Design and Implementation of a Low-Cost, Switch-Controlled Fire Alarm System

Created by: Aneela Musarrat

Date: 19/ 12/ 2022

Abstract

Fire detection and prevention systems are essential for protecting life and property. This project aims to design and demonstrate a fire alarm system using low-cost, easily accessible components such as an Arduino Uno, MQ-2 smoke sensor, buzzer, LED indicators, and multiple switches to control and simulate system behavior. The alarm system continuously monitors the environment for smoke using the MQ-2 sensor. When smoke exceeds a predefined threshold, the system activates a buzzer and changes LED indicators to signify danger. It features a power switch to turn the system ON/OFF, a test switch to simulate fire detection manually, and a reset switch to turn off the alarm after activation. This project not only serves as a practical application of sensor technology and microcontroller interfacing but also as an educational demonstration of real-world safety systems.

Introduction

Fire is one of the most dangerous hazards in homes, offices, and industrial environments. Rapid detection of fire or smoke can prevent disaster. While commercial fire alarm

systems are available, they are often expensive and complex. This project focuses on building a simple, cost-effective fire alarm system suitable for small spaces and educational demonstrations. The prototype uses an Arduino Uno as its controller and features an MQ-2 smoke sensor to detect gas and smoke in the environment. It also includes three distinct switches: a **Power Switch**, a **Test Switch**, and a **Reset Switch** to allow manual control and simulation of fire events. Audible and visual outputs—via a buzzer and LEDs—inform users of the fire detection status.

Objective

- Design a low-cost fire detection system.
- Understand interfacing of smoke sensors and control switches with a microcontroller.
- Allow manual simulation and reset of fire conditions.
- Alert users via buzzer and visual indicators.
- Create a reliable, real-time embedded system for educational and safety demonstration purposes.

Components and Materials Used

Component	Quantity	Description
Arduino Uno	1	Microcontroller for control logic

MQ-2 Smoke Sensor	1	Detects smoke and gases
Buzzer	1	Sounds alarm upon detection
Red LED	1	Indicates fire detection
Green LED	1	Indicates normal status
Power Switch	1	Toggles main system power
Test Switch	1	Simulates a fire condition
Reset Switch	1	Manually stops alarm
Resistors (220Ω)	3	For LEDs and safety
Breadboard	1	For prototype wiring
Jumper Wires	Several	For connections
USB/Battery Power	1	Powers Arduino

Supply

System Design and Switch Functionality

Power Switch

- **Type**: Toggle switch or push-button
- **Connection**: Inline with Arduino's VIN or digital pin
- **Function**: Activates or deactivates the entire circuit. When off, the system consumes no power and does not sense or react.

Test Switch

- **Type**: Normally open push-button
- **Connection**: Digital pin with pull-down resistor
- **Function**: Simulates fire detection by sending a HIGH signal to the microcontroller. Useful for demonstration without real smoke.

Reset Switch

- **Type**: Normally open push-button

- **Connection**: Digital input pin

- **Function**: Resets the system to its initial “safe” state, turning off the buzzer and red LED and turning on the green LED.

These three switches provide manual control and make the prototype flexible for classroom demonstration and testing without relying on real smoke.

