# Week 6 – Semiconductor Components

**Lab Report**

## Related image Image result for photodetector

|  |  |  |
| --- | --- | --- |
| Names: | Jacqueline Radding | Date: 10/22/19 |
|  |  |  |
| EE 151 | Section: Online Monday | Lab Bench (In-lab Only): |

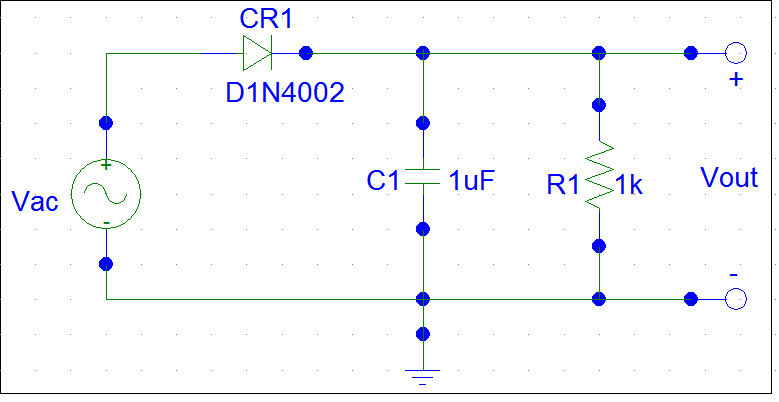
**Learning Objectives for this Lab:**

* Predict the flow of currents in a circuit containing diodes, based on whether or not they are forward biased properly.
* Able to determine the Cut-In voltage of a diode from experimental data.
* Recognize the anode and cathode leads of a physical diode, based on its markings.
* Able to use external triggering on an oscilloscope. (In-Lab only)
* Draw a simple rectifier circuit (half-wave) and describe its operation.
* Describe the operation of a full-wave rectifier circuit.
* Know how to connect a photoresistor in a voltage divider to detect light level.
* Command an Arduino servo motor to different angular positions, using an Arduino Library

**Section 2 – Half-Wave Rectifier Circuit**

**Oscilloscope**

**Ch 2**



1N4007

**GenOut**

1 KHz

5 Vpp

**Ch1**

#### **Fig. 1** Half-Wave Rectifier Circuit

**Section 2, step d: source AC and half-wave rectified waveforms**

## Diagram Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Measurement** | **Input Signal**  **(Function Generator Output)** | **Rectifier Output Signal** |
| **Peak Positive Voltage (compared to ground)** | + volts | + volts |
| **Peak Negative Voltage**  **(compared to ground)** | * volts | * volts |

## Table 1. Half-Wave Rectifier Peak Voltages

## Questions: Section 2

1. Identify which half of the source sinusoidal (AC) signal (the positive voltage half or the negative voltage half) the diode CR1 is allowing current to flow through it. Note that a diode conducts when it is forward-biased: the voltage is positive on the anode with respect to the voltage on the cathode. Current in a conducting diode flows in the direction of the arrow symbol.

