



Microcontrôleurs - ESP8266 Feather Huzzah



[Mise à jour le 10/8/2023]

Ressources

- Généralités sur [Wikipédia](#)
- Sites [Espressif](#) et [Adafruit](#)
- [ESP8266/Arduino](#) sur Github
- [Connectique des différents modèles de cartes](#)
- [Arduino IDE 2](#)

Lectures connexes

- [Découverte des ESP8266 : le microcontrôleur connecté par Wifi pour 2€ au potentiel phénoménal](#)
- Wiki "[Matériels - Capteurs, afficheurs, préactionneurs, etc.](#)"
- Wiki "[Mettre en oeuvre un client MQTT sur un EP8266 \(ESP32\) Feather Huzzah ou MKR1010](#)"
- Wiki "[Le système de fichiers LittleFS \(ESP\)](#)" (Mini Serre)
- Exemples "[ESP8266 First Web Server](#)"
- Arduino pour l'ESP8266 (Documentation) : [ESP8266 Arduino Core](#)
- [Arduino Library List](#)

Distributeurs

- [GO TRONIC](#), ([Adafruit](#))

1. Généralités

La carte Feather HUZZAH ESP8266, développée par Adafruit, est une carte de développement WiFi "tout-en-un" à ESP8226 avec USB intégré et chargeur de batterie.

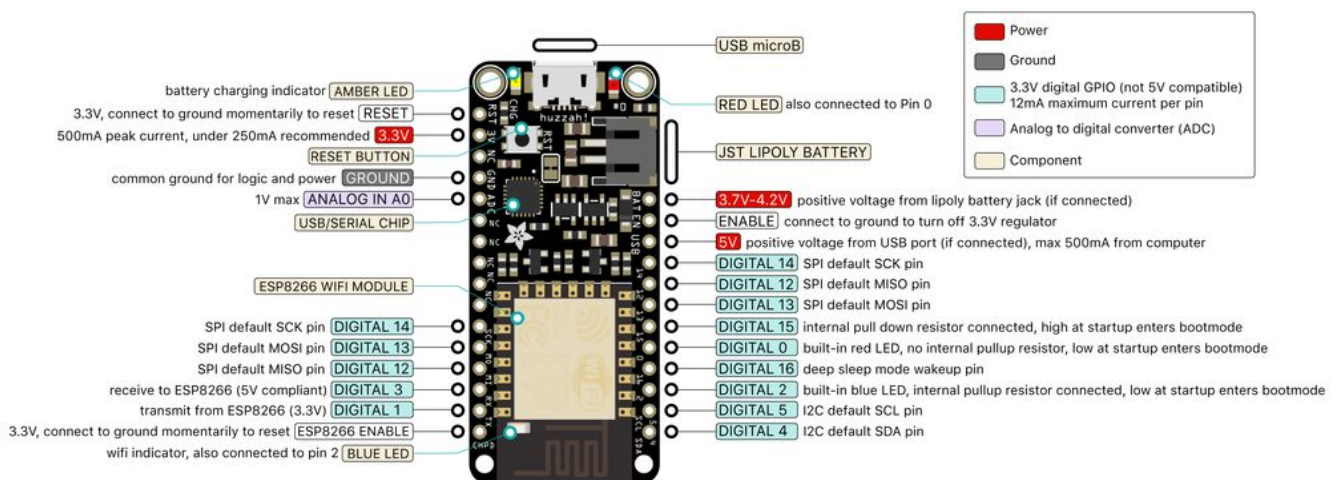
2. Principales caractéristiques

- SOC : [ESP8266](#)

- **Processeur** : 32-bit CPU @ 80MHz
- **Wifi** 802.11 b/g/n (WEP/WPA-PSK/WPA2-PSK)
- **Flash** : 4MB
- **SRAM** : 96K
- **GPIO** : 9 (logique 3,3V) partagés avec I2C et SPI
- **I²C** : 1 (logiciel)
- **SPI** : 1
- **UART** : 2
- **Entrée analogique** : 1 (**1V max**)
- **Alimentation** : 5V régulateur 3.3V (pic de courant 500mA max)
- **Chargeur Lipo intégré** : (3,7V - 4,2V) - 100mA
- **LED** rouge à usage général sur la broche 0
- **Bouton** de Reset
- **Consommation** : 60 à 200mA
- **Dimensions** : 51mm x 23mm x 8mm
- **Masse** : 6g

3. Brochage

- Description de la connectique des **différents modèles** de cartes [Random Nerd Tutorials](#).
- **Identification** et description des entrées / sorties de la carte Huzzah sur le site [Adafruit](#).



4. Accessoires



- **Adafruit Grove Shield FeatherWing** pour Particle Mesh et tous les Feathers¹⁾.

- **Caractéristiques**

- 8 connecteurs Grove :

- 3 connecteurs analogiques, 2 broches analogiques par connecteur, 6 broches analogiques au total;
- 2 connecteurs numériques, 2 broches numériques par connecteur, 4 broches numériques au total;
- 2 connecteurs I2C
- 1 connecteur UART

- **Schéma** de la carte [ici](#)

- **Photo de l'adaptation** à réaliser sur la carte Adafruit Grove Shield (R=10k)²⁾ : [ici](#)

- **Distributeurs** : [Mouser](#) et ([Adafruit](#))

- **Table de correspondance**

Shield	ESP	Commentaires
D2	2	GPIO 2 est connectée à la LED bleue située près de l'antenne WiFi. Utilisable comme sortie.
D3	16	GPIO 16 utilisable comme sortie.
D4	0	GPIO 0 n'a pas de pull-up interne. Est connectée à la LED rouge. Utilisable comme sortie.
D5	15	GPIO 15 utilisable comme sortie.



