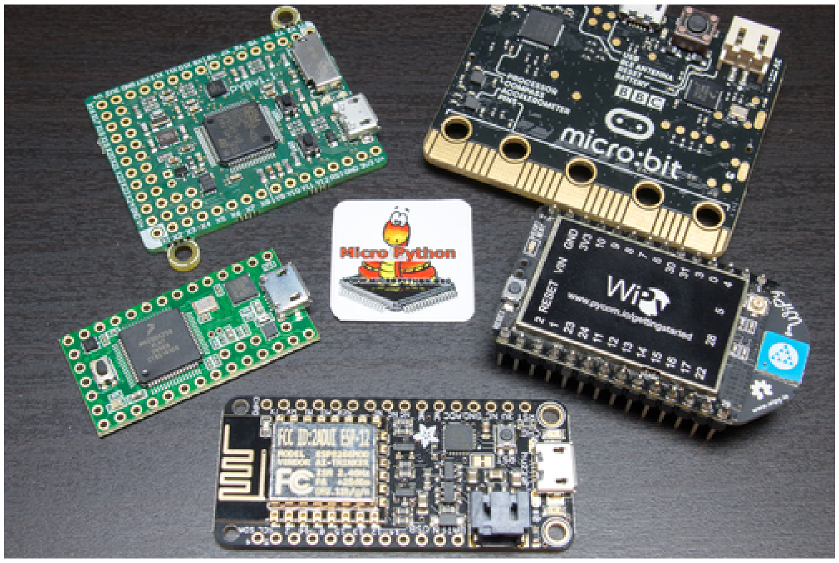
adafruit learning system

MicroPython Basics: How to Load MicroPython on a Board

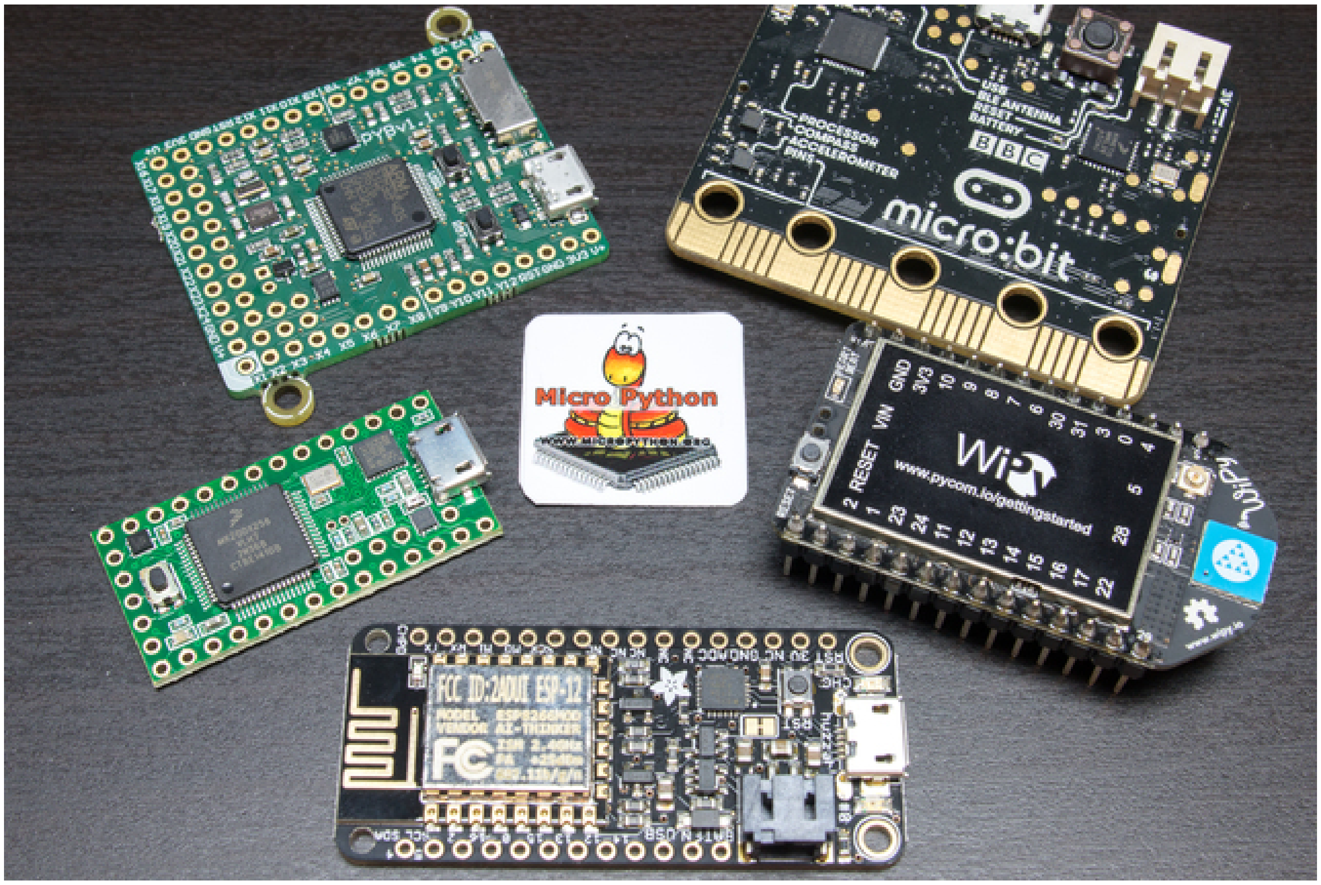
Created by Tony DiCola



Last updated on 2017-08-23 07:45:06 PM UTC

Guide Contents

Overview



The first step to using MicroPython (http://adafru.it/pMa) is loading it onto a development board so you can connect to it and start running Python code that controls hardware. Although some boards come with MicroPython loaded out of the box, in most cases you'll need to load the latest MicroPython firmware on your board to get started. This guide explains how to load MicroPython onto the following development boards:

- pyboard (http://adafru.it/pFb)
- ESP8266 (like the Adafruit Feather HUZZAH ESP8266 (http://adafru.it/n6A))
- SAMD21-based boards like the Feather M0 & Arduino Zero
- WiPy (http://adafru.it/pFf)
- BBC micro:bit (http://adafru.it/pFi)
- Teensy 3.x (http://adafru.it/j7a)

If you aren't familiar with MicroPython be sure to read this guide that explains what is MicroPython (http://adafru.it/pMb)!