**Current conducting when the AC input voltage is: the positive voltage half**

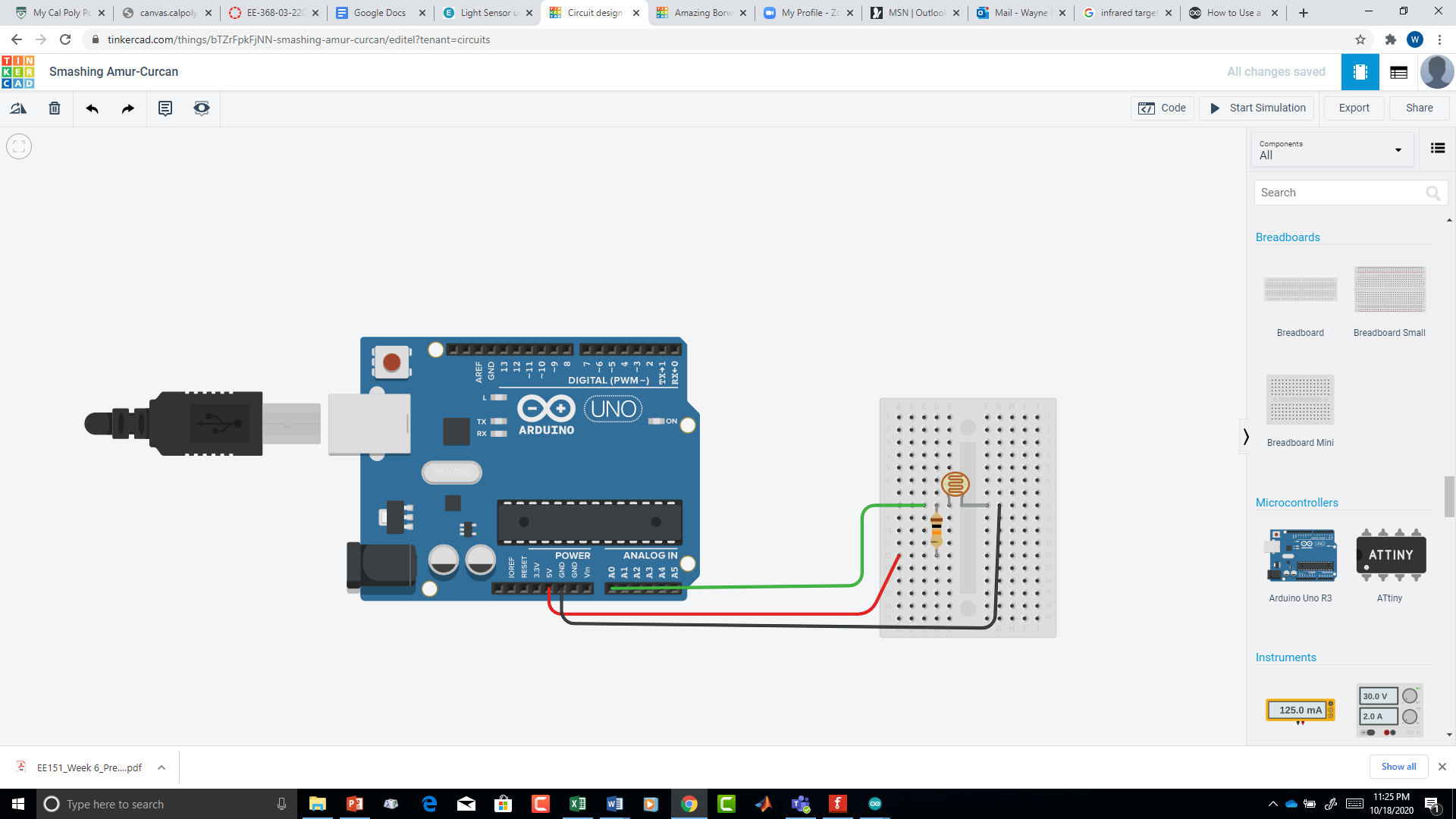
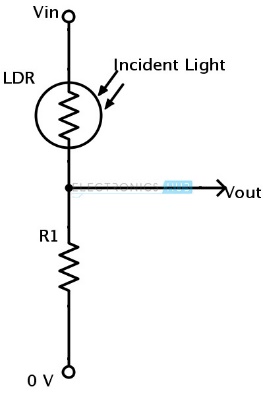
1. What is the forward cut-in voltage of the diode you tested, based on your peak voltage measurements? The forward cut-in voltage is the voltage drop across a diode when it is forward biased and conducting current.

**VF = ~5 volts**

1. From the Manufacturer’s Datasheet for a 1N2007 diode (available on-line or on Canvas), what is the maximum instantaneous forward voltage drop for this device? Under what testing conditions?

**VF max = 1 volts at forward current: iF = 30 amps, and at temperature: -65- 150 oC**

## Section 3) Arduino Project 1: Photoresistor Light Sensor

**Project 1A: Simple Light Sensor**

Create an Arduino sketch that does the following:

* + 1. Read the voltage Vout on analog input A0.
    2. Convert the analogRead value for Vout to a “brightness” percentage:

AnalogRead() value 1023 = 100 BrightnessPercent

* + 1. Output the BrightnessPercent to the Serial Monitor.

|  |  |
| --- | --- |
| **Measurement** | **Brightness Percent value** |
| **Sensor Blocked** | **0** |
| **Normal Ambient Light** | **15** |
| **Maximum Light Expected** | **93** |

**Arduino Sketch:**

// Photoresistor

//by Jacqueline Radding

//gets percentage brightness of a photoresistor

int val; // this will store the analog value

float BrightnessPercent; // percent

void setup() {

digitalWrite(A0, INPUT); // sets A0 as input

Serial.begin(9600); // sets up serial monitor

}

void loop() {

val = analogRead(A0);

