

Wireless Mesh Network

Biên soạn: Nguyễn Văn Bảo
Liên hệ: baonv@uit.edu.vn

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Tìm hiểu wireless mesh network.
- Hiện thực wireless mesh network trên ESP8266.

2. Môi trường, thiết bị thực hành

- Arduino IDE.
- ESP8266.

B. GIỚI THIỆU

1. Wireless mesh network

Mesh network là một mô hình mạng cục bộ trong đó các nút kết nối trực tiếp với nhau, động và không phân cấp với nhau, hợp tác với nhau để định tuyến hiệu quả dữ liệu đến/từ các client.



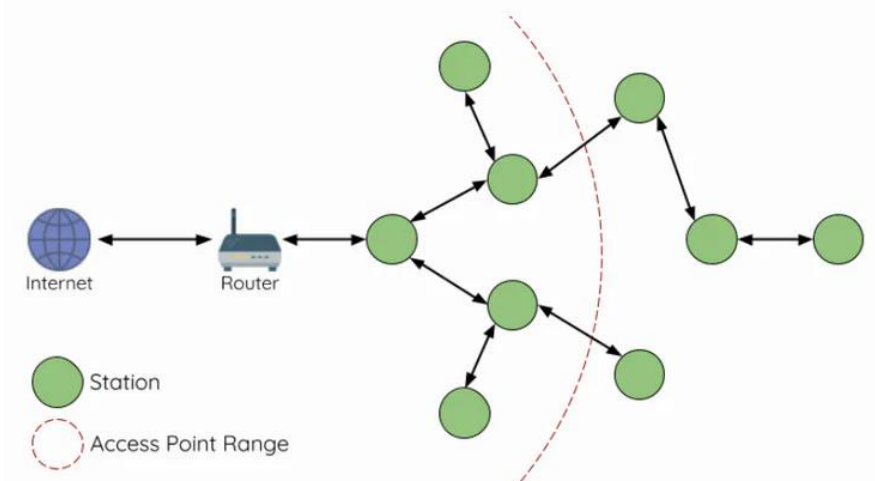
Wireless mesh network là một mesh network trong đó các nút là các thiết bị vô tuyến (không dây) kết nối trực tiếp với nhau. Nó cũng là một dạng của mạng wireless ad hoc.

Trong kiến trúc mạng truyền thống, một nút duy nhất (access point - AP) được kết nối với tất cả các nút (STA) khác. Mỗi nút có thể giao tiếp với nhau bằng AP. Tuy nhiên, điều này được giới hạn trong phạm vi phát sóng vô tuyến của AP. Mỗi STA phải nằm trong phạm vi để kết nối trực tiếp với điểm truy cập.

Với wireless mesh network, các nút không cần kết nối trực tiếp với một nút trung tâm. Các nút chịu trách nhiệm chuyển tiếp thông tin cho nhau. Điều này cho phép nhiều thiết bị trải rộng trên một khu vực vật lý lớn. Các nút có thể tự tổ chức và tự động nói chuyện với

Lab 5: Wireless Mesh Network

nhau để đảm bảo rằng gói đến đích cuối cùng. Nếu bất kỳ nút nào bị xóa khỏi mạng, nó có thể tự tổ chức để đảm bảo rằng các gói đến đích của chúng.



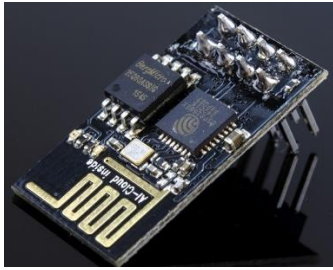
Wireless mesh network được ứng dụng trong các môi trường khó khăn như tình huống khẩn cấp, đường hầm, giàn khoan dầu, giám sát chiến trường, truy cập mạng công cộng, nhà thông minh, ... Hầu hết các ứng dụng trong wireless mesh network tương tự như các ứng dụng trong wireless ad hoc network.

