


JOINT MASTER'S DEGREE IN
NANOTECHNOLOGIES FOR ICTS

ACADEMIC YEAR 2018/2019

HELD BY:
POLITECNICO DI TORINO
GRENOBLE INP-PHELMA
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

SoC Configuration for Real-Time EMG Sensing and Wireless Control of a Robotic Prosthesis

Author:
Costantino COSENTINO

Supervisor:
Prof. Guido MASERA

External Supervisor:
Dr. Marc PONS

Co-supervisor:
Prof. Carlo RICCIARDI

COMPANY: CENTRE SUISSE D'ÉLECTRONIQUE
ET DE MICROTECHNIQUE (CSEM)

Acknowledgements

I would like to express my gratitude to Dr. Marc Pons for giving me the opportunity of working on such an ambitious project, entrusting me with important tasks that helped my professional growth and for adding my name to a research paper linked to this project. I am grateful to Mr. Stéphane Emery for allowing me to work in a stimulating and extremely qualified environment such as the System on Chip sector of CSEM. I would also like to thank Dr. Lorenzo Bergamini for the precious help and important feedback provided.

I would like to thank the Integrated & Wireless Systems division of CSEM: I had the possibility to work with experts in several fields thanks to whom I broadened my spectrum of knowledge and I received brilliant ideas for the development of my thesis. I would also like to extend my gratitude to Prof. Guido Masera, for his availability and kindness towards me.

To my little brother Andrea and my parents, Marina and Giovanni, for their endless endorsement during these years of studying.
To all the friends who shared these last years with me, for their warming support and enjoyable moments.

Last in order but not of importance, special thanks to Alessandra, for always being present and patient, above all in the period I spent abroad.

Lausanne, August 31, 2019 C. C.

Abstract

The field of the robotic prosthesis didn't progress for decades after its debut in the 60's, mainly because of the limits related to the kind of electrodes used and the lack of sensory feedback. This framework and the recent invention of an innovative technique for integrating the prosthesis inside the bones of a patient led to the creation of a very promising European project renamed DeTOP. Its main purpose is to develop a groundbreaking robotic prosthesis controlled by implanted electrodes and able to offer sensory feedback to the patient.

Thanks to its longstanding expertise in electronics, the CSEM contributed to this project through the development of a state of the art System on Chip able to sense the electric signals collected by the implanted electrodes and generated by the muscular contraction for controlling the robotic hand in a power efficient way. In addition, the device can control an external radio used for the wireless monitoring of the patient.

This thesis presents the work carried out on the previously mentioned System on Chip. Even though it was already designed and manufactured during the beginning of the experience, the hardware wasn't totally validated and the software for the final purpose of this chip was not fully developed. It is possible to divide this thesis into two main targets: the first one is related to create software for the robotic hand control through sensing of electromyographic signals while the second deals with a two-way wireless communication between the prosthesis and a computer.

Contents

List of Figures

List of Tables

Abbreviations

ADC	Analog to Digital Converter
AFE	Analog Front End
ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal Oxide Semiconductor
DETOP	Dexterous Transradial Osseointegrated Prosthesis
ECG	Electrocardiography/Electrocardiographic
ELF	Executable and Linkable Format
EMG	Electromyograpy/Electromyographic
FFT	Fast Fourier Transform
FIFO	First In First Out
GPIO	General Purpose Input Output
HPF	High Pass Filter
HW	Hardware
I^2C	Inter Integrated Circuit
IOT	Internet Of Things
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JTAG	Joint Test Action Group
LLC	Logical Link Control
LPF	Low Pass Filter
LNA	Low-Noise Amplifier
MAC	Media Access Control
MISO	Master Input Slave Output
MOSI	Master Output Slave Input
MPC	Miniature Processing and Communication
MUAP	Motor Unit Action Potential
MUX	Multiplexer
NSS	Not Slave Select
NVM	Non Volatile Memory
OHMG	Osseointegrated Human-Machine Gateway

OSI	Open Systems Interconnection
PCB	Printed Circuit Board
PGA	Programmable-Gain Amplifier
RAM	Random Access Memory
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
SAR	Successive Approximation
SOC	System On Chip
SPI	Serial Peripheral Interface
SW	Software
TDMA	Time Division Multiple Access
UART	Universal Asynchronous Receiver-Transmitter
WSN	Wireless Sensor Network

1 Introduction

Since its introduction in the 1960s, the technology related to the limb prosthesis did not have a remarkable improvement, mainly because of the electrodes applied to the patient, of the cumbersome and annoying socket used for the prosthesis support and for the lack of sensory feedback.
The pursuit of a direct prosthetic interface with the neuromuscular system using implanted electrodes went on for decades, facing different failures mainly caused by infections occurred a short time after the implantation. Recently, a new technique has been developed for the direct integration of the prosthesis to the bones of the patients, the osseointegration (1), showing some successful results over 92% of 51 patients with transfemoral amputation 2 years after the implantation (2).
The realization of an osseointegrated system able to implant electrodes inside the muscles and on the nerves of a patient to have both control and sensory feedback from a prosthesis (3) was the initial spark that led to the beginning of a European project (DeTOP) of which this thesis is a small fraction.

1.1 DeTOP project

The DeTOP (Dexterous Transradial Osseointegrated Prosthesis with neural control and sensory feedback) is a research project funded in the framework of the Horizon

2020 program for research and innovation (4). Its purpose is to develop and implement a novel prosthetic hand with improved functionality such as an intuitive control of the prosthesis and sensory feedback addressed to the recovery of hand function for patients with upper limb amputations.
The actual commercial technologies are based on myoelectric prosthesis developed 40 years ago and driven by surface electrodes. Multiple problems are associated with this kind of apparatus: motion artifacts due to the bad quality of the control signals recorded on the skin surface, limits in the movements for a patient, discomfort due to the socket in contact with the skin.
The core of innovation of DeTOP stands in a new technique for the direct skeletal attachment of the limb prostheses, the osseointegration (2), able to produce an efficient mechanical coupling and, above all, a bidirectional communication interface between the patient and the robotic hand, able to give both controls of the prosthesis and sensory feedback. As shown in figure 1.1, this new implant is able to provide a wider range of movements to the patients and, by avoiding the socket attached to the shoulder, it is possible to wear it for a prolonged amount of time on a daily basis.
Everything has to be encapsulated in smart devices for safe implantable technologies, able to be functional in the long term. The robotic arm will be also equipped with a wireless communication system that will allow constant monitoring over the patient through smart-phones or PC. A global view of the whole system is shown in figure 1.2.
The most interesting and complex aspect of this project consists in the combination of several scientific fields (robotics, electronics, surgery, etc.): this is why the most advanced European research institutions and ambitious companies decided to join this program. More specifically, the the ten partners involved are the following:

- Scuola Superiore Sant'anna, Pisa, Italy.
- University of Gothenburg, Gothenburg, Sweden.
- Prensilia SRL, Pontedera, Italy.

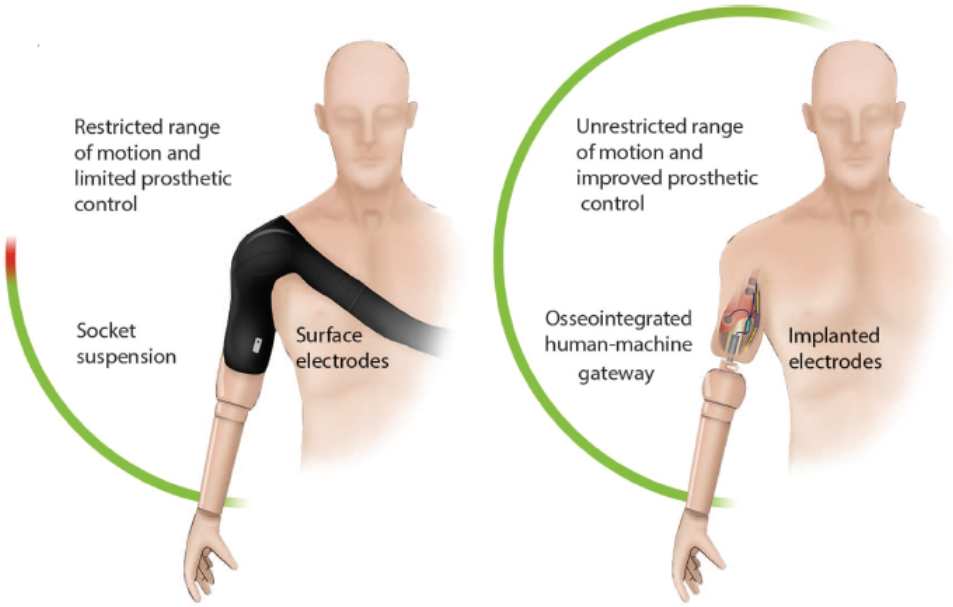


Figure 1.1 – A comparison between a prosthesis driven by surface electrodes (on the left) and by implanted electrodes connected to an osseointegrated implant (right). Adapted from (2).

- Integrum AB, Gothenburg, Sweden.
- Lund University, Lund, Sweden.
- CSEM SA, Neuchâtel, Switzerland.
- University of Essex, Colchester, United Kingdom.
- INAIL (National Workers' Compensation) Prosthetic Center, Bologna, Italy.
- Università Campus Bio-Medico, Roma, Italy.
- Istituto Ortopedico Rizzoli (IOR), Bologna, Italy.

This European project started in March 2016 and it will finish in February 2020.

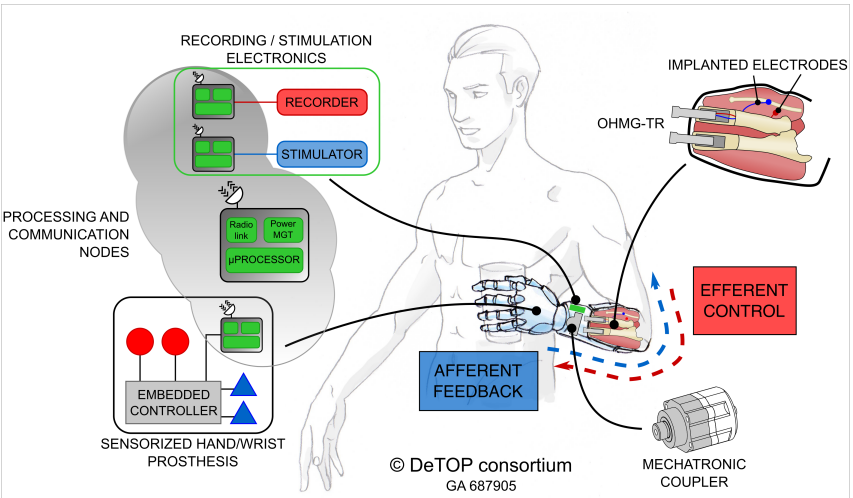


Figure 1.2 – An illustration of the DeTOP final hand implant. Adapted from (4).

1.2 The CSEM

The Swiss Center for Electronics and Microtechnology (CSEM) is a private, non-profit Swiss research and technology organization, headquartered in Neuchâtel with other centers in the cities of Zürich, Landquart, Muttenz and Alpnach.
The CSEM mainly operates in five macro domains: microsystems, systems, ultra-low-power integrated systems, photovoltaics, and surface engineering. In the context of DeTOP, the microelectronic division of CSEM is in charge of the development of a miniature processing and communication node for the control and wireless monitoring of the robotic prosthesis.

1.3 Goals of the experience

This internship took place three years after the beginning of the DeTOP project (March 2016) and a few months before its official end (February 2020).
Before the beginning of the internship, the System on Chip (SoC) for sensing electromyographic (EMG) signals and managing wireless communications was already designed and fabricated. The software was not developed to stream an output signal

compatible with the robotic hand (correct timing and output format), the interfaces for the reception and transmission of external signals for the wireless module were missing and the full functionality of the SoC and the Printed Circuit Boards (PCBs) were not completely tested and validated. As a consequence, the main targets of this experience were:

- Embedded software development of the chip for real-time control of the robotic hand through EMG sensing and wireless communication.
- Characterization and validation of the chip through electrical measurements (oscilloscope, current waveform analyzer, logic analyzer), development of scripts for real time EMG acquisition and improvement of the arrangement of the electrodes.
- Demonstrations of the robotic hand controlling and wireless monitoring of the patient together with DeTOP's partners.

1.4 Outline

The thesis is organized as in the following:

- **Chapter 2** shows the theoretical background needed in order to understand the work done in the next chapters. More specifically, it explains concepts related to the nature of the electromyographic signals, the device used for the control of the robotic hand developed by CSEM, the wireless protocol adopted to monitor the patient and the final demonstrations scheduled with the partners for the final project part.
- **Chapter 3** deals with the methods and setups used for the hardware and software configurations, in order to explain how the results obtained in the following chapters have been reached.

- **Chapter 4** describes the process flow adopted to configure the output of the sensing device to be compatible with the robotic hand controller and the test performed to validate the quality of the digitized electromyographic signals.
- **Chapter 5** is related to the wireless communication between two nodes used for the patient monitoring and eventual hand reconfiguration. It explains the steps for the software configuration and the creation of interfaces to receive and transmit data for each sensor of the wireless network.
- **Chapter 6** presents the work done for an in-house version of the robotic hand control without the components developed by the partners, it shows a possible new setup and applications for the processing and communication node developed by CSEM.
- **Chapter 7** summarizes the achieved results and explains what is still missing to end the DeTOP project.

2 Background of the project

The purpose of this chapter is to introduce the reader to fundamental aspects of the control of a robotic prosthesis. Most of the emphasis will be centered on topics such as biological signals sensing, a description of the hardware adopted in this experience and an overview of the communication protocol adopted for patient monitoring.

2.1 EMG sensing

2.1.1 EMG fundamentals

Electromyographic (EMG) signals measure the electrical current generated in muscles during their contraction (5). Muscles are controlled by the nervous system which is composed of particular connected cells called neurons; the latter are able to use electrical signals to communicate with different parts of the body in a rapid and efficient way.

Because of the ability of the muscular tissue to respond to electrical stimuli, it is possible to contract and extend muscles. The latter are classified as cardiac, smooth and skeletal: only the last kind is concerned for the EMG signal study. The skeletal muscles, attached to the bones, are well supplied with a particular type of neuron called "motor neuron", whose stimulus is able to depolarize and consequently contract the

muscle fiber generating the body movement. The depolarization is responsible for the generation of a local electric field: what is called EMG signal is the recording of the superposition of several electric fields generated in this manner whose magnitude varies between 10 μV and 5 mV and the bandwidth ranges from 5 to 450 Hz if the signal is recorded on the skin surface (the muscular tissues acts as a low pass filter (6)) otherwise it could reach up to 1 kHz if the signal is intramuscularly recorded (7).
In the last decades, researchers have been fascinated by the study of these kinds of biological signals and from the large amount of information that could be retrieved: some possibilities are related to understanding if the patient suffers from a muscular disease or tracing information concerning the whole nervous system. An example is shown in figure 2.1: a raw and apparently chaotic EMG signal can be modelized as a train of Motor Unit Action Potential (MUAP) (8), each one indicating the impulse generated by a different motor neuron connected to a specific cluster of muscle fibers. From this information, in case the magnitude of one impulse is much bigger than others, it is possible to associate this condition with a neurogenic disorder (9).

