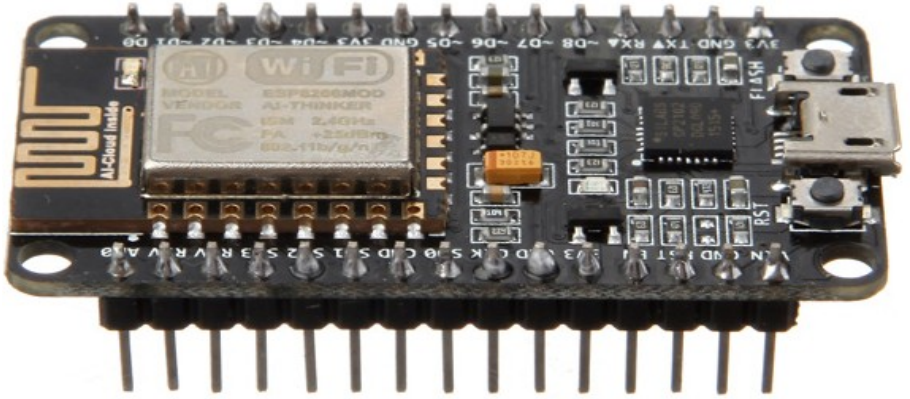
ESP8266

ESP-12E Development Board



Jordi Binefa i Martínez

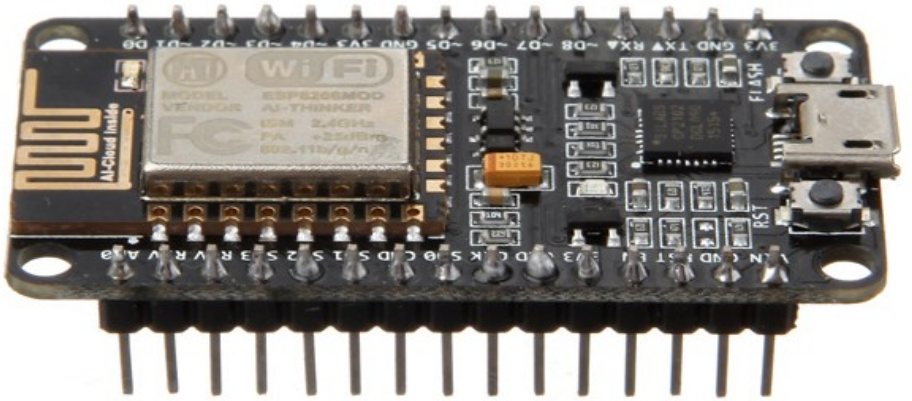
Responsable d'R+D+i a electronics.cat

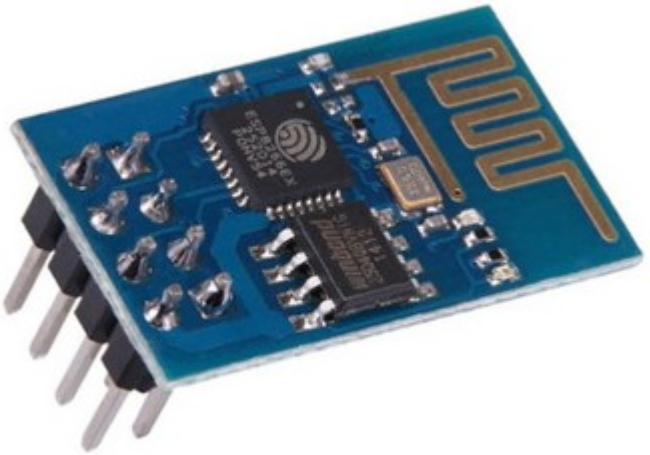
Professor de cicles formatius a JESUÏTES El Clot Escola del Clot

