**Construction Safety Monitoring System Documentation**

**The first home assignment for Construction Information Technology Project**

**Name:**

Munesh Kumar

**Neptun code:**

VGU6PB

# Table of Contents

[Table of Contents 0](#_Toc28731)

[1. Introduction: 1](#_Toc26808)

[2. Project Overview: 2](#_Toc16582)

[3. Components Used: 3](#_Toc15800)

[4. Installation Strategy: 4](#_Toc6784)

[i. Laser Module (Item 8): 4](#_Toc8323)

[ii. Photo-Interrupter Module (Item 12): 4](#_Toc13270)

[iii. Passive Buzzer (Item 7): 5](#_Toc23159)

[iv. RGB LED Module (Item 10): 5](#_Toc4790)

[v. Connection Information: 6](#_Toc15304)

[5. System Operation: 6](#_Toc2193)

[i. Flame Detection 6](#_Toc20971)

[ii. Laser Perimeter 6](#_Toc4012)

[iii. Temperature Alert 6](#_Toc20011)

[iv. Humidity Alert 6](#_Toc7131)

[v. Manual Override Button 7](#_Toc1527)

[5.1 Hardware Setup: 7](#_Toc16910)

[5.2 Software Setup: 7](#_Toc695)

[i. Code 7](#_Toc30048)

[ii. Testing and Calibration 8](#_Toc1317)

[iii. Safety Measures 8](#_Toc19848)

[6. Conclusion: 8](#_Toc16502)

[7. The code: 8](#_Toc10712)

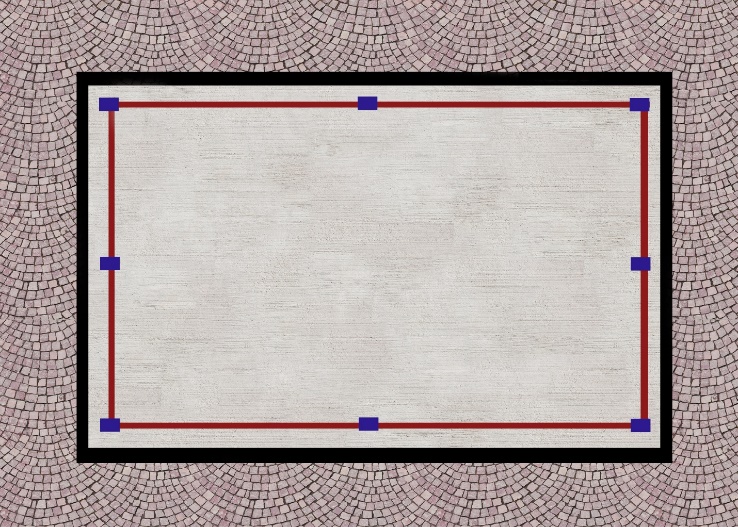
# Introduction:

With the expansion of technology and the field of architecture and the development of construction methods and structures, a new method has emerged in implementing buildings, more complex and dangerous than before, which has led to an increase in the risk factor for workers and supervising engineers on construction sites.

Unfortunately, according to OSHA (Occupational Safety and Health Administration), construction sites are the second most dangerous workplaces of all, from the aspect of the number of deaths and accidents. For example, one of the most common injuries workers can suffer is related to falling, as they slip on something, or a poorly constructed structure breaks off under them. These trips may have very serious health consequences and have to be handled with precaution.

Construction workers are also often exposed to open fire or flammable objects, and in many cases, they have to work under very challenging weather conditions, like rain, fealty, or high humidity.

Many indicator systems can signal dangerous areas and unsafe working conditions, our project work aims to establish a similar one with the capability of signaling different dangers.

 ****

# Project Overview:

The Construction Safety Monitoring System is designed to enhance safety measures at a construction site by covering three parts: fire alarm, temperature and humidity alarm, and security against falling from a roof’s edge.

