

## Article

# On the Application of the Raspberry Pi as an Advanced Acoustic Sensor Network for Noise Monitoring

Juan Emilio Noriega-Linares \* and Juan Miguel Navarro Ruiz

Department of Technical Sciences, UCAM Catholic University of Murcia, 30107 Murcia, Spain; jmnavarro@ucam.edu

\* Correspondence: jenoriega@ucam.edu; Tel.: +34-968-278-825

Academic Editors: Simon J. Cox and Steven J. Johnston

Received: 12 July 2016; Accepted: 20 October 2016; Published: 27 October 2016

**Abstract:** The concept of Smart Cities and the monitoring of environmental parameters is an area of research that has attracted scientific attention during the last decade. These environmental parameters are well-known as important factors in their affection towards people. Massive monitoring of this kind of parameters in cities is an expensive and complex task. Recent technologies of low-cost computing and low-power devices have opened researchers to a wide and more accessible research field, developing monitoring devices for deploying Wireless Sensor Networks. Gathering information from them, improved urban plans could be carried out and the information could help citizens. In this work, the prototyping of a low-cost acoustic sensor based on the Raspberry Pi platform for its use in the analysis of the sound field is described. The device is also connected to the cloud to share results in real time. The computation resources of the Raspberry Pi allow treating high quality audio for calculating acoustic parameters. A pilot test was carried out with the installation of two acoustic devices in the refurbishment works of a neighbourhood. In this deployment, the evaluation of these devices through long-term measurements was carried out, obtaining several acoustic parameters in real time for its broadcasting and study. This test has shown the Raspberry Pi as a powerful and affordable computing core of a low-cost device, but also the pilot test has served as a query tool for the inhabitants of the neighbourhood to be more aware about the noise in their own place of residence.

**Keywords:** noise monitoring; Internet-of-Things; acoustics; noise monitoring; Raspberry Pi; noise awareness

## 1. Introduction

In the last few years, world population has grown significantly and most of this increase has happened in urban areas and cities, according to data from the World Health Organization. This has brought new needs to cities and to urban planners. The new requirement for controlling citizens' welfare involves monitoring cities, including variables like CO<sub>2</sub> levels [1], water quality [2] or noise levels [3], amongst others. The psychological comfort of people is one of the most important factors of welfare in the cities, and noise pollution holds a key place in this ranking. Noise pollution is the excessive noise levels or annoying sounds that unsettle people, and also animals, in their place of residence, leisure areas or their workplaces. This kind of pollution has different effects on people's health, physically and psychologically [4]. Main sources of noise pollution are high volumes of traffic, building areas and human-based sounds, for example nightlife sounds [5,6]. All these sources, together with all the different sounds present in the cities, merge to create a unique combination of sounds creating the soundscape of the city [7,8]. Monitoring these sources of noise pollution is important for the understanding of how these sounds evolve with time, in order to study it, control it and prevent it.

The concept of Smart City has emerged in the last decade, creating the necessity of gathering more information about their cities. Population of cities has been growing and new urban planning strategies should be developed in order to manage this situation [9]. The collected information is widely used, from controlling free parking spaces in a certain neighbourhood [10] or checking structural integrity of buildings [11]. In the acoustics research field, noise monitoring has helped to delve into the knowledge that people and institutions have of their cities. The use of wireless sensor networks with acoustic sensor has been studied in several works [12–15].

Recently, there has emerged a new concept of computers. These new devices are also known as Single Board Computers (SBC) [16,17], being smaller than classic computers and with the distinguishing feature of being more economical and affordable. This new kind of small computers has demonstrated its computing power together with its scalability for big projects [18]. There are different SBCs in the market with different features of connectivity, computing power, size or energy usage. Raspberry Pi, BeagleBone, Arduino, ODroid, are widely used in this development field [12,19].

The role of acoustic researching in the smart city has a series of applications and benefits such as having more control of noise levels by permanent and real-time control, detecting new noise sources or using these tools as a showcase for informing the citizens among others. From the point of view of public administration, it helps to anticipate citizens' complaints and to complement and update the information provided by noise maps, in the design process of action plans. The investment and maintenance costs in this kind of devices is more economical than carrying out strategic noise maps repeatedly. These results end up in fewer administrative works and interventions on the part of the public administration.

Noise levels differ between day and night periods generally. In residential areas, the limits accepted are those which not exceed 65 dBA during the day, and 55 dBA for the night period [20]. Accepted standards for recommended permissible exposure time for continuous time weighted average noise, stated that for every 3 dBA over 85 dBA, the permissible exposure time before possible damage can occur is cut in half, e.g., 85 dBA is linked with a permissible exposure time of 8 h; 88 dBA for 4 h, 91 dBA for 2 h [21]. The use of percentile levels in the acoustic analysis helps to have an understanding of the noise fluctuations over time. These are commonly used for environmental noise monitoring, such as road traffic or community noise assessments. With the use of long-term measurements, changes on the levels can be observed from the data, and more advanced studies can be performed in order to assess the noise annoyance. The levels extracted every day can be used for long-term analysis, with large amounts of data from days, weeks or months of measurements.

The work presented in this paper describes the creation of an advanced acoustic sensor based on a Raspberry Pi. It aims to fill the gap in the noise sensors research and developing field, with the analysis and monitoring of the whole audio signal in the audible bandwidth. It also carries out the calculation of environmental noise parameters performed on-board instead of in a server. The device also share the results in an Internet-of-Things (IoT) publishing online platform. The result of the research and development carried out in this paper is a reliable prototype, built using low-cost components, of an advanced acoustic sensor for environmental noise monitoring.

This paper is divided in the following sections. First, the materials and methods for the design of the prototype are disclosed. The choices of the materials for creating the prototype, based on some requirements, are explained. Then, the prototype assembly is shown. The algorithms implementation and the cloud connection are shown. After these sections, a test pilot device is deployed and outlined in the results. Finally, the conclusions and the discussion are shown in the last section.

## 2. Materials and Methods

This section contains the different subsections where the design and creation of the prototype together with the algorithms implementation and its cloud connection are explained.

