Desarrollo de una red IIoT LoRaWAN para obtener y almacenar datos de máquinas y equipos industriales a través de la interfaz MODBUS TCP.

IIoT LoRaWAN Network Development to collect and store data from industrial machines and equipment through the MODBUS interface.

Presentación: 21/08/2024

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

Diego Iguri
Grupo de Investigación y Desarrollo en Electrónica (GIDE), Departamento de Ingeniería Electrónica, Facultad Regional San Francisco, Universidad Tecnológica Nacional
diguri@claro.com.ar

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

Resumen

El presente trabajo describe el desarrollo de nodos para enviar información de procesos, equipos o máquinas industriales, a través de una plataforma local IIoT (*Industrial Internet of Things*), basada en el protocolo LoRaWAN (*Long Range Wide Area Network*). En este caso se implementó un servidor MQTT (*Message Queuing Telemetry Transport*) en una máquina virtual dedicada al sistema, creada dentro de un servidor local. Se utilizaron herramientas de software GNU, como el entorno de programación "low code" denominado Node-RED y base de datos de series temporales como InfluxDB.

Se presenta la solución implementada, tomando como ejemplo dos controladores Lógicos Programables (PLC) de una marca reconocida con interfaz MODBUS TCP (*Transmission Control Protocol*), conectados al nodo IIoT desarrollado. El firmware del nodo utiliza como base el código y librerías para implementar el protocolo LoRaWAN denominado LMIC (*LoRaWAN MAC in C*), desarrollado por IBM. Se describe brevemente el despliegue

de la plataforma IIoT para la recolección y almacenamiento de la información enviada por los nodos, que incluye un único Gateway LoRaWAN.

Palabras clave: redes IIoT, protocolo Modbus, modulación LoRa, LoRaWAN.

Abstract

This work describes the development of nodes to send information from industrial processes, equipment, or machines through a local IIoT (*Industrial Internet of Things*) platform based on the LoRaWAN (*Long Range Wide Area Network*) protocol. In this case, an MQTT (*Message Queuing Telemetry Transport*) server was implemented on a virtual machine dedicated to the system, created within a local server. GNU software tools were used, such as the "low code" programming environment called Node-RED and time series databases such as InfluxDB.

The implemented solution is presented, using as an example two Programmable Logic Controllers (PLCs) from a well-known brand with MODBUS TCP interface, connected to the developed IIoT node. The node's firmware is based on the code and libraries for implementing the LoRaWAN protocol, known as LMIC (*LoRaWAN MAC in C*), developed by IBM. A brief description of the deployment of the IIoT platform for the collection and storage of information sent by the nodes is provided, which includes a single LoRaWAN Gateway.

Keywords: IIoT networks, Modbus protocol, LoRa modulation, LoRaWAN.

Introducción

Las tecnologías LPWAN (*Low Power Wide Area Network*) se encuentran muy desarrolladas en el mundo e Internet de las Cosas se nutre de estas herramientas de conectividad inalámbrica para llevar adelante los aspectos que tienen que ver con la capa física del modelo y que buscan obtener datos del mundo físico. LoRaWAN forma parte de este conjunto de recursos de hardware y software que resuelven las necesidades de comunicación de baja potencia, bajos requerimientos de velocidad de transmisión y amplio alcance.

LoRa (*Long Range*)(capa física de LoRaWAN) utiliza una técnica de modulación de espectro ensanchado derivada de la tecnología CSS (*Chirp Spread Spectrum*), que ofrece un compromiso entre sensibilidad y velocidad de datos, mientras opera en un canal de ancho de banda fijo de 125 kHz o 500 kHz (para canales de enlace ascendente), y 500 kHz (para canales de enlace descendente). Además, utiliza factores de dispersión (SF) ortogonales.

