

EMBEDDED FIRE DETECTION AND ALARM NOTIFICATION SYSTEM

TEAM MEMBERS:

1. Subani Sundaram
2. Vaanmathi G
3. Srimathi R
4. Saambhavi K
5. Yuvalakshime S

AIM :

To design and develop an **embedded fire detection and alarm notification system** that can accurately sense the presence of fire (using temperature, smoke, or flame sensors), trigger an immediate **local alarm** (buzzer/siren/LED) and send **remote notifications** (via SMS, call, or IoT/cloud platform) to ensure **quick response** and **minimize damage** in residential, commercial, or industrial environments.

COMPONENTS :

1. Raspberry Pi Pico W
2. OLED
3. Fire sensor
4. Buzzer

1. Raspberry Pi Pico W

- **Definition:**
A low-cost, low-power microcontroller board from Raspberry Pi with Wi-Fi capability for IoT applications.
- **Range:**
Wi-Fi range: typically up to 30 meters indoors, more in open space (depends on router).
- **Application:**
 - Acts as the brain of the system.
 - Reads data from sensors, processes it, triggers alarms.
 - Sends notifications (SMS, email, cloud alerts) via Wi-Fi.

2. OLED Display

- **Definition:**
A small, energy-efficient display (commonly 128×64 pixels) for showing text or graphics.

- **Range:**
Viewing range: readable from ~1–2 meters comfortably (depends on size).
- **Application:**
 - Displays real-time fire/smoke levels, temperature, or system status.
 - Can show alert messages

3. Fire Sensor

- **Definition:**
A sensor that detects fire, smoke, or flame. Types include:
 - Flame sensors (IR or UV-based)
 - Smoke sensors (e.g., MQ-2, MQ-135)
 - Temperature sensors (e.g., DHT11, LM35)
- **Range:**
 - IR flame sensor: typically detects flames up to ~80–100 cm.
 - Smoke sensor: effective within ~1–3 meters radius.
- **Application:**
 - Detects early signs of fire (heat, smoke, or flame).
 - Provides input to the microcontroller to trigger alarms and notifications.
 - Provides quick visual alert in case of fire.

4. Buzzer

- **Definition:**
A small audio signaling device that converts electrical signals into sound. Can be **active** (plays tone when powered) or **passive** (needs a frequency signal).
- **Range:**
 - Typically **70–85 dB**, audible up to ~**3–10 meters** depending on environment and type.

