A wearable miniaturized wireless multisensor of physiological signals for continuous monitoring

Auteur : Fyon, Arthur
Promoteur(s) : Redouté, Jean-Michel
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil biomédical, à finalité spécialisée
Année académique : 2020-2021
URI/URL : http://hdl.handle.net/2268.2/11515

A wearable miniaturized wireless multisensor of physiological signals for continuous monitoring

Master thesis submitted in partial fulfillment of the requirements for the degree

of

Master in Biomedical Engineering

by

FYON Arthur
s160913

Under the supervision of

Pr. Jean-Michel Redouté

MONTEFIORE INSTITUTE: ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

UNIVERSITY OF LIÈGE

ACADEMIC YEAR 2020-2021

DECLARATION

Project Title	A wearable miniaturized wireless multisensor of physiological signals for continuous monitoring
Author	FYON Arthur
Student ID	s160913
Supervisor	Pr. Jean-Michel Redouté

I declare that this master thesis entitled *A wearable miniaturized wireless multisensor of physiological signals for continuous monitoring* is the result of my own work except as cited in the references.

FYON Arthur
s160913

Department of Montefiore Institute: Electrical Engineering and Computer Science
University of Liège

Date: June 9, 2021

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my research supervisor, Jean-Michel Redouté, his PhD student, Hervé Pierre, and all workers at the laboratory Microsys for giving me the opportunity to do research and providing invaluable guidance throughout this work. Pr. Redouté's dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the work and to present the work as clearly as possible. It was a great privilege and honor to work and study under his guidance.

I am greatly indebted to my honorable teachers of the Department of Montefiore Institute: Electrical Engineering and Computer Science at ULiège who taught me during the course of my studies. Without any doubt, their teaching and guidance have completely transformed me to the person that I am today.

I am thankful to my parents and step-parents for their unconditional love, endless prayers, caring and immense sacrifices for educating and preparing me for my future. I also would like to thank my girlfriend for her support, advice and unlimited love.

Finally, I would like to thank all the people who have supported me to complete this project work directly or indirectly.

FYON Arthur
University of Liège
Date: June 9, 2021

To my mother
Nadia
father
Didier
and love
Floriane

ABSTRACT

This project tackles the problem of miniaturizing biomedical sensors by designing a brand new wireless device used to measure vital signs using electrocardiogram and photoplethysmogram signals as well as body temperature. The starting point of this project is wearIT4Health, a similar project conducted by the research laboratory supervising me, Microsys. Adequate electronic components are discussed and chosen permitting the best miniaturization possible. Moreover, electronic schematics are presented and explained as well as the printed circuit board design of 2 prototypes: one without microcontroller, due to the worldwide shortage in integrated circuit availabilities, and one with. Furthermore, the software running the microcontroller on the board it is soldered is presented and described as well as how data is transmitted from the sensors to the microcontroller. After that, the chosen wireless communication protocol, Bluetooth Low Energy, used to transfer data from the microcontroller to the pairing device is explained as well as how data will be collected and displayed on this pairing device, being an Android smartphone. Finally, tests are conducted on both prototypes and some parts of the device are validated: the battery management system, the radio frequency part and the body temperature sensor. This project is thus a proof that biomedical sensing devices, even very recent such as wearIT4Health, can be further miniaturized to improve comfort of the patient and reduce nurses amount of work.

Keywords: vital signs sensing, miniature wireless sensor, wearable and continuous monitoring of physiological signals

Contents

List of Figures

List of Tables

Introduction

In the world of today, monitoring physiological signals and medical data is one of the keys that made healthcare reaching such a good level. However, monitoring physiological signals is not as easy and comfortable as it could be. Two cases are mentioned here as an example:

- Monitoring hospitalized patients. In order to diagnose patients, it is necessary to hook them up to several devices monitoring physiological signals and vital signs such as heart rate (HR), oxygen saturation (SpO2) and body temperature. This situation is not comfortable since many cables are connected to electrodes or devices on the patients. Moreover, nurses take a lot of time to turn on each device as well as to connect correctly all monitoring devices to the patients[1].

- Monitoring non-hospitalized patients. Especially in the case of patients with suspected heart disease, the cardiologist can ask the patient to wear a holter monitoring electrocardiogram (ECG) during several days in order to diagnose the patient. This holter is often large, uncomfortable and restrictive in what the patient can do.

Thus, the development of a new generation of wearable biomedical monitoring devices becomes more and more demanded. This new generation would consist in very small wireless devices used to monitor physiological signals and vital signs so that comfort for the patient is ensured and physicians can diagnose at best the patient. As it will be discussed, this master thesis will try to meet this need by designing a new biomedical monitoring device (mainly for monitoring vital signs) being smaller, more comfortable and more wearable than those existing today.

Before entering in the core of this project, the state of the art of wearable biomedical sensing will be presented as well as a description of the wearIT4Health project being the starting point of this master thesis. After that the contribution of this project to this research field will be detailed.

State of the art of wearable biomedical sensing

As the demand of a more compact and comfortable new generation of biomedical sensors is increasing, a lot of research projects have contributed in this domain. Indeed, Wu et al. have designed a new rigid-flex wearable health monitoring sensor being connected to an IoT (Internet of Things) cloud to share vital signs data[2]. This device is innovative in its design being a spider-like device in which unused sensors on specific legs can be removed, thus reducing its size. Moreover, Thomas et al. have designed a wristwatch which can be used to calculate blood pressure without invasive instruments[3]. To do so, ECG and photoplethysmogram (PPG) signals are measured and fed to a model to extract approximations of blood pressure. Furthermore, Asada et al. developed a wearable ring used to continuously measure heart rate thanks to PPG sensors sharing data through a radio frequency (RF) module whose carrier frequency is 915 MHz[4]. However, this last device is relatively old and new much smaller PPG sensors such as the ones used in hospitals have been developed.

As demonstrated above, many researchers around the world are working on this new generation of biomedical sensors and several advantages are brought compared to the traditional medical ones: portability, compactness, comfortability for the patient, easy placement for the nurses, wireless and direct communication with the information technology (IT) network of hospitals, autonomous thanks to batteries, etc. Some of these projects are carried at the University of Liège by the research laboratory Microsys with which I realized my observation internship and master thesis[5]. Indeed, I was interested in this laboratory because their field of research is the development of ultra-low power miniaturized systems of connected sensors, whose application may be biomedical. What made me turn to it is one of their last research projects in the domain of biomedical sensing: wearIT4Health.

wearIT4Health is a "Wearable Integrated Technology for Health Monitoring of Hospitalized Patients in the Euregio Meuse Rhine"[6] that has been developed the last few years. It consists in a wearable multi-sensor device monitoring physiological signals that is comfortable for the hospitalized patients and compatible with the IT infrastructure of Euregio Meuse Rhine hospitals. The device is compact and measures all the signals that need to be monitored (SpO2, ECG, HR, etc.) and automatically sends data to the patient's EMR (Electronic Medical Records) wirelessly. It thus increases comfortability of the patient and reduces nurse's amount of work[1].

As a final step of the project, the objective is to obtain a validated/demonstrated monitoring device in a relevant environment, in this case in a hospital on test subjects and on which all certification constraints will have to be taken into account. The final Technology Readiness Level Description (TRL) of the monitoring device will be the end of TRL5 (Proof of concept or design refinement) or the beginning of TRL6 (Preclinical Evaluation)[1]. A picture of the prototype which has been tested at the CHU of Liège is represented in Figure 1.

Figure 1: wearIT4Health prototype[7]: around 5 cm x 8 cm.

Note that there also exists a variant of the wearIT4Health project which is the wearIT4COVID one. As the name indicates it, this variant of the wearIT4Health project is specialized for the detection of the COVID-19. Indeed, COVID-19 patients can deteriorate very rapidly and continuous monitoring of vital parameters are necessary to react rapidly and initiate oxygen therapy. Thus, continuous measurement is crucial for early detection of deterioration and minimizes undesirable physical contact between the patient and nurses. In particular, it has been shown

that respiratory rate (RR), SpO2, HR, and heart rate variability (HRV) are important parameters in COVID-19[8].

Contribution of my master thesis

My master thesis consists in improving the research project wearIT4Health and to contribute to advances in the field of biomedical sensing. Indeed, starting from the wearIT4Health project, I will create a totally new electronic device that will also monitor several physiological signals to deduce vital signs, for which the main goal is to miniaturize as much as possible the final implementation. Indeed, it has been discussed with Pr. Redouté that a much smaller device than the wearIT4Health prototype shown in Figure 1 can be created using brand new electronic components and integrated circuits (IC). Thus, my master thesis consists in the design, realization and test of a brand new biomedical multi-sensor that will be miniaturized as much as possible starting from the wearIT4Health project.

Moreover, the wearIT4Health prototype is not perfect and some shortcomings remain. These will be presented later and solved when designing my device. However, the purpose of my device compared to the one of wearIT4Health will be slightly different. Indeed, my device will not specially be focused on hospitalized patients of the Euregio Meuse Rhine hospitals and will be more accessible. It will be useful for any patients anywhere and will be connected to any smartphone equipped with a Bluetooth module since my device will communicate wirelessly. My device will at first measure PPG, ECG and body temperature signals so that 2 out of 4 vital signs will be deduced and more signals could be added later if the first prototype is working well. Finally, it will be powered by a battery and so will be portable and comfortable for the patient.

To sum up, my master thesis will try to show that miniaturizing prototypes such as wearIT4Health is possible while ensuring a good functioning, even if the targeted patients are different. To achieve this, I will use the most recent commercially available electronic components and integrated circuits and I can rely on the

help of Pr. Redouté, Hervé Pierre and the Microsys team.

In the following of this report, I will first give a detailed background about the physiological signals. Then, I will describe what electronic components I chose and the reasons they were chosen. Electronic schematics and printed circuit boards (PCB) designs will be shown and explained. After that, code for running the Bluetooth microcontroller (MCU) will be presented as well as the one use for the app of the target smartphone. Finally, tests and improvements will be discussed.

Chapter 1

Vital signs and physiological signals

Before entering into the description of the project design, a brief revision, information and background about the physiological signals and what can be diagnosed with them have to be given. These physiological signals are important since they give the opportunity to deduce vital signs. These signs, being a group of 4, are very important since they indicate the status of the body's vital functions. These are used to assess a patient's global health, give clues to possible illnesses and show progress toward recovery. The vital signs are the body temperature, the heart rate, the respiration rate and the blood pressure[9]. As described below, the first two are the ones of interest in this project because they are measurable with small non-invasive sensors[1].

In the following, a description of what main physiological signals we can monitor and what can they say in terms of diagnosis will be given. The deduction of vital signs from the physiological signals will also be mentioned.

1.1 Electrocardiogram

1.1.1 Biological description

One of the most famous and useful physiological signal is the ECG. The ECG signal results from the electrical activity of the heart when it contracts periodically[11].

[1]Except for the conventional and uncomfortable cuff, blood pressure sensing involves invasiveness (even if some non-invasive approximation are possible) while respiration rate sensors often consist in extendable belts (recent research shows that respiration rate can be measured with small strain gauge without a belt)[10].

Indeed, myocardial cells contract under the effect of an action potential being a depolarization of the cellular membrane due to the action of ion gate openings. Thus, when monitoring the voltage between two electrodes placed on specific regions of the chest, one can observe a typical signal being the manifestation of electrical activity of all myocardial cells one after the other during the propagation of the muscle wave. In this report, 2 leads ECG will only be discussed even if more advanced ECG recording techniques exist with, for instance, 7 leads in order to have different views of the heart. An example of a typical health ECG signal period is represented in Figure 1.1 as well as the name of particular regions of this signal.

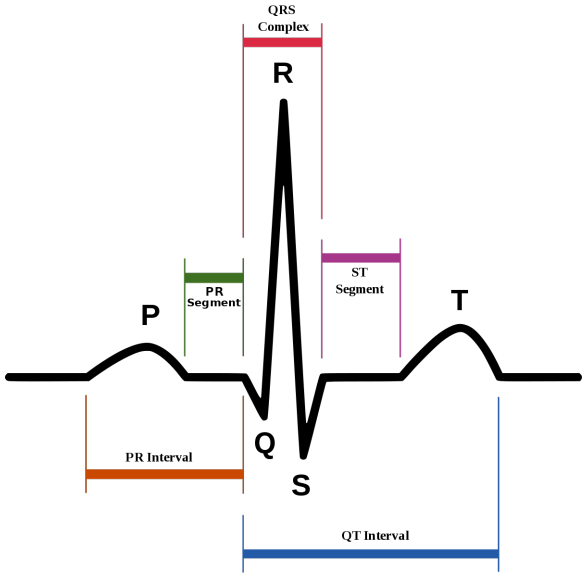


Figure 1.1: Example of an healthy ECG signal[11].

1.1.2 Diseases that can be detected

In Figure 1.1, one can observe the different parts of an ECG signal period and any modification in the waveform or the frequency of this signal can be indicative of different diseases. In the following, a non-exhaustive list of different diseases that can be detected with the ECG signal will be drawn[12].

- Heart enlargement: diagnosed by looking at the form of the ST segment or the QRS complex. An example of left ventricular hypertrophy ECG signal (in which a straight ST segment can be observed) is represented in Figure 1.2.

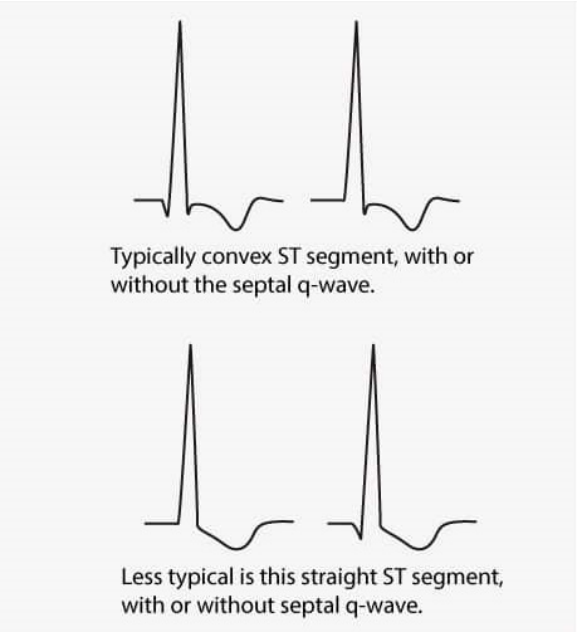


Figure 1.2: Example of heart enlargement ECG signal[13].

- Arrhythmia: diagnosed by irregular heart beats.
- Heart inflammation such as pericarditis (an inflammation of the pericardium): different stages of the disease can be diagnosed by very particular ECG waveform changes. The 4 ECG stages of pericarditis are represented in Figure 1.3.

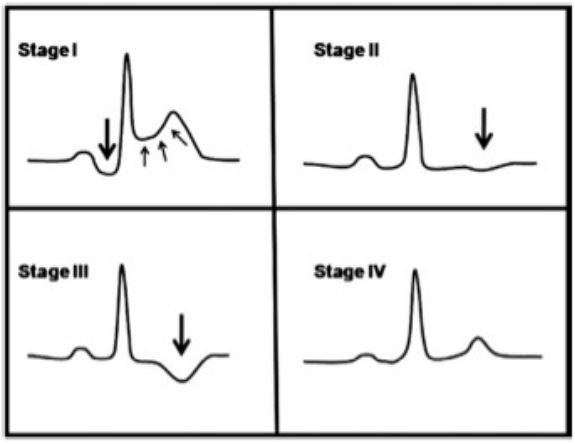


Figure 1.3: Example of 4 stages of pericarditis ECG signal[14].

- Cardiac arrest: diagnosed by a flat ECG signal.

- Abnormal level of electrolytes controlling electrical activity of myocardial cells thanks to ion gates: diagnosed by specific changes in the ECG waveform. For instance, changes in the concentration of calcium can be diagnosed thanks to the position of the T wave as represented in Figure 1.4.

Figure 1.4: Example of hypercalcaemia and hypocalcaemia ECG signal[15].

1.1.3 Measuring method

Measuring an ECG signal is theoretically straightforward: two electrodes (or more) are put on the chest of the patient and a device monitors the voltage between these electrodes. However, in practice, some precautions and design rules have to be taken into account.

First, one must take care of common mode parasitic signals. Indeed, the common mode voltage of the two ECG electrodes is the voltage of the body (with respect to some ground) and can vary a lot. Thus, one should maximize the common mode rejection ratio (CMRR) to minimize the impact of this parasitic signal on the read signal. This can be achieved using a good instrumentation amplifier and by a symmetrization of the two input ways (to avoid conversion of common mode to differential mode).

Then, since ECG signals are at relatively low frequencies (around 1-100 Hz), it is possible to attenuate high frequency parasitic signals that may come from inductive coupling. To do so, a low-pass filter can be applied. Note also that DC offset can also be observed in ECG signals due to the electrodes and attenuating it by a high-pass filter (having a very low cutoff frequency) is convenient.

Finally, mono frequency parasitic signals being in the same range of frequency as the ECG signal can be observed such as the 50 Hz parasitic signal coming from the mains. To minimize the impact of this, one can use a notch filter centered on the parasitic frequency.

1.2 Photoplethysmogram

1.2.1 Biological description

The PPG signal is nearly always monitored for hospitalized patients in order to compute the oxygen saturation (being the percentage of oxygenated hemoglobin over all hemoglobin molecules before delivering oxygen to cells) and heart rate. It often consists in a sensor that is clipped around one finger. As the name indicates, PPG is an optical method to measure plethysmogram signal consisting in measuring the change in volume within the body of an organ[16]. In the case of PPG signal, the goal is to measure volume changes induce by blood flooding periodically in the blood vessels. Indeed, as the blood flows in the body, the light absorption properties of the body changes and these changes are measured. Thanks to this, HR can be measured quite precisely using the period of the PPG signal.
Moreover, oxygen saturation can be measured because the proportion of oxyhemoglobin over all hemoglobin changes the light absorption property of the body, and because oxyhemoglobin and deoxyhemoglobin have different extinction coefficients with respect to red and infrared (IR) lights. As can be seen in Figure 1.5, signal amplitude is different when considering different light frequencies.

1.2.2 Diseases that can be detected

On the contrary of the ECG signal, the waveform of the PPG signal is not indicative of certain diseases. Indeed, the waveform can simply vary due to the position of the sensor on the body or due to motion artifacts. Thus, only the measured HR and SpO2 will be indicative of a certain disease. As already discussed for the ECG signal, HR can be used to diagnose arrhythmia.

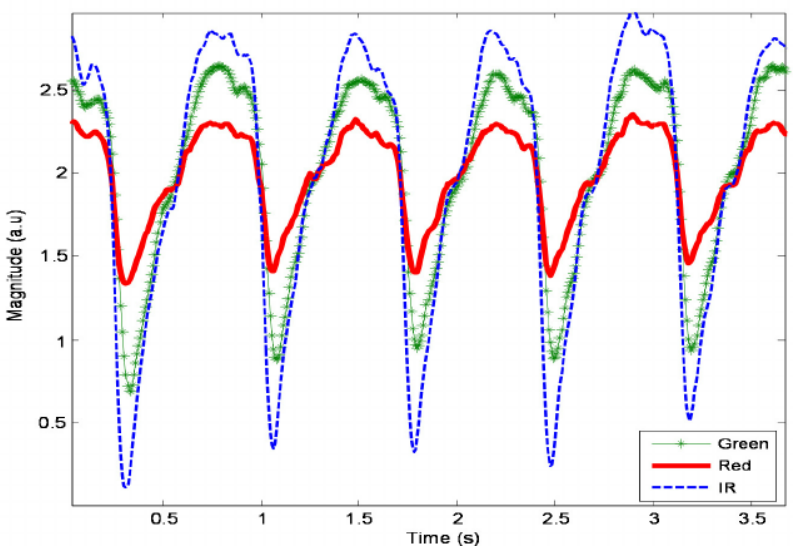


Figure 1.5: Example of an healthy PPG signal for different light frequency[17].

Concerning the oxygen saturation, a healthy patient one will always be high, i.e., larger than 95%[18]. On the contrary, a low SpO2 level is alarming and can be indicative of multiple diseases[19]:

- hypoxia;
- blood or water filling alveoli in the lungs;
- a person holding his breath, that may be due to gas intoxication;
- scarring or loss of lung tissue;
- COVID-19: some people exposed to COVID-19 with symptoms had a very low SpO2;
- ...

Thus, monitoring oxygen saturation is crucial to rapidly react to any dysfunctions of the human body.

1.2.3 Measuring method

A PPG sensor is one of the most basic biomedical sensors, it consists in light emitting diodes (LEDs) and photodiodes (PDs) in its simplest form. Indeed, light extinction coefficient changes have to be measured. To do so, LEDs emit light in the

body and photodiodes measure either reflected or transmitted light being representative of light extinction coefficient changes. Two configurations are possible: either both LEDs and photodiodes are on both sides of the body and the measured light is transmitted through the body either both LEDs and photodiodes are on the same side of the body and the measured light is reflected by the body.

As shown in Figure 1.5, different LEDs emitting different light frequencies are used to measure several PPG signals. Usually, green light is used to measure HR because measuring this frequency allows the best ambient light rejection since green is not a natural light. Red and infrared lights are both used to compute SpO2 due to different light extinction coefficient when considering oxy- and deoxyhemoglobin as already explained, formula will be derived later. Differences in light extinction coefficients are represented in Figure 1.6.

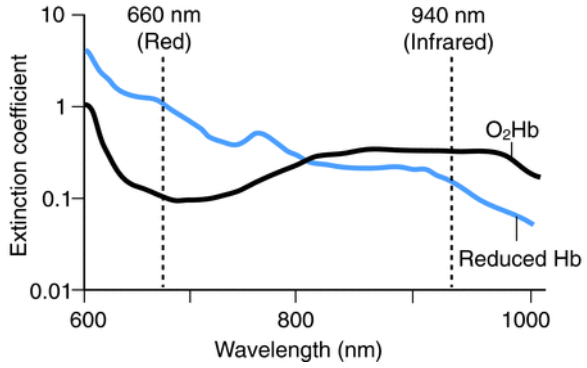


Figure 1.6: Differences in light extinction coefficients (extinction coefficients) of red and infrared when considering oxyhemoglobin (O_2Hb) and deoxyhemoglobin (reduced Hb)[20].

Finally, the PPG signal is thus contained in the current passing through the photodiodes (being proportional to the incoming light) and parasitic signals may be involved. To attenuate them, filters similar to the one described for the ECG signal can be used.

1.3 Electrodermal activity (EDA)

1.3.1 Biological description

The EDA (or skin conductance) is a measure of the impedance of the skin on its surface reflecting the activity of the sweat glands. Moreover, sweating is controlled by the sympathetic nervous system; thus, EDA is an image of psychological/physiological arousal. Indeed, there is a relationship between emotional arousal and sympathetic activity: the more the sympathetic branch of the nervous system is aroused, the more the sweating and the higher the skin conductance (physiological liquid conducts well electricity), and vice versa. Thus, a relationship between EDA and emotional arousal is established even if the type of emotion inducing changes in EDA can not be identified[21].

To sum up, EDA is a common measure of autonomic nervous system activity, even if recent research tends to show that EDA is much more complicated than that[21]. An illustration of an EDA signal is represented in Figure 1.7.

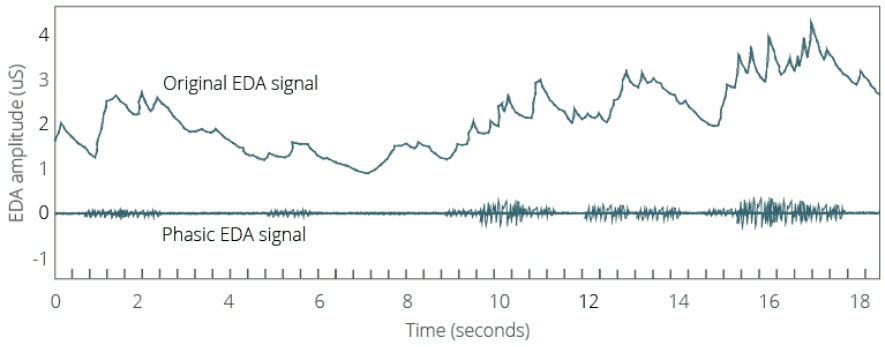


Figure 1.7: Example of an EDA signal[22].

As can be seen in Figure 1.7, two components appear in the EDA signal: a tonic base level drive fluctuating very slowly (periods in the order of magnitude of seconds) and a fast-varying component (periods in the order of magnitude less than seconds). Thus, this phasic component can be extracted from the original signal by applying a high-pass filter to the original signal, for instance. These phasic changes are a response to a certain external stimulus such as images, videos

or sounds while the slow varying changes reflect the general state of arousal and alertness[22].

1.3.2 Diseases that can be detected

Even if EDA gives a lot of information about the patients, it is not really used to detect specific diseases. Instead, it gives an information about the patient's neurological status without using uncomfortable methods, such as the electroencephalogram examination. Moreover, some recent studies suggest than the EDA could be used as an automatic detection of major depressive disorder[23] or as a method of pain assessment in premature born infants[24]. It can also be used along with other physiological signals such as PPG to calculate hydration of a patient.

1.3.3 Measuring method

The method of measuring EDA comes from the Ohm's law: it consists of circulating a very small current between two electrodes on the skin (such that the current passes through the skin) and to monitor the voltage between these electrodes. Thus, one can compute the electrical conductance G of the skin (generally in µS):

$$G = I/V$$

Again, filters such as high-pass and low-pass filters are often used to reduce the impact of parasitic signals. However, the changes in skin conductance are very tiny and detecting them can be challenging. Thus, to amplify these changes, the skin conductance can be used as one leg of an unbalanced Wheatstone bridge, followed by a differential amplifier (or, better, an instrumentation amplifier)[25] as shown in the schematics in Figure 1.8.

1.4 Body temperature

1.4.1 Biological description

A last obvious physiological signal is the body temperature. This is a measure of body's ability to produce and release heat[26] and is remarkably maintained within a range (36.1°C - 37.2°C)[27] if there are no abnormalities thanks to homeostasis.

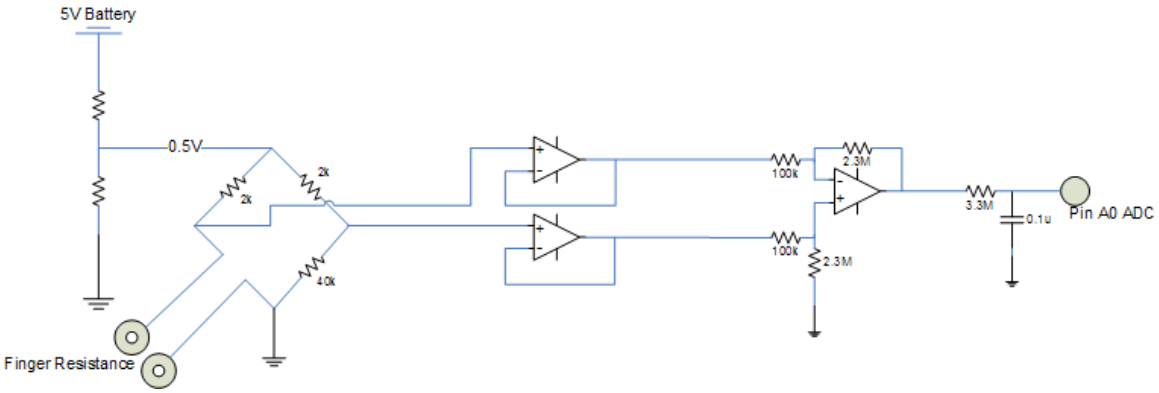


Figure 1.8: Example of an EDA sensor hardware[25].

1.4.2 Diseases that can be detected

Any temperature measure out of this range can be indicative of many dysfunctions. When the body temperature is higher than the upper bound of the normal range, we talk about fever or hyperthermia while we talk about hypothermia when the body temperature is lower than the lower bound. On one hand, hyperthermia can be caused mainly by infection but it can also be due to inflammation, blood clots, hormone disorder, autoimmune diseases, COVID-19, etc[28]. On the other hand, hypothermia is often caused by prolonged exposures to cold temperature (for instance due to wearing clothes that are not warm enough) and rarely by any biological diseases[29].

1.4.3 Measuring method

Basically, measuring human body temperature is anything else than measuring the temperature of an object. Thus, any temperature sensor would work. However, there exist temperature sensors that are optimized for the normal range of body temperature and those should be used for biomedical application. Two choices are offered: either the temperature measure is done without contact using infrared sensor either it is done with a close contact of the sensor with the skin using, for instance, a thermistor having a negative temperature coefficient (NTC). To use this last type of sensor, correct temperature measurement are assumed by ensuring a good thermal contact between the skin and the sensor.