novembre de 2015

ESP-12E Development Board

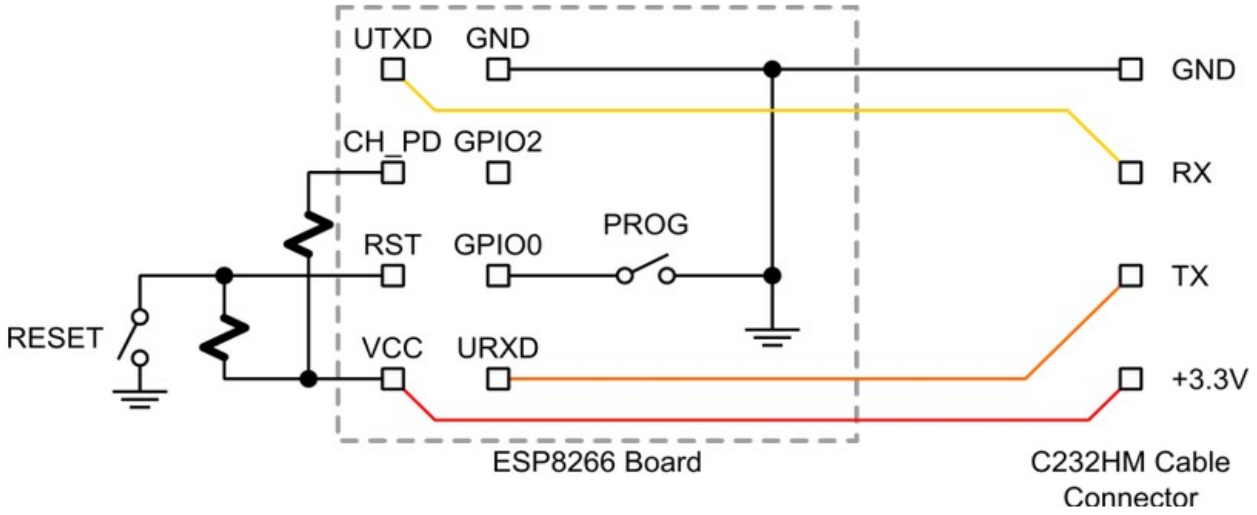
Algunes plaques amb l'ESP8266

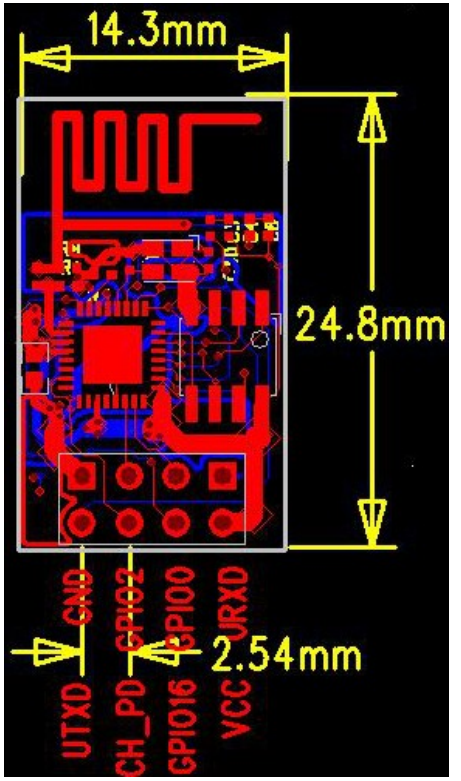


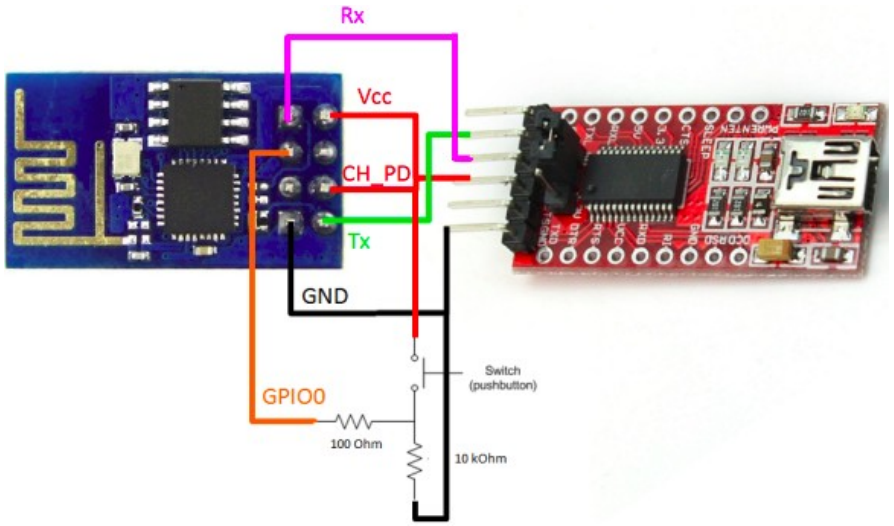


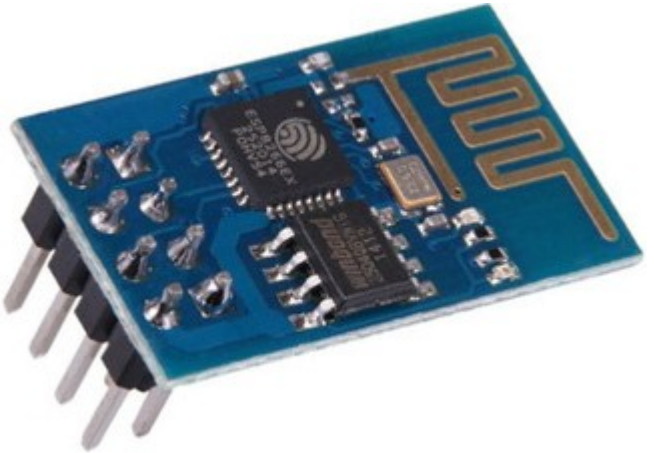
ESP-12E Development Board

Programant l'ESP8266 - ESP01



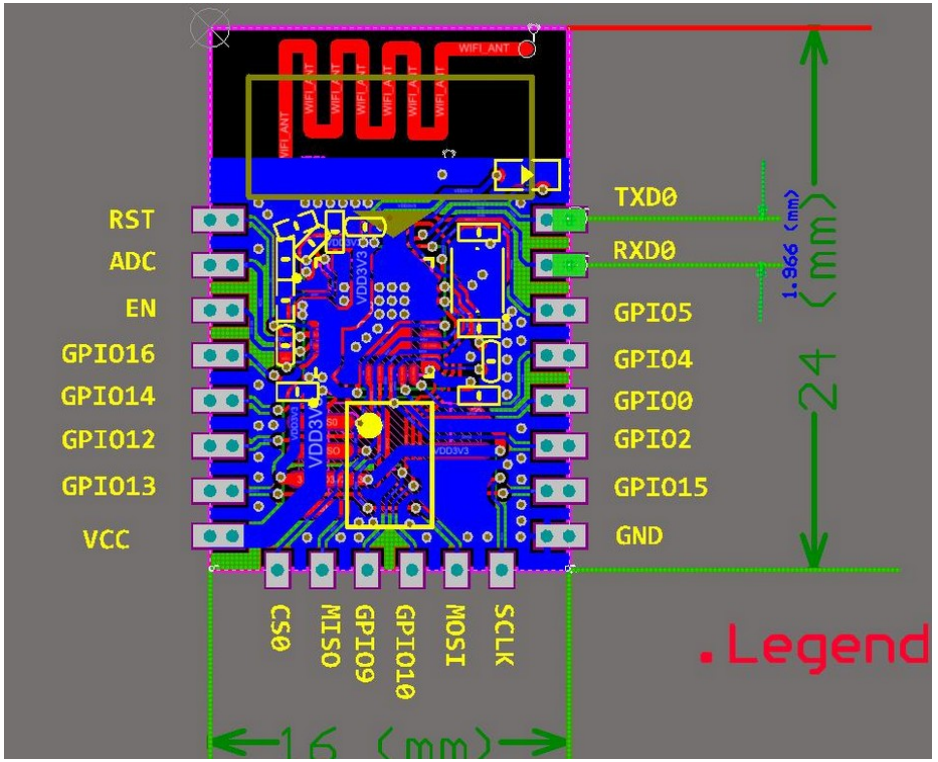






ESP-12E Development Board

Programant l'ESP8266 - ESP12E



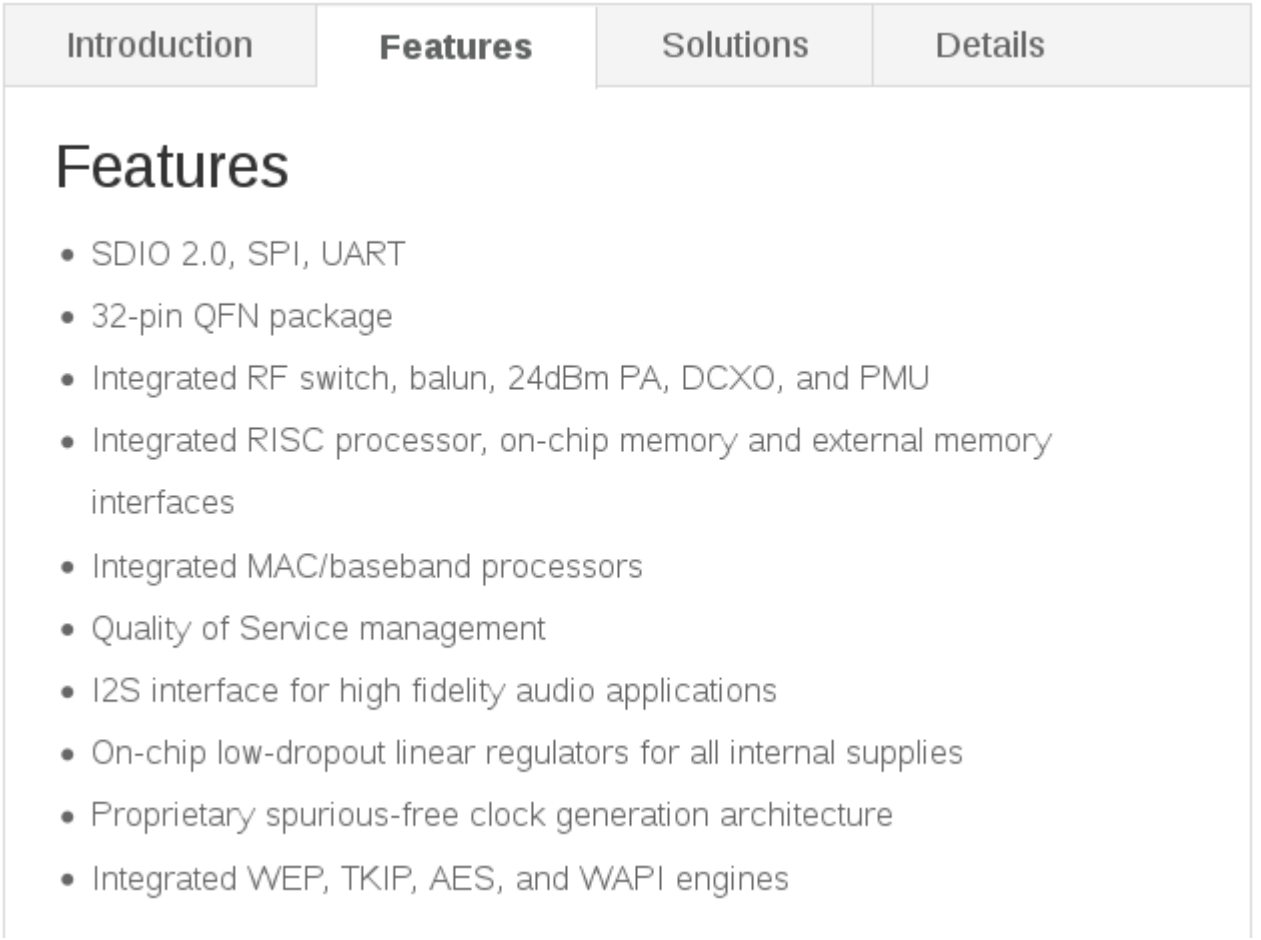
ESP-12E Development Board

Fabricant del xip: Espressif

http://espressif.com/en/products/esp8266/

ESP-12E Development Board

Característiques de l'ESP8266



Introduction | **Features** | Solutions | Details

Features

- SDIO 2.0, SPI, UART
- 32-pin QFN package
- Integrated RF switch, balun, 24dBm PA, DCXO, and PMU
- Integrated RISC processor, on-chip memory and external memory interfaces
- Integrated MAC/baseband processors
- Quality of Service management
- I2S interface for high fidelity audio applications
- On-chip low-dropout linear regulators for all internal supplies
- Proprietary spurious-free clock generation architecture
- Integrated WEP, TKIP, AES, and WAPI engines

http://espressif.com/en/products/esp8266/

ESP-12E Development Board

Especificacions de l'ESP8266

- 802.11 b/g/n
- WiFi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1/2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

http://espressif.com/en/products/esp8266/

ESP-12E Development Board

Datasheet de l'ESP8266

ESP8266EX Datasheet

Version 4.3

Espressif Systems IOT Team

http://bbs.espressif.com/

Copyright © 2015

Espressif Systems

ESP8266 Datasheet

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member Logo is a trademark of the WiFi Alliance.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

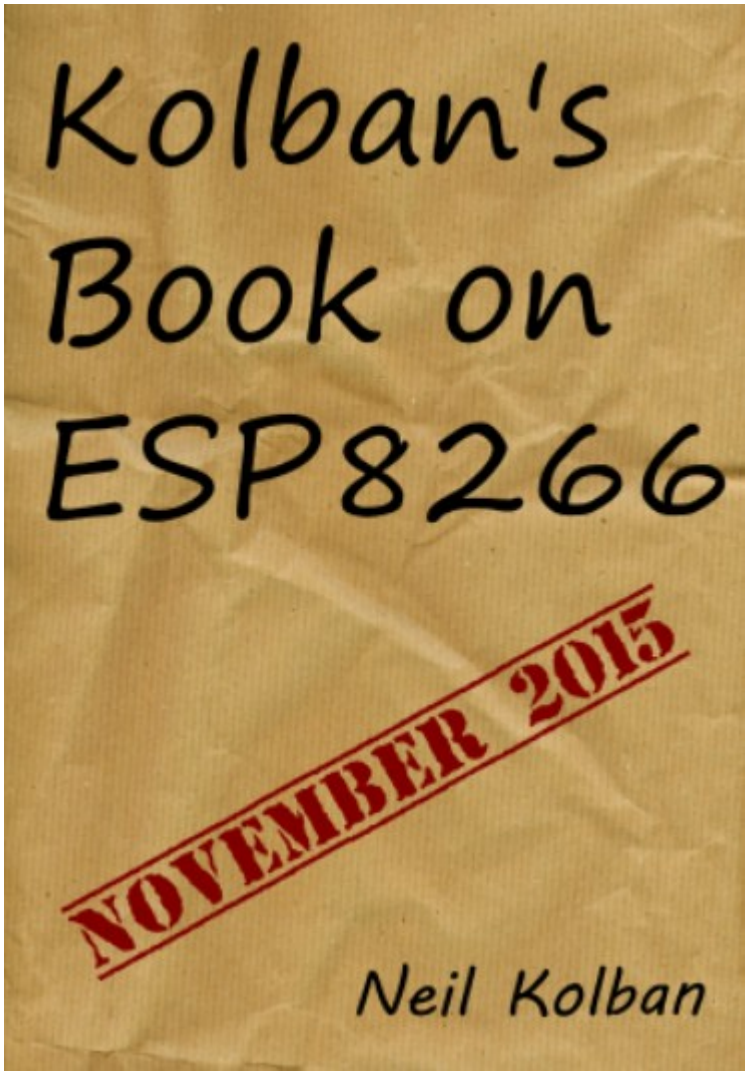
Copyright © 2015 Espressif Systems. All rights reserved.

