



Sujets : Communiquer

COMMENT COMMUNIQUER AVEC UN MODULE WIFI ?
PREMIERE PARTIE : CONNECTER LE MICROCONTROLEUR AU WIFI

Simon Macquet | 20/11/2019

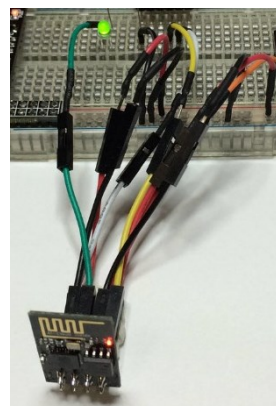
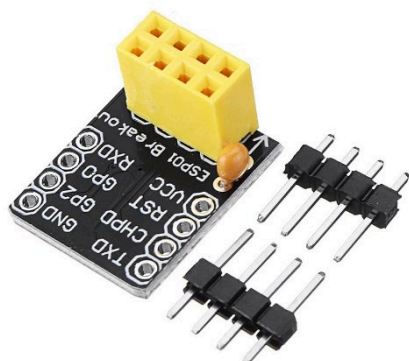


Table des matières

Table des matières.....	1
Introduction	1
Matériel.....	2
Connecter l'arduino uno	2
Initialisation automatique du composant	8
Code	8
Lexique.....	11
Liens utiles.....	12

Introduction

ESP8266 est un microcontrôleur capable de se connecter à un réseau wifi. Il dispose pour ça d'une antenne, d'une liaison série qui lui permet d'échanger des données et de deux entrées sorties digital (par conséquent, si votre application ne nécessite que de deux entrées digitales alors vous n'aurez pas besoin de votre carte arduino). Il y a différentes variantes plus ou moins similaires. On recommande, pour se lancer, la version esp01. Par contre, la carte n'est pas compatible avec un redboard ; alors soit vous fabriquez un adaptateur ou alors vous utilisez des câbles femelles (voir images ci-dessous).



