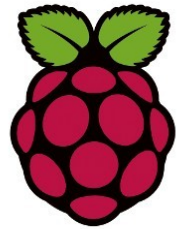


Sistema domòtic amb Internet de les Coses



IoT



MONGOOSE OS



Jordi Binefa i Martínez

Responsable d'R+D+i a



16 / setembre / 2017

Professor de cicles formatius a



JESUÏTES El Clot
Escola del Clot

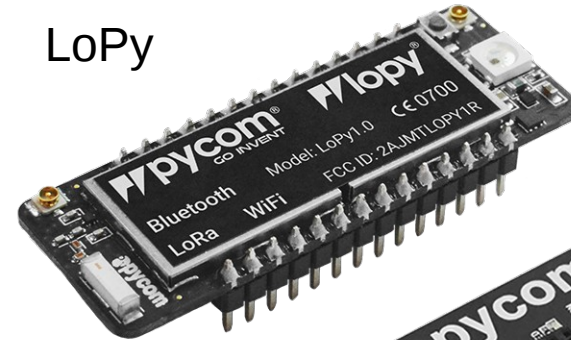
e⚙ Sistema domòtic amb Internet de les Coses²

Mòduls electrònics a preus populars

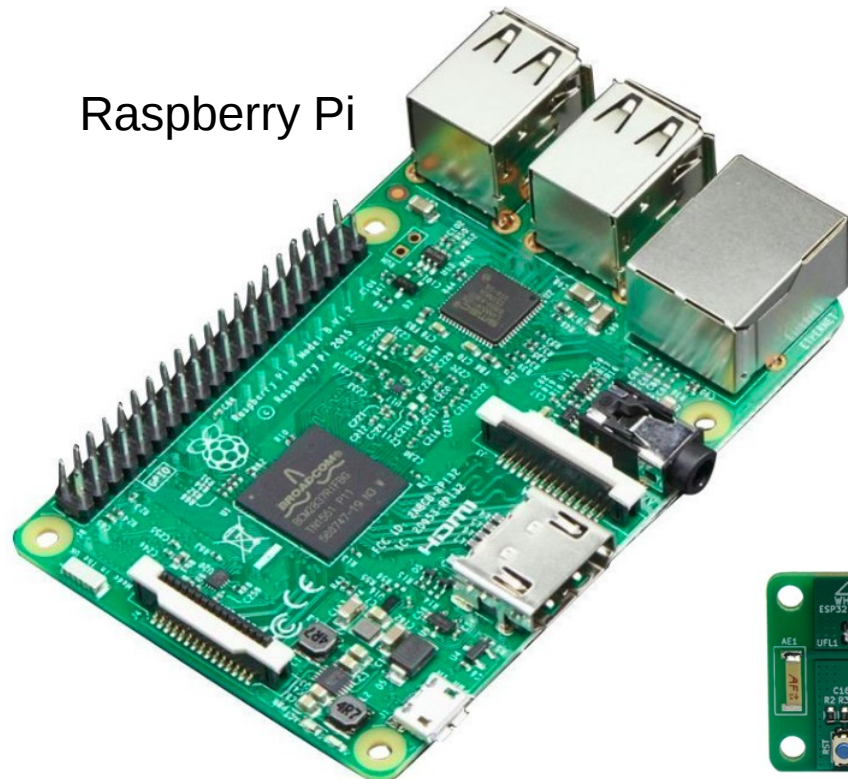
ESP8266



LoPy



Raspberry Pi



WhiteCatBoard



SiPy



e⚙ Sistema domòtic amb Internet de les Coses³

LoPy (LPWAN - LoRa)

Size: 55mm x 20mm x 3.5mm

Operating temperature:
-40 to 85 degrees celsius

ESP32 Dual Core
Microcontroller and
WiFi/Bluetooth 4.2
radio

3V3 Ultra-Low
-Noise switching
regulator

LoRa transceiver

32Mbit flash memory

WS2812 RGB
multi-colour
LED



External LoRa antenna
connector

RF switch

U.FL connector

Reset switch

Internal WiFi and
Bluetooth Antenna

e⚙ Sistema domòtic amb Internet de les Coses⁴

LoPy (LPWAN - LoRa)

Node A

```
from network import LoRa
import socket
import time

lora = LoRa(mode=LoRa.LORA, frequency=863000000)
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
s.setblocking(False)

while True:
    if s.recv(64) == b'Ping':
        s.send('Pong')
        time.sleep(5)
```

Node B

```
from network import LoRa
import socket
import time

lora = LoRa(mode=LoRa.LORA, frequency=863000000)
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
s.setblocking(False)

while True:
    s.send('Ping')
    time.sleep(5)
```



<https://docs.pycom.io/chapter/tutorials/lopy/lopy-lopy.html>

e⚙ Sistema domòtic amb Internet de les Coses⁵

SiPy (LPWAN - sigfox)

```
from network import Sigfox
import socket

# init Sigfox for RCZ1 (Europe)
sigfox = Sigfox(mode=Sigfox.SIGFOX, rcz=Sigfox.RCZ1)

# create a Sigfox socket
s = socket.socket(socket.AF_SIGFOX, socket.SOCK_RAW)

# make the socket blocking
s.setblocking(True)

# configure it as uplink only
s.setsockopt(socket.SOL_SIGFOX, socket.SO_RX, False)

# send some bytes
s.send(bytes([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]))
```

sigfox

Lost password

Dev Kit Activation

Activate your Sigfox subscription included with your Pycom kit.

Pick your country Device Information **Account details**

Account creation
Already have an account? [Sign in](#)

FIRST NAME *

LAST NAME *

EMAIL ADDRESS *

TIMEZONE
UTC

POSITION

COMPANY NAME *

COMPANY ADDRESS

[BACK](#) [SUBSCRIBE](#)



<https://docs.pycom.io/chapter/firmwareapi/pycom/network/sigfox.html>

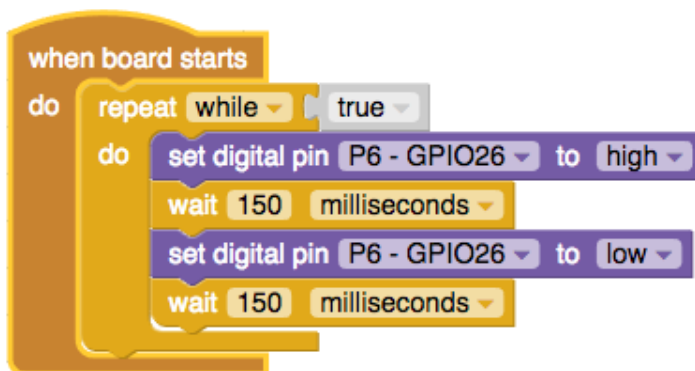
e⚙ Sistema domòtic amb Internet de les Coses⁶

WhiteCatBoard (LPWAN - LoRa)



- Dual-core Tensilica LX6 microprocessor
- Up to 240MHz clock frequency
- 520kB internal SRAM
- Integrated LoRa WAN transceiver working in the 868 (EU) Mhz / 915 (USA) Mhz, with onboard antenna, and UFL connector for external antenna
- Integrated 802.11 BGN WiFi transceiver, with onboard antenna, and UFL connector for external antenna
- Integrated dual-mode Bluetooth (classic and BLE)
- 3.3 to 5.5V operating range through input voltage regulator
- Second voltage regulator for power on / power off sensors through a dedicated GPIO
- Peripherals: SPI, I2C, I2S, SDIO, UART, CAN, ETHERNET, IR, PWM, DAC, ADC.
- RTC clock
- 4MB Flash memory
- Small form-factor: 78 x 26 mm

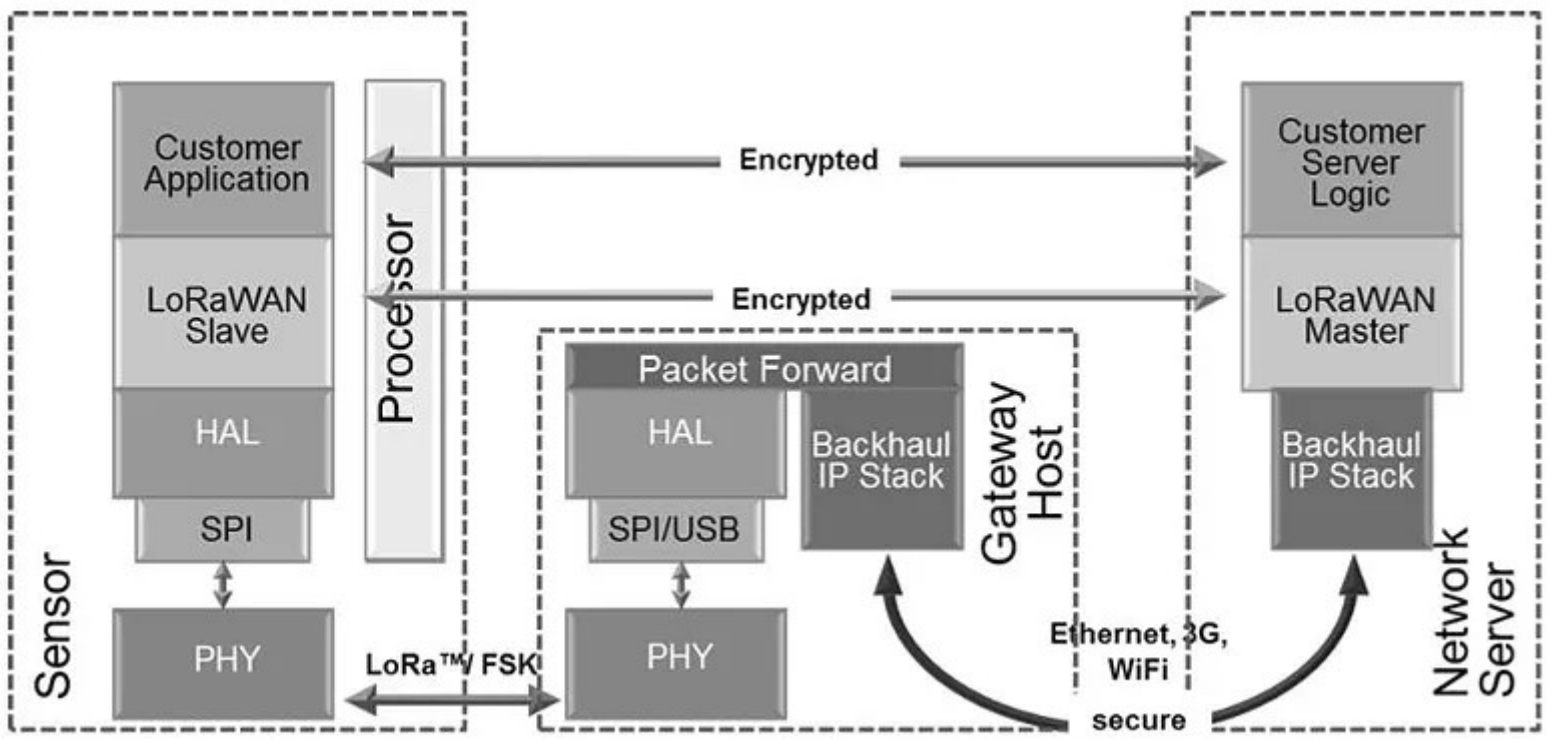
The Whitecat IDE supports two programming models: blocks, or the Lua programming language. No matter if you use blocks or Lua, both forms of programming are made from the same programming environment. You can decide, for example, to make a fast prototype using blocks, then change to Lua, and finally back to blocks.



