

INTERNET OF THINGS

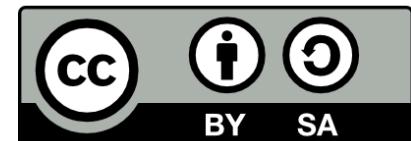


FUNDACIÓ
LACETÀPIA
Per al foment de la Formació Professional

 @rPrimTech

JAUME NOGUÉS

TBBT TBBT



Dispositius connectats

Cisco's Projections For The Internet of Things

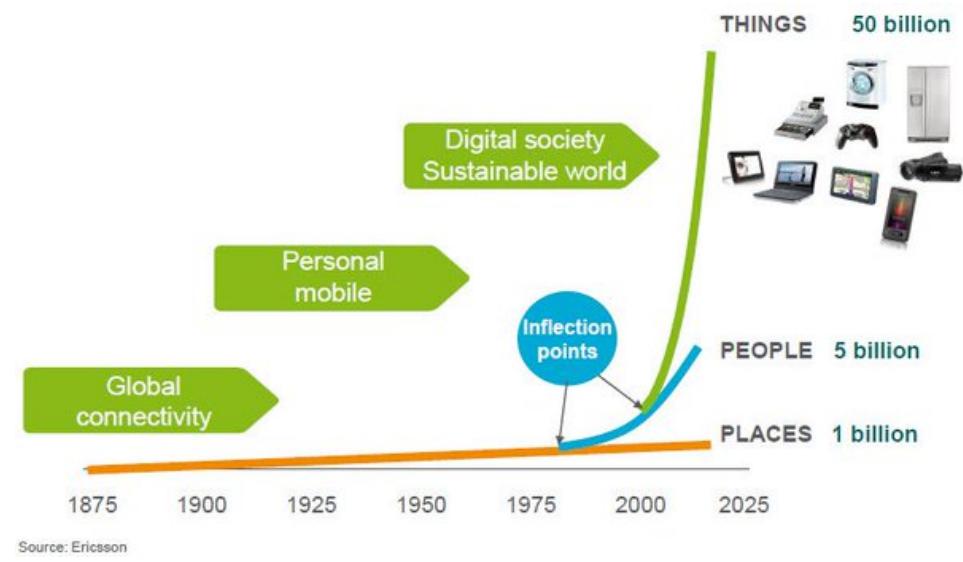
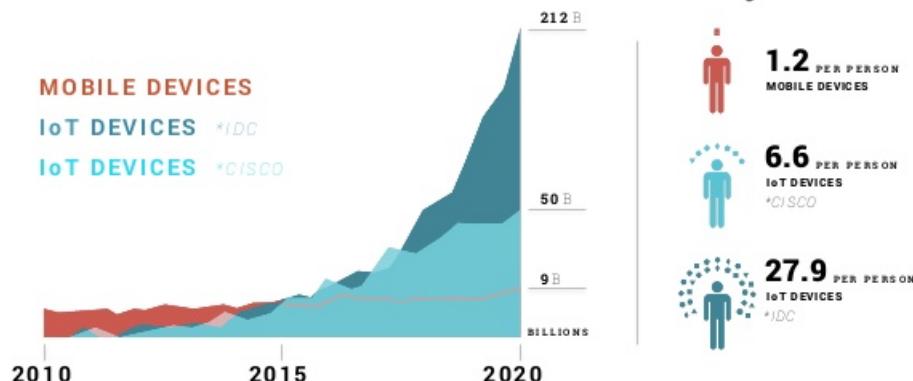


