euronoise 2021
25th to 27th of October
Madeira, Portugal - online

# Implementation and performance assessment of a MEMS-based Sound Level Meter

**Savanne Rémy Kham[1], Patrick Marmaroli[1], Jordan Minier[2], Romain Boulandet[1]**

[1]HEPIA, HES-SO University of Applied Sciences and Arts Western Switzerland, Geneva, Switzerland.
savanne.kham@gmail.com, patrick.marmaroli@hesge.ch, romain.boulandet@hesge.ch
[2]Occupational Health & Safety and Environmental Protection Unit, CERN, Geneva, Switzerland.
jordan.minier@cern.ch

**Abstract**

Long-established industries in rural areas are now creating noise issues for new communities that have developed nearby. Proactive control of noise emission has therefore become a priority for industries concerned with respecting the tranquillity of residents. The traditional method of noise monitoring utilizes a sound level meter (SLM) which must be installed to areas of interest by trained and experienced personnel. Since a class 1 SLM is relatively expensive, long-term monitoring of large areas can hardly be considered. In this paper, we discuss the practical implementation of a stand-alone MEMS-based SLM capable of recording one-minute time samples and wirelessly transmitting sound level values in third octave bands in the range from 20 Hz to 8 kHz, every 15 minutes. The current prototype uses a digital MEMS microphone with I2S interface connected to an Arduino MKR WiFi 1010 microcontroller for an overall cost of less than 100 CHF. Technical choices for the implementation of the prototype as well as a benchmark with a class 1 SLM are discussed in this paper.

**Keywords:** low-cost sensor, urban acoustics, MEMS transducers, sound monitoring.

## 1 Introduction

Noise pollution is a persistent problem for city dwellers. Recent studies have estimated that nine out of ten people in major cities are daily exposed to noise levels exceeding international guidelines [1]. Urban noise can have negative health effects, including sleep disturbances, hearing loss, cognitive impairment, and high blood pressure. Various low-cost sensor prototypes have recently been developed for noise monitoring and mapping in smart cities. In [2], a wireless noise surveillance device is built around a Raspberry Pi powered by an electrical network outlet and connected by Ethernet to a router and WiFi via a USB dongle. Sound acquisition is then performed from a USB electret microphone with analog-to-digital converter (ADC) or an analog MEMS microphone [3]. In [2] and [3], the global acoustic descriptors (instantaneous, equivalent continuous, day-evening-night, or percentiles sound pressure levels) are calculated in the time domain, and the frequency analysis is performed using a set of one-third octave band digital filters. In [4], a low-cost urban acoustic monitoring device based on a Tronsmart MK908ii mini PC with Wi-Fi is designed to study noise exposure in New York City. Moreover, an Arm Cortex-A9 processor is used to equalize the frequency response of the analog MEMS microphone using a FIR filter. More recently, a one-year measurement campaign was carried out in Carouge, Geneva, to map the city's urban noise [5]. Each unit was battery powered and equipped with a digital MEMS microphone. Low-Power Wide-Area Network (LPWAN) technology was used to transmit data with low power consumption. The overall sound level, in dB(A), was calculated in the time domain.

In this article, we discuss the practical implementation of a battery-powered MEMS stand-alone sound level meter prototype allowing frequency analysis of one-minute sound recordings in third octave bands from 20 Hz

to 8 kHz. We also present the acoustic performance of a prototype measured in an anechoic chamber and conclude with a comparison to a class 1 sound level meter.

# 2 Hardware design

This section describes briefly the main characteristics needed to select and implement an audio embedded system based on MEMS microphones for a low-cost noise monitoring application operating in the frequency domain.

## 2.1 MEMS microphones

MEMS microphones are micro-metric scale acoustic transducers that convert pressure waves into an electrical signal. These devices are embedded in a plastic package and a metal shield acting as a Faraday cage that protects the MEMS microphone from radiated disturbances [6]. Inside the package, there is usually an application specific integrated circuit (ASIC) accompanied to the MEMS sensor that performs conversion of the input and output signals at a faster rate than it would be by software. As the analog output level is small, typically a few mV, it must be amplified before converting it into a digital signal. A picture of MEMS microphone is depicted in Figure 1.

