

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

-----o0o-----



BÁO CÁO ĐỒ ÁN

**ĐỀ TÀI: HỆ THỐNG ĐO CHẤT
LƯỢNG KHÔNG KHÍ VÀ
CẢNH BÁO CHÁY**

Giảng viên hướng dẫn : ThS. Lê Anh Tuấn

Sinh viên thực hiện:

1. Bùi Đức Anh Tú - 21522735
2. Hồ Minh Trí - 21522701
3. Nguyễn Huy Hoàng - 21522094

Lớp : NT131.O21

TP. Hồ Chí Minh, 18 tháng 5 năm 2024

LỜI CẢM ƠN

Lời đầu tiên, xin chân thành cảm ơn Ban Giám hiệu Trường Đại học Công nghệ thông tin, khoa Mạng máy tính và truyền thông và quý thầy Lê Anh Tuấn đã tạo nền tảng, điều kiện giúp chúng tôi có hội tiếp cận và nghiên cứu chủ đề này.

Chúng em cũng muốn bày tỏ lòng biết ơn sâu sắc đến các thầy cô trong phòng Lab E3.1 cho phép chúng em sử dụng các thiết bị IOT để hoàn thành chủ đề này.

Trong quá trình thực hiện về chủ đề này, có thể do hiểu biết còn nhiều hạn chế nên bài làm khó tránh khỏi những thiếu sót. Chúng tôi rất mong nhận được những lời góp ý chân thành của thầy để bài báo cáo ngày càng hoàn thiện hơn.

Trân trọng.

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

MỤC LỤC	
LỜI CẢM ƠN	2
NHẬN XÉT CỦA GIẢNG VIÊN.....	3
MỤC LỤC	4
1. Giới thiệu.....	6
1.1. Lý do chọn đề tài.....	6
1.2. Ý tưởng triển khai	6
2. Thiết bị sử dụng.....	6
2.1. Arduino Uno R3.....	6
<i>Bảng 1: Thông số kỹ thuật Arduino Uno R3</i>	<i>6</i>
<i>Hình 1: Arduino Uno R3.....</i>	<i>7</i>
2.2. Wemos D1.....	7
<i>Bảng 2: Thông số kỹ thuật Wemos D1</i>	<i>7</i>
<i>Hình 2: Wemos D1</i>	<i>8</i>
2.3. DHT22.....	8
<i>Bảng 3: Thông số kỹ thuật DHT22</i>	<i>8</i>
<i>Hình 3: Hình ảnh và sơ đồ chân DHT22</i>	<i>9</i>
2.4. MQ2	9
<i>Bảng 4: Thông số kỹ thuật MQ2</i>	<i>9</i>
<i>Hình 4: Hình ảnh và sơ đồ chân MQ2</i>	<i>10</i>
2.5 GP2Y1010AU0F.....	10
<i>Bảng 5: Thông số kỹ thuật GP2Y1010AU0F</i>	<i>10</i>
<i>Hình 5: Hình ảnh và sơ đồ chân GP2Y1014AU0F</i>	<i>11</i>
3. Công cụ, Thư viện sử dụng.....	11
3.1. Công cụ	11
3.1.1 Arduino IDE.....	11
3.1.2 EC2, NodeJS, Nginx	12
3.2. Thư viện	12
3.2.1. SoftwareSerial.h.....	12
3.2.2. DHT.h	13
3.2.3. MQUnifiedsensor.h.....	13

3.2.4. ArduinoJson.h.....	13
3.2.5. ESP8266WiFi.h	13
3.2.6 ESP8266HttpClient.h	13
4. Triển khai mô hình.....	14
4.1 Lược đồ bảng mạch	14
<i>Hình 6: Lược đồ bảng mạch</i>	<i>14</i>
4.2. Mô hình thực tế.....	14
<i>Hình 7: Mô hình mạch thực tế</i>	<i>15</i>
4.3. Code xử lý	15
4.4 Web Application	18
<i>Hình 8: Giao diện đăng nhập</i>	<i>19</i>
<i>Hình 9, 10: Giao diện đăng xuất.....</i>	<i>20</i>
<i>Hình 11: Giao diện web application chưa đăng nhập</i>	<i>20</i>
<i>Hình 12: Giao diện web application đã được đăng nhập</i>	<i>21</i>
<i>Hình 13: Dữ liệu từ sensor được lưu</i>	<i>21</i>
<i>Hình 14: Dữ liệu user được lưu.....</i>	<i>22</i>
5. Tài liệu tham khảo:	22

1. Giới thiệu

1.1. Lý do chọn đề tài

Trong bối cảnh đẩy mạnh công nghiệp hóa, đô thị hóa dẫn đến việc ô nhiễm không khí ngày càng nghiêm trọng. Việc sử dụng hệ thống giám sát chất lượng không khí giúp người sử dụng có thể biết được thông tin về mức độ ô nhiễm không khí và có thể tránh xa các tác động xấu đối với sức khỏe. Ngoài ra người sử dụng còn Biết được sớm các nguy cơ cháy nổ hoặc nguy hiểm do khí CO. Cuối cùng là có thể điều chỉnh môi trường sống để tối ưu hoá sức khỏe và sự thoải mái

1.2. Ý tưởng triển khai

- Sử dụng 3 cảm biến DHT22, MQ2, GP2Y1010AU0F, thu thập dữ liệu gửi tới Arduino
- Arduino thiết lập 1 cổng giao tiếp tới Wemos D1 để gửi dữ liệu
- Wemos D1 đóng gói dữ liệu và gửi lên Web Application
- Web Application gửi cảnh báo cho người sử dụng khi quá mức cho phép

2. Thiết bị sử dụng

2.1. Arduino Uno R3

Là mạch để xử lý các dữ liệu được đọc từ cảm biến

Bảng 1: Thông số kỹ thuật Arduino Uno R3

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	khoảng 30mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

