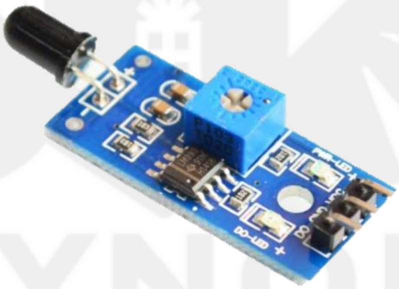
LAUNCH THE PROJECT - 11

Fire Detector

(via Flame Sensor)

On AYNOP® UNO Launchpad Kit



Author: AYNOP Enterprises | Doc Version: 1.0 | Date: 19-9-2025

Table of Contents

1. Overview

This project demonstrates how to detect the presence of a **Flame** using a 3-pin Flame Sensor module with the Arduino® UNO R4 Minima.

You will learn to:

- Connect a flame sensor to the UNO R4 Minima.
- Use digital input for Flame detection.
- Implement **filtering (100 ms confirmation)** and **state-change detection**.
- Apply a **lockout period (5 seconds)** after detecting a flame to avoid repeated triggers.
- Control the onboard LED (Pin 13) to follow the flame detection state

2. Components Required

- Arduino® UNO R4 Minima board
- USB Type-C data cable
- Flame Sensor Module (3-pin, digital output version)
- Breadboard
- Jumper wires

3. Software Required

- Arduino IDE (v2.3.6 or later recommended)
- No additional libraries required**.**

Note:

We assume the **Arduino UNO R4 Minima board package** is already installed on your machine, as explained in the 00_Getting_Started/00_GettingStarted_Arduino_R4_Minima guide. If it is not installed, please refer to that document and complete the installation before proceeding.

4. Hardware Setup

This section explains how to connect the components for Flame detection.

4.1 Wiring Diagram

- Flame Sensor VCC → 5V.
- Flame Sensor GND → GND**.**
- Flame Sensor DO (Digital Output) → D2.
- Onboard LED is internally connected to D13.

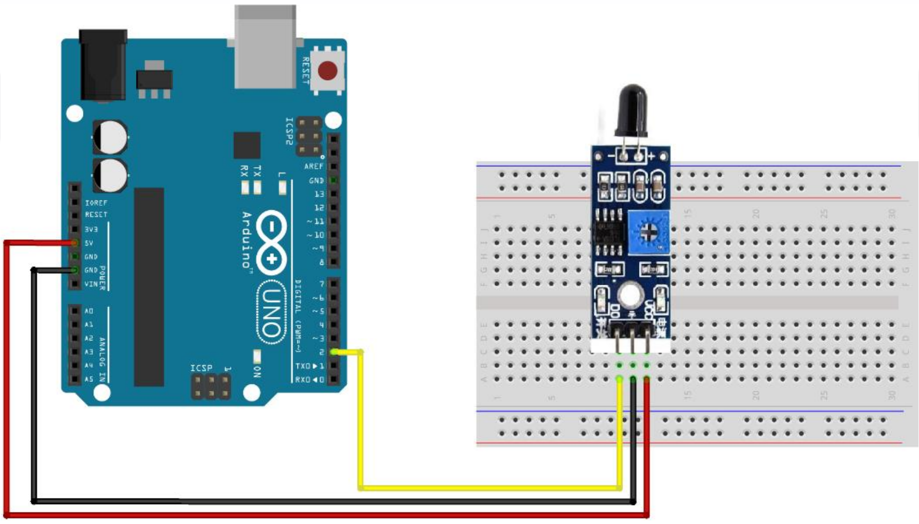


Figure 4.1 – Wiring diagram for Flame Detection project

💡 **Tip:** Always disconnect the USB cable before making or changing hardware connections.

5. Principle – How It Works

The 3-pin flame sensor module detects infrared light emitted by a Flame.

- When a Flame is present, the **DO pin outputs LOW**.
- When no Flame is present, the **DO pin outputs HIGH.**

In this project:

- A **confirmation delay (100 ms)** ensures stable readings and avoids false triggers.
- The system prints messages **only when the state changes**.
- A **lockout period of 5 seconds** is applied after detection to prevent repeated triggers.
- The onboard LED on pin 13 mirrors the detection state

6. Procedure – Steps to Run

1. **Build the Circuit**
    a. Connect the flame sensor as shown in the **Wiring Diagram (Figure 4.1)**.

2. **Connect the Board**
    a. Use a USB Type-C data cable to connect your UNO R4 Minima to your computer.

3. **Open the Project Code**
    a. Simply **double-click** the file *11_Fire_Detector.ino* in the project folder, and it will open directly in the Arduino IDE (if installed).

4. **Confirm Board Selection**
    a. The IDE usually auto-detects the UNO R4 Minima if the package is installed.
    b. If not installed, refer to the *00_Getting_Started/00_GettingStarted_Arduino_R4_Minima* document to install the necessary board package.
    c. Verify that *Arduino UNO R4 Minima* is selected in the IDE's board selector (top toolbar).

5. **Upload the Code**
    a. Click the **Upload** button (arrow icon) in the top-left corner of the IDE.
    b. Wait until the console displays "**Done uploading**."

