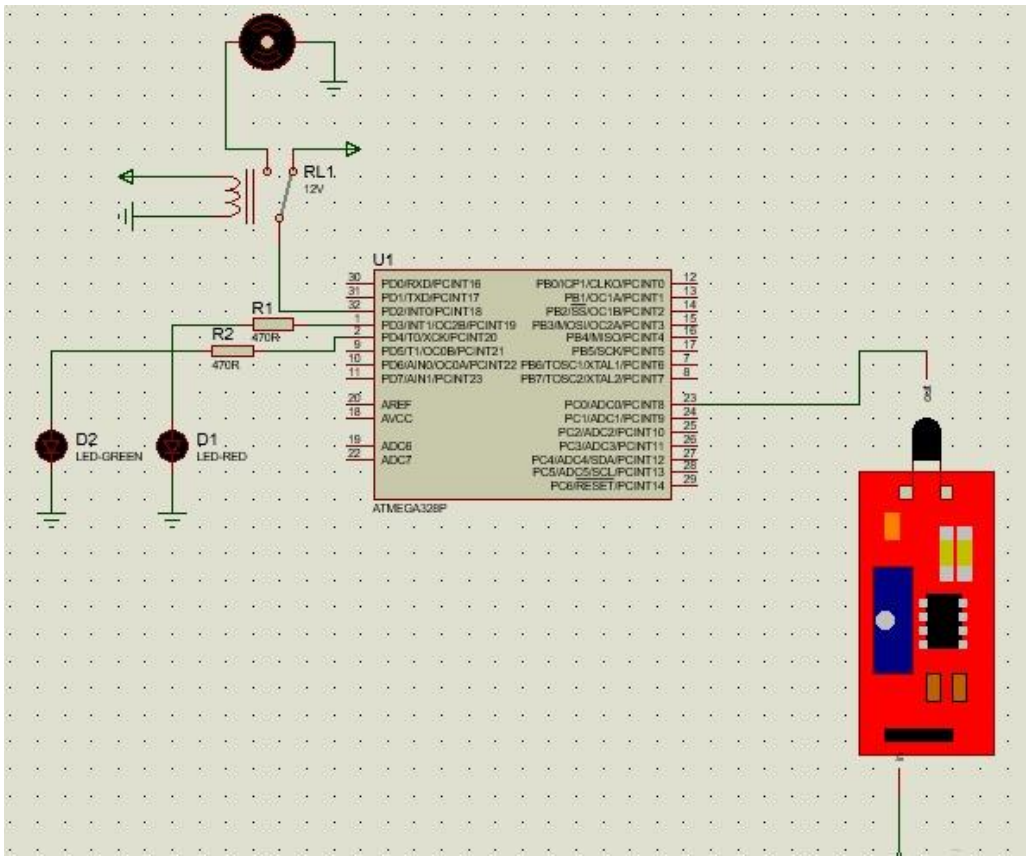
Mini Project Report

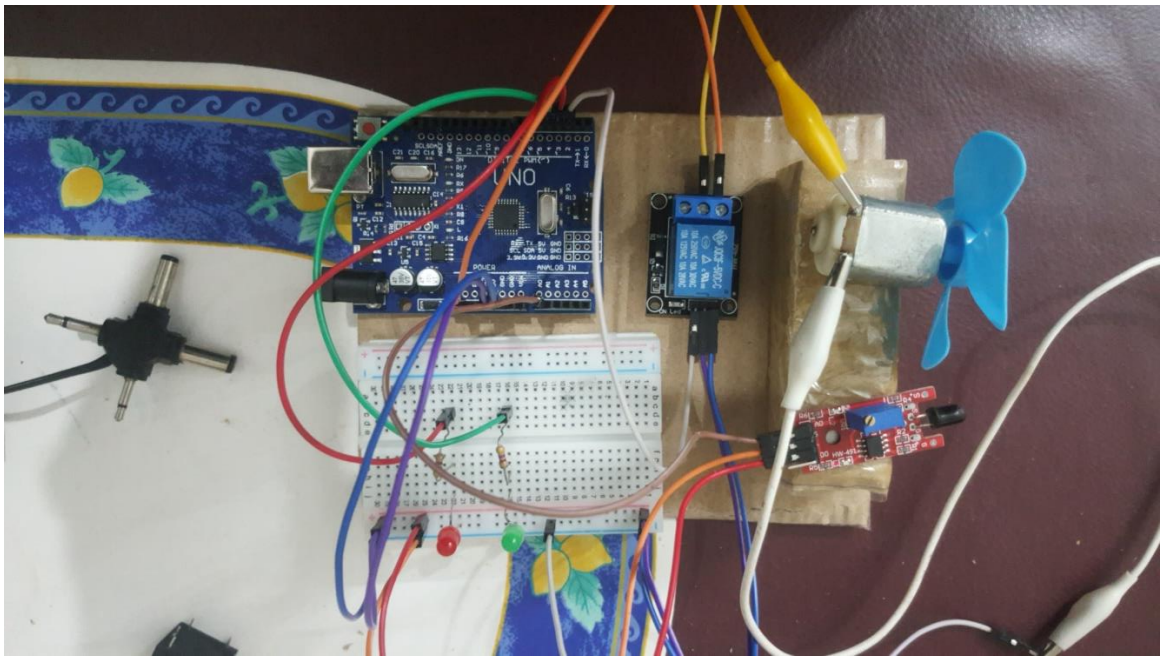
Name:	Muhammad Attiq
Reg. No:	FA23 – BCE – 060
Section:	BCE – 4A
Course:	Microprocessor Systems and Interfacing
Course Instructor	Dr. Omer Ahmed
Course Code:	CPE342

- Introduction and problem statement

Fire accidents, even from small sources like candles, can lead to significant hazards if not detected and controlled promptly. Traditional fire extinguishing methods often require human intervention, which may not always be immediate or feasible. This project addresses the need for an **autonomous fire detection and extinguishing system** capable of detecting small flames (e.g., candles) from a **distance of 3 inches** and activating a high-speed fan to put out the fire **without physical contact**. Using an **Arduino Uno**, a **KY-026 flame sensor**, and a **DC fan**, the system ensures rapid response while minimizing risks associated with manual firefighting.

