



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

PROTOTIPO DE MÓDULO TELEMÉTRICO DE CONSUMO
DE ENERGÍA ELÉCTRICA EN EL SECTOR RESIDENCIAL
DE GUAYAQUIL, BASADO EN IOT.

TRABAJO DE TITULACIÓN

Previa a la obtención del Título de:
MAGISTER EN TELECOMUNICACIONES

Presentado por:

JOSE LENIN MOREJON CAMPOVERDE
CARLOS GABRIEL JUMBO GRANDA

GUAYAQUIL – ECUADOR

AÑO: 2023

AGRADECIMIENTOS

A Dios por darme una bendición más en mi vida, por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad, a mi dupla por la constancia y el empuje, mis hijas e hijos y mis amados padres.

Lenin Morejón

Carlos Jumbo

DEDICATORIA

Mi agradecimiento está dedicado a Dios, por sus infinitas bendiciones al largo de carrera, por haberme dado la sabiduría necesaria para saber terminar con éxito esta maestría, al PhD. Washington Medina y Mgtr. Fernando Vasquez por ser nuestros maestros y guías en este trabajo.

Lenin Morejón

Carlos Jumbo

TRIBUNAL DE EVALUACIÓN

PRESIDENTE

DIRECTOR DEL TRABAJO DE TITULACIÓN

MIEMBRO PRINCIPAL DEL TRIBUNAL

DECLARACIÓN EXPRESA

La responsabilidad y la autoría del contenido de este trabajo de titulación, nos corresponde exclusivamente y damos consentimiento para que la ESPOL, realice la comunicación pública de la obra por cualquier medio con la finalidad de promover la consulta, la difusión y el uso público de la producción intelectual.

Lenin Morejón

Carlos Jumbo

RESUMEN

En el presente trabajo de titulación se desarrolló un módulo telemétrico con la capacidad de medir el consumo de energía eléctrica en la vivienda de los abonados de la ciudad de Guayaquil. Este prototipo permite realizar las funciones de medición, almacenamiento y visualización del consumo de energía eléctrica a través de un dispositivo *Internet Of Things (IoT)*.

Este prototipo consta de un medidor inteligente *MICROSTAR C2000* del cual se extraen datos contenidos en los registros internos de este, a través del puerto de comunicación RS232 haciendo uso del protocolo *DLMS COSEM* para comunicarse con el módulo de comunicación telemétrico *MICROCHIP* modelo RN2903A, empleando la tecnología *LoRaWAN*, para realizar el envío de información hacia la plataforma de servidores de red TTN "*The Thing Network*" la cual es recibida mediante un *gateway* tipo *outdoor* que contiene una antena dipolo y otra GPS para la referenciación del equipo dentro de la red *LoRaWAN*, posteriormente se configurará un *End Device* y se integrarán en una *API*, con el fin de cargar la información en la nube a través de una interfaz web basada en IoT.

El módulo telemétrico permitirá un modo de medición inteligente mediante redes cuyos nodos sean desagregados o independientes, con costos de despliegue bajos y costos de transmisión de información de bajo presupuesto, pudiendo transportar la información desde el medidor del hogar a los sistemas comerciales de las Unidades de Negocios de la Corporación Nacional de Electricidad, que puede ser desplegada por CNEL de una manera rápida y que pueda ir creciendo de manera sostenida y controlada.

Palabras clave: *Prototipo, Módulo telemétrico, Consumo de energía, IoT.*

ABSTRACT

This research paper develops a telemetric module for the measurement of electricity consumption in the residential sector of Guayaquil. This prototype allows the measurement, storage and visualization of electrical energy consumption in real time in an application using Internet Of Things (IoT).

This prototype consists of a smart meter MICROSTAR C2000 from which data contained in the internal records of this, through the communication port RS232 making use of the protocol DLMS COSEM to communicate with the telemetric communication module MICROCHIP model RN2903A, using the technology LoRaWAN, to carry out the transmission of information to the network server platform TTN "The Thing Network" which is received through an outdoor gateway that contains a dipole antenna and another GPS for the referencing of the equipment within the LoRaWAN network, later an End Device will be configured and integrated into an API, in order to upload information to the cloud via an IoT-based web interface.

The telemetry module will allow an intelligent measurement mode through networks whose nodes are disaggregated or independent, with low deployment costs and low-budget information transmission costs, Being able to transport the information from device on the home to commercial systems of the Business Department of National Electricity Corporation, the same one that can be deployed by CNEL in a fast way and that can go growing in a sustained and controlled way.

Keywords: Prototype, Telemetry module, Energy consumption, IoT.

ÍNDICE GENERAL

AGRADECIMIENTOS	iii
DEDICATORIA	iv
TRIBUNAL DE EVALUACIÓN.....	v
DECLARACIÓN EXPRESA.....	vi
RESUMEN.....	vii
ABSTRACT	viii
ÍNDICE GENERAL	ix
ÍNDICE DE FIGURAS.....	xiv
ÍNDICE DE TABLAS	xvi
ABREVIATURAS.....	xvii
GLOSARIO	xviii
INTRODUCCIÓN	21
CAPÍTULO I.....	24
1. PLANTEAMIENTO DEL PROBLEMA.....	24
1.1 Descripción del problema.....	24
1.2 Justificación.....	25
1.3 Objetivos	26
1.3.1 Objetivo general	26

1.3.2	Objetivos específicos:	26
1.4	Elementos diferenciadores o innovadores.....	26
1.5	Resultados esperados	28
1.6	Metodología.....	28
CAPÍTULO 2		29
2.	MARCO TEÓRICO.....	29
2.1	Medidor de energía eléctrica.....	29
2.2	Tipo de medidores	29
2.1.1	Medidores analógicos o electromecánico	29
2.1.2	Medidores digitales.....	30
2.3	Soluciones para lectura en medidores de energía eléctrica.....	30
2.4	Ventajas y desventajas de las soluciones implementadas por CNEC Guayaquil	31
2.5	IoT en la lectura de medidores de energía eléctrica.....	32
2.6.1	Internet de las Cosas (IoT).....	32
2.6.2	Requisitos mínimos necesarios para implementación de <i>IoT</i>	32
2.6.3	The Things Networks.....	34
2.6.4	Tecnología de red de comunicación de baja potencia <i>LoRaWAN</i>	35
2.6.5	Topología <i>LoRaWAN</i>	35
2.6	Marco regulatorio de las frecuencias ISM (ICM) en el Ecuador	38

CAPÍTULO III	40
3. DISEÑO DE MÓDULO TELEMÉTRICO	40
3.1 Módulo transceptor con tecnología de baja consumo y largo alcance (<i>LoRa</i>). 40	
3.1.1 Módulo RN2903 de <i>microchip</i> ®	40
3.1.2 Características.....	41
3.1.3 Rangos de operación.....	41
3.1.4 Funciones RF / analógicas	42
3.2 Medidor inteligente de tres cables de dos fases con conexión directa <i>MICROSTAR</i> C2000	42
3.2.1 Características	43
3.2.2 Sistema eléctrico	43
3.2.3 Opciones de comunicación	44
3.3 Diseño de módulo de comunicación LoRa (Superior).....	44
3.4 Diseño del módulo de comunicación LoRa (inferior)	45
3.5 Vista superior de módulo de comunicación <i>LORA</i>	45
3.6 Vista inferior de módulo de comunicación <i>LORA</i>	46
3.7 Diseño de Firmware para módulo telemétrico con protocolo DLMS COSEM para medidores de energía.....	47
CAPÍTULO 4	48
4. IMPLEMENTACIÓN DEL MÓDULO TELEMÉTRICO	48

4.1	Interconexión del módulo telemétrico con el medidor electrónico.	48
4.2	Configuración del <i>gateway LoRaWAN</i>	49
4.3	Pruebas de comunicación entre <i>gateway</i> y el medidor electrónico	58
CAPÍTULO 5		60
5.	RESULTADOS.....	60
5.1	<i>Comparación de desempeño de quipo CNEI vs ISCONET</i>	61
5.2	<i>Pruebas de comandos remotos</i>	61
5.2.1	Códigos de Transacción.....	62
5.2.2	Códigos de Error	62
5.2.3	Trama de requerimiento de lectura.....	63
5.2.4	Trama de ACK de requerimiento de lectura.....	64
5.2.5	Trama de respuesta de lectura.....	64
5.2.6	Trama de requerimiento de corte	66
5.2.7	Trama de ACK de requerimiento de corte.....	66
5.2.8	Trama de respuesta de corte.....	67
5.2.9	Trama de requerimiento de reconexión.....	68
5.2.10	Trama de respuesta de reconexión.....	68
5.2.11	Trama de requerimiento de apertura.....	70
CAPÍTULO 6		71
6.	CONCLUSIONES Y RECOMENDACIONES	71

6.1	Conclusiones.....	71
6.2	Recomendaciones	73
	REFERENCIAS BIBLIOGRÁFICAS.....	77
	ANEXOS.....	81
	Anexo 1: Firmware de medidor para microcontrolador ATMEGA 2560.....	81

ÍNDICE DE FIGURAS

Figura 2.1 Arquitectura de IoT	34
Figura 2.2. Casos de uso de LoRaWAN.	36
Figura 2.3. LoRaWAN Protocolos y Arquitectura	37
Figura 2.4. Estructura de red LoRaWAN.....	37
Figura 3.1. Transceptor RN2903 Microchip.	40
Figura 3.2. Medidor inteligente Microstar C2000.	42
Figura 3.3. Tarjeta de Circuito Impreso Capa Superior.	44
Figura 3.4. Tarjeta de Circuito Impreso Capa Inferior.	45
Figura 3.5. Vista Superior de módulo de comunicación LORA	45
Figura 3.6. Vista inferior de módulo de comunicación LORA	46
Figura 4.1. Módulo telemétrico.	48
Figura 4.2. Gateway tipo outdoor. Tomada por los autores.	49
Figura 4.3. Gateway tipo outdoor.	50
Figura 4.4. Integración mediante API.....	51
Figura 4.5. Integración mediante API.....	52
Figura 4.6. Integración mediante API.....	52
Figura 4.7. Integración mediante API.....	53
Figura 4.8. Integración mediante API.....	53
Figura 4.9. Integración mediante API.....	54
Figura 4.10. Integración mediante API.....	55
Figura 4.11. Integración mediante API.....	55

Figura 4.12. Integración mediante API.....	56
Figura 4.13. Integración mediante API.....	56
Figura 4.14. Integración mediante API.....	57
Figura 4.15. Integración mediante API.....	57
Figura 4.16. Integración mediante API.....	58
Figura 4.17. Integración mediante API.....	59
Figura 4.18. Integración mediante API.....	59
Figura 29 Lista de medidores instalados.....	60
Figura 30 Interface del Gateway IOT	62
Figura 31 Petición de lectura de medidor.....	63
Figura 32 Trama de estructura de keep alive ACK	64

ÍNDICE DE TABLAS

Tabla 5.1 Trama de requerimiento de lectura.....	63
Tabla 5.2 Trama de ACK de requerimiento de lectura	64
Tabla 5.3. Trama de respuesta de lectura	65
Tabla 5.4. Trama de requerimiento de corte	66
Tabla 5.5. Trama de ACK de requerimiento de corte.....	66
Tabla 5.6. Trama de respuesta de corte.....	67
Tabla 5.7. Trama de requerimiento de reconexión	68
Tabla 5.9. Trama de respuesta de reconexión	69
Tabla 5.10. Trama de requerimiento de apertura.....	70

ABREVIATURAS

AMI RF - *Advanced Metering Infrastructure*

AMR – *Adaptative Multi-Rate*

CNEL – Corporación Nacional Eléctrica

COSEM – *Companion Specification for Energy Metering*

DFU – *Device Firmware Update*

GPS – Sistema de Posicionamiento Global

GSM – Sistema Global de Comunicaciones Móviles

IoT – Internet de las cosas

ISM – *Industrial, Scientific, Medical* “Industrial, Científica, Médico”

kW - Kilovatio

KWh – Kilovatio Hora

Kvar – Kilovoltamperio reactivo

LPWAN – *Low Power Wide Area Networks*

LoRaWAN – Long Range Wide Area Network

LOT – Ley Orgánica de Telecomunicaciones

RF MESH - *Radio Mesh Network*

RF – Radio Frecuencia

RFID – Identificación de Radio Frecuencia

SMT – *Surface-Mount Technology* “Tecnología de montaje superficial”

UART – Transmisor-Receptor Asíncrono Universal

GLOSARIO

AMI RF: Sistema de medición que registra el consumo horariamente o con mayor frecuencia, entregando de manera diaria o en menor tiempo las mediciones hacia una central de recolección a través de una red de comunicación [1].

AMR: AMR significa Lectura Automática de Medidores, que es un proceso de lectura remota totalmente automático a través de Internet. Su núcleo lo conforman los nodos de red ubicados en las escaleras del edificio y que reciben los datos de consumo de los medidores cada cuatro horas [2].

Backend: Los sistemas back-end son sistemas empresariales que se utilizan para ejecutar un sitio web o un negocio, como la gestión de pedidos, el inventario y los sistemas de procesamiento de envíos. Este sistema recoge información de los usuarios o de otros sistemas de tratamiento de datos de la empresa. Responsable del tratamiento de la información proporcionada por los usuarios y recabada por el sitio web [3].

COSEM: El acrónimo "COSEM" significa Companion Specification for Energy Metering. “Es un modelo de interfaz para la comunicación con dispositivos de medición de energía que proporciona una visión general de la funcionalidad disponible a través de una interfaz de comunicación utilizando un enfoque orientado a objetos. El modelo COSEM proporciona un método controlado e independiente del fabricante para identificar, recuperar e interpretar la información de cualquier medidor”. [4].

DLMS: “La especificación de mensaje de idioma del dispositivo (DLMS), o especificación de mensaje para un idioma de dispositivo, se describe en el estándar internacional IEC 62056. Define el modelado estructurado y el intercambio de datos entre instrumentos en un entorno interoperable. DLMS admite aplicaciones como lectura remota de contadores, control remoto y servicios de valor añadido para medir diferentes tipos de energía, como electricidad, agua, gas o calor” [5].

Gateway: Un gateway es un dispositivo, normalmente un ordenador, que “permite conectar redes con diferentes protocolos y arquitecturas en todos los niveles de comunicación. Su propósito es transformar la información del protocolo utilizado en la red al protocolo utilizado en la red de destino” [5].

GSM: “Las siglas GSM corresponden al nombre en inglés del Sistema Global de Comunicaciones Móviles. Ha sido un estándar ampliamente utilizado desde el cambio de siglo y también se conoce como 2G porque es el salto de la comunicación analógica a la digital” [6].

Hand Held: Las computadoras son lo suficientemente pequeñas como para sostenerlas en la mano o llevarlas en el bolsillo. También conocida como computadora de mano. En algunos casos, los datos se pueden ingresar a mano. Otros tienen teclados incorporados [7].

IoT: El término IoT, o Internet de las cosas, se refiere a “la red colectiva de dispositivos y tecnologías conectados que facilitan la comunicación entre los dispositivos y la nube, así como entre los propios dispositivos. Esto significa que los dispositivos cotidianos, como cepillos de dientes, aspiradoras, automóviles y máquinas, pueden usar sensores para recopilar datos y responder de manera inteligente a los usuarios”. [8].

LoRaWAN: LoRa es una tecnología inalámbrica (como WiFi, Bluetooth, LTE, SigFox o Zigbee) que utiliza una forma de modulación de radiofrecuencia patentada por Semtech, un importante fabricante de chips de radio. Esta técnica de modulación, conocida como Chirp Spread Spectrum (o CSS), se ha utilizado en comunicaciones militares y espaciales durante décadas. [9].

LPWAN: La red de área amplia de baja potencia (LPWAN) es una tecnología de comunicación inalámbrica que permite la transferencia de datos entre dispositivos y estaciones base/puertas de enlace ubicadas a cientos de metros o kilómetros de distancia con un consumo de energía extremadamente bajo [10].

MODBUS: “Es un protocolo de comunicación abierto que se utiliza para transferir información entre dispositivos electrónicos a través de una red serial. El dispositivo que solicita la información se denomina maestro Modbus y el dispositivo que proporciona la información se denomina esclavo Modbus”. [11].

RF MESH: Una red de malla inalámbrica, una red conmutada o una red de malla inalámbrica es una red que es una combinación de las dos topologías de red anteriores, infraestructura y ad hoc [12].

TCP/IP: Es el nombre que se utiliza para identificar el grupo de protocolos de red que soportan Internet y”permiten la transferencia de datos entre redes informáticas. En concreto, se puede decir que TCP/IP hace referencia a los dos protocolos más importantes de este grupo: uno denominado Transmission Control Protocol (o TCP)” y el denominado Internet Protocol (referido por las siglas IP) [13].

WAN: Es un acrónimo de Wide Area Network ("WAN"). Este término se utiliza para nombrar redes informáticas que cubren grandes áreas, ya sea que cubran ciudades, países o incluso el mundo entero. Un ejemplo de una WAN es la propia Internet. [14].

ZigBee: “ZigBee está diseñado para proporcionar un alto rendimiento de datos en aplicaciones donde el bajo ciclo de trabajo y el bajo consumo de energía son consideraciones importantes”. (Muchos dispositivos que usan ZigBee funcionan con baterías). Debido a que ZigBee “se usa comúnmente en la automatización industrial y las operaciones de fábricas físicas, a menudo se asocia con la comunicación de máquina a máquina (M2M) y la Internet de las cosas (IoT)” [15].

INTRODUCCIÓN

La energía eléctrica es un recurso imprescindible dentro de la sociedad, su desarrollo y evolución constante depende de ella, por lo tanto, su demanda se ha incrementado considerablemente. Esto ha provocado que la tecnología avanza a pasos agigantados, en las últimas décadas el uso de equipos inalámbricos para el monitoreo de sistemas ha sido vertiginoso, dando así cabida al nacimiento de múltiples tecnologías de comunicación [16].

En Ecuador, CNEL-EP es la empresa encargada de otorgar energía eléctrica al país, y es la responsable de establecer las tarifas por consumo, conforme los kilovatios Hora (kWh) marcados por el medidor, a través de la lectura presencial del medidor por parte de sus empleados. El costo por concepto de consumo de energía eléctrica en los hogares ecuatorianos se vio incrementado por la pandemia de COVID-19, al hacer uso de información histórica para calcular los valores a pagar, generando inconformidad entre los clientes al cuantificar un consumo excesivo, precisando costos elevados en las planillas [17].

Se considera importante optimizar el consumo de energía eléctrica, además de facilitar el trabajo de los inspectores haciendo uso de tecnología de vanguardia para ello, se propone el uso del internet de las cosas (IoT), el cual trata una centralización en una red dispositivos inalámbricos que integran sensores y otras tecnologías que apoyan la comunicación entre ellos y la red de internet consintiendo la transmisión y recepción de datos, a través del uso de medidores inteligentes, nodos y *gateways* que se comunican mediante tecnología *LoRaWAN* “*LoRa Wide Area Network*”, el cual consiste en un protocolo de comunicación bidireccional que usa bajas frecuencias definidas internacionalmente como libres.

Los capítulos dentro de este documento esta organizados de la siguiente forma en el cual se describirán los siguientes contenidos.

Capítulo 1:	Se contempla el planteamiento del problema, su justificación, los objetivos los elementos diferenciadores o innovadores y la metodología a usar.
Capítulo 2:	Se establecen los conceptos sobre los cuales se desarrollará el trabajo de titulación.
Capítulo 3:	Se desarrolla la propuesta mediante el diseño de un módulo de comunicación usando un chip LoRaWAN.
Capítulo 4:	Se detalla la implementación del medidor inteligente y la configuración de la aplicación para su uso.
Capítulo 5:	Se documenta las operaciones realizadas por el medidor puesto en funcionamiento.
Capítulo 6:	Se describe los hallazgos más relevante hechos durante la puesta en marcha del dispositivo y las recomendaciones al momento de su implementación.

