



POLITECNICO DI TORINO  
DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATIONS

# 3D Printed Instrumented Packaging for Implantable Devices

*Author:*  
Cristina Gentili

*Supervisor:*  
Danilo Demarchi

*Co-Supervisors:*  
Timothy Costandinou  
Dorian Haci

A thesis submitted in fulfillment of the requirements for the degree of  
*MSc Electronic Engineering*  
in the  
*Centre of Bio-Inspired Technology*  
*Imperial College London*

3 April 2020

## Acknowledgements

Before starting, I would like to express my most sincere gratitude towards all the people who sustain me during my study period.

I would like to thank professor Danilo Demarchi, professor Timothy Constandinou and Dorian Haci to have given me the opportunity to work for my master thesis project in the Centre of Bio-Inspired Technology of the Imperial College of London. In these last six months, I have learnt more than what I could expect and I dedicated myself in a very interesting project which, after several efforts, gave me many satisfactions. Thank you for trusting and for having followed me during my thesis work.

I would like also to thank all the researches and the PhD students of the Centre of Bio-inspired Technology. You are a very amazing group and I am grateful to have spent six months with you. In particular, my most sincere thanks are for Daryl and Bryan. You are very beautiful people and I have learnt a lot from you. You have dedicated your time in teaching me tasks I have never done before and I am so grateful to you. Many other thanks are for Daryl who also helped me in always being motivated and in never losing confidence in myself even when i felt alone and nothing was going well. Thank you. I hope to see you again in future.

My greatest thanks are for my family. Thank you for always supporting me and for making me the person who today I am. Thank you dad for your always present support and simply thanks for everything you do and for the person you are. You are my anchor and my first example to follow in life. In everything I did and in every success I achieved, I have always tried to make you proud of me. Thank you mum to have taught me the true meaning of to never give up. Every day you show to everyone what is the willpower. You, more than everyone else, are the example of how powerful is to believe that everything will be better, without never losing hope. Despite the appearances, you are the strongest woman I have ever met. Thank to my brother Niccolò, too. Thanks and sorry for having had to stand my hysteria. However, I'd like to let you know that I will be always present for you and that I love you, even if we never say.

I would like also to thank all the people who have spent time with me during these last two years in Turin: my friends, my colleagues and my housemates. Despite the hard study, I spent fantastic moments with you and I had fun just by doing very simple things like watching a film or cooking something together. I am lucky to have met friends like you. Thanks for these unforgettable two years.

At last, with a little awkwardness, I thank Valerio. Thank you for staying with me and for never making me to feel alone, despite the distance. Only you have been able to give me light-heartedness and serenity, even in the most stressful and hardest moments, when everything else was going wrong. When I am with you, I feel like I'm at home.

## **Abstract**

Active Implantable Medical Devices (AIMDs) are essentially complex electronic systems which have to be surgically introduced inside the human body for diagnostic or therapeutic purposes. To ensure a good coexistence between the implant and the biological organism, a key role is played by the package. This thesis project deals with the design of a 3D printed polymeric package for medical electronic implants. The package is in direct contact with the living body environment and for this purpose, it must be hermetic and biocompatible. Since any living body implant site constitutes a wet environment, it is fundamental to guarantee a good moisture barrier to protect the electronics inside the cavity and to ensure safety for the patient. To fabricate an implantable device encapsulation, typical materials adopted are titanium alloys, glass or ceramics. In recent years, the idea to exploit polymers for these tasks has become more and more attractive since polymers give the opportunity to extend the potentiality of an implant, bringing lots of benefits and lowering the manufacturing cost. Continuing along these lines, 3D printing additive technology has been exploited to fabricate a low cost polymeric package. Since the greatest challenge for a polymeric encapsulation is to ensure hermeticity, an in-situ humidity monitoring system has been developed to test its resistance against moisture ingress. Finally, accelerated life-time tests have been conducted to verify package suitability.

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Motivations . . . . .	7
1.2	Objectives . . . . .	8
1.3	Report Outline . . . . .	8
<b>2</b>	<b>Background</b>	<b>10</b>
2.1	Medical electronic implants . . . . .	10
2.1.1	Requirements for medical electronic implants . . . . .	11
2.2	Package for AIMDs . . . . .	12
2.2.1	Package Requirements . . . . .	14
2.2.2	Typical Materials for Encapsulation . . . . .	16
2.3	Measurement of Hermeticity . . . . .	17
2.3.1	Hermeticity test methods . . . . .	18
2.3.2	Humidity measurement . . . . .	18
2.3.3	Humidity measurement methods . . . . .	20
<b>3</b>	<b>Humidity Monitoring System</b>	<b>21</b>
3.1	System Specifications . . . . .	21
3.2	Design Flow . . . . .	22
3.3	Humidity Monitoring System 1 . . . . .	22
3.3.1	Components . . . . .	22
3.3.2	Schematic and PCB Layout . . . . .	24
3.4	Humidity Monitoring System 2 . . . . .	25
3.4.1	Components . . . . .	25
3.4.2	Schematic and PCB Layout . . . . .	28
<b>4</b>	<b>Firmware development</b>	<b>29</b>
4.1	Inter Integrated Circuit (I2C) serial bus protocol . . . . .	30
4.1.1	Nordic TWI - Two Wire Interface . . . . .	31
4.2	Bluetooth Low Energy (BLE) protocol . . . . .	34
4.2.1	Nordic BLE application . . . . .	34
4.2.2	Generic Access Profile (GAP) . . . . .	36
4.2.3	Generic Attribute Profile (GATT) . . . . .	39
4.2.4	ATT - Attribute Protocol . . . . .	42
4.3	Results . . . . .	45
4.3.1	Nordic nRF52 Development Board and SHTW2 sensor . . . . .	45
4.3.2	Custom PCAs Serial Wire Debug (SWD) . . . . .	48
<b>5</b>	<b>3D Printed Package</b>	<b>54</b>
5.1	Figure 4 Standalone Printer . . . . .	54
5.2	DLP - Digital Light Processing . . . . .	54
5.3	Material . . . . .	55
5.3.1	Figure 4 MED-AMB 10 . . . . .	55
5.4	CAD Model Design and Fabrication . . . . .	56
5.4.1	Sealing method . . . . .	57
5.4.2	Package fabrication . . . . .	58

<b>6</b>	<b>Hermeticity testing</b>	<b>59</b>
6.1	Experiment setup . . . . .	59
6.2	Test 1 . . . . .	60
6.3	Test 2 . . . . .	61
6.3.1	Estimated life-time . . . . .	62
6.3.2	Comparison between Test 1 and Test 2 . . . . .	65
6.4	Test 3 . . . . .	66
6.5	Test 4 . . . . .	66
6.5.1	Comparison between Test 1 and Test 4 . . . . .	67
6.5.2	Comparison between Test 2 and Test 4 . . . . .	68
<b>7</b>	<b>Conclusion</b>	<b>69</b>
7.1	Further work . . . . .	69
<b>A</b>	<b>Electronic Components</b>	<b>71</b>
A.1	SHTW2 . . . . .	71
A.2	ISP1607-AX . . . . .	72
A.3	nRF52832 . . . . .	73
A.4	TPS82740A . . . . .	74
A.5	LEDs . . . . .	75
A.6	SHTC3 . . . . .	76
A.7	BL652-SA . . . . .	77
A.8	LMZ21700 . . . . .	78
A.9	TXS0104E . . . . .	80
<b>B</b>	<b>Schematics</b>	<b>81</b>
<b>C</b>	<b>PCBs 3D views</b>	<b>84</b>
	<b>Bibliography</b>	<b>89</b>

# List of Figures

2.1	Active implant scheme. . . . .	10
2.2	Active implant basic features and requirements [1]. . . . .	11
2.3	Final result of the Foreign Body Reaction [1]. . . . .	13
2.4	Helium permeability chart for common packaging materials [2]. . . . .	15
2.5	Polymeric encapsulation leak rate problems. . . . .	17
2.6	Closed chamber with liquid water and vapor [3]. . . . .	19
2.7	Saturated Vapour Pressure $e_w(t)$ [3] . . . . .	19
3.1	Humidity monitoring system block diagram . . . . .	21
3.2	Typical and maximal tolerance for the temperature measure in °C [4]. . . . .	23
3.3	Relative humidity measures tolerances . . . . .	23
3.4	3.7 V Lithium Polymer Battery . . . . .	24
3.5	Composite drawings - Top and Bottom layers of Humidity Monitoring System 1. . . . .	25
3.6	Typical accuracy for relative humidity (in %RH) and for temperature measurements (in °C) [5]. . . . .	27
3.7	Composite drawings - Top and Bottom layers of Humidity Monitoring System 2. . . . .	28
4.1	nRF52 DK (Copyright © Nordic Semiconductor) . . . . .	29
4.2	nRF5x SDK Modules [6]. . . . .	30
4.3	Example of a I2C communication between a master and three peripherals. . . . .	30
4.4	Example of a BLE topology. . . . .	34
4.5	Nordic BLE applications stack [6]. . . . .	35
4.6	Softdevice S132 stack scheme [7]. . . . .	35
4.7	GAP state diagram. . . . .	36
4.8	Advertising sequence [8]. . . . .	36
4.9	Scanner and Advertiser operations [8]. . . . .	37
4.10	BLE Connection initialization [9]. . . . .	38
4.11	Configuration of the profile, its service and characteristics that are going to be programmed. . . . .	40
4.12	Attribute table to define for the custom profile. . . . .	43
4.13	SHTW2 humidity sensor used to test the I2C program by the Nordic nRF52 DK board. . . . .	46
4.14	Schematic of the components connections . . . . .	46
4.15	SHTW2 profile with one service and one characteristic, view from <i>Light Blue</i> iPhone application. . . . .	47
4.16	Example of 4-byte characteristic value in hexadecimal format representing the outputs from the sensor for relative humidity and temperature measures. . . . .	47
4.17	Sensor measurements sent by Bluetooth to the iPhone App. . . . .	48
4.18	Elaborated values from the SHTW2 outputs. . . . .	48
4.19	Serial Wire Debug (SWD) schematic. . . . .	49
4.20	SHTW2 profile with one service and five characteristics, view from <i>LightBlue</i> iPhone Application. . . . .	49
4.21	Humidity Monitoring System 1: Elaborated values from the 5 SHTW2 sensors outputs. . . . .	50
4.22	SHTC3 profile with one service and three characteristics set up. . . . .	50
4.23	Bluetooth Connection with SHTC3 device, by <i>nRF Connect</i> . . . . .	51
4.24	Humidity Monitoring System 2: Elaborated values from the 3 SHTC3 sensors outputs. . . . .	51
4.25	Current consumption scope views. . . . .	52

LIST OF FIGURES

---

5.1	Digital Light Processing explanation drawing [10]. . . . .	55
5.2	Package CAD Model . . . . .	56
5.3	O-ring slot . . . . .	57
5.4	Photograph of package internal parts with PCB and o-ring. . . . .	58
5.5	Photograph of the final sealed package with PCB and battery inside . . . . .	58
6.1	Experiment set-up inside <i>ESPEC</i> SH-221 humidity chamber. . . . .	59
6.2	Experiment 1: elaborated values from the 3 SHTC3 sensors outputs. . . . .	60
6.3	Connection between Nordic nRF52DK board (master) and the custom designed PCB module (peripheral). . . . .	61
6.4	Experiment 2: elaborated values from the 3 SHTC3 sensors outputs. . . . .	62
6.5	Experiment 2: fitting function and experimental model of the relative humidity evolution. . . . .	63
6.6	Calculated relative humidity evolution model over time. . . . .	64
6.7	Mean relative humidity - limit at 80% and limit at 70%. . . . .	65
6.8	Comparison between Test 1 and Test 2. . . . .	65
6.9	Experiment 3: elaborated values from the 3 SHTC3 sensors outputs. . . . .	66
6.10	Experiment 3: elaborated values from the 3 SHTC3 sensors outputs. . . . .	67
6.11	Comparison between Test 1 and Test 4. . . . .	67
6.12	Comparison between Test 2 and Test 4. . . . .	68
A.1	SHTW2 Humidity and Temperature sensor from Sensirion [4] . . . . .	71
A.2	Functional block diagram of SHTW2 [4] . . . . .	72
A.3	ISP1507 [11] . . . . .	72
A.4	ISP1507 Block diagram [11] . . . . .	73
A.5	nRF52832 module description [12]. . . . .	73
A.6	TPS82740A reference illustration . . . . .	74
A.7	TPS82740A Parameter Measurement Information [13] . . . . .	74
A.8	TPS82740A Pin configuration - Top and Bottom views [13] . . . . .	74
A.9	LEDs representation . . . . .	76
A.10	SHTC3 Humidity and Temperature sensor from Sensirion [5] . . . . .	76
A.11	Functional block diagram of SHTC3 [5] . . . . .	77
A.12	Laird BL652-SA module . . . . .	77
A.13	BL652-SA Block diagram [14] . . . . .	78
A.14	LMZ21700 in SIL0008E Package [15] . . . . .	78
A.15	LMZ21700 simplified schematic [15]. . . . .	79
B.1	Humidity Monitoring System 1 -I2C sensors communication Schematic . . . . .	81
B.2	Humidity Monitoring System 1 Complete Schematic . . . . .	82
B.3	Humidity Monitoring System 2 Complete Schematic . . . . .	83
C.1	Humidity Monitoring System 1 - PCB 3D Top and Bottom views from Altium Designer 19.1.8 . . . . .	84
C.2	Humidity Monitoring System 2 - PCB 3D Top and Bottom views from Altium Designer 19.1.8 . . . . .	85

# List of Tables

3.1	SHTW2 I2C Address [4]. . . . .	22
3.2	SHTW2 I2C commands [4]. . . . .	22
3.3	SHTW2 Soft Reset and Read ID commands [4]. . . . .	22
3.4	SHTC3 I2C Address [5]. . . . .	25
3.5	SHTC3 I2C commands [5]. . . . .	26
3.6	SHTC3 Sleep and Wakeup commands [5]. . . . .	26
3.7	Components values for VOUT 1.8 V [15]. . . . .	27
4.1	I2C parameters selection - nRF52 DK programming . . . . .	32
4.2	I2C parameters selection - Humidity monitoring system 1 programming . . . . .	32
4.3	I2C parameters selection - Humidity monitoring system 2 programming . . . . .	32
5.1	Liquid Figure 4 MED-AMB 10 material properties, from [16]. . . . .	56
5.2	Mechanical Properties of the Post-Cured Figure 4 MED-AMB 10 material, from [16].	56
5.3	O-ring slot parameters . . . . .	57
6.1	Estimated parameters of the linear regression models for relative humidity evolution.	63
A.1	SHTW2 pin assignment [4] . . . . .	71
A.2	TPS82740A Pin names [13] . . . . .	75
A.3	Output Voltage Setting TPS82740A [13]. Output voltage can be selected by the user by means of three voltage select pins (VSEL), within a range from 1.8 V to 2.5 V. . . . .	75
A.4	LEDs key features . . . . .	76
A.5	SHTC3 pin assignment [5] . . . . .	77
A.6	LMZ21700 Pin names [13] . . . . .	79
A.7	TXS0104ED Pin Functions [17]. . . . .	80

# Chapter 1

## Introduction

### 1.1 Motivations

In recent years implantable electronics has assumed more and more relevance in medicine. Living in an aging society in which chronic diseases, such as arthritis, diabetes, epilepsy, heart disease... are increasing, electronic diagnostic and therapeutic systems that can be implanted inside human body to perform a constant monitoring action, in most situations, can truly save patients or, in any case, preserve quality of their life [18]. These medical devices have become possible especially thanks to electronics advances in keeping reliability in harsh environments [19]. Human body organism is a very critical site and, obviously, the medical device must not damage body tissue, but, since each biological system tries always to reject a foreign material, it's fundamental for the implant to survive to the attack, as well. Device package assumes a true key role in these applications since it constitutes the implant part which enters in direct contact with the biological tissue. Therefore, it has to satisfy many essential requirements, more than the ones assumed by standard ICs packages, such as biocompatibility, biostability and so on.

Typical materials adopted for implantable medical devices packages are titanium, glass and ceramics. These materials benefit of good reliability properties since they are able to provide a good moisture barrier to prevent condensation and they are proved to be long-term biocompatible. Indeed, package has to be mechanically robust to withstand stress, flexion and wear, but it has to be smooth and soft, as well, to not damage human body organism [19]. Anyway, packages manufacturing cost is high; the production processes are complex, expensive and generally difficult to reproduce without confiding in specific professional companies.

The core objective of the project developed for the master thesis is to investigate additive manufacturing for hermetic packaging of active implantable medical devices, in order to achieve a very low manufacturing cost and to make implantable device package fabrication as much as possible easy and affordable. Even if medical devices market is general an high level market, since high cost should not be a problem for systems saving human life or, in any case, safeguarding patients health, the importance of lowering cost and of making the production easier is due to give the possibility to every research groups to easy fabricate a package structure by themselves and to make the overall medical device testing procedure easier, cheaper and faster. As known, all new medical devices must follow a strict approval path before entering in the market: a lot of tests have to be performed not only to verify the correct device behaviour, but, most important, to ensure safety for the patient. Because of this, all implantable devices have to be tested before in-vitro and, then, they have to be implanted in-vivo in mice to study their biological responses. Thus, a low cost implantable device package may be exploited for testing purposes to be implanted in animals for a pre-clinical working step. This will make the overall testing procedure cheaper, but also faster.

Therefore, to achieve this goal, this thesis work suggests to fabricate a polymeric package for active implantable devices exploiting 3D printing manufacturing technology.

First of all, polymeric packages can offer many advantages because of the ease of fabrication, low cost, feasibility, flexibility, low density, RF-transparency and biocompatibility [20, 21]. However, a polymeric encapsulation is not able to adequately provide a barrier against moisture and this make it to fail before typical clinical procedures. Anyway, it is possible to exploit it for pre-clinical

experiment works, just assuring that it is able to give a proper protection for at least weeks or months [21]. A typical fabrication technique to build proper polymeric packages is based on an epoxy casting process. If from one hand polymers are very low cost materials, from the other hand this technological process is quite complex: a fully epoxy-encapsulated device is not easy to produce and it is made by too much steps, all to be carefully performed.

Consequently, the possibility to exploit 3D printing to realize this kind of structures is much attractive. Generally, 3D printed objects suffer from high porosity, lack of adhesion between layers and low surface quality that make them not proper suitable for implantable devices purposes. Anyway, 3D printing technology was subject to a large amount of evolutions in the last years: the multiple building mechanisms in which it is based on have been improved and new materials have been developed. In conclusion, the idea is to exploit this easy manufacturing process to realize testing implantable devices packages in a very easy, cheap, fast and affordable way.

## 1.2 Objectives

This master thesis main objective is to build an implantable device package by means of 3D printing additive technology. The 3D printed manufacturing technique adopted is the, so called, Digital Light Processing (DLP) which works with polymerized resins that are able to solidify once illuminated by a light source. Thus, the aim is to print a polymeric package which satisfies all the requisites properly set for implantable devices encapsulations. Concerning the 3D printer that is available in the Centre of Bio-Inspired Technology laboratory of the Imperial College of London, one of the material that can be exploited is declared as biocompatible, biostable and it can also be sterilized by autoclave. All of these characteristics make it attractive in exploiting it for medical purposes. Anyway, if from one hand that polymeric encapsulation brings the advantage to be extremely low cost and very easy and to build, from the other hand, generally, it reveals to be permeable to most of gases, including water vapor, and it makes the printed structures not suitable for long-term implants. For that reason, the greatest challenge to face in this project is to satisfy the hermeticity requirement. To summarise, all of the following working steps have been developed:

- At first, it has been designed an humidity monitoring system that it has to be placed inside the package cavity. This electronic chip should be able to read over time humidity and temperature values during an accelerated testing procedure performed to understand how much the moisture is let to penetrate inside the encapsulation. To develop this in-situ humidity monitoring system, at first all electronic components have been selected, then it has been designed a schematic diagram, finally the PCB layout has been completed and all the needed PCBs have been manufactured and properly assembled.
- The second step treats of firmware implementation. The aim is to to rightly program the microcontroller unit to read measures from the humidity sensors and to trasmit them to the external via a Bluetooth wireless communication.
- Once the monitoring system was ready, it has been designed the package under test following some basic specifications and finding a good hermetic sealing procedure to properly close the package. In the end, the final structure has been fabricated by means of *Figure 4 Standalone* printer.
- The last step deals with the set-up of an accelerated life time test to verify if the hermeticity requirement has been satisfied as much as is necessary for a long-term implantation.

## 1.3 Report Outline

- **Chapter 2: Background**

As the title suggests, this chapter briefly explains all the background knowledges needed to understand and to develop the working steps implemented during this thesis project. At first, medical implantable devices are introduced, characterizing their basic functionalities, working principles, features and requirements. Then, the importance of active implantable medical devices packages is highlighted, focusing on main challenges and requisites. Since the

primary objective is to test package hermeticity, a review about hermeticity testing methods is executed, describing physics of humidity and types of humidity measurements, as well.

- **Chapter 3: Humidity Monitoring System**

This chapter deals with the first phase of thesis work, that is the design of an humidity monitoring system. The idea is to develop a small printed circuit board which should be placed inside the package cavity to monitor temperature and the humidity over time. The measurements are thought to be sent to the user via Bluetooth wireless communication. Hence, to this purpose, first of all a careful choice of electronic components has to be done, next it has to be designed a proper schematic diagram and, in the end, the PCB layout. Indeed, two different PCBs have been developed since in the first one several complications arose from the soldering step.

- **Chapter 4: Firmware development**

Once that PCB has been assembled, the microcontroller unit has to be programmed in order to properly execute the application function. The aim is to write a C code which tells to the microcontroller to collect data from the humidity sensors and to transmit these measurements via Bluetooth. Chapter 4 can be divided in three parts: the first one deals with the I2C communication between the microcontroller and the sensors, the second talks about Bluetooth Low Energy wireless protocol and the last one shows all the results that have been obtained by testing the code on Nordic nRF52DK board and on the printed circuit assembled boards that have been developed.

- **Chapter 5: 3D Printed Package**

At this point, the package structure has to be manufactured. The technological process to exploit is 3D printing additive technology based on Digital Light Processing (DLP) method. In this chapter, it has been described Figure 4 Standalone which is the 3D printer available in the Centre of Bio-Inspired Technology of the Imperial College of London, its printing method (DLP) and it has been described the adopted material. Then, the package CAD design has been shown and, finally, it has been defined the sealing method adopted to close the package.

- **Chapter 6: Hermeticity testing**

In the end, accelerated tests have been conducted. The chapter describes the experiment set-up, how the tests have been executed and the obtained results have been reported. More exactly, four tests have been performed and they have all been compared to each others discussing and analysing their final outcomes.

- **Chapter 7: Conclusion**

The last chapter summarises all the developed working steps and discusses the final results that have been obtained. Finally, possible further improvements have been proposed for possible future works.

# Chapter 2

## Background

### 2.1 Medical electronic implants

From 1959, when the first pacemaker was implanted, the interest and the research in implantable devices have always led to more powerful and smarter systems which have assumed, in the years, more and more functions and utilities. In fact, nowadays, implantable medical devices are in-vivo tools employed in multiple tasks: to replace or sustain a missing biological structure [22], to monitor, measure and solicit physiological responses [23] or to release drugs. Examples of implantable medical devices are: cardiac pacemakers, retinal implants, brain-computer interfaces, cochlear implants, drug delivery implants ... and so on.

Indeed, implantable devices (but, more in general, all medical devices, as well) can be divided in two classes: passive and active ones. A passive implantable medical device is, for instance, an hip implant, a dental implant or a stent, and it does a fixed function such as just mechanical sustain. An active implant, instead, is a much smarter device since it performs a specific function depending on possible contingencies.

*‘Active medical device’ means any medical device relying for its functioning on a source of electrical energy or any source of power other than that directly generated by the human body or gravity.*

This is the definition given for Active Medical Device by Food and Drug Administration (FDA) Directive 90/385/EEC. In few words, it says that an active medical device needs electronics. The same directive defines also an Active Implantable Medical Device (AIMD) as *any active medical device which is intended to be totally or partially introduced, surgically or medically, into the human body or by medical intervention into a natural orifice, and which is intended to remain after the procedure.*

Active implantable medical devices are complex systems under extremely tight design constraints [24]. Requirements for implants involve general common design rules and principles, but application-specific requirements, as well [24]. For instance, the device specific function, sensing mechanism, placement in the body, implantation time... and so on, have all to be taken into account in the whole design phase.



Figure 2.1: Active implant scheme.

Going deeper toward a functional classification, there are two main types of Active Implantable Medical Devices: Diagnostics AIMDs and Therapeutic ones [25]. Diagnostic AIMDs have to monitor a physiological signal, such as blood pressure, blood oxygen level, glucose level, inner ocular pressure, electrocardiograph and so on. Most of them don't need electrical interfaces to the tissue [25]. Therapeutic AIMDs, instead, have to do an active treatment [25]. Examples are cardiac pacemaker, neurostimulators, neurorecorders, neuromodulators... and so on. Most of them need electrical interfaces to the tissue.

Anyway, a typical AIMD working principle can be sketched by the following main blocks [1] (figure 2.1):

- **Sensing:** physiological signals are captured by sensors.
- **Intelligence:** the electronics elaborates and analyzes the electrical input signals, coming from the sensing block.
- **Actuation:** the device performs a certain action depending on the elaborated informations which should properly describe the human body conditions.

### 2.1.1 Requirements for medical electronic implants

In figure 2.2 a basic AIMD illustration with essential features and requirements is depicted. Here, it is possible to observe the main blocks explained in figure 2.1, as well. Indeed, the sensing block are two: one is sensible to a physiological signal coming from the human tissue, while the other is able to catch an electric input signal. The intelligence works by means of electronics and it elaborates all the inputs to obtain informations about the human body condition. It exploits the telemetry unit, used to transmit the collected data to the external remote controller device or needed, for instance, to catch the energy coming from the external to recharge the battery. Lastly, there are depicted two actuation blocks: one makes a specific action toward the tissue depending on the elaborated information, while the other is responsible for sending an electrical output contingent on the human body state.

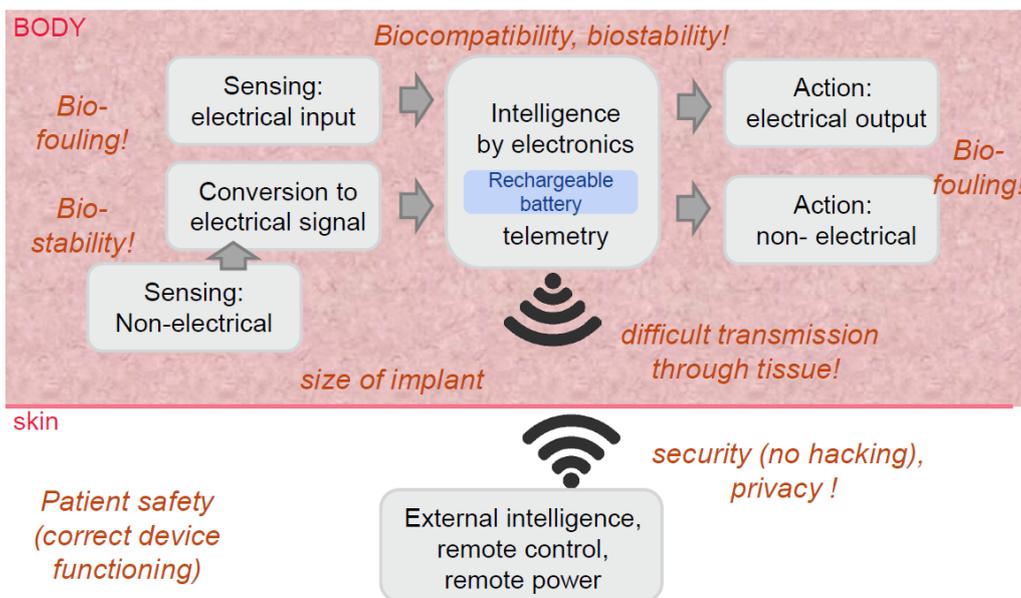


Figure 2.2: Active implant basic features and requirements [1].

The requirements for an active implantable medical device can be divided in four main categories: the functional requirements, the environmental requirements, the specific requirements for superior device-tissue interaction and the risk assessment based requirements [1].

### 1. **Functional requirement**

These are about the basic functionality of the device. For instance, the sensing unit requires sensitivity; the intelligence unit requires computational power and low energy consumption. Furthermore, the telemetry for wireless devices needs an antenna for transmission of signals or energy [1].

### 2. **Environmental requirements**

These are very important specifications, since an electronic device is put inside an human body and this will always cause some consequences. The device under test and the host tissue must coexist without causing damages to each other. These requirements mostly regard the package of the implantable device, therefore they are the most interesting requisites for the current thesis project. By looking at figure 2.2, we can immediately list some of package requisites: no bio-fouling, bio-stability, biocompatibility,.. and so on. Actually, these requirements deeply depend both on the duration of the implant and on the specific part of the body where the devices should be placed [1].

### 3. **Specific requirements for superior device-tissue interaction:**

As the name suggests, these requirements are specific for some applications regarding particular interactions. For instance, depending on the tissue where the implant should be placed, one requirement can be miniaturization or, maybe, to develop a fitting flexible device [1].

### 4. **Risk assessment based requirements**

These requisites are strongly dependent on individual devices: each device has its risk assessment test to be done and for each device it is fundamental to take arrangements to avoid these risks [1].

## 2.2 Package for AIMDs

When an implant is performed, the interaction between the medical device and the living organism is very critical. Since, the electronics, which is the heart of the AIMD, can't interface directly with the human body environment, the package, which is generally required for any IC to give mechanical support, protection, heat dissipation and so on, now is getting more and more significant, since it is the true responsible for a good coexistence.

Anyway, the purposes of packaging in medical devices which are the same as ICs, are [26]:

- **Support:** to provide a mechanical support.
- **Protection:** to protect the electronics from the environment (dust, stress, shock, moisture...).
- **Interfacing:** to bring signals in and out of the chip.
- **Heat removal:** to ensure the heat generated inside the chip is properly expired to the environment.

However, differently from ICs packages, depending on its functionality, its place on the body, the duration of the implant ..., the package for a specific device should be different. In fact, as it is done for MEMS, it must always be built ad-hoc for each particular application. Moreover, it is obvious that the challenges brought by AIMDs packages are different from those encountered by IC packaging.

First of all, the human body environment is very harsh. It's very corrosive: the body solution is a highly oxygenated saline electrolyte of a pH of around 7.4 and a temperature of 37 °C [27]. An external object has to be put inside a body and it must not only avoid damaging the hosting organism, but also avoid being damaged by the biological system. Dealing with the interaction from the implant to the tissue, it is necessary that the implantable device is hermetic and biocompatible. This means: no diffusion of toxic materials toward the tissue and a faint tissue reaction upon implantation [1]. On the other hand, the tissue will automatically try to destroy the implant and, because of this, the implant should be hermetic and biostable. It is necessary to prevent the leaching of body fluids into the implant, the chemical degradation (i.e. oxidation) and the adverse

response to mechanical stress (i.e. water, fatigue) [1].

