

SISTEMA AUTÓNOMO DE TELEMETRÍA A LARGA DISTANCIA LoRa – IoT DE VARIABLES AMBIENTALES

AUTONOMOUS LONG DISTANCE TELEMETRY SYSTEM LoRa – IoT OF ENVIRONMENTAL VARIABLES

Denisse Yemirel Bautista Vivanco

Instituto Politécnico Nacional, México
dbautistav1901@alumno.ipn.mx

Luz Noé Oliva Moreno

Instituto Politécnico Nacional, México
loliva@ipn.mx

Ian Ilizaliturri Flores

Instituto Politécnico Nacional, México
iflores@ipn.mx

Recepción: 26/noviembre/2024

Aceptación: 27/diciembre/2024

Resumen

Este trabajo presenta el desarrollo de un sistema de monitoreo remoto de variables ambientales utilizando un sistema embebido para facilitar la telemetría a larga distancia y en entornos de difícil acceso. El sistema está conformado por un nodo acoplado a un panel solar con baterías recargables para su autonomía y sin conexión a red eléctrica. El nodo se encuentra diseñado para adaptar un conjunto de sensores y de manera predeterminada está equipado con un sensor de temperatura y presión. El sistema se diseñó con el objetivo de tener una eficiencia energética alta y largas distancias de transmisión, utilizando tecnología de comunicación LoRa (Long-Range) y sistemas electrónicos apropiados. En conjunto se desarrolló un dispositivo puerta de enlace (Gateway) LoRa- Ethernet con protocolo MQTT (Message Queuing Telemetry Transport) enlazado al servidor Eclipse Mosquitto para su futuro almacenamiento y procesamiento a la nube de cómputo AWS IoT (Amazon Web Services Internet of Things). El sistema alcanzó 271.55 m de transmisión, demostrando ser una solución innovadora para el monitoreo remoto sin depender de redes celulares comerciales o infraestructura convencional, abriendo nuevas oportunidades para la telemetría cualquier entorno.

Palabras Claves: Gateway, LoRa, Nodo, Sistema embebido, Variables ambientales.

Abstract

This work presents the development of a remote monitoring system for environmental variables using an embedded system to facilitate long-distance telemetry and in difficult-to-access environments. The system is made up of a node coupled to a solar panel with rechargeable batteries for its autonomy and without connection to the electrical grid. The node is designed to accommodate a set of sensors and by default is equipped with a temperature and pressure sensor. The system was designed with the objective of having high energy efficiency and long transmission distances, using LoRa (Long-Range) communication technology and appropriate electronic systems. Together, a LoRa-Ethernet gateway device was developed with the MQTT (Message Queuing Telemetry Transport) protocol linked to the Eclipse Mosquitto server for future storage and processing in the AWS IoT (Amazon Web Services Internet of Things) computing cloud. . The system reached 271.55 m of transmission, proving to be an innovative solution for remote monitoring without depending on commercial cellular networks, or conventional infrastructure, opening new opportunities for telemetry in challenging environments.

Keywords: *Embedded system, Environmental variables, Gateway, LoRa, Node.*

1. Introducción

Actualmente, el monitoreo de variables ambientales es crucial para abordar problemas en la gestión medioambiental. Además, es una oportunidad para innovar en sectores donde se requiere información, pero no es posible mantener un monitoreo permanente debido al acceso a áreas remotas, pues las condiciones físicas y el costo que ello conlleva son un desafío. Derivado de lo anterior, el monitoreo remoto de variables es una oportunidad de innovación y desarrollo en agricultura inteligente, meteorología, gestión de desastres naturales y de recursos ambientales, siendo una herramienta clave para impulsar el desarrollo de sistemas sustentables.

Por su parte, [Singh, 2023] menciona que la implementación de LoRa en aplicaciones inteligentes ha demostrado ser eficaz para la creación de infraestructuras conectadas de manera ubicua, permitiendo servicios y aplicaciones innovadoras que son atractivas tanto para los consumidores como para la industria. De igual manera, [Dinesh, 2024] señala que la tecnología LoRa proporciona una solución de bajo costo y bajo consumo energético para la agricultura inteligente, facilitando la recolección y transmisión de datos críticos, mejorando así la toma de decisiones, por ejemplo, en la gestión agrícola.

De acuerdo con [Yang, 2020], la tecnología LoRa, conocida por sus capacidades de comunicación de largo alcance y bajo consumo de energía, se está integrando cada vez más en los sistemas de agricultura inteligente para mejorar la eficiencia y la fiabilidad de la transmisión de datos en extensos campos agrícolas.

