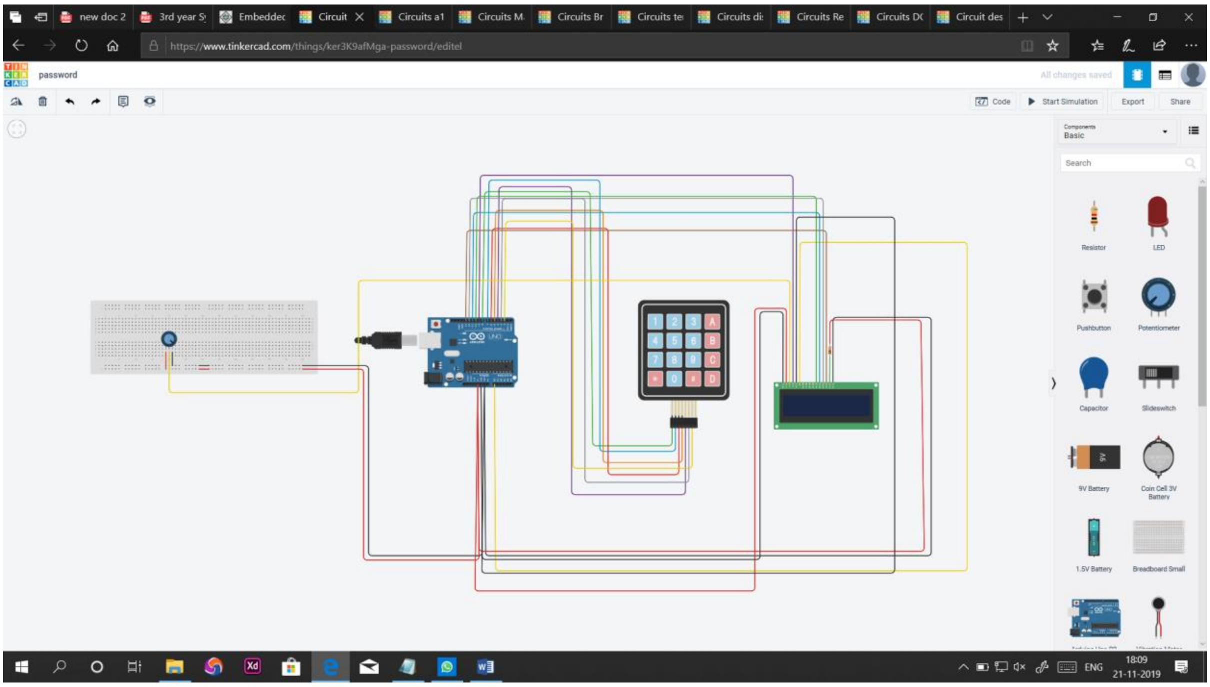
# KEYPAD WITH PASSWORD



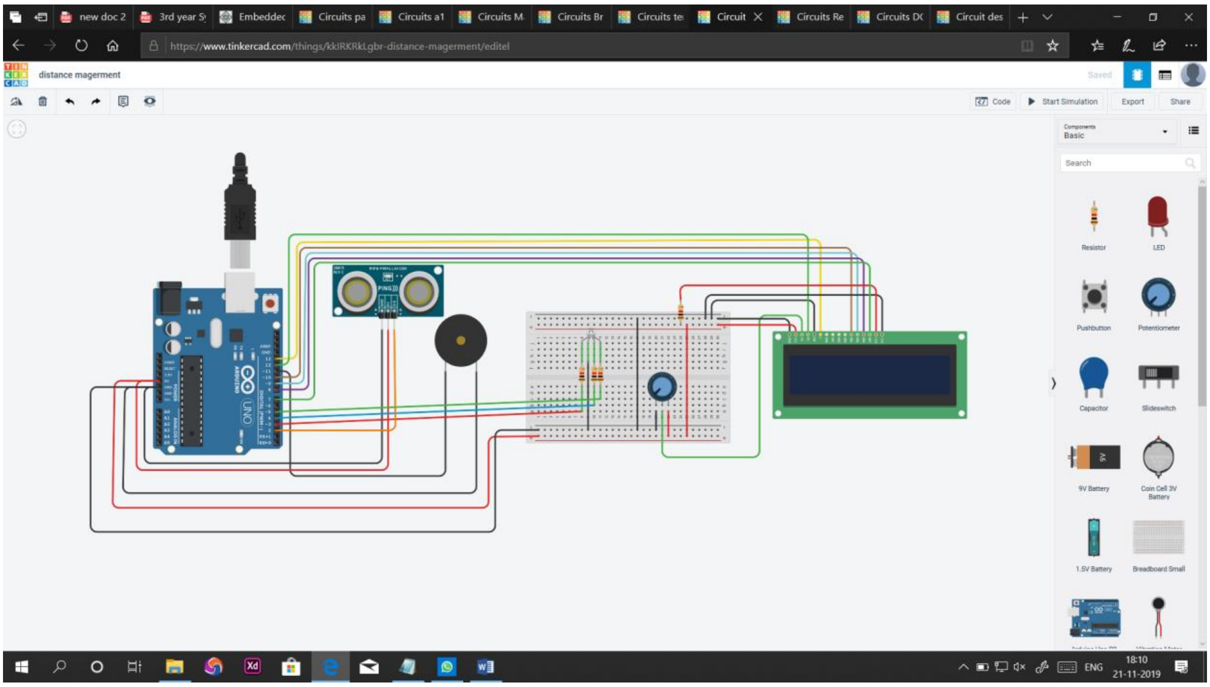
```
#include <Keypad.h>

#include <LiquidCrystal.h>


#define PASSWD "75675"

const byte row = 4;

const byte col = 3;

char keys[row][col] = {

  {'1','2','3'},

  {'4','5','6'},

  {'7','8','9'},

  {'*','0','#'}

};

byte rowPins[row] = {8,7,6,5};

byte colPins[col] = {4,3,2};
```

```
Keypad keypad = Keypad(makeKeymap(keys),rowPins,colPins,row,col);


LiquidCrystal lcd(9,A0,10,11,12,13);
char key;
String input_password = "";
int a = 0;
bool locked = false;

void setup()
{
  Serial.begin(9600);
  Serial.println("Setup is running");
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("Hello anik");
  lcd.setCursor(0,1);
  lcd.print("Please Login");
  delay(2000);
  Serial.println("Setup has finished");
}

void loop()
{
  if(!locked)
  {
    lcd.clear();
```

```
      lcd.print("Password: ");
      do {
   key = keypad.getKey();
   if((key != NO_KEY) && (key != '*') && (key != '#'))
   {
      lcd.print('*');
      Serial.print("Typed by user: ");
      Serial.println(key);
      input_password += key;
   }
  } while((key == NO_KEY) || ( (key != '*') && (key != '#') ));

  if(input_password == PASSWD)
  {
   lcd.clear();
   lcd.print("Login Sucessful");
   Serial.println("\nACCESS GRANTED");
  }
  else
  {
   lcd.clear();
   lcd.print("Login Failed");
   Serial.println("\nACCESS DENIED!");
