

Article

A Multi-Channel Electromyography, Electrocardiography and Inertial Wireless Sensor Module Using Bluetooth Low-Energy

Giorgio Biagetti ^{*,†} , Paolo Crippa [†] , Laura Falaschetti [†] , Claudio Turchetti [†] 

DII–Dipartimento di Ingegneria dell’Informazione, Università Politecnica delle Marche, Via Brecce Bianche 12, I-60131 Ancona (AN), Italy; p.crippa@univpm.it (P.C.); l.falaschetti@univpm.it (L.F.); c.turchetti@univpm.it (C.T.)

* Correspondence: g.biagetti@univpm.it

† These authors contributed equally to this work.

Received: 23 April 2020; Accepted: 26 May 2020; Published: 4 June 2020



Abstract: This paper proposes a wireless sensor device for the real-time acquisition of bioelectrical signals such as electromyography (EMG) and electrocardiography (ECG), coupled with an inertial sensor, to provide a comprehensive stream of data suitable for human activity detection, motion analysis, and technology-assisted nursing of persons with physical or cognitive impairments. The sensor is able to acquire up to three independent bioelectrical channels (six electrodes), each with 24 bits of resolution and a sampling rate up to 3.2 kHz, and has a 6-DoF inertial platform measuring linear acceleration and angular velocity. The bluetooth low-energy wireless link was chosen because it allows easy interfacing with many consumer electronics devices, such as smartphones or tablets, that can work as data aggregators, but also imposes data rate restrictions. These restrictions are investigated in this paper as well, together with the strategy we adopted to maximize the available bandwidth and reliability of the transmission within the limits imposed by the protocol.

Keywords: EMG; ECG; accelerometer; gyroscope; wireless sensor; bluetooth low-energy

1. Introduction

Nowadays, wearable wireless sensors placed on the skin of the human body are becoming very common in many application fields such as healthcare, sport, fitness, ambient assisted living, entertainment, autonomous driving systems [1,2]. In particular, three body signals (i.e., electromyography (EMG), electrocardiography (ECG), and acceleration (ACC)) that can be easily captured by these sensors, have been demonstrated to be able to provide accurate and reliable information on people’s activities and behaviors. Therefore, low-cost and low-energy wearable wireless EMG/ECG/ACC sensors can be efficiently used for all day long telemonitoring of patients affected by dementia or neurodegenerative disorders characterized by progressive decline in cognitive and functional abilities. Behavioural and psychiatric symptoms are often present in these cases and make patient management very hard. The quality of life (QoL) experienced by family caregivers has been shown to be lower than the QoL of caregivers for persons who do not have dementia [3,4]. Research has shown that lower QoL can increase absences from work and reduce job productivity.

To identify a potential system able to help caregivers in patient daily-management will therefore be of value. It will allow to improve quality of life of the patients and their caregivers and will reduce the social cost linked to these diseases. In particular, caregivers designate certain activities as dangerous, and it is of paramount importance to detect such activities in order to provide alarm functions to these patients [5,6]. An accurate detection of activities can be reached by acquiring information both on acceleration and other biosignals such as ECG and EMG signals at the same time [7,8]. With these

considerations in mind, combining EMG, ECG and inertial sensors in a single device allows to obtain all the necessary information for injury prevention during normal living activities [9–13], and for human activity monitoring in general of patients affected by cognitive diseases or dementia.

This work thus presents a multichannel EMG, ECG and inertial wireless sensor based on bluetooth low-energy (BLE) that could be used for daily activity monitoring of patients affected by physical or cognitive diseases as well as in a large number of situations in which the body movements and activities have to be accurately monitored and recognized, thus realizing a concrete solution which should reduce the impact of disease on patients (and families) and increase quality of life for people with dementia and for their caregivers, reducing healthcare costs.

In order to achieve these goals, the technology used should be as unobtrusive as possible. We chose to use BLE 4.0 as the communication link due to its widespread availability, and though its design limitations are clearly a challenge for high-throughput, low-latency, reliable real-time data streaming [14], it allows the system to dispense with dedicated radio receivers and use commodity hardware for receiving and processing the captured data. An accurate analysis of the BLE protocol allowed these challenges to be solved by developing a custom adaptation layer on top of BLE. Although the built sensor is also BLE 5 ready, so that when a suitable receiver is available, limitations to data rates and number of simultaneously streaming channels can be lifted, we made an effort to make it compatible with BLE 4.0 because of its large installed base, hence ensuring the widest adoptability of the proposed solution.

Despite using a radio protocol not designed for this type of applications, a careful selection of all the system components, system integration, upper protocol design and software/firmware cooperation, allowed our system to achieve very tight time synchronization, with a root mean square (RMS) timing error of just 6.6 μ s, which is important in multiple-sensors scenarios. The developed sensor also sports low-latency operation, less than 10 ms, necessary if the data must be acted upon quickly as in emergency situations, and resilience to radio interference, with lossless, error-free data transfer even with up to 1/3 of the radio channel capacity compromised.

System cost and battery life are also of concern. We designed the system with maximum power savings in mind, so that it can operate continuously streaming data for days on a single charge, while remaining compact and lightweight to limit discomfort while wearing. Such a system would also be of little usefulness if its cost prevented a widespread adoption, since it is intended to be used mostly at home and outside of clinical settings. This consideration guided the design towards the adoption of cost-effective, highly integrated sensing solutions whenever possible.

The paper is organized as follows. Section 2 presents the state of the art in EMG, ECG and inertial wearable sensors. Section 3 illustrates the system architecture, with emphasis on power saving strategies, while Section 4 contains a detailed description of the developed custom protocol, implemented as a service on top of a standard BLE stack, to achieve reliability, strict time synchronization, and controlled latency. Section 5 presents experimental results and measures on a fabricated prototype of the sensor to show its performance and the operation of the developed protocol, while Section 6 concludes the work.

2. Related Work

The modern advancements in micro and nanoelectronics technologies, in wireless wearable communication systems, as well as in machine learning and signal processing of biological data [15,16] are leading to a new generation of health monitoring systems [17,18] and interfaces for easy interaction with the real world. Miniaturized low-power lightweight wireless sensors have been embedded into clothes, belts, shoes, sunglasses, smartwatches and smartphones, or placed straight to the skin in order to collect data related to skin temperature, body movements and positions, heart rate, and muscle fatigue. The captured signals such as electromyography (EMG), electrocardiography (ECG), and acceleration provide accurate and reliable information on people's activities and behaviors. Thus, wearable sensors of this kind, placed on the skin of the human body are becoming very common

in many application fields such as healthcare, sport, fitness, ambient assisted living, entertainment, autonomous driving systems, surveillance-based security [1,2].

On the one hand, with the progress of the signal processing techniques, more and more information can be derived from biosignals [19–23]. It has been widely demonstrated how capturing EMG and ECG signals of a person can efficiently and easily assess its health status [24,25], body posture [26], fitness level [27], and physical performance [28–30]. Besides, due to the fact that acquiring EMG and ECG signals need intrinsically noninvasive measurement devices, and are relatively easy to obtain, many applications based on them have been developed. As an example, both EMG and ECG signals are recently being used in modern car systems for developing driving interfaces that allow to start, steer, accelerate, and brake the vehicle [31], or for monitoring the driver health and cognitive status for guaranteeing road safety [32]. Traditionally, devices used to record bioelectrical potentials generated by skeletal muscles are different from the ones used to detect heart activity [33], with few exceptions [34,35]. The main reason for this differentiation is that the ECG signal bandwidth is typically below 100 Hz, whereas EMG signals have significant spectral components extending at least up to 500 Hz, and are more likely to be acquired during movements and are hence affected by artifacts induced by body and cable movements at frequencies typically below 5 Hz. In order to avoid saturation, these low-frequency motion artifacts must be rejected by the high-input-impedance, low-noise amplifier that should be carefully designed to capture the relatively low amplitude of the EMG signal.

On the other hand, if the goal is to detect and recognize activities of daily living (ADL), inertial measurements are very important to evaluate the physical activity as they allow to record variations in orientation and easily detect body movements. Thus, many accelerometry-based wearable systems for the pervasive monitoring ADL have been developed. Some of these systems are simply based on one inertial sensor located usually on upper arms, wrist, waist, ankle [36–39] or consist of applications based on smartwatch and smartphone data [40–43]. However, they are effective in monitoring and classifying activities that involve repetitive body motions, such as running, walking, climbing stairs, cycling, or lifting weights [44]. Thus, for better detecting human activities, some techniques based on several accelerometer sensors placed on the body have been recently proposed [45,46]. Additionally, data derived from a triaxial accelerometer have been demonstrated to be very useful in the reduction of motion artifacts corrupting the surface EMG (sEMG) signals recorded from the wrist or arm of the subjects that are performing various activities [47].

Other recent research on these topics include, e.g., [48], where a detailed description of multichannel electrocardiography is provided and the most important vital signs that could be provided by multifunctional body sensors are presented. Among them, a system monitoring the patient by combining an ECG sensor (placed on an appropriate position on the user's chest) and two accelerometers (attached on the user's chest and the right thigh using elastic Velcro straps) is presented in [49]. It recognizes the user's activities and detects falls using the accelerometer data.

A system for collecting acceleration, ECG, and sEMG signals, was developed using a modified off-the-shelf BITalino, to allow its use in water related bio-signal research [50]. The BITalino was secured to over the head and the sensors were connected to the prototype as follows: a ground electrode was placed behind the ear over the edge of the skull, three ECG leads were placed in the horizontal plane precordial position, the triaxial accelerometer sensor was placed on the lumbodorsal fascia and the four EMG sensors were placed on triceps brachii, caput laterale and on rectus femoris. It can reliably stream up to 24 kbit/s over BLE.

In [51], a wearable bracelet for automotive applications was developed to monitor the driver's health status collecting live bio-signals data through embedded multiple bio-sensors such as EMG, ECG, accelerometer, electrodermal activity, and temperature. It has demonstrated to be able to detect health threatening and emergency situations.

Research has also been done in developing on-silicon system-on-chip such as [16], a highly integrated batteryless sensor node implemented in TSMC 0.18 μm CMOS technology for ECG/EMG

applications with wireless powering and compressed sensing. The entire implementation contains the analog front-end (AFE), analog-to-digital converter (ADC), digital signal processor (DSP), power management, and an RF-to-DC rectifier and its size is of 1.7 mm × 2.5 mm.

Recently, in [52] a wearable hardware platform to measure ECG and EMG with an additional IMU sensor to detect the motion artifacts has been developed. Bringing all the sensors on single platform resolves the sensor fusion problems. Additionally, the developed platform successfully detected the motion artifacts during all the events and the necessary features of ECG and EMG waveforms are very well extracted by using signal processing techniques. The work is a contribution towards the development of integrated wearable biomedical sensors for long-term monitoring.

Finally, in [53] a novel ExG measurement system built around a custom designed ultra-low noise instrumentation amplifier (ULN-INA) has been proposed. Two versions of the prototype were presented: (a) with one INA and measures all the ExG signals in time-multiplexed manner; (b) with four INAs and can measure up to four parallel channel ExG simultaneously. Both have programmable gain and bandwidth features, which allows measurement of different bio-potentials such as ECG, EMG, EEG, EOG.

Despite all the research that has been done on these body sensors, most has been focused on sensing and signal processing for compression and recognition, while very little attention has been paid to the transport of the acquired signals through the wireless channel and to the overall system optimization. Unoptimized protocols are often used, resulting in sub-optimal performance, like 100 ms latency and only 6 h operation in [18], or low throughput in [50], to name a few that employ BLE. In our work, the communication protocol, though remaining fully standards-compliant to BLE, has been adapted to ensure reliable, controlled-latency transmission of the raw data acquired by the sensors, resulting in latencies lower than 10 ms and net data rates (not counting protocol overhead) in excess of 80 kbit/s, while keeping data integrity and tight time synchronization even in the presence of severe RF interference, and allowing days of battery-powered continued operation.

3. Hardware and System Description

As the system is expected to be used for prolonged amounts of time, its design has been driven by the requirement of lowering its power consumption as much as possible, so as to maximize battery life. Hence, particular emphasis will be given to the low-power operation of the whole sensor, which is achieved by a careful selection of hardware components and features, combined with a tailored power management strategy developed in software.

In the following two subsections an overview of the hardware components selected to meet our goals is provided, together with a description of the device operating modes, which include an acceleration-based wake-up system designed so that the unit can just be “worn and forgotten” (at least for a few days, until the battery needs recharging).

3.1. Circuit Implementation

The wireless sensor module is composed of the following main components:

- Microcontroller with integrated wireless transceiver. Using the versatile nRF52840 System-on-Chip (SoC) by Nordic Semiconductor, programmed to use the BLE radio protocol stack, it offers the needed wireless connectivity together with wired (USB) connectivity for battery charging, simplified and secure pairing, etc.
- Biopotential analog front-end. Based on the Texas Instruments' ADS1293 highly integrated front-end (FE), which includes configurable digital filters, instrumentation amplifiers and ADCs, augmented with an external biasing network suited for both EMG and ECG signal acquisition through either DC or AC-coupled electrodes.
- Inertial platform. Made with the ultra-low-power LSM6DSO digital accelerometer and gyroscope by ST Microelectronics, it features an always-on accelerometer used to wake up the system when motion is detected, and is used to stream 6-axes motion data when the system is operational.