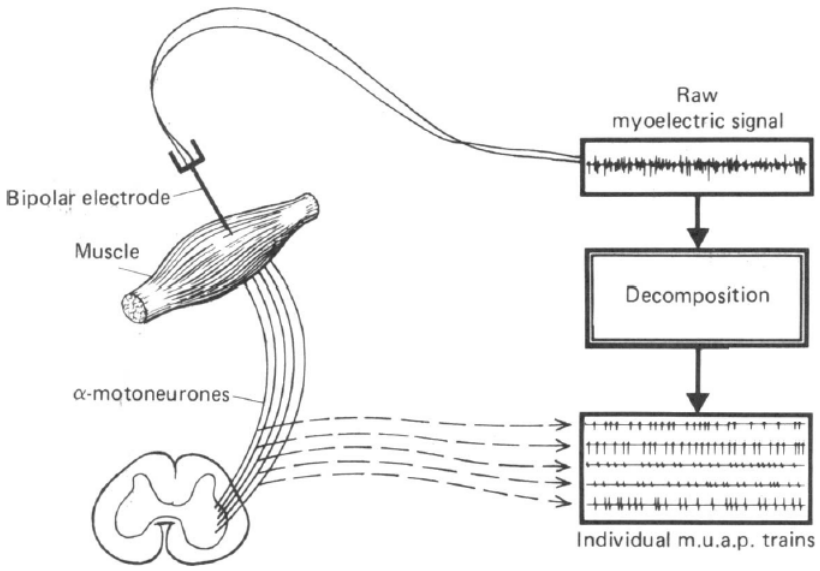


Figure 2.1 – Decomposition of a raw EMG signal in a train of MUAP. Adapted from (8).

2.1.2 Recording apparatus

Several types of electrodes for EMG recording are available on the market: they could be invasive or not, passive or active and so on, as accurately classified in (11). In the DeTOP project the main focus is on intramuscular electrodes, successfully implanted on a first patient by Gothenburg University (10), as shown in figure 2.2.

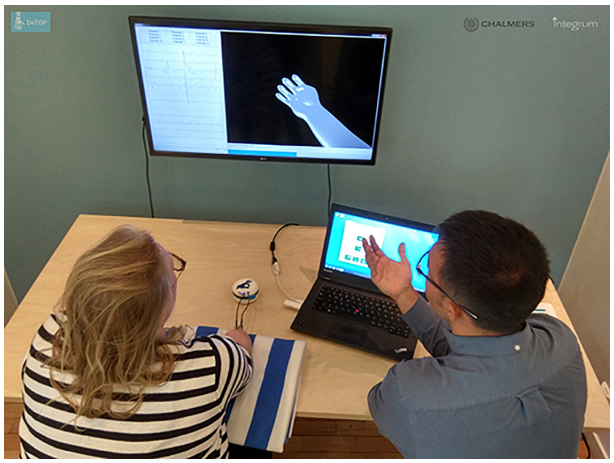


Figure 2.2 – The first DeTOP's patient with implanted electrodes during the training for the prosthesis. Adapted from (10)

During the development of the EMG sensor in CSEM, considering that the patient wasn't available, some surface electrodes were used to test the control of the prosthesis. To have a more detailed view on the two kinds of electrodes used in the project:

- **Passive surface electrodes**: the ones used in this work are the passive type, the same used for the heart monitoring (ECG): differently than the active ones, they don't have any pre-amplifier embedded. They are equipped with a special conductive gel to have good skin contact; the sensor is an Ag/AgCl surface able to sense current on the skin of the patient.
  The main advantages are that they are easy to use and apply, cheap and reusable. The main drawbacks are related to the limitation of only superficial muscles recording, errors due to motion artifacts, susceptibility to environmental condi-

tions and myoelectric crosstalk (it involves the recording of unwanted signals coming from the closest muscles) (3). An example of electrodes used in this work is shown in figure 2.3.

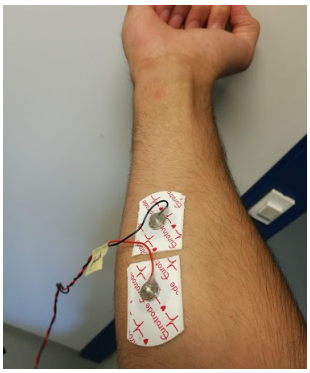


Figure 2.3 – Passive surface electrodes used to test eh SoC for EMG sensing.

- **Epimysial (implanted) electrodes**: they belong to the wider branch of muscle-based electrodes. They are sewn onto the epimysium, the fibrous tissues that coat the skeletal muscles (12). As shown in figure 2.4 A, these electrodes are made of PtIr discs embedded in a silicone backing; they generally adopt a bipolar configuration, removing the common mode noise by performing a simple subtraction with the sensor (11).

  Compared to the surface electrodes, the implanted ones are better for their long-term stability, the selectivity of the EMG signal, reduced cross-talking. On the other side, they require clinical implantation (figure 2.4 B) and they could damage the muscular fiber in contact because of the mechanical stress.

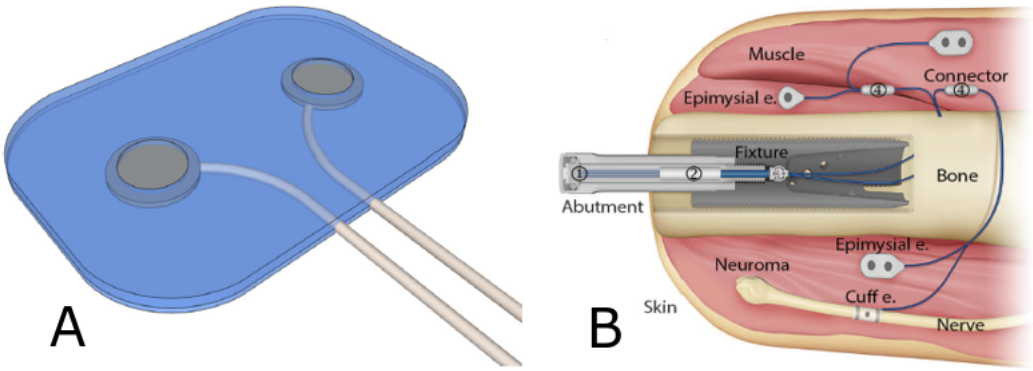


Figure 2.4 – Epimysial electrodes: **A** shows a bipolar epimysial electrode with two PtIr discs embedded in a silicon sheets (adapted from (12)); **B** shows an osseointegration system and an example of implanted electrodes (adapted from (3)).

2.1.3 Noise in biological measurements

When dealing with biological measurements, several elements like the contact with the human body, the electrodes and the environment could play an important role in the signal interpretation. It is important when dealing with EMG sensing to identify the sources of noise and to adopt specific solutions to improve the quality of the signals.

This experience dealt with the following sources of noise, deeply analyzed in (13):

- Electrode noise: this disturbance is related to the high impedance between surface electrodes and skin or implanted electrodes and muscles. In both cases this problem can be minimized by increasing the size of the electrodes or, in the surface ones, using a conductive gel on the skin, cleaning the latter with alcohol and removing hair.

- Electrode motion artefact: it depends on the relative movement between the electrode and the underlying skin or because of a variation of the skin potential because of its stretching or deformation. This is a low frequency disturbance

and an high pass filter (cutoff set to 10 Hz) is a very efficient solution to reduce this noise without losing an important portion of EMG spectrum.

- Ambient noise: it is generated by electromagnetic fields in proximity of power lines and electric equipment. The frequency of these fields is at the frequency of the AC power supply (50 Hz in Europe) and its harmonics. The origin can be magnetic or electric (14) and, if shielding the recording apparatus is not effective, an ideal solution could be the use of bipolar recording electrodes to remove the common mode noise if the electrode impedance is exactly the same for the pair. Other solutions are offered by the use of shorter, twisted leads.

2.2 Serial interfaces

The serial interfaces played an important role in this work. Aim of this section is to give the reader the basis for the comprehension of two interfaces often used for the EMG sensor, i.e. the *Serial Peripheral Interface* (SPI) and the *Universal asynchronous receiver/transmitter* (UART). Their main characteristics could be summarized as in the following:

- UART: it is a type of hardware characterized by two wires, the first for reception and the second for transmission, able to communicate serial data asynchronously and without any clock signal.
- SPI: it is a protocol for serial communication in a synchronous and clocked way in which a master device manages the communication with several slaves. The latter are connected through 4 wires to the master: the MISO (Master Input Slave Output), the MOSI (Master Output Slave Input), the CS or SS (Chip Select or Slave Select) and the clock.

The SPI and UART configurations are shown in figure 2.5 while the main differences between these two technologies are exhibited in table 2.1

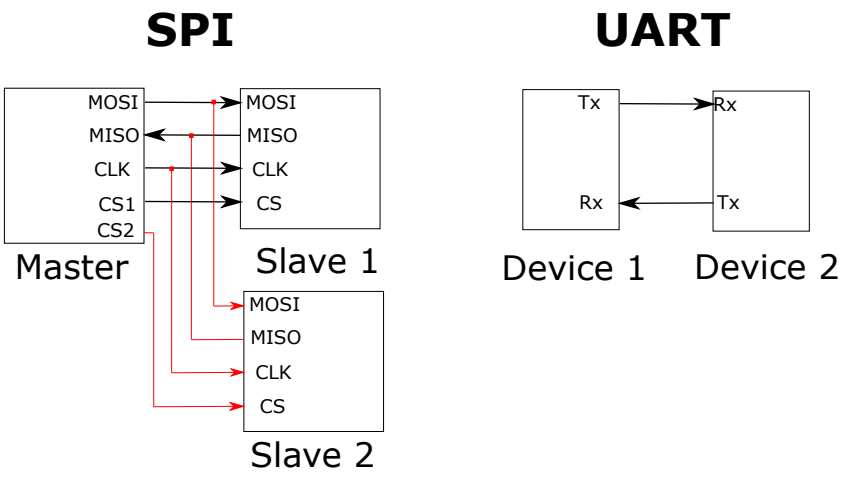


Figure 2.5 – General overview of the SPI and UART interfaces.

Technology	Transfer type	Number of wires/pins	Data rate @ distance	Cost
UART	Asynchronous	2	20 kbps @ 15m	Moderately expensive
SPI	Synchronous	4+	250 Mbps @ 1m	Moderately inexpensive

Table 2.1 – Main differences between SPI and UART communications (15).

2.3 MPC description

The task of CSEM in the DeTOP project was the development of a Miniature Processing and Communication node (MPC) for the hand controlling and monitoring. As agreed, the company has developed an ultra low power 20X EMG sensing channels System on Chip (SoC) with a 55 nm CMOS technology for the real-time control of an Osseo-integrated prosthesis, as explained in (16), able to eventually control an external RF transceiver always developed in house.

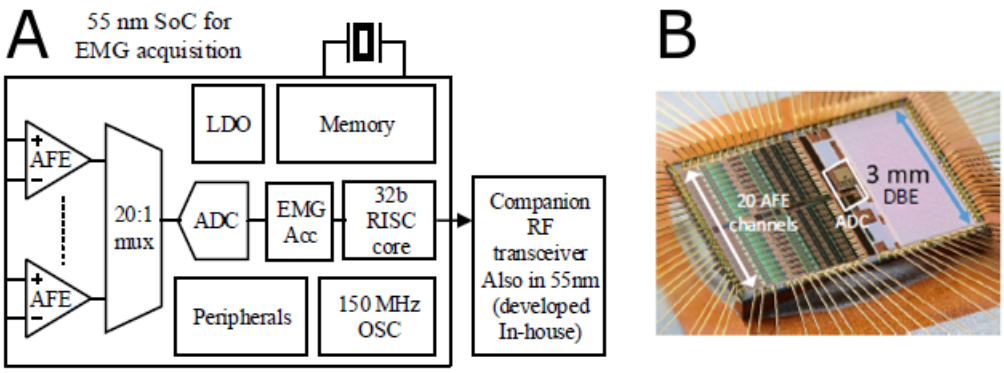


Figure 2.6 – The SoC developed by CSEM for real-time control of the robotic hand and for managing wireless communication. **A** shows a general overview of the block while **B** shows a real picture of the chip. Adapted from (16).

The main components of the MPC (SoC and radio), shown in figure 2.6, are listed below:

- The electrodes are directly connected to 20 re-configurable analog front end (AFE) channels. As shown in figure 2.7, each channel is a cascade of a Low Noise Amplifier (LNA), Programmable gain amplifier (PGA) and a low pass filter (LPF).

Figure 2.7 – A single AFE channel.

- A 12b Successive Approximation Analog to Digital Converter (SAR ADC) digitizes the EMG signals filtered and amplified by the AFE channels. It is possible to tune the sampling frequency from 120kS/s to 1.2MS/s through a specific register configuration.

- An EMG pre-processing accelerator is used to optimize the EMG stream: when activated it transmits the difference of two consecutive samples per each acti-

vated channel in order to compress the information, thus reducing the power consumption.

- The next stage is a 32 bit Reduced Set Instruction Computer (RISC) core developed by CSEM, the Icyflex2 (19): it is in charge of managing the EMG sensor and implementing a specific wireless protocol to enable low power communication through the external radio.

- Besides the core, the digital back end contains standard peripherals (1xJTAG (18), 3xUART, 2xSPI, 2xI^2C (17) and 16xGPIO), a RAM (256 kB) and a ROM (64 kB).

- The external RF transceiver (the IcyTRX (25)) is an ultra-low-power 2.4 GHz RF transceiver developed by CSEM and able to support Bluetooth smart enabled ASICs and SoCs. The main innovations of this device consist in the extremely reduced area (less than 2mm^2 in 55 nm CMOS technology) and less than 10 mW consumption in receive mode using a 1.2V supply.

2.4 State-of-the-art review

Several state-of-the-art SoCs ((20),(21),(22)) have been compared to the device developed by CSEM. As evinced in Table 2.2, no one else offers the same complete functionality of sensing and data processing in a single power-efficient chip with a high number of channels and without the need of using an external controller.

