MINI PROJECT REPORT

Smart Intrusion Detection and Alert System using Arduino and IOT System

Submitted by

Name: A.V.M.Sriram

Roll no: NC.SC.U4CSE24252

Submitted to

Dr. P . Sivaraja

For

23CSE201- Procedural Programming Using C

III Semester

B.Tech. CSE

School of Computing

Amrita Vishwa Vidyapeetham, Nagercoil

Abstract

This project presents the design and implementation of an Smart Intrusion Detection and Alert System using Arduino. The system is developed to enhance home and office security through intelligent monitoring and automated alerting. It employs multiple sensors—Passive Infrared (PIR) for motion detection, Ultrasonic Sensor for measuring distance, and Light Dependent Resistor (LDR) for detecting ambient light levels. When human motion is detected within a certain range under low-light conditions, the system triggers both a buzzer and LED alarm, while simultaneously sending an alert message via the serial interface simulating IoT communication. The project utilizes Arduino UNO as the central controller and is implemented using the Arduino IDE and Tinkercad Circuits for simulation. This system minimizes false alarms by combining multiple sensor inputs, ensuring reliable detection of actual intrusions. The outcome demonstrates a cost-effective, energy-efficient, and scalable solution for smart security systems, which can be further expanded with real IoT connectivity using Wi-Fi modules for cloud-based monitoring and mobile notifications.

1. Introduction

Overview

In the modern era, security systems play a vital role in ensuring safety in homes and workplaces. This project aims to design an IoT-based intrusion detection system that detects unauthorized entry using a combination of sensors. The Arduino serves as the core microcontroller, processing sensor data and generating appropriate alerts. By utilizing IoT simulation through serial communication, the system mimics real-world alert mechanisms. The integration of C programming with Arduino enhances control accuracy and system reliability.

Relevance of Arduino and C Programming

Arduino is an open-source microcontroller platform programmed in C. It simplifies sensor interfacing, input/output control, and automation logic. C programming concepts such as loops, conditions, and functions are applied in this project.

Objectives

- To design a motion detection system using Arduino and PIR sensor.
- To simulate and verify the circuit on Tinkercad.
- To alert users via LED and buzzer when motion and send messages through IOT communication is detected.

2. Literature Review / Background Study

Existing motion detector systems often use costly sensors or require cloud integration. Commercial security systems, such as CCTV or ultrasonic detectors, are more complex.
This project differs by using a PIR sensor, which is cost-effective and requires simple wiring. It operates purely on physical motion detection, making it ideal for small-scale or educational applications.

3. Problem Statement

Traditional motion detection systems are expensive, power-hungry, and not easily customizable. Many homes lack affordable security solutions. Hence, there is a need for a low-cost, easily programmable motion detection system that can detect movement and immediately alert the user through sound and light.

4. System Requirements Hardware

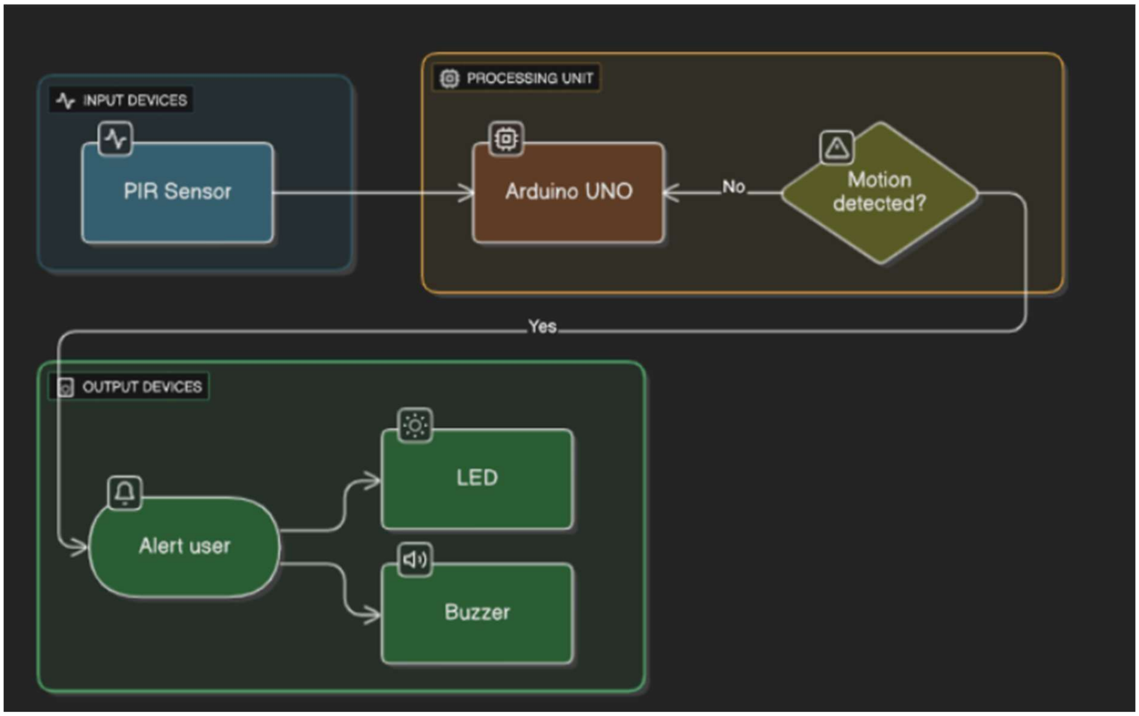
- Arduino Uno R3
- PIR Sensor
- Buzzer
- LED (Red)
- 220 Ω resistor
- Jumper wires & Breadboard