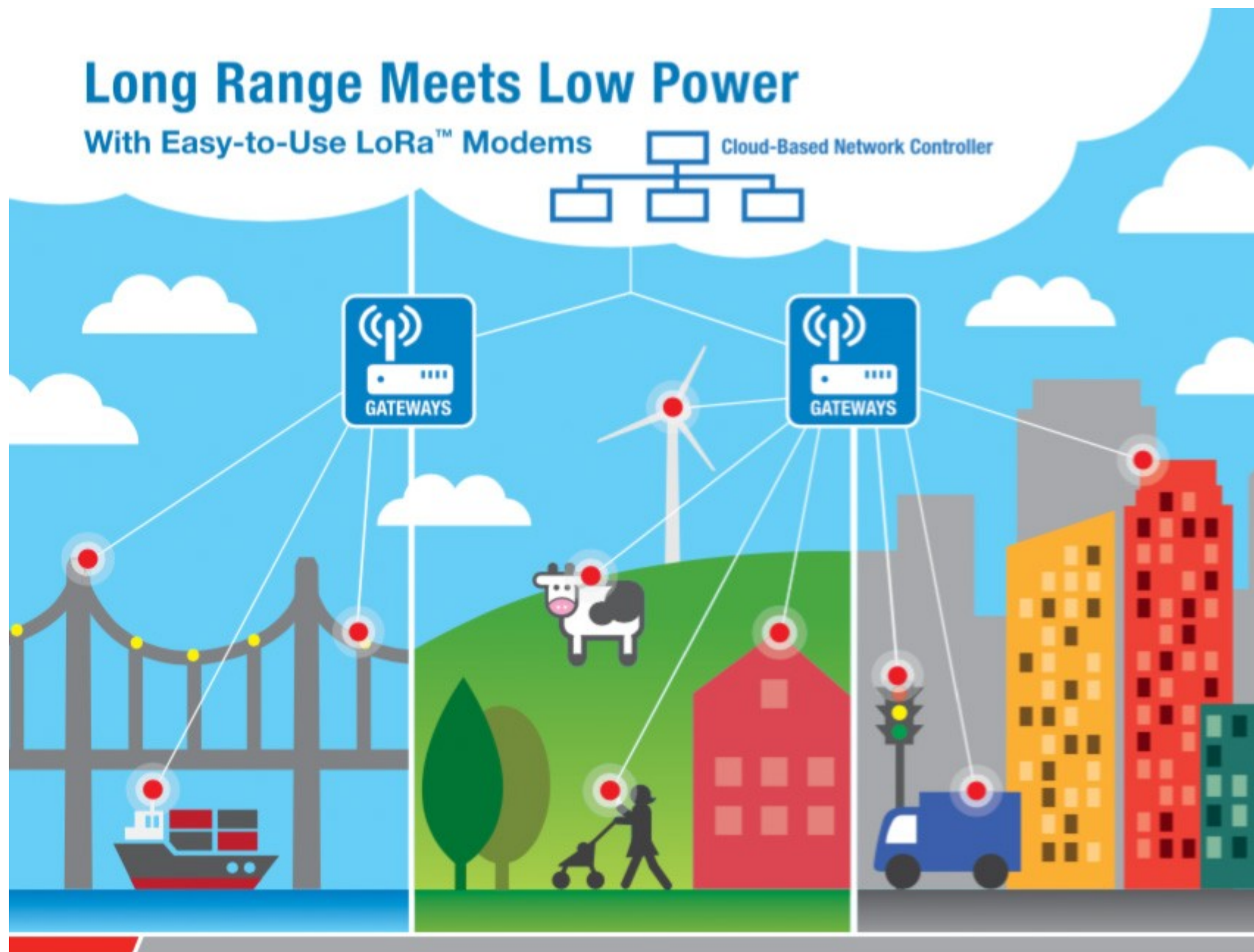
```
thread.start(function()
  pio.pin.setdir(pio.OUTPUT, pio.GPIO26)
  pio.pin.setpull(pio.NOPULL, pio.GPIO26)

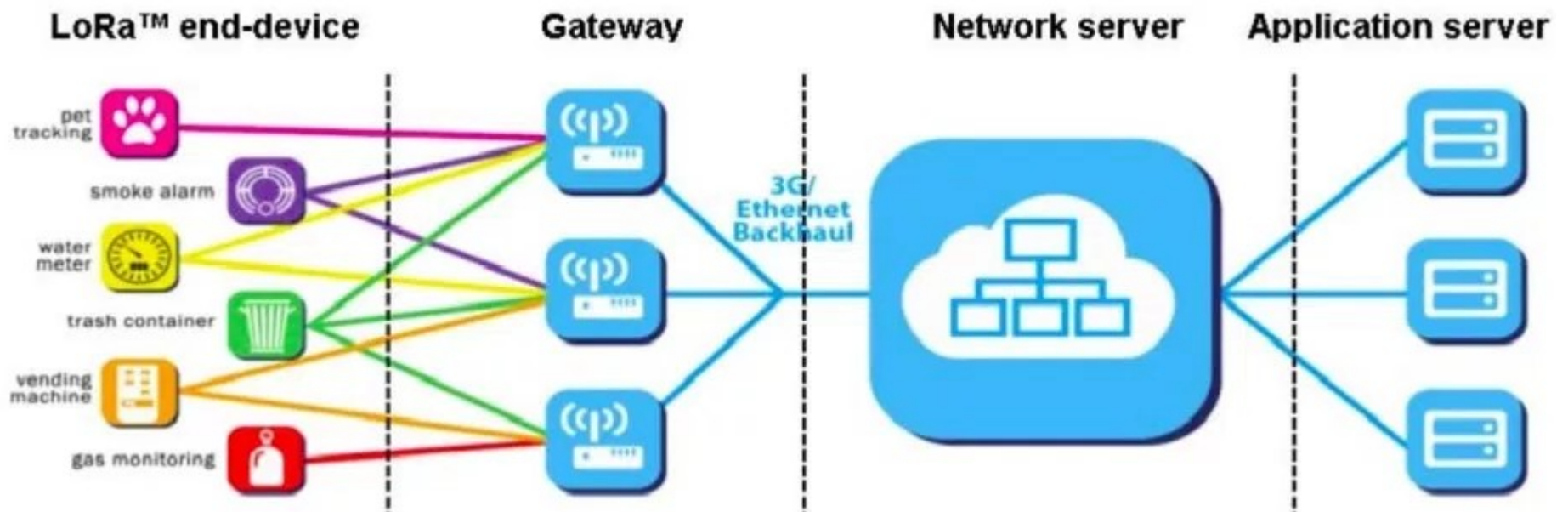
  while true do
    pio.pin.setval(1, pio.GPIO26)
    tmr.delayms(150)
    pio.pin.setval(0, pio.GPIO26)
    tmr.delayms(150)
  end
end)
```

<https://whitecatboard.org/>



	Local Area Network Short Range Communication	Low Power Wide Area (LPWAN) Internet of Things	Cellular Network Traditional M2M
🏠	40%	45%	15%
😊	Well established standards In building	Low power consumption Low cost Positioning	Existing coverage High data rate
😞	Battery Live Provisioning Network cost & dependencies	High data rate Emerging standards	Autonomy Total cost of ownership
	Bluetooth 4.0 Wi-Fi	LoRa	GSM 3G+ / H+ 4G

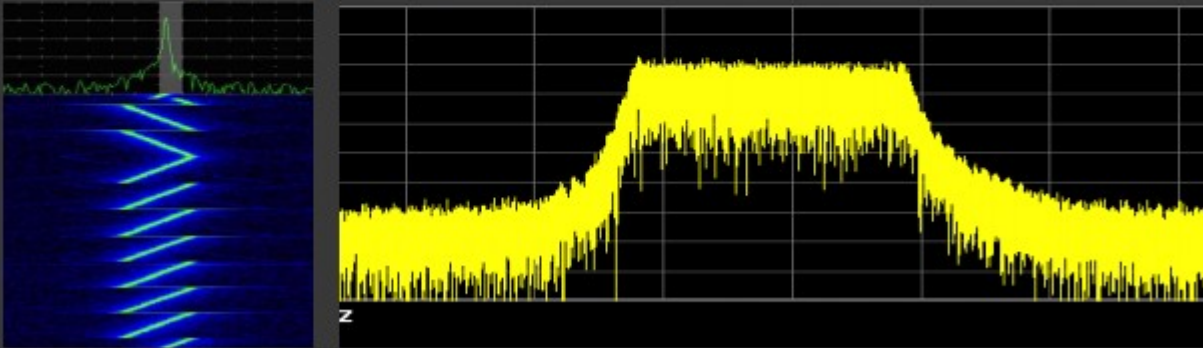




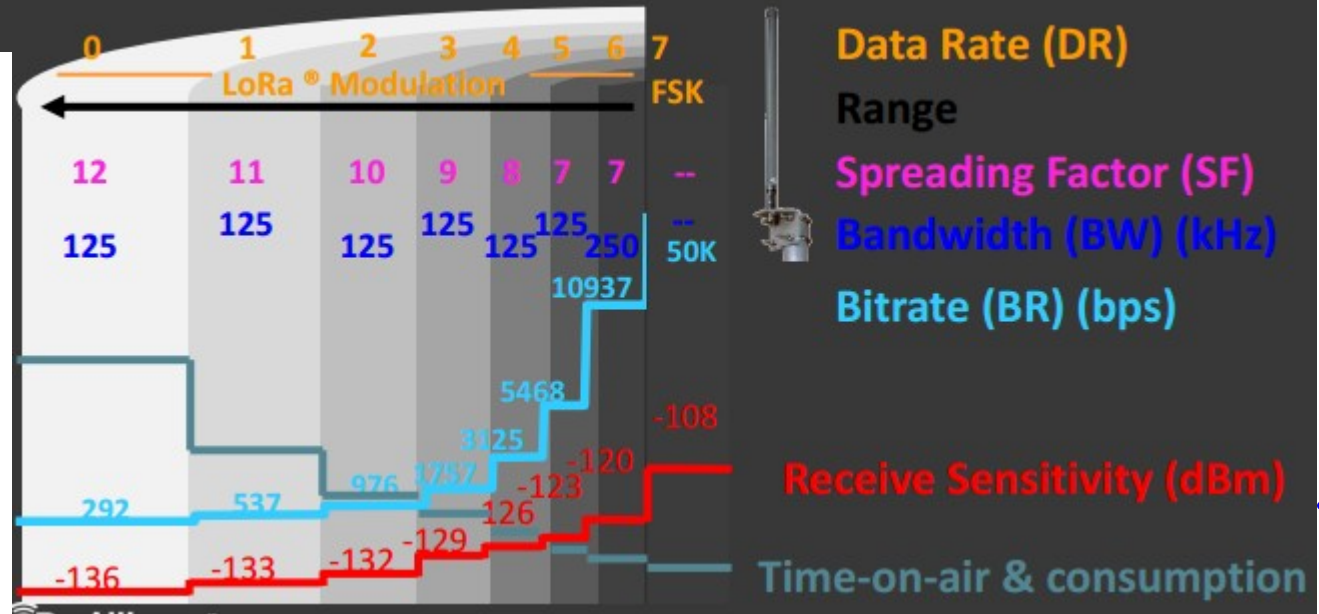
e⚙ Sistema domòtic amb Internet de les Coses¹⁰

LoRaWAN

- A Spread Spectrum Technology
 - Developed by Semtech Corporation (<http://www.semtech.com/>)
 - Chirped-FM modulation, symbols of ramping frequency
 - Processing gain = increased receive sensitivity
 - Enables longer range at expense of lower data rate



LoRa Alliance™
Wide Area Networks for IoT



PDF sobre LoRaWAN

e⚙ Sistema domòtic amb Internet de les Coses¹¹

LoRaWAN

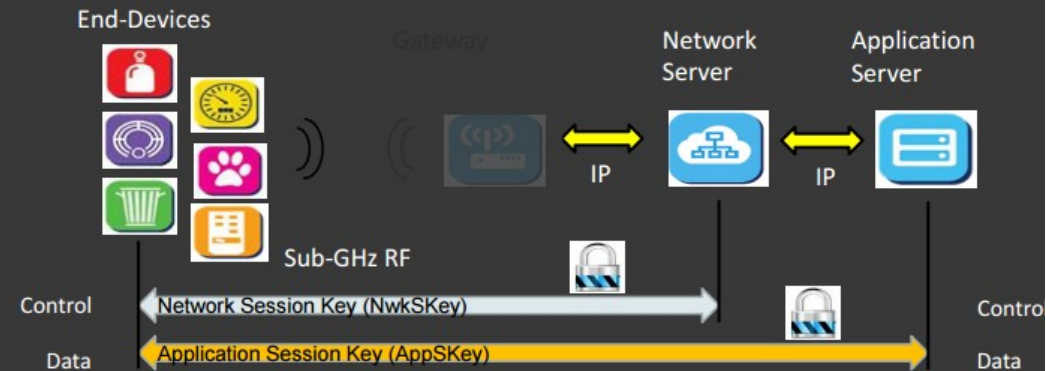
LoRaWAN™ Network Topology



LoRa Alliance™
Wide Area Networks for IoT

LoRa-Alliance.org

Logical Data Flow (Programmer's Model)



LoRa Alliance™
Wide Area Networks for IoT

LoRa-Alliance.org

- Battery Powered – **Class A**
 - Bidirectional communications
 - Unicast messages
 - Small payloads, long intervals
 - End-device initiates communication (uplink)
 - Server communicates with end-device (downlink) during predetermined response windows:



- No Latency – **Class C**
 - Bidirectional communications
 - Unicast and Multicast messages
 - Small payloads
 - Server can initiate transmission at any time
 - End-device is constantly receiving



- Low Latency – **Class B**
 - Bidirectional with scheduled receive slots
 - Unicast and Multicast messages
 - Small payloads, long intervals
 - Periodic beacon from gateway
 - Extra receive window (ping slot)
 - Server can initiate transmission at fixed intervals



e⚙ Sistema domòtic amb Internet de les Coses

MQTT

12



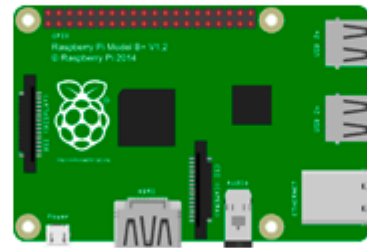
MQ Telemetry Transport

Port 1883

Python Web Server
MQTT Client



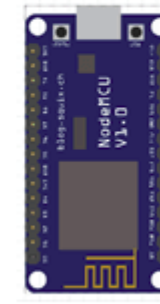
Raspberry Pi
Mosquitto Broker



Publish in:
/esp8266/gpio5

Message: "1"

