Environmental Monitoring using IoT:

Problem Definition:

The rapid degradation of our environment due to various factors such as climate change, pollution, and natural disasters poses a significant threat to our planet and human well-being. To address these challenges, there is an increasing need for effective environmental monitoring systems. Internet of Things (IoT) technology offers a promising solution by enabling the collection of real-time data from various sensors and devices to monitor and manage environmental conditions. This document focuses on defining the problem and exploring the application of design thinking principles to create a robust IoT-based environmental monitoring system. The primary problem at hand is the lack of a comprehensive and scalable environmental monitoring system that can provide accurate, real-time data on various environmental parameters. The current methods are often fragmented, relying on manual data collection and periodic analysis, which limits our ability to respond promptly to environmental issues. Key challenges include:

- Data Inaccuracy: Existing environmental monitoring systems often suffer from data inaccuracies due to limited sensor coverage, calibration issues, and outdated technology. This leads to unreliable information for decision-making.
- Limited Scalability: Many current monitoring systems are localized and lack the scalability to cover large geographic areas or diverse environmental parameters. This restricts our ability to monitor and manage environmental conditions effectively.
- Delayed Response: Traditional monitoring approaches involve time-consuming data collection and analysis, delaying the response to critical environmental events such as wildfires, air quality deterioration, or water contamination

DESIGN THINKING

- Empathize: The first step is to understand the needs and pain points of various stakeholders, including environmental agencies, researchers, policymakers, and the general public. Conduct surveys, interviews, and workshops to gain insights into their requirements.
- Define: Clearly define the problem based on the information gathered during the empathize stage. Prioritize the most critical environmental parameters to monitor and identify the key challenges that need to be addressed.
- Ideate: Encourage brainstorming sessions with a diverse group of participants to generate innovative ideas for an IoT-based environmental monitoring system. Consider technology, sensors, data collection methods, and data visualization techniques.
- Prototype: Create prototypes of the proposed monitoring system to visualize how it would work in practice. Use mock-ups, simulations, and basic sensor setups to test the feasibility of the concept.
- Test: Conduct pilot tests and gather feedback from users and stakeholders. Evaluate the prototype's effectiveness in addressing the identified problems and making improvements based on user input.
- Implement: Develop a full-scale IoT-based environmental monitoring system based on the refined prototype. Utilize scalable IoT platforms and advanced sensors to ensure accurate data collection.
- Monitor and Iterate: Continuously monitor the system's performance, data accuracy, and user satisfaction. Make regular updates and improvements to address emerging environmental challenges and evolving user needs.
- Multi-Parameter Sensors: Incorporate sensors capable of measuring multiple environmental parameters such as air quality, temperature, humidity, water quality, and soil conditions.
- Real-time Data Transmission: Ensure that data from sensors are transmitted in real-time to a central server or cloud-based platform for immediate analysis and action.
- Data Visualization: Create user-friendly dashboards and mobile applications that provide easy access to real-time environmental data, making it understandable to both experts and the general public.

System hardware design:

- The system hardware includes: main controller, power supply and step-down circuit, fan relay control circuit, NB-IoT module circuit, Usart-GPU serial touch LCD screen circuit, sensor circuit.
- The system power supply is completed by a 12V-3A power supply and a 5V output step-down circuit. The sensor circuit collects and displays environmental information by the main controller, each sensor and the touch screen.
- The connection between the entire system and the cloud platform is done by NB-IoT The module is complete.
- The microcontroller will remotely report the converted environmental data to the cloud platform through the narrowband IoT communication module, and the user can view the field site environmental data in real time through the cloud platform.

