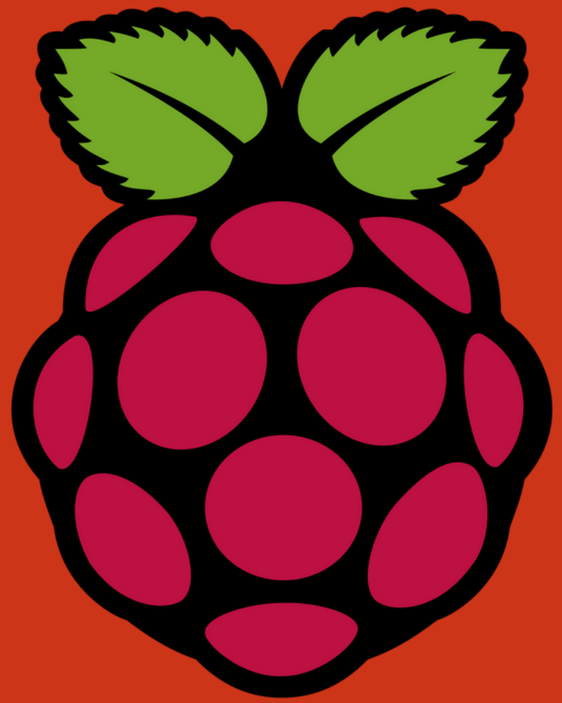


Multi-Processing Fire Detection System using Raspberry Pi

II Year CSE-A

Our Team

Aman Kshetri

CH.EN.U4CSE21003

Raj Sah Rauniyar

CH.EN.U4CSE21051

Saroj Kumar Bhagat

CH.EN.U4CSE21060

Tej Mahesh

CH.EN.U4CSE21041

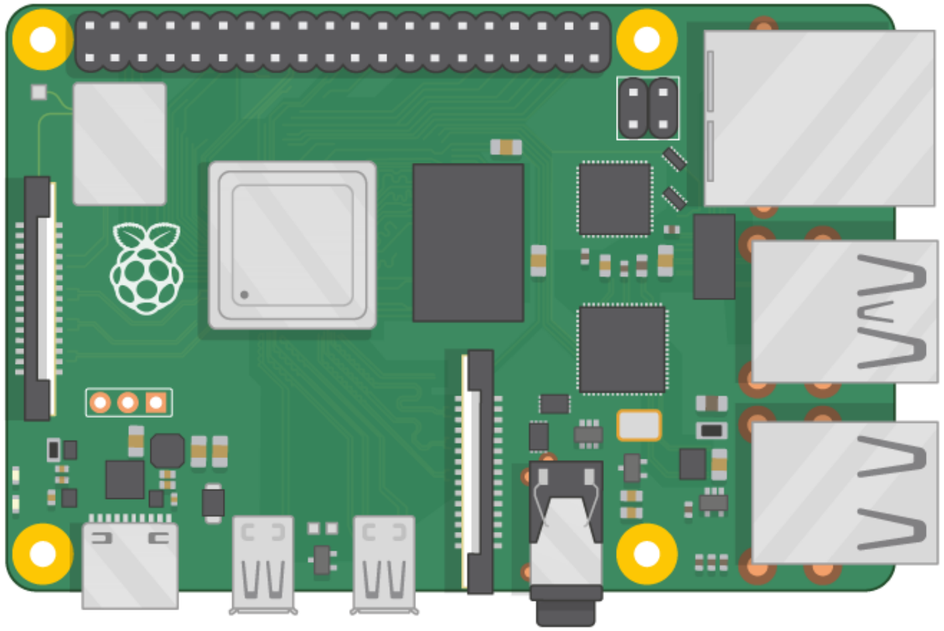
Problem Definition

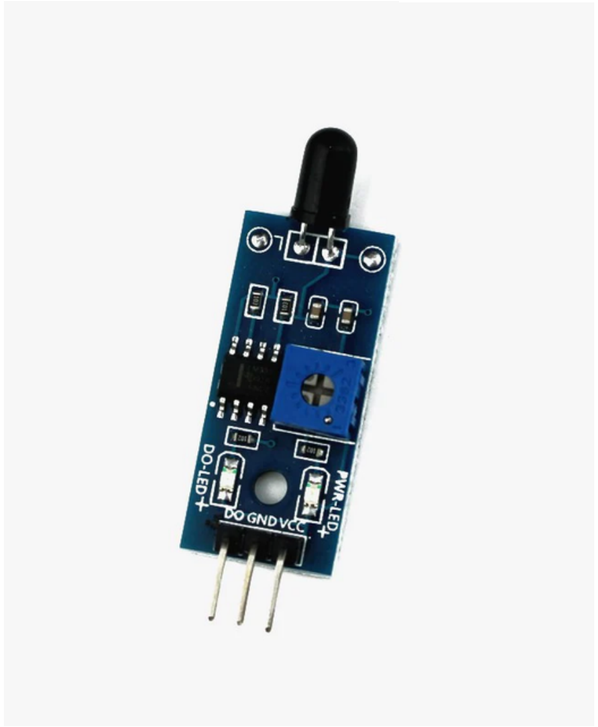
