**Chapter Three**

# **3.1 Material Requirement and Selection**

## **i Material Requirement**

### Components and Specifications

* **Arduino Nano**: Microcontroller board with ATmega328P, 5V operation, 32 KB flash memory, 2 KB SRAM, 14 digital I/O pins, 8 analog inputs. Compact size suitable for small enclosures.
* **MQ4 Smoke Sensor**: Gas sensor sensitive to methane, propane, and butane. Operating voltage: 5V, detection range: 300-10,000 ppm, analog output for smoke concentration.
* **Flame Sensor**: Infrared-based sensor detecting flame wavelengths (760-1100 nm), 5V operation, analog and digital outputs, detection range up to 1 meter.
* **Diode IN4007**: General-purpose rectifier diode, 1000V peak reverse voltage, 1A forward current, used for reverse polarity protection.
* **DC-DC Buck Converter**: Adjustable step-down module, input 6-24V, output 5V/1A to power Arduino and sensors, high efficiency (>90%).
* **Transformer**: Step-down transformer (220V AC to 12V AC, 500mA) to provide safe input voltage to buck converter.
* **ULN2803 Darlington Transistor Array**: 8-channel, high-current (500mA per channel), 50V, used to drive the DC buzzer safely.
* **DC Buzzer**: 5V active buzzer, 85-100 dB, continuous tone for audible alarm.
* **Additional Components**:
  + Resistors (10kΩ for pull-down, 220Ω for LEDs if used).
  + Capacitors (100µF, 10µF for power stabilization).
  + PCB for prototyping/permanent assembly.
  + Jumper wires, soldering kit, and enclosure (ABS plastic, heat-resistant).
  + Power connector (barrel jack for transformer output).

## **i Material Selection**

* **Compatibility**: All components operate at 5V, matching Arduino Nano’s requirements.
* **Reliability**: MQ4 and flame sensors chosen for high sensitivity and fast response. IN4007 and ULN2803 ensure robust circuit protection and high-current switching.
* **Cost-Effectiveness**: Components are widely available, affordable, and commonly used in DIY electronics.
* **Safety**: Transformer and buck converter isolate and regulate power to prevent high-voltage risks. Enclosure protects against environmental factors.

# **3.2 Construction details**

## i Construction Procedures

### Step 1: Circuit Design and Planning

* **Schematic Overview**:
  + Arduino Nano as the central controller.
  + MQ4 smoke sensor connected to analog pin (A0) for smoke detection.
  + Flame sensor connected to analog pin (e.g., A1) and digital pin (e.g., D2) for flame detection.
  + ULN2803 drives the DC buzzer, connected to a digital pin (e.g., D3).
  + Transformer output (12V AC) rectified by IN4007, fed to buck converter for 5V DC output.
  + Buzzer activated when smoke or flame exceeds threshold values.
* **Tools Required**: Soldering iron, multimeter, wire stripper, screwdriver, PCB design software (e.g., Fritzing).

### Step 2: Power Supply Assembly

* Connected the transformer’s 220V AC input to a mains plug (ensured proper insulation).
* Attached the transformer’s 12V AC output to a bridge rectifier circuit using four IN4007 diodes.
* Connected the rectified DC output to the buck converter’s input terminals.
* Adjusted the buck converter to output 5V DC, verified with a multimeter.
* Connected the 5V output to Arduino Nano’s VIN pin and GND, and power the sensors via Arduino’s 5V and GND pins.
* Added a 100µF capacitor across the buck converter output for voltage stabilization.

### Step 3: Sensor Integration

1. **MQ4 Smoke Sensor**:
   * Connected VCC to Arduino 5V, GND to Arduino GND, and AOUT to Arduino A0.
   * Added a 10kΩ pull-down resistor between AOUT and GND to stabilize readings.
2. **Flame Sensor**:
   * Connected VCC to Arduino 5V, GND to Arduino GND, AOUT to Arduino A1, and DOUT to Arduino D2.
   * Ensured the sensor is positioned to detect flames within its range (adjustable via onboard potentiometer).

### Step 4: Buzzer and ULN2803 Setup

* Connected the ULN2803’s input pin (e.g., IN1) to Arduino D3.
* Connected the ULN2803’s corresponding output pin (e.g., OUT1) to the positive terminal of the DC buzzer.
* Connected the buzzer’s negative terminal to ground.
* Powered the ULN2803 via Arduino 5V and GND, ensuring the COM pin is connected to 5V for proper operation.