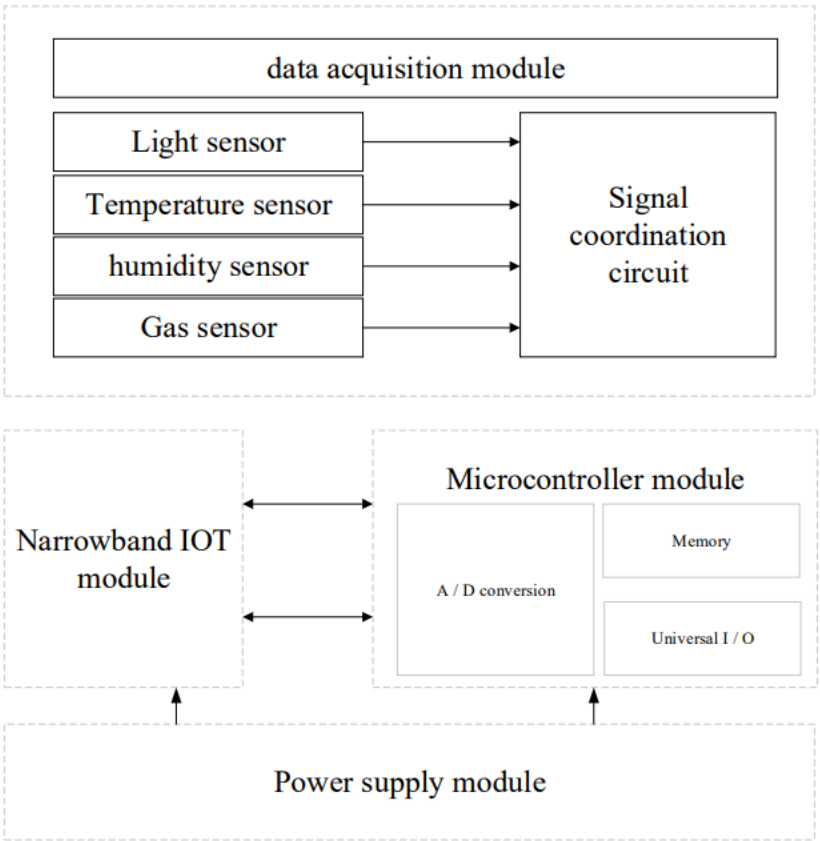


Fig. 2 Hardware composition diagram

System software design:

The main control program of the system includes the main controller to collect, display and report data on the environment's light intensity, temperature, humidity, and gas concentration. After the system is started, reset the device, reset the peripheral interface of the micro controller various sensors in the system, and the narrowband IoT communication module Then, the system checks whether the narrowband IoT communication module is successfully connected to the operator's network.

