

РОБОТА №4

Створення веб-сервера на базі модуля ESP8266

Мета: дослідження роботи модуля ESP8266 в режимі веб-сервера.

Завдання: навчитися працювати з модулем ESP8266, використовуючи середовище Arduino IDE, дослідити роботу модуля ESP8266 в режимі веб-сервера.

Обладнання: модуль ESP8266, проводи, світлодіоди, USB-кабель.

Загальні відомості

Веб-сервер – це місце, де зберігаються, обробляються та доставляються веб-сторінки веб-клієнтам. Веб-клієнт – будь-який пристрій, який може отримати доступ до мережі (наприклад, смартфон, розумний годинник, ноутбук або ПК) і запитувати дані з веб-сервера (як правило, за допомогою браузера). Зв'язок між клієнтом та сервером здійснюється за допомогою спеціального протоколу передачі гіпертексту – HTTP (рис. 4.1).



Рисунок 4.1 – Взаємодія веб-сервера та веб-клієнта

За останні кілька років модулі ESP8266 стали незамінними в проєктах, пов'язаних з IoT або WiFi. Це надзвичайно економний модуль WiFi, який з невеликими зусиллями може бути запрограмований для створення автономного веб-сервера чи веб-клієнта.

Однією з найважливіших функцій, яку забезпечує ESP8266, є здатність не тільки підключатися до існуючої WiFi мережі, але також створювати власну мережу, дозволяючи іншим пристроям підключатися до нього та отримувати доступ до веб-сторінок.

Робота ESP8266 може здійснюватися у трьох різних режимах:

- Станція (STA).
- Точка доступу (AP).
- Комбінований.

Режим станції. Модуль ESP8266, який підключається до існуючої WiFi мережі (створеної бездротовим маршрутизатором), називається *Станція* (STA). В даному режимі ESP8266 отримує IP-адресу від бездротового маршрутизатора, до якого він підключений. З цією IP-адресою він може настроїти веб-сервер і доставляти веб-сторінки на всі підключені пристрої в існуючій мережі WiFi (рис. 4.2).

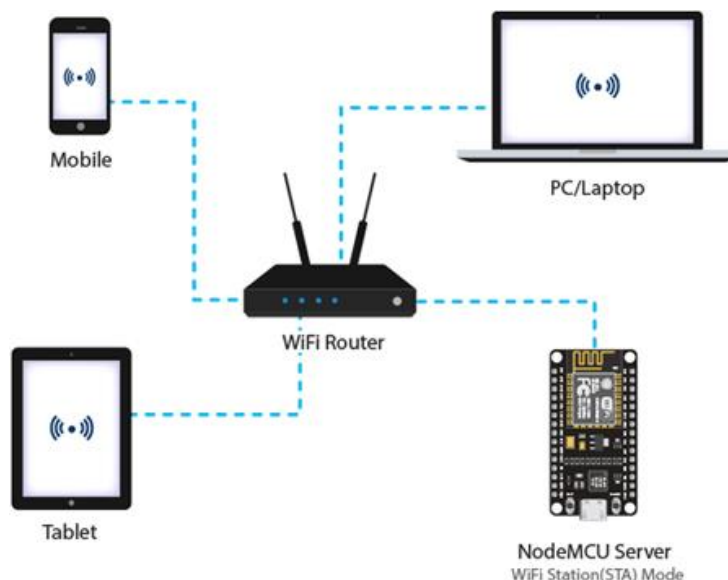


Рисунок 4.2 – Робота модуля ESP8266 в режимі станції

Режим точки доступу. Модуль ESP8266, який створює свою власну WiFi мережу і діє як маршрутизатор WiFi для однієї або декількох станцій, називається *Точкою доступу* (AP). На відміну від WiFi-маршрутизатора, він не має підключення до дротової мережі. Максимальна кількість станцій, які можуть підключитись до нього, обмежена п'ятьма (рис. 4.3).

