

CREACIÓN DE UN PROTOTIPO DE SENSOR ACÚSTICO CONECTADO DE BAJO COSTE PARA EL ANÁLISIS DEL CAMPO SONORO

PACS 43.50.Yw

Noriega Linares, Juan Emilio¹; Navarro Ruiz, Juan Miguel¹
1 Dpt Telecomunicación, Univ. Católica San Antonio de Murcia (UCAM)
Avda. de los Jerónimos, s/n, 30.107 Guadalupe (Murcia), España
jenoriega@ucam.edu, jmnavarro@ucam.edu

RESUMEN

En este estudio se describe la creación de un prototipo de sensor acústico realizado con elementos de bajo coste para su utilización en el análisis del campo sonoro. Se presenta el diseño del sensor, sus componentes, la metodología a seguir preparando los algoritmos y la conexión a la nube para compartir los resultados en tiempo real. Finalmente se aportan conclusiones sobre la experiencia en el diseño y las futuras mejoras e implementaciones del sensor.

ABSTRACT

In this paper, the prototyping of a low-cost acoustic sensor for its use in the analysis of the sound field is described. The sensor design and a report of the components are presented too. It is shown also the methodology followed to prepare the algorithms and the connection to the cloud, to share the results in real time. Finally, conclusions about the experience in the design and future works and implementations are provided.

INTRODUCCIÓN

En los últimos años ha surgido un nuevo concepto de mini ordenador, también referido como *system-on-a-chip*, SoC [1], con el rasgo distintivo de tener un bajo coste siendo accesible económicamente a casi todo el mundo. La Raspberry Pi [2] es uno de los SoC más extendido. Gracias a su precio asequible y rápida distribución, que han facilitado su expansión, han llegado a un conjunto de nuevos desarrolladores, desde profesionales a usuarios amateurs, que han comenzado a crear numerosas aplicaciones capaces de interactuar con el mundo real. Este dispositivo Raspberry Pi [2], de tamaño reducido y consumo muy bajo, lo hacen ser el candidato perfecto para su uso en cantidad de proyectos, desde centros multimedia o estaciones meteorológicas [3] [4], hasta proyectos más complejos por parte de investigadores profesionales como, por ejemplo, el uso de conjuntos de Raspberry para su uso en computación en paralelo en la nube [5].

El objetivo de este trabajo es presentar el diseño y desarrollo de la creación de un sensor acústico basado en la plataforma Raspberry Pi para la monitorización del ambiente sonoro. En otros trabajos de los mismos autores [6] [7] se han usado plataformas de bajo coste para la medición de ruido. En este caso el cálculo de los parámetros se pretende realizar *on-board* junto con la transmisión en tiempo real a un servicio de almacenamiento y visualización de datos en la nube.

El concepto, recientemente en auge, de ciudad inteligente ha dirigido el interés y la necesidad a numerosas instituciones de recopilar más información sobre sus ciudades, tanto como para obtener información del aire [8], sistemas de aparcamiento inteligente [9] o para comprobar la integridad de las edificaciones [10]. A nivel acústico, la monitorización del ruido ha permitido profundizar en el conocimiento de las ciudades. Se ha estudiado el uso redes de sensores en ciudades en recientes artículos como [6], [11], [12], [13]. En [6] se describe el uso de Raspberry Pi como nodo de una red de sensores acústicos inalámbricos en los que se lleva a cabo captura de la señal de audio, la cual es enviada a un nodo servidor para su análisis de niveles de ruido. En [11] se estudian distintas clases de sensores acústicos y se discuten diversas estrategias de implementación en redes. En [12] se detalla el funcionamiento de la red de sensores acústicos instalada en la ciudad de Barcelona, comparando una red principal con sensores de clase I y una secundaria usando sensores de un coste menor y en [13] se presenta una red de sensores acústicos para monitorización del ruido con medidas sincronizadas en distintas posiciones en tiempo real para un parámetro (L_{eq}) y se analiza el rendimiento de la red. En el sensor presentado en este artículo se aporta la capacidad de la publicación de los resultados en tiempo real en la nube, calculados ejecutando los algoritmos de manera local *on-board*.

