Distributed Full Synchronized System for Global Health Monitoring Based on FLSA

Giuseppe Coviello, *Senior Member, IEEE*, Antonello Florio, *Student Member, IEEE*, Gianfranco Avitabile, *Senior Member, IEEE*, Claudio Talarico, *Senior Member, IEEE*, and Janet M. Wang-Roveda

***Abstract*—In modern medicine, smart wireless connected devices are gaining an increasingly important role in aiding doctors' job of monitoring patients. More and more complex systems, with a high density of sensors capable of monitoring many biological signals, are arising. Merging the data offers a great opportunity for increasing the reliability of diagnosis. However, a huge problem is constituted by synchronization. Multi-board wireless-connected monitoring systems are a typical example of distributed systems and synchronization has always been a challenging issue. In this paper, we present a distributed full synchronized system for monitoring patients' health capable of heartbeat rate, oxygen saturation, gait and posture analysis, and muscle activity measurements. The time synchronization is guaranteed thanks to the Fractional Low-power Synchronization Algorithm (FLSA).**

***Index Terms*—Clock synchronization, internet of medical things, time measurement, time dissemination, wireless body area network, wireless sensor networks.**

I. Introduction

TIME synchronization (usually simply referred to as "synchronization") in computer networks and distributed systems theory has always gathered attention from the scientific community. As summarized by *Freris et al.* [1], there are typically three levels of specifications for time synchronization. In the first level, it is only important the event order. The goal is to create the right chronology for the events in the network, while the exact knowledge of each time instant is not important. In the second level, it is important to achieve a relative synchronization in the network, that is, estimating and compensating the relative clock drift among different clocks; the nodes do not need to agree on a common time reference, but they have enough information to transpose each clock value to their own time reference. In the third and strongest level the goal is to achieve absolute synchronization. Each node owns a local clock value and the synchronization aims to have each clock of the network agree on a common time. It is evident that the a higher level of synchronization implies the characteristics of lower ones. Also, as we consider upper levels, improved accuracy is reached for the entire synchronization process, determining an increase of the computational load on the network facilities. The choice of the level is strongly influenced by the type of application, as this latter introduces some constraints on the allowable complexity of the synchronization protocol, and requires trading-off complexity and accuracy.

Over the years, due to the introduction of new technology paradigms like the Internet of Things (IoT), the meaning of the term 'network' has significantly changed. In the IoT, a huge density of smart and, usually, wireless-connected devices join networks with different extensions. In this paper, we will focus on Wireless Body Area Networks (WBANs), i.e. networks with a small geographical extension, based on a mesh or star topology [2]. Each node of a WBAN is usually equipped with sensors and powered by a battery, to enhance its portability. In this context, IoT is often employed for supporting medicine giving rise to the Internet of Medical Things (IoMT).

In IoMT synchronization is crucial, since data coming from different sensors placed or implanted along the body are merged to obtain the final diagnosis. However, different biological parameters vary with different rates. Hence, we can introduce two possible levels of synchronization, depending on the level of required accuracy. In weak network time synchronization (WTS), we allow for a lower degree of accuracy, thus relaxing constraints on the synchronization process complexity. WTS is suitable for slowly varying phenomena such as body and environment temperature variations. In strong network time synchronization (STS), the degree of accuracy must be kept as high as possible since even small mismatches on time-stamping can lead to huge errors in the data merging phase. This is the case of surface electromyography (sEMG) correlated to Inertial Measurement Units (IMUs) for gait analysis. Nevertheless, a higher degree of synchronization calls for higher complexity for the potential increased number of exchanged messages.

Manuscript received 15 February 2022; revised 27 April 2022; accepted 5 May 2022. Date of publication 10 May 2022; date of current version 10 October 2022. The work of Giuseppe Coviello was supported by the Italian Ministry of Education, University, and Research through the European National Operational Program (PON) Project Attraction and International Mobility (AIM), activity 1, line 1 under Grant AIM1892824. This paper was recommended by Associate Editor Chetan Thakur. *(Corresponding author: Giuseppe Coviello.)*

Giuseppe Coviello, Antonello Florio, and Gianfranco Avitabile are with the Department of Electrical and Information Engineering, Polytechnic University of Bari, 70125 Bari, Italy (e-mail: giuseppe.coviello@poliba.it; antonello.florio@poliba.it; gianfranco.avitabile@poliba.it).

Claudio Talarico is with the Department of Electrical and Computer Engineering, Gonzaga University, Spokane, WA 99258 USA (e-mail: talarico@gonzaga.edu).

Janet M. Wang-Roveda is with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721 USA (e-mail: meilingw@arizona.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TBCAS.2022.3173586.

Digital Object Identifier 10.1109/TBCAS.2022.3173586

Another difference when comparing IoMT devices is between online and offline data acquisition for post-processing. In particular, we refer to online processing when a device forwards the sensor measurement to the gateway and/or the outer network as soon as the data is consolidated. In the offline processing, the data is stored in a persistent memory with which the device is equipped, and only after a predefined time interval, the acquisitions are sent to the gateway. Online data processing calls for a stable network infrastructure that is usually feasible to be realized in controlled environments such as medical laboratories. For most medical protocols, it is important for the patients to behave naturally, as when no measurement is in progress. In [3] *Renggli et al.* provide data highlighting a huge difference between offline and online data processing for what concerns the reliability of the measurements for diagnosis purposes. In particular, a long off-laboratory off-line processing is more reliable than an in-lab online measurement.

In this paper, we introduce an absolute low-cost synchronized platform for off-line acquisition of heartbeat rate (HR), peripheral oxygen saturation (SpO_2), gait analysis, posture analysis, and muscle activity related to gait analysis. The system is arranged in a star topology with a master/slave (M/S) architecture. The network time synchronization is achieved thanks to the algorithm that the authors presented in [4] and [5].

A. *Related Works*

In literature, it is possible to find many examples of synchronized IoMT systems. *Mo et al.* in [6] propose a synchronized system for monitoring the spent energy and breath volume. The system comprises three different units, arranged in an M/S architecture. Two units are equipped with an accelerometer (placed, respectively, on the wrist and on the hip) and another unit uses a ventilator sensor (placed on the abdomen). The synchronization problem arises because, along with the clock drift problems, depending on the configuration, packets can be long enough to determine synchronization artifacts. The proposed approach is a slot-data synchronization method, performed at the master level. The master is in charge of the network synchronization, and it reorders the packets in the correct time order taking into account the reception time, the length of the packet, and the sampling time of the sensors.