2. ESP8266

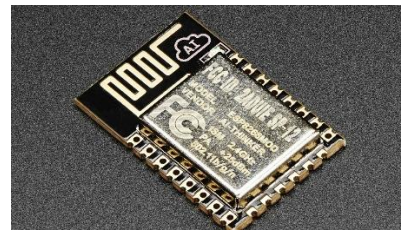
ESP8266 là một vi mạch Wi-Fi 2.4Ghz giá rẻ, có hỗ trợ bộ giao thức TCP/IP và khả năng có thể lập trình được. ESP8266 có một cộng đồng các nhà phát triển lớn trên thế giới, cung cấp nhiều module lập trình mã nguồn mở giúp nhiều người có thể tiếp cận và xây dựng ứng dụng nhanh chóng.

Hiện nay có rất nhiều module sử dụng chip ESP8266 như ESP-01, ESP-07, ESP-12E, ESP-12F, ... và nhiều board mạch sử dụng các module này như NodeMCU, WeMos, ...

Lab 5: Wireless Mesh Network



Hình 1: Module ESP-01



Hình 2: Module ESP12-E



Hình 3: Mạch ESP8266 NodeMCU

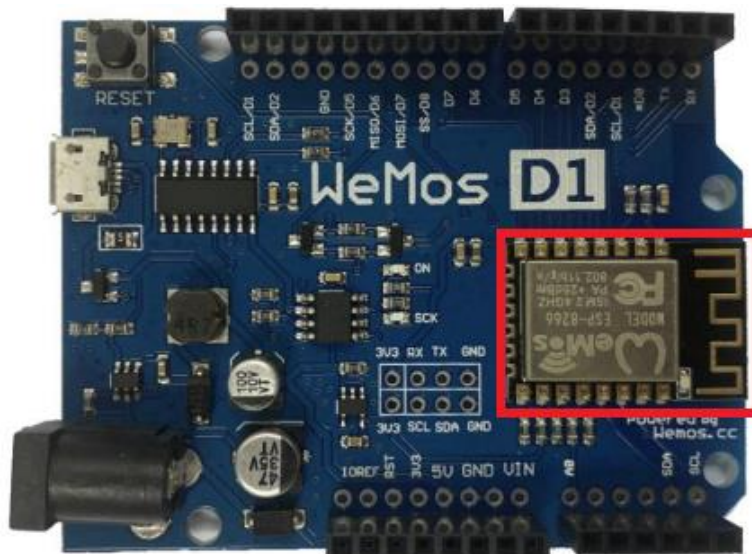


Hình 4: Mạch WeMos D1 R2

Công cụ phát triển phần mềm cho ESP8266 cũng có rất nhiều trong đó có Arduino, do đó chúng ta có thể lập trình cho ESP8266 giống như việc lập trình cho các thiết bị Arduino khác.

a. Wemos D1

Wemos D1 là một board mạch có thiết kế giống như Arduino UNO, sử dụng module ESP-12F.



Module ESP-12F

Các chân GPIO, các chân cắm trên Wemos D1 được đánh số khác với so với Arduino Uno. Điều này có nghĩa là việc khai báo một chân trên Wemos cũng sẽ khác so với việc khai

Lab 5: Wireless Mesh Network

báo trên Arduino Uno. Trong thư viện của board Wemos đã định nghĩa các hằng số cho các chân với GPIO tương ứng với chân đó.

Wemos	D0	D1	D2	D3	D4	D5	D6	D7	D8
ESP8266	GPIO1	GPIO	GPIO	GPIO	GPIO	GPIO1	GPIO1	GPIO1	GPIO1
6	5	5	4	0	2	4	2	3	5

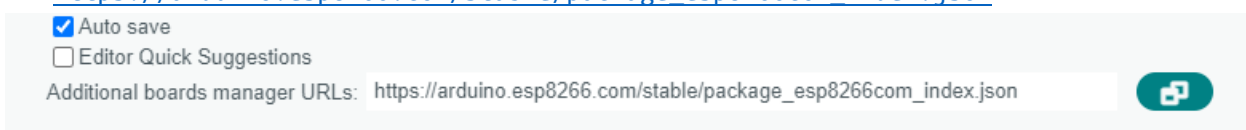
Như vậy ví dụ để khai báo cho chân D5 trên Wemos D1 chúng ta có hai cách như sau:

```
pinMode(14, OUTPUT);
pinMode(D5, OUTPUT);
```

b. Cài đặt

Để lập trình cho ESP8266 nói chung và board mạch Wemos nói riêng trên Arduino IDE chúng ta cần phải tiến hành một số cài đặt.

