1. Introduction

This project is a motion-controlled color changer that uses a PIR sensor to detect motion. It activates a buzzer and changes the color of an RGB LED based on whether motion is detected, indicating either a "Safe" or "Alert" mode.

2. Key Components

- PIR Motion Sensor
- RGB LED
- Buzzer
- Arduino Board (e.g., Arduino Uno)
- Jumper wires

3. Working Principle

- **Motion Detection**: The PIR sensor detects any motion in the vicinity. When motion is detected, it sends a HIGH signal to the Arduino.
- **Alert Mode (Motion Detected)**:
  - When motion is detected, the buzzer is activated.
  - The RGB LED turns red to indicate an "Alert" status.
- **Safe Mode (No Motion Detected)**:
  - When no motion is detected, the buzzer is turned off.
  - The RGB LED turns green to indicate a "Safe" status.
- **Code Execution**: The system continuously checks for motion and updates the LED color and buzzer state accordingly, providing real-time feedback.

4. Circuit Overview

- **PIR Sensor** is connected to pin 7 of the Arduino to detect motion.
- **Buzzer** is connected to pin 3 to provide an audible alert.
- **RGB LED** is connected to pins 9, 10, and 11 for controlling the red, green, and blue colors.
- The system uses analogWrite() to control the LED colors and digitalWrite() to control the buzzer.

5. Code

cpp

CopyEdit

```cpp
/*Code written by-
Fuad Hasan
BME, KUET */


/*Motion Controlled Color Changer*/


int pirPin = 7;
int buzzer = 3;
int redPin = 9;
int greenPin = 11;
int bluePin = 10;


void setup() {
  pinMode(pirPin, INPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  //pinMode(relayPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int motionDetected = digitalRead(pirPin);
  if (motionDetected == HIGH) {
    // Alert Mode
    digitalWrite(buzzer, HIGH);
    //digitalWrite(relayPin, HIGH);
    setColor(255, 0, 0); // Red - Alert
    Serial.println("Motion Detected!");
    delay(5000); // Alarm duration
  } else {
    // Safe Mode
    digitalWrite(buzzer, LOW);
    //digitalWrite(relayPin, LOW);
    setColor(0, 255, 0); // Green - Safe
  }
}

void setColor(int red, int green, int blue) {
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

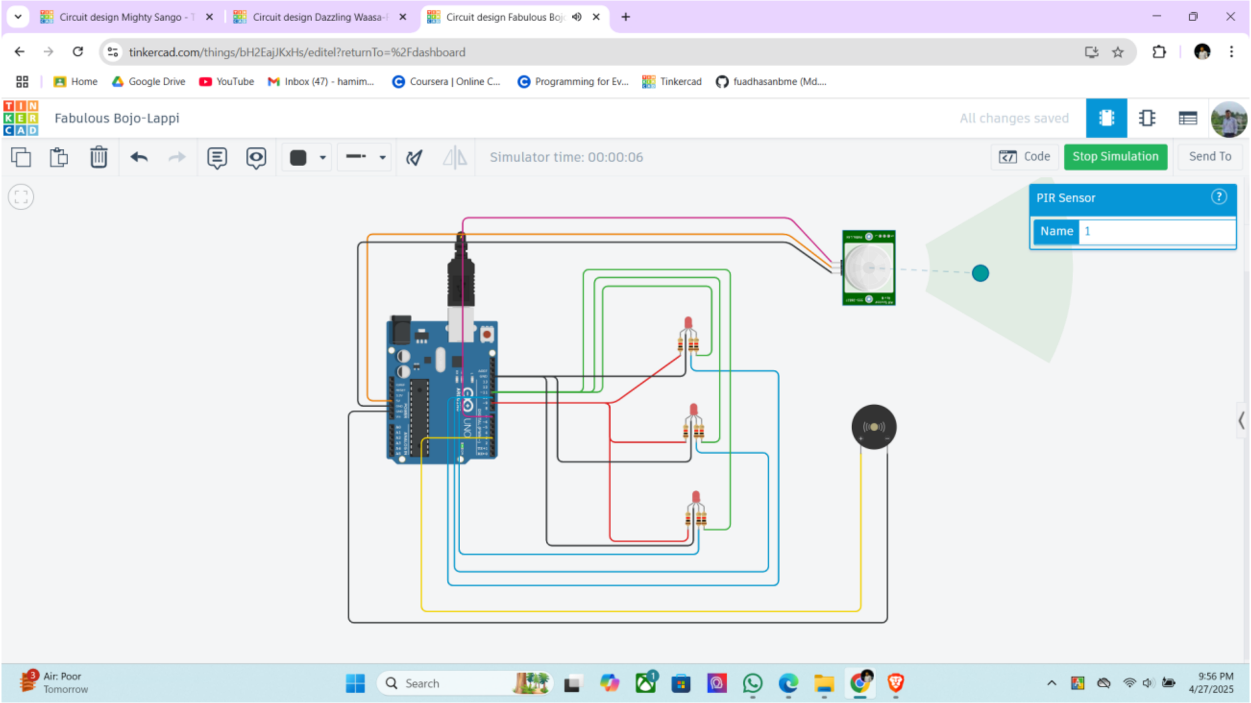
6. Code Explanation

- **Pin Setup**:

  - pirPin = 7: PIR sensor connected to pin 7 to detect motion.
  - buzzer = 3: Buzzer connected to pin 3 for sound alert.
  - redPin = 9, greenPin = 11, bluePin = 10: These pins control the RGB LED.
  - Serial.begin(9600): Initializes the serial monitor for debugging.
- **Main Loop**:
  - int motionDetected = digitalRead(pirPin);: Reads the PIR sensor value.
  - If motion is detected (motionDetected == HIGH):
    - The buzzer is turned on (digitalWrite(buzzer, HIGH)).
    - The LED is set to red (setColor(255, 0, 0)).
    - A message "Motion Detected!" is printed to the serial monitor.
    - The alarm lasts for 5 seconds (delay(5000)).
  - If no motion is detected:
    - The buzzer is turned off (digitalWrite(buzzer, LOW)).
    - The LED is set to green (setColor(0, 255, 0)).
- **setColor() Function**:
  - This function uses analogWrite() to control the brightness of the red, green, and blue pins of the RGB LED, thereby changing the color.

7. Conclusion

This project provides an effective solution for motion detection with visual and auditory feedback. The PIR sensor allows for real-time detection, and the RGB LED, combined with the buzzer, provides an intuitive interface for alerting users to changes in their environment.



Simulator time: 00:00:06

PIR Sensor

Name 1