Practical Python - Internet of Things

Clinton Roy

The Edge, State Library of Queensland

October 28, 2017

Outline

Introduction

Installation

Circuits

Resources

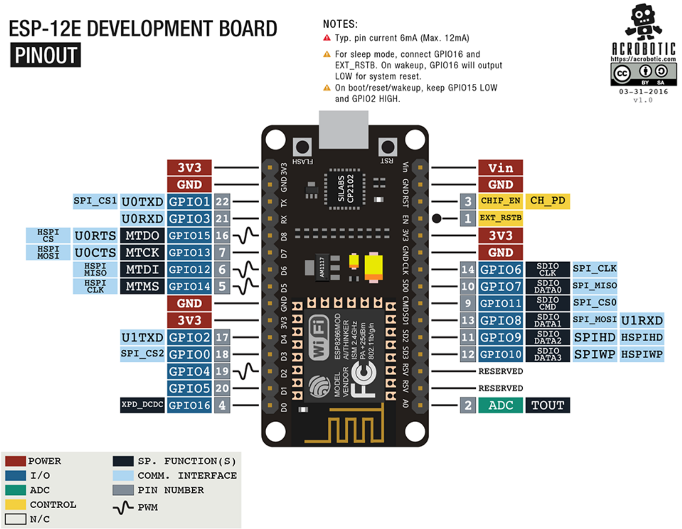
Introducing MicroPython

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

Introducing the ESP8266

- 80MHZ processor
- lots of GPIO (General Purpose Input/Output) pins
- Digital interfaces
- One Analogue
- Wifi
- Cheap
- reasonably well documented (now)

ESP8266 Development Board

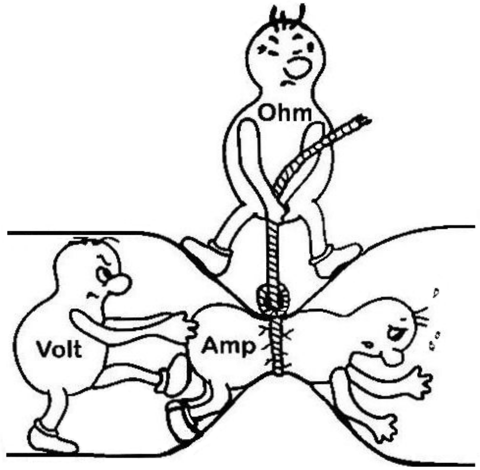

ESP-12E DEVELOPMENT BOARD
PINOUT
NOTES:
Typ. pin current 6mA (Max. 12mA)
For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.
ACROBOTIC
https://acrobotic.com
03-31-2016
v1.0
3V3
GND
SPI_CS1 U0TXD GPIO1 22
U0RXD GPIO3 21
HSPI CS U0RTS MTDO GPIO15 16
HSPI MOSI U0CTS MTCK GPIO13 7
HSPI MISO MTDI GPIO12 6
HSPI CLK MTMS GPIO14 5
GND
3V3
U1TXD GPIO2 17
SPI_CS2 GPIO0 18
GPIO4 19
GPIO5 20
XPD_DCDC GPIO16 4
Vin
GND
3 CHIP_EN CH_PD
1 EXT_RSTB
3V3
GND
14 GPIO6 SDIO CLK SPI_CLK
10 GPIO7 SDIO DATA0 SPI_MISO
9 GPIO11 SDIO CMD SPI_CS0
13 GPIO8 SDIO DATA1 SPI_MOSI U1RXD
11 GPIO9 SDIO DATA2 SPIHD HSPIHD
12 GPIO10 SDIO DATA3 SPIWP HSPIWP
RESERVED
RESERVED
2 ADC TOUT
FLASH
RST
SILABS CP2102
AM1117
ESP8266MOD
AI-THINKER
ISM 2.4GHz
PA +25dBm
802.11b/g/n
MODEL
VENDOR
Wi Fi
FC
3V3 GND TX RX D8 D7 D6 D5 GND 3V3 D4 D3 D2 D1 D0
Vin GND RST EN 3V3 GND CLK SD0 CMD SD1 SD2 SD3 RSV RSV A0
POWER
I/O
ADC
CONTROL
N/C
SP. FUNCTION(S)
COMM. INTERFACE
PIN NUMBER
PWM

