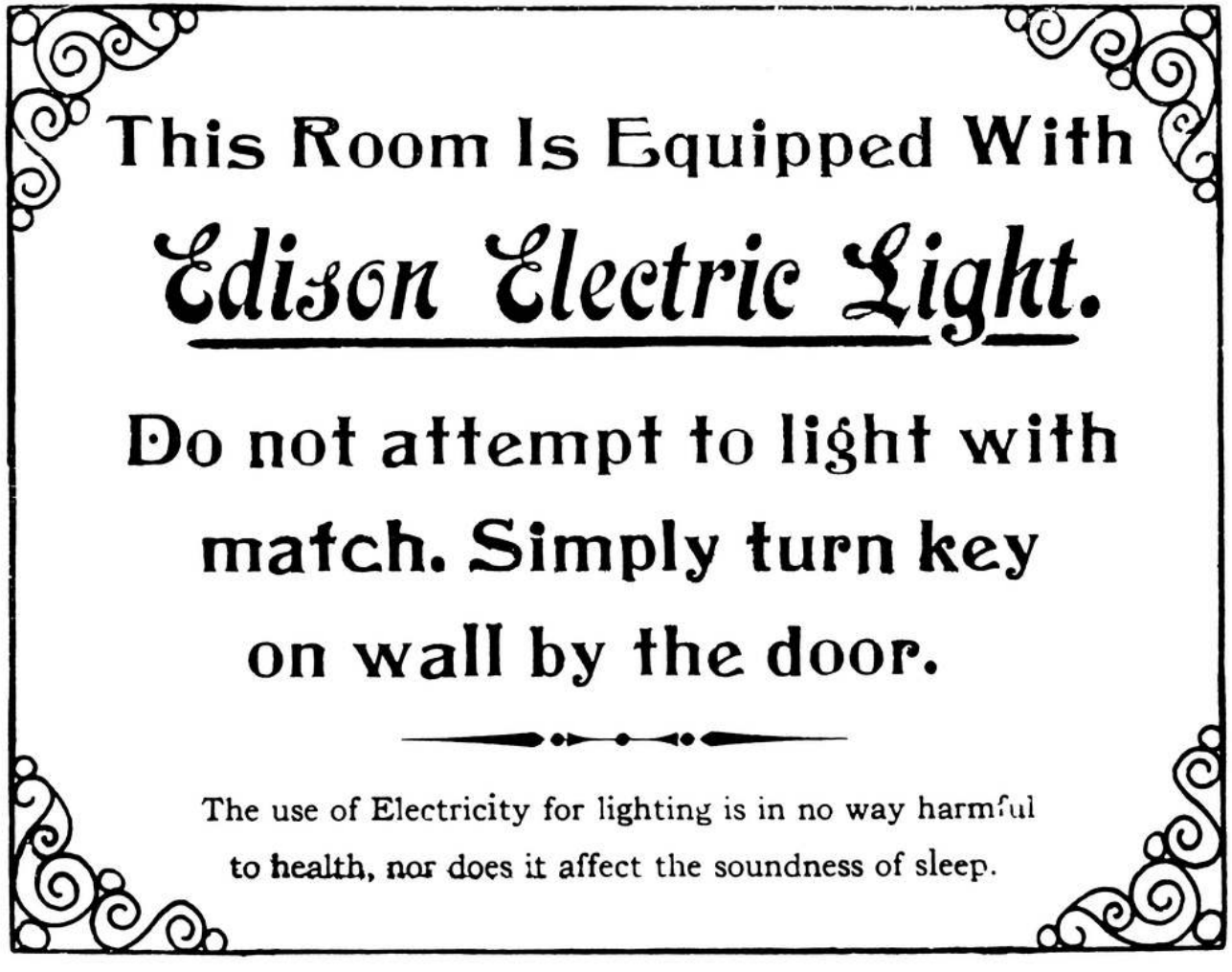
HOME AUTOMATION WITH PYTHON

... HOW TO RULE YOUR HOME WITH PYTHON

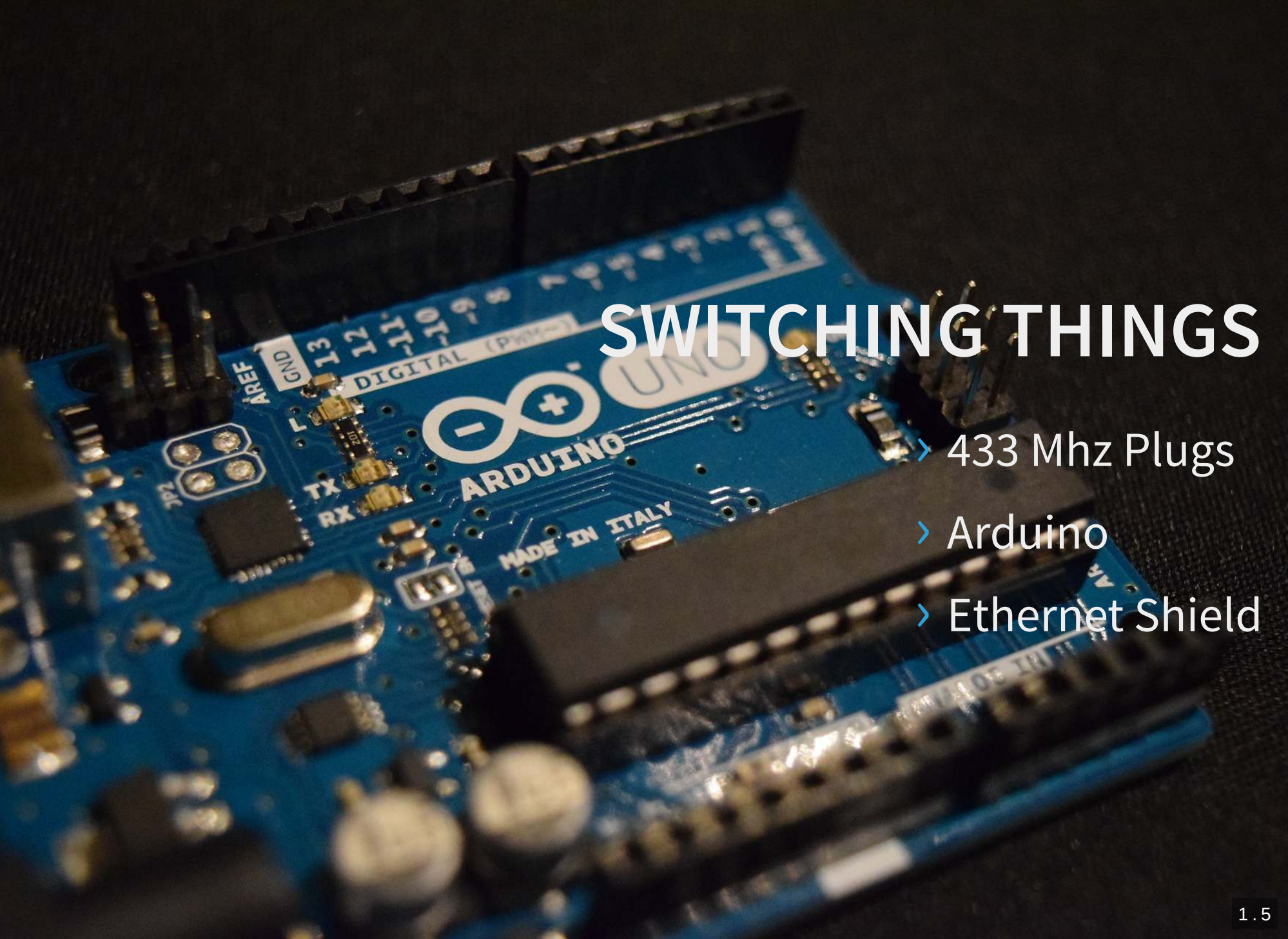
TIMES CHANGE



WHY HOME AUTOMATION?