The system incorporates various sensors and components to monitor conditions such as flames, abnormal temperature, and humidity. Visual and audible alerts are provided in real-time, and a manual override button is included for immediate response.

The utilization of advanced sensor technology, such as laser modules and detectors, provides a reliable means to create a virtual safety boundary on the roof. When a worker crosses this warning line, the system responds by triggering alarms, such as a buzzer sound and red LED lights, alerting both the worker and others in the vicinity to the potential danger. OSHA also limits the distance between the edge of a rooftop and the working area to 6 feet, so our entrance-detecting laser beam could be set up at that distance too.

Roof-top edge perimeter warning line system includes OSHA-compliant delineators, bases, and perimeter flags for increased fall protection and OSHA compliance.

Another sensor checks the temperature and humidity that have to be within an OSHA-defined range: 35 °C or 29 °C and 75% humidity. If these limits are exceeded, workers have the right to stop working, but sometimes there are not enough measurement devices that could signal values of temperature and humidity. When the temperature changes, the system responds by triggering alarms, such as a buzzer sound and LED lights.

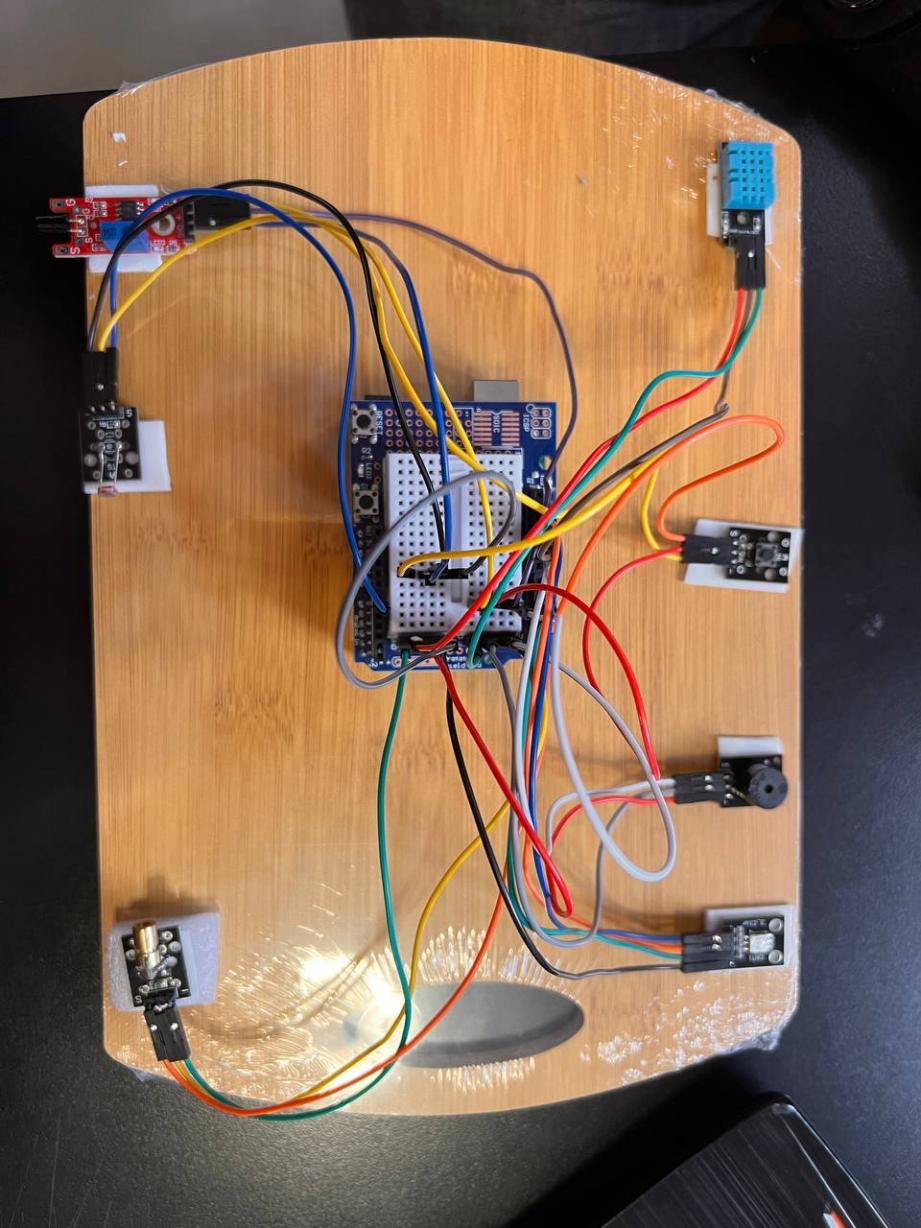
Lastly, a sensor for a fire alarm was built in that detects if there is a flame on site and the system also alerts by activating a buzzer sound and LED lights of another color. (Multiple colors are used for indicating other types of dangers to make sure which occurred.)