- Power supply. It consists of a lithium-polymer battery charger and a bunch of linear and switching regulators to provide power to the different subsystems only when actually needed, and designed to maximize energy efficiency.

A block diagram of the system, with the power path highlighted, is shown in Figure 1.

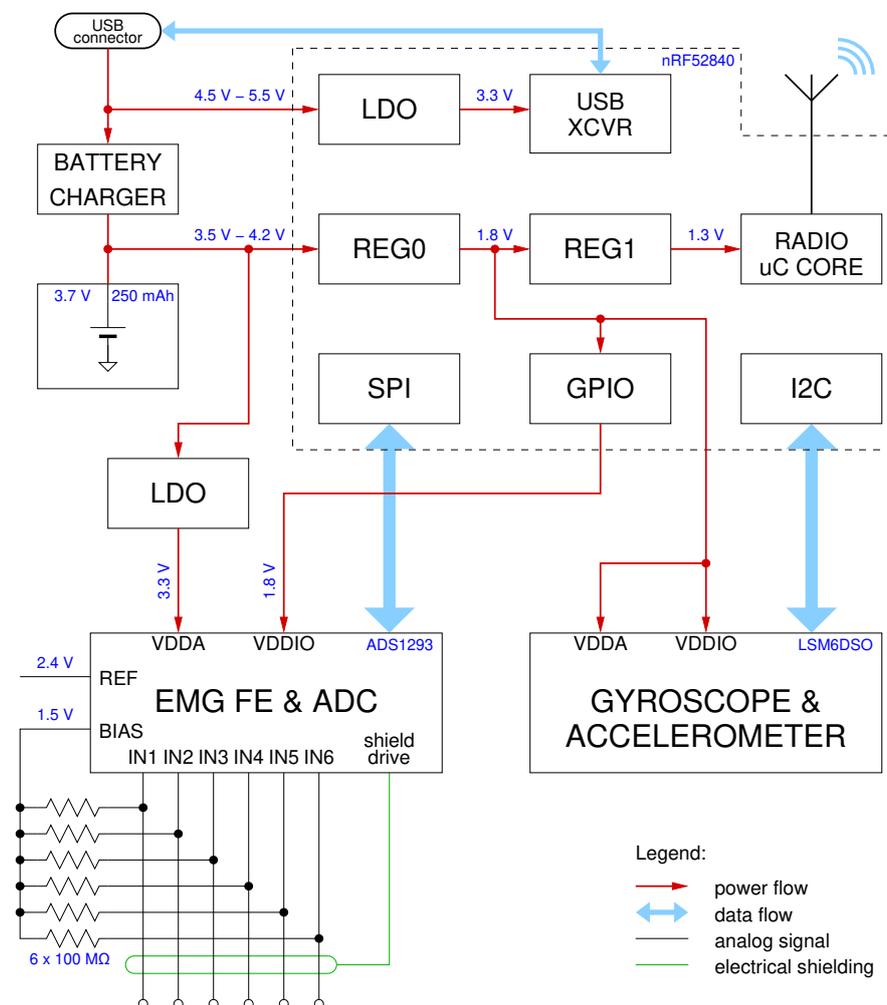


Figure 1. Block diagram of the sensor node highlighting the various power supply subsystems exploited to minimize energy consumption and the data interfaces. The ADS1293 acquires the analog signals and transmits them to the SoC through a serial peripheral interface (SPI) when enabled by the general-purpose input/output (GPIO) line. The LSM6DSO integrates the inertial platform and streams its data to the microcontroller (uC) through an inter-integrated circuit (I2C) interface.

The nRF52840 SoC already integrates a number of voltage regulators, specifically one low dropout (LDO) linear regulator dedicated to the USB transceiver (XCVR) (inactive during battery-powered operation), a high-voltage regulator (REG0) that can accept the voltage range from a lithium-polymer battery to produce the main I/O supply voltage V_{DD} , and the normal-voltage regulator (REG1) that further reduces V_{DD} to the internal supply used by the microcontroller core and radio. These two latter regulators can be configured to operate either as LDOs or switching regulators.

Although REG0 can be controlled by software to produce a voltage between 1.8 V to 3.3 V in 0.3 V steps, and so it could in principle be used to also power the EMG front-end (which accepts a 2.7 V to 5.5 V range), an external low-noise LDO was added to the system to improve the noise performance of the analog front-end and also reduce current consumption when the ADS1293 needs not to be powered, as the LDO provides a much lower stand-by current than the ADS1293 itself.

The ADS1293 contains the analog FE, suitable for sensing either ECG or EMG signals. It has three fully differential simultaneous ADCs, so it can stream three different signals at once, with digitally controllable filters that simultaneously adjust the bandwidth and the output data rate. These filters can thus be programmed according to the type of signal (ECG or EMG) being acquired. Each ADC differential channel can then be connected to any two of the six input leads, under software control, for maximum usage flexibility.

System timing is provided by two crystal oscillators controlled by the SoC: a 32.768 kHz low-frequency crystal oscillator (LFXO), which is used for the bluetooth radio slot scheduling and overall application timing, and a 64 MHz high-frequency crystal oscillator (HFXO), used when accurate high-frequency clocks are necessary (radio, USB, etc.) and to timestamp sampled data. The 409.6 kHz clock necessary for the sigma-delta ADC in the analog front-end is also derived by means of a fractional divider from the HFXO.

A photograph of an assembled prototype circuit board is shown in Figure 2. The overall circuit occupies 51 mm × 27 mm, but a much tighter integration is clearly possible for the final version. Electrodes will be connected (possibly through shielded cables) to contacts J1–J6.

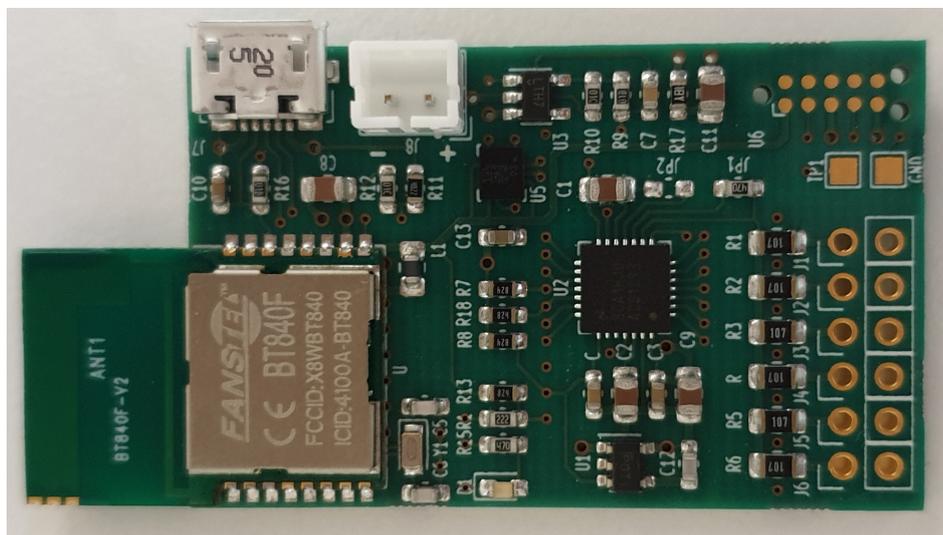


Figure 2. Photograph of the sensor prototype built for testing and evaluation purposes. The EMG/ECG sensing chip is U2, the inertial sensor is U5.

3.2. System Operational Modes

At any given time the system can be in one of five different operational modes (OFF, IDLE, ADVERTISING, CONNECTED, STREAMING), each of which is characterized by different power requirements:

- **OFF:** This is the lowest power state, with the system completely disabled, meant to be used for long-term storage, with a current drain comparable or lower than the battery self-discharge rate. It is entered upon request from the host, or when a low-battery condition is detected. It can only be exited by attaching the device to the battery charger. In this state, the CPU and accelerometer are programmed in the power-down state for lowest current consumption. The EMG front-end is not powered at all.
- **IDLE:** This is entered after a period of inactivity, i.e., after advertising for a long time without any BLE central requesting a connection. In this state the accelerometer is kept active in an ultra-low-power state and a low (12.5 Hz) sampling rate. When it detects a change in the acceleration measured on any of its three axes (e.g., when the sensor is rotated around an axis not parallel to gravity, or it is slightly shaken), it resets the CPU and moves the system to the advertising state. The EMG front-end is not powered at all as in the OFF state.

- **ADVERTISING:** This state is entered after a wake-up from IDLE, a reset from OFF, or after the host disconnects the BLE link. In this state the accelerometer is kept as in IDLE to prevent the advertising from timing out if there is movement, as is the EMG front-end, still unpowered. The CPU is timed from an external LFXO to reduce power consumption with respect to the internal RC oscillator.
- **CONNECTED:** Entered when a BLE central establishes a connection. In this state the hosts can control most of the functions of the sensor. By default, only the battery monitor is enabled, the LSM6DSO is powered down, while the ADS1293, which has a relatively long start-up time due to the high-value biasing resistors, is powered on and kept in stand-by awaiting for the host to request data streaming. The CPU is still timed by the LFXO.
- **STREAMING:** Entered when the host enables notifications for a specific service. In this state the requested data source is enabled at the prescribed data rate, and the uncompressed data stream is sent over the BLE link. For accurate data timestamping, the high-frequency crystal oscillator (HFXO) is kept active at all times and not only during radio activity as is done in the previous two states. This results in a slightly higher current consumption but it is necessary to ensure proper data synchronization.

A summary of the expected current consumption for the three main subsystems in the different modes is reported in Table 1. The “ADS1293” column also includes the quiescent current of its dedicated external low-noise voltage regulator. The “min” and “max” streaming currents refer to the number of active channels: from one to three EMG channels for the ADS1293, linear acceleration only or linear acceleration plus angular velocity (gyroscope) for the LSM6DSO.

Table 1. Nominal current consumption of the different subsystems in the different operational modes. The microcontroller nominal current in the active states is not indicated because it depends on the running software, see Section 5.4 for actual measured values.

Mode	nRF52840	ADS1293	LSM6DSO
OFF	1.0 μ A	1.0 μ A	3.0 μ A
IDLE	1.0 μ A	1.0 μ A	5.5 μ A
ADVERTISING	-	1.0 μ A	5.5 μ A
CONNECTED	-	135 μ A	3.0 μ A
STREAMING (min)	-	410 μ A	170 μ A
STREAMING (max)	<15 mA	900 μ A	550 μ A

4. Data Communication and Processing

The developed hardware platform is all but a small part of the whole system design, as to be able to stream all the data the sensors can acquire, over a BLE 4.0 link, requires a carefully crafted protocol to make the most of the available bandwidth while ensuring reliable transmission. This section analyses the challenges, and details the proposed solutions we implemented to achieve our overall goal.

4.1. Protocol Description

As the system is designed to be highly configurable, with different sampling frequencies available, different number of channels, and simultaneous EMG or ECG and inertial sensor acquisition, a versatile data transmission protocol was chosen. Indeed, since the only difference between ECG and EMG signal acquisition lies in the setting of the sampling frequency (ODR), as the integrated filters adjust automatically, there is no need to differentiate the two cases in the protocol, and in the following we will only refer to “EMG” for both cases.

The sensor implements the BLE generic attribute profile (GATT) server, which exposes three main custom services, besides the customary device information and battery services. The first custom service is the electromyography service, the second is the inertial platform service, the third is a system configuration and telemetry service. Each custom service provides a characteristic from which

notifications of sampled data are sent, and a writable control characteristic to set operating parameters, like output data rate (ODR), full-scale settings, number of active channels, etc.

This enables both flexibility and ease of implementation at the receiver end. If the default data rates (800 Hz for EMG and 104 Hz for the inertial platform) are used, the receiver only has to enable notifications for the desired services after having established a connection, and decode the incoming stream of notifications. As an example, a schematic representation of the data format for the three custom characteristics is show in Figure 3.

Electromiography characteristic (UUID: cef92ff1-4179-5abe-9ab4-f41a779bd20f)											
SEQ number	EMG sample A channel #1	EMG sample A channel #2	EMG sample A channel #3	EMG sample B channel #1	EMG sample B channel #2	EMG sample B channel #3	SUB channel				

Gyroscope characteristic (UUID: 23786211-74c8-581e-8cd9-a1ac54584ef3)										
SEQ number	TYPE	THERMO LSM temperature	GYRO channel X	GYRO channel Y	GYRO channel Z	ACC channel X	ACC channel Y	ACC channel Z	SUB channel	

System status characteristic (UUID: b77f3a31-bfe3-5d75-b732-0402ba0ff0e7)											
PWR mode	CHIP mode	SYS health	SYS status	VBUS +5V rail	VUSB +3.3V internal	VBAT battery monitor	IBAT battery charger	VDDA +3.3V rail	TMPA ADS temperature	VDDD +1.8V rail	TMPD SoC temperature

Figure 3. Illustration of the different packet structures. The maximum length of each packet is 20 bytes. Fields in a gray background are present only in a few packets and contain metadata with structure and timing information. The EMG data format is flexible so that different combinations of active channels and samples per packet are allowed, as described by the associated metadata, and can result in shorter packets, with anything between one and six samples per packet.