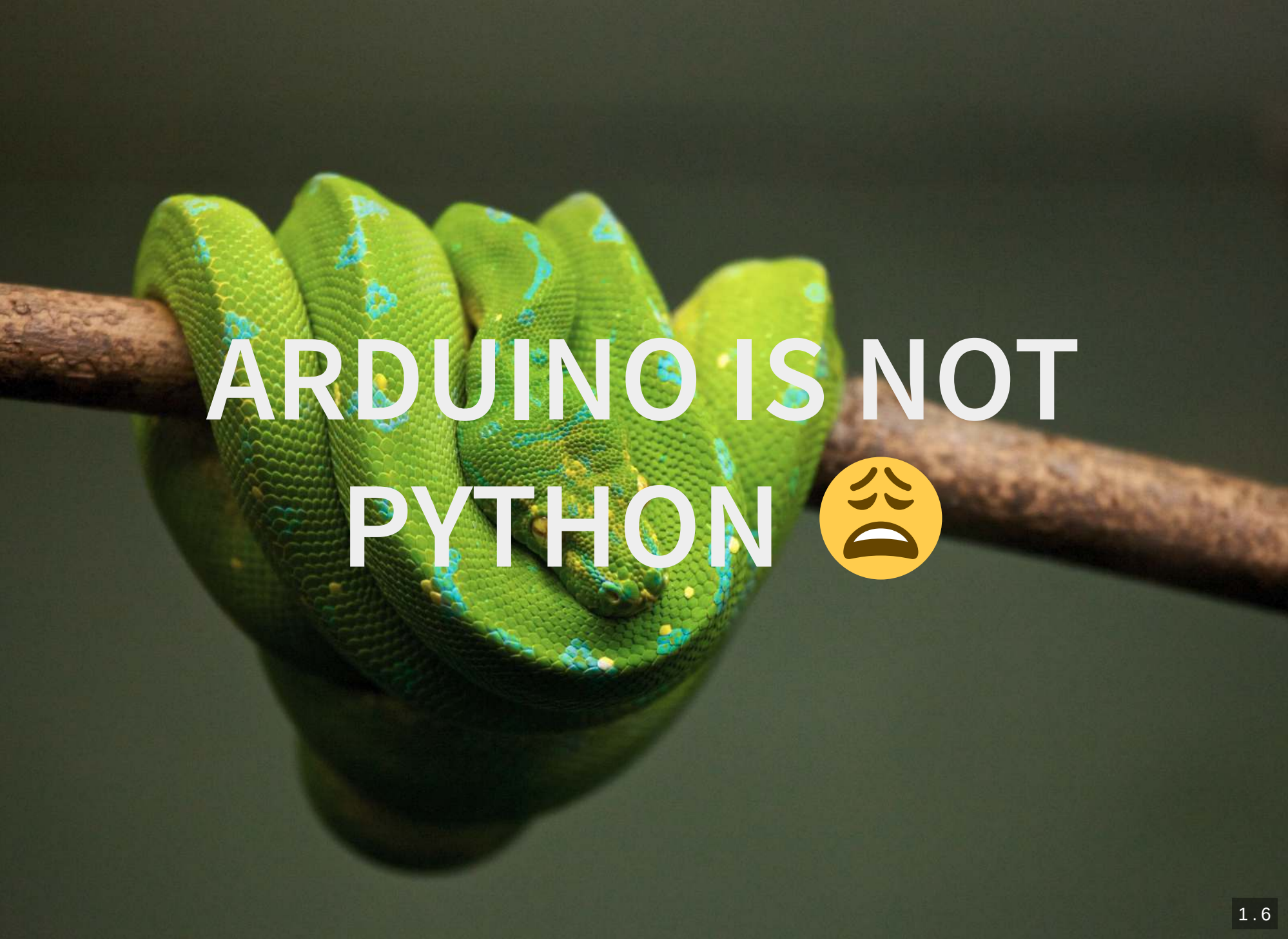
- fun & because we can 🎉
- make switching things great again 💺
- connect things together 🚪📲
- make life easier, automate! 📅

