Industrial Fire Safety System

By
Rohan Chatla (VIT, Vellore - 21BEC0290)
Mid-term project for course on Microcontroller based Industrial Applications
25th October 2023

Abstract

This Fire 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. Along with flame sensor, the Arduino (UNO) based Fire safety system is built with the MQ-2 gas sensor to detect smoke and methane, while the DHT-11 sensor is used to record temperature and humidity. If a fire triggers or harmful gases get leaked or temperature exceeds certain threshold, appropriate sensors get active and send the information to the Arduino and the buzzer gets active and gives an alert to the people nearby.

Introduction

With 27,027 deaths, every fifth fire-related death in the world in 2017 took place in India. Around 9 million fire incidents and 1.2 lakh deaths were recorded across the globe that year. Of these incidents, India recorded 1.6 million fires and 27,027 deaths, according to a 195-nation analysis by Global Diseases Burden published in The BMJ Injury Prevention journal recently. 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 happens 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 around.

Requirements for Implementation

Hardware Requirements:

- **Arduino Uno:** Arduino Uno is a microcontroller from Arduino family, a friendly board based on the ATmega328. For more details, pinout and datasheet, one can visit: https://docs.arduino.cc/hardware/uno-rev3.

- **Temperature sensor (DHT11):** DHT11 sensor is used to measure the temperature and humidity. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air.

- **Flame sensor:** It detects the flame and responds to the presence of the flame.

- **Gas sensor (MQ2):** There are many MQ sensors to detect various hazardous gases. This sensor provides both digital and analogy output.

- **Buzzer:** The buzzer is a piezo device that generates sounds.

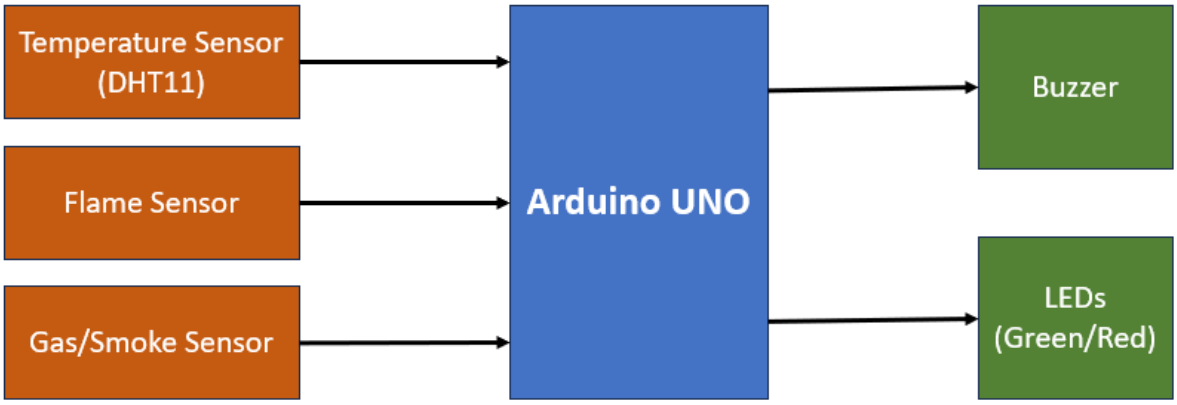
- **LEDs:** A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Red and Green LEDs are used to indicate the detection status.

Software Requirements:

Arduino IDE: Arduino IDE (Integrated development environment) is software that is used to load 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. More details can be found here: https://www.arduino.cc/en/software.