### 2.1. Design and Requirements

The design of the device had to accomplish some requirements for achieving the final goals proposed. Some statements have to be achieved in order to follow the low-cost, but reliable, final device. The requirements for this prototype were:

- The device has to use low-cost components to create affordable sensor networks of several devices with a relation cost-quality.
- The device has to have reliability for long-term measurements.
- The device should have capability to be connected to the cloud for remote updates of the software and for sharing results.
- The quality of the measurements has to be enough for advanced audio parameters' calculation.
- The device has to have enough computing power to do on-board calculations.
- The device has to be able to connect to the peripherals needed for the purposes of the project (e.g., a microphone).
- The device has to be able to interpret MATLAB programming language.
- The sound flow acquisition has to have the less noise inputs as possible, for avoiding extra filtering steps.
- The final device has to be protected against outdoor conditions using a protective housing.
- The device needs to have different connectivity options (i.e., WiFi or Ethernet).
- The distance from the nodes to the power source should be a maximum of 100 m.

These requirements secure an affordable and suitable design for the creation of a working prototype.

### 2.2. Selection of the Components

The main component of the device is the processing unit, which is also used for the data acquisition and the connectivity. For achieving the established requirements, the design of the noise monitoring device was based in a Raspberry Pi 2 Model B single board computer [22]. The Raspberry Pi platform offers a number of advantages as its good computing power, its high versatility and the existence of libraries of MATLAB functions. The power consumption and the price allow the construction of numerous devices based on this platform, resulting in affordable and durable nodes. Those qualities, together with the upgrades that the platform undergoes in its hardware over time in the form of new models, made the Raspberry Pi the selected option for the development of a working prototype.

The use of this kind of platform leads some limitations. In long-term measurements in outdoor conditions, the working temperature of the board has to be monitored in order to safeguard the integrity of the board and avoid its fault. In the proposed system, the addition of backup batteries or a UPS (Uninterruptible Power Supply) system allows protection for power cuts. For an increase in the computing resources of the system, a new version of the Raspberry Pi would be needed.

One important requirement is the ability of having less stages in the audio acquisition process, decreasing the number of noise inputs while capturing the audio signal. The second main part of the device is the sound acquisition hardware, i.e., the microphone and the sound card. In this project, and thanks to the ports of the Raspberry Pi, a USB microphone which integrates an audio capture card itself, a T-Bone GC 100 USB [23], was chosen. The omnidirectional directivity pattern and its frequency response, together with the USB connectivity makes the GC100 the perfect candidate for the prototype, reducing the stages of the acquisition process.

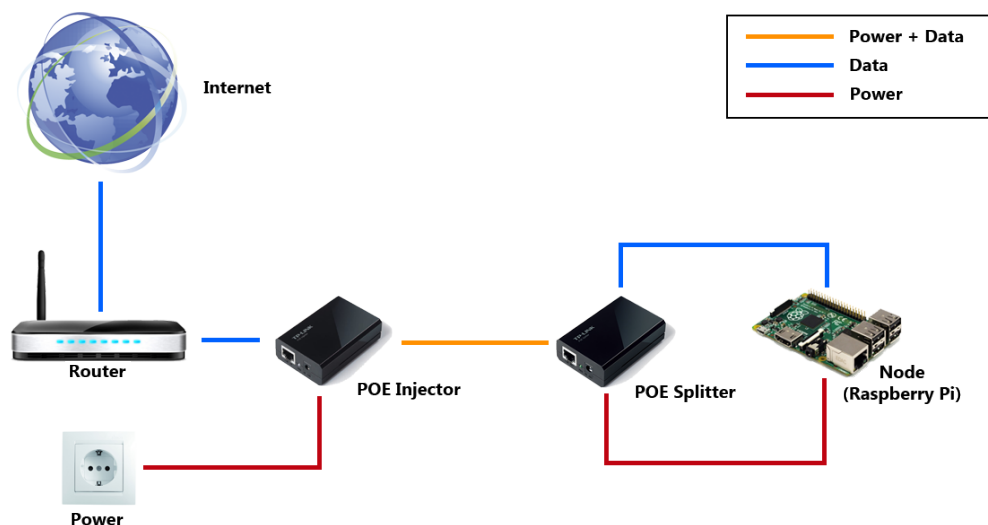
#### 2.2.1. The Core: Raspberry Pi

A benefit of using the Raspberry Pi is the possibility of working under a free operating system. In the case of this device, a Raspbian distribution [24], a GNU/Linux OS distribution for Raspberry Pi, has been used. The algorithms were developed in MATLAB and compiled in C. The Internet connection of the device also provides the possibility of working remotely, thanks to the SSH (Secure SHell) [25]. Through a command terminal it is possible to access the device and update the algorithms, software

maintenance tasks or checking the system can be achieved remotely. ALSA library [26] controls the audio configurations, managing the audio in an optimised way.

