



THE THINGS
NETWORK

TALLER DE CONSTRUCCIÓN DE GATEWAY Y NODO SINGLE CHANNEL

Juan Félix Mateos

juanfelixmateos@gmail.com

Makespace Madrid, 18 de enero de 2020

PROGRAMACIÓN DEL TALLER

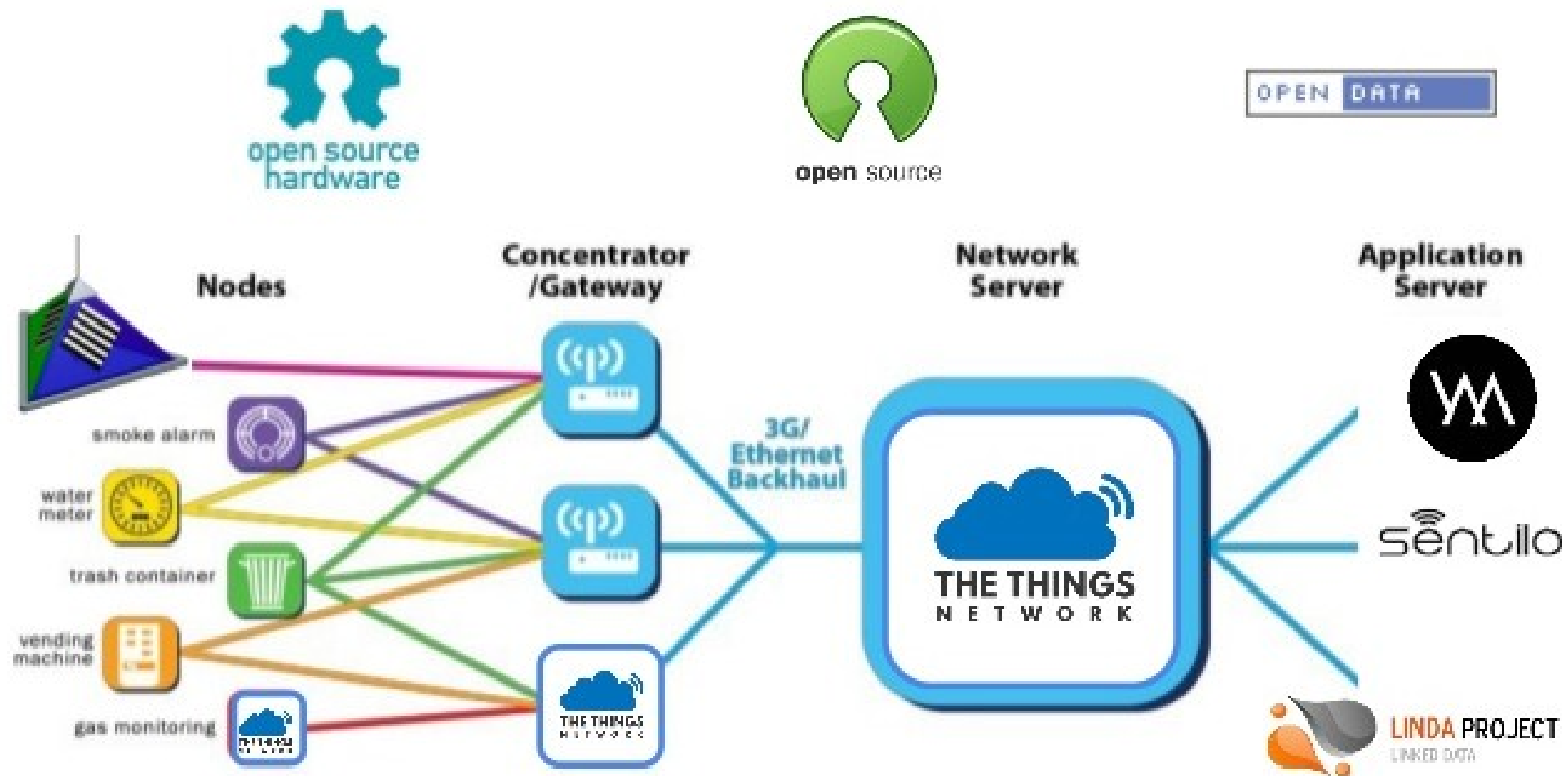
- Presentación de la comunidad The Things Network Madrid
- Normas básicas de seguridad y técnicas de soldadura
- Construcción del gateway
- Programación del gateway
- Crear una cuenta en The Things Network, alta del gateway, y creación de una aplicación para gestionar el nodo
- Construcción del nodo
- Programación del nodo
- Integración del nodo con IFTTT y Cayenne para recibir notificaciones y SMS en nuestro móvil cada vez que el sensor detecte que se ha abierto la puerta
- Preguntas y sugerencias

LA COMUNIDAD The Things Network Madrid

- Divulgación
- Formación
- Asesoramiento
- Implantación

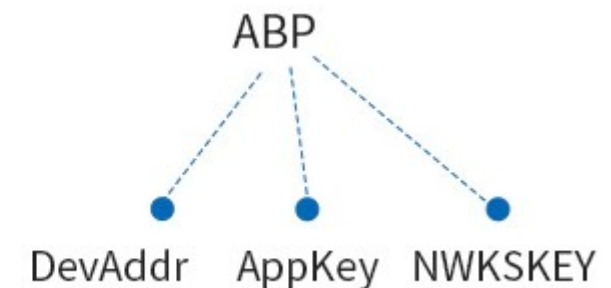
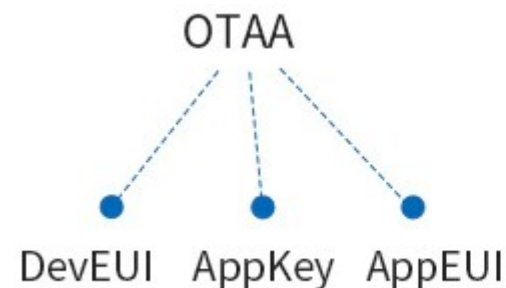
Arquitectura de The Things Network

