



SOMDIPTO MAJUMDAR
BATCH 20(1)

Microcontroller based Industrial applications

Industrial Fire Safety System

Title: Industrial Fire Safety System

Aim: To develop a prototype system which can generate and evacuate the workers under fire accident.

ABSTRACT:

This industrial safety system project aims to decrease the damage caused by fire outbreaks in industries due to leakage in petroleum, chemicals, and kerosene oil, which results in human loss and property damage. It is important to have a system in every place that can keep locations secure and appropriately give an alert in case of an emergency. It can also send the information to the Occupational Safety (OSH) team so that they can save and help the people in the workplace. The **Arduino (UNO)** system is built in such a way that it can detect **Ammonia, Sulphur, Benzene, Carbon dioxide**, and other harmful gases and smoke using **MQ-135 gas sensor** and fire using a flame sensor, while the **DHT-11** sensor is used to record temperature and humidity. If harmful gases get leaked the sensors get active and send the information to the **Arduino** and the **buzzer** gets active and the **LED bulb** glows and gives an alert to the people nearby.

INTRODUCTION:

Industrial safety system was developed to protect those working people from danger. The chemical reactions are protected by the industrial safety system, which also secures the environment and the plant itself. Over 4 million people worldwide pass away from injuries each year. In 2015, India reported 413,457 accidental injury-related fatalities. The manufacturing sector contributes significantly to injury morbidity and mortality. This project helps to reduce the risk to the workers working in the industry. If any harmful gas is leaking or a fire accident may happen then this device gives an alert to the worker so that they can save their lives. The main advantage of this industrial safety system project is that accidents can be avoided, and immediate action is taken by the people.

Components Required:

1. Hardware Component

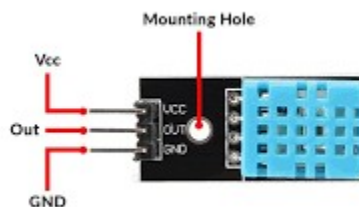
- [Arduino UNO R3](#)
- [DHT11 Sensor](#)
- [MQ135 Sensor Module](#)
- [Flame Sensor Module](#)
- [Buzzer](#)
- [Connecting Wires](#)
- [Breadboard](#)
- [LED](#)

2. Software Components

- [Arduino IDE](#)

COMPONENTS([Description](#)):

DHT11 SENSOR:



DHT sensor is used to measure the temperature and humidity. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air.

MQ135:



The **MQ-135 Gas sensor** can detect gases like Ammonia (NH_3), sulfur (S), Benzene (C_6H_6), CO_2 , and other harmful gases and smoke. The MQ135 air quality sensor module operates at 5V and consumes around 150mA. It requires some pre-heating before it could actually give accurate results.

FLAME SENSOR(NTC TEMPERATURE SENSOR):



This Flame Sensor measures temperature changes based on thermistor resistance. This module has both digital and analog outputs, there is a potentiometer to adjust the detection threshold on the digital interface.

BUZZER PIEZO:



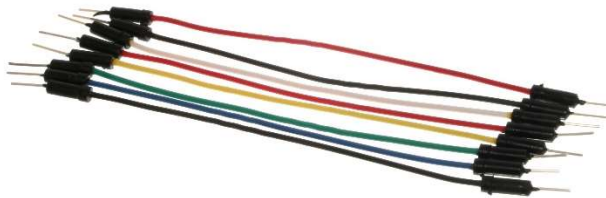
A piezo buzzer is a type of electronic device that's used to produce a tone, alarm or sound. It's lightweight with a simple construction, and it's typically a low-cost product.

LED:



A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it.

CONNECTING WIRES:



A connecting wire allows travels the electric current from one point to another point without resistivity. Resistance of connecting wire should always be nearly zero. Copper wires have a very low resistance and are therefore suitable for connecting wires.

ARDUINO UNO R3:



The Arduino Uno R3 is a popular open-source microcontroller board that is part of the Arduino platform. It is widely used for prototyping and creating electronic projects.

