

Article

# Easily-Deployable Acoustic Local Positioning System Based on Auto-Calibrated Wireless Beacons

José A. Moreno \* , Fernando J. Álvarez , Teodoro Aguilera  and José A. Paredes 

Sensory System Research Group, University of Extremadura, 06006 Badajoz, Spain; fafranco@unex.es (F.J.Á.); teoaguibe@unex.es (T.A.); japaredesm@unex.es (J.A.P.)

\* Correspondence: josan@unex.es

Received: 31 January 2019; Accepted: 15 March 2019; Published: 20 March 2019



**Abstract:** Self-calibrated Acoustic Local Positioning Systems (ALPS) generally require a high consumption of hardware and software resources to obtain the user's position at an acceptable update rate. To address this limitation, this work proposes a self-calibrated ALPS based on a software/hardware co-design approach. This working architecture allows for efficient communications, signal processing tasks, and the running of the positioning algorithm on low-cost devices. This fact also enables the real-time system operation. The proposed system is composed of a minimum of four RF-synchronized active acoustic beacons, which emit spread-spectrum modulated signals to position an unlimited number of receiver nodes. Each receiver node estimates the beacons' position by means of an auto-calibration process and then computes its own position by means of a 3D multilateration algorithm. A set of experimental tests has been carried out where the feasibility of the proposed system is demonstrated. In these experiments, accuracies below 0.1 m are obtained in the determination of the receptor node position with respect to the set of previously-calibrated beacons.

**Keywords:** Acoustic Local Positioning System (ALPS); wireless beacons; auto-calibration; digital signal processing; embedded systems

## 1. Introduction

The applications and services of people and object localization and context analysis are a growing demand in consumer society, so it is necessary to investigate and develop alternative and complementary technologies to existing global location [1] to cover also indoor environments with better performance, precision, and robustness [2,3].

Most indoor positioning systems proposed so far can be classified into three main categories. First are those based on the measurement of radio frequency Received Signal Strength (RSS). These systems require an external infrastructure to generate the RF signals, but they usually benefit from already deployed WPAN [4,5] or WLAN [6,7] transceivers. This technology provides a positioning accuracy between one and a few meters, due to the complex propagation mechanisms of RF signals indoors that cause a large variability of their RSS measurements. Secondly, we find those systems that make use of inertial sensors (accelerometers and gyroscopes) to perform dead reckoning [8,9]. These systems do not need an external infrastructure to operate, but they provide low positioning accuracy due to their inherent cumulative error, typically around 10% of the total traveled distance. Finally, the third category covers all those systems that measure either the Time of Arrival (ToA) or the Time Difference of Arrival (TDoA) of the acoustic signals emitted from a set of fixed beacons, to render the target's location by following a trilateration or multilateration procedure, respectively [10]. By using encoded emissions and pulse compression techniques to detect them, these systems can achieve centimetric positioning accuracy, high robustness to in-band noise, and multiple access capability [11].

Nevertheless, one of the main inconveniences that are attributed to Acoustic Local Positioning Systems (ALPS) is their limited scalability. The only way to extend the operational area of these systems is to deploy new beacons at known and precise locations, whose coordinates will be used to solve the trilateration or multilateration equations, provided that the simultaneous coverage of a minimum of three or four beacons is always ensured, respectively. Needless to say, this is a tedious task prone to multiple errors during the beacons' placement procedure, even more given the usually wired infrastructure of these beacons.

To overcome the problem stated above, this work proposes a novel ALPS based on a set of easily-deployable wireless beacons, whose main feature is their capability to conduct an auto-calibration process that eliminates the need to measure their precise location during deployment. Several works can be found in the literature that have already proposed the auto-calibration of a set of acoustic beacons. Hence, in [12], the authors propose a Simultaneous Calibration and Navigation (SCAN) algorithm for Mobile Robots (MR) over a wide area. This system consists of a set of ALPS, some of which are calibrated and Globally Referenced (GRALPS), whereas the rest are not calibrated and Locally Referenced (LRALPS). The MR is in charge of calibrating the beacons of the different LRALPS under which it moves by using multiple filters running in parallel (as many as GRALPS and LRALPS). This allows obtaining the transformation vector that converts the coordinates of the local reference system to the global system. The precision obtained in the estimation of the beacons' location is below 0.2 m and 0.5 m when trilateration or multilateration are used, respectively. In [13], a self-calibration algorithm was also proposed to calculate the position of a set of ultrasonic transmitters using the nonlinear least squares minimization principle and a set of distance readings previously obtained by a walking user. This method achieves an accuracy of 20 cm in the determination of a set of four beacons' location. However, it is also worth mentioning that this system stops working when the beacons are arranged in the same plane or describing a circle. Furthermore, in [14], the self-calibration of an ultrasonic location system was implemented using two methods. The first method (explorer) used three pre-calibrated beacons to explore the complete positioning area using the Time Division Multiple Access (TDMA) protocol, while the second method (distribute and erase) progressively refined the position of the remaining beacons using an adapted trilateration system. By combining these methods, it is possible to reach accuracies of 1 mm with a distribution of  $11 \times 11$  beacons and after 1200 algorithm iterations. Nevertheless, the main limitation of this system resides in the small exploration area where it operates. Finally, in [15], the authors proposed the LOSNUS system (Localization of Sensor Nodes by Ultrasound) based on the measurement of the ToFs of a set of six transmitters and four receivers that together with the speed of sound allow defining a system of 24 equations that, once solved through a minimization algorithm, allows the definition of a system of coordinates. Using this method, calibration accuracies of less than one centimeter were achieved.

A key feature of the system proposed in this work is its implementation through hardware/software co-design [16]. This architecture of operation allows: (1) the integration of RF synchronization devices and acoustic sensors in low-cost System on Chip (SoC) platforms and (2) the efficient running of signal processing tasks and a positioning algorithm that enable real-time system operation.

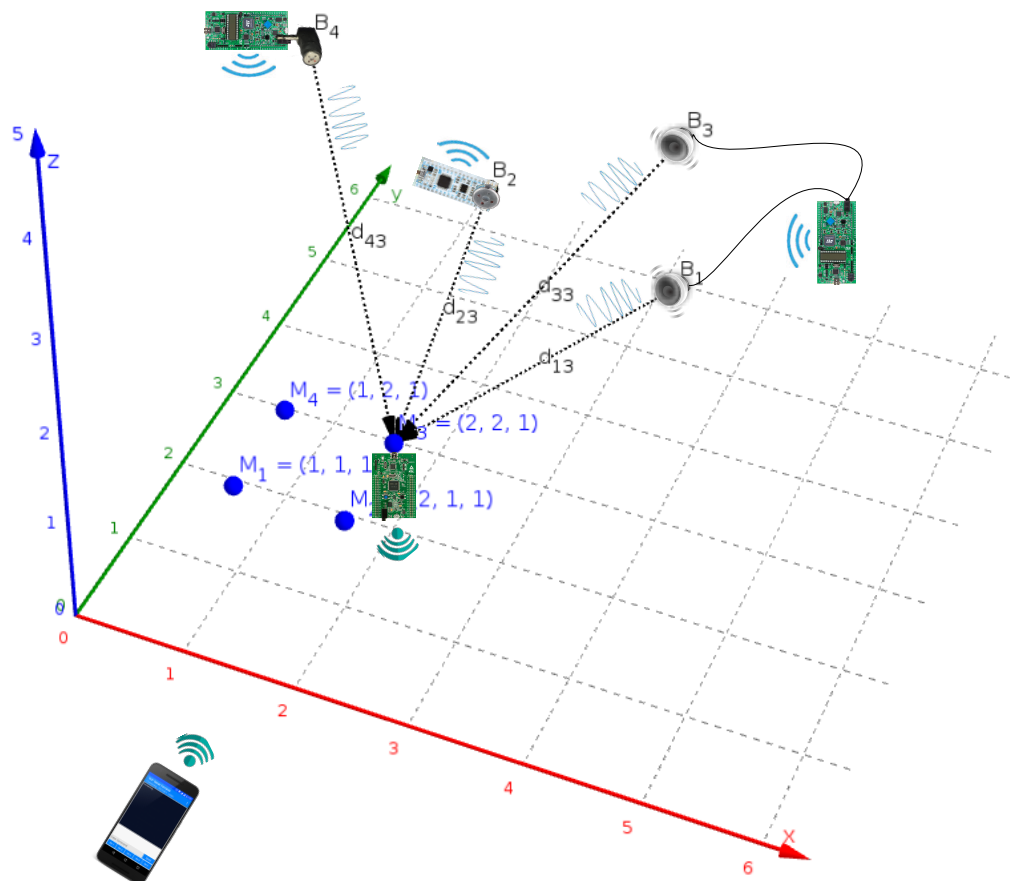
The remainder of the manuscript is organized as follows: Section 2 gives a brief system overview; Section 3 describes the proposed emitter subsystem (beacons) and signals' generation; Section 4 explains in detail the estimation of the position and the receiver subsystem; Section 5 presents the practical implementation and some results; finally, the main conclusions of this work are given in Section 6.

## 2. General Overview

This section offers an overview of the entire system, as well as the signals in the acoustic communication. The system has been built focusing on both the emission and the reception of

the acoustic signals mentioned in Section 1. Besides, it is supported by similar platforms based on the ARM [17] Cortex-M4 [18] SoC architectures. Their main characteristics will be detailed in later sections.

In Figure 1, the four beacon nodes are depicted as  $B_i$  with  $i \in \{1, 2, 3, 4\}$ . It is possible to configure the system with different implementations, but the choice has been made with the aim of minimizing the cost. There are two operation modes: the most advanced alternative supports a dual beacon operation mode by using a set of home stereo speakers ( $B_1$  and  $B_3$ ). The simplest mode uses individual loudspeakers, soldered ( $B_2$ ) or jack encapsulated ( $B_4$ ), chosen due to their small size and good performance for the desired functionality. All nodes also communicate via an RF protocol that provides the required synchronization to conduct the simultaneous emission of acoustic signals.



