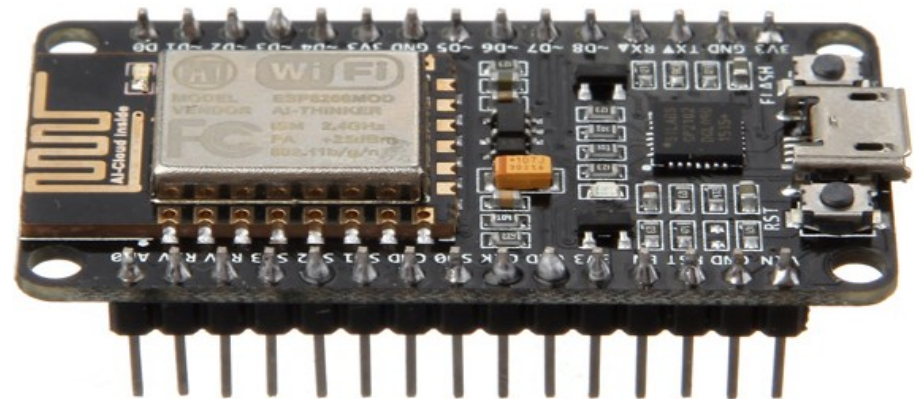


ESP8266

ESP-12E Development Board



Jordi Binefa i Martínez

Responsable d'R+D+i a

novembre de 2015

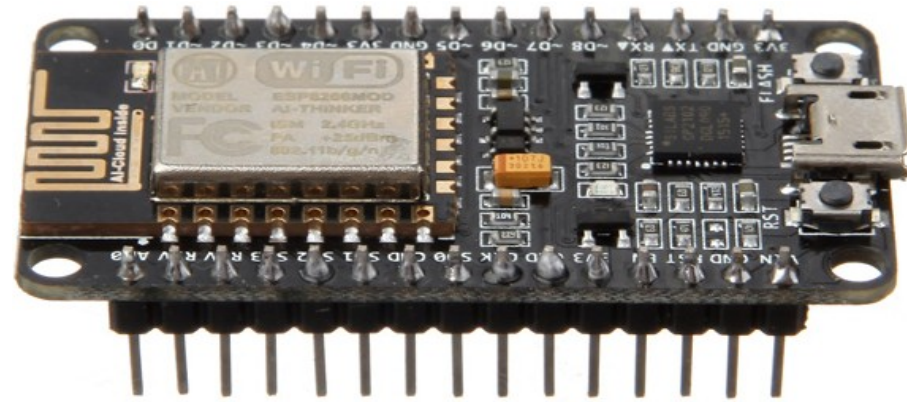
Professor de cicles formatius a





ESP-12E Development Board

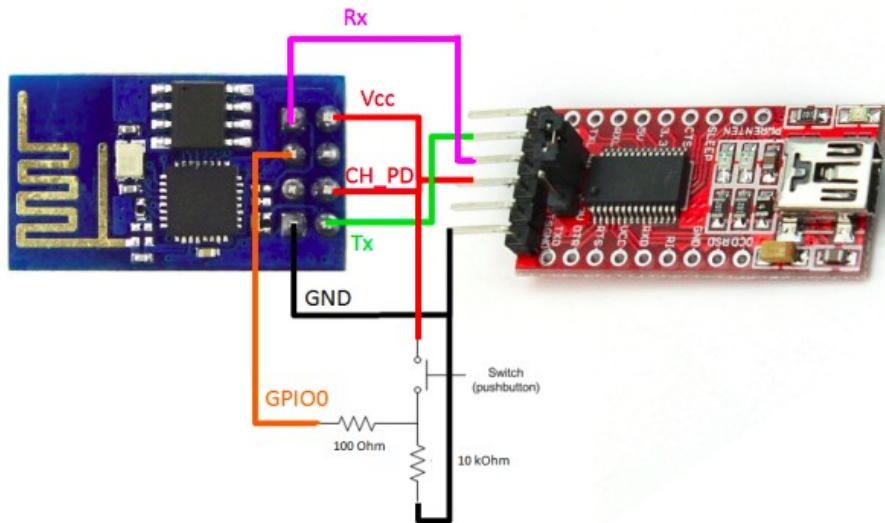
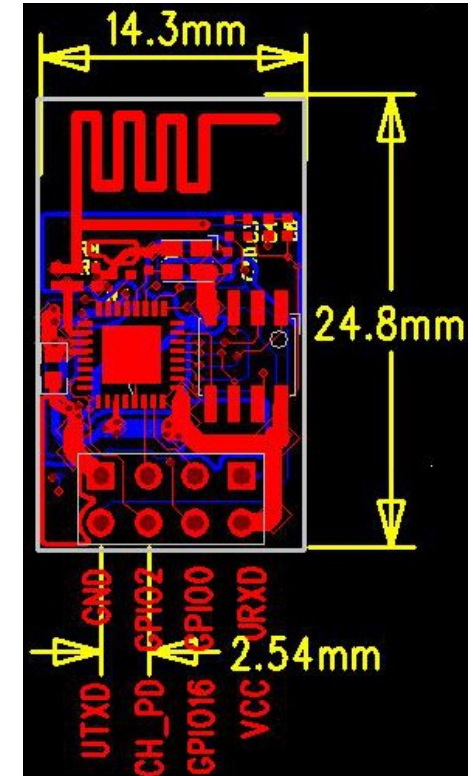
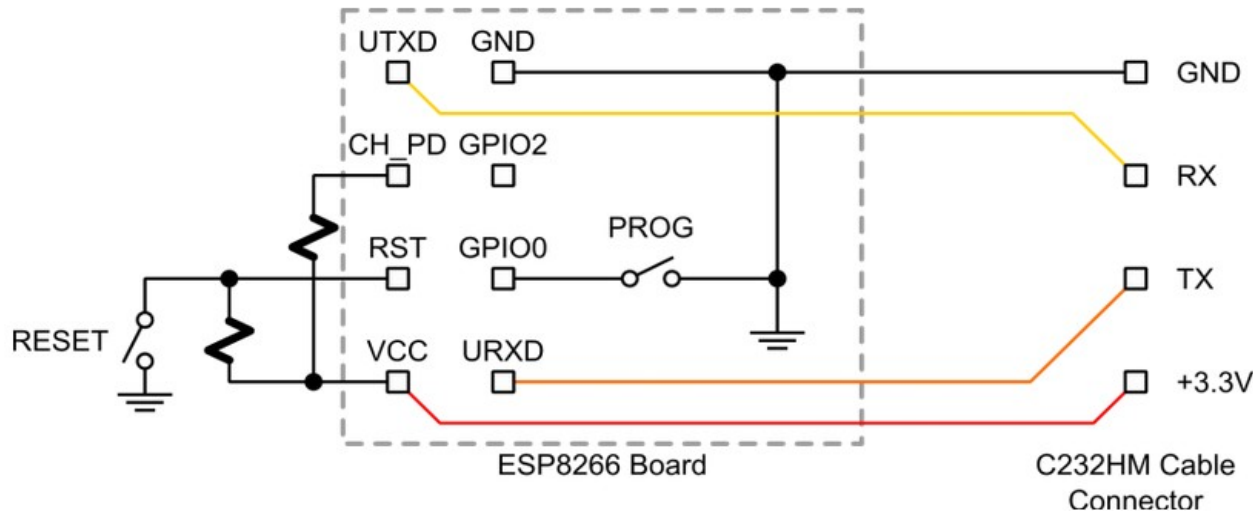
Algunes plaques amb l'ESP8266





