



Laboratorio 2: LoRaWAN

Este laboratorio se propone poner en práctica los principales conceptos de LoRaWAN, configurando una red utilizando The Things Network (TTN), y programando dispositivos para enviar y recibir datos.

Introducción

En este laboratorio del curso utilizaremos la placa de desarrollo Arduino Zero y el módulo RN2903 (de Microchip) integrado en el kit *LR 2 click* (MikroElectronica) conectado a través de la placa de adaptación.

Se utilizarán *gateways* LoRaWAN de bajo costo, con salida WiFi, del tipo *The Things Indoor Gateway* ([TTIG](#)).

Objetivos

Objetivos generales

Configurar una red LoRaWAN utilizando The Things Network (TTN) y programar dispositivos que envíen y reciban datos de una aplicación.

Objetivos particulares

- Realizar la configuración de una aplicación en TTN obteniendo las claves necesarias para activar dispositivos.
- Escribir un programa sencillo para registrar el dispositivo en la red y que envíe datos de manera autónoma.
- Escribir un programa en FreeRTOS creando una tarea encargada de la comunicación LoRaWAN.

Materiales

- Computadora con Arduino IDE instalado y configurado (ya utilizado en el laboratorio 1).
- Placa de desarrollo [Arduino Zero](#).
- Placa de adaptación [MikroBus para Arduino](#).
- Placa de comunicación [LR 2 Click](#)

Fundamentos

Generalidades: arquitectura LoRaWAN

Una red LoRaWAN típica consiste en:

1. *End devices* (dispositivos o IoT devices según nomenclatura general usada en el curso): envía y recibe mensajes de *Gateways* usando modulación LoRa.

2. *Gateway*: recibe mensajes de los *End devices* y los reenvía al *Network Server* y viceversa.
3. *Network Server*: es el servidor que maneja la red (compartida entre varios usuarios y aplicaciones)
4. *Application servers*: es el servidor responsable de procesar de forma segura los datos de aplicación.
5. *Join Server*: es el servidor que procesa los mensajes de solicitud de unirse a la red (join-request)

En el laboratorio utilizaremos el Arduino Zero conectado a un módulo RN2903 como *End devices*.

Los *gateways* pueden clasificarse en: *indoor (picocell)* y *outdoor (macrocell)* *gateways*. Los *gateways outdoor* proveen mayor cobertura que los *gateways indoor* y pueden ser utilizados tanto en medios rurales como urbanos. Los *gateways indoor* son soluciones más económicas para dar cobertura a pequeñas superficies, en edificios u hogares. También son una solución económica para uso en laboratorios.

La siguiente figura muestra los componentes anteriores en la implementación de [The Things Stack](#), presentada más adelante.