When you're ready grab your board and pick the appropriate page in this guide to learn how to load MicroPython.

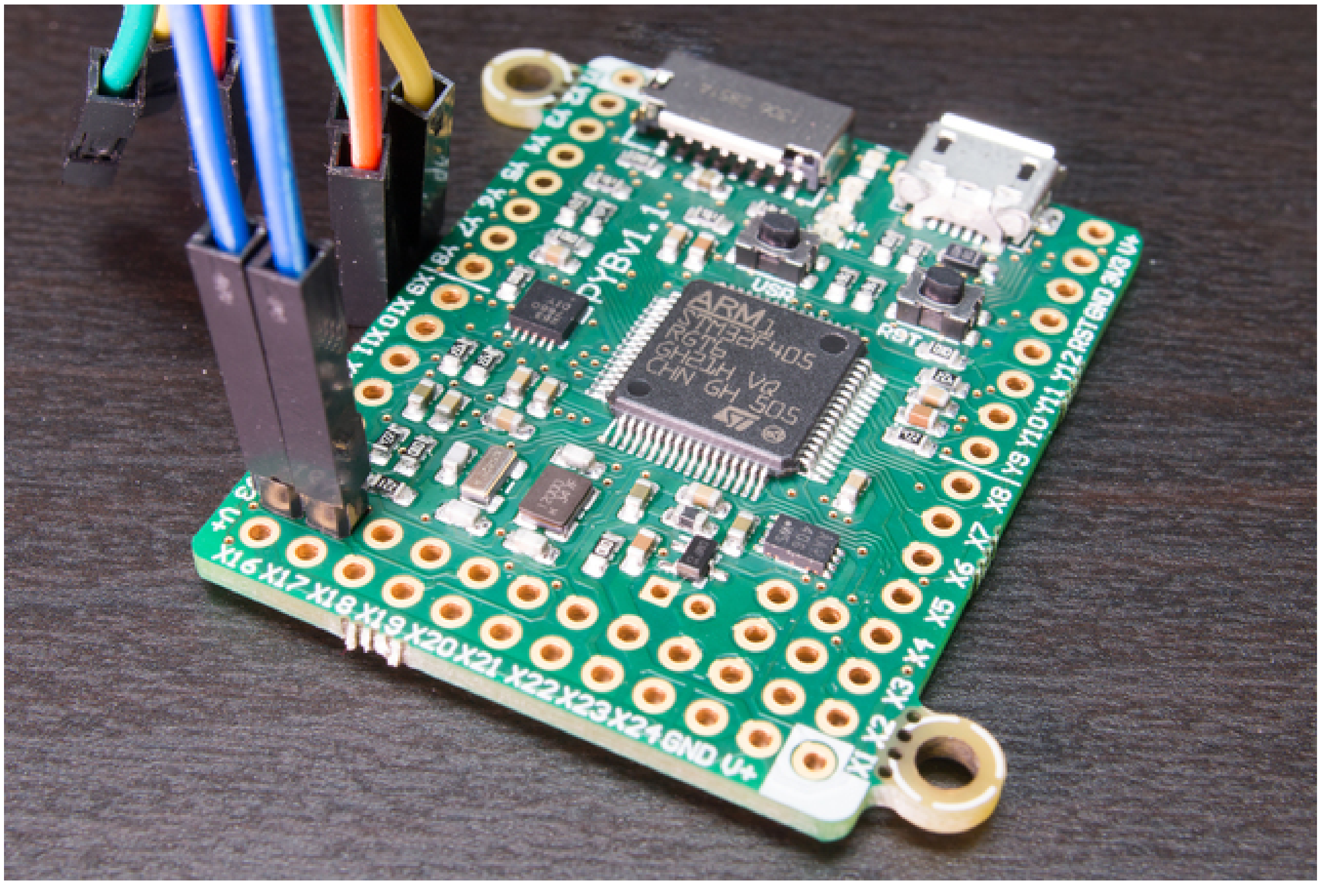
pyboard

The pyboard comes with MicroPython firmware running on it out of the box so you can get started using it right away. If you'd like to update the firmare to the latest version of MicroPython you'll want to carefully follow the official instructions for updating pyboard firmware (http://adafru.it/pMc). At a high level these steps are:

- Download the latest MicroPython firmware (http://adafru.it/pMd) for the pyboard. Be sure to get the version of firmware that matches your pyboard (like v1.0, v1.1, etc.).
- With the board disconnected, connect the DFU pin to the 3.3V pin to enable DFU mode. You might need to solder headers to these pins on the pyboard and connect them with a wire. See the photo below for an example of headers soldered to the pyboard v1.1.
- **On Windows** follow the steps in the instruction PDF (http://adafru.it/pMe). You'll need to at least follow the USB driver section 4 DFU to install a special STM DFU driver. Then follow the "Using DFU to upgrade Micro Python" section at the end to use the DfuSe GUI tool to perform the DFU with the firmware file. After updating the firmware be sure you've followed the CDC serial driver install steps below to access the REPL over a COM port.
- **On Mac OSX or Linux** follow the instructions to use the dfu-util tool (http://adafru.it/pMc) (installed with a package manager like Homebrew on Mac OSX, or your native Linux package manager) to perform the DFU with the firmware file.

Jump to the Serial REPL page (http://adafru.it/pMf) in this guide to learn how to access MicroPython's 'command prompt' on the board.

Example of soldered in headers and wires connecting the DFU pin to 3.3V:



Windows CDC Serial Driver Install

On Windows you'll need to install a special driver to make the pyboard's serial REPL available. Unfortunately this process is a little complicated if you're using Windows 8 or above as you'll need to reboot into safe mode to support the installation of an unsigned driver. Carefully [follow the USB driver section of the pyboard Windows instructions](http://adafru.it/pMe) (http://adafru.it/pMe) to boot into safe mode if necessary and then install the pybcdc.inf driver. Once installed your pyboard should show up as a serial COM port when connected to the computer.

