

POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering



Master's Degree Thesis

Design and characterization of implantable drug delivery devices with remote pressure measurement

Supervisors

Prof. Giorgio De Pasquale

Dr. Alessandro Grattoni

Candidate

Marta Graziano

Academic Year 2021/2022

Abstract

Several drug delivery systems for the treatment of chronic diseases directly rely on or are affected by osmosis and osmotic pressure. Currently, these osmotic phenomena have only been experimentally investigated *in vitro*, limiting their applicability to translational purposes. The topic of the thesis is the design and characterization of implantable devices with embedded sensors for remote pressure measurement to achieve investigation of osmotic phenomena both *in vitro* and *in vivo*. The project is developed in collaboration with Dr. Grattoni Laboratory, Houston Methodist Research Institute, Texas and with the Department of Mechanical and Aerospace Engineering, Politecnico di Torino. The first prototypes consist of drug reservoirs with embedded strain sensors fabricated through metal and polymer additive manufacturing processes. The design is supported by structural simulations of the system using finite element analysis. The device consists in two symmetric discoidal membranes coupled with one peripheral collar; a linear strain gauge is installed at the center of the capsule flat circular surface. Experimental characterization of the reservoir in terms of strain-pressure relationship is achieved using a specifically designed test bench. The results obtained on titanium and polyamide devices demonstrate the applicability to passive drug delivery systems. To achieve direct pressure measurement, a second device prototype is developed based on the use of an integrated digital pressure sensor with I2C interface. Data acquisition and sensor testing are performed through Arduino and C++ coding. The device body features two chambers on the top part: the drug reservoir and the electronics chamber. The electronics chamber hosts the digital pressure sensor wired to the printed circuit board which contains the microcontroller, specifically programmed for continuous pressure data acquisition and remote transmission via Bluetooth Low Energy communication protocol. The bottom part of the body hosts an implantable discoidal battery to provide power supply to both the sensor and the communication system. A semipermeable membrane is embedded on top of the reservoir to allow for osmotic exchange. Extended *in vitro* experiments demonstrate the accuracy of pressure measurements and the stability of the Bluetooth wireless transmission. The prototype of the implantable device for direct pressure measurement proves the applicability to perform *in vitro* and *in vivo* investigation of osmotic pressure. The last part of the research focuses on *in vitro* experimentation performed loading the device reservoir with different solutions of glucose, sucrose, glycerol and glycine with different molar concentrations in order to measure the osmotic

pressure variation. The main outcome of the thesis is the design, development, characterization and experimental testing of two different prototypes of implantable devices to achieve indirect and direct remote pressure measurement inside a drug reservoir. This research will assist further investigation of the effect of osmotic pressure on drug delivery both *in vitro* and *in vivo*, setting the foundation for a new generation of more effective and reliable drug delivery systems.

Keywords: Implantable devices, Embedded sensors, Osmotic pressure, Drug delivery, Additive Manufacturing, Micro electro mechanical systems (MEMS), Bluetooth remote communication, I2C interface.

Acknowledgment

I would like to express my deep gratitude to Professor Giorgio De Pasquale who guided me throughout this project providing active support and precious advice. I also acknowledge Politecnico di Torino for facilities and resources dispensed during the initial part of the thesis. The main part of this research was accomplished at Houston Methodist Research Institute, I would like to thank Dr. Alessandro Grattoni for this great opportunity that allowed me to grow incredibly both professionally and personally. I would also like to offer my sincere gratitude to my mentor Dr. Nicola Di Trani for his constructive guidance, technical directions and for all his daily help during my research activity. I wish to express my special thanks to all the people of the Grattoni Lab for the support they provided me during these months away from home. Finally, I would like to express my deepest and heartfelt thanks to my family and to all my friends for having always been by my side throughout this five-years journey.

List of Contents

Abstract	2
Acknowledgment	4
List of Contents	5
List of Figures	7
List of Tables	9
1 Introduction	10
1.1 Problem definition.....	10
1.2 Objectives.....	10
1.3 Scope and contribution of the work	11
1.4 Thesis organization	11
2 Design and fabrication of drug reservoirs with embedded strain sensors	14
2.1 Structural analysis of discoidal pressure vessel	14
2.2 FEM structural simulations	16
2.2.1 Capsule geometry and simulation methodology	16
2.2.2 Simulation results for titanium alloy and polyamide	18
2.3 Capsule design and fabrication	22
2.3.1 Capsule design using SolidWorks.....	22
2.3.2 Metal and polymer additive manufacturing processes.....	23
2.3.3 Capsule assembly and sealing.....	25
2.4 Strain gauge sensing system.....	25
2.4.1 Strain gauge theory	26
2.4.2 Strategy of experimental investigation.....	26
2.4.2.1 Resistance-strain relationship.....	26
2.4.2.2 Strain-pressure relationship.....	27
2.4.3 Strain gauge installation procedure.....	29
3 Experimental characterization and validation of drug reservoirs with strain sensors. 30	
3.1 Test bench design and fabrication.....	30
3.1.1 Design principles.....	30
3.1.2 Components and fabrication	33
3.2 Measurement chain and data acquisition	35
3.2.1 Data acquisition system	35
3.2.2 Validation of the acquisition system.....	37

3.3	Setup and testing methodology	41
3.4	Results and discussion.....	43
3.4.1	Titanium alloy prototype.....	44
3.4.2	Polyamide device	45
4	Design and fabrication of implantable device with digital pressure sensor	49
4.1	Integrated digital pressure sensors	50
4.1.1	I2C communication protocol	52
4.1.2	Code development and testing	55
4.1.2.1	HSPPAD143A sensor: Arduino coding	56
4.1.2.2	HSPPAD143A sensor: C++ coding	60
4.1.2.3	MS5837-30BA sensor: C++ coding.....	61
4.1.3	Results and discussion	66
4.2	Bluetooth remote communication	67
4.3	Design and assembly procedure.....	69
4.3.1	Design and fabrication	69
4.3.2	Components and assembly procedure.....	70
5	Experimental validation of implantable device with digital pressure sensor	74
5.1	Test bench and testing methodology.....	74
5.2	Results and discussion.....	76
6	In vitro experimentation	78
6.1	Reverse osmosis membrane testing.....	78
6.2	Stability of the remote communication	80
6.3	<i>In vitro</i> methodology.....	81
6.4	Results and discussion.....	82
	Conclusions and Future Directions	86
	References	88
	Appendices.....	90
	Arduino code for HSPPAD143A sensor.....	90
	C++ code for HSPPAD143A sensor.....	91
	C++ code for MS5837-30BA sensor	94
	Matlab code for data acquisition from pressure transducer	98
	Matlab code for data acquisition from the device.....	100

List of Figures

Figure 1_Circular flat plate with uniform pressure and fixed boundary	15
Figure 2_Discoidal capsule (Ansys Design Modeler)	17
Figure 3_Titanium alloy FEM simulation results.....	19
Figure 4_Comparison of simulation results for stress, strain and deformation	21
Figure 5_Capsule drawings (SolidWorks).....	23
Figure 6_Ti6Al14V and PA12 capsules	24
Figure 7_Assembly and sealing of titanium and polyamide capsules	25
Figure 8_Electrical linear strain gauge schematic	26
Figure 9_Strain distribution in circular plate with fixed boundaries and uniform pressure [16] .	28
Figure 10_Installation of the linear electric strain gauge.....	29
Figure 11_Pneumatic cylinder schematic	31
Figure 12_Pneumatic cylinder for test bench	32
Figure 13_Pressure adjustment system in the test bench.....	32
Figure 14_Test bench schematic.....	33
Figure 15_Custom-made fitting.....	34
Figure 16_Test bench components	35
Figure 17_QuantumX MX1615B	36
Figure 18_Catman settings for the strain gauge	37
Figure 19_Cantilever beam subject to a force W	38
Figure 20_Cantilever beam in quarter bridge configuration.....	39
Figure 21_Quarter bridge configuration for QuantumX.....	40
Figure 22_Comparison between theoretical and experimental strain for the cantilever beam.....	41
Figure 23_Experimental setup	42
Figure 24_Strain data acquisition with Catman.....	43
Figure 25_Theoretical and experimental strain-pressure characteristics for titanium alloy.....	45
Figure 26_Interpolation of experimental data for polyamide	47
Figure 27_Theoretical and experimental strain-pressure characteristics for polyamide	47
Figure 28_HSPPAD143A sensor.....	51

Figure 29_MS5837-30BA sensor	52
Figure 30_I2C master-slave configuration	53
Figure 31_LAUNCXL board.....	54
Figure 32_HSPPAD143A register map	56
Figure 33_HSPPAD143A algorithm	57
Figure 34_HSPPAD143A I2C circuit.....	58
Figure 35_HSPPAD143A pin layout.....	59
Figure 36_Schematic of hardware setup for HSPPAD143A sensor.....	59
Figure 37_Hardware setup for HSPPAD143A testing	60
Figure 38_Hardware setup for testing with LAUNCHXL board	61
Figure 39_MS5837-30BA algorithm.....	64
Figure 40_MS5837-30BA circuit	64
Figure 41_MS5837-30BA pin layout	65
Figure 42_Schematic of hardware setup for MS5837-30BA sensor	65
Figure 43_Output during sensor testing.....	66
Figure 44_Scan Response Data structure	67
Figure 45_Example of data acquisition through BLE	68
Figure 46_Device drawings (SolidWorks)	70
Figure 47_Device assembly (SolidWorks rendering).....	71
Figure 48_Device assembly procedure	73
Figure 49_Reference pressure transducer	75
Figure 50_Experimental setup	75
Figure 51_Experimental results for pressurized air test	76
Figure 52_Experimental results for pressurized water test.....	77
Figure 53_Experimental apparatus	79
Figure 54_RO membrane testing results.....	80
Figure 55_Stability testing results	81
Figure 56_In vitro experimentation setup and methodology	82
Figure 57_In vitro results for glucose, glycerol, sucrose and glycine solutions.....	84
Figure 58_Comparison of in vitro results	84
Figure 59_In vitro results with glycine solution.....	85

List of Tables

Table 1_ FEM simulation results for Titanium alloy and Polyamide.....	18
Table 2_ Structural parameter-pressure characteristic for titanium alloy and polyamide	21
Table 3_ Electrical strain gauge specifications.....	29
Table 4_ Test bench components.....	35
Table 5_ Titanium alloy capsule specifications.....	44
Table 6_ Titanium alloy capsule theoretical and experimental results.....	44
Table 7_ Polyamide capsule specifications	45
Table 8_ Polyamide capsule theoretical and experimental results	46
Table 9_ Experimental results for titanium and polyamide.....	48
Table 10_ HSPPAD143A slave address	56
Table 11_ MS5837-30BA slave address.....	61
Table 12_ Device dimensions	69

1 Introduction

1.1 Problem definition

The increasing prevalence of chronic diseases drives the development of novel technologies to offer personalized care to patients and relieving the economic burden on healthcare providers [1]. Chronic disorders require long-term therapeutic strategies, and the use of implantable drug delivery devices has the potential of increasing medication efficacy, often hindered by patient poor adherence, and reducing treatment-monitoring clinical visits [2]. Several drug delivery systems either directly rely on or are affected by osmosis and osmotic pressure [3]. Osmosis is a naturally occurring process that involves a net exchange of solvent from a low to a high concentrated solution across a semi-permeable membrane. Among the implantable drug delivery strategies under study in the last years, osmotic pumps and reservoir-based devices are the most promising. Osmotic pumps enclose a drug core in semi-permeable membrane that once implanted absorbs water through osmosis. The resulting increase in pressure leads to drug expulsion through laser-driller miniature holes [4]. Implantable reservoirs instead, base their drug release on concentration driven diffusion through opportunely designed semi-permeable membranes [5, 6]. However, osmotic pressure build-up in these reservoirs can also affect the drug release and may be sufficient to cause the failure of implants with associated risks to patients [7]. Currently, these osmotic phenomena have only been experimentally investigated *in vitro*, limiting their applicability to translational purposes. The aim of this study is the development and characterization of implantable devices equipped with embedded sensors to perform remote continuous monitoring of osmotic pressure both *in vitro* and *in vivo*.

1.2 Objectives

The aim of this study is to design, fabricate, characterize and validate implantable drug delivery devices able to perform remote pressure measurement inside a drug reservoir. The main objectives are the following: design a sensing system able to achieve continuous pressure data acquisition with a certain accuracy and stability over the time; develop a strategy to transmit data acquired in a reliable and wireless way in order to make the device suitable for *in vivo* experimentation;

provide power supply to the electronic components; integrate the fluidic semipermeable membrane in the design of the device; identify the proper biocompatible materials and manufacturing process for device fabrication; achieve small dimensions and compact shape to allow for implantation in small animals such as mice and rats. Another main goal, after the design and fabrication of the device, consists in performing experimental testing to prove the accuracy of pressure measurement and the reliability of the data acquisition system and to achieve characterization and validation of the implantable devices. The final goal of this research is to perform a set of *in vitro* experiments to acquire pressure data inside the reservoir filled with different highly concentrated solutions in order to achieve a better understanding of the effect of osmotic phenomena on drug delivery.

1.3 Scope and contribution of the work

This study will provide prototypes of two implantable drug delivery devices able to perform respectively indirect and direct remote pressure measurement thanks to custom made sensing systems based on embedded sensors. Furthermore, extended experimental testing will lead to characterization and validation of the measurement strategy, of the accuracy of pressure values and stability of the data acquisition system. Additionally, *in vitro* experimentation will be conducted using the prototypes designed and fabricated in order to acquire information about osmotic pressure behavior inside the drug reservoir. The result of this thesis will also assist further investigation of the effect of osmotic pressure on drug delivery both *in vitro* and *in vivo*. The outcome of this research will lead to a better understanding of osmotic phenomena, setting the foundation for a new generation of more effective and reliable drug delivery systems for chronic disease treatment.

1.4 Thesis organization

After the introductory chapter where objectives and purposes of the thesis are explained, *Chapter 2* describes the design and fabrication of drug reservoirs with embedded strain sensors for and continuous pressure measurement. This research activity is conducted in the Department of Mechanical and Aerospace Engineering at Politecnico di Torino. Firstly, a theoretical overview on the structural analysis of pressure vessel is reported, while the following part explains the design methodology based on structural simulations using finite element analysis. The next section of the

chapter focuses on the fabrication of two prototypes made of metallic and polymeric additive manufacturing materials. After an introduction on strain gauge theory, the overall device assembly procedure is described in detail together with the measurement chain for data acquisition from the embedded strain sensor.

Chapter 3 illustrates the methodology and results of *in vitro* experimentation performed with the two prototypes of drug reservoir with embedded strain sensor. The first subchapter illustrates the test bench conceiving and fabrication, followed by the development and description of the acquisition system. The last section explains the experimental results regarding the device characterization in terms of strain-pressure relationship, together with the evaluation of pressure measurement accuracy through comparison with theoretical values.

In *Chapter 4* is described the research conducted during a six-months internship in Dr. Grattoni Laboratory at Houston Methodist Research Institute, Texas. The purpose of the research is the development of a second device able to perform direct pressure measurements using a digital integrated pressure sensor. This second solution is more expensive but allows simple integration of the sensor with the printed circuit board already developed in Dr. Grattoni Laboratory. The first part of the chapter describes the analysis and testing of two integrated pressure sensors, followed by the description of the C++ code to interface the sensor with the microcontroller of the PCB and to achieve remote data communication via Bluetooth transmission. The last subchapter illustrates the design and fabrication of the device using a CAD software and 3D printing technique, together with the description of the device components and assemble.