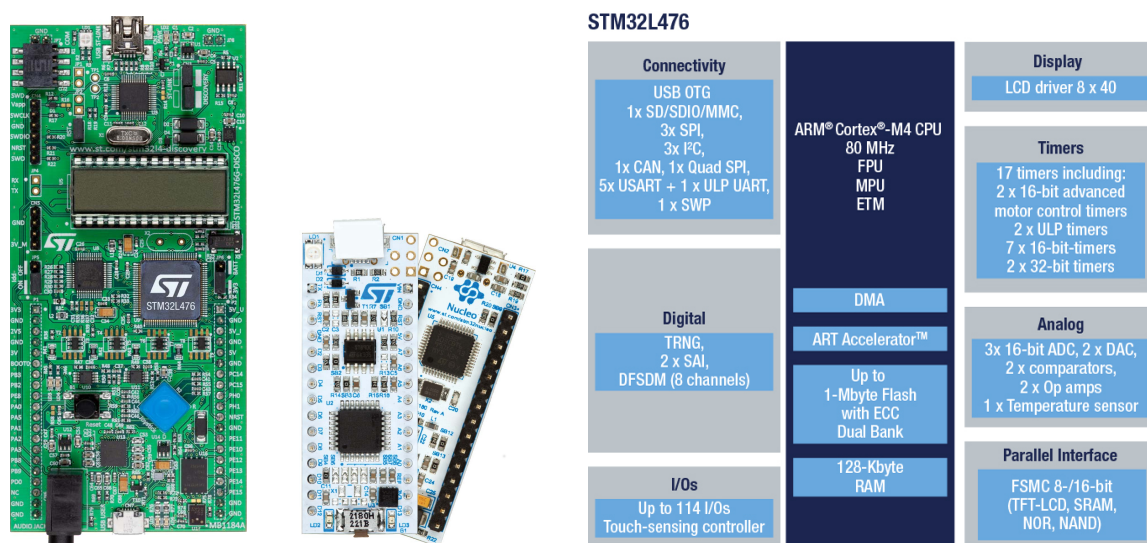
**Figure 1.** Overview of the proposed local positioning system.

Moreover, the 3D points  $M_j$  with  $j \in \{1, 2, 3, 4\}$  depict measured locations where the receiver platform, which includes a MEMS audio sensor, performs the signal acquisition and processing. The beacons are previously calibrated, and their results are remotely communicated from an adapted WiFi interface to a smartphone that shares a WLAN network.

### 3. Beacons' Structure and Signals' Generation

In order to implement a flexible and low-cost infrastructure of acoustic emission beacons, the use of energy-saving SoC-based devices has been considered. These devices can be powered by long-lasting batteries, communicate with each other wirelessly, and transmit, in a synchronized way between them, encoded sequences. This section describes the designed hardware platform, the coding of the sequences to be transmitted and the hardware/software implementation of the beacons system, including the synchronization between them.

A development board based on the 32-bit micro-controller ARM Cortex-M4 has been used for beacon prototyping. This board requires minimum performance and storage, and it is able to carry out DAC conversion. It also has some programmable timers and clock sources. This prototype works at a low power consumption rate, and it features an efficient power management capability. Each beacon makes use of an independent battery. Besides, the L4 [19] family of MCUs from ST Microelectronics [20] has been chosen, which stands out in performance and Ultra-Low Power (ULP) consumption. The available set of interfaces and peripherals in SoC STM32L476 [21] of Discovery Kit 32L476 GDISCOVERY [22] (Figure 2 left) from ST is represented on the right-hand side of the figure, although for the corresponding application, we will make use of 2 × DAC, 2 × SPI, 1 × USART, 1 × channel, 1 × timer, internal SRAM, and Flash memory for the storage of generated signal samples and the LCD controller and GPIO inputs of joystick cursors for the minimal user interface.



**Figure 2.** Development kits L476Discovery (left), L432Nucleo (middle), and a summary of resources in STM32 SoC included (right).

External to SoC, some PCB peripherals were used. Noteworthy are the audio stereo DAC CS43L22 with a Class D amplifier for external mono/stereo speakers via the jack connector, several GPIO (General Purpose Input/Output) pins, like LEDs, push buttons, joystick, and an LCD for the user interface. The SAI module in combination with the audio DAC is essential in the beacon's application, as an alternative to the SoC DAC, in order to support a dual beacon by stereo DAC code outputs. Moreover, a rear socket enables supply to the rest of the components in the base PCB through a 3.0-V lithium manganese battery with a nominal capacity of 200 mAh.

In order to enable wireless communication between the beacons and to carry out the necessary synchronization for the simultaneous emission of codes, the module MRF89XAM8A [23] of Microchip [24] has been added (Figure 3). This module is a multichannel FSK/OOK transceiver operating in the 863–870 MHz ISM frequency band. It has a PCB antenna, based on the Microchip MRF89XA [25] single-chip RF transceiver. The main characteristics are the sensitivity of  $-107$  dBm to 25 Kbps FSK with digital RSSI, the current consumption of 25 mA in transmission and 3 mA in reception, with modes of low consumption in periods of inactivity, an FIFO of 64-byte for transmit and receive data, and recognition of input synchronized words (patterns). The connection with the host platform is made by means of SPI facilities.

Finally, a miniature magnetic speaker (10 mm) of 0.3 W and  $8 \Omega$  by Kingstate has been used with the previous amplifier as part of the ALPS acoustic subsystem (Figure 3). As can be seen in the middle image of this figure, an alternative assembly has been made with the audio mono jack connection via SAI to the Discovery platform.

Each beacon has been modeled around a designed PCB (Figure 4), connected to a development platform to emit pattern codes generated from binary sequences, whose modulation symbol has been stored in their RAM blocks.

This set of components forms each beacon, which operates simultaneously with the rest of the nearby beacons by making use of the wireless connection provided by the RF transceiver module. After synchronization, these beacons emit their corresponding patterns, previously programmed through an USB port from a computer. This system is powered by its own battery incorporated in a socket of the MCU kit.

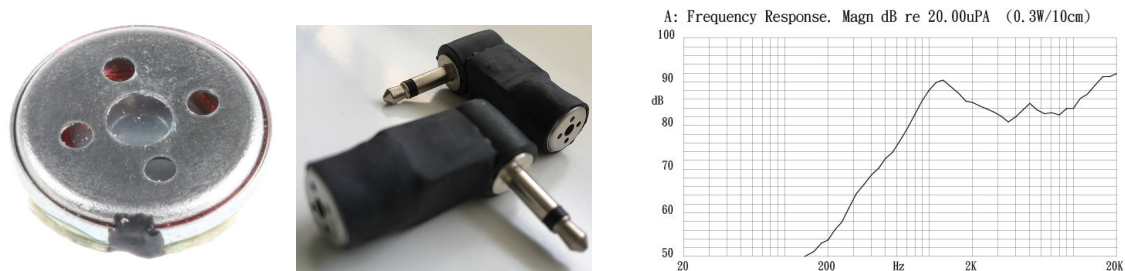


Figure 3. Miniature speaker KDMG10008C encapsulated with jack connectors and its frequency response.

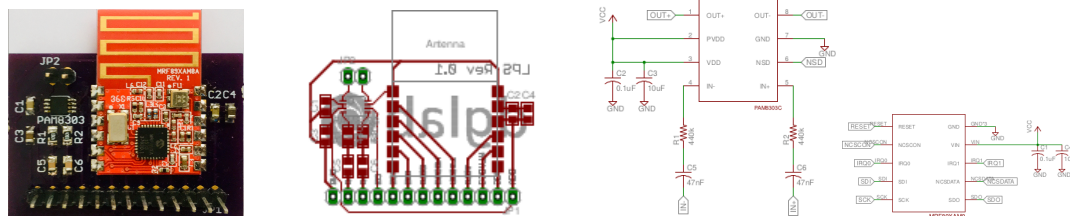


Figure 4. PCB views of the ALPS beacon including the MRF89XAM8A RF 868-MHz transceiver module.

### 3.1. Issued Sequences

The sequences' emission is carried out with the HW structure by means of the corresponding execution of the FW programmed in the flash memory of the MCU.

In the description of the developed SW and its operation, two different parts can be distinguished. They are, in sequential order, (a) the synchronization stage between environment beacons through the RF transceiver, which it is carried out in a timely way at the beginning of the system (it also serves to resynchronize possible temporary drifts); and (b) the generation of periodic sequences by means of the DAC and the acoustic transducer.

To enable TOF-based positioning, all acoustic beacons fire their emissions simultaneously. The choice of the codes has been based on getting low cross-correlation values to reduce the effects of multiple access interference. Several families of sequences have been studied for these systems, for instance, gold codes, Complementary Sequence Sets (CSS), and LS (Loosely Synchronized) codes. However, in a recent work [26], it has been demonstrated that not all of these families are equally resilient to adverse effects such as multi-access, multipath, and Doppler shift. Kasami sequences are known to represent a good compromise between the good cross-correlation values and the slow degradation of these properties associated with the movement of the receiver. Kasami sequences belong to the well-known group of pseudo-random sequences (PN), which can be generated using Linear Feedback Shift Registers (LFSR) and XOR logic gates [27]. A new Kasami sequence can be obtained from an  $m$  sequence and the decimated and chained version of this sequence carrying out the module two addition between the first and the delayed version of the second one, that is:

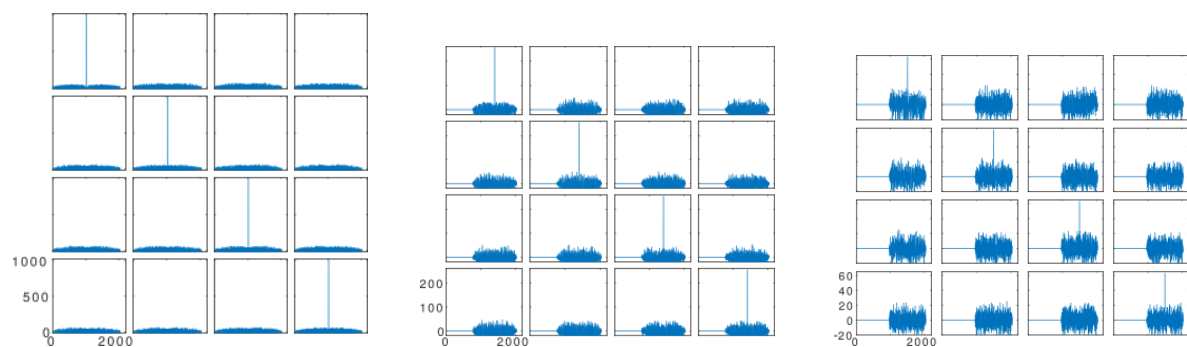
$$c = m_1 \oplus D^l m_2 \quad \text{with} \quad l < L \quad (1)$$

where  $m_1$  is the maximal sequence of length  $L = 2^N - 1$  with  $N$  even;  $m_2$  is the obtained sequence decimating  $m_1$  a factor of  $q = 2^{N/2} + 1$  and concatenating result  $q$  times;  $\oplus$  is the module two

sum operator and  $D^l m_2$  the resulting sequence by circular shifting  $l$  positions the  $m_2$  sequence. Following this procedure, families of Kasami sequences have been generated for  $N = 6, 8, 10$ , with  $L = 63, 255, 1023$  bits, respectively, each, from which the set of four sequences with better values has extracted lower cross-correlation as emission patterns for the LPS in each case.