   a++;
  }
  if(a >= 3)
```

```
    {
      locked = true;
      lcd.clear();
      lcd.print("blocked");
      Serial.println("LOCKED ACCESS!");
    }

    input_password = "";
    delay(5000);
  }
}
```

# DISTANCE SENSOR



```
#include <LiquidCrystal.h>

const int RED = 3;

const int BLUE = 4;

const int GREEN = 5;

const int pingPin = 2;


LiquidCrystal LCD(12,13,10,9,8,7);


const int trigPin = 9;

const int echoPin = 10;

const int buzzer = 11;

long duration;

int distance;

int safetyDistance;
```

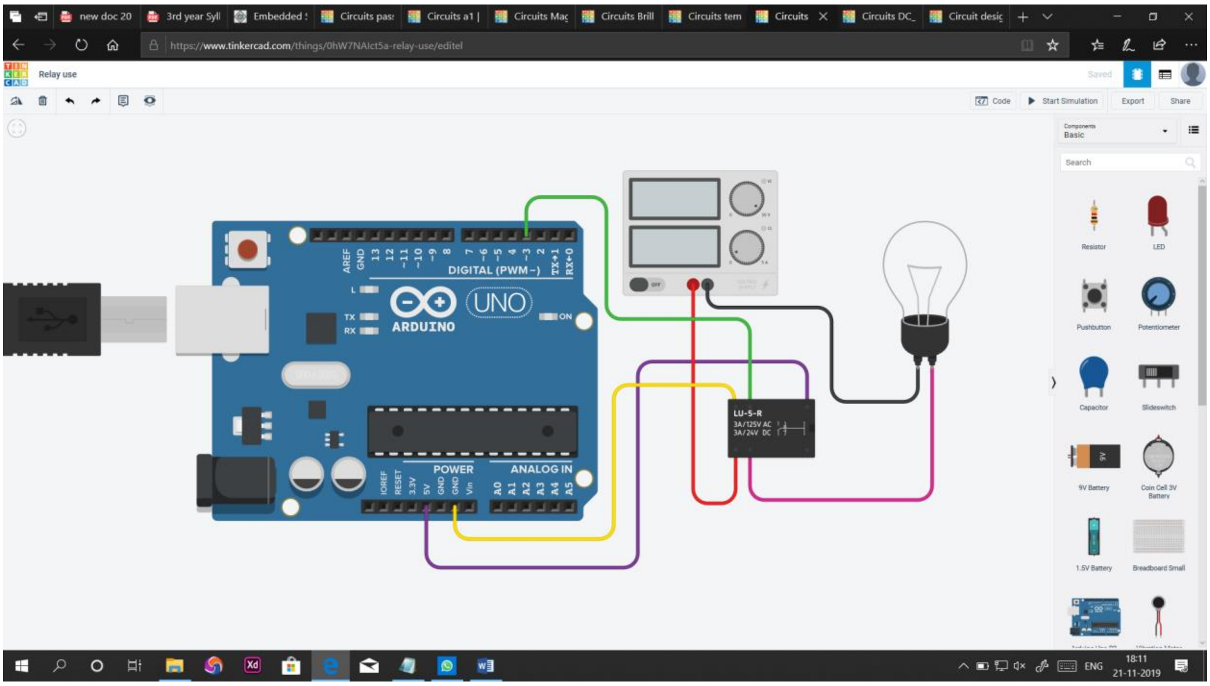
```
void setup()
{
        pinMode(trigPin, OUTPUT);
        pinMode(echoPin, INPUT);
        pinMode(buzzer, OUTPUT);
        Serial.begin(9600);
        pinMode(13, OUTPUT);
        pinMode(RED, OUTPUT);
        pinMode(GREEN, OUTPUT);
        pinMode(BLUE, OUTPUT);

        LCD.begin(16,2);
        LCD.setCursor(0,0);
        LCD.print("Your Distance : ");

}

void loop()
{

        long duration,inches,cm;
        pinMode(pingPin,OUTPUT);
        digitalWrite(pingPin, LOW);
        delayMicroseconds(2);
        digitalWrite(pingPin, HIGH);
        delayMicroseconds(5);
        digitalWrite(pingPin, LOW);


        pinMode(pingPin,INPUT);
        duration = pulseIn(pingPin, HIGH);


        inches = microsecondsToInches(duration);
        cm = microsecondsToCentimeters(duration);
        if (cm>0 && cm<50)
   {
   digitalWrite(RED, HIGH);