Chapter 2

Design choices and electronic components

Before presenting the electronic schematics and PCB, the functionalities and purpose of the device have to be set, as well as which electronic components and integrated circuits will be used.

2.1 Design choices

2.1.1 The physiological signals and targeted patients

As already said, my device will be a miniaturized version of wearIT4Health. Thus, the first design choice we encountered was the signals the device will monitor. As the delay to develop this new device is much shorter than the one accorded to wearIT4Health, only the most important physiological signals will be monitored: ECG, PPG and body temperature. Indeed, it is better to start with a first prototype with only few signals in order to iterate the design so that new physiological signals such as EDA or blood pressure estimation can be implemented. Moreover, monitoring these 3 signals only can already give very informative medical data such as heart rate or oxygen saturation. Another design choice is that the device will not be exclusively compatible with the IT infrastructure of some hospitals but will rather intended to be used by anybody, enlarging the targeted patients to not be restricted to hospitalized patients in the Euregio Meuse Rhine hospitals.

2.1.2 The wireless protocol

Then, a wireless communication protocol has to be chosen. Concerning the prototype of wearIT4Health, it communicates thanks to the Wi-Fi protocol. However, this kind of wireless communication is not easy to implement and requires more power than some other protocols. Moreover, compatibility with some IT infrastructure has to be implemented and thus the device would not work with any IT infrastructure whose compatibility is not implemented. Thus, the device of this project would use another protocol to communicate since it has been chosen that it would be more portable and compatible with any external device, without depending on a hospital IT infrastructure. To do so, two main choices were considered:

- Bluetooth protocol: a widely spread protocol used nowadays for nearly every daily wireless device,
- IEEE 802.15.4 protocol: a low energy and easy to use protocol.

The IEEE 802.15.4 protocol offers many advantages such as very low power thanks to power management functions, easy to implement on any compatible MCU and very low cost[30]. However, the main disadvantage is that an adapter has to be connected to a computer to receive data via IEEE 802.15.4. Moreover, the arising of the Bluetooth Low Energy (BLE) protocol as one of the most used by wireless devices to communicate with a phone or a laptop in a very low power way have discarded IEEE 802.15.4 of being the wireless protocol.

The BLE offers a way of communicating consuming 10 times less power than traditional Bluetooth while keeping the same order of magnitude of data rate[31]. Moreover, every nowadays smartphone or laptop and a lot of MCUs are compatible with BLE making the device portable and easy to use since, for instance, only a smartphone has to be used to receive data from BLE devices such as a fitness watch. The only disadvantage of BLE compared to IEEE 802.15.4 is that no Microsys member has experience in it and the MCU software will be more complicated. Thus, for all these reasons, the chosen protocol of this master thesis project

will be the BLE, mainly for its universal aspect and easy to use in the patient/-physician side (everybody can use the Bluetooth module of its smartphone).

2.1.3 Power management part

After that, the power management system has to be discussed. Either the device relies on a battery or not that could be rechargeable or not as well. For this project, the device will rely on a rechargeable lithium-ion battery so that the device will be portable and the battery will not need to be replaced each time it is empty. However, there is a trade off between battery capacity and space requirement: the larger the battery, the higher the capacity. Since the goal of this project is to miniaturize as much as possible the size of the complete device, the smallest battery will be selected while filling some requirements.

Concerning the model of the battery, button cell will be used since these batteries can be very small and thin. They can also deliver acceptable voltage while having a large enough capacity to power the device (everything will be low power as will be discussed later). Moreover, battery holders are easily found on the market for these types of batteries so that custom holders will not be needed to be designed. The exact model of the battery will be discussed later since some requirements involve chosen electronic components.

2.2 Electronic components and integrated circuits

In this section, the electronic components that were chosen for this master thesis device will be shown and comparison with some other potential devices will be drawn.

2.2.1 Microcontroller

The core of any embedded system projects like this one is the MCU, being the brain of the device. Indeed, this electronic device is always necessary when some codes have to be run and acts as the processor of the device. It thus will run some codes

to realize the tasks it has to.

The main characteristics of the MCU for this project are: being a BLE MCU, ultra low-power, as small as possible and a minimal number of passive components. Indeed, to minimize the size of the device, it has been chosen that the MCU will have a BLE module on-board instead of an external RF module. Moreover, the MCU has to be available in Belgium and, for programming facilities, it would be easier to have an 16-bits or 32-bits MCU instead of an 8-bits one.

With these constraints and after an intensive investigation and selection process, 3 BLE MCUs from 3 different constructors emerge: the CC2640 family from *Texas Instruments*, the nRF528XX family from *Nordic Semiconductor* and the DA1458X family from *Dialog Semiconductor*. All the 3 are small ultra-low power BLE (at least 4.2 versions) MCU. Table 2.1 draws a comparison of the main characteristics of the 3 MCUs. The chosen models for the comparison are the most suited for this device (often the smallest). Note that two types of packages will be discussed: Quad Flat No-leads (QFN) package being pinless packages and Wafer-Level Chip Scale Packaging (WLCSP) package being just a bare die with only a redistribution layer to redistribute the pins on the die so that they can be large enough to be soldered[32]. This last type of package is the smallest nowadays but is obviously harder to solder.

Looking at Table 2.1, it clearly appears that the nRF52805 is the best BLE MCU for this project. Indeed, it is nearly the smallest (the DA14580 is only slightly smaller), it possesses the best CPU core and the clock speed is the fastest if necessary. Moreover, it can be seen that it is the lowest power MCU from the 3 by looking at the sleep mode current and peak current in TX (transmission) and RX (receiver) mode. Furthermore, it has the best receiver sensitivity. Only the DA14580 has more RAM quantity.

However, the chosen MCU will be the CC2640 for several practical reasons.

- WLCSP packages are hard to solder and not suited for a first embedded system project like this one. Thus, QFN packages are preferred when available.

Characteristics	CC2640	nRF52805	DA14580
Smallest QFN size	4 mm x 4 mm	Do not exist	6 mm x 6 mm
Smallest WLCSP size	Do not exist	2.5 mm x 2.5 mm	2.436 mm x 2.436 mm
CPU-core	Arm Cortex-M3	Arm Cortex-M4	Arm Cortex-M0
Max Clock speed	48 MHz	64 MHz	16 MHz
Supply voltage	1.8-3.8 V	1.7-3.6 V	0.9-3.3 V
Sleep mode current with RAM retention	1 µA	0.8 µA	1.2 µA
Peak current TX	6.1 mA	4.6 mA	4.8 mA
Peak current RX	6.1 mA	4.6 mA	5.1 mA
Receiver sensitivity	-97 dBm	-97 dBm	-93 dBm
RAM	20 kbit	24 kbit	48 kbit

Table 2.1: Comparisons between different microcontrollers from several constructors[33][34][35].

- The Microsys team has already programmed MCU from *Texas Instruments* contrary to *Nordic Semiconductor* and *Dialog Semiconductor*. Thus, they can better help me with the CC2640 compared to the two other MCUs.
- As it will be seen later, the I2C interface will be needed for this project but the nRF52805 does not support this interface.
- *Texas Instruments* offers a really good training for new developers like me called the *SimpleLink* academy[36] on the contrary to the other brands whose training seems less complete. Moreover, *Texas Instruments* proposes a very good integrated development environment (IDE) , which had been used by the Microsys team, with a lot of code examples helping the developers to code the BLE MCU.
- The evaluation board of the CC2640 is easily available on *Farnell* compared to the other MCU evaluation boards.

Thus, even if the most logic and efficient choice would be to take the nRF52805, the CC2640 is chosen for practical reasons. Moreover, since this is a very short in time embedded project, it is preferred to focus also on the feasibility of the project rather than focusing only on the efficiency. It is also preferred to start on a known basis instead of learning a totally new IDE, etc. Figure 2.1 represents the CC2640 in its smallest QFN package.

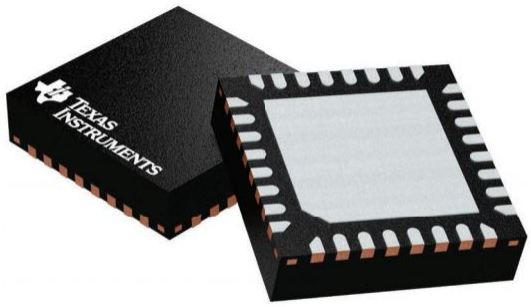


Figure 2.1: CC2640 in the 4 mm x 4 mm QFN package[37].

2.2.2 Power and battery charge management solution

The charge of a battery is not straightforward and an electronic device in charge of that and of supplying the other devices is needed. Indeed, there are 3 main phases when charging a lithium-ion battery: a precharge followed by a constant current phase finished by a constant voltage one[38]. First, the precharge is a low-rate charging phase until a threshold voltage of the battery necessary to provide recovery of the passivating layer which might be dissolved after prolonged storage in deep discharge state is reached. Then, the constant current charging phase occurs to rapidly charge the battery until the battery voltage reaches a predefined voltage. After that, the charging voltage stays constant and the charging current decreases exponentially until a certain threshold called the termination current to terminate the charging. This combination permits to have a fast-charging method without the risk of overcharging and is suitable for lithium-ion and other types of battery. Indeed, only charging a battery at constant current can lead to overheating of the battery if it is overcharged, reducing drastically the battery life.

The electronic device in charge of the power and battery charge management is critical and choosing the right one and carefully design the connections to other integrated circuits is tricky. Since I do not have any experience in power electronics and under the advice of Hervé Pierre, I took the same power and battery charge management device than the one in wearIT4Health: the BQ25125 from *Texas Instruments*[39].

Indeed, this electronic device is totally suited for my device.

- It has a low quiescent current ($I_Q \approx 700nA$) being the current drawn by the device in a no-load and nonswitching but enabled condition. This is required for low power, thus extending battery life.
- It enables a fine control of the battery charging, being in 3 phases, such as tuning the charge current, the termination current, etc. It handles the 3 states of standard lithium-ion battery charging.
- It is very small: 2.5 mm x 2.5 mm in a Die-Size Ball Grid Array (DSBGA)[1] package.
- It can provide two output voltages: one with a buck regulator and one with a low-dropout regulator (LDO).
- A battery voltage monitoring is also possible using specific register of the BQ25125.

Thus, the BQ25125 is the perfect candidate for the power and battery charge management in this project. Note that communication between the BQ25125 and the MCU is made through I2C interface. Figure 2.2 represents the BQ25125.

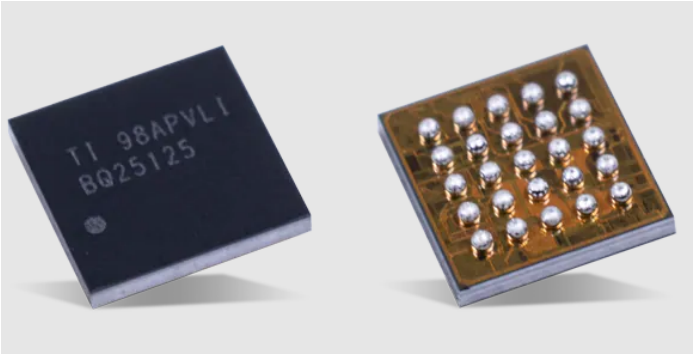


Figure 2.2: BQ25125 in its DSBGA package[40].

2.2.3 PPG and ECG sensor

In its simplest form, an ECG sensor consists in 2 electrodes and a PPG sensor consists in LEDs and photodiodes. However, there exist a lot of analog front end

[1]This package is equivalent to the WLCSP one.

electronic devices used to measure such signals by filtering and converting them into digital signals using analog to digital converter (ADC). Thus, it is convenient to use them since they are designed for measuring such physiological signals.

In wearIT4Health, both the AFE4400[41] and the ADS1292R[42] from *Texas Instruments* were used, being an integrated analog front end for heart-rate monitors and low-cost pulse oximeters and a 2-Channel 24-Bit ADC with integrated respiration impedance and ECG front end respectively. Obviously, only one of the two devices were used to measure the ECG signal. Measuring an ECG signal is straightforward and was done as a project in the Bioelectronics course while measuring PPG might be tricky. Indeed, choosing the right LEDs and photodiodes is not straightforward and driving the LEDs and the photodiodes must be accurate in terms of timing (the time the LEDs flash and the time when the device receives the signal from the photodiodes). Thus, choosing another device such as the MAX86150[43] being an integrated photoplethysmogram and electrocardiogram bio-sensor module for mobile health from Maxim Integrated would not be the best choice since this would thus mean that I would start from zero.

As a consequence, I was first ready to use the AFE4400 as the PPG and ECG sensor but a brand new electronic device was released on the market: the ADPD4100/01 from *Analog Devices*. It is a multimodal analog front end used for wearable health and fitness monitors: heart rate, heart rate variability, stress, blood pressure estimation, SpO2, hydration, body composition and industrial monitoring (CO, CO2, smoke, and aerosol detection) and home patient monitoring. More specifically, "the multiple operation modes of the ADPD4100/01 accommodate various sensor measurements, including PPG, ECG, EDA, impedance, capacitance, temperature, gas detection, smoke detection, and aerosol detection for various healthcare, industrial, and consumer applications"[44]. Thus, it basically can be used to measure any physiological signal for this project. Moreover, it is a very small device: 3.11 mm × 2.14 mm, 33-balls WLCSP and 35-balls WLCSP. Furthermore, the supply voltage is 1.8 V, which is perfectly compatible with the other devices. The ADPD4100 is represented in Figure 2.3

Figure 2.3: ADPD4100 in its WLCSP package[45].

Concerning the technical characteristics, the ADPD4100/01 can communicate with the MCU either by SPI (ADPD4100) either by I2C (ADPD4101). It can also drive up to 8 LEDs and measure the return signal on up to 8 input channels being current inputs. Moreover, the analog front end (AFE) rejects signal offsets and corruption from asynchronous modulated interference, typically from ambient light, eliminating the need for optical filters or externally controlled DC cancellation circuitry. Thus, it acts as a hub for synchronous measurements of photodiodes, biopotential electrodes, resistance, capacitance, and temperature sensors.

Given all the characteristics available in such a small and recent device, Ir. François Dupont, the Microsys laboratory supervisor, proposed to use this device. The LEDs and photodiodes that will work as a pair with the ADPD4100/01 will be the same as the ones used in wearIT4Health and are included in the SFH7072 bundle being a multi-chip package featuring two green emitters (LEDs), one red emitter, one infrared emitter and two detectors (photodiodes) optimized for a strong PPG signal[46]. This bundle size is 7.5 mm x 3.9 mm and is thus perfectly suitable for this master thesis and is represented in Figure 2.4. Finally, since the BQ25125 also has an I2C communication interface, it has been chosen that the ADPD4101 will be used so that every electronic device will be connected on the same I2C bus.

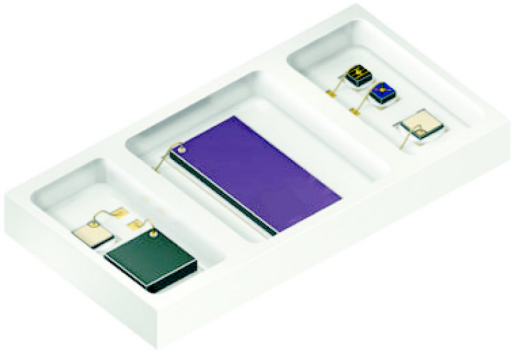


Figure 2.4: SFH7072 bundle[46].

2.2.4 Temperature sensor

Even if the ADPD4101 seems to measure all possible signals, it is not explained in the datasheet how it can be used to measure temperature. After a discussion with the *Analog Devices* support, it seems that an external device is necessary to measure temperature. For measuring body temperature, there are two possibilities as already explained: either the measurement is contactless either it is not. However, I cannot use the same temperature sensor as in wearIT4Health since, in this last device, there was always a temperature offset of around 5°C, which is unacceptable while measuring body temperature. Moreover, some precautions will be taken when designing the PCB as it will be seen.

First of all, one needs to decide if the temperature measurement will be contactless or not. Contactless temperature sensors use infrared light to measure body temperature (such as the MLX90632[2][47], a miniature digital infrared thermometer) and are often more precise than a simple thermistor or any contact temperature sensor. However, the power consumption is much higher in the case of contactless sensors and the size is also often larger. For these last reasons, a non contactless temperature sensor will be used. For convenience, an I2C compatible device will be chosen so that every integrated circuit will be connected on the same bus.

Then, a first idea would be to use the LMT70 from *Texas Instruments* being a precision analog temperature sensor whose accuracy around normal body tem-

[2]This sensor could have been chosen since it is very small, however it was expensive.

peratures is 0.1°C and its size equals to 0.88 mm x 0.88 mm DSBGA package[48]. However, as the name indicates, it is an analog sensor while a digital device compatible with I2C would be better integrated in my device. Thus, for this last reason, we considered another (already used by Microsys in other projects) digital temperature sensor: the TMP1075 from *Texas Instruments*. It is an I2C temperature sensor with performance upgrades to industry standard LM75/TMP75. This device is compatible with I2C and provides a resolution of 0.0625°C. It is also small enough: 2 mm x 2 mm in its smallest package. Moreover, the supply voltage can be as low as 1.7 V, the device is thus perfectly integrated with the other above cited devices. Finally, it is also low power since the average consumption current is 2.7 µA and the average shutdown current is as low as 0.37 µA. The only negative point of this device is that the accuracy is around 1°C[49]. This could seem large for a body temperature sensor but it is a reliable and very easy to use device. Thus, this device is chosen despite the fact that it has a poor accuracy. The main objective of this device is to not recidivate and cancel the observed offset in wearIT4Health, probably due to heat diffusion on the PCB board. The TMP1075 is represented in 2.5.

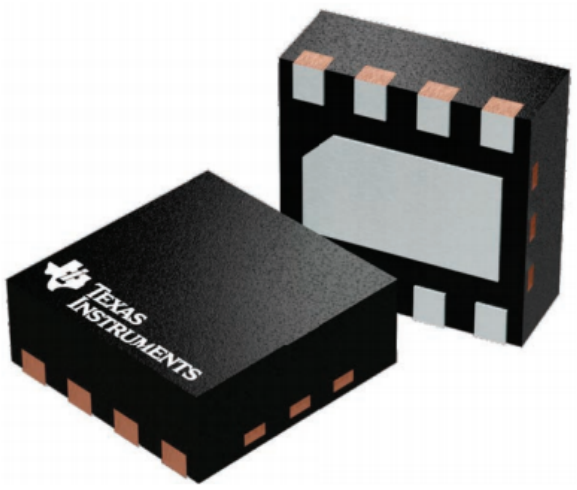


Figure 2.5: TMP1075 in its smallest package[50].

2.2.5 Battery, battery holder and charging input

As already said, rechargeable coin cells will be used in this project. Moreover, since the charging phases of the BQ25125 are the ones of a lithium-ion battery, these types of batteries will be used. This brings us to the lithium ion rechargeable (LIR) coin cell. Furthermore, LIR batteries provide 3.6-3.7 V, being compatible with the input voltage of the BQ25125.

After that the size of the coin cell has to be chosen. As already said, the larger the battery, the higher its capacity: there exist a lot of LIR dimensions whose diameters and heights vary as well as the capacity. For this project, minimizing the diameter and the height while keeping an acceptable capacity will be the focus of this choice. Thus, a good compromise between small size and enough capacity (and easy availability) is the LIR2032. This battery is 20 mm in diameter and 3.2 mm in height and has 40mAh in capacity[51]. This battery is represented in Figure 2.6.

Figure 2.6: LIR2032 being the chosen battery[52].

To connect this cell to the PCB, a battery holder surface mounted is needed. The smallest that had been found is the S8421-45R represented in Figure 2.7.

Finally, one has to define which connection will be used to recharge the battery. Indeed, the BQ25125 has an IN pin that needs to be connected to a voltage of at most 20 V to recharge the battery[39]. Thus, for convenience, the 5 V micro USB is the connection used to recharge the battery.

Figure 2.7: LIR2032 surface mounted battery holder[53].

2.2.6 Crystal oscillator

To give the clock frequency to the MCU, an external crystal oscillator is needed. In a crystal oscillator, the quartz crystal vibrates mechanically at its resonance frequency thanks to piezoelectric effect. This provides a stable electrical oscillation used as input clock frequency of the MCU.

First, the CC2640 needs an external 24 MHz crystal oscillator with some constraints summarized in Table 2.2.

Parameters	Test conditions	Minimum value	Max value
Crystal frequency		24 MHz	24 MHz
Crystal frequency tolerance		-40 ppm	40 ppm
Crystal load capacitance C_L		5 pF	9 pF
ESR Equivalent series resistance	$C_L > 6$ pF	20 Ω	60 Ω
ESR Equivalent series resistance	$C_L \leq 6$ pF		80 Ω

Table 2.2: Specifications of the 24 MHz crystal oscillator[33].

Taking a look at which crystal oscillators respecting these constraints are available on *Farnell*, the AV-24.000MAGV-T has been chosen and is represented in Figure 2.8. Obviously, its frequency is 24 MHz and the frequency tolerance is equal to 30 ppm. Moreover, its load capacitance is 8 pF and its equivalent series resistance is 60 Ω, making it perfect for the CC2640. Note that the device size is 3.2 mm x

2.5 mm which is not the smallest, but smaller devices respecting these constraints could not be found[54].

Figure 2.8: AV-24.000MAGV-T crystal oscillator[55].

After that, the CC2640 may need a 32.768 kHz for real-time application as BLE. However, there exists an alternative in which the CC2640 uses its own, however less precise, 32 kHz oscillator. This alternative is preferred in this project to limit the total size of the device by minimizing the number of passive components.

2.2.7 Antenna

To transmit data via BLE, an antenna is needed. In this project, a 2.4 GHz antenna (BLE is 2.4 GHz) will be chosen according to some criteria: maximizing the performances while minimizing the size as usual. Different models from different constructors will be compared here. The A5839H from *Antenova*, the ANT-DB1-nSP250-T from *Linx* and the AH316M245001-T from *Taiyo Yuden*. The performances of antennas can be discussed through several criteria.

- The return loss: defined as the loss of power in the reflected signal due to an impedance mismatch in the antenna. This value should be as high as possible as we want to have less reflected signal as possible[56].
- The voltage standing wave ratio (VSWR): defined as how well the antenna is impedance matched to the transmission line it is connected to. Mathematically, if the reflection coefficient is denoted Γ, the VSWR is defined as

$$VSWR = \frac{1 + |\Gamma|}{1 - |\Gamma|}$$

This parameter should be as small as possible since we do not want to have reflected signal, the minimum value of the VSWR is, of course, 1[57].

- Peak gain: defined as the maximum gain in the main beam direction as a ratio relative to an isotropic antenna source. This parameter should be as high as possible[58].

Table 2.3 compare the different antennas using the above criteria, in the range of frequency of BLE.

Performance criteria	A5839H	ANT-DB1-nSP250-T	AH316M245001-T
Maximum return loss	-11 dB	-11 dB	No data
Maximum VSWR	1.8:1	2.0:1	2.0:1
Peak gain	2.1 dBi	2.7 dBi	2.9 dBi
Size	12.8 mm x 3.9 mm	9.6 mm x 8.4 mm	3.2 mm x 1.6 mm

Table 2.3: Comparisons of different BLE antennas[59][60][61].

The smallest antenna is the AH316M245001-T and its performances are nearly equivalent to the other antennas even if the return loss is not given. However, its availability in Europe is very poor and Hervé Pierre was not confident to have such a small antenna. Thus, the A5839H, whose brand is trustworthy, had been chosen mainly due to its very good performances and size. Indeed, the antenna, represented in Figure 2.9 is elongated and not squared, like the ANT-DB1-nSP250-T, making it easily integrated in the PCB as it will be seen later. Moreover, this antenna exists in right or left version so that the corner in which it will be placed can be arbitrary.

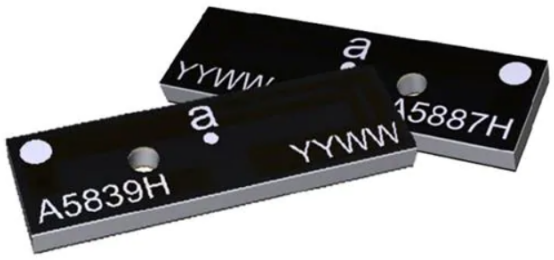


Figure 2.9: A5839H right and left version[62].

2.2.8 Additional components for testing and debugging

In this project, additional components have been bought to test the codes and to debug the custom device in case of failure of the RF part.

2.2.8.1 Evaluation board

The CC2640 also exists in the form of an evaluation board, the LAUNCHXL-CC2640R2 represented in Figure 2.10. This board will be used to test some codes in the case of BLE and I2C communications.

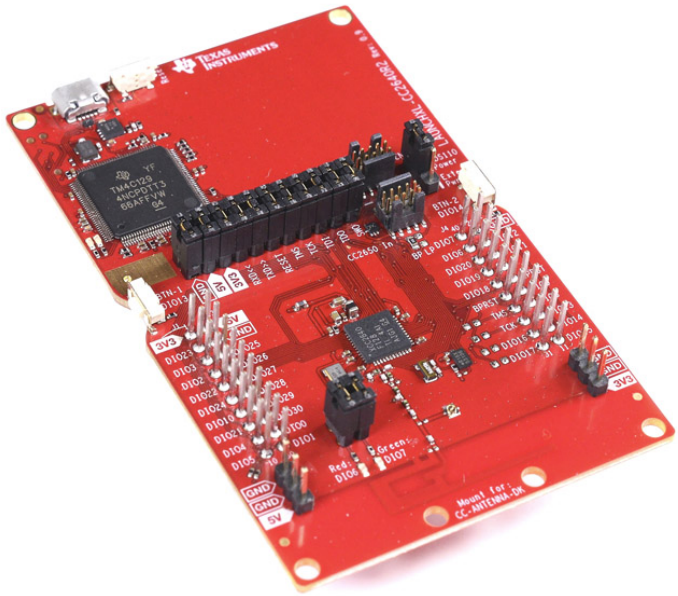


Figure 2.10: The evaluation board of the CC2640, LAUNCHXL-CC2640R2[63].

This evaluation board will also be used to program the custom device through the programmer observed on the top of Figure 2.10, the TM4C129.

2.2.8.2 Additional device in case of custom RF part failure

In case of failure of the custom RF part, I2C communication between the custom and the evaluation board can be used to still transmit data wirelessly thanks to the RF part of the evaluation board. However, the I2C bus voltage of the custom

board is 1.8 V while the voltage of the I2C bus of the evaluation board is 3.3 V. Thus, a device allowing the communication between both buses have to be added: the PCA9306 from *Texas Instruments*. It is a dual bidirectional I2C bus voltage-level translator and measures 0.928 mm x 1.928 mm in its DSBGA package. This device is illustrated in Figure 2.11.

Figure 2.11: PCA9306[64].

Chapter 3

Proposed design

In this chapter, the electronic design will be presented and described. First, the electronic schematics used to schematize the connections between all components cited in the previous chapter will be presented. Then, the PCB design, being the final hardware, will be explained. Note that both the electronic schematics and PCB have been designed in the software *Altium Designer*.

3.1 Electronic schematics

The electronic schematics include all the active and passive components as well as all the connections between these components so that communications between, for instance, MCU and sensors are ensured. These schematics are purely representative and the real positions of the tracks and electronic devices will be included in the PCB design section.

This section will be divided in 3 main parts: the power, logic and sensor parts. Thus, the final electronic schematics will be divided in 3 sheets.

3.1.1 Electronic schematics: power part

In this part, connections between the BQ25125, the battery and the charging port will be shown so that the power supplies of both the electronic devices at 1.8 V and LEDs at 3.3 V will be created thanks to the BQ25125. Moreover, the connections of the PCA9306 will also be presented.

3.1.1.1 BQ25125 pins

First of all, the pin functionalities of the BQ25125 have to be discussed in order to understand how the BQ25125 has to be connected. Figure 3.1 shows all the pins of the BQ25125 while Tables 3.1 and 3.2 summarize their functionalities. Note that only the useful pins for this project will be presented.

