Lab Report for Digital Logic Design Project

Project Title: Flame Guard: An IoT-Enabled Fire Alarm System
Course Name : Digital Logic Design Lab
Course Code : CSE0613204
Trimester : Fall 2025
Section : A
Instructor : Hasibul Hasan Rupok
Submission Date : 15 December, 2025
Team Members :

Student ID	Name
2233081299	MD. MONIRUL ALAM SIFAT
2243081167	ABU HASAN
2243081175	TANJINA AHMED
2243081180	SANJIDA AKTER
2243081034	RAFI AKANDA
2243081014	ISRAT JAHAN JUMI

Table of Contents

1. **Abstract**
2. **Introduction**
3. **Objectives**
4. **Theory / Background Study**
5. **System Design**
    - 5.1 Problem Analysis
    - 5.2 Truth Table / State Table
    - 5.3 Boolean Expression / Logic Condition
    - 5.4 Circuit Diagram / Logic Diagram
    - 5.5 Simulation Setup
6. **Results & Discussion**
    - 6.1 Simulation Results
    - 6.2 Hardware Implementation
    - 6.3 Discussion
7. **Conclusion**
8. **References**
9. **Appendix**

   9.1 Arduino Source Code

1. Abstract

This project presents the design and implementation of an IoT-enabled fire alarm and monitoring system named **Flame Guard**. The system is designed to detect fire-related hazards using smoke/gas, temperature, and infrared (IR) flame sensors. An Arduino Uno microcontroller processes sensor data and triggers visual and audible alarms when dangerous conditions are detected. The system also sends alert messages via a Bluetooth module for remote monitoring. The project demonstrates a reliable, low-cost, and real-time fire safety solution suitable for homes and small facilities.

2. Introduction

Fire accidents pose a serious threat to life and property, especially in residential and industrial environments. Early detection is crucial to minimize damage and ensure safety. Traditional fire alarm systems often rely on a single sensing mechanism, which may lead to false alarms or delayed detection.

The **Flame Guard** system overcomes these limitations by combining multiple sensors—smoke/gas, temperature, and IR flame detection—to improve accuracy and reliability. The integration of Bluetooth communication allows real-time alerts to be sent to nearby mobile devices, enhancing user awareness and response time.

3. Objectives

The main objectives of this project are:

- To design a fire alarm system using multiple sensors for improved detection accuracy.
- To implement the system using Arduino as a microcontroller platform.
- To analyze sensor data and apply logical conditions for alarm triggering.
- To provide visual (LED) and audible (buzzer) alerts during fire conditions.
- To enable wireless alert transmission using a Bluetooth module.
- To test and verify the system through simulation and hardware implementation.

4. Theory / Background Study

The system is based on basic digital logic concepts and sensor interfacing techniques. Each sensor provides input signals that are processed using conditional logic inside the microcontroller.

Sensors Used:

- **MQ Gas Sensor:** Detects smoke and combustible gases.
- **DHT11 Temperature Sensor:** Measures ambient temperature.
- **IR Flame Sensor:** Detects infrared radiation emitted by fire.

Logic Concept:

The system follows a logical OR condition:

Alarm = Smoke Detected OR High Temperature OR Flame Detected

If any one of the conditions is true, the alarm state is activated.

5. System Design

5.1 Problem Analysis

The system continuously monitors environmental conditions to detect fire hazards. The inputs are analog and digital signals from sensors, and the outputs are LEDs, a buzzer, and Bluetooth alert messages.

- **Inputs:** Smoke value, temperature value, IR sensor signal
- **Outputs:** Green LED (safe), Red LED (alarm), buzzer, Bluetooth alert

The system behaves as a combinational logic system where the output depends on current sensor readings.