Working Principle

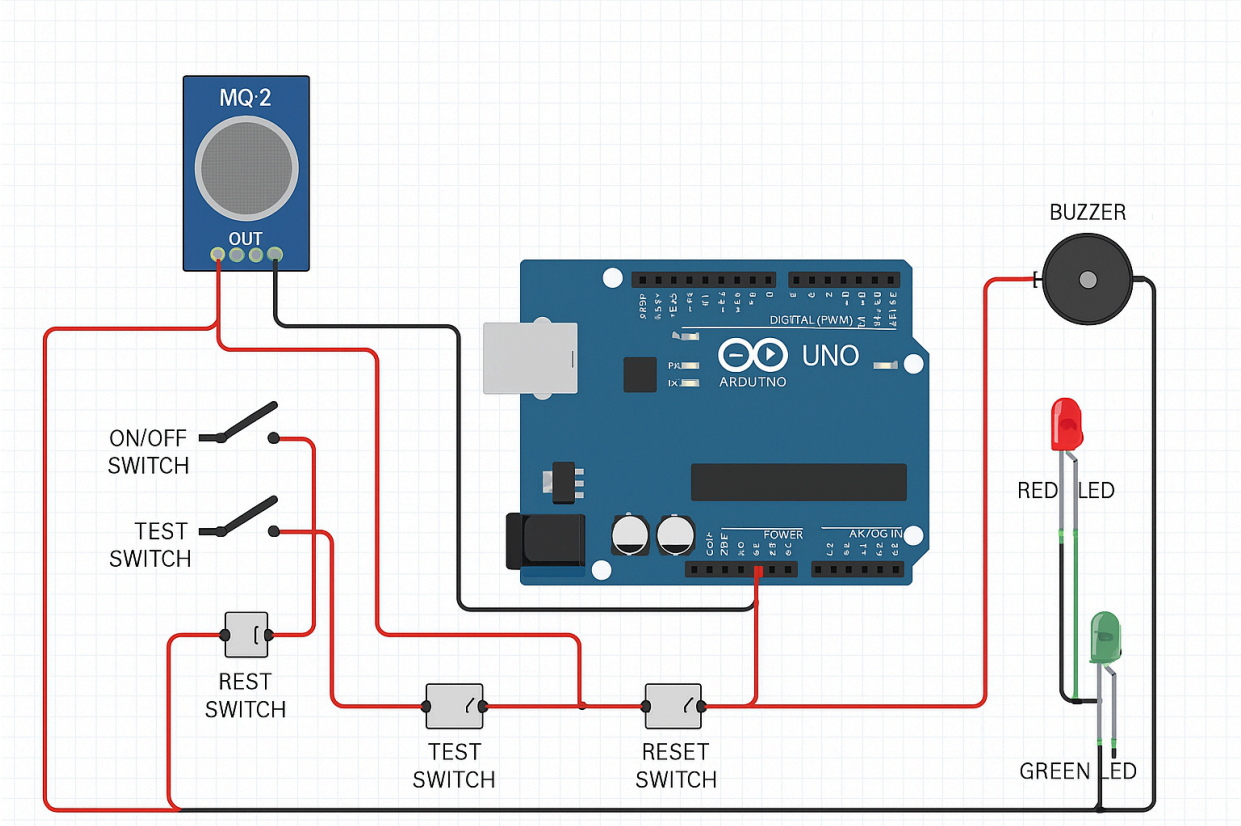
1. **Initialization**:
   When the power switch is turned ON, the Arduino begins reading input from the smoke sensor and test switch.

2. **Normal State**:
   - Smoke levels below threshold.
   - Green LED is ON, Red LED is OFF, Buzzer is OFF.

Fire Detected: If the MQ-2 sensor detects smoke levels above a calibrated threshold OR if the test switch is pressed:

- Red LED turns ON
- Green LED turns OFF
- Buzzer is activated to alert people

Circuit Diagram



The smoke sensor is connected to analog pin A0. The LEDs and buzzer are connected to digital output pins. All switches are connected to digital input pins with pull-down resistors to prevent floating signals.

Explanation with code

```
int smokeSensor = A0;
int redLED = 8;
```

```
int greenLED = 9;
int buzzer = 10;
int testSwitch = 2;
int resetSwitch = 3;

void setup() {
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(testSwitch, INPUT);
  pinMode(resetSwitch, INPUT);
  Serial.begin(9600);
}

void loop() {
  int smoke = analogRead(smokeSensor);
  int test = digitalRead(testSwitch);
  int reset = digitalRead(resetSwitch);

  if (smoke > 300 || test == HIGH) {
    digitalWrite(redLED, HIGH);
    digitalWrite(greenLED, LOW);
    digitalWrite(buzzer, HIGH);
  }

  if (reset == HIGH) {
    digitalWrite(redLED, LOW);
    digitalWrite(greenLED, HIGH);
    digitalWrite(buzzer, LOW);
```

```
  }
}
```

Testing and Demonstration

Test Condition	Expected Result
Power ON	Green LED ON, system idle
Simulated Smoke (Test Button)	Red LED + Buzzer ON
Real Smoke (e.g., matchstick)	Red LED + Buzzer ON
Reset Button Pressed	System returns to green LED, buzzer OFF

Tests were repeated multiple times with different stimuli (air, smoke, button presses). The system reliably responded and returned to idle using the reset switch.

Applications

- Home safety systems
- School and college lab demos
- Educational IoT/embedded system projects
- Portable smoke alarms for small areas
- Temporary alert systems at events

Advantages

- Low-cost and reliable
- Real-time fire detection
- Easy to test using manual switches
- Portable and modular design
- Good for learning electronics and coding

Limitations

- No internet/mobile alerting system
- False positives possible from dust or cooking
- No flame or heat detection without add-ons
- Manual reset required
- Power-dependent (no battery backup)

Future Scope

- GSM or Wi-Fi integration for remote alerts
- Use of flame and temperature sensors for reliability

- Solar-powered version with battery backup
- Integration with cloud dashboard for remote monitoring
- Mobile app for alerts and status control

Conclusion

This project demonstrates how simple hardware and microcontrollers can be used to build a reliable, cost-effective fire alarm system. By integrating switches, smoke sensors, and LEDs, we gain control over both automatic and manual simulation of alarm conditions. The system is functional, easy to build, and can be extended with more advanced features in the future.