Contemplando las mismas ventajas, [Sulaiman, 2022] considera que la tecnología LoRa ha ganado considerable atención debido a su capacidad para proporcionar conectividad de largo alcance con un bajo consumo de energía, lo cual la hace ideal para aplicaciones en el Internet de las Cosas. Por su parte, [Haxhibeqiri, 2018] resalta que esta tecnología ha atraído la atención en la investigación debido a su capacidad para comunicarse con bajo consumo y baja tasa de datos en grandes áreas.

[Gaitan, 2020] también considera que la tecnología LoRa es innovadora y eficiente para transmitir datos a baja velocidad y con bajo consumo en áreas extensas, siendo útil en la detección de incendios forestales en tiempo real, permitiendo una rápida respuesta en zonas no vigiladas. No obstante, esta no es la única utilidad de LoRa. De acuerdo con [Edwin, 2022], la tecnología LoRa es muy adecuada para aplicaciones de IoT en la agricultura inteligente debido a su amplio alcance, bajo consumo de energía y capacidad para manejar un gran número de nodos en una red. De igual manera, [Ji, 2019] resalta que el esquema de monitoreo basado en LoRa en el sector agrícola con IoT ofrece una solución robusta y eficiente para la supervisión remota de las condiciones físicas, destacando su capacidad para cubrir grandes áreas con una infraestructura de red mínima. Dado lo anteriormente mencionado, se puede afirmar que la tecnología LoRa es una herramienta que

permite la obtención constante de información, facilitando la interpretación de grandes volúmenes de datos y la identificación de patrones y tendencias significativas de forma automatizada y eficiente.

La implementación de una comunicación inalámbrica con tecnología LoRa es de gran utilidad para este tipo de sistemas donde se requiere un bajo consumo energético y largo alcance, pues permite la independencia de las redes celulares comerciales para el envío de información.

Para abordar esta situación, se desarrolló el sistema en dos partes. En primera instancia, se trabajó con el nodo de transmisión de información, como ya se ha descrito, el cual envía inalámbricamente la información a la segunda parte del sistema desarrollado, que es el Gateway LoRa-Ethernet, que recibe la información y la sube al servidor Eclipse Mosquitto a través de la tarjeta Raspberry Pi.

2. Métodos

Diseño conceptual del sistema de comunicación a larga distancia

El sistema general de monitoreo remoto consiste en dos subsistemas. Uno de ellos está encargado de la recolección y transmisión de variables, operando de manera remota e independiente a través de un sistema de energía solar fotovoltaica. El otro subsistema es el encargado de la recepción de la información, el cual está conformado por un Gateway que recibe los datos utilizando el protocolo de comunicación LoRa. Posteriormente, mediante una tarjeta Raspberry Pi, la información se sube a un servidor utilizando comunicación Ethernet y el protocolo MQTT. La estructura general del sistema se puede observar en la Figura 1.



Fuente: elaboración propia

Figura 1 Diagrama a bloques del sistema de monitoreo remoto.

Diseño del sistema embebido para la recopilación y transmisión de las variables

Se consideraron los siguientes elementos para su incorporación en la placa del subsistema embebido de transmisión:

- Sensor de temperatura y presión BMP280.
- Pantalla de visualización tipo OLED SSD1306.
- Módulo de comunicación LoRa E220-900T22D.
- Microcontrolador Rp2040 a través de la tarjeta Raspberry Pi Pico.

Las variables de monitoreo son temperatura y presión, ya que son las que proporciona el sensor seleccionado, el BMP280. Cabe destacar que para la implementación de dicho dispositivo se tomaron en cuenta las siguientes consideraciones:

- La medición de presión está vinculada y es directamente dependiente de la medición de temperatura.
- Tiene un error por luminosidad, por ello se debe evitar que la luz incida directamente sobre él.
- El protocolo de comunicación empleado es I²C.

También se consideró el sensor BME280 para futuras implementaciones y pruebas, ya que funciona bajo el mismo protocolo de comunicación y, adicionalmente, proporciona mediciones para una variable extra: la humedad.

El funcionamiento del sistema, al trabajar con los dispositivos mencionados, consiste en tomar las mediciones (inicialmente de temperatura y luego de presión) a través del sensor BMP280. La información se visualiza en la pantalla OLED SSD1306 para verificar los datos durante las pruebas de autonomía, y posteriormente las mediciones se envían a través del módulo LoRa E220-900T22D. Además, se integró el módulo de tecnología LoRa SX1276 al sistema embebido para reemplazar al módulo principal en caso de que alguna situación externa impida su funcionamiento. Derivado del microcontrolador seleccionado, se utilizó el

lenguaje de programación MicroPython para operar los dispositivos listados, bajo los protocolos de comunicación que se muestran en la Tabla 1.

