A Microcontroller-based Fire Protection System for the Safety of Industries in Bangladesh

Under supervision of Dr/ Laila Abou Hashem

By

1-Mazen Sayed

2-Aboubaker Adel

3-Mazen Magdy

4-Ahmed Gamal Aldin

5-Abdallah Mohamed

6-Khaled Mohamed

7-Mohamed Ahmed Galal

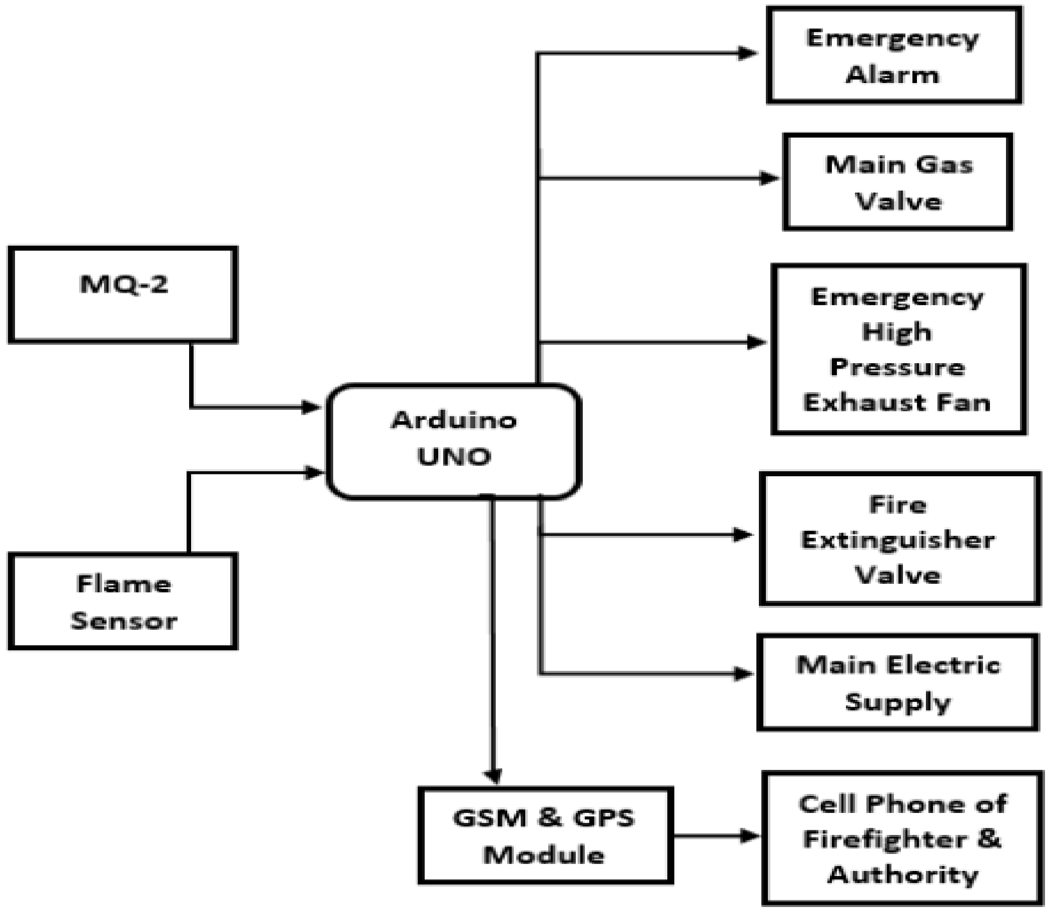
Introduction

- 4621 registered garment industries in Bangladesh employ about 4 million workers.
- fire-safety has become a major concern as We have lost more than 140 lives the past year due to several fire incidents across the country.
- Most of Deaths reasons are short circuits, leaking gas, inadequate fire protection system, or lack of effective fire alarm.
- There is two type of fire protection:

1-Early warning of fire detection.

2- A system that will not only detect the fire but also take essential attempts to stop it.

- Block diagram of the system.



How it works ?