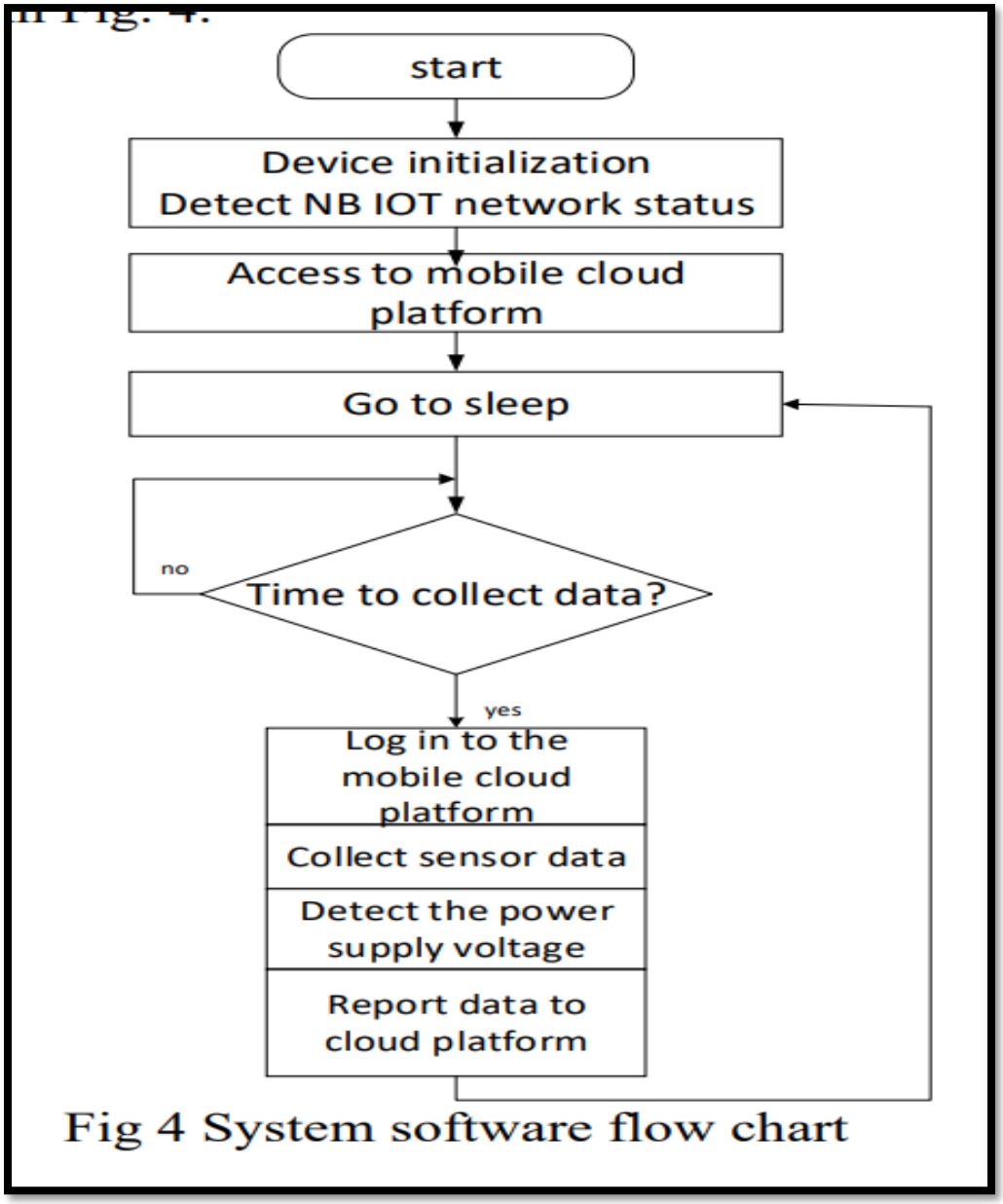
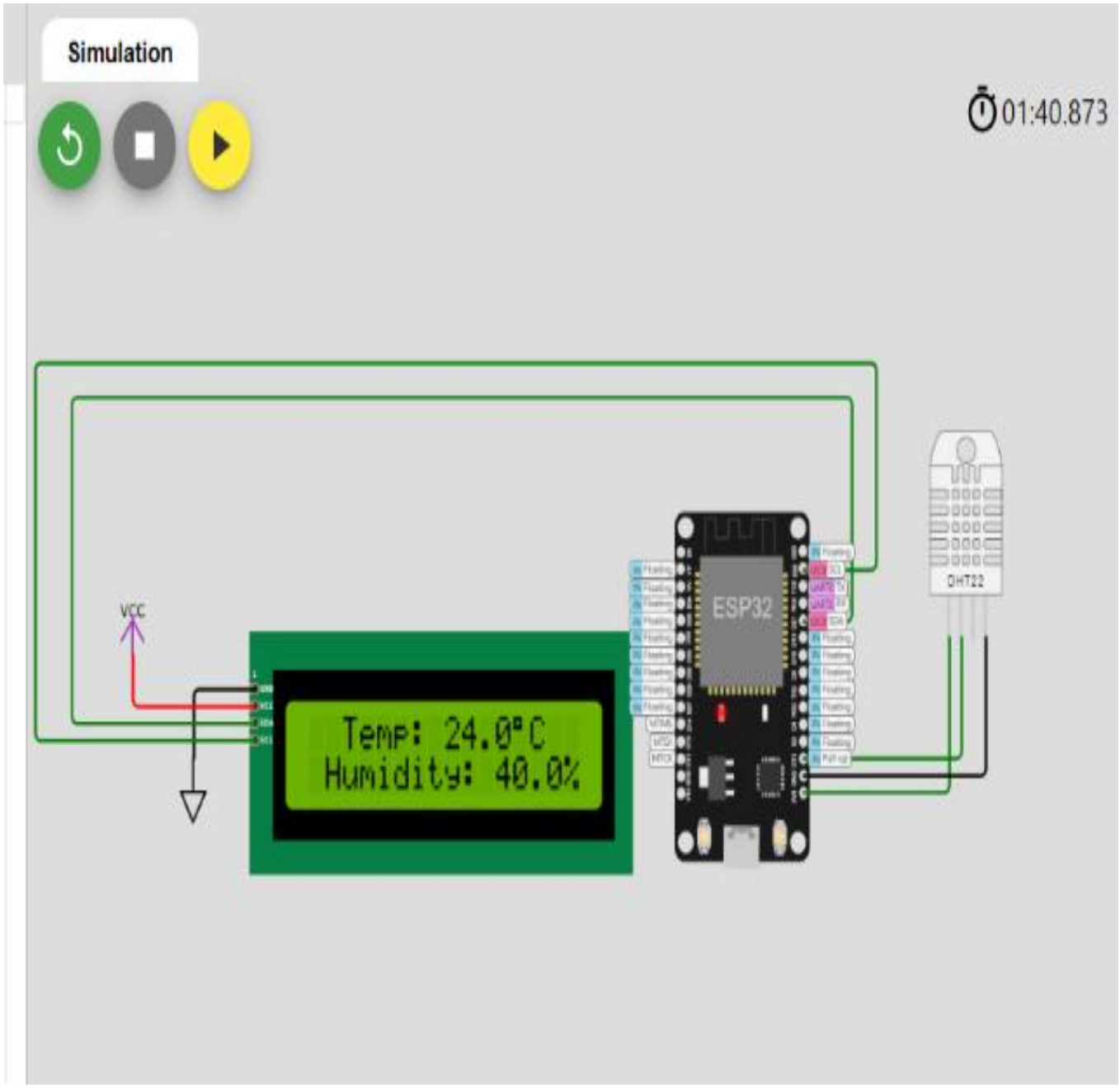


