







ENTREGABLE 02:

GUÍA DE CONFIGURACIÓN DISPOSITIVOS LORAWAN VÍA THE THINGS NETWORKS.

(Mayo 2023)







1 LORAWAN.	3
2 The Things Networks.	4
2.1 Puertas de enlace.	4
2.2 Servidor de red.	4
2.3 Aplicaciones.	5
2.4 The Things Network Console.	5
3 Naturaleza TTN	5
4 Despliegue HIBA	6
5 Decoders asociados.	11
5.1 UC51x Series (atd-uc511-d1, atd-512-d1)	12
5.2 EM500-PP (atd-em500-pp)	13
5.3 EM500-SMT (atd-em500-smtc)	15
5.4 EM300TH (atd-em300th).	17
5.5 ZTEMP (atd-ztemp)	19
5.6 Node MINI Cesens (atd-cesens)	20









1.- LORAWAN.

LoRaWAN es un protocolo de red de área amplia de bajo consumo diseñado específicamente para la comunicación de largo alcance entre dispositivos en el Internet de las cosas (IoT). Permite la comunicación segura y bidireccional entre dispositivos IoT y puertas de enlace de red a largas distancias, normalmente varios kilómetros, con un consumo mínimo de energía.

LoRaWAN opera en las bandas industriales, científicas y médicas (ISM) sin licencia, como la banda de 868 MHz en Europa y la banda de 915 MHz en América del Norte. Utiliza la técnica de modulación de espectro ensanchado chirp, que permite la comunicación de largo alcance con bajo consumo de energía.

Las redes LoRaWAN suelen constar de tres componentes principales:

- Dispositivos o nodos: Estos son los dispositivos loT equipados con capacidades de comunicación LoRaWAN. Pueden ser sensores, actuadores o cualquier otro tipo de dispositivo conectado que necesite transmitir o recibir datos.
- Puertas de enlace: las puertas de enlace actúan como intermediarios entre los dispositivos y la red LoRaWAN. Reciben mensajes de dispositivos cercanos y los reenvían a un servidor de red central. Las puertas de enlace pueden cubrir grandes áreas geográficas y proporcionar conectividad para múltiples dispositivos.
- Servidor de red: el servidor de red gestiona la comunicación entre las puertas de enlace y los dispositivos. Maneja tareas como la autenticación de dispositivos, el cifrado/descifrado de mensajes y el enrutamiento de datos a la aplicación o servicio apropiado.

LoRaWAN admite diferentes clases de dispositivos, cada uno con diferentes capacidades de comunicación y consumo de energía.

- Los dispositivos de clase A son los más comunes y tienen el menor consumo de energía. Tienen un patrón de comunicación bidireccional, donde los dispositivos pueden transmitir datos al servidor de la red en cualquier momento y recibir datos del servidor durante las ventanas de recepción programadas.
- Los dispositivos de clase B tienen ranuras de recepción adicionales programadas por el servidor de red para proporcionar una comunicación más flexible.









 Los dispositivos de clase C, por otro lado, tienen ventanas de recepción casi continuas, lo que permite la comunicación en tiempo real pero consume más energía.

LoRaWAN ha ganado popularidad en varias aplicaciones de loT, incluidas ciudades inteligentes, agricultura, seguimiento de activos, monitoreo ambiental y automatización industrial. Sus capacidades de largo alcance, bajo consumo de energía e idoneidad para implementaciones de bajo costo lo convierten en una opción atractiva para conectar una gran cantidad de dispositivos en un área amplia.

2.- The Things Networks.

The Things Network (TTN) es una red LoRaWAN global impulsada por la comunidad para aplicaciones IoT. Proporciona una infraestructura para que los dispositivos se conecten y se comuniquen a largas distancias mediante el protocolo LoRaWAN. TTN es una red abierta y descentralizada, lo que significa que cualquiera puede contribuir configurando puertas de enlace y participando en la red.

Los componentes clave de The Things Network incluyen:

2.1.- Puertas de enlace.

Estos son dispositivos que reciben y transmiten datos entre dispositivos LoRaWAN y el servidor de red. Las puertas de enlace generalmente están conectadas a Internet y brindan cobertura para un área geográfica específica.

2.2.- Servidor de red.

El servidor de red gestiona la comunicación entre dispositivos y aplicaciones. Maneja la autenticación de dispositivos, el enrutamiento de datos, el cifrado/descifrado y la integración con varios servicios o aplicaciones.