   digitalWrite(GREEN, HIGH);
   digitalWrite(BLUE, HIGH);
        digitalWrite(buzzer, HIGH);
        delay(100);


        }
        else if (cm>50 && cm<100)
        {
        digitalWrite(RED, HIGH);
   digitalWrite(GREEN, HIGH);
   digitalWrite(BLUE, HIGH);
        delay(100);
        }
        else if (cm>100 && cm<150)
        {
        digitalWrite(RED, HIGH);
```

```
digitalWrite(GREEN, HIGH);
digitalWrite(BLUE, HIGH);
        delay(100);
        }
        else if (cm>150 && cm<200)
        {
        digitalWrite(RED, HIGH);
digitalWrite(GREEN, HIGH);
digitalWrite(BLUE, HIGH);
        delay(100);
        }
        else if (cm>200 && cm<250)
        {
        digitalWrite(RED, HIGH);
digitalWrite(GREEN, HIGH);
digitalWrite(BLUE, HIGH);
        delay(100);
        }
        else if (cm>250 && cm<300)
        {
        digitalWrite(RED, HIGH);
digitalWrite(GREEN, HIGH);
digitalWrite(BLUE, HIGH);
        delay(100);
        }
        else if (cm>300 && cm<350)
        {
        digitalWrite(RED, HIGH);
   digitalWrite(GREEN, HIGH);
   digitalWrite(BLUE, HIGH);
        delay(100);
        }

        else
   {
   digitalWrite(RED, HIGH);
   digitalWrite(GREEN, HIGH);
   digitalWrite(BLUE, HIGH);
        delay(100);

        }
        Serial.print(inches);
        Serial.print("in, ");
        Serial.print(cm);
        LCD.setCursor(0,1);
        LCD.print(cm);
        Serial.print("cm");
        LCD.print("cm");