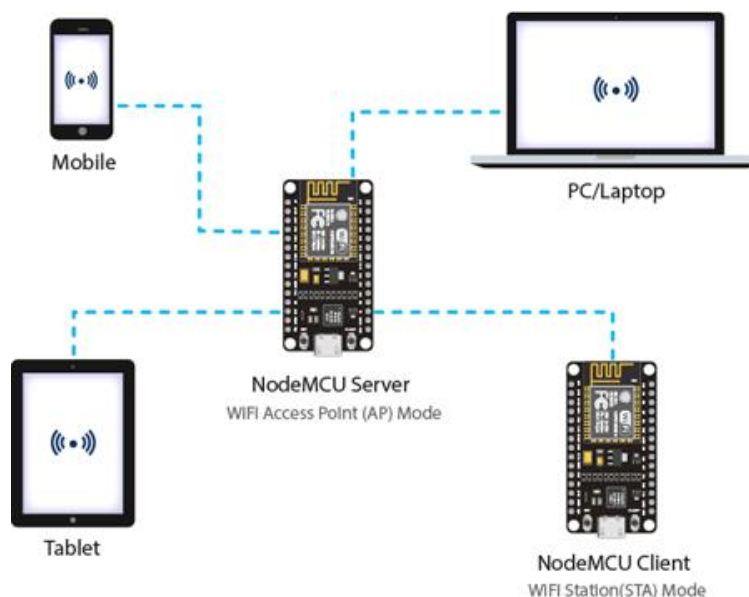


Рисунок 4.3 – Робота модуля ESP8266 в режимі точки доступу

Хід виконання роботи

1. Встановлення підтримки плат ESP в IDE Arduino

Для того, щоб можна було працювати з платами ESP в Arduino IDE необхідно в головному меню перейти в File -> Preferences і на вкладці Settings в параметрі Additional boards manager URLs вказати наступні строки:

http://arduino.esp8266.com/stable/package_esp8266com_index.json,
https://dl.espressif.com/dl/package_esp32_index.json

та зберегти зміни, натиснувши кнопку ОК (рис. 4.4)