Hình ảnh Arduino Uno R3:



Hình 1: Arduino Uno R3

2.2. Wemos D1

Là mạch để nhận dữ liệu được xử lý bởi Arduino và gửi lên web application

Bảng 2: Thông số kỹ thuật Wemos D1

Vi điều khiển	ESP8266EX
Điện áp hoạt động	3.3V
Xung clock	80-160 MHz
Điện áp vào giới hạn	9-24V DC
Số chân Digital I/O	11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
Số chân Analog	1 (điện áp vào tối đa 3.3V)
Dòng ra tối đa (5V)	1A
Wifi	2.4GHz
Bộ nhớ flash	4MB
Hỗ trợ bảo mật	WPA/WPA2
Tích hợp giao thức	TCP/IP

Hình ảnh Wemos D1:



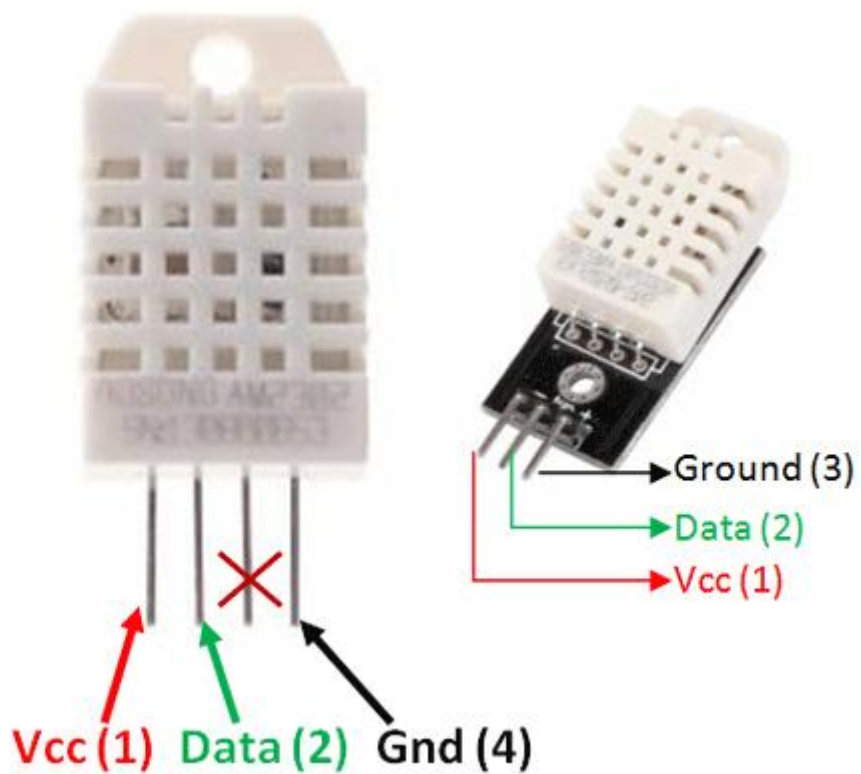
Hình 2: Wemos D1

2.3. DHT22

Là cảm biến dùng để đo nhiệt độ, độ ẩm trong không khí

Bảng 3: Thông số kỹ thuật DHT22

Nguồn sử dụng	3-5 V DC
Dòng sử dụng	2,5mA (khi truyền dữ liệu)
Phạm vi nhiệt độ	-40 ° C đến 80 ° C
Phạm vi độ ẩm	0% đến 100%
Độ phân giải	16 bit
Độ chính xác	±0,5 ° C và ± 1%



Hình 3: Hình ảnh và sơ đồ chân DHT22

Vcc	Chân nguồn
Data	Tín hiệu Digital đầu ra
Gnd	Chân nối đất

2.4. MQ2

Là cảm biến dùng để đo nồng độ khí CO, LPG, H2 trong không khí

Bảng 4: Thông số kỹ thuật MQ2

Nguồn sử dụng	3-5 V DC
Dòng sử dụng	150mA
Dãy hoạt động	300ppm – 10000ppm.
Dạng tín hiệu	Analog



Hình 4: Hình ảnh và sơ đồ chân MQ2

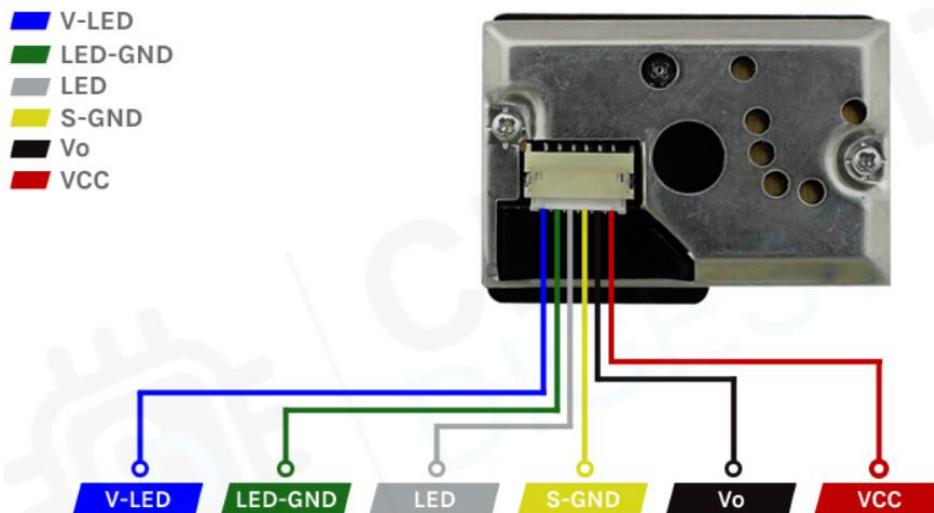
Vcc	Chân nguồn
Aout	Tín hiệu Analog đầu ra
Dout	Tín hiệu Digital đầu ra
Gnd	Chân nối đất

