CSE 3009: Internet of Things (IoT)

Project Review 2&3

On

Short Range Fire Smoke Detection System

By

Khushee Jain (19BCE0914)

Achint Arya Mishra (19CE2578)

Panshul Jindal (19BCE2227)

Jatin (19BCE2256)

Slot: A2

Under the Guidance of

Prof. Sudha M

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering Vellore Institute of Technology (VIT), Vellore Tamil Nadu – 632 014

August 2020

Abstract:

Our aim is to build a short range fire smoke detector to reduce risk of fire disaster in private and public places. The device will change the colour of LEDs and emit sound as soon as it detects any kind of fire smoke nearby.

The device information is stored in a real time database which is displayed through our android application Smokie which displays if the smoke is detected and has a feature of making emergency calls and contacting Ambulance as well as Fire Brigade.

Introduction:

This project describes hardware synthesis for a fire smoke detector with an Node MCU Arduino microcontroller. The system consists of MQ-2 smoke sensor which will sense if there is smoke nearby. If the smoke is detected, then the alarm will start ringing and the LED will change it's colour from white to blue. The project specification is presented in accordance with the expected functionality of the detector. Based on this specification hardware, the proper elements are selected and described. The output of MQ-2 sensor is uploaded in the real time database and is fetched by the application.

Materials Required:

- Node MCU ESP8266 Wi-Fi Module
- MQ2 Gas sensor
- LED Lights
- Peizzo Buzzer
- 220-ohm resistor
- Jumping wires
- Breadboard

Finally, we propose a hardware scheme for our fire smoke detector project and an android application Smokie.

Literature Review:

The domain of our project in Internet of Things (IoT) is Smart Home. Smoke detectors are installed in homes to detect smoke that is typically an early sign of fire. The MQ2 gas sensor detects methane gas and carbon monoxide (CO) present in the atmosphere. The following are the three major methods of smoke detection we came across after referring to different papers:

- Smoke detection using Arduino microcontroller
- Vision based smoke detection system
- Video based smoke detection system.

(Video based smoke detection system):

Video based fire smoke detection using cameras is found to be very effective in detecting smoke or fire. These systems can be used to provide early fire smoke detection if a reliable fire detection software is installed in the system of CCTV. It was found in our referred papers that smoke is always visible before fire in most outdoor scenarios. This motivates us to research on detecting smoke in the absence or presence of flame from a single frame of video.

But this method is not suitable for our project as we are restricted to implant CCTV cameras for its implementation and our project focuses on smart home. Installing cameras at personal places like home, PGs, hostel rooms etc. would lead to privacy issues. Moreover, this method includes machine learning and deep learning algorithms which increases the complexity of the project.

(Vision based smoke detection system):

Due to the rapid developments in digital camera technology and developments in content-based video processing, the use of vision-based smoke detection system is introduced. Vision based systems make use of three characteristic features of fire: colour, motion and geometry. The colour information is used as a pre-processing step in the detection of possible smoke. In this method, novel models for smoke detection using vision processing is provided. The models use different colour models for smoke. The colour models are extracted using a statistical analysis of samples extracted from different type of video sequences and images. The extracted model can be used in complete smoke detection system which combines colour information with motion analysis.

This method is not suitable for our project because they use a mixture of ten three dimensional gaussians in RGB colour space to model a smoke detection which combines generic colour model based on RGB colour space and motion information. This makes the process complex for us as the Arduino microcontroller method needs mostly software content and this requires knowledge in big data management.

(Smoke detection using Arduino microcontroller)

In this paper, we want to depict equipment union for a versatile gas finder to locate the risky zone where numerous fixed identifiers are eccentric or as well costly. In this manner, we make equipment to tackle this problem a self-propelled robotic gas detector to identify hazardous gas. In the undertaking, we put the best accentuation on gathering, handling, and sending information. For this reason, the Arduino microcontroller is being utilized by our team for the above stated purposes.

Arduino is an open-source prototyping stage intended to be anything but difficult to use for amateurs who have no programming or electronic elevated level understanding. We in this project will be facing this problem to work upon as we will have to first understand how Arduino works and write a code accordingly for our project.

It very well may be used to create intelligent items that can react to signals from the general condition. This microcontroller can control an assortment of things, for example, catches, engines, GPS units, LEDs, sensors, cameras and more. We have decided to use: -The microcontroller Arduino ESP8266 in our project in the hardware part because of its extended functionalities.

Firebase:

Firebase is a toolset provided by Google to build, improve, and grow an application. It includes services like analysis, authentication, databases, file storage, etc. which application developers won't really want to build themselves. The Firebase Realtime Database is a cloud-hosted database. The services are hosted in the cloud with very little efforts of developer. Data is stored as JSON and synchronized in realtime to every connected client. When a cross-platform application is build all the clients share one Realtime Database instance and automatically receive updates with the newest data. Firebase uses data synchronization every time to update data in place of typical http requests. It provides a collaborative platform without thinking over networking codes. Firebase applications remain responsive even when offline as it persists the data to disk and synchronizes it with the current server state as soon as the connection is re-established.

