

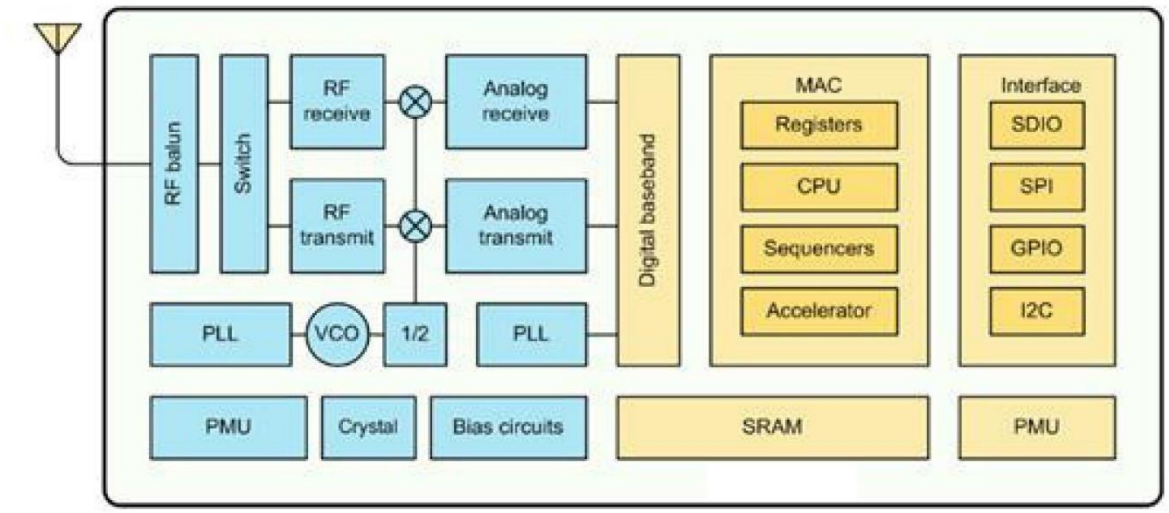
Mise en œuvre
Des modules ESP8266
Et ESP32

Mise en œuvre du module ESP8266 et ESP32

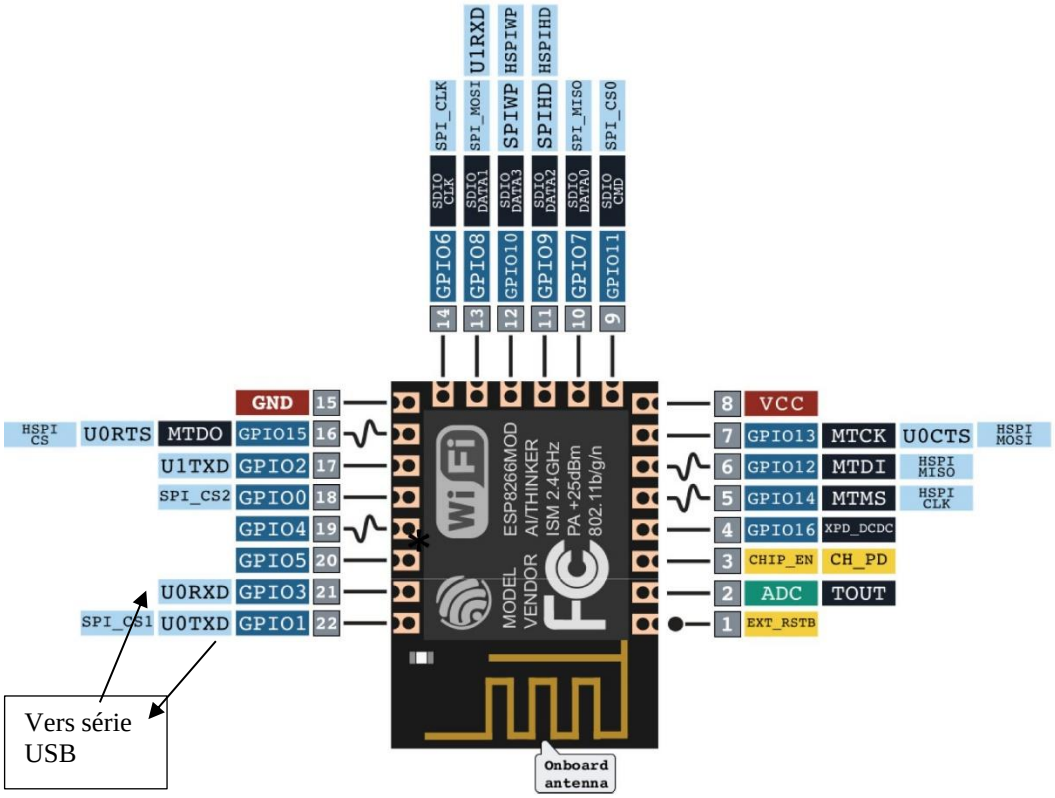
Ce document présente la mise en œuvre logicielle du module ESP8266 ou ESP32, le flashage et la programmation du module en commande AT ou en langage LUA

Présentation du module ESP8266

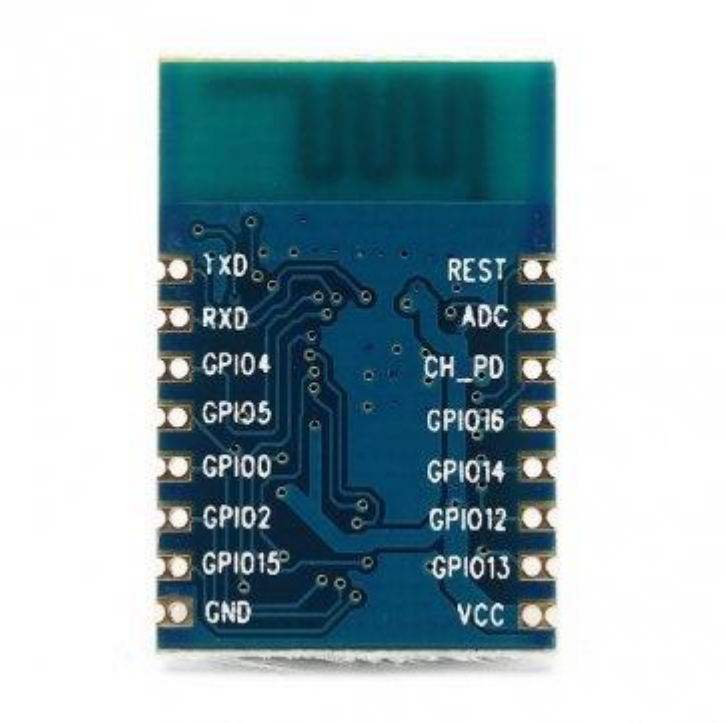
Le module ESP8266 est l'association sur une même puce d'un SOC (system on chip et d'un module WIFI). Sa constitution interne est illustrée ci-dessous :



Le module WIFI est du type 2.4GHz b,g,n WPA/WPA2.
Sa mémoire flash peut aller de 512 Koctets à 16 Moctets.
Son point fort est pouvoir s'adapter à des périphériques externes SPI, I2C, GPIO ….



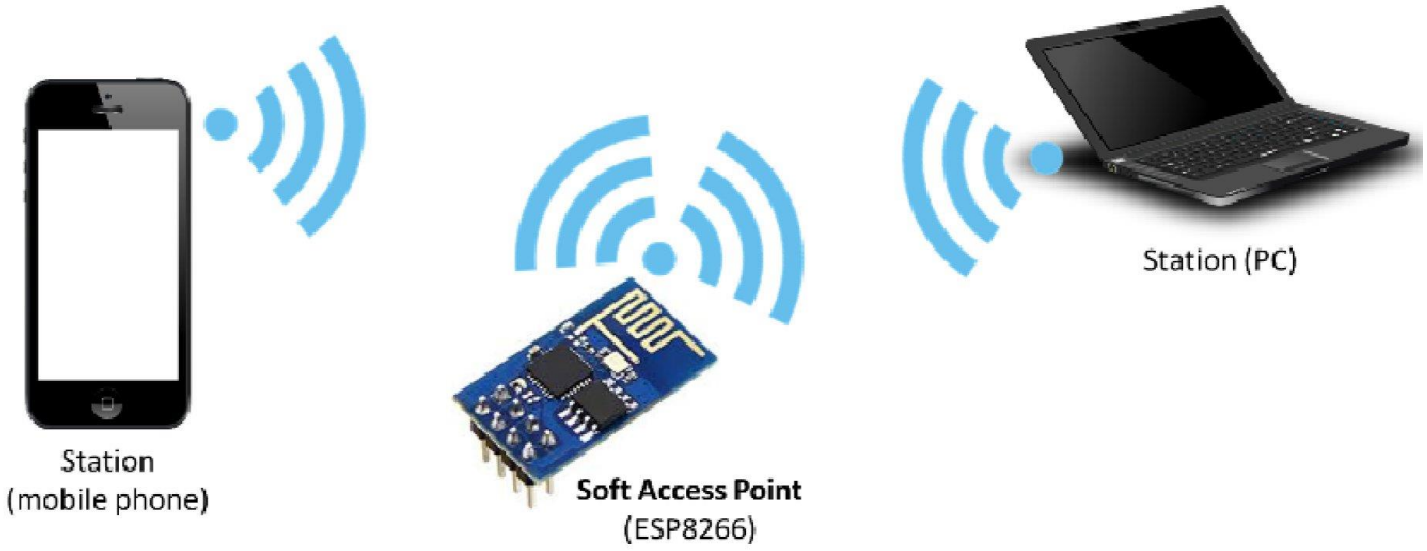
Le module utilisé ESP-07 est un peu plus limité en entrée sortie :