ESP8266

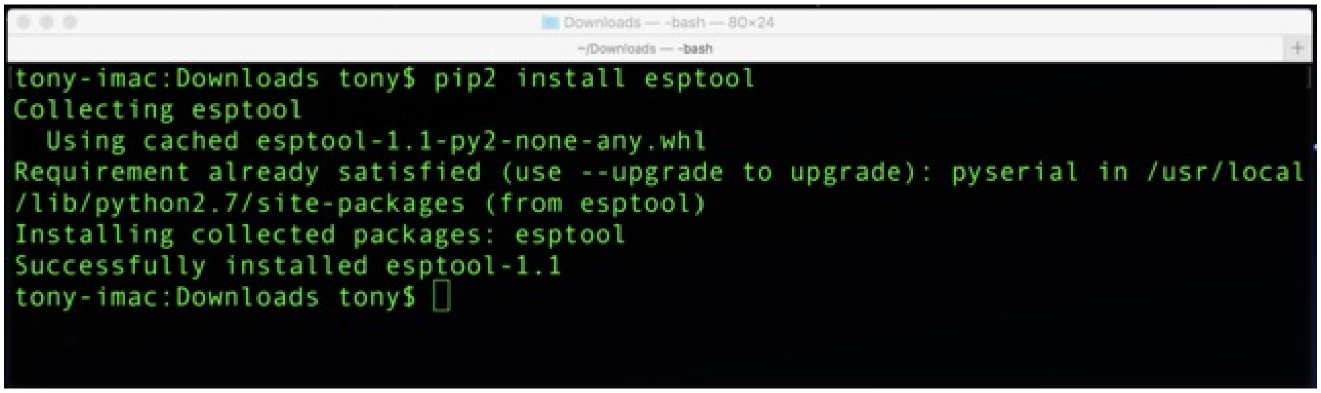
To use MicroPython with the ESP8266 you'll need to first flash it with the latest MicroPython firmware. Follow the official instructions to load MicroPython ESP8266 firmware (http://adafru.it/pMA) which are summarized below.

First install the esptool.py software which enables firmware flashing on the ESP8266. The easiest way to install this tool is from Python's pip package manager. If you don't have it already you'll need to install Python 2.7 (http://adafru.it/fa7) (make sure you check the box to put Python in your system path when installing on Windows) and then run the following command in a terminal:

```
pip install esptool
```

Note on Mac OSX and Linux you might need to run the command as root with sudo, like:

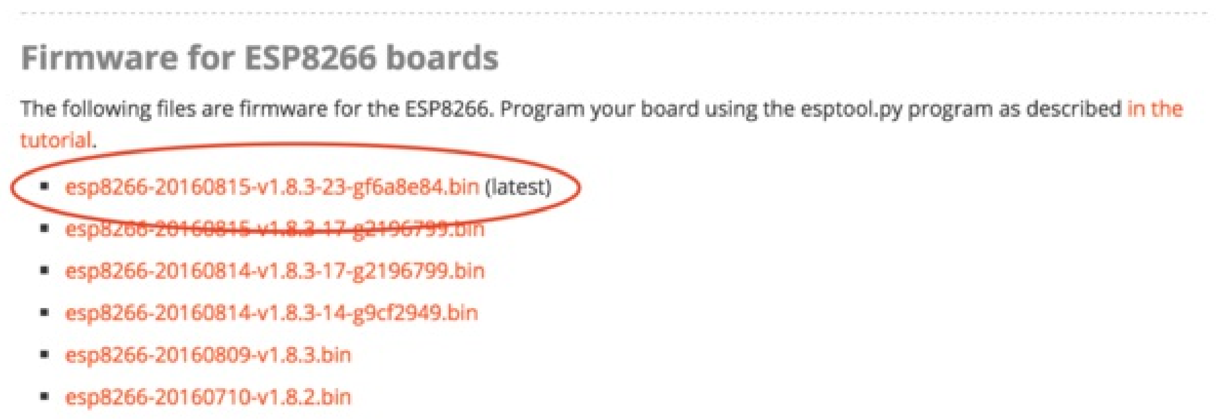
```
sudo pip install esptool
```



```
tony-imac:Downloads tony$ pip2 install esptool
Collecting esptool
  Using cached esptool-1.1-py2-none-any.whl
Requirement already satisfied (use --upgrade to upgrade): pyserial in /usr/local
/lib/python2.7/site-packages (from esptool)
Installing collected packages: esptool
Successfully installed esptool-1.1
tony-imac:Downloads tony$ 
```

If you receive an error that esptool.py only supports Python 2.x try running again with the **pip2** command instead of pip (likely your system is using Python 3 and the pip command is getting confused which version to use).

Next download the latest MicroPython ESP8266 firmware file(http://adafru.it/pMd). These firmware files are generated daily from the latest code on Github. If you'd like to build your own MicroPython ESP8266 firmware (like if you're customizing it or including other modules & scripts) check out this handy guide on compiling firmware in a special virtual machine (http://adafru.it/pMB).


Firmware for ESP8266 boards

The following files are firmware for the ESP8266. Program your board using the esptool.py program as described in the tutorial.