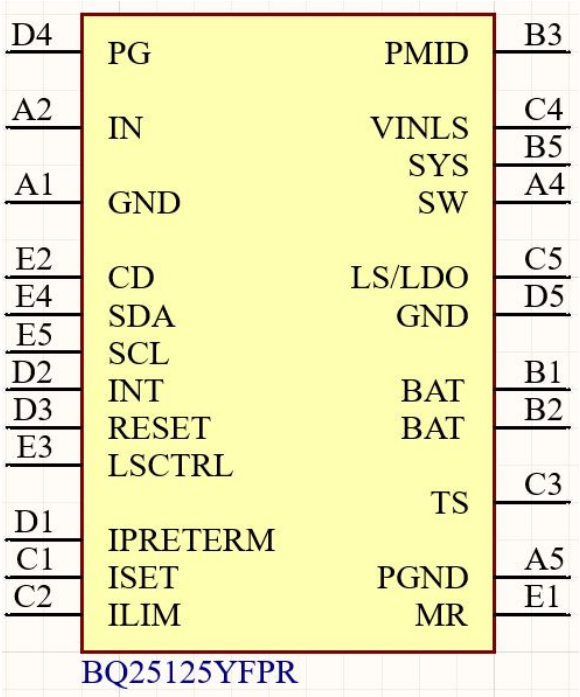


Figure 3.1: Pins of BQ25125.

Thus, following the guidelines cited in Tables 3.1 and 3.2, one has the electronic schematics depicted in Figure 3.2. This last Figure is the final electronic schematic that will be implemented in the PCB. To sum up, the BQ25125 is connected to the LIR2032 battery on its battery holder and charging can be made by plugging a 5 V micro USB cable in the micro USB port. Moreover, charging status information is conveyed through the blue LED between PMID and PG (note that R1 is just a resistor to limit the current through the LED). Furthermore, the BQ25125 has two output power supplies: $V\,\mu C$ being the stable 1.8 V power supply for all integrated circuits in the PCB and *LDO 3V* being the LDO output at 3.3 V that will supply the LEDs for the PPG. Note that R2 and R3 are just pull up resistors and that the values of the components have been chosen to be the same as in wearIT4Health.

Pin name	Description
GND	Ground
IN	DC external input power supply (max 20 V), a bypass capacitor of at least 1 µF is required. This corresponds to the micro USB power supply in this project.
PMID	High side bypass connection, a bypass capacitor of at least 3 µF is required. This pin will be set high when battery charging occurs (when a micro USB cable is connected to the device) and will only be used to feed a LED to inform the user about charging status.
CD	Chip disable, this pin will be driven by the MCU. Two modes can be defined with respect of the value of this pin. • 0: Enable battery charging and disable I2C if no charger. • 1: Disable battery charging and enable I2C if no charger.
SDA/SCL	Serial data and serial clock for I2C communications.
ILIM	Adjustable input current limit programing, connect a resistor from ILIM to GND to program the input current limit. In this project, internal default threshold for input current limit is chosen by connecting ILIM to GND.
LSCTRL	LDO control input. This will be pulled high so that the LS/LDO output (that will supply the LEDs) will be enabled.
ISET	Fast-charge current programing input, connect a resistor from ISET to GND to program the fast-charge current level. Internal default level is chosen by connecting ISET to GND.
IPRETERM	Termination current programing input. Connect a resistor from IPRETERM to GND to program the termination current threshold. Again, internal default threshold is chosen by connecting IPRETERM to GND.
INT	Status output, it is an open-drain output that signals charging status and fault interrupts. This pin will not be used in this project and will thus be pulled high.

Table 3.1: Description of the BQ25125 pins, first part[39].

Pin name	Description
PG	Open-drain power good status indication output, this pin is pulled to GND when charging occurs and will thus be used as the current sink of the visual indication LED.
RESET	Reset output, this pin will not be used and left floating.
SW	Inductor connection, have to be connected to the switched side of the external inductor.
MR	Manual reset input. This pin will be connected to ground since it will not be used.
SYS	System voltage sense connection, a bypass capacitor of at least 4.7 µF is required. This has to be connected to the system output at the output bulk capacitors, being the supply voltage of all the integrated circuits of the PCB, i.e. stable 1.8 V.
VINLS	Input to the load switch/LDO output, a bypass capacitor of at least 4.7 µF is required. As indicated in the datasheet, this pin will be connected to SYS.
LS/LDO	Load switch or LDO output, a bypass capacitor of at least 1 µF is needed to assure stability. This pin will be the supply voltage of the LEDs of the SFH7072 whose voltage will be 3.3V[46], which should be enough to turn on LEDs as indicated in the datasheet of the SFH7072 (> 3 V).
BAT	Battery connection, a bypass capacitor of at least 1 µF is required. This will be connected to the positive terminal of the battery.
TS	Battery pack NTC monitor. This pin is connected to the center tap between IN and GND when an NTC thermistor is used to limit the current passing through the battery during charging, when the battery is too hot. Indeed, the hotter the lower the NTC resistance and the more the charging current passing through it instead of the battery. However, this security mechanism will not be used in this project to reduce the number of passive components around the BQ25125, thus reducing the size of the overall device.

Table 3.2: Description of the BQ25125 pins, second part[39].

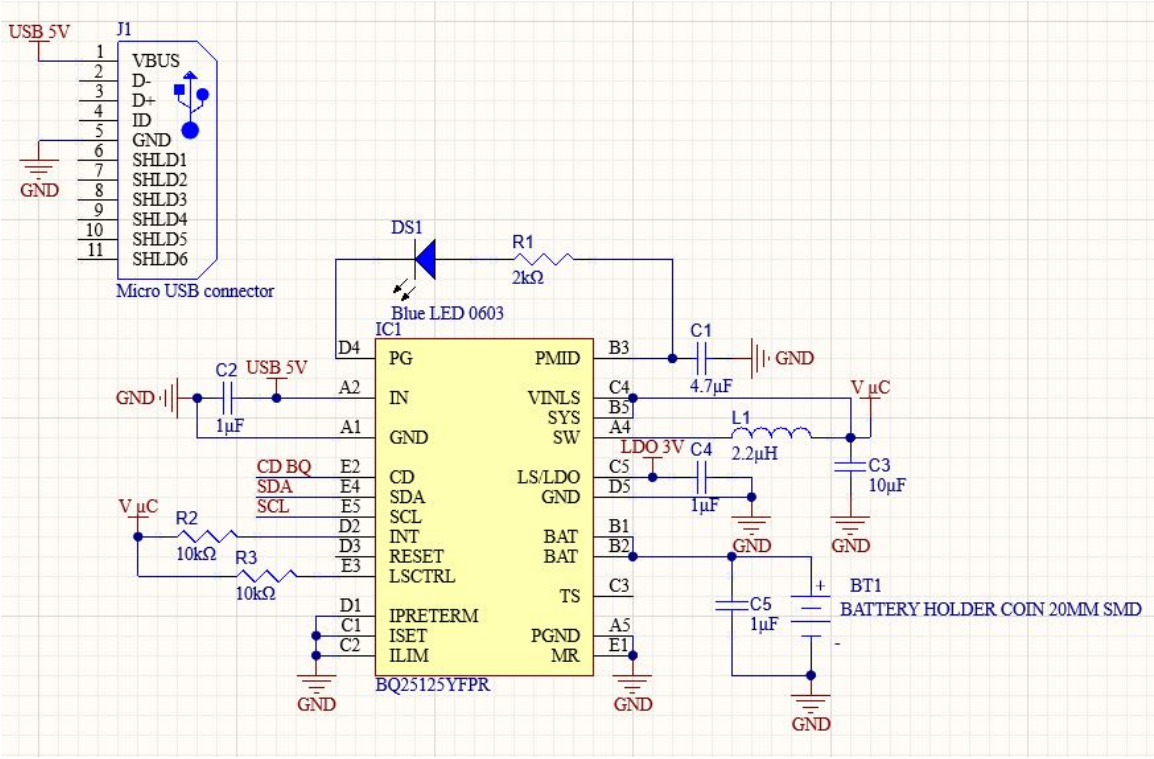


Figure 3.2: Electronic schematic of BQ25125 connections.

3.1.1.2 PCA9306 pins

After this, the circuit diagram of the PCA9306 has been created. This device consists in being connected to I2C bus lines at a certain voltage to output I2C bus lines at another voltage conveying the same information at the same speed. In our case, the PCA9306 takes as input the I2C bus lines at 1.8 V being the ones interconnecting the MCU, BQ25125, ADPD4101 and TMP1075 and transforms them into I2C bus lines at 3.3 V, being the GPIO reference voltage of the CC2640 evaluation board. To do so, the SCL1 and SDA1 pins will be connected to the 1.8 V I2C bus lines while the SCL2 and SDA2 will be the transformed bus lines at 3.3 V. VREF1 will also be connected to the stable 1.8 V power supply: $V\,\mu C$. After that, the VREF2 pin will be connected to the stable 3.3 V power supply of the evaluation board through a 200 kΩ pull-up resistor as well as the EN pin (switch enable input). Moreover, bypass capacitors are added to further stabilize DC voltages[65]. All these connections are shown in Figure 3.3.

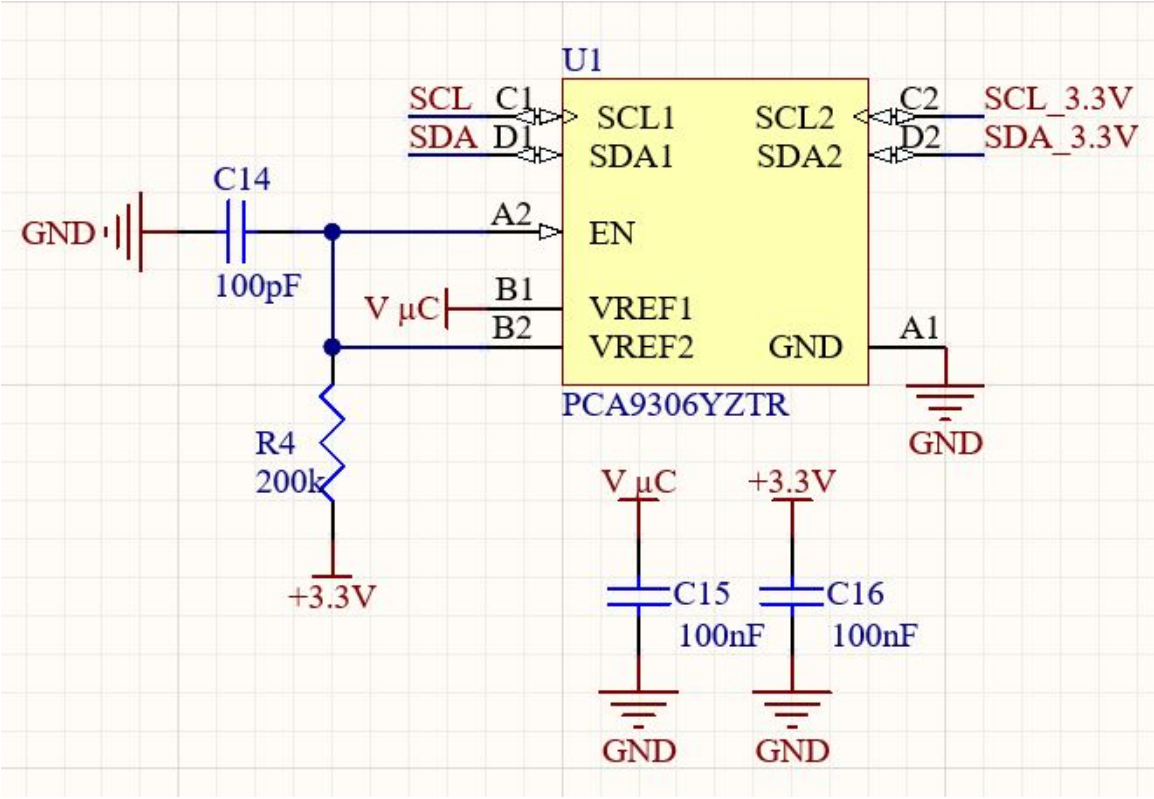


Figure 3.3: Electronic schematic of PCA9306 connections.

Finally, the complete power sheet of the electronic schematics, representing both the BQ25125 and the PCA9306 is included in Appendix A.

3.1.2 Electronic schematics: logic part

After having designed everything needed concerning the power part of the device, the microcontroller connections can be designed. In this subsection, pins of the CC2640 will be presented as well as their functionalities followed by the passive elements needed such as bypass capacitors, inductors and capacitors for impedance matching in the case of the antenna, etc. Figure 3.4 shows all the pins of the CC2640 in its smallest package while Tables 3.3 and 3.4 summarize their functionalities.

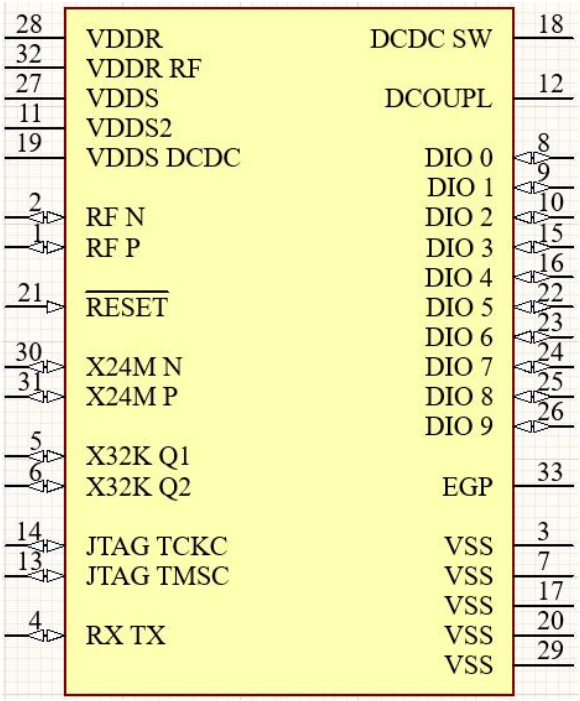


Figure 3.4: Pins of CC2640.

Pin name	Description
DCDC SW	Output from internal DC-DC. Indeed, the CC2640 has an internal DC-DC converter such that the device can be feed with any voltage in the range 1.8-3.8 V. However, since the BQ25125 offers a stabilized 1.8 V, this one will be used instead of the internal DC-DC converter of the MCU. Thus, this pin will be tied to ground.
DCOUPL	1.27 V regulated digital supply decoupling capacitor. This pin only requires to be connected to ground through a 1 µF bypass capacitor.
DIOX	Digital input output pin X. These pins are GPIO with some having a high-drive capability and others that can be turned into analog pins. As it will be seen, no analog pins will be needed in this project. Just note that DIO3 and DIO4 are JTAG TDO and TDI pins respectively that need to be connected to the programmer while implementing a code into the MCU.
JTAG TMSC	JTAG TMSC pin that needs to be connected to the programmer.
JTAG TCKC	JTAG TCKC pin that needs to be connected to the programmer.
RF N	Negative input output pin for RF communication.
RF P	Positive input output pin for RF communication.

Table 3.3: Description of the CC2640 pins, first part[33].

Pin name	Description
RF TX	Optional bias pin for RF communication.
RESET	Reset pin being active-low with no internal pull-up. This pin has to be pulled up by a 100 kΩ according to the datasheet. Note that a decoupling capacitor of 100 nF is also needed.
VDDS	Main chip supply.
VDDS2	GPIO supply.
VDDS DCDC	Internal DC-DC supply, this pin will be tied to ground since external regulator mode is used.
VDDR	1.7-1.95 V power supply, typically connected to the output of internal DC-DC converter except for external regulator mode.
VDDR RF	1.7-1.95 V power supply for RF communication, typically connected to the output of internal DC-DC converter except for external regulator mode.
VSS	Ground.
EGP	Ground, exposed ground pad.
X32K	32 kHz crystal oscillator pins, tied to ground since internal 32 kHz oscillator will be used in this project.
X24M	24 MHz crystal oscillator pins.

Table 3.4: Description of the CC2640 pins, second part[33].

As already mentioned, external regulator mode will be used since the BQ25125 provides a stable 1.8 V power supply, and this mode reduces the number of external passive components around the CC2640. Thus, DCDC SW and VDDS DCDC will be tied to ground. Concerning the GPIO pins, DIO0 is the SCL of the I2C bus lines while DIO1 is the SDA. Moreover, DIO2 will drive the chip disable pin of the BQ25125. Furthermore DIO7, 8 and 9 will be used as interrupt pins coming from the ADPD4101 and TMP1075 as it will be seen. Note that the pins RESET as well as JTAG TCKC, TMSC, TDO and TDI are used by the programmer to implement a desired code in the MCU. Thus, all these pins will be connected to pads so that external wires could be soldered later. Taking all of this into account, Figure 3.5 shows the connections and labels assigned to the CC2640 pins.

After this, 4 aspects have to be discussed: the decoupling capacitors, the passive components used for impedance matching of the antenna, the crystal oscillator connections and the pull-up resistors of the I2C bus lines.

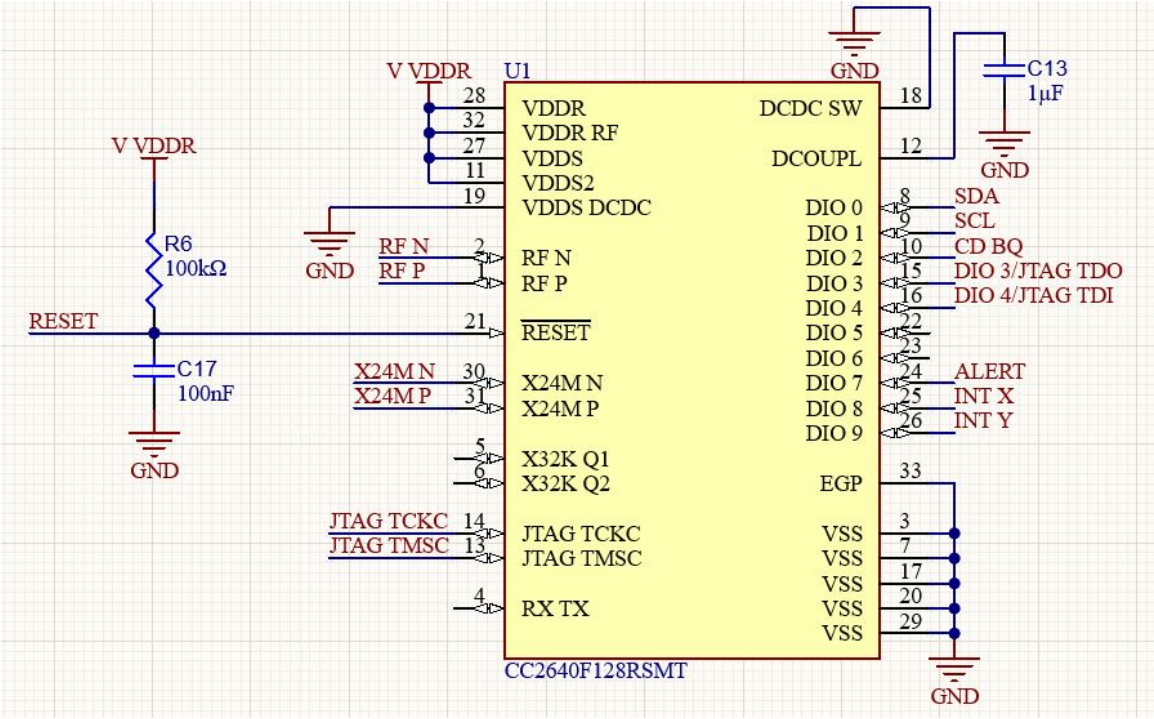


Figure 3.5: Schematic of CC2640 connections.

3.1.2.1 Decoupling capacitors

As for all power pins of any digital electronic devices, decoupling capacitors are used and placed next to pins VDDR, VDDR RF, VDDS and VDDS2. The value of these decoupling capacitors is 100 nF, as the ones used in the schematic examples in the CC2640 datasheet. Upstream of this, ferrite beads (FB) are also used to suppress high frequency noise on the DC line as well as a dedicated 2.2 µF decoupling capacitor as also mentioned in the supply voltage configurations of the datasheet[33]. A schematic of these components is represented in Figure 3.6.

3.1.2.2 RF part components

Between the RF pins of the CC2640 and the antenna (having one FEED RF pin, one ground pin and 3 unused pins for mechanical strengths), passive components such as capacitors and inductors are necessary. Before that, one has to choose the RF front-end configuration: either differential- or single-ended and either internal or external biasing. These configurations allow to have a trade-off between cost, board space and RF performances: differential operation with external bias gives

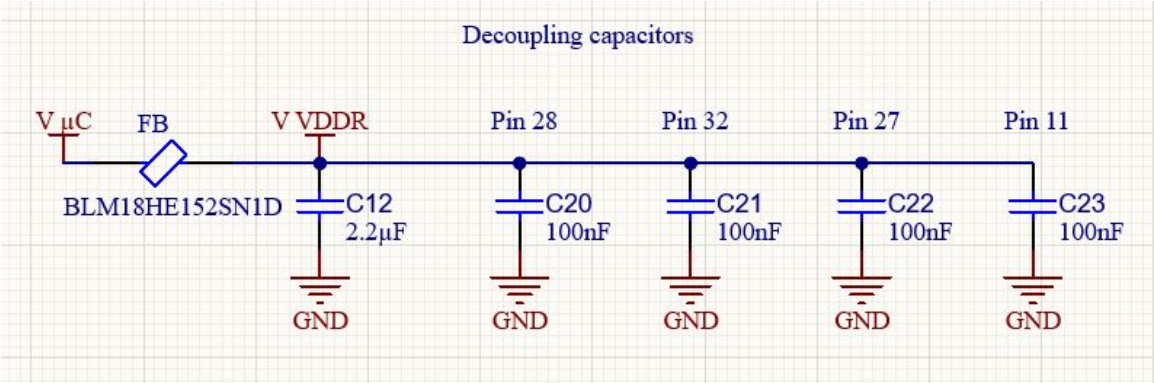


Figure 3.6: Schematic of CC2640 decoupling capacitors.

the best performances while single-ended operation with internal bias gives the least amount of external components and the lowest power consumption[33]. In this project, miniaturization is the key point while a good functioning of the device is observed. For these reasons, single-ended operation with external biasing is chosen (since external biasing only adds one inductor compared to internal biasing and improves RF performances). Thus, external passive component values can be chosen according to both the datasheet of the A5839H from Antenova[59] and the CC2640 (only for the external biasing inductor)[33]. Figure 3.7 shows the RF part of the electronic schematics.

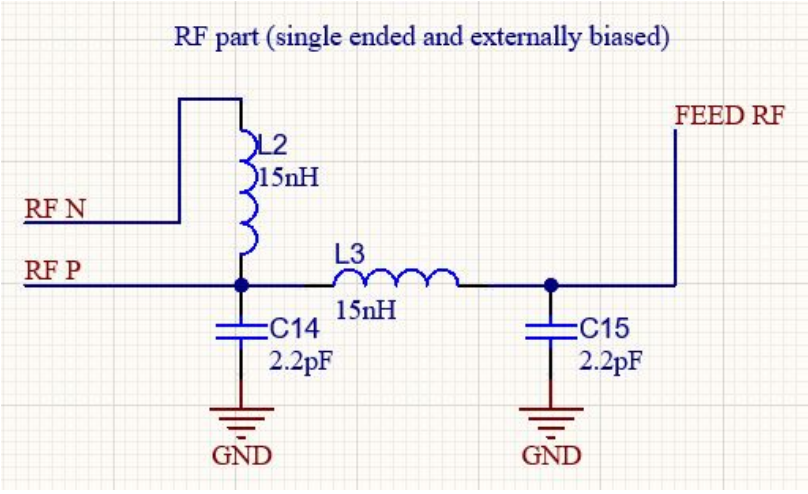


Figure 3.7: Schematic of CC2640 RF part.

3.1.2.3 24 MHz crystal oscillator

To give the main clock frequency, a 24 MHz crystal oscillator has to be connected to the CC2640. The 24 MHz crystal has 4 pins, one for X24M N, one for X24M P and two tied to ground[54]. In addition to this, optional capacitors can be connected between X24M P/N and ground to ensure that the crystal oscillator will oscillate at its fundamental and not at one of its harmonics. Such connections are represented in Figure 3.8.

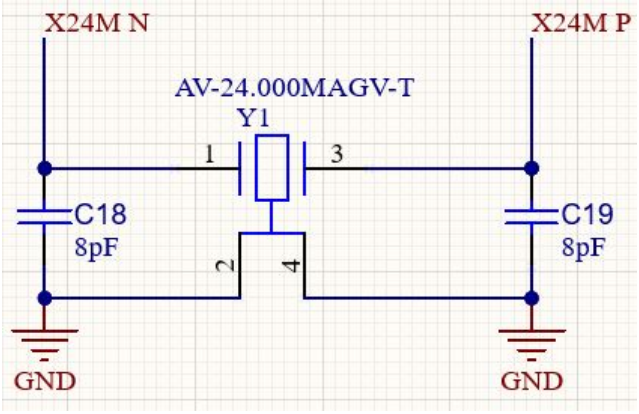


Figure 3.8: Schematic of CC2640 24MHz crystal oscillator.

3.1.2.4 Pull-up resistors of I2C bus lines

Since I2C bus lines consist in pulled-up lines to $V\,\mu C$, pull-up resistors are needed. Indeed, the master and slaves connected to it can only write low value (by pulling down the lines) as writing high value on the lines consist in doing nothing[1]. One has to choose carefully the value of the pull-up resistors to tune a trade-off between current consumption (and heating) and the time it takes to the lines to get high when master and/or slaves release their low value. Indeed, as there always exists parasitic capacitance between the bus lines and the ground, an RC time constant governs the time these capacitors charge and thus the time the lines take to go from a low value to a high one. Thus, the larger the pull-up resistor values, the longer it takes for the bus lines to go from a low to a high value and the smaller the current consumption, and vice versa. Since it would be convenient to use the fastest I2C

[1]Complete reminder of the I2C protocol will be done in the next chapter.

communication available on the CC2640, i.e., 400 kHz, pull-up resistor value range can be calculated thanks to[66]

$$R_{min} = \frac{V\,\mu C - 0.4}{3mA}$$

$$R_{max} = \frac{300ns}{C_{bus}}$$

Using $V\,\mu C = 1.8$ V, $R_{min} \approx 500\ \Omega$. However, C_{bus} is quite hard to approximate since it depends on the capacitance of the lines but also on the input capacitance of each device connected to the I2C bus. As an extremely rough approximation, $C_{bus} \approx 50$ pF meaning that $R_{max} \approx 6$ kΩ. This leads to choosing pull-up resistors values of 5 kΩ since current consumption has to be minimized to maximize the battery life. Note that this seems very reasonable since the standard value of I2C pull-up resistors is 4.7 kΩ in nearly every standard I2C design.

Finally, the complete logic sheet of the electronic schematics, including all the logic part is included in Appendix B.

3.1.3 Electronic schematics: sensor part

The last sheet of the electronic schematics concerns the sensor and thus the TMP1075, the ADPD4101 and SFH7072 bundle. As a reminder, body temperature will be measured thanks to the TMP1075, the PPG thanks to both the SFH7072 and ADPD4101 and ECG thanks to the ADPD4101. Moreover, data will be sent through I2C to the MCU since all devices are I2C compatible.

3.1.3.1 TMP1075 pins

The TMP1075 is a very small device that measures periodically (or when ordered by the MCU) the temperature and stores the result in a register accessible through I2C. Figure 3.9 shows all the pins of the TMP1075 while Table 3.5 summarizes their functionalities.

Such connections had been realized and it has been chosen that A0 will be tied to V+ and A1/A2 to ground so that the I2C slave address is 1001001, not corresponding to any other slave address on the bus. Thus, conflicts are avoided. Complete schematics of the TMP1075 connections are represented in Figure 3.10.

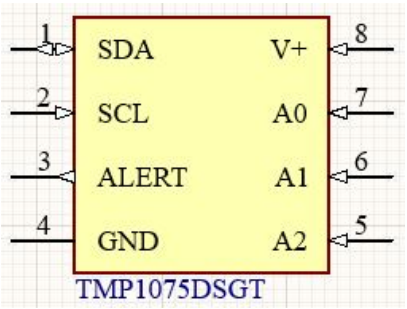


Figure 3.9: Pins of TMP1075.

Pin name	Description
SDA/SCL	Serial data and serial clock for I2C communications.
ALERT	Over temperature alert interrupt pin, active-low. This pin requires a pull-up resistor.
AX	Address select pin, can be connected to either GND, V+, SDA or SCL and different combinations give different I2C slave addresses.
V+	Supply voltage, from 1.7 to 5.5 V. A bypass capacitor of 10 nF is required.

Table 3.5: Description of the TMP1075 pins[49].