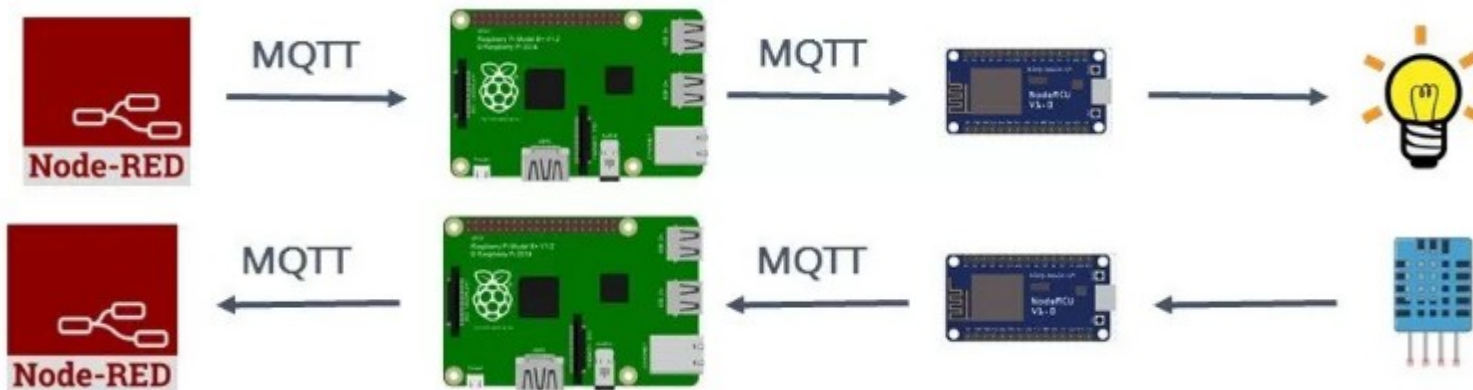
ESP8266
MQTT Client



GPIO 5 - ON

Subscribe to:
/esp8266/gpio5

<http://www.recantha.co.uk/blog/?p=16336>



<https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>

e⚙ Sistema domòtic amb Internet de les Coses MQTT

13

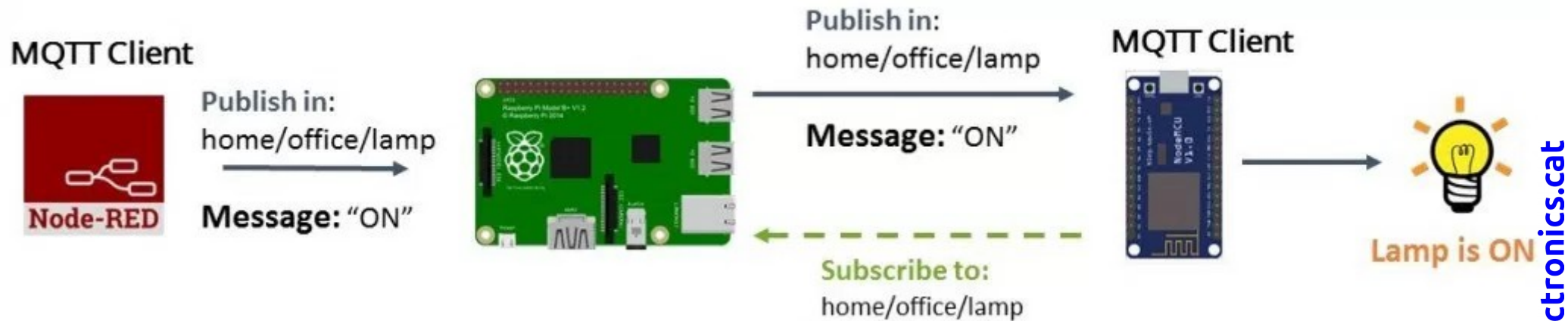


Topic level separator

home/office/lamp

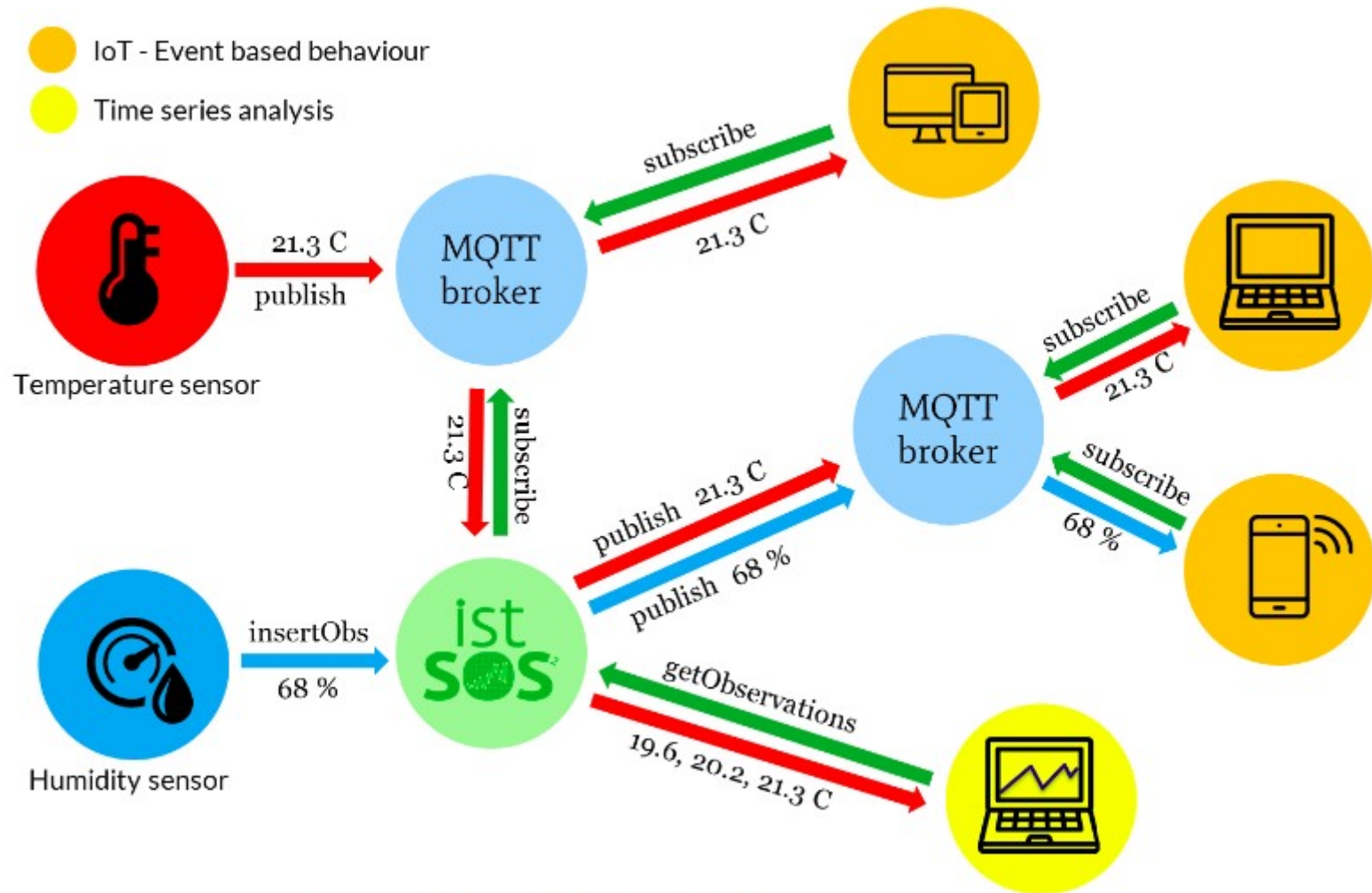
home/office/lamp $\neq$ home/office/Lamp

Topic level Topic level



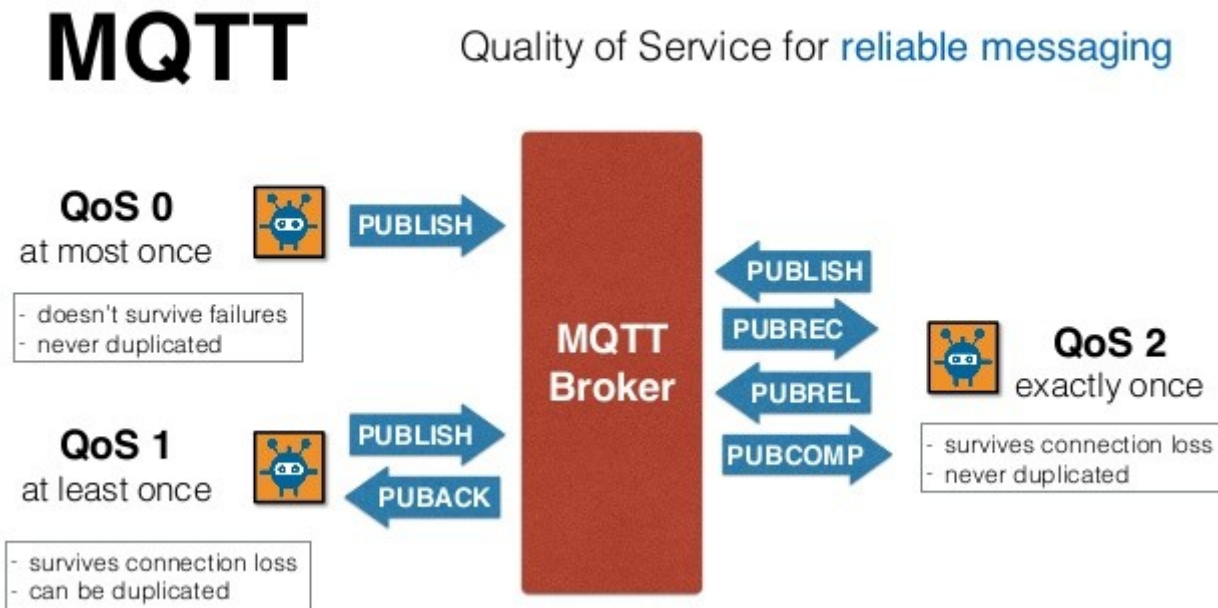
e⚙ Sistema domòtic amb Internet de les Coses MQTT

14



Schematic data flow of an MQTT integrated istSOS

http://istsos.org/en/trunk/doc/ws_mqtt.html

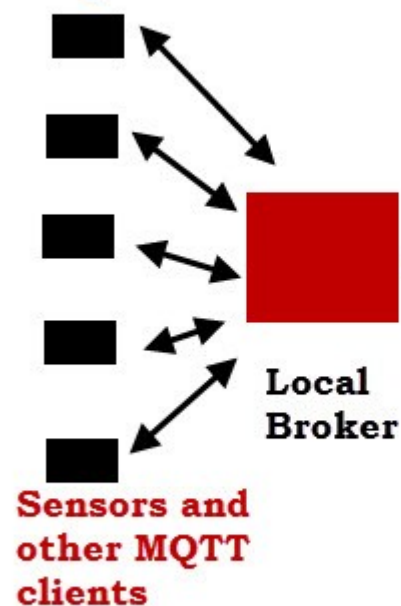


La importància de la qualitat del servei (QoS) en sistemes de molt baix consum

e⚙ Sistema domòtic amb Internet de les Coses

MQTT - Pont

Internal Topics



Shared Topics

Bridge

MQTT Bridge Illustration

Central or Remote Broker

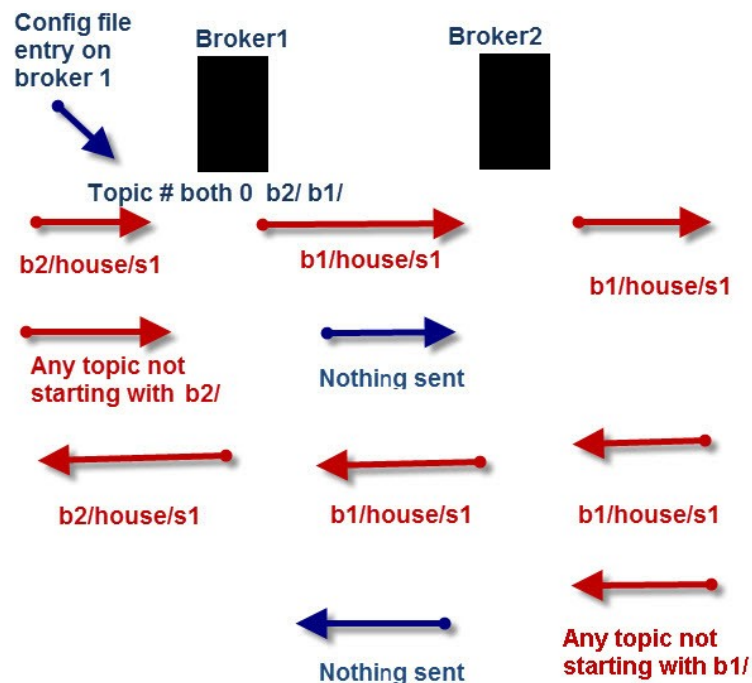
Mosquitto Bridge Simple Configuration

```
#connection <name>

connection bridge-01
address 192.168.1.184:1883

topic # out 0 "" b1/
topic # in 0 "" b1/
```

MQTT Bridging Topic Remapping Example



e⚙ Sistema domòtic amb Internet de les Coses 17

MQTTs

Step 6 — Configuring MQTT SSL

To enable SSL encryption, we need to tell Mosquitto where our Let's Encrypt certificates are stored. Open up the configuration file we previously started:

```
$ sudo nano /etc/mosquitto/conf.d/default.conf
```

Paste in the following at the end of the file, leaving the two lines we already added:

```
/etc/mosquitto/conf.d/default.conf
```

```
...  
listener 1883 localhost  
  
listener 8883  
certfile /etc/letsencrypt/live/mqtt.example.com/cert.pem  
cafile /etc/letsencrypt/live/mqtt.example.com/chain.pem  
keyfile /etc/letsencrypt/live/mqtt.example.com/privkey.pem
```

We're adding two separate `listener` blocks to the config. The first, `listener` updates the default MQTT listener on port `1883` which is what we've been con-

Cal nom de domini

Configuració d'un sistema MQTT segur



The screenshot shows the Duck DNS web interface. At the top, there's a navigation bar with links: spec, about, why, install, faqs, logout, and a 'logged in' status. The main header features the Duck DNS logo (a yellow duck) and the text 'Duck DNS'. Below this, account details are displayed: 'account' is 'example@gmail.com', 'type' is 'free', 'token' is a long alphanumeric string, 'token generated' is '2 weeks ago', and 'created date' is 'Dec 3, 2016 9:34:21 PM'. A section titled 'domains' shows a list of domains with columns for 'domain', 'current ip', 'ipv6', and 'changed'. The first domain listed is 'example.duckdns.org' with a 'current ip' of '79.131.102.1' and a 'changed' status of '15 hours ago'. There are buttons for 'update ip' and 'update ipv6' next to the IP addresses.