Espressif Systems 2/31 June 1, 2015

Adafruit's link to ESP8266EX Datasheet v4.3

ESP-12E Development Board

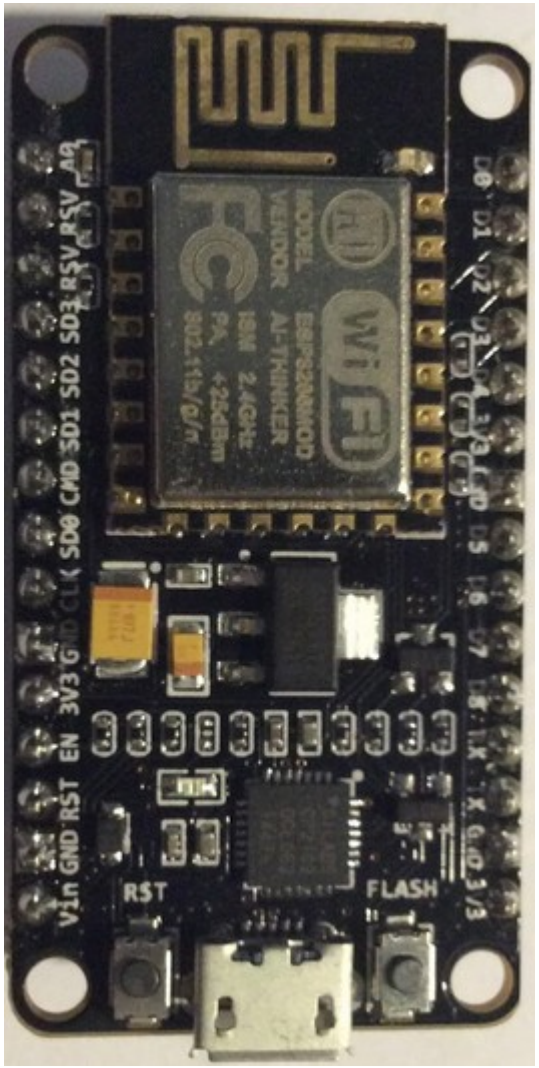
Kolban's Book on ESP8266

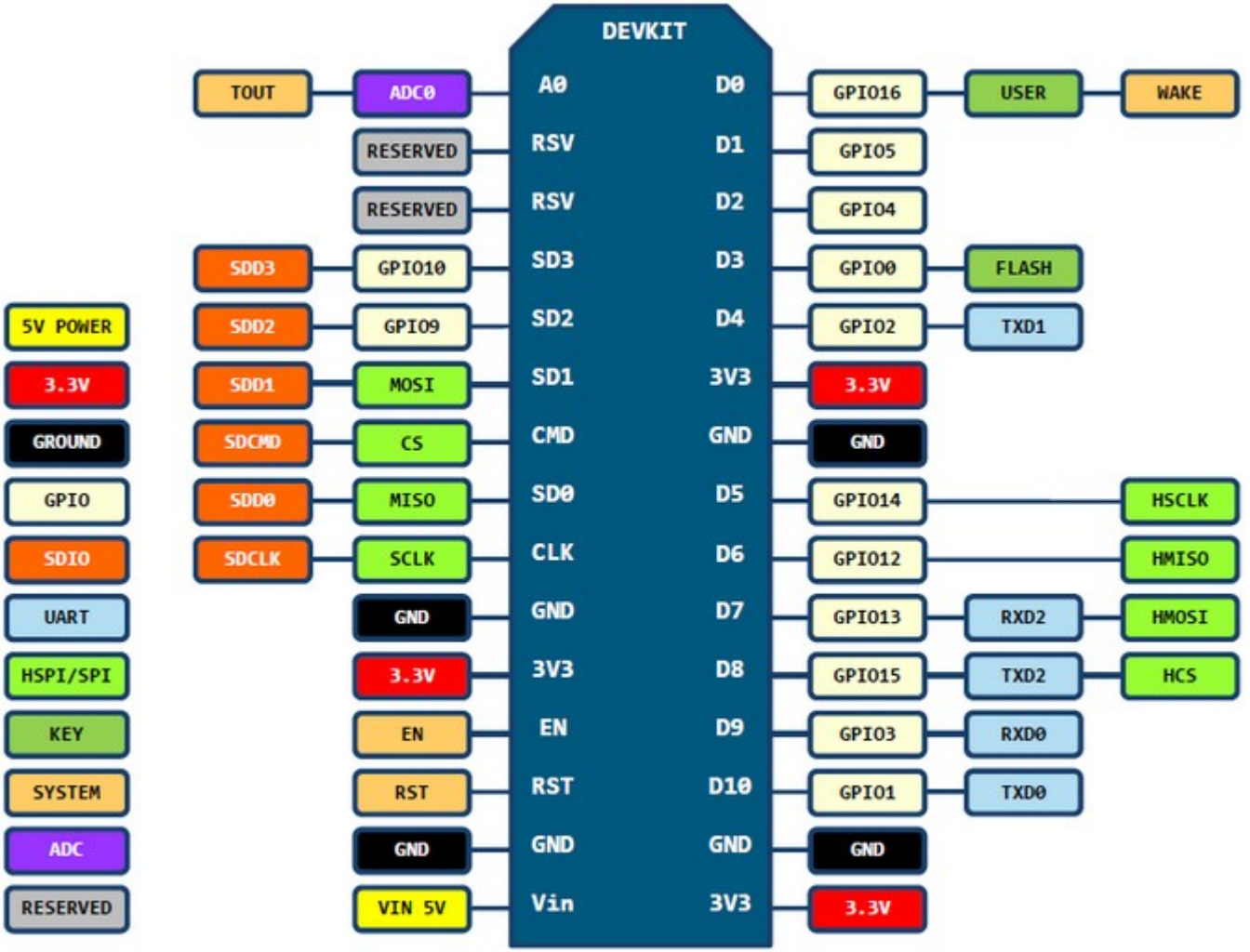


http://neilkolban.com/tech/esp8266/

ESP-12E Development Board

Disposició de pins de l'ESP8266 - NodeMcu





D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

ESP-12E Development Board

Programant l'ESP8266 – Ordres AT - Eines

```
# apt-get install python-pip

$ pip install -U setuptools
```

http://binefa.cat/php/esp8266/eines/esptool-master.zip

```
root@debian8:/home/ecat/Documents/dam/uf3/esptool-master# ls -ls
total 64
28 -rwxr-xr-x 1 ecat ecat 27179 set  2 03:03 esptool.py
20 -rw-r--r-- 1 ecat ecat 18092 set  2 03:03 LICENSE
 4 -rw-r--r-- 1 ecat ecat    34 set  2 03:03 MANIFEST.in
 8 -rw-r--r-- 1 ecat ecat  7286 set  2 03:03 README.md
root@debian8:/home/ecat/Documents/dam/uf3/esptool-master# sudo python setup.py install
running install
running bdist_egg
```

Seqüencia prèvia de preparació de l'ESP8266 per a ser programat :

Prémer el botó de RST, sense deixar-ho anar, prémer el botó de FLASH.
I mentre es pressiona el botó de FLASH es deixa anar el botó de RST.
Finalment es deixa de prémer el botó de FLASH

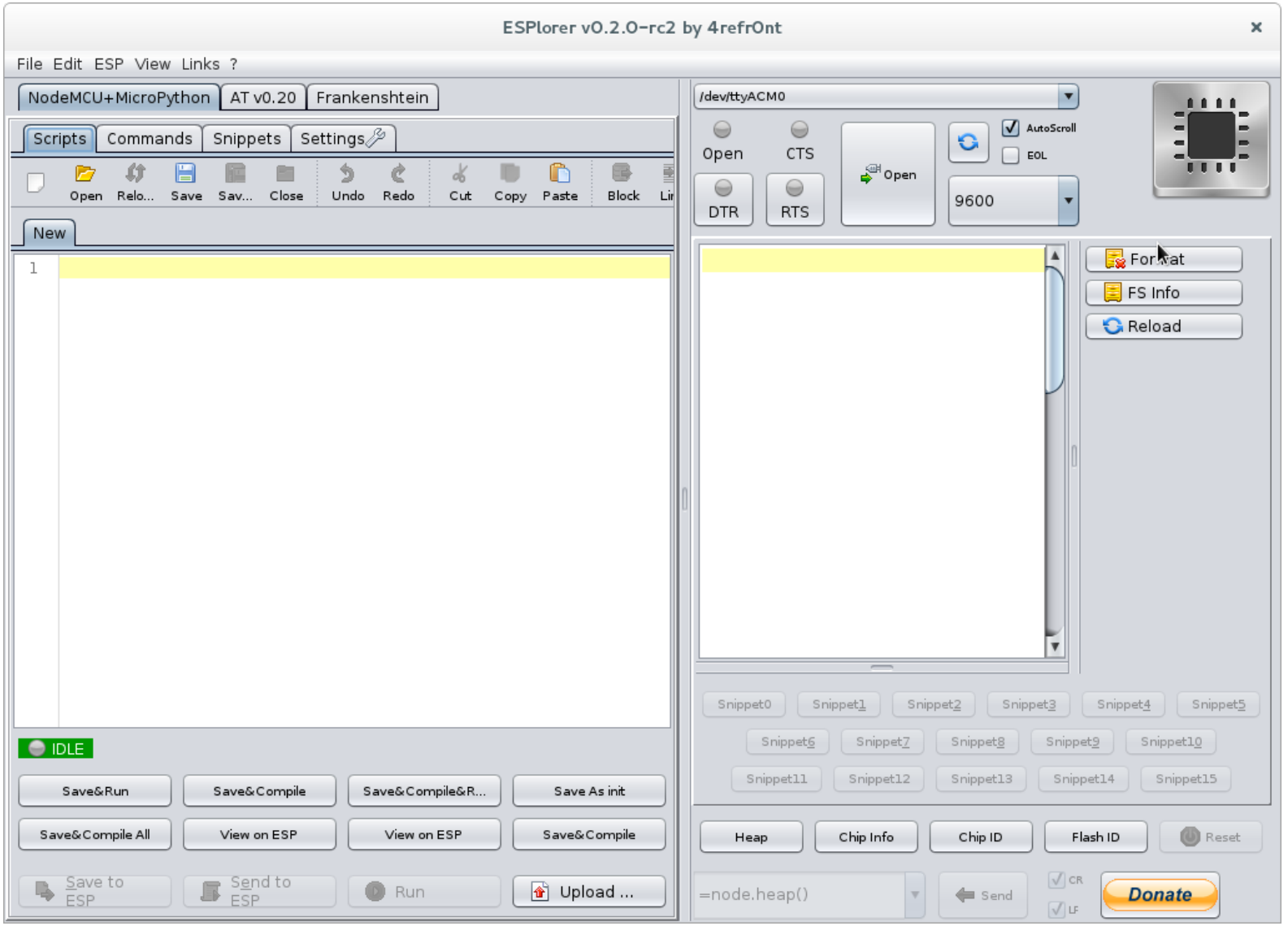
https://github.com/JhonControl/ESP8266-Flasher/tree/master/Firmware%20ESP8266

```
root@debian8:/home/ecat/Documents/dam/uf3/esptool-master# python esptool.py --port
 /dev/ttyUSB0 write_flash 0x00000 v0.9.5.2\ AT\ Firmware.bin
Connecting...
Erasing flash...
Wrote 520192 bytes at 0x00000000 in 49.4 seconds (84.3 kbit/s)...

Leaving...
root@debian8:/home/ecat/Documents/dam/uf3/esptool-master#
```

ESP-12E Development Board

Eines - ESPlorer

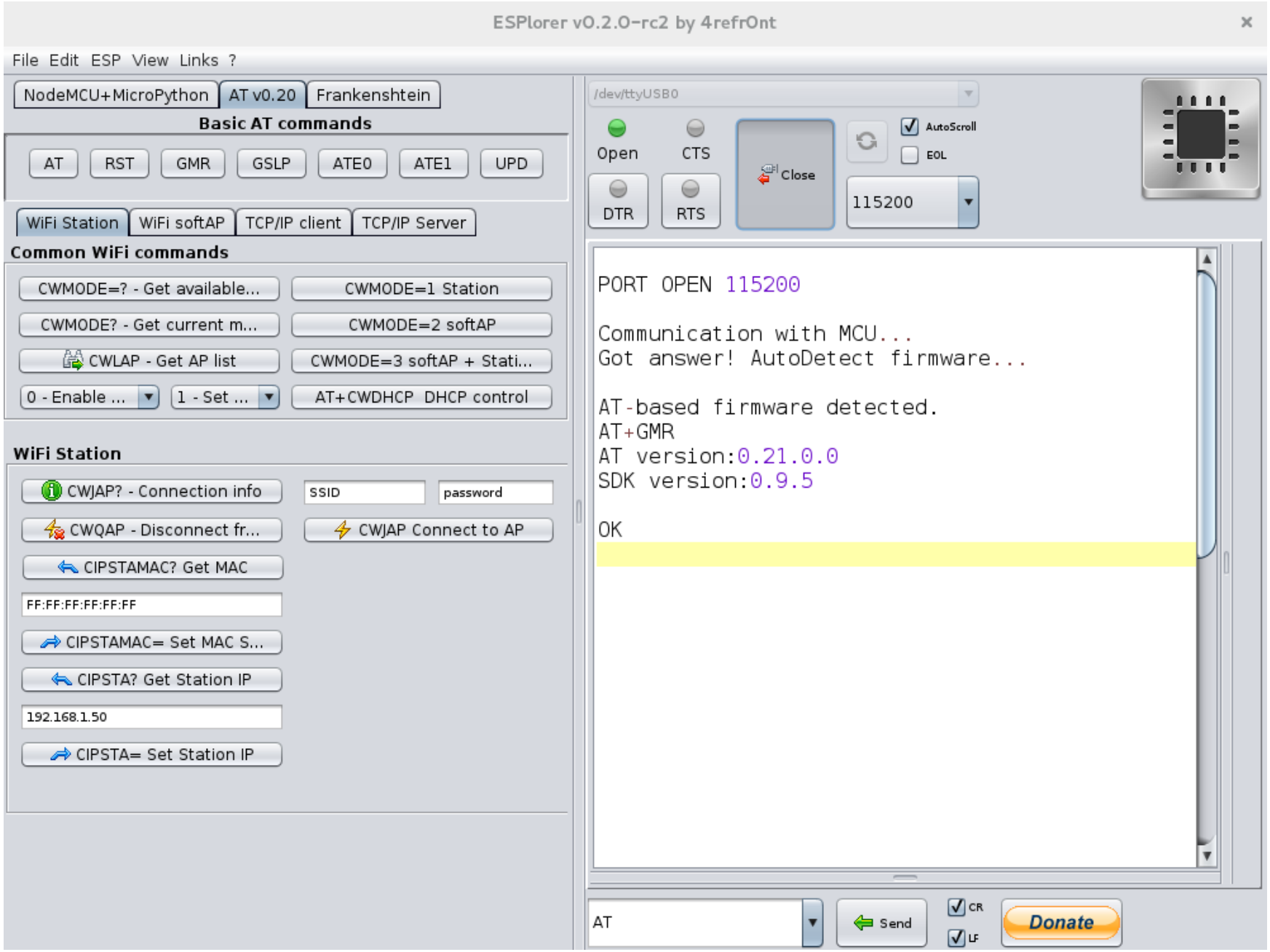
```
ecat@debian8:~/Documents/ESPlorer$ java -jar ESPlorer.jar
```



