Embedded System Design Techniques™

Designing IoT Sensor Nodes using the ESP8266

Session 2: Getting Started with the ESP8266

July 11th, 2017

Jacob Beningo

DesignNews CEC CONTINUING EDUCATION CENTER

Presented by:

Digi-Key ELECTRONICS ROHDE&SCHWARZ

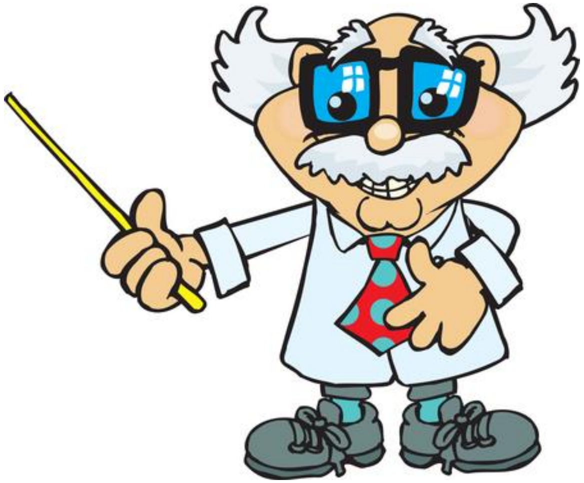
Course Overview

Topics:

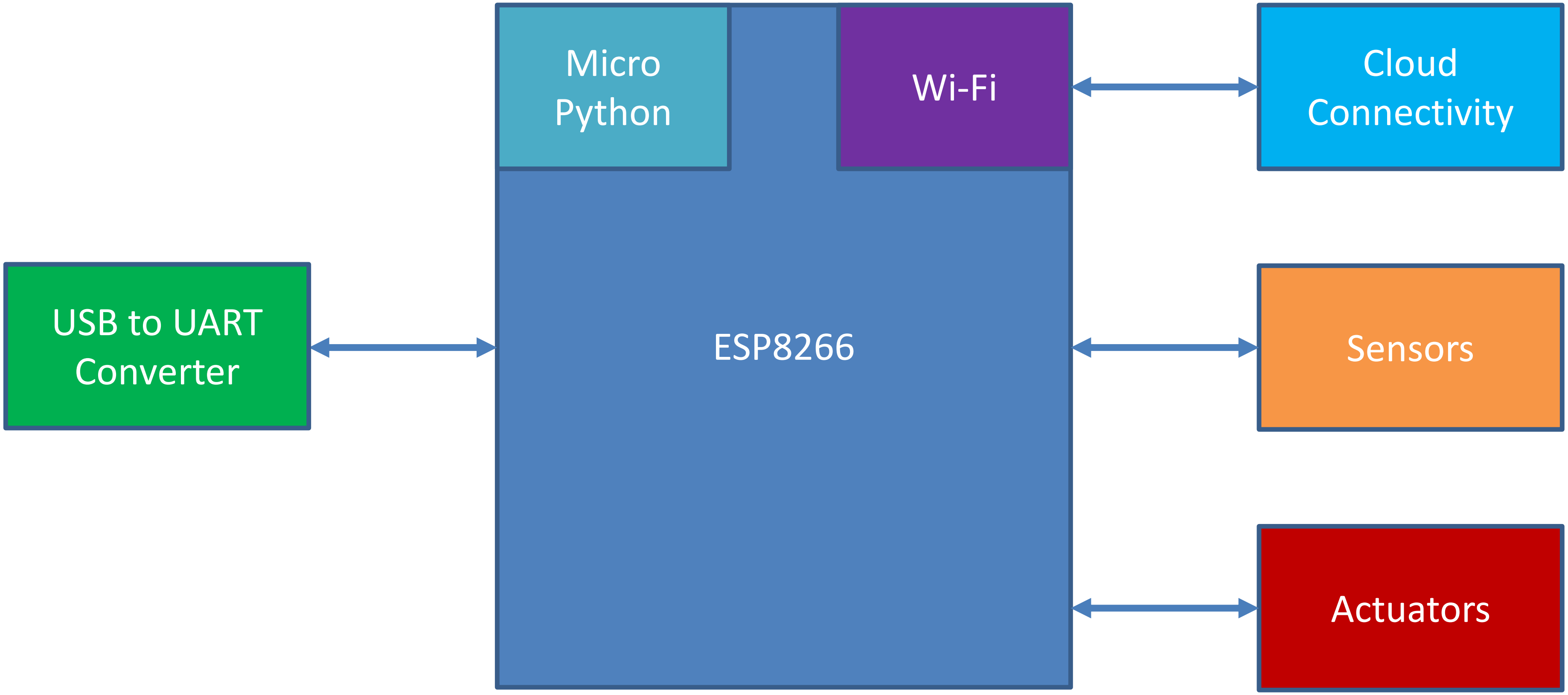
- The IoT Architecture
- **Getting Started with the ESP8266**
- Interfacing Sensors to the ESP8266
- Connecting the ESP8266 to the internet
- Device Management and the Automated Universe