IT STARTS WITH THIS



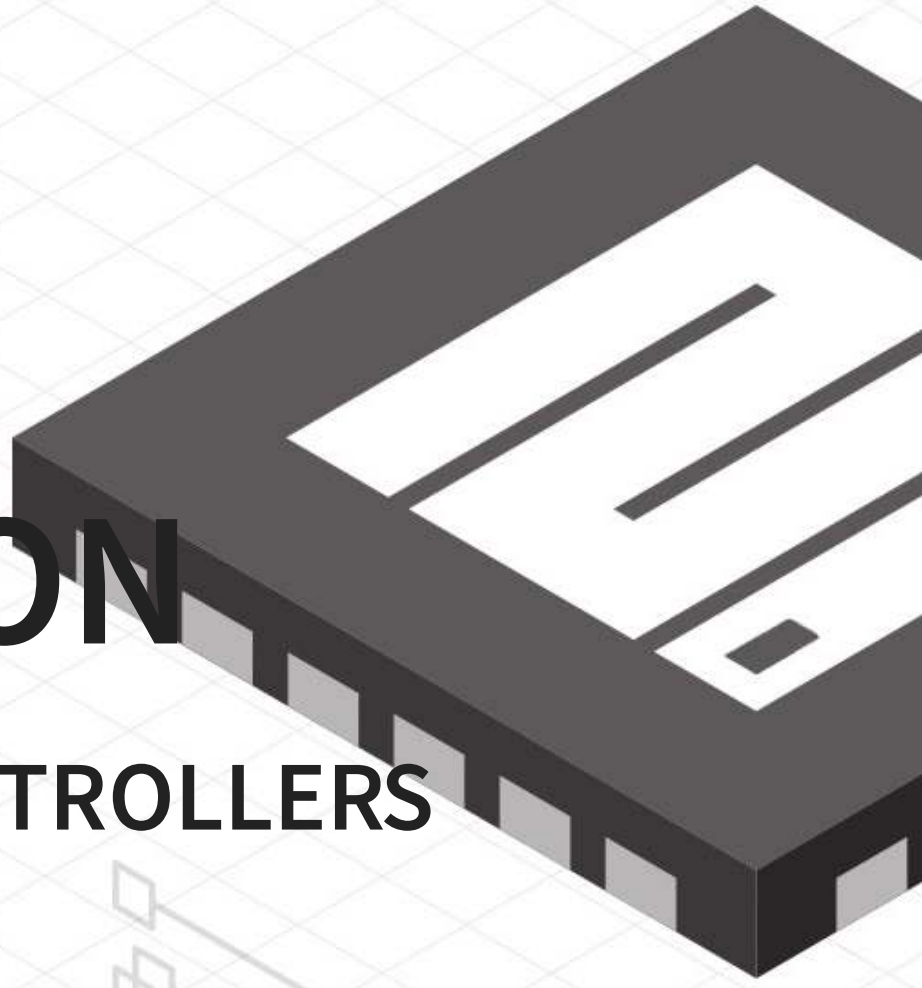
SWITCHING THINGS

- 433 Mhz Plugs
- Arduino
- Ethernet Shield


ARDUINO IS NOT
PYTHON 😫

MICROPYTHON

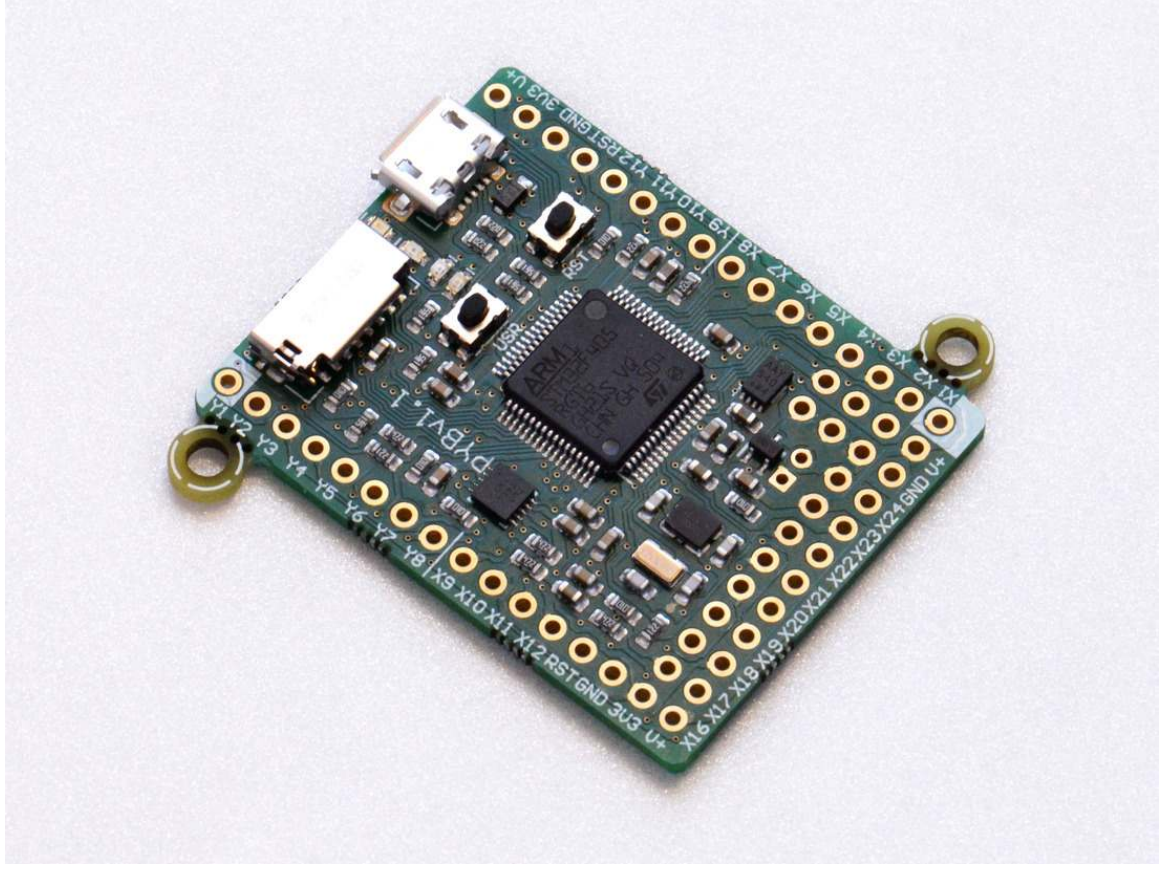
PYTHON FOR MICROCONTROLLERS



MICROPYTHON

- Kickstarter in 2014, by Damien George
  - Python 3, Open Source Hardware (micropython.org)
  - ARM chip, 192MB RAM
  - ~€33 per board

PYBOARD V1.1


PYBV1.1
USR
RST
ARM
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 RST GND 3V3 V+
Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 Y11 Y12 RST GND 3V3 V+
X16 X17 X18 X19 X20 X21 X22 X23 X24 GND V+

MICROPYTHON

- fits in 265k of code space and 16k of RAM
- small file system
- boot.py, main.py
- include other libs

MICROPYTHON

```
import pyb

# turn on an LED
pyb.LED(1).on()
# print some text to the serial console
print('Hello MicroPython!')
```

MICROPYTHON

```
from machine import Pin

# create an I/O pin in output mode
p = Pin('X1', Pin.OUT)

# toggle the pin
p.high()
p.low()
```

MICROPYTHON

```
import os

# list root directory
print(os.listdir('/'))

# print current directory
print(os.getcwd())

# open and read a file from the SD card
with open('/sd/readme.txt') as f:
    print(f.read())
```

MICROPYTHON

```
# full range of numeric types
# small integer (fits in a machine word)
>>> 123
123
# big integer
>>> 1 << 160
1461501637330902918203684832716283019655932542976
# floating point
>>> 1.23e6
1230000.0
# complex numbers
>>> (1 + 2j) * 4j
(-8+4j)
```

