

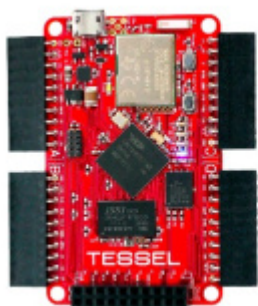
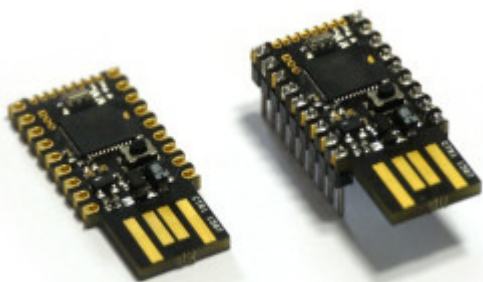
# Running Javascript on ESP8266 (NodeMCU) ใช้งานจาวาสคริปต์บน ESP8266

 HTML Content

## Running Javascript on ESP8266 (NodeMCU) ใช้งาน จาวาสคริปต์บน ESP8266

สวัสดีครับวันนี้เราลองมาใช้งานจาวาสคริปต์ Javascript บน ESP8266 กันดูครับ ในบทความนี้ผมใช้ NodeMCU v0.9 ในการทดลองครับ

เพื่อนๆคงจะเคยใช้ ESP8266 กับภาษา Lua หรือ C/C++ กับ Arduino IDE กันเป็นส่วนใหญ่ เพราะค่อนข้างเข้าใจง่าย มีเครื่องมือและบทความต่างๆมากมายให้ศึกษากัน แต่ถ้าบอกว่าเรามารัน JavaScript บนESP8266กันดูก็คงบอกว่าทำได้หรือไม่ มีความจำเป็นมากน้อยแค่ไหนจริงๆแล้วต้องขึ้นอยู่กับนักพัฒนาว่าถนัดในภาษาใด ความยืดหยุ่น ความเร็วในการทำงาน ซึ่งอาจจะยากง่ายแตกต่างกันไปครับ โดยส่วนตัวผมว่านี่ก็เป็นทางเลือกหนึ่งครับ เพราะภาษา JavaScript ก็มีความนิยมและใช้แพร่หลายกันมากเป็นคอร์เทคโนโลยีหลักในการพัฒนาเว็บไซต์ อีกทั้งอาจจะง่ายสำหรับผู้เริ่มต้นศึกษาการเขียนโปรแกรมมือใหม่ แหล่งเรียนรู้มากมาย เช่นที่ Code Academy <https://www.codecademy.com/learn/javascript>



JavaScript สำหรับบอร์ด Microcontroller นั้นโดยเริ่มแรกมีการถูกพัฒนาเพื่อใช้งานบนบอร์ด Espruino (<http://www.espruino.com/Get+it>) และ Tessel (<https://tessel.io/>) แต่ก็ยังไม่ได้รับความนิยมอย่างกว้างขวางเหมือน Arduino หรือ ESP8266 เหล่าสาวก Espruino มีการเรียกร่องผู้พัฒนา Espruino ([Gordon Williams](#)) ว่าอยากจะใช้งาน JavaScript บน ESP8266 ... ขอมาฝากท่านก็เลยจัดให้ (ใจกว้างมากตั้งแม่น้ำเทมส์ ณ.กรุงลอนดอน)

เรามาเริ่มเตรียมเครื่องมือกันเลยครับ (Installing tools)

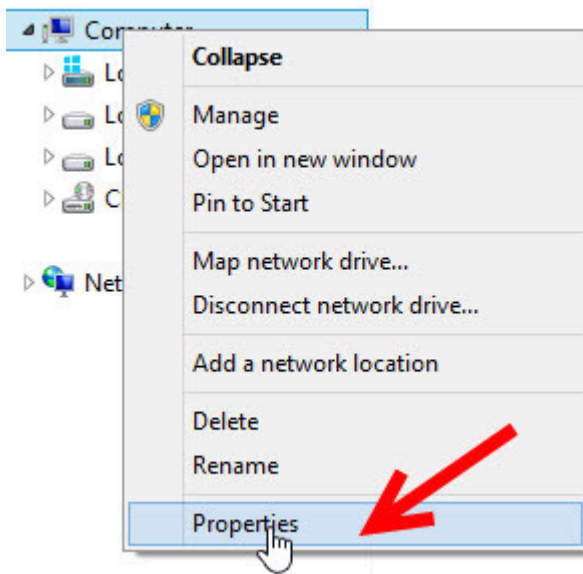
1. Python 2.7.xx <https://www.python.org/downloads/> ในที่นี้ผมใช้ Python 2.7.11 สำหรับ Python 3.5.xx ไม่สามารถใช้ได้นะครับ (ผมลองมาแล้วฮา..)
2. Git for Windows <https://git-scm.com/download> บทความนี้เรา Flash จาก Windows 8
3. Espruino build for Esp8266  
<http://forum.espruino.com/conversations/279176/>
4. [Espruino Web IDE](#) โปรแกรมที่ใช้เขียนโปรแกรมลงบนบอร์ดเรา ในที่นี้เราจะใช้ผ่าน Chrome เว็บเบราว์เซอร์
5. ESP8266 board ผมใช้ [NodeMCU](#) v.0.9 แต่เท่าที่ดู EPS8266-01 ก็ทำได้ครับ