Session Overview

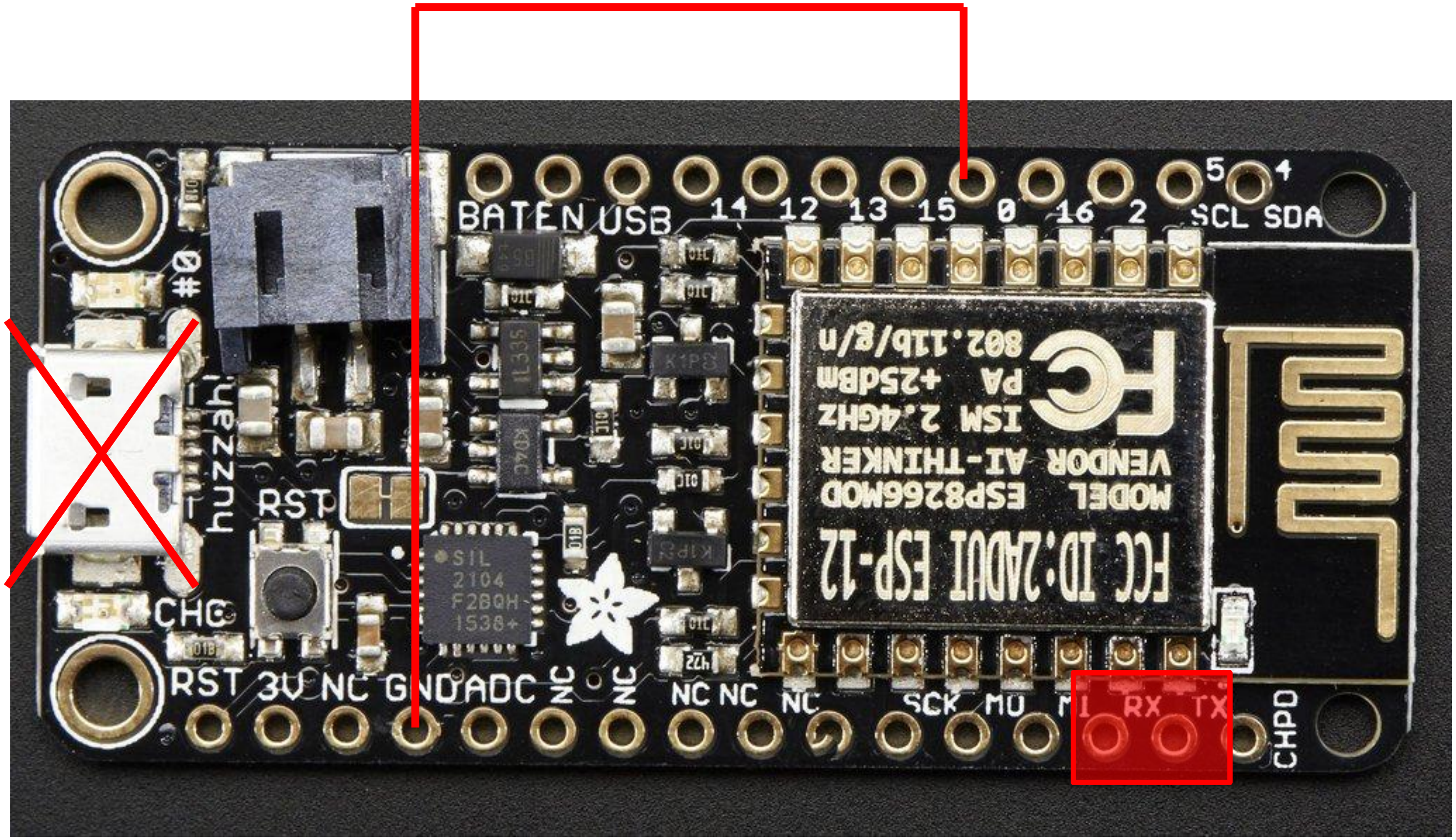
- Introduction
- The ESP8266
- Installing Micro Python
- Testing Micro Python
- Setting up a dev environment



