

Gateway LoRa experimental para despliegue de dispositivos de medición IoT

Experimental LoRa gateway for deployment of IoT measurement devices.

Presentación: 26 y 27 de octubre de 2022

Sergio Felissia

Grupo de Investigación y Desarrollo en Electrónica (GIDE), Departamento de Ingeniería Electrónica, Facultad Regional San Francisco, Universidad Tecnológica Nacional
sfelissia@facultad.sanfrancisco.utn.edu.ar

Gastón Peretti

Grupo de Investigación y Desarrollo en Electrónica (GIDE), Departamento de Ingeniería Electrónica, Facultad Regional San Francisco, Universidad Tecnológica Nacional
gperetti@facultad.sanfrancisco.utn.edu.ar

Jorge Bossio

Grupo de Investigación y Desarrollo en Electrónica (GIDE), Departamento de Ingeniería Electrónica, Facultad Regional San Francisco, Universidad Tecnológica Nacional
jbossio@facultad.sanfrancisco.utn.edu.ar

Gastón Pautasso

Grupo de Investigación y Desarrollo en Electrónica (GIDE), Departamento de Ingeniería Electrónica, Facultad Regional San Francisco, Universidad Tecnológica Nacional
gaston_paut@hotmail.com

Alejo Casas

Grupo de Investigación y Desarrollo en Electrónica (GIDE), Departamento de Ingeniería Electrónica, Facultad Regional San Francisco, Universidad Tecnológica Nacional
atlcasas15@gmail.com

Resumen

Se describe el dispositivo de medición y la red IoT ad-hoc implementada, utilizando la tecnología LoRa, a través de módulos de sistemas embebidos, cuyo concentrador (*gateway*) se ubicó en la azotea de un edificio del centro de la ciudad a una altura aproximada de 100 metros. El sistema así implementado nos permitió un radio de cobertura de alrededor de 15 km a través del enlace en 915Mhz, lo cual nos aseguró cubrir la totalidad del radio urbano de la ciudad de San Francisco. El enlace con el dispositivo de medición se realiza a través de transceptores LoRa, basados en el módulo SX1276, que permiten con una mínima potencia de 20 dbm (100mW), alcanzar el rango de cobertura deseado. Para el enlace de *backhaul* se utiliza la red 4G. La red Wifi implementada dentro del gabinete del *gateway* permite conectar el transceptor LoRa y una placa Raspberry Pi 3 con la cual se realiza la programación del módulo

LoRa en forma remota. Un servidor MQTT y una aplicación Node-Red ejecutándose en la nube y registrando en una base de datos, los valores de las variables medidas completan el paquete tecnológico.

Palabras clave: Red IoT, LoRa, Servidor MQTT, Gateway LoRa.

Abstract

The measurement device and the IoT ad-hoc network implemented using LoRa technology are described, through embedded modules, whose hub (gateway) was located on the roof of a building in the city center at an approximate height of 100 m. meters. The system implemented in this way allowed a coverage range of around 15 km through the 915Mhz link, which ensured that we covered the entire urban radius of the city of San Francisco. The link with the measure device is made through LoRa transceivers, based on the SX1276 module, which allow, with a minimum power of 20 dbm (100mW), to reach the desired coverage range. For the backhaul link, the 4G network is used. The Wi-Fi network implemented inside the gateway cabinet allows connecting the LoRa transceiver and a Raspberry Pi 3 board with which the LoRa module programming is done remotely. An MQTT server and a Node-Red application running in the cloud and recording in a database the values of the measured variables complete the technological package.

Keywords: IoT Network, LoRA, MQTT server, LoRA Gateway.

Introducción

El presente trabajo busca el desarrollo de una plataforma capaz de llevar a cabo mediciones, tanto fijas como móviles, en distintos puntos del ejido urbano de la ciudad de San Francisco. El módulo embebido de medición lee y procesa la información y la envía al concentrador (Gateway) de la red IoT que se encuentra instalado en altura, en una locación céntrica. Para el enlace con los dispositivos de medición se utilizan módulos transeptores que transmiten y reciben en una banda de frecuencias no licenciada (ISM), de 915 MHz a 928 MHz, que está identificada por la alianza de empresas que financian el desarrollo de LoRa como AU915, que es el estándar que se adoptó en Argentina. Esta forma de modulación y transmisión de RF, tiene ventajas para su utilización en aplicaciones IoT, donde se miden de variables físicas en campo. Las ventajas de la tecnología LoRa son:

- Utiliza la banda de frecuencia ISM sin licencia.
- Es una solución flexible que se puede adaptar fácilmente.
- Es escalable.
- Es compatible con la comunicación bidireccional.
- Proporciona un alto nivel de seguridad debido a los algoritmos de encriptación.
- Proporciona eficiencia energética.

