TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol. 19, No. 1, Enero-Abril, 2020, p.48- 60 ISSN 1729-3804

EQUIPO DE ADQUISICIÓN DE AUDIO MULTICANAL USANDO RASPBERRY PI Y PYTHON

M. Sc. Ing. Pedro García Otero[1]

Centro de Investigación y Desarrollo Técnico (CIDT), Cuba
email: pedroeo@nauta.cu

RESUMEN

Los Sensores Acústicos Distribuidos (SAD) son ampliamente usados para la recolección de información acústica destinada a investigaciones de planificación urbana, control ambiental y seguridad pública. El surgimiento de nuevos algoritmos de bajo costo computacional y de dispositivos de computo más pequeños, económicos y asequibles, han permitido que investigadores y científicos desarrollen sensores acústicos de bajo costo que pueden ser desplegados en Redes de Sensores Acústicos (RSA). En este trabajo se presenta el diseño y desarrollo de un equipo de adquisición de ocho canales de audio basado en una Raspberry Pi, que puede ser empleado como nodo en una RSA. Para el desarrollo del software que se ejecuta en la Raspberry Pi se empleó el lenguaje de programación Python, obteniéndose una biblioteca de funciones que permite el trabajo con varias tarjetas de audio de forma simultánea. Las pruebas realizadas demostraron el poder computacional y la capacidad de una Raspberry Pi para ser usada como núcleo de procesamiento de un nodo de una RSA.

PALABRAS CLAVES: Audio, Raspberry Pi, Python, Adquisición, Sensor Acústico.

MULTICHANNEL AUDIO ACQUISITION DEVICE USING RASPBERRY PI AND PYTHON

ABSTRACT

Acoustic Distributed Sensors (ADS) are widely used for the gathering of acoustic information dedicated to the research of urban planning, environmental control, and public security. The emergence of new algorithms of computational low cost and computing devices more small, economic and affordable, has created the possibility of developing a low-cost acoustic sensor that can be deployed in an Acoustic Sensor Network (ASN). In this work, the design and development of a multichannel audio acquisition device based on Raspberry Pi are described. This device can be used as a node of an ASN. For the development of the software that is executed in the Raspberry Pi, the programming language Python was used, being obtained a library of functions that allows the work with several audio cards in a simultaneous way. The made tests have shown the Raspberry Pi as a powerful and affordable computing core for ASN's node.

KEY WORDS: Audio, Raspberry Pi, Python, Acquisition, Acoustic Sensor.

1. INTRODUCCIÓN

El concepto de ciudad inteligente (Smart City) y el monitoreo de parámetros acústicos en ciudades son áreas de investigación que han atraído la atención de los científicos e investigadores durante la última década [1]. El crecimiento poblacional en los grandes asentamientos urbanos trae consigo la necesidad de crear nuevas estrategias de planificación urbana, de control ambiental y seguridad pública. La recolección de información acústica es ampliamente usada para este objetivo, los cuales pueden ir desde el control de tráfico vehicular, análisis medioambiental de contaminación acústica, detección de disparos de armas de fuego, detección de hechos delictivos y supervisión acústica en áreas de alta seguridad. En la Fig. 1 se puede observar un nodo de procesamiento del Sistema SENTRI destinado a la detección y localización de disparos de armas de fuego en entornos urbanos.

Figura 1: Sistema SENTRI de la compañía estadounidense Safety Dynamic.

En la última década ha surgido un nuevo concepto de computadora. Estos nuevos dispositivos denominados como computadoras de placa simple (SBC, por sus siglas en inglés), son más pequeños, económicos y asequibles que una computadora clásica. Las SBC han demostrado su poder computacional unido a su escalabilidad para grandes proyectos [2]. Existen gran variedad de SBC en el mercado con diferentes características de conectividad, capacidad de procesamiento, tamaño y consumo. Raspberry Pi, BeagleBone, Arduino, ODroid, son algunos de los más conocidos.

La plataforma de Internet de las Cosas (IoT, por sus siglas en inglés) y las SBC, han permitido incrementar de forma sustancial y a un bajo costo las capacidades de los sistemas SAD [3]. La creación de equipos o nodos que adquieren la información de uno o varios sensores, procesan la información en tiempo real y transmiten los resultados a través de redes de datos. Mediante esta plataforma se ha permitido la creación de Redes de Sensores Acústicos (RSA), abriendo nuevas posibilidades y desafíos para hacer del sonido una fuente valiosa de información para el desarrollo de nuevos servicios.

El presente trabajo tiene como objetivo obtener un equipo de adquisición de 8 canales de audio basado en una Raspberry Pi. Este equipo tiene la funcionalidad de adquirir y transmitir la información de audio de cada canal hacia un servidor de procesamiento remoto empleando una red de datos Ethernet. En caso de detectar una falla de conexión de red, el equipo almacena la información de audio de forma local, permitiendo su extracción, una vez restablecida la comunicación, empleando un servidor web.

El artículo está dividido en las siguientes secciones. Primero, los materiales y métodos para el diseño del prototipo son presentados. Se explica la selección de los materiales basados en los requerimientos. Seguidamente se presenta el ensamble del prototipo, así como el desarrollo del programa de aplicación y la implementación del servidor web. Finalmente se presenta la discusión de los resultados y las conclusiones.

2. CONTENIDO

Durante el desarrollo de esta investigación fueron empleados métodos teóricos, que permitieron arribar a conclusiones procesando la información obtenida con la revisión bibliográfica centrada en los temas objeto de estudio, se aplicaron métodos empíricos, que permitieron realizar simulaciones y comprobaciones. Además, se empleó la sistematización para el procesamiento y aplicación de la información recopilada y el análisis y síntesis para la evaluación de los resultados. Se emplearon también métodos como la observación, la medición y la experimentación para comprobar el funcionamiento del equipo desarrollado.

Requerimientos de diseño

Los requerimientos que debe cumplir el equipo a diseñar son los siguientes:

- Bajo costo. Permitirá la reproducción del equipo, obteniéndose nodos de adquisición que tributan a una red de sensores acústicos asequible con una buena relación calidad/costo.

