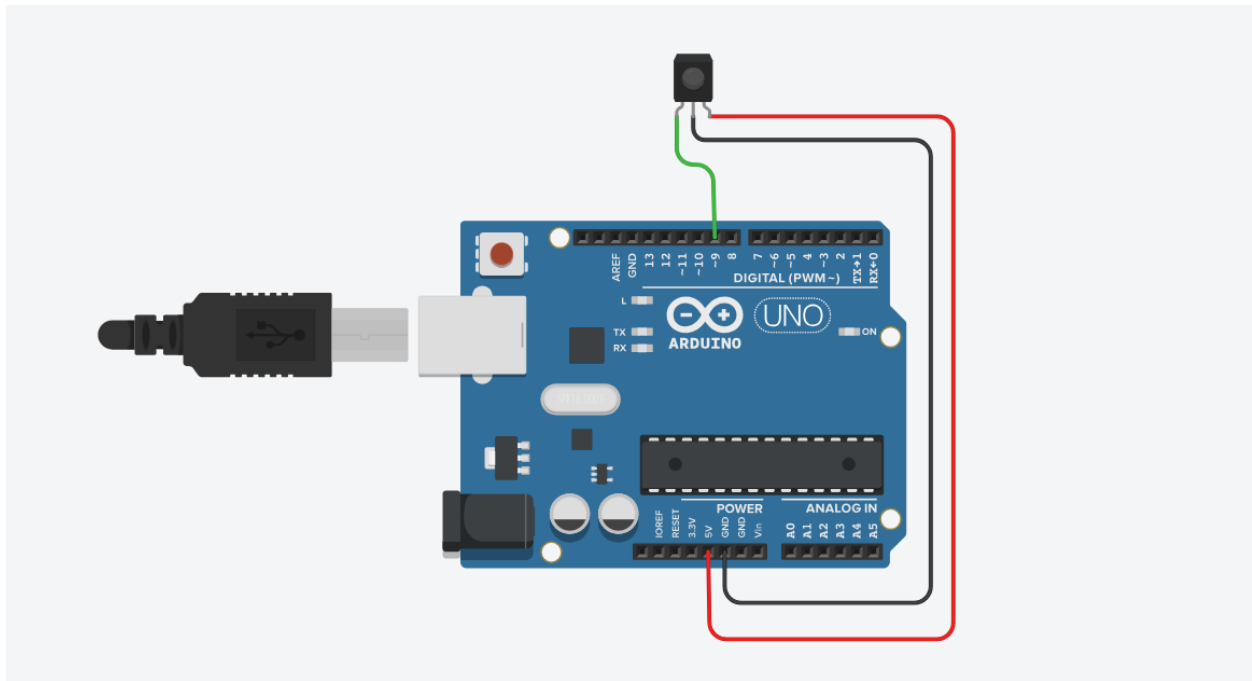


PRACTICAL NO: 1

Understanding the connectivity of Raspberry-Pi / Arduino with IR sensor. Write an application to detect obstacle and notify user using LEDs.



PROGRAM:

```
// Arduino IR Sensor Code
```

```
int IRSensor = 9; // connect ir sensor module to Arduino pin 9
```

```
int LED = 13; // connect LED to Arduino pin 13
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200); // Init Serial at 115200 Baud
```

```
  Serial.println("Serial Working"); // Test to check if serial is working or not
```

```
  pinMode(IRSensor, INPUT); // IR Sensor pin INPUT
```

```
  pinMode(LED, OUTPUT); // LED Pin Output
```