MICROPYTHON

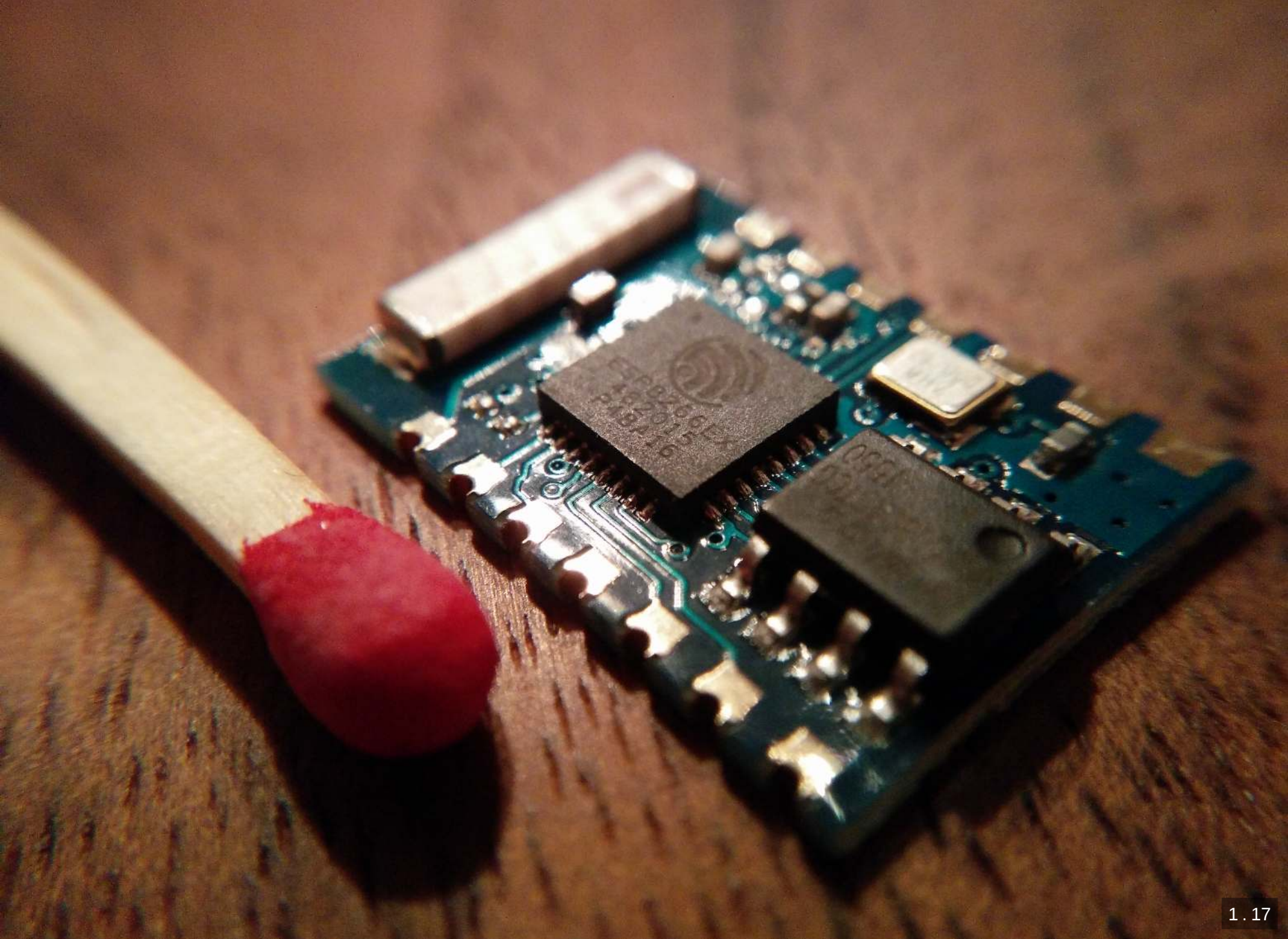
```
# MicroPython has an inline assembler
import micropython

# define a Thumb-code inline-assembler function
@micropython.asm_thumb
def asm_add(r0, r1):
    add(r0, r0, r1)

# use it as a normal Python function
total = asm_add(1, 2)
```

