FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Development of a Smart Sensor Node based on BITALINO

Pedro Tavares

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Gil Gonçalves

Co-Supervisor: João Reis

Co-Supervisor: Rui Pinto

August 11, 2016

Development of a Smart Sensor Node based on BITALINO

Pedro Tavares

Mestrado Integrado em Engenharia Informática e Computação

Chair: Pedro Souto

External Examiner: Jorge Cardoso

Supervisor: Gil Gonçalves

July 26, 2016

Abstract

Biometric data gathering has gained significance of the past years. The aim of the dissertation is to extend this concept to an industrial setup and take advantage of it to leverage workload and allow for a faster response for any kind of events that happen in such context. The main idea to explore is the concept of a reactive environment where all agents, human and non-human, are interconnected and working towards the same goal.

Nowadays, biometric data gathering equipment is widely available and low cost. This reality allows for the development of this project where a low cost solution is applied to an industrial context.

This specific industrial context is one where machines are monitored and connected to a network. It's in such environment that human tracking fits and this new source of data is mixed will the existing ones, allowing a factory to become fully responsive.

Acknowledgements

I personally want to thank professor Gil Gonçalves for giving me the chance to take this project and for the given guidance throughout the past months. João Reis and Rui Pinto for all the technical help and guidance and overall positivism towards the project, as well as for their great efficiency whenever I had to mix my work with theirs. Last, but not least, to my closest family and friends that accompanied and supported me during this stage of my life.

Pedro Tavares

“Life is whatever we conceive it to be.”

Fernando Pessoa

Contents

List of Figures

LIST OF FIGURES

Abbreviations

EEG	Electroencephalogram
ECG	Electrocardiogram
EMG	Electromyogram
EOG	Electrooculography
GSR	Galvanic skin response
MEG	Magnetoencephalogram
MMG	Mechanomyogram
OTA	Over the air
PMS	Patient Monitoring System
WWBAN	Wireless Wearable Body Area Network
HRV	Heart Rate Variability
MAC	Media Access Control

Chapter 1

Introduction

This dissertation falls under the biometric data gathering field of study. It consists in inserting the human element in an already existing network of sensors. As any other living form, biosignals of many kinds can be extracted from humans and studied afterwards.

1.1 Context

This dissertation lies into an industrial context, where the machines are monitored. The machine's sensors create a sensor network that can be supervisioned externally. This allows a manufacturing system to be self-sustaining with automatic health monitoring mechanisms. With the added human element, all the elements in the industrial environment become monitored and capable to react to their surroundings.

Industry is one of the pillars of the European economy, the manufacturing sector alone accounts for 2 million enterprises, 33 million jobs and 60% of productivity growth, in the European Union.

It is no surprise that the technological revolution of the past years is capable of bringing benefits to the industry and, given that it is one of the backbones of the world economy, any positive change can have a very positive effect on society itself.

The work developed in this dissertation directly interacts with the SelSus project, using one of its sensor networks as an example of a network that maps both machines and human operators.

1.2 Motivation

The main motivation for this work is to include the human element to a sensor network. Having access to the worker's health condition can allow for better coordination and task scheduling. Also, health concerns can be assessed immediately as a result of altered state of a worker, this can lead to a safer workplace and safer work methods.

Active monitoring of workers in assembly lines is a concept that can have a positive effect on their work.

Sensor networks are represented by multiple different sensors. The challenge of correlating them altogether, specially biometric sensors, tests the network flexibility to adapt to new kinds of sensors and new types of data.

One of the main ambitions of this dissertation consists in having a working prototype using low cost material by exploring the Bitalino's and Raspberry Pi's capabilities. A low cost solution for human biometric data makes the concept more appealing and desiring for enterprises that intend to implement such systems.

1.3 Objectives

This dissertation aims to deliver a solution to track workers in a factory with a sensor network. The data gathered from humans should be configurable just like any other node in the network and the biometric data should go through a first stage of gathering data locally and later uploading it to the cloud in batches. A process in the cloud should be responsible for processing the biometric data and extracting valuable information from it. This flow of information is aimed to have a direct positive influence in the production process and facilitate the human-machine interaction.

The purpose of this implementation is to provide a safer workplace for workers in assembly lines, where machines become reactive to the human presence and act accordingly.

This work aims also to simulate a real world scenario where the sensor node and the network can be put at test to confirm that its feasible to conciliate human sensorial data with machine sensors. This real world simulation should consist of a registered node capturing biometric data, a processing unit capable of extracting valuable information out of the gathered data, and a working machine that adapts to its surroundings, namely the operator around it. By doing such, the whole data flow, from human to network, becomes operational and extensible to multiple scenarios. More precisely, the work developed in the context of this dissertation intends to display one robotic arm adapting its speed according to the condition of the person working alongside it.

This dissertation intends to prove that adding human sensor data to a sensor network can be advantageous and display actual benefits of such integration.

1.4 Dissertation Structure

Besides the introduction, this dissertation contains 5 more chapters. In chapter 2, the state of the art of biosignals, health monitoring and sensor networks is described and related projects are presented.

Chapter 3 presents the methods of analysis of ECG data throughout its different stages, from capturing to processing.

In chapter 4, the specific implementation details are presented. The communication between the different devices is explained as well as data processing.

In chapter 5, the pratical example is explained and its results are displayed.

The conclusion chapter 6 features the main difficulties faced, the conclusion of the work that was developed, next steps of work and personal notes.

Chapter 2

State of the art

It is hard to come up with a straightforward definition for biometric systems due to their variety of applications, although common principles are shared between them. Input is typically gathered from a subject by a sensor and the collected data is processed afterwards.

2.1 Biosignals

Biosignals are signals emitted by living beings that can be actively monitored and measured. Typically, these signals are referent to change in electric current by the sum of an electrical potential difference across a specialized tissue, organ or cell system. Some of the most well known bioeletrical signals are:

- Electroencephalogram (EEG)
- Electrocardiogram (ECG)
- Electromyogram (EMG)
- Mechanomyogram (MMG)
- Electrooculography (EOG)
- Galvanic skin response (GSR)
- Magnetoencephalogram (MEG)

Nowadays, the ability to measure biosignals is at the reach of many because it no longer requires expensive equipment at specialized environments. In fact, with the advances made in wireless technologies, many products that track information like heart rate, sleep, steps and much more, have been publicly released and are publicly available.

To actively monitor workers, a reliable device is need like the ECG systems based on low energy Bluetooth have been documented by Bin Yu et al. [1]. By combining an ECG signal acquisition module, a Bluetooth module and a smartphone, the system is able to transmit ECG data, process and display it on a smartphone. BLE (Bluetooth Low Energy) technology allows this system to work for a long period of time.

Other approach to human monitoring consists in adapting everyday wearables to monitor a patient is also its own field of study, as shown by Lee, Y.D. and Chung, W.Y. [2]. The shirt can read ECG signals, as well as acceleration, in real time. The electronic components consume low power and are small enough to fit into a shirt.

2.2 Biosignals storage

Storing biosignal data is fundamental in order to be able to analyze and compare it. However, there is still no *de facto* standard for biosignal data, although there have been multiple attempts. Standardization allows for better understanding of the scope of the problems, which leads to better solutions, it facilitates the exchange of data as well. Most recently, the focus has shifted for more semantic approaches, where not only core biosignal data is stored, but also contextual information, such as the user and acquisition procedure. According to E. Kohlwey, et al. [3], the biometric databases of the Federal Bureau of Investigation, alongside the US State Department, Department of Defense and the Department of Homeland Security, are expected to grow significantly over the next years to accommodate several hundred millions of identities. Expectations like this make it necessary to prepare an highly scalable biometric technology, capable of handling and storing huge amounts of data.

2.3 Cloud Computing

Cloud computing, as defined in “The NIST Definition of Cloud Computing” [4], emerged as a model that enables on-demand access to a shared pool of multiple computing resources that are easily provisioned and released.

There are three service models and four deployment models according to the NIST definition of Cloud Computing. The service models are as follows:

- Software as a Service (SaaS). Allows the consumer to use the provider’s application running on a cloud infrastructure.
- Platform as a Service (PaaS). This model provides the capability for the consumer to deploy consumer-created or acquired applications onto the cloud.
- Infrastructure as a Service (IaaS). The capability for the consumer to manage computer resources like processing power, memory, networks and other fundamental computer resources where arbitrary software can be deployed to.

The deployment models are:

- Private cloud. Cloud infrastructure provisioned for the use of a single organization that may comprise multiple costumers.

- Community cloud. Cloud infrastructure provisioned for the use of a specific community of consumers from organizations that have shared concerns. It may be owned and managed by one or more organizations from the community, a third party of a combination of them.

- Public cloud. Cloud infrastructure provisioned for the use of the general public. It can be owned by an business, governmental or academic organization, or a combination of them.

- Hybrid cloud. Cloud infrastructure composed of two or more distinct cloud infrastructures (private, community or public) that are unique entities from one another, but are tied together by standardized or proprietary technology.

Such characteristics make the cloud platform highly suitable form various applications, including biometrics.

2.3.1 Biometrics in the cloud

Networking is a concept that has always been paired with biometric systems. Heavy computing work related to biometric data is typically done by an external, more capable, system. The usage of a cloud based infrastructure also make the system highly scalable, prepared to store huge amounts of data.

When developing a cloud-oriented biometric solution, many design choices have to be made. Perhaps the most relevant decision is deciding which components should be moved to the cloud and which should be implemented locally. Commercial solutions, like BioID [5] and Ceelox ID [6], tend to operate in a client-server model. The client is responsible for capturing biometric samples from the users and sending it to the server, where the matching process is executed. Security protocols are used to guarantee the integrity and authenticity of both client and server.

Parane, KA et al. [7] have designed a cloud based health care monitoring system that allows a patient to receive medical feedback. The patient's condition is actively monitored and uploaded to a remote location. This resembles the way a sensor network operates, where machines actively report their status and actions can be determined based on it.

2.3.2 Sensor Network

Sensor networks represent the foundation for this dissertation. They are made of sets of sensor nodes that operate in independent manner but allow for distributed storage. The data generated by the network is relevant to the operational behavior of the factory and it has the potential of providing important information that may be used to predict system's failure or optimize power usage.

According to Neto, L. et al. [8], the sensor cloud is a virtualization of multiple physical sensors, allowing the remote control of them via web browsers. The cloud is composed by three layers:

- Sensor Integration Layer
- Logical Layer
- User and Application Layer

It allows users to take advantage of sensors by having the monitoring and data gathering capabilities, remotely via their web browsers, without worrying about their location or specifications.

The mapping and monitoring of an whole assembly line allows the responsiveness in cases of failure, for example, when a machine fails, the whole assembly line can be automatically stopped.

2.4 Real time wireless health monitoring

Mobile health monitoring is a service modern healthcare systems look after to provide patients in an economic and friendly manner. Currently, there has been a cultural shift towards a modernized patient centered approach. In the traditional approach, health care professionals played the major role, with the constant need to actively visit the patients for necessary diagnosis and advising. This last approach has two major problems associated. First, the healthcare professionals must be on site of the patient all the time and, secondly, the patient has to stay at the hospital facilities, wired to existing biomedical instruments. The patient oriented approach has been conceived to circumvent both these problems. With this approach, patients are more knowledgeable so they can have a more active role in disease diagnosis and prevention. The key element that makes this approach possible is a patient monitoring system (PMS).