- B1: Mở Arduino IDE chọn **File -> Preferences** (hoặc nhấn tổ hợp **Ctrl + ,**)
- B2: Thêm vào phần **Additional boards manager URLs** đường dẫn sau và lưu lại https://arduino.esp8266.com/stable/package_esp8266com_index.json



- B3: Vào **Tools -> Boards -> Boards Manager** (**Ctrl + Shift + B**)
- B4: Tìm **esp8266** sau đó bấm **Install**

Cài CH340 driver: <https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all>

Kiểm tra cài đặt: sử dụng chương trình blink trong example để thử nghiệm với board Wemos, trước khi nạp nhớ chọn đúng board: **LOLIN(WEMOS) D1 R2 & mini**

c. Thư viện **painlessMesh**

Thư viện [painlessMesh](https://gitlab.com/painlessMesh/painlessMesh) cho phép chúng ta tạo một mesh network với ESP8266 nhanh chóng. Tham khảo thêm về thư viện: <https://gitlab.com/painlessMesh/painlessMesh>

Cài đặt thư viện:

- B1: **Tools -> Manage Libraries**
- B2: Tìm từ khóa **painlessmesh**

Lab 5: Wireless Mesh Network

LIBRARY MANAGER

painlessmesh

Type: All

Topic: All

Painless Mesh by Coopdis,Scotty Franzyshe,Edwin van Leeuwen,Germán Martín,Maximilian Schwarz,Doanh Doanh
Version 1.5.0
A painless way to setup a mesh with ESP8266 and ESP32 devices
A painless way to setup a mesh with ESP8266 and ESP32 devices
[More info](#)

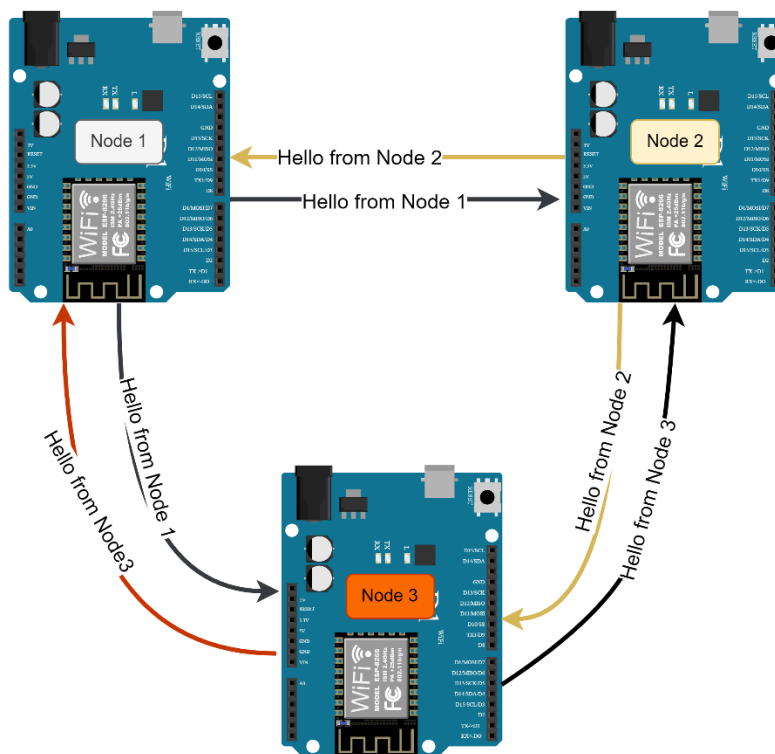
1.4.10 **INSTALL**

- B3: Bấm install, do thư viện painlessmesh cần một số thư viện liên quan nên sẽ có một cửa sổ hiện lên ta cần cài thêm các thư viện đó.
- B4: Cài đặt thêm thư viện ESPAsyncTCP từ file zip: <https://github.com/me-no-dev/ESPAsyncTCP>