Nous allons voir comment connecter l'arduino uno à un réseau wifi

Matériel

Arduino Uno

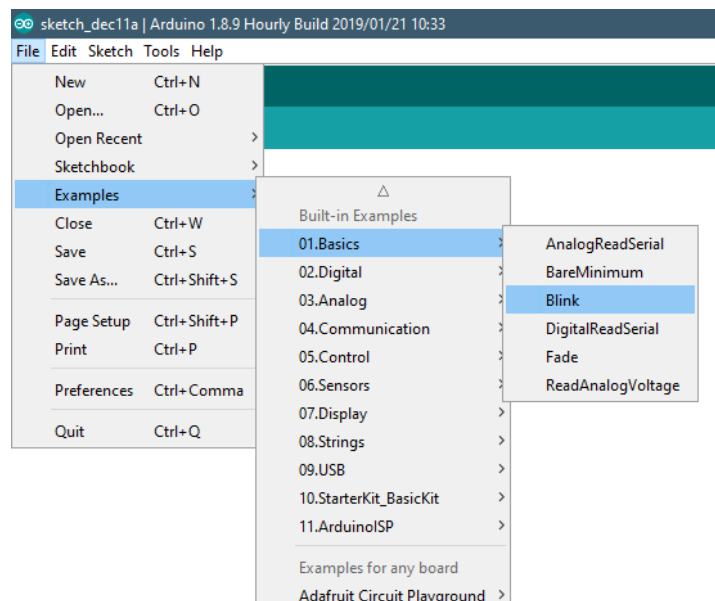
ESP 8266

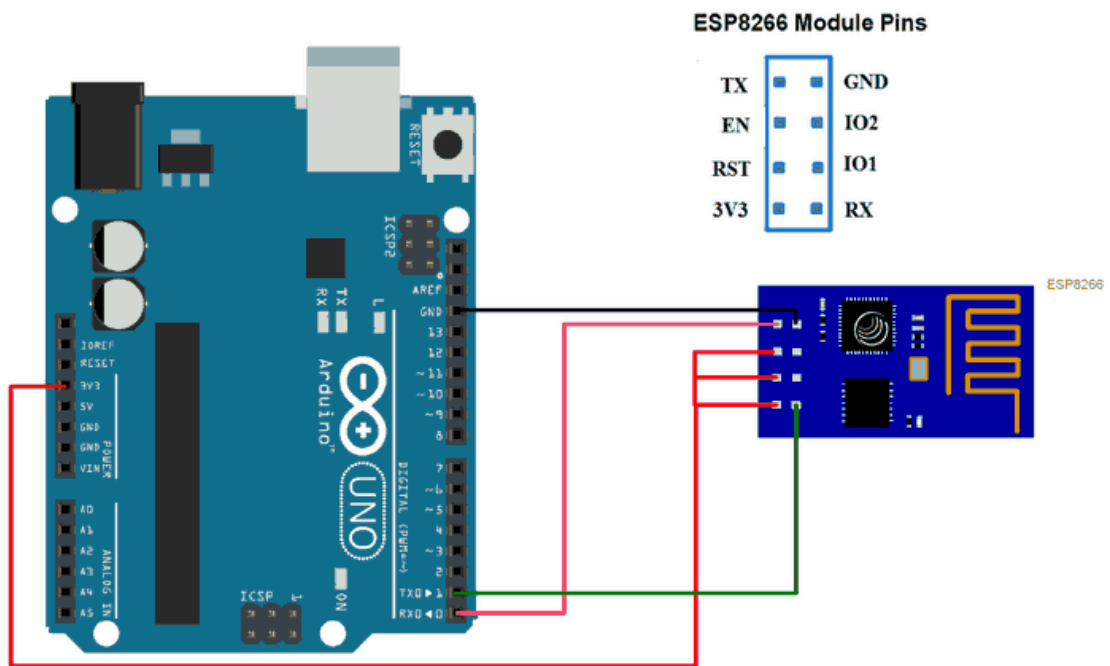
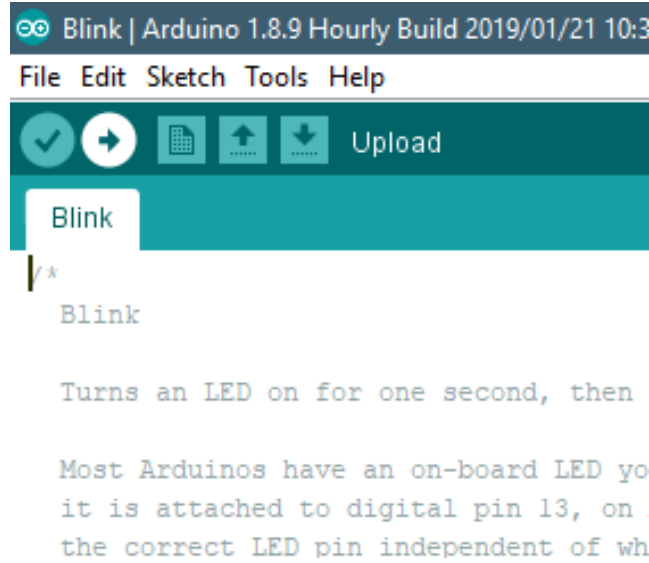
Adaptateur (voir intro)

(Résistances : 1*10 kΩ ; 1*20 k Ω)