En los siguientes apartados se presenta la descripción del diseño del sensor y sus componentes, la metodología a seguir a la hora de preparar los algoritmos y la conexión a la nube para mostrar los resultados en tiempo real. Se finaliza aportando las conclusiones del trabajo realizado y de las futuras mejoras e implementaciones del sensor.

EL PROTOTIPO

Las redes de sensores acústicos permiten monitorizar en distintos puntos en el espacio así como también en distintos momentos de tiempo. En la figura 1 se muestra una fotografía del aspecto exterior del prototipo de sensor acústico creado, el cual está basado en la plataforma Raspberry Pi. Se observa en la parte inferior de la caja estanca el micrófono externo.



Figura 1. Diseño final del sensor acústico encapsulado en caja estanca para su instalación en exteriores

La plataforma Raspberry Pi como núcleo principal del sensor ha sido elegida debido a su alta versatilidad, facilidad de implementación de los algoritmos usando un lenguaje de programación conocido, en este caso, C adaptado de Matlab, así como por características como su reducido tamaño y su bajo precio, permitiendo así la idea de ser una plataforma de bajo coste. También tiene un consumo energético reducido, equilibrado con una potencia de cálculo suficiente (Tabla 1), adecuada para las tareas empleadas en este proyecto.

En la figura 2 se observa un diagrama de bloques del sistema usado por el sensor acústico donde se identifican los bloques del sistema formado por la Raspberry Pi, con módulo de adquisición de sonido y adecuación de la señal así como de extracción de parámetros y el canal de salida a la nube. También se observan las conexiones de datos y conexión eléctrica.

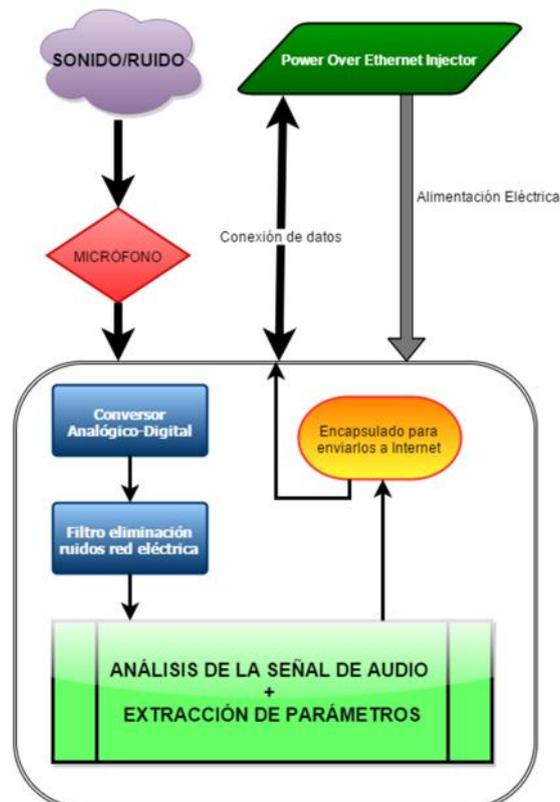


Figura 2. Diagrama de bloques del sistema

Características Raspberry Pi Model B+	
Precio	35 €
SoC (System on a chip)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM)
CPU	700 MHz single-core ARM1176JZF-S
GPU	Broadcom VideoCore IV 250 MHz
Memoria (SDRAM)	512 MB
Puertos USB 2.0	4 puertos
Salida de audio	1 salida analógica 3.5 mm minijack 1 salida digital via HDMI
Almacenamiento	Ranura MicroSD
Conectividad en red	10/100 Mbits/s Ethernet adaptador
Conectividad low-level	17 x GPIO (entradas/salidas de propósito general)
Consumo energético	600 mA (3.0 W)
Fuente de alimentación	5 V vía MicroUSB
Tamaño	85.60 mm x 56.5 mm
Peso	56 g

