

Practical Python - Internet of Things

Clinton Roy

The Edge, State Library of Queensland

October 28, 2017

Outline

Introduction

Installation

Circuits

Resources

Introducing MicroPython

- ▶ Python for microcontroller boards
- ▶ Started as a Kickstarter
- ▶ All Open Source

Introducing the ESP8266

- ▶ 80MHZ processor
- ▶ lots of GPIO pins
- ▶ Digital interfaces
- ▶ One Analogue
- ▶ Wifi
- ▶ Cheap
- ▶ reasonably well documented (now)

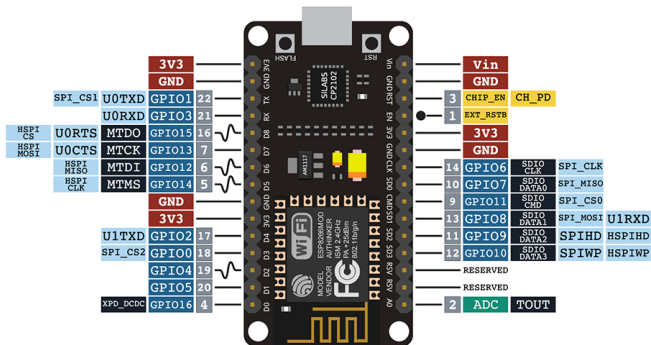
ESP8266 Development Board

ESP-12E DEVELOPMENT BOARD

PINOUT

NOTES:

- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.



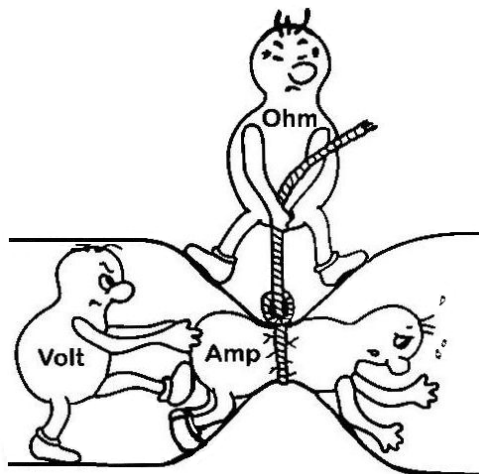
| | |
|---------|-----------------|
| POWER | SP. FUNCTION(S) |
| I/O | COMM. INTERFACE |
| ADC | PIN NUMBER |
| CONTROL | PWM |
| N/C | |

Pin Mapping

| Micropython | Board |
|-------------|-----------------------------|
| 0 | D3 |
| 2 | D4 (also Led1 but inverse)* |
| 4 | D2 |
| 5 | D1 |
| 9 | SD2 |
| 10 | SD3 |
| 12 | D6 |
| 13 | D7 |
| 14 | D5 |
| 15 | D8 |
| 16 | D0 (also Led2 but inverse)* |

Introduction to Electricity

- The water analogy



<http://www.electricalonline4u.com/2015/10/ohm-law-explanation-guide.html>

Installation

- ▶ Install Python, e.g. <http://anaconda.org>
- ▶ Install esptool

```
> pip install esptool
```

- ▶ Clear flash memory

```
> esptool.py --port /dev/ttyUSB0 erase_flash
```

- ▶ Download MicroPython
<http://github.com/micropython/micropython/releases>

- ▶ Flash board

```
> esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash  
--flash_size=detect 0 ~/Downloads/esp8266-$version.bin
```


Setup

- ▶ Connect to the board

```
> screen /dev/ttyUSB0 115200
```

- ▶ Hit enter!
- ▶ Work out WIFI name

```
> import network
```

```
> ap = network.WLAN(network.AP_IF)
```

```
> ap.config("ssid")
```

- ▶ Write this down!
- ▶ Turn on the Web Server

```
> import webrepl_setup
```

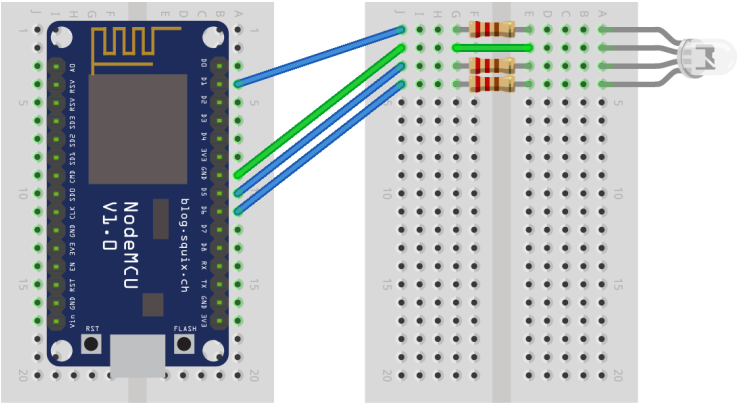
- ▶ Follow the prompts. Pick a simple password.
- ▶ Turn on Web Server
- ▶ Reboot...
- ▶ Connect to Board, password is micropython

