



**COLLEGE OF ARTS AND SCIENCES (CAS)**

**SCHOOL OF COMPUTING (SOC)**

**BACHELOR OF SCIENCE (COMPUTER SCIENCES)**

**FIRST SEMESTER OF THE 2023/2024 SESSION**

**SKIH3113 SENSOR-BASED SYSTEM (A)**

**MID-TERM TEST**

**INDIVIDUAL**

PREPARED BY:

**NAME: SUCHIRA A/P SUMON**

**MATRIC NUMBER: 288316**

PREPARED FOR:

**LECTURER NAME: ENCIK AHMAD HANIS BIN MOHD SHABLI**

DUE DATE:

**15TH JUNE 2024**

## **1.0 Introduction**

### **1.1 Components:**

ESP8266: A Wi-Fi enabled microcontroller used for data processing and communication.

MQ135 Air Quality Sensor: A sensor capable of detecting a wide range of gases including ammonia, nitrogen oxides, alcohol, benzene, smoke, and carbon dioxide.

IR (Infrared) Sensor: A sensor that detects the presence of objects or motion by using infrared light.

### **1.2 Functionality:**

Air Quality Monitoring: The MQ135 sensor continuously measures the concentration of various gases in the air. It provides an analog output that is read by the ESP8266. The microcontroller processes this data to determine the air quality index or the presence of specific harmful gases.

Object or Motion Detection: The IR sensor detects the presence of objects or motion within its range. This can be used to monitor occupancy in a room or detect movement.

### **1.3 Applications:**

Environmental Monitoring: Can be deployed in various environments to monitor pollution levels.

Security Systems: Using the IR sensor for intrusion detection and integrating with home security systems.

## 2.0 Project Design

Figure 1: Fritzing prototype of Project Component

Figure2: Connection of the Project Component

## 3.0 Arduino Code

```
#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>


const char* ssid = "Lenovo Tab";

const char* password = "abcde123";

const char* serverUrl = "http://192.168.39.60/MidTerm/data.php"; //
Replace with your server URL


#define IR_SENSOR_PIN D1

#define MQ135_AO_PIN A0

#define MQ135_DO_PIN D2

#define LED_PIN D5


void setup() {

    Serial.begin(115200);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {

        delay(500);
```

```

        Serial.print(".");
    }

    Serial.println("");

    Serial.println("WiFi connected");

    Serial.print("IP address: ");

    Serial.println(WiFi.localIP());


    pinMode(IR_SENSOR_PIN, INPUT);

    pinMode(MQ135_AO_PIN, INPUT);

    pinMode(MQ135_DO_PIN, INPUT);

    pinMode(LED_PIN, OUTPUT);
}


void loop() {

    // Read IR sensor value

    int irValue = digitalRead(IR_SENSOR_PIN);

    irValue = !irValue; // Invert irValue (0 becomes 1, and 1 becomes 0)


    // Read MQ135 sensor values

    int mq135AnalogValue = analogRead(MQ135_AO_PIN);

    int mq135DigitalValue = digitalRead(MQ135_DO_PIN);


    // Prepare data to send to PHP script

    String postData = "ir_value=" + String(irValue) +

        "&mq135_analog_value=" + String(mq135AnalogValue) +

```

```

        "&mq135_digital_value=" +
String(mq135DigitalValue);

// Send HTTP POST request to PHP script

HTTPClient http;

WiFiClient wclient;

http.begin(wclient, serverUrl); // Replace with your PHP script URL

http.addHeader("Content-Type", "application/x-www-form-urlencoded");

int httpResponseCode = http.POST(postData); // Send the actual POST
request

if (httpResponseCode > 0) {

    Serial.print("HTTP Response code: ");

    Serial.println(httpResponseCode);

    String response = http.getString(); // Get the response from the
server

    Serial.println(response); // Print the response

} else {

    Serial.print("Error code: ");

    Serial.println(httpResponseCode);

}

http.end(); // Free resources

delay(10000); // Delay between readings

}