Source: Cisco



Source: Cisco IBSG, April 2011

212BB Connected Devices by 2020



Source: Ericsson

On trobem IOT

SMART HOME

SMART DEVICE

CONNECTED HOME

D2D DEVICE TO DEVICE

M2M MACHINE TO MACHINE

SMART FARM

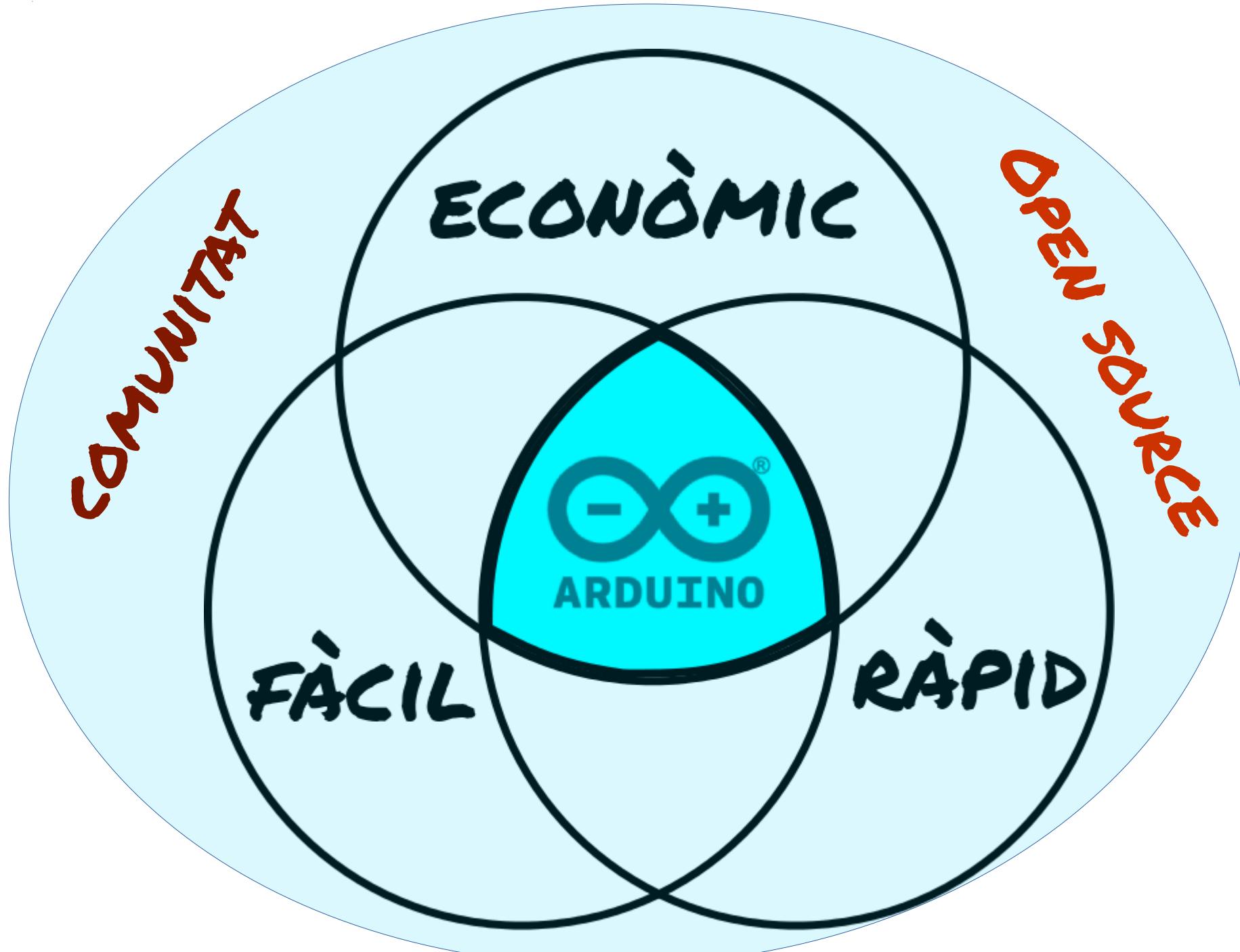
SMART FACTORY

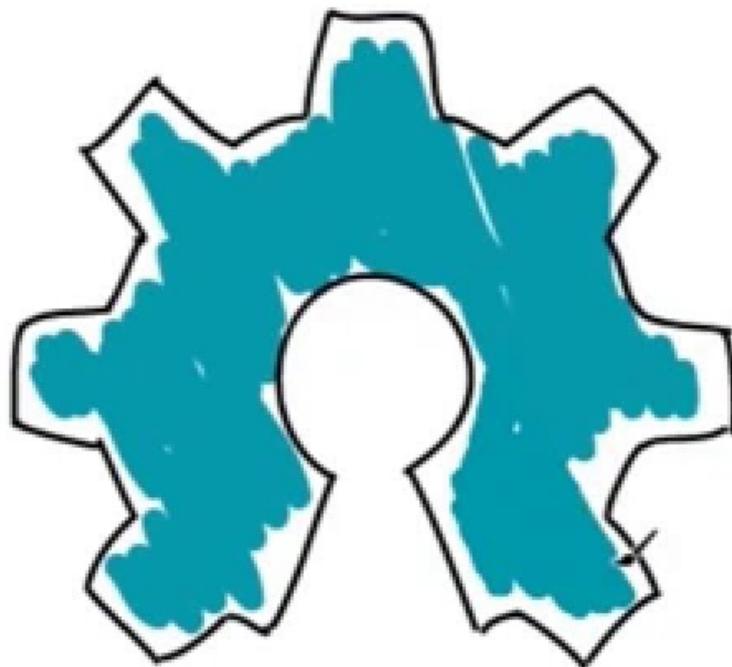
CONNECTED CAR



LIFE SIMPLIFIED WITH CONNECTED DEVICES

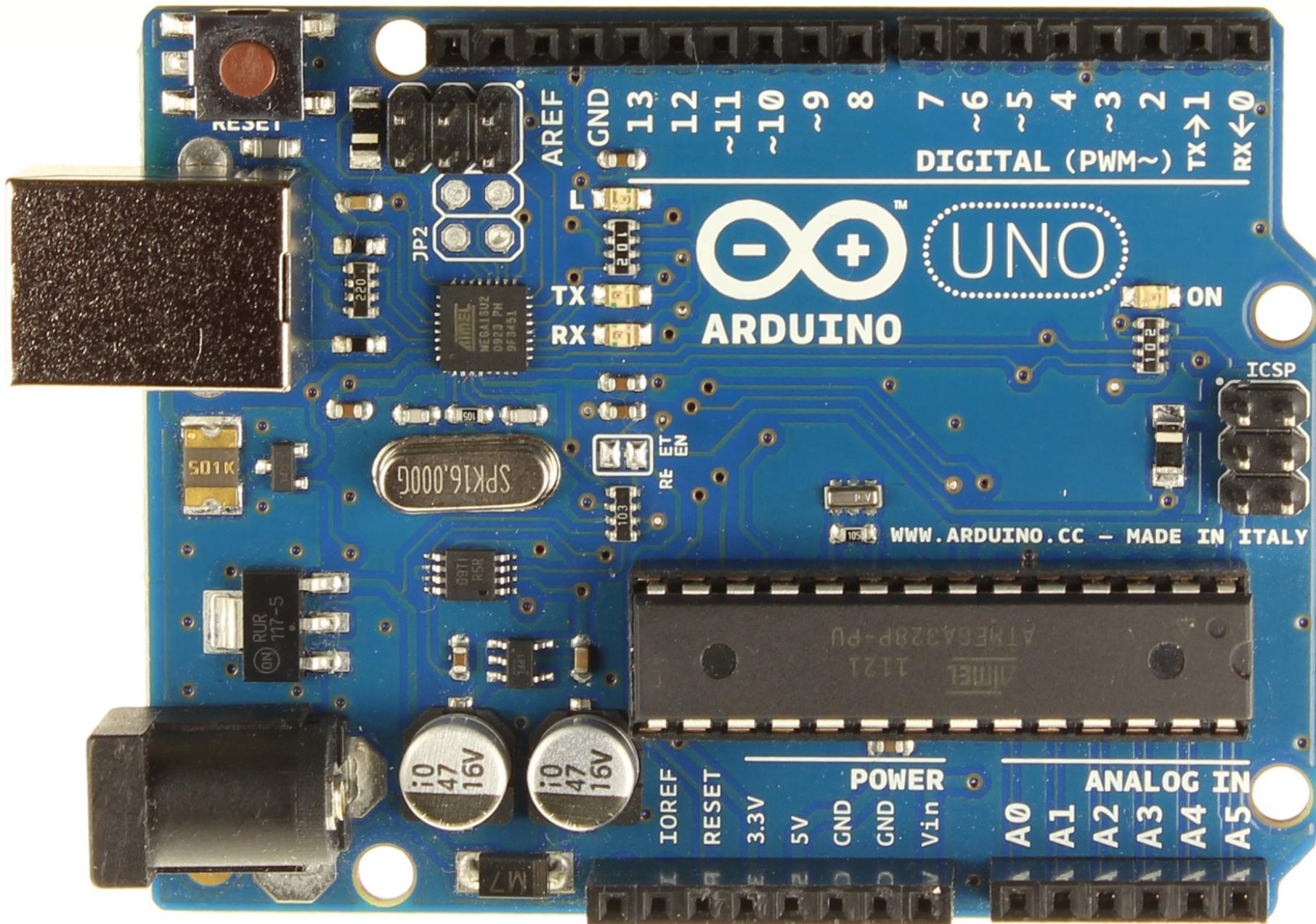
IoT Do It Yourself





open source
hardware

Arduino uno, l'inici de tot



l'equip

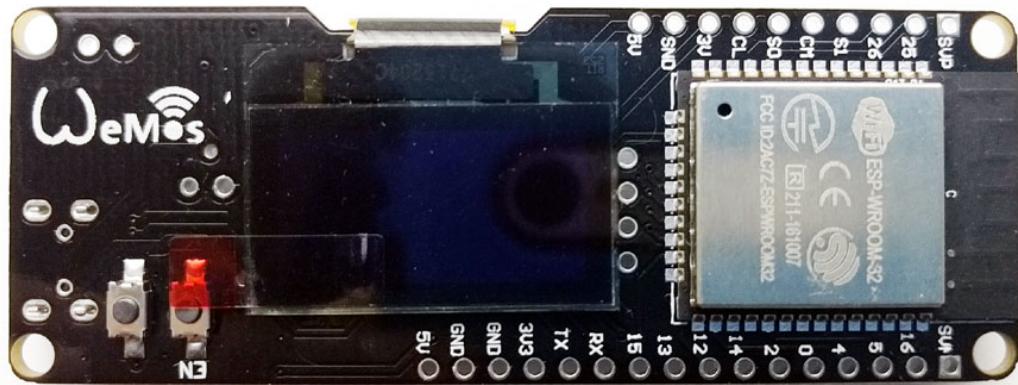
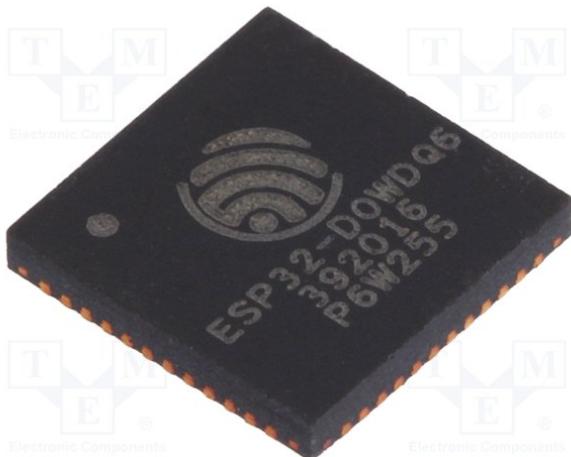