Chapter 5 focuses on the testing methodology for the experimental validation of the implantable device with integrated pressure sensor and begins illustrating the test bench setup and the testing methodology. Then, the second subchapter illustrates the results obtained with discussion and conclusions.

Chapter 6 presents the *in vitro* experimentation performed with the remotely controlled implantable device. The first section describes the membrane testing procedure performed to achieve membrane fluidic characterization and to acquire preliminary information about osmotic pressure behaviour. The next subchapter focuses on *in vitro* experiments aiming at proving the accuracy of pressure measurements and the stability of the Bluetooth remote communication. After an overview of the methodology and the experimental design used for conducting *in vitro* experimentation, the last section reports the pressure results obtained reloading the device with

different solutions of glucose, sucrose, glycine and glycerol at different molar concentrations.

Section 7 summarizes the achievements and outcomes of the research project and presents the future directions. Finally, *Sections 8* and *9* report respectively the list of references and the scripts of the main Arduino, C++ and Matlab codes developed in the different stages of the work.

2 Design and fabrication of drug reservoirs with embedded strain sensors

This chapter describes the design and fabrication of drug reservoirs with embedded strain sensors in order to perform continuous pressure measurement. The first subchapter describes the theory applied in the structural analysis of pressure vessels, the aim is to give a general overview of stresses and deformations involved when a vessel is subjected to an internal pressure. The following part explains the design methodology based on structural simulations using finite element analysis; the modelling of the capsule is accomplished using a computer-aided design (CAD) software. The next subchapter focuses on the device fabrication: two prototypes are produced through additive manufacturing processes using metal and polymeric biocompatible materials. Further, an overview of the functioning of a strain gauge is reported, followed by the procedure for strain gauge installation on the capsule. The last part describes the device assemble and the measurement chain for data acquisition from the embedded stain sensor.

2.1 Structural analysis of discoidal pressure vessel

In the discoidal pressure vessel, the most stressed surfaces are the two circular surfaces while the lateral surface is subjected to minor stresses and therefore can be neglected. The two circular surfaces can be analyzed as flat plates with uniform thickness and fixed boundaries under a uniform pressure. The formulas of this section can be considered sufficiently accurate so long as the following assumptions hold true [8, 9]:

- the plate is flat, with uniform thickness and homogeneous isotropic material;
- the thickness is not more that about one-quarter of the least transverse dimension, and the maximum deflection is not more than about one-half the thickness;
- all forces, loads and reactions, are normal to the plane of the plate;
- the plate is nowhere stressed beyond the elastic limit.

Radial and circumferential stresses respectively are obtained with the following formulas:

$$\sigma_r = \frac{12m_r}{h^3}z \quad (1)$$

$$\sigma_{\theta} = \frac{12m_{\theta}}{h^3} z \quad (2)$$

where

m_r = bending moment per unit of length acting in the rz plane

m_{θ} = bending moment per unit of length acting in a plane normal to rz plane

z = direction normal to plane of the plate

h = thickness of the plate

For a circular plate with uniform pressure and fixed boundary the radial and circumferential bending moments are respectively defined as follows:

$$m_r = \frac{q}{16} [a^2(1 + \nu) - r^2(3 + \nu)] \quad (3)$$

$$m_{\theta} = \frac{q}{16} [a^2(1 + \nu) - r^2(1 + 3\nu)] \quad (4)$$

where q is the uniform pressure applied on the flat plate, a is the plate radius and ν is the Poisson's ratio as described in the following figure.

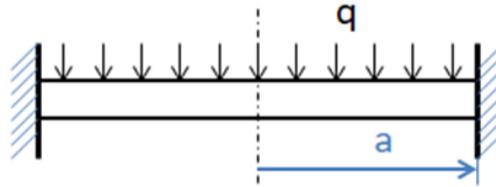


Figure 1_Circular flat plate with uniform pressure and fixed boundary

Radial and circumferential strains respectively can be obtained with the following formulas, where E is the young's modulus:

$$\varepsilon_r = \frac{\sigma_r}{E} \quad (5)$$

$$\varepsilon_{\theta} = \frac{\sigma_{\theta}}{E} \quad (6)$$

The most stressed point corresponds to the centre of the plate, in this point bending moments are equal therefore also stresses and strains are equal, and the following formulas can be applied:

$$m_r = m_{\theta} = \frac{qa^2}{16} (1 + \nu) \quad (7)$$

$$\sigma_{max} = \frac{3qa^2}{4h^2} \quad (8)$$

The vertical displacement w at the centre of the plate is defined as follows:

$$w_{max} = \frac{qa^4}{64D} \quad (9)$$

$$D = \frac{Eh^3}{12(1 - \nu^2)} \quad (10)$$

The flat plate theory provides a sufficiently accurate model that can be applied for the study and analysis of stresses, strains and deformations to which a discoidal pressure vessel is subjected. The formulas reported above are used in the following design phase and in particular to determine the most loaded point and the expected strain as function of internal pressure. This information is essential in the choice of the strain gauge sensor and in the identification of the most suitable location site for strain gauge application on the device capsule.

2.2 FEM structural simulations

In the first phase of the design process structural simulations of a discoidal pressurized vessel are performed to study stresses, strains and deformations developed on the circular surfaces as function of different internal pressures applied. Simulations focuses on two additive manufacturing materials: the first material is a titanium alloy and the other one is a polymeric material called polyamide. The simulation software used is Ansys, which allows to perform structural analysis based on Finite Element Method (FEM).

2.2.1 Capsule geometry and simulation methodology

The first step is to define the geometry of the model realizing a 3D model of the capsule using the Design Modeler tool in Ansys. Geometry specifications are the following:

- external radius: 12,5 mm
- internal radius: 11,5 mm
- height: 6 mm
- wall thickness: 1 mm

The figures below show the capsule model realized in ANSYS with indication of dimensions.

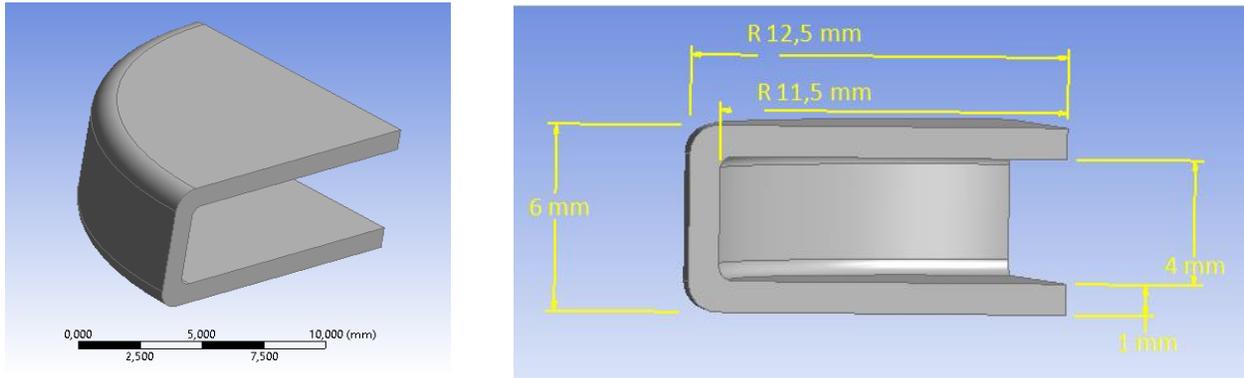


Figure 2_Discoidal capsule (Ansys Design Modeler)

The first material under study is the titanium alloy Ti6Al4V; this material is predefined in Ansys with the following specifications:

- Density: $4.62 \times 10^{-6} \text{ kg/mm}^3$
- Young's modulus: $9.60 \times 10^4 \text{ MPa}$
- Poisson's ratio: 0.36
- Shear modulus: $3.53 \times 10^4 \text{ MPa}$
- Tensile ultimate strength: $1.07 \times 10^3 \text{ MPa}$
- Tensile Yield strength: $0.93 \times 10^3 \text{ MPa}$

The second material under study is a polymeric material called Polyamide PA6 which presents the following specifications:

- Density: $1.14 \times 10^{-6} \text{ kg/mm}^3$
- Young's modulus: $1.11 \times 10^3 \text{ MPa}$
- Poisson's ratio: 0.35
- Shear modulus: $4.11 \times 10^2 \text{ MPa}$
- Tensile ultimate strength: $7.19 \times 10 \text{ MPa}$
- Tensile Yield strength: $4.31 \times 10 \text{ MPa}$

Simulations are conducted using, for simplicity, a quarter of the complete discoidal capsule model and are focused on the analysis of the following structural parameters:

- Equivalent stress (Von Mises) expressed in MPa
- Equivalent elastic strain (Von Mises) expressed in mm/mm

- Total deformation expressed in mm

The following subchapter describes the results obtained from Ansys structural simulations with the two additive manufacturing materials.

2.2.2 Simulation results for titanium alloy and polyamide

After definition and assignment of material, structural simulations with different values of internal pressure have been performed. In the following table are reported, as example, the results obtained with the pressure value:

$$p = 8 \text{ bar} = 0.81 \text{ MPa}$$

<i>Titanium alloy</i>		
Structural parameter	Minimum	Maximum
Equivalent Stress, MPa	3.44	7.40 x 10
Equivalent Elastic Strain, mm/mm	2.66 x 10 ⁻⁴	7.70 x 10 ⁻⁴
Total Deformation, mm	4.49 x 10 ⁻⁴	3.92 x 10 ⁻²
<i>Polyamide</i>		
Structural parameter	Minimum	Maximum
Equivalent Stress, MPa	3.36	7.25 x 10
Equivalent Elastic Strain, mm/mm	2.28 x 10 ⁻²	6.53 x 10 ⁻²
Total Deformation, mm	3.82 x 10 ⁻²	3.38

Table 1_ FEM simulation results for Titanium alloy and Polyamide

The following figures show, as an example, the Ansys output results for equivalent stress, equivalent elastic strain and total deformation respectively after performing the FEM structural analysis with an applied internal pressure of 8 bar with Titanium alloy material.

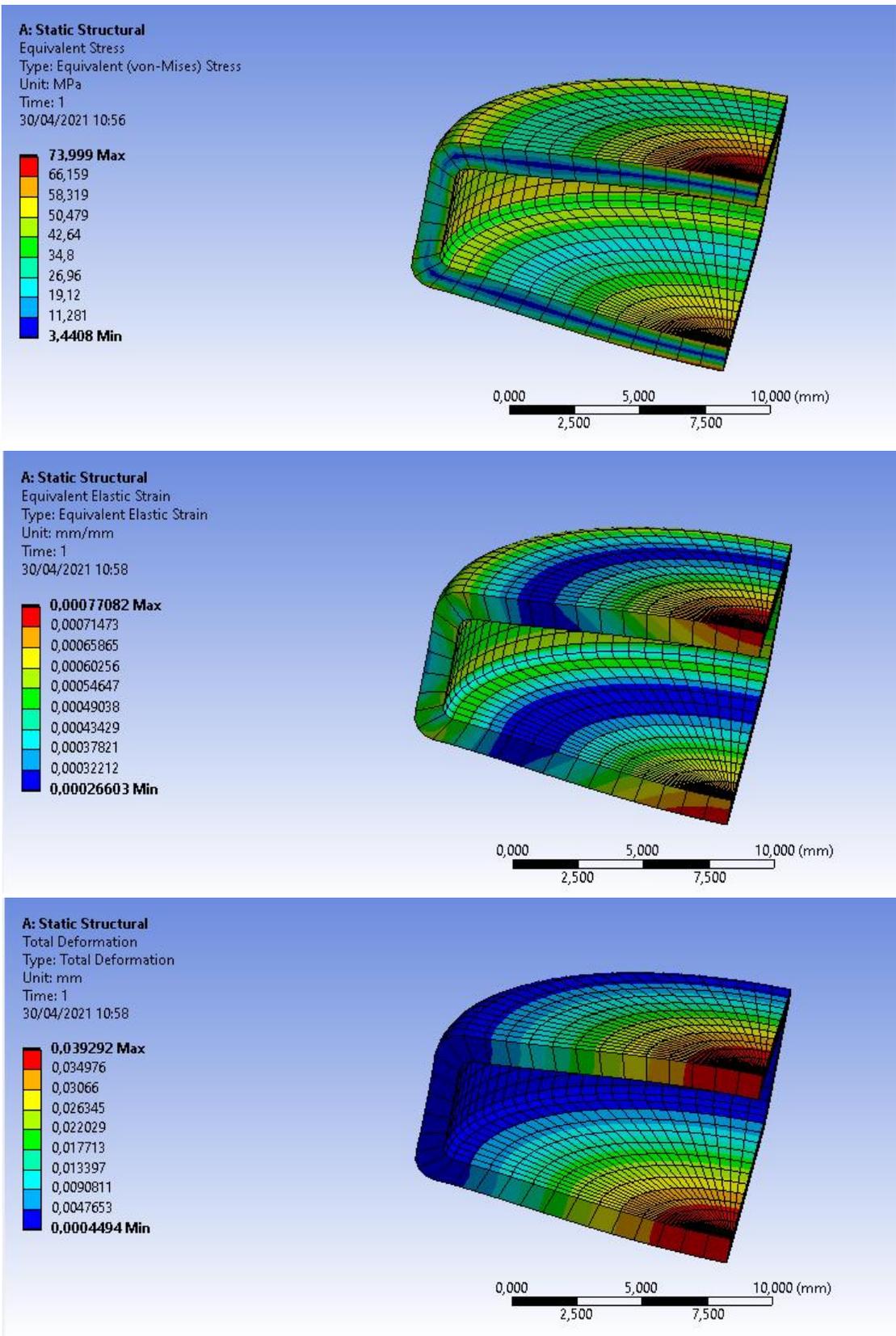
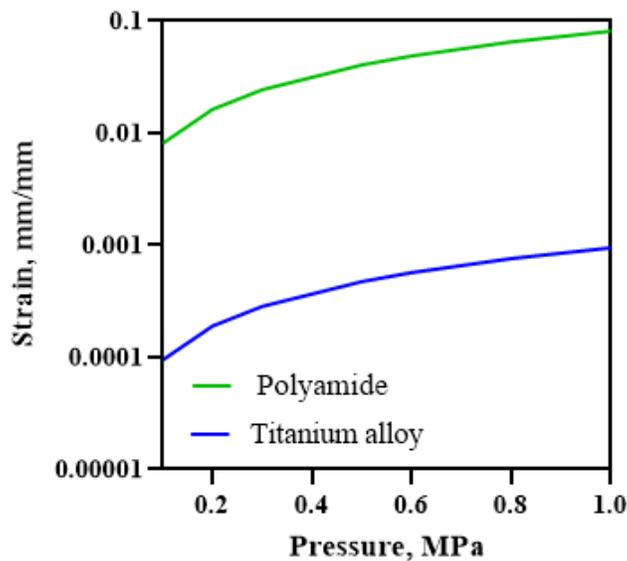
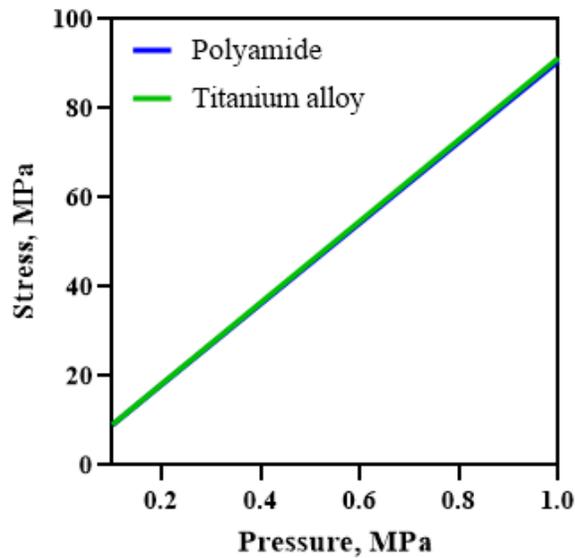


Figure 3_Titanium alloy FEM simulation results

FEM structural analysis of the model for different values of internal pressure are executed, focusing on the range 0-1 MPa to reflect the working condition of the device in the real future application. Data analysis is performed starting from results obtained with FEM simulations. The main interest is for the most loaded point of the capsule; for this reason, the graphs are built considering the maximum structural parameters values. After data analysis with Excel, three charts showing the behavior of stress, strain and deformation as a function of the internal pressure are obtained. In figures below are reported the data analysis results, in the strain and deformation graphs the Y axis is in logarithmic scale in order to show the comparison between the two materials.