MICROPYTHON

- 2016, 2nd Kickstarter
  - software only
  - port of µPy to ESP8266

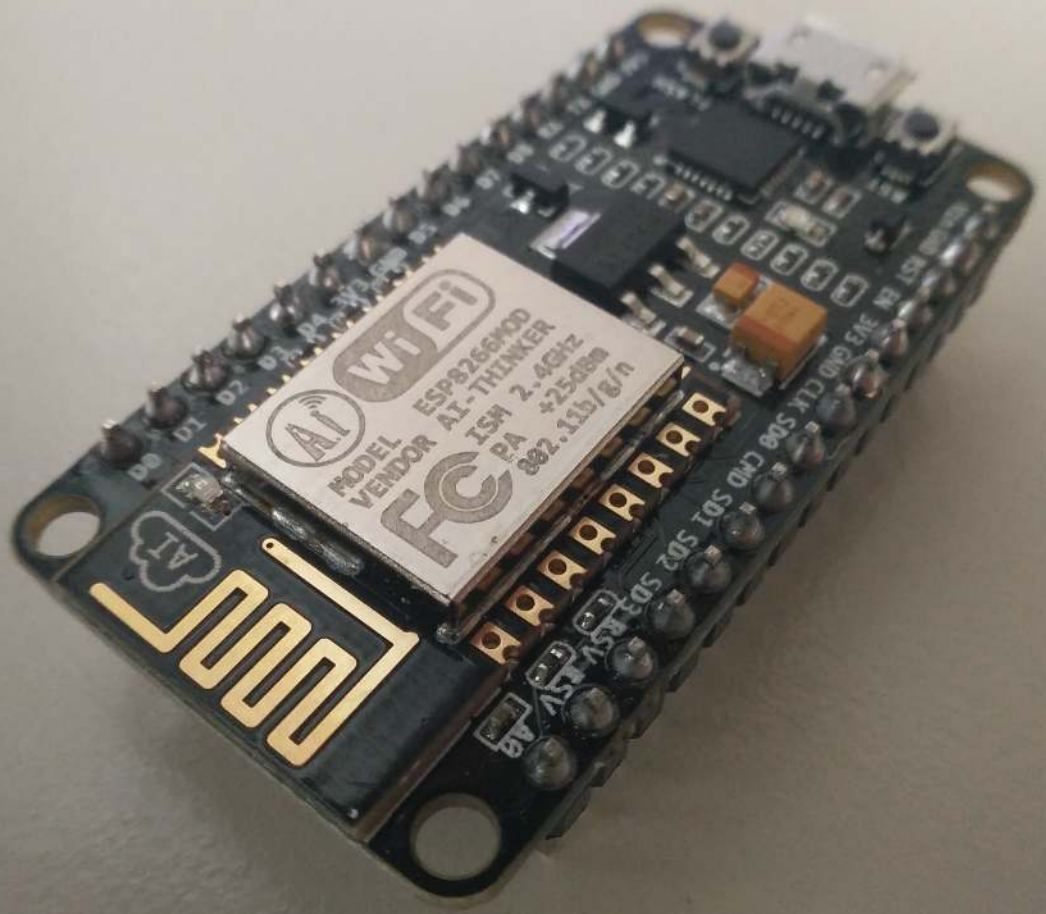


ESP8266

- WIFI enabled
- 32bit @ 80 Mhz
- 16 GPIO pins
- one 10-bit ADC
- cheap: ~$2

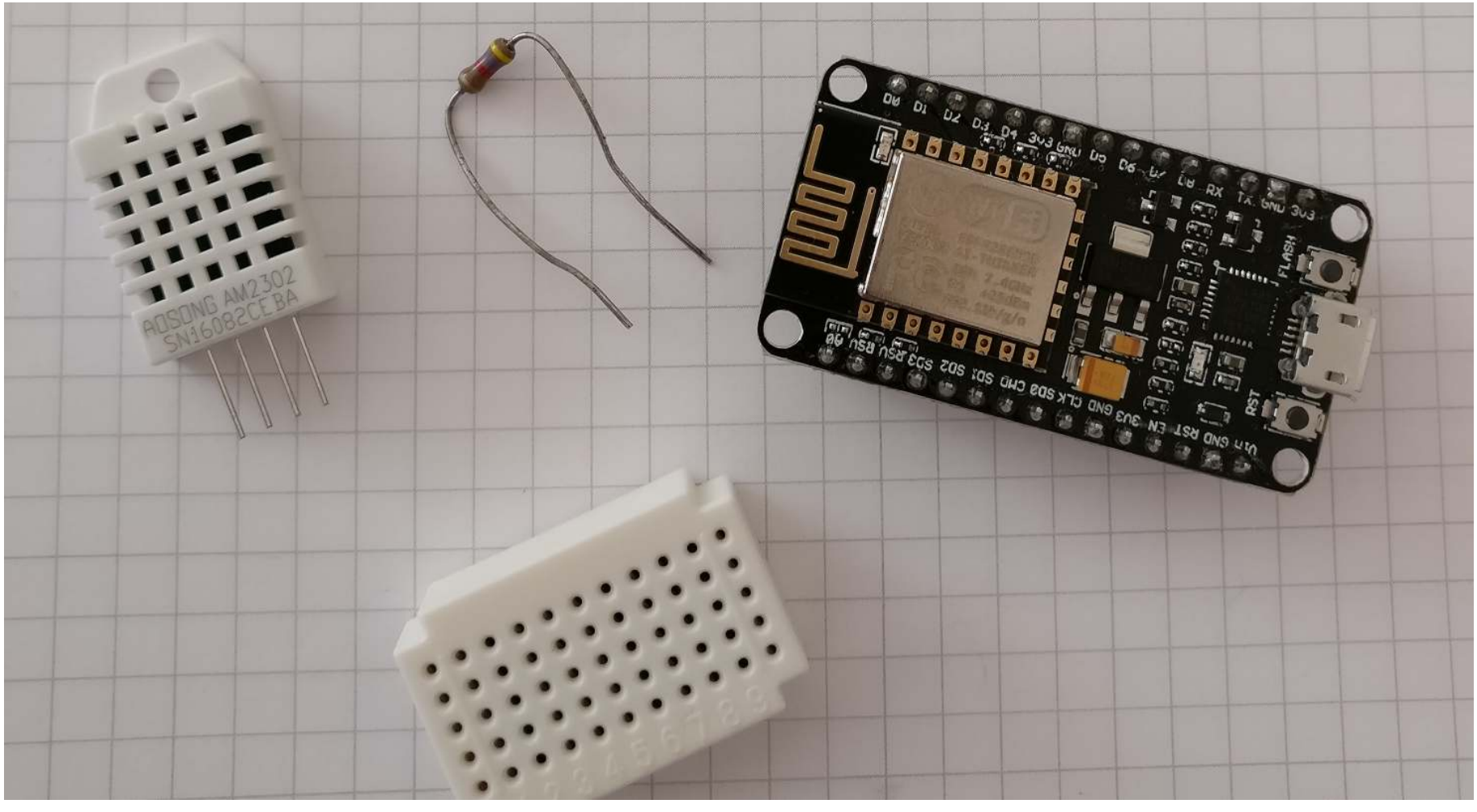
ESP8266 VERSIONS

- ESP modules:
  - ESP-12, ESP-12E: 4MiB flash memory, ESP-12F: better antenna
- others (with UART to USB):
  - NodeMCU Devboard (Lua): ~$3-4
  - WeMos D1
  - Adafruit Huzzah ESP8266

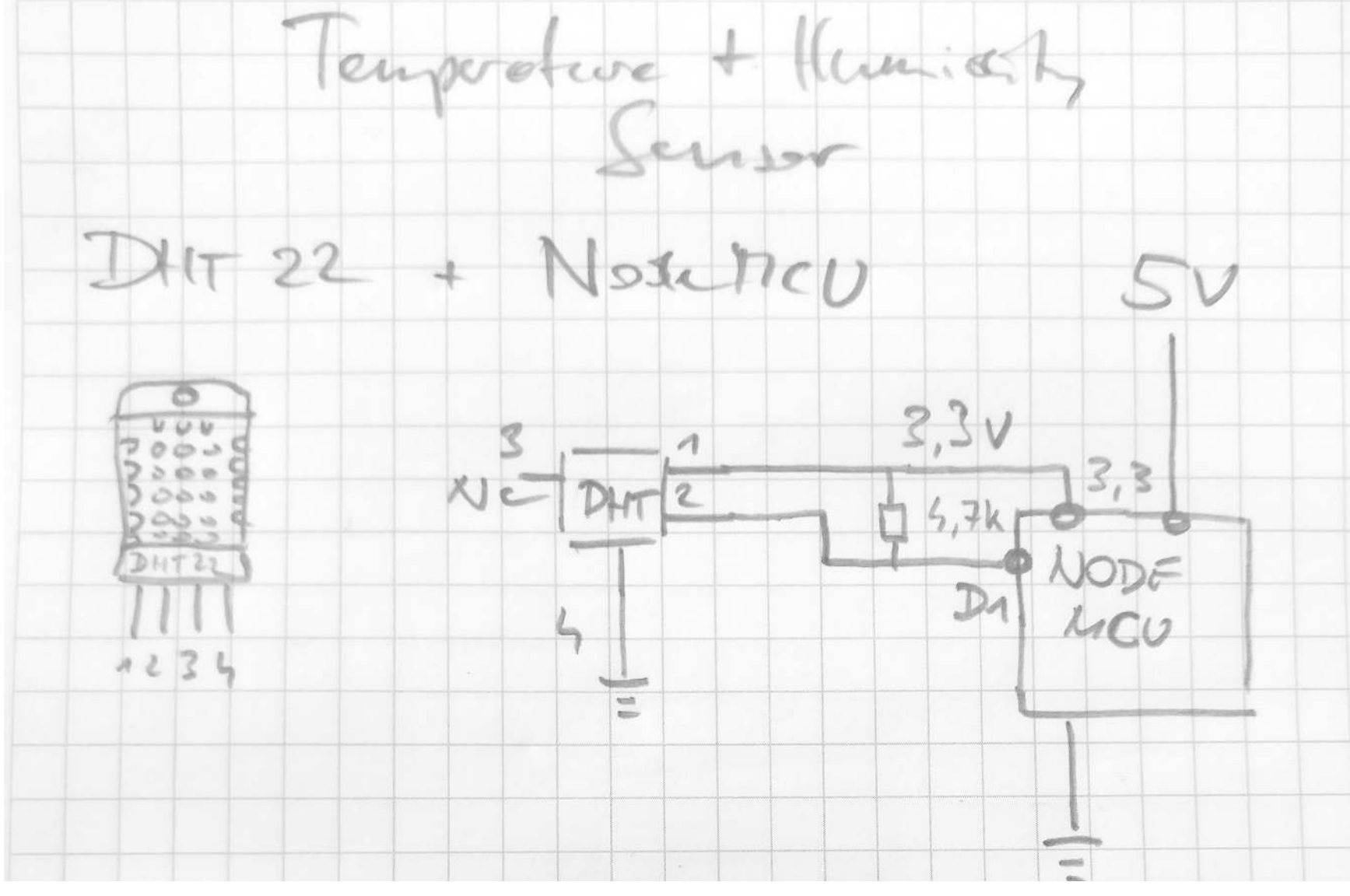

WiFi
ESP8266MOD
AI-THINKER
ISM 2.4GHz
PA +25dBm
802.11b/g/n
MODEL
VENDOR
AI
D1
D2
D3
D4
3V3
GND
CLK
SD0
CMD
SD1
SD2
SD3
RSV
A0

SENSORS EVERYWHERE!

