

DESPLIEGUE DE RED LORA

Después de haber mostrado y comentado todas las características de las redes LORA vamos a mostrar un ejemplo práctico de despliegue de una red LORA con fines didácticos. Para lo cual se va a utilizar material con un coste contenido (Por debajo de 150€) y de fácil adquisición, de manera que nos permita estudiar de primera mano los detalles de este tipo de dispositivos además de contribuir a la expansión de este tipo de redes.

La red que vamos a desplegar se trata de una red de corto alcance para el estándar LORA en torno a los 500m o 1Km sobre todo al querer limitar el desembolso en material, pero los conceptos y despliegue sería igual en el caso de querer desplegar una red de largo alcance, únicamente deberíamos sustituir algunos elementos de la red por otros orientados a largas distancias.

En primer lugar, vamos a escoger nuestros clientes LORA, en este caso como hemos dicho la potencia de emisión no es un problema por lo que el factor que nos ayudará a decidirnos por un dispositivo u otro será la facilidad de uso y en este caso hemos optado por los módulos Heltec Wifi LORA con un ESP32 como SoC del que pasamos a detallar sus características:



Ilustración 1- Placa de desarrollo LoRa

Microcontrolador: ESP32 (240MHz LX6 dual-Core, 520KB SRA)

WiFi and Bluetooth

Chip LoRa: SX1276 (Versión de 868 y 915MHz)

Potencia: 18dBm (Versión americana)

Display: 0.96 pulgadas / 128x64 OLED

Conversor UART: CP2102

Batería opcional de 3.7V

Consumo en reposo: 800uA

Para la conexión a Internet y acceso remoto a los datos, confiaremos en un Gateway ya montado de fábrica con un precio contenido, evitando soluciones más manuales que a pesar de reducir costes, pueden darnos más quebraderos de cabeza y problemas de compatibilidad entre componentes llegando al punto de hacernos perder tiempo y dinero en el despliegue.

Para ello usaremos el Gateway Dragona LG01, el cual a pesar de ser un Gateway Single Channel, es decir, solo nos permitirá la conexión simultánea de un único nodo LORA nos va a permitir un despliegue sencillo de nuestro primer Gateway LORA como veremos después, aunque antes haremos un repaso por sus características con el fin de conocer lo más a fondo posible nuestro hardware y saber hasta dónde podremos llegar con el:



Ilustración 2 - Dragona Gateway

SoC - MIPS 24k - 400MHz (Parte de gestión de red)

LoRa: Chipset SX1276

Lora SoC: ATmega328P (32Kb EPROM, 2Kb RAM)

Interfaces:

10/100Mb RJ45

WiFi 802.11 b/g/n

Además, podemos ver la arquitectura interna del dispositivo, ya que es una arquitectura muy típica que podemos utilizar a la hora de montar o desplegar nuestro propio Gateway construido a mano por nosotros.

LG01N System Overview:

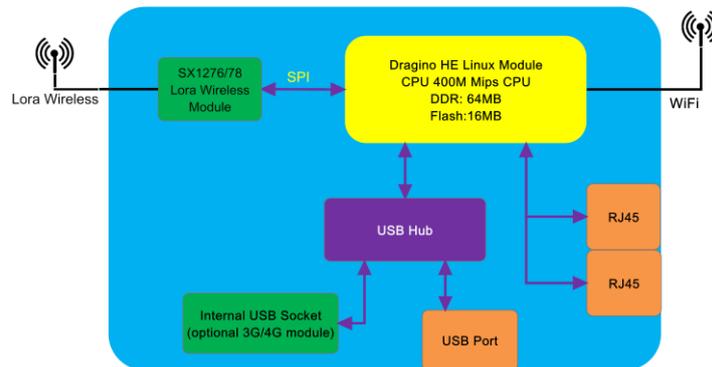


Ilustración 3 - Esquema interno de Gateway

En el podemos ver como disponemos de un SoC con conectividad a internet mediante Ethernet y Wifi y mediante un interfaz SPI conectamos al módulo SX1276 que nos aporta la conectividad LoRa. A mayores este dispositivo dispone de una conexión USB para poder disponer de conexión 3/4G mediante un modem USB.

Con este hardware ya tendremos lo necesario para poder transmitir paquetes mediante la tecnología LORA, en este caso vamos a realizar una prueba sencilla enviando el típico “Hola mundo” pero una vez tengamos la red funcionando podremos utilizarlo para cualquier otra aplicación como hemos comentado al inicio de este apartado.

El primer paso será disponer de nuestro Gateway LORA funcionando de manera que podamos conectar nuestros nodos con Internet y nuestro servidor IoT. Para ello nos vamos a ayudar de la plataforma The Things Network (<https://www.thethingsnetwork.org/>) ya que de esta manera tendremos nuestro alcance una red desplegada dentro de la que integrarnos, lo cual nos va a facilitar el despliegue de nuestros dispositivos IoT.