CAPÍTULO I

1. PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción del problema

Existe malestar de parte de varios usuarios del sector urbano residencial de Guayaquil por las elevadas facturas que reciben ya que no están acorde a la lectura de sus medidores, por esta razón se presume que CNEL realizó promedios del histórico de los registros de consumo, esto provoco en inconsistencias que se manifiesto en reclamos indicando que no reflejan el consumo real demandado, por lo que muchos usuarios se negaron a cancelar los valores.[18]

El consumo eléctrico actualmente se registra por medio de medidores de energía que están instalados en el predio del usuario, para ello, CNEL puede utilizar medidores electromecánicos como electrónicos o digitales. La lectura del medidor por parte de CNEL se la realiza en forma manual en el caso electromecánicos y para medidores electrónicos sin módulo de telemetría, en el caso de medidores electrónicos con módulo de telemetría se usa tecnología AMI RF (Advanced Metering Infrastructure – Infraestructura de Medición Avanzada) [19]

Las mediciones para los equipos de telemetría se realiza mediante el uso de un equipos portátiles, o una red de comunicación con tecnología RF MESH (Radio Mesh Network – Sistema de malla RF), posterior a esta toma de lecturas de consumo eléctrico esta debe ser ingresada al sistema comercial para la emisión de factura y su posterior cobro, lo cual puede traer errores en la digitación por parte de CNEL, cabe recalcar que este servicio es llevado a cabo por empresas tercerizadas por CNEL.[12]

El problema en la toma de valor medido de forma manual de los equipos de medidores de CNEL Guayaquil se ha acentuado más aún en tiempos de pandemia por el COVID-19, debido a que las empresas encargadas de la recolección de estos datos no podían realizarlo y la decisión de CNEL de utilizar el promedio histórico de registros de consumo, lo que

provocó un alza en la facturación, causando así malestar en sus abonados, retrasos en los cobros, coactivas y cortes de energía.

Cabe mencionar que en el sector industrial existen soluciones de medición telemétrica usando conexiones punto a punto y multipunto, red GSM, Fibra Óptica, las mismas que no son factibles en un sector residencial, dado a su alto costo de despliegue, consumo y mantenimiento con relación a tecnologías IoT.[20]

1.2 Justificación

Para la solucionar el problema descrito, es necesario diseñar para una posterior implementación, un prototipo de comunicación bidireccional telemétrico que permita la lectura del consumo de energía eléctrica generada en los hogares de manera automática y remota, para así automatizar procesos de facturación, recaudación, corte y reconexión.

Una ventaja con una propuesta de esta característica es disminuir la intervención humana en la información de consumo de energía eléctrica de los usuarios, así mismo contar con información actualizada, en línea, de los consumos reales de los abonados.

La propuesta comprende de dotar un módulo de telemetría instalado en el medidor digital de energía eléctrica, aplicando el uso de medidores electrónicos para uso residencial Clase 100, homologado por el departamento de medidores de CNEL Guayaquil, para la transmisión de los datos se usará una red de comunicación de internet de las cosas (IoT) de bajo consumo LPWAN.[21]

La ejecución de este proyecto permitirá a CNEL tener una lectura automática de los datos de cada uno de los medidores electrónicos residenciales en el cantón Guayaquil.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar un dispositivo electrónico con capacidad de telemetría para medir el consumo de energía eléctrica en la ciudad de Guayaquil, basado en IoT.

1.3.2 Objetivos específicos:

- Diseñar el Módulo telemétrico de lectura del consumo eléctrico mediante el uso de tecnología IoT.
- Implementar el módulo telemétrico para lectura del consumo eléctrico.
- Evaluar el desempeño del módulo telemétrico mediante pruebas de lectura en medidor de energía.

1.4 Elementos diferenciadores o innovadores

Tecnología IoT: La implementación de tecnología de Internet de las Cosas (IoT) permite las siguientes fortalezas y capacidades:

- Captura automática de la data referente a los parámetros de energía entregada y consumida, en tiempo real y/o por demanda programada.
- Disminución o eliminación de la intervención de personal técnico en los equipos de medición una vez instalados, y si lo hubiere, poder ser programada y auditada.

Procesamiento de información y auditoría: Procesando la data capturada es posible entre otras optimizaciones las siguientes:

- Modelar y contrastar la demanda por circuitos de alimentadoras, lo que da como resultado en una planificación y dimensionamiento adecuados de las mejoras de la red de servicio.

- Implementar estrategias para corregir las pérdidas no técnicas por conexiones directas o por desbalances de carga u otros efectos sobre la red.
- Implementar procesos de auditoría y calidad de energía en los usuarios que presente alertas de acuerdo con la energía que se suministra y la facturada.

Inteligencia de usuario: La implementación de sistemas inteligentes de medición (equipo de medición y plataforma de procesamiento) permite las siguientes fortalezas con el usuario:

- Permitir acceso al usuario del servicio, a la información que corresponda y sirva para que conozca su perfil de consumo, horas pico, monto consumido al momento de la consulta, entre otras. Con esta funcionalidad se fortalece la confianza por parte del usuario, en el sistema inteligente de medición y en la correcta lectura de su demanda a cancelar.

Inteligencia del sistema de energía: Con la pronta incorporación al diario vivir, de nuevos servicios y usos de la energía eléctrica, la implementación de medición inteligente permite entre otras:

- Manejar adecuadamente la demanda de energía en horas pico, por la carga de vehículos eléctricos que en el corto plazo empieza a implementarse, permitiendo orientar el consumo en las horas valle por medio de tarifas especiales y sistemas que permitan la “Carga Inteligente” para que el usuario se acople a esta estrategia de prioritaria, para una buena operación de las redes de servicio[22].
- Manejar adecuadamente la Demanda de energía por la operación de cocinas de inducción, por los efectos sobre el factor de potencia que afectan a una correcta medición de la energía realmente consumida[4].

1.5 Resultados esperados

Con el desarrollo de este planteamiento esperamos obtener:

- El dispositivo electrónico para la toma de datos de los medidores de energía eléctrica.
- Contrastar la información de consumo de energía eléctrica que llega a la plataforma informática y comparar con la registrada en el medidor de energía.
- Informe del desempeño de consumo de energía eléctrica usando la red *LoRaWAN*.

1.6 Metodología

- Análisis del Escenario, problema y propuesta de solución CNEL Guayaquil en sector urbano residencial.
- Evaluación de la topología de comunicación y tipo de tecnología a utilizar para diseñar el prototipo.
- Método experimental en la construcción del módulo de comunicación para la lectura de medidores de energía.
- Pruebas de prototipo en escenarios críticos.

CAPÍTULO 2

2. MARCO TEÓRICO

2.1 Medidor de energía eléctrica

Un medidor o controlador de consumo eléctrico es un dispositivo electrónico que mide y registra el consumo de energía eléctrica de forma continua. El medidor eléctrico se calibra para medir el consumo de potencia en kilovatios por hora. Un kilovatio por hora es la cantidad de energía dispuesta para suministra 1000 vatios de energía eléctrica en una hora. [23]

2.2 Tipo de medidores

Los medidores eléctricos pueden clasificarse en dos grandes grupos:

- Medidores analógicos;
- Medidores digitales.

2.1.1 Medidores analógicos o electromecánico

Los medidores analógicos son dispositivos electromecánicos que funcionan gracias a la creación de campos magnéticos haciendo uso de 4 bobinas que hacen girar el disco, siendo su velocidad proporcional a la potencia consumida. El consumo de energía eléctrica se registra por hora, su unidad de medida es el kilovatio-horas (KWH), y es mostrado en la parte frontal del medidor a través de un contador electromecánico, el cual se va incrementando conforme el consumo de energía eléctrica.

2.1.2 Medidores digitales

Son dispositivos que registran el consumo eléctrico de forma más detallada, ya que hacen una estimación del consumo realizado indicando el consumo real e individualizado de cada una de las viviendas, mostrados en una pantalla LCD.

Hoy en día también podemos incluir los medidores inteligentes, los cuales son dispositivos que recogen datos del consumo de energía, los encriptan y luego los transmiten al proveedor de energía eléctrica de manera automática y continua a través de comunicación inalámbrica, estos pueden ser instruidos y reconfigurados a distancia de forma remota y casi en tiempo real. [24]

2.3 Soluciones para lectura en medidores de energía eléctrica

Los sistemas de lectura automática de contadores por sus siglas en Ingles, AMR (Automatic Meter Reading), un AMR (medidor automático) remoto en tiempo real.

El sistema de medición remota para el suministro de agua, gas, electricidad se toma como un ejemplo típico en los experimentos. La estructura del sistema emplea una estructura distribuida basada en redes de sensores inalámbricos, que consta de medidores, sensores nodos, recolectores de datos, servidor y red de comunicación inalámbrica. Para una transmisión de corta distancia, el recolector de datos recopila datos de los sensores del medidor utilizando la comunicación RF y ZigBee. Para una transmisión de larga distancia, desde el recolector de datos hasta el servidor, el sistema utiliza una red celular GSM fibra óptica o RF. Los datos del contador se reciben en el servidor a través de WAN mediante el protocolo TCP / IP.[25]

2.4 Ventajas y desventajas de las soluciones implementadas por CNEL Guayaquil

CNEL cuenta con varios sistemas tecnológicos implementados, entre ellas podemos mencionar el de lectura de medidores mediante radio frecuencia (RF), el cual se implementa en ciudadelas cerradas y hace que existan un número considerable de medidores, ya que éste usa tecnología de comunicación *Mesh* o tipo malla, aquí la información viaja mediante saltos entre los diferentes dispositivos de comunicación hasta llegar al Gateway o concentrador de datos y, esta a su vez, tiene un puerto de salida a internet ya sea mediante fibra óptica o un enlace inalámbrico tipo *WiFi* o por red celular para envío de información al data center de CNEL, de por sí esto ya es una desventaja ya que solo es válido para una gran concentración de medidores y no en forma desagregada o distribuida, además de que la comunicación entre medidores en este caso la distancia entre los mismos no debe ser mayor a unos 30 metros, y por otro lado al existir un concentrador para todos los medidores este puede ser una falencia ya que al presentar problemas de comunicación prácticamente todos los medidores quedarían incomunicados [26] [27].

Otro de los problemas que se encuentra es que, en algunos sectores de la ciudad, más en el sector urbano marginal estos sectores son inseguros y muchas veces el personal que realiza la operación de lectura han sido asaltados. Las empresas que realizan los procesos de lectura, corte de servicio eléctrico y reconexión en época de invierno se complicado el acceso debido al mal estado de las calles.

Este tipo de soluciones implementadas por la Corporación Nacional de Electricidad son importadas con costos muy elevados, además de que hay costos de licenciamiento del software que hay que renovar anualmente.

2.5 IoT en la lectura de medidores de energía eléctrica

2.6.1 Internet de las Cosas (IoT)

La frase “Internet de las Cosas” (*IoT*) fue empleado por primera vez en 1999 por el pionero británico Kevin Ashton en el AUTO-ID Center en el Massachusetts *Institute Of Techonolgy* (MIT), para describir una red dispositivos que pueden acceder al internet y transmitir información del estado del ambiente cercano a través de sensores[26].

Este proceso permite conectar elementos físicos que nos rodean en nuestro día a día al Internet: estos pueden ser objetos domésticos existentes en nuestros hogares , como los focos de luz, sensores de medición, dispositivos médicos; abarcando prendas y cosas personales incluyendo los sistemas de las ciudades inteligentes, está tecnología se ha venido utilizando en diferentes ecosistemas industriales, como la agricultura, la salubridad, la fabricación inteligentes [28].

En 2011 Cisco pronosticaba que IoT o también denominado en ese tiempo Internet de las cosas, lo cambiaría todo, y es así como en la actualidad ha revolucionado la industria gracias a las telecomunicaciones, con una alta disponibilidad, adaptabilidad y costo de transmisión [29].

2.6.2 Requisitos mínimos necesarios para implementación de *IoT*

De acuerdo con la UIT Unión Internacional de Telecomunicaciones en 2012 que señala en su escrito los requisitos mínimos necesarios para implementación de un sistema de *IoT* y que debe cumplir con los siguientes aspectos funcionales establece [30]:

1. La Conectividad de los dispositivos conectados debe estar basada en la identificación, ya que para que un objeto pueda establecer conexión es necesario identificarlo.

2. La Compatibilidad que debe existir entre los sistemas conectados, es decir tanto dispositivos que suministran información como los equipos que consumen dicha información.
3. Las Redes deben operar en forma automática, y que está relacionado con las operaciones de gestión, configuración, restablecimiento, optimización y protección con el fin de que puedan adaptarse a diversos escenarios y situaciones.
4. La Configuración de servicios debe ser automática, los mismos que pueden ser configurados por los operadores o los clientes, a partir de la información o datos obtenidos por los dispositivos conectados.
5. Se debe tener la capacidad de ubicación de los dispositivos conectados ya que la información obtenida estará asociados a su ubicación física y de esta forma pueden ser geolocalizados en forma automática.
6. Se debe implementar mecanismos de seguridad, contando con técnicas y políticas de seguridad ya que el Internet de las cosas implica que todo objeto que está conectado está expuesto a posibles amenazas.
7. Se debe contar con protección de la privacidad en todo momento desde la transmisión de la información hasta el procesamiento de esta, ya que dicha información puede ser almacenada en servidores de empresas prestadoras de servicios en la nube, así como también en servidores locales conectados a internet.
8. La red de comunicaciones debe ser administrable para un adecuado mantenimiento y operación.

2.6.3 The Things Networks

Los sistemas de *Backend* que son los programas desarrollados para aplicaciones de internet de las cosas son los responsables de enrutar los datos IoT entre los dispositivos electrónicos y el software de interfaz de usuario.

Comúnmente una red de IoT requiere *puertas de acceso* o pasarelas como enlace entre los diferentes estándares de conectividad al Internet. Puede existir casos en que los propios dispositivos conectados ya soportan la conectividad a internet y tienen implementado todo el *Stack IP*, estos *gateways* sólo deben tener implementadas las funciones de reenvío los datos IP. Otros protocolos diferentes al IP, por ejemplo, *LoRaWAN*, necesita de un enrutamiento de los mensajes y tratamiento de la información antes de que los mensajes puedan ser entregados al usuario. La red se sitúa entre dos puntos de red y la interfaz como se muestra en la **Figura 2.1** se encarga de estos pasos de enrutamiento de la información desde la fuente al destino a través de una canal inalámbrico hacia una central de procesamiento que mostrara la medición de los sensores en tiempo real.

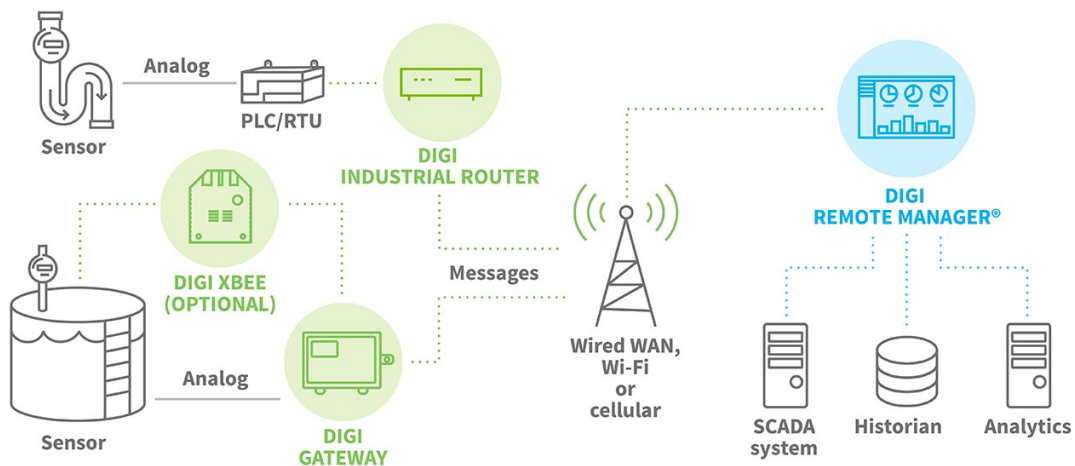


Figura 2.1 Arquitectura de IoT
Fuente: DIGI,2022

2.6.4 Tecnología de red de comunicación de baja potencia *LoRaWAN*

*LoRaWAN*TM es un protocolo de red de gran extensión geográfica, y de baja potencia como lo indican sus Siglas: *Long Range Wide Area Network*, diseñado para redes inalámbricas de sensores de propósito general y las mismas que operan con baterías que lo proveen de autonomía, debido a su bajo consumo de potencia, enviando datos hacia Internet. El protocolo de *LoRaWAN* proporciona herramientas que permiten una conexión bidireccional de bajo ancho de banda para una comunicación de pequeños datos recolectado por los sensores. En la actualidad con el crecimiento de estas redes la seguridad es un punto importante a considerar, dado que esta herramienta controla muchas de las funciones de los equipos que se encuentra dentro de las residencias y oficinas, constituyendo un factor de riesgo alto [31].

2.6.5 Topología *LoRaWAN*

La capa física que se encuentra descrita en el protocolo *LoRa*[®] y *LoRaWAN* están dados para que permitan la interoperabilidad entre dispositivos de diferentes fabricantes tanto de *gateway* como de módulos de comunicación *LoRaWAN* y que los mismos sean equipos con certificación.

Los dispositivos de la red LAN que se encuentra conectados al servidor central por medio de la red TCP/IP o través de una red inalámbrica bajo el estándar IEEE 802.15 para él envío de la información. *LoRaWAN* utiliza una técnica de acceso al medio de espectro extendió UWB para minimizar al máximo el consumo de la batería interna del dispositivo y la necesidad de estar realizando cargas periódicas de los equipos.[32].

Dentro de todas las características de *LoRaWAN*, resaltan las principales para nuestra investigación son las siguientes:

- Topología estrella
- Tres clases de nodos

- Dispositivos administrables
- Configuración de red pública y privada
- Duración de batería hasta 10 años.
- Largo alcance hasta 48km con línea de vista.
- Frecuencias de operación: 915 MHz
- Baja transferencia de datos (hasta 242 bytes)

Los Casos de uso de la tecnología *LoRaWAN*, va desde la logística, monitoreo de ambientes, ciudades inteligentes, Industria IoT, Dispositivos portables, utilitarios, todos aquellos, gracias a su bajo consumo eléctrico y gran cobertura, se basa en la disponibilidad en un ecosistema abierto y estándares abiertos [33].



Figura 2.2. Casos de uso de LoRaWAN.
Fuente: CatSensor,2022

LoRaWAN está siendo desarrollada por *LoRa Alliance*, que contiene un ecosistema abierto y en gran crecimiento por varias entidades de desarrollo y fábricas de chips y proveedores en la nube [34].

En las **Figura 2.3** y **Figura 2.4**, se muestra estructura de la red *LoRaWAN*, tanto los protocolos de comunicación y periféricos permitidos, esto permite que las características de *LPWAN* se mantengan para operar en bandas ISM (ICM). *LoRa Alliance* se destaca por la labor de agrupar todos los elementos de la tecnología para estar acorde al presente y futuro.