The reaction that is always present when an external entity is introduced in a living organism, is called *Foreign Body Reaction* (FBR).

### Foreign Body Reaction

Foreign Body Reaction is the name given to the fact that each healthy human body tries always to destroy an extraneous material [1].

The first attack against the external device happens by means of white blood cells action; next, if the foreign survives, macrophages and FB giant cells will try to expel it. In details, they try to swallow the material and digest it by secreting super-oxides. The designed electronic device should persist after these attacks and, finally, a fibrous encapsulation is formed to insulate the resisting foreign object [1]. Thus, the FBR final effect is an encapsulation around the medical device, how it is illustrated in figure 2.3.

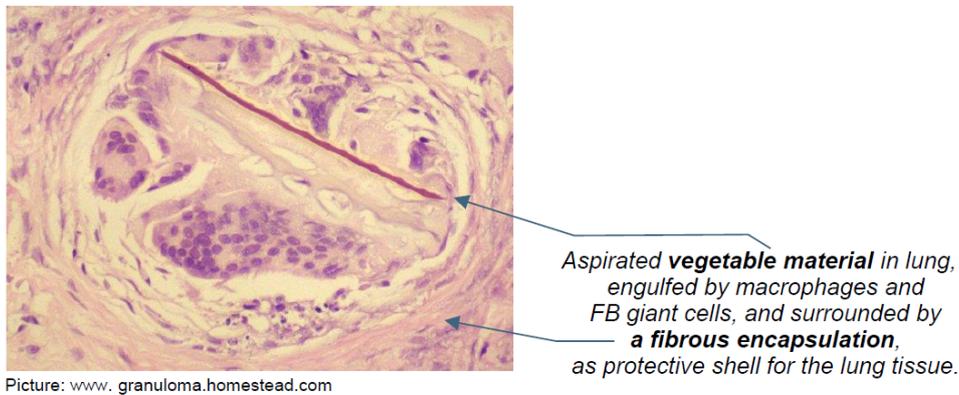


Figure 2.3: Final result of the Foreign Body Reaction [1].

As a general rule, the aim is to do everything it's possible to limit the resulting FBR encapsulation. The advantage of FBR is that the encapsulation gives mechanical anchoring and protection for the implant, but the drawbacks are in any case in majority. In fact it limits the sensors sensibility and isolates the implant, giving problems for the device in communicating with the human organism and the external environment, as well. Moreover, FB encapsulation might cause chronic pain due to chronic inflammation (e.g. when the implant loses small particles by friction or by degradation) [1].

It's not possible to completely avoid Foreign Body Reaction, but there are many factors which seem to influence it, such as [1]:

- **Package related factors**

1. Implant size and shape: it's better to have small packages and rounded shapes.
2. Micro-motion causing local irritation: it's possible to limit the micro-motion by adopting rounded and soft packages.
3. Surface chemistry: material selection, use of coatings, use of drugs...
4. Surface roughness: roughness optimization.
5. Porosity of the surface: optimum size of pores (diameter).

- **Other factors**

1. implant location (host tissue);
2. characteristics of the host (age, general health...);
3. quality of surgery;
4. ... and so on.

### 2.2.1 Package Requirements

In the following, the most important requisites and challenges about packaging for active implantable medical devices are itemized:

1. biocompatibility;
2. bio-stability;
3. suitable sterilization technique;
4. hermeticity;
5. RF transparency.

#### Biocompatibility

*Biocompatibility: the ability of a material/device to perform with an appropriate host response in a specific application* (Ratner, 2004).

Putting an external object inside a body will always cause a reaction. It can't be said that a biocompatible material do not interact with the hosting body or do not cause any response because there will be always a certain effect, for instance even just the Foreign Body Reaction.

Therefore, the correct definition of biocompatibility says that the resulting reaction should be mild and controlled. In practical words, a biomaterial must not infect or cause damage to the human organism: it mustn't cause an allergic or toxic reaction, destroy cells, tissues or enzymes, cause illness or tumors...[1].

Summing up, a biocompatible material, in a specific situation, has the ability to coexist with the hosting organism, causing an appropriate body response. Anyway, it's actually a contextual property and, following the ISO 10993 standard classification, the bio-compatibility is a function of:

- the **place** in the body (tissue, bone, blood);
- the **duration** of the contact/exposure: the typical distinction is made by three time intervals:
  - limited (<24 hours);
  - prolonged (24 hours - 30 days);
  - permanent (>30 days).
- the **type** of device contact:
  - surface (electrodes, trans-dermal delivery devices, ...);
  - external communication (urinary catheters, endoscopes, ...);
  - implants: tissue and bone/blood (bone screws, sutures, vascular stents, ...).

ISO10993 is a document describing testing for biocompatibility. Indeed, it's only a guideline providing a detailed test table. On any case, there a lot of methods for testing a material biocompatibility and it's still not easy to assert if a material is biocompatible, especially if the objective is to prove it for a long-term implant.

#### Bio-Stability

Bio-stability means that the device must stay inside the host body for the required time without degrading and without loosing functionalities [28]. In other words, materials for AIMDs packages, exposed to fluids or to biological tissues, must be unalterable in human body environment during all the period of implantation [29]. Bio-stability is strongly correlated with biocompatibility, since a biocompatible material, which degrades, becomes, actually, non-biocompatible [28].

Guideline ISO 10993-9 (Framework for identification and quantification of potential degradation products) describes bio-stability tests for implantable materials and it describes general principles followed by case specific methods explained in ISO 10993-15 (metals and alloys), ISO 10993-14 (ceramics) and ISO 10993-13 (polymers) [29, 30].

### Suitable sterilization technique

Before any surgical operation, the sterilization of everything which should be put in contact with the human body is a key requisite. The sterilization of the package surface has to be executed to kill any contaminants to not infect the host organism [29].

Sterilization techniques involve chemical or gas treatments, destruction by heat or irradiation... and so on. Obviously, the encapsulation material should sustain these aggressive treatments. Sometimes, the electronic system can't be exposed to irradiation or it can't support too high temperatures. Thus, the choice of the sterilization technique has to be taken into account before developing the whole electronic design in which the implantable device is based on [29].

### Hermeticity

Hermeticity is a very fundamental requirement for AIMDs packages since electronics needs a dry condition to maintain its reliability [31] and the human body, as it is well known, is a wet environment.

Hermeticity is critical to define, since the ideal aim is to build an impenetrable barrier against any liquids and gasses, but actually all materials are gas permeable to some degree [28]. Moisture ingress can cause open circuits, surface electrical leakages and it can corrode the metallic parts. The intrusion of some other gases, such as oxygen, can lead to failures too since, for instance, can origin solder oxidation [28].

Generally the purpose is to guarantee an appropriate moisture barrier just for all the required time of the implantation. Therefore, the question is: to obtain an hermetic encapsulation, what kind of leak rate may be acceptable?

Indeed, the purpose is not only to save the electronic chip from the human body environment, but

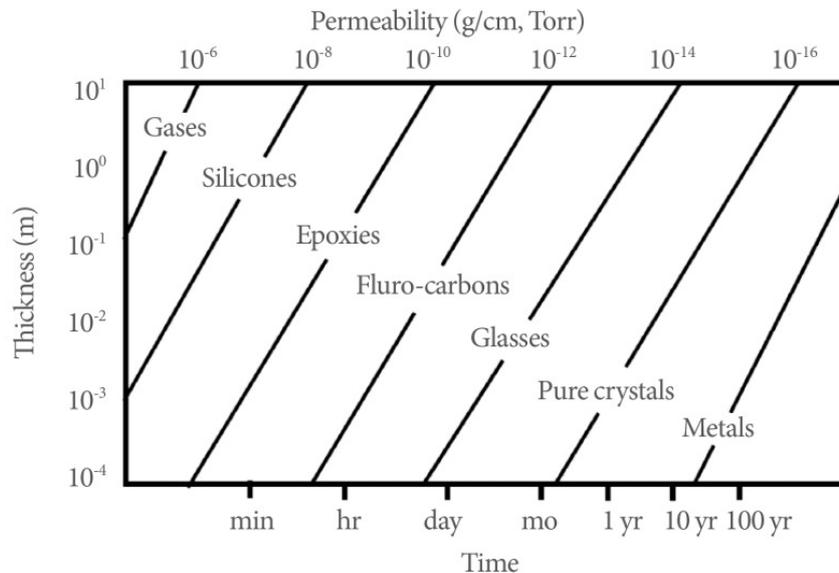


Figure 2.4: Helium permeability chart for common packaging materials [2].

also to not damage the host organism by letting penetrating in it some toxic materials. Because of this, the encapsulation must constitute a bidirectional barrier such that to isolate everything that is inside the package from the host organism.

The factors which influence the hermeticity level inside the cavity are the choice of package material, the fabrication process, the external environment and the final seal design [28].

From what concerns the material choice, the parameter to consider is permeability. Permeability is defined as the measure of how much fluids are let to penetrate through a material [32, 33]. It depends on mass, distance, time and pressure [29]. In figure 2.4 it is shown helium permeability chart for some common packages materials and their expected lifetime depending on structure thickness. As it can be noticed, polymers are not able to provide an impermeable barrier. On the contrary metals, ceramics and glass can be generally adopted for long-time medical implants [29].

### Material having RF transparency

Percutaneous wires could origin moisture ingress problems and, due to them, the risk of infection for the patient increases significantly [29].

For most implantable devices, the need to communicate via a wireless communication has become a key requirement.

Metals aren't able to provide RF transparency, as a consequence, to communicate via electromagnetic signals through free space, they need an external receiving and transmitting system placed outside the structure. This causes many problems: first of all, the dimensions increase and the implant could result too invasive for the host body. Second, the polymeric encapsulation, which often is used to protect the external antenna system, lets the humidity to enter inside the polymeric film causing a worsening of Q factor which decreases the gain of the receiving and transmitting stage [29].

Therefore, to adopt a RF transparent material, such as glass, bioceramics or hermetic polymers, is becoming more and more a necessary requirement for the fabrication of an implantable device package.

### 2.2.2 Typical Materials for Encapsulation

In few words, what is essentially needed it's a biocompatible and biostable material which is able to provide an hermetic bi-directional barrier between the implant and the human body environment. General choices are based on three main material groups: biocompatible metals, glass, ceramics and polymers.

**Metals - Titanium alloys** Biocompatible metals are largely adopted as implant package material and, as an example, the current cardiac pacemaker housing is made by titanium.

Metals provide better mechanical and stability properties with respect to other materials and a widely adopted biocompatible metal is Titanium. More exactly, the titanium alloy  $TiAl_6V_4$  (Titanium Grade 5) housing is the one generally exploited [34].

Benefits of titanium, regarding AIMDs packages requirements, are: biocompatibility, mechanical stability, low permeability to ions and water vapor and a well defined hermetic sealing procedure, such as laser welding, which ensures a true hermetic barrier [35]. It is also possible to make full use of a well established hermetic feedthroughs technology to allow electrical signals to enter and exit the package [36]. However, an important disadvantage of titanium housing is that it is not RF transparent. Due to this, it needs an optical window [35] or a communication antenna/power receiving coil to be placed outside the package [36], if it should dialogue by means of a wireless communication. This is a great drawback since an important trend for implantable medical devices (and for any electronic device, as well) is miniaturization and this forces the device to take up more space than the actual required for the electronic chip. Moreover, if a glass window was open on the surface, this means more fabrication complexity and more manufacturing cost.

Another important drawback, concerning metals and alloys, is the high corrosion risk. For instance, the most common corrosion type is the galvanic corrosion. The metal package can be subjected to an electrical potential due to the electronic activity inside the cavity. When two different metals are putted in an electrolyte (as the human body solution), an electrochemical reaction may start [36]. Actually, there are other types of corrosion involving metals and, duo to this, corrosion is one of the greatest responsible for metallic package devices failures [36].

Summing up, for reasons of cost and dimensions other types of materials, in recent years, have become more and more inviting.

**Ceramics** Instead of metals, ceramics exhibit a good RF transparency, moreover, as titanium, they can provide a good hermetic barrier. Because of its biocompatibility and good hermeticity,  $Al_2O_3$  ceramic is firmly used for implants [35]. However, the disadvantages of ceramics include: high temperature sealing process, poorer mechanical properties than metals, high risk for cracking and, to ensure hermeticity, the walls should be thicker than the corresponding metal housings walls. The greatest drawback is the ceramic intrinsic brittleness, which causes mechanical vulnerability and can bring to catastrophic failures [35]. Despite of these, when RF transparency is a key requirement, a ceramic encapsulation is generally preferred with respect to titanium, since it can provide smaller implant volumes, attenuating Foreign Body Reaction, as well [35].

**Glass** Glasses have similar properties to ceramics. They are RF transparent and they are able to provide a good hermetic barrier against moisture ingress, too. Many biocompatible glasses can be adopted for implantable devices. One example is borosilicate glass (Kimbel N51A) that is employed for neuromuscular microstimulator [28].

**Polymers** Polymers have become more and more interesting due to their multiples advantages and opportunities. First of all, they give more possibilities in choosing if to build flexible, transparent, thermal insulating, soft and small structures depending on the particular application and its specifications [1, 35]. The possibility to build soft, porous, flexible packages has advantages regarding Foreign Body Reaction, as well, and, in this way, more implantable sites can become possible due to polymers easy feasibility. Anyway, the greatest advantage of polymers encapsulation is its low cost manufacturing process.

Among numerous polymeric materials, only few of them can be used for AIMDs packages. For instance, some candidates could be: silicones, parylenes, silicon-carbons, epoxies, polyimides, silicon-carbons and liquid crystal polymers (LCPs) [36]. Obviously, they should be biocompatible, biostable and hermetic. Hermeticity of polymeric materials is very challenging because, generally, they aren't able to provide an impermeable barrier and, therefore, they aren't properly suitable for a long term implantation. In fact, polymers are gases permeable and, generally, the moisture (water vapor) is allowed to penetrate inside the encapsulation [35].

Recent studies have proved great improvements in polymers hermeticity by building multilayer structures. For instance, the deposition of alternated nano-layers of  $Al_2O_3$  and  $HfO_2$  (by Atomic Layer Deposition ALD) has brought toward very low level of Water Vapor Transmission Rate (WVTR) [1], that is an excellent result. Anyway, another important challenge about polymeric encapsulation, to keep the moisture out, is to ensure a suitable hermetic sealing technique [35]. In fact, to guarantee an hermetic seal could be critical facing with polymers.

If only polymers were demonstrated to be able to perform a well established hermetic encapsulation, the medical device industry would be transformed regarding design opportunities and volume output [35].

Therefore, the aim of this thesis work is to develop a polymeric package suitable for implantable applications, exploiting a low cost manufacturing technique. As a matter of fact, polymers give the opportunity to adopt a very low cost and easy fabrication technique that is 3D printing technology.

## 2.3 Measurement of Hermeticity

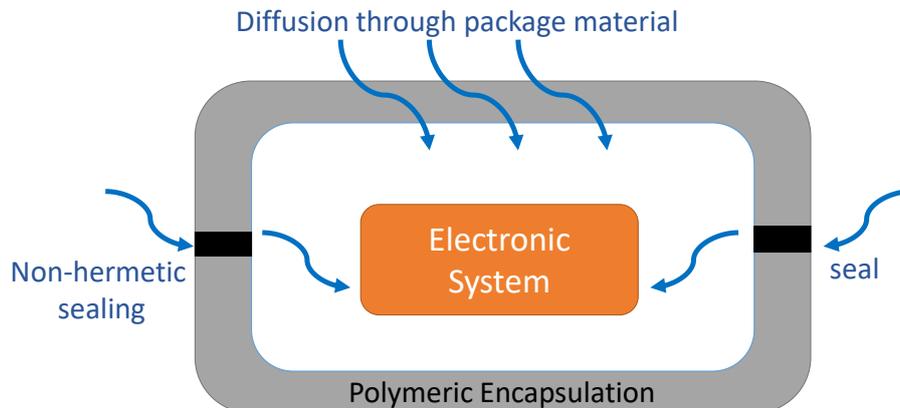


Figure 2.5: Polymeric encapsulation leak rate problems.

Following the definition, hermeticity means "the state of being airtight or gastight".

One of the greatest challenges for polymeric encapsulation is to establish a good hermetic barrier, since polymeric substrates often exhibit permeability. Unfortunately, it's very fundamental for an implantable device to be both water tight and ion tight [37].

Summing up, since the intent of this project is to develop a polymeric package for a medical active implant, the main challenge to face is to satisfy the hermeticity requirement. As figure 2.5 illustrates, the polymeric encapsulation that is going to be fabricated, has two main moisture ingress problems: the first is due to the diffusion of water molecules through the polymeric material due to the permeability of the substrates, while the second is due to the difficulty in establishing a good hermetic sealing procedure to close the package [35].

### 2.3.1 Hermeticity test methods

The most common procedure that is adopted to test package hermeticity is helium leak rate detection. This method is described by Military Standard MIL-STD-883, method 1014. It consists in exposing the package under test, for a given time, in a pressured helium chamber. Next, it should be transferred to a mass detector. The mass detector measures the quantity of helium escaping from the non-hermetic package, since during the exposition, it has penetrated inside the cavity [38]. Indeed, this procedure, which involves the mass spectrometer, could be practiced with other gasses, as well, but helium is generally preferred because of its few presence in normal atmosphere [39]. An important drawback of this technique is that the sensitivity drops dramatically for small volumes and vacuum applications, such that the leak may result undetectable [40]. Anyway, for current object, the worst drawback is that this approach is not suitable for polymers and glass packages, since they are permeable to helium. So that the leak rate calculation can be accurate, it is needed a permeable material. For glass, it may be used a different tracer gas, but for polymers, which are permeable to almost all the gasses, it is actually not convenient [40].

Other typical adopted procedures are: radioisotope fine leak test, optical leak test, Fourier Transmission Infrared Spectroscopy (FTIR), Raman Spectroscopy, Q-factor methods and many others.

For the project, an in-situ monitoring system involving humidity and temperature sensors has been exploited. The humidity inside the cavity will be measured over time and the data will be transmitted by means of Bluetooth communication.

In-situ methods have demonstrated to provide the best sensitivity for leak detection and permeation of implants packages, especially when micro packages are involved. Moreover, these measurements systems can be easily integrated in CMOS dies [39].

The disadvantage of this procedure is that it's necessary to add the electronic structure to make use of these monitoring sensors, but, anyway, the advantages are that any package material can be used and there are no sensibility limitations due to cavity dimensions. In fact, in situ methods are independent on the volume. Furthermore, it is possible to detect the outgassing of internal materials, as well [40].

### 2.3.2 Humidity measurement

Humidity is defined as the measure of water vapor amount in a certain gas, like air.

Many parameters are defined to express the moisture content, such as relative humidity, absolute humidity, dew point temperature and mixing ratio.

**Physics of humidity - Origin of water vapor** As the previous definition states, humidity is a water vapor content estimation.

To understand the origin of humidity and its physical mechanism, let's take into consideration a closed chamber with liquid water, as the one illustrated in figure 2.6.

Temperature governs the average kinetic energy of the system and, therefore, the  $H_2O$  molecules impulses in the liquid state, driven by kinetic energy, depend strongly on temperature, as a consequence. Some particles, which exhibit higher energy than the average, can exit from the liquid state toward evaporation. Since the gaseous particles move around the closed chamber, some of them will collide with the surface of the liquid water and may be forced to come back there, in a liquid state, again. After a while, equilibrium occurs at a point such that the number of water particles escaping from the liquid state is the same of the number of particles which condensate rejoining liquid water. In other words, when the evaporation rate is equal to the condensation rate, the system has reached equilibrium. When this state is achieved, the number of water vapor

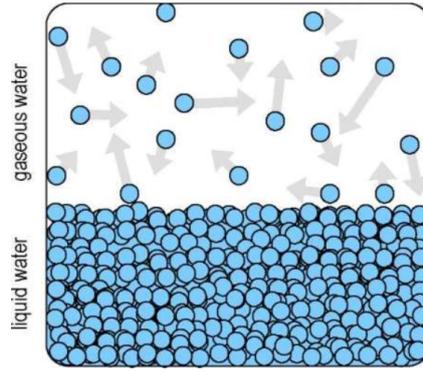
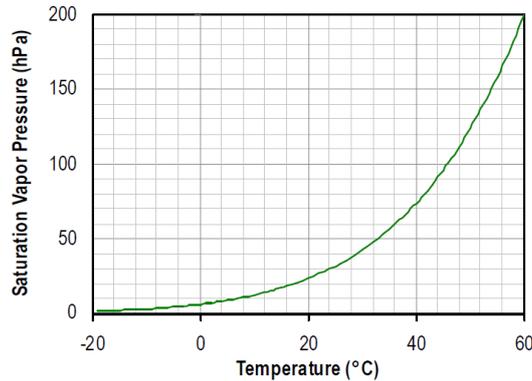


Figure 2.6: Closed chamber with liquid water and vapor [3].

molecules stays unchanged [3].

As it can be noticed, the gaseous water particles apply a pressure to the chamber walls. At equilibrium, the water vapor molecules have achieved their maximum number, therefore their pressure corresponds to the maximum water vapor pressure that the closed chamber can hold. This is called *saturated vapor pressure*.

Then, following the Magnus Formula [3], the saturated vapor pressure above water depends on temperature and it is illustrated in figure 2.7.

Figure 2.7: Saturated Vapour Pressure  $e_w(t)$  [3]

**Absolute humidity** Absolute humidity is defined as the mass of water vapor per unit volume of dry air [41].

$$RH_{abs} = \frac{m_{h_2o}}{V_{air}} \quad (2.1)$$

$RH_{abs}$  is the absolute humidity and it is expressed in  $g/m^3$ ,  $m_{h_2o}$  is the mass of water vapor and  $V_{air}$  is the volume of dry air.

**Relative humidity** Relative humidity is a pure number, generally expressed as a percentage, and it is an expression of water vapor content in relation with air temperature. As already said, the saturation water vapor pressure corresponds to the maximum pressure exercised by evaporated water molecules, achieved at equilibrium. In this state, the relative humidity has reached its maximum value (100 %). Therefore, relative humidity is defined as the quantity of water vapor pressure with respect to the saturation vapor pressure at a given temperature.

$$RH = \frac{p_{h_2o}}{e_w} \quad (2.2)$$

$RH$  is the Relative Humidity,  $p_{h_2o}$  is the water vapor pressure and  $e_w$  is the saturation water vapor pressure.

Relative humidity depends on temperature since saturation water vapor pressure depends strongly on temperature. In fact, if the temperature increases, the kinetic energy increases, as a consequence, and the probability that the liquid molecules start evaporating increases. Therefore, at higher temperature, the saturation water vapor pressure is higher, as well.

The dependence of saturation vapor pressure from the temperature can be also easily observed in figure 2.7.

**Dew point temperature** Dew point temperature is another parameter to quantify the moisture content in the air. It is defined as the temperature, considering the humidity constant, at which the air should be cooled to achieve saturation. In other words, dew point temperature is the temperature at which the relative humidity reaches 100% due to temperature cooling down which causes the decrease of the saturation vapor pressure.

The dew point gives a measure of the absolute humidity which corresponds to the saturation vapor density at that temperature [42].

**Specific Humidity** Specific humidity is the ratio of the mass of water vapor to the total mass of air containing it [41].

$$RH_{spec} = \frac{m_{h_2o}}{m_{tot}} \quad (2.3)$$

$RH_{spec}$  is the specific humidity,  $m_{h_2o}$  is the mass of water vapor and  $m_{tot}$  is the total mass of air.

**Mixing ratio** Mixing ratio is the ratio of the mass of water vapor to the mass of dry air [41].

$$MR = \frac{m_{h_2o}}{m_{air}} \quad (2.4)$$

MR is the Mixing Ratio,  $m_{h_2o}$  is the mass of water vapor and  $m_{air}$  is the mass of dry air.

### 2.3.3 Humidity measurement methods

There are several types of hygrometers. These can involve the absolute humidity measurement, the relative humidity one, .. and so on. Some of them are: hair tension hygrometer, psychrometer, resonance hygrometer, dew point hygrometer and capacitive/resistive humidity sensing mechanisms. Due to their easy integration on CMOS technology, humidity sensors working principle exploits a capacitive or a resistive hygrometer method.

The sensors that are going to be adopted in the project are all capacitive humidity sensors, therefore, due to this, in the following, the capacitive detection method is going to be explained.

#### Capacitive humidity sensor

Capacitive humidity sensing exploits, as working principle, the variation of capacitance value due to the change of the relative permittivity constant of the dielectric which is placed between the capacitor electrodes.

The dielectric between the electrodes is often a humidity-sensitive polymer characterized by high porosity. This characteristics makes the polymeric thin film able to capture the moisture of the target environment, so that the change of the permittivity is a result of the moisture incorporation within the polymer film.

## Chapter 3

# Humidity Monitoring System

### 3.1 System Specifications

Temperature and humidity monitoring system is an embedded system whose aim is to measure temperature and humidity inside an active implantable device package cavity in order to keep under control the moisture ingress. Indeed, the purpose is to monitor many points inside package cavity and, because of this, it should be provided with more humidity sensors located in different places on the board. The measurements obtained are transferred from sensors to the Micro-Controller Unit (MCU) by means of I2C serial-bus communication interface and they have been sent to the user via Bluetooth 5 wireless communication.

The block diagram, depicted in figure 3.1, shows the basic system functionalities.

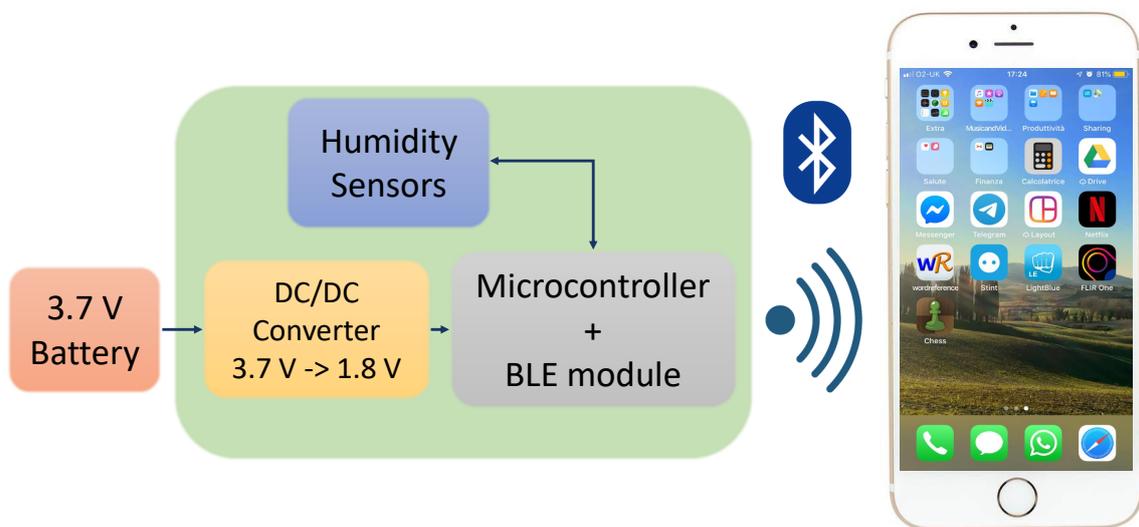


Figure 3.1: Humidity monitoring system block diagram

Requirements and featuring specifications can be itemized in the following list:

- MCU module already provided of Bluetooth antenna;
- temperature range: 0 - 80°C;
- humidity range: 0 - 100% RH;
- very long duration battery;
- low power system,
- reduced size of PCB;
- smartphone or computer application to receive data.

## 3.2 Design Flow

To develop the embedded system, the following steps have been developed:

- components selection;
- schematic diagram design by means of Altium Designer 19.1.8;
- PCB layout design by means of Altium Designer 19.1.8;
- Gerber file generation following the manufacturing specifications;
- PCB fabrication and components assembly;
- firmware implementation.

## 3.3 Humidity Monitoring System 1

### 3.3.1 Components

#### Temperature and Humidity Sensor

The temperature and humidity sensor chosen is the SHTW2 produced by *Sensirion*. Its working principle consists of a capacitive humidity sensor and a bandgap temperature sensor. For what concerns characteristics and some datasheet main informations, refer to appendix A.1.

**SHTW2 I2C Communication** SHTW2 is provided with I2C interface, which makes it suitable for a serial bus communication between a microcontroller and other devices.

SHTW2 I2C address is shown in table 3.1, while in table 3.2 there are listed all the commands that can be sent to the sensor to read its measurements.

SHTW2	Bin.	Dec.	Hex.
I2C Address	1110000	112	0x70

Table 3.1: SHTW2 I2C Address [4].

Clock Stretching Enabled		Clock Stretching Disabled	
Read T First	Read H First	Read T First	Read H First
0x7CA2	0x5C24	0x7866	0x58E0

Table 3.2: SHTW2 I2C commands [4].

Specifically, there are two big groups of commands. The first one involves the clock stretching operation which means that the peripheral is able to put low the clock line until it is not ready to respond to a master command. The second group does not supply this possibility, so the programmer should guarantee that the time required for a sensor response was respected. Then, there is the possibility to choose if to receive the temperature measurement at first, or the relative humidity one.

In addition, there are other 2 important commands for SHTW2: the first is the *READ ID* one, which has to be sent to read the product specific codes and to verify the correct presence of the sensor in the communication bus; the other is the *SOFT RESET* which forces the device in a well defined low power state, without switching off the power supply. The two commands are specified in table 3.3 [4].

Command	Bin.	Hex.
Read ID Register	1110111111001000	0xEFC8
Soft Reset	100000001011101	0x805D

Table 3.3: SHTW2 Soft Reset and Read ID commands [4].

**SHTW2 Accuracy** Reading from SHTW2 datasheet, typical and maximal tolerance of temperature measures are illustrated in figure 3.2.

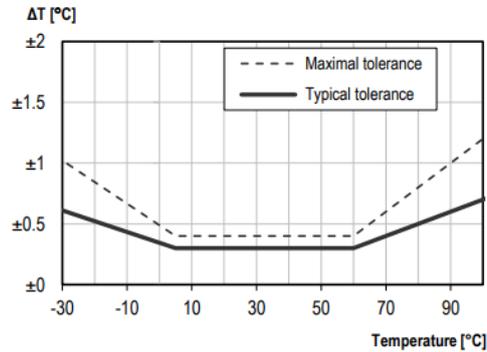
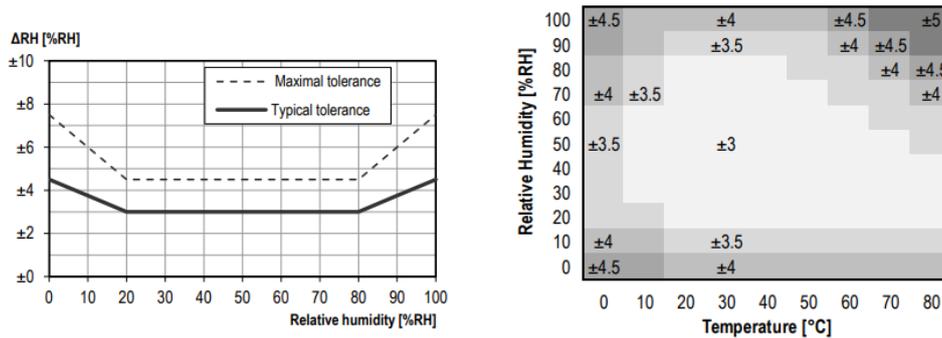


Figure 3.2: Typical and maximal tolerance for the temperature measure in °C [4].