        Serial.println();
        delay(100);
        }
long microsecondsToInches(long microseconds)
```

```
   {
   return microseconds / 74 / 2;
   }
long microsecondsToCentimeters(long microseconds)
   {
   return microseconds / 29 / 2;
   }

void showSpectrum()
{
        for(int x=0; x <= 767;x++)
        {
                RGB(x);
                delay(10);
        }

}

void RGB(int color)
{
        int redIntensity;
        int greenIntensity;
        int blueIntensity;

        color = constrain(color,0,767);
```

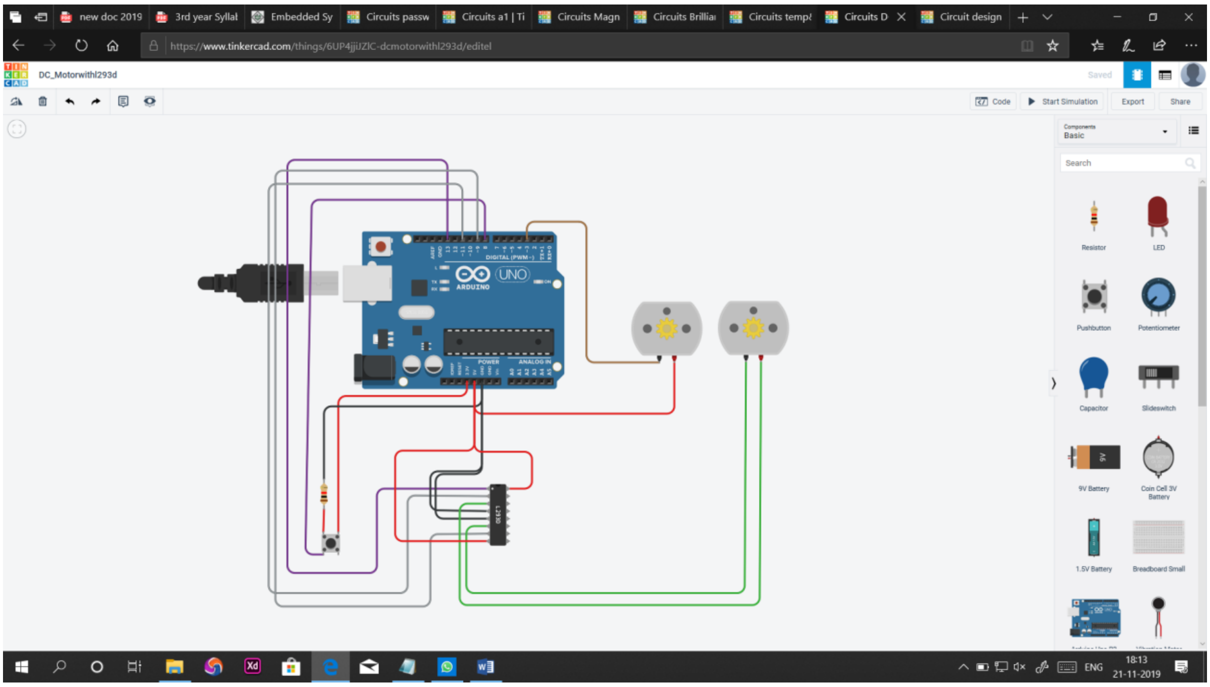
```
	if(color<=255)
	{
		redIntensity = 255 - color;
		greenIntensity = color;
		blueIntensity = 0;
	}

	else if(color<=511)
	{
		redIntensity = 0;
		greenIntensity = 511 - color;
		blueIntensity = color - 256;
	}
	else
	{
		redIntensity = color - 512;
		greenIntensity = 0;
		blueIntensity = 767 - color;
	}
	analogWrite(RED,redIntensity);
	analogWrite(GREEN,greenIntensity);
	analogWrite(BLUE,blueIntensity);
}
```

# Relay



```
void setup()
{
  pinMode(3, OUTPUT);
}


void loop()
{
  digitalWrite(3, HIGH);
  delay(100); // Wait for 1000 millisecond(s)
  digitalWrite(3, LOW);
  delay(100); // Wait for 1000 millisecond(s)
}
```

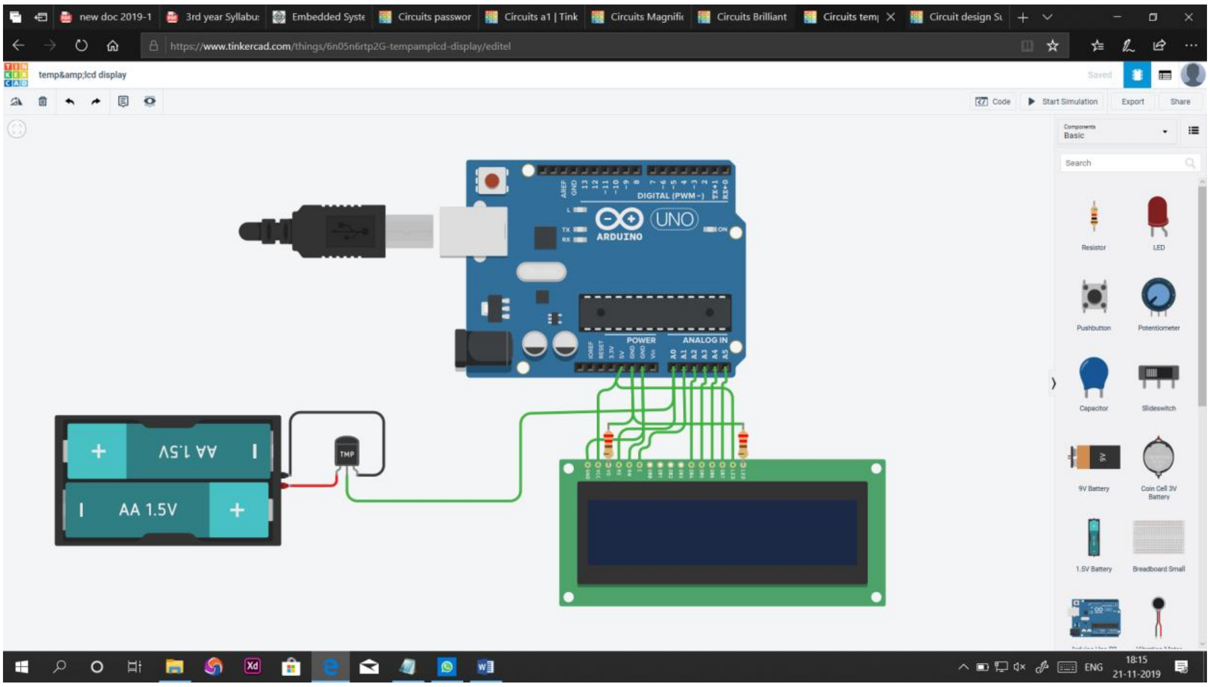
# DC moter normal and with L296



```
int boton = 8;

int motor = 3;

void setup()

{

        pinMode(boton , INPUT);

        pinMode(motor , OUTPUT);