Problem Statement:

Sometimes just a flame of candle or unexhausted matchstick leads to a major fire incident. Fire disasters typically results in loss of life and property. An ordinary smoke detector just detects and rings the alarm which can only alert people present at that place. So if a fire incident takes place when no one is around it could lead to a major disaster. Also commercial fire detectors are costly thus not every family can afford it.

There is a need of a cheaper fire smoke detector which can even alert people nearby as well as who are not around to avoid any accident.

Proposed Model:

Pseudo Code:

Defining the firebase host
Defining the firebase auth
Defining the wifi name
Defining the wifi password
Declaring the led white port = D0
Declaring the led blue port = D1
Declaring the buzzer port = D2
Declaring the sensor port = A0
Declaring the data input from the sensor = 0 (initially)
Declaring the firebasedata
Void setup
{
    Serial begin of 115200 ms
    Initiating pinmode for led white as output
    Initiating pinmode for led blue as output
    Initiating pinmode for buzzer as output
    Beginning the wifi using wifi name and wifi password
    Printing connecting to wifi
    While loop if wifi status is not equal ‘WL_CONNECTED’
    {
        Print ‘connection failed’
        Delay of 300 ms
    }

```
        Print line
        Print ‘Connected with IP:’
        Print the local IP Address of the user
        Beginning the firebase using firebase host and firebase auth
        Reconnecting the firebase with WiFi
}

Void loop{
        Analog read the data from mq2 sensor and assigning it to integer data
        Printing the integer data
        Digital write the led white as HIGH (Turn on the white led)
        While(data>800) {
                Analog read the data from mq2 sensor and assigning it to integer
                data
                Digital write the led white as LOW (Turn off the white led)

                Digital write the led blue as HIGH (Turn on the blue led)
                Digital write the buzzer as HIGH (Turn on the buzzer)
                Sending the firebase, the value of data with the node ‘Value’
                Delay of 5000ms
        }
        Digital write the led white as HIGH (Turn on the white led)
        Digital write the led blue as LOW (Turn off the blue led)
        Digital write the buzzer as LOW (Turn off the buzzer)
        if(Sending the firebase, the value of data with the node ‘Value’){
                Print ‘Upload Successful’
        }
        else {
                print ‘Upload Failed’
        }
}
```

Arduino Code:

```
#include "FirebaseESP8266.h"
#include <ESP8266WiFi.h>

#define FIREBASE_HOST "http://smoke-detector-8175f.firebaseio.com"
#define FIREBASE_AUTH
"sSLJ9JK6T4hnTwA307yRKN8ICMzk2dKKJEVPRfls"
#define WIFI_SSID "PANSHUL"
#define WIFI_PASSWORD "9818493390"

int led_white = D0; // led white light is connected with the digital pin D0
int led_blue = D1; // led blue light is connected with the digital pin D1
int buzzer = D2; // buzzer is connected to the digital pin D2
int mq2 = A0; // smoke sensor is connected with the analog pin A0
int data = 0; // Data input taken from the sensor

FirebaseData firebaseData;

void setup()
{
 Serial.begin(115200);
 pinMode(led_white, OUTPUT); // Initiating pin for white led
 pinMode(led_blue, OUTPUT); // Initiating pin for blue led
 pinMode(buzzer, OUTPUT); // Initiating pin for buzzer
 WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // Beginning the wifi
 Serial.print("Connecting to wifi");
 while (WiFi.status() != WL_CONNECTED)
 {
  Serial.print("Connection Failed");
  delay(300);
 }
 Serial.println();
 Serial.print("Connected with IP: ");
 Serial.println(WiFi.localIP());
 Serial.println();
 Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // Beginning
firebase
 Firebase.reconnectWiFi(true); // Connecting firebase with WiFi
}
```

```
void loop()
{
  data=analogRead(mq2); // Data read from the pin
  Serial.println(data);
  digitalWrite(led_white,HIGH); // led white on
  while(data>800){
   data=analogRead(mq2); // Data read from the pin
   digitalWrite(led_white, LOW); // led white off
   digitalWrite(led_blue, HIGH); // led blue on
   digitalWrite(buzzer,HIGH); //buzzer on
   Firebase.setInt(firebaseData, "/Value", data); // Setting the value of data in
the firebase database
   delay(5000);
  }


   digitalWrite(led_white, HIGH); // led white on
   digitalWrite(buzzer,LOW); // led buzzer off
   digitalWrite(led_blue,LOW); // led blue off
  if(Firebase.setInt(firebaseData, "/Value", data)) // Setting the value of data
in the firebase database
 {
    Serial.println("Upload successful");
 }
 else{
  Serial.println("Upload failed");
 }
}
```

