

**PRAKTIKUM SIMULASI SISTEM MONITORING SUHU,  
KELEMBAPAN, DAN KONTROL LED BERBASIS  
ESP32 DENGAN PLATFORM WOKWI SIMULATOR  
DAN BLYNK**



Ifadah Aulia Muhti Sinaga

233140707111034

**D III TEKNOLOGI INFORMASI  
FAKULTAS VOKASI  
UNIVERSITAS BRAWIJAYA**

## Abstrak

Perkembangan teknologi Internet of Things (IoT) membuka peluang besar dalam pemantauan data lingkungan secara waktu nyata. Praktikum ini menyimulasikan sistem pemantauan cuaca berbasis ESP32 yang memperoleh informasi dari API OpenWeather, lalu menampilkannya pada LCD 16x2 menggunakan komunikasi I2C. Data yang disajikan meliputi suhu, suhu terasa, kelembapan, curah hujan, kecepatan dan arah angin, tekanan udara, tutupan awan, serta waktu matahari terbit dan terbenam. Sistem menyediakan fitur navigasi data menggunakan tombol "Next" dan "Prev" untuk menelusuri informasi secara bergantian. Seluruh proses dilakukan melalui Wokwi Simulator tanpa memerlukan perangkat keras fisik. Berdasarkan simulasi yang dilakukan, sistem terbukti dapat menampilkan data cuaca secara responsif, terstruktur, dan interaktif.

Kata Kunci: IoT, ESP32, LCD I2C, API OpenWeather, Wokwi

## Abstract

The advancement of Internet of Things (IoT) technology has opened new opportunities for real-time environmental data monitoring. This practicum simulates a weather monitoring system using the ESP32 microcontroller, which retrieves weather data from the OpenWeather API and displays it on a 16x2 LCD via I2C communication. The displayed parameters include temperature, feels-like temperature, humidity, precipitation, wind speed and direction, air pressure, cloud coverage, and sunrise and sunset times. Users can navigate through this information using Next and Prev buttons to scroll between data items. The entire system is simulated using the Wokwi platform, eliminating the need for physical hardware. Simulation results show that the system effectively and responsively displays weather data in an organized and interactive manner.

**Keywords:** IoT, ESP32, LCD I2C, OpenWeather API, Wokwi

## 1. Pendahuluan

### 1.1 Latar Belakang

Internet of Things (IoT) memungkinkan perangkat fisik berkomunikasi secara langsung melalui jaringan internet, termasuk dalam bidang pemantauan kondisi cuaca. Dengan memanfaatkan layanan API publik seperti OpenWeather, pengguna dapat memperoleh informasi cuaca secara langsung dan terintegrasi. Dalam praktik ini, mikrokontroler ESP32 digunakan sebagai pusat pengolahan data yang terhubung ke internet, mengambil data dari API, dan menampilkannya pada LCD 16x2. Navigasi antar informasi dilakukan menggunakan tombol untuk menggulir (scroll) data, sehingga pengguna dapat melihat seluruh parameter cuaca secara bergantian.

### 1.2 Tujuan Penelitian

Tujuan dari praktikum ini antara lain:

- Menjelaskan konsep dasar integrasi API dengan perangkat IoT.
- Mempelajari proses pengambilan data cuaca dari OpenWeather API.
- Mengimplementasikan tampilan data ke dalam modul LCD 16x2 menggunakan protokol I2C.
- Menambahkan fungsi navigasi data dengan tombol Next dan Prev.
- Melakukan simulasi sistem secara virtual menggunakan platform Wokwi.

## 2. Metodologi

### 2.1 Peralatan dan Komponen

Perangkat:

- Laptop atau komputer
- Wokwi Simulator
- Visual Studio Code dengan PlatformIO

Komponen Virtual:

- ESP32 Devkit V1
- LCD 16x2 (dengan modul I2C)
- Dua tombol pushbutton (untuk navigasi)
- Jaringan Wi-Fi Wokwi Guest

### 2.1 Langkah Implementasi

Langkah-langkah pelaksanaan simulasi adalah sebagai berikut:

#### Simulasi pada Wokwi Web:

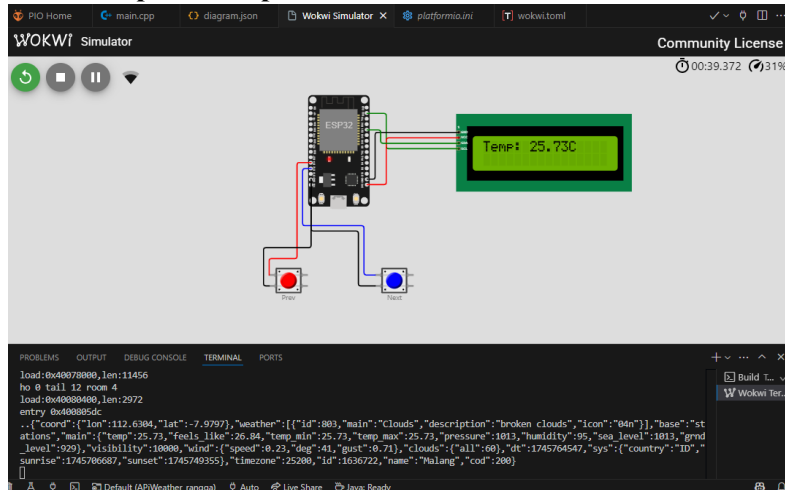
1. Akses situs [wokwi.com](https://wokwi.com) dan buat proyek baru menggunakan board ESP32.
2. Tambahkan komponen virtual: ESP32, LCD I2C, tombol Next (D12), dan tombol Prev (D14).
3. Lakukan koneksi pin sesuai kebutuhan (SDA ke D21, SCL ke D22, tombol ke GND dan pin digital).
4. Tulis dan unggah kode ke dalam proyek, kemudian jalankan simulasi.
5. Pastikan LCD berhasil menampilkan data dan tombol berfungsi sebagai navigasi.

#### Simulasi pada VSCode dengan PlatformIO :

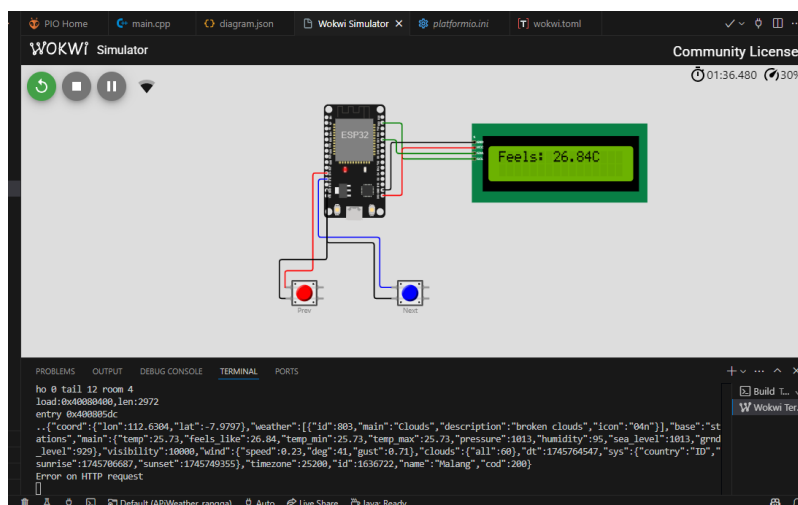
1. Buat proyek baru dengan board ESP32 dan framework Arduino.
2. Salin diagram.json dari Wokwi ke dalam folder proyek.
3. Masukkan program ke dalam src/main.cpp, dan tambahkan konfigurasi wokwi.toml.
4. Kompilasi proyek, lalu jalankan simulasi langsung dari VSCode.
5. Amati tampilan data pada LCD serta respon dari tombol navigasi.

