Rodrigo Sanchez

rodrigo@accelerando.com.au

Develop Microcontroller Based Applications using MicroPython using Visual Code Studio in ESP8266 modules

Last updated on 2017-11-20  
Rev by: Christopher Biggs

****

HOW TO START WITH MICROPYTHON

ESP8266 MCU MODULES

Guide Contents

1. **Guide Contents1**
2. **What is MicroPython?**2
3. **Requirements** 2
4. **Flashing MCU3**
5. **Code Editor with REPL Mode**7
6. **References** 13
7. **What is MicroPython?**

***MicroPython*** is a lean, efficient and open source implementation of the [***Python 3***](http://www.python.org/) programming language that includes a small subset of the Python standard library and is optimized that runs on embedded development boards with microcontrollers and in constrained environments. With this interpreter you can write clean and simple Python code to control hardware instead of using complex low-level languages such as: C or C++. Using features like ***REPL*** [Read Evaluate Print Loop] executes code connected directly to your board without the necessity of compiling and uploading to the microcontroller [MCU] with makes it easier for prototyping.

1. **– Requirements**

* Development Board with ESP8666 (e.g : AI-Thinker ESP8266MOB, Webmos D1mini), make sure your MCU fits the minimum requirements of MicroPython (256 Kb Flash, 16 Kb RAM, 80 MHz CPU Clock)
* Computer with WIN10 OS, that comes with all the Serial drivers otherwise you need to install serial drivers for your board.
* Micro USB to USB Cable (Power and Comms)
* Leds, push buttons, resistors (if the board does not have inbuilt Reset/Programming)
* Flash programmer in this case NODEMCUand blank bin file

<https://nodemcu.readthedocs.io/en/master/en/flash/>

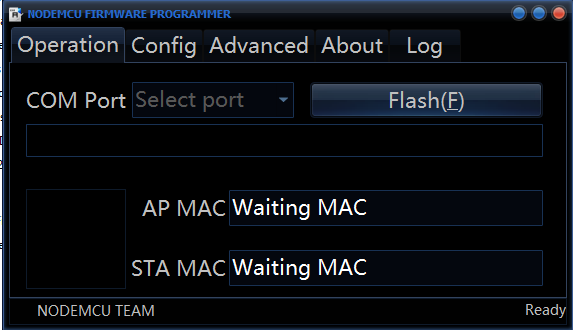
<http://www.pratikpanda.com/completely-format-erase-esp8266-flash-memory/>

* Latest MicroPython firmware for ESP8266 Boards, <https://micropython.org/download>.
* Putty (SSH/Telnet Client, <http://www.putty.org>)
* Code editor, Microsoft Visual Studio Code (open source, multiplatform, <https://code.visualstudio.com/Download> )
* Node.js (<https://nodejs.org>)

1. **- Flashing the MCU**

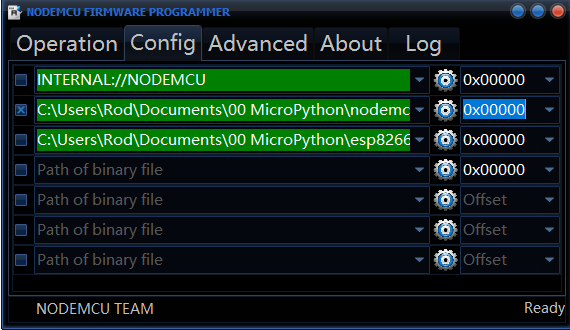
There are different Flash Programmers applications such as esptool.py in this case we are going to use NodeMCU with is an easy application windows based.

* - Install NodeMCU Firmware Programmer

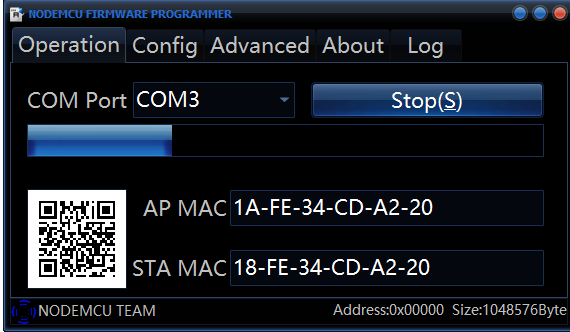


**Fig 3.1 Node MCU Flash Programmer**

* For the first time using MicroPython the MCU has to be completely erased all the flash.
* -Utilize the bin blank file bin file and start writing in address 0x0000 of the MCU, flash it, remember to setup the MCU in programing mode (Reset + GPO\_0 to GND), select your COM serial port.
* NodeMCU does not have an “erase flash” option, use a Blank 1MB flash(0x000000 to 0x1000000) blank bin file to load up.