Nodos, aplicaciones, gateways, backend, y aplicaciones



APLICACIONES Y NODOS

- Aplicación
 - Application EUI
 - Application Key
- Nodo (una aplicación puede contar con varios nodos)
 - Device EUI
 - Activación: **OTAA** (over the air activation) vs **ABP** (activation by personalization)
 - Device Address
 - Network Session Key
 - Application Session Key



SEGURIDAD Y SOLDADURA



Riesgos

- **Químicos**

- Plomo nunca más
- Rosin (Colofonía) ¡Ojo con asmáticos!

- **Térmicos**

- Quemarse (poca importancia salvo que afecte a los ojos)

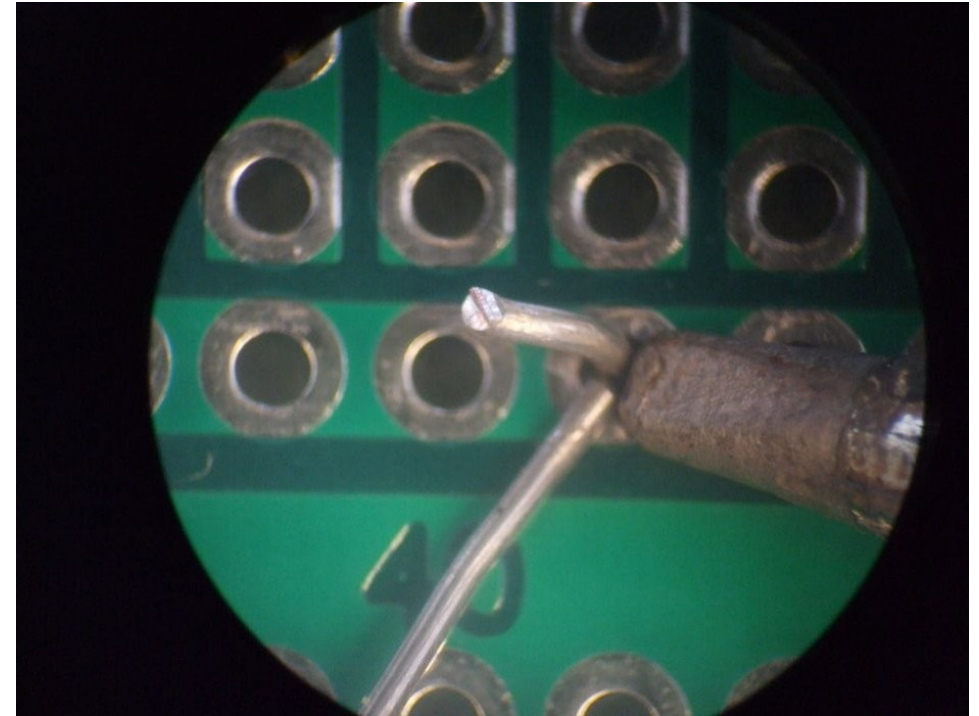
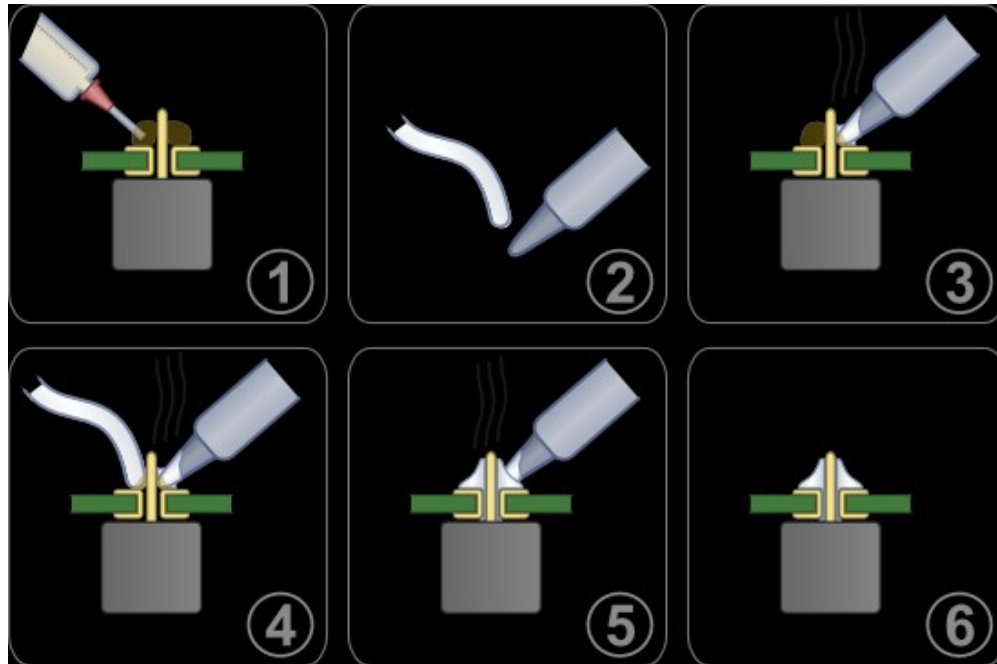
- **Eléctricos**

- Quemar el cable del soldador → Salta el magnetotérmico y susto
- No usar pulseras anti-estáticas → Estorban

