Taha Aflouk

A00300601

Design report

Solar Tracker

[Intro 1](#_Toc1715753966)

[Context/Rationale: 1](#_Toc713055038)

[Specification: 1](#_Toc613788831)

[Design 1](#_Toc1408786308)

[Build 1](#_Toc1801743437)

[Code 1](#_Toc499517801)

[Test 4](#_Toc400920791)

[Conclusion 5](#_Toc1249224425)

# Intro

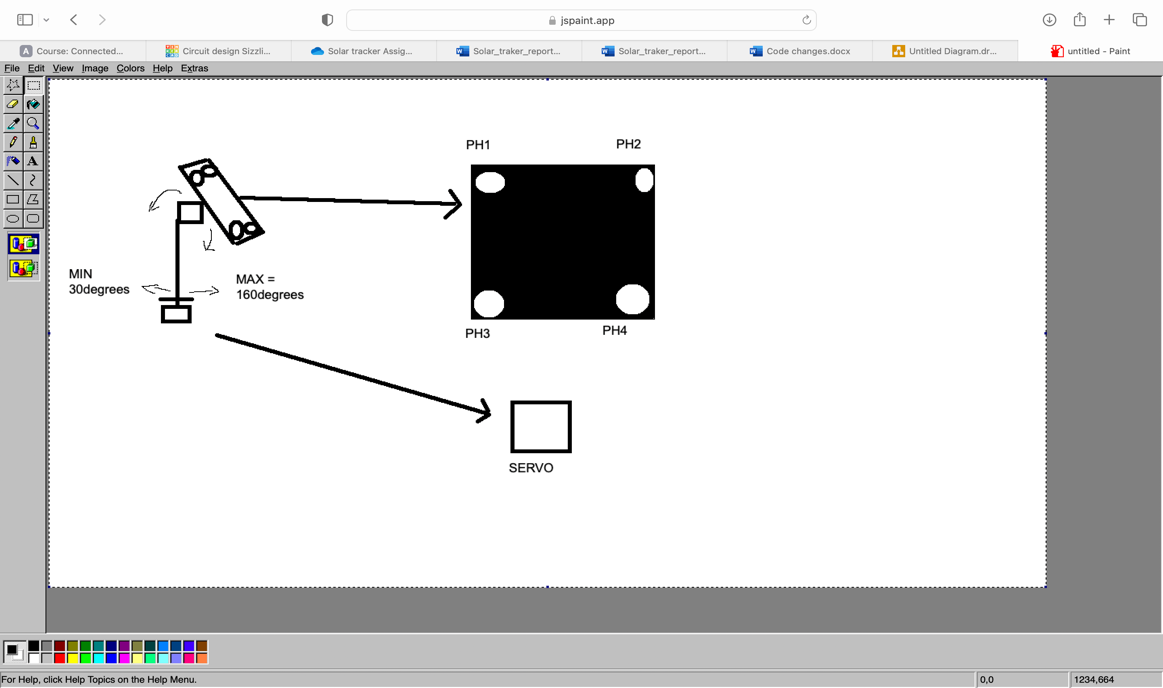
Solar tracker to track the light, the solar board or panel finds the side that has the lightest which is has more energy and then it turns to that side so to apply that idea I am using Arduino controller, 2 servos, 4 Photoresistor, 1 button, and 1 LED.

Context/Rationale:

The main point of this system is to generate as much energy as possible from the light and the purpose of the solution is to make the system more efficient by getting higher voltage by tracking the light and it is friendly for the environment for some degrees. So, in that way we can save electricity or store it in a safe way and then use the electricity at different times that does not have enough output voltage from the system to feed what we need.

Specification: Explanation of what the program should do.

The system should read values from the light in general, then based on that value we control the servos to move the board RIGHT, LEFT, UP, and DOWN in my program assuming that the board is a solar panel board that converts the light to energy like that one used in real life as we know most solar energy systems work on the sun and the sun does not stay on one side in the picture below we see the components.