The main data channels contain first an 8 bit sequence number, needed to properly reorder samples at the receiver in case of packet loss and retransmission, as will be detailed later, and last some optional metadata split into different “subchannels”. Each subchannel is a 32 bit word of information, describing aspects of the stream such as ODR and channel config (subchannel 0), data sampling timestamps (subchannel 2), and packet transmission timestamps (subchannel 4), as depicted in Table 2.

For the EMG characteristic, since only 8 bits per packet are available, each subchannel word is split into four consecutive packets, and are identified by the sequence number n in the packet (subchannel m goes into packets with $m = \lfloor n/4 \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor rounding operator). For the other characteristic where 32 bits per packet are available, it is simply $m = n$.

To maximize bluetooth bandwidth efficiency, multiple samples from the sensors are possibly accumulated and sent in a single notification packet, and multiple packets are sent within a BLE connection interval. This is depicted in Figure 4.

As each physical sensor samples data, they are timestamped and put in a small ring buffer. The clock used for these timestamps is derived from the main HFXO, has a resolution of 1 μ s, and is latched in hardware when the “data ready” signal from the sensor is asserted, so that it is not affected by jitter and delays introduced by firmware code execution. When enough samples have been accumulated to fill a BLE packet, the packet is assembled, a sequence number is assigned, and the resulting formatted data packet is “tentatively” scheduled for transmission.

“Tentatively” here means that, to deal with the possibility of congestion in the RF channel, we keep track of the number of packets Q in the TX queue, and avoid adding another packet to it if it is too full ($Q \geq 6$). This can be done because packets are added to the queue by the application code and removed when a BLE link-layer acknowledgment (ACK) is received from the central—though we use notifications and not indications, link-layer ACKs are always mandated by the BLE protocol. The BLE stack also retries a packet indefinitely, until either an ACK is received or the connection is considered lost. To better demonstrate a typical sequence of events, Figure 5 shows a couple of examples without (top) and with (bottom) some RF interference.

Table 2. Subchannel definitions. Metadata is optionally added to packets to describe acquisition parameters such as data rate, number of active channels, full-scale ranges, and timing information. The type and format of metadata depends on which subchannel it is transmitted on. Subchannels are identified by the sequence number of the packet containing them. Four consecutive packets are needed to convey a single subchannel for the EMG stream as only extra 8 bits are available on the data packets. For the GYR stream this splitting is not needed and a whole subchannel is included in a single data packet. Care has been taken in choosing subchannel indices, as of course configuration metadata is needed to interpret the main data and must come with the first packet, while transmission timestamps (which refer to the time packet #0 was transmitted) are available after the fact and their inclusion in the stream must be postponed to account for queuing delays.

Stream	SUB #	SEQ	Bits	Content
electromyography	0	0	0 ÷ 3	ODR [from 50 Hz to 6400 Hz, encoded as $1 + \log_2(F_S \cdot 10 \text{ ms})$]
			4 ÷ 5	number of active EMG channels (1 ÷ 3)
			6 ÷ 7	[reserved for future use]
		1	8 ÷ 10	channel 1 negative lead number (1 ÷ 6)
			11 ÷ 13	channel 1 positive lead number (1 ÷ 6)
			14 ÷ 15	channel 1 test mode
		2	16 ÷ 18	channel 2 negative lead number (1 ÷ 6)
			19 ÷ 21	channel 2 positive lead number (1 ÷ 6)
			22 ÷ 23	channel 2 test mode
		3	24 ÷ 26	channel 3 negative lead number (1 ÷ 6)
27 ÷ 29	channel 3 positive lead number (1 ÷ 6)			
30 ÷ 31	channel 3 test mode			
	2	8 ÷ 11	0 ÷ 31	sampling timestamp $t_{AD}/1 \mu\text{s}$ (encoded as 32 bit unsigned integer)
	4	16 ÷ 19	0 ÷ 31	transmit timestamp $t_{TX}/1 \mu\text{s}$ (encoded as 32 bit unsigned integer)
gyroscope	0	0	0 ÷ 1	accelerometer full-scale range [2 g, 4 g, 8 g, 16 g]
			2 ÷ 3	gyroscope full-scale range [250 dps, 500 dps, 1000 dps, 2000 dps]
			4 ÷ 7	ODR [from 12.5 Hz to 6666 Hz, encoded as $1 + \log_2(F_S \cdot 76.8 \text{ ms})$]
			8 ÷ 31	[reserved for future use]
	2	2	0 ÷ 31	sampling timestamp $t_{AD}/1 \mu\text{s}$ (encoded as 32 bit unsigned integer)

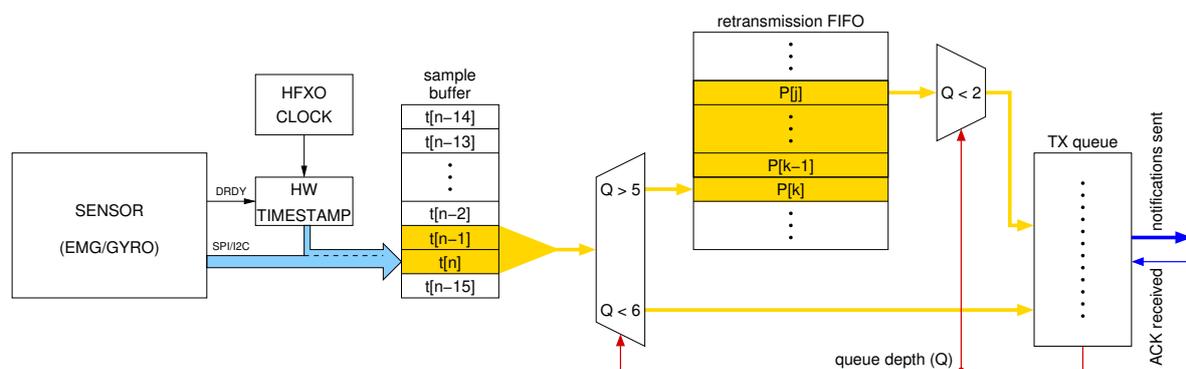


Figure 4. Illustration of the buffering strategy employed.

As can be seen, the total supply current measurement provides many insights into the operation of the system, with clearly visible spikes corresponding to packet transmission. Spikes are grouped in bursts corresponding to each connection event. Under normal operation, three EMG packets are transmitted during each connection event, and one GYR packet about every four EMG packets, as expected at the default data rates. RX timestamps are obtained in software on the PC, so there is a small amount of delay and jitter, but the sequence is clearly correct. On the bottom panel, on the other hand, it is possible to see that several connection events fail altogether (they could also be cut short instead of totally failing, though this did not happen in the time span shown). In this case, queued packets are simply delayed until the next connection event, when they are transmitted together (before, actually) the newly arrived packets. As can be seen, a connection event has room for up to nine packets

this size, so there is quite some room available for retransmissions since only three or four slots are normally used.

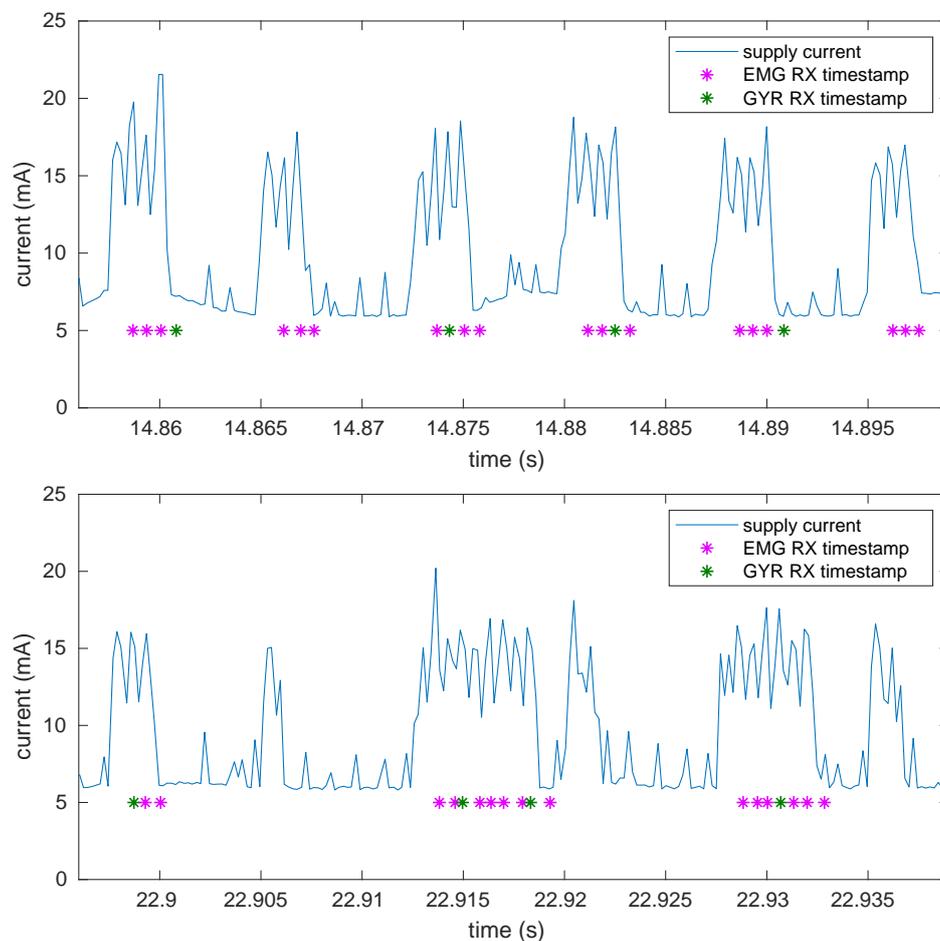


Figure 5. Total supply current to the sensor node and packet reception timestamps. Spikes around 15 mA correspond to the activation of the radio transmitter, while the lower spikes (around 8 mA) to CPU activity (mostly sensor data readout). The baseline is higher than normal because the measurement was performed with a debug probe attached to better analyse timing and firmware behavior, but it does not otherwise affect the results. Stars (placed at an arbitrary height) denote the time at which a packet was received by the PC acting as central. The time scales of the current measurement instrument and of the PC with the bluetooth receiver had been synchronized by servoing both clocks to the same, local time server. A BLE 4.0 dongle was used for the measure, so the on-air bit-rate is 1 Mb/s.

This automatic retransmission scheme inherent in BLE ensures that a queued packet will eventually always reach the central, unless the connection is severed. However, it also means that the latency is potentially unbounded, which is undesirable in the present application, and in case of severe RF interference the TX queue can also fill up, leading to loss of the most recent packets in favor of the older ones already queued, another undesirable behavior.

To mitigate these problems, a separate retransmission FIFO is held by the application firmware, where packets that are not immediately sent to the TX queue (during congestions) are temporarily stored. This FIFO is much bigger than the TX queue (currently 256 entries in the FIFO and 16 in the TX queue) and operates in “overwrite” mode, with new packets overwriting the oldest ones in case it fills up. These packets will then be removed from the FIFO and queued for transmission only after the congestion is resolved, i.e., when $Q < 2$ as shown in the figure. Typically, one extra packet is inserted per connection event. These “late” packets are specially marked as such, and the bits normally reserved for subchannel data are instead used to extend the sequence number from 8 bit to 15 bit, so as

to allow proper reordering at the receiver side in case they get delayed by more than the time it takes for the standard 8 bit counter to wrap around.

This strategy increases the likelihood that the most recent samples reach the receiver in a timely manner, while ensuring that, if some additional latency is tolerated, packets that could not be sent due to RF channel congestion/interference have a chance to be transmitted later and placed in the correct position in the data stream.

As previously said, the protocol is also flexible in terms of sampling rates and EMG channel configurations. Figure 4 showed the case where a packet contains $N_S = 2$ samples of data, which happens, e.g., for a 3-channel EMG signal sampled at $F_S = 800$ Hz with a $N_B = 24$ bit resolution. As each sample thus produces 9 bytes of data, two of them fit nicely into the maximum 20 bytes GATT payload of version 4.0 BLE, with room to spare for the packet index and one byte of subchannel data. A new packet is thus ready to be queued every 2.5 ms, and using the minimum allowed $T_1 = 7.5$ ms connection interval, on average $N_P = 3$ packets per connection event will then be sent. So, though the exact number of packets sent per connection event is not exactly fixed because the BLE timing is controlled by the central, whose clock is not synchronized to the clock that drives the ADC on the peripheral, under normal circumstances there is still plenty of airtime for other data (inertial sensor, retransmissions, etc.).

Table 3 shows a selection of different possible buffering schemes allowed by the presented strategy. N_C is the number of active channels, F_S the specified ODR, N_B the resolution of each sample. N_S is then the number of samples per packet, chosen so that $N_G < 20$ bytes, where $N_G = N_C N_B N_S$ is the number of data bytes per packet. The average number of packets per connection event is also computed as $N_P = T_1 F_S / N_S$. Simultaneous EMG and ACC or ACC + GYRO data streaming is allowed as long as the total N_P does not exceed the available airtime or the receiver queue handling capabilities, with six being the recommended limit from the SoC manufacturer.