        pinMode(13, OUTPUT);

        pinMode(11, OUTPUT);

        pinMode(9, OUTPUT);

        digitalWrite(13, HIGH);//set this pin to HIGH to enable motor driver

}

void loop()

{

  int estadoBoton = digitalRead(boton);

  if (estadoBoton == 1)

  {

  digitalWrite(motor , LOW);
```

```
 }
 else
 {
 digitalWrite(motor , HIGH);
 }




 digitalWrite(11, HIGH);
 digitalWrite(9, LOW);
 //Use following block of code to run DC motor in anti clockwise
 //Use only if it is required
 delay(2000);
 digitalWrite(11, LOW);
 digitalWrite(9, HIGH);
 delay(2000);
}
```

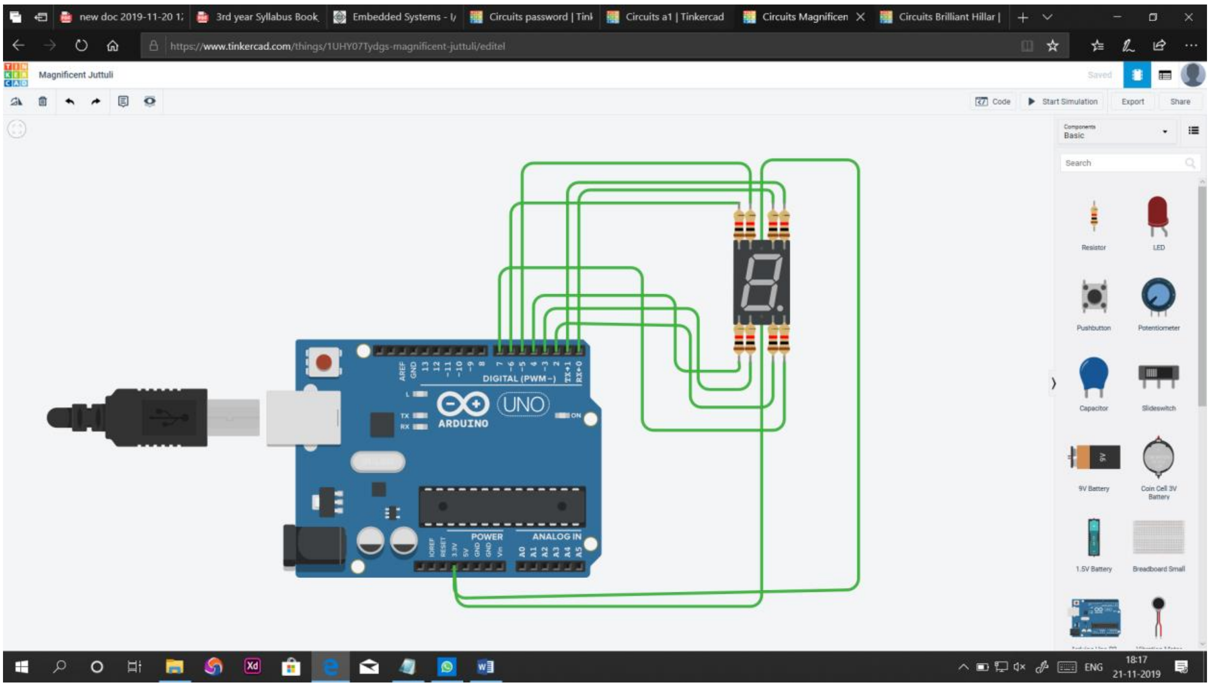
# Temp with lcd



```
#include <LiquidCrystal.h>

int sensorPin = 0;

int tempC, tempF;

LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);

int get_temperature(int pin) {

// We need to tell the function which pin the sensor is hooked up to. We're using

// the variable pin for that above

// Read the value on that pin

int temperature = analogRead(pin);

// Calculate the temperature based on the reading and send that value back

float voltage = temperature * 5.0;

voltage = voltage / 1024.0;

return ((voltage - 0.5) * 100);

}
```

```
int celsius_to_fahrenheit(int temp) {
return (temp * 9 / 5) + 32;
}
//Digital pins to which you connect the LCD
const int inPin = 0;                  // A0 is where you connect the sensor
void setup()
{
  lcd.begin(16,2);
}
void loop()
{
        tempC = get_temperature(sensorPin);