Horizontal servo = The bottom servo

Vertical servo = The top servo

PH = Photoresistor

TOP = PH1 & PH2

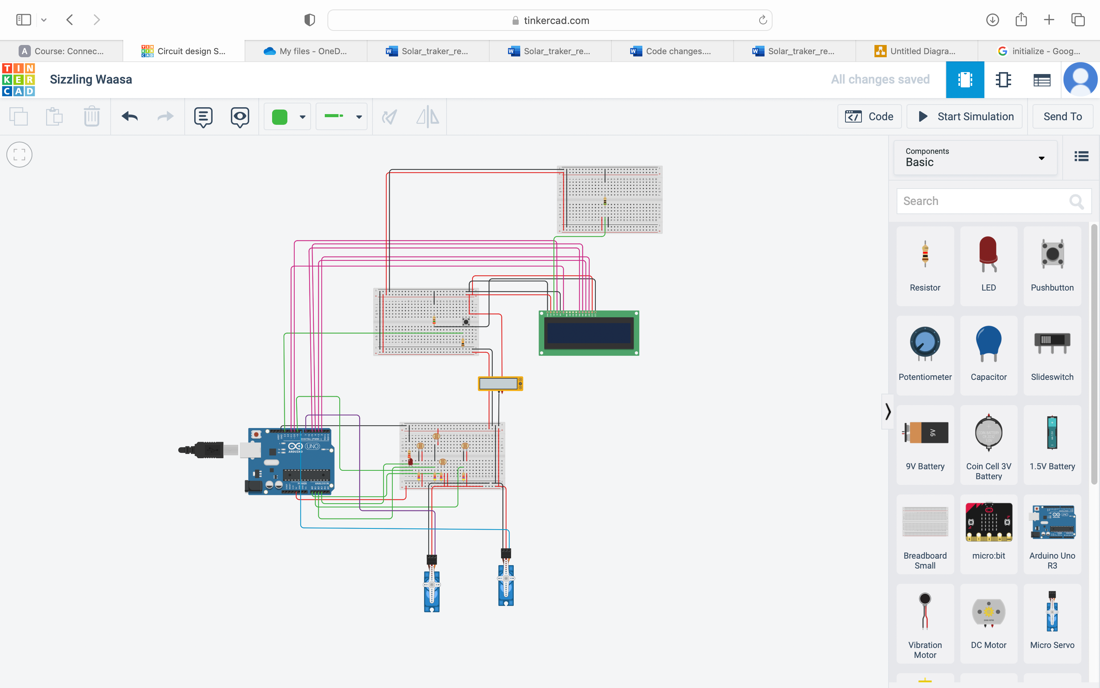
BOTTOM = PH3 & PH4

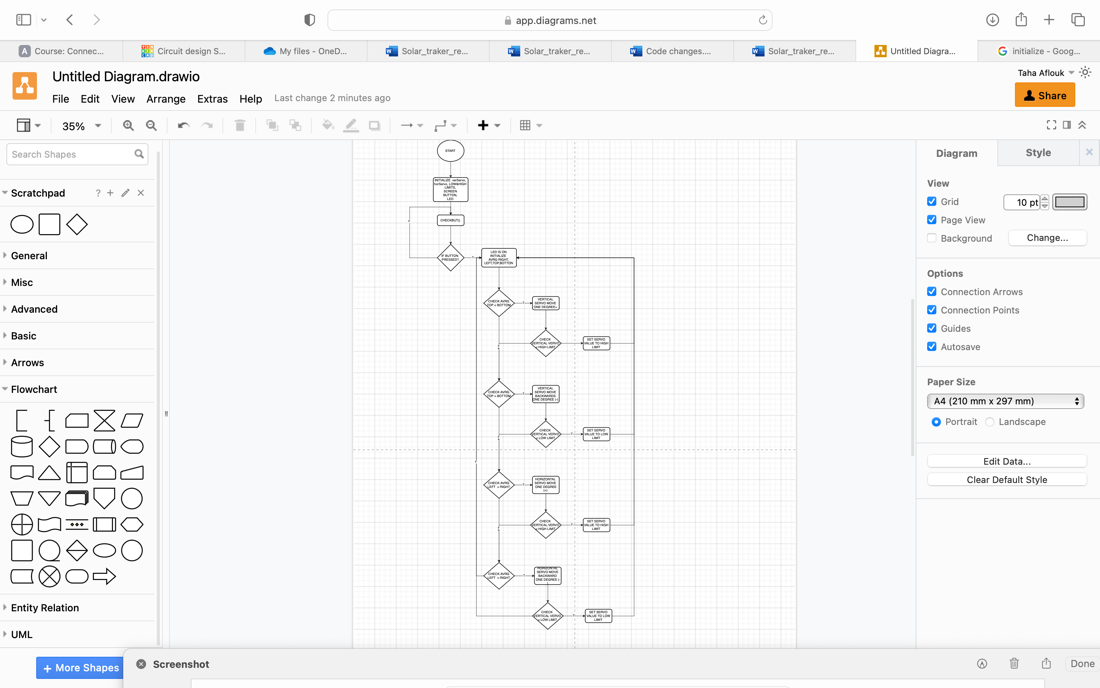
RIGHT = PH2 & PH4

LEFT = PH1 & PH3

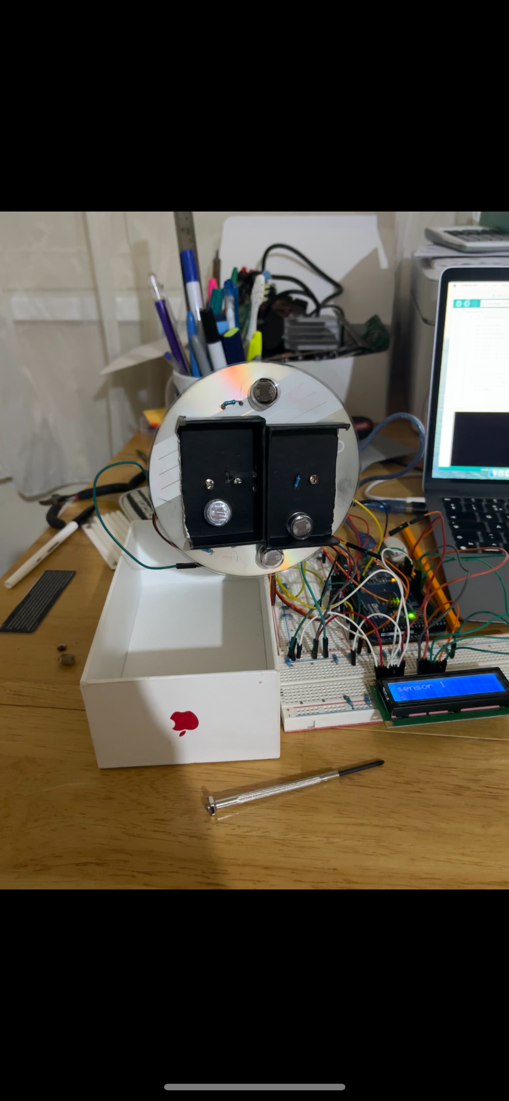