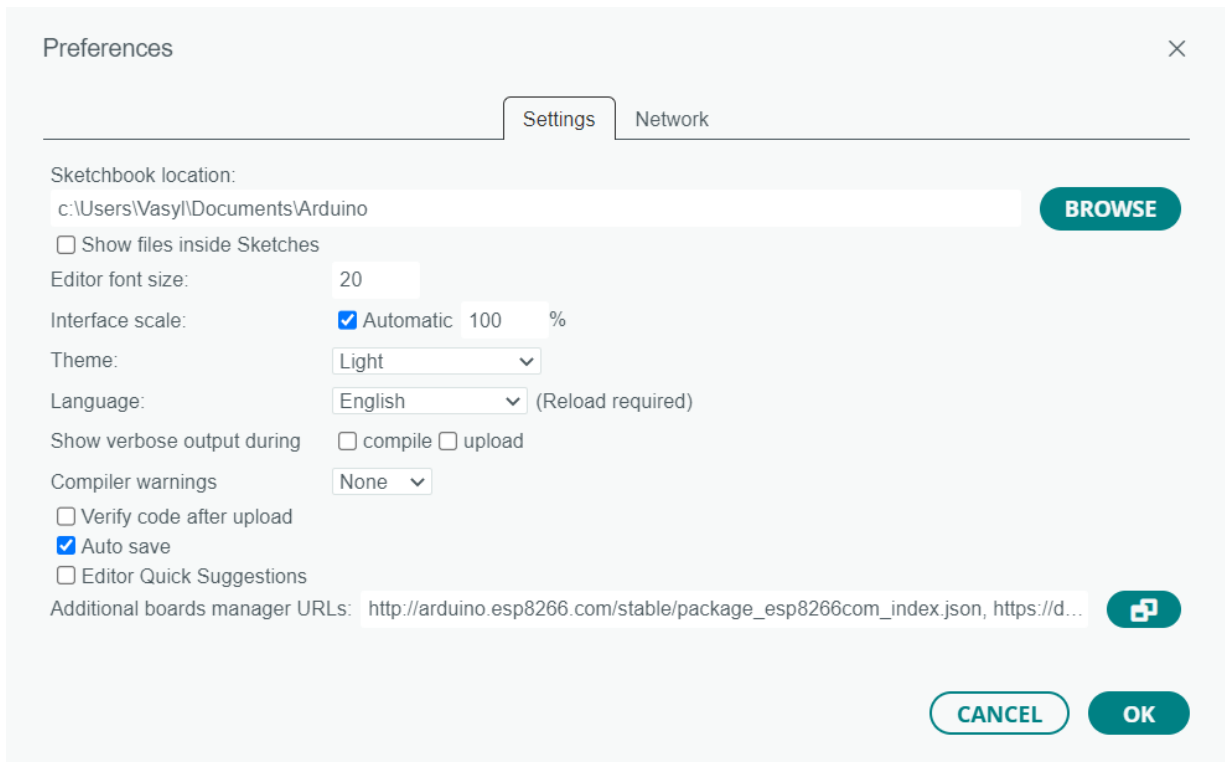


Рисунок 4.4 – Встановлення підтримки плат ESP в IDE Arduino

2. Підключення елементів схеми

Використовуючи модуль ESP8266 NodeMCU, 2 світлодіоди, 2 резистори опором 220Ом та з'єднувальні проводи, змонтуйте наступну схему (рис. 4.5):

ESP8266 NodeMCU	Бредборд
Pin D6	Резистор 220 Ом
Pin D7	Резистор 220 Ом
GND	Катод (коротший контакт) світлодіода

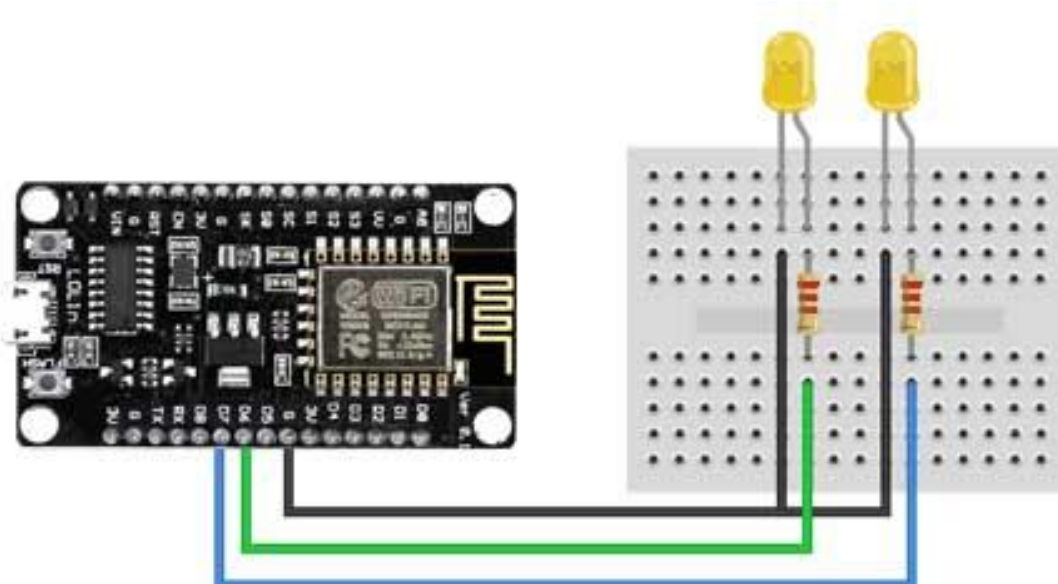


Рисунок 4.5 – Схема підключення модуля ESP8266 NodeMCU

Примітка. Нумерація пінів (GPIO) на платі NodeMCU відрізняється від їх нумерації на модулі ESP8266-12E (рис. 4.6). Врахуйте це при вказуванні номерів пінів програмному коді.