Table 3. Sampling and buffering strategies for a few different combinations of data to be streamed and ODR, which fits within the limits of BLE 4.0.

Type	N_C	F_S (Hz)	N_B (Bits)	N_S	N_P	N_G (Bytes)
EMG	1	3200	24	6	4	18
EMG	2	1600	24	3	4	18
EMG	3	800	24	2	3	18
ECCG	3	200	24	2	0.75	18
ACC	3	104	16	2	0.39	12
ACC + GYRO	6	104	16	1	0.78	12
ACC + GYRO	6	208	16	1	1.56	12
ACC + GYRO	6	416	16	1	3.12	12

4.2. Receiver-Side Processing

The job of the receiver is to decode the incoming stream of notifications, interpret metadata, and reorder and synchronize the data streams so that they can be sent for easy processing to the data consumer(s).

Data consumers can be any module suited for real-time display of the signals, disk-recording of the streams for offline processing and analysis, artificial intelligence (AI) modules for automated real-time signal analysis, etc. They are outside the scope of the present paper, here we will only present the processing needed to present them with ready-to-use data streams.

Different consumers may pose different requirements on allowed latency. For instance, AI modules that then send alerts or otherwise react to critical situations might require very low latencies. Data display modules may tolerate latencies up to fractions of a second, while disk-recording is not sensitive to latency at all.

Hence, the receiver software we implemented provides multiple output ports from which data can be received by the consumers, and each port has an independently configurable latency.

From “minimum latency” (around 9 ms, see later), where a direct path from receiver to consumer is formed, discarding delayed packets and with a coarse clock synchronization, to “infinite latency”, suitable for off-line analysis. Of course, the higher the latency the more time is given to the transmitter to perform “retransmissions” in case of packet loss, thus filling the gaps in the signal, and allows more time to the filters used for clock synchronization to settle and reduce jitter. Time-critical applications must instead deal with the possibility of a higher packet-loss rate. This is unavoidable in a possibly lossy wireless link such as BLE.

A block diagram of the data flow within the receiver software is shown in Figure 6.

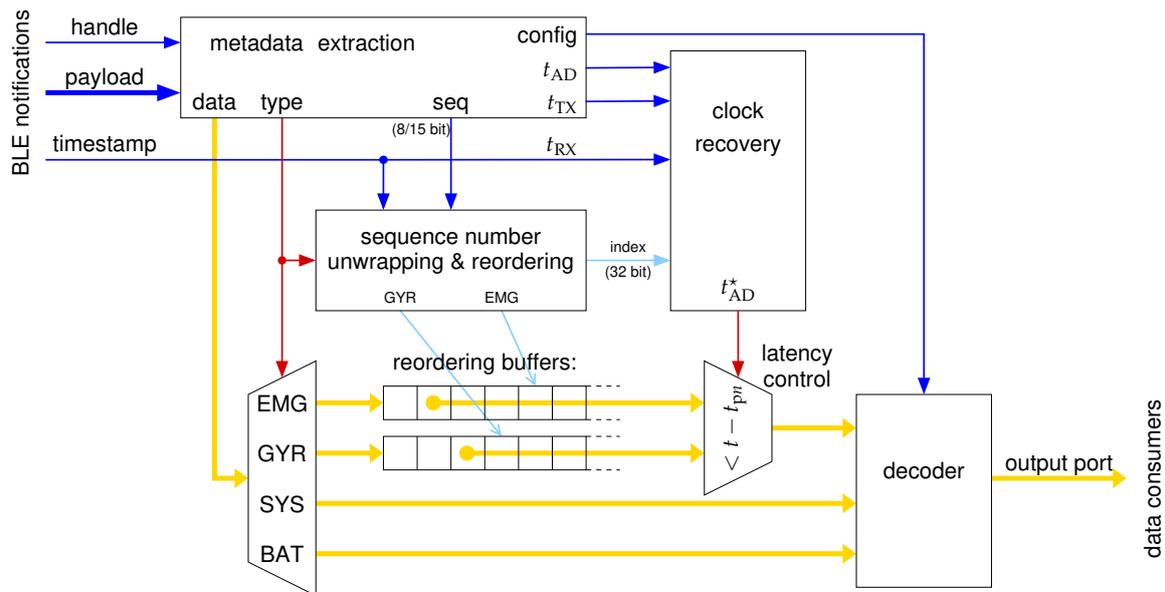


Figure 6. Block diagram of the data flow within the receiver software. Only one output port is shown, the “latency control” block is replicated for each port using a different value of the latency t_{pn} , n being the port number.

The BLE stack provides the incoming stream of notifications to the receiver, each notification is tagged with its reception timestamp (as measured by the PC clock, assumed synchronized to real-time clock) and the GATT handle through which it was received, so that the packet type can be recognized. The first step is to extract metadata from the GATT payload, namely, 8 bit sequence number and subchannel data (configuration information, sampling timestamps t_{AD} , and packet transmission timestamps t_{TX}).

Next, the sequence number must be unwrapped to form the complete packet index (32 bit in our implementation but it can be easily extended if need be). Transmit-side retransmissions and the possibility of packet loss (packets might also be lost at the receiver side if the CPU is under stress for unrelated tasks) must be accounted for. Thus, the reconstruction procedure makes use of the packet reception timestamps t_{RX} to rebuild the higher-order bits of the index as follows. Let ΔT be the time interval between the reception of the previous packet and the current packet, and let T_P be the expected (average) time interval between two packets. Let n_1 be the current packet sequence number (LSB only), and n_0 the same for the previous packet, so that we can define $\Delta n = (n_1 - n_0)_{256}$, where the symbol $(\cdot)_{256}$ denotes signed modular arithmetics, such that $-128 < \Delta n < +128$. Let then us finally define

$$N = \frac{\Delta T}{T_P} - \Delta n \tag{1}$$

as the difference between the expected number of packets within the time elapsed, and the actual increment of the sequence number. Normally, N should be close to 0. Specifically, since packets are transmitted in bursts and up to six packets might be queued for transmission waiting for the next

connection event, it should be $-6 < N < 6$, even when a counter wraparounds occurs, as this is taken care of by the modular arithmetics in computing Δn . In case of a wraparound during a long sequence of missed packets, on the other hand, N might be close to an integral multiple of 256. It thus suffice to add

$$\Delta i = \Delta n + 256 [N/256] \quad (2)$$

(where $[\cdot]$ denotes rounding to nearest integer) to the current value of the packet index i kept in the receiver memory to obtain the correct, unwrapped, index.

Of course, the above procedure does not make sense in case the packet being processed is an automatic retransmission of a previously lost packet. In such case, ΔT is meaningless. These packets can be detected by a special flag bit set in the sequence number itself together with the presence of subchannel data. This allows 15 bit of sequence number to be transmitted, enough to avoid wraparounds for over 80 s. Packets older than that will simply be discarded at the transmitter side, it would not make much sense to keep retrying packets that old.

Of course, separate indices for the two main data services are kept. As depicted always in the same Figure 6, these indices are then used to store the data part of the received packet in the proper position within dedicated reordering buffers, so that retransmitted packets ends up in the correct order.

The output port control logic for port n then extracts packets from these buffers after a specified latency t_{pn} , i.e., when $t > t_{AD}^* + t_{pn}$, where t_{AD}^* is the estimated sampling time derived from the packet index, to allow time for retransmissions to arrive and to let the clock recovery block do its calculations and filtering. Extracted packets are then decoded from their on-air format to physical units, and individual samples sent out to the appropriate data consumer together with their estimated sampling timestamp t_{AD}^* , obtained from t_{AD} after adjustment of the timescale of the sender to that of the receiver as will be shown in the next section.

4.3. Clock Synchronization

This last task is performed by the “clock recovery” block, whose operation can be described with the aid of Figure 7. In the figure dots represent the difference $\delta = t_{RX} - t_{TX}$, where t_{RX} represent the real time at which a particular packet (all those with sequence number equal to 0) was received, and t_{TX} represent the time, according to the device clock, at which the same packet was transmitted. Note that the BLE protocol itself does not guarantee latency, so transmit timestamps can only be obtained after the fact, when the packet had already been transmitted. For this reason these timestamps are sent in subchannel 4 (packets 16–19 of the EMG sequence), to allow enough time for packet 0 to actually be transmitted under normal circumstances.

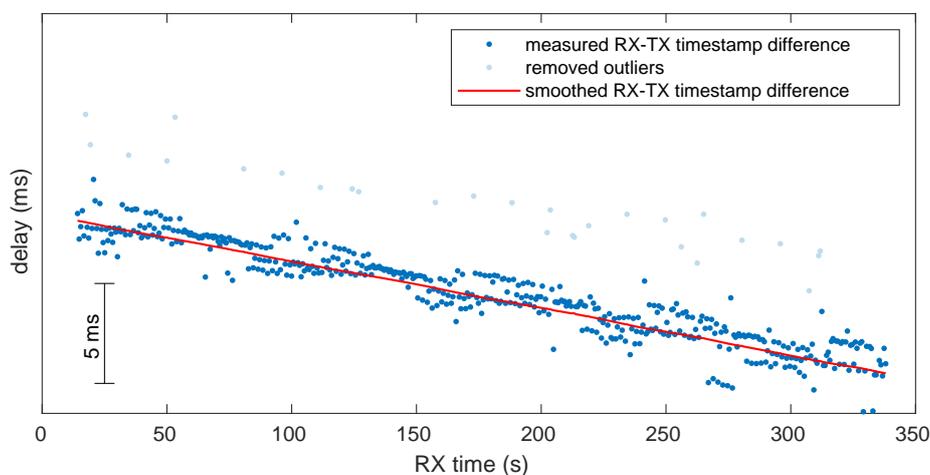


Figure 7. Example of measured delay between TX timestamp and RX timestamp. The graph is plotted with an arbitrary vertical offset because TX and RX clocks do not have a common starting epoch.

This difference allows to extract two essential pieces of information: the absolute difference d between transmit clock and real time (not shown in the figure because it varies randomly between runs, as the clock does not have a fixed starting epoch), and the clock drift $r \approx -\partial\delta/\partial t_{RX}$ (the slope of the interpolating curve).

Individual measures are inevitably noisy (RX software timestamping can not be very accurate), so, after removing outliers, they are cleaned by a double exponential smoothing filter that aims to approximate in real time the true regression line, as follows:

$$d[i] = k_d (d[j] - (t_{RX}[i] - t_{RX}[j]) r) + (1 - k_d) \delta[i] \quad (3)$$

$$r[i] = k_r r[j] - (1 - k_r) \frac{d[i] - d[j]}{t_{RX}[i] - t_{RX}[j]} \quad (4)$$

where i and j denote the current and previous packet indices, respectively (only packets with timestamps are considered, of course, so i and j are in general not consecutive), and $k = 1 - 1/\tau$ are smoothing factors, τ being the corresponding filter time constant expressed in number of samples. The shown results have been obtained with $\tau_d = 256$ and $\tau_r = 8192$. Every sample for which $|\delta[i] - d[j]| > T_I/2$ is considered an outlier and not used in updating Equations (3)–(4).

A double exponential smoothing filter was employed to avoid the delays introduced by traditional low-pass filters when processing ramp-like signals, but for its proper operation an estimate of the drift is needed. Equation (4) will adjust and track slow drift variations, such as those due to crystal ageing and temperature variations, but with such a big τ_r it would take hours to set initially. The initial drift estimate is thus performed at calibration time by linear regression and stored within the system, so that (4) will only need to track small, slow changes.

The smoothed curve $d[i]$ provides the necessary information to convert the sampling timestamps as transmitted in subchannel 2 into the corresponding real-time timestamps, by inverting the relation $t_{RX}[i] \mapsto t_{RX}[i] - d[i]$. To ease the process, the received measures, which due to noise may not be uniformly sampled neither along the RX nor the TX time base, are first resampled to a uniform interval T_0 along the t_{TX} time scale using linear interpolation. Let us call $t_{RT}[n]$ the interpolated t_{RX} real time corresponding to transmit time $t_{TX} = n T_0$.

Using a uniform sampling time $T_0 = 1$ s makes the resampling process effectively neutral from a noise perspective, as shown in Figure 8, which reports samples of the $t_{TX} \mapsto t_{RX}$ relation both before (dots) and after (stars) resampling. To better highlight the curves, only the difference between the estimated real-time t_{RX} and the a posteriori computed regression line is shown.

The uniformly sampled points are then interpolated (or extrapolated in case of low-latency output ports) to associate the estimated sampling timestamp according to the receiver-side time base. The quality of this interpolation/extrapolation procedure clearly depends on the requested output latency. In the lowest latency ports, it is only possible to extrapolate the latest measure using the current drift estimate, leading to the horizontal tracts of the “staircase” line in Figure 8 (tracts appear horizontal instead of approximately 45° because the overall regression line had been subtracted for graphical purposes). Increasing the latency to above 1 s allows linear interpolation between the last two samples to be performed (red line). Higher latencies, lastly, allow more aggressive filtering to be performed. In this case we interpolated the data using Gaussian FIR low-pass filtering with impulse responses twice as long as the requested latency, so as to allow a high degree of smoothing while keeping the filter causal as required for real-time implementation, as follows.