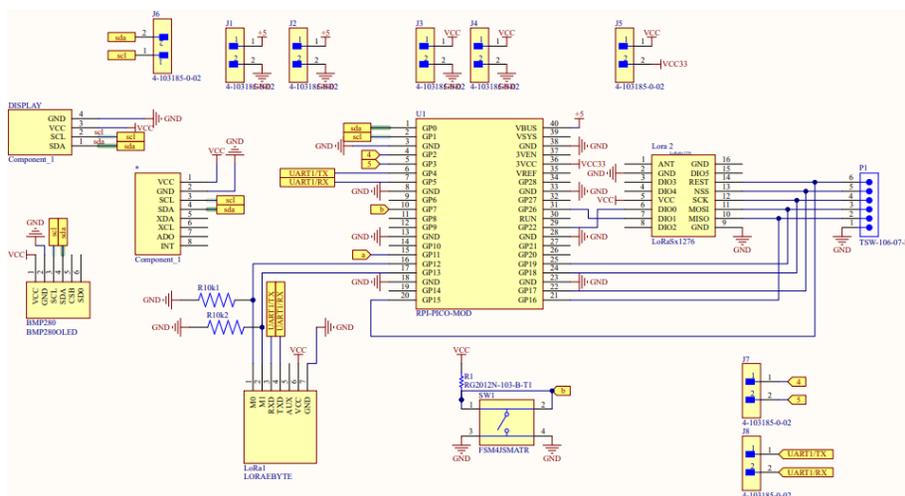
Tabla 1 Protocolos de comunicación empleados.

Dispositivo	Protocolo de comunicación
BMP280	I2C
Oled SSD1306	I2C
LoRa E220-900T22D	UART
Lora SX1276	SPI

Fuente: elaboración propia

El sistema embebido tiene una arquitectura de hardware flexible, lo que permite realizar el cambio de sensores o dispositivos sin la necesidad de construir una nueva placa. Por ello, el diseño de la placa de circuito impreso (PCB) integró las conexiones a través de pines, los cuales permiten conectar los dispositivos al microcontrolador sin necesidad de cableado, pero con la facilidad de desmontarse del sistema.

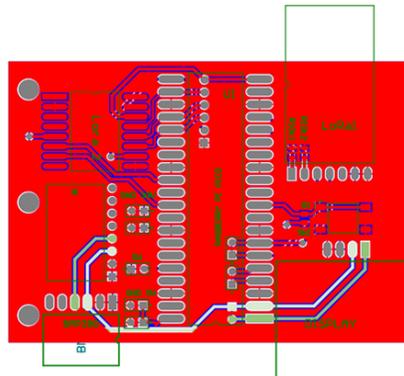
Bajo este mismo enfoque, dentro del sistema de transmisión se implementaron elementos adicionales para activar o desactivar manualmente el sistema en caso de ser necesario. Entre estos se encuentran pines para las conexiones de alimentación, entre otros, como se puede observar en la Figura 2.



Fuente: elaboración propia

Figura 2 Esquemático del sistema de transmisión.

Una vez definidos los pines del microcontrolador y sus conexiones, se realizaron algunas pruebas de funcionamiento para verificar dichas conexiones. Posteriormente, se diseñó la tarjeta de circuito impreso (PCB) para el sistema utilizando el software Altium, como se muestra en la Figura 3.

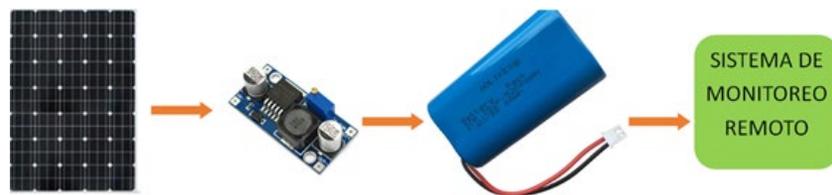


Fuente: elaboración propia

Figura 3 Diseño PCB para el sistema embebido de medición y transmisión.

Implementación del subsistema de potencia

Para garantizar la autonomía en el nodo, se implementó un sistema de alimentación con energía solar utilizando una celda fotovoltaica de 10 watts a 12 V, que produce corriente directa. Para reducir el voltaje de alimentación a 3.3 V, se incorporó un módulo Buck LM2596 de conversión DC-DC, el cual ayuda a regular el voltaje de alimentación y permite almacenarlo en un banco de baterías, Figura 4.



Fuente: elaboración propia

Figura 4 Diagrama de bloques de sistema de alimentación-potencia.