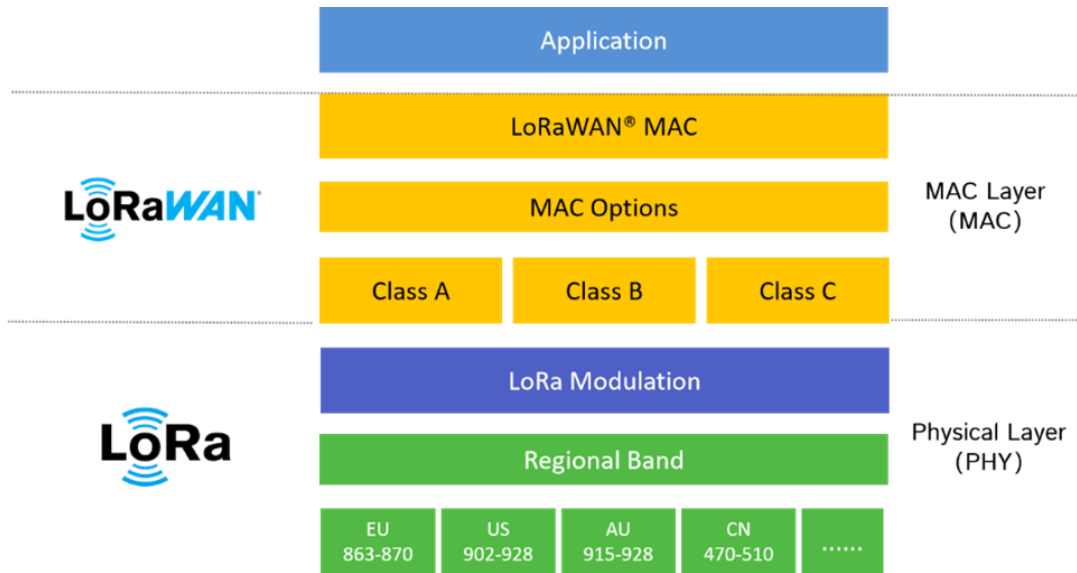


Figura 2.3. LoRaWAN Protocolos y Arquitectura
Fuente: Vencuel, 2022

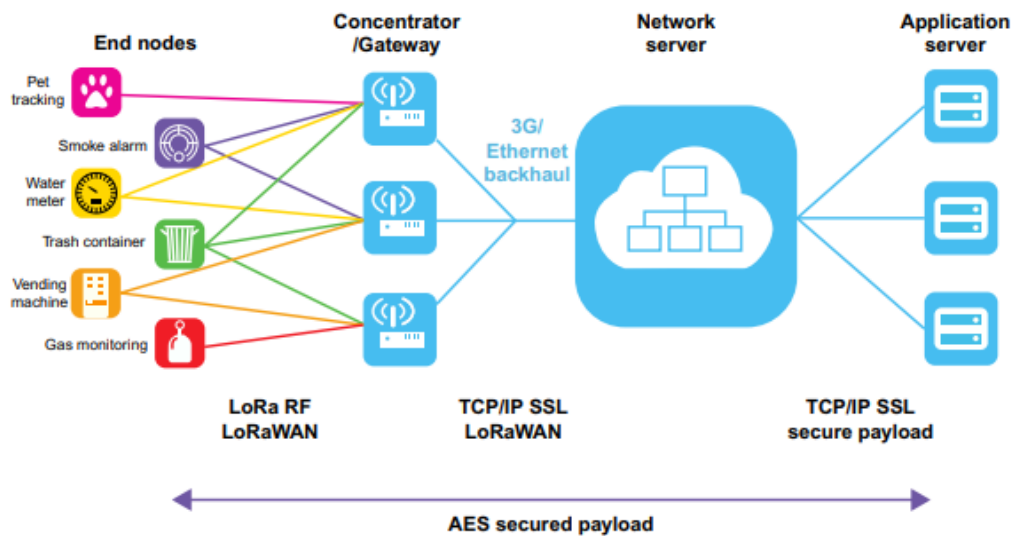


Figura 2.4. Estructura de red LoRaWAN.
Fuente: Medium, 2022

2.6 Marco regulatorio de las frecuencias ISM (ICM) en el Ecuador

En el Ecuador las telecomunicaciones son consideradas como un sector estratégico y el uso de frecuencias esta normado con la Ley Orgánica de Telecomunicaciones, y el uso de bandas libres (ISM) que son destinadas para el uso experimental del espectro electromagnético y de uso médico [35].

Así mismo IOT indica:

“Que, la Constitución de la República en su artículo 16, consagra el derecho de todas las personas en forma individual o colectiva al acceso en igualdad de condiciones al uso de las frecuencias del espectro radioeléctrico para la gestión de estaciones de radio y televisión públicas, privadas y comunitarias, y a bandas libres para la explotación de redes inalámbricas” [36].

De esta manera permite el uso de bandas libre para su explotación con fines públicos. También señala la LOT sobre el uso de bandas libres lo siguiente:

“Que, según lo consagrado en el artículo 17 de la misma Carta Magna, el Estado fomentará la pluralidad y la diversidad en la comunicación, y al efecto, garantizará la asignación, a través de métodos transparentes y en igualdad de condiciones, de las frecuencias del espectro radioeléctrico, para la gestión de estaciones de radio y televisión públicas, privadas y comunitarias, así como el acceso a bandas libres para la explotación de redes inalámbricas, y precautelaré que en su utilización prevalezca el interés colectivo” [36].

En base a esto, según el estándar técnico vigente para IOT, se señala a las frecuencias de uso libre y sus potencias, en las cuales constan las bandas 915Mhz a 928Mhz utilizadas por la tecnología LoRaWAN usadas en el presente diseño de prototipo de medición, se encuentran debidamente habilitadas en el Ecuador, para su uso, bajo el termino de bandas

libres. Para el uso y explotación de servicio con fines industriales, científicos y medicina (ISM) en el Ecuador está permitido cumpliendo criterios de potencia [37].

CAPÍTULO III

3. DISEÑO DE MÓDULO TELEMÉTRICO

En el presente capítulo se diseña el hardware del módulo de comunicación con el Chip modem *LoRaWAN* y *Software* del prototipo de módulo telemétrico de medición de consumo eléctrico, el registro de la información será cargada en la nube a través de una interfaz web.

Hardware

Ordenador portátil

- Modelo: ASUS K555U;
- Procesador: Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz 2.59 GHz;
- Memoria: 8GB RAM;
- Sistema Operativo: 64 bits Windows 10.

3.1 Módulo transceptor con tecnología de baja consumo y largo alcance (*LoRa*)

3.1.1 Módulo RN2903 de *microchip*®

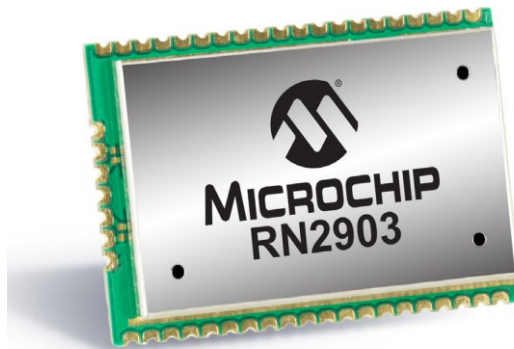


Figura 3.1. Transceptor RN2903 Microchip.
Fuente: Microchip Technology, 2022

El módulo transceptor RN2903 de tecnología *LoRa* de *Microchip* se muestra en la **Figura 3.1** proporciona una solución de monitoreo de alta densidad en una red con extensión geográfica de grandes dimensiones.

El dispositivo RN2903 cumple con las especificaciones del protocolo *LoRaWAN* Clase A. Integra la sección de RF, un controlador de banda base, un procesador de interfaz de programación de aplicaciones (API) con líneas de comando, lo que lo convierte en una solución completa para aplicaciones inalámbricas de largo alcance.[38]

3.1.2 Características

- Pila de protocolo *LoRaWAN*™ integrada;
- Comunicación UART ;
- Dimensiones: 17,8 x 26,7 x 3,34 mm;
- Pads SMT diseñados para montaje en tarjeta de circuito impreso en forma sencilla para un montaje seguro y confiable;
- Cumple con normas *RoHS* para cumplir con protección al medio ambiente;
- Cumplimiento:
 - Certificado modular para los Estados Unidos (FCC) y Canadá (IC)
 - Taiwán
- Actualización de firmware del dispositivo (DFU) a través de UART, consulte “RN2903 *LoRa*® Guía del usuario de referencia de comandos del módulo tecnológico” (DS40001811) [39].

3.1.3 Rangos de operación

- Voltaje de funcionamiento: 2,1 a 3,6 [V];
- Límites de temperatura: -40 ° C a + 85 ° C;
- Eficiencia energética;
- Tasa de bits de comunicación de RF programable hasta 300 kbps con modulación FSK, 12500 bps con tecnología Modulación *LoRa*®;
- MCU integrado, cristal, *EEPROM* número de serie y de nodo EUI-64, módulo de comunicación analógica, y acoplamiento de alta impedancia.[40]

3.1.4 Funciones RF / analógicas

- Transceptor de largo alcance de baja potencia que funciona en la banda de frecuencia de 915 MHz;
- Alta sensibilidad del receptor: hasta -146 dBm;
- Potencia TX: ajustable hasta +18,5 dBm *Power Amplifier* de alta eficiencia;
- Modulación de tecnología *FSK*, *GFSK* y *LoRa*;
- IIP3 = -11 dBm;
- Cobertura de hasta 15 km en área suburbana y cobertura de hasta 5 km en área urbana;
- 14 GPIO (entradas y salidas de propósito General) para control y estado, compartidos con 13 Entradas analógicas Funciones RF / analógicas. [41]

3.2 Medidor inteligente de tres cables de dos fases con conexión directa **MICROSTAR C2000**



Figura 3.2. Medidor inteligente Microstar C2000.
Fuente: Microstar, 2022

Es un equipo de conexión directa, residencial o comercial, de 3 cables de 2 fase.

3.2.1 Características

- Medición precisa de kWh y kvarh;
- Registro de demanda máxima de kW y kvar;
- Instrumentación de voltaje, corriente, potencia y factor de potencia
- Control de tiempo de uso (TOU) con 4 tarifas y programación flexible;
- Registros de eventos con hasta 200 eventos de manipulación o calidad de energía;
- Pantalla LCD de 8 dígitos de alta calidad con caracteres grandes y luz de fondo;
- Reloj interno en tiempo real (RTC) con supercondensador y batería de respaldo;
- Admite protocolos de comunicación DLMS, IEC 62056-21 o MODBUS;
- Diseño de carcasa de medidor de tipo encapsulado para alta confiabilidad y seguridad;
- Pantalla de energía activa en LCD sin fuente de alimentación.

3.2.2 Sistema eléctrico

- | | |
|-------------------------|------------------------------|
| • Tipo de conexión | Conexión directa a la línea. |
| • Modos de medición | 2 fases 3 hilos |
| • Clase de precisión | Clase 1.0, Clase 0.5 |
| • Voltaje nominal | 110 V, 220 V, 230 V o 240 V |
| • Corriente nominal | 5 (80) A, 10 (100) A |
| • Corriente de arranque | <0,1% I |
| • Frecuencia nominal | 50 o 60 Hz |
| • El consumo de energía | |
| • Voltaje de circuito | <1W, <2.5VA por fase |
| • Corriente de circuito | <0.1 VA |

3.2.3 Opciones de comunicación

- Comunicación local Óptica / Infrarrojos, RS232, RS485,
- PLC, RF (433 / 866MHz)
- Comunicación remota GPRS / GSM, Wi-Fi, Ethernet
- Protocolo de comunicación DLMS / COSEM, IEC62056-21, MODBUS [42]

3.3 Diseño de módulo de comunicación LoRa (Superior)

La **Figura 3.3** se observa el diseño final de la vista superior de la tarjeta de Circuito Impreso (PCB) desarrollada con Software de diseño electrónico de ruteo de pistas para tarjetas de circuito impreso de *Eagle Autodesk*.

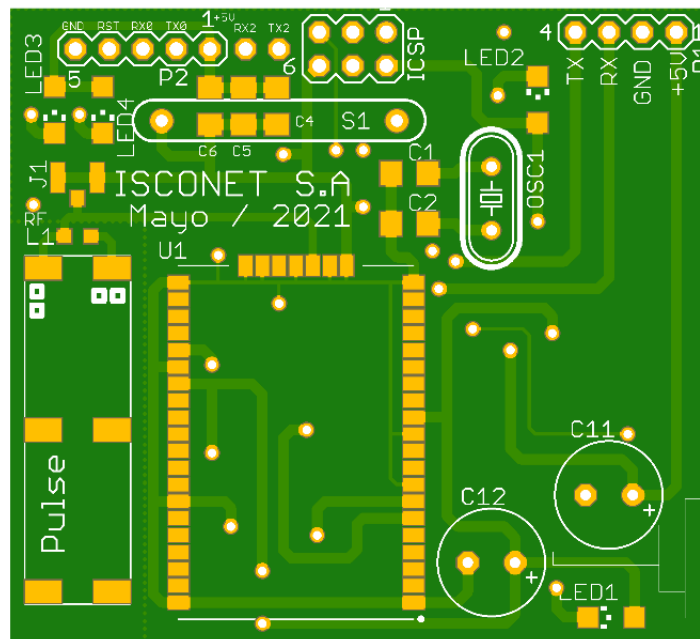


Figura 3.3. Tarjeta de Circuito Impreso Capa Superior.

Fuente: ISCONET S.A, 2022

3.4 Diseño del módulo de comunicación LoRa (inferior)

La **Figura 3.4** se observa el diseño final de la vista inferior de la tarjeta de circuito impreso (PCB) desarrollada con *software* de diseño electrónico de ruteo de pistas para tarjetas de circuito impreso de *Eagle Autodesk*.

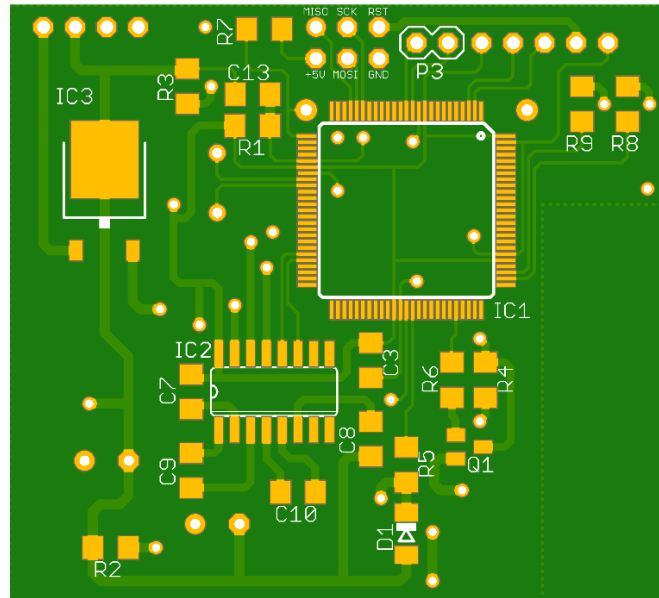


Figura 3.4. Tarjeta de Circuito Impreso Capa Inferior.

Fuente: ISCONET S.A, 2022

3.5 Vista superior de módulo de comunicación LORA.

La **Figura 3.5** muestra tarjeta de circuito impreso en modelo 3D capa superior.

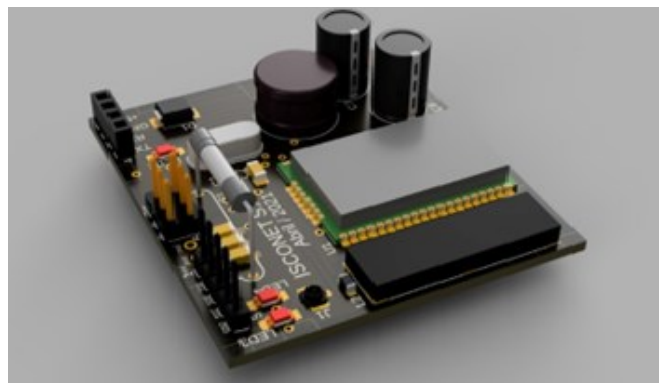


Figura 3.5. Vista Superior de módulo de comunicación LORA

Fuente: ISCONET S.A., 2022

3.6 Vista inferior de módulo de comunicación *LORA*.

La **Figura 3.6** se observa la vista inferior de la tarjeta de circuito impreso en modelo 3D.

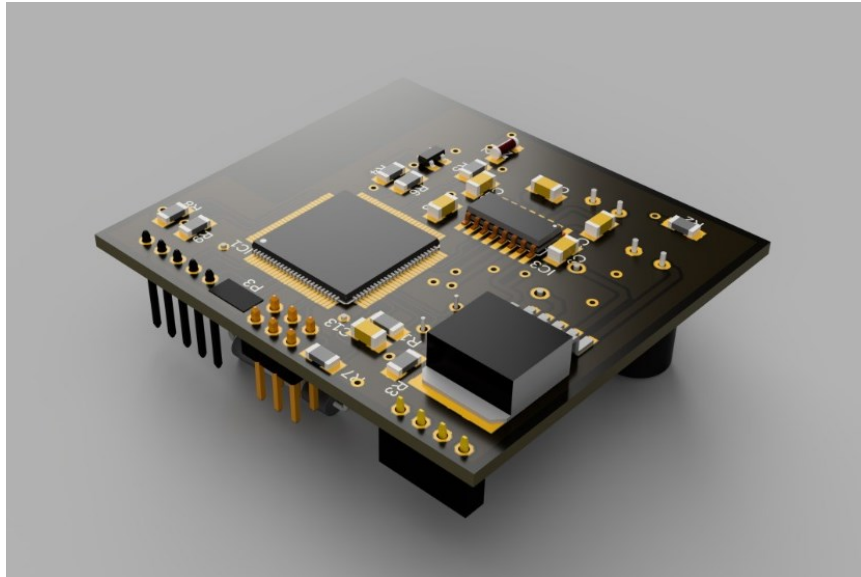


Figura 3.6. Vista inferior de módulo de comunicación LORA

Fuente: ISCONET S.A., 2022

El diseño comprende el módulo electrónico de telemetría a ser instalado como un dispositivo adicional al medidor de energía marca *Microstar* de 2 fases 3 hilos (Línea Neutro Línea), que cuenta con un puerto de comunicación RS232, adicional al puerto óptico y que como protocolo estándar usa DLMS/COSEM el cual define el estándar para comunicación con medidores inteligentes, entendiéndose como medidores inteligentes de cualquier tipo de energía sea esta eléctrica, gas, agua o calor, sus principales aplicaciones están enfocadas para lectura remota y control en nuestro prototipo es para control de corte y reconexión del suministro de energía eléctrica de forma remota.

El módulo de comunicaciones está basado en un módulo de comunicación de la firma *MICROCHIP* modelo RN2903A y que implementa la Comunicación *LoRaWAN* el mismo que está en su diseño electrónico operado por el micro controlador *ATMEGA 2560* el cual contiene embebido el firmware y que implementa el protocolo del medidor DLMS para

obtener la información de los parámetros eléctricos que están en los registros internos del medidor y que luego será enviado por la interface de red inalámbrica con el protocolo *LoRaWAN* hacia la plataforma de Servidores de Red de “*The Thing Industries*”.

3.7 Diseño de Firmware para módulo telemétrico con protocolo DLMS COSEM para medidores de energía

El diseño del *firmware* permite implementar el protocolo DLMS, a fin de que el módulo pueda conversar con el microcontrolador y con el medidor para así extraer la información de los registros internos, y *LoRa* lo utiliza como una interfaz de red para utilizar los datos. El *firmware* que se encuentra cargado sobre el microcontrolador ATMEGA 2560. [43]

CAPÍTULO 4

4. IMPLEMENTACIÓN DEL MÓDULO TELEMÉTRICO

4.1 Interconexión del módulo telemétrico con el medidor electrónico.

Para realizar la interconexión del módulo de telemetría con el medidor se usa un puerto de comunicación RS232 que trae implementado el medidor, con el cual se va a extraer la información de los parámetros eléctricos desde los registros internos del medidor usando el protocolo *DLMS COSEM* el mismo que esta implementado en *firmware* en el microcontrolador del módulo de comunicación y que luego es enviada la información usando la tecnología *LoRaWAN* mediante el módulo de radio que está contenido en la tarjeta de telemetría, tal como muestra la **Figura 4.1** el módulo de comunicación con su tarjeta electrónica y el módulo de comunicación *LoRaWAN* de *Microchip* RN2903A.

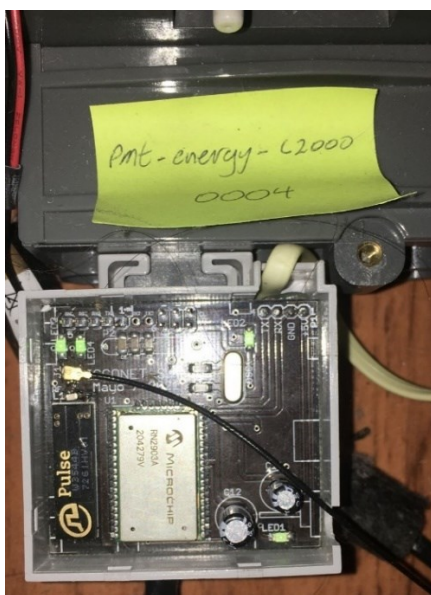


Figura 4.1. Módulo telemétrico.