ESP-12E Development Board

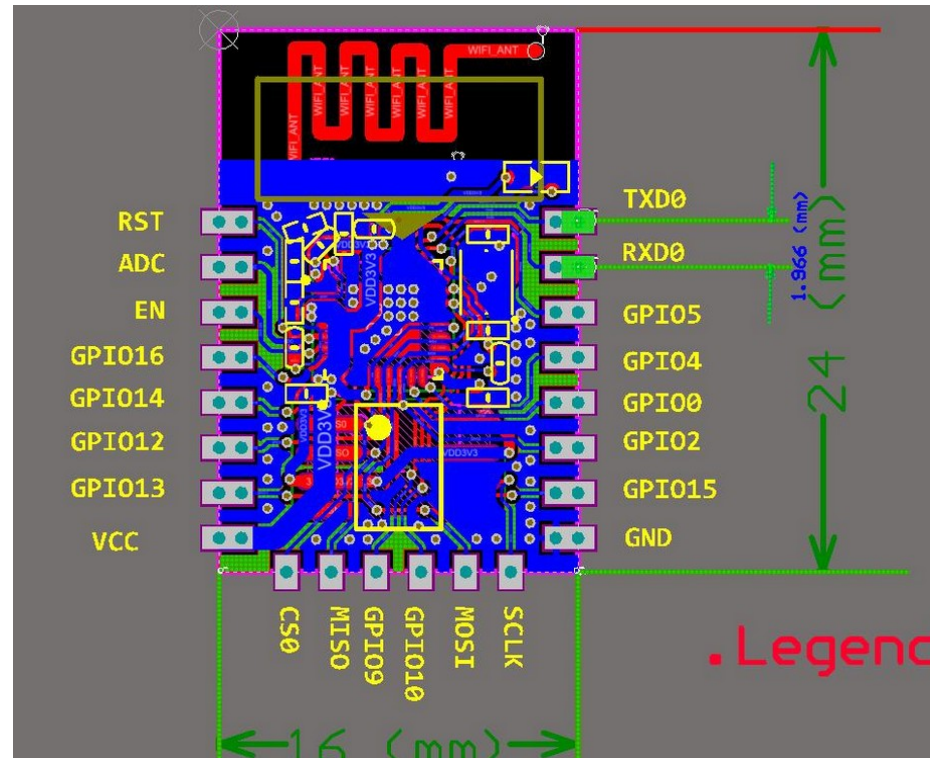
Programant l'ESP8266 - ESP01





ESP-12E Development Board

Programant l'ESP8266 - ESP12E





ESP-12E Development Board

Fabricant del xip: Espressif

ESP8266 | Espressif 乐鑫 - Iceweasel

ESP8266 | Espressif ...

espressif.com/en/products/esp8266/

Cerca

ESPRESSIF

IoT Solutions Products Downloads Espressif English 简体中文

Introduction Features Solutions

Details

The ESP8266

The ESP8266 is a highly integrated chip designed for the needs of an increasingly connected world. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host applications or offload all Wi-Fi networking functions from another application processor.

The ESP8266 has powerful on-board processing and storage capabilities that allow it to be used with sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry. The entire solution, including the module, is designed to occupy minimal PCB area.

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



ESP-12E Development Board

Caractéristiques de l'ESP8266

Introduction	Features	Solutions	Details
<h3>Features</h3> <ul style="list-style-type: none">• 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 <math><10\mu\text{A}</math>
- 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 <math>< 2\text{ms}</math>
- Standby power consumption of <math>< 1.0\text{mW}</math> (DTIM3)