<http://www.duckdns.org/>

e⚙ Sistema domòtic amb Internet de les Coses

MQTT a LoPy i SiPy (microPython)

18

```
from network import WLAN
from mqtt import MQTTClient
import machine
import time

def sub_cb(topic, msg):
    print(msg)

wlan = WLAN(mode=WLAN.STA)
wlan.connect("yourwifinetwork", auth=(WLAN.WPA2, "wifipassword"), timeout=5000)

while not wlan.isconnected():
    machine.idle()
print("Connected to Wifi\n")

client = MQTTClient("device_id", "io.adafruit.com", user="your_username", password="your_api_key", port=1883)
client.set_callback(sub_cb)
client.connect()
client.subscribe(topic="youraccount/feeds/lights")
while True:
    print("Sending ON")
    client.publish(topic="youraccount/feeds/lights", msg="ON")
    time.sleep(1)
    print("Sending OFF")
    client.publish(topic="youraccount/feeds/lights", msg="OFF")

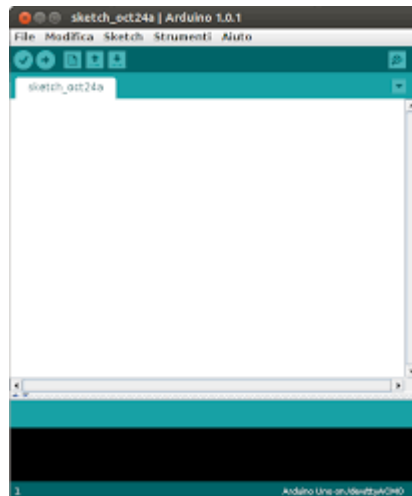
    time.sleep(1)
```

<https://www.hackster.io/bucknalla/mqtt-micropython-044e77>

e⚙ Sistema domòtic amb Internet de les Coses

ESP8266 - Witty

19



```
// Testing ESP8266 board using LDR, RGB led & button
//
// by Jordi Binefa - twitter.com/jordibinefa
// 20160328 - www.binefa.cat/blog & www.electronics.cat
#include <ESP8266WiFi.h>
```

```
#define GPIO5 5
#define GPIO4 4
#define GPIO0 0
#define GPIO2 2
#define GPIO15 15
#define GPIO13 13
#define GPIO12 12
#define GPIO14 14
#define GPIO16 16
```

```
String szMsg;
```

```
void setup() {
  Serial.begin(9600);

  pinMode(GPIO5, OUTPUT);
  pinMode(GPIO4, INPUT ); // For button. It can be used as output as well
  pinMode(GPIO0, OUTPUT);
  pinMode(GPIO2, OUTPUT); // Internal blu led
  pinMode(GPIO15,OUTPUT); // Red of RGB led
  pinMode(GPIO13,OUTPUT); // Blue of RGB led
  pinMode(GPIO12,OUTPUT); // Green of RGB led
  pinMode(GPIO14,OUTPUT);
  pinMode(GPIO16,OUTPUT);
  pinMode(A0 ,INPUT ); // LDR
}
```

```
void delayESP8266(unsigned long ulMilliseconds){
  unsigned long ulPreviousMillis = millis();
```

Codi de prova per al Witty

<https://twitter.com/JordiBinefa/status/714548974301093888>

PDF sobre l'ús de l'ESP8266 (material propi)

e⚙ Sistema domòtic amb Internet de les Coses

Mongoose OS

20

Mongoose OS – Mozilla Firefox

Mongoose OS x +

127.0.0.1:1992/# Cerca

Device File Manager

Device Files

Projects

Device Config

Device Service

Terminal

Forum

Docs

Update mos tool... esp8266 | Sep11 | 192.168.43.86

MONGOOSE OS Device status: *online* ●

Step 1 ✓ done! Plug in your device, choose device address, click Select

▼ /dev/ttyUSB0 ? Select

Step 2 ✓ done! Select firmware, click Flash

▼ mos-esp8266-latest Flash

Mongoose OS is installed on this device:
esp8266 | Sep11 | 192.168.43.86 | 124k/228k | 36k/51k

Step 3 ✓ done! Configure WiFi

pvSL03 Save

</> Start coding

Device logs

```
[Sep 16 16:52:17.549] Tock uptime: 26  
[Sep 16 16:52:18.549] Tick uptime: 27
```

<https://mongoose-os.com/>

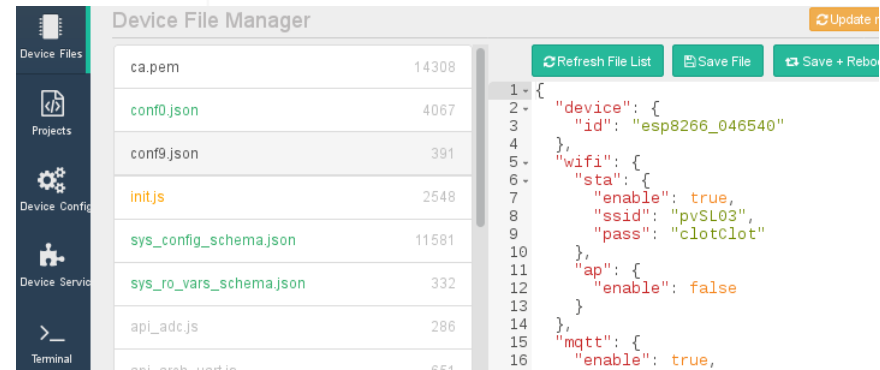
e⚙ Sistema domòtic amb Internet de les Coses

Mongoose OS

21

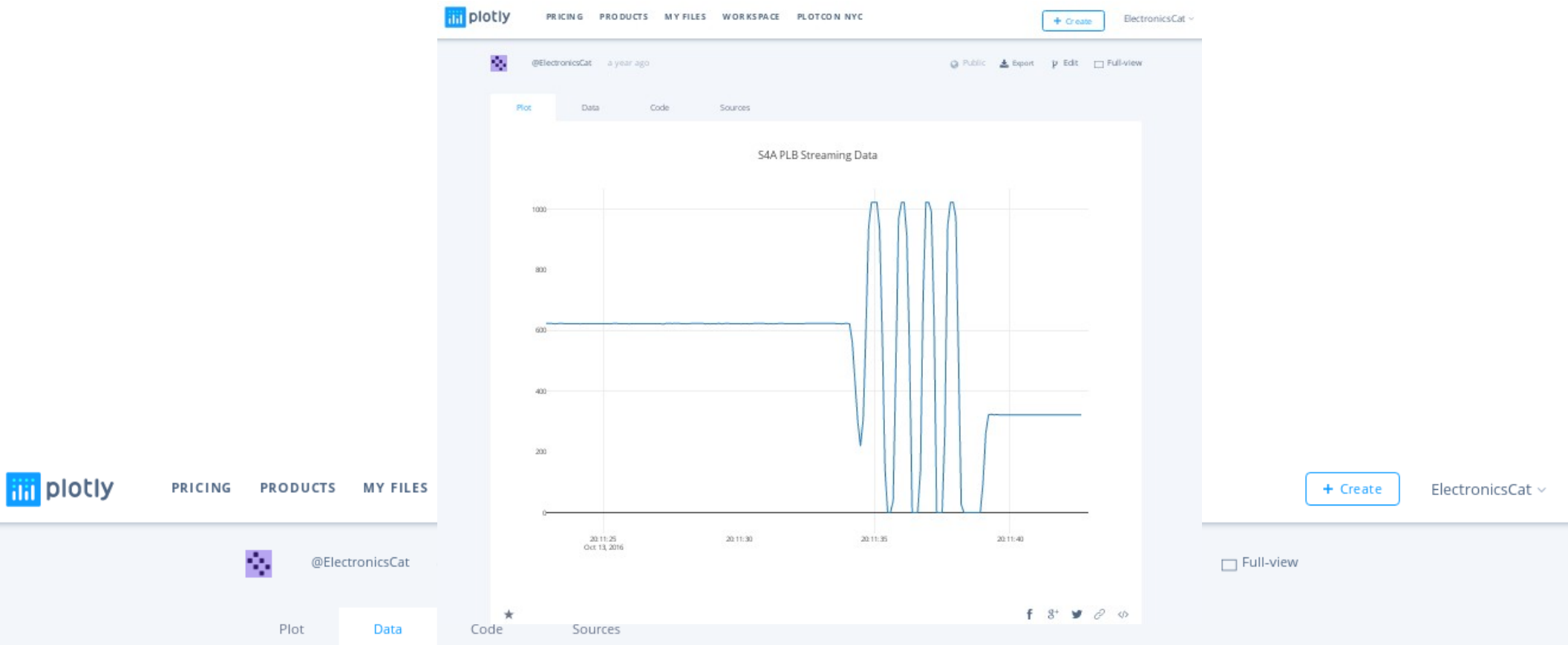
```
38 let message = "";
39 if(GPIO.read(button) === 1){
40   message = "UNPRESSED";
41 }else{
42   message = "PRESSED";
43 }let ok = MQTT.pub(topic, message, 1);
44 print('Published:', ok ? 'yes' : 'no', 'topic:', topic, 'message:', message);
45 }, null);
46
47 GPIO.set_mode(ledR, GPIO.MODE_OUTPUT);
48 GPIO.set_mode(ledG, GPIO.MODE_OUTPUT);
49 GPIO.set_mode(ledB, GPIO.MODE_OUTPUT);
50 ADC.enable(ldr);
51
52 MQTT.sub('/my/topic/#', function(conn, topic, msg) {
53   print('Topic:', topic, 'message:', msg);
54   let myTopic = "/my/topic/" + Cfg.get('device.id') + '/';
55   if(topic === myTopic){
56     print('Per a mi! - It\'s for me!');
57     if(msg === "R"){
58       GPIO.toggle(ledR);
59     }
60     if(msg === "G"){
61       GPIO.toggle(ledG);
62     }
63     if(msg === "B"){
64       GPIO.toggle(ledB);
65     }
66     if(msg === "L"){
67       //print('Published:', ok ? 'yes' : 'no', 'topic:', topic, 'message:', message);
68       //print("LDR: " + JSON.stringify(ADC.read(ldr)));
69       let topic = '/devices/' + Cfg.get('device.id') + '/ldr';
70       let message = JSON.stringify(ADC.read(ldr));
71       let ok = MQTT.pub(topic, message, 1);
72       print('Published:', ok ? 'yes' : 'no', 'topic:', topic, 'message:', message);
73     }
74   }
75 }, null);
76 }, null);
```

```
{
  "device": {
    "id": "esp8266_046540"
  },
  "wifi": {
    "sta": {
      "enable": true,
      "ssid": "pvSL03",
      "pass": "clotClot"
    },
    "ap": {
      "enable": false
    }
  },
  "mqtt": {
    "enable": true,
    "server": "192.168.43.1:1883",
    "reconnect_timeout_min": 2.0,
    "reconnect_timeout_max": 60.0,
    "clean_session": true,
    "keep_alive": 60
  }
}
```



e⚙ Sistema domòtic amb Internet de les Coses 22

Panells de control - plot.ly



	x	y (1)
0	2016-10-13 20:11:22.625274	622
1	2016-10-13 20:11:22.725682	624
2	2016-10-13 20:11:22.826449	623
3	2016-10-13 20:11:22.926858	624
4	2016-10-13 20:11:23.027285	624
5	2016-10-13 20:11:23.128061	623
6	2016-10-13 20:11:23.228394	622
7	2016-10-13 20:11:23.328868	623
8	2016-10-13 20:11:23.429664	624
9	2016-10-13 20:11:23.530024	623
10	2016-10-13 20:11:23.630386	623
11	2016-10-13 20:11:23.731193	623

<https://hackaday.io/project/5797-plotly-esp8266/details>

<https://binefa.cat/blog/?p=373>

<https://plot.ly>

e⚙ Sistema domòtic amb Internet de les Coses 23

Panells de control - plot.ly

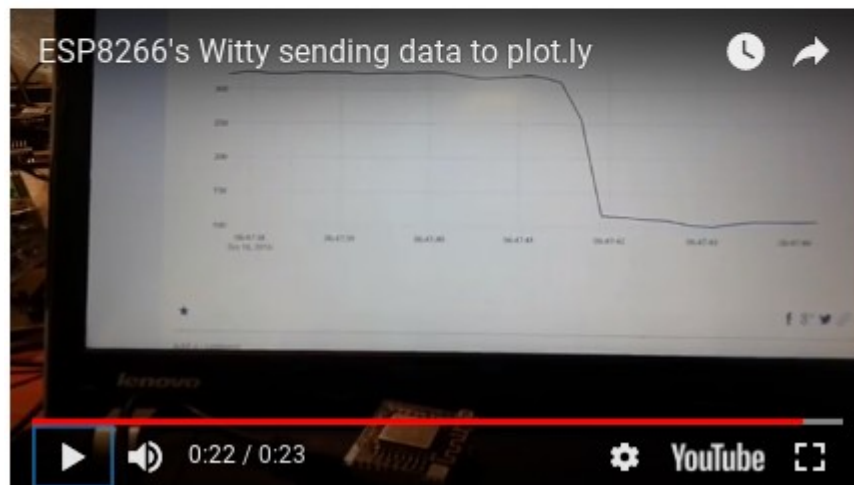
ESP8266'S WITTY SENDING DATA TO PLOT.LY

© 2016/10/15 JORDI BINEFA



Testing a [ESP8266's Witty board](#) sending data to [plot.ly](#)

