PROJECT REPORT

Embedded Systems

Final Project



Smart Electrical Panel Protection and Emergency Alert System

STUDENT NAME **Asem Ameen Qaid**

NIM **24524038**

Computer Engineering Lab

Undergraduate Program of Electrical Engineering

Faculty of Industrial Technology

Islamic University of Indonesia

2026

ABSTRACT

Electrical distribution panels are a common origin of fires caused by overheating connections, insulation breakdown and short circuits. This report presents the design, implementation and testing of a low-cost embedded system that monitors an electrical panel and responds automatically to developing hazards. The system is built around an Arduino Nano microcontroller and observes three independent indicators: panel temperature (DHT11), smoke and combustible-gas concentration (MQ-2) and flame presence (IR flame sensor).

A three-state finite state machine (Normal, Warning and Emergency) determines the response. An elevated temperature activates a cooling fan and sends a warning SMS. Smoke, gas or flame additionally sounds a buzzer, energises a relay to disconnect the electrical load and sends a detailed emergency SMS through a SIM800L GSM module. A browser-based dashboard built on the Web Serial API displays live sensor data and the system event log.

The system was developed on a breadboard, assembled on a soldered PCB and validated against six test scenarios. In all trials the system transitioned correctly between states, disconnected the load within one 2-second polling cycle of an emergency being detected, and delivered SMS alerts to the responsible person's phone.

Keywords: embedded systems, Arduino, fire detection, GSM alert, electrical safety, finite state machine, relay control

CONTENTS

1 Introduction

2 System Design

3 Hardware Implementation

4 Software Design

5 Testing and Results

6 Discussion

7 Conclusion and Future Work

References

1 Introduction

1.1 Background

Electrical panels are a critical point of failure in residential and industrial installations. Faults such as overheating connections, insulation breakdown or short circuits can develop gradually and silently into fires long before they are noticed by a human observer. Panels are frequently left unattended, so a developing fault may already be uncontrolled by the time smoke or flame becomes visible.

Continuous automated monitoring allows a hazardous condition to be detected and mitigated within seconds and allows a responsible person to be notified remotely, even when nobody is present at the site.

1.2 Problem Formulation

1. How can the main early indicators of an electrical panel fire be monitored continuously at low cost?
2. How should the system respond in graduated stages, from early warning to protective action?
3. How can the electrical load be disconnected automatically when a real hazard is detected?
4. How can a responsible person be informed immediately when nobody is on site?

1.3 Objectives

- To monitor panel temperature, smoke/gas concentration and flame presence continuously.
- To activate cooling and issue a warning before conditions become dangerous.
- To disconnect the load automatically as soon as smoke, gas or flame is detected.
- To send a detailed SMS alert so that a responsible person can assess the situation and act remotely.
- To demonstrate a complete embedded design cycle (sensing, decision logic, actuation and wireless communication) within the constraints of an Arduino Nano.

2 System Design

2.1 Architecture Overview

The system follows a sense–decide–act architecture (Figure 1). Three sensors feed the Arduino Nano, which evaluates their combined readings every two seconds and drives a three-state machine. Each state maps to a fixed set of outputs: a three-colour traffic-light module, a cooling fan, a buzzer, a load-disconnect relay and SMS alerts sent through the SIM800L GSM module. A USB serial link carries live data to a companion dashboard on a PC.