### ขั้นตอน (Flashing process)

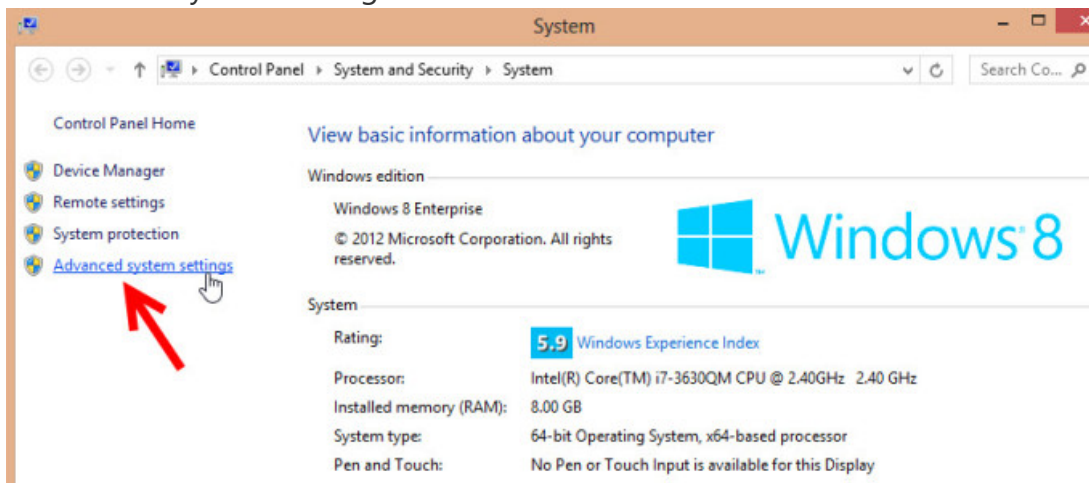
#### Install Python

1. ดาวน์โหลด Python 2.7.xx <https://www.python.org/downloads/>

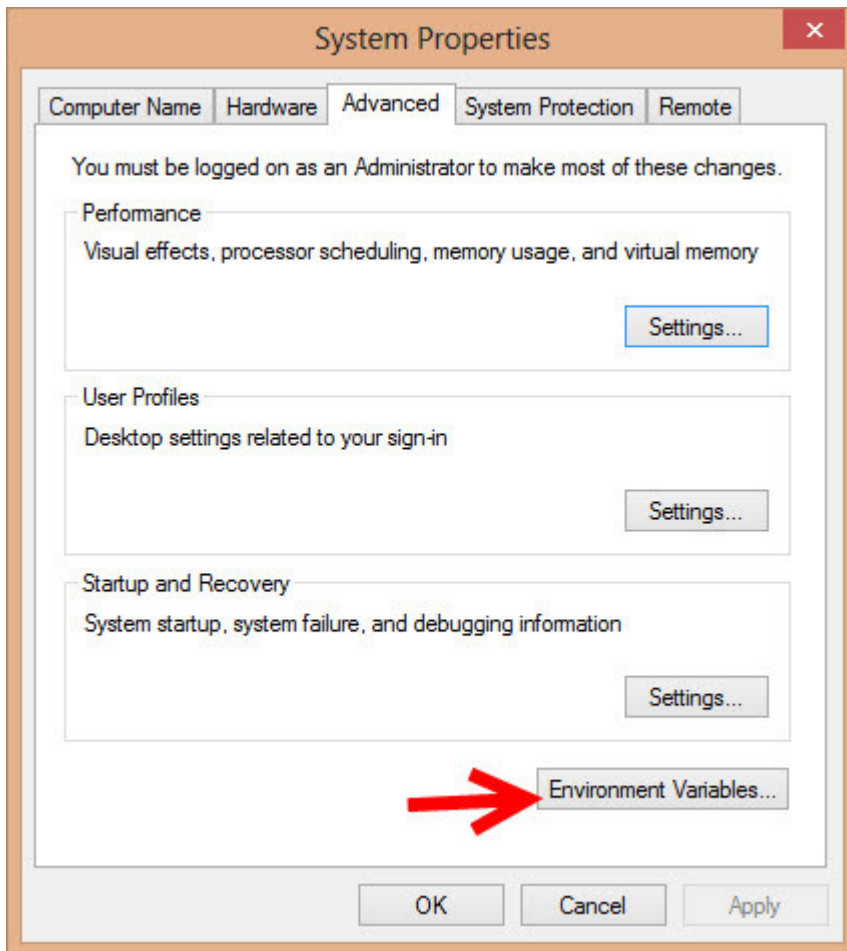
### 3. ไปที่ Computer > Properties



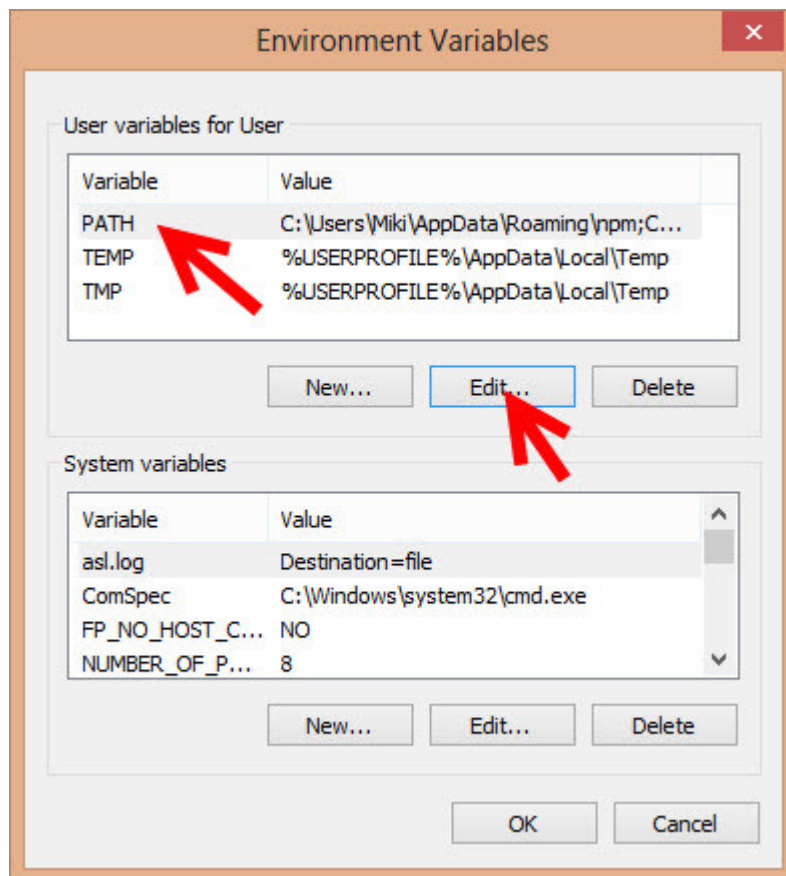
### 4. > Advanced system setting



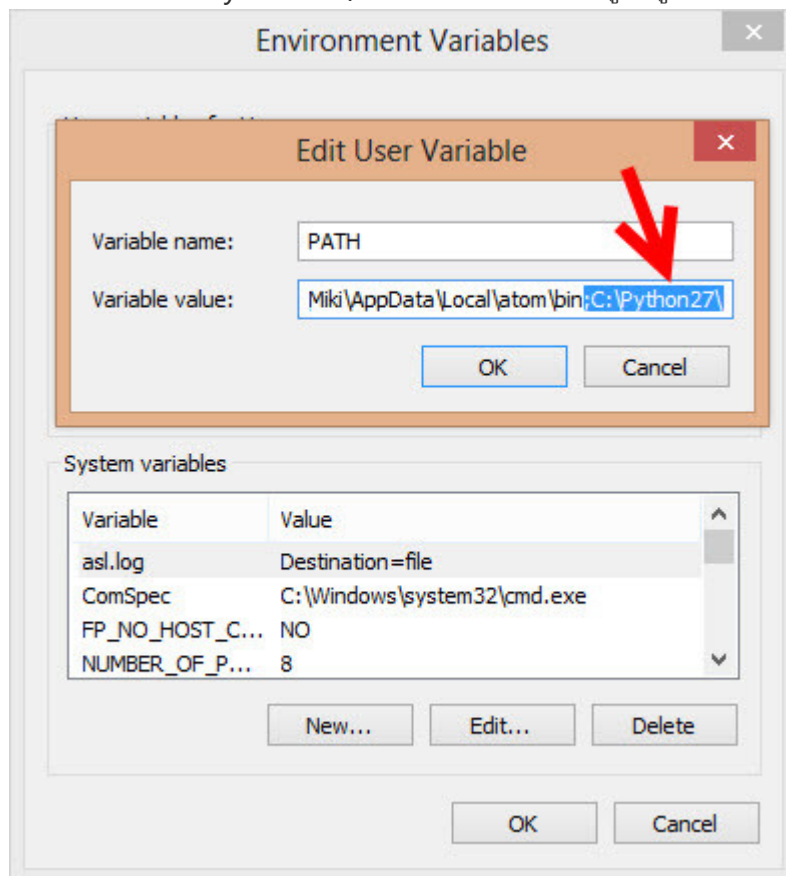
5. > Environment Variables...



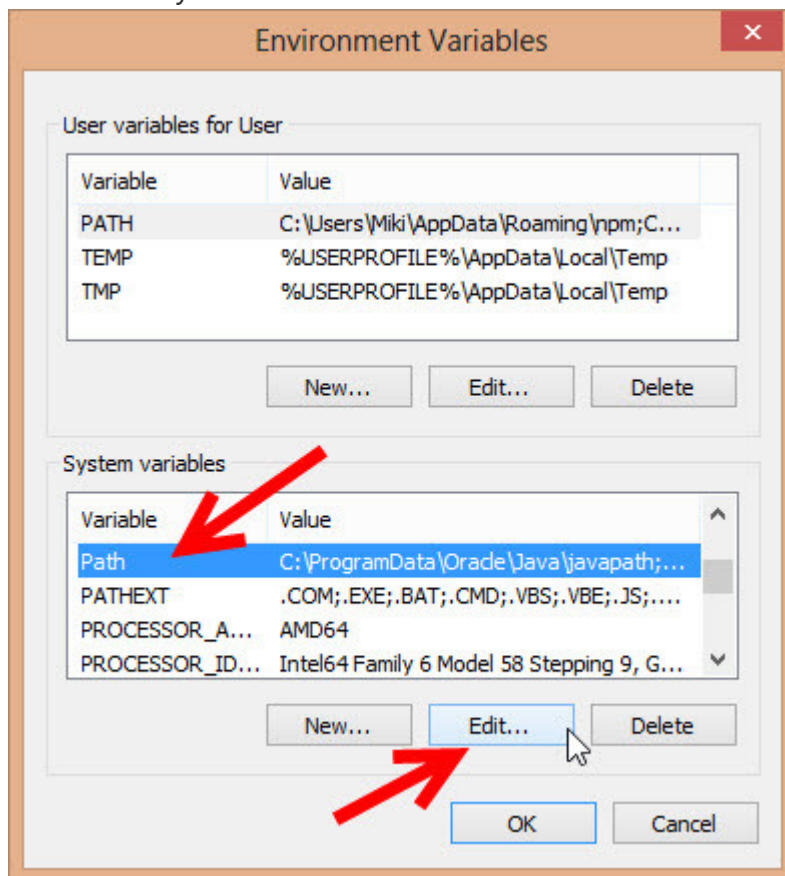
## 6. แก้ไขที่ PATH variable



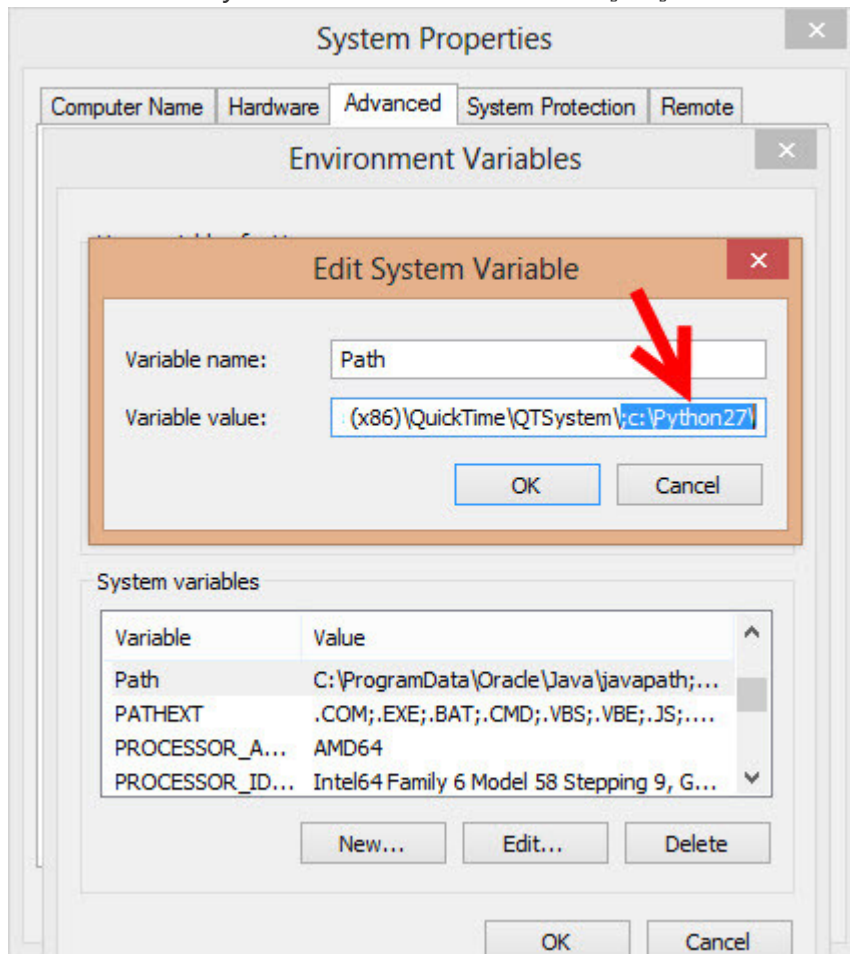
7. เพิ่มข้อความ C:Python27\ ต่อท้ายค่าเดิมที่มีอยู่ดังรูป



