



Ingeniería Electrónica

Proyecto Final
Medidor de Ruidos Ambientales IOT (MAG)

Autor/es

Adolfo Apaza
Miguel Ángel Tomas Llorens
Gianfranco Pennisi

Director o Tutor

Mg. Guillermo R. Friedrich
Ing. Adrián H Laiuppa

Bahía Blanca | 9 de agosto de 2024

Contenido

Resumen.....	3
Introducción.....	4
Objetivo.....	5
Marco Teórico.....	6
Diagrama de bloques.....	9
Hardware.....	12
Microcontroladores.....	12
Micrófono.....	14
Comunicación entre dispositivos.....	18
BMP280.....	23
RTC.....	25
Diseño del Esquemático.....	26
Diseño de PCB.....	28
Tabla de conexiones.....	29
Nodos:.....	33
Gateway.....	41
Pruebas de campo.....	43
Micrófono.....	43
Transmisión.....	47
Dispositivo completo.....	49
Mejoras.....	50
Conclusión.....	53
Bibliografía.....	54

Figuras

Figura 1: Diagrama de bloques. Imagen de elaboración propia.....	10
Figura 2: Microcontrolador ESP32. Imagen extraída de [5]	13
Figura 3: Micrófono ICS43434. Imagen extraída de [6].....	17
Figura 4: Esquemático Micrófono ICS43434. Imagen Extraída de [6].....	18
Figura 5: Dispositivo LoRa SX1276. Imagen extraída de [7].....	23
Figura 6: Modulo BMP280. Imagen extraída de [8]	24
Figura 7: Modulo RTC DS3231. Imagen extraída de [9].....	26
Figura 8: Diseño Esquemático Gateway. Imagen de elaboración propia.....	26
Figura 9: Diseño Esquemático Nodo Placa Uno. Imagen de elaboración propia.....	27
Figura 10: Diseño esquemático Nodo Placa Dos. Imagen de elaboración propia.....	27
Figura 11: PCB Gateway. Imagen de elaboración propia.	28
Figura 12: PCB Nodo Uno. Imagen de elaboración propia.....	28
Figura 13: PCB Nodo Dos. Imagen de elaboración propia.....	29
Figura 14: Ensayo de comparación con el dispositivo homologado. Imagen de elaboración propia.....	43
Figura 15: Comparación de las muestras obtenidas en un ambiente ruidoso. Imagen de elaboración propia.....	44
Figura 16: Comparación de las muestras obtenidas en un ambiente controlado. Imagen de elaboración propia.	45
Figura 17: Dashboard realizada en Azure Central IoT. Imagen extraída de [10].....	50

Resumen

El objetivo de este proyecto es visibilizar un fenómeno cotidiano del que muchas veces no somos conscientes: los niveles de ruido ambiental en las ciudades y su impacto en la población. Este proyecto está dirigido a agentes gubernamentales, quienes podrán implementarlo en las ciudades para ofrecer a los ciudadanos la posibilidad de conocer los niveles acústicos a los que están expuestos. A partir de estos datos, se busca generar conciencia y promover medidas que prevengan la exposición a sonidos que puedan afectar la calidad de vida de las personas.

El desarrollo planteado en este proyecto propone un dispositivo que puede producirse en masa de manera económica, lo que permitirá abarcar más área al generar un mapa acústico de las ciudades. Con solo añadir más dispositivos a la red, podrán comunicarse entre sí. Posteriormente, los usuarios, tras recibir la capacitación correspondiente, podrán agregar nuevos dispositivos a la página web destinada a generar las métricas necesarias.

Aunque el proyecto está orientado a los gobiernos interesados en implementarlo en sus ciudades, los usuarios finales serán los ciudadanos, quienes podrán acceder a información detallada sobre cómo la contaminación acústica afecta su vida cotidiana. De este modo, podrán tomar las precauciones necesarias para proteger su calidad de vida. El mapa de ruido, que será de dominio público, mostrará los niveles promedio de sonido en las distintas zonas donde estén ubicados los dispositivos, así como los horarios pico en los que se generan las emisiones más fuertes de ruido.

Palabras Claves: Medición de ruido – Mapa de ruido – IoT – Ciudades Inteligentes – Ruidos Ambientales

Introducción

La contaminación acústica es un problema creciente en las áreas urbanas, donde la densidad de población y la actividad comercial generan niveles de ruido que pueden afectar negativamente la calidad de vida de los habitantes. A uno de los miembros de nuestro equipo le afectó el problema por la proximidad de su vivienda con un local nocturno de música bailable. Esta situación no solo afectaba su descanso, sino también el bienestar de los vecinos aledaños.

A raíz de esta experiencia personal, surgió la idea de desarrollar un sistema de medición de ruidos ambientales que permitiera monitorear y registrar los niveles de ruido en tiempo real. Este proyecto tiene como objetivo proporcionar una herramienta accesible y eficiente, utilizando la tecnología IoT (Internet de las Cosas) con dispositivos ESP32.

Nuestro enfoque inicial fue investigar las regulaciones existentes en torno a la contaminación acústica y las medidas que los gobiernos implementan para mitigar este problema. Descubrimos que, aunque existen normativas claras, la aplicación y el monitoreo continuo de los niveles de sonido son desafíos persistentes. Por tanto, un sistema automatizado de bajo costo y fácil implementación podría ser una solución valiosa tanto para ciudadanos como para autoridades reguladoras.

El dispositivo que desarrollamos está diseñado para ser instalado en diferentes puntos de una zona afectada por la contaminación acústica, proporcionando datos precisos y en tiempo real que pueden ser analizados para tomar decisiones informadas. En este informe detallaremos el diseño, implementación y resultados obtenidos con nuestro sistema IoT, destacando su potencial impacto en la mejora de la calidad de vida en áreas urbanas afectadas por la contaminación acústica.

Objetivo

Objetivo General

El objetivo principal de este proyecto es desarrollar un sistema integral de medición de ruidos ambientales utilizando tecnología IoT, específicamente con dispositivos ESP32, que permita monitorear y registrar los niveles de sonido en tiempo real. Este sistema busca ser una herramienta accesible, económica y eficiente para controlar la contaminación acústica en áreas urbanas, mejorando así la calidad de vida de sus habitantes y facilitando la toma de decisiones informadas por parte de las autoridades reguladoras.

Objetivos Específicos

Uno de los primeros objetivos específicos del proyecto es diseñar y desarrollar un dispositivo de medición que esté basado en el microcontrolador ESP32. Este dispositivo debe ser capaz de medir los niveles de contaminación acústica con alta precisión y fiabilidad, utilizando sensores de sonido adecuados que funcionen eficazmente en diversas condiciones ambientales.

La implementación de la tecnología IoT es otro aspecto crucial del proyecto. Es necesario desarrollar una red de dispositivos interconectados que permitan la transmisión de datos, previamente recopilados, en tiempo real. Asegurar la conectividad y la estabilidad de esta red es esencial para garantizar un monitoreo continuo y fiable de los niveles de ruido.

Paralelamente, es fundamental hacer uso de una plataforma robusta de análisis de datos. Esta plataforma debe ser capaz de almacenar, visualizar y analizar los datos recopilados por los dispositivos de medición. Se debe emplear herramientas de visualización que faciliten la interpretación de estos datos y la identificación de patrones de ruido, permitiendo así una evaluación precisa de la situación acústica en las áreas monitoreadas.

La evaluación y validación del sistema desarrollado es otro objetivo específico importante. Se llevarán a cabo pruebas tanto en entornos controlados como en situaciones reales para validar la precisión y fiabilidad del sistema de medición. Los datos obtenidos serán comparados con los niveles de ruido permitidos según las regulaciones locales para evaluar su cumplimiento y eficacia.

Identificar posibles mejoras en el diseño y funcionamiento del sistema es también un objetivo clave. Basado en los resultados obtenidos durante las pruebas, se propondrán e

implementarán mejoras que optimicen el rendimiento del sistema. Además, se explorará la posibilidad de expandir el uso del sistema a otras áreas urbanas afectadas por la contaminación acústica, ampliando así su impacto positivo.

Finalmente, promover la conciencia sobre la contaminación acústica es un objetivo esencial del proyecto, contribuyendo así a la creación de entornos urbanos más saludables y sostenibles.

Marco Teórico

Contaminación Acústica y su Impacto

La contaminación acústica se refiere a niveles excesivos de ruido en el entorno que pueden causar efectos adversos en la salud y el bienestar humano. Los principales efectos de la contaminación acústica incluyen trastornos del sueño, problemas cardiovasculares, estrés y ansiedad, y disminución de la capacidad cognitiva. La exposición prolongada puede dificultar el sueño, provocando insomnio y disminuyendo la calidad del sueño, lo que puede llevar a fatiga, irritabilidad y problemas de concentración. Además, estudios han demostrado una correlación entre lo detallado anteriormente y el aumento en la presión arterial, así como un mayor riesgo de enfermedades cardíacas y accidentes cerebrovasculares. La exposición constante puede causar estrés psicológico, lo que a su vez puede desencadenar o agravar trastornos de ansiedad. En ambientes ruidosos, la capacidad de concentración y el rendimiento cognitivo pueden verse reducidos, afectando negativamente el desempeño académico y laboral.

Propagación del Sonido y Factores Ambientales

El sonido se propaga en forma de ondas mecánicas longitudinales a través de un medio, generalmente el aire. La velocidad y la intensidad del sonido pueden verse afectadas por varias condiciones ambientales. La velocidad del sonido en el aire está dada por la fórmula:

$$v = \sqrt{\frac{\gamma * R * T}{M}}$$

donde v es la velocidad del sonido, γ es la razón de capacidades caloríficas (aproximadamente 1.4 para el aire), R es la constante de los gases ideales (8.314 J/(mol·K)), T es la temperatura absoluta en kelvin, y M es la masa molar del aire (aproximadamente 0.029 kg/mol). La velocidad del sonido aumenta con la temperatura del aire, aproximadamente 0.6 m/s por cada grado Celsius de incremento. La velocidad del sonido también aumenta con la humedad, ya que el aire húmedo tiene menor densidad que el aire seco. El viento puede influir en la propagación del sonido, ya que la velocidad efectiva del sonido en la dirección del viento es la suma de la velocidad del sonido en el aire y la componente del viento en la misma dirección. La presencia de obstáculos como edificios, vegetación y terrenos irregulares puede reflejar, absorber o dispersar las ondas sonoras, afectando su propagación.

Medición del Ruido Ambiental

La evaluación del ruido ambiental se lleva a cabo utilizando sonómetros, dispositivos que registran los niveles de presión sonora en decibelios (dB). La presión sonora p se relaciona con el nivel de presión sonora L_p mediante la fórmula:

$$L_p = 20 \log_{10} \left(\frac{p}{p_0} \right)$$

donde L_p es el nivel de presión sonora en decibelios (dB), p es la presión sonora registrada, y p_0 es la presión de referencia, típicamente $20 \mu Pa$. Según la norma IEC 61672-1 [1], los sonómetros se clasifican en Clase 1 y Clase 2. Los de Clase 1 ofrecen mayor precisión y se utilizan para verificaciones profesionales, mientras que los de Clase 2 son adecuados para aplicaciones generales. Las valoraciones del ruido pueden ponderarse para reflejar mejor la sensibilidad del oído humano a diferentes frecuencias. La ponderación A (dBA) se emplea para evaluar el entorno sonoro general, ya que simula la respuesta del oído humano a niveles bajos y medios de presión sonora. La ponderación C (dBC) se utiliza para valorar el impacto de sonidos de alta intensidad, mientras que la ponderación Z (dBZ) proporciona un registro lineal de todas las frecuencias.

Normas IRAM e ISO para la Medición de Ruido

Las normas IRAM (Instituto Argentino de Normalización y Certificación) e ISO (Organización Internacional de Normalización) establecen directrices para la evaluación y análisis de la contaminación acústica. La norma IRAM 4062 [2] define los métodos para medir el ruido en entornos ocupacionales y residenciales en Argentina, proporcionando pautas sobre el uso de sonómetros, la selección de puntos de monitoreo y la interpretación de los datos obtenidos.

La norma ISO 1996-1[3] establece los métodos para la descripción, cuantificación y evaluación del ruido ambiental, incluyendo directrices para la ponderación de frecuencias y la selección de puntos de registro. Esta norma especifica que las capturas de datos deben realizarse a una altura de 1.2 a 1.5 metros sobre el suelo y a una distancia mínima de 3.5 metros de cualquier superficie reflectante. La norma ISO 9613-2 [4] proporciona un método para calcular la atenuación del sonido durante su propagación en exteriores, considerando factores como la absorción atmosférica, la vegetación y otras características del terreno. La fórmula para la atenuación del sonido A es:

$$A = A_{atm} + A_{gr} + A_{bar} + A_{veg}$$

donde A_{atm} es la atenuación por absorción atmosférica, A_{gr} es la atenuación por efectos del suelo, A_{bar} es la atenuación por barreras, y A_{veg} es la atenuación por vegetación.