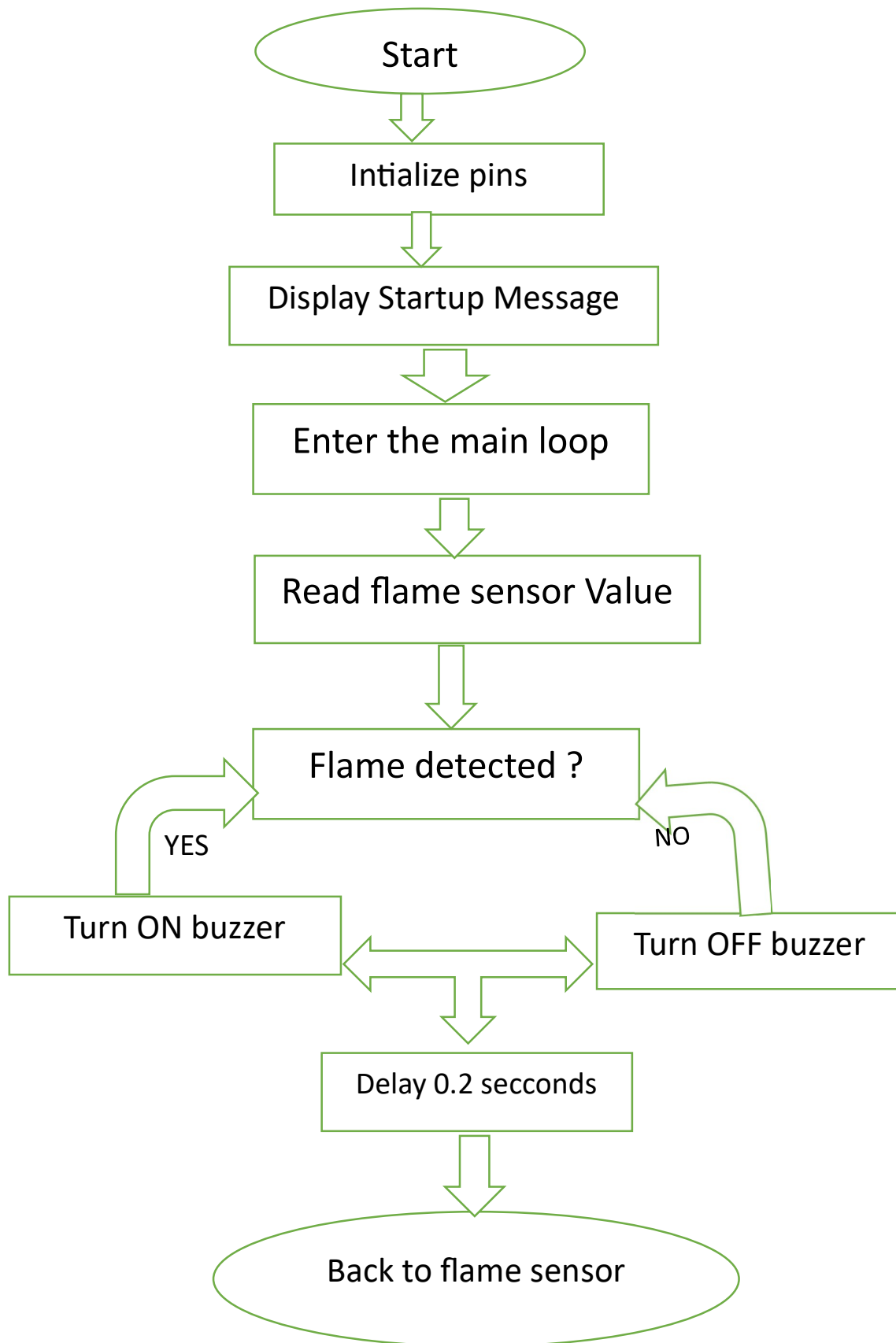
- **Application:**

- Provides **audible alarm** when fire is detected.
- Warns people nearby even if they're not looking at the LED or OLED.
- Can use **different beep patterns** for warning vs. confirmation (e.g., continuous for fire).

PIN TABLE :

Components	Pin on module	Connect to
Fire Sensor	OUT	GP15
	VCC	3V3
	GND	GND
Buzzer	+ (Positive)	GP16
	-(Negative)	GND
OLED	SCL	GP1
	SDA	GP0
	VCC	3V3
	GND	GND

FLOWCHART :



EXECUTION :

The screenshot shows the Thonny IDE interface. The top bar indicates the file is 'fire.py' on a Raspberry Pi Pico. The left sidebar shows the file explorer with 'fire.py' selected. The main editor displays the following Python code:

```

19 time.sleep(4)
20
21 # == Main Loop ==
22 while True:
23     if flame_sensor.value() == 0:  # LOW means Fire detected
24         # Buzzer ON
25         oled.fill(0)
26         oled.text("🔥 FIRE ALERT!", 0, 20)
27         oled.show()
28         print("FIRE ALERT!")
29         print(flame_sensor.value())
30     else:
31         buzzer.value(1)  # Buzzer OFF
32         oled.fill(0)
33         oled.text("SAFE", 40, 20)
34         oled.show()
35         print("SAFE")
36         print(flame_sensor.value())
37
38 time.sleep(0.2)
39
40