Configuración del sistema de recepción de información

El sistema de recepción está conformado por el Gateway E90-DTU y una Raspberry Pi modelo 3B. Su funcionamiento radica en que el Gateway, que también posee tecnología LoRa, recibe la información enviada por el módulo E220-900T22D,

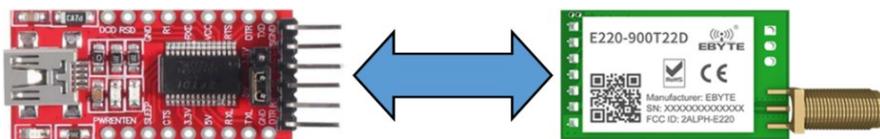
los cuales, según la descripción del fabricante, tienen una distancia de cobertura de 10 km. Se planea que el Gateway se ubique al aire libre para facilitar la recepción de la señal. No obstante, trabaja en conjunto con la Raspberry Pi para subir la información al servidor Eclipse Mosquitto en un tópico denominado “prueba/nueva/lora”. La Raspberry Pi debe estar conectada a una red WiFi y al alcance del usuario para poder acceder a ella de forma remota, o bien, a través de la conexión básica que incluye un monitor, ratón y teclado. Para lograr esto, se implementó la tecnología PoE (Power over Ethernet).

Comunicación E220-900T22D con el Gateway E90-DTU

Si bien se ha mencionado que ambos dispositivos cuentan con tecnología LoRa, esto no es suficiente para establecer comunicación entre ellos. Es necesario configurar el canal en ambos dispositivos para que coincidan en la frecuencia de operación del continente americano, que es de 915 MHz, dato proporcionado por el fabricante, solo así se podrá realizar correctamente el proceso de transmisión y recepción.

Configuración de canal del LoRa Ebyte E220-900T22D

Para configurar el canal del módulo E220-900T22D, se empleó un módulo conversor USB a serial TTL, siguiendo las conexiones que se muestran en la Figura 5, posteriormente se descargó del sitio web de Ebyte el software de configuración correspondiente a este módulo.

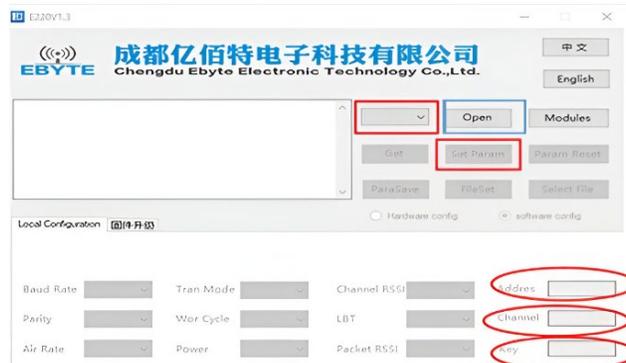


Fuente: elaboración propia

Figura 5 Conexión del módulo E220-900T22D al conversor FT32.

La configuración del canal consistió en abrir el software proporcionado por el fabricante, habiendo conectado el módulo a un puerto serial o USB de la computadora a través del conversor FT32. Una vez verificada la conexión o el puerto

COM correspondiente al dispositivo en el administrador de dispositivos, se abrió el software y se desplegó una ventana, Figura 6.



Fuente: elaboración propia

Figura 6 Software de configuración para el módulo ebyte E220-900T22D.

Como se indicó en la Figura 6, en la parte superior se seleccionó el puerto COM correspondiente al dispositivo y, se accedió a los parámetros de configuración que se muestran en la parte inferior. El fabricante indica que se debe modificar únicamente la parte de canal al número 65. Esta selección se basó en que la frecuencia con la que trabaja el dispositivo en el continente americano, que es de 915 MHz, corresponde al número de canal 65. También era posible configurar los parámetros de "address" y "key", cuidando que, al configurar el Gateway, estos parámetros coincidan. En este caso, ambos permanecieron en 0. Posteriormente, se fijaron los parámetros y se cierra esta configuración

Configuración de Gateway E90 DTU

Para realizar esta configuración, se trabajó con el software proporcionado por el fabricante Ebyte. Sin embargo, en este caso, dado que el Gateway operaría a través de conexión Ethernet, se configuró mientras estaba conectado al dispositivo con el que iba a operar. Debido a esto, no fue posible realizar la configuración mediante la descarga del software, ya que se trabajó con la tarjeta Raspberry Pi, que utiliza Raspbian como sistema operativo, y Ebyte no proporciona un software ejecutable para este sistema. Por ello, la configuración debe realizarse en línea dentro de la Raspberry Pi, ya que esta permite acceder a sitios web a través de la conexión WiFi,

con la que sí cuenta para conectarse a la red. Debido a lo anterior, se comenzó a trabajar con la Raspberry Pi. Inicialmente, se configuró la tarjeta de desarrollo con el sistema operativo Raspbian y, posteriormente, para poder conectar con el sitio web de configuración en línea del Gateway, se modificó la IP estática de la red “Eth0” de la tarjeta Raspberry. Para ello, se conectó el Gateway con un conector de tipo Ethernet a la tarjeta Raspberry Pi y se procedió a modificar el archivo “dhcpcd” a través del comando: “sudo nano /etc/dhcpcd” en la consola. Dentro de este archivo, se retiró el comentario de las líneas correspondientes a la configuración de la red Ethernet, Figura 7. Una consideración de suma importancia es conectar la antena al Gateway antes de encenderlo para evitar dañar el equipo.

```
# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caac:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caac:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
```

Fuente: elaboración propia

Figura 7 Configuración de IP estática de la tarjeta Raspberry Pi.

