MECE E3028 Mechanical Engineering Laboratory II
Professor Qiao Lin
Spring 2022

Experiment E5: Arduino Solar Tracking

LABORATORY REPORT

Lab Group 11

Leon Aharonian

Christian Eberhard

Henry Jensen

Emily Milian,

Joaquin Palacios

Ethan Shek

Samuel Wustefeld

Columbia University
Department of Mechanical Engineering

February 16th, 2022

Contents

List of Figures 1

I Introduction 2

II Theory 2

III Apparatus and Approach 3
- III-A Apparatuses Used . . . 3
  - III-A.1 Photo Resistors . . . 3
  - III-A.2 Stepper Motor . . . 3
  - III-A.3 DC Motor . . . 3
  - III-A.4 Solar Panel . . . 3
  - III-A.5 LED . . . 3
  - III-A.6 Potentiometer . . . 3
  - III-A.7 Buttons . . . 3
  - III-A.8 Photo Resistor Mount . . . 3
  - III-A.9 Arduino . . . 4
- III-B Approach . . . 4

IV Results 4

V Discussion 5
- V-A Mechanical Design & Electronics . . . 5

VI Conclusion 6

References 6

VII Appendix 7
- VII-A Arduino Code . . . 7

List of Figures

1 Diagram of how changing the angle tilt for a solar panel changes the angle of incidence to get more energy [1]. 2

2 Exploded view of CAD model with labeled components. . . . 4

3 Render of solar tracker CAD model pointing at the sun. . . . 4

4 Front view of solar tracker. Shows the DC motor, wired circuit broads, Buttons, Photo resisters, solar panel, LED, and potentiometer, as well as the laser cut and machined components what comprise the structure. . . . 4

5 Back view of solar tracker. Shows the Arduino, Motor Shield, Photo resisters, solar panel, LED, and potentiometer. 5

6 Circuit schematic: (1) Power Source, (2) 5V and Ground, (3) Photo-resistors, (4) Buttons, (5) Potentiometer, (6) Analog Connections, (7) Digital Connections . . . 5

***Abstract*— The goal of this experiment was to design and build a solar tracking system that follows and directs a solar cell array towards an external light source, optimizing orientation to maximize energy output. To achieve this, an Arduino-based system was designed which utilized a DC servo, stepper motor, potentiometer, and two photo-resistors in addition to other electronic components to track and position the solar array toward the light source. Our design successfully tracked a moving light source by determining the position of the highest light intensity in real-time by comparing the signal inputs from the photo-resistors, and then by using the stepper motor to position the array appropriately. This provided a great way to gain familiarity with the Arduino system functionality and with a variety of working components that are commonly used in prototyping.**

I. INTRODUCTION

In this experiment, an Arduino-based solar tracking system was designed, built, and programmed using a variety of common prototyping components. The main goal was to increase familiarity with the Arduino architecture, in addition to gaining experience with using such a system to control several different sensors and motors. Arduino is an open-source platform that makes use of simple hardware and software, and this makes it ideal for use in small-scale prototyping and hobby uses. An Arduino board is equipped with a microcontroller, which can receive instructions, and is able to read various forms of digital and analog input. Because the Arduino is capable of accessing and controlling multiple components simultaneously, this allows for the integration of several elements in a single functional system. In this case, the major components include photo-resistors, a potentiometer, and two different types of motors - a DC motor and a stepper motor. Commonly used for applications that require high positional accuracy, stepper motors can be positioned with very high precision, which makes them the optimal choice to use as the main driver for positioning the solar array. While DC motors may not have such fine positional capabilities when compared to stepper motors, they are much easier to run. Needing only direct voltage input, in this application the DC motor is used to turn the entire assembly that is mounted upon the base plate.

The goal of solar tracking systems is to optimize the angle at which the sun's rays meet the surface of the solar panel known as the "angle of incidence", which affects how efficiently the panel can convert the incoming light into usable energy. Essentially, the smaller the angle of incidence, the more energy a photo-voltaic panel can produce. The use of solar trackers helps to minimize this angle by orienting panels such that light will strike the arrays perpendicular to their surfaces to maximize the energy output.

Using tracking systems for solar arrays can increase the efficiency of upwards of 20-40 percent [2]. While passive solar trackers are available, they are far less capable than active solar trackers in their movement and orientation functionalities. Active solar trackers which use motors to help orient the array may require initial costs to install and incur additional upkeep costs, but these are often outweighed by the return seen by increased energy yield. Though they are not often available to the commercial consumer, active solar trackers can boost the efficiency of the solar array to those higher percentages [3].