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



ESP-12E Development Board

Datasheet de l'ESP8266

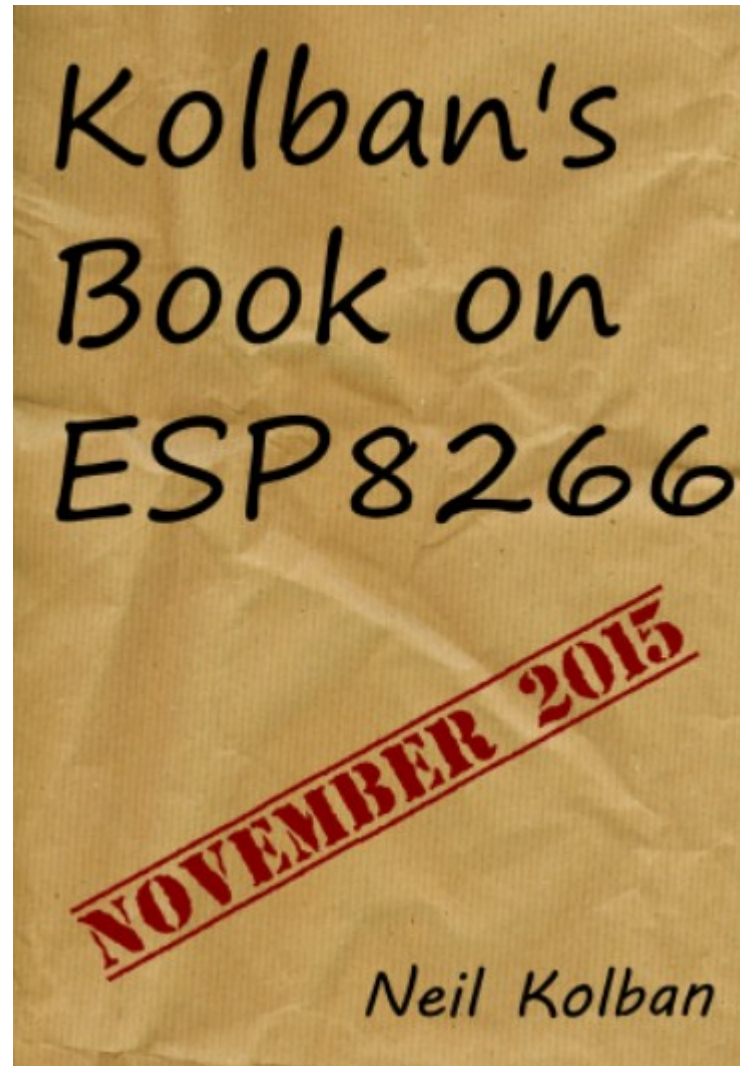


Adafruit's link to ESP8266EX Datasheet v4.3



ESP-12E Development Board

Kolban's Book on ESP8266

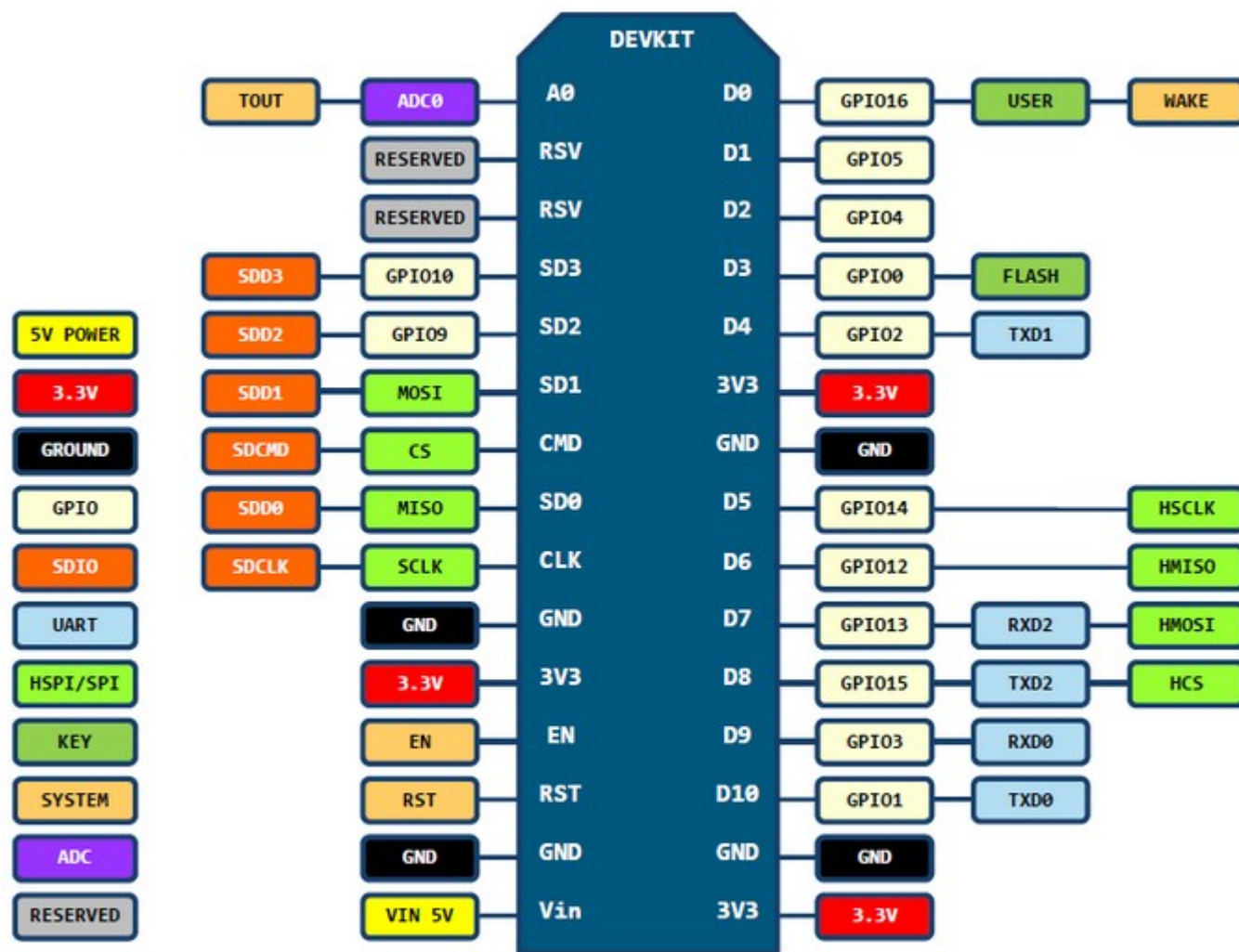
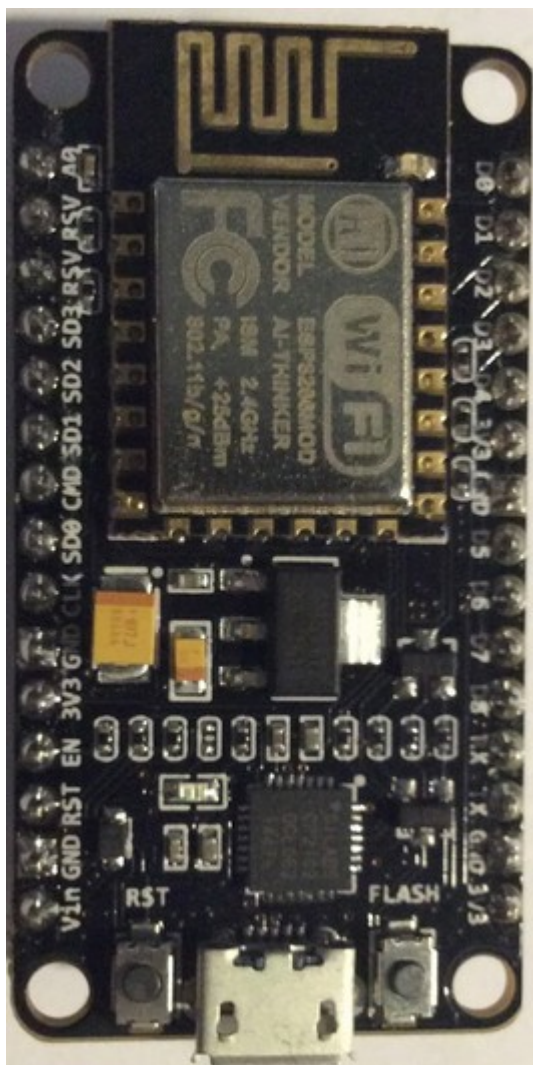