Connecter l'arduino uno

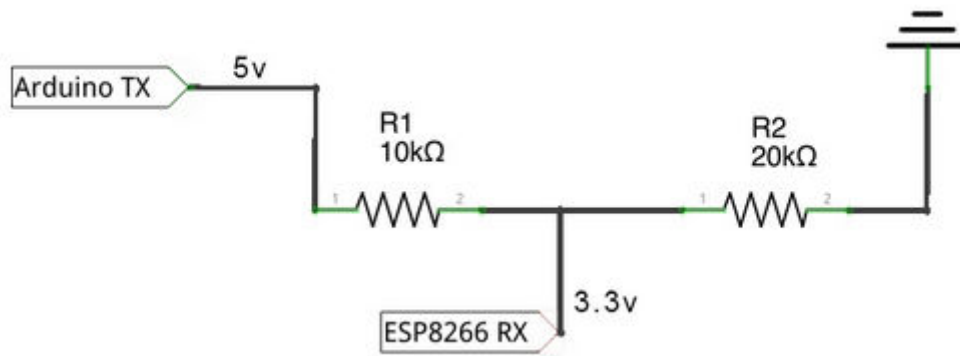
Avant de passer au montage vous allez charger le programme blink dans votre arduino pour s'assurer que la liaison série ne sera pas utilisé par la suite.





Attention l'esp s'alimente seulement en 3.3V.

A priori, il faudrait adapter les niveaux de tension sur la liaison série mais comme elle possède un circuit de protection de 5.8V on peut s'en passer. Néanmoins, si vous êtes amené à utiliser un autre modèle qui ne bénéficie pas de cette protection (voir la documentation technique du composant utilisé) il vous suffit de faire un diviseur de tension à l'aide de 2 résistances :



$$\frac{V_{esp}}{V_{ard}} = \frac{R1}{R1 + R2} \Rightarrow V_{esp} = \frac{R2}{R1 + R2} * V_{ard} = 3.3 V$$

Ensuite lancez l'interface de développement d'arduino sans charger de programme et ouvrez le moniteur série.

Configurez la liaison série¹ en Both NL & CR (New Line et Carriage Return) à 9600 baud² et enfin cliquer sur le bouton *reset* de votre carte.

A partir de là nous allons utiliser des commandes AT³ pour communiquer avec l'ESP.

Commençons par utiliser un reset avec la commande AT+RST.



Ensuite, configurer l'ESP (avec AT+CWMODE) pour qu'il puisse agir en tant que serveur puis en tant que client TCP (Transmission Control Protocol)⁴.

AT+CWMODE - WIFI mode (station/softAP/station+softAP)

Type	Instruction	Response	Function
Query	AT+CWMODE?	+CWMODE:mode OK	Query AP's info which is connect by ESP8266.
Execute	AT+CWMODE=mode	OK	Set AP's info which will be connect by ESP8266.

Parameter Description

mode :

1 means Station mode

2 means AP mode

3 means AP + Station mode



```
AT+CWMODE=3
AT+RST
OK
(a ÉI y`@Çr!BúD K:òb^+
[Vendor:www.ai-thinker.com Version:0.9.2.4]
ready
AT+CWMODE=3
OK
```

Une fois que c'est fait, cherchez un réseau wifi avec la commande AT+CWLAP. Connectez votre resau wifi avec la commande AT+CWJAP, en precisant d'abord le nom du reseau et ensuite le mot de passe.

AT+CWLAP - Lists available APs

Type	Instruction	Response	Function
Set	AT+CWLAP=ssid,mac,ch	+CWLAP:ecn,ssid,rssi,mac OK	Search available APs with specific conditions.
execute	AT+CWLAP	+CWLAP:ecn,ssid,rssi,mac OK	Lists all available APs.

Parameter description :

ecn:

0 OPEN

1 WEP

2 WPA_PSK

3 WPA2_PSK

4 WPA_WPA2_PSK

ssid: string, SSID of AP

rssi: signal strength

mac: string, MAC address

AT+CWJAP – Connect to AP

Type	Instruction	Response	Function
Query	AT+CWJAP?	+ CWJAP:ssid OK	Query AP's info which is connect by ESP8266.
execute	AT+CWJAP=ssid,pwd	OK	Set AP's info which will be connect by ESP8266.

Parameter description :

ssid : string, AP's SSID

pwd : string, MAX: 64 bytes

Example :

```
AT+CWJAP="wifi-1","12345678"
```

Une fois connecté il suffit juste de taper la commande AT+CIFSR pour connaître les adresses IP de l'ESP.

