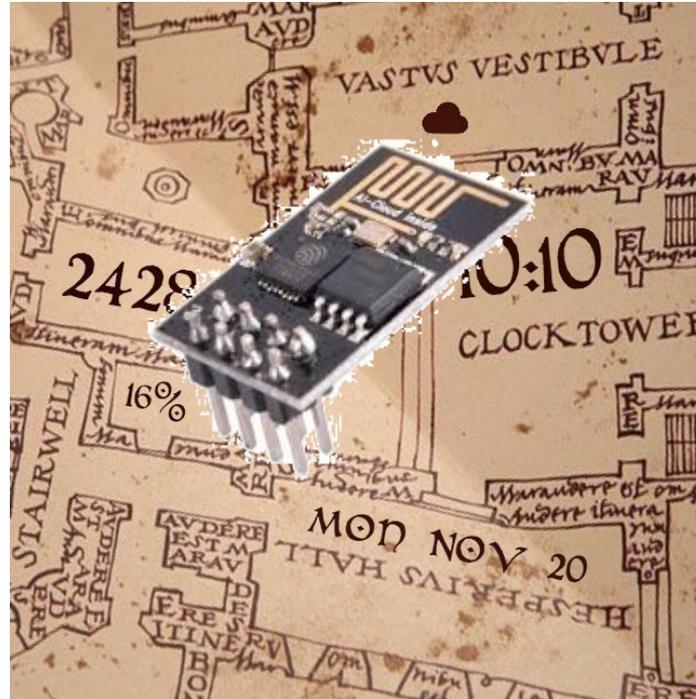


# ESP8266 Node Red Marauder's Map

Scott Wegner

<https://sites.google.com/isd786.org/swegner/home>

scott.wegner@isd786.org



# Post Brainstorm 2019 Conference Note!

At the conference I didn't have a website up for the info - but I will be adding more content (code - examples - etc) onto a site now.

<https://sites.google.com/isd786.org/swegner/home>

Please pardon the delay!

# But First!... a quick update on last year's project!

## The Raspberry Pi Bell System!



THIS

Replaced This  
EVIL thing



I'm mentioning this because it falls into a category similar to where the Arduino sits... making things with single board computers, IOT devices etc....



**Editing Schedule A**

```
#Schedule-a-Regular-Schedule.txt
10 8 * * 1-5 sudo -u pi /home/pi/default-bell.sh #GET READY bell! (start)
15 8 * * 1-5 sudo -u pi /home/pi/default-bell.sh #1st hr bell (start)
3 9 * * 1-5 sudo -u pi /home/pi/default-bell.sh #1st hour bell (end)
7 9 * * 1-5 sudo -u pi /home/pi/default-bell.sh # advisor time (start)
22 9 * * 1-5 sudo -u pi /home/pi/default-bell.sh # advisor (end)
26 9 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 2nd hr bell (start)
14 10 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 2nd hour bell (end)
18 10 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 3rd hr bell (start)
6 11 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 3rd hour bell (end)
10 11 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 4th hr bell (start)
58 11 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 4th hour bell and lunch start (end)
24 12 * * 1-5 sudo -u pi /home/pi/default-bell.sh # lunch (end)
25 12 * * 1-5 sudo -u pi /home/pi/default-bell.sh # lunch (end)
29 12 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 5th hr bell (start)
17 13 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 5th hr bell (end)
21 13 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 6th hr bell (start)
9 14 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 6th hr bell (end)
13 14 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 7th hr bell (start)
1 15 * * 1-5 sudo -u pi /home/pi/default-bell.sh # 7th hr bell (end)
```