8. ในส่วนของ System variables ทำเช่นเดียวกัน



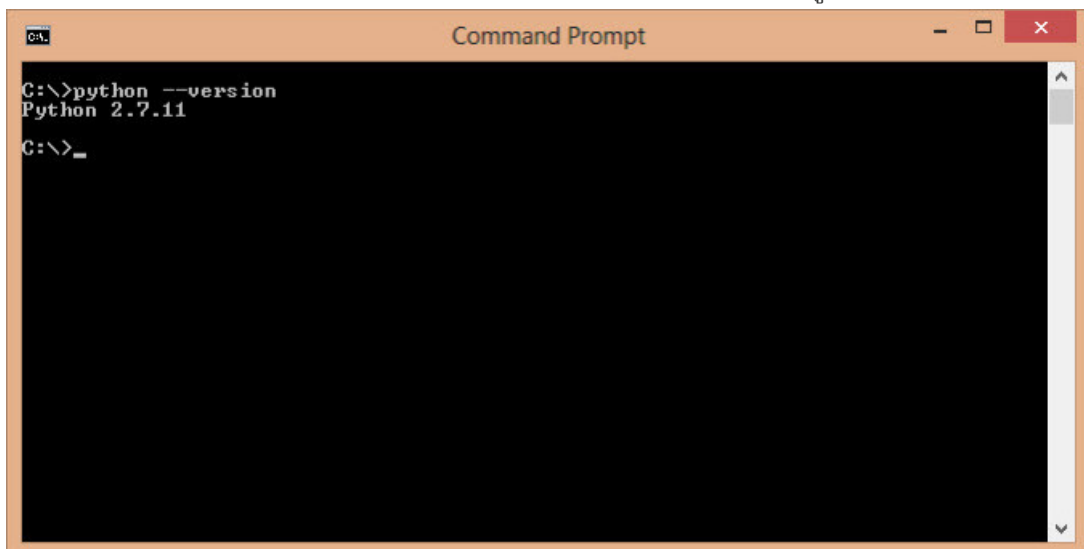
9. เพิ่มข้อความ C:\Python27\ ต่อท้ายค่าเดิมที่มีอยู่ดังรูป



10. เราลองทดสอบเช็คเวอร์ชันของ Python กันดูที่ Windows Command Prompt โดยไปที่ Run แล้ว พิมพ์ cmd หลังจากนั้นก็ลอง พิมพ์คำสั่ง

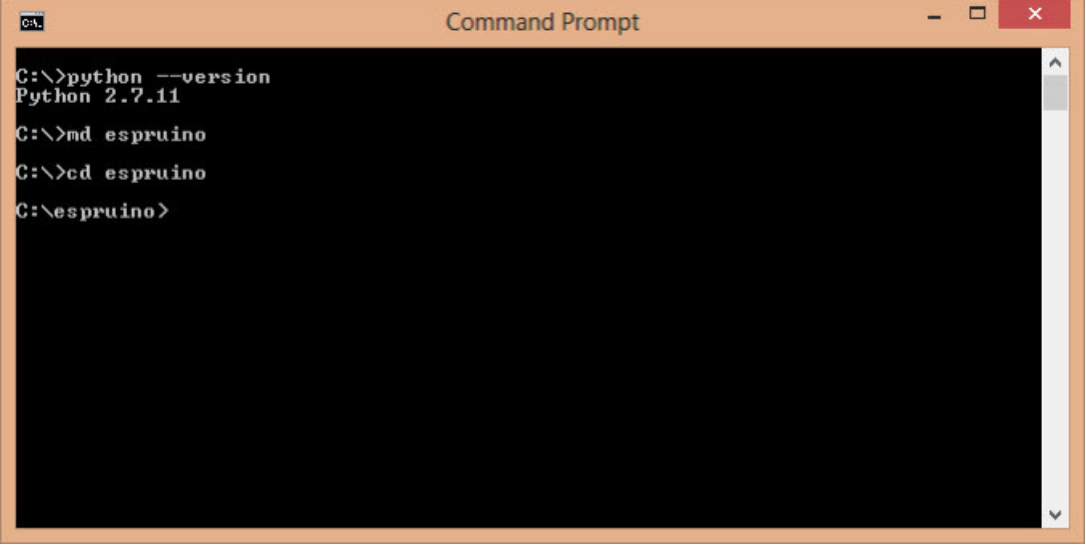
```
python --version
```

ถ้าการตั้งค่าเรียบร้อยดี ที่ Terminal จะโชว์ เลขเวอร์ชันขึ้นมามีดังรูป





11. ขั้นตอนต่อไปเราก็สร้าง folder ชื่อ espruino ไว้ที่ drive C: เพื่อเตรียมไฟล์ต่างๆที่จะดาวน์โหลดมาเก็บครับ

A screenshot of a Windows Command Prompt window. The title bar is orange and says "Command Prompt". The window has a black background with white text. The text shows the following commands and their outputs: 

```
C:\>python --version
Python 2.7.11

C:\>md espruino

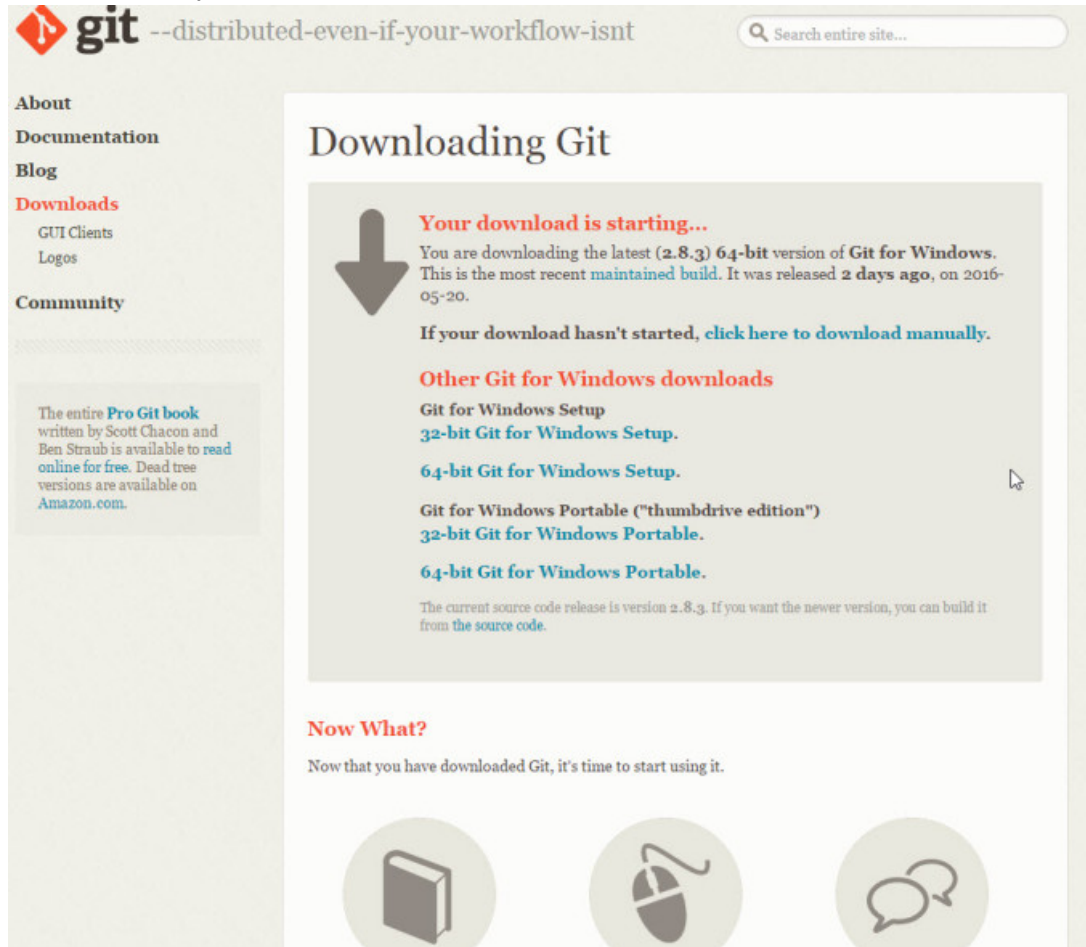
C:\>cd espruino

C:\espruino>
```

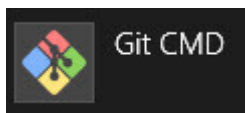
**Install ESP tool**

1. หลังจากที่เราติดตั้ง Python กันไปแล้ว

ตอนนี้ให้เราดาวน์โหลด Git for Windows <https://git-scm.com/download> มาก่อน  
เพื่อใช้เรียก Espruino Tool มาไว้ในเครื่องคอมพิวเตอร์ของเราก่อน



2. หลังจากติดตั้ง Git แล้วให้เปิดตัว Console ของ Git ขึ้นมาเลยครับ

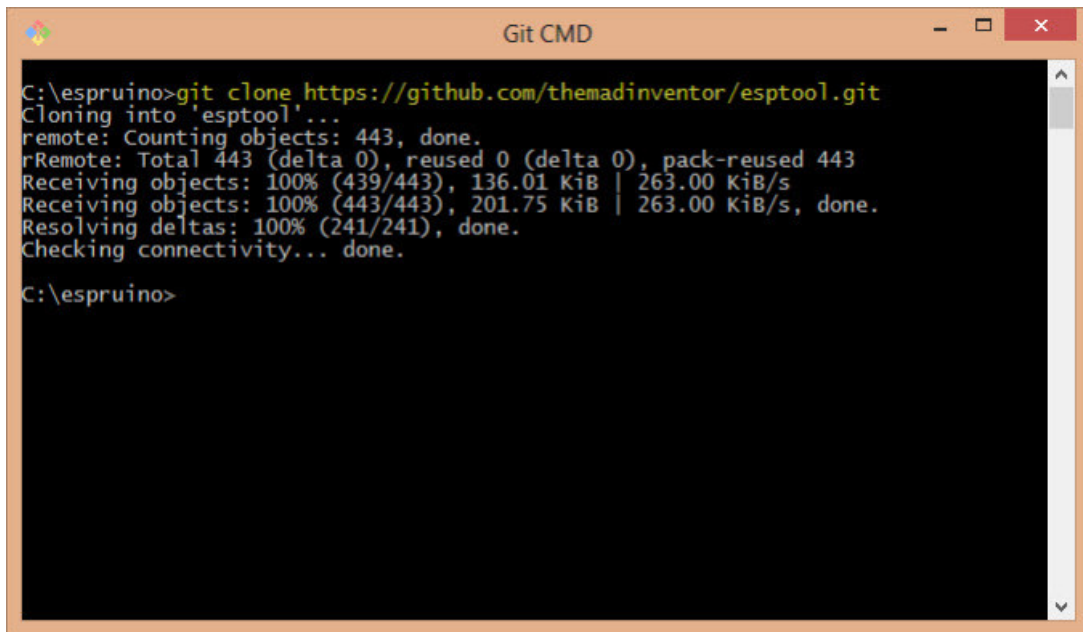


3. เข้าไปที่ folder ชื่อ espruino ที่เราเคยสร้างไว้ที่ drive C:

4. พิมพ์คำสั่งด้านล่างเพื่อ clone esptool มาไว้ใน folder

```
1git clone https://github.com/themadinventor/esptool.git
```

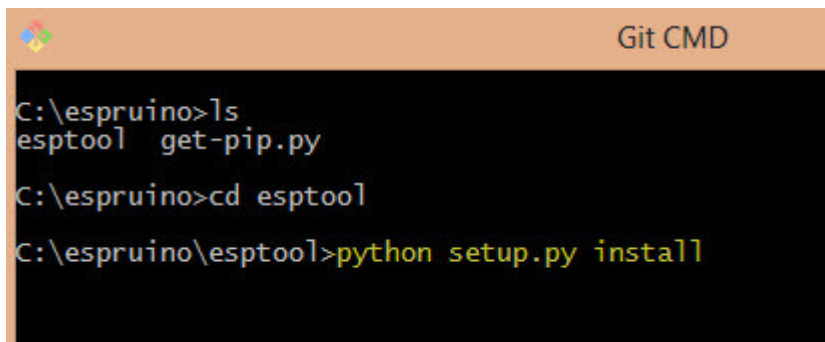
5. หลังจากที่เรา Clone มาเรียบร้อยแล้วจะเห็นข้อความดังภาพ



```
Git CMD
C:\espruino>git clone https://github.com/themadinventor/esptool.git
Cloning into 'esptool'...
remote: Counting objects: 443, done.
remote: Total 443 (delta 0), reused 0 (delta 0), pack-reused 443
Receiving objects: 100% (439/443), 136.01 KiB | 263.00 KiB/s
Receiving objects: 100% (443/443), 201.75 KiB | 263.00 KiB/s, done.
Resolving deltas: 100% (241/241), done.
Checking connectivity... done.
C:\espruino>
```

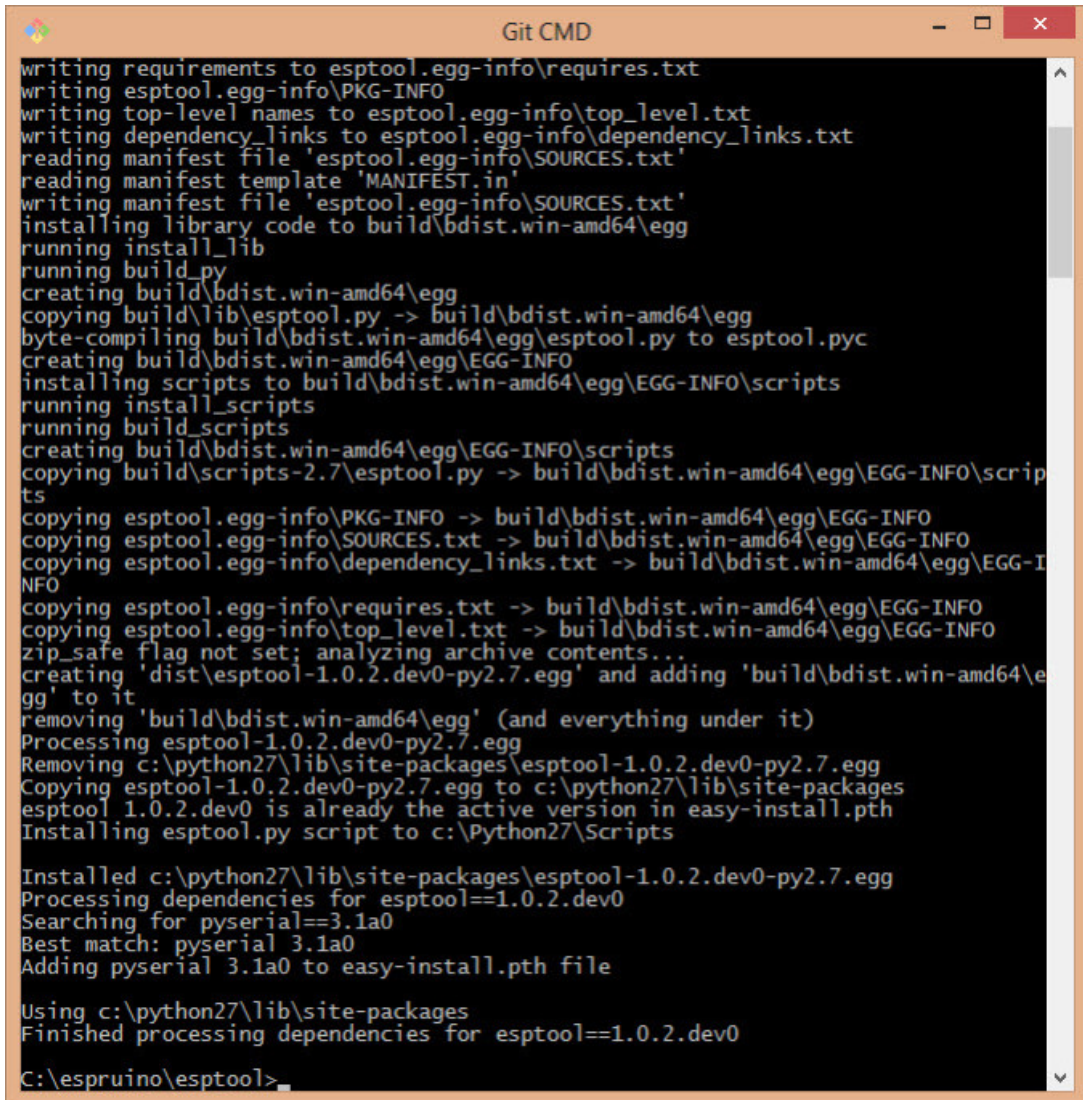
6. ต่อไปก็ Install esptool กันโดย ใน folder esptool ให้พิมพ์คำสั่ง

1python setup.py install



```
Git CMD
C:\espruino>ls
esptool  get-pip.py
C:\espruino>cd esptool
C:\espruino\esptool>python setup.py install
```

## 7. เมื่อเสร็จแล้วจะเป็นดังภาพด้านล่าง



```
writing requirements to esptool.egg-info\requires.txt
writing esptool.egg-info\PKG-INFO
writing top-level names to esptool.egg-info\top_level.txt
writing dependency links to esptool.egg-info\dependency_links.txt
reading manifest file 'esptool.egg-info\SOURCES.txt'
reading manifest template 'MANIFEST.in'
writing manifest file 'esptool.egg-info\SOURCES.txt'
installing library code to build\bdist.win-amd64\egg
running install_lib
running build_py
creating build\bdist.win-amd64\egg
copying build\lib\esptool.py -> build\bdist.win-amd64\egg
byte-compiling build\bdist.win-amd64\egg\esptool.py to esptool.pyc
creating build\bdist.win-amd64\egg\EGG-INFO
installing scripts to build\bdist.win-amd64\egg\EGG-INFO\scripts
running install_scripts
running build_scripts
creating build\bdist.win-amd64\egg\EGG-INFO\scripts
copying build\scripts-2.7\esptool.py -> build\bdist.win-amd64\egg\EGG-INFO\scripts
copying esptool.egg-info\PKG-INFO -> build\bdist.win-amd64\egg\EGG-INFO
copying esptool.egg-info\SOURCES.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying esptool.egg-info\dependency_links.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying esptool.egg-info\requires.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying esptool.egg-info\top_level.txt -> build\bdist.win-amd64\egg\EGG-INFO
zip_safe flag not set; analyzing archive contents...
creating 'dist\esptool-1.0.2.dev0-py2.7.egg' and adding 'build\bdist.win-amd64\egg' to it
removing 'build\bdist.win-amd64\egg' (and everything under it)
Processing esptool-1.0.2.dev0-py2.7.egg
Removing c:\python27\lib\site-packages\esptool-1.0.2.dev0-py2.7.egg
Copying esptool-1.0.2.dev0-py2.7.egg to c:\python27\lib\site-packages
esptool 1.0.2.dev0 is already the active version in easy-install.pth
Installing esptool.py script to c:\Python27\Scripts

Installed c:\python27\lib\site-packages\esptool-1.0.2.dev0-py2.7.egg
Processing dependencies for esptool==1.0.2.dev0
Searching for pyserial==3.1a0
Best match: pyserial 3.1a0
Adding pyserial 3.1a0 to easy-install.pth file

Using c:\python27\lib\site-packages
Finished processing dependencies for esptool==1.0.2.dev0

C:\espruino\esptool>
```

## 8. เย้เย้..เสร็จแล้วสำหรับเครื่องมือที่เตรียมใช้ในการ Flash ESP8266

**ดาวน์โหลดไบনারีไฟล์ Espruino for ESP8266**

## 1. ไปที่ฟอรัมของ Espruino ตามลิงค์นี้

<http://forum.espruino.com/conversations/279176/> เพื่อดาวน์โหลดไฟล์ล่าสุดของ Espruino for ESP8266



tve a month ago - #59

It looks like it's time for a new *experimental* build! This build moves a ton of symbol table stuff to flash, freeing up gobs of RAM. The result is 1600 jsvars (instead of 1400) and >10KB of free heap ensuring you don't run out of memory if you run STA+AP and have multiple sockets open.

This build is experimental for those of you that want to play with the bleeding edge or to help test this. For a stable firmware please download an official build!