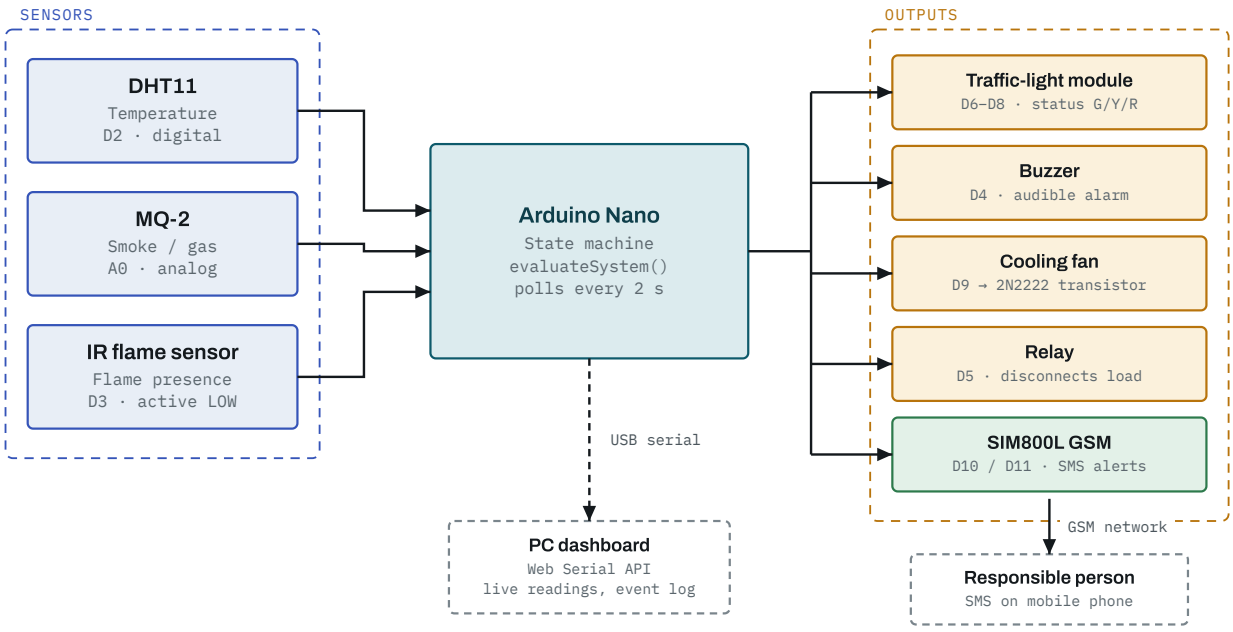


Figure 1. System block diagram showing sensor inputs, the Arduino Nano controller, output devices and communication paths.

2.2 State Machine Logic

The control logic is organised as a three-state finite state machine (Figure 2, Table 1). The Warning state is entered on temperature alone, while smoke, gas or flame moves the system directly to Emergency from any state. When all hazards clear, the system returns to Normal and reconnects the load.

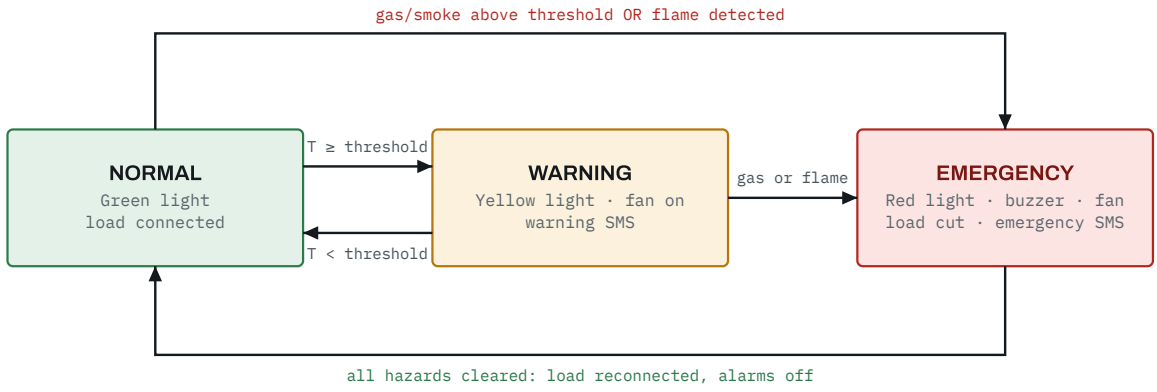


Figure 2. State transition diagram of the protection logic.

