ELC152.2 Final Project: Light Following Solar Panel Using Light Dependent Resistors and Servo Motors

Post-laboratory Report; Submitted: 14 December 2019

Victor Lorenzo B. Tirona, Ray Mart M. Montesclaros
BS Electronics Engineering
Electronics, Computer, and Communications Engineering
Ateneo de Manila University
Philippines

***Abstract*— In collecting solar energy, using a two-axis sun tracking system has proven effective in significantly increasing the total energy collected. This project aims to create a miniature two-axis sun tracking system using light-dependent resistors, servo motors, and an Arduino unit as the main components. The light-dependent resistors, placed on four corners of a rectangular piece of cardboard determine the average resistances at the top, bottom, left, and right sides of the cardboard. The Arduino Uno compares these averages and determines which vertical and horizontal side have lower average resistances or is exposed to more light. The Arduino unit then outputs to the servo motors, attached to a wooden frame and the cardboard, to rotate towards the sides exposed to more light. Through creating the project, the proponents were able to successfully create a two-axis sun tracking system. While the proponents state that the movement of the system is slow as the system was tested with a flashlight, in actual application sun tracking, the movement need not be fast.**

***Index Terms*— solar energy, two-axis, sun tracking system, light-dependent resistors, servo motors, Arduino, Arduino Uno.**

I. Introduction

The sun is an essential part of life on earth. In fact, one of the most commonly used forms of generating renewable is through harnessing energy from the sun to collect solar energy. So common that there are large plots of land with millions of solar panels, dedicated to harnessing solar energy. One of the largest of these plants is Kamuthi solar facility in India with around 2.5 million solar panels covering 2,500 acres of land. With this coverage, Kamuthi solar facility has a total generation capacity of 648MW.

While a significant amount of energy is already obtained from the sun, various methods are being developed to harness more energy. Some of these methods include changing the shape of solar panels to be convex or flower-shaped, adding reflectors to redirect sun rays to the solar panel, and adding moving mechanisms to the solar panel to change its angle according to the position of the sun.

Solar tracking has been found to be one of the most common methods in improving solar energy collection. Research conducted by Eke and Senturk in 2012 compared fixed PV systems to PV systems with two-axis sun tracking based on their power output. Two-axis sun tracking systems were found to obtain 30.79% more electricity compared to the fixed PV systems with the former obtaining 15.98MWh and the latter obtaining 11.53MWh [2].

The objective of this project is to create a two-axis sun tracking PV system using light-dependent resistors and servo motors. The project will make use of a relatively small solar panel and will use an Arduino Uno as the main processing unit.

II. Theoretical Background

A. Light Dependent Resistor

The light following solar panel utilized the light dependent resistor (LDR) for sensing or detecting the presence or level of light. The LDR or photoresistor is an electronic component that changes resistance values based on the light that falls upon it [3]. See Fig. 1 for a typical light dependent resistor.

As mentioned, the resistance across the LDR varies based on how much light is pointed towards it. As the level of light increases the resistance across the LDR decreases. Its sensitivity to light is mainly due to the semiconductor materials present in the LDR. Wide varieties of semiconductor materials can be used for an LDR, the most commonly used are Cadmium Sulphide (CdS), Cadmium Selenide (CdSe), Lead Sulphide (PbS) and Indium Antimonide (InSb), to name a few. However, the use of cadmium based semiconductor materials are restricted in Europe due to environmental risks and repercussions [1]. It is also imperative to know that LDRs are passive devices since they do not possess a PN junction [1]. In schematic diagrams LDRs are represented in Fig. 2. The arrows located at the top left of the symbol represent the light that is falling on the component.