Implementación de Tecnología IoT en la Medición de Ruido

La tecnología IoT permite la creación de sistemas de monitoreo de ruido ambiental en tiempo real. Un dispositivo basado en ESP32 es ideal para esta tarea debido a su conectividad Wi-Fi, bajo consumo de energía, capacidad de procesamiento, y costo asequible. Además, su doble núcleo, soporte para múltiples interfaces de comunicación y amplia comunidad de desarrollo lo hacen una opción versátil y eficiente.

Un sistema IoT de monitoreo de contaminación acústica incluye sensores de sonido, dispositivos que capturan niveles de presión sonora y los convierten en señales eléctricas. Los sensores electret y MEMS (Microelectromechanical Systems) son comúnmente utilizados debido a su precisión y tamaño compacto. El microcontrolador ESP32 procesa las señales del sensor de sonido y transmite los datos a través de una red Wi-Fi. Los datos

recopilados se envían a una plataforma central donde se almacenan, visualizan y analizan. Herramientas de índole como SaaS (Software as a Service) pueden ser utilizadas para la visualización de datos en tiempo real, como también control de los dispositivos conectados.

Aplicación y Beneficios del Sistema

La implementación de un sistema de medición de ruido ambiental basado en IoT ofrece múltiples beneficios. Permite la vigilancia constante de los niveles de ruido en diversas ubicaciones, proporcionando datos en tiempo real que son cruciales para la toma de decisiones rápidas y efectivas. Los datos recopilados pueden ser accesibles desde cualquier lugar y en cualquier momento a través de plataformas en línea, facilitando el acceso y la gestión de la información.

La capacidad de analizar detalladamente los datos recopilados y generar informes personalizados permite a las autoridades reguladoras y a los ciudadanos comprender mejor la situación acústica y actuar en consecuencia. Al difundir los resultados, se puede aumentar la conciencia pública sobre los efectos negativos de la contaminación acústica, promoviendo acciones comunitarias y políticas más estrictas para su mitigación. Los datos precisos y en tiempo real permiten identificar rápidamente las fuentes de ruido y planificar intervenciones efectivas, como la instalación de barreras acústicas, la regulación de horarios de funcionamiento de establecimientos ruidosos y la promoción de prácticas de construcción más silenciosas.

A partir de lo planteado anteriormente damos un marco teórico acerca de la factibilidad y la funcionalidad del proyecto hacia el lector de este informe y así poder comenzar con la parte técnica a la cual está enfocada el mismo, en dicha parte comenzaremos explicando la solución a grandes rasgos y entraremos en detalle de los componentes y algoritmos utilizamos además de las aplicaciones para la visualización y almacenado de los datos.

Diagrama de bloques

En esta instancia debemos realizar una discrepancia ya que los dispositivos funcionan de manera continua y al tener dos tipos distintos de dispositivos y aplicaciones involucradas cada una tiene una funcionalidad específica, pero todo debe trabajar en conjunto para que la finalidad completa del proyecto sea eficiente.

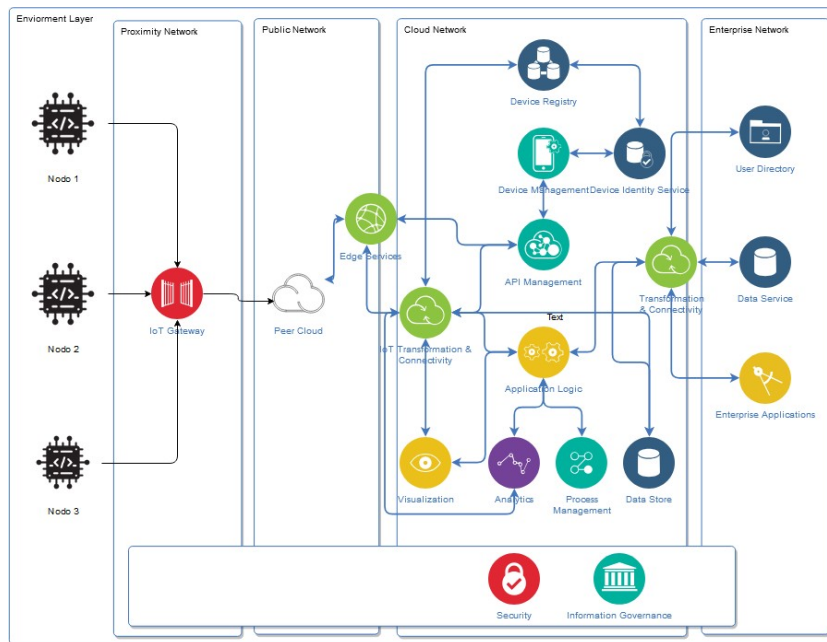


Figura 1: Diagrama de bloques. Imagen de elaboración propia.

Visto de esta manera el diagrama de bloques resulta confuso, por lo que lo vamos a separar en tres etapas:

En la primera etapa, los nodos son responsables de capturar los valores de potencia de ruido en la vía pública, siguiendo un conjunto de tareas secuenciales. Comienzan encendiendo el dispositivo e inicializando el micrófono, el cual durante un minuto recolecta las diferentes muestras. A estas muestras se les aplica un filtro de ponderación A y un filtro IIR para mejorar la calidad de la señal. Una vez recopiladas todas las muestras, se calcula un promedio de las mismas y se formatean los datos para su posterior envío al Gateway.

Como segunda tarea, se realiza un sensado de variables como la temperatura, presión, altura del dispositivo y el tiempo de captura. Estas variables ayudan a mejorar la precisión de los valores obtenidos por el micrófono. La altura se utiliza como medida de seguridad, asegurando que los dispositivos se mantengan a una altura específica para un registro eficiente. Si se detecta una fluctuación que pueda generar inconsistencias, se realizan las correcciones necesarias en el posicionamiento del dispositivo. Finalmente, se registra la hora en la que se tomó la muestra. Una vez completada la adquisición de los distintos valores, se pasa a la siguiente tarea.

La tercera tarea del dispositivo es la transmisión de los datos recolectados en las dos tareas anteriores. Se genera una cadena de caracteres con un código de verificación del mensaje y, utilizando la tecnología LoRa, se envían los datos al Gateway.

Como cuarta y última tarea, los nodos entran en modo de bajo consumo, ya que, al estar alimentados por batería, es fundamental gestionar inteligentemente el uso de energía para la longevidad del proyecto. Por lo tanto, se estableció desde el inicio que los dispositivos no estarían recolectando datos de manera continua, sino que su intervalo de adquisición sería de 1 minuto cada 10 minutos.

Pasamos al próximo dispositivo el cual es el Gateway, el mismo es el encargado de recibir los datos de los distintos nodos, cabe aclarar que en este proyecto se definió que la cantidad de nodos es igual a dos pero existe la posibilidad de expansión en un futuro, una vez aclarado esto también hay que hacer la aclaración que en caso de querer hacer una expansión del proyecto se tiene que tener en cuenta que se deberá también analizar la cantidad de Gateways que se requieran ya que por el volumen de datos y la cantidad de dispositivos apuntando hacia uno solo no será capaz que poder alojar todos los nodos que se desee.

El Gateway es el encargado de recibir los datos que son transmitidos a través de los nodos, una vez recibidos los mismos este se deberá encargar primero de analizar los datos mediante una verificación de la veracidad y la no corrupción de estos. En caso de que la verificación sea fallida este descartará los datos y esperará a una próxima recepción de estos. En caso positivo será el encargado el primer paso es realizar y verificar la conexión a internet mediante la tecnología Wifi, esta se supone que será provista por la entidad que desee realizar el proyecto ya que tiene que ser una red segura para que no exista interferencia a la hora del envío y recepción de datos. Posterior a esto se deberá realizar la conexión con el software encargado del análisis y visualización de datos. Una vez finalizado el envío de los datos hacia la nube el Gateway quedará a la espera de la próxima recepción de los datos.

Como último bloque del proyecto se encuentra el software de análisis y visualización de datos, en este caso se optó por la utilización de un servicio provisto por Microsoft, conocido como Azure IoT Central, el cual al ser un servicio de pago por suscripción nos brinda la posibilidad de poder generar distintas métricas y gráficos correspondiente a nuestras necesidades a un precio acorde al mismo. Otra facilidad que nos provee es la

posibilidad de poder tener distintos niveles de seguridad y escalabilidad a la hora de querer expandir el proyecto con más dispositivos.

Hardware

Microcontroladores

Características del Microcontrolador ESP32

Conectividad Integrada

El ESP32 ofrece conectividad Wi-Fi y Bluetooth (clásico y BLE) integradas, lo que lo hace ideal para aplicaciones IoT que requieren comunicación inalámbrica. Esta conectividad integrada elimina la necesidad de módulos adicionales, simplificando el diseño del hardware y reduciendo los costos.

Alto Rendimiento

El ESP32 cuenta con un procesador dual-core Xtensa LX6 que puede funcionar a una velocidad de hasta 240 MHz. Esta capacidad de procesamiento permite manejar tareas complejas y múltiples periféricos simultáneamente.

Bajo Consumo de Energía

El ESP32 está diseñado para ser eficiente en términos de consumo de energía. Ofrece varios modos de bajo consumo, como el modo de suspensión profunda, que permite reducir significativamente el consumo de energía cuando el dispositivo no está en uso activo. Esto es crucial para dispositivos que funcionan con baterías y necesitan maximizar su duración.

Soporte para Periféricos

El ESP32 soporta una amplia gama de periféricos, incluyendo ADC, DAC, PWM, I2C, SPI, UART, I2S y más. Esta versatilidad permite a los desarrolladores conectar diversos sensores y actuadores, ampliando las posibilidades de aplicación del microcontrolador.

Capacidades de Seguridad

El ESP32 incluye características de seguridad avanzadas, como cifrado de hardware, arranque seguro, y almacenamiento seguro de claves. Estas características son esenciales

para proteger los datos y la comunicación en aplicaciones donde la seguridad es una preocupación crítica.

Amplia Comunidad y Soporte

El ESP32 cuenta con una amplia comunidad de desarrolladores y una gran cantidad de recursos disponibles, incluyendo bibliotecas, ejemplos de código, y documentación. Esto facilita el desarrollo y la resolución de problemas, acelerando el tiempo de desarrollo y mejorando la fiabilidad del producto final.

Tamaño Compacto y Fácil Integración

El tamaño compacto del ESP32 y su diseño de fácil integración permiten su uso en una variedad de proyectos, desde dispositivos portátiles hasta sistemas de monitoreo ambiental. Su capacidad para manejar múltiples periféricos y conexiones inalámbricas lo hace extremadamente versátil.

Conectividad y Comunicación

Para un dispositivo de medición, la capacidad de conectarse a redes Wi-Fi para transmitir datos es crucial. El ESP32, con su conectividad Wi-Fi integrada, facilita la transmisión de datos a servidores o plataformas en la nube para su análisis y almacenamiento.

Procesamiento de Señales

La alta capacidad de procesamiento del ESP32 permite manejar los datos de audio capturados por el micrófono ICS43434, incluyendo tareas de filtrado y procesamiento de señales. Esto asegura que el dispositivo pueda realizar análisis en tiempo real y tomar decisiones basadas en los datos capturados.



Figura 2: Microcontrolador ESP32. Imagen extraída de [5]

Micrófono

La elección del micrófono adecuado para un dispositivo de medición de ruidos ambientales es crucial para garantizar precisión, fiabilidad y eficiencia. En el mercado existen diversos tipos de micrófonos, cada uno con sus características específicas y aplicaciones ideales.

Micrófonos MEMS

Los micrófonos MEMS utilizan tecnologías de microelectromecánica para construir dispositivos extremadamente pequeños y precisos. Estos micrófonos consisten en un diafragma y un circuito de procesamiento integrados en un solo chip. Son conocidos por su tamaño compacto, alta fiabilidad, bajo consumo de energía y buena respuesta de frecuencia. Los micrófonos MEMS son ideales para aplicaciones en dispositivos portátiles, teléfonos inteligentes, y especialmente en dispositivos IoT.

Tamaño y Peso

Una de las principales ventajas de los micrófonos MEMS es su tamaño extremadamente pequeño y su peso ligero. Esto permite su integración en dispositivos compactos y portátiles, lo que es crucial para los dispositivos que deben ser discretos y fáciles de instalar en diferentes entornos.

Consumo de Energía

Los micrófonos MEMS tienen un bajo consumo de energía en comparación con otros tipos de micrófonos, como los de condensador. Esto es vital para los dispositivos que a menudo funcionan con baterías y necesitan maximizar la vida útil de estas para reducir la necesidad de reemplazos frecuentes o recargas.