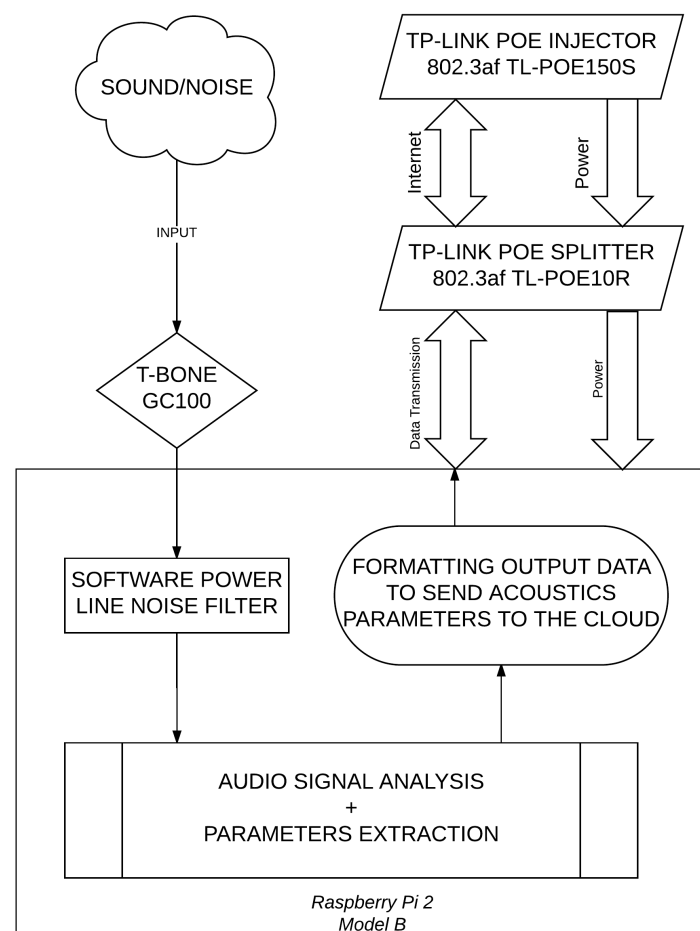
Based on the bandwidth requirements for continuous information transmission, the board allows different options, principally Ethernet, Wi-Fi, ZigBee or a 3G connection. In the case of this device and its deployment, the option of the Ethernet connection was chosen. For farther locations, where Ethernet connection is not available, the nodes would be equipped with wireless communication systems and the powering could come from different sources, such as batteries, solar panels or connected to an electricity supply, e.g., a lamppost. The use of a LAN connection in the deployment instead of a wireless system seeks for two main objectives: first, while a wireless connection is subject to more interference than a wired connection, Ethernet cables can be properly shielded, avoiding these unwanted effects. Although Ethernet cables can also experience signal degradation, this problem is easier to manage and avoid, taking into consideration the maximum distance, of 100 m, for proper communications and the categories and qualities of the LAN cables. Secondly, seizing the opportunity of using a cable network, a POE (Power Over Ethernet) scheme, based on IEEE 802.3af, for powering the devices was chosen [27].

In this way, a POE injector [28] has been placed in the network input, and a POE splitter in the output of the circuit, inside the box where the components are placed [29]. With the use of a category 5, 5e or 6 cable, a maximum distance of 100 m from the injector to the splitter can be used. In this work, the maximum length used is less than 10 m and the category of the cable is 5e. The splitter is in charge of the division between the current supply, for powering the device, and the data channel, for providing the Raspberry Pi with Internet connection, as seen in Figure 1. The output voltage is adjustable in the splitter model used, and it is adjusted to 5 volts, in order to correctly supply the Raspberry Pi and its components. In this way, there is no need for using an additional voltage transformer in the device.



**Figure 1.** Operation scheme for the power and Internet connectivity via Power over Ethernet (POE), IEEE 802.3af.

In Figure 2, a complete block diagram of the whole acoustic device is presented, where all the functioning blocks of the system are shown. One of the inputs of the system is the acquisition of the sound by the microphone capturing the ambient sound and converting the analogue sound signal to a digital form with the sound card integrated in the microphone. Then it goes to the Raspberry Pi processing stage for the parameters extraction. Next, the data are formatted to send the acoustic parameters to the cloud.



**Figure 2.** Block diagram of the complete system. The inputs of the system are the sound acquisition that goes through the T-Bone GC100 and the power and the Internet connectivity that are connected to the device through the TL-POE 150S Injector. In the Raspberry Pi 2 Model B, a digital filtering stage is in charge of removing any power line noises. The signal analysis and the parameters' extraction are performed and their results are processed to send them to the cloud service.

The components used in the final version of device are listed in Table 1.

**Table 1.** List of the components used in the final version.

Part	Comercial Name	Price
Main board	Raspberry Pi 2 Model B	35 \$
Microphone	T-Bone GC 100 USB	16 \$
POE Splitter	TP-Link TL-POE10R Splitter Power Over Ethernet	17 \$
POE Injector	TP-LINK TL-POE150S Injector Power Over Ethernet	26 \$
Enclosure	150 mm × 200 mm × 85 mm IP67 Enclosure	15 \$
Cables and consumables	various	12 \$
Total price per node		121 \$

## 2.2.2. The Acquisition: The Microphone

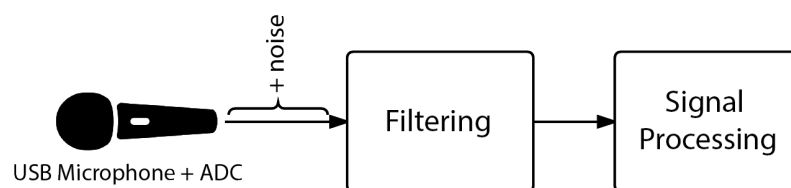
The massive production of consumer electronics microphones has paved the way for using these microphones in affordable applications for noise monitoring. These applications range from ambient

noise monitoring to noise maps validations. The microphones are a critical stage in monitoring and measuring devices, because its features affect to the final result of the measurements.

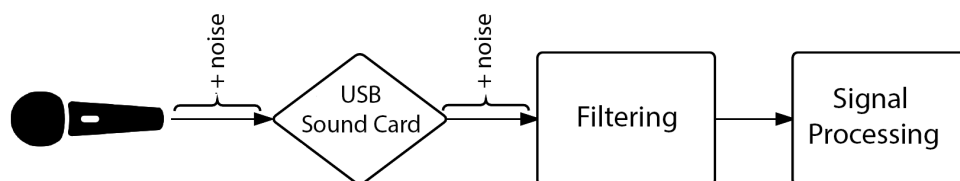
In [30], different affordable microphones are analysed. In the study, a deviation of around 1 dBA was found in a 6-month continuous test. In long-term measurements, in outdoor tests, these affordable microphones tended to deviate, in a comparison with the reference microphones. This deviation is produced by environmental agents like humidity and temperature. The possibility of replacing the deteriorated microphones, when the deviations with a control microphone were higher than 1 dBA, can be easily attempted with a low economic impact. Human intervention is necessary for certain tasks like periodic calibration, therefore, the substitution of deteriorated microphones could be integrated in a maintenance routine of the devices.