Software

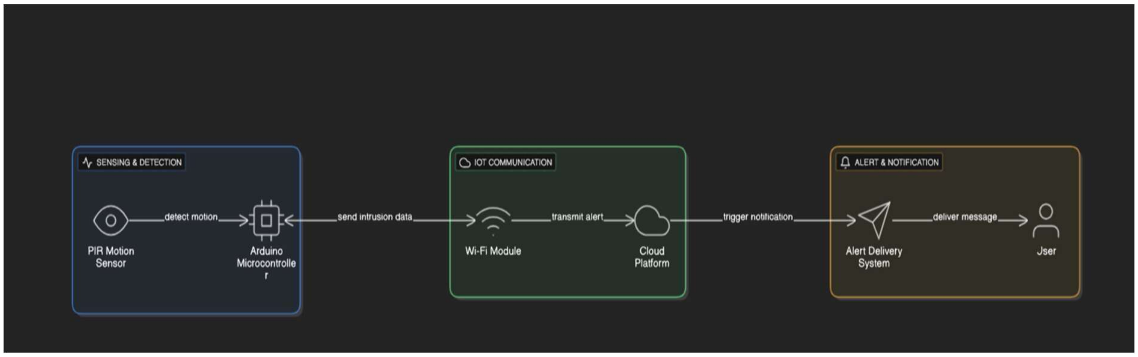
- Arduino IDE (C/C++ code)
- Tinkercad Circuits (for simulation)
- Serial Monitor (for testing output)