5.2 Truth Table / State Table

Smoke	Temperature	IR Sensor	Alarm State
0	0	0	OFF
1	0	0	ON
0	1	0	ON
0	0	1	ON
1	1	1	ON

(1 = Hazard detected, 0 = Normal)

5.3 Boolean Expression / K-map Simplification

Let:

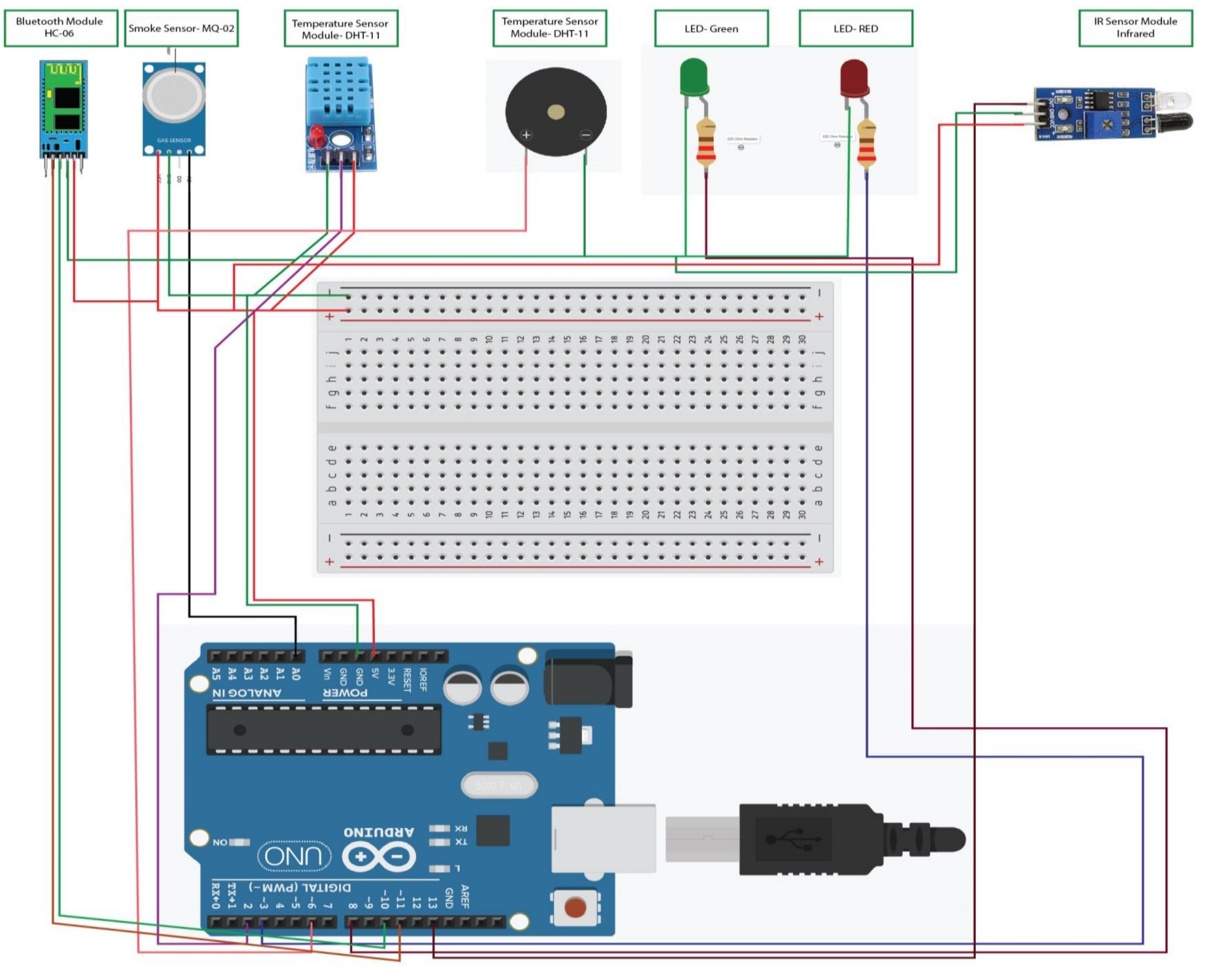
- S = Smoke detected
- T = High temperature
- F = Flame detected

Alarm (A) = S + T + F

This expression indicates that the alarm is activated if any sensor detects danger.