The build is available at: [s3.voneicken.com/espruino/espruino\\_1v85.tve\\_master\\_cee7141\\_esp8266.tgz](http://s3.voneicken.com/espruino/espruino_1v85.tve_master_cee7141_esp8266.tgz)

Changes:

- 1600 jsvars
- >10KB heap
- on exception prints stack dump, see ./targets/esp8266/printstack to print stack backtrace

creationix a month ago - #60

With this latest build, I can't get my chip to boot. It is stuck in a reboot loop with the message **Invalid saved code in flash!**

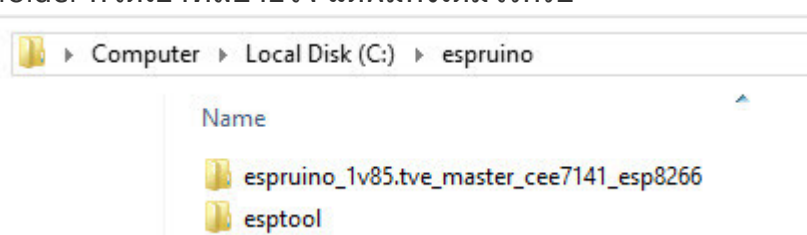
I'm using an Adafruit feather esp8266 (ESP-12mod) and using the esp-12 settings in the README.

virtualcodewarrior a month ago - #61

I can confirm that the same thing happens when I build it from source using the latest github version when running on an ESP8266-ESP12-F. In my case it loops with this output :

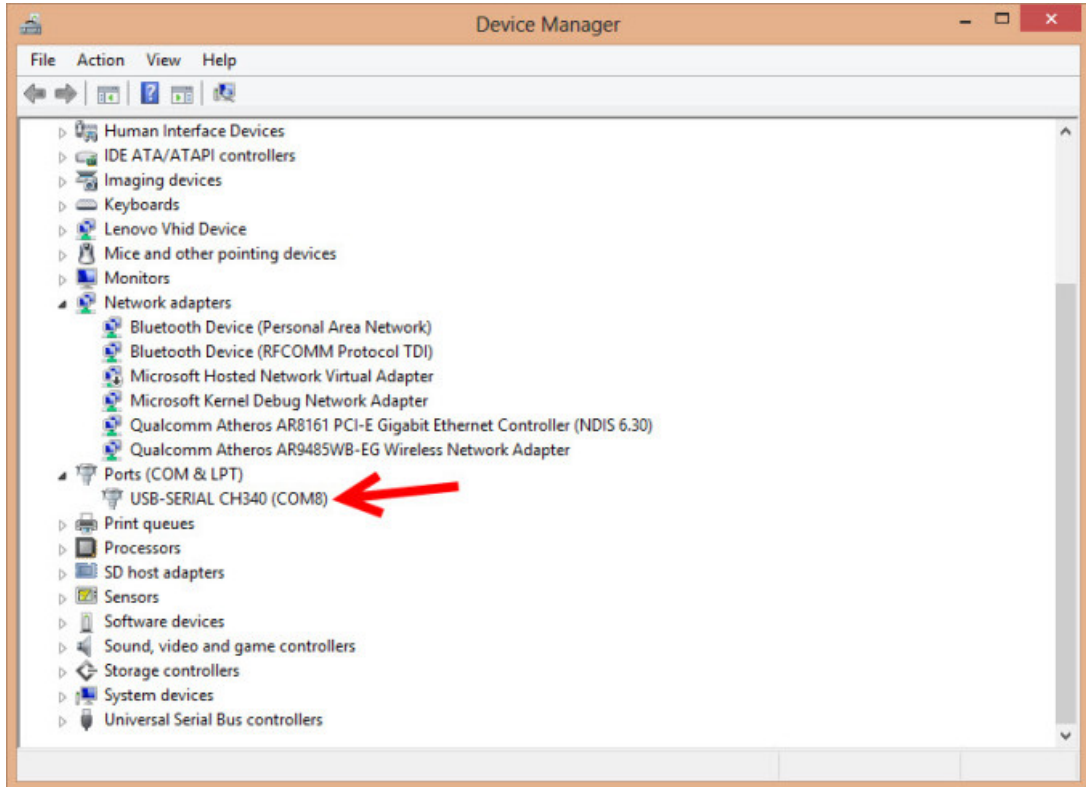
```
1. 2nd boot version : 1.4(b1)
2.  SPI Speed      : 80MHz
3.  SPI Mode       : QIO
4.  SPI Flash Size & Map: 32Mbit(512KB+512KB)
5.  jump to run user1 @ 1000
```

## 2. ทำการ Unzip และไว้ใน folder espruino ใน drive C: ที่เราสร้างไว้ จะเปลี่ยนชื่อ folder ก็ได้เอาที่สบายใจ แต่ผมคงเดิมไว้ครับ



**เชื่อมต่อ ESP8266 เพื่อทำการ Flashing (Flashing ESP8266)**

1. ตอนนี้ก็ต่อ ESP8266 กับคอมพิวเตอร์ของเราผ่านสาย USB กันเลยครับ เมื่อเราต่อแล้ว เราอาจสงสัยว่า Port อะไรที่เราเชื่อมต่ออยู่ ให้เราไปเช็คที่ Device Manager ดูนะครับ ในรูปของผมเป็น Port COM8 ครับ เลข Port สำคัญมากครับเพราะเราจะต้องไป กำหนดในการ Flash



2. ขั้นตอนต่อไปเราจะทำการล้างไฟล์ใน erase ESP8266 ให้สิ้นซากก่อนที่เราจะ Flash เจ้า espruino ลงไป...
3. กลับมาที่ Git CMD แล้วใช้คำสั่งด้านล่างเพื่อทำการลบไฟล์เก่าที่อยู่ใน ESP8266 (ใช้ คำสั่งใน folder esptool) ออ อย่างลืมเปลี่ยน COM port ของท่านนะครับของผม – port COM8

```
python ../esptool/esptool.py --port COM8 erase_flash
```

```
C:\espruino\esptool>python esptool.py --port COM8 erase_flash
esptool.py v1.0.2-dev
Connecting...
Erasing flash (this may take a while)...
C:\espruino\esptool>
```

4. หลังจากที่เราลบไฟล์เก่าแล้ว เราก็พร้อมจะ Flash espruino แล้ว
5. ตอนนี้ให้เราเข้าไปที่ folder espruino\_1v85.... ชื่อยาวว เพื่อจะทำการเรียกไฟล์ต่างๆ ใน folder นี้ flash ลงไปที่ ESP8266 ครับ ดังรูปด้านล่าง

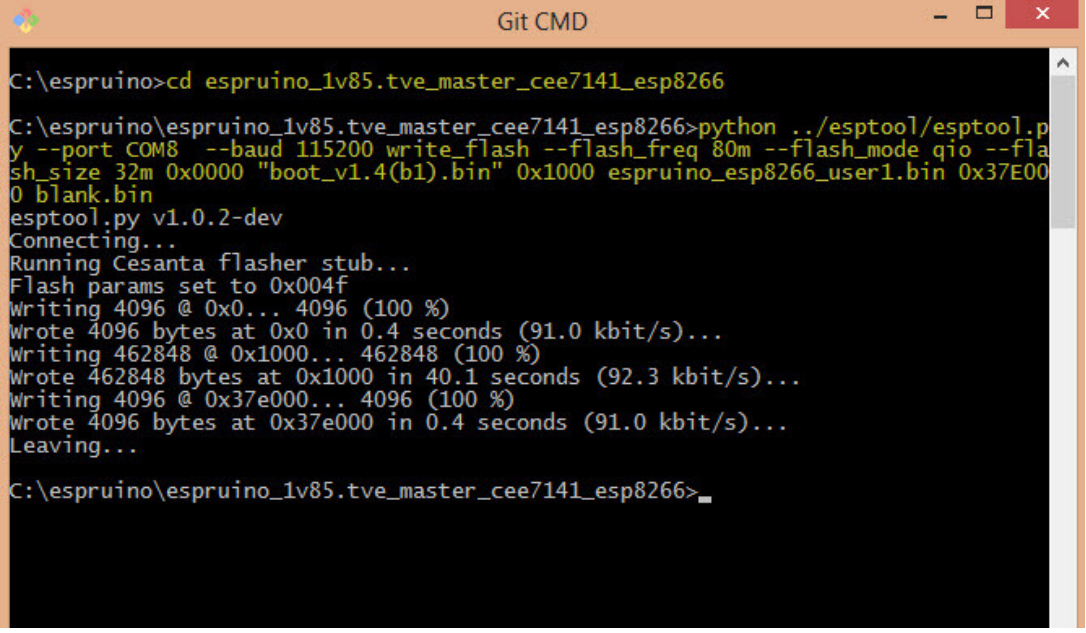


6. แล้วให้ใช้ Python เรียกรัน esptool.py ใน folder esptool โดยพิมพ์ตามด้วย พารามิเตอร์ต่างๆดังนี้ (ยาวนิดนึง)

```
python ../esptool/esptool.py --port COM8 --baud 115200  
write_flash --flash_freq 80m --flash_mode qio --  
1 flash_size 32m  
2 0x0000 "boot_v1.4(b1).bin" 0x1000  
   espruino_esp8266_user1.bin 0x37E000 blank.bin
```

จุดสังเกต...หลักๆจะเป็น COM port และ Baud rates ที่นี้เรากำหนด 115200

ถ้าถูกต้องจะได้ผลดังภาพด้านล่างครับ เย่เย่เย่.....