	CSEM Soc	(20)	(21)	(22)
Tech	55 nm	55 nm	180 nm	NA
Area	11 mm^2	18.5 mm^2	NA	NA
Blocks	20x EMG AFE EMG accelerator 32b RISC core	2x ECG 3x Bio - Z 1x PPGx M4	3x ECG ARM Cortex M0 accelerator	32x EMG
AFE power	19 μW/channel	20 μW/channel	31 μW/channel	24 μW/channel
AFE noise	9.5 μVrms/1kHz	9.5 μVrms/150Hz	612 nVrms/150 Hz	2.4 μVrms
ADC resolution	12b	13b	12b	16b
ADC power	77 μW@1.2MS/s	0.9 μW@32kS/s	<15 μW@1kS/s	1.5 mW
Digital core	36 μW/MHz	22.9 μW/MHz	120 μW/MHZ	

Table 2.2 – Comparison among different EMG sensing SoC.

2.5 Robotic prosthesis apparatus

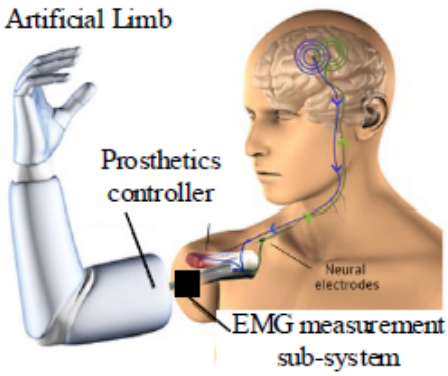


Figure 2.8 – The final DeTOP set-up. Adapted from (16).

The final set-up of the robotic prosthesis, shown in figure 2.8, is based on three main components:

- The CSEM MPC. It is connected to the implanted electrodes of a patient and

used for the EMG sensing and wireless communication through a low power multimode protocol, the WiseTOP (30). It is able to combine extreme miniaturization and power efficiency with a wide range of operations for the control of an artificial limb.

- The hand controller developed by Integrub AB (23). It decodes the EMG signals received by the CSEM board, it recognizes a pattern previously trained and executes the corresponding movement on the robotic hand.

- The robotic hand Mia (24), developed by Prensilia s.r.l., a spin-off from Scuola Superiore Sant'Anna (SSSA), an extremely light prosthesis able to perform multiple grasps and to sense objects.

More specifically, the DeTOP's consortium planned two final demonstrations in order to show the correct working of the whole apparatus:

1. The first demonstration, renamed **EMG demo**, consisted in a patient equipped with electrodes connected to the CSEM board in its turn connected to the hand controller via UART (figure 2.9). Thanks to a Bluetooth connection available on the controller, it was possible to train several movements through an open-source Pattern Recognition software called BioPatRec (37), upload a trained network on the controller, recognize the EMG patterns coming from the CSEM board and actuate the robotic hand.

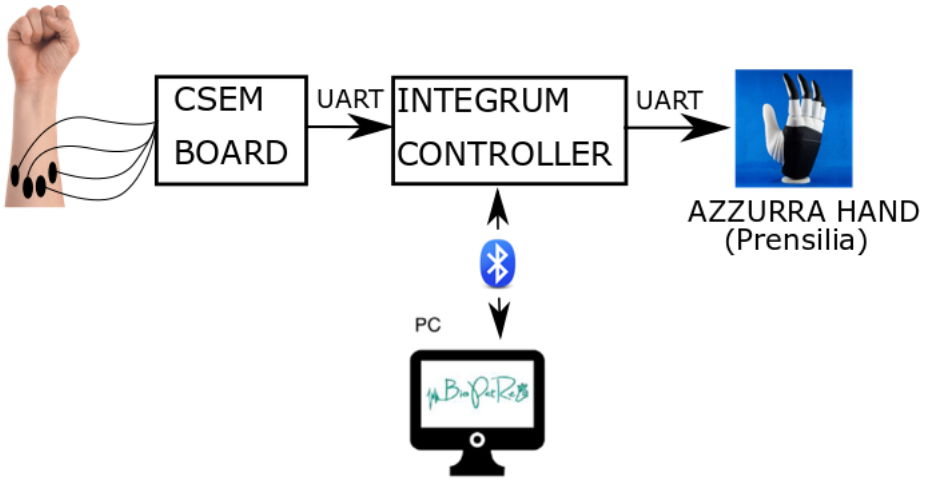


Figure 2.9 – The setup of the EMG demo.

2. The goal of the second demonstration renamed **RF demo** and shown in figure 2.10, was to use the low power multimode wireless protocol developed by CSEM to monitor the patient while controlling the prosthesis and eventually send a reconfiguration signal from a PC (or smartphone) to change the settings of the robotic hand. This time, the EMG sensing was performed by the controller (equipped with an INTAN EMG sensing chip (22)). Two MPCs (one attached to the prosthesis and a second one to a PC) were involved in the demonstration.

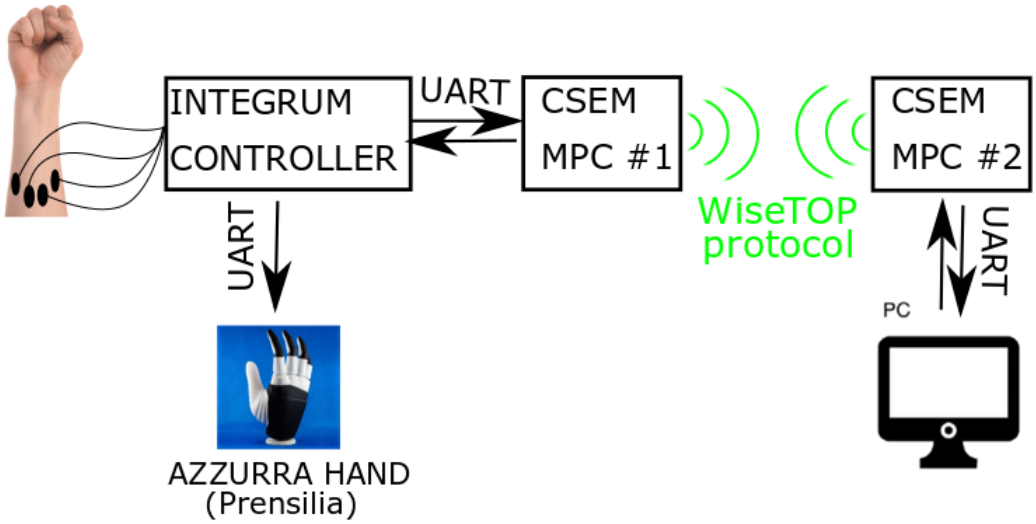


Figure 2.10 – The setup of the RF demo.

2.6 Wireless protocol

2.6.1 The OSI model

In order to have a clear view of the communication systems discussed in the next chapters, a short review on the Open Systems Interconnection (OSI) model developed by the International Organization for Standardization (ISO) and the International Telecommunications Union (ITU) is needed.
The OSI is a conceptual model used for adding structure to communications software. Thanks to this model it is easier to make changes in a complex software where several tasks are involved (transmission, physical networks, multiple kinds of operations). The OSI divides the communication processes into seven layers, each one of them performing a particular action for the layers above while giving services to the ones below it (26). The main difference among these layers is that the lowest deal with hardware or firmware (i.e. a software executed on a specific HW) while the highest on software only. Layers from one to three deal with traffic communication through the network to an end system, the ones from four to seven involve the end system in order to complete the process.
A general description of the OSI layers can be found in Table 2.3. The main focus in the following will be on a sub-level of the second layer, the Data Link, named Media Access Control (MAC). The interested reader can find more information on the unaddressed layers in (27).

OSI Model		
	Data Unit	**Layer**
Host layers	Data	#7 Application
		#6 Presentation
		#5 Session
	Segments	#4 Transport
Media Layers	Packets	#3 Network
	Frames	#2 Data link (MAC + LLC)
	Bits	#1 Physical

Table 2.3 – Description of the seven layers of the OSI model (27).

2.6.2 MAC fundamentals

In the Internet of Things (IoT) era, the Wireless Sensor Networks (WSN) are generally made of battery-supplied sensors organized in a node used to collect and process data (the sink node) and several others used as sources (slave nodes). The consumption of each node for the transmission has become a critical topic. In this scenario, the leitmotiv for the development of wireless communication is not related to classical performance parameters (throughput, latency, and fairness) but it is oriented towards a reduction in energy consumption to maximize the lifetime of the network (28).
In order to coordinate the access to each node for a low power communication of a WSN, a sub-layer of the OSI model is involved: the Media Access Control (MAC) protocol.
Each MAC protocol can adopt different rules for turning the radio off. A standard implementation is a fixed duty cycle or a traffic change according to time or place.
The challenging aspect of the development of such a protocol is that the common medium among the nodes, the wireless channel, is broadcast in the atmosphere or space where several pre-existing communications may cause interference: this explains the pursuit of optimization for the channel access or for the transmission of

the packets and their retransmission (29).

2.6.3 The WiseTOP protocol

In particular applications in which the traffic increases only for a given duration, it is possible to use several Multiple Access Control (MAC) protocols through a switching MAC system: this is the case of the WiseTOP protocol developed by CSEM for the DeTOP project (30), initially designed to establish a two-way communication between a first MPC embedded in the patient, sending EMG data to the prosthesis, and a second one in the robotic limb able to send sensory feedback to the patient. If the robotic hand is not worn, a low traffic protocol is entered where the only operation is a periodic check on the traffic status.
To fit the DeTOP project, requirements needed are highlighted in table 2.4.

Uplink datarate equal to 960 kbps.
Downlink traffic for configuration and mode change.
Adaptable data throughput to minimize energy consumption.
End-to-end latency less than 100 ms (in both directions).
Low power mode.
Single hop from sensors to sink (star topology).
Resistance to interference from other 2.4 GHz technologies (Bluetooth).

Table 2.4 – Targets of the WiseTOP protocol (30).

The following MAC protocols were adopted by WiseTOP to fulfill the previous requests:

1. The WiseMAC (31), a low power protocol adopted during low traffic conditions.
2. Time Division Multiple Access (TDMA) (32), used in high traffic conditions.

This experience was entirely based on a high traffic communication and on the TDMA protocol, this is why the following will be entirely focus on it.

2.6.4 TDMA structure

TDMA protocol divides time into intervals of equal duration (in the following named *Beacon intervals*) as shown in figure 2.11. The beginning of each interval starts with a *Beacon*, sent by the sink node. The next amount of time is divided in 32 slots in which a payload can be sent from the sink towards a particular slave or vice-versa; the remaining time of the beacon interval is used to let the nodes in the network finish their operation before sending a new beacon.

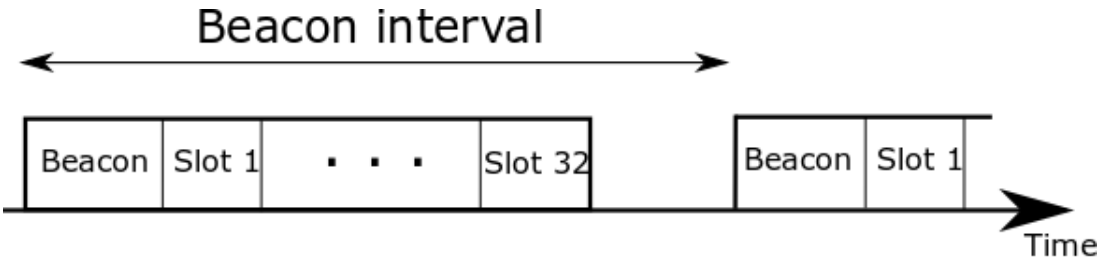


Figure 2.11 – Simplified sketch of the time intervals adopted by the TDMA MAC protocol (30).

The Beacon is needed to keep synchronization of the slave nodes with the sink, to acknowledge the transmission of the previous period and to communicate the slot allocation. The beacon interval and the payload size transmitted during each slot can be changed.

3 Methods

This chapter aims to describe the methodologies and lab equipment used for the test, validation and development of the hardware that will be taken for granted in the next chapters.

3.1 PCBs

During this work, the SoC was hosted on two different printed circuit boards. Several components were soldered on them, mostly voltage amplifiers and resistors, in order to correctly set all the pins and to allow a wide range for the voltage supply (2.7V - 6V). The main differences between the two mentioned boards were the following:

- The first one, the **Characterization board**, had been designed and manufactured before the beginning of this internship. As shown in figure 3.1, it contained a socket in the middle to host the chip without any soldering procedure needed. The big size of this board (15 x 16 cm) was due to two main reasons: it was used to test the system for different temperatures (explaining the empty circle around the socket) and it contained several jumpers used to easily characterize each pin (current or voltage measurements).
- The second one, from now on called **Final board**, was conceived to be integrated

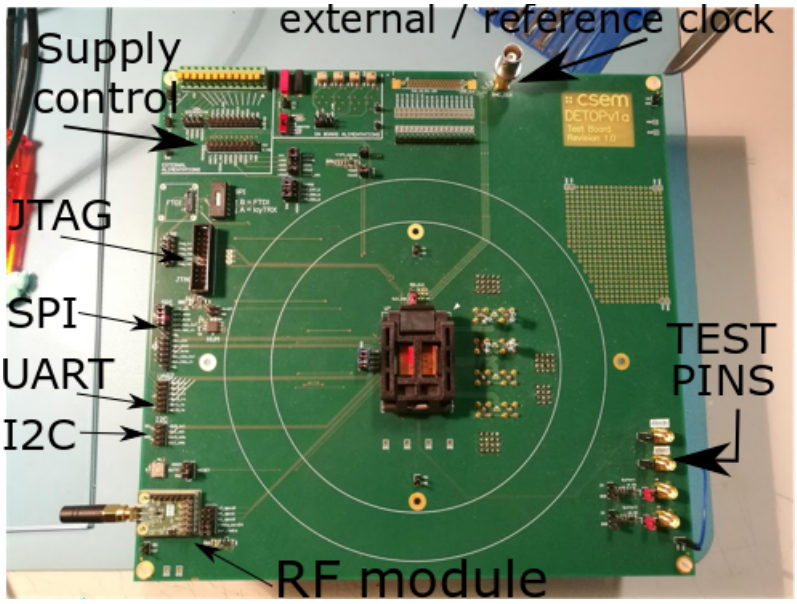


Figure 3.1 – The PCB used for the complete validation of the SoC, the "characterization board".

inside the robotic arm. To minimize the design, the SoC was soldered on the board. Two slightly different PCBs were assembled for the two demonstrations, the first (figure 3.2 A and B) for the EMG demo and the second (figure 3.2 C and D) for the RF one. A short review on their differences is shown in table 3.1.