Save

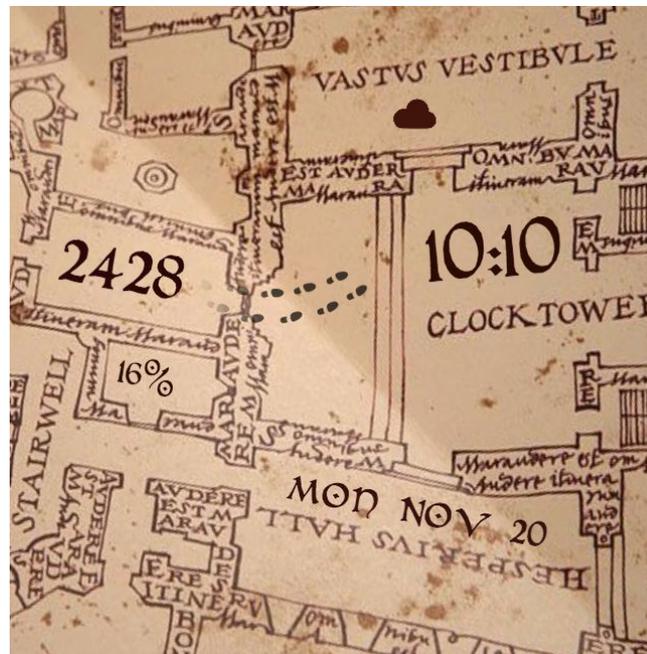
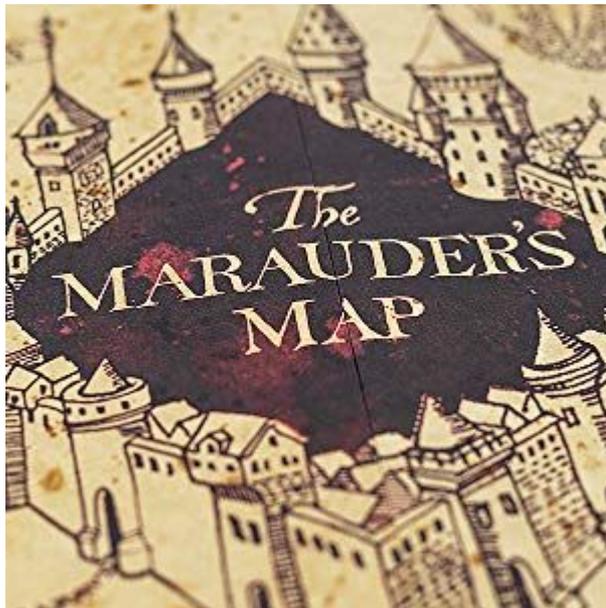
[Back to Main Page](#)

# So what is our agenda here?

- 1 - show you what the Marauder's Map is - what it does.
- 2 - Look at specifically what is an ESP8266.
- 3 - What are the pieces and how do they interact - broad overview.
- 4 - Programming an ESP8266
- 5 - Look at Node Red.
- 6 - Demo it all together.

**I finally managed to get rid of that nasty electrical charge I've been carrying. .... I'm ex-static!**

# What is the Marauder's Map?





## 2PCS ESP8266

ESP-01 Serial WIFI  
Wireless  
Transceiver Module



2PCS ESP8266



\$ Have one to sell?

[Sell now](#)

## 2X ESP8266 Serial WIFI Wireless Transceiver Module Send Receive AP+STA ESP-01 US

★★★★★ 10 product ratings

Condition: **New**

Quantity:

1

More than 10 available

98 sold / [See feedback](#)

Price: **US \$5.95**

[Buy It Now](#)

[Add to cart](#)

[Add to watch list](#)

100% buyer satisfaction

98 sold

Free shipping

Shipping: **FREE** Standard Shipping | [See details](#)

Item location: Duluth, Georgia, United States

Ships to: United States

Delivery: Estimated on or before **Mon. Mar. 11** to 60002 ?