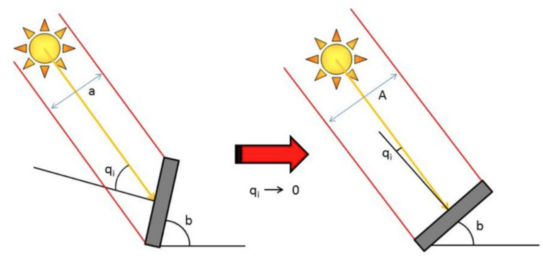

Fig. 1. Diagram of how changing the angle tilt for a solar panel changes the angle of incidence to get more energy [1].

Increasing the efficiency of the output energy of solar arrays has become increasingly relevant over the last decade as the need and desire for renewable sources of energy have grown immensely. The S.E.I.A. (Solar Energy Industries Association) has said recently that "In the last decade alone, solar has experienced an average annual growth rate of 42 percent. There are now more than 100 gigawatts of solar capacity installed nationwide, enough to power 18.9 million homes." [4] With this growing extreme demand across the nation it seems more imperative than ever to accelerate our comprehension of collecting solar energy as efficiently as possible. With this experiment, the idea is to provide a basic working knowledge of the basic components that make up the rudimentary solar tracker.

II. THEORY

The Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read inputs and turn them into an output. More specifically, the Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). [5] It has 14 digital input/output pins, 6 of which can be used as PWM outputs, 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header, and a reset button.

While they are both cheap and easily controlled, stepper motors give most of their benefit when being used as a precision machine. In other words, they provide control as an advantage over mechanical power generation and should be used if accuracy is a primary concern. However, a disadvantage to stepper motors is that they are not very efficient because they consume current at max load whenever the stator poles are energized, which reduces the energy efficiency and increases the losses due to heat. [6]

DC motors are best used in applications requiring constant torque across the motor's speed range. DC motors offer sustained output power and endurance. However, a disadvantage to DC motors is that they typically do not generate high torques at low speeds because they are more suited towards continuous uses, as their torque is constant over their speed range. In other words, they cannot push hard from rest like a stepper motor can, as it has a more sustained output.

The Arduino board is connected to a computer via USB, where it connects with the Arduino development environment (IDE). The user writes the Arduino code in the IDE, then uploads it to the microcontroller which executes the code, interacting with inputs and outputs.

The fundamental operation of a permanent magnet motor has an outer stator that holds windings of coils fed by a power source. With this, the rotor freely rotates based on the forces imparted by the stator coils.

The defining features of the permanent magnets (PM) within their rotor are acted upon by the rotating magnetic field (RMF) of the stator windings and are repelled into rotational motion. With this in mind, it can be said that its rotor's magnetic field is permanent and does not require a source of power to be generated. Additionally, permanent magnets require a variable frequency drive to operate, which acts as a control system that smooths out the torque produced. By turning the current on and off to the stator windings at certain stages of rotor rotation, the PM drive controls both torque and current. This data is then used to calculate rotor position, and therefore the speed of the shaft output. This calculation shows that their rotational speed matches the speed of the RMF, meaning they are synchronous machines. [7]

A stepper motor differs from a DC motor in that it does not have a spinning wire coil inside stationary magnets. Instead, it has a spinning gear-like magnet inside a stationary wire coil. The stepper motor receives electrical pulses and can use these to precisely rotate the gear-like magnet a fixed amount of angular rotation. This restrains the motor's movement to small steps but allows for high precision movements.

This project requires the use of a motor shield. A motor shield is an extension that can be added to an Arduino to allow it to power and control multiple motors at once. We will use the Adafruit Motor Shield V2 to connect and control a stepper motor and a DC motor at once. In order to use this motor shield, the Adafruit Motor Shield V2 Library is required. This allows the Arduino to be able to communicate to the motor shield.

To measure the light level, we will make use of a photocell. A photocell is a semiconductor that can decrease its resistance to light. When photons hit the semiconductor, electrons within the material are freed and thus decrease its electrical resistance. By measuring the resistance across the photocell, the amount of light incident on its surface can be measured. In order for the Arduino to be able to read this light level, a voltage divider must be created. This can be done by adding a single resistor after the photocell in a circuit. This divides the total output voltage before going to the Arduino.

