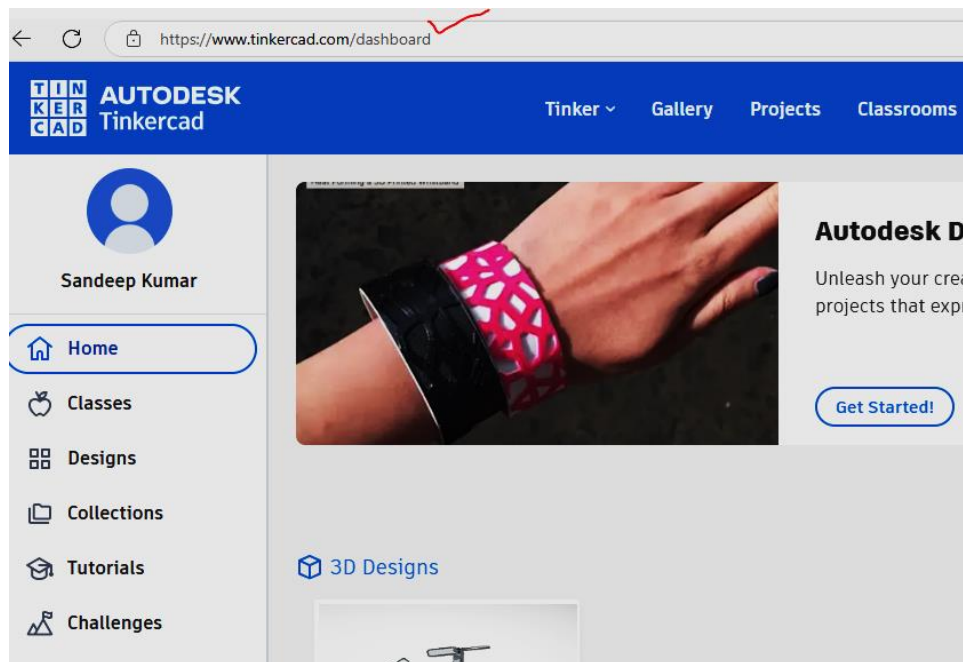## Exp-1: Hand-on Practice of IoT Circuits and Systems on the Tinkercad.

Step-1: Open the online IoT platform **Tinkercad** by using the link https://www.tinkercad.com/

Click on *login*, select login with '*Student accounts'*. On the subsequent window, click '*Sign in with Google'*. [Use your BMU email account]

You will be directed to the dashboard as shown below:

Step-2: On the dashboard, click on '*Create*' and select the '*circuit*'. You will be directed to simulation window as shown in snap below. Notice the components strip (window) on the right side; we can drag and drop the requisite components from here into the simulation workspace. In case any component is not visible here, we can search it in the search box.