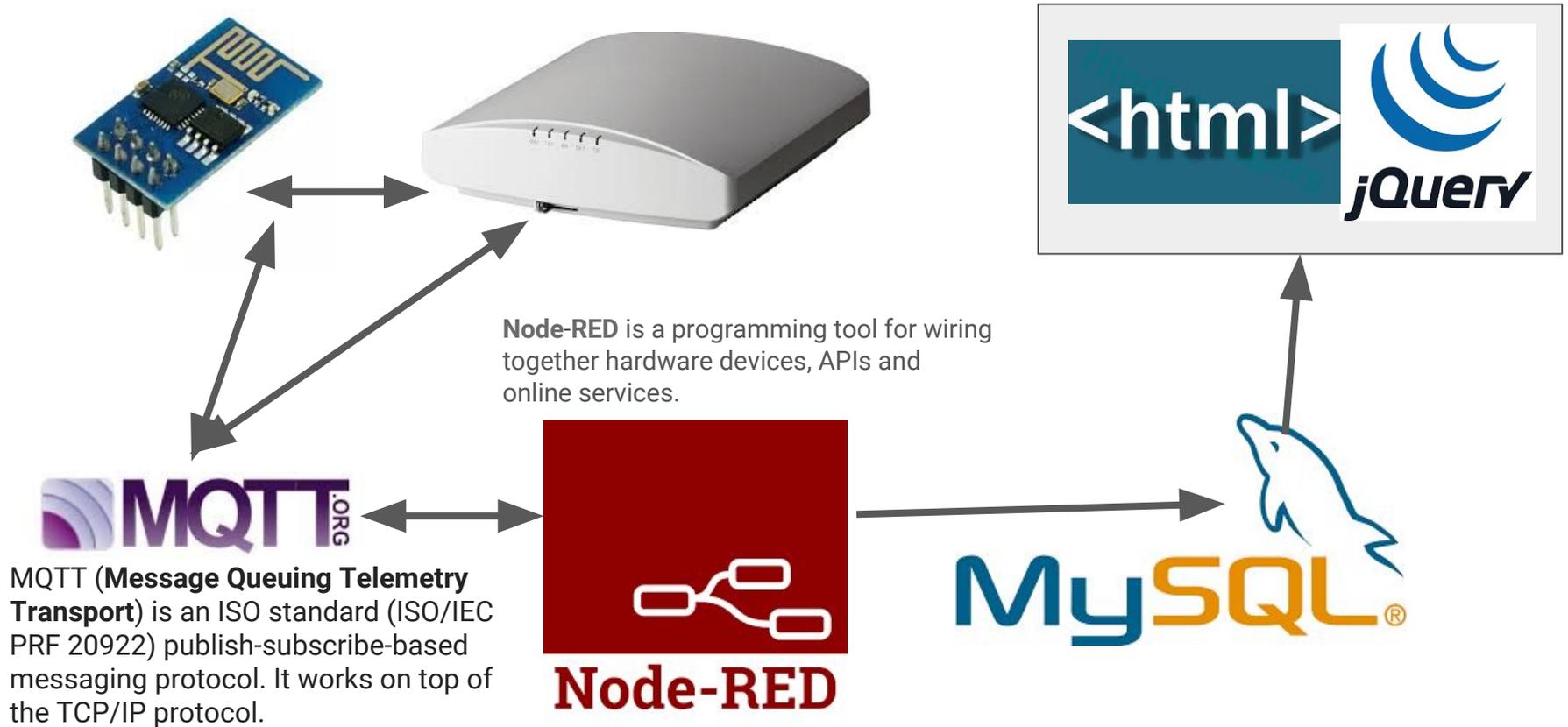
Payments:



**PayPal CREDIT**

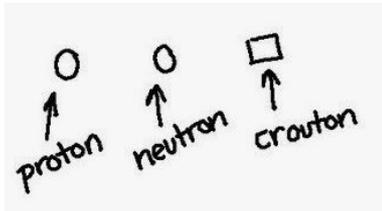
Special financing available. [Apply Now](#) | [See terms](#)

# Broad Overview of the ESP8266 Tracking Sys



## Possible uses for a very inexpensive WIFI wireless tracking device:

- 1 - some kind of visitor badge? Track all visitors in our school?
  - 2 - Maybe a student pass system that tracks?
  - 3 - Maybe a device that tracks where I am??!? (believe me - I'd rather have the office secretaries know where I am because when they don't - it is painful for everyone!)
- The above all had privacy concerns or other potential minor HR issues.... So in the end I decided the most simple and best use for this fun thing would be to track where the \*&##^&@^@ **step ladders** go all the time!





