Andrea De Gaetano - @dega1999

An adventure with ESP8266 and IOT

Andrea De Gaetano

Genova, Italy

My blog: http://pestohacks.blogspot.com

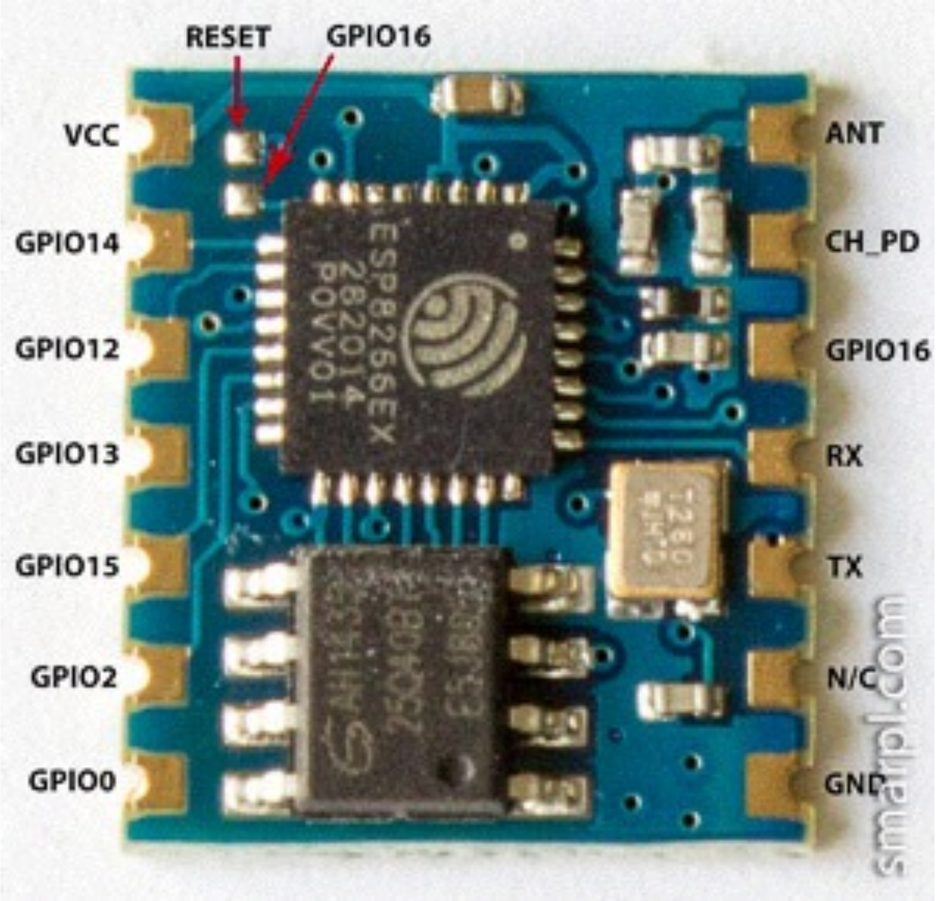
"What I have learnt since I started playing with ESP8266"

- Introduction to ESP8266 hardware
- Getting started: ESP8266 versions, software and hardware requirements, wirings
- Official firmware, arduino and the EspressIF software
- Alternative firmwares: frankestain, micropython, nodemcu
- The NodeMCU project
- MQTT and Mosquitto
- Visualize data: web client
- Demo project
- Future..

HARDWARE

Wifi enabled - Microcontroller 802.11 b/g/n

CPU: 80 Mhz (160 Mhz overclock?)
RAM: 64 kb istruction + 96 kb data
ROM: from 512kb to 4mb
I2C, GPIO, SPI



C++ SDK Available
Cheap! < 7-8€

Wifi Capabilities

- Station Mode:
  - act as an access point

- Access Point Mode:
  - connect to an access point
  - issues with password < 8 characters

- Mixed Mode:
  - the two mode together

What this product is useful for?