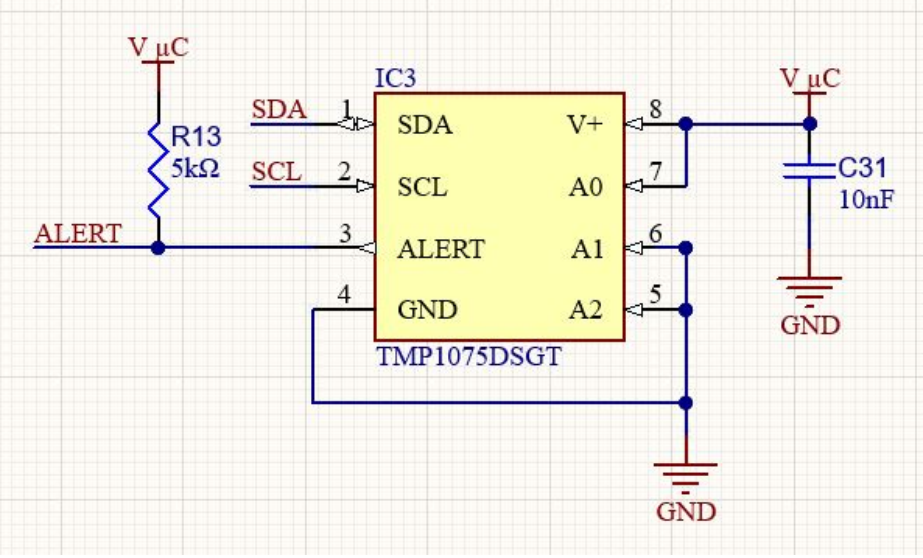


Figure 3.10: Schematic of TMP1075 connections.

3.1.3.2 ADPD4101 pins

The ADPD4101 is a multimodal front end that can stimulate up to 8 LEDs and measuring return signal (either from PPG or ECG) on up to 8 separate current inputs. To do so, a sampling period of the ADPD4101 consists in 12 time slots in which separate signal can be measured. After that, measurements are stacked in a FIFO vector that can be read through I2C. In this project, 5 time slots will be used: measuring PPG signal when either green, red or infrared LEDs are turned on and measuring ECG signal as well as lead off detection (a signal that detects if electrodes are properly stick to the skin or not).

A schematic of the ADPD4101 pins is represented in Figure 3.11 as well as a complete description of the pin functionalities in Table 3.6.

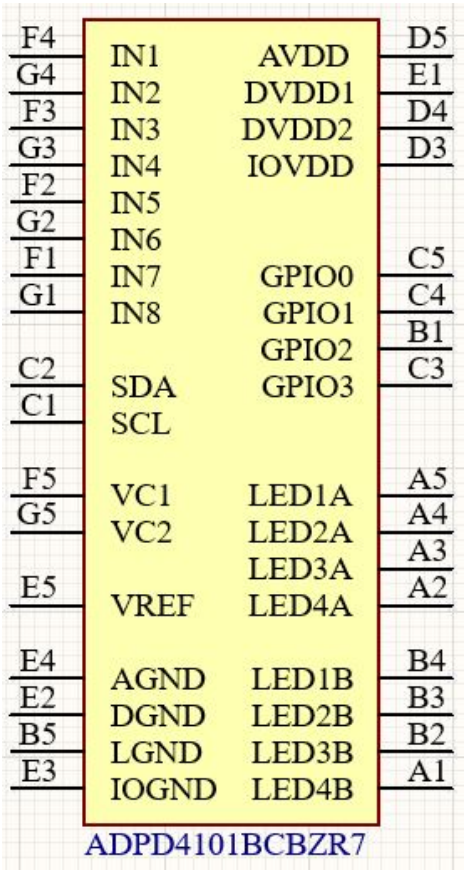


Figure 3.11: Pins of ADPD4101.

Pin name	Description
LEDxA/B	LED driver xA/B current sink, leave this pin floating if not used. This has to be connected to the cathode of the LED of interest.
LGND	LED driver ground.
GPIOx	GPIO x, used for interrupts and various clocking options.
SDA/SCL	Serial data and serial clock for I2C communications.
AVDD	1.8 V analog power supply, a bypass capacitor of 1 µF is required.
DVDDx	1.8 V digital power supply, a bypass capacitor of 100 nF is required.
IOVDD	1.8 - 3.3 V IO driver power supply, a bypass capacitor of 100 nF is required.
VREF	Internally generated ADV voltage reference, buffer this pin with a 1 µF capacitor to ground.
AGND	Analog ground.
IOGND	IO driver ground.
DGND	Digital ground.
VCx	Output voltage source x for photodiode common cathode bias or other sensor stimulus. In case of use of photodiodes, this pin has to be connected to the cathode of the photodiode.
INx	Current input x, leave this pin floating if not used.

Table 3.6: Description of the ADPD4101 pins[44].

In this project, PPG signal with the SFH7072 (2 PDs and 4 LEDs) and ECG signal with lead off detection will be measured with the ADPD4101.

PPG signal measurement To measure the PPG signal, LED1A and LED2A will be connected to the green LED cathodes and LED3B and LED4B will be connected respectively to the red and infrared LED cathodes. Moreover, the *LDO 3V* of the BQ25125 will feed the LEDs by connecting this power supply to the anodes of the 4 LEDs as already mentioned. After that, the VC1 pin will be connected to both PD anodes (one broadband and one infrared). Finally, the anode of both PDs will be connected to IN1 to collect the current of the PDs carrying the useful signal.

ECG signal measurement To measure the ECG signal, the electrodes are not directly connected to a pair of current inputs [2], a low-pass filter is required to minimize the impact of high frequency parasitic signals. The cut off frequency of this

[2]The 8 current inputs work either as a pair either not, 4 current input pairs are available: IN1-IN2, IN3-IN4, IN5-IN6, IN7-IN8.

filter is about 4 kHz, as mentioned in the ADPD4101 datasheet. Concerning the lead off detection, two methods are available: 3 electrodes or 2 electrodes. Three electrodes method has the advantage to detect which one of the two electrodes is not correctly connected but requires more passive components around ADPD4101. For this last reason, two electrodes method will be chosen to keep a minimum board space for the ECG. The functioning of the lead off detection is the following: current is injected through the two electrodes thanks to VC2 and this current is collected by one current input: IN3. Passive components related to lead off detection have been chosen in accordance with the ADPD4101 datasheet.

Finally, the ADPD4101 supports 2 different interrupts on 2 GPIO pins. These interrupts are named INTX and INTY and can be triggered on many events, chosen by programing the ADPD4101 through I2C. Details about this will be given in the next chapter. INTX is chosen to be on GPIO1 and INTY on GPIO2. A complete schematic of the connections described above is represented in Figure 3.12.

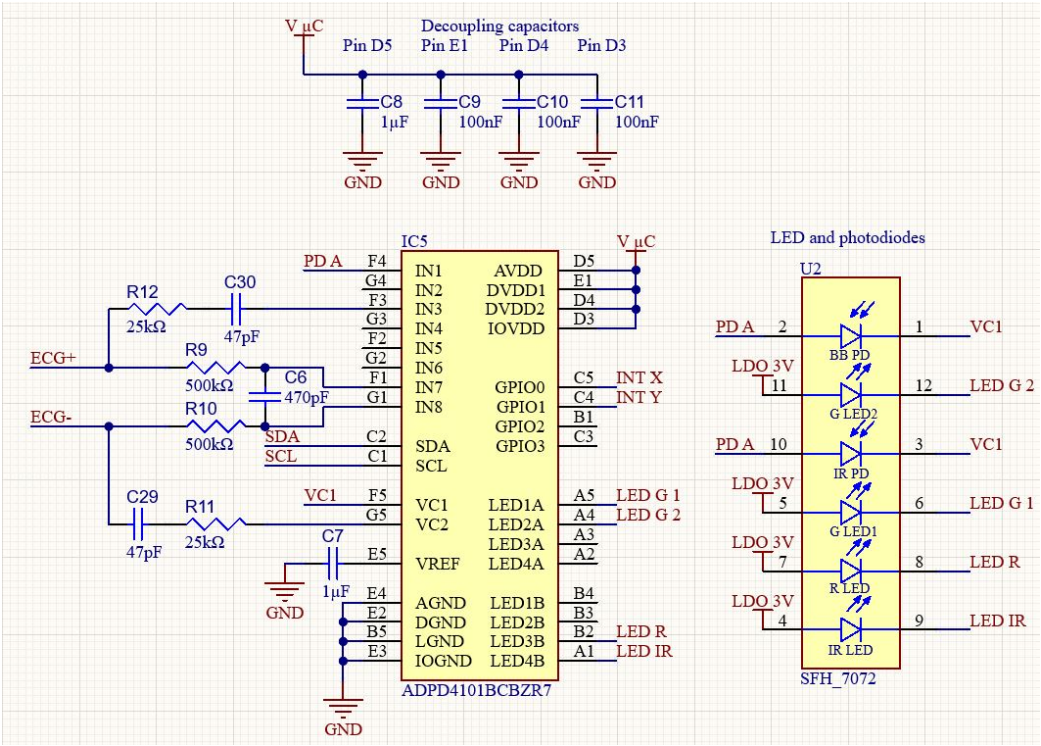


Figure 3.12: Schematic of ADPD4101 and SFH7072 connections.

Finally, the complete sensor sheet of the electronic schematics, including all the sensor part is included in Appendix C.

3.2 Printed circuit board design

After having realized the electronic schematics representing the connections between all electronic devices and passive components, the practical implementation of the circuit on a PCB has to be realized. Designing a PCB consists in placing the footprints of the different electronic devices and passive components and connecting them correctly (following what has been done for the electronic schematics) by tracks and vias. The tracks are made of conductive metal and correspond to the wires of the circuits while vias consist in holes filled with conductive metal so that different layers of the PCB can be connected to each other. In this project, a 2-layer PCB (thus no hidden layer) will be made since there is no need of having more layers than 2 in view of the small number of components and given that adding layers enlarge the PCB. In the following, since describing a complete design of a PCB is not suited for a report, I will first give some rules and dimensions that were fixed before designing the PCB. Then, I will only focus on main points of the PCB designs instead of describing everything as it has been done previously. Note that PCBs will be constructed by both *JLCPCB* and *EuroCircuit*.

3.2.1 Some design rules

Before starting the actual PCB design, some rules about the dimensions of key elements of the PCB have to be set. Here is a list of these rules:

- Track widths: a default track width has been set to 100 µm, since this is the smallest track width that can be used to again minimize the overall size of the board. However, the smaller the width of the tracks, the higher their resistance and the higher the heat produced by Joule effect. Thus, for the power lines such as V μC and the RF lines, width of up to 500 µm will be tolerated. Indeed, Joule effect is undesirable since the heat can propagate through the board and reach the temperature sensor, inducing a shift in the measurements.

- Vias size: default hole diameters and diameters have been set to 200 µm and 400 µm, since this is the smallest vias size possible. This is done in order to further minimize the size of the PCB.

- Most of the components will be placed on the top layer of the PCB such as the MCU, the battery holder, the micro USB connector, the BQ25125, the ADPD4101, ... while some other components are required to be on the bottom layer: the SFH7072 and the TMP1075. Indeed, these components need a close contact to the skin, and the bottom layer will be placed towards the skin.

- The size of the passive components such as resistors, capacitors and inductors is 0402 imperial (being 1005 metric). These means that the components will be 1 mm long and 0.5 mm large. This size is rather small (imperial sizes can go up to 6332 being 2512 metric) but is not the smallest, being 01005 imperial or 0402 metric (0.4 mm x 0.2 mm). Indeed, smaller components are harder to solder and 0402 imperial components are small enough to not take too much place on the board as it will be seen. Note that an exception was made for the LED informative diode of the BQ25125 being 0603 imperial (1608 metric).

- ECG electrode connections will be made through multi-layer pads and programmer connections will be made through pads located on the bottom of the board, of diameter roughly 1.5mm. Moreover, pads for $V\,\mu C$, ground and I2C bus lines (SDA and SCL) were added for debugging.

- The ground voltage will be propagated through the board thanks to ground planes, instead of ground lines. Indeed ground planes are preferred due to their easy use and lower ground impedance, thus limiting voltage differences between several ground connections. Moreover, via stitching will be used to reduce ground loops.

- Roughly, the power part of the system will be placed on the upper and middle left part of the board while logic part will be placed on the lower left part as well as the RF part. Finally, the sensor part will be placed on the middle and lower right part of the board. Note that this has been chosen arbitrarily and other placements could have been decided.

3.2.2 RF design

The first tricky part in the PCB design is the RF one. Indeed, choosing adequate placement of the antenna as well as the passive components (L2, L3, C14, C15) and track widths are crucial for good RF performances. Thus, guidelines given in the datasheet of the antenna were followed[59]:

- The transmission lines should be the smallest.
- The matching network (L2, L3, C14, C15) should be the closest of the antenna.
- It is strongly recommended to place at the edge of the board with the feed point of the antenna as close to the same corner of the PCB as possible.
- Ground planes and other components should be of at least 2 mm (D) of the antenna. Moreover, the antenna should be clear of ground from both sides and from the bottom.
- Placement of the matching network also follows what is schematized in the datasheet, represented in Figure 3.13.

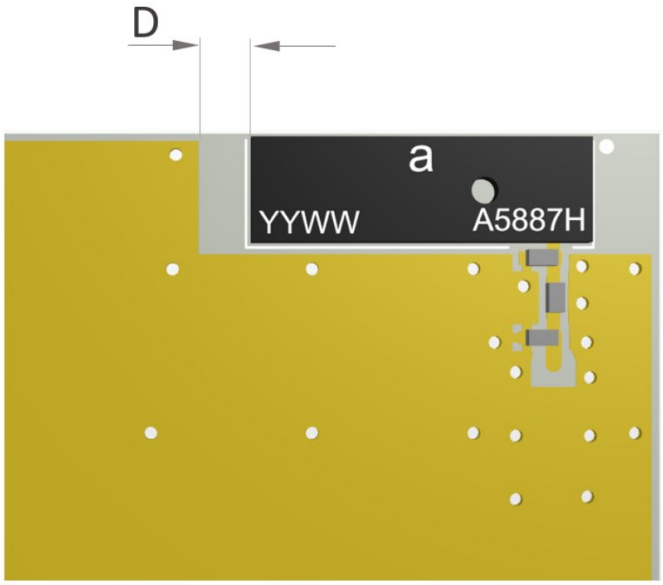


Figure 3.13: Reference placement of the antenna and the matching network[59].

Following these guidelines, PCB design (both top layer schematic in red and 3D model) of the RF part for this project is represented in Figure 3.14. Just note that no 3D model for the antenna was available on *Altium*. Thus, only footprints are represented.

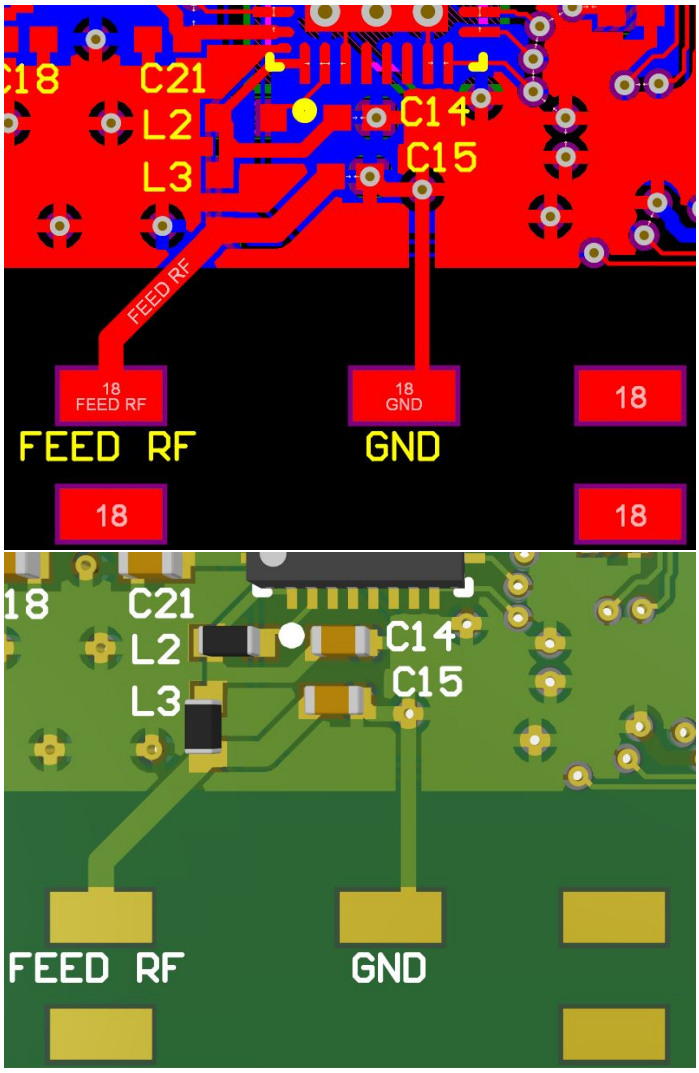


Figure 3.14: PCB design of the antenna and the matching network. Schematic on top and 3D model on bottom.

3.2.3 Temperature sensor

The second tricky part of the PCB design is the temperature sensor. Indeed, as already mentioned, the placement of the temperature sensor in wearIT4Health was sub-optimal and a shift of 5°C in the measurements could be observed due to heat diffusion from the WiFi part. Indeed, the non-conductive substrate materials (often FR-4) of the PCB board conducts relatively well heat compared to air. Thus, so as not to repeat the errors that have been made on the original wearIT4Health board, cut-outs around the temperature sensor have been realized to insulate it in terms of heat from other heat sources such as the RF part, the MCU or the power part. The insulation is not perfect but the improvement is adequate. Indeed, the thermal conductivity of air is around 0.024 W/mK[67] while the one of the FR-4 is around 1 W/mK[68] (worse coefficients are observed with alumina, aluminum nitride and beryllium oxide). Using this technique, PCB design (both bottom layer schematic in blue and 3D model) of the temperature sensor is represented in Figure 3.15.

3.2.4 Complete PCB design

The other electronic components were routed by taking established precautions such as not placing critical analog sensor lines (like ECG ones) close to parasitic lines such as the I2C bus lines or power lines. The board measures 44.5 mm x 27 mm ($\approx$ 1200 mm^2) being around 3 times smaller than wearIT4Health one, which measures 42 mm x 73 mm ($\approx$ 3100 mm^2). Thus, this proves that smaller devices than wearIT4Health while keeping nearly the same functionalities is possible. 3D models of the top and bottom views are represented in Figure 3.16[3].

Finally, pictures of top and bottom views of the real PCB with components soldered are represented in both Figures 3.17 and 3.18 respectively. Note that the top view picture had been realized with a 2€ coin next to the PCB for visual size comparison.

[3]Note that the PCA9306 is not present in this design since the idea of such device comes up on the second PCB design as it will be seen

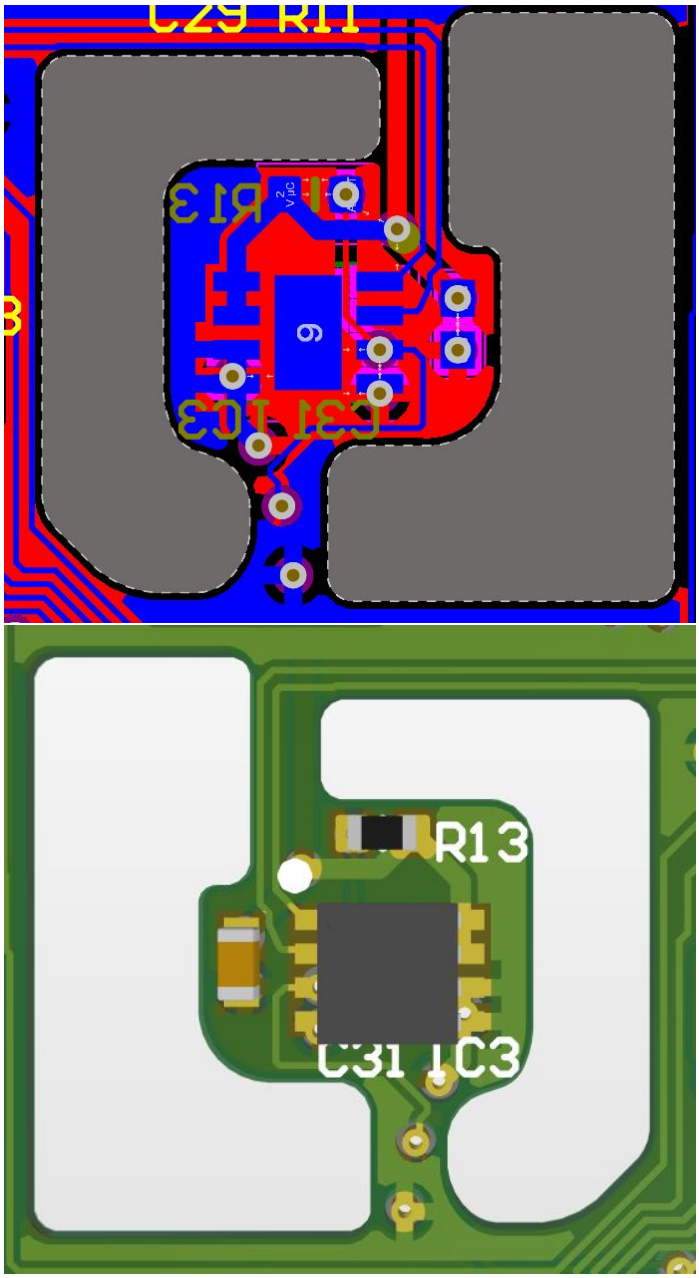
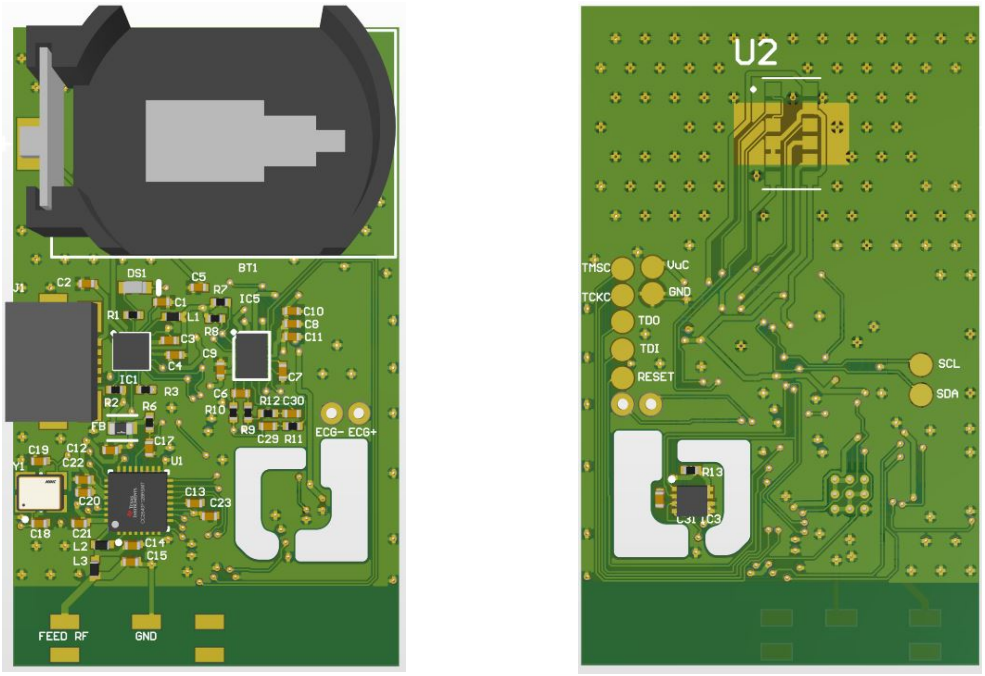


Figure 3.15: PCB design of the temperature sensor. Schematic on top and 3D model on bottom.



(a) Top view (b) Bottom view

Figure 3.16: 3D model of the PCB design.

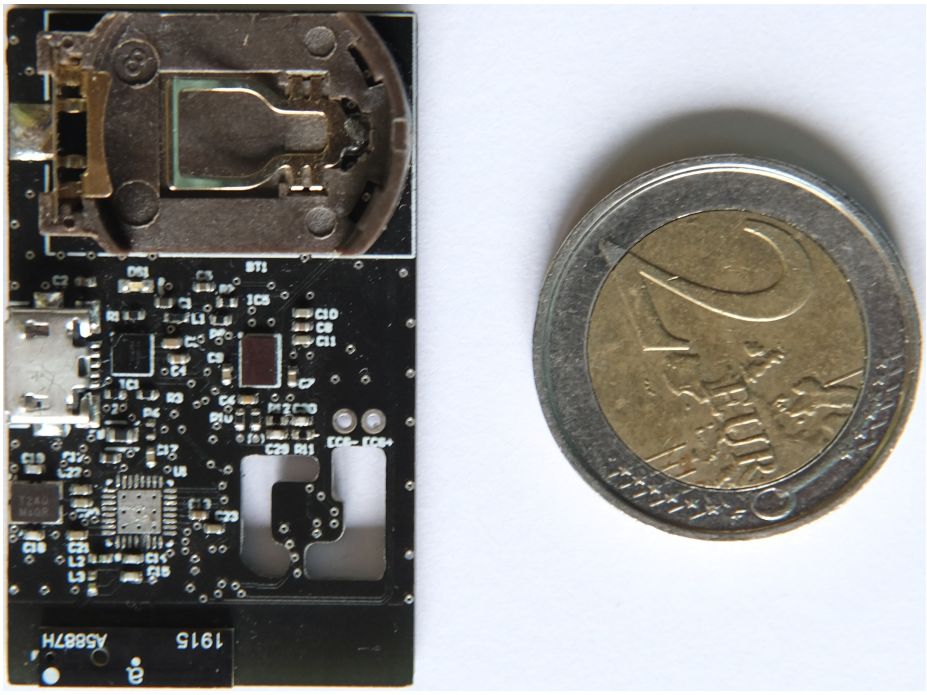


Figure 3.17: Top view of the real PCB design.

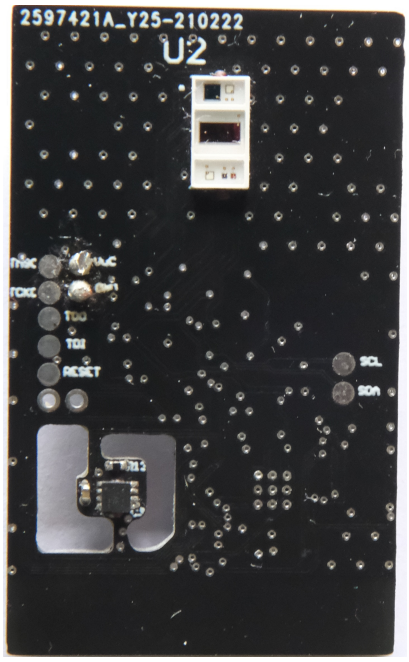


Figure 3.18: Bottom view of the real PCB design.

3.3 Update in the design

Due to COVID crisis and because of a worldwide shortage in IC availabilities, the chosen CC2640 never arrived since it was out of stock worldwide. Indeed, as can be seen in Figure 3.17, no MCU has been soldered on the PCB. At the moment we were doing the components order to Mouser and Farnell, Mouser informed us that the model we chose for the MCU was out of stock worldwide and sent us an "equivalent". However, they did not specify that the footprint of the MCU was different: we ordered the 4 mm x 4 mm and they sent us the 5 mm x 5 mm (more precisely, the CC2640R2LRHBR[69]). Thus, the received MCU could not be soldered on the PCB. The only solution to this problem was to redo the electronic schematics and the PCB design with the 5 mm x 5 mm MCU footprint, and to fabricate the new PCB. I also needed to reorder the components that have been soldered since I did not have spares. Thus, with the ordering delays (being even longer due to COVID crisis), I waited more than one month for my device and waiting them to be soldered on the new PCB[4].

[4]During this time, I was focusing on MCU and app codes but no testing was possible.

This section will present some updates that have been made in both the electronic schematics and in the PCB design since no more errors were possible due to the short time it was left to me to get the device working.

3.3.1 Electronic schematics update

First of all, the MCU footprint was changed. The only change was that the number of GPIO pins was increased to 15 even if this was not necessary. Moreover, JTAG TDO and TDI pins were moved on DIO5 and DIO6 respectively[33]. Furthermore, a pin indicating if a charger is connected to the device or not was added. This was realized by a voltage divider between the *USB 5V* tag and ground to bring back this voltage to 1.8 V, being the GPIO voltage.

Then, a second change was about the RF part. Indeed, Hervé Pierre and I were not confident about my first RF design due to its very poor number of external components (even if it was what is in the antenna datasheet). Thus, since no more PCB design was possible in view of the time passing, he advised me to copy the RF part that was on the evaluation board to be sure that it will work, since tests with the evaluation board showed very good performances[63]. The mode used in the evaluation is differential-ended with internal bias. This means that the number of passive components in the matching network will increase as well as power consumption but with better performances than before. Moreover, a U.FL connector was added in order to add the ability to connect an external antenna. Electronic schematics of this new RF part is represented in Figure 3.19.