Dentro de la configuración Ethernet del archivo mencionado, se cambió la dirección IP a 192.168.4.100. Esta selección se realizó teniendo en cuenta que es una dirección IP adecuada para establecer conexión con el Gateway, que tiene asignada la dirección IP 192.168.4.101. Para garantizar la conexión entre ambos, deben coincidir al menos los primeros 8 dígitos de la dirección IP. Asimismo, por indicaciones del fabricante, fue necesario modificar la máscara de subred, que tuvo que ser 255.255.255.0. Como indicativo para configurarla bajo esta dirección, se adicionó el número 24 a la IP Ethernet de la siguiente manera: 192.168.4.0/24. Una vez modificada la IP Ethernet de la tarjeta Raspberry (manteniendo la conexión física a través de Ethernet entre la tarjeta Raspberry y el Gateway), se abrió el

navegador y se escribió la dirección IP del Gateway, que es 192.168.4.101, lo que redirigió al sitio de configuración en línea. En la Figura 8, se muestra la sección de parámetros a configurar. Dentro de dicha configuración, solo se ajustaron algunos parámetros. En primer lugar, se modificó el canal, estableciéndolo en 65 para hacerlo coincidir con el canal del módulo E220-900T22D. Además, en la sección de "Address" y "Key", se mantuvieron ambos parámetros en 0 para coincidir con la configuración del otro módulo.

LORA parameter		
mode address	0	Range:1-65535s
Channel	23	Range:0-83
Net ID	0	Range :0-255
Air baud	2.4Kbps	
Tx power	maximum	
Trans mode	Normal	
packetlength	240 Bytes	
WOR role	Close	
WOR cycle	2000ms	
KEY	0	Range:1-65535s
<input type="checkbox"/> LBT enable <input type="checkbox"/> Relay enable <input type="checkbox"/> Packet Rssi <input type="checkbox"/> Channel Rssi		
Enable remote config		Configure remote Lora

Fuente: elaboración propia

Figura 8 Configuración de parámetros de tipo LoRa para el Gateway E90 DTU de Ebyte.

Comunicación del Gateway E90-DTU con el servidor Eclipse Mosquitto

Antes de cerrar la configuración en línea del Gateway, también se modificaron otros parámetros que permitieron la comunicación con el servidor implementado de la plataforma Eclipse Mosquitto. Para ello, dentro de la misma configuración, en el apartado de "Network Parameter", se seleccionó el modo "TCP server", ya que el Gateway es el dispositivo receptor de información y es el encargado de publicar dicha información recibida en un tópico dentro del servidor Mosquitto. Se dejó el puerto local en 8886, Figura 9.

Network parameter		
DHCP	disable	
Work mode	TCP server	
MAC	84-C2-E4-3F-10-AC	
Local IP	192.168.4.101	
Local port	8886	
Web port	80	
MASK	255.255.255.0	
Getway	192.168.4.1	
DNS	114.114.114.114	
Target IP	192.168.4.100	
Target port	8887	

Fuente: elaboración propia

Figura 9 Configuración de red para conectar el Gateway al servidor mosquitto.

Debido a la selección de “TCP server” como modo de funcionamiento, en la parte inferior de la pantalla de configuración surgió un apartado adicional donde se introdujo como IP la dirección del servidor Mosquitto, que es 127.0.0.1. En el campo de puerto se colocó 1883, ya que estos son los parámetros generales para dicho servidor. En cuanto al tópico, se escribió lo siguiente: “prueba/lora/nueva”. Fue de suma importancia reservar esta dirección, ya que con ella se creó el tópico al cual se suscribió el Gateway para que publique en este apartado. Finalmente, se fijaron los parámetros y el software en línea solicitó una contraseña, que por defecto es 123456.

Para concluir el proceso de comunicación entre el Gateway y el servidor, se implementó un cliente MQTT a través de un código en Python 3, teniendo como eje principal la librería “paho.mqtt.client”, que aplica internamente el código de sockets para recibir la información que el Gateway publicó en el tópico “prueba/lora/nueva”, permitiendo visualizarla cuando dicho código se ejecute a través de la consola o de manera autónoma al iniciar la tarjeta Raspberry Pi.