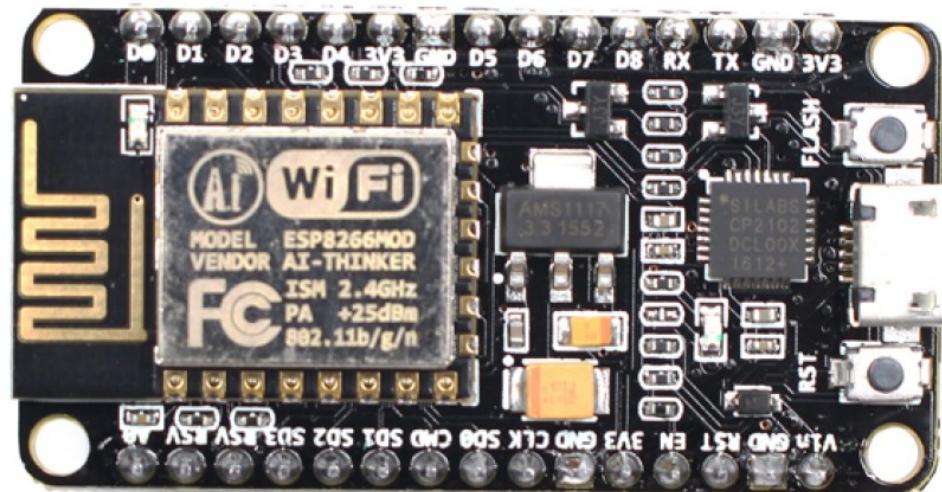
Il Team Arduino



Esp8266 / esp32, present i futur



1 €

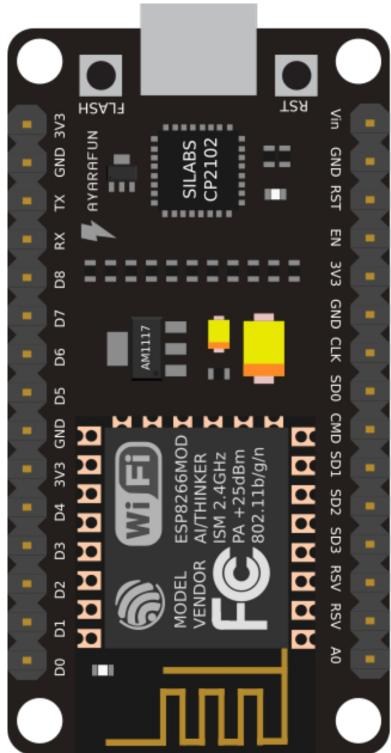


hardware

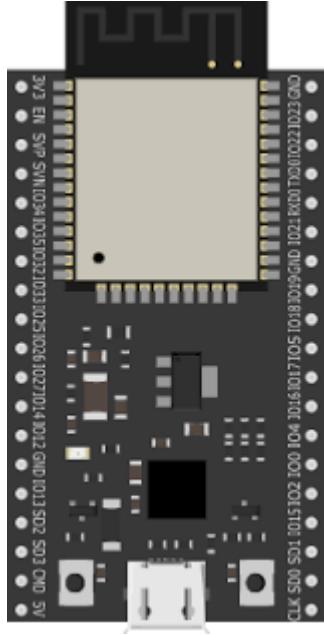


MKR WIFI 1010

NODEMCU ESP8266



ESP32

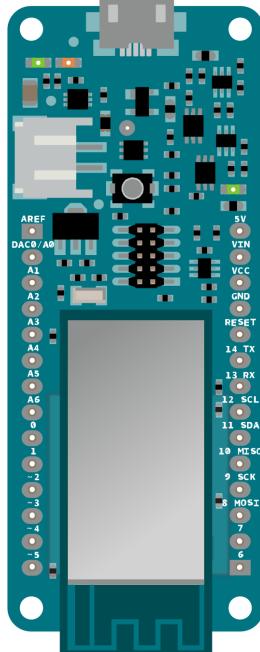


MKR WAN 1300



WHITECATBOARD

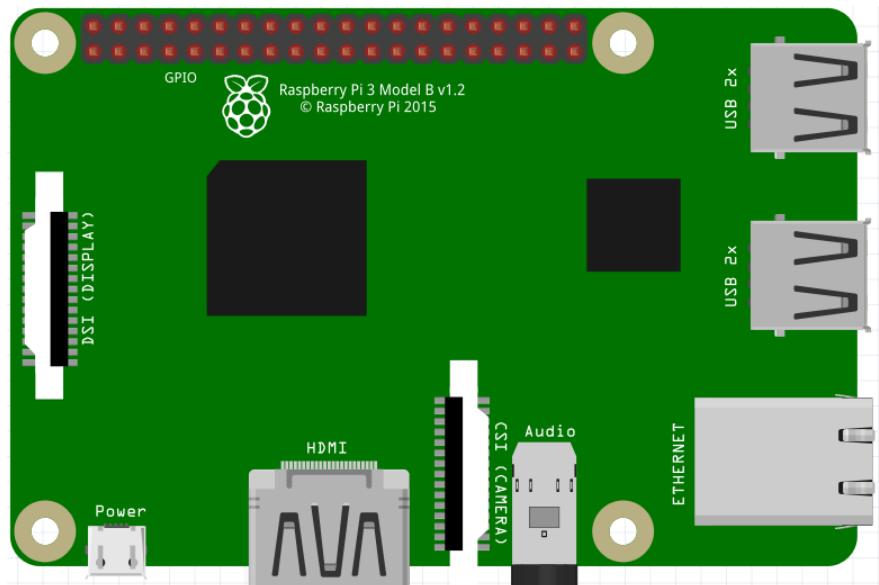
MKR1000



MKR FOX 1200



RASPBERRY PI



Tecnologies de connectivitat



NB-IoT™



ZigBee®



3G



4G



sigfox

ARDUINO IDE

Es tracta de crear una versió portable de l'IDE que sigui exportable a la resta d'ordinadors d'una aula.