Based on <https://hackaday.io/post/17879>



[ESP8266's Arduino IDE code](#)

```
// Based on https://hackaday.io/post/17879
// 20161015 - www.binefa.cat/blog
#include <ESP8266WiFi.h>
#define ssid "your_WiFi_SSID"
#define pass "your_WiFi_password"
#define userName "your_plot.ly_user_name"
#define APIKey "your_plot.ly_API_key"
#define fileName "test"
#define fileopt "overwrite"
#define nTraces 1
#define maxpoints "30"
#define world_readable true
#define convertTimestamp true
#define timezone "Europe/Brussels"

char *tokens[nTraces] = {"your_plot.ly_streaming_API_token"};
char stream_site[25] = {0};
WiFiClient client;

/* ----- Setup ----- */
void setup() {
  // Setup Serial
  Serial.begin(9600);
  delay(2000);
  Serial.println();
  Serial.println();
  ESP8266_Init();
  ESP8266_Connect("plot.ly", 80);
  Serial.println("Initializing plot with Plot.ly server...");
  plotly_init();
  Serial.println("Making sure disconnected...");
  client.stop();
  Serial.println("Done!");
  ESP8266_Connect("arduino.plot.ly", 80);
}
```

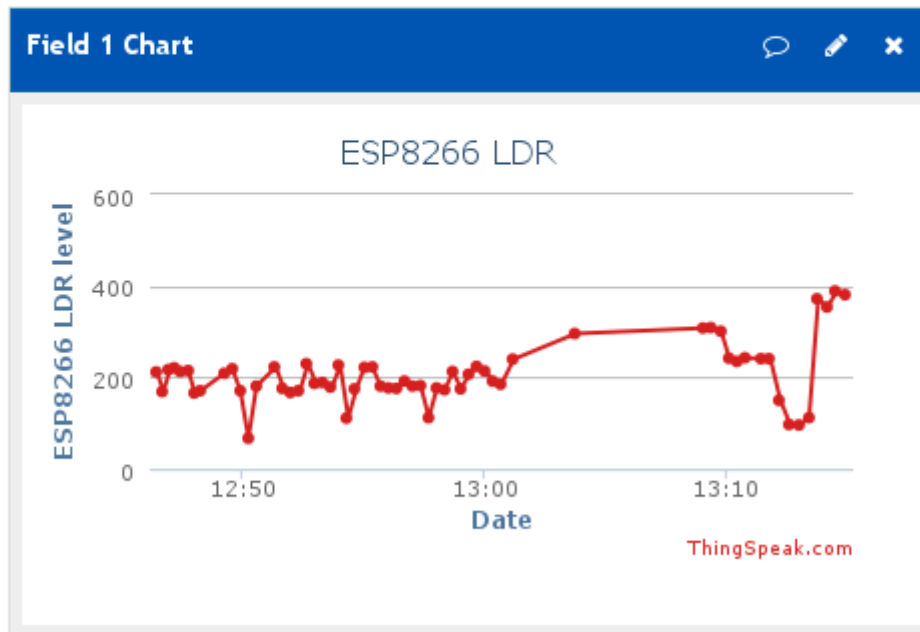
Codi font per a l'ESP8266 a l'Arduino IDE

<https://binefa.cat/blog/?p=373>

e⚙ Sistema domòtic amb Internet de les Coses 24

Panells de control - ThingSpeak

ThingSpeak™ Channels Apps Blog Support ▾
Last Entry 3 minutes ago
12 Entries



```
WiFi Connected  
58RJS0G7WVMVZB&field1=354  
ESP8266 in sleep mode  
;l d00| 0$0|00 00 $00c|!0000{0  
WiFi Connected  
58RJS0G7WVMVZB&field1=389  
ESP8266 in sleep mode  
rl d00| 010|00 00$00c|000000{  
WiFi Connected  
58RJS0G7WVMVZB&field1=381  
ESP8266 in sleep mode  
☒ Desplaçament automàtic
```

My Account

Edit Account

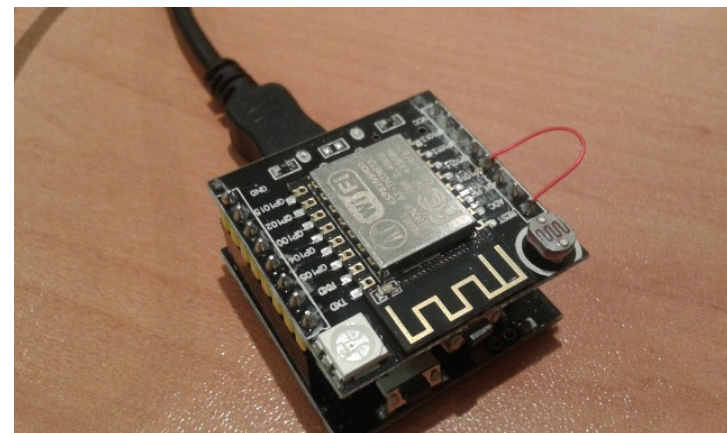
User ID ElectronicsCat

Email electronicscat@gmail.com

Time Zone Brussels

API Key D6KXXXXXXXXXXXXD0

Generate New API Key



e⚙ Sistema domòtic amb Internet de les Coses

Panells de control - ThingSpeak

```
// 20161014 - www.binefa.cat/blog
#include <ESP8266WiFi.h>
```

```
WiFiClient client;
const char* MY_SSID = "your_SSID";
const char* MY_PWD = "your_wifi_password";
const char* TS_SERVER = "api.thingspeak.com";
String TS_API_KEY = "your_thingspeak_key";
const long int sleepTimeS = 20;
```

```
void setup(){
  Serial.begin(115200);
  delay(10);
  connectWifi();
  sendDataTS();
  vDeepSleep(sleepTimeS);
}
```

```
void loop(){
}
```

```
void connectWifi(){
  int nCnt = 0;
  Serial.print("Connecting to " + *MY_SSID);
  WiFi.begin(MY_SSID, MY_PWD);
  while (WiFi.status() != WL_CONNECTED){
    delay(1000);
    Serial.print(".");
    nCnt ++;
    if(nCnt > 20) vDeepSleep(sleepTimeS);
  }
  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println("");
}
```

```
void vDeepSleep(long int nSeconds){
  Serial.println("ESP8266 in sleep mode");
  ESP.deepSleep(nSeconds * 1000000L);
}
```

```
void sendDataTS(void){
  if (client.connect(TS_SERVER, 80)){
    String postStr = TS_API_KEY;
    postStr += "&field1=";
    postStr += String(analogRead(A0));
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + TS_API_KEY + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);

    Serial.print(postStr);

    delay(1000);
  }
  client.stop();
}
```

ThingSpeak™

Channels ▾

Apps

Blog

Support

Last Entry 3 minutes ago

12 Entries



Field 1 Chart

ESP8266 LDR



ThingSpeak.com

<https://binefa.cat/blog/?p=365>

<https://thingspeak.com/>

e⚙ Sistema domòtic amb Internet de les Coses 26

Panells de control - thethings.io

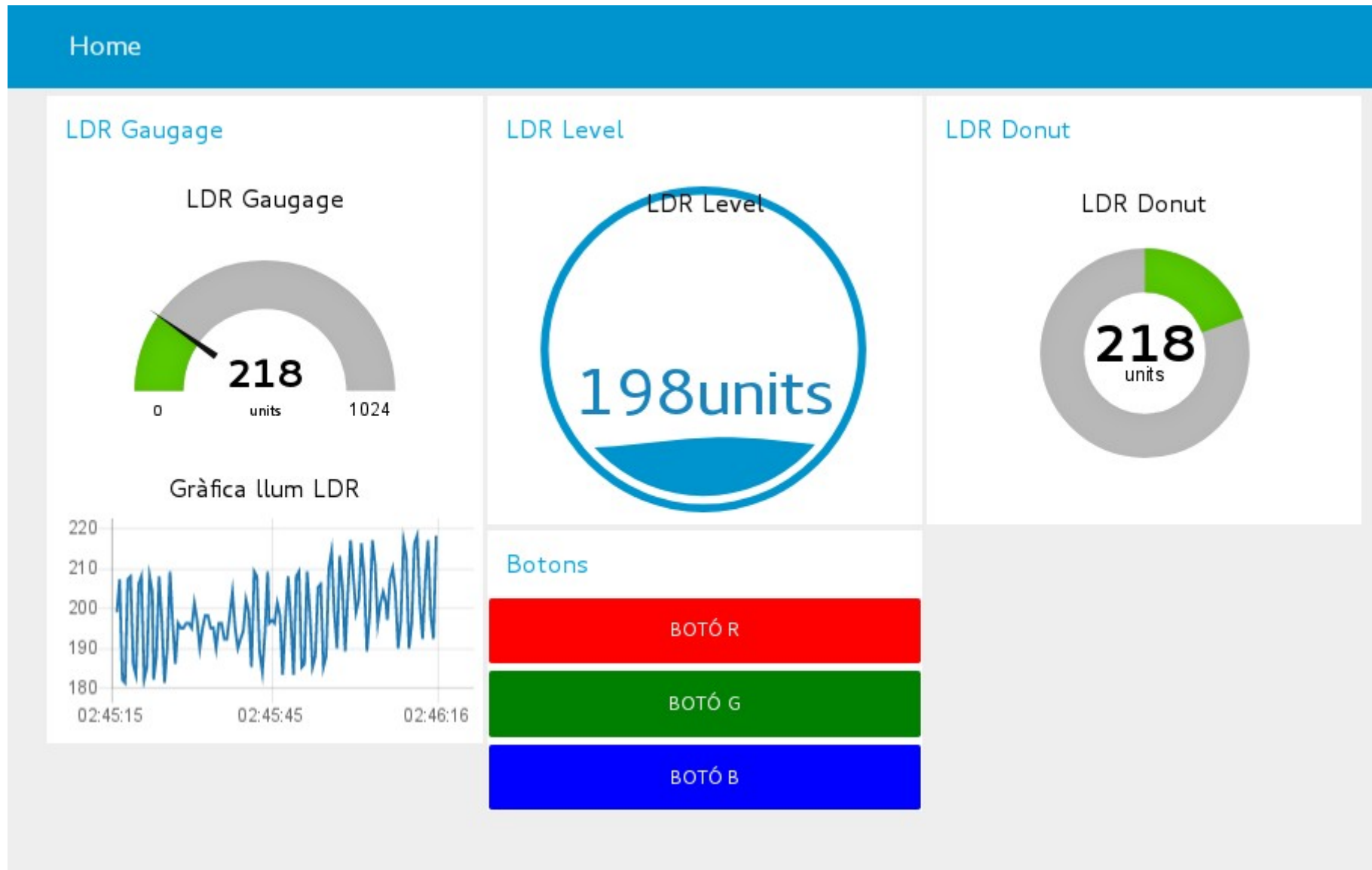


<https://binefa.cat/blog/?p=470>

<https://thethings.io/>

e⚙ Sistema domòtic amb Internet de les Coses 27

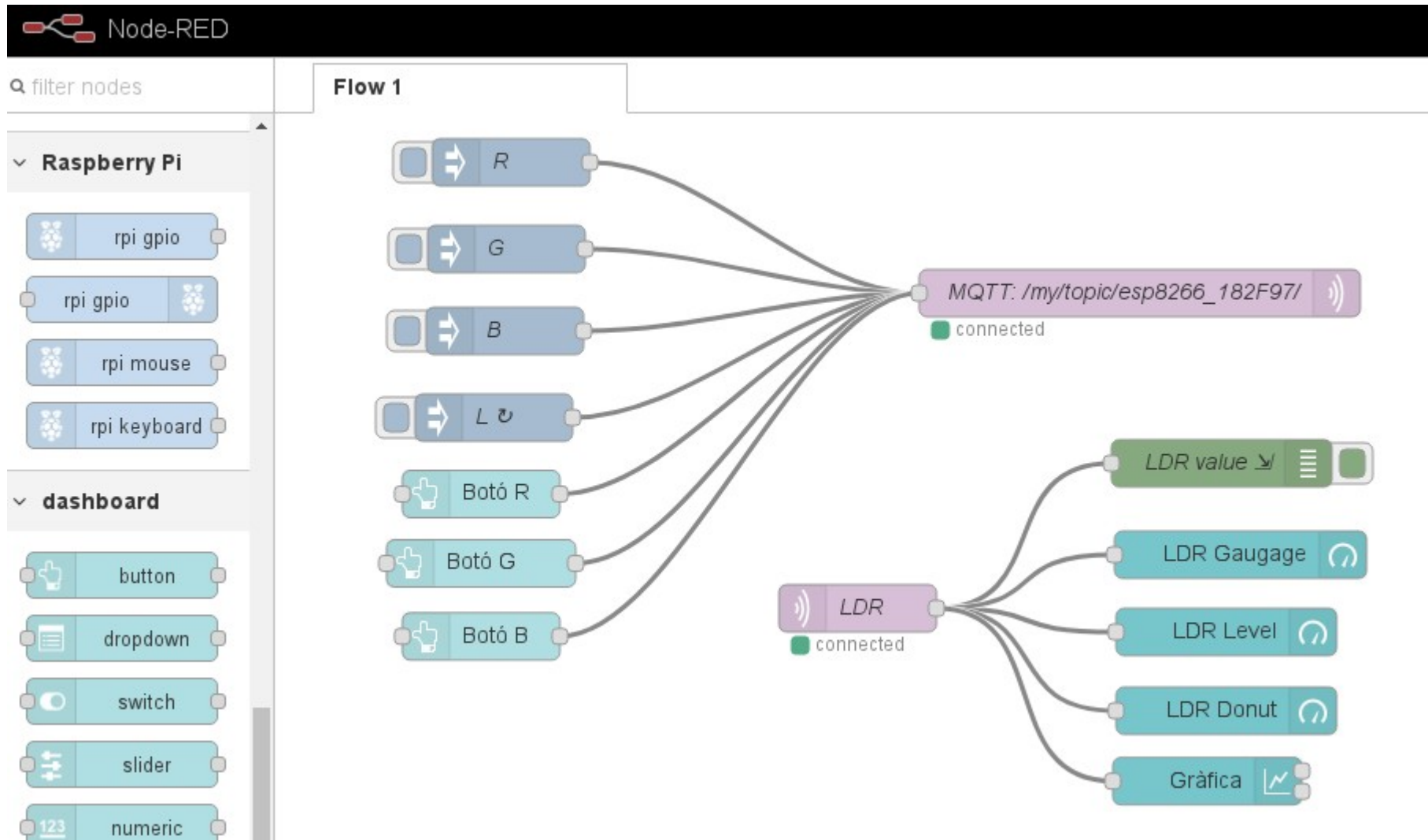
Panells de control - NodeRED



Instal·lació del NodeRED a la Raspberry Pi
Instal·lació del panell de control al NodeRED
<https://nodered.org>

