NOTRE DAME UNIVERSITY BANGLADESH

Microprocessor & Assembly Language Project Report

Submitted to:

Dr. Saheena Sultana

Submitted by: CSE-19

Shazidul Alam - 0692220005101009

Anisur Rahman Naeem – 0692220005101015

Esha Silvia Rozario – 0692220005101004

Tasdiqul Huq – 0692220005101006

Index:

1. Introduction

In today's world, safety and security have become major concerns for homeowners. Traditional security systems often lack automation in emergency situations such as gas leaks or fires. This project aims to design a smart home security system that not only detects dangerous situations but also acts on them by unlocking doors automatically and raising alerts, ensuring safety.

(i) Project Objectives

To develop a home security system that can detect gas leaks, fire, and intrusions.

Automate the process of unlocking the door for a safe escape during emergencies.

Alert the homeowners and neighbors using a buzzer and LED lights in case of unauthorized access or danger.

(ii) Project Motivation

The increasing number of home accidents and burglaries highlights the need for advanced security systems that can respond in real-time. This project was motivated by the need to create a reliable and affordable solution that integrates safety with security, ensuring that homeowners can respond to threats like gas leaks, fire, and unauthorized entry efficiently.

2. Organization of the Project

This section provides a overview of the different stages involved in developing the smart home security system. It follows a logical flow, starting with the hardware components used, then software development phases. Finally, it shows achieved outputs and encountered challenges.

Hardware Section: This section details the components used to build the system, including the microcontroller (Arduino Uno), sensors (gas, flame, and ultrasonic motion), actuators (servo motor, buzzer, and LED), and additional elements like a breadboard, jumper wires, and power source.

Software Section: This section focuses on the software development. It explains how the code interacts with the sensors to detect threats and activate the actuators accordingly. Additionally, it covers testing procedures to ensure system functionality and debugging techniques employed to address issues.

Output & Pictures: This section showcases the successful functionalities of the system, including automatic door unlocking during emergencies, homeowner and neighbor alerts using buzzer and LED, and intrusion detection. It may also include pictures of the prototype for visualization.

Challenges Faced: This section discusses the difficulties encountered during project development. It then elaborates on the solutions implemented to overcome these challenges.

Future Scope: This section explores potential future improvements and functionalities that could be added to the system.

References: This section provides a list of resources used for research and development in the project.

Conclusion: This section summarizes the project's achievements, highlighting the successful development of an integrated smart home security and safety system.

3. Hardware Section:

(i) Components

1) Arduino Uno R3: Main microcontroller.

2) Servo Motor: Controls the door mechanism.

3) MQ2 Gas Sensor: Detects harmful gases.

4) Flame Sensor: Detects fire.

5) Ultrasonic Motion Sensor: Detects motion for intrusion alerts.

6) Buzzer: Alerts homeowners and neighbors in emergencies.

7) LED: Visual alert indicator.