0. Instal·lar els drivers USB [link](#).

1. Descarregar l'Arduino IDE 1.8.5 [link](#). La versió zip, no instal·lable.

2. Descomprimir per exemple en C:\arduino1.8.5.

3. Crear una carpeta anomenada portable: C:\arduino1.8.5\portable.

4. Llançar l'executable arduino.exe.

5. Instal·lar suport per ESP8266:

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 install esp8266 platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).

ARDUINO IDE

6. Tancar l'Arduino IDE.
7. Descarregar el contingut de la carpeta portable [link](#) .
8. Descomprimir a C:\arduinol.8.5\portable
9. Descarregar el suport per ESP32 [link](#) .
10. Descomprimir el contingut en una carpeta C:\arduinol.8.5\portable\sketchbook\hardware\espressif\esp32
11. Entrar a la carpeta C:\arduinol.8.5\portable\sketchbook\hardware\espressif\esp32\tools i executar get.exe.
12. Ara ja podem utilitzar l'IDE:-)

ARDUINO IDE

PubSubClient

SimpleTimer [link](#)

Adafruit Unified Sensor

DHT Sensor library

Dallas Temperature

OneWire

Blynk [link](#)

Settimino [link](#)

Justwifi V2* [link](#)

Time [link](#)

Adafruit BME280

ArduinoJson 5.13.12*

Embedis

IRRemoteESP8266

AsyncMqttClient [link](#)

DebounceEvent [link](#)

ESPAsyncTCP [link](#)

ESPAsyncWebServer [link](#)

ESPSoftwareSerial [link](#)

MDNSResolver [link](#)

Embedis

NoFUSS [link](#)

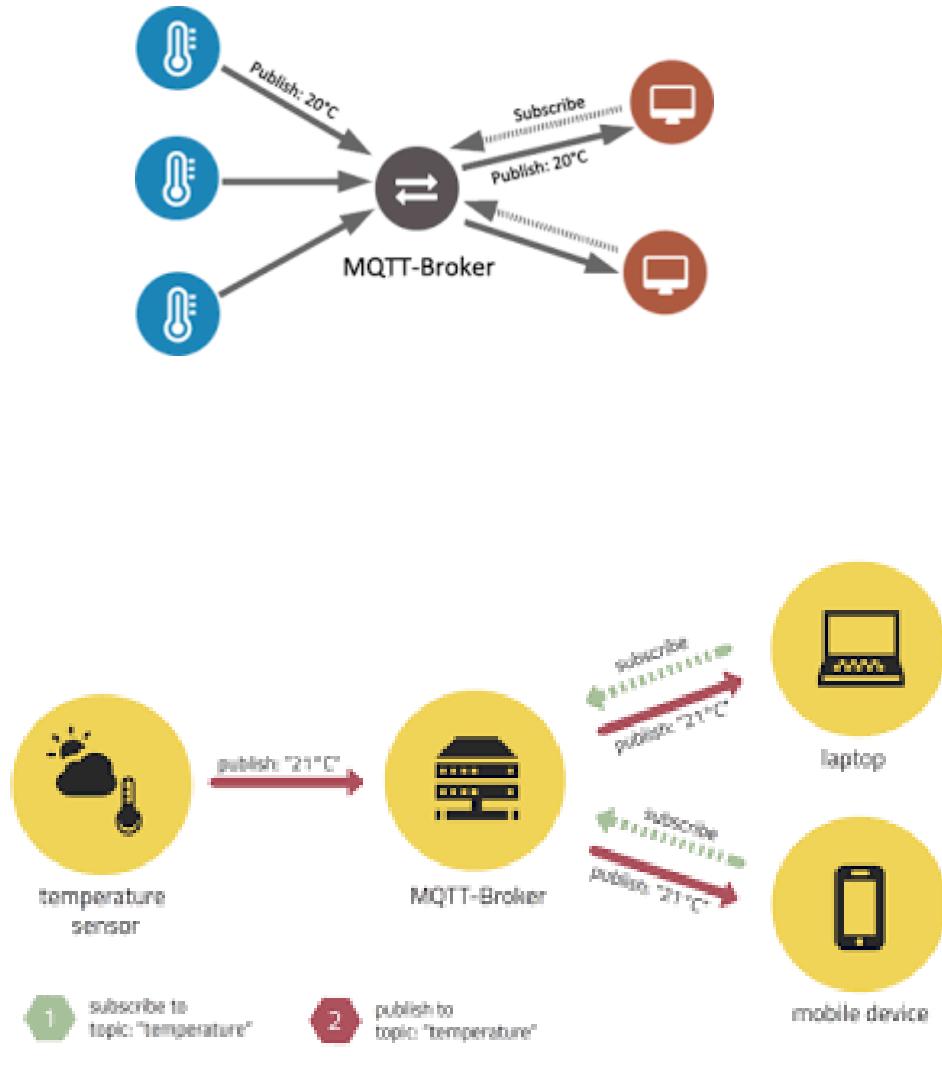
NtpClientLib [link](#)

Oled [link](#)

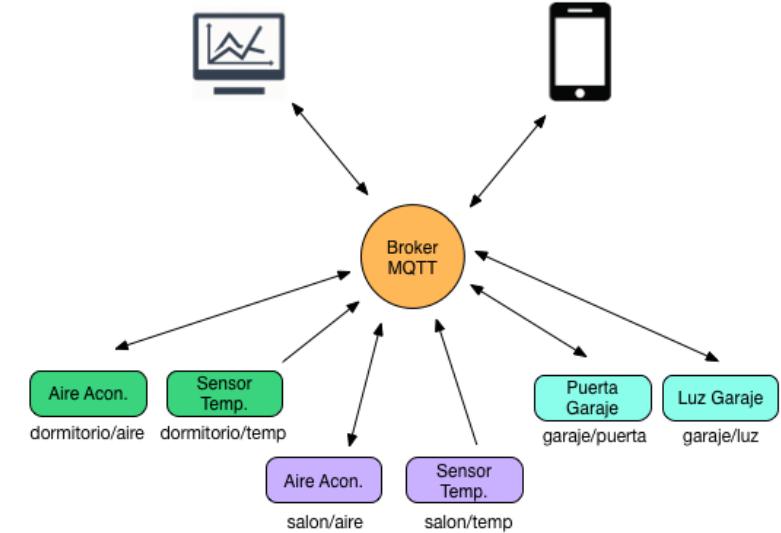
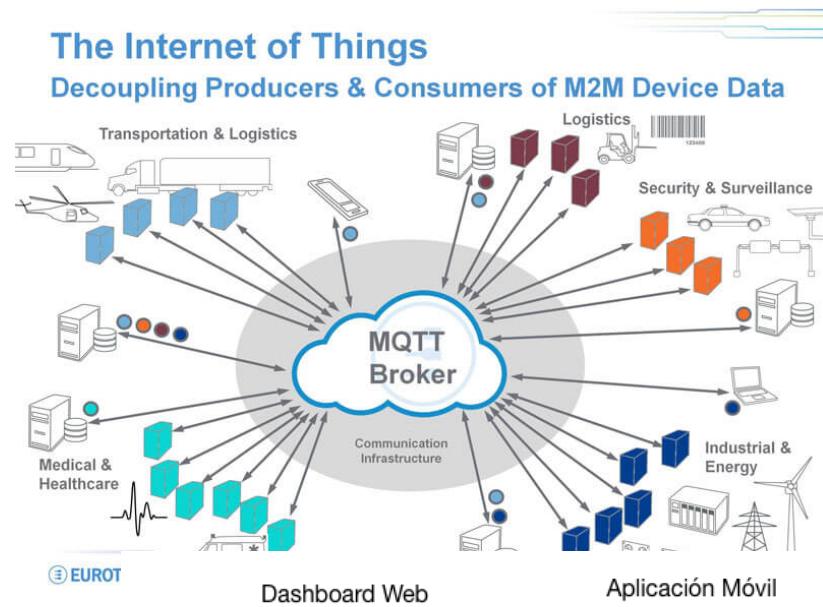
eeprom_rotate [link](#)