### 3. Hasil dan Pembahasan

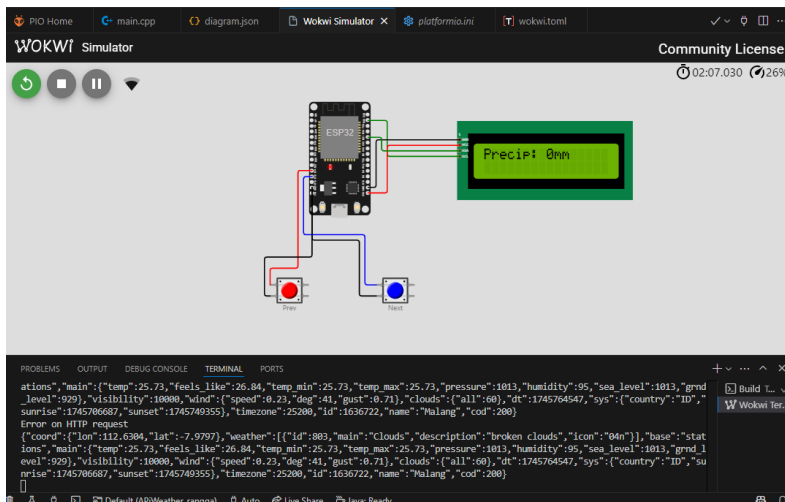
- Hasil Tampilan Temperature



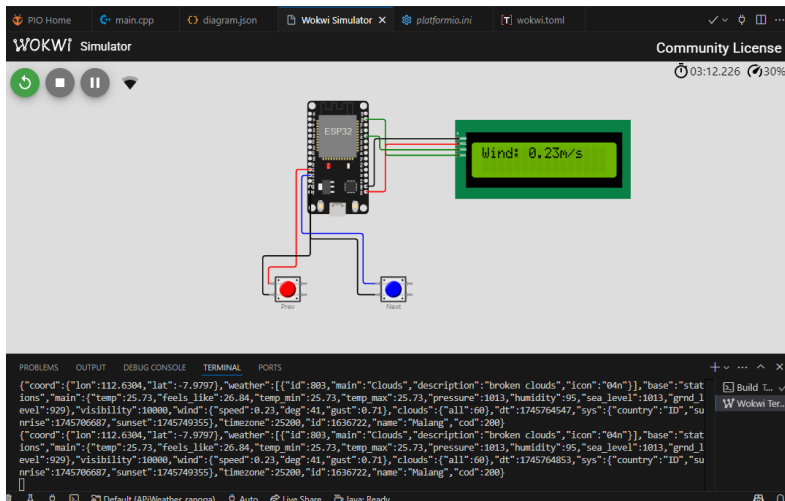
- Hasil Tampilan Feels Like



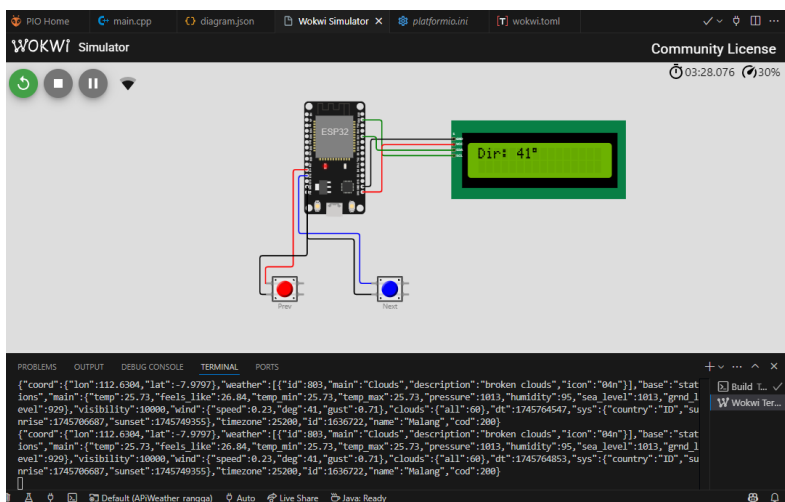
- Hasil Tampilan Precipitation



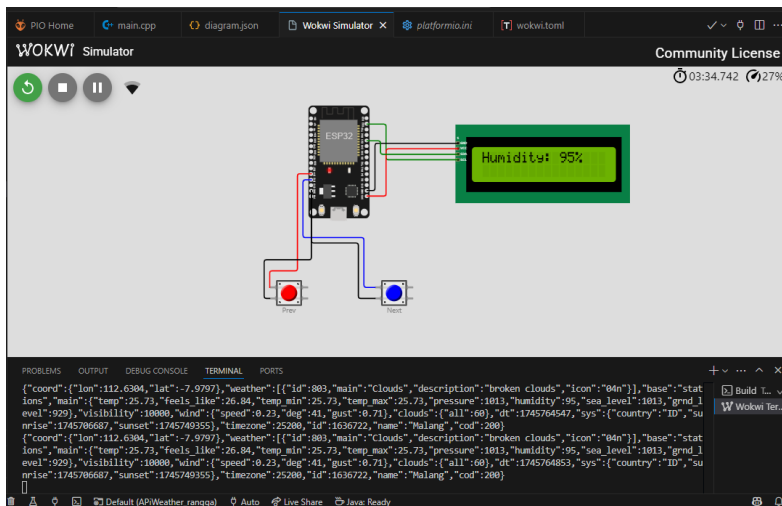
- Hasil Tampilan Wind Speed



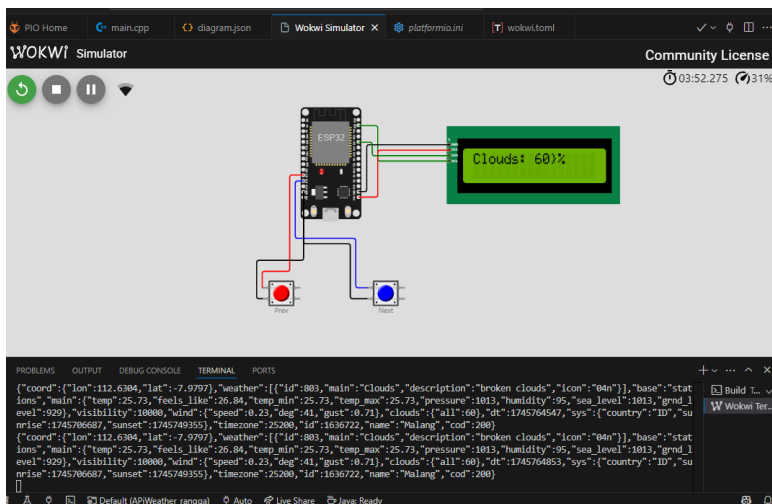
- Hasil Tampilan Wind Direction



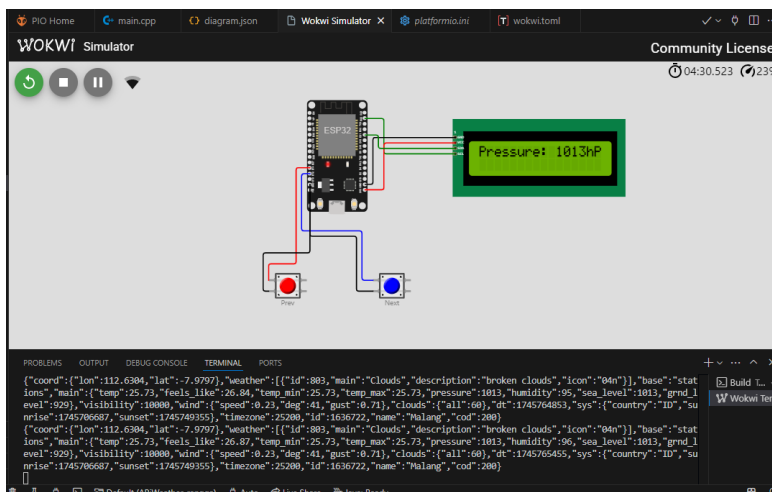
- Hasil Tampilan Humadity



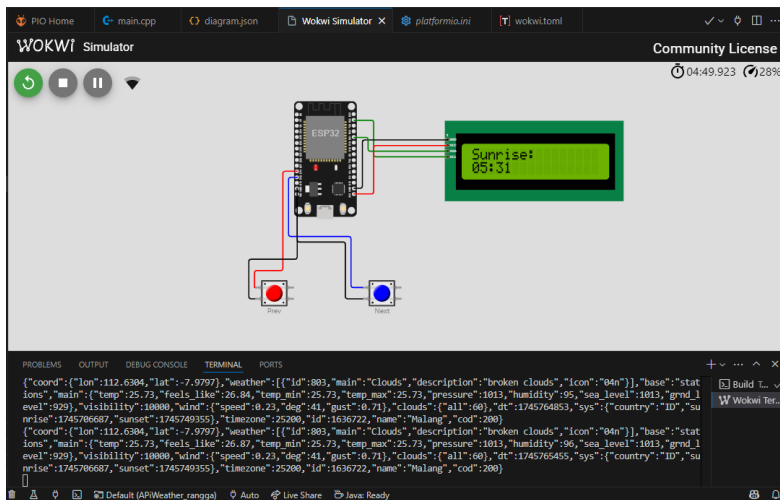
- Hasil Tampilan Clouds



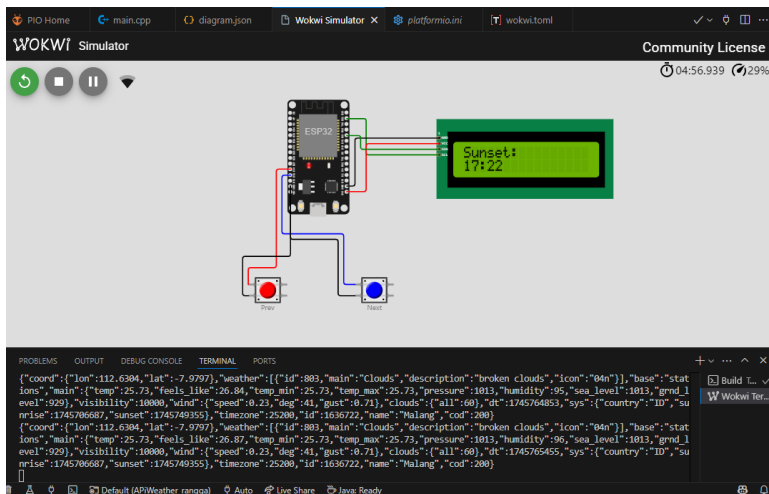
- Hasil Tampilan Pressure



- Hasil Tampilan Sunrise



## • Hasil Tampilan Sunset



## • Pembahasan

Sistem yang dirancang menunjukkan bahwa ESP32 mampu membaca data dari internet dan menampilkannya secara lokal. Fungsi pengambilan dan parsing data dilakukan tanpa menggunakan library tambahan seperti ArduinoJson, melainkan memakai fungsi manual seperti `getValue()` dan `getOptionalValue()` sehingga program menjadi lebih ringan.

Karena simulasi dilakukan di lingkungan virtual, data yang diterima bersifat statis (dummy). Meski begitu, sistem mampu menunjukkan alur kerja yang mendekati kondisi nyata, terutama jika diterapkan pada perangkat fisik dengan koneksi internet aktual.

Keseluruhan komponen seperti koneksi Wi-Fi, HTTP client, parsing JSON, serta antarmuka tombol telah berfungsi dengan baik sesuai tujuan praktikum.

## 4. Kode Program

### 4.1 main.cpp

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "Wokwi-GUEST";