For what concerns the relative humidity, the accuracy depends on temperature value, as well. At 25°C the datasheet shows, in figure 3.3(a), typical and maximal tolerance for relative humidity sensor. Considering all the temperature range, the accuracies for humidity values are given in figure 3.3(b).



(a) Typical and maximal tolerance for relative humidity in %RH at 25°C [4] (b) Typical accuracy of relative humidity measurements given in %RH for temperatures 0–80°C [4].

Figure 3.3: Relative humidity measures tolerances

## MCU Module

As a main component, it has been chosen ISP1507-AX chip, from *Insight*, that is an high performance multiprotocol module suitable for Bluetooth applications, which integrates MCU and Antenna. ISP1507-AX version module is based on nRF52 Nordic Semiconductor System-On-Chip (SoC), more exactly on nRF52832, which is provided with an ARM Cortex M4 CPU at its hearth and supports Bluetooth 5.0 Low Energy. Main features and characteristics of ISP1507-AX chip are described in appendix A.2, while in appendix A.3 it is presented the Nordic nRF52832 chip. An important remark about ISP1507 deals about I2C interface. In fact, the choice of SCL and SDA lines depends on the programmer: he can select any digital input/output pins for this purpose. When the two inputs/outputs should be configured as I2C lines, there is the possibility to enable internal pull-up resistors (integrated inside nRF52832 SoC), whose value is 13 kΩ. Therefore, in the schematic design it is not necessary to add external pull-up resistors to properly enable I2C communication.

## Battery

The battery that it has been chosen is illustrated in figure 3.4. It's a 3.7V 165 mAh lithium polymer (LIPO) rechargeable battery. It is fabricated by *BAK* manufacturer, its model number is LP-402025-IS-3 and its standard dimensions are of 26 mm x 20 mm x 3.8 mm (L x W x H).



Figure 3.4: 3.7 V Lithium Polymer Battery

## DC/DC Voltage Regulator

The battery can power the system at 3.7 V, but, the idea is to power the MCU at the same voltage level supported by the sensors, that is 1.8 V. Because of this, and to regularize the voltage, it is necessary a DC/DC Converter. It has been chosen TPS82740A Step Down Converter Module (refer to appendix A.4), developed by *Texas Instruments*. The main advantage in adopting this module is that it includes all the required passive components and, moreover, it makes very easy to set the wanted output voltage level by simply setting three pins.

## LEDs and Passive Components

Since, from the list of bio-compatible polymeric materials that can be 3D printed, there is the possibility to choose a particular plastic which is translucent, the idea is to add some optical signals to transmit informations that can be easily visualized. Because of this, two LEDs have been added: the first is a blue LED, while the second is a green one (for their basic characteristics, refer to appendix A.5). Each light emitting diode (LED) has a maximum current that it can safely manage. Overpassing that value, even for a short time, may damage irreversibly the component. Thus, to limit the current flowing through a LED, a common method is to add a series resistor [43].

Thus, it is necessary to include two passive components, as well.

To calculate the right resistors values, the formula is the following:

$$R = \frac{V_s - V_{LED}}{I_{LED}}$$

where  $V_s$  is the supply voltage,  $V_{LED}$  is the forward voltage of the LED and  $I_{LED}$  is the forward current. So, considering as a power supply, the voltage coming from the battery (3.7 V) and LEDs corresponding forward voltages and currents (table A.4 in appendix A.5), the 2 resistors that are needed have values:  $20\Omega$  and  $24.9\Omega$ . Both the selected resistors are SMD elements whose size is 0603 (1608 metric).

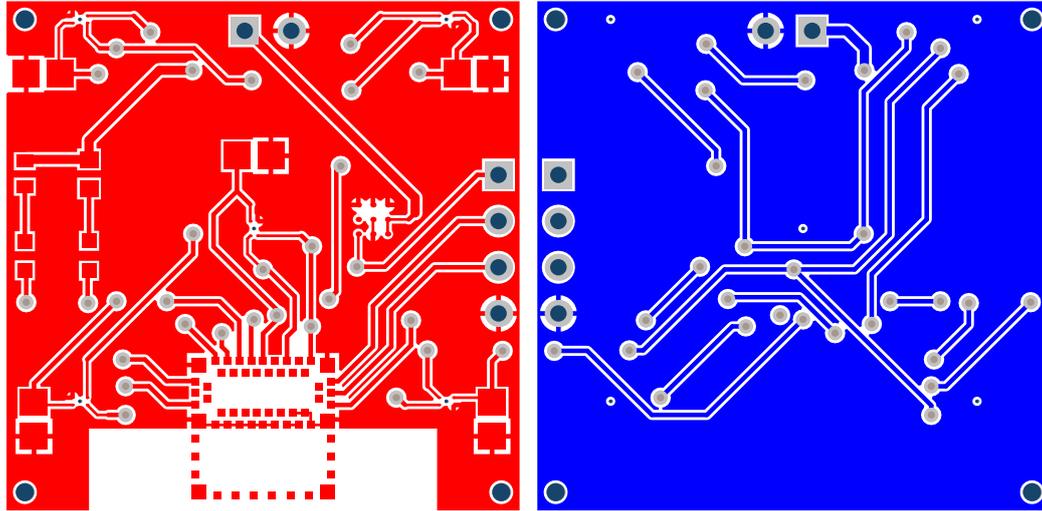
## Connectors

Finally, the project, to be completed, needs two connectors. The first is a 2-pin header that is needed to link the battery to the circuit. The second is a connector to program the MCU module. Nordic nRF52832 SoC is provided of a SWD (Serial Wire Debug) interface. Following SWD protocol, it needs only two lines to program the SoC: the SWDIO and the SWDCLK lines. Because of this, the debugging connector should have 4 pins: the first for the supply voltage, the second for the clock SWDCLK, the third for the data SWDIO and the last for the ground.

### 3.3.2 Schematic and PCB Layout

The humidity monitoring system complete schematic is depicted in figure B.2 reported in Appendix B, while in figure B.1, always in Appendix B, it is shown the sheet symbol schematic representing the sensors communication with the Nordic SoC.

Once the schematic is completed, the next step is to project the PCB (Printed Circuit Board) layout. The completed 3D representation (provided by Altium Designer 19.1.8 software) is illustrated in figure C.1, in Appendix C, where it can be seen the top and the bottom layers views. The board is a square whose dimensions are of 29 mm x 29 mm. In figure 3.5 there are depicted the composite drawings of the designed PCB: in subfigure 3.5(a) It is shown the top copper plane, while in subfigure 3.5(b) it is illustrated the bottom layer.



(a) Composite drawing - Humidity Monitoring System 1 Copper plane (Top layer) (b) Composite drawing - Humidity Monitoring System 1 Copper plane (Bottom layer)

Figure 3.5: Composite drawings - Top and Bottom layers of Humidity Monitoring System 1.

## 3.4 Humidity Monitoring System 2

A new system has been designed due to problems raised in soldering components of the first one. BGA packages and the too tiny dimensions make too difficult the assembling step. Therefore, a new PCB was designed and fabricated in order to find easier the assembling by hand of all the components packages. Board overall dimensions are always the same (29 mm x 29 mm), but all components have been changed, even though conserving the same functionalities.

### 3.4.1 Components

#### Temperature and Humidity Sensor

The temperature and humidity sensor chosen is the SHTC3 produced by *Sensirion*. Refer to appendix A.6 to read about SHTC3 main features. As SHTW2 sensor, its working principle consists of a capacitive humidity sensor and a bandgap temperature sensor.

**SHTC3 I2C Communication** SHTC3 is provided with I2C interface, which makes it suitable for a serial bus communication between a microcontroller and other devices. Its I2C address is shown in table 3.4, while in table 3.5 there are listed all the commands that can be sent to the sensor to read its measurements.

SHTC3	Bin.	Dec.	Hex.
I2C Address	1110000	112	0x70

Table 3.4: SHTC3 I2C Address [5].

Specifically, there are two big groups of commands. The first one involves the clock stretching operation which means that the sensor is able to put low the clock line until it is not ready to respond to a master command. The second group, instead, does not supply this possibility, so

	Clock Stretching Enabled		Clock Stretching Disabled	
	Read T First	Read H First	Read T First	Read H First
<b>Normal Mode</b>	0x7CA2	0x5C24	0x7866	0x58E0
<b>Low Power Mode</b>	0x6458	0x44DE	0x609C	0x401A

Table 3.5: SHTC3 I2C commands [5].

the programmer should ensure that the time required for a sensor response was respected. Then, there is the possibility to choose if to receive the temperature measurement at first, or the relative humidity one. All the commands can be adopted in normal or low power mode.

In addition to these, there are other two commands which are necessary to start and to end the communication with the SHTC3. Upon the power supply reaching the right level, the sensor stays in an idle state and it should be set to sleep with a specific command. So, when the master wants to communicate with SHTC3, first of all it should sent a wake up command to it and the communication can start. These two commands are listed in table 3.6.

Command	Bin.	Hex.
<b>Sleep</b>	1011000010011000	0xB098
<b>Wakeup</b>	0011010100010111	0x83517

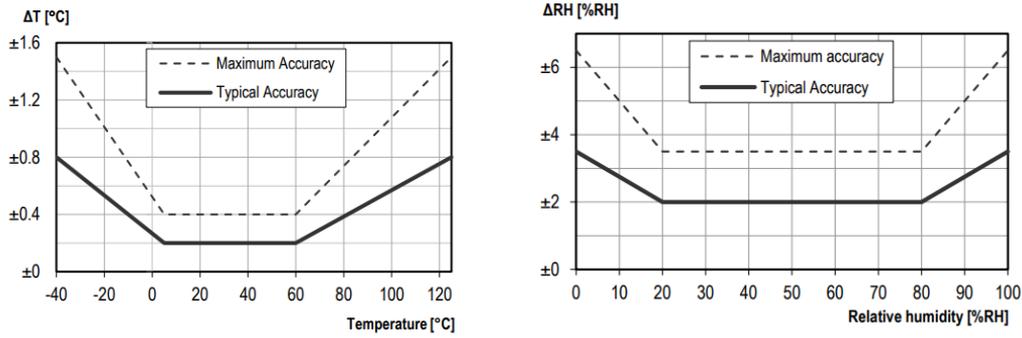
Table 3.6: SHTC3 Sleep and Wakeup commands [5].

The communication sequence is the following:

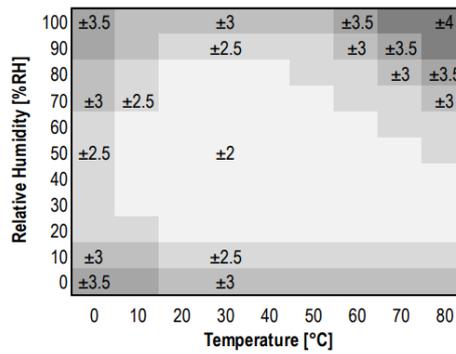
- the master sends in the I2C bus the address of the device it wants to communicate with and write 0 to its 8th bit, which means that it wants to write in the sensor.
- Once the sensor has sent the ACK, the master sends the wake-up command and waits for ACK after each byte it has sent.
- Wait for a wake up time, specified in datasheet [5].
- The master sends in the I2C bus the address of the device and write 0 to its 8th bit, to write a measurement command to the sensor.
- Once the sensor has sent the ACK, the master sends two bytes of the command and waits for ACK after each byte it has sent.
- To read the sensor measurements, the master should send again the device address, but, this time, the 8th bit should be set to logic 1, which stays for read command.
- After that, the sensor start sending its measurements. Depending on the command the master has sent, the first three bytes are related to first value the command has said to want read, while the last three to the other value. The first byte corresponds to the MSB byte and the third bytes, for each measurement, is a CRC checksum byte. The master, if it wants to keep on reading, should send an ACK after each byte it has received.
- Master transmits in the I2C bus the address of the device and write 0 to its 8th bit, to send the sleep command to the sensor.

**SHTC3 Accuracy** From reading SHTC3 datasheet, typical and maximal tolerance regarding the measurements of temperature sensor are illustrated in figure 3.6(a).

For what concerns the relative humidity measurements, the accuracy depends on the temperature, as well. At 25°C the datasheet shows, in figure 3.6(b), the typical and the maximal tolerance for relative humidity sensor. Considering all the temperature range, the accuracies for humidity values are given in figure 3.6(c).



(a) SHTC3 Typical and maximal tolerance for the temperature measure in °C [5] (b) SHTC3 Typical and maximal tolerance for the relative humidity in %RH at 25°C [5]



(c) SHTC3 Typical accuracy of relative humidity measurements given in %RH for temperatures 0 – 80°C. [5]

Figure 3.6: Typical accuracy for relative humidity (in %RH) and for temperature measurements (in °C) [5].

### MCU Module

As a main component, it has been chosen BL652-SA (see appendix A.7) developed by *Laird* that is an high performance module suitable for Bluetooth Low Energy applications. As ISP1507-AX, it is based on nRF52832 Nordic chip (appendix A.3).

### DC/DC Voltage Regulator

It has been chosen LMZ21700 Step Down DC/DC module (appendix A.8), developed by *Texas Instruments*. The module includes a switching regulator and an integrated inductor.

As before, the aim is to obtain 1.8 V as output voltage. The voltage selection occurs by means of two resistors that have to be set for the voltage feedback. The components values for  $V_{OUT}=1.8$  V are the ones itemized in table 3.7.

Component	Value	Description	Series
$C_{IN}$	22 $\mu F$	$\geq 25V$	X7R or X5R
$C_{OUT}$	22 $\mu F$	$\geq 10V$	X7R or X5R
$SS$	3300 pF	$\geq 10V$	X7R or X5R
$R_{FBT}$	147 $k\Omega$	1 %	
$R_{FBB}$	118 $k\Omega$	1 %	

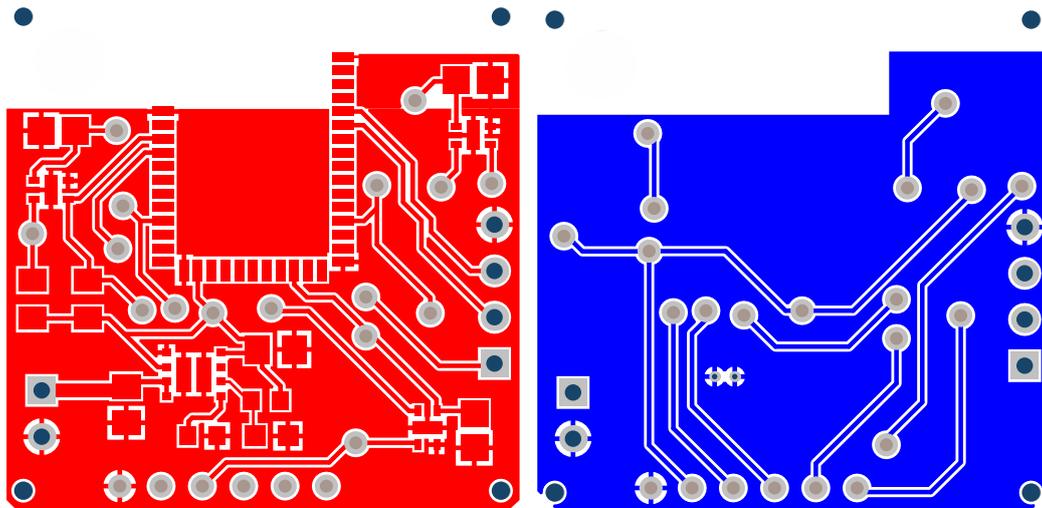
Table 3.7: Components values for  $V_{OUT}$  1.8 V [15].

### 3.4.2 Schematic and PCB Layout

The complete schematic of this second humidity monitoring system is reported in figure B.3, in Appendix B.

Once that the schematic design is finished, the following step is to project the PCB (Printed Circuit Board) layout. *Altium Designer* allows to visualize the designed PCB even in a 3D view. In figure C.2(a) (Appendix C) it is shown the top plane 3D view, while the bottom layer is illustrated in figure C.2(b).

In figure 3.7 there are depicted the composite drawings of this second designed PCB: in subfigure 3.7(a) It is shown the top copper plane, while in subfigure 3.7(b) it is illustrated the bottom layer.



(a) Composite drawing - Humidity Monitoring System 2 Copper plane (Top layer) (b) Composite drawing - Humidity Monitoring System 2 Copper plane (Bottom layer)

Figure 3.7: Composite drawings - Top and Bottom layers of Humidity Monitoring System 2.

## Chapter 4

# Firmware development

As soon as PCB is designed, it's time to start programming the Microcontroller Unit (MCU) to send sensor measurements to the external controller. This chapter deals with the firmware implementation for the required application.

The first phase of the work was tested on Nordic Development board nRF52 DK in order to make easier the corrections of errors, the debug and the overall programming that it can be easily executed via USB connection. Next, the code was fitted for Nordic chip nRF52832 integrated within both the modules considered for the custom embedded systems.

Nordic Semiconductor nRF52 DK is a single-board development kit that it is suitable for many applications, including Bluetooth Low Energy, and it is shown in figure 4.1.

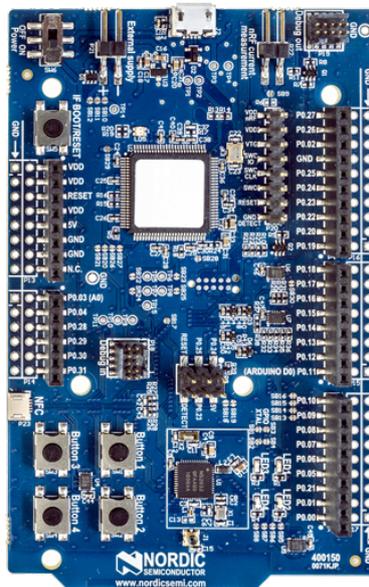


Figure 4.1: nRF52 DK (Copyright © Nordic Semiconductor)

The Development Kit gives the possibility to easily access to all the GPIOs and interfaces via connectors and it is provided with four programmable buttons and LEDs. The SoC mounted on the board is nRF52832, the same one included in the BLE modules of both the PCB projects.

Nordic offers many SDK (Software Development Kit) versions which contain many examples and application codes. From the Nordic website, it is possible to freely download these SDK packets and, for the current project, it has been selected the SDK 15.0.0 version. In figure 4.2 it is illustrated a scheme showing how the software development kit works with software and wireless protocol stack libraries enabling the programming of nRF52 System on a Chip.

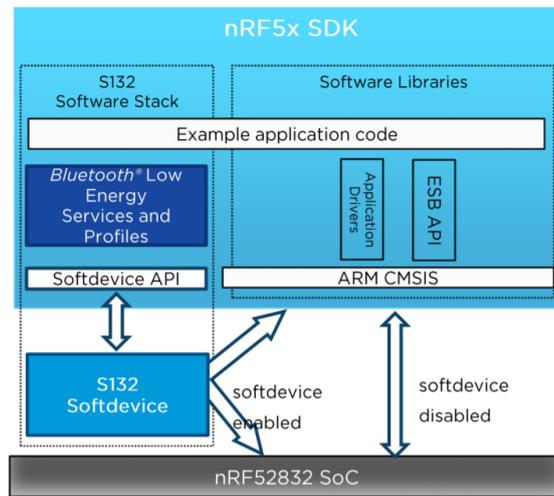


Figure 4.2: nRF5x SDK Modules [6].

For what concerns the IDE, it has been chosen *SEGGER Embedded Studio* (SES). The reason for this choice is to have the possibility to freely download this software without any commercial licence and because it makes all the interfacing drivers easily available for the download. Another useful tool is *J-Link RTT Viewer* which allows to visualize debug messages and error informations on the terminal. The last software to download is *nRFgo Studio*, which has been used to erase the board from previous applications and to program it with the wanted version of Softdevice, needed for establishing the Bluetooth Low Energy stack. For all BLE codes, nRF52832 SoC has been programmed with *s132 Softdevice version 6*. Finally, to receive sensors measurement by Bluetooth, it is possible to freely download the iPhone application *LightBlue*, developed by *Punch Through*, or to exploit a computer program, called *nRF Connect* to read data from PC.

In more details, the overall programming code can be divided in two main parts: the first deals with I2C communication between MCU (master) and the sensors (slaves), while the second part is responsible for sending via Bluetooth the data collected from the sensors to the external controller.

The code was tested, at first, by just adopting nRF52 Development Kit and an external SHTW2 sensor (connected to the board by wires), then it was run on both the humidity monitoring systems that have been designed.

## 4.1 Inter Integrated Circuit (I2C) serial bus protocol

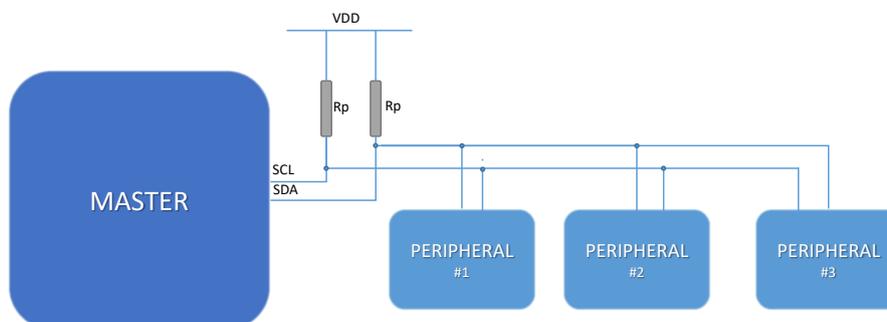


Figure 4.3: Example of a I2C communication between a master and three peripherals.

Inter Integrated Circuit (I2C or IIC) protocol has been developed to allow the communication between a microcontroller and more peripheral devices by means of a parallel connection made by

just two lines. So that, more devices can be connected to the same serial bus and each one must be identified by an unique 7-bit address.

The communication happens, basically, by establishing two roles: the master and the peripheral. It can be just a one master at time and all the others have assumed to be peripherals or slaves. Usually, the microcontroller joins the master role, while the others (memories, sensors and so on) are considered as peripherals. Only the master has the possibility to start a communication and the slaves should just respond to a master command. An example of a I2C serial bus communication between a master and three peripherals is shown in figure 4.3.

The two serial bus lines are called Serial Clock (SCL) and Serial Data (SDA). They are both bidirectional and SCL is necessary to synchronize the communication between devices, while, by means of SDA, a device can receive or transmit data which have been transferred in 8-bit packets. They are both open drain input-outputs and, because of this, a pull-up resistor is necessary for each of them [44]. In fact, in figure 4.3 a typical pull-up resistors configuration is illustrated. This means that, when clock and data are not used, they are at high logic level and devices can communicate by just lowering them. Thus, the participants can only drive a logic 0, while logic 1 is interpreted when the lines are realised.

The fundamental rule of I2C protocol is that in SDA line there mustn't be a transition when SCL is high. Because of this, a violation of the rule is interpreted as a special condition, that is START or STOP. In more details, when SDA goes from high to low level, this is a START. On the contrary, when SDA changes state from logic 0 to logic 1, it is interpreted as a STOP condition.

As already introduced, the communication starts by decision of the master. Actually, each device may have a different sequence and procedure, but, generally the sequence is the following:

1. Start condition
2. The master sends in the data line the address of the device it wants to communicate with, setting the last (8th) bit to zero, which means that it wants to write to it.
3. Once having receiving the acknowledgement (ACK) from the peripheral, the master starts sending the command.
4. Stop condition.
5. If the master wants to read from the peripheral, now it has to send again the device address by setting the last bit to 1, which means *read*.
6. Once having sent the ACK, the slave starts sending bytes to the master.
7. At each byte, the master has to send an ACK to the peripheral until it wants to keep on reading.
8. To stop the communication, the master should drive a stop condition or 'send' a NACK to the peripheral.

Sometimes, it could happen that the slave isn't able to provide data to the master at the rate requested by it. For instance, this could occur when the data to send are not already ready or when a previous operation is not finished, yet [45].

Thus, if a peripheral device admits this procedure, it can drive the SCL line low (even if, generally, only the master can control the clock line). In this way, the master knows that the slave is busy or that it's not ready to respond to the command, so it should just wait. Once the SCL line is released, the master understands that the peripheral is finally ready and the communication sequence can keep going on.

#### 4.1.1 Nordic TWI - Two Wire Interface

I2C communication is called TWI - Two Wires Interface in Nordic Documentation and the *twi sensor* example in the *peripheral* folder of the SDK 15.0.0 examples list has been taken as a reference.

For what concerns the code to test on the development board and for the first humidity monitoring system program, the sensor to consider is SHTW2, while for the second humidity monitoring system, it has to be considered SHTC3 humidity sensor.

### I2C initialization

Before starting a communication, it's fundamental to initialize the board by configuring the GPIOs (General Purpose Input/Outputs) involved for SCL and SDA lines, clock frequency and so on.

First of all, it is necessary to enable the TWI peripheral that is going to be adopted. Drivers and nRF52832 peripherals can be easily enabled or disabled by means of *CMSIS Configuration Wizard*, accessible from *SEGGER Embedded Studio*. It has been chosen the peripheral *TWI 1* as a driver and in the `main.c` file the code representing this choice is the following:

```

1 /* TWI instance ID. */
2 #define TWI_INSTANCE_ID      1
3
4 /* TWI instance. */
5 static const nrf_drv_twi_t m_twi = NRF_DRV_TWI_INSTANCE(TWI_INSTANCE_ID);

```

Listing 4.1: TWI 1 driver selection

Next, to complete the initialization, it is important to define the GPIOs which will be involved for the bus, to define the frequency for the communication, the interrupt priority and to select the digital outputs for powering the sensors. These decisions are shown in tables 4.1, 4.2 and 4.3.

<b>frequency</b>		100 kHz
<b>SCL GPIO</b>		P0_12
<b>SDA GPIO</b>		P0_11
<b>VDD_sensor</b>		P0_27

Table 4.1: I2C parameters selection - nRF52 DK programming

<b>frequency</b>		100 kHz
<b>SCL</b>		P0_5
<b>SDA</b>		P0_8
<b>VDD_sensor_1</b>		P0_28
<b>VDD_sensor_2</b>		P0_29
<b>VDD_sensor_3</b>		P0_30
<b>VDD_sensor_4</b>		P0_31
<b>VDD_sensor_5</b>		P0_2

Table 4.2: I2C parameters selection - Humidity monitoring system 1 programming

<b>frequency</b>		100 kHz
<b>SCL</b>		P0_27
<b>SDA</b>		P0_26
<b>VDD_sensor_1</b>		P0_19
<b>VDD_sensor_2</b>		P0_6
<b>VDD_sensor_3</b>		P0_11

Table 4.3: I2C parameters selection - Humidity monitoring system 2 programming

Since I2C lines need a specific configuration, the GPIOs reserved for them must be configured as open-drain lines. This is one of the functions that `nrf_drv_twi_init` function does. More exactly, it sets up the 2 GPIOs as output pins, enabling their internal pull-up resistors (13 k $\Omega$ ).

For what concerns the second custom designed PCB, BL652-SA module has reserved pins for I2C lines and they can't be changed. In its configuration, it disables the internal pull-up resistors, so, as it is specified in BL652 datasheet, it is necessary to add externally proper pull-up resistors for these signals lines.

As an example, the initialization code regarding the communication with SHTW2 sensor is the following:

```

1 void twi_init (void)
2 {
3     ret_code_t err_code;
4
5     const nrf_drv_twi_config_t twi_shtw2_config = {
6         .scl           = SCL_PIN,
7         .sda           = SDA_PIN,
8         .frequency     = NRF_DRV_TWI_FREQ_100K,
9         .interrupt_priority = APP_IRQ_PRIORITY_LOW,
10        .clear_bus_init = false
11    };
12
13    err_code = nrf_drv_twi_init(&m_twi, &twi_shtw2_config, twi_handler, NULL);
14    APP_ERROR_CHECK(err_code);
15
16    nrf_drv_twi_enable(&m_twi);
17 }

```

Listing 4.2: TWI initialization

As it can be observed, the interrupt priority was set up to LOW. This is done because, when Sofdevice is enable, the highest priorities levels should be reserved for it, hence it is important to ensure that TWI driver does not not interfere in Softdevice activity.

### I2C communication sequence

As soon as TWI interface was initialized, it's time to start reading humidity and temperature values from the sensor.

The I2C sequence to implement in the main loop is the following:

1. Set up the GPIO pin, which has to power the sensor, as an output and program it at high logic level.
2. Start execution: send to the sensor the measurement command. For instance: send the command READ HUMIDITY FIRST.
3. Wait until the bus is free and the sensor is ready to give measurements to the master.
4. Send to the sensor the read command and save all the receiving bytes in a buffer.
5. Select the bytes dedicated for temperature and for relative humidity values.
6. Disable the digital output pin which has powered the sensor.
7. Wait for TWI\_TIME\_INTERVAL before starting a new measurement cycle.

Obviously, if there are more sensors, the whole procedure has to be repeated for each sensor, by selecting the proper power supply pin, and, just after this, wait for TWI\_TIME\_INTERVAL. Indeed, because of low power purposes, the whole TWI cycle has been implemented in a timer handler function. In fact, it has been created a timer that operates at each TWI\_TIME\_INTERVAL and, then, it is periodically repeated.

**Results elaboration** Results have to be elaborated through a *Matlab* script. They are sent in row values, therefore they need a calculation to be understood as actual humidity and temperature measures, as sensors datasheets suggest.

At first, the temperature and the humidity values are represented by 16 bits as sensor outputs. Following the indications, the procedure is to convert them in a decimal format and, finally, to apply these two equations [4]:

**Relative humidity conversion formula (result in %RH):**

$$RH = 100 \cdot \frac{S_{RH}}{2^{16}} \quad (4.1)$$

where  $S_{RH}$  is the 16-bit raw sensor output for relative humidity in a decimal format.

**Temperature conversion formula (result in °C):**

$$T = -45 + 175 \cdot \frac{S_T}{2^{16}} \quad (4.2)$$

where  $S_T$  is the 16-bit raw sensor output for temperature in a decimal format.

## 4.2 Bluetooth Low Energy (BLE) protocol

This section deals with the implementation of Bluetooth Wireless transmission. The aim is to create a simple profile provided of one service and a characteristic for each sensor measurements. The characteristic value should transmit both the temperature and humidity measures.

Bluetooth is a telecommunication standard which allows many nearby devices to exchange informations to each other. The last version is Bluetooth 5 and it is the one employed for current thesis project since nRF52832 module was adopted and it has available this kind of connectivity. Bluetooth 5 provides the opportunity to be low power (especially with respect to the oldest versions) by adopting the LE wireless technology, which means *Low Energy*, also called *Bluetooth Smart* [46].

Thus, the first key feature for BLE is to be a low power protocol. Moreover, it is capable to sustain a connection for a long time; it is a stateless protocol, which means that each device request has an independent transaction from the others and it can connect a large amount of devices organized in a very flexible topology which can be fitted and adjusted depending on the specific different cases [6]. An example of a Bluetooth topology is illustrated in figure 4.4.