Un *gateway* LoRaWAN recibe mensajes modulados en *LoRa* desde cualquier dispositivo final en la distancia de audición y reenvía estos mensajes de datos al *Network Server* (NS), que está conectado a través de una red troncal IP. No existe una asociación fija entre un dispositivo final y *gateway* específico. En cambio, el mismo sensor puede ser atendido por múltiples *gateways* en el área. Con LoRaWAN, cada paquete de enlace ascendente enviado

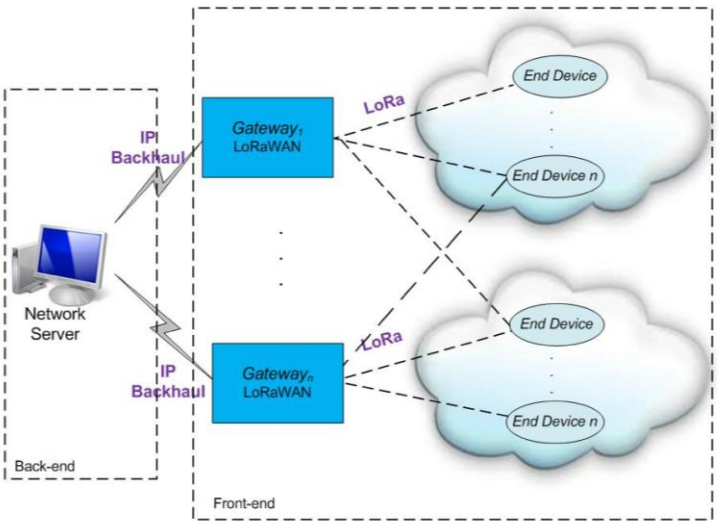

Figura 1. Topología básica de LoRaWAN. (Lavric A. et al., 2017)

por el dispositivo final será recibido por todas las puertas de enlace a su alcance, como se ilustra en la Figura 1.

Esta disposición reduce significativamente la tasa de errores de paquetes (ya que las posibilidades de que al menos una puerta de enlace reciba el mensaje son muy altas), reduce significativamente la sobrecarga de la batería para sensores móviles y permite la geolocalización de bajo costo (suponiendo que las puertas de enlace en cuestión tengan capacidad de geolocalización) (Lavric A. et al., 2017).

Si bien LoRaWAN está orientada a redes de área extendida, como por ejemplo aplicaciones de Smart City, sus características de robustez y seguridad lo hacen particularmente interesante para implementaciones industriales de IIoT. En este caso, la modulación de espectro expandido LoRa se adapta perfectamente a ambientes con alta presencia de ruido eléctrico.

A raíz del crecimiento del campo de investigación de IoT en años recientes, diversas bibliotecas de LoRaWAN han sido creados para plataformas de desarrollo como Arduino. Entre estas bibliotecas, una de las más destacadas y utilizadas es LMIC de IBM lanzada en 2016. Esta biblioteca es compatible con el entorno de Arduino y soporta las especificaciones 1.0.2 y 1.0.3 (Terry M.,2021).

Para cumplir con el estándar, en los nodos implementados con sistemas embebidos compatibles con Arduino se instaló la biblioteca LMIC para Arduino. La biblioteca LMIC proporciona una implementación bastante completa de LoRaWAN de clase A y clase B, compatible con las bandas EU-868, US-915, AU-915, AS-923 e IN-866.

Protocolo MODBUS TCP

Modbus TCP es una comunicación de tipo cliente/servidor a través de una red Ethernet TCP/IP (Liu Q; Li Y, 2006: 432-435). Modbus TCP es equivalente a Modbus RTU, que se ejecuta a través de una interfaz Ethernet.

El ciclo de mensajería de Modbus TCP consta de cuatro pasos. En el primer paso, el cliente envía una consulta (solicitud de conexión) al servidor; en el segundo paso, el servidor reconoce o acepta esta consulta; en el tercer paso, el servidor envía respuestas para el código de función y, en el cuarto paso, el cliente envía una señal de confirmación al servidor que puede desconectar la conexión TCP. El formato de mensaje de Modbus TCP se muestra en la Figura 2.