Given a t_{AD} time to convert, the filter operates similarly to the double exponential filter, except that the drift estimate is already available, and the differences are filtered using a FIR filter instead of an IIR filter. Indeed, in this case it is not possible to directly filter the ramp-like signal, as the residual ringing of the filter would then increase in amplitude with time, eventually exceeding the amount of wander it was supposed to reduce in the first place.

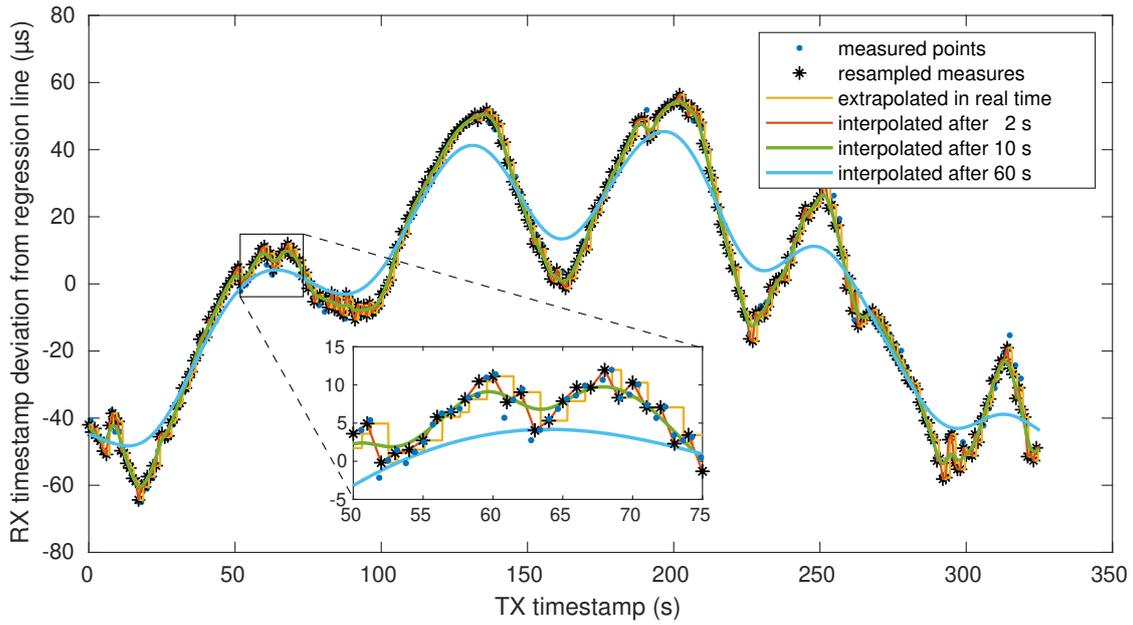


Figure 8. Example of the operation of the smoothing and interpolating filter used to convert the TX-side time scale to that of the receiver (assumed to match real time). Dots represent the measures out of the first-stage double exponential smoother, they are still somewhat noisy at microsecond scale ($1/F_S = 1250 \mu s$). Stars are the same measures resampled on a uniform time base using linear interpolation, so they possess the same amount of noise. Curves are the result of the final interpolation to match the sampling frequency F_S . Higher latencies allow for longer filter impulse responses and better interpolation/filtering. To make microsecond deviations graphically visible, the vertical axis shows differences between actual values and the ideal regression line computed after all data had been received.

Let $n = \lfloor t_{AD}/T_0 \rfloor$ be the “center” point of the FIR filter, then, letting $T_D = T_0/(1+r)$ be the real-time equivalent of the interval T_0 , estimated using the current drift estimate r , the difference between t_{RT} and the interpolating straight line can be computed as

$$e[i] = t_{RT}[i] - (t_{RT}[n] + (i - n)T_D) \tag{5}$$

with i spanning the interval $n - L, \dots, n + L$. This difference $e[i]$ can then be filtered by convolution with the Gaussian impulse response $g(\tau)$

$$e = \sum_{i=-L}^L e[i] g(t_{AD} - (i + n) T_0) \tag{6}$$

and then the final RX-side time estimate t_{AD}^* obtained by re-adding the straight line

$$t_{AD}^* = e + t_{RT}[n] + (t_{AD}/T_0 - n)T_D \tag{7}$$

The Gaussian impulse response is defined as

$$g(\tau) = e^{-(4\tau/LT_0)^2} w(\tau) \tag{8}$$

where $w(\tau)$ is a suitable raised-cosine window with a small roll-off factor $\beta = 1/4$ employed to reduce the effects of truncation errors

$$w(\tau) = \begin{cases} 0 & |\tau| \geq L T_0 \\ 1 & |\tau| < (L - \beta) T_0 \\ \frac{1}{2} - \frac{1}{2} \cos\left(\pi \frac{|\tau| - L T_0}{\beta T_0}\right) & \text{otherwise} \end{cases} \quad (9)$$

and was chosen because the Gaussian filter limits ringing in its step response, helping in reducing the residual wander. As can be seen from the figure, jitter is also effectively completely eliminated by the Gaussian filters. Of course, in order for (6) to be computable, the output port latency t_p must satisfy $t_p > L T_D$, otherwise some samples of $t_{RT}[i]$ might not already be available. Actually, even with proper latency, some of these will not be available near the beginning of the transmission, and after it is stopped when data buffers are flushed to the output ports with no other incoming data. During these transients, to allow the system to operate, missing samples are extrapolated using the current drift estimate as usual.

Lastly, to obtain the proper t_{AD} to transform, a relationship between packet index (or sample number), and sampling timestamp, must be derived. In the case of the EMG data, the relationship between sequence number and timestamp is fixed, since both the clock used for timestamping and for sampling the data is derived from the same HFXO. This allows an accurate signal alignment to be made at the receiver side even in case of packet loss and clock drifts between central and peripheral, because sampling timestamps can be computed exactly from packet index and the initial offset transmitted in subchannel 2. For GYR data, on the other hand, there is not an exact linear relationship between packet index and sampling time, because sampling is governed by an internal clock within the sensor chip, that is not externally synchronizable. For them, sampling timestamps can indeed be accurately estimated using the same interpolation procedure described above from the metadata always transmitted in subchannel 2. Given the typically lower data rates of GYR data with respect to EMG data, a slightly higher synchronization error is usually not perceivable.

5. Results

In this section the operation and performance of the prototype built will be analysed both with artificial signals, to characterize the system frequency response, noise floor, overall latency and synchronization, and power consumption, and with natural signals, to demonstrate its operation in a real scenario.

5.1. Frequency Response

The system frequency response was determined using a sinusoidal function generator to produce frequencies from 0.1 Hz to 1 kHz, using three discrete steps per decade at an amplitude of 5 mV. The signal was then acquired in the digital domain via the bluetooth connection, and digitally demodulated by means of a Hilbert transform to determine the exact frequency and amplitude for each step.

Results are shown in Figure 9 for the three different data rates. As can be seen, the passband is almost exactly flat, and the signal bandwidth, which can be estimated by computing the intersections of the interpolating curves to the -3 dB line, is approximately 1/5 of the output data rate, namely 159 Hz, 318 Hz, and 633 Hz respectively for 800 Hz, 1600 Hz, and 3200 Hz ODRs.

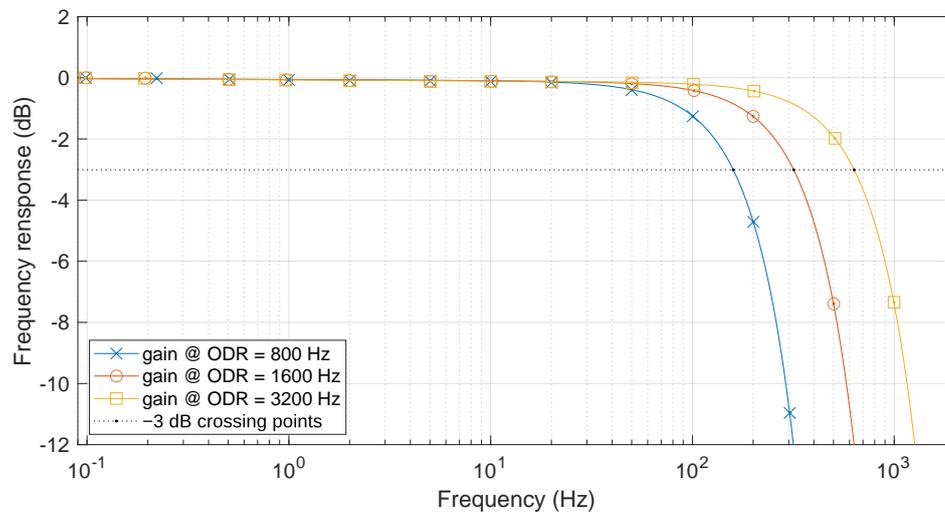


Figure 9. Measured frequency response of the EMG signal path, with the output data rate set to 800 Hz, 1600 Hz, and 3200 Hz (markers denote measured data points, solid lines are interpolated).

5.2. Noise Floor

Figure 10 reports the power spectral density of the noise floor, measured connecting the inputs of an EMG channel to a reference voltage source. The RMS noise is 2.1 μV , 2.9 μV , and 4.3 μV for data rates of 800 Hz, 1600 Hz, and 3200 Hz, respectively. For comparison, the power spectral density of a typical EMG signal recorded from the biceps brachii during a dumbbell exercise (see next section for details) is also shown in black on the same figure. It is worth noticing that there are no line frequency disturbances, as all precautions have been taken in the design of the circuit board to balance input lines to avoid common-mode to differential conversion.

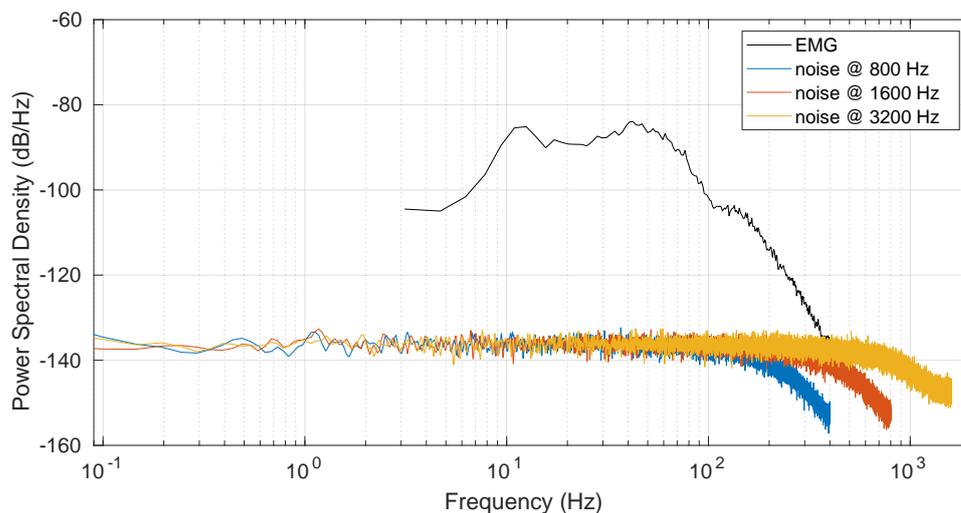


Figure 10. Noise floor, for different output data rates, compared with the power spectrum of a typical EMG signal recorded on the biceps brachii.

5.3. Latency and Synchronization

In some applications involving EMG sensors to detect muscle activity and then operate in the environment, or assist the patient, latency may be of concern.

To evaluate the system performance in this regard, a synthetic signal consisting of a pulse train, with 10,000 pulses spaced by 29.9 ms (about 33.4448 Hz), and accurately synchronized to the clock of the receiver (with a measured average synchronization error of just 154 ns), was generated. The pulse spacing was selected to not be a multiple of the connection interval, so that pulses occur at random times with respect to the BLE timing. At the receiver, each received packet is timestamped and the instant at which the signal crosses a threshold (set at mid-height of the pulse) is interpolated from these time stamps. The latency so defined thus includes all the effects of the amplifiers, converters, digital filters, buffering, transmission, and reception.

The distribution of the observed latencies is shown in Figure 11, the average being 8.8 ms. Please note that the ADS1293 incorporates a three-stage fifth-order sinc digital filter (sinc^5), and a sinc^5 filter at an 800 Hz data rate is on average responsible for a 2.5 ms latency.