A modern PMS should encapsule the advantages of modern bioinstrumentation, computers and telecommunication technologies in order to acquire, record, display and transmit the biometric data from the patient's body to a remote location at any given time. For a more efficient emergent medical care, the PMS must also possess an alarm system that should be able to warn the patient and the health care service providers. The health services's system must be prepared to permanently store the data. By using necessary firmware and software, the server can be connected to an open communication network via TCP/IP protocol. Thus, a patient can be monitored from a remote location.

Bourouis, A. et al. [9] presented a real time health care system monitoring for the elderly patients for both indoor and outdoor locations. It relies on a bio-signal sensor and a smartphone. The existing mobile network is used to transmit sensor data to the system's server. This solution relies on the concept of Wireless Wearable Body Area Network (WWBAN) that consist of one or more sensors attached to the body of the patient that gather and transmit biometric data to a second component named Intelligent Central Node (ICN). A regular smartphone has the necessary characteristics, Bluetooth and mobile network connections, to perform the ICN role. This star topology clearly displays a centralized architecture where the processing is concentrated in a central node.

Additionally, an Intelligent Central Server (ICS) maps and gathers data from all the ICN. Data related to the patients is stored in a database to become available for analysis. Such analysis is performed autonomously, without human intervention, comparing the patient's vitals against previous records of the same patient alongside recommendations prescribed by the patient's doctor or healthcare professional.

A solution like the previously mentioned can be adapted to an industrial context. The mobile monitoring equipment already in use for many health related applications could serve workers in a factory. The added challenge of bringing this concept to an industrial context is combining the data gathered from both humans and machines and make automated decisions based on that data, given the heterogeneous nature of the network.

2.5 Emotional detection from biosignals

Pattern recognition techniques can be applied to deduce the emotional status of the carrier. This is proved to be effective to detect the stress level of drivers, where the usage of patterns improved stress recognition of drivers from 62.2% to 88.6% [10], by analyzing data from multiple sensors signals (EMG, ECG, GSR and respiration). To measure stress, a metric for it has to be defined. J. Healey and R. Picard [11] created a continuous stress metric to compare results from different driving sessions under different conditions. The stress effect was modeled as having anticipatory and persistence effects. This implies that before a stressor is observed, there is a stress increase due to anticipatory, monitoring and planning effects. In addition, the physiological effect of a stressor may take several seconds or even minutes to recover. The stress level was calculated from different signals: the normalized mean of the EMG and the normalized mean and variance for respiration, heart rate and skin conductivity on the hand and foot. The first three were normalized by subtracting the mean of a rest period sample before each drive. The skin conductivity signals were normalized by subtracting the baseline minimum and dividing by the baseline range. Heart rate was uniformly sampled and smoothed using a heart rate tachometer.

A similar strategy can be adopted, the targeted purpose is to get fatigue levels as well as extreme peaks that could be correlated to emergency situations. On top of it, the system should be totally configurable so that the processing stages can alter accordingly.

2.6 Sensor Networks

With the technologies advancements over the past years, the concept of Sensor Networks emerged to face multiple sets of existing challenges.

Although monitoring and tracking techniques are now frequently used in industrial environments, they still rarely aim to improve the occupational health and safety. According to Kortuem, et al. [12], health related computing has shown great promise in many domains such as hospitals, elder care and emergency rescue, but it still lacks more prominence in an industrial setting. A great challenge can be found inKortuem the task of designing the technology that fits each industrial organization's formal and informal practices, rules and habits. A greater challenge resides in designing the technology that is able to adapt to each setting.

The advances in micro-electro-mechanical systems technology, as well as wireless communications enabled, according to I.F. Akyildiz, et al. [13], the development of low-cost sensor nodes capable of untethered wireless communication. These small sensor nodes, capable of sensing, processing data and communicating with the surrounding network leverage the idea of sensor networks based on collaborative effort of a varied number of nodes. Sensor networks represent an improvement over traditional sensors, they can be deployed in two different ways:

- Sensors positioned far from the actual phenomenon, capable of distinguishing targets from environmental noise.
- Several sensors with the exclusive function of sensing and emitting captured data. The communications and positions of such sensors are carefully engineered.

Sensor networks consist of various types of sensors, such as thermal, infrared and acoustic, capable of monitoring a wide variety of ambient conditions that include the following:

- Temperature
- Humidity
- Vehicular movement
- Lightning condition
- Pressure
- Soil makeup
- Noise levels
- Speed

The sensor network must be able to adapt to different sensors measuring different kinds of metrics.

2.7 Mixing human and machine data

The aim of this dissertation is to study the added value of adding a set of biometric data to an already existing set of data from different types of machines.

This merge opens new possibilities of human-machine cooperation by making the machines aware of the human presence and condition. For instance, some machines that require big physical effort to operate could try to adapt to the condition of the operator, by adapting their cadence after realizing the fatigue of the human(s) operating it.

On some extreme cases, emergencies could be detected by obtaining signal peaks from workers and available emergency stop mechanisms could be automatically triggered.

With this, machines stop being the only element reporting their condition and all agents in an assembly process become similarly monitored and present in the sensor network.

2.8 Conclusion

Biosignals study has come a long way over the past century and are a reliable source of information regarding living beings. Their different nature raise challenges concerning their computer modeling and storage. They often are correlated with big amounts of data, so scalable options for processing and storage are considered.

Despite the fact real time health monitoring has been in practice for a long time, there has been a huge focus in making the same monitoring available remotely.

Sensor networks have gained popularity with recent technological advancements and widespread availability of sensorial components.

Chapter 3

Methods of analysis

The approach used to gather and process biometric data aims to provide factory workers the possibility of being monitored. One of the objectives was to obtain a feasible solution that was also inexpensive and easily accessible. Such is the case when mixing Bitalino board's and Raspberry Pi's. These low cost solutions provide a capable solution for the given problem.

3.1 Equipment

3.1.1 Bitalino

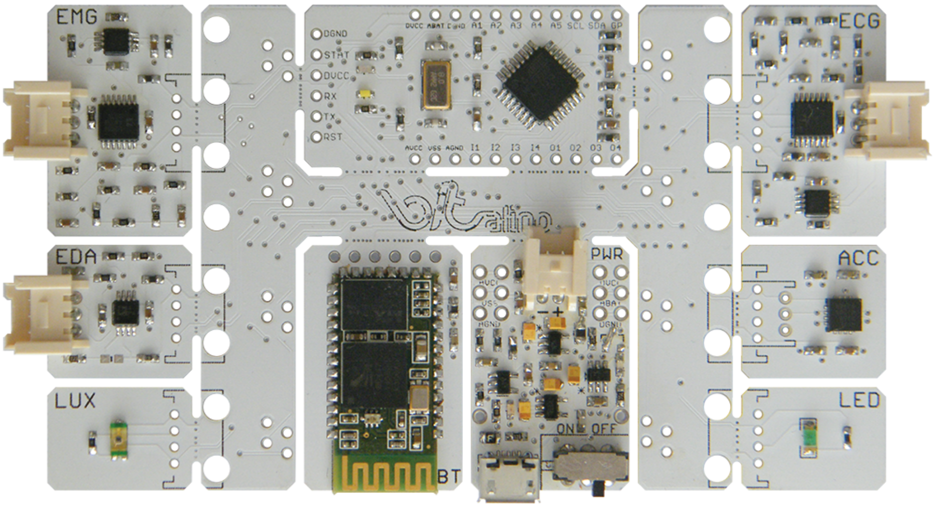


Figure 3.1: Bitalino board

Bitalino is a low-cost and highly versatile hardware framework designed to aid the process of creation applications with physiological sensors.

It is a board with the size of a credit card that integrates multiple measurement sensors for bioelectrical and biomechanical data acquisition. The digital back-end is supported by a control block based on the ATmega328P microcontroller, a power management block, and a communication block that uses a Class II Bluetooth v2.0 module for wireless data transfer.

Bitalino comes, by default, as a single board, with its onboard sensors pre-connected to analog and digital ports on the control block. However, the control, power, and communication blocks, as well as the firmware are completely general purpose, enabling people to use only the digital back-end of the Bitalino with their own custom sensor and actuator designs.

The default configuration of Bitalino brings sensors for EMG, ECG, EDA and accelerometer data gathering.

Bitalino is also available in its Freestyle version that works like the default one but its pre-connected pieces come separated, allowing posterior soldering to create unique custom-made wearables. The Plugged version resembles the default configuration, but additionally has a pluggable cabled connection between the sensors and the main board, allowing for maximum versatility to Bitalino or other third party boards.

3.1.2 Raspberry Pi

The Raspberry Pi is a low cost, also credit card sized, multi-purpose board that can serve multiple purposes. The initial purpose for its creation was to promote the teaching of basic computer science by providing a low cost device that could work just like a normal computer.

This small size computer will be the link between the Bitalino and the Sensor Cloud. It's networking capabilities will allow to interact with the cloud in order to fetch and install new configurations and upload biometric data after processing it accordingly.

The Raspberry Pi Model B is the model used for this project and it features 512 MB of RAM, 700 MHz single-core processor, two USB ports and a 100mb Ethernet port.

A Bluetooth USB adapter was used to make the communication between the Raspberry Pi and the Bitalino board possible.

3.1.3 Smart Sensor Node

The smart sensor node is be made of a Bitalino board and a Raspeberry Pi. The communication between both components is made via Bluetooth. Both boards connected together represent a smart sensor node that should be carried by workers. An external power source is mandatory to guarantee its portability.

A Smart Sensor Node should be configurable. Since different biometric data can be extracted from the Bitalino, the node should be able to filter and process it.

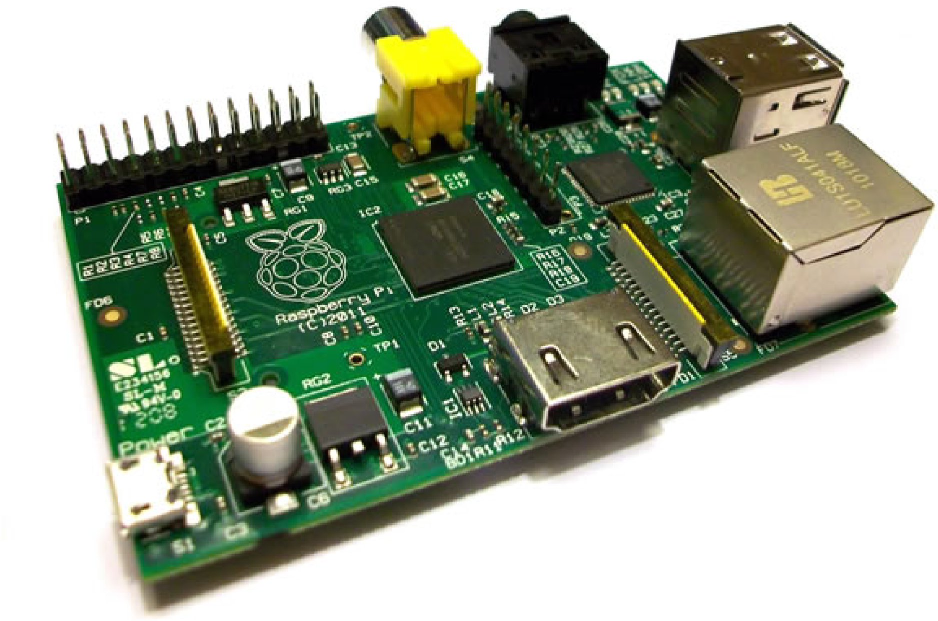


Figure 3.2: Rasbperry Pi B model

3.2 Software

3.2.1 Pi ECG Data gatherer

This program written in Python orchestrates the communication between the Bitalino and the Sensor cloud. The communication to the Bitalino board is made via Bluetooth, and via HTTP requests to the Sensor Cloud.