We will also use a potentiometer. A potentiometer is similar to a photocell in that they are both variable resistors. However, while a photocell changes resistance based on light levels, a potentiometer changes resistance based on the angular position of a rod. This can be used in tandem with a DC motor to track the angle of rotation. Because a DC motor cannot accurately move, it cannot track its position. By connecting a potentiometer to the DC motor's shaft, the DC motor's angular displacement can be tracked.

III. Apparatus and Approach

A. Apparatuses Used

1) Photo Resistors: A photo-resistor is a two-terminal component whose electrical resistance increases proportionally to the amount of light the sensor is exposed to. By setting up a simple voltage divider circuit, depicted in figure 6 (component 3), an analog input pin of an Arduino (figure 6, component 6, blue and purple cables) can be used to record this resistance in real-time. The presented solar tracker used two photo resisters, which were mounted on opposite ends of a shadow casting divider. This setup was used to determine which direction the light was coming from at any given instant based on the magnitude of resistances measured.

2) Stepper Motor: This component is a high torque motor that rotates in discrete steps in either direction (figure 2). This apparatus was used to rotate the solar panel and photo-resistors.

3) DC Motor: A DC motor has only two leads and rotates continuously when powered. The direction of rotation depends on the polarity of the leads. This element was used in order to rotate the tracking mechanism with respect to the ground (figure 2).

4) Solar Panel: A solar panel is a device capable of harnessing the energy of light, primarily light emitted by the sun and converting it into electricity (figure 2). This apparatus was used to determine the success of the solar tracking mechanism.

5) LED: A light-emitting diode, LED, is a low-power semiconductor that glows when a potential difference is applied across its terminals. Since it is a diode it has an anode and a cathode which means it only allows current to flow in one direction. An LED was used to demonstrate the effectiveness of the solar panel.

6) Potentiometer: A potentiometer is a three-terminal resistive element with a rotating contact that adjusts the resistance across any two terminals. This element was used to track the position of the solar panel and ensure it did not rotate past a determined point in order to avoid cable entanglement (shown in figure 2). It was used as a voltage divider (shown in figure 6, component 5), in order to track angular position as a function of the voltage measured. Voltage was measured using a cable connected to an analog input in the Arduino (figure 6, component 6, orange cable).

7) Buttons: These elements are physical mechanisms that complete electrical circuits to enable current flow and selectively operate an electrical component (figure 6, component 4). Their on and off status was measured using digital connections to the Arduino (figure 6, component 7). They were used to operate the DC motor and rotate the base of the mechanism, each button controlling a direction of rotation (clockwise and counter-clockwise).

8) Photo Resistor Mount: This small, 3D printed component, depicted in figure 2, was designed to mount the photo resisters to the hub connecting the potentiometer to the solar panel and keep them parallel to the panel. The mount has a divider separating the two resisters. This was designed to cast a shadow on whichever sensor is further from the sun, amplifying the resistance difference between the sensor.

9) Arduino: An Arduino is a single-board micro-controller with numerous analog and digital pins that can be programmed to control simple electrical components such as the ones used in this design. This element was used to control all other electrical components and implement their functions, as described above (figure 6, element 1). The code used to operate the Arduino can be referenced in the Appendix (Section VII).

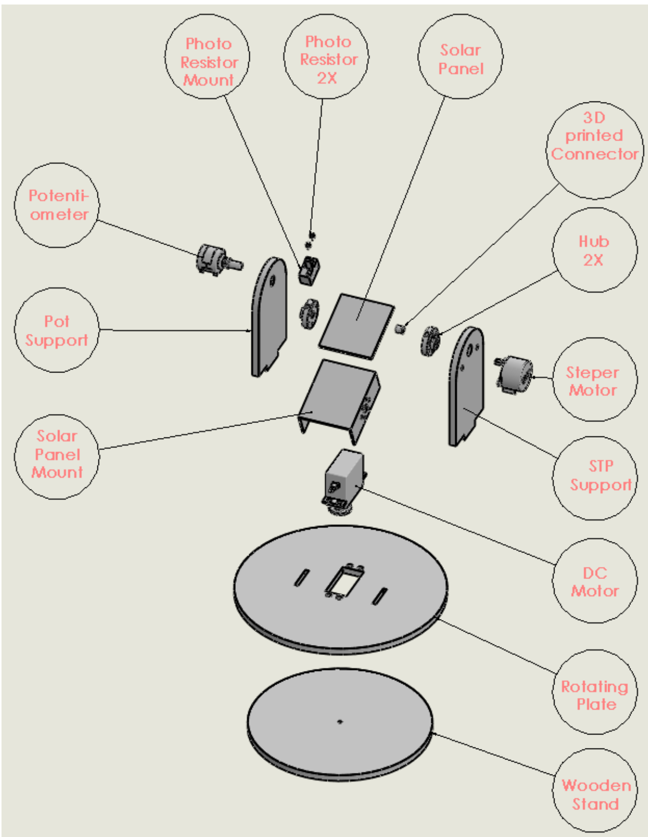

Fig. 2. Exploded view of CAD model with labeled components.