http://esp8266.ru/esplorer/

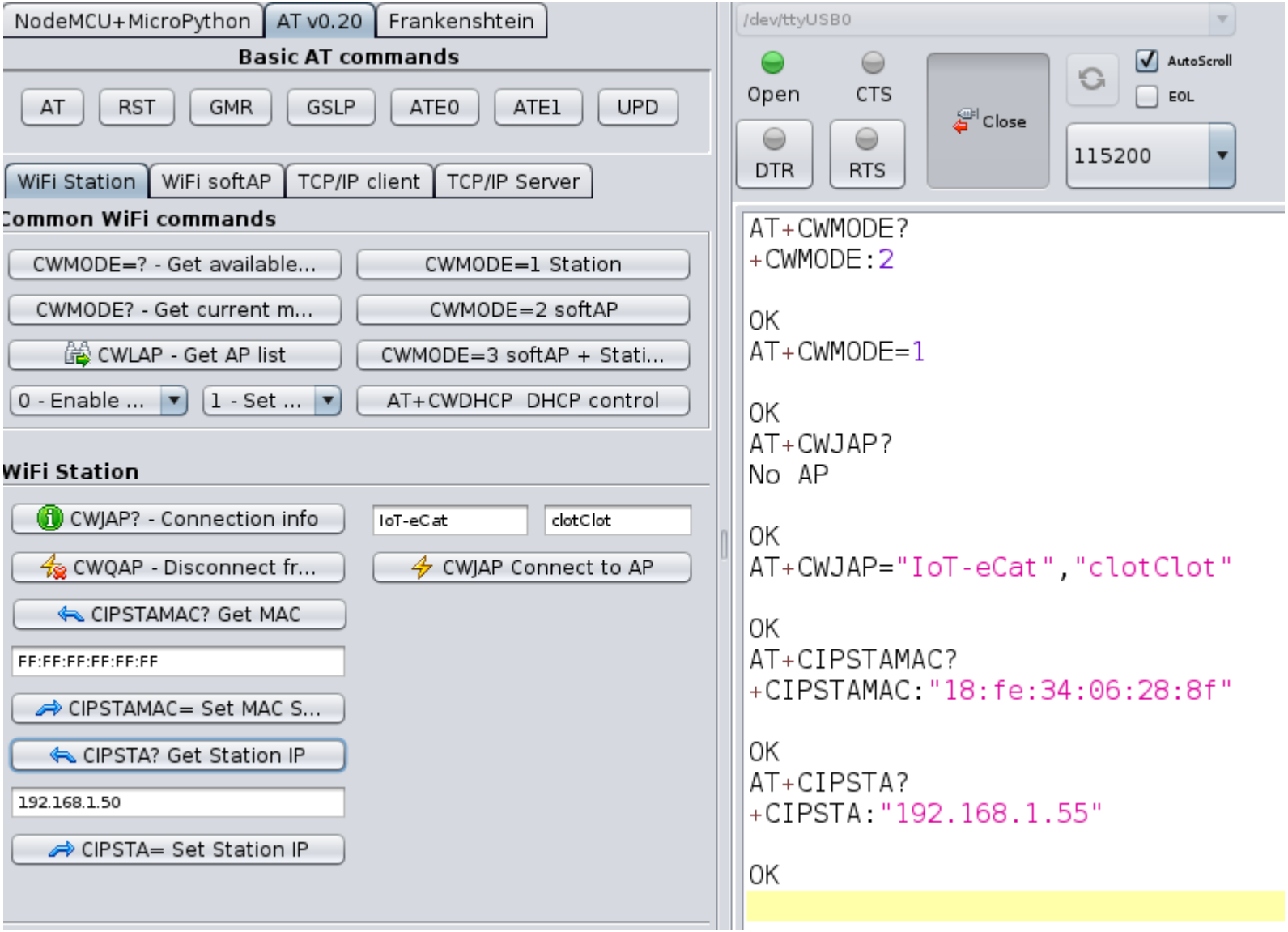
ESP-12E Development Board

Eines - ESPlorer



ESP-12E Development Board

Eines - ESPlorer



ESP-12E Development Board

Programant l'ESP8266 – LUA - Eines

Seqüencia prèvia de preparació de l'ESP8266 per a ser programat :

Verifiqueu que no hi ha cap programa emprant el port sèrie (p.e.: ESPlorer)
Prémer el botó de RST, sense deixar-ho anar, prémer el botó de FLASH.
I mentre es pressiona el botó de FLASH es deixa anar el botó de RST.
Finalment es deixa de prémer el botó de FLASH

https://github.com/nodemcu/nodemcu-devkit-v1.0
https://github.com/nodemcu/nodemcu-firmware
https://github.com/nodemcu/nodemcu-firmware/releases
http://nodemcu-build.com/

Downloads

nodemcu_float_0.9.6-dev_20150704.bin 451 KB

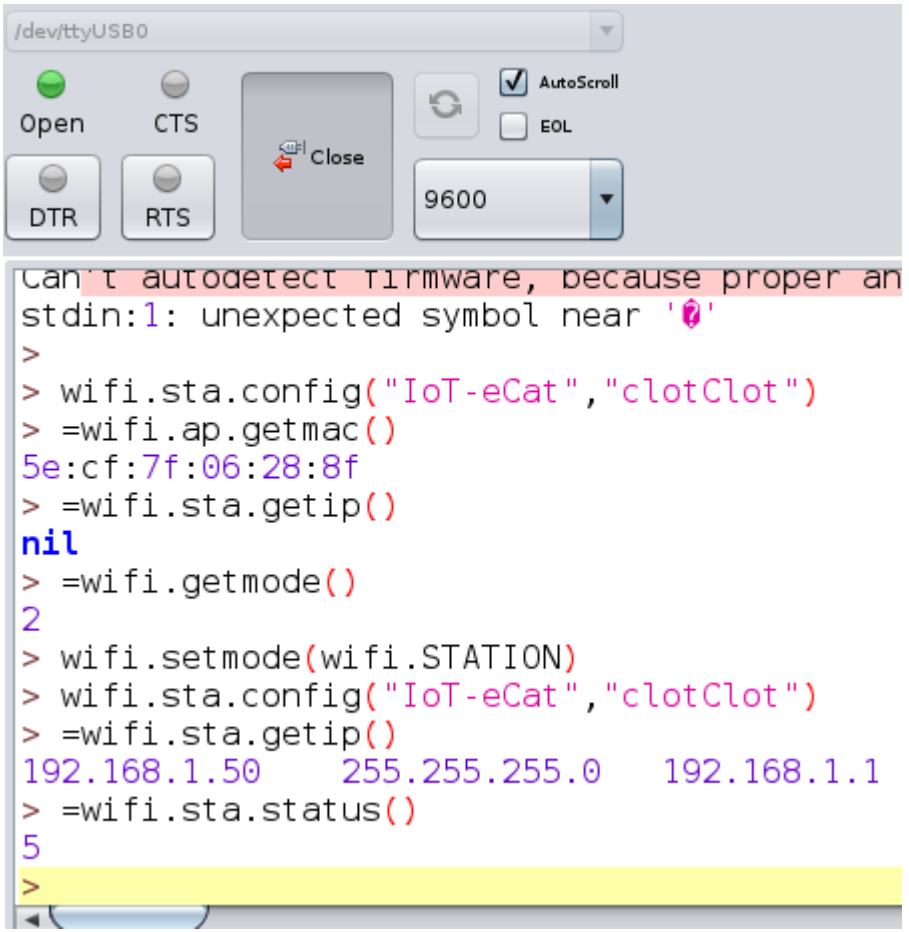
nodemcu_integer_0.9.6-dev_20150704.bin 440 KB

Source code (zip)

```
ecat@debian8:~/Documents/esp8266/lua$ esptool.py --port /dev/ttyUSB0 write_flash 0x00
000 nodemcu_integer_0.9.6-dev_20150704.bin
Connecting...
Erasing flash...
Wrote 450560 bytes at 0x00000000 in 43.3 seconds (83.2 kbit/s)...

Leaving...
ecat@debian8:~/Documents/esp8266/lua$ ls -ls
total 900
456 -rw-r----- 1 ecat ecat 461984 gen 21 13:37 nodemcu_float_0.9.6-dev_20150704.bin
444 -rw-r----- 1 ecat ecat 450072 gen 21 13:37 nodemcu_integer_0.9.6-dev_20150704.bin
ecat@debian8:~/Documents/esp8266/lua$
```

