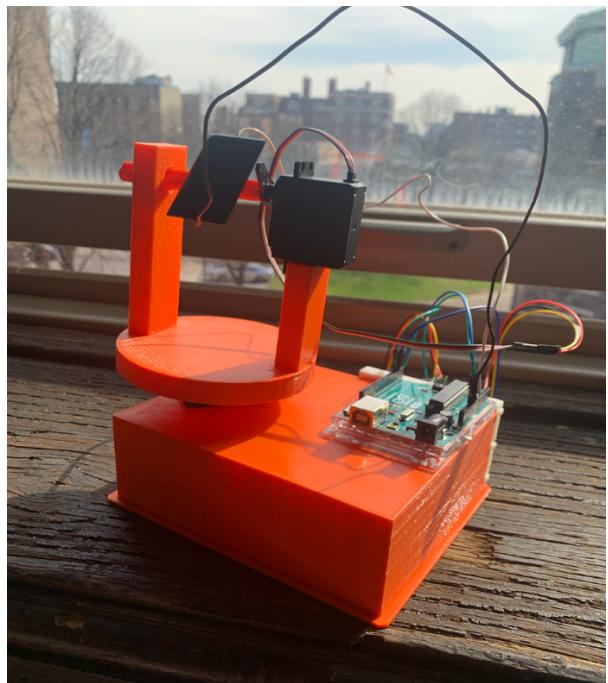


Figure X: Final model from the front.

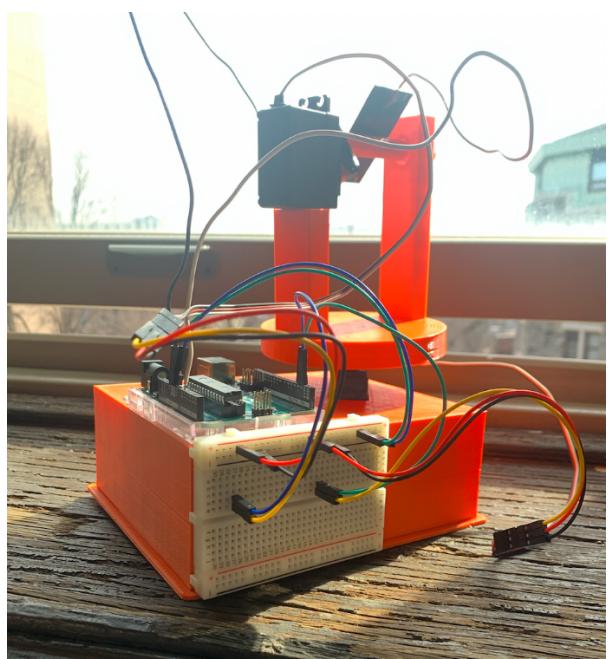


Figure XI: Final model from the rear.

Upon completion of assembly, it was time to test the solar tracker. Once testing began, we quickly realized that the previous code was insufficient to find the maximum light intensity. In the previous code, the input taken from the solar panel was not transmitting to the Arduino. Both servo motors were moving simultaneously without considering information from the solar panel. The goal was to have one servo motor rotate and complete its 90-degree rotation, followed by the second servo motor completing its 90-degree rotation. Instead of stopping after completing the full rotations, the servo motors ran through the code again. They would not go back to the maximum angle, which was a major issue. Our next step was to see how we could delay the code after its initial scan and send the two servo motors to the maximum angle.

Upon realizing that the code needed to be updated, we had to do further research on Arduino coding and C++ programming

language. Given that we both had a very minor understanding of programming, it took trial and error to come up with a code that would make both servo motors rotate 90 degrees at separate instances while also receiving information from the solar panel.

In creating the final code, it was decided to have each servo motor rotate in increments of 10-degrees, until each motor had rotated a total of 90 degrees. At each increment of 10, the code would note the quantity of the light detection. After the first servo motor completed its 90-degree rotation, the second servo motor would go on to do the same process. This servo motor also oscillated in 10-degree increments until it rotated a full 90 degrees. At the completion of the two rotations, the code compiled at what angle the maximum light detection occurred for both servo motors. At this point, both servo motors returned to the position where maximum light detection was observed. (See appendix C).

Conclusion

Upon completion, our project was successful. Both servo motors were able to oscillate 90 degrees to scan the classroom for light intensity. Once maximum light intensity was detected, the solar cell moved back to the position of maximum light.

Despite the success of the model, there were various challenges faced throughout the semester while working through the project. One challenge was having minimal knowledge of Solidworks, 3D printing, and coding. Through the completion of this project, we were able to learn how to code and use Solidworks to successfully 3D print pieces. Another difficulty was creating a final design that would maximize light intensity while also serving as the most efficient model. Throughout the course of the semester, we crafted many designs that we eventually had to rule out due to inefficiencies.

Even though the solar tracker turret succeeded in detecting the maximum light intensity, there are still various improvements that could be made to our project. With our design, we chose to only use one solar cell. One improvement could include adding additional solar cells, so that there is a greater chance of light absorption. If we had access to greater technology and stronger materials, other improvements could have been made in the functionality of the design. Given that the servo motors could only rotate 90 degrees, there were certain limitations within the oscillations of the motors. Motors that could rotate 180 degrees would provide a stronger opportunity for the solar panels to detect light from various angles of the room. If the light source was underneath the solar cell, a motor that rotated more than 90 degrees may be able to pick up that light source. Regardless of any improvements, the project satisfied all requirements and was successful.