```

## 4.0 Interfaces

data.php

```
<?php
```

```
$servername = "localhost";
```

```
// REPLACE with your Database name
```

```
$dbname = "environmentaldata";
```

```
// REPLACE with Database user
```

```
$username = "root";
```

```
// REPLACE with Database user password
```

```
$password = "";
```

```
// $ir_value = $mq135_analog_value = $mq135_digital_value = $timestamp = "";
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Check connection
```

```
if ($conn->connect_error) {
```

```
    die("Connection failed: " . $conn->connect_error);
```

```
}
```

```
if (isset($_POST["ir_value"]) && isset($_POST["mq135_analog_value"]) &&
```

```
isset($_POST["mq135_digital_value"])) {
```

```
    $ir_value = $_POST["ir_value"];
```

```
    $mq135_analog_value = $_POST["mq135_analog_value"];
```

```
    $mq135_digital_value = $_POST["mq135_digital_value"];
```

```
    $sql = "INSERT INTO sensordata (ir_value, mq135_analog_value,  
mq135_digital_value) VALUES (" . $ir_value . "," . $mq135_analog_value . "," .  
$mq135_digital_value . ")";
```

```
    // Execute the SQL query and check if it was successful
```

```
    if ($conn->query($sql) === TRUE) {
```

```
        echo "Values inserted in MySQL database table.";
```

```
    } else {
```

```
        // If the query execution failed, print the error
```

```
        echo "Error: " . $sql . "<br>" . $conn->error;
```

```
    }
```

```
}
```

```
echo json_encode($data);
```

```
?>
```

index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<title>Sensor Data</title>
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 20px;
    background-color: #f4f4f4;
  }
  table {
    width: 100%;
    border-collapse: collapse;
    margin-bottom: 20px;
  }
  table, th, td {
    border: 1px solid #ddd;
  }
  th, td {
    padding: 12px;
    text-align: left;
  }
  th {
    background-color: #0074D9; /* Blue color */
    color: white;
  }
  tr:nth-child(even) {
    background-color: #f2f2f2;
  }
  /* Apply color coding based on mq135_analog_value */
  td[data-range="low"] {
    color: black;
  }
  td[data-range="medium"] {
    color: orange;
  }
  td[data-range="high"] {
    color: red;
  }
</style>
</head>
<body>
  <h1>Sensor Data</h1>
  <div>
    <label for="ir-filter">IR Value:</label>
    <select id="ir-filter">
      <option value="all">All</option>
      <option value="1">1</option>
      <option value="0">0</option>
    </select>
    <label for="mq135-filter">MQ135 Analog Value:</label>
    <select id="mq135-filter">
      <option value="all">All</option>

```

```

        <option value="low">Low (&lt; 300)</option>
        <option value="medium">Moderate (300 - 600)</option>
        <option value="high">High (&gt; 600)</option>
    </select>
</div>
<table>
    <thead>
        <tr>
            <th>IR Value</th>
            <th>MQ135 Analog Value</th>
            <th>MQ135 Digital Value</th>
            <th>Timestamp</th>
        </tr>
    </thead>
    <tbody id="data-table">
    </tbody>
</table>
<script>
    document.addEventListener("DOMContentLoaded", function() {
        const irFilter = document.getElementById('ir-filter');
        const mq135Filter = document.getElementById('mq135-filter');
        let sensorData = [];

        fetch('fetchdata.php')
            .then(response => {
                if (!response.ok) {
                    throw new Error('Network response was not ok');
                }
                return response.json();
            })
            .then(data => {
                sensorData = data;
                displayData(sensorData);
            })
            .catch(error => console.error('Error fetching data:', error));

        function displayData(data) {
            const tableBody = document.getElementById('data-table');
            tableBody.innerHTML = "";
            data.forEach(row => {
                const tr = document.createElement('tr');
                let range = "";
                if (row.mq135_analog_value < 300) {
                    range = 'low';
                } else if (row.mq135_analog_value >= 300 && row.mq135_analog_value
<= 600) {
                    range = 'medium';
                } else {
                    range = 'high';
                }
            }

```