AT+CIFSR – Get local IP address

Type	Instruction	Response	Function
Execute	AT+CIFSR	+CIFSR:IP_address +CIFSR:IP_address OK	Get local IP address.

Parameter description :

IP_address:

IP address of ESP8266 softAP

IP address of ESP8266 station

Examples

AT+CIFSR

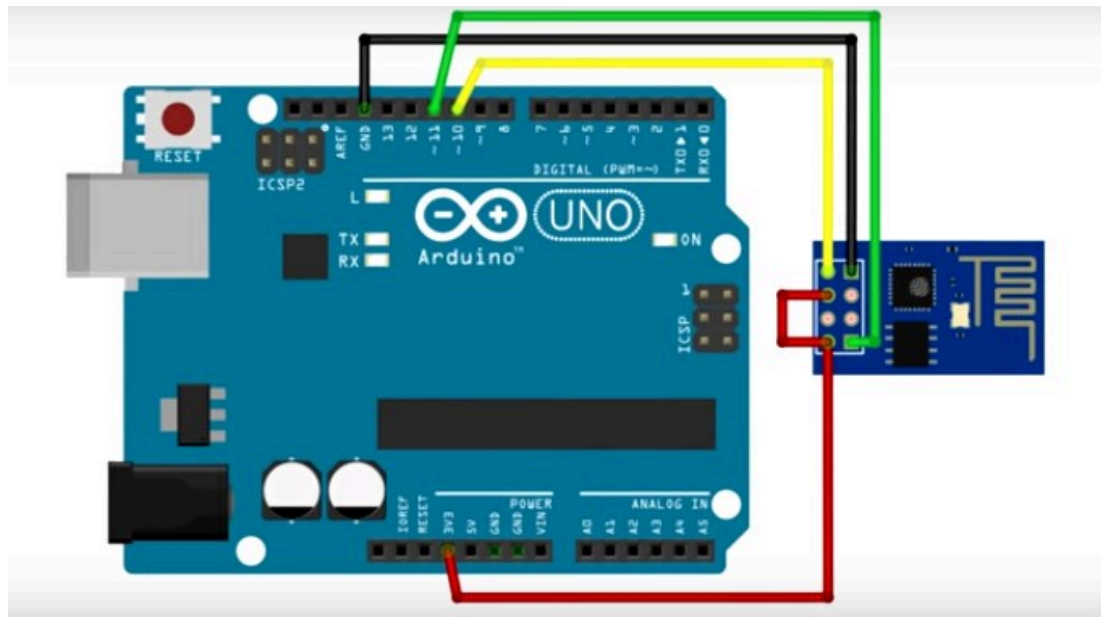
100.43.22.99

L'une est l'adresse IP en tant que serveur et l'autre en tant que client.

Il est possible aussi d'initialiser automatiquement le composant avec un programme. C'est ce qu'on va voir dans la suite.

Initialisation automatique du composant

Avant toute chose, vous devez modifier le câblage comme ci-dessous (Nous allons utiliser une deuxième liaison série) :