As is well known, the higher the sequence length, the better the correlation under noisy critical conditions, so increasing this size can lead to a greater reliability and robustness of the system. As a counterpart, the larger the emission period, the heavier the computational load at the receiver.

In the proposed system, sequences of (1023) bits (Figure 5 left) were used in the following way: Four codes of 255 bits have been generated from four Kasami codes of 1023 bits with better properties (Figure 5 middle) by taking the central part of the previous ones (bits from  $512 - 127 - 512 + 127$ ). Similarly, four codes of 63 bits were generated by taking the bits from  $512 - 31 - 512 + 31$  (Figure 5 on the right). All families thus generated retain good cross-correlation properties, quasi-optimal orthogonal performance, and good cross-correlation values with the parent codes of the 1023-bit family.



**Figure 5.** Autocorrelation of generated Kasami codes with 1023 bits (left), 255 bits (middle), and 63 bits (right).

The four sequences have been digitally modulated with a Binary Phase Shift Modulation (BPSK) scheme, allowing the adaptation of the emission spectrum to the frequency response of the acoustic transducers. This modulation scheme has been widely used to transmit binary codes in sonar systems, performing the detection of received echoes by pulse compression. Each bit of code  $c[n]$  is modulated in the proposed BPSK scheme with only one period of carrier signal, whose phase, zero or  $\pi$ , is given by the value of the corresponding bit to obtain the modulated pattern as:

$$p[n] = \sum_{i=0}^{L-1} c[i] \cdot s[n - i \cdot M] \quad (2)$$

where  $L$  is the code length,  $s[n]$  represents the modulation symbol, and  $M$  is the number of samples in this symbol, whose value is given by the product between the number of carrier cycles that constitute the symbol and the quotient between the sampling and carrier frequencies.

### 3.2. Implementation

The resources used in the MCU for this phase were: a timer, a DAC, and a DMA channel between the memory and the peripherals. The pin configuration was required to use external resources such as the DAC output, as shown in Figure 6, in addition to a correct clock configuration for the use of the timer and other resources of the MCU (Figure 7).

The DAC module in the STM32 was a 12-bit converter with two output channels that support audio functions. Only one of these channels will be necessary in the proposed application. The data format used was 12 bits aligned to the right, giving a range of  $000_{16} - FFF_{16}$  ( $4096_{10}$ ). To start the conversion of the DAC channel, there were timers associated with a trigger register. In this case, the conversion started with the update event of a timer, which corresponded to its overflow. The timer

was configured with a sampling frequency of  $f_s = 96$  kHz. The system clock was set as a multiple of that frequency to favor the exact division and precisely obtain  $f_s$  in the timer.

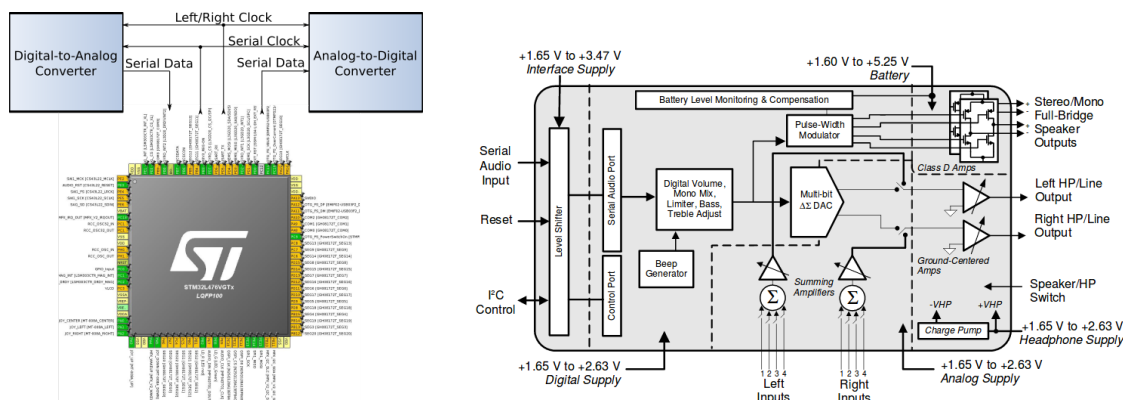


Figure 6. Simplified scheme of the SAI interface with the SoC pinout and the functional diagram of audio DAC.

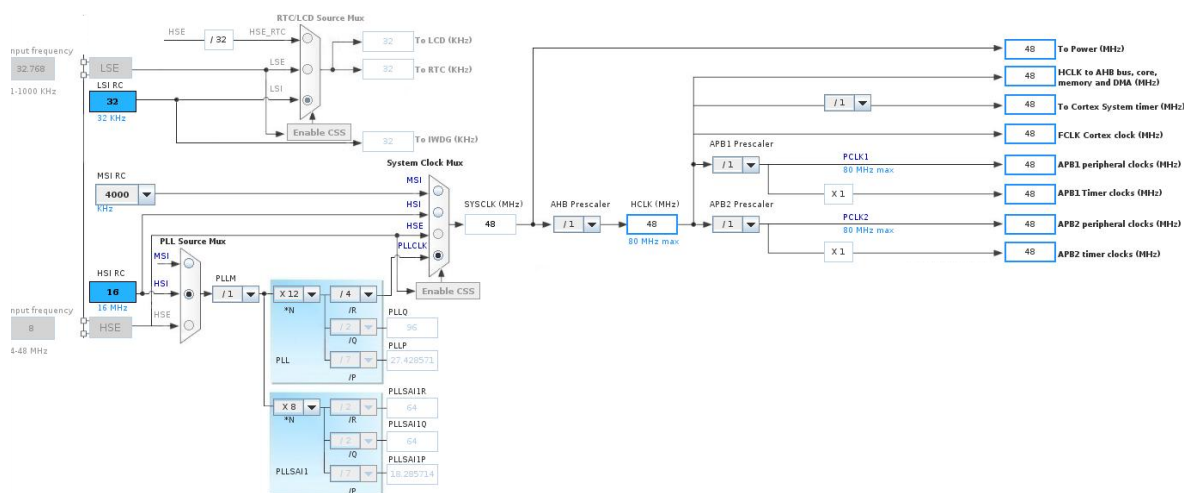


Figure 7. Clock settings for signal generation with STM32L476VG.

In order to handle external loads without using an operational amplifier, the DAC channels integrate output buffers that can be activated depending on the application, thus avoiding any voltage drop with respect to the reference voltage. The use of this functionality was necessary in our case to obtain a maximum output voltage level of  $V_{ref} = 3.3$  V. If the digital value of the samples was in the range  $[-1, 1]$ , it needed to be calibrated again to have a positive value between zero and  $FFF_{16}$  (corresponding to the range from 0 V–3.3 V):

$$y_{digital}(x) = (s(x) + 1) \frac{(0xFFF + 1)}{2} \tag{3}$$

The digital outputs were converted to output voltages by a linear conversion between zero and  $V_{ref}$ , being determined by the following expression:

$$DAC_{out} = y_{analog}(x) = V_{ref} \frac{y_{digital}(x)}{DAC_{maxval}} = 3.3V \frac{y_{digital}(x)}{(0xFFF + 1)} \tag{4}$$

The sample value table was saved in memory and transferred by DMA, in steps triggered by the same timer as the DAC.

In addition, two Serial Audio Interfaces (SAI) are available on the STM32 for digital audio devices' connection. This interface is one of the most common mechanisms for transferring data from two

audio channels between devices within a system, selecting between several standard protocols such as  $I^2S$  [28], PCM-coded pulse modulation, Intel AC'97 [29], or user-defined ones. As can be seen in Figure 6, in brief, a SAI can establish a communication bus with ADC and DAC. One of the SAI interfaces of the STM32 was used in this case to communicate with the audio DAC CS43L22 [30] on the platform STM32L476 via bus  $I^2C$  in address  $94_{16}(148_{10})$  and by means of a stereo jack connector to an omnidirectional MEMS microphone, which provides a Pulse Density Modulated (PDM) digital signal.

The alternative of using SAI with audio DAC from the CS43L22 codec with respect to the initial DAC configuration has several advantages. On the one side, it simplifies the PCB design without a Class D amplifier already included in the codec; on the other side, data scaling becomes unnecessary for the unsigned format with 12 bit with respect to inherent the flexibility of data types and configuration in codec via registers and BSPs from the STM32.

The transfer of the 1023-bit code, BPSK modulated with an oversampling factor ( $\frac{f_s}{f_c} = 6$ ), was carried out by associating a buffer with 6138 samples to a DMA channel connected to the SAI, making possible a direct transmission from memory without the need for a CPU intervention. The periodicity of the emission was obtained through a circular configuration of the pointed DMA buffer.

### 3.3. Inter-Beacons' Synchronization

The SoC resources used for this phase consisted solely of the SPI peripheral for serial communication with the RF module, although the LCD mini-display and GPIO pins at the BSP level were also used for debugging purposes. The pin-out configuration is shown in Figure 6 and the clock configuration for these resources in Figure 7.

In this case, the SW development was based on the construction of an RF module device driver for the basic transmission and reception initialization functions. From these functions, a simple protocol was used by a beacon with master functions to send a synchronization message to the rest of the slave beacons. The slave ones responded with a configuration message and restarted the corresponding sequence-sending timer.

## 4. Receiver Module and Signal Processing

The receiver prototype was based on the evaluation board STM32F4 Discovery Kit [31,32] (Figure 8) assembled with the 32-bit microcontroller ARM [17] Cortex-M4 [18] STM32F407VG [33] from the F4 [34] family of ST Microelectronics [20] MCUs, suitable for RT audio processing. Its main resources are 168-MHz CPU, FPU, and ART accelerator, equipped with the MEMS audio sensor omnidirectional digital microphone MP45DT02 [35] (Figure 9), along with a CS43L22 DAC [30]. Similar to the rest of the SoC family, it includes multiple interfaces and peripheral resources, as shown on the right-hand side of Figure 8.