- esp8266-20160815-v1.8.3-23-gf6a8e84.bin (latest)
- esp8266-20160815-v1.8.3-17-g2196799.bin
- esp8266-20160814-v1.8.3-17-g2196799.bin
- esp8266-20160814-v1.8.3-14-g9cf2949.bin
- esp8266-20160809-v1.8.3.bin
- esp8266-20160710-v1.8.2.bin

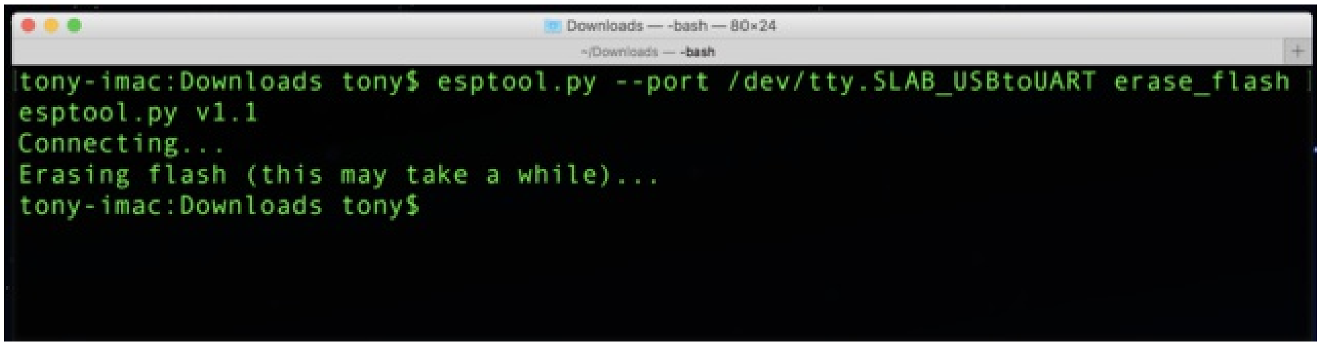
Now you'll need to put the ESP8266 into its firmware flashing mode. Each ESP8266 board is slightly different:

- **For a raw ESP8266 module** you'll need to wire up buttons to ground for the GPIO0 and RESET pins. Hold the GPIO0 button down (or connect the line to ground) and while still holding GPIO0 to ground press and release the RESET button (or connect and release the line from ground), then release GPIO0.
- **For the HUZZAH ESP8266 breakout (http://adafru.it/f9X)** buttons for GPIO0 and RESET are built in to the board. Hold GPIO0 down, then press and release RESET (while still holding GPIO0), and finally release GPIO0.
- **For the Feather HUZZAH ESP8266 (http://adafru.it/n6A)** you don't need to do anything special to go into firmware flashing mode. This board is built to detect when the serial port is opened for flashing and automatically configure the ESP8266 module to receive firmware. **Be sure to first install the SiLabs CP210x driver (http://adafru.it/jCs) on Windows and Mac OSX to make the board's serial port visible!** On Windows you want the normal VCP driver, not the 'with Serial Enumeration' driver.

It's recommended to erase the entire flash memory of the ESP8266 board before uploading firmware. Run the following command in a terminal to perform this erase:

```
esptool.py --port /path/to/ESP8266 erase_flash
```

Where **/path/to/ESP8266** is the path or name of the serial port that is connected to the ESP8266. The exact name of the device varies depending on the type of serial to USB converter chip so you might need to look at the serial ports with and without the device connected to find its name.

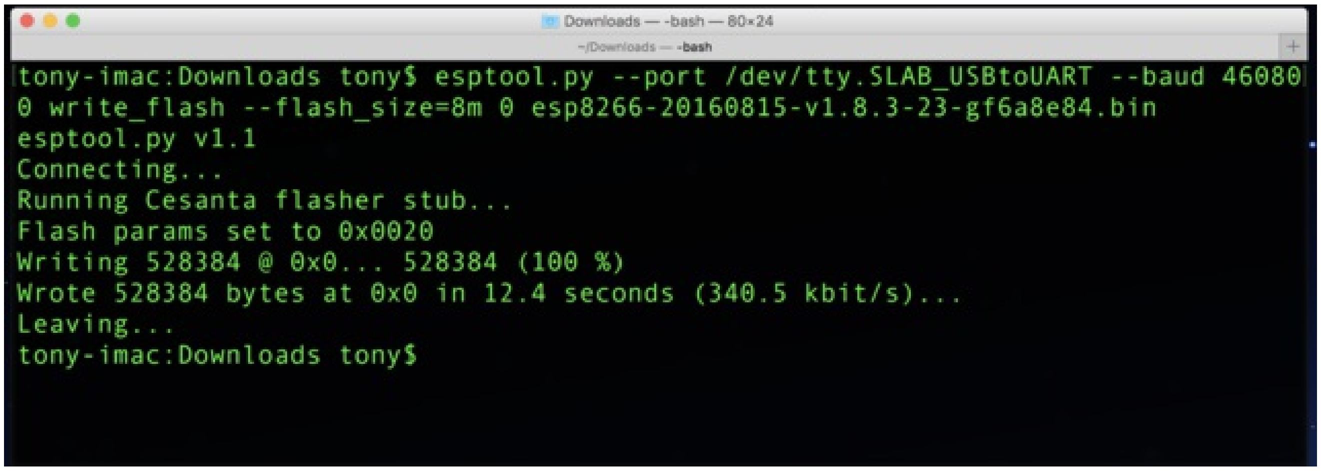


```
tony-imac:Downloads tony$ esptool.py --port /dev/tty.SLAB_USBtoUART erase_flash
esptool.py v1.1
Connecting...
Erasing flash (this may take a while)...
tony-imac:Downloads tony$
```

Now put the ESP8266 back into firmware flashing mode and run the following command to load the downloaded firmware file:

```
esptool.py --port /path/to/ESP8266 --baud 460800 write_flash --flash_size=detect 0 firmware.bin
```

Again set /**path/to/ESP8266** to the path or name of the serial port that is connected to the ESP8266. In addition set **firmware.bin** to the name or path to the firmware file you would like to load.



```
tony-imac:Downloads tony$ esptool.py --port /dev/tty.SLAB_USBtoUART --baud 46080
0 write_flash --flash_size=8m 0 esp8266-20160815-v1.8.3-23-gf6a8e84.bin
esptool.py v1.1
Connecting...
Running Cesanta flasher stub...
Flash params set to 0x0020
Writing 528384 @ 0x0... 528384 (100 %)
Wrote 528384 bytes at 0x0 in 12.4 seconds (340.5 kbit/s)...
Leaving...
tony-imac:Downloads tony$
```

Once the tool finishes flashing the firmware (you'll usually see a blue light on the ESP8266 module flashing during this process) press the RESET button on the ESP8266 board or disconnect and reconnect it to your computer. You should be all set to start using the latest MicroPython firmware on the board!