In this work, two schemes for acquiring audio were compared (Figure 3):

1. A low-cost USB sound card together with a dynamic microphone.
2. An electrec USB microphone without an external sound card.



(a) USB Microphone with integrated soundcard



(b) Dynamic microphone connect by a USB external soundcard

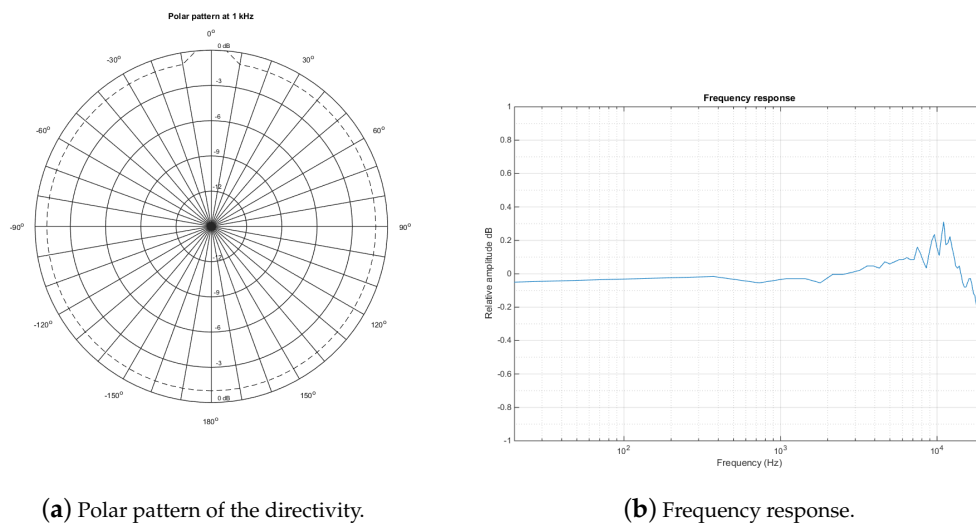
**Figure 3.** Different sound acquisition settings. (a) An electrec USB microphone without an external sound card; (b) Low-cost USB sound card with a dynamic microphone.

The dynamic microphone tested with the sound card was based on a multi-purpose Panasonic capsule [31], WM-61A, which has an omnidirectional pattern, a signal-to-noise ratio of 62 dB and a frequency response from 20 Hz to 16,000 Hz. With the use of the tested USB sound card, some noise was detected in the acquired sound. This noise was probably originated from interferences from the power supply. The USB sound card was not shielded and hence it is exposed to interferences. The selection was finally to use a USB microphone [23] which had integrated the Analog-to-Digital Converter (ADC).

The noise measurements with the USB microphone with the pre-amplifier and the ADC embedded on it offered better results, hence, it was the option selected. A high-pass filter from 100 Hz was applied to remove a peak of noise in the 50 Hz frequency. Because the sound pressure levels are analysed from the band of 125 Hz, this was the frequency chosen for the filter.

The acoustic device is calibrated, as shown in Figure 4, prior to in-field measurement, to capture audio and to calculate noise level with precision. First, using the frequency response of the microphone a correction filter is implemented to weigh its losses. Then, a verification and level adjustment was made comparing with a sound level calibrator Rion [32].





**Figure 4.** Characteristics of the selected microphone, T-Bone GC100.

For the assembly of all the components chosen for the final device, an isolated box was selected, and the different parts were all connected and placed inside the protective enclosure, as it can be seen in Figure 5a. The box is sealed and a metal anchor is placed in the back of it for its outdoor placement. The final aspect of the installation is shown in Figure 5b.



**Figure 5.** Views of the physical prototype built. (a) Final design of the acoustic device. In the picture, the Raspberry Pi and the other components are inside a protected box. At the bottom part of the box, the microphone can be observed; (b) Interior view of the waterproof case. The components inside of it are the Raspberry Pi together with the POE splitter, the microphone at the bottom of the picture and all the cables for the connection.

### 2.3. Algorithms Implementation and Cloud Connection

The Raspberry Pi platform was chosen as the core of the device because of its high versatility and its ease implementing the algorithms using a known and wide spread programming language, C, but implemented from MATLAB language through Simulink. The audio acquisition was made through an ALSA Audio Capture block which uses a component from the Linux kernel meant to provide the

system with audio functionalities, e.g., automatic configuration for sound cards and the controlling of devices using one Linux system [26]. It is possible to configure features for the audio acquire such as the sampling rate and the frame size, which is the number of samples per window. Once the audio acquisition is configured, a previous filtering is performed in order to remove electrical noise from the power line.

The audio signal is acquired in a linear way, i.e., instant pressure values, so the next step performed is the conversion to a logarithmic scale. It is also adapted by a spectral correction factor, got from an empirical fitting from the calibration task. Through Simulink blocks where MATLAB code is executed, the different acoustic parameters are calculated:

- Instant sound pressure level,  $L_p$ .
- Percentile levels  $L_{10}$ ,  $L_{50}$ ,  $L_{90}$ .
- Equivalent sound pressure level,  $L_{eq}$ .
- $L_{den}$ , day-evening-night level.
- Third octave sound pressure level (from 125 Hz to 8000 Hz)

The description of the implemented equations in the measurement algorithm for different acoustics parameters are shown in the following:

$L_p$ , sound pressure level is a logarithmic measure of the RMS sound pressure of a sound relative to a reference value, that is the threshold of hearing.

$$L_p = 20 \cdot \log_{10} \left( \frac{P}{P_0} \right) (dB), \quad (1)$$

where  $P$  is the instantaneous sound pressure of the sound signal and  $P_0$  is the reference sound pressure of 20  $\mu$ Pa.