Resistencia y Durabilidad

A pesar de su tamaño, los micrófonos MEMS son sorprendentemente robustos y pueden soportar condiciones ambientales adversas. Son menos susceptibles a daños por polvo, humedad y vibraciones, lo que los hace ideales para aplicaciones en exteriores y entornos industriales.

Sensibilidad y Precisión

Los micrófonos MEMS ofrecen una buena respuesta de frecuencia y alta sensibilidad, lo que permite una medición precisa de los niveles de ruido ambiental. Esta precisión es

crucial para aplicaciones de monitoreo donde se necesita una captura fiel de los niveles de sonido para análisis y toma de decisiones.

Facilidad de Integración

Los micrófonos MEMS son fáciles de integrar con otros componentes electrónicos y sistemas de procesamiento de señales. Esto facilita su uso en dispositivos donde la miniaturización y la integración eficiente son esenciales para reducir costos y complejidad de diseño.

Coste

Los micrófonos MEMS suelen ser más económicos en producción a gran escala debido a las técnicas de fabricación de semiconductores. Esto es una ventaja significativa cuando se desarrollan dispositivos que deben ser rentables y accesibles para un amplio mercado.

La decisión de implementar un micrófono tipo MEMS en un dispositivo de medición de ruidos ambientales se basa en una combinación de factores críticos: tamaño compacto, bajo consumo de energía, resistencia, alta sensibilidad y precisión, facilidad de integración y coste efectivo. Estos micrófonos proporcionan una solución óptima que satisface las necesidades técnicas y económicas de los dispositivos modernos, asegurando un rendimiento fiable y duradero en diversas aplicaciones.

Una vez realizada esta explicación de los micrófonos tipo MEMS concluimos que era la mejor opción para realizar este proyecto. Por eso nos decidimos por utilizar el dispositivo ICS43434. En los siguientes párrafos daremos más detalle acerca de este dispositivo.

Características del ICS43434:

Entrada Digital I2S

El ICS43434 utiliza una interfaz I2S (Inter-IC Sound), que permite la transmisión de datos de audio en formato digital directamente a un procesador o microcontrolador sin necesidad de conversión analógica-digital externa. Esto simplifica el diseño del circuito y reduce la interferencia introducida por conversiones adicionales.

Baja Distorsión y Alta Relación Señal-Ruido (SNR)

Este micrófono proporciona una baja distorsión y una alta relación señal-ruido (SNR) de 65 dB, lo que garantiza una captura clara y precisa del sonido ambiental. Esto es crucial para aplicaciones que requieren una alta fidelidad.

Amplia Respuesta de Frecuencia

El ICS43434 tiene una respuesta de frecuencia de 50 Hz a 20 kHz, lo que le permite capturar una amplia gama de sonidos, desde bajos profundos hasta agudos nítidos. Esta característica es esencial para el monitoreo de ruido ambiental, donde es importante capturar todas las frecuencias relevantes.

Reducción de Ruido

El diseño del ICS43434 incluye mecanismos para la reducción de ruido, lo que mejora la calidad de la señal capturada en ambientes con alta contaminación acústica, y permite obtener mediciones más precisas.

Bajo Consumo de Energía

El micrófono consume aproximadamente 1 mA en operación, lo que lo hace adecuado para dispositivos que dependen de fuentes de energía limitadas, como baterías. Esto contribuye a una mayor vida útil de la batería y reduce la necesidad de mantenimiento frecuente.

Tamaño Compacto

El ICS43434 tiene un tamaño muy pequeño (4.72 mm x 3.76 mm x 1.0 mm), lo que permite su integración en diseños compactos y portátiles. Esta característica es especialmente importante en aplicaciones donde el espacio es limitado y se requiere un diseño discreto.

Documentación y Recursos

El ICS43434 viene con una amplia documentación y recursos de soporte que facilitan su integración en proyectos. Existen ejemplos de código y bibliotecas disponibles que pueden ayudar a acelerar el desarrollo y la implementación.

Prototipado Rápido

El tamaño compacto y la interfaz I2S del ICS43434 permiten un prototipado rápido y eficiente. Los desarrolladores pueden integrar el micrófono en sus prototipos y comenzar a realizar pruebas rápidamente, lo que agiliza el ciclo de desarrollo del producto.

Flexibilidad en el Diseño

Por su tamaño reducido y bajo consumo de energía, el ICS43434 puede usarse en configuraciones y aplicaciones, desde dispositivos portátiles hasta sistemas de monitoreo ambiental fijos.

Conclusión

El ICS43434 I2S digital microphone es una opción excelente para un dispositivo de medición de ruidos ambientales debido a su alta calidad de audio, bajo consumo de energía, tamaño compacto y facilidad de integración. Sus características técnicas y ventajas operativas hacen que sea una elección ideal para garantizar un rendimiento fiable y preciso en aplicaciones de monitoreo, permitiendo un desarrollo eficiente y una implementación exitosa en entornos variados.

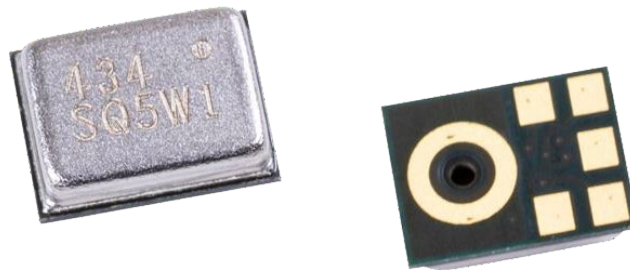


Figura 3: Micrófono ICS43434. Imagen extraída de [6]

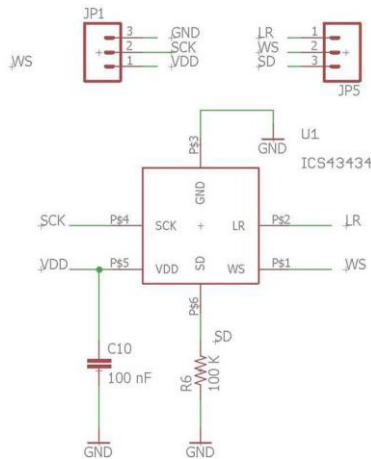


Figura 4: Esquemático Micrófono ICS43434. Imagen Extraída de [6]

Comunicación entre dispositivos

LoRa (Long Range)

LoRa es una tecnología de comunicación inalámbrica de largo alcance y bajo consumo de energía. Utiliza la modulación de espectro ensanchado para lograr comunicaciones a larga distancia (hasta 15 km en áreas rurales) con un consumo de energía muy bajo. LoRa es ideal para aplicaciones IoT que requieren la transmisión de datos a largas distancias y donde la eficiencia energética es crucial.

Características de LoRa:

Alcance Extendido

Una de las características más destacadas de LoRa es su alcance extendido. En condiciones óptimas, LoRa puede transmitir datos a distancias de hasta 15 km en áreas rurales y hasta 5 km en áreas urbanas. Este alcance extenso permite la comunicación entre dispositivos dispersos en áreas grandes sin la necesidad de repetidores adicionales.

Bajo Consumo de Energía

LoRa está diseñado para ser extremadamente eficiente en términos de energía. Los dispositivos que utilizan LoRa pueden operar durante años con baterías estándar, lo que es ideal para aplicaciones que requieren una vida útil prolongada y mantenimiento mínimo.

Comunicación Bidireccional

LoRa permite la comunicación bidireccional, lo que significa que los dispositivos pueden tanto enviar como recibir datos. Esto es esencial para aplicaciones que requieren no solo la transmisión de datos de sensores, sino también el envío de comandos de control o la actualización de firmware.

Alta Capacidad de Red

LoRa puede soportar miles de dispositivos conectados a una sola puerta de enlace (gateway). Esta alta capacidad de red es esencial para aplicaciones de IoT masivas, como la agricultura de precisión, la gestión de ciudades inteligentes y el monitoreo ambiental.

Robustez y Fiabilidad

LoRa utiliza la modulación de espectro ensanchado, lo que le permite ser resistente a las interferencias y ofrecer una transmisión de datos fiable en entornos ruidosos. Esto asegura que los datos críticos se transmitan de manera fiable, incluso en condiciones adversas.

Seguridad

LoRa ofrece encriptación de extremo a extremo, autenticación de dispositivos y seguridad a nivel de red. Esto es crucial para proteger los datos y asegurar que solo los dispositivos autorizados puedan comunicarse en la red, lo que es especialmente importante en aplicaciones sensibles.

Flexibilidad y Escalabilidad

LoRa es altamente flexible y escalable, permitiendo la implementación de redes pequeñas y grandes con facilidad. Las redes LoRa pueden adaptarse a una amplia variedad de aplicaciones y requisitos, desde pequeñas implementaciones locales hasta redes nacionales.

Por qué LoRa es la mejor opción para este proyecto IoT

Alcance y Cobertura

Para un proyecto de medición de ruidos ambientales que podría cubrir grandes áreas, LoRa ofrece la ventaja del largo alcance, asegurando que los datos capturados por los dispositivos puedan ser transmitidos a grandes distancias sin la necesidad de infraestructura adicional, como repetidores.

Eficiencia Energética

La capacidad de operar con un bajo consumo de energía permite que los dispositivos IoT funcionen durante largos periodos con baterías, minimizando el mantenimiento y las recargas. Esto es crucial para dispositivos que se implementan en ubicaciones remotas o de difícil acceso.

Fiabilidad en la Transmisión de Datos

La robustez de LoRa en la transmisión de datos asegura que las mediciones realizadas sean transmitidas de manera fiable, incluso en entornos con interferencias. Esto garantiza que los datos críticos lleguen a su destino sin pérdidas.

Coste-Efectividad

LoRa ofrece una solución rentable para la comunicación de largo alcance y bajo consumo de energía. La infraestructura de red requerida es mínima comparada con otras tecnologías, lo que reduce los costos de implementación y operación.

Seguridad

La seguridad avanzada que ofrece LoRa asegura que los datos de ruido ambiental estén protegidos contra accesos no autorizados y manipulaciones. Esto es vital para la integridad y confiabilidad de los datos en aplicaciones IoT.

Escalabilidad

La alta capacidad de red de LoRa permite que se conecten múltiples dispositivos a una sola red, facilitando la escalabilidad del proyecto a medida que se requiera más cobertura o más dispositivos.

Conclusión

LoRa es la opción ideal para un proyecto de medición de ruidos ambientales debido a su largo alcance, bajo consumo de energía, fiabilidad, seguridad y coste-efectividad. Su capacidad para transmitir datos a largas distancias y operar eficientemente en condiciones adversas asegura una implementación exitosa y duradera del sistema de medición de sonido, permitiendo una monitorización precisa y continua en diversas aplicaciones y entornos.

Posterior a este análisis terminamos optando por el dispositivo LoRa SX1276 de Semtech, que es una solución avanzada para la comunicación inalámbrica en aplicaciones que requieren largo alcance y bajo consumo de energía. A continuación, se describen las características principales y las ventajas que ofrece el SX1276, así como su idoneidad para proyectos de medición de ruido ambiental.

Características del Módulo LoRa SX1276:

Modulación LoRa

El SX1276 implementa la modulación LoRa, caracterizada por su espectro ensanchado de gran dispersión (CSS) que permite transmisiones a largas distancias con baja potencia. LoRa maximiza la relación señal/ruido (SNR) y ofrece robustez frente a interferencias, asegurando enlaces fiables incluso en entornos de alta densidad espectral e interferencia electromagnético.

Rango de Frecuencias

El SX1276 soporta una amplia gama de frecuencias, incluyendo 137 MHz a 1020 MHz. Esta flexibilidad de frecuencia permite su uso en diversas regiones y aplicaciones, cumpliendo con las regulaciones locales de espectro de radiofrecuencia.

Alta Sensibilidad

El SX1276 tiene una alta sensibilidad de hasta -148 dBm. Esta capacidad de detección permite la recepción de señales muy débiles, extendiendo el alcance efectivo de la comunicación y mejorando la fiabilidad de la transmisión de datos en entornos difíciles.

Alta Selectividad

Gracias a su alta selectividad, el SX1276 puede operar eficazmente en entornos con altas interferencias, distinguiendo entre señales deseadas y no deseadas. Esto asegura que los datos se transmitan y reciban con precisión incluso en áreas con mucha distorsión de radiofrecuencia.

Baja Potencia de Consumo

El SX1276 está diseñado para consumir muy poca energía, lo que es ideal para aplicaciones alimentadas por baterías. Con modos de baja potencia, los dispositivos pueden operar durante largos periodos sin necesidad de recargas frecuentes.

Comunicación Bidireccional

El SX1276 soporta comunicación bidireccional, permitiendo tanto la transmisión como la recepción de datos. Esta capacidad es esencial para aplicaciones que requieren retroalimentación o control remoto, además de la simple transmisión de datos de sensores.