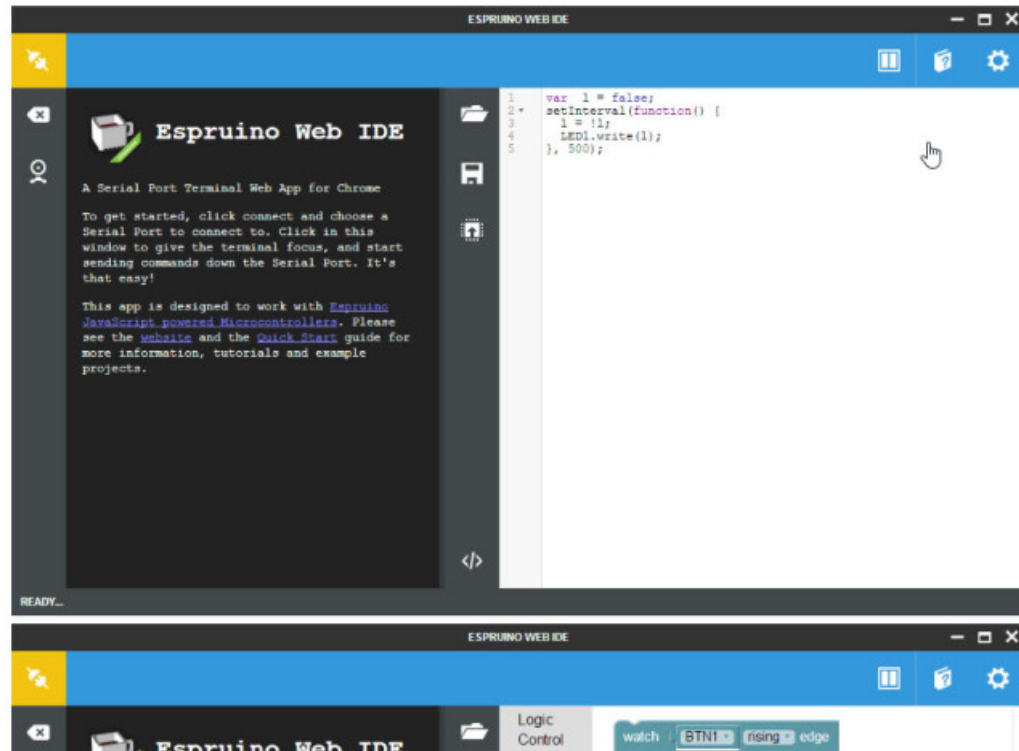
```
Git CMD  
C:\espruino>cd espruino_1v85.tve_master_cee7141_esp8266  
C:\espruino\espruino_1v85.tve_master_cee7141_esp8266>python ../esptool/esptool.py --port COM8 --baud 115200 write_flash --flash_freq 80m --flash_mode qio --flash_size 32m 0x0000 "boot_v1.4(b1).bin" 0x1000 espruino_esp8266_user1.bin 0x37E000 blank.bin  
esptool.py v1.0.2-dev  
Connecting...  
Running Cesanta flasher stub...  
Flash params set to 0x004f  
Writing 4096 @ 0x0... 4096 (100 %)  
Wrote 4096 bytes at 0x0 in 0.4 seconds (91.0 kbit/s)...  
Writing 462848 @ 0x1000... 462848 (100 %)  
Wrote 462848 bytes at 0x1000 in 40.1 seconds (92.3 kbit/s)...  
Writing 4096 @ 0x37e000... 4096 (100 %)  
Wrote 4096 bytes at 0x37e000 in 0.4 seconds (91.0 kbit/s)...  
Leaving...  
C:\espruino\espruino_1v85.tve_master_cee7141_esp8266>
```

โปรแกรมด้วยภาษาจาวา JavaScript บน ESP8266 (Running JavaScript on ESP8266)

1. เราสามารถใช้ โปรแกรม Terminal อะไรก็ได้ติดต่อไปยัง ESP8266 เช่น [PuTTY](#) เป็นต้น แต่ทางผู้พัฒนา Espruino ได้จัดทำ [Espruino Web IDE](#) ขึ้นมาซึ่งมีหน้าตาหล่อเหลาเอากการ ผมจึงแนะนำให้ใช้ตัวนี้เลยครับ

## Espruino Web IDE gitter join chat

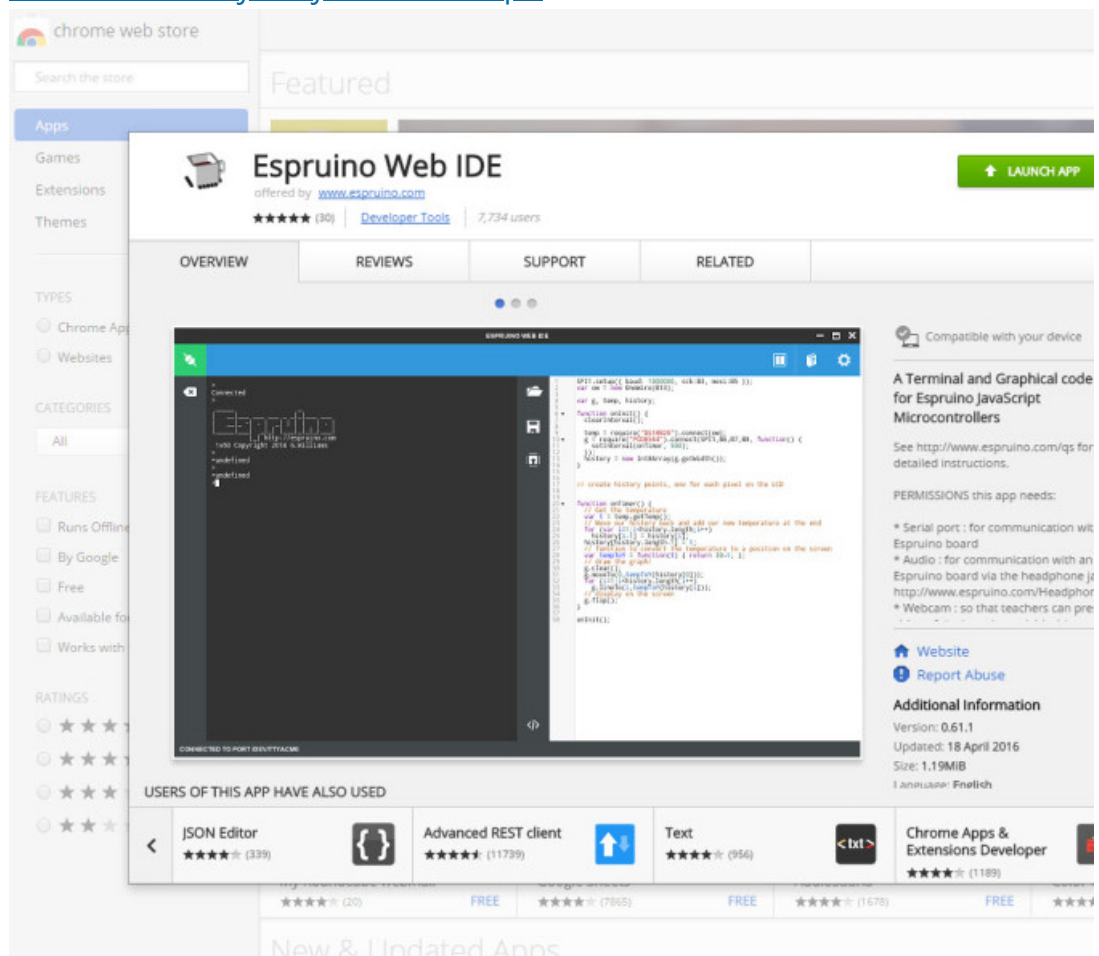
A VT100 Serial Terminal as a Chrome Web App (with both syntax highlighted and graphical editors) - designed for writing code on microcontrollers that use the [Espruino JavaScript interpreter](#). It can also run natively via [Node.js](#) and [Electron](#), or a version with limited IO can be used [as a website](#).



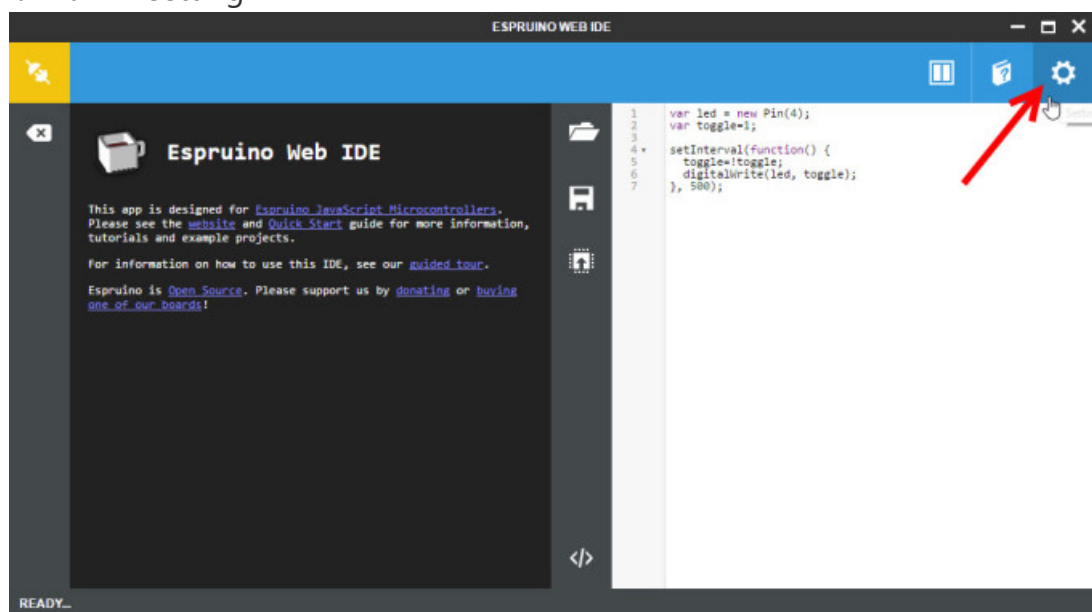


2. สามารถติดตั้งเป็น extension ให้กับ Chrome ได้

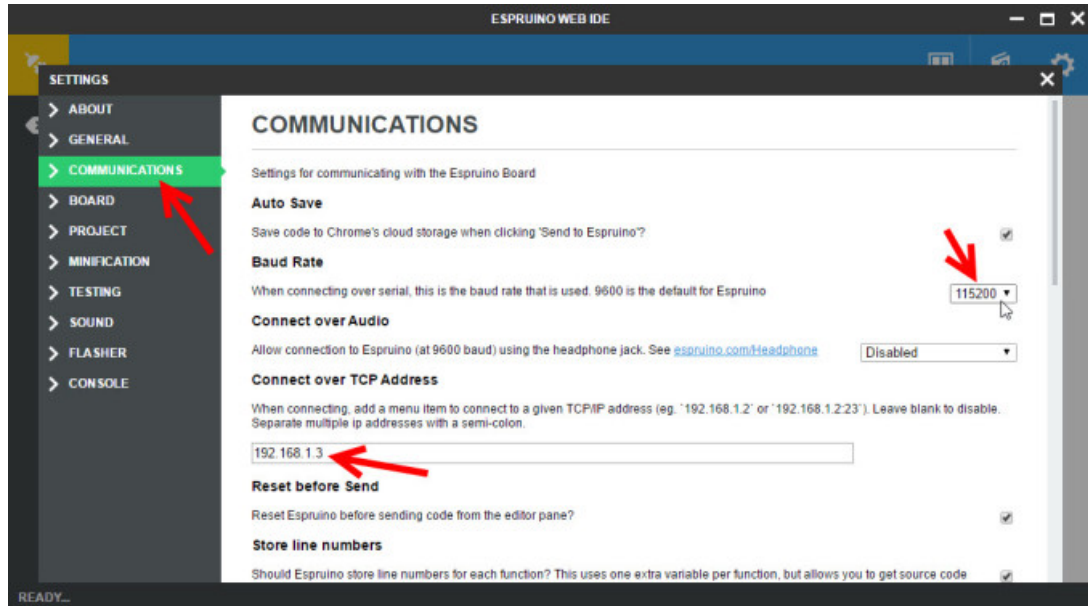
เลย <https://chrome.google.com/webstore/detail/espruino-web-ide/bleoifhkdaibjfbobjackfdifdneehpo>