Note that if you see an error that detect is not a valid flash_size parameter you might be using an older version of esptool.py. To upgrade to the latest version run the following command:

```
pip install --upgrade esptool
```

Use sudo as necessary on Linux or Mac OSX. Try the flash command again after esptool.py is upgraded.

Jump to the Serial REPL page (http://adafru.it/pMf) in this guide to learn how to access

MicroPython's 'command prompt' on the board.

Feather/Metro M0 / Arduino Zero

For SAMD21-based boards like the Feather M0, Metro M0, Arduino Zero, and more they use CircuitPython which is Adafruit's open source derivative of MicroPython. See the what is MicroPython guide (http://adafru.it/pMb) for a more detailed look at the differences between CircuitPython and MicroPython.

To load CircuitPython onto a SAMD21-based board follow the instructions from your board's guide to learn how to load that latest firmware using a special UF2 bootloader mode of the board. With the UF2 firmware loading CircuitPython is as simple as dragging a file onto a USB drive. See the Metro M0 Express CircuitPython guide page (http://adafru.it/xAX) (or a guide specific to your board) for details on loading CircuitPython.

WiPy

The WiPy comes with MicroPython firmware preloaded out of the box. If you'd like to upgrade to the latest firmware be sure to follow the official instructions (http://adafru.it/pMC). At a high level the steps are:

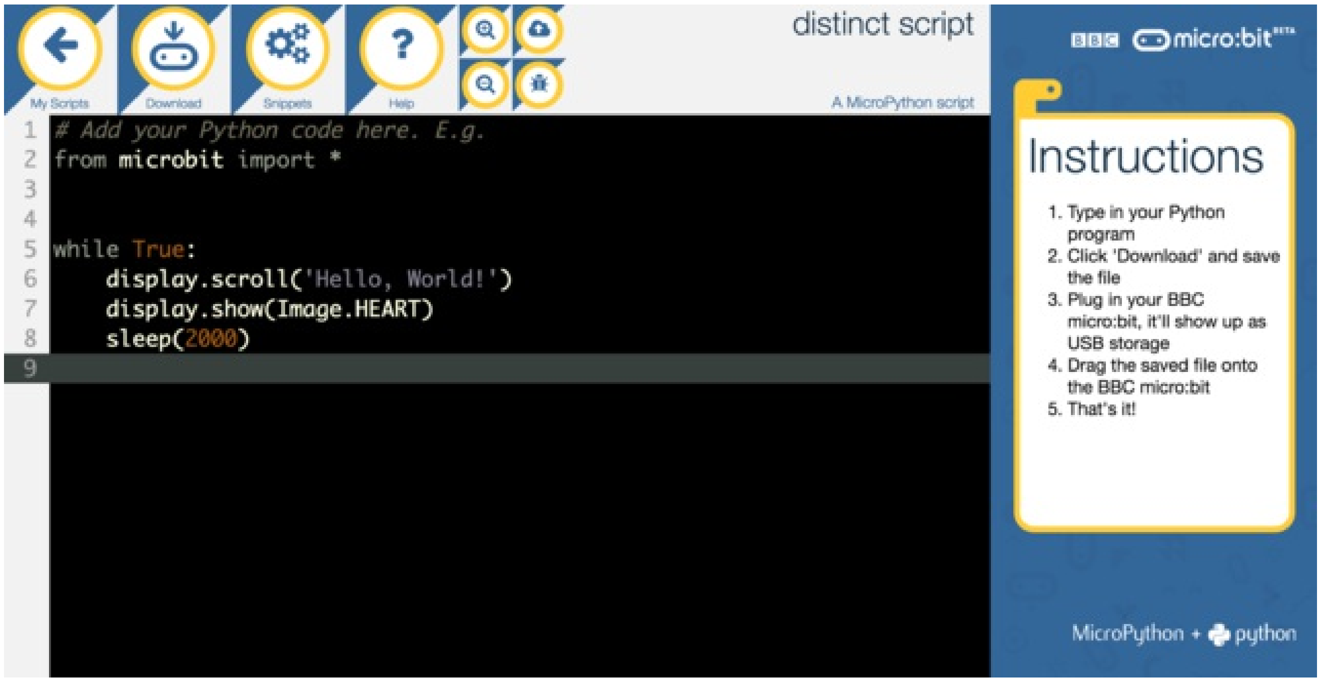
- Connect to the WiPy board's wireless access point. See thegetting started instructions (http://adafru.it/pMD) for how to connect to the board.
- Download the latest WiPy firmware file (http://adafru.it/pMd).
- Use a FTP client (likeFilezilla (http://adafru.it/d3S) but note these special FileZilla configuration instructions (http://adafru.it/pMC)) to upload the firmware file to the board (http://adafru.it/pMC).

BBC micro:bit

Loading MicroPython on the BBC micro:bit is very easy. When you plug the board into your computer it will appear as a USB storage drive. If you drag a firmware .hex file to the drive it will be programmed to the board. You can use either the online BBC micro:bit code editor (http://adafru.it/pME) or the desktop mu MicroPython code editor (http://adafru.it/pMF) to enter MicroPython code and push it to the board.

Note on Windows if you'd like to access the board's REPL over a serial connection you'll need to follow the Windows CDC serial driver install steps below.

With the online editor (http://adafru.it/pFi) you'll download a .hex file and drag it to the microbit's drive.