Fig. 1. Leaded Light Dependent Resistor [1]

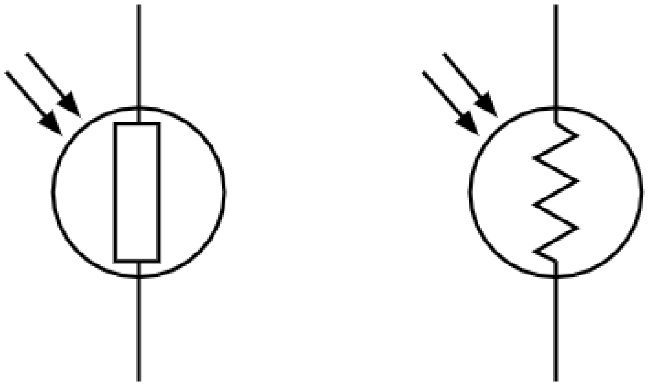

Fig. 2. Light Dependent Resistor Symbol [1]

Knowing that conductors are characterized by the large amount of free electrons due to potential difference and insulators are the opposite having only a few free electrons, one can get a simple understanding of how LDRs work [1]. When little to no light falls on the semiconductor, the resistance across the LDR is high due to the fact that the vast majority of electrons are latched onto the LDR's crystal lattice rendering them immobile. On the other hand, if "light falls on the semiconductor, the light photons are absorbed by the semiconductor lattice and some of their energy is transferred to the electrons" [1]. As a result, the electrons gain enough energy to break through the crystal lattice allowing them to conduct electricity. The more conductive a material is, the less resistance across the LDR. In summary, the more light that falls on the LDR the lesser the resistance. On the other hand, the less light that falls on the LDR, the higher the resistance [1].

B. Servo Motor

For the motor control of the light following solar panel, the proponents used an RC servo motor. "A servo motor is an electrical device which can push or rotate an object with great precision" [2]. This was chosen for the project's motor control since it can produce high torque for small and lightweight applications.

The "Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN" [2]. See Fig. 3.

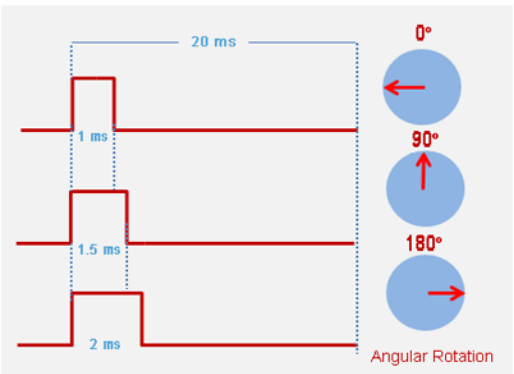

Fig. 3. Servo Motor PWM Principle[2]

A servo motor is basically composed of a DC motor, a variable resistor and a set of gears. Given that work is equal to the product of the force multiplied by distance. In servo motors, the force is higher but the distance covered is less, as opposed to DC motors where the force is low but with greater distance [2]. For this application, the servo motor was able to rotate from 0 to 180 degrees and was supplied by the Arduino 5-volt supply voltage. See Fig. 4.

As seen in the figure above, the servo motor has three wires. The brown colored wire is the GND, the red wire is for VCC and the orange wire is for the digital pin [3].

In order to speed up the development process, the proponents used the Servo.h arduino library. This allowed servo motor control through the use of the attach, read and write commands available in this library.

Fig. 4. Servo Motor [2]

III. Methodology

The project was patterned after the Arduino Solar Tracker project by ElectronicsHub with there being modifications due to the available materials [4]. This project consists of two main parts: (1) the light-sensitive circuit and (2) the rotating wooden frame. The light-sensitive circuit makes use of light dependent resistors to determine which direction there is more light and rotating wooden frame makes use of servo motors to rotate the solar panel to face the mentioned direction. Both schematic diagrams can be seen in Fig. 5 with the former in a green box and the latter in a red box [4].

