Fire Protection System Using Arduino Uno

```
/* Includes Section*/
#include <Wire.h>
#include <SoftwareSerial.h>
#include "LiquidCrystal.h"

// Initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 7, en = 8, d4 = 3, d5 = 4, d6 = 5, d7 = 6;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

/* Variables Section*/
#define Gas_Sensor_Output 9    // The output signal of the gas detector
#define Flame_Sensor_Output 2  // The output signal of the flame sensor
#define Emergency_LED 10
#define Emergency_Alarm 11
#define ELECTRIC_GAS_VALVE A0
#define FIRE_EXTINGUISHING_VALVE A1
#define Emergency_Exhaust_Fan A2
#define LOAD 12

bool Flame_Sensor_state;
bool Gas_Sensor_state;

void setup()
{
  /* Setting Pin Modes */
  pinMode(Flame_Sensor_Output, INPUT);
  pinMode(Gas_Sensor_Output, INPUT);
  pinMode(Emergency_Alarm, OUTPUT);
  pinMode(Emergency_LED, OUTPUT);
  pinMode(ELECTRIC_GAS_VALVE, OUTPUT);
  pinMode(FIRE_EXTINGUISHING_VALVE, OUTPUT);
  pinMode(Emergency_Exhaust_Fan, OUTPUT);
  pinMode(LOAD, OUTPUT);

  digitalWrite(LOAD, HIGH);  // Connect the main building's electricity.
  digitalWrite(ELECTRIC_GAS_VALVE, HIGH);   // Activate the main gas valve

  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("It's All good");

  // Begin the serial connection with baud rate 9600
  Serial.begin(9600);
}

void loop()
{
  Flame_Sensor_state = digitalRead(Flame_Sensor_Output);  // Check the output signal of the flame sensor.
  Gas_Sensor_state = digitalRead(Gas_Sensor_Output);      // Check the output signal of the gas sensor.

  // If the flame is detected
  if (Flame_Sensor_state == LOW)
  {
    digitalWrite(Emergency_Alarm, HIGH);  // Activate the emergency alarm.
    digitalWrite(Emergency_LED, HIGH);    // Activate the emergency alarm.
    digitalWrite(LOAD, LOW);              // Disconnect the building's primary power source.
    digitalWrite(ELECTRIC_GAS_VALVE, LOW);          // Shut down the main gas valve to stop gas flow.
    digitalWrite(FIRE_EXTINGUISHING_VALVE, HIGH);   // Activate the extinguishing valve to put down the fire.
    digitalWrite(Emergency_Exhaust_Fan, HIGH);      // Activate an emergency high-pressure exhaust fan to remove leakage gas.

    /* Display on the LCD "Flame Detected!!"*/
    lcd.setCursor(0, 0);
    lcd.print("Fire Alert!!!!");

```

```
    lcd.setCursor(0, 1);
    lcd.print("Flame Detected!!");
  }
  else if (Gas_Sensor_state == LOW)
  {
    digitalWrite(Emergency_Alarm, HIGH);  // Activate the emergency alarm.
    digitalWrite(Emergency_LED, HIGH);    // Activate the emergency alarm.
    digitalWrite(ELECTRIC_GAS_VALVE, LOW);          // Shut down the main gas valve to stop gas flow.
    digitalWrite(Emergency_Exhaust_Fan, HIGH);      // Activate an emergency high-pressure exhaust fan to remove leakage gas.

    /* Display on the LCD "Gas Detected!!"*/
    lcd.setCursor(0, 0);
    lcd.print("Fire Alert!!!!");
    lcd.setCursor(0, 1);
    lcd.print("Gas Detected!!");
  }
  else if (Flame_Sensor_state != LOW && Gas_Sensor_state != LOW)
  {
    delay(500);
    digitalWrite(Emergency_Alarm, LOW);  // Deactivate the emergency alarm.
    digitalWrite(Emergency_LED, LOW);    // Deactivate the emergency alarm.
    digitalWrite(LOAD, HIGH);            // Connect the building's primary power source.
    digitalWrite(ELECTRIC_GAS_VALVE, HIGH);         // Open the main gas valve to stop gas flow.
    digitalWrite(FIRE_EXTINGUISHING_VALVE, LOW);    // Deactivate the extinguishing valve to put down the fire.
    digitalWrite(Emergency_Exhaust_Fan, LOW);       // Deactivate an emergency high-pressure exhaust fan to remove leakage gas.

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("It's all good");
  }
}
```