All the sensors can be shut down manually by a pushing button.

# Components Used:

Following are the components used in project:

1. Flame Sensor
2. Laser Emitter
3. Laser Resistor
4. Temperature and Humidity Sensor
5. LED
6. Button
7. Buzzer

****

2

1

3

4

**7**

**6**

**5**

**4**

**2**

**1**

**Fig. 1**

# Installation Strategy:

1. **Laser Module (Item 8):**

• Install the laser modules along the roof edge at regular intervals, at a distance of 6 feet from the edge.

• Ensure that the lasers create an uninterrupted perimeter line.

• Angle the lasers to cover the desired warning zone effectively.

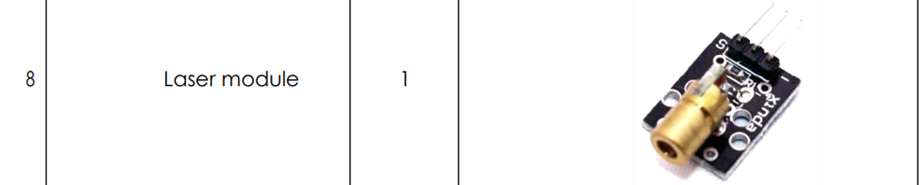


Fig. 2

1. **Photo-Interrupter Module (Item 12):**

• Install photo-interrupter modules parallel to the laser modules.

• Position them to detect interruptions in the light beams created by the lasers.

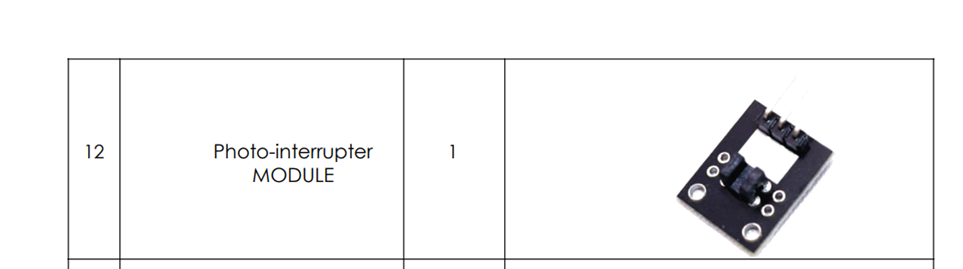


Fig. 3

1. **Passive Buzzer (Item 7):**

• Connect the passive buzzer to the Arduino Uno.

• Ensure the buzzer is audible from various points on the construction site to increase the chance of getting heard.

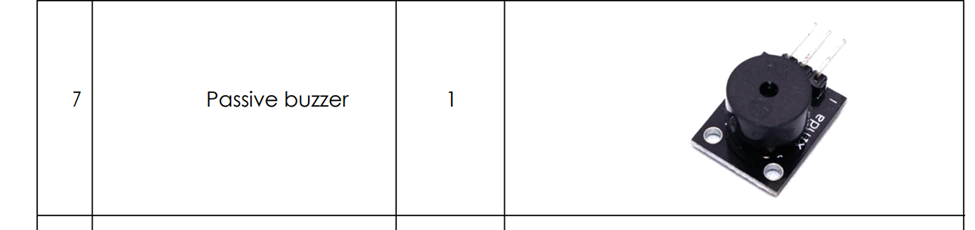


Fig. 4

1. **RGB LED Module (Item 10):**

• Optionally, install RGB LED modules along the roof edge.