<http://neilkolban.com/tech/esp8266/>



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
```

Seqüència 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 0x000000 v0.9.5.2\ AT\ Firmware.bin
```

```
Connecting...
```

```
Erasing flash...
```

```
Wrote 520192 bytes at 0x00000000 in 49.4 seconds (84.3 kbit/s)...
```

```
Leaving...
```

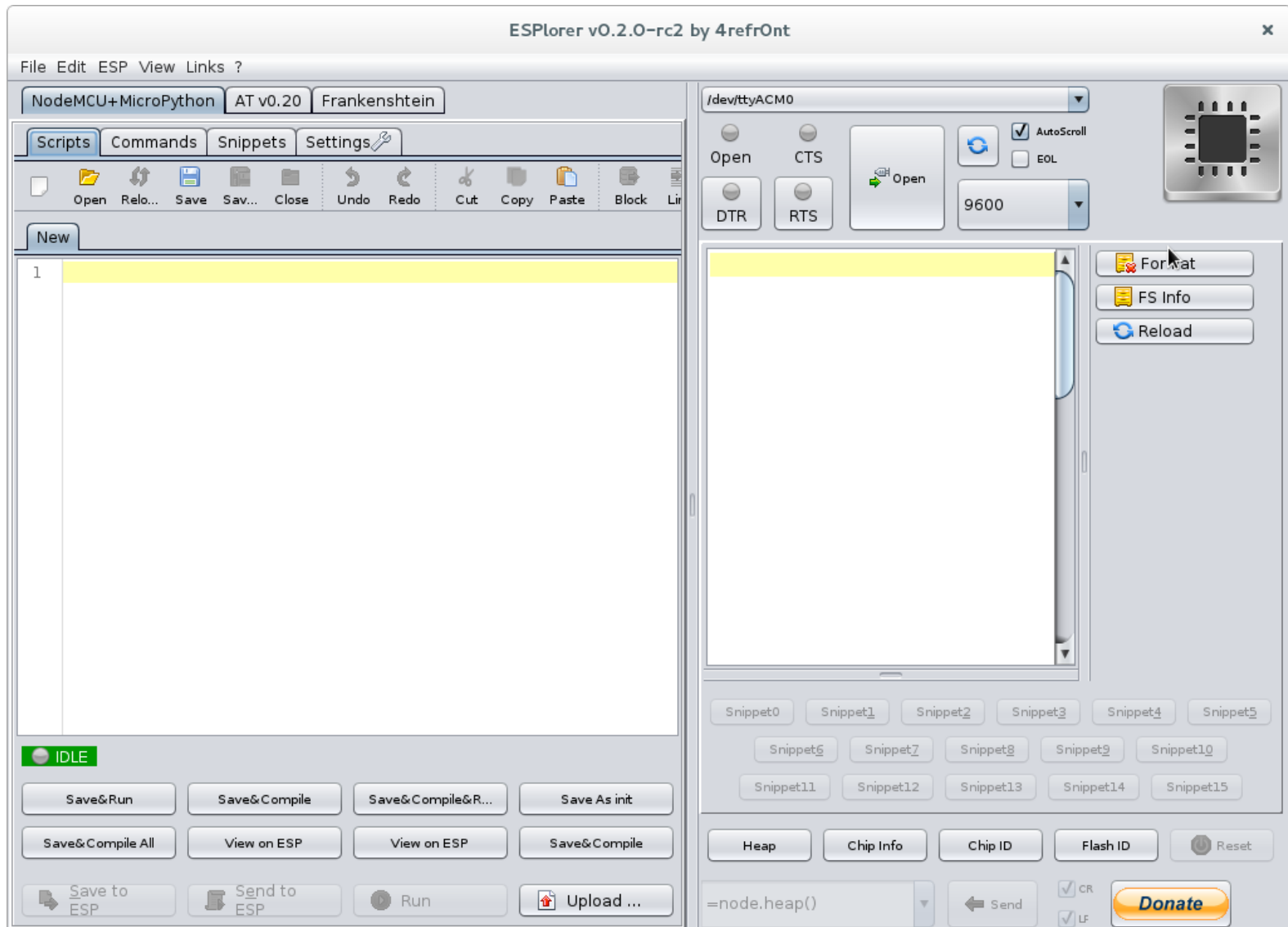
```
root@debian8:/home/ecat/Documents/dam/uf3/esptool-master#
```



ESP-12E Development Board

Eines - ESPlorer

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



<http://esp8266.ru/esplorer/>



ESP-12E Development Board

Eines - ESPlorer

ESPlorer v0.2.0-rc2 by 4refrOnt

File Edit ESP View Links ?