$L_{eq}$ , equivalent sound pressure level. It quantifies the noise environment to a single value of sound level for a determined duration. This parameter correlates with the effects of noise on people.  $L_{eq}$  can be calculated as:

$$L_{eq} = 10 \cdot \log_{10} \left[ \frac{1}{T_M} \cdot \int_0^{T_M} \left( \frac{P(t)}{P_0} \right)^2 \cdot dt \right], \quad (2)$$

where  $L_{eq}$  is the equivalent continuous sound pressure level determined over a time interval of  $T_M$  seconds. For the addition of the  $L_{eq}$  levels, in order to calculate other parameters, it can be performed as shown:

$$TotalL_{eq} = 10 \cdot \log_{10} \cdot \left( \frac{10^{\frac{L_{eq,1}}{10}} + 10^{\frac{L_{eq,2}}{10}} + \dots + 10^{\frac{L_{eq,n}}{10}}}{n} \right). \quad (3)$$

The parameter  $L_{den}$ , day-evening-night level, is the  $L_{eq}$  measured over a 24 h period with a 10 dB penalty added to the levels between 23:00 and 07:00 and a 5 dB penalty added to the levels between 19:00 and 23:00. This is applied to reflect people's extra sensitivity to noise during these periods. As the  $L_{eq}$  correlates with the effects of noise on people,  $L_{den}$  extrapolates this to a daily value and, in long-term measurements, in weekly, monthly or yearly data for more advanced studies.

$$L_{den} = 10 \cdot \log \left( \frac{12 \cdot 10^{\frac{L_{day}}{10}} + 4 \cdot 10^{\frac{L_{evening}+5}{10}} + 8 \cdot 10^{\frac{L_{night}+10}{10}}}{24} \right), \quad (4)$$

where  $L_{day}$  is the level for the day period, between 07:00 and 19:00 h;  $L_{evening}$  is the level for the evening period, between 19:00 and 23:00 and  $L_{night}$  is the level for the night period, between 23:00 and 07:00.

$L_{90}$  describes the level which was exceeded for 90% of the time;  $L_{10}$  the level exceeded for 10% of the time and  $L_{50}$  is an indicator for the median sound level. These parameters describe the behaviour of the noise in long-term measurements and help in the study of the intervals statistics. The  $L_{90}$  level is often used as approximation of the background noise level. Likewise,  $L_{10}$  is an approximation for the



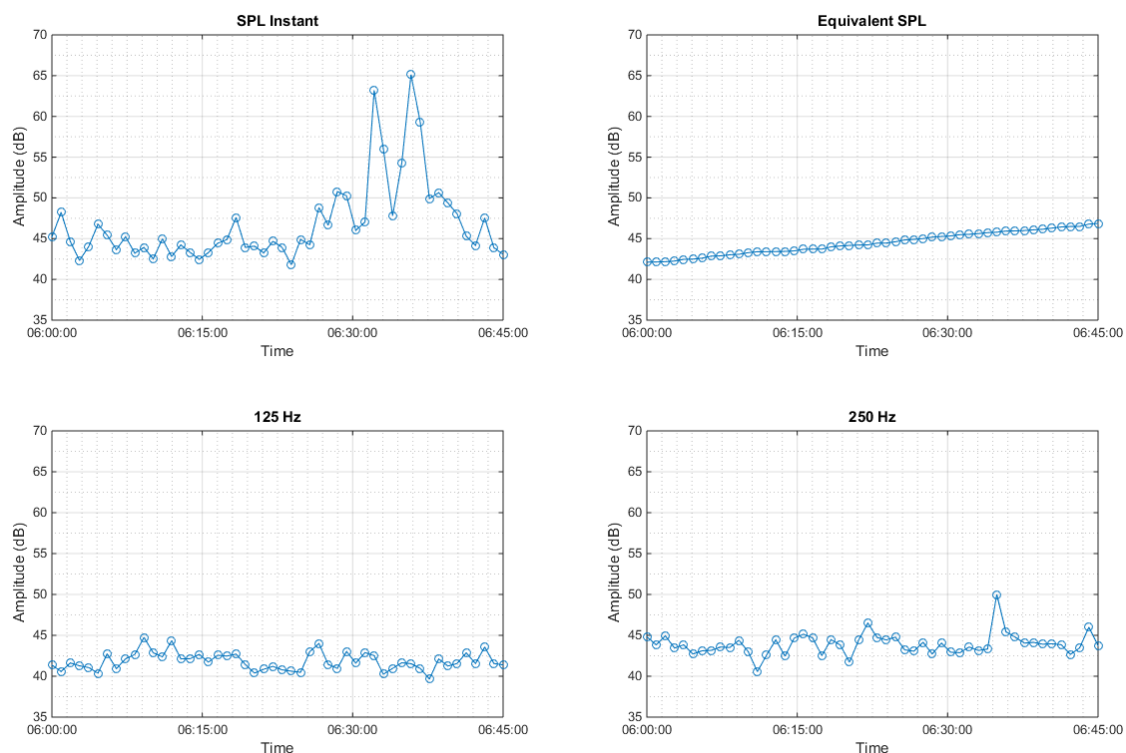
peak levels. A practical example would be a measurement time of 10 min, with  $L_{90} = 80$  dBA and  $L_{10} = 90$  dBA. This means that for 9 min, the level has been higher than 80 dBA and for a minute, levels above 90 dBA. With these data, it could be concluded that the acoustic environment analysed would be too loud, with high sound pressure levels and health risks for long-term exposition.

For the frequency analysis, an inner filtering stage is performed where the spectrum of the input signal is analysed. The processor passes the collected audio pieces through a set of a third-octave-band filters and splits the spectrum of the sound for further sound pressure level per band calculations. This filter analyses in third-octave bands from 125 Hz to 8000 Hz.

The main objectives to achieve on this version of the device have been the optimisation of the parameters calculations and the presentation and publishing of the data in real time. The potential of this audio analysis network lies in the capability of carrying out the calculation in several nodes, using one device for each position, and in different periods of time [33]. Mover, making use of the connectivity of the network, it is possible to monitor and remotely manage the nodes [34].