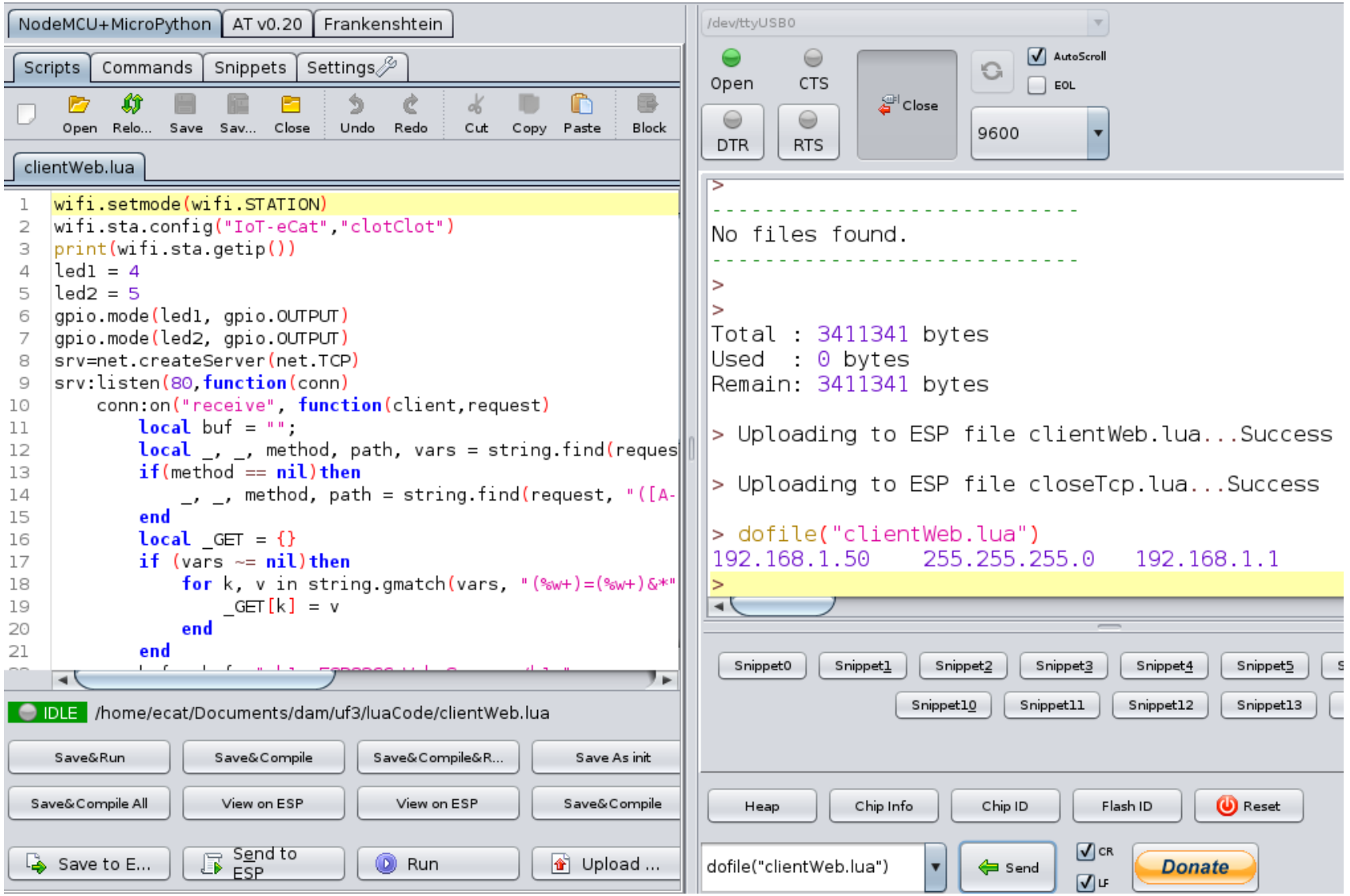
ESP-12E Development Board

Programant l'ESP8266 - LUA



ESP-12E Development Board

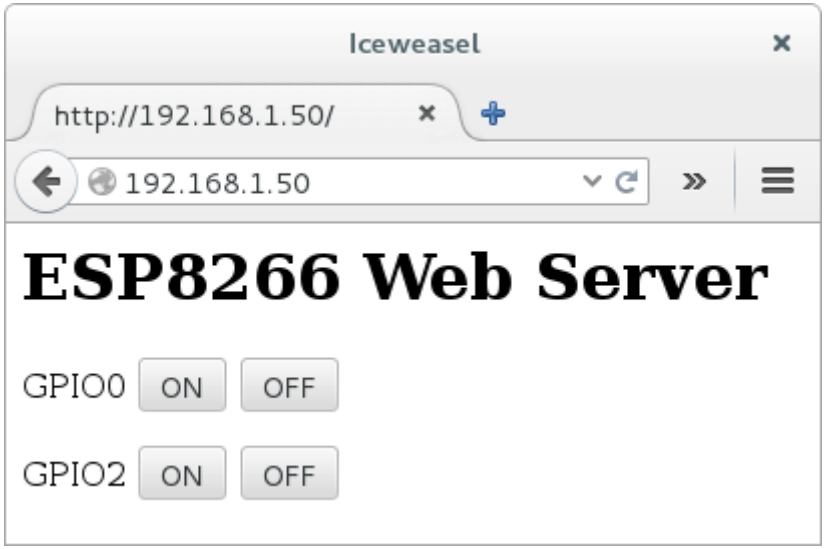
Programant l'ESP8266 - LUA



http://binefa.cat/php/esp8266/codis/lua/clientWeb.lua
http://binefa.cat/php/esp8266/codis/lua/closeTcp.lua

ESP-12E Development Board

Programant l'ESP8266 - LUA



http://binefa.cat/php/esp8266/codis/lua/clientWeb.lua
http://binefa.cat/php/esp8266/codis/lua/closeTcp.lua

ESP-12E Development Board

Programant l'ESP8266 - LUA

Connect to the wireless network

```
print(wifi.sta.getip())
--nil
wifi.setmode(wifi.STATION)
wifi.sta.config("SSID","password")
print(wifi.sta.getip())
--192.168.18.110
```

Blinking Led

```
lighton=0
tmr.alarm(0,1000,1,function()
if lighton==0 then
  lighton=1
  led(512,512,512)
  -- 512/1024, 50% duty cycle
else
  lighton=0
  led(0,0,0)
end
end)
```

A pure lua telnet server

```
-- a simple telnet server
s=net.createServer(net.TCP,180)
s:listen(2323,function(c)
  function s_output(str)
    if(c~=nil)
      then c:send(str)
    end
  end
  node.output(s_output, 0)
  -- re-direct output to function s_ouput.
  c:on("receive",function(c,l)
    node.input(l)
    --like pcall(loadstring(l)), support multiple separate
lines
  end)
  c:on("disconnection",function(c)
    node.output(nil)
    --unregist redirect output function, output goes to
serial
  end)
  print("Welcome to NodeMCU world.")
end)
```

Bootstrap

```
--init.lua will be excuted
file.open("init.lua","w")
file.writeline([[print("Hello World!")]])
file.close()
node.restart()  -- this will restart the module.
```

HTTP Client

```
-- A simple http client
conn=net.createConnection(net.TCP, false)
conn:on("receive", function(conn, pl) print(pl) end)
conn:connect(80,"121.41.33.127")
conn:send("GET / HTTP/1.1\r\nHost:
www.nodemcu.com\r\n"
  .."Connection: keep-alive\r\nAccept: */*\r\n\r\n")
```

HTTP Server

```
-- a simple http server
srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
  conn:on("receive",function(conn,payload)
  print(payload)
  conn:send("<h1> Hello, NodeMCU.</h1>")
  end)
end)
```

Use timer to repeat

```
tmr.alarm(1,5000,1,function() print("alarm 1") end)
tmr.alarm(0,1000,1,function() print("alarm 0") end)
tmr.alarm(2,2000,1,function() print("alarm 2") end)
-- after sometime
tmr.stop(0)
```

http://www.banggood.com/NodeMcu-Lua-WIFI-Development-Board-For-ESP8266-Module-p-976440.html

ESP-12E Development Board

Programant l'ESP8266 – microPython - Eines

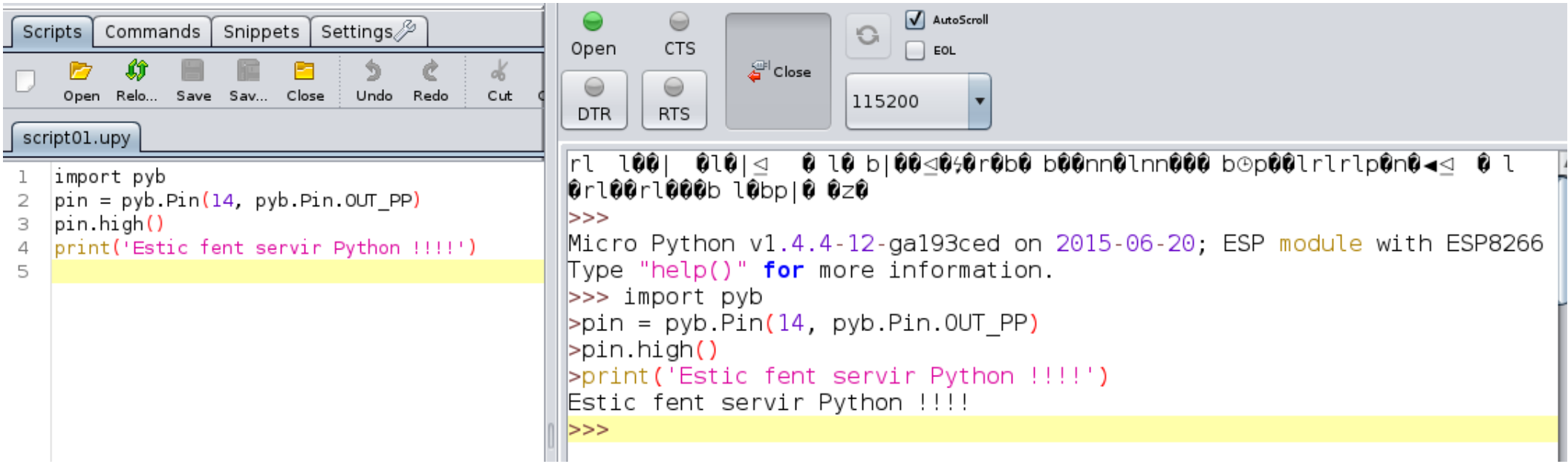
Seqüencia prèvia de preparació de l'ESP8266 per a ser programat :
Verifiqueu que no hi ha cap programa emprant el port sèrie (p.e.: ESPlorer)
Prémer el botó de RST, sense deixar-ho anar, prémer el botó de FLASH.
I mentre es pressiona el botó de FLASH es deixa anar el botó de RST.
Finalment es deixa de prémer el botó de FLASH

https://micropython.org/
https://learn.adafruit.com/building-and-running-micropython-on-the-esp8266
http://www.electrodragon.com/w/MicroPython_ESP8266
http://www.0x43.nl/esp8266-and-micropython/
http://dev.0x43.nl/wp-content/uploads/2015/06/build-MP-esp8266-2015-06-20.tar.gz

```
ecat@debian8:~/Documents/esp8266/flash/build$ esptool.py --port /dev/ttyUSB0 wri
te_flash 0x00000 firmware-combined.bin
Connecting...
Erasing flash...
Wrote 327680 bytes at 0x00000000 in 31.6 seconds (83.0 kbit/s)...

Leaving...
ecat@debian8:~/Documents/esp8266/flash/build$
```

ESP-12E Development Board

Programant l'ESP8266 – microPython - Eines



http://binefa.cat/php/esp8266/codis/micropython/script01.upy

https://learn.adafruit.com/building-and-running-micropython-on-the-esp8266/micropython-usage

ESP-12E Development Board

Programant l'ESP8266 – WiFi a sèrie - Eines

Seqüencia prèvia de preparació de l'ESP8266 per a ser programat :
Verifiqueu que no hi ha cap programa emprant el port sèrie (p.e.: ESPlorer)
Prémer el botó de RST, sense deixar-ho anar, prémer el botó de FLASH.
I mentre es pressiona el botó de FLASH es deixa anar el botó de RST.
Finalment es deixa de prémer el botó de FLASH

http://en.doit.am/doit_esp_wifi_serial.zip

```
ecat@debian8:~/Documents/esp8266/ESP8266 Serial©\to©\WiFi Transmission Firmware$ esptool.py
 --port /dev/ttyUSB0 write_flash 0x00000 ESP8266_Doit_ser2net\(v2.4\).bin
Connecting...
Erasing flash...
Wrote 455680 bytes at 0x00000000 in 43.4 seconds (84.0 kbit/s)...

Leaving...
ecat@debian8:~/Documents/esp8266/ESP8266 Serial©\to©\WiFi Transmission Firmware$
```