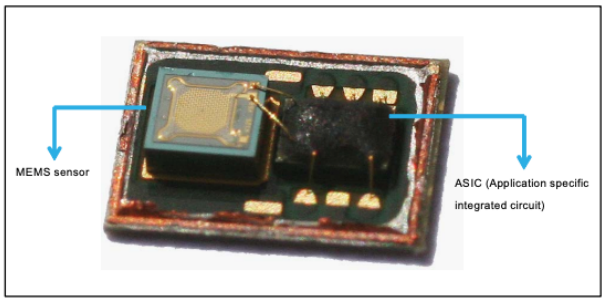


Figure 1: Example of MEMS microphone inside its package [6].

Digital MEMS microphones are analog microphones with digital output that contain filters, amplifiers and analog-to-digital converter (ADC) inside an ASIC. The main advantage of the digital MEMS microphone over analog ones is that it takes less integration efforts in addition to an immunity against radio frequency noise and electromagnetic interference due to differential transmission lines and logic signal voltage levels.

For this project, we selected a breakout board from Adafruit [7] for the Knowles SPH0645LM4H digital MEMS microphone [8]. It is an I2S MEMS microphone which is convenient to ease and accelerate the development of our proof of concept. At the time of writing this article, this unit costs approximately 7 CHF. With a signal to noise ratio of 65 dB and a dynamic range of 91 dB, this unit theoretically allows measuring sound pressure level from 29 dB SPL to 120 dB SPL. The maximum power consumption is 600 µA and it runs between 1.62 V to 3.6 V. Another attractive point is that the sensitivity (-26 dBFS / 50 mV/Pa) given in the datasheet is linear between 50 Hz and 10 kHz.

### 2.2 Computing platform

Our specification in terms of memory was to be able to measure the equivalent continuous sound levels over one minute for frequencies up to 8 kHz. With a sample rate set of 16 kHz, each sound level estimation therefore requires 960'000 samples. Since the output data word length is 24 bits per channel for the selected MEMS microphone, the I2S interface must be configured with a minimal bit per sample length of 32 bits (4 bytes). Consequently, the needed memory size to store one minute of signal is $4 \cdot 960'000 = 3.84$ MB, which is way larger than the classical storage capacity of microcontrollers. So we used 16 MB of SRAM working with the SPI interface to store the one-minute recordings. A microSD card breakout (see Figure 2) has also been employed to store the raw samples during characterization and verify that the system is functioning properly.

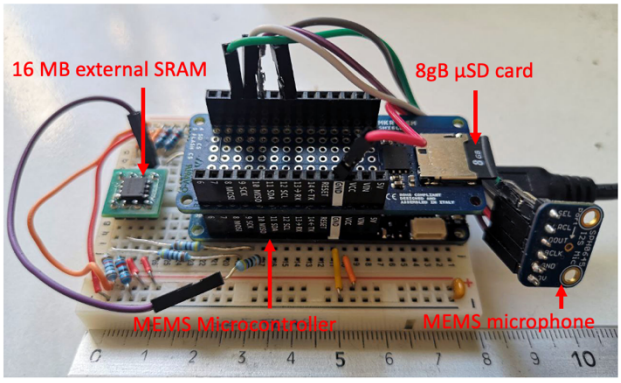


Figure 2: Proof of concept prototype.

Another constraint is that the sound level meter must operate outdoors and permanently. Therefore, power consumption is another important criterion to consider. In order to save battery, we used an embedded system based on an Arduino MKR WiFi 1010 [9] microcontroller running on an Arm Cortex-M0+ [10] processor, including I2S interfaces, a built-in wireless connection integrated and a battery management system.

The overall cost of the prototype shown in Figure 2 is around 80 CHF.

## 3 Software design

In the following, we briefly explain the step-by-step process performed in the frequency domain to calculate the A-weighted equivalent continuous sound pressure levels in one-third octave bands (see Figure 3).