For this first prototype, it has been equipped with Internet connectivity through an Ethernet connection and this capability has been exploited to store and show the results of the extracted acoustics parameters. This task has been carried out using an online platform called ThingSpeak. It is an open source IoT application and Application Programming Interface to store and retrieve data from the devices using the Hypertext Transfer Protocol over the Internet [35]. Moreover, the platform enables the creation of sensor logging applications with status updates. In MATLAB, the connection with the device and ThingSpeak is performed using the API key, the number or variables to send and the updating address.

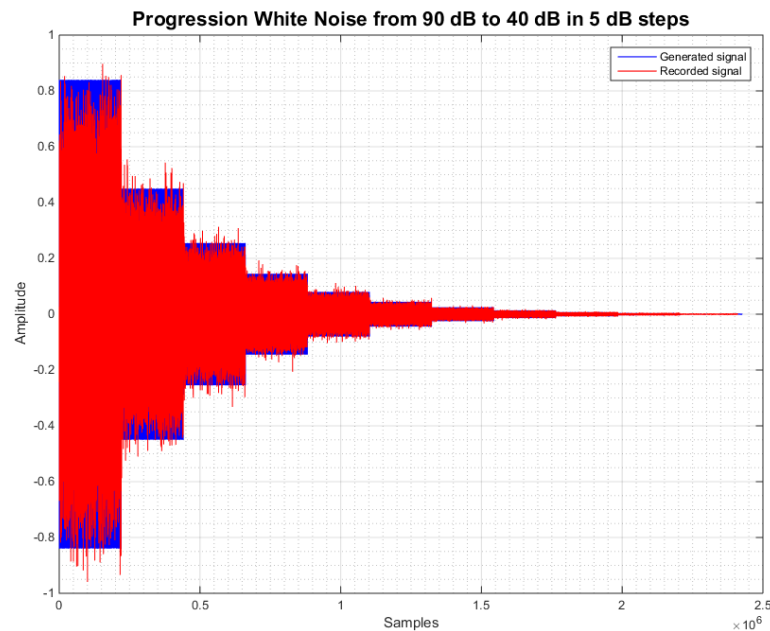
In Thingspeak, graphs have been created showing real time data extracted directly from the sensor, which is calculated and sent to the cloud, as shown in Figure 6 and can be consulted in [36,37]. Once the data is gathered, the channels can be set up as public or private. Data can be extracted in different formats, e.g., JSON, XML or CSV, for offline tests, backups or analysis of the data.



**Figure 6.** Views of the instant sound pressure level broadband and for 125 Hz and 250 Hz third octave bands and for the equivalent sound pressure level.

### 3. Results

The laboratory test for checking the accuracy of the system was carried out with an integrating sound level meter, Rion NL-05 with a flat-frequency-response UC-52 microphone, using a dodecahedron speaker and an audio amplifier connected to a software-based signal generator. These tests were performed in a semi-anechoic chamber. With the signal generator and the speaker, a white-noise signal was emitted and with the use of the sound level meter, it was adjusted to different gains and decibels. The readings from the sound level meter were compared with the readings of the device and results can be observed in Table 2 and in Figure 7.



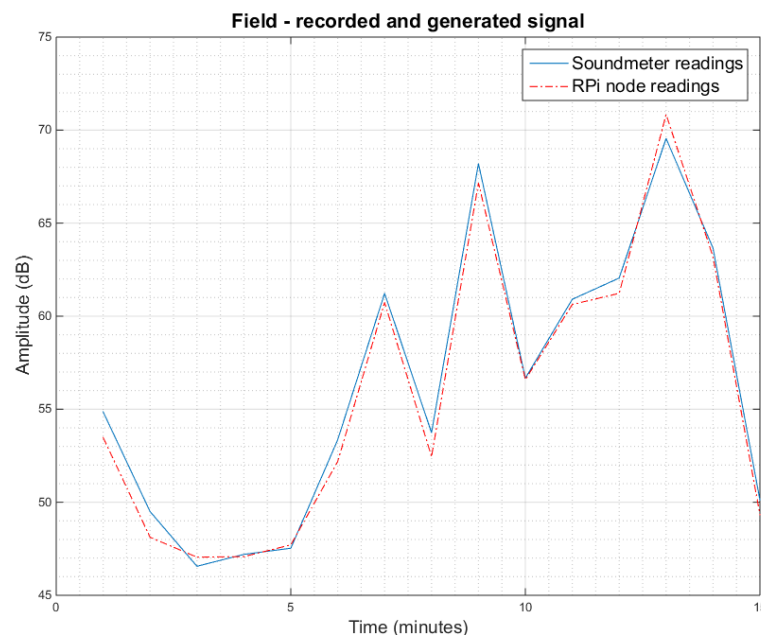
**Figure 7.** Waveform from the readings from the sound level meter and readings from the device.

**Table 2.** Levels from the sound level meter and the Raspberry Pi 2 Model B device (dB) for 15 s.

Rion NL-05	Raspberry Pi Node
90	91.1
85	85.1
80	80.3
75	75.45
70	70.39
65	65.27
60	60.2
55	55.21
50	50.18
45	46.15
40	39.85

The device was also field tested, using the integrating sound level meter, Rion NL-05. The test consisted in measurements of the  $L_{eq}$  in periods of 1 min. Fifteen sets of 1 min were tested and compared. The device and the sound level meter were placed in the façade of a building at a height of 4 m in a residential area and in front of a street with low amount of traffic. The values obtained from the sound level meter and the Raspberry node are presented in Figure 8. The differences between the levels are attributed to the higher precision of the sound level meter in its dynamic range. The precision

of the Raspberry Pi node offers enough accuracy for long-term measurements where the changes over long period of time are evaluated.



**Figure 8.** Data obtained from the sound level meter and the Raspberry node in the field test.

#### Pilot Test

A pilot test was performed using the prototype. It was conducted in the refurbishment works of the neighbourhood of “La Viña”, in the city of Lorca, Spain, which was affected by an earthquake in 2011. In these refurbishment works, innovative devices were installed and studied and, among these, the ambient noise metering device were set.

In this pilot test, two acoustic devices have been deployed. One of them was deployed inside a multi-purpose room of a neighbourhood and the other outside the building, as it can be seen in Figure 9.