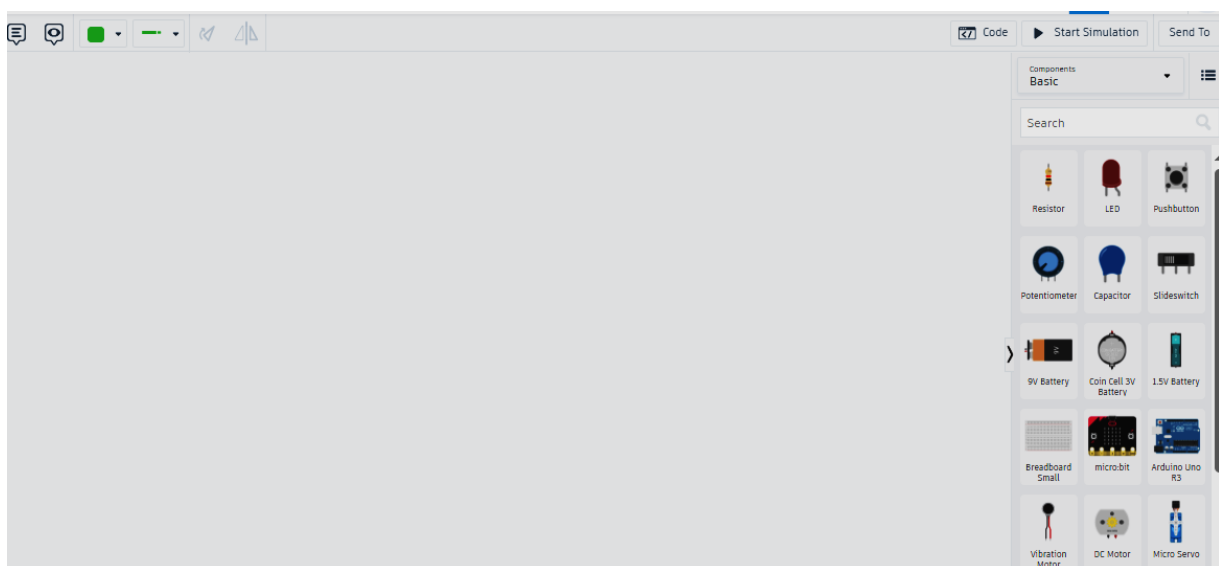
Step-3 Practice example-1 - Simulating the basic LED blink circuit using Arduino Uno.

Drag and drop the requisite components as shown in figures below. Click on the end-points or pins to create the wiring threads. Double click to join two wires together to create the junctions. To delete any component, use delete button of keyboard. Scroll on a selected component to vary its size. See the options to rotate/flip the select component on the top.
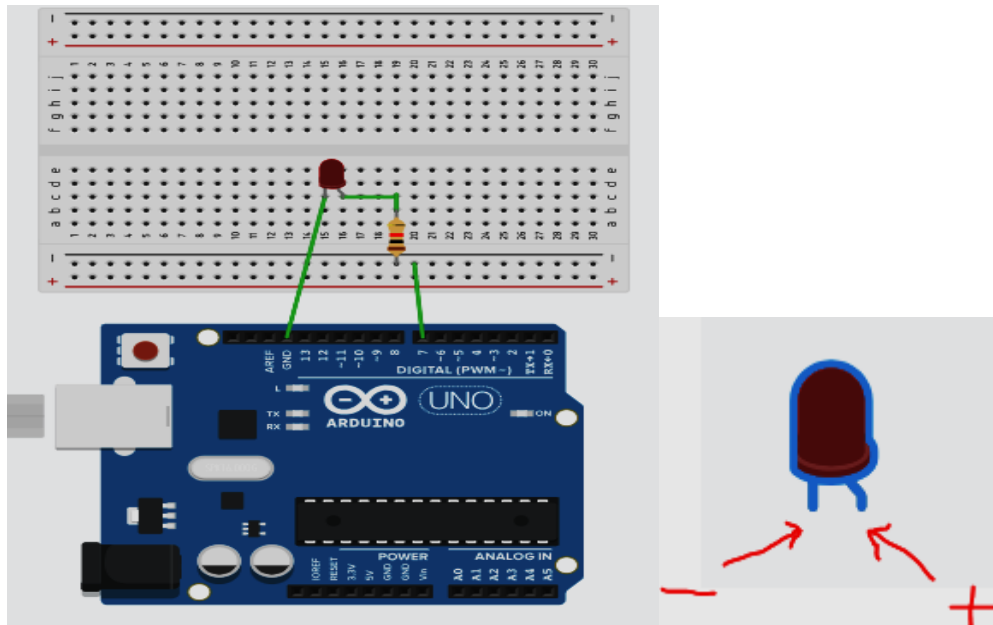
Once the circuit is completed, click on '*code*' button on the top right corner and change the option from blocks to '*text*'. See snap below.