MQTT

Message Queuing Telemetry Transport



The Internet of Things
Decoupling Producers & Consumers of M2M Device Data



MQTTmosquitto broker



Instalar mosquitto:

- Windows [link](#)

- Raspberry Pi:

```
sudo apt-get update
```

```
sudo apt-get install mosquitto
```

```
sudo apt-get install mosquitto-clients
```

·Ubuntu/Mint/Debian:

```
sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
```

```
sudo apt-get update
```

```
sudo apt-get install mosquitto mosquitto-clients
```

Brokers lliures:

iot.eclipse.org test.mosquitto.org

RPi3 192.168.100.17

Rpi2 192.168.100.222

MQTTmosquitto broker



Iniciar mosquitto en W7/W10:

C:\mosquitto\mosquitto -v

Parar o iniciar mosquitto com servei en linux:

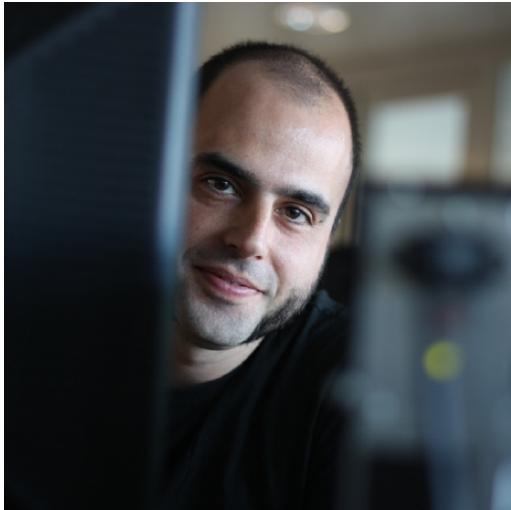
```
sudo service mosquitto stop  
sudo service mosquitto start  
sudo service mosquitto status
```

Client mqtt per Chrome [link](#)

mosquitto-clients

```
mosquitto_sub -v -h iot.eclipse.org -t /lacetania/#  
mosquitto_pub -h iot.eclipse.org -t /lacetania/juan -m Hola  
mosquito_sub -v -u pi -P raspberry -h 192.168.100.222 -t /lacetania/juan - m Hola
```

ESPurna, client mqtt esp8266



Creat per Xosé Pérez ([thinkerman](#))

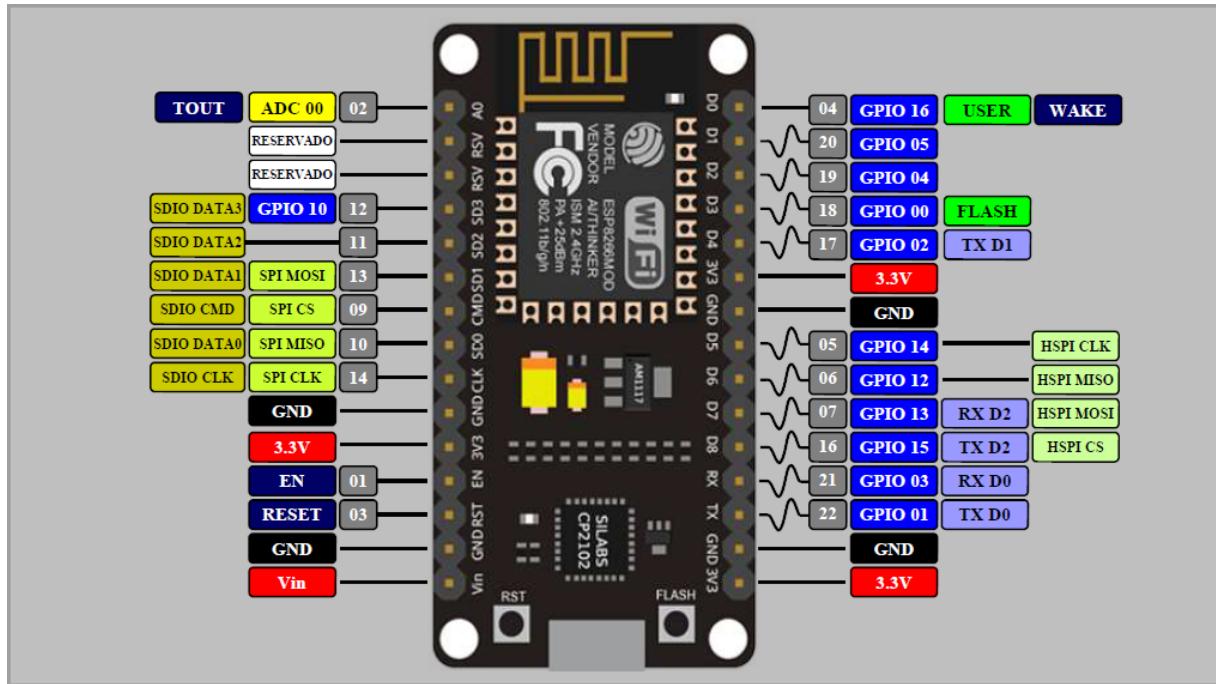
GitHub [link](#)

Interessant escoltar aquest podcast
[link](#) per conèixer el projecte.

Inicialment pensat pels dispositius Sonoff de ITEAD [link](#). Adaptable a multitud de sistemes basats en el esp8266.