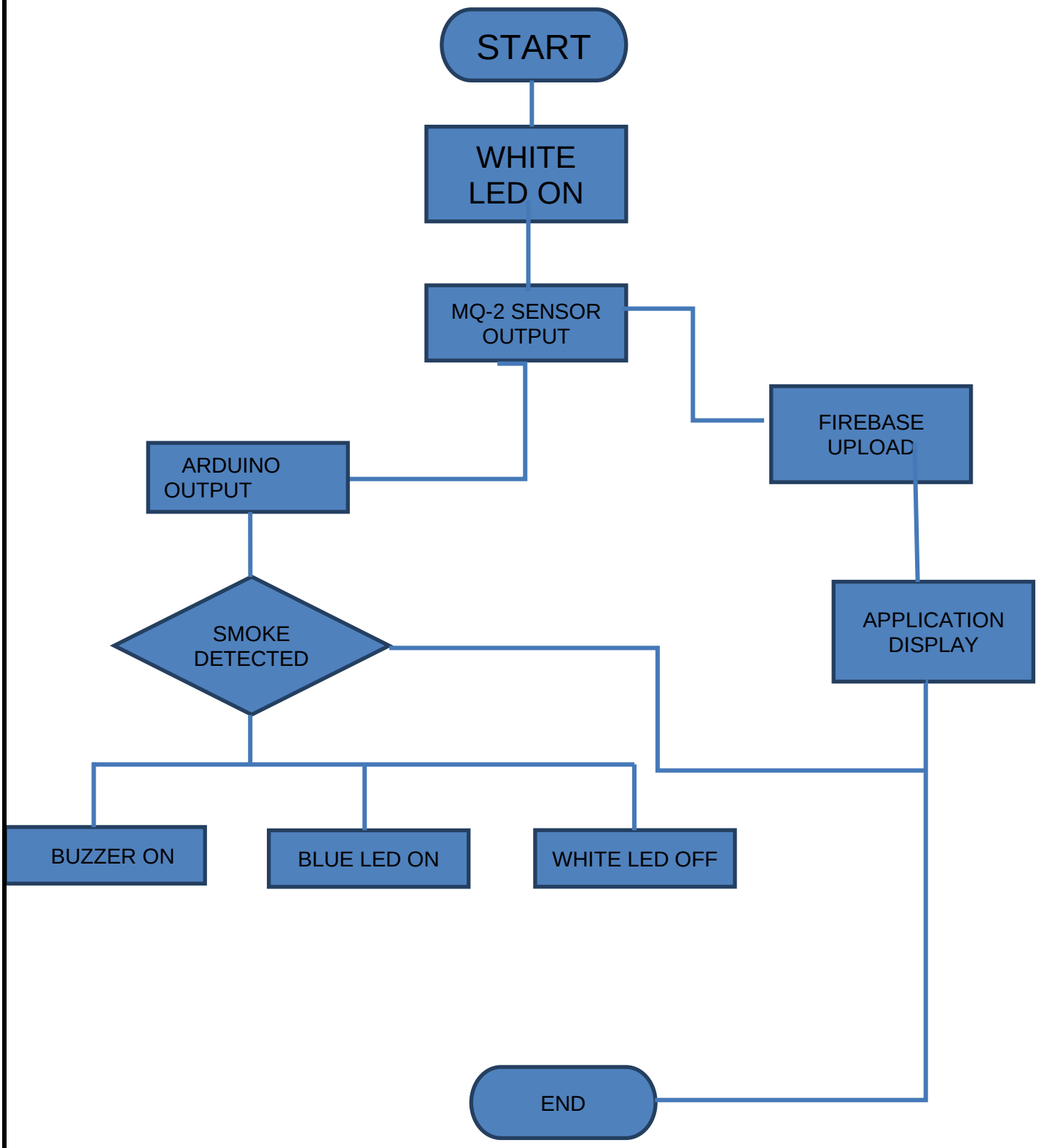
Android Studio Code:

```
package com.example.smokedetection;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.INotificationSideChannel;
import android.util.Log;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
public class Landing_page extends AppCompatActivity {
    TextView value,smokeDetected,ambulance,fire,call;
    Button detectSmoke;
    ImageView calling;
    @Override



protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);


        setContentView(R.layout.activity_landing_page);
        findViewById();
        clicklistener();
    }
    public void findViewById(){
        value = findViewById(R.id.valueTextView);
        smokeDetected=findViewById(R.id.detectedTextView);
        detectSmoke=findViewById(R.id.detectSmoke);
        fire=findViewById(R.id.fire);
        ambulance=findViewById(R.id.ambulance);
        call=findViewById(R.id.emergency);
        calling=findViewById(R.id.calling);
    }
```

```
public void firebase(){
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myref = database.getReference();
        myref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                Log.i("Value","hello");
                String ph = dataSnapshot.child("Value").getValue().toString();
                Integer smokeValue= Integer.parseInt(ph);
                value.setText(smokeValue.toString());
                if(smokeValue>800){
                    //alertdialogue();
                    smokedetect();
                }
                else {
                    smokenotdetect();
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Log.i("Failed to read value",error.toException().toString());
            }
        });
    }


public void smokedetect(){
        detectSmoke.setVisibility(View.INVISIBLE);
        smokeDetected.setVisibility(View.VISIBLE);
        call.setVisibility(View.VISIBLE);
        fire.setVisibility(View.VISIBLE);
        ambulance.setVisibility(View.VISIBLE);
        calling.setVisibility(View.VISIBLE);
        value.setTextColor(Color.RED);
        value.setTypeface(null, Typeface.BOLD);
    }


public void smokenotdetect(){
        value.setTextColor(Color.WHITE);
        value.setTypeface(null, Typeface.NORMAL);
        detectSmoke.setVisibility(View.VISIBLE);
        smokeDetected.setVisibility(View.INVISIBLE);
        call.setVisibility(View.INVISIBLE);
        fire.setVisibility(View.INVISIBLE);
        ambulance.setVisibility(View.INVISIBLE);
        calling.setVisibility(View.INVISIBLE);
    }
    public void alertdialogue(){
        AlertDialog.Builder builder = new AlertDialog.Builder(Landing_page.this);
        builder.setCancelable(true);
        builder.setTitle("Smoke Detected");
        builder.setMessage("");
        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{


            @Override
```

```
        public void onClick(DialogInterface dialogInterface, int i) {
                dialogInterface.cancel();
            }
        });
        builder.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                smokenotdetect();
            }
        });
        builder.show();
    }
    public void clicklistener(){ detectSmoke.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {
                firebase();
                Animation animAlpha =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.anim_alpha);
                view.startAnimation(animAlpha);
            }
        });
        if(ActivityCompat.checkSelfPermission(Landing_page.this,
Manifest.permission.CALL_PHONE)== PackageManager.PERMISSION_GRANTED) {
            call.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    String s = "tel:" + "112";
                    Intent intent = new Intent(Intent.ACTION_CALL);
                    intent.setData(Uri.parse(s));
                    startActivity(intent);
                }
            });


             ambulance.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            String s = "tel:" + "102";
                            Intent intent = new Intent(Intent.ACTION_CALL);
                            intent.setData(Uri.parse(s));
                            startActivity(intent);
                        }
                    });
                    fire.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            String s = "tel:" + "101";
                            Intent intent = new Intent(Intent.ACTION_CALL);
                            intent.setData(Uri.parse(s));
                            startActivity(intent);
                        }
                    });
              }
              else{
                  ActivityCompat.requestPermissions(Landing_page.this,new
        String[]{Manifest.permission.CALL_PHONE},44);

            }} }
```