        tempF = celsius_to_fahrenheit(tempC);
        lcd.setCursor(0,0);
        lcd.print(tempF); lcd.print(" "); lcd.print((char)223); lcd.print("F");
        delay(200);

  int value = analogRead(inPin); // read the value from the sensor
  lcd.setCursor(0,1);
  float millivolts = (value / 1024.0) * 5000;
  float celsius = millivolts / 10;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(celsius);
  lcd.print("C");
  lcd.setCursor(0,1);
  lcd.print((celsius * 9)/5 + 32); //turning the celsius into fahrehait
  lcd.print("F");
  delay(1000);
}
```

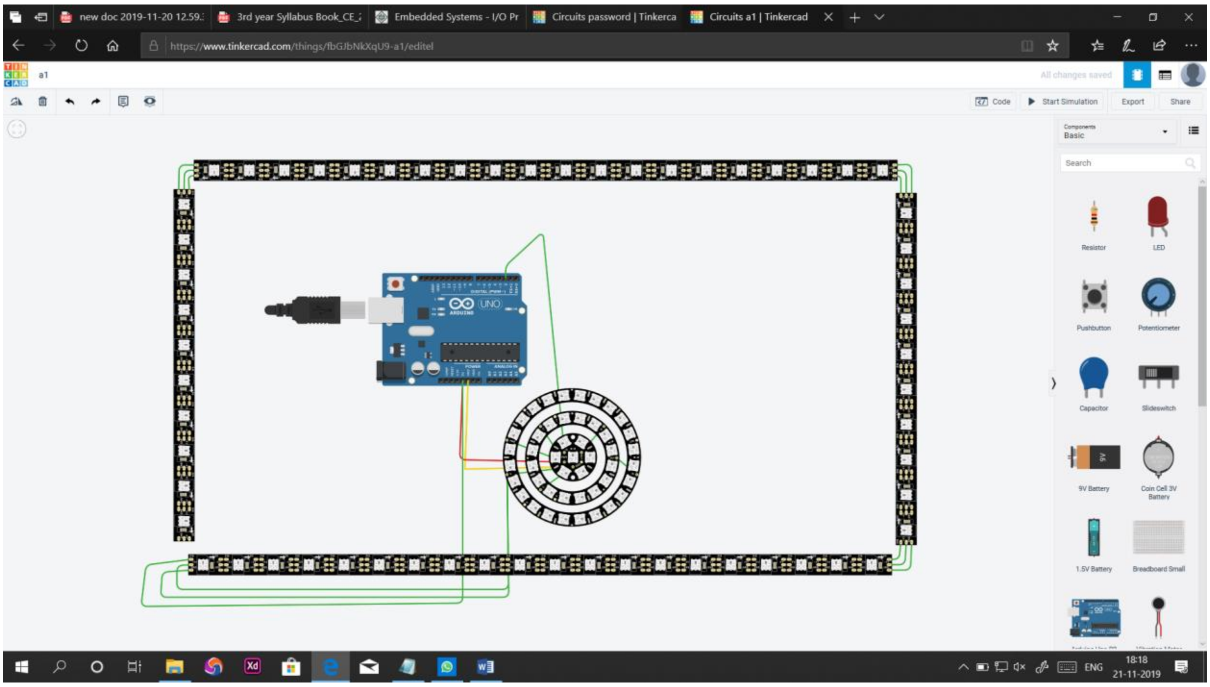
# 7 segment



```
void Display_Segment(int);

int digit[10][7] = {{1,0,0,0,0,0,0},    // Digit "0"

                   { 1,1,1,1,0,0,1},    // Digit "1"

                   { 0,1,0,0,1,0,0},    // Digit "2"

                   { 0,1,1,0,0,0,0},    // Digit "3"

                   { 0,0,1,1,0,0,1},    // Digit "4"

                    { 0,0,1,0,0,1,0},    // Digit "5"

                    { 0,0,0,0,0,1,0},    // Digit "6"

                    { 1,0,1,1,0,0,0},    // Digit "7"

                    { 0,0,0,0,0,0,0},    // Digit "8"

                    { 0,0,1,0,0,0,0 }};  // Digit "9"

void setup()

{

   for(int a=0 ; a<=6; a++){

   pinMode(a, OUTPUT);
```

```
    }
 }


void loop()
{
  for (int value = 0; value<=9; value++)
  {
   delay(1000);
   Display_Segment(value);   }
  delay(1500);
}