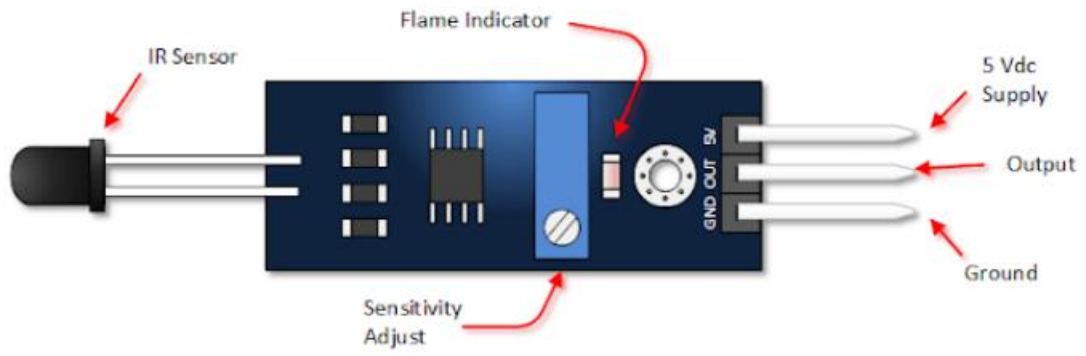
- When the system is powered on it automatically starts to sense the environment for toxic gas or flames.
- The sensors will send these data to Arduino UNO. Arduino UNO will examine them according to the installed program
- If the flame is detected the system will activate the emergency alarm, disconnect the building's primary power source, shut down the main gas valve to stop gas flow, activate an emergency high-pressure exhaust fan to remove leakage gas, and GSM & GPS module to send notification and position to the firefighter and authority
- During gas leakage the gas valve will be shut off, the alarm & exhaust fan will be activated and notification will be sent

System 's components

- Flame Sensor
- Gas Sensor
- Arduino UNO
- GSM Modules
- Buzzer
- Electric Gas Valve
- Magnetic Contactor

Flame sensor

- detecting the presence of flame or fire where it is placed using the infrared flame flash method.
- Why we flame senso insted of heat sensors?
- As it can respond faster and more accurately than a heat sensor for its mechanism
- When the sensor detects flame or fires it sends a signal to the microcontroller and the microcontroller simultaneously active the buzzer & sends a notification to the authority through the GSM module.