Look Around

- ▶ Micropython implements a filesystem, just like a computer.

```
> import os  
> os.listdir()
```

RGB LED Layout



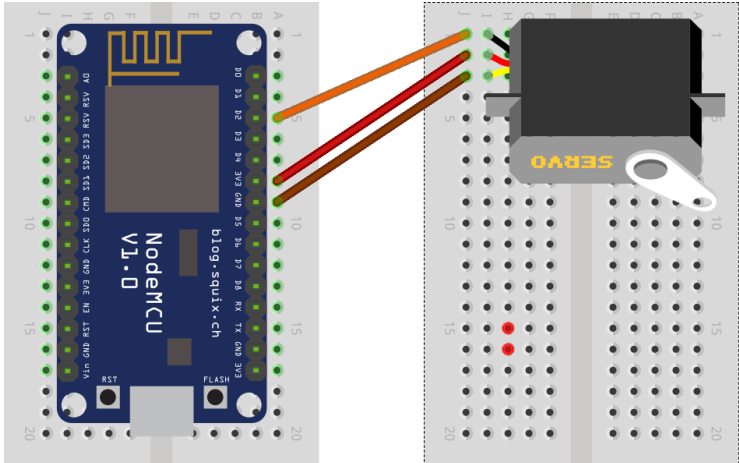
fritzing

RGB LED Code

```
> from machine import Pin, PWM
> r = PWM(Pin(4), freq=500, duty=0)
> g = PWM(Pin(12), freq=500, duty=0)
> b = PWM(Pin(14), freq=500, duty=0)
>
```

Servo Layout

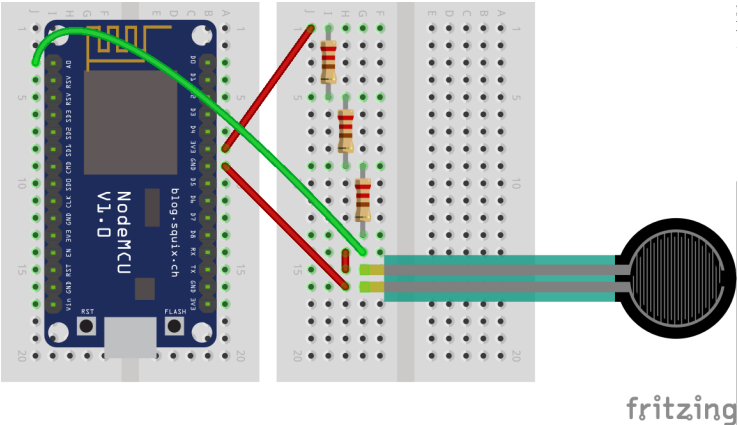
- ▶ Orange - PWM
- ▶ Red - +V
- ▶ Brown - Ground



Servo Code

```
> from machine import Pin, PWM  
> s = PWM(Pin(14), freq=50, duty=0)
```

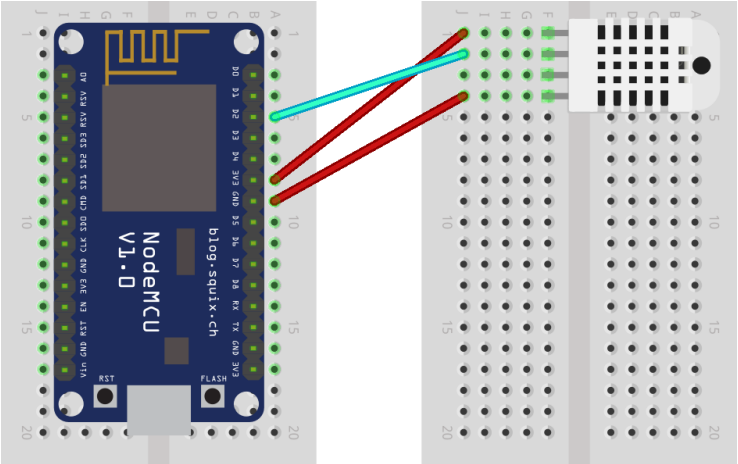
Force Sensor Layout



Force Sensor Code

```
> from machine import ADC  
> adc = ADC(0)  
> adc.read()
```


Temperature Sensor Layout



fritzing

Temperature Sensor Code

```
> import dht
> import machine
> d = dht.DHT11(machine.Pin(4))
>
> # Only once a second.
> d.measure()
> d.temperature()
> d.humidity()
```

Stand Alone Mode

- ▶ Special files:
 - ▶ `boot.py`, `main.py`

Resources

- ▶ These notes: Copyright © 2017 Clinton Roy
 - ▶ <https://github.com/clintonroy/slq2017python> © ⓘ
- ▶ <https://docs.micropython.org/en/latest/esp8266/esp8266/tutorial/>
- ▶ <http://fritzing.org/>