AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # OEL

Experiment Title: Fire Alarm System Using Arduino Uno and Flame Sensor

Date of Perform:	20th May 2025	**Date of Submission:**	21st May 2025
Course Title:	Microprocessor and Embedded Systems Lab		
Course Code:	EEE4103	**Section:**	**R**
Semester:	Spring 2024-25	**Degree Program:**	BSc in CSE
Course Teacher:	**Prof. Dr. Engr. Muhibul Haque Bhuyan**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case Study is my/our original work; no part has been copied from any other student's work or any other source except where due acknowledgment is made.
3. No part of this Assignment/Case Study has been written for me/us by any other person except where such collaboration has been authorized. by the concerned teacher and is acknowledged in the assignment.
4. I/we have not previously submitted or am submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived to detect plagiarism.
6. I/we permit a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea, or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic, and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

** Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 05

Sl No	Name	ID	PROGRAM	SIGNATURE
1	Shahriar Hossain	22-48990-3	BSc in CSE	
2	Al Mubtasim	22-49002-3	BSc in CSE	
3	Adiba Tanzila	22-49012-3	BSc in CSE	
4	MD.Rakib Hasan	22-49029-3	BSc in CSE	
5	Md.Imdadul Hasan(Ayon)	22-49959-3	BSc in CSE	

Faculty use only		
FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Assessment Materials and Marks Allocation:

COs	CO Statement	Assessment Materials	POIs	Marks
CO1	***Simulate*** *laboratory experiments using microcontrollers, sensors, actuators switches, display devices, etc., and a suitable simulator related to the fields of electrical and electronic engineering.*	Open Ended Laboratory Report	P.e.2.P4	15

Assessment Rubrics:

COs-POIs	Excellent [13-15]	Proficient [10-12]	Good [7-9]	Acceptable [4-6]	Unacceptable [1-3]	No Response [0]	Secured Marks
CO1 P.e.2.P4	The OEL was developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are demonstrated by combining all input patterns with several outcomes.	The OEL was developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are demonstrated with some outcomes and limited input patterns.	The OEL was developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are not demonstrated with some outcomes and input patterns.	The OEL was developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are not demonstrated with a few outcomes for a few patterns.	The OEL was developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. are not appropriate. The simulation and implementation processes are not demonstrated with any outcomes and not for any pattern.	No Response at all/copied from others/ identical submissions with gross errors/image file printed	
Comments					**Total marks (15)**		

CO/ CLO Number	CO/CLO Statement	K	P	A	Assessed Program Outcome Indicator	BNQF Indicator	Teaching-Learning Strategy	Assessment Strategy
1	**Simulate** laboratory experiments using microcontrollers, sensors, actuators switches, display devices, etc., and a suitable simulator related to the fields of electrical and electronic engineering.		P1, P4, P5		P.e.2.P4	FS.6	Practical Demonstration	OEL Report

Table of Contents

Objectives:

The objectives of this experiment are to:
• Design a basic fire detection system using Arduino Uno and a flame sensor.
• Understand the working principles of the flame sensor used with Arduino.
• Build a circuit that triggers an alarm (e.g., buzzer or LED) when a flame is detected.
• Learn how to interface sensors and output devices with Arduino for safety applications.

Theory and Methodology

Microcontrollers like the Arduino Uno interact with sensors and peripheral devices using various input/output mechanisms. In this experiment, we used a digital flame sensor that outputs either a LOW or HIGH signal depending on the presence of a flame. The sensor is connected to a digital pin (D3) configured as an input.

Unlike interrupt-based methods, this implementation continuously reads the sensor state using digitalRead(). When the flame is detected, the sensor outputs a LOW signal, which the Arduino reads and responds to immediately by activating the alarm indicators.

The LED connected to pin 13 serves as a visual alert, while the buzzer on pin 11 provides an audible alarm by simply switching its digital output HIGH or LOW (no PWM/tone generation used here). Serial communication via USART is established using Serial.begin(9600), allowing the system to log flame detection status in real-time on the Serial Monitor for monitoring and debugging purposes.

