



# Industrial Safety Monitoring System

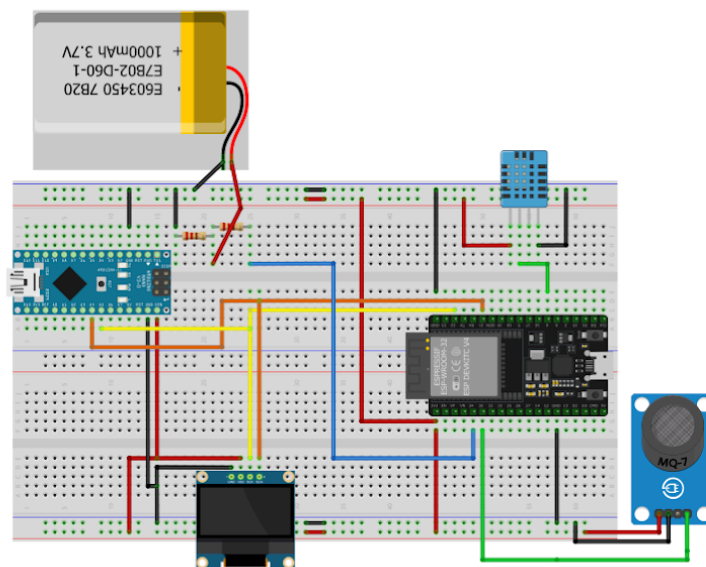
21z301 - AKSHAYAA M  
21z332 - PRATHIBHA G  
21z356 - SREENITHI R  
21z362 - SWETHA M  
21z363 - THIRISHA S

## Problem Statement

Existing methods for monitoring environmental conditions in industries lack real-time capabilities, risking worker safety and accidents. Manual checks and standalone sensors are inefficient, prompting the need for a robust Industrial Safety Monitoring System (ISMS).

The main challenge is to design an embedded system capable of continuous monitoring of temperature, smoke levels, and voltage, providing real-time data analysis and alerts in harsh industrial environments. Objectives include integrating sensors for comprehensive coverage, implementing anomaly detection algorithms, and ensuring seamless integration with existing control systems. Addressing these challenges will enhance workplace safety, reducing accidents and safeguarding industrial workers' well-being.

## Schematic diagram



## Arduino Program

```
#include <WiFi.h>
#include <WebServer.h>
#include <DHT.h>
#include <Adafruit_SSD1306.h>

#define DHT_PIN 4
#define SMOKE_SENSOR_PIN 35
#define VOLTAGE_SENSOR_PIN 34
#define OLED_SDA 21
#define OLED_SCL 22

#define OLED_WIDTH 128
#define OLED_HEIGHT 32

const char *ssid = "Battery Magmnt"; // SSID of the ESP32 soft AP
const char *password = "password"; // Password for the ESP32 soft AP

DHT dht(DHT_PIN, DHT11);
Adafruit_SSD1306 display(OLED_WIDTH, OLED_HEIGHT, &Wire, -1);
WebServer server(80);

float temperature = 0;
float humidity = 0;
int smokeLevel = 0;
float voltage = 0;
```

```
void setup() {
  Serial.begin(115200);
  Wire.begin(OLED_SDA, OLED_SCL); // Initialize I2C communication for the OLED display
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    while (1);
  }
  dht.begin();
  WiFi.softAP(ssid, password);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);
  server.on("/", handleRoot);
  server.begin();
}

void loop() {
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  smokeLevel = analogRead(SMOKE_SENSOR_PIN);
  voltage = analogRead(VOLTAGE_SENSOR_PIN) * (5.0 / 4095.0); // Convert analog reading to voltage

  updateDisplay();
  server.handleClient();
}
```

```

void handleRoot() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  int smokeLevel = analogRead(SMOKE_SENSOR_PIN);
  float voltage = analogRead(VOLTAGE_SENSOR_PIN) * (5.0 / 4095.0); // Convert analog reading to voltage

  // Check if smoke level exceeds threshold
  bool smokeDetected = (smokeLevel > 1550);
  bool voltageHigh = (voltage > 3.0); // Assuming threshold voltage is 3.0V
  // Generate the HTML response
  String webpage = "<!DOCTYPE html><html><head><title>ESP32 Sensor Readings</title>";
  webpage += "<style>";
  webpage += "body {";
  webpage += "  font-family: Arial, sans-serif;";
  webpage += "  background-color: #f0f0f0;";
  webpage += "  margin: 0;";
  webpage += "  padding: 0;";
  webpage += "}";
  webpage += "#container {";
  webpage += "  width: 80%;";
  webpage += "  margin: 20px auto;";
  webpage += "  background-color: #fff;";
  webpage += "  border-radius: 10px;";
  webpage += "  padding: 20px;";
  webpage += "  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);";
  webpage += "}";
  webpage += "h1 {";
  webpage += "  color: #333;";
  webpage += "}";
  webpage += "p {";
  webpage += "  color: #666;";

  webpage += "}";
  webpage += "#voltage-status {";
  webpage += "  color: red;";
  webpage += "  font-weight: bold;";
  webpage += "}";
  webpage += "</style>";
  webpage += "</head><body>";
  webpage += "<meta http-equiv='refresh' content='2'>";
  webpage += "<div id='container'>";
  webpage += "<h1>ESP32 Sensor Readings</h1>";
  webpage += "<p>Temperature: " + String(temperature) + " °C</p>";
  //webpage += "<p>Humidity: " + String(humidity) + " %</p>";
  webpage += "<p>Smoke Level: " + String(smokeLevel) + "</p>";
  webpage += "<p>Voltage: " + String(voltage, 2) + " V</p>";

  // Add voltage status
  if (voltageHigh) {
    webpage += "<p id='voltage-status'>High Voltage Detected</p>";
  } else {
    webpage += "<p id='voltage-status'>Normal Voltage</p>";
  }

  // Add smoke detection status
  if (smokeDetected) {
    webpage += "<p id='smoke-status'>Smoke Detected</p>";
  } else {
    webpage += "<p id='smoke-status'>No Smoke Detected</p>";
  }

  webpage += "</div>";
  webpage += "</body></html>";

  // Send the HTML response
  server.send(200, "text/html", webpage);
}

```

```
void updateDisplay() {
  bool smokeDetected = (smokeLevel > 1550);
  bool voltageHigh = (voltage > 3.0); // Assuming threshold voltage is 3.0V
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Temperature: " + String(temperature) + " C");
  //display.println("Humidity: " + String(humidity) + " %");
  display.println("Smoke Level: " + String(smokeLevel));
  display.println("Voltage: " + String(voltage, 2) + " V");

  // Add voltage status
  if (voltageHigh) {
    display.println("High Voltage Detected");
  } else {
    display.println("Normal Voltage");
  }

  // Add smoke detection status
  if (smokeDetected) {
    display.println("Smoke Detected");
  } else {
    display.println("No Smoke Detected");
  }

  display.display();
}
```

## Output