In [7], *Vedaei et al.* propose a WTS system based on Raspberry Pi Zero for monitoring the HR, the SpO_2, and temperature for COrona VIrus Disease 2019 (COVID-19) diagnosis. The system transmitted the data acquired from the sensors to a smartphone and, then, to a cloud infrastructure for further processing. In this case, STS is not required in the system since hearth rate, SPO_2 and temperature are slowly varying parameters. Thus, in this case, a large degree of error on the synchronization is accepted.

The solution proposed in [8] allows for both WTS and STS. In this work, *Pathinarupothi et al.* propose an IoT system for global health monitoring based on edge computing. The system uses an M/S hierarchy when STS is required, for example in the case when it is used for symptomatic patients. The master is chosen by the gateway, connecting the BAN with the outer networks, and it is in charge of guaranteeing the network synchronization. In the case of WTS mode, a peer-to-peer (P2P) structure is adopted, with the gateway being the network coordinator, loosely constraining the time synchronization.

The work of *Wang et al.* [9] introduces a plantar pressure acquisition system for diabetic foot prevention and diagnosis. The system is composed of two special shoes with pressure sensors connected to a wireless Data Acquisition (DAQ) board, one for each leg. Each DAQ is synchronized to the other and it is in charge of querying the connected sensors, collecting data, and exposing them to the outer network through the Bluetooth connection of a smartphone. In this domain, STS is mandatory, since it is a constraint of the gait analysis [10]. However, no information on the employed synchronization protocol is available.

B. *Main Contributions and Paper Organization*

We summarize here the most relevant contributions of our work.

- We introduce a low-cost synchronized platform for offline patients' long-term global health monitoring. The system is low-power and relies on custom-designed boards based on the nRF52832 SoC [11], specific sensors for measuring the HR, SpO_2, gait behavior, posture, and muscle activity related to the gait through sEMG. The employed network topology is star-like with one predefined master and several slaves.
- We discuss FLSA, the employed synchronization algorithm implementation, briefly recalling the theoretical background introduced in [4].
- We highlight and propose a solution to the mutual synchronization problem between master and slaves. This problem arises when all the system units expose sensors for capturing data that are involved in the data merging process.
- We implement the FLSA in the monitoring system. Thus, we present the preliminary results in terms of achieved synchronization accuracy after one-month-long experiment in the laboratory and the ones obtained by testing the entire monitoring system on a healthy subject in a one-day test in everyday life.

The remaining part of the paper is organized as follows. In Section II we recall the theory behind the synchronization algorithm proposed by the authors, highlighting its limits as it does not allow for a mutual synchronization between master and slaves, in the way it was deployed. Hence, we introduce a supplementary procedure to handle also this situation. In Section III we describe in detail the low-cost system for health monitoring. Section IV describes an experimental campaign validating the system's operations. Conclusions close this work.

II. Fractional Low-Power Synchronization Algorithm

The authors proposed FLSA [4], [5] as a possible solution for the low-power multi-board time synchronization problem in medical applications. The timer correction is performed totally at hardware level, thus making the approach flexible and transparent to the upper layers of the system. In this section, we

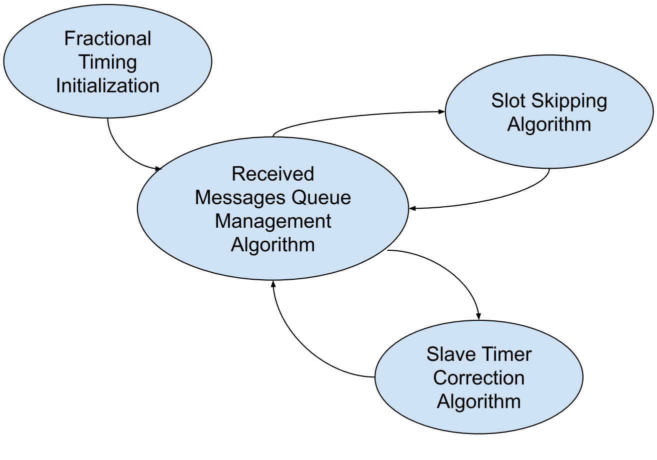

Fractional Timing Initialization
Slot Skipping Algorithm
Received Messages Queue Management Algorithm
Slave Timer Correction Algorithm

Fig. 1. FLSA routines interaction diagram [4].

briefly recall its formulation and results, along with some new, recently introduced improvements.

A. A Fractional Approach to Timer Correction

FLSA performs synchronization thanks to four sub-algorithms, which are summarized in Fig. 1 along with their interactions. FLSA makes use of two $N/N+1$ type comparators/counters $\{\mathrm{A}, \mathrm{B}\}$ to perform the dual modulus division. The number of ticks, $\mathrm{N_M}$, needed to form a synchronization slot of length $\mathrm{T_M}$ is:

$$\mathrm{N_M} = \left\lfloor \frac{\mathrm{T_M}}{\mathrm{t_{RTC}}} \right\rceil \tag{1}$$

where $\lfloor\cdot\rceil$ represents the rounding at the closest integer number operator, and t_{RTC} is the granularity of the Real-Time Clock (RTC). Directly from the RTC it is possible to extract the System Timer (ST) granularity t_{ST} with the formula [4]

$$\begin{cases} \mathrm{R_C^A} = \left\lfloor \dfrac{t_{\mathrm{ST}}}{t_{\mathrm{RTC}}} \right\rceil \\ \mathrm{R_C^B} = \mathrm{R_C^A} + 1 \end{cases} \tag{2}$$

with R_C^i being the number of ticks needed to form a t_{ST} for comparator i. The synchronization slot is determined by the variation of the comparator values, according to

$$\mathrm{N_M} = \mathrm{R_C^A} \cdot \mathrm{I_A} + \mathrm{R_C^B} \cdot \mathrm{I_B} \tag{3}$$

with I_i being the number of times comparator i must be used.

Every $\mathrm{T_M}$ seconds the master broadcasts to the slaves a synchronization packet containing its ST. The synchronization packet is processed with the highest priority with respect to every other possible type of message thanks to the Received Messages Queue Management (RMQM) Routine, and this is true for master and slaves. After the synchronization message has been received, each slave first evaluates the equivalent accumulated time difference in a single slot, $\mathrm{T_M}$. Then, the slave uses this value in the evaluation of the number of the time interval in which keep the radio off according to the slot-skipping routine.

When a slave is said to be synchronous it can switch-off its radio section. The slot-skipping routine operates in terms of synchronization slots. First of all, the number of slots in which the node j can be considered to be synchronous to the master, $\mathrm{S_{sync}}(j)$, is updated, incrementing by one unit the previously stored value. We assume the slots in which the radio section is turned off to be synchronous, by hypothesis. The radio is switched off when the number of synchronous slots exceeds a predefined threshold $S_{\mathrm{sync}}^{\mathrm{min}}$. The ST is adjusted at the end of each synchronization slot, taking into account the weighted mean value of the delays. This value represents the average error for each synchronization slot and it is subtracted in each skipped synchronization interval. After that, the skipped slot count is updated. The routine ends with the radio being switched on for receiving a new synchronization packet, at the reception of which, the RMQM routine is executed. The skipped slot number linearly increases when the systems are synchronous.