Device prototyping

IOT prototyping

Extends your project with wireless support

Simulate a network of devices

Network fuzzer

Variants / Revisions

ESP-01

ESP-02

ESP-03

ESP-04

ESP-05

ESP-06

ESP-07

ESP-08

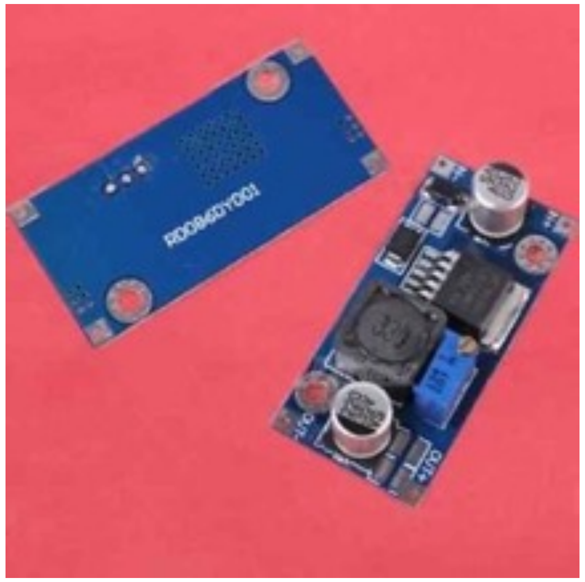
ESP-09

ESP-10

ESP-11

Spoiler: not all the variants are here

Tools



Stable 3.3v power

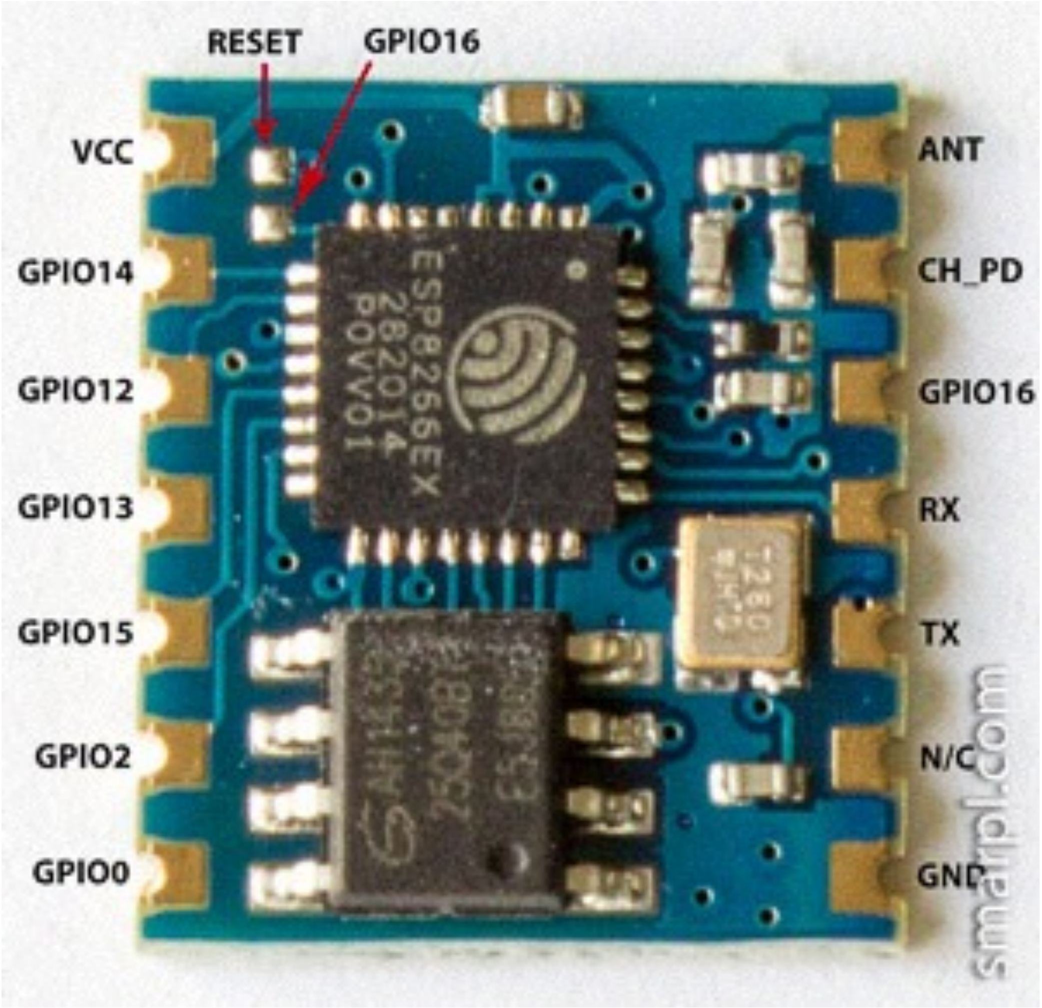
USB to serial TTL 3.3v

serial terminal: kermit, minicom …

esptool.py: flasher tool

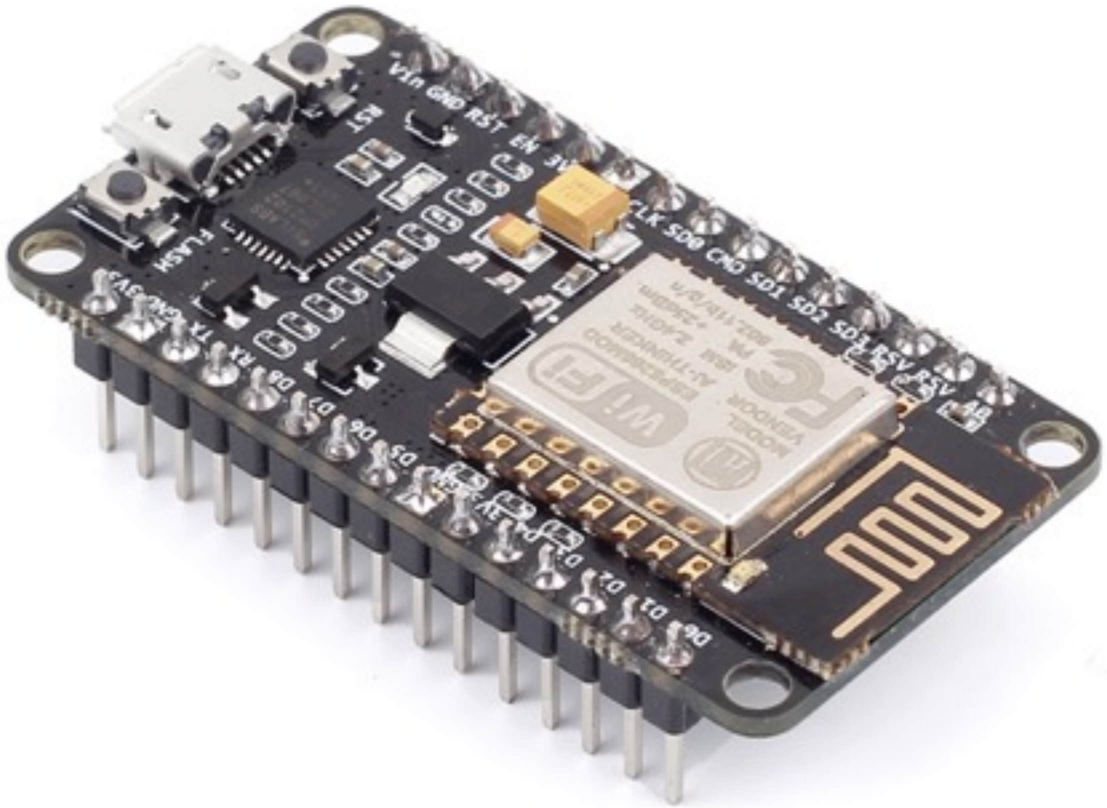
luatool: (nodemcu project only)

ESP8266-04 Wirings



Esp8266	CH340CG - USB	Power Supply
RX	TX	
TX	RX	
VCC		+3.3v
GND	**GND**	**GND**
GPIO15		**GND**
CH_PD		+3.3v
GPIO0 (only for flashing)		**GND**