BREADBOARD:



A breadboard is a construction base used to build semi-permanent prototypes of electronic circuits.

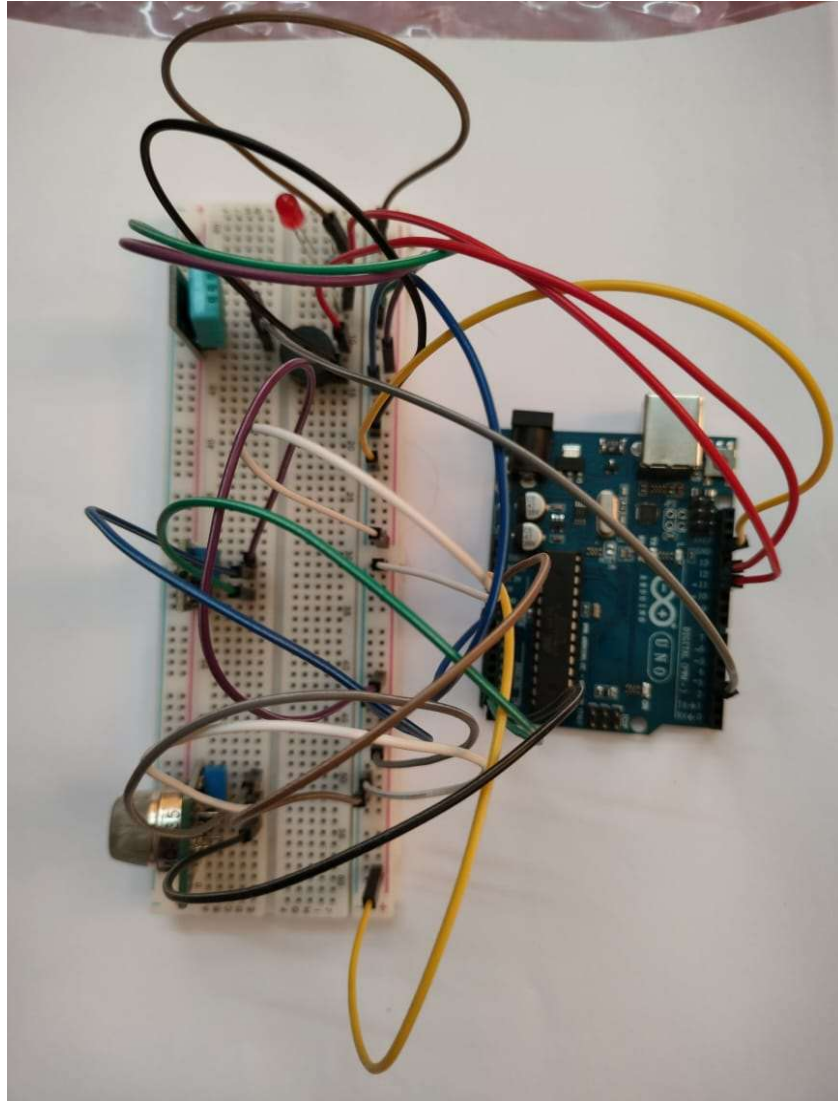
Software Requirements:

Arduino IDE :