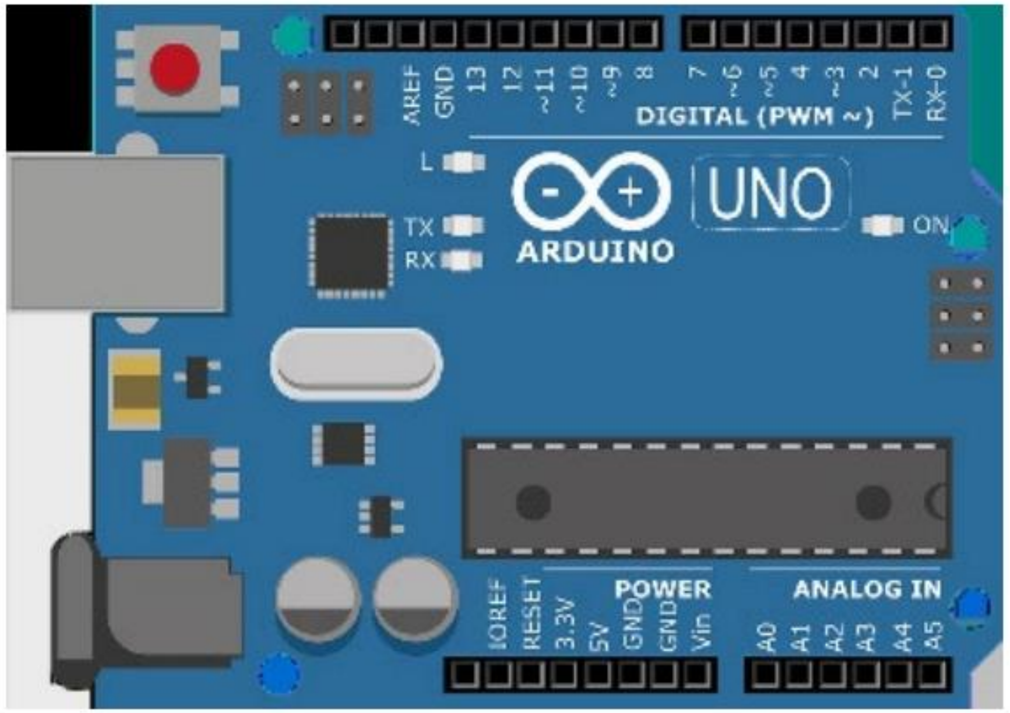


Gas Sensor

- we use MQ-2 as a gas or smoke detector sensor.
- It has a high sensitivity to detect flammables h aving a concentration of 300- 10000ppm

Arduino UNO

- It is the central processing unit of the system.
- Consist of
- It has 6 analog input pins
- 14 digital input/output pins (6 pins of it used as PWM output.
- It can be programmed with the help of Arduino IDE using a type B USB cable .
- It is operated at the voltage of 5V to 12V.



Buzzer

- known as a beeper.
- It converts electrical energy into sound energy with the help of transistors & capacitors. It is widely used in alarm & timer circuits. In our system, it is used to get alerts when the sensor detects smoke or fire.

GSM module.

- is a specialized hardware device that utilizes GSM technology to enable communication capabilities through cellular networks. When incorporated into an application, it allows for bidirectional wireless communication by sending and receiving both data and voice calls.
- A SIM can be inserted in the module to send signals, messages, or to make calls

Electric Gas Valve

- is a high-quality solenoid valve that is used to control the flow of gas
- we use this electric valve to control the flow of the main gas valve when leakage smoke or fire is detected in the sensor.

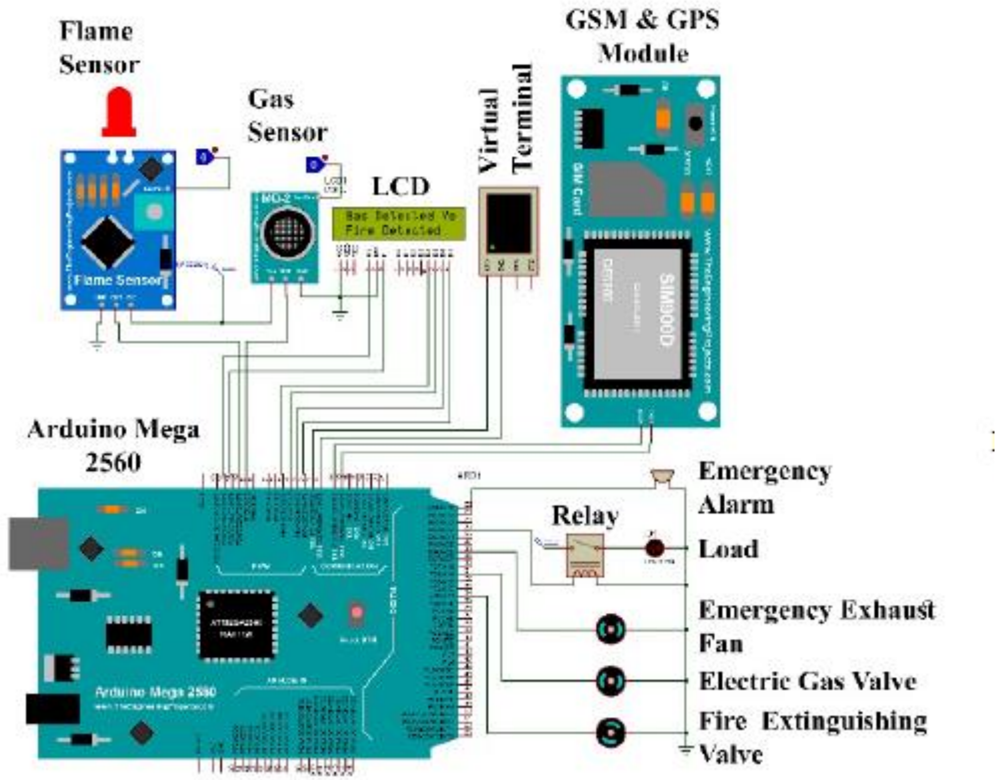
Magnetic Contactor

- In this system when smoke or fire is detected, the microcontroller sends a signal to this magnetic contactor or relay to turn off the main supply of electricity.