...JUST ADD DHT22



DHT22 SENSOR



DHT22 SENSOR CODE

```
import machine
import dht
import time, json, math

d = dht.DHT22(machine.Pin(5))
led = machine.Pin(2,machine.Pin.OUT)
while True:
  led.value(0)
  try:
    d.measure()
  except Exception as e:
    print(e)
    continue
  …
```

DHT22 SENSOR CODE (2)

```
#while True:
…
  temp = math.ceil(d.temperature()*10-0.5)/10.
  humi = math.ceil(d.humidity()*10-0.5)/10.
  datat = json.dumps({
      b"id": DEV_ID,
      b"temperature": temp
  })
  datah = json.dumps({
      b"id": DEV_ID,
      b"humidity": humi
  })
```

DHT22 SENSOR CODE (3)

```
from umqtt.robust import MQTTClient
topic = "/sensor/uPy/"+ DEV_ID
c = MQTTClient(DEV_ID, mqtt_server, user=user, password=pw)
#while True:
…
  try:
    c.connect()
    c.publish(topic + "/temp",datat)
    c.publish(topic + "/humi",datah)
    c.disconnect()
  except:
    print("ERROR: connecting or sending data to MQTT server!")
  time.sleep(60)
```

MQTT

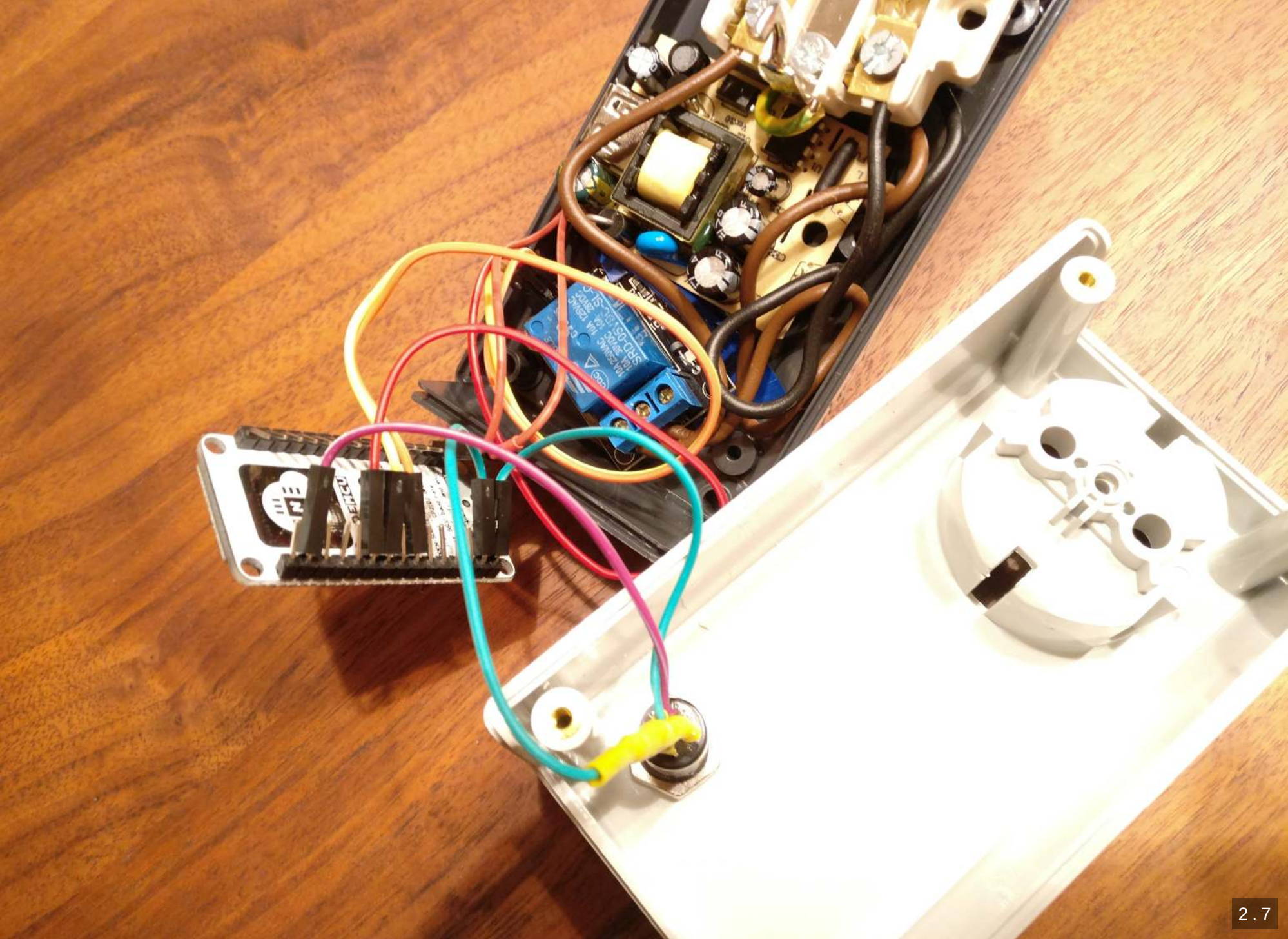
- message protocol for telemetry
- ISO/IEC PRF 20922
- subscribe/publish to topics via message broker
- for wireless networks with low bandwidth and unreliable connections

MQTT (2)