**Figure 9.** Photography of the multi-purpose room where the devices are installed. A close-up of the outdoor device is shown. Inside of the building there are the gateway, the server and the indoor acoustic device.

This deployment has allowed to compare noise values between inside and outside locations and also evaluating the sound levels’ evolution in long-term measurements. The building is placed in the middle of a park of the neighbourhood, therefore the noise the device analyses mainly comes from humans and human activities sources like the noise generated by the own neighbours, the different business at the park, together with some refurbishment works next to the park, in charge of rebuilding damaged buildings after the earthquake.

The data obtained is published online for the neighbours where the noise levels in an user-friendly view are available. This tool could help people to be more aware to this kind of pollution. The understanding of the noise parameters can benefit people by adapting their activities according to decrease the noise levels. The data used in this publication can be found in <https://doi.org/10.5281/zenodo.159359>.

#### 4. Discussion

In this work, the design of an acoustic sensor that is low-cost but reliable, its components and its deployment in a pilot test have been presented.

In a first part, the design of the sensor, its components and the methodology followed for building the algorithms and the cloud connection in real time are shown. In other studies of the authors [12], low-cost platforms for ambient noise acquisition were proposed without on-board calculations and without connectivity. In this work, a fully functional sensor with cloud connectivity have been proposed and tested. Added features of on-board calculations and real-time data presentation remotely and online are included.

The platform and the methods chosen for the software developed offer some advantages such as a great versatility, a low price for the components and a big simplicity for integrating the final device in outdoors facilities, being capable of turning into part of a sensor network as a node.

The acquisition of the audio in high-quality allows us to get more advanced results, like the psychoacoustic parameters as a future outcomes. Also the power of the Raspberry Pi as the core of the device gives the possibility of doing the calculations on-board, instead of sending the raw data to a sink node or a server for doing the calculations

In a second part of the paper, a pilot test where two devices were deployed was validated. In this deployment, the two devices were working and publishing the results in real time in an IoT publishing platform. The analysis of the sound field in long-term measurements inside the device with on-board calculations and the sending and publishing of the data obtained with ease and precision have been achieved. Like an innovative challenge, and through a research process, this sensor have been used for environmental acoustics parameters calculation and for being a platform where the inhabitants of the neighbourhood could check the noise levels of their place. This make the deployment as a tool for noise awareness, apart from simply a tool for gathering data with research purposes.

In future work, the use of a different programming language would be useful in order to optimise and having more control about all the processes without the translation from the Simulink schemes to C language. Python would be the first option due to its features, being a high-level, general purpose and interpreted programming language. Its syntax is similar to Matlab and allows us to simplify the codes with less lines than C or Java.

The Raspberry Pi has proved to be a powerful, versatile and affordable computer that can be integrated in a sensor network scheme. Thanks to its connectivity options and the specifications of its core, a standalone device for high-quality sound acquisition and noise meter platform connected to the cloud have been proved to be feasible. Based on the hardware and the software development for the algorithms and connectivity, it has been demonstrated by creating a fully operational prototype deployed in a pilot test that the platform has the potential for creating a sensor network with Raspberry Pi-based nodes.

**Acknowledgments:** The work presented in this paper have been funded by project PMAFI-02-14 by the UCAM Catholic University of Murcia, Spain.

**Author Contributions:** Juan Emilio Noriega-Linares together with Juan Miguel Navarro Ruiz are co-authors of the original idea for an advanced acoustic device based in low-cost components. Juan Emilio Noriega-Linares was responsible for the design and construction of the device and he also wrote the text of this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CO <sub>2</sub>	Carbon Dioxide
SBC	Single Board Computer
IoT	Internet of Things
USB	Universal Serial Bus
GNU	GNU is Not Unix
SO	Operating System
SSH	Secure Shell
ALSA	Advanced Linux Sound Architecture
POE	Power over Ethernet
dBA	A-weighted Decibels
ADC	Analog-to-digital converter
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
XML	eXtensible Markup Language
CSV	Comma-Separated Values

## References

1. Dekoninck, L.; Botteldooren, D. Sound sensor network based assessment of traffic noise and air pollution. In Proceedings of the Euronoise 2015, Maastricht, The Netherlands, 31 May–3 June 2015; pp. 2321–2326.
2. O’Flynn, B.; Martinez-Catala, R.; Harte, S.; O’Mathuna, C.; Cleary, J.; Slater, C.; Regan, F.; Diamond, D.; Murphy, H. SmartCoast: A wireless sensor network for water quality monitoring. In Proceedings of the 32nd IEEE Conference on Local Computer Networks (LCN 2007), Dublin, Ireland, 15–18 October 2007; pp. 815–816.
3. Santini, S.; Ostermaier, B.; Vitaletti, A. First experiences using wireless sensor networks for noise pollution monitoring. In Proceedings of the Workshop on Real-World Wireless Sensor Networks, Glasgow, UK, 1–4 April 2008; pp. 61–65.
4. Ising, H.; Kruppa, B. Health effects caused by noise: Evidence in the literature from the past 25 years. *Noise Health* **2004**, *6*, 5–13.
5. Goines, L.; Hagler, L. Noise pollution: A modern plague. *South. Med. J.* **2007**, *100*, 287–294.
6. Singh, N.; Davar, S.C. Noise pollution-sources, effects and control. *J. Hum. Ecol.* **2004**, *16*, 181–187.
7. Jeon, J.Y.; Lee, P.J.; You, J.; Kang, J. Perceptual assessment of quality of urban soundscapes with combined noise sources and water sounds. *J. Acoust. Soc. Am.* **2010**, *127*, 1357–1366.
8. Brown, A.L.; Kang, J.; Gjestland, T. Towards standardization in soundscape preference assessment. *Appl. Acoust.* **2011**, *72*, 387–392.
9. Healey, P. *Urban Complexity and Spatial Strategies: Towards a Relational Planning for Our Times*; Routledge: Abingdon, UK, 2006.
10. Chinrungrueng, J.; Sunantachaikul, U.; Triamlumlard, S. Smart parking: An application of optical wireless sensor network. In Proceedings of the International Symposium on Applications and the Internet Workshops, Hiroshima, Japan, 15–19 January 2007; p. 66.
11. Kim, S.; Pakzad, S.; Culler, D.; Demmel, J.; Fenves, G.; Glaser, S.; Turon, M. Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks. In Proceedings of the 6th International Symposium on Information Processing in Sensor Networks, Cambridge, MA, USA, 25–27 April 2007; pp. 254–263.
12. Segura-Garcia, J.; Felici-Castell, S.; Perez-Solano, J.J.; Cobos, M.; Navarro, J.M. Low-Cost Alternatives for Urban Noise Nuisance Monitoring Using Wireless Sensor Networks. *IEEE Sens. J.* **2015**, *15*, 836–844.
13. Manvell, D. Utilising the Strengths of Different Sound Sensor Networks in Smart City Noise Management. In Proceedings of the Euronoise 2015, Maastricht, The Netherlands, 31 May–3 June 2015; pp. 2303–2308.
14. Farrés, J.C. Barcelona noise monitoring network. In Proceedings of the Euronoise 2015, Maastricht, The Netherlands, 31 May–3 June 2015; pp. 218–220.
15. Hakala, I. Area-based environmental noise measurements with a wireless sensor network. In Proceedings of the Euronoise 2015, Maastricht, The Netherlands, 31 May–3 June 2015; pp. 2351–2356.
16. Yen, J. H. U.S. PCI/ISA Bus Single Board Computer Card/CPU Card and Backplane Using Eisa Bus Connectors and Eisa Bus Slots. U.S. Patent 5,852,725, 22 December 1998.