- Protección para aplicaciones en exteriores con condiciones ambientales adversar y con un régimen de trabajo continuo.
- Capacidad de conexión a una red de datos Ethernet.
- Parámetros de la adquisición de audio:
  - Frecuencia de muestreo: 11025Hz.
  - Resolución: 16 bits/muestra.
- Capacidad de tele-alimentar a los micrófonos empleando el mismo par trenzado usado para la transmisión de la señal de audio.

Propuesta de diseño

En la Fig. 2 se muestra un diagrama en bloques del nodo de adquisición propuesto y su interconexión con los otros elementos del sistema. Como se puede observar el nodo de adquisición está compuesto por tres etapas fundamentales. La primera es la etapa de acondicionamiento de las señales de audio que se obtienen de los sensores microfónicos. La segunda es la etapa de adquisición, la cual se encarga de convertir las señales analógicas de audio al dominio discreto y por último la etapa de procesamiento, que se encarga de adquirir las muestras digitales encapsularlas en un paquete de datos TCP y trasmitirlas hacia el servidor remoto, además realiza el almacenamiento de la información de audio si detecta fallas en la conexión de red.

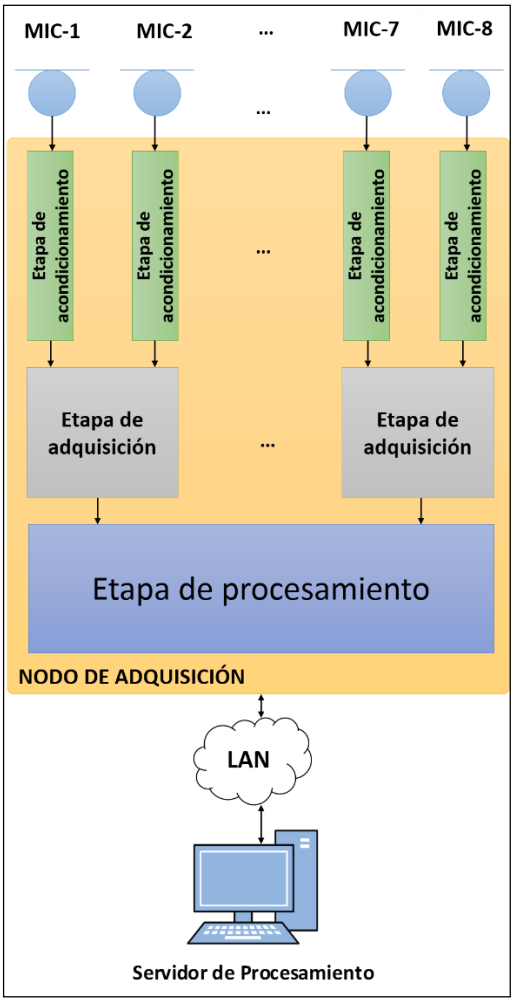

Figura 2: Diagrama en bloques y de interconexión del nodo de adquisición propuesto.

Etapa de acondicionamiento

La etapa de acondicionamiento está compuesta por un circuito de tele-alimentación que permite energizar el micrófono y recibir la señal de audio por un mismo par trenzado (cable de dos vías). Además, se empleó el transformador de audio 42TL002-RC para obtener un aislamiento galvánico en las entradas de audio del equipo, evitando así los problemas de acoplamiento que pudieran surgir (véase Fig. 3).

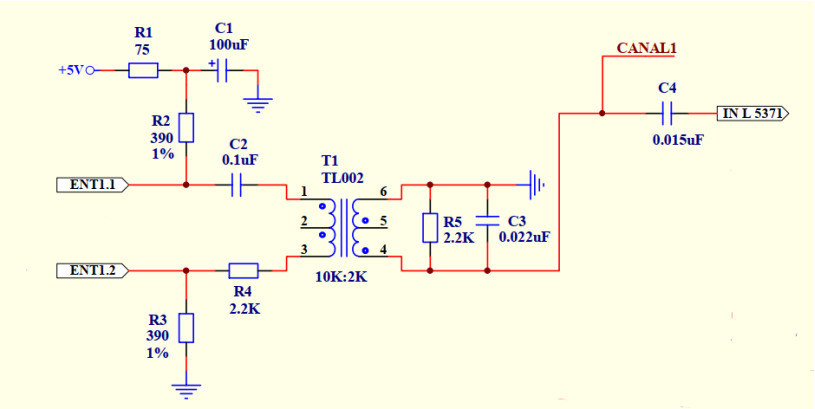

Figura 3: Esquema eléctrico de la etapa de acondicionamiento.

Etapa de adquisición

Debido a que ninguno de los modelos de la familia Raspberry Pi presenta conversor A/D, fue necesario identificar un circuito integrado (CI) con esta funcionalidad y que, además, fuera compatible con la Raspberry Pi y con los requerimientos de adquisición de audio antes expuestos. El CI seleccionado fue el AK5371 [4] de la firma AKM (Fig. 4). Este CI es un conversor A/D estéreo con una interfaz USB clase Audio incluida, por lo cual el sistema operativo de la Raspberry Pi lo reconoce como una tarjeta de sonido de dos canales.

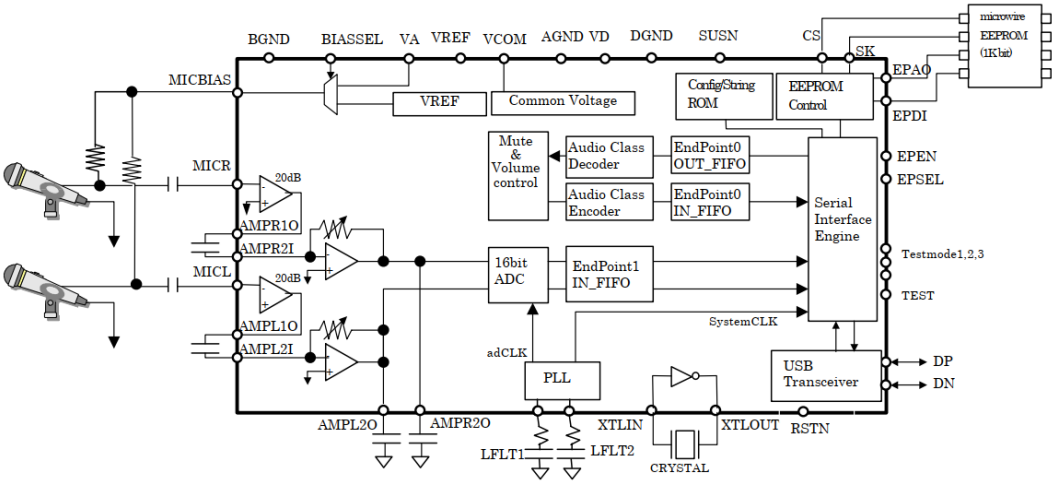

Figura 4: Diagrama en bloques del CI AK5371.

