

## SMART FIRE SYSTEM USING IOT

Ali Hasan Ali

Education Directorate of Thi-Qar, Ministry of Education, Iraq

### ARTICLE INFO

#### Article history:

Received 16 Jan 2023

Revised form 15 Feb 2023

Accepted 30 Mar 2023

### Аннотация

The project consists of transceivers. Transmitters contain sensors. These sensors detect air molecules, such as gas in the air, as well as fire and smoke. These devices transmit the readings via the Arduino to the receiver wirelessly via Wi-Fi. In order for the recipient to receive these readings and display them in the form of an alert by means of an application on the mobile phone in addition to the bell,

If the readings are high, it means that there is gas or fire in the place of the transmitters, thus the red LED will be triggered in addition to the alarm buzzer. If the readings are not high, the green will be on.

These devices are placed in important places such as warehouses, hospitals and banks because of their importance and the presence of these devices is very important, especially as they work on wireless communication between the transmitter and the receiver, as their range reaches 1000 meters in open places and decreases in closed places such as the presence of walls and others.

**Ключевые слова:** Integrated Development Environment, Universal Serial Bus, Lesser General Public License, Input and Output, Automatic voltage Regulator, reduced instruction set computer, surface-mount device, Pulse Width Modulation, General Purpose Input Output.

© 2023 Hosting by Central Asian Studies. All rights reserved.

\*\*\*

## 1. INTRODUCTION

### 1.1. Introduction

Detecting carbon monoxide, fire, smoke, or other fire-associated emergencies with a fire alarm system is a vital safety feature. Smoke detectors and heat detectors can initiate these alarms automatically, or manual fire alarm activation devices, such as pull stations and call points, can trigger them manually. There are various types of alarms, including motorized bells, sounders, and horns that can be mounted on walls. Also, these might be speaker strobes that emit an alert accompanied by a voice evacuation message telling those within the building to stay away from the lifts. Depending on the country and device manufacturer, the fire alarm may be tuned to several frequencies and to a variety of tones, including low, medium, and high. A siren with alternating frequencies is the most common type of fire alarm system in Europe. In the United States and Canada, fire alarm electrical devices are referred to as "horns," and they may either be set to various codes or continuous. There are also different volume levels available for fire alarm warning devices [1].



**Figure 1.1: United States and Canada fire alarm notifications that are widely utilized**

Devices that are manually actuated are also termed as break glass stations, fire alarm boxes, manual or simple pull stations, and (in Europe) call points. Devices for manually activating fire alarms are placed at or close to exits, and are operated and identified. Pulling a lever or breaking glass is usually needed to activate them.

There are many types of automatically actuated devices that respond to different types of detectable changes linked to fires: sprinklers operations for a water-flow detector, combustion products for a smoke detector, thermal energy via convection for the detector of heat, combustion gases for a fire gas detector, and energy via radiation for a flame detector. A new set of inventions can investigate the visible impacts of fire and movement using cameras and computer algorithms in applications where other detection methods are ineffective or hostile [2]

## **1.2. Objectives**

1. Early warning of a fire or gas leak
2. Make wireless monitors easy to place in different places
3. Alert people to the presence of a fire for ease of action before it spreads
4. Make the monitoring devices small, so that they can be placed in several places and do not require much effort
5. Make all sensor sources in one central monitoring direction

## **2. THEORY OF THE PROJECT**

### **2.1. Introduction:**

The project is divided into the software system and the hardware system. These two systems complement each other, so anything we do in programming will show results in the devices.[1]

### **2.2. Software Sections**

The Arduino Integrated Development Environment (IDE) is a cross-platform application (available for Windows, macOS, Linux) written using functions in C and C++. It is used to write and upload programs to Arduino compatible boards, but also development boards from other vendors with the help of third-party kernels.

Source code for the IDE is governed by the GNU General Public License, version 2. C and C++ languages are supported by the Arduino IDE with specific code structure guidelines. A software library from the Wiring project is available through the Arduino IDE and includes numerous widely used input and output

functions. The main program loop and the sketch are the only two functions that user-written code needs to start them. Using the GNU toolchain provided with the IDE release, these functions are compiled and linked with the program stub `main()` in an executable loop execution program. With the help of the `avrdude` software included in the Arduino IDE, executable code is converted into a hex-encoded text file that can be loaded by the board firmware's loader and uploaded to the Arduino board. In most cases, Arduino boards are uploaded using `Avrdude` as an upload tool [3].

Various vendors have begun implementing specialized open-source compilers and tools (kernels) that can create sketches and upload them to other Arduino official microcontroller families supported microcontrollers as a result of the increasing popularity of Arduino as a software platform. The Arduino community started offering early access to a new Arduino Pro IDE with debugging and other advanced features in October 2019. the figure indicates



**Figure 2.1: Arduino IDE Program**

### 2.3. Hardware Sections

The system includes the Arduino source, the Arduino is considered an open source, where all the sensors are connected to the Arduino, and it is also the main source for the project. the Arduino is connected to a computer via USB to download the programming for the project, or we can also operate it via USB or by battery. The goal of using Arduino is that it is cheap and easy to use.[3][4]

### 2.4. ARDUINO

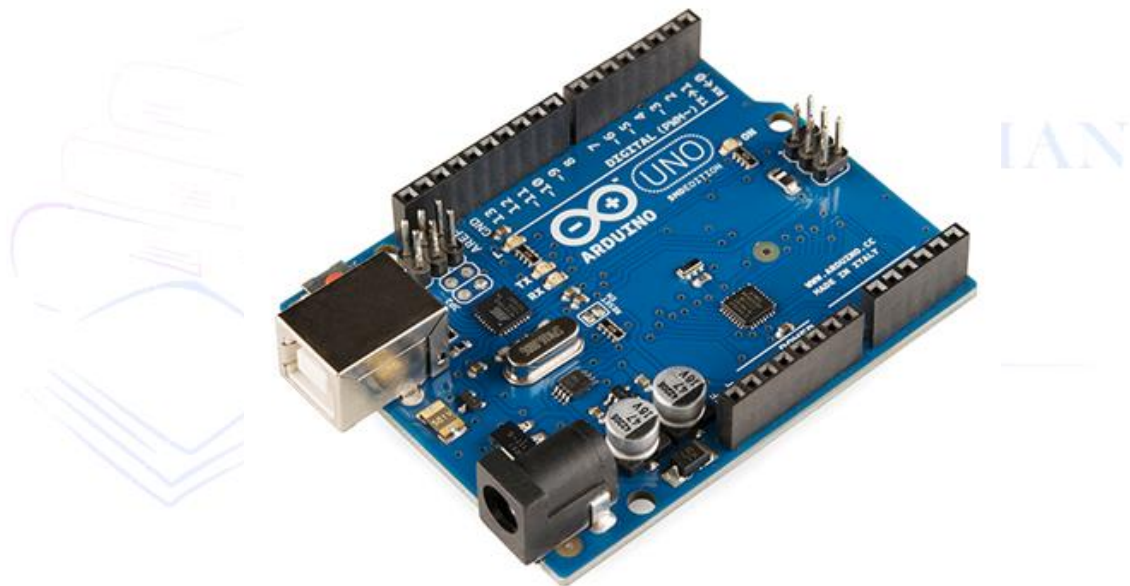
Arduino is a firm that creates and produces single-board microcontrollers and microcontroller kits for creating digital devices. It is an open-source hardware and software initiative. The software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), whilst the company's hardware products are licensed under the CC-BY-SA license [1], enabling the production of

Arduino boards and the dissemination of its software by anybody. In addition to the official website, authorized dealers also sell Arduino boards.

Different microprocessors and controllers are employed in Arduino board designs. These boards have a collection of digital and analog input/output (I/O) pins that can be linked to different expansion boards (also known as "shields"), breadboards (used for prototyping), and other circuits. These boards contain serial communication ports, some of which can load programs and include Universal Serial Bus (USB) on some models. The C and C++ programming languages, sometimes referred to as the "Arduino Language," are used to program microcontrollers. The Arduino projects include a command-line tool (Arduino-cli) and an integrated programming environment (IDE) in addition to the conventional compiler toolchain for Go development.[3][4]

The Arduino project began in 2005 as a teaching tool for interaction design students at the Ivrea School of Interaction Design in Italy. Its goal was to provide a simple and affordable method for both amateurs and experts to build devices that use sensors and actuators to interact with the environment. Simple robots, thermostats, and motion detectors are a few examples of these gadgets that are frequently used by novices.

The tavern in Ivrea, Italy, where some of the project's founders used to hang out is where the name Arduino originated. According to the image, the bar is called after Arduin of Ivrea, Marquis of the March of Ivrea and King of Italy from 1002 to 1014. as the picture shown.



**Figure 2.2: Arduino UNO SMD R3**

## 2.5. Why Arduino Boards

Arduino boards have been used to make different engineering projects and different applications. The Arduino software is simple and easy to use for beginners, yet flexible enough for advanced users. It runs Windows, Linux and Mac. School teachers and students use it to design low-cost scientific instruments to verify the principles of physics and chemistry. There are many other microcontroller platforms available for physical computing. Related features are available from Netmedia's BX-24, Parallax Basic Stamp, MIT's Handyboard, Phidget, and many others.[3]

Arduino also simplifies working with microcontrollers, but it offers teachers, students, and beginners some advantages over other systems.