Crearemos una cuenta dentro de la plataforma usando el siguiente enlace <https://account.thethingsnetwork.org/register> y rellenamos los datos que se nos solicitan:

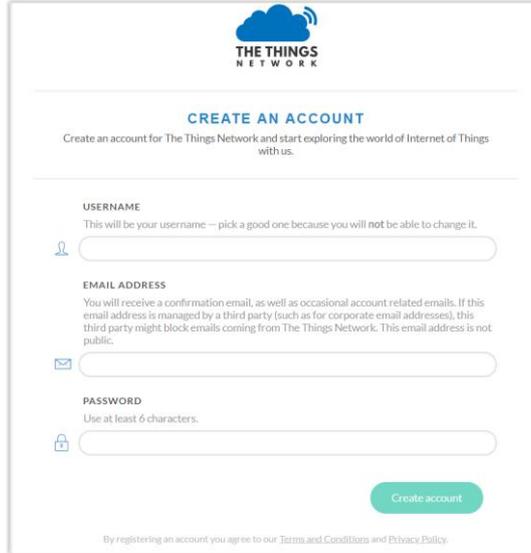


Ilustración 4 - Creación de la cuenta

Y una vez creada la cuenta podremos acceder a la consola de administración donde podremos dar de alta nuestras aplicaciones y nuestros Gateway como veremos a continuación:

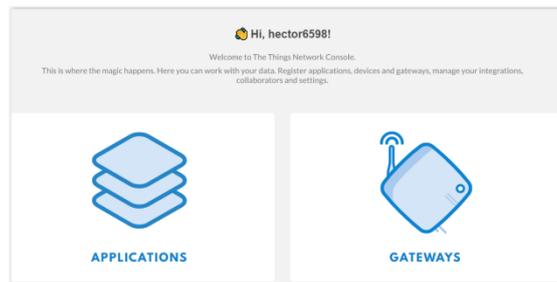


Ilustración 5 - Acceso a aplicaciones y Gateways

Pulsaremos sobre la imagen de Gateways, y encontraremos un enlace que dice “Register gateway” que nos llevará a los datos que debemos completar en nuestro Gateway:

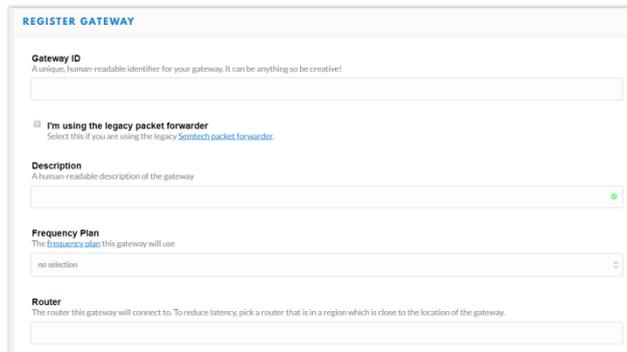


Ilustración 6 - Registro del Gateway

Debemos completar todos los datos y es muy importante marcar la opción: “I’m using the legacy packet forwarder” para que los paquetes emitidos o recibidos a través de LORA pasen a la parte IP del Gateway y lleguen hasta nuestras aplicaciones alojadas en Internet.

El Gateway ID debe ser único por eso puede facilitarnos mucho la elección de ese ID el usar la dirección MAC de nuestro Gateway ya que este identificador debe ser único para cada dispositivo.

Una vez que tenemos nuestro Gateway dado de alta dentro de la plataforma de The Things Network (Desde ahora TTN para poder recordarlo más fácil) debemos pasar a configurar nuestro dispositivo y conectarlo a internet.

Para ello simplemente colocaremos un cable de red en la toma etiquetada como WAN y el otro extremo del cable irá a alguna boca libre de nuestro router, una vez tengamos red en el dispositivo podremos acceder al panel web de administración para comenzar con la configuración.

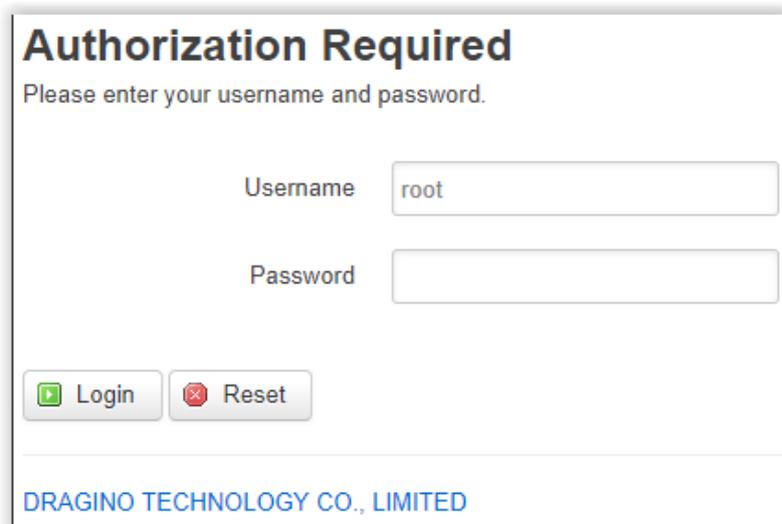


Ilustración 7 - Acceso al Gatey via web

Las credenciales por defecto son:

usuario → root

password → dragino

Y una vez dentro del panel de administración podremos comenzar con el proceso de configuración de la parte LoRa de nuestro Gateway, para lo cual debemos saber

que este Gateway dispone de un microcontrolador ATmega328P igual que el de Arduino al cual está conectado nuestro chip LoRa.