void Display_Segment(int value)
{
  int startPin= 0;
  for (int x=6; x >= 0; x--) {
   digitalWrite(startPin, digit[value][x]);
   startPin++;



 }
}
```

# NeoPixel

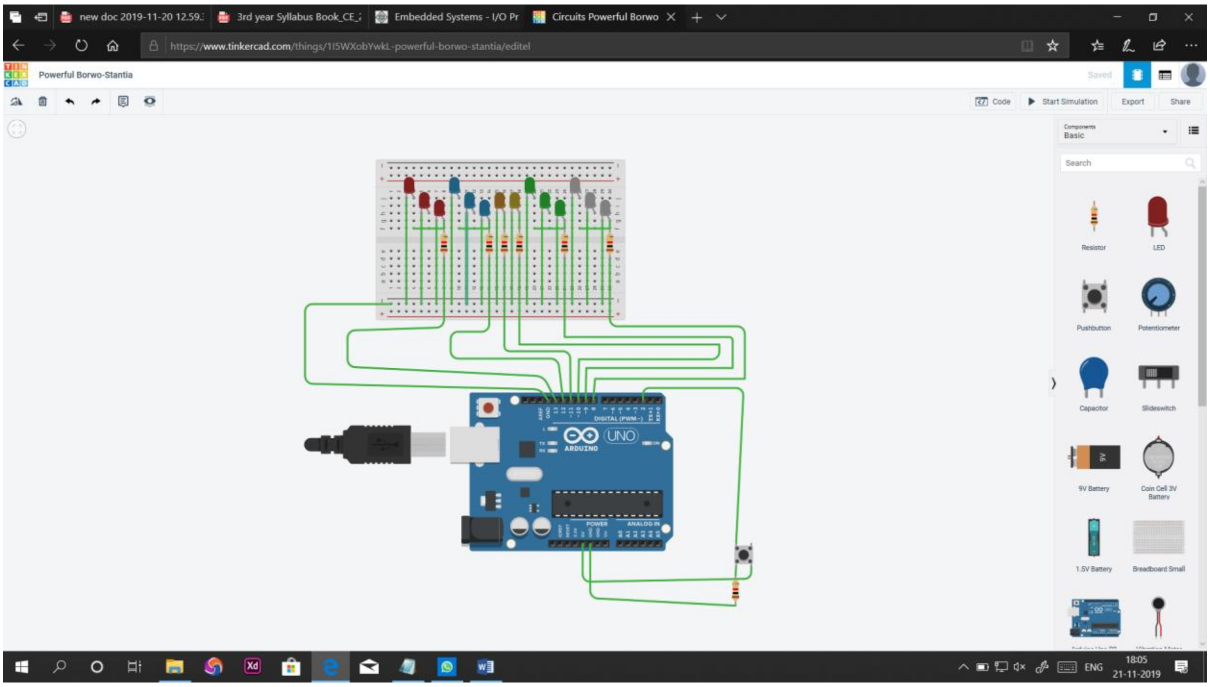


```
#include<Adafruit_NeoPixel.h>


#define PIN 2


#define NUMPIXELS 107


Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);




int i;


void setup()
```

```
{



 pixels.begin();
 pixels.show();


}


void loop()
{


 for( i=0;i<108;i++)
 {
 pixels.setPixelColor(i,random(0,255),random(0,255),random(0,255));
   pixels.show();
        delay(0.0000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000001);
```

```
     pixels.setPixelColor(i,0,0,0);
    pixels.show();


  }
}
```

# Led light



```
int buttonState = 0;

int counter = 0;

int lastbuttonstate = 0;

void setup()

{


  pinMode(13, OUTPUT);

  pinMode(12, OUTPUT);

  pinMode(11, OUTPUT);

  pinMode(10, OUTPUT);

  pinMode(9, OUTPUT);

  pinMode(8, OUTPUT);



  pinMode(2,INPUT);
```

```
}


void loop()
{
  buttonState = digitalRead(2);


  if(buttonState == HIGH && counter==0)
    {



  digitalWrite(13, HIGH);
 delay(250); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(250); // Wait for 1000 millisecond(s)

  digitalWrite(12, HIGH);
  delay(250); // Wait for 1000 millisecond(s)
  digitalWrite(12, LOW);
  delay(250); // Wait for 1000 millisecond(s)

  digitalWrite(11, HIGH);
  delay(250); // Wait for 1000 millisecond(s)
