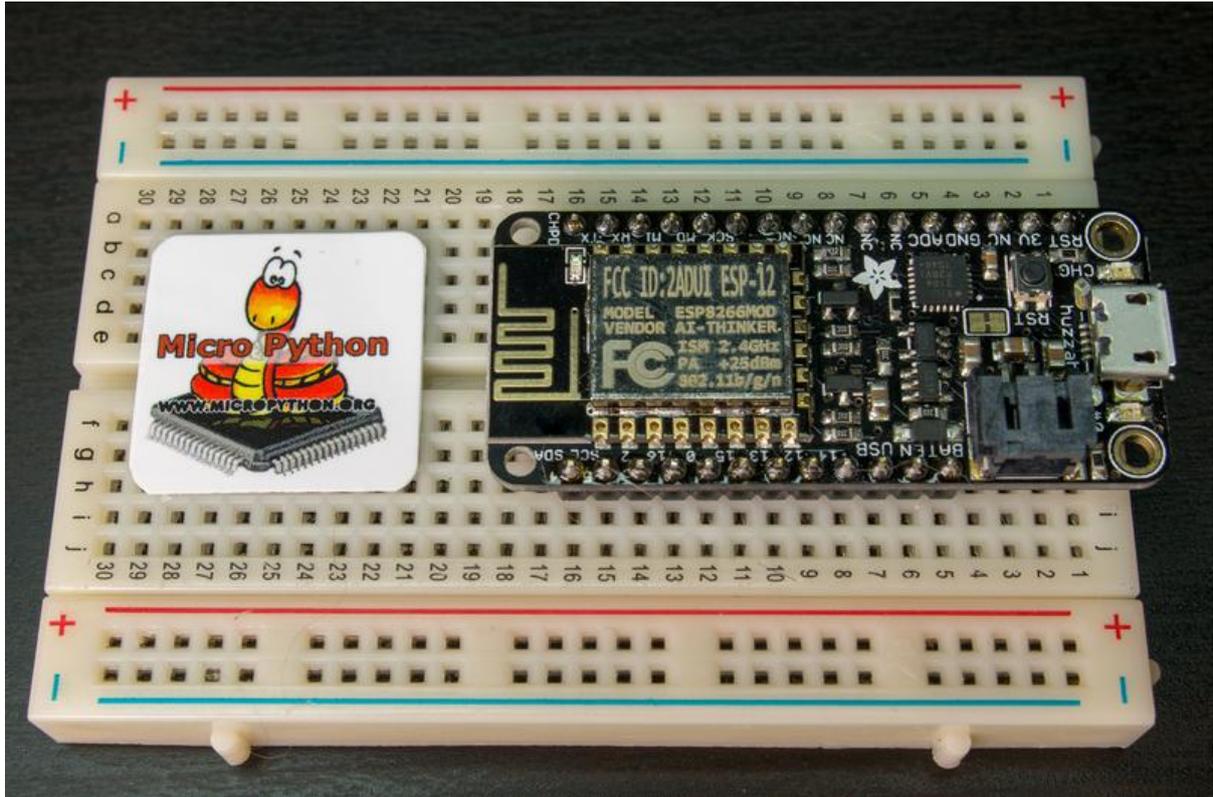




MicroPython Basics: ESP8266 WebREPL

Created by Tony DiCola



<https://learn.adafruit.com/micropython-basics-esp8266-webrepl>

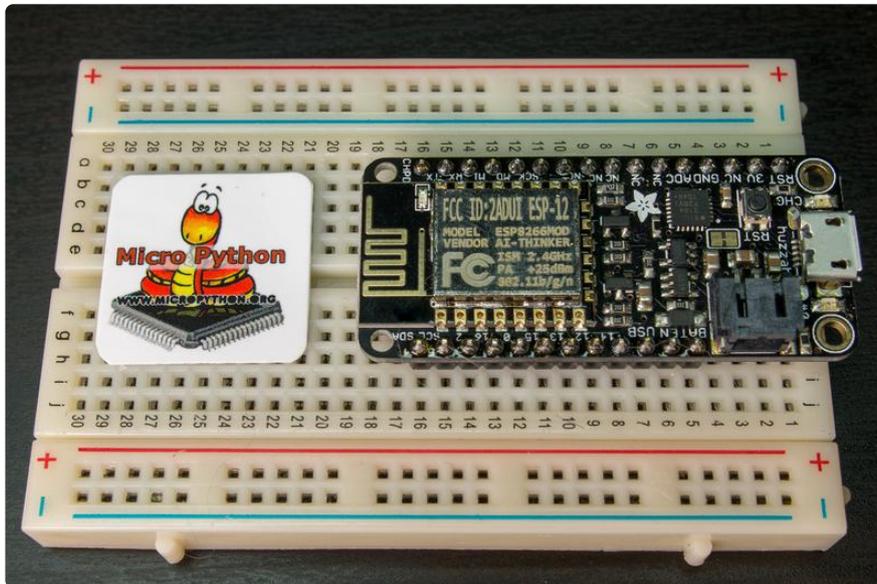
Last updated on 2023-08-29 03:12:47 PM EDT

Table of Contents

Overview	3
Access WebREPL	4
<ul style="list-style-type: none">• Enable WebREPL• Access WebREPL• Access WebREPL From Other WiFi Networks	
Send & Get Files	9

Overview

Note this guide was written for MicroPython.org firmware and not Adafruit CircuitPython firmware.



One unique feature of [MicroPython \(\)](#) on the ESP8266 and its WiFi microcontroller is a WebREPL (read-evaluate-print loop, like a Python 'command line') accessible through a webpage. Instead of using a serial connection to the board you can run Python code directly from your browser in a simple terminal. You don't even need to connect the board to a WiFi network, it can create its own that you use to access the WebREPL!

This guide will walk through how to enable and access the WebREPL on MicroPython ESP8266. Note only the ESP8266 version of MicroPython supports the WebREPL, and be aware the WebREPL is an experimental feature that is under active development. Follow this guide to learn how to explore the WebREPL on MicroPython ESP8266 yourself!

Before you get started make sure you have a ESP8266 running the latest MicroPython firmware. In addition you'll want to be familiar with the basics of MicroPython by reading these guides first:

- [MicroPython Basics: What is MicroPython? \(\)](#)
- [MicroPython Basics: How to Load MicroPython on a Board \(\)](#)
- [MicroPython Basics: Load Files & Run Code \(\)](#)

Access WebREPL

Enable WebREPL

The first step to using the WebREPL is to make sure it is available and enabled on your ESP8266 board. As of the latest 1.8.6 MicroPython releases the WebREPL is not enabled by default and must be switched on with a one time command run through the serial port. Make sure you're using [MicroPython 1.8.6 or higher \(\)](#) to follow the below instructions.

Check out this [handy guide on loading MicroPython firmware \(\)](#) if you aren't sure how to load firmware.

To enable the WebREPL connect to the board using its serial connection and run the following command as [mentioned in the documentation \(\)](#):

```
import webrepl_setup
```

You will be prompted to enable or disable the WebREPL, and to set a password. Enter your choices for both prompts then press y when prompted to save and reboot.