# Let's program an ESP8266

## Download and install the Arduino IDE



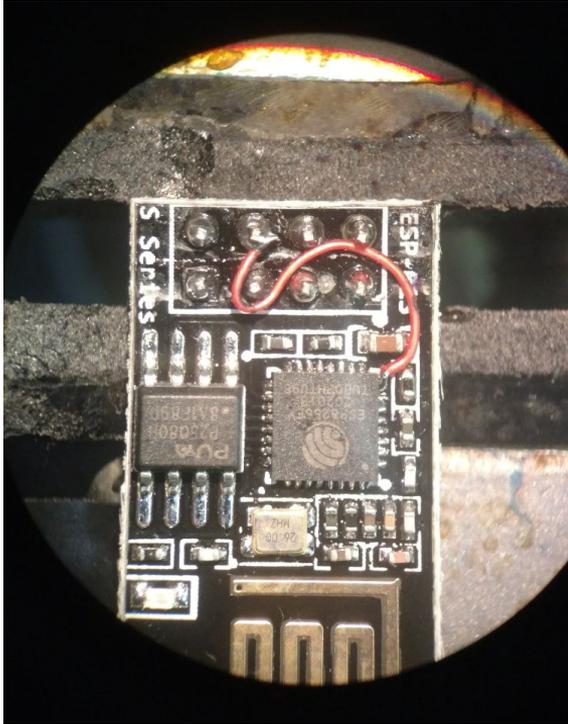
### Download the Arduino IDE

A screenshot of the Arduino IDE download page. On the left is the Arduino logo. To its right, the text reads: "ARDUINO 1.8.8 The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the Getting Started page for installation instructions." On the right side, there are three download options: "Windows installer, for Windows XP and up" and "Windows ZIP file for non admin install" (with a "Get #1" badge), "Windows app" (with a "Requires Win 8.1 or 10" badge), and "Mac OS X 10.8 Mountain Lion or newer". At the bottom, there are links for "Linux 32 bits", "Linux 64 bits", "Linux ARM", "Release Notes", "Source Code", and "Checksums (SHA512)".

## Add the ESP8266 Board to the IDE

- Start Arduino and open Preferences window.
- Enter `http://arduino.esp8266.com/stable/package_esp8266com_index.json` into *Additional Board Manager URLs* field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and find *esp8266* platform.
- Select the version you need from a drop-down box.
- Click *install* button.
- Don't forget to select your ESP8266 board from Tools > Board menu after installation.

# A quick note about “deep sleep”



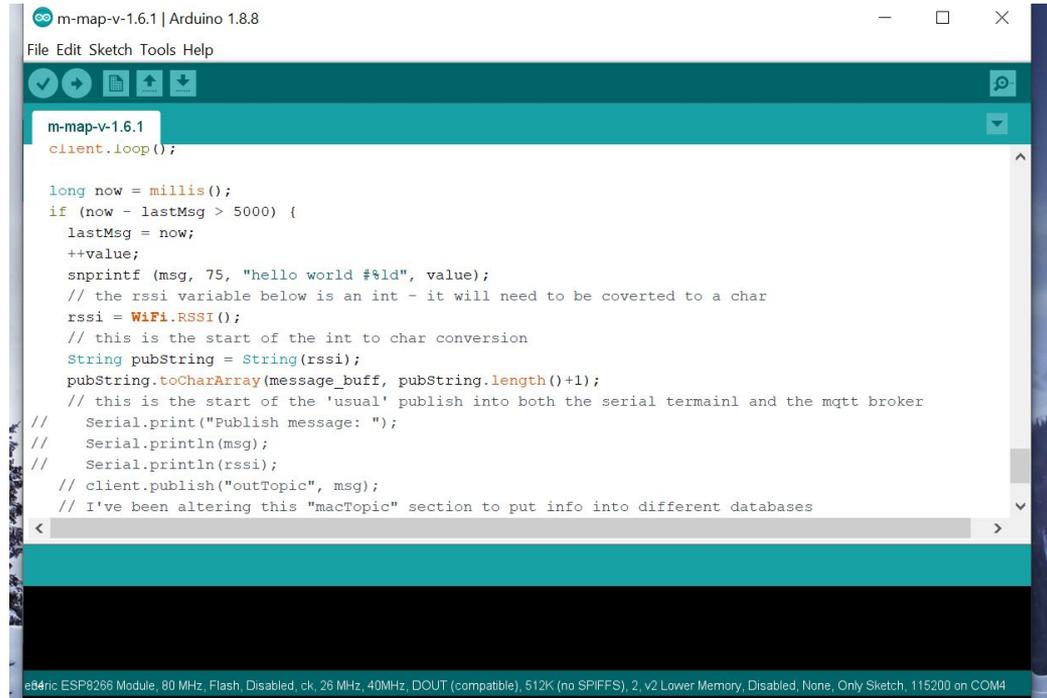
Search for:  
“esp8266 deep sleep wiring image”



# Demo of two ESP8266 Programs:

BLINK - (Hello World for arduinos)

AND “M-Map”



```
m-map-v-1.6.1 | Arduino 1.8.8
File Edit Sketch Tools Help

m-map-v-1.6.1
client.loop();

long now = millis();
if (now - lastMsg > 5000) {
  lastMsg = now;
  ++value;
  snprintf (msg, 75, "hello world %ld", value);
  // the rssi variable below is an int - it will need to be converted to a char
  rssi = WiFi.RSSI();
  // this is the start of the int to char conversion
  String pubString = String(rssi);
  pubString.toCharArray(message_buff, pubString.length()+1);
  // this is the start of the 'usual' publish into both the serial terminal and the mqtt broker
  // Serial.print("Publish message: ");
  // Serial.println(msg);
  // Serial.println(rssi);
  // client.publish("outTopic", msg);
  // I've been altering this "macTopic" section to put info into different databases
```

