



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRONICS & COMMUNICATION

FALL SEMESTER 2023 – 2024

ECE – 4003

EMBEDDED SYSTEM DESIGN

Professor: Sundar. S

LAB ASSESSMENT – 4

**SENSORS INTERFACE FOR EMBEDDED
APPLICATIONS**

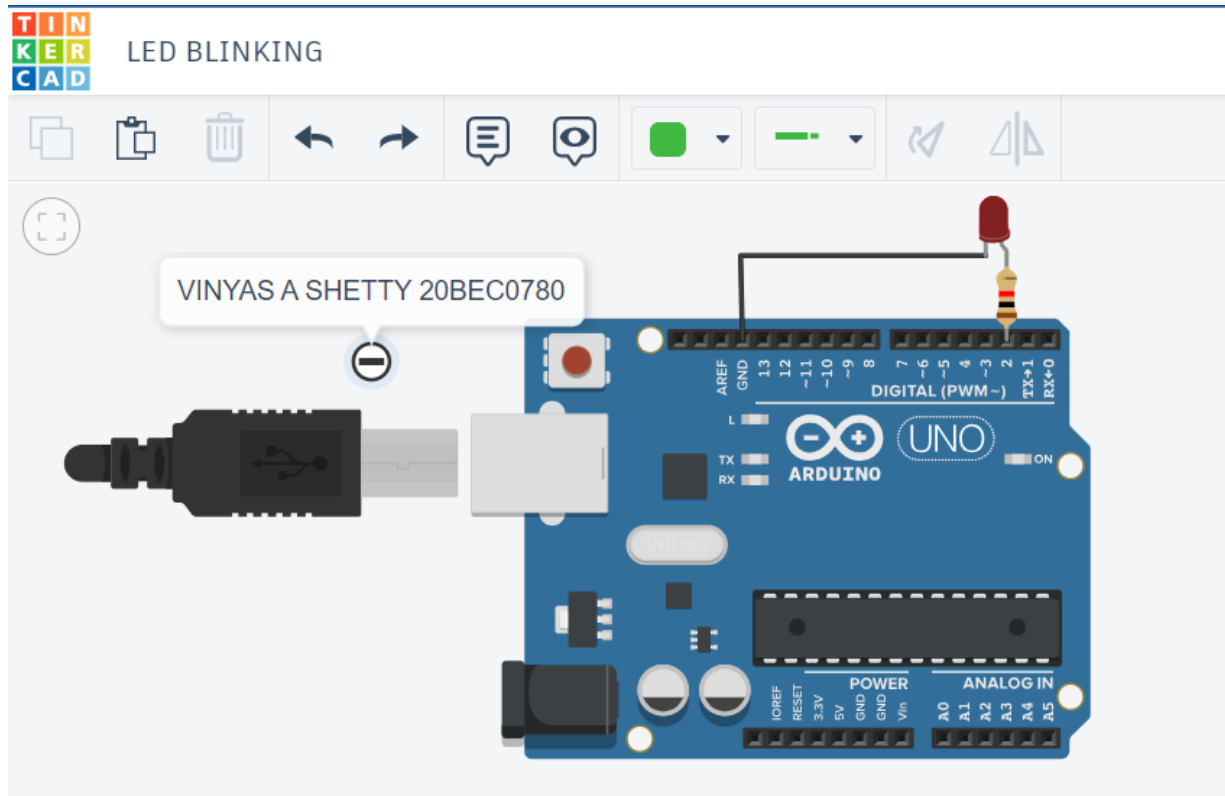
VINYAS A SHETTY

20BEC0780

QUESTION 1A:

Simulate a circuit for LED blinking

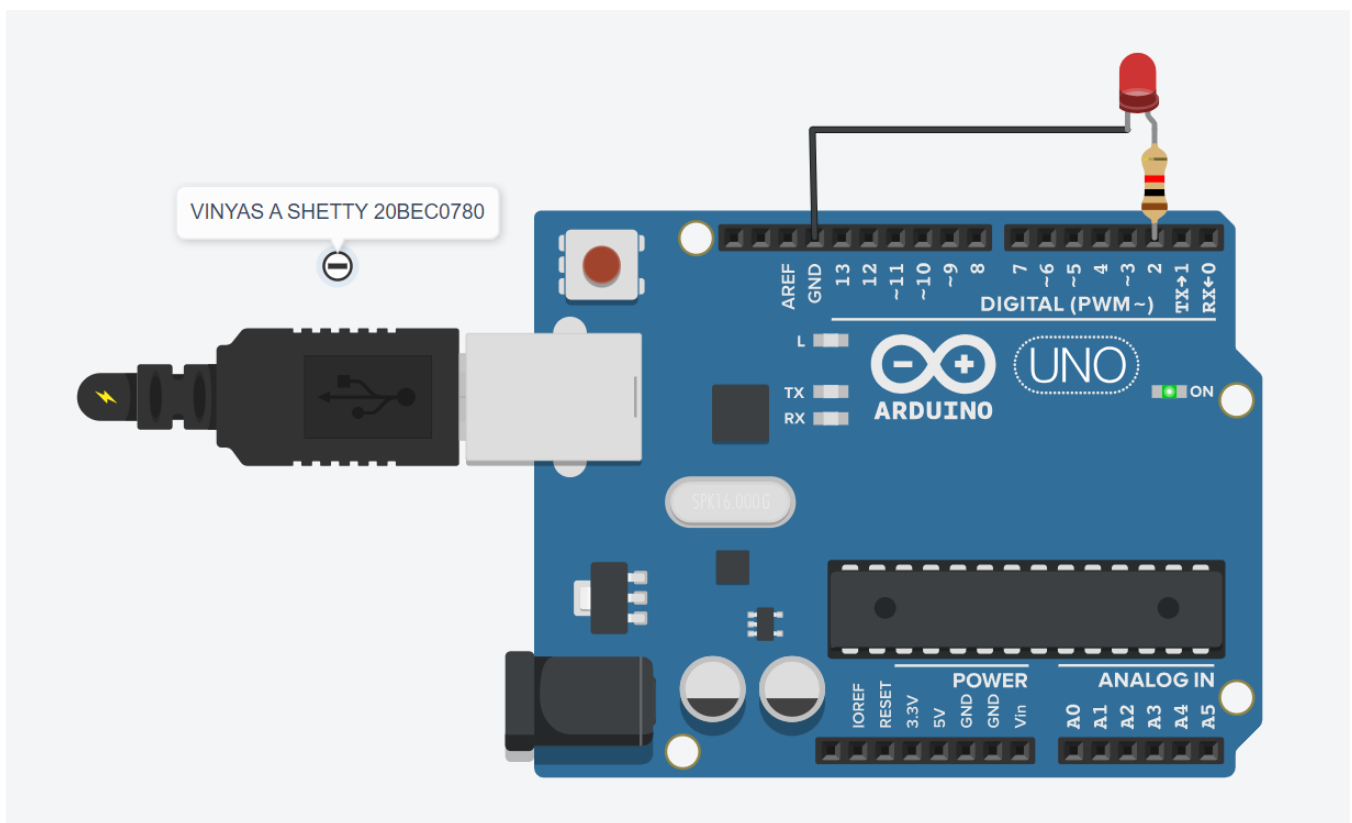
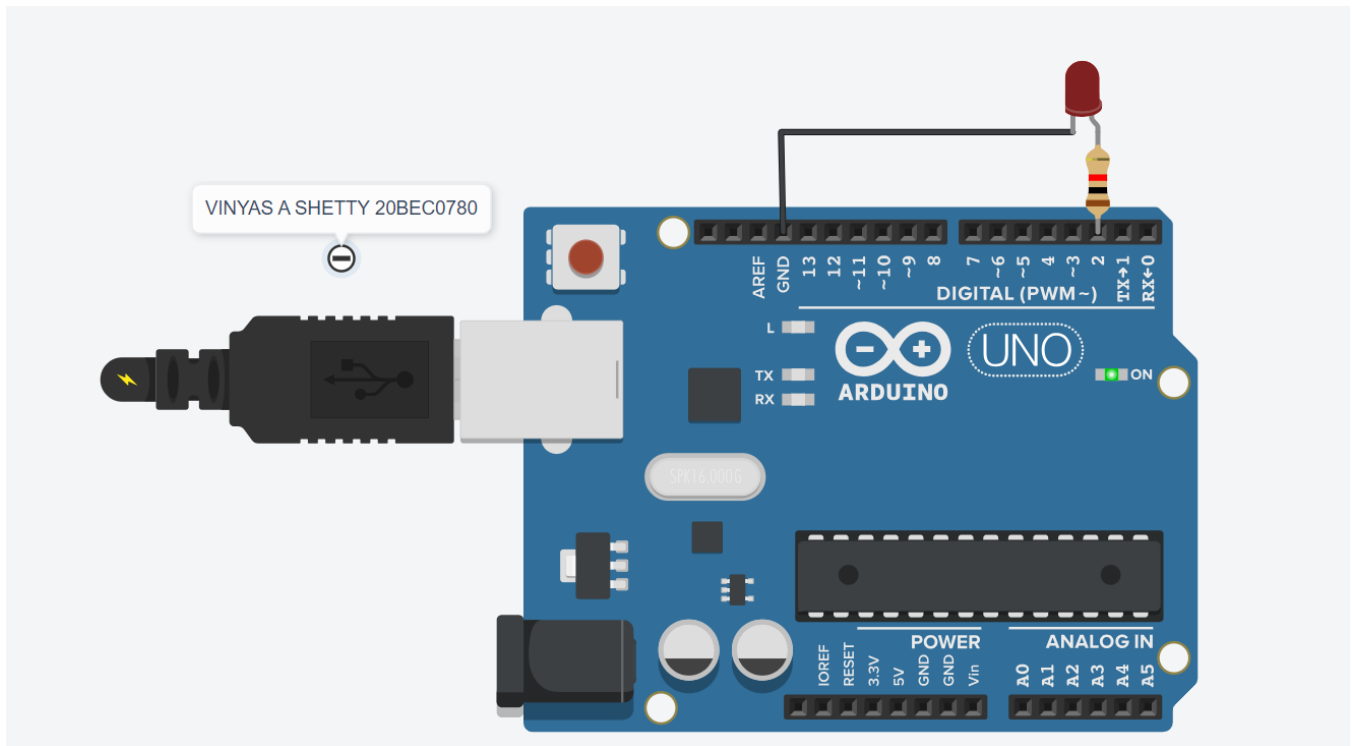
CIRCUIT DIAGRAM:



CODE:

```
void setup()
{
  pinMode(2, OUTPUT);
}
void loop()
{
  digitalWrite(2,HIGH);
  delay(1000);
  digitalWrite(2,LOW);
  delay(1000);
}
```

RESULTS SCREENSHOTS:

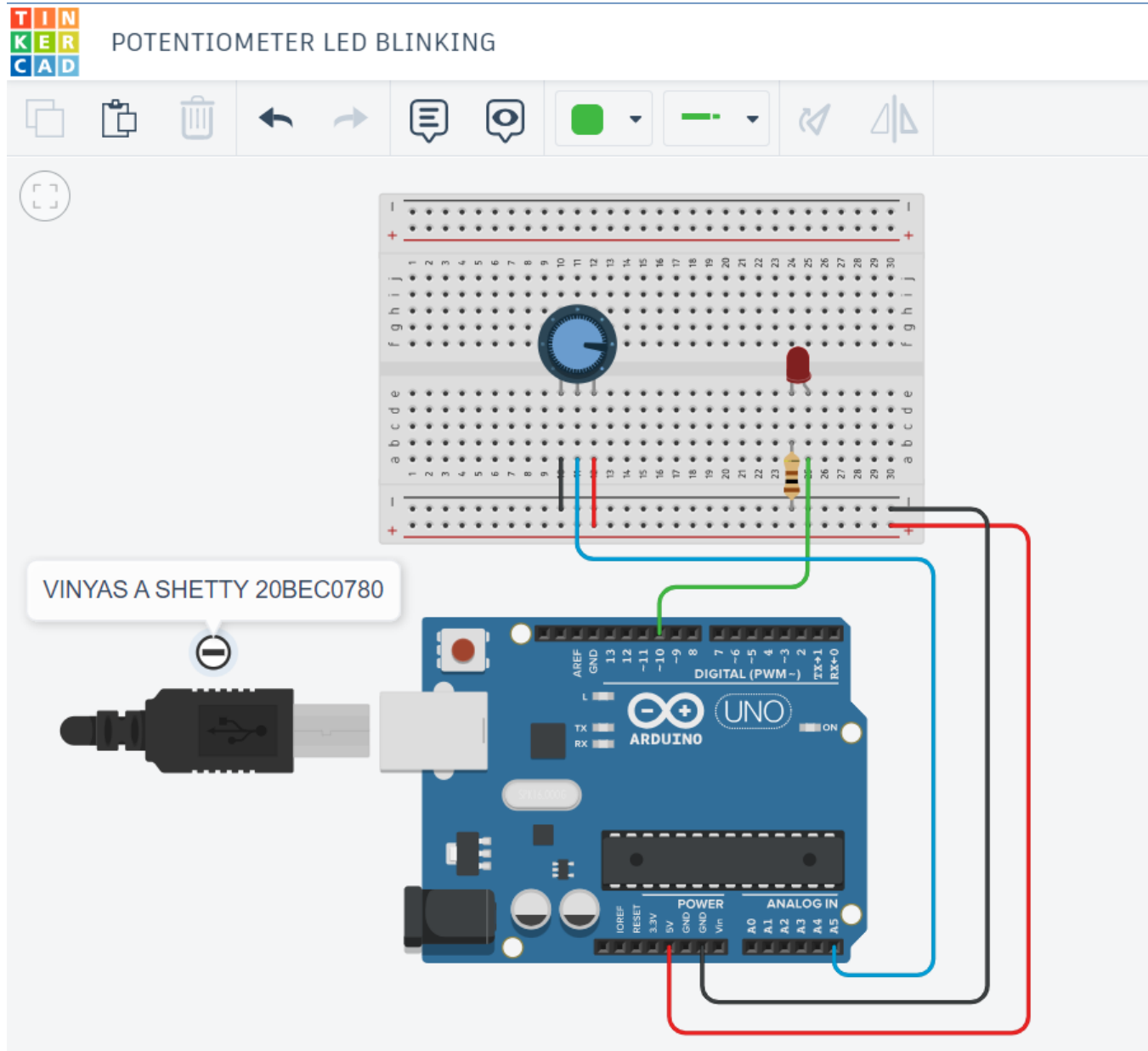


OBSERVATIONS:

We observed that the LED blinking with 1 sec ON and 1 sec OFF.

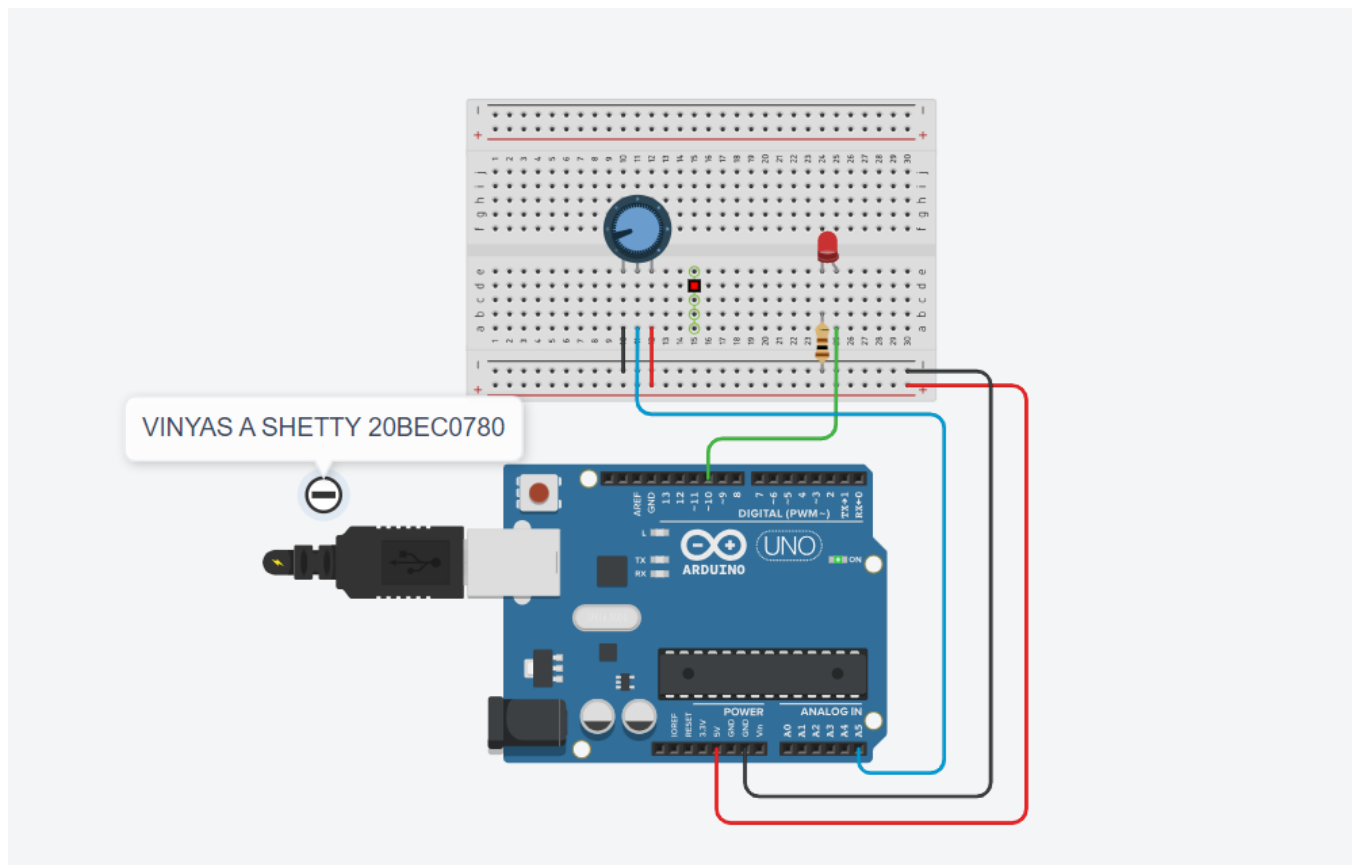
QUESTION 1B:

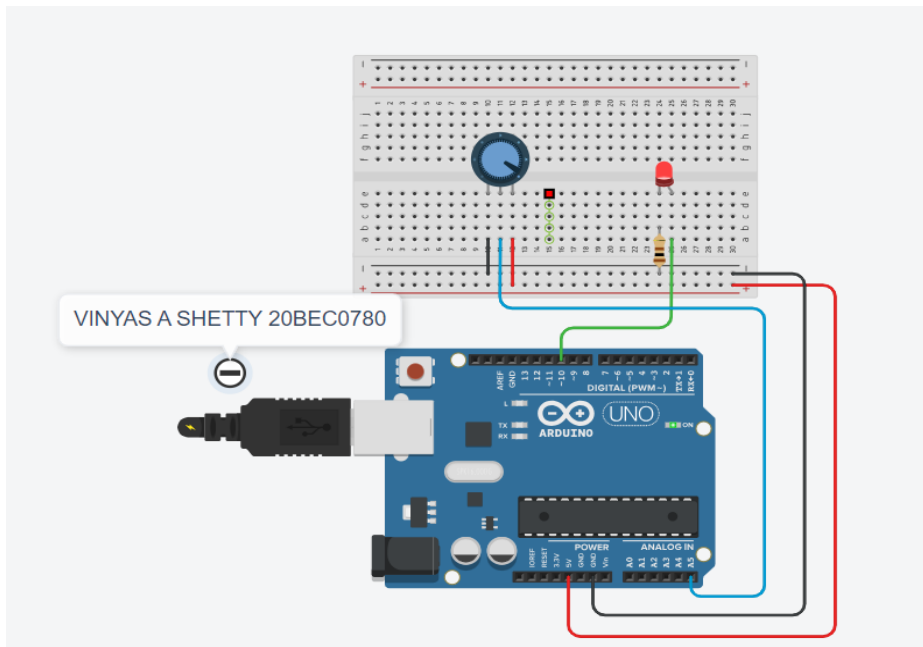
Simulate a circuit for controlling LED Brightness Using Potentiometer