The Slave Timer correction Routine is governed by the set of equations:

$$\begin{cases} \sum_{i=A}^{B} \Delta \mathrm{R_C^i} \, \Delta \mathrm{I_i} = \mathrm{N_M^{new}} \\ \sum_{i=A}^{B} \Delta \mathrm{I_i} = \left\lfloor \frac{\mathrm{T_M}}{t_{\mathrm{ST}}} \right\rceil \\ \{\mathrm{I}_i, \mathrm{R}_\mathrm{C}^i\} > 0 \; \forall i \end{cases} \tag{4}$$

being R_C^i the updated value of comparator i, $\Delta\mathrm{I_i}$ the updated value of its weight and $\mathrm{N_M^{new}}$ the compensated time slot duration. The system of equations in (4) does not have a single solution. Therefore, a solution can be found imposing the constraint $\delta\mathrm{R_C^A} = \delta\mathrm{R_C^B} = 0$ and start an iterative procedure. The simplified equation system is:

$$\begin{cases} \delta\mathrm{I_B} = \Delta\mathrm{N_M} \\ \delta\mathrm{I_A} = -\Delta\mathrm{N_M} \end{cases} \tag{5}$$

The new parameters to be set are available only when the system has a solution. If that is not the case, the sign of $\Delta\mathrm{N_M}$ indicates the advance or delay of the slave with respect to the master, and so new values of $\mathrm{I_A}$ and $\mathrm{I_B}$ are chosen with the initialization step being executed. It is very important to highlight that the method detects the t_{ST} minimum deviation while the correction precision is equal to t_{RTC}, that is, the minimum allowed value determined by the specific hardware implementation.

The preliminary measurement campaign was 7 days long and it employed a time periodicity for the synchronization $\mathrm{T_M} = 20$ s. The experiment showed that in the best case for about 22 hours the time mismatch was lower than 500 μs. From a power consumption perspective, the power saving obtained was in the 85–96% range when compared to the continuous transmission of synchronization packets.

B. The Problem of M/S Reciprocal Synchronization

In the first implementation of FLSA [5], we focused only on guaranteeing the maximum time accuracy on which slaves agree given the lowest power consumption. This is sufficient when the application calls for a master that is only in charge of orchestrating the entire network procedures. The analysis of different physical parameters in a controlled environment, such as a laboratory or a medical center, provides a possible example of a field of application. In this particular situation, since the

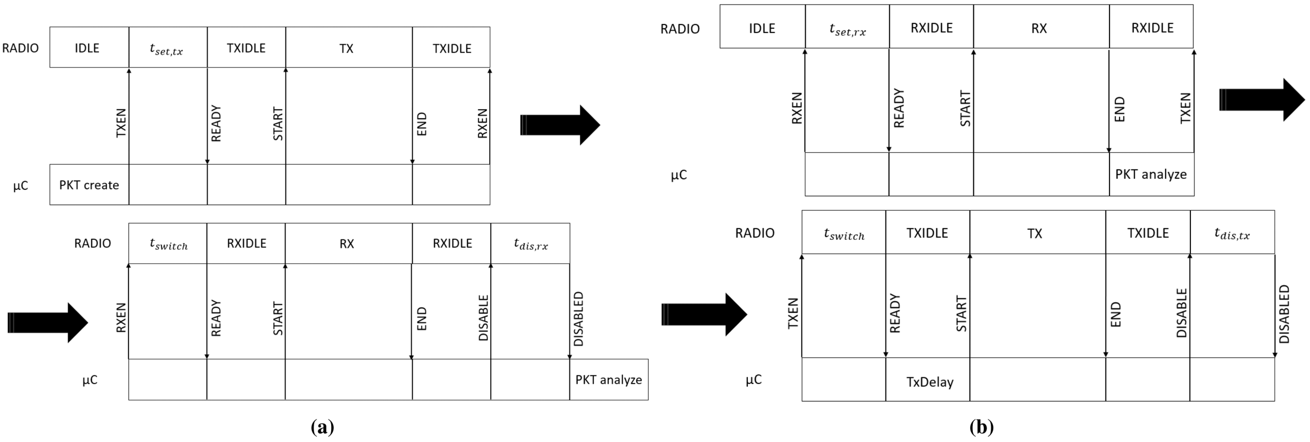


Fig. 2. Time diagram of signals and events describing the μC-Radio interaction. In detail: (a) Tx-Rx sequence (b) Rx-Tx sequence.

TABLE I
NOMINAL TIME VALUES FOR THE PHASE DURATION BETWEEN THE LISTED EVENT/SIGNAL ENDPOINTS DECLARED ON THE DATASHEET FOR THE nRF52832 SoC [11]

Symbol	Value [µs]	Time interval endpoints
$t_{\text{set},tx}$	140	{TXEN, READY}
$t_{\text{dis},tx}$	6@1MSps/4@2MSps	{*TXDISABLE*, DISABLED}
$t_{\text{set},rx}$	140	{RXEN, READY}
$t_{\text{dis},rx}$	0	{*RXDISABLE*, DISABLED}
$\min\{t_{\text{switch}}\}$	20	Tx ⇄ Rx

master was only periodically broadcasting the synchronization message, we made the hypothesis of excluding the times needed for the packet generation, dispatching, and traveling through the channel. In fact, these times are clearly equal for each slave, and does not impact the overall slave synchronization. In this scenario, the aim is to minimize the impact of the monitoring devices on the patients' life. Therefore, one key factor is to reduce the number of boards employed, and so the master itself should expose some sensors in order to capture biosignals.