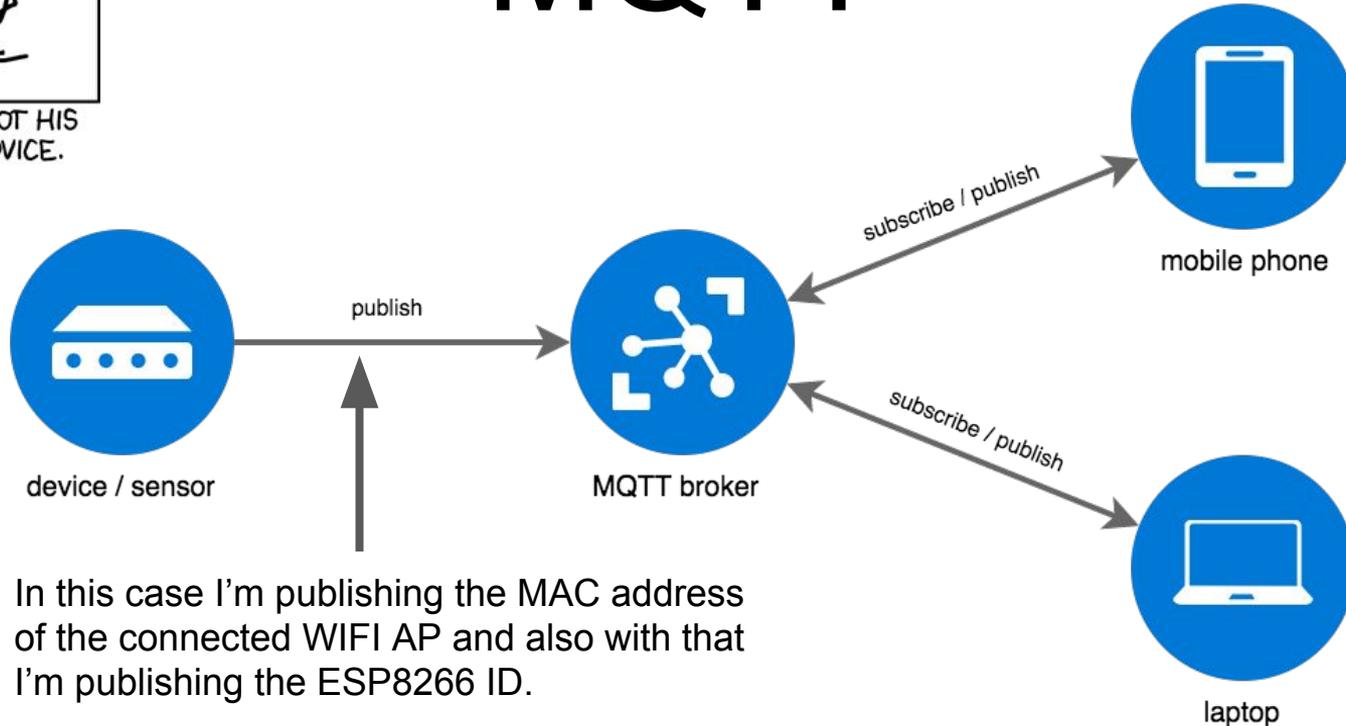
ESP8266 Module, 80 MHz, Flash Disabled, ck, 26 MHz, 40MHz, DOUT (compatible), 512K (no SPIFFS), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4