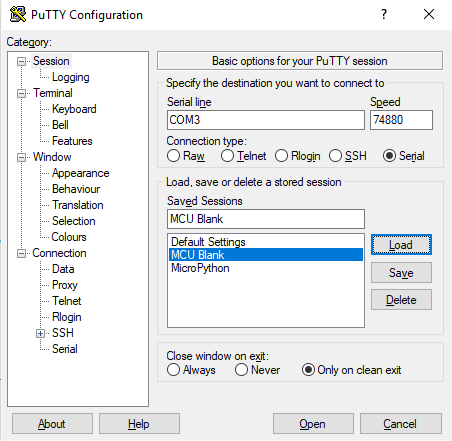


**Fig 3.2 Flash erasing using 1MB range blank bin file**

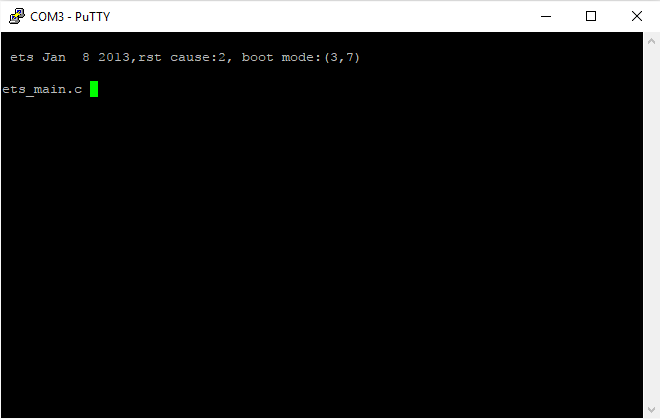


**Fig 3.3 MCU in programming mode, connected and programming**

* Using Putty set up to serial connection (74880,8,1,N,XON/XOFF) you can see that the MCU responds the following:

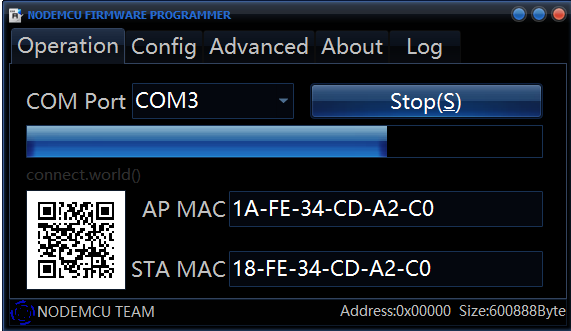


**Fig 3.4 Putty Terminal setup to serial connection**



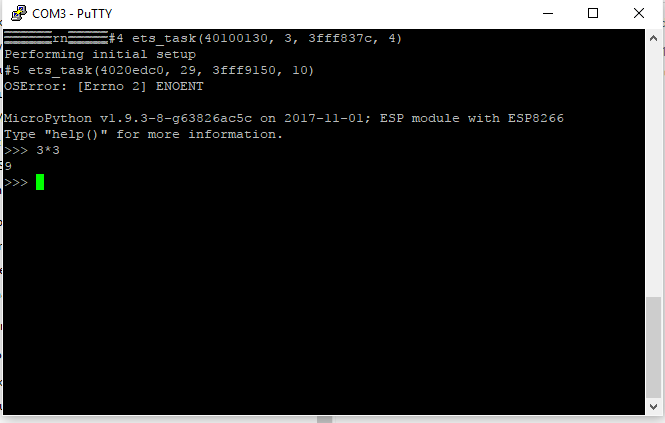
**Fig 3.5 MCU after reset in terminal**

* Use the latest MicroPython estable release and load it to the board staring at 0x000000



**Fig 3.6 Flashing MicroPython into the ESP8266**

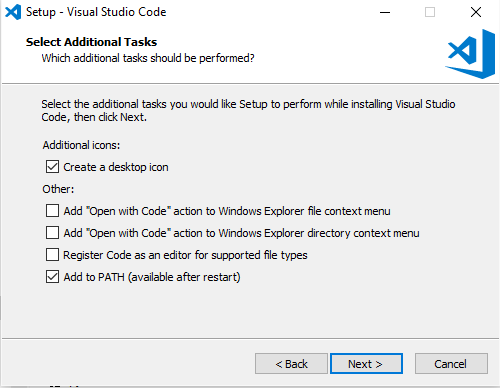
* Use Putty (115200,8,1,N,XON/XOFF) to see a terminal window with MicroPython booting up, remember to close NodeMCU to release the COM port. The terminal will be in REPL mode ready to program, make some quick Python test like a calculation.