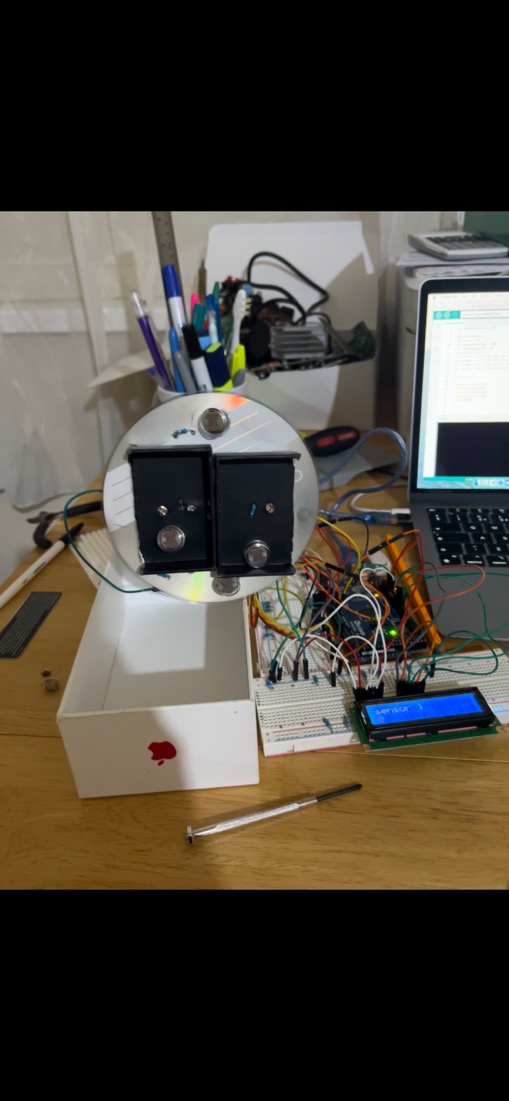
So, we get the average for each side and then we must move the board that side for example, if the left side has the highest value, we must move the vertical to the right side, but we must keep in mind that we cannot move more than the limit.