Process Flow Diagram:



Implementation Steps:

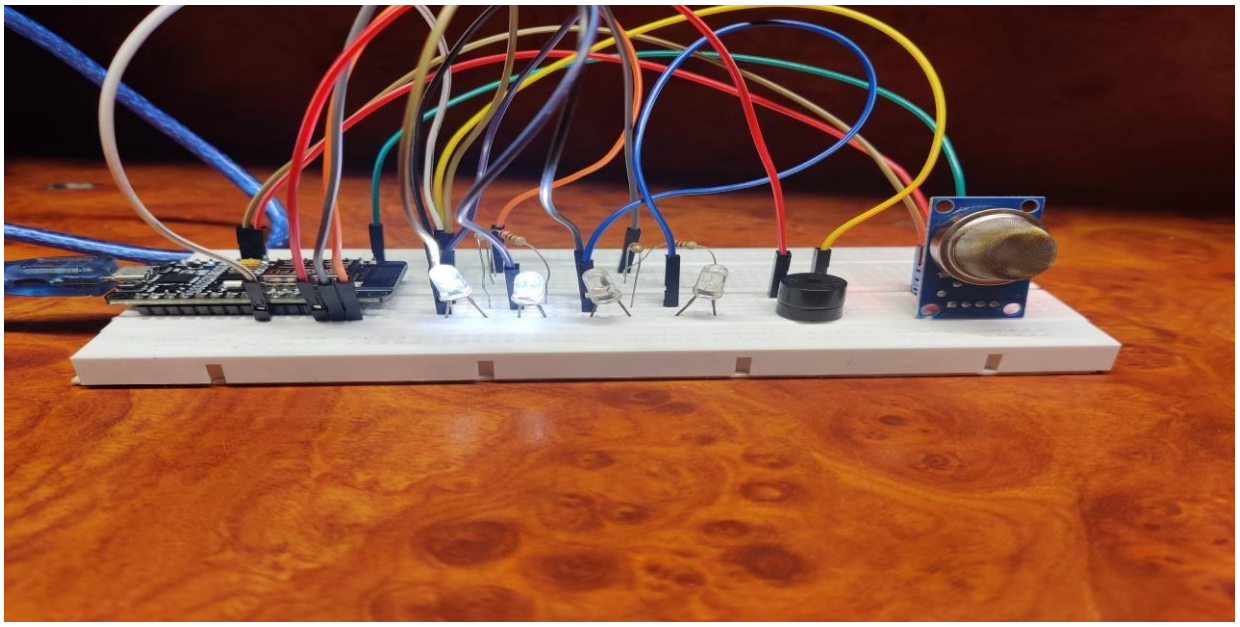
1. Download Arduino IDE from https://www.arduino.cc/en/Main/software.
2. Connect the board to the computer via a micro USB cable.
3. Then install any drivers required for the Arduino IDE.
4. Then launch the Arduino IDE.
5. Download the board type (Node MCU ESP8266) from board manager in the IDE.
6. Select the board to Node MCU ESP 8266 from the tools and boards section.
7. Select the port to COM3.
8. Install the libraries required.
9. Write the required sketch code.
10. Create the firebase project name ‘Smoke Detection’.
11. Get hardware required for the project.
12. Compile the hardware according to the diagram.
13. Upload the code to the Arduino.
14. Make the android application using Android Studio.
15. Connect the firebase Realtime database with the Arduino and android application.