  digitalWrite(11, LOW);
  delay(250); // Wait for 1000 millisecond(s)

  digitalWrite(10, HIGH);
  delay(250); // Wait for 1000 millisecond(s)
  digitalWrite(10, LOW);
  delay(250); // Wait for 1000 millisecond(s)

  digitalWrite(9, HIGH);
```

```
delay(250); // Wait for 1000 millisecond(s)
digitalWrite(9, LOW);
delay(250); // Wait for 1000 millisecond(s)

digitalWrite(8, HIGH);
delay(250); // Wait for 1000 millisecond(s)
digitalWrite(8, LOW);
delay(250); // Wait for 1000 millisecond(s)
            counter++;
}
else if (buttonState == HIGH && counter==1)

{
 digitalWrite(8, HIGH);
delay(250); // Wait for 1000 millisecond(s)
digitalWrite(8, LOW);
delay(250); // Wait for 1000 millisecond(s)

digitalWrite(9, HIGH);
delay(250); // Wait for 1000 millisecond(s)
digitalWrite(9, LOW);
delay(250); // Wait for 1000 millisecond(s)

digitalWrite(10, HIGH);
delay(250); // Wait for 1000 millisecond(s)
digitalWrite(10, LOW);
delay(250); // Wait for 1000 millisecond(s)

digitalWrite(11, HIGH);
delay(250); // Wait for 1000 millisecond(s)
digitalWrite(11, LOW);
```

```
  delay(250); // Wait for 1000 millisecond(s)


  digitalWrite(12, HIGH);
  delay(250); // Wait for 1000 millisecond(s)
  digitalWrite(12, LOW);
  delay(250); // Wait for 1000 millisecond(s)


  digitalWrite(13, HIGH);
  delay(250); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(250); // Wait for 1000 millisecond(s)
  counter=0;
  }


}
```