Board's name	Special features
EMG Board	- 2x Molex Connectors (10 bipolar electrodes each) - Battery socket
RF Board	- RF external module attached on it

Table 3.1 – Main features of the miniaturized versions of the characterization board.

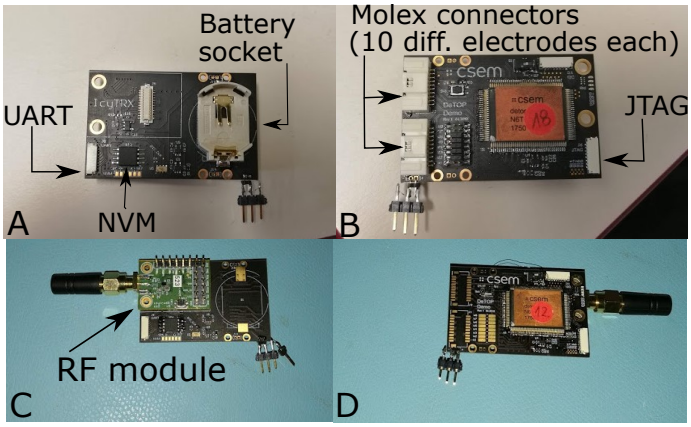


Figure 3.2 – The final boards adopted in this work. **A** and **B** show the back and front of the EMG demo board, **C** and **D** the RF demo board with the IcyTRX attached on the back.

3.2 Development environment

In order to load software on the board, a C script was compiled and thanks to a list of directives (contained in a "make file") an Executable and Linkable Format (ELF) file was generated. Each script was always containing the common CSEM libraries, defining the registers and functions for the controller, and a file containing the initialization for the AFE settings. Once generated, the ELF file was loaded through a high-speed JTAG debugger in the chip's RAM; the setup is shown in figure 3.3.

To understand if the software was correctly running on the SoC, each script was written in such a way to have a LED connected to a GPIO port toggling during the execution .

To verify the correctness of the timings and format of the logic output generated by a particular software, a logic analyzer generally connected to the UART or SPI ports was used. The setup of data acquisition is shown in figure 3.4.

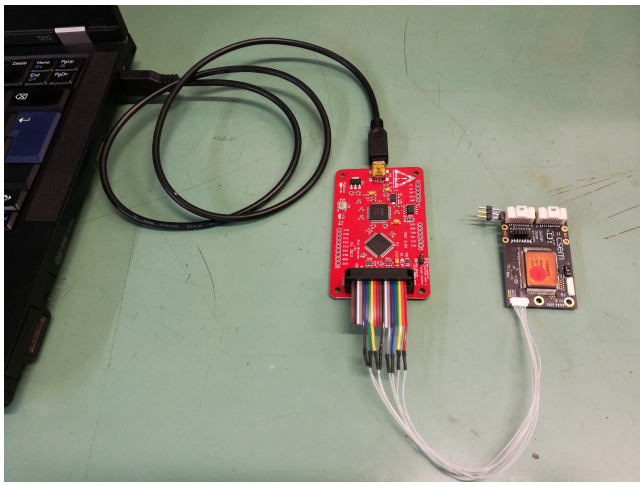


Figure 3.3 – The JTAG debugger loading data from the PC to the chip's RAM.

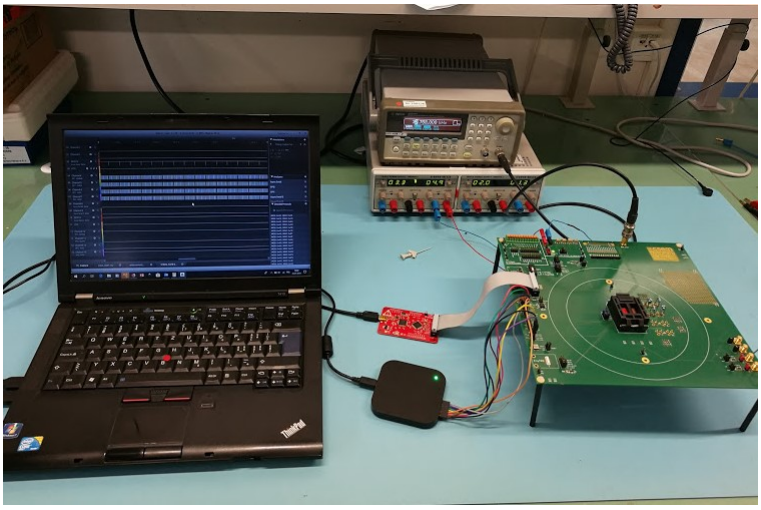


Figure 3.4 – Data acquisition setup with the logic analyzer.

3.3 AFE testing

Due to the small amplitude and bandpass of the EMG signals, an accurate control over the analog front end configuration was necessary, considered that before this experience the hardware was mostly simulated and not totally tested.
Because of its reconfigurability through some registers settings, it was possible to display one AFE channel differential output on two test pins only available on the characterization board. Similarly, as shown in figure 3.5, it was possible to bypass the

AFE blocks in order to isolate and test just one or two constituting blocks.

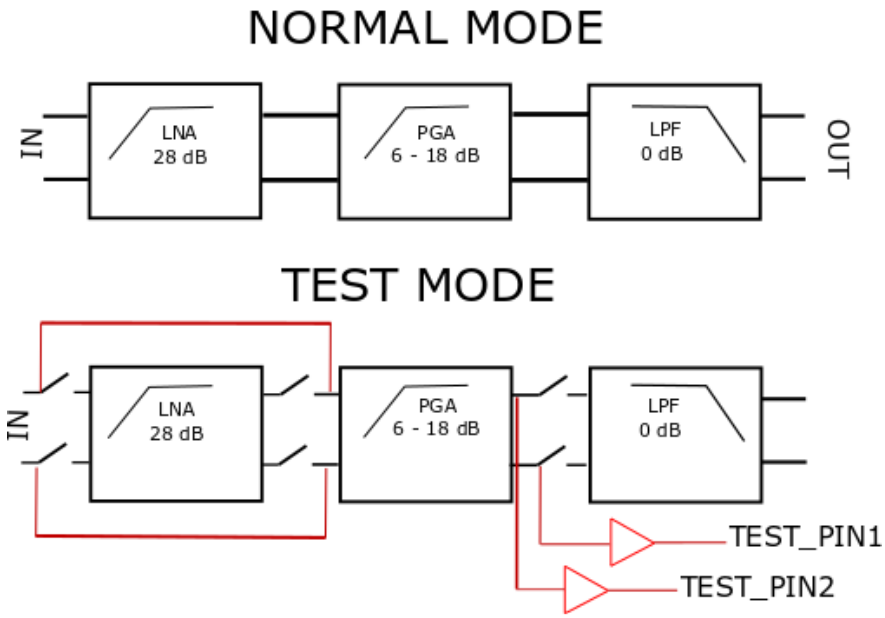


Figure 3.5 – An example of the bypass method adopted by the AFE to characterize isolated blocks thanks to the test pins of the characterization board.

Once a particular test configuration was chosen and the software containing all the register configuration written in the chip's RAM via JTAG, the electrical test was performed (figure 3.6 A). A double signal generator was connected to the selected input pins, injecting two sinus waves with 5 mV of amplitude, no offset and 180° out of phase. Both inputs and outputs (test pins) were displayed on a digital oscilloscope, as shown in figure 3.6 B.
The parameters extracted were two:

- Gain: the ratio of the amplitude of the signal on the test pins against the input one.
- Cutoff frequency: the amplitude of the signal was measured in the bandpass; by increasing the frequency, the cutoff corresponded to a 3 dB reduction of the amplitude of the output wave.

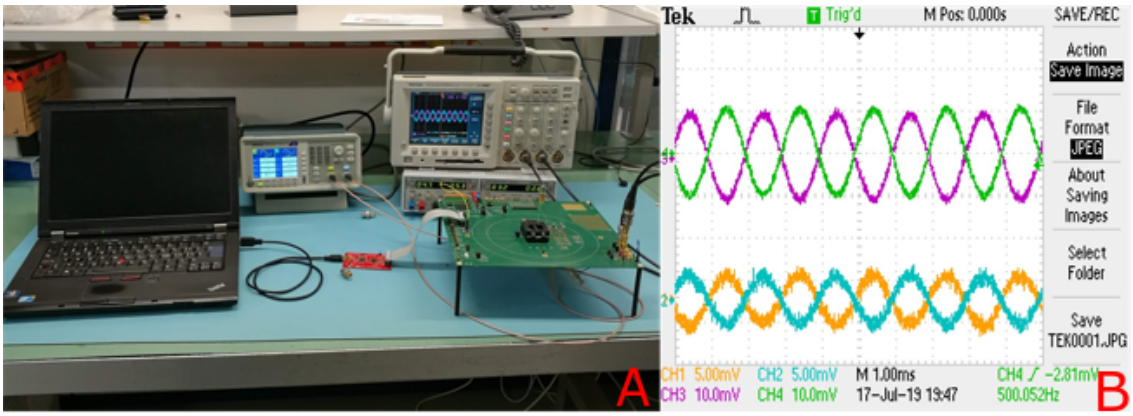


Figure 3.6 – **A** The experimental set-up used to test the analog front-end. **B** An example of signals visualized on the oscilloscope: top ones represent the differential AFE output, bottom ones the input.

3.4 Non Volatile Memory configuration

The JTAG debugger was not the most practical way for the software loading, especially during a demonstration. This is why all the PCBs were equipped with a Non Volatile Memory (NVM), the *ST M25P32* (33), connected to the main chip through SPI. Once a software was written a first time on the NVM through the controller of the SoC, just by pressing the reset button a re-boot was quickly performed without any cable connected to a PC. The directives for the compilation were changed in order to generate a binary file from the ELF one to be directly loaded inside the NVM.

During the external memory writing, a problem related to a watchdog time-out activation was preventing the whole transmission of bytes. More specifically, only the first 14 bytes of the binary code (with an original size of around 200 kB) were sent from the Master (main chip) to the Slave (NVM), as shown in figure 3.7. Once several measurements were performed with the oscilloscope to isolate the problem, the solution involved a simple deactivation of the time-out through a register configuration.

Even after a complete NVM writing, during the re-boot the software was not correctly running on the RAM: this problem was caused by a bug in the compiler, identified during this work. It consisted in the corruption of few registers containing the NVM addresses in which performing the reading. This problem was solved by CSEM's

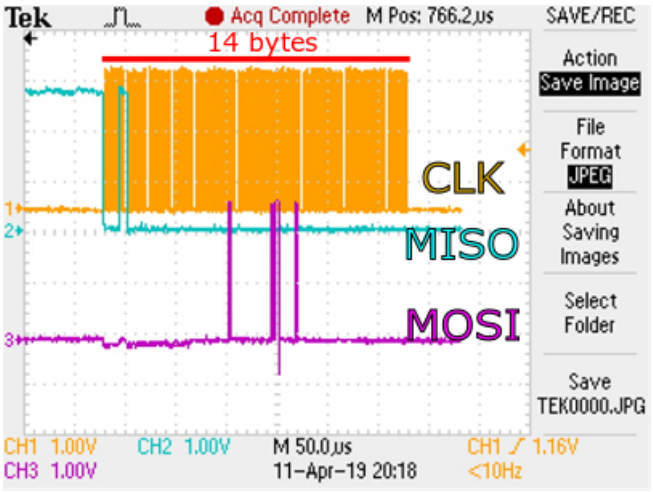


Figure 3.7 – The watchdog timeout activation after that 14 bytes were written on the NVM. This measurement was performed through the oscilloscope connected on the SPI.

engineers by using a boot manager allocated in few incorruptible addresses and able to manage the RAM writing when loading from the NVM.

3.5 Serial data acquisition

In order to validate the serial output of a device two options were available:

1. Recording for some seconds the output with the logic analyzer, exporting it into an excel file and using a script Matlab or Python for its decoding and plotting.

2. Connecting the UART output directly into the USB port through a device able to convert UART to USB (FTDI cable TTL-232R-3V3 (34)) and create a script able to plot in real-time the acquired EMG signals.

Both the scripts were written with *Matlab 2012a*.
The first option was generally used when the exact acquisition time of each byte was needed, generally when sending precise data to the partners of the project for the compatibility with their blocks.

The second one (shown in appendix B) required more effort to be developed but it was very helpful in order to get fast recordings and to study the noise influence of the surrounding environment, also thanks to a real-time Fast Fourier Transform (FFT) running in parallel. To elaborate this script, a knowledge of the FIFO like buffer present in each USB port was needed. The whole setup is shown in figure 3.8.

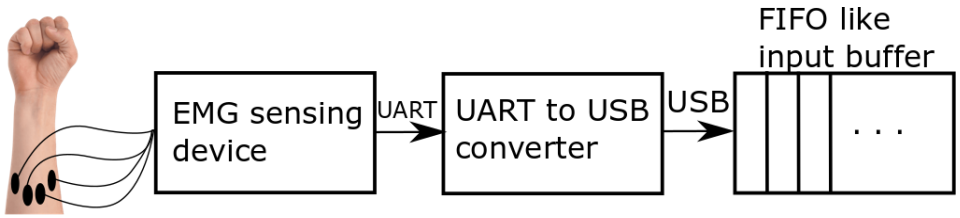


Figure 3.8 – A simplified sketch to illustrate the setup used to interface the SoC with the PC.

For the PC used in this experience, the size of the input buffer was 512 bytes and the *fread* Matlab function was used to extract binary data from it. It was not possible to read in a single *fread* execution more bytes than the size of the buffer, this is why an initial cycle was created to read 30 times 512 bytes and store them into a vector eventually decoded and plotted. The Real time included the continuous running of this algorithm and update of the plot.

4 Real-time EMG streaming

This chapter describes all the steps needed to configure both HW and SW to adapt the output stream of the EMG sensing SoC in order to have compatibility with the robotic hand.

4.1 AFE configuration

The first step towards EMG sensing was the AFE configuration. As previously discussed in section 3.3, the settings of each analog block were specifically programmed in order to improve the quality of the acquired signal.
In particular, for each block:

- **LNA**: the only programmable parameter was the high pass cutoff frequency, set to 10 Hz; the gain of this block was 28 dB.
- **PGA**: the gain of this block was set to 6 dB and its high pass cutoff frequency to around 7 Hz.
- **LPF**: the gain of this block was constant (0 dB) and the low pass cutoff frequency was set to 1 kHz.

The total gain of each AFE channel was 34 dB and the bandpass between 10 Hz and

1 kHz, perfectly fitting the EMG spectrum of both surface and intramuscular EMG signals (section 2.1.1). It could have been possible to increase the gain of the PGA but the quality of the signal was hugely deteriorated in that case. The AFE transfer function is shown in figure 4.1.