const char* password = "";
String apiKey = "0c854b30810ba8e3e1f55b2eeabe6da6";
String city = "Malang";
String units = "metric";
String server = "http://api.openweathermap.org/data/2.5/weather?q=" + city +
"&units=" + units + "&appid=" + apiKey;

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int buttonNext = 12;
const int buttonPrev = 14;

int scrollIndex = 0;
unsigned long lastUpdate = 0;
unsigned long updateInterval = 60000;

// Data weather
String temp = "N/A";
String feels_like = "N/A";
String precipitation = "0"; // Default 0 mm
String windSpeed = "N/A";
String windDirection = "N/A";
String humidity = "N/A";
String clouds = "N/A";
String pressure = "N/A";
String sunrise = "N/A";
String sunset = "N/A";

// Function prototypes
void fetchWeather();
void displayData(int index);
String getValue(String data, String key);
String getOptionalValue(String data, String key);
String formatTime(String unixTimeStr);

void setup() {
  Serial.begin(115200);

  lcd.init();
```



```

    lcd.backlight();

    pinMode(buttonNext, INPUT_PULLUP);
    pinMode(buttonPrev, INPUT_PULLUP);

    // Tampilan awal nama pembuat
    lcd.setCursor(0, 0);
    lcd.print("Created by");
    lcd.setCursor(0, 1);
    lcd.print("Ifadah");
    delay(2000);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Ipadd");
    delay(2000);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Connecting...");

    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Connected!");
    delay(1000);
    lcd.clear();

    fetchWeather(); // Fetch pertama