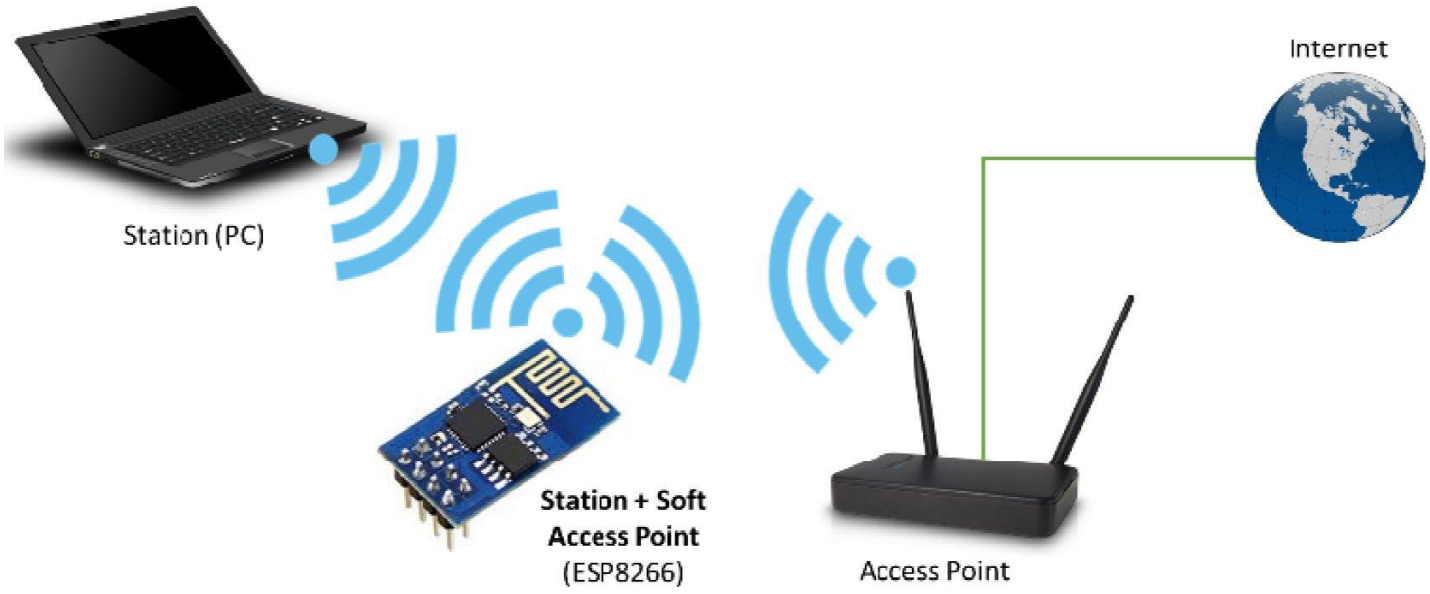
Certaines broches peuvent partager plusieurs fonctions :

GPIO4	PWM3,SDA
GPIO12	PWM0,HSPI_MISO
GPIO14	PWM2,HSPI_CLK
GPIO15	PWM1, HSPI_CS
GPIO13	HSPI_MOSI
GPIO6	
ADC	TOUT
GPIO5	SCL

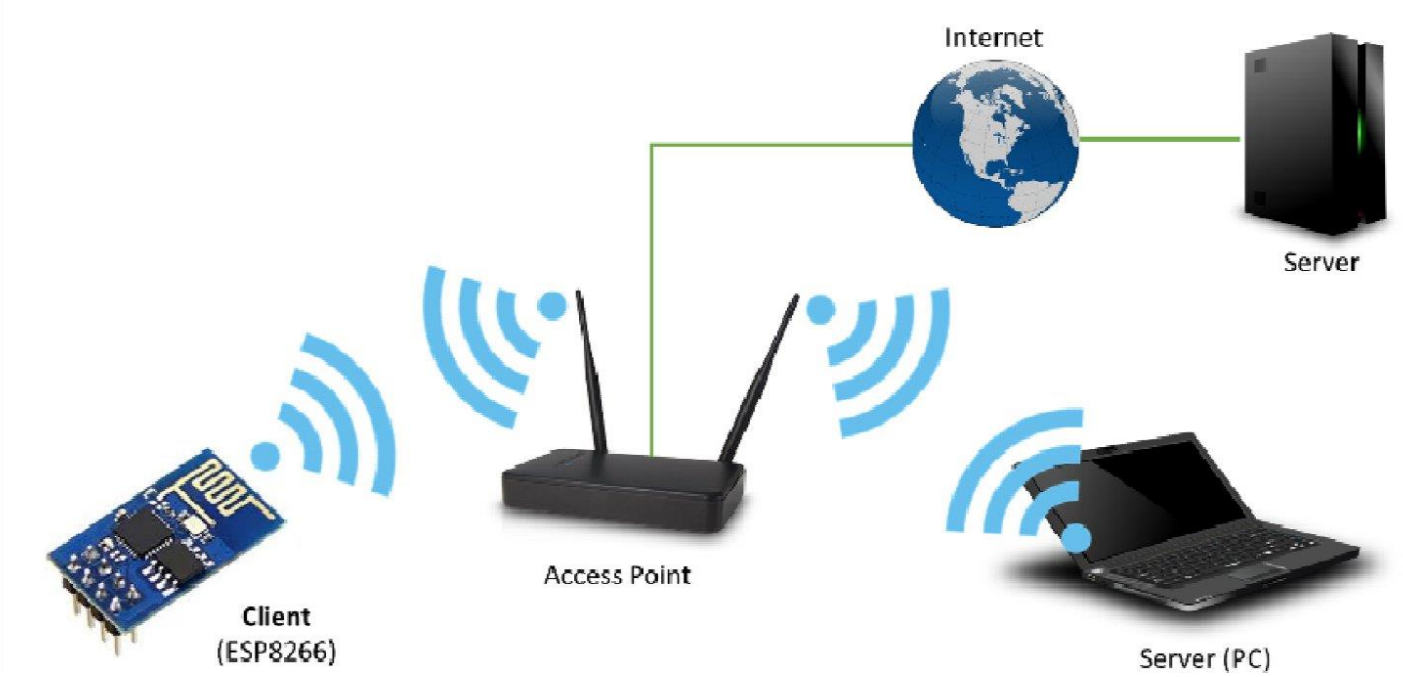
En mode WIFI le module peut travailler sous 3 modes :