Soporte para Redes en Malla

El SX1276 puede ser utilizado en configuraciones de redes en malla, donde los datos pueden ser reenviados a través de múltiples nodos para alcanzar el destino final. Esto amplía aún más el alcance y la cobertura de la red.

Modo de Ancho de Banda Ajustable

El SX1276 permite ajustar el ancho de banda de transmisión, lo que proporciona flexibilidad para optimizar la velocidad de datos y el alcance según las necesidades específicas de la aplicación.

Idoneidad para Proyectos de Medición de Ruido Ambiental:

El SX1276 es particularmente adecuado para este proyecto por varias razones:

- **Largo Alcance:** Permite la colocación de los dispositivos en ubicaciones dispersas, cubriendo grandes áreas sin necesidad de infraestructura adicional.
- **Bajo Consumo de Energía:** Los transceptores pueden operar durante largos periodos con baterías, reduciendo la necesidad de mantenimiento.
- **Fiabilidad:** Asegura la transmisión precisa de datos, incluso en entornos con interferencias electromagnéticas.
- **Seguridad:** Protege los datos sensibles recolectados contra accesos no autorizados, garantizando la integridad de la información.

Conclusión

El módulo LoRa SX1276 de Semtech ofrece una combinación ideal de largo alcance, bajo consumo de energía, alta fiabilidad y seguridad, lo que lo convierte en una opción excelente para proyectos de medición de ruido ambiental. Su flexibilidad y robustez aseguran un rendimiento superior en una variedad de entornos y aplicaciones,

proporcionando una solución eficiente y rentable para la comunicación inalámbrica en dispositivos IoT.



Figura 5: Dispositivo LoRa SX1276. Imagen extraída de [7]

BMP280

El BMP280 es un sensor barométrico de presión y temperatura fabricado por Bosch Sensortec, diseñado específicamente para aplicaciones móviles y de IoT. A continuación, se detallan sus características y ventajas.

Características del BMP280:

Medición de Presión

El BMP280 mide la presión barométrica en un rango de 300 hPa a 1100 hPa con una precisión absoluta de ± 1 hPa. Esta capacidad permite su uso en aplicaciones que requieren mediciones precisas de altitud y presión atmosférica.

Medición de Temperatura

El sensor también mide la temperatura en un rango de -40°C a $+85^{\circ}\text{C}$ con una precisión de $\pm 1^{\circ}\text{C}$. Esto permite la compensación de temperatura en las mediciones de presión, mejorando la precisión general del sensor.

Bajo Consumo de Energía

El BMP280 está diseñado para ser muy eficiente en términos de consumo de energía, con un consumo típico de $2.7 \mu\text{A}$ en modo de medición normal. Esto lo hace adecuado para dispositivos alimentados por batería que requieren una larga duración de la batería.

Interfaz Digital

El BMP280 utiliza una interfaz digital I2C y SPI, lo que facilita su integración con una amplia variedad de microcontroladores y sistemas embebidos. La interfaz digital también simplifica la comunicación y el diseño del hardware.

Alta Resolución

El sensor ofrece una resolución de hasta 0.18 Pa para la medición de presión y 0.01°C para la medición de temperatura. Esta alta resolución permite la detección de cambios sutiles en la presión y temperatura, lo que es esencial para aplicaciones de precisión.

Amplia Gama de Aplicaciones

El BMP280 puede utilizarse en diversas aplicaciones, desde la medición de altitud en dispositivos de navegación hasta el monitoreo de la presión atmosférica en estaciones meteorológicas. Su versatilidad lo convierte en una opción popular para muchos proyectos.

Tamaño Compacto

El pequeño tamaño del BMP280 permite su uso en diseños compactos y portátiles, lo que es esencial para dispositivos móviles y aplicaciones donde el espacio es limitado.

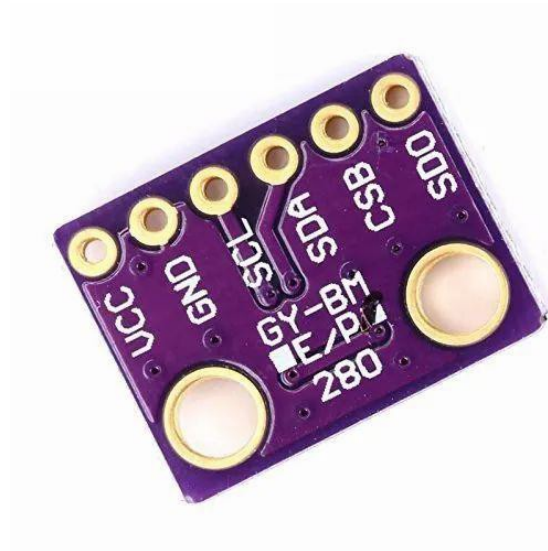


Figura 6: Modulo BMP280. Imagen extraida de [8]

RTC

Características y Ventajas del Módulo RTC (Real-Time Clock)

Los módulos RTC (Real-Time Clock) son esenciales para mantener la precisión del tiempo en aplicaciones que requieren una sincronización exacta y continua. A continuación, se describen las características y ventajas de los módulos RTC.

Características del RTC:

Alta Precisión

El DS3231 es un RTC de alta precisión con un cristal oscilador integrado que proporciona una precisión de ± 2 ppm, lo que equivale a una desviación de aproximadamente 1 minuto por año. Esta precisión es esencial para aplicaciones que requieren una medición exacta del tiempo.

Alimentación de Reserva con Batería

El DS3231 incluye una función de alimentación de reserva con batería, lo que permite al reloj continuar funcionando y manteniendo la hora correcta incluso cuando el dispositivo principal está apagado. Esto asegura una continuidad en la medición del tiempo sin interrupciones.

Interfaz I2C

El módulo utiliza una interfaz I2C para la comunicación, lo que facilita su integración con microcontroladores y sistemas embebidos. La interfaz I2C es comúnmente utilizada en aplicaciones y facilita la conexión de múltiples dispositivos en un solo bus.

Alarmas Programables

El DS3231 incluye alarmas programables que pueden configurarse para generar interrupciones en momentos específicos. Estas alarmas son útiles para despertar un sistema de bajo consumo o para realizar acciones programadas en tiempos precisos.

Sensor de Temperatura Interno

El DS3231 tiene un sensor de temperatura interno que permite la compensación de temperatura, mejorando la precisión del reloj a lo largo de una amplia gama de temperaturas operativas.

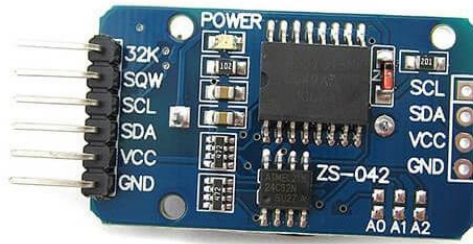


Figura 7: Modulo RTC DS3231. Imagen extraida de [9]

Diseño del Esquemático

Ya que el dispositivo todavía es un prototipo para un sistema posterior comercial utilizamos los recursos que poseíamos en el momento de desarrollo por lo que quedaron dos placas distintas a la hora de realizar los nodos, esto debido a que poseemos dos modelos de ESP32 distintas y una tercera placa que cumplirá la función de Gateway. A continuación, se adjuntarán las imágenes correspondientes a cada diseño de las placas.

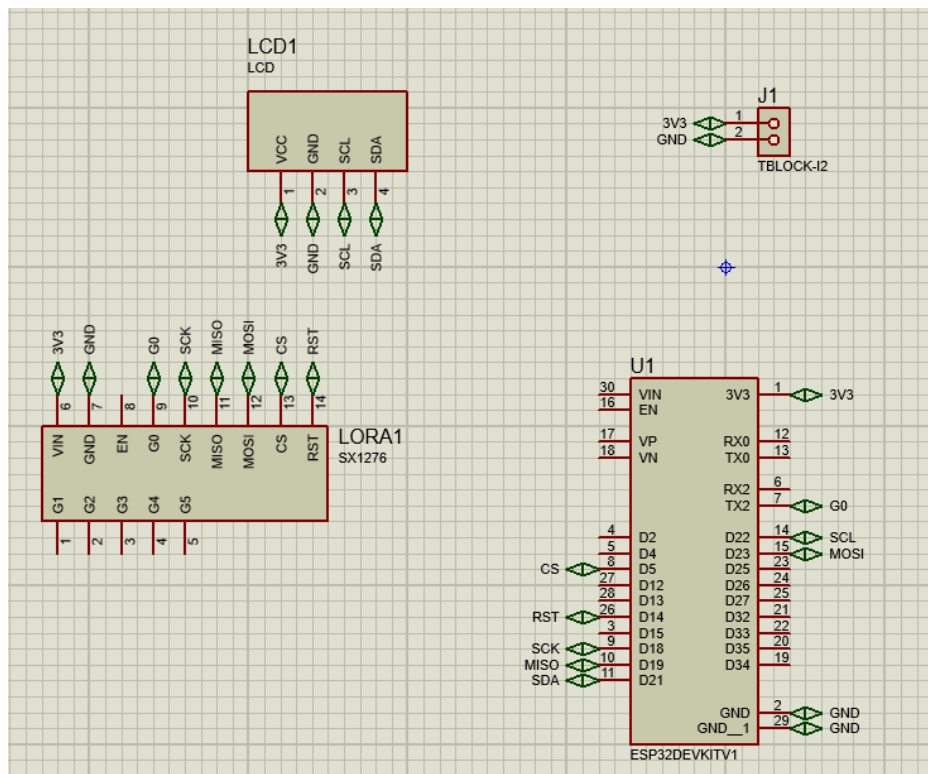


Figura 8: Diseño Esquemático Gateway. Imagen de elaboración propia

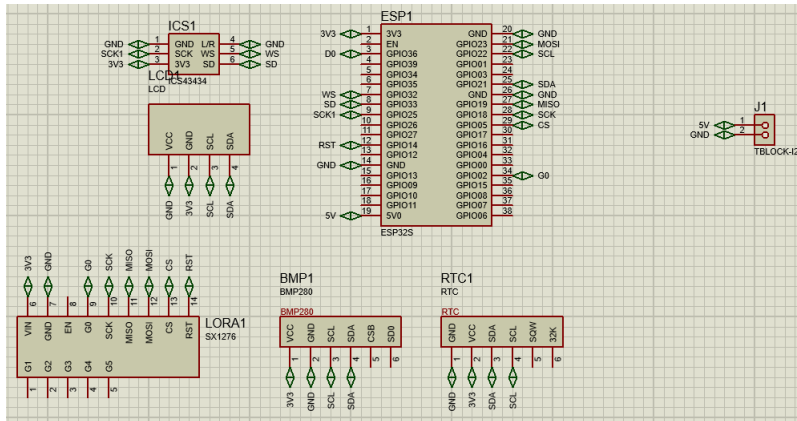


Figura 9: Diseño Esquemático Nodo Placa Uno. Imagen de elaboración propia

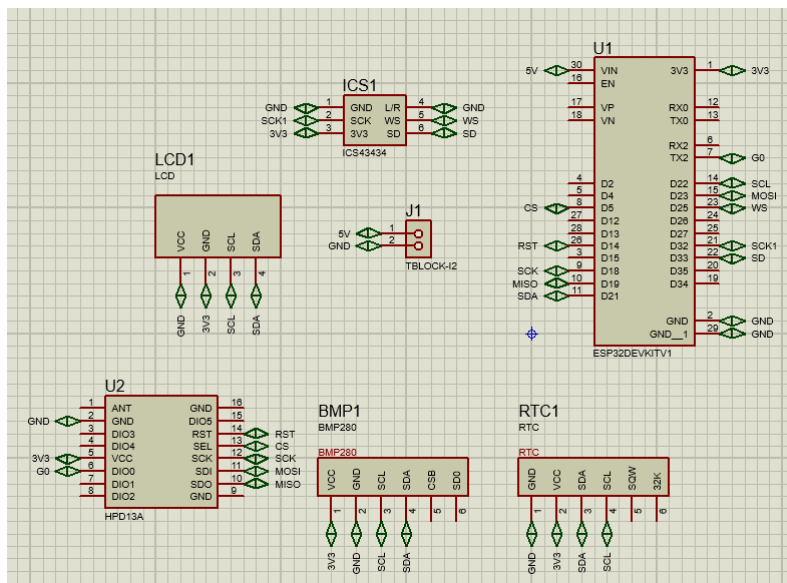


Figura 10: Diseño esquemático Nodo Placa Dos. Imagen de elaboración propia

Diseño de PCB

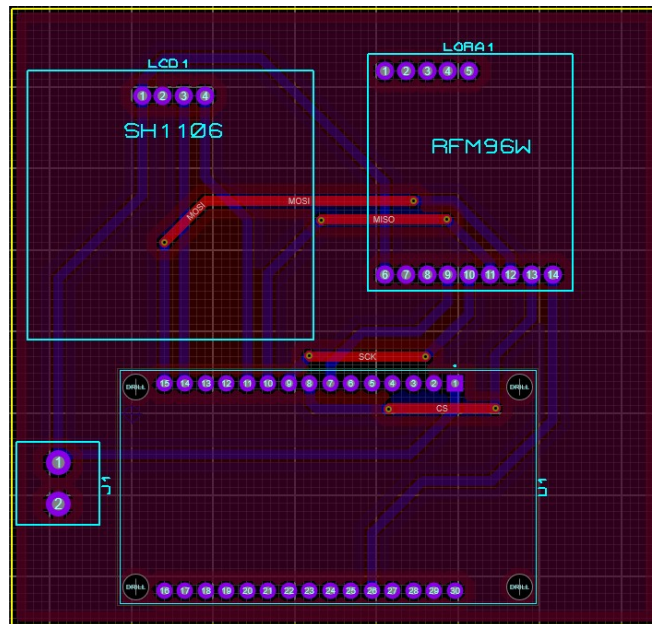


Figura 11: PCB Gateway. Imagen de elaboración propia.