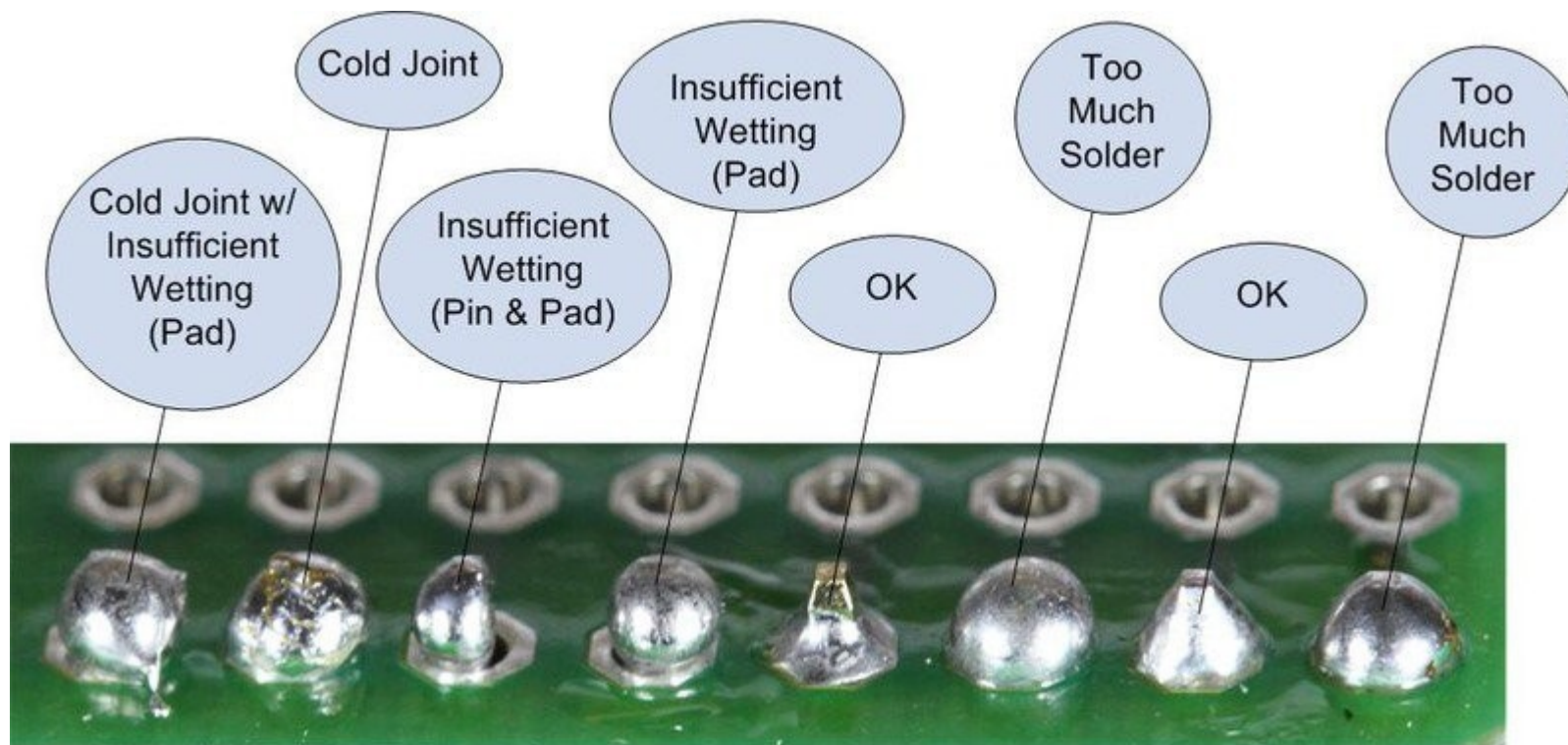
- **Mecánicos**

- Punta del soldador → Pinchar al compañero

¿CÓMO SOLDAR?



SOLDADURA CORRECTA



LISTA DE MATERIALES

- Disponible en el repositorio:
 - https://github.com/IoTopenTech/thethingnetwork-madrid-taller-gateway-nodo-single-channel/blob/master/thethingnetwork-madrid-taller-gateway-nodo-single-channel_BOM.ods

<u>Cantidad</u>	<u>Descripción</u>
1	<u>Wemos D1 mini</u>
1	<u>Arduino Pro Mini Compatible ATmega328 3v3/8MHz</u>
2	<u>RFM95W SX1276 868MHz</u>
1	<u>Conector antena RP-SMA</u>
1	<u>Antena 868MHz 3db</u>
1	<u>Antena helicoidal 868MHz</u>
1	<u>Sensor magnético de láminas NO</u>
1	<u>Clips batería AA</u>
2	<u>Condensadores electrolíticos 10uF</u>
1	<u>Condensador electrolítico 1uF</u>
1	<u>Cargador de teléfono móvil microUSB</u>
1	<u>Placa adaptadora RFM95W Gateway</u>
1	<u>Placa adaptadora RFM95W nodo</u>
1	<u>Carcasa gateway impresa en 3D</u>
1	<u>Carcasa nodo impresa en 3D</u>

CONSTRUCCIÓN DEL GATEWAY

1. Soldar los pines macho-hembra en el wemos dejando el conector microusb del lado de las hembras
2. Soldar los componentes de la placa adaptadora del gateway en el siguiente orden:
 1. RFM95W
 2. Conector RP-SMA
 3. Condensadores de 10uF y 100uF (atención a la polaridad). Estos 2 componentes son opcionales.
 4. Pines hembra hacia el lado de la cara de componentes