    displayData(scrollIndex);
}

void loop() {
    unsigned long currentMillis = millis();

    // Update data tiap 60 detik
    if (currentMillis - lastUpdate > updateInterval) {
        lastUpdate = currentMillis;
        fetchWeather();
        displayData(scrollIndex);
    }

    // Tombol Next

```

```

    if (digitalRead(buttonNext) == LOW) {
        scrollIndex++;
        if (scrollIndex > 9) scrollIndex = 0;
        displayData(scrollIndex);
        delay(300); // Debounce
    }

    // Tombol Prev
    if (digitalRead(buttonPrev) == LOW) {
        scrollIndex--;
        if (scrollIndex < 0) scrollIndex = 9;
        displayData(scrollIndex);
        delay(300); // Debounce
    }
}

void fetchWeather() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(server);
        int httpCode = http.GET();

        if (httpCode > 0) {
            String payload = http.getString();
            Serial.println(payload);

            temp = getValue(payload, "temp");
            feels_like = getValue(payload, "feels_like");
            humidity = getValue(payload, "humidity");
            windSpeed = getValue(payload, "speed");
            windDirection = getValue(payload, "deg");
            pressure = getValue(payload, "pressure");
            clouds = getValue(payload, "all");
            precipitation = getOptionalValue(payload, "\"rain\":{\\\"1h\\\":}");
            sunrise = formatTime(getValue(payload, "sunrise"));
            sunset = formatTime(getValue(payload, "sunset"));