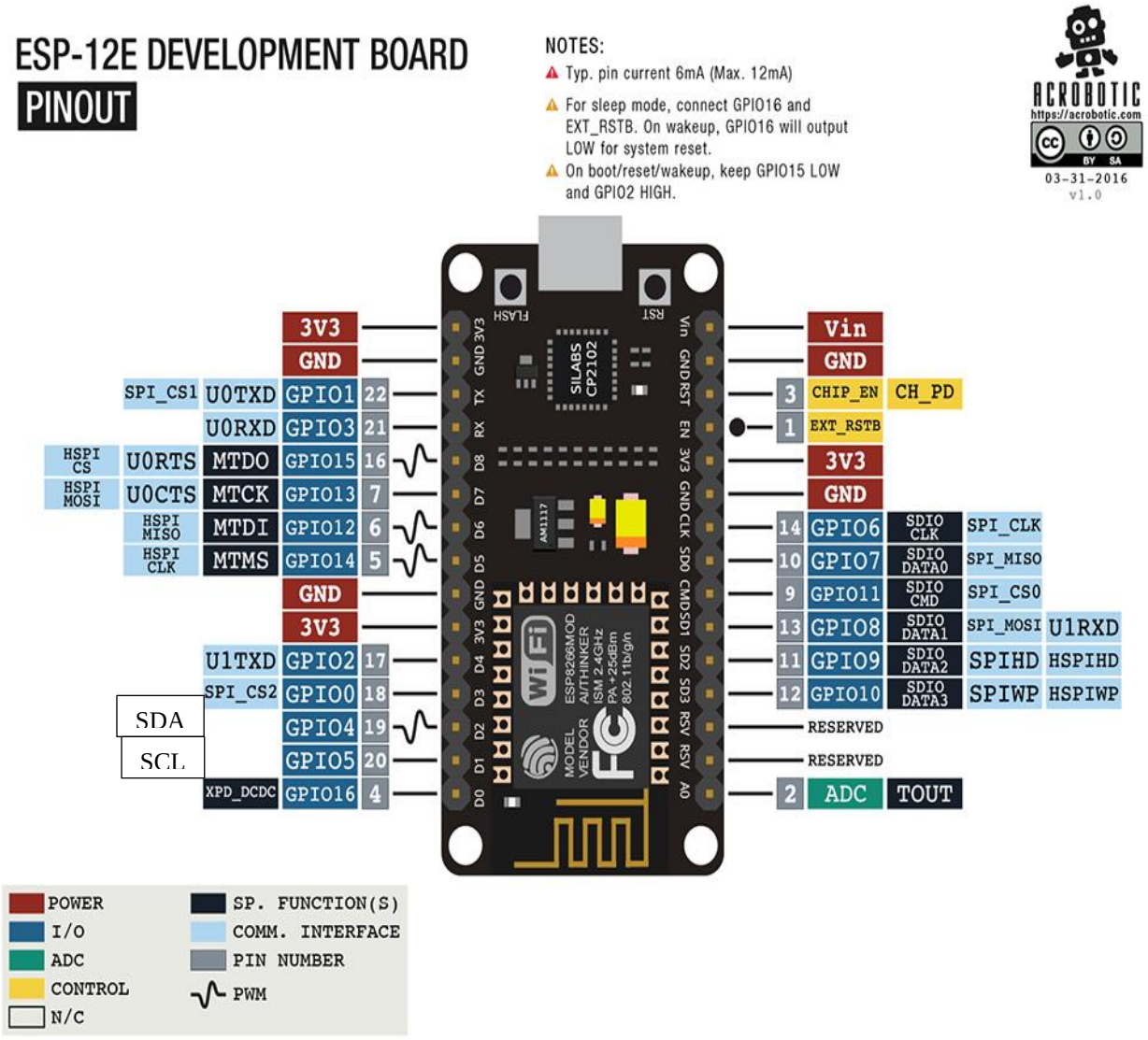
Mode Point d'accès (soft AP)



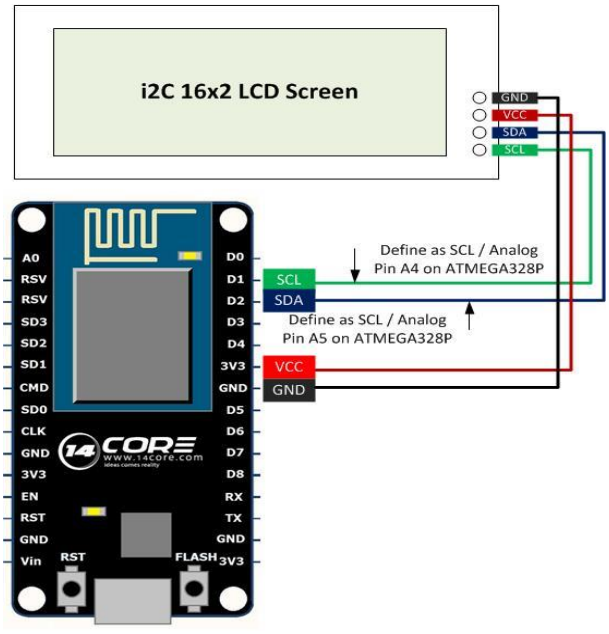
Station plus soft AP



station

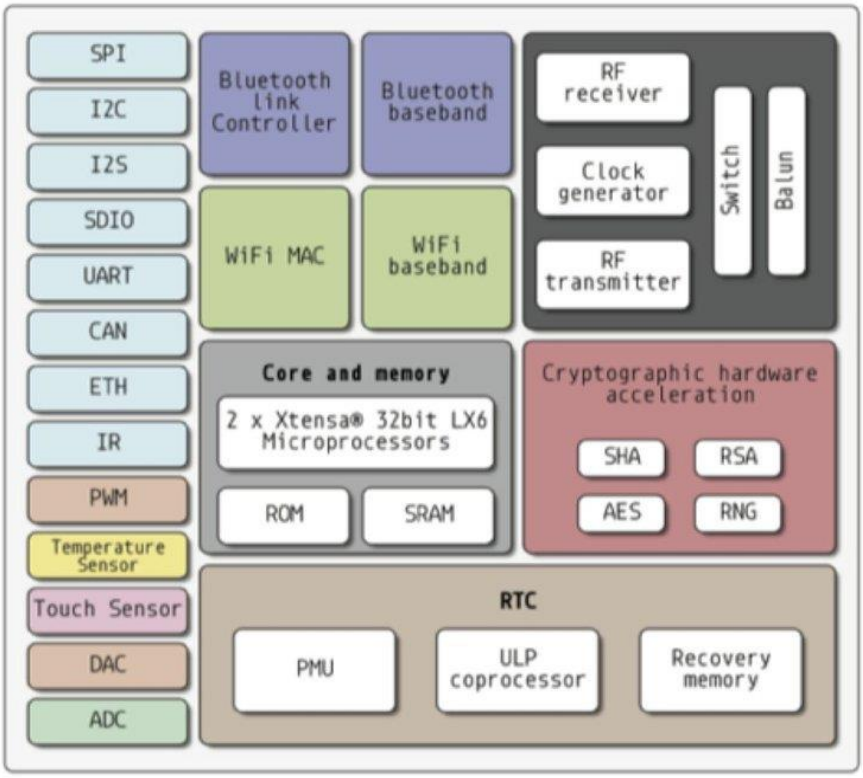


Le module choisi est le AZDelivery NodeMCU Lua Amica Module V2 ESP8266 ESP-12[E].



Le module ESP32

Ce module est le modèle qui a précédé l'ESP8266, son architecture est fournie ci-dessous :



Le tableau ci-dessous nous donne un comparatif entre les deux modèles :

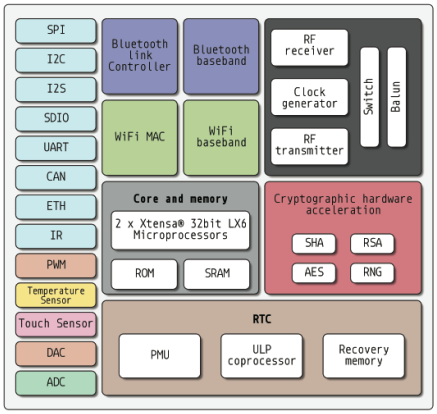
Specifications	ESP8266	ESP32
MCU	Xtensa® Single-Core 32-bit L106	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	None	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash , up to 16 MBytes	SPI Flash , up to 16 MBytes
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	None	1
Ethernet MAC Interface	None	1
Touch Sensor	None	Yes
Temperature Sensor	None	Yes
Working Temperature	-40℃ ~ 125℃	-40℃ ~ 125℃