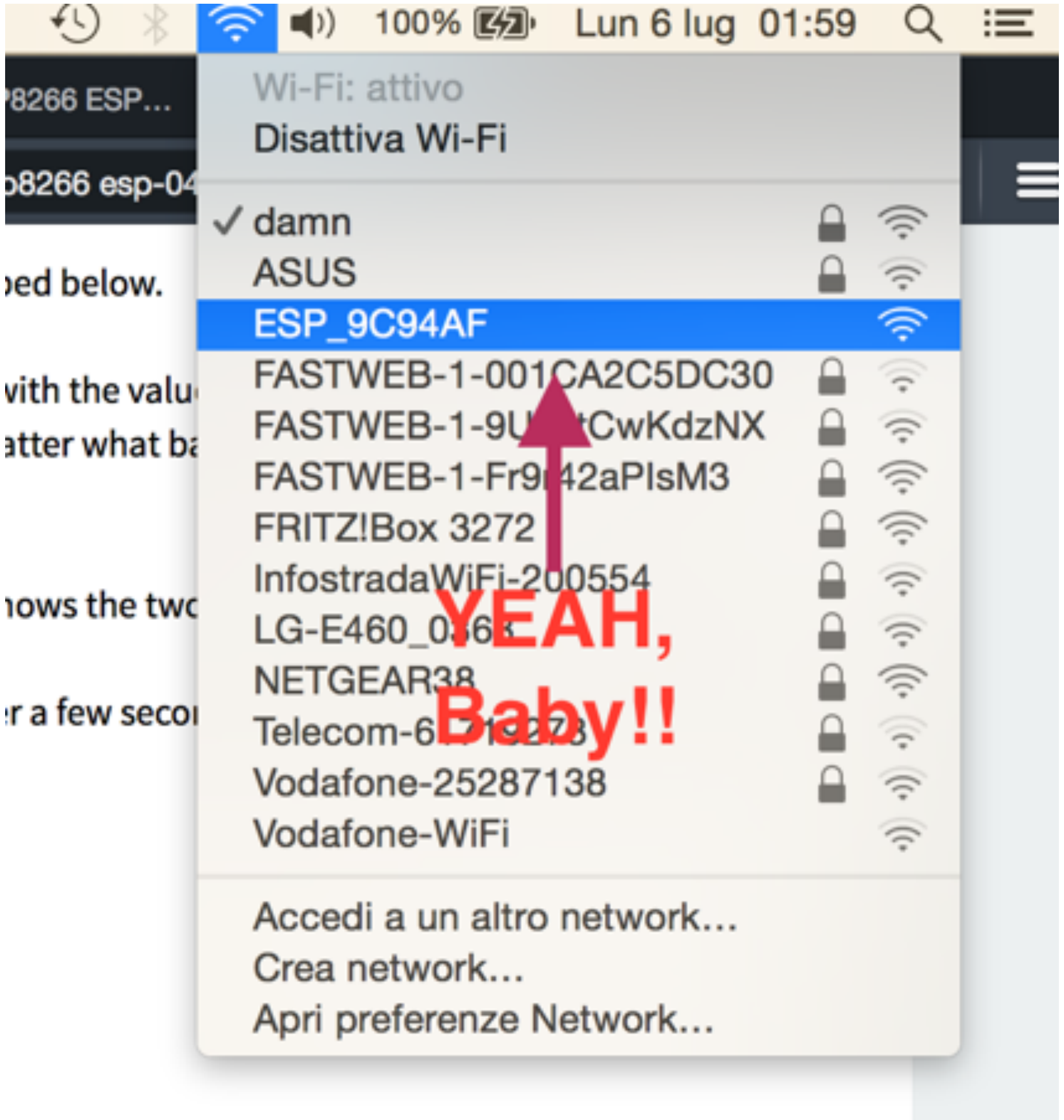
NodeMCU DevKit



- Easy: no solder required
- Less funny
- Price ~15 euro
- Why didn't I used it?

because I burned one… :/

When you Power On with official firmware



100% Lun 6 lug 01:59

8266 ESP...
8266 esp-04
ed below.
vith the valu
atter what ba
iows the twc
r a few seco

Wi-Fi: attivo
Disattiva Wi-Fi

✓ damn
ASUS
ESP_9C94AF
FASTWEB-1-001CA2C5DC30
FASTWEB-1-9U…tCwKdzNX
FASTWEB-1-Fr9…42aPlsM3
FRITZ!Box 3272
InfostradaWiFi-200554
LG-E460_0…6…
NETGEAR38
Telecom-6…
Vodafone-25287138
Vodafone-WiFi

Accedi a un altro network...
Crea network...
Apri preferenze Network...

YEAH, Baby!!

It creates a wireless network with a "ESP" prefix …

net packets to /dev/null

A first check "it works!"

Serial speed is 9660 kbps or 115200 kbps

Configurations depends on the firmware version

Lesson: **Flash a well known firmware**

Flash the firmware

```
python esptool.py -p /dev/tty.usbserial write_flash
0x00000 bin/0x00000.bin 0x10000 bin/0x10000.bin
```

https://github.com/themadinventor/esptool

- /dev/tty.usbserial depends on USB to Serial converter
- Addresses (e.g. 0x00000) depends on firmware, check docs!

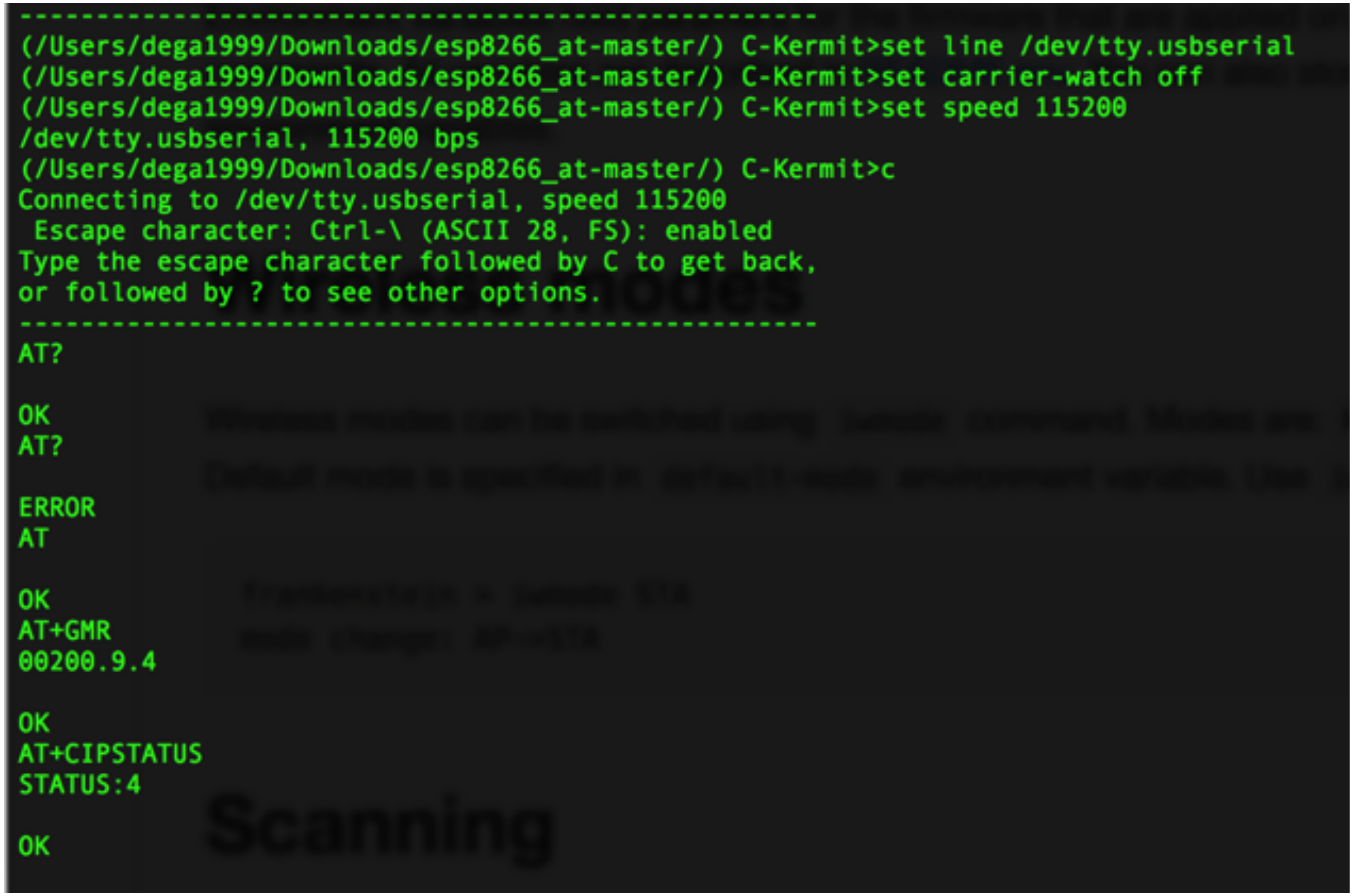
Note for NodeMCU devkit

Use their flasher: https://github.com/nodemcu/nodemcu-flasher

- for Windows
- a Multiplatform version with QT: Not available yet

An Old Friend: AT Commands

The official firmware comes with AT commands support



```
----------------------------------------------------
(/Users/dega1999/Downloads/esp8266_at-master/) C-Kermit>set line /dev/tty.usbserial
(/Users/dega1999/Downloads/esp8266_at-master/) C-Kermit>set carrier-watch off
(/Users/dega1999/Downloads/esp8266_at-master/) C-Kermit>set speed 115200
/dev/tty.usbserial, 115200 bps
(/Users/dega1999/Downloads/esp8266_at-master/) C-Kermit>c
Connecting to /dev/tty.usbserial, speed 115200
 Escape character: Ctrl-\ (ASCII 28, FS): enabled
Type the escape character followed by C to get back,
or followed by ? to see other options.
----------------------------------------------------
AT?

OK
AT?

ERROR
AT

OK
AT+GMR
00200.9.4

OK
AT+CIPSTATUS
STATUS:4

OK
```

http://wiki.iteadstudio.com/ESP8266_Serial_WIFI_Module#AT_Commands

(Some) Official Firmware Features

Control Device settings:

- Wireless modes: access point/station mode
- Wireless network name and password (station mode)
- on/off wifi
- …

Send / Receive data over TCP

- client TCP: a rudimental browser
- server TCP: a slow http server

Frustrating experience over terminal

EspressIF SDK (the official SDK)

- Released in Oct 2014
- Works under Linux
- Give major control over the device: GPIO, I2C, SPI
- Can Extends AT commands
- Generate firmware
- It's C++: hard debugging and possible infinite reboot on faulty fw

http://bbs.espressif.com/

Alternatives: Arduino's Way

- Use Arduino IDE modified: https://github.com/esp8266/arduino
- Write sketch with "Arduino-C"

Or …

- Connect an Arduino with RX/TX and Use AT commands: because arduino shields for wifi are expensive

PRO: useful to give connectivity to existing arduino projects

Not interested… Never tried…

Alternatives: Firmwares

Some months later the community bring us new **opensource** firmwares!

- Frankenstein: described as "quick and dirty firmware". Good console support.
  https://github.com/nekromant/esp8266-frankenstein
- MicroPython: programmable with python. Good for prototyping
  https://github.com/micropython/micropython/tree/master/esp8266
- NodeMCU: programmable with LUA, very good API!
  https://github.com/nodemcu/nodemcu-firmware

Repeat: All code is opensource!

Nodemcu

Program are written with LUA: easy syntax, easy to learn.

Program === **Scripts: no** need to recompile and flash firmware

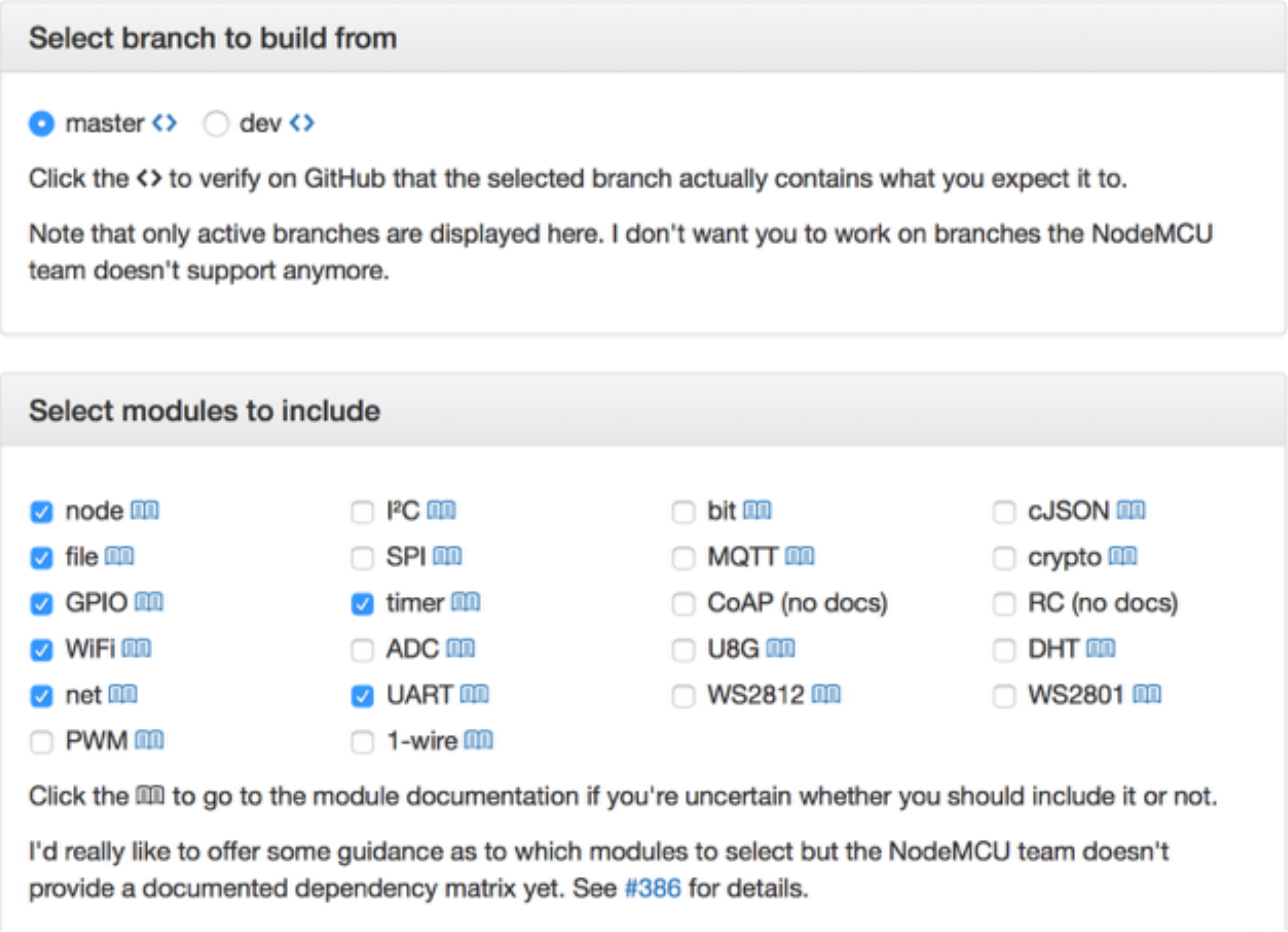
Commands testable directly from the terminal

A lot of good and useful api for:

- GPIO
- I2C
- SPI
- MQTT
- File management read/write
- UART
- 1-wire bus

Working with NodeMCU

Download a firmware or build a new one



Select branch to build from

● master <> ○ dev <>

Click the <> to verify on GitHub that the selected branch actually contains what you expect it to.

Note that only active branches are displayed here. I don't want you to work on branches the NodeMCU team doesn't support anymore.

Select modules to include

- ☑ node 📖
- ☑ file 📖
- ☑ GPIO 📖
- ☑ WiFi 📖
- ☑ net 📖
- □ PWM 📖
- □ I^{2}C 📖
- □ SPI 📖
- ☑ timer 📖
- □ ADC 📖
- ☑ UART 📖
- □ 1-wire 📖
- □ bit 📖
- □ MQTT 📖
- □ CoAP (no docs)
- □ U8G 📖
- □ WS2812 📖
- □ cJSON 📖
- □ crypto 📖
- □ RC (no docs)
- □ DHT 📖
- □ WS2801 📖

Click the 📖 to go to the module documentation if you're uncertain whether you should include it or not.