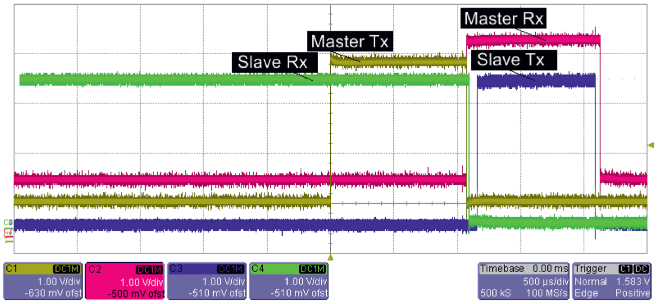
As a consequence, a further improvement on FLSA is necessary, since the slave agreement on a common time is a necessary but not sufficient condition to the master agreement. This is due to several reasons. To better understand, let us consider the stages during the transmission and reception of a data packet on the nRF52832 SoC by Nordic [11], which will be also the SoC employed in our experiments. The SoC is built around an Arm Cortex M4 CPU with floating-point unit running at 64 MHz. We will refer to the microcontroller unit as μC. The radio section (to which we will refer to as radio) is able to handle multiple protocols and it is also being capable of managing full protocol concurrency. In particular, it supports Bluetooth LE, NFC, ANT, and 2.4 GHz proprietary protocols. For the description of the general transmission scheme, we will refer to what is represented in Fig. 2(a). In the first phase, the packet is generated on the transmitter side. After that, the transmitter enabling signal TXEN is issued. At this point, the $t_{\text{set,tx}}$ is needed, that is the time interval in which the radio is ramping up and preparing for transmission. After this phase, the READY signal is issued from the radio to the μC. During the *TXIDLE* phase, the radio is ready for transmission to be started with the start signal generated by the μC. The transmission lasts for an interval corresponding to the *TX* phase in which the radio is transmitting a packet. The END signal indicates the radio section going idle.

It is possible to identify similar steps in the receiving sequence diagram depicted in Fig. 2(b). In fact, as soon as the RXEN signal is issued, the time $t_{\text{set,rx}}$ is needed from the radio for ramping up and preparing for the packet reception. During the *RXIDLE* phase, the START signal allows the radio to start the packet reception, hence starting the *RX* phase. Once the reception has been completed the radio enters an *RXIDLE* state with the END signal being issued to the μC. Analogously to the transmitter case, in the *RXDISABLE* phase, the radio is shutdown after the DISABLE signal is issued from the μC. At the end, the packet parser elaborates the received information.

Finally, the case of transmission and immediate reception of a packet and vice-versa are shown in Fig. 2(a) and (b) respectively. In these cases the radio is not disabled, so we considered the time t_{switch} employed for the radio to switch from the transmission (Tx) to the reception (Rx) of packets and vice-versa, maintaining the same frequency. The meaning of the *TxDelay* value will be explained later on.

In order to better appreciate the impact of the duration of each phase, we summarize in Table I each timing value from the datasheet [11]. From those data, it is possible to say that the minimum delays introduced by the radio are in the order of hundreds of microseconds in both transmission and reception. In addition to these time values, we also should consider the preparation time and parsing time for the packet, as well as the traveling time of the packet in the radio medium.

We prepared an experimental setup to validate the previously shown values and rely on measuring the time intervals through GPIO toggling and an oscilloscope. For each board, we considered two pins. The first one was enabled during the Rx phase starting from the time instant in which the firmware initializes the receiving phase by issuing $RXEN$ until the packet has been parsed. The other pin was enabled during the Tx phase, starting



C1 DC1M 1.00 V/div -630 mV ofst
C2 DC1M 1.00 V/div -500 mV ofst
C3 DC1M 1.00 V/div -510 mV ofst
C4 DC1M 1.00 V/div -510 mV ofst
Timebase 0.00 ms 500 µs/div 500 kS 100 MS/s
Trigger C1 DC Normal 1.583 V Edge Positive

Fig. 3. Oscilloscope screenshots of the GPIO toggling describing, the M/S sending a packet, the M/S complete calibration and the S/M sending a packet back.

from the packet creation until its transmission was completed (RXENABLE signal in Fig. 2(a)). The time values for a packet of length 15 bytes transmitted at a speed of 250 kbps from the master to the slaves are shown in Fig. 3, where it is evident a 1.07 ms time difference between the beginning of the master transmission and the slave reception of the packet.

This implies the timestamp received by the slave is 1.07 ms late with respect to the master timestamp.

A possible solution can be represented by forcing the master to send the synchronization packet proactively. To achieve this goal, we have to first evaluate the time needed for the packet to arrive from the master to the slave. The strategy we propose to make such evaluation simply calls for the master sending a synchronization packet to the slave that, in turn, directly sends it back without any change. Therefore, we also analyzed the packet bouncing times. A delay (*TxDelay*) of 100 μs between the end of the slave calibration packet reception and its back-transmission to the master (see Fig. 2(b)) has been introduced. This was necessary since the slave does not process the received packet, it just sends it back to the master, hence this delay helps handling this asymmetry in the packet exchange. In the Fig. 3 reports the measurements obtained observing this process. The overall measured delay is equal to 2.12 ms, which is the time necessary for the master to send the calibration packet and get an answer back from the slave. The measured delay for the S to M packet transfer and processing is about 1.05 ms.

The delays difference between master-slave and slave-master packet transmission is obviously due to the slightly different timing of the SoCs in switching from idle to Tx ready state ($t_{\mathrm{set},tx}$) and, then, from Tx to Rx ($t_{\mathrm{switch},tx}$) and vice-versa. Furthermore, the slave does not have a packet preparation phase since it simply sends back the same received packet.

The delay can be automatically evaluated by the master, comparing the timestamps at the beginning of packet transmission and reception, obtaining the total time from which, in turn, it is possible to extract the value to be summed to the timestamp sent by the master to the slave to obtain a perfect synchronization. In the practical implementation, the estimated values will be rounded to the implemented t_{ST}, which, in our case, it is equal to 100 μs. As mentioned before, the measured transmission time from master to slave is slightly greater than the reciprocal time. Based on this experimental evidence, the delay can be evaluated using the formula:

Algorithm 1: FLSA M/S Time Synchronization—Master Side.

```
Require: S set of slaves, $t$ timestamp.
1. Randomly choose $s \in \mathbf{S}$
2. Switch to Tx mode.
2.$P\{t = \mathrm{T_{tx}}\} \rightarrow s$ Send a packet $P$ to $s$ containing the
   current timestamp $T_{sent}$
3. Switch to Rx mode and wait until RXEN event.
4. Receive a packet $s \rightarrow P\{t = \mathrm{T_{tx}}\}$ at time $\mathrm{T_{rx}}$
5. Compute $\Delta_\tau = \mathrm{T_{tx}} - \mathrm{T_{rx}}$
6. Compute $\Delta\mathrm{T_M^{sync}} = \lceil \Delta_\tau / 2 \rceil \cdot \mathrm{t_s}$
```