e⚙ Sistema domòtic amb Internet de les Coses 28

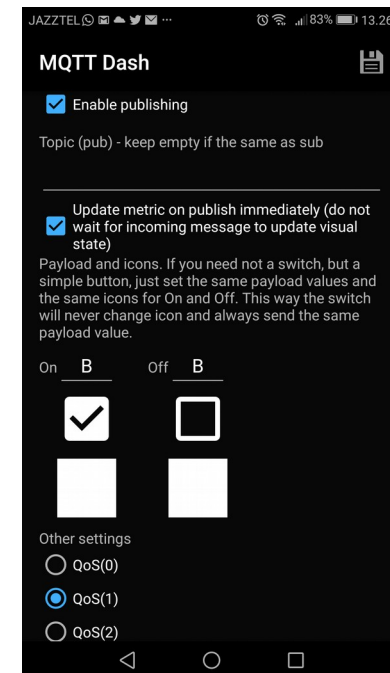
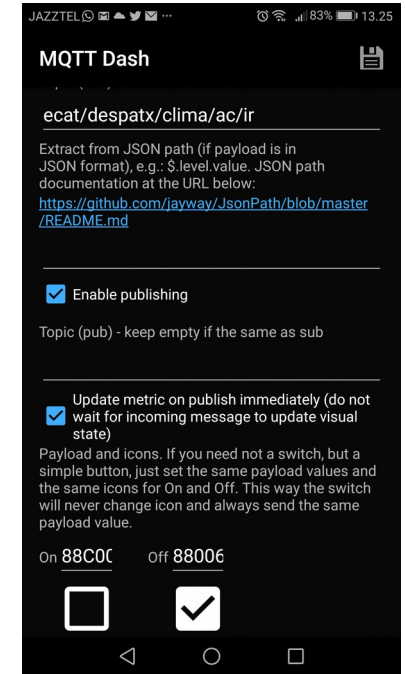
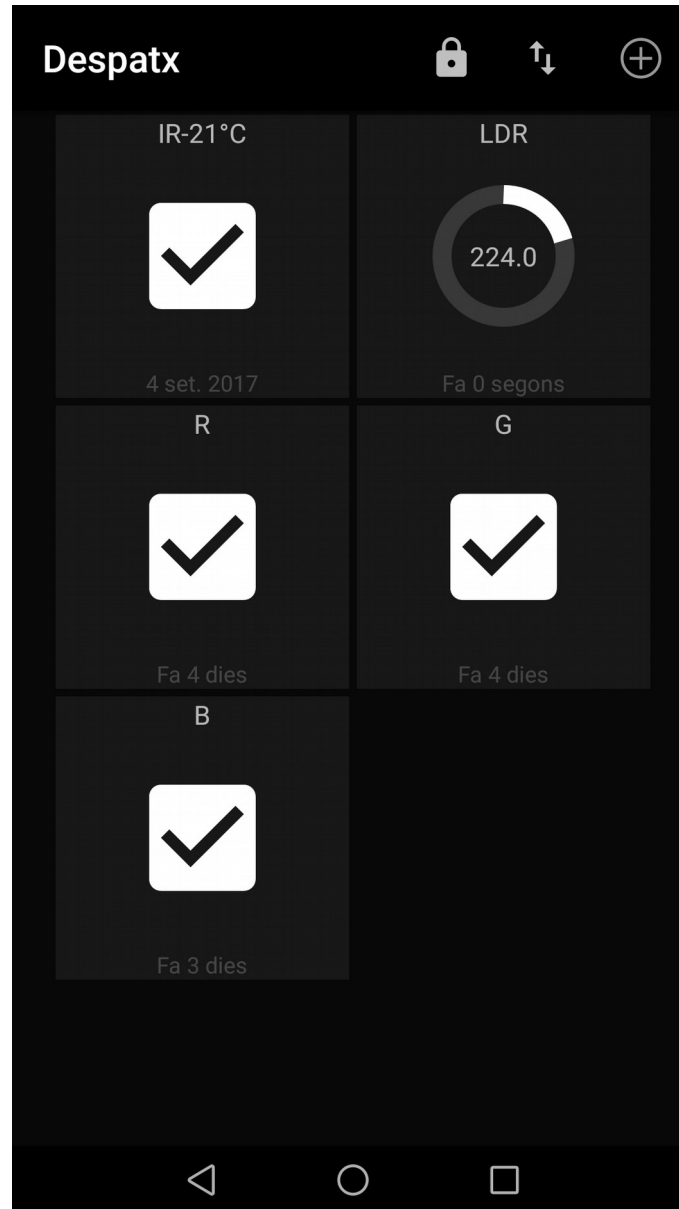
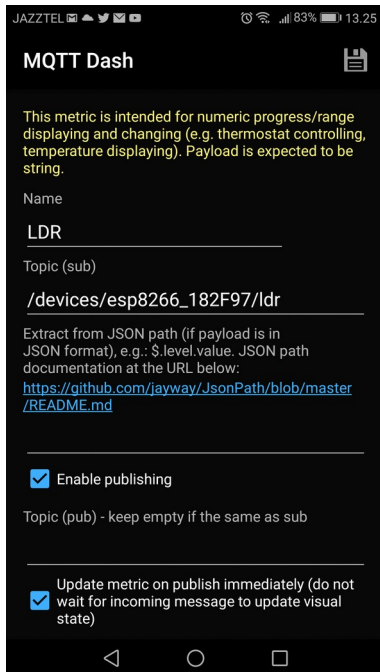
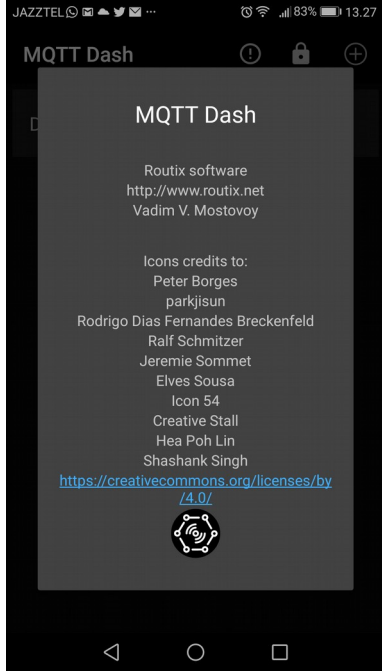
Panells de control - NodeRED



e⚙ Sistema domòtic amb Internet de les Coses

Panells de control - MQTT Dash

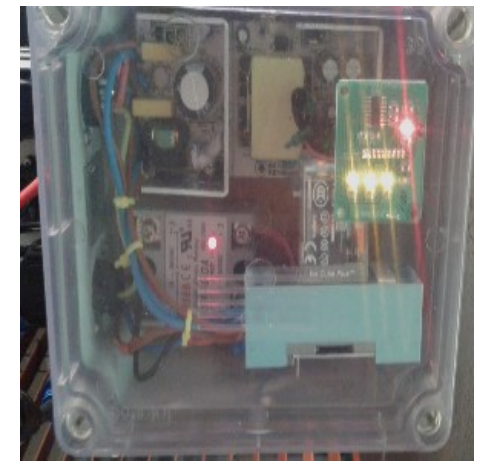
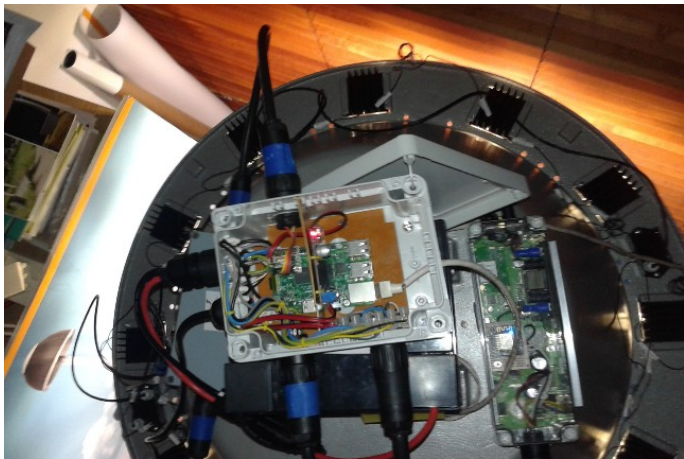
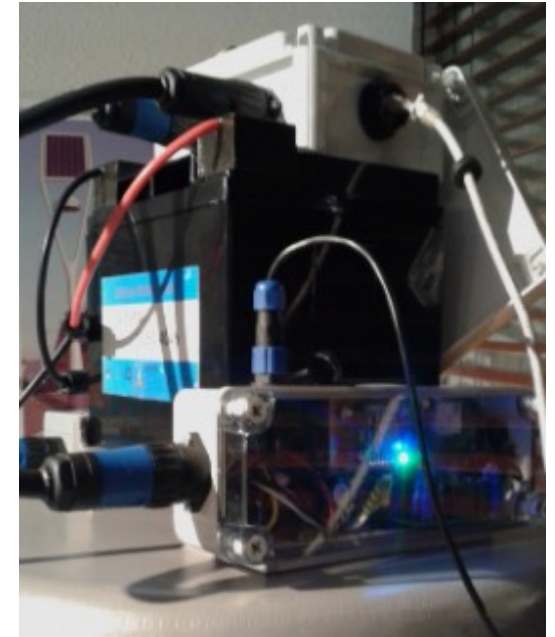
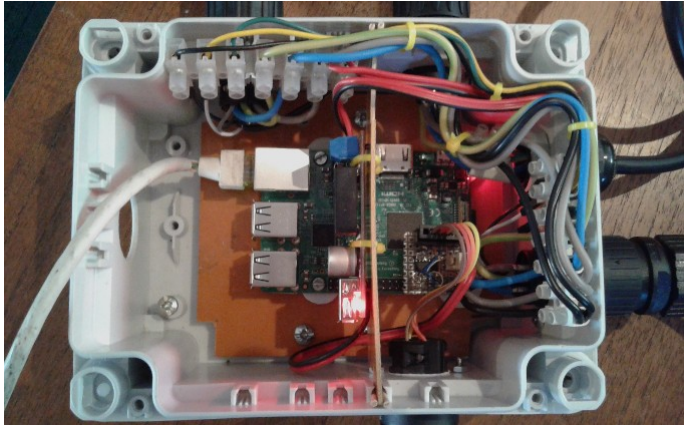
29



Enllaç a l'aplicació

e⚙ Sistema domòtic amb Internet de les Coses 30

Aplicació pràctica - Fanals fotovoltaics



<http://siarq.com/>
<https://binefa.cat/blog/?p=470>
<http://www.electronics.cat>