Figura 2. Formato de mensajes de Modbus TCP

Este formato de mensaje consta de MBAP (*ModBus Application Protocol*), de siete bytes que incluye el identificador de transacción, el identificador de protocolo, la longitud del mensaje y el identificador de cliente. En Modbus TCP, para que el servidor se conecte, se requiere el identificador de conexión y el número de puerto del servidor para establecer la comunicación y, para un cliente, se requiere una dirección IP del servidor, el identificador de cliente y el número de puerto en el formato del mensaje.

El código de función es un campo de dirección de 1 byte. El código de función en cuestión le dice al dispositivo esclavo qué tipo de acción ejecutar (Xiong P et al., 2003: 586-590). El código de función es un dato de 1 byte. Algunos códigos de función que se usaron con frecuencia en el momento de la comunicación se muestran en la TABLA 1 (MODBUS.ORG, Página web).

TABLA 1. Códigos de función.

Function Code	*Action*
01	Read discrete output coils
02	Read discrete input contacts
03	Read analog output holding registers
04	Read analog input registers
05	Write single discrete output coils
06	Write single analog output holding registers
15	Write multiple discrete output coils
16	Write multiple analog output holding registers

Desarrollo

Se dispuso de un *gateway* marca Milesight, modelo UG65, para interiores de 8 canales. Este dispositivo adopta el chip SX1302 LoRa de Semtech con una CPU de cuatro núcleos de alto rendimiento, admite la conexión con más

de 2000 nodos. El modelo UG65, que disponemos, admite dos sistemas de conectividad *back-haul* con Ethernet y WiFi. Tiene un *Network Server* (NS) incorporado, así como también puede ser integrado a NS en plataformas en la nube (como *The Things Network*, *ChirpStack*, *Milesight IoT Cloud* y otras).

Se configuró el *gateway* para utilizar el *Network Server* integrado. Se seleccionó el protocolo MQTT como el modo de transmisión de *backhaul* hacia la nube de la información recibida desde los nodos. Los datos son enviados en un formato de texto tipo JSON (JSON.ORG, Página web: "Introducing JSON"). La clave "data" contiene el dato enviado por el nodo identificado a través de las claves "devEUI" y "deviceName". El dato es codificado por el *gateway* en Base64. Además del dato enviado por el nodo, el *gateway* envía otra información relacionada con parámetros de recepción de la radio LoRa, como nivel de señal (RSSI), relación señal ruido (LoRaSNR), frecuencia (frequency) y factor de expansión (spreadFactor).

Se dispuso de una máquina virtual instalada en la nube a través de los servidores de la Institución, donde fue instalado un servidor de protocolo MQTT.

Los paquetes de software instalados en la máquina virtual son: EMQX (Servidor de MQTT), Node-RED (Entorno de programación basado en el framework de Node.JS) para realizar aplicaciones ejecutables en navegador (Node-RED, Página web: "Node-RED Cookbook", 2022); InfluxDB es una base de datos de series temporales (TSDB) de código abierto desarrollada por la empresa InfluxData. Se utiliza para el almacenamiento y la recuperación de datos de series temporales en campos como la monitorización de operaciones, las métricas de aplicaciones, los datos de sensores de Internet de las cosas y el análisis en tiempo real (Turnbull J.: 206-, 2014).

Nodo MODBUS

Se implementó un hardware compuesto por una placa controladora ESP32 a la cual se conectó a través de sendos puertos SPI (*Serial Peripheral Interface*) identificados como VSPI y HSPI, el módulo de radio LoRa y el módulo Ethernet W5500 (Figura 3). Se observa una foto del prototipo implementado en la Figura 4, donde se puede conectar al puerto Ethernet cualquier equipo con comunicación compatible con MODBUS TCP.