ESP-12E Development Board

Programant l'ESP8266 – WiFi a sèrie - Eines

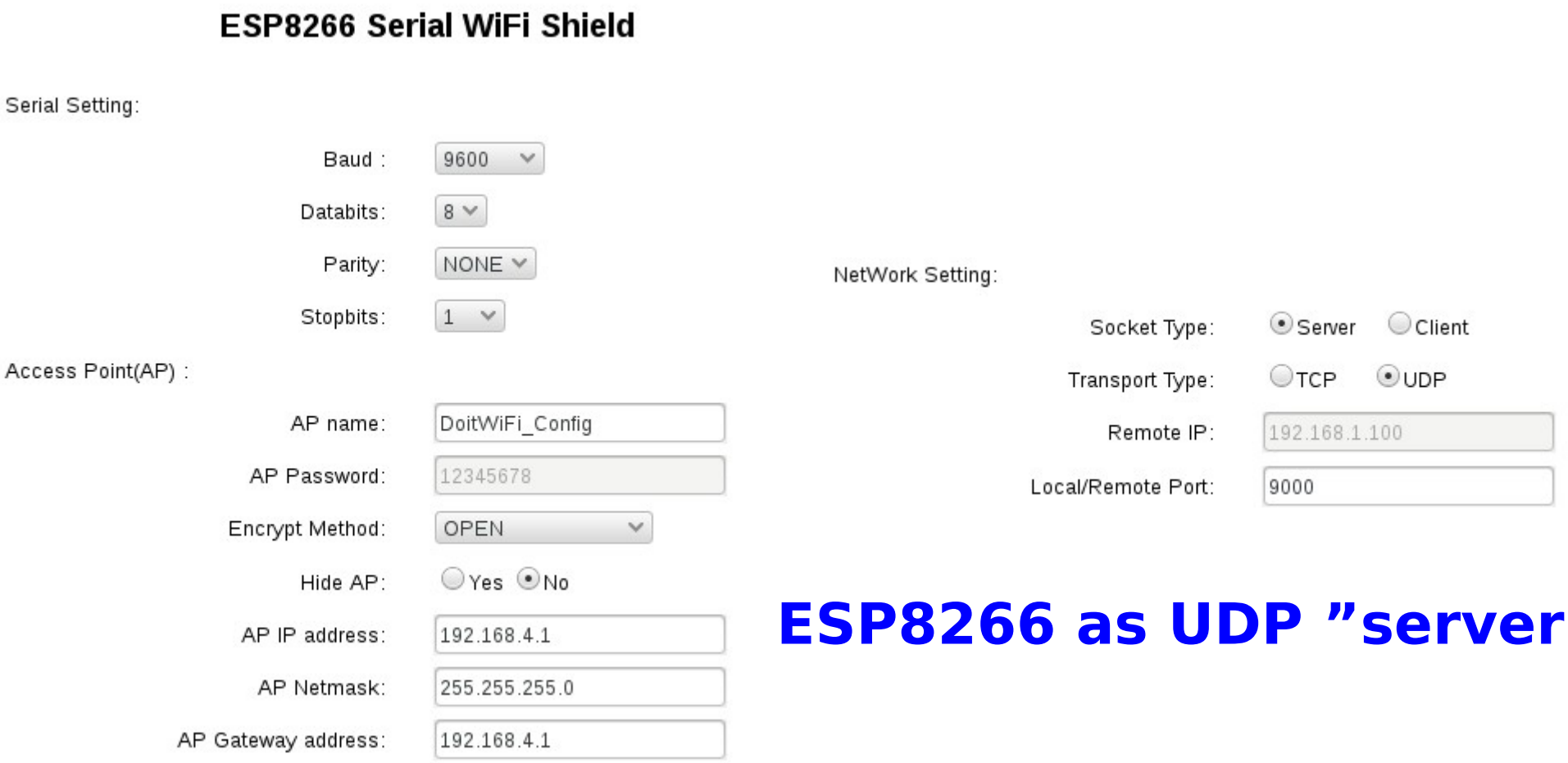
```
root@debian8:~# iwlist wlan0 scanning | grep Doit
                    ESSID:"DoitWiFi_Config"
root@debian8:~#
```

```
root@debian8:~# iwlist wlan0 scanning | grep Doit
                    ESSID:"DoitWiFi_Config"
root@debian8:~# iwconfig
eth0      no wireless extensions.

wlan0     IEEE 802.11bgn  ESSID:"DoitWiFi_Config"
          Mode:Managed  Frequency:2.412 GHz  Access Point: 5E:CF:7F:06:28:8F
          Bit Rate=1 Mb/s   Tx-Power=16 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=70/70  Signal level=-23 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:1  Invalid misc:63   Missed beacon:0

wwan0     no wireless extensions.

lo        no wireless extensions.

root@debian8:~#
```

http://en.doit.am/doit_esp_wifi_serial.zip

ESP-12E Development Board

Programant l'ESP8266 – WiFi a sèrie - Eines



ESP8266 as UDP ”server”

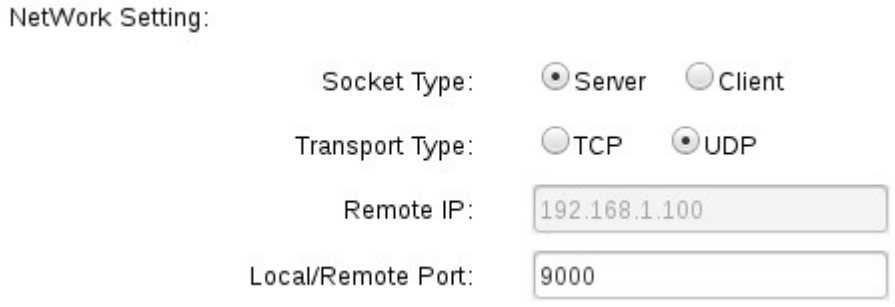
http://en.doit.am/doit_esp_wifi_serial.zip

ESP-12E Development Board

Programant l'ESP8266 – WiFi a sèrie - Eines

```
ecat@debian8:~/Documents/dam/uf3/udp/qtPyUdp$ minicom -b 9600 -o -D /dev/ttyUSB0
```

Obriu el **minicom** i **qtPyUdp.py** a terminals diferents



http://binefa.cat/php/esp8266/codis/Qt_Python/qtPyUdp.tar.gz

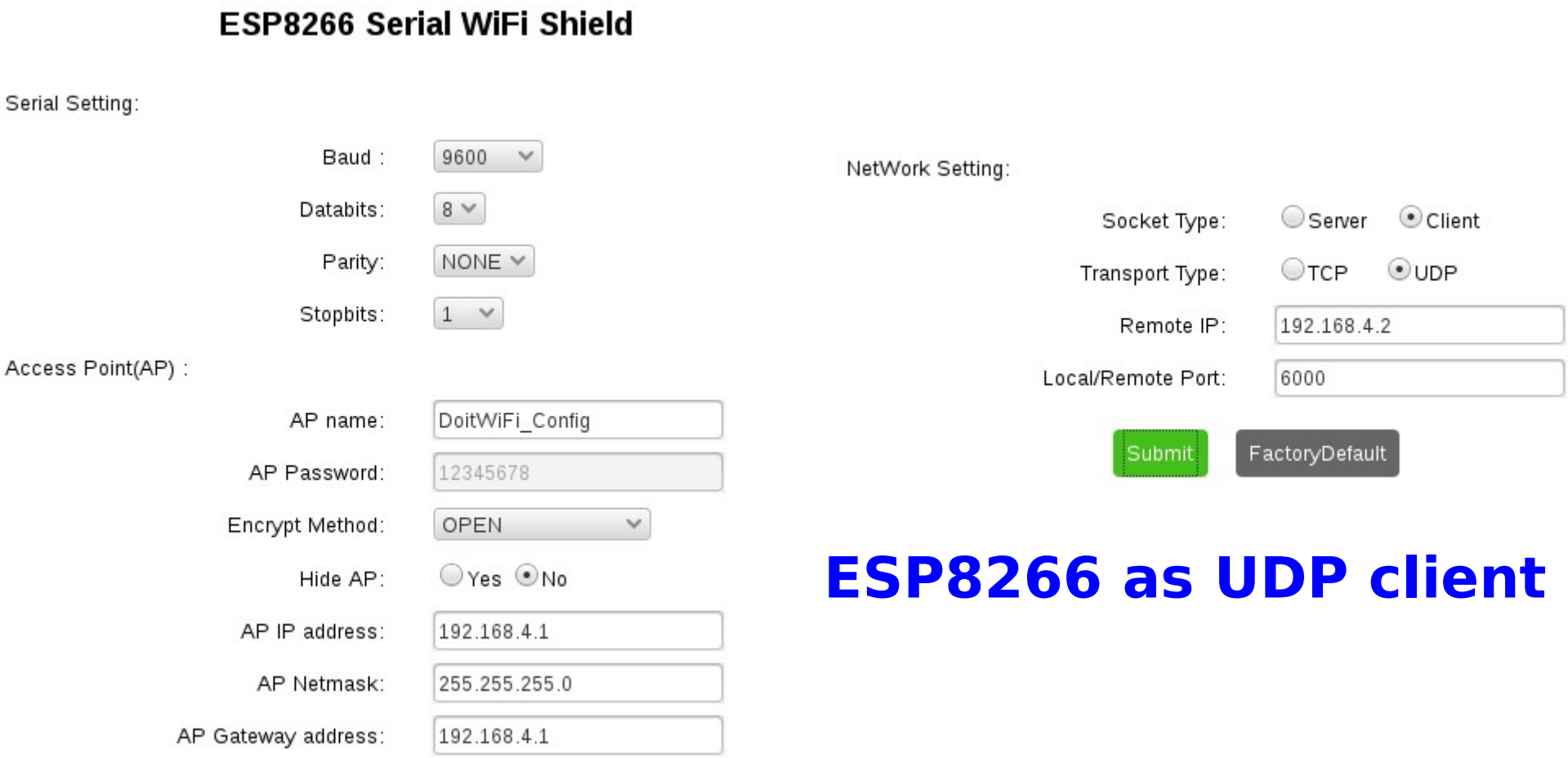
```
ecat@debian8:~/Documents/dam/uf3/udp/qtPyUdp$ ./qtPyUdp.py 6000
```

```
Welcome to minicom 2.7

OPTI+-----------------------------------------
Comp| A -    Serial Device      : /dev/ttyUSB0
Port| B - Lockfile Location     : /var/lock
    | C -   Callin Program      :
Pres| D -  Callout Program      :
    | E -    Bps/Par/Bits       : 9600 8N1
    | F - Hardware Flow Control : No
    | G - Software Flow Control : No
    |
    |    Change which setting? 
    +-----------------------------------------
              | Screen and keyboard      |
              | Save setup as dfl        |
              | Save setup as..          |
              | Exit                     |
              +--------------------------+
```


```
Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda

Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan  1 2014, 09:30:18.
Port /dev/ttyUSB0, 13:01:03

Press CTRL-A Z for help on special keys

Hola
```