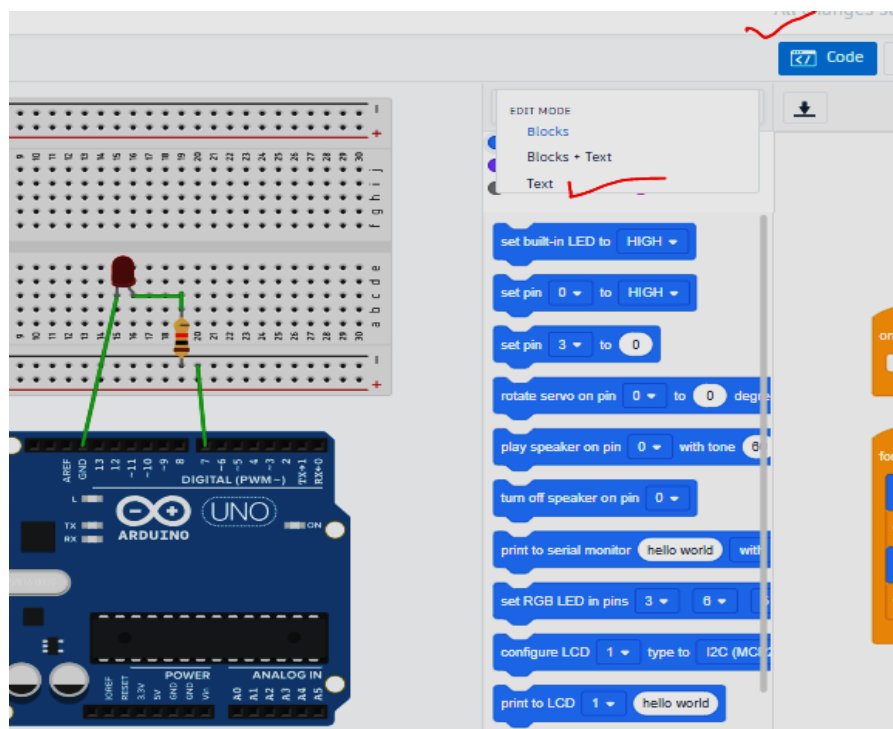
Select the font size as 'medium' on the font option (ᴀA). Write the code as shown in the snap below.
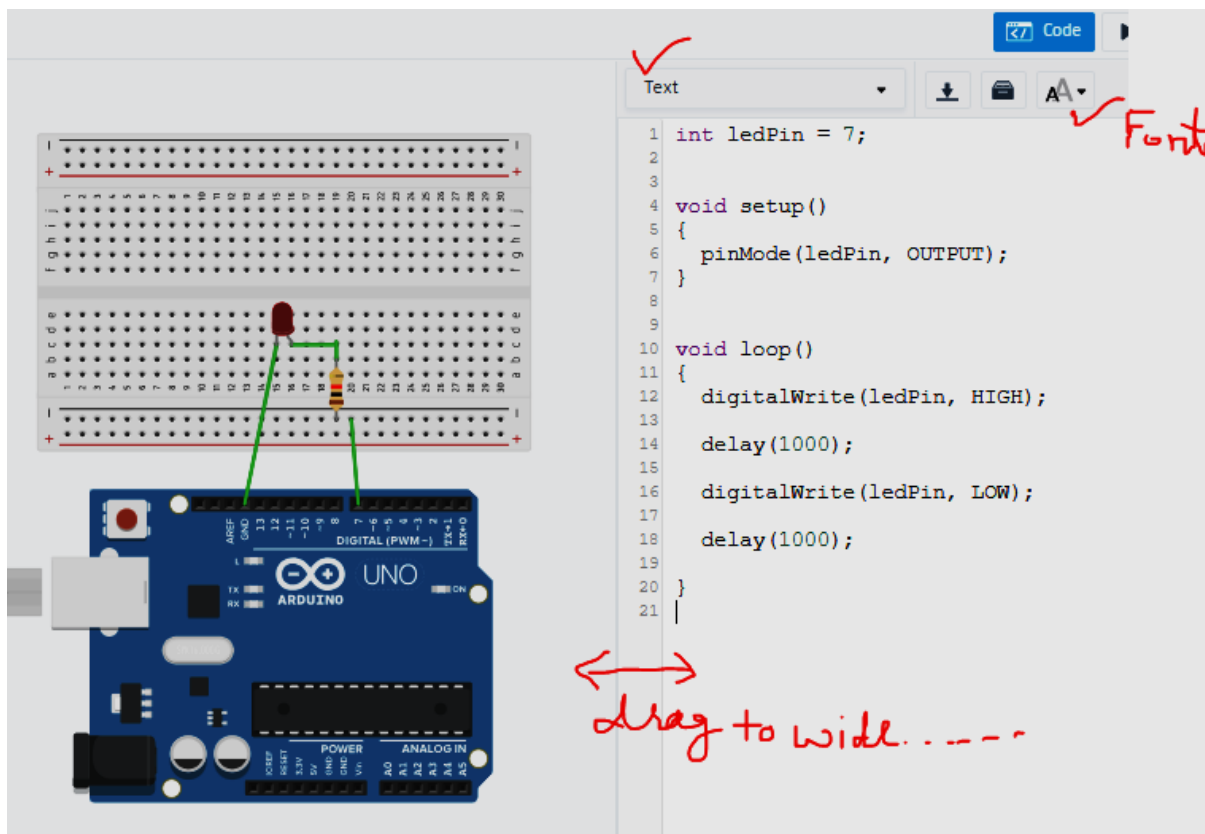
Finally, click on '**Start Simulation**' button (on top right corner) to simulate the designed circuit. You must see the LED blinking with given delay (1 sec).

To make any changes in the code, first stop the simulation, then do the requisite changes in code (e.g. making delay as 100 ms), and start the simulation once again. You must see the changes in the results (LED blinking speed).
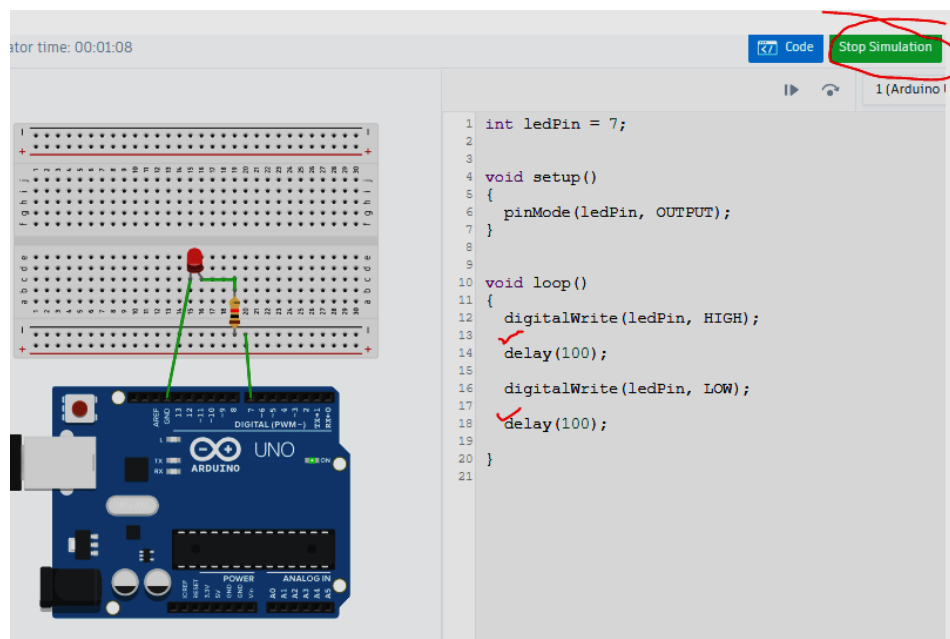