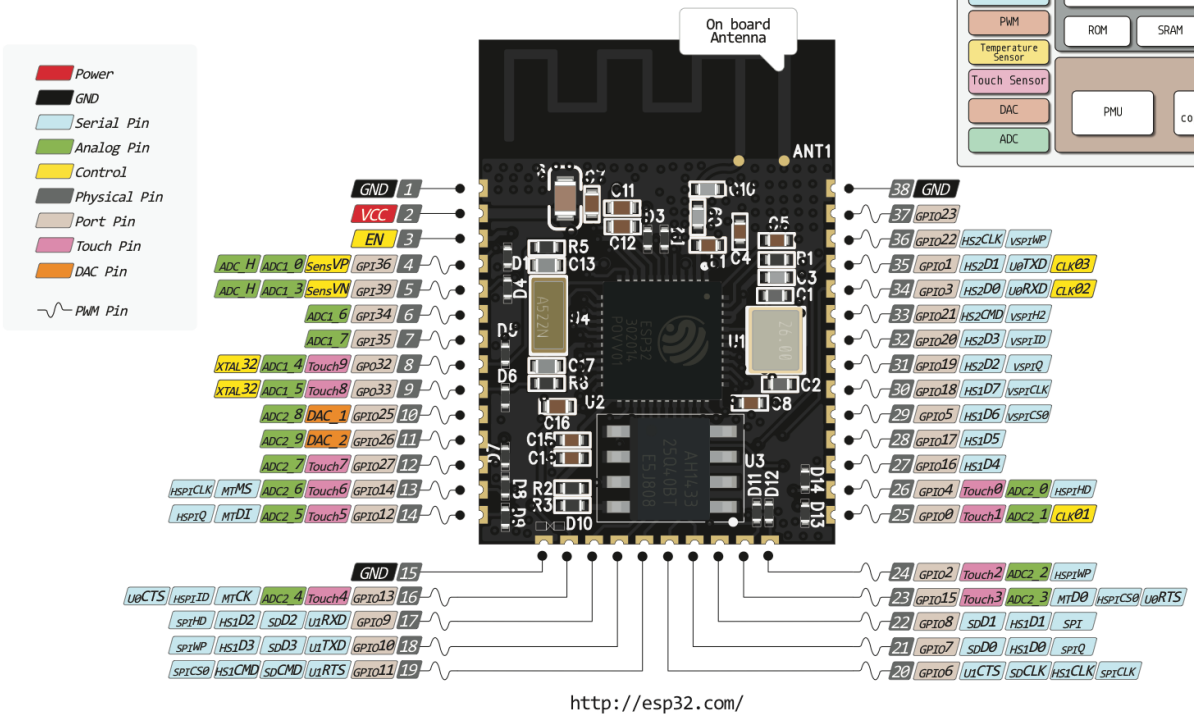
ESP32 Dev Board PINMAP

					3.3V	GND							
(pu)				RESET	EN	GPIO23	VSPI MOSI						SPI MOSI
SVP			ADC0		GPIO36	GPIO22							Wire SCL
SVN			ADC3		GPIO39	GPIO1	TX0						Serial TX
			ADC6		GPIO34	GPIO3	RX0						Serial RX
			ADC7		GPIO35	GPIO21							Wire SDA
		TOUCH9	ADC4		GPIO32	GND							
		TOUCH8	ADC5		GPIO33	GPIO19	VSPI MISO						SPI MISO
DAC1			ADC18		GPIO25	GPIO18	VSPI SCK						SPI SCK
DAC2			ADC19		GPIO26	GPIO5	VSPI SS					(pu)	SPI SS
		TOUCH7	ADC17		GPIO27	GPIO17							
	TMS	TOUCH6	ADC16	HSPI SCK	GPIO14	GPIO16							
(pd)	TDI	TOUCH5	ADC15	HSPI MISO	GPIO12	GPIO4		ADC10	TOUCH0		(pd)		
					GND	GPIO0	BOOT	ADC11	TOUCH1		(pu)		
	TCK	TOUCH4	ADC14	HSPI MOSI	GPIO13	GPIO2		ADC12	TOUCH2		(pd)		
				FLASH D2	GPIO9	GPIO15	HSPI SS	ADC13	TOUCH3	TDO	(pu)		
				FLASH D3	GPIO10	GPIO8	FLASH D1						
				FLASH CMD	GPIO11	GPIO7	FLASH D0						
					5V	GPIO6	FLASH SCK						









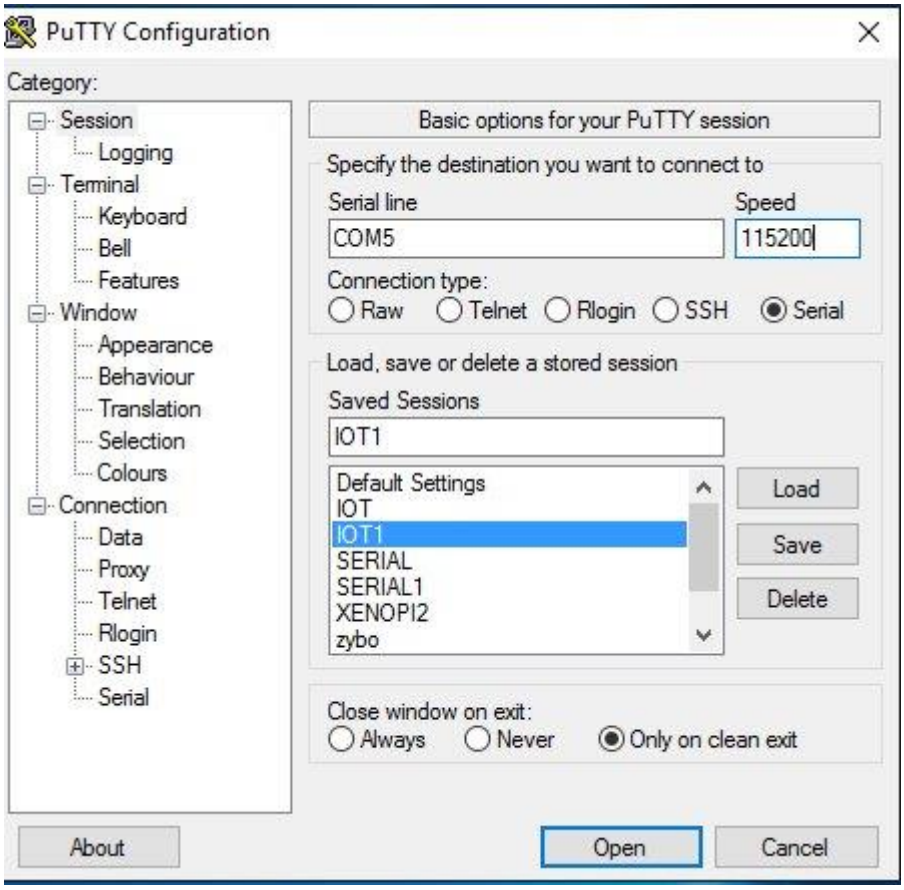
http://esp32.com/

bq
10 DEC 2015
ver 1 rev 3

Programmation en LUA avec nodeMCU

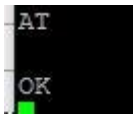
1) Connexion série vers le module ESP8266

Se connecter sur putty avec les paramètres suivants :



Lancez la première commande AT pour voir si le module répond ok :

AT



1.1) Modification du SSID et du password

AT+CWSAP= »ESP », »password »,3,0
Le SSID est ESP, le mot de passe password, le canal le N°3, le dernier chiffre 0 indique aucun chiffrement . Il faut fournir un password même s'il n'y a pas de chiffrement.
Pour le dernier chiffre

0	aucun
1	WEP
2	WPA
3	WPA2-PSK

On ne peut obtenir l'@IP du module ?
AT+CIFSR ne donne rien (sans doute le firmware non pris en charge avec la version initiale) ?

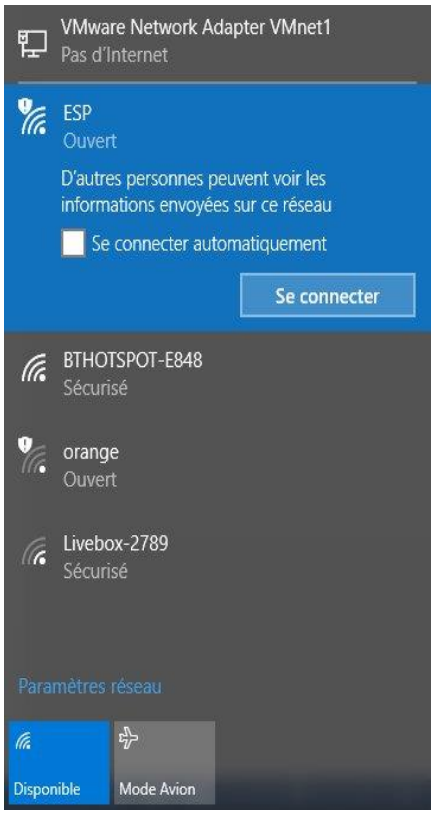
```
AT+CIFSR

ERROR
```

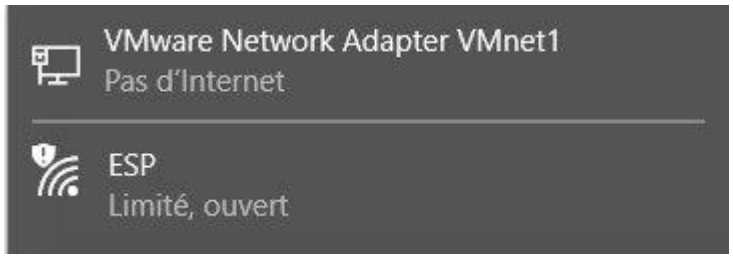
On doit redémarrer le module
AT+RST

1.2)Connexion IP vers le module

Connectons-nous sur le module.



La connexion aboutit :



Lançons une fenêtre terminale MSDOS pour vérifier notre connexion.