Algorithm 2: FLSA M/S Time Synchronization—Slave Side.

```
Require: $m$, master node.
1. Switch to Rx mode and wait until RXEN event.
2. Receive a packet $m \rightarrow P\{t = \mathrm{T_{tx}}\}$
3. Send the same packet $P\{t = \mathrm{T_{tx}}\} \rightarrow m$
4. Idle.
```

$$\Delta\mathrm{T_M^{sync}} = \left\lceil \frac{T_{\mathrm{tx}} - T_{\mathrm{rx}}}{2} \right\rceil \cdot \mathrm{t_{ST}} = \left\lceil \frac{\Delta_\tau}{2} \right\rceil \cdot t_{ST} \tag{6}$$

where the operator $\lceil \cdot \rceil$ returns the rounding to the greater integer and T_{tx} and T_{rx} are the sent timestamp and the current timestamp expressed as multiples of t_{ST}. As reported before, a 2.12 ms total time was measured and, according to (6):

$$\Delta\mathrm{T_M^{sync}} = 1.1\ \mathrm{ms} \tag{7}$$

From the master perspective, the whole M/S mutual calibration procedure is summarized in Algorithm 1. For what concerns the slaves, the calibration steps are summarized in Algorithm 2. This sequence should be repeated several times and the obtained values averaged in order to obtain the optimal $\Delta\mathrm{T_M^{sync}}$. In our experiments the value obtained is the same as in (7).

III. System Description

As described in Section I, in this paper we also present a low-cost synchronized platform for the long-term off-line global health monitoring of patients. The system has an M/S architecture with a star network topology. In particular, the system is made up of one master unit and a group of three slaves. The master unit is the HRSpO2board-HW-1-0-0 board shown in Fig. 4(a). It is based on a nRF52832 SoC, allocating on-board a high-sensitivity $\mathrm{SpO_2}$ sensor, an HR sensor for wearable health on the backside of the board, and a light and environment temperature sensor. The board supports microSD cards for data storage and it is power supplied by a CR2032 coin lithium battery. This unit is responsible for the network time synchronization and samples the HR and $\mathrm{SpO_2}$ sensors with a 1 sps rate. Two different types of slaves are used. Both of them are based on the same nRF52832 SoC as the master [12].

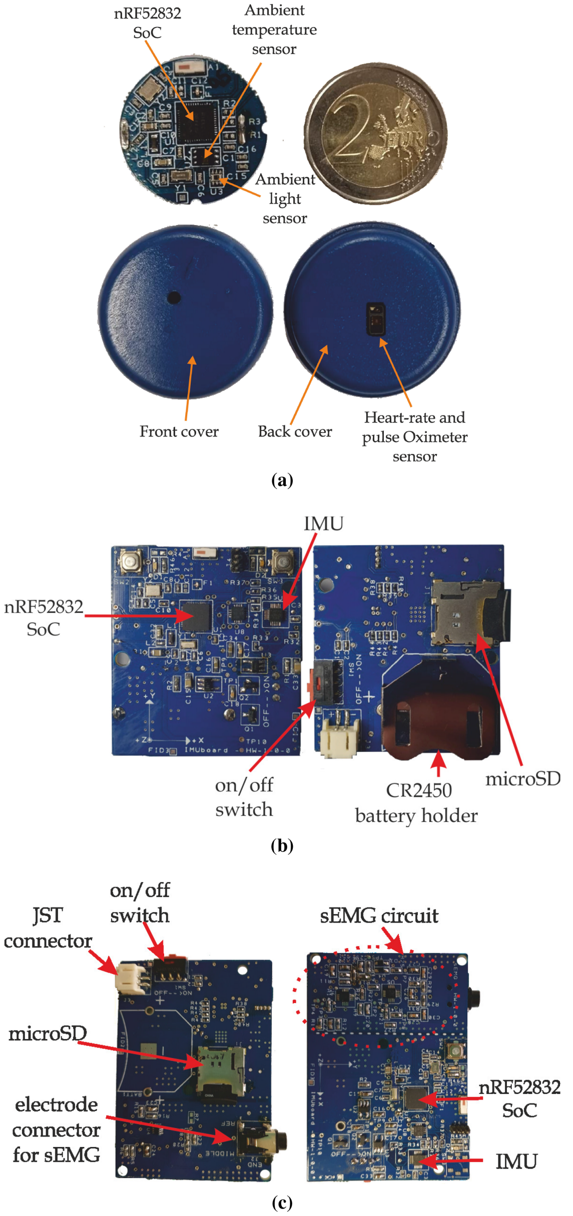


Fig. 4. The custom designed boards. In detail: (a) The HRSpO2board-HW-1-0-0 board that has role of master in the proposed system, (b) the sEMG-HW-1-0-0, and (c) IMUboard-HW-1-0-0 boards having slave roles.

TABLE II
COMPARISON FOR THE VALUES OF THE MASTER AND SLAVES f_t

Before the synchronization phase			
Master [Hz]	**Slave#1** [Hz]	**Slave#2** [Hz]	**Slave#3** [Hz]
72.84	75.86	93.62	83.95
After the synchronization phase			
Master [Hz]	**Slave#1** [Hz]	**Slave#2** [Hz]	**Slave#3** [Hz]
72.84	72.84	72.82	72.82

The first is the IMUboard-HW-1-0-0 and it is shown in Fig. 4(b). It allocates a nine degrees of freedom (DoF) MPU-9250 nine-axis MEMS MotionTracking device from TDK-InveSense (San Jose, CA, USA), a microSD card for data storage, and it is powered by a CR2450 lithium coin cell battery. It samples the onboard IMU sensor with a 100 Hz rate. The sEMG-HW-1-0-0 is shown in Fig. 4(c). It is an extension of the IMUboard-HW-1-0-0 previously described. It allocates a nine DoF MPU-9250 nine-axis MEMS MotionTracking device from TDK-InveSense (San Jose, CA, USA), a microSD for data storage, the circuit for the sEMG, and it is powered by a lithium battery connected to the JST connector. It samples the onboard IMU sensor at 100 Hz and the sEMG output circuit at 50Hz. The sEMG output circuit is a square wave that indicates if there is muscle activity or not as described in [12].

IV. MEASUREMENT CAMPAIGN

In this section, we present the results achieved during two measurement campaigns. In particular, after implementing the FLSA in the monitoring system described in the previous section, we first validated the results in terms of achieved synchronization accuracy after a one-month-long experiment in the laboratory. Second, having demonstrated the feasibility of the approach, we tested the entire monitoring system on a healthy subject for one day.