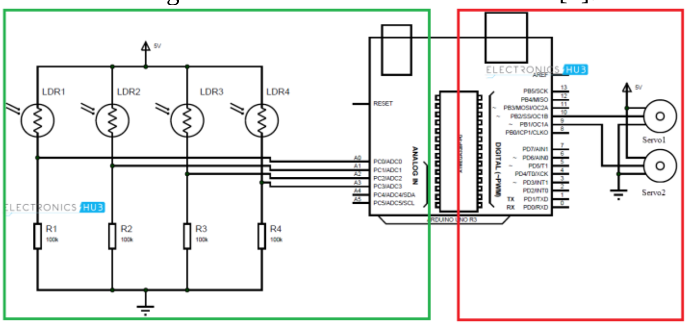

Fig. 5. Schematic Diagram of Project. The green box contains the light-sensitive circuit while the red box contains the servo units for the rotating frame [4].

A. Light-Sensitive Circuit using Light Dependent Resistors

For the light sensitive circuit, four light dependent resistors, four 100kΩ resistors, an assortment of jumper cables, and a rectangular piece of cardboard were used.

First, the outline of the solar panel was traced on the cardboard using a pencil. On the four corners of the solar panel, 8 holes were dug in total to fit the wires of the light dependent resistors. The researchers made sure that the light dependent resistors were high enough such that the solar panel would not cast a shadow on them at any angle. This setup is illustrated in Fig. 6 with the red circles representing the light dependent resistors.

Second, the circuit seen in the green box in Fig. 6was recreated on the other side of the rectangular piece of cardboard. Each component was glued onto the cardboard using a hot glue gun and their wires were soldered together as illustrated in Fig. 7. The wires were colored red, green, and blue depending on their use. Red wires in the figure are connected to the 5V pin of the Arduino while blue wires are connected to the GND pin of the Arduino. The green wires are connected to the analog input pins of the Arduino. The figure also illustrates which pin these outputs are connected to. The resistance on LDR0 was connected to pin A0 and the pattern remains for LDR1, LDR2, and LDR3.

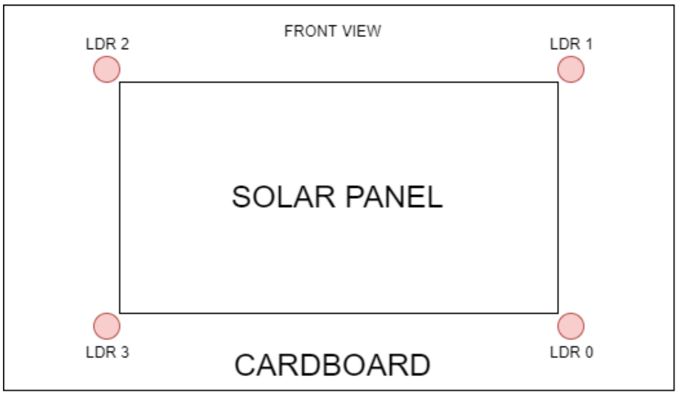

Fig. 6. Front View of the Rectangular Cardboard Containing the Light-Sensitive Components. Red circles represent the Light-Dependent Resistors.

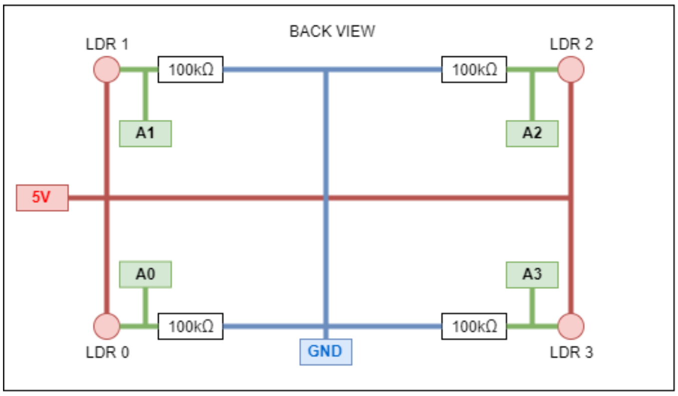

Fig. 7. Back View of the Rectangular Cardboard the Circuit Connections. Red wires are connected to the Arduino 5V pin, blue wires are connected to the GND pin, and the green wires are connected to the analog input pins.