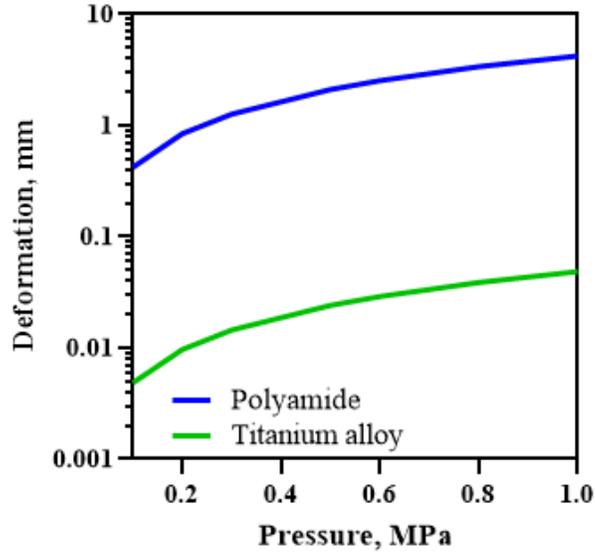


Figure 4_ Comparison of simulation results for stress, strain and deformation

The trend results to be linear in the pressure range 0-1 MPa for all the three parameters analyzed. The parameter-pressure characteristic is computed for Titanium alloy and Polyamide capsules and results obtained are summarized in the table below.

Structural parameter	Titanium alloy	Polyamide
Equivalent Stress	$y = 91.29 x$	$y = 90.65 x$
Equivalent Elastic Strain	$y = 10^{-3} x$	$y = 8.16 \cdot 10^{-2} x$
Total Deformation	$y = 4.85 \cdot 10^{-2} x$	$y = 4.22 x$

Table 2_ Structural parameter-pressure characteristic for titanium alloy and polyamide

In conclusion, the purposes of the FEM simulations performed on the discoidal model are the following:

- simulate and evaluate the structural behavior of the capsule as a function of the internal pressure applied;
- identify the most loaded point of the capsule and obtain its strain and deformation values.

The analysis of the results leads to the identification of the most loaded point in the centre of the circular surface. The main interest is for the strain results since the detection of the internal pressure is achieved using a strain gauge, as described in section 2.4. This information is used in the choice of the strain gauge sensor and in the selection of the position of application. After conducting this

preliminary structural analysis, it is possible to proceed with the detailed design of the capsule and its fabrication described in the next section of the manuscript.

2.3 Capsule design and fabrication

This section describes the design procedure for the capsule model using a computer-aided design (CAD) software. SolidWorks uses the principle of parametric design and generates three kinds of interconnected files: the part, the assembly, and the drawing. After definition of the final design, the capsule fabrication is achieved through metal and polymer Additive Manufacturing (AM) processes. Two prototypes of the capsule are manufactured using the two AM material previously presented: titanium alloy as metal material and polyamide as polymeric material.

2.3.1 Capsule design using SolidWorks

The design of the capsule is developed considering the specifications required for the future experimental testing of the device. For this reason, an inlet pipe is included in the model to facilitate the capsule pressurization during the experimental validation phase and to improve the sealing. The capsule design consists in two symmetric membranes coupled with one peripheral collar which provides the mechanical connection [10]. The capsule presents a discoidal shape and it is made of two different parts that are subsequently coupled thanks to the complementary edges inserted on the two components. A cubic nub is added for the insertion of a custom-made fitting to achieve the connection between the capsule and the test bench pipes circuit.

Capsule dimensions are the following:

- Diameter: 28 mm
- Thickness: 6 mm

The following figures show the capsule drawing built on SolidWorks.



Figure 5_Capsule drawings (SolidWorks)

2.3.2 Metal and polymer additive manufacturing processes

The capsules are manufactured through AM processes, in particular the titanium alloy capsule is made with Ti6Al4V by Selective Laser Melting (SLM) process and the polyamide capsule is made with grey PA12 by Multi Jet Fusion (MJF) process. SLM is based on the progressive melting of powder layers in vertical direction under inert atmosphere. This AM process involves the complete melting of the metallic powder particles under the action of a laser beam. The advantage of SLM is that near full density parts can be produced without the need for post-processing steps [11]. Different materials are suitable for this process such as aluminum alloys and stainless steel, but the most diffused materials are titanium alloys since they show high strength, corrosion resistance and biocompatibility. Because of these characteristics several AM titanium applications are present in the medical field [12]. MJF is a layer-by-layer polymeric AM process that involves the

use of a fusing agent and of a detailing agent in order to bond the powder through IR radiation. After the deposition of the build material, a thin layer of fusing agent and subsequently of detailed agent are placed to define the part geometry. Energy is then supplied in order to fuse the areas where the fusing agent has been applied. Transforming agents can be used to provide additional properties and to place them at specific desired locations in the fabricated part with wide flexibility. Another main advantage of this technology compared to other powder bed fusion processes is that the use of planar radiation can considerably reduce the build time [13]. Pictures of the Ti6Al4V capsule and PA capsule manufactured through the AM processes previously described are reported below.

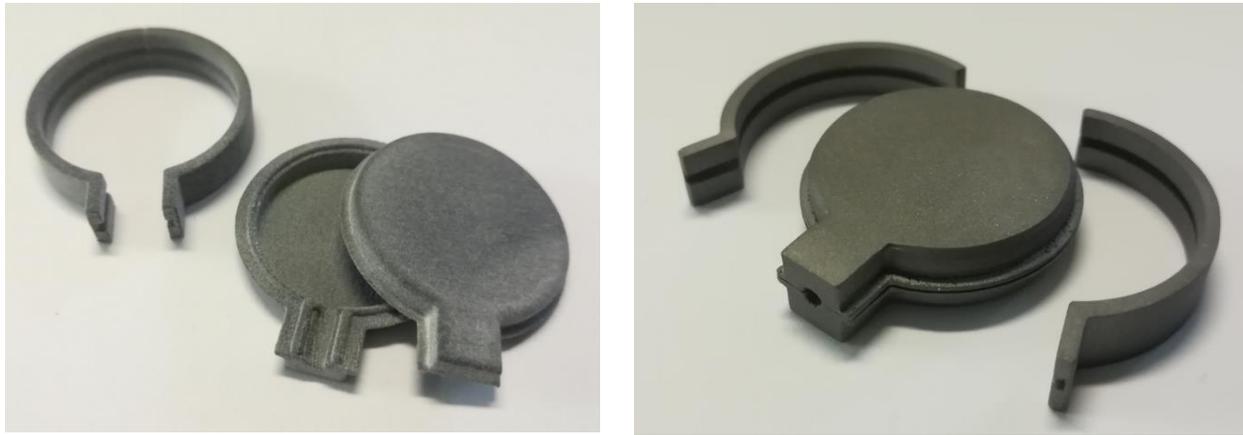


Figure 6_Ti6Al4V and PA12 capsules

2.3.3 Capsule assembly and sealing

The next step, after the manufacturing of the capsule component, is the device assembly and sealing. Different solutions have been developed and tested to avoid leakages and achieve effective connection to the hydraulic line of the test bench. The two parts of the capsule are sealed with epoxy resin and the inner edges provide the relative alignment and host the sealant adhesive [10]. The customized hand-made fitting is inserted in the cubic nob by means of a thread and sealed with the capsule during the assembly procedure. The following figures show the final set up after the assembly and sealing procedure; as shown in the figures, the device is connected to the push-in hydraulic fitting through a custom-made fitting. Details about constructions of this component are provided in chapter 3.1.



Figure 7 _Assembly and sealing of titanium and polyamide capsules

2.4 Strain gauge sensing system

This section focuses on the design and development of the sensing system based on the use of a linear strain gauge. The purpose of the sensing system is to provide a measure of the pressure inside the drug reservoir. The use of a strain gauge results to be an optimal solution since it returns as output a resistance value that allows to obtain the related strain value. From the strain measurement, through some analytical computations, it is possible to calculate the corresponding value of internal pressure. In the following pages the strain gauge working principle is explained

with a specific focus on the relationship between output resistance and strain measurement; also the theoretical formula to relate strain and internal pressure is illustrated. Further, the procedure for the strain gauge application and the device assembly are described.

2.4.1 Strain gauge theory

A strain gauge is a specific type of resistive sensor. In a resistive sensor the change in some physical variable produces a change in the resistance of a wire. By measuring this change in resistance, it is possible to determine the corresponding change in the physical variable. The electrical strain gages are transducers that transform displacement measurements into an electrical signal. The gauge is attached to the object by a suitable adhesive; as the object is deformed, the foil is deformed, causing its electrical resistance to change. The figure below shows the different parts of the strain gauge: the active grid length is the operative length of a strain gauge that is useful in strain measurement, the solder pads are used to connect the resistive foil with the electrical wires to acquire values from the sensor [14].

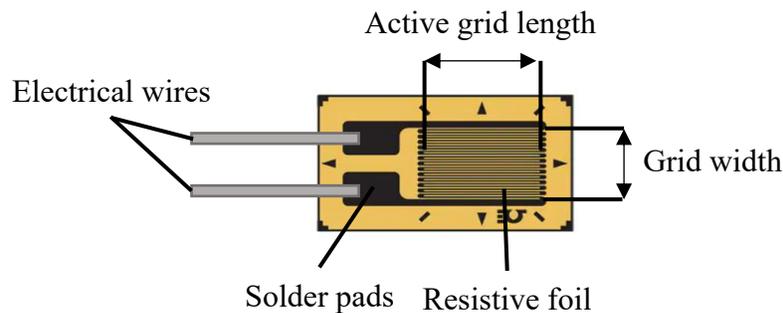


Figure 8_Electrical linear strain gauge schematic

2.4.2 Strategy of experimental investigation

The initial design phase of the activity is devoted to the study of a method to measure the pressure inside the device. In the following chapter is explained the strategy of experimental investigation that has been developed for this specific purpose.

2.4.2.1 Resistance-strain relationship

Strain gages are made of thin metal wires whose resistance changes whenever it is strained. As the wire is strained, its length L and its cross-sectional area A changes, which leads to a change in resistance R given by the formula:

$$R = \frac{\rho L}{A} \quad (11)$$

Deriving the equation above with respect to each variable it is possible to obtain the following:

$$\frac{dR}{R} = \frac{dL}{L} + \frac{d\rho}{\rho} + \frac{dA}{A} \quad (12)$$

From the previous equation, considering that $\varepsilon = \frac{dL}{L}$, the gage factor GF can be defined as follows:

$$GF = \frac{dR}{R\varepsilon} \quad (13)$$

The formula that relates the resistance value to the strain is summarized by the following expression:

$$\varepsilon = \frac{\Delta R/R}{GF} \quad (14)$$

The final result is the relationship between the strain and the change in resistance of the wire inside the strain gauge [15].

2.4.2.2 Strain-pressure relationship

In order to obtain the transfer function that relates the strain and the pressure, it is necessary to identify the literature case that best approximates the device. The aspects considered are the following:

- the strain gauge is applied in the most loaded point, this point is identified by means of FEM structural analysis and corresponds to the centre of the circular surface of the device.
- the surface of the device on which the strain gauge is applied can be considered as a circular plate with uniform pressure and fixed boundaries.

Therefore, the device is analysed referring to the theory of a flat circular plate with uniform load applied and clamped edges. From literature, it is known that tangential and radial strain are the same at the centre of the plate, the figure below shows the strain distribution in a circular plate with fixed boundaries and uniform pressure applied.

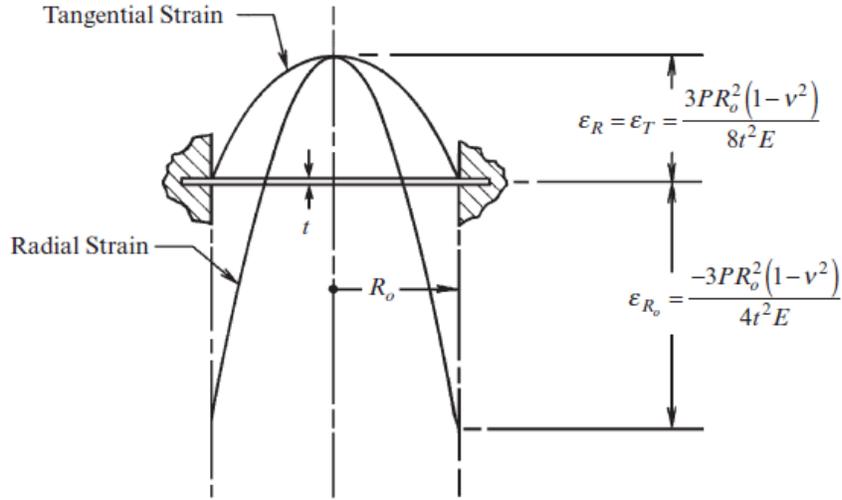


Figure 9 _Strain distribution in circular plate with fixed boundaries and uniform pressure [16]

It is possible to compute the theoretical value of the strain at the centre using the following formula [16]:

$$\varepsilon_R = \varepsilon_T = \frac{3PR_o^2(1 - \nu^2)}{8t^2E} \quad (15)$$

where

ε_R is the radial strain

ε_T is the tangential strain

P is the pressure applied [Pa]

R_o is the radius of the circular plate [mm]

t is the plate thickness [mm]

ν is Poisson's ratio

E is the Young's modulus [Pa]

Equation (15) shows the relationship between strain and pressure, it is possible to reformulate this equation in order to have the pressure at the centre of the plate as a function of the strain. The final transfer function that relates strain and pressure is the following:

$$P = \frac{8t^2E\varepsilon}{3R_o^2(1 - \nu^2)} \quad (16)$$

In conclusion, this section describes the strategy to measure the pressure inside the capsule starting from the output resistance value of a linear electrical strain gauge.

2.4.3 Strain gauge installation procedure

The strain gauge is applied in the most loaded point where strain and deformation are maximum, in this way it is possible to obtain the best accuracy in the final measurement of the internal pressure. Theoretical analysis and FEM simulations result in identification of the most loaded point at the center of the plate. The electrical strain gauge is installed on one of the two circular surfaces of the drug reservoir. Before application, the surface of the capsule has been properly prepared and cleaned with abrasive paper and cleaning agents. Cleanliness during the installation process is crucial to obtain accurate measurement results. Installation is realized using a cyanoacrylate adhesive. The strain gauge used presents the specifications reported in table below.