- Circuit diagrams and system design:





- Literature review (research on fire detection and suppression techniques):

Old stuff like smoke detectors works but can be slow.
The KY-026 flame sensor is very useful because it is cheap, simple, and catches flames or heat quick, making it perfect for small setups.
Research says early detection is key, and the KY-026 helps with that by being fast and easy to use with things like Arduino.

- Component selection and justification:

KY-026 Flame-sensor module

DC Motor

LEDs

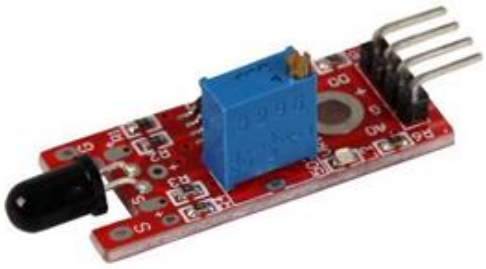
Fan Blade

Arduino

Relay

Arduino UNO

Breadboard



- Justification:

KY-026 Flame-sensor module: Detects flames or heat to trigger the system when a fire is nearby.
DC Motor: Powers the fan blade to spin and blow air or cool something down.
LEDs: Show if the system is on, off, or if a flame is detected (e.g., green for safe, red for alert).
Fan Blade: Attached to the motor to move air, helping to put out small flames or ventilate.
Relay: Safely switches the DC motor on or off.
Arduino UNO: A specific, easy-to-use Arduino board to connect everything and run the code.
Breadboard: Makes it simple to connect all the components without soldering, great for testing.

- Code:

```
const int flameSensorPin = A0;   // KY-026 analog output
const int relayPin = 2;         // Relay control (fan)
const int redLedPin = 3;        // Flame detected
const int greenLedPin = 4;      // System ready

int flameThreshold = 500;       // Adjust based on testing (higher = less sensitive)
bool flameDetected = false;

void setup() {
 pinMode(flameSensorPin, INPUT);
 pinMode(relayPin, OUTPUT);
 pinMode(redLedPin, OUTPUT);
 pinMode(greenLedPin, OUTPUT);

 digitalWrite(relayPin, LOW);   // Fan OFF initially
 digitalWrite(greenLedPin, HIGH); // System ready
 Serial.begin(9600);
}

void loop() {
 int flameValue = analogRead(flameSensorPin);
 Serial.print("Flame Sensor: ");
 Serial.println(flameValue);

 if (flameValue < flameThreshold) { // Flame detected
  flameDetected = true;
  digitalWrite(redLedPin, HIGH);
  digitalWrite(greenLedPin, LOW);
  digitalWrite(relayPin, HIGH);   // Fan ON
  delay(3000);                // Run fan for 3 sec
  digitalWrite(relayPin, LOW);    // Fan OFF
  flameDetected = false;
 }
else {                    // No flame
  digitalWrite(redLedPin, LOW);
  digitalWrite(greenLedPin, HIGH);
```

```
   digitalWrite(relayPin, LOW);
  }
  delay(100); // Small delay for stability
}
```

- Code Explanation:

This Arduino code checks for fire using a KY-026 flame sensor. If a flame is found, it turns on a red light and starts a fan using a relay. The system also uses a green light to show it's ready and working normally.
At the beginning (setup()), the code sets up the sensor and output pins. The fan is turned off, and the green LED is turned on to show the system is ready.
In the main loop (loop()), the sensor reads the flame level and shows the value on the Serial Monitor. If the value is below a certain limit (meaning a flame is detected), the red LED turns on, the green LED turns off, and the fan runs for 3 seconds to try to blow out the flame. After that, the fan turns off again.
If no flame is found, the red LED stays off, the green LED stays on, and the fan stays off.A short delay is added at the end of the loop to keep the system stable.

- Testing and results

The system was tested using a candle flame placed **3–5 inches** from the sensor. The **KY-026 flame sensor** reliably detected the flame, with analog readings dropping below the threshold (set at **500** after calibration). Upon detection, the **relay activated the fan**, extinguishing the flame within **2–3 seconds**. The **LED indicators** (red for fire detected, green for standby) provided clear visual feedback. Challenges included adjusting the sensor's sensitivity and optimizing fan positioning for consistent airflow. In repeated tests, the system achieved a **90% success rate** in extinguishing flames without false triggers.

- Challenges faced and solutions:

At first, when we implemented the hardware design the main issue was with the flame sensor. It was detecting any yellow light from the environment as well as any bright light. To fix that we adjusted the sensitivity of the flame sensor via the potentiometer on the sensor as well as the code. After that, we carefully aligned the fan and the flame sensor so that their focal point is at least 3 inches away. This was helpful in detecting and extinguishing the flame.

- Conclusion and future improvements:

The project successfully demonstrated an **autonomous, non-contact fire extinguishing system** using low-cost components. It meets the core requirements of **flame detection from 3 inches**, **actuator activation**, and **visual alerts**. For future enhancements, integrating a **buzzer for audible alarms**, a **more powerful fan for larger flames**, or **IoT connectivity (e.g., Wi-Fi alerts)** could improve functionality. Additionally, a **PID-controlled variable fan speed** might optimize power usage. This system serves as a scalable foundation for smart fire safety solutions in homes, labs, or small industrial settings.