Desarrollo

En la Figura 1, se observa el diagrama en bloques del sistema completo, donde se pueden identificar las dos partes que componen el sistema, el módulo concentrador y el dispositivo de medición móvil.

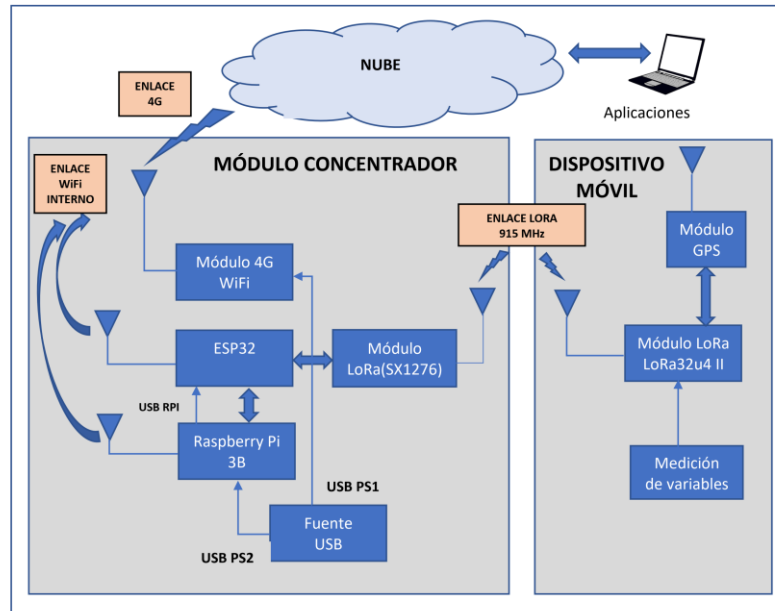


Figura 1: Diagrama en bloques del sistema completo

Módulo concentrador:

Este módulo hace las funciones de Gateway entre los dispositivos de medición móviles o fijos distribuidos sobre la superficie del ejido urbano de la ciudad y alrededores, con un radio de cobertura de aproximadamente 15 km. Consta de un módulo de radio LoRa, con su correspondiente antena, conectado a una placa embebida de la firma Espressif, ESP32 que incluye conectividad WiFi estándar. Por otro lado, un modem 4G-WiFi permite la conexión a Internet del módulo ESP32, a través de la red de telefonía celular. Se incluye dentro del módulo una placa Raspberry Pi 3B, para realizar monitoreo remoto del sistema y además realizar cambios en la programación del módulo embebido ESP32 que controla la radio LoRa, basada en el integrado SX1276 de Semtech.

La conectividad LoRa.

LoRa es un tipo de modulación SS-Spread Spectrum, y la novedad de esta técnica consiste en el uso de una señal chirp que varía constantemente con la frecuencia [2]. La ventaja de utilizar este método es que el desplazamiento en tiempo y frecuencia para el emisor y el receptor es el mismo, lo que reduce considerablemente la complejidad del receptor. La técnica de modulación LoRa es del tipo Chirp Spread Spectrum (CSS). Utiliza diferentes factores de dispersión de modulación (SF) que van desde SF7 a SF12. Este mecanismo proporciona resistencia a la interferencia y al desvanecimiento por trayectos múltiples [3],[4]. Así, es posible ajustar individualmente por nodo la tasa de modulación y la potencia de transmisión. Si se aumenta el SF, el tamaño del paquete se reducirá, lo que dará como resultado una mayor potencia sobre el canal y una mayor distancia de comunicación. La Tabla 1 presenta los factores de dispersión de LoRa para un ancho de banda de 125 kHz. Como se puede observar, si se aumenta el factor de expansión, se reduce la tasa de bits, se aumenta el parámetro de tiempo en el aire, pero se mejora significativamente el límite de SNR, y por lo tanto aumenta el rango de alcance.