CIRCUIT DIAGRAM:

CODE:

```
int ledpin=10;  
int potentio_meterpin =A5;  
void setup()  
{  
  pinMode(ledpin, OUTPUT);  
}  
void loop()  
{  
  int pot_meter_value = analogRead(potentio_meterpin);  
  int brightness = pot_meter_value / 4;  
  analogWrite(ledpin, brightness);  
}
```

RESULTS SCREENSHOTS:



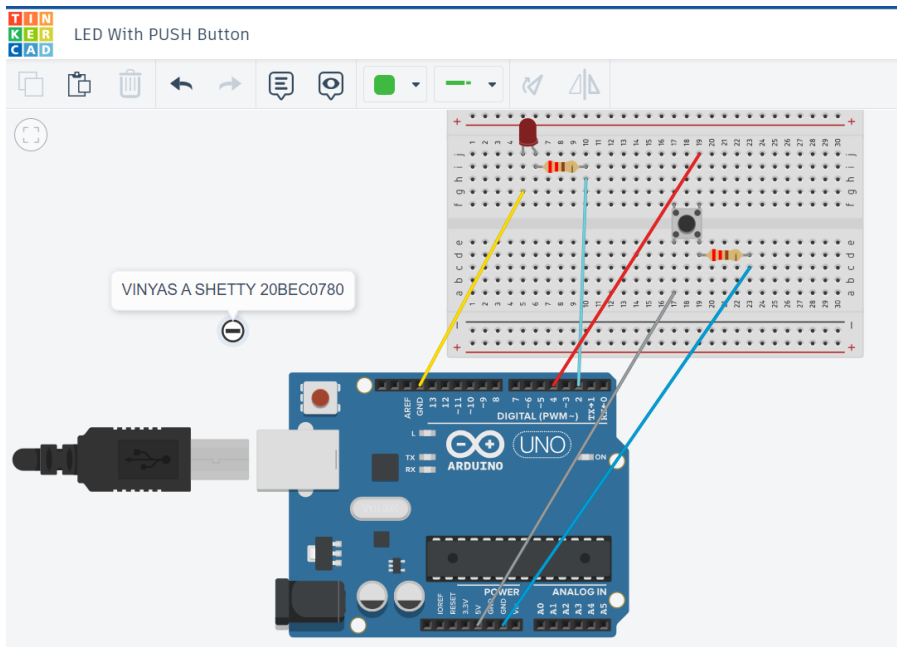
OBSERVATIONS:

We observe that the brightness of LED changes when you rotate the knob of the potentiometer

QUESTION 1C:

Control an LED With PUSH Button

CIRCUIT DIAGRAM:

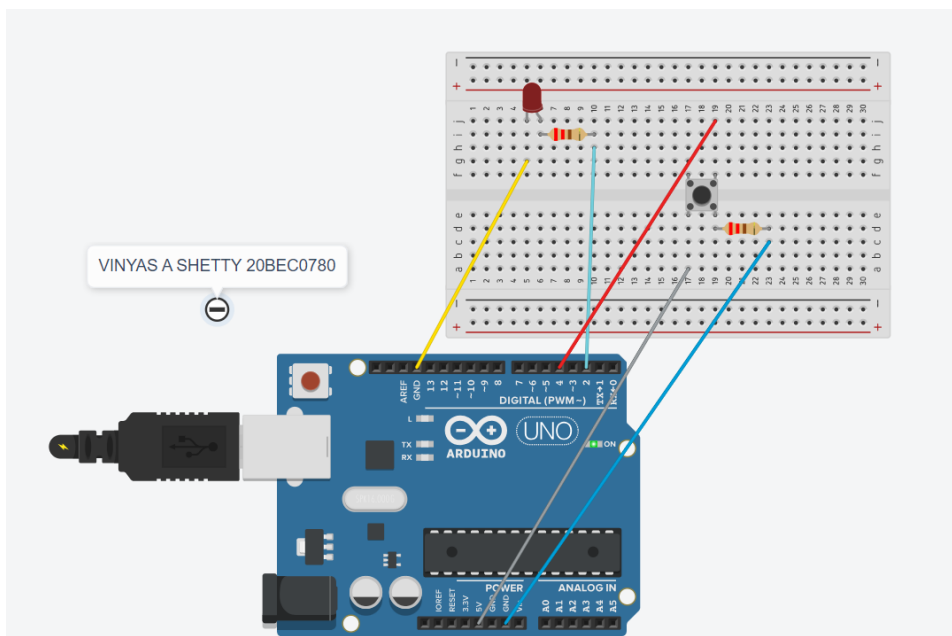


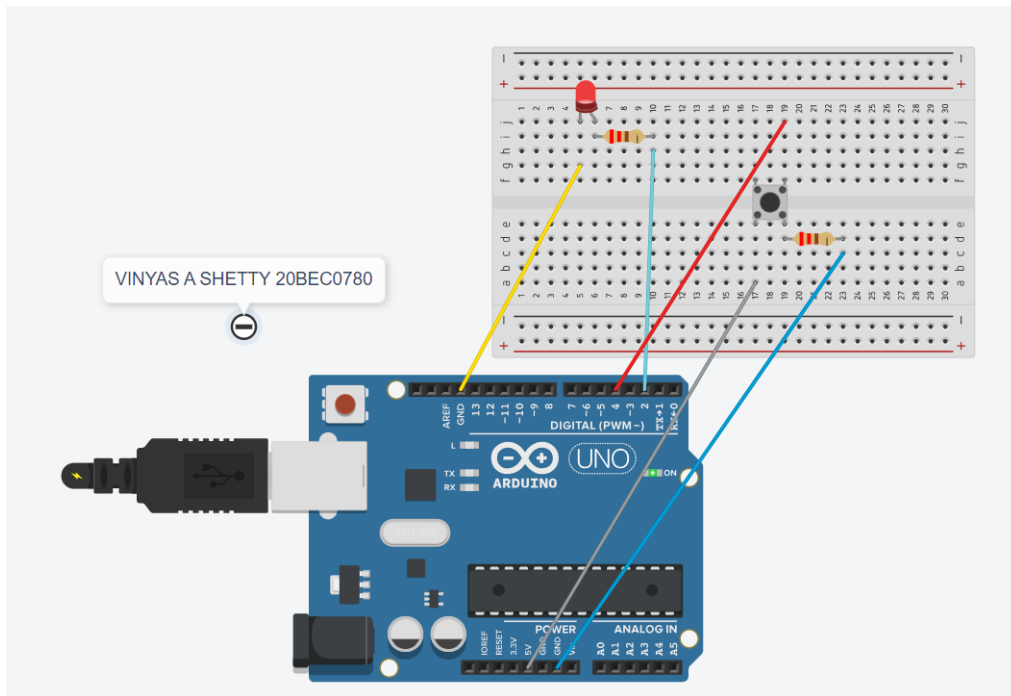
CODE:

```

int led = 2;
int pushButton = 4;
void setup()
{
  pinMode(pushButton, INPUT);
  pinMode(led, OUTPUT);
}
void loop()
{
  int buttonState = digitalRead(pushButton);
  if (buttonState == 1) {
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  }
  else {
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  }
}

```

RESULTS SCREENSHOTS:



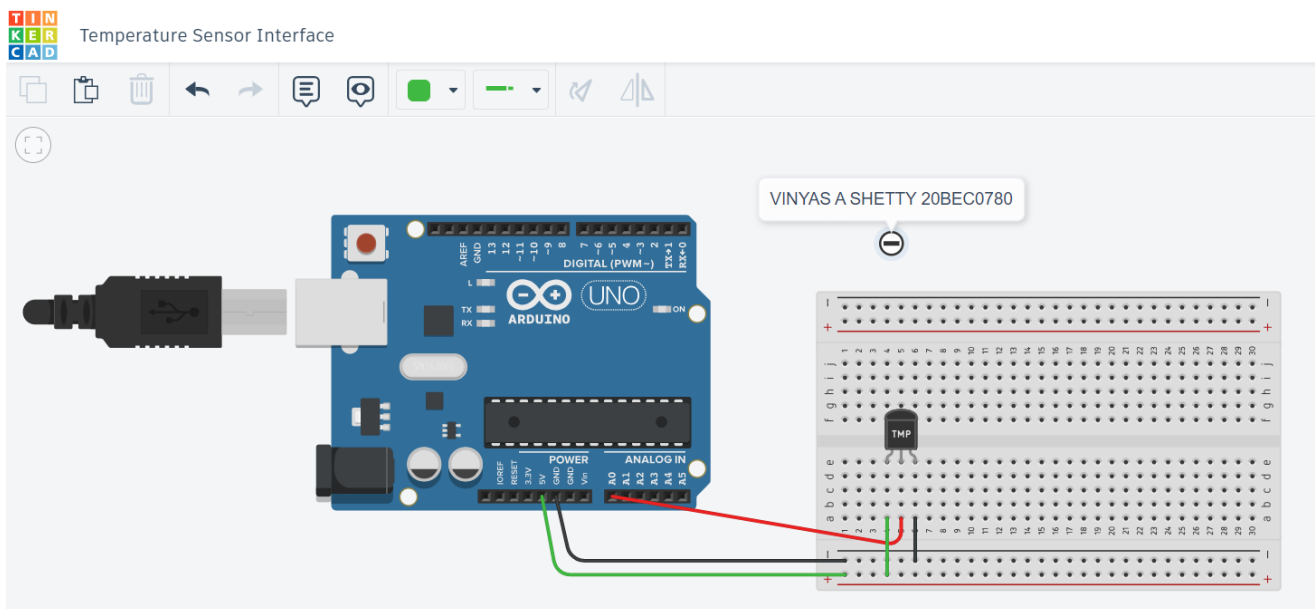
OBSERVATIONS:

We observe the LED switches ON when you press the push button

QUESTION 1D:

Simulate a circuit to understand Temperature Sensor Interface

CIRCUIT DIAGRAM:

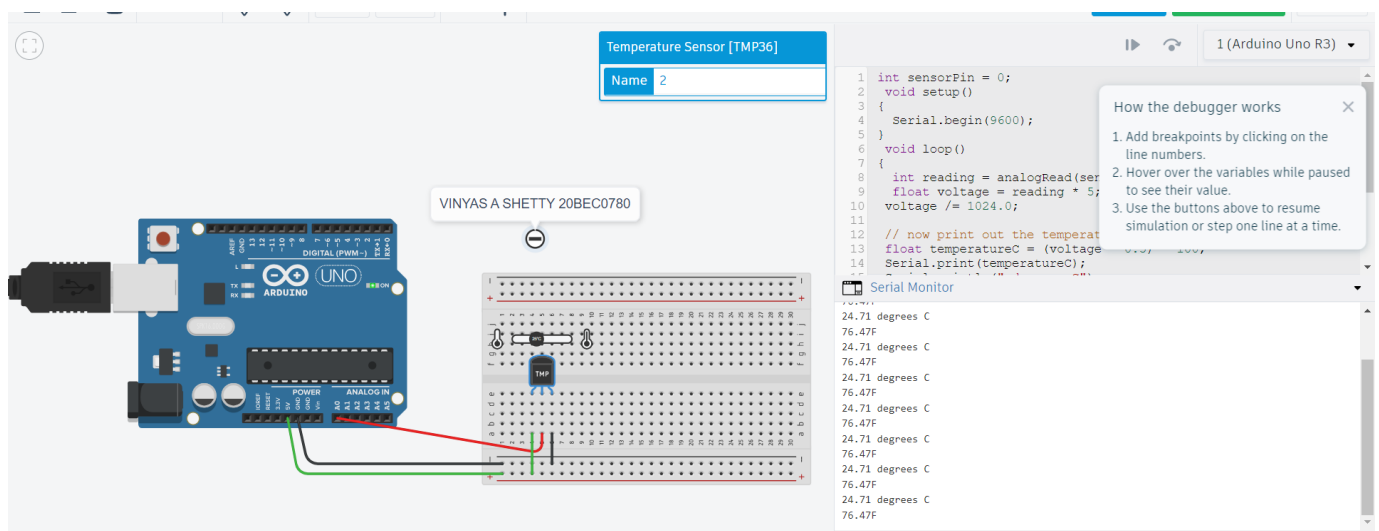


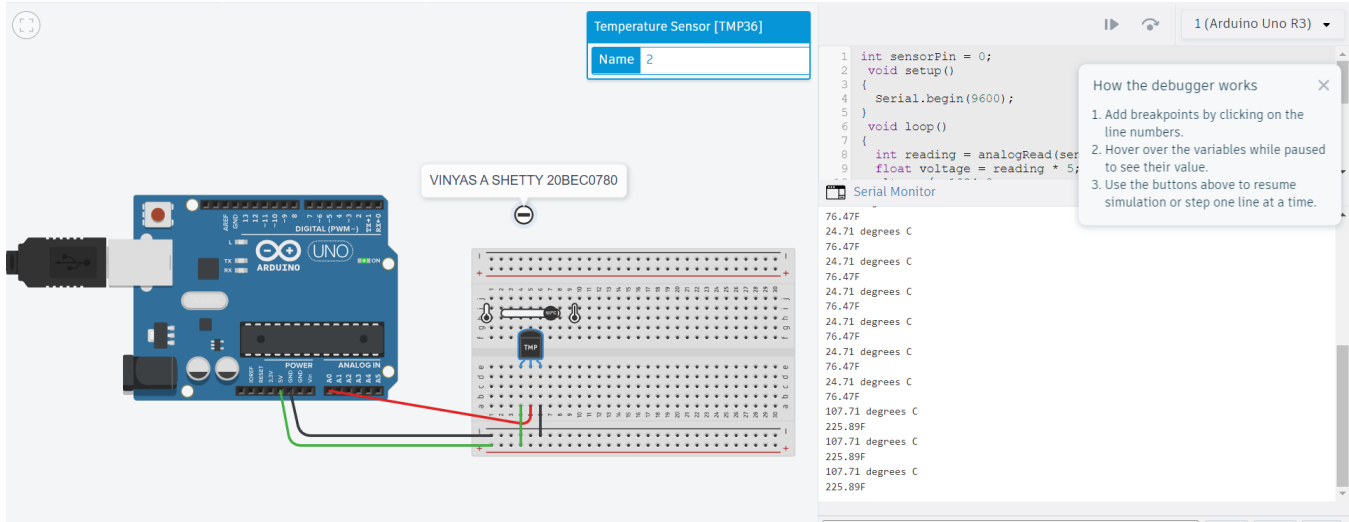
CODE:

```

int sensorPin = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  int reading = analogRead(sensorPin);
  float voltage = reading * 5;
  voltage /= 1024.0;
  float temperatureC = (voltage - 0.5) * 100;
  Serial.print(temperatureC);
  Serial.println(" degrees C");
  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
  Serial.print(temperatureF);
  //Serial.print("\xC2\xB0"); // shows degree symbol
  Serial.println("F");
  delay(2000);
}

```

RESULTS SCREENSHOTS:



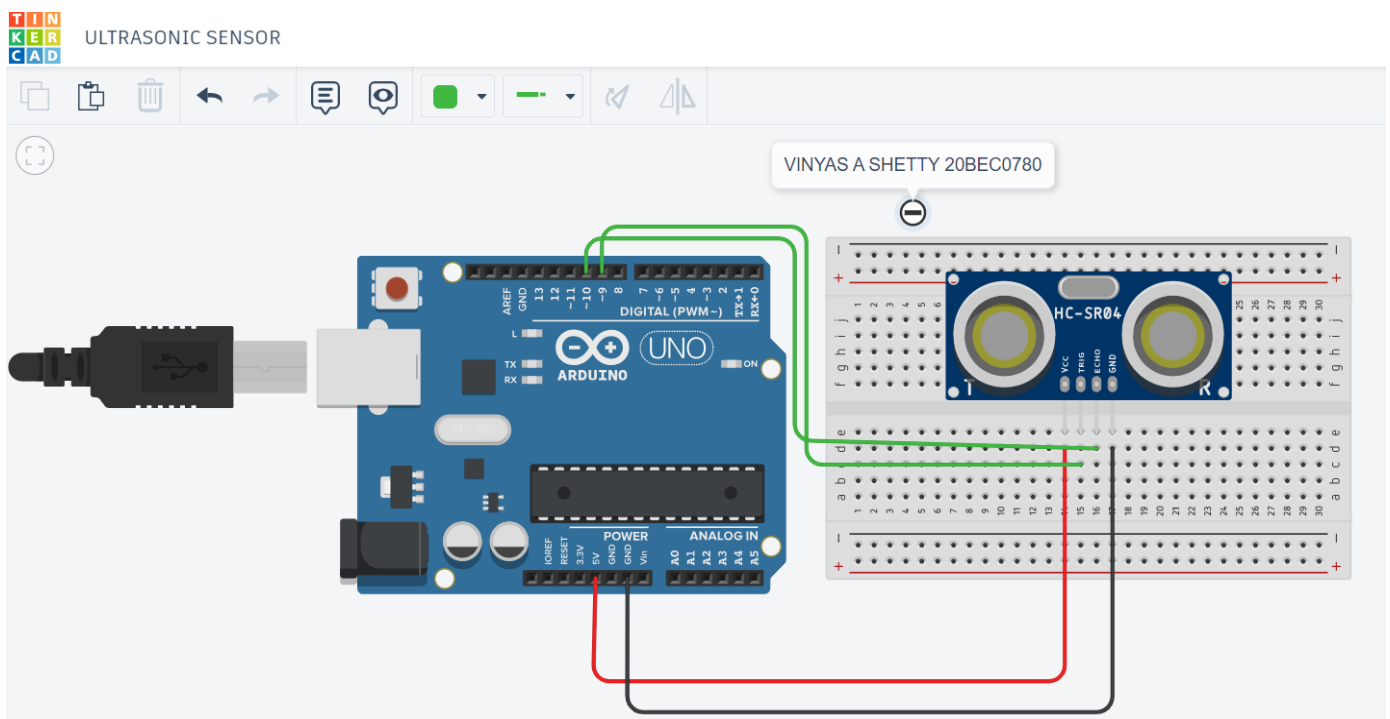
OBSERVATIONS:

We observed that we were able to display the live temperature of the surroundings on the serial monitor.

QUESTION 1E:

Simulate a circuit to understand Ultrasonic Sensor

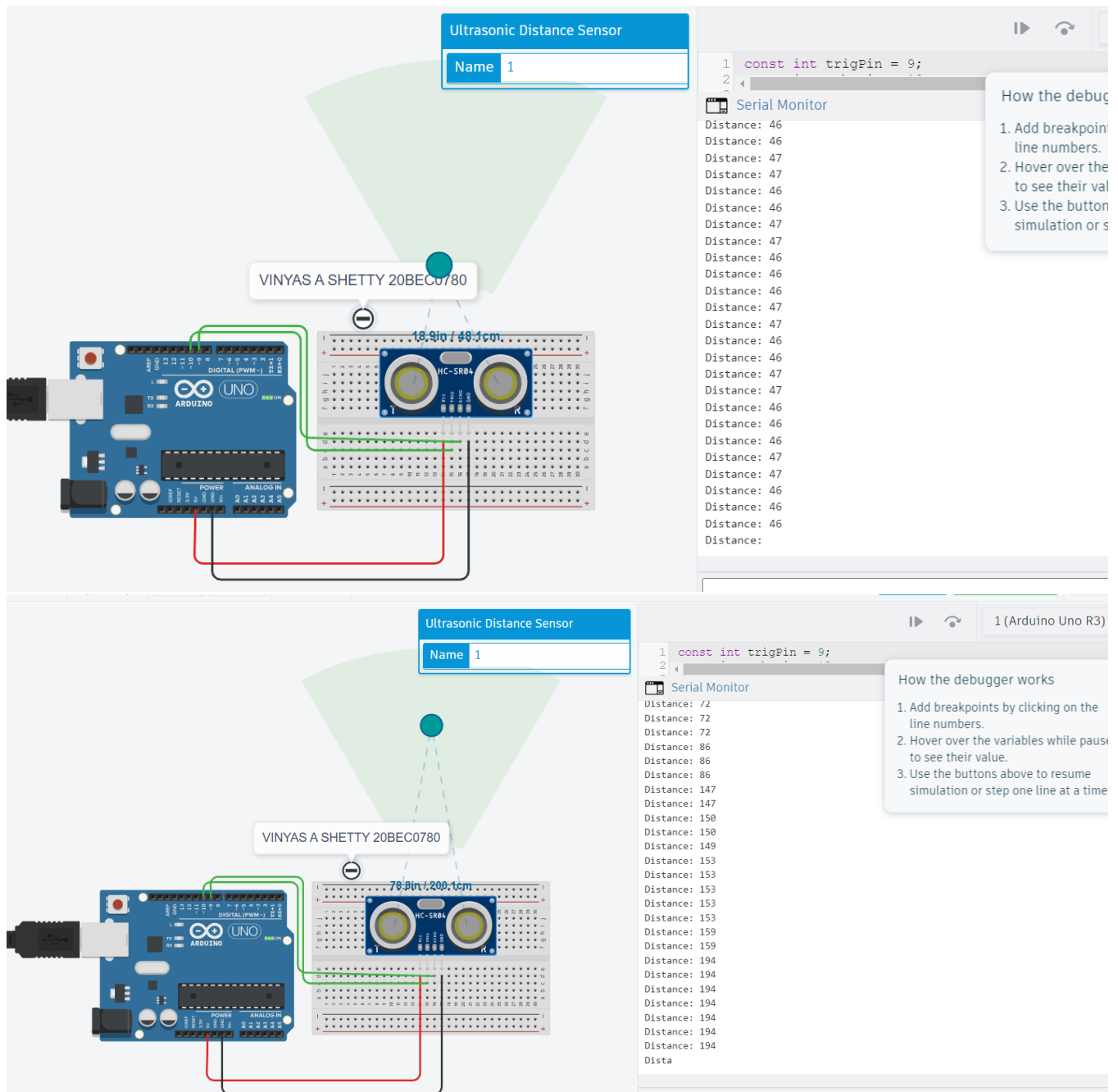
CIRCUIT DIAGRAM:



CODE:

```
const int trigPin = 9;
const int echoPin = 10;
// defines variables
long duration;
int distance;
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}
void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(200);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(100);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}
```

RESULTS SCREENSHOTS:



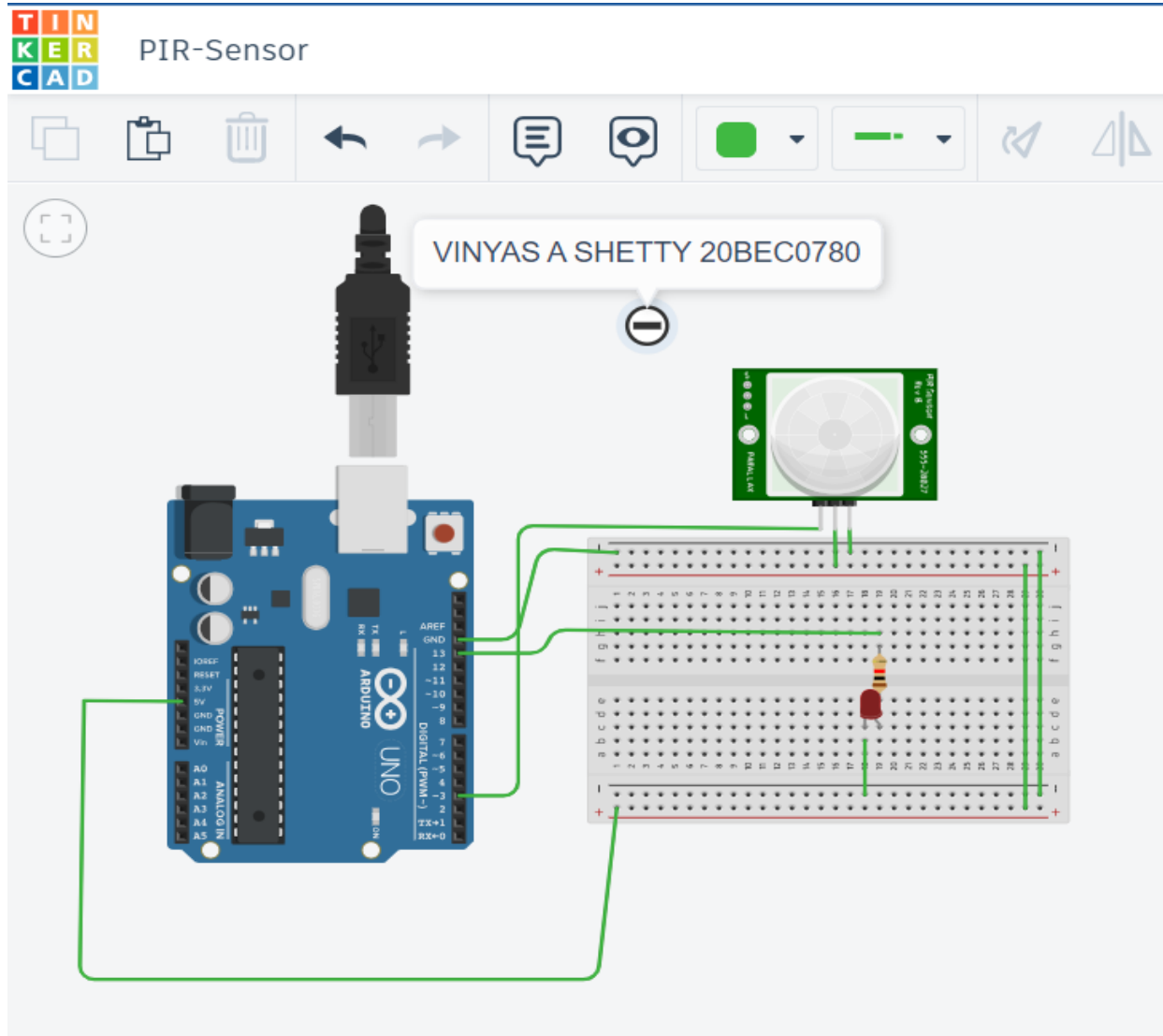
OBSERVATIONS:

We were able to display the distance of an object from the ultrasonic sensor on the serial monitor

QUESTION 1F:

Simulate a circuit to understand PIR Sensor

CIRCUIT DIAGRAM:



CODE:

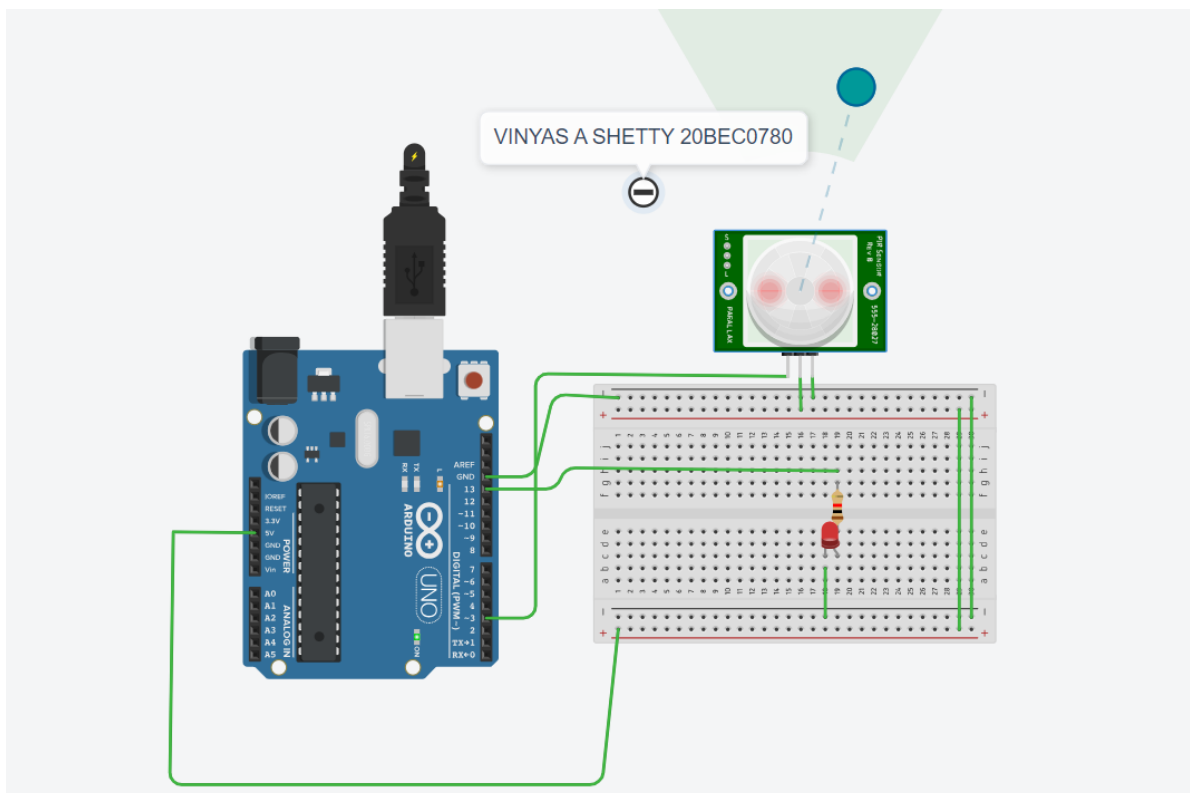
```

int PIR_SENSOR_OUTPUT_PIN = 3;      /* PIR sensor O/P pin */
void setup()
{
  pinMode(PIR_SENSOR_OUTPUT_PIN, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}

```

```
void loop ()  
{  
  int sensor_output;  
  sensor_output = digitalRead(PIR_SENSOR_OUTPUT_PIN);  
  if (sensor_output==HIGH)  
  {  
    digitalWrite(LED_BUILTIN,HIGH);  
  }  
  else  
  {  
    digitalWrite(LED_BUILTIN,LOW);  
  }  
}
```

RESULTS SCREENSHOTS:



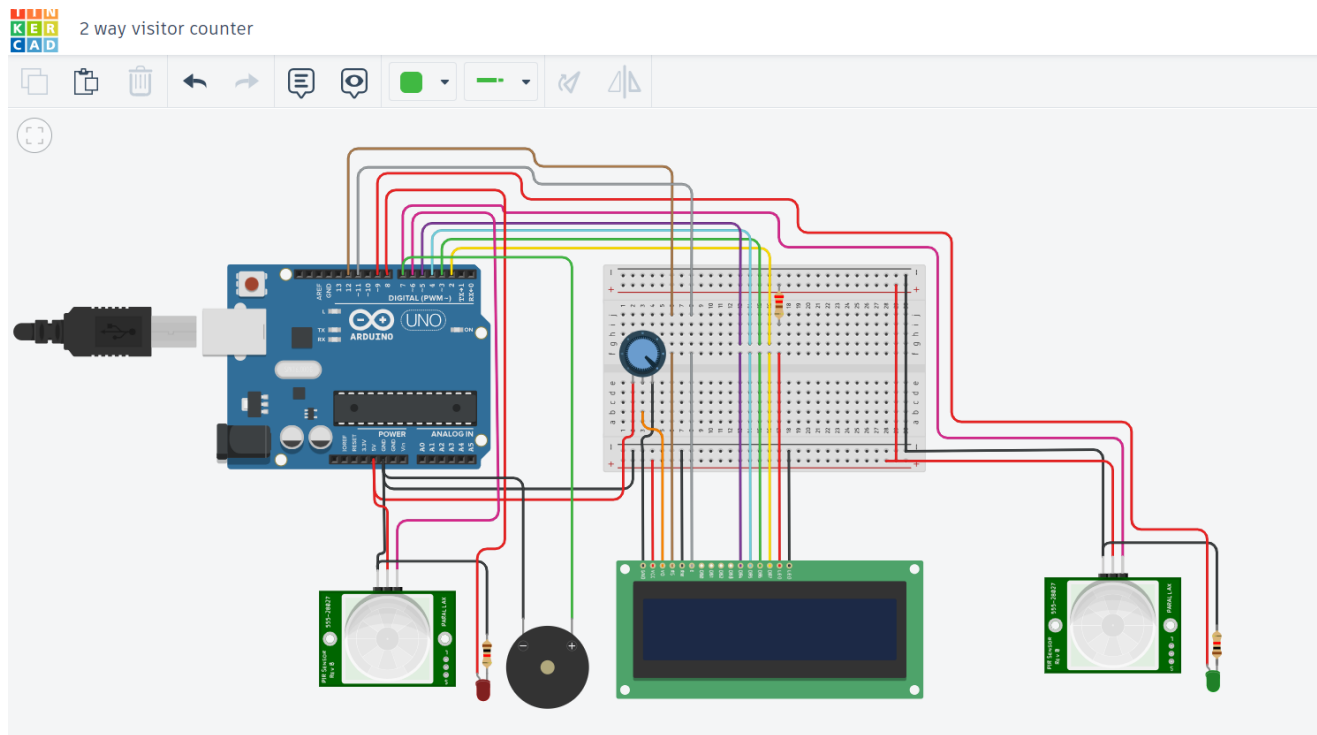
OBSERVATIONS:

We observed that LED switched ON whenever there was movement

QUESTION 2:

Construct a two-way visitor counter

CIRCUIT DIAGRAM:



CODE:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const int pir1=6,pir2=7;
const int red=8,green=9;
int in=0,out=0;
void setup()
{
  lcd.begin(16, 2);
  lcd.print(" Welcome To ");
  lcd.setCursor(2,1);
  lcd.print(" 2 WAY VISITOR COUNTER ");
  delay(1000);
  pinMode(pir1, INPUT);
```

```
pinMode(pir2, INPUT);
pinMode(green, OUTPUT);
pinMode(red, OUTPUT);
lcd.begin(16, 2);
}
void loop()
{
  if (digitalRead(pir1))
  {
    digitalWrite(red,HIGH);
    in = in+1;
    delay(1000);
  }
  else
    digitalWrite(red,LOW);

  if (digitalRead(pir2))
  {
    digitalWrite(green,HIGH);
    out = out+1;
    delay(1000);
  }

  else
    digitalWrite(green,LOW);

  lcd.clear();

  lcd.setCursor(1,0); //position for main word
  lcd.print("Visitors");

  //lcd.setCursor(5,0);
  lcd.setCursor(0,1); //in position
```



```

lcd.print("IN: ");

lcd.setCursor(3,1);
lcd.print(in); //number position

lcd.setCursor(5,1); //out position
lcd.print("OUT:");

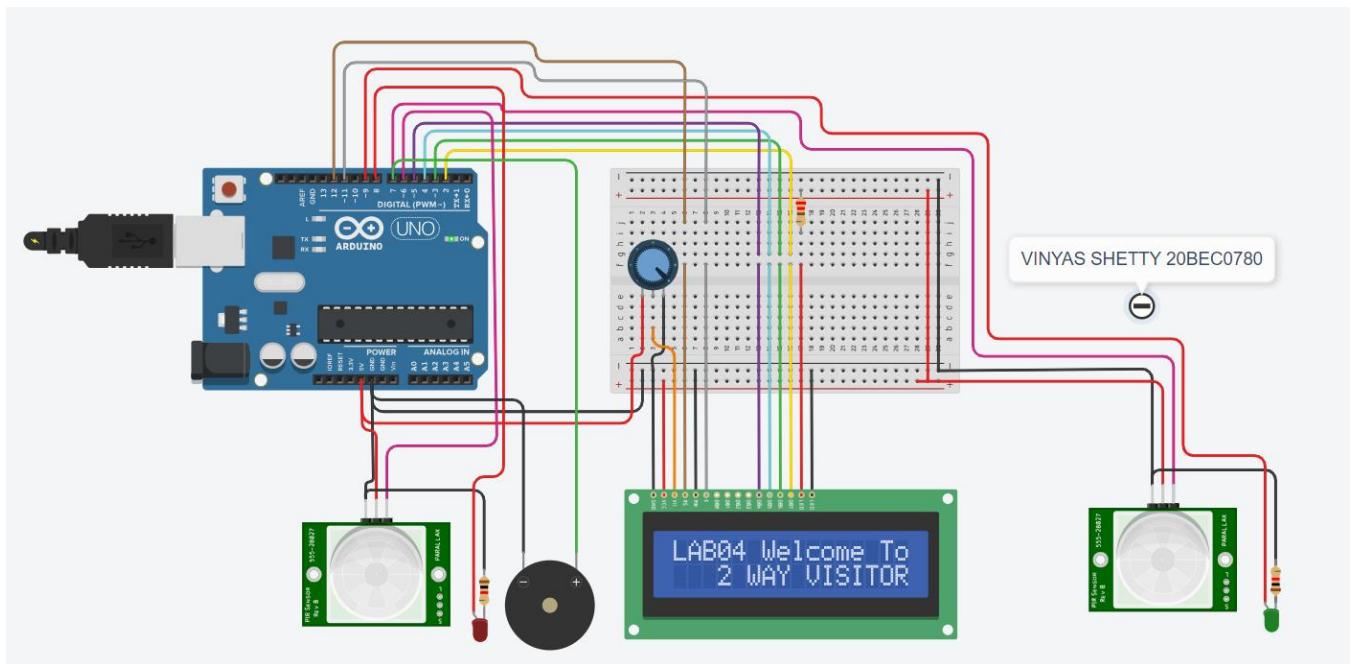
lcd.setCursor(9,1); //number position
lcd.print(out);