La commande IPCONFIG

```
Carte réseau sans fil Wi-Fi :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .: fe80::dd8d:8271:c74e:e324%6
   Adresse IPv4. . . . . . . . . . . . . .: 192.168.4.102
   Masque de sous-réseau. . . . . . . . . : 255.255.255.0
   Passerelle par défaut. . . . . . . . . : 192.168.4.1
```

Notre @IP sur l'ordinateur est 192.168.4.102
La passerelle du point d'accès WIFI est de 192.168.4.1

1.3) Mise en place d'un point d'accès et d'un serveur TCP sur ESP8266

En commande AT on a :

AT+CWMODE=3 -- pour se positionner en point d'accès
AT+CWSAP? -- pour obtenir le login et le password
SERVEUR TCP
AT+CIPMUX=1 -- de multiples connexions sont possibles
AT+CIPSERVER=1 -- port 333 par défaut si vous ne mettez rien
AT+CWLIF -- on obtient les systèmes informatiques connectés sur ESP8266

1.4) Mise en place d'un client TCP sur l'ordinateur

Lancez eclipse et le projet ESP8266SERVERTCP

Lancez le programme suivant sous eclipse :

```
import java.net.*;
import java.io.*;
import java.util.Scanner;

public class TestClientTcpbyte {
	final static int port = 333;
	static boolean stop=false;
	public static void main(String[] args) {

		Socket socketClient; DataInputStream userinput;
		PrintStream theOutputStream;

		try  {
		//InetAddress serveur = InetAddress.getByName(args[0]);
		socketClient = new Socket("192.168.4.1", port);

		DataOutputStream fluxReseau = new DataOutputStream(new
BufferedOutputStream(socketClient.getOutputStream()));
		fluxReseau.flush();
		DataInputStream fluxinputReseau = new DataInputStream(new
BufferedInputStream(socketClient.getInputStream()));
		System.out.println("je suis ici1");
		while(stop==false){
		//byte []  octets = new byte[6];
byte[] octets = new byte[1024];
int longueur =fluxinputReseau.read(octets);
System.out.printf("la taille est de %d  ",longueur);
if (longueur>0){
		for (int j=0;j<longueur;j++)
		{
		  System.out.printf("%x ",octets[j]);
		  if (octets[j]==0x00) stop=true;

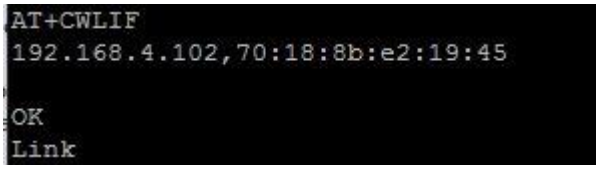
		}
		//fluxReseau.writeInt(longueur);
		fluxReseau.write(octets,0,longueur);
		fluxReseau.flush();
		//fluxinputReseau.flush();
		}
			}
		socketClient.close();
```

```
            }
            catch (Exception e) {
            e.printStackTrace();}

    }

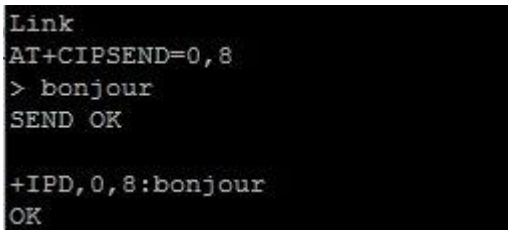
}
```

A noter l'@IP de la passerelle 192.168.4.1 et le numéro de port 333 le module ESP répond LINK



Lancez alors sur putty avec la commande qui permet l'envoi de 8 octets:

AT+CIPSEND=0,8 -- liaison 0 8 octects envoyés



Le prompt > s'ouvre tapez votre chaine de caractère ici bonjour

Le module reçoit alors la réponse +IPD avec bonjour.

2) Flash ou mise à jour du firmware du module ESP8266

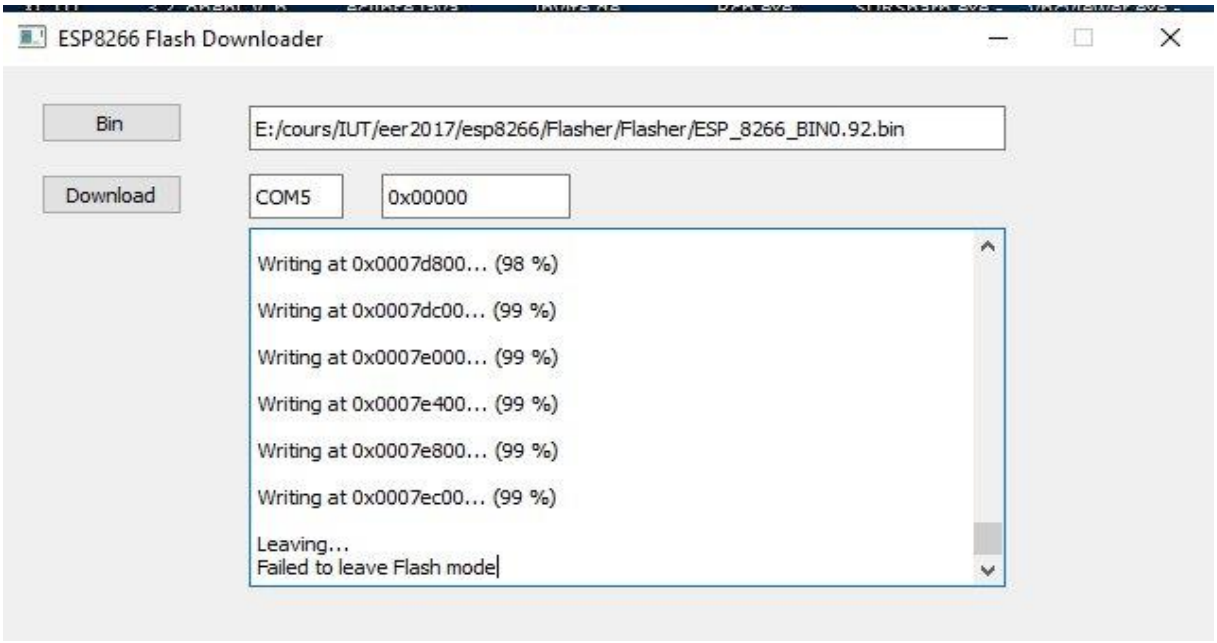
L'état de certaines broches de l'ESP8266 va permettre de définir son fonctionnement :

	GPIO 0	GPIO 2	GPIO 15	
Flash Startup (Normal)	1	1	0	Fonctionnement normal sur flash interne au démarrage
UART Download Mode (Programming)	0	1	0	Mode flash ou programmation
SD-Card Boot	0	0	1	

Pour changer le firmware positionnez la broche GPIO0 à 0

fermez putty éteindre le module puis relancez le afin qu'il démarre en mode flash

Puis lancez le logiciel ESP8266 flash downloader



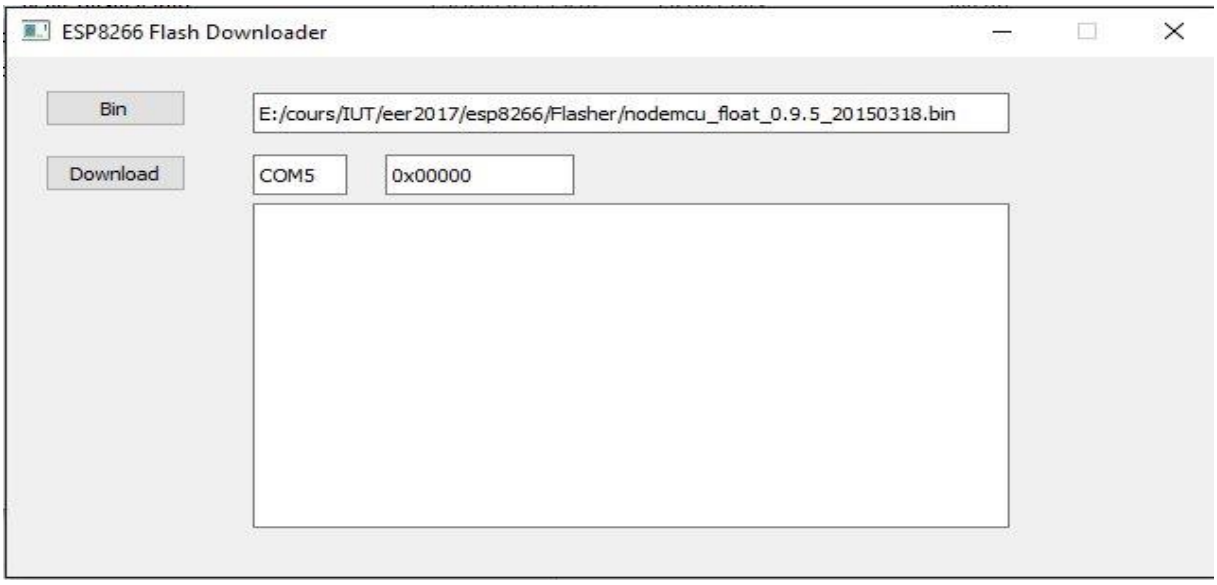
Si le téléchargement s'arrête en plein milieu recliquez sur le bouton download
Ne pas tenir compte du message d'erreur à la fin (Seul le firmware ci-dessus fonctionne avec le chip implanté sur le modul firmware d'origine)
Enlevez la masse sur la broche GPIO0 puis éteindre le module et relancez-le pour la prise en compte des commandes AT.