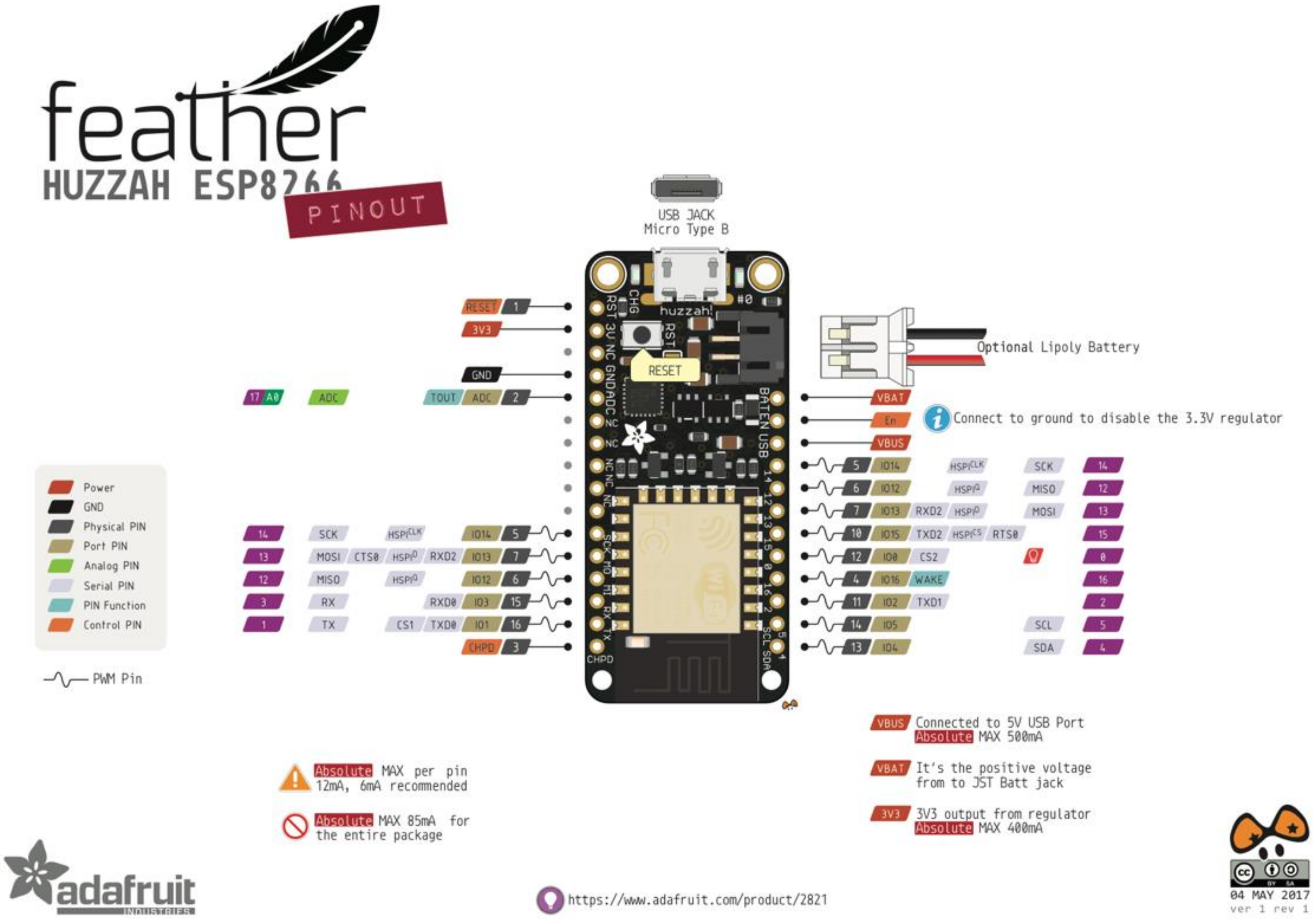
Our IoT Sensor Node



The Adafruit ESP8266 Featherboard

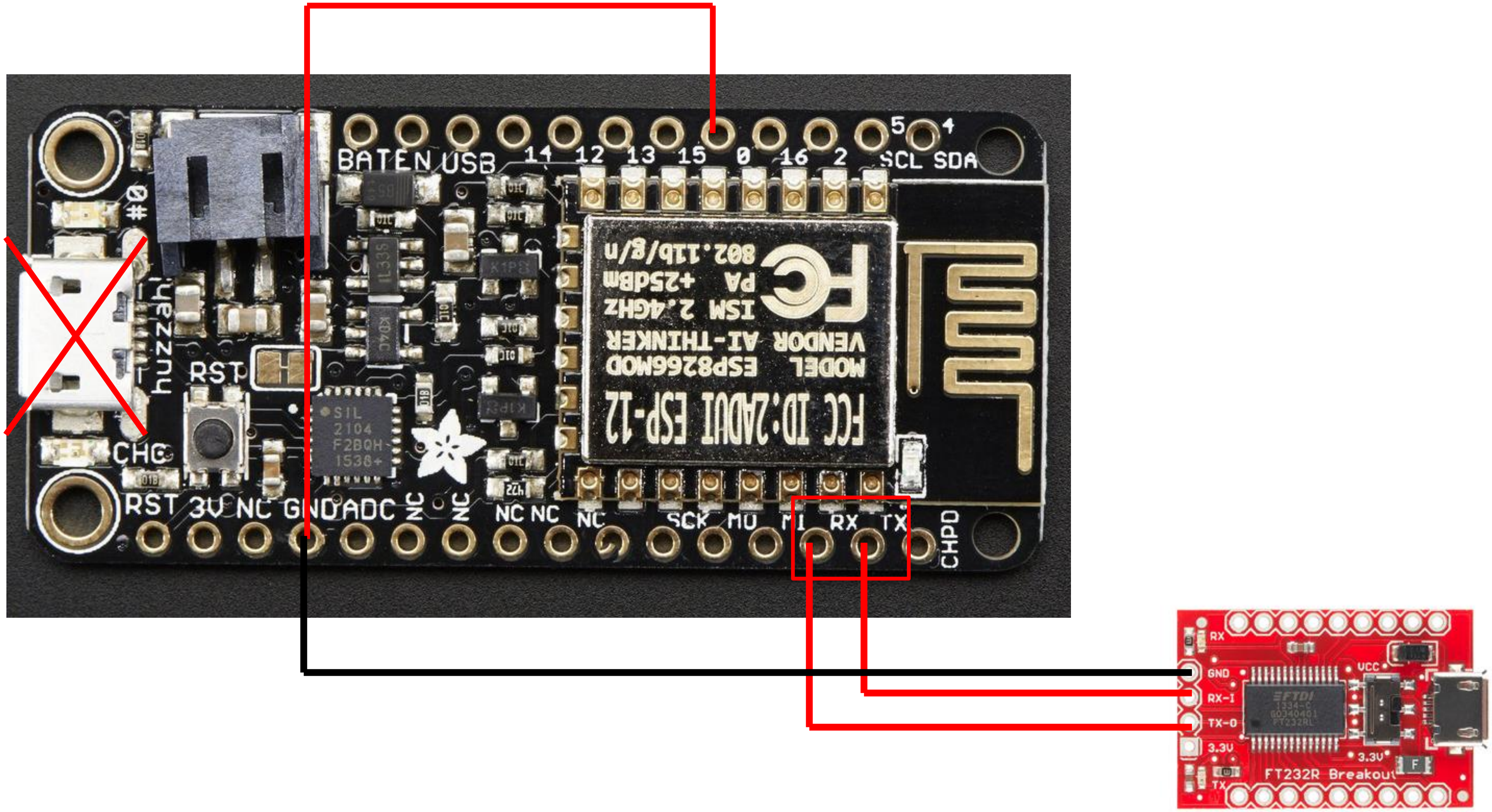


The Adafruit ESP8266 Featherboard



Presented by:

The Adafruit ESP8266 Featherboard



Presented by:

Installing Micro Python on the ESP8266

- 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.