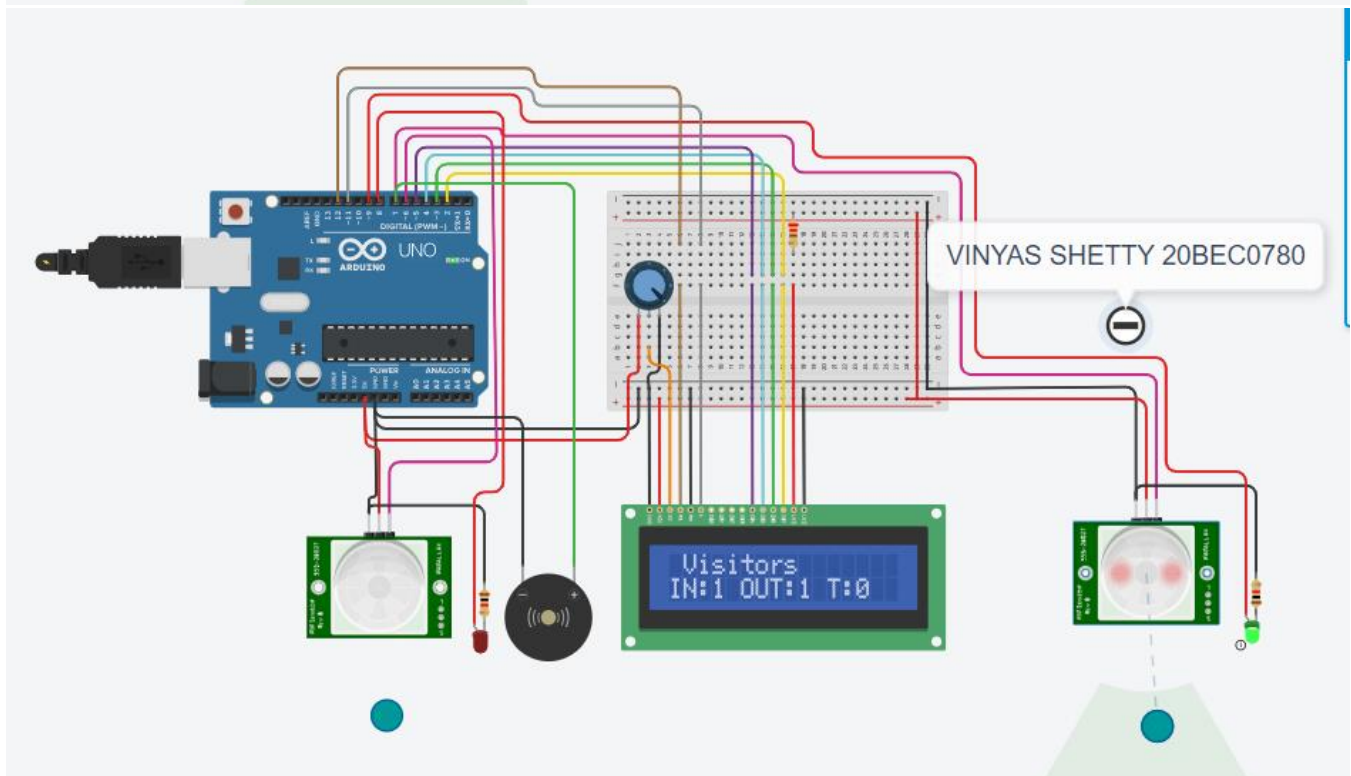
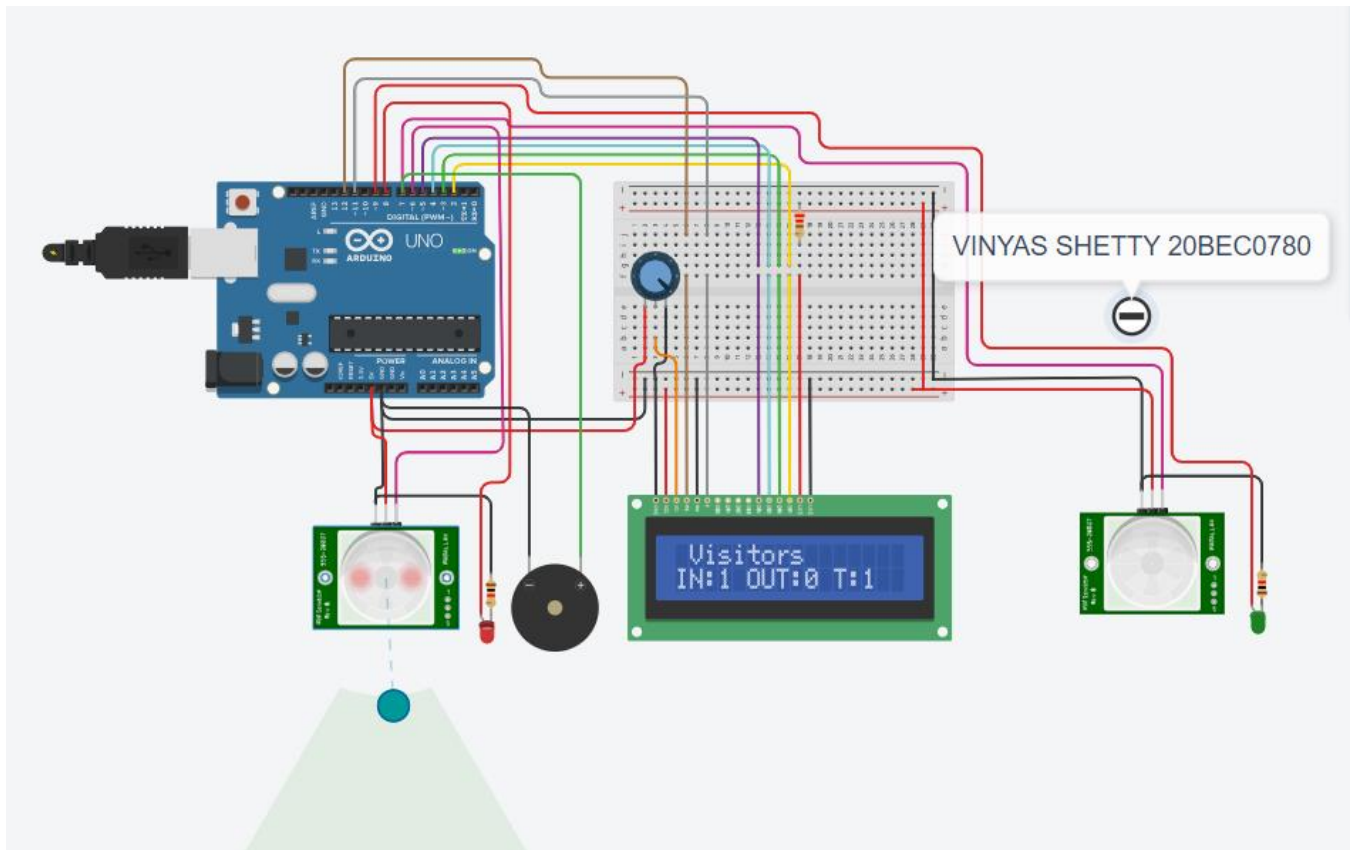
lcd.setCursor(11,1); //total position
lcd.print("T:");

lcd.setCursor(13,1); //number position
lcd.print(in - out);
delay(1000);
}

```

RESULTS SCREENSHOTS:



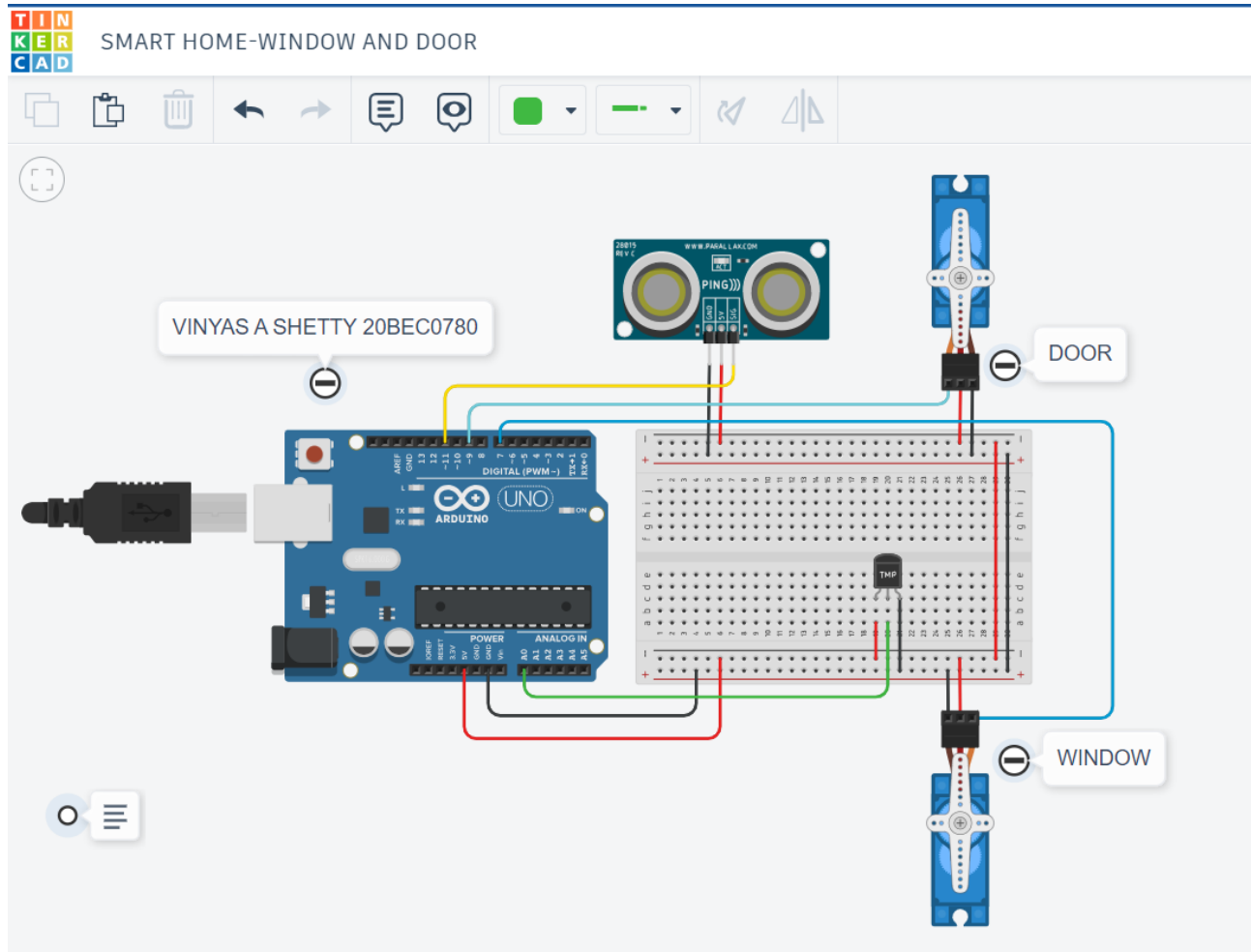


OBSERVATIONS:

We observed that the total number of people inside, total number of people checking IN and total number of people checking OUT was displayed on the LCD display

QUESTION 3:

Construct an application of temperature sensor and ultrasonic sensor –**Smart home**

CIRCUIT DIAGRAM:

CODE:

```

#include <Servo.h>
int distanceValue = 0;
int LDRValue = 0;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
    pinMode(triggerPin, OUTPUT); // Clear the trigger
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    // Sets the trigger pin to HIGH state for 10 microseconds
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    pinMode(echoPin, INPUT);
    // Reads the echo pin, and returns the sound wave travel time in microseconds
    return pulseIn(echoPin, HIGH);
}

Servo servo_7;
Servo servo_9;

void setup()
{
    pinMode(A0, INPUT);
    servo_7.attach(7, 500, 2500);
    servo_9.attach(9, 500, 2500);
}

void loop()
{
    distanceValue = 0.01723 * readUltrasonicDistance(11, 11);
    LDRValue = (-40 + 0.488155 * (analogRead(A0) - 20));
    if (LDRValue >= 30) {
        servo_7.write(90);
    }
}