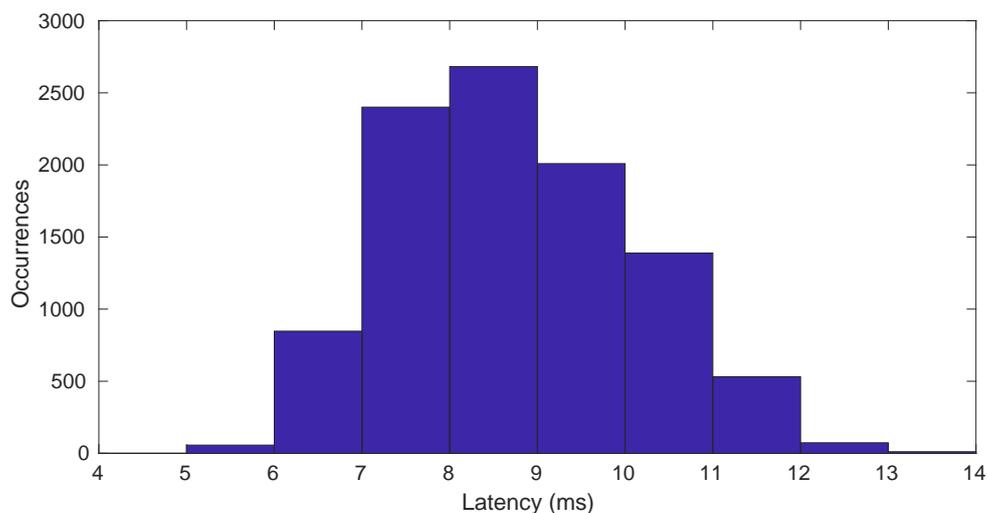


Figure 11. Histogram of the measured latency of the EMG signal path, with the output data rate set to 800 Hz, tested using 10,000 occurrences. The bluetooth connection interval was set to the minimum allowed by the specifications, i.e., 7.5 ms.

Of course, data in the above referenced figure reflects what can be achieved from a “real-time” (zero latency) output port, and represent the elapsed time from the physical event to when data is actually available. Timestamps associated to such data are still compensated for transmission delays and can be used to precisely locate events in time, exploiting the clock synchronization algorithm detailed above.

To demonstrate this a similar experiment was performed, wherein 10 bursts of 100 impulses, each 10 ms wide, were applied at random times t_j , and the measured signal plotted using $t_{AD}^* - t_j$ as the horizontal axis. An ensemble showing one impulse for each burst is reported in Figure 12. The delay from the input edge to the mid-swing crossing, averaged over all 1000 realizations, is 2.53 ms, and the RMS timing error is 6.6 μs .

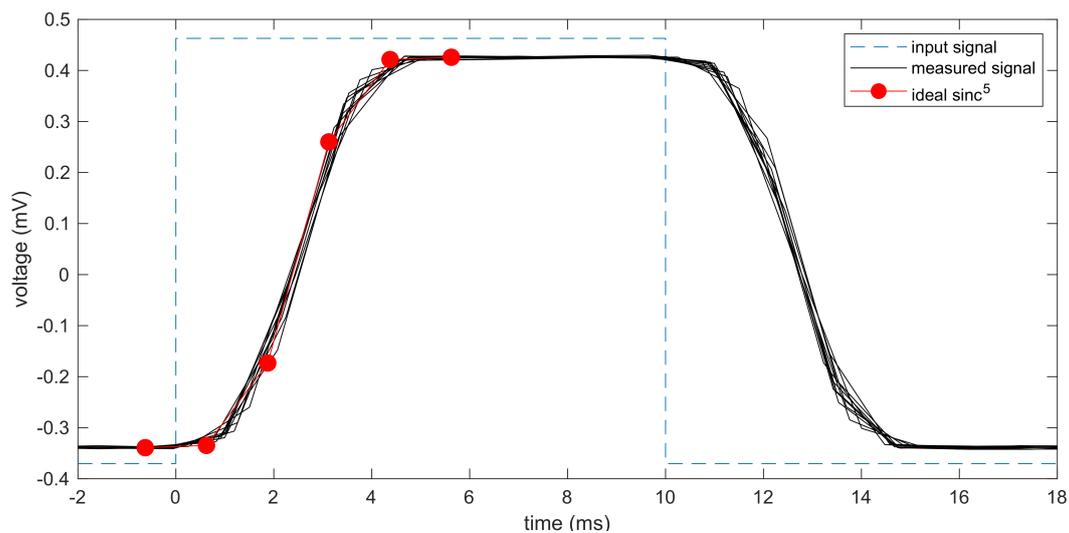


Figure 12. Measured output from the EMG signal chain corresponding to a rectangular impulse digitally provided (and hence shown not to scale). An ensemble of ten different measures across a two-minute time span is reported to show clock synchronization effectiveness. For reference, the ideal step response of a sinc^5 digital filter like those used in the analog front-end is reported as obtained by a VHDL simulation. The ODR was set to 800 Hz and so samples are 1.25 ms apart.

Similarly, accurate synchronization is also achieved on the inertial platform. To test this, a stimulus which elicits simultaneous electrical and mechanical responses was applied by tapping the on-board EMG electrode connectors with a small, solid, conductive rod. The tap is meant to cause the board to rotate, thus the motion can be registered by the gyroscope, and the electrical contact is registered as a train of disturbances on the EMG channel as the rod makes contact and rebounds on the connector. Figure 13 shows the resulting signals, recorded after 15 min of system operation to allow ample time for a possible oscillator drift to manifest itself. Actually, the estimated drifts are such so as to cause more than a 5 s shift between the pulses, if uncorrected. After relocking, instead, the onsets on the two signals appear simultaneous as they should be.

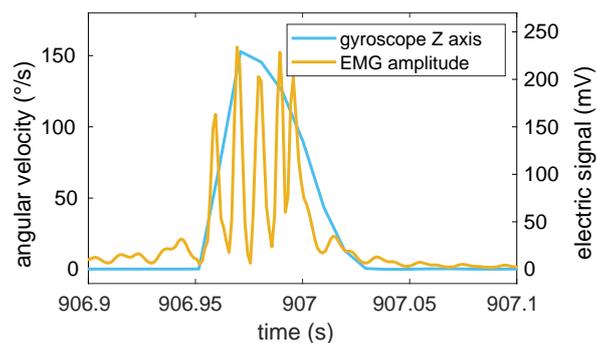


Figure 13. Example of the synchronization between the EMG and the gyroscope sensors. A stimulus causing simultaneous electrical and mechanical excitations of the device was applied after about 15 min of operation. Since the two sensors have independent clocks, without synchronization a drift of several seconds would have accumulated after 15 min. The relocked signals are instead synchronized to within sampling resolution.

Finally, it is of interest to analyse the effects of radio packet loss, and of the retransmission strategy adopted, on the system latency. To this end, Figure 14 shows the results of a simple “transmission quality” test performed over a 10 min time span. For the experiment, the sensor was kept within normal distance from the bluetooth receiver (a few meters) for the first 100 s. Then, the antenna was shielded to cause a surge in packet loss rate from about 100 s to 200 s. Then again, from 300 s to about

500 s the sensor was brought away from the receiver until an average 10 % TX loss rate was observed. A last move of the sensor at the edge of the useful radio range before bringing it back to normal distance caused the last increment in TX loss rate and the ramp in delay. It did not cause a large RX loss only because of its relatively short duration that did not cause the retransmission buffer to overflow.

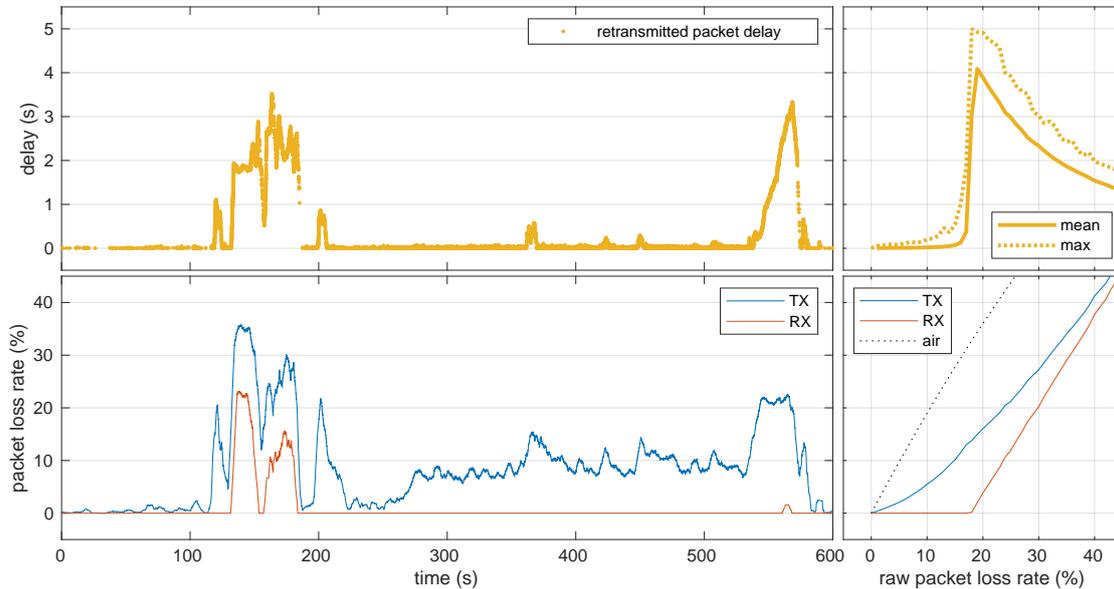


Figure 14. Results of a retransmission efficiency experiment (left panels) and of a steady-state system simulation (right panels). The yellow lines are the delays introduced by retransmissions. The blue lines (TX) are the average fraction of packets discarded by the transmitter due to radio channel congestion. The red lines (RX) are the average fraction of missing packets at the receiver. The dotted black line (air) denotes the air-time capacity reduction of the radio channel due to missed packets and connection events (see text for details).

In detail, the top left panel of the figure shows a dot for each retransmitted packet (from our retransmission buffer), each representing the delay between the time it should have been received and the actual reception time. The bottom left panel shows the percentage of the packets lost at the TX side (packets being discarded by our protocol because the TX queue is already full, line labeled “TX”), along with the percentage of packets still missed at the receiver after allowing enough time for retransmissions to eventually arrive (line labeled “RX”). The lines are moving averages over a 5 s span.

Since it is not easy to generate a steady state of transmission errors with a prescribed probability using the real hardware, results from a system-level simulation of the retransmission protocol are also shown on the right, as a function of the raw packet loss rate, so that they can be used as a reference. This was a simple simulation to verify the correctness of the protocol implementation and its effectiveness in mitigating radio channel congestion situations, and is by no means meant to be extremely accurate. Indeed, it was performed using the SystemC framework to model the behavior of the sensor node in queuing and transmitting packets, implementing the very same algorithm implemented in firmware as detailed earlier in this paper, while the radio channel was simulated using a very simple packet loss model, which assumes the same raw packet loss probability p both as the probability of missing a connection event altogether, and the probability of losing a data packet within the connection event, independently of packet length. Despite these simplifications, the model matches experimental results quite well, as can be seen by comparing the relative heights of the blue and red lines during the surge of errors in the experiment (left) and the simulation (right).

Moreover, as can be seen in the bottom right panel, which reports steady state averages for varying p , retransmissions effectively avoid losing data for $p < 17\%$. The “TX” curve is usually lower

than p because automatic link-layer retransmissions are not counted (nor are accessible by firmware with the used stack) and we are counting the packets “set aside” by the transmitter. As a reference, the curve $p + (1 - p)p$ is also plotted (labeled “air”), to denote the air-time capacity reduction due to the possibility of losing connection events. Since our protocol only retransmits one packet per connection event, and there are on average three fresh data packets per connection event with the settings adopted, only a 1/3 reduction in channel capacity can be recovered. Indeed, the “air” curve crosses the 1/3 level at the same p at which the “RX” curve starts to rise from zero.

Finally, in the top right panel the expected (“mean”) and maximum (“max”) value of the simulated delays are reported versus p . These can be useful to set the latency of the output ports. As can be seen, a 5 s latency allows for every retransmission to arrive, but 0.5 s is enough for $p < 15\%$.

5.4. Current Consumption

Table 4 reports the measured current consumed by the system (from the battery) in its different operational states. Of course, power consumption in the active states is strongly dependent on the firmware being run, and further reductions are expected by optimizing critical code paths so that the processor can spend more time sleeping. As it stands, with a small 250 mA battery, the sensor can be used to stream three channels continuously for over 2 days, and can sit in idle mode for over a year.

Table 4. Current consumption at $V_{BAT} = 4.0$ V and $V_{DD} = 1.8$ V.

Mode	I_{BAT}	Description
OFF	4.1 μ A	all devices off
IDLE	15.3 μ A	accelerometer enabled for wake-up
ADVERTISING	0.2 mA	bluetooth enabled (200 ms advertising interval)
CONNECTED	1.2 mA	all subsystems powered (7.5 ms connection interval)
STREAMING	4.6 mA	3 EMG channels streaming at 800 Hz (0 dBm TX power)

5.5. Real-Life Example

To test the operation and performance of the device in a real-life situation, a data recording session was performed in the laboratory with the device worn by the experimenter on the upper portion of the right arm while performing three different simple exercises commonly used during fitness training or rehabilitation, namely, biceps curls, lateral raises, and vertical raises. Of course, the experimenter signed a written informed consent form before performing the tests.

The EMG signals were recorded from the biceps brachii, deltoideus medium, and triceps brachii, using standard 26 mm Ag/AgCl pre-gelled ECG electrodes (model PG10C made by FIAB). They have a nominal impedance of 55 Ω at 10 Hz and manifest a very low offset voltage of only 2.1 mV. The positioning was chosen so that the three different exercises should result in three different, clearly identifiable, activation patterns of the muscle groups. The EMG signal was acquired at the default 800 Hz sampling rate, which happens to be the maximum at which three channels can simultaneously be streamed over BLE 4.0, and at this ODR the frontend has a resolution of about 0.16 μ V, with an integrated RMS noise of 2.1 μ V.