Platforms Used:

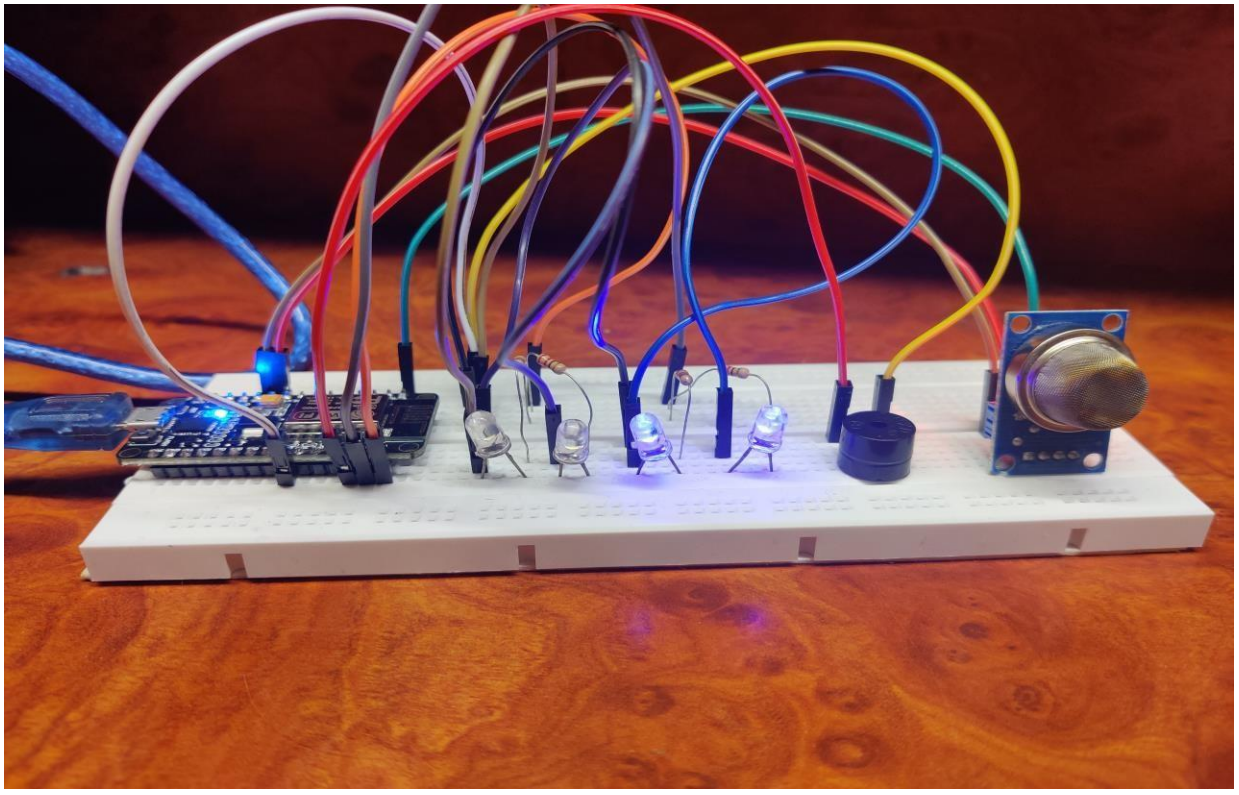
- Firebase
- Android Studio
- Arduino IDE

Experimental Setup & Results:

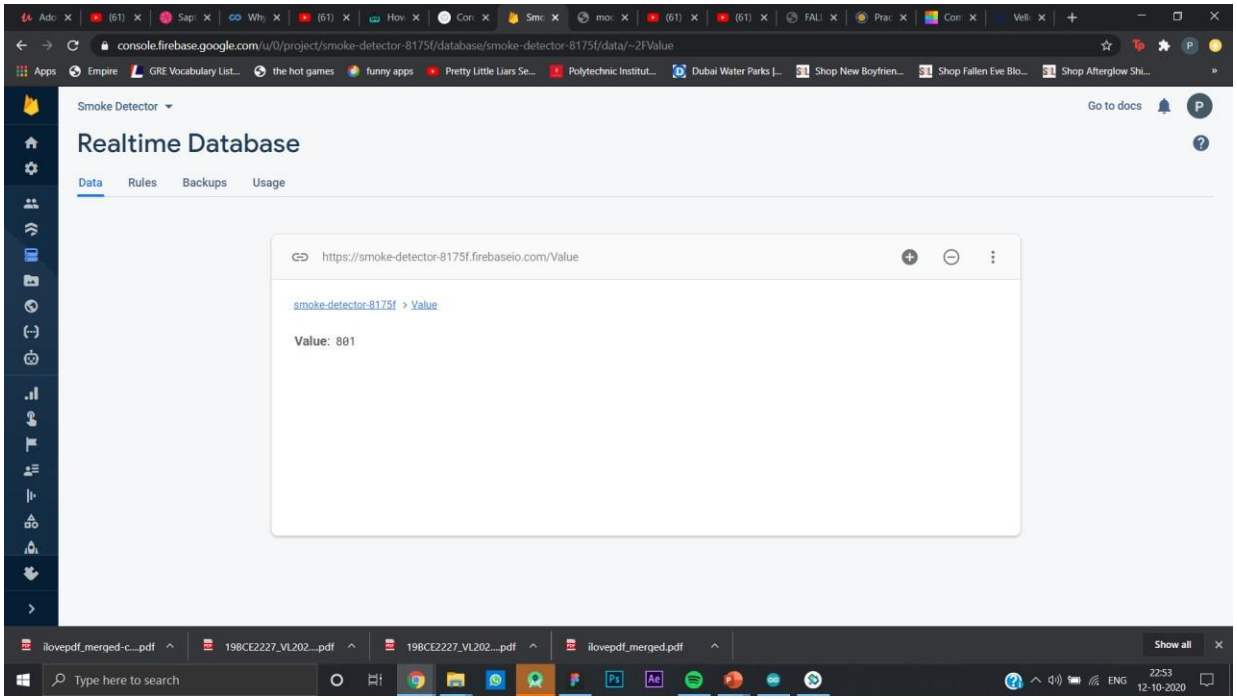
Before detecting Smoke:



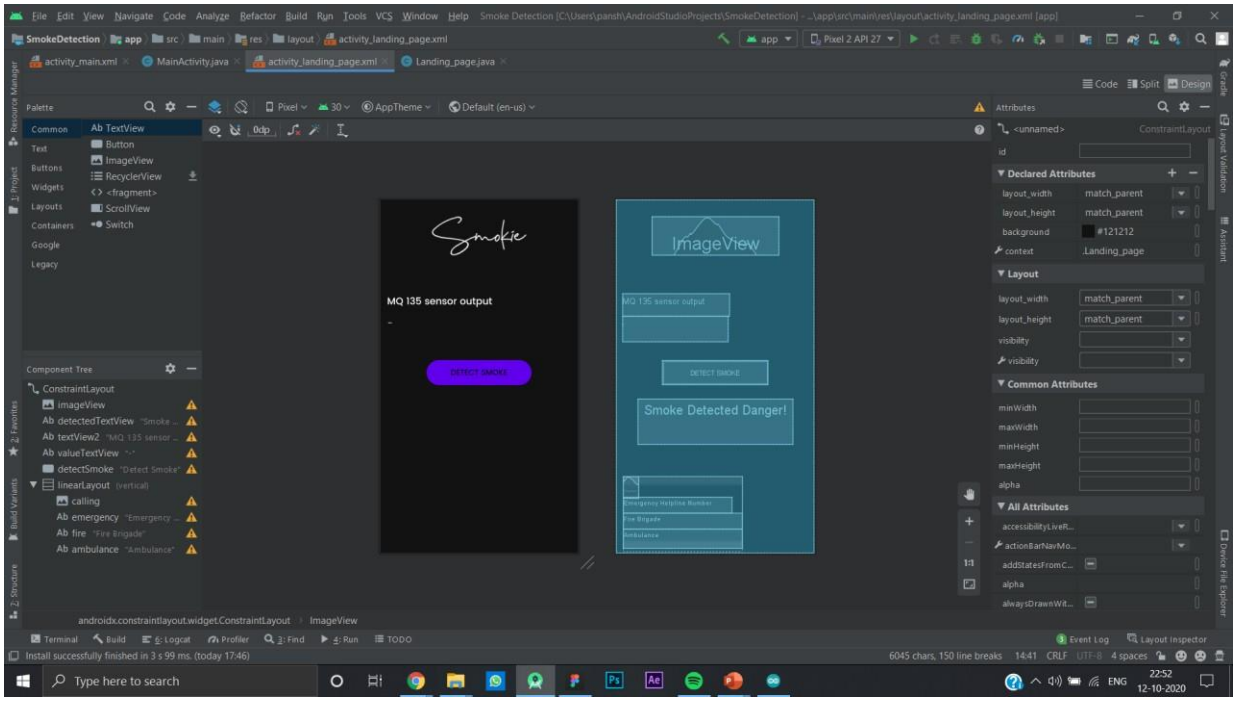
After Smoke detection:



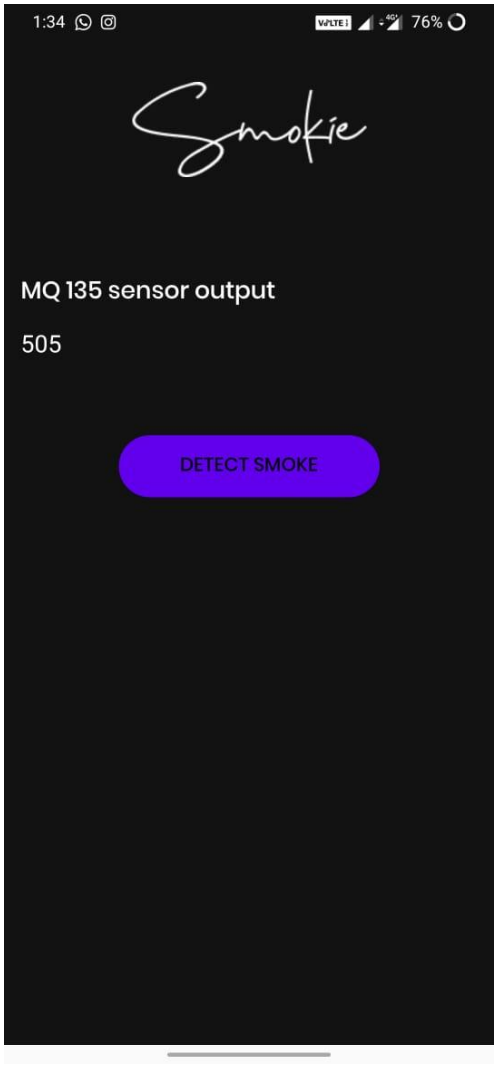
Firebase Results:



Application Design:



Application Output:



Conclusion:

The MQ-2 sensor detects the smoke in ppm. The uploaded Arduino code executes and if smoke is detected the alarm rings and LEDs changes the colour from white to blue alerting the people. The output of MQ-2 sensor is sent to real time database using Firebase and is fetched by the android application. The app displays if the smoke is detected and gives the option to call the emergency services.

References:

[1] WIRELESS SMOKE DETECTION SYSTEM

James Acevedo, 136 N. Ocean Ave,

Freeport, NY (US) 11520

July 16, 2002

[2] Vision based smoke detection system using image energy and color information

Simone Calderara · Paolo Piccinini · Rita Cucchiara DII,

University of Modena and Reggio Emilia, Modena, Italy

21 May 2010

[3] SMOKE DETECTION SYSTEM

Christopher Girling, Bounds Cross, Shaw Green, Rushden Hertfordshire SG9 OTB; Mark Symonds

4 Barfolds, North Mymms, Hatfield, Hertfordshire AL97NG; Peter David Fox, The Lodge, Stagenhoe Park, St. Pauls Walden, NR, Whitwell, Hertfordshire, SG48DD, all of United Kingdom

Jan. 30, 1996

[4] SMOKE DETECTION METHODS, DEVICES, AND SYSTEMS

Thomas David Petite,

Douglasville, GA (US)

SIPCO, LLC, Atlanta, CA (US)

Nov. 13, 2007

[5] TOBACCOSMOKE DETECTION SYSTEM TAMPER DETECTION

Dan Deacy,

7149 Canoga Rd., Auburn, NY (US) 13021

Sep. 6, 2005

[6] SMOKE DETECTION system

Francis C. Evans, Dongan Hills, Staten Island, and Khatchik O. Donelian,

New York, N.Y., as signors to The American District Telegraph Company, New York, N. Y corporation of New Jersey

December 10, 1938,

[7] A Mobile Gas Detector with an Arduino Microcontroller

Arkadiusz ŚPIEWAK, Wojciech SAŁABUN

Department of Artificial Intelligence Methods and Applied Mathematics, Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin, ul. Żołnierska 49, 71-210 Szczecin, Poland