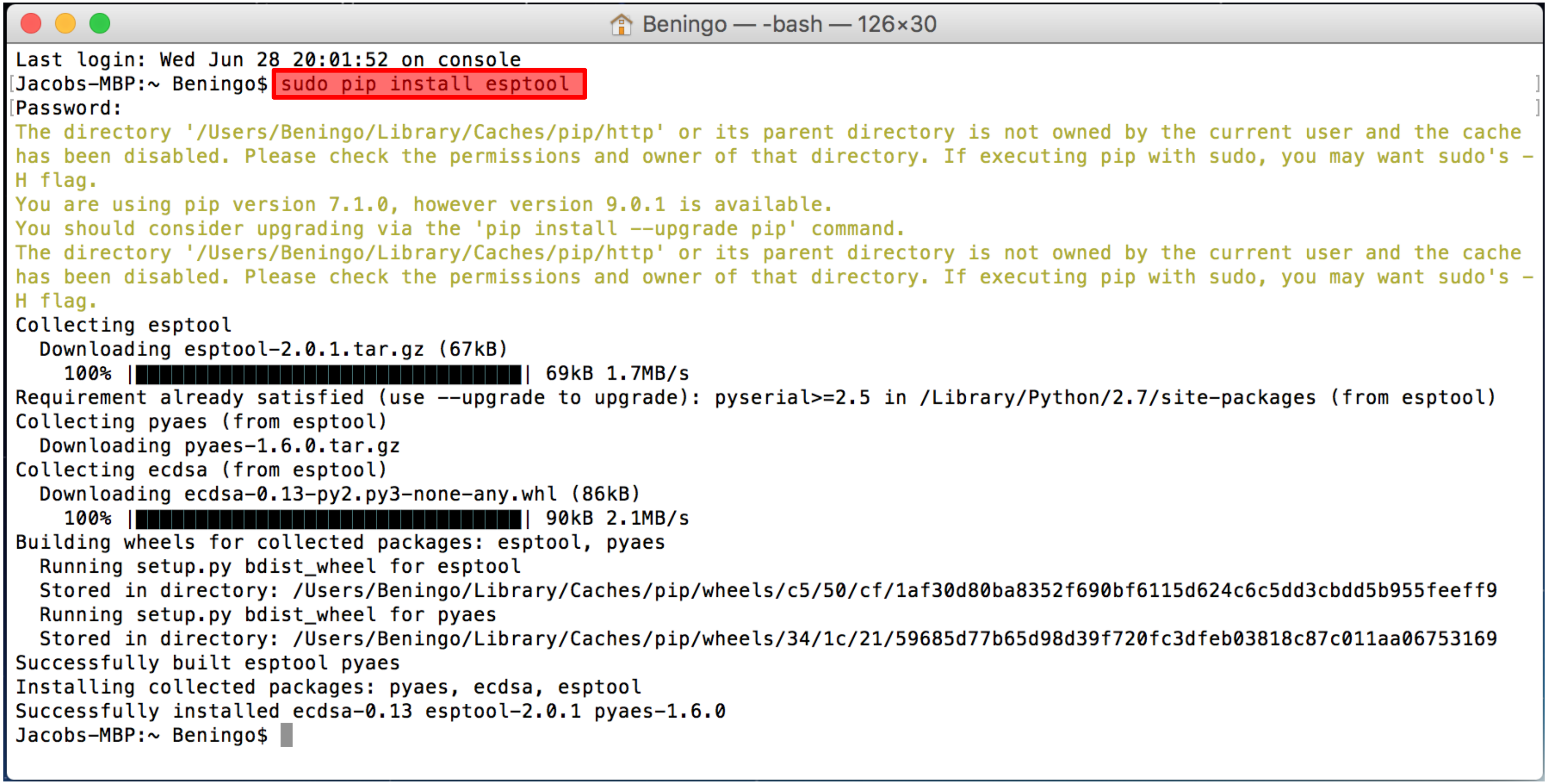
Run the following command in a terminal:

```
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
```