Tabla 1. Factores de expansión LORA para ancho de banda de 125 KHz. [2],[3]

Spreading Factor	Symbols/second	SNR limit	Time-on air for 10 bytes packet [ms]	Bitrate [bits/sec]
7	976	-7.5	56	5469
8	488	-10	103	3125
9	244	-12.5	205	1758
10	122	-15	371	977
11	61	-17.5	741	537
12	30	-20	1483	293

Placa LoRa32u4 II

Es una placa ligera y de bajo consumo basada en el Atmega32u4 con módulo LoRa 868MHZ/915Mhz. Tiene un conversor A/D de 10bits, cargador integrado de baterías de 100 mA. El módulo LoRA está basado en el módulo SX1276 de Semtech, interfaz SPI, función RSSI digital, corrección automática de frecuencia, control de ganancia automática, función Sleep, detección de bajo voltaje y sensor de temperatura.

Receptor GPS

Se utiliza un módulo receptor GPS estándar de la empresa SIMCOM (SIM808), el cual se comunica a la placa LoRa32u4 II a través de un puerto serie, que permite configurar, monitorear y leer la información de localización GPS por medio de comandos AT. La salida de los datos de posicionamiento se configura para un formato específico del protocolo NMEA. Los datos son transmitidos a través de tramas con caracteres ASCII, separados por comas. Cada sentencia es totalmente independiente de otras. Comienza con el signo "\$" y termina con una secuencia de <CR><LF> (retorno de carro, salto de línea). Los primeros dos caracteres después del signo "\$", identifican al equipo. Para los receptores GPS, el prefijo es GP. Este va seguido de una secuencia de tres letras que definen el tipo de información que se está enviando, como puede apreciarse en la Figura 2. La información suministrada por la salida en este formato es latitud, longitud, altitud, fecha y hora.

```
AT+CGPSOUT=32
OK
$GPRMC,153832.000,A,3125.5649,S,06205.5254,W,0.21,123.80,030922,,A*62
```

Figura 2: Comandos AT para activar la transmisión de información y su correspondiente salida.

Software

Debemos identificar distintos componentes del software del sistema completo. A saber: Firmware de la radio LoRa controlada por el ESP32, el firmware del dispositivo de medición móvil, los softwares instalados en el servidor virtual, servidor MQTT y flujos NodeRED (Figura 4) [6] de manejo de la base de datos y paneles de visualización (Figura 3). Se dispone de una máquina virtual instalada en la nube a través de un proveedor (ISP) donde se encuentra instalado un servidor (Broker) de protocolo MQTT. El protocolo MQTT (*Message Queuing Telemetry Transport*) es un protocolo PubSub de *Message Service* que actúa sobre TCP. Destaca por ser ligero, sencillo de implementar. Resulta apropiado para dispositivos de baja potencia como los que frecuentemente tenemos en IoT. Está optimizado para el *routing* activo de un gran número de clientes conectados de forma simultánea. MQTT es un servicio de mensajería con patrón publicador/suscriptor (pub-sub). En este tipo de infraestructuras los clientes se conectan con un servidor central denominado *broker*[1].

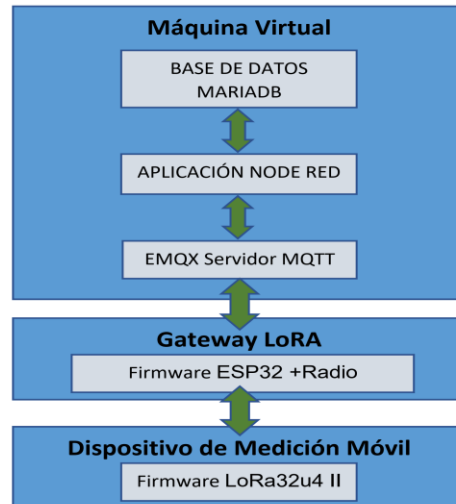


Figura 3: Esquema de los componentes de software.

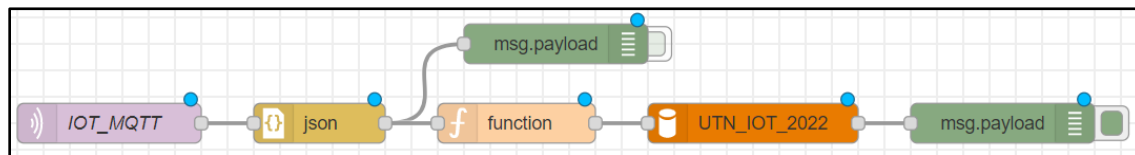


Figura 4: Flujos NodeRED para recibir objetos JSON, separar datos y almacenar en base de datos.