ESP-12E Development Board

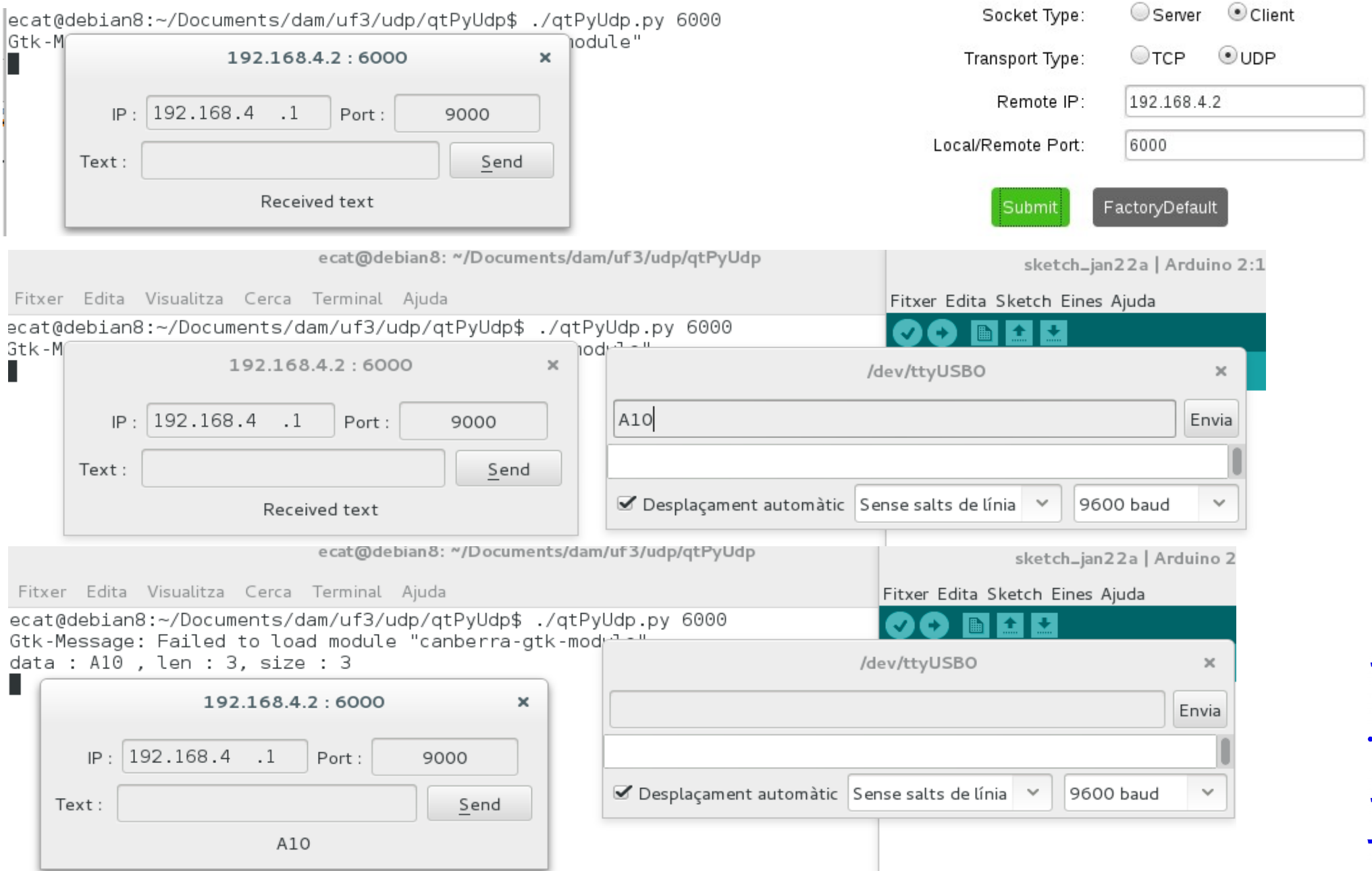
Programant l'ESP8266 – WiFi a sèrie - Eines



ESP8266 as UDP client

ESP-12E Development Board

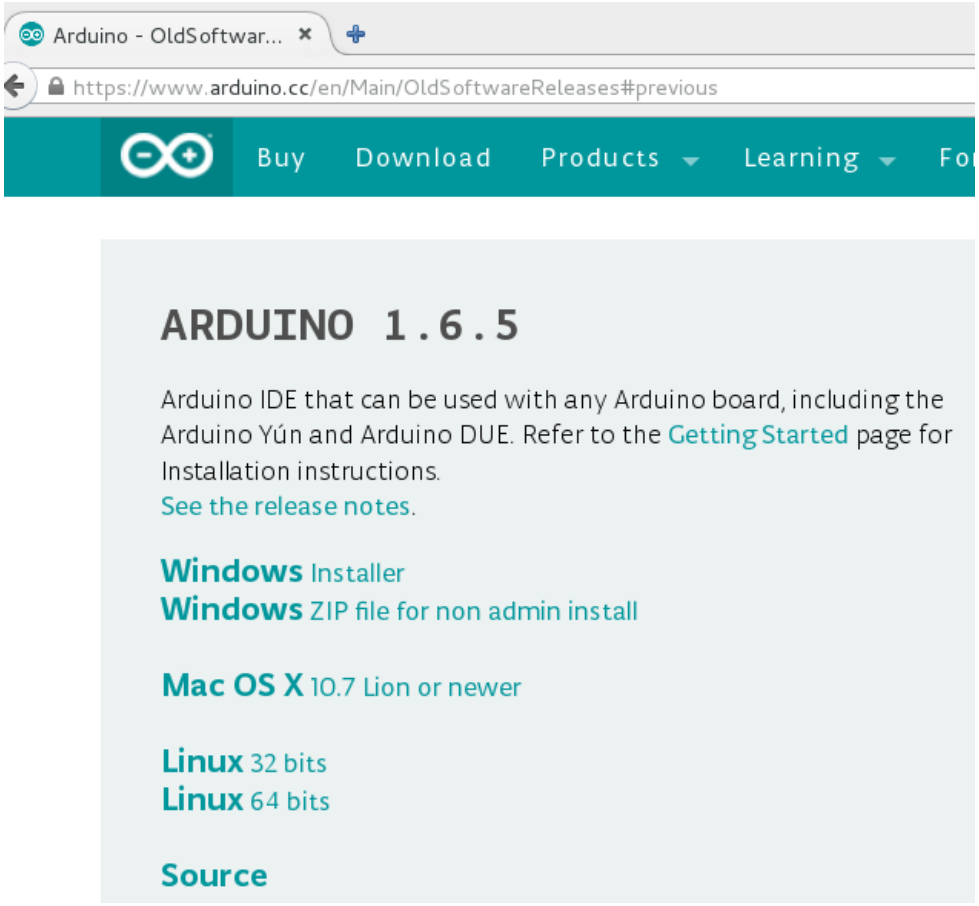
Programant l'ESP8266 – WiFi a sèrie - Eines



ESP-12E Development Board

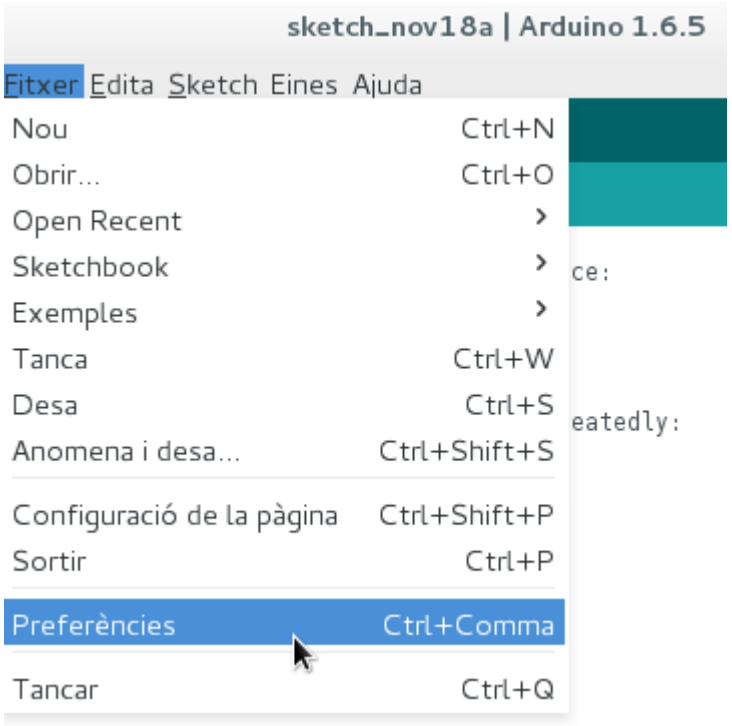
Desenvolupant amb l'ESP8266 sobre l'IDE d'Arduino

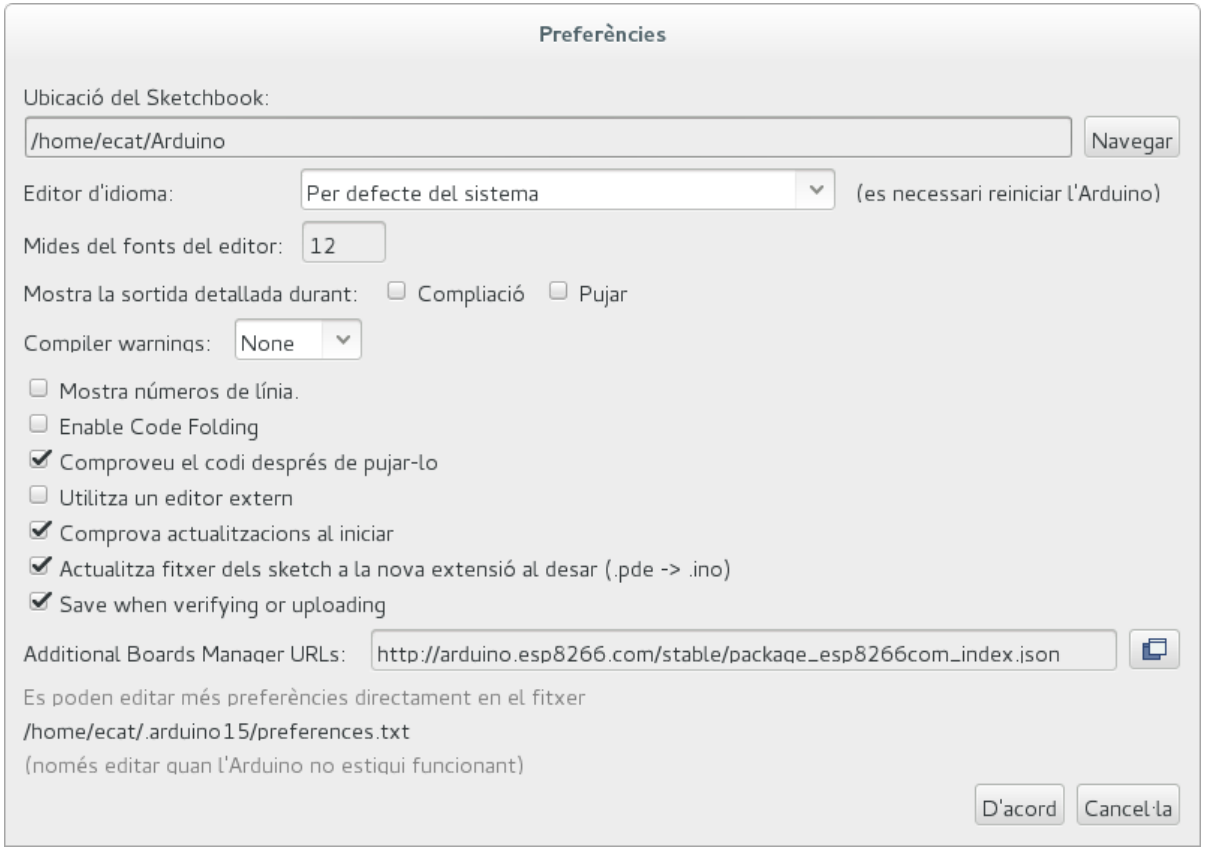
Descarregueu-vos l'instal·lador de l'IDE d'Arduino 1.6.5. des de:
https://www.arduino.cc/en/Main/OldSoftwareReleases