NodeMCU+MicroPython AT v0.20 Frankenshtein

Basic AT commands

AT RST GMR GSLP ATE0 ATE1 UPD

WiFi Station WiFi softAP TCP/IP client TCP/IP Server

Common WiFi commands

CWMODE=? - Get available... CWMODE=1 Station
 CWMODE=? - Get current m... CWMODE=2 softAP
 CWLAP - Get AP list CWMODE=3 softAP + Stati...
 0 - Enable ... 1 - Set ... AT+CWDHCP DHCP control

WiFi Station

CWJAP? - Connection info SSID password
 CWQAP - Disconnect fr... CWJAP Connect to AP
 CIPSTAMAC? Get MAC
 FF:FF:FF:FF:FF:FF
 CIPSTAMAC= Set MAC S...
 CIPSTA? Get Station IP
 192.168.1.50
 CIPSTA= Set Station IP

/dev/ttyUSB0

Open CTS Close AutoScroll EOL
 DTR RTS 115200

```

PORT OPEN 115200

Communication with MCU...
Got answer! AutoDetect firmware...

AT-based firmware detected.
AT+GMR
AT version:0.21.0.0
SDK version:0.9.5

OK
  
```

AT Send CR LF Donate



ESP-12E Development Board

Eines - ESPlorer

NodeMCU+MicroPython AT v0.20 Frankenshtein

Basic AT commands

AT RST GMR GSLP ATE0 ATE1 UPD

WiFi Station WiFi softAP TCP/IP client TCP/IP Server

Common WiFi commands

CWMODE=? - Get available... CWMODE=1 Station

CWMODE? - Get current m... CWMODE=2 softAP

CWLAP - Get AP list CWMODE=3 softAP + Stati...

0 - Enable ... 1 - Set ... AT+CWDHCP DHCP control

WiFi Station

CWJAP? - Connection info IoT-eCat clotClot

CWQAP - Disconnect fr... CWJAP Connect to AP

CIPSTAMAC? Get MAC

FF:FF:FF:FF:FF:FF

CIPSTAMAC= Set MAC S...

CIPSTA? Get Station IP

192.168.1.50

CIPSTA= Set Station IP

/dev/ttyUSB0

Open CTS Close AutoScroll EOL

DTR RTS 115200

```

AT+CWMODE?
+CWMODE:2

OK
AT+CWMODE=1

OK
AT+CWJAP?
No AP

OK
AT+CWJAP="IoT-eCat","clotClot"

OK
AT+CIPSTAMAC?
+CIPSTAMAC:"18:fe:34:06:28:8f"

OK
AT+CIPSTA?
+CIPSTA:"192.168.1.55"

OK

```



ESP-12E Development Board

Programant l'ESP8266 - LUA - Eines

Seqüència 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/>

<https://github.com/nodemcu/nodemcu-firmware/releases>



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

```

/dev/ttyUSB0
Open CTS Close AutoScroll
DTR RTS 9600 EOL

Can't autodetect firmware, because proper an
stdin:1: unexpected symbol near ''''
>
> wifi.sta.config("IoT-eCat","clotClot")
> =wifi.ap.getmac()
5e:cf:7f:06:28:8f
> =wifi.sta.getip()
nil
> =wifi.getmode()
2
> wifi.setmode(wifi.STATION)
> wifi.sta.config("IoT-eCat","clotClot")
> =wifi.sta.getip()
192.168.1.50 255.255.255.0 192.168.1.1
> =wifi.sta.status()
5
>

```




ESP-12E Development Board

Programant l'ESP8266 - LUA

The screenshot displays the MicroPython IDE interface. The left pane shows a Lua script named 'clientWeb.lua' with the following code:

```

1  wifi.setmode(wifi.STATION)
2  wifi.sta.config("IoT-eCat","clotClot")
3  print(wifi.sta.getip())
4  led1 = 4
5  led2 = 5
6  gpio.mode(led1, gpio.OUTPUT)
7  gpio.mode(led2, gpio.OUTPUT)
8  srv=net.createServer(net.TCP)
9  srv:listen(80,function(conn)
10     conn:on("receive", function(client,request)
11         local buf = ""
12         local _, _, method, path, vars = string.find(request, "([A-Z]+) (.*) (%*)")
13         if (method == nil) then
14             _, _, method, path = string.find(request, "([A-Z]+) (.*)")
15         end
16         local _GET = {}
17         if (vars ~= nil) then
18             for k, v in string.gmatch(vars, "(%w+)=(%w+)&.*") do
19                 _GET[k] = v
20             end
21         end
22     end
23 end

```

The right pane shows the terminal output for the /dev/ttyUSB0 device. The output indicates that no files were found initially, followed by memory statistics and successful uploads of 'clientWeb.lua' and 'closeTcp.lua' to the ESP. A subsequent 'dofile' command is shown with IP addresses: 192.168.1.50, 255.255.255.0, and 192.168.1.1.

The IDE interface includes a menu bar (Scripts, Commands, Snippets, Settings), a toolbar with standard editing actions, and a status bar at the bottom showing 'IDLE' and the file path '/home/ecat/Documents/dam/uf3/luaCode/clientWeb.lua'. The bottom right corner features a 'Donate' button.