e⚙ Sistema domòtic amb Internet de les Coses³¹

Panells de control - thethings.io

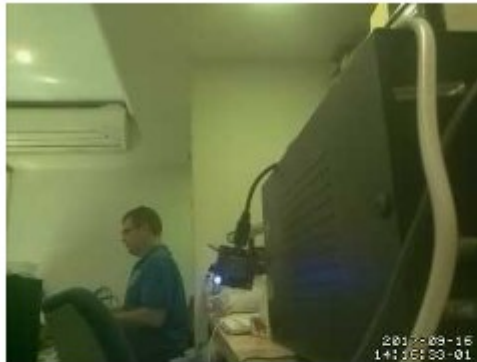


<https://binefa.cat/blog/?p=470>

<https://thethings.io/>

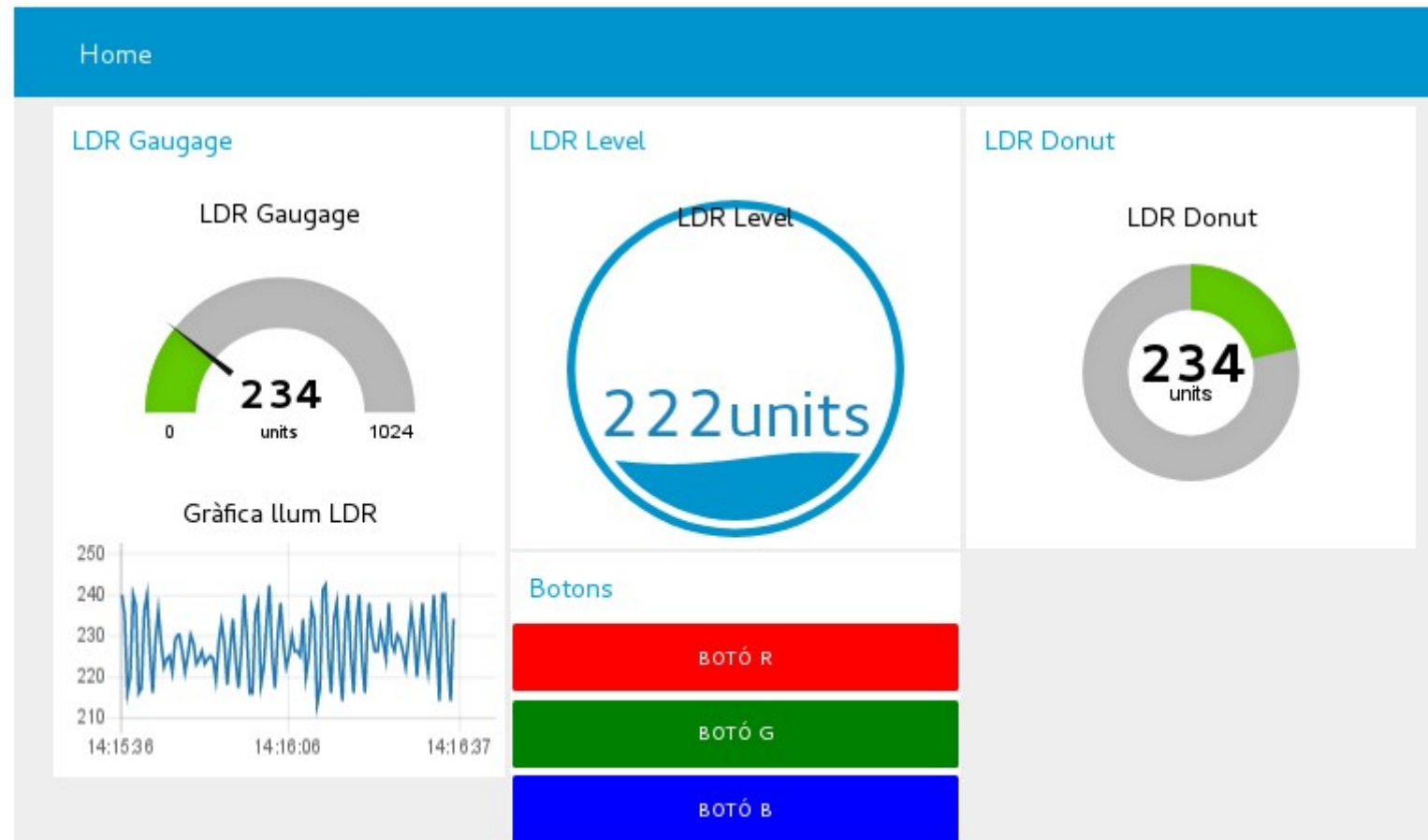
e⚙ Sistema domòtic amb Internet de les Coses

Aplicació pràctica - Control domòtic



Aire condicionat
21°C Apaga

[Surt de la sessió](#)



e⚙ Sistema domòtic amb Internet de les Coses

Aplicació pràctica - Control domòtic

33

Apagat: 88C0051

20°C: 8800549

21°C: 8808642

Encès a 21°C: 880064A

22°C: 8808743

Encès a 22°C: 880074B

23°C: 8808844

Encès a 23°C: 880084C

24°C: 8808945

Encès a 24°C: 880094D

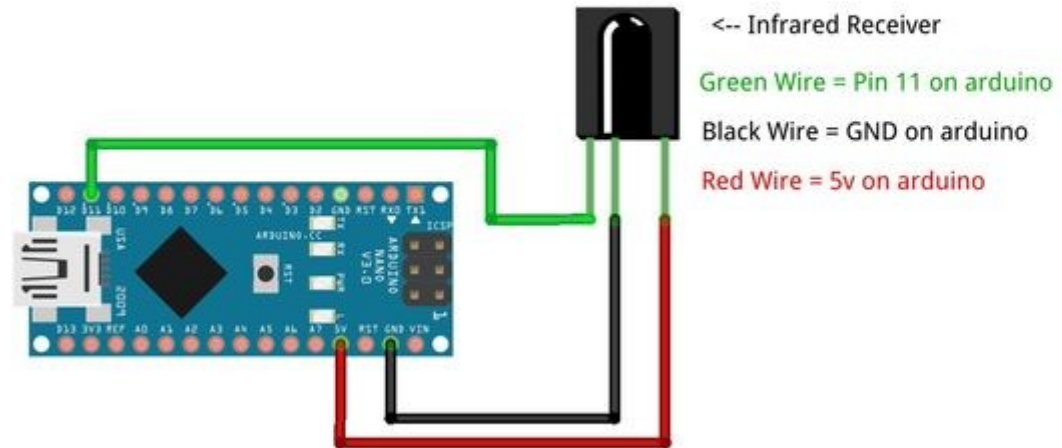
25°C: 8808A46

26°C: 8808B47

27°C: 8808C48

28°C: 8808D49

Diagram to use with Easy Decode sketch



fritzing

e⚙ Sistema domòtic amb Internet de les Coses

Aplicació pràctica - Control domòtic

34

ac_lg_02

```
// Controlling LG's AC
// http://blog.binefa.cat
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <IRremoteESP8266.h>
#include <IRsend.h>
```

```
#define WITTY_BUILTIN_LED 2
#define WITTY_BUTTON 4
#define R_RGB_LED 15
#define G_RGB_LED 13
#define B_RGB_LED 12
#define IR_SEND 14
```

```
#define TOPIC_LED "ecat/despatx/clima/ac/led"
#define TOPIC_IR "ecat/despatx/clima/ac/ir"
IRsend irsend(IR_SEND);
```

```
const char* ssid = "pvSL03";
const char* password = "clotClot";
const char* mqtt_server = "192.168.43.1";
```

```
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
```

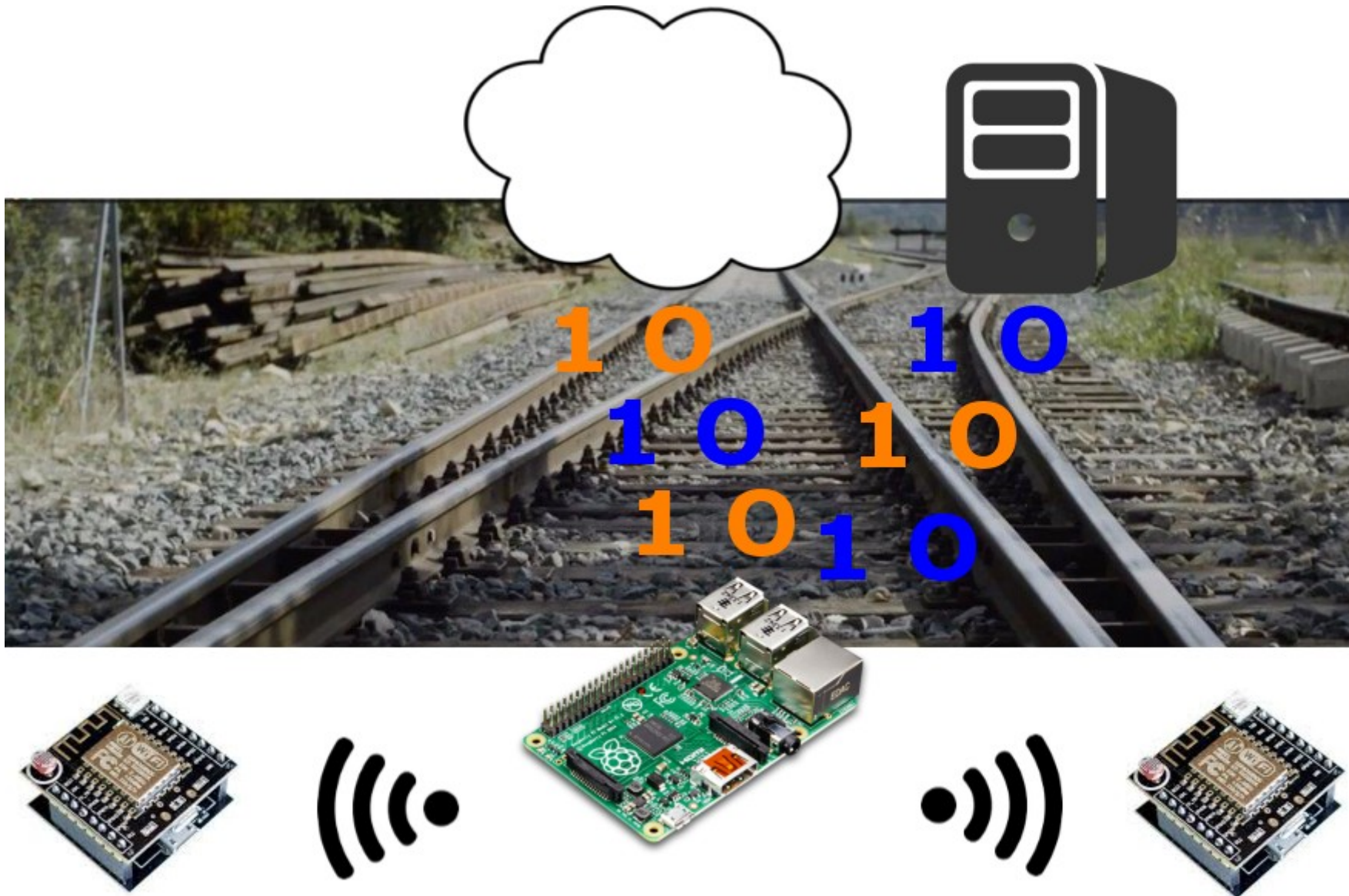
```
if(szTopic == TOPIC_IR){
  vIrSend(strtol(cPayload, NULL, 16));
  if (szPayload == "88C0051") {
    digitalWrite(B_RGB_LED, LOW);
  }
  if (szPayload == "880064A") {
    digitalWrite(B_RGB_LED, HIGH);
  }
}
```

```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("ESP8266Client_AC_LG")) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish("outTopic", "hello world");
      // ... and resubscribe
      client.subscribe(TOPIC_LED);
      client.subscribe(TOPIC_IR);
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

void loop() {
  if(WiFi.status() != WL_CONNECTED){
    setup_wifi();
  }
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```

e⚙ Sistema domòtic amb Internet de les Coses³⁵

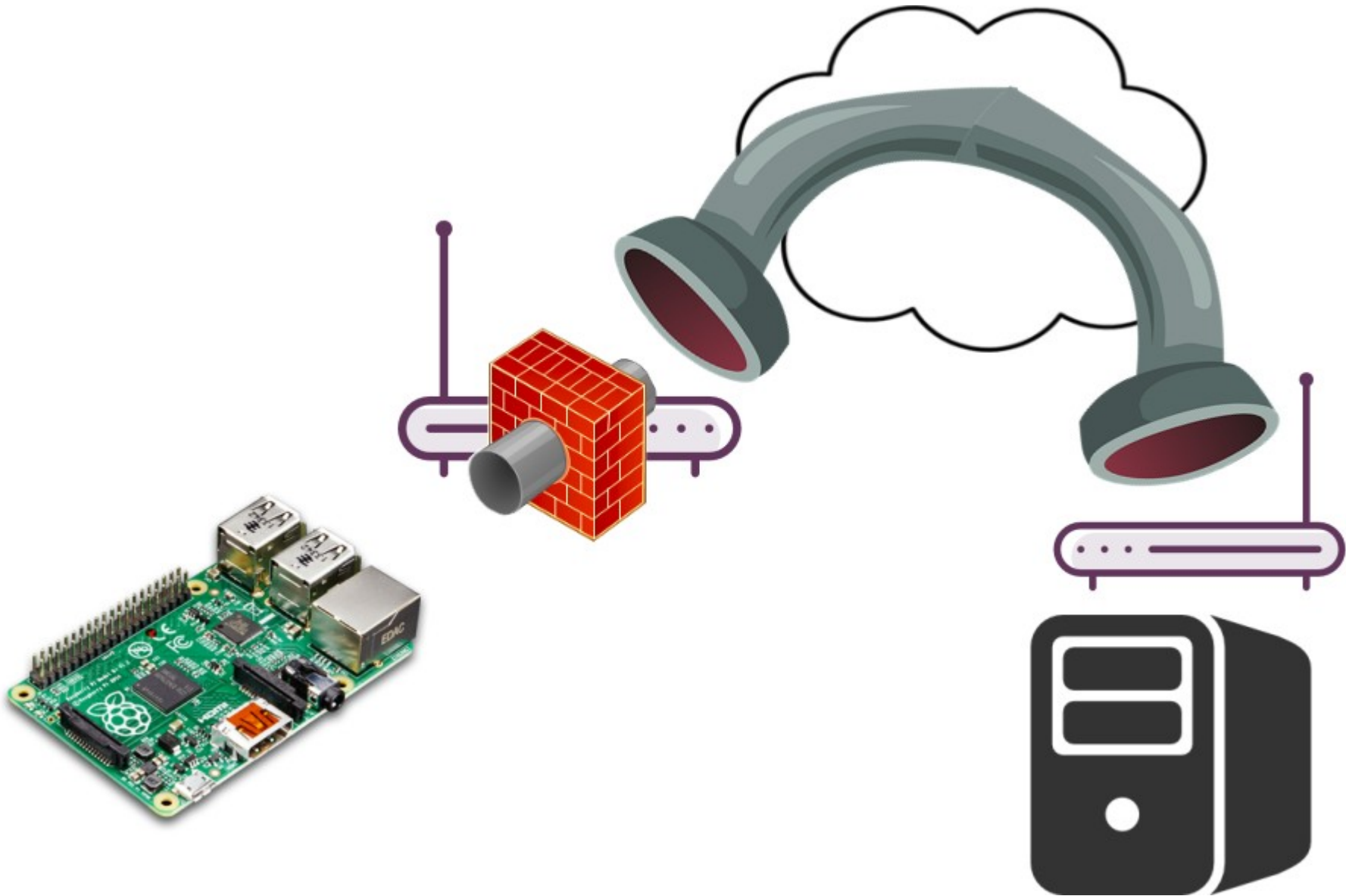
Aplicació pràctica - Control domòtic



e⚙ Sistema domòtic amb Internet de les Coses

Aplicació pràctica - Control domòtic

36



Comunicacions amb Raspberry Pi

SSH

ssh -X pi@192.168.1.17

```
jordi@debianJB:~$ ssh -X pi@192.168.1.17
pi@192.168.1.17's password:
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
Last login: Wed Nov 30 10:09:47 2016 from 192.168.1.12
```

```
pi@raspberrypi:~$ █
```

L'opció -X ens permet poder executar remotament programes amb interfície gràfica d'usuari.

En sistemes operatius Windows existeix el programa putty (<http://www.putty.org/>)

e⚙ Comunicacions amb Raspberry Pi

Accés SSH sense contrasenya

<https://docs.webfaction.com/user-guide/access.html>

Genereu una clau al vostre ordinador (si no existeix `~/ .ssh/id_rsa.pub`):

- 1- Obriu una sessió de terminal.
- 2- Creeu la carpeta `~/ .ssh`, en cas de que no existeixi. (`mkdir -p $HOME/.ssh`)
- 3- Aneu al directori `~/ .ssh` (`cd ~/ .ssh` i premeu Enter)
- 4- Genereu les vostres claus (`ssh-keygen -t rsa`)
- 5- Premeu Enter per a contestar totes les preguntes per defecte.

Desplegueu la clau a la Raspberry Pi

- 1- Copieu la clau a la Raspberry Pi. Entreu `scp ~/ .ssh/id_rsa.pub pi@192.168.1.17:temp_id_rsa_key.pub` (canvieu 192.168.1.17 per la IP de la vostra Raspberry Pi) i premeu Enter.
- Entreu la contrasenya quan us ho demani (la contrasenya per defecte és raspberry).