Resistance R at 24°C	350 $\Omega \pm 0.3\%$
Gage factor GF at 24°C	2.08 nom

Table 3_Electrical strain gauge specifications

In figures below are shown the installation of the strain gauge on the polyamide capsule and a schematic of the linear strain gauge used.



Figure 10_Installation of the linear electric strain gauge

The next step after the installation of the strain gauge is the welding of the wires in order to create the electrical circuit used for data acquisition from the sensor, detailed description of the data acquisition procedure is reported in section 3.2. An electrical tin solder is used for the soldering procedure and tin is used as welding material.

3 Experimental characterization and validation of drug reservoirs with strain sensors

The purpose of this section is to describe the methodology for the characterization in terms of strain-pressure relationship of the drug reservoirs with embedded strain sensors. Additionally, experimental validation and discussion of results obtained is reported. The first subchapter focuses on the test bench conceiving and fabrication; the aim of the test bench is to pressurize the device using a fluid and providing an adjustable value of internal pressure. The following part explains the development and realization of the measurement acquisition system; for this purpose, a strain gauge bridge amplifier is used to acquire the strain gauge resistance output value and convert it into a strain measurement. The main objectives of the experimental activity can be resumed as follows:

- evaluate the accuracy and repeatability of the measurement chain;
- provide range of measure and gain factors;
- compare results obtained from experimental tests with theoretical computations to validate the strategy of experimental investigation;
- demonstrate applicability of the device in performing continuous measure of pressure inside the capsule.

3.1 Test bench design and fabrication

The second phase of the project is the design and fabrication of the test bench, the purpose of the test bench is to impose a pressure inside the device and obtain a system able to adjust the pressure value in a simple and effective way. Through the use of the test bench, it is possible achieve device characterization in terms of strain-pressure relationship and to test the device and the sensor system to validate the method of measure.

3.1.1 Design principles

The purpose of the test bench is to pressurize the device; the most suitable solution to reach this goal is the use of a pneumatic cylinder able to pressurize a fluid through the application of a longitudinal force. A basic linear cylinder consists of a body containing a movable piston and a

rod connected to the piston. End caps are fastened to the cylinder body barrel. The rod passes through a hole in one end of the cylinder. In figure below is shown the schematic of a pneumatic cylinder with its main components.

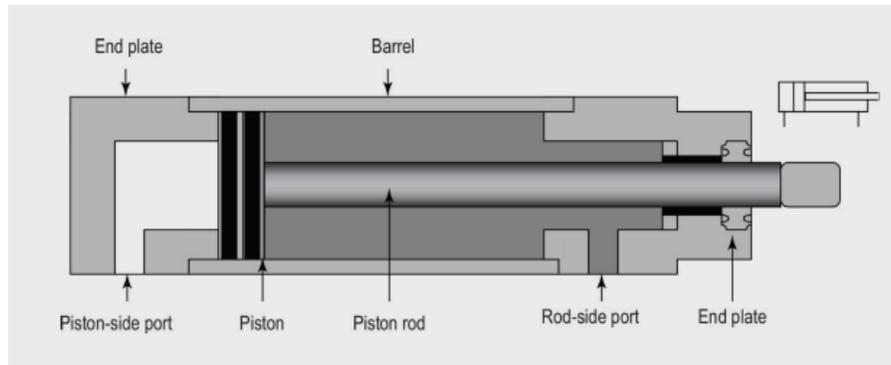


Figure 11_Pneumatic cylinder schematic

The piston can move in horizontal direction performing extraction and retraction movements. The piston is a loose fit inside the cylinder barrel, and it is held by a suitable seal. Apart from transmitting force to the rod, it must also act as a sliding bearing in the barrel. Pistons are usually made of cast iron or steel [17]. It is possible to obtain the relationship between the force applied to the piston rod and the pressure of the fluid. The effective area considered for the calculation is the full area of the cylinder bore and we can write the following equation:

$$F = \frac{\pi D^2 P}{40} \quad (17)$$

where

F is the force expressed in [N]

P is the pressure of the fluid in [bar]

D is the movable piston diameter in [mm]

The following figure shows the real pneumatic cylinder used for the fabrication of the test bench.



Figure 12_Pneumatic cylinder for test bench

The test bench needs a mechanism that allows to adjust the fluid pressure value in a simple and effective way. A specific solution based on the screw mechanism has been designed using a hexagonal head screw, a tapped standoff and a wrench. It is possible to realize the kinematics to move the piston rod and set the pressure in the following way: the head of the screw is located in a fixed position using an angle bracket, through the wrench the hexagonal tapped standoff is turned and the screw is extracted realizing a longitudinal displacement that produces a force on the piston rod. The schematic in figure below explains the kinematics of the mechanism composed of screw and tapped standoff.

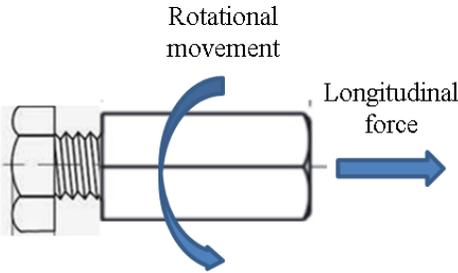


Figure 13_Pressure adjustment system in the test bench

In the fabrication of the test bench this mechanism has been located on a wooden board using an angle bracket to maintain the components in a fixed position, another angle bracket installed on the board is used to constraint the end cap of the cylinder. The pressure regulation system is completed with a pressure gauge used to display the pressure value and perform the desired adjustments.

3.1.2 Components and fabrication

The previous chapter is devoted to design principles on which the development of the test bench is based. The following pages are dedicated to the description of the components of the test bench and explanation of the fabrication procedure and general set up. The main component of the test bench is the pneumatic cylinder used to pressurize the device. The capsule samples are connected to a hydraulic piston to modify the inner pressure which is controlled by a barometer. Water is used as incompressible fluid in the tests. The choice of water is made in order to simulate the real working condition. The cylinder is fixed to a wooden board with cable ties. Two angle brackets are screwed to the board, one is used to constraint the end cap of the cylinder, the other provides a rigid support through which the piston rod can be moved. A hexagonal head screw, a tapped standoff and a wrench are used to set the pressure value. Using some metal plates, fixed to the bracket, a seat for the hexagonal head of the screw is created to avoid its rotation during the unscrewing of the standoff from it. A circuit of PFA pipes is used to bring the pressurized water to the device. This circuit is connected on one side to the posterior chamber of the pneumatic cylinder and on the other side to the capsule through the custom-made fitting. Push-in fittings are used to connect the components and the pressure gauge is located at the end of a branch through a T push-in fitting to indicate the operating pressure in a 0÷10 bar range scale. The sealing is guaranteed wrapping PTFE seal tape at the ends of PFA pipes where the tube is inserted in the fitting. A schematic of the test bench is represented in the following figure.

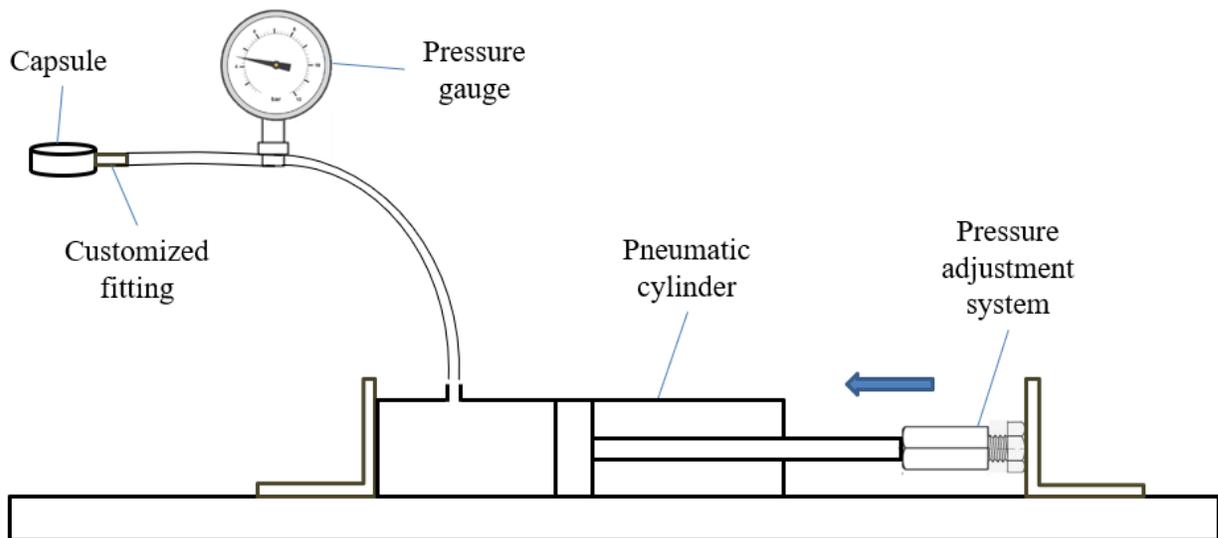


Figure 14_Test bench schematic

A critical part of the test bench design is the realization of the fitting that is inserted in the device and enables pressurization of the capsule. To this purpose a specific hand-made fitting is designed and produced based on the following considerations:

- on the one hand the hand-made fitting must fit with the push-in fitting;
- on the other hand, it needs to be inserted inside the capsule.

The push-in fitting has a threaded M5 extremity with diameter of 5 mm hence the hand-made fitting needs a threaded hole of the same measure. The device has a hole of 4 mm hence a M4 grub screw is needed on the other side. For the fabrication of this component, we use an aluminum rod drilled with a through hole threaded at both ends. At one end we tightened a fitting, connected to the main tube of the circuit, on the other we screwed in a grub screw which was first perforated, to create the connection with the capsule. Each capsule must first be drilled and threaded to be connected to the circuit. The design of the customized hand-made fitting is shown in the figure below.

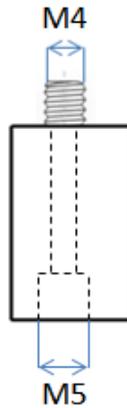


Figure 15_ Custom-made fitting

A list of all the components used for the test bench fabrication is provided in the following table, together with some images of the main components.

Test bench component	Quantity
Wooden board	1
SMC double-acting pneumatic cylinder	1
Angle brackets	2
Cable tie	2
Hexagonal head screw	1

Tapped standoff	1
Wrench	1
Pressure gauge with range 0-10 bar	1
Push-in fitting	5
Custom-made fitting	1
PFA pipes	3
Screws and nuts	8
PTFE seal tape	

Table 4_Test bench components



Figure 16_Test bench components

3.2 Measurement chain and data acquisition

After fabrication of the test bench, it is necessary to identify the procedure to acquire strain values from the strain gauge applied on the device. The following pages describe the method of measure and instrumentation used.

3.2.1 Data acquisition system

Data acquisition is realized using a specific instrument called QuantumX MX1615B. This strain gauge amplifier is suitable for precise and safe data acquisition from strain gauges in full-bridge, half-bridge and quarter-bridge configuration, as well as for strain gauge-based transducers, potentiometers, resistance thermometers. The module is equipped with 16 individually configurable sensor inputs. The linear strain gauge applied on the device needs to be connected to

the strain gauge amplifier through a quarter bridge configuration. The quarter bridge configuration can be implemented with a 3-wires or a 4-wires circuit. The quarter bridge 4-wires circuit is used in the design of the data acquisition system since it allows to reduce measurement errors compared to the 3-wires method. QuantumX strain gauge bridge amplifier is shown in the following figure.



Figure 17_QuantumX MX1615B

QuantumX is associated with a specific data acquisition software called Catman. This software allows data visualization, analysis and storage during the measurement and reporting after. In Catman it is possible to choose the type of strain gauge used for the application and to set up specifics such as resistance and gage factor and the software displays directly strain values, read from the strain gauge. The first step is to select “New DAQ Project” in Catman starting window: the software visualize the 16 input ports of the device and we are able to select the type of sensor we want to connect to each port, in this case “SG 4 wire 350 Ohm”. Second passage is to set the specifications of the strain gauge, in particular the gage factor using “Sensor adaptation” after right-clicking on sensor type, as shows in the following picture.

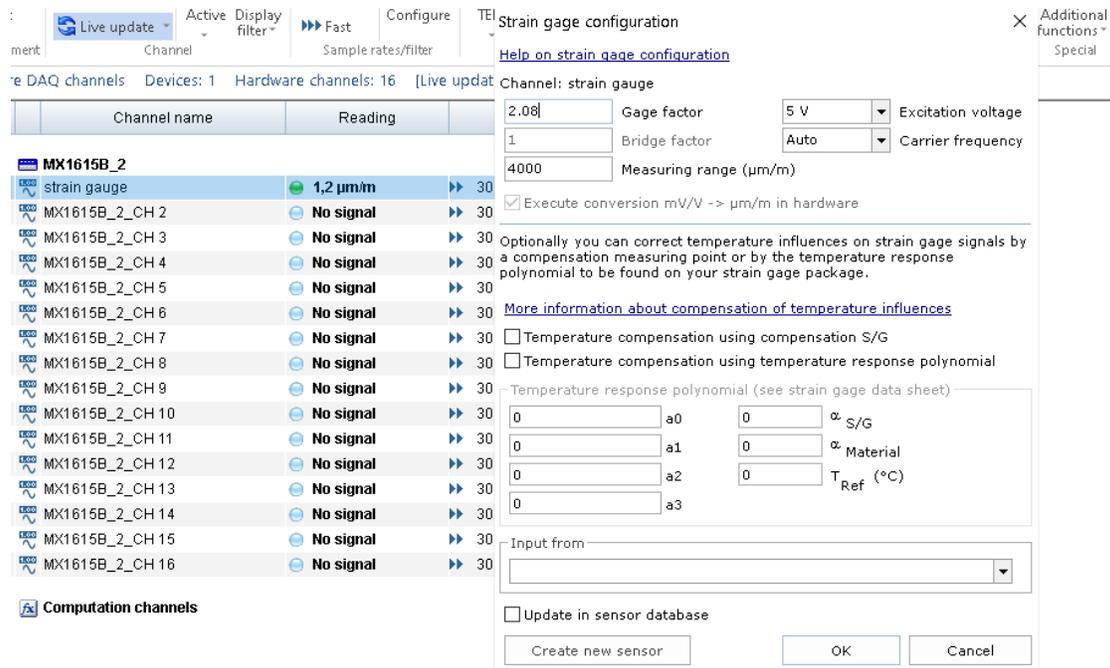


Figure 18_Catman settings for the strain gauge

It is then necessary to realize zero balance in order to assign the zero to the initial configuration of the strain gauge before starting the experimental data acquisition. Catman software returns as output the strain value in $\mu\text{m}/\text{m}$ as a function of the time and allows to perform continuous strain measurements.

3.2.2 Validation of the acquisition system

An important step is the validation of the acquisition system to ensure that the strain gauge is working properly and to verify the correctness of the wiring with the QuantumX amplifier. The validation procedure consists in computing the theoretical expected strain and compare it with the experimental one; for this purpose, a cantilever beam has been chosen. The mechanical strain in the cantilever beam is measured using a linear strain gauge and the results are compared with theoretical strain values calculated from equation derived from solid mechanics. Considering a simple cantilever beam subject to a force at the end of the beam, the strain formula can be obtained from literature [15]. The linear strain gauge is applied on the top of the cantilever beam in a quarter bridge configuration, as shown in the following figure.