Fuente: ISCONET S.A.

4.2 Configuración del gateway *LoRaWAN*

Para la interconexión del medidor con el módulo de telemetría *LoRaWAN* a la red TTN (*The Things Network*) se la realiza mediante un gateway tipo *outdoor* que fue instalado en un pequeño mástil en la terraza de una vivienda en las calles Letamendi y la 13 en la ciudad de Guayaquil como se muestra en la **Figura 4.2**, dicho gateway tiene conexión a internet mediante cable *ethernet* al *router* del proveedor de servicio de la residencia.

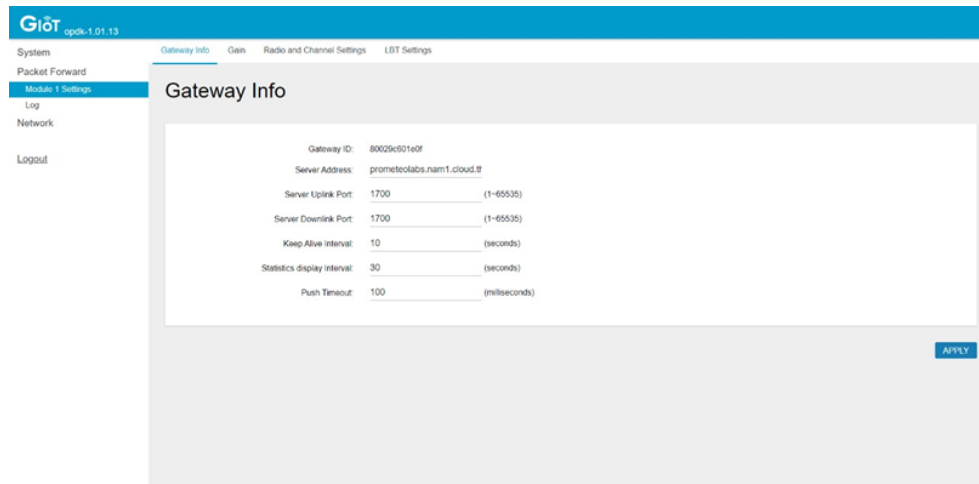


Figura 4.2. Gateway tipo outdoor. Tomada por los autores.

Fuente: ISCONET S.A., 2022

El *gateway* tipo *outdoor* contiene una antena dipolo y también una antena GPS la misma que sirve para realizar la geo referenciación del equipo en la red *LoRaWAN*, con la cual se puede realizar tareas de cobertura georreferenciada de punto de acceso LoRa descrito en la **Figura 4.3**.



Figura 4.3. Gateway tipo outdoor.

Fuente: ISCONET S.A., 2022

Para la correcta comunicación entre el medidor y la red *LoRaWAN* mediante el *gateway* se debe configurar dicho equipo en la red TTN para crear el dispositivo en la red y que sea reconocido en la misma, además luego hay que crear el equipo final como un *end device* y realizar la integración mediante un API, dicho procedimiento se describe a continuación, la **Figura 4.4** muestra la creación de 2 *gateway* en la plataforma LNS (*Local Network Servers*) en *The Things Industries* que será el portal web a donde se enviara la información del medidor de prueba crear *gateway*.

Paso 1: Dar clic en “Add Gateway”

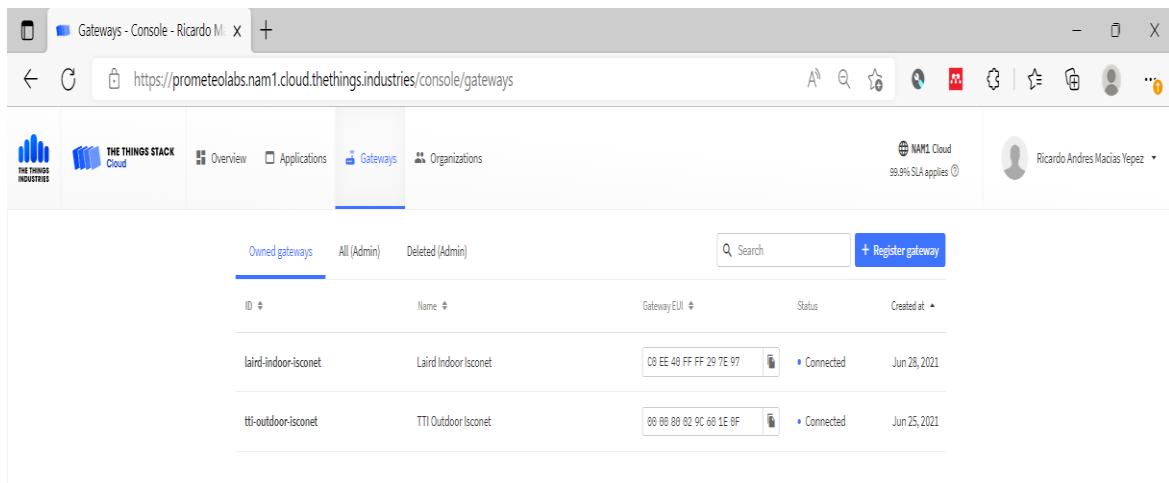


Figura 4.4. Integración mediante API.

Fuente: ISCONET S.A., 2022

Paso 2: Llenar los campos (el *gateway* que lo provee el fabricante del equipo)

Add gateway

General settings

Owner*
nicardomaciay

Gateway ID*
ttt-outdoor-isconet

Gateway EUI
00 00 00 02 9C 60 1E 0F

Gateway name
TTT Outdoor Isconet

Gateway description
Description for my new gateway
Optional gateway description; can also be used to save notes about the gateway

Gateway Server address
prometeolabs.nam2.cloud.things.industries
The address of the Gateway Server to connect to

Require authenticated connection
 Enabled
Controls whether this gateway may only connect if it uses an authenticated Basic Station or MQTT connection

Gateway status
 Public
The status of this gateway may be visible to other users

Figura 4.5. Integración mediante API.

Fuente: ISCONET S.A., 2022

The status of this gateway may be visible to other users

Gateway location
 Public
The location of this gateway may be visible to other users and on public gateway maps

Attributes
[+ Add attributes](#)
Attributes can be used to set arbitrary information about the entity, to be used by scripts, or simply for your own organization

LoRaWAN options

Frequency plan*
United States 902-928 MHz, FSB 2 (used by TTN)

Schedule downlink late
 Enabled
Enable server-side buffer of downlink messages

Enforce duty cycle
 Enabled
Recommended for all gateways in order to respect spectrum regulations

Schedule any time delay*
530 milliseconds
Configure gateway delay (minimum: 130ms, default: 530ms)

Gateway updates

Automatic updates
 Enabled
Gateway can be updated automatically

Channel
Stable
Channel for gateway automatic updates

[Create gateway](#)

ngs Industries

Figura 4.6. Integración mediante API.

Fuente: ISCONET S.A., 2022

Paso 3: "Create gateway"

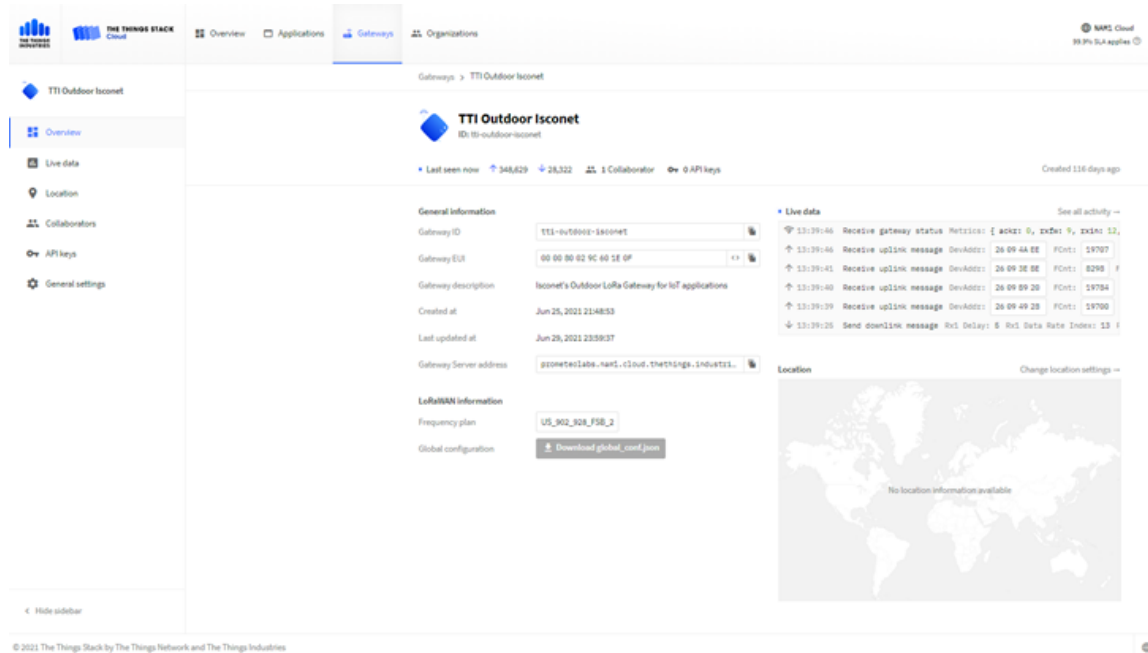


Figura 4.7. Integración mediante API.

Fuente: ISCONET S.A., 2022

Paso 4: Crear *Application* y *End Device*

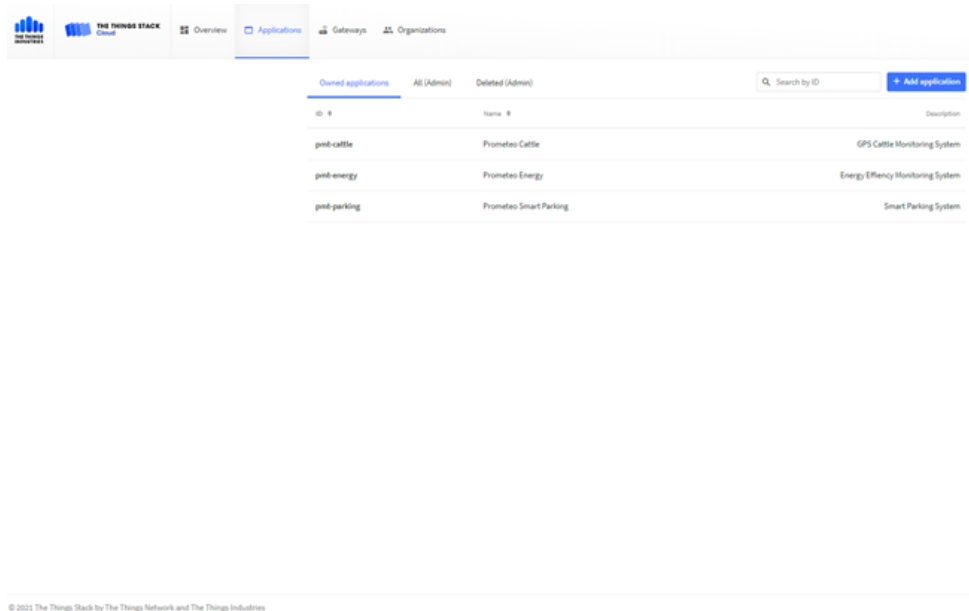


Figura 4.8. Integración mediante API.

Fuente: ISCONET S.A., 2022

Paso 5: Creación de aplicación en la red TTN.

The screenshot shows the 'Add application' form in the TTN console. The form includes the following fields:

- Owner:** A dropdown menu with 'org:isconet-002' selected.
- Application ID:** A text input field containing 'pmt-energy'.
- Application name:** A text input field containing 'Prometeo Energy'.
- Description:** A text area containing 'energy efficient monitoring system'.

Below the form is a blue button labeled 'Create application'.

Figura 4.9. Integración mediante API.

Fuente: ISCONET S.A., 2022

Paso 6: En “*Create Application*” se muestra el panel de control de la aplicación, posteriormente damos clic en “*End Devices*”

The screenshot shows the application control panel for 'Prometeo Energy' in the TTN console. The panel includes the following elements:

- Overview:** A sidebar menu with options like 'End devices', 'Live data', 'Payload formatters', 'Integrations', 'Collaborators', 'API keys', and 'General settings'.
- General Information:** A section showing application details such as 'Application ID', 'Created at', and 'Last updated at'.
- Live data:** A section showing real-time data messages with timestamps and content.
- End devices (3):** A table listing end devices with columns for ID, Name, DevEUI, AppEUI, and Last seen.

ID	Name	DevEUI	AppEUI	Last seen
pmt-energy-c2000-0000	c2000-device-0000	00 04 A3 05 00 F9 28 30	00 00 00 00 00 00 00	9 seconds ago
pmt-energy-c2000-0001	c2000-device-0001	00 04 A3 05 00 F9 09 61	00 00 00 00 00 00 00	15 seconds ago
pmt-energy-c2000-0002	c2000-device-0002	00 04 A3 05 00 F9 75 4E	00 00 00 00 00 00 00	11 seconds ago
pmt-energy-c2000-0003	c2000-device-0003	00 04 A3 05 00 F9 45 FC	00 00 00 00 00 00 00	6 seconds ago
pmt-energy-c2000-0004	c2000-device-0004	00 04 A3 05 00 F9 4F 9E	00 00 00 00 00 00 00	1 second ago

Figura 4.10. Integración mediante API.

Fuente: ISCONET S.A., 2022

Paso 7: Creando el equipo Final (medidor con módulo de comunicación LoRa)

“Add End Device”

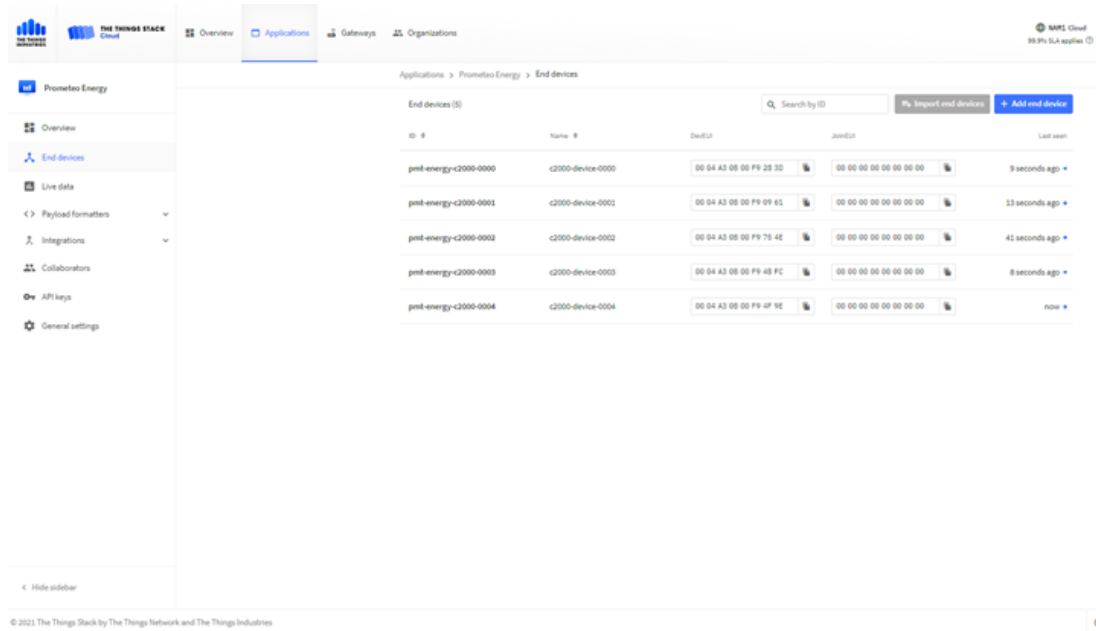


Figura 4.11. Integración mediante API.

Fuente: ISCONET S.A., 2022

Paso 8: Se llena con información los campos en esta sección “Create End Device”

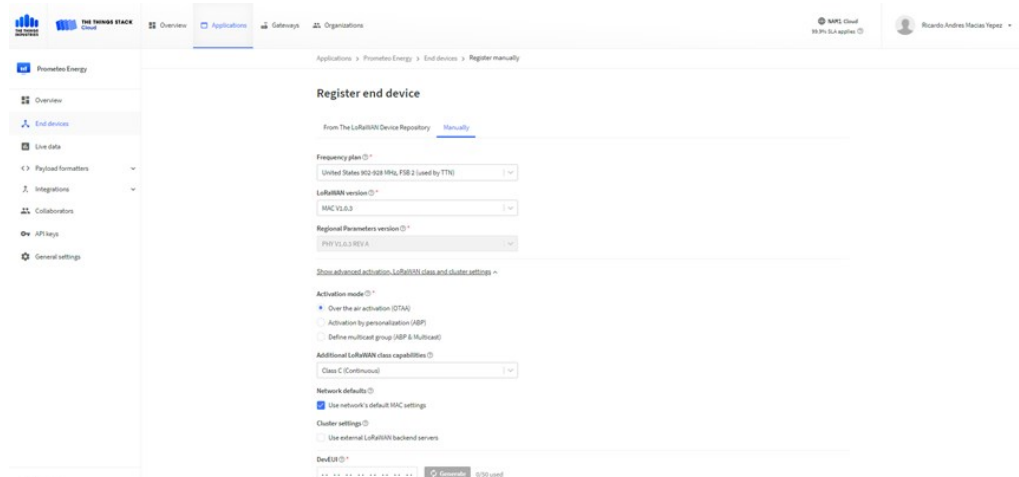




Figura 4.12. Integración mediante API.

Fuente: ISCONET S.A., 2022

Paso 9: Una vez finalizado este proceso en “*Create End Device*” se muestra el panel de control del *End Device*.

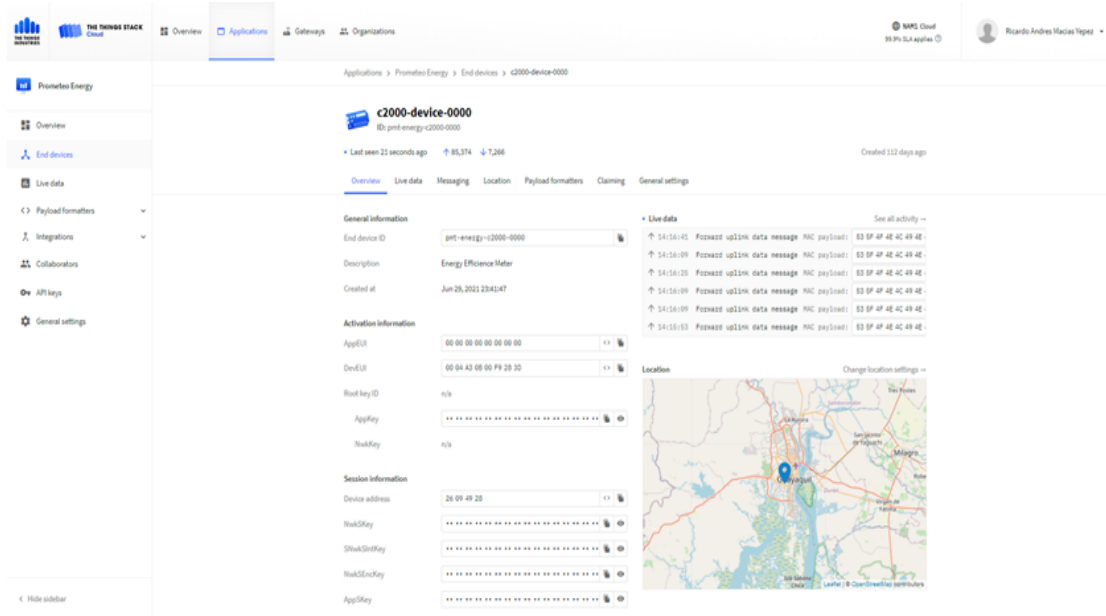


Figura 4.13. Integración mediante API

Fuente: ISCONET S.A., 2022

Para que los datos provenientes del medidor en puedan ser almacenados en los Servidores de la Red *LoRaWAN* hay que realizar la integración mediante un API en la sección “*Integrations*” y luego en “*Webhooks*”.