I'd really like to offer some guidance as to which modules to select but the NodeMCU team doesn't provide a documented dependency matrix yet. See #386 for details.

http://frightanic.com/nodemcu-custom-build/

A Project with NodeMCU

I decided to test the device on 2 topics I didn’t know:

- MQTT
- I2C

Project Summary:

Use ESP8266 to:

- Sends accelerometer data(x,y,z) coming from an I2C sensor
- Use MQTT
- visualize the data in the browser

It’s all started with the idea of building a drone.. but this is another story :)

The accelerometer: ADXL345

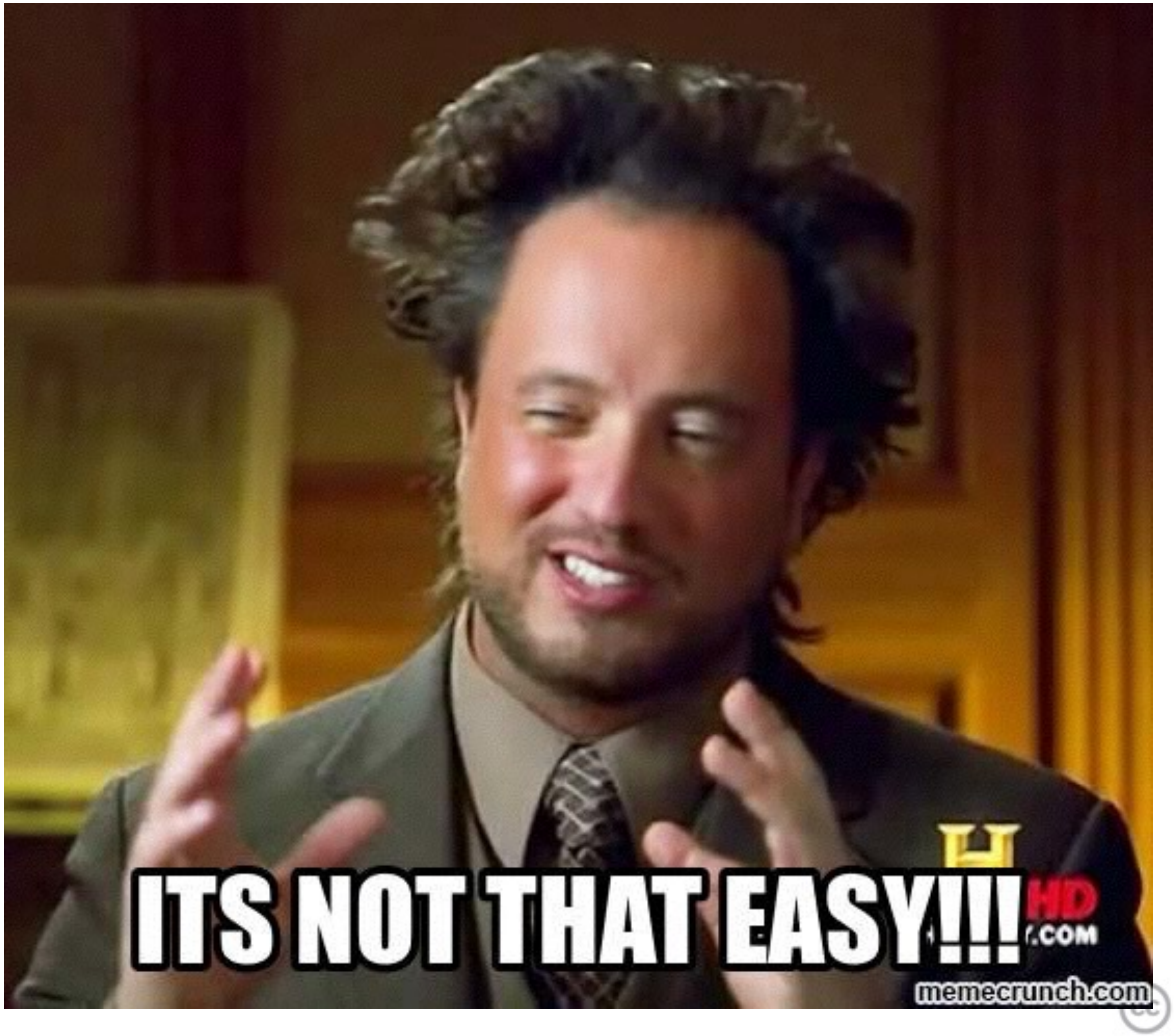
It supports:

- I2C
- SPI

Send:

- Raw data
- Recognize gestures: tap, double tap..
- Recognize events: free fall
- …