01 September 2015.

[8] Cigarette Smoke Detectors for Non-Smoking Areas in the Building

Sancha Panpaeng , Phattharamon Phanpeang Ekkasit Metharak

Information and Communication Technology Department of Science, Maehongson College, Chiangmai Rajabhat University Maehongson, 58000, Thailand [2018]

[9] SLEEPER'S SMOKE-ALARM CLOCK

Clifford G. Flittie,

33 Stonehouse Rd., Scarsdale, N.Y. 10583

Jan. 29, 1980

[10] POWER STRIP WITH SMOKE DETECTION AUTO-SHUTOFF

Michael Dayoub

2625 Bethany Creek U.S. PATENT DOCUMENTS Ct., Alpharetta, GA (US) 30004

Dec. 26, 2006

[11] SMOKE ALARM

Andre M Landais

567 E. 88th St., Brooklyn, NY (US) 11236

May 7, 2002

[12] Air Contaminants Monitoring of Carbon Monoxide and Hydrogen using Standalone Microcontroller Based System for Passive Smoker

M. K. Zaman M. Zaiedi

Malaysian Institute of Information Technology, Universiti Kuala Lumpur 1016 Jalan Sultan Ismail, 50250 Kuala Lumpur, Malaysia

August 2014

[13] WIRELESS ALARM SYSTEM

Jack Reinowitz

563 Union Rd., Spring Valley, NY. 10977

Dec. 7, 1982

[14] SMOKE ALARM WITH HIGH AND LOW PITCHED TONES

Oneida V. Hess; James L. Hess.

Both of 5084 Chase Rd., Dearborn. Mich.

Mar. 10, 1998

[15]A Mobile Gas Detector with an Arduino Microcontroller

Arkadiusz ŚPIEWAK, Wojciech SAŁABUN Department of Artificial Intelligence Methods and Applied Mathematics, Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin, ul. Żołnierska 49, 71-210 Szczecin, Poland

August 2015

[16] Home Automation using Arduino and IOT

Priyanka G.Gawade Prajakta D.Jadhav Madhavi B. Danane

Department of Electronics and Telecommunication Engineering. Sou. Sushila Danchand Ghodawat Charitable Trust's and Group Of Institutions Kolhapur, Maharashtra, India

March– 2018

[17] Implementation of IoT eith ESP8266 Part II – Home Automation

K. Priyanka Srinivasulu Reddy

Research and Development Engineer, Videocon Industries Ltd., Aurangabad, Maharashtra, India

20 August 2019

[18] SMOKE DETECTOR

Hiroyuki Yokota, Tokyo (JP)

Nohmi Bosai Ltd., Tokyo (JP)

Jan. 31, 2012

[19] Smoke Detection System By Using PIC Microcontroller

Aung Naing Oo , Mar Mar Cho

Lecturer, Dr, Department of Physics, Yadanabon University

2019

[20] SMOKE DETECTOR

Stephen John Penney, Chelmsford (GB)

Thorn Security Limited, Sunbury-on-Thames (GB)

May 10, 2011

[21] EXTENDED SMOKE ALARMSYSTEM

Bradly Jay Billman, San Antonio, TX(US)

United Services Automobile Association (USAA), San Antonio, TX (US)

May 11, 2010

[22] PORTABLE COMBINATION LAMP, SMOKE DETECTOR AND POWER FAILURE ALARM

Anthony M. Jarosz, Dracut, Mass.; Randy R. Stevens, Delray Beach, Fla.

T. J. Company, Ayer, Mass.

December 6, 1983

[23] SCATTERING LIGHTSMOKE ALARM

Joachim Schneider, Nattheim (DE); Anton Pfefferseder, Sauerlach-Arget (DE) Andreas Hensel, Vaihingen (DE); Ulrich Oppelt, Zorneding (DE)

Robert Bosch GmbH, Stuttgart (DE)

February 4, 2003

[24] PORTABLE SMOKE ALARM

Joseph P. Bellino, Shelton, Conn

General Electric Company, New York, N.Y.

June 25, 1985

[25] EXTRA SECURITY SMOKE ALARM SYSTEM

William Robert Banga, 1910 Needle Palm Dr., Edgewater, Fla. 32141

March 30, 1999

[26] PHOTOELECTRIC SMOKE DETECTION SYSTEM

Francis C. Evans Dongan Hills, Khatchik O. Donelian

New York, N. Y., assignors to American District Telegraph Company, Jersey City, N.J., a corporation of New Jersey

March 15, 1939

[27] BATTERY POWERED SMOKE ALARM SAFETY LOCKOUT SYSTEM

Gary Belano, Lake Zurich, Ill

Seatt Cor oration Downers Grove, III

September 26, 1989

[28] PORTABLE LAMP, SMOKE DETECTOR AND POWER FAILURE ALARM

Anthony M. Jarosz, Dracut, Mass.; Randy R. Stevens, Delray Beach, Fla.

T. J. Company, Ayer, Mass.

December 6, 1983