- Cheap
- Cross-platform
- Simple and clear programming environment



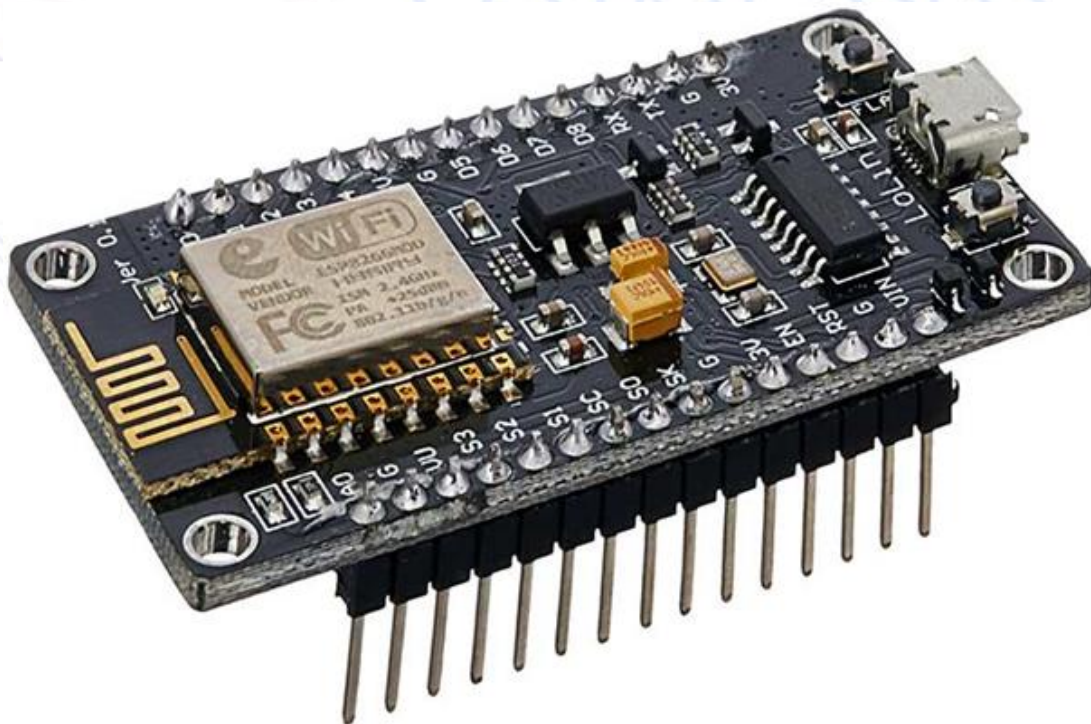
- Open source and extensible software
- Open source and scalable hardware

## 2.6. Type of Arduino and ESP Boards

The list of Arduino boards includes the following such as

- Arduino Uno (R3)
- Arduino Nano
- Arduino Micro
- Arduino Due
- LilyPad Arduino Board
- Arduino Bluetooth
- Arduino Diecimila
- RedBoard Arduino Board
- Arduino Mega (R3) Board
- ESP32
- NodeMCU ESP8266WIFI

To this project we will use the **NodeMCU ESP8266 WIFI**, as figure shown



**Figure 2.3: NodeMCU ESP8266 WIFI**

## 2.7. NodeMCU ESP8266 WIFI

NodeMCU is an open-source firmware that can be designed using an open-source prototyping board. The name "NodeMCU" combines "node" and "MCU" (microcontroller unit). The term "NodeMCU" strictly refers to the firmware and not the associated development kit. [citation needed]

Both firmware and prototyping board designs are open source.

The firmware uses the Lua scripting language. The firmware is based on the eLua project and built on the Espressif Non-OS SDK for ESP8266. It uses many open-source projects such as lua-cjson and SPIFFS. Due to resource constraints, users need to select modules relevant to their projects and build firmware that suits their needs. Also implemented support for 32-bit ESP32.[6][7]

In prototyping hardware, the USB controller is integrated with a surface mount board containing an MCU and antenna on a dual in-line package (DIP) circuit board. It is easy to prototype on a breadboard with DIP format. The design was based primarily on the ESP-12 module of the ESP8266, a Wi-Fi SoC that incorporates the Tensilica Xtensa LX106 core and is extensively utilized in applications of IoT (see associated topics).

## 2.8. Part of NodeMCU ESP8266 WIFI

### A- Power Requirements:

Operating voltage: 2.5 to 3.3 v

Onboard 3.3v 600mA voltage regulation

800mA operating current

20  $\mu$ A during sleep mode

. as figure shown

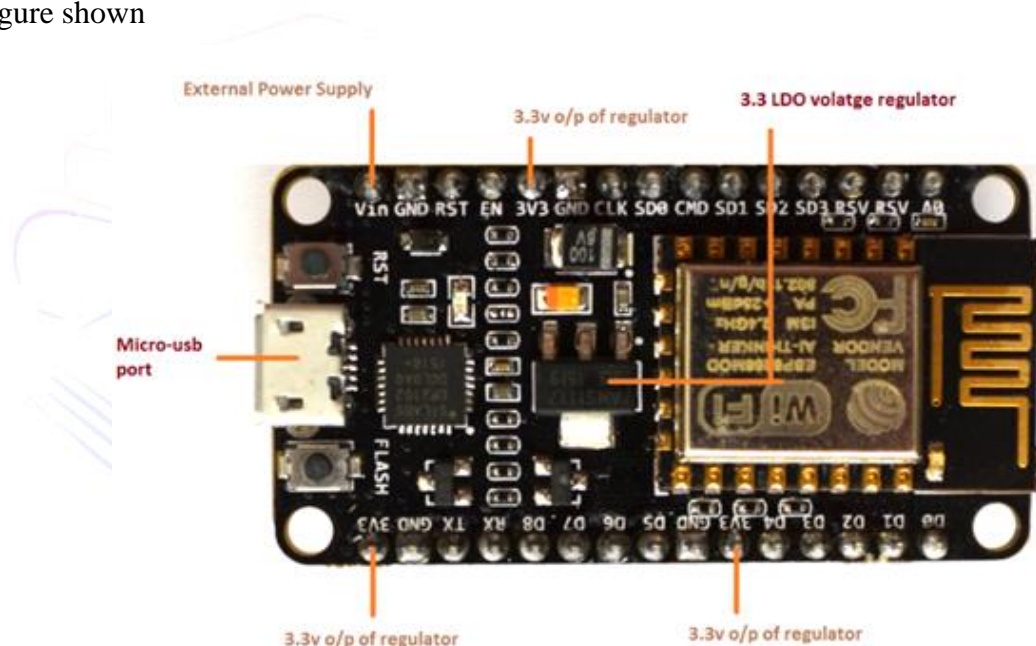
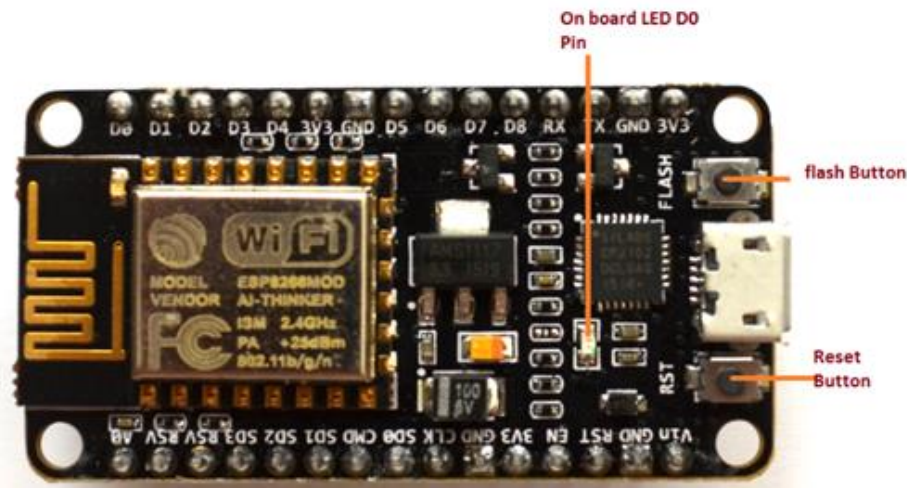


Figure 2.4: NodeMCU Power Requirements

### B- On Board Switches and LED Indicator:

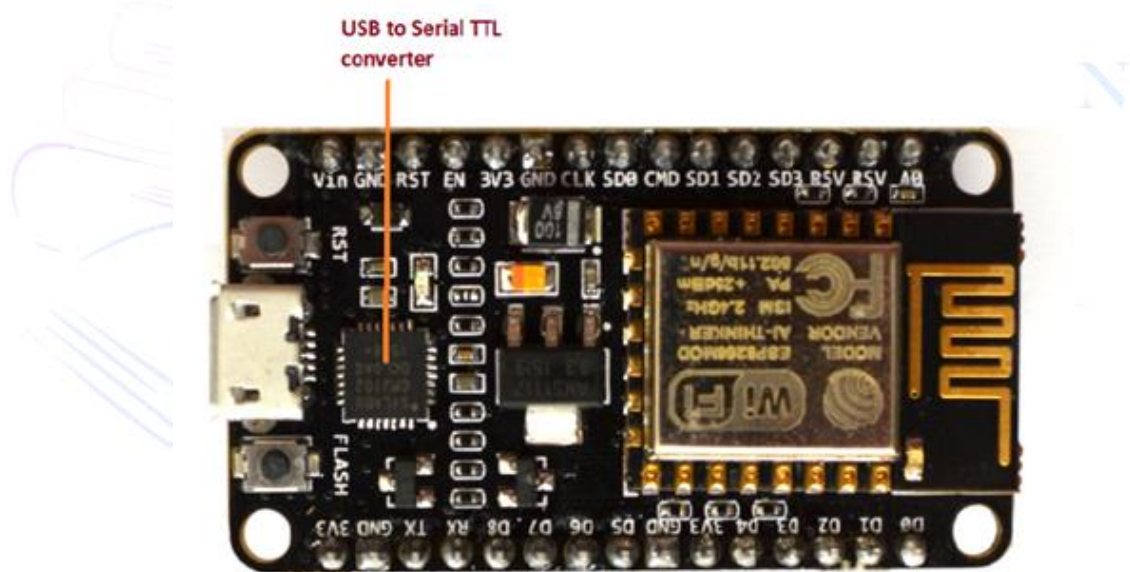
ESP8266 comes with 2 switches, one is reset, the other is flash button, reset button is used to reset nodemcu, flash button is used to download and upgrade firmware. The board also has a built-in LED indicator that is connected to the D0 pin and can be programmed. as figure shown.[6][7]



**Figure 2.5: NodeMCU Board Switches and LED Indicator**

### C-Serial Communication:

Nodemcu includes the CP2102 USB to UART bridge controller that helps convert USB signals to serial and allows your computer to program and communicate with the ESP8266 chip. It also has a communication speed of 4.5Mbps. as the picture shows.