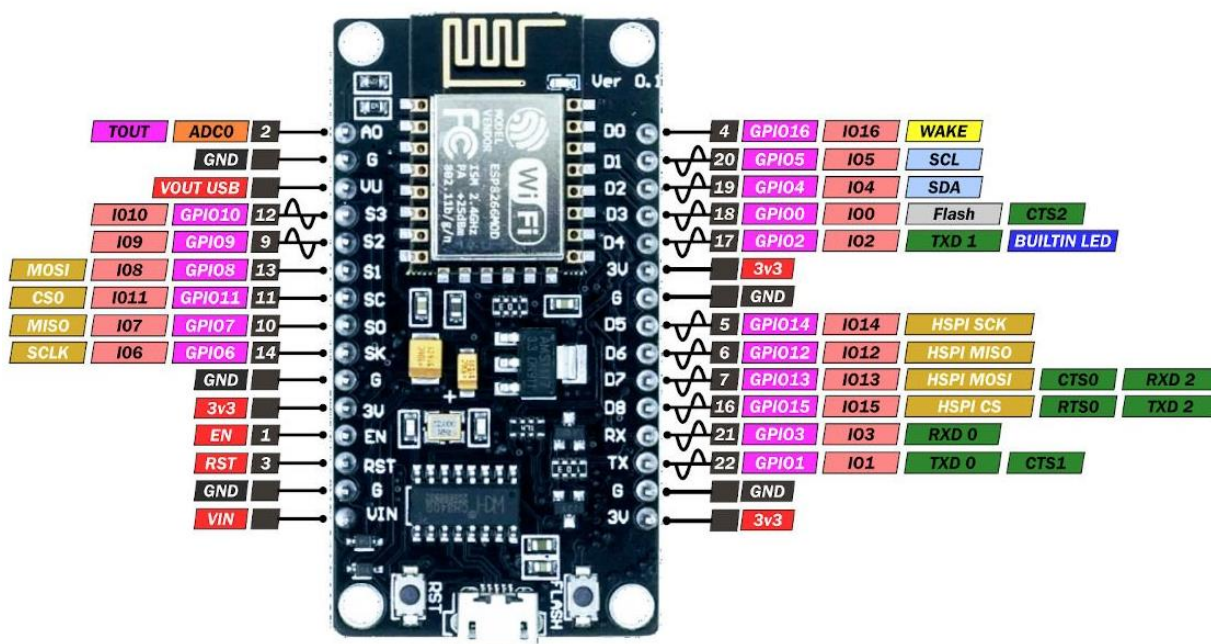


Рисунок 4.6 – Схема виводів ESP8266 NodeMCU

3. Реалізація програмного коду

Підключіть ESP8266 NodeMCU до комп'ютера. Вкажіть відповідний порт та вид плати: Tools -> Board -> esp8266 -> Generic ESP8266 Module (рис. 4.7):