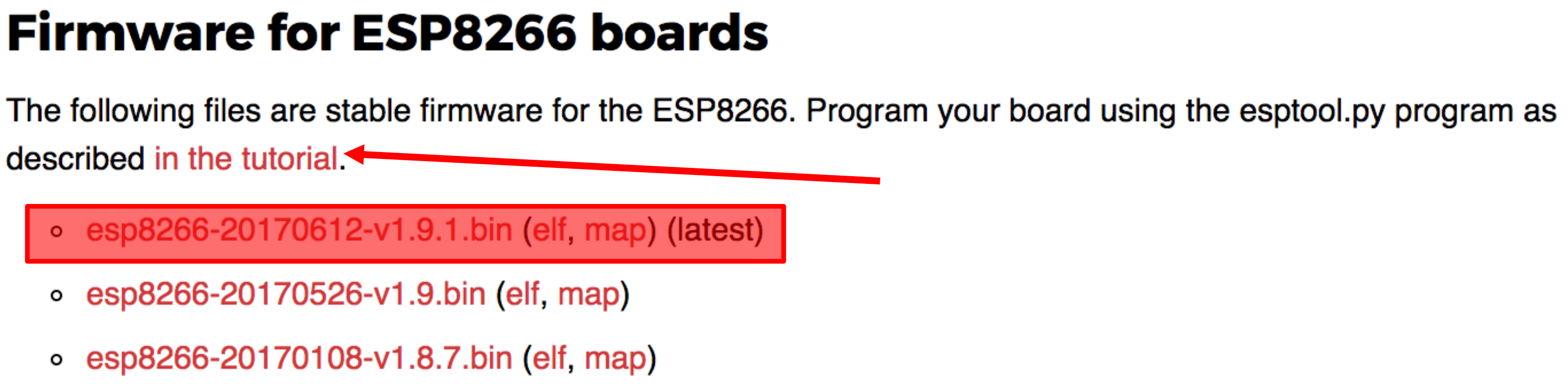
Installing Micro Python on the ESP8266



```
Last login: Wed Jun 28 20:01:52 on console
Jacobs-MBP:~ Beningo$ sudo pip install esptool
Password:
The directory '/Users/Beningo/Library/Caches/pip/http' or its parent directory is not owned by the current user and the cache
has been disabled. Please check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -
H flag.
You are using pip version 7.1.0, however version 9.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
The directory '/Users/Beningo/Library/Caches/pip/http' or its parent directory is not owned by the current user and the cache
has been disabled. Please check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -
H flag.
Collecting esptool
  Downloading esptool-2.0.1.tar.gz (67kB)
    100% |████████████████████████████████| 69kB 1.7MB/s
Requirement already satisfied (use --upgrade to upgrade): pyserial>=2.5 in /Library/Python/2.7/site-packages (from esptool)
Collecting pyaes (from esptool)
  Downloading pyaes-1.6.0.tar.gz
Collecting ecdsa (from esptool)
  Downloading ecdsa-0.13-py2.py3-none-any.whl (86kB)
    100% |████████████████████████████████| 90kB 2.1MB/s
Building wheels for collected packages: esptool, pyaes
  Running setup.py bdist_wheel for esptool
  Stored in directory: /Users/Beningo/Library/Caches/pip/wheels/c5/50/cf/1af30d80ba8352f690bf6115d624c6c5dd3cbdd5b955feeff9
  Running setup.py bdist_wheel for pyaes
  Stored in directory: /Users/Beningo/Library/Caches/pip/wheels/34/1c/21/59685d77b65d98d39f720fc3dfeb03818c87c011aa06753169
Successfully built esptool pyaes
Installing collected packages: pyaes, ecdsa, esptool
Successfully installed ecdsa-0.13 esptool-2.0.1 pyaes-1.6.0
Jacobs-MBP:~ Beningo$ 
```

Installing Micro Python on the ESP8266

- Download the latest Micro Python ESP8266 firmware file.
  - http://micropython.org/download#esp8266
  - Select the latest full build



Firmware for ESP8266 boards

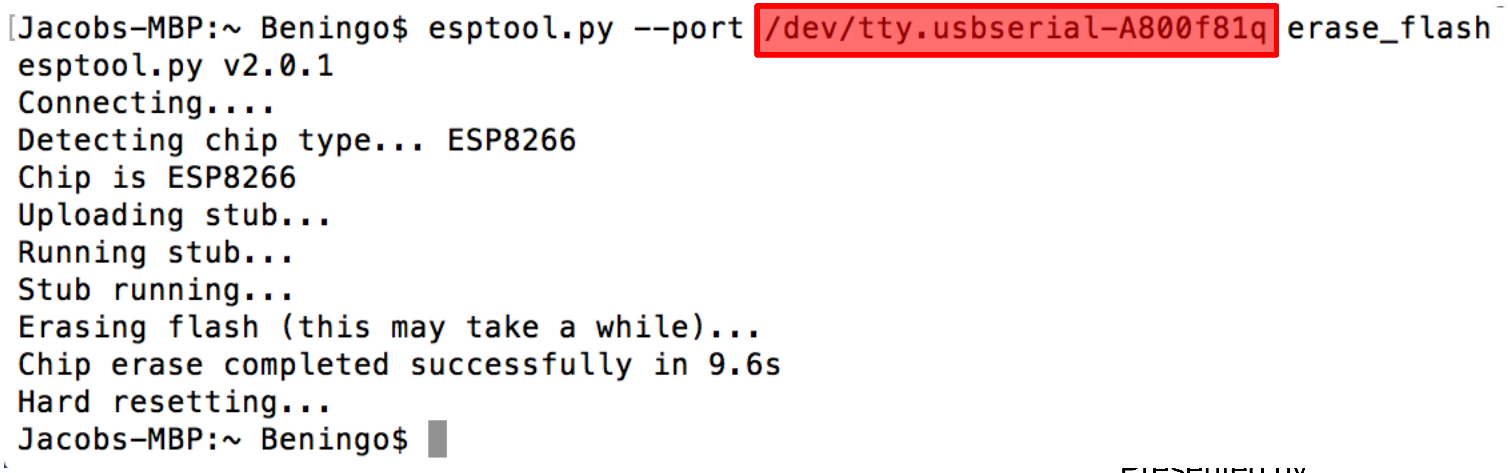
The following files are stable firmware for the ESP8266. Program your board using the esptool.py program as described in the tutorial.

