ESP8266 & IoT - Or how I made an IoT appliance for less than $15.00

by Aaron Grothe

Introduction

Why?

ESP8266s are available for as cheap as $1.00/chip in quantity, for very large quantities :-) for a single chip you can find one for a couple of dollars.

There are a lot of different ways to program the chips.

For very simple tasks (such as remote sensors) you can power them off a couple of lithium-ion batteries for a couple of years.

Introduction (Cont'd)

How does it Compare to the Raspberry PI?

The ESP8266 is a lot less powerful than the Raspberry PI. Where most people run Linux on their PIs. the ESP8266 tends to run very small OSes.

The ESP8266 uses less electricity than a Raspberry PI and may work in environments where the PI is overkill such as remote sensors.

Introduction (Cont'd)

I'm still experimenting/learning with the ESP8266. If anybody has questions feel free to ask and I'll do my best but I could be wrong.

History of the ESP8266

- Arduino Happens
- People want to wifi enable them
- Espressif releases an internet enabled light bulb
- Espressif also releases the chip as an add-on for arduino
- People figure out that the ESP8266 also HAS an Ardunio as part of it
- People read the datasheets and reverse engineer the documentation and SDKs
- Docs are google translated from Chinese to English via Google Translate and then cleaned up
- People start buying and using a lot of these

A Quick Look at Some ESP8226s

Lets hit a few sites and check out some ESP8266 options

Amazon
AliExpress
Banggood.com
DX.com

Prices vary depending on options/quantities and so on

Major Things to Look For

Built in Serial to USB adapter
NodeMCU
Number of pins
Ram (512k or 1024k)

Pretty much have to read the datasheets to get all the info

Available SDKs/OSes

Arduino C++ firmware
NodeMCU - Lua supporting firmware
Micropython - Python supporting firmware
Esprunio - Node.js like

NodeMCU and Micropython run their languages as the OS

Tons of others

Basic programming language
Mongoose OS
FreeRTOS

Micropython

- Micropython is a subset of Python 3
- Some stuff is a subset of regular python ujson vs json, urequest vs request
- Missing some things like the string class doesn't have ljust and so on

There are places Micropython isn't the best choice. E.g. performance intensive tasks. Also for tasks requiring as much available memory as possible.

My ESP8266

This is my ESP8266. There are many others like it, but this one is mine. -- Paraphrased from Full Metal Jacket

I've got a handful of ESP8266s and just got an ESP32

It is a simple ESP8266 with a built in 0.91" OLED screen. Pins are already hooked up and available

Is from heltec.cn - Time to take a look at their website

At least English is an option :-)

Now we'll take a look at the datasheet for it.

Disclaimer

We're going to be running some commands to flash an ESP8266 please be careful with this.

Done incorrectly these can causes issues on your system.

Time to Install Micropython

Steps

Plug it in
Erase the memory on the device
Install Micropython
Connect to the device
Turn on webrepl
Hook up the network
Hit the webrepl site
Run some tests

Erase the Memory on the Device

We'll use the esptool for this. Esptool is available via Python PIP

python pip install esptool

esptool.py --port /dev/ttyUSB0 erase_flash

Install Micropython

We've already grabbed the image from micropython.org

You can build it from scratch if you'd like
They have stable and nightly builds (recommend the stable)

```
# esptool.py --port /dev/ttyUSB0 --baud 460800
write_flash --flash_size=detect 0
esp8266-20180511-v1.9.4.bin
```

Connect to the Device

Need to start a terminal to connect to the ESP8266

We'll use picocomm for this

```
# picocom /dev/ttyUSB0 -b 115200
```

Hit return and hope we get the ">>>" :-)

Bootstrap Networking

By default it'll come up with 192.168.4.1 once we turn on WebRepl want to change that so we'll do the following.

Remember we don't have an O/S just a REPL (Read Evaluate Print Loop).

So we'll have to do thing the harder way.

Will put a tiny bit of networking code in the main.py of the machine.

Bootstrap Networking (Cont'd)

```
>>> a = """import network

station = network.WLAN(network.STA_IF)
station.active(True)
station.connect ("ssid", "psk")
print (station.ifconfig())"""

>>> file = open ("main.py", "w")
>>> file.write(a)
>>> file.close()
```

And reset the device and we should get the output

Turn on WebREPL

WebREPL is a web-enabled version of the Read Evaluate Print Loop that is run via the terminal.

WebREPL allows you to upload files to the device. We'll need this later.

>>> import webrepl_setup

Set a password for the device (optional but highly recommended)

Does a soft reboot of system

WebREPL (continued)

The IP address of the device should hang around for the time being. So we should be able to flip over to the battery pack, pass it around and keep working on it remotely :-)

WebREPL (cont'd)

Hit the WebREPL site

http://micropython.org/webrepl/

You can also download the webrepl locally as well. For this one we'll just use the web version. I also have the local version in case we have "issues".