**Fig 3.6 MicroPython ready to use in REPL mode**

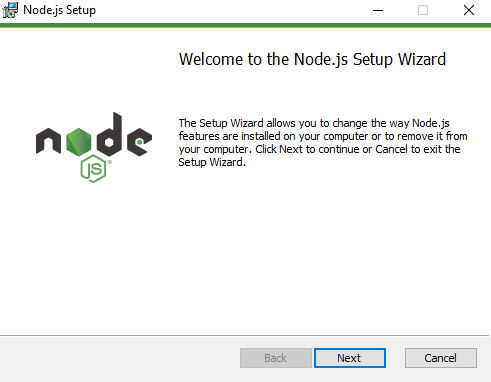
1. **Code Editor with REPL Mode**

* Any code editor can be used or use the REPL to start coding , in this case we are going to use Visual Code Editor because is an open source editor, multiplatform and will give us a plug in for the terminal in REPL
* Install Visual Studio Code [VSC], make sure that during the installation to choose the option to add to Path.



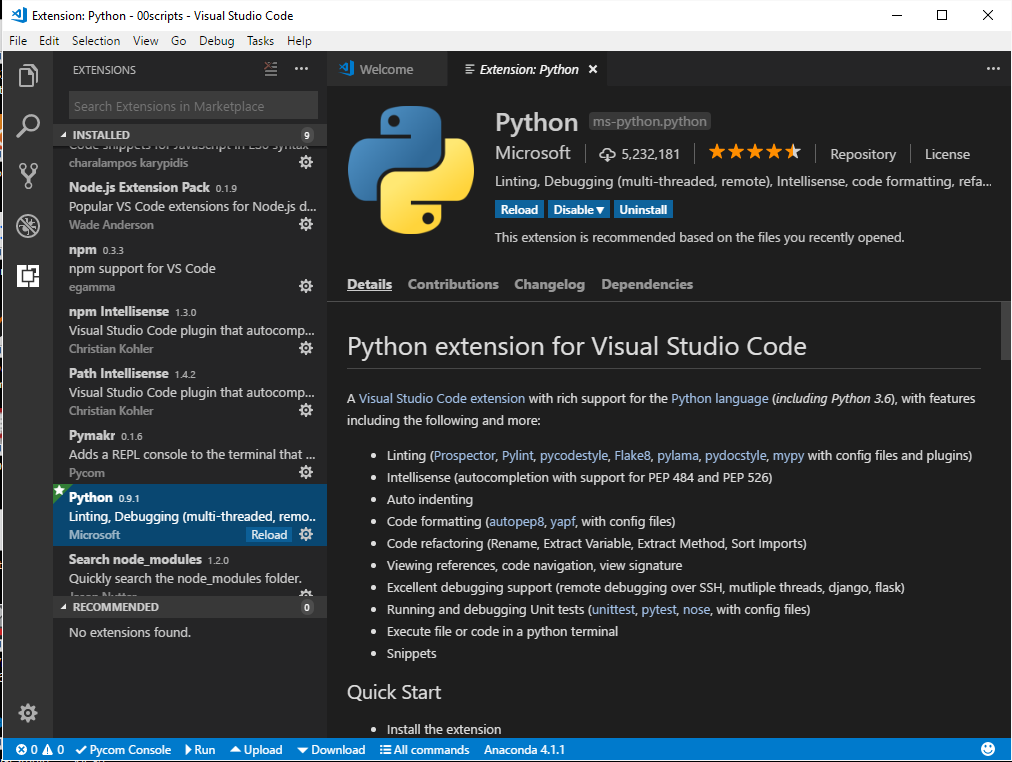
**Fig 4.1 Installation of Visual Studio Code**