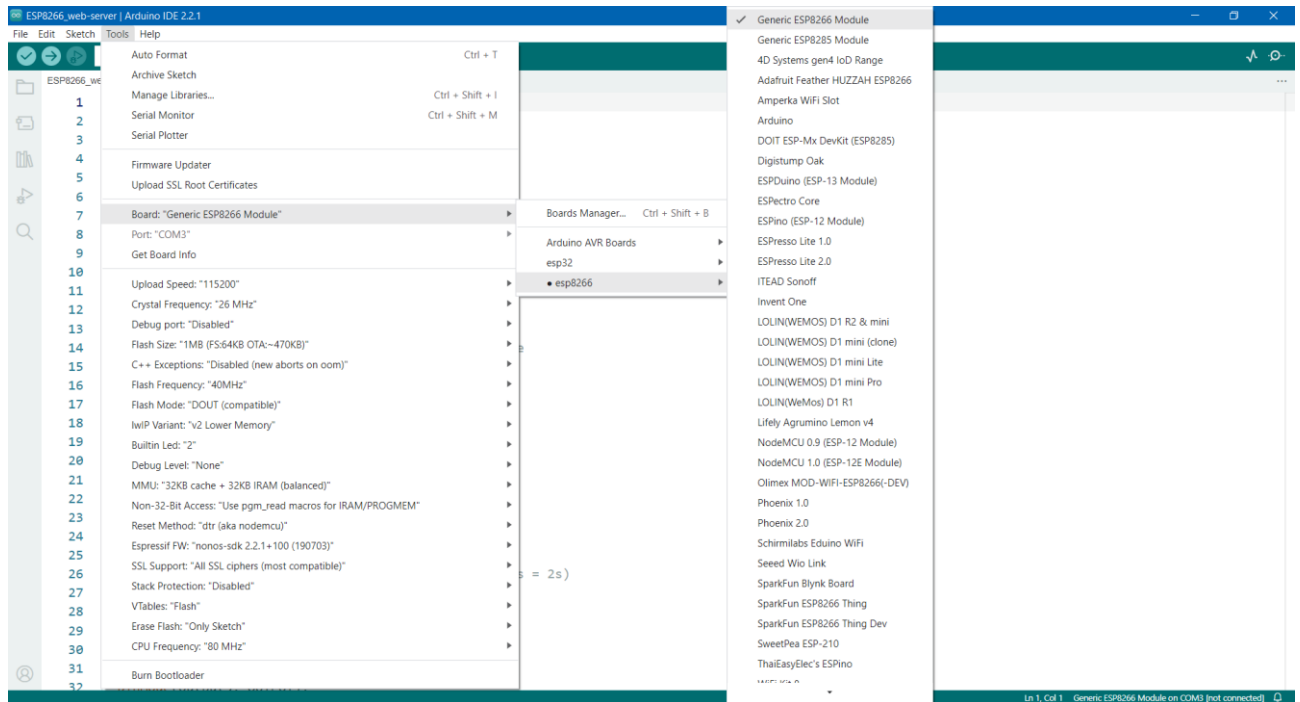


Рисунок 4.7 – Вибір плати Generic ESP8266 Module

Вивантажте на плату наступний скетч:

```
// Підключення бібліотеки Wi-Fi
#include <ESP8266WiFi.h>

// Замініть на свої назву та пароль
// точки доступу WiFi
const char* ssid    = "Your_WiFi";
const char* password = "Your_pwd";

// Встановіть порт web-сервера рівним 80
WiFiServer server(80);

// Змінна для зберігання HTTP-запитів
String header;

// Змінні для зберігання поточних станів
// світлодіодів
String output5State = "off";
String output4State = "off";

// Змінні для зберігання пінів GPIO,
// підключених до світлодіодів
const int output5 = 5;
const int output4 = 4;

// Поточний час
unsigned long currentTime = millis();
// Попередній час
unsigned long previousTime = 0;
// Визначення часу в мілісекундах
const long timeoutTime = 2000;

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

```

// Задання режиму роботи пінів GPIO
pinMode(output5, OUTPUT);
pinMode(output4, OUTPUT);
// Задання початкового стану пінів GPIO
digitalWrite(output5, LOW);
digitalWrite(output4, LOW);

// Підключення до WiFi
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
// Вивід в Serial Monitor отриманої IP-адреси
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
server.begin();
}

void loop(){
    // Перевірка підключення клієнтів
    WiFiClient client = server.available();

    if (client) { // Якщо клієнт підключений,
        Serial.println("New Client.");
        String currentLine = "";
        currentTime = millis();
        previousTime = currentTime;
        while (client.connected() && currentTime - previousTime <=
timeoutTime) { // Циклічний повтор поки підключений клієнт
            currentTime = millis();
            if (client.available()) { // Якщо отримані байти даних від клієнта,
                char c = client.read(); // прочитати байти та
                Serial.write(c); // вивести в Serial Monitor
                header += c;
                if (c == '\n') {
                    if (currentLine.length() == 0) {
                        client.println("HTTP/1.1 200 OK");
                        client.println("Content-type:text/html");
                        client.println("Connection: close");
                        client.println();

                        // Ввімкнення та вимкнення пінів GPIOs
                        if (header.indexOf("GET /5/on") >= 0) {
                            Serial.println("GPIO 5 on");
                            output5State = "on";
                            digitalWrite(output5, HIGH);
                        } else if (header.indexOf("GET /5/off") >= 0) {
                            Serial.println("GPIO 5 off");
                            output5State = "off";
                            digitalWrite(output5, LOW);
                        } else if (header.indexOf("GET /4/on") >= 0) {
                            Serial.println("GPIO 4 on");
                            output4State = "on";
                            digitalWrite(output4, HIGH);
                        } else if (header.indexOf("GET /4/off") >= 0) {

```

