Fire Protection System Using Arudino Mega

```
/* 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 = 53, en = 52, d4 = 51, d5 = 50, d6 = 49, d7 = 48;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Initialize an object from the class SoftwareSerial
SoftwareSerial sim800l(0, 1);

/* Variables Section*/
#define Gas_Sensor_Output 22    // The output signal of the gas detector
#define Flame_Sensor_Output 23  // The output signal of the flame sensor
#define Emergency_LED 24
#define Emergency_Alarm 25
#define ELECTRIC_GAS_VALVE 26
#define FIRE_EXTINGUISHING_VALVE 27
#define Emergency_Exhaust_Fan 28
#define LOAD 29

unsigned long lcdTimer = 0;
unsigned long lcdInterval = 500;
unsigned long smsTimer = 0;
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
  sim800l.begin(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 == HIGH)
  {
    SendSMS();                                      // GSM & GPS module to send notification and position to the firefighter and authority.

    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 == HIGH)
  {
    SendSMS();                                    // GSM & GPS module to send notification and position to the firefighter and authority.

    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
  {
    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.
    smsTimer = 0;                                 // Sets the smsTimer to 0;

    if (millis() - lcdTimer >= lcdInterval)
    {
      lcd.clear();
      lcdTimer = millis();
      lcd.setCursor(0, 0);
      lcd.print("It's all good");
    }
  }
}

void SendSMS()
{
  if(smsTimer == 0)
  {
    Serial.println("Sending Location...");
    sim800l.print("AT+CMGF=1\r");
    sim800l.print("AT+CMGS=\"180\"\r");
    sim800l.print("SIM8001 is working");
    sim800l.println();
    Serial.println("Location Sent.");
    smsTimer = millis();
  }
}
```