Tabla 1. Especificaciones de la Raspberry Pi B+ [DATASHEET RASPPi]

La comunicación, en este caso, con la Raspberry Pi se hace mediante una conexión Ethernet 10/100 Mbits/s, aunque también está la posibilidad de configurar una conexión inalámbrica mediante un adaptador USB WiFi (IEEE 802.11b [14]), en el caso de que no se pudiese operar en una red cableada. En el caso del sensor desarrollado, además, se ha aprovechado la capacidad de los cables Ethernet, no sólo como canal de transmisión para los datos, sino también como transmisores de corriente eléctrica [15]. De esta manera, se ha utilizado un inyector de corriente a través de Ethernet [16] en la entrada de red, y un *splitter* de corriente [17] insertado en la caja estanca. El *splitter* se encarga de separar la corriente eléctrica por un lado, para así alimentar a la Raspberry Pi, y el canal de datos por otro, para proporcionar conectividad a la placa (Figura 3). La tensión de salida es regulable en el modelo de *splitter* utilizado, y el voltaje de salida es configurado a 5 voltios. De esta manera, se evita la necesidad de incluir un transformador en la caja estanca que contiene a la Raspberry, el *splitter* y el cableado.

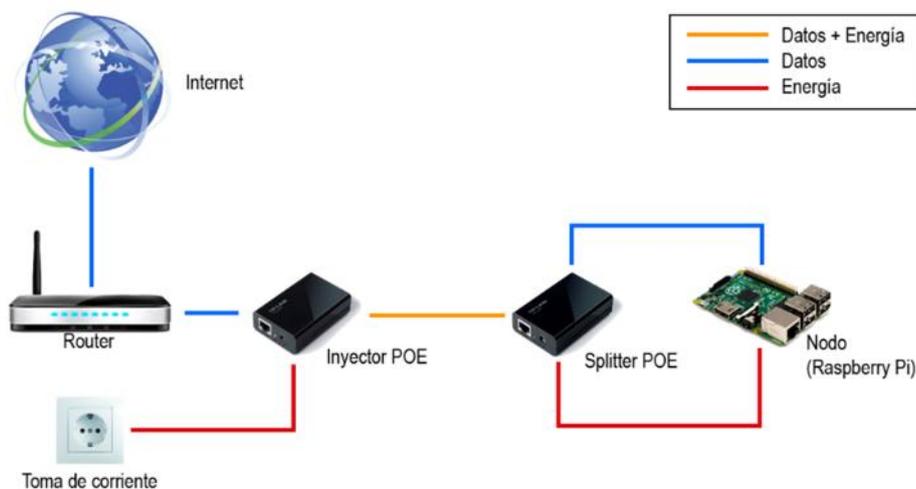


Figura 3. Esquema de funcionamiento de alimentación y conectividad a la red mediante POE



Figura 4. Interior de la caja estanca con la Raspberry Pi en primer plano, el cableado Ethernet y de alimentación, y el *splitter* debajo de la caja que contiene la Raspberry.

Otra de las grandes ventajas al trabajar con desarrollos en Raspberry Pi es la posibilidad de operar sobre un sistema operativo libre. En el caso de nuestro dispositivo, se ha utilizado una distribución *Raspbian* [18], una distribución del SO GNU/Linux para Raspberry Pi. Los algoritmos son desarrollados en Matlab y compilados en C, y éstos son ejecutados en *Raspbian*. Al estar la placa conectada en red, otra ventaja ligada al uso de este sistema operativo es la capacidad de controlar remotamente la placa a través de SSH (*Secure Shell*, en español: intérprete de órdenes segura) [19]. A través de un terminal de comandos se accede al sistema y se llevan a cabo tareas remotas como la actualización de los algoritmos o la comprobación del correcto funcionamiento del sistema, entre otras.