Se incluyen en el software de la placa ESP32, la librería LMIC para la comunicación LoRaWAN, la librería para administrar la comunicación Ethernet y librería MODBUS para Arduino.

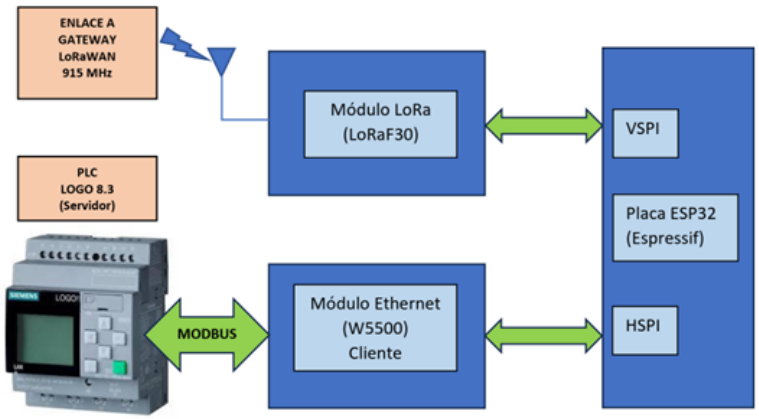

Figura 3: Diagrama en bloques del nodo LoRaWAN MODBUS conectado a un PLC Siemens LOGO.

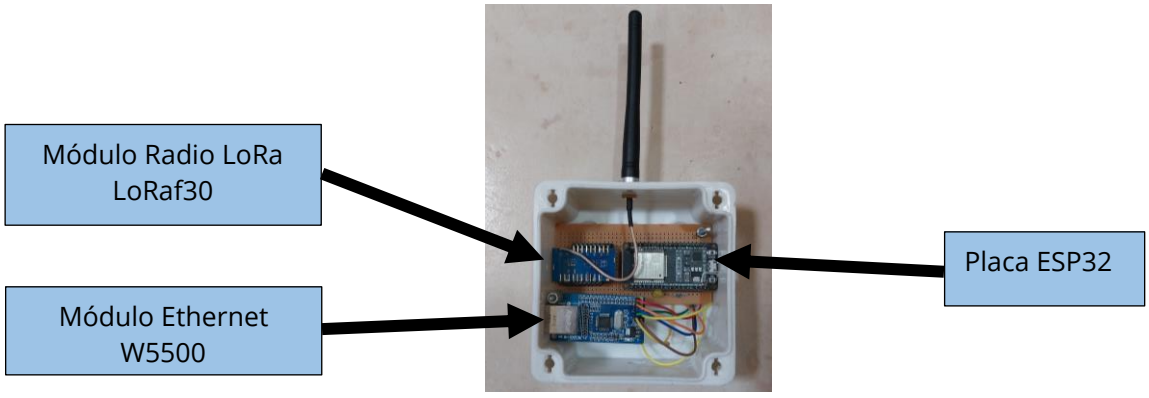

Figura 4: Prototipo del nodo LoRaWAN MODBUS.

Esta biblioteca implementa el protocolo Modbus sobre dos tipos de transporte diferentes: comunicación serial sobre RS485 con RTU (*Remote Terminal Unit*) o comunicación Ethernet y WiFi con protocolo TCP. Existen algunas diferencias en las API según el transporte, pero la mayoría de las funciones son las mismas para ambos. Modbus también es un protocolo cliente-servidor donde Cliente = maestro y Servidor = esclavo en la terminología Modbus (Página web, Librería MODBUS para Arduino en www.arduino.cc).

Como regla general, la comunicación RTU es multipunto y, por lo tanto, se debe especificar el ID de la unidad involucrada en la comunicación. TCP es punto a punto utilizando la dirección IP y, por lo tanto, no es necesario un ID en los parámetros.

En este proyecto se utilizó el transporte TCP utilizando la interfaz Ethernet. Se conectó a través de esta interfaz, un PLC de la empresa Siemens, modelo LOGO 8.3, el cuál puede ser configurado como Servidor.