3) Utilisation de l'interpréteur LUA

Il est nécessaire de reprogrammer le firmware du module afin qu'il puisse interpréter les commandes LUA

3.1) Flash du module ESP8266

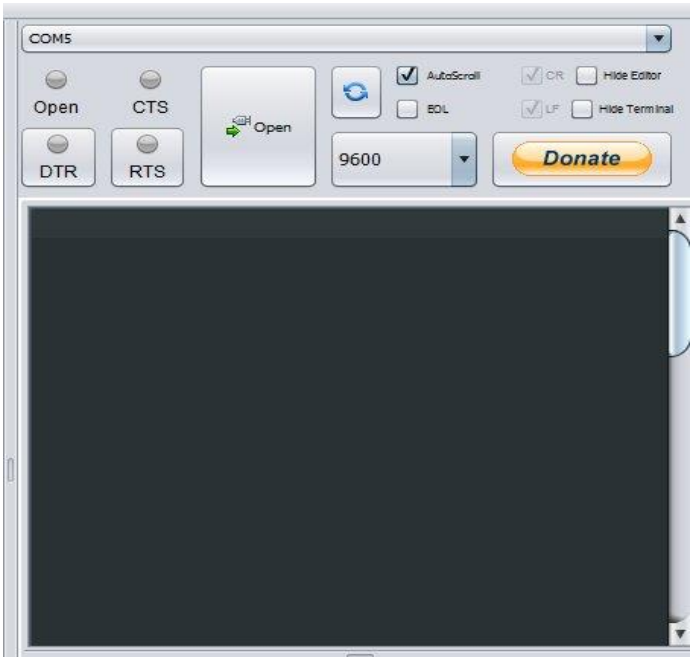
Afin de pouvoir bénéficier d'un interpréteur de commande LUA on doit modifier le firmware du ESP8266. Lançons le logiciel ESP8266 flash downloader en n'oublions pas de mettre la broche GPIO0 à 0 et d'éteindre et de rallumer le module. Attention partir toujours d'un firmware à base de commante AT

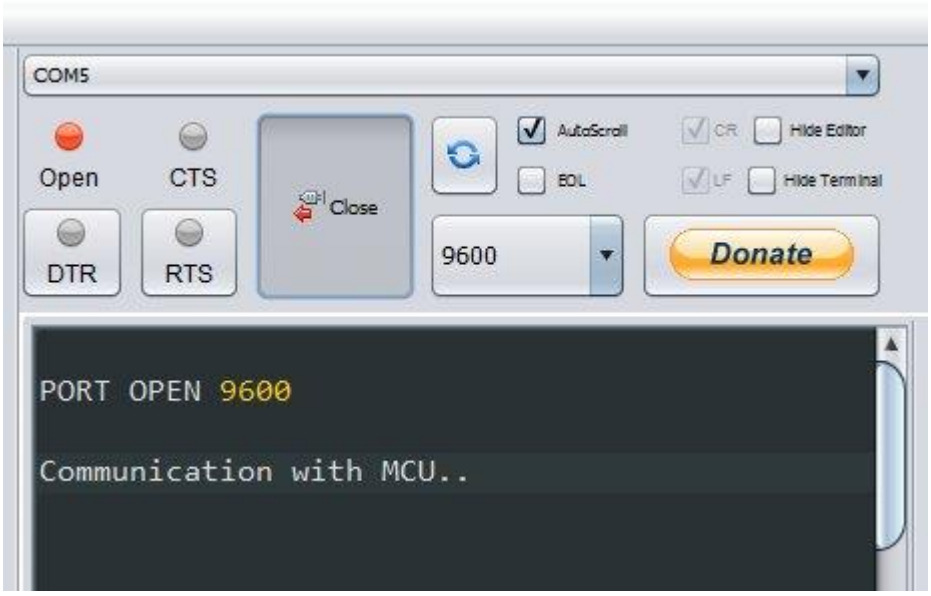


Enlevez la masse sur la broche GPIO0 éteindre rallumer le module.

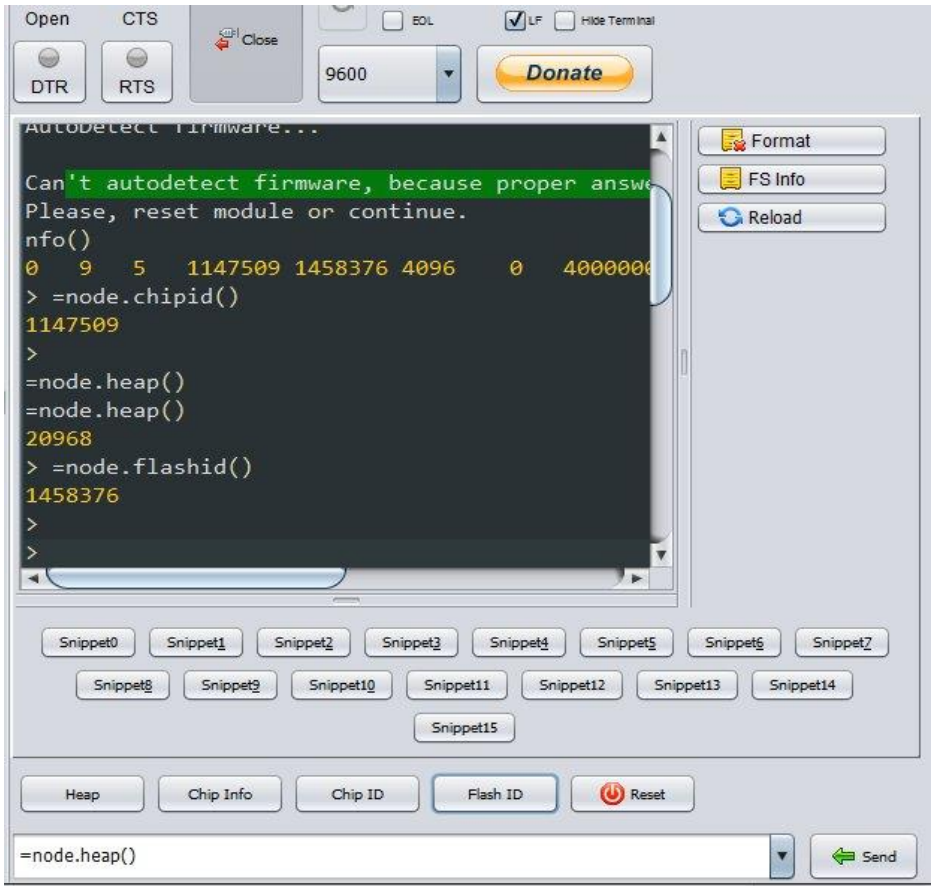
3.2) Test le l'interpréteur LUA

Lancez le logiciel explorer et réglez le débit à 9600bit/s plus cliquez sur open



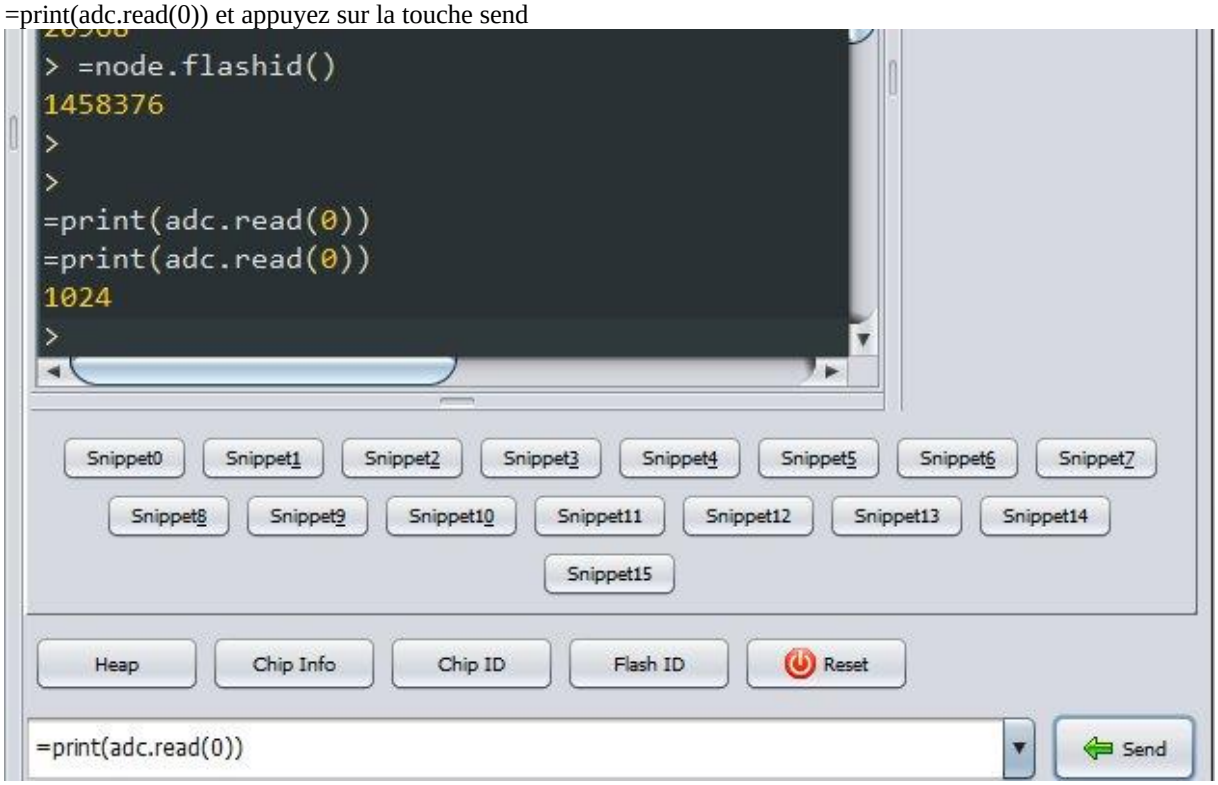


Cliquez sur chipid



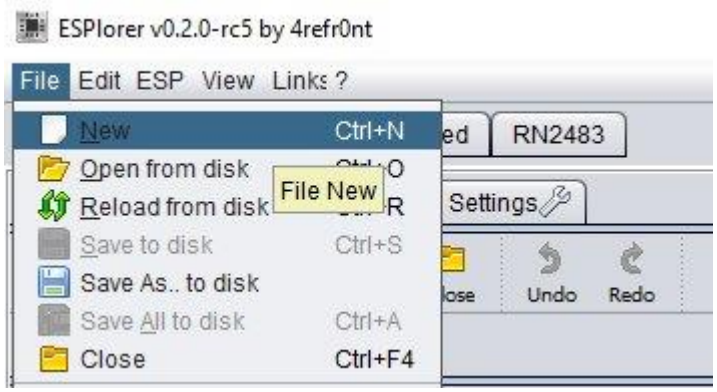
Le menu du bas se débloque on peut alors tapez une instruction LUA.