Problem: no driver for control the device
Solution: Ok, write the driver

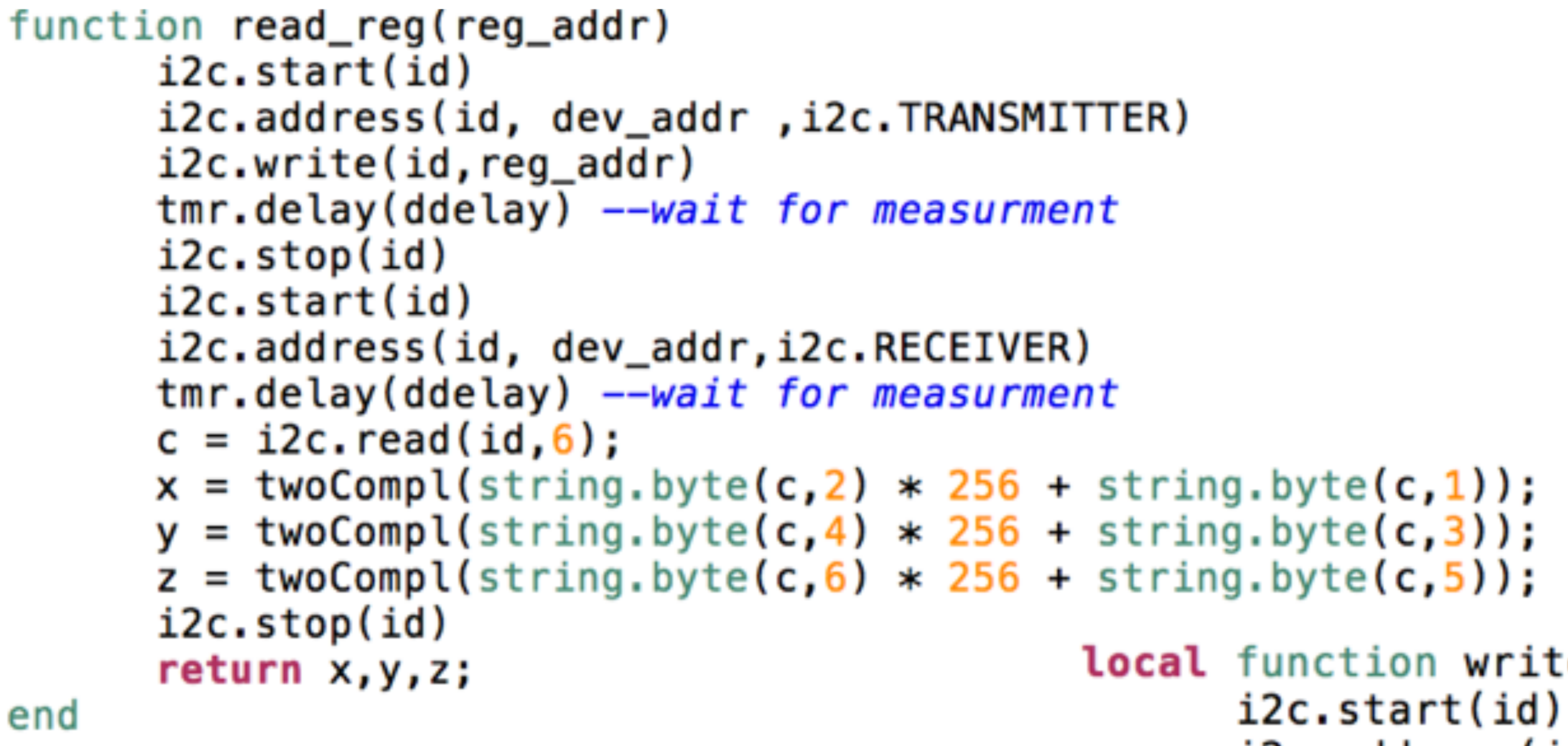

ITS NOT THAT EASY!!!
HD
.COM
memecrunch.com

ADXL345 Driver

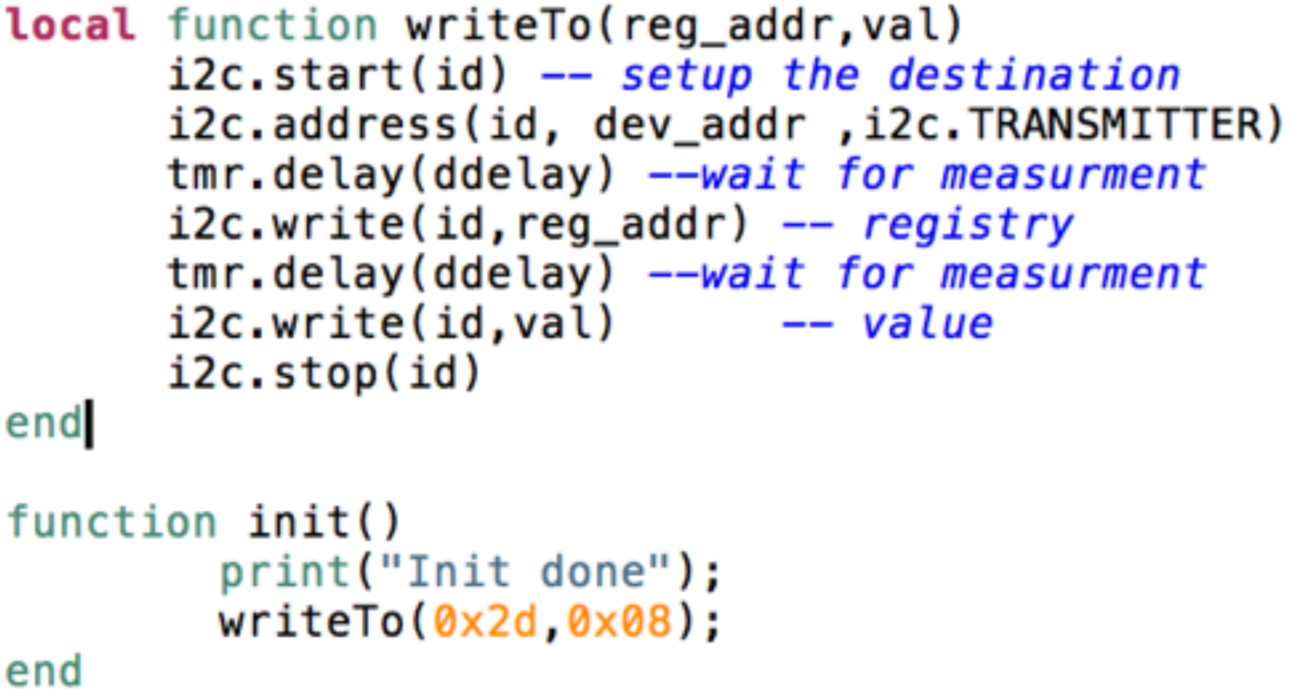
How I wrote the I2C driver:

- Read about I2C protocol
- Read the data sheet of ADXL345
- Read the Arduino driver/libraries C++ sourcecode(s)
- Find the I2C pin: SDA,SCL (i2c_scanner.lua)
- while (notworking)
  - read nodemcu API
  - write code
  - upload code
  - test
- while (somethingIsWorkingButNotYet)
  - read documentation again
  - change the code
  - test

ADXL345 Driver’s Code!



```
function read_reg(reg_addr)
        i2c.start(id)
        i2c.address(id, dev_addr ,i2c.TRANSMITTER)
        i2c.write(id,reg_addr)
        tmr.delay(ddelay) --wait for measurment
        i2c.stop(id)
        i2c.start(id)
        i2c.address(id, dev_addr,i2c.RECEIVER)
        tmr.delay(ddelay) --wait for measurment
        c = i2c.read(id,6);
        x = twoCompl(string.byte(c,2) * 256 + string.byte(c,1));
        y = twoCompl(string.byte(c,4) * 256 + string.byte(c,3));
        z = twoCompl(string.byte(c,6) * 256 + string.byte(c,5));
        i2c.stop(id)
        return x,y,z;
end
```



```
local function writeTo(reg_addr,val)
        i2c.start(id) -- setup the destination
        i2c.address(id, dev_addr ,i2c.TRANSMITTER)
        tmr.delay(ddelay) --wait for measurment
        i2c.write(id,reg_addr) -- registry
        tmr.delay(ddelay) --wait for measurment
        i2c.write(id,val)        -- value
        i2c.stop(id)
end

function init()
        print("Init done");
        writeTo(0x2d,0x08);
end
```

It will be available on my github

IT JUST
"WORKS"
memegenerator.net

MQTT shortly

MQ Telemetry Transport is a publish-subscribe based "light weight" messaging protocol for use on top of the TCP/IP protocol.

It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited.

Andy Stanford-Clark and Arlen Nipper of Cirrus Link Solutions authored the first version of the protocol in **1999**.