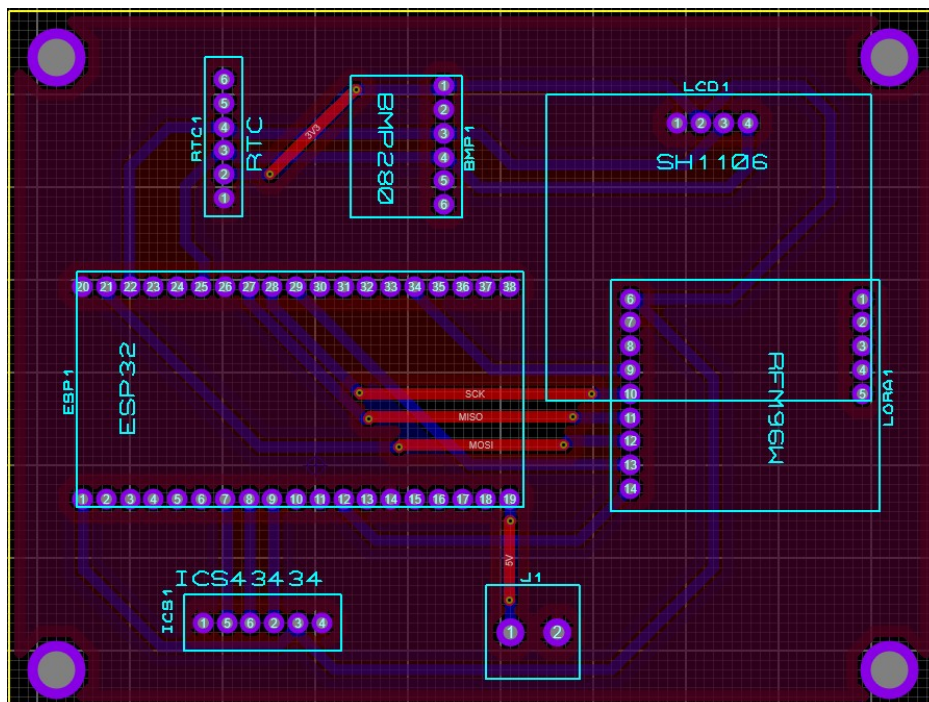


Figura 12: PCB Nodo Uno. Imagen de elaboración propia

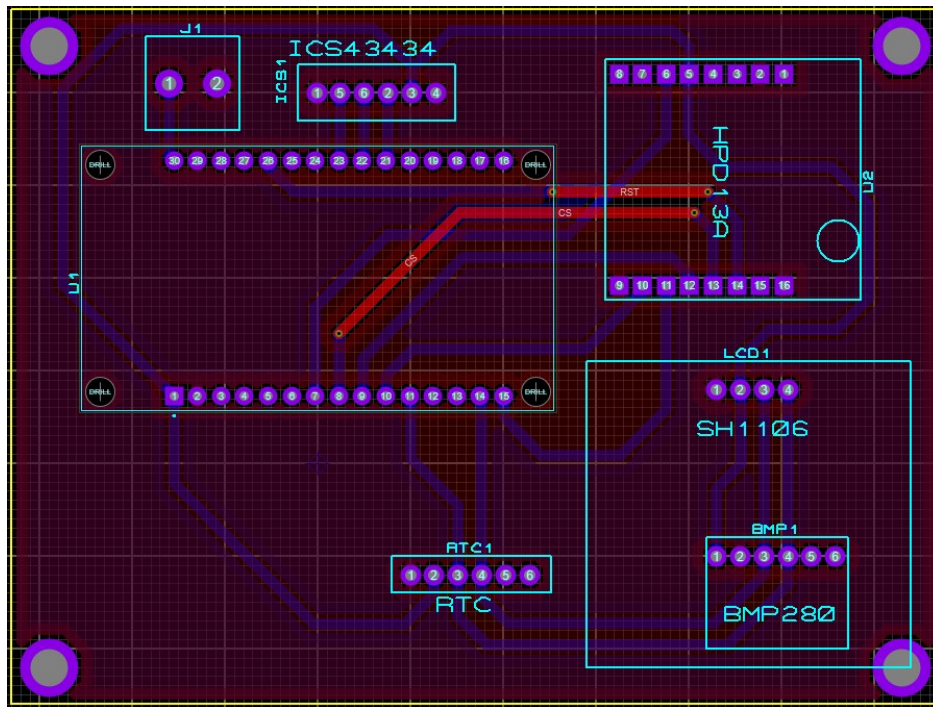


Figura 13: PCB Nodo Dos. Imagen de elaboración propia

Tabla de conexiones

Nodo 1:

Módulos Utilizados	Pin	Conexión ESP
ESP32S Node MCU	N/A	N/A
ICS43434	GND	GND
	SCK	GPIO25
	3V3	3V3
	L/R	GND
	WS	GPIO32
	SD	GPIO33
LCD	VCC	3V3
	GND	GND
	SCL	GPIO22
	SDA	GPIO21
SX1276	G1	N/A
	G2	N/A
	G3	N/A

	G4	N/A
	G5	N/A
	G0	GPIO2
	VIN	3V3
	GND	GND
	EN	N/A
	SCK	GPIO18
	MISO	GPIO19
	MOSI	GPIO23
	CS	GPIO5
	RST	GPIO14
BMP280	VCC	3V3
	GND	GND
	SCL	GPIO22
	SDA	GPIO21
	CSB	N/A
	SD0	N/A
RTC	GND	GND
	VCC	3V3
	SDA	GPIO21
	SCL	GPIO22
	SQW	N/A
	32K	N/A
BATERIA	5V	5V0
	GND	GND

Nodo 2:

Módulos Utilizados	Pin	Conexión ESP
ESP32 DEVKIT V1	N/A	N/A
ICS43434	GND	GND
	SCK	GPIO32
	3V3	3V3

	L/R	GND
	WS	GPIO25
	SD	GPIO33
LCD	VCC	3V3
	GND	GND
	SCL	GPIO22
	SDA	GPIO21
HPD13A	DI01	N/A
	DI02	N/A
	DI03	N/A
	DI04	N/A
	DI05	N/A
	DI00	TX2
	VCC	3V3
	GND	GND
	ANT	N/A
	SCK	GPIO18
	SDO	GPIO19
	SDI	GPIO23
	SEL	GPIO5
	GND	N/A
	GND	N/A
RST	GPIO14	
BMP280	VCC	3V3
	GND	GND
	SCL	GPIO22
	SDA	GPIO21
	CSB	N/A
	SD0	N/A
RTC	GND	GND
	VCC	3V3
	SDA	GPIO21
	SCL	GPIO22
	SQW	N/A

	32K	N/A
BATERIA	5V	5V0
	GND	GND

Gateway:

Módulos Utilizados	Pin	Conexión ESP
ESP32 DEVKIT V1	N/A	N/A
LCD	VCC	3V3
	GND	GND
	SCL	GPIO22
	SDA	GPIO21
SX1276	G1	N/A
	G2	N/A
	G3	N/A
	G4	N/A
	G5	N/A
	G0	TX2
	VIN	3V3
	GND	GND
	EN	N/A
	SCK	GPIO18
	MISO	GPIO19
	MOSI	GPIO23
	CS	GPIO5
RST	GPIO14	
BATERIA	5V	5V0
	GND	GND

Firmware

Antes de comenzar a explicar el código debemos realizar una aclaración ya que al ser dos dispositivos distintos el nodo y el Gateway cada uno posee su código particular para las funciones que debe realizar con respecto a las tareas necesarias.

Nodos:

Para la programación del firmware de los nodos primero se decidió como iba a ser el funcionamiento de los mismos por lo cual se optó por una serie de tareas de manera secuencial que se realizan mediante FreeRTOS. Esto nos sirvió para tener una mejor visibilidad a la hora de saber que estaba realizando el dispositivo en los distintos momentos por lo cual lo separamos en 4 tareas distintas. Además, al separarlo por tareas podemos tener un mejor control de cómo es el consumo de la energía en el dispositivo y con esto generar una mayor durabilidad de la batería. Ya que el dispositivo no se encontrará en todo momento realizando mediciones, sino que se realizaran de manera periódica en un intervalo de tiempo al utilizar tareas podemos tener mayor control de lo mismo generando una tarea específica para que el dispositivo en los momentos donde no deba realizar mediciones se encuentre “durmiendo” para el ahorro de energía.

Micrófono

Comenzaremos explicando la cual nosotros consideramos es la tarea principal del dispositivo ya que es la encargada de realizar la inicialización del micrófono y posterior toma de los distintos valores que serán enviados hacia la página web.

La funcionalidad principal del código incluye la lectura de datos del micrófono, el procesamiento de estos datos con filtros digitales y el cálculo de los niveles de presión sonora (SPL) ponderados, que luego se muestran en una pantalla OLED.

Librerías y Definiciones

Se incluyen varias librerías para manejar la comunicación I2S (Inter-IC Sound), la pantalla OLED y los filtros digitales:

- *driver/i2s.h*: Proporciona funciones para manejar la comunicación I2S, que se utiliza para recibir datos del micrófono.
- *Adafruit_SSD1306.h* y *Adafruit_GFX.h*: Usadas para manejar la pantalla OLED.

- *sos-iir-filter.h*: Define los filtros IIR (Infinite Impulse Response) usados para el procesamiento de señales.
- *defines.h*, *display.h*: Archivos personalizados que contienen definiciones y funciones auxiliares.

Variables y Configuración de Filtros

El código configura varios filtros digitales con coeficientes específicos para el micrófono y el procesamiento de señales:

- Filtros *SOS_IIR_Filter*: Se definen varios filtros IIR, incluyendo el filtro de ecualización y los filtros de ponderación A y C, que son usados para procesar los datos del micrófono y calcular el nivel de presión sonora ponderado.
- Variables Globales:
 - *display*: Objeto para controlar la pantalla OLED.
 - *samples_queue*: Cola de FreeRTOS para pasar datos entre tareas.
 - *samples*: Buffer para almacenar muestras del micrófono.

Función *microfonoSetup()*

Esta función inicializa el sistema y realiza el procesamiento de los datos del micrófono:

Inicialización:

- Configura la pantalla OLED.
- Calcula la amplitud de referencia del micrófono.
- Crea una cola para pasar datos entre tareas.
- Crea una tarea de FreeRTOS (*mic_i2s_reader_task*) para leer datos del micrófono.

Cálculo del *Leq*:

- Lee datos de la cola y calcula el nivel equivalente de presión sonora (*Leq*) en decibelios.
- Muestra el *Leq* en la pantalla OLED.

Función *mic_i2s_reader_task*

Esta tarea se encarga de leer datos del micrófono y procesarlos:

Inicialización:

- Configura el periférico I2S para la lectura de datos del micrófono.

Lectura y Procesamiento:

- Lee datos del micrófono a través de I2S.
- Convierte las muestras de enteros a flotantes.
- Aplica filtros de ecualización y ponderación.
- Calcula la suma de los cuadrados de las muestras para usar en el cálculo del SPL y del *Leq*.
- Envía los resultados procesados a la cola de FreeRTOS.

Función *mic_i2s_init*

Configura el periférico I2S para la lectura de datos del micrófono:

Configuración del I2S:

- Define el modo del I2S, la frecuencia de muestreo y el formato de comunicación.
- Configura los pines para la señal de reloj, la señal de palabra y los datos de entrada.
- Instala el driver de I2S y configura los registros si es necesario para solucionar problemas de temporización.

Sensado

Este código está diseñado para obtener y mostrar información de fecha y hora, así como datos de temperatura, presión y altitud, utilizando un sensor de presión barométrica BMP280 y un módulo RTC DS3231 en un microcontrolador. Los resultados se almacenan en variables para su uso posterior. A continuación, se detalla la funcionalidad de cada parte del código:

Librerías Incluidas

- *RTC.h* y *RTCLib.h*: Librerías para la gestión del módulo RTC (Real Time Clock) DS3231, que proporciona la fecha y la hora actuales.