BrightnessPercent = val / 1023.0; //calculate to find percentage

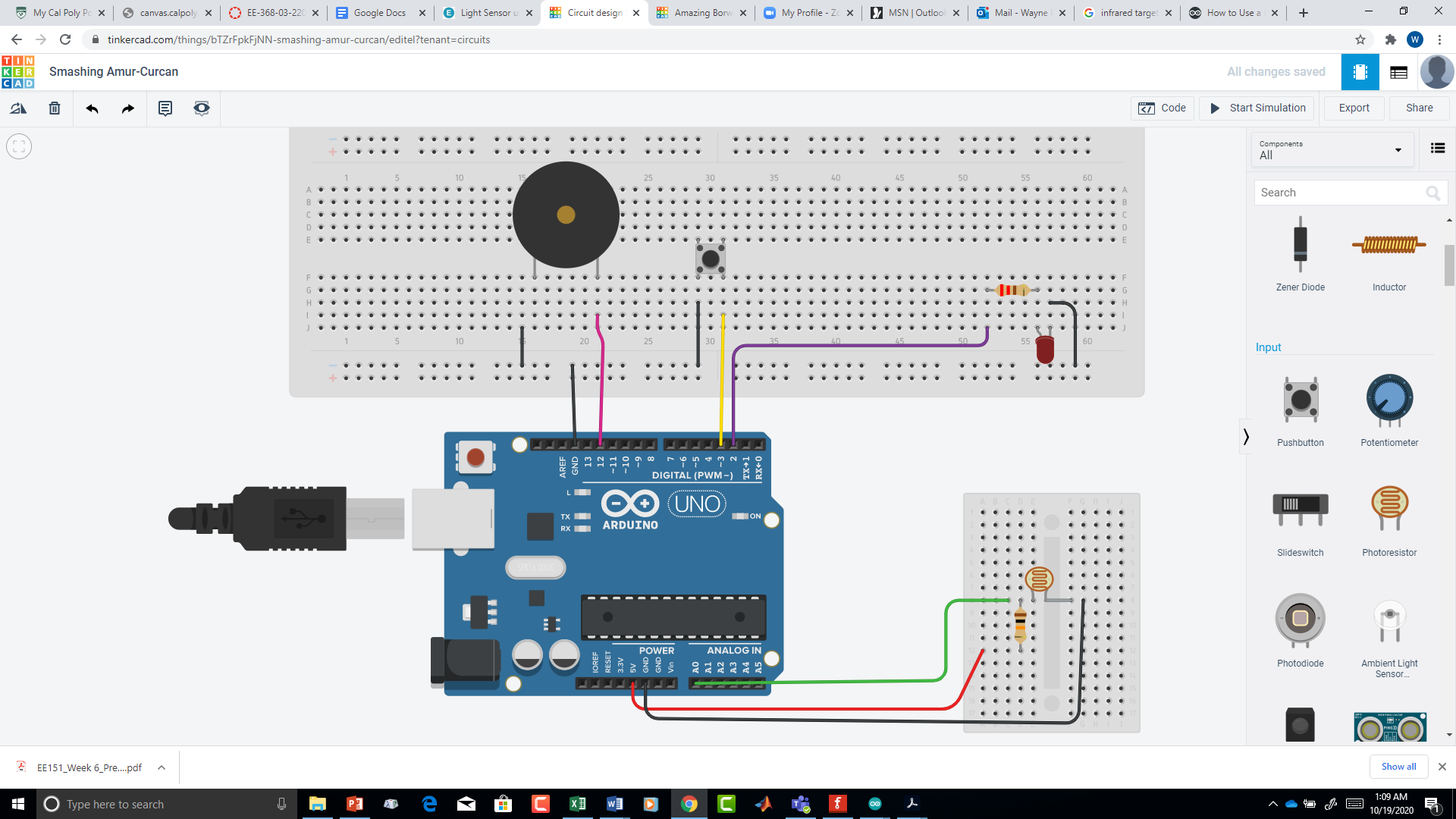
BrightnessPercent = BrightnessPercent \* 100; // to finalize the percentage by mulitplying it by 100 to get a percentage