SIMULATION

We have used

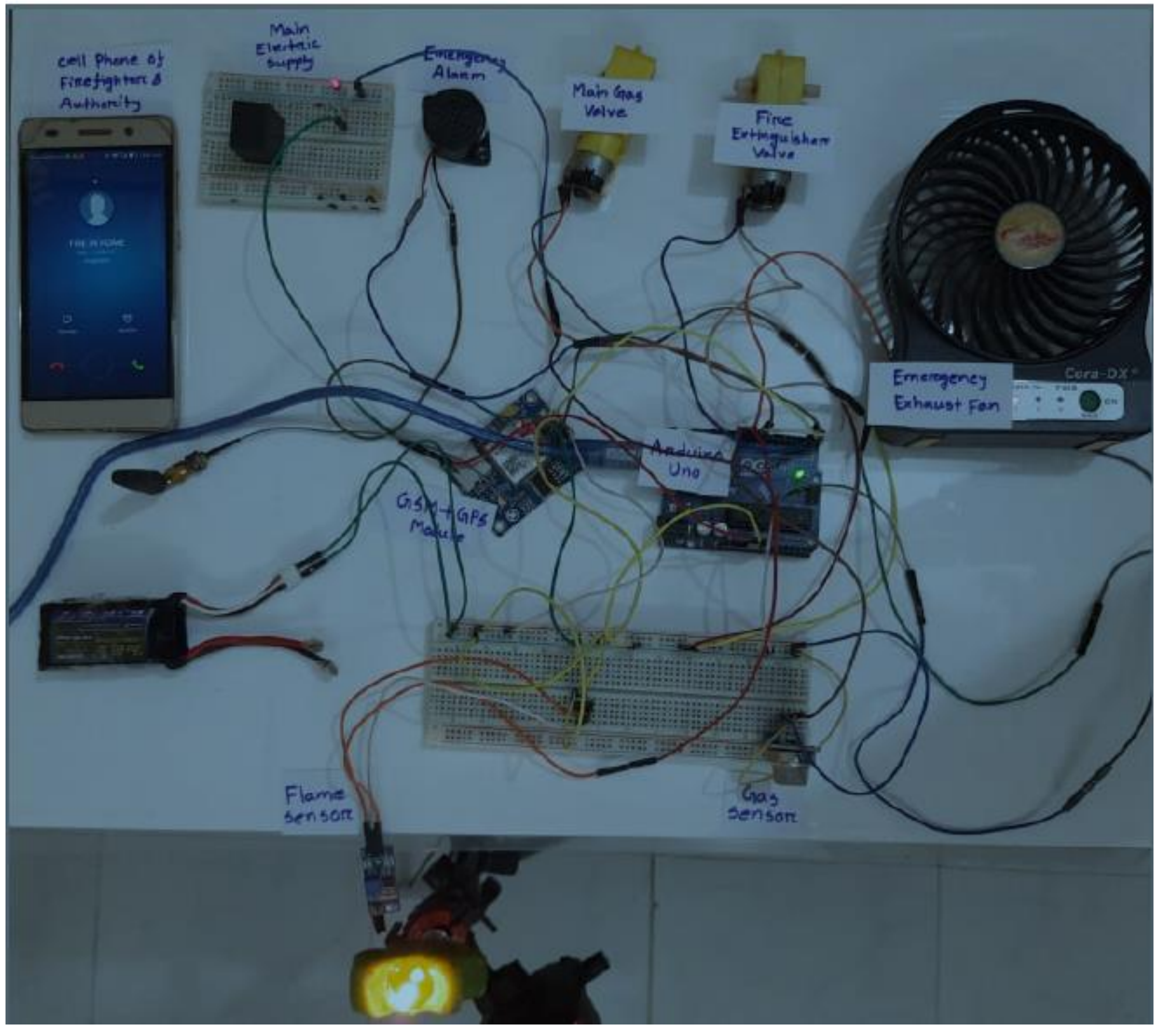
- Arduino Mega 2560as the central processing unit
- Flame sensor and a Gas senor as the input unit
- load, emergency alarm, motor control valve, GSM & GPS module, etc. as the output