Design





Build



## Code

// C++ code

//

#include <Servo.h>

#include <LiquidCrystal.h>

// -- CONSTANT --

const int PHOTORESITSTOR1 = 0;

const int PHOTORESITSTOR2 = 1;

const int PHOTORESITSTOR3 = 2;

const int PHOTORESITSTOR4 = 3;

const int rs = 10, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// -- GLOBALS --

int readPerSec = 5000;

int redLED = 9;

Servo servoHori;

int currentServoHoriValue = 20;

int servoHoriLimitHigh = 160; // set the limits for the servos UP DOWN RIGHT LEFT

int servoHoriLimitLow = 20;

Servo servoVerti;

int currentServoVerValue = 20;

int servoVerLimitHigh = 160;

int servoVerLimitLow = 20;

int flag = 0;

const int BUT\_ON = 13;

int ldrTopLeft = 2; //top left LDR green

int ldrTopRight = 1; //top right LDR yellow

int ldrBottomLeft = 3; // bottom left LDR blue

int ldrBottomRight = 0; // bottom right LDR orange

// -- FUNCIONTS --

void setup() {

// setUp the servos and the lcd screen

Serial.begin(9600);

pinMode(BUT\_ON, INPUT);

pinMode(redLED, OUTPUT);

lcd.begin(16, 2);

servoHori.attach(8);

servoVerti.attach(7);

servoHori.write(20);

servoVerti.write(20);

lcd.clear();

for(int i = 5 ; i >= 0; i--){

lcd.setCursor(0,0);

lcd.print("Please Wait...");

lcd.setCursor(0,1);

lcd.print(i);

lcd.print(" Second");

delay(10);

}

}

void checkBut(){

int ButtonStateON = digitalRead(BUT\_ON);

if(ButtonStateON == HIGH)

{

Serial.println("button is pressed.");

flag = 1;

Serial.print("flag is ");

Serial.println(flag);

}

}

void startServos(){

flag = 1;

digitalWrite(redLED, HIGH);

currentServoHoriValue = servoHori.read();

currentServoVerValue = servoVerti.read();

//capturing analog values of each LDR

int readTopLeft = analogRead(ldrTopLeft);

int readTopRight = analogRead(ldrTopRight);

int readBottomLeft = analogRead(ldrBottomLeft);

int readBottomRight = analogRead(ldrBottomRight);

// calculating average

int avgTop = (readTopLeft + readTopRight) / 2; //average of top LDRs

int avgBottom = (readBottomLeft + readBottomRight) / 2; //average of bottom LDRs

int avgLeft = (readTopLeft + readBottomLeft) / 2; //average of left LDRs

int avgRight = (readTopRight + readBottomRight) / 2; //average of right LDRs

// check (bottom & top, top & bot, right & left, left & right)

if (avgTop < avgBottom) // check for avrage bottom and top

{

// increament the vertically servo

servoVerti.write(currentServoVerValue +1);

if (currentServoVerValue > servoVerLimitHigh) // check if the servo goes over limit