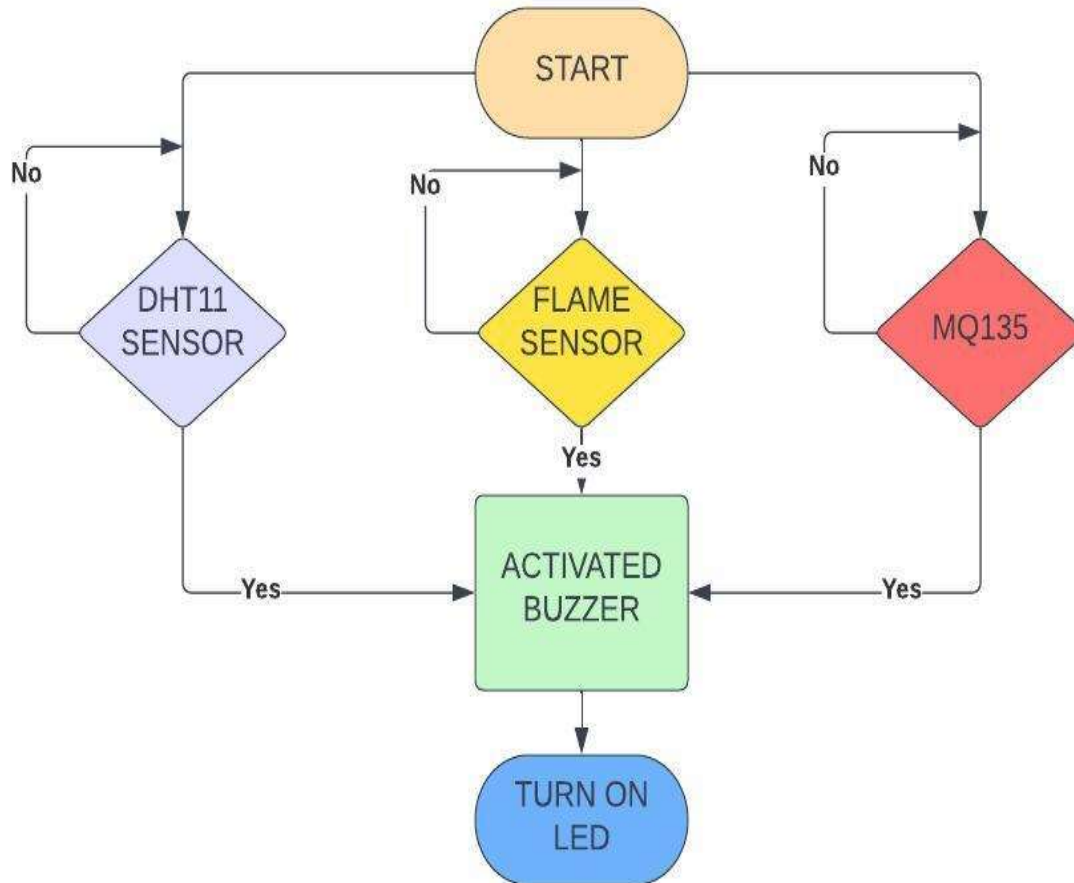


Arduino IDE (Integrated development environment) is software that is used to dump the program into boards. Arduino- IDE's primary use is to build electronics-related projects. Arduino is an open-source platform simple and easy-to-understand platform for coding.

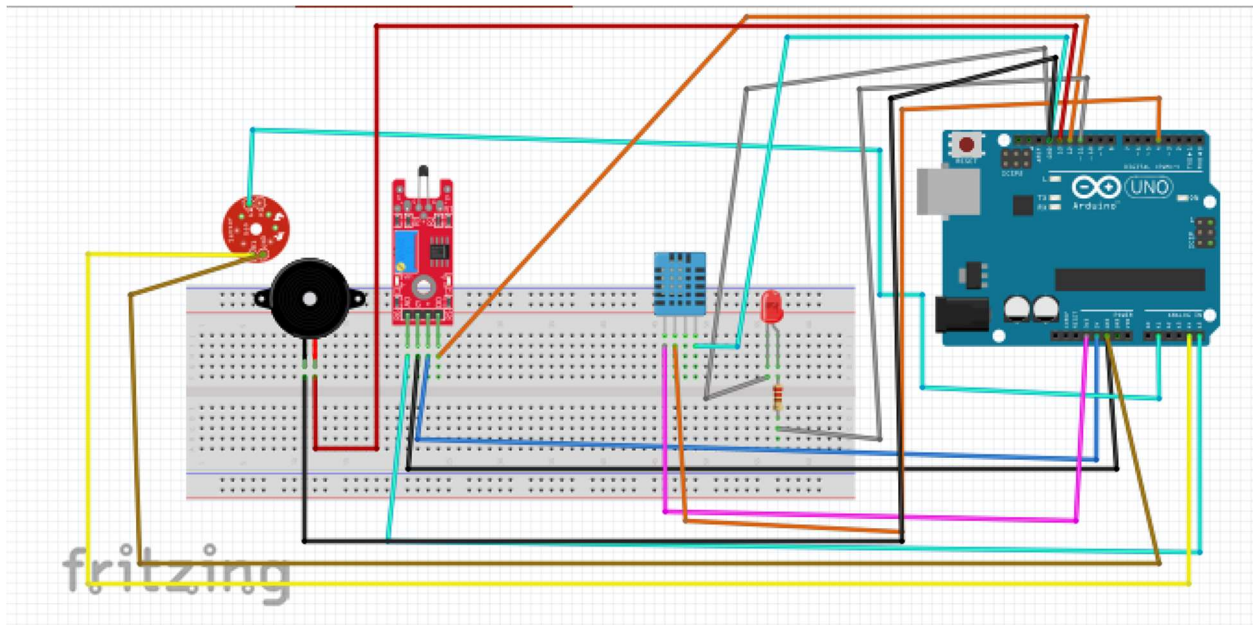
ARDUINO PROJECT SETUP:



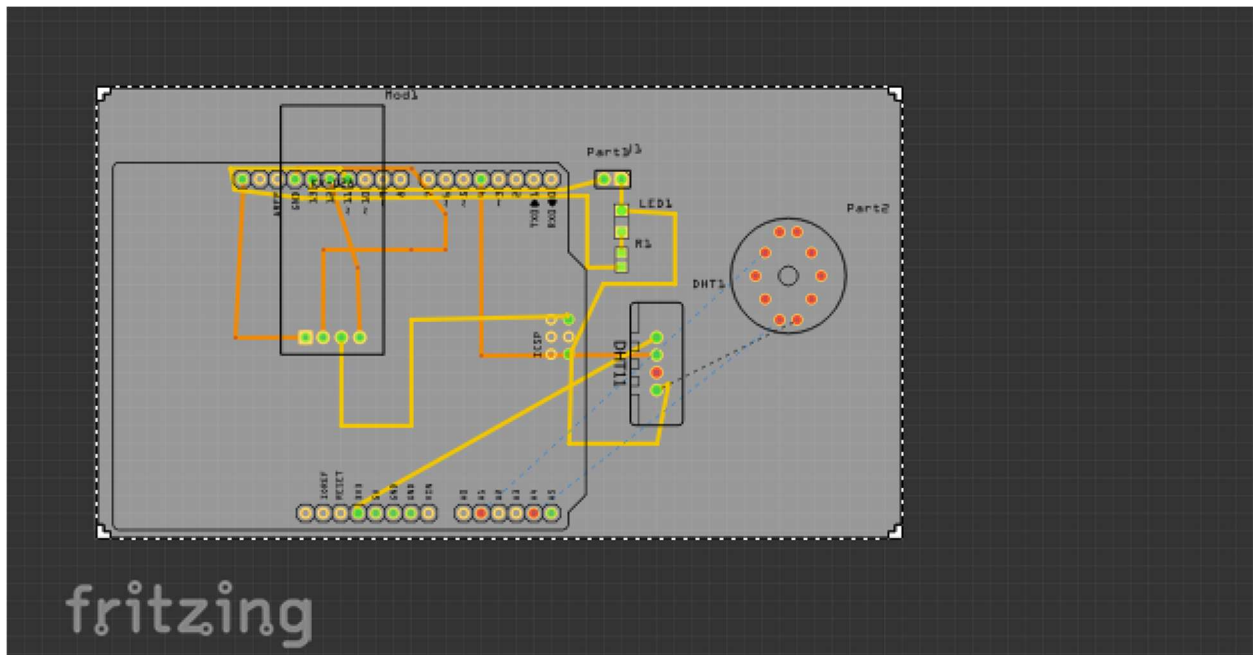
FLOWCHART DIAGRAM OF THE PROJECT SETUP



CIRCUIT DIAGRAM SETUP:



GERBER FILE (PCB LAYOUT):



Arduino IDE Code:

```
#include <SimpleDHT.h>

#define MQ135 A5          //smoke sensor pin connecetd to pin A0 of Arduino Uno
#define BUZZER_PIN 12      //buzzer pin connecetd to pin 8 of Arduino Uno
#define LED 8             //LED pin connecetd to pin 9 of Arduino Uno
#define flamePin A4

#define DHTPin 2

#define DHTTYPE DHT11  // DHT 11

#define GAS_THRESHOLD 200

#define TEMP_THRESHOLD 20


SimpleDHT11 dht;

int Flame = HIGH;

int value1;                //integer value is define to store the output of MQ135 sensor
int value2;

void setup()
{
    pinMode(MQ135, INPUT);    //Set MA3 as INPUT device
    pinMode(BUZZER_PIN, OUTPUT); //Set Buzzer as INPUT device
    pinMode(LED, OUTPUT);    //Set LED as INPUT device
    pinMode(flamePin, INPUT);

    Serial.begin(9600);
}

void mq135() {
    value1 = analogRead(MQ135);    // reads the analog value from smoke sensor
```

Somdipto Majumdar

```
Serial.println(value1);

if ( value1 > GAS_THRESHOLD )      //if smoke is detected
{
    digitalWrite ( LED , HIGH );   // turns the LED on
    digitalWrite(BUZZER_PIN,HIGH); // turns the buzzer on

}

else {
    digitalWrite(LED, LOW);        // turns the LED off
    digitalWrite(BUZZER_PIN,LOW);  // turns off
}

delay (500);
}

void flamesensor()
{
    Flame = digitalRead(flamePin);
    if (Flame== LOW)
    {
        digitalWrite(BUZZER_PIN, HIGH);
        digitalWrite(LED , HIGH);

    }

    else
    {
        digitalWrite(BUZZER_PIN, LOW);
        digitalWrite(LED , HIGH);
    }
}
```

Somdipto Majumdar

```
}  
}  
  
void loop(){  
    // Read data from DHT11 sensor  
    byte temperature = 0;  
    byte humidity = 0;  
    if (dht.read(DHTPin, &temperature, &humidity, NULL)) {  
        Serial.print("Read DHT11 failed.");  
        delay(1000);  
        return;  
    }  
  
    // Read data from MQ135 sensor  
    int gasValue = analogRead(MQ135);  
  
    // Print temperature, humidity, and gas values  
    Serial.print("Temperature: ");  
    Serial.print((int)temperature);  
    Serial.print(" *C, Humidity: ");  
    Serial.print((int)humidity);  
    Serial.print(" %, Gas Value: ");  
    Serial.println(gasValue);  
  
    // Check if the gas level is above the threshold  
    if (gasValue > GAS_THRESHOLD) {  
        // Blink LED and trigger buzzer  
        digitalWrite(LED, HIGH);
```