Para poder cargar el programa que se encargará de gestionar debemos dirigirnos al menú Sensor --> Flash MCU

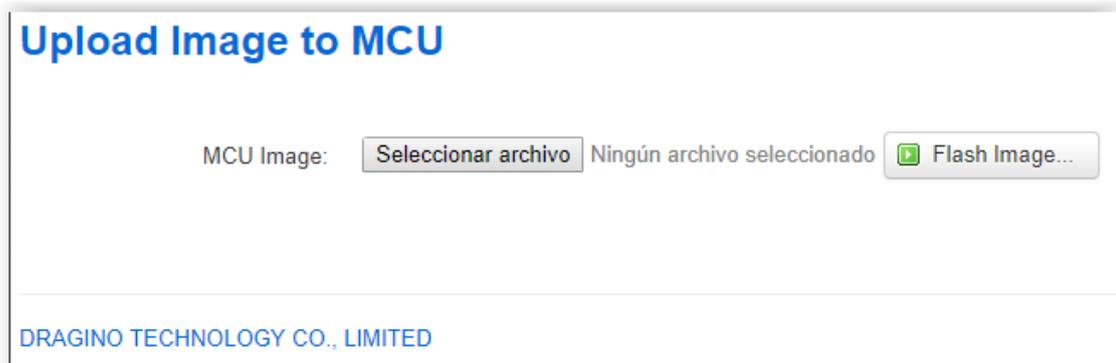


Ilustración 8 - Carga del programa LoRa

Y aquí deberemos cargar la última versión Del software que esté disponible en la página web del fabricante (<http://www.dragino.com/downloads/index.php?dir=motherboards/lg01/sketch/>) en este caso la v004.

Índice de [/ downloads / motherboards / lg01 / sketch /](#)

Archivo	Tamaño	Modificado
Directorio Padre		
changelog	235.0 B	2018-Aug-07
single_pkt_fwd_v002.hex	49.3 KB	2017-Aug-19
single_pkt_fwd_v003.ino.hex	47.5 KB	2018-May-14
Single_pkt_fwd_v004.ino.hex	49.1 KB	2018-Sep-08
4 Archivos - 0 Carpetas	Tamaño total: 146.2 KB	

Powered by [AutoIndex PHP Script](#)

Ilustración 9 - Versiones del programa

Una vez que obtengamos el mensaje de confirmación de que el firmware se ha cargado correctamente debemos reiniciar el dispositivo para que los cambios tengan efecto y se cargue durante el arranque del sistema.

Para comprobar que todo está cargado y funcionando correctamente es necesario acceder al menú: Sensor --> MicroController y revisar que la configuración coincide con la que se muestra a continuación:

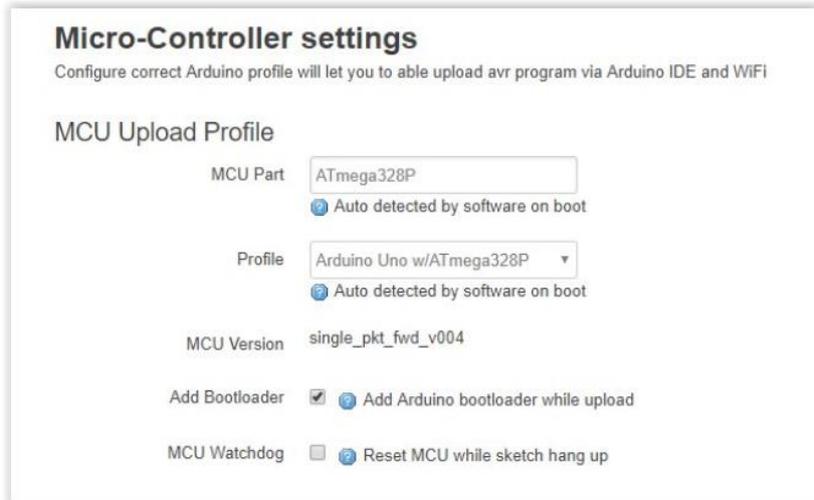


Ilustración 10- Comprobación del firmware

Con el sketch cargado es el momento de pasar a configurar el chip LoRa para que trabaje dentro de los rangos de frecuencia permitidos para Europa, y además configuraremos los datos de conexión con la plataforma TTN tal y como podemos ver a continuación:

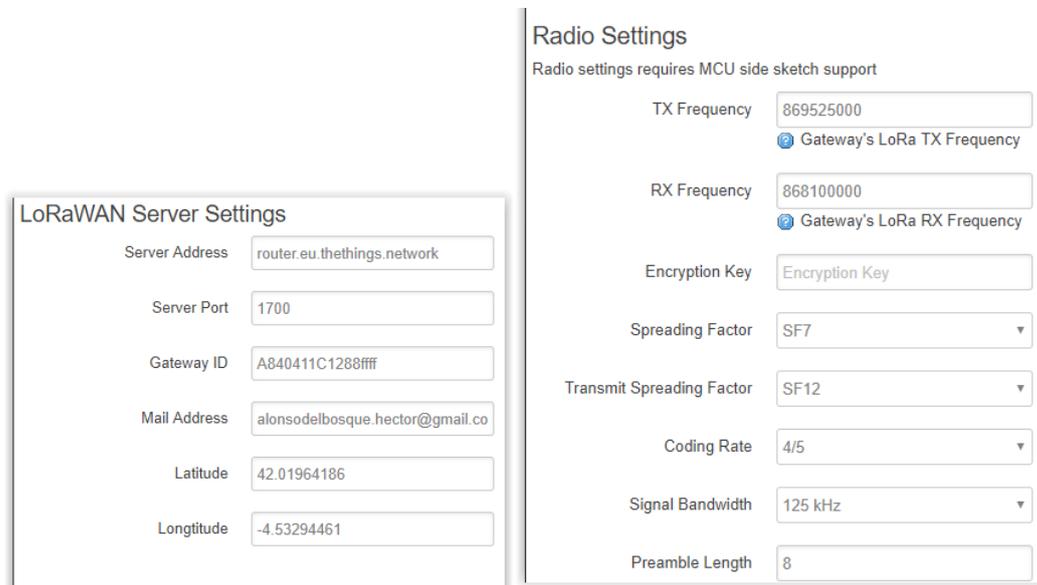


Ilustración 11- Configuración servidor TTN

Es importante configurar correctamente el Spread Factor tal y como vemos en la imagen sobre todo en el apartado que afecta a la transmisión LoRa es decir la Frecuencia TX y el “Transmit Spreading Factor” ya que estos dos valores son

usados posteriormente por la plataforma TTN para comunicarse con los módulos LoRa y pueden darnos problemas a la hora de hacer la autenticación de nuestros nodos en la red.

Por último, debemos configurar el tipo de servidor IoT que vamos a usar en nuestro Gateway ya que soporta diferentes protocolos, en nuestro caso seleccionaremos LoRaWAN como servidor IoT.

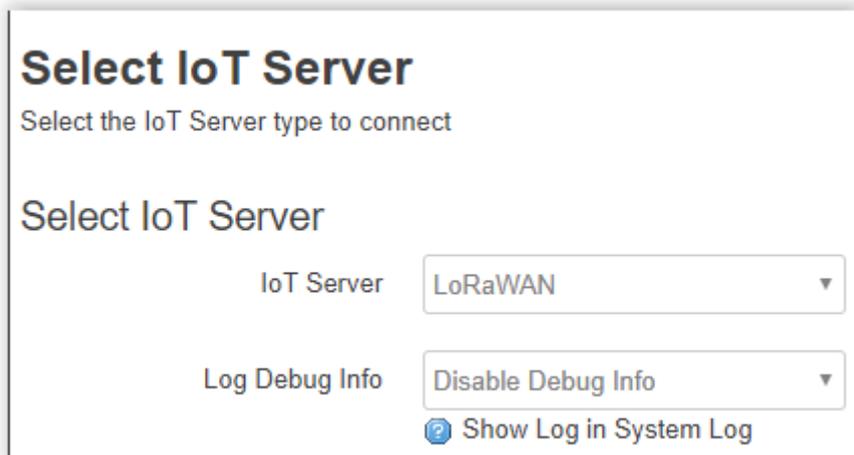


Ilustración 12 - Servidor IoT de destino

Una vez finalizado el proceso podremos volver a la plataforma TTN para comprobar que realmente nuestro Gateway está levantado y funcionando tal y como podemos ver en la siguiente imagen donde nos indica el estado junto a un indicador verde.

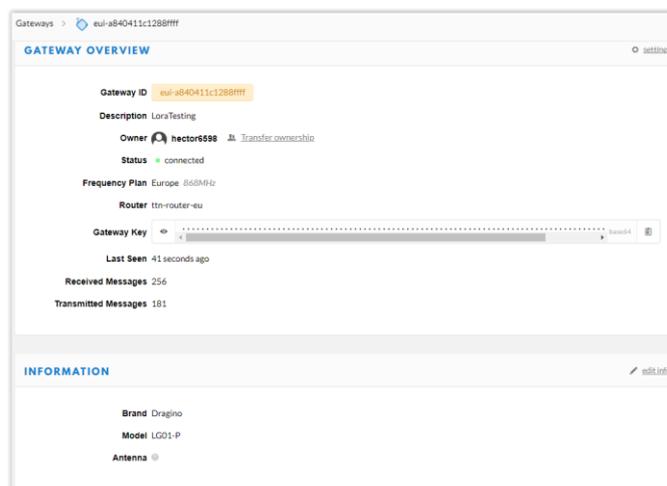


Ilustración 13 - Comprobación del Gateway

Una vez que hemos comprobado que nuestro gateway funciona vamos a pasar a configurar un nodo que se una a la red para poder transmitir mensajes. En este caso esta labor la hemos delegado en un módulo de desarrollo de la marca Heltec que nos facilita mucho la creación de un nodo al disponer ya de todo el hardware montado.

La finalidad de este primer nodo será simplemente el envío de una cadena de caracteres a través de LoRa y que se muestre dentro de una aplicación que crearemos en la plataforma TTN, es decir hacer nuestro primer “Hola mundo” dentro de un entorno LoRaWAN.