USO DE MICRÓFONOS DE ELECTRÓNICA DE CONSUMO PARA LA MONITORIZACIÓN AMBIENTAL

La producción masiva de micrófonos para la electrónica de consumo ha abierto el camino para usar estos micrófonos en aplicaciones asequibles de medida de ruido. Estas aplicaciones pueden ir desde monitorización del campo sonoro hasta la validación de mapas de ruido calculados. Los micrófonos son uno de los puntos críticos en los dispositivos de medición, ya que sus características afectan directamente al resultado final de las medidas.

En un reciente estudio [20], en el que se analizan diversos micrófonos de bajo coste, se observó una desviación en las mediciones de alrededor de 1 dBA a lo largo de 6 meses de mediciones continuas. En el mismo estudio se concluye también, que los micrófonos de bajo coste de electrónica de consumo pueden ser utilizados en tareas de monitorización del campo sonoro aunque con ciertos inconvenientes. A largo plazo, en las pruebas en exteriores, los micrófonos tendían a variar sus resultados frente a los de referencia. Probablemente, debido a causas del medio como la temperatura del aire o la humedad ambiente. Aunque se concluye que estos errores son nimios ante la diferencia de precio existente entre micrófonos profesionales para mediciones y estos micrófonos. El reemplazo de los micrófonos que, tras un cierto periodo de tiempo, hayan fallado, es una opción viable. Se concluye también que la intervención humana se hace necesaria para, por ejemplo, la calibración periódica, así que la sustitución de estos micrófonos de bajo coste podría formar parte de una rutina de mantenimiento de los sensores.

En este trabajo se ha comparado la utilización de dos opciones de captación:

- 1- Una tarjeta de sonido USB de bajo coste junto con un micrófono dinámico.
- 2- Un micrófono USB electret sin tarjeta de sonido.

Se comenzó con el primer caso, usando una tarjeta de sonido externa USB junto con a un micrófono dinámico. Desde el comienzo, con esta configuración, el sonido era capturado con mucho ruido, debido a la calidad limitada de la tarjeta de sonido USB. Se hicieron distintas pruebas implementando filtros digitales para eliminar ese ruido pero finalmente se optó por el segundo caso citado, con el uso de un micrófono USB [21], el cual incorpora el módulo conversor analógico-digital (ADC) integrado. La eliminación de un paso del sonido en la cadena de transmisión redujo el ruido eléctrico, siendo fácilmente tratable usando filtros digitales (Figura 5).

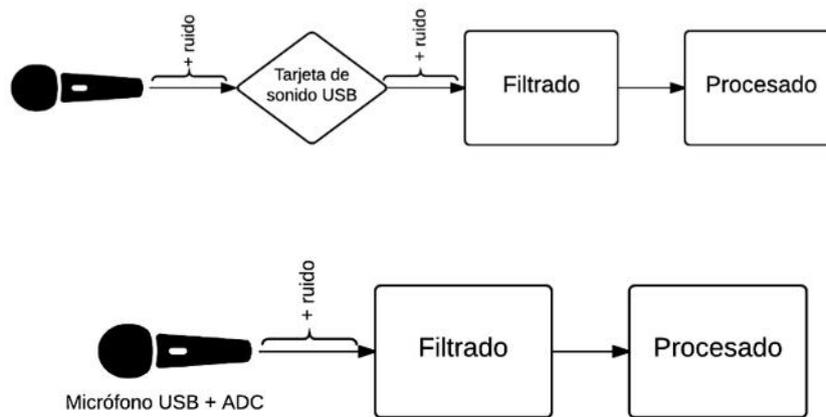


Figura 5. Configuraciones de captura del sonido

La tarea de calibración de los dispositivos también es una fase muy importante en el desarrollo de estos dispositivos. En el caso del sensor construido, se realizó una corrección espectral de la respuesta en frecuencia del micrófono después de una medición de la misma en laboratorio. Además, antes de la puesta en marcha se realizó una verificación y ajuste de nivel mediante un calibrador sonoro marca *Rion* [22].