Try some Basics

```
>>> print (3 + 4)
>>> print (“hello world”)
```

See how it compares to regular python

```
>>> import json   # fails
>>> import ujson # succeeds
```

Some of this are only empty libs right now

>>> import configparser # will work but you won’t have any of the functions of it as it is all stubbed out

Turn the OLED off and On

There are a lot of OLEDs available that work with the ESP8266. We'll use the one built into this device

```
>>> import machine
>>> import ssd1306

>>> i2c = machine.I2C(-1, machine.Pin(5), machine.Pin(4))
>>> oled = ssd1306.SSD1306_I2C(128, 32, i2c, 0x3c)
>>> oled.fill(0) # 0 is black, 1 is white
>>> oled.text("Hello World", 0, 0)
>>> oled.show()
```

Remember you don't have a real OS

Import os library

>>> import os

How do you get a listing of files on the system

>>> os.list_dirs()

To put a file on the filesystem you have to write a python program or upload via webrepl

Remember you don't have a real OS

How to cat a file

```
>>> with open ('boot.py') as f:
... print (f.read())
```

How to reset machine

```
>>> import machine
>>> machine.reset()
```

Some libs like shutil aren't available

Where are We At?

The system is running Micropython. We've enabled the WebREPL. Time to show our project.

Project

Air Quality Display

Airnow.gov has sensors around the US and makes the data available via their website and also via a webapi

Air Quality Display - Project

Project is Pretty Simple

3 Screens displayed on the tiny screen

#1. Location
#2. Numerical Values for Ozone (03), Particles (2.5 microns) and Particles (10 microns)
#3. Quality of Air (Good, Moderate, Poor, Others)

Let's take a look at the Script

Script is less than 100 lines

Loop
- Get data
- Display Screen #1
- Sleep 2 seconds
- Display Screen #2
- Sleep 2 seconds
- Display Screen #3
- Sleep 2 seconds
- If it has been an hour get the data again

Let's take a look at the Script

Need to update the initialization section

Set PSK, SSID for network access
Set API key to your registered key

Then need to transfer to the ESP8266 with the name main.py

Reboot the device and viola you should have a working Air Quality display

Optimizations (Left to Reader)

Some ideas how to optimize the script

Turn off wifi when not using it. Will reduce the power quite a bit.

Put a webserver enable the app so you can set the parameters and then restart the device.

Move the parameters out of the .py and into another file so you don't have to update for every deployment.

Next Gen AQI???

If I were to decide do a next gen version of this project. I'd probably do along the following

TTGO T5 - ESP32 / 1.54" ePaper display with a speaker

Speaker = Audio alerts possible, ePaper always on lower power draw

Cost at Amazon about $35.00 also available on Aliexpress and so on.

Hit the TTGO sites and you'll even see photos of stores with these as the price labels on the counters.

How to I get started with ESP8266

Squix/ThingPulse has a really nice little kit called the ESP8266 weather station. It sells for about $25.00 on Amazon

You can also do a search for ESP8266 Weather Stations on Aliexpress and others and find similar kits for $12.00

You can also get an ESP8266 with a small built-in OLED for less than $10.00

If you don't need an OLED you can get one for even cheaper

How to I get started with ESP8266

If you want to consider working with AC outlets consider the Electrodragon site. They have ESP8266 with a couple of 220V relays for as cheap as $10.00.

Of course I encourage caution on this as well :-)

Things like MQTT, IFTT and others give you the power to use these as a source for data that has some potential as well.

Other Projects

ESP8266 can use sensors that work with Arduino so there are a lot of projects. Moisture, Light, Temp sensors and so on.

The ESP8266 weather station is pretty cool. It also has a project to show airplanes overhead as well as a remote temp/moisture sensor project that is very cool.

Other things like IFTT support exists can be hooked up to things like firebase and so on. You've got urequests, ujson support you can do a lot of things.

Future of the ESP8266

Still going strong. New tools are being developed and the Arduino/IDE is continuing to improve.

Espressif created a successor to the ESP8266 called the ESP32. The ESP32 is available in single and dual core versions which run at 160 or 240 mhz. As opposed to 80/160 on the ESP8266. Also has dual-mode bluetooth so that opens up a lot of new possibilities as well.

Summary

The ESP8266 is a pretty darn neat little computer that can do a lot of things. It has a lot of development options and is continuing to get better tools and versions.

I have another project that uses the ESP8266 I'm working on. When I get it in a presentable state I'd be happy to share it with the world. It is closer to a save the world kind of thing.

Got to say I've had a lot of fun playing around with the ESP8266.

Q & A

Thanks for listening

Links

Airnow.gov - site where we get our info for the project

http://www.airnow.gov

Micropython - home of the Micropython language

http://www.micropython.org

Squix/ThingPulse - Creators of the ESP8266 Weather Station

https://thingpulse.com

Links

Aliexpress.com

http://www.aliexpress.com

Banggood.com

http://www.banggood.com

DealExtreme.com

http://www.dealextreme.com

Links

Amazon.com

http://www.amazon.com

Heltec.cn - home for my ESP8266 (English site)

http://www.heltec.cn/?lang=en

Talk Python to Me - Has 2 Podcasts on Micropython

http://www.talkpython.fm

Links

My Github for Air Quality Indicator project

https://github.com/ajgrothe/aqi