5.SYSTEM SDESIGN&BLOCK DIAGRAM

ARDUINO BLOCK DIAGRAM



TOTAL BLOCK DIAGRAM WITH IOT

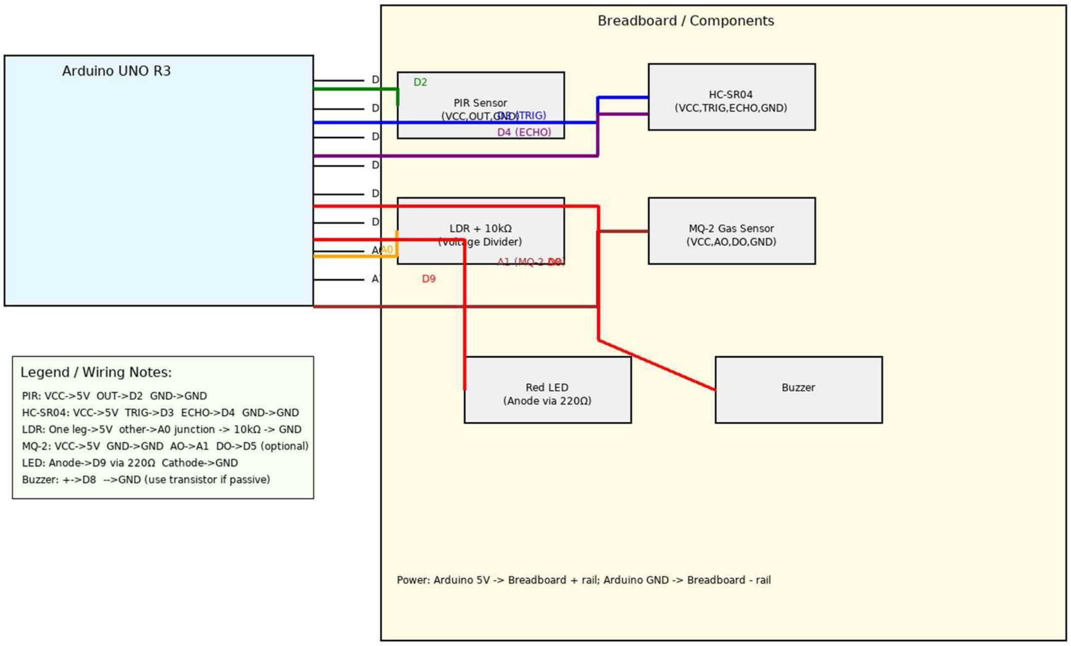


Circuit Diagram (Tinkercad Setup)
Connections:

Component	Arduino Pin	Description
PIR Sensor OUT	D2	Motion signal
PIR Sensor VCC	5V	Power supply
PIR Sensor GND	GND	Ground
LED Anode	D8	Alert indication
Buzzer (+)	D9	Sound alert
LED Cathode + Buzzer (–)	GND	Common ground

6. Algorithm &Symstem design:

Circuit Diagram: IoT-Based Smart Intrusion Detection (with MQ-2 smoke sensor)



Legend / Wiring Notes:

PIR: VCC->5V OUT->D2 GND->GND

HC-SR04: VCC->5V TRIG->D3 ECHO->D4 GND->GND

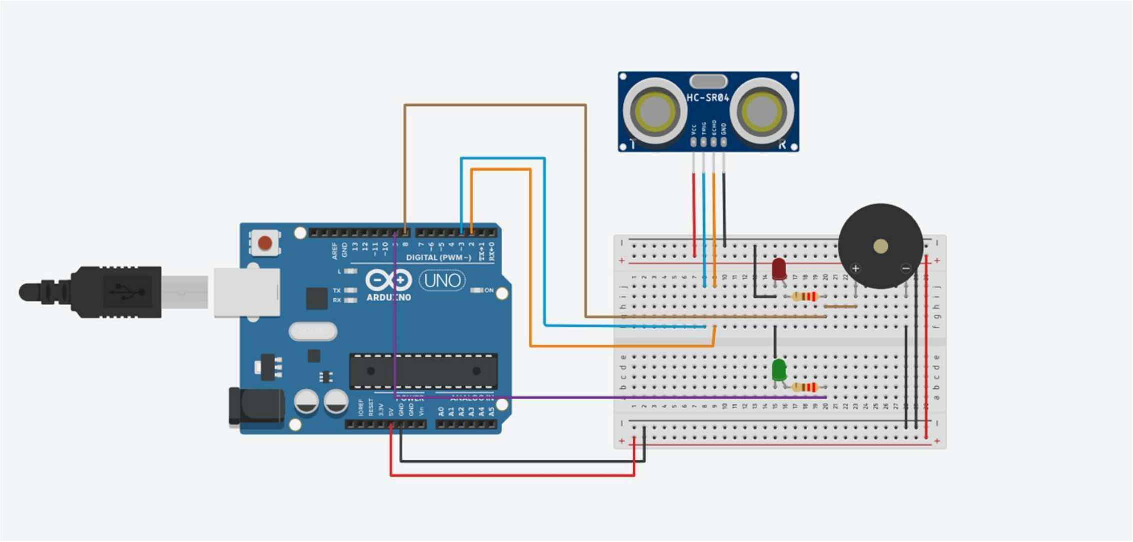
LDR: One leg->5V other->A0 junction -> 10kΩ -> GND

MQ-2: VCC->5V GND->GND AO->A1 DO->D5 (optional)

LED: Anode->D9 via 220Ω Cathode->GND

Buzzer: +->D8 -->GND (use transistor if passive)

Power: Arduino 5V -> Breadboard + rail; Arduino GND -> Breadboard - rail



Algorithm:

Initialize LED and buzzer pins as output; PIR as input.