- esp8266-20170612-v1.9.1.bin (elf, map) (latest)
- esp8266-20170526-v1.9.bin (elf, map)
- esp8266-20170108-v1.8.7.bin (elf, map)

Installing Micro Python on the ESP8266

- Put the ESP8266 into bootloader mode
  - For the HUZZAH ESP8266 breakout:
    - Hold GPIO0 down, then press and release RESET (while still holding GPIO0), and finally release GPIO0.
- Erase the on-board firmware using:

esptool.py --port /dev/ttyUSB0 erase_flash

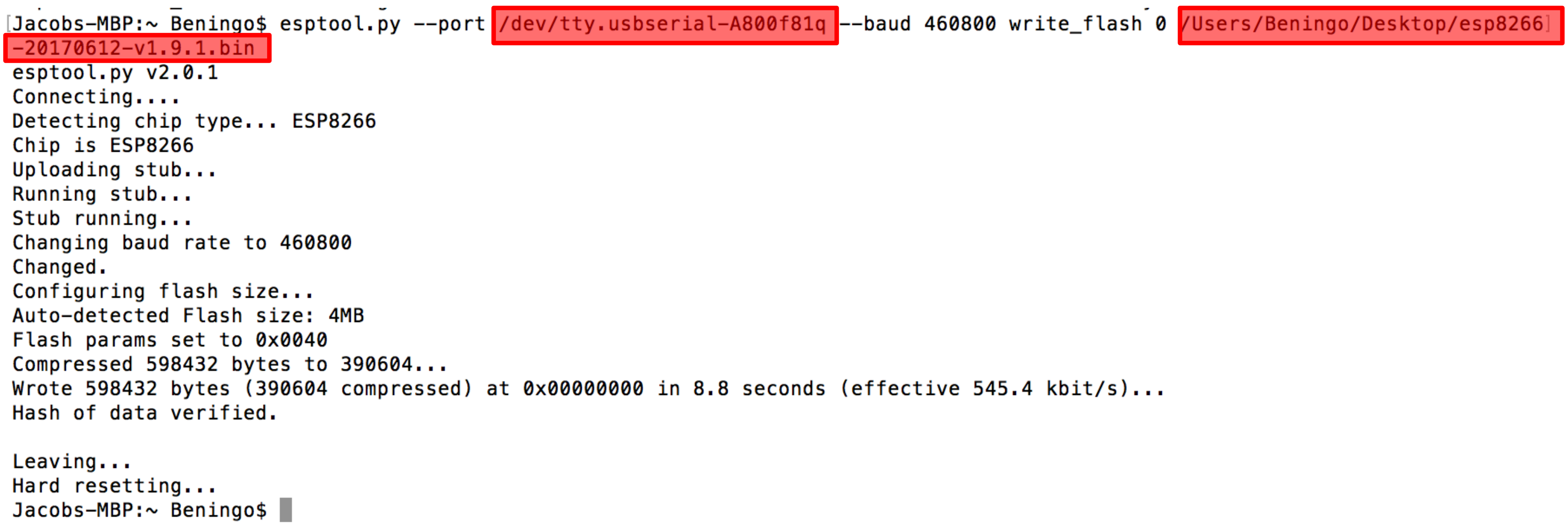


```
Jacobs-MBP:~ Beningo$ esptool.py --port /dev/tty.usbserial-A800f81q erase_flash
esptool.py v2.0.1
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 9.6s
Hard resetting...
Jacobs-MBP:~ Beningo$ 
```

Installing Micro Python on the ESP8266

- Downloaded the firmware file:

esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash --flash_size=detect 0 esp8266-20170612-v1.9.1.bin



```
[Jacobs-MBP:~ Beningo$ esptool.py --port /dev/tty.usbserial-A800f81q --baud 460800 write_flash 0 /Users/Beningo/Desktop/esp8266]
-20170612-v1.9.1.bin
esptool.py v2.0.1
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0040
Compressed 598432 bytes to 390604...
Wrote 598432 bytes (390604 compressed) at 0x00000000 in 8.8 seconds (effective 545.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting...
Jacobs-MBP:~ Beningo$
```

Getting a Micro Python REPL prompt

REPL stands for **R**ead **E**valuate **P**rint **L**oop.

There are two ways to access the REPL: either via a wired connection through the UART serial port, or via WiFi.

REPL over the serial port:

- Available on UART0:
  - GPIO1 for TX
  - GPIO3 for RX
  - Baud rate is 115200.

Testing Micro Python

Using the REPL:

Try typing the following at the prompt:

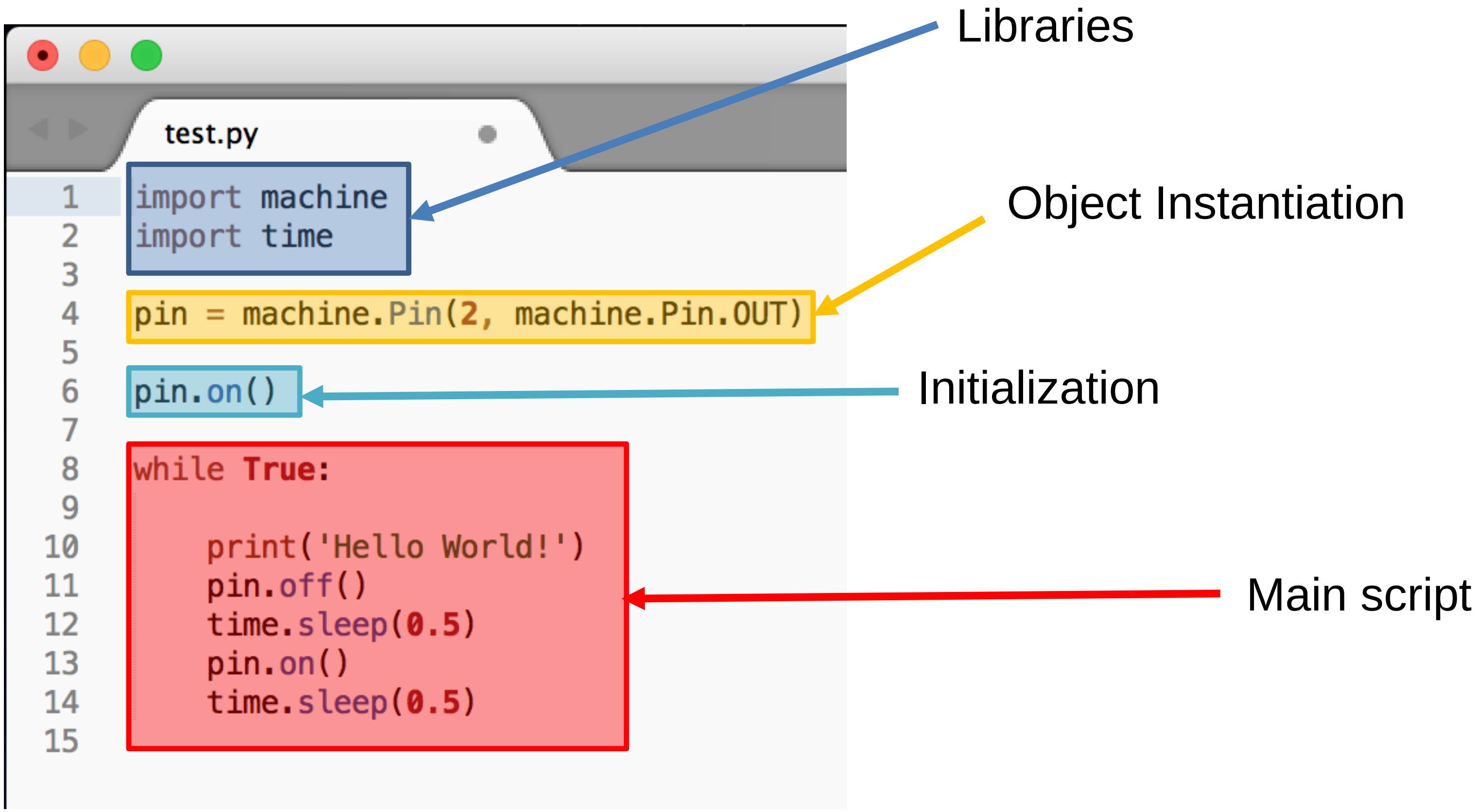
```
>>> print(’Hello World!')
Hello World!

>>> 1 + 2
3
```

Installing Ampy

- sudo pip install adafruit-ampy
- sudo pip3 install adafruit-ampy –upgrade
- Disable debug info
  - import esp
  - esp.osdebug(None)

Creating a Script



```python
import machine
import time

pin = machine.Pin(2, machine.Pin.OUT)

pin.on()

while True:

    print('Hello World!')
    pin.off()
    time.sleep(0.5)
    pin.on()
    time.sleep(0.5)
```

Libraries

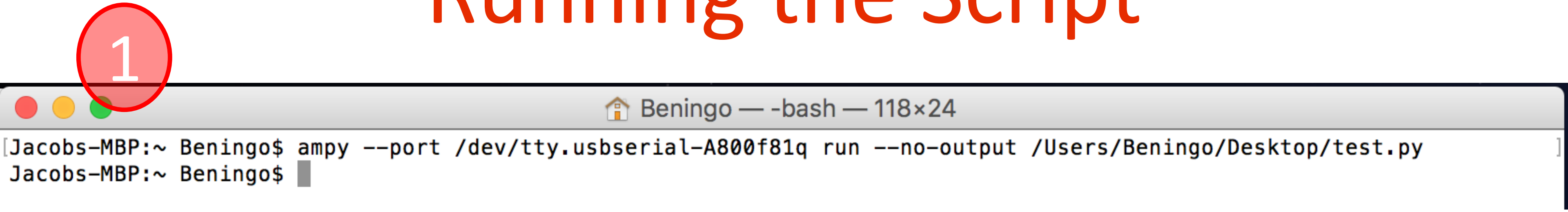
Object Instantiation

Initialization

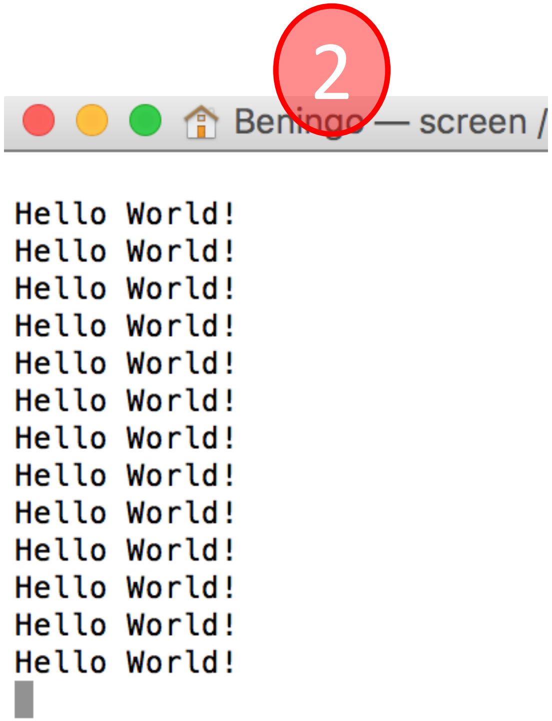
Main script

Presented by:

Running the Script



```
Jacobs-MBP:~ Beningo$ ampy --port /dev/tty.usbserial-A800f81q run --no-output /Users/Beningo/Desktop/test.py
Jacobs-MBP:~ Beningo$ 
```