2.3.- Aplicaciones.

TTN permite a los usuarios crear aplicaciones y servicios sobre la red. Las aplicaciones pueden recibir datos de los dispositivos, procesarlos y desencadenar acciones en función de la información recibida.

2.4.- The Things Network Console.

La consola es una interfaz basada en la web que proporciona una forma sencilla de administrar puertas de enlace, dispositivos y aplicaciones. Permite a los usuarios monitorear la red, registrar dispositivos y configurar varias configuraciones de red.

3.- Naturaleza TTN

Una de las principales ventajas de The Things Network es su naturaleza impulsada por la comunidad. Las personas y las organizaciones pueden contribuir configurando sus propios portales y ampliando la cobertura de la red. Este enfoque colaborativo ayuda a crear una gran infraestructura LoRaWAN global que permite implementaciones de loT en varios dominios.

Things Network admite aplicaciones comerciales y no comerciales. Proporciona una plataforma para crear prototipos, probar e implementar soluciones de IoT, fomentando la innovación y la colaboración entre desarrolladores y usuarios. Además, TTN ofrece integración con varias plataformas y servicios en la nube, lo que facilita la conexión y el procesamiento de datos desde dispositivos IoT.

En general, The Things Network desempeña un papel importante en la promoción de la adopción de la tecnología LoRaWAN y permite el desarrollo de soluciones de IoT escalables y accesibles.







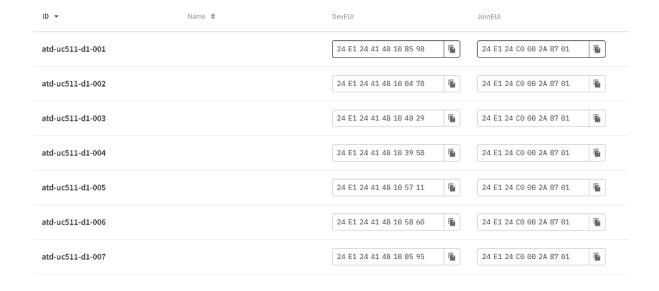
4.- Despliegue HIBA

El despliegue de los sensores del proyecto HIBA en la Universidad de Córdoba ha sido realizado según recomendación y asesoramiento del Aula de Transformación Digital FIWARE de la Universidad de Córdoba (https://www.uco.es/atdfiware) a partir de ahora ATDFIWARE.

La sensórica con protocolo LoraWAN está contemplada dentro de la plataforma de datos desplegada para el proyecto HIBA como **protocolo de comunicación preferente** estando desplegados bajo las credenciales de ATDFIWARE,

Los diferentes tipos de sensores se clasifican por categorías estableciendo para el proyecto HIBA inicialmente las siguientes aplicaciones TTN.:

atd-uc512-d1: nodo monitor/actuador con batería.











atd-uc511-d1: nodo monitor/actuador con batería y panel solar.

ID \$	Name \$	DevEUI	JoinEUI
atd-uc511-d1-007		24 E1 24 41 4B 10 05 95	24 E1 24 C0 00 2A 87 01
atd-uc511-d1-006		24 E1 24 41 4B 10 58 60	24 E1 24 C0 00 2A 87 01
atd-uc511-d1-005		24 E1 24 41 4B 10 57 11	24 E1 24 C0 00 2A 87 01
atd-uc511-d1-004		24 E1 24 41 48 10 39 58	24 E1 24 C0 00 2A 87 01
atd-uc511-d1-003		24 E1 24 41 48 10 40 29	24 E1 24 C0 00 2A 87 01
atd-uc511-d1-002		24 E1 24 41 48 10 04 78	24 E1 24 C0 00 2A 87 01
atd-uc511-d1-001		24 E1 24 41 4B 10 85 98	24 E1 24 C0 00 2A 87 01

• atd-em500-pp: nodo monitor de presión.