ESP-12E Development Board

Desenvolupant amb l'ESP8266 sobre l'IDE d'Arduino



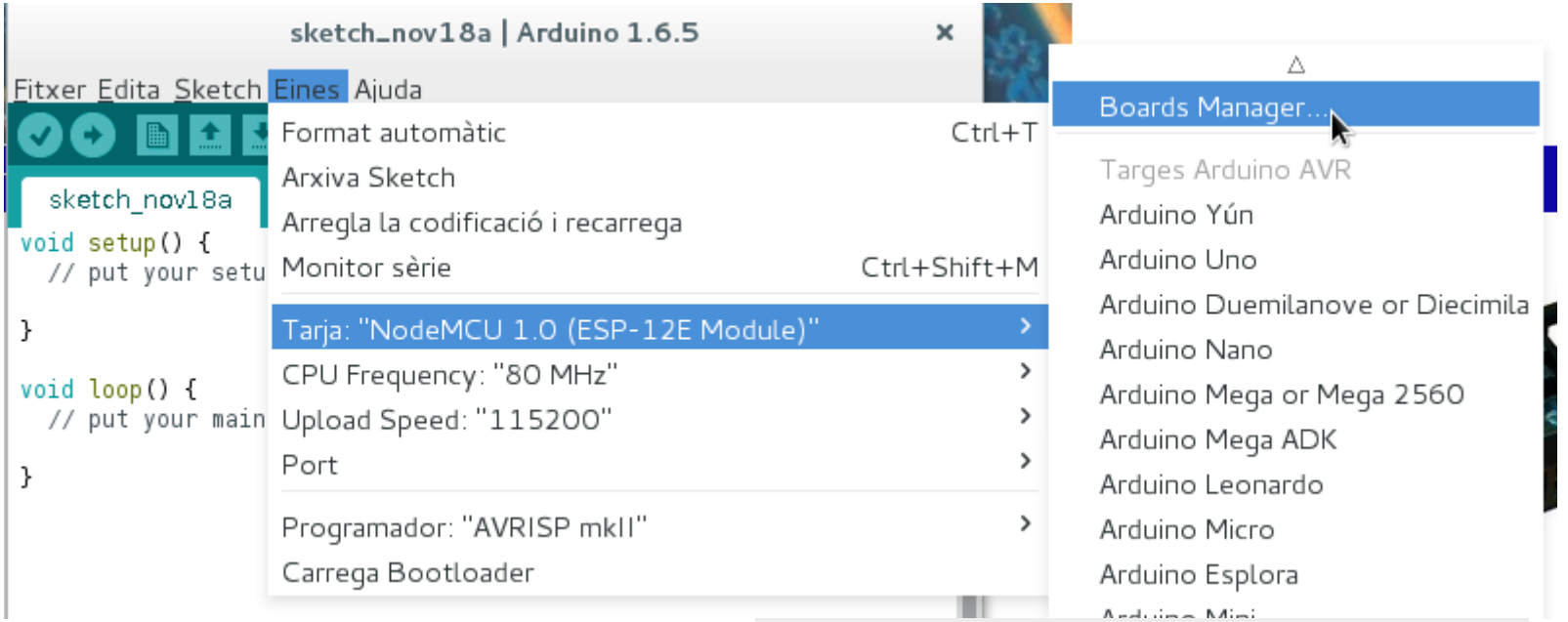


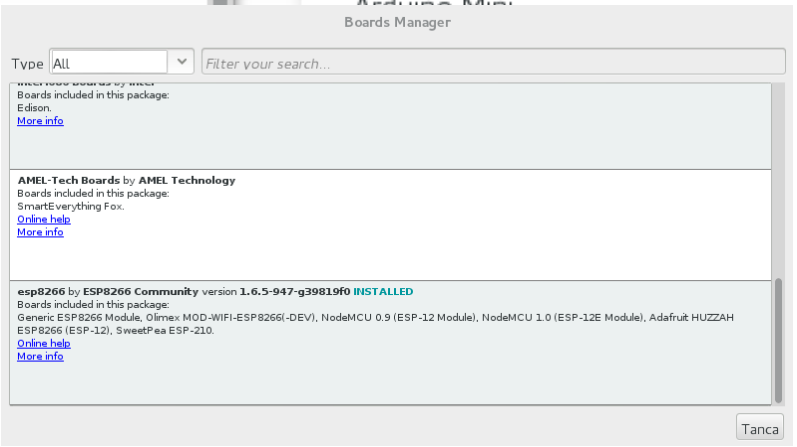
Entreu aquesta adreça a ”Additional Boards Manager URLs”:
http://arduino.esp8266.com/stable/package_esp8266com_index.json

ESP-12E Development Board

Desenvolupant amb l'ESP8266 sobre l'IDE d'Arduino

Tools → Board: “current” → Boards Manager ...





Click install ESP8266
(It is about 450 MB to download) :

Una altra manera de fer la instal·lació de l'IDE d'Arduino per a ESP8266:
https://libraries.io/github/adafruit/ESP8266-Arduino

ESP-12E Development Board

Trametent trames UDP amb l'ESP8266

UDPClientMAC02 | Arduino 1.6.5

Fitxer Edita Sketch Eines Ajuda

UDPClientMAC02

```
#include <ESP8266WiFi.h>
//#include <WiFiUDP.h>
#include <WiFiUdp.h>


#ifdef ESP8266
extern "C" {
#include "user_interface.h"
}
#endif

const char* ssid     = "IoT-eCat";
const char* password = "clotClot";

// A UDP instance to let us send and receive packets over UDP
WiFiUDP Udp;


void setup() {
  Serial.begin(115200);
  delay(10);

  // We start by connecting to a WiFi network
```

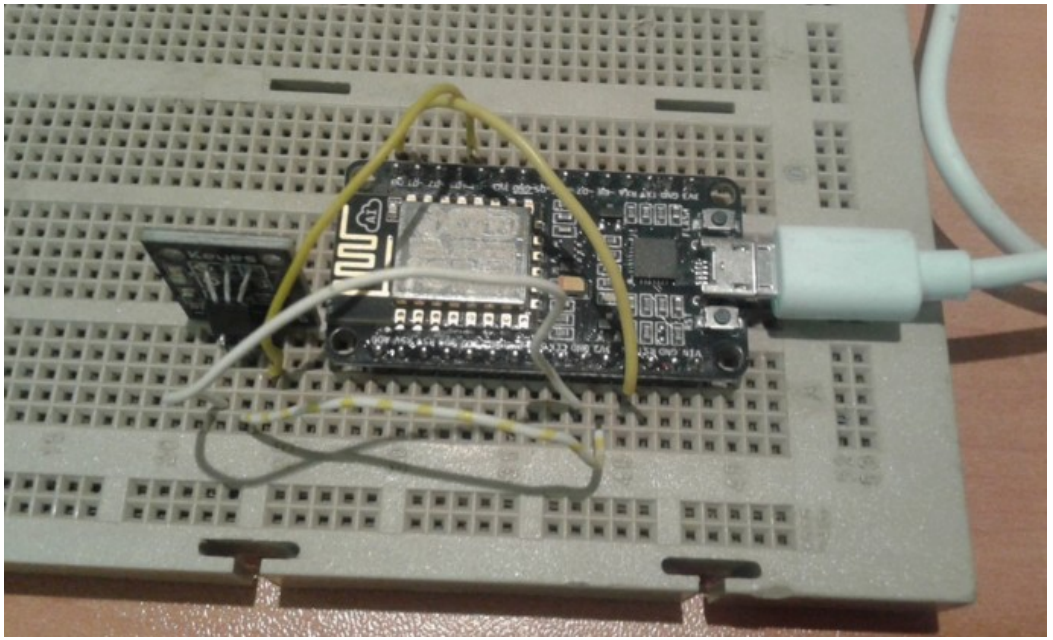
NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200 on /dev/ttyUSB0

```
void loop() {
//  char temp[20];      // buffer for for
  char temp[28];      // buffer for forma

  vGetMac(temp);

  Udp.write(temp);
  Udp.endPacket();

  Serial.print(temp);
  Serial.println(" sent to UDP server");

  delay(5000);
}
```

http://binefa.cat/php/esp8266/codis/arduinoIDE/UDPClientMAC02/UDPClientMAC02.ino

ESP-12E Development Board

ESP8266 sobre l'IDE d'Arduino - MQTT



```
Fitxer  Edita  Vi
Client mosqsub
Client mosqsub
24.94
Client mosqsub
25.00
Client mosqsub
24.94
Client mosqsub
25.00
Client mosqsub
Client mosqsub
Client mosqsub
25.00
Client mosqsub
25.06
^C
ecat@debian8:~$ mosquitto_sub -d -t sensors/test/temperature
Client mosqsub/15509-debian8 sending CONNECT
Client mosqsub/15509-debian8 received CONNACK
Client mosqsub/15509-debian8 sending SUBSCRIBE (Mid: 1, Topic:
Client mosqsub/15509-debian8 received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/15509-debian8 received PUBLISH (d0, q0, r0, m0,
24.81
Client mosqsub/15509-debian8 received PUBLISH (d0, q0, r0, m0,
24.75
Client mosqsub/15509-debian8 received PUBLISH (d0, q0, r0, m0,
24.69
```

```
Entering deep sleep mode for 3 seconds...
rl  l@@|  @l@|◁   @ l@ b|@@◁@,@r@b@ b@@nn@lnn@@@ b⊙p@@lrlrlp@n
Connecting to IoT-eCat
.............
WiFi connected
IP address:
192.168.1.12
Attempting MQTT connection...connected
Requesting DS18B20 temperature...
Sending temperature: 24.69
Closing MQTT connection...
Closing WiFi connection...
Entering deep sleep mode for 3 seconds...
rl  l@@|  @l@|◁   @ l@ b|@@◁@,@r@b@ b@@nn@lnn@@@ b⊙p@@lrlrlp@n
Connecting to IoT-eCat
....
☑ Desplaçament automàtic
```

http://www.jerome-bernard.com/blog/2015/10/04/wifi-temperature-sensor-with-nodemcu-esp8266/

ESP-12E Development Board

Instal·lació de l'esp-open-sdk

```
root@debian8:~# sudo aptitude install git

ecat@debian8:~/Documents/esp8266$ git clone --recursive https://github.com/pfalcon/esp-open-sdk.git

# apt-get install make unrar autoconf automake libtool gcc g++ gperf flex \
   bison texinfo gawk ncurses-dev libexpat-dev python sed git libtool-bin

$ make STANDALONE=y
```

Configuració del PATH

Per a poder cridar els binaris xtensa-lx106-*generats, podeu afegir la ruta del SDK al PATH. Podeu actualitzar el PATH cada vegada que inicieu una sessió de terminal:

```
export PATH=[your esp-open-sdk directory]/bin:$PATH

Per exemple:
export PATH=/home/ecat/Documents/esp8266/esp-open-sdk/bin:$PATH

Per a fer aquest canvi permanent afegiu-ho a la darrera línia de l'arxiu .profile del vostre directori
d'usuari.
```

```
ecat@debian8:~$ tail .profile
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

export PATH=/home/ecat/Documents/esp8266/esp-open-sdk/bin:$PATH
ecat@debian8:~$
```

http://www.esp8266.com/wiki/doku.php?id=setup-linux-compiler-esp8266

ESP-12E Development Board

Programant l'ESP8266 - microPython

```
root@debian8:~# sudo aptitude install git

ecat@debian8:~/Documents/esp8266$ git clone --recursive https://github.com/pfalcon/esp-open-sdk.git

# apt-get install make unrar autoconf automake libtool gcc g++ gperf flex \
   bison texinfo gawk ncurses-dev libexpat-dev python sed git libtool-bin

$ make STANDALONE=y
```

Torn de preguntes ...

... i sessió pràctica.