**Figure 2.6: NodeMCU Serial Communication:**

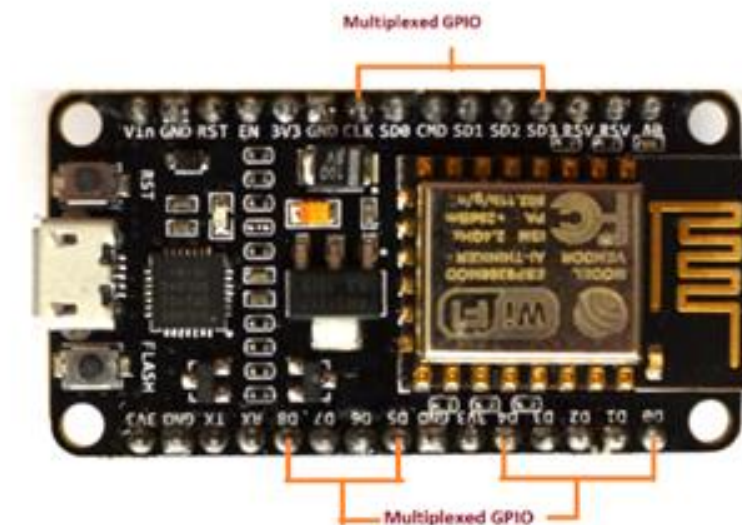
### D- Input/Output Peripherals and Pins:

The ESP6266 Nodemcu board has 17 GPIO pins, which are assigned to various peripheral tasks such as

- UART interface
- PWM output
- I2S, I2C, SPI interfaces: connect numerous devices and sensors
- A 10 bit ADC channel

as figure shown





**Figure 2.7: NodeMCU I/O Pins and Peripherals:**

#### **E- ESP8266 NodeMCU Pinout:**

- **Power Pins**

There are four power pins, ie. One VIN pin and three 3.3V pins. If you have a regulated 5V voltage source, the VIN pin can be used to directly power the ESP8266 and its peripherals. The 3.3V pin is the output of the onboard regulator. These pins can be used to power external components.

- **Ground**

It is the ground pin of the ESP8266 NodeMCU board.

- **I2C pins**

These pins are used to connect various I2C sensors and peripherals in the project. I2C Master and I2C Slave are supported. The I2C interface function can be programmed with a clock frequency of up to 100 kHz. It is important to note that the slave device's slowest clock frequency must be less than the I2C clock frequency [7].

- **GPIO Pins**

The ESP8266 NodeMCU has 17 GPIO pins that could be programmed to assign to numerous uses like PWM, UART, I2S, I2C, IR remote, buttons, and LED lights. There are a variety of options for each digital enable GPIO, including high impedance, pull-up, and pull-down. It can be set to edge-trigger or level-trigger to generate CPU interruptions when configured as an input [6].

- **SPI Pins**

ESP8266 has two SPIs (SPI and HSPI) in slave mode and master mode.

These SPIs also support the following general-purpose SPI functions:

- 1- 4 timing modes of SPI format transmission
- 2- Divide clock up to 80 MHz and 80 MHz
- 3- FIFO up to 64 bytes

- **SDIO Pins**

The ESP8266 has a Secure Digital Input/Output Interface (SDIO) for SD cards direct connection. 4-bit 50 MHz SDIO v2.0 and 4-bit 25 MHz SDIO v1.1 are backed.



- **PWM Pins**

There are four channels for Pulse Width Modulation (PWM) on the board. It is possible to program PWM outputs to power LEDs and digital motors. The scope of PWM frequency is tuned from 1000-10000  $\mu$ s, i.e. between 100 Hz-1 kHz.

- **Control Pins**

The ESP8266 is controlled by these pins. These pins comprise WAKE pin, reset pin (RST), and chip enable pin (EN).

- **EN pin**

Pulling the EN pin high enables the ESP8266 chip. The chip uses the least amount of power when pulled low.

- **RST pin**

The ESP8266 chip can be reset using the RST pin.

## 2.9. Sensor MQ-135

The MQ-135 gas sensor can identify hazardous chemicals and pollution, including ammonia ( $\text{NH}_3$ ), sulfur (S), benzene ( $\text{C}_6\text{H}_6$ ), and  $\text{CO}_2$ . This sensor features analog and digital output pins, just as the other gas sensors in the MQ series. The digital pin goes high when the concentration of these gases in the air exceeds the threshold limit. Using the integrated potentiometer, this threshold can be set. The analog output pins generate analog voltages that can be used to calculate the atmospheric concentrations of various gases.

The MQ135 air quality sensor module uses 150mA of power and runs at 5V. To get results that are truly accurate, it needs to warm up.[8]



**Figure 2.8 MQ135 Sensor**

## 2.10. About MQ135 Sensor

One of the MQ series of sensors most often used gas sensors is the MQ135, which is frequently employed in air quality monitoring apparatus. It features analog and digital outputs and runs between 2.5 and 5.0 volts. Below is a diagram of the MQ135 module's pinout and key parts. Are shown below.[8][9]

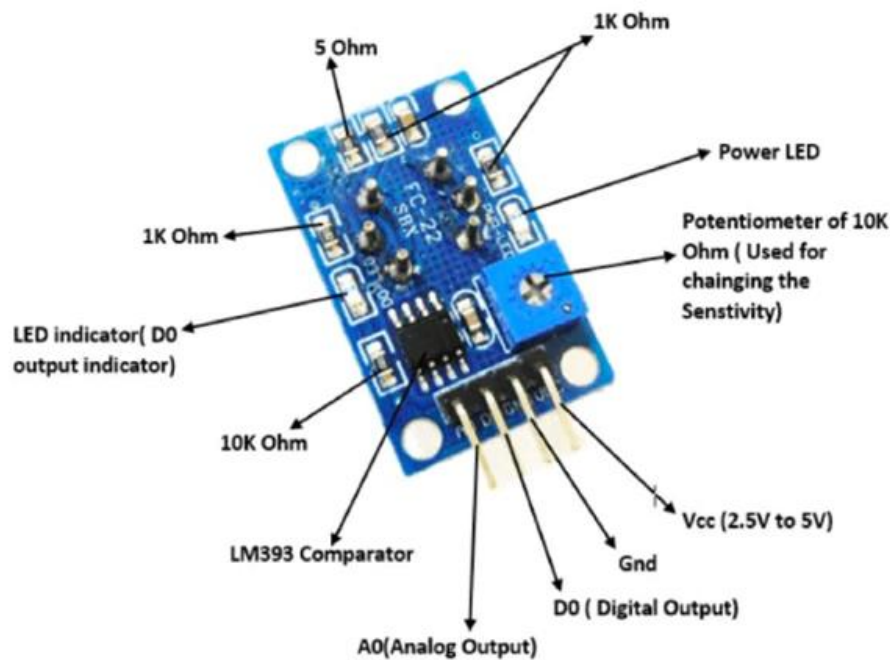


Figure 2.9 MQ135 Part

### Technical Specifications of MQ135 Gas Sensor

Operating voltage ranges from 2.5 to 5.0 volts, with a 150 mA power draw.

- NH<sub>3</sub>, Nox, CO<sub>2</sub>, Alcohol, Benzene, and Smoke Detection/Measurement
- 5 V is the typical operating voltage.
- 0 V to 5V (TTL Logic) Digital Output @ 5V Vcc
- 0-5V @ 5V Vcc Analog Output

#### 2.11. Detect Harmful Gases using Digital Pin:

The digital pins on a sensor's output can be used to recognize dangerous gases in the environment. Using a 10k potentiometer, the sensitivity of the digital pins can be adjusted. Indicator LED D0 will activate if gas is found, and the digital pins will switch from logic high to logic low (0V).