L'instruction suivante permet d'obtenir le résultat de la conversion analogique numérique de la tension présente sur la broche ADC0.
=print(adc.read(0)) et appuyez sur la touche send

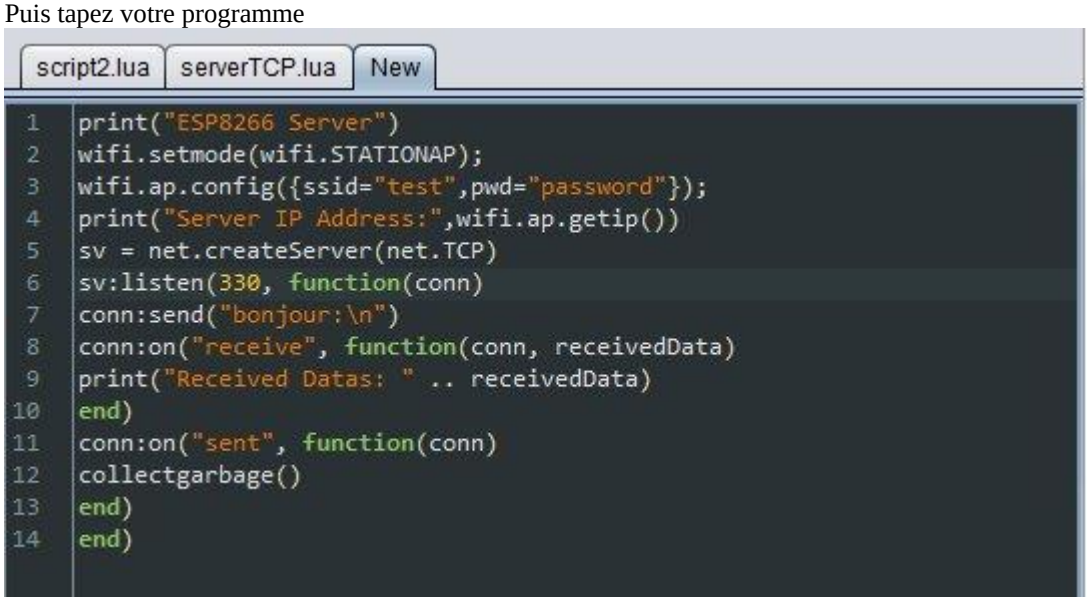


Il est possible de sauvegarder un programme, le compiler et l'exécuter sur le module :