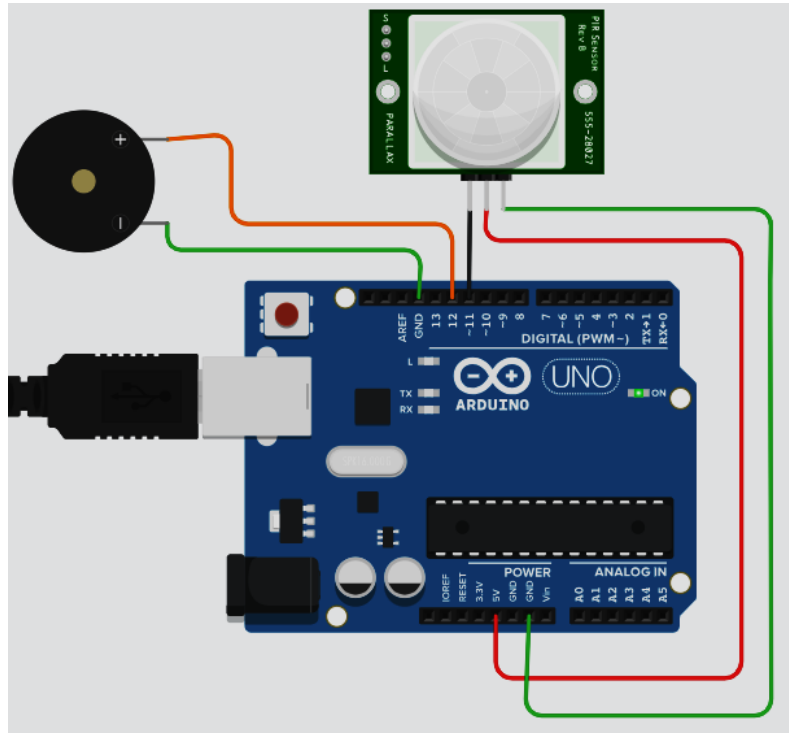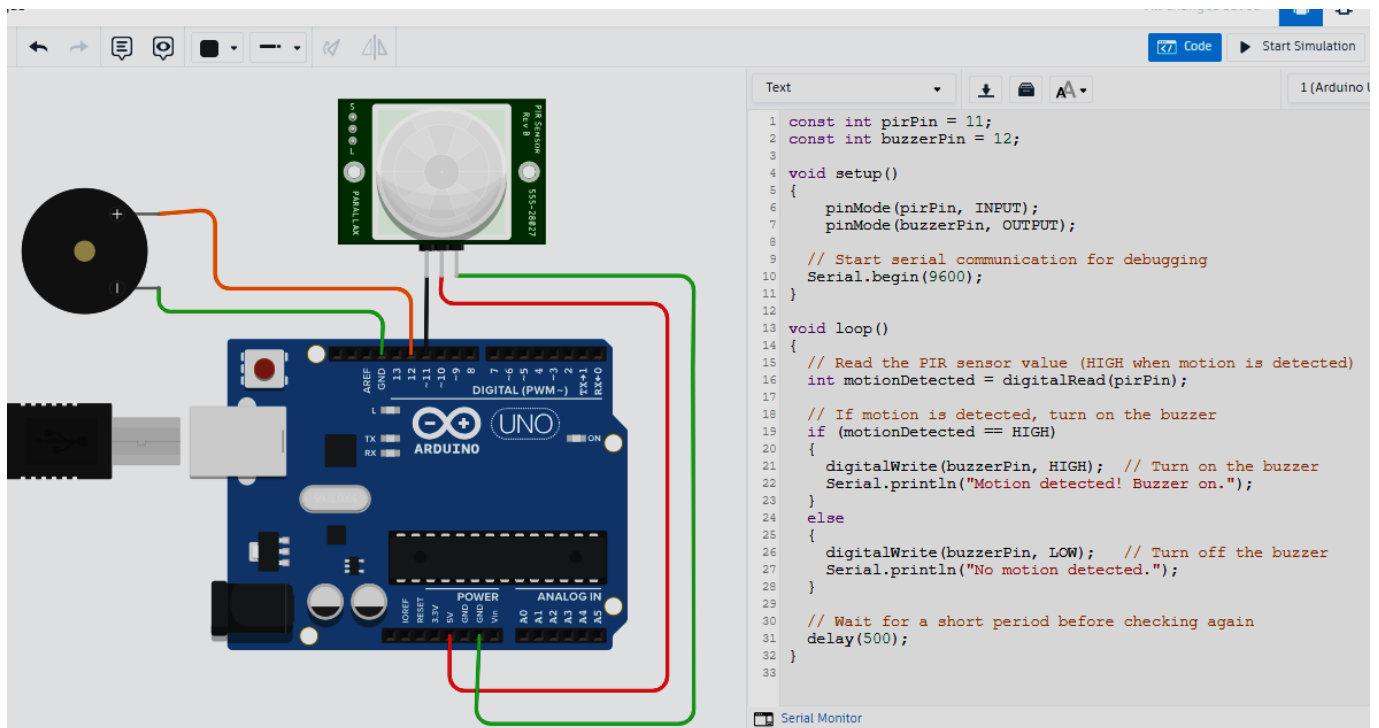**Step-4** Stop the current simulation and go to the dashboard again by clicking on the 'TENKERCAD' icon on the top left corner. Once again, *create* a new *circuit* as did in step-3.

Practice Example-2 – **Design a circuit using Arduino UNO, PIR sensor and a buzzer such that the** *buzzer will sound up as the PIR sensor detects a motion*. (Circuit, code and other info in snaps below)


[For the PIR sensor, BLACK wire is signal, RED is Vcc and GREEN is ground]

```
1  const int pirPin = 11;
2  const int buzzerPin = 12;
3
4  void setup()
5  {
6      pinMode(pirPin, INPUT);
7      pinMode(buzzerPin, OUTPUT);