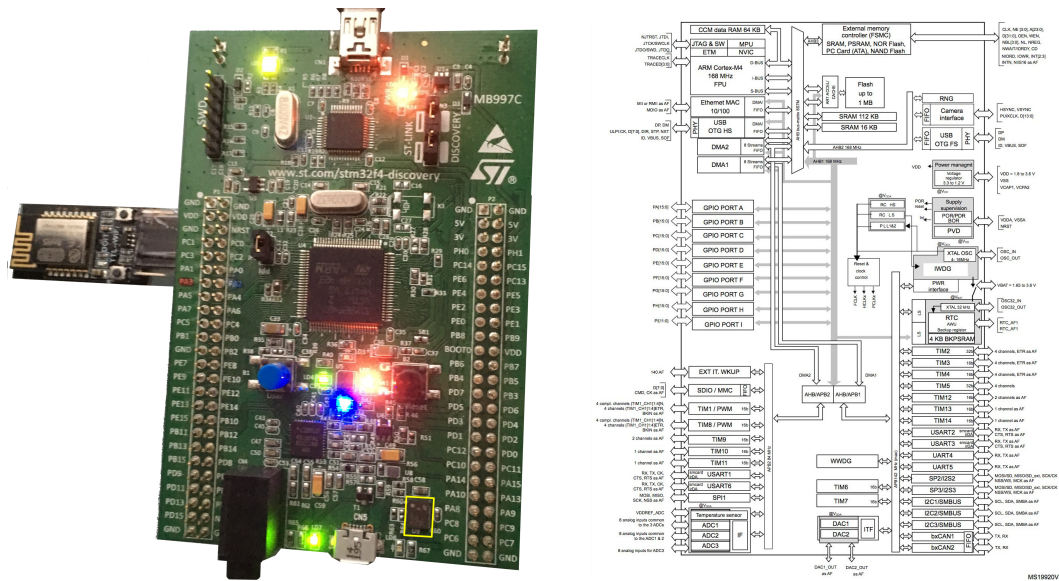


Figure 8. STM32F4Discovery board with MP45DT02 and STM32F407VG SoC overview.

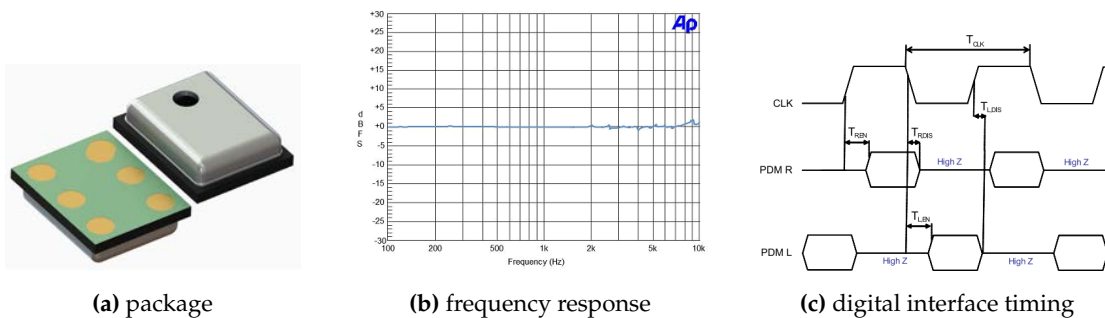
As depicted in this figure, a serial WiFi module (DT-06 Wireless WiFi) [36] was attached via an UART interface of the main board to interact remotely during the calibration process and to obtain positioning data from a smartphone in wireless mode.

4.1. MEMS Microphone

Digital MEMS audio sensors are quite inexpensive to make, apart from the obvious advantage of its tiny size. They also provide better isolation from nearby RF signals (as from a cell phone transmitter) and simplify the electronic design by reducing the required supporting circuits to just PCB traces for clock and data, power supply, and a configuration pin.

The MEMS microphone used in this system was the MP45DT02 [35]. Its main characteristics are the reduced size (HLGA package:  $4.72 \times 3.76 \times 1.25$  mm), the low power consumption, and the omnidirectional sensing. The aperture to capture sound is approximately 1 mm in diameter. If the stages most prone to noise effects are under control, well shielded, and isolated from the rest of the board system, the digital audio stream can be of higher quality (61 dB SNR,  $-26$  dBFS sensitivity, 120 dB SPL AOP).

In order to deal with the received audio signal, there is a high bit rate serial signal input (1–3.25 MHz) for two channels (timings on Figure 9c). It requires digital signal processing to transform the audio signal into the data standard for targeted audio application handled by the micro-controller, in this case for positioning purposes. The format in which peripherals deliver data is Pulse Density Modulation (PDM) [37].



(a) package (b) frequency response (c) digital interface timing  
 Figure 9. MP45DT02 MEMS audio sensor.

#### 4.2. From PDM to PCM

Creating a PDM bit stream from the analog measurement of the membrane's position is a well-understood design problem that is closely related to a class of analog-to-digital converters called sigma delta converters.

A smaller and reliable serial connection was set up in this system, since a 12- or 16-bit precision would require many parallel wires, pins, or even complicated serial protocols. STM32 microcontrollers offer Serial Peripheral Interface (SPI) blocks to communicate with external devices. Some of them also have the possibility of using the Inter-IC Sound audio protocol (I2S), a well-established serial protocol for sound data that uses a data wire, a bit clock, and a frame clock to convey pairs of PCM samples. For a microphone, the I2S has a clear disadvantage: the signal quality depends on the algorithms inside the microphone.

Most digital MEMS microphones use PDM instead of PCM. The former provides a much higher sample rate at a lower precision, which moves most of the parts of the signal processing into the receiving device and out of the microphone's package, while still providing a digital interface [38]. This has a direct effect on the sound quality.

PDM represents the diaphragm position, which is proportional to the instantaneous air pressure variation, in a binary way (one or zero). For example, a dead silent room will produce a stream of alternating ones and zeroes; a 1-kHz tone will appear as a first half of the ones and a second half of the zeroes, in repeating pattern (depending on sample rate: in this case, 2.8224 MHz).

The described MP45DT02 MEMS microphone expects to be clocked at any rate between 1 MHz and 3.25 MHz, which produces finished PCM audio samples at all the common PCM rates ranging from 16 kHz (1.024 MHz)–48 kHz (3.072 MHz) when down-sampled by a factor of 64.

Given the high data rate of the PDM bit stream, recovering a lower data rate PCM bit stream (Figure 10) is easily done by applying a suitable Low Pass Filter (LPF) with decimation, normally a factor of 64. The first stage of decimation is used to reduce the sampling frequency, followed by a High Pass Filter (HPF) to remove the signal DC offset.

Since the density of ones is proportional to the audio amplitude, the beginning can simply consist of taking a Moving average (MA) of the PDM samples. This is a basic form of a Finite Impulse Response (FIR) filter; however, it is still an average of 1-bit samples. In order to provide as much useful information as possible with each averaged sample, an average of a unique set of PDM bits is wanted. To achieve this, oversampling the PDM microphone by the same factor is needed.

So as to limit aliasing effects, the LPF is applied after the sampling process. LPFs can be implemented with an FIR filter. The LPF's coefficients must be carefully chosen so that the filtering process reduces the effects of any high frequency components in the stream. It is worth noting that the number of taps in the FIR filter can be equal to or greater than the desired decimation factor. A good reason for a greater number of taps is to improve the filter attenuation of high frequencies. More taps will be required to produce more granular signal levels in the output signal. However, a higher PDM sampling rate and more taps require more CPU cycles for processing.

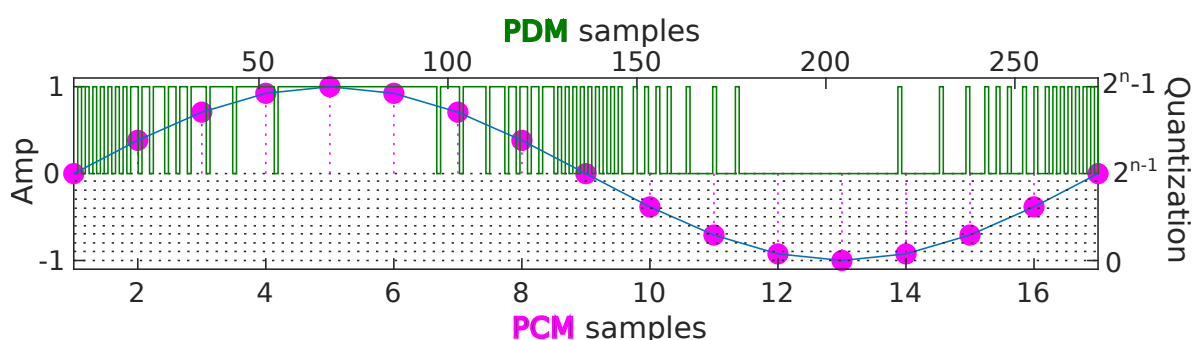


Figure 10. From analog signal to PDM and PCM processing.

ST supplies the PDM2PCM software library to convert a PDM bit stream into a PCM audio stream [39,40] with the function of decimating and filtering out the PDM stream. It has a limited sampling rate (768 kHz–2.048 MHz) and a PCM output stream with 16-bit resolution. The purpose of this library focuses on basic audio applications with lower frequencies.

Acoustic Sensor Integration

The main parts of a digital microphone are depicted in Figure 11. The Clock input (CLK) was used to control the PDM modulator. The range for the MP45DT02 mic goes from 1 MHz–3.25 MHz, and according to the desired sampling rate (48 kHz), it will be fixed to 3.072 MHz. This value has been calculated for a discrete-time representation (PDM bitstream) with an oversampling rate and a corresponding decimation factor of 64. To drive an I2S clock of such precise frequency, PLL clock resources are configured in the SoC from the 8 MHz HSE clock input to the PLL I2S clock by pre-scaling to 96 MHz and, next, by the I2S clock internal divisor to obtain the desired frequency, as shown in the bottom of Figure 12.

The MP45DT02 is designed to allow stereo audio capture. Two of the devices can operate simultaneously, sharing a bus. The microphone’s output is driven to the proper level on a selected clock edge and then goes into a high impedance state for the other half of the clock cycle (Figure 11 right). The channel select pin LR (Left-Right) defines the clock edge on which the digital microphone outputs valid data, and it is connected to GND for mono mode (“Left” channel) on the falling edge of the clock. The microphone will generate valid data for half of the clock period, then goes into a high impedance state for the other half.