Fig 4 System software flow chart

INTERFACING DHT22 SENSOR WITH ESP32 IN WOKWI SIMULATOR

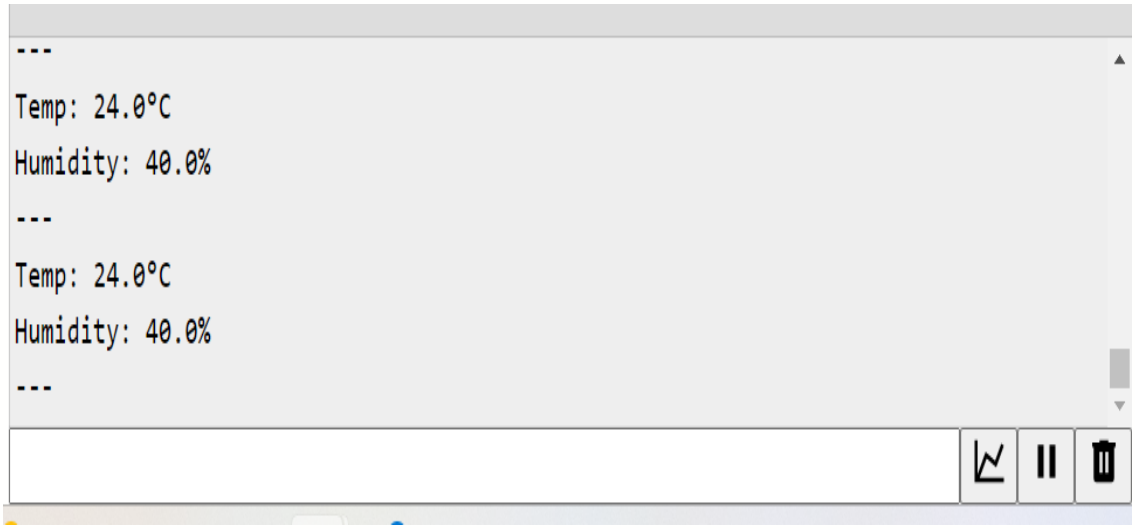
INTERFACING DIAGRAM:



CODE:

```
#include "DHTesp.h"
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR    0x27
#define LCD_COLUMNS 16
#define LCD_LINES   2

const int DHT_PIN = 15;

DHTesp dhtSensor;

LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLUMNS, LCD_LINES);

void setup() {

  Serial.begin(115200);
  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
  lcd.init();
  lcd.backlight();

}

void loop() {

  TempAndHumidity  data = dhtSensor.getTempAndHumidity();
  Serial.println("Temp: " + String(data.temperature, 1) + "°C");

  Serial.println("Humidity: " + String(data.humidity, 1) + "%");
  Serial.println("---");

  lcd.setCursor(0, 0);
  lcd.print("  Temp: " + String(data.temperature, 1) + "\xDF"+"C  ");
  lcd.setCursor(0, 1);
  lcd.print(" Humidity: " + String(data.humidity, 1) + "% ");
  lcd.print("Wokwi Online IoT");

  delay(1000);
}
```

OUTPUT:



SIMULATION LINK:

https://wokwi.com/projects/378936628540765185

TO DEVELOP THE WEBSITE PLATFORM:

```
const express = require('express');
const app = express();
const port = 3000;

// Serve static files (HTML, CSS, JavaScript)
app.use(express.static('public'));

// Endpoint to provide fixed temperature and humidity data
app.get('/api/data', (req, res) => {
    const data = {
        temperature: 24.0,
        humidity: 40.0,
    };
    res.json(data);
```

```
});

app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
<!DOCTYPE html>
<html>
<head>
  <title>Real-Time Environment Data</title>
</head>
<body>
  <h1>Real-Time Environment Data</h1>
  <div>
    <p>Temperature: <span id="temperature">Loading...</span></p>
    <p>Humidity: <span id="humidity">Loading...</span></p>
  </div>
  <script>
    const temperatureElement = document.getElementById('temperature');
    const humidityElement = document.getElementById('humidity');

    // Function to fetch real-time data and update the web page
    function updateData() {
      fetch('/api/data')
        .then(response => response.json())
        .then(data => {
          temperatureElement.textContent = data.temperature + ' °C';
          humidityElement.textContent = data.humidity + '%';
        })
        .catch(error => {
          console.error('Failed to fetch data:', error);
```

```
            });
        }

        // Periodically update data (e.g., every 5 seconds)
        setInterval(updateData, 5000);
    </script>
</body>
</html>
node server.js
```

OUTPUT:

PROJECT OBJECTIVES:

- Empathize: The first step is to understand the needs and pain points of various stakeholders, including environmental agencies, researchers, policymakers, and the general public. Conduct surveys, interviews, and workshops to gain insights into their requirements.
- Define: Clearly define the problem based on the information gathered during the empathize stage. Prioritize the most critical environmental parameters to monitor and identify the key challenges that need to be addressed.
- Ideate: Encourage brainstorming sessions with a diverse group of participants to generate innovative ideas for an IoT-based environmental monitoring system. Consider technology, sensors, data collection methods, and data visualization techniques.
- Prototype: Create prototypes of the proposed monitoring system to visualize how it would work in practice. Use mock-ups, simulations, and basic sensor setups to test the feasibility of the concept.
- Test: Conduct pilot tests and gather feedback from users and stakeholders. Evaluate the prototype's effectiveness in addressing the identified problems and making improvements based on user input.
- Implement: Develop a full-scale IoT-based environmental monitoring system based on the refined prototype. Utilize scalable IoT platforms and advanced sensors to ensure accurate data collection.
- Monitor and Iterate: Continuously monitor the system's performance, data accuracy, and user satisfaction. Make regular updates and improvements to address emerging environmental challenges and evolving user needs.
- Multi-Parameter Sensors: Incorporate sensors capable of measuring multiple environmental parameters such as air quality, temperature, humidity, water quality, and soil conditions.
- Real-time Data Transmission: Ensure that data from sensors are transmitted in real-time to a central server or cloud-based platform for immediate analysis and action.
- Data Visualization: Create user-friendly dashboards and mobile applications that provide easy access to real-time environmental data, making it understandable to both experts and the general public.