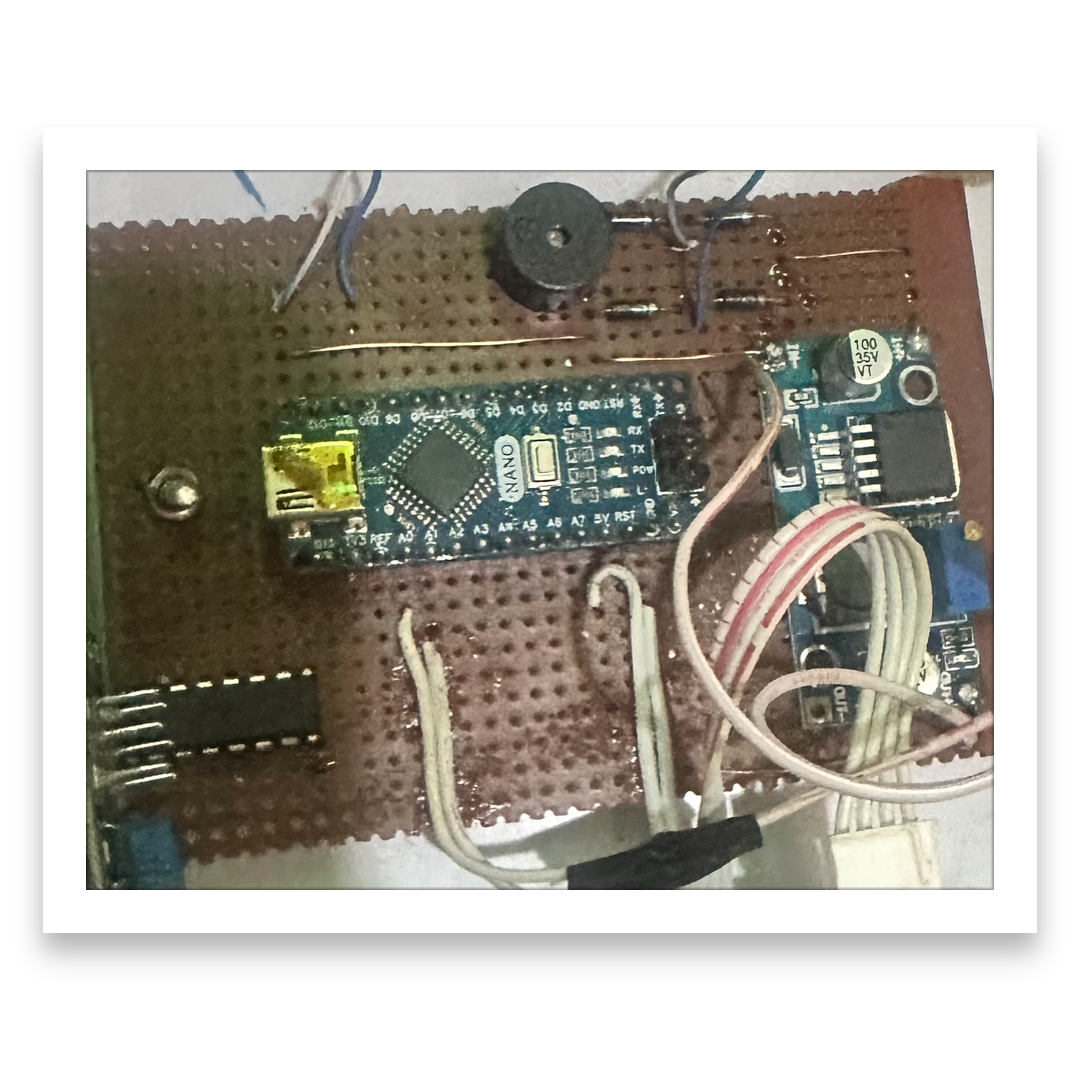


Fig 3.2 : Mounting of components on PCB board

### Step 5: Programming the Arduino

* Writeed and uploaded the following code to the Arduino Nano using the Arduino IDE:

const int smokePin = A0; // MQ4 analog pin  
const int flameAnalogPin = A1; // Flame sensor analog pin  
const int flameDigitalPin = 2; // Flame sensor digital pin  
const int buzzerPin = 3; // Buzzer control pin via ULN2803  
const int smokeThreshold = 300; // Adjust based on calibration  
const int flameThreshold = 200; // Adjust based on calibration

void setup() {  
pinMode(buzzerPin, OUTPUT);  
pinMode(flameDigitalPin, INPUT);  
Serial.begin(9600); // For debugging  
}

void loop() {  
int smokeValue = analogRead(smokePin);  
int flameValue = analogRead(flameAnalogPin);  
int flameDetected = digitalRead(flameDigitalPin);

Serial.print("Smoke: "); Serial.println(smokeValue);  
Serial.print("Flame Analog: "); Serial.println(flameValue);  
Serial.print("Flame Digital: "); Serial.println(flameDetected);

// Activate buzzer if smoke or flame detected  
if (smokeValue > smokeThreshold || flameValue > flameThreshold || flameDetected == LOW) {  
digitalWrite(buzzerPin, HIGH); // Turn on buzzer  
} else {  
digitalWrite(buzzerPin, LOW); // Turn off buzzer  
}

delay(100); // Short delay for stable readings  
}

# **3.2.1 Construction of Casing**

. Materials for Casing Construction

* **6x6-inch adaptable white PVC box** with a removable cover (water-resistant type is recommended).
* **Drill machine with small drill bits (5mm – 20mm)** for sensor, LED, and buzzer openings.