B. Approach

In order to house our mechanism, an acrylic base and two vertical supports were laser cut and hot glued to hold each component in place, as depicted in figure 3. The stepper motor and potentiometer were then attached to a solar panel mount which hosted the solar panel, LED light, photo-resistor mount, and photo-resistors. The DC motor was then screwed in place to the acrylic base, and the motor's shaft was screwed to a wood stand (figure 2). The electronic components were then connected following the schematic depicted in figure 6. For aesthetic purposes, all the wires were cut to size and soldered together creating neat, heat shrink-wrapped joints. Front and back images of the finished assembly can be seen in figures 4 and 5 respectively. For a full description of the design choices made, see section V.

Fig. 3. Render of solar tracker CAD model pointing at the sun.

IV. Results

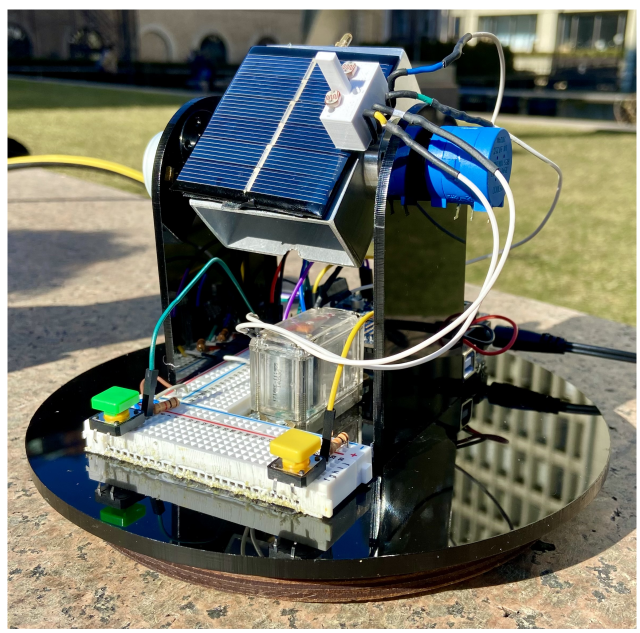

Fig. 4. Front view of solar tracker. Shows the DC motor, wired circuit broads, Buttons, Photo resisters, solar panel, LED, and potentiometer, as well as the laser cut and machined components what comprise the structure.

From exhaustive testing of the apparatus using a handheld flashlight, the mechanism was found to effectively track the position of the light source and rotate the solar panel accordingly, optimizing the energy output of the solar panel. Additionally, the mechanism motion was smooth and no electrical components showed signs of drawing voltage and current above their specifications (no overheating). The arrangement of electrical components and cables was also successfully implemented to avoid entanglement and interference.

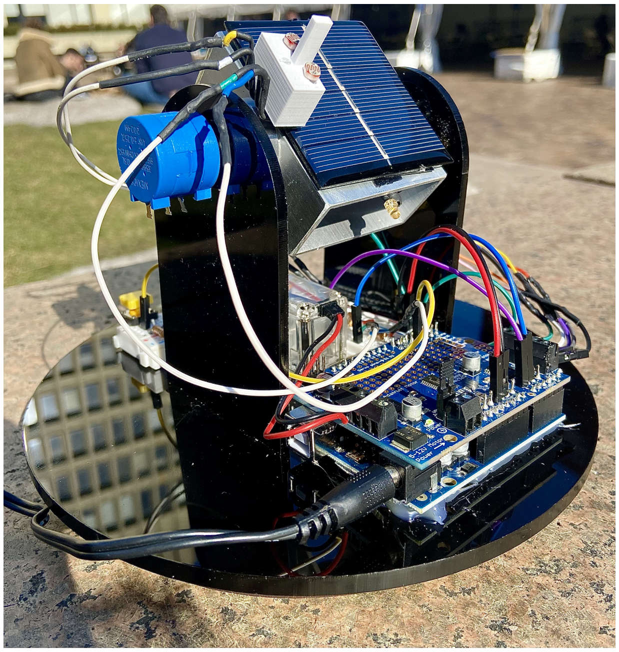

Fig. 5. Back view of solar tracker. Shows the Arduino, Motor Shield, Photo resisters, solar panel, LED, and potentiometer.