Although more advanced features like interrupts or PWM for buzzer tone generation are not implemented here, the design effectively demonstrates a straightforward digital sensor reading, output control, and serial logging approach for a basic fire detection system.

Arduino Code Explanation

- **Pin Declaration:**
  The flame sensor is connected to pin 3, the LED to pin 13, and the buzzer to pin 11:

```
int LED = 13;
int Flame_sensor = 3;
int Buzzer = 11;
int Flame_detected;
```

- **Setup Function:**
  In setup(), we initialize the pins: the flame sensor as INPUT, and the LED and buzzer as OUTPUT. Serial communication is started at 9600 baud for monitoring:

```
void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  pinMode(Flame_sensor, INPUT);
}
```

- **Loop Function:**
  The loop() continuously polls the flame sensor using digitalRead(). Since the sensor outputs LOW when flame is detected, the code checks if Flame_detected == LOW. If yes, the LED and buzzer are turned ON by setting their pins HIGH, and a warning message is printed to the Serial Monitor. Otherwise, the outputs are turned OFF and a safe message is logged:

```
void loop() {
  Flame_detected = digitalRead(Flame_sensor);

  Serial.print("Flame_detected = ");
  Serial.println(Flame_detected);

  if (Flame_detected == LOW) {
    Serial.println("Flame detected...! Take action immediately.");
    digitalWrite(LED, HIGH);
    digitalWrite(Buzzer, HIGH);
  } else {
    Serial.println("No Flame detected. Stay cool.");
    digitalWrite(LED, LOW);
    digitalWrite(Buzzer, LOW);
  }

  delay(500);
}
```

Apparatus:

1. Arduino IDE (2.0.1 or any recent version)
2. Arduino UNO
3. LED
4. Buzzer
5. Fire/Flame Sensor
6. Breadboard
7. Connecting Wires

Circuit Diagram:

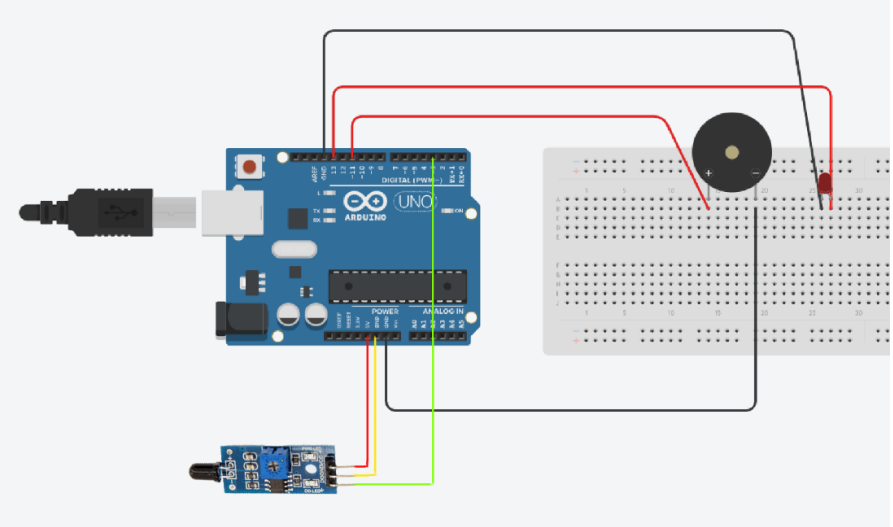

Fig : Flame Sensor with Arduino Uno and Buzzer (Schematic Diagram)

Experimental Output Results:

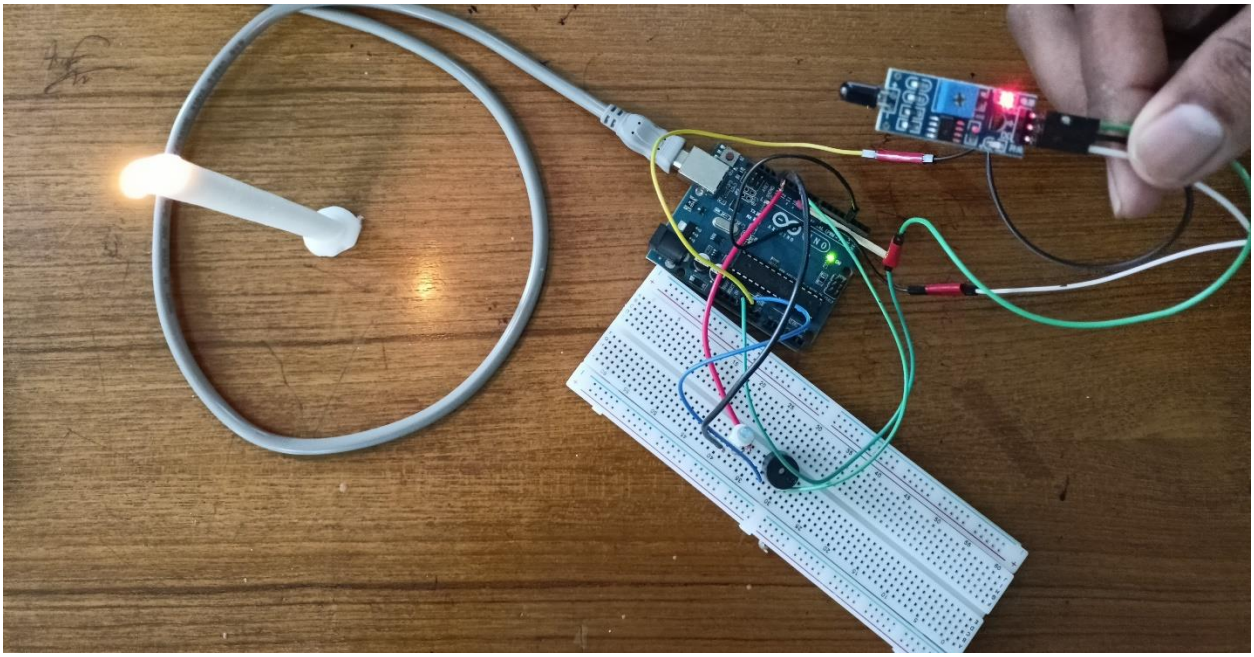

Figure: Flame Sensor Hardware Implementation(Fire Far from Sensor)