- *Wire.h*: Librería para la comunicación I2C, necesaria para la comunicación con los sensores BMP280 y el módulo RTC.
- *bmp.h*: Archivo de cabecera personalizado que contiene funciones relacionadas con el sensor BMP280.
- *sensado.h*: Archivo de cabecera personalizado que define la estructura *fecha_data* y las funciones relacionadas con la obtención de datos.

Variables Globales

- *RTC_DS3231*: Objeto para interactuar con el módulo RTC DS3231.
- *Adafruit_BMP280*: Objeto para interactuar con el sensor BMP280 a través de I2C.
- *daysOfWeek[7][10]*: Array que contiene los nombres de los días de la semana en español.
- *fecha, hora, tiempo*: Arrays de caracteres para almacenar la fecha y la hora en diferentes formatos.

Función sensado

Esta función obtiene datos del sensor BMP280 y del módulo RTC DS3231, y los almacena en variables para su uso posterior. También formatea la fecha y la hora para su presentación.

Inicialización:

- *rtcSetup(&rtc)*: Configura el módulo RTC.
- *bmpSetup(&bmp)*: Configura el sensor BMP280.

Obteniendo Datos del RTC:

- *DateTime now = rtc.now()*: Obtiene la fecha y la hora actuales del módulo RTC.
- *sprintf(fecha_envio->fecha, "%04u-%02u-%02u", now.year(), now.month(), now.day())*: Formatea la fecha en el formato YYYY-MM-DD y la almacena en el campo *fecha* de la estructura *fecha_envio*.
- *sprintf(fecha, "%02u/%02u/%04u", now.day(), now.month(), now.year())*: Formatea la fecha en el formato DD/MM/YYYY y la almacena en la variable *fecha*.

- `sprintf(fecha_envio->hora, "%02u:%02u:%02u", now.hour(), now.minute(), now.second())`: Formatea la hora en el formato HH:MM:SS y la almacena en el campo hora de la estructura `fecha_envio`.
- `sprintf(hora, "%02u:%02u:%02u", now.hour(), now.minute(), now.second())`: Formatea la hora en el formato HH:MM:SS y la almacena en la variable hora.
- `strcpy(fecha_envio->dia, daysOfWeek[now.dayOfTheWeek()])`: Copia el nombre del día de la semana al campo día de la estructura `fecha_envio`.

Obteniendo Datos del BMP280:

- `*temperatura = bmp.readTemperature()`: Lee la temperatura del sensor BMP280 y la almacena en la variable temperatura.
- `*presion = bmp.readPressure() / 100`: Lee la presión del sensor BMP280, la convierte de Pa a hPa (dividiendo por 100) y la almacena en la variable presión.
- `*altura = bmp.readAltitude(1013.25) * (-1)`: Calcula la altitud a partir de la presión medida (asumiendo una presión estándar de 1013.25 hPa) y la almacena en la variable altura. La multiplicación por -1 es para ajustar el signo de la altitud.

LoraSender

Este código está diseñado para configurar y utilizar un módulo LoRa para enviar datos a través de una red de punto a punto. Utiliza la biblioteca RHMesh para la comunicación de malla y la biblioteca CRC32 para calcular un checksum para la integridad del mensaje. A continuación, se detalla la funcionalidad de cada parte del código:

Librerías Incluidas

- `LoRaSender.h`: Archivo de cabecera personalizado para definir funciones y constantes relacionadas con la comunicación LoRa.
- `CRC32.h`: Proporciona funciones para calcular un checksum CRC32, que se usa para verificar la integridad de los datos.

Variables Globales

- `latitud` y `longitud`: Variables que contienen las coordenadas geográficas del nodo.

- *buf[RH_MESH_MAX_MESSAGE_LEN]*: Buffer para recibir mensajes.
- *res*: Variable para almacenar el resultado de las operaciones de envío.

Función loraSetup

Configura el módulo LoRa y los parámetros de la comunicación:

Configuración del SPI:

- *SPI.begin(LLG_SCK, LLG_MISO, LLG_MOSI, LLG_CS)*: Inicializa la interfaz SPI con los pines especificados.

Inicialización del Manager:

- *manager->init()*: Inicializa el objeto de gestión de malla. Si falla, se imprime un mensaje de error.

Configuración del Transmisor:

- *rf95->setTxPower(15, false)*: Establece la potencia de transmisión del módulo LoRa. El primer parámetro es la potencia en dBm, el segundo parámetro indica si se usa *PA_BOOST (false)* o *RFO (true)*.
- *rf95->setFrequency(915.0)*: Configura la frecuencia de operación del módulo LoRa a 915 MHz.
- *rf95->setCADTimeout(500)*: Establece el tiempo de espera para la detección de canales.

Configuración del Módem:

- Si *longRange* es verdadero, se configura el módem con parámetros personalizados para rango largo. De lo contrario, se usa la configuración predeterminada de rango medio.

Mensajes de Estado:

- Imprime mensajes para indicar el estado de la inicialización y configuración.

Función loraSender

Envía datos a través de la red LoRa y maneja la recepción de mensajes:

Creación del Mensaje JSON:

- *sprintf(dataMensaje, "...")*: Formatea los datos sensados en una cadena JSON, incluyendo la latitud, longitud, y un timestamp en formato YYYY-MM-DDTHH:MM:SSZ.

Cálculo del Checksum:

- Utiliza CRC32 para calcular el checksum de la cadena de datos y lo agrega al mensaje en formato hexadecimal.

Envío del Mensaje:

- Convierte la cadena JSON en un arreglo de bytes (*arreglo_bytes*).
- Envía el mensaje al *BRIDGE_ADDRESS* utilizando el método *sendtoWait* del objeto manager. Imprime el resultado del envío y maneja posibles errores.

Recepción de Mensajes:

- El nodo, una vez que envía el dato espera un acknowledge del Gateway. Si la respuesta es errónea, éste mide y reenvía el mensaje.

SleepTask

Esta tarea es la encargada de realizar que el dispositivo entre en modo de sueño o deep sleep la cual nos facilita que el mismo reduzca considerablemente el consumo de energía y así poder ahorrar baterías de los dispositivos. A continuación, se realizará una pequeña explicación de su funcionamiento.

La función *esp_deep_sleep* es una característica fundamental del ESP32, diseñada para optimizar el consumo de energía en aplicaciones que requieren un funcionamiento prolongado con batería. A continuación, se detalla cómo funciona y por qué es importante para la eficiencia energética de los sistemas basados en ESP32.

Introducción al Modo de Sueño Profundo

El ESP32 es un microcontrolador versátil con múltiples modos de bajo consumo energético. El modo de sueño profundo es el más eficiente en términos de consumo de energía, permitiendo que el ESP32 reduzca su consumo a niveles extremadamente bajos cuando no está realizando tareas activas. Este modo es ideal para aplicaciones que deben operar durante largos períodos con baterías limitadas, como dispositivos IoT, sensores remotos y dispositivos portátiles.

¿Qué Hace la Función `esp_deep_sleep`?

La función `esp_deep_sleep` se utiliza para poner el ESP32 en modo de sueño profundo. Cuando se llama a esta función, el microcontrolador realiza una serie de pasos para minimizar su consumo de energía:

- **Desactivación de Componentes:** La mayoría de los componentes internos del ESP32, incluidos el CPU y la mayoría de los periféricos, se apagan. Solo algunos componentes esenciales permanecen activos para permitir la reactivación del dispositivo.
- **Preservación de Datos:** En este modo, el ESP32 puede conservar ciertos datos en la memoria RTC (Real-Time Clock) para su recuperación cuando el dispositivo se despierta. Esto es útil para mantener la configuración del sistema o almacenar datos críticos que deben persistir entre ciclos de sueño.
- **Configuración del Temporizador de Despertar:** La función permite configurar un temporizador que define el período de tiempo tras el cual el ESP32 debe despertarse automáticamente. Este temporizador puede ser configurado en milisegundos, lo que ofrece flexibilidad para adaptarse a las necesidades específicas de la aplicación.

Consideraciones

- **Interrupciones y Fuentes de Despertar:** Además del temporizador, el ESP32 puede configurarse para despertarse en respuesta a otras fuentes, como interrupciones externas o señales de otros periféricos. Esto permite una gran flexibilidad en cómo y cuándo el dispositivo debe reactivarse.
- **Consumo de Energía:** Aunque el modo de sueño profundo reduce drásticamente el consumo de energía, el diseño del hardware y la configuración del sistema también influyen en la eficiencia energética. Es importante considerar todos los aspectos del diseño del sistema para lograr el máximo ahorro energético.

Gateway

Para la realización del Gateway primero comenzamos basándonos en una plantilla provista por Azure el cual nos aseguraba la conexión hacia el IoT Central el cual fue el sistema usado para la visualización de los datos en los distintos dashboards. Por lo cual comenzaremos explicando cómo funciona dicho código y las características que posee.

Primero, se incluyen las librerías necesarias para la conexión WiFi, el protocolo MQTT y Azure IoT. Estas librerías proporcionan las funciones y métodos necesarios para establecer la conexión y manejar la comunicación con Azure. Las credenciales de WiFi y los detalles específicos del dispositivo y del servicio IoT se almacenan en variables definidas como:

- *WIFI_SSID*
- *WIFI_PASSWORD*
- *AZURE_IOT_DPS_ID_SCOPE*
- *AZURE_IOT_HUB_HOSTNAME*
- *AZURE_IOT_DEVICE_ID*
- *AZURE_IOT_PRIMARY_KEY*

Estas variables son esenciales para la autenticación y la configuración de la conexión.

La función *setupWiFi()* se encarga de conectar el ESP32 a la red WiFi utilizando las credenciales previamente definidas. Dentro de esta función, *WiFi.begin()* inicia la conexión y *WiFi.status()* verifica el estado de la conexión, asegurándose de que el dispositivo esté correctamente conectado a la red antes de proceder.

Una vez establecida la conexión WiFi, la función *initTime()* sincroniza la hora del ESP32 con un servidor NTP. Esto es crucial para la autenticación con Azure IoT, ya que muchos protocolos de seguridad requieren una marca de tiempo precisa. Posteriormente, *initializeIoTHub()* se encarga de inicializar la conexión al hub IoT de Azure. Esta función utiliza el SDK de Azure para establecer una conexión segura y mantener la comunicación con el hub. La variable *IOTHUB_CLIENT_LL_HANDLE* es clave en esta parte del proceso, ya que maneja la conexión al hub IoT.

Para registrar el dispositivo con Azure, se utiliza la función *provisionDevice()*, que interactúa con el Azure Device Provisioning Service (DPS). Este servicio facilita el registro del dispositivo en el hub IoT correcto basado en la configuración proporcionada.

La variable *prov_device_ll_handle* gestiona la conexión de provisión del dispositivo durante este proceso.

El envío de datos de telemetría al hub IoT se realiza mediante la función *sendTelemetry()*. Esta función puede recopilar datos de sensores conectados al ESP32 y enviarlos a Azure IoT Central. El método *IoTHubClient_LL_SendEventAsync()* permite enviar estos datos de manera asíncrona, asegurando que la aplicación no se bloquee durante la transmisión de datos. La variable *messageHandle* se utiliza para gestionar los mensajes que se envían al hub IoT.

La función *loop()* es el núcleo del código que se ejecuta continuamente. En este bucle principal, se mantiene la conexión a WiFi y Azure IoT, y se procesan las comunicaciones entrantes y salientes. El método *IoTHubClient_LL_DoWork()* es fundamental aquí, ya que mantiene la conexión al hub IoT activa y maneja tanto la recepción como el envío de mensajes.

El código también incluye funciones de callback como *deviceTwinCallback()* y *directMethodCallback()*. Estas funciones manejan las actualizaciones de propiedades del dispositivo y los comandos directos enviados desde Azure IoT Central. Por ejemplo, *deviceTwinCallback()* se activa cuando hay una actualización en las propiedades del dispositivo, permitiendo que el dispositivo reaccione a los cambios configurados desde la nube.

La gestión de errores y reconexiones es otra parte crucial del código. El método *IoTHubClient_LL_SetConnectionStatusCallback()* configura un callback que maneja los cambios en el estado de la conexión. La función *connectionStatusCallback()* se encarga de manejar estos cambios y puede intentar reconectar automáticamente si se pierde la conexión, asegurando una comunicación continua y fiable con Azure IoT Central.

Una vez configuradas todas estas funciones y variables provistas por dicho template lo único que nos quedó realizar fue la toma de datos provenientes del módulo LoRa y acomodar el mensaje de tipo JSON para subir dichos datos a la nube.

