Beehive weight monitoring

3 year project

teacher:

Paulo Torres

Student:

Fábio Gonçalves

**Summary**

[1 Start steps on ESP8266 configuration 3](#_Toc21278580)

[1.1 Download and installation of the ARDUINO IDE 3](#_Toc21278581)

[Download Arduino IDE 3](#_Toc21278582)

[Installation of arduino IDE 3](#_Toc21278583)

[1.2 Download and installation of the python 3 4](#_Toc21278584)

[Download Python 4](#_Toc21278585)

[Installation of Python 4](#_Toc21278586)

[1.3 Download and installation of the Drivers for ESP8266 5](#_Toc21278587)

[1.4 Configure IDE to allow programming Node MCU Board 7](#_Toc21278588)

[1.5 Install and configure Raspberry Pi as mqtt broker 10](#_Toc21278589)



# Start steps on ESP8266 configuration

To start programming the ESP8266, it’s essential to do these simple steps:

1. Download and installation of the ARDUINO IDE
2. Download and install python
3. Download and install drivers of ESP8266
4. Configure IDE to allow programming the Node MCU Board
5. Download and install the required Libraries of Node MCU Board
6. Programing Node MCU Board

## Download and installation of the ARDUINO IDE

To download and install the Arduino IDE it’s required access to internet or installation package of software.

Download Arduino IDE

Open web browser and go to this link: <https://www.arduino.cc/en/Main/Software> and download the version adapted to operational system (exist versions for Windows, Linux and Mac Os X).

Install the program using the package downloaded in Arduino.cc Web Site.

Installation of arduino IDE

After the package download, start the installation, running the installation file on the system.

It will open an windows with installation options, select all of them.

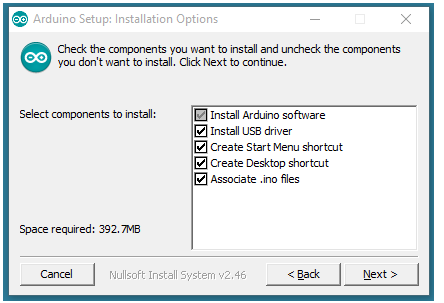


Fig. 1 - Installation Options

After this step, select the file location on destination folder by click on button “Browse …”, or click on install.

The Installation will proceed, and the installer will extract all necessary files to execute the “Arduino Software (IDE)”.

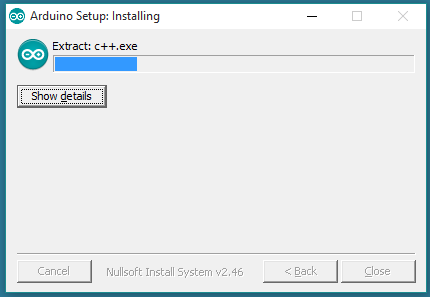


Fig. 2 - Arduino IDE installing

The installation will finish and it’s now possible to start Arduino IDE clicking on shortcut on Desktop or search in windows start menu by “Arduino”.

## Download and installation of the python 3

Download Python

In this part of the project, Python 3 allow to connect the librarys posteriorly downloaded in the next steps to Arduino IDE.

So to download the Python it’s essential to access <https://www.python.org/downloads/> and select the version. This version must be higher than version 3 of python and and adapted to operational system.



Fig. 3 - Downloading steps of Python 3.7.1

After the file downloaded, run the executable file (only for Windows) and follow the next steps.

Installation of Python

The executable file will open an windows like Fig.4 when it’s needed to select to “Add Phython 3.7 to Path” and after that click on “Install Now”. Alternatively for expert users the possibility to a run a “Customized installation” is allow.

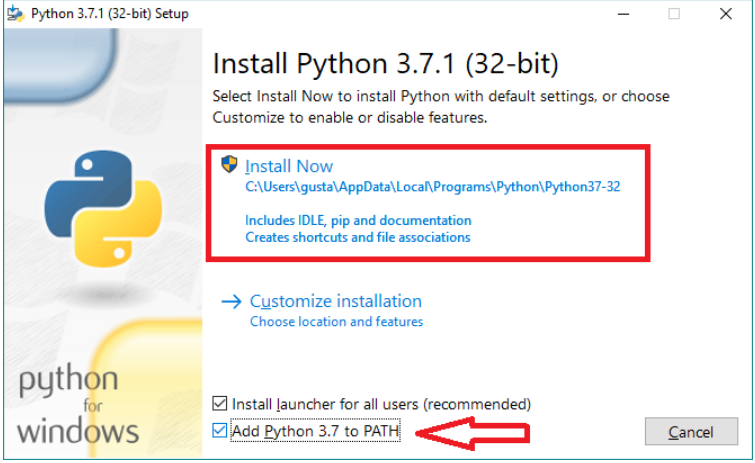


Fig. 4 - First steps to install Python

Wait a few moments, and when the installation was finished the python setup will show a message like that:

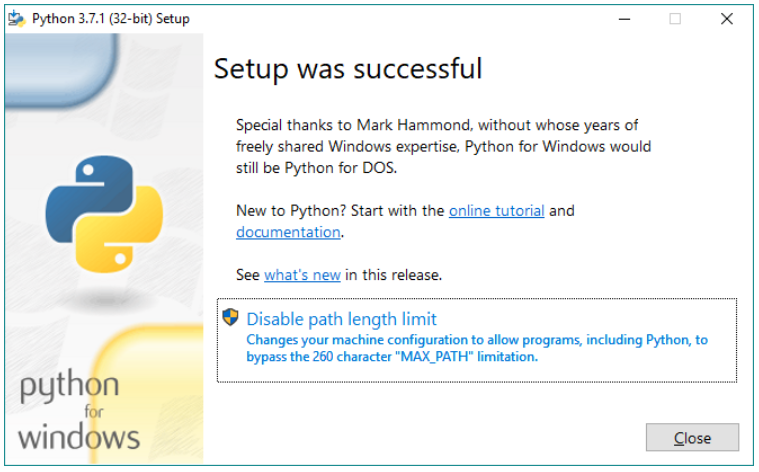


Fig. 5 - Finish message of Python installation

Now the python was successfully installed on machine, for check that, open command prompt by typing “cmd” on windows start menu and type again python. Command prompt will send a message and with the version installed and other specification of the python installation

## Download and installation of the Drivers for ESP8266

Probably in this step, Node MCU Board are not recognized for windows, so it’s necessary download the drivers using an executable file that could be downloaded here: <http://www.wch.cn/download/CH341SER_EXE.html> note that the site could be in a Chinese version.

For an “official” driver installation on Windows 10 it’s need to access the “Device manager” clicking with right button mouse on start menu and select “device manager”, an windows will be open and its possible to see a “trouble” in “other devices”, as you can see on Fig. 6.

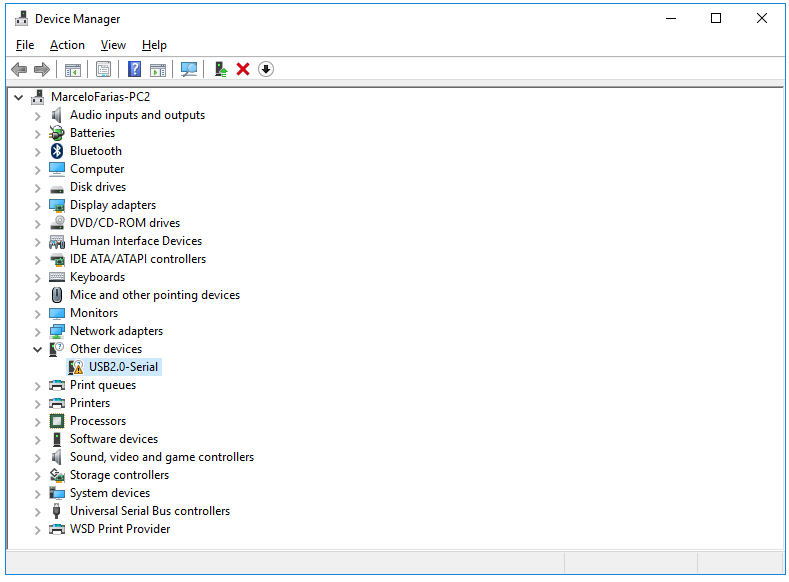


Fig. 6 - Windows Device Manager

By click with the right button on there, it’s possible to “install controllers” and windows will search a specific official driver. Alternatively, it’s possible to update controllers, and windows will seach by the last software drivers.

After these steps, the “Device Mager” will update and now it’s possible to see an update windows that contains ports corresponding to Node MCU Board registered as COM & LPT ports, in this example the connected port its COM 21, note that COM Port number is automatically administered by the operation system.