Puede parecer algo muy sencillo, pero veremos que es la mejor manera de ir repasando cada concepto visto anteriormente, vamos a comenzar en primer lugar creando una aplicación dentro de la plataforma TTN que nos permitirá recibir los datos enviados por nuestro nodo y visualizarlos, para posteriormente poder recogerlos o hacer con ello lo que consideremos necesario. Para ello dentro de la consola de TTN pulsaremos sobre el botón “Applications”.

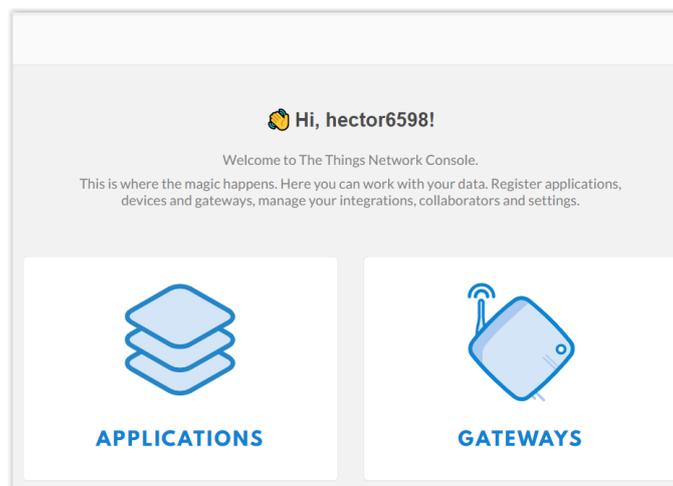


Ilustración 14 - Acceso a aplicaciones o Gateways

Tendremos un enlace llamado “add application” que pulsaremos para poder acceder al formulario donde introduciremos los datos de nuestra aplicación.

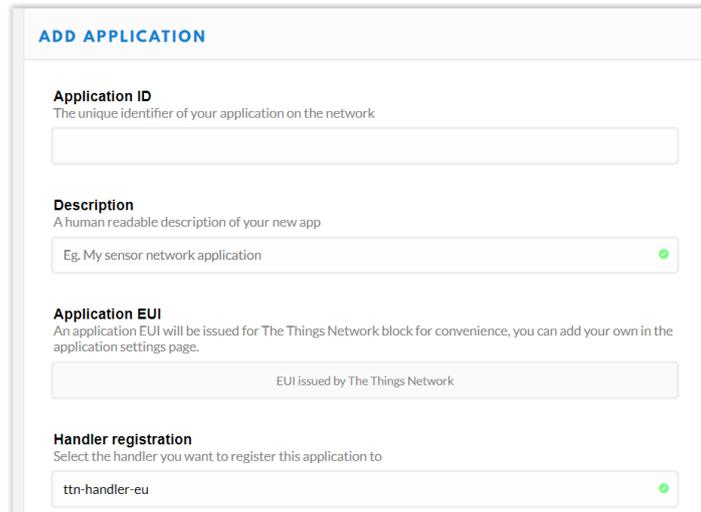


Ilustración 15 - Creación de una aplicación

Como vemos, se nos solicita únicamente un ID que sea único para nuestra aplicación ya que el resto de campos vienen configurados por defecto, aunque podremos cambiarlos después si lo necesitamos. Sobre todo, es importante tener configurado el Handler con la dirección “ttn-handler-eu” ya que tal y como vimos antes es el servidor que configuramos en nuestro Gateway.

El identificador de aplicación EUI se genera automáticamente, pero puede ser cambiado posteriormente, esto tiene como finalidad el poder añadir dispositivos a esta red que puedan venir con ciertos datos pregrabados de fábrica que necesitemos usar en nuestra red.

Veremos que nuestra aplicación se ha creado y está a la espera de que la alimentemos de datos añadiendo nuevos nodos a ella que será nuestro siguiente paso.

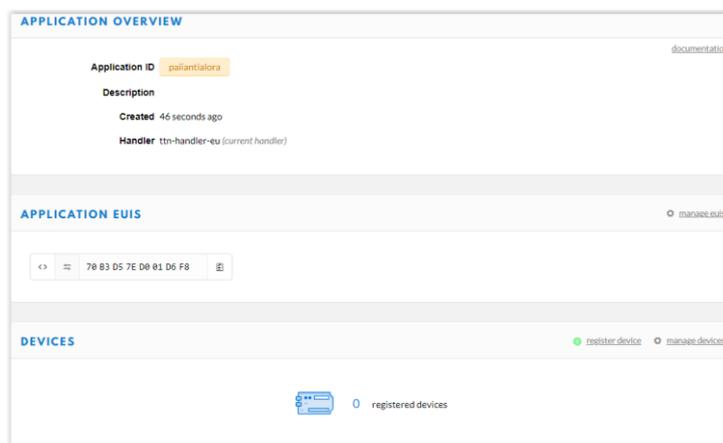


Ilustración 16 - Estado de la aplicación

Para ello en el apartado de “Devices” pulsaremos sobre el enlace “register device” para acceder a un nuevo formulario que nos ayudará a añadir nuestro nodo.

Ilustración 17 - Registro de un dispositivo

Únicamente tenemos que asignar un “Device ID” que deberemos mantener siempre que queramos seguir usando este dispositivo en la aplicación ya que es algo que no podremos cambiar.