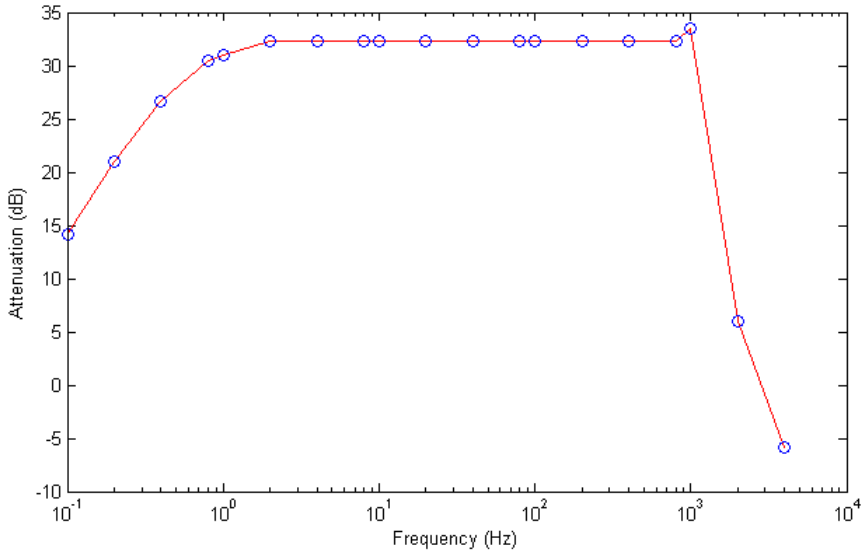


Figure 4.1 – Selected transfer function for EMG sensing.

4.2 Requests from partners

To have compatibility with the Integrum board (the hand controller mentioned in section 2.5), the output of the SoC had to fulfill the following requests:

1. A UART interface for the data transmission with a Baud Rate of 460800 bit/s.
2. EMG signals recorded using only 8 differential channels over 20.
3. Sampling frequency for each channel equal to 1ksps.
4. The size of each message equal to 19 bytes (uint, most significant byte and least significant bit first), starting with 3 headers (0xC0, 0xB0, 0xA0) to indicate the beginning of the transmission and followed by 8 pairs of bytes, each one

containing the ADC output associated to a different channel. An example is partially shown in figure 4.2.

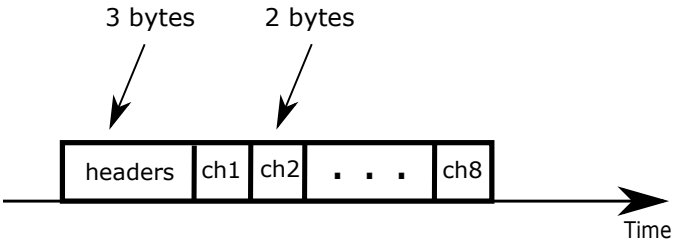


Figure 4.2 – The format of each 19 bytes packet outputted from the SoC.

4.3 Embedded software development

The first step toward the output configuration was the adjustment of the Baud Rate; the following formula, found in the specification of the device, was used for its calculation:

$$BaudRate = \frac{clockfrequency}{UART\_BAUD + 1} \tag{4.1}$$

With an internal clock frequency set to 153 MHz, the parameter $UART\_BAUD$ was set to 331 to achieve a baud rate of 460800 bits/s. To verify its accuracy, some dummy data were sent via UART, the time interval associated with a single bit was measured through the logic analyzer and its reciprocal (the definition of Baud Rate) calculated. Once the size of each bit was correct, the next move was the activation of 8 differential EMG channels over 20 available (numbered from 0 to 19). During this experience, only the ones numbered from 1 to 8 were used. The activation of a channel was realized by writing several ones in 20 reserved bits of a register.

Despite 20 EMG channels available, only one ADC was present (figure 4.3): after each sampling, the ADC generated an interrupt that was sent to the processor and consequently the multiplexer selected the next activated channel as ADC input.

Because of the system calibration performed before this experience, the ADC sampling frequency was set to 120 kS/s. As shown in equation 4.2, if the channels activated were 8, the sampling rate for each one of them was 15 kS/s, against 1kS/s asked from the

partners.

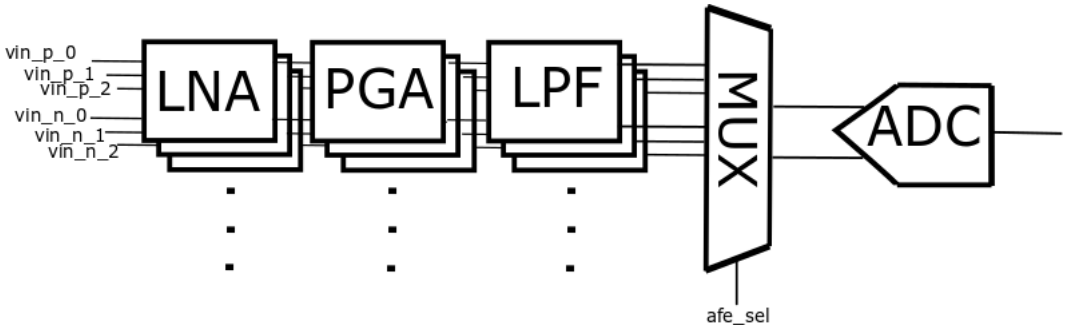


Figure 4.3 – The structure of the AFE channels inputting the ADC through a multiplexer.

$$Single\,channel\,sampling\,frequency = \frac{ADC\,sampling\,frequency}{number\,of\,activated\,channels} \quad (4.2)$$

To downsample by a factor of 15, considering 8 consecutive adc samplings as a "cycle", during 15 consecutive cycles just one of them was stored into a buffer and then sent to the UART, as sketched in figure 4.4. The data of each channel were released from the ADC on 12 bits: the 4 MSBs and 8 LSBs were split into 2 different bytes and saved on a buffer containing the transmission headers (0xC0, 0xB0 and 0xA0) as first 3 bytes.

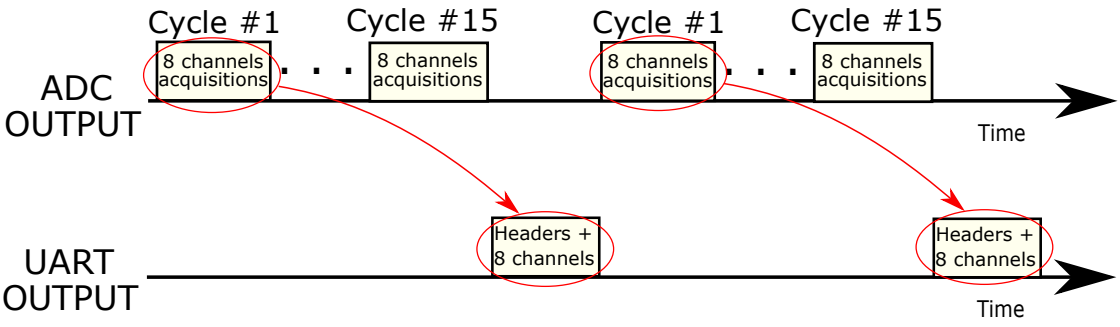


Figure 4.4 – The downsampling technique used to fulfill the partners' requests.

Even though the Baud Rate and the sampling frequency were correct, there was

a problem concerning the time interval between consecutive packets, larger than predicted. To debug this error a toggling function, connected to a led associated with a GPIO, was activated during each ADC interrupt.

Figure 4.5 – The ADC interrupts blocked during UART transmission.

As shown in figure 4.5, it was clear that while the UART was sending data, the ADC was blocked because of a cycle that was blocking the software while the UART FIFO (8 bytes size) was not empty.
This problem was solved by avoiding a blocking approach and pushing 8 bytes over 19 of the packet towards the UART FIFO during alternating cycles, as shown in figure 4.6.

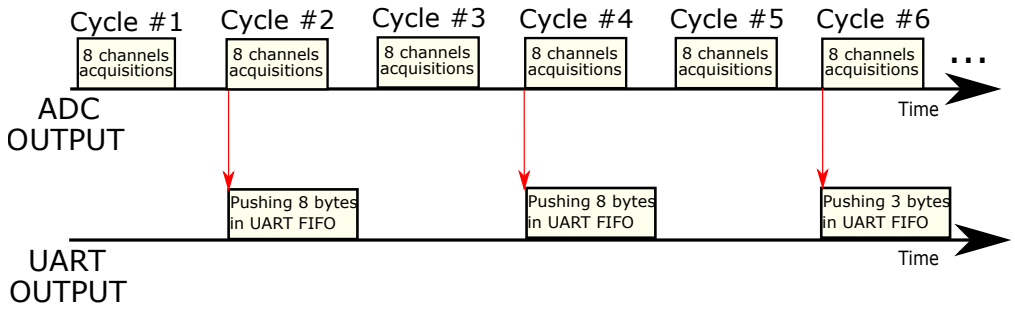


Figure 4.6 – The workaround used for the 19 bytes transmission.

The final result is shown in figure 4.7, showing a stable output and the ADC unblocked during the UART transmission.

Figure 4.7 – Unblocked ADC interrupts during UART transmission.

4.4 EMG measurements

Once the output format was correct, the EMG sensing had to be validated before sending the results to the partners.
Considering that in real patients the electrodes had to be distributed around the abutment, close to the amputation area, the electrodes configuration shown in figure 4.8 A was chosen in order to have results closer to the final ones.

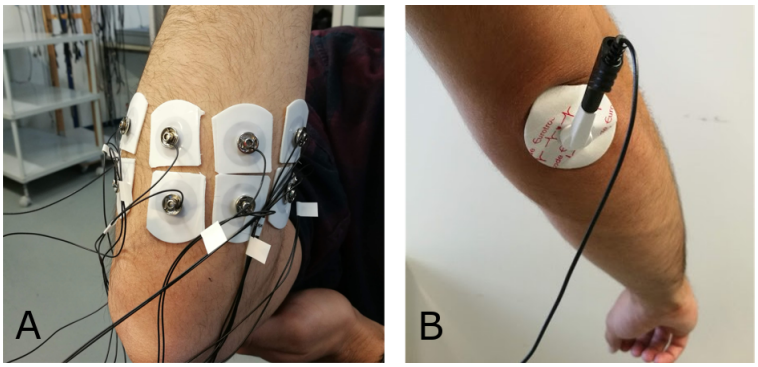


Figure 4.8 – Electrodes configuration. **A** shows the 16 electrodes connected to 8 differential AFE channel while **B** shows the reference electrode.

As explained in (35) a reference electrode is always needed when performing electromyographic measurements; it has to be placed as far as possible from the differential electrodes and it must be on an electrically neutral tissue. During this experience, a bony prominence as the elbow was chosen as reference (figure 4.8 B).
During the first phase, only 2 electrodes were glued to the arm, representing only one differential channel while the remaining seven were floating. A huge 50 Hz noise appeared both on the channel with electrodes and on the floating ones (figure 4.9).
Several options were attempted with poor results: twisting the electrode wires, using a battery instead of a supplier to avoid the contact with the electric line, reducing the length of the wires.

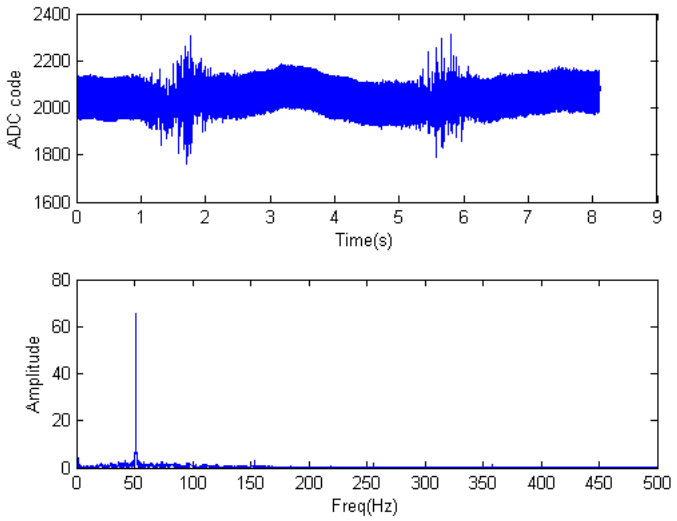


Figure 4.9 – A 50 Hz noise example.

Thanks to the use of the real-time plotting script mentioned in section 3.5 it was possible to understand that several factors in the lab room, above all a metallic table on which the measurements were performed, were contributing to the ambient noise. A backup solution developed together with partners was the implementation of a 50 Hz notch filter on the Integrum controller in order to remove only the ambient noise frequency, in the end unneeded because of the noise reduction obtained when performing the 8 channels measurements with not floating pins.

An important factor influencing the ambient noise was the electrodes overlap: initially (figure 4.10 A) all the electrodes were overlapping, both differential pairs and closest channels. As suggested from the partners, a cut on the electrodes, as evinced in figure 4.10 B, helped in the noise reduction as well.

A final example of an 8 channel measurement using the real time script is shown in figure 4.11.

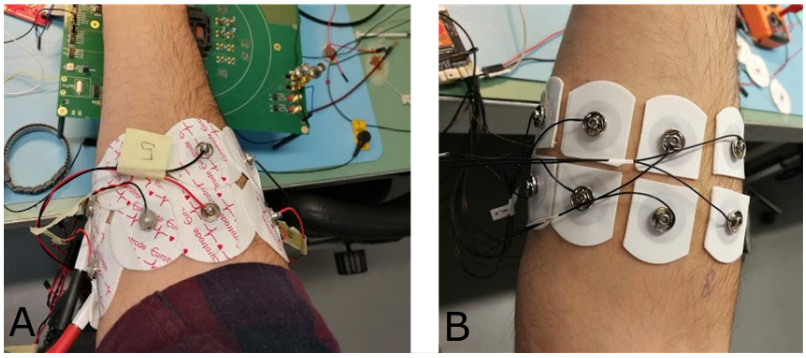


Figure 4.10 – **A** Overlapping surface electrodes initially used for the measurements. **B** Non-overlapping sliced electrodes adopted for noise reduction.