The complete new logic sheet of the electronic schematics, including all the logic part is included in Appendix D

3.3.2 PCB design update

As expected, changes along with the changes cited previously also occured in the PCB (in additions to some improvements):

- the footprint of the MCU was increased to 5 mm x 5 mm;

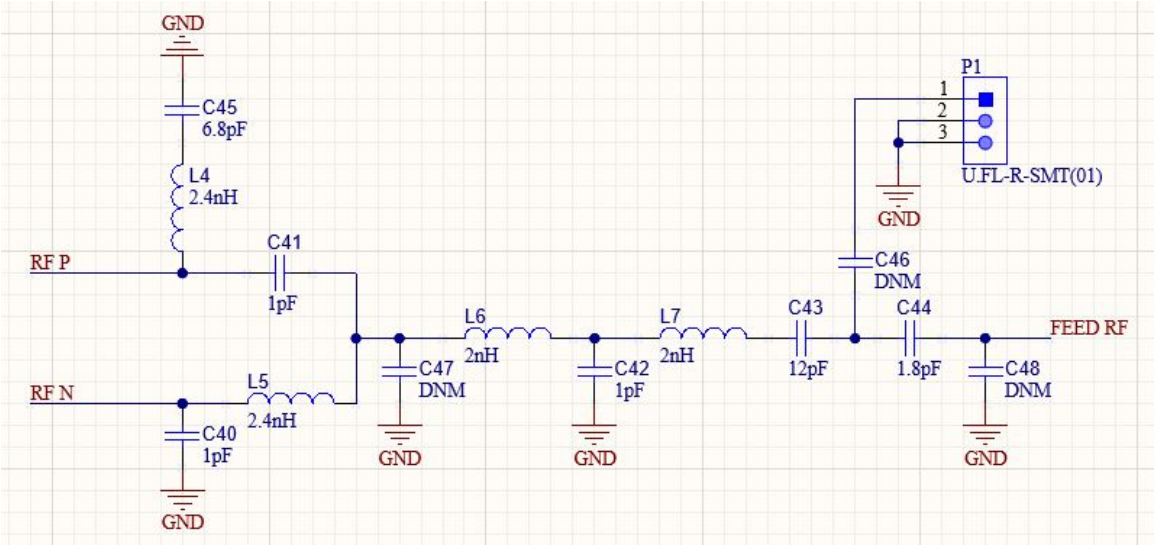


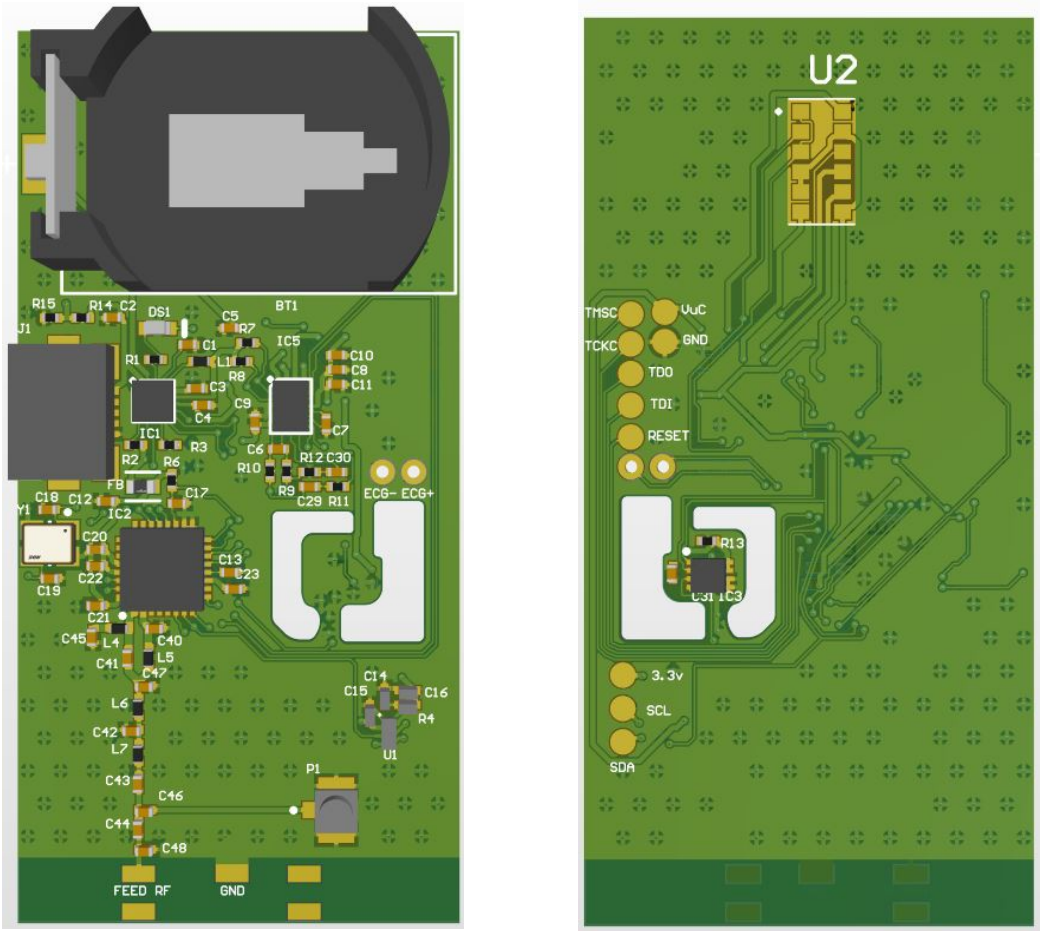
Figure 3.19: New schematic of CC2640's RF part.

- the design of the RF part is now a copy of the one in the evaluation board;
- the U.FL connector was added;
- the PCA9306 was included in the PCB design since the idea of this device comes up;
- the vias were all tented to avoid any short-circuit while soldering wires on pads;
- the solder mask of the SFH7072 was turned of 90° since it was inverted in the last design.

The only disadvantage of this new design is that the PCB is longer due to the RF part (54.4 mm x 27 mm $\approx$ 1450 mm^2).

3D models of the top and bottom views are represented in Figure 3.20.

Finally, pictures of top and bottom views of the real PCB with components soldered are represented in both Figures 3.21 and 3.22 respectively. Note that the top view picture had been realized with a 2€ coin next to the PCB for visual size comparison. Note that the absence of the BQ25125 and the SFH7072 bundle will be explained later.



(a) Top view (b) Bottom view

Figure 3.20: 3D model of the second PCB design.

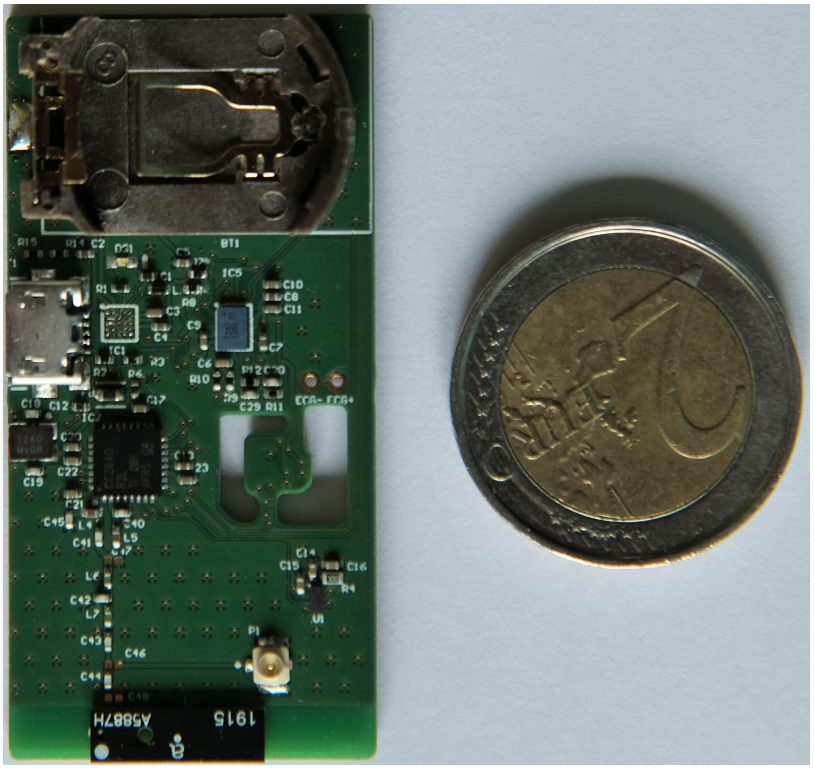


Figure 3.21: Top view of the real second PCB design.

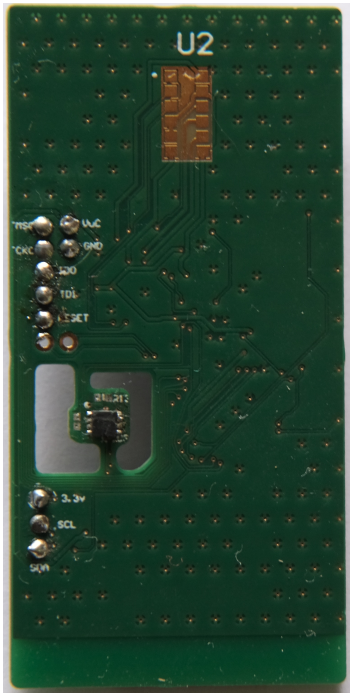


Figure 3.22: Bottom view of the real second PCB design.

Chapter 4

Microcontroller: embedded software

In this chapter, the software embedded in the MCU will be described. More precisely, the MCU will be responsible of, first, initializing the other electronic devices, then collecting data from the sensors and the BQ25125 through I2C to finally send sensor data wirelessly through BLE. Indeed, no signal processing is realized by the MCU since it is convenient to only assign it to the data transmission task in order to minimize the power consumption, since the more tasks it has to realize, the higher the power consumption and thus the shorter the battery life.

Thus, the architecture of the code, in C language, will be presented first. Then, initialization and functioning of the peripheral electronic devices will be described as well as how the I2C communication is handled thanks to the *Texas Instruments* drivers. Finally, implementation of the BLE communication will also be discussed. This has been realized thanks to the *Code Composer Studio* IDE.

Before that, a description of the I2C protocol as well as the BLE one is needed to understand the codes.

4.1 I2C protocol

The I2C protocol is a synchronous, multi-master, multi-slave, packet switched, single-ended, serial communication bus widely used for attaching lower-speed peripheral integrated circuits to processors and MCU in short distance, intra-board communication[70]. Basically, it sets the rules when a data communication is needed between the MCU and other electronic devices (in this case either the

BQ25125, the ADPD4101 or the TMP1075). As already said, the hardware of the I2C protocol consists in 2 pulled-up lines, SDA (serial data) and SCL (serial clock), that devices connected to the bus can only pull down. This is done to avoid electronic short circuit: for instance, if a device wants to write a low value while another wants to write a high value, a low value will be read on the line and the device that wanted to write a high value (which just let the voltage line high) knows it is not alone on the bus and becomes silent until the transaction is over. The following description will be focused on the software part since the bus has been already designed on the PCB. Just note that the I2C protocol has been chosen in this project since it only requires two lines, thus reducing PCB size.

Each I2C communication consists in a master (here the MCU) wanting to communicate with a slave (here either the ADPD4101, the BQ25125 or the TMP1075). The master controls the SCL line by alternating periodically high and low values at typically 100 kHz or 400 kHz (fast mode, that will be used in this project). Concerning the SDA line, data will be transferred from the sender to the receiver in the same way: 8 bits of data are transferred by the sender and the receiver acknowledges the reception by putting a low value on the SDA line as a "ninth" bit. Each bit value is characterized by the value of SDA when the SCL is high (thus the value of SDA has to be maintained constant during a high value of SCL). Before transmitting data, a start condition initiates the I2C communication and is characterized by a high to low transition on SDA while SCL is high. To stop the communication, a low to high transition on SDA while SCL is high has to be observed. Note that only the master can create start and stop conditions.

The first data transmission is different and is always sent by the master and consists in addressing the right slave. Indeed, the slave of the communication has to be identified since several devices can be connected to the bus. To do so, each slave can be addressed by a 7-bit address and a read/write bit. Indeed, to address a specific slave, the first data transmission consists in the master writing the corresponding 7-bit address followed by a 0 if the master wants to write something into the slave (thus the master is the sender and the slave is the receiver) or a 1 if the master wants to read something from the slave, always followed by an acknowledgment of the slave. In the following, no evidence of the acknowledge will

be cited again but note that, if the receiver does not acknowledge the 8-bit packet, either the communication is restarted either the communication is aborted.

If the master is in writing mode, it will write on the SDA line the address of the slave register (can be 8-bit or 16-bit) it wants to write, followed by the value it wants to write in this specified register (either 8-bit or 16-bit). If the master is in reading mode, it also writes the address of the slave register he wants to read and the slave writes the value of the specified register just after. If another register has to be written or read, the master can use a restart condition, similar to the start condition, restarting a new I2C communication without using a stop condition.

In addition to that, some additional features are present in the I2C protocol. For instance, if the master is going too fast for the slave (the frequency of SCL is too high) the slave can stop the transaction momentarily by keeping SCL low so that it has the time to be ready for the rest of the transmission, this is called clock stretching. Another example is the arbitration when several masters are connected to the bus. This will not be described as only one master is present in this project. Example of an I2C communication can be observed in Figure 4.1.

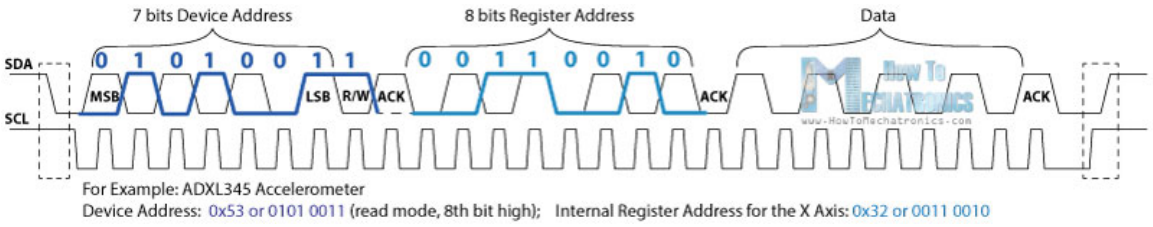


Figure 4.1: Example of an I2C communication in read mode[71].

4.2 Bluetooth Low Energy protocol

BLE is a synchronous radio frequency protocol consisting in a wireless personal area network technology aimed at novel applications in the healthcare, fitness, beacons, security, and home entertainment industries[31]. The advantage of such communication protocol compared to Classic Bluetooth (CB) is the strongly reduced power consumption and cost while maintaining similar communication ranges and data rates. Indeed, CB offers data rate ranging from 0.7-2.1 Mbit/s

while BLE offers 125 kbit/s - 2 Mbit/s data rates. Moreover, the power consumption and peak current consumption of CB are respectively 1 W and 30 mA while BLE offers minimized power consumption ranging from 0.01 W to 0.5 W and peak current consumption of 15 mA[31].

The 3 main markets intended for BLE, identified by Bluetooth Special Interest Group, are[31]:

- low power requirements, operating for "months or years" on a button cell;
- small size and low cost;
- compatibility with a large installed base of mobile phones, tablets and computers.

Thus, this wireless protocol makes the perfect candidate for this project.

Concerning the radio interface, the frequency range in which BLE works is 2.400–2.4835 GHz, being the same as CB. The only difference between CB and BLE in terms of frequency range is the use of different channels (being small frequency ranges in the total reserved frequency band). Within a channel, the modulation used is the Gaussian frequency-shift keying, being a frequency modulation technique in which a binary signal is filtered by a Gaussian filter to smooth the signal so that spectral bandwidth is reduced.

Before starting a communication, a pairing between the BLE device and another device (here that will be a smartphone) has to be done. To do so, the BLE device will advertise by broadcasting advertising data packets (that contain information such as the name of the device) on 3 different channels reserved for advertising. These packets are sent periodically, the time interval between 2 advertising packets is called the advertising interval. On the other side, the smartphone, called the observer, listens to these channels during a certain time interval, called the scan window, in order to detect BLE devices around it. Once the observer has found the BLE device he wants to connect to, they can pair so that a BLE connection is set and communications can occur[31].

A BLE connection consists in a link between two (or more) devices over time. Since BLE is a synchronous radio frequency protocol, any data transmission has to be scheduled. Thus, a BLE connection can be seen as a series of meetings where two devices transmit and receive information at the same time, on the same radio frequency. In order for both devices to be present at each meeting, they have to decide at each meeting what will be the frequency on which they will communicate on the next meeting but also in how much time[72]. This can be seen as fixing a date (the time interval between two meetings) and a location (the frequency band, i.e., the channel) in order to meet someone. Note also that several mechanisms such as frequency hopping (being fast changes in the carrier frequency) are implemented.

After having described some fundamentals of the BLE protocol, one can be interested in the data structure which is used in this protocol. From an application point of view, data are exchanged using the Generic Attribute Protocol (GATT). Hierarchically, there exist services being a collection of characteristics containing attributes[73].

The attribute is the smallest addressable unit of data used by GATT. The address of this attribute is contained in the handle being 16 bits long while the universally unique identifier (UUID) describes the type of the attribute, being 16, 32 or 128 bits long. The data is contained in the value field and consists in a byte array that can be as long as 512 bytes. In practice, 20 bytes data field are used at most[1]. When accessing to the data, three main modes exist: either read, write or notify. The 2 first modes consist in reading or writing the data field while the notification mode consists in a reading operation each time the data field is changed. This last mode can be useful when the smartphone has to be notified each time the data field changes.

The characteristic contains several attributes and consists of at least one declaration attribute and one value attribute. The declaration attribute always comes before the value attribute it declares and contains several information: the modes (read, write or notify) that are available for the value attribute, the UUID of the

[1]For instance, Android limits the data field to 20 bytes.

characteristic and the handle of the characteristic value attribute. To make things clearer, Table 4.1 represents one declaration attribute and one value attribute of a characteristic.

Handle	UUID	Value	What
31	0x2803	02:20:00:AA:BB	Characteristic declaration
32	0xBBAA	41:42:43	Characteristic value

Table 4.1: Example of one declaration attribute and one value attribute of a characteristic[73].

On the first line of Table 4.1, the handle is the address of the characteristic declaration while the UUID 0x2803 describes the type of the attribute, being a characteristic declaration. The value of this declaration gives information about the associated characteristic value: 02 (called the char value permissions) means that the value is in reading mode only, 20:00 = 0x0020 = 32 (called the char value GATT handle) is the handle of the associated characteristic value and AA:BB = 0xBBAA (called the characteristic UUID) is its UUID. On the second line, the actual value of the data field can be read as 41:42:43, being the useful data for the user.

The service is a collection of characteristics. This is what the peer device will first discover once the pairing process is over. For instance, heart rate, battery, body temperature and pulse rate oximeter services will be used in this project. Table 4.2 extends Table 4.1 such that the service declaration is included.

Handle	UUID	Value
30	0x2800	AA:AA
31	0x2803	02:20:00:AA:BB
32	0xBBAA	41:42:43

Table 4.2: Example of a BLE service[73].

As can be seen in Table 4.2, the UUID 0x2800 is reserved for primary service declarations. On top of that, one can find profiles that describe what the services consist in. For instance, a profile could describe what is the service in Table 4.2: 0xAAAA stands for Acronym Service while 0xBBAA is a Random Acronym String with no null terminator that is always 3 characters (41, 42 and 43 in ASCII being

ABC) long.

The practical implementation of services can be realized by following the *Custom Profiles* training of the *SimpleLink* academy[73]. Moreover, this training offers a service generator that generates C codes and headers necessary to the declaration of a custom service. Indeed, by entering the service name and the service UUID, one can add one or several characteristics by entering their names, UUID, data field length and permission concerning the characteristic value (read, write or notify). In this project, several services have been created by respecting the UUIDs (both for the service and characteristic when existing) convention as described in the *Assigned Numbers* part of the Bluetooth official website[74]. The created service and associated characteristics, in which notify and read modes are permitted, are:

- The battery service containing the battery level characteristic;
- The ECG service containing the ECG signal and lead-off characteristics;
- The PPG service containing the PPG signals characteristic;
- The body temperature service containing the body temperature characteristic.

4.3 Code description

The code that will be used for running the MCU in this project will be strongly inspired from the *Project Zero* of *Texas Instruments*[75]. Indeed, *Project Zero* is one of the BLE example projects that consists in writing the LED state (ON/OFF), reading or getting notified of the state of 2 buttons or reading some data that has been set by the user in the MCU. This project also provides some codes to advertise on the BLE frequency bands, pair to an external device and send some data. Moreover, initialization board files are provided so that the hardware of the CC2640 evaluation board is known by the MCU as well as the drivers it will use. To construct the code project for this master thesis, changes to the *Project Zero* can be split into 2 types: changes in the initialization board files and changes in the main code so that communication with the BQ25125, ADPD4101 and TMP1075 can be implemented. Indeed, no changes are needed in the BLE stack (being the set of functions used

for a proper BLE communication) since the BLE protocol is the same for every BLE project, except the definitions of the services that have already been discussed.

4.3.1 Initialization of board files

As already mentioned, these files are responsible to inform the MCU what is the hardware around it (such as pin connections)[76] and also to inform and initialize which *Texas Instruments* drivers will be used[77]. These files are split in one C code responsible for the driver initialization and one header file responsible for defining the RF mode that is used and mapping pin names to board signal aliases[2] so that code will be clearer. Only minor changes to this last header files occurred since the RF mode is the same as the one used in the evaluation board (differential-ended with internal bias) and the number of GPIO pins is reduced from 31 (the package used in the evaluation board) to 15 (the package used in this project).

Concerning the drivers that will be used, both the I2C[78] and PIN[79] drivers have to be tuned. The I2C driver initialization consists in declaring which DIO pin is connected to the SDA line and to the SCL line. In this project, DIO0 is connected to SDA and DIO1 to SCL. Note that the I2C communication is set in blocking mode, meaning that no further code can be executed until the I2C communication is over, this in order to avoid starting a new I2C communication while the previous one is not over.

After that, the PIN driver initialization consists in constructing a PIN table that will inform the MCU of the functionalities of each GPIO pin. In this project, 5 GPIO pins are used:

- DIO2 connected to the CD pin of the BQ25125, called the CD pin;
- DIO3 connected to the potential divided between the *USB 5V* and the ground, called the CHARGE pin;
- DIO7 connected to the alert pin of the TMP1075, called the ALERT pin;
- DIO8 and DIO9 connected to the INTX and INTY of the ADPD4101 respectively, called the INTX and INTY pins.

[2]This just corresponds to comprehensive tags assigned to the right DIO pin by a define macro.

In the PIN table, only the CD pin is declared as an output obviously since other pins carry signal coming from other electronic devices. Moreover, the CD pin is initially set to 0 and is connected to a push-pull configuration inside the MCU such that the pin can be freely set to 1 or 0. The 4 other pins are declared as inputs and no pull mechanisms is present inside the MCU since the other electronic devices are responsible of pulling high or low these pins. On one hand, the CHARGE pin is not assigned to an interrupt mechanism since this pin is for reading purpose only to set the appropriate value to the CD pin. On the other hand, the ALERT, INTX, INTY pins are assigned to an interrupt mechanisms since any value transition on these pins is responsible for specific event occurring either in the TMP1075 for the ALERT pin either in the ADPD4101 for the INTX and INTY pins. Just note that interrupt is triggered on high to low transition for the ALERT pin since this one is active low while interrupts are triggered on low to high transition for the INTX and INTY pins since these are active high. Note also that pin hysteresis has been activated for these 3 pins to avoid triggering several interrupts on one transition.

4.3.2 Main code architecture

The architecture used in the *Project Zero* is a real time operating system (RTOS) combined with a waiting queue. Indeed, the RTOS is mainly used for its easy implementation of drivers, timers and BLE stack functions but only contains one unique task in which a waiting queue processes messages from interrupts triggered on a timer or a pin. This architecture is suitable for this master thesis also since there is no use of several tasks in the RTOS and the waiting queue is already implemented (such that the main difficulty of using this architecture is not relevant).

Obviously, before entering the main loop and starting the RTOS, an initialization task is necessary. This one is responsible for initializing everything needed for the BLE communication such as initialization of a default value to the BLE characteristics of all services, the waiting queue, the timers that will be used, the opening of the useful pins by the pin driver, the opening of the I2C driver, the initialization of the internal 32 kHz oscillator[3] and the initialization of all the other electronic

[3]This is required since no 32.768 kHz crystal oscillator is connected externally.

devices on the board by I2C. This last operation will be detailed in the following, after having described what the main loop does.

Basically, the main loop waits for a signal coming from either a timer timeout callback function either an interrupt callback function. There exist signals related to the BQ25125, to the ADPD4101 and to the TMP1075. In order to not mix all the information concerning these 3 devices together, functioning and initialization as well as how data communication is handled in the main loop of these 3 devices will be described separately, after a description of how an I2C communication is realized using the I2C driver.

4.3.2.1 I2C communication codes

Using the TI RTOS I2C driver[78], an I2C transaction is represented by an `I2C_Transaction` data structure. In this one, one has to specify the slave address `slaveAddress`, a writing buffer `writeBuf` containing the data that has to be written as well as a writing counter `writeCount` indicating the number of bytes that needs to be written. Moreover, a reading buffer `readBuf` containing the data that has to be read as well as a reading counter `readCount` indicating the number of bytes that needs to be read need to be specified. After having initialized the transaction, this last can be started using the function `I2C_transfer`, this last returning a Boolean value representing if the transaction executed well or not. Concerning the read-/write bit written after the I2C address of the slave, the I2C driver automatically chooses the appropriate value (either read or write mode) by looking at the write and read counters. Obviously, if the read counter equals 0, the transaction is in write mode and vice versa.

From this, one can derive the four types of transaction that will be needed in this project: writing an 8-bit register with an 8-bit address, reading an 8-bit register with an 8-bit address, writing a 16-bit register with a 16-bit address and reading a 16-bit register with a 16-bit address. Indeed, the BQ25125 has 8-bit register with an 8-bit address while the ADPD4101 has 16-bit register with either 7-bit address or 15-bit address. Note that, for the ADPD4101, only the 15-bit address mode called long read and write will be used since every 7-bit address can be translated in a

15-bit address with a header of 8 zeros. In this mode, the bit after the acknowledgment of the ADPD4101 address and R/W bit has to be set to 1 in order to inform the ADPD4101 that a long operation will be performed[44]. The functions used for these 4 transaction modes are called respectively `I2C_writeRegister`, `I2C_readRegister`, `I2C_writeLongRegister` and `I2C_readLongRegister`, and are present in the `MT.c` code that can be accessed via Appendix E.

4.3.2.2 BQ25125 software

As a reminder, the BQ25125 will be in charge of using the battery voltage to produce a stable 1.8 V and 3.3 V to feed the ICs and the LEDs respectively and of the battery charge management. Moreover, communication with the BQ25125 will be used to first initialize it to operate as we want and, second, to extract the percentage of the battery level in order to know when the charger has to be plugged in. Note that each time the BQ25125 is accessed through I2C, a suitable voltage has to be applied on its CD pin: 0 if the charger is connected and 1 if the charger is not connected. After the I2C transaction, the CD pin has to be set low.

The initialization consists in modifying some register values to tune the behavior of the BQ25125. All this information can be found in the BQ25125 datasheet[4][39]. First of all, the output SW has to be enabled and set to 1.8 V. After that, the LS/LDO output has to be enabled to 3.3 V. Concerning the charging of the battery, the termination current has been set to 2 mA and the fast charge mode has been enabled with a current of 290 mA (the maximum possible current is 300 mA but this should be avoided since it consists in pushing the BQ25125 to its limit). Moreover, the input limit current has been increased to 400 mA (being a standard when using LEDs). Note that these changes relative to the battery charging are the same that the ones used in wearIT4Health and seems to work well. Finally, the BQ25125 is informed that the TS pin is disabled since no NTC is present in the vicinity of the battery and that the INT pin is also disabled since it is not used in this project. Complete initialization code of the BQ25125 can be found in the `I2C_slave_init` function in the `MT.c` code that can be accessed via Appendix E.

[4]In order to not overload the report, only the goals of the initialization will be mentioned here. For a complete description of which value is written in which register, refer to the appendices.