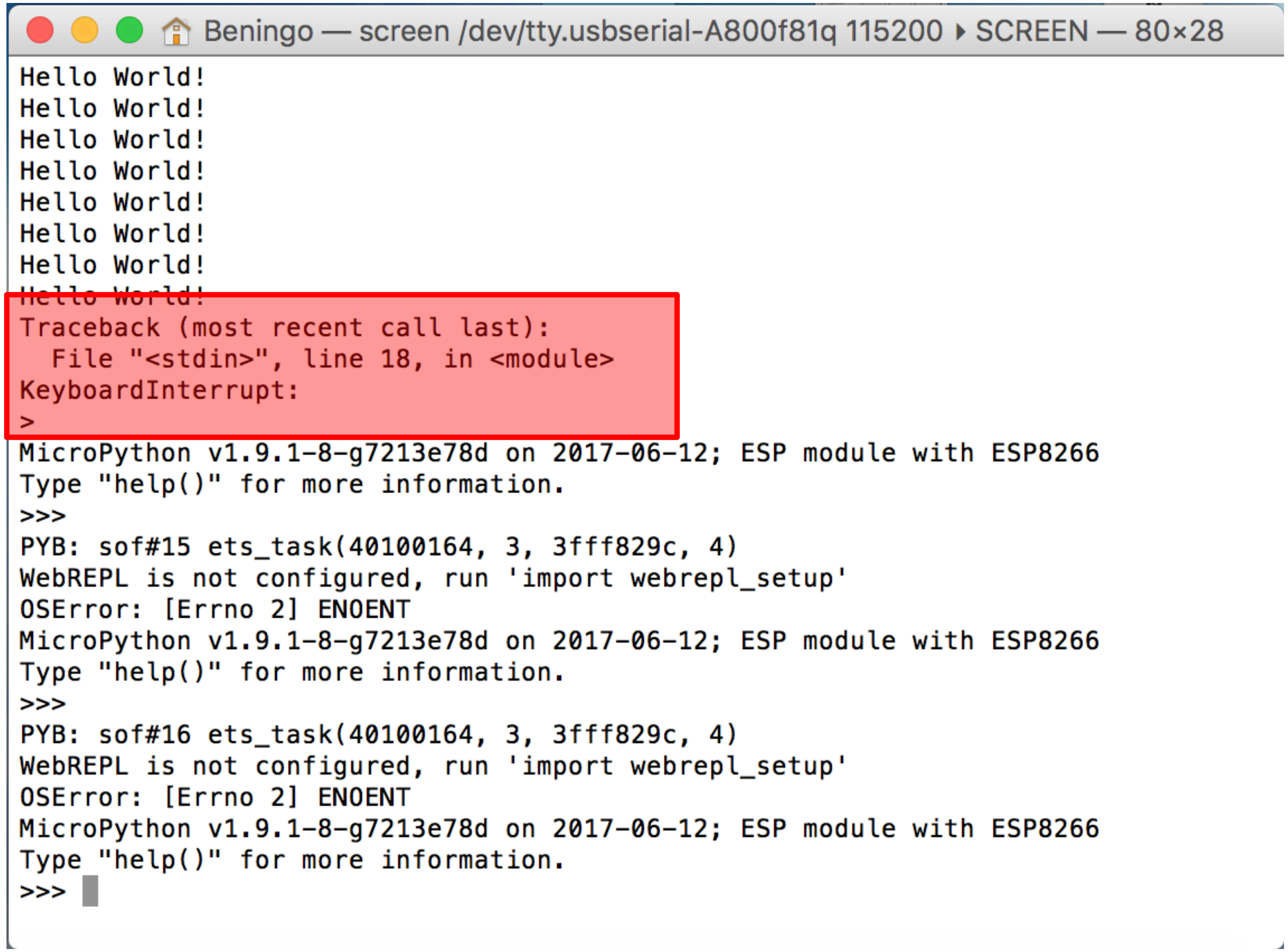


```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

Two Terminals each which need to access the same communication port.

1. Programs the target
2. Runs the REPL

Exiting the Application


Beningo — screen /dev/tty.usbserial-A800f81q 115200 ▸ SCREEN — 80×28

```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Traceback (most recent call last):
  File "<stdin>", line 18, in <module>
KeyboardInterrupt:
>
MicroPython v1.9.1-8-g7213e78d on 2017-06-12; ESP module with ESP8266
Type "help()" for more information.
>>>
PYB: sof#15 ets_task(40100164, 3, 3fff829c, 4)
WebREPL is not configured, run 'import webrepl_setup'
OSError: [Errno 2] ENOENT
MicroPython v1.9.1-8-g7213e78d on 2017-06-12; ESP module with ESP8266
Type "help()" for more information.
>>>
PYB: sof#16 ets_task(40100164, 3, 3fff829c, 4)
WebREPL is not configured, run 'import webrepl_setup'
OSError: [Errno 2] ENOENT
MicroPython v1.9.1-8-g7213e78d on 2017-06-12; ESP module with ESP8266
Type "help()" for more information.
>>>
```

Additional Resources

- Download Course Material for
  - Python Doxygen Templates
  - Example source code
  - Blog
  - YouTube Videos
- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

From www.beningo.com under

- Blog > CEC – Designing IoT Sensor Nodes using the ESP8266

Presented by:

The Lecturer – Jacob Beningo

Jacob Beningo

Principal Consultant

BENINGO EMBEDDED GROUP

Social Media / Contact

E : jacob@beningo.com

T : 810-844-1522

: Jacob_Beningo

f : Beningo Engineering

in : JacobBeningo

EDN : Embedded Basics

ARM Connected Community

Consulting

- Advising
- Coaching
- Content
- Consulting
- Training

www.beningo.com