Figura 6. Micrófono del sensor desarrollado

IMPLEMENTACIÓN DE LOS ALGORITMOS

Para la programación de los algoritmos se ha escogido el lenguaje de programación Matlab, usando Simulink. De esa manera, se han podido centrar los esfuerzos en el desarrollo de los algoritmos y la estructura del programa.

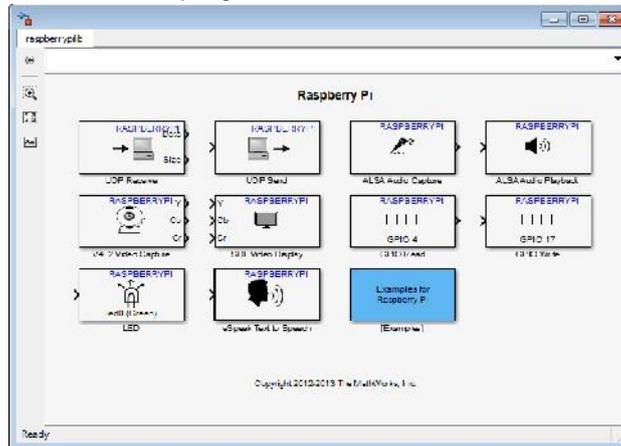


Figura 7. Ventana con bloques de programación de Simulink (MATLAB) para Raspberry Pi.

La captura del audio se hace a través de un bloque, *ALSA Audio Capture* (ALSA, *Advanced Linux Sound Architecture*) que usa un componente del núcleo Linux destinado a proveer al sistema de funcionalidades de sonido como la configuración automática de tarjetas de sonido y el manejo de los dispositivos en un solo sistema Linux [23]. Se puede configurar como características de la salida del sonido a capturar la frecuencia de muestreo y el tamaño de *frame* (número de muestras por ventana). Configurada correctamente la adquisición de audio, se realiza un acondicionamiento de la señal donde se hace un filtrado digital para la eliminación del ruido proveniente de la red eléctrica.

La señal de audio en escala lineal es convertida a escala logarítmica y corregida usando la calibración mencionada en el apartado relativo a la adquisición de audio. Usando bloques en Simulink en los que se pueden integrar funciones de MATLAB, se programan los algoritmos encargados de extraer los distintos parámetros acústicos de la señal de audio.

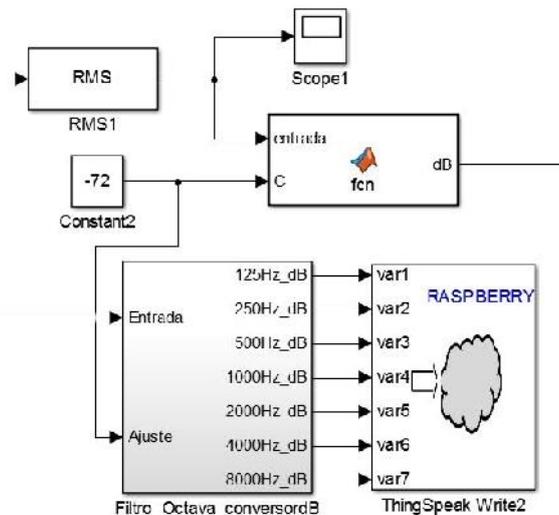


Figura 8. Bloque de función de MATLAB en Simulink.

En este caso se han elegido los parámetros:

- Nivel de presión sonora instantáneo.
- Percentiles L10, L50 y L90.
- Nivel de presión sonora equivalente.
- Nivel de presión sonora en tercios de octava (desde 125 Hz hasta 8000 Hz).
- Parámetros psicoacústicos *loudness* y *sharpness* [24].

Los esfuerzos en esta primera versión del sensor han sido enfocados en la optimización de los cálculos de los parámetros antes mencionados y la correcta presentación y publicación de los datos obtenidos a través de Internet en tiempo real. Para el envío de resultados en tiempo real a la nube, se usa otro bloque de Simulink que agiliza la tarea de realizar la conexión con el gestor de la base de datos. El funcionamiento más detallado de esta conexión se presenta en el siguiente apartado.