2.5 GP2Y1010AU0F

Là cảm biến dùng để đo nồng độ bụi mịn PM2.5 trong không khí nhờ vào hiện tượng tán xạ ánh sáng bởi các hạt bụi

Bảng 5: Thông số kỹ thuật GP2Y1010AU0F

Điện áp cung cấp	4,5-5V DC
Dòng tiêu thụ	11mA
Các hạt nhỏ nhất giá trị phát hiện	0.8m
Độ nhạy	0.5V / (0.1mg / m3)



Hình 5: Hình ảnh và sơ đồ chân GP2Y1010AU0F

V-LED	Chân cấp nguồn 5V cho LED
LED-GND	Chân nối đất của LED
LED	Chân sử dụng để bật LED phát ra ánh sáng hồng ngoại
S-GND	Chân nối đất của cảm biến bụi quang GP2Y1010AU0F
Vo	Chân tạo ra các xung điện áp ở dạng tín hiệu tương tự (analog) tỉ lệ với lưu lượng bụi.
VCC	Chân cấp nguồn cho cảm biến quang học

3. Công cụ, Thư viện sử dụng

3.1. Công cụ

3.1.1 Arduino IDE

- Arduino IDE (Integrated Development Environment) là một phần mềm mã nguồn mở được sử dụng để viết và tải mã lên bo mạch Arduino. Đây là công cụ chính thức phát triển cho nền tảng Arduino, giúp lập trình viên viết mã, biên dịch và tải mã lên bo mạch một cách dễ dàng.
- Các tính năng chính của Arduino IDE:
 - Trình soạn thảo mã nguồn:
 - Hỗ trợ cú pháp và màu sắc cho mã Arduino.
 - Tích hợp các chức năng như cắt, dán, sao chép và tìm kiếm.

- Biên dịch mã:
 - Biên dịch mã nguồn thành mã máy có thể chạy trên bo mạch Arduino.
 - Cung cấp thông tin về lỗi cú pháp hoặc lỗi biên dịch.
- Tải mã lên bo mạch:
 - Gửi mã đã biên dịch lên bo mạch Arduino thông qua cổng USB.
 - Hỗ trợ nhiều loại bo mạch Arduino khác nhau.
- Công cụ quản lý thư viện:
 - Hỗ trợ cài đặt và quản lý các thư viện bổ sung để mở rộng chức năng của chương trình Arduino.
- Serial Monitor:
 - Cho phép giao tiếp với bo mạch Arduino thông qua giao diện dòng lệnh.
 - Giúp theo dõi và gỡ lỗi mã bằng cách hiển thị dữ liệu từ bo mạch Arduino.

3.1.2 EC2, NodeJS, Nginx

- Amazon EC2 (Elastic Compute Cloud)

Amazon EC2 là một dịch vụ web của Amazon Web Services (AWS) cung cấp khả năng thuê máy chủ ảo để chạy các ứng dụng trên đám mây.

- Node.js

Node.js là một nền tảng mã nguồn mở và đa nền tảng cho phép chạy mã JavaScript trên máy chủ. Nó dựa trên công cụ V8 của Google Chrome và được thiết kế để xây dựng các ứng dụng mạng hiệu suất cao, đặc biệt là các ứng dụng thời gian thực.

- Nginx

Nginx (phát âm là "engine-x") là một phần mềm máy chủ web mã nguồn mở, có thể hoạt động như một máy chủ proxy ngược (reverse proxy), máy chủ cân bằng tải (load balancer), và máy chủ cache HTTP. Nginx được biết đến với hiệu suất cao, khả năng chịu tải tốt và cấu hình linh hoạt.

3.2. Thư viện

3.2.1. SoftwareSerial.h

- SoftwareSerial.h là một thư viện trong Arduino IDE, cho phép bạn tạo thêm các cổng giao tiếp UART (serial) bằng phần mềm trên các chân kỹ thuật số bất kỳ của bo mạch Arduino. Điều này rất hữu ích khi bạn cần nhiều cổng serial để giao tiếp với nhiều thiết bị cùng một lúc, trong khi bo mạch chỉ có một hoặc hai cổng serial phần cứng (hardware serial).

Link: <https://github.com/arduino/ArduinoCore-avr/blob/63092126a406402022f943ac048fa195ed7e944b/libraries/SoftwareSerial/src/SoftwareSerial.h>

3.2.2. DHT.h

- DHT.h là một thư viện trong Arduino IDE dùng để giao tiếp với các cảm biến nhiệt độ và độ ẩm DHT11, DHT22 và AM2302. Các cảm biến này rất phổ biến trong các dự án IoT và tự động hóa nhờ vào tính đơn giản và chi phí thấp.

Link: <https://github.com/adafruit/DHT-sensor-library.git>

3.2.3. MQUnifiedsensor.h

- MQUnifiedsensor.h là một thư viện trong Arduino IDE, dùng để giao tiếp với các cảm biến khí độc hại dựa trên công nghệ cảm biến MQ. Thư viện này giúp bạn đọc dữ liệu từ các cảm biến MQ một cách dễ dàng và chính xác, để theo dõi chất lượng không khí hoặc phát hiện các khí độc hại.

Link: <https://github.com/miguel5612/MQsensorsLib.git>

3.2.4. ArduinoJson.h

- ArduinoJson.h là một thư viện mã nguồn mở trong Arduino IDE, được sử dụng để phân tích và tạo chuỗi JSON (JavaScript Object Notation) trên các bo mạch Arduino và các thiết bị nhúng khác. JSON là một định dạng dữ liệu phổ biến được sử dụng để truyền dữ liệu giữa các ứng dụng và các thiết bị trên mạng.