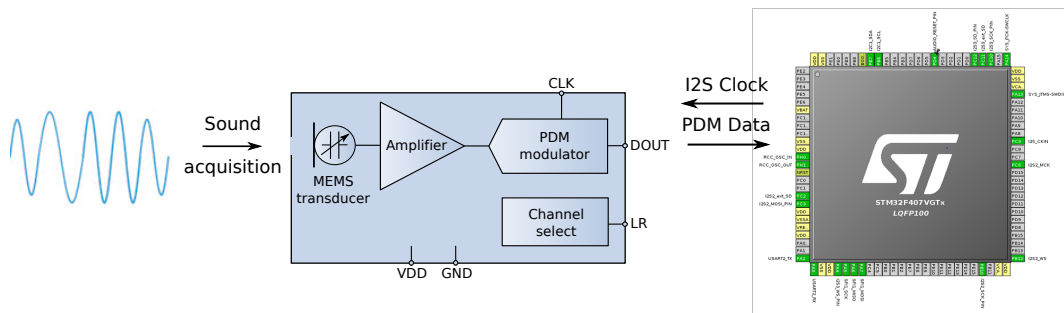


Figure 11. PDM digital microphone MP45DT02 block diagram and connection.

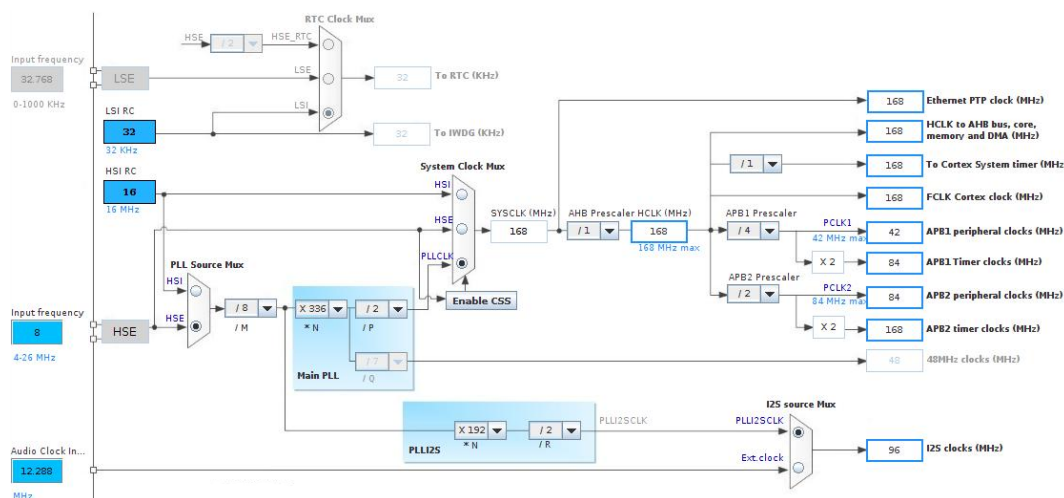
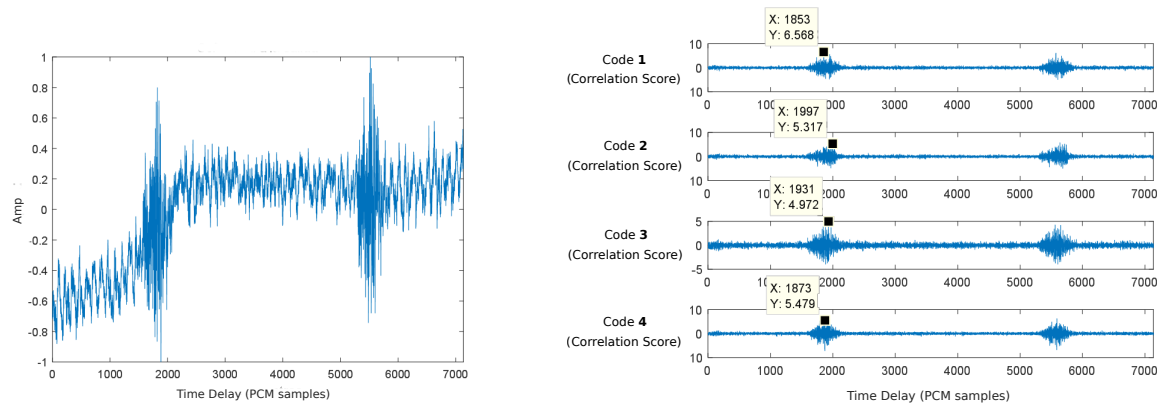


Figure 12. Clocks configuration in STM32F407VG SoC for PDM acquisition.

The samples acquired by the SPI block, in I2S half-duplex master mode, were stored into a buffer memory using a DMA channel with interrupt signaling through the receive data register configured to provide 16 contiguous bits in MSB-first (left justified) format.

### 4.3. Position Decoding

Figure 13 represents an acquired signal (left) and the relative positions extracted from the cross-correlation of this signal with the four Kasami codes assigned to the different beacons (right).



**Figure 13.** Relative positions extracted from the cross-correlation of the acquired signal with the four Kasami codes.

Following Algorithm 1, the received signal is correlated with the four Kasami codes associated with the different beacons, thus computing the DToAs that are later used by Algorithm 2 to determine the position of the receiver. This DToAs computation process is performed following an adaptive approach that started with the correlation of the sorter codes (63-bit). If the corresponding correlation peaks exceed a certain validation threshold, DToAs are computed and sent to the position determination algorithm. Otherwise, new correlations are subsequently performed with the longer 255- and 1023-bit sequences until valid correlation peaks are obtained.

**Algorithm 1:** PeaksDet

**Input:** A sampled signal stream  $S$  (length  $n_S$  samples) with modulated codes received from  $n_B$  beacons

**Input:** An array set  $K$  of  $n_B$  modulated codes (length  $n_K$  samples)

$K_{1,1}$	$\dots$	$K_{1,n_K}$	} $ n_B $
$\vdots$	$\ddots$	$\vdots$	
$K_{n_B,1}$	$\dots$	$K_{n_B,n_K}$	
} $ n_K $			

**Input:** A value  $\Delta_t$  of sensitivity for neighbor peaks

**Input:** A threshold value  $Thr$  of amplitude

**Output:** A sequence  $Pk$  of  $n_B$  sample indexes (peaks) detected

```

1  $n_C \leftarrow 2 * MAX(n_S, n_K)$ 
2  $Pk_{min} \leftarrow n_C$ 
3  $Pk_{max} \leftarrow 0$ 
4 for  $b \leftarrow 1$  to  $n_B$  do
5    $C \leftarrow \text{Correlation}(S, K[b])$ 
6    $Pk[b] \leftarrow 0$ 
7   for  $i \leftarrow 1$  to  $n_C$  do
8     if  $C[i] > C[Pk[b]]$  then
9        $Pk_1[b] \leftarrow i$ 
10    if  $|C[Pk_1[b]]| > Thr$  then
11      if  $Pk_1[b] < Pk_{min}$  then
12         $Pk_{min} \leftarrow Pk_1[b]$ 
13         $first \leftarrow b$ 
14      if  $Pk_1[b] > Pk_{max}$  then
15         $Pk_{max} \leftarrow Pk_1[b]$ 
16         $last \leftarrow b$ 
17      for  $i \leftarrow Pk_1[b] + \Delta_t$  to  $n_C - Pk_1[b] - \Delta_t$  do
18        if  $C[i] > C[Pk[b]]$  then
19           $Pk_2[b] \leftarrow i$ 
20        if  $Pk_2[b] > Pk_{max}$  then
21           $Pk_{max} \leftarrow Pk_2[b]$ 
22           $last \leftarrow b$ 
23        if  $|C[Pk_1[b]]| > Thr$  and  $|C[Pk_2[b]]| > Thr$  then
24           $Pk_1[b] \leftarrow Pk_2[b]$ 
25      else
26         $Pk_1[b] \leftarrow 0$ 
27       $Pk[b] \leftarrow Pk_1[b] - n_S$ 
28 return  $Pk$ 

```

**Algorithm 2:** Positioning

---

**Input:** A finite set  $B$  of  $n_B$  emitter beacons points  $(x_b, y_b, z_b)_{b=1}^{n_B}$   
**Input:** A finite set  $D$  of  $n_B - 1$  measured DToA receiver-emitter  
**Input:** A value  $Tol$  for tolerance condition between algorithm iterations  
**Input:** A value  $K_{max}$  as the maximum of algorithm iterations  
**Output:** The computed position  $P = (x_p, y_p, z_p)$  in the receiver

```

1  $ix \leftarrow MIN(D)$ 
2  $D_{min} \leftarrow D[ix]$ 
3 for  $i \leftarrow 1$  to  $n_B$  do
4    $D[i] \leftarrow D[i] - D_{min}$ 
5 Interchange column  $D[1] \leftrightarrow D[ix]$ 
6 Get a copy set  $B' \leftarrow B$ 
7 Interchange row  $B'[1] \leftrightarrow B'[ix]$ 
8 while true do
9   for  $i \leftarrow 1$  to  $n_B$  do
10     $d[i] \leftarrow \text{EuclideanDistance}(P, B'[i])$ 
11   for  $i \leftarrow 1$  to  $n_B - 1$  do
12     $\Delta d[i] \leftarrow D[i+1] - D[1]$ 
13    for  $j \leftarrow 1$  to 3 do
14      $A[i][j] \leftarrow \frac{B'[1][j]-P[j]}{d[1]} - \frac{B'[i+1][j]-P[j]}{d[i+1]}$ 
15      $B[i] \leftarrow D[i+1] - \Delta d[i]$ 
16   Solve equation system:  $X \leftarrow ((A^T \cdot A)^{-1} \cdot A^T) \cdot B$ 
17   if  $(|x_1| \leq Tol) \wedge (|x_2| \leq Tol) \wedge (|x_3| \leq Tol) \wedge (k < K_{max})$  then
18     break
19    $k \leftarrow k + 1$ 
20   Obtain current position  $P \leftarrow P + X$ 
21 return  $P$ 

```

---

#### 4.4. Calibration Process

The proposed procedure of self-calibration is a functionality introduced in this work with the purpose of easing the deployment of the beacons, without having to know previously their position, and taking advantage of the same digital processing of acoustic signals that are in the indicated positioning. To do this, with the same receiver, positioning measures were taken from a set of five points of known coordinates into the environment, and the combination of such measurements, with the procedure described below, and the implementation of the corresponding Algorithm 3, enabled obtaining from the its own receiver the reference coordinates of the beacons for the subsequent positioning of the same receiver within that environment.