Serial.println(BrightnessPercent); // print brighness percentage to serial

delay(100); // slight delay

}

**Project 1B: Enhanced Light Sensor**



**Arduino Sketch:**

// Photoresistor

//by Jacqueline Radding

//gets percentage brightness of a photoresistor and can turn button on and off with a beeper

int beeper = 12; // this is the pin the beeper is located

int val; // this will store the analog value

float BrightnessPercent; // percent

int LED = 2; // this will be

void setup() {

digitalWrite(A0, INPUT); // sets A0 as input

digitalWrite(beeper, OUTPUT); // beeper pin

digitalWrite(2, OUTPUT); //lightbulb pin

Serial.begin(9600); // sets up serial monitor

}

void loop() {

int light = digitalRead(3); // reads if button is not pressed

if (light == LOW){ // if button is pressed

digitalWrite(LED, HIGH); // Turn on the LED

delay(100); // small dealy

if (light == LOW){ // makes sure button is not pressed

digitalWrite(LED, LOW);

}

if(light == HIGH){ // if button is pressed

digitalWrite(LED, LOW);

}

}

val = analogRead(A0);

BrightnessPercent = val / 1023.0; //calculate to find percentage

BrightnessPercent = BrightnessPercent \* 100; // to finalize the percentage by mulitplying it by 100 to get a percentage

Serial.println(BrightnessPercent); // print brighness percentage to serial

delay(100); // slight delay

int brightmap = map(BrightnessPercent, 0, 100, 50, 1000); // this will be used for the tone. maps brighness to max/ min tone

if (BrightnessPercent < 5){ // if bright percent is below five no sound

noTone(beeper);

}

if (BrightnessPercent < 100 && BrightnessPercent>5){ // play mapped sound between 5 percent and 100

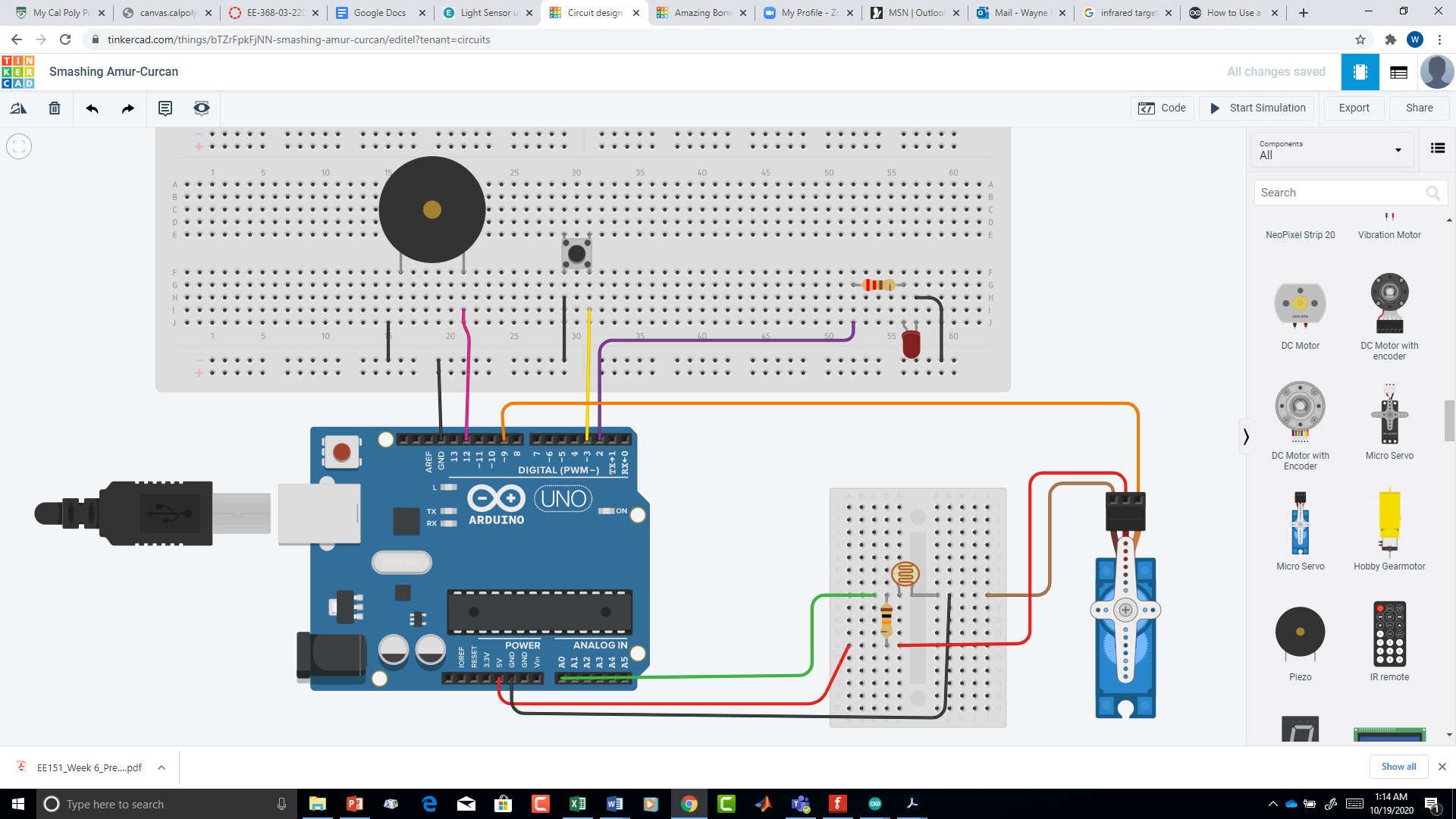
tone(beeper, brightmap);