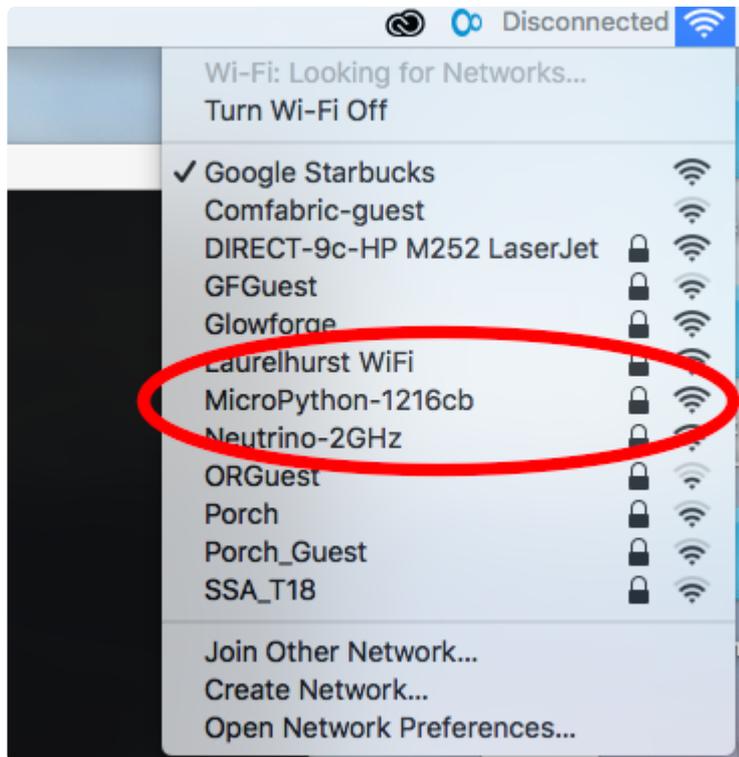
```
>>> import webrepl_setup
WebREPL daemon auto-start status: disabled

Would you like to (E)nable or (D)isable it running on boot?
(Empty line to quit)
> E
To enable WebREPL, you must set password for it
New password: python
Confirm password: python
Changes will be activated after reboot
Would you like to reboot now? (y/n) y
>>>
```

Access WebREPL

Once the WebREPL is running you can connect to it over the ESP8266's WiFi radio. If the board isn't connected already to a wireless access point then it will create one which you can connect to from your computer.

To connect to the board's WiFi network look for a WiFi network advertised as MicroPython-*



Where the * is the a set of numbers and letters unique to each ESP8266 board (based on its MAC address).

You will need to enter the password micropythoN to connect to the board's access point (the N is capitalized!). If you have trouble see more details on [connecting to the board's WiFi network \(\)](#) in the official documentation.

Next you'll need to have available the WebREPL client web page on your machine. You can either [access the client webpage from micropython.org \(\)](#) (assuming your computer can be connected to both the internet and your MicroPython board's network) or [download it from its home on GitHub \(\)](#) by clicking the button below and extracting the files:

Download MicroPython WebREPL Client

If you download the WebREPL client extract the files and open webrepl.html in your browser (Firefox and Chrome are the officially supported browsers at this time).

Note you can't access the WebREPL client page directly from the ESP8266 board. To save space in flash memory the ESP8266 only provides a websocket interface and not the actual client webpage. You'll need to download the WebREPL client page ahead of time or access it from the internet.

Once the WebREPL client page is loaded you should see something like the following (as of ~August 2016, be aware the WebREPL is under active development and might change how it looks over time!):



Now make sure your computer is connected to the ESP8266's WiFi network, then click the Connect button (using the default address of ws://192.168.4.1:8266/ next to the button).

Once connected the WebREPL should prompt you to enter the password you set while running the webrepl_setup command earlier. Enter the password that you set to continue.



Try entering Python code at the REPL to see the output of it running. The WebREPL should function just like the serial REPL so you can use it to experiment with MicroPython, run code you've loaded on the board, and more:



That's all there is to accessing the WebREPL from the ESP8266's WiFi network!

Access WebREPL From Other WiFi Networks

Accessing the WebREPL over the ESP8266's own WiFi access point like above is handy if there's no router or other WiFi network available. However if you do have access to a WiFi network you can instead connect the ESP8266 to this network and access the WebREPL from any other machine on the network. This makes it easier to use the WebREPL without losing access to the internet or your own WiFi network on a laptop that can only be connected to one WiFi network at a time.

To use the WebREPL on another WiFi network you'll first need to connect the ESP8266 to the WiFi network. The [MicroPython ESP8266 documentation shows \(\)](#) a few simple commands to connect to a network. From the serial REPL (don't use the WebREPL since you'll lose access when the commands change to a different WiFi network!) run the following commands:

```
import network
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect('ssid', 'password')
```



```
>>> import network
>>> wlan = network.WLAN(network.STA_IF)
>>> wlan.active(True)
>>> wlan.connect('tony-2.4', ' ')
>>>
```

Where ssid is the name of the WiFi access point SSID, and password is the password required to use the access point (try an empty string "" if no password is required).