B. Rotating Wooden Frame using Servos

The rectangular piece of cardboard seen in Figures 6 and 7 needed to be attached to a rotating frame that could rotate the cardboard to face the light source. For this project, a two-axis frame was chosen to accommodate horizontal and vertical rotation. For the rotating frame, two 5V servos, four wide popsicle sticks, and jumper cables were used.

Popsicle sticks were joined together using hot glue to create a wooden frame to attach to the rectangular cardboard. These popsicle sticks were screwed and glued onto the rotating arms of the servo motors as illustrated in Fig. 8.

In the illustration, the orange blocks represent the popsicle sticks while the blue blocks represent the servo motors. The red blocks represent the wires of the servo motor while the green blocks represent the 5V, GND, and analog input wires coming from the cardboard. It is important to note that the illustration is not to scale and that these components were placed and were long enough not to be an obstruction to other components when rotating. The wires and the popsicle sticks' lengths were crucial to ensure that the solar tracker rotated smoothly.

The wires from the servo motor and the light-sensitive circuit were then attached to the Arduino Uno unit based on the connections seen in Fig. 3.1. To act as a counterweight, the bottom servo motor was attached to a cardboard box. The Arduino Uno was placed inside the box and a hole was cut in the cardboard box in order to cleanly hold the wires and route them to the Arduino unit.

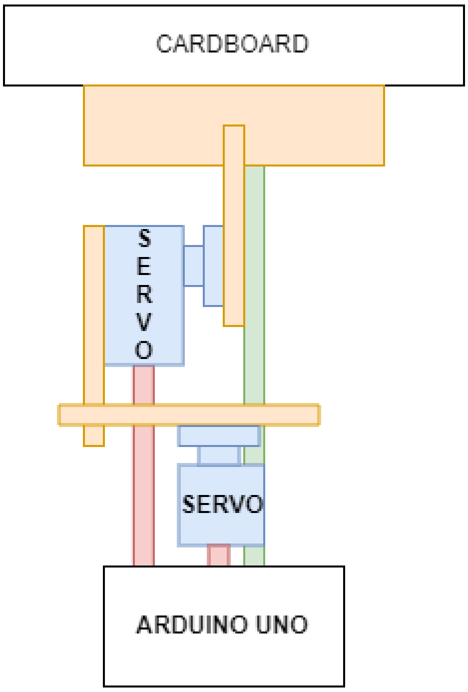

Fig. 8. Not to Scale Illustration of Rotating Wooden Frame Setup. Orange blocks represent the wooden popsicle sticks. Blue blocks are servo units. Red blocks are the wires of the servo units. Green blocks are the wiring from the light-sensitive circuit on the cardboard.

IV. Results and Discussion

For the final project, the proponents used the Servo.h Arduino library in order to control the two servo motors used in the light following solar panel. As mentioned in the methodology, the four LDRs were placed on the top right, top left, bottom left and bottom right of the solar panel. The resistance readings from these LDRs were used as input to indicate which position had the most light. The two servo motors were responsible for the vertical and horizontal motor movements of the light following solar panel.

The following discussions will delve into the project's source code which was based on an Electronics Hub Open Source Project. Variations in the code were due to differences in hardware setup and also due to the different LDR readings being received.

```
#include <Servo.h>
//defining Servos
Servo servoHorizontal;
int servoh = 0;
int servohLimitHigh = 180; //highest possible horizontal rotation
int servohLimitLow = 0; ////lowest possible horizontal rotation

Servo servoVertical;
int servov = 0;
int servovLimitHigh = 120; //highest possible vertical rotation
int servovLimitLow = 40; //lowest possible vertical rotation
//Assigning LDRs
int ldrtopLeft = 2; //top left LDR green
int ldrtopRight = 1; //top right LDR yellow
int ldrbotLeft = 3; // bottom left LDR blue
int ldrbotRight = 0; // bottom right LDR orange
```