- **Batterie Lipo 3,7V 500mA**

- **Caractéristiques**

- Tension: 3,7 Vcc
- Intensité: 400 mAh
- Courant de décharge maxi: 880 mA
- Connecteur: 2 broches type JST
- Dimensions: 35,5 x 25,5 x 5,2 mm
- Longueur du câble: 100 mm
- Poids: 10 g

- **Distributeurs** : [Gotronic](#)

5. Préparation de l'IDE Arduino

5.1 Généralités

L'**ESP8266 Arduino core** est livré avec des bibliothèques permettant :

- de communiquer via le WiFi en utilisant les protocoles TCP et UDP,
- de configurer des serveurs HTTP, mDNS, SSDP et DNS,
- d'effectuer des mises à jour OTA,
- d'utiliser un système de fichiers en mémoire flash et
- de travailler avec des cartes SD, des servos, des périphériques SPI et I2C.

5.2 Installation du support pour les cartes à "ESP8266"

- L'**installation du support ESP8266** pour Arduino se fait en passant par l'édition des préférences (**Fichier** → **Préférences**).

Dans « URL de gestionnaire de cartes supplémentaires » ajoutez

```
https://arduino.esp8266.com/stable/package_esp8266com_index.json
```

URL de gestionnaire de cartes supplémentaires 

Remarque : une URL par ligne, pas de séparateur !

5.3 Installation du gestionnaire de carte

- Dans le menu « **Outils** », « **Type de carte** » et « **Gestionnaire de carte** », rechercher « **esp8266 by ESP8266 Community** » et installer le composant.

esp8266 by ESP8266 Community

Cartes incluses dans ce paquet:

Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Sseed Wio Link, ESpectro Core.

[Online help](#)

[More info](#)

2.5.0



Installer

6. Installer MicroPython

[MicroPython - Les modules Espressif ESP32 et ESP8266](#)

7. Démarrer avec la carte ESP8266

7.1 Premier Programme (blink)

*.cpp

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
```

```
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

7.2 Test du Wifi

Le SSID et le mot de passe du réseau doivent avoir été préalablement chargés avec [infoClientMQTT_ESP8266.ino](#)

*.cpp

```
// Connexion au wifi
#include <ESP8266WiFi.h>
// mDNS pour la résolution des noms des hôtes
#include <ESP8266mDNS.h>
// EEPROM : émule une EEPROM dans l'ESP8266
#include <EEPROM.h>
// -----
// Structure pour la configuration de la connexion au réseau wifi
struct EEconf
{ // Les champs sont remplis par le croquis infoClientMQTT_ESP8266.ino
  // avec les données stockées dans l'EEPROM (émulée)
  char ssid[32]; // SSID du réseau. Exemple : SynBoxLAN,
  char password[64]; // Mot de passe du réseau. Exemple : 12345678
  char myhostname[32]; // Nom donné au client MQTT. Exemple :
ESP8266_1
} readconf;
// Objet pour la connexion au réseau wifi
WiFiClient espClient;

// Connexion au Wifi
// -----

void setup_wifi()
{
  // Mode station
  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Tentative de connexion à ");
  Serial.println(readconf.ssid);
```

```
// Connexion au Wifi
WiFi.begin(readconf.ssid, readconf.password);
while (WiFi.status() != WL_CONNECTED)
{
    delay(5000);
    Serial.print(".");
}
// Affichage
Serial.println("");
Serial.println("Connexion au Wifi ok");
Serial.print("MAC: ");
Serial.println(WiFi.macAddress());
Serial.print("Adresse IP : ");
Serial.println(WiFi.localIP());
// Configuration de mDNS
WiFi.hostname(readconf.myhostname);
if (!MDNS.begin(readconf.myhostname))
{
    Serial.println("Erreur de configuration mDNS !");
}
else
{
    Serial.println("Répondeur mDNS démarré");
    Serial.println(readconf.myhostname);
}
}

void setup()
{
    // Configuration du moniteur série
    Serial.begin(115200);
    delay(500);
    // Lecture des paramètres sauvegardés par
    ARD_ESP_SauveInfosClientMqtt.ino
    EEPROM.begin(sizeof(readconf));
    EEPROM.get(0, readconf);
    // Connexion au Wifi
    setup_wifi();
}

void loop()
{
    delay(100);
}
```

Exemple de résultat attendu

```
Tentative de connexion à SynBoxLAN
.
Connexion au Wifi ok
MAC: BC:DD:C2:2D:C0:1B
Adresse IP : 192.168.200.23
Répondeur mDNS démarré
ESP8266FHmno4
```



[Télécharger](#) le projet PlatformIO pour VSCode.

8. Résolution des problèmes

8.1 Installation du driver USB

- Si le port COM n'est pas identifiable lors de la connexion de l'ESP8266 au PC, installer le driver USB depuis [Silicon Lab](#).

8.2 Réinstallation ou mise à jour du firmware

- **Programmation avec Arduino** : [nodemcu-flasher](#) pour Windows 64-bit pour réinstaller le paquet [package_esp8266com_index.json](#) disponible sur github.

Pour aller plus loin

- **ESP8266 SDK** - [Getting Started Guide](#)

1) , 2)

Avec une ESP8266 Feather Huzzah, il est nécessaire de connecter une résistance de 10k entre la broche CHPD et le 3,3V

From:
<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:
<http://webge.fr/dokuwiki/doku.php?id=microc:uc:esp8266&rev=1692356100>

Last update: **2023/08/18 12:55**