Entre sus principales características se encuentran:

- Conversor A/D con una relación Señal/Ruido de 84dBA (44100Hz).
- Presenta un preamplificador de ganancia fija de 20dB y un amplificador de ganancia programable de +24dB a -31dB con pasos de 1dB.
- Soporta 5 frecuencias de muestreo: 8000Hz, 11025Hz, 22050Hz, 44100Hz y 48000Hz.
- Alimentación de 3.3V.
- Soporta personalización del descriptor USB mediante la conexión de una memoria EEPROM.

Una de las características claves a la hora de la selección de este CI fue la posibilidad de personalizar el descriptor USB de la misma, pudiéndose modificar el ID del Vendedor y el ID del producto. Esto permite identificar en la

TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol. 19, No. 1, Enero-Abril, 2020, p.48- 60 ISSN 1729-3804

Raspberry Pi cada tarjeta durante el inicio del sistema operativo y por consiguiente el canal de audio que tiene conectado físicamente.

Etapa de procesamiento

El bloque fundamental del nodo de adquisición es la etapa de procesamiento, la cual es usada para el procesamiento de los datos adquiridos, su transmisión y almacenamiento. Para lograr los requisitos propuestos el diseño del equipo se basó en el empleo de una Raspberry Pi 3 Modelo B. Esta SCB ofrece: buena capacidad de computo, alta versatilidad y bajo precio. Otro de los beneficios del uso de la Raspberry Pi es la posibilidad de trabajar sobre un sistema operativo libre. En el caso de este proyecto se empleó la distribución Raspbian Buster, una distribución del sistema operativo GNU/Linux para Raspberry Pi. La conectividad a una red de datos que ofrece esta SCB permite trabajar con ella de forma remota empleando SSH (Secure SHell). A través de una terminal de comando es posible acceder al dispositivo y actualizar el programa de aplicación desarrollado, dar mantenimiento de software y chequear el estado de la SBC. Para el trabajo de las tarjetas de audio se empleó como base la librería ALSA que forma parte del sistema operativo y permite controlar y administrar los dispositivos de audio que se encuentren conectados.

Para la transmisión continua de la información mediante una red de datos Ethernet la Raspberry Pi 3 presenta conectividad alámbrica mediante un conector Ethernet y conectividad inalámbrica empleando Wi-Fi. Para este proyecto fue empleado una conexión alámbrica siguiendo dos objetivos específicamente: evitar posibles interferencias en la comunicación y tener la posibilidad en un futuro de emplear alimentación POE (Power Over Ethernet) para energizar el nodo de adquisición de forma remota.

Ensamble de las partes

Para el ensamble de todos los componentes que conformar al equipo, se diseñaron dos placas de circuito impreso. La primera placa incluye la electrónica de la etapa de acondicionamiento de señal y los conectores de las entradas de audio. La segunda placa se encarga de la etapa de adquisición y de regular y entregar los niveles de voltaje que emplean cada una de estas etapas y la tele-alimentación de los micrófonos. Por último, se seleccionó un gabinete plástico con protección para exteriores para ubicar la electrónica. En las figuras 5 y 6 se muestran vistas del resultado final. Los cables rojos y azul de la Fig. 5 muestran la conexión USB entre los CI AK5371 y la Raspberry Pi (terminales P36, P41, P42, P43, P44; P45, P46 y P47 de la Raspberry Pi).

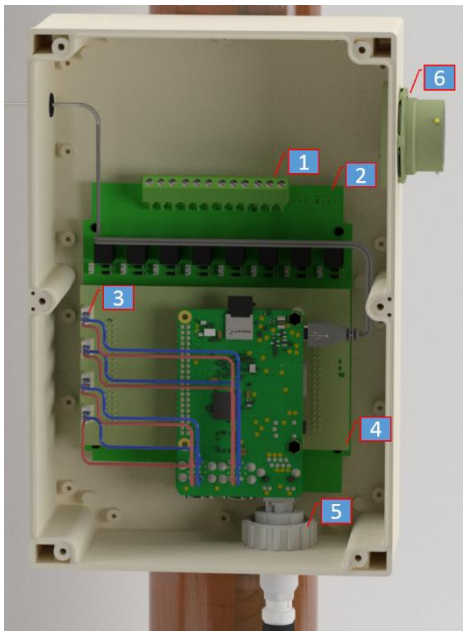

Figura 5: Vista del ensamble con el gabinete abierto.

Leyenda:

1. Conectores de entrada de audio.
2. Placa de acondicionamiento.
3. Conexión USB entre la Raspberry Pi y los CI AK5371
4. Placa de adquisición.
5. Conector de red con protección para exteriores.
6. Conector para la entrada de los pares trenzados.

Figura 6: Vista del ensamble con el gabinete cerrado.

Desarrollo del programa de aplicación e implementación del servidor web

El lenguaje de programación empleado para el desarrollo del programa de aplicación fue Python en su versión 3.7. Python es un lenguaje de programación interpretado, con gran aceptación a nivel internacional y con una extensa comunidad de desarrollo que aportan cada día nuevas bibliotecas y soluciones a problemas de la sociedad. Como ya se mencionó, para lograr la interacción con las tarjetas de audio se empleó el bloque de captura de audio de ALSA, el cual emplea componentes del kernel de Linux con el objetivo de proveer al sistema con funcionalidades de control de audio.