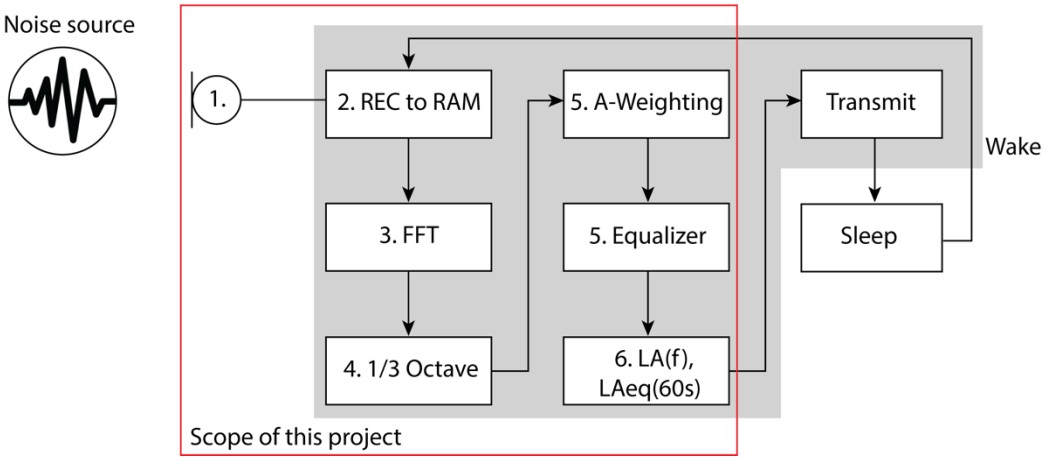


Figure 3: Overview of the signal processing pipeline.

1. Once the I2S interface is ready, the MEMS microphone is powered up and the audio recording in the SRAM starts a few seconds later.

2. The data is partitioned into chunks of 128 samples (equivalent to 8 ms at a sampling rate of 16 kHz). A one-minute recording therefore represents 7500 chunks. Once entirely acquired, the DC component of this recording is removed with the help of the function arm_mean_f32 from the CMSIS DSP library. Also, the samples are converted into Pascal by scaling them with the sensitivity previously estimated during calibration.

3. Time domain samples stored in the SRAM are then partitioned into slices of 2048 samples; each slice is transformed into the frequency domain with 7.8 Hz resolution using the function arm_rfft_fast_f32 from the CMSIS DSP library.

4. For each slice, the level of a given third octave band is obtained by summing the power of each bin belonging to the considered frequency range, before being converted into dB.

5. Finally, both A-weighting and corrective equalisation are applied onto each third octave band.

6. Step 3 to 5 is repeated until the whole recording is processed, and level per bands are averaged over time for delivering a result.

Rather than employing IIR or FIR filters which require many taps and whose coefficients can be difficult to estimate with precision, the above FFT-based method is more advantageous in terms of computation time. Note that advanced microcontrollers such as Arm Cortex-M4 or Cortex-M7 have built-in DSP hardware and libraries for FFT which would further reduce computational time.

## 4 Results

This section describes the methodology followed for assessing the sound level estimation algorithm. All measurements were carried out in an anechoic condition unless specified.

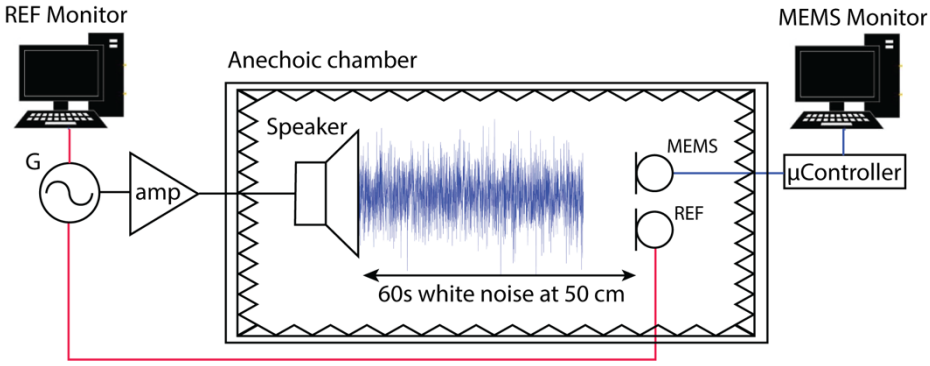


Figure 4: Characterization setup in anechoic chamber.