C. THỰC HÀNH

1. Broadcast messages

Trong phần này chúng ta sẽ tạo ra một mạng lưới trong đó tất cả thiết bị sẽ broadcast messages tới những thiết bị còn lại



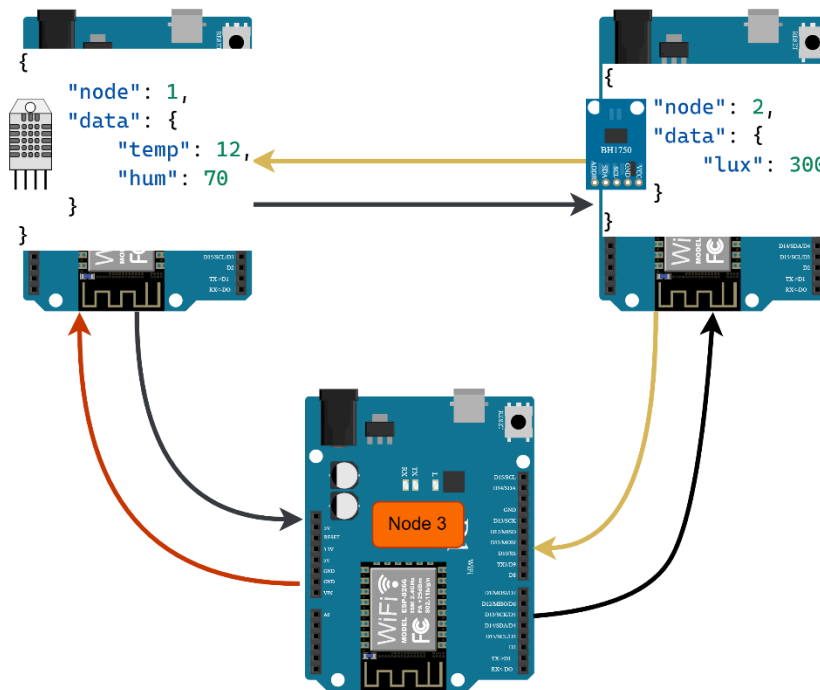
Chương trình: sử dụng chương trình có sẵn nằm trong *File -> Example -> Painless Mesh -> basic*

Lab 5: Wireless Mesh Network

Nạp chương trình lên các board mạch Wemos, sử dụng serial monitor để quan sát các message trên một board Wemos bất kỳ

```
STARTUP: init(): 0
STARTUP: AP tcp server established on port 5555
--> New connection, nodeId = 980353246
Changed connection
Received from 980353246 msg=Hi from node 7274718980353246
Received from 980353246 msg=Hi from node 7274718980353246
Received from 980353246 msg=Hi from node 7274718980353246
Changed connection
Received from 980353246 msg=Hi from node 7274718980353246
Received from 3257223716 msg=Hi from node 24438123257223716
Received from 980353246 msg=Hi from node 7274718980353246
```

2. Trao đổi thông tin cảm biến giữa các nút



Sử dụng lại chương trình ở bài 1 và chỉnh sửa lại để:

- Wemos đọc được cảm biến (DHT22, BH1750), tương tự như code Arduino.
- Đóng gói dữ liệu thành dạng Json, sử dụng thư viện ArduinoJson.

```
StaticJsonDocument<2048> data;
JsonObject sensorData = data.createNestedObject("data");
data["node"] = mesh.getNodeId();
sensorData["temp"] = dht.readTemperature();
sensorData["hum"] = dht.readHumidity();

String message;
```

Lab 5: Wireless Mesh Network

```
serializeJson(data, message);  
// Send message broadcast
```

- Broadcast đoạn json vừa tạo.