The problem we are trying to solve with this project is to detect fires in real-time using a Raspberry Pi-based system.

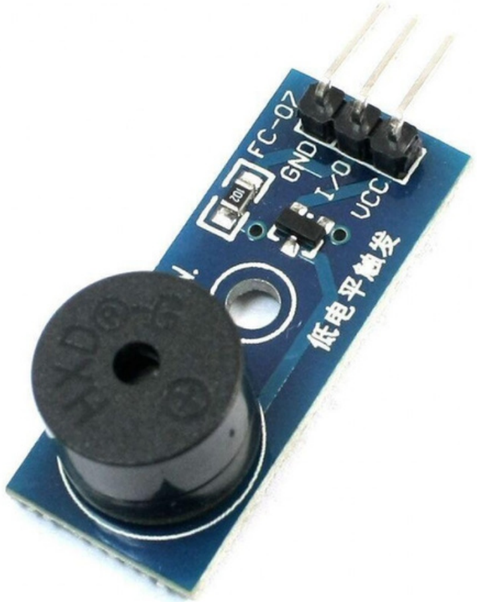
Fires can cause significant damage to property and loss of life if not detected and addressed quickly.

Our project aims to detect fires as quickly as possible using sensors connected to a Raspberry Pi, which will alert users through a buzzer and notification messaging and allows to take appropriate action.