- Wiki [link](#)
- Dispositius suportats [link](#)

ESPurna en la nodeMCU



Sketch per Arduino IDE [link](#)

Cal assegurar-nos de tenir instal·lades totes
aquestes llibreries [link](#)

ESPurna en la nodeMCU

- I. Baixat espurna-1.13.0.
 2. Descomprimeix-lo en qualsevol ubicació, p.ex. C:\kk\
 3. Obre des de l'Arduino IDE l'sketch C:\kk\espurna-1.13.0\code\espurna\espurna.ino
 4. Compila i puja per la placa nodeMCU I.O
 5. Un cop pujat el firmware, obre el serial monitor, a 115200 bps i Ambdos NL & CR.
 6. Configura via serial el nou password d'administrador: **set adminPass Melda2018**
 7. Configura les credencials wifi ssid: **set ssid0 IoT-2** password: **set pass0 pitufito**
 8. Reseteja el wifi de l'esp8266: **reset.wifi**
 9. Observa al serial monitor la IP assignada pel router a la nodeMCU, p.ex 192.168.100.72
 10. Mitjançant el navegador apunta a aquesta IP. Accedeix amb l'usuari **admin** i el password **Melda2018**
- II. Ara ja pots acabar de configurar la nodeMCU mitjançant l'interfície web.

Recordeu que l'unic fitxer que es pot modificar (ojo) és el: **C:\kk\espurna-1.13.0\code\espurna\config\hardware.h**

ESPurna, hardware.h

```
// -----
// Development boards
// -----  
  
#elif defined(NODEMCU_LOLIN)  
  
// Info  
#define MANUFACTURER      "NODEMCU"  
#define DEVICE            "LOLIN"  
  
// Buttons  
//  
#define BUTTON1_PIN        5  
#define BUTTON1_MODE       BUTTON_PUSHBUTTON //| BUTTON_DEFAULT_HIGH  
#define BUTTON1_PRESS      BUTTON_MODE_NONE  
#define BUTTON1_CLICK      BUTTON_MODE_TOGGLE //| BUTTON_MODE_ON //| BUTTON_MODE_OFF  
#define BUTTON1_DBCLICK    BUTTON_MODE_NONE //| BUTTON_MODE_AP  
#define BUTTON1_LNGCLICK   BUTTON_MODE_NONE //| BUTTON_MODE_RESET  
#define BUTTON1_LNGLNGCLICK BUTTON_MODE_NONE //| BUTTON_MODE_FACTORY  
#define BUTTON1_RELAY      1  
  
#define BUTTON2_PIN        4  
#define BUTTON2_MODE       BUTTON_PUSHBUTTON //| BUTTON_DEFAULT_HIGH  
#define BUTTON2_RELAY      2  
  
#define BUTTON3_PIN        14  
#define BUTTON3_MODE       BUTTON_PUSHBUTTON //| BUTTON_DEFAULT_HIGH  
  
#define BUTTON4_PIN        12  
#define BUTTON4_MODE       BUTTON_PUSHBUTTON //| BUTTON_DEFAULT_HIGH  
  
//PIR  
#define BUTTON5_PIN        10  
#define BUTTON5_MODE       BUTTON_PUSHBUTTON | BUTTON_DEFAULT_HIGH | BUTTON_SET_PULLUP  
#define BUTTON5_CLICK      BUTTON_MODE_TOGGLE  
#define BUTTON5_DBCLICK    BUTTON_MODE_NONE //| BUTTON_MODE_AP  
#define BUTTON5_LNGCLICK   BUTTON_MODE_NONE //| BUTTON_MODE_RESET  
#define BUTTON5_LNGLNGCLICK BUTTON_MODE_NONE //| BUTTON_MODE_FACTORY  
#define BUTTON5_RELAY      4  
  
// Relays  
#define RELAY1_PIN          15  
#define RELAY1_TYPE         RELAY_TYPE_NORMAL  
#define RELAY2_PIN          13  
#define RELAY2_TYPE         RELAY_TYPE_NORMAL  
#define RELAY3_PIN          2  
#define RELAY3_TYPE         RELAY_TYPE_INVERSE  
#define RELAY4_PIN          0  
#define RELAY4_TYPE         RELAY_TYPE_INVERSE
```

ESPurna, hardware.h

```
// LEDs
#define LED1_PIN          16
#define LED1_PIN_INVERSE   1

// DB18B20//if (gpio == 10) return true;//Jaume Nogués al gpio.ino
#define DALLAS_SUPPORT      0
#define DALLAS_PIN           10
#define DALLAS_UPDATE_INTERVAL 5000
#define TEMPERATURE_MIN_CHANGE 1.0

//DHT-11/22
#define DHT_SUPPORT         0
#define DHT_PIN              10
#define DHT_TYPE             DHT_CHIP_DHT11 //DHT_CHIP_DHT22

// BME280/BMP280
#define BMX280_SUPPORT       0
#define I2C_SDA_PIN          12//SDA
#define I2C_SCL_PIN          14//SCL
#define BMX280_ADDRESS        0x76    // 0x00 means auto
#define BMX280_MODE            1      // 0 for sleep mode, 1 or 2 for forced mode, 3 for normal mode
#define BMX280_STANDBY         0      // 0 for 0.5ms, 1 for 62.5ms, 2 for 125ms
                                         // 3 for 250ms, 4 for 500ms, 5 for 1000ms
                                         // 6 for 10ms, 7 for 20ms
#define BMX280_FILTER          0    // 0 for OFF, 1 for 2 values, 2 for 4 values, 3 for 8 values and 4 for 16 values
#define BMX280_TEMPERATURE     1    // Oversampling for temperature (set to 0 to disable magnitude)
#define BMX280_HUMIDITY         1    // Oversampling for humidity (set to 0 to disable magnitude, only for BME280)
#define BMX280_PRESSURE         1    // Oversampling for pressure (set to 0 to disable magnitude)

//Analog input
#define ANALOG_SUPPORT        1
#define ADC_VCC_ENABLED        1
```

Alternativa a ESPurna amb ESP8266

Si volem utilitzar un sketch menys complexe, podem fer servir aquest [link](#).

Cal descomentar algunes línies segons els perifèrics que tinguem connectats, DHT11/DHT22, DS18B20, BME280 o NEOPIXEL.

Dins del codi podem veure els diferents comandaments per interactuar amb el hardware, Per exemple q15on, q15off, q15tog, q15?, ...

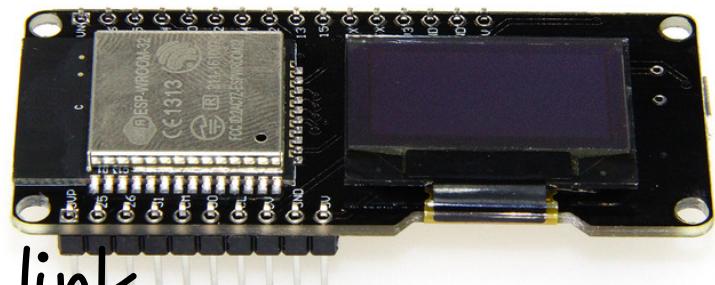
Alternativa a ESPurna amb ESP32

Utilitzar l'ESP32 encara és una mica experimental, cal no desanimar-se. Sembla que en poc temps hi haurà el plugin per l'Arduino IDE. De moment la instal·lació és una versió beta [link](#).