D. YÊU CẦU & NỘI BÀI

1. Yêu cầu

- 1) Mô tả chi tiết về cách thức vận hành của một Wireless Mesh Network. Hoàn thành tiếp tục câu 2 ở trên để các node gửi broadcast dữ liệu sensor với nhau.

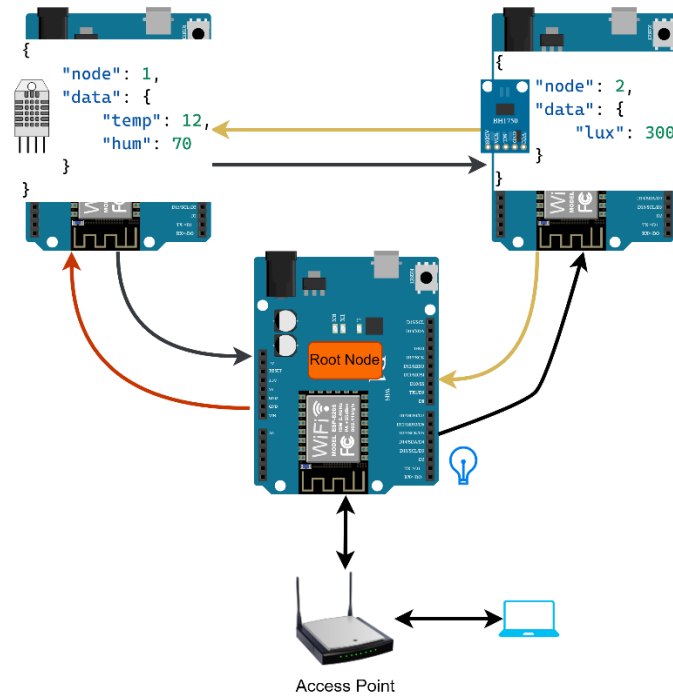
Lưu ý, chuỗi JSON ở dạng sau:

```
{  
  "node": nodeid,  
  "data": {  
    "temp": tem_value,  
    "hum": hum_value,  
  }  
}
```

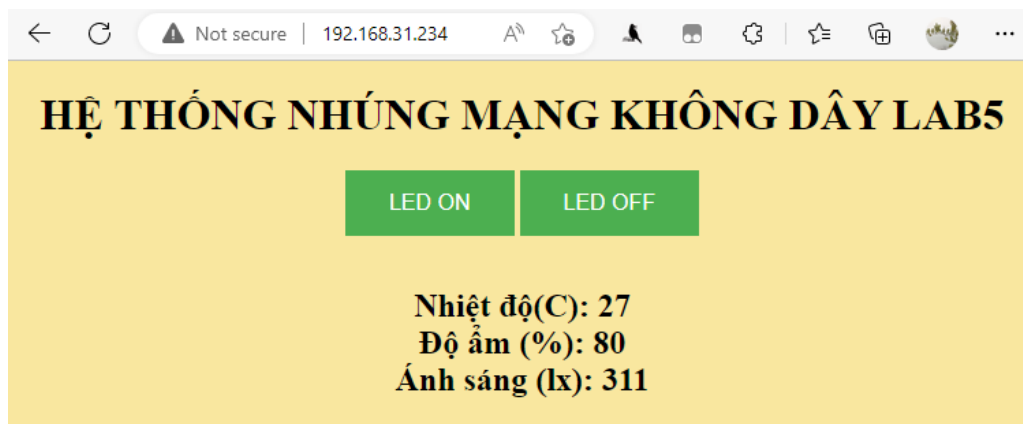
Bên nhận sẽ chỉ in ra giá trị của “node”, “tem” và “hum”.