Inertial data was also simultaneously recorded from the sensor, at the default 104 Hz ODR, with a 500°/s gyroscope full-scale sensitivity and an 8 g accelerometer full-scale sensitivity. The sensor was secured to the arm with a hook-and-loop closure so as to limit spurious oscillations and vibrations of the device during movements. The setup is shown in Figure 15, together with a description of the orientation of the inertial sensor.

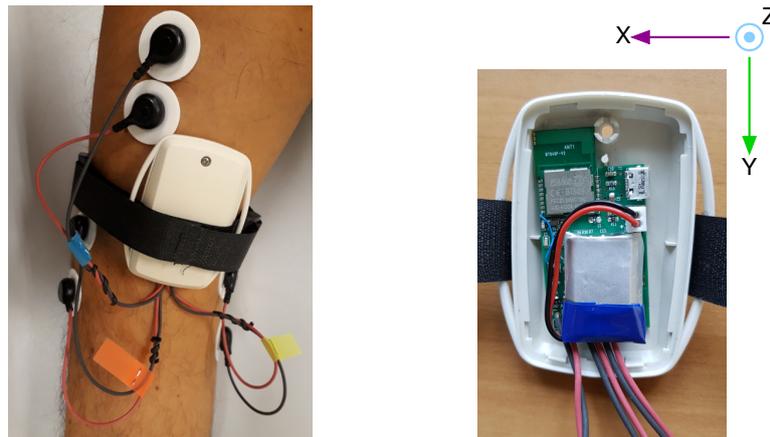


Figure 15. Positioning of the electrodes and device on the upper portion of the right arm. The electrodes identified by the yellow tab contact the biceps brachii, those identified with the orange tab the triceps brachii, and those with the blue tab the deltoideus medium. The orientation of the mechanical axes is visible in the right picture, showing the interior of the case with the battery on top of the circuit board. Once worn, axes are oriented so that the Y direction is along the arm, pointing downwards at rest, X points towards the front, parallel to the sagittal plane, and Z points towards the arm, along the coronal plane.

The acquisition was performed using an ordinary laptop PC as a bluetooth receiver, using the integrated wireless adapter compatible with the 4.0 specifications, and the custom-made software described above to record and synchronize the stream of incoming notifications.

The raw recorded EMG signals are shown in Figure 16 (left panel), from which it is possible to see that the biasing network operates as expected, as the signals are nearly exactly centered around the zero. Since the ADS1293 input stage can accept common mode voltages up to 0.95 V from the rails, there is ample headroom for possible DC wander induced by cable movements or skin stretch.

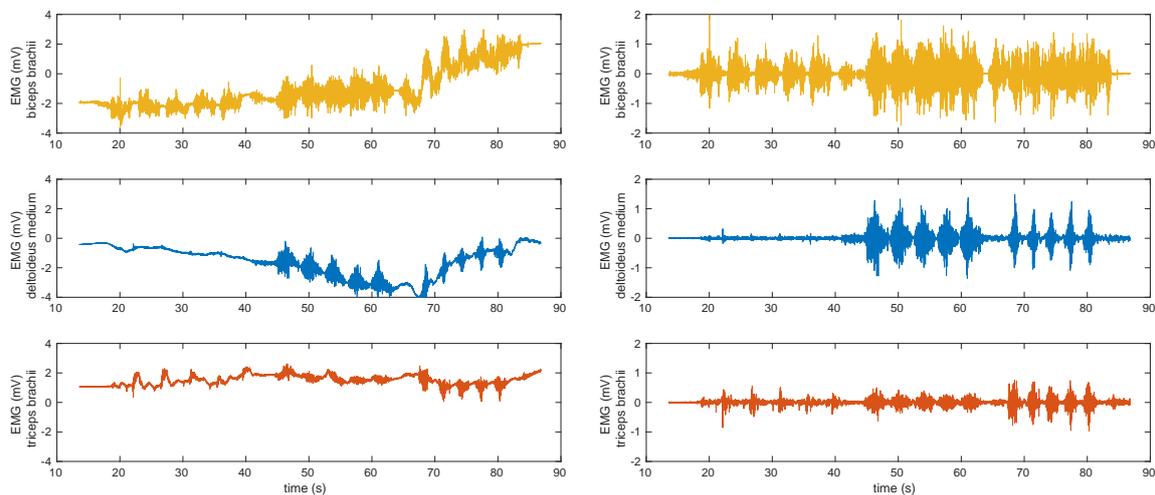


Figure 16. (Left panel) raw EMG signals acquired from the upper arm muscles, during three different exercises (5 reps each): biceps curls, lateral raises, vertical raises. (Right panel) the same signals after digital high-pass filtering to remove baseline wander.

Indeed, since the system shall also be used to record ECG signals, no analog high-pass filtering was included, as is instead customary to do for EMG signals. The high common-mode compliance and large differential-mode full-scale of the converter, combined with the biasing strategy, makes the analog filter unnecessary.

Yet, to better appreciate the EMG signal due to muscle contraction and remove the DC wander, a high-pass digital filter with a cut-off frequency of 1 Hz was applied to the recorded signals. The result is shown on the right panel of the same Figure 16, from which it is easier to assess muscle involvement during the three different exercises. As can be seen, during biceps curls, only the biceps brachii is involved, with just a minor activation of the triceps brachii for stabilization. All the three muscles are somewhat involved during the other exercises, with the deltoideus medium being exerted the most during lateral raises, as it should be.

The raw data coming from the inertial sensor is shown in Figure 17. The angular velocity (left panel) clearly shows that the upper arm was kept essentially at rest during the first exercise (biceps curls), and the concentric and eccentric phases of the other exercises can easily be identified by just looking at the sign of the velocity. Acceleration can also be used to infer the inclination of the arm, as easily seen in right panel of the same Figure 17.

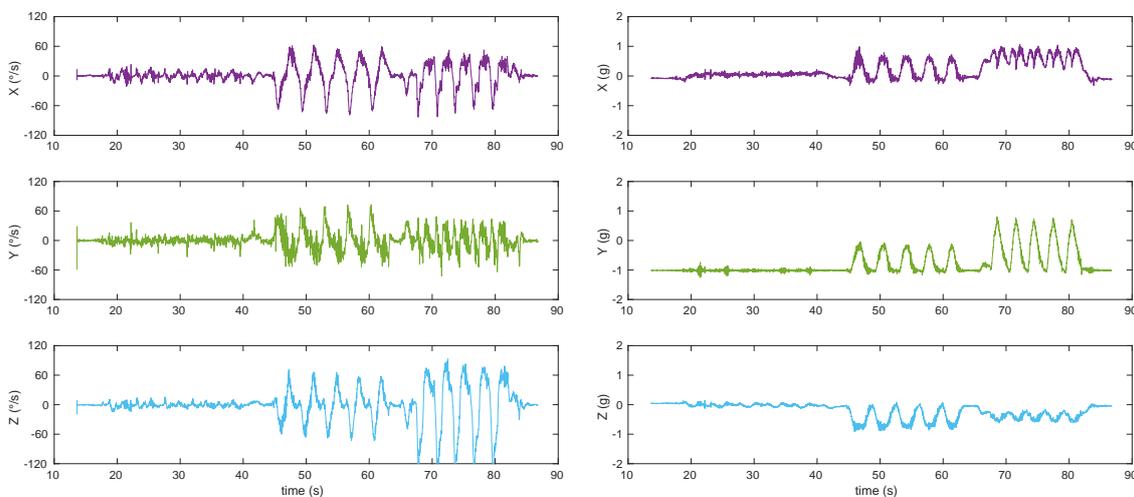


Figure 17. Signals recorded by the inertial platform during the exercises described in Figure 16. (Left panel) angular velocity as measured by the gyroscope. (Right panel) linear acceleration as measured by the inertial platform.

Of course, the two sets of data can be combined together to compute an orientation vector, possibly with the help of a magnetometer (not included in this first prototype), but this is outside the scope of the present experiment.

To complete the experimentation, the capability of the sensor to acquire an ECG signal was also tested. Figure 18 shows the unfiltered ECG signal as acquired using a two-leads configuration from the chest. As can be seen, the signal is quite clean and no visible artifacts are present.

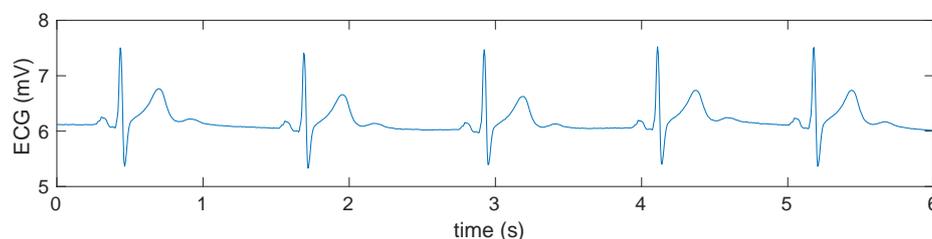


Figure 18. Raw ECG signal sampled at 200 Hz.

6. Conclusions

This work presented a wireless sensor able to acquire either EMG or ECG signals, with up to three channels, while simultaneously capturing movement information from an inertial platform. It uses a bluetooth low-energy radio for real-time data transmission so as to minimize energy consumption,

and a carefully designed transmission scheduling allowed to maximize the data throughput while keeping latency reasonably low, as demonstrated by extensive experimentation. This newly proposed adaptation layer on top of BLE also allows tight time synchronization, and provides reliability of the data transmission by ensuring lossless communication with up to 1/3 of the radio channel capacity compromised, which are all important features if the sensor is to be used to perform research or real-time monitoring. The acquired signal quality was assessed by both objective measurements, and evaluated in a real-life scenario, and it met design expectations.

The design of the whole system was driven by energy efficiency and ease of use as primary objectives. Indeed, using the motion-triggered wake-up capability provided by the accelerometer helped achieving both goals simultaneously, as the system just activates itself when worn, without needing any operator intervention. Plus, usage of a very widespread radio technology as bluetooth, allows the use of commodity hardware as smartphones or laptops as data receivers, further simplifying deployment.

Author Contributions: Conceptualization, G.B., P.C., L.F. and C.T.; Data curation, L.F.; Funding acquisition, P.C. and C.T.; Investigation, G.B., P.C., L.F. and C.T.; Project administration, P.C. and C.T.; Software, G.B.; Supervision, P.C. and C.T.; Validation, G.B. and L.F.; Visualization, G.B.; Writing—original draft, G.B. and P.C.; Writing—review & editing, G.B., P.C., L.F. and C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a Università Politecnica delle Marche Research Grant (Progetto Strategico di Ateneo 2017).

Conflicts of Interest: The authors declare no conflict of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Liu, S.; Liu, X.; Jiang, Y.; Wang, X.; Huang, P.; Wang, H.; Zhu, M.; Tan, J.; Li, P.; Lin, C.; et al. Flexible Non-contact Electrodes for Bioelectrical Signal Monitoring. In Proceedings of the 40th Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 18–21 July 2018; pp. 4305–4308.
2. Magno, M.; Benini, L.; Spagnol, C.; Popovici, E. Wearable low power dry surface wireless sensor node for healthcare monitoring application. In Proceedings of the 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Lyon, France, 7–9 October 2013; pp. 189–195.
3. McKhann, G.M.; Knopman, D.S.; Chertkow, H.; Hyman, B.T.; Jack, C.R.; Kawas, C.H.; Klunk, W.E.; Koroshetz, W.J.; Manly, J.J.; Mayeux, R.; et al. The diagnosis of dementia due to Alzheimer’s disease: Recommendations from the National Institute on Aging-Alzheimer’s Association workgroups on diagnostic guidelines for Alzheimer’s disease. *Alzheimer’s Dement.* **2011**, *7*, 263 – 269. [[CrossRef](#)]
4. Hazzan, A.A.; Ploeg, J.; Shannon, H.; Raina, P.; Oremus, M. Association between caregiver quality of life and the care provided to persons with Alzheimer’s disease: Protocol for a systematic review. *Syst. Rev.* **2013**, *2*, 17. [[CrossRef](#)]
5. Vuong, N.K.; Chan, S.; Lau, C.T.; Chan, S.Y.W.; Yap, P.L.K.; Chen, A.S.H. Preliminary results of using inertial sensors to detect dementia-related wandering patterns. In Proceedings of the 37th Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 25–29 August 2015; pp. 3703–3706.
6. Sánchez, D.; Tentori, M.; Favela, J. Activity Recognition for the Smart Hospital. *IEEE Intell. Syst.* **2008**, *23*, 50–57. [[CrossRef](#)]
7. Biagetti, G.; Crippa, P.; Falaschetti, L.; Luzzi, S.; Turchetti, C. Recognition of Daily Human Activities Using Accelerometer and sEMG Signals. In *Intelligent Decision Technologies 2019*; Czarnowski, I.; Howlett, R.J.; Jain, L.C., Eds.; Springer: Singapore, 2019; pp. 37–47.
8. Ghasemzadeh, H.; Jafari, R.; Prabhakaran, B. A Body Sensor Network With Electromyogram and Inertial Sensors: Multimodal Interpretation of Muscular Activities. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 198–206. [[CrossRef](#)] [[PubMed](#)]