<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_output.
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 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)
```

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")
```

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)
```



ESP-12E Development Board

Programant l'ESP8266 - microPython - Eines

Seqüència 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 write_flash 0x000000 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

The screenshot shows a microPython IDE interface. On the left, a script editor displays the following code:

```

1 import pyb
2 pin = pyb.Pin(14, pyb.Pin.OUT_PP)
3 pin.high()
4 print('Estic fent servir Python !!!!')
5

```

On the right, the serial terminal shows the output of the script:

```

rl l00| 0l0|< 0 l0 b|00<04;r0b0 b00nn0lnn000 b0p00lrirlp0n0<< 0 l
0rl00rl000b l0bp|0 0z0
>>>
Micro Python v1.4.4-12-ga193ced on 2015-06-20; ESP module with ESP8266
Type "help()" for more information.
>>> import pyb
>pin = pyb.Pin(14, pyb.Pin.OUT_PP)
>pin.high()
>print('Estic fent servir Python !!!!')
Estic fent servir Python !!!!
>>>

```

<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

Seqüència 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 0x000000 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

ESP8266 Serial WiFi Shield – Iceweasel ×

ESP8266 Serial WiFi ... × +

← 192.168.4.1 ↻ >> ☰

ESP8266 Serial WiFi Shield

Serial Setting

Baud :

Parity :

Stopbits :

Access Point(AP) :

AP name :

AP Password :

Encrypt Method :

Hide AP : Yes No

AP IP address :

AP Netmask :

AP Gateway address :

Station :

Enable : Yes No

AP Name :

AP List :

AP Password :

DHCP Enable : Yes No

STA IP address :

STA Netmask :

STA Gateway address :

Network Setting

Socket Type : Server Client

Transport Type : TCP UDP

Remote IP :

Local/Remote Port :

Version 2.4 based on ESP_IOT_SDK_v1.4.0

©Copyright 2015 www.doit.am

QQ Group: 43923759



ESP-12E Development Board

Programant l'ESP8266 - WiFi a sèrie - Eines

ESP8266 Serial WiFi Shield

Serial Setting:

Baud :

Databits:

Parity:

Stopbits:

Access Point(AP) :

AP name:

AP Password:

Encrypt Method:

Hide AP: Yes No

AP IP address:

AP Netmask:

AP Gateway address:

NetWork Setting:

Socket Type: Server Client

Transport Type: TCP UDP

Remote IP:

Local/Remote Port:

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

NetWork Setting:

Socket Type: Server Client

Transport Type: TCP UDP

Remote IP:

Local/Remote Port:

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

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

192.168.4.2 : 6000 x

IP: . Port:

Text:

Received text

```
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 Serial WiFi Shield

Serial Setting:

Baud :

Databits:

Parity:

Stopbits:

Access Point(AP) :

AP name:

AP Password:

Encrypt Method:

Hide AP: Yes No

AP IP address:

AP Netmask:

AP Gateway address:

NetWork Setting:

Socket Type: Server Client

Transport Type: TCP UDP

Remote IP:

Local/Remote Port:

ESP8266 as UDP client



ESP-12E Development Board

Programant l'ESP8266 - WiFi a sèrie - Eines

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

192.168.4.2 : 6000

IP : 192.168.4 .1 Port : 9000

Text :

Received text

Network setting:

Socket Type: Server Client

Transport Type: TCP UDP

Remote IP:

Local/Remote Port:

```
ecat@debian8: ~/Documents/dam/uf3/udp/qtPyUdp
Fitxer Edita Visualitza Cerca Terminal Ajuda
ecat@debian8:~/Documents/dam/uf3/udp/qtPyUdp$ ./qtPyUdp.py 6000
Gtk-M
```

192.168.4.2 : 6000

IP : 192.168.4 .1 Port : 9000

Text :

Received text

sketch_jan22a | Arduino 2:1

Fitxer Edita Sketch Eines Ajuda

Desplaçament automàtic Sense salts de línia 9600 baud

```
ecat@debian8: ~/Documents/dam/uf3/udp/qtPyUdp
Fitxer Edita Visualitza Cerca Terminal Ajuda
ecat@debian8:~/Documents/dam/uf3/udp/qtPyUdp$ ./qtPyUdp.py 6000
Gtk-Message: Failed to load module "canberra-gtk-mod
data : A10 , len : 3, size : 3
```

sketch_jan22a | Arduino 2

Fitxer Edita Sketch Eines Ajuda

Desplaçament automàtic Sense salts de línia 9600 baud

192.168.4.2 : 6000

IP : 192.168.4 .1 Port : 9000

Text :

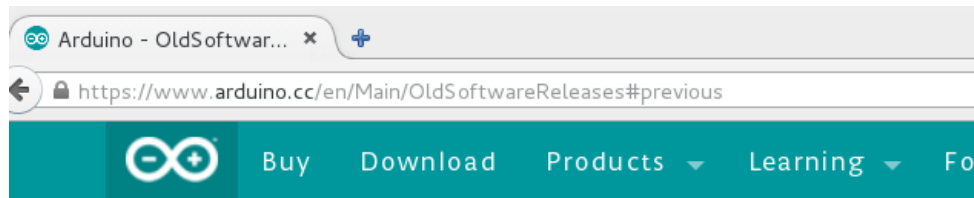
A10



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>



ARDUINO 1.6.5

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.
[See the release notes.](#)

Windows [Installer](#)
Windows [ZIP file for non admin install](#)

Mac OS X [10.7 Lion or newer](#)

Linux [32 bits](#)
Linux [64 bits](#)