ACCESIBILIDAD Y GESTIÓN DE LOS DATOS

La potencia real de este tipo de sistemas analizadores de audio reside en la capacidad de llevar a cabo los cálculos en distintas posiciones en distintos instantes de tiempo, y también en la capacidad para estar interconectados unos con otros [25]. Esto dota a los nodos de conectividad, permitiendo su monitorización y gestión remota [26]. Un paso más allá es la conexión directamente con la nube y los servicios que en ella se pueden gestionar.

Al sensor desarrollado se le ha dotado de conectividad con Internet mediante la conexión Ethernet y se ha aprovechado esta conexión a la nube para almacenar y mostrar los resultados de los parámetros acústicos calculados sobre una plataforma web para el Internet de las Cosas llamada *ThingSpeak*. En Matlab, a través de Simulink, indicando la API Key de escritura, el número de variables a enviar y la dirección de actualización se configura la interconexión del sensor Raspberry con *ThingSpeak* (Figura 9).

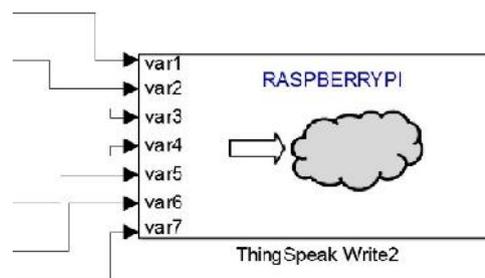


Figura 9. Bloque Simulink de conectividad con ThingSpeak

Cada canal incluye hasta 8 campos que pueden contener cualquier tipo de datos. Una vez que los datos son recogidos, éstos pueden ser visualizados de manera privada o pública, a través de aplicaciones de la plataforma *ThingSpeak* (Figura 10).

Usando las posibilidades de visualización de *ThingSpeak* se han creado diversas gráficas para visualizar en tiempo real los datos que el sensor está calculando y enviando a la nube. También se pueden extraer los datos de un canal en diversos formatos (JSON, XML o CSV) para su análisis *offline* o *backups*.

