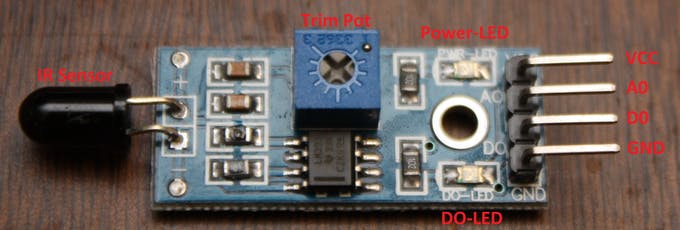
**Flame Sensor**

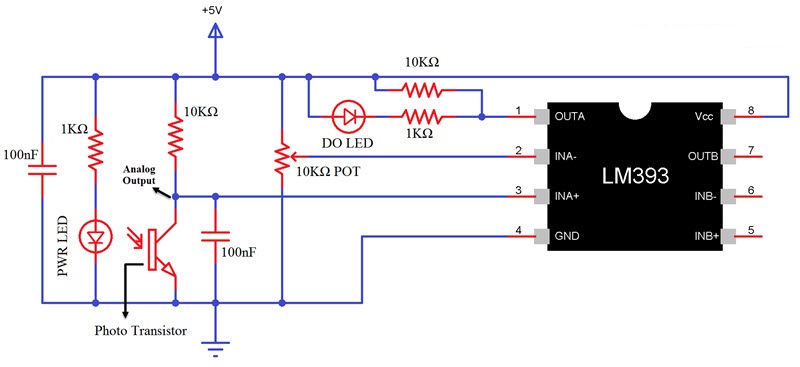
A **flame detector** is a sensor designed to detect and respond to the presence of a flame or fire. Responses to a detected flame depend on the installation but can include sounding an alarm, and activating a fire suppression system, or sending message or Fire alert using GSM. The IR Flame sensor used in this project is shown below, these sensors are also called **Fire sensor module** or **flame detector sensor** sometimes.



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 a 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°.  The 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 onboard potentiometer. Working voltage is between 3.3v and 5v DC, with a digital output. A logic high on the output indicates the presence of flame or fire. A logic low on output indicates the absence of flame or fire.



Below is the Pin Description of the Flame sensor Module:

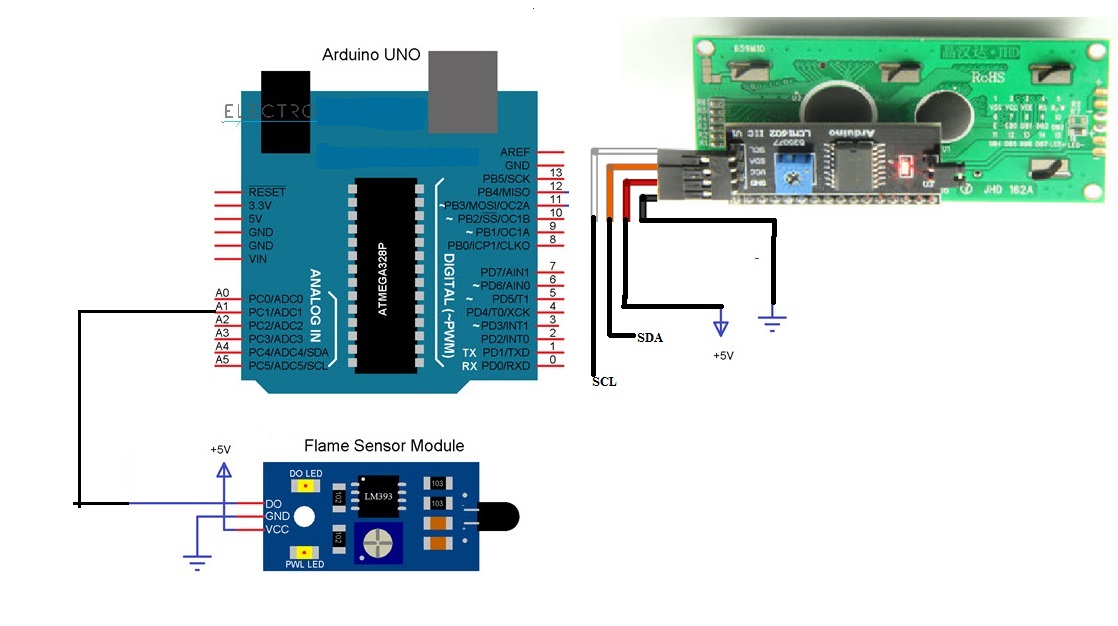
|  |  |
| --- | --- |
| **Pin** | **Description** |
| Vcc | 3.3 – 5V power supply |
| GND | Ground |
| Dout | Digital output |

**Components Required**

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

**Circuit Diagram of Arduino with Flame Sensor and LCD**





I2C LCD (16X2)

**Working of Flame Sensor with Arduino**

[Arduino](https://circuitdigest.com/arduino-projects) Uno is a open-source microcontroller board based on the 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 a 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 a 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.

Flame Sensor has three pins (some may have four pins): VCC, GND and DO. Connect VCC and GND to +5V and GND of the power supply (can be connected to Arduino’s +5V). the DO (short for Digital Output) is connected to Analog I/O Pin A1 of Arduino(used as Digital input with pull up resistor).

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

**#define flame\_sensor A1**

**int flame\_detected ;**

**we are interfacing flame sensor and also LCD(16X2)with Arduino. As LCD interfacing is explained ,we will concentrate on only flame sensor interfacing**

**void setup()**

**{**

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

**}**

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

**Flame\_Val = digitalRead(flame\_sensor) ;**

Based on the value stored in “F*lame\_Val*”, we have to display message on the LCD. In this part of the code, we **compare the value stored in “*flame\_detected*” with 0 or 1**.

**If its equal to 0**, 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 1**, 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 == 0)**

**{**

**Message\_Str="Fire Alert At Location: ";**

**lcd.clear();//Clean the screen**

**lcd.setCursor(0, 0);**

**lcd.print("Fire Detected ");**

**delay(1000);**

**}**

**delay(1000);**

**Procedure**

1. Download and Install the Library **LiquidCrystal\_I2c** and include **wire.h library**.
2. Write the code for the Arduino UNO using Arduino IDE.
3. Connect the components as the circuit attached.
4. Edit the code as need to change the text on LCD.
5. Upload the code on Arduino UNO.
6. Now check the text on LCD and make required corrections to fit text on LCD.

**Code**

*#include <Wire.h>*

*#include <LiquidCrystal\_I2C.h>*

*LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //*

*#define Flame\_Pin A1*

*void setup()*

*{*

*//Initialisation*

*lcd.begin(16, 2);*

*lcd.clear();*

*lcd.backlight();//Power on the back light*

*pinMode(Flame\_Pin, INPUT\_PULLUP);*

*}*

*void loop()*

*{*

*int Flame\_Val;*

*Flame\_Val=digitalRead(Flame\_Pin);*

*if (Flame\_Val == 0)*

*{*

*lcd.clear();//Clean the screen*

*lcd.setCursor(0, 0);*

*lcd.print("Fire Detected ");*

*delay(1000);*

*}*

*delay(1000);*

*}*