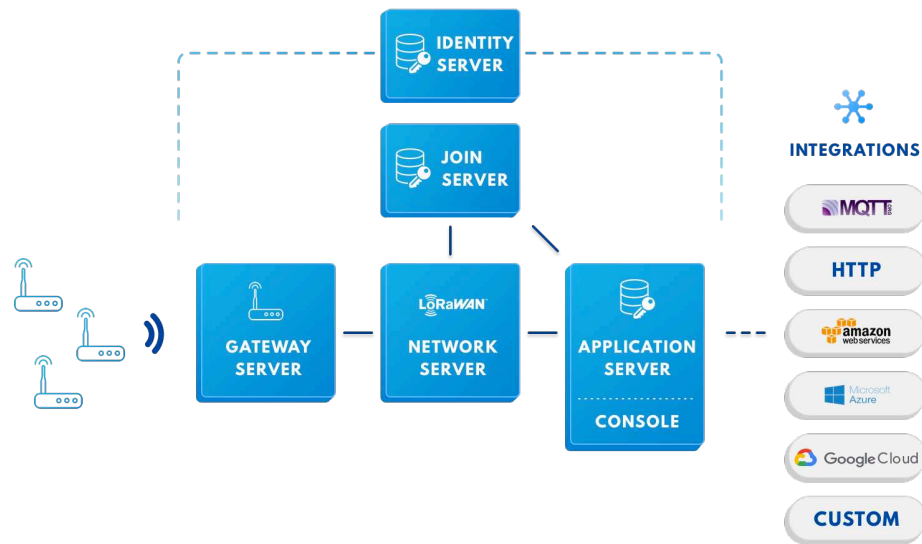


Figure 1: Arquitectura

Para que un dispositivo pueda enviar y recibir datos necesita registrarse para unirse a la red realizando el procedimiento de *join* o activación. Existen dos métodos: *Over-The-Air-Activation* (OTAA) y *Activation By Personalization* (ABP). El primero es el que utilizaremos en el curso, el más seguro y recomendado, durante el cual se asigna una dirección dinámica al dispositivo y se ne-

gocian claves de seguridad entre el *Network Server* y el dispositivo. Por más información se puede consultar el artículo [ABP vs OTAA](#).

Para la activación el dispositivo utiliza los siguientes datos (que debe tener almacenados en memoria):

- AppEUI
- DevEUI y
- AppKey.

La **AppKey** es una clave secreta AES de 128 bits conocida como clave raíz (*root key*) y es la misma que utiliza la red donde se va a registrar el dispositivo. Esta clave nunca se envía por la red. Por otra parte la **AppEUI** y **DevEUI** no son secretos y son visibles para todos.

Después de la activación, la siguiente información se almacena en el dispositivo final:

- **DevAddr**: una dirección de dispositivo de 32 bits asignada por el servidor de red para identificar el dispositivo final dentro de la red actual.
- **NwkSKey**: el dispositivo final y el servidor de red utilizan la clave de sesión de red para calcular y verificar el código de integridad del mensaje (MIC) de todos los mensajes de datos. La **NwkSKey** también se usa para cifrar y descifrar cargas útiles con comandos MAC.
- **AppSKey**: la clave de sesión de la aplicación se utiliza para cifrar y descifrar las cargas útiles de la aplicación en los mensajes de datos para garantizar la confidencialidad del mensaje.

Nota: Se recomienda la lectura en [The Things Fundamentals](#) de:

- [LoRaWAN Architecture](#)
 - [End Device Activation](#)
 - [Security](#)
-

The Things Network (TTN) (y otros)

[The Things Network](#) (TTN) es un ecosistema colaborativo global de Internet de las Cosas que crea redes, dispositivos y soluciones utilizando LoRaWAN. The Things Network utiliza [The Things Stack Community Edition](#), que es una red LoRaWAN abierta, descentralizada y de colaboración abierta. Finalmente [The Things Industries](#) es una entidad comercial que proporciona funciones avanzadas de servidores de red LoRaWAN de nivel empresarial y soporte profesional para empresas.

TTN es desarrollado por una comunidad en crecimiento en todo el mundo y se basa en contribuciones voluntarias al proyecto. Su sitio web presenta diferentes guías para permitir a los usuarios implementar *gateways* en su ciudad para

hacer crecer la red. Estos *gateways* proporcionan cobertura de largo alcance con LoRaWAN.

La [Comunidad TTN en Montevideo](#) es todavía muy pequeña con unos pocos *gateways* integrados a TTN, por lo que utilizaremos *gateways* propios. Nos basaremos en TTN, en lugar de instalar un servidor propio o utilizar el servicio de una empresa, ya que nos facilita la instalación y es de uso comunitario.

LR 2 Click (RN2903)

El módulo [LR 2 Click](#) de MikroElectronica cuenta con un módulo de comunicación RN2903 y una interfaz MikroBus. El módulo RN2903 utiliza comandos serie para su configuración y operación (similares a comandos AT). En la página del curso se encuentran publicados documentos que detallan los comandos para algunas aplicaciones.