```

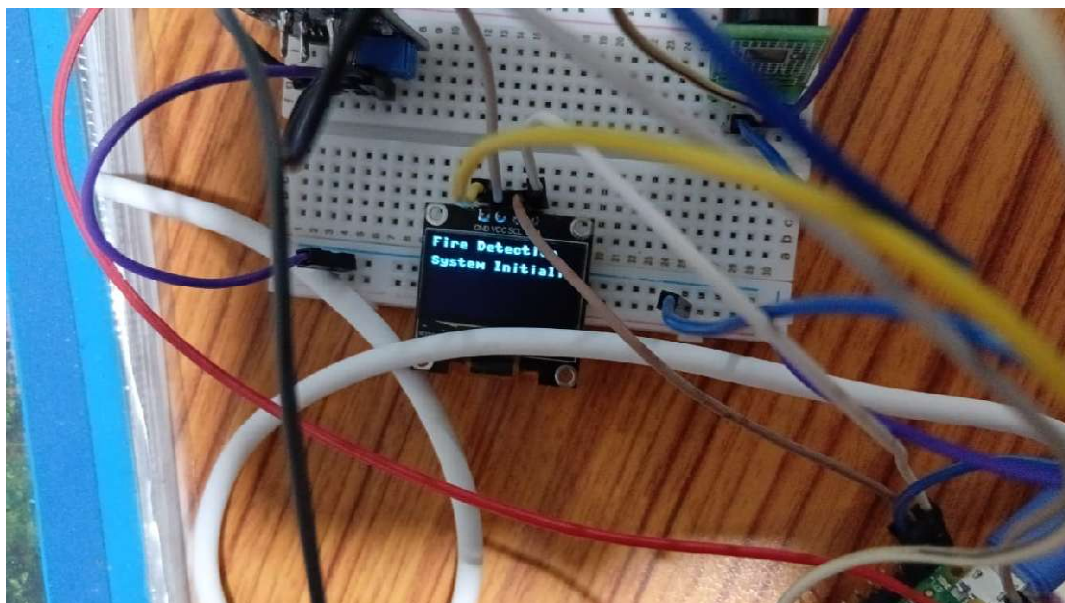
The bottom panel shows the Shell output, which contains the following text:

```

1
SAFE
1
SAFE
1
SAFE
1
SAFE
1
SAFE

```

The status bar at the bottom right indicates 'MicroPython (Raspberry Pi Pico) • Board CDC @ COM4' and the time '13:21 04-09-2023'.



Thonny - Raspberry Pi Pico 2 / fire.py @ 32:21

File Edit View Run Tools Help

Files

This computer
C:\Users\ADMIN

- Contacts
- Documents
- Downloads
- Favorites
- Links
- Music
- OneDrive
- Saved Games
- Searches
- Videos
- VirtualBox VMs
- fire.py
- from transformers ir
- main.py
- project
- project 2

Raspberry Pi Pico

- lib
- fire.py
- gasensor.py
- main.py
- oled.py
- sh1106.py
- ssd1306.py

```
19 time.sleep(4)
20
21 # === Main Loop ===
22 while True:
23     if flame_sensor.value() == 1: # LOW means fire detected
24         buzzer.value(1) # Buzzer ON
25         oled.fill(0)
26         oled.text("🔥 FIRE ALERT!", 0, 20)
27         oled.show()
28         print("FIRE ALERT!")
29         print(flame_sensor.value())
30     else:
31         buzzer.value(0) # Buzzer OFF
32         oled.fill(0)
33         oled.text("SAFE", 40, 20)
34         oled.show()
35         print("SAFE")
36         print(flame_sensor.value())
37
38 time.sleep(0.2)
39
40
```

Shell

```
1
1 FIRE ALERT!
1
1 FIRE ALERT!
1
1 FIRE ALERT!
1
1 FIRE ALERT!
1
```

MicroPython (Raspberry Pi Pico) • Board CDC @ COM4

34°C Partly sunny

Search

ENG IN 13:18 04-09-2025