A. *Validation of the Time Synchronization Accuracy*

In a preliminary phase, we verified the M/S synchronicity. The free-running timer is visualized by measuring the GPIO toggling rate on an oscilloscope. Setting the ST $t_{ST} = 100\ \mu$s and toggling the GPIO every 50 ticks results in a $\mathrm{T}_t = 10$ ms nominal square-wave output period ($f_t = 100$ Hz). The reported results show that at the beginning of the measurement campaign, each unit has a different frequency, quite far from f_t, as reported in Table II and Fig. 6(a). This measure of the RTC mismatch is due to many reasons, including the tolerance of the low-cost crystal oscillator used, the aging of the component, the PCB parasitics, etc. [13]. After the synchronization phase, the master imposes its ST to the slaves, as shown in Table II and Fig. 6(b). It is worth notin\g that the STs are synchronous, despite the clocks being not synchronized and having, in any case, a ±100 μs uncertainty. The ±100 μs uncertainty is now related to the master clock time.

In this measurement campaign, we also evaluated what happened during the transient of the synchronization phase. Fig. 7(b) sketches the experimental setup. A PC was connected to the master using Bluetooth Low-Energy (BLE) to issue the ST start command. The master, then, sent a packet, using a proprietary protocol, to the set of slaves, starting their own STs (see 7a). After that, the master waited for the previously calculated interval, $\Delta\mathrm{T}_{\mathrm{M}}^{\mathrm{sync}}$, to start its own ST, thus ensuring the simultaneous start for all the units in the system. Starting from this phase, each device wrote on the microSD card every 10 ms its own timestamp. The field dedicated to the sensor was kept empty to prevent recording any possible delay associated with the availability of the data from the sensors.

We used a $\mathrm{T}_{\mathrm{M}} = 10$ s and imported the data in MATLAB for analyzing the ST variation of each unit. In the forthcoming

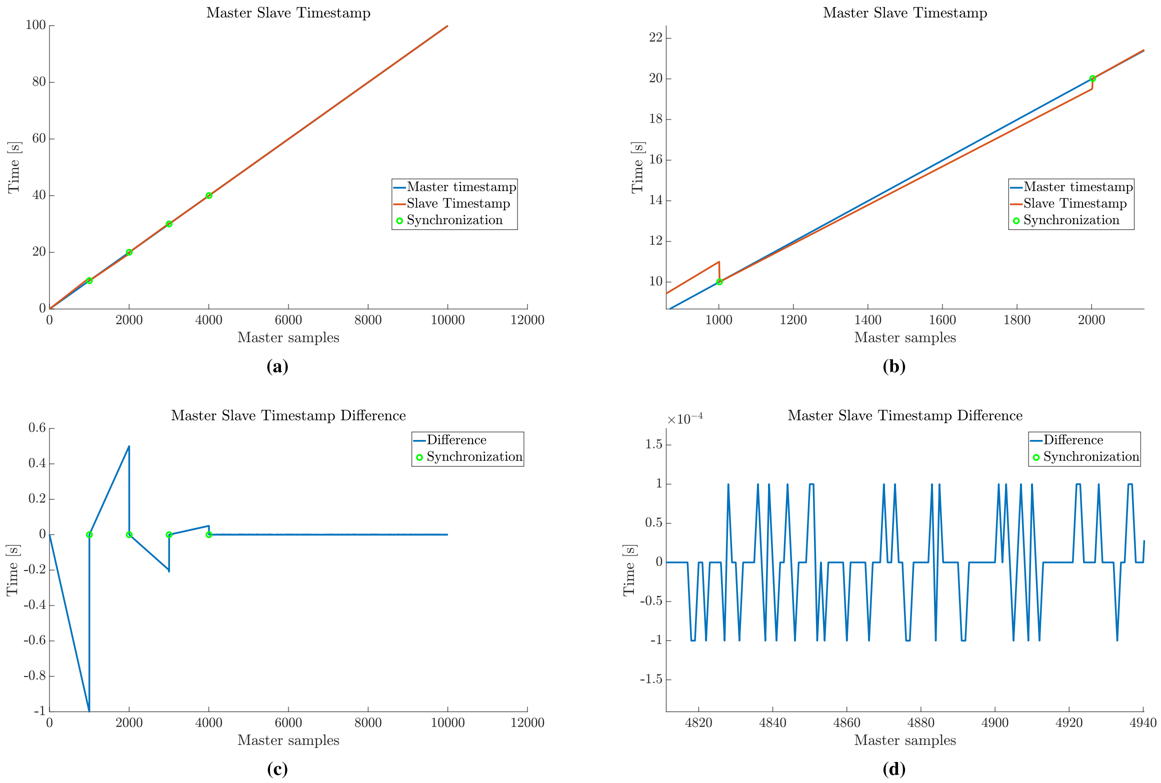
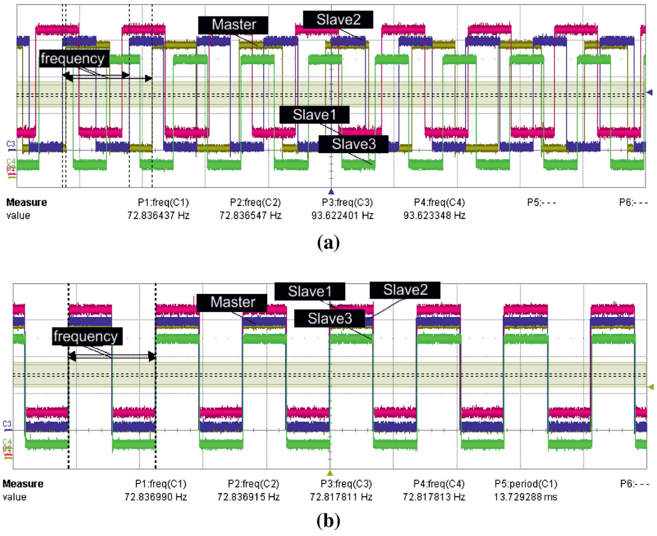


Fig. 5. (a) Master and slave#2 ST. (b) Zoom on Master and slave#2 ST. (c) Master and slave#2 ST difference. (d) Zoom on Master and slave#2 ST difference.



Measure value P1:freq(C1) 72.836437 Hz P2:freq(C2) 72.836547 Hz P3:freq(C3) 93.622401 Hz P4:freq(C4) 93.623348 Hz P5:- - - P6:- - -

Measure value P1:freq(C1) 72.836990 Hz P2:freq(C2) 72.836915 Hz P3:freq(C3) 72.817811 Hz P4:freq(C4) 72.817813 Hz P5:period(C1) 13.729288 ms P6:- - -

Fig. 6. Oscilloscope measurement screenshots of (a) Master and slaves clocks not synchronized and (b) Master and slaves clocks synchronized after calibration.