Pruebas de campo

Micrófono

Comparación con el Sonómetro SVAN 958A

Para validar la precisión y fiabilidad de nuestro sonómetro, realizamos comparaciones con un sonómetro comercial SVAN 958A ampliamente utilizado en aplicaciones de medición de ruido. En el laboratorio de Ing. Mecánica, con la ayuda de Adrián Azurro y Martin Sequeira se llevaron a cabo mediciones en entornos controlados con diferentes niveles de presión acústica.

Los resultados de las mediciones se registraron y se compararon con los valores proporcionados por el SVAN 958A. Se evaluaron la precisión, la estabilidad y la respuesta en frecuencia de nuestro sonómetro en comparación con el dispositivo comercial.

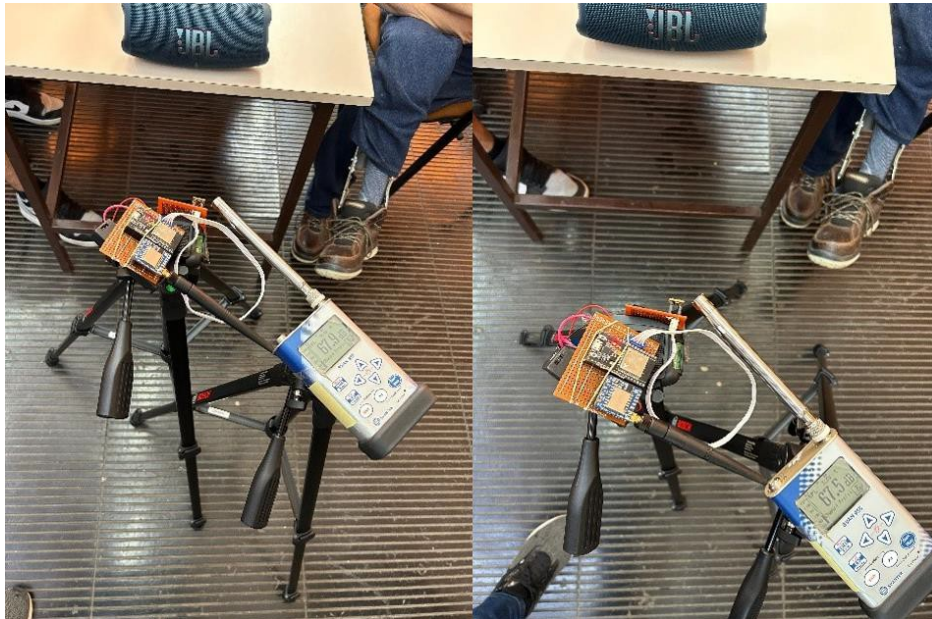


Figura 14: Ensayo de comparación con el dispositivo homologado. Imagen de elaboración propia

Realizamos dos tipos de mediciones la primera en un ambiente donde teníamos ruido ambiente de fondo y con distintas señales de tonos puros los cuales nos dieron los siguientes resultados.

<i>Freq MHz</i>	<i>Sonómetro Calibrado</i>	<i>Sonómetro Experimental</i>	<i>Diferencia</i>
50Hz	62,6	62,6	0
100Hz	82,4	81,4	1
125Hz	75,1	74,1	1
250Hz	84,4	83	1,4
500Hz	88,8	87,7	1,1
1000Hz	85,8	85,3	0,5
2000Hz	81,8	85,3	3,5
4000Hz	66	73,5	7,5
8000Hz	63,9	68,6	4,7

Tabla 1: Comparación 1 de sonómetros en un barrido de frecuencia. Imagen de elaboración propia

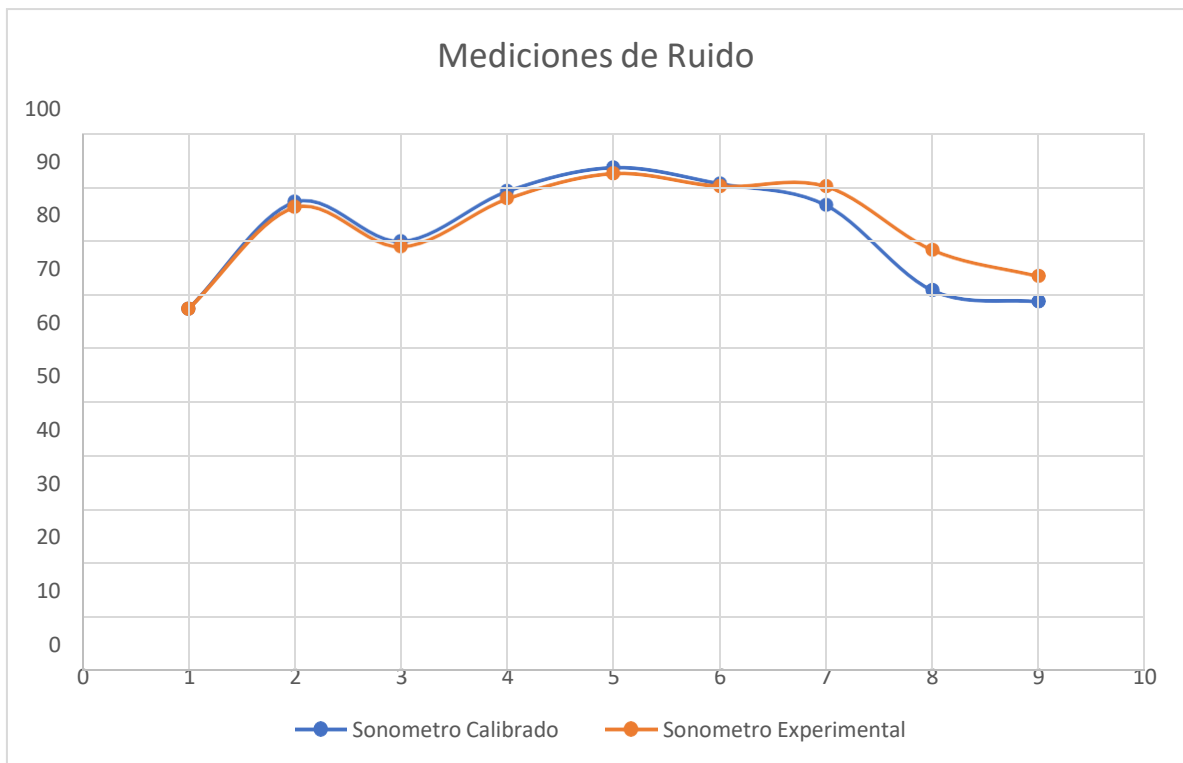


Figura 15: Comparación de las muestras obtenidas en un ambiente ruidoso. Imagen de elaboración propia.

A partir de esta medición realizamos una segunda medición agregando valores de frecuencias de tonos puros en un ambiente controlado, y este fue el resultado:

<i>Frecuencia</i>	<i>Sonómetro Calibrado</i>	<i>Sonómetro Experimental</i>	<i>Diferencia</i>
50	53,3	54,2	-0,9
63	61,5	62,3	-0,8
100	67,2	67,3	-0,1
125	78,4	78,5	-0,1
150	79,8	79,9	-0,1
250	69,9	69,8	0,1
400	69,5	69,4	0,1
500	81,1	81	0,1
630	67	66,9	0,1
1000	74,7	74,6	0,1
1250	73	73,3	-0,3
2000	74,2	74,7	-0,5
3000	68	70,5	-2,5
4000	64,7	67	-2,3
6300	71,8	69	2,8
8000	71	75,5	-4,5
10000	62,6	70,3	-7,7

Tabla 2: Comparación 2 de sonómetros en un barrido de frecuencia. Imagen de elaboración propia

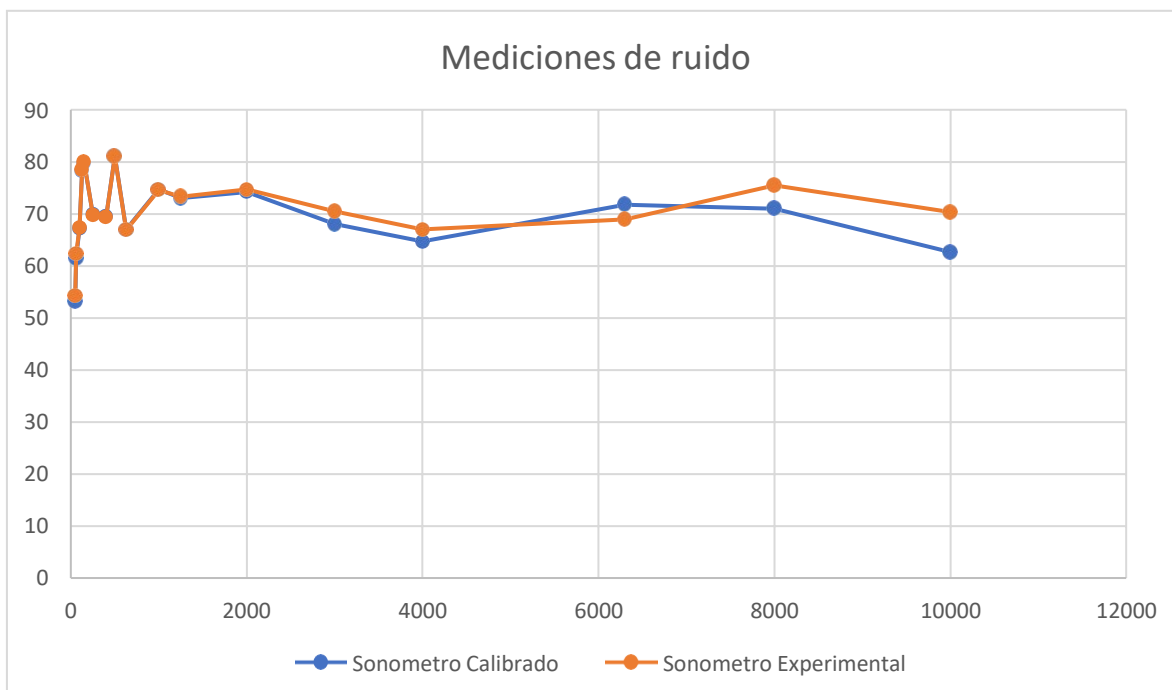


Figura 16: Comparación de las muestras obtenidas en un ambiente controlado. Imagen de elaboración propia.

Como último experimento para poder tener una medición más real ya que las anteriores fueron realizadas sobre tonos puros de frecuencias distintas lo que se realizó es mediante el audio de sonido ambiente de calle, se tomaron 37 mediciones las cuales se realizó el cálculo de sus valores en pascales y posterior se sacó el valor medio de la potencia en dBA.

<i>Valores medidos por el SVAN 958A</i>			<i>Valores medidos por el dispositivo</i>		
<i>dBA</i>	<i>Presión hPa</i>	<i>Promedio Ponderado</i>	<i>dBA</i>	<i>Presión hPa</i>	<i>Promedio Ponderado</i>
68,4	6918309,71		70,07	10162486,93	
69,8	9549925,86		68,22	6637430,704	
68,2	6606934,48		70,15	10351421,67	
67,2	5248074,6	66,46 dBA	68,87	7709034,691	66,95 dBA
66,4	4365158,32		67,49	5610479,76	
64,3	2691534,8		67,14	5176068,32	
65,3	3388441,56		64,57	2864177,97	
65,8	3801893,96		65,47	3523708,71	
65,1	3235936,57		66,19	4159106,105	
67,7	5888436,55		65,53	3572728,382	
67,4	5495408,74		68,31	6776415,076	
65,1	3235936,57		66,02	3999447,498	
67	5011872,34		67,38	5470159,629	
68,4	6918309,71		66,64	4613175,746	
68,8	7585775,75		68,72	7447319,739	
64,5	2818382,93		68,95	7852356,346	
63,7	2344228,82		64,91	3097419,299	
63,6	2290867,65		64,53	2837919,028	
67,7	5888436,55		63,48	2228435,149	
68,6	7244359,6		66,88	4875284,901	
65,1	3235936,57		69,65	9225714,272	
65	3162277,66		65,35	3427677,865	
64,9	3090295,43		65,17	3288516,309	
65	3162277,66		65,59	3622429,984	
64,7	2951209,23		65,13	3258367,01	
64,6	2884031,5		64,84	3047894,99	
63,9	2454708,92		65,21	3318944,576	
62,3	1698243,65		64,23	2648500,139	
62,2	1659586,91		62,89	1945360,082	
62,6	1819700,86		62,12	1629296,033	
64,2	2630267,99		63,03	2009092,813	
73,3	21379620,9		63,93	2471724,145	
63,5	2238721,14		73,06	20230191,79	
63,6	2290867,65		68,99	7925013,305	
61,4	1380384,26		64	2511886,432	
63,4	2187761,62		61,75	1496235,656	
67,1	5128613,84		63,34	2157744,409	

Tabla 3: Resultados obtenidos del ensayo. Imagen de elaboración propia

Observaciones

Los niveles de presión sonora (dBA) medidos por ambos dispositivos son bastante comparables en la mayoría de los casos, con algunas diferencias que podrían ser atribuibles a variaciones en las condiciones de medición o a la sensibilidad de los dispositivos. Por ejemplo, el dispositivo SVAN 958A registró un rango de niveles de dBA que varía entre 61,4 dBA y 73,3 dBA, mientras que el otro dispositivo mostró mediciones dentro de un rango similar, pero con algunas variaciones en las mismas instancias de tiempo.