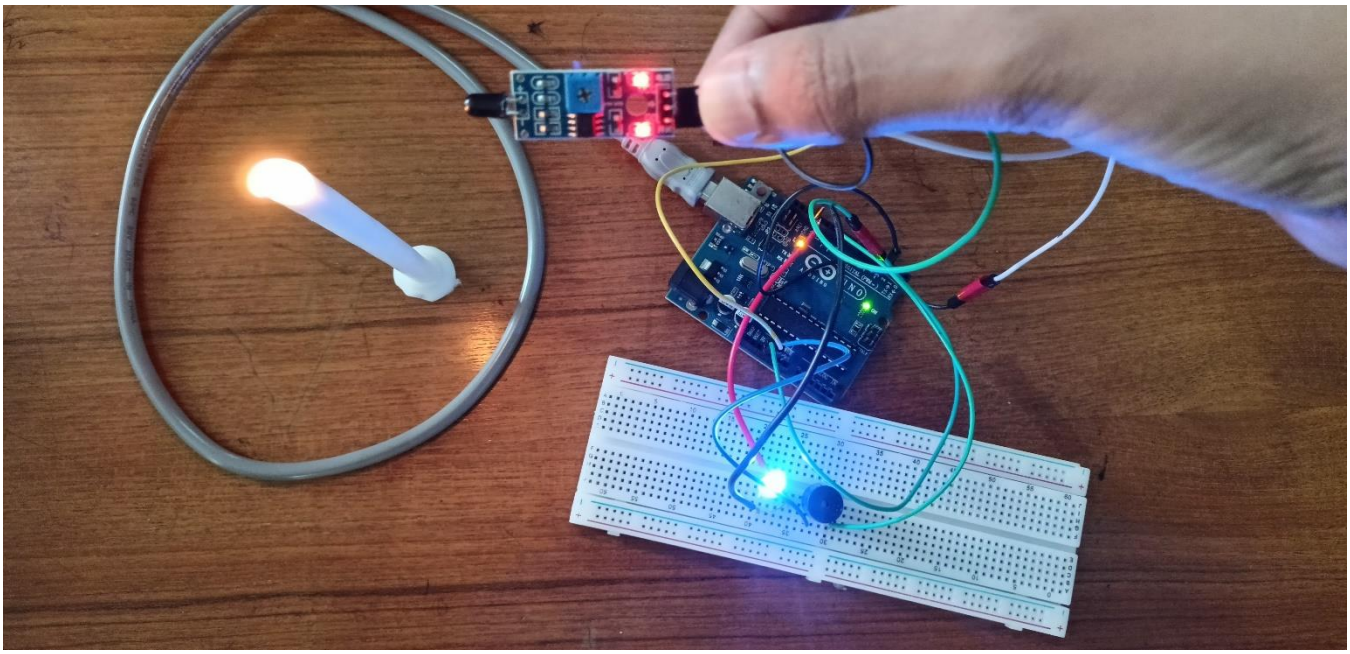

Figure: Flame Sensor Hardware Implementation(Fire near Sensor)

Simulation Output Results:

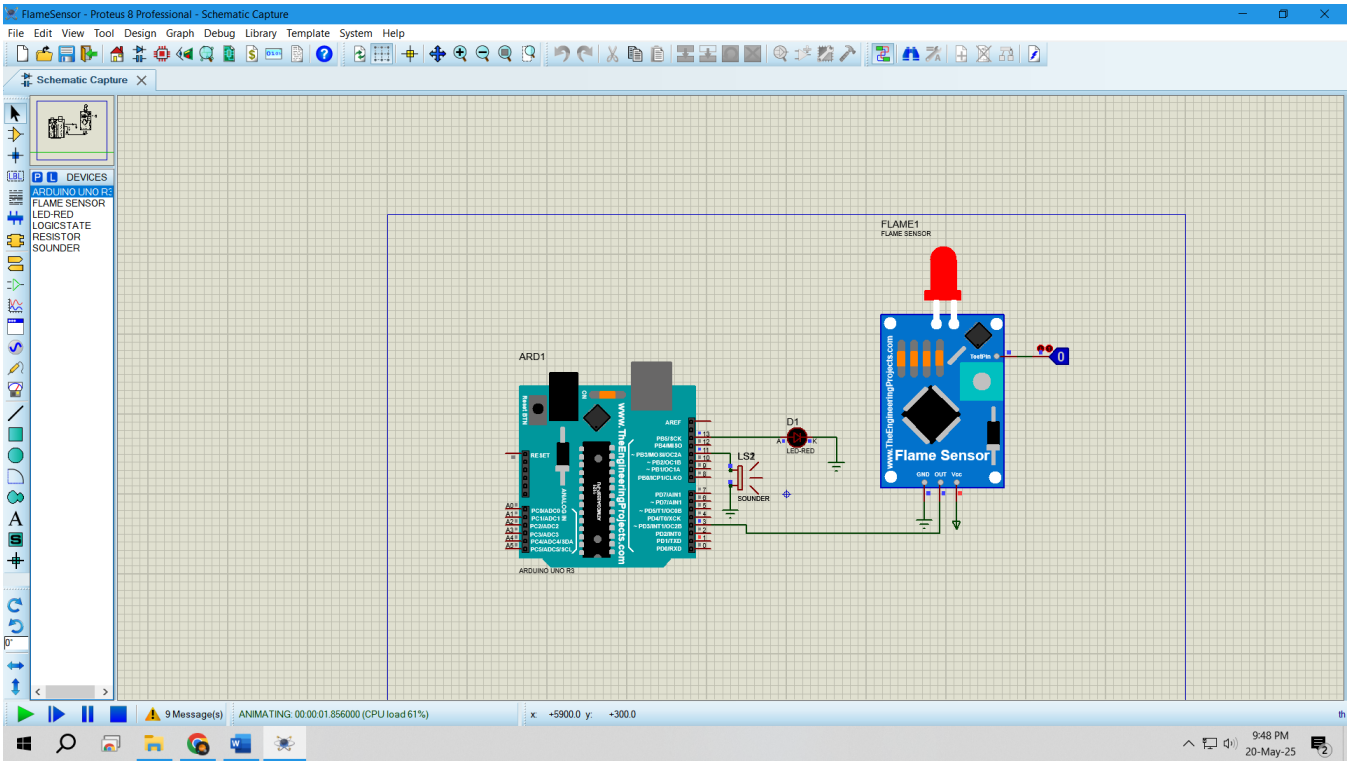

Figure: Flame Sensor with 0 input using Logicstate,Buzzer and LED off

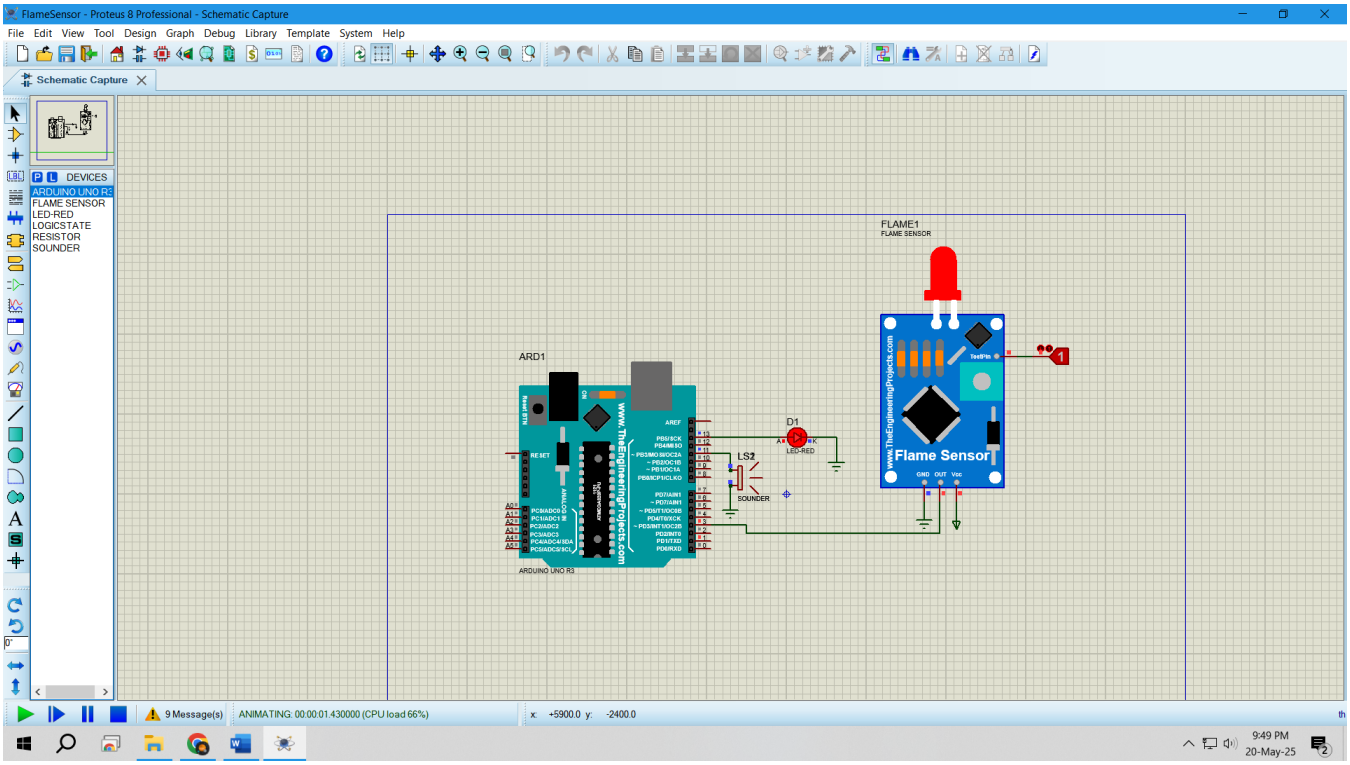

Figure: Flame Sensor with 1 input using Logicstate,Buzzer and LED on

Explaination:

- Proteus was opened, and a new project was created.
- Required components such as Arduino Uno,LED, Buzzer, Logicstate and Flame Sensor were added from the library.
- The components were connected accordingly.
- The arduino codes were compiled and HEX file was created.
- The HEX file was loaded into the Arduino Unos in Proteus.
- The simulation was run, and the LED behavior for both high and low input were observed.

Code of the Program:

```
int LED = 13;              // LED pin
int Flame_sensor = 3;      // Flame sensor digital output pin
int Buzzer = 11;           // Buzzer pin
int Flame_detected;        // Stores the sensor reading

void setup()
{
  Serial.begin(9600);              // Start Serial Monitor
  pinMode(LED, OUTPUT);            // Set LED pin as OUTPUT
  pinMode(Buzzer, OUTPUT);         // Set Buzzer pin as OUTPUT
  pinMode(Flame_sensor, INPUT);    // Set flame sensor pin as INPUT
}

void loop()
{
  Flame_detected = digitalRead(Flame_sensor);  // Read sensor
  Serial.print("Flame_detected = ");
  Serial.println(Flame_detected);

  if (Flame_detected == LOW)  // Flame detected (LOW signal)
  {
    Serial.println("Flame detected...! Take action immediately.");
    digitalWrite(LED, HIGH);       // Turn on LED
    digitalWrite(Buzzer, HIGH);    // Turn on Buzzer
  }
  else