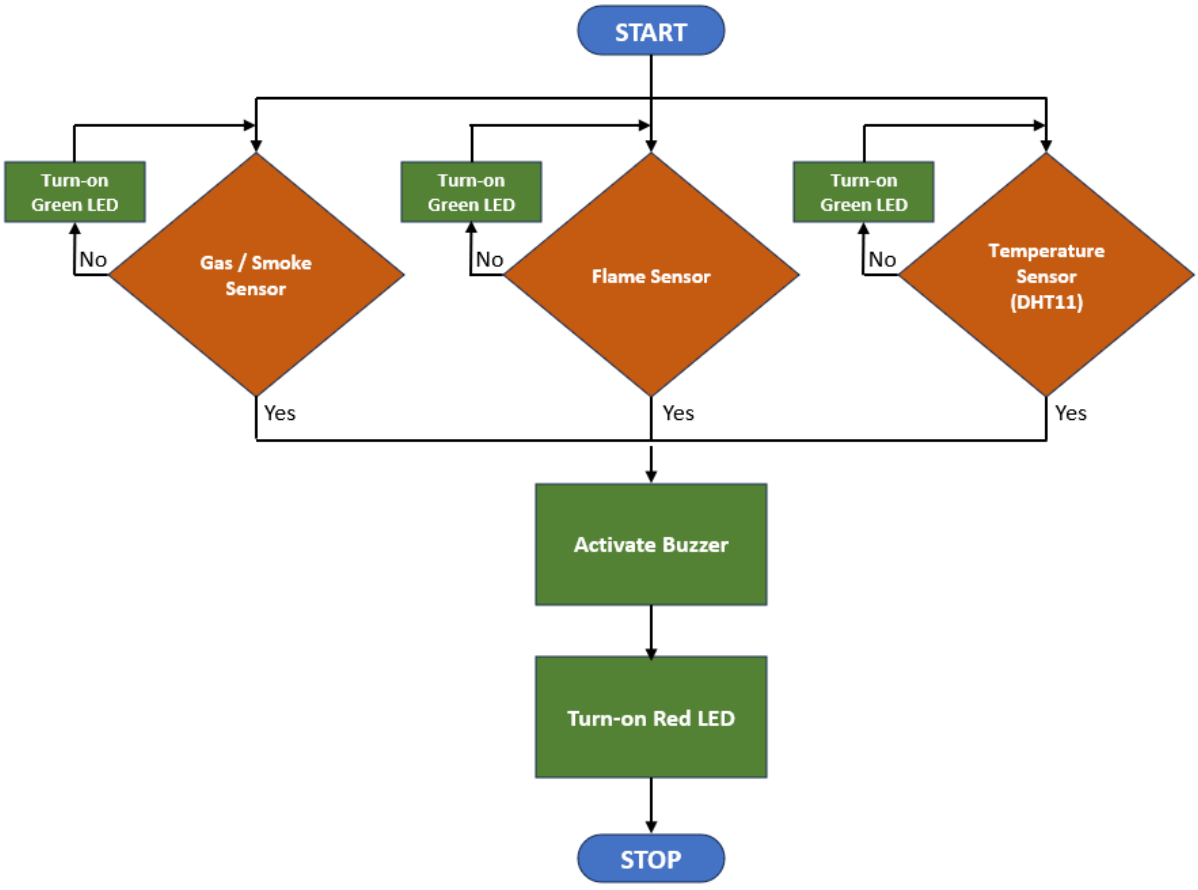
Block diagram

Following is the block diagram with temperature sensor (DHT11), Flame sensor and the gas sensor as input device that sends data to Arduino. Buzzer and LEDs as output devices that receive information from the Arduino.



Flow chart

Following is the flow chart of the implementation logic.



Pin Wiring

Gas / Smoke Sensor (MQ2)

The following pin connection is about the connection of the gas sensor with Arduino Uno. The gas sensor has 4 pins: MQ-2 is connected to 5v of Arduino Uno, analog pin to A0 and the Ground (GND) is connected to the GND of Arduino UNO.

Gas/Smoke sensor pins	Arduino UNO
VCC	5V
GND	GND
Digital out	-
Analog out	A0

Flame Sensor

The following pin connection is about the connection of the flame sensor with Arduino UNO. The flame sensor has 4 pins: The flame sensor is connected to 5v of Arduino UNO, analog pin to A2 and the Ground (GND) is connected to the GND of Arduino UNO.

Flame sensor pins	Arduino UNO
VCC	5V
GND	GND
Digital out	-
Analog out	A2

Temperature Sensor (DHT11)

The following pin connection is about the connection of the temperature sensor (DHT11) with Arduino UNO. The temperature sensor has 3 pins; DHT11 is connected

to 5v of Arduino UNO, digital pin to pin# 2 and the Ground (GND) is connected to the GND of Arduino Uno.