Link: <https://github.com/bblanchon/ArduinoJson.git>

3.2.5. ESP8266WiFi.h

- ESP8266WiFi.h là một thư viện trong Arduino IDE được sử dụng để kết nối bo mạch Arduino với mạng Wi-Fi bằng module Wi-Fi tích hợp trên bo mạch ESP8266. Điều này cho phép bo mạch Arduino (ví dụ như Wemos D1) có khả năng kết nối với Internet thông qua mạng Wi-Fi.

Link:

<https://github.com/esp8266/Arduino/blob/685f2c97ff4b3c76b437a43be86d1dfdf6cb33e3/libraries/ESP8266WiFi/src/ESP8266WiFi.h>

3.2.6 ESP8266HTTPClient.h

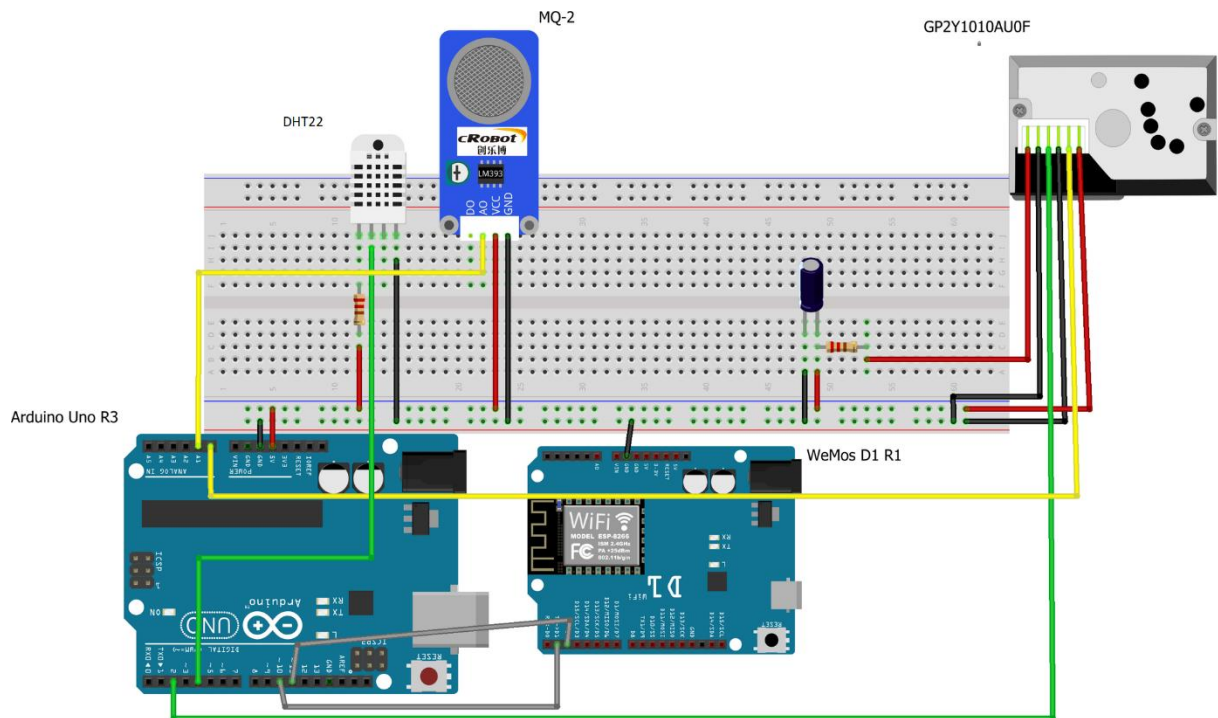
- ESP8266HTTPClient.h là một thư viện trong Arduino IDE, được sử dụng để tạo và gửi các yêu cầu HTTP từ bo mạch Arduino sử dụng module Wi-Fi ESP8266. Thư viện này giúp bạn gửi yêu cầu HTTP đến các máy chủ web và nhận phản hồi từ chúng, cho phép bạn tương tác với các dịch vụ web hoặc API thông qua giao thức HTTP.

Link:

<https://github.com/esp8266/Arduino/blob/685f2c97ff4b3c76b437a43be86d1dfdf6cb33e3/libraries/ESP8266HTTPClient/src/ESP8266HTTPClient.h>