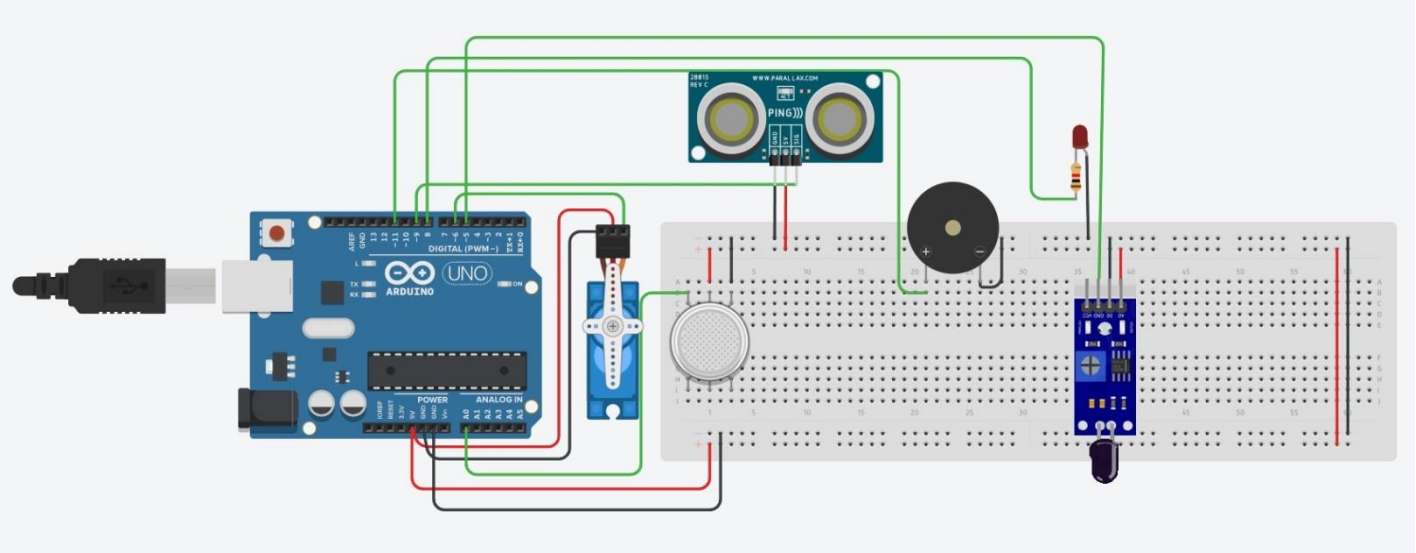
8) 9V Battery: Power source for the system.

9) Jumper Wires: For circuit connections.

10) Breadboard: For assembling the components.

(ii) Circuit Diagram

The circuit diagram shows how each component is connected to the Arduino. It includes connections from the sensors, buzzer, LED, and servo motor to the appropriate digital and analog pins on the Arduino Uno. The 9V battery is connected to the Arduino to power the system.



4. Software Section:

(i) IDE & Codes

The Arduino IDE was used to develop the code for this project. The code handles sensor inputs and actuates the servo motor, buzzer, and LED when specific conditions are met. Each sensor is constantly monitored, and depending on the readings, the system responds accordingly.

(ii) Code Analysis

Sensor Input Handling: Code reads from the MQ2 gas sensor and flame sensor to detect gas leaks and fire. If values exceed thresholds, the system triggers the motor to open the door.

Motion Detection: The ultrasonic sensor detects intrusions, sounding the buzzer and lighting the LED if motion is detected when the system is armed.

(iii) Detailed Explanation of Instructions

Gas Detection: When the gas sensor detects a certain threshold of gas, it activates the buzzer and opens the door for ventilation and safe escape.

Fire Detection: If the flame sensor detects fire, the same actions are taken to allow evacuation.