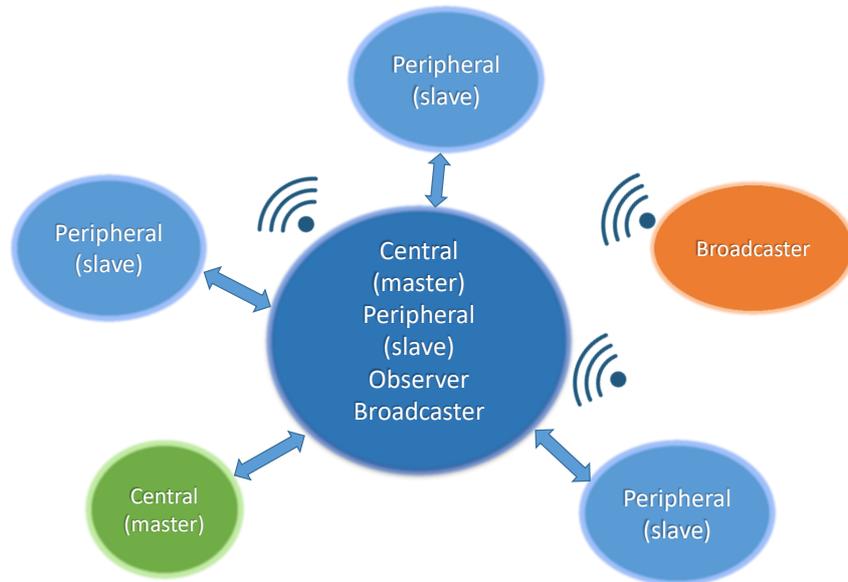


Figure 4.4: Example of a BLE topology.

A topology defines how many devices are involved in a communication, how they are connected to each other and their specific role. Bluetooth protocol defines four roles and each device may perform a specific one of them, but also more together. They are: broadcaster, observer, peripheral and central [6]. The broadcaster is a device that can only transmit; no device can connect to it. For instance, a purpose of a broadcaster could be to let the others know about its presence. On the contrary, the receiver, as the name suggests, can only receive and not transmit. Then, there are the other two which are the most popular: central and peripheral. Central device supports the master role [6]; it can manage many connections and it is the one responsible in initiating the communication with some peripherals. On the other hand, the peripheral supports a slave role [6] and it should be led by the master.

### 4.2.1 Nordic BLE application

Actually, before starting to implement the code for Bluetooth transmission, a brief description of the Nordic BLE applications and stack is indispensable.

Nordic Semiconductor has developed already precompiled libraries which implement the whole stack needed for the Wireless communication protocol. In this way, it allows a clear separation

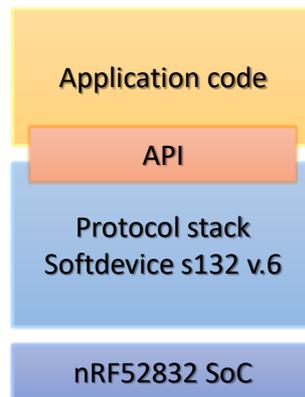


Figure 4.5: Nordic BLE applications stack [6].

between the BLE stack code and the application. In figure 4.5 it is represented a simplified illustration about the protocol layers from the application to the System On a Chip. As it can be seen, the protocol stack which links the application code to the Nordic SoC is implemented by Softdevice.

### Softdevice S132

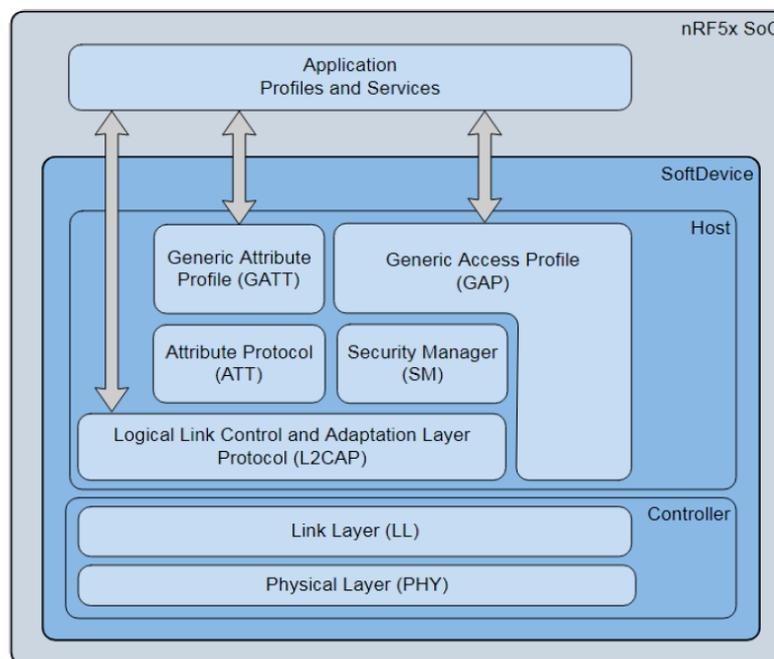


Figure 4.6: Softdevice S132 stack scheme [7].

Softdevice S132 is a Bluetooth 5 protocol stack, suitable for nRF52832 SoC. It supports observer, broadcaster, central and peripheral roles [6]. It allows the establishment of up to 8 connections with one observer or one broadcaster and maximum one peripheral connection [6]. It's a stack which includes GAP, GATT, ATT, SM, L2CAP and Link Layer, as it is shown in figure 4.6.

For this project, Softdevice s132 v.6 was downloaded and programmed on nRF52832 SoC before starting the implementation of the application code.

## 4.2.2 Generic Access Profile (GAP)

At first, considering Bluetooth Low Energy protocol stack, it is necessary to deal with the GAP layer, where GAP stays for *Generic Access Profile* [47]. Briefly, GAP defines the topology of a BLE network [48]. Indeed, it is responsible for advertising, scanning and connection establishment that are going to be better explained in the following sections. In figure 4.7 the GAP state diagram is illustrated to better understand its key concepts and devices states [47].

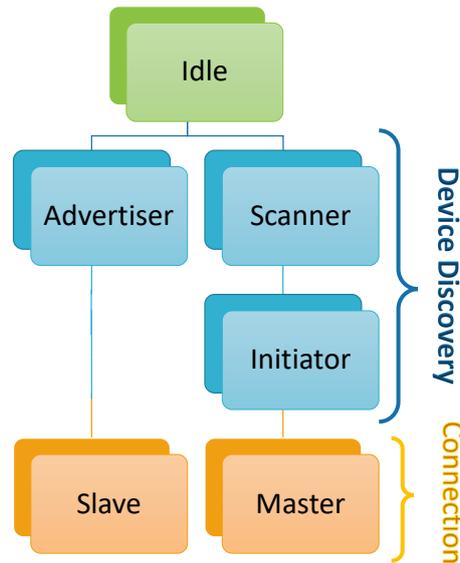


Figure 4.7: GAP state diagram.

The first state is the idle (or standby) upon reset. A device becomes an advertiser when it starts sending advertising packets to let the others know that it is a connectable device [47]. If from one hand there is a device doing advertising, in the other hand it should be present a device doing the scan. This device exploits the scanner role and, once that it has captured an advertising packet from the advertiser, it can initiate a connection with it. Therefore, when the connection is created, the advertiser becomes a slave and the scanner becomes a master.

### Advertising

Advertising is the way to make other devices aware of your presence. It is the step number zero, since it must be executed before establishing a connection. During the advertising, it's possible to broadcast two types of data packet (up to 31 bytes of payload): the advertising and the scan response packet [8]. The overall advertising procedure is described in figure 4.8.

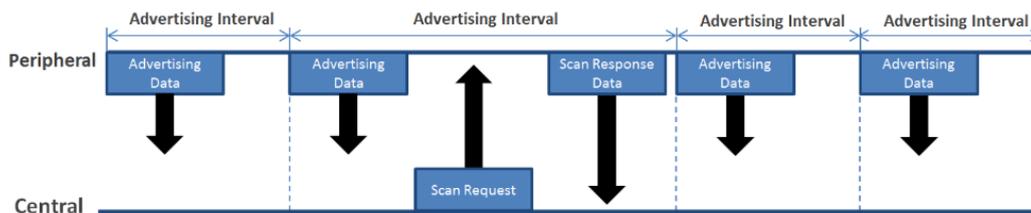


Figure 4.8: Advertising sequence [8].

The peripheral starts sending advertising packets to let the master conscious of its presence (the packet payload contains the advertiser address). These packets are periodically broadcast with an

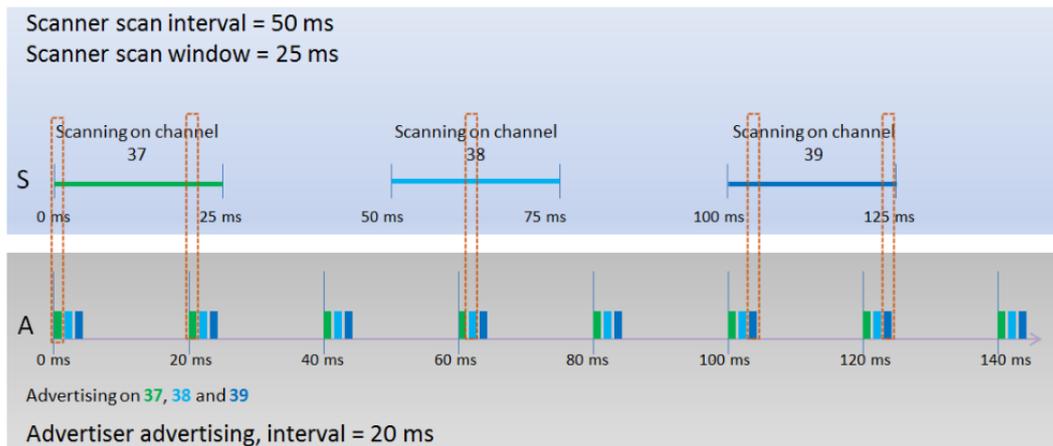


Figure 4.9: Scanner and Advertiser operations [8].

advertising interval that can be configured. At each advertising event, the packets are transmitted in all the three advertising channels: channel 37 @ 2402 MHz, channel 38 @2426 MHz and channel 39 @2480 MHz.

The device which desires to establish a connection with the advertiser, should perform a scan on the 3 channels and, because of this, it is named as scanner. Following network topology, each device should do an advertising or a scan, but, actually, the two actions can be executed even simultaneously by a single device [8]. The scan is executed in sequence: first on channel 37, then on channel 38 and finally on channel 39. Each scan lasts a scan window time interval and the repetition of the scans respects an established scan interval. For instance, as it is shown in figure 4.9, the advertising interval can be 20 ms, the scan window 25 ms and the scan interval 50 ms.

Once the active scanner has captured an advertising packet, it can send a *Scan Request Packet* to the advertiser containing the scanner address. Eventually, the advertiser can answer with a *Scan Response Data* and a connection could be now initiated.

Thus, at this point, it is necessary to set up and to start to transmit advertising packets. To do this, Softdevice provides two main APIs (Application Programming Interfaces):

- `sd_ble_gap_adv_data_set(uint8_t const *p_data, uint8_t dlen, uint8_t const *p_sr_data, uint8_t srhlen):` to set up the advertising data and the scan response data;
- `sd_ble_gap_adv_start(ble_gap_adv_params_t const *p_adv_params):` to start advertising with the parameters defined as input [8].

**Advertising set up** To set up the advertising it's necessary to define the input data for the first Advertising API mentioned before.

- `p_data`: this array represents the raw data of the advertising packet.
- `dlen`: this indicates the length (bytes) of `p_data`. If the data are *NULL*, it should be zero.
- `p_sr_data`: this array represents the data to be transmitted in the scan response packet.
- `srhlen`: as before, it indicated the length of the scan response packet data.

**Advertising start** To call the second provided API for the advertising, it is necessary to configure the following parameters (`p_adv_params`):

- Advertising interval;

- Advertising timeout: The time until the advertising will stop because no connection was established. After this timeout, the user will receive a `BLE_GAP_EVT_TIMEOUT` which is an event signaling this;
- Advertising types: connectable, not connectable, ..;
- Peer address;
- Filter Policy;
- Whitelist;
- Channel(s): channels in which the advertising happens.

When the advertising is started, the event that the user can receive are three: `BLE_GAP_EVT_TIMEOUT`, which is signaling that advertising timeout is expired, `BLE_GAP_EVT_SCAN_REQ_REPORT`, when the advertiser receives a scan request and the `BLE_GAP_EVT_CONNECTED` where a controller device sends a `CONNECT REQUEST` packet and the connection is now started [8].

In the following code there are itemized the advertising timings that have been chosen for the first testing code.

```

1 #define APP_ADV_INTERVAL    50  /**< The advertising interval (in units of 0.625 ms
   . This value corresponds to 187.5 ms). */
2 #define APP_ADV_DURATION    18000 /**< The advertising duration (180 seconds) in
   units of 10 milliseconds. */

```

Listing 4.3: Advertising parameters

## Connection

When the scanner sends a `CONNECT REQUEST` packet, a connection is started [9]. The request should be sent during the RX window which the advertiser opens after having broadcast an advertising packet. Then, after having received the `CONNECT REQUEST`, the advertiser stops the advertising. The procedure is explained in figure 4.10 where it is depicted an example showing a connection request packet received on channel 38.

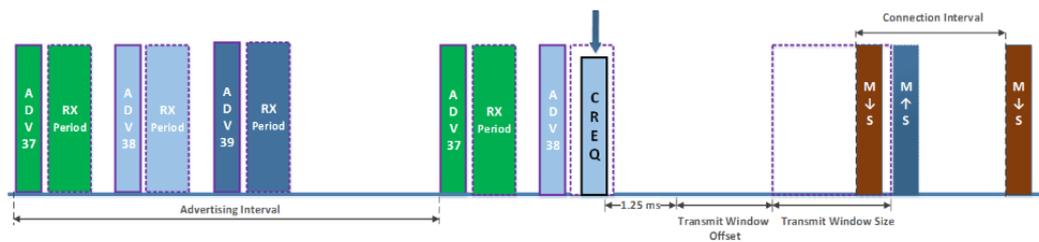


Figure 4.10: BLE Connection initialization [9].

The connection is based on parameters included in the connect request [9] and, after that, the scanner becomes a master whereas the advertiser performs a slave role. Thus, the slave becomes a receiver and waits for a second packet from the master. Once that the slave receives this second packet, it answers and the connection is established [6]. The devices are now ready to exchange data to each other.

The connection parameters are transmitted within the `CONNECT REQUEST` payload and they are the following [9]:

- **AA**: Access address of the connection.
- **CRCInit**: CRC calculation initialization value.
- **WinSize**: Transmit window size:  $\text{WinSize} * 1.25 \text{ ms}$ .

- **WinOffset:** Transmit Window Offset:  $\text{WinOffset} * 1.25 \text{ ms}$ .
- **Interval:** Interval of the connection:  $\text{Interval} * 1.25 \text{ ms}$ .
- **Latency:** Slave Latency. It represents the number of connections reminds that the slave can skip to respond (to save power).
- **Timeout:** It determines how long master keep sending connection events without any response from the slave.
- **ChM:** Channel Map. It says which channels are going to be used and which not.
- **Hop:** It says how big the channel should be jumped inside the channel map list.
- **SCA:** Sleep Clock Accuracy. The slave exploits this value to decide how much its RX window should be large.

The next step is to set up, start and close a connection.

Softdevice provides the following APIs regarding connection step [9]:

- `sd_ble_gap_scan_start(ble_gap_scan_params_t const *p_scan_params)`: to start scanning e to discover if there are connectable devices which are advertising;
- `sd_ble_gap_connect(ble_gap_addr_t const *p_peer_addr, ble_gap_scan_params_t const *p_scan_params, ble_gap_conn_params_t const *p_conn_params)`: to send a `CONNECT REQUEST` packet to an advertiser which is connectable with the scanner.
- `sd_ble_gap_disconnect(uint16_t conn_handle, uint8_t hci_status_code)`: to terminate a connection;
- `sd_ble_gap_scan_stop()`: to stop scanning;
- `sd_ble_gap_connect_cancel()`: to cancel a connection establishment.

When a connection is initiated, the events that could be received are: `BLE_GAP_EVT_ADV_REPORT` which means that it is possible to read the advertising or the scan response packets data and that, from now, it is possible to send a connect request and the `BLE_GAP_EVT_CONNECTED` when the connection has been established. When a connection has been deactivated, the event that is going to be receive is `BLE_GAP_EVT_DISCONNECTED` [9].

### 4.2.3 Generic Attribute Profile (GATT)

It has been briefly explained how the GAP layer manages the advertising and the connection functionalities. Now, it's time for the GATT to come into play. GATT stays for *Generic Attribute Profile* and it conducts the transfer of data between already connected device [48]. It establishes two roles: server and client. Server is the device which collects data and values, while client is the one which could ask for these data. Generally, the procedure establishes that the client sends a request to the server in order to read/write the values stored by it [48].

*The GATT Profile specifies the structure in which profile data is exchanged. This structure defines basic elements as services and characteristics, used in a profile.*

This is the definition given by the Bluetooth Core Specification (v4.2, Vol.3, Part G). It mentions three unknown elements: profile, service and characteristic. Hence, in the following sections, these terms are going to be better explained.

#### Profile setup

Let's say that a connection is established. Thus, how data have been exchanged between a client and a server? Obviously, the way the devices communicate depends on the application. In BLE an application is called *Profile*. A profile represents the overall application functionality. It is made by one or more services, which, in turn, are made by one or more characteristics.

## UUID - Universally Unique ID

UUIDs are universally defined numbers used to identify services and characteristics. All the services and all the characteristics must have their own UUID which has to be transmitted to the client to make it aware about what the server can provide or do for it.

There are two types of UUIDs: short 16-bit UUID and 128-bit UUID [49].

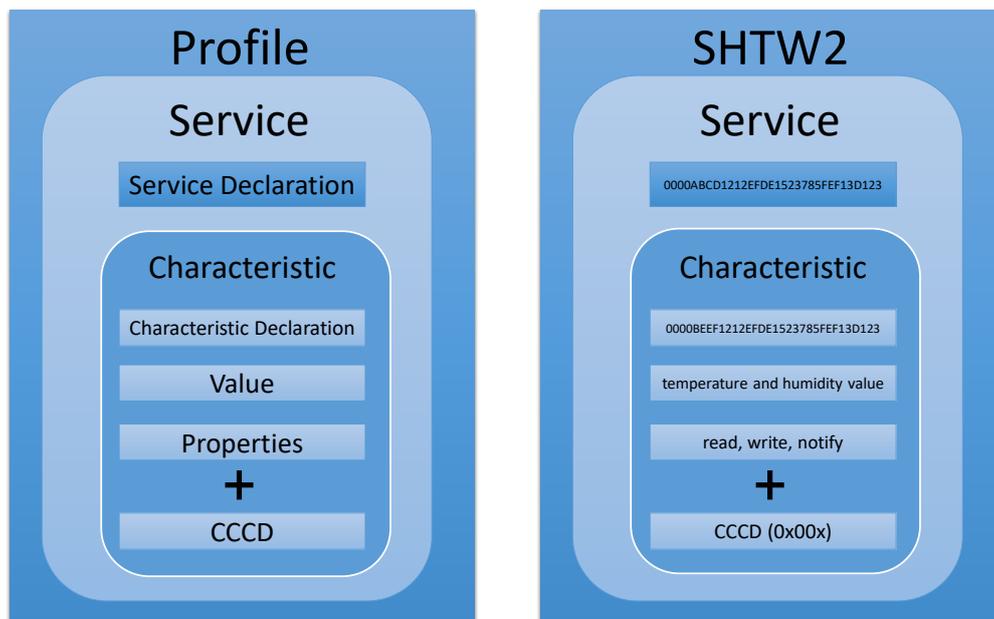
By means of the short 16-bit UUIDs, they can be identified only the already predefined applications/profiles. In fact, BLE is provided with a lot of profiles already defined by Bluetooth SIG which have a specific behaviour and contain their own services and characteristics. An example is the *Heart Rate Profile*. It incorporates a service called the *Heart Rate Service* whose UUID is 0x180D and, in turns, the *Heart Rate measurement* characteristic which is identified by 0x2437 [49].

To create custom services and characteristics, it is necessary to adopt the second type of UUID. This is defined vendor specific UUID and, for convenience, it is often divided in two parts: the BASE UUID and the custom 16 bit UUID defined to taste. An example of a BASE UUID is the following: 0x4A98xxx-1CC4-E7C1-C7F7-F1267DD021E8 and it should be generated randomly by, for instance, any UUID-generator tool. The 16 bits indicated as an "x" in the base address must be filled by a custom defined identifier [49], as the user wants.

## Service

Before starting implementing a new service, it is remarkable to know that there are always two mandatory services: Generic Access Service and Generic Attribute Service. The first one, whose UUID is 0x1800, contains general informations about the device. It has three mandatory characteristics: the device name (UUID: 0x2A00), the appearance (UUID: 0x2A01) and the Peripheral preferred Connection Parameters (UUID: 0x2A04) [49]. This last characteristics is used to defined some parameters once that the connection was established [49]. The Generic Attribute Service (UUID: 0x1801) has just one optional characteristic: the Service Changes, whose UUID is: 0x2A05. This is used to notify the central device about optional changes made to the service structure on peripheral.

At this point, it's time to start developing a new custom service. The service structure that is



(a) Profile configuration.

(b) SHTW2 custom profile configuration.

Figure 4.11: Configuration of the profile, its service and characteristics that are going to be programmed.

going to be implemented, adopting nRF52 DK board in communication with SHTW2 sensor, is

represented in figure 4.11.

To start writing the related code, two programming files has been included in the `main.c`: `shtw2.c` and `shtw2.h` (considering the communication with SHTW2 sensor). The first one is a C file useful to implement all the service functions we want to create, while the second is its related header file.

**Service UUID** Since a new service is going to be created from scratch, it is necessary to define a universally unique ID made by 128 bits. First of all, it has been adopted nRFgo Studio to generate a random BASE UUID, that it is used to define the included characteristic(s), as well. Then, it has been designated a short 16-bit UUID to complete the overall identifier. The declaration of them has been done in `shtw2.h` file and it is the following:

```
1 #define BLE_UUID_SHTC3_BASE_UUID {0x26, 0xFB, 0x3B, 0x6D, 0x51, 0xF6, 0x42, 0x40,
   0x96, 0x7F, 0x07, 0x3B, 0x6A, 0xA1, 0x1B, 0x9E} // 128-bit base UUID
2 #define BLE_UUID_SHTC3_SERVICE 0xABCD // random number
```

Listing 4.4: Service UUID declaration

**Service Declaration** It's time to declare the service structure, which is where all informations and service data are going to be stored.

For instance, it's important to store the handles that are numbers adopted to identify all the profile elements in the attribute table. They are assigned by the Softdevice.

The structure, called `ble_shtw2_t`, is the following:

```
1 typedef struct
2 {
3     uint16_t    conn_handle;    //Handle of the connection
4     uint16_t    service_handle; // Handle of the SHTW2 Service
5     ble_gatts_char_handles_t char_handles; // Handle of the characteristic
6
7 }ble_shtw2_t;
```

Listing 4.5: Service structure declaration

Finally, let's declare a variable of type `ble_shtw2_t` in `main.c`:

```
1 ble_shtw2_t m_shtw2_service;
```

Listing 4.6: Service declaration

**Service Initialization** In the main function, at this point, it is necessary to define a function to initialize the service structure. The following code shows how service initialization function is implemented.

```
1 void shtw2_service_init (ble_shtw2_t * p_shtw2_service)
2 {
3     uint32_t    err_code;
4
5     //Declare Service 128-bit UUID
6     ble_uuid_t    service_uuid;
7     ble_uuid128_t    base_uuid = BLE_UUID_BASE_UUID;
8     service_uuid.uuid = BLE_UUID_SERVICE;
9     err_code = sd_ble_uuid_vs_add(&base_uuid, &service_uuid.type);
10    APP_ERROR_CHECK(err_code);
11
12    //Set service connection handle to default value.
13    p_shtw2_service->conn_handle = BLE_CONN_HANDLE_INVALID;
14
15    //Add the service
16    err_code = sd_ble_gatts_service_add(BLE_GATTS_SRVC_TYPE_PRIMARY, &
17    service_uuid, &p_shtw2_service->service_handle);
18    APP_ERROR_CHECK(err_code);
19
20    //Add a characteristic to the service.
21    temp_char_add(p_shtw2_service);
22 }
```

Listing 4.7: Service initialization

As it can be observed, the first part is responsible for adding the vendor specific UUID to the table of UUIDs in the BLE stack. Then, the service is added by calling the `sd_ble_gatts_service_add()` API. This will define a table containing the shtw2 service.

**Advertising** Since a new service was added, the advertising packet must be modified in order to transmit the service UUID. This means that the device, during the advertising, should notify the scanner about the service of which it is provided with.

Thus, the steps are to declare a variable holding the service UUID and, next, in the `advertising_init()` declaration in `main.c`, the scan response packet data payload should be filled by the Service UUID. The related codes are the following:

```

1 static ble_uuid_t m_adv_uuids [] =
2 {
3     {
4         BLE_UUID_SERVICE ,
5         BLE_UUID_TYPE_VENDOR_BEGIN
6     }
7 };

```

Listing 4.8: Service UUID variable declaration

```

1 init.srdata.uuids_complete.uuid_cnt = sizeof(m_adv_uuids) / sizeof(m_adv_uuids[0]);
2 init.srdata.uuids_complete.p_uuids = m_adv_uuids;

```

Listing 4.9: Scan Response Packed data modification

#### 4.2.4 ATT - Attribute Protocol

Bluetooth Low Energy Protocol is built upon a number of layers, as figure 4.6 illustrates [50]. In particular, this stack shows that GATT is built upon ATT. ATT is based on Client-Server relationship [50]. The Server is the device which holds data like sensor measurements, positions, state of a light switch and so on. All the data that can be collected by the Server are organized inside an attribute table. Therefore, when a Client wants to access to a certain information, that could be, for instance, a sensor measure, it should refer to a certain row in the attribute table by means of the ATT [50]. Each row in this table is defined as an attribute.

The fields in the Attribute table are [50]:

- **Attributes Handles**

Attribute handle is an index used to uniquely identify an attribute in the table. SoftDevice makes an extended use of handles which are 16-bit numbers that actually depend on the number of attributes [50].

- **Attributes Types (UUIDs)**

They are the universally unique ID used to uniquely identify each type of attribute present in the table. As already introduced, they are 16-bit numbers if they are used to identify predefined types of attributes, while the custom defined attributes must be 128-bit numbers.

- **Attributes Permission**

Attributes permission regard attributes values and they indicate how the user can interact with these: if he can read or write them and what authorizations are needed to do that [50].

- **Attribute Value**

The value can be whatever. For instance, in the case of a characteristic value attribute, its value represent the information to transmit to the user or the data that can be write by the user. In the case of a characteristic or service declaration attribute, the value may represent information about other attributes, as, for instance, the following characteristic value or a characteristic declaration included by declared service [50].

It can be said that GATT profile carries out the role to group specific attributes in the table. In fact, GATT specifies the attributes structure: how they are linked to each other and their specific properties.

After having briefly introduce in general Bluetooth Low Power Protocol, it's now time to start developing the own GATT structure building up specific service and characteristic needed for the custom project.

### Characteristic

SHTW2 Service	Handle	UUID Type of Attribute	Attribute Permission	Attribute Value
<b>Service Declaration</b>	0x00X	Service declaration: 0x2800	Read only, no authentication, no authorization	Custom service UUID: 0x0000F00D-1212-EDFDE-1523-785FEF13D123
<b>Characteristic Declaration</b>	0x00X	Characteristic declaration: 0x2803	Read only, no authentication, no authorization	Properties: notify, read,write. Value handle (0x00X) Custom characteristic UUID: 0x0000BEEF-1212-EDFDE-1523-785FEF13D123
<b>Characteristic Value Declaration</b>	0x00X	Temp_charac: UUID found in the characteristic declaration value	Read only, no authentication, no authorization (it has to be configured)	Temperature and humidity values (in array of 4 bytes)
<b>Descriptor Declaration</b>	0x00X	CCCD: 0x2902	Read only, no authentication, no authorization	Notification enabled: 0x00-XX

Figure 4.12: Attribute table to define for the custom profile.

To add and configure the characteristic which has to be included in the created service, it is useful to refer to the attribute table. In figure 4.12 it is illustrated the attribute table that it should be implemented for the custom profile.

Actually, the depicted attribute table is suitable just for the code related to nRF52 DK programming since the integrated chip inside the Nordic board should dialogue just with one SHTW2 sensor.

To fit the table for the humidity monitoring systems that have been designed, it is indispensable to add more characteristics (one for each sensor). Hence, under the Service Declaration row, there must be five Characteristic Declaration, Characteristic Value Declaration and Descriptor Declaration fields for what concerns the first PCB system. While, the second PCB has to be programmed considering three repetitions of the same fields in the Attribute table.

**Add and Configure the Characteristic** First of all, the characteristic should have its own UUID, in the same way as the service in which belongs. The BASE UUID is always the same, while the short 16-bit UUID has been chosen to be the following:

```

1 #define BLE_UUID_OUR_BASE_UUID {0x23, 0xD1, 0x13, 0xEF, 0x5F, 0x78, 0x23, 0x15, 0
   xDE, 0xEF, 0x12, 0x12, 0x00, 0x00, 0x00, 0x00} // 128-bit base UUID
2 #define BLE_UUID_CHARACTERISTIC_UUID 0xBEEF // Just a random, but recognizable
   value

```

Listing 4.10: Characteristic UUID

Now, as it is shown in the listing 4.7, the last called function (`temp_char_add()`) adds a characteristic to the custom service. Actually, this is a custom implemented function and the actual Softdevice API which adds and declares the characteristic (and that, obviously, it has to be implemented inside the `temp_char_add()`) is the following:

```

1 sd_ble_gatts_characteristic_add(p_shtw2_service->service_handle,
2                               &char_md,
3                               &attr_char_value,
4                               &p_shtw2_service->char_handles);

```

Listing 4.11: API which adds the characteristic to the service

This API has four inputs: the first is the service handle and it is needed to say to the Softdevice where the characteristic has to be placed, the third and the second inputs are variables representing characteristic value attributes and characteristic metadata, respectively, and the last one is the characteristic handle which belongs to the service structure declared before [50].

The characteristic value attributes and the characteristic metadata are parameters that has to be initialized and configured. In more details, the most important variables to configure are: the attribute metadata, the characteristic metadata and the characteristic value attribute [50]. The first is a structure which says the permissions and the authorizations required by the characteristic value attributes [50]. For instance, it says if the characteristic value can be read or write. It also says where the characteristic value is stored (for this project, it has always been chosen to store it the Softdevice controlled part of memory) and its length [50]. The characteristic metadata is a structure which holds the characteristic value properties, the CCCD metadata and optional other descriptors [50]. At last, the characteristic value attribute is a structure which stores the actual characteristic value (temperature value, in this first case), the maximum length of this value and its UUID, as well.