Las consideraciones que se tuvieron al implementar el cliente MQTT fueron:

- La dirección de host fue la correspondiente a la dirección del servidor Mosquitto: '127.0.0.1'.
- El puerto de igual manera tuvo que corresponder con el del servidor Mosquitto y el configurado en el Gateway que fue 18883.
- El tópico al cual se suscribió fue: 'prueba/lora/nueva', que debía corresponder con aquel colocado en la configuración del Gateway.
- El Quos seleccionado fue 2.

Alimentación del sistema de recepción

Se implementó la alimentación de tipo PoE (Power over Ethernet) con el objetivo de extender la conexión y permitir la instalación del Gateway en exteriores, dejando a la tarjeta Raspberry Pi en interiores o en zonas donde pudiera conectarse a la red WiFi para acceder al servidor como cliente. La implementación del PoE no se realizó directamente; en su lugar, se diseñó un circuito con los materiales siguientes:

- Inyector PoE del fabricante TP-Link.

- Inyector PoE de tipo Ubiquiti.
- Módulo convertidor DC-DC LM2596.

Las conexiones se realizaron como se muestra en la Figura 10. El funcionamiento de las conexiones presentadas radica en alimentar al inyector Ubiquiti de corriente alterna en el mismo sitio donde se ubique la tarjeta Raspberry Pi, la cual estará conectada a él mediante Ethernet a través del puerto LAN del inyector. Desde el puerto PoE, se tomó un cable de alimentación de 15 m para el inyector TP-Link, que se encuentra ubicado en el mismo sitio donde se instaló el Gateway con su antena.



Fuente: elaboración propia

Figura 10 Conexión de dispositivos para implementación del PoE.

El inyector TP-Link es el encargado de alimentar al Gateway; no obstante, dicho inyector tiene una salida de 24 V. A pesar de que, según indicaciones del fabricante, el Gateway soporta esa cantidad como máximo, para evitar dañar el sistema se prefirió alimentarlo con un voltaje intermedio dentro del rango de alimentación que soporta el Gateway, que es de 8 a 24 V. Para ello, la salida DC del inyector TP-Link se vinculó a un convertidor LM2596, con el cual se reguló el voltaje a 8.6 V, cantidad que, tras realizar pruebas de alimentación, resultó ser la magnitud mínima aceptable para una correcta operación. Por último, para vincular la conexión Ethernet de la tarjeta Raspberry con el Gateway, el puerto Ethernet del inyector TP-Link se conectó al puerto Ethernet del Gateway.

Implementación del sistema

Para implementar el sistema, en el caso del sistema de transmisión, se le adicionó un soporte reclinable a base de tubo de PVC. Por debajo de este, se fijó

una carcasa de plástico con cobertura de acrílico, donde se instaló internamente el sistema embebido diseñado para la recopilación y transmisión de información, Figura 11. El soporte adicional permite que el sistema de transmisión sea portátil. Por otra parte, el sistema receptor está instalado de manera fija. En la parte del Gateway y su sistema de alimentación, que abarca el inyector TP-Link y el convertidor DC-DC LM2596, se instalaron en un contenedor para exteriores. Se fijaron a través de un retenedor rectangular generado por impresión 3D, con dimensiones coincidentes en largo y ancho al contenedor para exteriores.



Fuente: elaboración propia

Figura 11 Chasis que resguarda el Gateway en conjunto del cableado para ethernet.

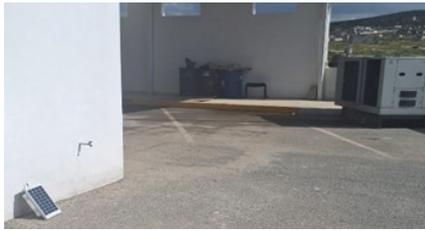
Al chasis se realizaron modificaciones, entre ellas unas perforaciones en la parte inferior para el paso de los cables ethernet-PoE de alimentación del inyector y para la salida del cable de la antena del Gateway, que debía instalarse en el exterior. La instalación de este sistema se realizó en la parte superior del edificio 5 al norte de la Unidad Profesional Interdisciplinaria de Ingeniería Campus Hidalgo (UPIIH) como se muestra en la Figura 12.



Fuente: elaboración propia

Figura 12 Instalación del sistema para el Gateway en el exterior.

Por otra parte, la tarjeta Raspberry Pi y el Inyector Ubiquiti se ubicaron de manera interna en el edificio, teniendo la ventaja que el cable de longitud larga dirigido al puerto PoE del inyector permite montar y desmontar la tarjeta. En cuanto a la instalación del nodo transmisor este se instaló en la zona sur de la UPIIH, para realizar pruebas se colocó a nivel de suelo de manera inclinada para lograr una adecuada incidencia de la luz solar sobre el sistema fotovoltaico, Figura 13.



a) Instalación del nodo transmisor.