Additionally, the MQ135 gas sensor module can be applied without an external microcontroller. due to a onboard LM393 comparator IC. Just power up the module and use the potentiometer to set the sensitivity of the digital pin, if the module detects gas, the digital pin will go low. With a simple transistor, this digital pin can be used directly to drive a buzzer or LED.[9]

#### 2.12. Flame Sensor

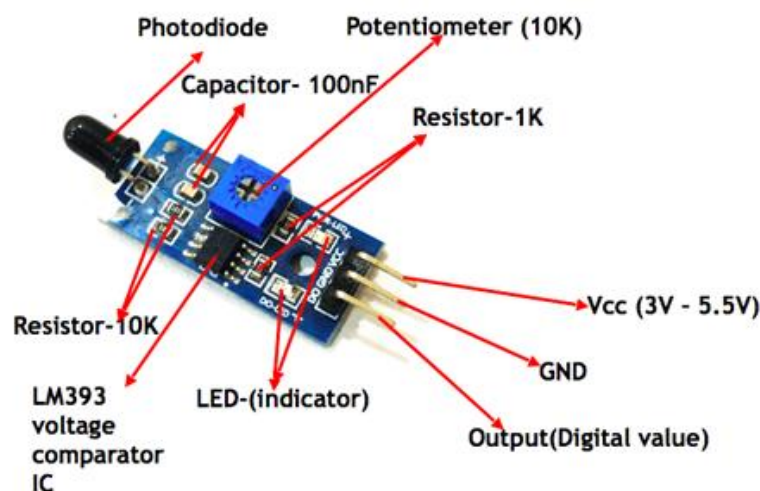
The sensor that detects and responds to flames or fires is known as a flame detector Depending on the installation, the response to a detected flame may include deactivating fuel lines (like a natural gas or a propane line), or sounding an alarm, and triggering the system of fire suppression. Flame sensors are often utilized in industrial furnaces as a means of confirming that the furnace is working appropriately, as well as turning off the ignition system, although they rarely perform any direct action beyond informing the control system or operator. As a result of the mechanisms used to detect the flame, a flame detector often responds faster and more accurately than a smoke or heat detector [10].



Photo by ElectroPeak

**Figure 2.10: Flame Sensor**

This flame sensor comprises of 10K-2N0 resistor SMD package, SMD led-2No, 10K- 1No potentiometer, photodiode, 1K-2N0 resistor SMD package, 100nF – 2No capacitor SMD package.



**Figure 2.11: Flame Sensor part**

### 2.13. Working of IR Flame/Fire Sensor Module

1- A photodiode's resistance decreases when it senses the IR waves emitted by the flame. IR waves generated by flames decrease resistance depending on their intensity. Photodiodes will act as short circuits when flames generate higher IR waves' intensity, offering the minimum resistance to the voltage received. Immediately afterward, the voltage at the non-inverting terminal will now decrease and become similar to the potential of the ground. The voltage appearing at the inverting terminal can be changed by varying the pot connected to the inverting terminal. This can be performed to alter the sensor sensitivity. (The module is said to have high sensitivity if the potentiometer is kept at the minimum value. While a module with low sensitivity is one where the potentiometer is set to its maximum value). The voltage at the inverting terminal is greater than the voltage at the non-inverting terminal at this stage (suppose that the voltage at the GND potential is lower than that at the inverting terminal), which causes the LM393 output to have a low value (0V). At the present stage, Due to the potential difference between its anode and cathode terminals, the indicator LED located between Vcc and Output will begin to glow. Pin 1 of the LM393 provides the output.

2- There are no IR waves detected by the photodiode when there is no flame. As a result, the circuit acts as an open circuit and offers maximum resistance to incoming voltage. Then this causes Vcc voltage will be noticeable at the non-inverting terminal. The LM393's output will be logic high (5V /VCC) as the voltage at

the non-inverting terminal exceeds the voltage at the inverting terminal (compared to  $V_{cc}$ , the inverting terminal has a low voltage). Due to the lack of potential difference between the anode and cathode terminals, the indicator LED connected between  $V_{cc}$  and Output will not glow at this stage. Pin 1 of the LM393 provides the output [10].

#### 2.14. Flame Sensor Pins

The flame sensor module has a total of 4 pins. Two power pins and two signal pins. The pinout of a flame sensor module is as follows:

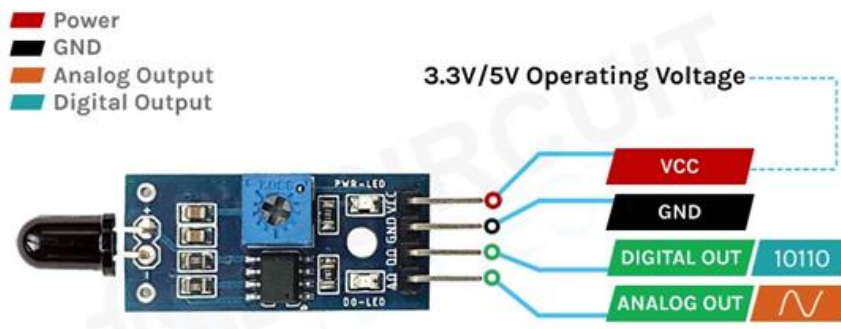


Figure 2.12: Flame sensor Pin

**CC** Provides power for the module, Connect to the 5V pin of the Arduino.

**GND** Ground Connected to Ground pin of the Arduino.

**DO** Digital Output Pin.

**AO** Analog Output Pin.

#### 2.15. LED (Light emitting diodes)

An unsung hero in the electronics realm is the light-emitting diode, commonly referred to as an LED. Essentially, the LED is simply a tiny light bulb that fits easily into electrical circuits. The big difference between them and ordinary incandescent bulbs is that none of their filaments can burn out, nor do they get very hot particularly. Their illumination is solely caused by electron movement in semiconductors [12].

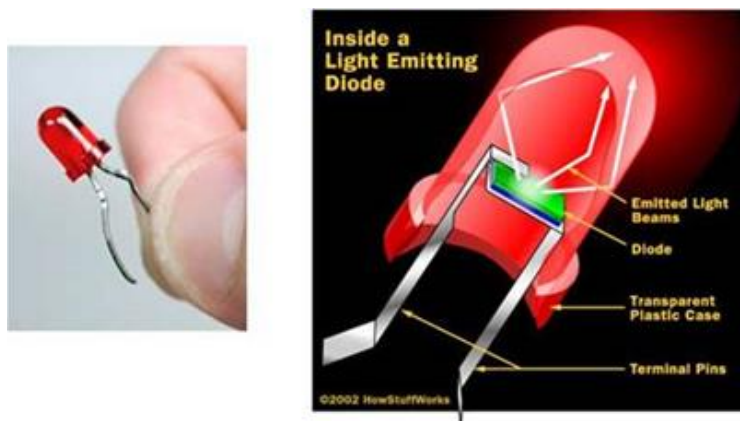


Figure 2.13: LED (Light emitting diodes)

In 2030, LEDs would be able to reduce general lighting energy consumption by nearly half, preserving energy bucks and emissions of carbon in this procedure. Many lighting applications benefit from their unique characteristics, involving store signs, night lights, traffic lights, exit signs, and holiday lights [12].



## 2.16. BUZZER

- The small bell and its sound are also suitable for electronic projects.
- buzzer voltage (5V).
- The buzzer has two electrodes (positive and negative), which connect the positive with the transmitter to control the Arduino, and the negative electrode with the ground on the Arduino uno.[11]



**Figure 2.14: BUZZER**

## 2.17. RemoteXY Application

1. RemoteXY is an easy way to make and use a mobile GUI for your controller board, which can be controlled from a smartphone or tablet. The system includes:
2. Mobile GUI editor for controller boards at remotexy.com
3. Smartphone App RemoteXY, It enables a connection to the controller and graphical control of it. Install the application. [13]

**RemoteXY** is an easy way to control microcontroller devices by making unique graphical interfaces from mobile applications such as Arduino.

### **allows RemoteXY:**

- Develop any GUI, using any combination of control, elements for display and decoration. For any task, a graphical interface can be created., using an online editor to place components on the screen. The web editor is published on the remotexy.com website.
- Once the GUI has been developed, You will have access to the microcontroller's source code, which implements your interface. The structure of source code allows your software to communicate with controls and interfaces So, it will be simple for you to include the management system into the function of the item you are constructing.
- Use a graphical interface on a mobile to operate microcontroller devices. For maintaining used mobile applications, use RemoteXY.
- Using one mobile App, With different display management interfaces, you will manage a number of devices. Since the microcontroller device stores the interface description. as the picture shows.[13]



Figure 2.15: Remote XY Application

### 3. SYSTEM PREPROCESSING STEPS

#### 3.1. Project circuit diagram

The circuit for the project consists of the NodeMCU ESP8266, and it is considered the main component in the project, which in turn distributes the tasks to the rest of the sensors associated with it, as in the figure below.