Nota:

* El firmware del RN2903 soporta la “LoRaWAN specification 1.0.2” y fue actualizado para la región AU915, que es la utilizada en Uruguay.

Tareas y actividades a realizar

Para realizar el laboratorio deberán crear un usuario de [The Things Stack](#) eligiendo la región *North America 1* (nam1). Una vez creado el usuario, enviarle el nombre de usuario a los docentes del curso para que les den acceso a la información de los Gateway.

Registro de *gateway*

En el curso utilizaremos los gateways [TTIG](#) ya configurados por los docentes del curso. De todos modos, en caso de ser necesario, [este instructivo](#) es una referencia para configurar nuevos gateways en TTN.

Configurar conexión WIFI del *gateway* Para que el *gateway* se conecte a una nueva wifi (por defecto se conectarán a la *wifing* de Facultad de Ingeniería), es necesario presionar el botón **SETUP** durante 10 segundos. Esto hará que el *gateway* se ponga en modo AP publicando una SSID del tipo MINIHUB-xxxxxx donde xxxxxx son los últimos 6 dígitos del EUI, y la clave para conectarse a esa wifi está escrita en su etiqueta. Luego de conectarse, ingresando con un navegador a la dirección 192.168.4.1 se puede configurar una nueva wifi a la

cual el *gateway* debe conectarse. Por más información sobre este procedimiento se puede consultar [esta guía](#).

Agregado de una aplicación.

Para establecer la comunicación con un dispositivo es necesario agregar dicho dispositivo a una aplicación. Los datos que se envían a través de la red LoRaWAN serán recibidos en una aplicación para su posterior uso. Para esto entonces es necesario antes que nada crear esta aplicación, obteniendo sus datos y claves para comunicarse con ella.

1. Crear la aplicación siguiendo los pasos indicados [aquí](#) accediendo a la [consola](#). No es necesario por ahora la creación de claves API.
2. A continuación se deberán registrar los dispositivos con los que se enviarán los datos en la aplicación creada, siguiendo las instrucciones del siguiente paso.

Agregado de dispositivos y registro en una aplicación ¹

Obtener parámetro DevEUI

1. Agregar la biblioteca *TheThingsNetworks* al Arduino IDE.
2. Abrir el ejemplo `SendOTAA` de la librería *TheThingsNetworks* en el IDE de Arduino.
3. Analizar el código y deducir lo que hace.
4. Establecer la frecuencia de trabajo (`freqPlan`) según la norma TTN_FP_AU915 y agregar en el `setup()` el siguiente código para encender el módulo RN2903:

```
// Reiniciamos el módulo RN2903 con un pulso en el pin RESET
pinMode(pky, OUTPUT);
digitalWrite(pky, LOW);
delay(1000);
digitalWrite(pky, HIGH);
pinMode(pky, INPUT_PULLUP);
delay(1000);
```

donde `pky` debe ser configurado en 17 o 16 según se utilice el MikroBus 1 o MikroBus 2 en la placa de adaptación.

5. Compilar el código y programarlo en el dispositivo.
6. Observar la salida en la terminal y anotar el DevEUI.

Registrar el dispositivo en la aplicación Registrar nuestro dispositivo en una de las aplicaciones en la [consola](#), seleccionando *Manually*. Se debe especificar:

¹Para registrar un dispositivo en una aplicación se pueden seguir los pasos generales indicados para [agregar dispositivos](#). Esta sección toma en cuenta la guía publicada en [The Things Uno](#).

- Frequency plan: Australia 915-928 MHz, FSB 2 (used by TTN)
- LoRaWAN version: LoRaWAN Specification 1.0.2
- Regional Parameters version: RP001 Regional Parameters 1.0.2
- JoinEUI: Un número cualquiera, pero debe ser igual que el AppEUI que se define en el programa de Arduino (en caso que ya se encuentre registrada la dupla JoinEUI/DevEUI utilizar un número distinto para JoinEUI).
- DevEUI: obtenido en el paso anterior.
- AppKey: seleccionar generar y copiarlo para ponerlo en el programa de Arduino.