Faire file→new

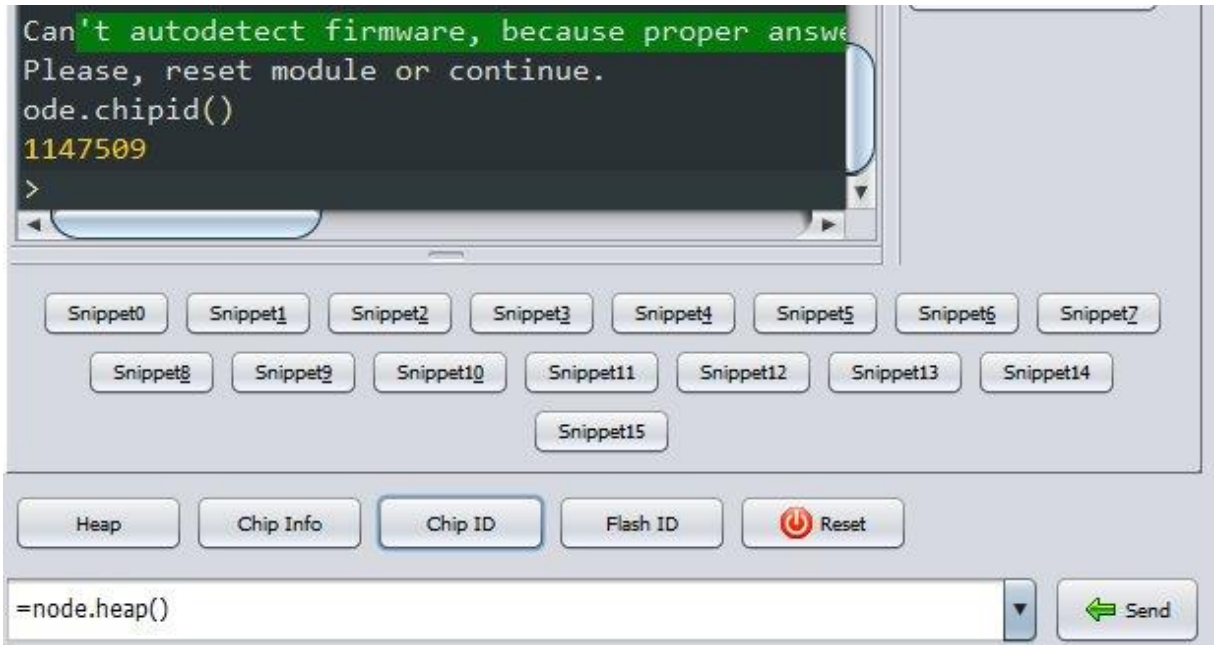


Puis tapez votre programme

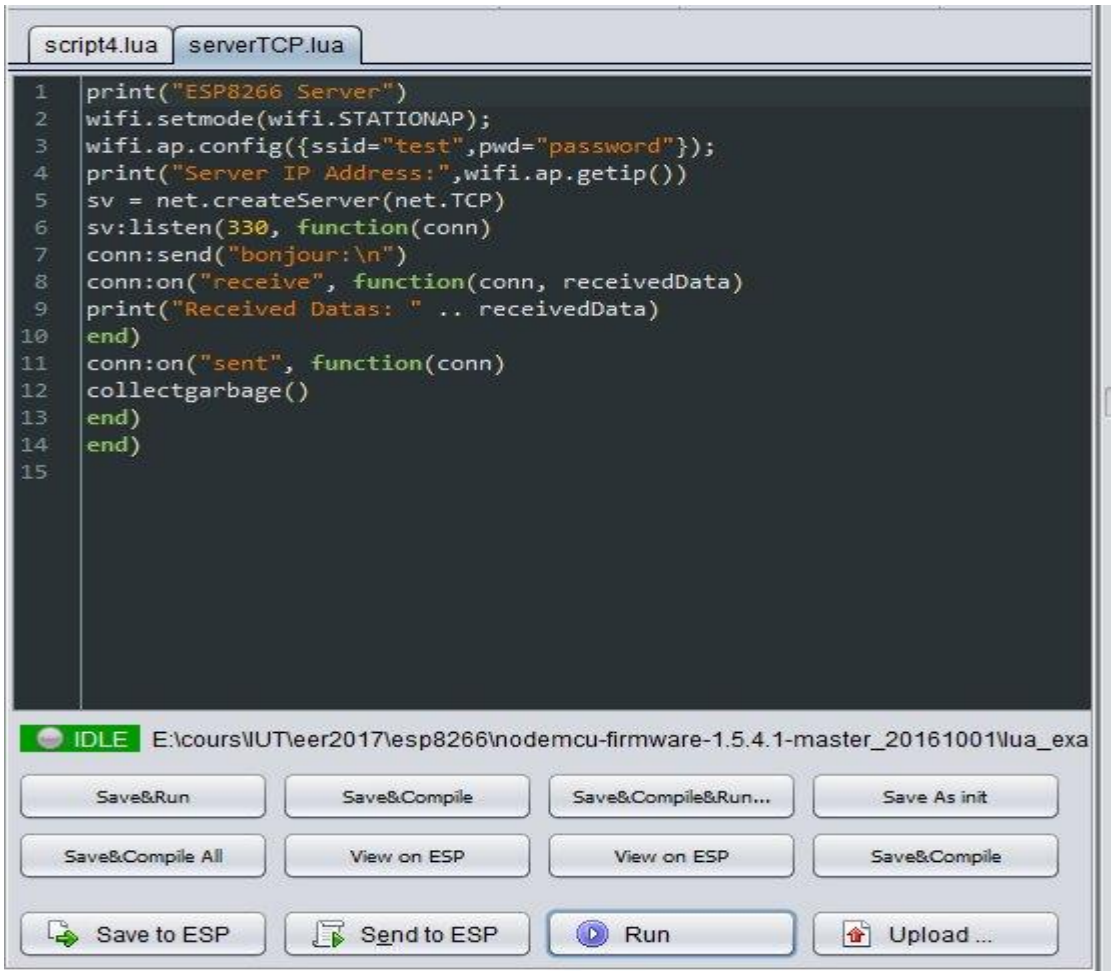


Puis cliquez sur save

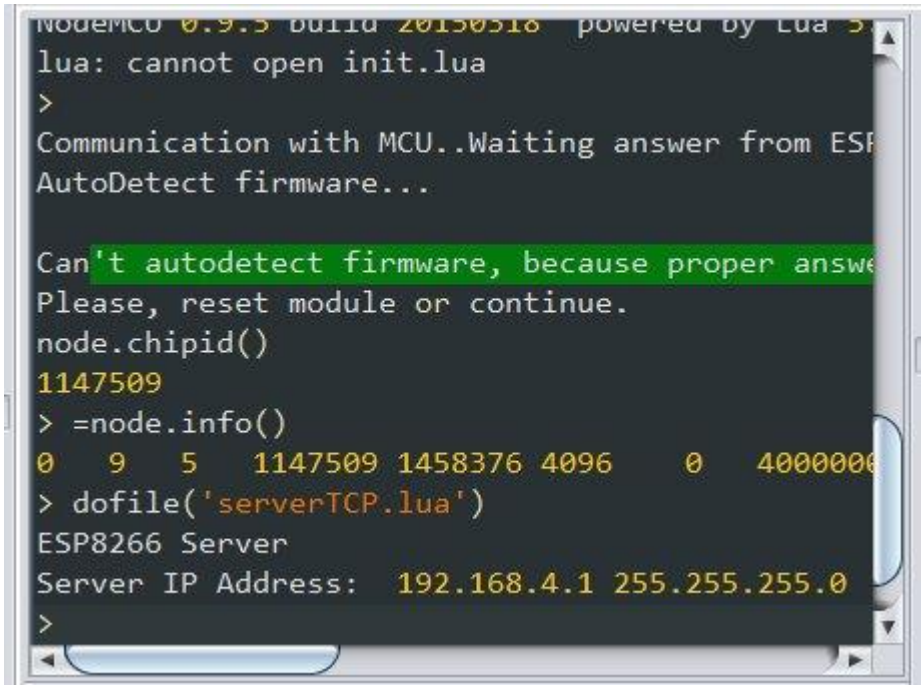
Pour débloquer le mode run cliquez sur chip ID



Tapez le nom du fichier source et cliquez sur run



La console nous indique l'état du programme :



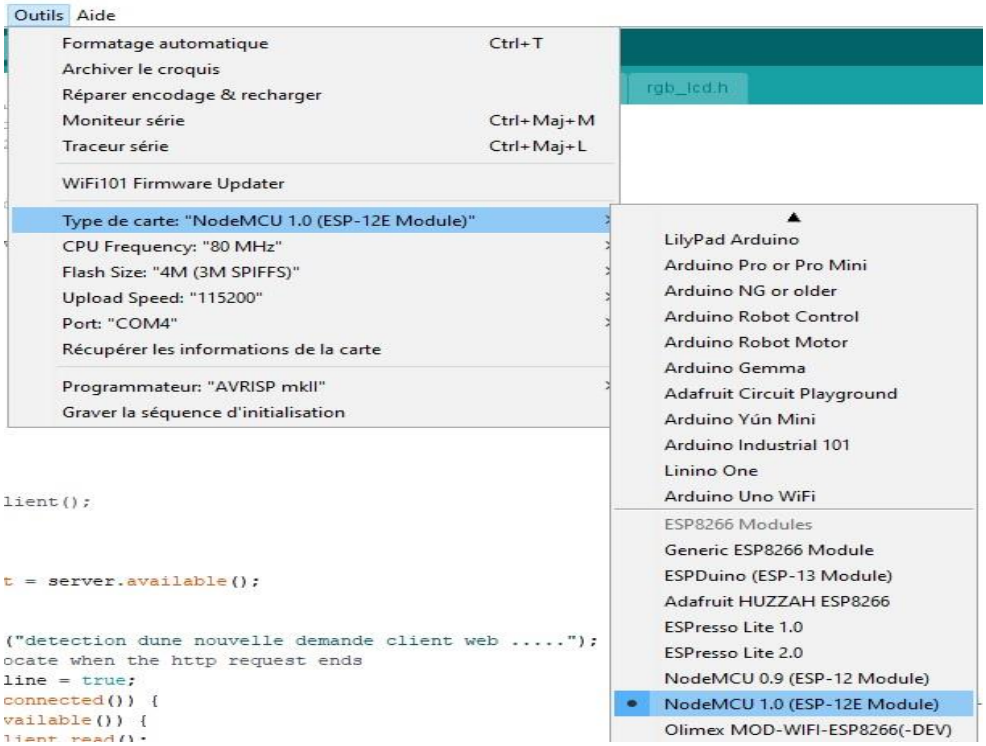
Un fichier init.lua peut être créé et lancer à chaque démarrage du module. Pour cela baptisé votre fichier source init.lua et cliquez sur save to esp



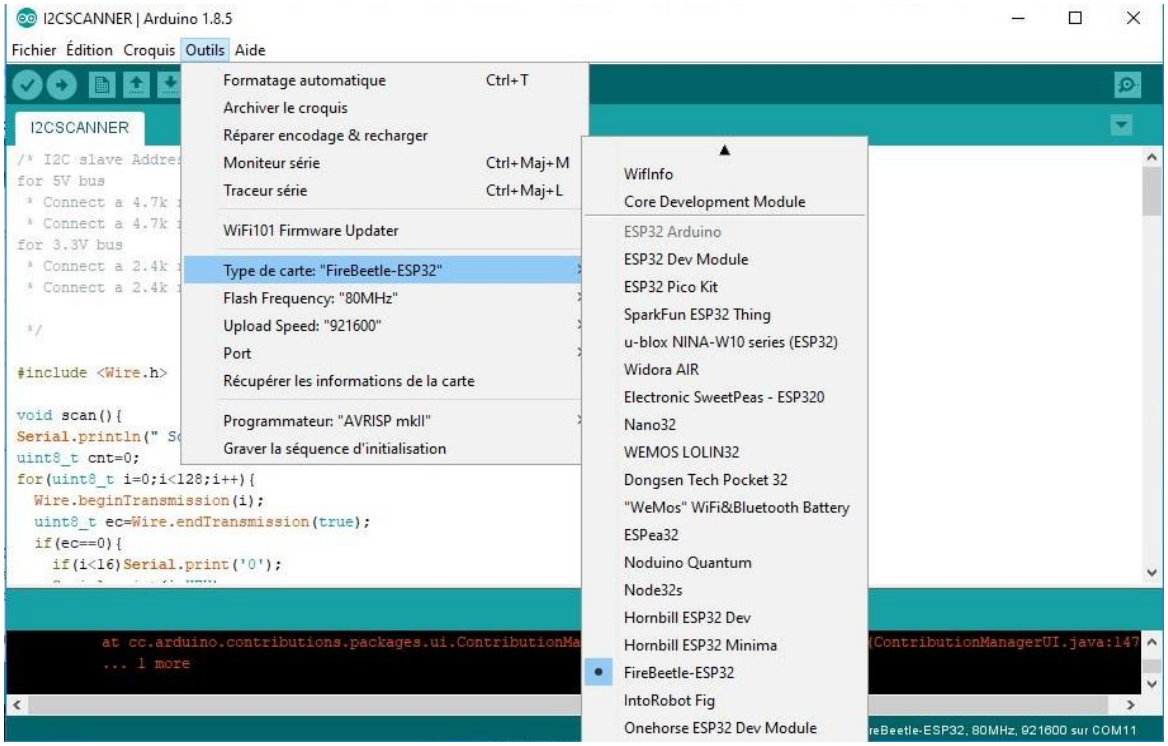
Au démarrage le module travaille avec le fichier init.lua de manière autonom

4) Utilisation de l'interface Arduino IDE

Ouvrir l'interface ARDUINO. Si les packages pour le module ESP8266 ou ESP32 ont été installés allez dans outils→ type de cartes (choisir le module NodeMCU 1.0 (ESP-12E Module) pour l'ESP8266 carte intégré AZDelivery NodeMCU Lua Amica Module V2 ESP8266 ESP-12), generic ESP8266 dans les autres cas.



(choisir le module Firebeetle pour l'ESP32)



Choisir le port COM virtuel sur lequel est branché le module :