4. Triển khai mô hình

4.1 Lược đồ bảng mạch

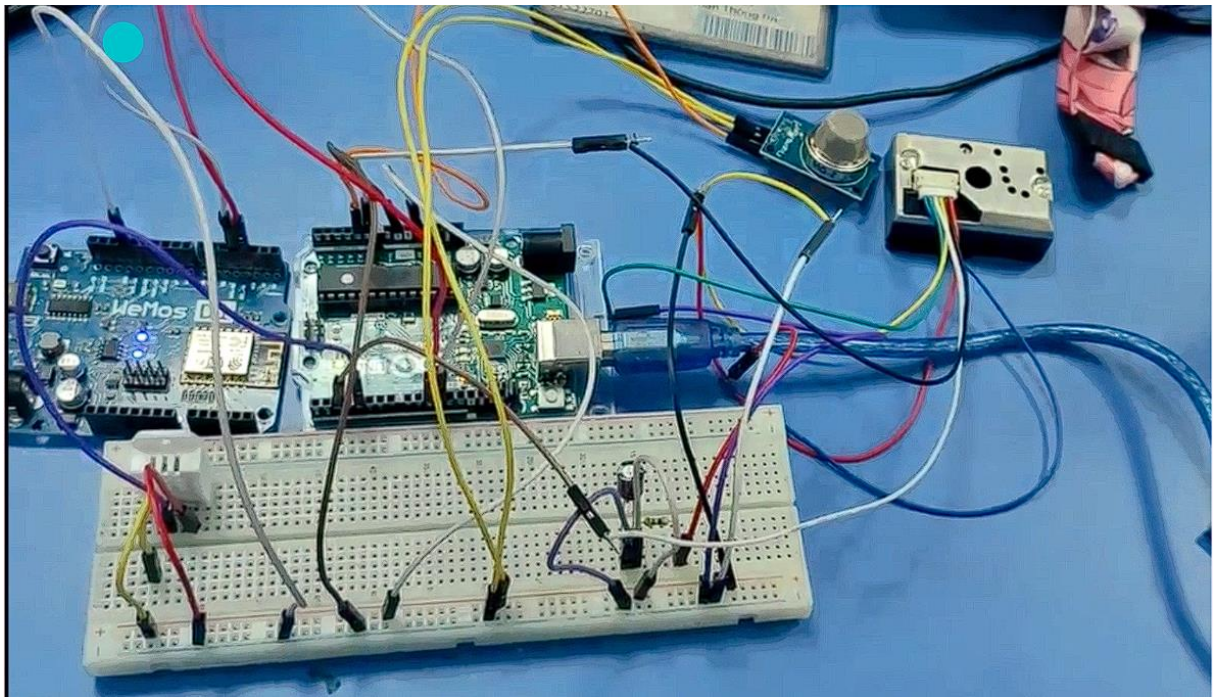


Hình 6: Lược đồ bảng mạch

Giải thích hoạt động:

- Sử dụng 3 cảm biến DHT22, MQ2, GP2Y1010AU0F để thu thập các dữ liệu cần thiết gửi tới Arduino
- Arduino giao tiếp uart với Wemos D1 để gửi các dữ liệu thu thập được qua Wemos
- Wemos tiến hành kết nối wifi và gửi dữ liệu nhận được từ Arduino lên Web Application

4.2. Mô hình thực tế



Hình 7: Mô hình mạch thực tế

4.3. Code xử lý

- Arduino

Khai báo các thư viện cũng như các biến để thu thập dữ liệu

```
#include "DHT.h"
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
#include <MQUnifiedsensor.h>

#define DHTPIN 4
#define DHTTYPE DHT22

#define Board ("Arduino UNO")
#define Pin (A1) //Analog input 1 of arduino
/*****Software Related Macros*****/
#define Type ("MQ-2") //MQ2
#define Voltage_Resolution (5)
#define ADC_Bit_Resolution (10) // For arduino UNO/MEGA/NANO
#define RatioMQ2CleanAir (9.83) //RS / R0 = 9.83 ppm

int measurePin = A0;
int ledPower = 2;

int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;

float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;
```

Khai báo các đối tượng cổng giao tiếp, đối tượng DHT, MQ2, và file JSON.

```
#define RX_PIN 10 // Chân RX của cổng nối tiếp mềm
#define TX_PIN 11 // Chân TX của cổng nối tiếp mềm

SoftwareSerial mySerial(RX_PIN, TX_PIN); // Tạo một đối tượng cổng nối tiếp mềm
DHT dht(DHTPIN, DHTTYPE);
MQUnifiedsensor MQ2(Board, Voltage_Resolution, ADC_Bit_Resolution, Pin, Type);
StaticJsonDocument<100> doc;
```

Hàm xử lý thu thập nhiệt độ, độ ẩm

```
float h = dht.readHumidity();
float t = dht.readTemperature();
if (isnan(h) || isnan(t)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
}
```

Hàm xử lý thu thập nồng độ CO

```
MQ2.setRegressionMethod(1); // _PPM = a*ratio^b
MQ2.setA(36974);
MQ2.setB(-3.109); //
MQ2.init();
float calcR0 = 0;
for (int i = 1; i <= 10; i++) {
    MQ2.update(); // Update data, the arduino will read the voltage from the analog pin
    calcR0 += MQ2.calibrate(RatioMQ2CleanAir);
    Serial.print(".");
}
MQ2.setR0(calcR0 / 10);
Serial.println(" done!.");

if (isinf(calcR0)) {
    Serial.println("Warning: Connection issue, R0 is infinite (Open circuit detected) please check your wiring and supply");
    while (1)
        ;
}
if (calcR0 == 0) {
    Serial.println("Warning: Connection issue found, R0 is zero (Analog pin shorts to ground) please check your wiring and supply");
    while (1)
        ;
}
/***** MQ Calibration *****/

MQ2.serialDebug(true);

MQ2.update(); // Update data, the arduino will read the voltage from the analog pin
float CO_concentration = MQ2.readSensor();
```


Hàm tính nồng độ bụi mịn

```
digitalWrite(ledPower, LOW);          // Bật IR LED
delayMicroseconds(samplingTime);      //Delay 0.28ms
voMeasured = analogRead(measurePin);  // Đọc giá trị ADC V0
delayMicroseconds(deltaTime);         //Delay 0.04ms
digitalWrite(ledPower, HIGH);         // Tắt LED
delayMicroseconds(sleepTime);         //Delay 9.68ms

// Tính điện áp từ giá trị ADC
calcVoltage = voMeasured * (5.0 / 1024); //Điện áp Vcc của cảm biến (5.0 hoặc 3.3)
dustDensity = 170 * calcVoltage - 0.1;
```

Tiến hành đóng gói dữ liệu lại thành file JSON và gửi qua Wemos

```
doc["temp"] = t;
doc["humi"] = h;
doc["co"] = CO_concentration;
doc["dust"] = dustDensity;
serializeJson(doc, mySerial);
mySerial.println();
serializeJson(doc, Serial);
Serial.println();
delay(1000);
```

- Wemos

Khai báo các thư viện cần thiết và các biến

```
#include <SoftwareSerial.h> // Thư viện cho EspSoftwareSerial
#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include < >
#define RX_PIN 4 // Chân RX của EspSoftwareSerial
#define TX_PIN 5 // Chân TX của EspSoftwareSerial
StaticJsonDocument<100> doc;
SoftwareSerial mySerial(RX_PIN, TX_PIN); // Tạo một đối tượng EspSoftwareSerial

const char* ssid = "Nummm";
const char* psswd = "21522735";
```