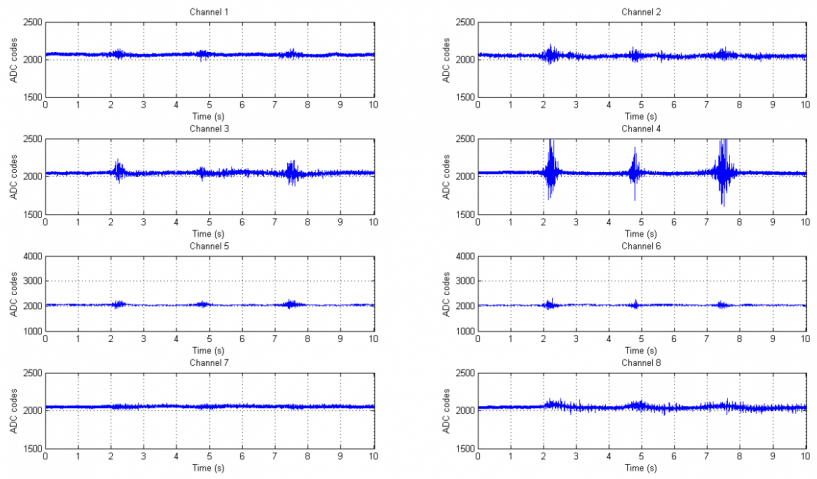


Figure 4.11 – Grasps recorded with the real time software.

4.5 Final EMG demonstration

Once the signal recorded on the final PCB was clear, the data were sent to the partners; they agreed to come to Neuchâtel to merge the setup for the EMG demo.
During this period the robotic hand was not yet available, so a virtual one was used instead. That day four movements were recorded (hand closed /open, wrist flexion /extension) and after some software training, the pattern recognition allowed the virtual hand to perform in real time the same movements of the patient.

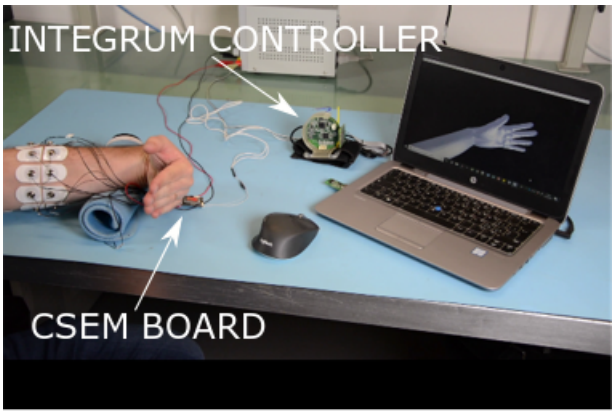


Figure 4.12 – Final EMG demo with the virtual prosthesis.

After a few weeks, the robotic hand was available and the partner sent to CSEM a video in which 6 movements were recorded, trained and uploaded on the controller. The hand was perfectly responding to each movement mimicking the behavior of the patient (figure 4.13).

Figure 4.13 – Final EMG demo with the robotic hand.

5 Wireless communication

This chapter describes the steps followed in order to establish a stable wireless communication between two board and to create some input/output interfaces for the hand monitoring and reconfiguration.

5.1 IcyTRX configuration

Before starting this experience, the wireless communication between the two boards was working on the characterization board but the software was not yet adapted for the final one.
The main difference between the two PCBs was the interface with the IcyTRX: in the characterization board (figure 5.1 A) two different SPIs (numbered 0 and 1 respectively) were used to connect NVM and RF module; in the final board the PCB designers decided to adopt a more compact solution by sharing the same SPI with the only exception of the Not Slave Select (NSS) pin, substituted by a GPIO for the IcyTRX (5.1 B).
Thanks to accurate measurements through Logic Analyzer, the first problem was found to be related to the slave selection. A different GPIO was used for debugging and, due to the lack of masking during a bitwise assignment in the register for the activation of all GPIOs, the NSS signal for the IcyTRX was uncontrollably changing.

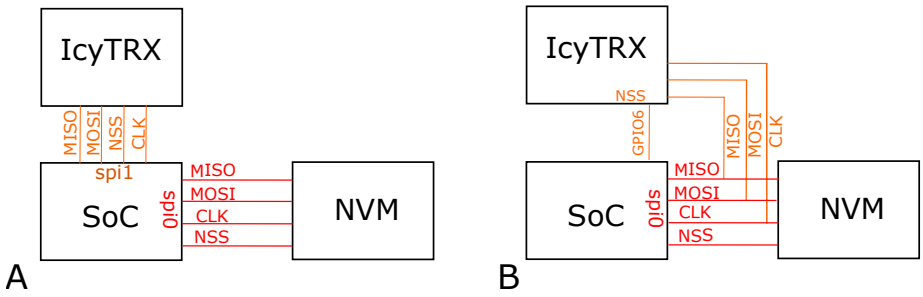


Figure 5.1 – **A** shows the SPI connections on the characterization board while **B** the hardware modifications on the final PCB.

Once identified, this problem was easily solved by masking through OR and XOR operations the previous assignment during the debug.

Even when the Slave Select was stable, the radio was not yet recognized by the motherboard. An inspection with the oscilloscope exposed a problem related to the IcyTRX supply ramp (figure 5.2) when turned on. During the reboot, the software was trying to identify a component that was still off: to solve the issue, the software was blocked for few milliseconds in the interval between the supply turned on and the identification of the radio in order to stabilize the voltage signal, thus establishing the correct execution of the software on one single node.

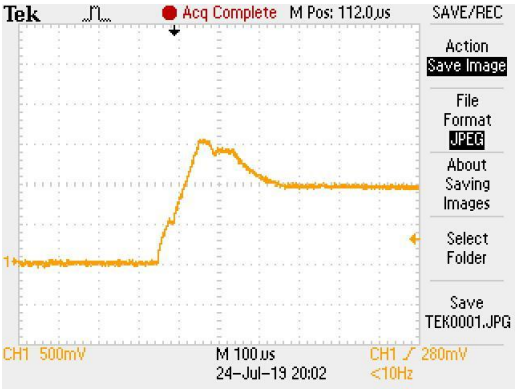


Figure 5.2 – Voltage ramp on the IcyTRX supply when turned on.

In the following, to simplify the validation of the wireless module, only the TDMA protocol was activated instead of the whole WiseTOP (section 2.6.3).

In order to have communication between the two nodes, the software for the slave

and the sink were installed on two different boards: the protocol was designed for the communication to start when the slave received the first beacon signal. The last event happened correctly but the time intervals generated on the two boards were different for the two nodes and they were changing during each reboot, blocking the communication right after the first beacon reception.
The first block investigated was the timer. As shown in figure 5.3, this block consisted in a programmable clock divider followed by a counter whose value was incremented after each clock period. When the value stored in "Threshold Register" equaled the counter register, the latter was reset and an interrupt generated, thus marking the time in the wireless protocol.

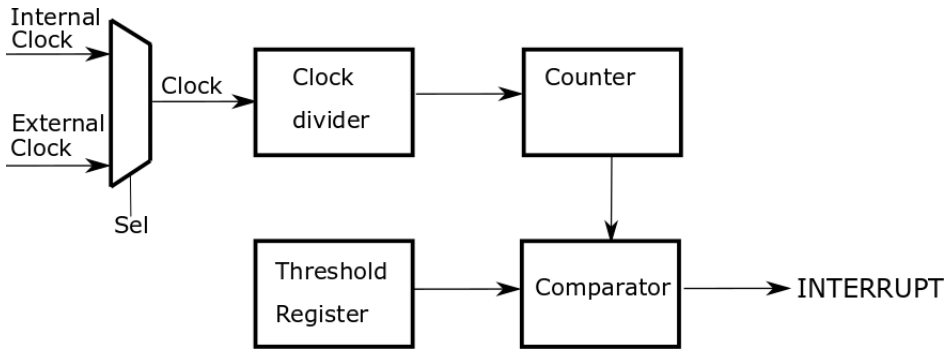


Figure 5.3 – The main blocks composing the timer of the SoC.

A simple software for toggling a GPIO each timer interrupt was created and loaded on one board. The pin under discussion was measured through the logic analyzer, showing a toggling period with the same problems previously listed (intervals different than the ideal design and changing after each reboot).
In order to check if the problem was coming from a previous stage, the PCB configuration was changed in order to select on the multiplexer shown in figure 5.3 an external signal coming from a 75 MHz oscillator as internal clock, against the 153 MHz originally supposed to be internally generated. Thanks to this operation, the problem related to the timer instability disappeared, allowing the communications between the two boards.

5.2 UART interfaces

Before this experience, the communication between two nodes was done through a dummy payload internally generated. To adapt the wireless protocol to the DeTOP's requests, the payload had to consist in data received from the UART port on a first node of the WSN and, once received by a second node, the payload had to be re-transmitted via UART, as shown in figure 5.4, with a Baud Rate equal to 460800. It was supposed to be a two-way communication.

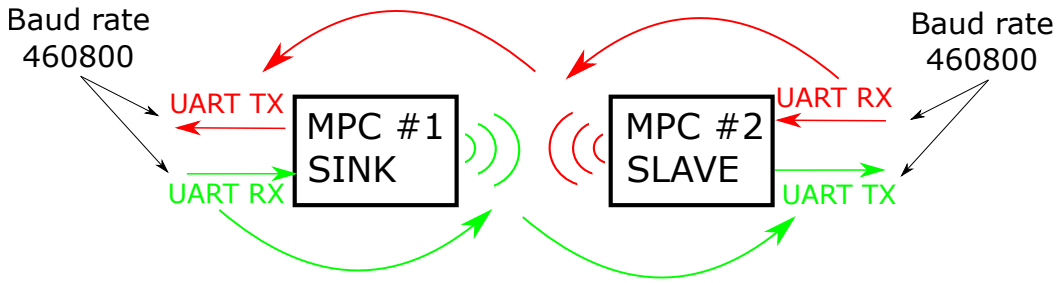


Figure 5.4 – The UART interfaces to be developed for the wireless protocol.

The beacon signal sent by the sink towards the slave, besides activating the communication, contained various items of information linked to the number of bytes contained in the payload of each slot or the reservation of the latter (if they were reserved for transmission from sink to slave or vice-versa).
As explained in figure 5.5, in the initial part of this experience the 32 slots available during a beacon interval, each one containing a 50 bytes payload, were reserved for transmission from slave to sink. Table 5.1 contains the specifications used for communication.

Slots Reservation	Each slot for tx from slave to sink
Beacon int	110 ms
Payload size per slot	50 bytes
Additional operations	13 ms

Table 5.1 – Values used to configure the TDMA protocol.

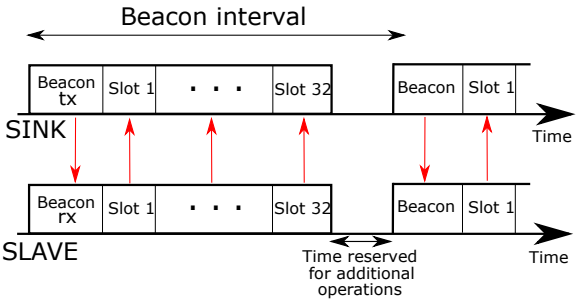


Figure 5.5 – Assignments of each slot for the TDMA module during this experience.

At this point, it is important to highlight that the software loaded on the board was working as in the following:

- The main software, containing the initialization of several functions and an infinite loop running on the SoC.

- For each interrupt generated by the timer, the TDMA protocol (working as an interrupt handler) was called and executed for a specific amount of time, before re-executing the main software.

- As shown in the following sections, an operation performed in the main software wouldn't have affected the behavior of TDMA protocol, considering that the timer interrupt had the priority over other operations performed in it. On the contrary, an additional operation performed on the TDMA protocol, if too long, could generate a potentially fatal delay for the communication.

In the next sections, the additional operations previously mentioned, the reception and transmission of a payload through UART, were added to the software.

5.2.1 UART transmission

The first operation added to the software was the transmission via UART of a payload wirelessly received by a node. This operation was added to the TDMA software because of several callback functions available, each one called during a different phase of the protocol (beacon transmission, beacon reception, data transmission, etc.). For this task, the callback function used was the one activated during the reception of a TDMA packet (sent during each slot of the transmission and containing both headers and payload bytes).
In the initial test a dummy payload was generated and sent by the slave to the sink, as shown in figure 5.6.

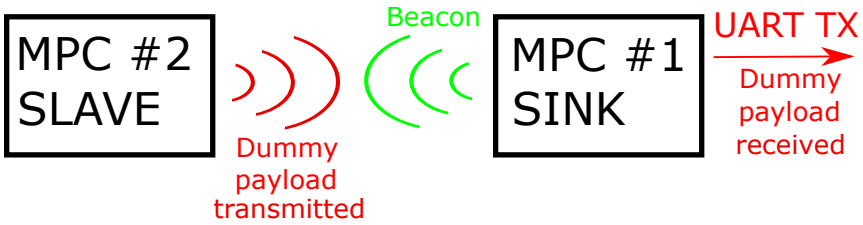


Figure 5.6 – Sketch of the dummy payload transmission from the slave to be transmitted on the UART pin of the sink.

The UART was equipped with two 8 bytes FIFO registers, respectively for transmission and reception. If 50 bytes were directly sent to the FIFO, because of the low UART speed, only the first 8 bytes were transmitted and the remaining lost.
The first approach for the transmission consisted in a polling operation: each byte of the payload was pushed into the UART FIFO only when the empty flag of the latter was active. As shown in the sequence diagram of figure 5.7, with a Baud Rate set to 460800, the time needed to transmit 50 bytes and thus the blocking time was 870 μs (multiplied for 32 slots it became almost 30 ms). As shown in the previous section (Table 5.1), the time for additional operations was 13 ms: the polling approach implied an overflow that stopped the communication after the first beacon interval.

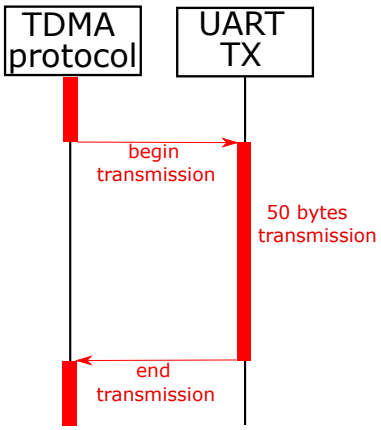


Figure 5.7 – Sequence diagram of the blocking buffer used to transmit a payload through UART.

In order to minimize the blocking time for the TDMA protocol, a second approach based on the interrupts of the UART FIFO, inspired by (36), was used:

- In the callback function related to a packet reception, the empty FIFO interrupt of the UART transmission was enabled.

- An interrupt handler associated with the previous interrupt condition loaded 8 bytes belonging to the payload on the FIFO, a counter was updated and the TDMA software was run again while the UART was transmitting the data previously pushed.

- The previous operation was repeated until the whole payload transmission (the counter reached a threshold), at that point the interrupt of the empty FIFO was disabled.

The sequence diagram of this approach is shown in figure 5.8.