ID \$	Name \$	DevEUI	JoinEUI
atd-em500-pp-010		24 F1 24 12 6B 42 20 99	24 E1 24 C0 00 2A 00 02
atd-em500-pp-009		24 E1 24 12 6B 42 22 08	24 E1 24 C0 00 2A 00 02
atd-em500-pp-008		24 E1 24 12 6B 42 20 18	24 E1 24 C0 00 2A 00 02
atd-em500-pp-007		24 E1 24 12 6B 42 76 90	24 E1 24 C0 00 2A 00 02
atd-em500-pp-006		24 E1 24 12 68 42 22 04	24 E1 24 C0 00 2A 00 02
atd-em500-pp-005		24 E1 24 12 6B 42 78 28	24 E1 24 C0 00 2A 00 02
atd-em500-pp-004		24 E1 24 12 6B 42 73 12	24 E1 24 C0 00 2A 00 02
atd-em500-pp-003		24 E1 24 12 68 42 73 03	24 E1 24 C0 00 2A 00 02
atd-em500-pp-001		24 E1 24 12 68 42 76 34	24 E1 24 C0 00 2A 00 02
atd-em500-pp-002		24 E1 24 12 68 42 76 88	24 E1 24 C0 00 2A 00 02









• atd-em500-smtc: nodo monitor de temperatura, humedad, conductividad en tierra.

ID 🕶	Name \$	DevEUI	IU3nioL
atd-em500-smtc-001		24 E1 24 12 6B 42 14 49	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-002		24 E1 24 12 6B 42 13 55	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-003		24 E1 24 12 6B 42 68 95	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-004		24 E1 24 12 6B 42 75 97	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-005		24 E1 24 12 6B 42 75 15	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-006		24 E1 24 12 6B 42 68 36	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-007		24 E1 24 12 6B 42 75 72	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-008		24 E1 24 12 6B 42 15 28	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-009		24 E1 24 12 6B 42 75 29	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-010		24 E1 24 12 68 42 15 11	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-011		24 E1 24 12 6B 42 16 59	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-012		24 E1 24 12 6B 42 15 88	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-013		24 E1 24 12 6B 42 16 88	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-014		24 E1 24 12 6B 42 13 94	24 E1 24 C0 00 2A 00 01
atd-em500-smtc-015		24 E1 24 12 6B 42 91 56	24 E1 24 C0 00 2A 00 01

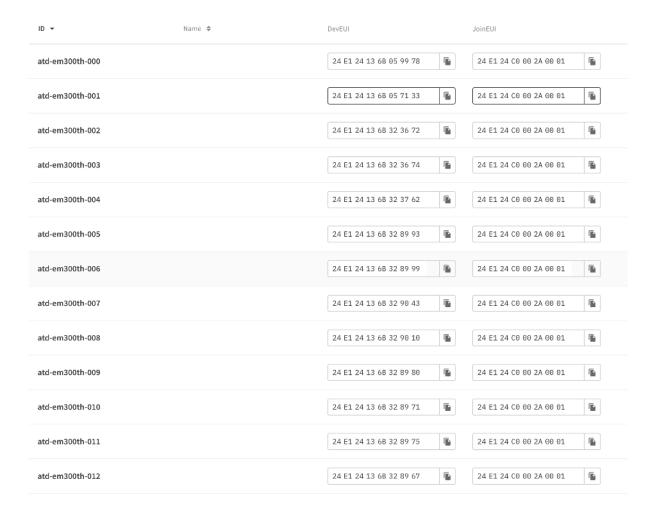








• atd-em300-th: nodo monitor de temperatura y humedad.

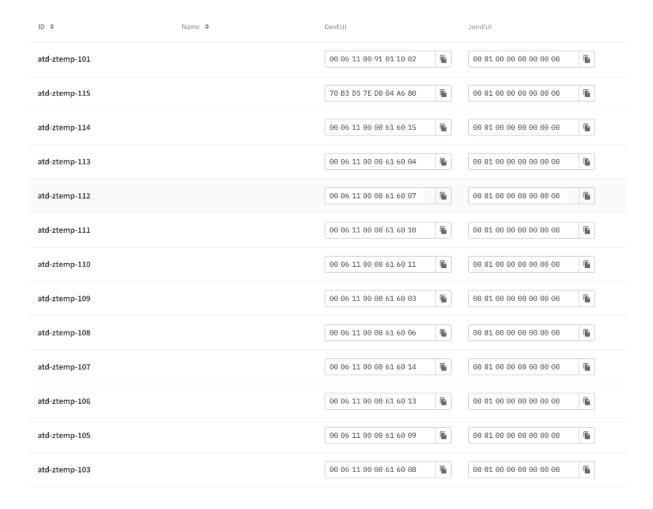








atd-ztemp; nodo monitor de temperatura.