8
9    // Start serial communication for debugging
10   Serial.begin(9600);
11 }
12
13 void loop()
14 {
15   // Read the PIR sensor value (HIGH when motion is detected)
16   int motionDetected = digitalRead(pirPin);
17
18   // If motion is detected, turn on the buzzer
19   if (motionDetected == HIGH)
20   {
21     digitalWrite(buzzerPin, HIGH);  // Turn on the buzzer
22     Serial.println("Motion detected! Buzzer on.");
23   }
24   else
25   {
26     digitalWrite(buzzerPin, LOW);   // Turn off the buzzer
27     Serial.println("No motion detected.");
28   }
29
30   // Wait for a short period before checking again
31   delay(500);
32 }
```
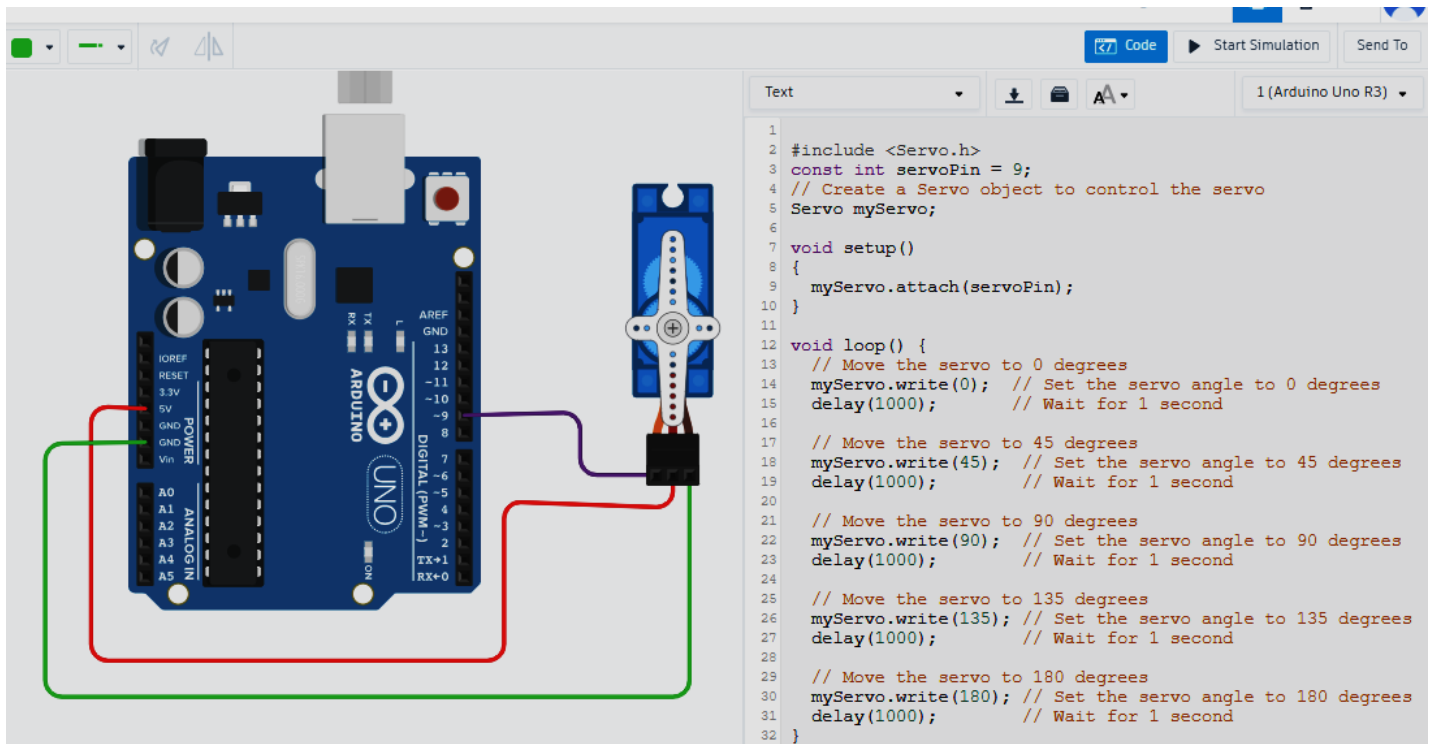
Start the simulation and click on the PIR sensor. Once you will move the object (ball in snap below), buzzer will sound up (increase volume of your laptop to hear). Also, click on serial monitor at the bottom (snap below) to see the printed messages.
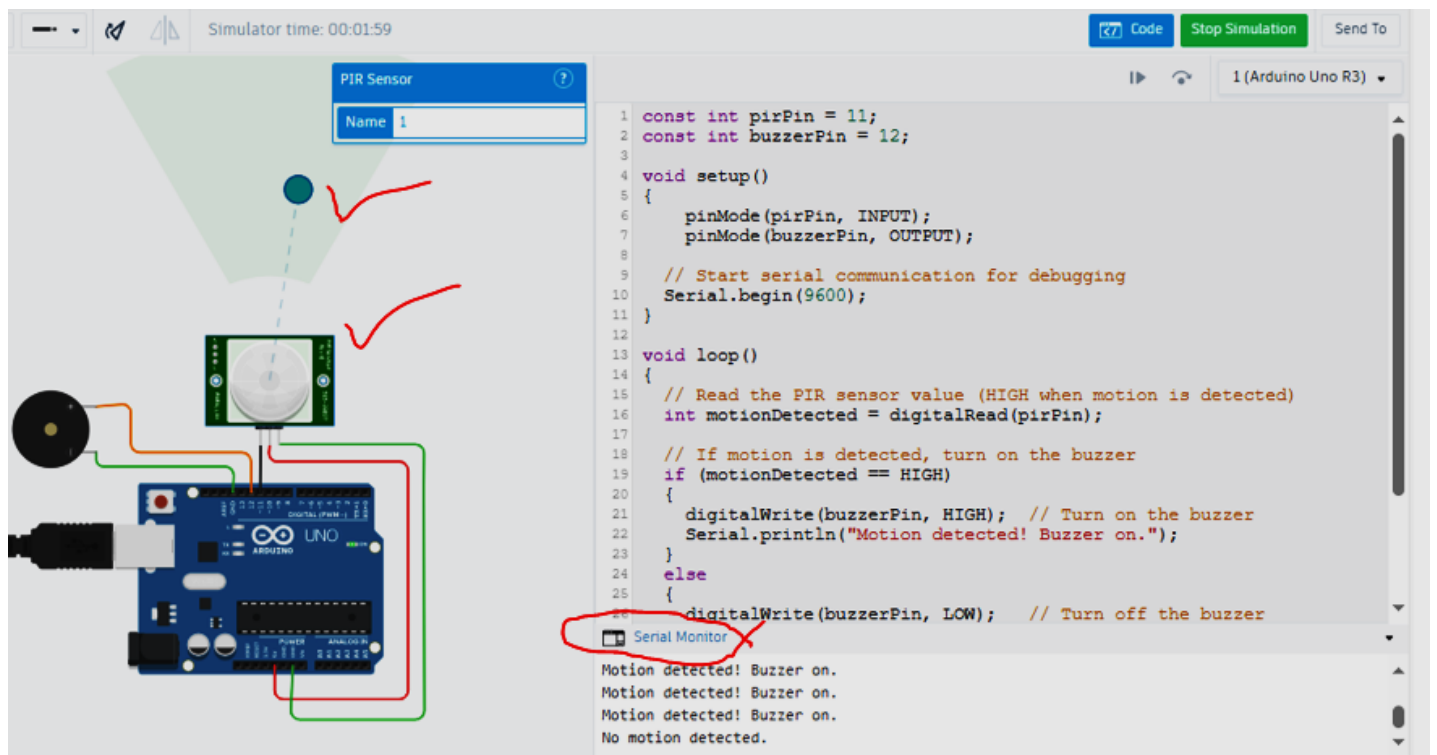
Step-5 Stop the current simulation and go to the dashboard again by clicking on the 'TENKERCAD' icon on the top left corner. Once again, *create* a new *circuit* as did in last step.

Practice Example-3: Controlling a servo motor using Arduino. (see all procedure in snap below).


```cpp
#include <Servo.h>