CODE

Le but du code est de créer et initialiser les deux liaisons séries et initialiser l'ESP 8266 pour transmettre les commandes AT et ensuite après une attente (timeout) on lit la réponse reçu.

```

#include <SoftwareSerial.h>
//permet de créer des interfaces série en utilisant les GPIOs.

// On crée une communication série software pour l'ESP8266
SoftwareSerial ESP8266(10, 11); // RX, TX

// SSID du réseau Wi-Fi (Garder les guillemets)
String NomduReseauWifi = "Entrez le nom de votre Box ou point d'accès Wifi";
// Mot de passe (Garder les guillemets)
String MotDePasse = "Entrez le mot de passe de votre Box ou point d'accès Wifi";

/**** INITIALISATION ****/

void setup()
{
  Serial.begin(9600);
  // On ouvre une communication serie hardware pour les messages destinés au
  // moniteur série avec le baudrate par défaut de l'ESP8266
  ESP8266.begin(9600);
  // On ouvre la communication serie software pour l'ESP8266
  initESP8266();
}

/**** Fonction qui initialise l'ESP8266 ****/

void initESP8266()
{
  Serial.println("*****");
  Serial.println("***** DEBUT DE L'INITIALISATION *****");
  Serial.println("*****");
  envoieAuESP8266("AT+RST"); // Transmission des commandes AT
  recoitDuESP8266(2000); //on lit les réponses reçu avec le timeout (2000)
  Serial.println("*****");
  envoieAuESP8266("AT+CWMODE=3");
  recoitDuESP8266(5000);
  Serial.println("*****");
  envoieAuESP8266("AT+CWJAP=\"" + NomduReseauWifi + "\",\"" + MotDePasse + "\"");
  recoitDuESP8266(10000);
  Serial.println("*****");
  envoieAuESP8266("AT+CIFSR");
  recoitDuESP8266(1000);
  Serial.println("*****");
  envoieAuESP8266("AT+CIPMUX=1");
  recoitDuESP8266(1000);
  Serial.println("*****");
  envoieAuESP8266("AT+CIPSERVER=1,80");
  recoitDuESP8266(1000);
}

```

```

Serial.println("*****");
Serial.println("***** INITIALISATION TERMINEE *****");
Serial.println("*****");
Serial.println("");
}

/** Fonction qui envoie une commande à l'ESP8266 */

void envoieAuESP8266(String commande)
{
  ESP8266.println(commande);
}

/** Fonction qui lit et affiche les messages envoyés par l'ESP8266 */

void recoitDuESP8266(const int timeout)
{
  String reponse = "";
  long int time = millis();
  while((time+timeout) > millis())
  {
    while(ESP8266.available())
    {
      char c = ESP8266.read();
      reponse+=c;
    }
  }
  Serial.print(reponse);
}

```

Votre carte arduino est maintenant connecté en Wifi.

Lexique

1 : liaison série

En communication série, on découpe l'information à transmettre en petits blocs de taille fixe avant de la transmettre. La taille des blocs correspond au nombre des lignes disponibles pour la transmission des données.

Ce type de communication s'oppose à la communication parallèle. En communication parallèle, il y a une ligne par bits à transmettre. Tous les bits sont donc transmis en même temps. Pour une même fréquence de communication, la communication parallèle est donc plus rapide.

L'avantage de la communication série sur la communication parallèle est qu'elle nécessite moins de lignes, donc moins de broches, donc moins de composants. Son coût est donc plus faible.

2 : baud

unité de rapidité de modulation en télégraphie, en communication de données et en transmission numérique ; la rapidité de modulation d'un signal, exprimée avec cette unité, est égale à l'inverse de la durée en secondes du plus court élément du signal, ou de l'intervalle unitaire dans un signal numérique composé d'éléments de signal de durée constante.

3 : commande AT

Votre ESP-01 est configuré par défaut avec un firmware qui embarque un langage de commandes interactif que l'on utilise sur la voie série. Ce langage de commande est spécifique à votre module et est connu sous le nom de Commandes Hayes ou commandes AT. Les Commandes Hayes, parfois appelées Commandes AT, constituent un langage de commandes développé à l'origine pour le modem Hayes Smartmodem 300. Ce jeu de commandes s'est ensuite retrouvé dans quasiment tous les modems produits.

4 : TCP

protocole de transmission utilisé sur les réseaux IP. Il est, dans le modèle Internet, rarement cité seul, mais bien souvent en compagnie de l'acronyme IP pour former la suite de protocoles TCP/IP.

Liens utiles

Quelques commandes AT

<https://caron.ws/diy-cartes-microcontroleurs/les-commandes-at/>

Pour ceux qui s'y connaissent en langage html

<http://allaboutee.com/2014/12/30/esp8266-and-arduino-webserver/>

tutoriel : affichage vidéo

<http://zeflo.com/2014/esp8266-weather-display/>

Comment utiliser l'ESP 8266 et l'arduino comme un serveur Web

<http://allaboutee.com/2014/12/30/esp8266-and-arduino-webserver/>