El intérprete de Python no contiene en su biblioteca estándar un módulo para trabajar directamente con ALSA, para ello, la comunidad de desarrollo, ha implementado diferentes módulos. Uno de los más usados es PyAudio [5], el cual está diseñado para enlazar a Python con la biblioteca PortAudio. PortAudio está desarrollada en C++ y permite realizar el control de dispositivos de audio en diferentes sistemas operativos incluidos Windows, Macintosh y Unix/Linux.

Durante el proceso de investigación se pudo determinar que, para lograr la adquisición de los 8 canales de forma simultánea es necesario emplear el módulo de multiproceso de Python (multiprocessing) [6]. La aplicación está conformada por un hilo principal que se encarga de la etapa de inicialización de los restantes procesos, de la configuración del sistema, de la transmisión de la información a través de la red empleando una conexión TCP y del manejo de errores. Asociados al hilo principal existen cuatro procesos, uno por cada tarjeta de audio, que se encargan de manejar la adquisición, procesamiento y almacenamiento de la información de audio. Para realizar el almacenamiento de la información en ficheros de audio con formato WAV se empleó el módulo wave. La comunicación de estos procesos con el hilo principal es mediante una cola para la transmisión de comandos, desde el hilo principal hacia el proceso, y una cola de transmisión de datos, desde los procesos hacia el hilo principal (véase Fig. 7).

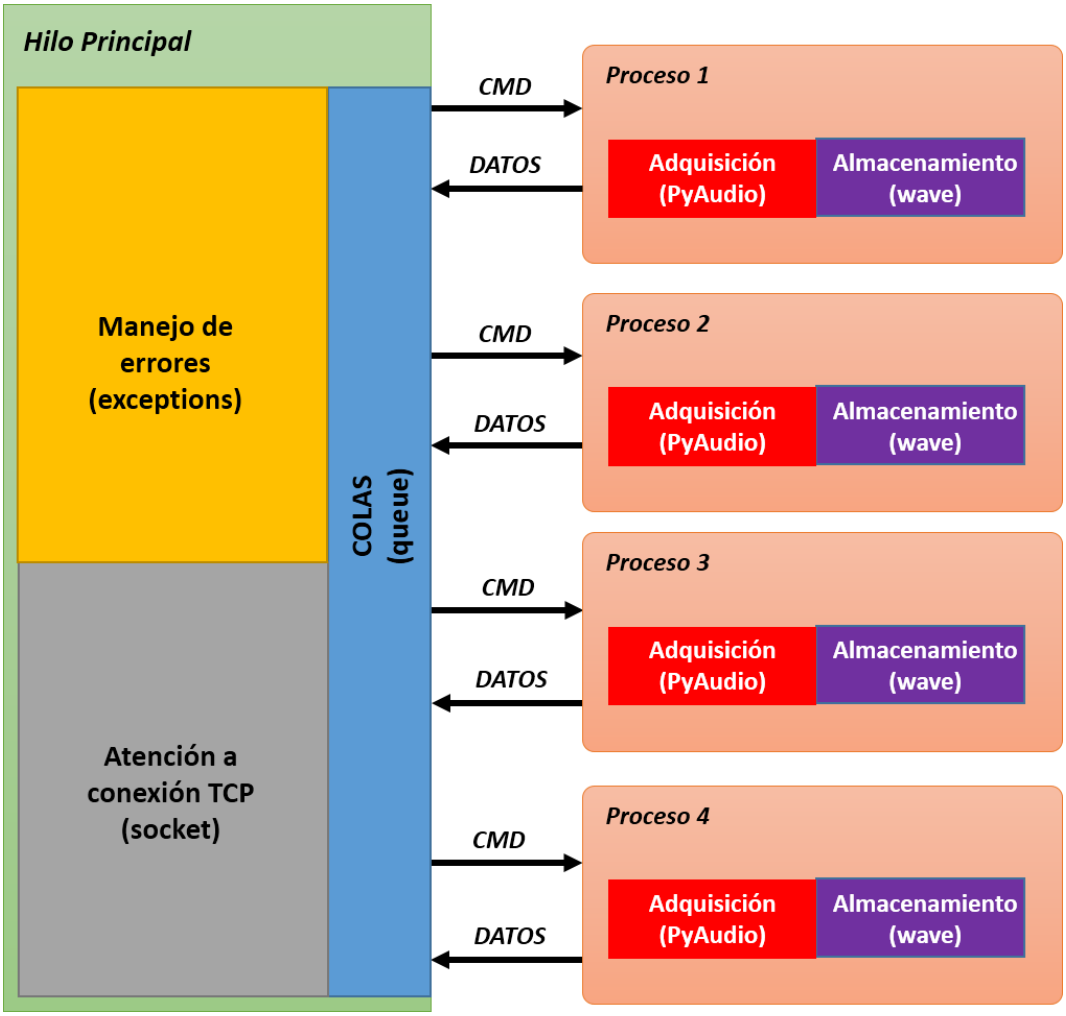

Figura 7: Estructura de hilos y procesos del firmware desarrollado.

Hilo principal

En la Fig. 8 se muestra el diagrama de flujo del hilo principal. Sus principales funciones son:

- Inicialización y ejecución de los procesos de adquisición. Estos procesos son inicializados mediante la creación de un objeto de la clase *DaqAudio*. En la siguiente sección se explica en mayor detalle esta clase.
- Inicialización de la conexión TCP mediante el empleo del módulo *socket*.
- Manejo de errores.
- Recepción de la información de los canales de audio enviada por los procesos de adquisición, conformación del paquete de datos y su transmisión por red empleando TCP.
- Atención y desencadenamiento de las tareas asociadas a los comandos de inicio y fin de transmisión de información enviados por el servidor de procesamiento remoto.