const int servoPin = 9;
// Create a Servo object to control the servo
Servo myServo;

void setup()
{
  myServo.attach(servoPin);
}

void loop() {
  // Move the servo to 0 degrees
  myServo.write(0);   // Set the servo angle to 0 degrees
  delay(1000);        // Wait for 1 second

  // Move the servo to 45 degrees
  myServo.write(45);  // Set the servo angle to 45 degrees
  delay(1000);        // Wait for 1 second

  // Move the servo to 90 degrees
  myServo.write(90);  // Set the servo angle to 90 degrees
  delay(1000);        // Wait for 1 second

  // Move the servo to 135 degrees
  myServo.write(135); // Set the servo angle to 135 degrees
  delay(1000);        // Wait for 1 second

  // Move the servo to 180 degrees
  myServo.write(180); // Set the servo angle to 180 degrees
  delay(1000);        // Wait for 1 second
}
```

In the above circuit, the VOILET wire is signal, RED is Vcc and GREEN is ground.

Once you start the simulation, you can see the servo motor rotating at specified angles (given in program).

Step-6: Do it yourself exercise.

Design an automatic street light controlling system by using Arduino UNO, LDR sensor and a bulb actuator (LED). The recommended circuit diagram is provided below. Implement it in Tinkercad and demonstrate the working through simulation.