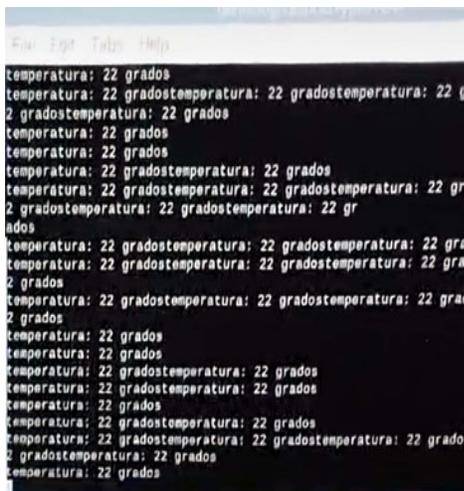
b) Estructura del nodo (sistema embebido).

Fuente: elaboración propia

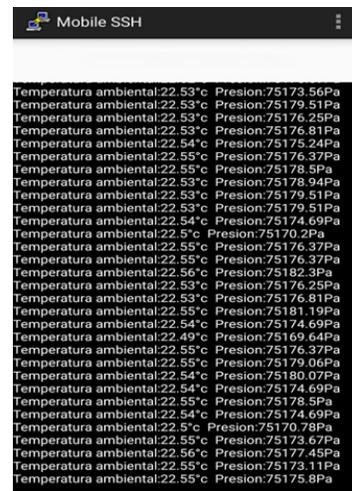
Figura 13 Nodo transmisor.

3. Resultados

Las pruebas de comunicación se realizaron inicialmente con una variable: la temperatura, la cual se visualizó en la consola de la tarjeta Raspberry Pi. Posteriormente, se efectuó una segunda prueba con temperatura y presión, cuyos resultados se visualizaron por acceso remoto, Figura 14.



a) Temperatura en primera prueba.



b) Temperatura y presión en la segunda prueba.

Fuente: elaboración propia

Figura 14 Recepción de información en el servidor.

Ambas pruebas se realizaron a una distancia de 271.55 m, equivalente a la separación entre el edificio 5 de la UPIIH y la zona de estacionamiento ubicada al sur de la institución, donde se ubicó el nodo para las pruebas, realizadas entre las 14:30 y 15:30 horas. Se recibieron mediciones de temperatura y presión de manera eficiente, presentándose retardos tras 15 min de operación, con una duración entre 1 y 15 s. Las mediciones se pueden observar en la Tabla 2. En la Figura 15 se muestra gráficamente las ubicaciones, trayecto y distancia aproximada en las que ocurrió el proceso de transmisión y recepción de información con el sistema desarrollado. En cuanto al ahorro energético, los resultados serán generados a partir de pruebas futuras, donde se pretende incrementar la distancia de prueba y realizar operación continua durante varios días.

Tabla 2 Mediciones de temperatura y presión entre 14:30 y 15:30 horas.

Hora	Temperatura (°C)	Presión (Pa)
14:32	22.14	75,190.31
14:40	22.19	75,198.02
14:46	22.24	75,205.35
14:52	22.33	75,196.22
15:01	22.38	75,201.56
15:12	22.47	75,180.07
15:23	22.53	75,179.3
15:30	22.56	75,182.04

Fuente: elaboración propia



Fuente: elaboración propia

Figura 15 Longitud del trayecto recto de comunicación.

4. Discusión

Como se observa en los resultados, el alcance obtenido en las pruebas más recientes es de 271.55 m aproximadamente. Acorde a [Figueiredo, 2024], en el

trabajo que presentó en ciencia viva en conjunto de la fundação luso americana para o desenvolvimento, se implementaron sistemas para identificar fuentes de contaminación a través de mediciones enviadas con el módulo LoRa E220 a un Gateway para subir la información a la web. El alcance obtenido fue de 400 m entre el colegio Ribadouro y la Rua de Santa Catarina – Porto, Portugal, la cual es un área urbanizada y el trayecto de transmisión posee edificaciones continuamente [Figueiredo, 2024]. Por otra parte, [Amir, 2024], en pruebas realizadas para un proyecto de acciones de conservación de ecosistemas acuáticos realizado con la Universidad Muhammadiyah Parepare, Indonesia, indicó un alcance de 500 m en la transmisión de mediciones de parámetros oceanográficos con retrasos entre 2 y 10 s en transmisión empleando el módulo LoRa E220. Comparando con otros estudios, la distancia obtenida en este trabajo es similar a la de los mencionados. Aunque hubo variaciones de 129 a 229 m en el alcance de transmisión, el máximo registrado fue de 271 m en pruebas realizadas dentro de la institución, limitadas por condiciones geográficas. Se planean nuevas pruebas con el módulo LoRa E220 en un vehículo en movimiento desde la UPIIH hasta la ciudad del conocimiento (1,963.74 m). Estudios en Portugal [Figueiredo, 2024] registraron 400 m en una zona urbana, lo que sugiere que este trabajo podría lograr un mayor alcance en áreas más aisladas. Se espera extender las pruebas hasta la carretera Actopan-Pachuca y el Boulevard del Minero (5,000 a 7,000 m), Figura 16.



a) Entre la UPIIH y Ciudad del conocimiento.



b) Entre UPIIH y carretera Pachuca-Actopan.