Wemos



PROGRAMACIÓN DEL GATEWAY

Requisitos previos

1. Instalar una versión reciente del entorno de desarrollo Arduino.
 - <https://www.arduino.cc/en/Main/Software>
2. Configurar en el IDE Arduino el gestor de placas ESP8266
 - http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Descargar el repositorio del taller:
 - https://github.com/IoTopenTech/thethingnetwork_madrid_taller_gateway_nodo_single_channel

PROGRAMACIÓN DEL GATEWAY

Configuración de la ubicación

4. En el IDE de Arduino, abrir el archivo (se encuentra en el repositorio descargado en el paso 3):
 - ESP-1ch-Gateway-v6.0-master\ESP-sc-gway\ESP-sc-gway.ino
5. Seleccionar el archivo configGway.h
6. Introducir la información oportuna en las líneas 226-232 (las coordenadas puedes obtenerse con la opción ¿Qué hay aquí? De Google Maps)

```
226 // Gateway Ident definitions. Where is the gateway located?
227 #define _DESCRIPTION "Single Channel 868.1MHz SF7" // Name of the gateway
228 #define _EMAIL "juanfelixmateos@gmail.com" // Owner
229 #define _PLATFORM "ESP8266"
230 #define _LAT 40.39927
231 #define _LON -3.7022
232 #define _ALT 14 // Altitude
```


PROGRAMACIÓN DEL GATEWAY

Configuración de las credenciales WiFi

1. En el archivo configNode.h, introducir la información oportuna a partir de la línea 125 (pueden configurarse varias SSID, pero la línea 126 debe dejarse intacta)

Nota: Esto permite la comunicación entre el Gateway y la red. Los datos enviados por el Nodo serán recogidos por el Gateway y enviados a la consola TTN.

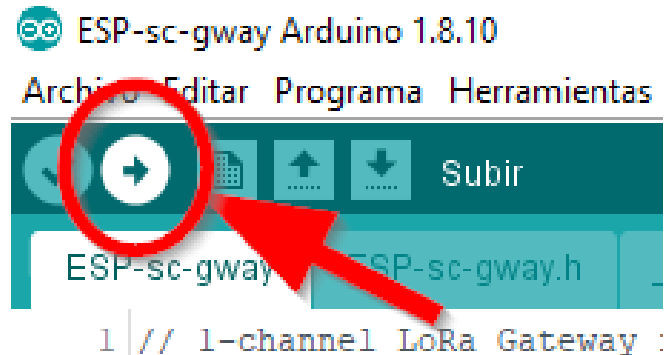
```
125 wpas wpa[] = {  
126     { "", "" }, // Reserved for WiFi Manager  
127     { "WiFil", "LaContraseñaDeLWiFil" },  
128     { "WiFi2", "LaContraseñaDeLWiFi2" }  
129 };  
---
```

PROGRAMACIÓN DEL GATEWAY

Subir el firmware al gateway

7. Conectar el Wemos al ordenador (sin acoplarle la placa adaptadora del gateway) y programarlo con el IDE de Arduino

- Si es necesario, instalar las librerías no encontradas usando como referencia la sección Getting Started del README del repositorio



- LoRaCode (Version 1.0.0, see library shipped)
- gBase64 (changed name from Adam Rudd's Base64 version)
- TinyGPS++ (Version 1.0.0)

Through Library Manager:

- ArduinoJson (version 6.12.0)
- Heltec ESP32 Dev-Boards (Version 1.0.6)
- Heltec ESL8266 Dev-Boards (Version 1.0.2)
- SPI (Version 1.0.0)
- SPIFFS (Version 1.0.0)
- Streaming (Version 5.0.0)
- Ticker (Version 1.1.0)
- Time (Version 1.5.0)
- Update (Version 1.0.0)
- Webserver (Version 1.0.0)
- WiFiClientSecure (Version 1.0.0)
- WifiEsp (Version 2.2.2)
- Wire (Version 1.0.1)
- WifiManager (Version 0.12.0 by Tzapu) for ESP8266

PROGRAMACIÓN DEL GATEWAY

Montar el gateway

1. Acopar el wemos y la placa adaptadora (los conectores microusb y RP-SMA quedan orientados hacia el mismo lado).
2. Enroscar la antena (nunca debe alimentarse un módulo de radio sin antena; podría dañarse irreversiblemente)



PROGRAMACIÓN DEL GATEWAY


Averiguar el Gateway ID

1. Con el gateway conectado al ordenador, abrir el Monitor Serie del IDE de Arduino configurado a 115200 baudios.
2. Tomar nota del Gateway ID (16 dígitos hexadecimales).

```
COM12

.
Error Wifi network connect
0:1:3. WiFi connect SSID=[REDACTED], pass=[REDACTED]
Host=esp8266-124[REDACTED] WiFi Connected to [REDACTED] on IP=192.168.1.151
Gateway ID [REDACTED]. Listening at SF7 on 868.10 MHz.
writeGwayCfg: Saturday 21:58:39
WWW Server started on port 80
WARNING:: writeSeen, file not exists, formatting /gwayNum.txt
OLED_ADDR=0x3C
-----
```


ALTA DEL GATEWAY EN THE THINGS NETWORK

 **THE THINGS NETWORK** **CONSOLE**
COMMUNITY EDITION

ApplicationsGateways

Gateways > Register

REGISTER GATEWAY

Gateway EUI
The EUI of the gateway as read from the LoRa module

2C1F8 bytes

☒ I'm using the legacy packet forwarder
Select this if you are using the legacy [Semtech packet forwarder](#).