Table 1. System state logic

State	Trigger	Outputs
Normal	All readings within range	Green light; load connected
Warning	Temperature ≥ threshold	Yellow light; fan on; warning SMS

State	Trigger	Outputs
Emergency	Smoke/gas high, or flame detected	Red light; buzzer on; fan on; load disconnected; emergency SMS

Flame and gas/smoke hazards are treated as independent alert sources, each with its own SMS cooldown timer, so that an alert for one hazard type can never suppress a notification for a different hazard occurring at the same time.

3 Hardware Implementation

3.1 Bill of Materials

Table 2. Bill of materials

Component	Qty	Function
Arduino Nano	1	Main controller
MQ-2 gas/smoke sensor	1	Gas and smoke detection
IR flame sensor	1	Flame detection
DHT11 sensor	1	Panel temperature
SIM800L GSM module	1	SMS alert transmission
Single-channel relay	1	Load disconnect
NPN transistor (2N2222)	1	Fan switching
5 V DC fan	1	Active cooling
Traffic-light module	1	Visual status (green/yellow/red)
Buzzer	1	Audible emergency alarm
1000 µF electrolytic capacitor	1	Buffers SIM800L current spikes

3.2 Pin Assignment

Table 3. Arduino Nano pin assignment

Component	Pin	Notes
DHT11 data	`D2`	Digital
Flame sensor DOUT	`D3`	Active LOW
MQ-2 AOUT	`A0`	Analog
Buzzer	`D4`	
Relay IN	`D5`	Active LOW

Component	Pin	Notes
Traffic light green / yellow / red	D6 / D7 / D8	
Fan transistor base	D9	Via 220 Ω resistor
SIM800L TX	D10	Direct wire
SIM800L RX	D11	Via voltage divider

3.3 Power Architecture

Two isolated power domains were used to prevent high-current actuator switching from corrupting sensitive sensor readings (Figure 3). The Arduino Nano's regulated 5 V pin supplies the DHT11, flame sensor, MQ-2, buzzer and traffic-light module. A separate external 5 V / 2 A supply powers the relay coil, the fan (through the transistor) and the SIM800L module, whose transmission bursts can draw current spikes of approximately 2 A. A 1000 µF electrolytic capacitor across the external supply rail near the SIM800L buffers these spikes. Both domains share a single, direct ground connection so that all control signals have a common reference.

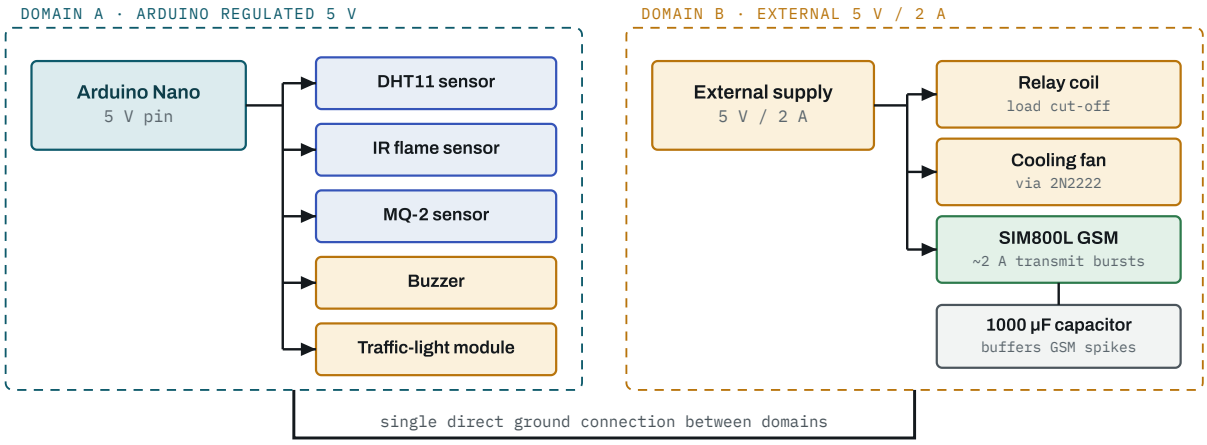


Figure 3. Power architecture with two isolated supply domains and a shared ground.