After a moment the ESP8266 should connect to the WiFi network. You can run the ifconfig function to check that the board has been assigned an IP address and is on the network:

```
wlan.ifconfig()
```



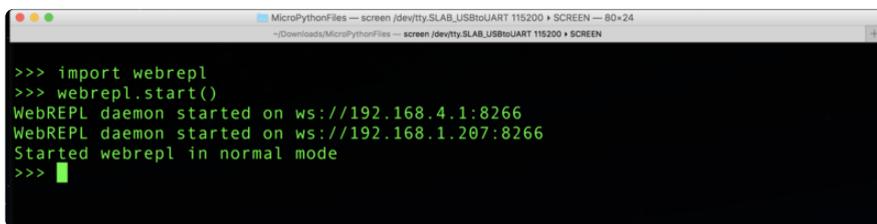
```
>>> wlan.ifconfig()
('192.168.1.207', '255.255.255.0', '192.168.1.1', '192.168.1.1')
>>>
```

Notice the first result from the ifconfig call is the IP address assigned to the ESP8266 (192.168.1.207 in this example, but it will differ depending on your WiFi network). If

you don't see an IP address then something is wrong with the WiFi configuration or connection to the network. Double check the SSID and password are correct and try the connect call again.

One important thing to note is that the ESP8266 will always remember the last WiFi network it used and attempt to connect automatically on reboot. This is handy since it means when the ESP8266 boots up it will automatically connect to the last network without you having to run the commands above again. The board will only remember the last network though and not an entire history of older networks.

Once the board is connected to the WiFi network enable the WebREPL (as described in the previous section) and notice it prints two IP addresses that it's listening on:

A terminal window showing the execution of Python code to start the WebREPL daemon. The output shows two IP addresses: 192.168.4.1 and 192.168.1.207.

```
>>> import webrepl
>>> webrepl.start()
WebREPL daemon started on ws://192.168.4.1:8266
WebREPL daemon started on ws://192.168.1.207:8266
Started webrepl in normal mode
>>> █
```

In this case the WebREPL is listening on 192.168.4.1 which is the ESP8266's access point, and 192.168.1.207 which is the IP address of the ESP8266 on a WiFi network.

Now open the WebREPL and use the IP address of the board on your network instead of the 192.168.4.1 default:

A terminal window showing a successful connection to the WebREPL daemon on the IP address 192.168.1.207. The connection is established and the prompt is ready for input.

```
ws://192.168.1.207:8266 Disconnect
Welcome to MicroPython!
Password:
WebREPL connected
>>> █
```

Once connected you can use the WebREPL just as you were using it on the ESP8266's WiFi network!

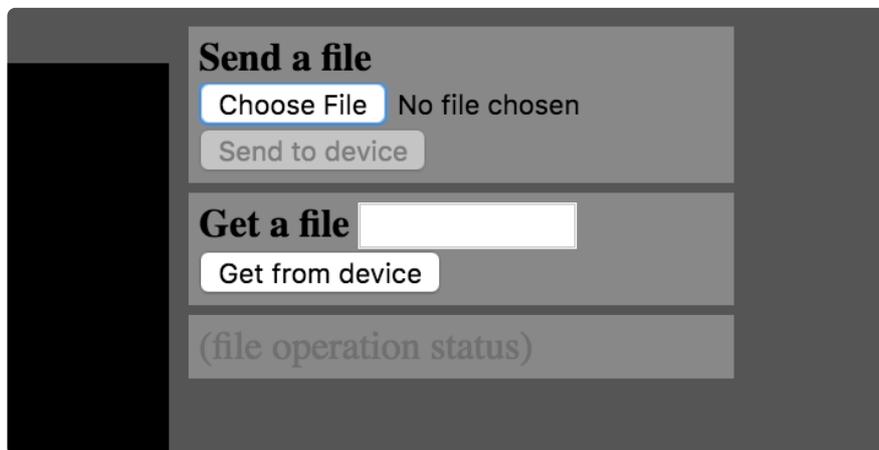
Note if you're connecting the board to a WiFi network it is a very smart idea to [disable debug output \(\)](#) since connecting and disconnecting WiFi network and can spew debug output to the serial REPL. If the board can't connect to the WiFi network, like if you move it to a different location away from the network, you'll see a constant stream of debug output as the board tries and fails to reconnect to the network. This debug output can cause problems with tools like ampy that try to use the board's serial REPL and don't expect the debug output.