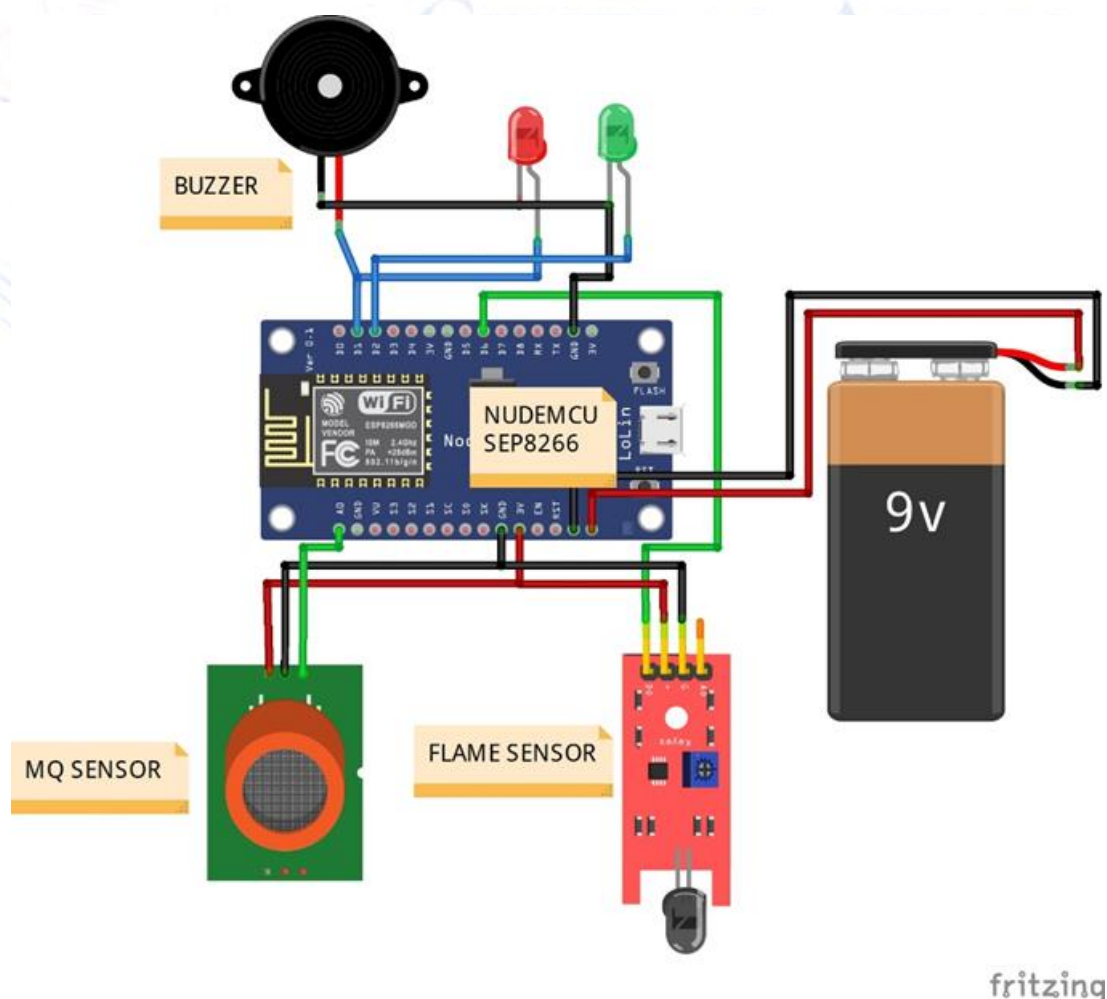
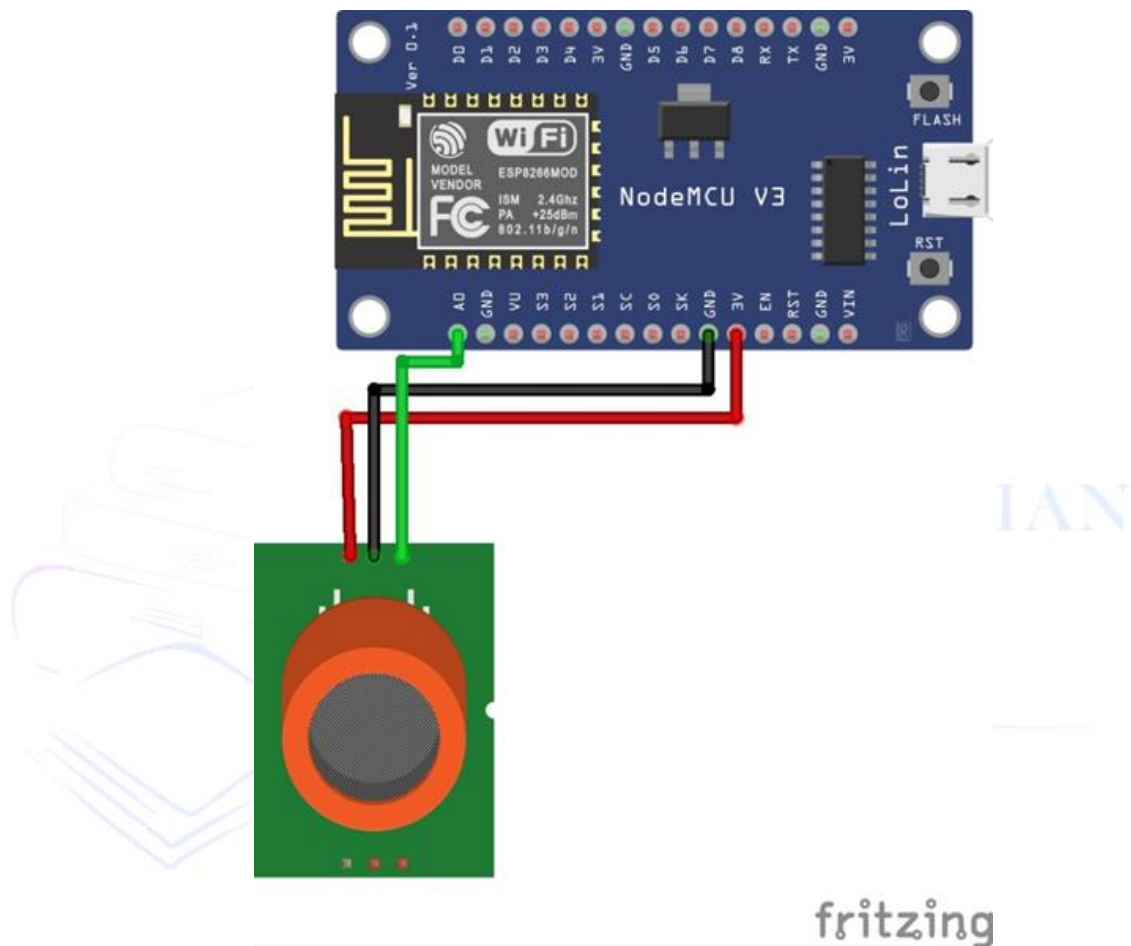


Figure 3.1 Project circuit diagram

Where the circuit is powered by a 9V battery, this battery feeds the ESP8266 and the rest of the sensor is fed from the ESP8266

### 3.2. MQ-135 Sensor Connection With NodeMCU ESP8266

Sensors (MQ-135) have three ports (VCC- GND -OUT), port (VCC) connected to NodeMCU ESP8266 port (3V), port (GND) connected to port (GND) on NodeMCU ESP8266 and finally the transfer port is sent The data is connected to the NodeMCU ESP8266 via the Analog port (A0), where the sensor reads the particles in the air and then analyzes them and sends the readings to the NodeMCU ESP8266, and the data is transmitted through the analog port, as in the diagram below



**Figure 3.2 MQ-135 sensor Connection With NodeMCU ESP8266**

### 3.3. Flame Sensor Connection With NodeMCU ESP8266

The flame sensor has three ports (VCC-GND -OUT), a (VCC) port connected to the NodeMCU ESP8266 port (3V), a (GND) port connected to the port (GND) on the NodeMCU ESP8266 and finally the transport port connected to the NodeMCU ESP8266 sends data via Digital (D6), where the sensor reads the molecules in the air and the flame in front of it and then analyzes it and sends the readings to the NodeMCU ESP8266, and the data is transmitted through the digital port to the application via Wi-Fi as in the diagram below

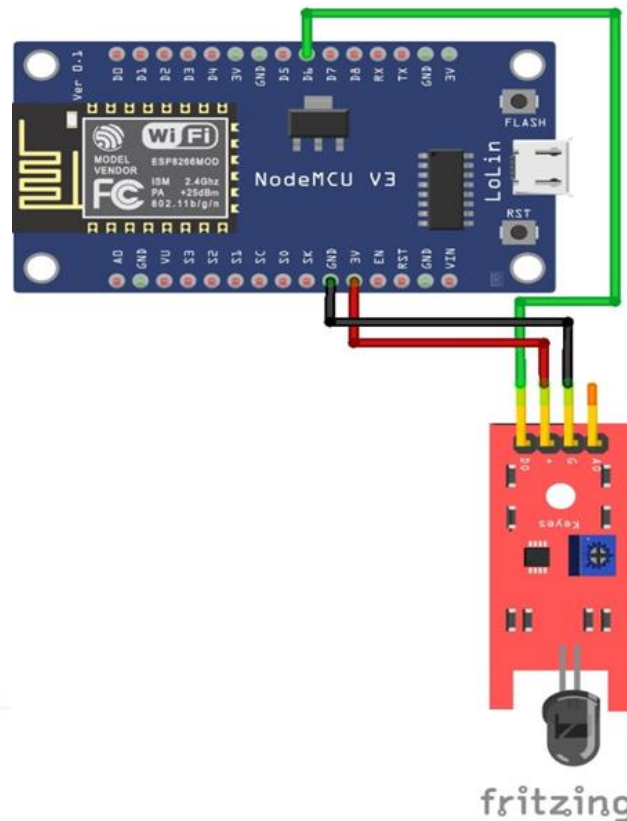


Figure 3.3 Flame sensor Connection With NodeMCU ESP8266

### 3.4. LEDs and BUZZER Connection With NodeMCU ESP8266

LEDs and buzzers generally contain only two poles (positive and negative), where the positive pole of the red LED is connected to the port (D1) in ESP8266 and the negative pole to the negative pole in (ESP8266), as for the buzzer, it is connected to the red LED because it works when it works For the red LED, as for the green LED, the positive pole is connected to the port (D2) in ESP8266 and the negative pole to the negative in ESP8266 As in the figure below