The process starts by registering itself in the sensor network, with its description and afterwards tries to establish connection to the Bitalino board.

It communicates with the Bitalino to set its state, frequency and mode, as well as keeping track of the obtained samples from the frames sent by the board. The frames are deconstructed and the ECG raw data is extracted. The raw ECG data is gathered and transmitted in chunks inside a payload to the sensor cloud.

3.2.2 Matlab

Matlab is a platform optimized for solving mathematical problems. It's a matrix-based language, making it a natural fit to express computational mathematics. Built-in graphics capabilities also allow data visualization for better insights. It comes out of the box with prebuilt toolboxes that allows the usage of algorithms for a wide range of domains. The raw ECG data collected from Bitalino is processed by a Matlab process. The raw inputs are converted to relevant voltage data and the heart rate is properly extracted.

3.3 Networking

3.3.1 Bluetooth

Bluetooth is the wireless technology supported by the Bitalino board. It is a short range wireless technology with small power consumption, making it ideal for various small, battery-driven devices, ranging from phones to laptops. Its main purpose during development stages was to replace all sorts of cables used to connect different devices. This idea evolved throughout time and Bluetooth is now not only a point-to-point solution, but a network solution as well.

Low cost and low power Bluetooth modules meet the requirements of wireless sensor networks with its Ad-hoc connection establishment capability, throughput, usage of frequency-hopping scheme with time division duple to minimize the impact o interference in the ISM band, different power saving modes paired with its availability and standardized specification, making them a solid solution as a wireless technology, as noted by Krco, S. [14]

Bluetooth has passed through multiple iterations, each one aimed to improve upon power consumption, data rate, security, etc. Each iteration brought the following features [15]:

3.3.1.1 Version 1.x

With theoretical max data rate of 1Mbps, the devices that use the first iterations of Bluetooth are considered extinct nowadays.

3.3.1.2 Version 2.x

This popular variant introduced the optional Enhanced Data Rate (EDR) capability, allowing devices to have a theoretical maximum data rate of 3Mbps. Version 2.1 facilitated the pairing between devices from different manufacturers, increased the reliability of the pairing process and featured Secure Simple Pairing (SSP) that not only improved the pairing experience, but also increased the use and strength of security. SSP is essential to protect data from man in the middle (MITM) attacks.

3.3.1.3 Version 3.x

This version introduced the support for an alternate lower layer, that meant all applications available with Bluetooth radio can also be run over a different radio (802.11). It also came along an optional feature called High Speed (HS) that intends to improve speed, with theoretical HS max data rate of 24Mbps when using 802.11 WiFi physical layer.

3.3.1.4 Version 4.x

This iteration introduced support for collecting data from very low-rate devices. The main objective of this feature, named Low Energy (LE), is to aggregate data from various sensors, like heart rate monitors, thermometers, etc.

Note that most of the features introduced with newer revisions are optional, and one should pay attention to the added feature, not only the Bluetooth version. For example, Bluetooth 3.0 does not necessarily have the optional HS capability, it should be advertised as 3.0+HS. Also, a Bluetooth 3.0 or 4.0 device may not even support EDR, or SSP (Secure Simple Pairing introduced in v.2.1), as they are optional features.

3.3.2 HTTP

HTTP connections are established between the Smart Sensor Node and the Sensor Network. They represent the main outgoing communication of the Smart Sensor Node.

The HTTP, or Hypertext Transfer Protocol, is an application-level protocol aimed for distributed information systems. It is a stateless protocol that can be used for many tasks besides its use for hypertext, like name servers and distributed object management , through the extension of its request methods, error codes and headers [16].

3.4 SelSus

SelSus is a project that aims to create a new standard for component monitoring and self-healing in order to maximize their performance over longer life times, targeting timely repair, renovation and upgrading. The project focuses in the European industry and its variety of manufacturing production systems.

The loss of production due to systems downtime, the costs of maintenance due to wear, labour and spare parts directly affect competitiveness of a company. Improving availability and prevent failures is one of the main economic factors for a premium assembly system.

The next generation machines, fixtures and tools come along with extended sensory capabilities and smart materials that enable self-diagnosis and allows them to become self-aware and to self-repair. These self-away devices develop a synergetic relationship with their human operators and maintenance personnel through pro-active communication.

In order to achieve the aforementioned objectives, SelSus relies on its network, the SelSus Network, to which every machine connects to. This network maps all its nodes and stores the health condition of each. The heterogeneity of the machines in an industrial context, requires the network to be able to adapt to each node.

3.5 ECG signals

ECG is a method to measure varying electrical potentials of the heart. Developed by Willem Einthoven in the early 1900s, the ECG may roughly be separated into phases of depolarization ad repolarization of the muscle fibers making up the heart.

The P-wave (atrial depolarization) and QRS-wave (ventricles depolarization) correspond to the depolarization phases. The T-wave and U-wave represent the repolarization (ventricular repolarization).

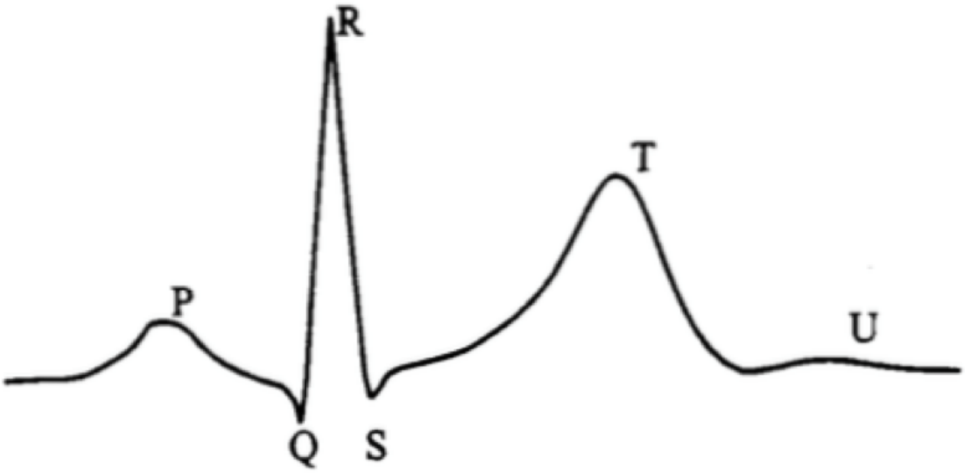


Figure 3.3: ECG wave

3.5.1 Heart rate variability

The autonomic nervous system (ANS) is the most prominent of the physiologic factors that determine cardiac functions such as heart rate.

The heart rate time series is defined by beat to beat variability over a wide range. Since the heart is under tonic inhibitory control by parasympathetic influences, the sympathetic influence on the heart is too slow to produce beat to beat changes. The basic data for the calculations of HRV are the time intervals between heart beats. This inter-beat interval time series is used to calculate the variability in the timing of the heart beat. Since the heart is dually innervated by the autonomic nervous system, relative increases in sympathetic activity are associated with heart rate increases and relative increases in parasympathetic activity are associated with heart rate decreases.

One of the main functions of the cardiovascular system is to maintain optimal arterial blood pressure and to provide the correct blood flow to the vital organs. The blood pressure and the distribution of the blood flow throughout the body are adjusted as a response to environmental demands, according to Thayer, Julian F. et al. [17].

The heart rate variability proves to be a reliable source of information regarding working men, as it directly translates fatigue and stress, therefore this was the main metric adopted.

3.6 Possible approaches

In order to have a network react to its human counterparts with the existing equipment, different approaches could be adopted.

One possible approach would be processing all related biometric data locally at sensor node level. This would leverage the work in the network, that would only consist of having the installed nodes adapt to the data emitted by the human sensor node. Given that we're using a Raspberry

Pi Model B, this amount of processing proved to be difficult due to its limitations and would not scale well in the scenario where multiple sensor data is being captured.

This leads to the second and adopted approach that consists of having the human sensor node capture and store the latest biometric samples and upload them to the network, making it its only purpose. This adds the network the responsibility of guiding the raw biometric data to its processing unit, also deployed remotely. This approach allows for more scalable solutions where multiple processing units can be deployed in the network, allowing it to scale in regards to number of total users and different kinds of sensors that require different types of processing.

ECG signals were the main and only focus of biometric data gathering due to being linked more directly to fatigue levels out of all biometric sensors existing in the Bitalino board.

3.7 Conclusion

Despite being low cost, Bitalino carries the necessary features for biometric data gathering that, when tagged alongside a Raspberry Pi, forms the desired sensor node that can be deployed to a person.

The data processing is in charge of a specialized Matlab process that takes advantages of the tool's mathematical capabilities to process ECG data and calculate the heart rate from it, and given that the heart rate variability represents a valid metric to correlate with stress and fatigue, it becomes possible to gain perception of the user's physical condition from it.

Chapter 4

Implementation

This chapter will go into further implementation details about the user data gathering and processing, and the data journey starting from Bitalino's modules to the Sensor network.

The chosen architecture consists of multiple Raspberry Pi's, each one paired to one or more Bitalino board's, representing the local hardware resources. This setup requires a valid network connection to ensure the connective to the sensor cloud is possible.

4.1 Data gathering

The first step of the implementation consists in gathering the necessary user data. This chapter focus essentially on gathering ECG data. For ECG data gathering, the three ECG electrodes (positive, negative and ground) are connected to the users body. There are many possible configurations:

4.1.1 Chest electrodes

1. Positive electrode on top right part of chest

2. Negative electrode on bottom left part of chest

3. Ground electrode on top left part of chest

4.1.2 Hand electrodes

1. Positive electrode on right hand

2. Negative and Ground electrode on left hand

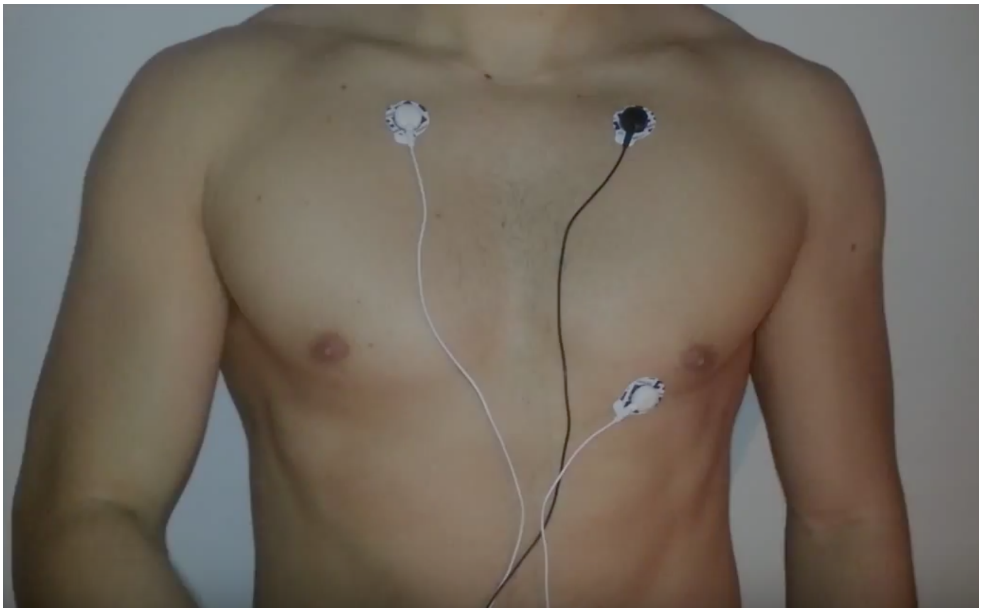


Figure 4.1: ECG electrodes connected to chest

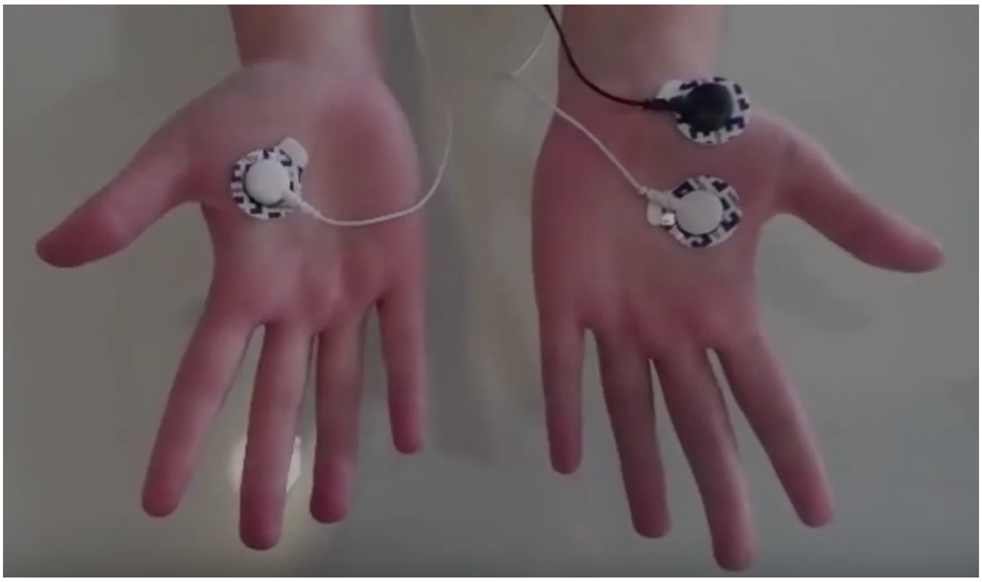


Figure 4.2: ECG electrodes connected to hand

4.1.3 Hand & Leg electrodes

1. Positive electrode on right wrist
2. Negative electrode on left wrist
3. Ground electrode on right ankle

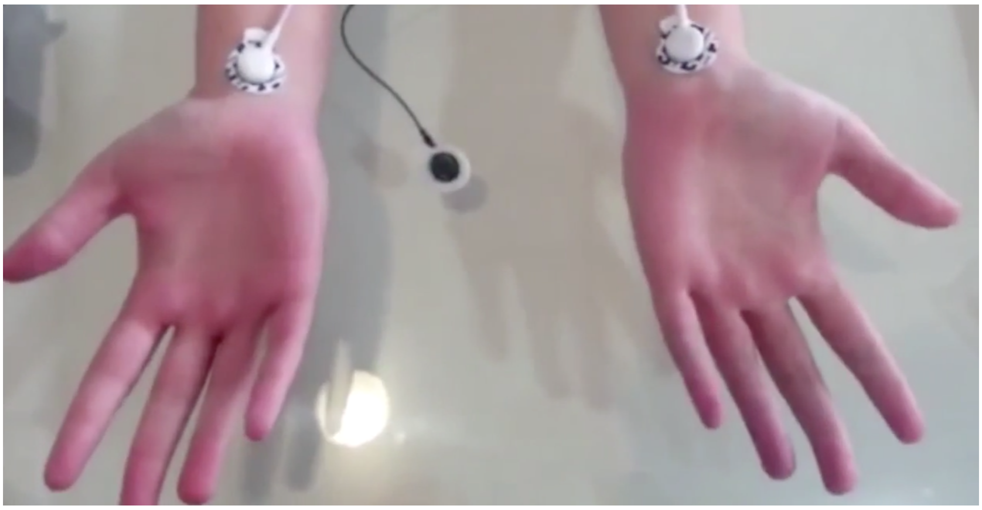


Figure 4.3: ECG positive and negative electrodes connected to hand

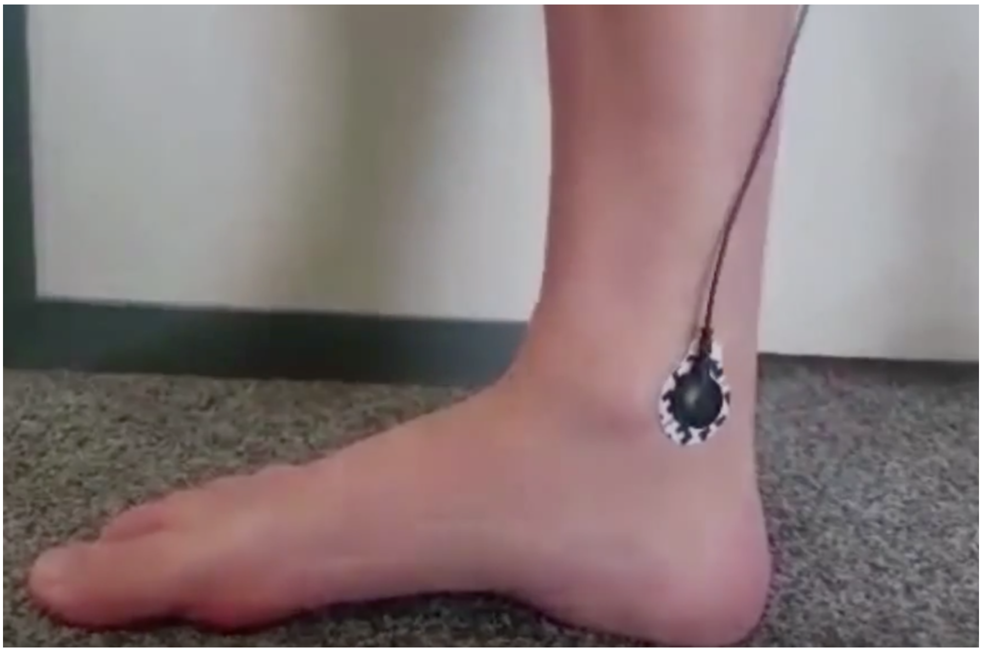


Figure 4.4: ECG ground electrodes connected to ankle

4.1.4 Raspberry Pi Data Gatherer

A process running in the Raspberry Pi is responsible for interfacing with the Bitalino. This process is responsible for registering in the Sensor Cloud as a Sensor Node, as well as establishing the connection to the Bitalino board.

4.1.4.1 Communication with the Sensor Network

This communication is made via HTTP. The first step consists in sending an XML file called *Selcomp Self Descriptor* that gathers all the information about that node. Information like the node's IP, mac address and ID are present in this file.

The ECG module is registered as well in the *Selcomp Self Descriptor*, as a device of type observation. The type and range of values are also set here so that the Sensor Network is able to properly receive and process them later.

The ECG data is sent to the Sensor Network also in XML format, this time in a payload file. This payload file may contain a variable number of datasets. Each dataset is made of its measurement value (defined in the *Selcomp Self Descriptor*) and its corresponding timestamp.

4.1.4.2 Communication with the Bitalino

This process is responsible for all the communication with the Bitalino. This communication happens via Bluetooth after the registration in the Sensor Network.

After opening the connection to the Bitalino board, by opening a socket based on the board's MAC address, the sampling rate and acquisition channels are parameters sent to the Bitalino board.

The sampling rate varies between 1, 10, 100 or 1000 samples per second. The latter was used so that a more accurate line of the pulse could be obtained out of it.

The acquisition channels represent which Bitalino modules are going to be activated. For this scenario, only the ECG module is considered. The ECG data is represented with 10 bits of precision.

Both the sampling rate and the acquisition channels are gathered in a single byte frame each and sent to the board to start ACQUISTION mode.

After this step, the board begins the transmission of data via Bluetooth. A configurable number of samples per payload is set. After each frame received, the ECG data is extracted from it and stored alongside the current timestamp. After retrieving the necessary amount of samples, a payload is generated and sent to the Sensor Network.

4.1.5 Matlab Data Processor

The Sensor Network is responsible for storing and sending the received payload data to the existing Matlab process. After the retrieval of the raw ECG data, the Matlab process will then apply a formula to extract the heart rate.

4.1.5.1 Savitzky-Golay smoothing filter

It starts by applying a Savitzky-Golay smoothing filter to smooth out noisy signal. This filter proved to be efficient in smoothing the received data, while other FIR filters tended to filter out a significant portion of the signals high frequency content along with the noise. This digital filter achieves smoothing trough a process of convolution, by joining successive sub-sets of adjacent data points with a low-degree polynomial by the method of a linear least squares, and applying an analytical solution to all sub-sets.

4.1.5.2 Peaks detection

The peaks are detected after applying the Savitzky-Golay smoothing filter. The desired wave to extract is the QRS-wave in which the higher value of each beat is found. The R peak is detected when values with greater value than 0.75 are extracted, with the minimal peak distance of 40 samples. These values proved effective in detecting the QRS-wave while, at the same time dismissing the T-wave.

The intervals between the QRS-waves are then computed to extract the heart rate.

This value is then returned to the sensor network.

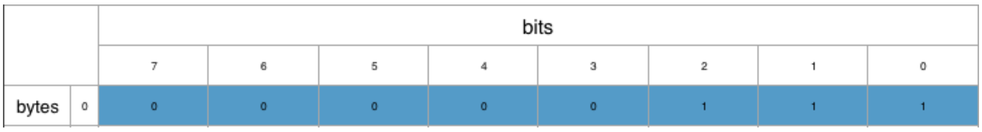
4.2 Bitalino protocol

The Bitalino board is able to respond the different commands and has two different operation modes: IDLE and ACQUISITION.

4.2.1 IDLE

When IDLE, Bitalino's default operating mode, it is possible to perform multiple operations.

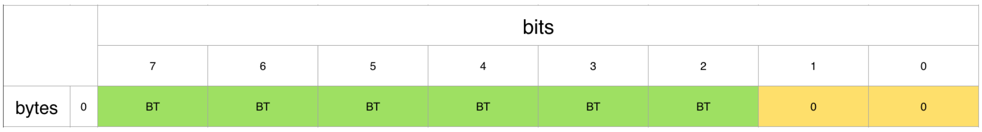
It is possible to obtain Bitalino's version with the command shown in figure 4.5.



		bits							
		7	6	5	4	3	2	1	0
bytes	0	0	0	0	0	0	1	1	1

Figure 4.5: Version command

It is possible to set the battery's threshold, where the minimum accepted value, 0, corresponds to 3.4 Volts, and the maximum value, 64, corresponds to 3.8 Volts. This value is bit-wised shifted to the right twice, leaving the first two bits as 0.