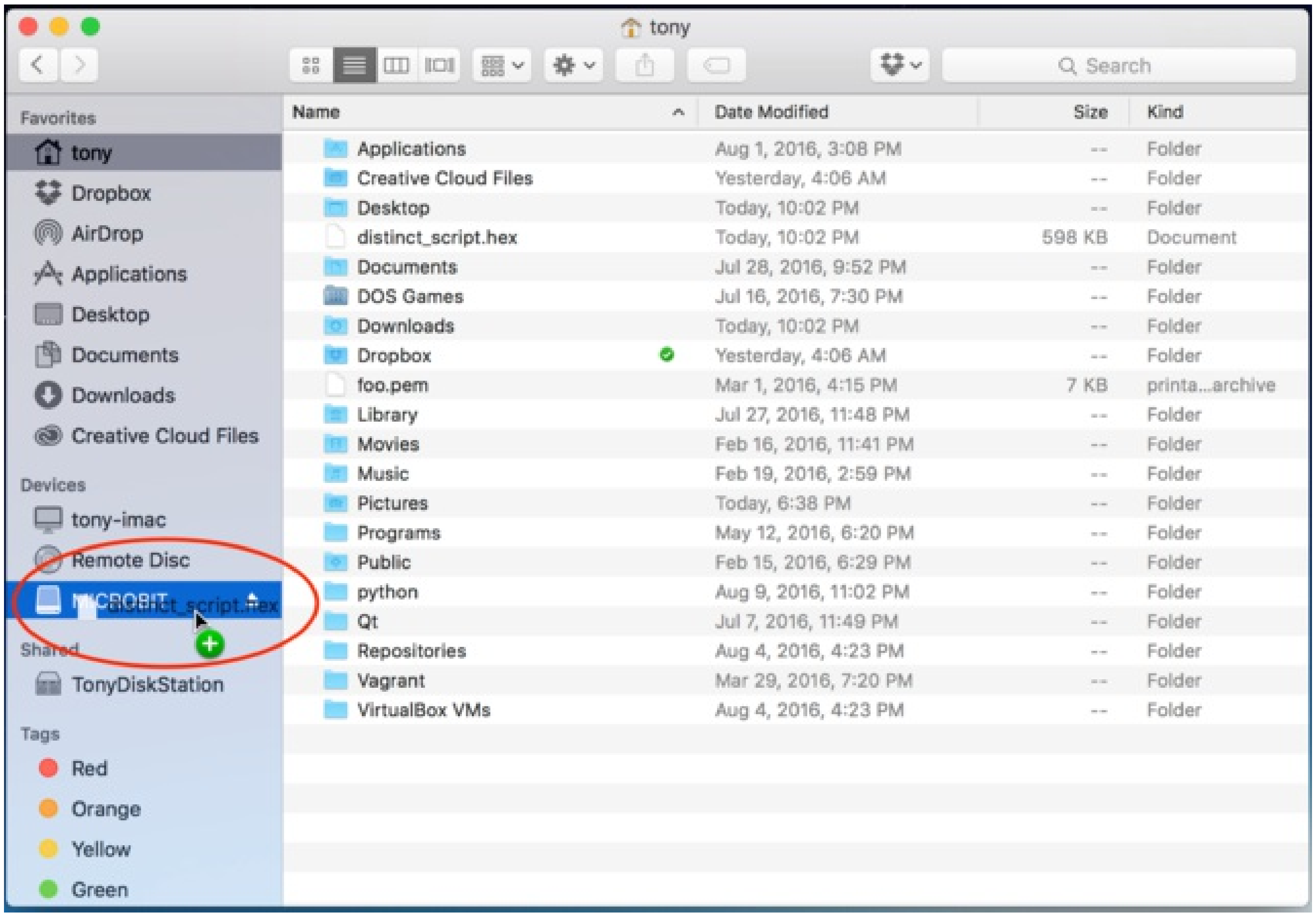


```
# Add your Python code here. E.g.
from microbit import *


while True:
    display.scroll('Hello, World!')
    display.show(Image.HEART)
    sleep(2000)
```