```

        Serial.println("GPIO 4 off");
        output4State = "off";
        digitalWrite(output4, LOW);
    }

    // Відображення HTML-сторінки сервера
    client.println("<!DOCTYPE html><html>");
    client.println("<head><meta name=\"viewport\"
content=\"width=device-width, initial-scale=1\">");
    client.println("<link rel=\"icon\" href=\"data:,\>");
    client.println("<style>html { font-family: Helvetica;
display: inline-block; margin: 0px auto; text-align: center;}");
    client.println(".button { background-color: #195B6A; border:
none; color: white; padding: 16px 40px;");
    client.println("text-decoration: none; font-size: 30px;
margin: 2px; cursor: pointer;}");
    client.println(".button2 {background-color:
#77878A;}</style></head>");

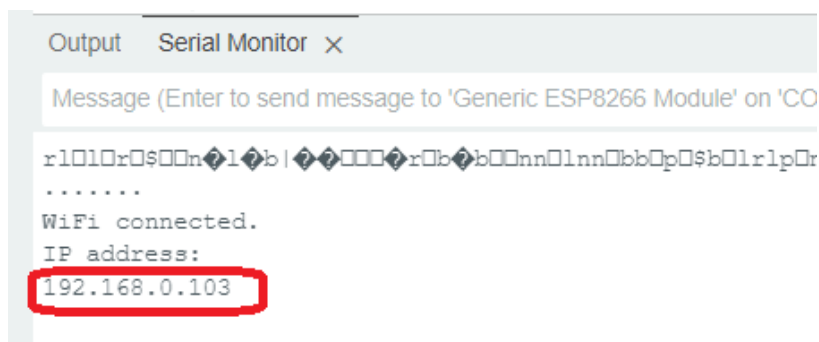
    client.println("<body><h1>ESP8266 Web Server</h1>");
    client.println("<p>GPIO 5 - State " + output5State + "</p>");
    if (output5State=="off") {
        client.println("<p><a href=\"/5/on\"><button
class=\"button\">ON</button></a></p>");
    } else {
        client.println("<p><a href=\"/5/off\"><button
class=\"button button2\">OFF</button></a></p>");
    }
    client.println("<p>GPIO 4 - State " + output4State +
"</p>");
    if (output4State=="off") {
        client.println("<p><a href=\"/4/on\"><button
class=\"button\">ON</button></a></p>");
    } else {
        client.println("<p><a href=\"/4/off\"><button
class=\"button button2\">OFF</button></a></p>");
    }
    client.println("</body></html>");
    client.println();

    break;
} else {
    currentLine = "";
}
} else if (c != '\r') {
    currentLine += c;
}
}
}
header = "";
// Завершення підключення
client.stop();
Serial.println("Client disconnected.");
Serial.println("");
}
}

```

Примітка: Для завантаження коду необхідно натиснути на платі кнопку *FLASH* та утримувати її до початку завантаження.

Після успішного завантаження відкрийте вікно Serial Monitor та переконайтеся, що модуль підключився до вашої мережі WiFi та дізнайтеся IP-адресу, яку він отримав від точки доступу.



```
Output Serial Monitor x
Message (Enter to send message to 'Generic ESP8266 Module' on 'CO
r1010r0s00n1?b|??000?r0b?b00nn0lnn0bb0p0$b0lr1p0r
.....
WiFi connected.
IP address:
192.168.0.103
```

Рисунок 4.8 – Отримана від точки доступу IP-адреса

Тепер відкрийте будь-який браузер та перейдіть за отриманою вище IP-адресою. Завантажена сторінка повинна виглядати як представлено на рисунку 4.9.

ESP8266 Web Server

GPIO 5 - State off



GPIO 4 - State off



Рисунок 4.9 – Веб-інтерфейс сервера

Натискаючи кнопки GPIO відслідкуйте роботу світлодіодів.

Самостійна робота

1) Доповніть дану схему ще одним світлодіодом та реалізуйте код його керуванням.

2) Доповніть дану схему датчиком DHT11 та реалізуйте відображення температури та вологості через веб-інтерфейс.