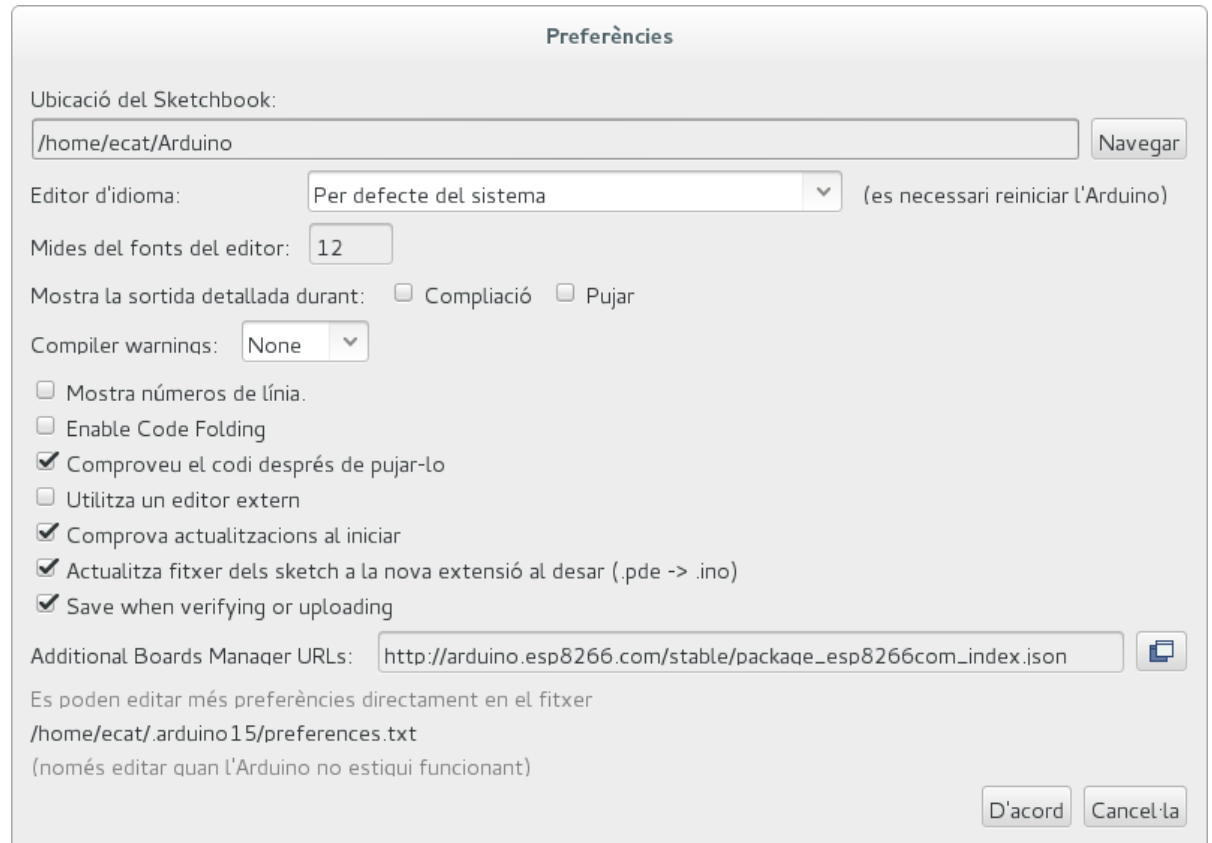
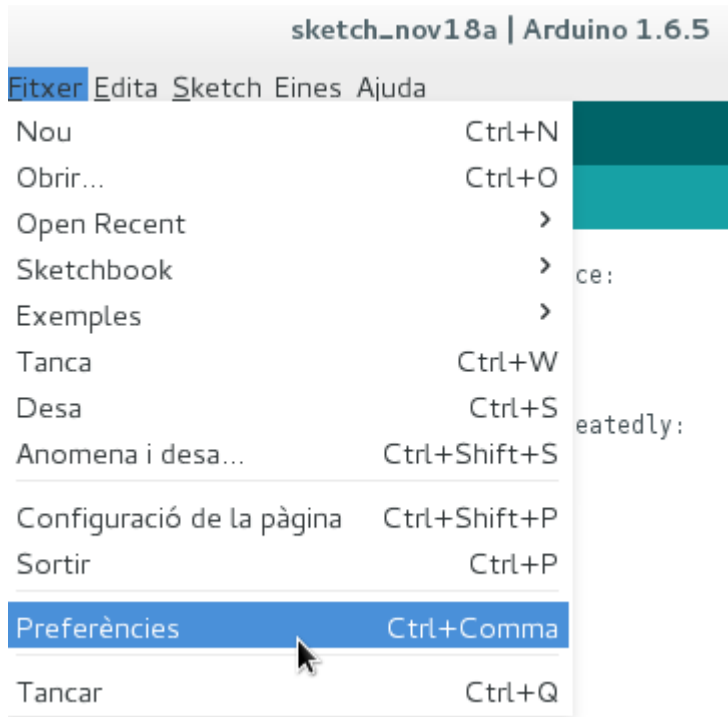
[Source](#)





ESP-12E Development Board

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



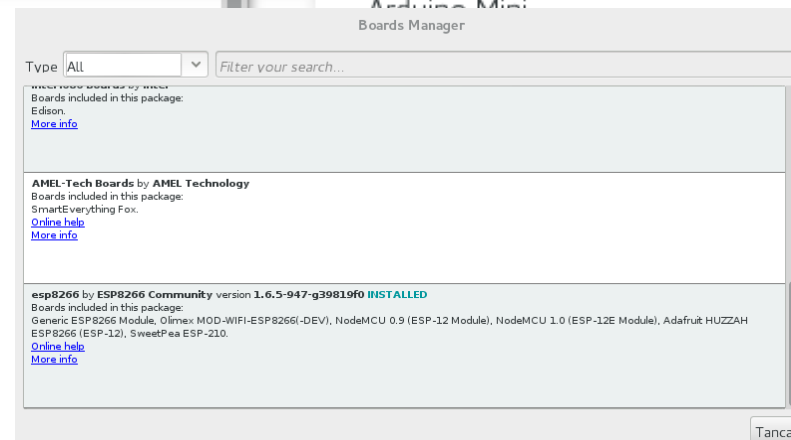
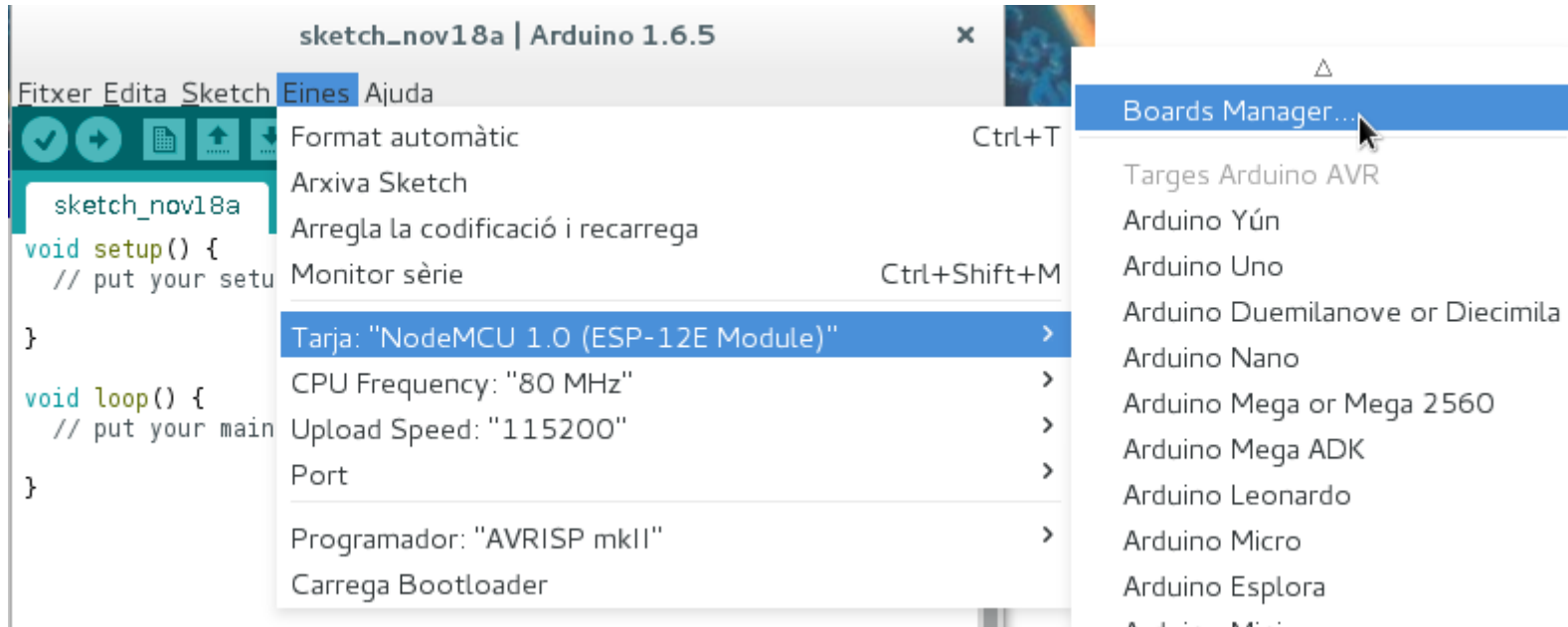
Entreu aquesta adreça a "Additional Boards Manager URLs":
http://arduino.esp8266.com/stable/package_esp8266com_index.json