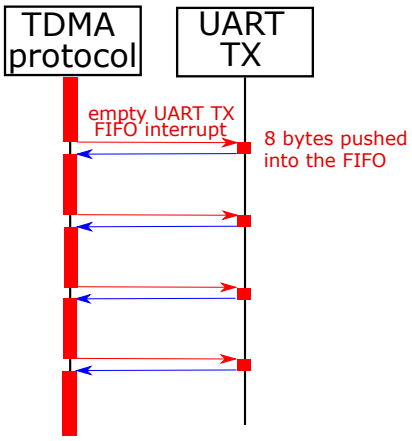


Figure 5.8 – Sequence diagram of the non-blocking buffer based on interrupts used to transmit a payload through UART.

To measure the time required to push 8 bytes into the FIFO, a GPIO was set to 1 only during the interrupt handler activation (figure 5.9).

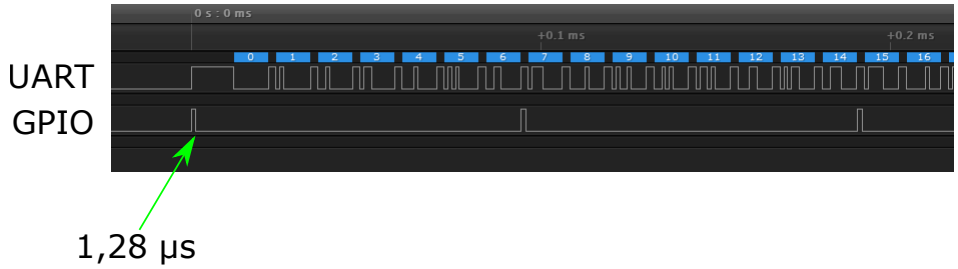


Figure 5.9 – Measurements on the toggling GPIO to calculate the time reserved for calling the interrupt handler.

The blocking interval during each interrupt handler activation was 1.28 μs; considering that to send 50 bytes the interrupt handler was activated 7 times, the blocking time for a whole payload transmission was around 9 μs, only 1% of the first approach blocking interval, thus allowing to not overflow the beacon interval, ensuring a stable communication between the nodes.

A review of the blocking times in the two approaches is shown in Table 5.2.

Tx approach	Blocking time per slot	Blocking time per beacon interval
Blocking buffer (Polling)	870 μs	27.8 ms
Non Blocking buffer (interrupt based)	9 μs	288 μs

Table 5.2 – Overview of the timing performances of the two buffers adopted for the UART transmission of wirelessly received data.

5.2.2 UART reception

The interface for the data received from an external source on the UART port of the slave was developed in the main software and not on the TDMA one. As previously explained, there were no time constraints in such software and, as a consequence, a polling approach was used without issues to store data to be sent in a queue fired during each slot interval.

5.3 Final demonstration

Figure 5.10 – Arrangement of the in-house RF demo.

The setup shown in figure 5.10 was used to test the new interfaces used to receive EMG signals with the slave connected to the robotic apparatus and to send them to a PC via the sink. Considering that this was an unofficial demonstration, the EMG board configured in chapter 4 was used as input for the RF board instead of the robotic hand

apparatus.

A final example of the communication between nodes, recorded with a logic analyzer, is shown in figure 5.11. It is interesting to observe the beacon transmission and reception, the EMG data transmission on the slave, their reception on the sink and the re-transmission on the UART.

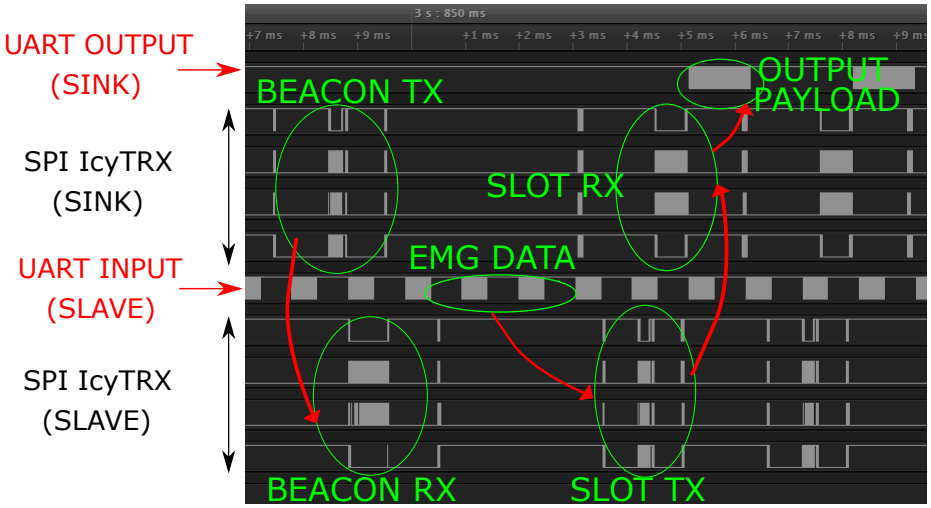


Figure 5.11 – Logic analyzer measurements of the communication between sink and slave nodes.

Once the EMG data were correctly received by the slave and sent to the sink, it was possible to display them on the PC thanks to the real-time Matlab script used in section 4.4, shown in figure 5.12.

This demonstration successfully showed the accomplishment of the wireless monitoring of the robotic prosthesis integrated on a patient.

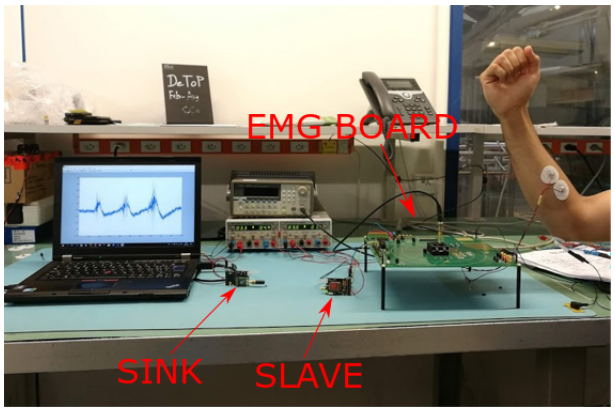


Figure 5.12 – Final example of the working wireless protocol with the UART interfaces.

5.4 Two-way communication

As mentioned in section 2.5, the wireless module should be able to monitor the control of the robotic arm and eventually send from a PC a sequence of bytes able to change the configuration of the hand (figure 5.13).

A description of the correct hand configuration from the partners was still missing during this experience. In order to prove the feasibility of the two-way transmission, the same non-blocking buffer shown in section 5.2.1 was used to output on the UART the beacon received from the slave while the EMG data transmission was still ongoing. This test showed a stable communication and confirmed the possibility to recycle the non-blocking buffer once the bytes needed for the hand configuration will be clarified by the partners.

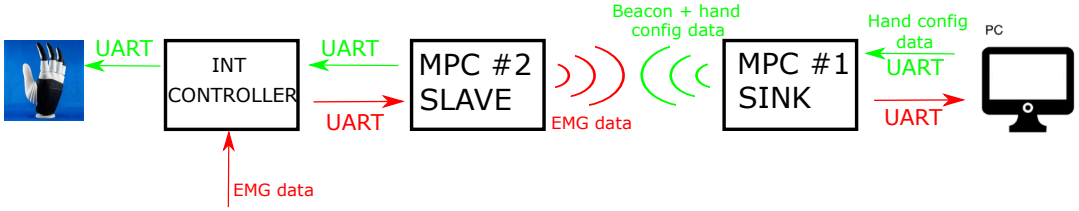


Figure 5.13 – RF demo setup including the double ways communication.

6 DeTOP follow-up

The demonstrations shown in the previous chapters needed components developed by external companies (Integrum AB and Prensilia s.r.l.) not available for CSEM. The idea behind this chapter is to create a new setup totally owned by the company to replicate the previous demonstration and, eventually, for new interesting purposes.

6.1 Pattern Recognition configuration

During the EMG demo (section 2.5) an open-source software named BioPatRec (37) was used to perform pattern recognition for the electromyographic signals through the hand controller.

The first step towards the creation of an independent setup was to connect the UART of the SoC through FTDI cable directly into the PC with BioPatRec installed, bypassing the Integrum controller.

The software loaded on the SoC was very similar to the one used for the EMG demo (8 AFE channels, sampling frequency 1ksps each), the only difference was in the output format: no more initial headers for each packet and the ADC output, initially coded on 12 bits, was compressed on 1 byte (the 4 LSBs were removed) to have compatibility with the software on the PC.

Three movements (hand flexion, extension and rest) were recorded on BioPatRec for a

total of 10 repetitions each. The latter was used for the Pattern Recognition training, performed through a MultiLayer Perceptron algorithm (38).
The result of the training is the confusion matrix (39) of figure 6.1. The closer is the square color to yellow, the higher is the number of movements associated. The emerging yellow diagonal confirms the good quality of the pattern recognition, showing that the CSEM SoC could replace the Integrum controller for the training of the movements.

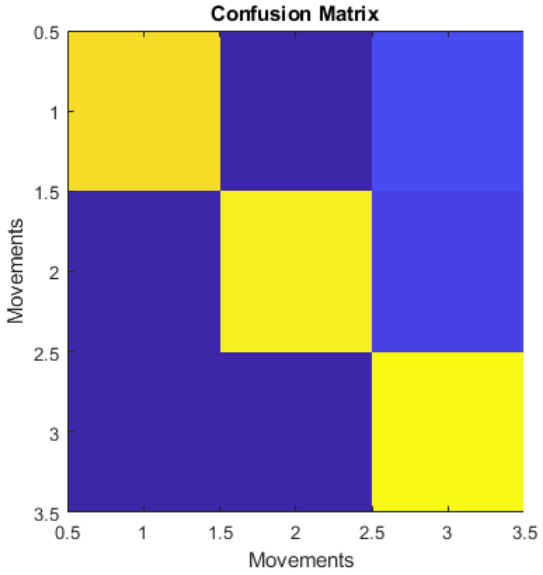


Figure 6.1 – The confusion matrix recorded with BioPatRec.

6.2 Hand control setup

As previously shown, it is possible to train a network with the single SoC in order to recognize specific movements of the hand through a PC. The final interest would be to make the setup wearable without the need of being plugged to a PC.
The problem related to the development of a fully independent and wearable setup is the RAM of the SoC, 256 kB only, making it challenging to develop complex Machine Learning algorithms on it. The idea is to integrate on the setup another chip containing a neural network whose weights are uploaded from the PC after the training of BioPatRec. A first option, shown in figure 6.2, could be to start the development of

this device on an FPGA, as explained in (40): after the training of the network, during the recognition of a pattern it will be possible to send a signal to a robotic hand (a model already available on the market to be purchased by CSEM) in order to perform a specific movement.

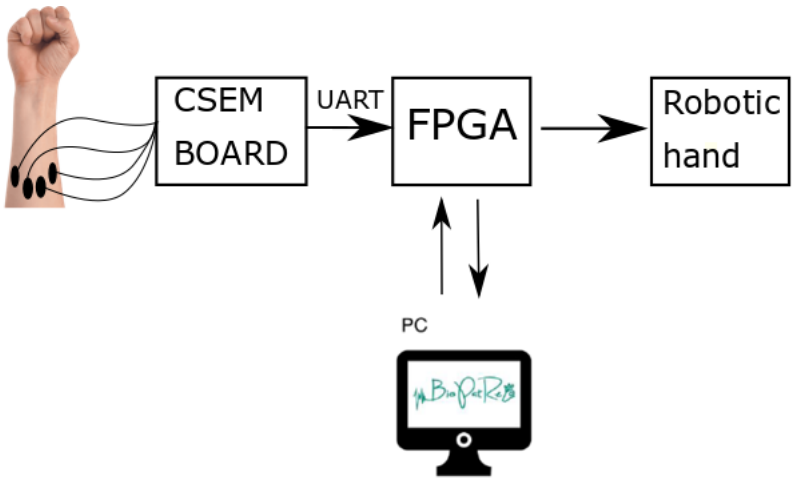


Figure 6.2 – In-house implementation of the real-time control of a robotic hand.

Eventually, the same nodes used for the RF demonstration (chapter 5) could be easily integrated into the robotic hand for the real-time control and a possible reconfiguration through PC or smart-phone. The challenge would be to merge the software for the wireless protocol and the EMG sensing.

6.3 New possible scenarios

The use of the SoC developed by CSEM is not only limited to applications for prosthesis control but opens a plethora of options in the field of wearable sensors.
A possible new purpose could be related to a bracelet containing dry electrodes: their main advantages would be the multi - reusability and their ease of application. As shown in (41) the main challenge would be related to a higher skin-electrode impedance and a higher sensitivity to motion artifact. An example of such a bracelet, already available on the market, is shown in figure 6.3.

Figure 6.3 – The bracelet containing dry electrodes where the EMG sensing device could be integrated. Adapted from (42).

A possible implementation of the new bracelet setup could contain the MPC connected to a hardware accelerator, preferably a specific purpose IC more than an FPGA for its miniaturization, in order to perform pattern recognition of electromyographic signals.

A hypothesis is shown in figure 6.4: the ideal system would consist of a single MPC able to implement both sensing and wireless module. Thanks to a Bluetooth connection, from the other side of the communication a wide range of devices could be used: smart-phones, computer or television are the first examples.

The application could space from muscular rehabilitation to virtual/augmented reality for performing a manual work from remote: the perfect illustration would be a surgeon operating a patient on the other side of the planet thanks to the control of a laser, a blade or other specific tools.

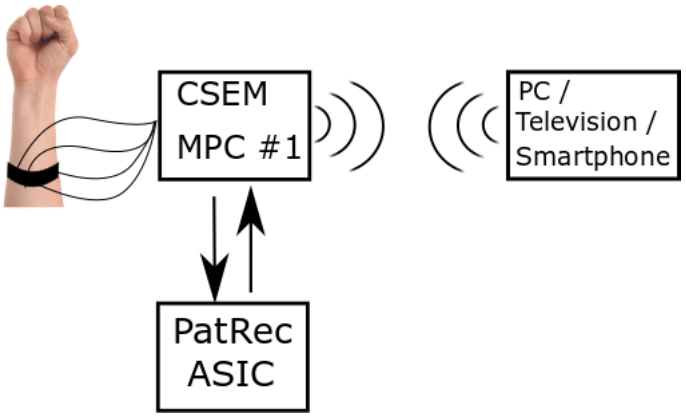


Figure 6.4 – The bracelet containing the EMG sensor able to communicate with an external device for new virtual/augmented reality applications.

7 Conclusion

The final results of this project were the validation and configuration of a SoC used for the control of a robotic hand through sensing of electromyographic signals and the establishment of a two-way wireless connection between the robotic hand and a PC. Aim of this chapter is to provide the reader with a short review of the main results obtained and what could still be improved.