The part of the main code in which the BQ25125 intervenes consists in a periodic battery level percentage reading. Indeed, a periodic clock triggering an interrupt each 30s[5] is started in the code initialization. This interrupt pushes a message in the waiting queue carrying the information that the MCU should inform the BQ25125 that a new VBATREG (the register containing the percentage of the battery level) reading has to be initiated. After this, a new clock will trigger an interrupt 5ms later its start. This clock is implemented so that the MCU can wait enough time to let the BQ25125 finishing the reading of the battery level. This one shot interrupt pushes another message to the waiting queue informing the MCU that it can read the voltage based battery monitor register containing the percentage of the battery level. Note that this percentage ranges from 60 to 100% with steps of 2% and is a comparison between the current voltage across the battery and its maximum voltage. Thus, this percentage is an indicator of the depth of discharge but should not be interpreted as the usual battery percentage being the battery capacity percentage. Table 4.3 shows one example of a relation between the voltage percentage being read by the MCU (called VBMON) and the usual battery capacity percentage.

Battery full	VBMON $>$ 90%
95% to 65% remaining capacity	VBMON = 88%
65% to 35% remaining capacity	VBMON = 86%
35% to 5% remaining capacity	VBMON = 84%
Battery empty	VBMON $<$ 82%

Table 4.3: Relation between the read battery percentage by the MCU and the usual battery capacity percentage[39].

The function `I2C_get_SoC_BQ` used to extract the VBMON value from the voltage based battery monitor register value is present in the `MT.c` code that can be accessed via Appendix E, so that data can be sent over the air using BLE.

4.3.2.3 TMP1075 software

The TMP1075 will be in charge of the body temperature reading. This device can also trigger an interrupt on the ALERT pin if the temperature measured is out

[5]It is important to access it periodically since a safety watchdog timer of 50s is implemented in the BQ25125 and reset the BQ25125 parameters if this last is not accessed through I2C at least every 50s.

of bounds that can be defined by the user. This interrupt mechanism will not be used since there is no need of a feedback system for this project, the TMP1075 will simply act as a temperature sensor and nothing else.

The initialization consists in choosing in which mode the TMP1075 will operate: either in one-shot mode either in continuous conversion mode[49]. In this first mode, the TMP1075 is in shutdown mode (very low power consumption) and starts a temperature measurement only when it is asked by the MCU (by setting a specific bit in a specific register). On the contrary, in the second mode, the device continuously performs periodic temperature measurement whose highest period is 220ms, being much larger than what is needed in this project. Obviously, for low power consumption reasons, the one-shot mode will be chosen in this project. Complete initialization code of the TMP1075 can also be found in the `I2C_slave_init` function in the `MT.c` code that can be accessed via Appendix E. Note that the TMP1075 has 16-bit register addressed by 8-bit address but the configuration register is accessed as an 8-bit register since the 8 least significant bits of this register are not used.

The part of the main code in which the TMP1075 intervenes consists in a periodic body temperature reading. The same code architecture as with the BQ25125 is used: a periodic interrupt sends a message to the main code informing the MCU it must order a new temperature conversion by the TMP1075 to finally access the temperature measurement 5ms after ordering the conversion. Temperature measurements are processed every 30s, being an adequate period since this induces low power consumption (the higher the period, the lower the power consumption) and changes in body temperature occurs at much lower frequency than the sampling one. Finally, temperature readings are accessed thanks to the function `I2C_get_TMP` present in the `MT.c` code that can be accessed via Appendix E, so that data can be sent over the air using BLE. Note that this function does not use the I2C communication functions presented earlier since the register containing the conversion results worth 16 bits and is accessed by an 8-bit address. Indeed, the temperature value is contained in the 12 most significant bits of this register.

The value read by `I2C_get_TMP` is the temperature but is not expressed in degree Celsius. Indeed, the read value has to be multiplied by 0.0625°C to obtain a temperature measurement expressed in °C. However, this operation will rather be performed by the application since the MCU does not support float operation (an ARM cortex M4 is required for such operations).

4.3.2.4 ADPD4101 software

As already mentioned, the ADPD4101 is a multimodal analog front end and will be responsible in measuring the ECG and PPG signals. To do so, a low frequency period of the ADPD4101 can be divided in at most 12 time slots each responsible for a very specific measurement. In this project, 5 time slots will be used: 3 for the PPG measurement using the green, red or infrared LED(s), 1 for the ECG signal and 1 for the ECG lead off detector. Thus, one needs to configure each time slot to operate as the user wants. After passing through the analog front end and analog to digital converter (ADC), the resulting digital signals are stored in two locations: either in a 32-bit time slot corresponding register (thus being overwritten at each new sample) or in a FIFO vector (either 8, 16, 24 or 32 bits). Since real-time plots will be drawn and the sampling frequency of PPG and ECG will be fixed to 100 Hz, the FIFO vector will not be used and data will be read each 10ms, right after the conversion of the signal in time slot number 5. Indeed, doing in this way means that a point will be added each 10ms on the real-time plots, meaning that the plot will have a refresh rate of 100 Hz, being suitable (maybe a bit fast) for such plots.

The initialization consists in configuring some global functions of the ADPD4101 like on which event INTX and INTY will trigger as well as the analog front and ADC configurations in each time slot. First of all, concerning the global configuration of the ADPD4101, the low frequency internal oscillator (responsible for controlling the sample timing, wake-up states, and overall operation) is set to 1 MHz since it is the standard for ECG and PPG measurements. After that, the ADPD4101 is informed that it will use 5 time slots and that the input pins 7 and 8 will be used as differential pair, since these pins are responsible for the ECG signal. Furthermore, INTX is mapped to GPIO0 and INTY to GPIO1 as referred in the electronic schematics. Finally, INTX will be active when the FIFO is full (even if the FIFO is not used, this permits to easily implement the use of the FIFO in another project)

and INTY will be active as soon as the conversion of the signal of the 5th time slot is over, meaning that the 5 signals being measured by the ADPD4101 are ready to be read. This INTY signal will be the one responsible for informing the MCU it has to read the current values of these 5 signals.

After that, the three first time slots, responsible of the PPG signals, are configured in the same way. The input pin IN1 is enabled and the correct LED(s) are activated and deactivated. Moreover, the mode in which the analog front end and ADC operate is the continuous connect mode consisting in a single analog integration of the incoming charge for each ADC conversion. According to the ADPD4101 datasheet[44], this mode is the most typical when measuring PPG signal in which the incoming charge corresponds to the photodiode response to the LED pulse being reflected by blood and surrounding tissues. In this mode, the analog front end consists in a transimpedance amplifier followed by a band-pass filter and an analog integrator feeding the ADC. The transimpedance amplifier is used to transform a current to a voltage that can be filtered with the band-pass filter. In addition to this, a precondition period, in which the anode voltage of the photodiode equals TIA_VREF (being the internal voltage reference signal for the transimpedance amplifier) and the cathode voltage equals TIA_VREF + 215 mV, is applied so that the photodiode capacitance and noise in the signal path are reduced thanks to the 215 mV reverse bias. Finally, it has been mentioned in the datasheet of the ADPD4101 that the signal to noise ratio can be further improved using integrator chopping[44]. Indeed, the integrating amplifier can add a low frequency parasitic signal that can be suppressed when one inverts the integration sequence every other time, i.e., integrator chopping. Note also that the width of the LED pulse (whose default value is 2µs) can be tuned even if the default value will be used in this project. This tuning is used when more ambient light rejection is necessary (by further reducing LED pulse width) for instance.

Concerning the ECG signal measurement, an external RC network is needed as represented in the electronic schematics. This network is used as a low-pass filter (500 kΩ resistors and 470 pF capacitor) to reduce the high frequency noise coming from the electrode skin contact. Moreover, the 500 kΩ resistors reduce the input current in case of a short circuit. The mode used when measuring ECG signal, for

any type of electrodes, is the sleep float mode in which the sensing capacitor is always floating except when the transfer of charge to the ADC occurs. Thus, the ADPD4101 has to be configured in this mode for the fourth time slot. Moreover, the band-pass filter has to be shut down in this mode and will be so bypassed. Furthermore, it is recommended to increase the gain of the transimpedance amplifier up to 100 kΩ since the ECG signal only provides ultra low currents. In addition to that, the ADPD4101 must be informed that the input pair IN7-IN8 will be used as a differential pair in this time slot. Finally, some requirements concerning the timings of the integrator offset and width as well as offset and width of the connected pulses have to be met. These timings are the same as the ones recommended in the datasheet[44].

Finally, the two electrode lead off measurement is performed on the last time slot. For this measurement, IN3 is the input pin, and will collect current injected by VC2 through both electrodes to measure if a good electrode skin contact is present. Since I had never experienced such measurements, I followed what was advised in the datasheet[44] and followed exactly what they mention for this type of measurement. This means that a precondition period as the one used in the PPG measurement is configured as well as the timings mentioned for the ECG measurement.

As a global consideration, the FIFO is filled with 16-bit signal at each time slot such that the FIFO use can easily be implemented in another project. Complete initialization code of the ADPD4101 can also be found in the `I2C_slave_init` function in the `MT.c` code that can be accessed via Appendix E.

The part of the main code in which the ADPD4101 intervenes consists in a periodic reading of the 5 described above signals. As already mentioned, the INTY signal will be the one ordering the MCU to read the signal values. To do so, the INTY triggers an interrupt in the MCU which pushes a signal in the waiting queue ordering the MCU to read the values of the 5 first time slots of the ADPD4101. These values are read thanks to the function `I2C_read_ADPD` present in the `MT.c` code that can be accessed via Appendix E, so that data can be sent over the air using BLE. In addition to that, a function clearing the FIFO is called when the INTX is triggered. Finally, note that the ADPD4101 and TMP1075 are not started at the

instant when the device is powered up as for the BQ25125 but rather waits that a pairing BLE device is connected to get started. This is done such that no power is lost on, for instance, blinking the LEDs if there is no need since the LEDs will probably be what will consume the most of the battery charge.

4.3.2.5 Main code functions

The main code is composed of two functions: a task function being the entry point of the RTOS which basically waits for an event (either from BLE or from the application) in order to call the right function and an application message function which is used to process an application message. Those two functions are present in the `MT.c` code that can be accessed via Appendix E.

Chapter 5

Peripheral device software: Android app

In this project, a BLE pairing device is necessary to receive, process and plot data on a screen. An *Android*[1] app will be developed in the context of this project since using this kind of platform allows a mobile use and is within everyone's reach. In brief, this app will be responsible of monitoring battery level, temperature, ECG (signal and lead-off) and PPG data so that battery level, body temperature, HR and SpO2 can be displayed on the screen. Moreover, the ECG and PPG (coming from the green LEDs) signals can be plotted in real time. The software used for developing this app is *Android Studio*, being a reference in the domain, and it offers some great tools when debugging the app. The language for programing this app is *Java*. In the following, the code architecture will be described and focus will be put on the signal processing part of the app.

5.1 Code architecture

The code used for this app has as a starting point the *BluetoothLeGatt* example[80]. This code example enables a connection with a BLE device and displays raw characteristic values when asked for a read or notify operation. The main code of this example is composed of two main parts: one (called the BLE part) responsible of scanning BLE channels, connecting to a BLE device, discovering services and re-

[1] An iOS app would also have been developed but I am in possession of an *Android* smartphone and not an iPhone

ceiving data from these and one (called the layout part) responsible of processing, displaying and plotting data.

5.1.1 BLE codes

The changes made to this part of the *BluetoothLeGatt* example consists in informing the *Android* app of which service corresponds to which physiological signal. To do so, one has to match service and characteristic UUIDs to what they consist in the real world, by following what had been said in section 4.2. Moreover, in the used code example, there exists a communication system between the BLE and layout parts consisting in only one string array for each characteristic.

This communication system is not sufficient for the constraints of this project and may cause some misunderstandings concerning the characteristic from which the data string originates. For this reason, I decided to add a single character in front of each string data so that the layout part can identify from which service the data is coming from by looking at the first element of the string array. A list of the flags used is represented in Table 5.1.

Flag	Corresponding characteristic
`B`	Battery level
`T`	Body temperature
`E`	ECG signal
`L`	ECG lead off
`P`	PPG signals

Table 5.1: List of the flags used in the *Android* app.

For instance, if the BLE part receive the value `88` from the battery level characteristic, the string array `B88` will be transferred to the layout part. Moreover, this way of doing is suitable for the PPG characteristic: this characteristic containing 3 values (green, red and IR), the flag `P` can also be used as a delimiter to separate the 3 PPG signals. Thus, if the BLE part receives `50`, `56` and `42` from the PPG characteristic, the layout part will receive the string array `P50P56P42`. This is so a suitable way of communicating data inside the app such that the source of data can be directly identified.

5.1.2 Layout codes

In the *BluetoothLeGatt* example, the layout part consists in printing raw data. However, it is not sufficient for this project since plotting and some processing are needed. First of all, the plotting library that will be used is *GraphView*[81]. This library is one of the most used libraries and is enough customizable for this project.

Concerning the battery service, there is no real need of data processing since the received value will be directly displayed on the screen. Thus, the displayed battery level corresponds to the battery capacity level of Table 4.3. This has been done so that the user can easily know when to connect the charger.

After that, the received body temperature from the MCU is not expressed in °C as already mentioned. To transform it to °C, one has to multiply the received value by 0.0625°C.

Concerning the ECG signal, this one is plotted in real time and is also used to calculate the HR in bpm (beat per minute). The way the HR is calculated is presented in the following section. Moreover, since the ECG lead off signal can be seen as a binary signal[2] (either the electrodes are well connected, either not), comparing the ECG lead off value to a threshold (situated in the middle of the two levels) is enough to give an information about the electrodes skin contacts. Figure 5.1 represents the time evolution of a lead off signal in which, one or two electrodes is disconnected at time A, C and E and reconnected at time B, D and F respectively.

Finally, the PPG signal is also plotted in real time and used to calculate the SpO2. On one hand, the signal that will be plotted is the PPG signal coming from the green LEDs as this is usually done in most oxygen saturation sensors, mainly due to its good ambient light rejection. On the other hand, the PPG signals coming from red and infrared LEDs will be used to calculate oxygen saturation level. The way the SpO2 is calculated will be presented in the following section.

[2]The ECG lead off signal is obviously not perfectly a binary signal but this approximation can be accepted.

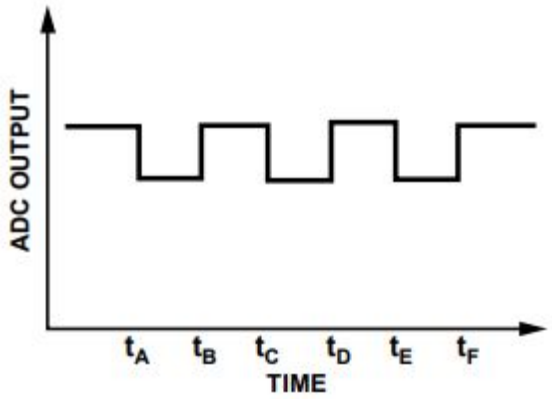


Figure 5.1: Example of a two electrodes lead off signal[44].

5.2 Signal processing

In this section details about the computation of both the heart rate and oxygen saturation level coming from the ECG signal and PPG ones respectively will be given.

5.2.1 Heart rate

As a reminder, a typical ECG signal is composed of several waves as represented in Figure 1.1. Calculating the HR from the ECG signal consists in determining the inverse of the ECG signal period. To do so, one very common way of doing is to detect the time interval between QRS complexes of different ECG signal periods and to invert this time interval[82]. This time interval is called the RR interval as shown in Figure 5.2.

Indeed, this is a suitable way of calculating the HR since the R peak of the QRS complex is the simplest ECG signal period part that can be detected thanks to its high amplitude compared to the rest of the signal. Thus, this R peak can be detected by means of a threshold and the use of hysteresis is also common to not detect several R peaks on a single R peak due to noise.

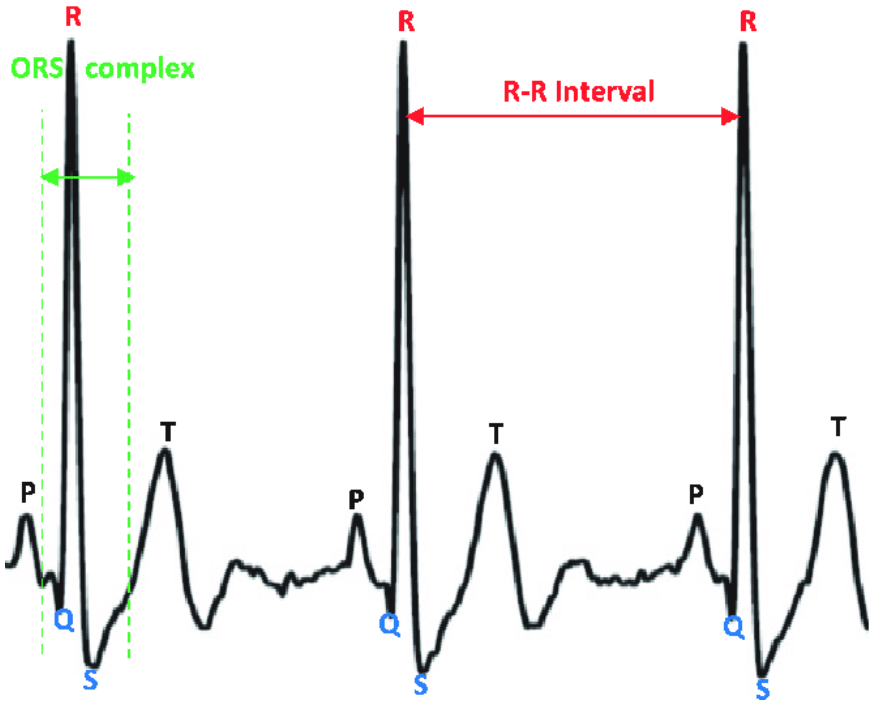


Figure 5.2: RR interval of an ECG signal[83].

Thus, if the RR interval is measured in a number of samples (here being in tens of milliseconds), the formula to calculate the HR in bpm is:

$$HR = \left(\frac{1}{RR\_interval}\right) \cdot 100 \cdot 60$$

5.2.2 Oxygen saturation level

As already mentioned, using the fact that the red and infrared light extinction coefficients of oxyhemoglobin and deoxyhemoglobin are different as observed in Figure 1.6, oxygen saturation level can be calculated thanks to PPG signals coming from red and infrared LEDs. Actually, these signals consist in a DC part and an AC part: the DC part contains light emitted by venous blood, tissues and non-pulsatile component of artery blood while the AC part contains pulsatile component of artery blood. This is represented in Figure 5.3.

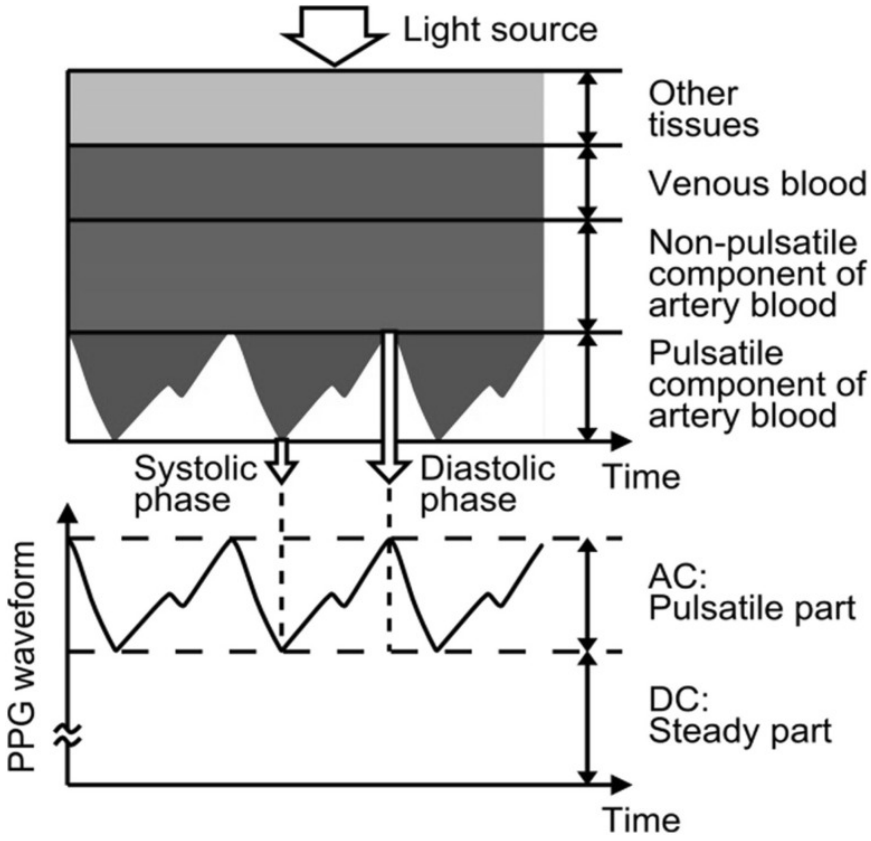


Figure 5.3: Example of a PPG signal[84].

Actually, the SpO2 is the percentage of oxyhemoglobin over all the hemoglobin available in artery blood. This value can be calculated exactly from the AC and DC parts of both red and IR PPG signals[85]:

$$SpO2 = \frac{\epsilon_{Hb}(\lambda_{red}) - \epsilon_{Hb}(\lambda_{IR}) \cdot R}{\epsilon_{Hb}(\lambda_{red}) - \epsilon_{HbO2}(\lambda_{red}) + (\epsilon_{HbO2}(\lambda_{IR}) - \epsilon_{Hb}(\lambda_{IR})) \cdot R} \cdot 100\%$$

with $\epsilon(\lambda)$ the extinction coefficient at light frequency λ, Hb the deoxyhemoglobin, $HbO2$ the oxyhemoglobin and R the ratio of normalized absorbances (calculated with red and IR PPG signals).

However, this expression is not practical and a very good approximation is commonly used[86]:

$$SpO2 = 110 - 25 \cdot R$$

with

$$R = \frac{\log\left(\frac{AC_{red}+DC_{red}}{DC_{red}}\right)}{\log\left(\frac{AC_{IR}+DC_{IR}}{DC_{IR}}\right)}$$

in which DC_{red} and DC_{IR} are the DC parts of the red and infrared PPG signals respectively and AC_{red} and AC_{IR} are the AC parts of the red and infrared PPG signals respectively.

5.3 A tour within the app

The app has first a main screen in which scanning is possible and it displays all the BLE devices ready for pairing around the smartphone. This main screen is shown in Figure 5.4.

After that, the user can decide to pair to a device and the app discovers the available services and displays them as shown in Figure 5.5.

Finally, when clicking on one characteristic, a read operation is performed. If a second click is realized, the notify mode is activated and continuous data flux is represented on the screen following what had been said earlier as shown in Figure 5.6 for the PPG signal. Note that the signal shown on the screen is a simulated sawtooth signal use for test purposes.

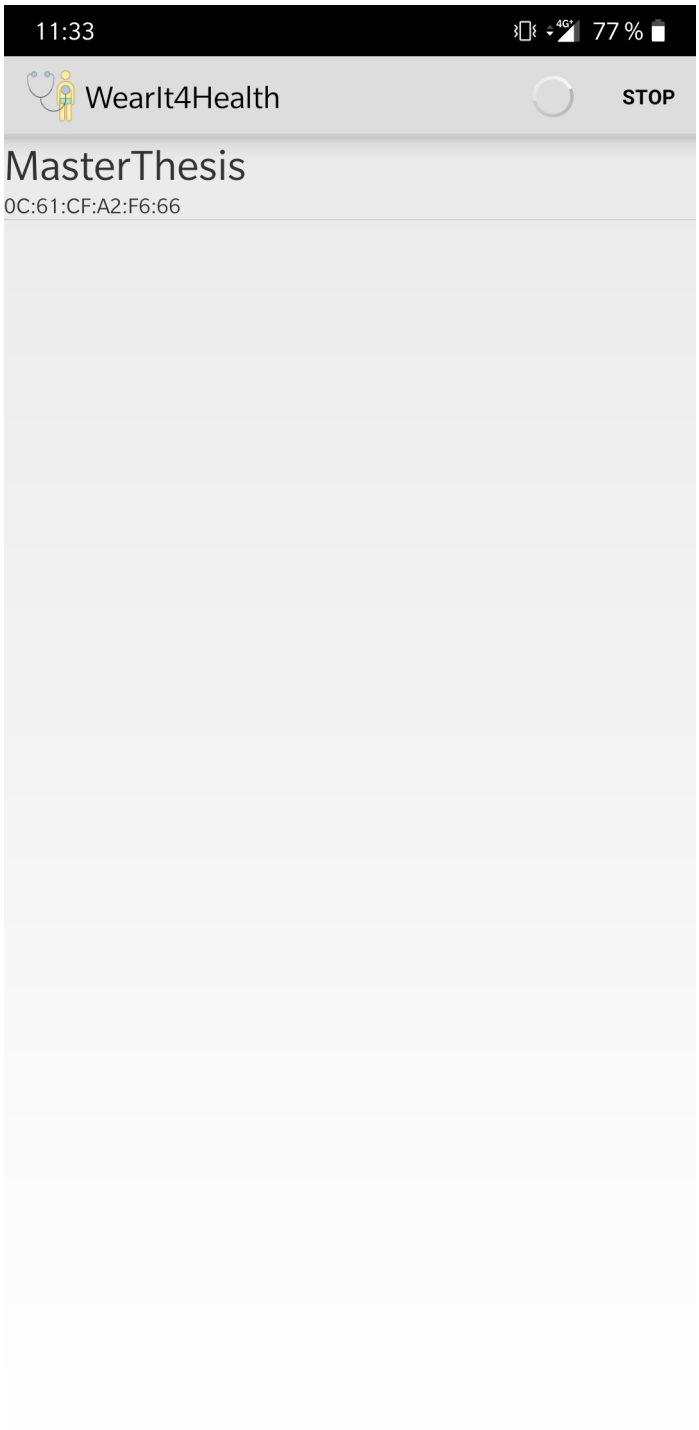


Figure 5.4: Scanning screen of the app.

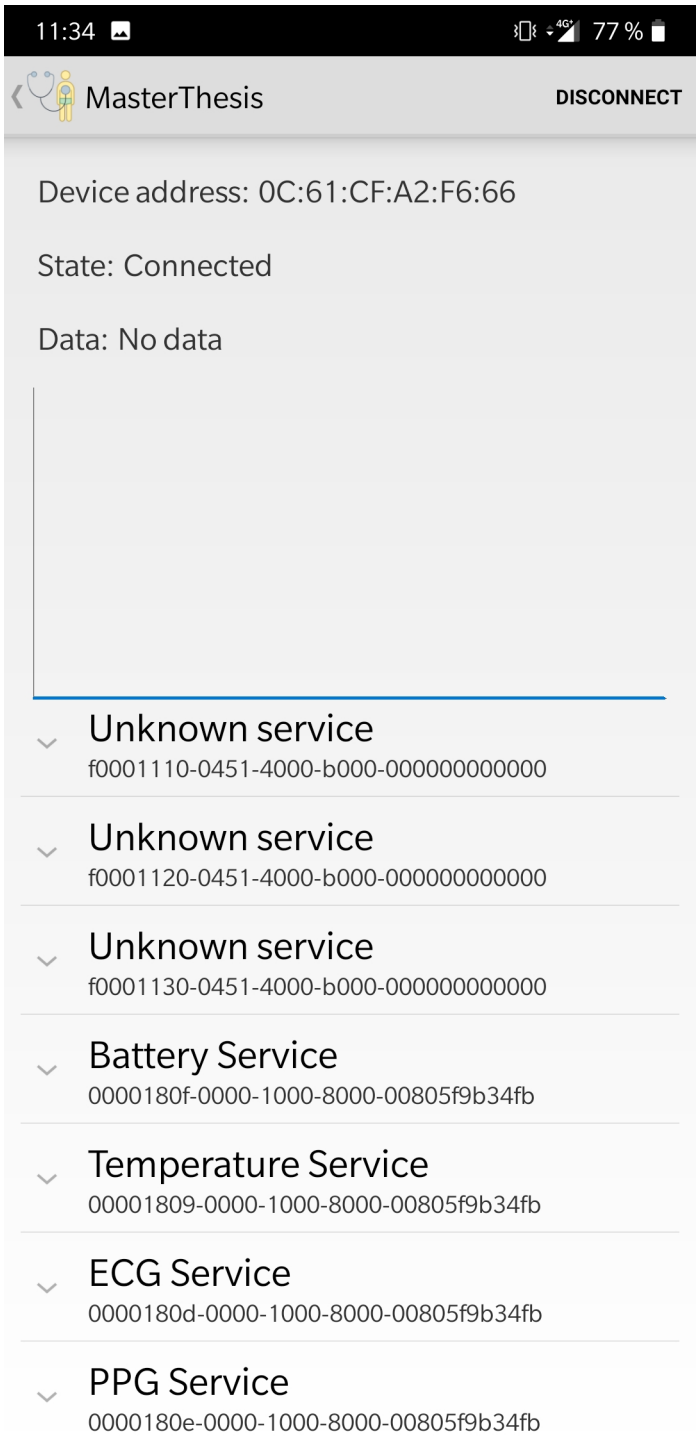


Figure 5.5: Characteristic list screen of the app.