5.4 Circuit Diagram / Logic Diagram

Circuit Diagram



5.5 Simulation Setup

- **Software Used:** Arduino IDE (for code verification)
- **Testing Method:** Serial Monitor output and sensor value observation
- **Logic Verification:** Conditional checking within Arduino code

6. Results & Discussion

6.1 Simulation Results

The system was tested by simulating different sensor conditions. When smoke concentration, temperature, or flame detection exceeded predefined thresholds, the alarm state was triggered successfully.

6.2 Hardware Implementation

The hardware was implemented using an Arduino Uno, sensors, LEDs, buzzer, and Bluetooth module on a breadboard. The system worked reliably under real-world testing conditions.

- Green LED remained ON during safe conditions.
- Red LED and buzzer activated during fire detection.
- Bluetooth alerts were successfully transmitted.

6.3 Discussion

The project performed as expected with accurate detection and timely alerts. Minor issues such as sensor calibration and wiring stability were resolved during testing. The multi-sensor approach significantly reduced false alarms compared to single-sensor systems.

7. Conclusion

The **Flame Guard** fire alarm system successfully demonstrates an effective and low-cost solution for early fire detection. Through this project, practical knowledge of digital logic, sensor interfacing, and microcontroller programming was gained. The system can be further enhanced by integrating Wi-Fi, GSM alerts, or automatic fire suppression mechanisms.

8. References

- Arduino Official Documentation
- DHT11 Sensor Datasheet
- MQ Gas Sensor Datasheet
- YouTube tutorials on Arduino fire alarm systems

9. Appendix

9.1: Arduino Source Code

```
#include <SoftwareSerial.h>
#include <DHT.h>

// --- PIN DEFINITIONS ---
#define GREEN_LED 8
#define RED_LED 3
#define BUZZER 6

#define GAS_SENSOR_PIN A0
#define DHT_PIN 2
#define IR_SENSOR_PIN 13
#define DHTTYPE DHT11

#define BT_RX_PIN 10
#define BT_TX_PIN 11

// --- THRESHOLDS ---
const int SMOKE_THRESHOLD = 150;
const float TEMP_THRESHOLD = 30.0;
const int IR_DETECTED_SIGNAL = LOW;

// --- OBJECTS ---
DHT dht(DHT_PIN, DHTTYPE);
SoftwareSerial BTSerial(BT_RX_PIN, BT_TX_PIN);

void setup() {
  Serial.begin(9600);
  BTSerial.begin(9600);

  pinMode(GREEN_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(IR_SENSOR_PIN, INPUT);

  dht.begin();

  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(RED_LED, LOW);
  digitalWrite(BUZZER, LOW);
}

void loop() {
  float t = dht.readTemperature();
  int smokeValue = analogRead(GAS_SENSOR_PIN);
  int irValue = digitalRead(IR_SENSOR_PIN);

  if (isnan(t)) {
    t = 25.0;
  }

  if (smokeValue > SMOKE_THRESHOLD || t > TEMP_THRESHOLD || irValue ==
IR_DETECTED_SIGNAL) {
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(RED_LED, HIGH);
    tone(BUZZER, 1000, 250);
```

```
    String alertMessage = "!!! FIRE ALERT !!! ";
    alertMessage += "Smoke: " + String(smokeValue);
    alertMessage += " | Temp: " + String(t) + "C";

    if (irValue == IR_DETECTED_SIGNAL) {
      alertMessage += " | FLAME DETECTED";
    }

    Serial.println(alertMessage);
    BTSerial.println(alertMessage);

    delay(300);
    noTone(BUZZER);
    delay(200);
  } else {
    digitalWrite(GREEN_LED, HIGH);
    digitalWrite(RED_LED, LOW);
    noTone(BUZZER);

    String statusMessage = "Status: SAFE | Smoke: " + String(smokeValue);
    statusMessage += " | Temp: " + String(t) + "C";
    statusMessage += String(" | IR: ") + (irValue == HIGH ? "Clear" : "Object");

    Serial.println(statusMessage);
    delay(1500);
  }
}
```