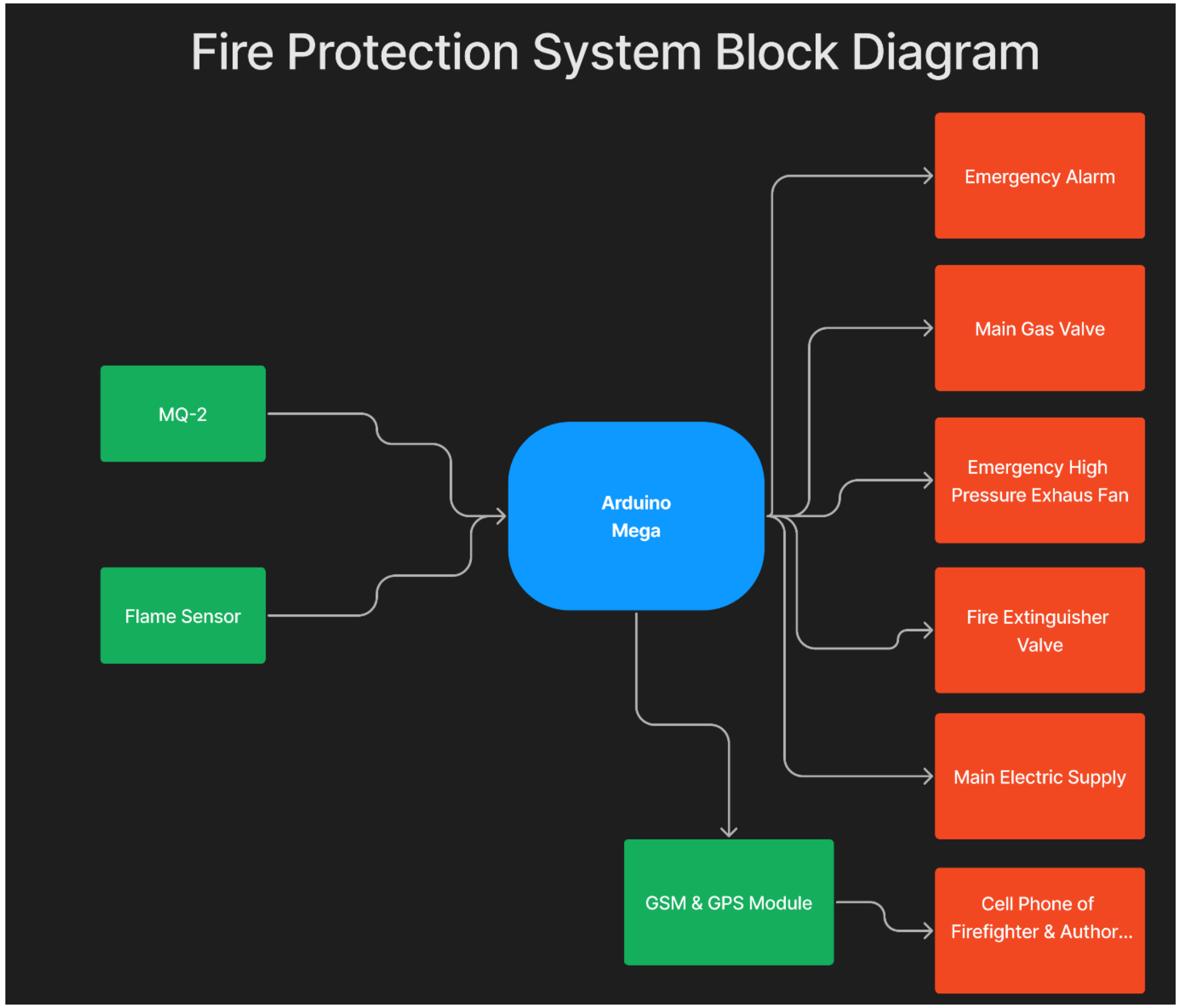
PRACTICAL EXPERIMENT

It's a copy of the simulation but we have used

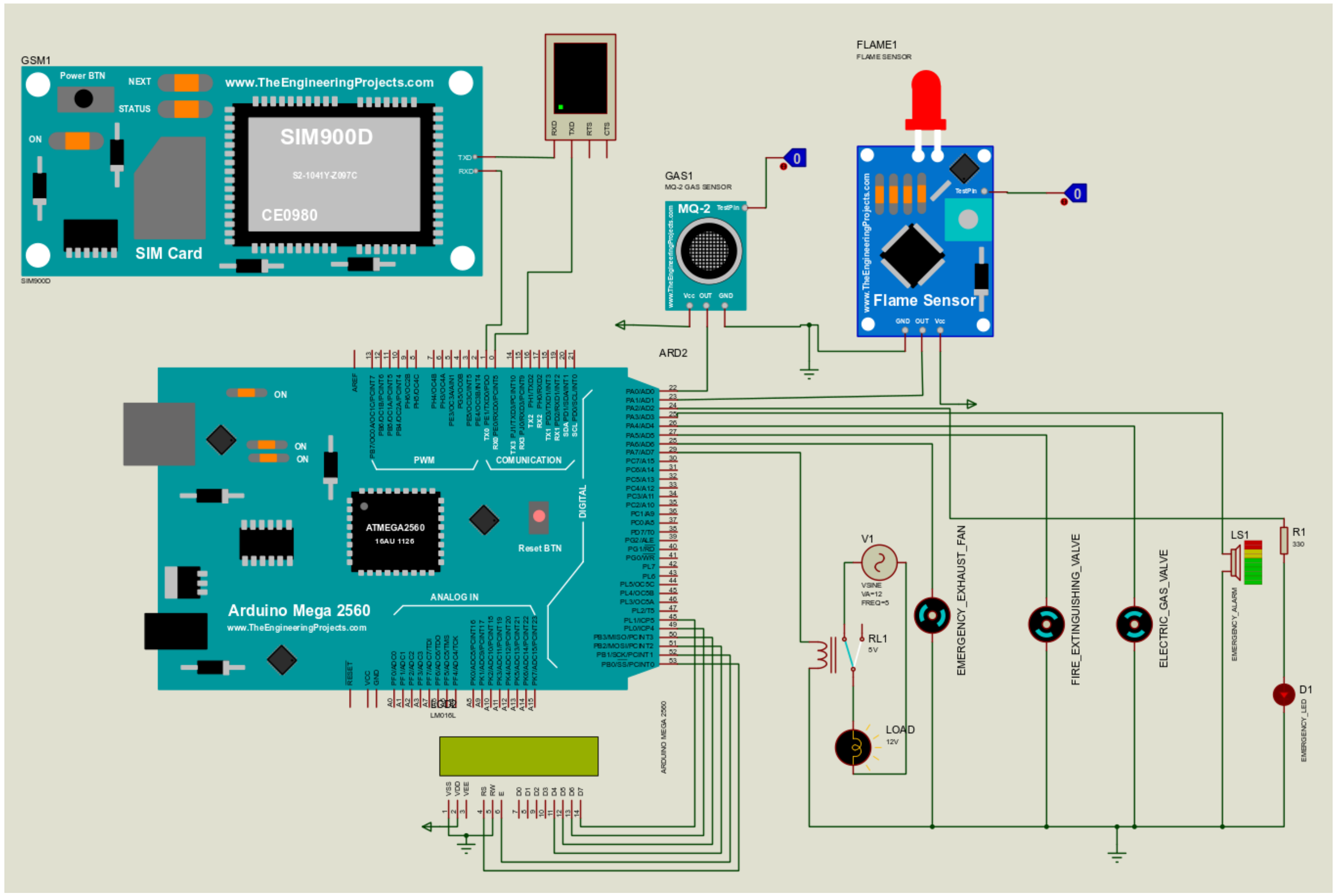
- candle light as a fire source
- a relay connected to a load as main power supply
- Motor as electric gas valve
- Fan as exhaust fan of the industry