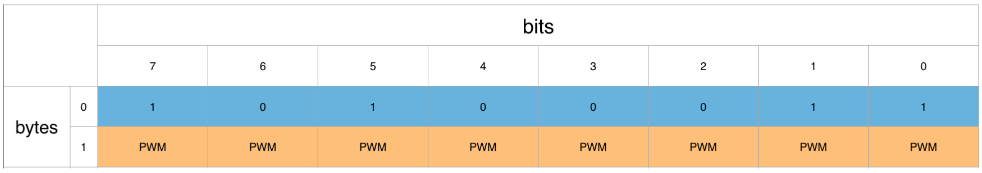


		bits							
		7	6	5	4	3	2	1	0
bytes	0	BT	BT	BT	BT	BT	BT	0	0

Figure 4.6: Battery command

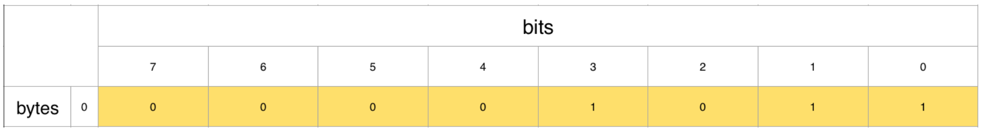
The PWM, or Pulse Width Modulation, can also be set with the commands shown in figure 4.7.

The state command allows the retrieval of all analog and digital channels, the battery threshold as well as the battery level. The response from Bitalino to this command can be seen in figure 4.9.



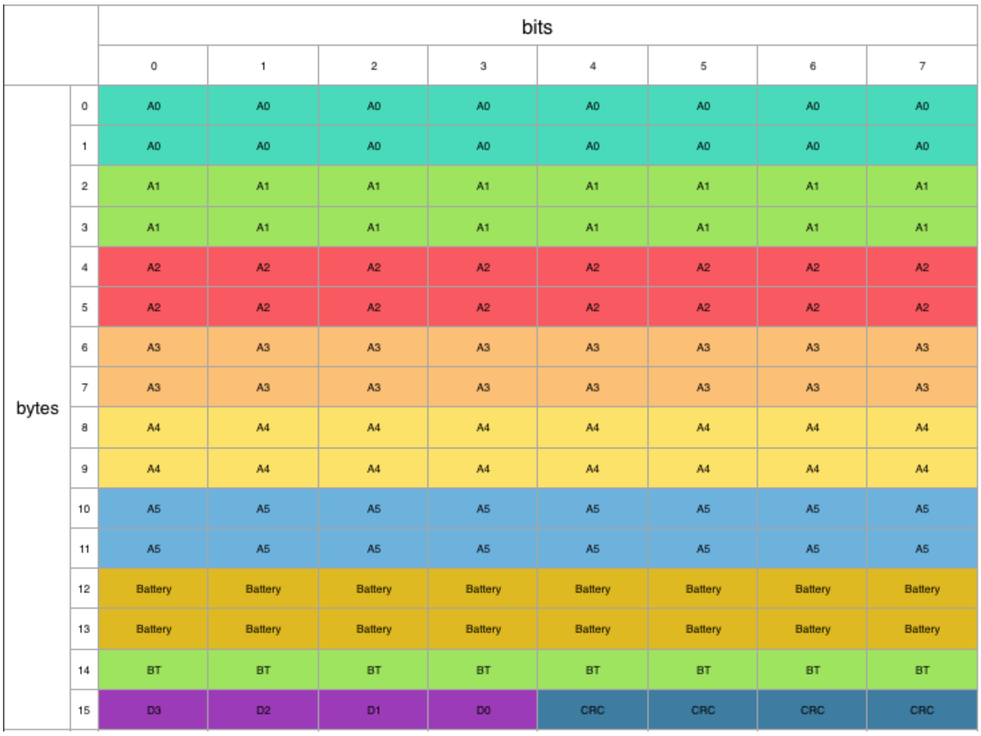
		bits							
		7	6	5	4	3	2	1	0
bytes	0	1	0	1	0	0	0	1	1
	1	PWM	PWM	PWM	PWM	PWM	PWM	PWM	PWM

Figure 4.7: PWM command



		bits							
		7	6	5	4	3	2	1	0
bytes	0	0	0	0	0	1	0	1	1

Figure 4.8: State command



		bits							
		0	1	2	3	4	5	6	7
bytes	0	A0	A0	A0	A0	A0	A0	A0	A0
	1	A0	A0	A0	A0	A0	A0	A0	A0
	2	A1	A1	A1	A1	A1	A1	A1	A1
	3	A1	A1	A1	A1	A1	A1	A1	A1
	4	A2	A2	A2	A2	A2	A2	A2	A2
	5	A2	A2	A2	A2	A2	A2	A2	A2
	6	A3	A3	A3	A3	A3	A3	A3	A3
	7	A3	A3	A3	A3	A3	A3	A3	A3
	8	A4	A4	A4	A4	A4	A4	A4	A4
	9	A4	A4	A4	A4	A4	A4	A4	A4
	10	A5	A5	A5	A5	A5	A5	A5	A5
	11	A5	A5	A5	A5	A5	A5	A5	A5
	12	Battery	Battery	Battery	Battery	Battery	Battery	Battery	Battery
	13	Battery	Battery	Battery	Battery	Battery	Battery	Battery	Battery
	14	BT	BT	BT	BT	BT	BT	BT	BT
	15	D3	D2	D1	D0	CRC	CRC	CRC	CRC

Figure 4.9: State response

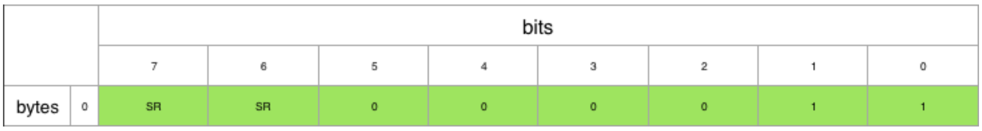
4.2.2 ACQUISITION

ACQUISTION state is triggered when two specific bytes are sent to the Bitalino board.

The first byte sets the acquisition rate. The board supports four sampling rates:

- 1 Hz (FS = 01)
- 10 Hz (FS = 10)
- 100 Hz (FS = 11)

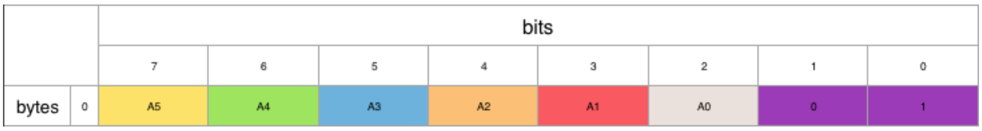
- 1000 Hz (FS = 11)



		bits							
		7	6	5	4	3	2	1	0
bytes	0	SR	SR	0	0	0	0	1	1

Figure 4.10: Sampling rate

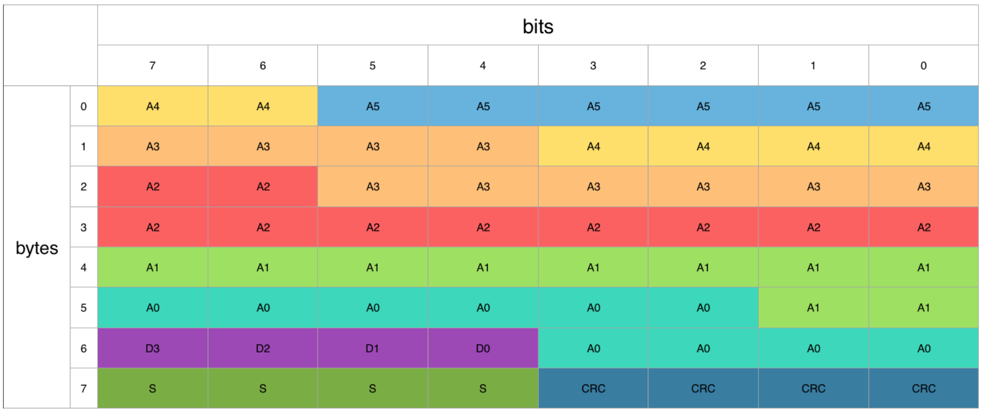
The second byte defines which of the six analog channels are to operate. After this command, the board will start to emit data from the chosen analog channels.



		bits							
		7	6	5	4	3	2	1	0
bytes	0	A5	A4	A3	A2	A1	A0	0	1

Figure 4.11: Selecting which channels to activate

After ACQUISITION mode is set, Bitalino emits frames with data from the selected channels. Figure 4.12 displays the scenario where all analog channels are retrieved. In this scenario, a full frame of data consists of 8 bytes. The last byte contains the sequence number that can be useful that guarantee the order of the frames (the sequence number ascends from 0 to 15 and then is reset to 0) and the CRC value that is calculated with the values from the previous bytes to guarantee the validity of the data frame.



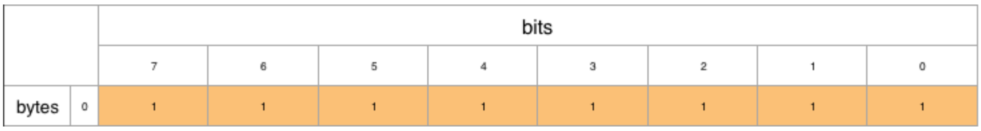
		bits							
		7	6	5	4	3	2	1	0
bytes	0	A4	A4	A5	A5	A5	A5	A5	A5
	1	A3	A3	A3	A3	A4	A4	A4	A4
	2	A2	A2	A3	A3	A3	A3	A3	A3
	3	A2	A2	A2	A2	A2	A2	A2	A2
	4	A1	A1	A1	A1	A1	A1	A1	A1
	5	A0	A0	A0	A0	A0	A0	A1	A1
	6	D3	D2	D1	D0	A0	A0	A0	A0
	7	S	S	S	S	CRC	CRC	CRC	CRC

Figure 4.12: Bitalino ACQUISITION frames

A STOP command can be sent to change the operating mode to IDLE, as seen in figure 4.13.

4.3 Conclusion

This chapter explains how the sensor node is setup physically with the placement of its electrodes, and how its components communicate with each other.



		bits							
		7	6	5	4	3	2	1	0
bytes	0	1	1	1	1	1	1	1	1

Figure 4.13: Stop command

The communication between the sensor node and the sensor network consists of the generation of XML files that include information about the node and the data gathered.

The Matlab process extracts the heart rate after reducing the noise and finding the peaks if received signals data.

Bitalino's commands are thoroughly explained and the steps required to start data acquisition are also shown. All of this exchange happens on top of a Bluetooth connection between the Raspberry Pi and the Bitalino board.

Chapter 5

Pratical experiment

To put the previously discussed setup in practice, a pratical experiment was arranged to simulate a real work scenario of human-machine interaction. In this scenario, a robot arm performs a linear task and its operating speed adapts to the user's heart rate.

The simple task performed by the robotic arm consists in picking a simple object and placing it on the other side of a table. The speed the arm performs this task depends on the user's heart rate, simulating an assembly line where machines adapt to the user's rhythm.

The setup of the experiment consists of a Bitalino with an ECG module, connected to a user, connected via Bluetooth to a Raspberry Pi B. The Raspberry Pi resided in the same network of the Selsus Network, allowing it to upload Selsus Sensor Descriptor, detailing the information of the data to gather as well as device properties like IP and ID, and the payload files with raw ECG data.

5.1 Registering a Sensor Node

The first step consists in registering a Bitalino + Raspberry Pi setup as a sensor node.