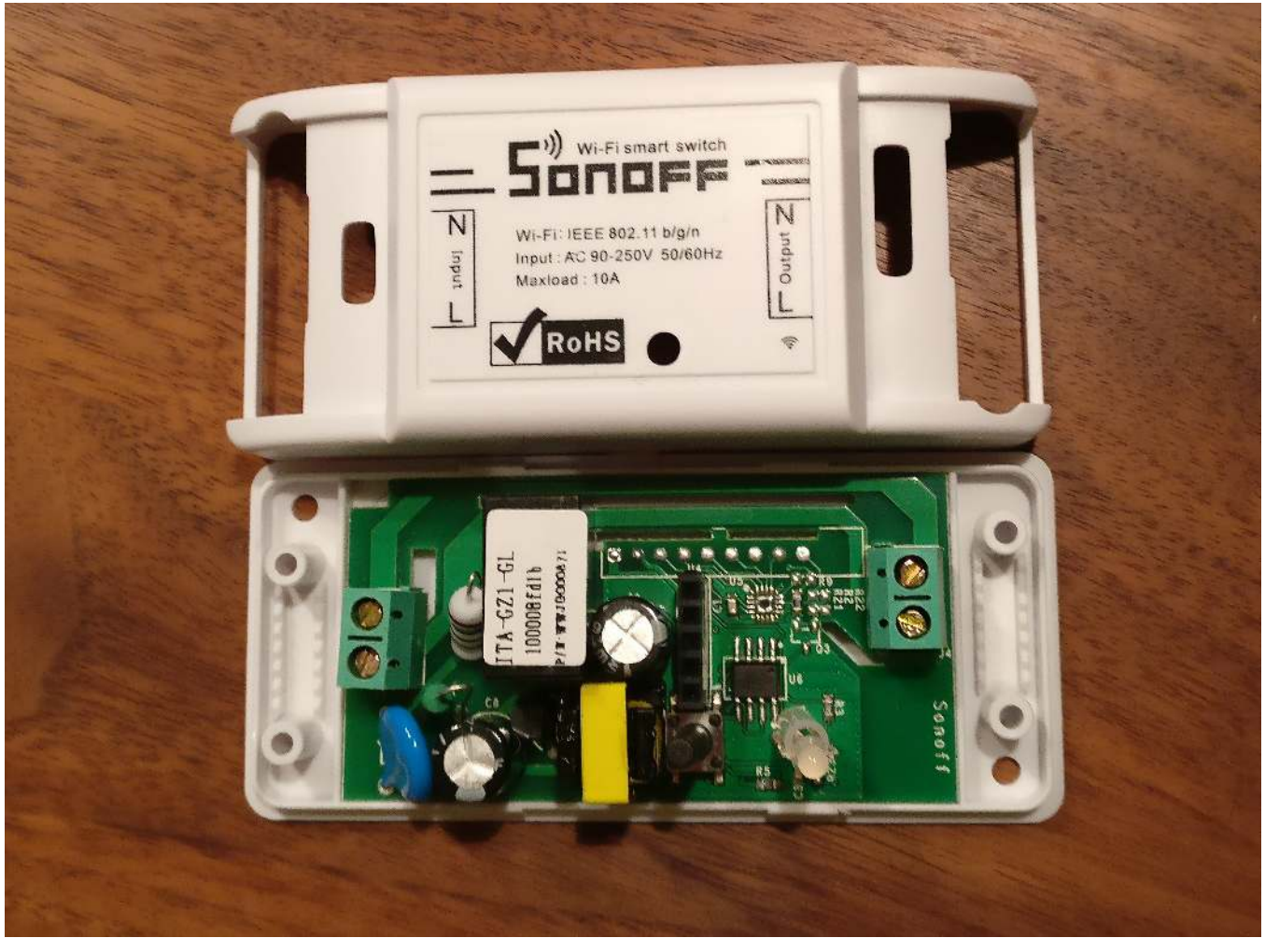
- topic: mylocation/device/sensor
  - e.g: home/livingroom/mPy_363188/temp
- last will and testament
- QoS: 0 (most once), 1 (at least once), 2 (exactly once)
- retained messages

MQTT (3)

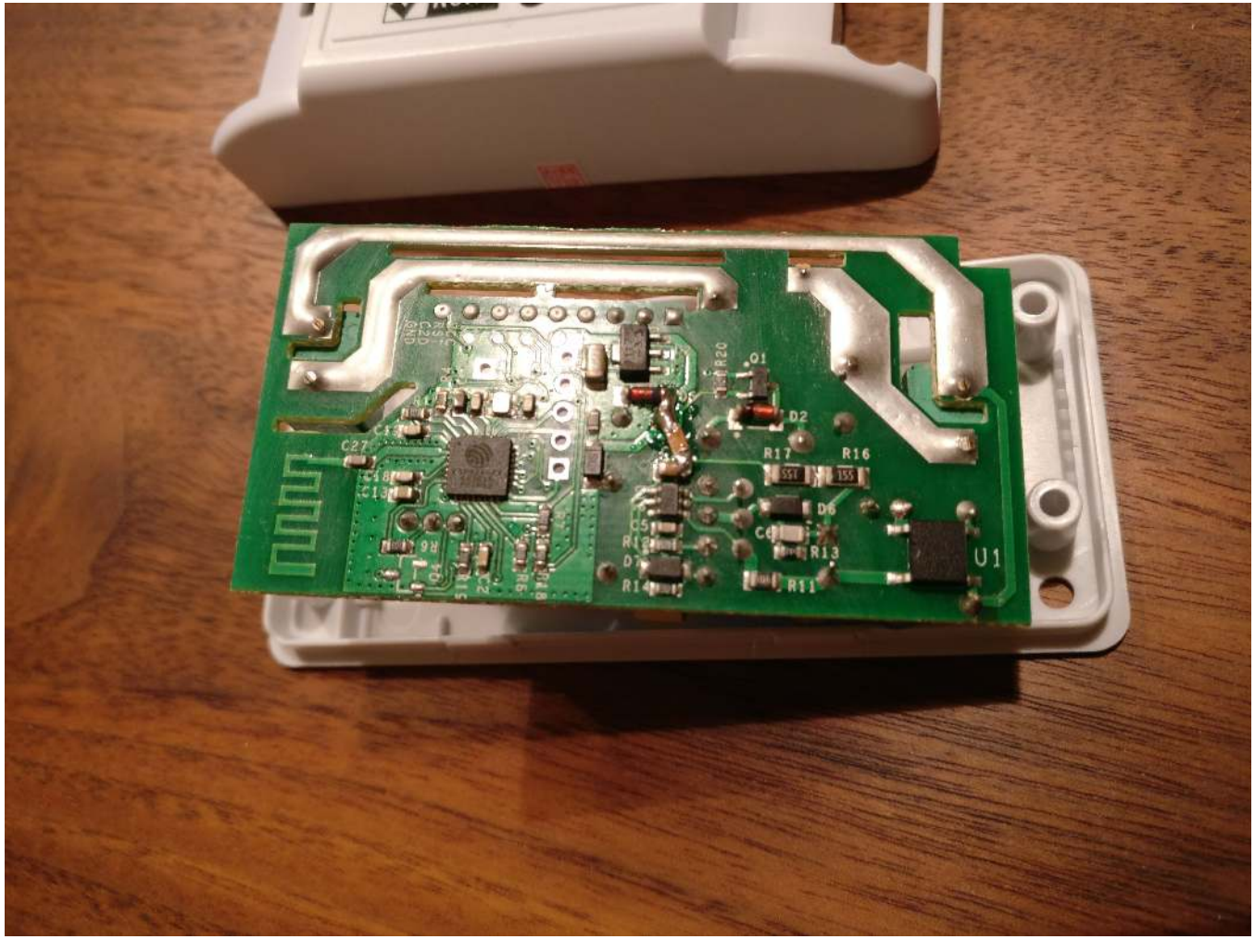
- switching devices with MQTT
- bidirectional communication
- device subscribes to topic
- publish to topic to switch device



...HACKING EXISTING ESP8266 SWITCHES



SONOFF (2)



chances are you not hacking 100% of your hardware yourself!

MY APPROACH

- be non disruptive
- 'need no handbook' approach
- fail gracefully
- components run without central control
- high WAF!!

WHAT SHOULD HOME AUTOMATION DO?

- switching lights
- controll heating
- know when you're home
- sensors for:
  - doors
  - temperature, wind
  - power

WHY FLOSS?

- no vendor lock in
- explorable, adaptable
- control your own data
- no cloud

NO CLOUD!!

Stuart Thomas @stuartthomas · 5h

Mmm. Can't turn some of my lights on at home cos @IFTTT is down. Welcome to the future!