The characterization setup is depicted in Figure 4. A source loudspeaker is connected to a signal generator (B&K type 3160-A-042) which delivers an input signal through a conditioning amplifier (B&K Nexus Type 2692-A-0I). The MEMS microphone under test is placed next to a reference microphone (B&K Type 4954 ¼'' free-field), at 50 cm from the source loudspeaker.

### 4.1 MEMS sensor calibration

Figure 5 shows how we have tested our algorithm for calculating the sound pressure level in the frequency domain. As shown in Fig. 5, we designed and used an adapter to connect the MEMS microphone to the slot of a pistonphone (B&K Type 4231).

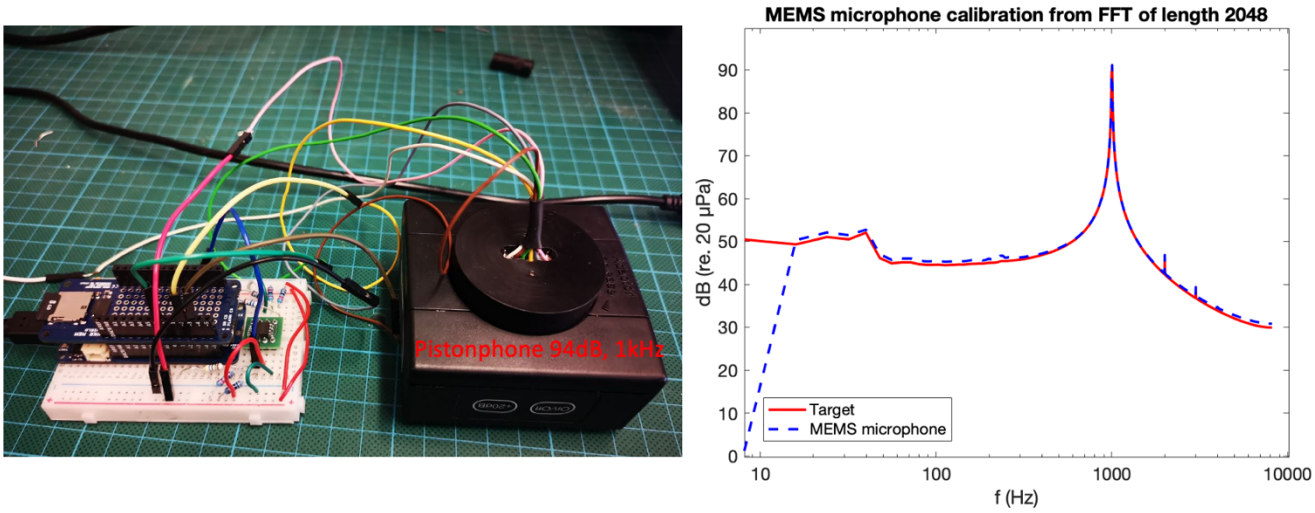


Figure 5: Setup for FFT algorithms verification and MEMS sensor calibration.

On the plot of the frequency response, "Target" refers to the FFT calculation in MATLAB and "MEMS microphone" refers to the FFT computed in the Arduino embedded system. This setup was also used to calibrate the MEMS microphone using an IIR bandpass filter between 800 Hz and 1200 Hz.

### 4.2 Frequency response

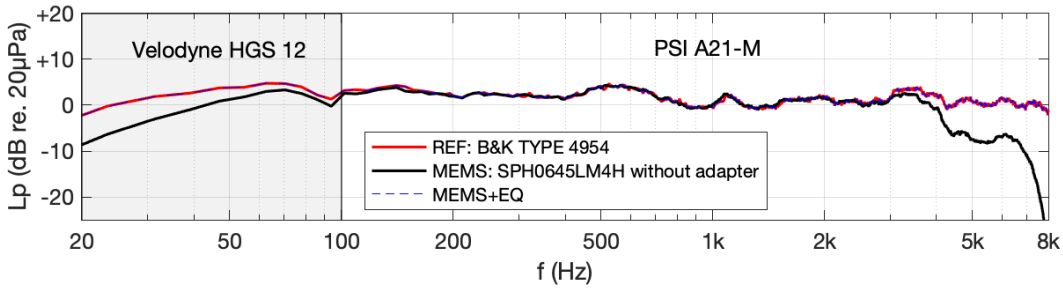


Figure 6: Measured frequency response of SPH0645LM4H.