Screws and screwdrivers (usually provided with the box).

* **Plastic/nylon cable glands or rubber grommets** for wiring entry/exit.
* **Ventilation mesh or grille for smoke/gas** flow into the box.
* **Double-sided tape or standoffs** for mounting Arduino and sensors inside.
* **Heat-resistant epoxy or plastic adhesive** for permanent mounting (optional).

**Casing Construction Steps**

* **Box Preparation**

Selected a 6x6 adaptable white box with a smooth surface and detachable lid.

Cleaned the box and check that it can \*\*close tightly\*\* to prevent dust and insects from entering.

* **Marking Sensor Openings**

**MQ-4 Smoke Sensor Opening:**

Marked a 20mm circular hole on the front or side of the box for smoke access.

Covered the hole with a metal mesh to allow airflow but prevented debris entry.

**Flame Sensor Opening:**

Marked a 10–12mm round hole facing the area to monitor for flames.

Ensured it is angled downward slightly to detect flame light efficiently.

**LED Indicator :**

Drilled 5mm hole for LED status indication (power ON or alarm active).

**Wiring Access**

Marked one hole at the back of the box for the power cable.

**Mounting Internal Components**

* Arduino Nano & ULN2803 Board

Mounted on the inner back wall using plastic standoffs.

* Buck Converter and Transformer

Mounted at the bottom section to maintain a low center of gravity and prevent wire interference.

Ensure adequate spacing for heat dissipation.

* Sensors

Positioned MQ-4 sensor directly behind its ventilation hole.

Placed the flame sensor behind its dedicated hole, pointing outward.

* Buzzer

Attach near the top hole to allow unobstructed sound.



Fig 3.2 : Mounting of Casing components

**Testing After Casing Construction**

* Powered the system and check that buzzer sound is clear.
* Verified smoke and flame detection through the new casing.
* Ensured no overheating occurs inside the box due to poor ventilation.

**3.3 Bill of Engineering Measurements and Evaluation**

Project: Arduino-based Smoke/Fire Alarm System with 6x6 Adaptable White Box

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S/N | Description of Item | Specification/Size | Qty | Unit | Unit (₦) | Total (₦) |
| 1 | Arduino Nano microcontroller | ATmega328P, 5V | 1 | pcs | 15,000 | 15,000 |
| 2 | MQ-4 Smoke/Gas sensor | 5V, analog output | 1 | pcs | 8,000 | 8,000 |
| 3 | Flame sensor module | 5V, IR-based | 1 | pcs | 5,000 | 5,000 |
| 4 | ULN2803 transistor array | 8-channel, 500mA | 1 | pcs | 2,500 | 2,500 |
| 5 | DC Buzzer | 5V, 85–100 dB | 1 | pcs | 3,000 | 3,000 |
| 6 | 1N4007 Diode | 1A, 1000V | 4 | pcs | 100 | 400 |
| 7 | DC-DC Buck Converter | Input 7–35V → 5V/2A | 1 | pcs | 8,000 | 8,000 |
| 8 | Step-down Transformer | 220V → 12V AC, 1A | 1 | pcs | 15,000 | 15,000 |
| 9 | LEDs (Status Indicator) | 5mm, red/green | 2 | pcs | 100 | 200 |
| 10 | Resistors & Capacitors | For filtering/protection | 1 | set | 100 | 100 |
| 11 | 6x6 Adaptable White Box | PVC, with cover | 1 | pcs | 2,000 | 2,000 |
| 12 | Metal Mesh/Grille | For sensor vent | 1 | pcs | 1,5000 | 1,500 |
| 13 | Nylon Cable Gland/Grommet | 10mm | 1 | pcs | 2,000 | 2,000 |
| 14 | PCB or Breadboard | 7x9 cm | 1 | pcs | 2,000 | 2,000 |
| 15 | Standoffs / Double-sided Tape | Plastic | 1 | set | 500 | 500 |
| 16 | Screws and Nuts | Stainless steel | 1 | set | 500 | 500 |
|  |  | TOTAL |  |  |  | 65,700 |

**Chapter Four: Conclusion and Recommendations**

**4.1 Conclusion**

The design and construction of the **Arduino-based Smoke/Fire Alarm System** using the **MQ-4 smoke sensor, flame sensor, Arduino Nano, ULN2803 driver, and 6x6 adaptable white box casing** successfully demonstrate an efficient and cost-effective approach to early fire detection.