• Use them for visual indications when the warning line is crossed.

• In our system: the fire alarm indicates a green light, exceeding the given humidity and temperature turning the light into blue, and in case of fall protection, the LED turns red.

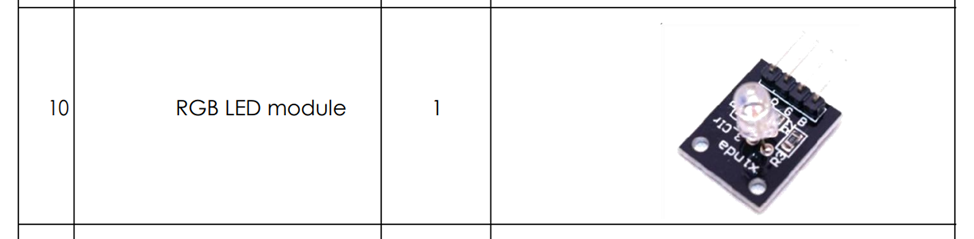


Fig. 5

1. **Connection Information:**

• Connect laser modules and photo-interrupter modules to digital output and input pins, respectively, on the Arduino Uno.

• Connect the passive buzzer and RGB LED modules to digital output pins on the Arduino Uno.

# System Operation:

1. **Flame Detection**

* The flame sensor monitors for the presence of flames.
* When flames are detected, the system triggers a visual and audible alarm.
* The LED blinks green, and the buzzer emits an intense sound.

1. **Laser Perimeter**

* The laser emitter creates a perimeter using a laser beam.
* The laser resistor assists in controlling the laser emission.

1. **Temperature Alert**

* The temperature and humidity sensor continuously monitors environmental conditions.
* If the temperature rises above a predefined threshold (e.g., 30°C), the system triggers an alarm.
* The LED blinks blue, and the buzzer emits an alert sound.

1. **Humidity Alert**

* If the humidity level goes above a predefined threshold (e.g. the 75% mentioned above), the system triggers an alarm.
* The LED blinks blue, and the buzzer emits an alert sound.

1. **Manual Override Button**

* A button is included to provide manual control and override functionality.
* Pressing the button stops the ongoing alarm triggered by flame detection, temperature, or humidity alerts. Before pushing the button, we have to make sure that the dangerous situation does not hold anymore.

# 5.1 Hardware Setup:

1. **Flame Sensor**

* Connect the flame sensor to digital pin 3.

1. **Laser Emitter and Resistor**

* Connect the laser emitter to digital pin 13.
* Connect the laser resistor to the laser emitter.

1. **Temperature and Humidity Sensor**

* Connect the temperature and humidity sensor to digital pin 12.

1. **LED**

* Connect the LED to digital pin 11.

1. **Button**

* Connect the button to digital pin 7.

1. **Buzzer**

* Connect the buzzer to digital pin 10.

# 5.2 Software Setup:

**i. Code**

* Upload the provided Arduino code to the microcontroller (Arduino board).
* The code continuously monitors sensor inputs and triggers appropriate alarms according to predefined conditions.

**ii. Testing and Calibration**

* Test the system in a controlled environment to ensure sensors are functioning correctly.
* Adjust sensor thresholds and alarm durations as needed for your specific conditions.

**iii. Safety Measures**

* Clearly define safety zones using the laser emitter.
* Train personnel on the system's response to different alerts.
* Implement emergency protocols and procedures.

# Conclusion:

The aim of the project work was to minimize the safety risks on construction sites by utilizing Arduino Uno technology and sensors.