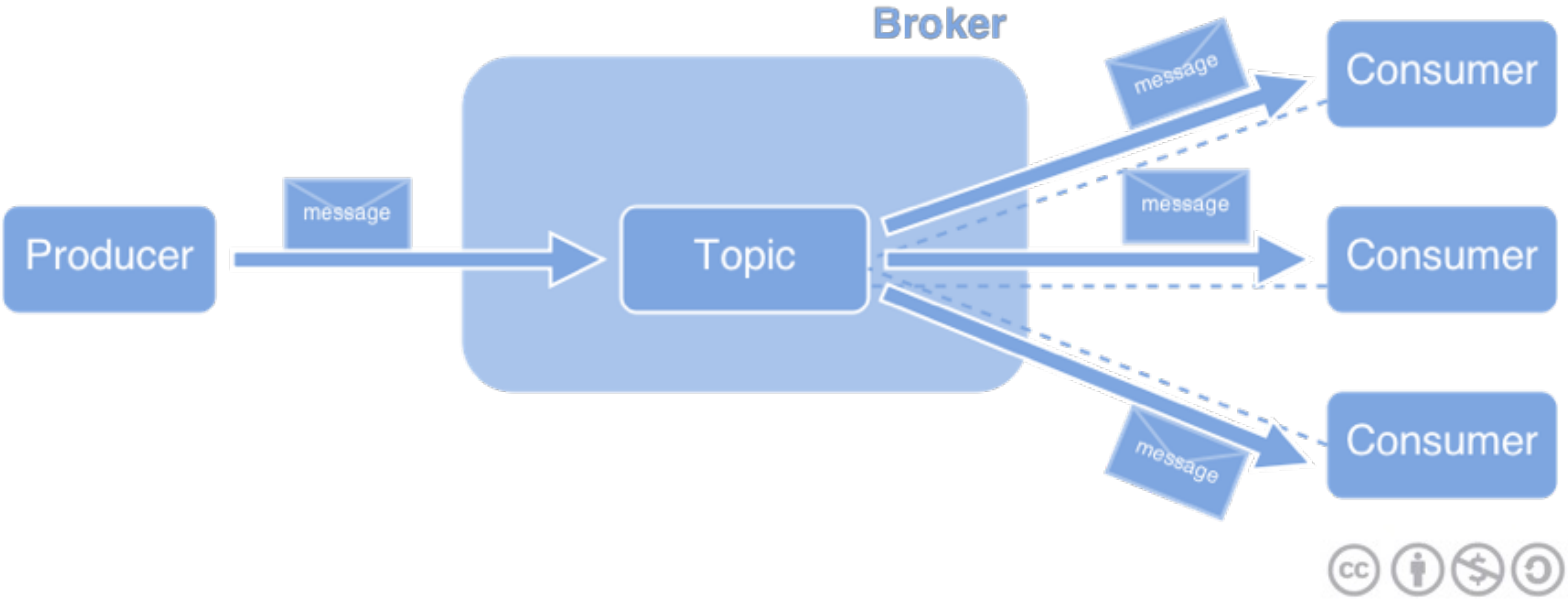
The specification does not specify the meaning of "small code foot print" or the meaning of "limited network bandwidth".

Used for M2M communication
Used on Facebook messenger

MQTT shortly

Actors:

- **broker**: who will “route” messages
- **subscriber**: who is subscribed to a **topic**
- **publisher**: who is producing messages on a **topic**



Mosquitto

Mosquitto is an open source (BSD licensed) message broker that implements the MQ Telemetry Transport protocol versions 3.1 and 3.1.1.

It is released with 3 main tools:

mosquitto: the broker
mosquito_pub: the publisher
mosquito_sub: the subscriber

MQTT: NodeMCU API

mqtt.Client()

- mqtt.client:connect()
- mqtt.client:close()
- mqtt.client:publish()
- mqtt.client:subscribe()
- mqtt.client:on()

Great!

MQTT: NodeMCU API first tests

Trying some LUA code:

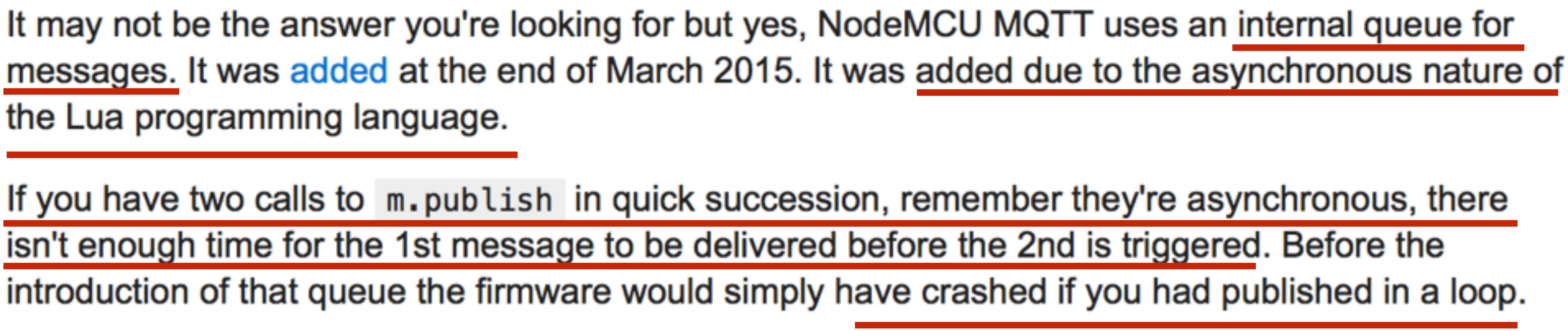
- sample code on the website works >> Great!
- sending multiple messages works >> Great!
- using it with mosquito works >> Great!
- sending more than 100 messages in a loop crash the firmware; the device reboots >> OH
- messages are slowly sent >> MY
- average frequency of 1msg/sec >> GOD!

Figuring out … Why?

Analyzing NodeMCU - MQTT implementation

After reading the C++ sourcecode on github of nodemcu..
I asked on SO what's the problem with slow messages

From stackoverflow



It may not be the answer you're looking for but yes, NodeMCU MQTT uses an internal queue for messages. It was added at the end of March 2015. It was added due to the asynchronous nature of the Lua programming language.

If you have two calls to `m.publish` in quick succession, remember they're asynchronous, there isn't enough time for the 1st message to be delivered before the 2nd is triggered. Before the introduction of that queue the firmware would simply have crashed if you had published in a loop.

Original thread:
http://stackoverflow.com/questions/33414441/nodemcu-and-esp8266-slow-mqtt-publish/

Compile a NodeMCU Firmware

- The power of source code
- Compile a new firmware is not that hard!! :)
- Patching code is not complicated
- NodeMCU is modular
- Enable debug (print) is useful but, verbose
- Disable unused modules
- There are vagrant images available on github

https://github.com/kmpm/esp8266-vagrant

NodeMCU Customized Firmware

Changes:

- F*ck queue for publishing messages
- Messages are published directly

Message order is controlled on subscriber side: if a message is in incorrect order… bye bye!

Send the message inside a thread means:

No crash… :)