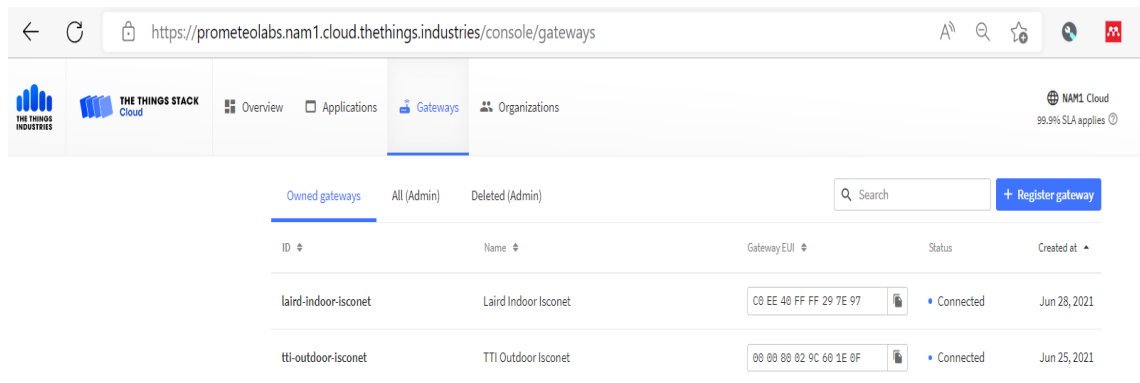


Figura 4.14. Integración mediante API.

Fuente: ISCONET S.A., 2022

Seleccionamos en nuestro caso el API “*custom webhook*”

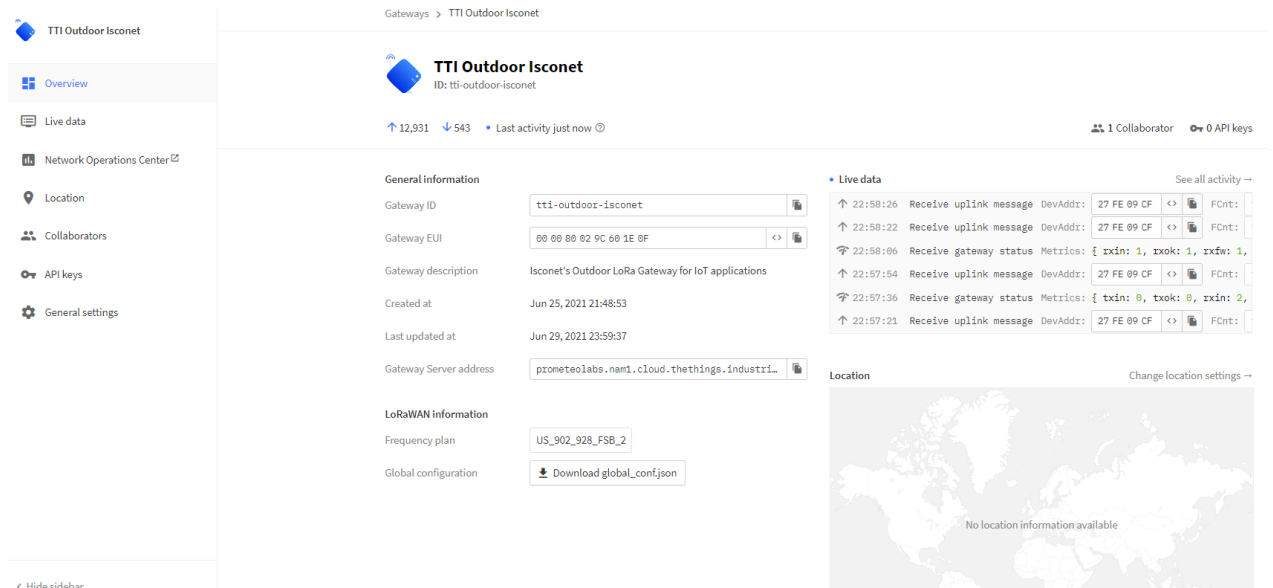


Figura 4.15. Integración mediante API.

Fuente: ISCONET S.A., 2022

Llenamos todos los campos y guardamos la configuración en “*save changes*”

The screenshot shows the 'General settings' page for a gateway in The Things Stack Cloud. The left sidebar contains navigation options: Overview, Live data, Network Operations Center, Location, Collaborators, API keys, and General settings (selected). The main content area is titled 'Basic settings' and includes a 'Collapse' button. The fields are as follows:

- Gateway ID**: tti-outdoor-isconet
- Gateway EUI**: 00 00 00 02 9C 60 1E 0F
- Gateway name**: TTI Outdoor Isconet
- Gateway description**: Isconet's Outdoor LoRa Gateway for IoT applications
- Gateway Server address**: prometeolabs.nam1.cloud.thethings.industries
- Require authenticated connection**: Enabled

Figura 4.16. Integración mediante API.

Fuente: ISCONET S.A., 2022

4.3 Pruebas de comunicación entre *gateway* y el medidor electrónico

Una vez configurado el *end device*, el *gateway* y el API de integración en los servidores de la red *LoRaWAN* ponemos en operación el sistema y podemos observar los datos del medidor en formato de sistema numérico hexadecimal y que son enviados en forma automática a la plataforma informática de *The Things Network* con un intervalo de envío de datos que es parametrizable, o que también mediante un aplicativo puede ser adquirido bajo demanda.

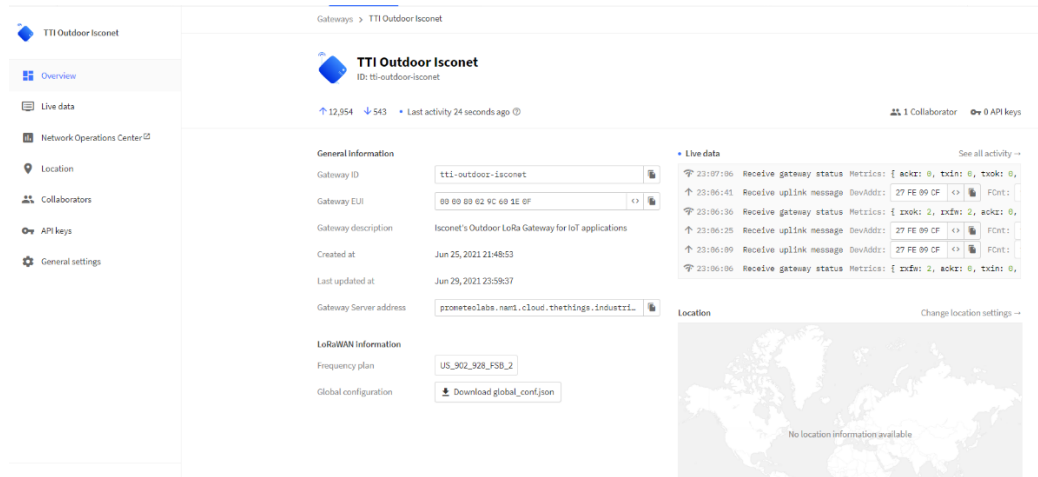


Figura 4.17. Integración mediante API.

Fuente: ISCONET S.A., 2022

Datos de los medidores en tiempo real.

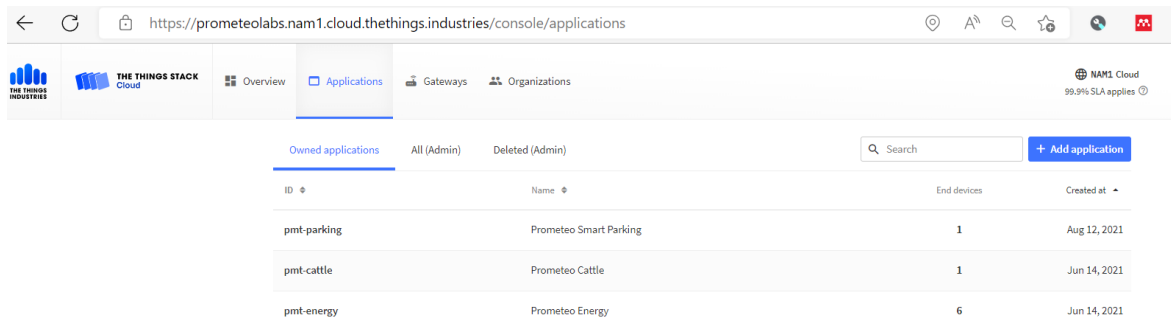


Figura 4.18. Integración mediante API.

Fuente: ISCONET S.A., 2022

CAPÍTULO 5

5. RESULTADOS

El prototipo del módulo telemétrico de facturación de energía eléctrica fue diseñado usando el concepto de internet de las cosas, tal como se lo define en el **CAPÍTULO III** se diseñó del módulo telemétrico. Para su desarrollo se usó un medidor inteligente MICROSTAR C2000, el cual es el encargado de registrar el consumo eléctrico y de donde se extraen los registros a través del puerto de comunicación RS232 usando el protocolo DLMS COSEM, para la comunicación con el MICROCHIP modelo RN2903A, empleando la tecnología de comunicación telemétrica LoRaWAN, resolviendo el problema de monitoreo en tiempo real a bajo costo y con baja potencia, posteriormente la información se envía a la plataforma de servidores de red TTN “The Thing Network”, aquí es recibida mediante un gateway tipo outdoor que contiene una antena dipolo y otra GPS, la cual ayuda a la referenciación del equipo dentro de la red LoRaWAN, posteriormente se configurará un End Device y se integrarán en una API, con el fin de cargar la información en la nube a través de una interfaz web basada en IoT. Así mismo, en el presente documento se demuestra que, mediante la decodificación de una trama que envía el medidor a la plataforma de “The Thing Industries”, se pueden obtener los diferentes parámetros eléctricos leídos en el medidor en forma remota.

ID	Name	DevEUI	JoinEUI	Last activity
my-lora-meter		E6 68 84 51 53 59 82 24	24 B2 59 53 51 84 68 E6	18 sec. ago
pmt-energy-c2000-0000	c2000-device-0000	00 04 A3 08 08 F9 28 3D	00 00 00 00 00 00 00 00	6 sec. ago
pmt-energy-c2000-0004	c2000-device-0004	00 04 A3 08 08 F9 4F 9E	00 00 00 00 00 00 00 00	15 sec. ago
pmt-energy-c2000-0003	c2000-device-0003	00 04 A3 08 08 F9 48 FC	00 00 00 00 00 00 00 00	6 sec. ago
pmt-energy-c2000-0002	c2000-device-0002	00 04 A3 08 08 F9 75 4E	00 00 00 00 00 00 00 00	Never
pmt-energy-c2000-0001	c2000-device-0001	00 04 A3 08 08 F9 89 01	00 00 00 00 00 00 00 00	May 3, 2022

Figura 19 Lista de medidores instalados

Fuente: ISCONET S.A, 2022

5.1 Comparación de desempeño de equipo CNEL vs ISCONET

El prototipo del módulo telemétrico de facturación de energía eléctrica, como se aprecia en el CAPÍTULO 4 implementación fue instalado en una vivienda ubicada en el sector sur oeste en la Ciudad de Guayaquil en las calles Letamendi y la 13. La implementación del sistema de comunicación telemétrico logrará mejorar el servicio prestado por la CNEL, ya que la comunicación bidireccional se efectúa en tiempo real, permitiendo atender eficientemente las demandas del servicio en horas pico administrándolas inteligentemente, asimismo, se ha conseguido una mejora de los tiempos, en lo que respecta el servicio de lectura, corte y reconexión, pues se puede obtener, suspender o reanudar la prestación del servicio inmediatamente, facilitando trámites engorrosos al usuario.

5.2 Pruebas de comandos remotos

Las pruebas de lectura realizadas al medidor implementado demuestran que este dispositivo funciona de acuerdo a las especificaciones requeridas por CNEL, dando respuesta al comando de lectura del medidor “770100”, enviando los parámetros de la trama de información en código hexadecimal como lo demuestra el Anexo 3, estos parámetros pertenecen a la cabecera de trama de lectura, el cual está compuesto por 2 dígitos que pertenecen al byte de sincronismo, 2 al byte de código de lectura, y 2 al byte de código de error; los 24 siguientes dígitos atañen al número de serie del medidor; los siguientes 8 dígitos corresponden a la fecha de lectura, en donde cuatro dígitos corresponden al año, y dos al mes en el que se toma la lectura y los 2 posteriores al día del año; los próximos 8 dígitos se dividen en 2 para determinar el día de la semana en el que se realiza la lectura, 2, a la hora, 2 a los minutos y 2 a los segundos; 16 dígitos pertenecen a la energía total activa medida en kWh; 16 a la energía total reactiva deducido en kVARh; 16 a la energía total aparente calculada en kVAh; 16 a demanda total en kW; 8 dígitos corresponden a la corriente fase A y 8 más a la corriente fase B valorados en Amp; los próximos 8 dígitos pertenecen al voltaje Fase A y los 8 posteriores al voltaje Fase B determinados en V; el factor de potencia se encuentra contenido en los 8 dígitos siguientes; la frecuencia estimada en Hz se contiene en 8 dígitos; y, la potencia activa instantánea calculado en kW constan en los últimos 8 dígitos en formato hexadecimal.

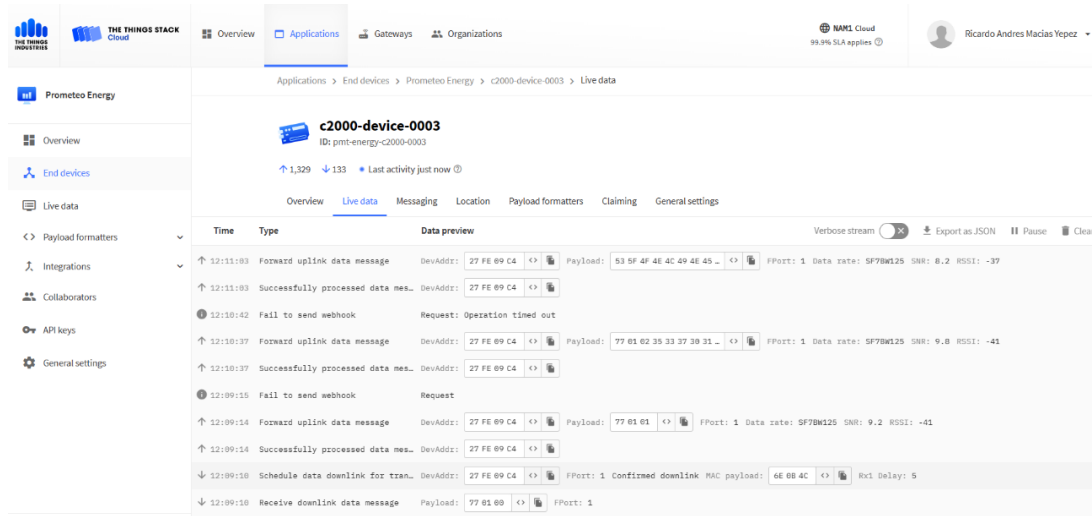


Figura 20 Interface del Gateway IOT

Fuente: ISCONET S.A, 2022

5.2.1 Códigos de Transacción

Los códigos de transición sirven para que el dispositivo empiece una acción especificada por el usuario ya sea de lectura de los valores consumidos, la ejecución de un corte o reconexión del servicio o la apertura del medidor en la vivienda del nuevo abonado.

01	LECTURA
02	CORTE
03	RECONEXIÓN
04	APERTURA

5.2.2 Códigos de Error

El código de sincronismo es un código aleatorio entre 01-99, este código es devuelto para identificar que los dispositivos estos activos o la ocurrencia de una falla en la comunicación de los dispositivos de la red MESH.

00	OK
01	OK ACK
99	ERROR

5.2.3 Trama de requerimiento de lectura

Dentro de la trama de requerimiento está compuesta por tres bytes que contendrán el código de sincronismo, el tipo de transacción a realizar (lectura, corte, reconexión o apertura), y si se diera el caso un código de error.

Tabla 5.1 Trama de requerimiento de lectura

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	77
CODIGO TRANSACCION	Código de transacción (01 lectura)	1	01
CODIGO ERROR	Código de error	1	00

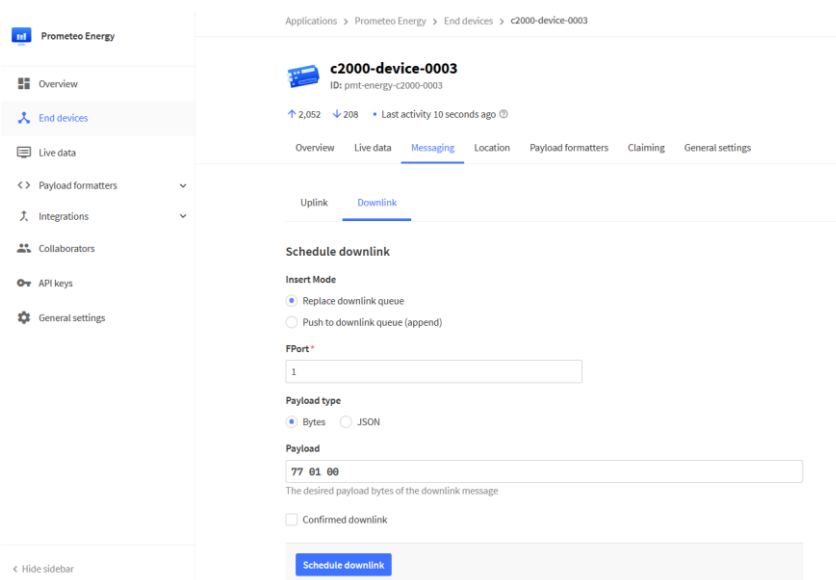


Figura 21 Petición de lectura de medidor

Fuente: ISCONET S.A, 2022

5.2.4 Trama de ACK de requerimiento de lectura

Cada vez que el dispositivo requiere hacer una solicitud desde la consola del administrado o usuario primero deberá verificar que el dispositivo se encuentra activo enviando una trama Acknowledge (ACK) como se aprecia en la **Tabla 5.2**

Tabla 5.2 Trama de ACK de requerimiento de lectura

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	77
CODIGO TRANSACCION	Código de transacción (01 lectura)	1	01
CODIGO ERROR	Código de error	1	01

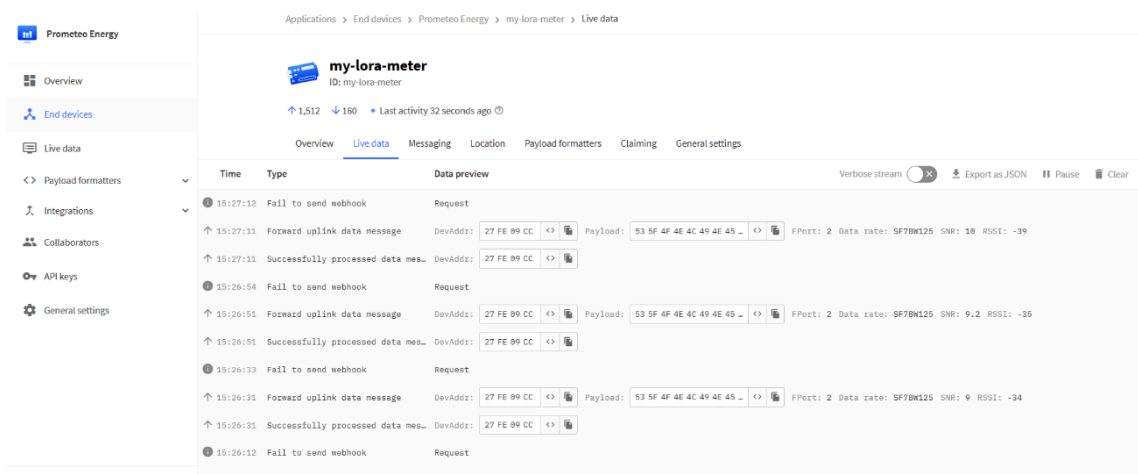


Figura 22 Trama de estructura de keep alive ACK

Fuente: ISCONET S.A, 2022

5.2.5 Trama de respuesta de lectura

Dentro de la trama de respuesta tendremos la información que el medidor se encuentra censando en tiempo real, donde tendremos la información del código del medidor, la fecha y

hora, la energía activa, la energía reactiva y aparente. De esta manera se podrá solventar la necesidad del cliente de llevar un registro con un error mínimo en las lecturas tomadas y de esta forma evitar el conflicto en la factura generadas para los abonados de la red.