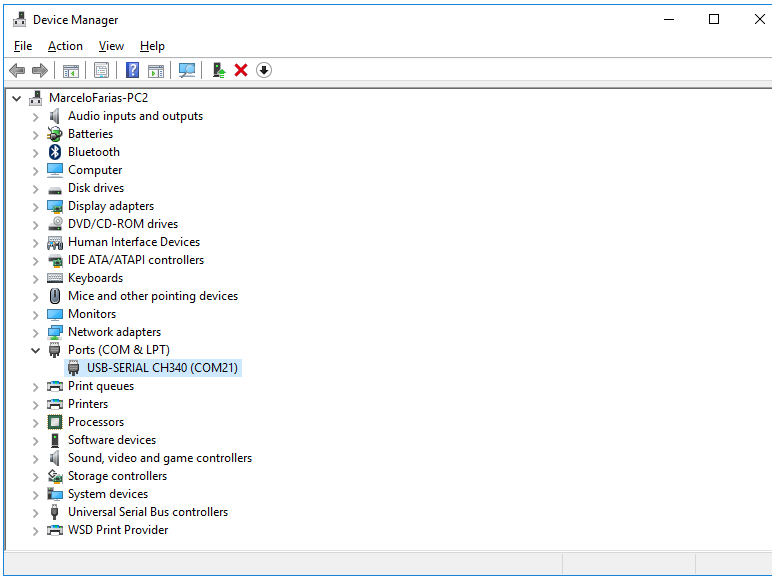


Fig. 7 - Device manager updated with the installed Controllers

## Configure IDE to allow programming Node MCU Board

In this phase is now possible to start the configuration of Arduino IDE to programming Node MCU Board. It means that the configuration of Arduino doesn’t support by default the board corresponding to ESP8266, so it have a lot of configurations to do essentials to programming and access the node MCU board. Note that this configurations steps allow the possibility to access some example files of codes that are not “accessible” for default on Arduino IDE.

In first step launch the Arduino IDE by clicking on IDE shortcut on desktop if it creates or launch by start menu on windows system by typing “Arduino”.

After running of Arduino IDE go to “File”🡪 “Preferences”. It will open the windows of Fig.8 and input this link: <http://arduino.esp8266.com/stable/package_esp8266com_index.json> on “Additional Board Manager URLs” as is showed on Fig.8.

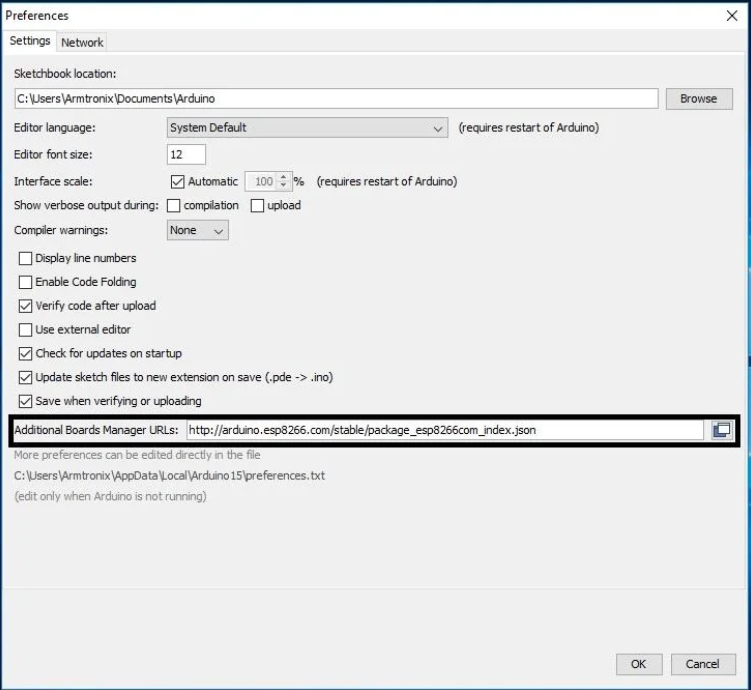


Fig. 8 - Arduino Preferences

The next step is installing the board on Arduino IDE, to do that, go to “Tools”🡪 “Board: …”🡪 “Board Manager”, as it’s possible to see on Fig.9.