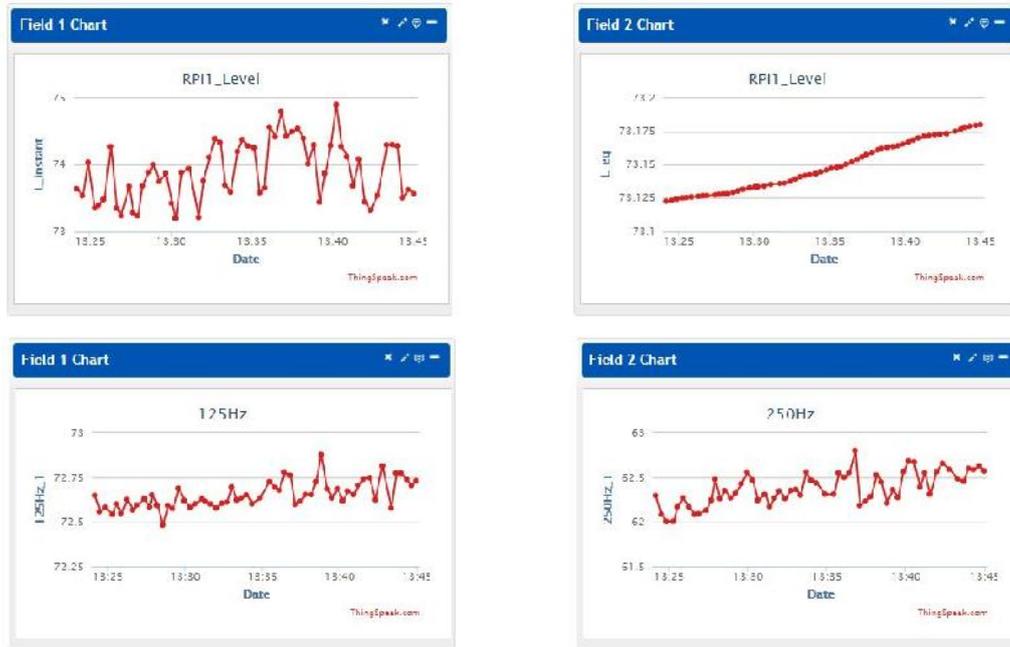


Figura 10. Ejemplos de visualización online de variables en la plataforma ThingSpeak

Los datos de uno de los sensores instalados están disponibles de manera pública y pueden ser consultados online [27, 28].

CONCLUSIONES

En este trabajo se ha descrito el diseño de un sensor, sus componentes y la metodología a seguir a la hora de preparar los algoritmos, así como la conexión a la nube para mostrar los resultados en tiempo real. En otros trabajos de los mismos autores se han propuesto las plataformas de bajo coste para la adquisición de sonido ambiental, pero en este artículo se ha presentado un sensor funcional con las características añadidas de cálculos *on-board* y muestra los resultados en tiempo real en la nube. Se concluye que la plataforma Raspberry Pi es viable para llevar a cabo cálculos *on-board* de parámetros más complejos sin perder funcionalidades de conectividad en tiempo real.

La plataforma y los métodos elegidos para su programación ofrecen varias ventajas como una gran versatilidad, facilidad de implementación de algoritmos, un bajo precio y una alta facilidad para integrarla en sistemas para instalación en exteriores, teniendo así una gran potencia como nodo de una red de sensores.

En trabajos futuros se pueden realizar nuevos algoritmos de cálculo de parámetros avanzados como para los parámetros psicoacústicos *roughness* y *fluctuation strength* junto con la optimización de *loudness* y *sharpness*, obteniendo los resultados en tiempo real calculándolos en Raspberry y no en cálculos posteriores, ya que se ha concluido que la plataforma tiene potencia de cálculo suficiente. Por otro lado, una mejora en la visualización de los parámetros publicados. También se harán pruebas usando micrófonos MEMS para la adquisición de sonido.

AGRADECIMIENTOS

Este trabajo está conjuntamente apoyado por los fondos del Ministerio de Educación, Ciencia y Deporte MECD español y FEDER europeo bajo el proyecto ref. TEC2012-37945-C02-01 y la UCAM Universidad Católica San Antonio bajo el proyecto ref. PMAFI-02-14.