Fuente: elaboración propia

Figura 16 Distancias y ubicaciones para pruebas futuras de alcance de transmisión.

5. Conclusiones

El sistema desarrollado permite la recolección de datos como en este caso la presión y la temperatura ambiental a través de una comunicación inalámbrica

utilizando tecnología LoRa, conocida por su largo alcance y bajo consumo energético. Además, integra un sistema de celda solar y baterías que garantizan su autonomía durante todo el día. Los datos recopilados por el Gateway y enviados a un servidor permiten visualizar la información, lo que resulta especialmente beneficioso para áreas remotas o de difícil acceso, por ende el impacto generado de esta metodología radica en haber conseguido una independencia a las redes celulares comerciales para envío de información a larga distancia a través de la integración de la tecnología LoRa y del servidor Eclipse Mosquitto mediante la Raspberry Pi para visualización directa y sin aplicaciones intermediarias.

6. Bibliografía y Referencias

- [1] Amir A. Oceanographic Parameter Measurement System Based on LoRA Communication Module, *JITE (Journal of Informatics and Telecommunication Engineering)*, Vol. 7 No. 2, 2024, Parepare, Indonesia. DOI 10.31289/jite.v7i2.10855.
- [2] Dinesh, G., Gupta, A. K., Nagaseshireddy, M., Prasanna, P. D., Varshini, M. S., & Gowtham, K. LoRa - Powered Smart Agriculture System for Monitoring and Controlling. In *2024 IEEE Wireless Antenna and Microwave Symposium (WAMS)*, 1-6. Visakhapatnam, India, 2024 <https://doi.org/10.1109/WAMS59642.2024.10528095>.
- [3] Edwin, et al. LoRa System with IOT Technology for Smart Agriculture System. In *2022 IEEE 20th Student Conference on Research and Development (SCOReD)*, 39-44. Bangi, Malaysia, 2022. <https://doi.org/10.1109/SCOReD57082.2022.9974084>.
- [4] Figueredo A.,_Projecto Guarda-Rios, *Ciencia viva-fundação luso americana para o desenvolvimento, Junior Atlântico*, 28-30, Porto, Portugal, 2024. <https://webstorage.cienciaviva.pt/public/pt.cienciaviva.io/cms/files/2XLrPM01ZqVbGRc3Cn.pdf>.
- [5] Gaitán, N. C., & Hojbota, P. Forest Fire Detection System using LoRa Technology, *International Journal of Advanced Computer Science and*

- Applications (IJACSA), vol. 11, núm. 5, 2020. DOI: 10.14569/IJACSA.2020.0110503.
- [6] Haxhibeqiri, J., De Poorter, E., Moerman, I., & Hoebeke, J. A Survey of LoRaWAN for IoT: From Technology to Application. *Sensors*, 18(11), 3995, 2018. <https://doi.org/10.3390/s18113995>.
- [7] Ji, M., Yoon, J., Choo, J., Jang, M., & Smith, A. LoRa-based Visual Monitoring Scheme for Agriculture IoT. In 2019 IEEE Sensors Applications Symposium (SAS), 1-6, Sophia Antipolis, France, 2019. <https://doi.org/10.1109/SAS.2019.8706100>.
- [8] Singh, S., & Sharma, A. A comprehensive review on LoRa implementation in smart applications. *AIP Conference Proceedings*, 2516(1), 140001, 2023. <https://pubs.aip.org/aip/acp/article/2516/1/140001/2828038/A-comprehensive-review-on-LoRa-implementation-in>.
- [9] Sulaiman, N., & Abdulmalek, S. A Survey on LoRaWAN Technology: Recent Trends, Opportunities, Simulation Tools and Future Directions. *Electronics*, 11(1), 164, 2022. <https://doi.org/10.3390/electronics11010164>.
- [10] Yang, Y. Design and Application of Intelligent Agriculture Service System with LoRa-based on Wireless Sensor Network. In 2020 International Conference on Computer Engineering and Application (ICCEA) 712-716, Guangzhou, China. 2020. <https://doi.org/10.1109/ICCEA50009.2020.00155>.