V. Discussion

A. Mechanical Design & Electronics

The main design decisions revolved around our intention of tracking the sun in real time and creating a mechanism whose performance would not falter with changing seasons.

To track the sun in real time we implemented a system with two photo resisters mounted to the stepper motor along with the solar panel. The decision to use two resistors was motivated by the intent of continuously tracking a light source, which was accomplished by measuring the voltage across the two photo-resistors, and if one of them had a higher value (due to lower light intensity) the mechanism would rotate in the opposite direction. This ensured that the mechanism continuously pointed in the direction of highest light intensity. It was found that the photocell sensitivity in detecting differences in light intensity was 20mV, so this value was used as a minimum threshold difference in order to determine when to rotate the mechanism. If the difference in voltage between the two resistors was less than 20mV, the mechanism was programmed to remain stationary at that optimized angle. As seen in figure 5 the sensors were mounted right next two each other with a small divider between them to cast a shadow on the one not facing the sun. This amplified the difference in light intensity whenever the mechanism was not pointed at the light source, allowing it to better correct itself.

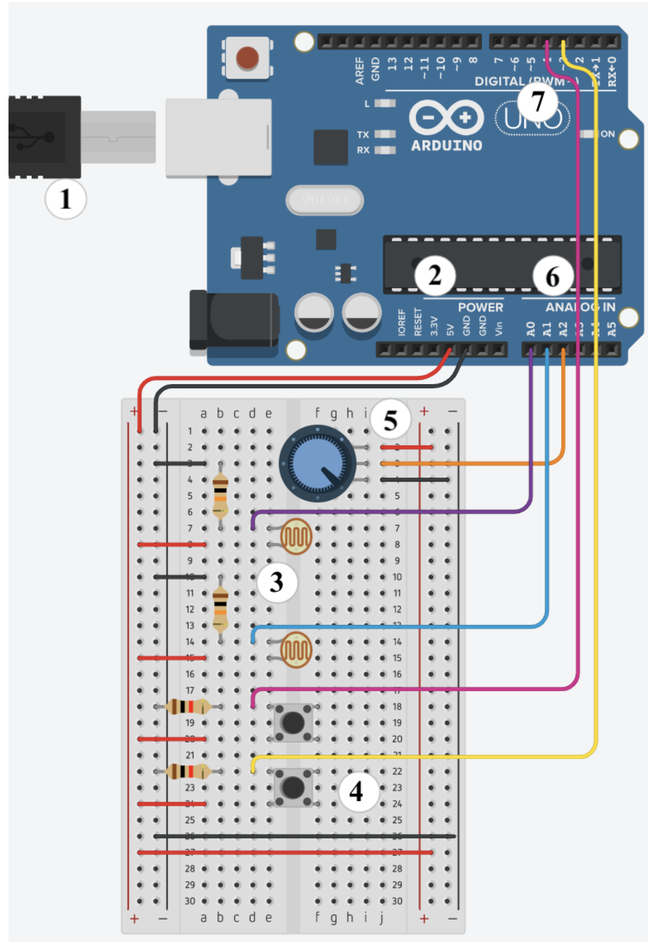

Fig. 6. Circuit schematic: (1) Power Source, (2) 5V and Ground, (3) Photo-resistors, (4) Buttons, (5) Potentiometer, (6) Analog Connections, (7) Digital Connections

The stepper motor was programmed to take one step toward whichever sensor recorded more light and then read both sensors again before making its next step in the appropriate direction. Because the solar panel was also mounted to the stepper motor, it automatically faced the sun as the photo resisters told the motor which way to turn. The potentiometer was mounted on the same axis as the solar panel and stepper motor, so that any rotation in the panel would rotate the potentiometer shaft and adjust the resistance. This was used to track the angular position and implement a fail safe to prevent the motor from turning further than the reach of the wires. This was accomplished by programming the Arduino to stop the stepper motor whenever any the potentiometer reached any of two values we set to be the limits.

The skeleton for the mechanism was made out of laser cut acrylic sheets and a machined piece of aluminum 2X1 box tubing. The laser cuter was chosen to create most of the components because it is extremely efficient and precise. Additionally, acrylic is lightweight, which was important because the entire setup had to rotate on the DC motor. Since the solar panel mount was an exceptionally key component, it was made out of thin aluminum as it is more durable. The box tubing was machined to fit the hubs and the solar panel with a CNC mill.