Description
A human-readable description of the gateway

WEMOS 868.1MHz|SF7

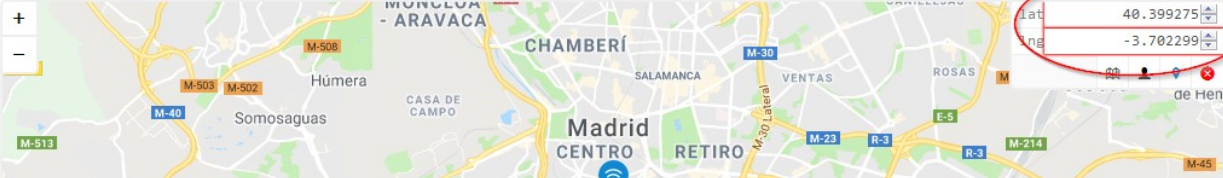
Frequency Plan
The [frequency plan](#) this gateway will use

Europe 868MHz


Router
The router this gateway will connect to. To reduce latency, pick a router that is in a region which is close to the location of the gateway.

ttn-router-eu

Location
The exact location of you gateway. This will be used if your gateway cannot determine its location by itself. Set a location by clicking on the map.



CREACIÓN DE UNA APLICACIÓN PARA GESTIONAR EL NODO

 **THE THINGS NETWORK** **CONSOLE**
COMMUNITY EDITION

ApplicationsGateways

Applications > Add Application

ADD APPLICATION

Application ID
The unique identifier of your application on the network

jfmateos_sensor_puerta_abierta

Description
A human readable description of your new app

Mi sensor de puerta abierta

Application EUI
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.


EUI issued by The Things Network

Handler registration
Select the handler you want to register this application to

ttn-handler-eu

CancelAdd application

AÑADIR EL NODO (DEVICE) A LA APLICACIÓN

 **THE THINGS NETWORK** **CONSOLE**
COMMUNITY EDITION

ApplicationsGateways

Applications > jfmateos_sensor_puerta_abierta > Devices

OverviewDevicesPayload FormatsIntegrationsDataSettings

REGISTER DEVICE[bulk import devices](#)

Device ID
This is the unique identifier for the device in this app. The device ID will be immutable.

Device EUI
The device EUI is the unique identifier for this device on the network. You can change the EUI later.

App Key
The App Key will be used to secure the communication between you device and the network.

App EUI

CancelRegister

CAMBIAR EL MÉTODO DE ASOCIACIÓN A ABP Y DESACTIVAR EL FRAME COUNTER CHECK

THE THINGS NETWORK CONSOLE COMMUNITY EDITION

Applications > jfmateos_sensor_puerta_abierta > Devices > puerta_principal > Settings

Overview Data Settings

DEVICE SETTINGS

General
Location

SETTINGS

Description
A human-readable description of the device

Device EUI
The serial number of your radio module, similar to a MAC address

Application EUI

Activation Method
OTAA ABP

Device Address
The device address will be assigned

Network Session Key
Network Session Key will

THE THINGS NETWORK CONSOLE COMMUNITY EDITION

Applications > jfmateos_sensor_puerta_abierta > Devices > puerta_principal > Settings

70 B3 D5 7E D0 00 B8 14

Activation Method
OTAA ABP

Device Address
The device address will be assigned by the network server

Network Session Key
Network Session Key will be generated

App Session Key
App Session Key will be generated

Frame Counter Width
16 bit 32 bit

☒ **Frame Counter Checks**
Disabling frame counter checks drastically reduces security and should only be used for development purposes

Delete Device Cancel Save

DATOS NECESARIOS PARA CONECTAR EL NODO

Activation Method **ABP**

Device EUI <> ⇅ 00 B7 94 6C CA 58 CD B5

Application EUI <> ⇅ 70 B3 D5 7E D0 00 AA AE

Device Address <> ⇅ 26

Network Session Key <> ⇅ msb { 0xCE, ... }


App Session Key <> ⇅ msb { 0xD3, ... }

Status ● 3 hours ago

Frames up 139 [reset frame counters](#)