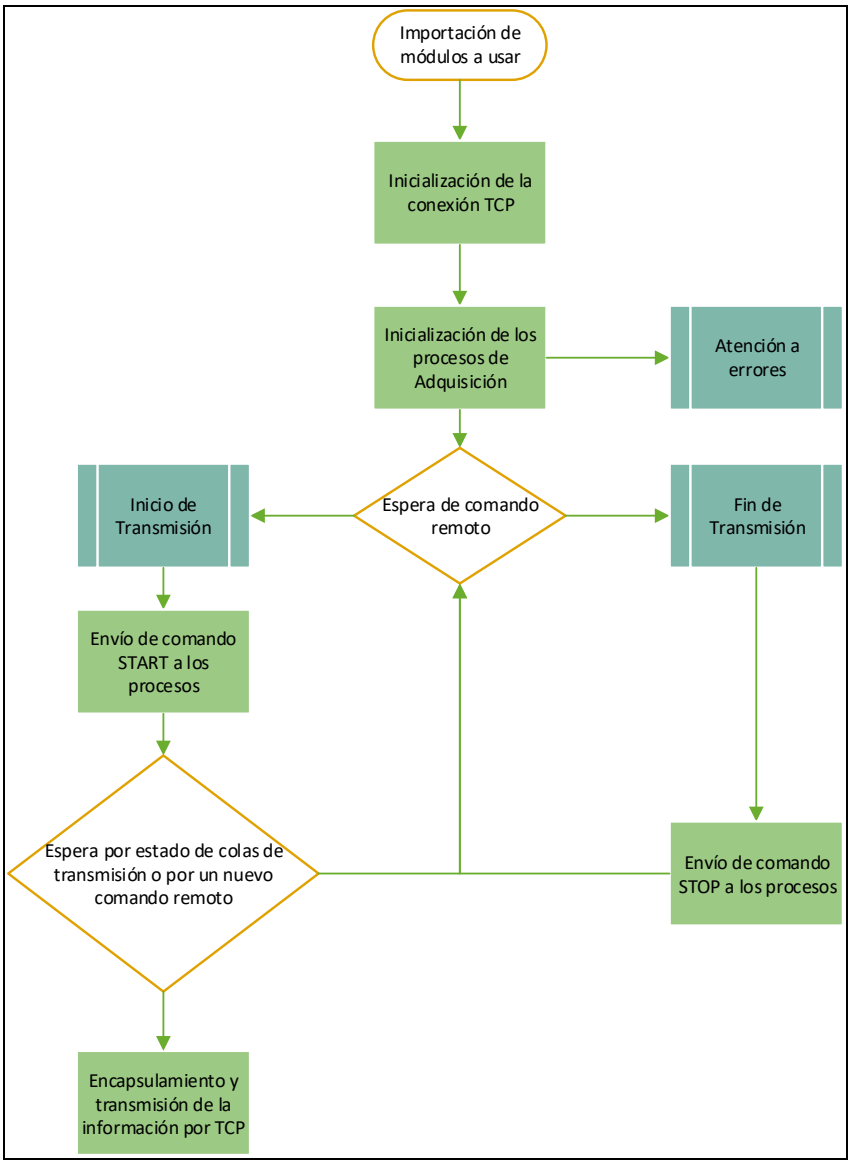

Figura 8: Diagrama de flujo del hilo principal.

Proceso de adquisición, procesamiento y almacenamiento

En la Fig. 9 se muestra el diagrama de flujo del proceso de adquisición. Para este proceso se creó la clase *DaqAudio* que hereda de la clase *multiprocessing.Process* lo cual permite crear múltiples procesos de la misma clase. Sus principales funciones son:

- Comprobar estado de conexión de las tarjetas de audio (CI AK5371).
- Inicializar el proceso de adquisición de PyAudio.
- Adquirir, reorganizar y enviar los datos de los canales izquierdo y derecho por la cola de datos.
- Atender a la cola de comandos del hilo principal.
- Realizar el almacenamiento de los canales de audio ante la detección de un error de conexión de red por parte del hilo principal.

La función de lectura de *PyAudio* entrega los datos correspondientes a una adquisición de dos canales a 16 bits en una cadena de bytes donde cada dos bytes corresponde a una muestra de un canal (véase Fig. 10a). Para crear el paquete de datos que se transmite por red fue necesario separar las muestras de cada canal y crear una cadena donde primero estén las muestras del canal izquierdo y después las del canal derecho (véase Fig. 10b).

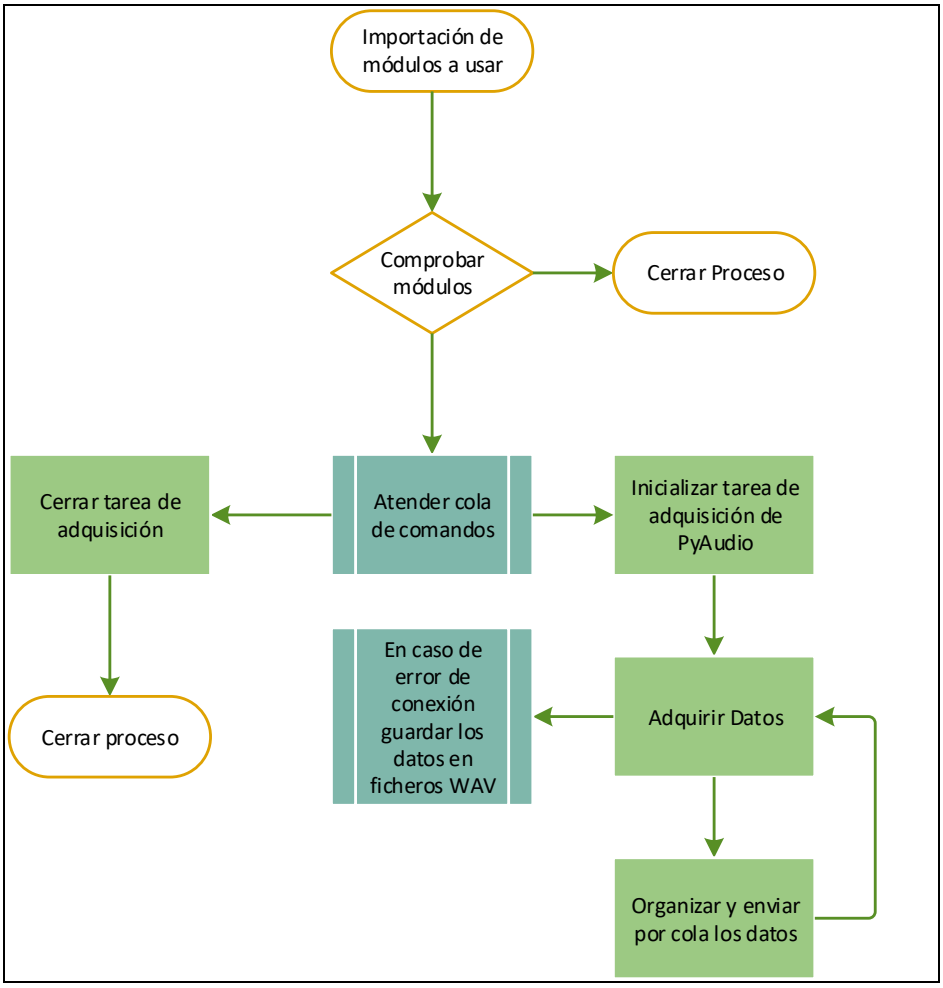

Figura 9: Diagrama de flujo del proceso de adquisición.