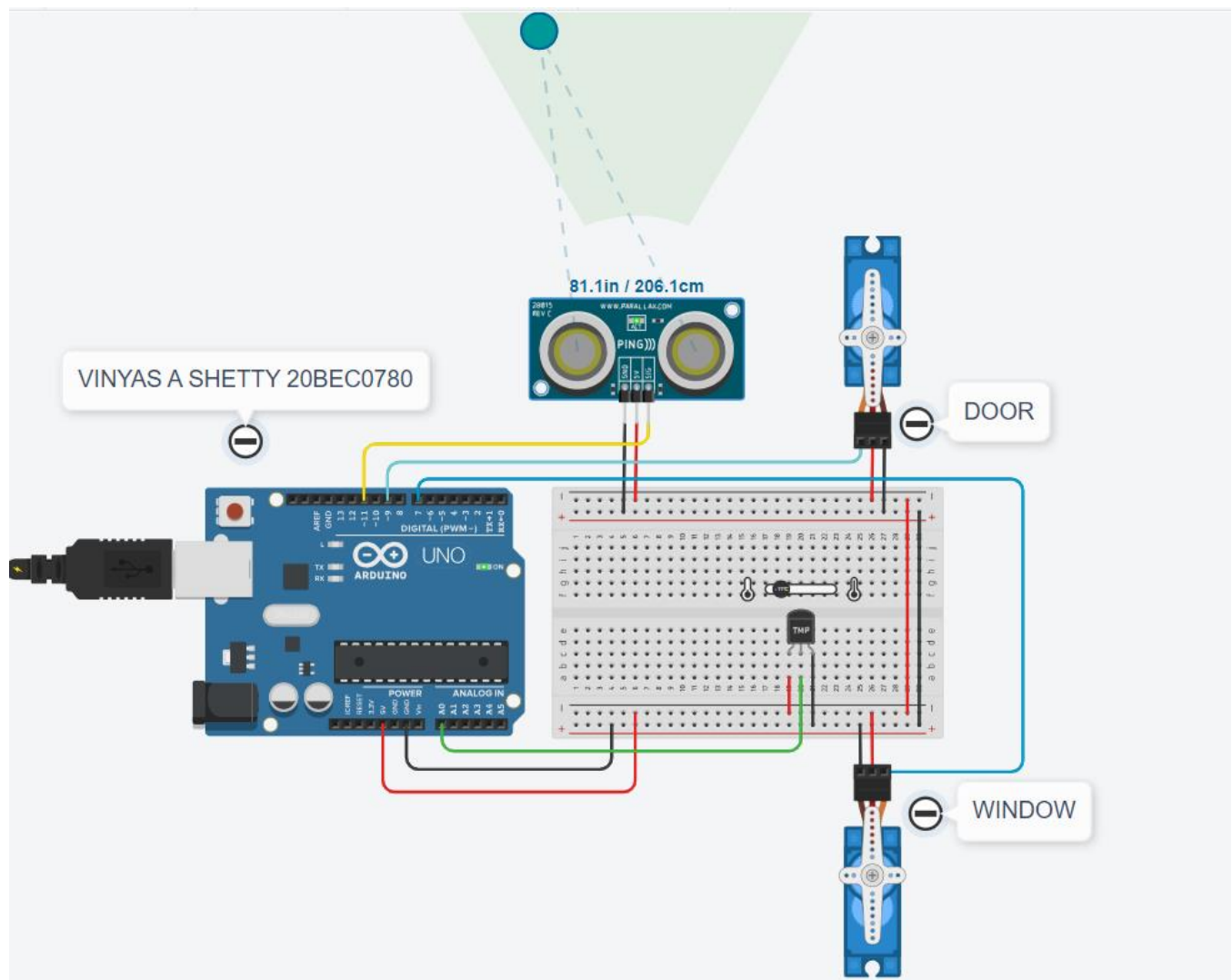
```

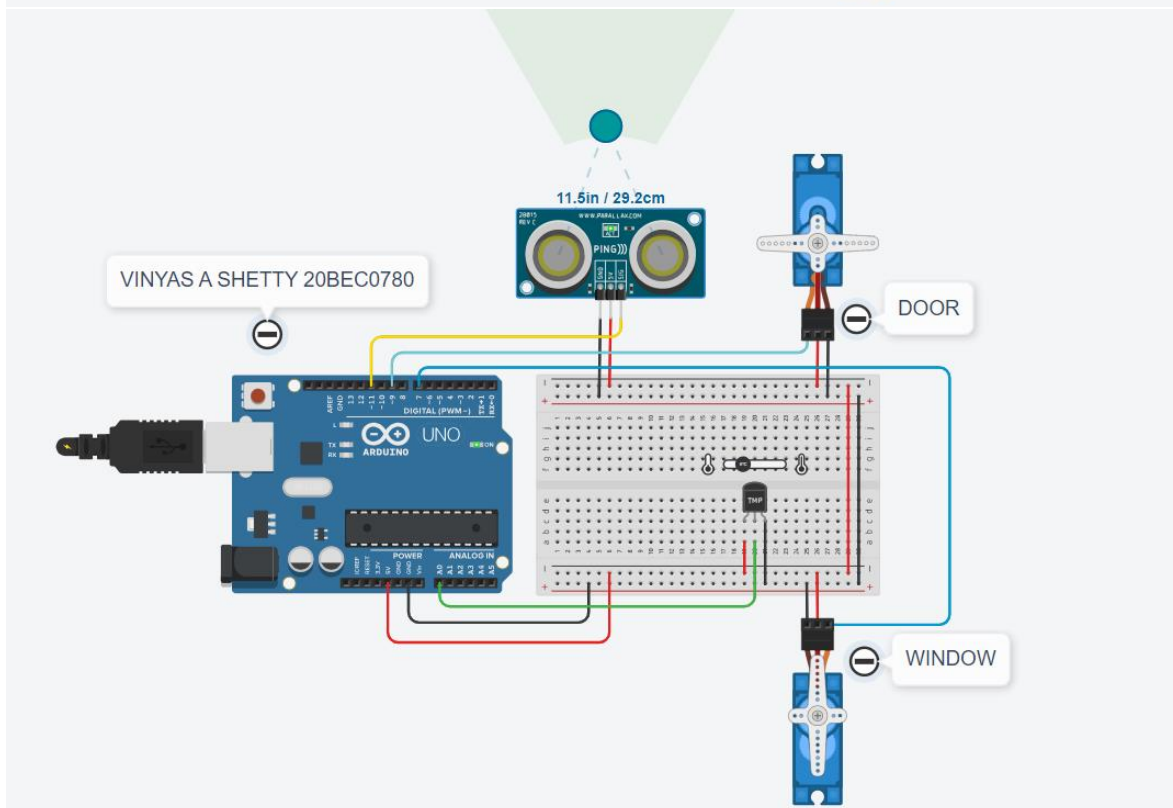
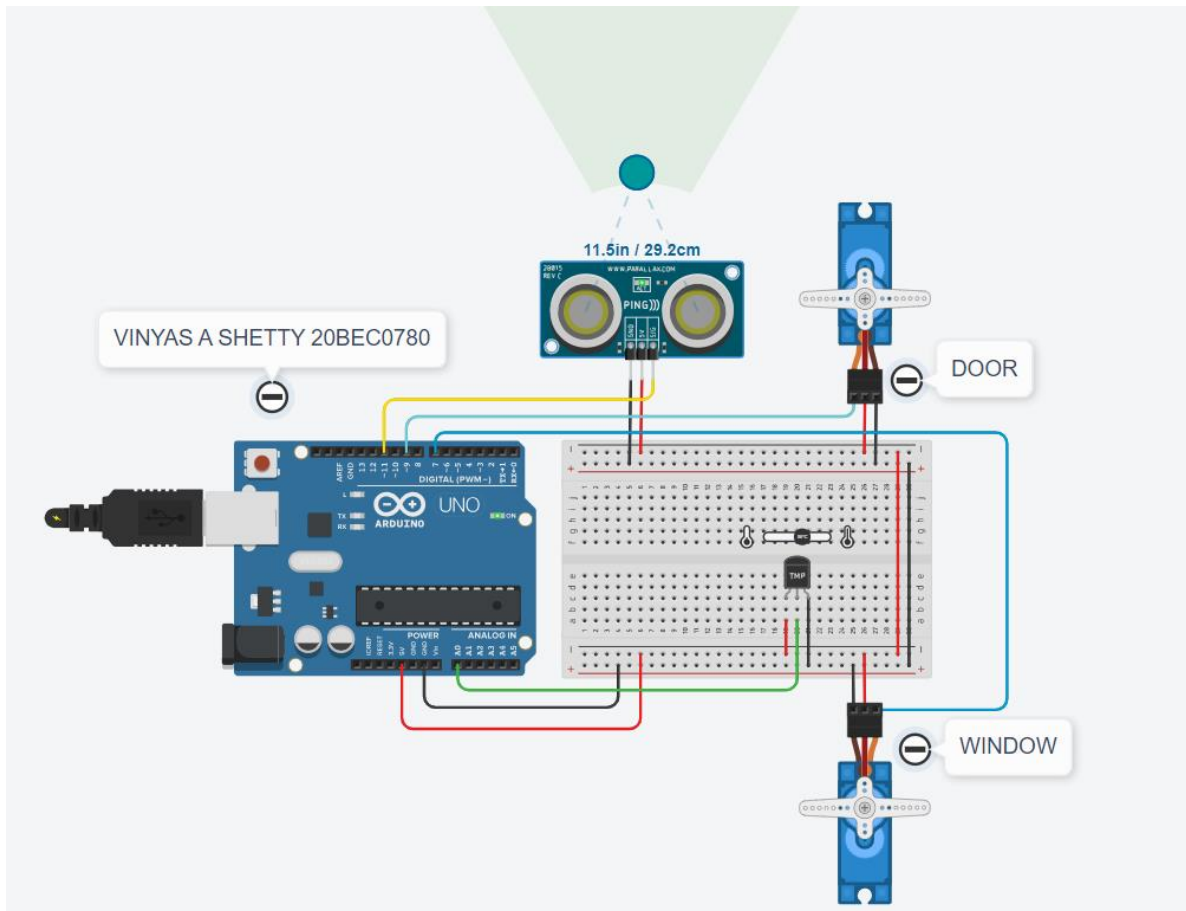
```

else {
    servo_7.write(0);
}
if (distanceValue <= 100) {
    servo_9.write(90);
} else {
    servo_9.write(0);
}
delay(10); // Delay a little bit to improve simulation performance
}

```

RESULTS SCREENSHOTS:





OBSERVATIONS:

We observed that the door opened whenever it detected an person and the window opened only when it exceeded the certain temperature