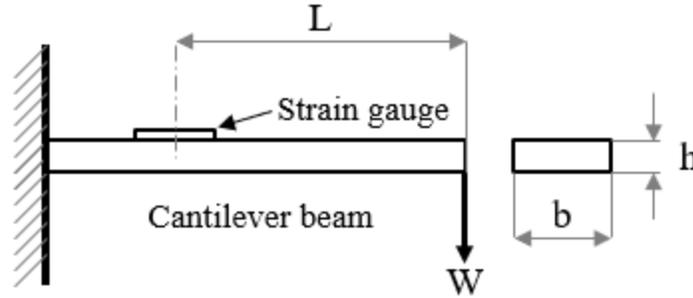


Figure 19_Cantilever beam subject to a force W

If a force W is applied at the end of the cantilever beam, the strain gauge is subjected to the following stress:

$$\sigma = \frac{6WL}{bh^2} \quad (18)$$

The stress and the strain are related with the following formula:

$$\sigma = \varepsilon E \quad (19)$$

Therefore, the resulting strain is expressed with the following equation:

$$\varepsilon = \frac{6WL}{Ebh^2} \quad (20)$$

where

W = force applied at the end of the beam

L = length between strain gauge and the point of application of the force

b = width of the beam

h = thickness of the beam

E = Young's modulus

In order to perform experimental measurements to compare with theoretical results, a real model of cantilever beam with a weight applied at the end of the beam is realized and the linear strain gauge is installed on the top of the beam at a precise distance from the end of the beam, as shown in the following figure.

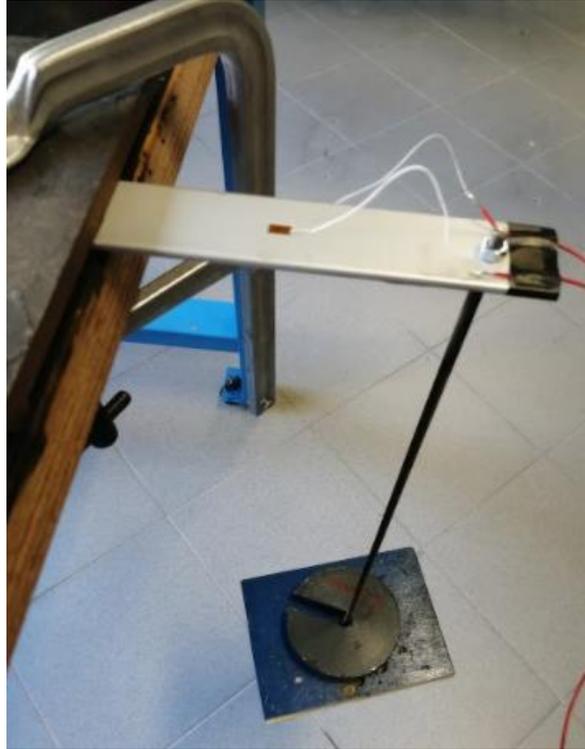


Figure 20 Cantilever beam in quarter bridge configuration

The cantilever beam presents the following characteristics:

$$h = 3.9 \text{ mm}$$

$$b = 3 \text{ cm}$$

$$L = 6.4 \text{ cm}$$

$$E = 6.5 * 10^{10} \text{ N/m}^2 (\text{Aluminum})$$

It is possible to compute the expected strain using the formula previously discussed, the strain is proportional to the weight applied according to the following:

$$W [N] = \text{weight [kg]} * 9,81 \left[\frac{m}{s^2} \right] \quad (21)$$

The strain gauge is connected to the acquisition system after performing the soldering of the two wires. Referring to QuantumX Operating Manual, the wiring has been realized using port 4, 5, 6 and 7 in order to realize 4-wire quarter bridge circuit used to acquire strain values from the linear strain gauge. A schematic of the connection circuit is reported in the following figure.

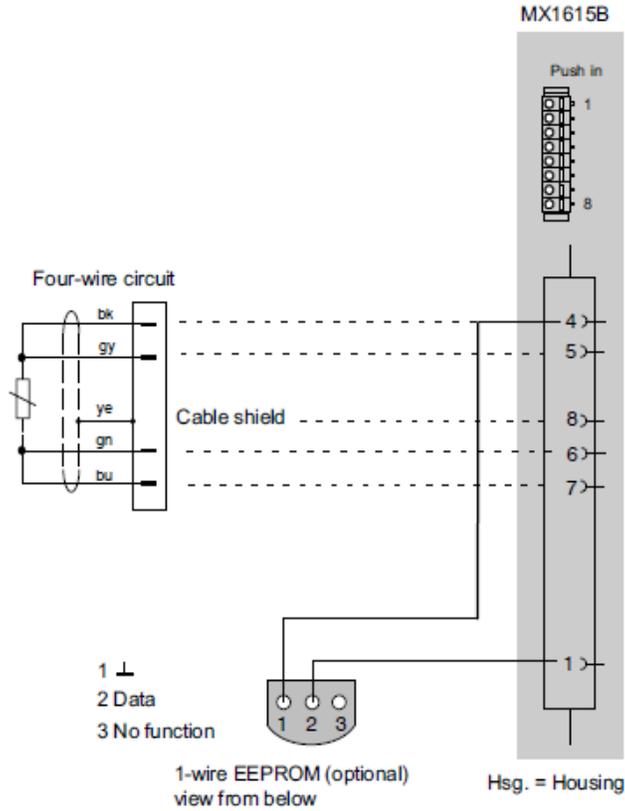


Figure 21 *Quarter bridge configuration for QuantumX*

After mounting the measurement chain, it is possible to perform data acquisition in order to compare experimental strain values with strain values computed theoretically through equation 20. In the theoretical computations the weight of the support structure must be considered, therefore a mass $m = 0.1$ kg is added. The experimental validation consists in the acquisition of strain values using QuantumX and Catman software with different weights applied at the end of the cantilever beam. Data analysis is performed and the transfer function of the strain as a function of the weight is obtained in order to compare the experimental results with the theoretical ones, as reported in figure 21.

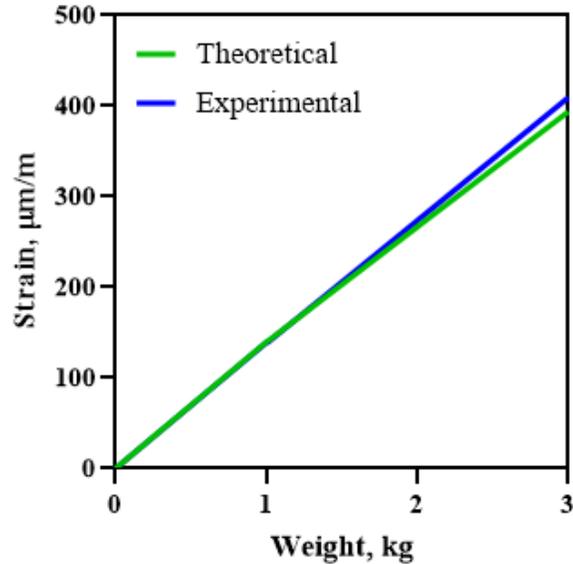


Figure 22_ Comparison between theoretical and experimental strain for the cantilever beam

The results confirm that the experimental values are consistent with the expected ones. The measurement chain is proved to be reliable and the data acquisition system in quarter bridge configuration for a linear electric strain gauge is validated. It is now possible to proceed with the experimental testing of the device through pressurization of the capsule using the test bench previously described.

3.3 Setup and testing methodology

This section describes the experimental testing of the two devices with the aim of achieving device characterization in terms of strain-pressure relationship and demonstrating reliability and applicability in continuous pressure measurement. The testing activity is performed using the test bench and the data acquisition system previously described. In the following pages the experimental setup and the testing procedure are explained; further, the results for titanium alloy and polyamide device prototypes are reported. The last part of this chapter focuses on the discussion of results obtained and comparison with the theoretical expected values. The experimental validation is performed with two capsules of metal and polymeric AM material, a linear strain gauge is installed at the center of one of the circular surfaces. The strain gauge is connected to the QuantumX amplifier through a 4-wire quarter bridge configuration. The testing activity is conducted according to the following standard procedure: the pressure inside the capsule

is gradually increased step by step, each pressure value is kept constant for 30 seconds. Through the use of QuantumX and Catman software, strain values are continuously acquired during the time. In the post-processing phase, strain values in the time interval of 30 seconds are analyzed to identify the mean strain value corresponding to a specific value of internal pressure applied. The final result of the post-processing is the attainment of the strain behavior as a function of the internal pressure. Interpolation of the results leads to obtaining the linear relationship between strain and pressure to compare it with the theoretical expected one. The overall instrumentation setup during the experimental activity is shown in the following figure.

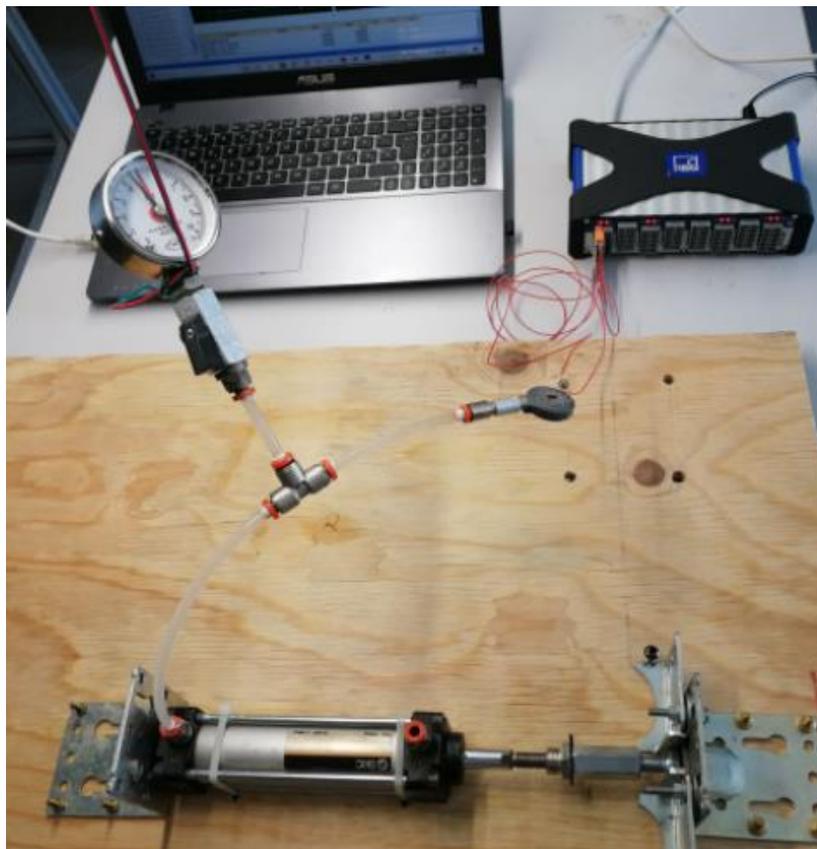


Figure 23_Experimental setup

An example of the strain continuous output displayed by Catman during the experimental activity is reported in the figure below.

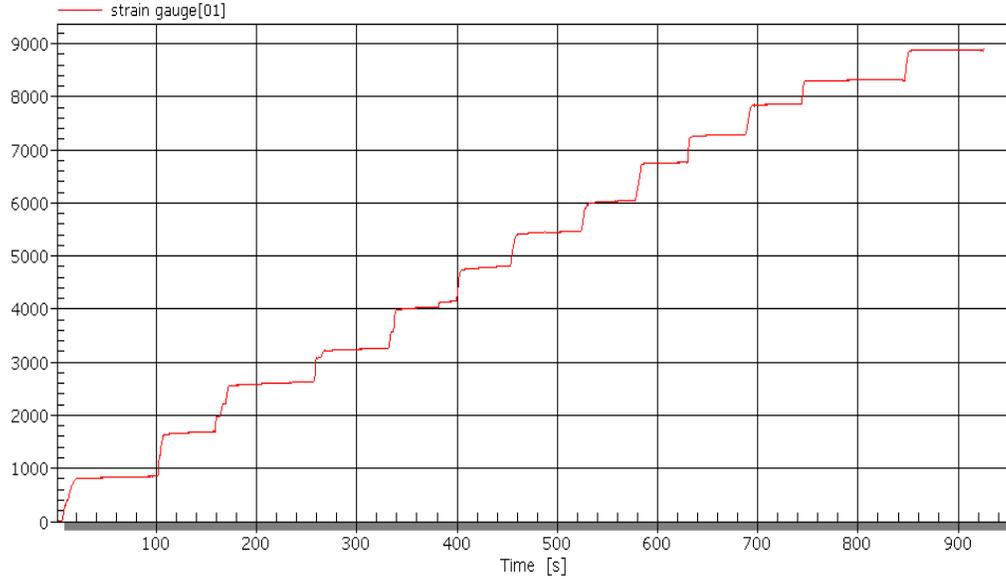


Figure 24 _Strain data acquisition with Catman

3.4 Results and discussion

This section focuses on the experimental outcomes for titanium alloy and the polyamide devices and on the discussion of the results obtained. Two tests under the same conditions are performed for each prototype in order to evaluate accuracy and repeatability of the measurement chain. The pressure gauge used to set the pressure value is a relative pressure gauge therefore pressure values are reported as relative pressures in bar. The strain values are acquired through Catman and displayed in $\mu\text{m}/\text{m}$. Data analysis leads to obtaining the strain as a function of internal pressure; the results are compared to theoretical ones to validate the device functioning and the overall measurement chain. The theoretical function that relates pressure and strain at the center of the circular surface is computed through equation 15, reported below:

$$\varepsilon_R = \varepsilon_T = \frac{3PR_0^2(1 - \nu^2)}{8t^2E}$$

For each capsule, material and dimension specifications are used to compute the strain with different pressure values, further the theoretical function is calculated and compared to the experimental one. In the following pages experimental results for the two devices are reported and discussed.

3.4.1 Titanium alloy prototype

Theoretical computations are performed considering specifications of material Ti6Al4V used for capsule fabrication; regarding the geometrical dimensions, the radius of the capsule excluding the peripheral collar is considered. The table below summarizes capsule properties for titanium alloy device.

Capsule property	Value
E Young's modulus, Pa	1.07×10^{11}
ν Poisson's coefficient	0.323
t Thickness, m	1.00×10^{-3}
R_o Radius, m	1.20×10^{-2}

Table 5_Titanium alloy capsule specifications

The following table resumes the theoretical and experimental results for the sensorized device made of titanium alloy.

Relative pressure, bar	TEST 1 Mean strain, $\mu\text{m}/\text{m}$	TEST 2 Mean strain, $\mu\text{m}/\text{m}$	Theoretical strain at the centre, $\mu\text{m}/\text{m}$
0	0	0	0
0.5	18.2	21.4	22.6
1.0	36.7	40.8	45.2
1.5	53.8	58.8	67.8
2.0	73.7	77.2	90.4
2.5	90.6	95.7	113.0
3.0	110.2	115.5	135.6
3.5	126.6	134.7	158.2
4.0	148.1	155.0	180.8
4.5	166.3	174.8	203.4
5.0	189.2	192.5	226.0
5.5	208.4	209.9	248.6
6.0	229.3	227.0	271.2

Table 6_Titanium alloy capsule theoretical and experimental results

The figure below shows the comparison between the theoretical strain-pressure function computed from equation 15 and the experimental ones, obtained from interpolation of data collected during the two tests performed.