La API de la librería suministra funciones para lectura y escritura de entradas, salidas, marcas y registros.

Se programó para enviar a través de LoRaWAN el estado binario de ocho entradas, ocho salidas y 8 marcas, un total de 24 bits (3 bytes) para estos campos. La trama se inicia con un byte ID del dispositivo, luego otro byte para el tipo de dispositivo, continúa con los tres bytes que contienen el estado de entradas, salidas y marcas, finalmente, ocho bytes para valores analógicos, registros, contadores y valores de timers. En la TABLA 2 se puede apreciar la distribución de los distintos campos en el payload.

TABLA 2: Distribución de los campos en el payload enviado por los nodos. En total 13 bytes.

ID del equipo (1byte)	Tipo de Equipo (1 byte)	Entradas (1 byte)	Salidas (1 byte)	Marcas (1 byte)	Registros, Contadores, Timers, Valores analógicos (8 bytes)
Número asignado al nodo	0-Otros 1-PLC 2-Drive	I1 a I8 en PLC LOGO	Q1 a Q8 en PLC LOGO	M1 a M8 en PLC LOGO	Para usos múltiples.

Se programó para que la trama de 13 bytes binarios se transforme en un string de 13 caracteres realizando una codificación en Base64. Base64 es un grupo de esquemas de codificación de binario a texto que transforman datos binarios en una secuencia de caracteres imprimibles, limitada a un conjunto de 64 caracteres únicos. Más específicamente, los datos binarios de origen se toman de a 6 bits por vez, luego este grupo de 6 bits se asigna a uno de los 64 caracteres únicos. Esquemas de codificación Base64 son frecuentemente utilizados cuando se necesita codificar datos binarios sobre todo cuando hay que almacenarlos y transferirlos a través de medios diseñados para tratar textos. Esta forma de codificación puede asegurar que los datos permanezcan intactos sin modificaciones durante el transporte. Se usa Base64 frecuentemente en varias aplicaciones incluso el correo electrónico a través de MIME, así como el almacenamiento de datos complejos en XML o JSON. (Página web, base64decode.org)

En el software del nodo se realiza la lectura del estado de las variables del PLC, de forma periódica, cada 100 milisegundos. El envío de información puede ser realizada de 2 maneras:

1) Forma periódica.
2) Por eventos.

En el primer caso, se programa un período de tiempo entre cada envío, que puede ser 10 segundos, 1 minuto, 5 minutos o 10 minutos.

En el segundo caso, se puede configurar uno o varios bits del campo de entradas, salidas y marcas como disparo del envío cuando se produce un cambio de estado. En el caso de varios bits, se realiza una operación EXOR con los todos los bits seleccionados para detectar un cambio de estado en alguno de los mismos.

Se creó en la máquina virtual un flujo en el entorno de programación "*low code*" denominado Node-RED (Figura 5), para recibir el *payload* desde el *gateway*, a través de la suscripción al tópico correspondiente en el protocolo MQTT, extraer los distintos campos del objeto JSON recibido e insertar información en la base de datos creada en InfluxDB.

Conclusiones

Se han realizado las pruebas con el nodo descripto y un Gateway instalado en el exterior de la institución. Los nodos disponen de una antena externa. Los resultados han mostrado que el *gateway* recibe datos de los nodos hasta con niveles de recepción de hasta -120 dBm. Con este límite, se ha observado que el nodo pudo realizar sus