Los datos relevados son publicados en un tópic definido en el firmware del dispositivo de medición. El formato de publicación es JSON[6]. En la Figura 5 se observa la estructura del objeto JSON diseñado.

```

1  {
2    "Lat": "31.255649",
3    "SN": "S",
4    "Long": "62.055254",
5    "WE": "W",
6    "Alt": "153.832",
7    "Valor 1": "0.3244",
8    "Valor 2": "7.7650",
9    "Dispositivo": "02"
10 }

```

Figura 5: Estructura del objeto JSON.

Software del Concentrador LoRa-4G.

Los paquetes de software instalados en la máquina virtual que actúa como servidor son: EMQX (Servidor de MQTT)[1], Node-Red (Entorno de programación basado en el *framework* de Node.JS [4] para realizar aplicaciones ejecutables en navegador (paneles de monitoreo y registro de datos), XAMPP: paquete del servidor web Apache que incluye servidor de base de datos SQL(MariaDB) y phpMyAdmin. Se configura Node-Red para que se ejecute como servicio, en segundo plano, de modo que permanentemente realice las tareas de conexión al servidor MQTT, obtenga el paquete de datos enviado por el dispositivo, separe los valores de las variables leídas, guarde en una base de datos esos valores y presente los datos en una pantalla de monitoreo (*dashboard*).

Resultados

Se realizaron pruebas de cobertura para determinar el alcance de la transmisión de los dispositivos utilizando distintos factores de expansión SF. Los experimentos se realizaron sobre una línea recta desde el dispositivo a la antena del concentrador. En la Tabla 2 se detallan los rangos alcanzados comparativamente para cada SF.

Tabla 2: Comparación de rangos de cobertura obtenidos para diferentes SF. Ancho de banda B=125 KHz.

Factor de expansión SF	Bit rate [bit/s] (Tabla 1)	Rango [m] (experimental)
7	5469	5600
10	977	9100
12	293	12550

Conclusiones

Se concluyó que el sistema es operativamente eficiente para tomar mediciones de dispositivos fijos y móviles desplegados en el área de cobertura y almacenarlas en una base de datos alojada en un servidor externo (nube). En este trabajo se describen los componentes que se pusieron en juego para implementar una red ad-hoc para tomar mediciones de cualquier variable susceptible de ser medida a través de sensores en un área de cobertura de aproximadamente 15 km. El sistema se implementó y se utilizó para realizar algunas mediciones de prueba, de distintos dispositivos de medición fijos y móviles. para integrar el primer concentrador (Gateway) de la red piloto desplegado en la ciudad para recibir los datos de los dispositivos de medición y registrarlos de forma que, en tiempo real o en cualquier momento, puedan ser accedidos por las aplicaciones. En próximas etapas, se trabajará para escalar el sistema a una mayor cantidad de nodos, ampliar el firmware del concentrador para agregar una metodología de acceso a la red Ad-Hoc que evite la pérdida de paquetes, asegure la integridad de los datos y se eviten las colisiones. La instalación de una placa ARM Raspberry Pi a bordo del equipo concentrador, le da la flexibilidad de modificar el firmware de forma remota, característica que es muy útil para las nuevas instancias de desarrollo.

Referencias

- [1] M. Bender, E. Kirdan, M. -O. Pahl and G. Carle, "Open-Source MQTT Evaluation," *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021, pp. 1-4, doi: 10.1109/CCNC49032.2021.9369499.
- [2] A. Lavric, V. Popa. "Internet of Things and LoRa Low-Power Wide Area Networks Challenges", 2017. ECAI 2017 - International Conference – 9th Edition Electronics, Computers and Artificial Intelligence 29 June - 01 July, 2017, Targoviste, Rumania.
- [3] Semtech, "LoRa and LoRaWAN: A Technical Overview" Accedido: 12/09/2022 Recuperado de: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>
- [4] Semtech, AN1200.22 "LoRa Modulation Basics, Application Note" Accedido: 05/09/2022 Recuperado de AN1200.22.pdf. <https://semtech.my.salesforce.com/>.
- [5] Node-RED, Página web: "Node-Red Cookbook", Accedido: 05/09/2022. Recuperado de: <https://cookbook.nodered.org/#mqtt> .
- [6] JSON.ORG, Página web: "Introducing JSON", Accedido 12/09/2022. Recuperado de: <https://www.json.org/json-en.html>