3.4 Design Constraints

Two component constraints shaped the circuit. Only 220 Ω resistors were available, so every base and divider resistor uses this value; the SIM800L RX voltage divider combines three 220 Ω resistors to obtain the required ratio. Only one relay module was available, so it was dedicated to the safety-critical load cut-off, and the fan is switched by a 2N2222 NPN transistor instead.

3.5 Prototype

The circuit was first assembled and debugged on a breadboard. The final prototype was then soldered on a PCB, shown in Figure 4.

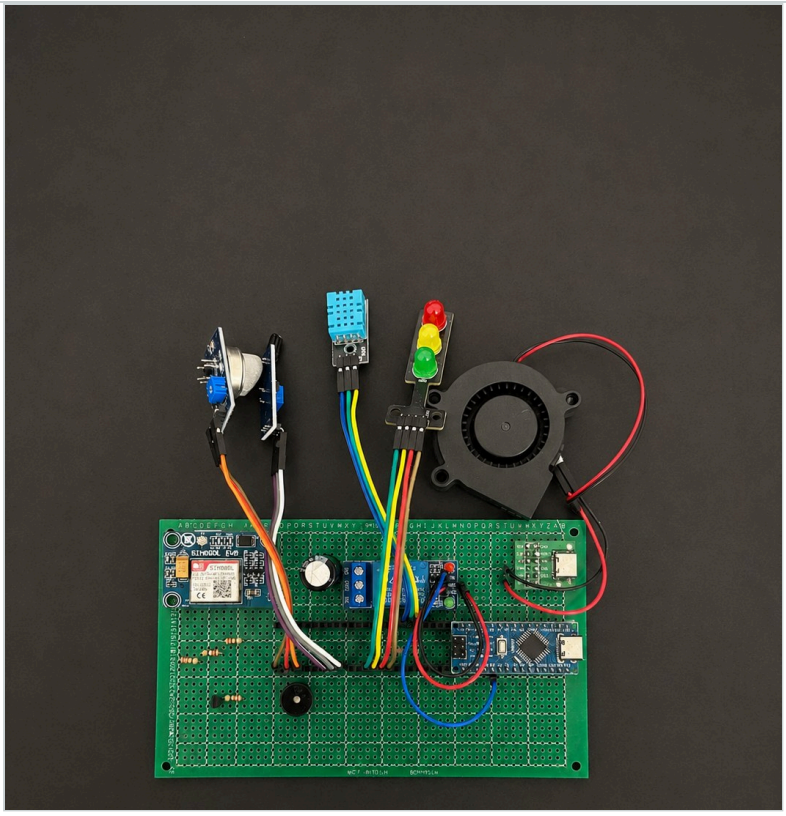


Figure 4. Final prototype soldered on a PCB, showing the sensors, traffic-light module, fan, buzzer, relay and SIM800L module.

4 Software Design

4.1 Firmware Overview

The firmware polls all three sensors every two seconds and evaluates the combined state in a single routine, `evaluateSystem()`, so that all outputs are driven from one consistent snapshot of the sensor readings (Figure 5). The relay and fan outputs default to a safe configuration (load connected, fan off) on every reset, so that a microcontroller reset never leaves the panel in an undefined state. Before an SMS is sent, the firmware checks the cooldown timer of the corresponding hazard type to prevent alert flooding.