Tabla 5.3. Trama de respuesta de lectura

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	77
CODIGO TRANSACCION	Código de transacción (01 lectura)	1	01
CODIGO ERROR	Código de error	1	00
CODIGO MEDIDOR	Código de medidor	12	35 33 33 30 31 36 31 31 30 30 30 33
FECHA HORA	Fecha y hora de transacción	8	07 e1 01 0a 02 0a 2a 13
ENERGIA ACTIVA	Energía active	8	00 00 00 00 00 00 00 00
ENERGIA REACTIVA	Energía reactiva	8	00 00 00 00 00 00 00 00
ENERGIA APARENTE	Energía aparente	8	00 00 00 00 00 00 00 00
+++DEMANDA MAXIMA	Demanda máxima	4	00 00 00 00
CORRIENTE FASE A	Corriente fase A	4	00 00 00 00

Operación	Descripción	Bytes	Ejemplo
CORRIENTE FASE B	Corriente fase B	4	00 00 00 00
VOLTAJE FASE A	Voltaje fase A	4	00 00 00 00

VOLTAJE FASE B	Voltaje fase B	4	00 00 00 00
FACTOR DE POTENCIA	Factor de potencia	4	00 00 00 00
FRECUENCIA	Frecuencia	4	00 00 00 00
POTENCIA ACTIVA INSTANTANEA	Potencia active instantánea	4	00 00 00 00

5.2.6 Trama de requerimiento de corte

La trama de requerimiento se efectúa cuando se requiere realizar un cambio en el estado del medidor, ya sea este por corte, reconexión o apertura. En el ejemplo expuesto en la **Tabla 5.4** podemos observar la trama **770200**, correspondiente a una acción de corte en el medidor.

Tabla 5.4. Trama de requerimiento de corte

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	77
CODIGO TRANSACCION	Código de transacción (02 corte)	1	02
CODIGO ERROR	Código de error	1	00

5.2.7 Trama de ACK de requerimiento de corte

Cuando se efectúa una acción de corte por parte del administrador la trama que se deberá genera debería tener la siguiente estructura **770201** como se aprecia en la **Tabla 5.5**.

Tabla 5.5. Trama de ACK de requerimiento de corte

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	77

CODIGO TRANSACCION	Código de transacción (02 corte)	1	02
CODIGO ERROR	Código de error	1	01

5.2.8 Trama de respuesta de corte

Cuando se efectuó una acción de corte, desde el dispositivo de usuario generara una trama de información con el siguiente bloque que informen el código del medidor, la fecha y hora, y la energía activa de dispositivo en el momento.

Tabla 5.6. Trama de respuesta de corte

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	77
CODIGO TRANSACCION	Código de transacción (02 corte)	1	02
CODIGO ERROR	Código de error	1	00
CODIGO MEDIDOR	Código de medidor	12	35 33 33 30 31 36 31 31 30 30 30 33
FECHA HORA	Fecha y hora de transacción	8	07 e1 01 0a 02 0a 2a 13
ENERGIA ACTIVA	Energía active	8	00 00 00 00 00 00 00 00
ENERGIA REACTIVA	Energía reactiva	8	00 00 00 00 00 00 00 00
ENERGIA APARENTE	Energía aparente	8	00 00 00 00 00 00 00 00
DEMANDA MAXIMA	Demanda máxima	4	00 00 00 00
CORRIENTE FASE A	Corriente fase A	4	00 00 00 00
CORRIENTE FASE B	Corriente fase B	4	00 00 00 00

VOLTAJE FASE A	Voltaje fase A	4	00 00 00 00
VOLTAJE FASE B	Voltaje fase B	4	00 00 00 00
FACTOR DE POTENCIA	Factor de potencia	4	00 00 00 00
FRECUENCIA	Frecuencia	4	00 00 00 00
POTENCIA ACTIVA INSTANTANEA	Potencia active instantánea	4	00 00 00 00

5.2.9 Trama de requerimiento de reconexión

Cuando se efectúa una acción de corte por parte del administrador la trama que se deberá genera debería tener la siguiente estructura 770301 como se describe en la **Tabla 5.7**.

Tabla 5.7. Trama de requerimiento de reconexión

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	77
CODIGO TRANSACCION	Código de transacción (03 reconexión)	1	03
CODIGO ERROR	Código de error	1	00

5.2.10 Trama de respuesta de reconexión

Cuando se efectuó una acción de reconexión, desde el dispositivo de usuario generara una trama de información con el siguiente bloque que informen el código del medidor, la fecha y hora, y la energía activa de dispositivo en el momento.

Tabla 5.8. Trama de respuesta de reconexión

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	77
CODIGO TRANSACCION	Código de transacción (03 reconexión)	1	03
CODIGO ERROR	Código de error	1	00
CODIGO MEDIDOR	Código de medidor	12	35 33 33 30 31 36 31 31 30 30 30 33

Operación	Descripción	Bytes	Ejemplo
FECHA HORA	Fecha y hora de transacción	8	07 e1 01 0a 02 0a 2a 13
ENERGIA ACTIVA	Energía active	8	00 00 00 00 00 00 00 00
ENERGIA REACTIVA	Energía reactiva	8	00 00 00 00 00 00 00 00
ENERGIA APARENTE	Energía aparente	8	00 00 00 00 00 00 00 00
DEMANDA MAXIMA	Demanda máxima	4	00 00 00 00
CORRIENTE FASE A	Corriente fase A	4	00 00 00 00

CORRIENTE FASE B	Corriente fase B	4	00 00 00 00
VOLTAJE FASE A	Voltaje fase A	4	00 00 00 00
VOLTAJE FASE B	Voltaje fase B	4	00 00 00 00
FACTOR DE POTENCIA	Factor de potencia	4	00 00 00 00
FRECUENCIA	Frecuencia	4	00 00 00 00
POTENCIA ACTIVA INSTANTANEA	Potencia active instantánea	4	00 00 00 00

5.2.11 Trama de requerimiento de apertura

Cuando se efectúa de apertura de corte por parte del administrador la trama que se deberá genera debería tener la siguiente estructura 000400 como se describe en la **Tabla 5.9**.

Tabla 5.9. Trama de requerimiento de apertura

Operación	Descripción	Bytes	Ejemplo
CODIGO SYNC	Código de sincronismo	1	00
CODIGO TRANSACCION	Código de transacción (04 apertura)	1	04

Operación	Descripción	Bytes	Ejemplo
CODIGO ERROR	Código de error	1	00

CAPÍTULO 6

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Es conocido que actualmente las empresas de Distribución eléctrica y en este caso específicamente CNEL UN Guayaquil cuenta con diversas soluciones tanto para la toma de lectura como corte y reconexión del servicio eléctrico. El sistema de telemetría para usuarios residenciales con medidores Clase 200 manejados de forma remota.

La solución implementada consiste en una red zigbee de tipo mesh, el cual se usa concentradores que trabajan como Gateway y que los mismos envían la información a una aplicación al centro de control mediante enlaces de última milla las mismas que son contratadas a operadores de telecomunicaciones y que cobran un valor mensual por el alquiler de estas últimas millas, esta solución se implementó para medidores concentrados como el caso de ciudadelas privadas, ya que no es económicamente efectiva cuando se tienen pocos medidores o los usuarios están muy dispersos, además de que el sistema que administra la información procedente de los medidores tiene un costo de licenciamiento anual.

Otra solución basada en tecnología inalámbrica y se encuentra implementada en sectores urbano marginal para usuarios con bajo consumo de energía y cuyos medidores son Clase 100, los mismos que cuentan con un módulo de comunicación denominado RF y que solo tiene implementado el proceso de lectura, ya que los medidores no cuentan con el relay para la operación de corte y reconexión del servicio eléctrico. El medidor se maneja con una persona que toma la medida mediante un Hand Held cerca del o los medidores para que en forma inalámbrica pueda realizar la operación de lectura de los medidores, es decir la persona o técnico de realizar las lecturas debe estar cerca de los medidores, una vez realizado este proceso debe ir al centro de control a transferir o descargar los datos para que luego sean transferidos al sistema comercial o de facturación. Cabe mencionar además pueden existir errores en las lecturas de los medidores debido a que no se conectaron al equipo Hand Held o descargo mal la información, motivo por el cual el personal de lecturas tiene que regresar

al sitio a realizar nuevamente la descarga de información al sitio, y en caso de que haya que cortar el servicio eléctrico por falta de pago, esta acción hay que realizarla de forma manual en el sitio mediante una cuadrilla de corte y reconexión que por lo general es un contrato de servicios ya contratado por la Unidad de Negocios.

Con la solución propuesta podemos nombrar las ventajas y bondades ya que se trata de una implementación de una solución totalmente inalámbrica y todas las operación se realizan de forma remota en lecturas corte y reconexión del servicio eléctrico, además de que en la implementación se debe diseñar la red de comunicaciones con la tecnología Lora WAN de acuerdo al área de cobertura donde van a ser instalados los medidores, como se sabe esta tecnología es de baja potencia y de largo alcance hasta algunos kilómetros podemos determinar que para cubrir toda el área urbana bastarían unos pocos Gateway, además de que es una red privada no se incurren costos de transmisión sino los costos de conexión a internet por cada Gateway instalado y que estos valores con el plan más bajo de internet puede comunicar a miles de medidores.

Y como valor agregado a la solución desarrollada podemos indicar como bondad que en la descarga de información no solamente se descarga la energía Activa que es la que sirve para el departamento comercial para el proceso de facturación, sino que también se descarga del medidor información de calidad de energía como factor de potencia, energía reactiva, demanda máxima, frecuencia, demanda total acumulada, voltajes de línea, corrientes de línea, entre otros, lo que diferencia de la solución con Hand Held que solo pueden leer energía activa y sin realizar la operación de corte y reconexión, por otro lado la solución desarrollada se implementó en el módulo electrónico de telecomunicación el evento de apertura de caja de medidor que sería una alerta y queda registrado mediante una trama, y el mismo puede enviar un comando de corte de energía.

Sería motivo de otro desarrollo de software mediante una integración a la plataforma donde arriban las tramas de datos procedentes de los medidores, para mostrar la información de los medidores en plataformas o aplicaciones web tipo DASHBOARDS, mediante UBIDOTS, AWS, MQTT, etc., con lo cual se tendría una solución completa tanto de

hardware como de software, que sería muy competitiva por contar con elementos adicionales y que no tienen las otras soluciones descritas anteriormente.

6.2 Recomendaciones

Para trabajos futuros dentro del ámbito de la telemetría en medidores de energía eléctrica, se recomienda realizar un estudio de propagación con el fin de determinar el despliegue de una infraestructura de telecomunicaciones con tecnología *LoRaWAN*, con el fin de considerar aspectos como la zona de implementación, ya que la densidad de edificios, zona de árboles, la topografía del sector y otros influyen; disminuyendo el área de cobertura con respecto a zonas en las cuales hay línea de vista.

Otro aspecto para tener en cuenta es la instalación del *router outdoor*, este debe ser instalado en partes o lugares altos, con el fin de extender el área de cobertura, de la misma manera, la antena a instalar en los módulos de comunicación debe ser de una buena ganancia, para garantizar una buena comunicación entre el dispositivo final instalado en el medidor y el *gateway* en ocasiones que no haya línea de vista entre ellos.

Se recomienda también usar módulos de comunicación *LoRaWAN* que sean certificados y que cuenten con *DEVEUI* de fábrica, también es recomendable no usar aplicaciones de generación de códigos para evitar que los módulos sean registrados en plataformas de *IoT*. Para el diseño de la tarjeta electrónica de comunicación *IOT*, es muy importante considerar la ubicación del módulo de comunicación en el diseño de la tarjeta electrónica.

Los sistemas de servicios públicos esenciales tales como el servicio de suministro de agua potable o energía eléctrica, son aquellos que directa o indirectamente presta o suministra la Administración Pública para satisfacer necesidades colectivas. Estos servicios están sujetos a un régimen jurídico especial y al control de autoridad competente. Por lo antes mencionado, en estos sistemas coexisten tres actores importantes a saber: el prestador del servicio, el regulador o autoridad competente y el usuario del servicio.

Indistintamente que el servicio público sea provisto directamente por el estado o por una delegación a un tercero (concesión u otro tipo de delegación), es de vital importancia para su éxito, la armonía entre los tres actores principales antes mencionados. Siendo que la prestación del servicio por parte del prestador, controlado por el regulador, no es gratuita y debe ser cubierta por el usuario por medio de una tarifa de uso, tarifa que debe considerar cubrir los costos totales del prestador pero sin incluir ineficiencias y en este caso siendo un servicio como el de energía eléctrica, debe además ser transparente y confiable en cuanto a lo que el consumidor percibe como correctamente facturado por el uso que ha hecho del servicio y no debido a “fallas” o dudas de un sistema que no puede auditar en forma directa, sino que debe confiar en que el mismo opera en forma correcta y sin fallas que lo perjudiquen.

La importancia entonces de medir en forma correcta el consumo de un recurso escaso, en este caso el recurso energía eléctrica, al que tiene derecho toda la población (derechos garantizados por la legislación de los países y por reglamentaciones supranacionales) y permitir que el usuario pueda seguir de alguna manera sus hábitos de consumo, de forma tal que al acceder a sus consumos; por medios controlados provistos por el prestador del servicio, le permita garantizar que pueda presentar sus reclamaciones a las que crea tenga derecho mediante un procedimiento adecuado y confiable.

Así mismo para el prestador, la medición correcta y oportuna de los consumos de los usuarios, de forma que le permita observar en tiempo real, los picos de demanda o las horas valle (demanda menor) le permitirá a su vez, ejecutar de forma más eficiente las políticas de administración de su cartera de clientes, de su sistema de facturación y cobranza así como de su planificación y administración “inteligente” de sus redes de servicios (subestaciones, redes alimentadoras y de acometida final al usuario). Todo esto le resultará propicio además de necesario, para la implementación de nuevas redes que permitan llevar el acceso a la energía eléctrica a los sectores poblados aún no atendidos.

Esto porque, si se mide en forma correcta, confiable y oportuna u recurso además escaso, se pueden establecer los sistemas tarifarios correctos entre los cuales están los que

consumen para uso doméstico con y sin subsidio, como para los que usan la energía para producción o generación de ingresos.

Las políticas públicas deben ir encaminadas a un mejor control de la medición y distribución de los recursos, en varios casos en forma de subsidios, dichas subvenciones son financiadas por los demás usuarios con tarifas diferentes, acorde a su actividad económica.

El desconocer lo que se consume y lo que se subsidia por dichos consumos, así como la errática forma de hacerlo (lecturas semi automáticas, manuales) le agregan al servicio y a la empresa que lo provee, una carga negativa e innecesaria de incertidumbre, que al final afecta la buena reputación de las empresas proveedoras, tal como se ha observado en estos últimos años en el Ecuador.

El conocer lo que se consume, colabora en crear un “usuario inteligente” del recurso, puesto que aprenderá de sus hábitos de consumo, esperamos sea mucha más proclive a aceptar los costos de ese uso eléctrico real y controlado por su voluntad, además facilitar su colaboración para tener conductas que mejoren disminuyendo su desperdicio lo que las abona en los objetivos de ahorro energético y disminución de su “huella de carbono” ante el desafío mundial del calentamiento global.

Por tanto, el modelo de medición de energía que se propone en este trabajo, al ser preciso, disponible para todos los actores (empresa prestadora y usuarios) y en tiempo real, hace de este elemento “módulo de medidor inteligente para el consumo de energía eléctrica” parte fundamental y esencial de un sistema “inteligente” de suministro de energía eléctrica a los usuarios.

Entre las características de funcionalidad de este módulo, tenemos:

- Datos de consumo;
- Factor de potencia;
- Calidad de energía recibida.

Entre otros, esto hace que los sistemas de la empresa distribuidora de energía puedan gestionar y administrar desde el modo llamado redes inteligentes de energía, pues este tipo de modelo requiere los ajustes técnicos en paralelo al desarrollo del servicio para atender picos de demanda, o habilitar tarifas que incentiven los segmentos valle o la recarga de energía a la red, como por ejemplo autos eléctricos a batería o sistemas de generación fotovoltaica o de otras fuentes.

En específico, con la tecnología de telecomunicaciones implementada en este módulo, (con tecnología LoRaWAN) el despliegue ordenado y planificado de esta red en la ciudad de Guayaquil, permitirá un procedimiento de medición inteligente mediante redes cuyos nodos sean desagregados o independientes, con costos de despliegue bajos y costos de transmisión de información de bajo presupuesto, pudiendo transportar la información desde el medidor del hogar a los sistemas comerciales de las unidades de negocio de CNEL, la misma que puede ser desplegada de una manera rápida y que pueda ir creciendo de manera sostenida y controlada.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Lanchuan Zhou and J. B. Schneider, “A Study of RF Dosimetry from Exposure to an AMI Smart Meter,” *IEEE Antennas Propag. Mag.*, vol. 54, no. 6, pp. 69–80, Dec. 2012, doi: 10.1109/MAP.2012.6387783.
- [2] T. Khalifa, K. Naik, and A. Nayak, “A Survey of Communication Protocols for Automatic Meter Reading Applications,” *IEEE Commun. Surv. Tutor.*, vol. 13, no. 2, pp. 168–182, 2011, doi: 10.1109/SURV.2011.041110.00058.
- [3] E. Haro, T. Guarda, A. O. Z. Peñaherrera, and N. Quiña, “Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot”.
- [4] K. D. Craemer and G. Deconinck, “Analysis of State-of-the-art Smart Metering Communication Standards”.
- [5] S. Feuerhahn, M. Zillgith, C. Wittwer, and C. Wietfeld, “Comparison of the communication protocols DLMS/COSEM, SML and IEC 61850 for smart metering applications,” in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Brussels, Belgium, Oct. 2011, pp. 410–415. doi: 10.1109/SmartGridComm.2011.6102357.
- [6] J. Steenbruggen, M. T. Borzacchiello, P. Nijkamp, and H. Scholten, “Mobile phone data from GSM networks for traffic parameter and urban spatial pattern assessment: a review of applications and opportunities,” *GeoJournal*, vol. 78, no. 2, pp. 223–243, Apr. 2013, doi: 10.1007/s10708-011-9413-y.
- [7] A. M. Lindquist, P. E. Johansson, G. I. Petersson, B.-I. Saveman, and G. C. Nilsson, “The Use of the Personal Digital Assistant (PDA) Among Personnel and Students in Health Care: A Review,” *J. Med. Internet Res.*, vol. 10, no. 4, p. e31, Oct. 2008, doi: 10.2196/jmir.1038.
- [8] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A Literature Review,” *J. Comput. Commun.*, vol. 03, no. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.
- [9] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, “LoRa and LoRaWAN testbeds: A review,” in *2017 IEEE AFRICON*, Cape Town, Sep. 2017, pp. 1496–1501. doi: 10.1109/AFRCON.2017.8095703.
- [10] J. de C. Silva, A. M. Alberti, P. Solic, and A. L. L. Aquino, “LoRaWAN - A Low Power WAN Protocol for Internet of Things: a Review and Opportunities”.
- [11] N. Goldenberg and A. Wool, “Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems,” *Int. J. Crit. Infrastruct. Prot.*, vol. 6, no. 2, pp. 63–75, Jun. 2013, doi: 10.1016/j.ijcip.2013.05.001.
- [12] B. Lichtensteiger, B. Bjelajac, C. Mueller, and C. Wietfeld, “RF Mesh Systems for Smart Metering: System Architecture and Performance,” in *2010 First IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, Oct. 2010, pp. 379–384. doi: 10.1109/SMARTGRID.2010.5622071.
- [13] A. Dunkels, J. Alonso, and T. Voigt, “Making TCP/IP Viable for Wireless Sensor Networks”.