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

Tramitent trames UDP amb l'ESP8266

```

UDPCliantMAC02 | Arduino 1.6.5
Fitxer Edita Sketch Eines Ajuda
✓ → 📄 ⬆️ ⬇️
UDPCliantMAC02
#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

```

```

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");

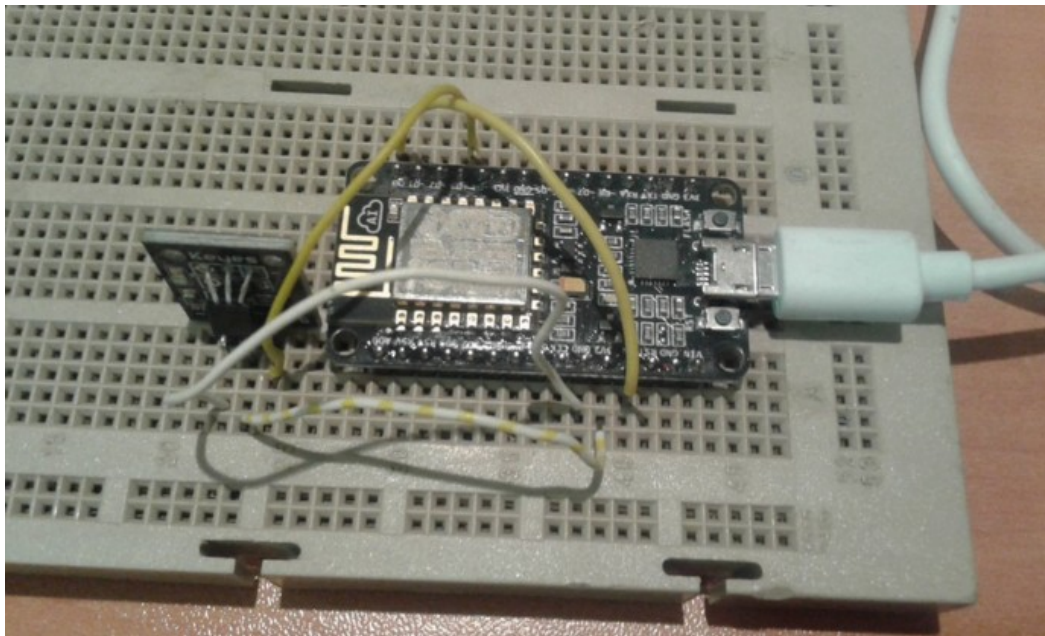
  delay(5000);
}

```



ESP-12E Development Board

ESP8266 sobre l'IDE d'Arduino - MQTT



```

Fitxer  Edita  Vi
Client mosqsub/15509-debian8 Entering deep sleep mode for 3 seconds...
Client mosqsub/15509-debian8 r l l00| 0l0|≤ 0 l0 b|00-0:0r0b0 b00nn0lnn000 b0p00lrlrlp0n
24.94 Connecting to IoT-eCat
Client mosqsub/15509-debian8 .....
25.00 WiFi connected
Client mosqsub/15509-debian8 IP address:
24.94 192.168.1.12
Client mosqsub/15509-debian8 Attempting MQTT connection...connected
Client mosqsub/15509-debian8 Requesting DS18B20 temperature...
25.00 Sending temperature: 24.69
Client mosqsub/15509-debian8 Closing MQTT connection...
Client mosqsub/15509-debian8 Closing WiFi connection...
Client mosqsub/15509-debian8 Entering deep sleep mode for 3 seconds...
Client mosqsub/15509-debian8 r l l00| 0l0|≤ 0 l0 b|00-0:0r0b0 b00nn0lnn000 b0p00lrlrlp0n
25.00 Connecting to IoT-eCat
Client mosqsub/15509-debian8 .....
25.06
^C
acat@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
 Desplaçament automàtic

```



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-ix106-*generats, podeu afegir la ruta del SDK al PATH. Podeu actualitzar el PATH cada vegada que iniciu 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.