Consider that there are  $P$  calibration points located at coordinates  $(x_p, y_p, z_p)_{p=1}^P$  and  $B$  emitters with unknown coordinates  $(x_b, y_b, z_b)_{b=1}^B$ . Let  $T_p$  be the time of flight from the source to the microphone at point  $p$ , and let  $v$  be the speed of sound in air according to measured temperature  $\vartheta$ . Then, the distance between the source and point  $p$  can be written as:

$$R_p = v_{air} \cdot T_p$$

$$v_{air} = 331.3 * \sqrt{1 + \frac{\vartheta}{273.15}}$$

and the difference in the time of flight between point  $p$  and Point 1 as:

$$\tau_p = T_p - T_1 \quad (5)$$

Multiplying Equation (5) by  $v$  yields:

$$v\tau_p = vT_p - vT_1 = R_p - R_1$$

and therefore:

$$R_p^2 = (v\tau_p + R_1)^2 = v^2\tau_p^2 + 2v\tau_p R_1 + R_1^2$$

Rearranging some terms, we can write:

$$0 = v\tau_p + 2R_1 + \frac{R_1^2 - R_p^2}{v\tau_p} \quad \text{for } p = 2, 3, \dots, P \quad (6)$$

Now, subtracting Equation (6) with  $p = 2$  from the same Equation with  $p = 3, 4, \dots, P$ , the following set of equations is obtained:

$$0 = v\tau_p - v\tau_2 + \frac{R_1^2 - R_p^2}{v\tau_p} - \frac{R_1^2 - R_2^2}{v\tau_2}, \quad \text{for } p = 3, 4, \dots, P \quad (7)$$

Then, substituting the expressions:

$$\begin{aligned} R_p &= \sqrt{(x_p - x)^2 + (y_p - y)^2 + (z_p - z)^2} \\ \Rightarrow R_p^2 &= x_p^2 - 2x_p x + x^2 + y_p^2 - 2y_p y + y^2 + z_p^2 - 2z_p z + z^2 \\ \Rightarrow R_1^2 - R_p^2 &= x_1^2 + y_1^2 + z_1^2 - x_p^2 - y_p^2 - z_p^2 - 2x_1 x - xy_1 y - 2z_1 z + 2x_p x + 2y_p y + 2z_p z \end{aligned}$$

into Equation (7) yields:

$$\begin{aligned} 0 &= v\tau_p - v\tau_2 + \frac{1}{v\tau_p} (x_1^2 + y_1^2 + z_1^2 - x_p^2 - y_p^2 - z_p^2 - 2x_1 x - xy_1 y - 2z_1 z + 2x_p x + 2y_p y + 2z_p z) \\ &\quad - \frac{1}{v\tau_2} (x_1^2 + y_1^2 + z_1^2 - x_2^2 - y_2^2 - z_2^2 - 2x_1 x - 2y_1 y - 2z_1 z + 2x_2 x + 2y_2 y + 2z_2 z) \end{aligned} \quad (8)$$

Rewriting Equation (8) more succinctly, we have:

$$0 = D_p + A_p x + B_p y + C_p z$$

where:

$$\begin{aligned} A_p &= \frac{1}{v\tau_p} (-2x_1 + 2x_p) - \frac{1}{v\tau_2} (2x_2 - 2x_1) \\ B_p &= \frac{1}{v\tau_p} (-2y_1 + 2y_p) - \frac{1}{v\tau_2} (2y_2 - 2y_1) \\ C_p &= \frac{1}{v\tau_p} (-2z_1 + 2z_p) - \frac{1}{v\tau_2} (2z_2 - 2z_1) \end{aligned}$$

and:

$$D_p = v\tau_p - v\tau_2 + \frac{1}{v\tau_p} (x_1^2 + y_1^2 + z_1^2 - x_p^2 - y_p^2 - z_p^2) - \frac{1}{v\tau_2} (x_1^2 + y_1^2 + z_1^2 - x_2^2 - y_2^2 - z_2^2) \quad (9)$$

for  $p = 3, 4, \dots, P$

The above set of  $P - 2$  expressions from Equation (9) can be written in matrix form as:

$$\begin{bmatrix} A_3 & B_3 & C_3 \\ A_4 & B_4 & C_4 \\ \vdots & \vdots & \vdots \\ A_P & B_P & C_P \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -D_3 \\ -D_4 \\ \vdots \\ -D_P \end{bmatrix}$$

Applying a pseudo-inverse method to both sides of the matrix, it is possible to solve  $x, y, z$  for one beacon. Iteratively, for the  $B$  emitters in coordinates,  $(x_b, y_b, z_b)$  for  $b = 1, 2, \dots, B$  can be obtained. Note that this algorithm yields a solution only when  $P \geq 5$ . In other words, five or more calibration points are needed. The iterative process to obtain the whole beacon set is described in Algorithm 3.

---

**Algorithm 3:** Calibration
 

---

**Input:** A finite set  $P$  of  $n_P$  calibration points  $(x_p, y_p, z_p)_{p=1}^{n_P}$

**Input:** An array set  $T$  of  $n_T = n_P \times n_E$  measured DToA receiver-emitter

$$\underbrace{\begin{bmatrix} \Delta t_{1,1} & \dots & \Delta t_{1,n_B} \\ \vdots & \ddots & \vdots \\ \Delta t_{n_P,1} & \dots & \Delta t_{n_P,n_B} \end{bmatrix}}_{|n_B|} \left. \vphantom{\begin{bmatrix} \Delta t_{1,1} & \dots & \Delta t_{1,n_B} \\ \vdots & \ddots & \vdots \\ \Delta t_{n_P,1} & \dots & \Delta t_{n_P,n_B} \end{bmatrix}} \right\} |n_P|$$

**Output:** A sequence  $E$  of  $n_E$  emitter beacon points  $(x_e, y_e, z_e)_{e=1}^{n_E}$

1 **for**  $e \leftarrow 1$  **to**  $n_E$  **do**

2     Get column vector  $T' \leftarrow T[b]$

3     Get minimum value  $T'_{min} \leftarrow MIN(T')$

4     **for**  $i \leftarrow 1$  **to**  $n_P$  **do**

5          $T'[i] \leftarrow T'[i] - T'_{min}$

6     Interchange column  $T'[1] \leftrightarrow T'[T'_{min}]$

7     Get a copy set  $P' \leftarrow P$

8     Interchange row  $P'[1] \leftrightarrow P'[T'_{min}]$

9     **for**  $p \leftarrow 3$  **to**  $n_P$  **do**

10          $\tau_p \leftarrow v * T'_p$

11          $\tau_2 \leftarrow v * T'_2$

12          $A_p = \frac{1}{\tau_p} * (-2 * P'_{1,x} + 2 * P'_{p,x}) - \frac{1}{\tau_2} * (2 * P'_{2,x} - 2 * P'_{1,x})$

13          $B_p = \frac{1}{\tau_p} * (-2 * P'_{1,y} + 2 * P'_{p,y}) - \frac{1}{\tau_2} * (2 * P'_{2,y} - 2 * P'_{1,y})$

14          $C_p = \frac{1}{\tau_p} * (-2 * P'_{1,z} + 2 * P'_{p,z}) - \frac{1}{\tau_2} * (2 * P'_{2,z} - 2 * P'_{1,z})$

15          $D_p = \tau_p - \tau_2 + \frac{1}{\tau_p} * (P'^2_{1,x} + P'^2_{1,y} + P'^2_{1,z} - P'^2_{p,x} + P'^2_{p,y} + P'^2_{p,z})$

16          $-\frac{1}{\tau_2} * (P'^2_{1,x} + P'^2_{1,y} + P'^2_{1,z} - P'^2_{2,x} + P'^2_{2,y} + P'^2_{2,z})$

17     Solve equation system:

$$\begin{bmatrix} A_3 & B_3 & C_3 \\ A_4 & B_4 & C_4 \\ \vdots & \vdots & \vdots \\ A_P & B_P & C_P \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} -D_3 \\ -D_4 \\ \vdots \\ -D_P \end{bmatrix}$$

    Obtain current beacon  $E[e] \leftarrow (x_e, y_e, z_e)$

18 **return**  $E$

---



## 5. Experimental Results

In order to analyze the accuracy of the proposed positioning system, a 3D experimental environment has been configured, where the hardware of four wireless synchronized beacons, described in Section 3, and a portable wireless receiver (Section 4) have been deployed. The position values computed from the measurements taken by the receiver were sent to a mobile terminal, wirelessly connected with the receiver, as depicted in Figure 1.

A set of five test points  $P_i$  have been defined, whose reference coordinates, as well as those of beacons  $B_j$  have been accurately measured by using a precision laser instrumentation. All these coordinates are represented with black circles in Figure 14 (left) and are also included in the first three columns of Tables 1–3.

As explained before, in the encoded signal detection procedure, the beacons emitted the longest codes (1023 bit), and the receivers performed the initial correlation with the shortest codes (63 bit). If the amplitude threshold was not reached, they would perform a new attempt with longer codes (255 bit) and then a final attempt with the longest codes (1023 bit), following an adaptive approach.

Figure 15 depicts the remote acquisition in a portable device of the positioning data computed by the acoustic receiver. This portable device was connected with the receiver via a WiFi local network, and it showed textual information corresponding to the position of the MEMS microphone included in the receiver, as well as the ambient temperature measured by the sensor included in the SoC. This ambient temperature was used to estimate the speed of sound in air and, from it, the DToAs of the detected signals.