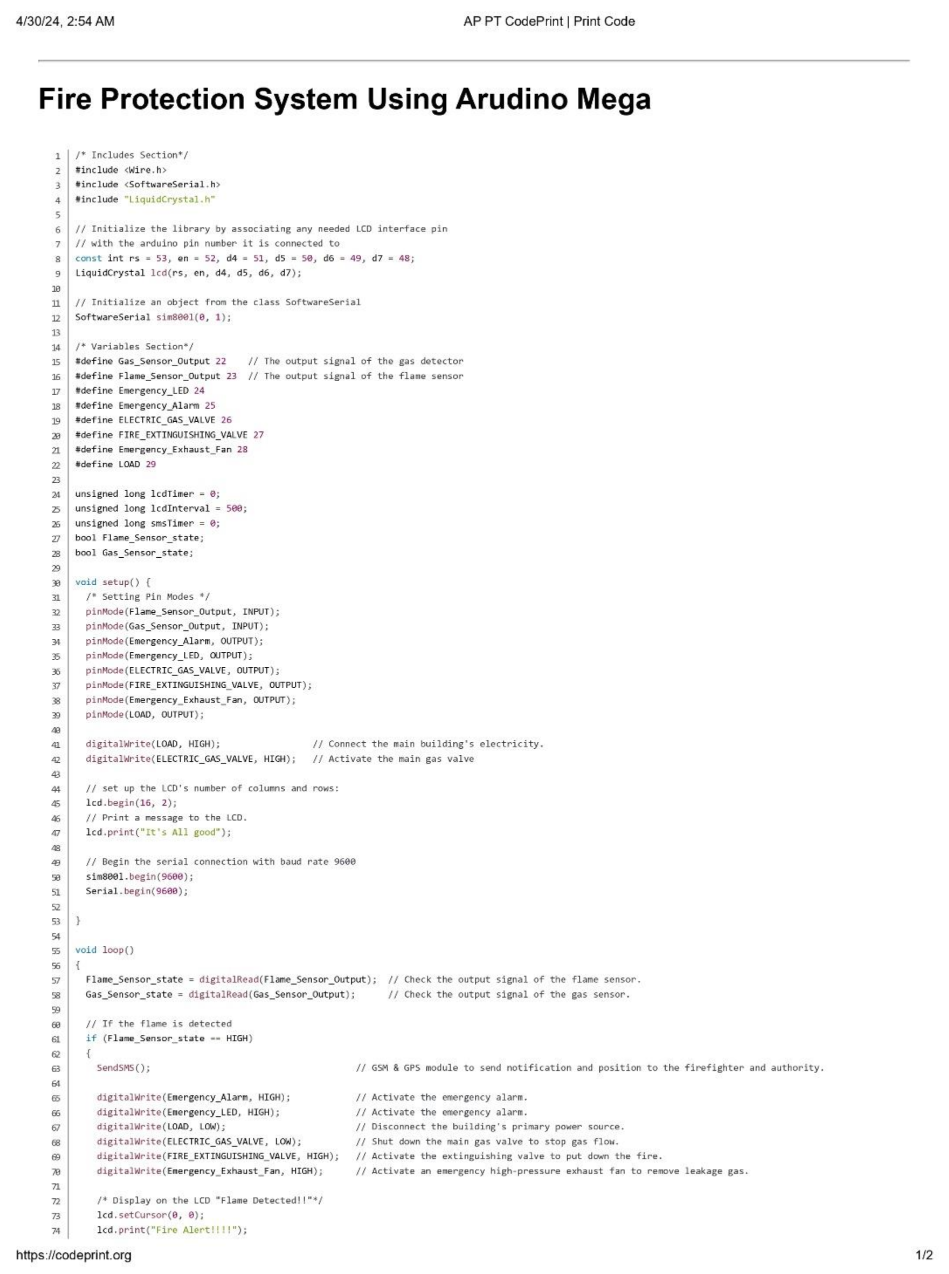
Block Diagram



Schematic


GSM1
SIM900D
SIM Card
www.TheEngineeringProjects.com
GAS1
MQ-2 GAS SENSOR
FLAME1
FLAME SENSOR
Flame Sensor
ARD2
Arduino Mega 2560
ATMEGA2560
Reset BTN
PWM
COMUNICATION
DIGITAL
ANALOG IN
LCD2
LM016L
V1
RL1
LOAD
EMERGENCY_EXHAUST_FAN
FIRE_EXTINGUISHING_VALVE
ELECTRIC_GAS_VALVE
LS1
EMERGENCY_ALARM
R1
330
D1
EMERGENCY_LED