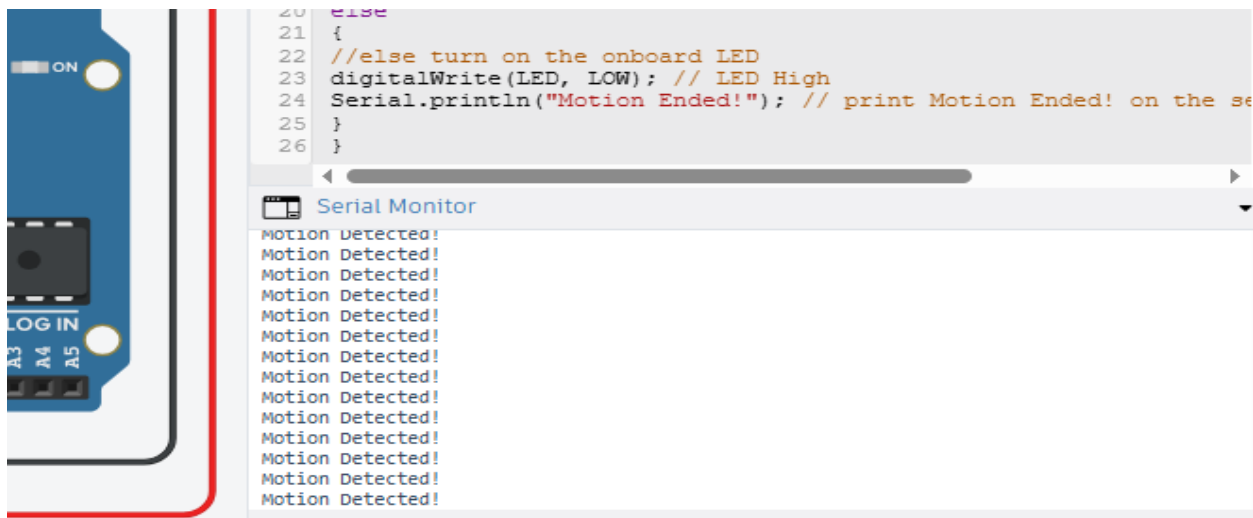
```
}
```

```

void loop()
{
int sensorStatus = digitalRead(IRSensor); // Set the GPIO as Input
if (sensorStatus == 1) // Check if the pin high or not
{
// if the pin is high turn off the onboard Led
digitalWrite(LED, HIGH); // LED LOW
Serial.println("Motion Detected!"); // print Motion Detected! on the serial monitor window
}
else
{
//else turn on the onboard LED
digitalWrite(LED, LOW); // LED High
Serial.println("Motion Ended!"); // print Motion Ended! on the serial monitor window
}
}

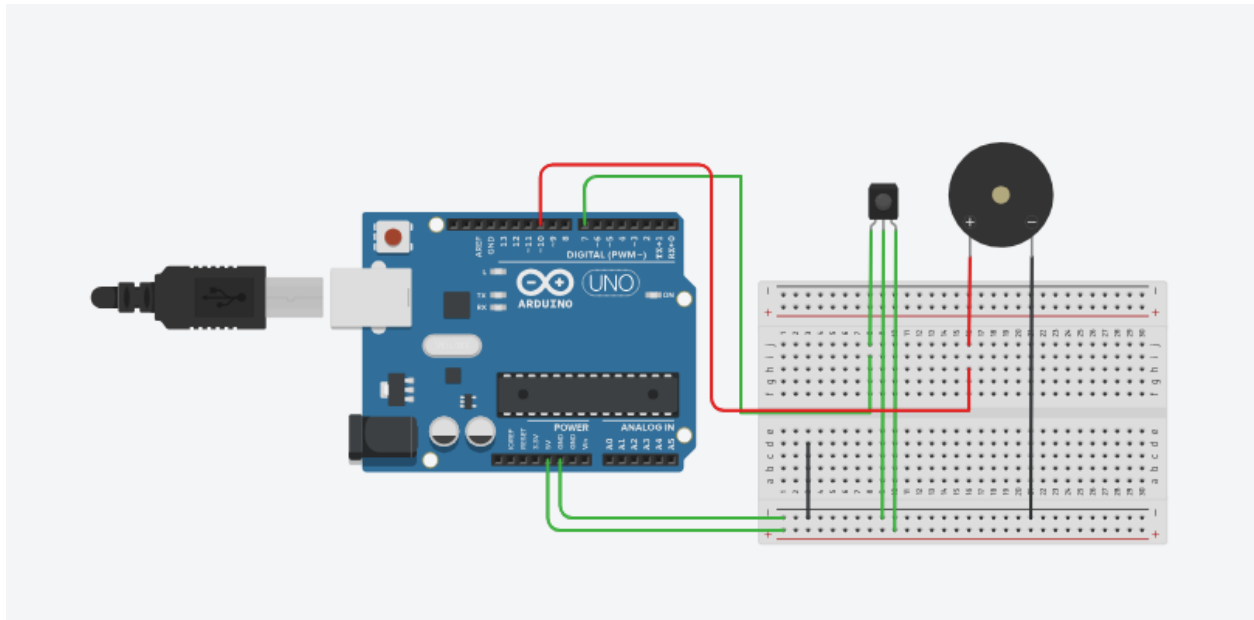
```

OUTPUT:



ASSIGNMENT 1

Draw Circuit diagram and Write a program using IR sensor when it finds an obstacle in front of it, the LED will glow or buzzer will start ringing.



PROGRAM:

```
// Define the pins for the components
```

```
const int IR_SENSOR_PIN = 7; // The IR sensor is connected to digital pin 7
```

```
const int LED_PIN = 9;      // The LED is connected to digital pin 9
```

```
const int BUZZER_PIN = 10;  // The Buzzer is connected to digital pin 10
```

```
// The setup() function runs once when you press reset or power the board
```

```
void setup() {
```

```
    // Initialize the digital pins as outputs or inputs
```

```
    pinMode(LED_PIN, OUTPUT);    // LED Pin is an Output
```

```
    pinMode(BUZZER_PIN, OUTPUT); // Buzzer Pin is an Output
```

```
pinMode(IR_SENSOR_PIN, INPUT); // IR Sensor Pin is an Input

// Start serial communication for debugging (optional but helpful)
Serial.begin(9600);
}

// The loop() function runs over and over again forever
void loop() {
    // Read the state of the IR sensor's output pin
    // The sensor outputs LOW when no obstacle is detected and HIGH when it is
    int sensorValue = digitalRead(IR_SENSOR_PIN);

    // Print the sensor value to the Serial Monitor for debugging
    Serial.print("Sensor Value: ");
    Serial.println(sensorValue);

    // Check if an obstacle is detected (sensor output is HIGH)
    if (sensorValue == HIGH) {
        // Turn the LED on by making the voltage HIGH
        digitalWrite(LED_PIN, HIGH);
        // Turn the Buzzer on by making the voltage HIGH
        digitalWrite(BUZZER_PIN, HIGH);
    } else {
        // Turn the LED off by making the voltage LOW
        digitalWrite(LED_PIN, LOW);
        // Turn the Buzzer off by making the voltage LOW
        digitalWrite(BUZZER_PIN, LOW);
    }
}
```

```
// A small delay to prevent overwhelming the serial monitor
delay(100);
}
```

OUTPUT:

[illegible]