Frames down 0

ESTABLECER EL PAYLOAD FORMAT DE CAYENNE

Applications >  jfmateos_puerta_abierta > Payload Formats

Overview

Devices

Payload Formats

Integrations

Data

Settings

PAYLOAD FORMATS

Payload Format

The payload format sent by your devices

Cayenne LPP



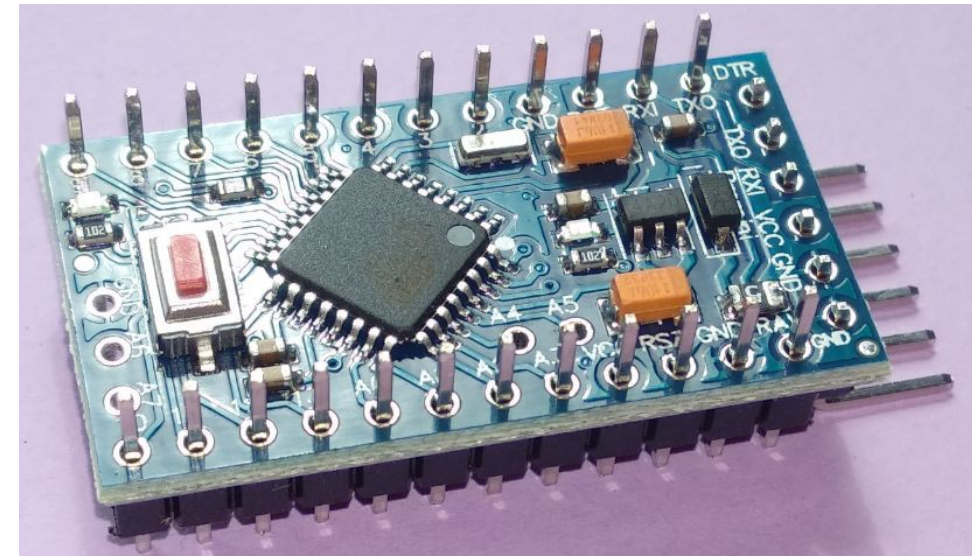
Cancel

no changes to save

CONSTRUCCIÓN DEL NODO

SOLDAR LOS PINES DE LA PLACA TIPO ARDUINO PRO MINI

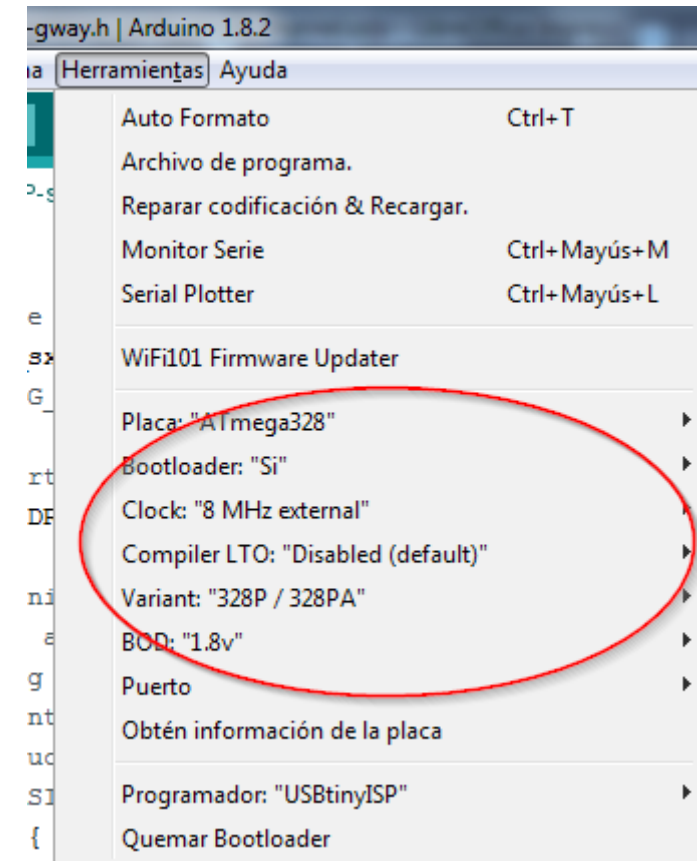
1. Insertar los pines acodados desde la cara de soldadura y soldarlos por la cara de componentes.
2. Insertar los pines macho desde el lado de la cara de soldadura de la placa tipo Arduino Pro Mini, y soldarlos por el lado de la cara de componentes.
 - Evitar dejar restos de estaño en los pines (para poner insertar en ellos cables tipo Dupont si fuera necesario)



CONSTRUCCIÓN DEL NODO

Sustituir el bootloader por MiniCore (opcional)

1. Con el bootloader por defecto, la placa Arduino Pro Mini deja de funcionar cuando el Brown Out Detector (BOD) detecta que la tensión de la batería ha caído por debajo de **2.7V**. Cambiando al bootloader MiniCore podemos configurar una tensión BOD de **1.8V**. Añadir el gestor de placas MiniCore (separarlo con una coma del de ESP8266 en Preferencias):
 - https://mcudude.github.io/MiniCore/package_MCUdude_MiniCore_index.json
2. Configurar la placa como se muestra en la imagen.
3. Quemar el nuevo bootloader usando un programador tipo USBTinyISP

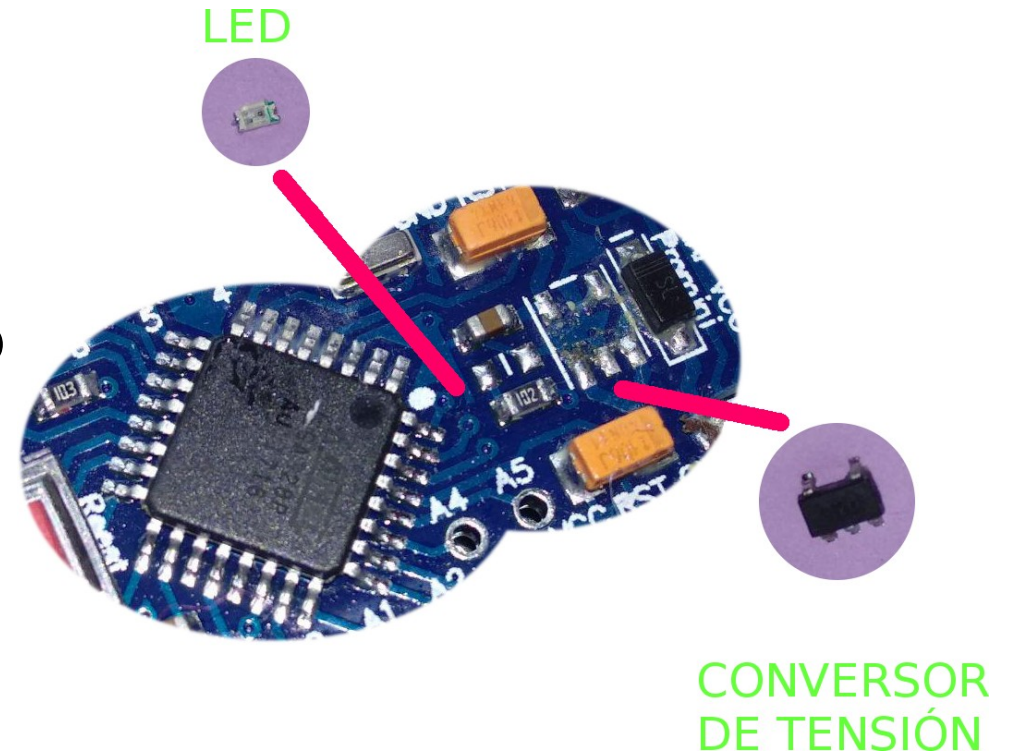


CONSTRUCCIÓN DEL NODO

Configurarlo para bajo consumo de energía (opcional)

