## *CONTENTS:*

* INTRODUCTION
* APPLICATIONS OF FLAME SENSOR
* COMPONENTS REQUIRED
* CIRCUIT DIAGRAM
* WORKING OF FLAME SENSOR WITH ARDUINO
* CODE EXPLANATION
* BIBLIOGRAPHY

***INTRODUCTION:***

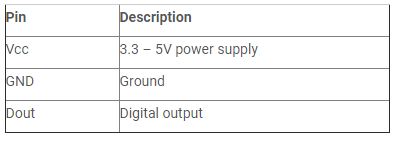
A **flame detector** is a sensor designed to detect and respond to the presence of a flame or fire. Responses to a detected flame depend on the installation, but can include sounding an alarm, deactivating a fuel line (such as a propane or a natural gas line), and activating a fire suppression system.

There are different types of flame detection methods. Some of them are: Ultraviolet detector, near IR array detector, infrared (IR) detector, Infrared thermal cameras, UV/IR detector etc.

When fire burns it emits a small amount of Infra-red light, this light will be received by the Photodiode (IR receiver) on the sensor module. Then we use an Op-Amp to check for change in voltage across the IR Receiver, so that if a fire is detected the output pin (DO) will give 0V(LOW) and if the is no fire the output pin will be 5V(HIGH).

In this project we are using an **IR based flame sensor**. It is based on the YG1006 sensor which is a high speed and high sensitive NPN silicon phototransistor. It can detect infrared light with a wavelength ranging from 700nm to 1000nm and its detection angle is about 60°.  Flame sensor module consists of a photodiode (IR receiver), resistor, capacitor, potentiometer, and LM393 comparator in an integrated circuit. The sensitivity can be adjusted by varying the on board potentiometer. Working voltage is between 3.3v and 5v DC, with a digital output. Logic high on the output indicates presence of flame or fire. Logic low on output indicates absence of flame or fire.

Below is the Pin Description of Flame sensor Module:



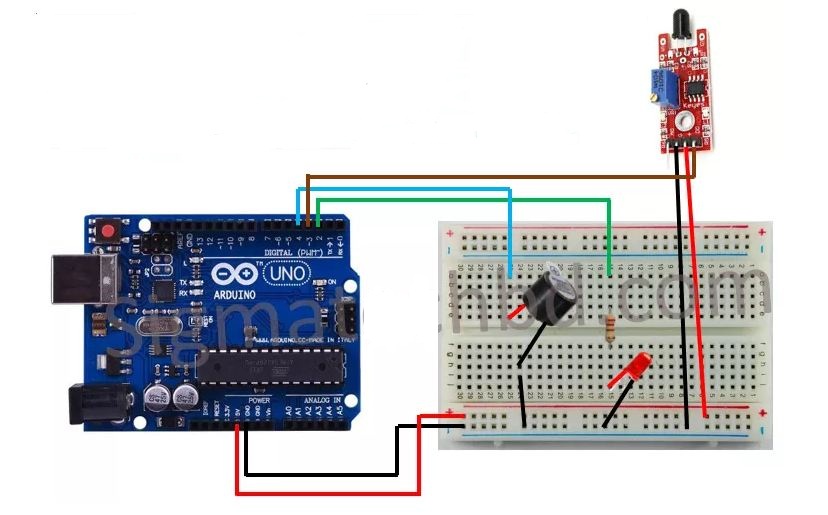
***Applications of flame sensors:***

* Hydrogen stations
* Combustion monitors for burners
* Oil and gas pipelines
* Automotive manufacturing facilities
* Nuclear facilities
* Aircraft hangars

### *****Components Required:*****

* Arduino Uno (any Arduino board can be used)
* Flame sensor
* LED
* Buzzer
* Resistor
* Jumper wires

### *****Circuit Diagram:*****



***Working of Flame Sensor with Arduino:***

Arduino Uno is a open source microcontroller board based on ATmega328p microcontroller. It has 14 digital pins (out of which 6 pins can be used as PWM outputs), 6 analog inputs, on board voltage regulators etc. Arduino Uno has 32KB of flash memory, 2KB of SRAM and 1KB of EEPROM. It operates at the clock frequency of 16MHz. Arduino Uno supports Serial, I2C, SPI communication for communicating with other devices. The table below shows the technical specification of Arduino Uno.

|  |  |
| --- | --- |
| Microcontroller | ATmega328p |
| Operating voltage | 5V |
| Input Voltage | 7-12V (recommended) |
| Digital I/O pins | 14 |
| Analog pins | 6 |
| Flash memory | 32KB |
| SRAM | 2KB |
| EEPROM | 1KB |
| Clock speed | 16MHz |

The **flame sensor detects the presence of fire** or flame based on the Infrared (IR) wavelength emitted by the flame. It gives logic 1 as output if flame is detected, otherwise it gives logic 0 as output. Arduino Uno checks the logic level on the output pin of the sensor and performs further tasks such as activating the buzzer and LED, sending an alert message.

### *****Code explanation:*****

The complete **Arduino code** for this project is given at the end of this article. The code is split into small meaningful chunks and explained below.

In this part of the code we are going to **define pins for Flame sensor, LED and buzzer** which are connected to Arduino. Flame sensor is connected to digital pin 3 of Arduino. Buzzer is connected to digital pin 4 of Arduino. LED is connected to digital pin 2 of Arduino.

Variable “flame\_detected” is used for storing the digital value read out from flame sensor. Based on this value we will detect the presence of flame.

**int LED=2;**

**int BUZZER=4;**

**int sensor=3;**

**int flame\_detected ;**

In this part of the code, we are going to **set the status of digital pins of Arduino** and configure

Baud rate for Serial communication with PC for displaying status of flame detection circuit.

**void setup()**

**{**

**Serial.begin(9600) ;**

**pinMode(buzzer, OUTPUT) ;**

**pinMode(LED, OUTPUT) ;**

**pinMode(flame\_sensor, INPUT) ;**

**}**

This line of code reads the digital output from flame sensor and stores it in the variable “flame\_detected”.

**flame\_detected = digitalRead(flame\_sensor) ;**

Based on the value stored in “flame\_detected”, we have to turn on the buzzer and LED. In this part of the code, we **compare the value stored in “**flame\_detected**” with 0 or 1**.

**If its equal to 1**, it indicates that flame has been detected. We have to turn on buzzer and LED and then display an alert message in Serial monitor of Arduino IDE.

**If its equal to 0**, then it indicates that no flame has been detected so we have to turn off LED and buzzer. This process is repeated every second to identify the presence of fire or flame.

**if (flame\_detected == 1)**

**{**

**Serial.println("Danger");**

**digitalWrite(buzzer, HIGH);**

**digitalWrite(LED, HIGH);**

**delay(200);**

**digitalWrite(LED, LOW);**

**delay(200);**

**}**

**else**

**{**

**Serial.println("It’s Ok");**

**digitalWrite(buzzer, LOW);**

**digitalWrite(LED, LOW);**

**}**

**delay(1000);**

**BIBLIOGRAPHY:**

* [**https://www.elprocus.com/types-analog-digital-sensors/**](https://www.elprocus.com/types-analog-digital-sensors/)
* [**https://en.wikipedia.org/wiki/Flame\_detector**](https://en.wikipedia.org/wiki/Flame_detector)
* [**https://circuitdigest.com/microcontroller-projects/arduino-flame-sensor-interfacing**](https://circuitdigest.com/microcontroller-projects/arduino-flame-sensor-interfacing)