6. **Observe the Behaviour**
    a. After uploading, bring a **small flame source** (like a candle or lighter) close to the sensor.
    b. The onboard LED (D13) will turn **ON** when the flame is detected and **OFF** when no flame is present
    c. Observe flame detection messages in real time **on** Serial Monitor.
    d. Move the flame closer to observe how the sensor responds.
    e. The MCU waits for about **5 seconds (lockout)** after a detection before allowing a new one, so don't worry if it doesn't trigger repeatedly right away.

💡 Tip: Always test safely — keep flames small and remove immediately after use.

6.1 How to Use the Serial Monitor:

The Serial Monitor is a **tool in the Arduino IDE** that lets you see text/data sent by the microcontroller (MCU) over the USB connection.

- In Arduino IDE, go to **Tools → Serial Monitor**, or press **Ctrl + Shift + M**.
- A new window will open showing real-time data from your UNO R4 Minima.
- At the bottom-right of the Serial Monitor window, set the baud rate to **9600**. This must match the *Serial.begin*(9600) in the code.

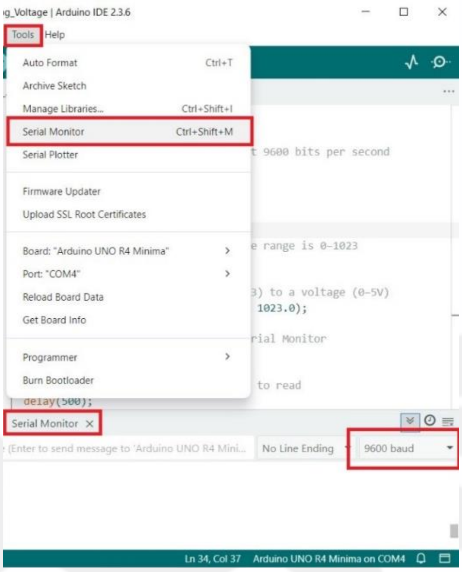


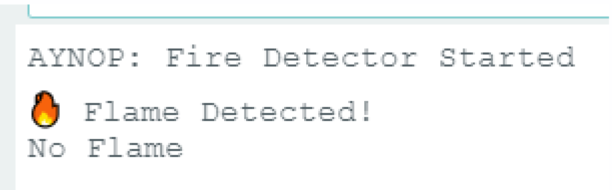
Figure 6.1 – Arduino IDE Serial Monitor set to 9600 baud

💡 **Tip:** If the Serial Monitor shows nothing, check that:

- The correct **COM port** is selected.
- The baud rate is set to **9600**

7. Expected Output

- The onboard LED (D13) turns ON when a flame is detected and OFF when no flame is present and The Serial Monitor will show data like this:



```
AYNOP: Fire Detector Started
🔥 Flame Detected!
No Flame
```

8. Code

The source code for this project is included in the downloaded folder:

📁 uno-launchpad-kit/01_Basic_Projects/11_Fire_Detector/11_Fire_Detector.ino

👉 **Tip:**

- To open the project, simply **double-click the .ino file**. If the Arduino IDE is installed, it will launch automatically and load the code.
- If you **haven't installed the Arduino IDE yet**, please refer to:
  📁 uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima
  to **download and install it.**

8.1 Function References

- setup() – runs once when the board is powered on or reset.
- loop() – runs continuously after setup() finishes.
- digitalRead(pin) – reads HIGH/LOW state from sensor pin.
- digitalWrite(pin, state) – controls onboard LED.
- Serial.begin(9600) – starts serial communication.
- Serial.println(data) – prints messages to Serial Monitor.
- delay(ms) – short pauses for filtering and loop control.
- millis() – provides system time in milliseconds (used for lockout).

📚 **For more details and advanced usage, visit:**

🔗 Arduino Language Reference — The official guide for all Arduino functions.

9. Troubleshooting Tips

- **No output in Serial Monitor?**
  - Check the baud rate (must be 9600).
  - Ensure Serial Monitor is open after upload.

- **Always shows “Flame Detected”?**
  - Adjust sensor sensitivity using the onboard potentiometer.
  - Move sensor away from bright IR sources (like sunlight).

- **Upload error in Arduino IDE?**
  - Verify that the correct board (**Arduino UNO R4 Minima**) is selected in the IDE.
  - Check that the correct **COM port** is chosen.

- **Board not detected vis USB?**
  - Ensure you are using a **data-capable USB Type-C cable** (some cables only provide charging).
  - Try reconnecting the cable or using a different USB port.

- **Board not listed in Arduino IDE?**
  - If you don’t see **Arduino UNO R4 Minima** in the board selector, the **board package is not installed**.

  - To fix this, follow the installation steps in:
    📂 *uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima*

💡 **Tip:** If nothing works, press the **RESET** button on the UNO R4 Minima and try uploading the code again.

10. License

11. Support & Feedback

We value your feedback and are happy to assist with any questions, troubleshooting, or suggestions you may have.

✉ Email: *support@aynop.com*

When sending an email, please include your kit name (AYNOP® UNO Launchpad Kit) and, if applicable, the project name in the subject line. This will help our team respond faster and more accurately. We aim to respond to all queries within 2–3 business days. Your feedback helps us improve our products and create even better learning experiences for all Arduino beginners.