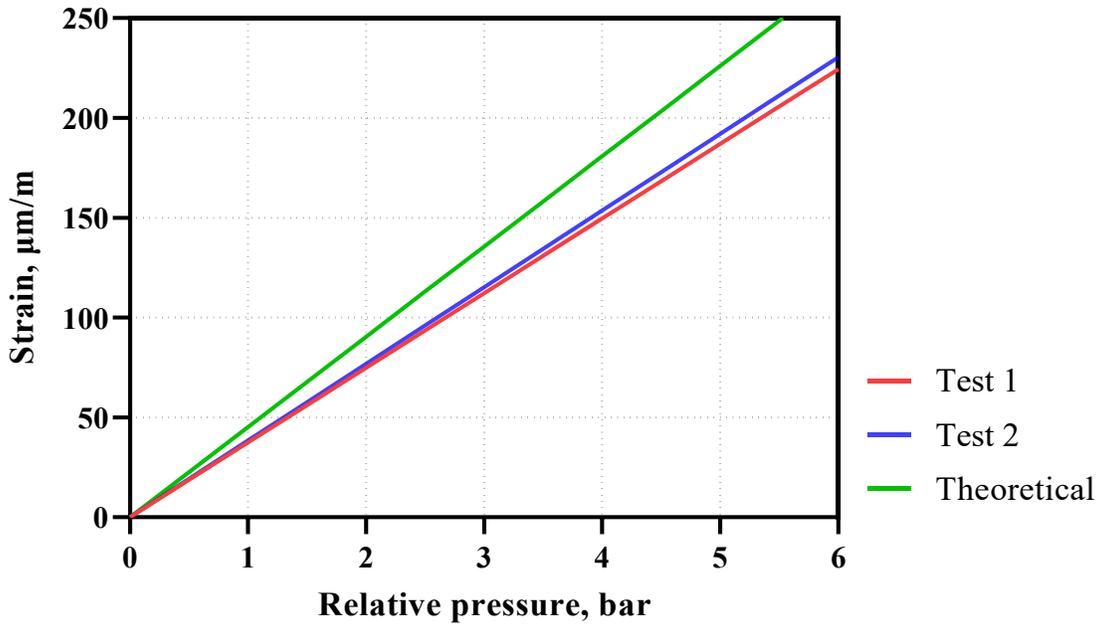


Figure 25_Theoretical and experimental strain-pressure characteristics for titanium alloy

3.4.2 Polyamide device

The table below reported the material and geometric specifications of the prototype fabricated with polyamide PA12.

Capsule property	Value
E Young's modulus, Pa	1.08×10^9
ν Poisson's coefficient	0.349
t Thickness, m	1.00×10^{-3}
R_o Radius, m	1.20×10^{-2}

Table 7_Polyamide capsule specifications

The following table resumes the theoretical and experimental results for the sensorized device fabricated with polyamide.

Relative pressure, bar	TEST 1 Mean strain, $\mu\text{m}/\text{m}$	TEST 2 Mean strain, $\mu\text{m}/\text{m}$	Theoretical strain at the centre, $\mu\text{m}/\text{m}$
0	0	0	0
0.2	818	737	528
0.5	1664	1675	1317
0.8	2581	2546	2108
1.0	3238	3174	2635
1.3	4024	3713	3425
1.5	4773	4348	3952
1.8	5436	5164	4742.3
2.0	6020	5538	5269
2.3	6750	5936	6059
2.5	7267	6373	6586
2.8	7853	6875	7377
3.0	8305	7228	7904
3.3	8873	7522	8694

Table 8_Polyamide capsule theoretical and experimental results

Data analysis leads to obtaining the strain-pressure characteristic through interpolation of experimental data. The discrepancy between the experimental data and the interpolation linear function can be explained considering that the polyamide shows high deformability and low tensile strength compared to the titanium alloy which brings to high sensitivity to deformation. It is also possible that the increasing of the pressure above a certain value caused plastic deformation of the capsule that results in a change of the strain behavior from linear to non-linear. The figures below underline the comparison between original experimental data and interpolation function.

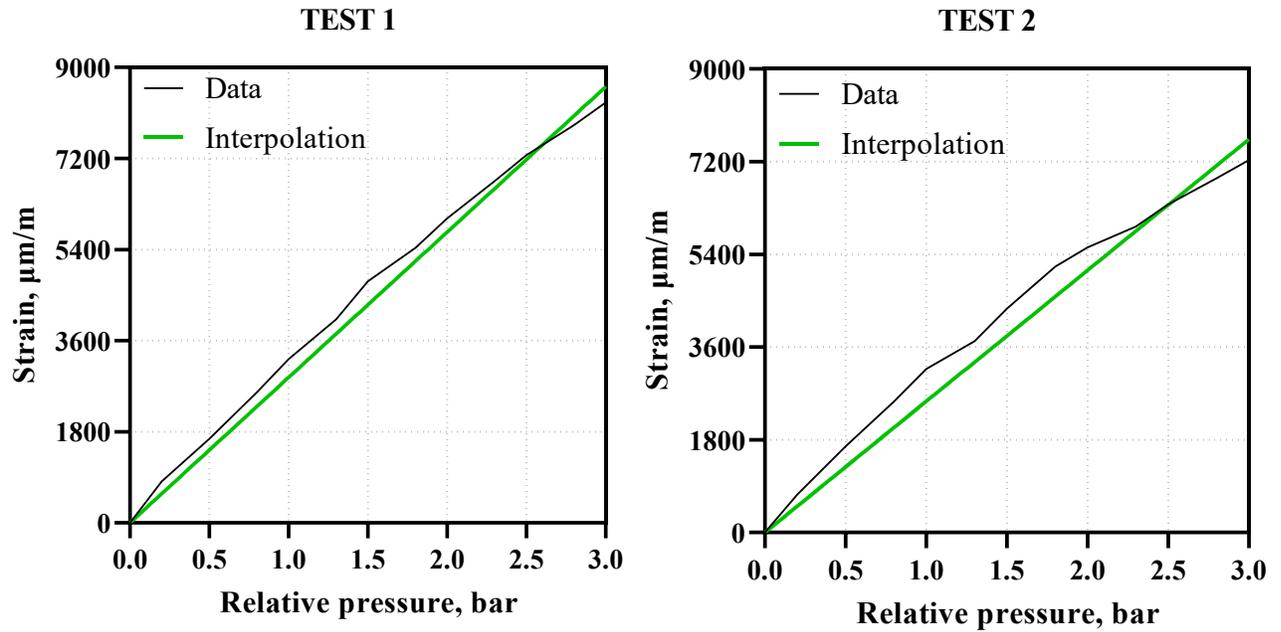


Figure 26_ Interpolation of experimental data for polyamide

The figure below shows the comparison between the theoretical strain-pressure characteristic computed from equation 15 and the experimental one, obtained from interpolation of data collected during the two tests performed.

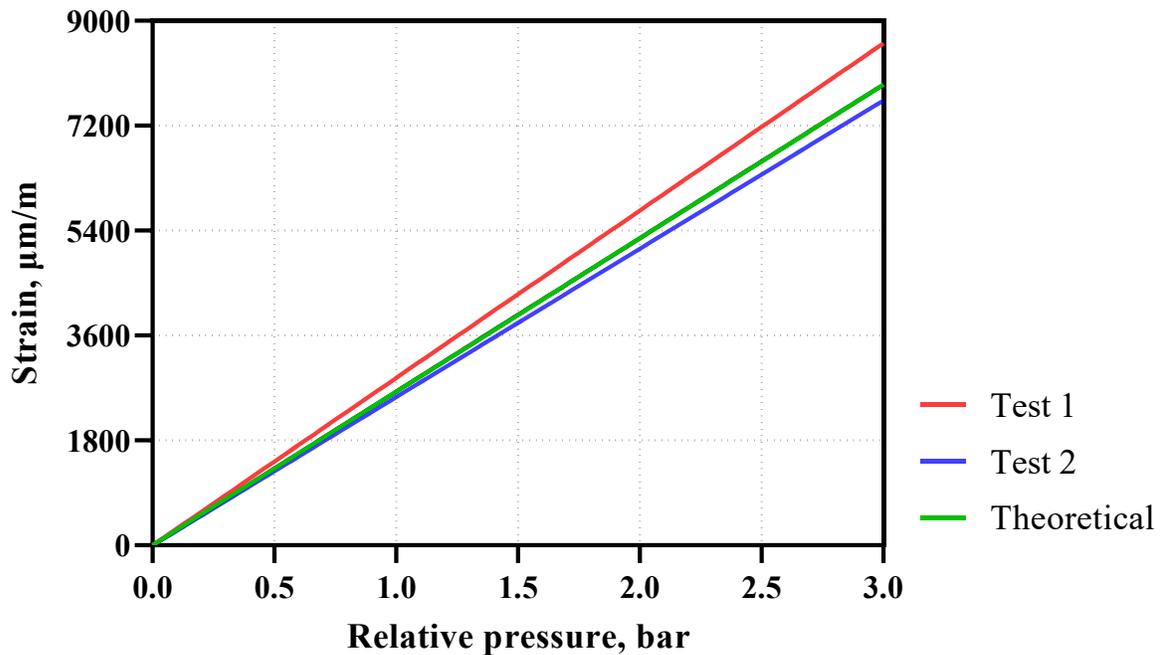


Figure 27_ Theoretical and experimental strain-pressure characteristics for polyamide

The following table summarizes the experimental results obtained with the prototypes made of metal and polymer AM materials: strain-pressure characteristics, gain factors and pressure range are reported.

Results	Titanium alloy	Polyamide
Test 1	$y = 37.4 x$	$y = 2871.2 x$
Test 2	$y = 38.4 x$	$y = 2544.1 x$
Theoretical	$y = 45.2 x$	$y = 2634.6 x$
Mean gain factor	$G = 37.9$	$G = 2707.6$
Relative pressure range, bar	$p = 0 - 6$	$p = 0 - 3.3$

Table 9_ Experimental results for titanium and polyamide

The comparison between theoretical and experimental strain-pressure characteristics demonstrates the accuracy of the measurement chain. The similarity of the results of the two experimental tests proves repeatability and stability of the overall system. The discrepancy is negligible and can be attributed to the following factors:

- the theoretical strain-pressure characteristic is computed at the center of the circular plate. However, the physical strain gauge returns the mean strain across the surface and not the exact point value at the center;
- the theoretical analysis is performed considering a circular plate with fixed boundaries subject to uniform pressure; this approximation is reasonable but does not consider the overall complexity of the device;
- the presence of a substrate of adhesive can affect the strain resulting on the surface.

The results obtained on titanium alloy and polyamide capsules equipped with strain sensors demonstrate the applicability to passive drug delivery. Through gain factors listed in table 9, it is possible to obtain the internal pressure from the strain measured with the strain gauge; therefore, the devices developed have proved to be suitable for continuous detection of the pressure inside the capsule. The resolution of the sensing method is comparable to the target resolution of osmotic pressure measurement, after filtering and decoupling the external disturbances. The AM titanium alloy shows a higher dimensional accuracy and mechanical strength, while high sensitivity and low cost are the advantages of the use of AM polyamide [10].

4 Design and fabrication of implantable device with digital pressure sensor

The second part of the project is the result of a six-months research activity at Houston Methodist Research Institute in Dr. Grattoni laboratory. The purpose of this work is the development of a second device able to perform direct pressure measurements using an integrated digital pressure sensor. The preliminary phase of the activity focuses on the study of a remotely controlled implantable device for tunable drug release. The aim of this analysis is to understand the structure of the device in order to use a similar technology for the development of a new device aiming at performing continuous monitoring of the osmotic pressure inside the reservoir. The device under study is developed to achieve sustained drug release with adjustable dosing and timing. A nanofluidic membrane allows to modify the drug release by leveraging a low intensity electric field. It is possible to increase, decrease or stop the rate of drug via Bluetooth remote command. The Printed Circuit Board (PCB) was designed for the control of therapeutic release and the Bluetooth low energy (BLE) communication was implemented through the system-on-chip (SoC) from Texas Instrument. In order to provide power supply to the board a commercially available discoidal battery was used. The wireless communication with a remote PC was implemented through a USB Bluetooth dongle and a Matlab script. In vitro test was performed to prove the stability of the communication system, the received signal strength indication (RSSI) was measured to demonstrate communication quality. Moreover, in vivo testing was performed through implantation in rats in order to prove biocompatibility and data transmission [18]. The integration of the strain gauge sensing system with the already developed PCB requires some modifications of the electrical circuit and in particular an amplifying and a filtering system are needed. In order to achieve direct pressure measurement and to avoid modifying the board circuit, new sensing solutions have been considered using a more expensive integrated digital pressure sensor. The first part of this chapter describes the analysis and testing of two different sensors in order to identify the best solution for our application. The next step is the development of the C++ code to interface the sensor with the microcontroller of the PCB and to achieve remote data communication via Bluetooth transmission. The last subchapter describes the device design and fabrication: the body structure is developed through a CAD software and fabricated through 3D printing techniques; moreover, device components and assembly procedure are explained in detail.

4.1 Integrated digital pressure sensors

Integrated digital pressure sensors result to be the most suitable option considering, in particular, the following factors:

- they can be directly integrated with the PCB without the need to modify the electrical circuit;
- they perform direct pressure measurement, providing a precise digital pressure value;
- they are miniaturized pressure sensors and their small dimensions are essential for device implantation for future *in vivo* experimentation.

The evaluation of many different integrated digital pressure sensors on the market results in the identification of the following two best solutions:

- HSPPAD143A digital absolute pressure and temperature sensor produced by AlpsAlpine.
- MS5837-30BA digital pressure and temperature sensor produced by TE Connectivity.

The first solution identified is a digital absolute pressure and temperature sensor with digital interface using I2C communication protocol. The pressure is detected by a MEMS sensor element using a piezo-resistive bridge circuit formed on the silicon diaphragm. The sensor element is connected to ASIC for signal conditioning, which has 17-bit ADC and temperature compensation capability [19]. The main specifications of the HSPPAD143A sensor are the following:

- Operating pressure range (absolute): 0.3 to 2.1 bar
- Dimensions: 3.1 x 3.1 x 2.6 mm
- Supply voltage: 1.7 to 3.6 V (typical 1.8V)
- Operating temperature: -40 to +85°C
- Current consumption: 1.8 μ A
- Digital interface I2C
- Output: 17-bit digital output
- Water resistant

Sensor drawings from different views are shown in the following figures, taken from the sensor datasheet.