Tiến hành kết nối Wifi

```

void setup() {
  Serial.begin(9600); // Khởi tạo cổng nối tiếp cứng với tốc độ baud là 9600
  mySerial.begin(9600); // Khởi tạo cổng nối tiếp mềm với tốc độ baud là 9600
  WiFi.begin(ssid, pswd);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected! IP address: ");
  Serial.println(WiFi.localIP());
}

```

Đọc các giá trị mà Arduino gửi qua

```

if (mySerial.available()) {
  // Đọc một dòng từ cổng nối tiếp
  String line = mySerial.readStringUntil('\n');

  // Chuyển đổi chuỗi JSON thành đối tượng JsonDocument
  DeserializationError error = deserializeJson(doc, line);

  // Kiểm tra xem có lỗi nào trong quá trình chuyển đổi hay không
  if (error) {
    Serial.print("deserializeJson() failed: ");
    Serial.println(error.c_str());
    return;
  }

  // Lấy giá trị nhiệt độ và độ ẩm từ đối tượng JsonDocument
  float temperature = doc["temp"];
  float humidity = doc["humi"];
  float CO_concentration = doc["co"];
  float dustDensity = doc["dust"];
}

```

Gửi các giá trị nhận được lên Web Application

```

Serial.println();
http.begin(client, "http://nhhoang15.site/api/data");
http.addHeader("Content-Type", "application/json");
Serial.print("[HTTP] POST...\n");
// int httpCode = http.POST("{\"temp\":\"" + String(temperature) + "\",\"humi\":\"" + String(humidity) + "\",\"co\":\"" + String(CO_concentration) + "\",\"dust\":\"" + String(dustDensity) + "\"}");
int httpCode = http.POST("{\"temp\":\"" + String(temperature) + "\",\"humi\":\"" + String(humidity) + "\",\"co\":\"" + String(CO_concentration) + "\",\"dust\":\"" + String(dustDensity) + "\"}");
if (httpCode > 0) {
  Serial.printf("[HTTP] POST... code: %d\n", httpCode);
  if (httpCode == HTTP_CODE_OK) {
    const String& payload = http.getString();
    Serial.print("received payload:");
    Serial.println(payload);
  }
} else {
  Serial.printf("[HTTP] POST... failed, error: %s\n", http.errorToString(httpCode).c_str());
}
http.end();
}

```

4.4 Web Application

Front-end

HTML,CSS(Bootstrap Framework): Dùng để xây dựng UI cho web

Back-end

Node js : môi trường để chạy socket IO

MongoDB: Database trên cloud

JSCharting: dùng để vẽ biểu đồ số liệu

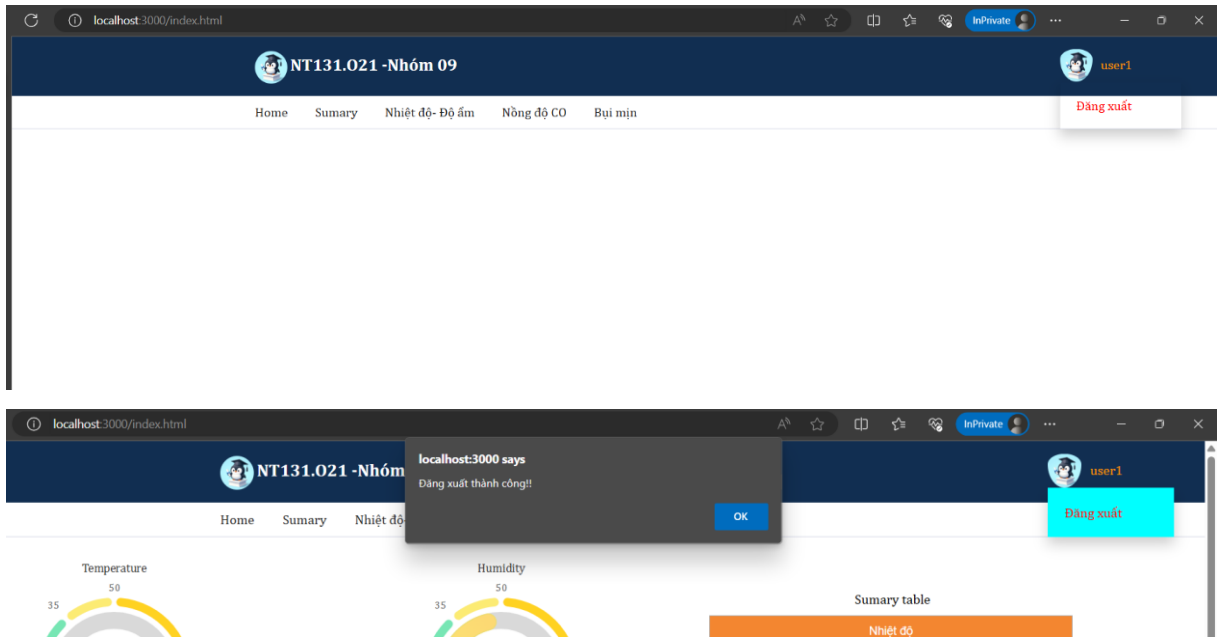
Giao diện Web

- Giao diện đăng nhập

The image shows a web browser window displaying a login page. The browser's address bar indicates the URL is 'localhost:3000/index.html'. The page has a dark blue header with a user profile icon and the text 'NT131.021 -Nhóm 09'. Below the header is a navigation menu with links: 'Home', 'Summary', 'Nhiệt độ- Độ ẩm', 'Nồng độ CO', and 'Bụi mịn'. The main body of the page is white and contains a centered 'Login' form. This form has two input fields labeled 'Username' and 'Password', followed by a blue 'Login' button. At the bottom of the page is a green footer bar with three links: 'Liên hệ với chúng tôi', 'Về Nhóm 09', and 'contact with us'.