17. Curran, M.A.; Diehl, R.J.; Peck, G.A.; Rahaman, K.L. Single Board Computer Quotation and Design System and Method. U.S. Patent 6,898,580, 24 May 2005.
18. Tso, F.P.; White, D.R.; Jouet, S.; Singer, J.; Pezaros, D.P. The Glasgow Raspberry Pi cloud: A scale model for cloud computing infrastructures. In Proceedings of the International Conference on Distributed Computing Systems, Philadelphia, PA, USA, 8–11 July 2013; pp. 108–112.
19. Noriega-Linares, J.E.; Navarro, J.M.; Felici, S.; Segura, J. Monitorización del Aislamiento a Ruido Aéreo Mediante Redes de Sensores Inalámbricas. In Proceedings of the Tecniacústica 2014, Murcia, Spain, 29–31 October 2014; pp. 351–359.
20. UNE-ISO 1996-1:2005. *Acoustics: Description, Measurement and Assessment of Environmental Noise. Part 1: Basic Quantities and Assessment Procedures*; ISO: Geneva, Switzerland, 2005.
21. Passchier-Vermeer, W.; Passchier, W.F. Noise exposure and public health. *Environ. Health Perspect.* **2000**, *108*, 123–131.
22. Raspberry Pi 2 Model B Datasheet. Available online: <https://cdn-shop.adafruit.com/pdfs/raspberrypi2modelb.pdf> (accessed on 5 July 2016).
23. T-bone GC100. Available online: <http://www.tbone-mics.com/en/product/information/details/-d9d6e87219/> (accessed on 5 July 2016).
24. Raspbian.org. Available online: <https://www.raspbian.org/> (accessed on 5 July 2016).
25. Secure Shell. Available online: [https://en.wikipedia.org/wiki/Secure\\_Shell](https://en.wikipedia.org/wiki/Secure_Shell) (accessed on 5 July 2016).
26. AlsaProject. Available online: [http://www.alsa-project.org/main/index.php/Main\\_Page](http://www.alsa-project.org/main/index.php/Main_Page) (accessed on 5 July 2016).
27. Mendelson, G. All You Need To Know About Power over Ethernet (PoE) and the IEEE 802.3af Standard. Available online: [http://kondorsecurity.com/store/media/pdf/PoE\\_and\\_IEEE802\\_3af.pdf](http://kondorsecurity.com/store/media/pdf/PoE_and_IEEE802_3af.pdf) (accessed on 5 July 2016).
28. Inyector PoE TL-POE150S. Available online: <http://www.tp-link.com/ar/products/details/?model=TL-POE150S> (accessed on 5 July 2016).
29. Splitter PoE TL-POE10R. Available online: <http://www.tp-link.com/ar/products/details/?categoryid=234&model=TL-POE10R#over> (accessed on 5 July 2016).
30. Van Renterghem, T.; Thomas, P.; Dominguez, F.; Dauwe, S.; Touhafi, A.; Dhoedt, B.; Botteldooren, D. On the ability of consumer electronics microphones for environmental noise monitoring. *J. Environ. Monit.* **2011**, *13*, 544–552.
31. WM-61A Datasheet. Available online: <http://industrial.panasonic.com/cdbs/www-data/pdf/ABA5000/ABA5000CE22.pdf> (accessed on 26 August 2016).
32. NC-74 Datasheet. Available online: <http://www.noise-and-vibration.co.uk/files/pdfs/NC-74.pdf> (accessed on 5 July 2016).
33. Mainetti, L.; Patrono, L.; Vilei, A. Evolution of wireless sensor networks towards the internet of things: A survey. In Proceedings of the 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 15–17 September 2011; pp. 1–6.
34. Luo, J.; Chen, Y.; Tang, K.; Luo, J. Remote monitoring information system and its applications based on the Internet of Things. In Proceedings of the International Conference on Future BioMedical Information Engineering (FBIE 2009), Sanya, China, 13–14 December 2009; pp. 482–485.
35. Internet of Things—ThingSpeak. Available online: <https://thingspeak.com/> (accessed on 5 July 2016).
36. IoT—ThingSpeak Results. Available online: <https://thingspeak.com/channels/38915> (accessed on 26 August 2016).
37. IoT—ThingSpeak Results. Available online: <https://thingspeak.com/channels/38916> (accessed on 26 August 2016).