7.1 Summary of the robotic control

During this experience, also thanks to constant communication with the partners of the project, it was possible to successfully configure the output of the sensing device in order to have the correct format and sampling frequency compatible with the controller of the robotic hand.
The final accomplishment of this task turned out to be very challenging and the main results could be summarized in the next points:

- A global insight of the device (both HW and SW) was achieved in order to understand where a problem could have been generated.
- New innovative setups were developed to perform several electrical validation tests on the PCBs and real-time analysis of the EMG signals outputted by the

sensing chip.

- A final demonstration with the partners was performed, showing the control of the robotic apparatus through electrodes attached to a patient.

7.2 Summary of the wireless monitoring

The aim of the wireless part was to integrate into the robotic prosthesis connected to the patient the same board used for the hand control with an external radio attached on it in order to establish a two-way wireless communication with a second board connected to a PC. More specifically, the control signals of the robotic hand had to be sent to a PC and the latter had to eventually send data to the hand's controller for a reconfiguration of the global settings.
The following results characterized the wireless part of this experience:

- After a full comprehension of the low power wireless module, the software was adapted to some modifications of the hardware.
- The input/output interfaces for the reception and transmission of external data were created.
- A final demonstration showing the real-time signals for actuating the prosthesis wirelessly streamed on a PC was successfully performed.

7.3 Next improvements

There is still room for improvements in both preceding parts.
For the robotic control, the quality of the EMG signals captured by the SoC could be improved by performing some data processing applications like filtering of ambient noise, averaging the ADC output dropped during the down-sampling or by performing some memory efficient Machine Learning algorithm for Pattern Recognition. It could

be possible to improve the signals acquired also by changing the kind of electrodes or their arrangement.
For the wireless part, the official demonstration (with the controller and robotic hand owned by the partners) is still missing. As soon as the timing constraints and the bytes format to re-configure the controller will be clarified from whom developed the hand controller, by using the same interfaces created during this experience and the blocks owned by our partners it will be possible to complete this last part.
When the RF demo will be officially performed with the partners and the prosthesis will be connected to the patient the DeTOP project will be concluded.

Bibliography

[1] Branemark, Rickard, et al. "Osseointegration in skeletal reconstruction and rehabilitation: a review." *Journal of rehabilitation research and development* 38.2 (2001): 175-182.

[2] Brånemark, R., et al. "A novel osseointegrated percutaneous prosthetic system for the treatment of patients with transfemoral amputation: A prospective study of 51 patients." *The bone & joint journal* 96.1 (2014): 106-113.

[3] Ortiz-Catalan, Max, Bo Håkansson, and Rickard Brånemark. "An osseointegrated human-machine gateway for long-term sensory feedback and motor control of artificial limbs." *Science translational medicine* 6.257 (2014): 257re6-257re6.

[4] Detop-project.eu. (2019). *Summary and Figures*. Available at: http://www.detop-project.eu/summary-and-figures/.

[5] Reaz, Mamun Bin Ibne, M. S. Hussain, and Faisal Mohd-Yasin. "Techniques of EMG signal analysis: detection, processing, classification and applications." *Biological procedures online* 8.1 (2006): 11.

[6] Farina, D., R. Merletti, and C. Disselhorst-Klug. "Multi-channel techniques for information extraction from the surface EMG." *Electromyography: Physiology, engineering, and non-invasive applications* (2004): 169-203.

[7] Motion-labs.com. (2019). *Selecting an EMG System*. Available at: https://www.motion-labs.com/movie/Selecting_EMG_systems.pdf.

[8] De Luca, C. J., et al. "Behaviour of human motor units in different muscles during linearly varying contractions." *The Journal of physiology* 329.1 (1982): 113-128.

[9] Dorcas, D. S., and R. N. Scott, “A three-state myoelectric control.” *Med Biol Eng* 4, 367 (1966).

[10] Detop-project.eu. (2019). *The first dexterous and sentient hand prosthesis has been successfully implanted*. Available at: http://www.detop-project.eu/news/the-first-dexterous-and-sentient-hand-prosthesis-has-been-successfully-implanted/.

[11] Motion-labs.com. (2009). *EMG analysis and EMG graphic software user tutorial*. Available at: https://www.motion-labs.com/pdf/emgtutorial_ug.pdf.

[12] Ortiz-Catalan, Max, et al. "On the viability of implantable electrodes for the natural control of artificial limbs: review and discussion." *Biomedical engineering online* 11.1 (2012): 33.

[13] Clancy, Edward A., Evelyn L. Morin, and Roberto Merletti. "Sampling, noise-reduction and amplitude estimation issues in surface electromyography." *Journal of electromyography and kinesiology* 12.1 (2002): 1-16.

[14] Webster, John G. "Reducing motion artifacts and interference in biopotential recording." *IEEE Transactions on Biomedical Engineering* 12 (1984): 823-826.

[15] Total Phase Blog. (2016). *Understanding the Difference Between UART and SPI*. Available at https://www.totalphase.com/blog/2016/06/spi-vs-uart-similarities-differences/.

[16] Pons, M., K. Badami, J. Deng, E. Azarkhish, L. Zahnd, C. Cosentino, L. Bergamini, P. Dallemagne, and S. Emery, “A 20 Channel EMG SoC with an Integrated 32b RISC Core for Real-Time Wireless Prosthetic Control.” *IEEE ESSCIRC, Session “Biomedical Interfaces”* (2019).

[17] Paret, Dominique, and Carl Fenger. *The I2C bus: from theory to practice*. New York: Wiley, 1997.

[18] Hopkins, Andrew BT, and Klaus D. McDonald-Maier. "Debug support for complex systems on-chip: A review." *IEE Proceedings-Computers and Digital Techniques* 153.4 (2006): 197-207.

[19] Pons, Marc, et al. "Sub-threshold latch-based icyflex2 32-bit processor with wide supply range operation." *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*. IEEE, 2016.

[20] Konijnenburg, Mario, et al. "22.1 A 769μW Battery-Powered Single-Chip SoC With BLE for Multi-Modal Vital Sign Health Patches." *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019.

[21] Van Helleputte, Nick, et al. "A 345 μW multi-sensor biomedical SoC with bio-impedance, 3-channel ECG, motion artifact reduction, and integrated DSP." *IEEE Journal of Solid-State Circuits* 50.1 (2014): 230-244.

[22] Intantech.com. (n.d.). *RHD2132 / RHD2216 Amplifier Boards*. Available at: http://intantech.com/RHD2132_RHD2216_amp_board.html.

[23] Mastinu, Enzo, et al. "Embedded system for prosthetic control using implanted neuromuscular interfaces accessed via an Osseointegrated implant." *IEEE transactions on biomedical circuits and systems* 11.4 (2017): 867-877.

[24] Prensilia. (n.d.). *Tech specs of Mia hand*. Available at: https://www.prensilia.com/portfolio/mia/.

[25] V. Peiris et al., "An Ultra-low-power Bluetooth Smart Integrated Solution," *CSEM Scientific and Technical Report*, 2012.

[26] Global Knowledge. (n.d.). *The OSI Model: Understanding the Seven Layers of Computer Networks*. Available at: http://ru6.cti.gr/bouras-old/WP_Simoneau_OSIModel.pdf.

[27] Briscoe, Neil. "Understanding the OSI 7-layer model." *PC Network Advisor* 120.2 (2000).

[28] H. Wu and Y. Pan, *Medium Access Control in Wireless Networks*, H. Wu and Y. Pan, Eds. Nova Science Publishers, 2008.

[29] Bachir, Abdelmalik, et al. "MAC essentials for wireless sensor networks." *IEEE Communications Surveys & Tutorials* 12.2 (2010): 222-248.

[30] Bergamini, Lorenzo, Jean-Dominique Decotignie, and Philippe Dallemagne. "WiseTOP: a multimode MAC protocol for wireless implanted devices." *Proceedings of the 26th International Conference on Real-Time Networks and Systems*. ACM, 2018.

[31] El-Hoiydi, Amre, et al. "WiseMAC, an ultra low power MAC protocol for the wiseNET wireless sensor network." *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003.

[32] Falconer, David D., Fumiyuki Adachi, and Bjorn Gudmundson. "Time division multiple access methods for wireless personal communications." *IEEE Communications Magazine* 33.1 (1995): 50-57.

[33] Jameco, (n.d.). *ST M25P32 NVM datasheet*. Available at: https://www.jameco.com/Jameco/Products/ProdDS/697928-DS01.pdf.

[34] Ftdichip. (n.d.). *USB TTL Serial datasheet*. Available at: https://www.ftdichip.com/Products/Cables/USBTTLSerial.htm.

[35] De Luca, Carlo J. "Surface electromyography: Detection and recording."*DelSys Incorporated* 10.2 (2002): 1-10.

[36] Beningo Embedded Group. (2015). *Embedded Basics – Blocking vs Non-Blocking Drivers*. Available at: https://www.beningo.com/embedded-basics-blocking-vs-non-blocking-drivers/.

[37] Ortiz-Catalan, Max, Rickard Brånemark, and Bo Håkansson. "BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms." *Source code for biology and medicine* 8.1 (2013): 11.

[38] Rathbun, Thomas F., et al. "MLP iterative construction algorithm." *Neurocomputing* 17.3-4 (1997): 195-216.

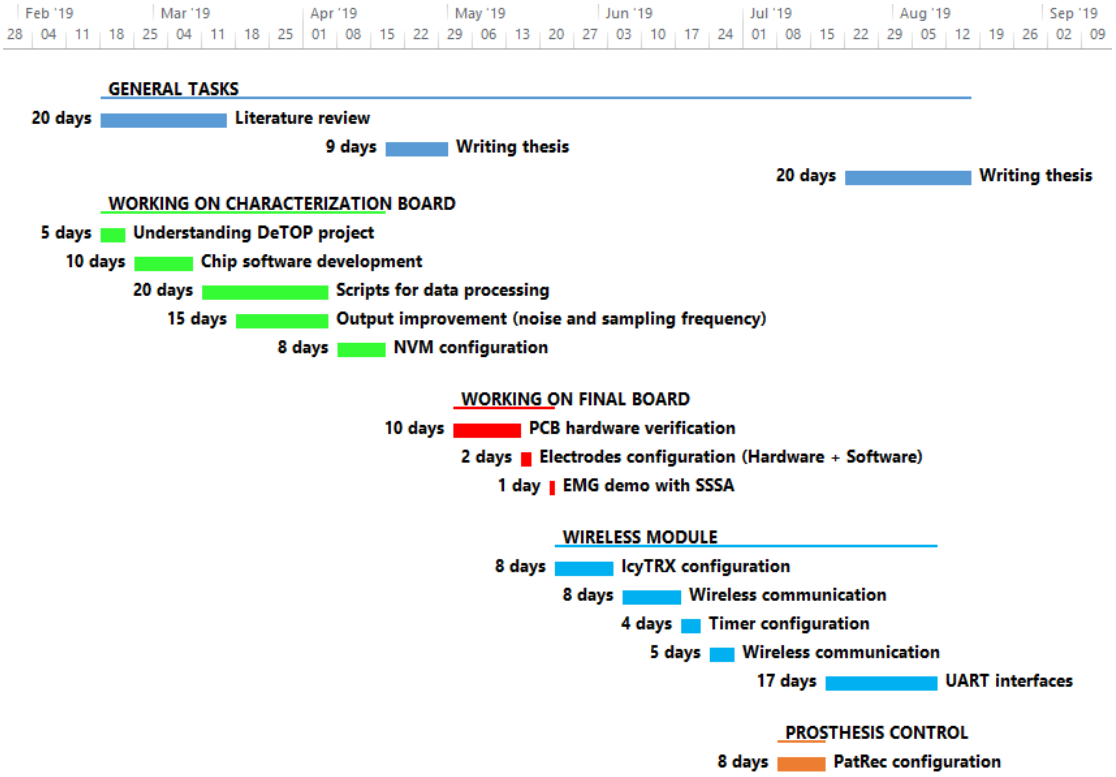
[39] Ting, Kai Ming. "Confusion matrix." *Encyclopedia of Machine Learning and Data Mining* (2017): 260-260.

[40] Shima, Keisuke, and Toshio Tsuji. "FPGA implementation of a probabilistic neural network using delta-sigma modulation for pattern discrimination of EMG signals." *2007 IEEE/ICME International Conference on Complex Medical Engineering*. IEEE, 2007.

[41] Laferriere, Pascal, Edward D. Lemaire, and Adrian DC Chan. "Surface electromyographic signals using dry electrodes." *IEEE Transactions on Instrumentation and Measurement* 60.10 (2011): 3259-3268.

[42] Oymotion Technologies. (n.d.). *gForce-Pro EMG Armband specs*. Available at: https://oymotion.github.io/assets/downloads/gForcePro_spec_v1.0-eng.pdf.

Appendix A: Gantt Diagram



Appendix B: Matlab Pseudo Code for Real Time EMG and FFT

```
Baud_Rate    =    460800;
data_Bits    =    8;
Stop_Bits    =    1;
start_time   =    0;

S = serial('COM6','BaudRate', Baud_Rate, 'dataBits', data_Bits, 'StopBits',
  Stop_Bits);

cyclecount   =    0;
channel_numb=     1;
n            =   30;
j            =    1;

fopen(S);
data_plot    =    0;

while(1)
        for i=0:n
           dat(i*512+1:(i+1)*512)=fread(S);    %string of 512*n bytes every cycle
        end

         numb_bytes=length(dat);

         k=1;

         for i=1:numb_bytes %decoding

                if(sequence_of_headers_is_found_in_dat(dat,i))
                     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                     %in this point decode the next
                     %16 bytes on 8 different arrays
                     %representing the 8 channels
                     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                end

         end



         %%%%%%%% SINGLE CHANNEL FFT %%%%%%%%%%%%%%%%
         T                    = 0.001; %ADC sampling period (seconds)
         Fs                   = 1/T; %sampling freq
         L                    = length(channel_1); %number of samples of a single
```

```
  channel
43          time                = (0:L-1)*T;
44          frequencies         = Fs*(0:(L/2))/L;
45          Y                   = fft(channel_1);
46          Norm_fft            = abs(Y/L);           %normalization
47          Positive_norm_fft   = Norm_fft(1:L/2+1); %only positive frequencies
48
49
50          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51          %in this point multiple plots of the 8
52          %channels previously decoded and the FFT
53          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
54
55           if(cyclecount<5)          %used to constantly update the plots
56               start_time=0;         %to make it Real-Time
57           else
58               start_time= (length(data_plot) -n*100);
59           end
60        drawnow;
61        pause(0.01)
62        cyclecount=cyclecount+1;
63
64  end
```