Code



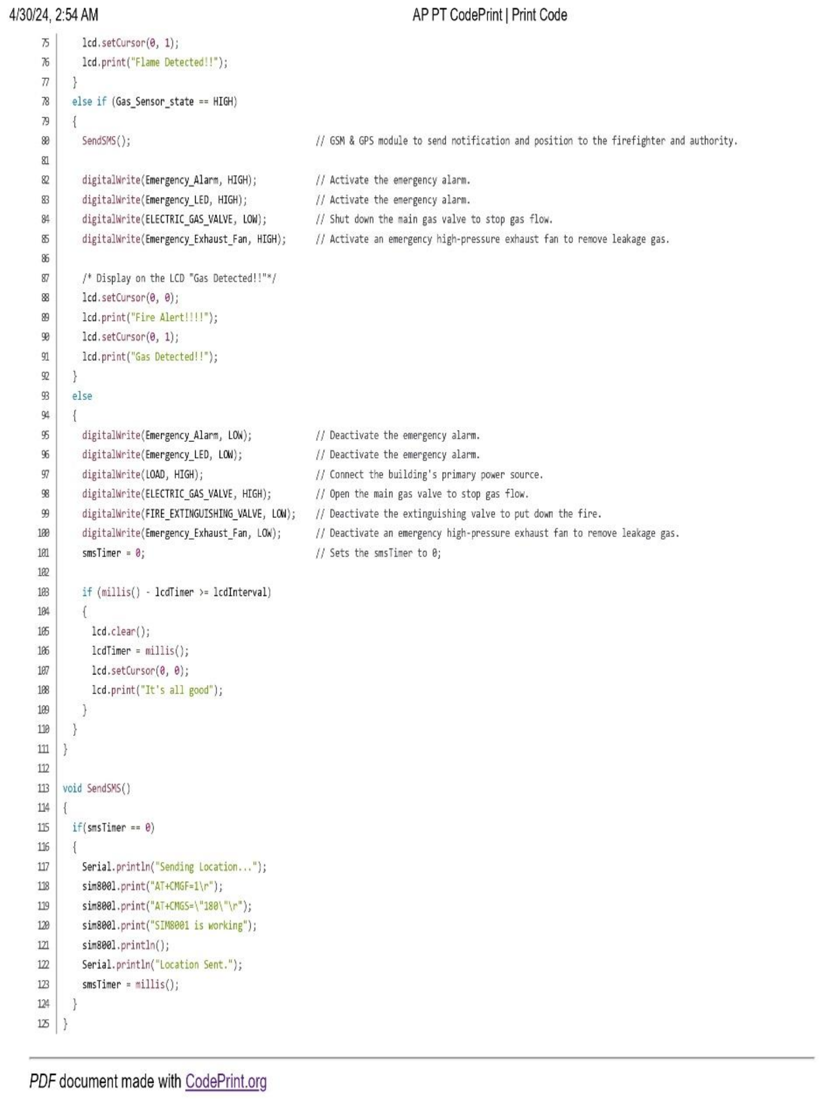
4/30/24, 2:54 AM AP PT CodePrint | Print Code

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!!!!");
```

https://codeprint.org 1/2

Code



4/30/24, 2:54 AM AP PT CodePrint | Print Code

```
    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=\"100\"\r");
    sim800l.print("SIM8001 is working");
    sim800l.println();
    Serial.println("Location Sent.");
    smsTimer = millis();
  }
}
```

PDF document made with CodePrint.org

CONCLUSION

- In this paper, we have proposed a microcontroller-based automated fire protection system that can detect any fire source and take immediate action to prevent it. At the same time, the system can also send an alert notification to the authority within a very short time to take extra measures