        } else {
            Serial.println("Error on HTTP request");
        }
        http.end();
    }
}

void displayData(int index) {
    lcd.clear();
    lcd.setCursor(0, 0);

```

```

switch(index) {
    case 0:
        lcd.print("Temp: " + temp + "C");
        break;
    case 1:
        lcd.print("Feels: " + feels_like + "C");
        break;
    case 2:
        lcd.print("Precip: " + precipitation + "mm");
        break;
    case 3:
        lcd.print("Wind: " + windSpeed + "m/s");
        break;
    case 4:
        lcd.print("Dir: " + windDirection + (char)223);
        break;
    case 5:
        lcd.print("Humidity: " + humidity + "%");
        break;
    case 6:
        lcd.print("Clouds: " + clouds + "%");
        break;
    case 7:
        lcd.print("Pressure: " + pressure + "hPa");
        break;
    case 8:
        lcd.print("Sunrise:");
        lcd.setCursor(0, 1);
        lcd.print(sunrise);
        break;
    case 9:
        lcd.print("Sunset:");
        lcd.setCursor(0, 1);
        lcd.print(sunset);
        break;
}
}

// Fungsi parsing data biasa
String getValue(String data, String key) {
    int startIndex = data.indexOf(key);
    if (startIndex == -1) return "N/A";
    startIndex = data.indexOf(":", startIndex) + 1;
    int endIndex = data.indexOf(",", startIndex);
    if (endIndex == -1) endIndex = data.indexOf("}", startIndex);
    String value = data.substring(startIndex, endIndex);
    value.trim();
    return value;
}

```

```

}

// Fungsi parsing rain opsional
String getOptionalValue(String data, String key) {
    int start = data.indexOf(key);
    if (start == -1) return "0"; // kalau tidak ada hujan, return 0
    start += key.length();
    int end = data.indexOf(",", start);
    if (end == -1) end = data.indexOf("}", start);
    String value = data.substring(start, end);
    value.trim();
    return value;
}

// Format UNIX Time ke WIB jam:menit
String formatTime(String unixTimeStr) {
    unsigned long unixTime = unixTimeStr.toInt();

    // Tambah offset 7 jam untuk WIB
    unixTime += 7 * 3600;

    int hours = (unixTime % 86400L) / 3600;
    int minutes = (unixTime % 3600) / 60;

    char buffer[6];
    sprintf(buffer, "%02d:%02d", hours, minutes);
    return String(buffer);
}

```

## 4.2 Diagram.json

```

{
    "version": 1,
    "author": "Ifadah",
    "editor": "wokwi",
    "parts": [
        { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left":
0, "attrs": {} },
        {
            "type": "wokwi-lcd1602",
            "id": "lcd1",
            "top": 35.2,
            "left": 255.2,
            "attrs": { "pins": "i2c" }
        },
        {

```

```

        "type": "wokwi-pushbutton",
        "id": "btn1",
        "top": 303.8,
        "left": -67.2,
        "attrs": { "color": "red", "xray": "1", "label": "Prev" }
    },
    {
        "type": "wokwi-pushbutton",
        "id": "btn2",
        "top": 303.8,
        "left": 115.2,
        "attrs": { "color": "blue", "xray": "1", "label": "Next" }
    }
],
"connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
    [ "lcd1:SCL", "esp:D22", "green", [ "h-124.8", "v-66.9" ] ],
    [ "lcd1:VCC", "esp:3V3", "red", [ "h-124.8", "v86.5" ] ],
    [ "lcd1:GND", "esp:GND.1", "black", [ "h-144", "v57.6" ] ],
    [ "lcd1:SDA", "esp:D21", "green", [ "h-134.4", "v-28.6" ] ],
    [ "btn2:2.1", "esp:GND.2", "black", [ "h-28.8", "v-96", "h-172.8"
] ],
    [ "btn2:1.1", "esp:D12", "blue", [ "h-19.2", "v-86.4", "h-105.6",
"v-105.6" ] ],
    [ "btn1:1.1", "esp:D14", "red", [ "h0", "v-28.8", "h48", "v-
163.2" ] ],
    [ "btn1:2.1", "esp:GND.2", "black", [ "h-9.6", "v-67", "h86.4" ]
]
],
"dependencies": {}

```