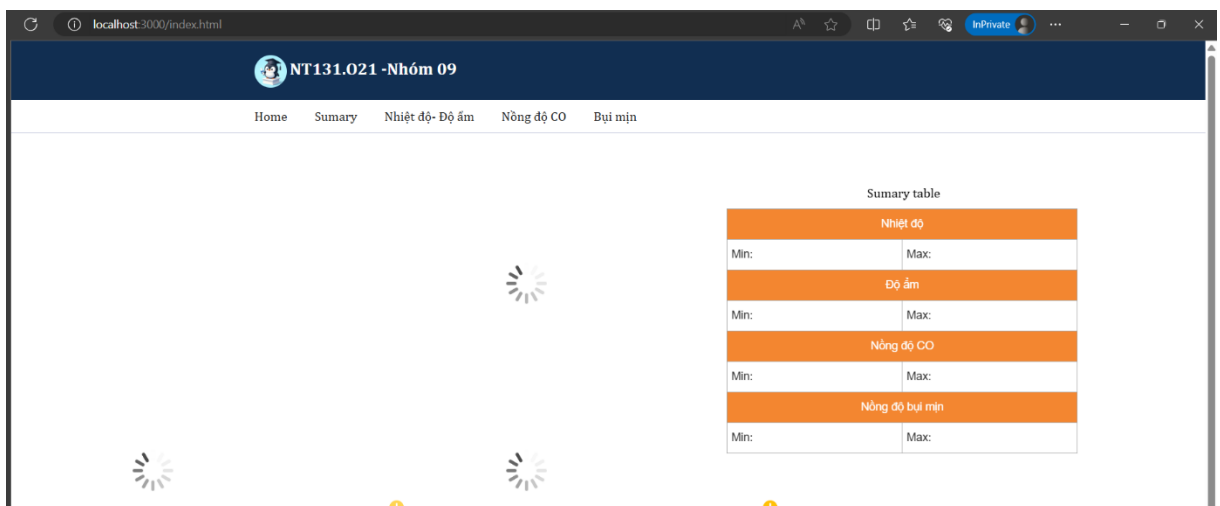
Hình 8: Giao diện đăng nhập

- Tính năng đăng xuất



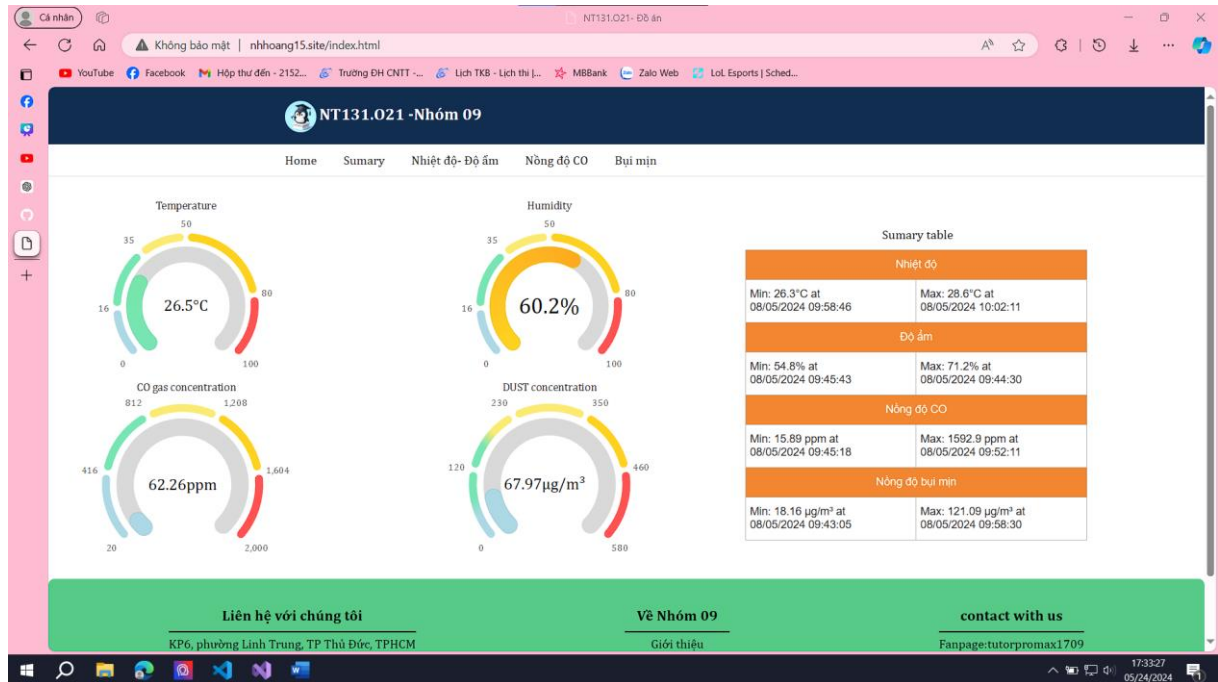
Hình 9, 10: Giao diện đăng xuất

- Giao diện khi chưa đăng nhập



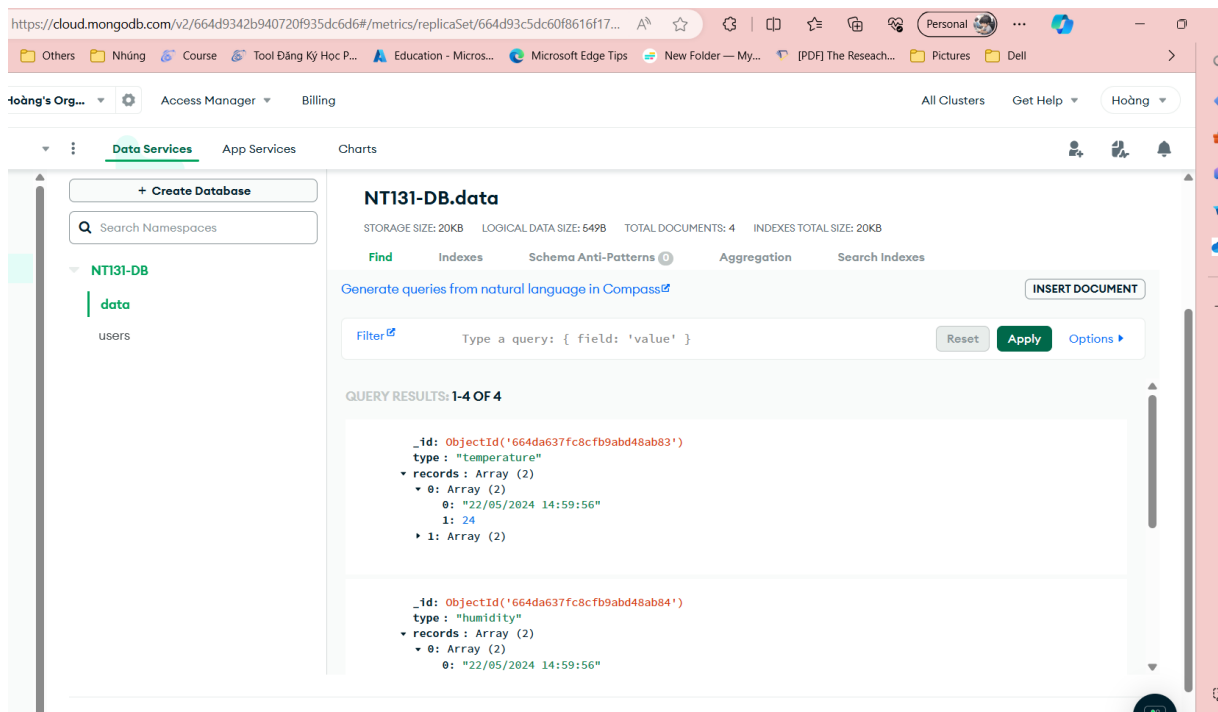
Hình 11: Giao diện web application chưa đăng nhập

- Giao diện khi đăng nhập



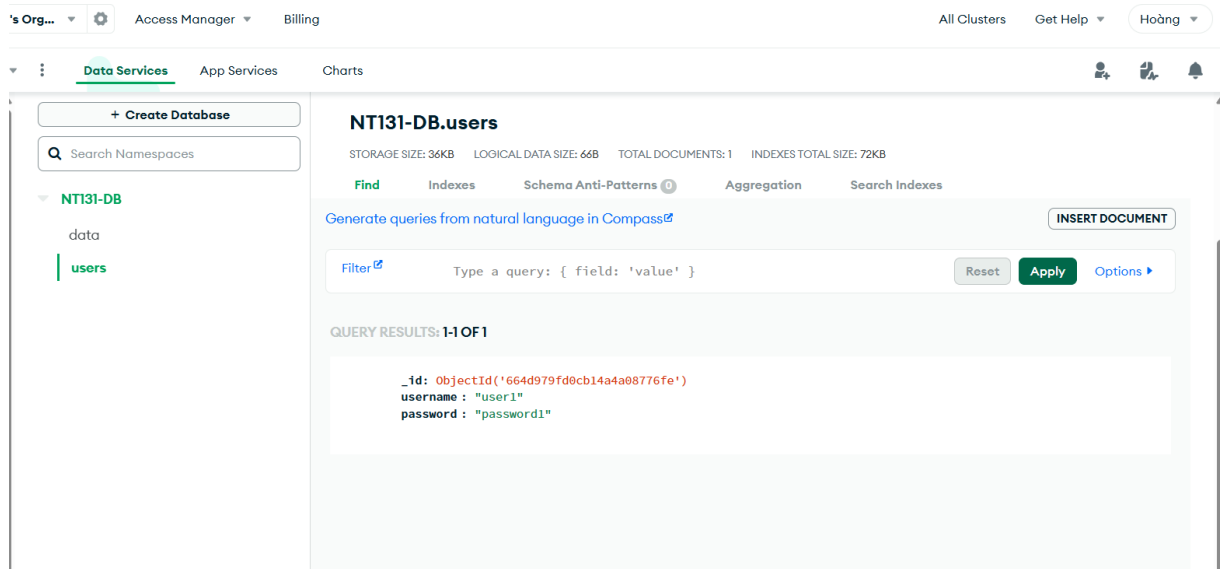
Hình 12: Giao diện web application đã được đăng nhập

- Data được lưu dưới dạng json



Hình 13: Dữ liệu từ sensor được lưu

- Dữ liệu người dùng



Hình 14: Dữ liệu user được lưu

5. Tài liệu tham khảo:

- [https://dienthongminhesmart.com/lap-trinh-esp8266/giao-tiep-uart-esp8266/#Code nap cho Nodemcu esp8266](https://dienthongminhesmart.com/lap-trinh-esp8266/giao-tiep-uart-esp8266/#Code%20nap%20cho%20Nodemcu%20esp8266)
- <https://www.pololu.com/file/0J309/MQ2.pdf>
- <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- [https://www.sparkfun.com/datasheets/Sensors/gp2y1010au e.pdf](https://www.sparkfun.com/datasheets/Sensors/gp2y1010au_e.pdf)
- <https://github.com/adafruit/DHT-sensor-library>
- <https://github.com/miguel5612/MQSensorsLib>