* Install NodeJS, this is a requirement for use REPL mode in VSC



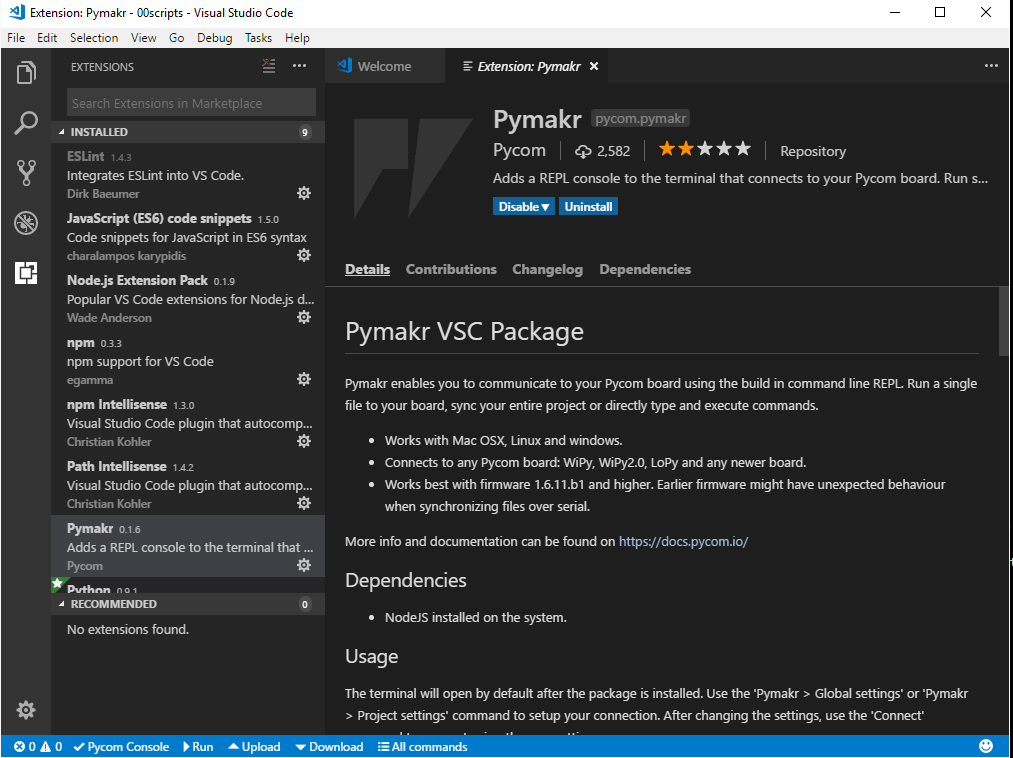
**Fig 4.2 Installation of Visual Studio Code**

* In VSC look for the Extension button then look for the Python extension, install and reload.

****

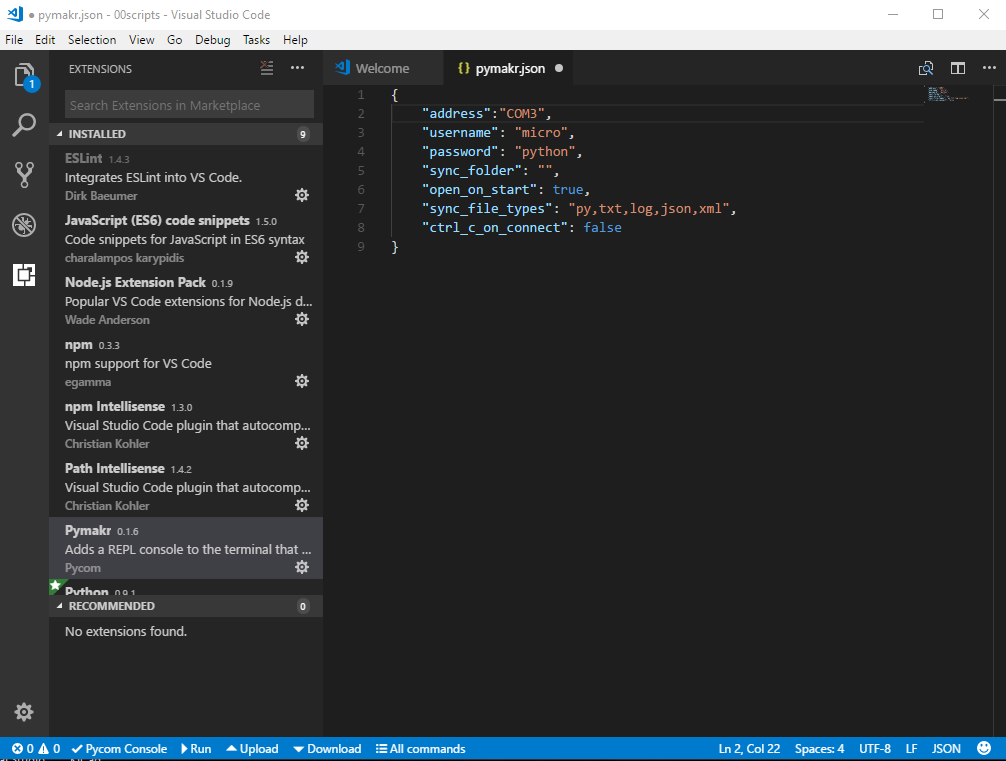
**Fig 4.3 Visual Studio Code with Python extension**

* Similar procedure for Pymakr extension that will give us the REPL integrated in VSC.



