

# 1. Project Overview

This project implements a **smart fire detection system** using a **Raspberry Pi** combined with multiple sensors, **AI-based risk prediction**, **real-time IoT monitoring (Blynk)**, and **SMS alerts (SIM800L GSM)**.

The system continuously monitors:

- 🔥 Flame presence
- ⚛ Temperature
- 🌡 Smoke/Gas concentration (MQ-2)

When a fire risk is detected (by sensors or AI), the system:

- Activates a **buzzer alarm**
  - Sends an **SMS alert**
  - Updates live data on **Blynk mobile app**
- 

# 2. System Architecture

## Hardware Components

- **Raspberry Pi**
- **MQ-2 Smoke/Gas Sensor** (via MCP3008 ADC)
- **DS18B20 Temperature Sensor**
- **Flame Sensor**
- **SIM800L GSM Module**
- **Buzzer**
- **MCP3008 ADC**
- **External Power Supply**

## Software Components

- Python 3
  - RPi.GPIO
  - Adafruit MCP3008 Library
  - NumPy
  - Joblib (AI model)
  - Blynk IoT Platform
  - Multithreading
-

### 3. Features

- ✓ Multi-sensor fire detection
  - ✓ AI-based fire risk prediction
  - ✓ Real-time IoT dashboard (Blynk)
  - ✓ SMS alerts with cooldown protection
  - ✓ Buzzer alarm
  - ✓ Threaded architecture (non-blocking)
  - ✓ Automatic MQ-2 calibration
- 

### 4. Blynk Configuration

```
BLYNK_AUTH = "PUT_YOUR_TOKEN_HERE"
```

#### Virtual Pins Used:

Virtual Pin	Data
V0	Temperature (°C)
V1	MQ-2 Smoke Value
V2	Flame Status
V3	AI Risk Probability
V4	Alert Status

---

### 5. Configuration Parameters

```
FLAME_PIN = 17
BUZZER_PIN = 27
SERIAL_PORT = "/dev/serial0"
BAUD_RATE = 9600
PHONE_NUMBER = "01025681976"
```

```
TEMP_THRESHOLD = 50.0
SMOKE_FACTOR = 1.5
AI_THRESHOLD = 0.70
```

Parameter	Description
TEMP_THRESHOLD	Max safe temperature
SMOKE_FACTOR	MQ-2 smoke multiplier
AI_THRESHOLD	Fire risk probability limit

---

## 6. Sensor Modules

### 🔥 Flame Sensor

- Connected via GPIO
  - Detects direct flame presence
  - Digital input (HIGH = fire)
- 

### 🌡 DS18B20 Temperature Sensor

```
def read_temp():
    ...
    • Reads temperature from Linux 1-Wire interface
    • Returns temperature in Celsius
```

---

### 🌫 MQ-2 Smoke Sensor (MCP3008 ADC)

```
mq2 = AnalogIn(mcp, MCP.P0)
    • Uses SPI communication
    • Returns analog smoke concentration
    • Values are smoothed using rolling average
```

---

## 7. MQ-2 Calibration

```
def calibrate_mq2():
```

### Process:

1. Warm-up for 15 seconds
2. Read 50 samples
3. Compute baseline
4. Define smoke threshold

```
Threshold = Baseline × SMOKE_FACTOR
```

---

## 8. AI Fire Risk Prediction

### Model

- Logistic Regression (joblib file)
- Features used:

Feature	Description
Temperature	Normalized
MQ-2 Value	Normalized
Flame	Binary
Smoke Ratio	Current / Baseline
$\Delta$ Temperature	Change rate
$\Delta$ Smoke	Change rate
sensor_data["risk"]	= lr_model.predict_proba(X) [0] [1]

---

## 9. GSM SMS Alert System

### SIM800L Configuration

```
SERIAL_PORT = "/dev/serial0"
BAUD_RATE = 9600
```

### SMS Protection

- Cooldown: **30 seconds**
- Prevents message spamming

```
SMS_COOLDOWN = 30
```

---

## 10. Alert Logic

An alert is triggered if **any condition is true**:

```
flame detected  
OR smoke > threshold  
OR temperature > threshold  
OR AI risk ≥ AI_THRESHOLD
```

Actions:

- Buzzer ON
  - SMS sent
  - Blynk alert updated
- 

## 11. Multithreading Design

Each function runs in its own thread:

Thread	Function
Flame Thread	Reads flame sensor
MQ-2 Thread	Reads smoke values
Temp Thread	Reads temperature
AI Thread	Predicts fire risk
Blynk Thread	Updates IoT + alerts

Threads are started as **daemon threads** to ensure clean shutdown.

---

## 12. Main Execution Flow

```
calibrate_mq2()  
start all threads  
while True:  
    blynk.run()
```

- ✓ System starts automatically
  - ✓ Continuous monitoring
  - ✓ Non-blocking execution
-

## 13. Safety & Reliability

- Thread locks prevent race conditions
  - SMS cooldown prevents flooding
  - Sensor averaging reduces noise
  - AI adds predictive intelligence
- 

## 14. Possible Improvements

- Add camera + fire image detection
  - Cloud logging (Firebase / AWS)
  - Mobile notifications instead of SMS
  - Auto-start using systemd service
  - Battery backup support
- 

## 15. Conclusion

This system provides a **robust, intelligent, and scalable fire detection solution**, combining:

- Hardware sensors
- AI intelligence
- IoT monitoring
- GSM communication

It is suitable for **homes, labs, and small industrial environments**.

---

Youssef Ehab Fawzy

Mohamed Wanis

Mohamed Atef

Omar Khaled

Abdullah