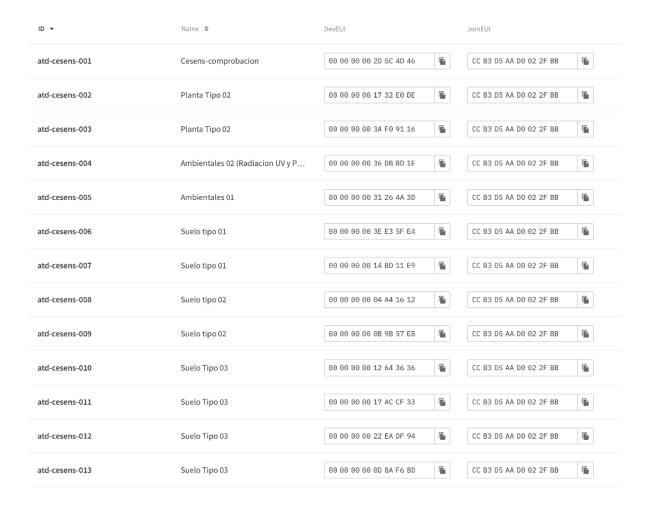








 atd-cesens: nodos multifunción caracterizados a su vez entorno de suelo, planta y ambientales



5.- Decoders asociados.

Para The Things Network (TTN), los decodificadores se utilizan para analizar e interpretar los datos sin procesar recibidos de los dispositivos LoRaWAN. Cuando un dispositivo envía una carga útil (datos) a TTN, llega en su forma binaria sin procesar. Los decodificadores ayudan a convertir esta carga útil binaria en un formato legible por humanos, lo que facilita la comprensión y el procesamiento de los datos.









Los decodificadores en TTN generalmente se implementan usando JavaScript y se ejecutan en el servidor de red TTN. El script del decodificador toma la carga útil sin procesar como entrada y devuelve un objeto estructurado que contiene los valores decodificados. El objeto decodificado puede luego procesarse, almacenarse o enviarse a otros sistemas para su análisis, en nuestro caso, la plataforma de datos basada en el ecosistema FIWARE.

Vamos a describir los decoders principales asociados a las aplicaciones:

5.1.- UC51x Series (atd-uc511-d1, atd-512-d1)

```
Copyright 2021 Milesight IoT
 * @product UC51x Series
function Decoder(bytes, port) {
   var decoded = {};
   for (var i = 0; i < bytes.length; ) {</pre>
        var channel_id = bytes[i++];
        var channel_type = bytes[i++];
        if (channel_id === 0x01 && channel_type === 0x75) {
            decoded.battery = bytes[i];
            i += 1;
        }
        else if (channel_id === 0x03 && channel_type == 0x01) {
            decoded.valve1 = bytes[i] === 0 ? "0" : "1";
            i += 1;
        }
        // VALVE 2
        else if (channel id === 0 \times 05 && channel type == 0 \times 01) {
            decoded.valve2 = bytes[i] === 0 ? "0" : "1";
            i += 1;
```









```
// VALVE 1 Pulse
        else if (channel_id === 0x04 && channel_type === 0xc8) {
            decoded.valve1_pulse = readUInt32LE(bytes.slice(i, i + 4));
            i += 4;
        // VALVE 2 Pulse
        else if (channel id === 0x06 && channel type === 0xc8) {
            decoded.valve2_pulse = readUInt32LE(bytes.slice(i, i + 4));
            i += 4;
        } else {
            break;
        }
   return decoded;
}
 * bytes to number
function readUInt32LE(bytes) {
   var value =
        (bytes[3] << 24) + (bytes[2] << 16) + (bytes[1] << 8) +
bytes[0];
   return value & 0xffffffff;
function readInt32LE(bytes) {
   var ref = readUInt32LE(bytes);
   return ref > 0x7ffffffff ? ref - 0x100000000 : ref;
```

5.2.- EM500-PP (atd-em500-pp)

```
/**
 * Payload Decoder for The Things Network
 *
```









```
Copyright 2021 Milesight IoT
function Decoder(bytes, port) {
    var decoded = {};
    for (var i = 0; i < bytes.length;) {</pre>
        var channel_id = bytes[i++];
        var channel_type = bytes[i++];
        if (channel_id === 0x01 && channel_type === 0x75) {
            decoded.battery = bytes[i];
            i += 1;
        // PRESSURE
        else if (channel_id === 0x03 && channel_type === 0x7B) {
            decoded.pressure = readInt16LE(bytes.slice(i, i + 2));
            i += 2;
        } else {
            break;
    }
    return decoded;
}
 * bytes to number
function readUInt16LE(bytes) {
    var value = (bytes[1] << 8) + bytes[0];</pre>
    return value & 0xffff;
}
function readInt16LE(bytes) {
    var ref = readUInt16LE(bytes);
    return ref > 0x7fff ? ref - 0x10000 : ref;
}
```