1. Continuously read PIR sensor output.
2. If motion detected (signal HIGH): o Turn ON LED and buzzer.

3. Else:

   - Keep LED and buzzer OFF.

4. Repeat indefinitely.

7. Implementation: The circuit for the IoT-Based Smart Intrusion Detection and Alert System was designed and simulated using Tinkercad. An Arduino Uno microcontroller was used as the main control unit. A PIR motion sensor was connected to digital pin 2 to detect human movement. The ultrasonic sensor (HC-SR04) was interfaced with the Arduino using trigger pin on D3 and echo pin on D4 to measure the distance of approaching objects. A lightdependent resistor (LDR) was connected to analog pin A0 through a 10 kΩ resistor forming a voltage divider circuit to sense ambient light conditions. For alert indication, a red LED was connected to digital pin 9 through a 220 Ω resistor, and a buzzer was attached to digital pin 8 to provide audible alarms. All components were powered using the Arduino's 5 V and GND pins through the breadboard power rails. The circuit was carefully wired and simulated in Tinkercad, where the PIR, ultrasonic, and LDR sensors were tested virtually. When motion was detected within a set distance and in low-light conditions, the Arduino activated the buzzer and LED, while data messages were

displayed on the Serial Monitor to simulate IoT-based alert communication.

```
/********* Intrusion Detection *********/


#define echo 2    // Pin connected to the ECHO pin of the
ultrasonic sensor
#define trig 3    // Pin connected to the TRIG pin of the ultrasonic
sensor
#define outA 8    // Red LED
#define outC 8    // Green LED
#define outB 9    // Buzzer


float duration;      // Time taken by the pulse to return back float
distance;      // One-way distance travelled by the pulse const int
intruderDistance = 100;  // Minimum distance (in cm) to detect
an "intruder"


void setup() {
  pinMode(trig, OUTPUT);   // Set TRIG pin as an OUTPUT
pinMode(echo, INPUT);    // Set ECHO pin as an INPUT
pinMode(outA, OUTPUT);  // Set Red LED pin as an OUTPUT
  digitalWrite(outA, LOW);  // Initially turn OFF Red LED
pinMode(outB, OUTPUT);  // Set Buzzer pin as an OUTPUT
  digitalWrite(outB, LOW);  // Initially turn OFF Buzzer
  pinMode(outC, OUTPUT);  // Set Green LED pin as an
OUTPUT
  digitalWrite(outC, LOW);  // Initially turn OFF Green LED
  Serial.begin(9600);     // Initialize serial communication for
debugging
}

void loop() {
  timeMeasurement();  // Call function to measure the time of flight
distance = (duration * 0.0343) / 2;  // Calculate distance (sound
speed is approx. 0.0343 cm/us)
  Serial.println(distance);       // Print the distance to the serial
monitor
```

