

ESP8266Sketch5_ConnectionWiFiAdrIP_Fixe

Ce programme utilise un module ESP8266 pour se connecter à un réseau WiFi spécifique en mode station (client). Il configure également une adresse IP statique, un masque de sous-réseau, une passerelle, et un serveur DNS, plutôt que de laisser le routeur attribuer ces paramètres dynamiquement. Une fois la connexion établie, il affiche les informations sur le réseau dans le moniteur série. Ce type de configuration est utile dans des environnements où une adresse IP fixe est requise pour le module.

```
#include <Arduino.h>
#include <ESP8266WiFi.h>

const char* mySSID = "XXXXXXXX";
const char* mypassword = "123456";

IPAddress IP(192, 168, 0, 50);
IPAddress gateway(192, 168, 0, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress dns(8, 8, 8, 8);

void setup(void){
  Serial.begin(115200);
  Serial.println("\r\n\r\n\r\n===== \r\n");
  Serial.print(" Connexion à ");
  Serial.println(mySSID);
  WiFi.disconnect();
  WiFi.config(IP, gateway, subnet, dns);
  WiFi.begin(mySSID, mypassword);
  WiFi.mode(WIFI_STA);

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

  Serial.print("\r\n\r\nConnecté à ");
  Serial.println(mySSID);
  Serial.print("Adresse IP: ");
  Serial.println(WiFi.localIP());
}

void loop() {
}
```

```
#include <ESP8266WiFi.h>
```

- **<ESP8266WiFi.h>** : Elle permet de contrôler les fonctionnalités WiFi du module ESP8266, comme la connexion à un réseau WiFi, la gestion des adresses IP, et la gestion des modes de fonctionnement.

```
const char* mySSID = "XXXXXXXX";  
const char* mypassword = "123456";
```

- **mySSID** et **mypassword** : Ces constantes contiennent respectivement le nom du réseau WiFi (SSID) et le mot de passe. L'ESP8266 utilisera ces informations pour se connecter au réseau.

```
IPAddress IP(192, 168, 0, 50);  
IPAddress gateway(192, 168, 0, 1);  
IPAddress subnet(255, 255, 255, 0);  
IPAddress dns(8, 8, 8, 8);
```

- **IPAddress** : Ce type d'objet permet de définir des adresses IP spécifiques.
 - **IP** : Représente l'adresse IP statique que l'ESP8266 va utiliser (ici 192.168.0.50).
 - **gateway** : Indique l'adresse IP de la passerelle, généralement l'adresse du routeur local.
 - **subnet** : Définit le masque de sous-réseau. Ici, c'est 255.255.255.0, ce qui est standard pour les réseaux domestiques.
 - **dns** : Définit l'adresse IP du serveur DNS à utiliser (ici 8.8.8.8, qui est le DNS public de Google).

```
Serial.print(" Connexion à ");  
Serial.println(mySSID);  
WiFi.disconnect();  
WiFi.config(IP, gateway, subnet, dns);  
WiFi.begin(mySSID, mypassword);  
WiFi.mode(WIFI_STA);
```

- **Serial.println()** : Affiche un message de bienvenue et le nom du réseau auquel l'ESP8266 tentera de se connecter.
- **WiFi.disconnect()** : Déconnecte l'ESP8266 de tout réseau WiFi auquel il pourrait être connecté. Cela garantit que la configuration suivante s'applique correctement.

- **WiFi.config(IP, gateway, subnet, dns)** : Configure l'ESP8266 avec une adresse IP statique, une passerelle, un masque de sous-réseau, et un serveur DNS. Cela remplace l'attribution automatique d'IP (DHCP) par le routeur, ce qui peut être utile dans des environnements où une adresse IP fixe est nécessaire.
- **WiFi.begin(mySSID, mypassword)** : Tente de se connecter au réseau WiFi en utilisant le SSID et le mot de passe définis.
- **WiFi.mode(WIFI_STA)** : Configure l'ESP8266 en mode station (**WIFI_STA**), c'est-à-dire en tant que client WiFi, se connectant à un réseau existant plutôt que d'agir comme point d'accès.

```
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
```

- **WiFi.status()** : Cette fonction renvoie l'état de la connexion WiFi. Le programme entre dans une boucle **while** tant que l'état n'est pas **WL_CONNECTED** (c'est-à-dire que l'ESP8266 n'est pas connecté au WiFi).
- **delay(500)** : Pause de 500 millisecondes (0,5 seconde) entre chaque tentative de connexion. Pendant ce temps, un point (.) est affiché dans le moniteur série pour indiquer que la tentative de connexion est toujours en cours.

```
Serial.print("\r\nConnecté à ");
Serial.println(mySSID);
Serial.print("Adresse IP: ");
Serial.println(WiFi.localIP());
```

Une fois la connexion réussie, les informations sur le réseau sont affichées, y compris :

- **mySSID** : Le nom du réseau auquel l'ESP8266 est connecté.
- **WiFi.localIP()** : L'adresse IP locale de l'ESP8266, qui a été configurée en tant que **192.168.0.50**.