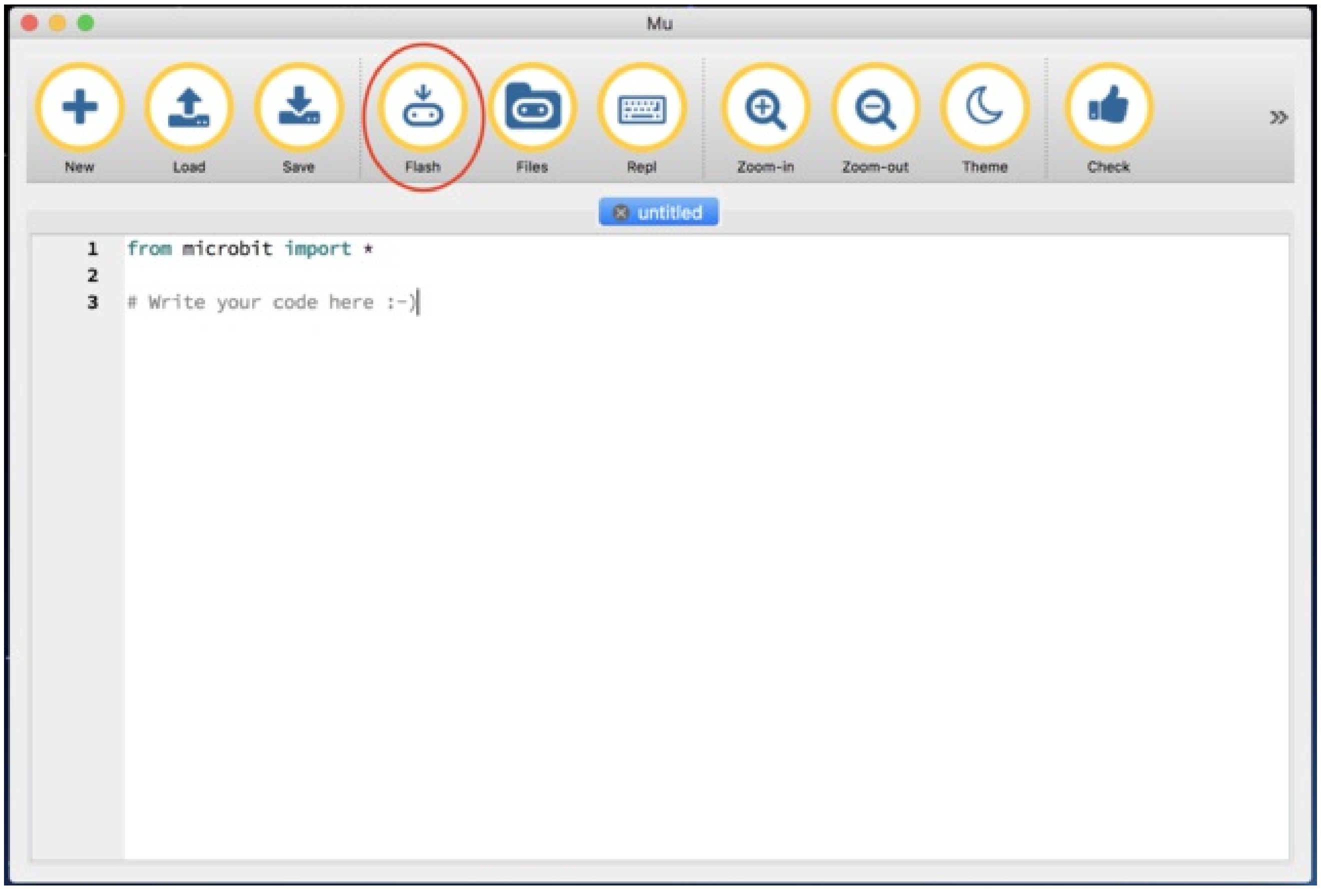
Instructions

1. Type in your Python program
2. Click 'Download' and save the file
3. Plug in your BBC micro:bit, it'll show up as USB storage
4. Drag the saved file onto the BBC micro:bit
5. That's it!



Name	Date Modified	Size	Kind
Applications	Aug 1, 2016, 3:08 PM	--	Folder
Creative Cloud Files	Yesterday, 4:06 AM	--	Folder
Desktop	Today, 10:02 PM	--	Folder
distinct_script.hex	Today, 10:02 PM	598 KB	Document
Documents	Jul 28, 2016, 9:52 PM	--	Folder
DOS Games	Jul 16, 2016, 7:30 PM	--	Folder
Downloads	Today, 10:02 PM	--	Folder
Dropbox	Yesterday, 4:06 AM	--	Folder
foo.pem	Mar 1, 2016, 4:15 PM	7 KB	printa...archive
Library	Jul 27, 2016, 11:48 PM	--	Folder
Movies	Feb 16, 2016, 11:41 PM	--	Folder
Music	Feb 19, 2016, 2:59 PM	--	Folder
Pictures	Today, 6:38 PM	--	Folder
Programs	May 12, 2016, 6:20 PM	--	Folder
Public	Feb 15, 2016, 6:29 PM	--	Folder
python	Aug 9, 2016, 11:02 PM	--	Folder
Qt	Jul 7, 2016, 11:49 PM	--	Folder
Repositories	Aug 4, 2016, 4:23 PM	--	Folder
Vagrant	Mar 29, 2016, 7:20 PM	--	Folder
VirtualBox VMs	Aug 4, 2016, 4:23 PM	--	Folder

With the desktop mu editor (http://adafru.it/pMF) just press the flash button with the board connected and it should automatically upload the MicroPython code & firmware to the board.



```
from microbit import *

# Write your code here :-)
```

Jump to the Serial REPL page (http://adafru.it/pMf) in this guide to learn how to access MicroPython's 'command prompt' on the board.

Windows CDC Serial Driver Install

To access the BBC micro:bit's serial port follow theinstructions to install the mbed serial port driver (http://adafru.it/pNa). Once installed the BBC micro:bit should show up as a COM port in Device Manager when connected.

Teensy 3.x

MicroPython on the Teensy 3.x microcontroller is for advanced users only! You'll need to be comfortable setting up an ARM toolchain and compiling code for the board.

With the Teensy 3.x series of boards you'll need to do a bit of work to load them with MicroPython. There is no pre-built firmware for these boards so you'll need to follow the instructions from MicroPython's teensy port code (http://adafru.it/pNb) to compile the firmware yourself.

Note that the Teensy MicroPython port instructions are written assuming you are using Linux. Consider setting up a Vagrant (http://adafru.it/epl) Linux virtual machine for building firmware if you aren't using Linux.

In addition the instructions link to pages for setting up the GCC ARM toolchain which unfortunately no longer exist. You will likely need to setup the toolchain on your own to follow the instructions. As mentioned above getting MicroPython on the Teensy 3.x board is only for advanced users right now!

Serial REPL

For most MicroPython boards you can access the MicroPython REPL (read-evaluate-print loop) over their USB serial connections. Be sure to first follow any instructions and install drivers specific to your board to ensure you can access the USB serial connection:

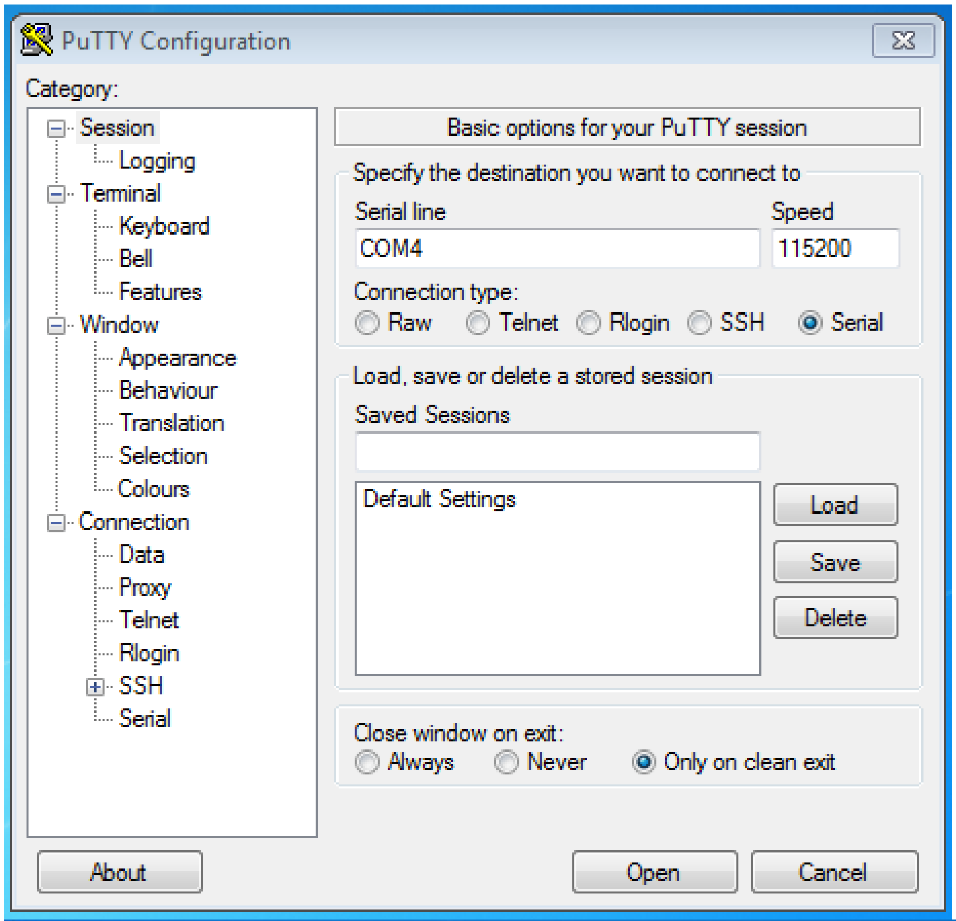
- pyboard needs drivers to be installed on Windows (http://adafru.it/pNc)
- Feather HUZZAH ESP8266 needs drivers to be installed on Mac OSX or Windows (http://adafru.it/jCs)
- BBC micro:bit needs drivers to be installed on Windows (http://adafru.it/pNd)

Note the WiPy board REPL is only accessible over its wireless interface with telnet. Follow the WiPy instructions for connecting to the telnet REPL (http://adafru.it/pMC) instead of the steps below for accessing with a serial terminal.

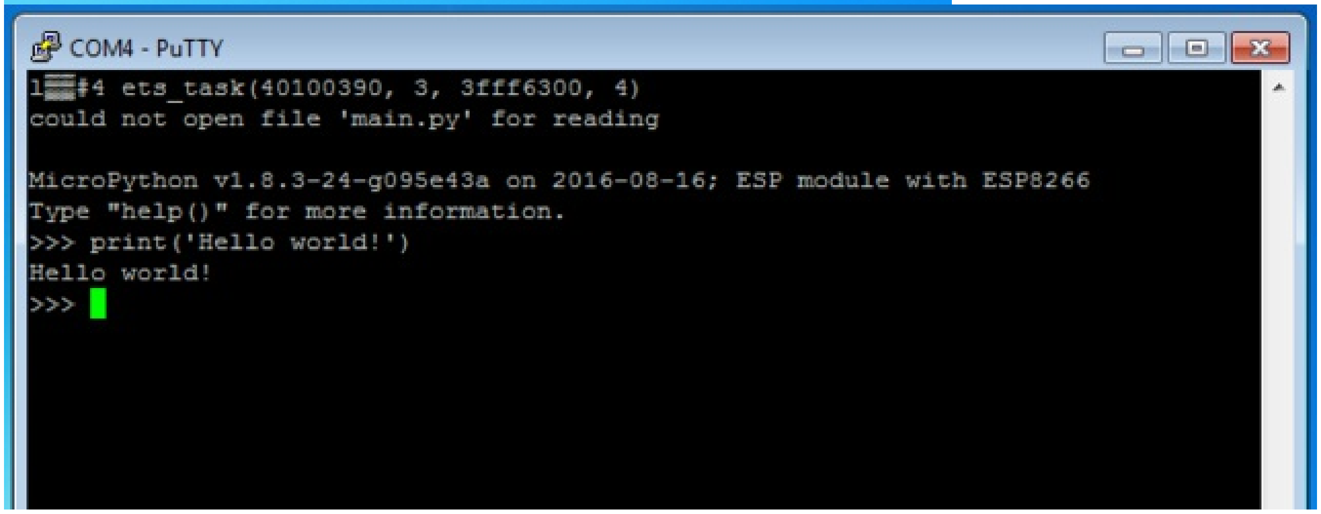
Determine the name of the serial port for your board. It's easiest to look at the serial ports with the board disconnected (on Windows check **Device Manager** under the **Ports (COM/LPT)** node, or on Mac OSX/Linux run the **ls /dev/tty.*** command in a terminal), then connect the board and look at the serial ports again to find the newly added port.

See the videos and and notes below for details on tools to access the REPL from different platforms.

On Windows you'll want to use a tool like PuTTY (http://adafru.it/pNe) to connect to the serial port. Download and run PuTTY, then configure it to use a serial connection to the board's COM port at 115200 baud similar to as shown below:



Serial line: COM4
Speed: 115200
Connection type: Raw Telnet Rlogin SSH [✓ Serial]
Close window on exit: Always Never [✓ Only on clean exit]



```
1▒▒#4 ets_task(40100390, 3, 3fff6300, 4)
could not open file 'main.py' for reading

MicroPython v1.8.3-24-g095e43a on 2016-08-16; ESP module with ESP8266
Type "help()" for more information.
>>> print('Hello world!')
Hello world!
>>>
```

On Linux or Mac OSX the screen command can be used to connect to the serial port. Run the following command to connect at 115200 baud:

```
screen /dev/tty.board_name 115200
```

Where **/dev/tty.board_name** is the name of the board's serial port.

When you're done using screen most versions of it allow you to exit by pressing **Ctrl-a** then **k** then **y** or presing **Ctrl-a** then typing **:quit** and pressing **enter**.

After you're connected to the serial REPL try pressing enter to confirm you see the **>>>** prompt. You can also type **help()** and press enter on most boards to see basic usage information.

If you can't get a **>>>** prompt to appear try pressing **Ctrl-c** a couple times to interrupt any running program on the board.

That's all there is to connecting to the board's serial REPL, you're ready to start typing in and running MicroPython code!