Temperature sensor pins (DHT11)	Arduino UNO
VCC	5V
GND	GND
Digital out	2

Buzzer

The below pin connection is about the connection of the buzzer with Arduino Uno. The buzzer has 2 pins: Buzzer is connected to ping 13 of Arduino UNO and the Ground (GND) is connected to the GND of Arduino Uno.

Buzzer	Arduino UNO
Positive	13
Negative	GND

LEDs

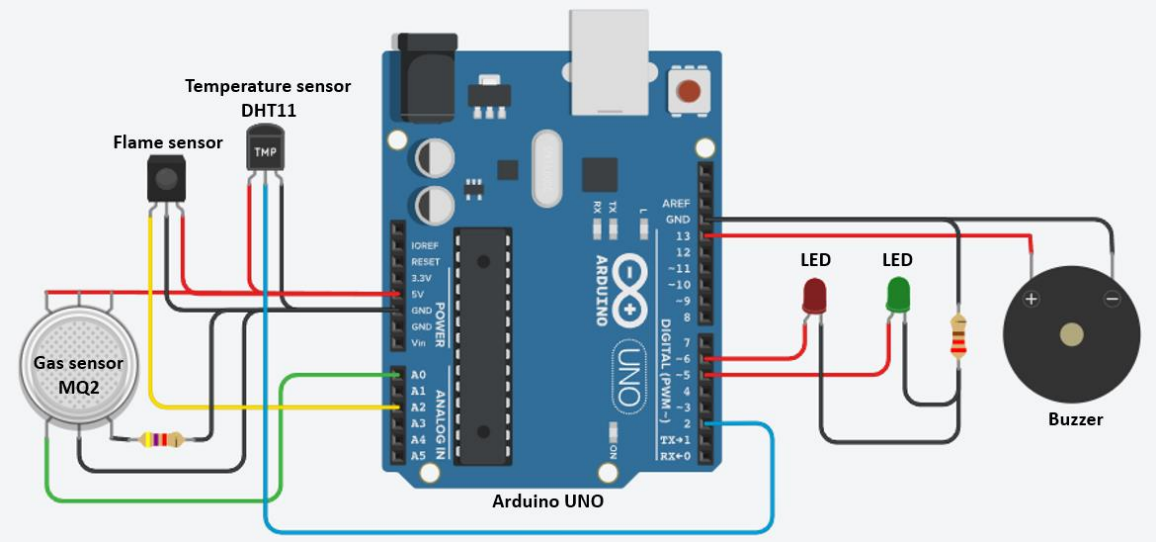
The below pin connection is about the connection of the LEDs (Red and Green) with the Arduino Uno. LEDs have 2 pins: Red LED is connected to pin# 6 and Green to pin# 5 of Arduino UNO. Ground is connected to the GND of Arduino UNO via ~260 ohms resister.

LED pins (Red / Green)	Arduino UNO
Positive	D6 / D5
Negative	GND

Circuit diagram

Simulated Circuit (tinkercad)