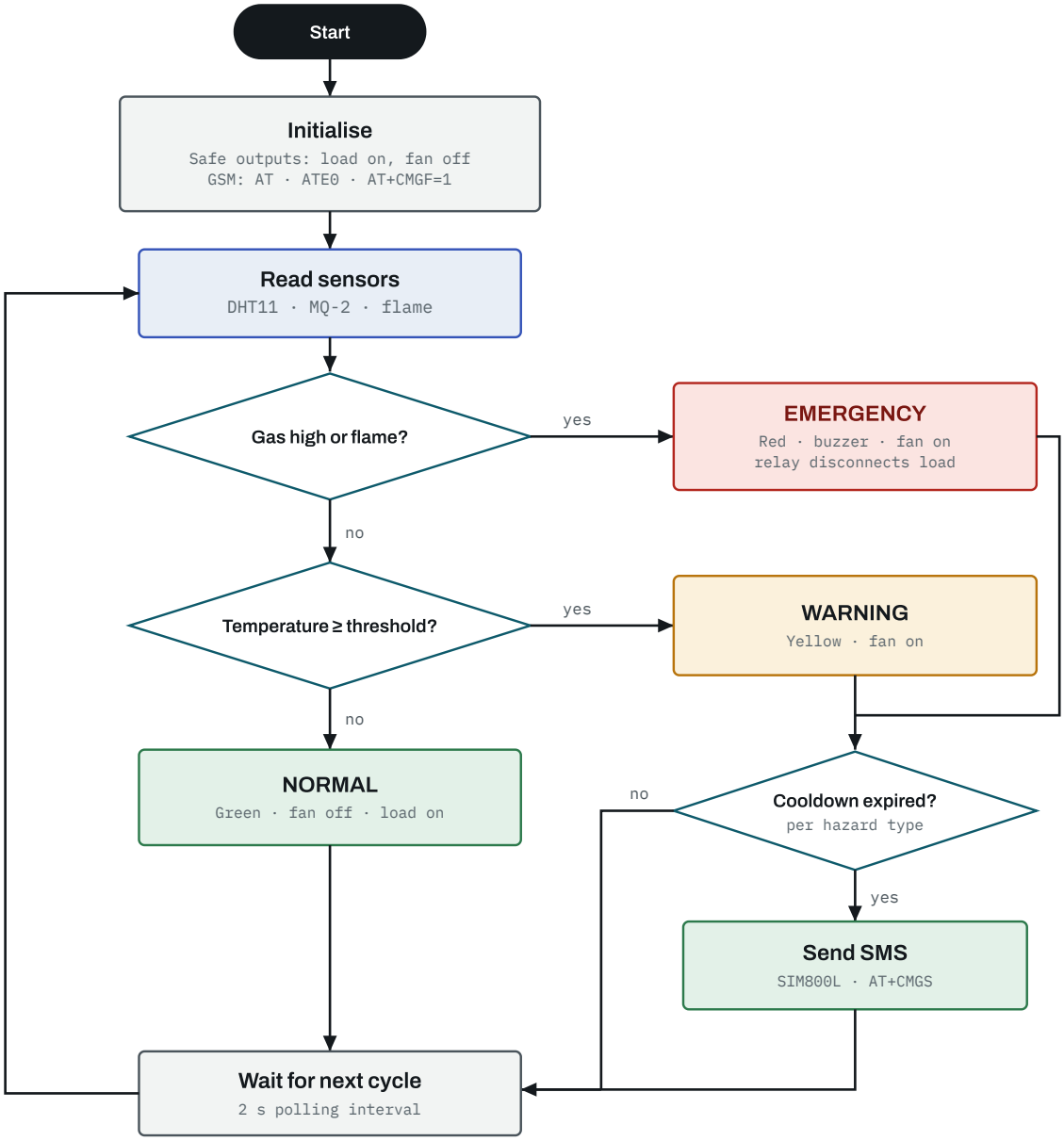


Figure 5. Firmware flowchart of the main control loop.