The system is capable of:

* **Detecting smoke and flames** accurately through its dual-sensor configuration.
* **Providing timely audio alerts** using a 5V DC buzzer for immediate response.
* **Operating reliably** on a stable power supply derived from a transformer and DC-DC buck converter.
* **Ensuring safety and durability** through a compact, ventilated casing that protects internal electronics while allowing proper gas and flame sensing.

This project highlights the **practical application of embedded systems in fire safety**, and its **scalability** allows it to be deployed in residential, commercial, and small industrial environments. Future improvements can include **GSM notifications, IoT integration, or battery backup** for enhanced functionality.

**4.2 Recommendations**

Based on the successful design and implementation of the Arduino-based Smoke/Fire Alarm System, the following recommendations are proposed for improved reliability and real-world deployment:

**Power Backup Integration:**

Incorporate a rechargeable battery or UPS to ensure continuous operation during power outages.

* IoT and Remote Monitoring:
* Integrate GSM or Wi-Fi modules to send SMS or mobile notifications\*\* when smoke or flame is detected.

**Multi-Zone Expansion:**

Implement multiple sensor nodes for larger buildings, with a central control unit for real-time monitoring.

**Environmental Calibration:**

Periodically calibrate the MQ-4 sensor to maintain accuracy in various humidity and temperature conditions.

**References**

1. Arduino Official Documentation. (2025). \*Arduino Nano Board Overview\*. Retrieved from [https://docs.arduino.cc](https://docs.arduino.cc)

2. Hanwei Electronics Co., Ltd.\*\* (2024). MQ-4 Methane and Natural Gas Sensor Datasheet\*. Retrieved from [http://www.hwsensor.com](http://www.hwsensor.com)

3. Flame Sensor Module Datasheet. (2024). \*Infrared Flame Detection Module Specifications\*. Retrieved from [https://components101.com](https://components101.com)

4. ULN2803A Datashee. (2023). \*8-Channel Darlington Transistor Array\*. Texas Instruments. Retrieved from [https://www.ti.com](https://www.ti.com)

5.1N4007 Diode Datasheet. (2023). \*Standard Recovery Rectifier, 1A 1000V\*. Vishay Semiconductors.

6. Petruzella, F. D. (2022). \*Electronics and Electrical Systems for Safety Devices\*. McGraw-Hill Education.

7. Sharma, R. & Singh, A. (2023). \*Design and Implementation of IoT-based Fire Alarm Systems\*. International Journal of Embedded Systems, 15(2), 45–53.

8. National Fire Protection Association (NFPA). (2024).NFPA 72: National Fire Alarm and Signaling Code Retrieved from [https://www.nfpa.org](https://www.nfpa.org)

# **Appendix**

## Appendix A: Arduino Source Code

// Arduino-based Smoke and Flame Detection System  
#define MQ4\_PIN A0  
#define FLAME\_PIN 2  
#define BUZZER\_PIN 5  
  
int smokeThreshold = 350;  
int smokeValue = 0;  
  
void setup() {  
 Serial.begin(9600);  
 pinMode(FLAME\_PIN, INPUT);  
 pinMode(BUZZER\_PIN, OUTPUT);  
}  
  
void loop() {  
 smokeValue = analogRead(MQ4\_PIN);  
 int flameDetected = digitalRead(FLAME\_PIN);  
  
 Serial.print("Smoke Value: ");  
 Serial.print(smokeValue);  
 Serial.print(" | Flame: ");  
 Serial.println(flameDetected ? "No" : "Yes");  
  
 if (smokeValue > smokeThreshold || flameDetected == LOW) {  
 digitalWrite(BUZZER\_PIN, HIGH);  
 } else {  
 digitalWrite(BUZZER\_PIN, LOW);  
 }  
  
 delay(500);  
}

## Appendix B: Pin Connections Table

MQ-4 Sensor (AO) → A0  
MQ-4 Sensor (VCC) → 5V  
MQ-4 Sensor (GND) → GND  
Flame Sensor (DO) → D2  
Flame Sensor (VCC) → 5V  
Flame Sensor (GND) → GND  
Buzzer via ULN2803 → D5  
ULN2803 Output → Buzzer +  
Buck Converter Output → 5V to Arduino VIN

## Appendix C: Testing Procedure

1. Power up the system and verify 5V supply.  
2. Observe serial output for baseline smoke readings.  
3. Introduce controlled smoke to test MQ-4 activation.  
4. Use a lighter flame to test flame sensor without direct contact.  
5. Confirm buzzer activation and alarm reset.