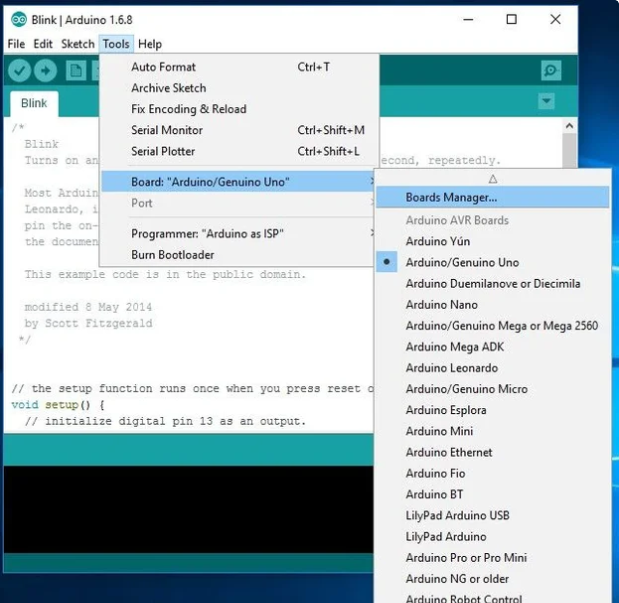


Fig. 9 - Arduino Tools🡪 Board Manager

Then a window called “Board Manager will open and it’s possible to search by the board, for that type “esp8266” on search box. When the search is complete, install the board called “esp8266 by esp8266 community” select the last version and click on install button. It is possible to see this step on Fig.10.

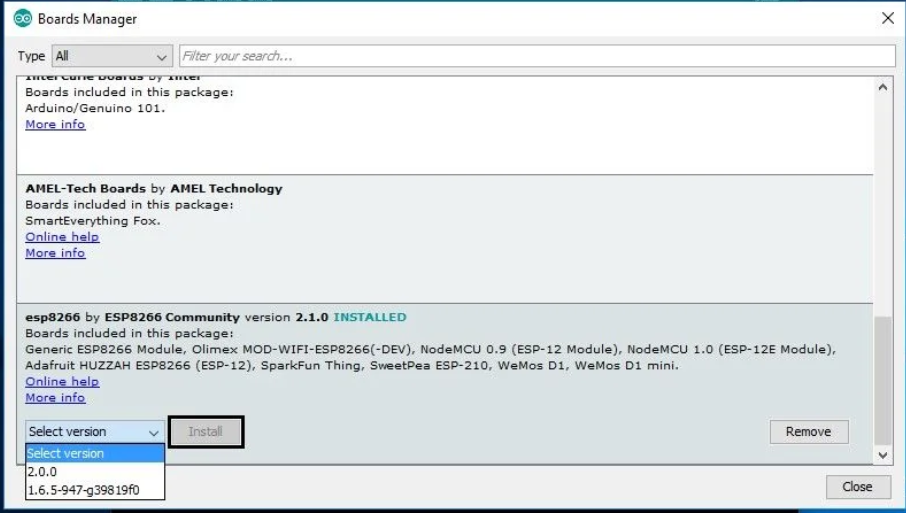


Fig. 10 - Install ESP8266

Wait for the program complete the installation. And after close “Board Manager” window and return to Arduino IDE interface. Go to “Tools”🡪 “Board…” and select the “NodeMCU 1.0 (ESP-12E Module)” as observed on Fig.11.