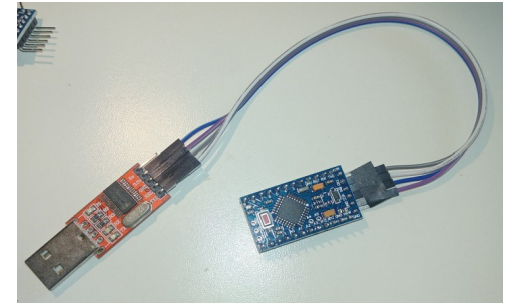
1. Si eliminamos de la placa tipo Arduino Pro Mini el LED y el conversor de tensión conseguiremos reducir el consumo en estado Power Down Sleep de 900uA a 4.5uA, con lo que se extiende enormemente la duración de las baterías.

- Utilizar el soldador de aire caliente para eliminar el LED y el conversor de tensión.



PROGRAMACIÓN DEL NODO

1. Abrir en el IDE de Arduino el archivo Nodo\lora_mini_node_ttn_madrid_abp_puerta_abierta\lora_mini_node_ttn_madrid_abp_puerta_abierta.ino.
2. Insertar los datos de conexión en las líneas 47, 52 y 55.
3. Programar el nodo usando un conversor USB → Serial.



- Según el USB Serial es necesario presionar el botón rojo del Arduino Pro Mini hasta que su estado en el IDE esté en “subiendo” y soltar

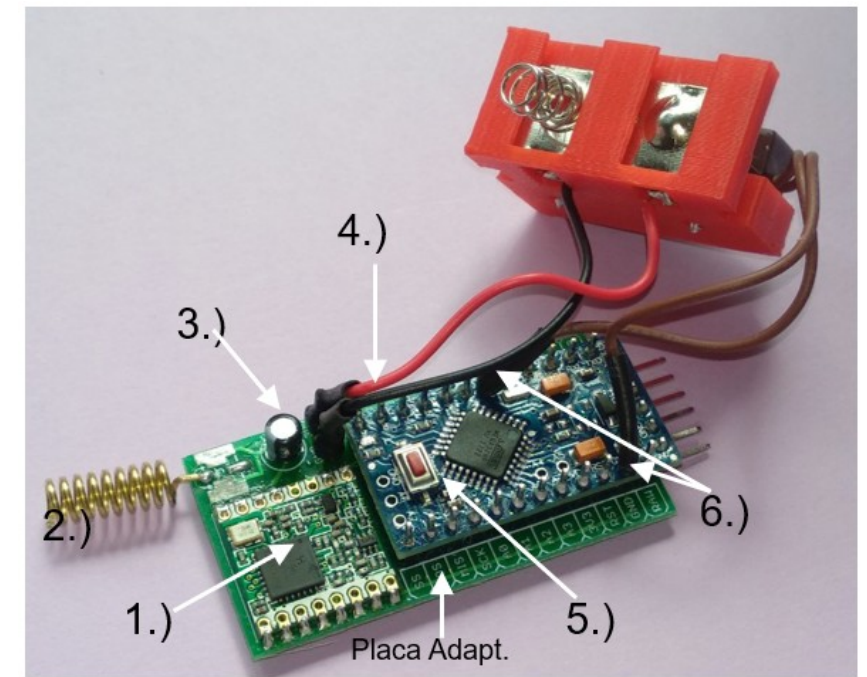
```
45 // This is the default Semtech key, which is used by the early prototype TTN
46 // network.
47 static const PROGMEM u1_t NWKSKEY[16] = { 0xCE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
48
49 // LoRaWAN AppSKey, application session key (formato MSB)
50 // This is the default Semtech key, which is used by the early prototype TTN
51 // network.
52 static const u1_t PROGMEM APPSKEY[16] = { 0xD3, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
53
54 // LoRaWAN end-device address (DevAddr) (formato MSB)
55 static const u4_t DEVADDR = 0x26000000 ; // <-- Change this address for every node!
56
```