{

currentServoVerValue = servoVerLimitHigh; // set the servo to the given limit

}

delay(10);

}

else if (avgBottom < avgTop) // again we check the bottom against the top

{

servoVerti.write(currentServoVerValue -1); //. we decreament the sensor value by 1

if (currentServoVerValue < servoVerLimitLow) // if the servo value goes below the limit

{

currentServoVerValue = servoVerLimitLow; // set the servo value to limit

}

delay(10);

}

else

{

servoVerti.write(currentServoVerValue); // else keep the servo value as it is on the current position

}

if (avgLeft > avgRight) // here we repeat the same step for the other sides

{

servoHori.write(currentServoHoriValue +1);

if (currentServoHoriValue > servoHoriLimitHigh)

{

currentServoHoriValue = servoHoriLimitHigh;

}

delay(10);

}

else if (avgRight > avgLeft)

{

servoHori.write(currentServoHoriValue -1);

if (currentServoHoriValue < servoHoriLimitLow)

{

currentServoHoriValue = servoHoriLimitLow;

}

delay(10);

}

else

{

servoHori.write(currentServoHoriValue);

}

delay(50);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("top ");

lcd.print(avgTop);

lcd.setCursor(0,1);

lcd.print(avgBottom);

}

void loop() {

checkBut();

if (flag == 1)