La comparación con el sonómetro comercial SVAN 958A demuestra que nuestro sonómetro es capaz de ofrecer resultados precisos y confiables en una variedad de entornos. Sin embargo, es importante tener en cuenta las limitaciones inherentes a un diseño casero en términos de calibración y certificación, por lo que se recomienda un proceso de calibración regular para mantener la precisión del dispositivo a lo largo del tiempo.

Transmisión

Una vez finalizadas las pruebas del micrófono, proseguimos a realizar las pruebas correspondientes para la transmisión de los datos. Estas pruebas se dividieron en dos fases principales, una prueba de distancia de envío de datos utilizando paquetes de diferentes tamaños y una prueba con el dispositivo completo, donde se evaluó el envío de datos medidos a diversas distancias.

Prueba de distancia con paquetes de diferentes tamaños

En la primera fase de las pruebas, se implementó una evaluación centrada en la transmisión de datos mediante paquetes de diferentes tamaños. Se seleccionaron tres tipos de paquetes, con tamaños decrecientes, para examinar cómo la reducción en el tamaño del paquete afectaba la confiabilidad de la transmisión. Estas pruebas se realizaron en un entorno urbano, en la ciudad, donde las condiciones incluyeron la presencia de interferencias como edificios y señales espurias. Durante estas pruebas, el dispositivo permaneció dentro del campo de visión, asegurando una línea de vista directa entre el transmisor y el receptor.

Distancia [metros]	Tamaño de mensaje					
	200 bits		700 bits		1344 bits	
	Cantidad	%	Cantidad	%	Cantidad	%
140	60	100	60	100	59	98.33
300	60	100	60	100	59	98.33
400	56	93.33	54	90	52	86.67

Tabla 4: Ensayo de envío de paquetes. Imagen de elaboración propia

Los resultados mostraron que, a medida que el tamaño del paquete disminuía, la confiabilidad en la transmisión mejoraba notablemente. Esto se debe a que los paquetes más pequeños son menos propensos a sufrir errores durante la transmisión, lo que reduce la posibilidad de pérdida de datos. Sin embargo, también se observó que la disminución en el tamaño del paquete incrementaba el número de transmisiones necesarias para enviar la misma cantidad de información, lo cual podría impactar en la eficiencia general del sistema en aplicaciones de largo plazo.

Prueba con el Dispositivo Completo

La segunda fase de las pruebas consistió en evaluar el rendimiento del dispositivo completo, que incluye el sensor de ruido ambiental, el micrófono digital, el sensor BMP280 y el módulo de comunicación LoRa, bajo condiciones operativas reales. En esta prueba, se midió la capacidad del dispositivo para enviar datos recogidos a diferentes distancias en la misma ciudad, pero con el dispositivo ubicado fuera del campo de visión directo. Esto introdujo un mayor desafío en la transmisión debido a los obstáculos físicos y las interferencias urbanas.

Origen: Quinto piso, segunda torre UTN	Distancia [metros]	Nodo2	Nodo3
11 de abril	20	Si	Si
Caronti y Alem	200	Si	Si
Casanova y Alem	285	Si	Si
Alem 725	320	No	No
Alem 717	303	No	No
Caronti 534	240	No	Si
Caronti 584	320	No	No

Tabla 5: Ensayo realizado en la FRBB. Imagen de elaboración propia

Durante estas pruebas, se detectó que la confiabilidad del dispositivo en el envío de datos disminuía a medida que aumentaba la carga de trabajo del procesador. Al realizar múltiples tareas, como la adquisición de datos de los sensores y su procesamiento, el dispositivo experimentó lapsos en los que las transmisiones no se completaron

exitosamente, a pesar de que, en pruebas previas, sin la carga completa del sistema, los envíos habían sido exitosos.

Observaciones

De las pruebas realizadas, se concluye que la confiabilidad del dispositivo es mayor cuando se envían paquetes aislados, especialmente si estos son de menor tamaño. No obstante, al integrar todas las funciones del dispositivo y operar en condiciones reales, donde la potencia del procesador es compartida entre diversas tareas, se observa una disminución en la eficiencia del envío de datos. Este comportamiento resalta la importancia de optimizar el uso de los recursos del procesador y de considerar la posibilidad de segmentar las tareas o ajustar la frecuencia de envío para mejorar la confiabilidad en la transmisión de datos en aplicaciones de campo.

Este análisis nos permite identificar áreas clave para futuras mejoras en el diseño del sistema, con el fin de garantizar un rendimiento óptimo bajo diversas condiciones de operación.

Dispositivo completo

Una vez realizada las tres pruebas anteriores, realizamos la prueba de los dispositivos en funcionamiento. Para esto ubicamos los dispositivos en dos ambientes distintos y los dejamos haciendo mediciones por periodos prolongados. Esto para poder tener un control y además ver el funcionamiento de este a lo largo del tiempo, esta prueba además nos funcionó para poder realizar un control de cómo se tomarían los datos provenientes en la plataforma Azure y cómo podríamos realizar los distintos dashboard que posterior serían la visualización de estos mismos datos. Para eso en distintos días dejamos los dispositivos funcionales por varias horas por lo que pudimos conseguir estas muestras que serán presentadas a continuación.



Figura 17: Dashboard realizada en Azure Central IoT. Imagen extraída de [10]

Mejoras

En función de las pruebas realizadas y el análisis de los resultados obtenidos, se identifican varias mejoras que pueden ser implementadas para optimizar el rendimiento y la confiabilidad del dispositivo diseñado para la medición ambiental. Estas mejoras buscan no solo resolver los desafíos encontrados, sino también aumentar la funcionalidad y la eficiencia del dispositivo en su aplicación en entornos reales.

Integración de una Tarjeta SIM para Envío de Datos

Actualmente, el dispositivo se comunica a través de un único Gateway mediante tecnología LoRa. Para mejorar la flexibilidad y cobertura del dispositivo, proponemos la integración de una tarjeta SIM que permita el envío de datos utilizando redes móviles. Esta adición permitiría al dispositivo operar en áreas donde la cobertura es limitada o inexistente, garantizando así una transmisión de datos más robusta y confiable. Además, al utilizar la red celular, se eliminaría la dependencia de un único Gateway, aumentando la adaptabilidad del dispositivo en diversos escenarios de despliegue.

Modificación del Paquete de Datos al Menor Tamaño Posible

Una de las conclusiones de nuestras pruebas fue que la confiabilidad de la transmisión mejora al reducir el tamaño de los paquetes de datos. Por lo tanto, se propone optimizar el formato del paquete para que sea lo más compacto posible, sin comprometer la integridad de la información transmitida. Esta reducción minimiza las posibilidades de errores durante la transmisión y mejora la eficiencia energética del dispositivo, ya que se requieren menos recursos para enviar y recibir paquetes más pequeños.

Incorporación de Actualización OTA (Over-The-Air)

Para facilitar el mantenimiento y actualización del dispositivo sin la necesidad de desmontarlo, se propone agregar la capacidad de realizar actualizaciones OTA. Mediante la conexión Wi-Fi o la tarjeta SIM, el dispositivo podría recibir actualizaciones de firmware de manera remota. Esta funcionalidad es crucial para corregir errores, mejorar el software, o añadir nuevas características, todo sin interrumpir la operación del dispositivo o requerir intervención física.

Implementación de LoRaWAN para Comunicación en Malla

Otra mejora significativa es la implementación del protocolo LoRaWAN para permitir la formación de una red en malla entre múltiples dispositivos. Este enfoque facilitaría la comunicación entre dispositivos cercanos, permitiendo que un dispositivo retransmita paquetes que no hayan sido enviados exitosamente por otro. Esta funcionalidad mejora la cobertura de la red, incrementa la resiliencia del sistema ante fallos en la transmisión y optimiza el uso de los recursos de comunicación.

Antenas de Mayor Ganancia

Para mejorar el alcance y la calidad de la transmisión de datos, se recomienda la incorporación de antenas de mayor ganancia en el dispositivo. Estas antenas mejorarían la recepción y emisión de señales, especialmente en entornos urbanos donde las interferencias y obstáculos pueden afectar significativamente la calidad de la señal. Una mejor ganancia en las antenas también contribuiría a extender la cobertura efectiva del dispositivo, asegurando una transmisión más confiable a mayores distancias.

Incorporación de Paneles Solares y Doble Batería

Finalmente, se sugiere añadir un sistema de alimentación basado en paneles solares junto con una doble batería para aumentar la autonomía del dispositivo. Esta configuración permitiría que el dispositivo funcione de manera continua y autosuficiente, incluso en ubicaciones remotas donde el acceso a la electricidad es limitado o inexistente. La doble batería garantizaría un suministro de energía estable, mientras que los paneles solares recargarían las baterías durante el día, extendiendo significativamente la vida útil del dispositivo en campo.

La implementación de estas mejoras proporcionará al dispositivo IoT de medición ambiental una mayor robustez, flexibilidad y autonomía. Estas actualizaciones no solo

mejorarán la confiabilidad de la transmisión de datos, sino que también aumentarán la vida útil y la eficiencia del dispositivo, haciéndolo más adecuado para su despliegue en una amplia gama de entornos y aplicaciones. Al adoptar estas mejoras, el dispositivo estará mejor equipado para enfrentar los desafíos de la medición ambiental en tiempo real y contribuirá a la creación de redes de sensores más eficientes y resilientes.

Conclusión

El proyecto de desarrollo de un medidor de ruidos ambiental utilizando tecnología IoT ha sido un ejercicio exhaustivo de integración de hardware y software, que ha puesto en práctica los conocimientos adquiridos a lo largo de la carrera en ingeniería electrónica. Al utilizar un micrófono digital ICS43434 I2S y un sensor BMP280 para capturar datos de ruido, temperatura y presión, se logró implementar un sistema de medición robusto y preciso. La inclusión de LoRa como tecnología de comunicación permitió establecer una conexión fiable y de largo alcance entre los dispositivos, lo que es esencial para aplicaciones de monitoreo ambiental en áreas extensas o de difícil acceso.

Las pruebas realizadas en diferentes entornos, tanto en campo abierto como en zonas urbanas con interferencias, demostraron la capacidad del sistema para operar en condiciones variadas. Aunque se encontraron desafíos en la transmisión de datos debido a obstrucciones y señales espurias, el dispositivo demostró ser capaz de mantener una transmisión efectiva dentro del rango especificado, reafirmando la viabilidad de utilizar LoRa en aplicaciones IoT para medición ambiental. Estos resultados resaltan la importancia de considerar las condiciones ambientales y geográficas en el diseño de sistemas de comunicación inalámbrica.

En conclusión, el proyecto ha logrado cumplir con los objetivos planteados, demostrando la viabilidad de un sistema IoT para la medición de ruido ambiental en tiempo real. Las experiencias y desafíos encontrados durante el desarrollo y las pruebas han proporcionado una valiosa perspectiva sobre las consideraciones técnicas necesarias para diseñar y desplegar sistemas electrónicos en entornos del mundo real. Este proyecto no solo refuerza las competencias técnicas en ingeniería electrónica, sino que también abre la puerta a futuras investigaciones y mejoras en el monitoreo ambiental mediante tecnologías IoT.

Bibliografía

- [1] IEC 61672-1, IEC 61672-1:2013, Electroacoustics - Sound level meters - Part 1: Specifications
- [2] IRAM 4062:2016, Ruidos molestos al vecindario, Método de medición y calificación
- [3] ISO 1996-1:2016(E), Acoustics — Description, measurement and assessment of environmental noise — Part 1: Basic quantities and assessment procedures
- [4] ISO 9613-2:1996(E), Acoustics - Attenuation of sound during propagation outdoors - Part 2: General method of calculation
- [5] Datasheet correspondiente ESP32. <https://www.espressif.com/en/products/socs/esp32>
- [6] ICS43434 I2S Digital Microphone. <https://www.tindie.com/products/onehorse/ics43434-i2s-digital-microphone/#specs>
- [7] SX1276 LoRa Connect™ 137MHz to 1020MHz Long Range Low Power Transceiver. <https://www.semtech.com/products/wireless-rf/lora-connect/sx1276>
- [8] Pressure sensor BMP280. <https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/>
- [9] DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal <https://www.analog.com/media/en/technical-documentation/data-sheets/ds3231.pdf>
- [10] Azure IoT Central. <https://azure.microsoft.com/es-es/products/iot-central>