TERMINAR EL NODO

1. Soldar los componentes de la placa adaptadora del nodo en el siguiente orden:

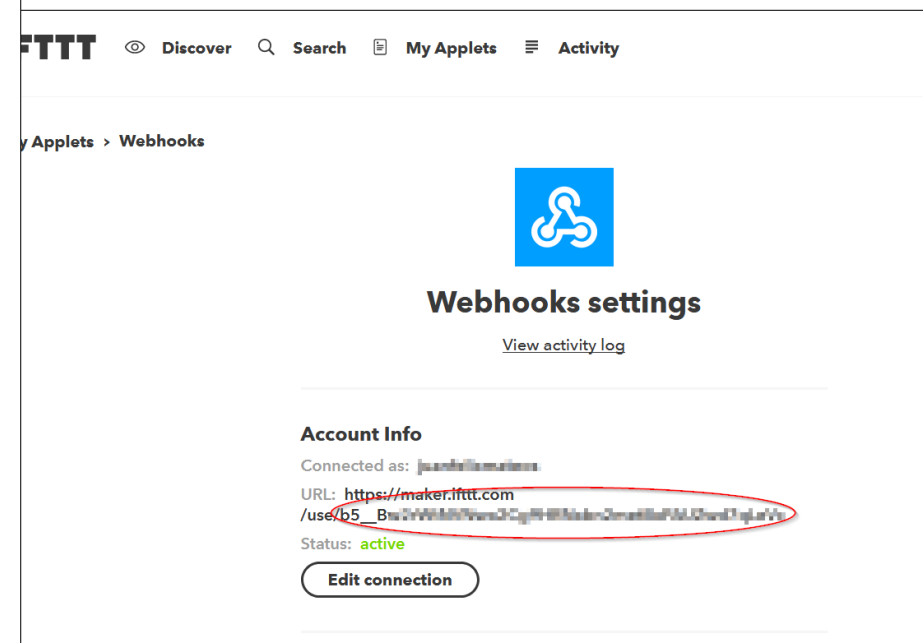
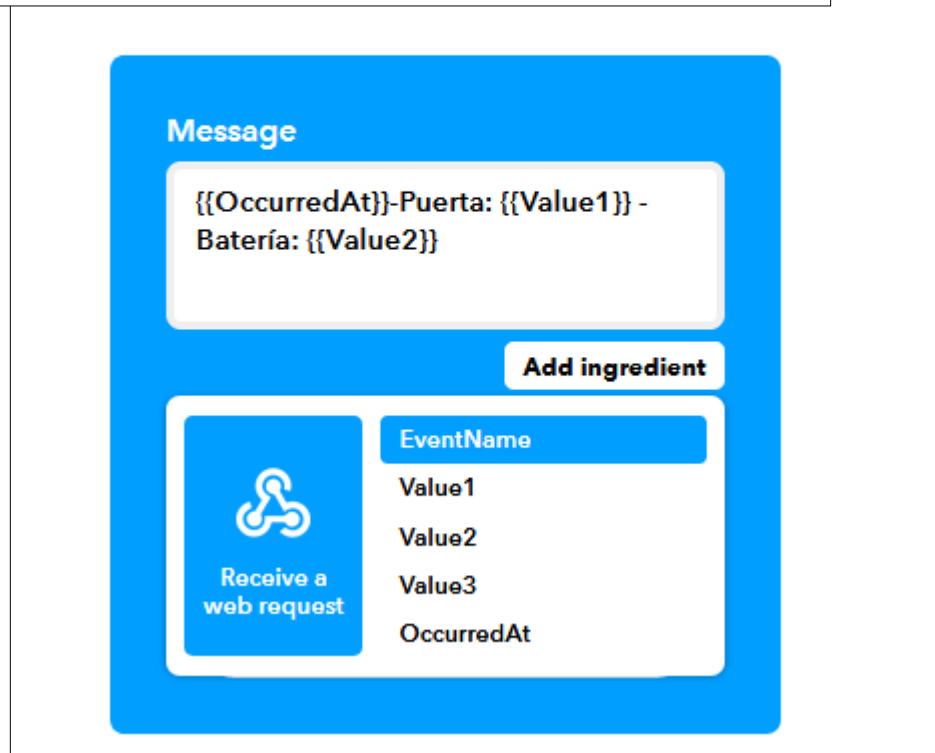
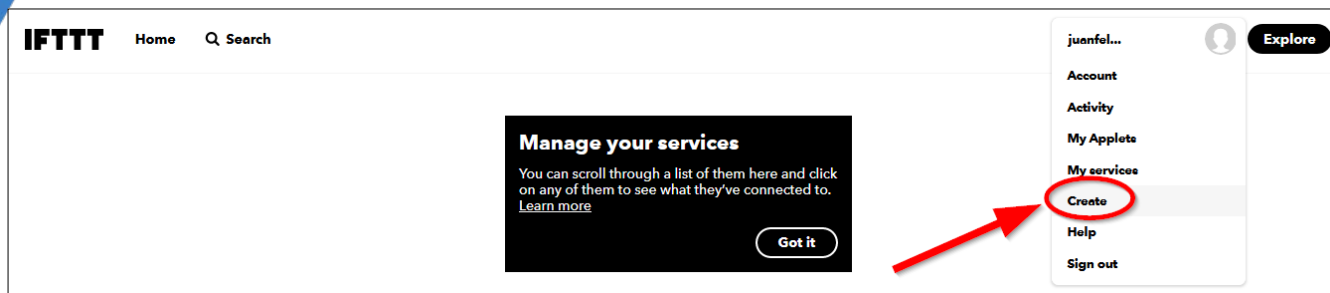
1. RFM95W
2. Antena helicoidal
3. Condensador de 10uF (opcional)
4. Pines GND y 3v3
5. Placa tipo Arduino Pro Mini
6. Cables del sensor reed en los pines 3 y GND, o en las posiciones adyacentes a la entrada de alimentación (paso siguiente).
7. Cables para el portabaterías en los pines GND y 3V3

2. Montar el portabaterías



INTEGRACIÓN IFTTT

CONFIGURACIÓN DEL WEBHOOK EN IFTTT



INTEGRACIÓN IFTTT

CONFIGURACIÓN EN THE THINGS NETWORK

SETTINGS

Event Name
The event name of your IFTTT recipe

puerta_abierta

Key
Your key

b5_Bv

Value 1
Payload field name to send as value 1

digital_in_4

Value 2
Payload field name to send as value 2

analog_in_3

Value 3
Payload field name to send as value 3



PREGUNTAS Y SUGERENCIAS ¡GRACIAS!

<http://www.ttnmad.org>

<http://www.thethingsnetwork.org/community/madrid/>

[@ttn_mad](#)

www.thethingsnetwork.org