**Project Report: Motion Detection System with Arduino and PIR Sensor**

1. Introduction:

The objective of this project is to create a motion detection system using an Arduino microcontroller and a Passive Infrared (PIR) sensor. The system will activate a buzzer when motion is detected. This application can find practical use in security systems, automated lighting, and other scenarios where motion detection is crucial.

2. Components Used:

- Arduino Uno

- PIR Sensor

- Buzzer

- Arduino Kit

- Jumper Wires

\*\*3. Wiring Instructions:\*\*

- Connect the VCC pin of the PIR sensor to the 5V pin on the Arduino.

- Connect the GND pin of the PIR sensor to the GND pin on the Arduino.

- Connect the OUT pin of the PIR sensor to analog pin A3 on the Arduino.

- Ensure that all components are properly powered and grounded.

![Wiring Diagram](https://example.com/wiring\_diagram.png)

4. Working Principle:

The PIR sensor detects changes in infrared radiation, such as those caused by the motion of an object. When motion is detected, the sensor sends a HIGH signal to the Arduino through the OUT pin. The Arduino, upon receiving this signal, activates the connected buzzer to indicate the presence of motion.

5. Code Explanation:

const int pirPin = A3; // PIR sensor connected to analog pin A3

const int buzzerPin = 4; // Buzzer connected to digital pin D4

void setup() {

pinMode(pirPin, INPUT);

pinMode(buzzerPin, OUTPUT);

Serial.begin(9600);

}

void loop() {

int pirValue = digitalRead(pirPin); // Read PIR sensor value

if (pirValue == HIGH) {

// Motion detected, turn on buzzer

digitalWrite(buzzerPin, HIGH);

Serial.println("Motion Detected!");

delay(1000); // Buzzer on for 1 second (you can adjust this duration)

digitalWrite(buzzerPin, LOW);

} else {

// No motion, turn off buzzer

digitalWrite(buzzerPin, LOW);

}

delay(500); // Adjust the delay based on your application and sensitivity

}

The code initializes the PIR sensor and buzzer pins, continuously checks the PIR sensor's output, and triggers the buzzer when motion is detected. The Serial.println() statement is included for debugging purposes and can be viewed using the Arduino IDE's Serial Monitor.

6. Conclusion:

In conclusion, this motion detection system provides a simple and effective solution for detecting motion using a PIR sensor and Arduino. The provided code, along with the wiring instructions, allows for easy implementation and customization based on specific project requirements. This system can serve as a foundation for more advanced applications in home automation, security, and beyond.