Vemos un campo marcado en rojo a propósito y es que salvo que tengamos un “Device EUI” concreto que nos haya venido con nuestro nodo, debemos pulsar sobre el icono de las flechas para que pase a ser un icono de un lápiz y sea la plataforma la que genere ese EUI.

Aunque esté creada nuestra aplicación antes de poder utilizada es necesario configurar otra serie de valores que hemos visto anteriormente como es el tipo de autenticación que determinaran la manera en la que deberemos configurar nuestros nodos.

En este caso al tratarse de un entorno de pruebas sin datos críticos y donde únicamente tendremos un nodo que gestionar, usaremos el sistema ABP de autenticación, pero en caso de disponer de una red más extensa de nodos o datos más sensibles configuraríamos un sistema de autenticación OTAA.

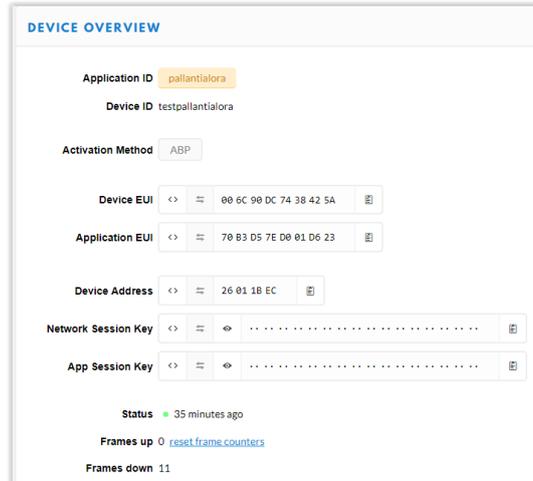


Ilustración 18 - Estado del dispositivo

Una vez finalizado este proceso tendremos todo lo necesario en la plataforma TTN para poder pasar a configurar nuestro nodo.

Para cargar los diferentes programas o sketch dentro de nuestro ESP32 necesitaremos realizar algunas configuraciones en el IDE de Arduino que será el programa que usaremos para realizar la carga de los programas en la memoria de nuestro nodo.

Para ello debemos ir a **Archivo** → **preferencias** y en la parte de Gestor de URLs Adicionales de Tarjetas añadir el siguiente enlace:

https://dl.espressif.com/dl/package_esp32_index.json

Posteriormente iremos al menú **Herramientas** → **Placa** → **Gestor de placas**

Y en el cuadro que nos aparece simplemente buscaremos por “esp32” e instalaremos la librería que nos aparece:

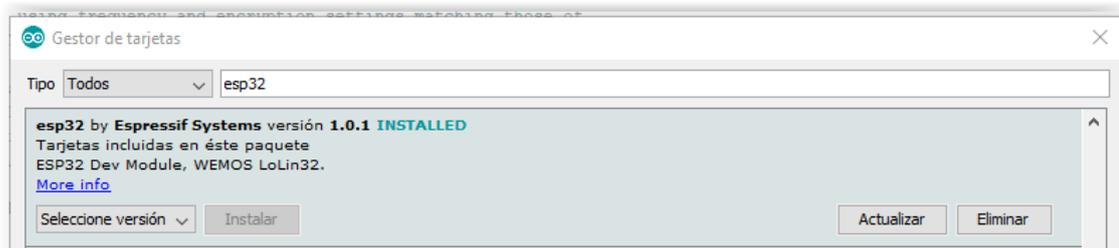


Ilustración 19 - Librería ESP32

Una vez instalada podremos ir a **Herramientas** → **Placa** y seleccionar la placa de desarrollo que tenemos:

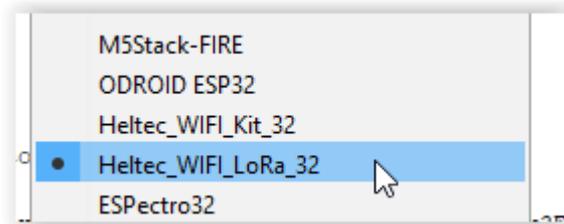


Ilustración 20 - Placa LoRa

Y posteriormente configurarla para poder utilizarla y cargar los programas de manera correcta, para lo cual usaremos la siguiente configuración:

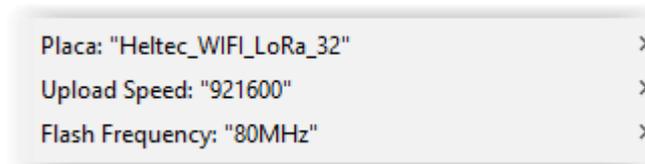


Ilustración 21 - Detalle de los datos de placa

Con esto ya podremos comenzar a utilizar nuestra placa de la misma manera que se hace con las placas de Arduino, facilitando así el desarrollo y carga de programas. Para poder controlar y el chip LoRa de nuestra placa de desarrollo, vamos a hacer uso de una librería que nos facilitará el acceso a todas las funciones de nuestra placa.