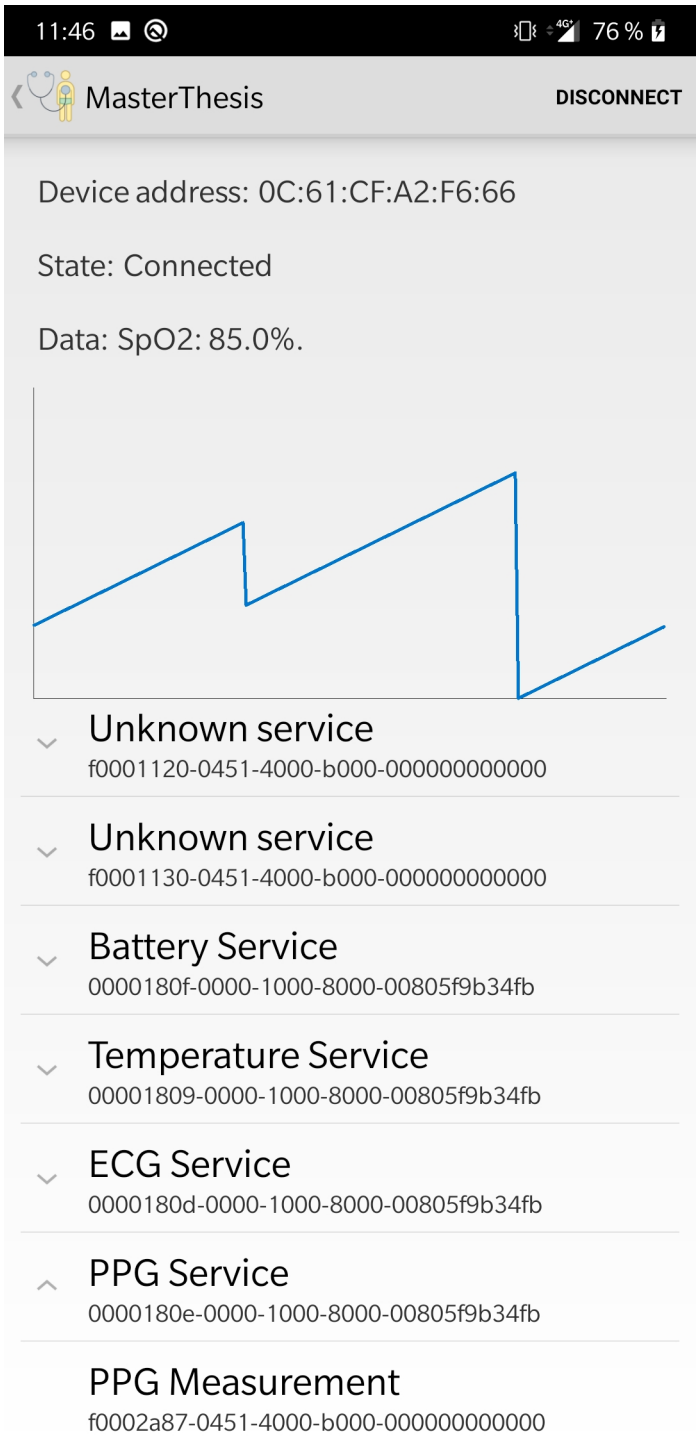


Figure 5.6: PPG test signal graph and SpO2 test displaying in the app.

Chapter 6

Experimental validation

In this chapter, the tests and results obtained with both prototypes will be presented. Focus will be put rather on testing modules separately instead of a complete prototype testing, especially due to practical issues that will be presented.

6.1 First prototype

This first prototype is represented in Figures 3.17 and 3.18 and is without a MCU due to the worldwide shortage of 4 mm x 4 mm CC2640. Thus, no data can be collected and sent via BLE with this prototype. Thus, the only test one can realize with it is to test the battery management system: the BQ25125. Indeed, even if this last is not programmed through I2C by the MCU, the default configuration enables to be connected to a battery, output a stable 1.8 V and to charge the battery. Thus, several discharge/charge cycles had been realized on one LIR2032 without any issues and a stable 1.8 V was observed between $V\ \mu C$ and GND. Moreover, a blue LED used for visual indication concerning charging is used[39]:

- LED is off: no charger is connected to the board or the charge is complete;
- LED is blinking: an incorrect/incompatible charger is connected, this disables battery charging;
- LED is on: a valid charger is connected, this enables battery charging.

Figure 6.1 shows a situation in which a valid charger is connected to the board.

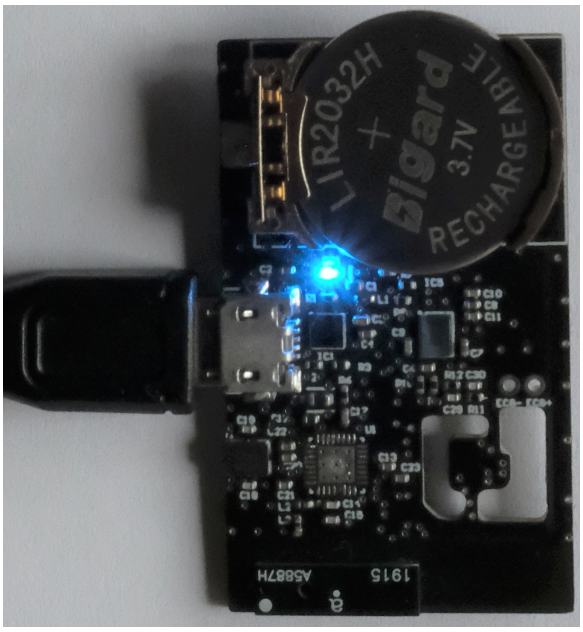


Figure 6.1: Top view of the first prototype when the battery is charging.

Thanks to this elementary test, the power part of the prototype could be validated.

6.2 Second prototype

This second prototype, represented in Figures 3.21 and 3.22, was ready nearly one month and a half after having the first one assembled (without the MCU). Before starting programming the MCU of this second prototype, short circuits were discovered between V μC and GND and between SDA and SCL, that were not present in the first prototype. Indeed, the BQ25125 and the TMP1075 were not soldered properly and generated a short circuit. Thus, in order to start programming the MCU as soon as possible, those components were desoldered permanently[1]. Thus, the tests on the second prototype were started without the BQ25125 and TMP1075, meaning that the board had to be supplied externally and that the temperature sensor would not work.

[1]Asking to resolder them correctly would have take some times since the technician from Microsys was fully busy by other projects, and time was running out for the tests.

In addition, the SFH7072 bundle was out of stock worldwide during the realization of the second PCB and, without this bundle, the PPG signals cannot be measured.

6.2.1 Programming the MCU

In order to write a program into the MCU, the XDS110 out connector of the CC2640 evaluation board is used. Indeed, this type of connection permits the programming of the MCU through JTAG (5 debugging pins) at any voltage thanks to a level shifter integrated in the programmer. Moreover, all the jumpers connecting the programmer and the CC2640 of the board have to be disconnected so that the programmer only sees the custom board. The XDS110 out connector is shown in Figure 6.2.

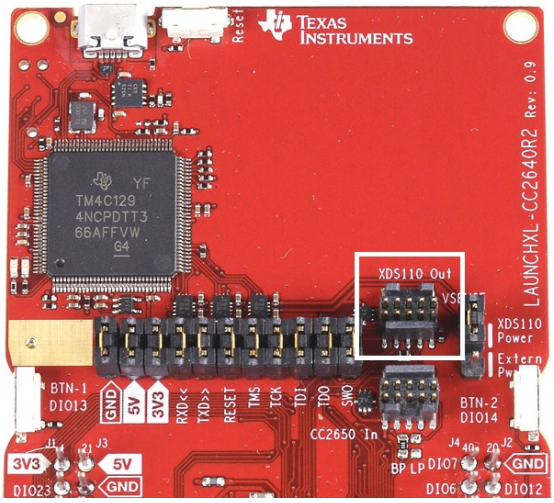


Figure 6.2: XDS110 debug probe of the CC2640 evaluation board[63].

However, no XDS110 debug probe was available at Microsys to connect the board to the XDS110 connector. For this reason, this last was desoldered to access the JTAG and level shifter pins directly on the evaluation board to solder some cables that could be connected to the custom board using a breadboard. The setup used to program the custom board is represented in Figure 6.3.

As can be seen in Figure 6.3, several cables connect the programmer to the custom board. More precisely, there are 2 cables for the level shifter (V μC in red and GND in black and blue) and 5 JTAG cables that follow a color code:

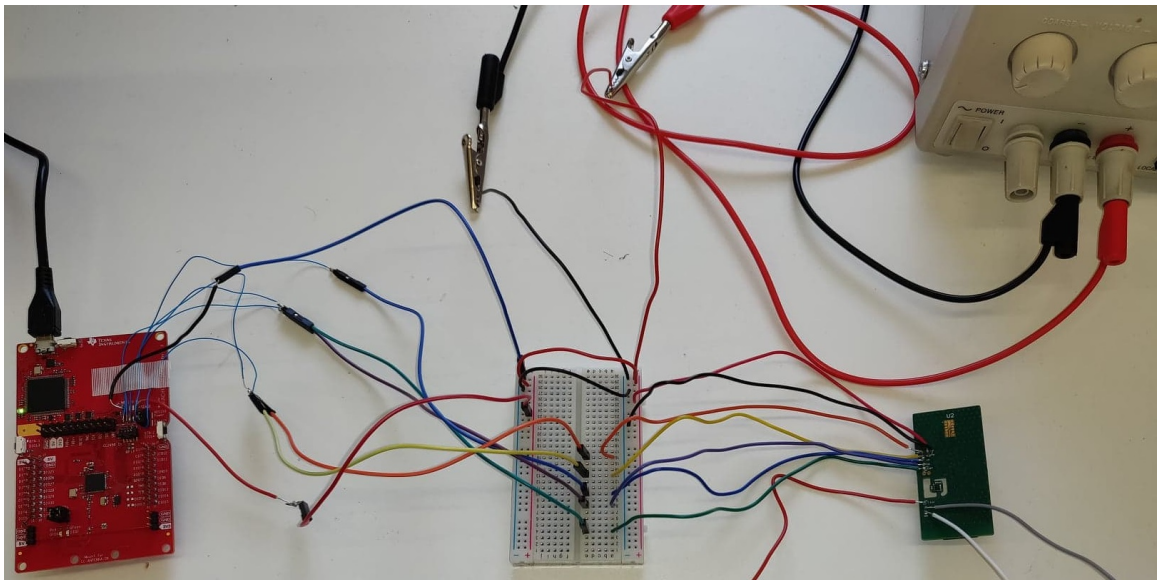


Figure 6.3: Debugging setup.

- Orange: JTAG TMSC;
- Yellow: JTAG TCKC;
- Blue: JTAG TDO;
- Purple: JTAG TDI;
- Green: RESET.

In addition to that, 3 cables can be connected to the level shifted I2C bus lines for debugging purposes, being the output of the PCA9306. Note that these bus lines need a reference voltage, being the GPIO voltage at 3.3 V of the evaluation board by default due to its easy accessibility.

6.2.2 Testing the antenna

Now that the MCU can be programmed, the first module that is tested is the wireless communication given this only involves the MCU. To do so, an elementary program is written in the MCU that only enables advertising and pairing, without any exchange of data. At first, the custom app was able to detect the board advertising but not to get paired with it, the connection was lost nearly immediately.

After a discussion with a *Texas Instruments* engineer, this was due to the calibration of the low frequency internal oscillator[87]. Indeed, the BLE protocol requires a stable low frequency oscillator acting as a real time clock in order to schedule the different meetings between the paired devices. In this case, the real time clock of the MCU did not have the same speed as the one of the smartphone, inducing the fact that the MCU was missing some meetings and the connection is terminated. The condition to not miss any meetings is that the frequencies of the low frequency oscillators must not be more than 500 ppm off the 32.768 kHz frequency[33]. Concerning this condition, it was clear that the problem was coming from the custom board and not from the smartphone, this last being calibrated such that the above condition is respected.

As already mentioned, the MCU either requires an external low frequency oscillator either requires periodic calibration of its internal low frequency RC oscillator. In this project, the internal oscillator is used such that space is saved on the board. Thus, it was clear that the pairing problem that was observed was coming from a faulty periodic calibration of the internal RC low frequency oscillator. From the TI RTOS documentation, a proper calibration of the internal RC oscillator is realized if[55]:

- The header file `rcosc_calibration.h` containing calibration functions is included;
- The periodic calibration function `RCOSC_enableCalibration()` is called;
- The RC oscillator is enabled in the power policy.

From this, the problem observed was coming from this last point which was badly executed by the MCU. Indeed, if the internal RC oscillator is not powered, it could not be calibrated. Finally, a stable connection could be made between the MCU and the smartphone such that the wireless module is also validated.

Note that the calibration period can be tuned and the longer this period, the lower the power consumption of the calibration. However, it was not possible to increase the period infinitely and keep a stable connection due to the fact that the

temperature of the MCU should no change more than 1°C between two calibrations, otherwise the frequency condition will not be respected anymore[88]. Thus, the chosen calibration period is 1 second, being the default one, since no change of more than 1°C/s should be observed for this project and since that the decrease in power consumption when increasing further this period is very small compared to the power consumption of other modules on the board.

The performances of this custom antenna can be evaluated by comparing the received signal strength indications (RSSI) of both the custom antenna and the antenna from the evaluation board (being much larger) at different distances. Indeed, the RSSI is a measure of the power of the received signal on the smartphone[89]. Thus, the larger the RSSI, the better the power received and the better the antenna. Figure 6.4 shows evolution curves of both RSSIs in function of the distance between the smartphone and the board[2].

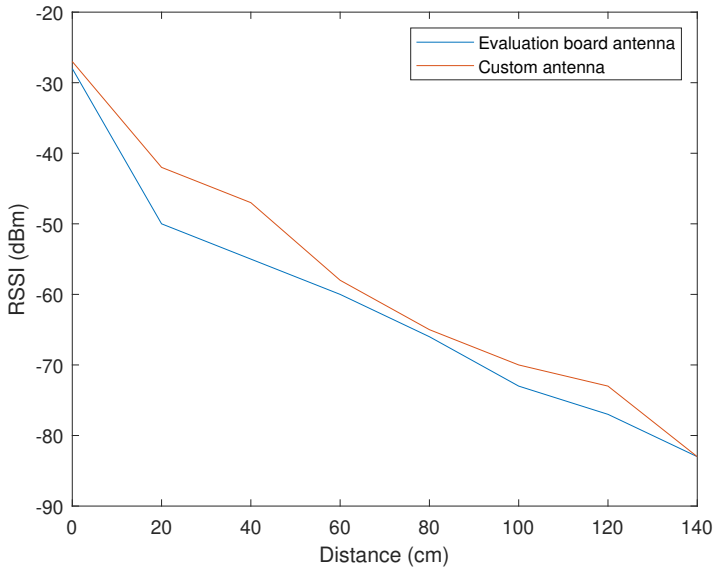


Figure 6.4: RSSI measures in function of the distance between the smartphone and the board for both antennas.

[2]Note that these measures are only indicative and were not meant to be precise. Indeed, the RSSIs are measured with the application *BLE scanner*.

As can be seen in Figure 6.4, the RSSI decreases as the distance between the smartphone and the board increases, and the curves are close to linearity. This was expected since the received power decreases with the squared distance and since the RSSI involves the logarithm of this received power. Moreover, it appears that performances of both antennas are comparable with slightly better RSSI values for the custom antenna. This demonstrates that the custom antenna and its matching network are well designed. Note that, for both antennas, connection is maintained up to roughly 10m without obstacles.

6.2.3 Testing the ADPD4101: the ECG signal

The next test that will be performed is to send useful data through BLE since a stable connection can be obtained. As already mentioned, only the ECG signal can be measured since none of the TMP1075 and the SFH7072 bundle are soldered on the second board.

In order to measure and collect the ECG signal, an I2C communication between the MCU and ADPD4101, whose I2C address is 0x24 by default[44], has to be ensured. A first test would be to read the chip ID of the used ADPD4101 located on register 0x0008. However, the soldered ADPD4101 did not acknowledge (NACK) to the address 0x24. To try to remedy to this problem, a global call had been realized but the ADPD4101 was still not responding. Note that a global call is addressing slaves with the address 0x00 being a special address on which every slave should acknowledge the call.

A last try was realized and consists in looping the calls by trying all possible 7-bit addresses, but the ADPD4101 answered to none of them. Thus, either the I2C bus hardware is not correct and signals could be noisy either the ADPD4101 is not supplied due to bad soldering for instance. This first potential problem was debugged using an oscilloscope and observing the I2C bus lines voltages. This debugging method showed that the I2C bus hardware was correct and readable signals were observed on the oscilloscope as represented in Figure 6.5.

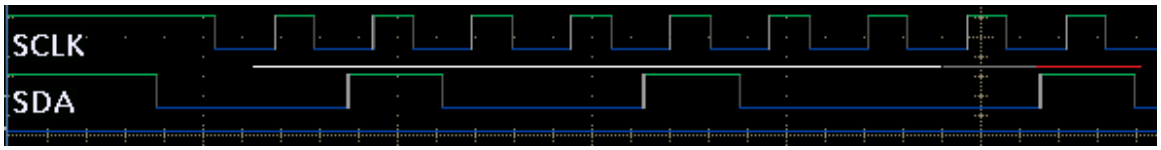


Figure 6.5: NACK example from the ADPD4101 when trying to access any register.

As can be seen in Figure 6.5, when trying to access any register, the MCU writes on the SDA line the ADPD4101 address (in white) and a read/write bit equaling 0 (in gray). However, the ADPD4101 seems to not hear the calls and a NACK is returned (in red), and this happened for all possible addresses.

For this last reason, the most likely is that the ADPD4101 welds are defective. This induces that no test was possible with the ADPD4101 at the beginning. Nevertheless, since I really wanted to measure at least one vital sign with this project before the deadline, the TMP1075 was resoldered on the custom board.

6.2.4 Testing the TMP1075

Since that the ADPD4101 was not working and validating at least one sensor would be great, a TMP1075 was resoldered on the board by Ir. François Dupont. It appeared that this new soldering was correct and no short circuit were observed. Moreover, the TMP1075 did acknowledge the I2C transactions and temperature could be extracted following what had been described above. As a example, an I2C transaction between the MCU and the TMP1075 consisting in a temperature reading is represented in Figure 6.6.

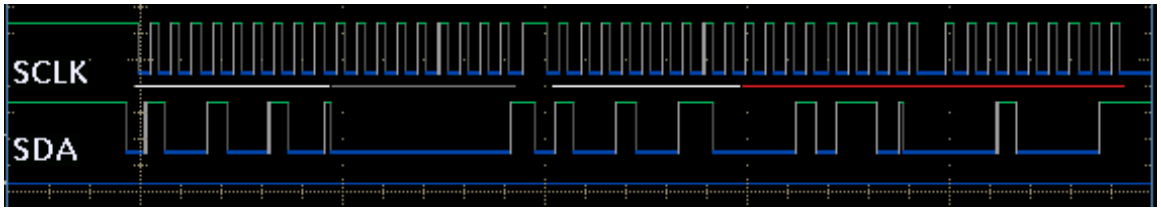


Figure 6.6: Example of an I2C transaction between the MCU and the TMP1075 in which the MCU read the temperature conversion result.

In Figure 6.6, it can be seen that the MCU starts writing the TMP1075 address in writing mode to call it (in white) followed by an acknowledgment. Then, the MCU

writes the 8-bit register address it wants to read, i.e. the result register, again acknowledged (in gray). After that, a restart condition is used to recall the TMP1075 by its address and in reading mode (in white). Finally, the TMP1075 communicates the two bytes result, acknowledged by the MCU (in red).

Using the designed app, temperature readings can be performed and seemed correct without any temperature shift, thanks to the cut-outs. Indeed, room temperature measurement had been performed as well as body temperature measurement, being in the physiological range. Just note that if any large change in temperature is observed (first measuring room temperature then body temperature), the temperature reading takes some time to stabilize to the correct value, as in any contact thermometer. Thanks to this, the body temperature measurement part of the device is validated.

6.3 Presentation of the final prototype

The final prototype of this project would be stick on the chest of the patient, like wearIT4Health, with two electrodes used to measure the ECG signal. This final prototype would be communicating useful medical data such as HR, SpO2 and body temperature wirelessly to a Bluetooth compatible smartphone. Figure 6.7 represents the second prototype of this master thesis worn on the chest of a potential patient.

As already said, this device is not completely operational due to the lack of the BQ25125 and the inability to communicate with the ADPD4101, being respectively responsible of the need of an external power supply and an inability to measure PPG and ECG signals. Moreover, a medical device derived from this prototype would be surrounded by a plastic package used to stick properly the device to the patient's chest as well as a clipping system to connect properly the electrodes to the device cables.

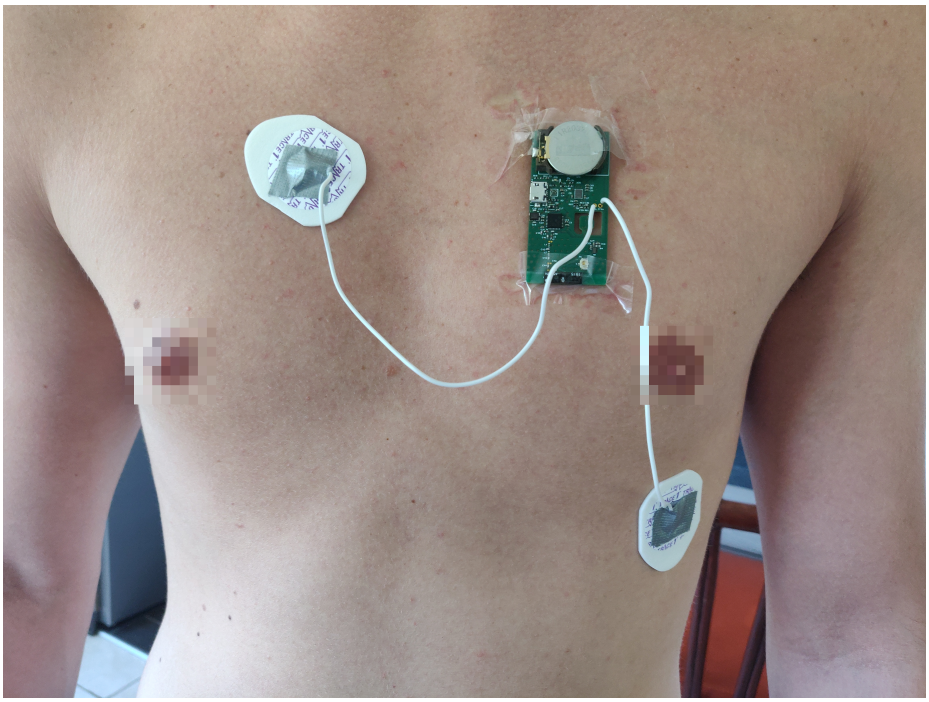


Figure 6.7: The second prototype of this master thesis worn by a patient.

Despite the fact that this prototype is not fully operational, it can still be used as a Bluetooth wearable thermometer by using the first prototype of this master thesis (whose battery management system works) as the external power supply needed for the second prototype.

Conclusion and perspectives

In this project, two prototypes of a wearable biomedical monitoring device have been designed and tested as well as an Android app. This device is designed in a way that some vital signs (heart rate, oxygen saturation and body temperature) can be extracted from physiological signals such as ECG and PPG signals that are sent wirelessly via Bluetooth Low Energy.

On one hand, the first version could not have been completed in due time since the delivery times of the MCU are exceptionally long. However, this prototype had been used to test the power part of the device in order to validate it. On the other hand, the second version was completed except the fact that the PPG signal could not be measured due to a lack of LEDs and PDs. This version was used to validate the RF part as well as the body temperature sensor one. Unfortunately, no communication with the PPG/ECG sensor could had been established.

Overall, the second prototype was only an operational wireless body temperature sensor, not being a revolution in the biomedical sensing domain. However, this project demonstrates, thanks to the validation of the separate parts of the device, that much smaller biomedical monitoring devices than those existing today can be designed and marketed for the best comfort of both the patients and the nurses. Indeed, even very recent devices such as wearIT4Health can be further miniaturized by using new electronic devices such as the ADPD4101.

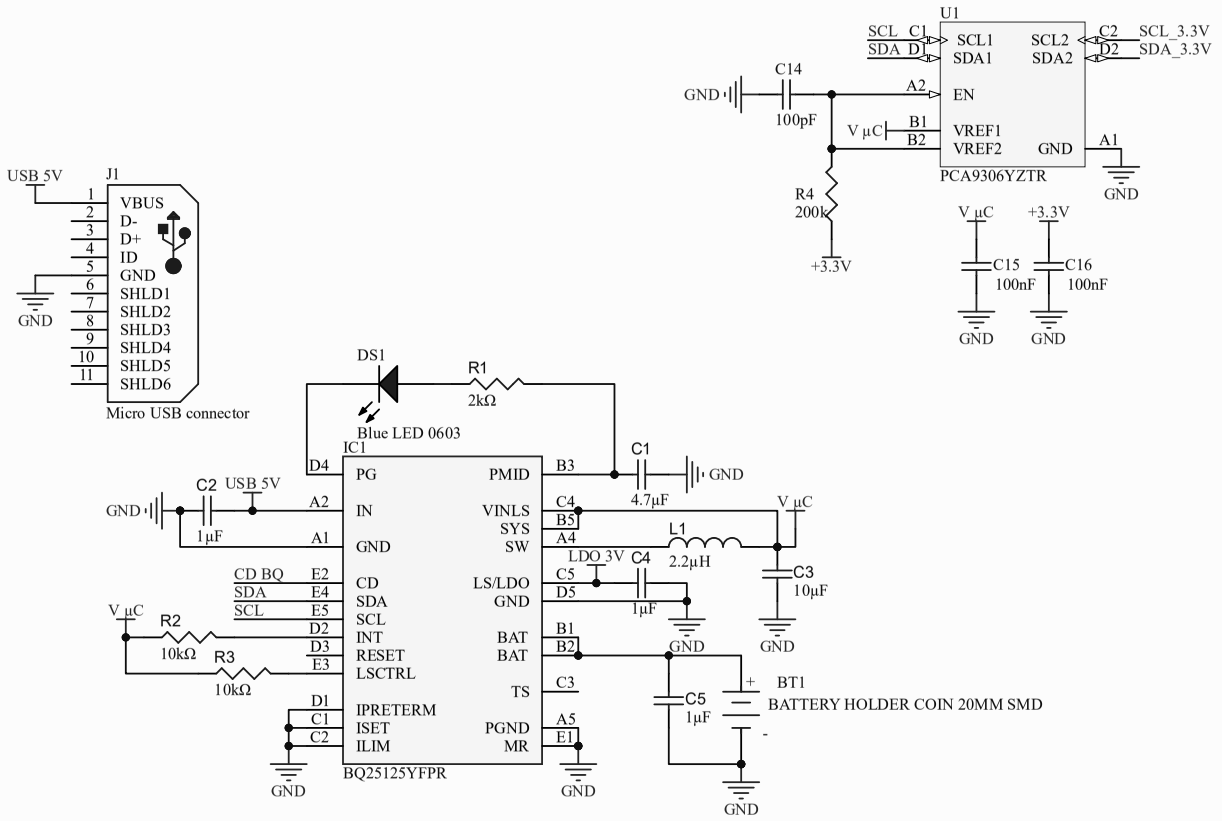
In the future, this device can be finished by resoldering all components that had been badly soldered so that a proper behavior on the I2C bus will be observed. Moreover, this work, being a sort of wearIT4Health miniaturization, could be applied to many biomedical monitoring devices to reduce their size and increase

comfortability. Over the years, biomedical devices had been smaller and smaller in order to achieve sizes comparable of the device of this project. As a final goal of this miniaturization story, biomedical devices would look like to millimetric patches that would be stick on the skin and invisible, such that comfortability of the patients would be maximum and amount of work for the nurses minimal.