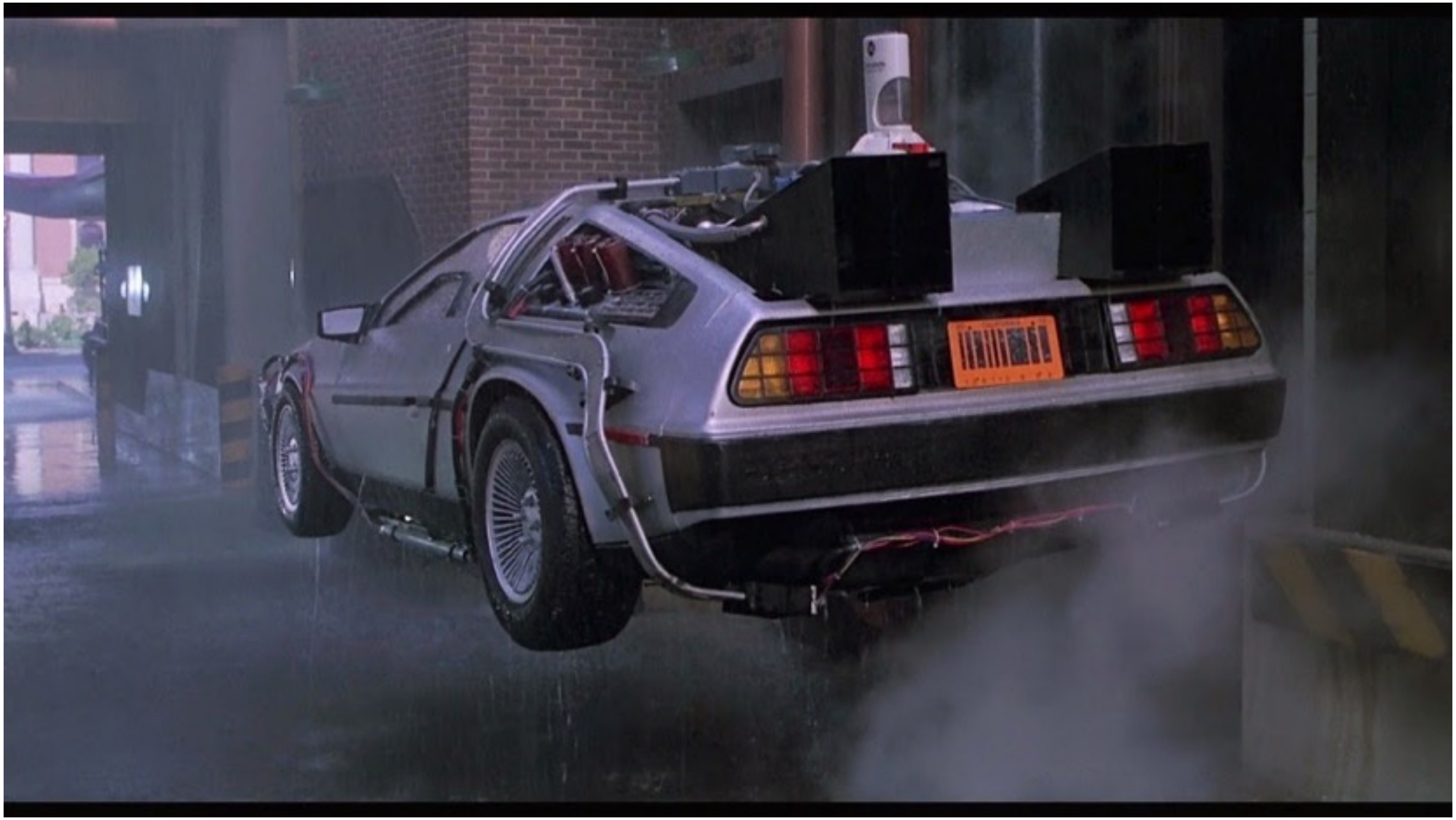
Summary of the algorithm

- Setup ADXL345
- Setup MQTT connection
- Every X ms times
    - Read the data from ADXL345
    - Publish the data with MQTT
    - repeat

MQTT on the browser? WebSocket is the answer!

Nodejs + websocket + mqtt.js

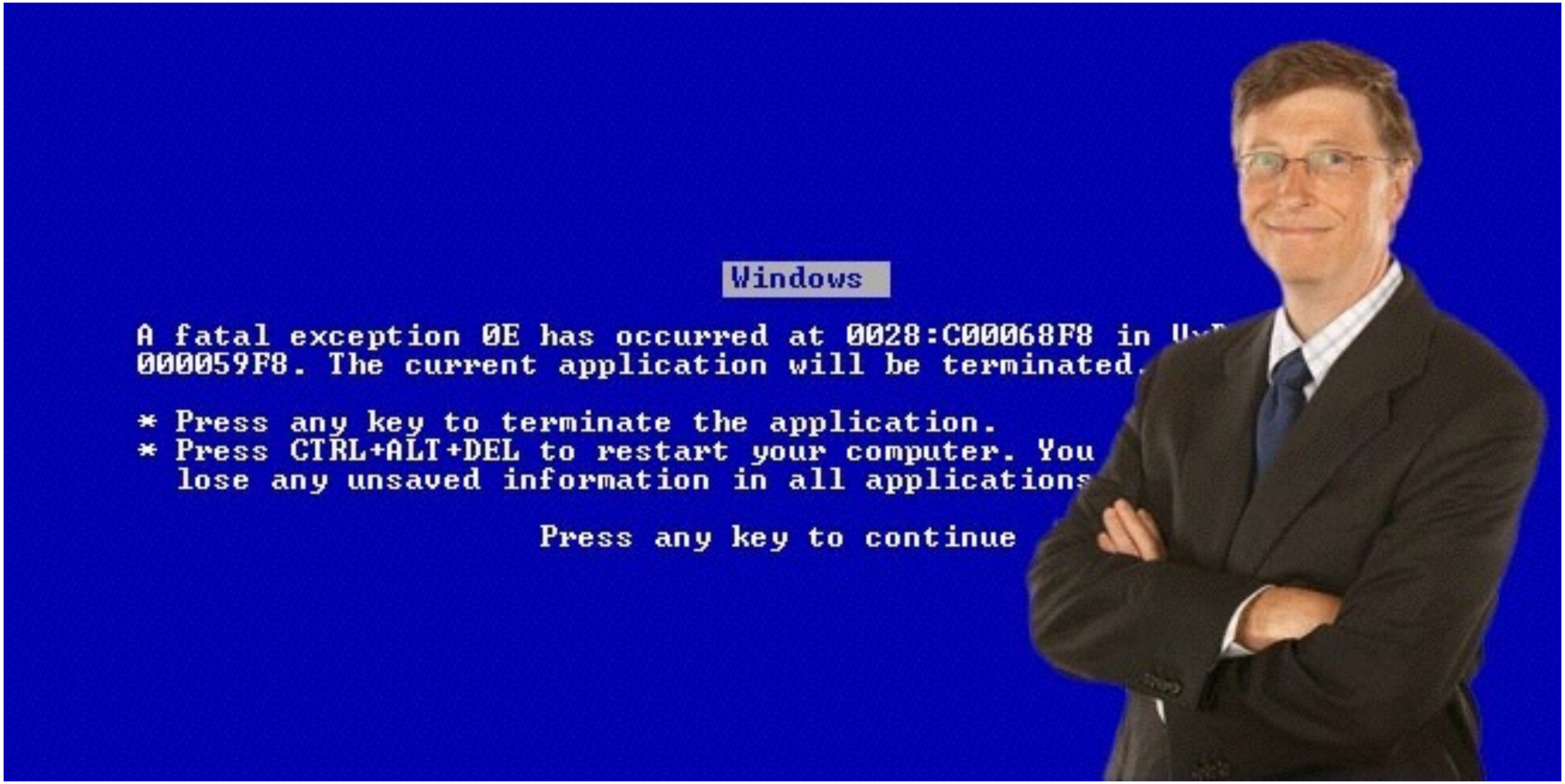
Future



Future: The ESP32

- Faster Wifi (~144 mb/s === streaming!)
- Double Processor: 160 Mhz (because "tciu is megl che uan")
- More GPIOs
- More ram: 400kb! (Remember: "640kb is enough" B.Gates)
- Bluetooth LE day one (Full Bluetooth support later)
- Not a replacement of ESP8266
- "NodeMCU compatible".. maybe!

Demo Time



Windows

```
A fatal exception 0E has occurred at 0028:C00068F8 in U
000059F8. The current application will be terminated.

* Press any key to terminate the application.
* Press CTRL+ALT+DEL to restart your computer. You
  lose any unsaved information in all applications

           Press any key to continue
```

All the code will be on repos here next week:
https://github.com/crazycoder1999/

check my twitter for update:
@dega1999

or my blog

Leave your feedback on Joind.in!
https://m.joind.in/event/codemotion-milan-2015