Para ello debemos descargarnos la librería desde el enlace de su desarrollador

<https://github.com/matthijskooijman/arduino-lmic> una vez descargado el archivo *.zip que contiene la librería LMIC (LoRaMAC-in-C) procedemos a cargarla en el IDE de Arduino pulsando sobre Programa → Incluir librería → Añadir biblioteca .ZIP

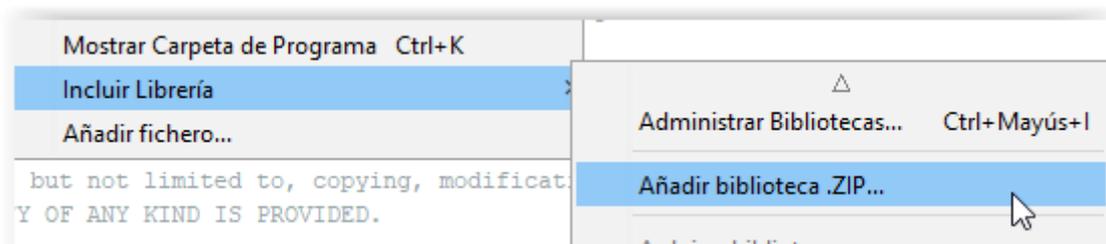


Ilustración 22 - Añadir librería desde ZIP

Y seleccionamos el archivo *.ZIP que acabamos de descargarnos, si todo ha sido correcto entonces ahora tendremos disponibles diferentes ejemplos que nos van a ayudar en esta primera conexión con el entorno LoRa, para ello iremos a **Archivo** → **Ejemplos** → **Arduino LMIC** → **ttn-abp**

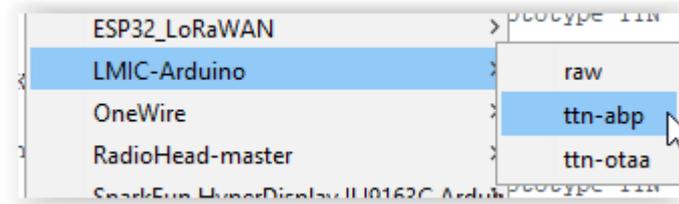


Ilustración 23 - Carga de un ejemplo

Nos cargará el programa de ejemplo para el sistema de autenticación ABP, pero es necesario saber que este programa está orientado al uso de Arduino como placa de control del chip LoRa por lo que necesitamos hacer algunos cambios en el mismo para que funcione ya que el pinout del interfaz SPI es diferente en la MCU de ESP32 respecto a los Atmega de Arduino.

Lo primero será buscar el comentario “// Pin mapping” y sustituir el contenido de la constante `lmic_pinmap` por el siguiente:

```
// Pin mapping

const lmic_pinmap lmic_pins = {

    .nss = 18,

    .rxtx = LMIC_UNUSED_PIN,

    .rst = 14,

    .dio = {26, 33, 32},

};
```

Además de ese cambio debemos inicializar el interfaz SPI en los siguientes pines colocando la instrucción mostrada a continuación dentro de la función “void setup()”

```
SPI.begin(5, 19, 27);
```

Con estas modificaciones tendríamos nuestro programa adaptado a la placa de desarrollo Heltec, pero aún debemos cambiar algunas líneas más para incluir los datos de conexión a nuestra aplicación para que al comunicar con el Gateway, este sepa donde tiene que enviarlos.

Estas líneas son las siguientes:

```
static const PROGMEM u1_t NWKSKEY[16] = { 0xF4, 0x1F,  
0xA5, 0x0D, 0x41, 0x88, 0x98, 0xEB, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00 };
```

```
static const u1_t PROGMEM APPSKEY[16] = { 0x33, 0x89,  
0xCA, 0x2A, 0x5D, 0x21, 0xCF, 0x95, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00 };
```

```
static const u4_t DEVADDR = 0x26011BEC ;
```

Donde sustuiremos los valores entre llaves por los que nos indica la plataforma TTN del mismo modo que el identificador Device Address:

The screenshot shows the configuration interface for an ABP (Activation Method) in the TTN console. The 'Activation Method' is set to 'ABP'. Below it, there are five configuration fields, each with a copy icon and a visibility toggle (eye icon):

- Device EUI:** 00 6C 90 DC 74 38 42 5A
- Application EUI:** 70 B3 D5 7E D0 01 D6 23
- Device Address:** 26 01 1B EC
- Network Session Key:** (represented by dots)
- App Session Key:** (represented by dots)

Ilustración 24 - Claves de la app

Con estos cambios realizados ya estamos listos para cargar el programa en nuestra placa de desarrollo, si no ha habido ningún error tendremos nuestro programa cargado que enviará el mensaje “Hello, Word!” al iniciar.

Una vez reiniciado nuestro nodo podremos ver en la plataforma TTN el tráfico generado por el nodo y los datos emitidos y recibidos en el apartado “payload”, toda esta información es transmitida en formato hexadecimal y mostrada directamente por nuestra aplicación ya que no le hemos indicado que haga nada con dicha información.

Filters				
time	counter	port		
▼ 04:49:41		1		payload: 12 34 67 89 34 34
▲ 04:49:39	0	1	retry	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
▼ 04:48:26		1		payload: 12 34 67 89 34 34
▲ 04:48:24	0	1	retry	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
▼ 04:29:11		1		payload: 12 34 67 89 34 34
▲ 04:29:10	0	1	retry	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21

Ilustración 25 - Paquete recibidos

También podremos ver de manera general el tráfico generado en nuestro Gateway ya que podríamos disponer de varias aplicaciones o si nuestro nodo es público nos puede interesar conocer el tráfico que se está generando para saber el estado y saturación de nuestro Gateway.