4.2 GSM Communication

The SIM800L is controlled through a software-emulated UART (`SoftwareSerial`), which keeps the hardware UART free for USB debugging and the dashboard. Table 4 lists the AT command sequence used to send an alert.

Table 4. SIM800L AT command sequence

Command	Purpose
`AT`	Verify module response
`ATE0`	Disable command echo
`AT+CMGF=1`	Set SMS text mode

Command	Purpose
`AT+CMGS="<number>"`	Begin composing an SMS
`Message + Ctrl+Z (0x1A)`	Terminate and transmit the message

4.3 Monitoring Dashboard

A companion browser dashboard reads live serial data from the Arduino over USB using the Web Serial API, so no additional hardware is required (Figure 6). It displays live temperature, gas and flame readings, a colour-coded state banner, the status of each actuator and a scrolling event log that includes every SMS sent. The dashboard runs in Chrome or Edge on a desktop computer.

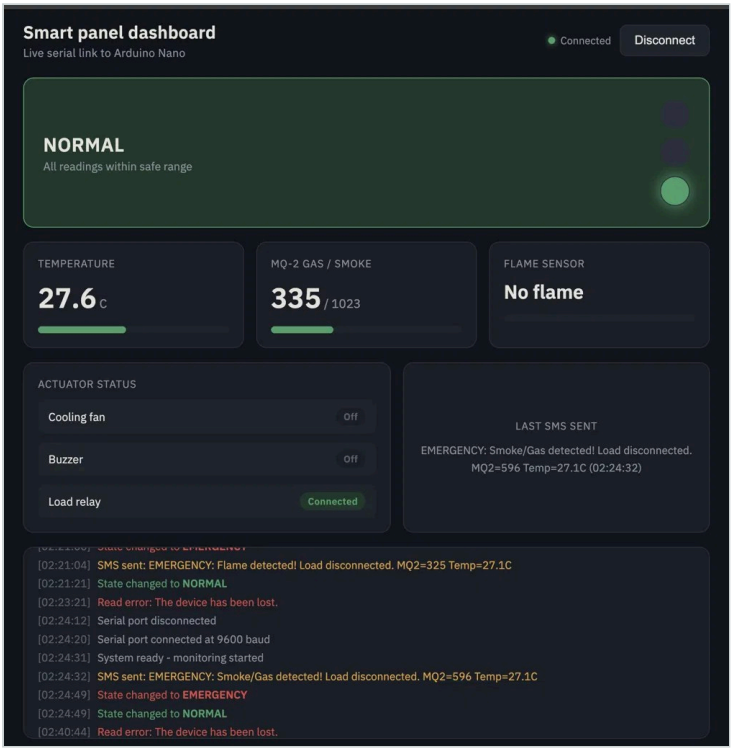


Figure 6. Live monitoring dashboard connected through the Web Serial API.

5 Testing and Results

5.1 Test Procedure

The system was validated against six scenarios covering start-up, communication, each hazard type and recovery (Table 5). Hazard conditions were created deliberately on the bench to trigger each state.

Table 5. Test scenarios and results

No.	Scenario	Expected behaviour	Result
1	Power-on self-check	Outputs start in the safe state; sensors respond	Passed
2	GSM registration	SIM800L responds to AT commands and registers on the network	Passed
3	Temperature-triggered warning	Yellow light, fan on, warning SMS	Passed
4	Gas/smoke-triggered emergency	Red light, buzzer, fan, load disconnected, emergency SMS	Passed
5	Flame-triggered emergency	Red light, buzzer, fan, load disconnected, emergency SMS	Passed
6	Automatic recovery	Return to Normal once hazards are removed	Passed

5.2 Representative Sensor Data

Table 6 shows representative readings recorded from the live serial output. The MQ-2 value is the raw 10-bit ADC reading (0–1023).