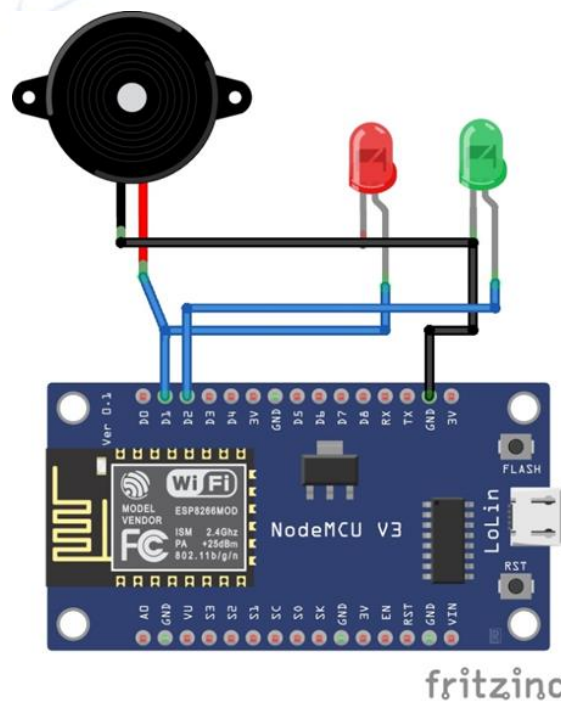


Figure 3.4 LEDs and BUZZER Connection With NodeMCU ESP8266



### 3.5. Power Supply Connection With NodeMCU ESP8266

A battery has been placed that can be charged several times in order to meet the needs of the project in general, as the circuit is equipped with a nine-volt power, connecting the negative pole and the positive pole to ESP8266, while the rest of the sensors take energy from the ESP8266 directly.. As in the figure below

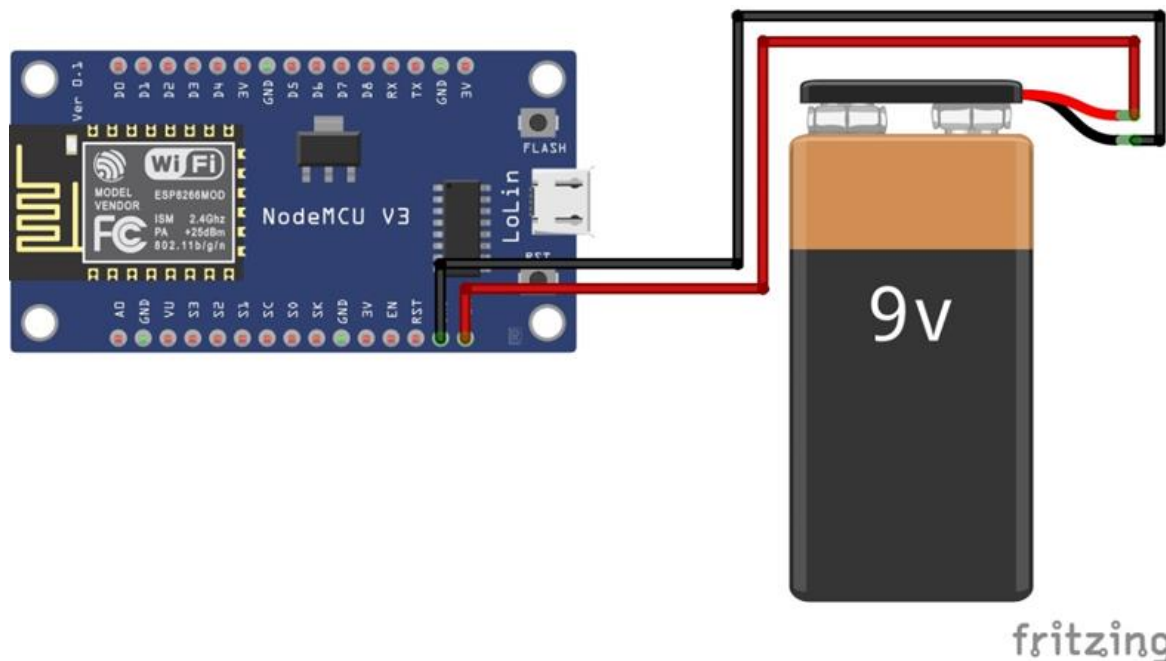


Figure 3.5 Battery 9v Connection With NodeMCU ESP8266

### 3.6. RemoteXY Application

1. RemoteXY is an easy way to make and use a mobile GUI for your controller board, which can be controlled from a smartphone or tablet. The system includes:
2. Mobile GUI editor for controller boards at remotexy.com
3. Smartphone App RemoteXY, It enables a connection to the controller and graphical control of it. Install the application. [15]

#### 3.6.1. Notable Features:

- The interface structure is stored in the controller. When connected, there is no interactive download interface with the server. The interface structure is install from the controller to the mobile App.
- One mobile application to control all your devices. Unlimited number of devices.

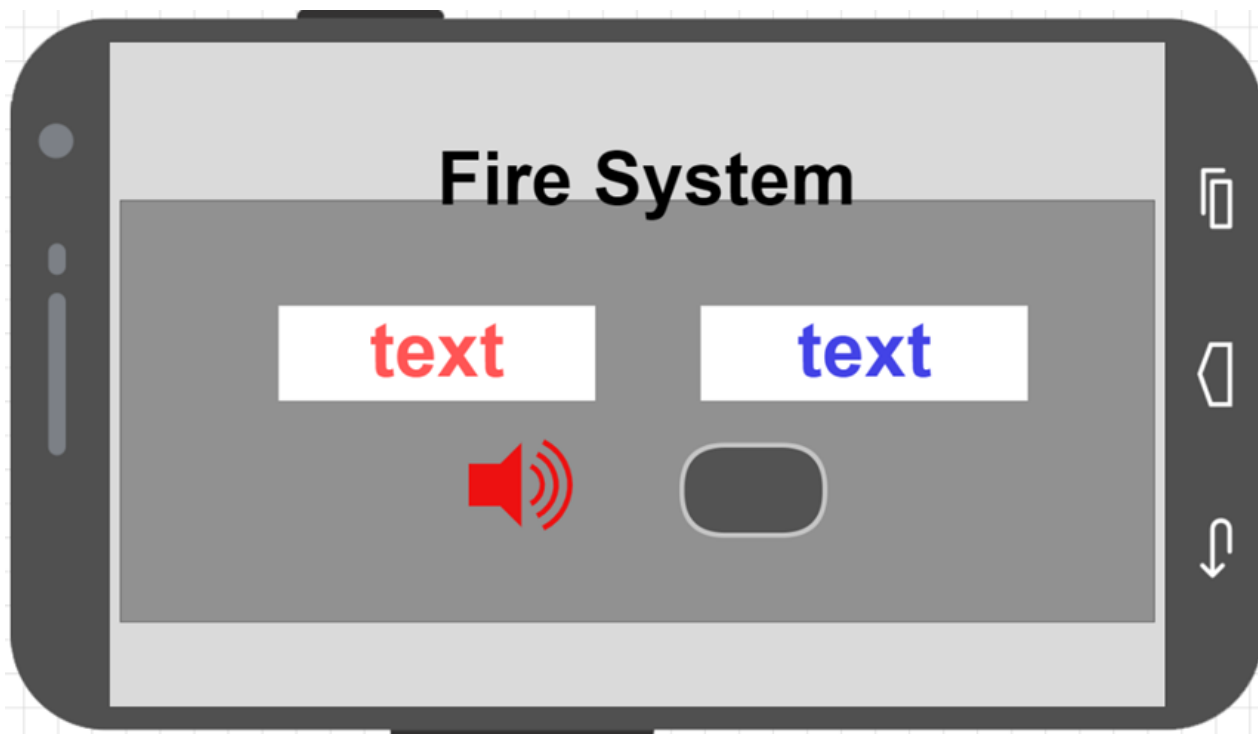
#### 3.6.2. Connection types:

1. cloud-based internet servers.
2. Wi-Fi clients and access points
3. Bluetooth.
4. URL or IP by Ethernet.

**RemoteXY** is an easy way to control microcontroller devices by making unique graphical interfaces from mobile applications such as Arduino.

**allows RemoteXY:**

- Develop any GUI, using any combination of control, elements for display and decoration. For any task, a graphical interface can be created., using an online editor to place components on the screen. The web editor is published on the remotexy.com website.
- Once the GUI has been developed, You will have access to the microcontroller's source code, which implements your interface. The structure of source code allows your software to communicate with controls and interfaces So, it will be simple for you to include the management system into the function of the item you are constructing.
- Use a graphical interface on a mobile to operate microcontroller devices. For maintaining used mobile applications, use RemoteXY.
- Using one mobile App, With different display management interfaces, you will manage a number of devices. Since the microcontroller device stores the interface description. as the picture shows.

**Figure 3.6 RemoteXY Application Design****3.7. Text string**

Any string information can be shown as text on a mobile screen. This could be the parameter or condition controller's real value. When using text, you must specify the structure's size (in bytes). RemoteXY will be given a buffer with the specified size. Text lines must end with a 0. Don't go over the limit of the variable you choose for text.15]

**Settings**

- **Variable Names** – the names of text and variables in the microcontroller source code, allows setting names for C-rule variables.
- **Length** - The number of characters assigned to the string variable. A single byte terminating zero is automatically added by the size of the text buffer.

**Example Code**

Text string passing, you want to insert the string into the corresponding field schema RemoteXY. Strings are zero-terminated. You can create string fields in a variety of ways.

Set any string immediately

```
strcpy (RemoteXY.text_1, "My text");
```

```
sprintf (RemoteXY.text_2, "My text");
```

### 3.8. Connect from the mobile application.

Install the RemoteXY mobile app to your phone.

Launch the app and click the + (plus) button on the top navigation bar. In the window that opens, select Cloud Server Connection. Enter your token in the Device token field of the window that opens. You created the token in step 3. Enter the token's alphanumeric characters, not the token's name. Press the connect button. If you did everything correctly, a graphical interface should open next.