Appendices

Appendix A

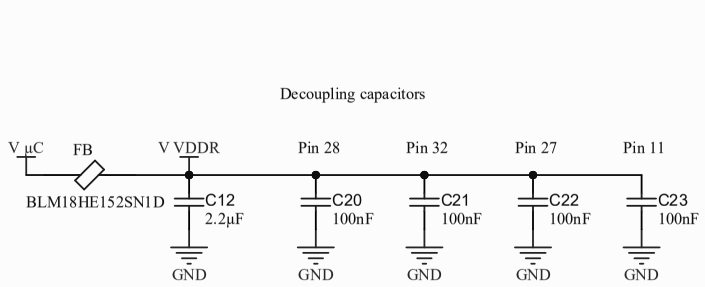
Electronic schematic of the power part

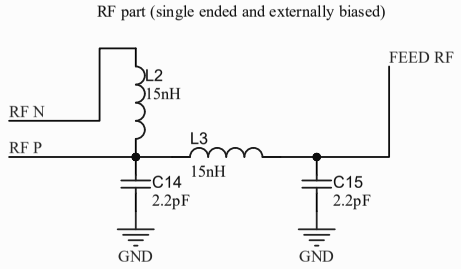


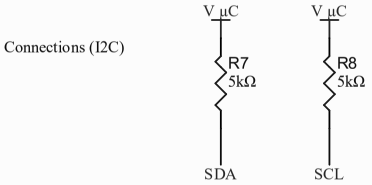
Title		
Size: A4	Number	Revision
Date: 4-20-2021		Sheet of
File: C:\Users\..\BQ25125.SchDoc		Drawn By:

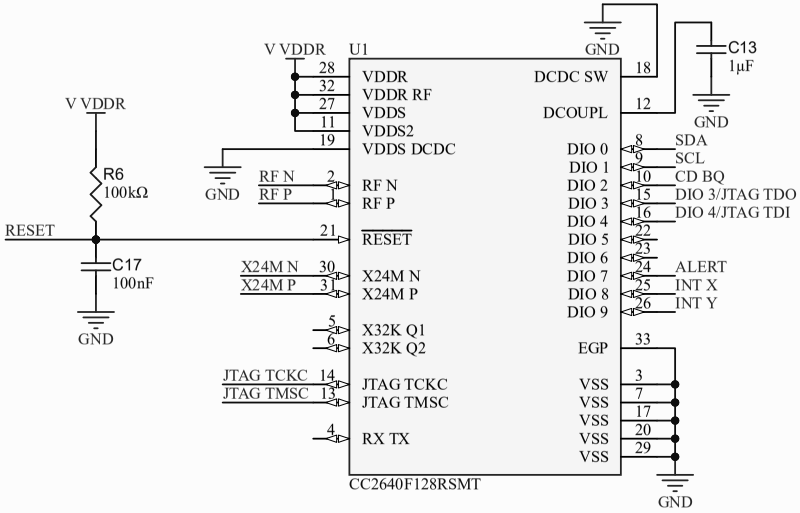
Appendix B

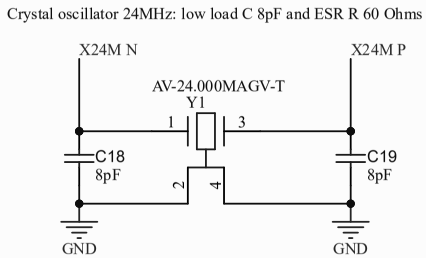
Electronic schematic of the logic part









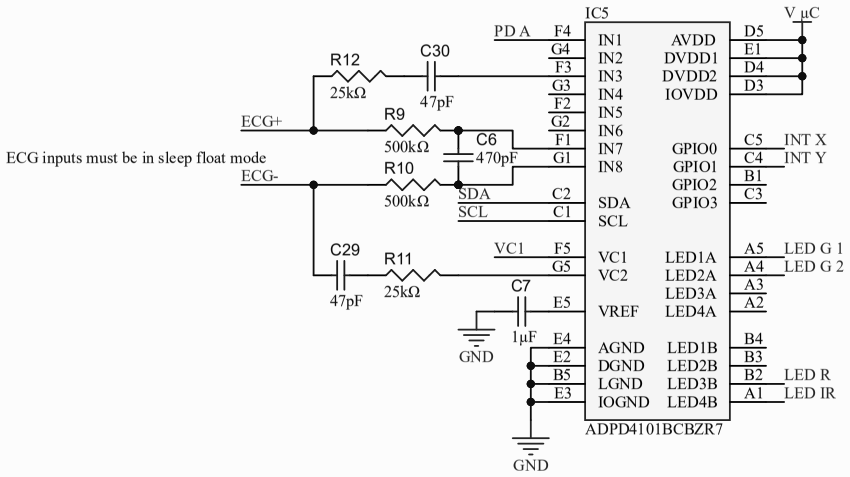


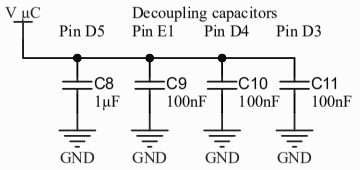
Title		
Size: A4	Number	Revision
Date: 4-21-2021		Sheet of
File: C:\Users\..\CC2640.SchDoc		Drawn By:

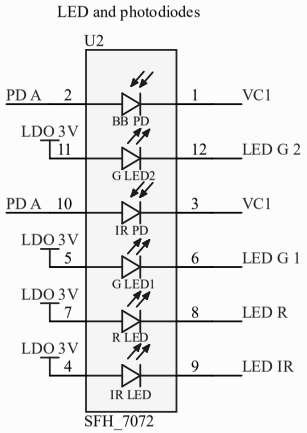
Appendix C

Electronic schematic of the sensor part

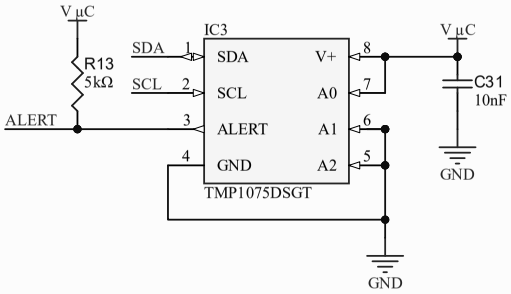
For instance, INT X for FIFO threshold interrupt and INT Y for FIFO over/underflow or rooting the data interrupt for a single time slot







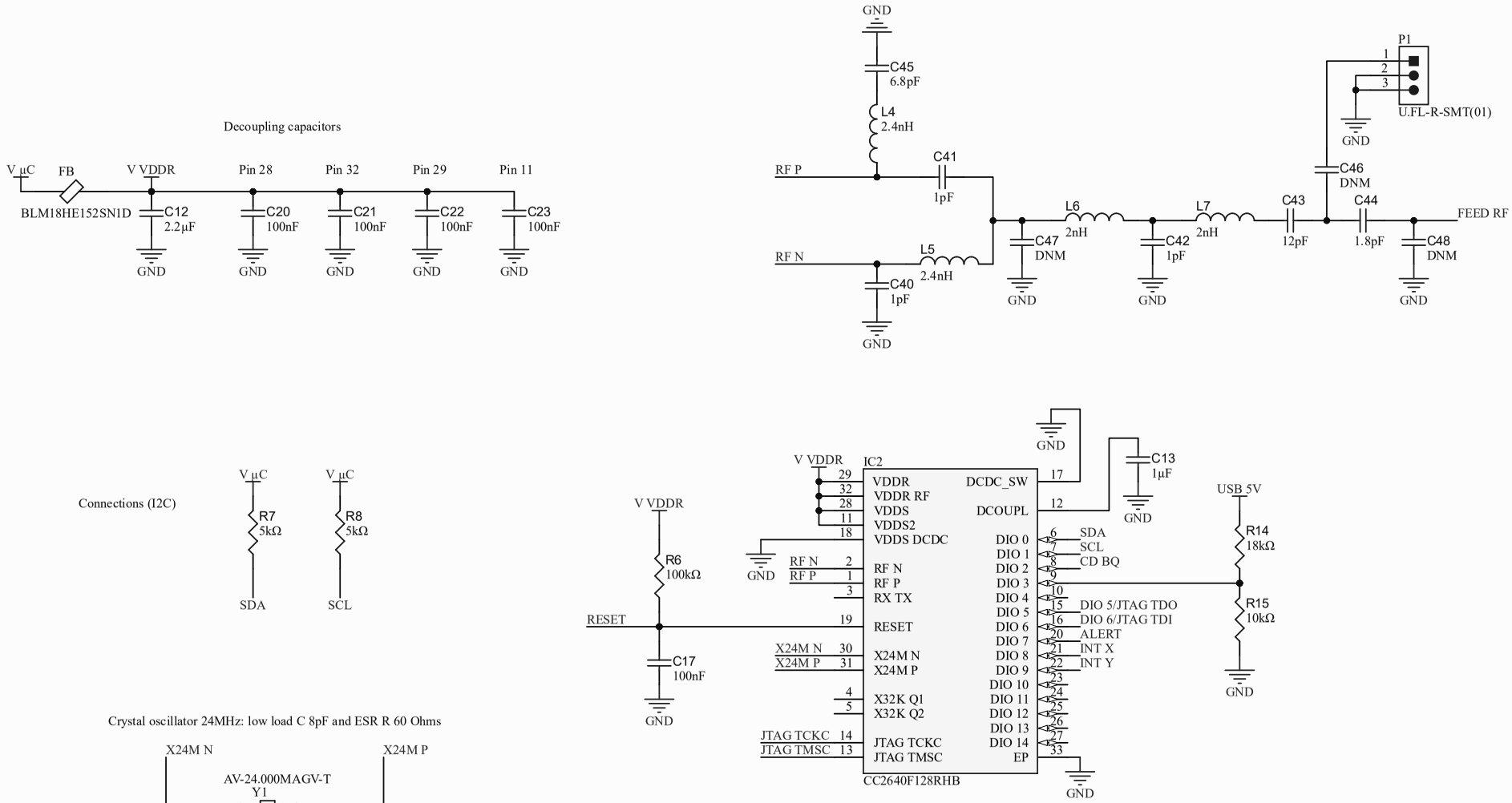
Temperature sensor



Title			
Size A4	Number		Revision
Date: 4-21-2021		Sheet of	
File: C:\Users\..\ADPD4101.SchDoc		Drawn By:	

Appendix D

New electronic schematic of the logic part



Title		
Size: A4	Number	Revision
Date: 4-22-2021		Sheet of
File: C:\Users\..\CC2640_differential.SchDoc		Drawn By:

Appendix E

C codes of the Code Composer Studio project

All the codes used to program the MCU in this master thesis project were uploaded on a *GitHub* repository (`https://github.com/arthur-fyon/Master-Thesis-CCS-Project`) publicly available.

Bibliography

[1] wearIT4health Coordination Office. *Description of the wearIT4Health project*. URL: http://www.wearit4health.com/page/activities. Last accessed on April 16, 2021.

[2] T. Wu et al. "A Rigid-Flex Wearable Health Monitoring Sensor Patch for IoT-Connected Healthcare Applications." In: *IEEE Internet of Things Journal* 7.8 (2020). DOI: 10.1109/JIOT.2020.2977164.

[3] S. S. Thomas et al. "BioWatch: A Noninvasive Wrist-Based Blood Pressure Monitor That Incorporates Training Techniques for Posture and Subject Variability." In: *IEEE journal of biomedical and health informatics* 20.5 (2016), pp. 1291–1300. DOI: 10.1109/JBHI.2015.2458779.

[4] H. H. Asada et al. "Mobile monitoring with wearable photoplethysmographic biosensors." In: *IEEE engineering in medicine and biology magazine* 22.3 (2003), pp. 28–40. DOI: 10.1109/MEMB.2003.1213624.

[5] Microsys. *List of the projects*. URL: http://www.microsys.uliege.be/index.php?menu=1&submenu=3&lang=en. Last accessed on April 16, 2021.

[6] Microsys. *Description of the wearIT4Health project*. URL: http://www.microsys.uliege.be/index.php?menu=1&submenu=3&lang=en#wearit4health. Last accessed on April 16, 2021.

[7] Interreg EMR. *Interreg EMR project: WearIT4Health*. URL: https://www.youtube.com/watch?v=LP6tfpxsoXs. Last accessed on April 16, 2021.

[8] wearIT4health Coordination Office. *Description of the wearIT4COVID project*. URL: http://www.wearit4health.com/page/wearIT4COVID. Last accessed on April 16, 2021.

[9] Wikipedia. *Vital signs*. URL: https://en.wikipedia.org/wiki/Vital_signs. Last accessed on April 20, 2021.

[10] Chu M. et al. "Respiration rate and volume measurements using wearable strain sensors." In: *npj Digital Med* 2.8 (2019). DOI: 10.1038/s41746-019-0083-3.

[11] Wikipedia. *Electrocardiography*. URL: https://en.wikipedia.org/wiki/Electrocardiography. Last accessed on April 20, 2021.

[12] Better Health channel. *ECG test*. URL: https://www.betterhealth.vic.gov.au/health/ConditionsAndTreatments/ecg-test. Last accessed on April 20, 2021.

[13] ECG Echo learning. *ECG in left ventricular hypertrophy: criteria and implications*. URL: https://www.betterhealth.vic.gov.au/health/ConditionsAndTreatments/ecg-test. Last accessed on April 20, 2021.

[14] M. Punja et al. "Electrocardiographic manifestations of cardiac infectious-inflammatory disorders." In: *American Journal of Emergency Medicine* 28 (2010), pp. 364–377.

[15] Ed Burns. *Hypercalcaemia*. URL: https://litfl.com/hypercalcaemia-ecg-library/. Last accessed on April 20, 2021.

[16] Wikipedia. *Photoplethysmogram*. URL: https://en.wikipedia.org/wiki/Photoplethysmogram. Last accessed on April 20, 2021.

[17] A. Alzahrani et al. "A Multi-Channel Opto-Electronic Sensor to Accurately Monitor Heart Rate against Motion Artefact during Exercise." In: *Sensors* 15.10 (2015), pp. 25681–25702. DOI: 10.3390/s151025681.

[18] Wikipedia. *Oxygen saturation (medecine)*. URL: https://en.wikipedia.org/wiki/Oxygen_saturation_(medicine). Last accessed on April 20, 2021.

[19] Banner Health. *What You Need to Know About Your Blood Oxygen Level*. URL: https://www.bannerhealth.com/healthcareblog/teach-me/blood-oxygen-level-what-you-need-to-know. Last accessed on April 20, 2021.

[20] E. D. Chan et al. "Pulse oximetry: Understanding its basic principles facilitates appreciation of its limitations." In: *Respiratory Medicine* 107.6 (2013), pp. 789–799. DOI: 10.1016/j.rmed.2013.02.004.

[21] Wikipedia. *Electrodermal activity*. URL: https://en.wikipedia.org/wiki/Electrodermal_activity. Last accessed on April 20, 2021.

[22] Bryn Farnsworth. *What is EDA? And how does it work?* URL: https://imotions.com/blog/eda/. Last accessed on April 20, 2021.

[23] A. Y. Kim et al. "Automatic detection of major depressive disorder using electrodermal activity." In: *Scientific Reports* 8 (2018). DOI: 10.1038/s41598-018-35147-3.

[24] J. Karpe et al. "Objective assessment of pain-related stress in mechanically ventilated newborns based on skin conductance fluctuations." In: *Anaesthesiology Intensive Therapy* 45.3 (2013), pp. 134–137. DOI: 10.5603/AIT.2013.0028.

[25] Harrison McCreary Phillip Wang. *EDA Sensor*. URL: https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2006/hmm32_pjw32/index.html. Last accessed on April 20, 2021.

[26] Michigan Medecine. *Body Temperature*. URL: https://www.uofmhealth.org/health-library/hw198785. Last accessed on April 20, 2021.

[27] MedlinePlus. *Body temperature norms*. URL: https://medlineplus.gov/ency/article/001982.htm#:~:text=The%5C%20average%5C%20normal%5C%20body%5C%20temperature,by%5C%20an%5C%20infection%5C%20or%5C%20illness.. Last accessed on April 20, 2021.

[28] WebMD. *Fever*. URL: https://www.webmd.com/first-aid/fevers-causes-symptoms-treatments. Last accessed on April 20, 2021.

[29] MedecineNet. *What Body Temperature Is Too Low? Hypothermia*. URL: https://www.medicinenet.com/what_body_temperature_is_too_low_hypothermia/article.htm. Last accessed on April 20, 2021.

[30] Wikipedia. *IEEE 802.15.4*. URL: https://en.wikipedia.org/wiki/IEEE_802.15.4. Last accessed on April 21, 2021.

[31] Wikipedia. *Bluetooth Low Energy*. URL: https://en.wikipedia.org/wiki/Bluetooth_Low_Energy. Last accessed on April 21, 2021.

[32] Wikipedia. *Wafer-level packaging*. URL: https://en.wikipedia.org/wiki/Wafer-level_packaging. Last accessed on April 21, 2021.

[33] Texas Instruments. *CC2640 datasheet*. URL: https://www.ti.com/lit/ds/symlink/cc2640.pdf. Last accessed on May 25, 2021.

[34] Nordic Semiconductor. *nRF52805 Product Specification*. URL: https://infocenter.nordicsemi.com/index.jsp?topic=%5C%2Fstruct_nrf52%5C%2Fstruct%5C%2Fnrf52805.html. Last accessed on April 21, 2021.

[35] Dialog Semiconductor. *DA14580 datasheet*. URL: https://www.dialog-semiconductor.com/sites/default/files/da14580_fs_3v4.pdf. Last accessed on April 21, 2021.

[36] Texas Instruments. *SimpleLink Academy 4.40.00 for SimpleLink CC2640R2 SDK 4.40*. URL: https://dev.ti.com/tirex/explore/node?node=AAliBv1MZ54NaV.m2AIT8w__krol.2c__LATEST. Last accessed on May 6, 2021.

[37] Texas Instruments. *VQFN 4mm x 4mm package for CC2640*. URL: https://www.ti.com/lit/ml/mpqf195b/mpqf195b.pdf. Last accessed on April 21, 2021.

[38] Grepow. *What are the 3 stages of lithium battery charging?* URL: https://www.grepow.com/blog/what-are-the-3-stages-of-lithium-battery-charging/. Last accessed on April 22, 2021.

[39] Texas Instruments. *BQ25125 datasheet*. URL: https://www.ti.com/lit/ds/symlink/bq25125.pdf. Last accessed on May 22, 2021.

[40] Texas Instruments. *BQ25125 presentation*. URL: https://www.mouser.be/new/texas-instruments/ti-bq25125-battery-ics/. Last accessed on April 22, 2021.

[41] Texas Instruments. *AFE440 presentation*. URL: https://www.ti.com/product/AFE4400. Last accessed on April 22, 2021.

[42] Texas Instruments. *ADS1292R presentation*. URL: https://www.ti.com/product/ADS1292R. Last accessed on April 22, 2021.

[43] Maxim integrated. *MAX86150 presentation*. URL: https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX86150.html. Last accessed on April 22, 2021.

[44] Analog devices. *ADPD4100/01 datasheet*. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/adpd4100-4101.pdf. Last accessed on May 29, 2021.

[45] Analog devices. *ADPD4100/01 presentation*. URL: https://www.mouser.be/new/analog-devices/adi-adpd4100-4101-sensor-front-ends/. Last accessed on April 22, 2021.

[46] OSRAM Opto Semiconductors. *SFH7072 presentation*. URL: https://www.osram.com/ecat/BIOFY%5C%C2%5C%AE%5C%20SFH%5C%207072/com/en/class_pim_web_catalog_103489/prd_pim_device_2220016/. Last accessed on April 22, 2021.

[47] Melexis. *MLX90632 presentation*. URL: https://www.melexis.com/en/product/mlx90632/miniature-smd-infrared-thermometer-ic. Last accessed on April 23, 2021.

[48] Texas Instruments. *LMT70 presentation*. URL: https://www.ti.com/product/LMT70. Last accessed on April 23, 2021.

[49] Texas Instruments. *TMP1075 datasheet*. URL: https://www.ti.com/lit/ds/symlink/tmp1075.pdf. Last accessed on June 2, 2021.

[50] Texas Instruments. *WSON 2mm x 2mm package for TMP1075*. URL: https://www.ti.com/lit/ml/mpds308c/mpds308c.pdf. Last accessed on April 23, 2021.

[51] Voltaplex. *Coin cells: LIR*. URL: https://voltaplex.com/lithium-ion-battery/coin-cell. Last accessed on April 24, 2021.

[52] Rozin. *10 Pieces 3.6V LIR2032*. URL: https://www.amazon.fr/ROZIN-LIR2032-Batterie-Rechargeable-Remplacer/dp/B089K5K44Q. Last accessed on April 24, 2021.

[53] Harwin. *Battery Holder, EZ-Coin, 2032 x 1, SMD*. URL: https://be.farnell.com/harwin/s8421-45r/battery-holder-2032-smd/dp/3517387?st=battery%5C%20holder%5C%202032. Last accessed on April 24, 2021.

[54] TXC. *SMD glass sealing crystals datasheet*. URL: http://www.farnell.com/datasheets/1639143.pdf. Last accessed on April 24, 2021.

[55] TXC. *AV-24.000MAGV-T*. URL: https://be.farnell.com/txc/av-24-000magv-t/crystal-24mhz-8pf-3-2-x-2-5mm/dp/2142562?ost=av-24.000magv-t. Last accessed on April 24, 2021.

[56] Wikipedia. *Return loss*. URL: https://en.wikipedia.org/wiki/Return_loss. Last accessed on April 24, 2021.

[57] Wikipedia. *Standing wave ratio*. URL: https://en.wikipedia.org/wiki/Standing_wave_ratio. Last accessed on April 24, 2021.

[58] HBR. *Peak gain*. URL: https://halberdbastion.com/resources/academic/rf-parameters/peak-gain. Last accessed on April 24, 2021.

[59] Antenova. *A5839H datasheet*. URL: https://www.antenova.com/wp-content/uploads/2021/03/Rufa-A5839H-A5887H-PS-1.1.pdf. Last accessed on May 2, 2021.

[60] Linx technologies. *ANT-DB1-nSP250-T datasheet*. URL: https://www.mouser.be/datasheet/2/238/ant-db1-nSP250-ds-1601355.pdf. Last accessed on April 24, 2021.

[61] Taiyo Yuden. *AH316M245001-T datasheet*. URL: https://www.mouser.be/datasheet/2/396/chipantenna01_e-1511309.pdf. Last accessed on April 24, 2021.

[62] Antenova. *A5839H presentation*. URL: https://www.mouser.be/ProductDetail/Antenova/A5839H?qs=T3oQrply3y%5C%252BEZ3VQ4VzeWg%5C%3D%5C%3D. Last accessed on April 24, 2021.

[63] Texas Instruments. *LAUNCHXL-CC2640R2 presentation*. URL: https://www.ti.com/tool/LAUNCHXL-CC2640R2. Last accessed on April 24, 2021.

[64] Texas Instruments. *PCA9306 presentation*. URL: https://www.mouser.be/ProductDetail/Texas-Instruments/PCA9306YZTR?qs=g5raLcSarOb57NqMRs3HLA==. Last accessed on April 24, 2021.

[65] Texas Instruments. *PCA9306 datasheet*. URL: https://www.ti.com/document-viewer/PCA9306/datasheet/features-scps1133700#SCPS1133700. Last accessed on April 27, 2021.

[66] jluciani. *Is there a correct resistance value for I2C pull-up resistors?* URL: https://electronics.stackexchange.com/questions/1849/is-there-a-correct-resistance-value-for-i2c-pull-up-resistors. Last accessed on April 27, 2021.

[67] HyperPhysics. *Thermal Conductivity*. URL: http://hyperphysics.phy-astr.gsu.edu/hbase/Tables/thrcn.html. Last accessed on April 28, 2021.

[68] Wikipedia. *FR-4*. URL: https://en.wikipedia.org/wiki/FR-4. Last accessed on April 28, 2021.

[69] Texas Instruments. *CC2640R2LRHBR presentation*. URL: https://www.ti.com/store/ti/en/p/product/?p=CC2640R2LRHBR. Last accessed on April 30, 2021.

[70] Wikipedia. *I²C*. URL: https://en.wikipedia.org/wiki/I%5C%C2%5C%B2C. Last accessed on May 2, 2021.

[71] How to Mechatronics. *How I2C Communication Works and How To Use It with Arduino*. URL: https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/. Last accessed on May 2, 2021.

[72] Texas Instruments. *SimpleLink Academy: BLE connections*. URL: https://dev.ti.com/tirex/explore/node?node=ADUitgeplCXSPR8jxiDeKw__krol.2c__LATEST. Last accessed on May 3, 2021.

[73] Texas Instruments. *SimpleLink Academy: BLE custom profiles*. URL: https://dev.ti.com/tirex/explore/node?node=AALRuBvnOq9sws.aXdTEQw__krol.2c__LATEST. Last accessed on May 3, 2021.

[74] Bluetooth. *Specifications: assigned numbers*. URL: https://www.bluetooth.com/specifications/assigned-numbers/. Last accessed on May 3, 2021.

[75] Texas Instruments. *SimpleLink Academy: Project Zero*. URL: https://dev.ti.com/tirex/explore/node?node=AN0x5envFVo6EJ.aMpS5HA__krol.2c__LATEST. Last accessed on May 4, 2021.

[76] Texas Instruments. *Running the SDK on Custom Boards*. URL: https://dev.ti.com/tirex/content/simplelink_cc2640r2_sdk_1_50_00_58/docs/blestack/ble_user_guide/html/ble-stack-3.x/index.html#running-the-sdk-on-custom-boards. Last accessed on May 26, 2021.

[77] Texas Instruments. *TI-RTOS Drivers*. URL: https://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/tirtos/2_20_00_06/exports/tirtos_full_2_20_00_06/products/tidrivers_full_2_20_00_08/docs/doxygen/html/index.html. Last accessed on May 5, 2021.

[78] Texas Instruments. *I2C Driver*. URL: https://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/tirtos/2_20_00_06/exports/tirtos_full_2_20_00_06/products/tidrivers_full_2_20_00_08/docs/doxygen/html/_i2_c_8h.html. Last accessed on May 5, 2021.

[79] Texas Instruments. *PIN Driver*. URL: https://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/tirtos/2_20_00_06/exports/tirtos_full_2_20_00_06/products/tidrivers_full_2_20_00_08/docs/doxygen/html/_p_i_n_8h.html. Last accessed on May 5, 2021.

[80] codingjeremy. *Android BluetoothLeGatt Sample*. URL: https://github.com/googlearchive/android-BluetoothLeGatt. Last accessed on May 10, 2021.

[81] jjoe64. *Chart and Graph Library for Android*. URL: https://github.com/jjoe64/GraphView. Last accessed on May 10, 2021.

[82] Richard E. Klabunde. *Determining Heart Rate from the Electrocardiogram*. URL: https://www.cvphysiology.com/Arrhythmias/A020. Last accessed on May 10, 2021.

[83] A. B. Slama et al. "Application of statistical features and multilayer neural network to automatic diagnosis of arrhythmia by ECG signals." In: *Metrology and Measurement Systems* 25.1 (2018). DOI: 10.24425/118163.

[84] T. Tamura. "Current progress of photoplethysmography and SPO2 for health monitoring." In: *Biomedical Engineering Letters* 9.1 (2019), pp. 21–36. DOI: 10.1007/s13534-019-00097-w.

[85] John G. Webster. *Design of Pulse Oximeters*. CRC Press, 1997, pp. 40–51. ISBN: 978-0750304672.

[86] S. Andruschenko et al. "Novel Algorithm for "AC/DC" Ratio Calculation of Pulse Signals using Energy criteria of the Pulse." In: *Biomed Tech* 55.1 (2010). DOI: 10.1515/BMT.2010.675.

[87] Jan Iglesias Morales. *CC2640R2L: Can see custom board advertising but unable to connect to it*. URL: https://e2e.ti.com/support/wireless-connectivity/bluetooth-group/bluetooth/f/bluetooth-forum/998034/cc2640r2l-can-see-custom-board-advertising-but-unable-to-connect-to-it/3687830#3687830. Last accessed on May 12, 2021.

[88] Texas Instruments. *Using 32-kHz Crystal-Less Mode*. URL: https://dev.ti.com/tirex/content/simplelink_cc2640r2_sdk_1_50_00_58/docs/blestack/ble_user_guide/html/ble-stack-3.x/custom-hardware.html#using-32-khz-crystal-less-mode. Last accessed on May 12, 2021.

[89] Wikipedia. *Received signal strength indication*. URL: https://en.wikipedia.org/wiki/Received_signal_strength_indication. Last accessed on May 12, 2021.