5.3.- EM500-SMT (atd-em500-smtc)

```
Copyright 2021 Milesight IoT
 * @product EM500-SMT
function decodeUplink(input) {
   var bytes = input.bytes;
   var port = input.f_port;
   var decoded = {};
   for (var i = 0; i < bytes.length;) {</pre>
        var channel_id = bytes[i++];
        var channel_type = bytes[i++];
        if (channel_id === 0x01 && channel_type === 0x75) {
            decoded.battery = bytes[i];
            i += 1;
        }
        // TEMPERATURE
        else if (channel_id === 0x03 && channel_type === 0x67) {
            decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) /
10;
            i += 2;
        else if (channel_id === 0x04 && channel_type === 0x68) {
            decoded.humidity = bytes[i] / 2;
```









```
i += 1;
        else if (channel_id === 0x04 && channel_type === 0xCA) {
            decoded.humidity = readUInt16LE(bytes.slice(i, i + 2)) /
100;
            i += 2;
        }
        else if (channel_id === 0x05 && channel_type === 0x7F) {
            decoded.ec = readUInt16LE(bytes.slice(i, i + 2));
            i += 2;
        } else {
            break;
        }
    }
    //return decoded;
    return {
      data: decoded,
     warnings: [],
     errors: []
    };
}
 * bytes to number
function readUInt16LE(bytes) {
    var value = (bytes[1] << 8) + bytes[0];</pre>
    return value & 0xffff;
}
function readInt16LE(bytes) {
   var ref = readUInt16LE(bytes);
    return ref > 0x7fff ? ref - 0x10000 : ref;
```









5.4.- EM300TH (atd-em300th).

```
function Decoder(bytes, port) {
   var decoded = {};
   for (var i = 0; i < bytes.length;) {</pre>
        var channel_id = bytes[i++];
       var channel_type = bytes[i++];
        if (channel_id === 0x01 && channel_type === 0x75) {
            decoded.battery = bytes[i];
            i += 1;
        }
        // TEMPERATURE
        else if (channel_id === 0x03 && channel_type === 0x67) {
            decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) /
10;
            i += 2;
        }
        // HUMIDITY
        else if (channel_id === 0x04 && channel_type === 0x68) {
            decoded.humidity = bytes[i] / 2;
            i += 1;
```









```
}
        else {
            break;
        }
    }
    return decoded;
}
function readUInt16LE(bytes) {
    var value = (bytes[1] << 8) + bytes[0];</pre>
    return value & 0xffff;
}
function readInt16LE(bytes) {
    var ref = readUInt16LE(bytes);
    return ref > 0x7fff ? ref - 0x10000 : ref;
}
```







5.5.- ZTEMP (atd-ztemp)

```
function decodeUplink(input) {
 // (array) of bytes to an object of fields.
 var bytes = input.bytes;
 var port = input.f_port;
 var decoded = {};
 // Conocer el signo de la temperatura.
 var signo = bytes[0] >> 4;
 // Dígitos Temperature
 var temp_b1 = bytes[0] & 0xF;
 var temp_b2 = bytes[1] >> 4;
 var temp_b3 = bytes[1] & 0xF;
 // Cálculo Temperature
 decoded.temperature = temp_b1*10+temp_b2+temp_b3/10;
 // Si signo es negativo
 if (signo==1) {
     decoded.temperature = -decoded.temperature;
 }
 // Batteries
 decoded.battery = parseInt(bytes[2].toString(16));
 return {
   data: decoded,
   warnings: [],
   errors: []
```









}

5.6.- Node MINI Cesens (atd-cesens)

Los nodos de esta categoría tienen cuatro salidas configurables bajo demanda con sondas de distinta naturaleza. Para el proyecto HIBA han sido categorizados por nodos de suelo, planta, ambientales.

El acceso a su decoder se realiza a través de un configurador de nodos en línea suministrado por la empresa proveedora. Este es: https://app.cesens.com/ttn-formatter.php