}

delay(10);

}

## Section 4) Arduino Project 2: Searching Light Sensor



**Project 2B: Target Searching Light Sensor**

Pseudocode:

Repeat for servo pointing angle = 0 to 180 in steps of 1 degree:

Move the servo to the proper angle

Take a light reading and convert it to brightness%

Output a tone at the right frequency

If this brightness is > the previous brightest reading,

Then

Remember this as the brightest angle and brightest reading.

Move the servo to the brightest reading angle

Repeat while waiting

Play a tone frequency for the brightest reading for 500 msec.

Pause the tone for 500 msec.

**Arduino Sketch:**

// Photoresistor

//by Jacqueline Radding

//gets percentage brightness of a photoresistor

#include <Servo.h>

int beeper = 12; // this is the pin the beeper is located

int val; // this will store the analog value

int BrightnessPercent; // percent

int bigang = 0; // this will notify what angle pops up the brightest

int brightest; // brightest point

int ang; // angle

const int LED = 2; // LED pin

const int ServoPin = 9; //Servo connection

int i =0;

void setup() {

pinMode(LED, OUTPUT); // LED pin and output

pinMode(3, INPUT\_PULLUP); // button

digitalWrite(LED, HIGH); // Turn on the led for scanning

Servo myservo; // create servo object to control servo

myservo.attach(ServoPin); // attach servo on pin 9 to servo object

const int ServoPin = 9; //Servo connection

digitalWrite(A0, INPUT); // sets A0 as input

digitalWrite(beeper, OUTPUT); // beeper pin

delay(10);

Serial.begin(9600); // sets up serial monitor

while (i < 180){ // i is each angle, when angles are under 180, do this

val = analogRead(A0);

BrightnessPercent = val / 1023.0 \* 100; //calculate to find percentage

ang = i; // to seperate angle and i values

myservo.write(i); //set servo position to full CW (0 degrees)

delay(10); // delay

takebright(); // takes the brightness value at this angle//determines which angle is brightest

i = i+1; // i increases by one so the device can move to the next angle

}

delay(2000);

myservo.write(bigang); //set servo position to big angle

delay(2000);

myservo.write(0); // returns to 0 or full cw position

delay(100); // delay

int brightmap = map(brightest, 0, 100, 50, 1000); // this will be used for the tone. maps brighness to max/ min tone

tone(beeper, brightmap, 1000); // plays tone according to brightest percent

Serial.print("brightest %:"); // displays the brightest percent

Serial.println(brightest); //prints brightest point

Serial.println("angle:");

Serial.println(bigang); // brightest angle

delay(10);

}

void loop() {

int light = digitalRead(3); // button

if (light == LOW){ // if button pressed

digitalWrite(LED, HIGH); // Turn on the LED

delay(100);

if (light == LOW){ // turn off if not pressed

digitalWrite(LED, LOW);

}

if(light == HIGH){ // turn off if not pressed

digitalWrite(LED, LOW); // off

}

}

}

void takebright(){

int brightmap = map(BrightnessPercent, 0, 100, 50, 1000); // this will be used for the tone. maps brighness to max/ min tone

tone(beeper, brightmap, 100); // make a beep

if (brightest < BrightnessPercent){ // find the biggest brightness percent if brighness is greater than last reading

brightest = BrightnessPercent; // make brightest the new biggest value

bigang = ang; // make the brightest angle the new stored angle

}

}

**Make and upload to Canvas a brief video demonstrating your system using the LED to test the bright target finding capability.**