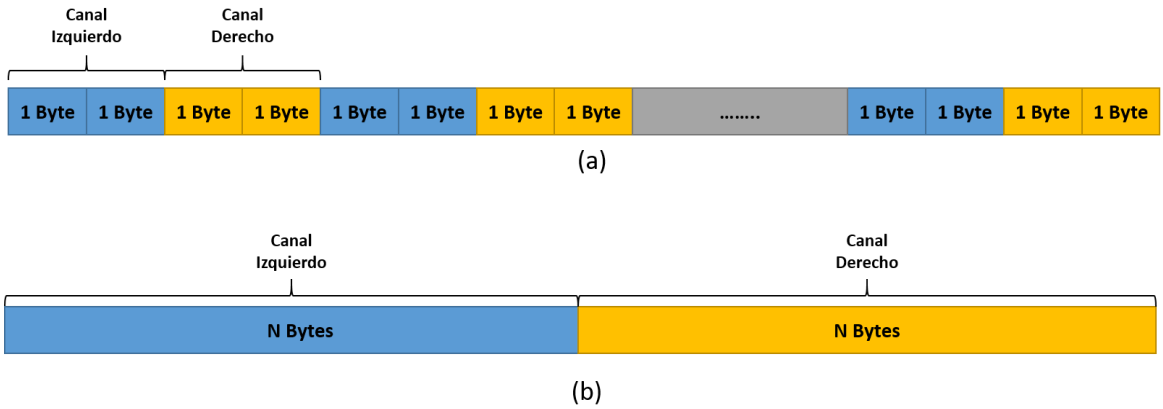

Figura 10: (a) Estructura de los datos obtenidos de la función de lectura de PyAudio. (b) Datos organizados para su transmisión.

Servidor Web

Para obtener la información de audio almacenada durante una falla de conexión se implementó un servidor web basado en el proyecto software FileRun [7] que se ejecuta sobre un servidor Apache. FileRun permite crear un servidor en la nube que se ejecuta en PHP. El acceso al contenido almacenado es mediante usuarios con reglas de acceso definidas por un administrador. El proceso de autentificación es a partir de un registro de usuarios que se encuentran en una base de datos creada con mysql. En la Fig. 10 se observa una vista de la página principal de este servidor, cargada en el navegador Mozilla.

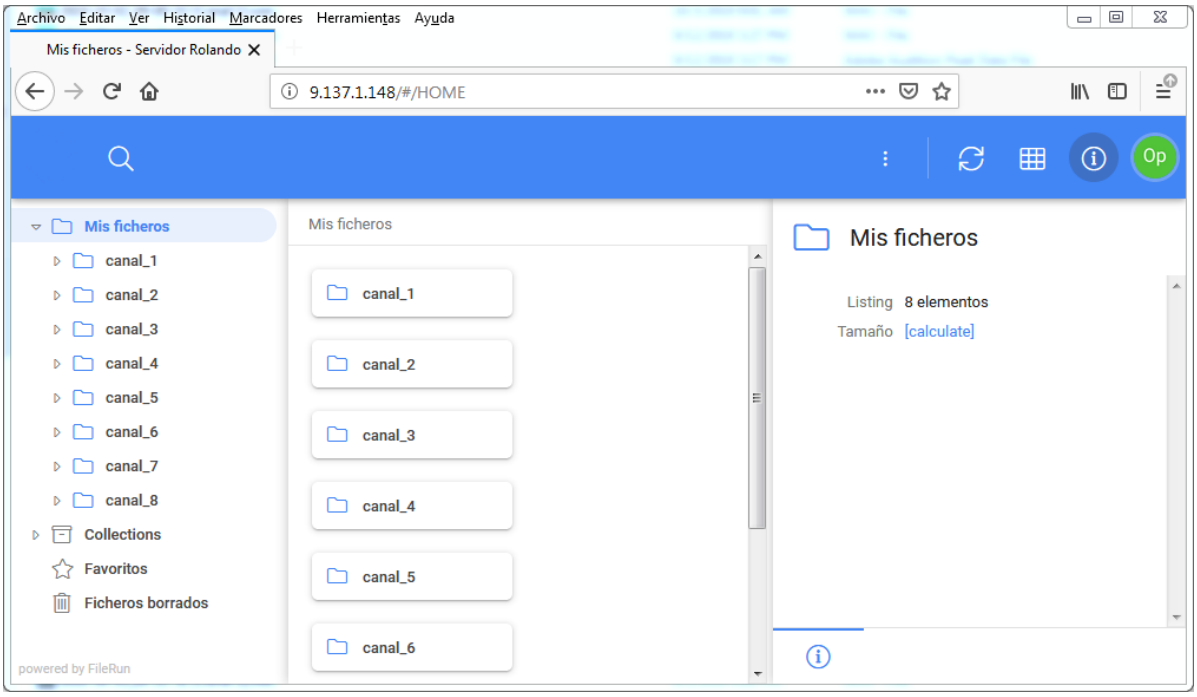

Figura 11: Vista de la página principal del servidor web FileRun.

3. VALIDACIÓN DE LOS RESULTADOS

Para la validación del equipo diseñado se realizaron un total de 3 pruebas. Estas pruebas son descritas a continuación.