Fig. 9. Library Include and Variable Declarations

As seen in Fig. 9, the proponents included the Servo.h Arduino library that made it possible for servo motor pin assignments, read and write commands. The servo objects, servoHorizontal and servoVertical were initialized so that the proponents could control the servo motors responsible for the horizontal and vertical movements respectively. It can be seen in the figure mentioned that the horizontal rotation ranges from 0 to 180 degrees. This meant that the solar panel could move freely from left to right using the minimum and maximum angle rotations that the servo motor can manage. However, for the vertical movement, the proponents decided to limit the angle of rotation from 40 to 120 degrees. The reason behind this is that the solar panel should always be angled since the source of light will never be located approximately below 40 degrees. Based on the professor's feedback, this scenario is possible since the sun sets and will eventually be located almost horizontally in relation to the solar panel. However, Changing the lower vertical limit to 0 degrees would damage the hardware setup since the solar panel holder comes in contact with the arduino container if it goes below 40 degrees. See Fig. 10.

As seen in Fig. 9, the proponents included the Servo.h Arduino library that made it possible for servo motor pin assignments, read and write commands. The servo objects, servoHorizontal and servoVertical were initialized so that the proponents could control the servo motors responsible for the horizontal and vertical movements respectively. It can be seen in the figure above that the horizontal rotation ranges from 0 to 180 degrees.

For the Arduino code's setup function, the baud rate was set to 9600 for debugging purposes and the two servo motors were assigned to their respective digital pins on the Arduino Uno. The write commands found in Fig. 11 were used

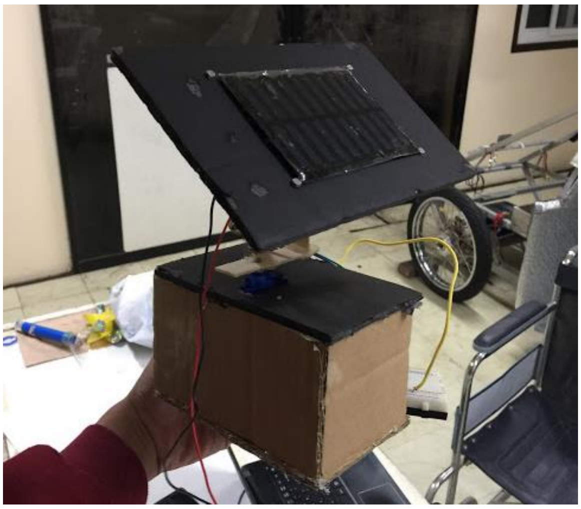

Fig. 10. Solar Panel Container and Servo Motor Setup

```
void setup ()
{
  Serial.begin(9600);
  servoHorizontal.attach(9);
  servoHorizontal.write(0);
  servoVertical.attach(10);
  servoVertical.write(0);
  delay(500);
}
```

Fig. 11. Arduino Code Setup Function

to set the panel to its initial position. During initial testing, the proponents encountered a problem since the developed code manifested differently when it was uploaded on the Arduino Uno. Eventually, the proponents realized that the pin assignments for the vertical servo motor was interchanged with the pin assignment for the horizontal servo motor which was the cause of the problem.