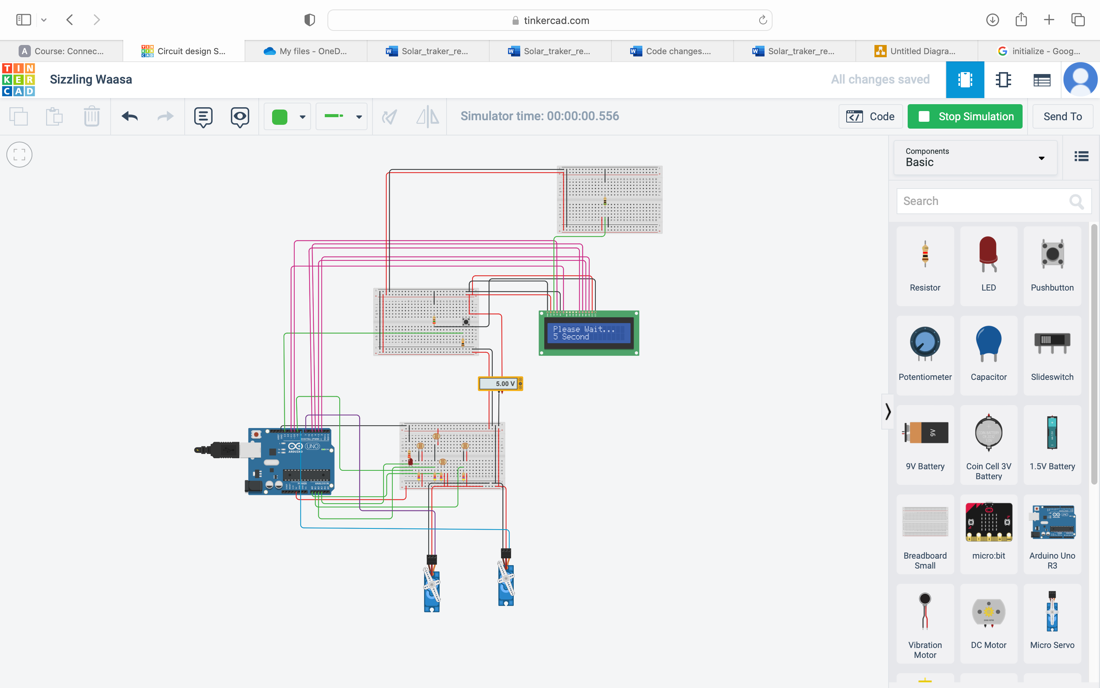
{

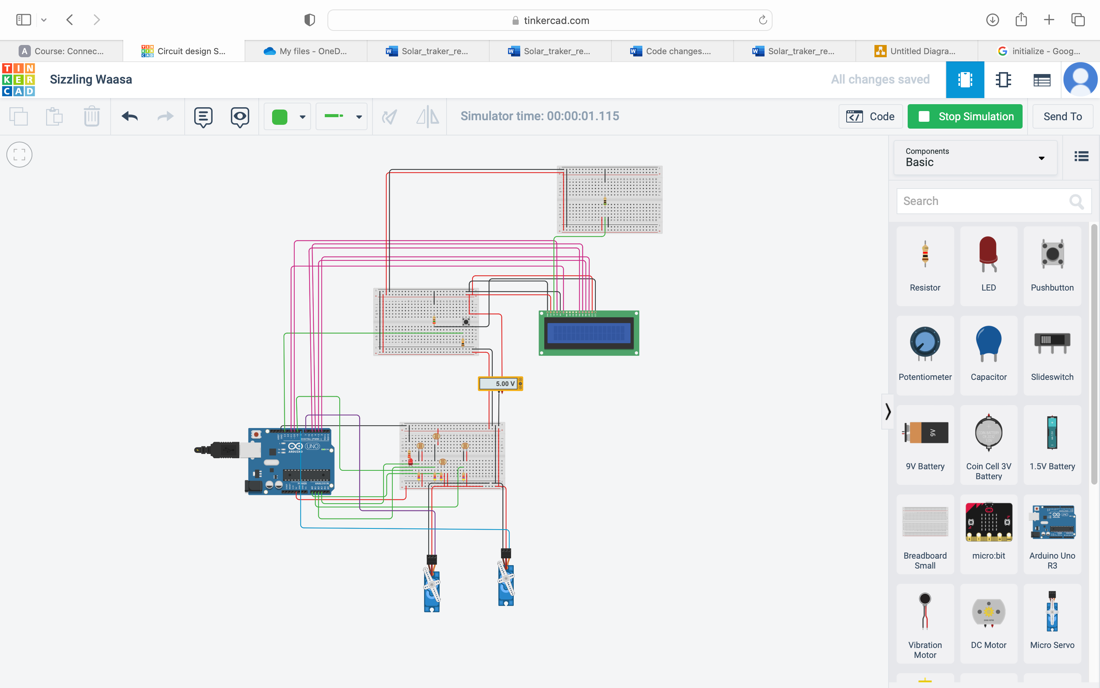
startServos();