Components

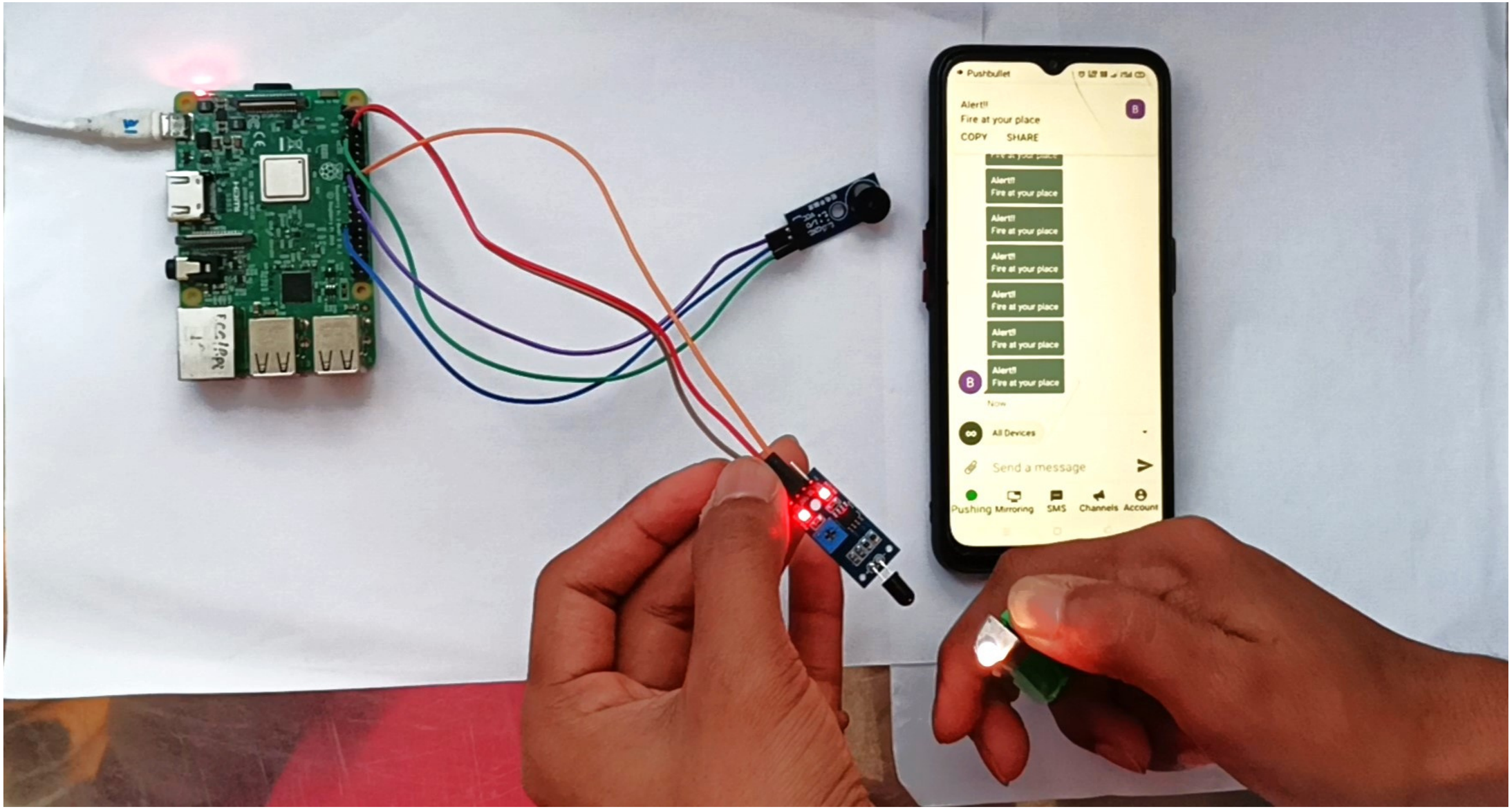






Methodology

- The methodology for this project involves using a Raspberry Pi and a fire sensor to detect fires in real-time.
- The fire sensor will detect the presence of flames and send a signal to the Raspberry Pi.
- If a fire is detected, the Raspberry Pi will activate a buzzer to alert users to the danger and send a notification message to the user's devices.
- In this methodology, Pushbullet API's will be used to connect the notification messaging system to the Raspberry Pi, which will ensure prompt delivery of notifications.



Implementation

Code

```python
import time
import RPi.GPIO as GPIO
from pushbullet import Pushbullet
from multiprocessing import Process, Value

flame_sen = 18
buzzer = 29
fire_detected = Value('i', 0)

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(flame_sen, GPIO.IN)
GPIO.setup(buzzer, GPIO.OUT)

pb = Pushbullet("o.59hrJyB39l75SDxN4tnxWh9XoYtp1GoE")
print(pb.devices)

def detect_flame(fire_detected):
    while True:
        if GPIO.input(flame_sen) == GPIO.LOW:
            print("Fire detected!")
            fire_detected.value = 1
        else:
            fire_detected.value = 0
        time.sleep(0.1)

def alert(pb):
    while True:
        if fire_detected.value == 1:
            print("Sending push notification...")
            GPIO.output(buzzer, GPIO.LOW)
            dev = pb.get_device('OPPO CPH1931')
            push = dev.push_note("Alert!!", "Fire at your place")
        else:
            GPIO.output(buzzer, GPIO.HIGH)
        time.sleep(0.1)

if __name__ == '__main__':
    p1 = Process(target=detect_flame, args=(fire_detected,))
    p2 = Process(target=alert, args=(pb,))
    p1.start()
    p2.start()
    p1.join()
    p2.join()
```

THANK YOU !