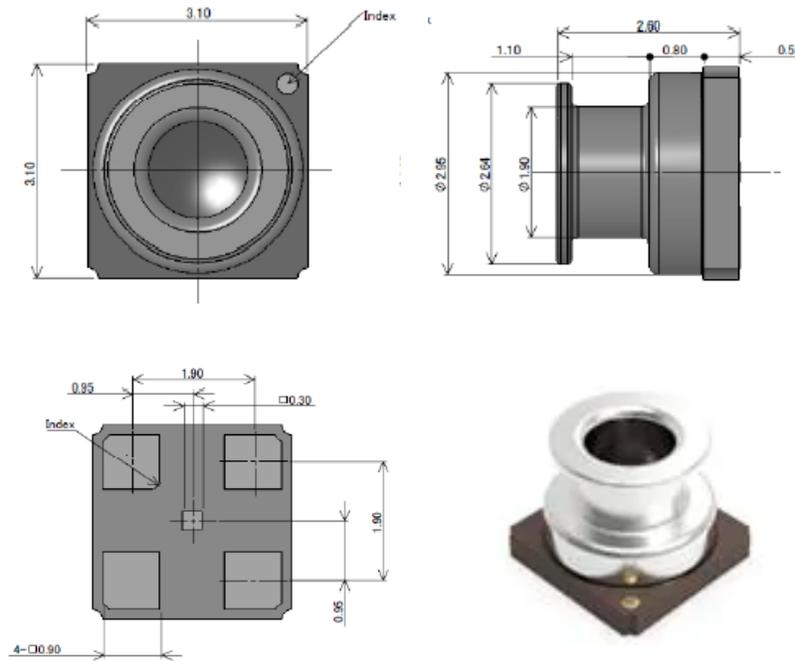


Figure 28_HSPPAD143A sensor

The second solution is the MS5837-30BA, a high-resolution digital pressure and temperature sensor with I2C bus interface. This sensor provides a precise digital 24-bit pressure and temperature value and can be interfaced to virtually any microcontroller. Moreover, a gel protection and antimagnetic stainless-steel cap make the module water resistant; this characteristic is essential since our application requires the sensor to work in aqueous environment [20]. The main features of the sensor are listed below:

- Operating range (absolute): 0 to 30 bar
- Dimensions: 3.3 x 3.3 x 2.75mm
- Supply voltage: 1.5 to 3.6 V
- Operating temperature: -20 to +85 °C
- High resolution module: 0.2 cm (in water)
- Current consumption: 0.6 μ A
- Digital interface I2C
- Output: 24-bit digital output
- Water resistant

The following figures from the sensor datasheet report the sensor drawing with dimensions.

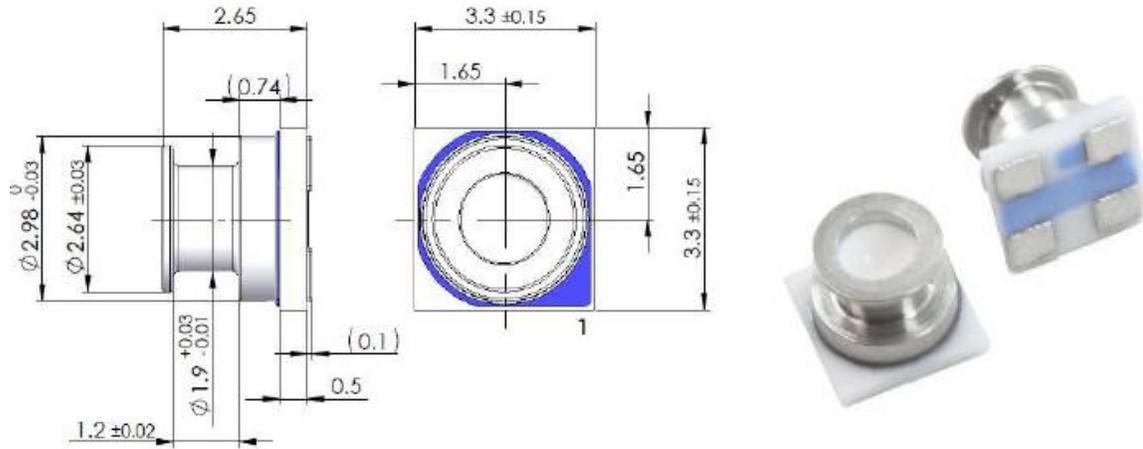


Figure 29_MS5837-30BA sensor

4.1.1 I2C communication protocol

One of the main advantages of these sensors is the use of the I2C protocol that allows the communication and data transmission between the sensor and the microcontroller of the PCB. The I2C protocol was developed by Philips Semiconductors in the early 1980s to allow for communication between various integrated circuits. It is also known as the “two-wire” protocol since two lines are used for communication: a clock line (SCL) and a bidirectional data line used for sending information back and forth between the master and the slaves (SDA). The I2C bus allows multiple slave devices to share communication lines with a single master device. The bus master is responsible for initiating all communications, the slave devices are not able to initiate communications, but only respond to requests that are sent by the master device. Each I2C slave device is characterized by a unique 7-bit address or ID number; when communication is initiated by the master, a device ID is transmitted and I2C slave devices react to data on the bus only when it is directed at their ID number [21]. It is possible to use the I2C protocol to achieve data transmission between an Arduino board and one or more sensors, in this case the Arduino acts as the master device. The standard I2C bus configuration requires pull-up resistors on both the clock and data lines, as shown in figure below. The value for these resistors depends on the specific slave device used.

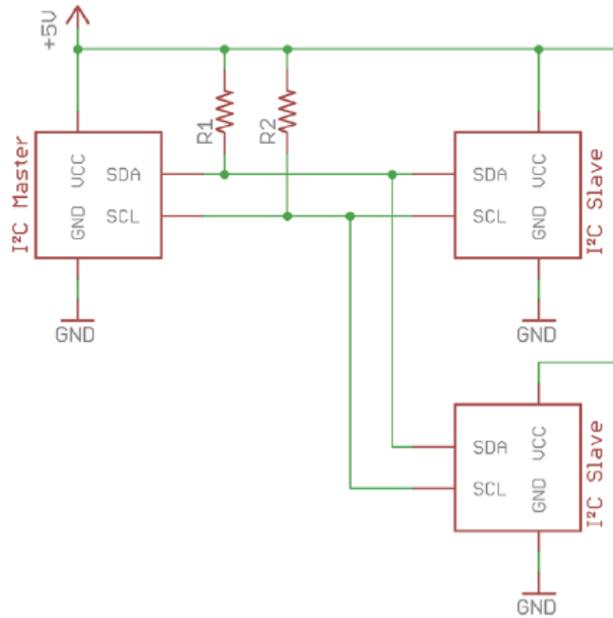


Figure 30_I2C master-slave configuration

The Arduino board presents specific pins devoted to I2C communication: the SDA and SCL pins corresponds to pins A4 and A5 respectively. In order to write the code that instructs the Arduino to request data from the I2C sensor through the Arduino’s built-in I2C interface, it is necessary to use the Wire Library. This library is included with the Arduino IDE and has ready-made I2C functions. To use the functions in the Wire library, we first need to add it to the script with the command `#include <Wire.h>`. After including the library, it is necessary to join the device on the I2C bus. The syntax for this is `Wire.begin(address)`. The address is optional for master devices and for the master Arduino sketch, it is just necessary to write the command `Wire.begin()` inside the `setup()` function. In order to reading data from the slave device to the Arduino board, the necessary steps are the following:

- send a message to the device’s address in ‘write mode’ to indicate that the master wants to read from the data register;
- send a message to the device’s address in the ‘read mode’ and request the necessary bytes of information from the device;
- confirm that all the bytes of information are received.

There are specific functions in the Arduino Wire Library that perform the actions listed above. `Wire.beginTransmission()` starts the communication with a slave device with the given address. `Wire.write()` is used to read from a specific register address.

Wire.endTransmission() indicates that the master finished writing to the device and that is now able to read from the slave I2C device.

Wire.requestFrom() asks for a certain amount of data and then returns the number of bytes that are actually received. This information is stored into a variable called returned bytes. This value is then checked; if it is zero, the I2C device did not return any data. This generally implies a hardware problem, such as the sensor not being wired up properly. Thus, an error is printed to the serial monitor and the program enters an endless wait loop if this condition is triggered. Assuming that data was received back from the I2C device, the 8-bit value is read into an integer variable with a *Wire.read()* command. After development of the Arduino code for the specific slave device, code is run on the Arduino board and through the serial monitor it is possible to read the values obtained from the sensor [21].

The LAUNCHXL-CC2640R2 board is a microcontroller development kit for rapid prototyping featuring the CC2640R2 microcontroller provided by Texas Instruments. This board is optimal for preliminary testing since the microcontroller is the same one mounted on the PCB. In order to realize I2C communication using the LAUNCHXL as the master device, DIO4 and DIO5 are used for SCL and SDA respectively. In figure below is reported a schematic from the board datasheet showing the pins layout.

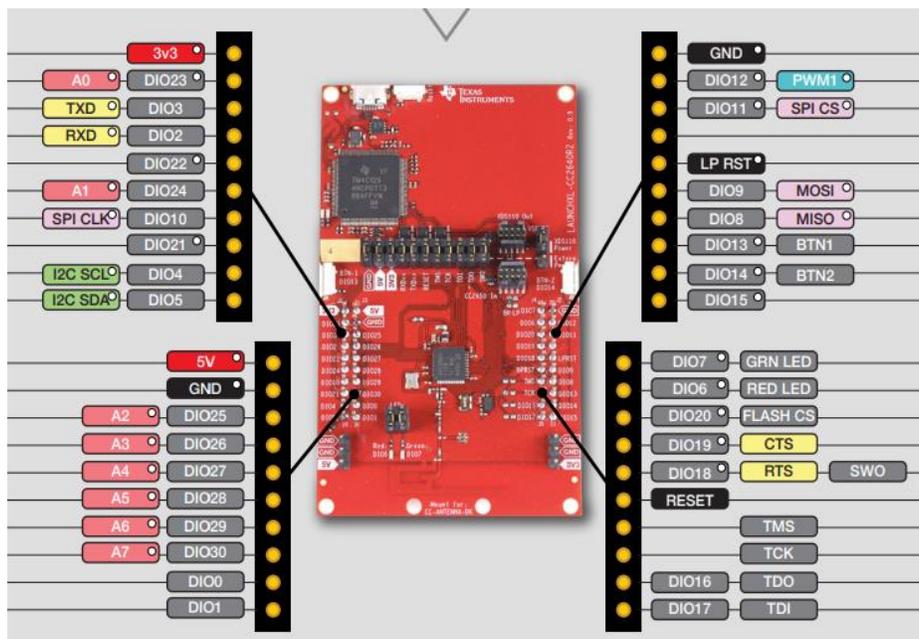


Figure 31 _LAUNCHXL board

A specific C++ code is developed and run on the board in order to acquire data from the slave pressure sensor. The C++ script is written using the ‘write’ and ‘read’ functions and the same logic of the Arduino script to perform data acquisition through I2C protocol. Data acquired are displayed in real time on the computer using the serial monitor.

4.1.2 Code development and testing

This section describes the algorithm for the development of the specific code for data acquisition from the two different sensors. For HSPPAD143A sensor the Arduino and the C++ code are developed; for MS5837-30BA sensor, for time reasons, we focused on the C++ code to proceed directly with the microcontroller programming. After code development, a testing procedure is designed to verify the correctness of software and hardware set up and to evaluate data transmission from the sensor through I2C communication protocol. The sensor testing requires the elements listed below.

- Pressurized reservoir: the sensor must be installed inside a closed reservoir that is pressurized in order to evaluate the pressure reading and transmission.
- Arduino or C++ code: it is necessary to develop a specific code to enable communication between the board and the device and acquire data from the sensor. After the uploading of the code on the board, the next step is to open the serial monitor window and ensure that your baud rate is set to 9600. It is possible to see a stream of data from the sensor showing the pressure in Pascals (Pa).
- Arduino or LAUNCHXL board: the board is the I2C master device that acquires pressure and temperature measurements from the sensor, which is the slave device. The board is also used to provide power supply to the sensor. The 4 pins of the sensor are connected to the board, 2 pins are dedicated to voltage supply, one pin is for the serial clock SCL and the last pin is for the serial data input/output SDA.

The following pages describe the details of the code algorithm for the two digital pressure and temperature sensors. The goal of this activity is to perform a sensor testing and validate the code for pressure data acquisition from the sensor.

4.1.2.1 HSPPAD143A sensor: Arduino coding

In order to write the Arduino code, the following information are needed:

- bus address of the sensor, also called slave address;
- addresses of the different registers.

The device or slave address is specified in the sensor datasheet (pag.18) as a binary-coded decimal number. The device has two different address number to perform reading and writing actions. The slave addresses for HSPPAD143A sensor in binary and hexadecimal numbering are reported in the table below.

Function	Binary	Hexadecimal
Read	10010001	0x91
Write	10010000	0x90

Table 10_HSPPAD143A slave address

Information regarding the register addresses and the bit assignments can be found in the register map reported in the sensor datasheet, shown in figure below.

12. REGISTER MAP

Reg add	Name	Cust. R/W/A	Test R/W/A	Full Name	Bit assignment								Init. Value
					bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
00	WIA	R	R	Who I am	0	1	0	0	1	0	0	1	49
01	INFO	R	R	Information	0	0	1	1	0	0	0	1	31
02	FFST	R	R	FIFO Status	FFE _V	-	-	FP[4:0]				00	
03	STAT	R	R	Status	BUSY	-	-	TRDY	-	PDOR	-	PRDY	00
04	POUTL	R	R	Pressure Output Low	POUT[7:0]								00
05	POUTM	R	R	Pressure Output Middle	POUT[15:8]								00
06	POUTH	R	R	Pressure Output High	-	-	-	-	-	-	-	POUT[16]	00
09	TOUTL	R	R	Temperature Output Low	TOUT[7:0]								00
0A	TOUTH	R	R	Temperature Output High	TOUT[15:8]								19
0E	CTL1	RW	RW	Control 1	-	-	-	-	-	-	PTAP[1:0]		13
0F	CTL2	RW	RW	Control 2	TMES	-	PMES	-	ODR[1:0]		MODE[1:0]		A0
10	ACTL1	RW	RW	Action Control 1	-	-	-	-	TDET	-	PDET	-	00
11	ACTL2	RW	RW	Action Control 2	SRST	-	-	-	-	-	-	-	00
12	FCTL	RW	RW	FIFO Control	FFEN	-	-	WMT[4:0]				10	
13	AVCL	RW	RW	Average Control	-	-	TFRQ[2:0]			AVG[2:0]			38
1C	PNUM	R	R	Product Number	1	1	1	0	0	0	0	0	E0
20	PDET	AC	AC	Pressure Detection Command	-	-	-	-	-	-	-	-	-
22	TDET	AC	AC	Temperature Detection Command	-	-	-	-	-	-	-	-	-
26	SRST	AC	AC	Software Reset Command	-	-	-	-	-	-	-	-	-
29	PTDET	AC	AC	P&T Detection Command	-	-	-	-	-	-	-	-	-

Figure 32_HSPPAD143A register map

Before reading pressure measurements from the sensor, it is necessary to send data to the sensor in order to set specific functions. The sensor can work according to three different modes:

- register action mode;
- continuous measurement mode;
- command action mode.

The Continuous mode results to be the most suitable for our application. To perform pressure measurement in continuous mode, the algorithm reported in the datasheet (figure 33) is used.

9. SAMPLE FLOW CHART

9.1 Continuous mode

This flow chart is showing from power on to pressure measurement in continuous mode.