For the initial part of the loop function, variables servoh and servov were assigned feedback values from the servo motors, this was to ensure that were no instantaneous change in angle rotation since the servo motors know their previous states. Next, the four LDR readings were stored to integer variables topLeft, topRight, botLeft and botRight. Multiple serial print commands were used in order to monitor the readings from the LDR during runtime. After this, the averages of each LDR pair locations were calculated. This was done in order for the system to know which part of the solar panel is receiving the most average light. This was a more robust approach since the readings differ for each setting, and assigning constant values in the beginning would make it difficult for the system to respond properly when placed in a

```
void loop()
{

  servoh = servoHorizontal.read();
  servov = servoVertical.read();
  //capturing analog values of each LDR
  int topLeft = analogRead(ldrtopLeft);
  int topRight = analogRead(ldrtopRight);
  int botLeft = analogRead(ldrbotLeft);
  int botRight = analogRead(ldrbotRight);

  Serial.print("top left: ");
  Serial.println(topLeft);
  Serial.print("top right: ");
  Serial.println(topRight);
  Serial.print("bottom left: ");
  Serial.println(botLeft);
  Serial.print("bottom right: ");
  Serial.println(botRight);
  Serial.println("------------");

  // calculating average
  int avgtop = (topLeft + topRight) / 2; //average of top LDRs
  int avgbot = (botLeft + botRight) / 2; //average of bottom LDRs
  int avgleft = (topLeft + botLeft) / 2; //average of left LDRs
  int avgright = (topRight + botRight) / 2; //average of right LDRs
```

Fig. 12. Arduino Code Loop Function

different setting. For example, if the topLeft and topRight LDRs had low analog readings, this meant that the top portion of the solar panel is receiving the most light on average. This will prompt the system to maneuver itself towards that position. See Fig. 12 for the source code.

Next, the code found in Fig. 13 performs a straightforward algorithm that allows the system to move horizontally and vertically towards a position with the most light. The obtained average for each LDR position pairs are compared. If the top average reading is less than the bottom average reading, then it means that there is more light at the top portion of the solar panel since more light equals less resistance vice-versa. The code increments or decrements one degree on the servo motor every time until it reaches its vertical. Next, if the left average reading is less than the right average reading then it means that there is more light at the left portion of the solar panel vice-versa. Again, the code increments or decrements one degree on the servo motor every time until it reaches its horizontal limits.

One problem found was that it does not find the shortest path towards the location with the most received light since it first moves vertically and then horizontally which takes more time than just moving diagonally. If the light source is not within the solar panel's line of sight, there is no way for it to find the light source unless the source is moving or if the proponents reposition the system. A recommendation made by the professor was to include a timer which would prompt the system to rotate back in the opposite direction once a certain time has elapsed.

```
  if (avgtop < avgbot)
  {
    servoVertical.write(servov + 1);
    if (servov > servovLimitHigh)
    {
      servov = servovLimitHigh;
      servoVertical.write(servov);
      Serial.println("HIGH HAS BEEN BREACHED");
    }
    delay(5);
  }
  else if (avgbot < avgtop)
  {
    servoVertical.write(servov - 1);
    if (servov < servovLimitLow)
    {
      servov = servovLimitLow;
      servoVertical.write(servov);
    }
    delay(5);
  }
  else
  {
    servoVertical.write(servov);
  }

  if (avgleft > avgright)
  {
    servoHorizontal.write(servoh - 1);
    if (servoh > servohLimitHigh)
    {
      servoh = servohLimitHigh;
    }
    delay(5);
  }
  else if (avgright > avgleft)
  {
    servoHorizontal.write(servoh + 1);
    if (servoh < servohLimitLow)
    {
      servoh = servohLimitLow;
    }
    delay(5);
  }
  else
  {
    servoHorizontal.write(servoh);
  }
  delay(50);
}
```

Fig. 13. Arduino Code Loop Function Continuation

V. Conclusion

In this final project, the students were able to successfully construct and develop a light following solar panel using light dependent resistors and servo motors. The system had certain limitations in terms of optimizing its rotational routes and trying to renavigate itself when the light source is not within its line of sight. Despite this, the system is still able to position the solar panel to a location that receives the most light on average given a certain range of rotational motion.

VI. References

[1] E. Notes, "Light Dependent Resistor LDR: Photoresistor," *Electronics Notes*. [Online]. Available: https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php. [Accessed: 14-Dec-2019].