Appendix

Appendix A: Code 1

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

int servoPin = 3; // Declare the Servo pin
int servoPin2 = 2;
Servo Servo1; // Create a servo object
Servo Servo2;

void setup() {
    Servo1.attach(servoPin); // We need to attach the servo to the used pin number
    Servo2.attach(servoPin2);

}

void loop(){
    Servo1.write(0); // Make servo go to 0 degrees
    delay(1000);
    Servo2.write(0);
    delay(1000);

    Servo1.write(90); // Make servo go to 90 degrees
    delay(1000);
    Servo2.write(90);
    delay(1000);

    Servo1.write(180); // Make servo go to 180 degrees
    delay(1000);
    Servo2.write(180);
    delay(1000);

}
```

Appendix B: Code 2

```
sigh=0
For i = 0,10,20,30,40,50,60,70,80,90
}
    Servo.write(i);
    Sig = readanalog(Aθ)
    If (sig > sigh) {
```

```

    Sigh = sig;
    Index = 1; }
}

```

Appendix C: Code 3

```

#include <Servo.h>           // Include the Servo library
#include <Math.h>             // Include the math library (fractions ect.)

Servo Servo1;                 // Create a servo object for servo1 & servo2
Servo Servo2;

int servoPin = 3;              // Declare the servo pins
int servoPin2 = 2;
int index;                     // Angle associated with max value -- servo 1
int index2;                    // Angle associated with max value -- servo 2

double sig;                   // sig = read from solar panel (volts) -- servo1
double sigh = 0.0;             // Previous max values -- servo1
double sig2;                  // sig2 = read from solar panel (volts) -- servo2
double sigh2 = 0.0;            // Previous max values -- servo2

void setup() {
    Servo1.attach(servoPin);      // We need to attach the servo to the used pin number
    Servo2.attach(servoPin2);

    Serial.begin(9600);          // Initialize serial communication at 9600 bits per second
}

void loop(){
    for (int i = 0; i <= 90; i = i + 10){ // Initialization; condition; increment
        Servo1.write(i);           // Telling servo1 to move in increments of i
        sig = analogRead(A0);       // Read the value from the solar panel
        Serial.println(sig);        // Prints the values of sig
        Serial.println(i);          // Prints the values at each increment of i (angle)
        Serial.println(sigh);        // Prints the value of sigh --> max value
        Serial.println();           // space
        delay(2000);               // Stop the program for 2000 milliseconds ~ 2
        seconds
    }

    if (sig > sigh) {           // updating maximum angle based on voltage received from solar
        panel
        sigh = sig;
    }
}

```

```

index = i;

}

Serial.println(index);           // Prints the values of index --> angle associated with max value
Serial.println();                // space
}

for (int j = 0; j<=90; j = j + 10) {

    Servo2.write(j);            // Telling servo2 to move in increments of j
    sig2 = analogRead(A0);      // Read the value from
    Serial.println(sig2);       // Prints the values of sig
    Serial.println(j);          // Prints the values at each increment of i (angle)
    Serial.println(sigh2);      // Prints the value of sigh --> max value
    Serial.println();           // space
    delay(2000);               // Stop the program for 2000 milliseconds ~ 2 seconds

    if(sig2 > sigh2) {         // updating maximum angle based on voltage received from solar
        panel

        sigh2 = sig2;
        index2 = j;

    }

    Serial.println(index2);     // Prints the values of index --> angle associated with max value
    Serial.println();           // space
}

Serial.println("The max angle on the x-axis is");   // Explaining what the next line is
Serial.println(index);                            // Print max value from servo1
Serial.println();                                // space

```

```
Serial.println("The max angle on the y-axis is");      // Explaining the next line
Serial.print(index2);                                // Print max value from servo2
Serial.println();                                    // space
Serial.println();                                    // space

Servo1.write(index);                                // Telling servo1 to move to index(max) angle
Servo2.write(index2);                               // Telling servo2 to move to index2(max) angle

delay(30000);                                     // delay for 30 seconds before the loop starts again

}
```

References

A Mohamad “Analysis of an Arduino Based solar Tracking System” et al 2021 J. Phys.: Conf.

Ser. 2051 012011. <https://iopscience.iop.org/article/10.1088/1742-6596/2051/1/012011/pdf>. Accessed Feb. 9, 2022.

Solar Energy Technologies Office. (n.d.). *How does solar work?* Energy.gov.

<https://www.energy.gov/eere/solar/how-does-solar-work>. Accessed Feb. 9, 2022.

Marsh, J., Marsh, J., About Jacob MarshJacob is a researcher and content writer at EnergySage,

View all posts by Jacob Marsh →, 2, J. F., 9, C. B. A., 18, J. N., & 1, A. D. F. (2022, January 12). *Solar trackers: What you need to know:* EnergySage. Solar News.

<https://news.energysage.com/solar-trackers-everything-need-know/>. Accessed Feb. 9, 2022.

BlueClaw7. “Solar Tracker Using Arduino and Raspberry Pi 3.” *Instructables.com*, 2018,

<https://www.instructables.com/Solar-Tracker-Using-Arduino-and-Raspberry-Pi-3/>. Accessed Jan. 26, 2022.

123samridhgarg. “Arduino Solar Tracker Servo-Controlled, Light Tracking.” *Create.arduino.cc*,

Dec. 11, 2019, <https://create.arduino.cc/projecthub/123samridhgarg/arduino-solar-tracker-servo-controlled-light-tracking-6a7bb9>. Accessed Jan. 26, 2022.

Sarjana, A. (2021, September 27). *Solar energy to the Earth*. Solar energy to the Earth – Energy

Education. https://energyeducation.ca/encyclopedia/Solar_energy_to_the_Earth.

Accessed Mar. 1, 2022.