Envío de datos Establecer el parámetro **AppKey** en el programa de Arduino. Verificar la correcta configuración observando los resultados en:

1. La salida del terminal serie de Arduino (*join status*).
2. Los registros del *gateway* en TTN (*live data*).
3. Los registros en la aplicación creada en TTN (*live data*).
4. Finalmente, observar en la consola de la aplicación (*live data*) la actualización del estado del dispositivo.

Aplicación de envío de temperatura

Modificar el ejemplo anterior para enviar la temperatura de la Arduino Zero. Para esto tener en cuenta que la función `ttn.sendBytes` espera un puntero a un arreglo de bytes así como la cantidad de bytes que se deben enviar (el largo del *payload*). Los enteros en Arduino Zero (SAMD) tienen un largo de 4 bytes, y un cast sencillo puede hacerse con un código del estilo:

```
ttn.sendBytes( (uint8_t*) &temperature, 4);
```

La temperatura debe visualizarse en la interfaz de TTN. Para esto debe agregarse un *formateador de payload* ya sea a nivel general de la aplicación o a nivel de dispositivo². Aquí hay una [referencia práctica](#) para ello, resumimos a continuación un fragmento:

```
function decodeUplink(input) {
  // input has the following structure:
  // {
  //   "bytes": [1, 2, 3], // FRMPayload (byte array)
  //   "fPort": 1
  // }
  return {
    data: {
      bytes: input.bytes
    },
    warnings: ["warning 1", "warning 2"], // optional
  }
}
```

²Es muy útil la interfaz para escribir el código específico para el dispositivo porque permite probar en tiempo real el resultado para distintos *payloads*.

```

        errors: ["error 1", "error 2"] // optional (if set, the decoding failed)
    };
}

```

Aplicación con FreeRTOS

En esta parte deben escribir un programa en FreeRTOS para el envío de datos via LoRaWAN de múltiples fuentes o sensores. Para ello se basarán en el programa realizado en el Laboratorio 1 y en la tarea anterior donde utilizaron el ejemplo SendOTAA, básicamente combinando ambas soluciones y haciendo algunas modificaciones.

Seguir las siguientes directivas:

1. Partir de la solución del Laboratorio 1 y crear la tarea LoRaWANTask, tarea encargada de la comunicación LoRaWAN, a partir de (y en sustitución de) la tarea SerialTask. Las tareas Task1 y Task2 no tendrán modificaciones, manteniendo la estructura de datos para el envío de información (se identifica el dato por el tipo).
2. Agregar al inicio de LoRaWANTask la inicialización del módulo RN2903 y el registro del dispositivo en la red.
3. En loop infinito se espera la llegada de datos en la cola, para luego enviarlos de forma similar a la parte anterior.
4. Modificar el código para utilizar un puerto diferente para cada tipo de datos. Se sugiere utilizar un arreglo para mapear tipo de dato (sensor) en un número de puerto específico.
5. Actualizar el formateador javascript en la web de TTN.

Entregables y Defensa

Cada estudiante debe asistir a la defensa con el laboratorio realizado en su completitud, y habiendo contestado de manera individual -y en tiempo y forma- la encuesta correspondiente publicada en la plataforma EVA.

El día de la defensa deberán poder explicar cada una de las partes de este laboratorio.

Referencias

- <https://docs.arduino.cc/hardware/zero>
- <https://www.mikroe.com/arduino-uno-click-shield>
- <https://www.mikroe.com/lr-2-click>
- Guía *Instalación del entorno de desarrollo* publicada en la web del curso.

Los materiales de este curso fueron parcialmente financiados por:



Co-funded by the
Erasmus+ Programme
of the European Union