9. Nawab, S.H.; Roy, S.H.; Luca, C.J.D. Functional activity monitoring from wearable sensor data. In Proceedings of the 26th Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC), San Francisco, CA, USA, 1–4 September 2004; pp. 979–982.
10. Spulber, I.; Georgiou, P.; Eftekhar, A.; Toumazou, C.; Duffell, L.; Bergmann, J.; McGregor, A.; Mehta, T.; Hernandez, M.; Burdett, A. Frequency analysis of wireless accelerometer and EMG sensors data: Towards discrimination of normal and asymmetric walking pattern. In Proceedings of the 2012 IEEE International Symposium on Circuits and Systems (ISCAS), Seoul, Korea, 20–23 May 2012; pp. 2645–2648.
11. Roy, S.H.; Cheng, M.S.; Chang, S.S.; Moore, J.; Luca, G.D.; Nawab, S.H.; Luca, C.J.D. A Combined sEMG and Accelerometer System for Monitoring Functional Activity in Stroke. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2009**, *17*, 585–594. [[CrossRef](#)] [[PubMed](#)]
12. Wang, Q.; Chen, X.; Chen, R.; Chen, Y.; Zhang, X. Electromyography-Based Locomotion Pattern Recognition and Personal Positioning Toward Improved Context-Awareness Applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *43*, 1216–1227. [[CrossRef](#)]
13. Zhang, X.; Chen, X.; Li, Y.; Lantz, V.; Wang, K.; Yang, J. A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. *IEEE Trans. Syst. Man, Cybern. Part A Syst. Humans* **2011**, *41*, 1064–1076. [[CrossRef](#)]
14. Tosi, J.; Taffoni, F.; Santacatterina, M.; Sannino, R.; Formica, D. Performance Evaluation of Bluetooth Low Energy: A Systematic Review. *Sensors* **2017**, *17*, 2898. [[CrossRef](#)]
15. Wang, Y.Z.; Wang, Y.P.; Wu, Y.C.; Yang, C.H. A 12.6 mW, 573–2901 kS/s Reconfigurable Processor for Reconstruction of Compressively Sensed Physiological Signals. *IEEE J. Solid-State Circuits* **2019**, *54*, 2907–2916. [[CrossRef](#)]
16. Tu, Y.H.; Yao, K.W.; Huang, M.H.; Lin, Y.Y.; Chi, H.Y.; Cheng, P.M.; Tsai, P.Y.; Shiue, M.T.; Liu, C.N.; Cheng, K.H.; Fu, J.S. A body sensor node SoC for ECG/EMG applications with compressed sensing and wireless powering. In Proceedings of the 2017 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, 24–27 April 2017.
17. Karlsson, J.S.; Wiklund, U.; Berglin, L.; Östlund, M.; Karlsson, M.; Bäcklund, T.; Lindecrantz, K.; Sandsjö, L. Wireless monitoring of heart rate and electromyographic signals using a smart T-shirt. In Proceedings of International Workshop on Wearable Micro and Nanosystems for Personalised Health, Valencia, Spain, 21–23 May 2008.
18. Dow, H.K.; Huang, I.J.; Rieger, R.; Kuo, K.C.; Guo, L.Y.; Pao, S.J. A Bio-Sensing System-on-Chip and Software for Smart Clothes. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019.
19. Biagetti, G.; Crippa, P.; Orcioni, S.; Turchetti, C. Surface EMG Fatigue Analysis by Means of Homomorphic Deconvolution. In *Mobile Networks for Biometric Data Analysis*; Springer International Publishing: Cham, Switzerland, 2016; pp. 173–188.
20. Mukhopadhyay, S.C. Wearable Sensors for Human Activity Monitoring: A Review. *IEEE Sens. J.* **2015**, *15*, 1321–1330. [[CrossRef](#)]
21. Crippa, P.; Curzi, A.; Falaschetti, L.; Turchetti, C. Multi-Class ECG Beat Classification Based on a Gaussian Mixture Model of Karhunen-Loève Transform. *Int. J. Simul. Sci. Technol.* **2015**, *16*, 2.1–2.10.
22. Brunelli, D.; Tadesse, A.M.; Vodermayr, B.; Nowak, M.; Castellini, C. Low-cost wearable multichannel surface EMG acquisition for prosthetic hand control. In Proceedings of the 6th Int. Workshop on Advances in Sensors and Interfaces, Gallipoli, Italy, 18–19 June 2015; pp. 94–99.
23. Biagetti, G.; Crippa, P.; Curzi, A.; Orcioni, S.; Turchetti, C. A Multi-class ECG Beat Classifier Based on the Truncated KLT Representation. In Proceedings of the 2014 European Modelling Symposium, Pisa, Italy, 21–23 October 2014; pp. 93–98.
24. Castroflorio, T.; Mesin, L.; Tartaglia, G.; Sforza, C.; Farina, D. Use of Electromyographic and Electrocardiographic Signals to Detect Sleep Bruxism Episodes in a Natural Environment. *IEEE J. Biomed. Health Inform.* **2013**, *17*, 994–1001. [[CrossRef](#)] [[PubMed](#)]
25. Chou, C.C.; Tseng, S.Y.; Chua, E.; Lee, Y.C.; Fang, W.C.; Huang, H.C. Advanced ECG processor with HRV analysis for real-time portable health monitoring. In Proceedings of the 2011 IEEE International Conference on Consumer Electronics, Berlin, Germany, 6–8 September 2011; pp. 172–175.

26. Lee, S.Y.; Koo, K.H.; Lee, Y.; Lee, J.H.; Kim, J.H. Spatiotemporal analysis of EMG signals for muscle rehabilitation monitoring system. In Proceedings of the 2013 IEEE 2nd Global Conference on Consumer Electronics, Tokyo, Japan, 1–4 October 2013; pp. 1–2.
27. Biagetti, G.; Crippa, P.; Falaschetti, L.; Orcioni, S.; Turchetti, C. A Rule Based Framework for Smart Training Using sEMG Signal. In *Intelligent Decision Technologies; Smart Innovation, Systems and Technologies*; Springer International Publishing: Cham, Switzerland, 2015; pp. 89–99.
28. Chang, K.M.; Liu, S.H.; Wu, X.H. A wireless sEMG recording system and its application to muscle fatigue detection. *Sensors* **2012**, *12*, 489–499. [[CrossRef](#)] [[PubMed](#)]
29. Fukuda, T.Y.; Echeimberg, J.O.; Pompeu, J.E.; Lucareli, P.R.G.; Garbelotti, S.; Gimenes, R.; Apolinário, A. Root mean square value of the electromyographic signal in the isometric torque of the quadriceps, hamstrings and brachial biceps muscles in female subjects. *J. Appl. Res.* **2010**, *10*, 32–39.
30. Pantelopoulos, A.; Bourbakis, N. A survey on wearable biosensor systems for health monitoring. In Proceedings of the 30th Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC), Vancouver, BC, Canada, 20–24 August 2008; pp. 4887–4890.
31. Oh, J.; Kwon, M.; Kim, Y.; Kim, J.; Lee, S.; Kim, J. Development and evaluation of myoelectric driving interface. In Proceedings of the 2013 IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 11–14 January 2013; pp. 248–249.
32. Son, J.; Kim, B.; Park, M. Lumbar cushion based real-time ECG sensing system for monitoring driver's state. In Proceedings of the 2015 IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 9–12 January 2015; pp. 261–262.
33. Burns, A.; Doheny, E.P.; Greene, B.R.; Foran, T.; Leahy, D.; O'Donovan, K.; McGrath, M.J. SHIMMER: An extensible platform for physiological signal capture. In Proceedings of the 2010 Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC), Buenos Aires, Argentina, 31 August–4 September 2010; pp. 3759–3762.
34. Kobayashi, H. EMG/ECG acquisition system with online adjustable parameters using ZigBee wireless technology. *Electron. Commun. Jpn.* **2013**, *96*, 1–10. [[CrossRef](#)]
35. Biagetti, G.; Crippa, P.; Falaschetti, L.; Orcioni, S.; Turchetti, C. Wireless surface electromyograph and electrocardiograph system on 802.15.4. *IEEE Trans. Consum. Electron.* **2016**, *62*, 258–266. [[CrossRef](#)]
36. Naranjo-Hernández, D.; Roa, L.M.; Reina-Tosina, J.; Estudillo-Valderrama, M.A. SoM: A Smart Sensor for Human Activity Monitoring and Assisted Healthy Ageing. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 3177–3184. [[CrossRef](#)]
37. Rodriguez-Martin, D.; Samà, A.; Perez-Lopez, C.; Català, A.; Cabestany, J.; Rodriguez-Molinero, A. SVM-based posture identification with a single waist-located triaxial accelerometer. *Expert Syst. Appl.* **2013**, *40*, 7203–7211. [[CrossRef](#)]
38. Mannini, A.; Intille, S.S.; Rosenberger, M.; Sabatini, A.M.; Haskell, W. Activity recognition using a single accelerometer placed at the wrist or ankle. *Med. Sci. Sport. Exerc.* **2013**, *45*, 2193–2203. [[CrossRef](#)]
39. Torres-Huitzil, C.; Nuno-Maganda, M. Robust smartphone-based human activity recognition using a tri-axial accelerometer. In Proceedings of the 2015 IEEE 6th Latin American Symposium on Circuits Systems, Montevideo, Uruguay, 24–27 February 2015; pp. 1–4.
40. Miao, F.; He, Y.; Liu, J.; Li, Y.; Ayoola, I. Identifying typical physical activity on smartphone with varying positions and orientations. *BioMedical Eng. Online* **2015**, *14*, 32. [[CrossRef](#)]
41. Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; Reyes-Ortiz, J.L. Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic. *J. Univers. Comput. Sci.* **2013**, *19*, 1295–1314.
42. Dernbach, S.; Das, B.; Krishnan, N.C.; Thomas, B.L.; Cook, D.J. Simple and Complex Activity Recognition through Smart Phones. In Proceedings of the 8th International Conference on Intelligent Environments, Guanajuato, Mexico, 26–29 June 2012; pp. 214–221.
43. Khan, A.M.; Lee, Y.K.; Lee, S.Y.; Kim, T.S. Human Activity Recognition via an Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis. In Proceedings of the 2010 5th International Conference on Future Information Technology, Busan, Korea, 21–23 May 2010; pp. 1–6.
44. Biagetti, G.; Crippa, P.; Falaschetti, L.; Orcioni, S.; Turchetti, C. An Efficient Technique for Real-Time Human Activity Classification Using Accelerometer Data. In *Intelligent Decision Technologies*; Springer International Publishing: Cham, Switzerland, 2016; pp. 425–434.

45. Mannini, A.; Sabatini, A.M. Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. *Sensors* **2010**, *10*, 1154–1175. [[CrossRef](#)]
46. Catal, C.; Tufekci, S.; Pirmir, E.; Kocabag, G. On the use of ensemble of classifiers for accelerometer-based activity recognition. *Appl. Soft Comput.* **2015**, *37*, 1018–1022. [[CrossRef](#)]
47. De Luca, C.J.; Donald Gilmore, L.; Kuznetsov, M.; Roy, S.H. Filtering the surface EMG signal: Movement artifact and baseline noise contamination. *J. Biomech.* **2010**, *43*, 1573–1579. [[CrossRef](#)] [[PubMed](#)]
48. Trobec, R.; Tomašić, I.; Rashkovska, A.; Depolli, M.; Avbelj, V., From Multichannel ECG to Wireless Body Sensors. In *Body Sensors and Electrocardiography*; Springer International Publishing: Cham, Switzerland, 2018; pp. 13–35.
49. Gjoreski, H.; Rashkovska, A.; Kozina, S.; Lustrek, M.; Gams, M. Telehealth using ECG sensor and accelerometer. In Proceedings of the 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 26–30 May 2014; pp. 270–274.
50. Pinto, A.G.; Dias, G.; Felizardo, V.; Pombo, N.; Silva, H.; Fazendeiro, P.; Crisóstomo, R.; Garcia, N. Electrocardiography, electromyography, and accelerometry signals collected with BITalino while swimming: Device assembly and preliminary results. In Proceedings of the 2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 8–10 September 2016; pp. 37–41.
51. Said, S.; AlKork, S.; Beyrouthy, T.; Abdrabbo, M.F. Wearable bio-sensors bracelet for driver's health emergency detection. In Proceedings of the 2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART), Paris, France, 30 August–1 September 2017; pp. 1–4.
52. Tanweer, M.; Halonen, K.A.I. Development of wearable hardware platform to measure the ECG and EMG with IMU to detect motion artifacts. In Proceedings of the 2019 IEEE 22nd International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS), Cluj-Napoca, Romania, 24–26 April 2019; pp. 1–4.
53. Das, D.M.; Vidwans, A.; Srivastava, A.; Ahmad, M.; Vaishnav, S.; Dewan, S.; Baghini, M.S. Design and development of an Internet-of-Things enabled wearable ExG measuring system with a novel signal processing algorithm for electrocardiogram. *IET Circuits Devices Syst.* **2019**, *13*, 903–907. [[CrossRef](#)]



© 2020 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/>).