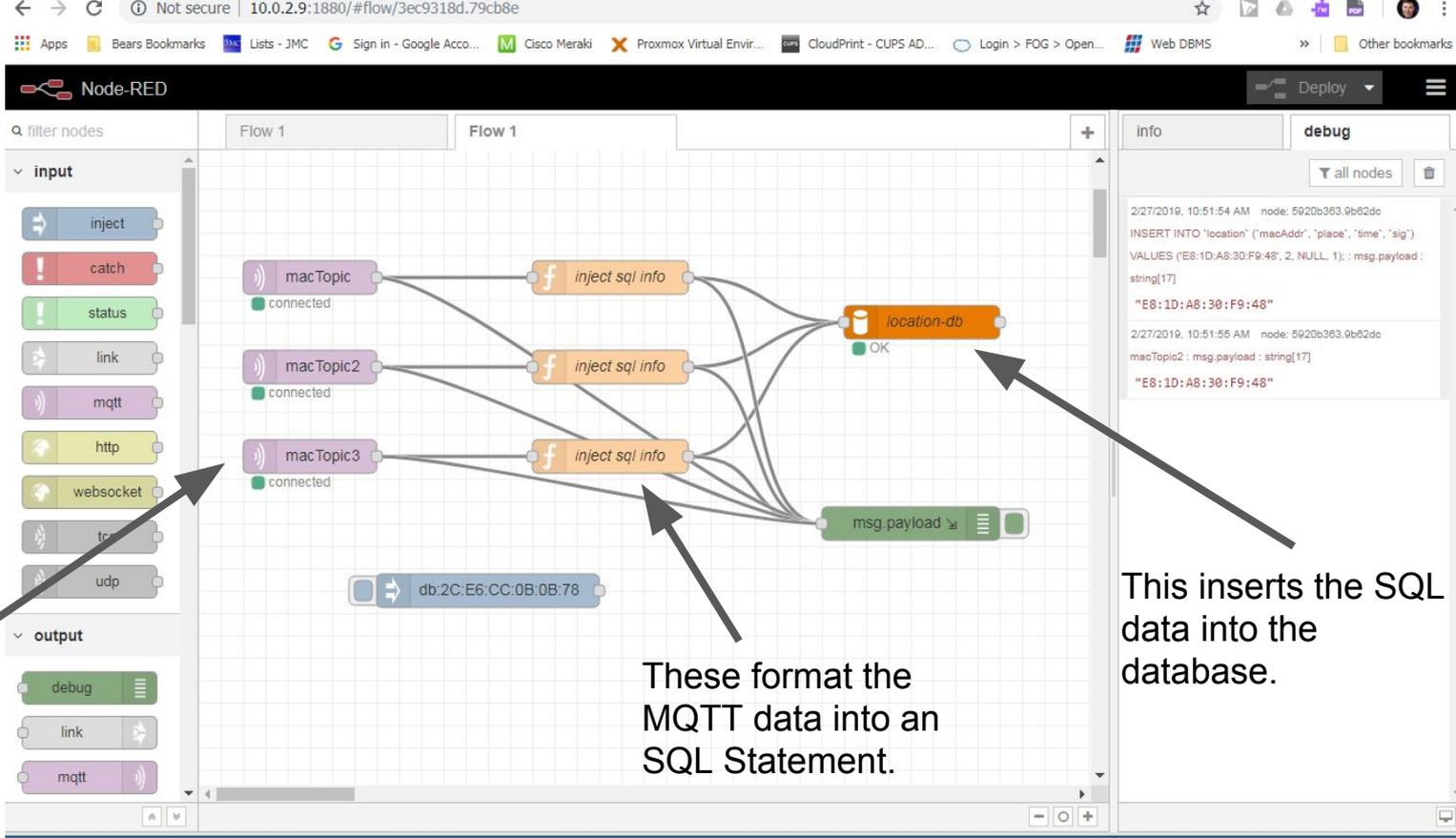
REMEMBER: WITH GREAT  
POWER COMES GREAT  
CURRENT SQUARED  
TIMES RESISTANCE.



OHM NEVER FORGOT HIS  
DYING UNCLE'S ADVICE.

# MQTT





After the ESP8266 reports the MAC address of the wifi AP it is connected to - that is fed to the MQTT Broker, and then into NODE-RED.

These "receive" from each ESP8266 - one for each unit. I only have 3 so far.

These format the MQTT data into an SQL Statement.

This inserts the SQL data into the database.