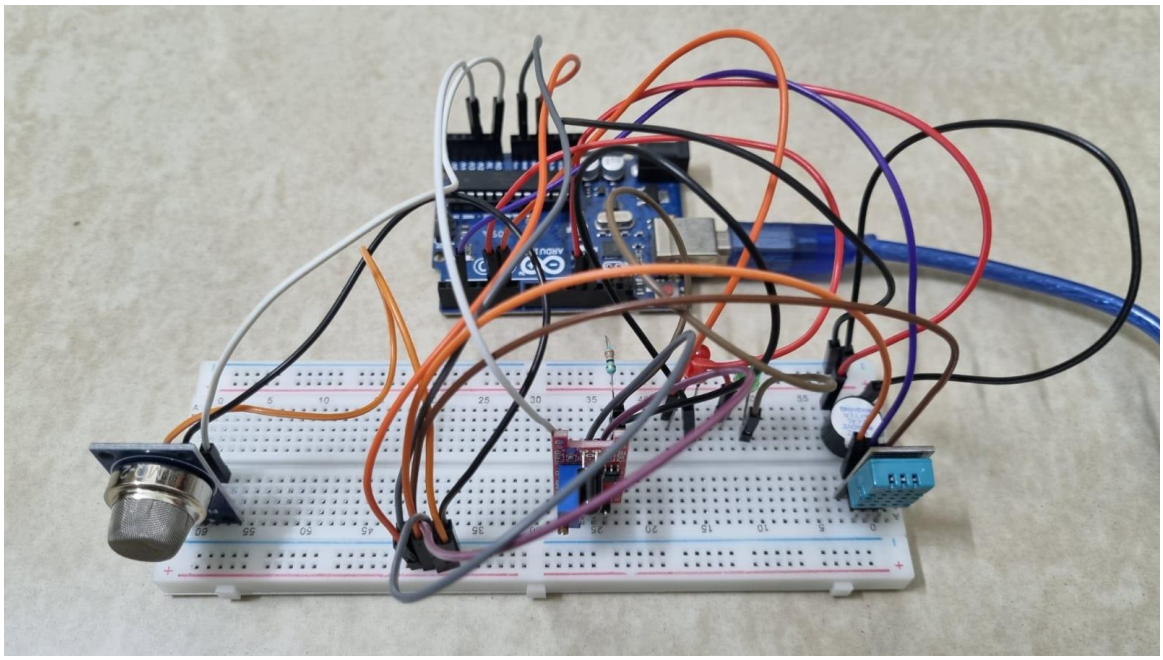
Follow the below circuit diagram to understand the connections made to different devices in tinkercad simulation tool.



Link to the working simulation circuit in tinkercad:

https://www.tinkercad.com/things/21JS9S86QMz?sharecode=HIC4eXvL-2ATykdpB88XFwgVlXGuQkvzF7Ecnl_2uBk

Picture of the actual working set-up



Working

The next step after setting up the connections is to upload the code to Arduino UNO and power it. When the system is turned on, Arduino gets input from the sensors, if any harmful gas and/or flame is deleted and/or raise in temperature then, the buzzer will ring and red led will be turned-on else they will be off.

Conclusion

Development of this Fire Safety System for industrial environment gave me an opportunity to experience the working of microcontroller and related sensors along with tools involved. Fire Safety System not only saves lives but also protects assets from destruction.

Working Code on the actual set-up

```
#include <dht.h>
#define RED_LED_PIN 6
#define GREEN_LED_PIN 5
#define BUZZER_PIN 13
#define DHT_SENSOR_PIN 2
#define SMOKE_SENSOR_PIN A0
#define FLAME_SENSOR_PIN A2
#define SMOKE_THRESHOLD 75
#define TEMP_THRESHOLD 40
#define FLAME_THRESHOLD 300

dht DHT;
int chk_dht;

void setup() {
  pinMode(RED_LED_PIN, OUTPUT);
  pinMode(GREEN_LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  Serial.begin(9600);
  pinMode(SMOKE_SENSOR_PIN, INPUT);
  pinMode(FLAME_SENSOR_PIN, INPUT);
}

void sirenAlarm(int buzzer) {
  for (int hz = 440; hz < 1000; hz++) {
    tone(buzzer, hz, 50);
    delay(3);
  }
```

```
  for (int hz = 1000; hz > 440; hz--) {
    tone(buzzer, hz, 50);
    delay(3);
  }
}

void loop() {
  chk_dht = DHT.read11(DHT_SENSOR_PIN);

  Serial.println("Smoke Val: " + String(analogRead(SMOKE_SENSOR_PIN)) + ", Temperature: " +
String(DHT.temperature) + ", Flame: " + String(analogRead(FLAME_SENSOR_PIN)));

  if (analogRead(SMOKE_SENSOR_PIN) > SMOKE_THRESHOLD || (DHT.temperature) > TEMP_THRESHOLD ||
analogRead(FLAME_SENSOR_PIN) < FLAME_THRESHOLD) {
    digitalWrite(RED_LED_PIN, HIGH);
    sirenAlarm(BUZZER_PIN);
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(GREEN_LED_PIN, LOW);
  } else {
    digitalWrite(RED_LED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
    noTone(BUZZER_PIN);
    digitalWrite(GREEN_LED_PIN, HIGH);
  }
  delay(400);
}
```