Below is a *Selcomp Self Descriptor* file generated by the Raspberry Pi. This file allows the Selsus Network to register the new device and becomes immediately available to store data sent by it.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<selcompselfdescription>
  <general>
    <data id="SELCOMP_NAME">HumanSensor</data>
    <data id="DEVICE_TYPE">SENSOR_NODE</data>
    <data id="SELCOMP_IP">192.168.1.76</data>
    <data id="SELCOMP_MAC">54-62-DF-A8-92-18</data>
    <data id="SELCOMP_PORT">8080</data>
    <data id="SELCOMP_ID">BITALINO_TEST_1</data>
  </general>
  <deviceset>
    <device id="ecg_data" type="OBSERVATION">
```

```
      <variables>
        <variable name="ECG">
          <elements>
            <element id="MeasurementType">ecg_data</element>
            <element id="DataType">java.lang.Double</element>
            <element id="max">1024</element>
            <element id="min">-1024</element>
            <element id="SourceState">RUNNING</element>
          </elements>
        </variable>
      </variables>
    </device>
    <device id="eda_data" type="OBSERVATION">
      <variables>
        <variable name="EDA">
          <elements>
            <element id="MeasurementType">eda_data</element>
            <element id="DataType">java.lang.Double</element>
            <element id="max">1024</element>
            <element id="min">-1024</element>
            <element id="SourceState">RUNNING</element>
          </elements>
        </variable>
      </variables>
    </device>
    <device id="HumanSensor" type="EQUIPMENT">
      <variables>
        <variable name="EquipmentDescription">
          <elements>
            <element id="Manufacturer">Bitalino</element>
            <element id="Model">Bitalino</element>
            <element id="PhysicalLinkType">RF</element>
            <element id="ProtocolName">Bluetooth</element>
            <element id="ProtocolNorm">802.11</element>
            <element id="ShortDescription">Monitor Operators Condition</element>
            <element id="YearBuilt">2015-04-14T23:00:00.000Z</element>
          </elements>
        </variable>
      </variables>
      <subscriptions>
        <id>ecg_data</id>
        <id>eda_data</id>
      </subscriptions>
    </device>
  </deviceset>
  <serviceinstanceset>
    <serviceinstance id="i1" type="ObservationService" dId="
        6710811111710049464846486810597103110111115116105998310111411810599101">
      <subscriptions><id>ecg_data</id></subscriptions>
    </serviceinstance>
    <serviceinstance id="i2" type="ObservationService" dId="
        6710811111710049464846486810597103110111115116105998310111411810599101">
      <subscriptions><id>eda_data</id></subscriptions>
    </serviceinstance>
  </serviceinstanceset>
  <servicedescriptionset>
    <servicedescriptionset dId="
        6710811111710049464846486810597103110111115116105998310111411810599101"
        name="Cloud" version="ObservationService" type="1.0.0">
      <recipeadjustments>
        <recipeadjustment name="AcquisitionRate">
          <variables>
            <variable name="rate">
              <elements>
                <element id="min">0</element>
                <element id="max">1000</element>
                <element id="unit">s</element>
              </elements>
            </variable>
          </variables>
        </recipeadjustment>
        <recipeadjustment name="AddLink">
          <variables>
            <variable name="ServiceID">
              <elements>
                <element id="Type">String</element>
              </elements>
            </variable>
          </variables>
        </recipeadjustment>
        <recipeadjustment name="RemoveLink">
          <variables>
            <variable name="ServiceID">
              <elements>
                <element id="Type">String</element>
                <element id="Type">String</element>
              </elements>
            </variable>
          </variables>
        </recipeadjustment>
        <recipeadjustment name="CreateInstance">
          <variables>
            <variable name="ServiceIDs">
              <elements/>
            </variable>
          </variables>
        </recipeadjustment>
        <recipeadjustment name="Interface">
```

```
          <variables>
            <variable name="Input">
              <elements>
                <element id="DataType">java.lang.Double</element>
              </elements>
            </variable>
            <variable name="Output">
              <elements>
                <element id="DataType">java.lang.Double</element>
              </elements>
            </variable>
          </variables>
        </recipeadjustment>
      </recipeadjustments>
    </servicedescriptionset>
  </servicedescriptionset>
</selcompselfdescription>
```

5.2 Data retrieval

ECG data is retrieved with one of the possible electrode configuration using the Bitalino's ECG module. The data is collected by the Raspberry Pi and uploaded to the Sensor Network. After a configured number of samples is captured, 5000 by default.

For each batch collected, a payload file is generated and uploaded to the Sensor Network.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<measurementdata xmlns="http://selsus.eu/sensorData">
    <service>
        <servicetype>ObservationService</servicetype>
        <deviceid>ecg_data</deviceid>
        <selcompid>BITALINO_TEST</selcompid>
        <cycle number="1">

            <dataset>
                <data id="measurement">0.0439453125</data>
                <data id="timestamp">2016-06-26T15:45:10.985356Z</data>
            </dataset>

            <dataset>
                <data id="measurement">0.0439453125</data>
                <data id="timestamp">2016-06-26T15:45:10.986446Z</data>
            </dataset>

            <dataset>
                <data id="measurement">0.0439453125</data>
                <data id="timestamp">2016-06-26T15:45:10.987827Z</data>
            </dataset>

            <dataset>
                <data id="measurement">0.03515625</data>
                <data id="timestamp">2016-06-26T15:45:10.989210Z</data>
            </dataset>

            <dataset>
                <data id="measurement">0.0205078125</data>
                <data id="timestamp">2016-06-26T15:45:10.990234Z</data>
            </dataset>

            <dataset>
                <data id="measurement">0.0205078125</data>
                <data id="timestamp">2016-06-26T15:45:10.991245Z</data>
            </dataset>

            <dataset>
                <data id="measurement">0.0029296875</data>
                <data id="timestamp">2016-06-26T15:45:10.992237Z</data>
            </dataset>

            <dataset>
                <data id="measurement">0.0</data>
                <data id="timestamp">2016-06-26T15:45:10.993263Z</data>
            </dataset>

            <dataset>
                <data id="measurement">-0.0029296875</data>
                <data id="timestamp">2016-06-26T15:45:10.994269Z</data>
            </dataset>

            <dataset>
                <data id="measurement">-0.005859375</data>
                <data id="timestamp">2016-06-26T15:45:10.995296Z</data>
            </dataset>
        </cycle>
    </service>
</measurementdata>
```

5.3 Data processing

A dedicated Matlab process was created to compute the raw ECG data, extract the user's current heart rate and calculate the robotic arm's operating speed according to it. Figures 5.1 and 5.2 display a batch a 5000 samples in order. Given the low amount of noise in this batch, the Matlab process was able to calculate the user's heart rate.

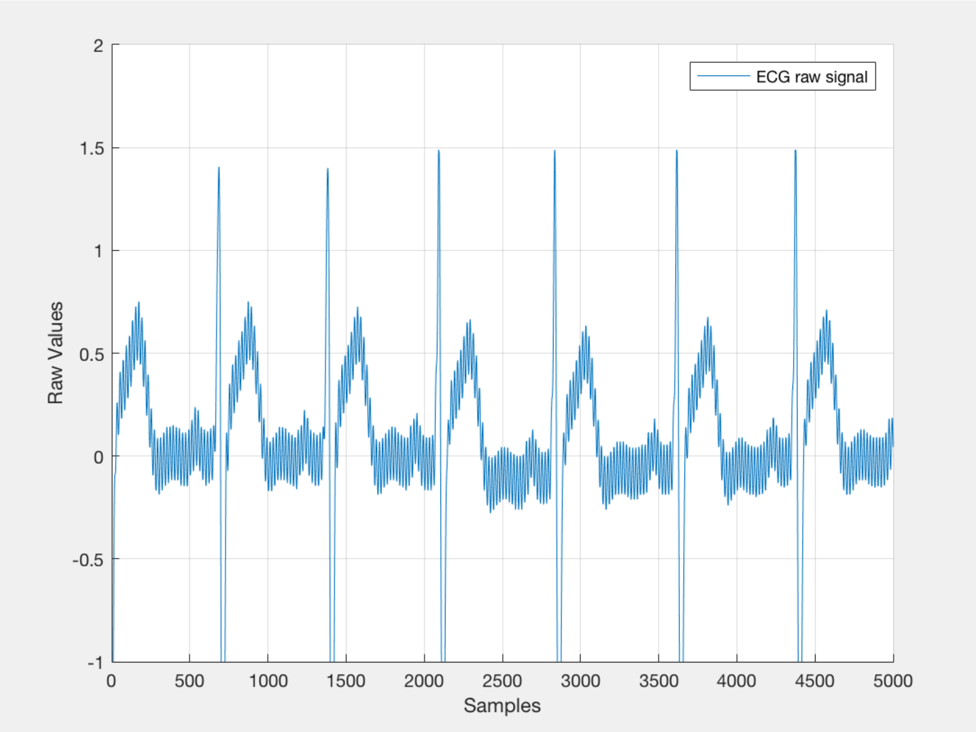


Figure 5.1: Plot of ECG raw samples

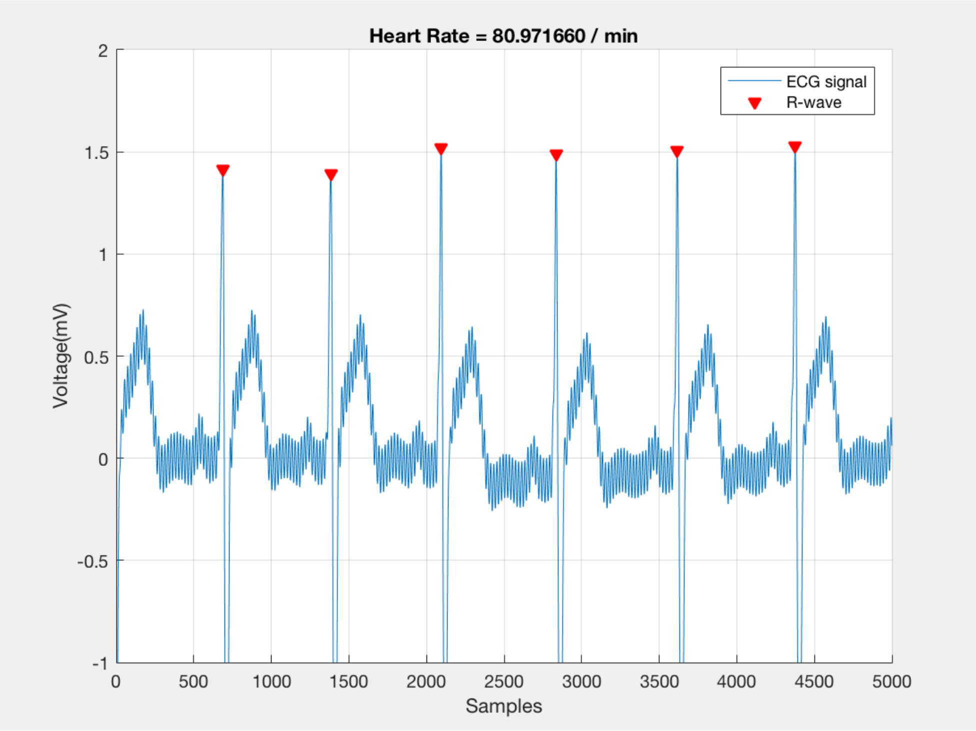


Figure 5.2: Plot of ECG samples after processing

One issue with the electrodes being used was discovered when measuring the heart rate during intense physical work. While trying to mock a scenario of heavy physical work, the chest electrodes configuration was used while the user performed push-ups throughout the entire sample collecting stage. As figures 5.3 and 5.4, the data gathered during intense workout cannot be used to extract useful information.