The next step is to set up the read and write permissions to the characteristic value, which means that the value can be both read and write by the user. To do this, there are basically two things to do. The first is to configure the characteristic metadata properties inside the custom defined `temp_char_add()` function in this way:

```

1 //Add read/write properties to the characteristic
2 ble_gatts_char_md_t char_md; //characteristic metadata declaration
3 memset(&char_md, 0, sizeof(char_md)); //initialization to zero
4 char_md.char_props.read = 1; //set read property
5 char_md.char_props.write = 1; //set write property

```

Listing 4.12: Add read/write properties to the characteristic metadata

The second step is to actually set the read and write permission by adopting the following MACRO: `BLE_GAP_CONN_SEC_MODE_SET_OPEN`.

```

1 BLE_GAP_CONN_SEC_MODE_SET_OPEN(&attr_md.read_perm);
2 BLE_GAP_CONN_SEC_MODE_SET_OPEN(&attr_md.write_perm);

```

Listing 4.13: Add read/write permission to the characteristic value

The last step, to properly read or write the characteristic value, is to initialize and configure it. First of all, it has to be configured its length in bytes and its initial value. In this case, just for example, it has been chosen a length of 4 bytes and as initial value:0x12345678 [50].

```

1 //Configure the characteristic value attribute
2 ble_gatts_attr_t attr_char_value;
3 memset(&attr_char_value, 0, sizeof(attr_char_value));
4 attr_char_value.p_uuid = &char_uuid;
5 attr_char_value.p_attr_md = &attr_md;
6 // Set characteristic length in number of bytes
7 attr_char_value.max_len = 4;
8 attr_char_value.init_len = 4;
9 uint8_t value[4] = {0x12,0x34,0x56,0x78};
10 attr_char_value.p_value = value;

```

Listing 4.14: Set characteristic length and initial value

**Add CCCD** Another purpose is to have the possibility to make the peripheral device independent on sending periodically data to the client. To do this, it has to be added the Client Characteristic Configuration Descriptor (CCCD), which it's defined by the BLE Core Specifications (Vol.3: Part G, Ch 3.3.3.3) in this way:

*The Client Characteristic Configuration declaration is an optional characteristic descriptor that defines how the characteristic may be configured by a specific client [...].*

It means that this descriptor has to be write by the client which could decide if to enable/disable notification or indication. For this custom program, the notification property is the one that is going to be adopted.

So, first of all, CCCD metadata have to be declared and configured:

```

1 ble_gatts_attr_md_t cccd_md; //characteristic attribute metadata declaration
2 memset(&cccd_md, 0, sizeof(cccd_md)); //initialization to zero
3 BLE_GAP_CONN_SEC_MODE_SET_OPEN(&cccd_md.read_perm); //set read permission
4 BLE_GAP_CONN_SEC_MODE_SET_OPEN(&cccd_md.write_perm); //set write permission
5 cccd_md.vloc = BLE_GATTS_VLOC_STACK; //store in the Softdevice part
   of memory
6 char_md.p_cccd_md = &cccd_md;
7 char_md.char_props.notify = 1; //set the notification property

```

Listing 4.15: Configuration of the CCCD metadata.

**Update Characteristic Value** Finally, it's time to update the characteristic with the temperature and humidity values coming from the SHTW2 and SHTC3 sensors (depending on the system to program).

To this purpose, it has to be realized a `temphum_characteristic_update()` function in the `shtw2.c` file, which implements these actions:

```

1 void temphum_characteristic_update(ble_shtw2_t *p_shtw2_service, int32_t *
   temphum_value)
2 {
3     //Update characteristic value
4     if (p_shtw2_service->conn_handle != BLE_CONN_HANDLE_INVALID)
5     {
6         uint16_t len = 4;
7         ble_gatts_hvx_params_t hvx_params;
8         memset(&hvx_params, 0, sizeof(hvx_params));
9
10        hvx_params.handle = p_shtw2_service->char_handles.value_handle;
11        hvx_params.type = BLE_GATT_HVX_NOTIFICATION;
12        hvx_params.offset = 0;
13        hvx_params.p_len = &len;
14        hvx_params.p_data = (uint8_t*)temphum_value;
15
16        sd_ble_gatts_hvx(p_shtw2_service->conn_handle, &hvx_params);
17    }
18 }

```

Listing 4.16: Update of the Characteristic Value

`hvx_params` is a structure which holds parameters for the notification property [50]. Here, the main API is `sd_ble_gatts_hvx()` which checks the CCCD value to know if indication or notification has been activated by the client. It can also automatically update the attribute value, by means of a single API call.

Finally, it must be provided data to send. To do this, a timer must be initialized, configured and started. Therefore, a timer handler function is the one responsible in triggering the notification events and in calling the `temphum_characteristic_update()` function which updates the characteristic value.

## 4.3 Results

As already introduced, the code, regarding I2C communication and BLE transmission, was fitted for three cases:

- Communication between an external SHTW2 sensor and the Nordic nRF52 Development board.
- Humidity monitoring system 1: the integrated MCU should dialogue with five SHTW2 sensors.
- Humidity monitoring system 2: the MCU has to communicate with three SHTC3.

### 4.3.1 Nordic nRF52 Development Board and SHTW2 sensor

At first, the I2C program has been tested by means of Nordic nRF52 DK and an external SHTW2 sensor. For sake of simplicity, it has been considered just one sensor which is positioned in a

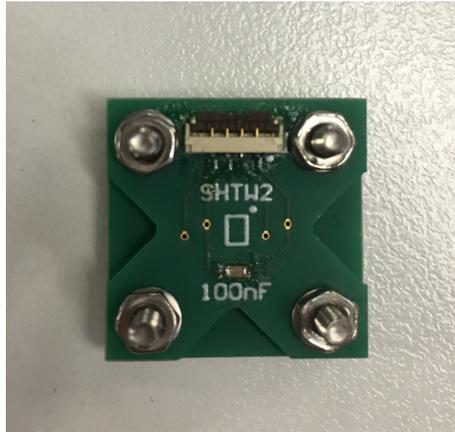


Figure 4.13: SHTW2 humidity sensor used to test the I2C program by the Nordic nRF52 DK board.

small board (together with its decoupling capacitor), as illustrated in figure 4.13. It has been used this sensor just for convenience, since it was already available at the Imperial College laboratory because a PhD student had exploited it, before, for his purposes.

Before starting doing measurements, it is indispensable to properly set the components and the connections between them. The main elements are: Nordic nRF52 development board, SHTW2 sensor and a level shifter. The last one is fundamental since the sensor must be powered at 1.8 V, while the nRF52832 chip inside nRF52 DK works at about 3V. The level shifter adopted is the TXS0104E, described in appendix A.9.

The schematic of the connections between all components is shown in figure 4.14.

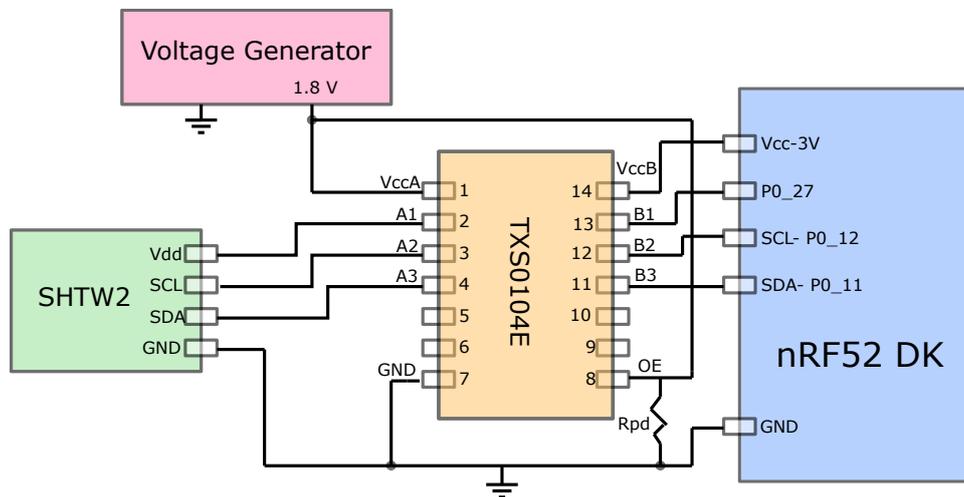


Figure 4.14: Schematic of the components connections

To test the obtained result, it has been exploited *BlueLight* iPhone application, developed by *PouchThrough*. The connection establishment with the peripheral device is shown in figure 4.15.

As, it can be noticed, at the connection, it is possible to read the profile name, the advertising data, the Service UUID and its characteristic UUID.

So, once that the connection was established, it's time to start read measures.

Characteristic value field is a 4-byte number expressed in a hexadecimal format: the first two bytes represent the relative humidity measure, while the other two bytes show the temperature value.

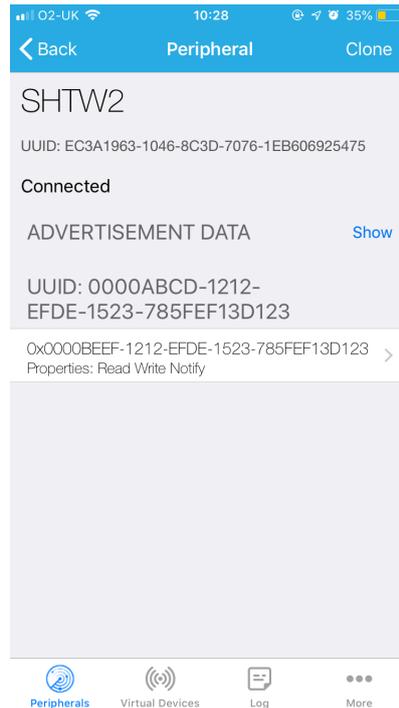


Figure 4.15: SHTW2 profile with one service and one characteristic, view from *Light Blue* iPhone application.

More exactly, the measures must be read byte per byte. A brief explanation on how to properly read the outputs from the sensor is depicted in figure 4.16.

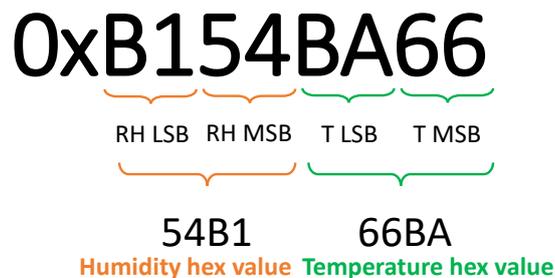


Figure 4.16: Example of 4-byte characteristic value in hexadecimal format representing the outputs from the sensor for relative humidity and temperature measures.

Finally, let's start reading some measurements. In figure 4.17 it is illustrated *BlueLight* application when the measurements are transmitted.

After having elaborated the hexadecimal numbers, first by converting them in a decimal format, then by applying equations 4.1 and 4.2, the obtained results are depicted in figure 4.18.

The system has been programmed to do a new measurement every three seconds, so that the obtained results over time are illustrated in figure 4.18. In figure 4.18(a) it is shown how the temperature is always around 25°C while, as it is possible to observe by looking at figure 4.18(b), the relative humidity is about 32%. More exactly, calculating the measures means (assuming constant values), it results that the temperature was 24.8°C ±0.4 °C, while the mean relative humidity was 32% ±3%.

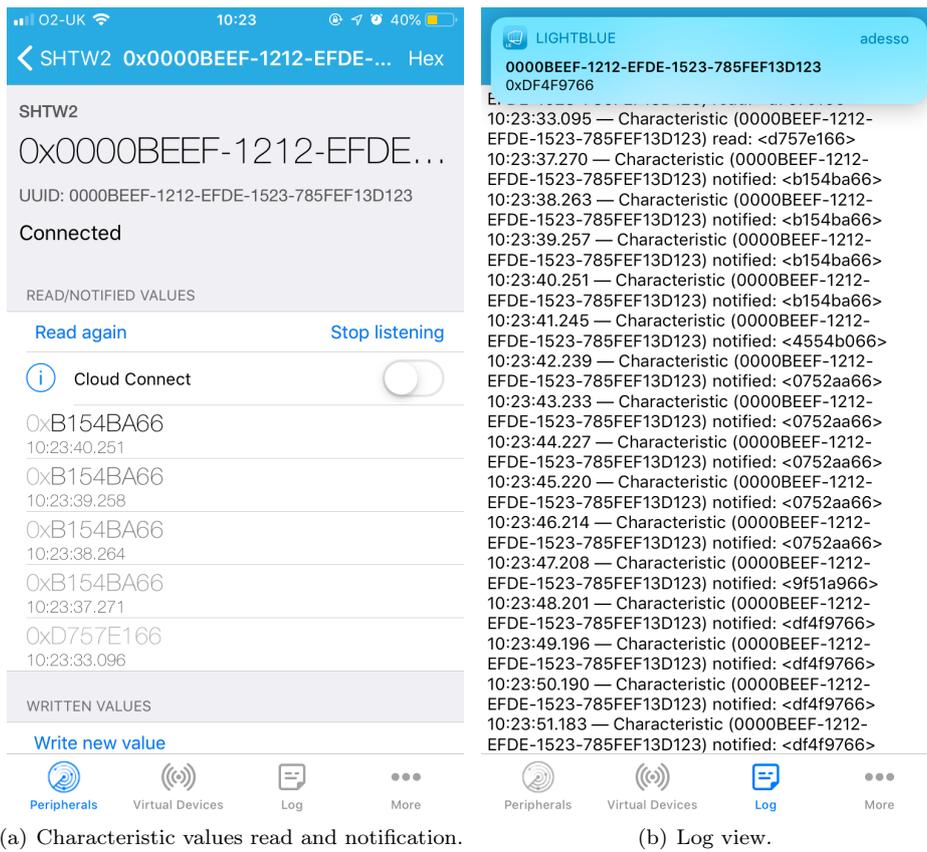


Figure 4.17: Sensor measurements sent by Bluetooth to the iPhone App.

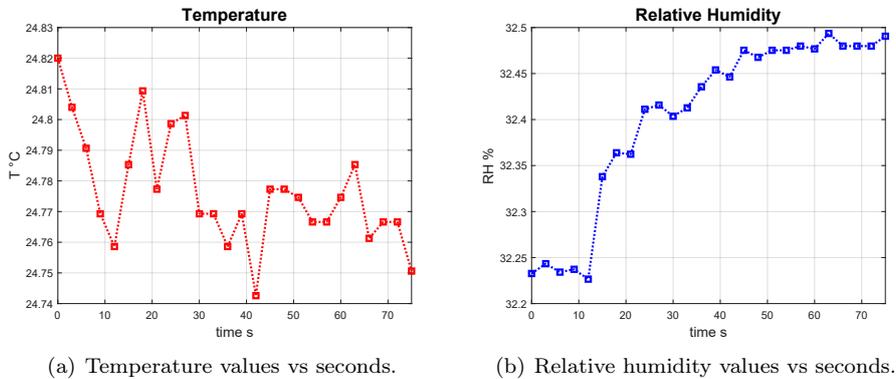


Figure 4.18: Elaborated values from the SHTW2 outputs.

### 4.3.2 Custom PCAs Serial Wire Debug (SWD)

The nRF52832 is a general-purpose multiprotocol SoC. It is built around an Arm® Cortex™-M4 CPU with floating point unit running at 64 MHz.

This ARM technology gives the possibility to debug this chip by means of the so called Serial Wire Debug Port (SW-DP). A Debug Access Port (DAP) allows the debugger access inside the whole System-on-Chip.

Usually, it is possible to access to the DAP by a standard JTAG or by a Serial Wire Debug (SWD) mode. In this way, some main functionalities that can be implemented are: memory and registers access, breakpoint and trigger settings, flash programming.. and so on [51].

This port uses the SWD protocol to access the DAP. As already said, nRF52832 has available a Serial Wire Debug Port (SW-DP) which exploits the SWD protocol to access the DAP. This

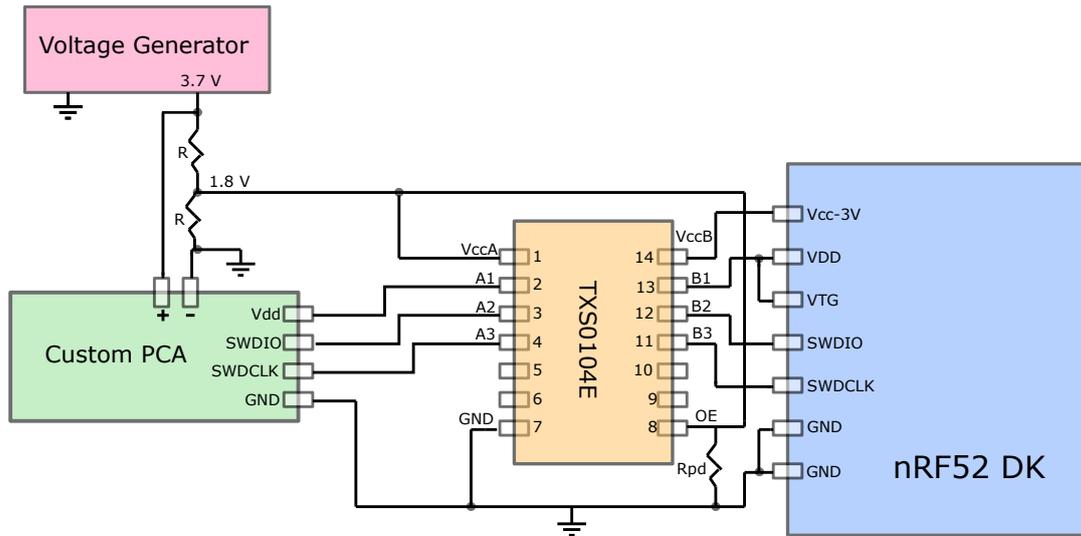


Figure 4.19: Serial Wire Debug (SWD) schematic.

works with just two lines: Serial Wire Data Input Output (SWDIO) and Serial Wire Debug Clock (SWDCLK) which represents the clock signal to target CPU.

The debugging happens by means of the Nordic development board nRF52 by connecting 4 main wires: power supply VDD, SWDIO, SWDCL and common Ground. As before, the problem is that the System-on-Chip in the board is working at about 3V, while the custom system should be powered by 1.8V. Therefore, even in this case, it is mandatory to adopt a level shifter. The schematic showing all the hardware connections for the debugging is shown in figure 4.19.

Both the humidity monitoring systems have been programmed following these indications. Then, in the next sections, the measurements results coming from the two systems will be presented.

### Humidity Monitoring System 1

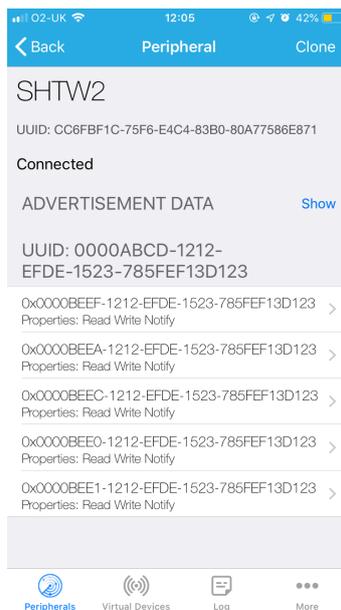


Figure 4.20: SHTW2 profile with one service and five characteristics, view from *LightBlue* iPhone Application.

The first PCA system has been programmed by triggering a measurement each 10 seconds. This circuit supply five SHTW2 sensors, therefore the application profile has to be implemented con-

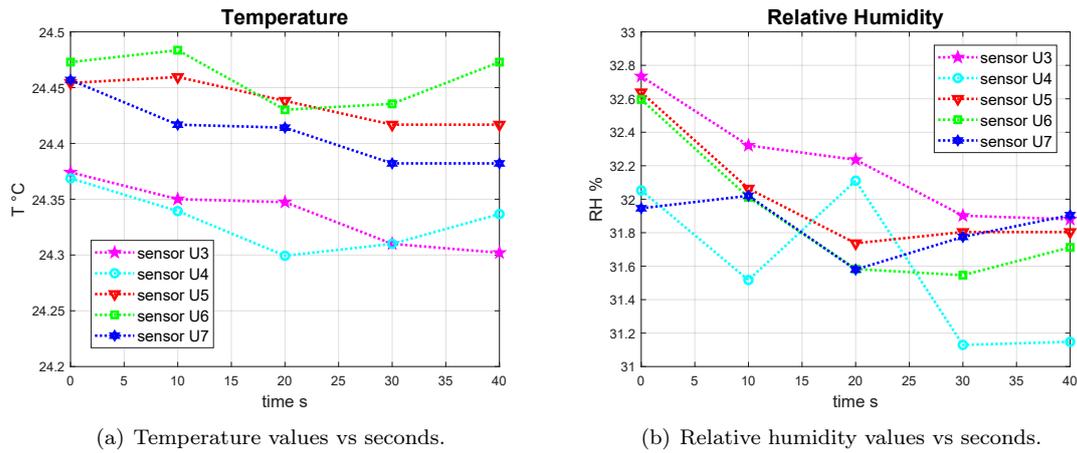


Figure 4.21: Humidity Monitoring System 1: Elaborated values from the 5 SHTW2 sensors outputs.

sidering 5 characteristics, all provided with their own CCCD set for notification capability. The 5-characteristic set up, during connection phase, is shown in figure 4.20. By accessing to all the 5 characteristic value fields and after collecting and elaborating each sensor outputs, the obtained results for temperature and relative humidity measures are illustrated in figure 4.21. Summing up, they show that, in that day, in the London workspace room, there were about 24.5 °C with 32 % of relative humidity.

### Humidity Monitoring System 2

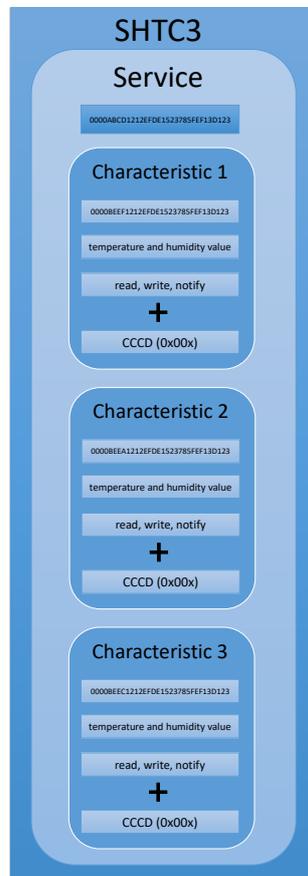


Figure 4.22: SHTC3 profile with one service and three characteristics set up.

As for the previous system, the programming procedure is always the one depicted in figure 4.19. In this case, it has been set to execute a new I2C cycle each 10 seconds. Moreover, the environment is no more the Centre of Bio-Inspired Technology research workspace, but the anechoic chamber in its laboratory.

Humidity Monitoring System 2 has available three SHTC3 humidity sensor, therefore the SHTC3 profile to set up is the one depicted in figure 4.22.

To connect to the device, this time it has been chosen to exploit *nRF Connect* which is a PC application. It works by means of nRF52 DK board which becomes a master, while the device under test is a peripheral. The connection establishment is shown in figure 4.23.

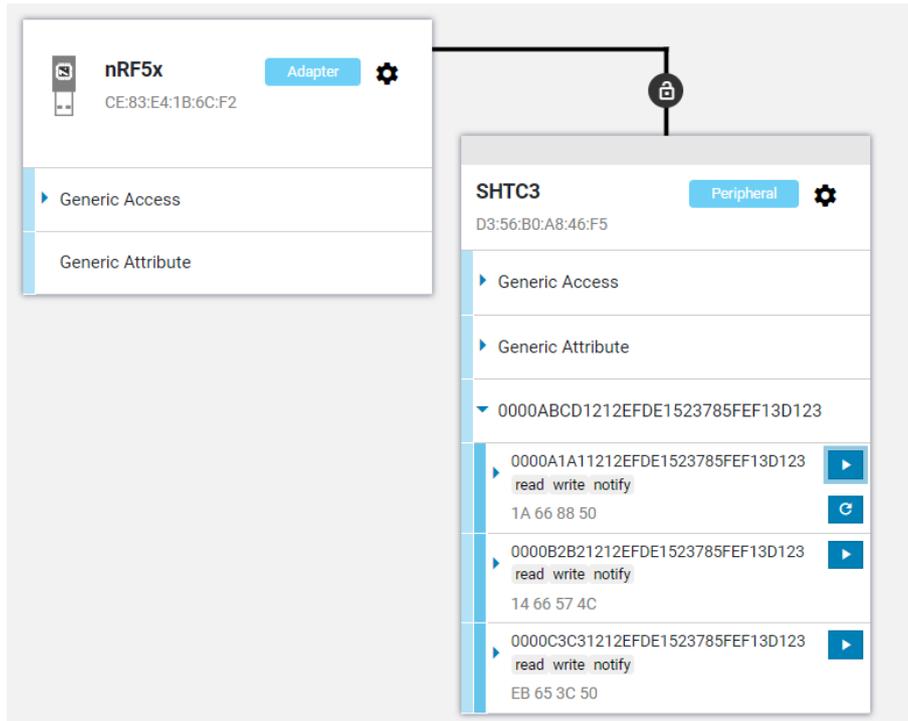
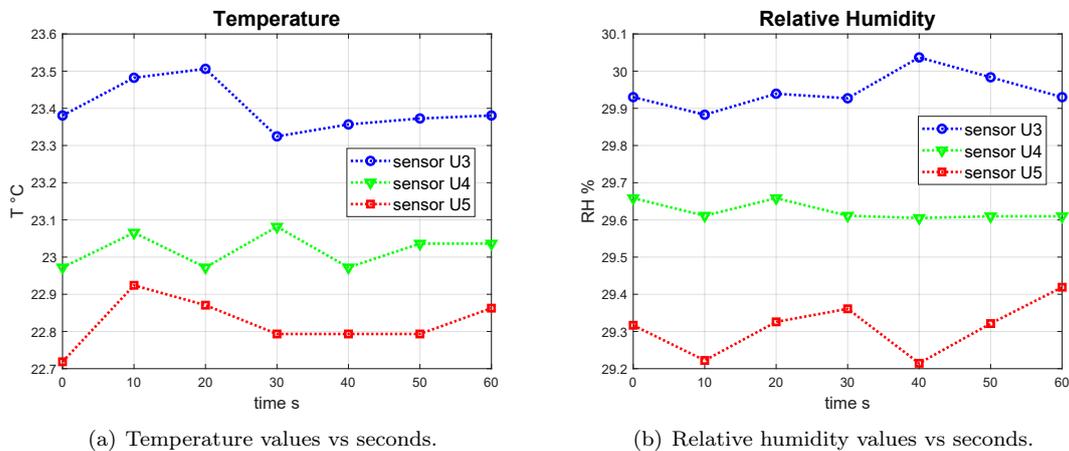


Figure 4.23: Bluetooth Connection with SHTC3 device, by *nRF Connect*.

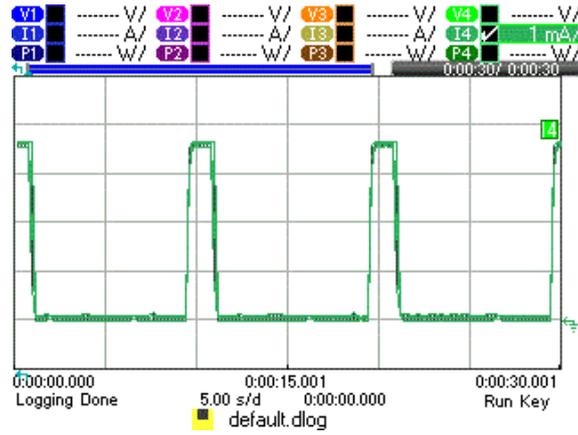
In the same way, after having elaborated the sensor outputs by applying equations 4.2 and 4.1, the obtained measures are illustrated in figure 4.24. They state that there were about 23 °C with 30% of relative humidity.



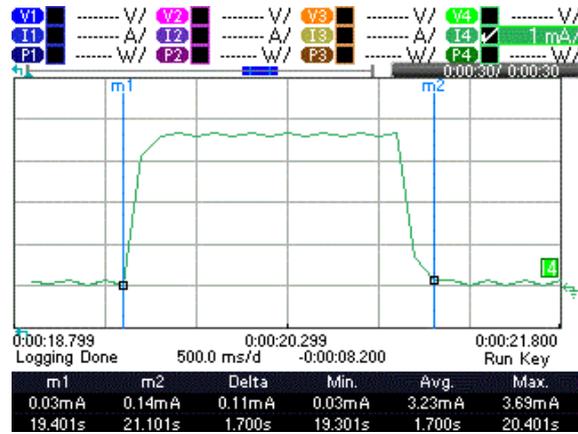
(a) Temperature values vs seconds.

(b) Relative humidity values vs seconds.

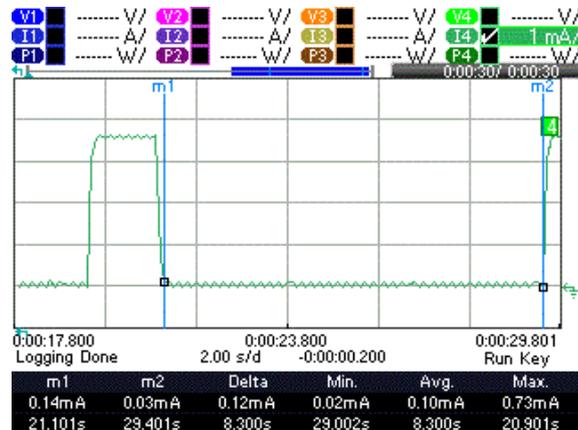
Figure 4.24: Humidity Monitoring System 2: Elaborated values from the 3 SHTC3 sensors outputs.



(a) Current profile during SHTC3 activity



(b) Current profile during I2C measurements time interval.



(c) Current profile during CPU idle state.

Figure 4.25: Current consumption scope views.

**Power Consumption** For SHTC3 device it has been performed a measure of the power consumption to see how long the selected battery should last. The aim is to use a LiPO secondary battery, that is the one generally adopted for implantable devices systems, but to not recharge it. In other words, the idea is to make the monitor system low power enough to never discharge completely the battery.

To measure the power consumption, it has been monitored the current spent to power the humidity monitoring system activities. It has been exploited the N6705A DC Power Analyzer and the PCB has been programmed to do a new I2C cycle at each 10 seconds. The current consumption over time is shown in figure 4.25(a). It is evident that it can be considered a periodic function with period of 10 s, as expected. The peak of current is due to sensor measurements and it lasts 1.7s, as it

can be observed in figure 4.25(b) and it reaches 3.69 mA as maximum value. During the idle state (that is the resting 8.3s), the maximum current that is employed is 0.73mA, as figure 4.25(c) shows.

The actual purpose of the designed humidity monitoring system is to do a measure every 90 minutes (it will be explained in chapter 6). Therefore, the current consumption has been measured considering the same current function shown in figure 4.25(a), but with a period of 90 min. In this way, it results that each cycle of 90 minutes consumes about 1085.933 mAs. Considering that the battery capacity is 165 mAh (594000 mAs), it turns that it allows to complete 546 cycles in total. 546 cycles of 90 minutes correspond to about 34 days, that are largely enough for this project testing procedure.