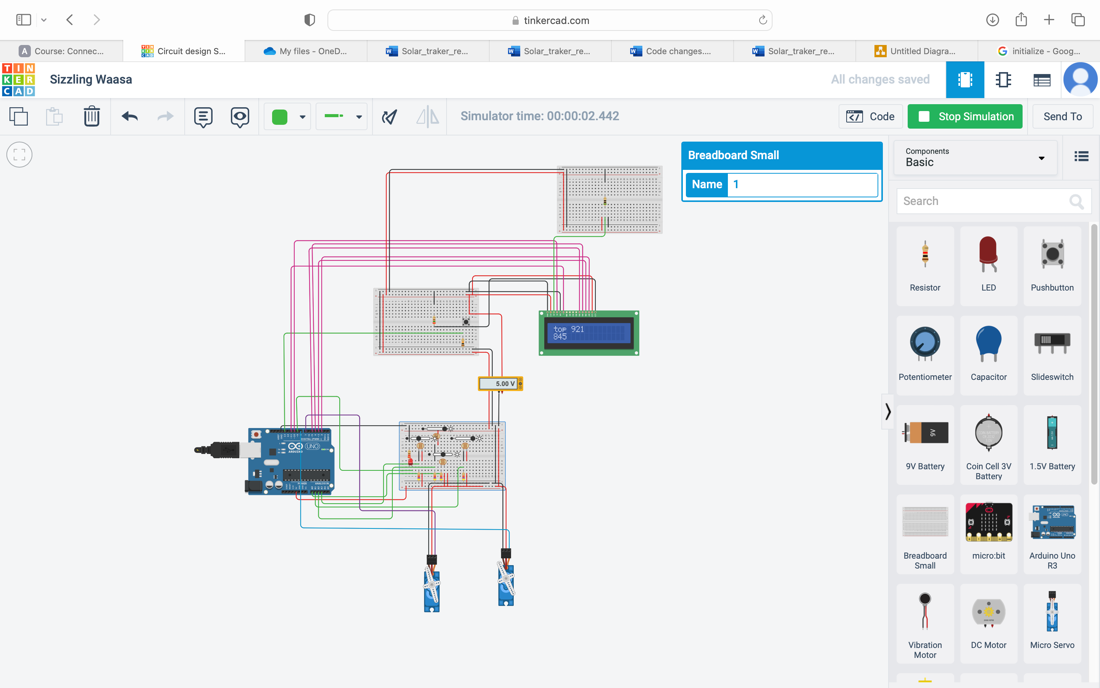
}

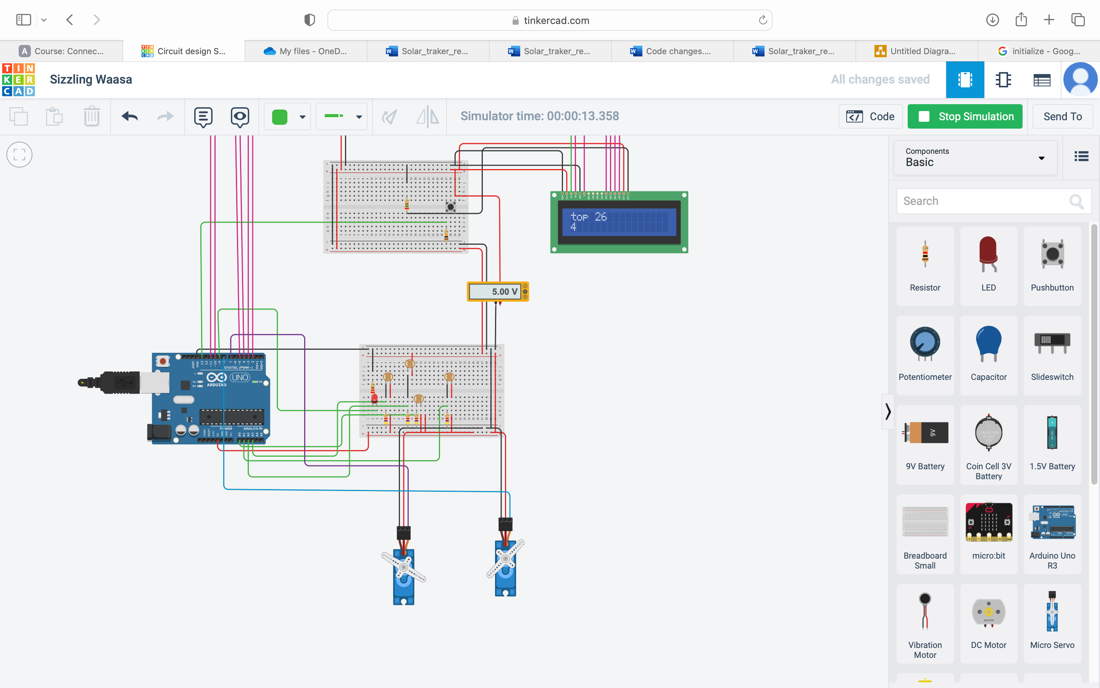
}

Test









Conclusion

The idea was to create a solar tracker system so the actual system runs on Arduino device. However, we had to use some mechanical technique in the system and mathematics (calculation) to make the system work as we expected.

So, in my opinion we must look out of the box sometimes to make the program efficient as much as possible by trying different techniques.