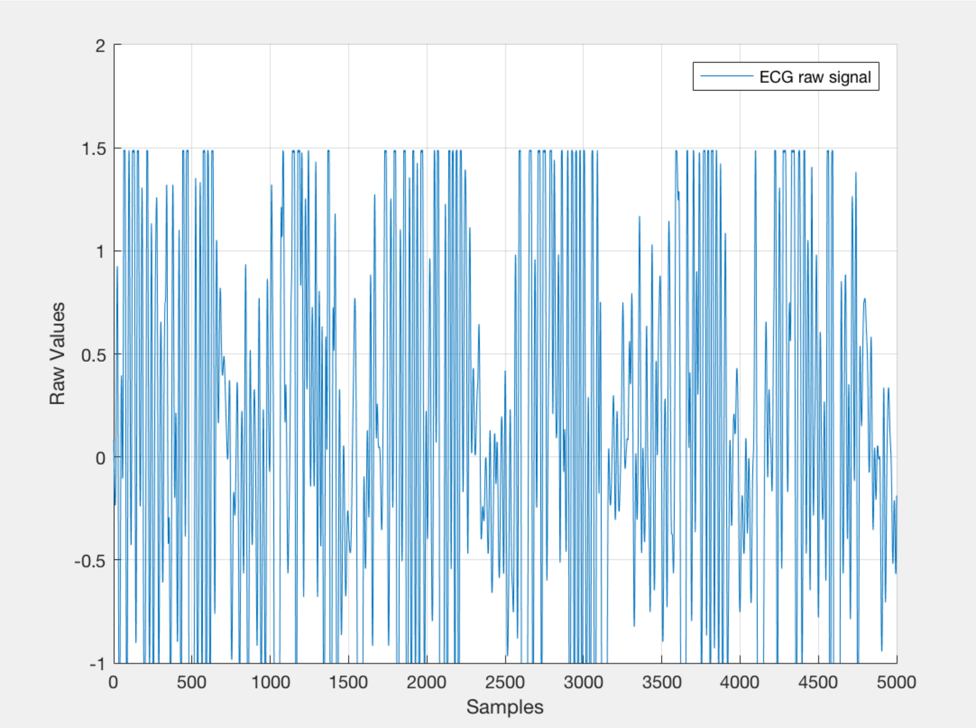


Figure 5.3: Plot of noisy ECG samples

5.4 Adjusting Robotic arm speed

The robotic arm is a device that is mapped as a *selcomp machine* in the sensor network. Its configuration informs the network that is has one configurable parameter: time to perform each movement. The movements themselves are also configurable by the network, that defines and sends the list of movements to the arm.

The robotic arm was the chosen device to reflect in real time the updates sent by the human sensor node, by adjusting its speed accordingly.

The speed of the robotic arm is set by the Sensor Network by sending an integer value to it. The default value of the arm is 500. This represents the time in milliseconds the arm takes to perform each move. That value is calculated with the function below that takes one argument: the user's heart rate.

The following formula is used to calculate the time it takes for the arm to perform each movement:

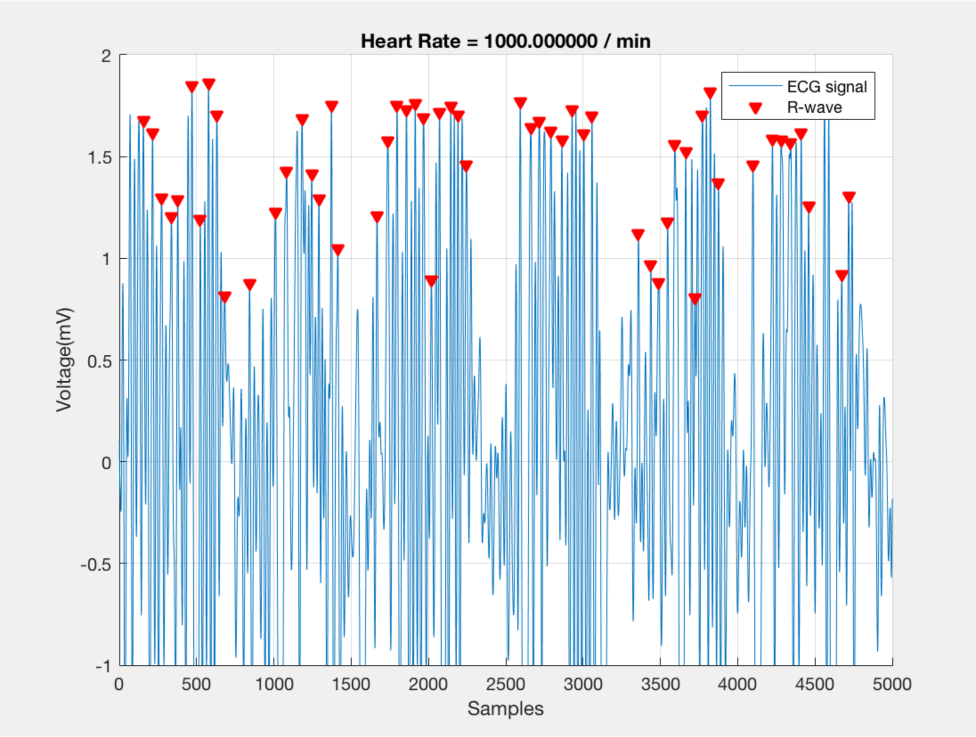


Figure 5.4: Plot of noisy ECG samples after processing

$hrmax = 160$

$hrmin = 60$

$tmax = 2000$

$tmin = 500$

$ft(hr) = \frac{(hr - hrmin) * (tmax - tmin)}{(hrmax - hrmin)} + timemin$

As the heart rate increases, so does the time the arm takes to perform each task.

Although simple, this algorithm could be at any time replaced for a more complex solution, although such implementation would be out of the scope of this dissertation.

Figures 5.6, 5.7 and 5.8 display speed of each motor of the robotic arm doing its movement operation during a given interval of time.

Figures 5.9 displays a decrease in velocity after receiving lower values for the time it takes to perform each movement. That value is a result of the above formula applied to the registered and more recent heart rate.

5.5 Conclusion

The integration of a sensor node was explained in this chapter, as well as the flow of data in the sensor network. The network relies on two kinds of files sent by each sensor node, a *Selcomp Self Descriptor* file that describes the whole node and lists the type of data it emits, and payload files with the measurements captured during acquisition.

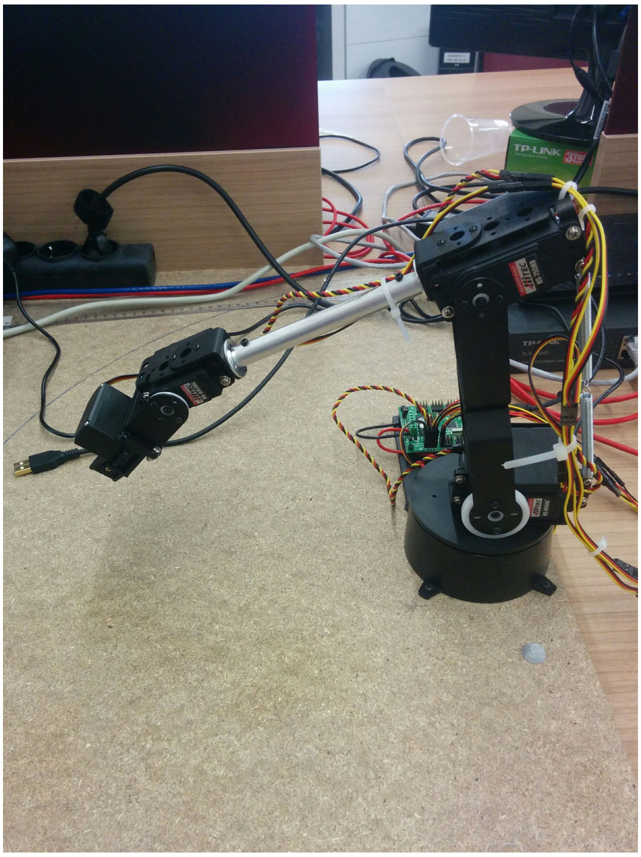


Figure 5.5: Robotic arm

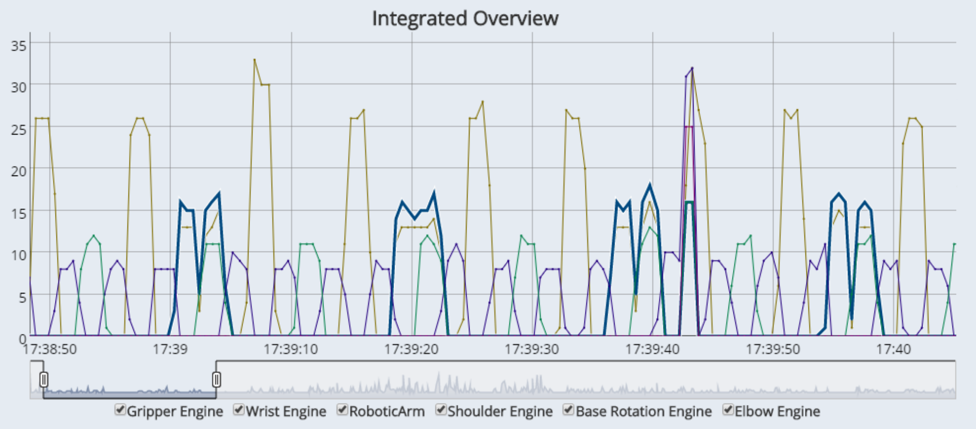


Figure 5.6: Robotic arm operating with 2000 ms interval between movements

Some of the limitations of the chosen hardware becomes visible with the illustration of noisy signal data captured during experiments.

The processed signal is converted to an heart rate value that in turn is transformed to the time of operation per movement of the robotic arm. The network has the responsibility of sending this value to the arm to make it adapt to the user's condition.