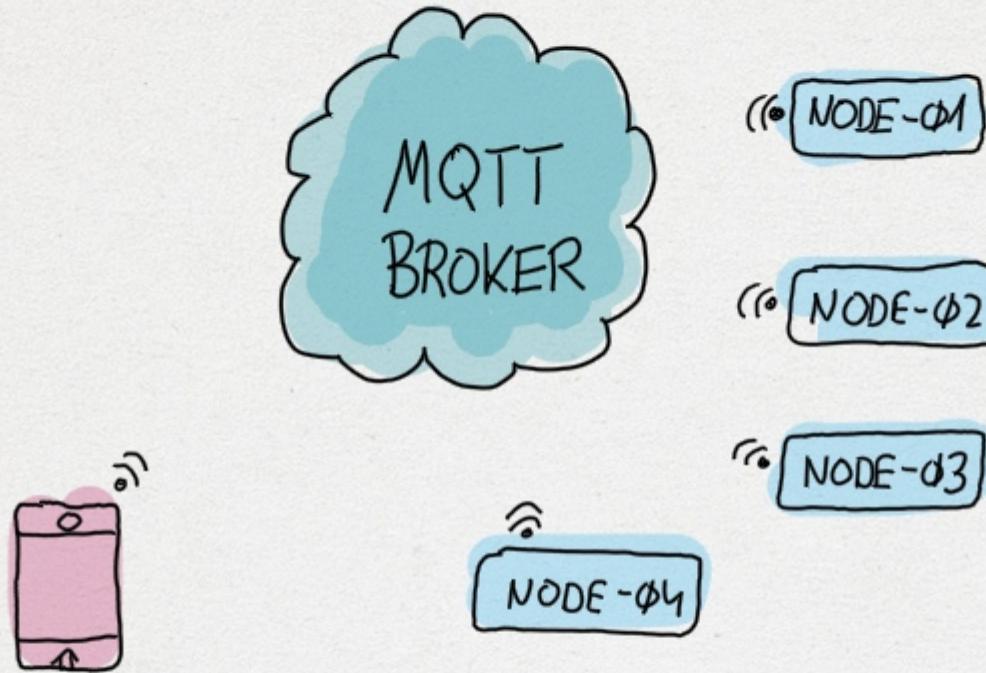
Hi ha multitud de formats de l'ESP32. Cap d'ells sembla que triomfi per damunt dels altres.

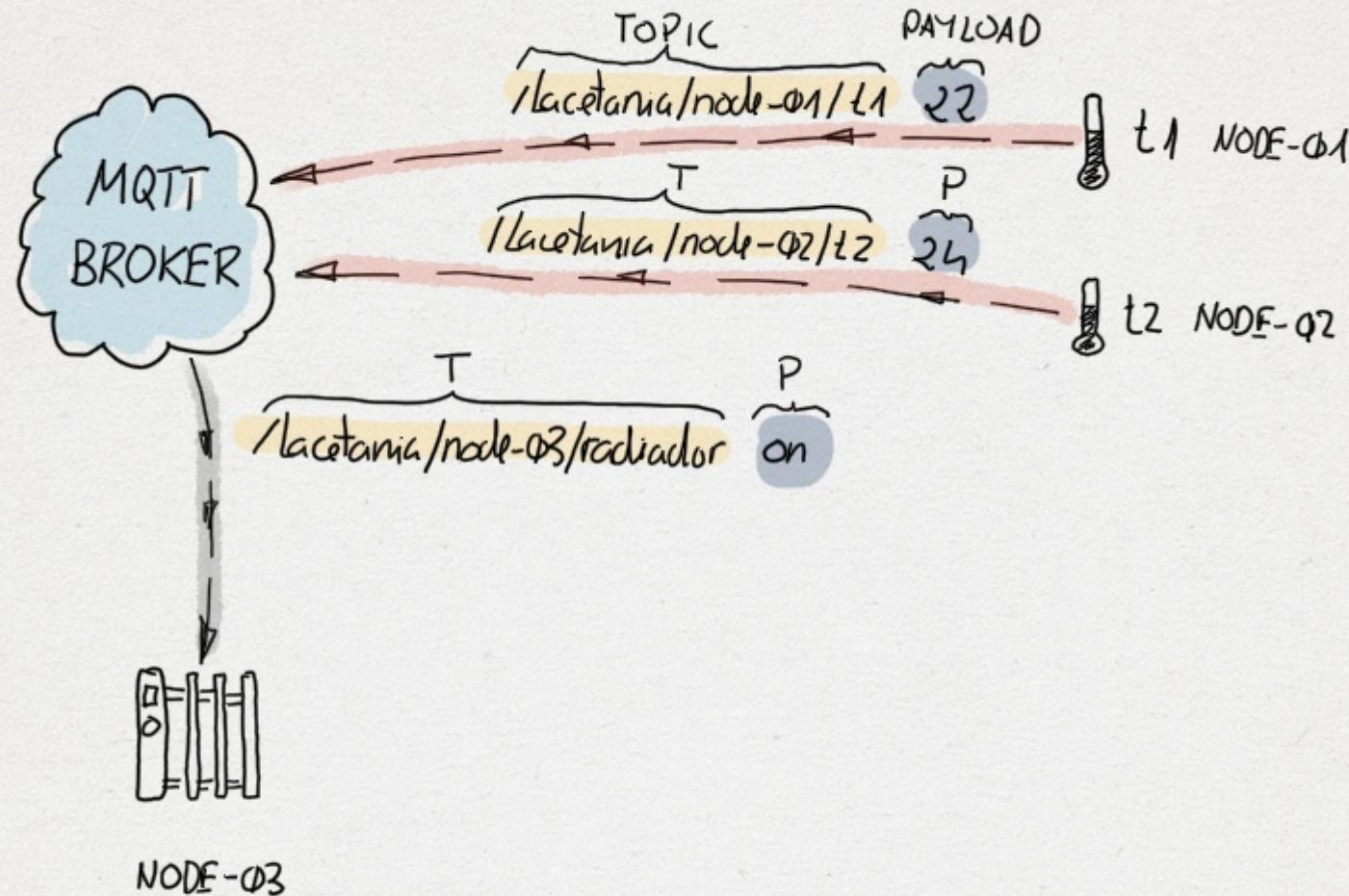
Nosaltres utilitzarem aquest:

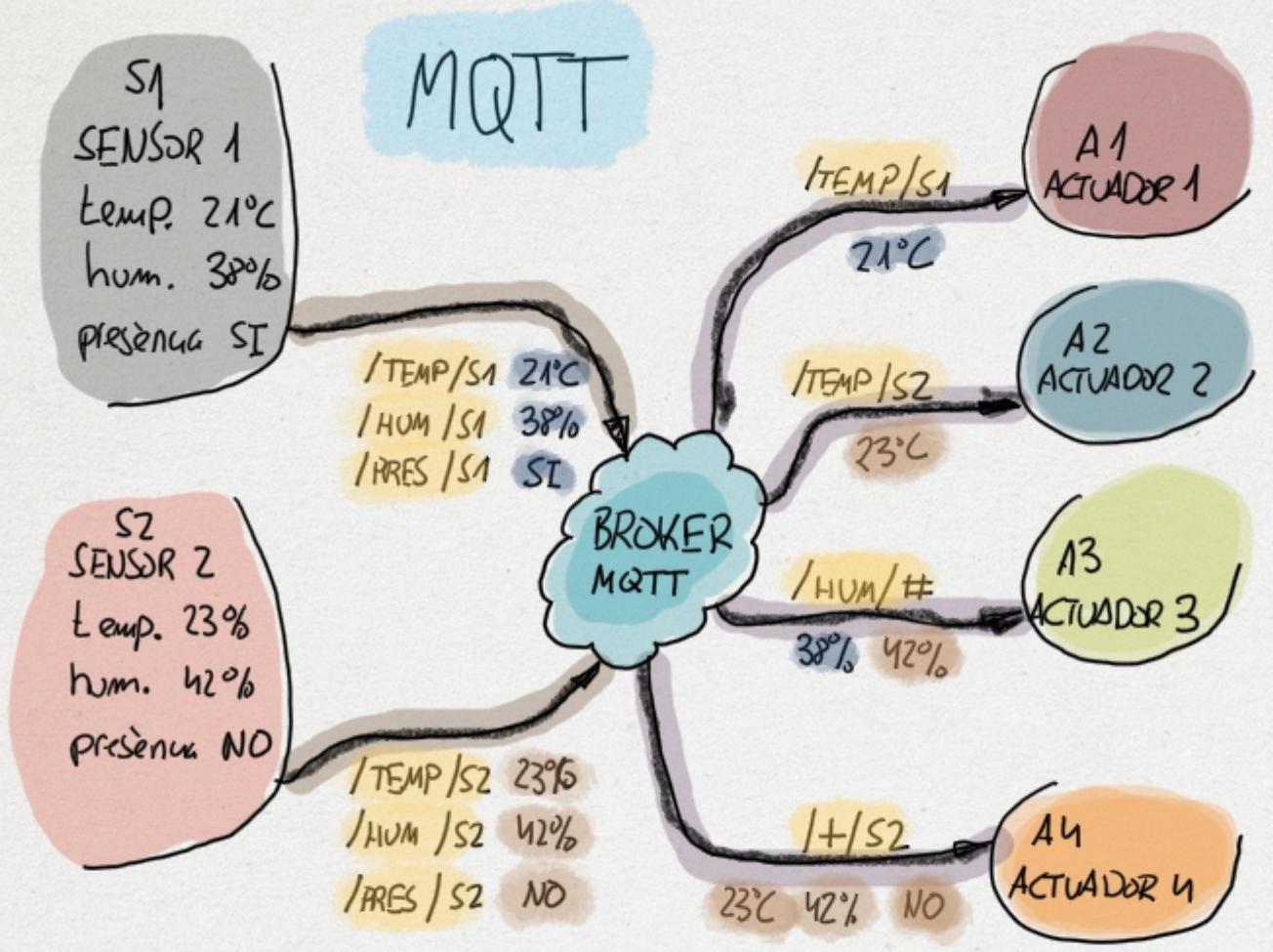
L'sketch necessari serà aquest [link](#).



vps249990.ovh.net
iot.eclipse.org







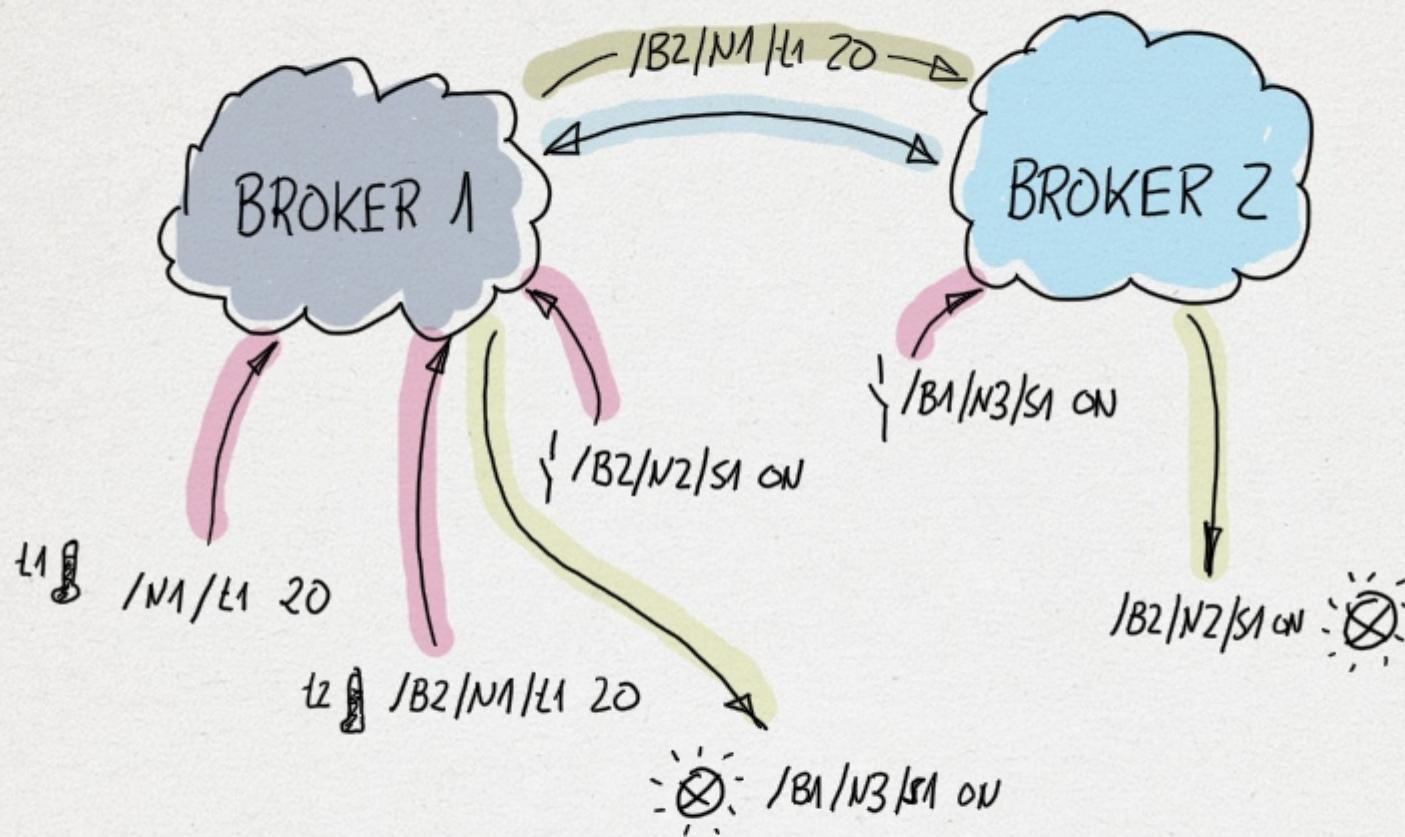
PUBLISH

SUBSCRIBE

TOPIC

WILDCARD #

+



MqttDash (android)



Instalar mqttDash [link](#)
Vídeo [link](#)