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("essid")
```

- 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`

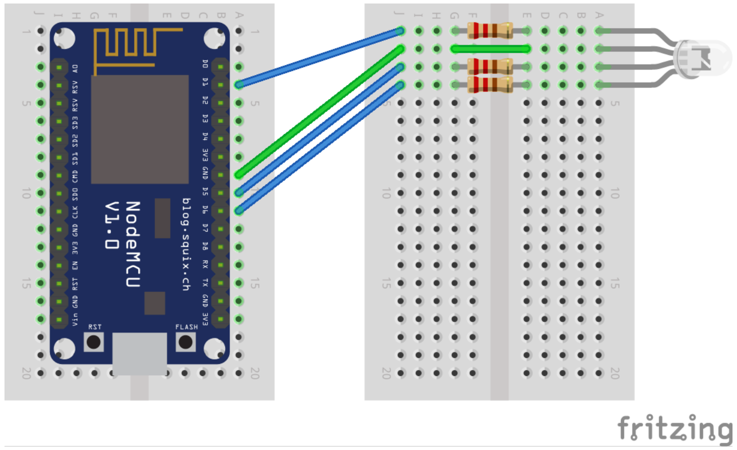
Look Around

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

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

RGB LED Layout

Red/Green/Blue Light Emitting Diode



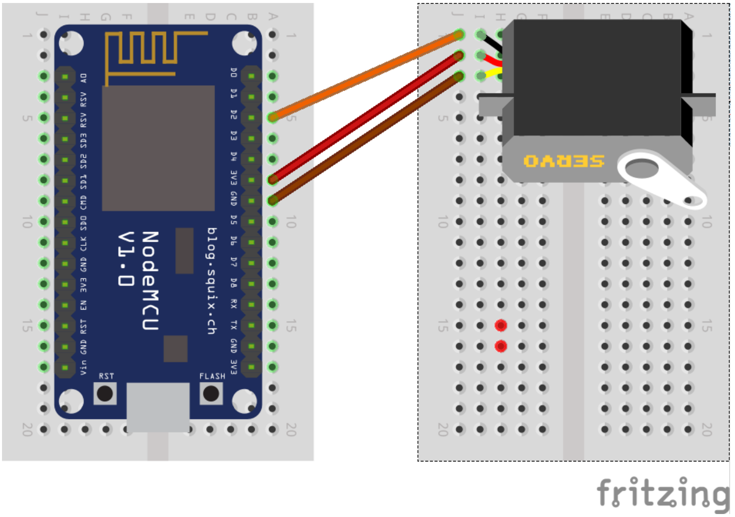
RGB LED Code

Pulse Width Modulation

```
> 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

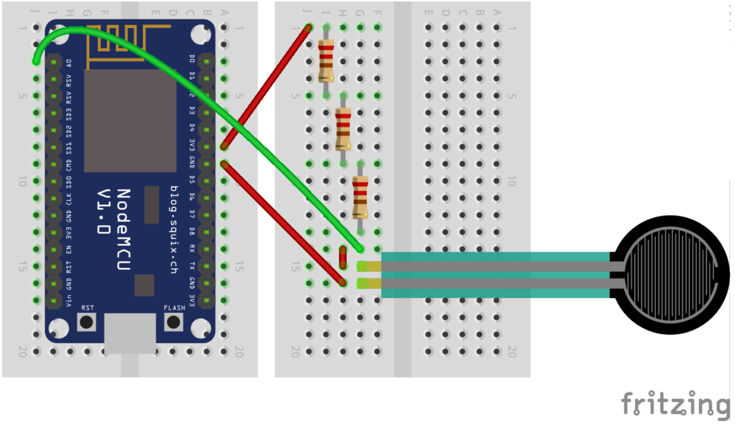
- Orange - PWM
- Red - +V
- Brown - Ground



Servo Code

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

Force Sensor Layout

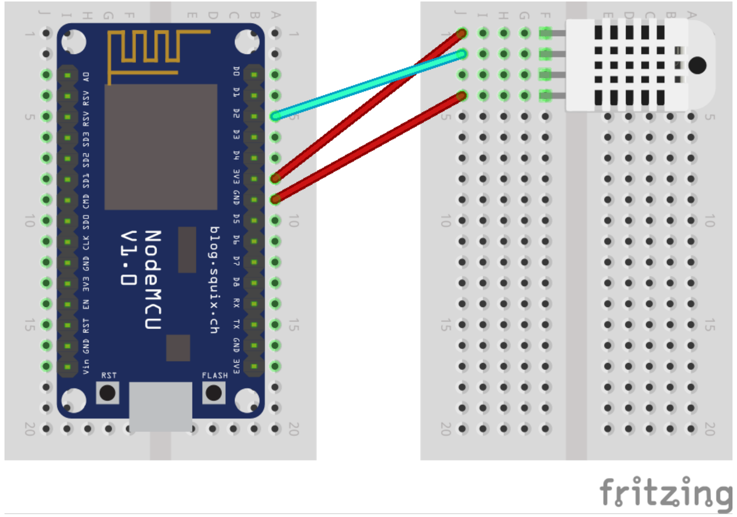


Force Sensor Code

- Analogue to Digital Converter

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

Temperature Sensor Layout



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
- Wifi

```
> import network
> n = network.WLAN(network.STA_IF)
> n.active(True)
> n.connect("slqwireless")
> n.isconnected()
> n.ifconfig()
```

Resources

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