Intrusion Detection: If the motion sensor detects movement when the system is armed, the buzzer sounds to alert occupants of potential intrusion.

```
#include <Servo.h>          // Include the Servo library

// Smoke and flame sensor pins
int smokeA0 = A0;
int flameSensorPin = 5;
int buzzer = 11;
int led = 8;

// Ultrasonic sensor pins
const int trigPin = 9;
const int echoPin = 10;

// Servo motor pin
const int servoPin = 6;     // Pin connected to the servo motor
Servo myServo;               // Create a Servo object

// Variables for ultrasonic sensor
long duration;
int distance;
float sensorValue;
boolean smokeDetected = false;
boolean flameDetected = false;

void setup() {
  // Initialize pin modes
  pinMode(buzzer, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(smokeA0, INPUT);
  pinMode(flameSensorPin, INPUT);

  // Initialize ultrasonic sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Initialize servo motor
  myServo.attach(servoPin); // Attach the servo to the specified pin
  myServo.write(180); // Move the servo to 180 degrees initially

  // Serial communication
  Serial.begin(9600);      // Serial monitor communication
  Serial.println("Sensors warming up!");
  delay(20000);            // Allow the MQ-2 to warm up
  noTone(buzzer);
}

void loop() {
```

```
  // Reading from smoke sensor
  sensorValue = analogRead(smokeA0);
  int flameValue = digitalRead(flameSensorPin);

  // Ultrasonic sensor - calculate distance
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2; // Calculate distance in cm

  // Print the distance for debugging
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  // Smoke and flame detection
  if (sensorValue > 300 || flameValue == LOW) {
    digitalWrite(led, HIGH);
    tone(buzzer, 1000);   // Activate the buzzer with a constant tone

    if (!smokeDetected && sensorValue > 300) {
      Serial.println("Smoke detected!");
      smokeDetected = true;
      myServo.write(90); // Move the servo to 90 degrees
    }

    if (!flameDetected && flameValue == LOW) {
      Serial.println("Flame detected!");
      flameDetected = true;
      myServo.write(90); // Move the servo to 90 degrees
    }
  } else {
    digitalWrite(led, LOW);
    noTone(buzzer);
    smokeDetected = false;
    flameDetected = false;
    myServo.write(180); // Move the servo back to 180 degrees if no smoke/flame detected
  }

  // Check if the ultrasonic sensor detects an object closer than 4 inches (10.16 cm)
  if (distance < 10.16) {
    Serial.println("Object detected within 4 inches (10.16 cm)!");
    // Buzzer sounds
    for (int i = 0; i < 2; i++) {
      tone(buzzer, 2000);  // Start beeping
      delay(200);          // Beep duration
      noTone(buzzer);      // Stop beeping
      delay(300);          // Delay before the next beep
    }
  }

  delay(500); // wait 0.5s for the next reading
}
```

(iv) Testing and Debugging

Testing: The system was tested under different scenarios (gas leak, fire, intrusion). Each sensor was calibrated, and the entire system was monitored for correct responses.

Debugging: Issues related to sensor sensitivity and false triggers were resolved by adjusting sensor thresholds and fine-tuning the code to ensure proper functionality.

(v) Bug Report

Servo Malfunction: During initial testing, the servo motor failed to open the door due to incorrect wiring. The issue was resolved by reviewing the circuit connections.