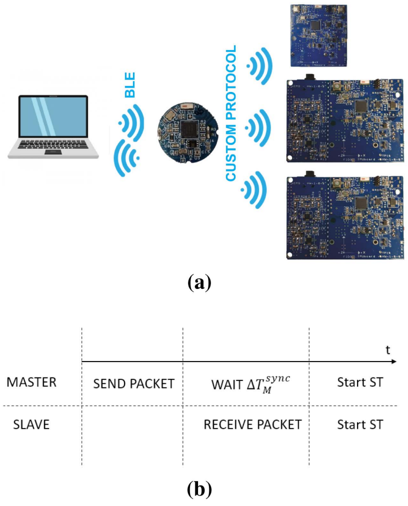
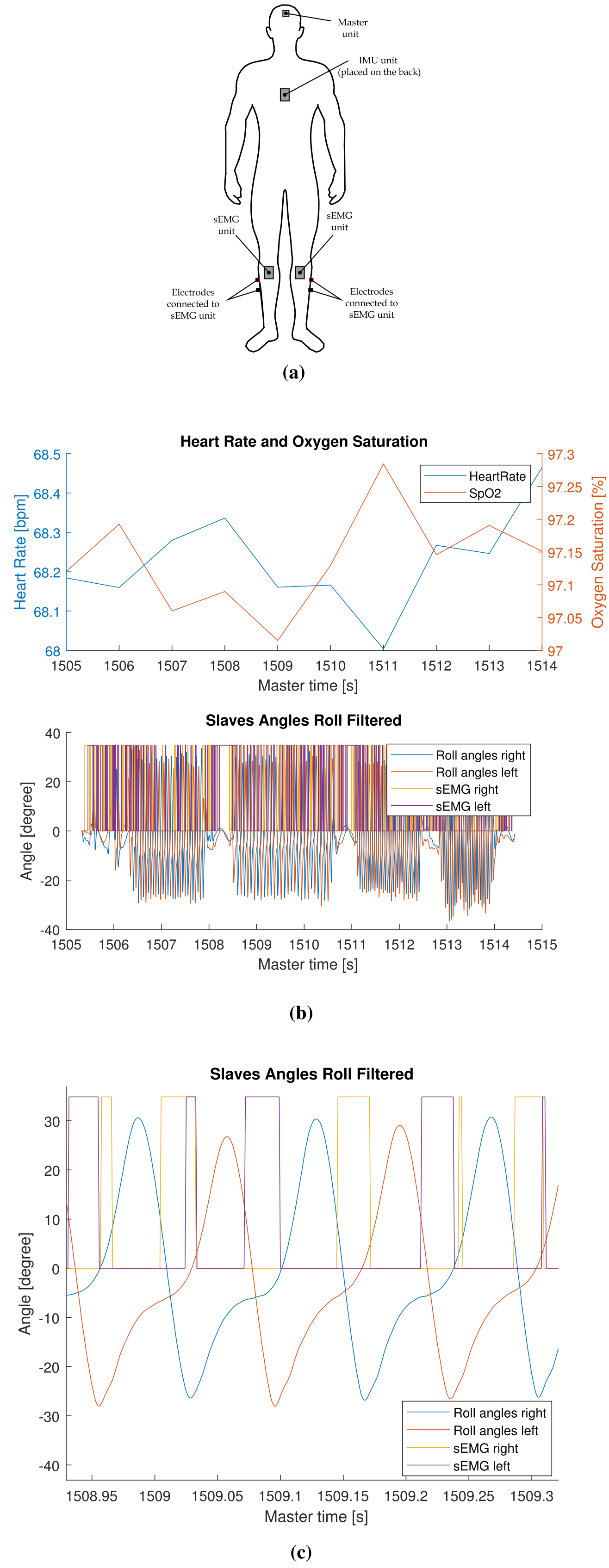


Fig. 7. Synchronization accuracy test experimental setup: (a) Test-bench organization and (b) Time diagram.

TABLE III
SYNCHRONIZATION EXPERIMENTAL RESULTS

	Slave #1	Slave #2	Slave #3
max S_s	253	229	251
min S_s	5	5	5
avg S_s	82	87	85

figures, for sake of clarity, we reported only the STs of the master and only a single slave. Fig. 5(a) reports the overall ST variation while Fig. 5(b) zooms on the initial phase, highlighting how eventually the slave synchronizes its timestamp to the master even if initially it was faster. To achieve this result, it changes its fractional ST parameters to become slower than the master. It is worth highlighting that in Fig. 5(a) and (b) the slave samples have been normalized to the master ones and, then, interpolated. Fig. 5(c) shows the timestamp difference between master and



(a)

(b)

(c)

Fig. 8. Overview of the real-world scenario experiments. In detail: (a) Positioning of master and slaves boards. (b) Analysis of HR, SpO_2 and rolls data collected through the experiments. (c) Close-up on the gait analysis.

slave, while Fig. 5(d) zooms on an apparent noise generated by the fact that, as discussed in Section II, the STs have been synchronized but not the clocks.

In the second group of experiments, the units operated continuously for four weeks in the laboratory, to verify the effectiveness of the already tested algorithm [5], with the addition of the new mutual M/S synchronization phase. Table III summarizes the measured results, where we recall that $\mathrm{S_s}(j)$ denotes the skipped slots for node j. The obtained values emphasize the perfect synchrony of the slaves with the master. Each slave stored in a microSD the time in which the synchrony is lost and, it assumes not to be in synchrony with the master if its ST is 500 μs greater of lower than that received by the master.The apparently low value of the means is due to the fact that the FLSA, every time it loses a single packet, it resets itself. The results are very good, showing that the slaves work for almost 9.5 hours before losing the synchrony, with an evident power consumption reduction.

B. *System Validation*

In this last group of measurements, the units have been used in a real scenario, with a healthy subject wearing them for a 1-day long monitoring campaign. The units have been positioned as shown in Fig. 8(a), with the goal of measuring the gait, the calf muscle activity, the subject posture, the HR and the SpO_2. The master is positioned in a hat on the subject head. This position gives the minimum interference with everyday activities, allowing a correct measurement of HR and SpO_2 [14].

Fig. 8(b) reports an excerpt of the data processing during the campaign recording. In the upper part there are shown the collected HR and SpO_2 values, recorded each second, while in the lower part there are shown the roll angles calculated in MATLAB R2021b starting from the shin-bones data recorded by the slaves on the left and right sides. The left and right sEMG are reported, too. The recorded data can be easily processed through a PC, allowing for finer and more complex analyses correlating the data recorded by many units.

In Fig. 8(c) the observation time has been reduced, giving better insight into the elaborated data. The shin-bones angles during the walk, along with the muscles activity, are clearly shown. These values are used for gait analysis, showing the perfect symmetry between right and left legs in a healthy subject. The result indirectly validates that the data collected by different units placed in different body regions still maintain perfect synchrony due to the proposed algorithm.

