

Εγκατάσταση micropython σε ESP 8266 (nodemcu) - Linux

Αρχικά εγκαθιστούμε το **esptool.py** στην Python 3 με την εντολή `pip3 install esptool` ή σε Python 2 με `pip install esptool`. Πρέπει να έχετε εγκαταστήσει το `pip3` ή το `pip` για την αντίστοιχη έκδοση της python. Για debian είναι τα πακέτα `python3-pip` και `python-pip` αντίστοιχα.

Για να βρούμε τον κατάλογο εγκατάστασης του καταλόγου `site-packages` γράφουμε `python3 -m site`. Για debian μας ενδιαφέρει το `~/.local/lib/python3.5/site-packages`. Πάμε σ' αυτή την διαδρομή και δίνουμε `python3 esptool.py` για να δούμε αν λειτουργεί. Αν θέλουμε μπορούμε να το βάλουμε στο path. Συνδέουμε το nodemcu με ένα καλώδιο σε μια θύρα USB του υπολογιστή μας. Με `sudo dmesg | grep tty` βρίσκουμε τις διαθέσιμες σειριακές θύρες του συστήματός μας.

```
[ 0.000000] console [tty0] enabled
[ 1.121904] 00:04: ttyS0 at I/O 0x3f8 (irq = 4, base_baud = 115200) is a 16550A
[ 23.770219] systemd[1]: Created slice system-getty.slice.
[33120.106174] usb 2-5: cp210x converter now attached to ttyUSB0
```

Επειδή η πρώτη είναι η ενσωματωμένη της μητρικής κάρτας, εμάς μας αφορά η δεύτερη που είναι USB με όνομα **ttyUSB0**.

Γράφουμε μέσα από τον κατάλογο `~/.local/lib/python3.5/site-packages :`

`python3 esptool.py --port /dev/ttyUSB0 flash_id`

και αν όλα πάνε καλά μας απαντάει το μέγεθος της μνήμης flash.

```
esptool.py v2.6
Serial port /dev/ttyUSB0
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
MAC: 84:f3:eb:18:34:3b
Uploading stub...
Running stub...
Stub running...
Manufacturer: 20
Device: 4016
Detected flash size: 4MB
Hard resetting via RTS pin...
```

Σβήνω το εγκατεστημένο firmware της flash με την εντολή :

`python3 esptool.py --port /dev/ttyUSB0 erase_flash`

και μας απαντάει:

```
esptool.py v2.6
Serial port /dev/ttyUSB0
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
MAC: 84:f3:eb:18:34:3b
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 7.9s
Hard resetting via RTS pin...
```

Έπειτα κατεβάζουμε την τελευταία σταθερή έκδοση της micropython από το επίσημο site <http://micropython.org/download#esp8266> . Η παρούσα έκδοση είναι η 20190125-v1.10.

Προγραμματίζουμε το νέο firmware που κατεβάσαμε με την εντολή:

`python3 esptool.py --port /dev/ttyUSB0 --baud 115200 write_flash --flash_size=detect 0 ~/Downloads/esp8266-20190125-v1.10.bin`

Και απαντάει:

```
esptool.py v2.6
Serial port /dev/ttyUSB0
Connecting....
```

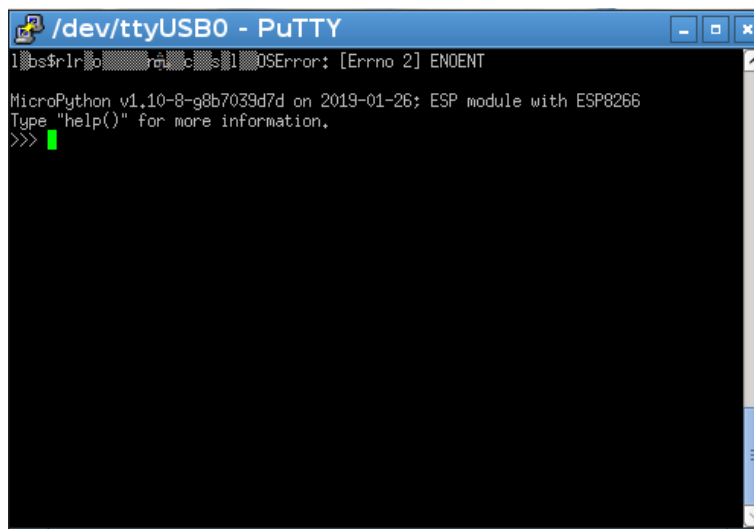
```

Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
MAC: 84:f3:eb:18:34:3b
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0040
Compressed 615388 bytes to 399928...
Wrote 615388 bytes (399928 compressed) at 0x00000000 in 35.5 seconds (effective 138.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