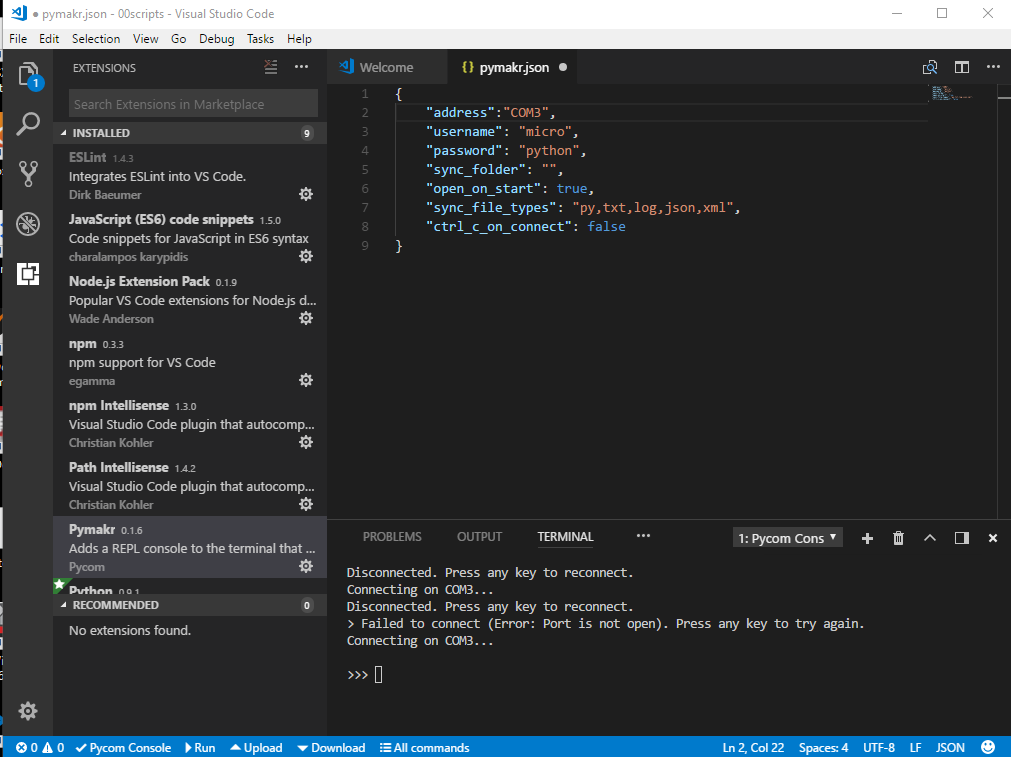
**Fig 4.4 Visual Studio Code with Pymkr extension**

* Setup the Python Terminal for REPL Mode, in the bottom Blue Menu bar go to:
  + All commands\Global Settings, to open the pymakr.json
  + All Commands\List serial ports that will put in the clipboard your COM Port
  + Return to the pymakr.json file and update the field "address":"192.168.4.1" with your COM Port (Paste/Ctrl+V)



**Fig 4.5 pymakr.json configuration file, updated for serial port connection.**

* Finally, close other Terminal connections and open the Integrated Terminal in VSC
  + - Top Menu Bar\View\Integrated Terminal
    - Bottom Blue Menu Bar\All Commands\Connect



**Fig 4.6 VCS with Integrated Terminal connected to the ESP8266 Board in REPL mode**

Ready to start prototyping in MicroPython, Happy coding…..! ☺

1. **References:**

<http://micropython.org/>

<https://learn.adafruit.com/micropython-basics-what-is-micropython/overview>

<http://www.agcross.com/2015/09/the-esp8266-wifi-chip-part-3-flashing-custom-firmware>

<https://nodemcu.readthedocs.io/en/master/en/flash/>

<http://www.pratikpanda.com/completely-format-erase-esp8266-flash-memory/>

<https://micropython.org>

<http://www.putty.org>)

<https://code.visualstudio.com>

<https://nodejs.org>