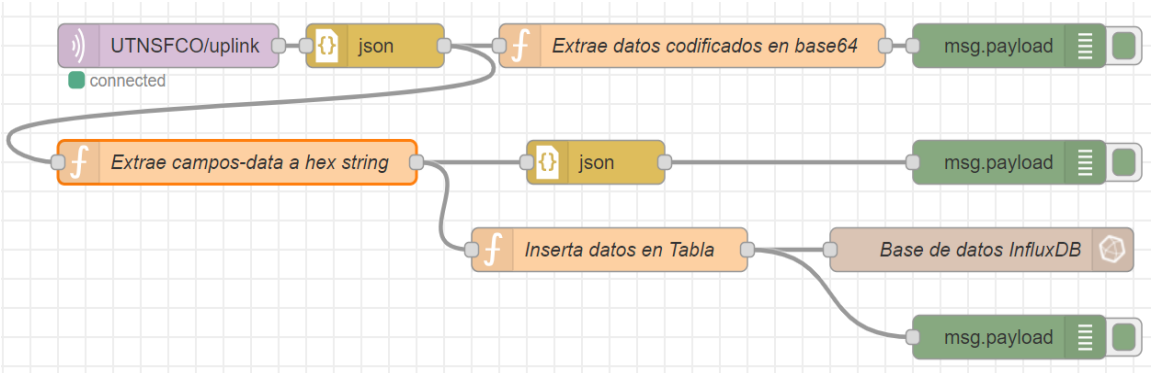

Figura 5. Flujos Node-RED para recibir el payload MQTT desde el *gateway*, separar e insertar datos en la base de datos InfluxDB.

envíos con éxito, es decir recibido por el Gateway y procesados por la aplicación Node-RED instalada en la máquina virtual, en un entorno con muchos obstáculos como paredes, techos de chapas metálicas y hormigón, aproximadamente en un radio de cobertura de 80 metros del Gateway. Resultaría ventajoso aumentar la cantidad de Gateways.

Los objetivos planteados al inicio de este trabajo, que prioritariamente eran los de construir una plataforma IIoT basada en LoRaWAN y un protocolo de gran uso en los equipos industriales como MODBUS, que nos permitiera la posibilidad de conectar equipos como PLCs, variadores de velocidad de motores, servocontroladores, para obtener y registrar información valiosa para el análisis de los procesos industriales como una herramienta más del paradigma de Industria 4.0.

Este trabajo sentará las bases para otros trabajos futuros, analizando el desempeño de la red en muchos escenarios distintos. Se pudo implementar una plataforma propia en la nube, de bajo costo, sin necesidad de recurrir a servicios de terceros. Aspectos como respuesta del sistema ante multiplicidad de nodos, pérdidas de paquetes, tiempo en el aire y consumo de energía de los nodos, funcionamiento en las distintas clases de dispositivos LoRaWAN, interferencias, colisiones, son alguno de los temas a considerar para posteriores investigaciones.

Referencias

ARDUINO.CC, Librería MODBUS para Arduino. Página web. Disponible en: https://www.arduino.cc/reference/en/libraries/arduinomodbus/

BASE64DECODE.ORG, Página web. Disponible en: https://www.base64decode.org/es/

JSON.ORG, Página web: “Introducing JSON”. Disponible en: https://www.json.org/json-en.html

Lavric, A; Popa, V. “Internet of Things and LoRa Low-Power Wide Area Networks Challenges”, 2017.

Liu Q; Li Y, “Modbus/tcp based network control system for water process in the firepower plant.” In Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on vol. 1 2006, pp 432-435.

MODBUS.ORG, Página web. Disponible en: https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Node-RED, Página web: “Node-Red Cookbook”. Disponible en: https://cookbook.nodered.org/#mqtt

Terry M., “"MCCI LoRaWAN LMIC Library", v4.1.1, 2021. GitHub. [online] Available: https://github.com/mcci-catena/arduino-lmic.

Turnbull J., “The Art of Monitoring”, 2014. pp. 206–. ISBN 978-0-9888202-4-1.

Xiong P.; Huang S; Yi K; Zhu K, “Design of communication port between dcs and computers of rtu” in Machine Learning and Cybernetics, 2003 International Conference on, vol. 1, Nov 2003, pp. 586-590 Vol. 1.