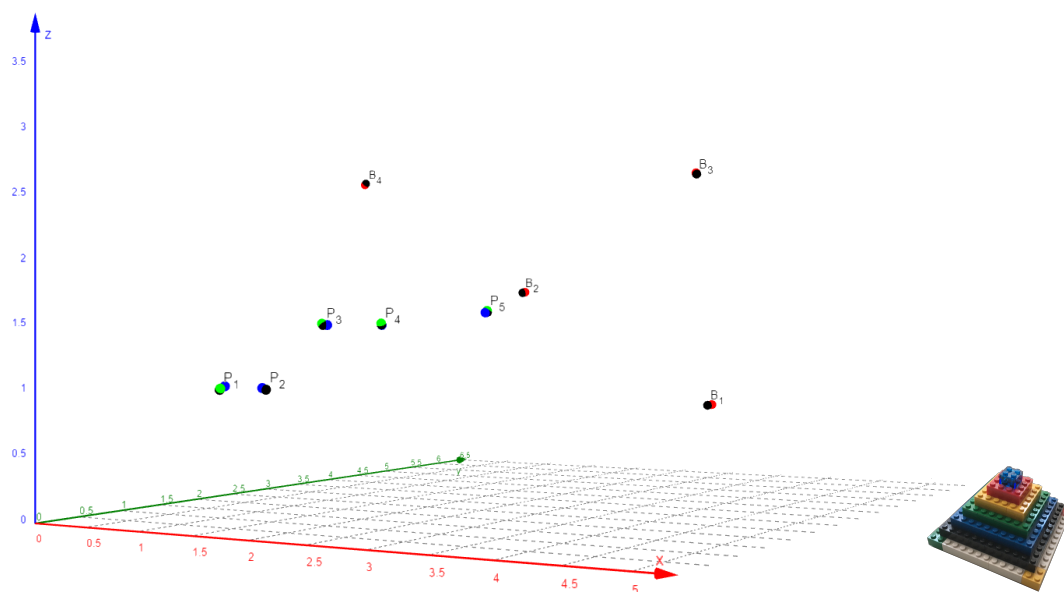


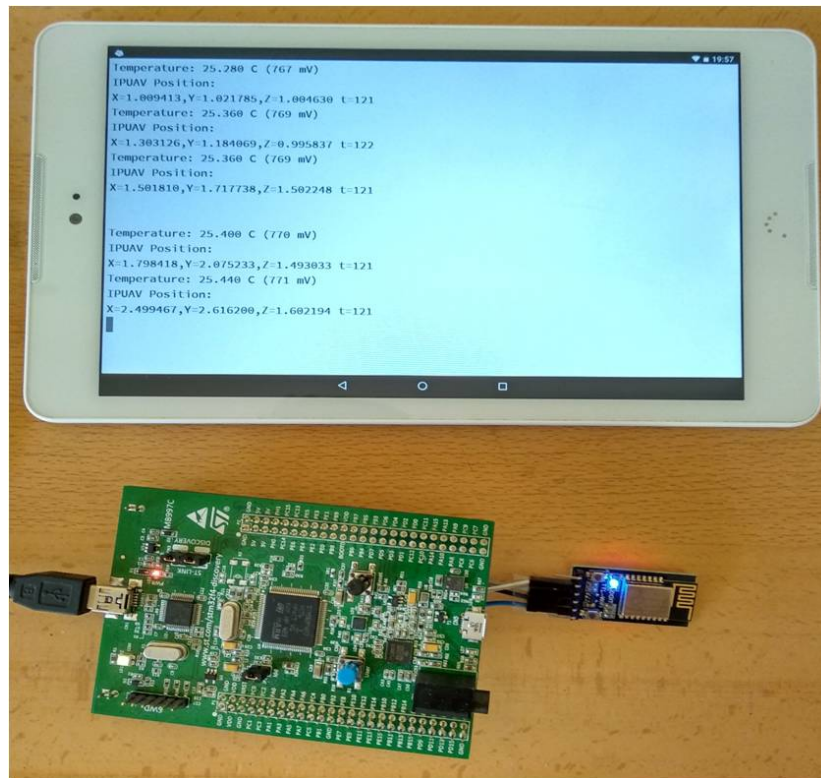
Figure 14. Experimental measures and calibration pyramid Lego.

The first computed coordinates for the test points  $P_i$  (green circles) were estimated from the ground-truth beacons coordinates introduced directly in the receiver. These values and derived statistics are shown in Table 1.

The following experimental coordinates calculated in the receiver are those estimated from the beacons calibration procedure. The Lego pyramid was placed with its blue vertex in  $P_1$  and in parallel to the coordinates axes established in the room. Then, measurements were taken with the receiver from the rest of the vertices with a pre-established sequence and with the help of a push-button on the receiver board. These computed  $B_j$  position coordinates (red circles), returned to the mobile terminal by the receiver, are shown in Table 2 together with their derived statistics.

Finally, new experimental measurements were taken based on those previously acquired in the calibration phase, already updated in the receiver, for the same  $P_i$  test points (blue circles). The derived

results are shown in Table 3 and serve to conduct a comparison of errors that is also reflected in the conclusions of Section 6.



**Figure 15.** Picture of the acoustic receiver and the portable device where the positioning data are remotely acquired and shown.

**Table 1.** Experimental measurements of test positions. PRE, Percent Relative Error.

$P_i$	Coordinates (m)			Measures (m)			Absolute Error (m)			PRE	MSE (m)
	Axis x	Axis y	Axis z	Axis x	Axis y	Axis z	Axis x	Axis y	Axis z		
$P_1$	1	1	1	1.0101	0.9931	1.0098	−0.0101	0.0069	−0.0098	2.67%	0.0157
$P_2$	1.3	1.2	1	1.2907	1.2092	1.0007	0.0093	−0.0092	−0.0007	1.55%	0.0131
$P_3$	1.5	1.7	1.5	1.4909	1.6988	1.5107	0.0091	0.0012	−0.0107	1.39%	0.0141
$P_4$	1.8	2.1	1.5	1.8007	2.0949	1.5096	−0.0007	0.0051	−0.0096	0.92%	0.0109
$P_5$	2.5	2.6	1.6	2.4994	2.6031	1.6107	0.0006	−0.0031	−0.0107	0.81%	0.0112

In all results' tables, the errors in the three coordinates are shown in the fourth column group, and the 3D positioning Percent Relative Error (PRE) and Mean Squared Error (MSE) are shown in the two last ones. It is worth mentioning that the errors made in the beacons calibration process (Table 2) are slightly below those made in the test points positioning (Table 1), a fact that can be attributed to the use of five reference calibration points in the latter case, with respect to the four used in the estimation of the test points' positions.

**Table 2.** Experimental measurements in the calibration process.

$B_j$	Coordinates (m)			Measures (m)			Absolute Error (m)			PRE	MSE (m)
	Axis x	Axis y	Axis z	Axis x	Axis y	Axis z	Axis x	Axis y	Axis z		
$B_1$	3.376	5.109	0.743	3.3998	5.1509	0.7485	0.0238	0.0419	0.0055	2.25%	0.0162
$B_2$	1.785	4.553	1.755	1.8004	4.5702	1.7600	0.0154	0.0172	0.0050	1.52%	0.0079
$B_3$	3.478	4.619	2.795	3.4671	4.6316	2.8058	−0.0109	0.0126	0.0108	0.97%	0.0066
$B_4$	0.055	4.581	2.803	0.0444	4.5816	2.7917	−0.0106	0.0006	−0.0113	24.34%	0.0052

The final Table 3 shows the measurement errors obtained during normal operation of the receiver, once the initial calibration process has been conducted in the currently-covered 3D environment. Note that these errors are greater than those appearing in the previous tables, mainly due to an accumulation of them, although their value is still around several centimeters, a result that can be considered acceptable for the applications for which the system is intended.

**Table 3.** Experimental measures of positions with calibrated beacons.

$P_i$	Coordinates (m)			Measures (m)			Absolute Error (m)			PRE	MSE (m)
	Axis x	Axis y	Axis z	Axis x	Axis y	Axis z	Axis x	Axis y	Axis z		
$P_1$	1	1	1	1.0022	1.0639	1.0254	0.0022	0.0639	0.0254	8.70%	0.0688
$P_2$	1.3	1.2	1	1.2645	1.1986	1.0121	−0.0355	−0.0014	0.0121	4.12%	0.0375
$P_3$	1.5	1.7	1.5	1.5237	1.7268	1.4993	0.0237	0.0268	−0.0007	3.15%	0.0358
$P_4$	1.8	2.1	1.5	1.7487	2.1876	1.4939	−0.0513	0.0876	−0.0061	7.35%	0.1017
$P_5$	2.5	2.6	1.6	2.4964	2.5784	1.5962	−0.0036	−0.0216	−0.0038	1.22%	0.0222

## 6. Conclusions

This work has presented the hardware/software co-designed infrastructure of an acoustic local positioning system, based on a set of wireless synchronized active beacons that cover an indoor environment. These beacons emit spread-spectrum signals (BPSK modulated Kasami codes) to a set of mobile receivers with computing capabilities to first estimate the beacons' position by means of an auto-calibration process and, then, compute its own position by following a 3D multilateration procedure.

Both types of system nodes are based on integrated SoC platforms, with omni-directional digital MEMS microphones in the case of the receivers and advanced audio DACs and RF communication modules coupled. Furthermore, the implementation of algorithms and digital signal processing methods, optimized for high performance and low-power consumption, have been presented. A set of experimental results has been conducted to evaluate the accuracy of the proposed system, obtaining 3D-positioning mean squared errors between 2.2 and 10.2 cm when estimating the receiver's location from the beacons' calibrated coordinates.

As near future improvements, we intend to combine the inertial MEMS sensors available in the receiver platform, namely the gyroscope and the accelerometer, in order to complement the position estimation algorithm; as well as to initiate the transition of the receiver's implementation to a low consumption platform, with similar performance to that used for the emitters.

**Author Contributions:** Conceptualization, J.A.M. and F.J.A.; formal analysis, J.A.M. and F.J.A.; methodology, J.A.M. and T.A.; software, J.A.M. and T.A.; validation, J.A.M. and T.A.; writing, original draft preparation, J.A.M., F.J.A., and J.A.P.

**Funding:** This work was funded in part by the European Commission through the Project LIMUS under Grant 0246-LIMUS-4-E, in part by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund (ERDF) through the Project TARSIVUS under Grant TIN2015-71564-C4-4-R, and in part by the Regional Government of Extremadura under Grant GR18038.