```

Ανοίγουμε ένα πρόγραμμα σειριακού τερματικού όπως το **putty** ή το **picocon** και δίνουμε σειριακή θύρα /dev/ttyUSB0 και ρυθμό 115200bps. Αφού ανοίξει πατάμε το κουμπί reset της πλακέτας και εμφανίζει:



```

/dev/ttyUSB0 - PuTTY
l bs$rlr o  rā c s l OSErrno: [Errno 2] ENOENT

MicroPython v1.10-8-g8b7039d7d on 2019-01-26; ESP module with ESP8266
Type "help()" for more information.
>>>

```

Η για picocon:

και τερματίζουμε με ctrl + A + X.

Εξετάζουμε τα αρχεία που υπάρχουν μέσα στο σύστημα αρχείων:

```

>>> import os
>>> os.listdir()
['boot.py']

```

Για να εξετάσουμε τον ελεύθερο χώρο του συστήματος αρχείων γράφουμε:

```

import os
os.statvfs("/")
και επιστρέφει:
(4096, 4096, 860, 859, 859, 0, 0, 0, 0, 255)

```

το πρώτο είναι το block size και το τρίτο είναι ο συνολικός χώρος σε blocks ενώ το τέταρτο είναι ο ελεύθερος χώρος σε blocks. Για κάθε αρχείο από 1 έως 4096 bytes δεσμεύει ένα block. Αν το αρχείο ξεπεράσει για λίγο το 4096 θα πάρει και άλλο block. Από τα 1024 συνολικά blocks, τα 164 τα δεσμεύει η micropython περίπου 610Kbytes. Δηλαδή τώρα έχω ελεύθερα 859*4096=3518464 bytes.

Για να δω τα εγκατεστημένα modules γράφω help('modules') για το esp8266

```

>>> help('modules')
__main__      http_client   socket        upip
__boot__      http_client_ssl  ssl1306       upip_utarfile

```

<code>_onewire</code>	<code>http_server</code>	<code>ssl</code>	<code>upysh</code>
<code>_webrepl</code>	<code>http_server_ssl</code>	<code>struct</code>	<code>urandom</code>
<code>apa102</code>	<code>inisetup</code>	<code>sys</code>	<code>ure</code>
<code>array</code>	<code>io</code>	<code>time</code>	<code>urequests</code>
<code>binascii</code>	<code>json</code>	<code>uasyncio/__init__</code>	<code>urllib/urequest</code>
<code>btree</code>	<code>lwip</code>	<code>uasyncio/core</code>	<code>uselect</code>
<code>builtins</code>	<code>machine</code>	<code>ubinascii</code>	<code>usocket</code>
<code>collections</code>	<code>math</code>	<code>ucollections</code>	<code>ussl</code>
<code>dht</code>	<code>micropython</code>	<code>ucryptolib</code>	<code>ustruct</code>
<code>ds18x20</code>	<code>neopixel</code>	<code>uctypes</code>	<code>utime</code>
<code>errno</code>	<code>network</code>	<code>uerrno</code>	<code>utimeq</code>
<code>esp</code>	<code>ntptime</code>	<code>uhashlib</code>	<code>uzlib</code>
<code>example_pub_button</code>		<code>onewire</code>	<code>uheapq</code>
<code>example_sub_led</code>	<code>os</code>	<code>uio</code>	<code>webrepl_setup</code>
<code>flashbdev</code>	<code>port_diag</code>	<code>ujson</code>	<code>websocket</code>
<code>framebuf</code>	<code>random</code>	<code>umqtt/robust</code>	<code>websocket_helper</code>
<code>gc</code>	<code>re</code>	<code>umqtt/simple</code>	<code>zlib</code>
<code>hashlib</code>	<code>select</code>	<code>uos</code>	

Plus any modules on the filesystem

Εγκατάσταση micropython σε ESP 8266 (nodemcu) – Windows

Για την εγκατάσταση της micropython στο nodemcu, ακολουθούμε τα ίδια βήματα και από Λ.Σ. Windows.

Για τον προγραμματισμό υπάρχει το IDE EsPy <https://github.com/jungervin/EsPy>