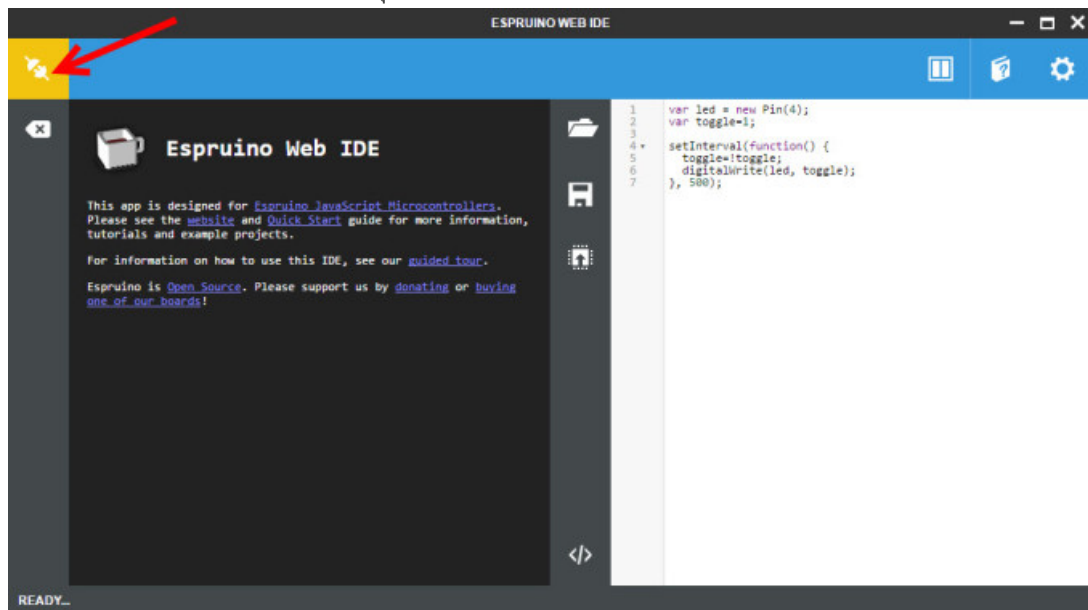
3. หลังจากทำการติดตั้งแล้วเราก็เปิดโปรแกรมขึ้นมาใช้งานกันเลยครับ แต่ก่อนอื่นให้เราไปตั้งค่าที่ setting



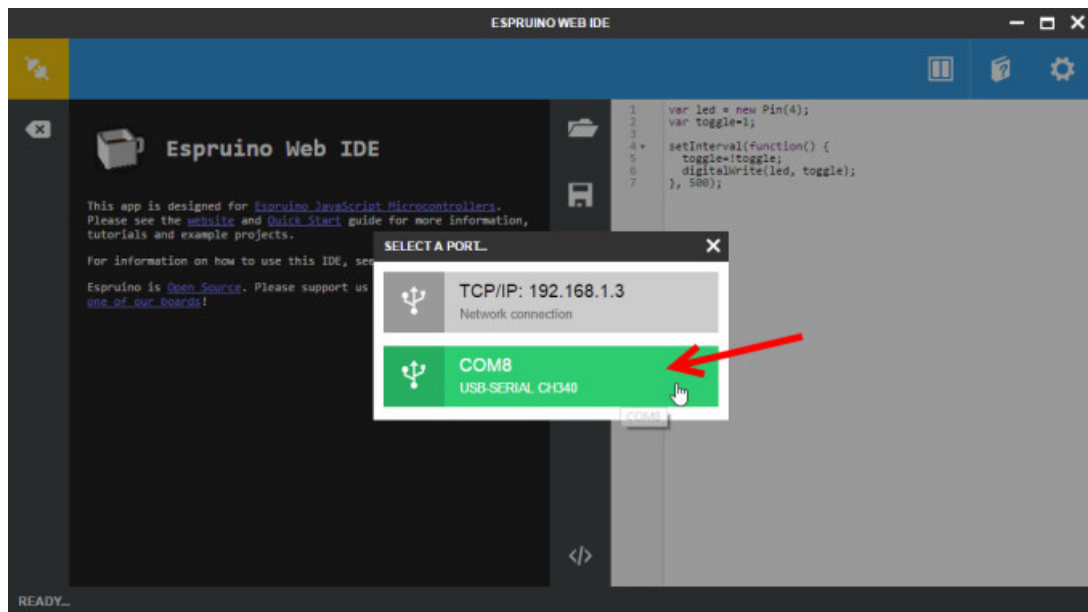
4. กำหนด Baud Rates ให้ตรงกับที่เราได้ทำการ Flash มาซึ่งเป็นค่า 115200 และในส่วนของ Connect over TCP Address เราสามารถใส่ค่า IP ของ ESP8266 ที่เราใช้ flash ได้ ซึ่งจะมีประโยชน์ในการอัปโหลดโปรแกรมที่เขียนได้ผ่าน WiFi โดยไม่ต้องต่อสาย USB (วิธีการเช็ค IP จากตัว ESP8266 จะอธิบายให้ทราบอีกทีครับ)



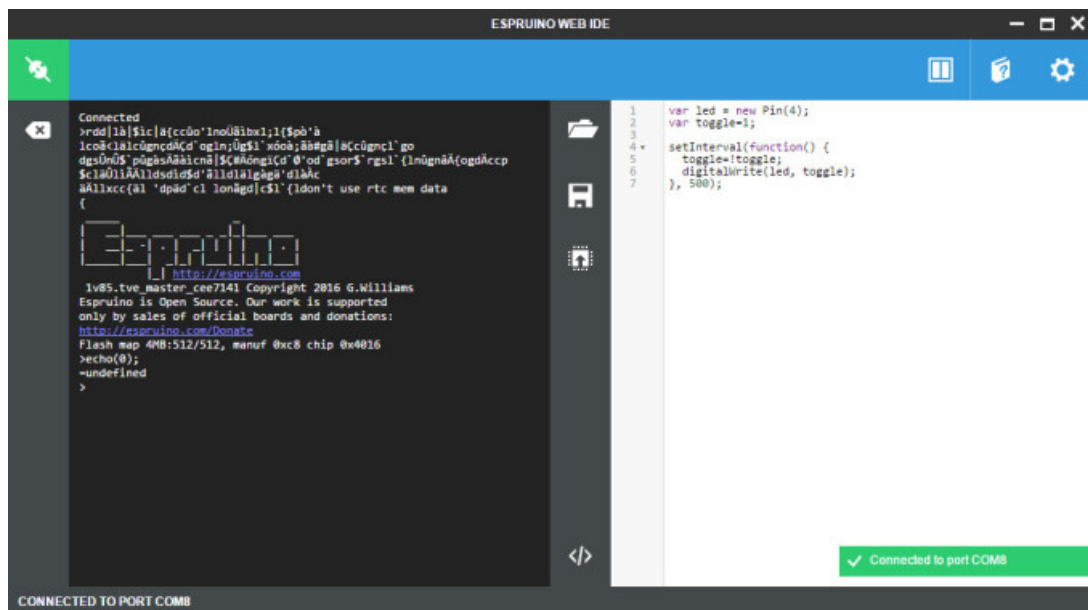
5. ทดสอบเชื่อมต่อเลยครับ กดปุ่มเชื่อมต่อดังภาพด้านล่างกันเลย



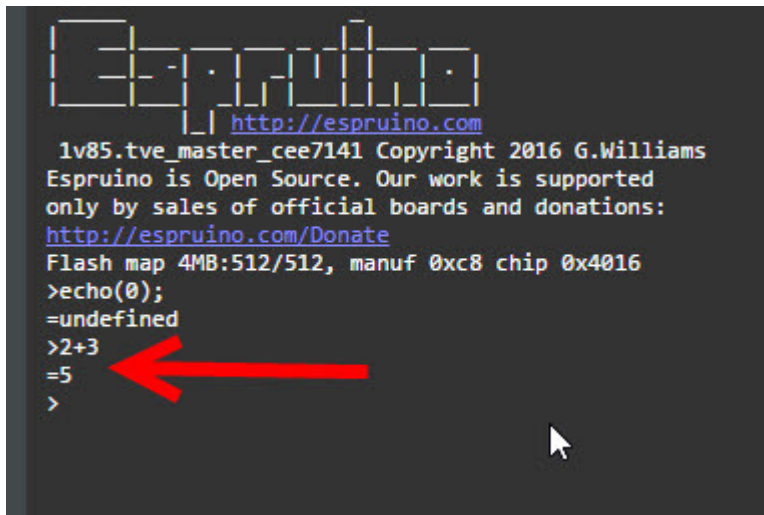
6. เลือกวิธีการเชื่อมต่อ ในที่นี้เพื่อนๆอาจมีแค่ต่อด้วยCOM port แต่ถ้าเราเพิ่มผ่านWiFiด้วยแล้วจะขึ้นมาดังภาพครับ ผมต่อด้วยCOM8ก่อนนะครับ



7. เชื่อมต่อได้แล้วจะมีข้อความและกราฟิกขึ้นมาดังภาพด้านล่างนะครับ เย้เย้เย้....