## Chapter 5

# 3D Printed Package

3D printing is a manufacturing process based on building structures, layer by layer, starting from a digital file representing the CAD model design. It's an additive process, which means that each layer is fabricated just by adding material: the features resolution is not achieved by an etching procedure, as happened, for instance, in Lithography which is, in fact, a subtractive manufacturing process. Indeed, there is a very large amount of different 3D printing technologies. They differ in materials which are allowed to print, in uses where they have been adopted, in price points and in building mechanisms [52].

For this project, the focus will be only on the technology which leads the 3D printer available in the Bio-Inspired Technology Centre laboratory of Imperial College of London. The 3D printed employed is *Figure 4 Standalone*, developed by *3D Systems, Inc.*

### 5.1 Figure 4 Standalone Printer

By *3D SYSTEMS* brochure, Figure 4 Standalone printer is defined as an *ultra fast, affordable industrial 3D printer*. The affordability is given by the low price and the industrial-grade durability, moreover it has a fast throughput speed and it can be used with a large amount of materials. Its versatility allows to use it for a large variety of applications as functional prototyping, design verification, digital texturing applications, casting patterns, rapid iteration, end-use parts for low volume production, jewelry, replacement parts, master patterns, fixtures, jigs and rapid tooling of molds [53]. The 3D printing technology in which it is based on is called DLP - Digital Light Processing (section 5.2). After the printing step, it requires a post-processing phase which, more exactly, is a UV post-curing, needed to give the expected mechanical properties to the created objects.

### 5.2 DLP - Digital Light Processing

Digital Light Processing is a 3D Printing Technology based on Vat polymerization, such as SLA Stereolithography.

The most common 3D printing technique is FDM - Fused Deposition Modeling and it is based on material extrusion, which means that each layer is fabricated by depositing step by step small samples of fused thermoplastic materials where are required. Vat Polymerization, instead, has its building principle in adopting a liquid photopolymer resin which solidifies when illuminated by a light source. Thus, each layer has been fabricated by illuminating, step by step, a 2D pattern on the liquid representing the cross-section of the layer.

The most common Vat Polymerization technique is SLA or stereolithography. SLA fabricates objects by tracing, layer by layer, cross-sections of the structure on the surface of the liquid photopolymer by a laser beam [52]. An elevator table should be below the liquid surface at depth equal to the already fabricated structure plus light absorption limit, such that, at each step, it can move toward the bottom to allow another layer to be fabricated. The laser beam is deflected by

galvanometer-driven mirrors, so that it scans the surface of the liquid to create the desired pattern [52]. In the end, to completely solidify the structure, a post curing must be performed.

DLP is similar to SLA: the main difference is the type of light source used to cured the photopolymer. In fact, DLP doesn't employ a laser beam, but a digital light projector screen. The procedure is well explained in figure 5.1. There, it is easy to observe a projector in the bottom which illumi-

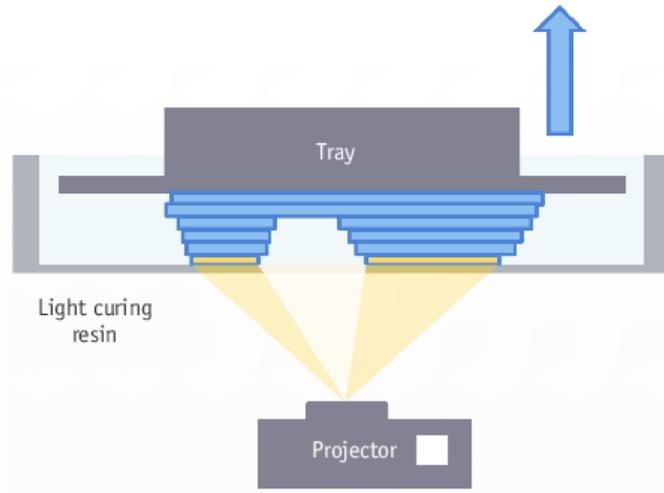


Figure 5.1: Digital Light Processing explanation drawing [10].

nates at each step the whole 2D image, on the liquid resin surface, representing the cross-section of the layer to cure. Then, the solidified part is moved upwards, so that the projector can flash the following layer cross-section image on the liquid resin. On the contrary of SLA, the resin units at each layer solidifies simultaneously, while in SLA process, the material should be cured "point to point" by means of the laser beam [54]. As a consequence, DLP is generally faster than SLA. The projector is a LED screen made by hundreds of thousands or even millions micromirror devices which constitute the, so called, Digital Micromirror Device (DMD). They should direct the light and create the desired 2D pattern of the layer. The image projected is made of pixels, which become voxels in the corresponding cured resin. The resolution of the 3D printer depends on the number of micromirrors involved to create the DMD [54].

Regarding Figure 4 Standalone, the maximum resolution is: 1920 x 1080 pixels. The Pixel Pitch is 65 microns and the light wavelength is 405 nm [55].

## 5.3 Material

For medical applications, Figure 4 Standalone printer gives the possibility to choose among two polymeric materials: Figure 4 MED-AMB 10 and Figure 4 MED-WHT 10. They are suitable for medical applications because of their properties to be biocompatible, thermal resistant and suitable for sterilization by autoclave. Biocompatibility of these materials is based on testing by *3D Systems* on a single geometry and sample set, following ISO 10993-5 and -10 guidelines.

To realize the package to test it has been selected the MED-AMB 10 resin.

### 5.3.1 Figure 4 MED-AMB 10

Figure 4 MED-AMB 10 is a rigid, biocompatible, translucent material. The main properties of the liquid resin are listed in table 5.1, while in table 5.2 the mechanical characteristic of the post-cured material have been reported.

**Water Absorption ASTM D570** From table 5.2 it is present the Water Absorption parameter, that could be interesting to well analyse since the purpose of this project is to develop a suitable

Measurement	Condition	Value (Metric)
Viscosity	@25°C	1138 cps
Liquid Density	@25°C	1.12 $g/cm^3$
Layer Thickness (Standard Mode)		0.05 mm
Vertical Build Speed (Standard Mode)		43 mm/hr
Vertical Build Speed (Draft Mode)		63 mm/hr

Table 5.1: Liquid Figure 4 MED-AMB 10 material properties, from [16].

Measurement	Condition	Value (Metric)
Solid Density ( $g/cm^3$ )	ASTM D792	1.20
Tensile Modulus (MPa)	ASTM D638	2760
Heat Deflection Temperature @0.45 MPa	ASTM D648	119 °C
Heat Deflection Temperature @0.45 MPa	ASTM D648	94 °C
Glass Transition ( $T_g$ )	ASTM E1640	110 °C
Coefficient of Thermal Expansion (CTE) (ppm/°C) @ < $T_g$	ASTM E831	84
Coefficient of Thermal Expansion (CTE) (ppm/°C) @ > $T_g$	ASTM E831	177
Water Absorption (24 hour)	ASTM D570	0.26 %

Table 5.2: Mechanical Properties of the Post-Cured Figure 4 MED-AMB 10 material, from [16].

3d printed object impermeable to water and gasses. Water Absorption parameter gives a value for the amount of water absorbed under specific conditions [56]. The ASTM D570 conditions explain how to calculate it. In the specific, the sample has to be dried in a oven (for a given time and temperature) and, next, it has to be desiccated to cool. Instantly after, the sample is weighted. The second step of the procedure is to submerge the sample in water (generally at 23°C for 24 hours) and then to weight it again [56]. The Water Absorption is expressed as a percentage of the increase in weight [56]:

$$\text{Water Absorption (\%)} = \frac{\text{Wet Weight} - \text{Dry Weight}}{\text{Dry Weight}} \times 100$$

## 5.4 CAD Model Design and Fabrication

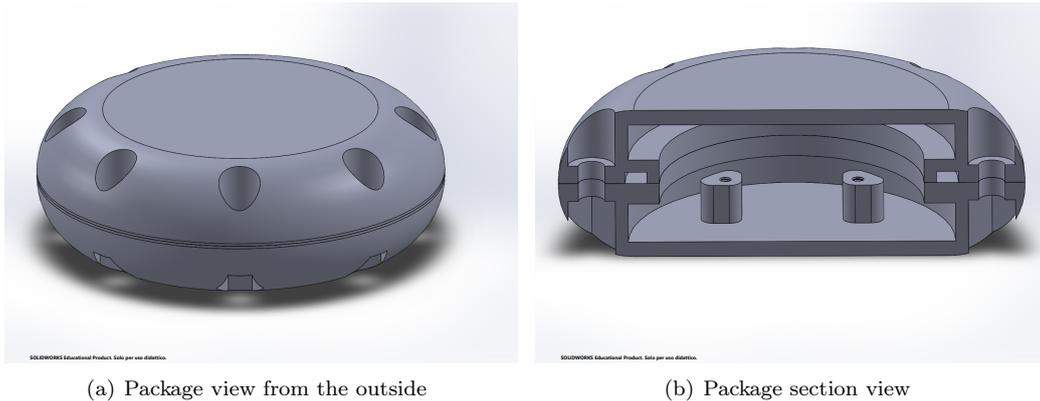


Figure 5.2: Package CAD Model

The package has been thought as a simple box for the humidity monitoring electronic circuit board that has been fabricated and assembled. The idea is to make it as much as possible rounded and without hard edges to imitate at best an actual implantable device shape. It was designed by means of *SolidWorks 2019* and, after that, the CAD file was transferred to *3DSprint* which is a software platform needed to initiate a printing process by Figure 4 Standalone 3D printer. The CAD model is shown in figure 5.2, where, in subfigure 5.2(a), the package is shown from an external point of view, while subfigure 5.2(b) illustrates its internal section view.

Looking at the section view (figure 5.2(b)), it is possible to observe that, inside the structure, 4 supports have been included. They need to sustain the PCB which is thought to be fixed by means of a screw at one of the corner. In fact, each corner is provided with a 8 mm depth hole (diameter = 1.6 mm) where a drill profile has been designed. The screw needed to fix the PCB is of M2x0.4 type and its length is 5 mm.

#### 5.4.1 Sealing method

To close the package, it has been chosen a mechanical sealing method. The top and the bottom parts have been closed together by means of 8 screws and nuts. The dimension chosen for them is M4x12mm. The hermetic seal has been achieved by exploiting an EPDM gasket in proximity to the closure. In figure 5.2(b) it is possible to notice a rectangular-section slot where it has to be placed the rubber ring (o-ring) to make hermetic the closing. The description of the slot is depicted in figure 5.3.

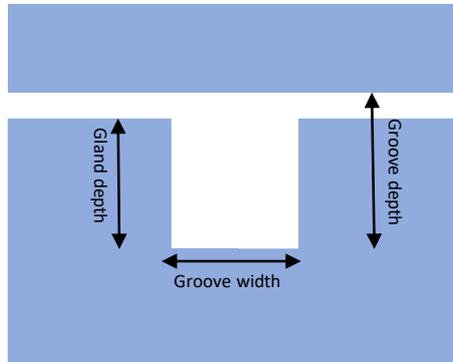


Figure 5.3: O-ring slot

To properly close the package, the parameters chosen for rubber ring slot are itemized in table 5.3 and it has been selected an EPDM o-ring whose internal diameter is 45 mm and whose cross-section is 2 mm.

<b>Groove diameter</b>	46.11 mm
<b>Groove width</b>	3.44 mm
<b>Gland depth</b>	1.92 mm
<b>Groove depth</b>	1.92 mm

Table 5.3: O-ring slot parameters

EPDM (ethylene-propylene diene monomer [M-class] rubber) is a type of synthetic rubber [57] which exhibits good properties in terms of versatility and durability. It shows a good resistance against heat and weather, which makes it a good candidate for medical devices, automotive manufacturing, water and wastewater systems, household appliances and other applications [57]. To achieve a good hermetic seal, the EPDM gasket has to be stretched and squeezed in its slot. For this design, which represents a static application characterized by an axial squeeze, it is strongly recommended to never overcome the 5% of stretching and, generally, a good o-ring squeezing could be about of 30% [58]. The formula to take into account in o-ring sealing design are the following [58]:

$$\text{O-ring I.D.} = \frac{\text{Groove diameter}}{\% \text{ of stretch desired} + 1} \quad (5.1)$$

$$\text{O-ring C.S.} = \text{Gland depth} \times (\% \text{ of squeeze} + 1) \quad (5.2)$$

Where I.D. stays for internal diameter and C.S. for cross section.

Hence, from the chosen dimensions, it results that the stretch and the squeeze are about, respectively, of the 2.5 % and 30.2 %.

### 5.4.2 Package fabrication

Finally, the package was manufactured in Med-AMB10 material and the photograph which shows the fabricated structure together with EPDM sealing ring and the humidity monitoring PCB is illustrated in figure 5.4. For all the executed tests only the second humidity monitoring system has been exploited.

Now, everything is ready to start doing accelerated tests monitoring moisture ingress inside the fabricated structure.



Figure 5.4: Photograph of package internal parts with PCB and o-ring.

Once closed, the package under test is shown in figure 5.5.



Figure 5.5: Photograph of the final sealed package with PCB and battery inside

## Chapter 6

# Hermeticity testing

Once the package sample and the humidity monitoring board have been realized, a reliability experiment has to be conducted. The aim is to estimate the hermeticity level of the 3D printed package structure by performing a test that reproduces the same conditions present in human body organism. It is common to represent it by a saline solution at  $37^{\circ}\text{C}$ , kept quite constant, in which the package under test must be immersed. Since the purpose is to quantify the moisture ingress rate for a long-term implantation, that is more than 30 days, actually an accelerated testing is required to short the experiment time period.

To setup the testing procedure, it has been followed the experimental result, based on Arrhenius equation, which says that a temperature raise of  $10^{\circ}\text{C}$ , it is equivalent to consider the device lifetime divided by half [59]. Actually, by increasing more and more the temperature, the rule validity gets worse due to other problems caused by high temperature values.

As a consequence, it has been set an experiment environment at constant  $77^{\circ}\text{C}$ , which means to reduce the experiment time by 16 times. In this way, 90 minutes at  $77^{\circ}\text{C}$  correspond to one day at  $37^{\circ}\text{C}$ , while 6 months at  $37^{\circ}\text{C}$  are 10 days at  $77^{\circ}\text{C}$ .

### 6.1 Experiment setup

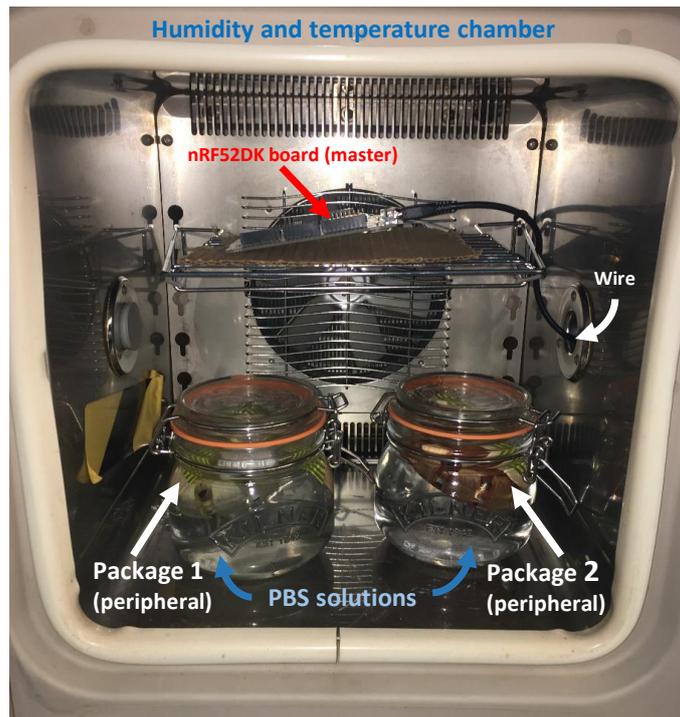


Figure 6.1: Experiment set-up inside *ESPEC* SH-221 humidity chamber.

To maintain constant temperature, it has been exploited, as test environment, the *ESPEC* SH-221 temperature and humidity. This machine has been programmed to run at constant temperature and relative humidity, respectively of 77°C and 5%.

The saline bath has been prepared inside a *Kilner* Clip Top Round Storage Jar of 0.5 litre. It has been chosen this kind of container, since it is finished with an orange rubber seal to not let the water inside to evaporate. In the end, the package under test has to be immersed inside the jar for all the test duration. Indeed, to accurately represent body fluids, it has been prepared a PBS solution. Phosphate-buffered saline (PBS) is a buffer water-based salt solution based on sodium chloride, disodium hydrogen phosphate and, in some formulations, potassium dihydrogen phosphate and potassium chloride. The buffer is necessary to maintain ph constant [60]. PBS is commonly exploited in biology since it represents quite closely the transportation of tissues in human body.

The adopted humidity chamber isn't provide of a glass window to let RF signals to pass, but it is possible to safely link wires and connections to it from the external. Therefore, it has been decided to put inside the chamber the Nordic nRF52DK board, as well, to perform master role for Bluetooth communication. The Nordic board was connected via an USB wire with computer in order to read the sensor measurements through *nRF Connect* application. The humidity monitoring system has been programmed to execute a new measurement cycle every 45 minutes, which is equivalent to measure temperature and humidity twice a day at 37°C.

The experiment setup is illustrated and described in figure 6.1, where it is shown a photograph of the humidity chamber cavity before starting the experiment.

## 6.2 Test 1

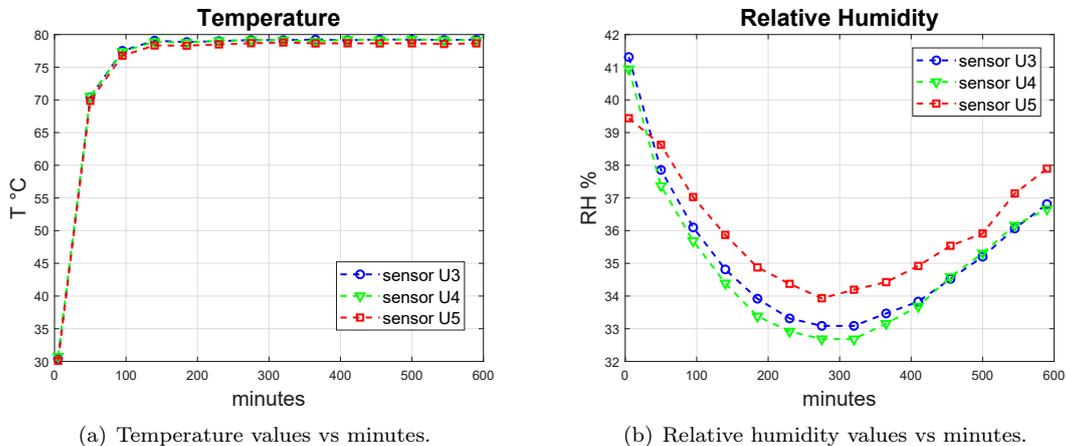


Figure 6.2: Experiment 1: elaborated values from the 3 SHTC3 sensors outputs.

The experiment has been started considering just one package under test, mechanically sealed by an EPDM o-ring and by eight screws, as described in the 3D printing chapter. The first measurement, that has been collected, happened 5 minutes after that the package was put inside the chamber. As it can be observed in figure 6.2(a), the system took about 150 minutes to reach 77°C, therefore, just after the third measurement, it is possible to consider the temperature constant, as the testing procedure requires. Looking at figure 6.2(b) it is possible to observe the relative humidity evolution over time inside the package cavity. The relative humidity starts at about 40%, then it decreases until reaching the 33% and, next, it starts increasing. The rising begins after 320 minutes and until this moment, we can definitively state that there is no moisture ingress inside the package. Unfortunately, the test ended after 590 minutes due to the discharge of the adopted LIPO battery, probably because, at the beginning of the experiment, it wasn't at its maximum charge level. Therefore, the Bluetooth connection has been lost.

Considering the equivalent time at 37°C, the end of this first test corresponds to a 5 days and half of implantation time. Analysing the humidity function, it can be stated that moisture starts to increase just after two days and half of implant time period, even if it remains quite constant until the test end.

### 6.3 Test 2

Since the first test was terminated due to battery problems, a new experiment has been performed changing the power supply. The package structure adopted is always the same as the previous test and, in this case, the whole experiment lasted almost 5 days (more exactly 113 hours and 30 minutes).

The Bluetooth connection happened by exploiting *nRF Connect* computer application and its representation is illustrated in figure 6.3. The Bluetooth Profile was called *SHTC3*, as sensors name, and the three characteristics were storing temperature and humidity values coming from the corresponding sensor.

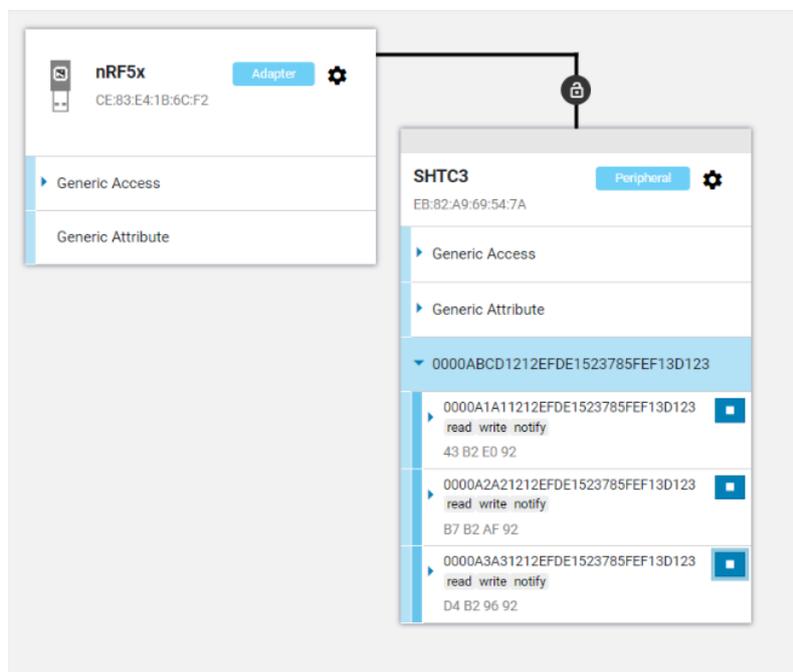
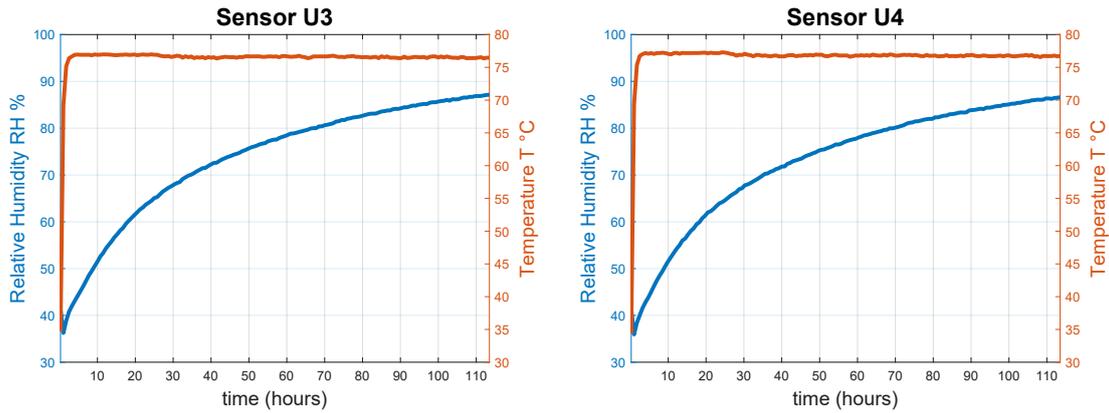


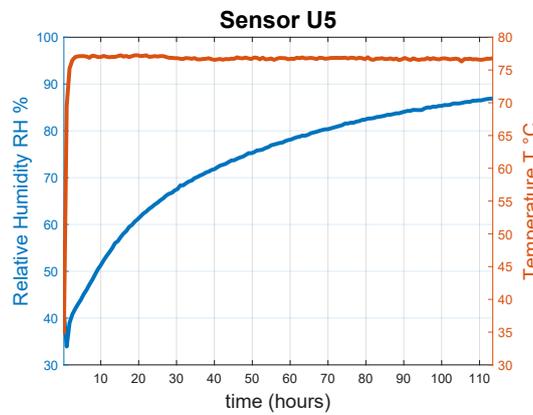
Figure 6.3: Connection between Nordic nRF52DK board (master) and the custom designed PCB module (peripheral).

The three SHTC3 sensors temperature and humidity measurements are shown in figure 6.4.

The first measurement was executed after 15 minutes inside the humidity chamber. The temperature was about 34°C and the relative humidity 41%. Just after 150 minutes, the temperature reached 77°C and then, it remained constant for all the test duration. After 150 minutes, the relative humidity was about 40%. In the time interval between 15 and 150 minutes, the humidity decreases due to the fast rising of temperature, but, next, it shows a monotonous rising behaviour which means that the moisture is allowed to enter inside the cavity just from the beginning. The experiment was carried on to see how fast the humidity increases and how much the device can support the leakage without damaging. As just said, the package under test was monitored for 5 days, which correspond to three months at 37°C. Test 2 has been stopped after that the humidity reached the 87%, then, the package has been removed from the humidity chamber and the humidity monitoring system 2 has been taken off to be able to exploit it once more for a next experiment. As it can be noticed, looking at figure 6.4, the relative humidity increases quite fast in the first test hours, but, then, it increases more and more slowly. This is quite reasonable, since it could



(a) Sensor U3 temperature and relative humidity values vs hours. (b) Sensor U4 temperature and relative humidity values vs hours.



(c) Sensor U5 temperature and relative humidity values vs hours.

Figure 6.4: Experiment 2: elaborated values from the 3 SHTC3 sensors outputs.

indicate that the water vapor entered inside the structure exercises an higher and higher pressure which may slow down a further moisture ingress.

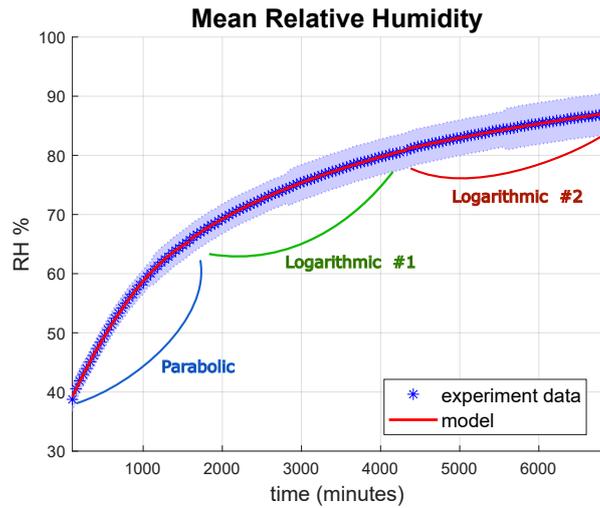
### 6.3.1 Estimated life-time

The analysis was carried on by trying to fit the humidity growth inside package cavity. At first, considering all the three sensors measurements, it has been calculated the mean relative humidity over time. Thus, the function has been approximated by a parabolic growing in a first phase (until the relative humidity reaches 65%) and, then, it has been considered a logarithmic increase until the end of the experiment. Indeed, since for sake of simplicity it has been exploited a linear regression fitting, the logarithmic model has been divided in two parts, due to the fact that the growing is more and more slowly. In details, the first logarithmic function starts at 65% and ends at 81% of relative humidity, while the second logarithmic fitting function is assumed to continue till the end. The mean relative humidity compared with the calculated fitting model is illustrated in figure 6.5(a).

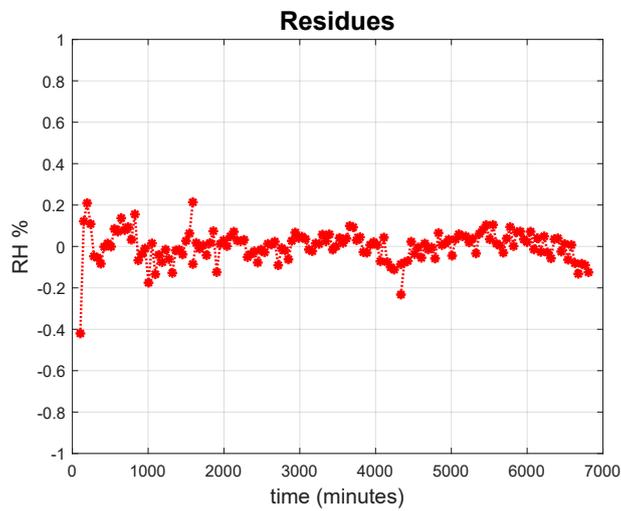
Going into mathematical details, the assumed fitting functions are the following:

$$y(x) = \begin{cases} ax^2 + bx + c, & \text{for } 40 \leq y \leq 65 \\ \alpha \log(x) + \beta, & \text{for } 65 \leq y \leq 81 \\ \gamma \log(x) + \delta, & \text{for } 81 \leq y \leq 100 \end{cases}$$

where  $y$  is the relative humidity expressed as a percentage and  $x$  is the time vector. Solving the linear regression fitting problem, the parameters that have been found are summarized in table 6.1.



(a) Fitting functions of the mean relative humidity measures.



(b) Residues analysis.

Figure 6.5: Experiment 2: fitting function and experimental model of the relative humidity evolution.

a	$-7.0351 \times 10^{-6}$
b	0.0296
c	36.1412
$\alpha$	15.1391
$\beta$	-45.8362
$\gamma$	13.0945
$\delta$	-28.5668

Table 6.1: Estimated parameters of the linear regression models for relative humidity evolution.

The residues analysis is shown in figure 6.5(b). By looking at their behaviour, it is reasonable to assume that the first two fitting functions are good models for relative humidity growing, that is quite straightforward since the training points are a lot for a small time interval. Unfortunately, residues for the last points show a non linear behaviour which actually can't be approximated by the function considered for this analysis. Actually, by experience, it seems that the logarithmic function changes over time, slowing down more and more. Anyway, just to do a simple approximation, the second logarithmic fitting model has been considered to last until that the system achieves 100%.

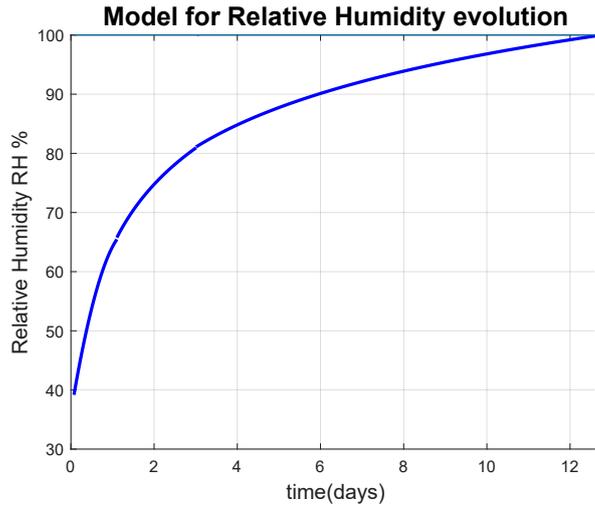


Figure 6.6: Calculated relative humidity evolution model over time.