[2] *Servo Motor Basics, Working Principle & Theory*, 01-Aug-2015. [Online]. Available:

https://circuitdigest.com/article/servo-motor-basics. [Accessed: 14-Dec-2019]
[3] “Servo,” *Arduino*. [Online]. Available: https://www.arduino.cc/en/reference/servo. [Accessed: 14-Dec-2019].
[4] Administrator, “Arduino Solar Tracker,” *Electronics Hub*. [Online]. Available: https://www.electronicshub.org/arduino-solar-tracker/?fbclid=IwAR2Ng-vCIioqLzl-VSlIV_YYBliZKL5Y9ZRwabzeQcm643gbwEDUHSQKz7Y. [Accessed: 14-Dec-2019].
[5] Eke, R. and Senturk, A. (2012). Performance comparison of a double-axis sun tracking versus fixed PV system. *Solar Energy*, 86(9), pp.2665-2672.

APPENDIX

```
#include <Servo.h>
//defining Servos
Servo servoHorizontal;
int servoh = 0;
int servohLimitHigh = 180; //highest possible horizontal
rotation
int servohLimitLow = 0; ////lowest possible horizontal rotation

Servo servoVertical;
int servov = 0;
int servovLimitHigh = 120; //highest possible vertical rotation
int servovLimitLow = 40; //lowest possible vertical rotation
//Assigning LDRs
int ldrtopLeft = 2; //top left LDR green
int ldrtopRight = 1; //top right LDR yellow
int ldrbotLeft = 3; // bottom left LDR blue
int ldrbotRight = 0; // bottom right LDR orange

void setup ()
{
  Serial.begin(9600);
  servoHorizontal.attach(9);
  servoHorizontal.write(0);
  servoVertical.attach(10);
  servoVertical.write(0);
  delay(500);
}

void loop()
{

  servoh = servoHorizontal.read();
  servov = servoVertical.read();
  //capturing analog values of each LDR
  int topLeft = analogRead(ldrtopLeft);
  int topRight = analogRead(ldrtopRight);
  int botLeft = analogRead(ldrbotLeft);
  int botRight = analogRead(ldrbotRight);

  Serial.print("top left: ");
  Serial.println(topLeft);
  Serial.print("top right: ");
  Serial.println(topRight);
  Serial.print("bottom left: ");
  Serial.println(botLeft);
  Serial.print("bottom right: ");
  Serial.println(botRight);
  Serial.println("-------------");

  // calculating average
  int avgtop = (topLeft + topRight) / 2; //average of top LDRs
  int avgbot = (botLeft + botRight) / 2; //average of bottom
LDRs
  int avgleft = (topLeft + botLeft) / 2; //average of left LDRs
  int avgright = (topRight + botRight) / 2; //average of right
LDRs

  if (avgtop < avgbot)
  {
    servoVertical.write(servov + 1);
    if (servov > servovLimitHigh)
    {
      servov = servovLimitHigh;
      servoVertical.write(servov);
      Serial.println("HIGH HAS BEEN BREACHED");
    }
    delay(5);
  }
  else if (avgbot < avgtop)
  {
    servoVertical.write(servov - 1);
    if (servov < servovLimitLow)
    {
      servov = servovLimitLow;
      servoVertical.write(servov);
    }
    delay(5);
  }
  else
  {
    servoVertical.write(servov);
  }

  if (avgleft > avgright)
  {
    servoHorizontal.write(servoh - 1);
    if (servoh > servohLimitHigh)
    {
      servoh = servohLimitHigh;
    }
    delay(5);
  }
  else if (avgright > avgleft)
  {
    servoHorizontal.write(servoh + 1);
    if (servoh < servohLimitLow)
```

```
    {
      servoh = servohLimitLow;
    }
    delay(5);
  }
  else
  {
    servoHorizontal.write(servoh);
  }
  delay(50);
}
```