Send & Get Files

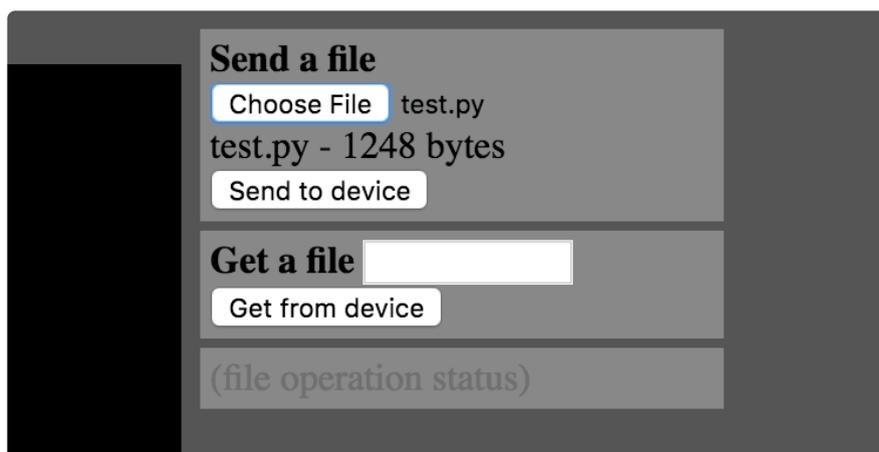
One handy feature of the WebREPL is that it can send and get files from the board's filesystem. [Tools like ampy can send and get files over the serial REPL \(\)](#) too, but accessing files over the WebREPL can sometimes be more convenient.

Be aware file access using the WebREPL is very simple and doesn't allow uploading to specific locations on the filesystem (everything is uploaded to the root). You'll need to [use other tools \(\)](#) to put files in different locations.

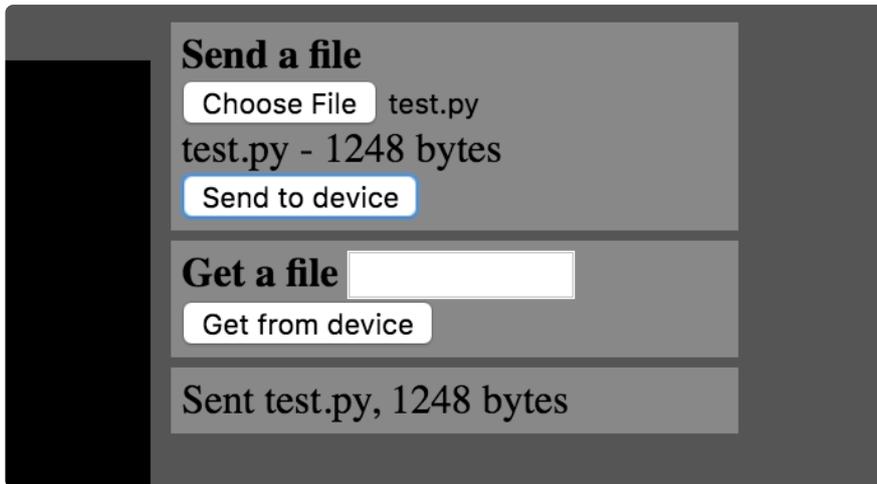
To use the file features of the WebREPL first connect to the board's WebREPL. Then notice the file operation block on the right of the WebREPL:



To upload a file click Choose File and select the file. Notice the page will show you the size of the file.



Click Send to Device to initiate the upload. Once finished the file operation status should show all the data was uploaded to the board.



To download a file from the filesystem just enter the name or path to the file in the Get a file box. Then click Get from device and the file should be downloaded through the browser to your computer.