The Construction Safety Monitoring System provides an effective solution for enhancing safety at construction sites. Regular testing, calibration, and personnel training are essential to ensure optimal performance and adherence to safety protocols. By installing such systems on construction sites and handling them with adequate competence, many accidents can be avoided and their occurrence probability will also decrease significantly.

# The code:

1. #include <DHT.h>
2. // Pin configurations
3. const int redPin = 12;          // Red pin of the RGB module
4. const int greenPin = 4;         // Green pin of the RGB module
5. const int bluePin = 11;         // Blue pin of the RGB module
6. const int buzzerPin = 8;        // Buzzer pin
7. const int laserEmitterPin = 1;  // Laser emitter pin
8. const int photoresistorPin = A0; // Photoresistor pin
9. const int flamePin = 13;        // Flame sensor pin
10. const int DHTPin = 10;          // DHT11 data pin
11. const int buttonPin = 3;        // Button pin
12. const int DHTType = DHT11;      // DHT11 sensor type
13. DHT dht(DHTPin, DHTType);
14. bool buttonPressed = false;
15. void setup() {
16. pinMode(redPin, OUTPUT);
17. pinMode(greenPin, OUTPUT);
18. pinMode(bluePin, OUTPUT);
19. pinMode(buzzerPin, OUTPUT);
20. pinMode(laserEmitterPin, OUTPUT);
21. pinMode(photoresistorPin, INPUT);
22. pinMode(flamePin, INPUT);
23. pinMode(buttonPin, INPUT\_PULLUP); // Button pin with internal pull-up resistor
24. Serial.begin(9600);
25. dht.begin();
26. }
27. void loop() {
28. // Check if the button is pressed
29. if (digitalRead(buttonPin) == LOW) {
30. // Button is pressed, turn off the specified pins
31. digitalWrite(redPin, LOW);
32. digitalWrite(greenPin, LOW);
33. digitalWrite(bluePin, LOW);
34. digitalWrite(buzzerPin, LOW);
35. buttonPressed = true;
36. } else {
37. // Button is not pressed, continue with your existing logic
38. if (buttonPressed) {
39. // If the button was previously pressed, reset the buttonPressed flag
40. buttonPressed = false;
41. }
43. delay(2000);  // 2 seconds delay between readings
44. float humidity = dht.readHumidity();
45. float temperature = dht.readTemperature();
46. Serial.print("Humidity: ");
47. Serial.print(humidity);
48. Serial.print(" %, Temperature: ");
49. Serial.print(temperature);
50. Serial.println(" °C");
51. bool redLight = (humidity > 50) || (temperature > 30);
52. bool blueLight = isLaserBlocked();
53. bool greenLight = (digitalRead(flamePin) == HIGH);
54. // Display RGB colors based on the conditions
55. displayRGBColors(redLight ? 255 : 0, greenLight ? 255 : 0, blueLight ? 255 : 0);
56. // Activate the buzzer if any condition is met
57. if (redLight || blueLight || greenLight) {
58. digitalWrite(buzzerPin, HIGH);
59. } else {
60. digitalWrite(buzzerPin, LOW);
61. }
62. }
63. }
64. void displayRGBColors(int redValue, int greenValue, int blueValue) {
65. // Display RGB colors on the RGB module
66. analogWrite(redPin, redValue);
67. analogWrite(greenPin, greenValue);
68. analogWrite(bluePin, blueValue);
69. }
70. bool isLaserBlocked() {
71. // Activate the laser emitter
72. digitalWrite(laserEmitterPin, HIGH);
73. delay(50);  // Allow time for the laser to stabilize
74. // Read the value from the photoresistor
75. int photoresistorValue = analogRead(photoresistorPin);
76. // Deactivate the laser emitter
77. digitalWrite(laserEmitterPin, LOW);
78. delay(50);  // Allow time for the laser to turn off (adjust this delay as needed)
79. // Check if the laser beam is blocked
80. return (photoresistorValue > 300);
81. }