We measured the frequency response of the MEMS microphone and compared it to a B&K reference microphone in order to implement a corrective equalizer. Results are depicted in Figure 6. The sound source were a Velodyne HGS 12 for frequencies below 100 Hz, and a PSI A21-M for frequencies above 100 Hz.

The uncompensated frequency response of the SPH0645LM4H (solid black line in Fig. 6) is relatively flat and comparable to that of the reference microphone (solid red line in Fig. 6) between 30 Hz and 4.5 kHz. As predicted in the datasheet, the sensitivity of the MEMS microphone is attenuated at low frequencies from 20 Hz to 50 Hz. However, we did not expect such an attenuation for frequencies above 4 kHz. We think this is most likely due to the breakout board to which the MEMS microphone is attached.

### 4.3 Corrective equalizer

The equalizer coefficients are determined by computing the difference between the reference microphone and the MEMS microphone sound levels for each third octave band. Figure 7 shows the frequency response function (FRF) in 1/3 octave bands from a pink noise test signal. The FRF measured after equalization is shown in dashed blue line.

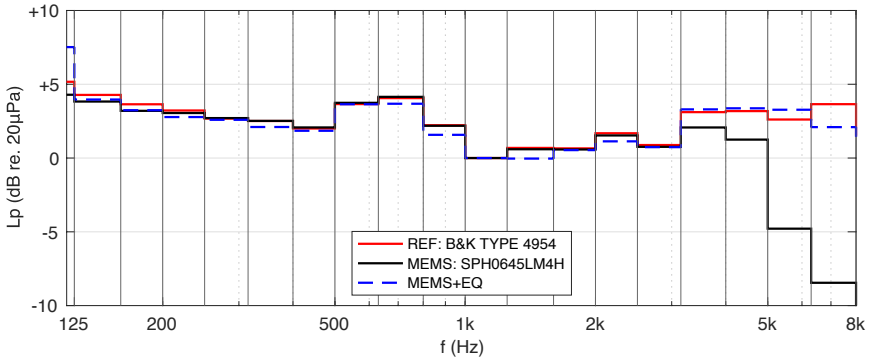


Figure 7: Pink noise test of the equalizer in one-third octave for SPH0645LM4H.

As expected, the measured FRF are relatively flat in the frequency range of interest. The correction applied in the high frequencies also works well even though it is greater than +5 dB in the 1/3 octave bands centred on 5 kHz, 6.3 kHz and 8 kHz third octave bands. The maximum error between the B&K reference microphone and

the SPH0645LM4H MEMS microphone is approximately 1.6 dB with pink noise and the standard deviation is reduced from 4.9 dB to 1.3 dB with equalization. The dynamic range is also decreased from 19.9 dB to 4 dB with the equalizer.

### 4.4 Directivity diagram

Figure 8 depicts the directivity diagram in third octave bands of MEMS microphone with and without the adapter used for the calibration step. The excitation signal is a 10 kHz bandwidth white noise centred on 5.12 kHz. Measurements are made using a Bruel & Kjaer Type 9640 turntable system by rotating the microphone relative to the loudspeaker in 5° steps.