- [14] Y. Zhang, N. Ansari, M. Wu, and H. Yu, "On Wide Area Network Optimization," *IEEE Commun. Surv. Tutor.*, vol. 14, no. 4, pp. 1090–1113, 2012, doi: 10.1109/SURV.2011.092311.00071.
- [15] S. Safaric and K. Malaric, "ZigBee wireless standard," in *Proceedings ELMAR 2006*, Zadar, Croatia, Jun. 2006, pp. 259–262. doi: 10.1109/ELMAR.2006.329562.
- [16] S. Kufeoglu and M. Lehtonen, "A review on the theory of electric power reliability worth and customer interruption costs assessment techniques," in *2016 13th International Conference on the European Energy Market (EEM)*, Porto, Portugal, Jun. 2016, pp. 1–6. doi: 10.1109/EEM.2016.7521239.
- [17] M. Rezaei, B. Ghobadian, S. H. Samadi, and S. Karimi, "Electric power generation from municipal solid waste: A techno-economical assessment under different scenarios in Iran," *Energy*, vol. 152, pp. 46–56, Jun. 2018, doi: 10.1016/j.energy.2017.10.109.
- [18] P. Kotsampopoulos *et al.*, "EMC Issues in the Interaction Between Smart Meters and Power-Electronic Interfaces," *IEEE Trans. Power Deliv.*, vol. 32, no. 2, pp. 822–831, Apr. 2017, doi: 10.1109/TPWRD.2016.2561238.
- [19] L. A. Elrefaei, A. Bajaber, S. Natheir, N. AbuSanab, and M. Bazi, "Automatic electricity meter reading based on image processing," in *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Amman, Jordan, Nov. 2015, pp. 1–5. doi: 10.1109/AEECT.2015.7360571.
- [20] J. Hamilton Ortiz, J. Ferney Castillo Garc韉, O. Ibrahim Khalaf, F. Velez Varela, P. Jefferson Barragan Baron, and J. Hadder Guzm醤 Atehort鷄, "Development of a Module for Measuring Electrical Variables in Power Transformers Based in IoT, to Manage and Monitoring by Telemetry Mechanism," *J. Internet Things*, vol. 3, no. 2, pp. 53–63, 2021, doi: 10.32604/jiot.2021.014736.
- [21] L. A. Irwin, "A high accuracy standard for electricity meters," in *IEEE PES T&D 2010*, New Orleans, LA, USA, 2010, pp. 1–3. doi: 10.1109/TDC.2010.5484291.
- [22] A. Morales-Acevedo, "Forecasting Future Energy Demand: Electrical Energy in Mexico as an Example Case," *Energy Procedia*, vol. 57, pp. 782–790, 2014, doi: 10.1016/j.egypro.2014.10.286.
- [23] L. P. C. Do, K.-H. Lin, and P. Molnár, "Electricity consumption modelling: A case of Germany," *Econ. Model.*, vol. 55, pp. 92–101, Jun. 2016, doi: 10.1016/j.econmod.2016.02.010.
- [24] T. Adiono and A. R. Daud, "Electricity Smart Meter Payment System Through Payment Gateway and User Interface Design," in *2021 15th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, Bali, Indonesia, Nov. 2021, pp. 1–4. doi: 10.1109/TSSA52866.2021.9768260.
- [25] M. Bonisławski, M. Hołub, T. Borkowski, and P. Kowalak, "A Novel Telemetry System for Real Time, Ship Main Propulsion Power Measurement," *Sensors*, vol. 19, no. 21, p. 4771, Nov. 2019, doi: 10.3390/s19214771.
- [26] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Inf. Syst. Front.*, vol. 17, no. 2, pp. 243–259, Apr. 2015, doi: 10.1007/s10796-014-9492-7.
- [27] Lun Cuifen, Zhang Xiaoqin, Li Yanping, and Liuce, "The electric meter reading system in rural areas based on wireless micro-computer," in *2010 International Conference On Computer Design and Applications*, Qinhuaungdao, China, Jun. 2010, pp. V1-109-V1-111. doi: 10.1109/ICCD.2010.5541117.

- [28] S. Aheleroff *et al.*, “IoT-enabled smart appliances under industry 4.0: A case study,” *Adv. Eng. Inform.*, vol. 43, p. 101043, Jan. 2020, doi: 10.1016/j.aei.2020.101043.
- [29] N. Hossein Motlagh, M. Mohammadrezaei, J. Hunt, and B. Zakeri, “Internet of Things (IoT) and the Energy Sector,” *Energies*, vol. 13, no. 2, p. 494, Jan. 2020, doi: 10.3390/en13020494.
- [30] R. Porkodi and V. Bhuvaneshwari, “The Internet of Things (IoT) Applications and Communication Enabling Technology Standards: An Overview,” in *2014 International Conference on Intelligent Computing Applications*, Coimbatore, India, Mar. 2014, pp. 324–329. doi: 10.1109/ICICA.2014.73.
- [31] P.-Y. Chen, L. Bhatia, R. Kolcun, D. Boyle, and J. A. McCann, “Contact-Aware Opportunistic Data Forwarding in Disconnected LoRaWAN Mobile Networks,” in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, Singapore, Singapore, Nov. 2020, pp. 574–583. doi: 10.1109/ICDCS47774.2020.00110.
- [32] G. Ferre and A. Giremus, “LoRa Physical Layer Principle and Performance Analysis,” in *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Bordeaux, Dec. 2018, pp. 65–68. doi: 10.1109/ICECS.2018.8617880.
- [33] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, “LoRa indoor coverage and performance in an industrial environment: Case study,” in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETF A)*, Limassol, Sep. 2017, pp. 1–8. doi: 10.1109/ETF A.2017.8247601.
- [34] D. Kjendal, “LoRa-Alliance Regional Parameters Overview,” *J. ICT Stand.*, Apr. 2021, doi: 10.13052/jicts2245-800X.914.
- [35] O. Ileri, D. Samarzija, T. Sizer, and N. B. Mandayam, “Demand responsive pricing and competitive spectrum allocation via a spectrum server,” in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, Baltimore, MD, USA, 2005, pp. 194–202. doi: 10.1109/DYSPAN.2005.1542635.
- [36] M. A. M. Almuahaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, “A Survey on LoRaWAN Technology: Recent Trends, Opportunities, Simulation Tools and Future Directions,” *Electronics*, vol. 11, no. 1, p. 164, Jan. 2022, doi: 10.3390/electronics11010164.
- [37] P. K. Devi, M. Sujatha, and J. A. Prasath, “Design of Reconfigurable Integrated patch antenna in ISM band for IoT applications,” in *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India, Jul. 2020, pp. 1–4. doi: 10.1109/ICSSS49621.2020.9202152.
- [38] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, “A survey of LoRaWAN for IoT: From technology to application,” *Sensors (Switzerland)*, vol. 18, no. 11. 2018. doi: 10.3390/s18113995.
- [39] S. Vishnu *et al.*, “IoT-Enabled Solid Waste Management in Smart Cities,” *Smart Cities*, vol. 4, no. 3, pp. 1004–1017, Jul. 2021, doi: 10.3390/smartcities4030053.
- [40] Q. Zhou, K. Zheng, L. Hou, J. Xing, and R. Xu, “Design and Implementation of Open LoRa for IoT,” *IEEE Access*, vol. 7, pp. 100649–100657, 2019, doi: 10.1109/ACCESS.2019.2930243.

- [41] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14–21, Mar. 2017, doi: 10.1016/j.icte.2017.03.004.
- [42] N. Chinchilla-Romero, J. Navarro-Ortiz, P. Muñoz, and P. Ameigeiras, "Collision avoidance resource allocation for LoRaWAN," *Sens. Switz.*, vol. 21, no. 4, 2021, doi: 10.3390/s21041218.
- [43] C. Lipps, A. Weinand, D. Krummacker, C. Fischer, and H. D. Schotten, "Proof of Concept for IoT Device Authentication Based on SRAM PUFs Using ATMEGA 2560-MCU," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, South Padre Island, TX, Apr. 2018, pp. 36–42. doi: 10.1109/ICDIS.2018.00013.

ANEXOS

Anexo 1: Firmware de medidor para microcontrolador ATMEGA 2560

```
#include <TheThingsNetwork.h>
```

```
#include <TheThingsMessage.h>
```

```
const char* HANDSHAKE1 = "2F3F414141414141414141414141210D0A";
```

```
const char* HANDSHAKE2 = "063235320D0A";
```

```
const char* HANDSHAKE3 = "7EA0070341935A647E";
```

```
byte  METERNUMBER0[27]  =  {0x7E,0xA0,0x19,0x03,0x41,0x32,0x3A,0xBD,  
0xE6,0xE6,0x00,0xC0,0x01,0xC1,0x00,0x01,0x01,0x00,0x00,0x00,0x00,0xFF,0x02,0x00,  
0xC3,0xAD,0x7E};
```

```
byte  CUT[29]  =  {0x7E,0xA0,0x1B,0x03,0x41,0x54,0x7C,0x82,0xE6,  
0xE6,0x00,0xC3,0x01,0xC1,0x00,0x46,0x00,0x00,0x60,0x03,0x0A,0xFF,0x01,0x01,0x0F,  
0x00,0x68,0xA6,0x7E};
```

```
byte  RECONNECT[29  ]  =  {0x7E,0xA0,0x1B,0x03,0x41,0x54,0x7C,0x82,  
0xE6,0xE6,0x00,0xC3,0x01,0xC1,0x00,0x46,0x00,0x00,0x60,0x03,0x0A,0xFF,0x02,0x01,  
0x0F,0x00,0xA5,0x83,0x7E};
```

```
const char* HANDSHAKEBYE0 = "7EA0070341F14E247E";
```

```
const char* HANDSHAKEBYE1 = "7EA00703415356A27E";
```

```
byte  HANDSHAKE4[70]  =  {0x7E,0xA0,0x44,0x03,0x41,0x10,0xB3,0xE1,  
0xE6,0xE6,0x00,0x60,0x36,0xA1,0x09,0x06,0x07,0x60,0x85,0x74,0x05,0x08,0x01,0x01,  
0x8A,0x02,0x07,0x80,0x8B,0x07,0x60,0x85,0x74,0x05,0x08,0x02,0x01,0xAC,0x0A,0x80,  
0x08,0x31,0x31,0x31,0x31,0x31,0x31,0x31,0x31,0x31,0xBE,0x10,0x04,0x0E,0x01,0x00,0x00,  
0x00,0x06,0x5F,0x1F,0x04,0x00,0x00,0x7E,0x1F,0x04,0xB0,0xDB,0x3A,0x7E};
```

byte CMD0[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x54, 0x0A, 0xBB, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x02, 0x00, 0xC3, 0xAD, 0x7E};

byte CMD1[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x76, 0x1A, 0xB9, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x08, 0x00, 0x00, 0x01, 0x00, 0x00, 0xFF, 0x02, 0x00, 0x60, 0x1A, 0x7E};

byte CMD2[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x98, 0x6A, 0xB7, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x08, 0x00, 0x00, 0x01, 0x00, 0x00, 0xFF, 0x03, 0x00, 0xB8, 0x03, 0x7E};

byte CMD3[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0xBA, 0x7A, 0xB5, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x08, 0x00, 0x00, 0x01, 0x00, 0x00, 0xFF, 0x04, 0x00, 0xB0, 0x4E, 0x7E};

byte CMD4[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0xDC, 0x4A, 0xB3, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x08, 0x00, 0x00, 0x01, 0x00, 0x00, 0xFF, 0x05, 0x00, 0x68, 0x57, 0x7E};

byte CMD5[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0xFE, 0x5A, 0xB1, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x08, 0x00, 0x00, 0x01, 0x00, 0x00, 0xFF, 0x06, 0x00, 0x00, 0x7D, 0x7E};

byte CMD6[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x10, 0x2A, 0xBF, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x08, 0x00, 0x00, 0x01, 0x00, 0x00, 0xFF, 0x07, 0x00, 0xD8, 0x64, 0x7E};

byte CMD7[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x32, 0x3A, 0xBD, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x08, 0x00, 0x00, 0x01, 0x00, 0x00, 0xFF, 0x08, 0x00, 0x10, 0xE7, 0x7E};

byte CMD8[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x54, 0x0A, 0xBB, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x08, 0x00, 0x00, 0x01, 0x00, 0x00, 0xFF, 0x09, 0x00, 0xC8, 0xFE, 0x7E};

byte CMD9[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x76, 0x1A, 0xB9, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x01, 0x08, 0x00, 0xFF, 0x02, 0x00, 0xE7, 0xF7, 0x7E};

byte CMD10[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x98, 0x6A, 0xB7, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x03, 0x08, 0x00, 0xFF, 0x02, 0x00, 0xB1, 0xFF, 0x7E};

byte CMD11[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0xBA, 0x7A, 0xB5, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x09, 0x08, 0x00, 0xFF, 0x02, 0x00, 0xBF, 0xD6, 0x7E};

byte CMD12[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0xDC, 0x4A, 0xB3, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x04, 0x01, 0x01, 0x01, 0x06, 0x00, 0xFF, 0x02, 0x00, 0xBD, 0x7F, 0x7E};

byte CMD13[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0xFE, 0x5A, 0xB1, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x04, 0x01, 0x01, 0x01, 0x06, 0x00, 0xFF, 0x05, 0x00, 0xB5, 0x32, 0x7E};

byte CMD14[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x10, 0x2A, 0xBF, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x1F, 0x07, 0x00, 0xFF, 0x02, 0x00, 0x09, 0xE6, 0x7E};

byte CMD15[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x32, 0x3A, 0xBD, 0xE6, 0xE6, 0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x33, 0x07, 0x00, 0xFF, 0x02, 0x00, 0x9D, 0x52, 0x7E};

```
byte CMD16[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x54, 0x0A, 0xBB, 0xE6, 0xE6,
0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x20, 0x07, 0x00, 0xFF, 0x02,
0x00, 0x50, 0x1C, 0x7E};
```

```
byte CMD17[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x76, 0x1A, 0xB9, 0xE6, 0xE6,
0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x34, 0x07, 0x00, 0xFF, 0x02,
0x00, 0x4C, 0x4E, 0x7E};
```

```
byte CMD18[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0x98, 0x6A, 0xB7, 0xE6, 0xE6,
0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x0D, 0x07, 0x00, 0xFF, 0x02,
0x00, 0xEF, 0xAC, 0x7E};
```

```
byte CMD19[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0xBA, 0x7A, 0xB5, 0xE6, 0xE6,
0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x0E, 0x07, 0x00, 0xFF, 0x02,
0x00, 0x92, 0xA0, 0x7E};
```

```
byte CMD20[27] = {0x7E, 0xA0, 0x19, 0x03, 0x41, 0xDC, 0x4A, 0xB3, 0xE6, 0xE6,
0x00, 0xC0, 0x01, 0xC1, 0x00, 0x03, 0x01, 0x01, 0x01, 0x07, 0x00, 0xFF, 0x02,
0x00, 0x1B, 0x9D, 0x7E};
```

```
int apertura = 0;
```

```
int caja = 22;
```

```
int _running = 23;
```

```
const char *appEui = "70B3D57ED0040A63";
```

```
const char *appKey = "7E3D09831E1C8AF276AA4251A1E3FA9A";
```

```
#define debugSerial Serial
```

```

#define medidorSerial Serial1
#define LoRaSerial Serial2

int reseteo_LoRa = 40;
#define freqPlan TTN_FP_US915
TheThingsNetwork ttn(LoRaSerial, debugSerial, freqPlan);
void EnviarLoRa(const byte * byteBuffer, int len)
{
    ttn.sendBytes(byteBuffer, len);
    delay(10000);
}
void EnviarStringToLoRa(String trama)
{
    int len = trama.length();
    byte byteBuffer[len + 1 ];
    trama.getBytes(byteBuffer, len + 1);
    ttn.sendBytes(byteBuffer, sizeof(byteBuffer));
    delay(10000);
}
void message(const uint8_t *payload, size_t size, port_t port)
{
    debugSerial.println("***MESSAGE***");
    debugSerial.println("Received " + String(size) + " bytes on port " + String(port) + ":");

    if(size > 0)
    {
        if(size == 3)
        {
            String sync = String(payload[0]);
            String trans = String(payload[1]);
            String denial = String(payload[2]);

```

```
debugSerial.println("SYNC: " + String(payload[0]));
debugSerial.println("TRANS: " + String(payload[1]));
debugSerial.println("DENIAL: " + String(payload[2]));
```

```
if(denial == "0")
```

```
{
```

```
  byte byteArrayAck[3] = {0};
  byteArrayAck[0] = payload[0];
  byteArrayAck[1] = payload[1];
  byteArrayAck[2] = 1;
  EnviarLoRa(byteArrayAck, 3);
```

```
if(trans == "1" || trans == "2" || trans == "3")
```

```
{
```

```
  if(trans == "2")
```

```
    Corte();
```

```
  if(trans == "3")
```

```
    Reconexion();
```

```
  int longitudLectura = 79;
```

```
  byte byteArrayLectura[longitudLectura] = {0};
```

```
  String trxLectura = "";
```

```
  trxLectura = trxLectura + Lectura(byteArrayLectura);// con la respuesta se envia por
```

LORA

```
  byteArrayLectura[0] = payload[0];
```

```
  byteArrayLectura[1] = payload[1];
```

```
  byteArrayLectura[2] = 2;
```

```

debugSerial.println();
debugSerial.println("Enviando a LORA...");
debugSerial.println("Buffer final: ");
for (int i = 0; i < longitudLectura ; i++)
{
    debugSerial.print("0x");
    if (byteArrayLectura[i] < 0x10)
        debugSerial.print("0");
    debugSerial.print(byteArrayLectura[i], HEX);
    debugSerial.print(" ");
}
debugSerial.println(" ");
EnviarLoRa(byteArrayLectura, longitudLectura);
}
}
}
}
debugSerial.println();
}

```

String PadLeft(String txt, String character, int len)

```

{
    String result = "";
    if(len > 0 && (txt.length() < len))
    {
        for(int i = 0; i < len; i++)
        {
            if(txt.length() == len)
                break;
            else
            {

```

```

        result = character + txt;
        txt = result;
    }
}
}
else
    if(len > 0 && txt.length() > len)
        txt = txt.substring(0, len);
return txt;
}

void CapturaSerial()
{
    String strConcat = "";
    int iAvailable = 1;
    int contador = 1;
    int c = 1;
    ("Respuesta: ");
    while(true)
    {
        while((iAvailable = medidorSerial.available()) > 0)
        {
            int incoming = medidorSerial.read();
            String str = String(incoming, HEX);
            debugSerial.print(str);
            debugSerial.print(", ");
            strConcat += char(incoming);
        }

        if(contador == 3)
            break;

```



```

else
{
    delay(250);
    contador++;
}
}
debugSerial.println();
}
String CapturaDataSerial(byte * byteBuffer,int indexBuffer, int inicio, int fin)
{
    String strConcat = "";
    int iAvailable = 1;
    int contador = 1;
    int index = 0;
    debugSerial.print("Respuesta: ");
    while(true)
    {
        while((iAvailable = medidorSerial.available()) > 0)
        {
            byte incoming = medidorSerial.read();

            String str = String(incoming, HEX);
            debugSerial.print(str);
            debugSerial.print(", ");
            if(index >= inicio && index <= fin)
            {
                byteBuffer[indexBuffer++] = incoming;
                strConcat += PadLeft(str,"0",2);
                //indexBuffer++;
            }
            index++;

```

```

    }

    if(contador == 3)
        break;
    else
    {
        delay(250);
        contador++;
    }
}
debugSerial.println("Salio!!!");
debugSerial.println();

return strConcat;
}