V. Conclusions and Future Work

In this paper, we presented a distributed full synchronized system for monitoring patients' global health, along with an evolution of the FLSA. In particular, this latter provided for the synchronization not only among the slaves but also with the master. The system was first tested in the laboratory and then in a one-day-long test on a healthy subject. The achieved results in terms of synchronization accuracy demonstrated how the improved version of FLSA allows reaching low synchronization errors and it is stable with time. Also, the preliminary results on a healthy patient demonstrated how the algorithm performs well in a real-world scenario.

It is beyond the scope of this paper to make an in-depth analysis of the measurement campaign on biological signals itself. However, in future developments, the system will be integrated with drug therapy and posture or gait analysis to evaluate the efficacy of the treatment [15].

References

[1] N. M. Freris, S. R. Graham, and P. R. Kumar, "Fundamental limits on synchronizing clocks over networks," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1352–1364, Jun. 2011.

[2] *Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 15–6: Wireless Body Area Network*, ISO/IEC/IEEE 8802-15-6:2017(E), pp. 1–274, 2018.

[3] D. Renggli *et al.*, "Wearable inertial measurement units for assessing gait in real-world environments," *Front. Physiol.*, vol. 11, 2020, Art. no. 90. [Online]. Available: https://www.frontiersin.org/article/10.3389/fphys.2020.00090

[4] G. Coviello, G. Avitabile, C. Talarico, J. Roveda, and A. Florio, "A fractional approach to time synchronization in wireless body area networks," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, 2021, pp. 933–936.

[5] G. Coviello, G. Avitabile, A. Florio, C. Talarico, and J. M. Wang-Roveda, "A novel low-power time synchronization algorithm based on a fractional approach for wireless body area networks," *IEEE Access*, vol. 9, pp. 134916–134928, 2021.

[6] L. Mo, S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, and P. S. Freedson, "Wireless design of a multisensor system for physical activity monitoring," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 11, pp. 3230–3237, Nov. 2012.

[7] S. S. Vedaei *et al.*, "COVID-safe: An IoT-based system for automated health monitoring and surveillance in post-pandemic life," *IEEE Access*, vol. 8, pp. 188538–188551, 2020.

[8] R. K. Pathinarupothi, P. Durga, and E. S. Rangan, "IoT-based smart edge for global health: Remote monitoring with severity detection and alerts transmission," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2449–2462, Apr. 2019.

[9] D. Wang, J. Ouyang, P. Zhou, J. Yan, L. Shu, and X. Xu, "A novel low-cost wireless footwear system for monitoring diabetic foot patients," *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 1, pp. 43–54, Feb. 2021.

[10] G. Coviello, G. Avitabile, and A. Florio, "The importance of data synchronization in multiboard acquisition systems," in *Proc. IEEE 20th Mediterranean Electrotechnical Conf.*, 2020, pp. 293–297.

[11] "nRF52832 Product Specification v1.4." Accessed: May 10, 2022. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf

[12] G. Coviello, G. Avitabile, and A. Florio, "A synchronized multi-unit wireless platform for long-term activity monitoring," *Electronics*, vol. 9, no. 7, 2020, Art. no. 1118.

[13] G. Coviello and G. Avitabile, "Multiple synchronized inertial measurement unit sensor boards platform for activity monitoring," *IEEE Sensors J.*, vol. 20, no. 15, pp. 8771–8777, Aug. 2020.

[14] S. K. Longmore, G. Y. Lui, G. Naik, P. P. Breen, B. Jalaludin, and G. D. Gargiulo, "A comparison of reflective photoplethysmography for detection of heart rate, blood oxygen saturation, and respiration rate at various anatomical locations," *Sensors*, vol. 19, no. 8, Apr. 2019, Art. no. 1874.

[15] S. U. Simjee, H. Jawed, J. Quadri, and S. A. Saeed, "Quantitative gait analysis as a method to assess mechanical hyperalgesia modulated by disease-modifying antirheumatoid drugs in the adjuvant-induced arthritic rat," *Arthritis Res. Ther.*, vol. 9, 2007, Art. no. R91.

Giuseppe Coviello (Senior Member, IEEE) was born in Bari, Italy, in 1981. He received the Laurea degree in electronic engineering and the Ph.D. degree in electric and information engineering from the Polytechnic University of Bari, Bari, in 2006 and 2015, respectively. He is currently an Assistant Professor with the Department of Electrical and Information Engineering, Polytechnic University of Bari. His research interests include RF and mixed-signal circuits and embedded systems for healthcare.

Antonello Florio (Student Member, IEEE) was born in Bari, Italy, in 1995. He received the B.Sc. degree (with Hons.) in computer and automation engineering and the M.Sc. degree (with Hons.) in telecommunications engineering from the Polytechnic University of Bari, Bari, where he is currently working toward the Ph.D. degree in electronics. He took part to a double degree program with the University of Nice-Sophia Antipolis, Nice, France, where he achieved the Master de Recherche in ubiquitous networking and computing. His research interests include phased arrays, theories and techniques for localization, and green wireless sensor networks.

Gianfranco Avitabile (Senior Member, IEEE) was born in Livorno, Italy, in 1958. He received the B.S. and M.S. degrees in electronics from Florence University, Florence, Italy in 1982. He joined the Department of Electronics of University of Florence as an Assistant Professor. From the 1998, he joined the Department of Electronics, Polytechnic University of Bari, Bari, Italy, as an Associate Professor, where he currently works. His research interests mainly include the high frequency circuits and systems for both civil and military applications and high-performance analog integrated circuits for telecommunication applications.

Claudio Talarico (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Genoa, Genoa, Italy, and the Ph.D. degree in electrical engineering from the University of Hawai‘i at Mānoa, Honolulu, HI, USA. He is currently a Professor of electrical and computer engineering with Gonzaga University, Spokane, WA, USA. Before joining Gonzaga University, he was with Eastern Washington University, Cheney, WA, The University of Arizona, Tucson, AZ, USA, and with industry, where he held both engineering and management positions with Siemens Semiconductors, IKOS Systems, and Marconi Communications. His research interests include digital and mixed analog/digital integrated circuits and systems, computer-aided design methodologies, and design and analysis of systems-on-chip.

Janet M. Wang-Roveda received the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, Berkeley, CA, USA, in 1998 and 2000, respectively. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ, USA. Her primary research interests interests include robust VLSI circuit design, biomedical instrument design, smart grid, VLSI circuit modelling/design and analysis, and low power multicore system design. She was the recipient of the NSF Career Award and the Presidential Early Achievement Award for Science and Engineering at White House in 2005 and 2006, respectively, 2008 R. Newton Graduate Research Award from the EDA Community, and 2007 USS University of Arizona Outstanding Achievement Award.