The estimated model, considering future times, is depicted in figure 6.6. This forecasting function says that the package under test will reach 100% of relative humidity after 12 days of experiment, that are equivalent to 6 months at 37°C. At 100% of relative humidity, the environment temperature becomes equal to the dew point temperature, so that the water vapor starts to condensate. The liquid phase of water, joined with PBS, is highly conductive, so that it will lead the electronic board to a catastrophic failure.

Indeed, all of this can be quite reasonable when dealing with systems at constant temperature. In fact, the actual enemy for electronic systems is water condensation which, at fixed temperature, it is achieved at 100% of relative humidity, when the environment temperature becomes equal to the dew point. However, it is demonstrated that there could be corrosion accelerating factors, for instance ionic contamination, which could be easily introduced through PCBA assembly process. Ionic contamination, in fact, lower the maximum amount of relative humidity that is allowed to not make condensation happen. Water vapor condensation can led to failure causing corrosion, conductive anodic filament formation and/or electrochemical migration [61, 62]. Moreover, human body can't be exactly considered as a constant temperature environment, in fact, some temperature variations have to be actually taken into account. Obviously, if the humidity percentage is higher, the allowed temperature variation is smaller, concerning the condensation risk. Therefore, the intention is to calculate the maximum relative humidity to not have condensation considering temperature variations up to 4°C.

To convert between relative humidity, dew point and temperature there is a simple approximation which is actually quite accurate within about  $\pm 1^\circ\text{C}$  as long as the relative humidity is greater than 50% [63]. The approximating equations are the following [63]:

$$T_{dp} \sim T - \frac{100 - RH}{5}$$

$$RH \sim 100 - 5(T - T_{dp})$$

where  $T_{dp}$  is the dew point temperature,  $T$  the environment temperature and  $RH$  the corresponding relative humidity. These equations are often exploited in a simple rule of thumb which says that for every 1°C difference in dew point and air temperature, the relative humidity, from 100%, decreases by 5%.

Therefore, if the temperature difference has considered to be maximum 4°C, to not have condensation, the upper limit for relative humidity is 80%. Actually, this limit was achieved after 4065 minutes of test, which are equivalent to about 44 days of actual implantation time at 37°C. This is illustrated in figure 6.7.

Considering sensors tolerances, the time interval after that it is possible to reach 80% is  $4065 \pm 765$  minutes, which means  $44 \pm 7$  days at 37°C.

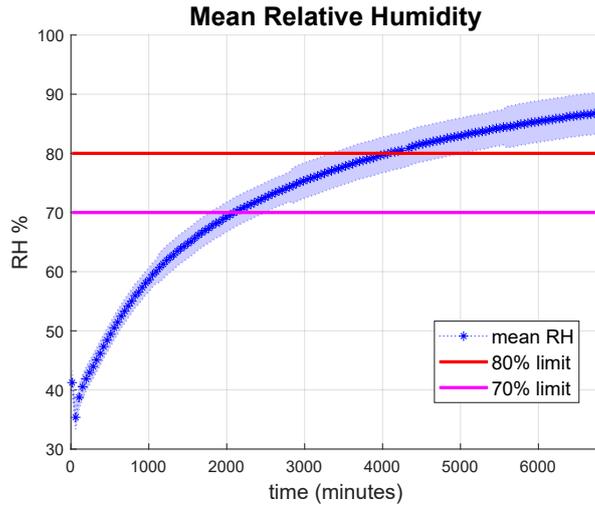


Figure 6.7: Mean relative humidity - limit at 80% and limit at 70%.

Indeed, this last evaluation is much more realistic and it confirms that the fabricated structure can't be exploited for too long-term implants, since the maximum implantation time period that is guaranteed is of about 37 days.

Moreover, regarding corrosion due to ionic contamination which actually acts as an aging factor for condensation, it has been proved that for contaminated printed circuit boards, condensation may happen at 70%, while for clean PCBAs, it could begin up to 98% [61, 64, 65]. Therefore, even the limit that has been set, considering dew point variations, could be too high for actual implants applications. Considering the safer RH upperlimit at 70%, figure 6.7 shows that it could be achieved just after about 2000 minutes of test, which are equivalent to 20 days at human body temperature (less than one month).

To conclude, experiment results state that this structure can't be exploited for long-term implant applications.

### 6.3.2 Comparison between Test 1 and Test 2

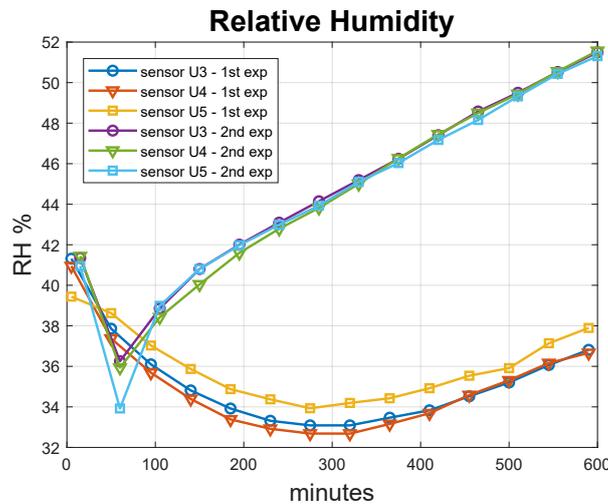


Figure 6.8: Comparison between Test 1 and Test 2.

To do a comparison between the two experiments, since the first one has lasted just for 590 minutes, the relative humidity evolution has been considered exclusively during this time interval. The

humidity measurements of the two tests are reported in the same graph in figure 6.8.

The experiment results show that the two tests started with almost the same relative humidity values. As already said, the temperature becomes constant at  $77^{\circ}\text{C}$  just after 150 minutes, thus it's reasonable to see an humidity decrease during this time interval due to temperature growth. For what concerns the second experiment, the moisture starts to enter inside the cavity almost immediately and the proof of this is given by the fact that humidity increases even when the temperature is increasing, as well. On the contrary, in the first experiment the package seems to be more hermetic since humidity goes up only after 300 minutes and, when it starts growing, the slope is slower than the one of experiment 2. This worsening in experiment 2 can be explained by blaming the sealing. In fact, very likely, the package for test number 2 hasn't been closed in a proper way. As a consequence, the bad sealing allows the moisture to penetrate more easily and this explains the fast increase just in the early stage of experiment number 2.

## 6.4 Test 3

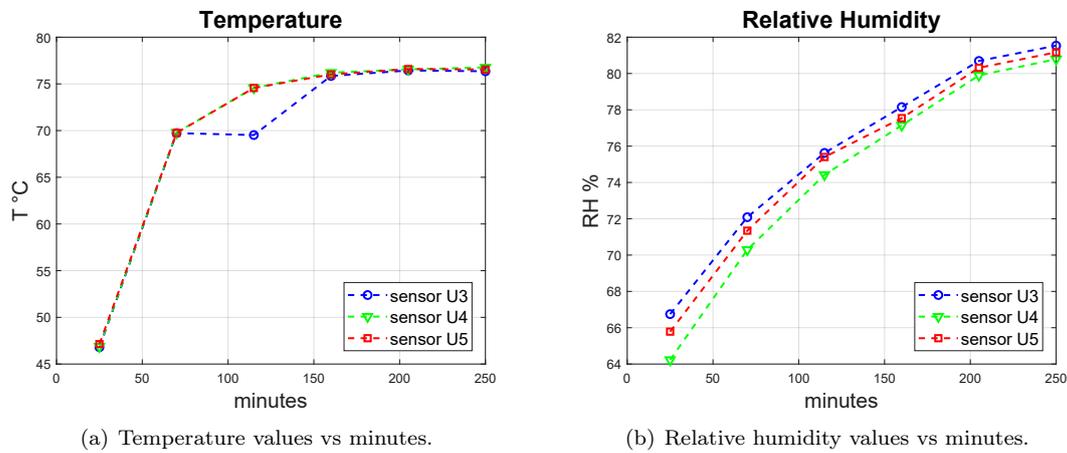


Figure 6.9: Experiment 3: elaborated values from the 3 SHTC3 sensors outputs.

The third test has been performed by adopting the same package structure employed in both the previous tests. Hence, after more than 5 days in PBS solution at  $77^{\circ}\text{C}$ , the surviving package system has been led to execute new measurements and the results are illustrated in figure 6.9.

As expected, temperature takes about 150 minutes to achieve the desired value and, differently from test 1 and 2, in that while, the relative humidity is strongly increasing. Even if the temperature goes up, the humidity always grows, as well. The growing slope is so fast that it reaches the 81% just after 250 minutes. Then, the experiment was stopped since it was so evident the unsuitability of the package. This demonstrates that the 3D printed structure, after 5 days of experiment at high temperature, can't be exploited again as a barrier against water vapor ingress. Moreover, this also proves that material permeability has changed over time. In fact, in this case, the bad sealing method is not the guilty for this fast worsening, but the actual responsible is the whole package substrate which has been deteriorated, losing completely impermeability.

## 6.5 Test 4

For test 4 it has been exploited a new package. The experiment has been terminated after 1260 minutes, which correspond to about 12 days at human body temperature. Therefore, the focus, in this case, was on analysing the humidity increasing in the early stage. The experiment outcomes can be observed in figure 6.10.

From there, as always, it is possible to observe that the system under test reached the humidity chamber temperature after about 150 minutes. From what concerns the relative humidity, it is

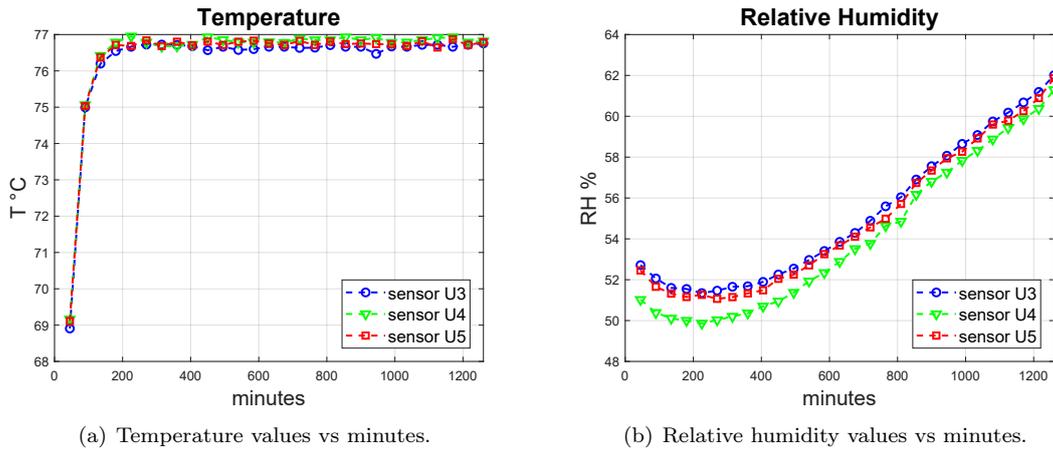


Figure 6.10: Experiment 3: elaborated values from the 3 SHTC3 sensors outputs.

remarkable that its starting value was quite high (52%). The reason why could be explained by the fact that, actually, before starting this test, the package was used before to try to do the same experiment twice. The previous two cases are not described in this project report because they terminated too early due to PCB problems not correlated with package hermeticity testing. Therefore, most likely, the package was just back from previous experiment phases which may cause a great starting humidity value. Anyway, the relative humidity starts to increase after 315 minutes (2 days at 37°C) but the growing is very slow, so that it is possible to consider it quite constant. The visible raise happens after 675 minutes, which are equivalent to almost 6 days of implantation at human body temperature. When the test was terminated, the relative humidity was almost 62%.

### 6.5.1 Comparison between Test 1 and Test 4

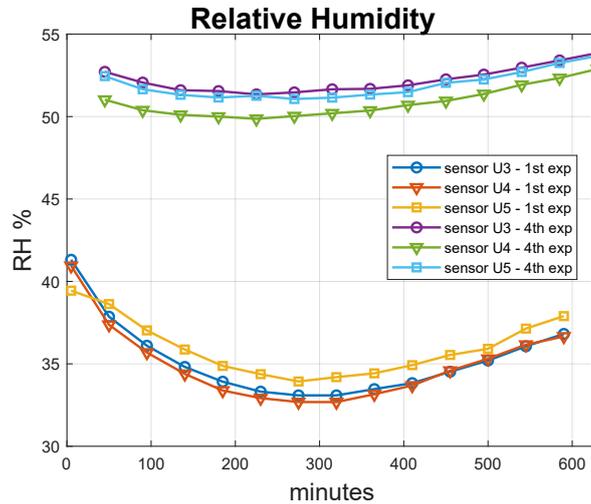


Figure 6.11: Comparison between Test 1 and Test 4.

As for the previous comparison, since the first experiment covers a 590 minutes time interval, only the relative humidity measures that happened during this beginning phase have been taken into account. The relative humidity evolutions for Test 4 and Test 1 are reported in the same plot in figure 6.11.

As expected, humidity in Test 4 starts at a higher level than the one of Test 1. Despite this, the evolutions are quite similar, in fact they both seem to maintain a constant level for all the time interval considered.

### 6.5.2 Comparison between Test 2 and Test 4

For this comparison, the time interval considered is the one of experiment number 4, since it is the shortest. The relative humidity results are shown in the same graph, for more clarity, in figure 6.12.

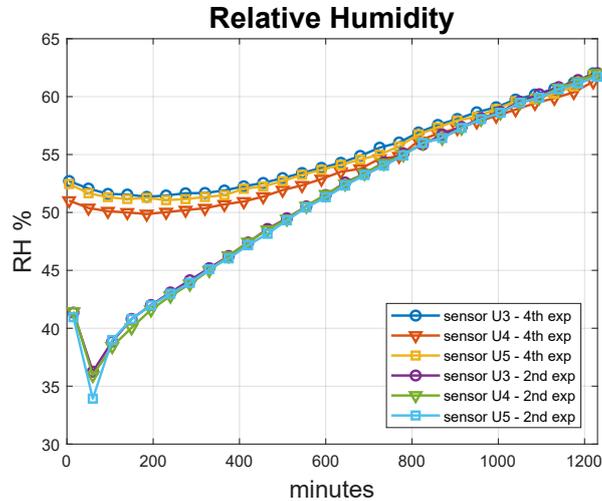


Figure 6.12: Comparison between Test 2 and Test 4.

The remarkable factor is that after about 725 minutes (6 days at 37°C), the two functions starts overlapping. In fact, relative humidity inside package of test number 4 starts at an high level and increases slowly, while relative humidity in test number 2 starts low but increases very fast, so that after 725 minutes reaches the same value of the last experiment, that is about 55%. In the subsequent minutes, the two relative humidity progressions seem to proceed together.

# Chapter 7

## Conclusion

As part of the project described in this thesis, a new type of polymeric package for implantable devices has been developed. This has been thought to be manufactured by means of 3D printing additive technology, based on Digital Light Processing building mechanism. The main reason of that choice is to lower the manufacturing cost, to carry out the developing process as much as possible easy and to make the overall package production affordable for every research groups. In terms of all essential requisites set for active implantable medical devices encapsulations, biocompatibility, biostability, RF-transparency are not big issues in facing with polymeric packages. On the contrary, hermeticity is an indispensable requirement which reveals to be a true challenge to take on. For that reason, accelerated aging tests have been carried on to test package suitability by measuring how much moisture is let to penetrate inside package cavity. To perform these measurements, an humidity monitoring system has been designed in a 29x29 mm printed circuit board to be placed inside package structure. The electronic system aim is to make relative humidity and temperature measures over time and to transmit them to the external by a Bluetooth wireless communication.

Four accelerated tests have been conducted and the results have been all analyzed and compared. For what concerns the estimated life time inside the human body organism, to allow a temperature variation of maximum 4°C, it turns that the maximum relative humidity allowed in the cavity, to not make condensation happen, is about 80%. This upperlimit was reached after 4065 minutes of test at 77°C, which are equivalent to about 37 days (considering sensors tolerances) at human body temperature. As a last remark, it has been highlighted that corrosion problems due to ionic contamination in printed circuit board assemblies may further lower the maximum relative humidity level. Therefore, considering this last safer limit, the implant lifetime is estimated to be about 20 days.

Hence, the final conclusions assert that the developed package structure is not suitable for a long term implantation. To make it acceptable for at least three months inside the human body organism, it actually needs improvements concerning both the substrate permeability and the sealing process.

### 7.1 Further work

Potential future works deal with all tasks which could improve package hermeticity against water, gasses and ions ingress. Obviously, no material is completely gas-tight, but it could be considered 'hermetic' a system whose leak rate is enough low related to the particular application in which it has to be employed. Generally, hermeticity depends on substrate material (its permeability), environment conditions and on the sealing technique. Therefore, a possible work, to enhance this characteristic, should properly handle with material permeability and should find a better sealing technique to make the device more appropriate for a medical implant lasting at least three months.

To improve permeability, an idea is to cover the whole structure by a Parylene-C film. Parylene is commonly exploited for medical devices and for PCB coatings. It shows good biocompatibility properties and it demonstrates to perform an exceptional moisture barrier [66]. Particularly,

Parylene-C is the one usually preferred because of its conformal films barrier properties, the low manufacturing cost and the easy deposition technique which is a simple chemical vapor deposition (CVD) that could be performed under vacuum at room temperature [66]. A Parylene CVD deposition could be a good improve in package permeability without increasing the manufacturing cost and the fabrication complexity. Hung-I Kuo et al. have proved in [67] that 25  $\mu\text{m}$ -thick Parylene-C conformal coating is able to provide a water barrier enough good for about 11 years.

Another improvement concerns the sealing technique. To ensure a good hermeticity for medical implantable devices, the best sealing methods are based on the so called direct bonding [35]. Direct bonding can be defined as a permanent joining technique implemented without exploiting adhesive or other products at the interface between the two parts to close. This bonding process can be also called welding, autoadhesion, fusion bonding or “glueless” bonding [35]. It can be applied to many thermoplastics and the three most common direct bonding techniques are thermal bonding, friction welding and transmission laser welding [35]. Always thinking about the complexity and the effective cost of these procedures, to stay in line with the main purpose of this project, the most easy technique that could be adopted to hermetically seal the package is thermal bonding.

Once that the fabricated package under test will be demonstrated to perform an excellent barrier against moisture for a long-term implant, the next step could be to design again the whole package structure adding feedthroughs. Feedthroughs are necessary to allow a reliable wire-communication between the implant and the external. They actually constitute a new challenge for package structure hermeticity, since they should be something placed in the middle between the external and the internal cavity. Therefore, accelerated aging tests should be once more performed and package suitability should be another time analysed and proved.

# Appendix A

## Electronic Components

### A.1 SHTW2

SHTW2, in figure A.1, is a relative humidity sensor produced by *Sensirion*.

As it can be noticed, it is built in a flip chip package. This technology makes it suitable for

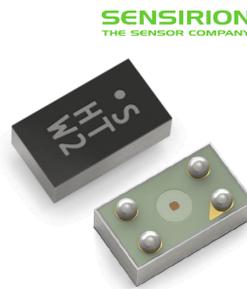


Figure A.1: SHTW2 Humidity and Temperature sensor from Sensirion [4]

applications with the tightest space constraints.

SHTW2 main features are:

- ultra-small 4-pin flip chip package:  $1.3 \times 0.7 \times 0.5 \text{ mm}^3$ ,
- ultra-low power consumption,
- 1.8 V Supply Voltage,
- I2C serial interface,
- typical measurements accuracy:  $\pm 3 \text{ \%RH}$  and  $\pm 0.4 \text{ }^\circ\text{C}$ .

The working principle consists of a capacitive humidity sensor and a bandgap temperature sensor. In more details, it's a digital system provided of A/D converter, analog and digital processing, a digital communication interface (I2C) and a calibration data memory.

Its block diagram description is shown in figure A.2.

The package pins are four, the minimum indispensable for any IC supporting I2C interface, and their functions are explained in table A.1.

Pin	Name	Comment
1	VDD	Supply Voltage
2	SCL	Serial clock
3	SDA	Serial Data
4	VSS	Ground

Table A.1: SHTW2 pin assignment [4]

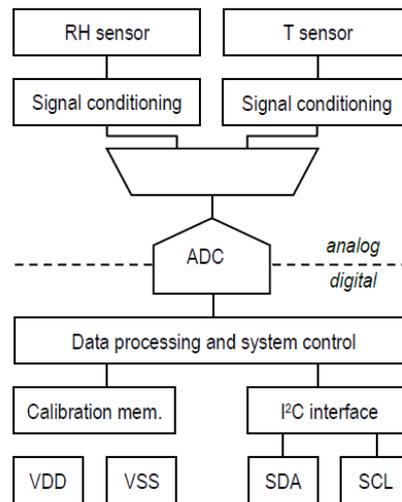


Figure A.2: Functional block diagram of SHTW2 [4]

## A.2 ISP1607-AX

ISP1507 is a MCU system (figure A.3), developed by *Insight*, that is an high performance multi-protocol module suitable for BLE / ANT+ / NFC with MCU and Antenna. In other words, it's an integrated circuit which includes, in the same system, microcontroller unit, bluetooth 5.0 BLE module and antenna.

It is built in a ultra-small LGA module, 8 x 8 x 1 mm and it is based on the nRF52 chip. This solution is the best for RF performance and low power consumption. Its main applications can be: to connect sensors for medical devices, IoT applications, wearable technology, home automation, ... and so on.

The ISP1607-AX version key features are summarized in the following list:

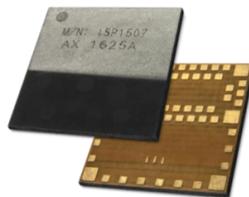


Figure A.3: ISP1507 [11]

- based on Nordic Semiconductor nRF52 (nRF52832 SoC),
- 32-bit ARM Cortex M4 CPU,
- single mode Bluetooth 5 stack,
- 512 kB Flash & 64 kB SRAM,
- 30 GPIOs (General Purpose Inputs/Outputs),
- ANT+ stack available,
- 2.4GHz low energy RF transceiver from nRF52 Nordic Semiconductor family,
- fully integrated RF matching and antenna,
- many interfaces SPI, UART, PDM, I2C,

- single 1.7 to 3.6 V supply,
- very small size 8.0 x 8.0 x 1.0 mm,
- temperature -40 to +85 °C.

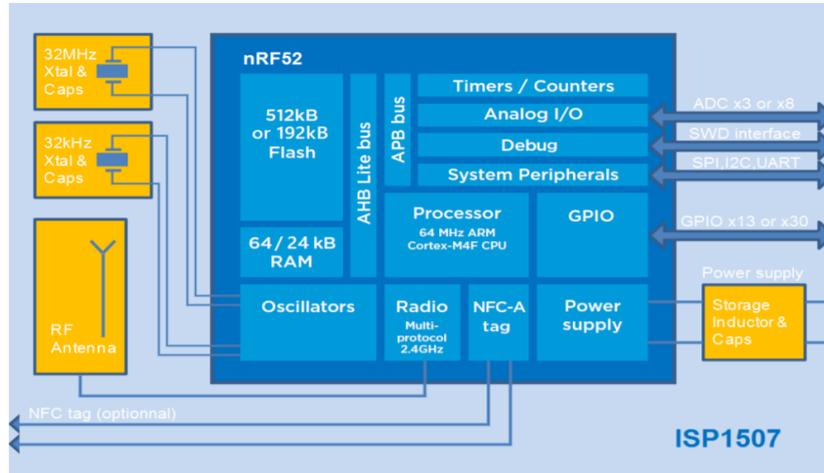


Figure A.4: ISP1507 Block diagram [11]

ISP1507 block diagram is depicted in figure A.4.

### A.3 nRF52832

nRF52832 Nordic Semiconductor System-On-Chip (SoC) is provided with an ARM Cortex M4 CPU at its hearth and supports Bluetooth 5.0 Low Energy. S112/s132 Softdevices implement fully qualified BLE stacks and they can be freely downloaded. In figure A.5 it has been reported a brief illustrative description.

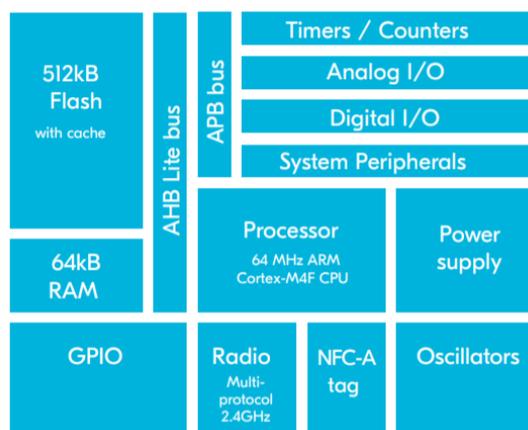


Figure A.5: nRF52832 module description [12].

Its key features are [12]:

- 2.4 GHz Transceiver,
- ARM Cortex-M4 32-bit processor with FPU, 64 MHz,
- Serial Wire Debug (SWD),

- Flexible power management: 1.7 V-3.6 V supply voltage range,
- 512 kB flash memory and 64 kB RAM.

## A.4 TPS82740A



Figure A.6: TPS82740A reference illustration

TPS82740A is a Step Down Converter Module (figure A.6), developed by *Texas Instruments*. It's a complete MicroSIP™ DC/DC step-down power module, particularly fitted for low power applications. This solution is based on switching regulator, integrated inductor and integrated input/output capacitors. It is characterized by a really small size (only  $6.7 \text{ mm}^2$ ). The basic schematic can be observed in figure A.7

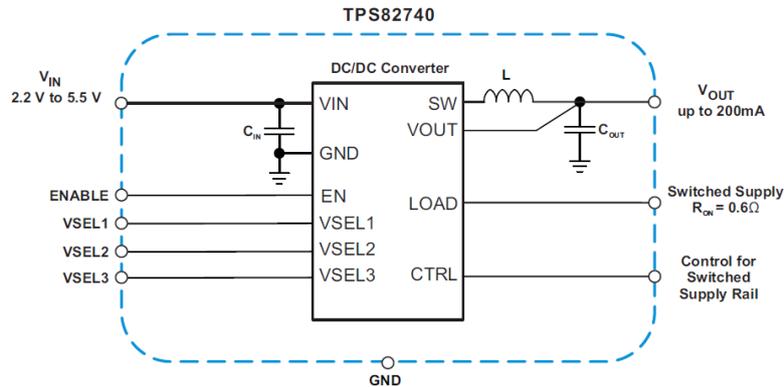


Figure A.7: TPS82740A Parameter Measurement Information [13]

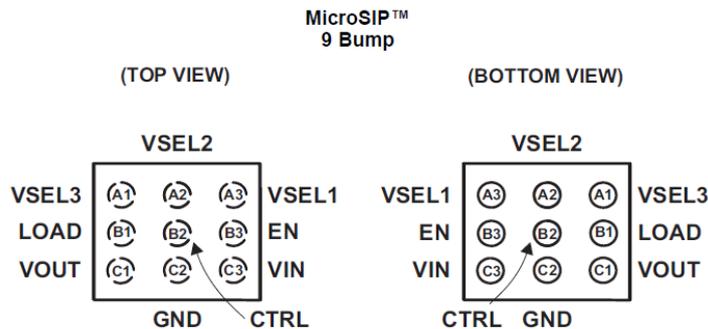


Figure A.8: TPS82740A Pin configuration - Top and Bottom views [13]

Its main features are the following:

- Pin-selectable output voltage in 100 mV steps,

- Up to 200 mA output current,
- Input voltage range  $V_{IN}$  from 2.2 V to 5.5 V,
- Low output voltage ripple,
- Total solution size  $< 6.7 \text{ mm}^2$ ,
- Small  $2.3 \text{ mm} \times 2.9 \text{ mm}$  MicroSIP™ package.

Package 9 bumps (figure A.8) are described in table A.2.

Name	NO	I/O
VIN	C3	IN
GND	C2	-
CTRL	B2	IN
VOUT	C1	OUT
LOAD	B1	OUT
EN	B3	IN
VSEL3	A1	IN
VSEL2	A2	IN
VSEL1	A3	IN

Table A.2: TPS82740A Pin names [13]

VOUT	VSEL3	VSEL2	VSEL1
1.8	0	0	0
1.9	0	0	1
2.0	0	1	0
2.1	0	1	1
2.2	1	0	0
2.3	1	0	1
2.4	1	1	0
2.5	1	1	1

Table A.3: Output Voltage Setting TPS82740A [13]. Output voltage can be selected by the user by means of three voltage select pins (VSEL), within a range from 1.8 V to 2.5 V.

The description about their functionalities, taken from the device datasheet, is the following:

- **VIN**: Input voltage supply pin.
- **GND**: Ground pin.
- **CTRL**: If  $CTRL = \text{low}$ , the **LOAD** output is pulled to ground.
- **VOUT**: Output voltage pin.
- **LOAD**: Load switch output pin controlled by the **CTRL** pin. With  $CTRL = \text{high}$ , **LOAD** pin is connected to the **VOUT** pin.
- **EN**: High level switches on the devices and low level disables the device.
- **VSELx**: Output voltage selection pins. See Table A.3 for **VOUT** selection.

## A.5 LEDs

The two LEDs are depicted in figure A.9, while in table A.4 their key features are itemized.

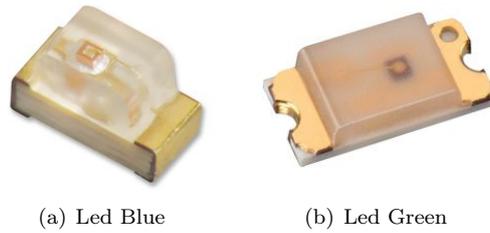


Figure A.9: LEDs representation

	Blue LED	Green LED
<b>Part number</b>	KP-1608QBC-D	150060GS55040
<b>Manufacturer</b>	KINGBRIGHT	WURTH ELEKTRONIK
<b>Voltage - Forward (Vf) (Typ)</b>	3.3 V	3.2 V
<b>Forward Current (If)</b>	20 mA	20 mA
<b>Mounting Type</b>	SMD	SMD
<b>Package / Case</b>	0603 (1608 Metric)	0603 (1608 Metric)

Table A.4: LEDs key features

## A.6 SHTC3

SHTC3 is a temperature and humidity sensor produced by *Sensirion* and it is depicted in figure A.10.



Figure A.10: SHTC3 Humidity and Temperature sensor from Sensirion [5]

As it can be noticed, SHTC3 sensor is built in a small DFN package which allows to involve it in applications in even the most limited of spaces [5].

The main features of SHTC3 are:

- Small DFN package:  $2 \times 2 \times 0.75 \text{ mm}^3$ ,
- ultra-low power consumption,
- Supply Voltage Range 1.62 V - 3.6 V,
- I2C serial interface,
- typical accuracy:  $\pm 2 \text{ \%RH}$  and  $\pm 0.2 \text{ }^\circ\text{C}$ .

The working principle consists of a capacitive humidity sensor and a bandgap temperature sensor. In more details, it's a digital system provided of a digital communication interface supporting (I2C), analog and digital processing, calibration data memory and a A/D converter. Its block diagram description is shown in figure A.11.