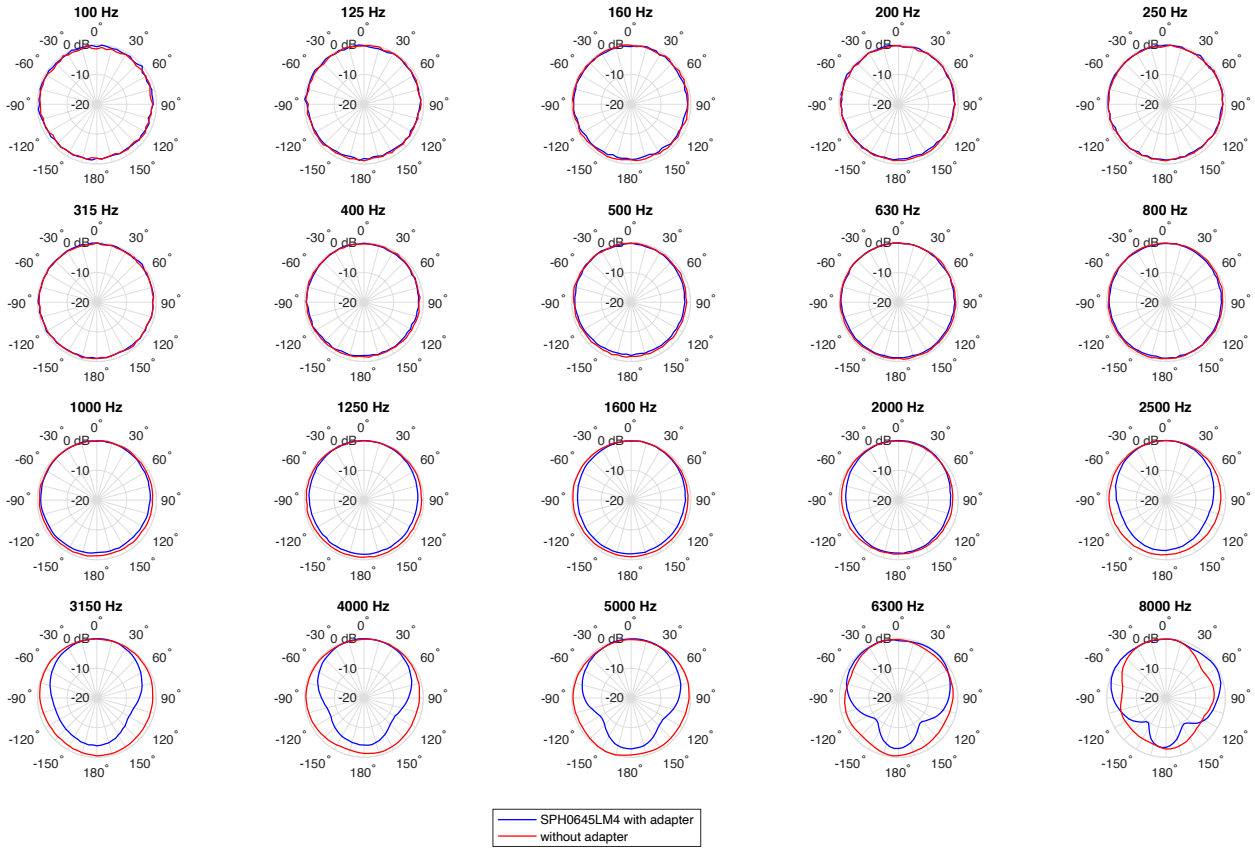


Figure 8: Directivity diagram in third octave band for SPH0645LM4H.

The SPH0645LM4H microphone is omnidirectional in the one-third octave bands from 100 Hz to 6.3 kHz, as expected from the datasheet, and nearly omnidirectional at 8 kHz. With the adapter, on the other hand, the directivity is affected above 2.5 kHz when the acoustic wavelength of interest becomes smaller than the protective case, thus creating interference.

### 4.5 Comparison with a Class 1 sound level meter

As a quick validation experiment, sound level difference between our prototype and the 01dB Fusion class 1 SLM [11] have been compared. Measurements were carried out inside a gymnasium using a drone flying over the measurement setup, as depicted in Figure 9.

Figure 9: Validation experiment setup.