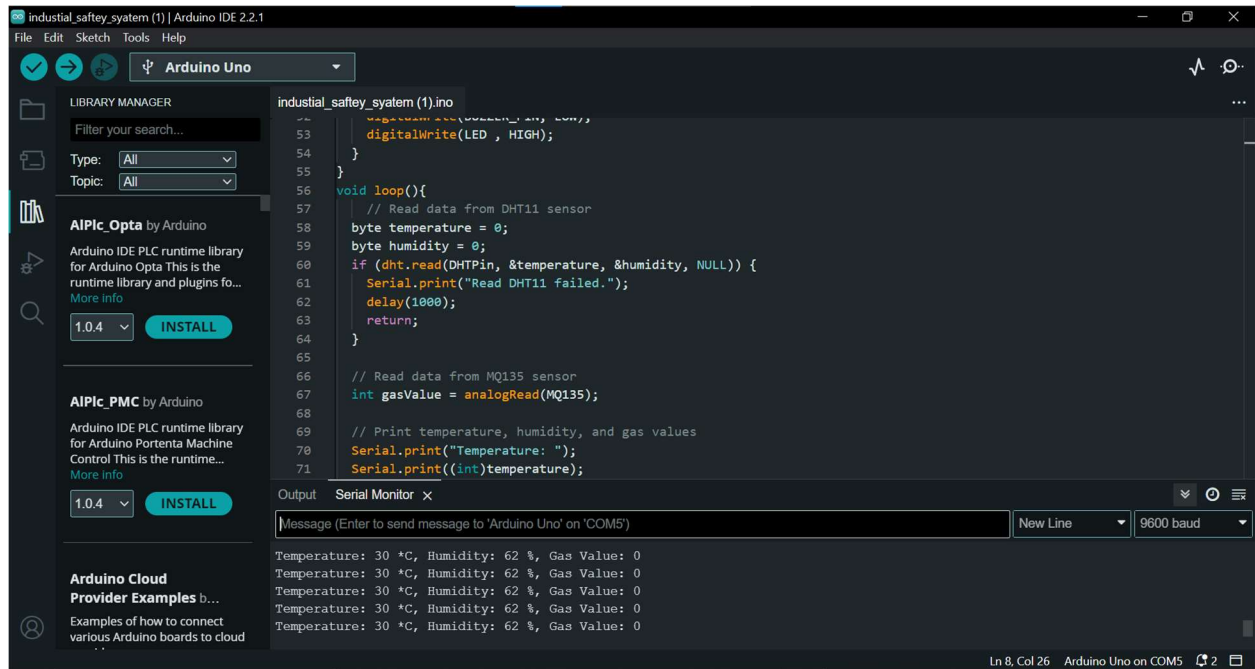
Somdipto Majumdar

```
digitalWrite(BUZZER_PIN, HIGH);  
delay(500); // Adjust the duration of the alarm as needed  
digitalWrite(LED, LOW);  
digitalWrite(BUZZER_PIN, LOW);  
}
```

```
// Check if the temperature is above the threshold  
if (temperature > TEMP_THRESHOLD) {  
    // Blink LED and trigger buzzer  
    digitalWrite(LED, HIGH);  
    digitalWrite(BUZZER_PIN, HIGH);  
    delay(500); // Adjust the duration of the alarm as needed  
    digitalWrite(LED, LOW);  
    digitalWrite(BUZZER_PIN, LOW);  
}
```

```
// Delay before the next loop  
delay(2000); // Adjust the delay based on your application  
}
```

WORKING CODE SNAPSHOT:



The screenshot displays the Arduino IDE 2.2.1 environment. The main editor window shows the sketch 'industrial_saftey_system (1).ino' with the following code:

```
53 digitalWrite(LED, HIGH);
54 }
55 }
56 void loop(){
57   // Read data from DHT11 sensor
58   byte temperature = 0;
59   byte humidity = 0;
60   if (dht.read(DHTPin, &temperature, &humidity, NULL)) {
61     Serial.print("Read DHT11 failed.");
62     delay(1000);
63     return;
64   }
65
66   // Read data from MQ135 sensor
67   int gasValue = analogRead(MQ135);
68
69   // Print temperature, humidity, and gas values
70   Serial.print("Temperature: ");
71   Serial.print((int)temperature);
```

The left sidebar shows the Library Manager with 'AIPlc_Opta' and 'AIPlc_PMC' libraries listed. The bottom panel shows the Serial Monitor output, which displays the following data:

```
Temperature: 30 *C, Humidity: 62 %, Gas Value: 0
Temperature: 30 *C, Humidity: 62 %, Gas Value: 0
Temperature: 30 *C, Humidity: 62 %, Gas Value: 0
Temperature: 30 *C, Humidity: 62 %, Gas Value: 0
Temperature: 30 *C, Humidity: 62 %, Gas Value: 0
```

RESULT:

Thus,in conclusion, the developed prototype system demonstrated effective fire detection, rapid worker alerting, and guided evacuation. Results indicate a promising solution for enhancing workplace safety in fire emergencies.