```
jordi@debianJB:~$ scp ~/ .ssh/id_rsa.pub pi@192.168.1.17:temp_id_rsa_key.pub
pi@192.168.1.17's password:
id_rsa.pub                                100% 396      0.4KB/s   00:00
jordi@debianJB:~$ █
```

e⚙ Comunicacions amb Raspberry Pi

Accés SSH sense contrasenya

Desplegueu la clau a la Raspberry Pi

2- Obriu una sessió SSH a la vostra Raspberry Pi des del vostre ordinador. Entreu **ssh pi@192.168.1.17** (canvieu 192.168.1.17 per la IP de la vostra Raspberry Pi) i premeu Enter. Entreu la contrasenya quan us ho demani (la contrasenya per defecte és **raspberrypi**).

```
jordi@debianJB:~$ ssh pi@192.168.1.17
pi@192.168.1.17's password:
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
Last login: Wed Nov 30 10:18:28 2016 from 192.168.1.12
```

```
pi@raspberrypi:~ $ █
```

3- Creeu la carpeta `~/ .ssh`, en cas de que no existeixi. (`mkdir -p $HOME/.ssh`)

4- Afegiu la vostra clau a l'arxiu `authorized_keys` . Entreu `cat ~/temp_id_rsa_key.pub >> ~/.ssh/authorized_keys` i premeu Enter.

```
pi@raspberrypi:~ $ cat ~/temp_id_rsa_key.pub >> ~/.ssh/authorized_keys
```

e⚙️ Comunicacions amb Raspberry Pi

Accés SSH sense contrasenya

Desplegueu la clau a la Raspberry Pi

5- Esborreu l'arxiu amb la clau temporal. Feu `rm temp_id_rsa_key.pub` i premeu Enter.

```
pi@raspberrypi:~ $ rm temp_id_rsa_key.pub
```

6- Protegiu l'arxiu de claus SSH. Entreu-hi `chmod 600 ~/.ssh/authorized_keys` i premeu Enter.

```
pi@raspberrypi:~ $ chmod 600 ~/.ssh/authorized keys
```

7- Protegiu el directori SSH. Entreu-hi `chmod 700 ~/.ssh` i premeu Enter.

```
pi@raspberrypi:~ $ chmod 700 ~/.ssh
```

8- Protegiu el vostre directori d'usuari. Entreu-hi `chmod go-w $HOME` i premeu Enter.

```
pi@raspberrypi:~ $ chmod go-w $HOME
```

9- Tanqueu la sessió SSH.

e⚙️ Comunicacions amb Raspberry Pi

Accés SSH sense contrasenya

Desplegueu la clau a la Raspberry Pi

```
pi@raspberrypi:~ $ cat ~/temp_id_rsa_key.pub >> ~/.ssh/authorized_keys
pi@raspberrypi:~ $ rm temp_id_rsa_key.pub
pi@raspberrypi:~ $ chmod 600 ~/.ssh/authorized_keys
pi@raspberrypi:~ $ chmod 700 ~/.ssh
pi@raspberrypi:~ $ chmod go-w $HOME
pi@raspberrypi:~ $ exit
logout
Connection to 192.168.1.17 closed.
jordi@debianJB:~$ █
```

**Verifiqueu que ja podeu accedir mitjançant SSH sense que us
pregunti la contrasenya**

```
jordi@debianJB:~$ ssh pi@192.168.1.17

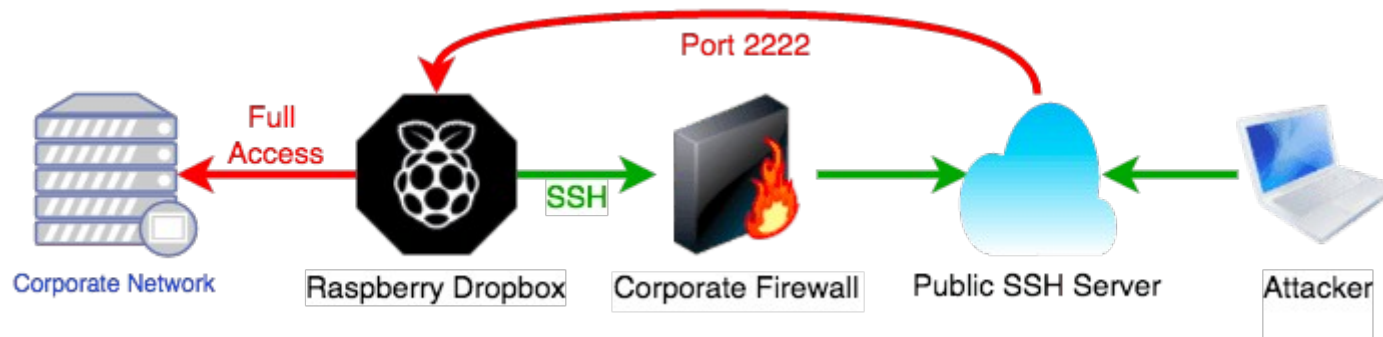
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Nov 30 11:03:54 2016 from 192.168.1.12
pi@raspberrypi:~ $ █
```

e⚙️ Comunicacions amb Raspberry Pi

Túnel SSH invers

<http://jerrygamblin.com/2016/04/23/persistent-reverse-ssh-tunnels-on-a-raspberrypi/>



Comunicacions amb Raspberry Pi

Túnel SSH invers

```
jordi@debianJB: ~
Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda

pi@raspberrypi:~ $ ssh -R 12345:localhost:22 jordi@192.168.1.12
The authenticity of host '192.168.1.12 (192.168.1.12)' can't be established.
ECDSA key fingerprint is 01:eb:89:96:25:8b:48:22:e9:a7:ed:5e:d4:98:c4:ac.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.12' (ECDSA) to the list of known hosts.
jordi@192.168.1.12's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
jordi@debianJB:~$
```

```
pi@raspberrypi: ~
Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda

jordi@debianJB:~$ ssh -p 12345 pi@127.0.0.1
The authenticity of host '[127.0.0.1]:12345 ([127.0.0.1]:12345)' can't be established.
ECDSA key fingerprint is db:79:95:3e:5d:97:96:cd:ce:57:64:96:63:82:15:e2.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[127.0.0.1]:12345' (ECDSA) to the list of known hosts.

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Nov 30 23:46:52 2016 from 192.168.1.12
pi@raspberrypi:~ $
```

Comunicacions amb Raspberry Pi

autoSSH

Si el túnel deixa de funcionar (acostuma a ocórrer en xarxes molt ocupades o d'escasa qualitat, com el 3G), es pot utilitzar el paquet autossh en lloc de l'ssh per a establir la connexió que s'encarregarà de mantenir el túnel obert reiniciant automàticament la connexió.

```
pi@raspberrypi:~ $ autossh -M 65500 -o ServerAliveInterval=20 -R 19994:localhost:22 ecat@web398.webfaction.com
```

L'autossh no funciona com s'espera si al connectar la Raspberry Pi no hi ha connexió a Internet. Per a evitar això, primer verifiquem la connectivitat fent un ping.

```
jordi@debianJB:~$ ssh ecat@web398.webfaction.com
ecat@web398.webfaction.com's password:
Last login: Wed Nov 30 23:21:27 2016 from 62.83.215.143
[ecat@web398 ~]$ ssh -p 19994 pi@localhost
pi@localhost's password:
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
Last login: Thu Dec  1 00:24:02 2016 from localhost
```

```
pi@raspberrypi:~ $ █
```

e⚙ Sistema domòtic amb Internet de les Coses 45

Aplicació pràctica - Control domòtic

```
pi@rpi-ecat00:~/codis $ cat /home/pi/codis/sshRemot.sh
#!/bin/bash

ORDRE="autossh -f -nNT -M 61100 -o ServerAliveInterval=20 -R 12346:localhost:22 ecat@web577.webfaction.com"
#echo $ORDRE
timeout 2 ssh -o ServerAliveInterval=20 ecat@web577.webfaction.com "netstat -putan | grep -v grep | grep 12346"
if [ $? -ne 0 ]; then
    timeout 2 $ORDRE
    echo $ORDRE
else
    echo "autossh amb webfaction en marxa"
fi
timeout 2 ps aux | grep autossh | grep 18880:localhost:1883
if [ $? -ne 0 ]; then
    timeout 2 autossh -f -nNT -M 59900 -o ServerAliveInterval=20 -R 18880:localhost:1883 ecat@web577.webfaction.com
fi

ORDRE="autossh -f -nNT -M 58800 -o ServerAliveInterval=20 -R 12336:localhost:22 popotamo@popotamo.binefa.cat -p 2244"

pi@rpi-ecat01:~/codis $ cat sshRemot.sh
#!/bin/bash

ORDRE="autossh -f -nNT -M 60000 -o ServerAliveInterval=20 -R 12347:localhost:22 ecat@web577.webfaction.com"
#echo $ORDRE
timeout 2 ssh -o ServerAliveInterval=20 ecat@web577.webfaction.com "netstat -putan | grep 12347"
if [ $? -ne 0 ]; then
    timeout 2 $ORDRE
    echo $ORDRE
else
    echo "autossh en marxa"
fi
```


e⚙ Sistema domòtic amb Internet de les Coses

Aplicació pràctica - Control domòtic

46

```
[ecat@web577 bin]$ cat /home/ecat/bin/tunel/estableix03.sh
#!/bin/bash

timeout 2 ssh -p 12346 pi@localhost "ls"
if [ $? -eq 0 ]; then
    echo "Hi ha connexió amb la RPi pel port 12346"
    ps aux | grep 13769:localhost:1880 | grep -v grep
    if [ $? -ne 0 ]; then
        ssh -N -o ServerAliveInterval=20 -L *:13769:localhost:1880 pi@localhost -p 12346 &
        echo $! > /home/ecat/bin/tunel/rt.pid
    else
        echo "Túnel 13769 [wf] <--> 1880 [RPi] establert"
    fi
else
    echo "No hi ha connexió amb la RPi pel port 12346"
    ps aux | grep 13769:localhost:1880 | grep -v grep
    if [ $? -ne 0 ]; then
        echo "Tampoc ha quedat el túnel 13769 [wf] <--> 1880 [RPi] establert"
    else
        echo "Túnel 13769 [wf] <--> 1880 [RPi] establert en segon terme. El matem."
        kill `cat /home/ecat/bin/tunel/rt.pid`
    fi
fi
```

```
[ecat@web577 bin]$ cat /home/ecat/bin/tunel/estableix04.sh
#!/bin/bash

timeout 2 ssh -p 12347 pi@localhost "ls"
if [ $? -eq 0 ]; then
    echo "Hi ha connexió amb la RPi pel port 12347"
    ps aux | grep 20271:localhost:8081 | grep -v grep
    if [ $? -ne 0 ]; then
        ssh -N -o ServerAliveInterval=20 -L *:20271:localhost:8081 pi@localhost -p 12347 &
        echo $! > /home/ecat/bin/tunel/rt04.pid
    else
        echo "Túnel 20271 [wf] <--> 8081 [RPi] establert"
    fi
else
```

e⚙ Sistema domòtic amb Internet de les Coses 47

Aplicació pràctica - Control domòtic

Build a Raspberry Pi Webcam Server in Minutes

by Gus | 📅 Feb 28, 2015 | 📷 Camera, Intermediate, Servers | 💬 264 comments



<https://pimylifeup.com/raspberry-pi-webcam-server/>



<https://www.linux-projects.org/uv4l/installation/>

e⚙ Sistema domòtic amb Internet de les Coses⁴⁸

Per a aprendre més



Neil Kolban, Texas, USA

<http://neilkolban.com/tech/esp8266/>



Andreas Spiess

41.990 subscribers

Canal d'Andreas Spiess
a YouTube

Torn de preguntes ...





Sistema domòtic amb Internet de les Coses

Presentació descarregable a : <http://binefa.cat/blog>

Correu electrònic de contacte : jordibinefa@electronics.cat

twitter



<https://twitter.com/JordiBinefa>

<https://twitter.com/electronicscat>



<http://es.linkedin.com/pub/jordi-binefa/13/717/90b>

Plaques disponibles a :

<http://www.electronics.cat>

<http://www.makeit.cat>



Moltes gràcies per la vostra atenció