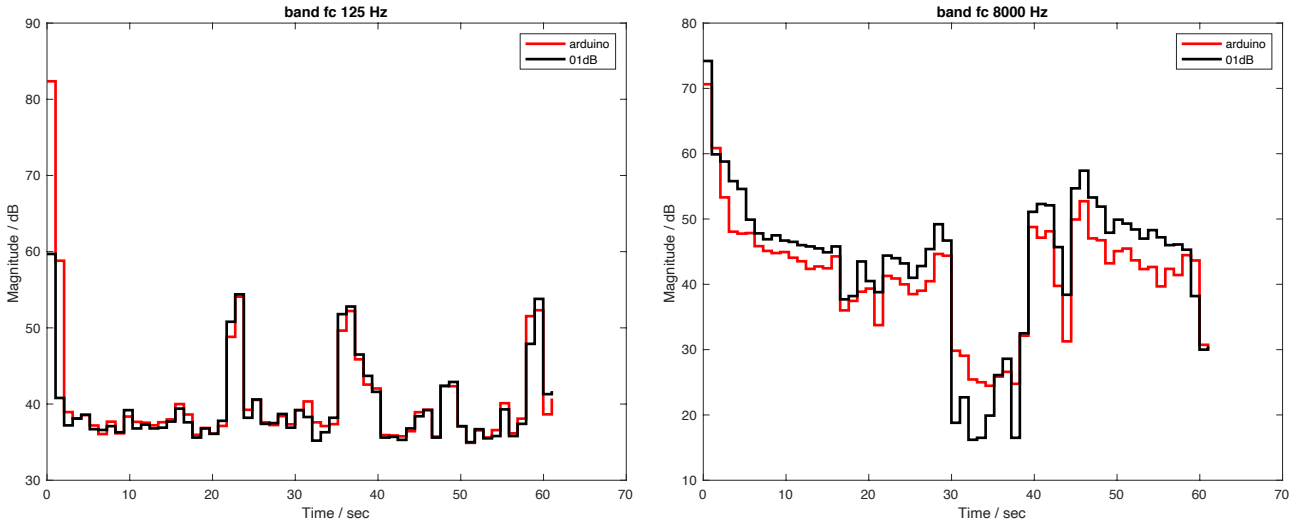


Figure 10: Benchmark of measured sound levels in the 1/3 octave bands centered on 125 Hz and 8 kHz.

Figure 10 shows a comparison of the sound levels (1 sec equivalent) obtained with the MEMS-based prototype and the class 1 SLM, in the 1/3 octave bands centred on 125 Hz and 8 kHz. Despite some differences the dynamic responses match relatively well.

In Figure 11 we provide the difference, in dB, averaged over time and for each 1/3 octave band between the prototype and the class 1 SLM. In low frequencies, below 100 Hz, the sound source (a drone) unfortunately did not provide enough energy to make a proper comparison. In high frequencies, typically above 4 kHz, some significative differences are observed with the class 1 SLM due to the fact the I2S MEMS microphone breakout board slightly affects the dynamic response. This could be improved by optimizing the design of the microphone holder. However, in the frequency bandwidth of the drone which ranges from 315 Hz to 2500 Hz, differences of less than 1 dB are observed.

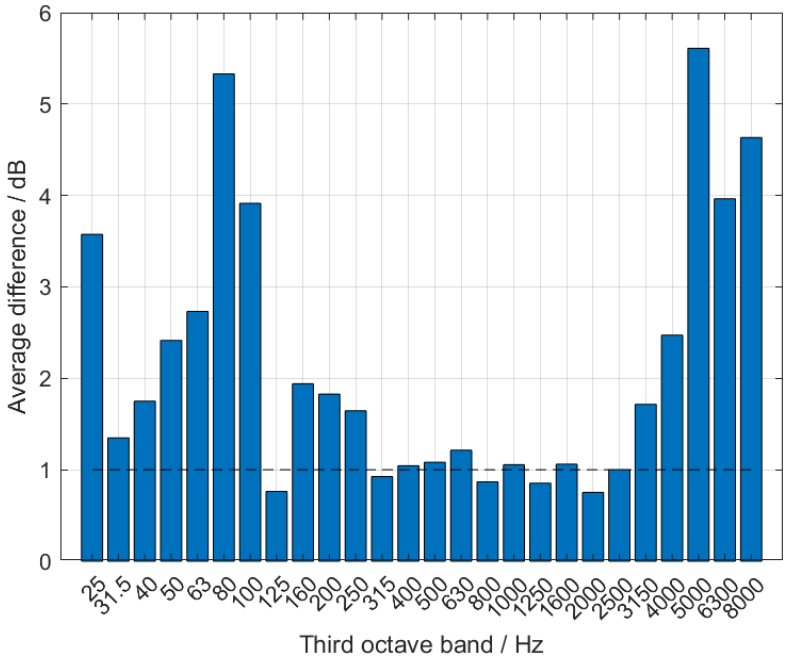


Figure 11: Average difference between the MEMS-based prototype and a class 1 SLM when measuring a drone indoor.