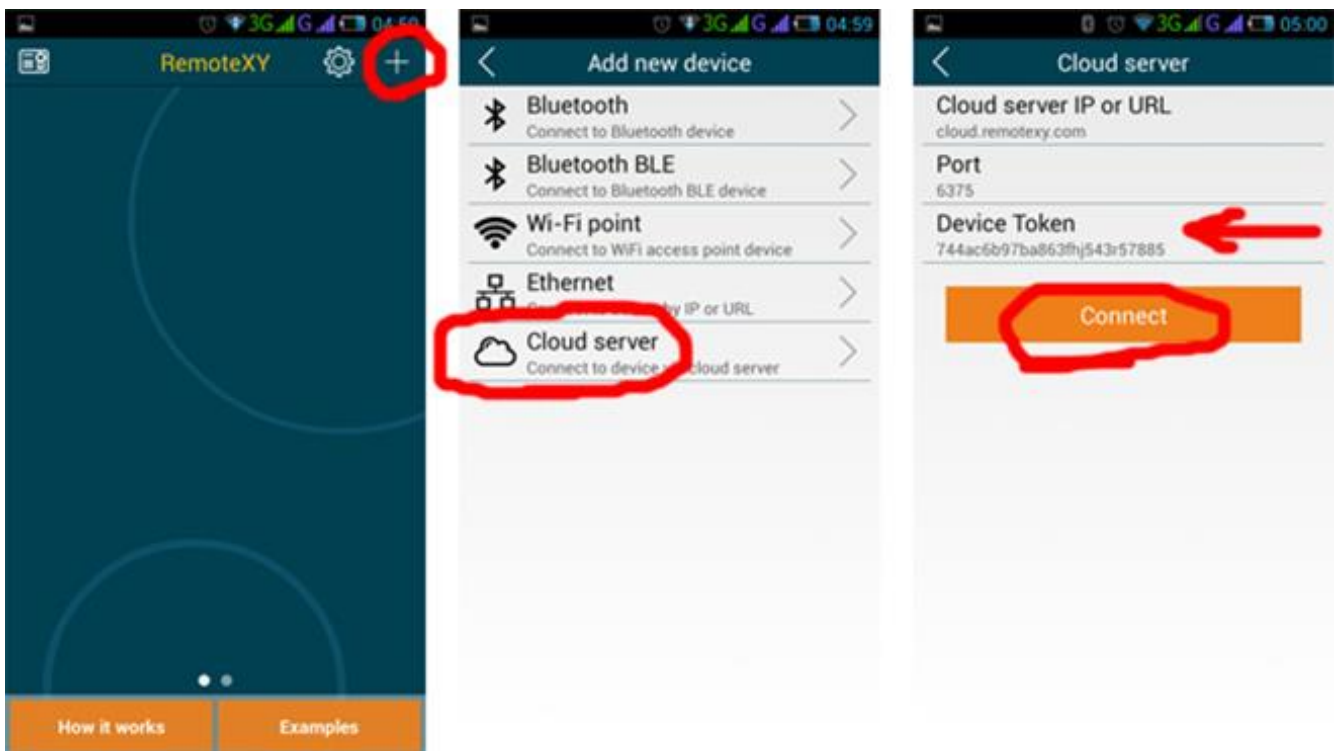


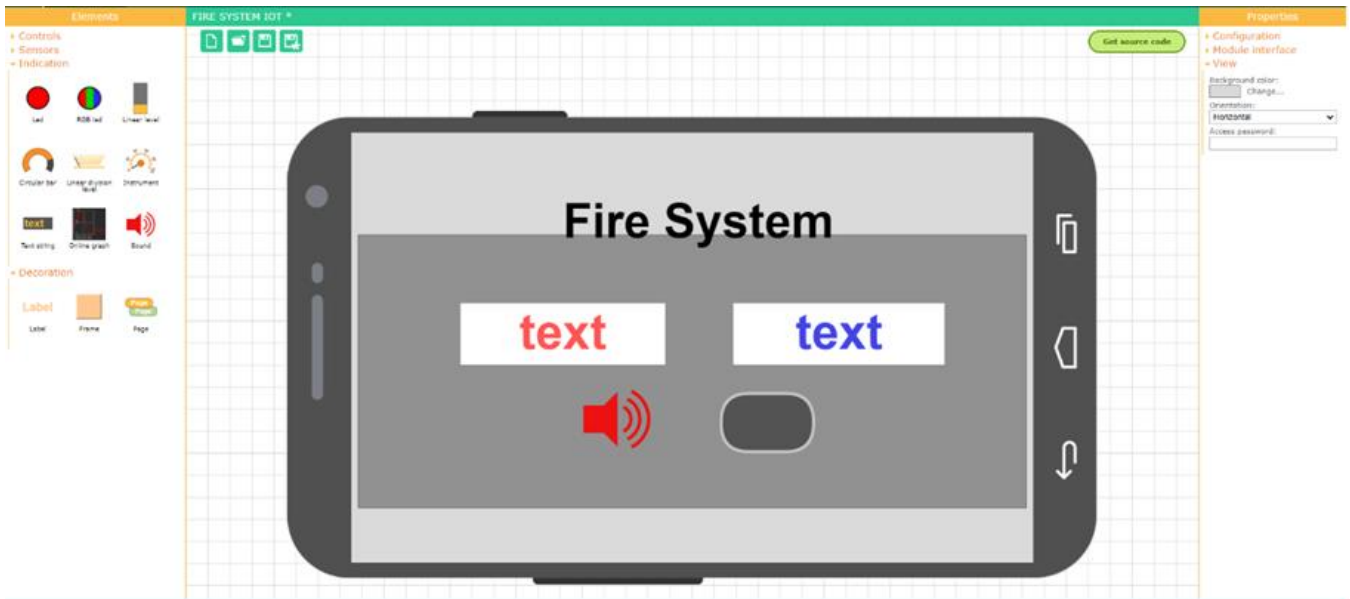
Figure 3.7 Connect from the mobile application

### 3.9. RemotyXY With NodeMCU ESP8266

- Using a web application, it is simple to create graphical user interfaces and generate the required source code for control boards (ESP8266, Arduino, etc.)
- Mobile application for Android, iOS allow for Wi-Fi-based or Bluetooth remote control of the remote board.

#### Step 1: Graphical Interface

Create graphical user interfaces using a straightforward drag-and-drop editor.



**Figure 3.8 Graphical Interface**

### Step 2: Source Code Configuration

In this manner, we instruct RemoteXY that the controller will be the nodeMCU board, the communication module will be the onboard ESP8266, and the programming environment will be Arduino.

In this situation, configuring the module interface section requires defining the SSID that the nodeMCU AP will broadcast.

In this situation, configuring the module interface section requires defining the SSID that the nodeMCU AP will broadcast.[15]

The TCP port on which the nodeMCU server will watch for inbound connections is also specified.

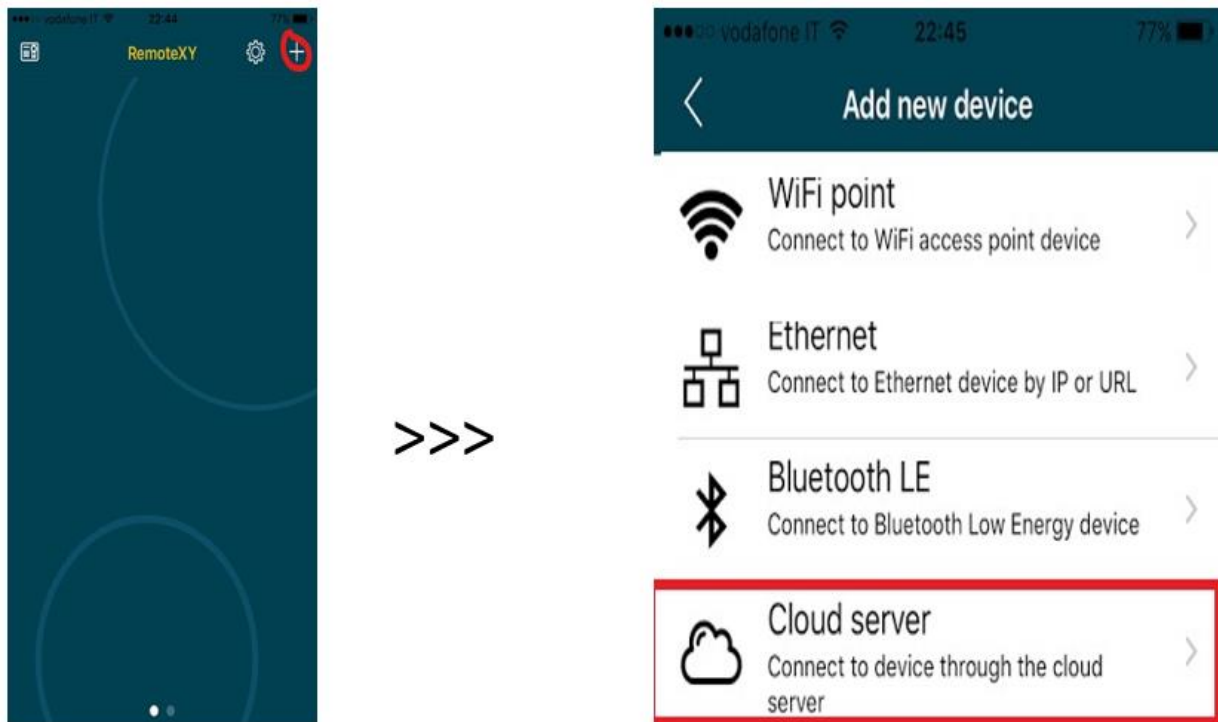
Once more, let's examine the open point for simplicity, which means Without authentication, a Wi-Fi network should be open.