Additionally, we used the DC motor to add a second degree of freedom via the rotation of the entire mechanism with respect to the ground. This design component was added to expand the functionality of the tracking system.

The code used to operate the Arduino and implement all of the aforementioned mechanisms can be referenced in the Appendix (section VII).

VI. Conclusion

In this experiment, an Arduino, solar cell, DC servo, stepper motor, potentiometer, and photo-resistors were effectively utilized in an active solar tracking design. The system was able to determine the angle at which the intensity of light was greatest with respect to its solar array, then rotate the panel with a stepper motor such that the direction of greatest intensity was oriented perpendicularly to its surface. This functionality increased the overall efficiency of the photovoltaic cell and thus maximized its effective energy output. Furthermore, this experiment provided experience with programming and fabrication through the design of a system with very practical real-world use cases; all while increasing familiarity with Arduino and other common prototyping components like DC/stepper motors and electronic systems. The system performed efficiently, and effectively fulfilled all of the stated requirements.

References

[1] B. Rudisill, "The solar resource."
[2] M. McHale, "Maureen mchale," Oct 2015.
[3] J. Marsh, "Solar trackers: What you need to know: Energysage," Jan 2022.
[4] M. Davis, "Solar market insight report 2021 q4."
[5] Arduino, "What is arduino?." https://www.arduino.cc/en/Guide/Introduction. Accessed: 2022-02-16.
[6] T. Publishing, "Stepper motors vs. dc motors - what's the difference?." https://www.thomasnet.com/articles/machinery-tools-supplies/stepper-motors-vs-dc-motors/. Accessed: 2022-02-16.
[7] T. Publishing, "All about permanent magnet motors." https://www.thomasnet.com/articles/machinery-tools-supplies/permanent-magnet-motors/. Accessed: 2022-02-16.

VII. Appendix

A. Arduino Code

```
#include <Adafruit_MotorShield.h>

int photores_left;
int photores_right;
int diff_res;
const int threshold = 20;

const int speed_dc = 50;
const int buttonPinForward = 4;
int buttonStateForward = 0;
const int buttonPinBackward = 3;
int buttonStateBackward = 0;

Adafruit_MotorShield AFMS = Adafruit_MotorShield();

Adafruit_StepperMotor *myMotor = AFMS.getStepper(200, 2);
Adafruit_DCMotor *myDMotor = AFMS.getMotor(1);


void setup() {
  Serial.begin(9600);           // set up Serial library at 9600 bps
  while (!Serial);
  Serial.println("Solar Panel Engaged");

  if (!AFMS.begin()) {
    Serial.println("Could not find Motor Shield. Check wiring.");
    while (1);
  }
  Serial.println("Motor Shield found.");

  myMotor->setSpeed(10);  // 10 rpm

// Set the speed to start, from 0 (off) to 255 (max speed)
  myDMotor->setSpeed(150);
  myDMotor->run(FORWARD);
// turn on motor
  myDMotor->run(RELEASE);
}

void loop() {
  photores_left = analogRead(A0);

  photores_right = analogRead(A1);

  diff_res = photores_left - photores_right;

  Serial.println("test");
  Serial.println(diff_res);
  Serial.println(photores_left);
  Serial.println(photores_right);

  if (diff_res < - threshold) {
      myMotor->step(20, FORWARD, MICROSTEP);
  } else if (diff_res > threshold) {
      myMotor->step(20, BACKWARD, MICROSTEP);
  }

  uint8_t i;

  buttonStateForward = digitalRead(buttonPinForward);
  buttonStateBackward = digitalRead(buttonPinBackward);
```

```
  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonStateForward == HIGH) {
    myDMotor->run(FORWARD);

    Serial.println("HIGH");

    for (i=0; i<speed_dc; i++) {
      myDMotor->setSpeed(i);
      delay(10);
    }
    for (i=speed_dc; i!=0; i--) {
    myDMotor->setSpeed(i);
    delay(10);
    }

  } else {
    Serial.println("LOW");
  }

  if (buttonStateBackward == HIGH) {
    myDMotor->run(BACKWARD);

    Serial.println("HIGH");

    for (i=0; i<speed_dc; i++) {
      myDMotor->setSpeed(i);
      delay(10);
    }
    for (i=speed_dc; i!=0; i--) {
    myDMotor->setSpeed(i);
    delay(10);
    }

  } else {
    Serial.println("LOW");
  }
  myDMotor->run(RELEASE);
  delay(1000);
}
```