## 5 Conclusions

In this paper, we shared our experience regarding the development of a battery-powered and low-cost MEMS standalone sound level meter. It has been proven empirically that such a technology is reliable enough to assess sound pressure levels in one-third octave band up to 8 kHz after applying an appropriate corrective equalizer. We also showed how to calibrate the system to ensure the correctness of the calculated sound levels. This could, therefore, help to meet any challenges of large-scale environmental noise monitoring. In addition, our technical choice consisting in estimating sound pressure level using a frequency domain analysis, paves the way to the extraction of much more informative features (like spectrograms) that could feed advanced classification or detection algorithms, for applications requiring deeper soundscape analysis. This concept is therefore innovative and has many perspectives which do not only concern the monitoring of environmental noise sources.

Currently, there is not much published research or standards in the field of cost-effective embedded systems for acoustic metrology, especially for obtaining sound pressure levels by frequency bands. The MEMS-based SLM presented in this article enables accurate and continuous monitoring of sound levels in 1/3 octave band suitable for an industrial site. Beyond the simple calculation of sound levels, the identification of the source(s) responsible for noise annoyance is of great importance. Other metrics (e.g., percentile sound levels) or spectrogram calculation could be easily implemented, or machine learning techniques could complement the system to further analyze the acoustic signature of sources. The next step is also to associate several units in a network in order to monitor the noise exposure of an industrial site in a more comprehensive way.

## References

[1] **World Health Organization (WHO).** Environmental Noise Guidelines for the European Region. https://www.euro.who.int/en/health-topics/environment-and-health/noise/publications/2018/environmental-noise-guidelines-for-the-european-region-2018/.

[2] **Noriega-Linares Juan Emilio and Navarro Ruiz Juan Miguel.** On the Application of the Raspberry Pi as an Advanced Acoustic Sensor Network for Noise Monitoring. *Electronics,* Vol. 5 (4), 2016.

[3] **Sotirakopoulos Kostas, Richard Barham and Ben Piper**. Designing and evaluating the performance of a wireless sensor network for environmental noise monitoring applications. *Euroregio*, 2016.

[4] **Mydlarz Charlie, Salamon Justin et Bello, Juan Pablo.** The Implementation of Low-cost Urban Acoustic Monitoring Devices. *Applied Acoustics,* Vol. 117, 2017, pp. 207-218.

[5] **Helal Didier, Dubouloz Samuel, Mortensen Jorgen, Baldi Paolo and Royo Paul.** Real time 3D environmental noise monitoring and mapping using large-scale internet of things. *INTERNOISE 2019, Madrid,* 2019.

[6] **ST Microelectronics.** AN4426 - Application Note - Tutorial for MEMS Microphone. https://www.st.com/resource/en/application_note/dm00103199-tutorial-for-mems-microphones-stmicroelectronics.pdf

[7] **Adafruit.** Adafruit I2S MEMS Microphone Breakout - SPH0645LM4H. https://www.adafruit.com/product/3421.

[8] **Knowles.** SPH0645LM4H-1 - I2S output digital microphone. https://www.knowles.com/docs/default-source/default-document-library/sph0645lm4h-1-datasheet.pdf?Status=Master&sfvrsn=751076b1_2

[9] **Arduino.** Arduino MKR WIFI 1010. https://store.arduino.cc/arduino-mkr-wifi-1010.

[10] **ARM.** Cortex-M0+. https://developer.arm.com/ip-products/processors/cortex-m/cortex-m0-plus.

[11] **Acoem.** 01dB Fusion SLM. https://www.01db.com/fr/nos-solutions/nos-produits/sonometre/.