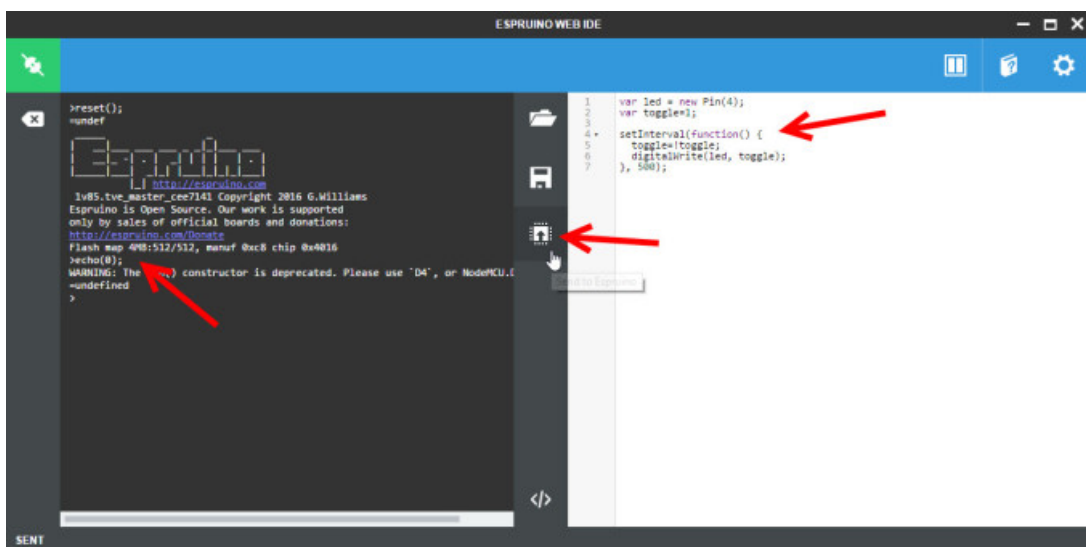
8. ลองพิมพ์บวกเลขง่ายๆในหน้าต่างTerminalดูว่าเวิร์คไหม ผลลัพธ์ออกมาถูกต้องเป็นอันใช้ได้ครับ



```
Espruino
http://espruino.com
1v85.tve_master_cee7141 Copyright 2016 G.Williams
Espruino is Open Source. Our work is supported
only by sales of official boards and donations:
http://espruino.com/Donate
Flash map 4MB:512/512, manuf 0xc8 chip 0x4016
>echo(0);
=undefined
>2+3
=5
>
```

9. ลองมาเขียนโปรแกรมไฟกระพริบตามสูตร (โปรแกรมพื้นฐานเหล่านี้ออกมา555) กันบ้างด้วยภาษา JavaScript ในหน้าต่างด้านขวามือ เขียนเสร็จก็กดปุ่ม Send to Espruino กันเลยครับ ออกเกือบลืมนต่อ LED ไว้ที่ขา GPIO4 ดังนั้นในโปรแกรมแรกก็ใช้ Pin(4)นะครับ

```
1var led = new Pin(4);
2var toggle=1;
3
4setInterval(function() {
5 toggle=!toggle;
6 digitalWrite(led, toggle);
7}, 500);
```

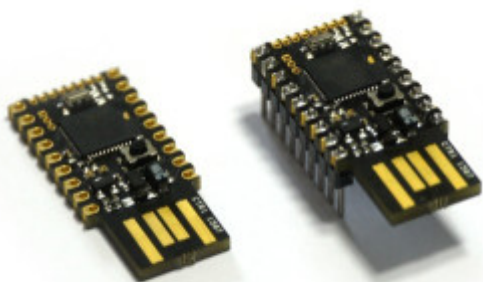




## Running Javascript on ESP8266 (NodeMCU) ใช้งาน จาวาสคริปต์บน ESP8266

สวัสดีครับวันนี้เราลองมาใช้งานจาวาสคริปต์ Javascript บน ESP8266 กันดูครับ ในบทความนี้ผมใช้ NodeMCU v0.9 ในการทดลองครับ

เพื่อนๆคงเคยใช้ ESP8266 กับภาษา Lua หรือ C/C++ กับ Arduino IDE กันเป็นส่วนใหญ่ เพราะค่อนข้างเข้าใจง่าย มีเครื่องมือและบทความต่างๆมากมายให้ศึกษากัน แต่ถ้าบอกว่าเรา มารัน JavaScript บนESP8266กันดูก็คงบอกว่าทำได้หรือไม่ มีความจำเป็นมากนักแค่ไหน ไหมจริงๆแล้วต้องขึ้นอยู่กับนักพัฒนาว่าถนัดในภาษาใด ความยืดหยุ่น ความเร็วในการทำงาน ซึ่ง อาจจะยากง่ายแตกต่างกันไปครับ โดยส่วนตัวผมว่านี่ก็เป็นทางเลือกหนึ่งครับ เพราะภาษา JavaScript ก็มีความนิยมและใช้แพร่หลายกันมากเป็นคอร์เทคโนโลยีหลักในการพัฒนา เว็บไซต์ อีกทั้งอาจจะง่ายสำหรับผู้เริ่มต้นศึกษาการเขียนโปรแกรมมือใหม่ แหล่งเรียนรู้มากมาย เช่นที่ Code Academy <https://www.codecademy.com/learn/javascript>



JavaScript สำหรับบอร์ด Microcontroller นั้นโดยเริ่มแรกมีการถูกพัฒนาเพื่อใช้งานบน บอร์ด Espruino (<http://www.espruino.com/Get+it>) และ Tessel (<https://tessel.io/>) แต่ก็ยังไม่ได้รับความนิยมอย่างกว้างขวางเหมือน Arduino หรือ ESP8266 เหล่าสาวก Espruino มีการเรียกร่องผู้พัฒนา Espruino ([Gordon Williams](#)) ว่าอยากจะใช้งาน JavaScript บน ESP8266 ... ขอมาฝากพี่ท่านก็เลยจัดให้ (ใจกว้างมากตั้งแม่น้ำเทมส์ ณ.กรุง ลอนดอน)

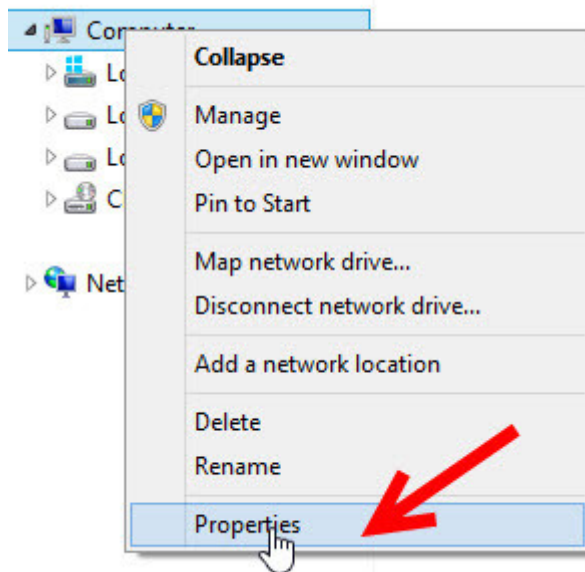
เรามาเริ่มเตรียมเครื่องมือกันเลยครับ (Installing tools)

1. Python 2.7.xx <https://www.python.org/downloads/> ในที่นี้ผมใช้ Python 2.7.11 สำหรับ Python 3.5.xx ไม่สามารถใช้ได้นะครับ (ผมลองมาแล้วฮา..)
2. Git for Windows <https://git-scm.com/download> บทความนี้เรา Flash จาก Windows 8
3. Espruino build for Esp8266 <http://forum.espruino.com/conversations/279176/>
4. [Espruino Web IDE](#) โปรแกรมที่ใช้เขียนโปรแกรมลงบนบอร์ดเรา ในที่นี้เราจะใช้ผ่าน Chrome เว็บเบราว์เซอร์
5. ESP8266 board ผมใช้ [NodeMCU](#) v.0.9 แต่เท่าที่ดู EPS8266-01 ก็ทำได้ครับ

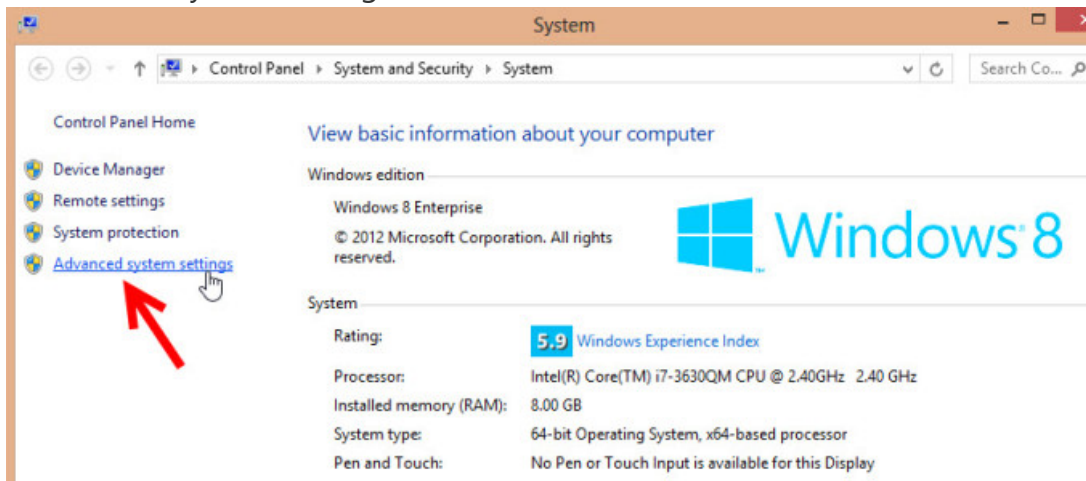
### ขั้นตอน (Flashing process)

#### Install Python

1. ดาวน์โหลด Python 2.7.xx <https://www.python.org/downloads/>
2. ตรวจสอบให้แน่ใจว่าเราจะใช้คำสั่ง Python ได้จาก Windows command line (CMD) โดยให้ไปกำหนดการเข้าถึงโปรแกรม Python ที่ Windows Environment
3. ไปที่ Computer > Properties



#### 4. >Advanced system setting



#### 5. >Environment Variables...