```

        tr.innerHTML = `
            <td>${row.ir_value}</td>
            <td data-range="${range}">${row.mq135_analog_value}</td>
            <td>${row.mq135_digital_value}</td>
            <td>${row.timestamp}</td>
        `;
        tableBody.appendChild(tr);
    });
}

function applyFilters() {
    const irValue = irFilter.value;
    const mq135Value = mq135Filter.value;

    const filteredData = sensorData.filter(row => {
        const irMatch = irValue === 'all' || row.ir_value.toString() === irValue;
        let mq135Match = true;
        if (mq135Value === 'low') {
            mq135Match = row.mq135_analog_value < 300;
        } else if (mq135Value === 'medium') {
            mq135Match = row.mq135_analog_value >= 300 &&
row.mq135_analog_value <= 600;
        } else if (mq135Value === 'high') {
            mq135Match = row.mq135_analog_value > 600;
        }
        return irMatch && mq135Match;
    });

    displayData(filteredData);
}

irFilter.addEventListener('change', applyFilters);
mq135Filter.addEventListener('change', applyFilters);
});
</script>
</body>
</html>

```

```

fetchdata.php
<?php
    // Database connection parameters
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "environmentaldata";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);

```

```
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// SQL query to fetch data
$sql = "SELECT * FROM sensordata";
$result = $conn->query($sql);

$data = array();

if ($result->num_rows > 0) {
    // Output data of each row
    while($row = $result->fetch_assoc()) {
        $data[] = $row;
    }
} else {
    echo "0 results";
}

echo json_encode($data);

$conn->close();
?>
```

## 5.0 Database

Server: 127.0.0.1 » Database: environmentaldata » Table: sensordata

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 ir_value	int(11)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	3 mq135_analog_value	int(11)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	4 mq135_digital_value	int(11)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	5 timestamp	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

## 7.0 Web interface

localhost / 127.0.0.1 / environm x Sensor Data

localhost/midTerm/index.html

Sensor Data

IR Value: 0 MQ135 Analog Value: Moderate (300 - 600)

IR Value	MQ135 Analog Value	MQ135 Digital Value	Timestamp
0	401	1	2024-06-14 23:20:47
0	400	1	2024-06-14 23:20:57
0	384	1	2024-06-14 23:21:07
0	398	1	2024-06-14 23:21:17
0	396	1	2024-06-14 23:21:27
0	394	1	2024-06-14 23:21:37
0	394	1	2024-06-14 23:21:47
0	393	1	2024-06-14 23:21:57
0	374	1	2024-06-14 23:32:29

- IR Value: Represents the infrared (IR) value. 0 represents no motion while 1 represents motion.
- MQ135 Analog Value: Indicates the analog value from an MQ135 gas sensor.
- MQ135 Digital Value: This column is not explicitly labeled, but it likely corresponds to the digital output from the MQ135 sensor (also '1' in this case).
- Timestamp: Shows the date and time when the sensor readings were recorded. The timestamps range from 2024-06-14 23:20:47 to 2024-06-14 23:32:29.

## 7.0 Challenges/Risks

One of the primary challenges in this project was establishing a reliable database connection to store and retrieve sensor data efficiently. Another significant hurdle was resolving the server URL errors that occurred while attempting to pass data from the Arduino to the cloud server, which disrupted the continuous data flow.

## 8.0 Appendix

Github link: [https://github.com/SuchiraSumon/MIDTERM\\_SensorBased.git](https://github.com/SuchiraSumon/MIDTERM_SensorBased.git)

Presentation video link:

[https://www.canva.com/design/DAGllieOAgY/mP2AKjVGzTRpyCntR9DFfg/view?utm\\_content=DAGllieOAgY&utm\\_campaign=share\\_your\\_design&utm\\_medium=link&utm\\_source=shareyourdesignpanel](https://www.canva.com/design/DAGllieOAgY/mP2AKjVGzTRpyCntR9DFfg/view?utm_content=DAGllieOAgY&utm_campaign=share_your_design&utm_medium=link&utm_source=shareyourdesignpanel)  
<https://youtu.be/BJ5i4AHi2Ek>