The red wire said to the black wire "Why are you so sad?"  
The black wire replied "I've been grounded"

## Demo of NODE-RED and the data coming into that from ESP8266

**NODE-RED  
routes that data  
into MYSQL.**

Server: localhost » Database: location » Table: location

Browse Structure SQL Search Insert Export Import Priv

time	macAddr	price	sig
2019-02-27 04:59:42	2C:E6:CC:09:84:68	2	1
2019-02-27 05:14:09	2C:E6:CC:09:84:68	2	1
2019-02-27 05:28:36	2C:E6:CC:09:84:68	2	1
2019-02-27 05:43:02	2C:E6:CC:09:84:68	2	1
2019-02-27 05:57:26	2C:E6:CC:09:84:68	2	1
2019-02-27 06:11:51	2C:E6:CC:09:84:68	2	1
2019-02-27 06:26:16	2C:E6:CC:09:84:68	2	1
2019-02-27 06:40:41	2C:E6:CC:09:84:68	2	1
2019-02-27 06:55:05	2C:E6:CC:09:84:68	2	1
2019-02-27 07:09:30	2C:E6:CC:09:84:68	2	1
2019-02-27 08:14:47	E8:1D:A8:30:F9:48	2	1
2019-02-27 08:27:21	E8:1D:A8:30:F9:48	2	1
2019-02-27 08:41:43	E8:1D:A8:30:F9:48	2	1
2019-02-27 08:56:06	E8:1D:A8:30:F9:48	2	1
2019-02-27 09:10:30	E8:1D:A8:30:F9:48	2	1
2019-02-27 09:24:55	E8:1D:A8:30:F9:48	2	1
2019-02-27 09:39:20	E8:1D:A8:30:F9:48	2	1
2019-02-27 09:53:44	E8:1D:A8:30:F9:48	2	1
2019-02-27 10:08:09	E8:1D:A8:30:F9:48	2	1
2019-02-27 10:22:34	E8:1D:A8:30:F9:48	2	1
2019-02-27 10:36:58	E8:1D:A8:30:F9:48	2	1

<< < 6387 ▾ | Number of rows: 25 ▾ | Filter rows: Search this table

Query results operations

- Why do programmers always get Christmas and Halloween mixed up?
- Because DEC 25 = OCT 31!!

**Another demo!**

# A Random Youtube plug / endorsement....

The screenshot shows a YouTube search results page for the query 'b'. The search bar at the top contains the letter 'b'. The navigation tabs include HOME, VIDEOS, PLAYLISTS, COMMUNITY, CHANNELS, and ABOUT. The search results are displayed in a grid format. The first row of results includes:

- Troubleshooting: Investigating why the LED is...** (15K views • 4 years ago)
- How a transistor amplifies current | Digital electronics ...** (41K views • 4 years ago)
- Connecting to a mystery signal | Digital electronics (...)** (16K views • 4 years ago)
- Powering our LED circuit | Digital electronics (3 of 10)** (17K views • 4 years ago)

The second row of results includes:

- Limiting current through an LED | Digital electronics (2 o...** (21K views • 4 years ago)
- Experimenting with LEDs | Digital electronics (1 of 10)** (42K views • 4 years ago)
- Programming Fibonacci on a breadboard computer** (106K views • 4 years ago)
- Comparing C to machine language** (1M views • 4 years ago)

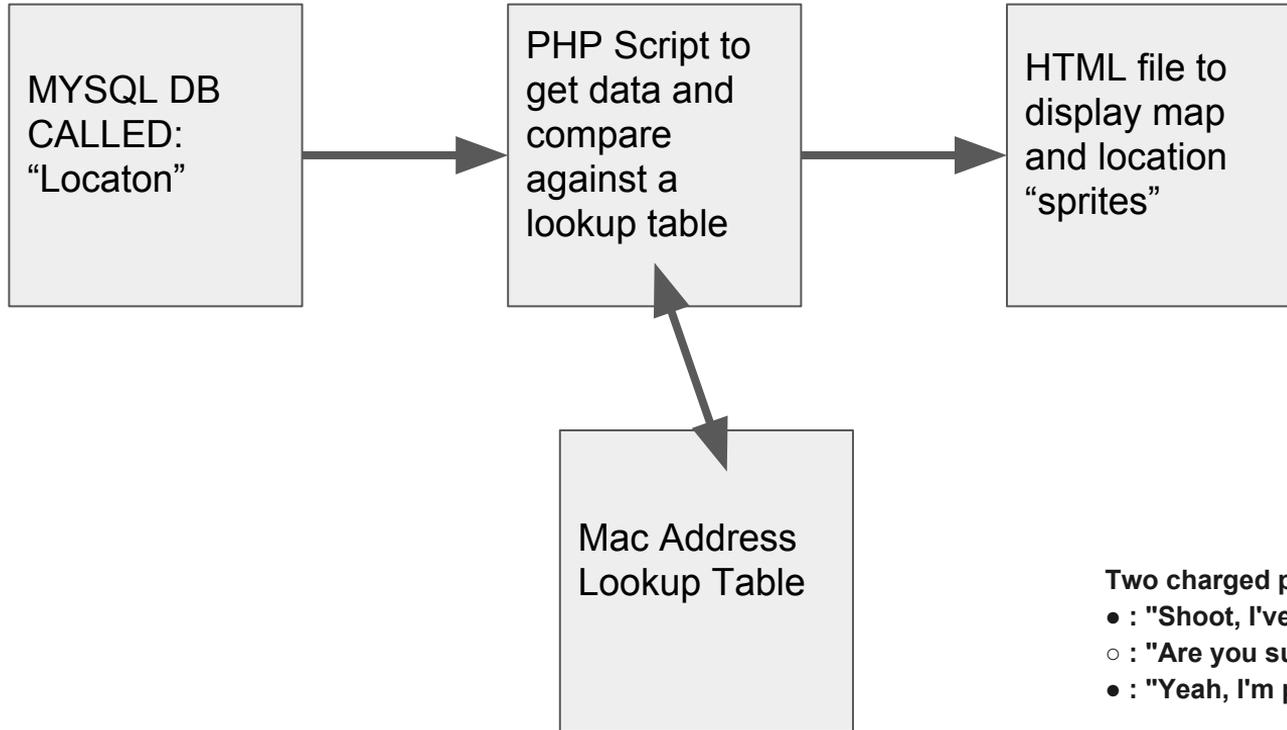
The third row of results includes:

- Stepping through a program on the 8-bit breadboard...** (113K views • 4 years ago)
- Programming my 8-bit breadboard computer** (408K views • 4 years ago)
- TCP: Transmission control protocol | Networking tutori...** (74K views • 4 years ago)
- Frame formats | Networking tutorial (6 of 13)** (55K views • 4 years ago)

The left sidebar shows navigation options: Home, Trending, Subscriptions, Library, History, Watch later, Blues Jam Tracks, sign, Show more, SUBSCRIPTIONS (Ben Eater), Browse channels, and MORE FROM YOUTUBE (YouTube Premium).

If you are geeked out by electronics and computer science, do yourself a favor and check out the youtube channel of “Ben Eater”. His series about building an 8-bit breadboard computer is MIND BLOWING. I binge watched it much like I binge watched the HBO series “Barry” recently. Could not stop myself from watching. I swear to you Ben is the “Bob Ross” of computer science / electronics. Go there now!

And Finally - the data gets polled from MySQL and displayed on a web page using some jQuery



Two charged particles on the way home from the bar:

- : "Shoot, I've lost an electron!"
- : "Are you sure? Check." {{ Checks... }}
- : "Yeah, I'm positive."

# Look at the php and html

## ***3 main files:***

- “test.html” .... Named it when I didn’t expect it to work... now I’m just too lazy to change it to something other than “test”.
- “Send\_sse.php” - this is the php script that moves map coordinates into the html from the MySQL server.
- “macTable.php” - this is the “rosetta stone” file that has a listing of all the WIFI MAC addresses and associates them with CSS screen coordinates.

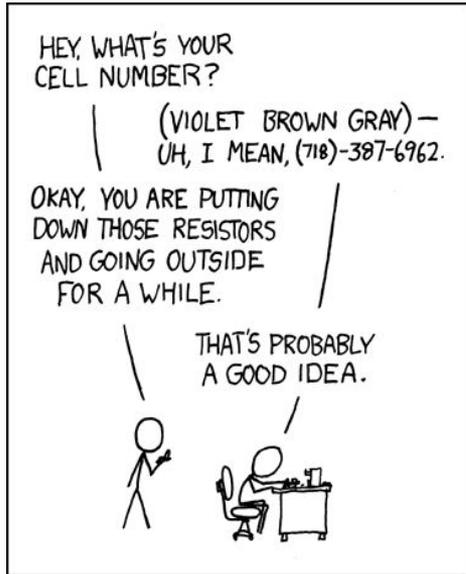
(also a couple image files “6ftladder2.png” etc.)





Things that **MUST**  
be installed onto  
your linux server:

MySQL  
httpd  
MQTT  
NODE-RED  
PHP



And this is helpful:  
phpMyAdmin

*Google search term hint: (preaching to the choir)*

*<operating system> <application> Install*

*eg: "centos msyql install"*

# Thank you!

Questions?

scott.wegner@isd786.org