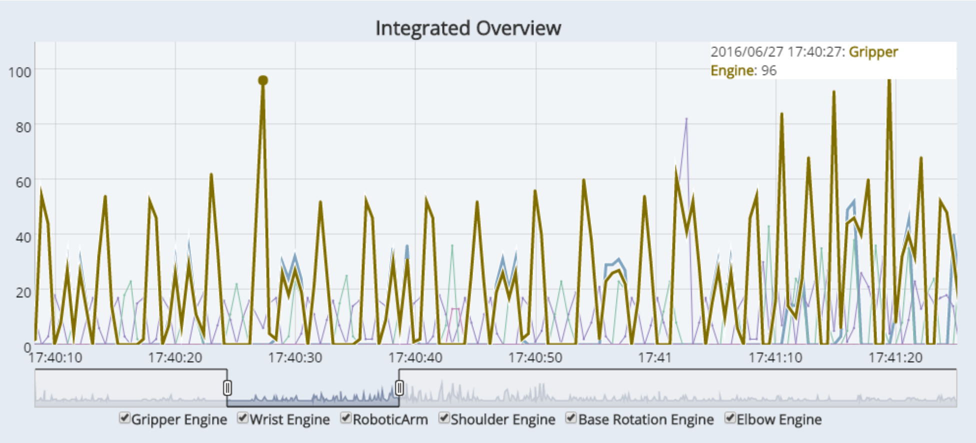


Figure 5.7: Robotic arm operating with 1000 ms interval between movements

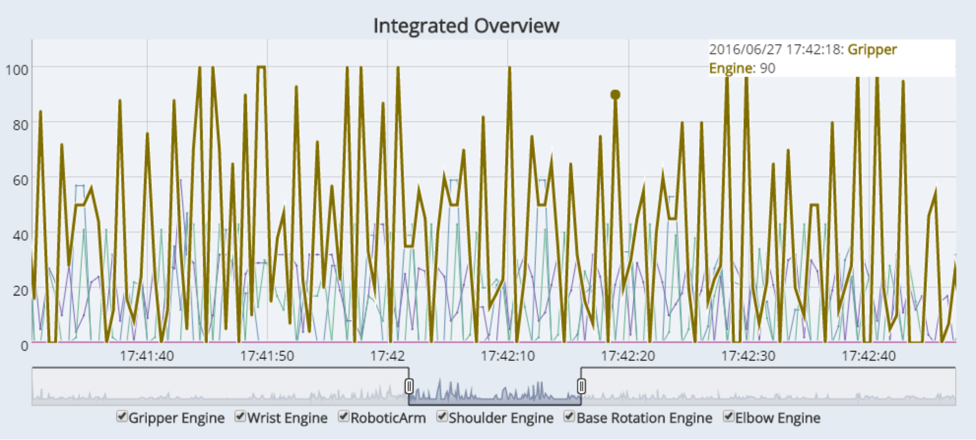


Figure 5.8: Robotic arm operating with 500 ms interval between movements

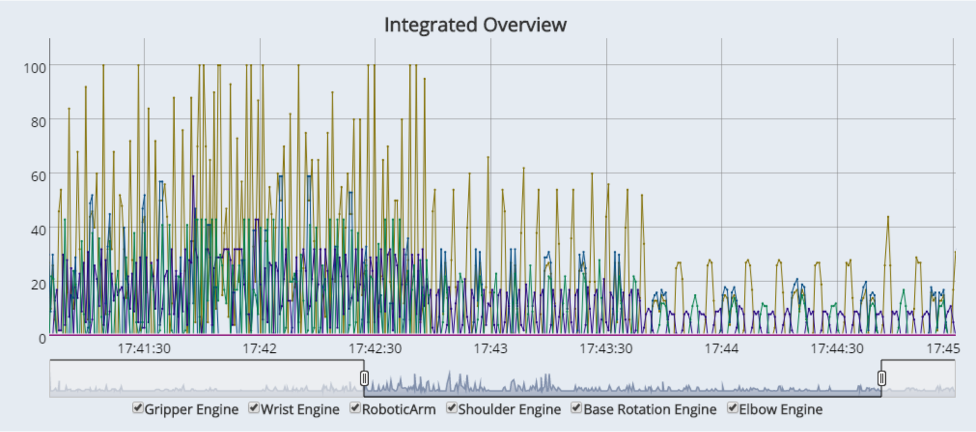


Figure 5.9: Robotic arm's speed adjusting after updates from network

Chapter 6

Conclusion

This work intended to link the human element to an already existing sensor network, previously meant exclusively of machines.

The Bitalino's biometric data gathering capabilities tagged alongside the Pi's networking and processing capabilities formed a valid foundation to approach a solution for the Smart Sensor Node.

6.1 Main difficulties

Given the heterogeneous nature of the project, difficulties were found on different layers of the project work.

6.1.1 Gathering Biometric Data

Learning how to interface with the Bitalino board as the first main challenge to overcome. The low-level nature of it raised many questions like "What's the right language to use to interface with the board?", "Is Bluetooth the only existing channel to interface with it?".

One of the first developments was to re-write existing API's for the Bitalino in NodeJS. This would not only be useful to gain full insight on the board itself, but also prepare ground work for later integration with the Sensor Cloud. However, after having most of the features implemented in NodeJS, the event driven nature of language and the limited processing power of the Raspberry Pi Model B+ proved to be ineffective to process the data emitted by the Bitalino board at higher frequencies. Using an equivalent approach but written in Python allowed the Raspberry Pi to interface with the Bitalino with the desired performance levels, even at the highest frequency.

6.1.2 Noisy ECG data

The Bitalino's electrodes were able to retrieve reliable ECG signal data. However, when connected to parts of the body where muscular activity occurred, like the chest for example, the retrieved data would become noisy and unreliable, therefore unusable. The best results were obtained when the user himself was idle. This raised problems during experiments where it became hard to simulate a real work scenario because the gathered samples were filled with noisy data and were impossible to process adequately.

6.1.3 Interfacing with the Sensor Cloud

The communication made with the Sensor Cloud was perhaps the easiest challenge to overcome but it's still worth of mention. The current changing nature of the Sensor Cloud required multiple adaptions to the XML files generated by the Raspberry Pi. Despite such changes, the required data required to be uploaded to the network remained the same, making these changes easy to implement.

6.1.4 Processing ECG data

One of the main challenges of this project was to convert raw ECG data to valuable data like heart rate. Because of the multiple mathematical formulas that had to be applied, and after some research, Matlab was considered the platform in which this computation should've been made. From a network point of view, having such computations being made in the Raspberry Pi would've been simpler because no extra connections would be needed. However, due to the high computational demand required by Matlab, it wouldn't be possible to implement the solution in the Raspberry Pi. Instead, the Matlab process responsible for retrieving raw ECG data and calculating the heart rate was spawned in a more powerful computer, accessible from within the network.

Matlab itself also presented some challenges with networking, messages received in TCP packets were not being correctly parsed, for instance. UDP requests were the solution found to receive such immense amount of data. Once retrieved, the raw data had to go trough different mathematical formulas that required some tweaking. Despite the challenges, Matlab proved to be a simple and effective way of processing raw ECG data.

6.2 Conclusions

The work developed allowed the main objectives to be achieved. The premise of this project was very ambitious - add an human element to an already existing network of machines - and this was achieved using low cost material and resources.

Despite the many roadblocks that emerged, the arranged setup was enough to orchestrate an automatized interaction between human and machine.

The pratical example arranged during the development of this project was the first hands-on experience with the human-machine interaction. It was possible to verify two things:

- The Selsus network is prepared for any heterogeneous sensor. The support for sensor descriptors allows them to be configured to cover all the data each sensor emits. After all, configuring the Bitalino and Raspberry Pi as one sensor node capable of emitting data from multiple sources simultaneously (ECG, EDA, EMG, and more) was not any different than configuring any other node.
- The chosen setup proved to be extensible, from the low-level hardware that can be upgraded for more reliable biometric data gatherers, to the network orchestration itself, where processing units, like the Matlab process that was spawned for this implementation, can be summoned to interact with one another. Although going through multiple stages, guiding the data from the Bitalino, computing it and making the robotic arm adapting to the user's heart beat, the flow can be easily tracked and extended. A more complex formula for processing the heart rate can be implemented, replacing the existing one. Extra sensors like EDA can also be mixed into the equation.

The technology exists and is accessible for anyone, human biometric tracking is possible and can be conciliated with machine tracking to gain better insight of the environment and automate tasks between them.

6.3 Future work

The developed work focused on the interaction of the Bitalino with the Sensor Cloud. However, this work can be extended and improved with the following:

- More reliable biometric data: Only ECG data was collected to calculate fatigue levels, but it is known that EDA data can also be used for such computations, making the conclusions obtained for it more reliable. This can still be achieved with the current setup, given that the Bitalino can also gather EDA data.
- Adapting to each user: Each person is unique and, as such, the data processing should take that into account. Heart rate is a metric that oscillates from person to person and takes into account many factors, like age, gender and condition. The implemented algorithms do not differentiate the users, making it more limited.

6.4 Personal notes

Personally, it was a pleasure to develop this dissertation and getting to work closely with the SelSus project. I feel that my academic formation became richer by working in a more low-level hardware setting and by working on a health related subject as well.

This was a great opportunity to bridge the current work that has been developed for SelSus with a more health related integration. This merge proved the flexible capabilities of the existing Sensor Networks and the added value of including human data to such networks.

Conclusion

After facing a good set of challenges and roadblocks, I felt true joy when the whole workflow worked out and I could see the network adapt to the captured heart rate.

I believe health related computing will gain even more traction over the next years, making it a desirable field of study. What this project achieved with such low cost devices proves that there are multiple applications for these devices and they’re accessible to everyone.

Bibliography

[1] Bin Yu ; Lisheng Xu ; Yongxu Li(2012) “Bluetooth Low Energy (BLE) based mobile electrocardiogram monitoring system”, In the proceedings of International Conference on Information and Automation, June 6-8, 2012, Shenyang, pp. 763-767

[2] Lee, Y.D. and Chung, W.Y. (2009) “Wireless Sensor Network Based Wearable Smart Shirt for Ubiquitous Health and Activity Monitoring”, Sensors and Actuators B: Chameical, Vol. 140, No. 2, July, pp. 390-395

[3] E. Kohlwey, A. Sussman, J. Trost, and A. Maurer, “Leveraging the Cloud for Big Data Biometrics: Meeting the performance requirements of the Next Generation Biometric Systems,” in Proceeding of the IEEE World Congress on Services, pp. 597-601, 2011

[4] NIST Special Publication 800-145 “The NIST Definition of Cloud Computing” Peter Mell Timothy Grance http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[5] Homepage of the BioID cloud-based biometric recognition solution, available from: http://www.bioid.com/, last visited: 07.01.2016

[6] Homepage of Ceelox ID Online, available from: http://www.ceelox.com/ceeloxidonline.html, last visited: 07.01.2016

[7] Parane, K.A., Patil, N.C. ; Poojara, S.R. ; Kamble, T.S(2014) “Cloud based Intelligent Healthcare Monitoring System”, In the proceedings of International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), February 7-8, Ghaziabad, Indian, pp. 697-701

[8] Neto, L. ; Reis, J. ; Guimarães, D. ; Gonçalves , G. (2015) "Sensor Cloud: SmartComponent Framework for Logical Reconfigurable Diagnostics in Intelligent Manufacturing Environments", Institute for Systems and Robotics, Faculty of Engineering of University of Porto

[9] Bourouis, A., Feham, M., and Bouchachia, A.(2011), “ Ubiquitous Mobile Health Monitoring System for Elderly (UMHMSE)”, International Journal of Computer Science and Information Technology, Vol.2, No. 3, June, pp. 74-82

[10] J. Healey, R. Picard, "SmartCar: Detecting Driver Stress", Massachusetts Institute of Technology

[11] J. Healey, R. Picard, "Detecting Stress During Real-World Driving Tasks Using Physiological Sensors", Cambridge Research Laboratory

[12] Gerd Kortuem, David Alford, Linden Ball, Jerry Busby, Nigel Davies, Christos Efstratiou, Joe Finney, Marian Iszatt White, Katharina Kinder, "Sensor Networks or Smart Artifacts? An Exploration of Organizational Issues of An Industrial Health and Safety Monitoring System", Lancaster University

[13] I.F. Akyildiz, W. Su*, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", Georgia Institute of Technology, 2001

[14] Krco, S., "Bluetooth Based Wireless Sensor Networks – Implementation Issues and Solutions", Applied Research Lab, Ericsson Ireland

[15] What is the difference between Bluetooth 2.0, 3.0 and 4.0? http://www.speedguide.net/faq/what-is-the-difference-between-bluetooth-20-30-460, accessed last time 23rd June 2016

[16] [RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, "Hypertext Transfer Protocol - HTTP/1.1", RFC2616, 1999

[17] Julian F. Thayera, Fredrik Åhsc, Mats Fredriksonc, John J. Sollers IIId, Tor D. Wager, "A meta-analysis of heart rate variability and neuroimaging studies: Implications for heart rate variability as a marker of stress and health"