```

```

String Lectura(byte * byteArrayLectura)
{
    String trx = "";
    String respuestaLectura = "";
    //int longitudLectura = 75;
    int indexBuffer = 3;
    //byte byteArrayLectura[longitudLectura] = {0};

    delay(500);
    int longitud = 17;
    byte byteArray[longitud] = {0};
    hexCharacterStringToBytes(byteArray, HANDSHAKE1);
    dumpByteArray(byteArray, longitud);

    delay(2000);
}

```

```

longitud = 6;
byteArray[longitud] = {0};
hexCharacterStringToBytes(byteArray, HANDSHAKE2);
dumpByteArray(byteArray, longitud);
delay(2000);
longitud = 9;
byteArray[longitud] = {0};
hexCharacterStringToBytes(byteArray, HANDSHAKE3);
dumpByteArray(byteArray, longitud);
delay(2000);
longitud = 70;
dumpByteArray(HANDSHAKE4, longitud);
delay(2000);
longitud = 27;
dumpByteArray(METERNUMBER0, longitud);
delay(2000);
int cantidadLectura = 12;
debugSerial.println("CODIGO MEDIDOR");
longitud = 27;
dumpByteArray(CMD0, longitud);
trx = CapturaDataSerial(byteArrayLectura,indexBuffer, 80,91);
//trx = CapturaDataSerial(80,91);//12
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);
debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
debugSerial.print("0x");
if (byteArrayLectura[i] < 0x10)
debugSerial.print("0");

```

```

    debugSerial.print(byteArrayLectura[i], HEX);
    debugSerial.print(" ");
}
debugSerial.println(" ");
delay(2000);
indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 8;

debugSerial.println("RELOJ");
longitud = 27;
dumpByteArray(CMD1, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 17,24);
//trx = CapturaDataSerial(17,24);//8
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);
debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
    debugSerial.print("0x");
    if (byteArrayLectura[i] < 0x10)
        debugSerial.print("0");
    debugSerial.print(byteArrayLectura[i], HEX);
    debugSerial.print(" ");
}
debugSerial.println(" ");
delay(2000);
debugSerial.println("P1");
longitud = 27;
dumpByteArray(CMD2, longitud);
CapturaSerial();
delay(2000);

```

```
debugSerial.println("P2");  
longitud = 27;  
dumpByteArray(CMD3, longitud);  
CapturaSerial();
```

```
delay(2000);  
debugSerial.println("P3");  
longitud = 27;  
dumpByteArray(CMD4, longitud);  
CapturaSerial();
```

```
delay(2000);  
debugSerial.println("P4");  
longitud = 27;  
dumpByteArray(CMD5, longitud);  
CapturaSerial();
```

```
delay(2000);  
debugSerial.println("P5");  
longitud = 27;  
dumpByteArray(CMD6, longitud);  
CapturaSerial();
```

```
delay(2000);  
debugSerial.println("P6");  
longitud = 27;  
dumpByteArray(CMD7, longitud);  
CapturaSerial();
```

```
delay(2000);  
debugSerial.println("P7");
```

```

longitud = 27;
dumpByteArray(CMD8, longitud);
CapturaSerial();

delay(2000);

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 8;

debugSerial.println("ENERGIA ACTIVA");
longitud = 27;
dumpByteArray(CMD9, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,23);
//trx = CapturaDataSerial(16,23);//8
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
debugSerial.print("0x");
if (byteArrayLectura[i] < 0x10)
debugSerial.print("0");
debugSerial.print(byteArrayLectura[i], HEX);
debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

indexBuffer = indexBuffer + cantidadLectura;

```

```

cantidadLectura = 8;

debugSerial.println("ENERGIA REACTIVA");
longitud = 27;
dumpByteArray(CMD10, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,23);
//trx = CapturaDataSerial(16,23);//8
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
  for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
  debugSerial.print("0x");
  if (byteArrayLectura[i] < 0x10)
    debugSerial.print("0");
  debugSerial.print(byteArrayLectura[i], HEX);
  debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 8;

debugSerial.println("ENERGIA APARENTE");
longitud = 27;
dumpByteArray(CMD11, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,23);
//trx = CapturaDataSerial(16,23);//8

```

```

respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
debugSerial.print("0x");
if (byteArrayLectura[i] < 0x10)
    debugSerial.print("0");
debugSerial.print(byteArrayLectura[i], HEX);
debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 4;

debugSerial.println("DEMANDA MAXIMA");
longitud = 27;
dumpByteArray(CMD12, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,19);
//trx = CapturaDataSerial(16,19);//4
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
debugSerial.print("0x");

```



```

if (byteArrayLectura[i] < 0x10)
    debugSerial.print("0");
debugSerial.print(byteArrayLectura[i], HEX);
debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

debugSerial.println("P8");
longitud = 27;
dumpByteArray(CMD13, longitud);
CapturaSerial();

delay(2000);
indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 4;
debugSerial.println("CORRIENTE FASE A");
longitud = 27;
dumpByteArray(CMD14, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,19);
//trx = CapturaDataSerial(16,19);//4
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
    debugSerial.print("0x");
    if (byteArrayLectura[i] < 0x10)
        debugSerial.print("0");

```

```

    debugSerial.print(byteArrayLectura[i], HEX);
    debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 4;

debugSerial.println("CORRIENTE FASE B");
longitud = 27;
dumpByteArray(CMD15, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,19);
//trx = CapturaDataSerial(16,19);//4
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
    debugSerial.print("0x");
    if (byteArrayLectura[i] < 0x10)
        debugSerial.print("0");
    debugSerial.print(byteArrayLectura[i], HEX);
    debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

```

```

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 4;

debugSerial.println("VOLTAJE FASE A");
longitud = 27;
dumpByteArray(CMD16, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,19);
//trx = CapturaDataSerial(16,19);//4
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
    debugSerial.print("0x");
    if (byteArrayLectura[i] < 0x10)
        debugSerial.print("0");
    debugSerial.print(byteArrayLectura[i], HEX);
    debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 4;

debugSerial.println("VOLTAJE FASE B");
longitud = 27;
dumpByteArray(CMD17, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,19);

```

```

//trx = CapturaDataSerial(16,19);//4
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
debugSerial.print("0x");
if (byteArrayLectura[i] < 0x10)
    debugSerial.print("0");
debugSerial.print(byteArrayLectura[i], HEX);
debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 4;

debugSerial.println("FACTOR DE POTENCIA");
longitud = 27;
dumpByteArray(CMD18, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,19);
//trx = CapturaDataSerial(16,19);//4
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{

```

```

debugSerial.print("0x");
if (byteArrayLectura[i] < 0x10)
  debugSerial.print("0");
debugSerial.print(byteArrayLectura[i], HEX);
debugSerial.print(" ");
}
debugSerial.println(" ");

delay(2000);

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 4;

debugSerial.println("FRECUENCIA");
longitud = 27;
dumpByteArray(CMD19, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,19);
//trx = CapturaDataSerial(16,19);//4
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);
delay(2000);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
  debugSerial.print("0x");
  if (byteArrayLectura[i] < 0x10)
    debugSerial.print("0");
  debugSerial.print(byteArrayLectura[i], HEX);
  debugSerial.print(" ");
}

```

```

debugSerial.println(" ");

indexBuffer = indexBuffer + cantidadLectura;
cantidadLectura = 4;

debugSerial.println("POTENCIA ACTIVA INSTANTANEA");
longitud = 27;
dumpByteArray(CMD20, longitud);
trx = CapturaDataSerial(byteArrayLectura, indexBuffer, 16,19);
//trx = CapturaDataSerial(16,19);//4
respuestaLectura += trx;
debugSerial.println("TRX ES: " + trx);

debugSerial.println("Buffer: ");
for (int i = indexBuffer; i < (indexBuffer + cantidadLectura) ; i++)
{
debugSerial.print("0x");
if (byteArrayLectura[i] < 0x10)
debugSerial.print("0");
debugSerial.print(byteArrayLectura[i], HEX);
debugSerial.print(" ");
}
debugSerial.println(" ");

respuestaLectura.toUpperCase();
return respuestaLectura;
}

void Reconexion()
{
int longitud = 17;

```

```
byte byteArray[longitud] = {0};
hexCharacterStringToBytes(byteArray, HANDSHAKE1);
dumpByteArray(byteArray, longitud);
delay(500);

delay(1500);

longitud = 6;
byteArray[longitud] = {0};
hexCharacterStringToBytes(byteArray, HANDSHAKE2);
dumpByteArray(byteArray, longitud);
delay(500);

delay(1500);

longitud = 9;
byteArray[longitud] = {0};
hexCharacterStringToBytes(byteArray, HANDSHAKE3);
dumpByteArray(byteArray, longitud);
delay(500);
delay(1500);

longitud = 70;
dumpByteArray(HANDSHAKE4, longitud);
delay(500);
delay(1500);
delay(1500);

longitud = 27;
dumpByteArray(METERNUMBER0, longitud);
delay(500);
```

```
delay(1500);
```

```
longitud = 29;
```

```
dumpByteArray(RECONNECT, longitud);
```

```
delay(500);
```

```
delay(1500);
```

```
longitud = 9;
```

```
byteArray[longitud] = {0};
```

```
hexCharacterStringToBytes(byteArray, HANDSHAKEBYE0);
```

```
dumpByteArray(byteArray, longitud);
```

```
delay(500);
```

```
delay(1500);
```

```
longitud = 9;
```

```
byteArray[longitud] = {0};
```

```
hexCharacterStringToBytes(byteArray, HANDSHAKEBYE1);
```

```
dumpByteArray(byteArray, longitud);
```

```
delay(500);
```

```
}
```

```
void Corte()
```

```
{
```

```
int longitud = 17;
```

```
byte byteArray[longitud] = {0};
```

```
hexCharacterStringToBytes(byteArray, HANDSHAKE1);
```

```
dumpByteArray(byteArray, longitud);
```

```
delay(500);
```

```
delay(1500);
```

```
longitud = 6;
```



```
byteArray[longitud] = {0};  
hexCharacterStringToBytes(byteArray, HANDSHAKE2);  
dumpByteArray(byteArray, longitud);  
delay(500);  
delay(1500);
```

```
longitud = 9;  
byteArray[longitud] = {0};  
hexCharacterStringToBytes(byteArray, HANDSHAKE3);  
dumpByteArray(byteArray, longitud);  
delay(500);  
delay(1500);
```

```
longitud = 70;  
dumpByteArray(HANDSHAKE4, longitud);  
delay(500);
```

```
delay(1500);  
delay(1500);
```

```
longitud = 27;  
dumpByteArray(METERNUMBER0, longitud);  
delay(500);  
delay(1500);
```

```
longitud = 29;  
dumpByteArray(CUT, longitud);  
delay(500);  
delay(1500);
```

```
longitud = 9;
```

```

byteArray[longitud] = {0};
hexCharacterStringToBytes(byteArray, HANDSHAKEBYE0);
dumpByteArray(byteArray, longitud);
delay(500);
delay(1500);

longitud = 9;
byteArray[longitud] = {0};
hexCharacterStringToBytes(byteArray, HANDSHAKEBYE1);
dumpByteArray(byteArray, longitud);
delay(500);
}

void setup() {
  LoRaSerial.begin(57600);
  debugSerial.begin(9600);
  medidorSerial.begin(9600);
  pinMode(apertura, INPUT_PULLUP);
  pinMode(reseteo_LoRa, OUTPUT);
  pinMode(_running, OUTPUT);
  pinMode(caja, OUTPUT);
  digitalWrite(reseteo_LoRa, HIGH);
  delay(2000);
  digitalWrite(reseteo_LoRa, LOW);

  attachInterrupt(0, ejecutarApertura, CHANGE);

  // Wait a maximum of 10s for Serial Monitor
  while (!debugSerial && millis() < 10000)
    ;
}

```

```

ttn.onMessage(message);
ttn.showStatus();
ttn.join(appEui, appKey);

digitalWrite(_running, LOW);
}

void ejecutarApertura() {
  apertura = 1;
}

const byte MaxByteArraySize = 27;

void hexCharacterStringToBytes(byte *byteArray, const char *hexString)
{
  bool oddLength = strlen(hexString) & 1;

  byte currentByte = 0;
  byte byteIndex = 0;

  for (byte charIndex = 0; charIndex < strlen(hexString); charIndex++)
  {
    bool oddCharIndex = charIndex & 1;

    if (oddLength)
    {
      // If the length is odd
      if (oddCharIndex)
      {
        // odd characters go in high nibble
        currentByte = nibble(hexString[charIndex]) << 4;

```

```

    }
    else
    {
        // Even characters go into low nibble
        currentByte |= nibble(hexString[charIndex]);
        byteArray[byteIndex++] = currentByte;
        currentByte = 0;
    }
}
else
{
    // If the length is even
    if(!oddCharIndex)
    {
        // Odd characters go into the high nibble
        currentByte = nibble(hexString[charIndex]) << 4;
    }
    else
    {
        // Odd characters go into low nibble
        currentByte |= nibble(hexString[charIndex]);
        byteArray[byteIndex++] = currentByte;
        currentByte = 0;
    }
}
}
}

void dumpByteArray(const byte * byteArray, const byte arraySize)
{

```

```

for (int i = 0; i < arraySize; i++)
{
    debugSerial.print("0x");
    if (byteArray[i] < 0x10)
        debugSerial.print("0");
    debugSerial.print(byteArray[i], HEX);
    debugSerial.print(", ");
}

medidorSerial.write(byteArray,arraySize);
debugSerial.println();
}

byte nibble(char c)
{
    if (c >= '0' && c <= '9')
        return c - '0';

    if (c >= 'a' && c <= 'f')
        return c - 'a' + 10;

    if (c >= 'A' && c <= 'F')
        return c - 'A' + 10;

    return 0; // Not a valid hexadecimal character
}

void loop() {
    if (apertura == 1)
    {
        digitalWrite(caja, LOW);
    }
}

```

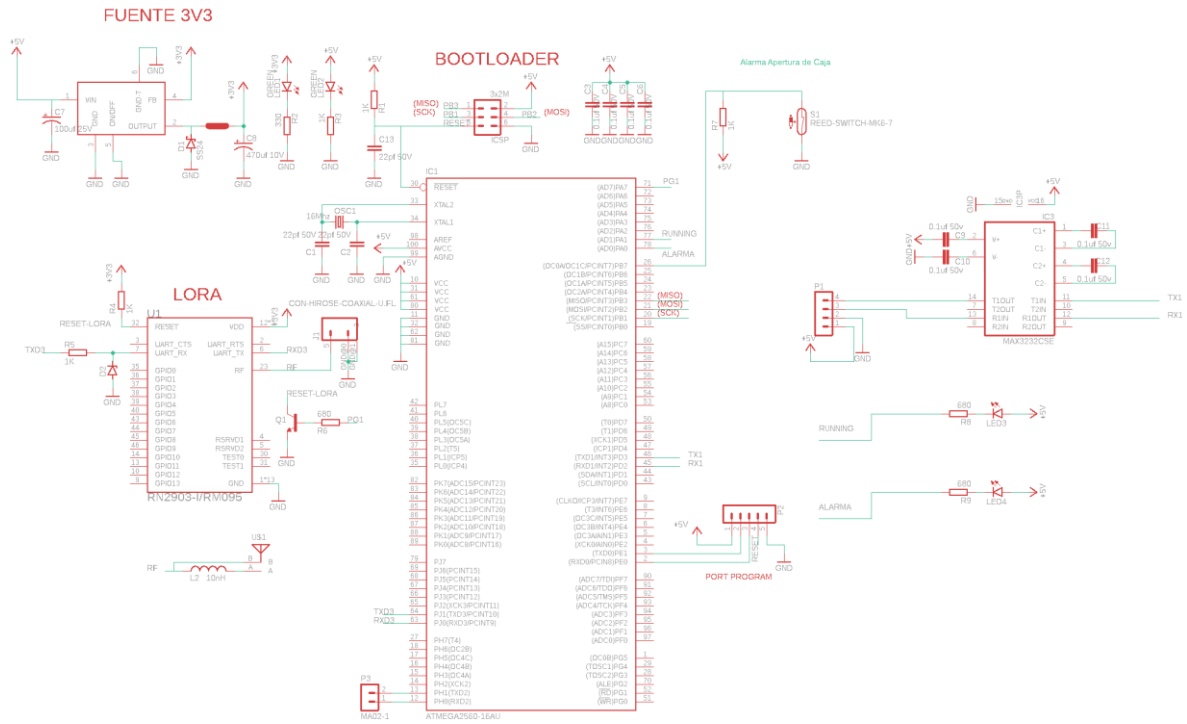
```
Serial.println("APERTURA: " + apertura);
apertura = 0;
byte byteArrayAck[3] = {0};
byteArrayAck[0] = 0;
byteArrayAck[1] = 4;
byteArrayAck[2] = 0;
EnviarLoRa(byteArrayAck, 3);
}

debugSerial.println("**LOOP**");
ttn.poll();

delay(10000);
}
```

ANEXO 2:

Diagrama esquemático de Tarjeta de medidor residencial con Modulo de Comunicación LoRa Wan.



ANEXO 3:

Decodificación de trama de datos enviada por el medidor a la plataforma The Thing Network

TRAMA ENVIADA AL MEDIDOR CON PETICION DE LECTURA:

77 01 00

TRAMA ENVIADA POR EL MEDIDOR:

77010235333730313730323030303607E6091D040B1E330000000000A6C4F8
00000000008283540000000000D7F8AC00008C7400002C600000082A00002E1800002F
440000032A00001772000030C4

DECODIFICACION DE TRAMA RECIBIDA DE ACUERDO CON FORMATO DE LOS CAMPOS DE ESTA.

CABECERA:

77 01 02:

Byte de sincronismo: 77

Byte de código de lectura: 01

Byte de código de Error: 02

NÚMERO DE SERIE DEL MEDIDOR (FORMATO HEXADECIMAL)

353337303137303230303036: 537017020006

FECHA (FORMATO HEXADECIMAL): AÑO, MES, DÍA.

07E6091D

07E6: 2022

09: Septiembre mes del año

1D: 29 día del mes.

DIA CON HORA DEL MEDIDOR

040B1E33

04: Dia de la semana jueves (Lunes 1, Martes 2, Miércoles 3, Jueves 4, Viernes 5, Sábado 6 y Domingo 7).

0B: Hora. 11 horas

1E: Minutos 30 minutos

33: Segundos 51 segundos.

ENERGIA TOTAL ACTIVA KWH (FORMATO HEXADECIMAL /1000)

000000000A6C4F8: $10929400 / 1000 = 10929.4$ kWh.

ENERGIA TOTAL REACTIVA KVARH (FORMATO HEXADECIMAL /1000)

000000000828354: $8553300 / 1000 = 8553.3$ kVARh

ENERGIA TOTAL APARENTE KVAH (FORMATO HEXADECIMAL / 1000)

000000000D7F8AC: $14153900 / 1000 = 14153.9$

DEMANDA ACTIVA TOTAL KW (FORMATO HEXADECIMAL / 10000)

00008C74: $35956 / 10000 = 3.5956$

CORRIENTE FASE A AMP (FORMATO HEXADECIMAL / 1000)

00002C60: $11360 / 1000 = 11.36$

CORRIENTE FASE B AMP (FORMATO HEXADECIMAL / 1000)

0000082A: $2090 / 1000 = 2.09$

VOLTAJE FASE A V(FORMATO HEXADECIMAL / 100)

00002E18: $11800 / 1000 = 118$

VOLTAJE FASE B V(FORMATO HEXADECIMAL / 100)

00002F44: $12100 / 100 = 121$

FACTOR DE POTENCIA (FORMATO HEXADECIMAL / 1000)

0000032A: $810 / 1000 = 0.81$

FRECUENCIA HZ (FORMATO HEXADECIMAL / 1000)

00001772: $6002 / 1000 = 60.02$

POTENCIA ACTIVA INSTANTANEA KW (FORMATO HEXADECIMAL / 10000)

000030C4: $12484 / 10000 = 1.2484$