Pin functions can be read in table A.5.

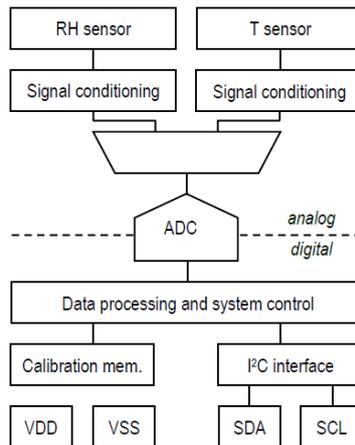


Figure A.11: Functional block diagram of SHTC3 [5]

Pin	Name	Comment
1	VDD	Supply Voltage
2	SCL	Serial clock
3	SDA	Serial Data
4	VSS	Ground

Table A.5: SHTC3 pin assignment [5]

## A.7 BL652-SA

BL652-SA (figure A.12), developed by *Laird*, is an high performance module suitable for Bluetooth Low Energy applications.

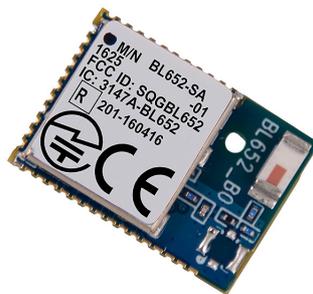


Figure A.12: Laird BL652-SA module

It is based on nRF52832 Nordic chip. This solution is the best for RF performance and low power consumption.

Its keys features are summarized in the following list:

- based on Nordic Semiconductor nRF52832 SoC,
- 32-bit ARM Cortex M4 CPU,
- single mode Bluetooth 5 stack,
- 512 kB Flash & 64 kB SRAM,
- 39 connection pads,
- many interfaces SPI, UART, PDM, I2C,

- single 1.8 to 3.6 V supply,
- small size 14 x 10 x 2.1 mm,
- pad pitch: 0.75 mm,
- temperature -40 to +85 °C.

BL652-SA block diagram is depicted in figure A.13.

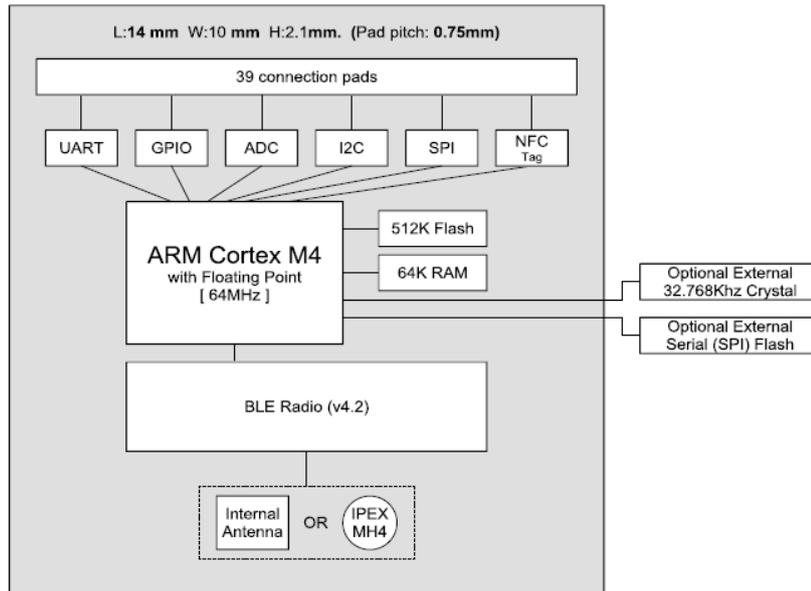


Figure A.13: BL652-SA Block diagram [14]

## A.8 LMZ21700

LMZ21700 is a Step Down DC/DC module, developed by *Texas Instruments*. It's a complete MicroSIPTM DC/DC step-down power module. The solution includes an integrated inductor and a switching regulator. A basic schematic can be seen in figure A.15.

Its main features are the following:

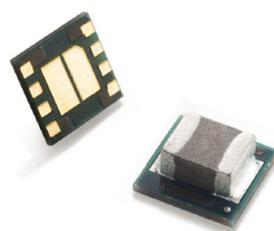


Figure A.14: LMZ21700 in SIL0008E Package [15]

- Miniature 3.5 mm × 3.5 mm × 1.75 mm Package,
- Adjustable Output Voltage,
- Integrated Inductor,

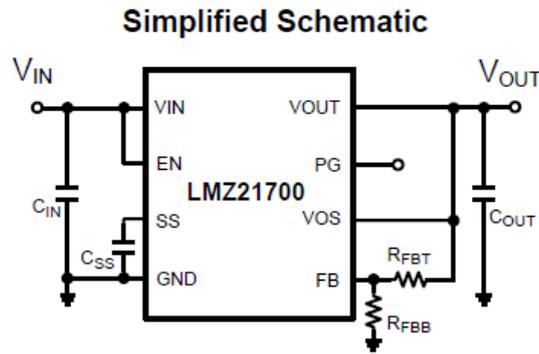


Figure A.15: LMZ21700 simplified schematic [15]

- Input Voltage Range 3 V to 17 V,
- Output Voltage Range 0.9 V to 6 V,
- Up to 650 mA Output Current.

The voltage selection occurs by means of two resistors ( $R_{FBT}$  and  $R_{FBB}$  in schematic of figure A.7) that have to be set for the voltage feedback.

Pin functions and names are itemized in table A.6.

Name	NO	I/O
VIN	8	IN
GND	5	-
SS	1	IN
VOUT	4	OUT
FB	2	IN
PG	3	OUT
VOS	6	IN
EN	7	IN
PAD	-	-

Table A.6: LMZ21700 Pin names [13]

The description about pins functionalities, taken from the device datasheet, is the following [15]:

- **VIN**: Input voltage supply.
- **GND**: Ground.
- **SS**: Soft-start pin.
- **VOUT**: Output Voltage.
- **FB**: Voltage feedback. Connect resistive voltage divider to this pin to set the output voltage.
- **PG**: Output power good (High = VOUT ready, Low = VOUT below nominal regulation).
- **VOS**: Output voltage sense pin and connection for the control loop circuitry.
- **EN**: Enable input (High = enabled, Low = disabled).
- **PAD**: GND.

## A.9 TXS0104E

TXS0104E, developed by *Texas Instruments*, is a 4-bit bidirectional voltage-level translator for open-drain and push-pull applications. It's built in a 14-pin SOIC package and its dimensions are 8.65 mm x 3.91 mm. If it should be connected to a breadboard, it's necessary an adapter (ARIES LCQT-SOIC14) which translates a 14-SOIC package in a 14-DIP one.

TXS0104E makes possible the communication between two logic domains powered at different voltage levels. The two ports are called port A and port B. It is recommended that the voltage level at port A is less or equal to the one at port B. Anyway, the description of the 14 pins and their functions is reported in table A.7.

Name	NO.	Type	Description
$V_{CCA}$	1	-	A-port supply voltage. $1.65 \leq V_{CCA} \leq 3.6$ V and $V_{CCA} \leq V_{CCB}$ .
A1	2	I/O	Input/Output A1. Referenced to $V_{CCA}$ .
A2	3	I/O	Input/Output A2. Referenced to $V_{CCA}$ .
A3	4	I/O	Input/Output A3. Referenced to $V_{CCA}$ .
A4	5	I/O	Input/Output A4. Referenced to $V_{CCA}$ .
NC	6	-	Not connected.
GND	7	-	Ground.
OE	8	I	3-state output-mode enabled. Referenced to $V_{CCA}$ .
NC	9	-	Not connected.
B4	10	I/O	Input/Output B4. Referenced to $V_{CCB}$ .
B3	11	I/O	Input/Output B3. Referenced to $V_{CCB}$ .
B2	12	I/O	Input/Output B2. Referenced to $V_{CCB}$ .
B1	13	I/O	Input/Output B1. Referenced to $V_{CCB}$ .
$V_{CCB}$	14	-	B-port supply voltage. $2.3$ V $\leq V_{CCB} \leq 5.5$ V.

Table A.7: TXS0104ED Pin Functions [17].

# Appendix B

## Schematics

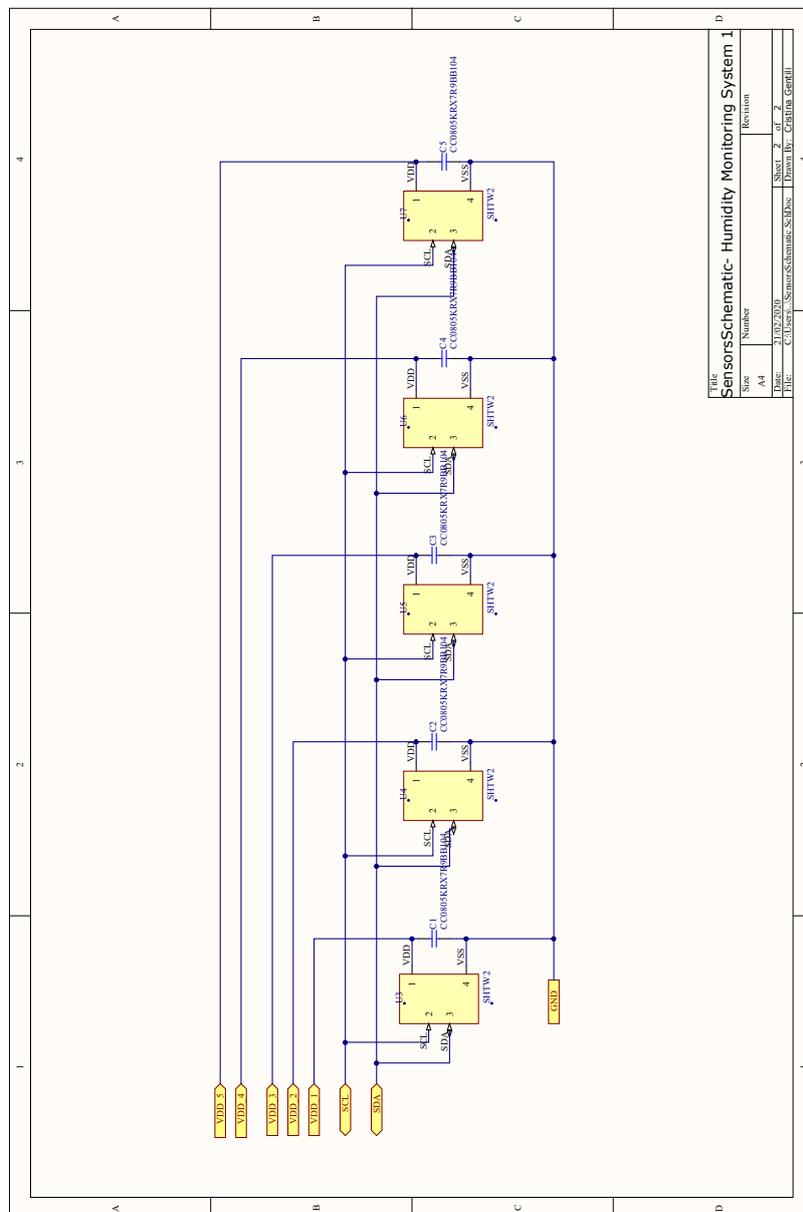


Figure B.1: Humidity Monitoring System 1 -I2C sensors communication Schematic

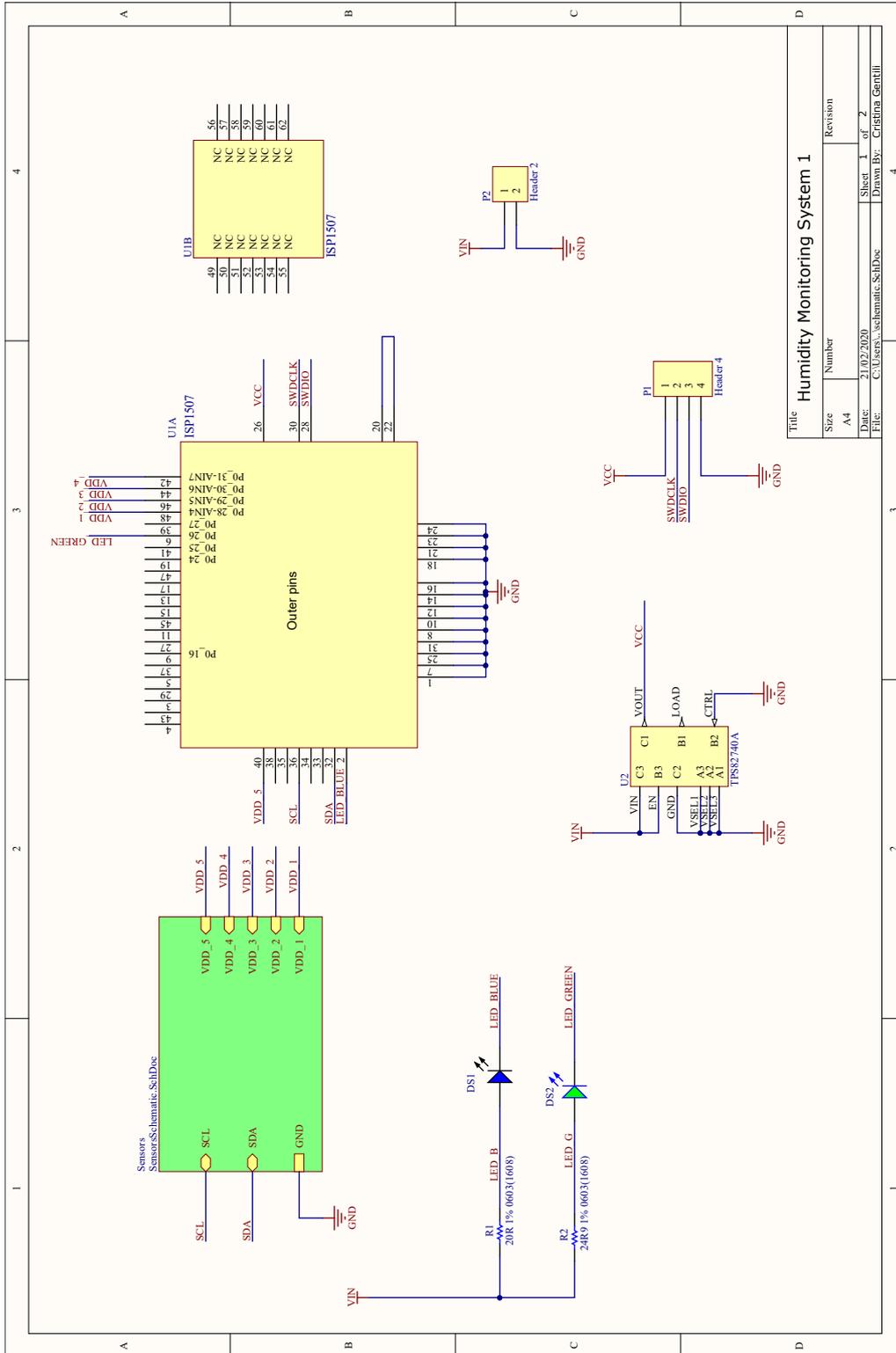
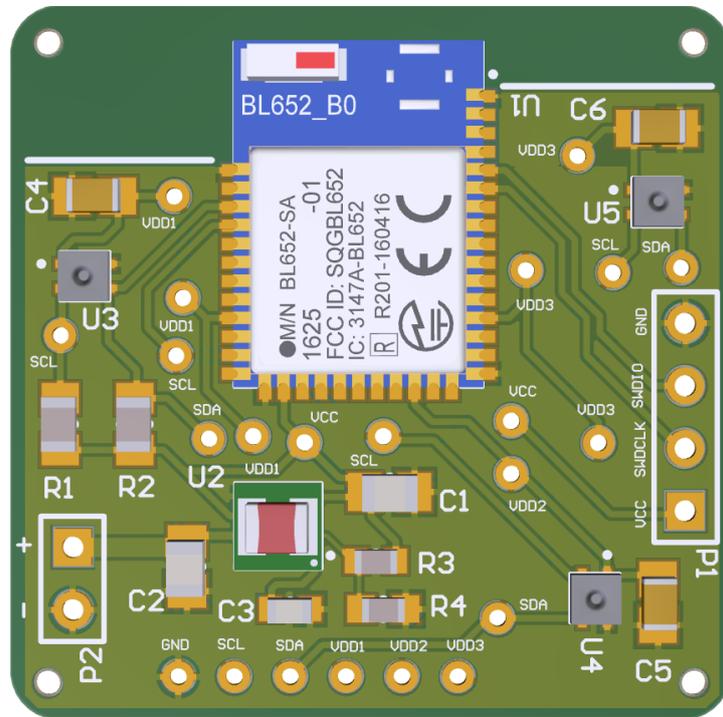


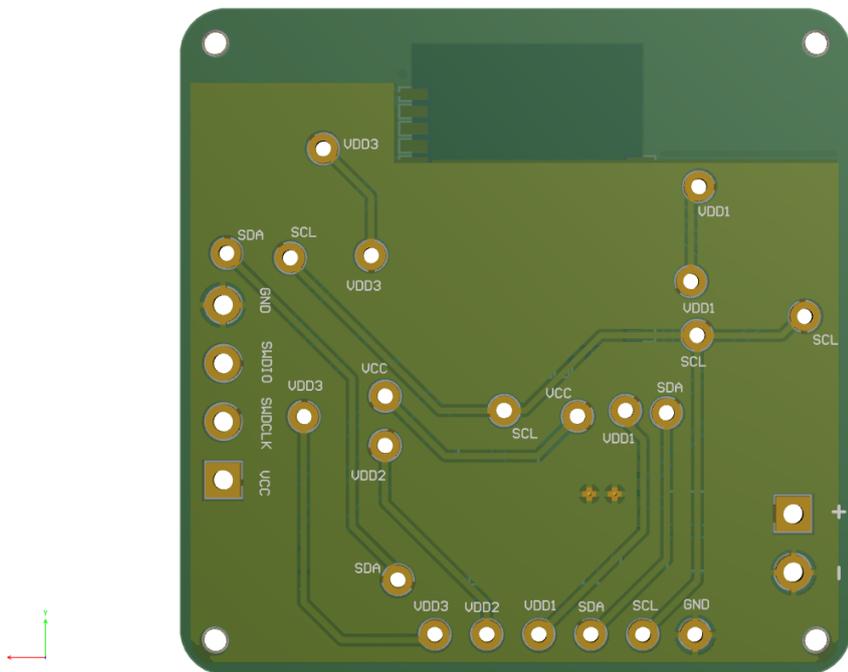
Figure B.2: Humidity Monitoring System 1 Complete Schematic







(a) Top view



(b) Bottom view

Figure C.2: Humidity Monitoring System 2 - PCB 3D Top and Bottom views from Altium Designer 19.1.8

# Bibliography

- [1] Maaïke Op de Beeck. Development of Implantable Electronics: issues and trends. BioCAS Tutorial, 2017.
- [2] Yeun-Ho Joung. Development of Implantable Medical Devices: From an Engineering Perspective. *It Neurovol J*, 17:98–106, 2013.
- [3] SENSIRION. *Introduction to Humidity - Basic Principles on Physics of Water Vapor*, 2009. [www.sensirion.com](http://www.sensirion.com).
- [4] SENSIRION. *Data Sheet SHTW2*, 2016. [www.sensirion.com](http://www.sensirion.com).
- [5] SENSIRION. *Data Sheet SHTC3*, 2019. [www.sensirion.com](http://www.sensirion.com).
- [6] Kristin Åstebøl - Nordic Semiconductor. Bluetooth low energy - a short introduction to bluetooth low energy and nordic semiconductor. <https://devzone.nordicsemi.com/nordic/short-range-guides/b/getting-started/posts/introduction-to-bluetooth-low-energy-and-ble-devel>, 2016.
- [7] Nordic Semiconductor. nrf5 sdk v15.2.0 documentation.
- [8] Hung Bui - Nordic DevZone. Bluetooth smart and the nordic's softdevices - part 1 gap advertising. <https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/bluetooth-smart-and-the-nordics-softdevices-part-1>, December 2015.
- [9] Hung Bui - Nordic DevZone. Bluetooth smart and the nordic's softdevices - part 2 connection. <https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/bluetooth-smart-and-the-nordics-softdevices-part-2>, February 2016.
- [10] Guk Bae Kim, Sangwook Lee, Haekang Kim, Dong Hyun Yang, Young Hak Kim, Yoon Soo Kyung, Choung Soo Kim, Se Hoon Choi, Bum Joon Kim, Hojin Ha, Sun U. Kwon, Namkug Kim. Three-Dimensional Printing: Basic Principles and Applications in Medicine and Radiology. *Korean Journal of Radiology*, 17(2):182–197, 2016.
- [11] Insight SiP. *ISP1507 High Performance Bluetooth 5 Ready, NFC & ANT Low Energy Module with MCU & Antenna*, 2019. [isp\\_ble\\_DS1507\\_R13.docx](http://isp_ble_DS1507_R13.docx).
- [12] Nordic Semiconductor. *nRF52832 Product Specification v1.4*, 2017.
- [13] Texas Instruments Incorporated. *TPS82740x 360-mA IQ MicroSIPTM Step Down Converter Module for Low Power Applications*, 2014. [www.ti.com](http://www.ti.com).
- [14] Laird. *Datasheet BL652-SA and BL652-SC*, 2017.
- [15] Texas Instrument. *LMZ21700 650-mA Nano Module With 17-V Maximum Input Voltage*, 2018.
- [16] 3D Systems, Inc. *Figure 4 MED-AMB 10*, 2019. Datasheet.
- [17] Texas Instruments. *TXS0104E 4-Bit Bidirectional Voltage-Level Translator for Open-Drain and Push-Pull Applications*, 2018. Datasheet.
- [18] Medical Device Developments. The future of implants: smart, electronic implantables. <https://www.medicaldevice-developments.com/features/featurethe-future-of-smart-electronic-implants-4787632/>, 2015.

- [19] A. Vanhoestenbergh. Implantable Electronic Devices Technology Challenges for Long-Term Human Implantation. *Sensor Review*, 29(4):345–348, 2009.
- [20] Inkyu Park Young-Jin Kim Adrien Teo, Abhinay Mishra.
- [21] Yao-Chuan Chang Umair Ahmed Stavros Zanos Timir Datta-Chaudhuri Jason Wright, Jason Wong. A low power implantable neurostimulator for small rodents with functional validation. *IEEE*, 2019.
- [22] Wikipedia. Implant (medicine). [https://en.wikipedia.org/wiki/Implant\\_\(medicine\)](https://en.wikipedia.org/wiki/Implant_(medicine)).
- [23] Djemel Lellouchi, J´er´emie Dhennin, Xavier Lafontan, David Veyrie, Jean-Fran,cois Le Neal, Francis Presseccq. Implantable devices: Issues and challenges. *Electronics*, 2:1–34, 2013.
- [24] Swarup Bhunia, Steve J.A. Majerus, Mohamad Sawan. *Implantable Biomedical Microsystems - Design Principles and Applications*. Elsevier Inc., The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB, UK, 225 Wyman Street, Waltham, MA 02451, USA, 2015.
- [25] Mauritis Ortmanns. Implantable Electronics for highly parallel neural interfaces. BioCAS tutorial, 2017.
- [26] Franck Chollet, Haobing Liu. *A (not so) short introduction to MEMS*. 2018.
- [27] Rony Jose James, Guido Spinola Durante, Mark Fretz, Ch. Bosshard, Stefan Mohrdiek. Optical and RF transparent long-term biocompatible micro-packages. CSEM, 2015.
- [28] David D. Zhou, Elias Greenbaum. *Implantable Neural Prostheses 2 - Techniques and Engineering Approaches*. Springer, Springer New York Dordrecht Heidelberg London, 2010.
- [29] Jorge Mario Herrera Morales, Claude Cl´ement. Technical Challenges of Active Implantable Medical Devices for Neurotechnology. *IEEE*, 2018.
- [30] Jorge Mario Herrera Morales. Evaluating biocompatible barrier films as encapsulants of medical micro devices. *Materials*, 2015. Universit´e Grenoble Alpes, 2015. English. NNT :2015GREAS016. tel-01289295.
- [31] Marc J. Madou. From MEMS to Bio-MEMS and Bio-NEMS: Manufacturing Techniques and Applications. In *Fundamentals of Microfabrication and Nanotechnology*, volume 3, page 213. CRC Press, 2011.
- [32] Wikipedia. Permeability (Earth sciences). [en.wikipedia.org/wiki/Permeability](https://en.wikipedia.org/wiki/Permeability).
- [33] Hossein Kaydani, Ali Mohebbi, Mehdi Eftekhari. Permeability estimation in heterogeneous oil reservoirs by multi-gene genetic programming algorithm. *Journal of Petroleum Science and Engineering*, 123:201–206, 2014.
- [34] Sebastian Winkler, Jan Edelmann, Christine Welsch, Roman Ruff. Different encapsulation strategies for implanted electronics. *DE GRUYTER*, 3(2):725–728, 2017.
- [35] Negin Amanat, Natalie L. James, David R. McKenzie. Welding methods for joining thermoplastic polymers for the hermetic enclosure of medical devices. *Elsevier Ltd Medical Engineering Physics*, 32:690–699, 2010.
- [36] Guangqiang Jiang, David D. Zhou. *Implantable Neural Prostheses 2*, chapter Technology Advances and Challenges in Hermetic Packaging for Implantable Medical Devices, pages 28–56. Springer Science+Business Media, 2010.
- [37] Heather Dunn. Hermetic packaging of implantable devices: How did we get here? And where are we going? Cirtec Medical, 2016.
- [38] Djemel Lellouchi, J´er´emie Dhennin, Xavier Lafontan, David Veyrie, Jean-Fran,cois Le Neal and Francis Presseccq. A new method for the hermeticity testing of waferlevel packaging. *Journal of micromechanics and microengineering*, 20(025031), 2010.

- [39] Federico Mazza, Yan Liu, Nick Donaldson, Timothy G. Constandinou. Integrated Devices for Micro-Package Integrity Monitoring in mm-Scale Neural Implants. *IEEE*, 18:978–1–5386–3603–9, 2018.
- [40] Suzanne Costello (née Millar), Marc P. Y. Desmulliez, Stewart McCracken. Review of test methods used for the measurement of hermeticity in packages containing small cavities. *IEEE Tran. on Components, Packaging and Manufacturing Technology*, 2(3):430–438, 2012.
- [41] F. W. Gole. *Introduction to Meteorology*. John Wiley & Sons Inc., New York, 1970.
- [42] Burak Okcan. *Humidity sensors using MEMS and standard CMOS technologies*. PhD thesis, The graduate school of natural and applied sciences of the middle east technical university, 2003.
- [43] All about Circuits. LED Resistor Calculator. <https://www.allaboutcircuits.com/tools/led-resistor-calculator/>, 2016.
- [44] Wikipedia. I2C. <https://it.wikipedia.org/wiki/I2C>.
- [45] SFUPTOWNMAKER. I2C. <https://learn.sparkfun.com/tutorials/i2c/all>. CC BY-SA 4.0.
- [46] Wikipedia. Bluetooth low energy. [https://it.wikipedia.org/wiki/Bluetooth\\_Low\\_Energy](https://it.wikipedia.org/wiki/Bluetooth_Low_Energy), December 2019.
- [47] Texas Instruments. Developing a bluetooth low energy application - generic access profile (gap), 2016.
- [48] Punch Through. How gap and gatt work. [dev.ti.com](http://dev.ti.com), June 2013.
- [49] MartinBL. Bluetooth low energy services, a beginner’s tutorial. <https://devzone.nordicsemi.com/nordic/short-range-guides/b/bluetooth-low-energy/posts/ble-services-a-beginners-tutorial>, August 2015.
- [50] MartinBL. Bluetooth low energy characteristics, a beginner’s tutorial. <https://devzone.nordicsemi.com/nordic/short-range-guides/b/bluetooth-low-energy/posts/ble-characteristics-a-beginners-tutorial>, March 2016.
- [51] Maurizio Menegotto, Marco Ferrario. *Debugging ARM Cortex<sup>TM</sup>-M con Trace*. Lauterbach, 2014.
- [52] Yosi Shacham. 2D & 3D printing for flexible nano and micro technologies. In *TAU NanoEl course: Fabrication methodologies for micro and nano systems 2018-2019*, 2018.
- [53] 3D Systems, Inc. *Figure 4 Standalone*, 2019. brochure.
- [54] Leo Greguric. *DLP 3D PRINTER OVERVIEW*, 2018. [all3dp.com/2/what-is-a-dlp-3d-printer-simply-explained/](http://all3dp.com/2/what-is-a-dlp-3d-printer-simply-explained/).
- [55] 3D Systems, Inc. *Figure 4 Standalone*, 2019. Technical Specifications.
- [56] Intertek Group PLC. Water Absorption ASTM D570. In *Testlopedia - The plastic Testing Encyclopedia*, 2015. [intertek.com/polymers/testlopedia/water-absorption-astm-d570/](http://intertek.com/polymers/testlopedia/water-absorption-astm-d570/).
- [57] ACE Seals. Epdm o-rings, gaskets & seals. <https://www.aceseal.com/ethylene-propylene-o-rings-gaskets-seals>, 2020.
- [58] Apple Rubber Products. Seal Design Guide.
- [59] Mohammad Kazemi, Eric Basham, Mohanasankar Sivaprakasam, Guoxing Wang, Damien Rodger, James Weiland1, Y.C. Tai, Wentai Liu, Mark Humayun. A Test Microchip for Evaluation of Hermetic Packaging Technology for Biomedical Prosthetic Implants. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society.*, 2004.
- [60] Wikipedia. Phosphate-buffered saline. [https://en.wikipedia.org/wiki/Phosphate-buffered\\_saline](https://en.wikipedia.org/wiki/Phosphate-buffered_saline).

- [61] David Budgett Simon Malpas Sarah Jane Guild Daniel McCormick Shun Long Cyril Au, Fu Yu Beverly Chen. Injection Molded Liquid Crystal Polymer Package for Chronic Active Implantable Devices with Application to an Optogenetic Stimulator. *IEEE Transactions on Biomedical Engineering*, 2019.
- [62] P. Viswanadham and P. Singh. Failure Modes and Mechanisms in Electronic Packages. *Chapman & Hall*, 53, 1998.
- [63] Wikipedia. Dew point. [https://en.wikipedia.org/wiki/Dew\\_point](https://en.wikipedia.org/wiki/Dew_point).
- [64] V. Verdingovas et al. Impact of NaCl contamination and climatic conditions on the reliability of Printed Circuit Board Assemblies. *IEEE Trans. Device Mater. Reliab.*, 14(1):1–10, 2013.
- [65] H. Conseil et al. *Experimental Study of Water Absorption of Electronic Components and Internal Local Temperature and Humidity into Electronic Enclosure*. 2014.
- [66] Wikipedia. Parylene. <https://en.wikipedia.org/wiki/Parylene>.
- [67] Hung-I Kuo, Rui Zhang, Wen H. Ko. Development of Micropackage Technology for Biomedical Implantable Microdevices using Parylene C as Water Vapor Barrier Coatings. *IEEE SENSORS 2010 Conference*, 2010.