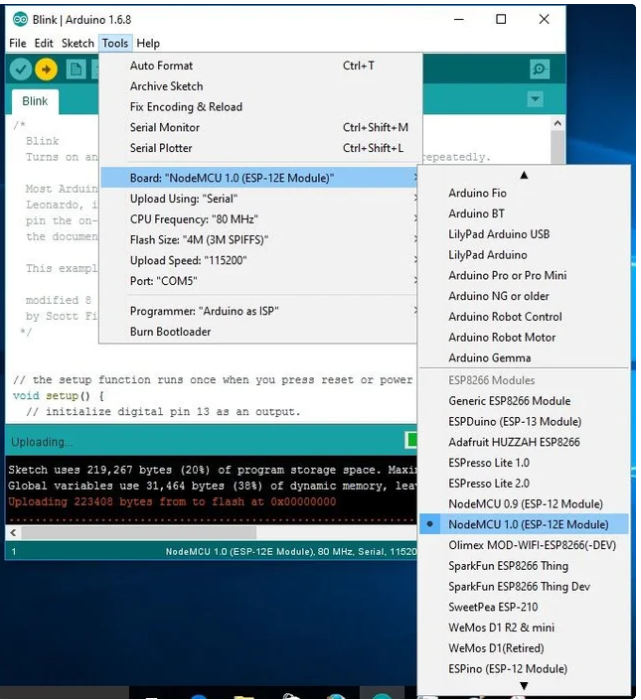


Fig. 11 - Board selection

Now the last step is select on Arduino IDE the correspondent COM for ESP8266 programming and see the Serial Monitor. For that go Back to Arduino Interface and select “Tools” 🡪 “Port…” and click on correspondent com of ESP8266 that is the same installed on Fig.7 of the last section of this document[[1]](#footnote-1), ass it’s possible to see on Fig.12. Exist the possibility of have different ports registered on Arduino IDE because of other services like Bluethooth.

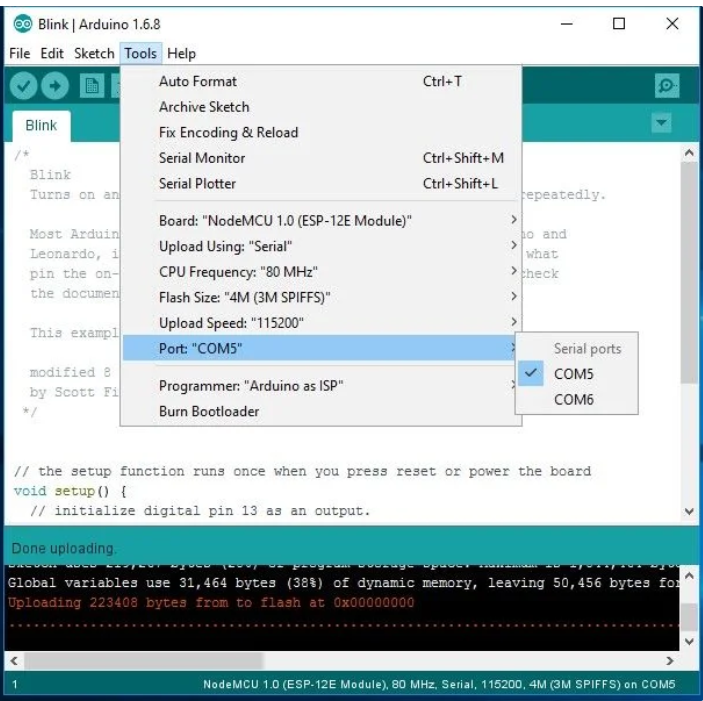


Fig. 12 - Select COM port of ESP8266

## Install and configure Raspberry Pi as MQTT broker

To install de “server” MQTT, it essential installing Mosquitto. It allows data transmission between Node MCU and Raspberry.

To do that, the first step is update and upgrade the Raspbian system. Using “Terminal” type:

* sudo apt-get update
* sudo apt-get upgrade

After this step Raspbian system is now updated, and proceed to the installation of Mosquitto and Mosquitto-clients running the next commands:

* sudo apt-get install -y mosquitto mosquitto-clients

And now active service running the follow command:

* sudo systemctl enable mosquitto.service.

It’s possible to see that with this command Raspbian activate the Mosquitto service and is now possible the non-secure communication. But a good practice is protecting the system against unauthorized access. Before any change stop the mosquitto service typing:

* sudo stop mosquitto

After create a file that is the responsible to save the MQTT credentials like username and password. In future this password is encrypted.

* sudo mosquitto\_passwd -c /etc/mosquitto/passwd <username>

In this step the field <username> must be changed by the username of MQTT Broker and the the Raspbian will request the password twice for a correct verification insert the same password in two times.

The next step is telling the system where credentials are stored and block access to unauthenticated users, so run in terminal the next command:

* sudo nano /etc/mosquitto/mosquitto.conf

And a windows nano will open with the file “mosquitto.conf” that is the preferences file of Mosquitto. In that file insert the following lines

* password\_file /etc/mosquitto/passwd
* allow\_anonymous false

The field allow\_anonymous must be set to “false” to prevent anonymous access.

To end the installation and configuration of Mosquitto service start again the raspberry as broker:

* mosquitto -c /etc/mosquitto/mosquitto.conf

To test if the correct operation run the next command.

* mosquitto\_sub -h localhost -p 1883 -t my topic -u <username> -P <senha>

</senha></username>

1. Note that the Port selected on Fig.7 and on Fig 12will be different **only** in this document. [↑](#footnote-ref-1)