  {
    Serial.println("No Flame detected. Stay cool.");
    digitalWrite(LED, LOW);        // Turn off LED
    digitalWrite(Buzzer, LOW);     // Turn off Buzzer
  }

  delay(500);  // Delay for readability and debounce
}
```

Table for Simulation:

Test No.	Input	LED	Buzzer	Overview
1	0	OFF	OFF	Both stayed turned off.
2	1	ON	ON	Both turned on.

Table for Hardware:

Test No.	Input	LED	Buzzer	Overview
1	No Flame	OFF	OFF	Both stayed turned off.
2	Flame	ON	ON	Both turned on.

Discussions:

The experiment successfully demonstrated a basic fire detection system using an Arduino Uno, flame sensor, LED, and buzzer. The flame sensor provided a digital output, which the Arduino read using digitalRead(). Upon detecting a flame (LOW signal), the system activated the LED and buzzer using digitalWrite(). When no flame was detected (HIGH signal), both outputs were turned off. The system functioned as expected during real-world testing, confirming the effectiveness of simple digital input/output operations. While no advanced protocols like SPI or I2C were used, the system did utilize USART through serial communication for real-time status updates and debugging. This experiment enhanced our understanding of interfacing digital sensors with Arduino and highlighted how basic components can be effectively combined to create a low-cost and reliable alert system for early fire detection.

Reference(s):

[1] https://www.arduino.cc/.
[2] ATMega328 manual
[3] https://circuitdigest.com/microcontroller-projects/interfacing-flame-sensor-with-arduino