Lab Assignment - 2

Problem Statement -Write an Arduino/ Raspberry pi program for interfacing with PIR sensor Experiment.

Component Required :

Quantity	Component
1	Arduino Uno R3
1	PIR Sensor
1	LED
1	Piezo
1	220 Ω Resistor

Steps To Perfrom :

Step 1: Connect the PIR Motion Sensor

Component Pin	Connect To	Note
Power Pin (VCC)	Arduino 5V Pin	Powers the sensor.
Signal Pin (OUT)	Arduino Digital Pin 2	Sends a "motion detected" signal.
Ground Pin (GND)	Arduino GND Pin	Completes the sensor's circuit.

Step 2: Connect the LED and Resistor

Component Pin	Connect To	Note
LED Anode (Longer Leg)	Arduino Digital Pin 3	This is the positive side of the LED.
LED Cathode (Shorter Leg)	One leg of the 220Ω Resistor	Connects the negative side to the resistor.
Resistor (Other Leg)	Arduino GND Pin	The resistor protects the LED from damage.

Step 3: Connect the Buzzer

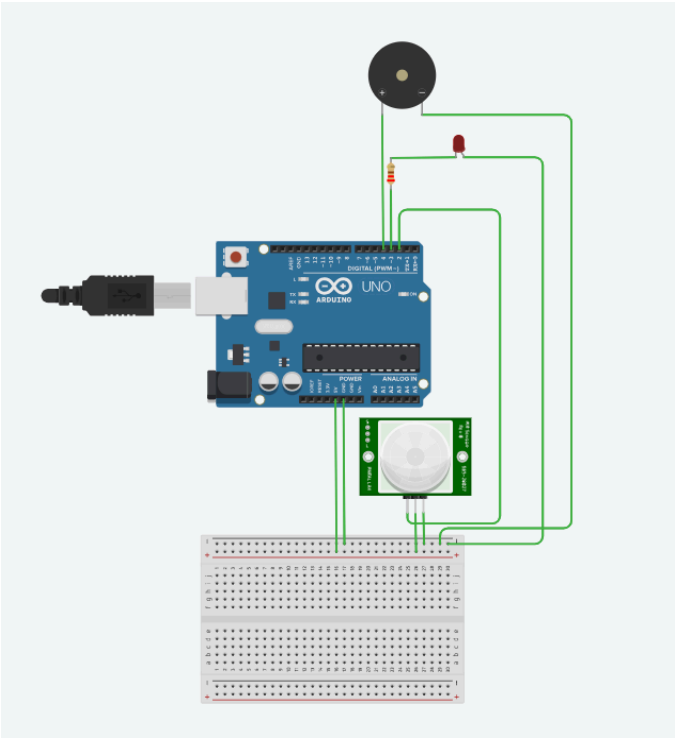
Component Pin	Connect To	Note
Buzzer Positive (+) Pin	Arduino Digital Pin 4	Receives the signal to make a sound.
Buzzer Negative (-) Pin	Arduino GND Pin	Completes the buzzer's circuit.

Code :

```
const int pirPin    = 2;   // PIR sensor output pin
const int ledPin    = 3;   // LED pin
const int buzzerPin = 4;   // Buzzer pin

int pirState = LOW;        // Tracks previous motion state

void setup() {
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  Serial.begin(9600);
  Serial.println("PIR Motion Detector Ready.");
}

void loop() {
  int motion = digitalRead(pirPin);

  if (motion == HIGH) {
    digitalWrite(ledPin, HIGH);
    playPoojaTone();              // Play three-note chime

    if (pirState == LOW) {        // Only print once when motion starts
      Serial.println("Motion Detected!");
      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW);
    noTone(buzzerPin);            // Stop buzzer tone
```

```
    if (pirState == HIGH) {        // Only print once when motion ends
      Serial.println("Motion Ended.");
      pirState = LOW;
    }
  }
}

void playTone() {
  // Three quick ascending notes like a small bell
  tone(buzzerPin, 800);   delay(200);
  tone(buzzerPin, 1000);  delay(200);
  tone(buzzerPin, 1200);  delay(200);
  noTone(buzzerPin);      delay(200);
}
```

Output :



Serial Monitor

```
PIR Motion Detector Ready.
Motion Detected!
Motion Ended.
```