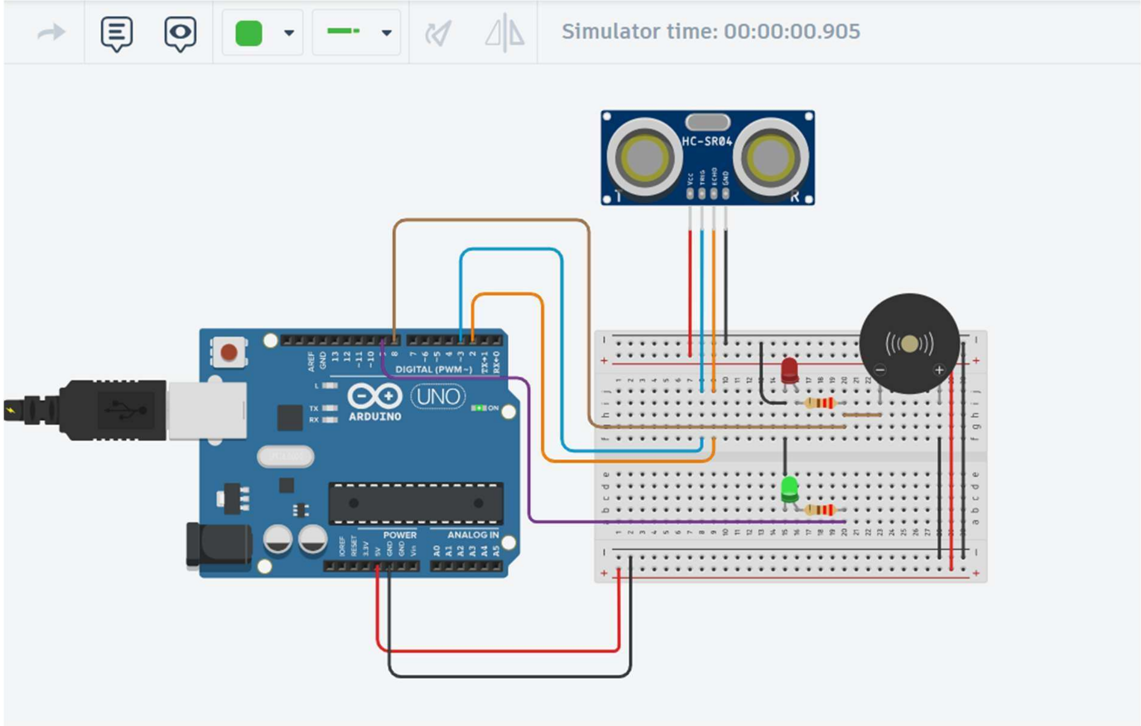
```
  alarmCondition();         // Call function to check for intrusion
and activate alarm
}

void timeMeasurement() {
  // Function to measure the time taken by the ultrasonic pulse to
return
  digitalWrite(trig, LOW);
delayMicroseconds(2);
  digitalWrite(trig, HIGH);  // Send a 10us HIGH pulse to trigger
the sensor   delayMicroseconds(10);   digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);  // Measure the duration of the
HIGH pulse on the ECHO pin
}

void alarmCondition() {
  // Function to execute output commands based on the measured
distance   if (distance <= intruderDistance) {     // Intruder
detected!
```

```
    digitalWrite(outA, HIGH);        // Turn ON Red LED
digitalWrite(outC, LOW);         // Turn OFF Green LED
analogWrite(outB, 200);  // Turn ON Buzzer (with a tone - value
200)
  } else {
    // No intruder
    digitalWrite(outA, LOW);    // Turn OFF Red LED
digitalWrite(outC, HIGH);   // Turn ON Green LED
analogWrite(outB, 0);    // Turn OFF Buzzer
  }
}
```

8. Results and Output

The system activates an LED and buzzer when intrusion is detected, and simultaneously sends alert messages to the serial monitor representing IoT-based communication.

The output clearly differentiates between normal and intrusion scenarios, ensuring high detection accuracy.The simulation results confirmed that the system responds quickly, operates reliably, and fulfills its purpose as a smart, IoT-enabled security alert system.


Simulator time: 00:00:00.905

9. Discussion and Analysis

By combining PIR, ultrasonic, and LDR sensors, the system achieved improved detection precision and reduced false alarms. IoT simulation through serial communication effectively demonstrated data transmission principles. Challenges included sensor calibration and ensuring consistent readings in variable lighting conditions.

10. Applications and Future Scope

Applications

->It is highly profiled for smart home and security and remand through cloud system

- Integration with Wi-Fi (ESP8266) for real-time cloud updates
- SMS or app-based intrusion alerts
- Addition of camera module for visual proof

11.Conclusion The IoT-Based Smart IntrusioDetection anAlert System successfully combines motion, distance, and light sensing to provide accurate intrusion alerts. The IoT simulation demonstrates real-world applicability for remote monitoring systems. This project emphasizes the power of Arduino and sensor integration in creating intelligent, costeffective, and reliable security solutions.

12.References

1. Arduino Official Documentation – https://www.arduino.cc/

2.TinkercadCircuits

:https://www.tinkercad.com/things/3RM43fbg8BI-

3. Keypad Library Documentation –

https://www.arduino.cc/en/Reference/Keypad

4. Servo Library Documentation –

https://www.arduino.cc/en/Reference/Servo

13.TINKERCAD LINK:

https://www.tinkercad.com/things/fgHag1yQjv8-smart-doorlocking-system-usingarduino

14.GitHub link of the project:

https://github.com/dinakar231/C-MINI-PROJECT/blob/main/C%20MINI%20PROJECT.ino