Análisis del consumo de potencia

Para realizar el cálculo de consumo de potencia del equipo se empleó el diagrama de la Fig. 11. La medición de consumo de corriente se realizó en cuatro momentos diferentes. Primero con el equipo apagado, pero energizado. La segunda en el instante siguiente al haber encendido el equipo (etapa de inicialización del sistema operativo de la Raspberry Pi). La tercera medición se realizó con el equipo encendido, pero sin realizar transmisión de la información y la cuarta con el equipo adquiriendo y transmitiendo la información por la red. Los resultados obtenidos son mostrados en la Tabla 1, donde se puede concluir que el equipo diseñado puede ser alimentado empleando una fuente de alimentación estándar que se oferta para la Raspberry Pi 3.

Tabla 1: Resultado de medición de consumo de potencia.

Medición	Consumo de corriente (mA)
Prueba 1	103.0
Prueba 2	629.0
Prueba 3	398.0
Prueba 4	462.0

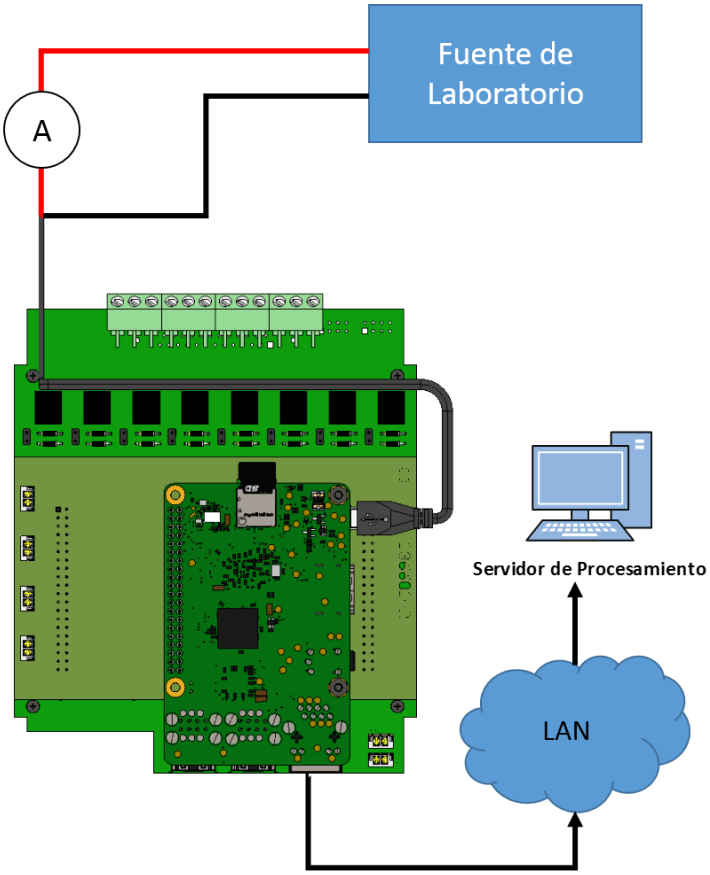

Figura 12: Esquema de conexión empleado para la medición de consumo de potencia.

Análisis de recursos de la Raspberry Pi

Para medir los recursos que consume la aplicación desarrollada se empleó el comando del sistema *top*, este comando brinda la información del uso de recurso de los procesos que se están ejecutando. La medición se realizó en el momento más crítico de la aplicación que es cuando se encuentra adquiriendo, transmitiendo la información de audio por red y el usuario descargando los ficheros almacenados en el servidor web. Los resultados fueron los siguientes:

- Uso de CPU: 18%.
- Uso de memoria RAM: 131MiB usados de 926.1MiB.

De esta prueba se puede concluir que es posible incluir otros algoritmos de procesamiento local en la Raspberry Pi sin comprometer el desempeño del equipo. Entre los algoritmos que se pueden incluir se encuentran:

- Filtros adaptativos para eliminación de ruido.
- Detección de huellas acústicas de interés.
- Algoritmos de localización de fuentes sonoras.

Comprobación de enlace de red e integridad de la información de audio adquirida

Para realizar esta prueba se desarrolló un instrumento virtual (VI, por sus siglas en inglés) de pruebas en LabVIEW 2018. En las figuras 13 y 14 se muestran el panel frontal y diagrama en bloques de este VI respectivamente. Como se puede apreciar en el diagrama en bloques, primeramente, se realiza la conexión TCP e inmediatamente se transmite un comando de inicio de trabajo. Seguidamente se entra en un lazo infinito que espera por la lectura por TCP de la cantidad de bytes asignada. El dato obtenido es decodificado separando la información por canal. Los datos de cada

canal son almacenados en ficheros WAV independientes. La información es mostrada en tiempo real empleando un indicador tipo *Waveform Graph*.

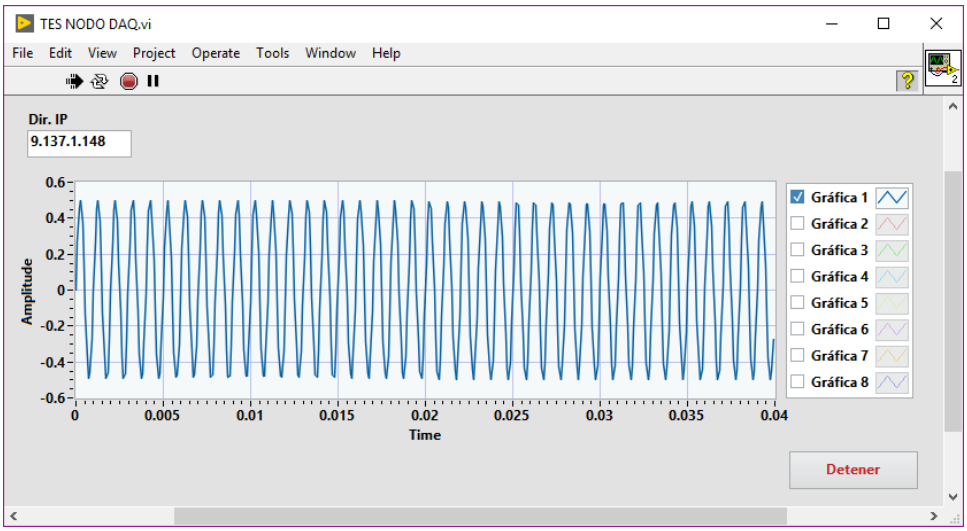

Figura 13: Panel frontal del VI de prueba.