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

## Abbreviations

The following abbreviations are used in this manuscript:

ALPS	Acoustic Local Positioning Systems
SoC	System on a Chip
RF	Radio Frequency
RT	Real-Time
BSP	Board Support Package
MEMS	MicroElectroMechanical Systems

HW	Hardware
SW	Software
FW	Firmware
TOA	Time of Arrival
TDOA	Time Difference of Arrival
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
FSK	Frequency Shift Keying modulation
OOK	On-Off Keying modulation
RSS	Received Signal Strength
SNR	Signal-to-Noise Ratio
PN	Pseudo-Noise code
dBFS	decibel Full Scale
dB SPL	decibel Sound Pressure Level
AOP	Acoustic Overload Point
ARM	Advanced RISC Machine, ARM Holdings
STM32	STMicroelectronics 32-bit
DSP	Digital Signal Processing
FPU	Floating Point Unit
ART	Adaptive Real-Time accelerator
PDM	Pulse-Density Modulation
PCM	Pulse-Code Modulation
OSR	Over-Sampling Rate
HLGA	Holed Land Grid Array device package
LSB	Least Significant (right-most) Bit
MSB	Most Significant (left-most) Bit
LPF	Low-Pass Filter
HPF	High-Pass Filter
BPF	Band-Pass Filter
FIR	Finite Impulse Response filter
MA	Moving Average
SPI	Serial Peripheral Interface
IC	Integrated Circuit
I2S	Inter-IC Sound audio protocol
PLL	Phase-Locked Loop
SAI	Serial Audio Interface
UAV	Unmanned Aerial Vehicle

## References

1. Misra, P.; Enge, P. *Global Positioning System: Signals, Measurements and Performance*; Ganga-Jamuna Press: Lincoln, MA, USA, 2010.
2. Mautz, R. *Indoor Positioning Technologies*; Institute of Geodesy and Photogrammetry, Department of Civil, Environmental and Geomatic Engineering, ETH Zurich: Zurich, Switzerland, 2012.
3. Brena, R.F.; García-Vázquez, J.P.; Galván-Tejada, C.E.; Muñoz-Rodríguez, D.; Vargas-Rosales, C.; Fangmeyer, J. Evolution of Indoor Positioning Technologies: A Survey. *J. Sens.* **2017**, *2017*, 2630413. [[CrossRef](#)]
4. Duong, S.N.; Trinh, A.V.; Dinh, T. Bluetooth Low Energy Based Indoor Positioning on iOS Platform. In Proceedings of the 2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc), Hanoi, Vietnam, 12–14 September 2018; pp. 57–63.
5. Salti, T.E.; Orlando, M.; Hood, S.; Knelsen, G.; Iarocci, M.; Lazzara, Z.; Xie, Y.; Cheung, J.C.; Woungang, I. A New Set of Bluetooth-Based Fingerprinting Algorithms for Indoor Location Services. In Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 1–3 November 2018; pp. 1–7.
6. Sun, W.; Xue, M.; Yu, H.; Tang, H.; Lin, A. Augmentation of Fingerprints for Indoor WiFi Localization Based on Gaussian Process Regression. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10896–10905. [[CrossRef](#)]

7. Caso, G.; Le, M.T.P.; De Nardis, L.; Di Benedetto, M.G. Performance Comparison of WiFi and UWB Fingerprinting Indoor Positioning Systems. *Technologies* **2018**, *6*, 14. [[CrossRef](#)]
8. Shi, L.; Zhao, Y.; Liu, G.; Chen, S.; Wang, Y.; Shi, Y. A Robust Pedestrian Dead Reckoning System Using Low-Cost Magnetic and Inertial Sensors. *IEEE Trans. Instrum. Meas.* **2018**, 1–8. [[CrossRef](#)]
9. Wu, Y.; Zhu, H.; Du, Q.; Tang, S. A Pedestrian Dead-reckoning System for Walking and Marking Time Mixed Movement Using a SHSs Scheme and a Foot-mounted IMU. *IEEE Sens. J.* **2018**, *19*, 1661–1671. [[CrossRef](#)]
10. Álvarez, F.J. Fundamentals of Airborne Acoustic Positioning Systems. In *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*; Conesa, J., Pérez, A., Torres, J., Montoliu, R., Eds.; Elsevier AP: Cambridge, MA, USA, 2018; Chapter 17, pp. 335–352.
11. Ureña, J.; Hernández, Á.; García, J.J.; Villadangos, J.M.; Pérez, M.C.; Gualda, D.; Álvarez, F.J.; Aguilera, T. Acoustic Local Positioning With Encoded Emission Beacons. *Proc. IEEE* **2018**, *106*, 1042–1062. [[CrossRef](#)]
12. Gualda, D.; Ureña, J.; García, J.C.; García, E.; Alcalá, J. Simultaneous calibration and navigation (SCAN) of multiple ultrasonic local positioning systems. *Inform. Fusion* **2019**, *45*, 53–65. [[CrossRef](#)]
13. Duff, P.; Muller, H. Autocalibration algorithm for ultrasonic location systems. In Proceedings of the Seventh IEEE International Symposium on Wearable Computers, White Plains, NY, USA, 21–23 October 2003; pp. 62–68.
14. Runge, A.; Baunach, M.; Kolla, R. Precise self-calibration of ultrasound based indoor localization systems. In Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation, Guimarães, Portugal, 21–23 September 2011; pp. 1–8.
15. Walter, C.; Syafrudin, M.; Schweinzer, H. A Self-Contained and Self-Checking LPS with High Accuracy. *ISPRS Int. J. Geo-Inform.* **2013**, *2*, 908–934. [[CrossRef](#)]
16. Teich, J. Hardware/Software Codesign: The Past, the Present, and Predicting the Future. *Proc. IEEE* **2012**, *100*, 1411–1430. [[CrossRef](#)]
17. ARM Ltd. ARM Website. 2018. Available online: <http://www.arm.com> (accessed on 20 November 2018).
18. ARM. ARM Cortex-M4 Technical Reference Manual. 2015. Available online: [https://static.docs.arm.com/100166\\_0001/00/arm\\_cortexm4\\_processor\\_trm\\_100166\\_0001\\_00\\_en.pdf](https://static.docs.arm.com/100166_0001/00/arm_cortexm4_processor_trm_100166_0001_00_en.pdf) (accessed on 20 November 2018).
19. STMicroelectronics. STM32L4 Series of Ultra-Low-Power MCUs. 2017. Available online: <http://www.st.com/en/microcontrollers/stm32l4-series.html> (accessed on 20 November 2018).
20. STMicroelectronics. ST Website. 2018. Available online: <http://www.st.com> (accessed on 20 November 2018).
21. STMicroelectronics. STM32L476VG Ultra-Low-Power with FPU ARM Cortex-M4 MCU 80 MHz with 1 Mbyte Flash, LCD, USB OTG, DFSDM. 2017. Available online: <http://www.st.com/en/microcontrollers/stm32l476vg.html> (accessed on 20 November 2018).
22. STMicroelectronics. 32L476GDISCOVERY Discovery Kit with STM32L476VG MCU. 2016. Available online: <http://www.st.com/en/evaluation-tools/32l476gdiscovery.html> (accessed on 20 November 2018).
23. Microchip Technology, Inc. MRF89XAM8A 868 MHz Radio Transceiver. 2011. Available online: <http://www.microchip.com/wwwproducts/en/MRF89XAM8A> (accessed on 20 November 2018).
24. Microchip Technology, Inc. Microchip Website. 2016. Available online: <http://www.microchip.com> (accessed on 20 November 2018).
25. Microchip Technology, Inc. MRF89XA Datasheet DS70622C—Ultra Low-Power, Integrated ISM Band Sub-GHz Transceiver. 2011. Available online: <http://www.microchip.com/datasheet/MRF89XA> (accessed on 20 November 2018).
26. Paredes, J.A.; Aguilera, T.; Álvarez, F.J.; Lozano, J.; Morera, J. Analysis of Doppler Effect on the Pulse Compression of Different Codes Emitted by an Ultrasonic LPS. *Sensors* **2011**, *11*, 10765–10784. [[CrossRef](#)] [[PubMed](#)]
27. Kasami, T. *Weigh Distribution Formula for Some Class of Cyclic Codes*; Technical Report R-285; University of Illinois—Urbana: Champaign, IL, USA, 1968.
28. Philips Semiconductors, Ltd. I2S Bus Specification and User Manual UM10204. 1996. Available online: [http://www.nxp.com/acrobat\\_download/various/I2SBUS.pdf](http://www.nxp.com/acrobat_download/various/I2SBUS.pdf) (accessed on 20 November 2018).
29. Intel Corporation, Inc. Audio Codec'97 Audio Component Specification. Revision 2.3. 2002. Available online: [http://download.intel.com/support/motherboards/desktop/sb/ac97\\_r23.pdf](http://download.intel.com/support/motherboards/desktop/sb/ac97_r23.pdf), 2002 (accessed on 1 December 2018).

30. Cirrus Logic, Inc. CS43L22—Portable Audio DAC with Integrated Class D Speaker Driver. 2010. Available online: <https://www.cirrus.com/products/cs43l22/> (accessed on 20 November 2018).
31. STMicroelectronics. STM32F4DISCOVERY Discovery Kit with STM32F407VG MCU. 2016. Available online: <http://www.st.com/en/evaluation-tools/stm32f4discovery.html> (accessed on 20 November 2018).
32. STMicroelectronics. Discovery Kit with STM32F407VG MCU. 2017. Available online: [https://www.st.com/resource/en/user\\_manual/dm00039084.pdf](https://www.st.com/resource/en/user_manual/dm00039084.pdf) (accessed on 20 November 2018).
33. STMicroelectronics. STM32F407VG High-Performance Foundation Line, ARM Cortex-M4 Core with DSP and FPU, 1 Mbyte Flash, 168 MHz CPU, ART Accelerator, Ethernet, FSMC. 2017. Available online: <http://www.st.com/en/microcontrollers/stm32f407vg.html> (accessed on 20 November 2018).
34. STMicroelectronics. STM32F4 Series of High-Performance MCUs with DSP and FPU Instructions. 2017. Available online: <http://www.st.com/en/microcontrollers/stm32f4-series.html> (accessed on 20 November 2018).
35. STMicroelectronics. MP45DT02 MEMS Audio Sensor Omnidirectional Digital Microphone. 2016. Available online: <http://www.alldatasheet.com/datasheet-pdf/pdf/519456/STMICROELECTRONICS/MP45DT02.html> (accessed on 20 November 2018).
36. Shenzhen Doctors of Intelligence & Technology Co., Ltd. DT-06 Wireless WiFi Serial Port Transparent Transmission Module. 2017. Available online: <http://ttl.doit.am> (accessed on 20 November 2018).
37. STMicroelectronics. Interfacing PDM Digital Microphones Using STM32 32-bit Arm<sup>®</sup> Cortex<sup>®</sup> MCUs—Application Note AN5027. 2017. Available online: [http://www.st.com/resource/en/application\\_note/DM00380469.pdf](http://www.st.com/resource/en/application_note/DM00380469.pdf) (accessed on 20 November 2018).
38. Kite, T. *Understanding PDM Digital Audio*; Audio Precision: Beaverton, OR, USA, 2012.
39. STMicroelectronics. STM32Cube PDM2PCM Software Library for the STM32F4/F7/H7 Series. 2018. Available online: [https://www.st.com/resource/en/user\\_manual/dm00482421.pdf](https://www.st.com/resource/en/user_manual/dm00482421.pdf) (accessed on 20 November 2018).
40. STMicroelectronics. PDM Audio Software Decoding on STM32 Microcontrollers Application Note AN3998. 2011. Available online: [http://www.st.com/resource/en/application\\_note/dm00040808.pdf](http://www.st.com/resource/en/application_note/dm00040808.pdf) (accessed on 20 November 2018).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).