**Figure 3.9 Source Code Configuration**

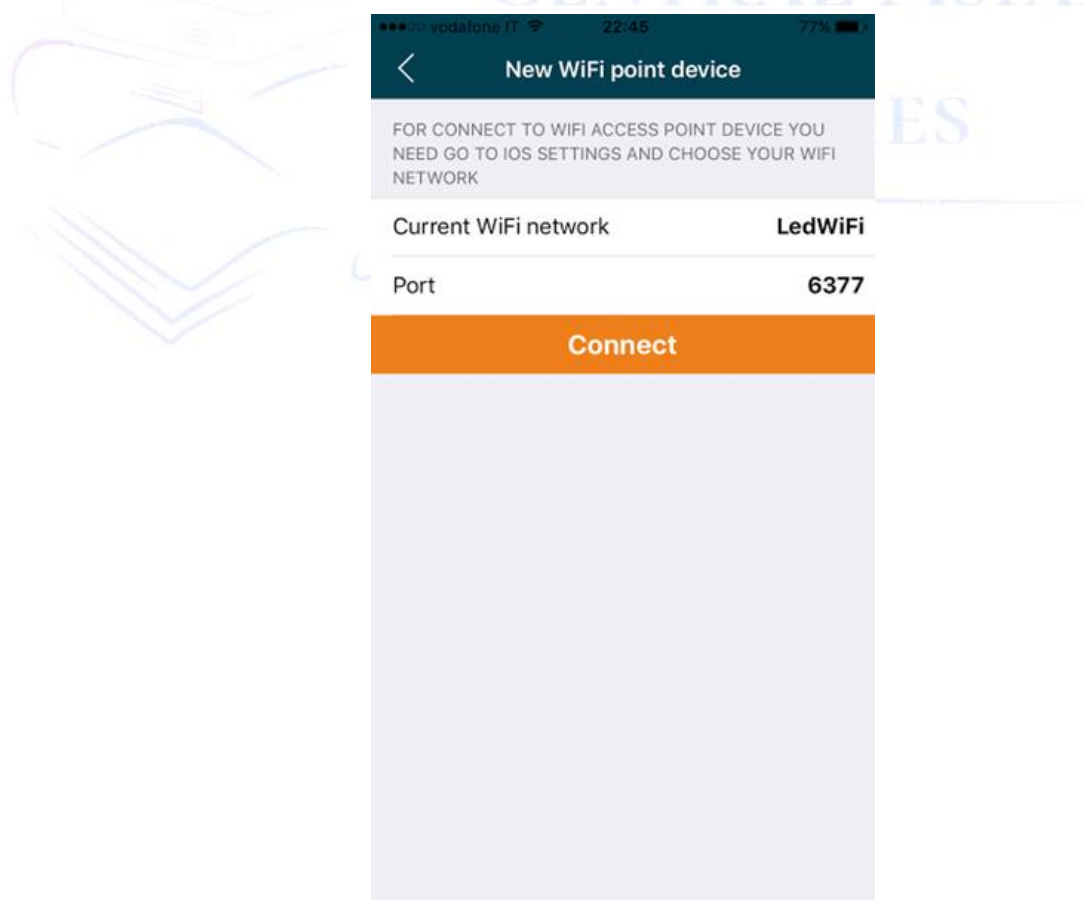
A- Run a RemoteXY app on your mobile device and click the + sign at the top





**Figure 3.10 Connection with Mobile**

A- Go to the **connect button** and the application connecting to the nodeMCU, then get the GUI.



A- Setting for remote control in the nodeMCU control panel

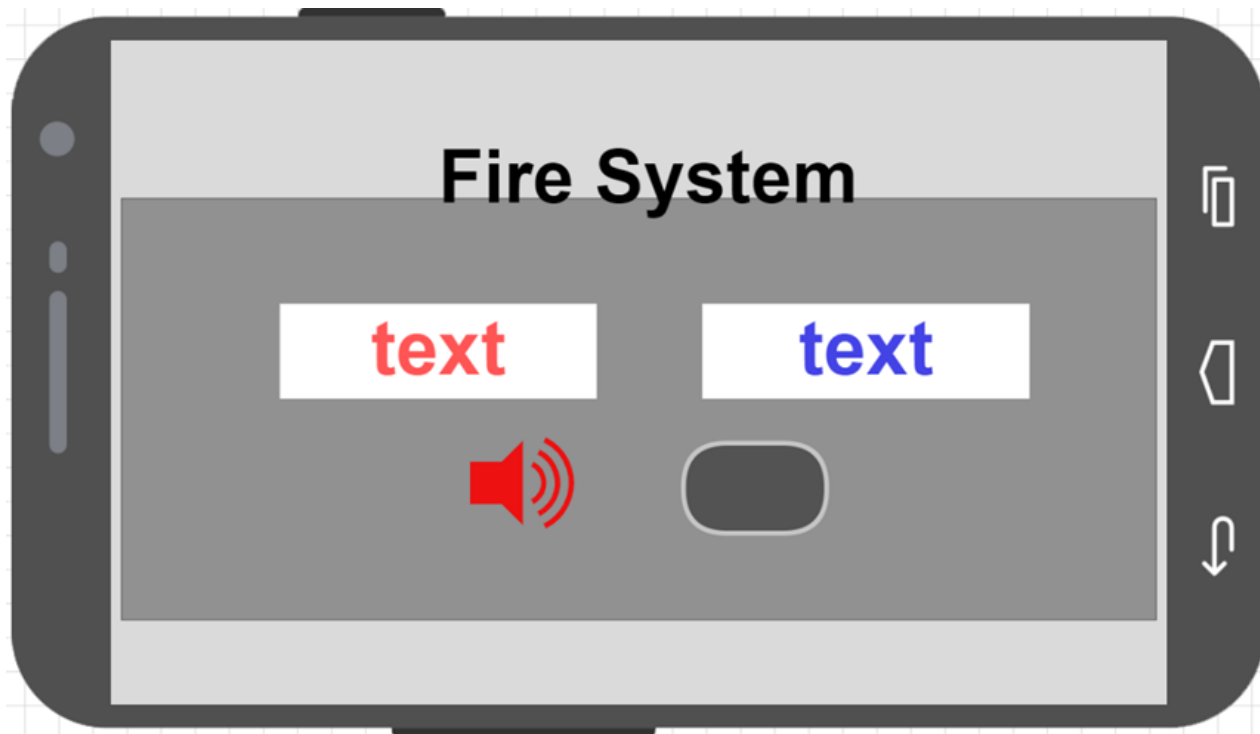


Figure 3.11 remote control of project

## 4. RESULTS

### 4.1. Result

The results of the project were very good, as the project was assembled with electronic materials, and the project was programmed according to the code that we wrote and loaded on the microcontroller.

When you open the phone application, the instructions from the project will reach via the cloud to the application to show alerts when there is a fire or gas. The purity of the air is sensed by the (MQ135) sensor. In the absence of fire or gases, the reading is normal and the green color turns on in the project and in the application. And when there is a fire or a certain gas emission, the sensors will sense this fire or gas emission, and thus the red color LED will turn on with the buzzer alarm in the project and in the application. As in the figure below

## 5. CONCLUSION AND FUTUR WORK

### 5.1. Conclusion

There are many reasons for designing a project to detect toxic and non-toxic gases, as well as to alert if there is a fire in a particular place,

One of the advantages of the project

1. It is possible to monitor via the Internet so that you can travel to different places while monitoring your home
2. It can be detected if there is a leak in other gases
3. Contains batteries that can be charged again

### 5.2. Future Work

*In the future, several other sensors will be added, such as the temperature and humidity sensor, to be sent also through the application, and the project can also be extended from a fixed object to a mobile one that is*

*sent to places containing fires to measure the level of gas leakage or the level of fire occurring in a specific location*

## Reference

1. ISO 7240-14:2013 | Fire detection and alarm systems — Part 14: Design, installation, commissioning and service of fire detection and fire alarm systems in and around buildings
2. ^ "CEN Community - List of members". *standards.cencenelec.eu*. Retrieved 26 March 2022.
3. "Getting Started: FOUNDATION > Introduction". *arduino.cc*.
4. ^ Jump up to: a b c David Kushner (2011-10-26). "The Making of Arduino". *IEEE Spectrum*.
5. ^ Justin Lahart (27 November 2009). "Taking an Open-Source Approach to Hardware". *The Wall Street Journal*. Retrieved 2014-09-07.
6. Kumar, Abhijeet, and Apoorva Sharma. "Internet of Life (IOL)." (2015). ISBN 978-93-5156-328-0
7. ^ Brian Benchoff. "An SDK for the ESP8266 Wi-Fi chip". *Hackaday*. Retrieved 2 April 2015.
8. [quartzcomponents.com/products/mq-135-air-quality-gas-sensor-module](https://quartzcomponents.com/products/mq-135-air-quality-gas-sensor-module)
9. [sensors/mq135-gas-sensor-for-air-quality](https://sensors/mq135-gas-sensor-for-air-quality)
10. IR Infrared Flame Sensor Module | How Fire Sensor Works » ElectroDuino
11. [buzzer - definition of buzzer by The Free Dictionary](#)". Retrieved 22 May 2015
12. HJ Round was a pioneer in the development of the LED". *www.myledpassion.com*. Archived from the original on October 28, 2020. Retrieved April 11, 2017
13. RemoteXY.com(Remotexy application)