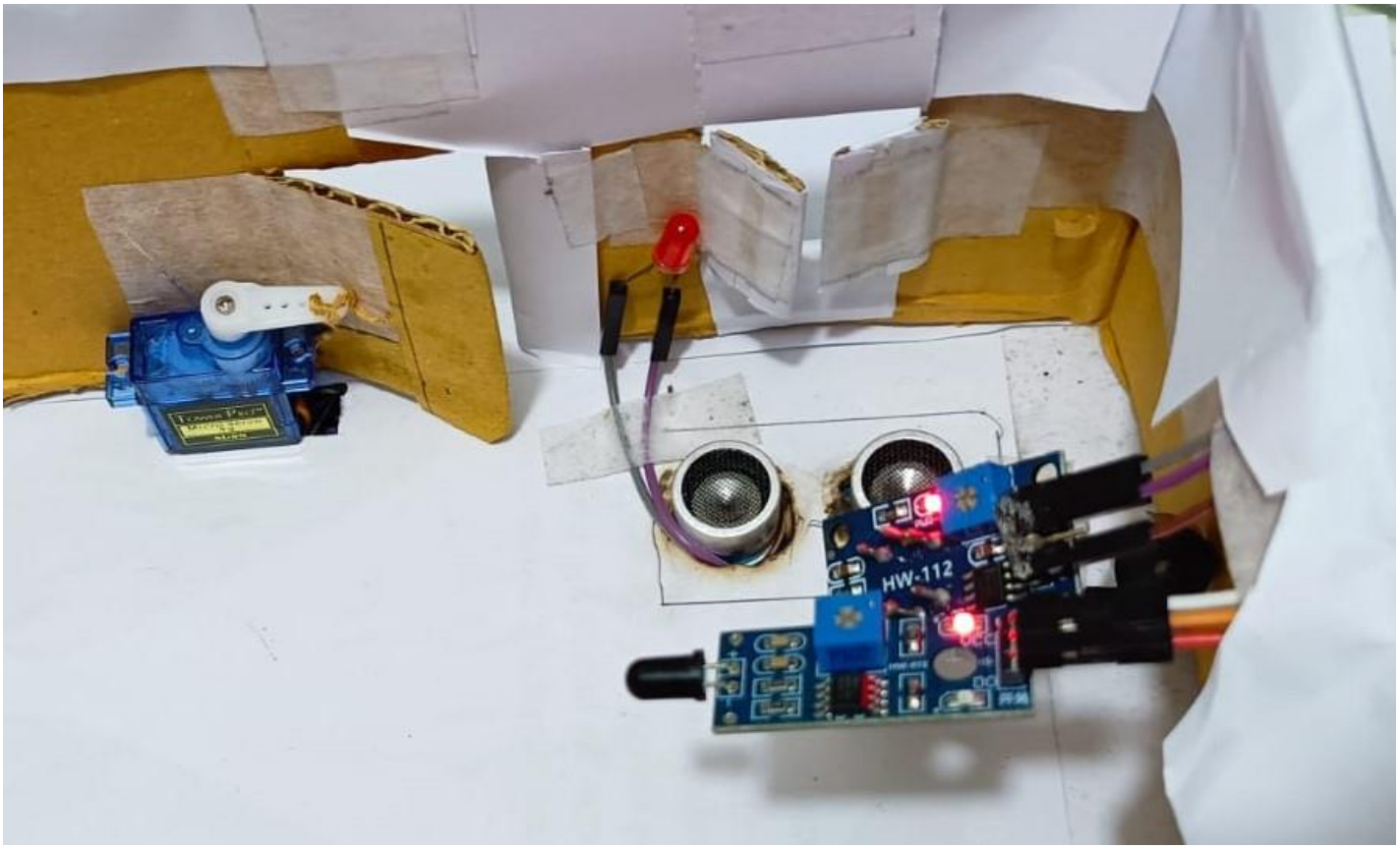
False Alarms: Some sensors triggered false alarms due to high sensitivity. This was corrected by adjusting the sensor thresholds in the code.

Output Control: The servo motor opens the door, the buzzer sounds alarms, and the LED flashes based on detected conditions.

5. Output & Pictures

Outputs: The system successfully opens the door upon gas or fire detection, alerts homeowners through buzzer and LED, and detects intrusions.

Pictures: Images of the prototype, showing the components assembled on the breadboard and the system in action.


HW-112

6. Challenges Faced

Sensor Sensitivity: The gas and flame sensors had varying sensitivity, which led to false alarms. This was solved by refining sensor calibration and thresholds in the code.

Power Supply: Ensuring a stable power supply was critical, especially when operating multiple components like the servo motor and buzzer simultaneously.

7. Future Scope

Adding GSM Module: Integrating a GSM module to send SMS alerts to homeowners in case of an emergency.

Wireless Expansion: Although wireless options were avoided in this project, future versions could use Wi-Fi or Bluetooth for remote control.

Camera Integration: Adding a camera for real-time monitoring.

8. References

1) Arduino Documentation:
https://www.arduino.cc/en/Guide/Introduction Collected on: 08-11-2024

2) MQ2 Gas Sensor Datasheet:
https://components101.com/sites/default/files/component_datasheet/MQ2%20Gas%20Sensor.pdf Collected on: 08-11-2024

3) Flame Sensor Documentation:
https://components101.com/sensors/flame-sensor-module

Collected on: 08-11-2024

4) Ultrasonic Motion Sensor Documentation:
https://components101.com/sensors/hc-sr04-ultrasonic-sensor

Collected on: 08-11-2024

9. Conclusion

This project successfully demonstrates an integrated smart home security and safety system. By combining various sensors and actuators, the system not only enhances home security but also provides a safe exit route in emergencies. The use of affordable and open-source hardware makes it scalable and customizable for further enhancements.