@internetofshit

Smart Home Solver @SmartHomeS... · 2h

List of #smarthome gadgets currently down due to the #AWS outage: Ring, Nest, IFTTT, Canary, Piper. Who am I missing? #IoT #tech


14:15
Tweet
ϡ( )ϑ Jeffries
@adrjeffries
There is nothing like having your phone stolen to make you feel like an idiot for buying smart light bulbs
7:40 AM · 11 Apr 17
485 RETWEETS 1,125 LIKES
Ted Han @knowtheory · 1d
Replying to @adrjeffries
Can the thief turn your lights on and off remotely?
1
ϡ( )ϑ Jeffries @adrjeffries · 1d
it's more like, i can't turn my lights on properly
4
13
Tweet your reply

08:59

Tweet

You Retweeted

Kate Crawford
@katecrawford

Amazon just confirmed that the videos and photos recorded by the Echo Look will be stored 'indefinitely.'

Amazon Wants to Put a Camera and Microphone in Your Bedroom Now
motherboard.vice.com

11:39 PM · 26 Apr 17

769 RETWEETS **426** LIKES

Tweet your reply

HOME AUTOMATION

OPEN SOURCE SOLUTIONS

- FHEM (Perl)
- OpenHAB (Java)
- Home Assistant (Python 3)

HOME ASSISTANT

HA - ARGUMENTS (1/2)

- open Source, Python 3
- **active** community!
  - 2 week release cycle
  - chatrooms, forums, video tutorials
- write no code (unless you want to)
- Automations

HA - ARGUMENTS (2/2)

- Cross Platform (runs on Win/Mac/Linux)
- many supported IoT devices
- pretty
- local or remote
- cost[1]

WORDING

- Platforms
  - types of devices (e.g: light, sensor)
- Components
  - instances of components

  (e.g: Hue, Trådfri...)

CONFIGURATION

```
homeassistant:
  # Name of the location where Home Assistant is running
  name: Home
  # to calculate the time the sun rises and sets
  latitude: 47.0732
  longitude: 15.4420
  # Impacts weather/sunrise data
  elevation: 350
  # metric for Metric, imperial for Imperial
  unit_system: metric
  time_zone: Europe/Vienna
  customize: !include customize.yaml
…
```

639 COMPONENTS

- MQTT, Zwave, Zigbee
- Multimedia (PLEX, Sonos, Kodi, ...)
- Lights (Trådfri, Hue, Lightify,...)
- presence detection (Owntracks, openwrt etc.)
- sensors (weather, ...)
- ...

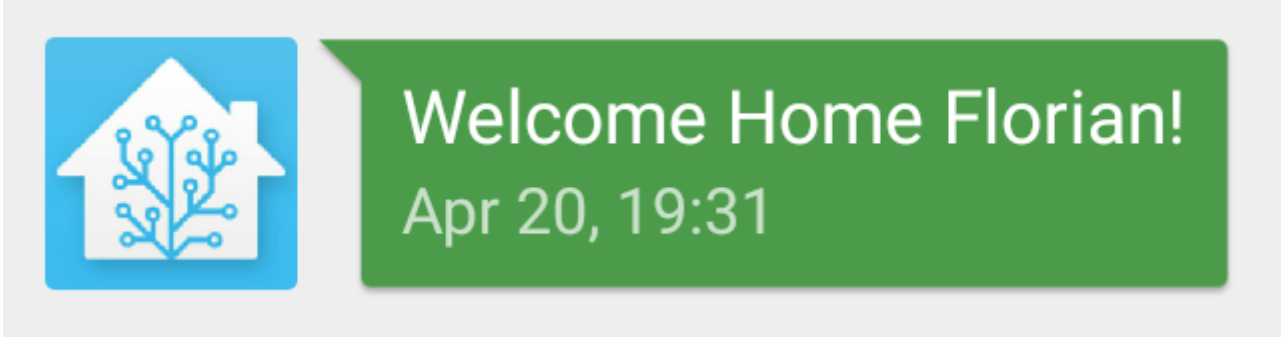

DEMO

LOCATION AWARE

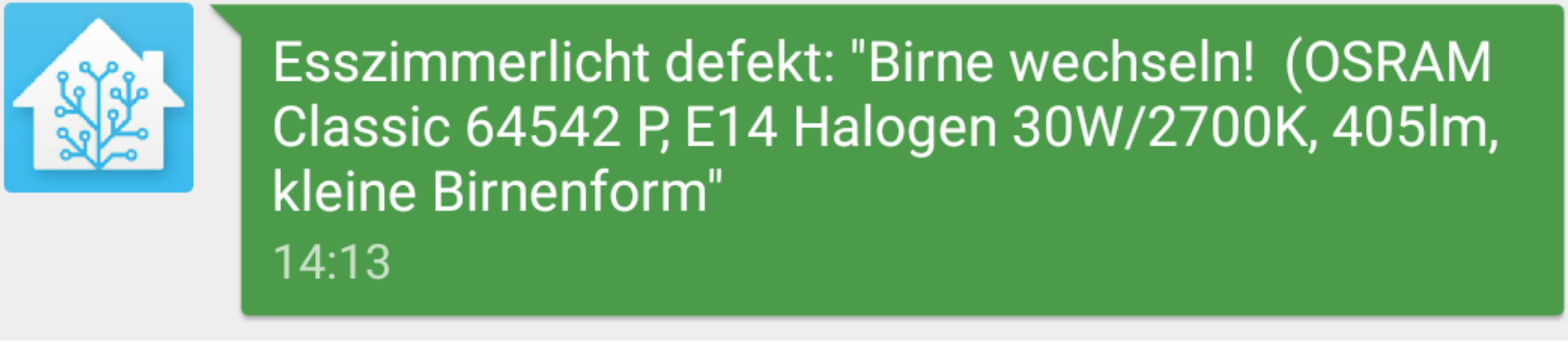
MY USE CASES

- detect when home
- switch lights
  - automated
  - scenes
- notify when bulb is broken

WELCOME HOME!



BULB BROKEN


Esszimmerlicht defekt: "Birne wechseln! (OSRAM Classic 64542 P, E14 Halogen 30W/2700K, 405lm, kleine Birnenform"
14:13

CONFIGURATION - NOTIFY

```
notify:
  - platform: xmpp
    name: jabber
    sender: !secret xmpp_user
    password: !secret xmpp_pwd
    recipient: !secret xmpp_recipient
    tls: True
```

ZWAVE

- Fibaro relay, in wall, working wall switch

AUTOMATION

```yaml
automation:
  alias: "Esstisch Lampe defekt"
  trigger:
    platform: state
    entity_id: switch.fibaro_relay_switch_2_0
    to: 'on'
  condition:
    - condition: numeric_state
      entity_id: sensor.fibaro_system_fgs223_relay_power
      below: 145
      above: 2
  action:
    - service: notify.jabber
      data:
        title: "Esszimmerlicht defekt"
        message: "Birne wechseln!"
```

LOOKOUT FOR SECURITY

IoT done right


THANKS FOR YOUR ATTENTION


@flowolf
blog.flo.cx