SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

BCA

FIRE ALARM

YOGARAJ

(15BCA0091)

VENKATESH

(15BCA0092)

KOUSALYA

(15BCA0093)

SUNIL KUMAR

(15BCA0102)

PREETHA

(15BCA0104)

Project Report

Of

ITA3009-Internet of Things

Fall 2017-18

Submitted to

Faculty: Prof.B.R.Kavitha **Signature:**

Date:

ABSTRACT

We have found in a survey that 80% losses caused due to fire would have been avoided if the fire was detected immediately. Arduino fire detector from Microtronics Technologies is the solution to this problem.

In this project, we have built fire alarm using Arduino Uno which is interfaced with a Bluetooth control, a smoke sensor and buzzer. The temperature sensor senses the heat and smoke sensor senses any smoke generated due to burning or fire. Buzzer connected to Arduino gives us an alarm indication. Whenever fire triggered, it burns objects nearby and produces smoke. A fire alarm can also be triggered due to small smoke from candlelight or oil lamps used in a household. Also, whenever heat intensity is high then also the alarm goes on. Buzzer or alarm is turned off whenever the temperature goes to normal room temperature and smoke level reduces.

LITERATURE REVIEW

Arduino fire alarm system is an important system for industrial purpose as well as for household purpose. Whenever it detects fire or smoke then it instantly alerts the user about the fire through the Bluetooth module. For this purpose, we are using Arduino Uno which is from Arduino family. Also, the Arduino interfacing with LCD display is done to display the status of the system whether the Smoke and Overheat is detected or not. And Arduino interfacing with Bluetooth module is done so that user gets an alert message. It intimates the user about the fire detection. This system is really useful whenever the user is not in house or industry or inside the premises. Whenever a fire occurs, the system automatically senses and alerts the user by sending an alert to an app installed on user’s Android mobile.

INTRODUCTION

We have seen many Arduino projects ideas which proves to be a life saver for us. Arduino fire detector is one of them. Fire detector based on Arduino board and smoke & temperature sensor is an Arduino Security alarm system with minimum configuration.

We need to protect our property and assets within the property. Today’s large modern structures are in need of a new solution to fight against fire. For example art gallery, museum, hotel, board rooms, offices, commercial ware houses. In these places, Arduino fire alarm system provides protection from fire.

Advantages:

1) It detects fire at an early stage. This, in turn, helps in early reaction, saving lives and property
2) It protects us and our residence.

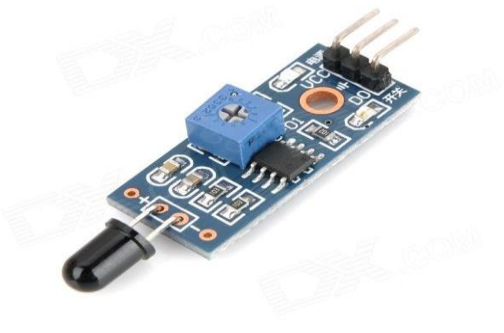
HARDWARE AND SOFTWARE REQUIREMENTS

ARUDINO UNO:

The **Arduino Uno** is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

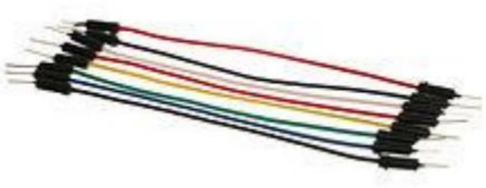
FLAME DETECTOR:

A **flame detector** is a sensor designed to detect and respond to the presence of a flame or fire, allowing **flame detection**. 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. When used in applications such as industrial furnaces, their role is to provide confirmation that the furnace is properly lit; in these cases they take no direct action beyond notifying the operator or control system. A flame detector can often respond faster and more accurately than a smoke or heat detector due to the mechanisms it uses to detect the flame.[1][2]



JUMP WIRES

Jump wires (also called jumper wires) for solderless bread boarding can be obtained in ready-to-use jump wire sets or can be manually manufactured. The latter can become tedious work for larger circuits. Ready to-use jump wires come in different qualities, some even with tiny plugs attached to the wire ends. Jump wire material for ready-made or homemade wires should usually be 22 AWG (0.33 mm2) solid copper, tin-plated wire - assuming no tiny plugs are to be attached to the wire ends. The wire ends should be stripped 3 ⁄16 to 5 ⁄16 in (4.8 to 7.9 mm). Shorter stripped wires might result in bad contact with the board's spring clips (insulation being caught in the springs). Longer stripped wires increase the likelihood of short-circuits on the board. Needle-nose pliers and tweezers are helpful when inserting or removing wires, particularly on crowded boards.



Differently colored wires and color-coding discipline are often adhered to for consistency. However, the number of available colors is typically far fewer than the number of signal types or paths. Typically, a few wire colors are reserved for the supply voltages and ground (e.g., red, blue, black), some are reserved for main signals, and the rest are simply used where convenient.

ARDUINO SOFTWARE:

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions:

- *setup ()*: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
- *loop ()*: After *setup()* has been called, function *loop()* is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Most Arduino boards contain a light-emitting diode (LED) and a load resistor connected between pin 13 and ground, which is a convenient feature for many tests and program function. A typical program for a beginning Arduino programmer blinks a LED repeatedly.

BLYNK APP:

Use Blynk hardware libraries to get your product online and connected to the cloud. We support 400+ different hardware models. And you can use WiFi, Ethernet, 2G, 3G, LoRa or BLE.

Drag-n-drop widgets you need to visualize sensor data, control GPIOs, send Push Notifications and much much more.

Free for developers, No coding, NO SDKs, NO X-Code, No Android Studio. It just works out of the box.

When app is tested and ready - add your icon, logo, company name, adjust the design to match your brand, and publish the app to App Store and Google Play in no time.

DESIGN METHODOLOGY

The Arduino boar senses the signal to all parts .At first program in which sensor value is less than 100 the buzzer will give a sound .After the flame went far away the buffer will automatically become off in which flame detector takes place.

The whole system contains FLAME SENSOR, ARDUINO UNO, BLUETOOTH, POWER SUPPLY (BATTERY), CONNECTING CABLES and BREAD BOARD.

Code:

```
#include<SoftwareSerial.h>
int sensorPin=A0;
int sensorValue=0;
int led=9;
int buzzer=7;
void setup() {
  // put your setup code here, to run once:
pinMode(buzzer,OUTPUT);
pinMode(led,OUTPUT);
Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
Serial.println("welcome");
sensorValue=analogRead(sensorPin);
Serial.println(sensorValue);
if(sensorValue<100)
{
  Serial.println("fire detected");
  Serial.println("LED ON");
  digitalWrite(led,HIGH);
  digitalWrite(buzzer,HIGH);
  delay(1000);
}
digitalWrite(led,LOW);
digitalWrite(led,LOW);
delay(sensorValue);
}
```