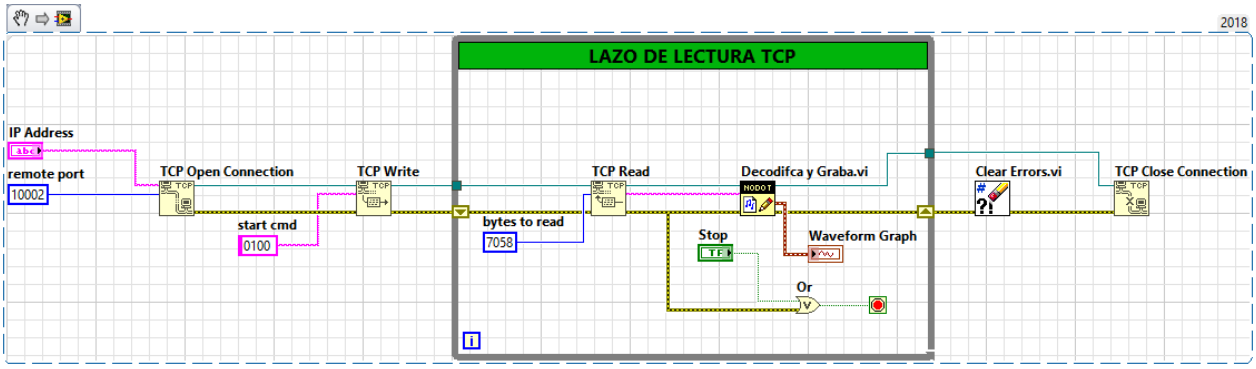

Figura 14: Diagrama en bloques del VI de prueba.

Esta prueba se realizó en dos etapas. Primeramente, se realizó la adquisición y transmisión de una señal sinusoidal de frecuencia 1000Hz. Las señales grabadas en la computadora empleando el VI desarrollado fueron analizadas buscando alguna discontinuidad o cambio de fase en el tono, para ello se empleó el programa de edición de audio *Adobe Audition CC 2015.* Los resultados obtenidos arrojaron que no existe discontinuidad alguna, lo que indica que no hay pérdida de muestras durante el proceso de adquisición y transmisión de la información de audio.

La segunda etapa consistió en el mantenimiento de forma continua de la transmisión durante 8 horas. Con esta prueba se buscó comprobar la estabilidad del enlace de red logrando resultados satisfactorios. Además, al finalizar esta prueba se realizó una desconexión física de la red de forma intencional. Se pudo comprobar que la aplicación al detectar un fallo en la red grabó la información de audio de forma local. En la Fig. 15 se muestra una de las grabaciones obtenidas.

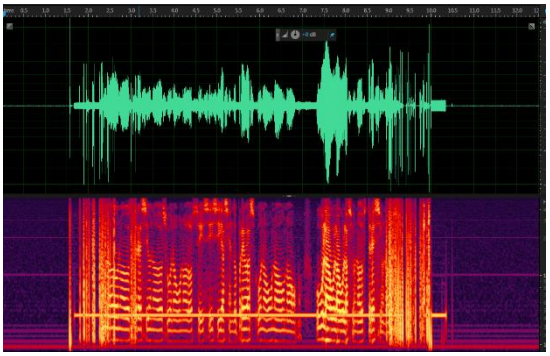

Figura 15: Forma de onda y espectro del fichero correspondiente al canal 1 creado durante la grabación local.

4. CONCLUSIONES

Con este trabajo se logró diseñar y desarrollar un equipo de adquisición de ocho canales de audio empleando una Raspberry Pi. La plataforma SBC y el lenguaje de programación empleado para el desarrollo del software ofrecieron algunas ventajas como una gran versatilidad, bajo precio de los componentes y un corto tiempo de desarrollo. Con esto se pudo obtener un equipo para uso en exteriores que puede formar parte de una red de sensores acústicos. La aplicación desarrollada consume menos del 20% de los recursos de cómputo de la Raspberry Pi, lo cual permitirá en un futuro incluir otros algoritmos de procesamiento, eliminando la necesidad de enviar el dato crudo hacia un servidor de procesamiento remoto.

REFERENCIAS

1. Healey, P.: "Urban Complexity and Spatial Strategies: Towards a Relational Planning for Our Times", Routledge: Abingdon, UK, 2006.
2. Noriega Linares, J.E. y Navarro Ruiz, J.M.: "On the Application of the Raspberry Pi as an Advanced Acoustic Sensor Network for Noise Monitoring", Electronics, 2016
3. Cobos, M. y Antonacci, F.: "Wireless Acoustic Sensor Networks and Application", Wireless Communication and Mobile Computing, Volumen 2017.
4. AKM: "AK5371 Technical Specifications", 2001.
5. Pham, H.: "PyAudio Documentation", 2006.
6. Hellman, D.: The Python 3 Standard Library by Example, Ed. Addison-Wesley, 2017.
7. FileRun: FileRun Documentation, 2019 disponible en web: http://www.filerun.com.
8. Virtanen, T., Plumbley, M.D. y Ellis, D.: Computational Analysis of Sound Scenes and Events, Ed. Springer, 2018.

SOBRE LOS AUTORES

Pedro García Otero, nació en La Habana, Cuba, en el año 1987. Se graduó en el año 2010 como Ingeniero en Telecomunicaciones y Electrónicas en el Instituto Superior Politécnico "José Antonio Echeverría". En el año 2014 obtiene el título de Máster en Ciencias en "Diseño de Sistemas Electrónicos" del mismo instituto. Desde el año 2010 hasta la actualidad trabaja como Investigador en el Centro de Investigación y Desarrollo Técnico del Ministerio del Interior. Desde sus inicios ha centrado su estudio e investigación al desarrollo de Sistemas de Instrumentación Virtual. Sus investigaciones actuales se centran en el desarrollo de sistemas de instrumentación virtual con el empleo de plataformas hardware de código abierto.