REFERENCIAS

- [1] Benini, L., & De Micheli, G. (2002). Networks on chips: a new SoC paradigm. *Computer*, 35(1), 70-78.
- [2] Raspberrypi.org. (2015). *Raspberry Pi Hardware - Raspberry Pi Documentación online*. Disponible: <https://www.raspberrypi.org/documentation/hardware/raspberrypi> [Consultado 2 Sep. 2015]
- [3] Richardson, M., & Wallace, S. (2012). *Getting started with raspberry Pi.* O'Reilly Media, Inc."
- [4] Upton, E., & Halfacree, G. (2014). *Raspberry Pi user guide*. John Wiley & Sons.
- [5] Tso, F. P., White, D. R., Jouet, S., Singer, J., & Pezaros, D. P. (2013). The Glasgow raspberry Pi cloud: A scale model for cloud computing infrastructures. *Proceedings - International Conference on Distributed Computing Systems*, 108–112. doi:10.1109/ICDCSW.2013.25
- [6] Segura-Garcia, J., Felici-Castell, S., Perez-Solano, J. J., Cobos, M., & Navarro, J. M. (2015). Low-Cost Alternatives for Urban Noise Nuisance Monitoring Using Wireless Sensor Networks. *Sensors Journal, IEEE*, 15(2), 836-844.
- [7] Noriega, J. E., Navarro, J. M., Felici, S., & Segura, J. (2014). Monitorización del Aislamiento a Ruido Aéreo Mediante Redes de Sensores Inalámbricas. *Tecniacústica 2014*.
- [8] Dekoninck, L., & Botteldooren, D. (2015). Sound sensor network based assessment of traffic noise, and air pollution. *Euronoise 2015*, 2321–2326.
- [9] Chinrungrueng, J., Sunantachaikul, U., & Triamlumlerd, S. (2007, January). Smart parking: An application of optical wireless sensor network. In *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on* (pp. 66-66). IEEE.
- [10] Kim, S., Pakzad, S., Culler, D., Demmel, J., Fennes, G., Glaser, S., & Turon, M. (2007, April). Health monitoring of civil infrastructures using wireless sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on* (pp. 254-263). IEEE.
- [11] Manvell, D. (2015). Utilising the Strengths of Different Sound Sensor Networks in Smart CityNoise Management. *Euronoise 2015*, 2303–2308.
- [12] Farrés, J. C. (2015). Barcelona noise monitoring network. *Euronoise 2015*, 218–220.
- [13] Hakala, I. (2015). Area-based environmental noise measurements with a wireless sensor network. *Euronoise 2015*, 2351–2356.
- [14] IEEE Standards Association & others (2001). *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE
- [15] Mendelson, G. (2004). All You Need To Know About Power over Ethernet (PoE) and the IEEE 802.3 af Standard. *Internet Citation, [Online] Jun*.
- [16] Tp-link.com. (2015). *Inyector PoE TL-POE150S*. Retrieved 7 September 2015, from <http://www.tp-link.com/ar/products/details/?model=TL-POE150S#over>
- [17] Tp-link.com. (2015). *Splitter PoE TL-POE10R*. Retrieved 7 September 2015, from <http://www.tp-link.com/ar/products/details/?categoryid=234&model=TL-POE10R#over>
- [18] Raspbian.org.. (2015). *FrontPage - Raspbian - Raspbian*. Retrieved 8 September 2015, from <https://www.raspbian.org/>
- [19] Es.wikipedia.org.. (2015). *Secure Shell*. Retrieved 8 September 2015, from https://es.wikipedia.org/wiki/Secure_Shell
- [20] Van Renterghem, T., Thomas, P., Dominguez, F., Dauwe, S., Touhafi, A., Dhoedt, B., & Botteldooren, D. (2011). On the ability of consumer electronics microphones for environmental noise monitoring. *Journal of Environmental Monitoring*, 13(3), 544-552.
- [21] Tbone-mics.com.. (2015). *Productdetails - t-bone*. Retrieved 6 September 2015, from <http://www.tbone-mics.com/en/product/information/details/-d9d6e87219/>
- [22] NC-74 Datasheet - <http://www.noise-and-vibration.co.uk/files/pdfs/NC-74.pdf>
- [23] Alsa-project.org.. (2015). *AlsaProject*. Retrieved 3 September 2015, from http://www.alsa-project.org/main/index.php/Main_Page
- [24] Fastl, H., & Zwicker, E. (2007). *Psychoacoustics: Facts and models* (Vol. 22). Springer Science & Business Media.
- [25] Mainetti, L., Patrono, L., & Vilei, A. (2011, September). Evolution of wireless sensor networks towards the internet of things: A survey. In *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on* (pp. 1-6). IEEE.
- [26] Luo, J., Chen, Y., Tang, K., & Luo, J. (2009, December). Remote monitoring information system and its applications based on the Internet of Things. In *BioMedical Information Engineering, 2009. FBIE 2009. International Conference on Future* (pp. 482-485). IEEE.
- [27] Thingspeak.com.. (2015). *RPi1_Octavas - ThingSpeak*. Retrieved 8 September 2015, from <https://thingspeak.com/channels/38907>
- [28] Thingspeak.com.. (2015). *RPI1_Level - ThingSpeak*. Retrieved 8 September 2015, from <https://thingspeak.com/channels/38912>