GATEWAY TRAFFIC <small>beta</small>						
time	frequency	mod.	CR	data rate	airtime (ms)	cnt
▼ 04:49:40	869.525	loro	4/5	SF 9 BW 125	185.3	4 dev addr: 26 01 1B EC payload size: 19 by
▲ 04:49:39	868100000	loro	4/5	SF 7 BW 125	61.7	0 dev addr: 26 01 1B EC payload size: 26 by
▼ 04:48:25	869.525	loro	4/5	SF 9 BW 125	185.3	3 dev addr: 26 01 1B EC payload size: 19 by
▲ 04:48:24	868100000	loro	4/5	SF 7 BW 125	61.7	0 dev addr: 26 01 1B EC payload size: 26 by
▼ 04:29:11	869.525	loro	4/5	SF 9 BW 125	185.3	2 dev addr: 26 01 1B EC payload size: 19 by
▲ 04:29:10	868100000	loro	4/5	SF 7 BW 125	61.7	0 dev addr: 26 01 1B EC payload size: 26 by
▼ 04:13:41	869.525	loro	4/5	SF 9 BW 125	185.3	1 dev addr: 26 01 1B EC payload size: 19 by
▲ 04:13:40	868100000	loro	4/5	SF 7 BW 125	61.7	0 dev addr: 26 01 1B EC payload size: 26 by

Ilustración 26 -Tráfico del Gateway

Por último, como leer los mensajes que recibamos en hexadecimal puede ser más incómodo podemos utilizar la herramienta de decodificación que nos ofrece TTN, para ello deberemos programar un pequeño código usando JavaScript que nos convierta la cadena hexadecimal en una cadena de texto ASCII.

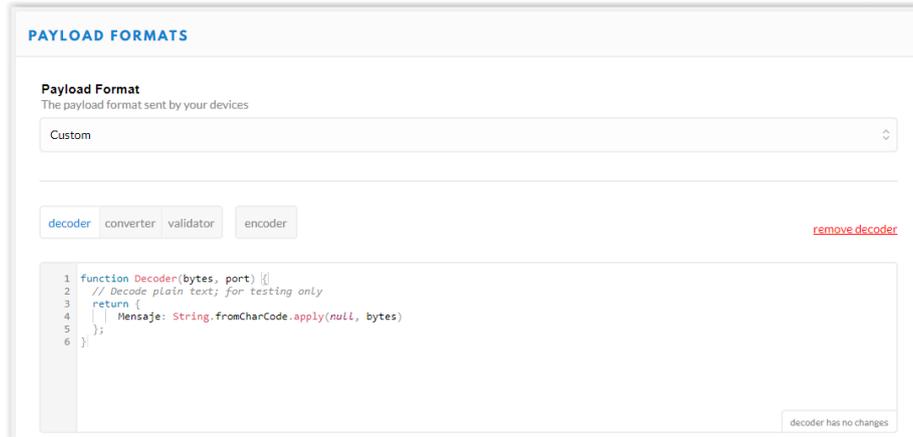


Ilustración 27 - Configuración de un decoder

Ahora veremos que nos aparece un nuevo campo en los datos de la aplicación donde nos aparece la información decodificada, siendo esto especialmente útil por ejemplo en el caso de estar recibiendo información desde varios sensores conectados a un único nodo para organizar como mostrar los datos recibidos.

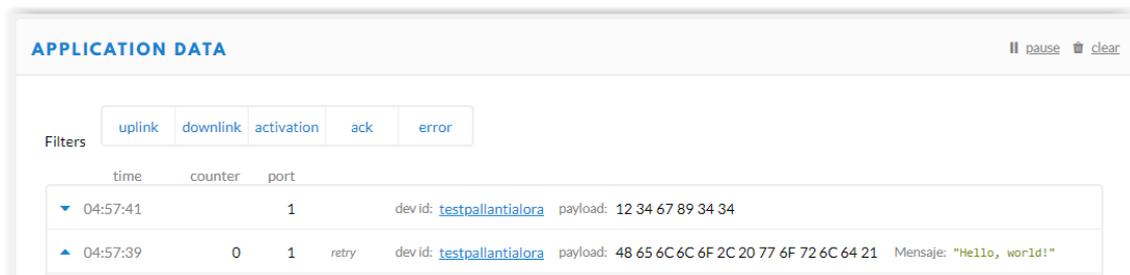


Ilustración 28 - Datos mostrados en claro

EJEMPLOS DE DESPLIEGUE

Conociendo ya un poco más sobre las características de LoRa y las redes LPWAN podremos tener más claro al tipo de proyectos y entornos al que se pueden aplicar de una manera más habitual, dejando una relación de escenarios donde ya se aplica con éxito:

- Monitorización de explotaciones agrarias y ganaderas
- Geolocalización de flotas y personas
- Recogida de datos en entornos industriales
- Monitorización climatológica y de contaminación
- Lectura de contadores de manera remota
- Control de stock de almacenes
- Control de sistemas de parking
- Sistemas de sharing de vehículos