Table 6. Representative test readings

Temperature (°C)	MQ-2 (ADC)	Flame	State
26.70	350	No	Normal
26.70	1005	No	Emergency (gas)
26.70	407	Yes	Emergency (flame)
27.10	393	No	Normal

In all trials the system transitioned correctly between states, the load relay disconnected within one polling cycle (2 s) of emergency detection, and SMS alerts were delivered to the responsible person's phone, confirmed against message timestamps (Figure 7).

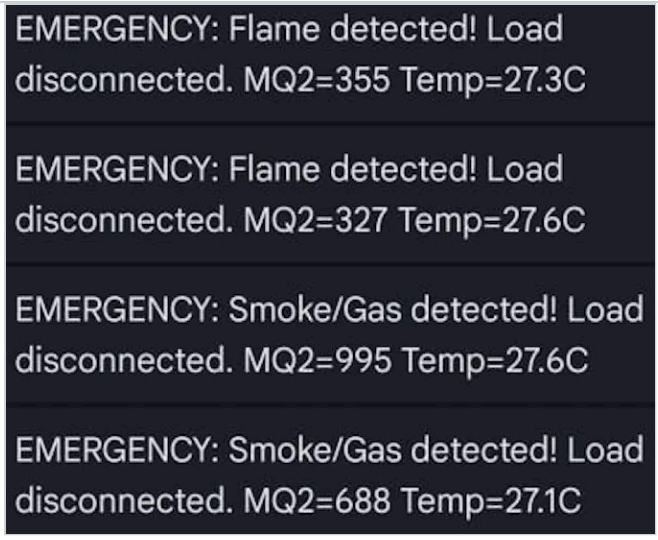


Figure 7. Emergency SMS alerts received during flame and gas tests, with the MQ-2 and temperature values at the time of detection.

6 Discussion

Several problems were identified during development and resolved before final testing (Table 7). The most significant was electrical: actuator switching on a shared supply rail interfered with sensor readings, which led to the two-domain power architecture described in Section 3.3.

Table 7. Problems identified during development

Problem	Cause	Solution
Intermittent false flame detection	Power-rail noise from the relay's inductive switching	Sensor power isolated on the Arduino's regulated rail, separate from the relay, fan and GSM domain
SMS not delivered	SIM800L brown-out during its ~2 A transmission burst	1000 µF capacitor added across the supply near the module
One hazard suppressing another	A shared cooldown timer blocked alerts of a different type for 60 s	Independent cooldown timers for flame and gas
No alert in the first 60 s after boot	Cooldown timer initialised so that it had not expired at start-up	Timer initialised to a value in the past
Temperature not formatted in SMS text	AVR `snprintf()` does not support floating-point formatting by default	Values converted with `dtostrf()` before insertion

The MQ-2 sensor's raw analog output required on-site threshold calibration, because its baseline varied with warm-up time, humidity and unit-to-unit variation. GSM SMS delivery latency, typically a few seconds to under a minute, was governed by network conditions rather than by firmware behaviour.

7 Conclusion and Future Work

This project demonstrates a practical, low-cost embedded solution for early fire and electrical hazard detection. By combining multi-sensor monitoring, a clearly defined state machine, automatic load disconnection and GSM-based alerting, the system fulfils its core safety objective: minimising the time between a hazardous condition arising and a responsible person being informed and able to act.

Future improvements include:

- calibrated gas concentration readings at ppm-level accuracy;
- per-circuit load isolation instead of a single cut-off point;
- local data logging (SD card and real-time clock) for post-incident analysis.

References

1. Winsen Electronics, *MQ-2 Semiconductor Sensor for Combustible Gas*, datasheet. Winsen Electronics Technology Co., Ltd.
2. SIMCom Wireless Solutions, *SIM800L Hardware Design Manual*. SIMCom Ltd.
3. Aosong Electronics, *DHT11 Humidity and Temperature Sensor*, datasheet. Aosong (Guangzhou) Electronics Co., Ltd.
4. Arduino S.r.l., *Arduino Nano Technical Specifications*. Available: https://docs.arduino.cc