- 2) Sử dụng node 3 làm Root Node để kết nối thêm một mạng khác (vd: UiTiOt-E3.1), viết một webserver trên Root Node này để cung cấp giao diện hiển thị dữ liệu cảm biến cùng với đó là khả năng điều khiển đèn găng trên root node.
- Mô hình triển khai:

Lab 5: Wireless Mesh Network



- Sử dụng máy tính bên ngoài để kết nối tới địa chỉ IP của Root Node trên access point sẽ cung cấp cho người dùng giao diện để điều khiển cũng như hiển thị thông tin.



Gợi ý: tham khảo ví dụ webServer của thư viện Painless Mesh

```
#define MESH_PREFIX "meshnetwork"
#define MESH_PASSWORD "password"
#define MESH_PORT 5555

#define STATION_SSID "UiTiOt-E3.1"
#define STATION_PASSWORD "password"
```

Viết một webpage giao diện tùy ý nhưng phải đảm bảo các chức năng

```
const char webpage[] PROGMEM = R"====(
```

Lab 5: Wireless Mesh Network

```
...// phần này đã được rút gọn
<h1>Hệ thống nhúng mạng không dây - LAB 5</h1>
  <button class="button" onclick="send(1)">LED ON</button>
  <button class="button" onclick="send(0)">LED OFF</button><BR>
</div>
<br>
<div><h2>
  Nhiệt độ(C): <span id="temp">0</span><br>
  Độ ẩm(%): <span id="hum">0</span><br>
  Ánh sáng(lx): <span id="lux">0</span><br>
</h2>
</div>
<script>
function send(led_sts)
{
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("state").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "led_set?state="+led_sts, true);
  xhttp.send();
}
...// phần này đã được rút gọn
```

Khai báo các HTTP Request trong setup()

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send(200, "text/html", webpage);
});
server.on("/message", HTTP_GET, message)
server.on("/led_set", HTTP_GET, led_control)
```

Hàm xử lý khi người dùng gửi http request tới /led_set của Root Node:

```
void led_control()
{
  String act_state = server.arg("state");
  if(act_state == "1")
  {
    digitalWrite(LED,HIGH);
  }
  else
  {
    digitalWrite(LED,LOW);
  }
}
```

Lab 5: Wireless Mesh Network

```
}
server.send(200, "text/plain", act_state);
}
```

- 3) Tương tự kịch bản 2, thêm đèn led ở tất các node (có thể 1 hoặc nhiều led/node).
Giao diện web thêm tính năng cho phép bật/tắt các đèn này.

Gợi ý:

Có thể định dạng dữ liệu gửi đi như sau:

```
{
  "node": nodeid,
  "data": {
    "temp": tem_value,
    "hum": hum_value,
    "leds": [
      { "id": 0, "state": 0 },
      { "id": 1, "state": 1 }
    ]
  }
}
```

Sử dụng hàm `sendSingle()` để gửi dữ liệu đến 1 node duy nhất:

```
mesh.sendSingle(nodeid, msg);
```

Giao diện gợi ý



2. Nộp bài

- Sinh viên tìm hiểu và thực hành theo hướng dẫn. Thực hiện theo **nhóm**.
- Sinh viên báo cáo kết quả thực hiện và nộp bài bằng file. Trong đó:

Lab 5: Wireless Mesh Network

- Trình bày chi tiết quá trình thực hành và trả lời các câu hỏi nếu có (kèm theo các ảnh chụp màn hình tương ứng).
- Video demo các kịch bản.
- Giải thích, tìm hiểu, lý giải các kết quả đạt được.
- Tải mẫu báo cáo thực hành và trình bày theo mẫu được cung cấp.

Nén **.ZIP** tất cả các file và đặt tên file theo định dạng theo mẫu:

NhomY-LabX_MSSV1_MSSV2

Ví dụ: Nhom1-Lab05_20520001_20520002

- Nộp báo cáo trên theo thời gian đã thống nhất tại website môn học.
- Các bài nộp không tuân theo yêu cầu sẽ **KHÔNG** được chấm điểm.

HẾT

Chúc các bạn hoàn thành tốt bài thực hành!