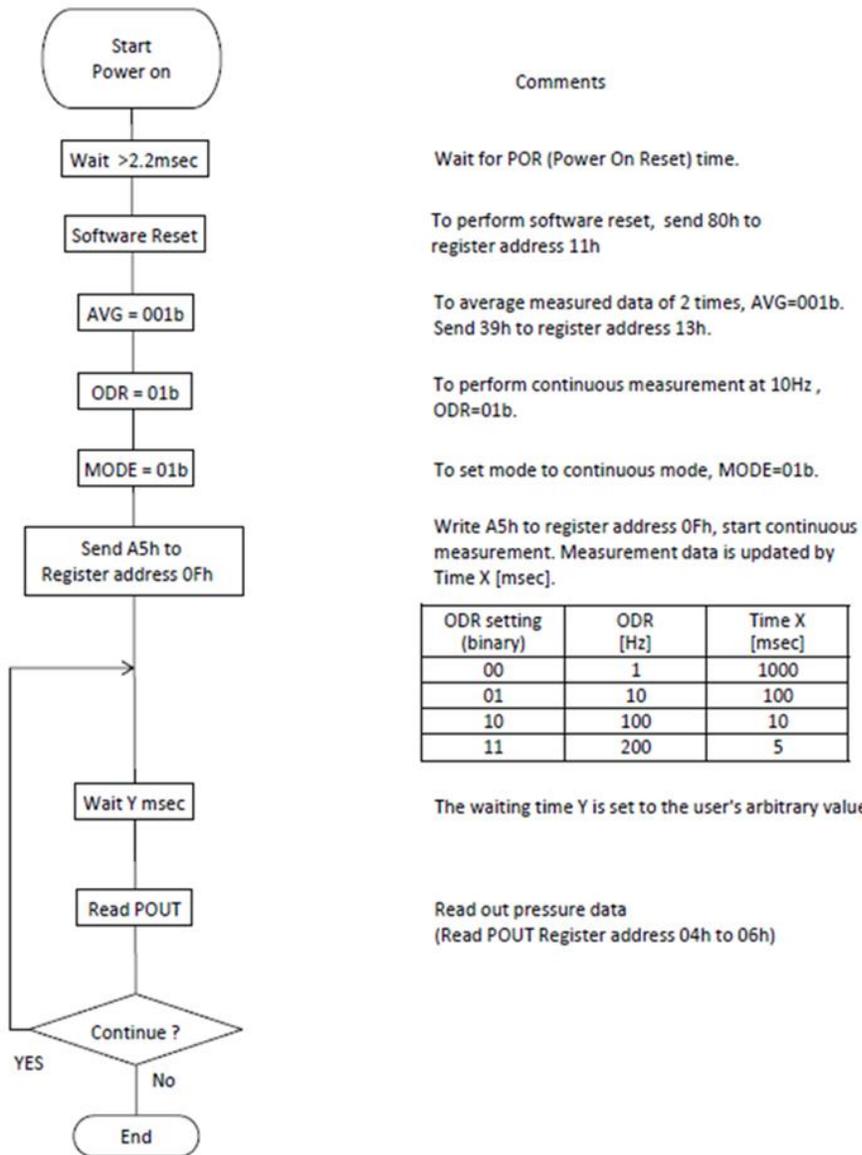


Figure 33_HSPPAD143A algorithm

The first section of the code is devoted to all the initializations necessary to start pressure measurement in continuous mode; initializations performed are the following:

- software reset;
- average measured data of 2 times;
- set mode to continuous mode;
- start continuous measurement.

The second part of the code performs the reading of the pressure values and there are also some calculations to translate the binary pressure value into a pressure value in bar. The complete Arduino code is reported in Appendix section. To connect the sensor to the host controller, it is necessary to implement a circuit using two 3.3kOhm resistors and a 100nF capacitor. The detailed schematic of the circuit from the datasheet is reported in figure below.

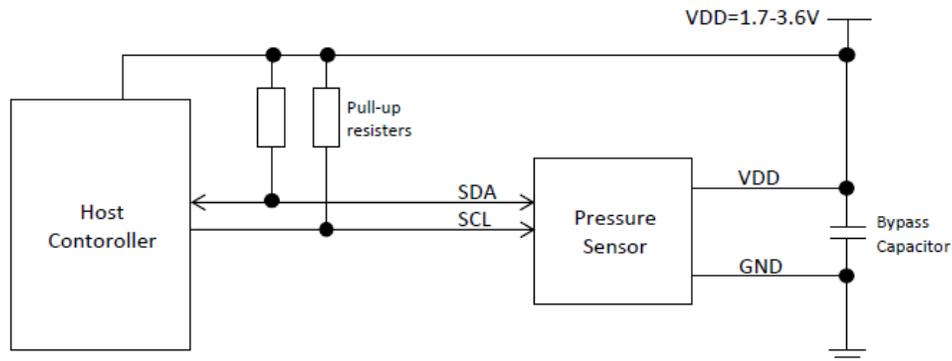


Figure 34_HSPPAD143A I2C circuit

The circuit is mounted on a breadboard which is wired both to the sensor and to the Arduino board. The sensor must be provided with a 3.3V power supply using the Arduino board and the pins devoted to I2C protocol are pins A4 and A5: A4 is for SDA while A5 is for SCL. The Arduino board is then connected to the computer on which the Arduino IDE software is used to run the code on the board. The sensor is mounted on a small reservoir connected to a syringe used to adjust the pressure inside the reservoir. The figures below show the pin layout from the datasheet and a drawing of the connections between the sensor and the Arduino board.

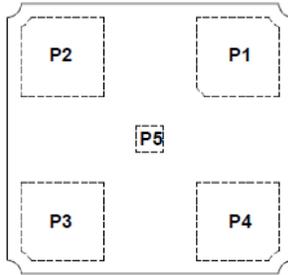


Fig.15 : Pin layout (Top view)

Table13 : Pin assign

P#	Name	Description
P1	GND	Ground
P2	VDD	Voltage supply
P3	SCL	Serial clock
P4	SDA	Serial data input/output
P5	NC	-

Figure 35_HSPPAD143A pin layout

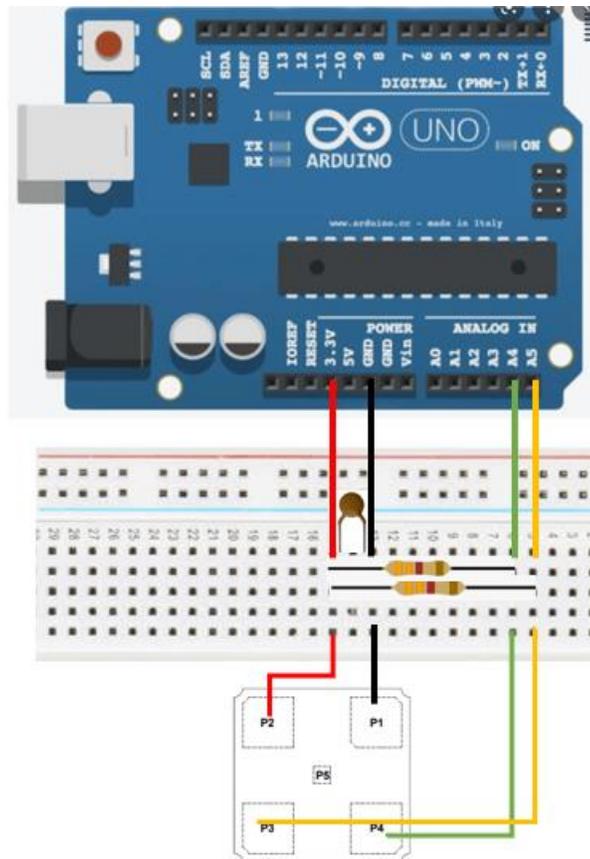


Figure 36_Schematic of hardware setup for HSPPAD143A sensor

The sensor testing is performed running the code on the board and pressure measurements are displayed on the serial monitor. The reservoir is pressurized with different pressure values to check the overall functioning of the sensing system. The set up during testing is shown in figures below.

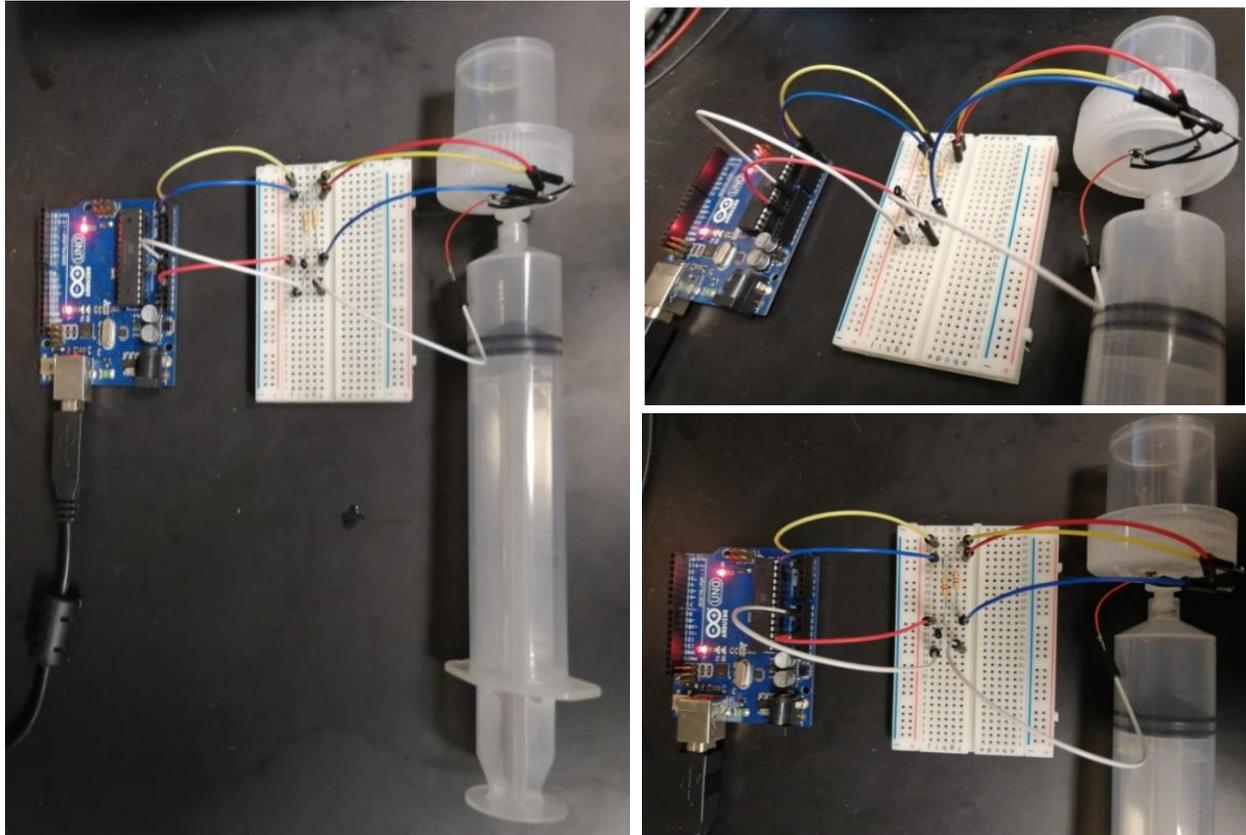


Figure 37_Hardware setup for HSPAD143A testing

4.1.2.2 HSPAD143A sensor: C++ coding

To acquire data using the CC2640R2 microcontroller mounted on the LAUNCHXL it is necessary to develop a C++ code. The general structure of the code is the same as the Arduino one. During this second testing both pressure and temperature readings are displayed on the serial monitor. The script of the C++ code is reported in Appendix. The hardware set up is similar to the set up previously described for the sensor testing with the Arduino board. The LAUNCHXL-CC2640R2 board is connected to a breadboard on which is mounted the electrical circuit with two 3.3kOhm resistors placed between the VDD and the SCL and SDA lines and a 100nF by-pass capacitor placed between VDD and GND. The testing is realized with the same methodology used for the

previous Arduino test: a syringe is used to adjust the pressure and verify the correctness of the measurements obtained with the sensor. Moreover, in this second test, a thermostat is used to verify also temperature values. The figures below show the hardware set up during the sensor testing.

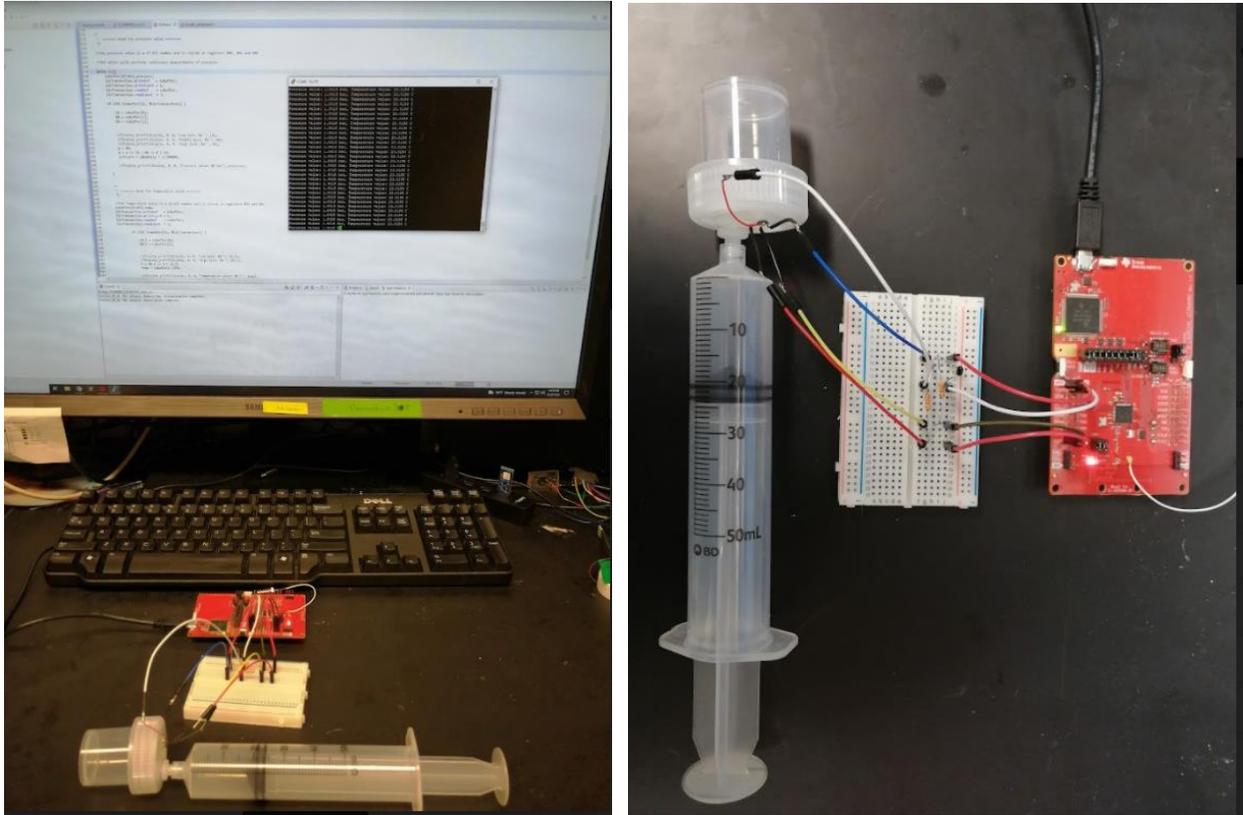


Figure 38_Hardware setup for testing with LAUNCHXL board

4.1.2.3 MS5837-30BA sensor: C++ coding

In order to acquire data using the CC2640R2 microcontroller it is necessary to develop a C++ code specific for the MS5837-30BA digital sensor following the algorithm reported in the datasheet. The first step is the identification of the slave address for the sensor to start the I2C communication protocol. The MS5837-30BA slave address in binary and hexadecimal numbering are reported in the table below.

Function	Binary	Hexadecimal
Read/Write	1110110	0x76

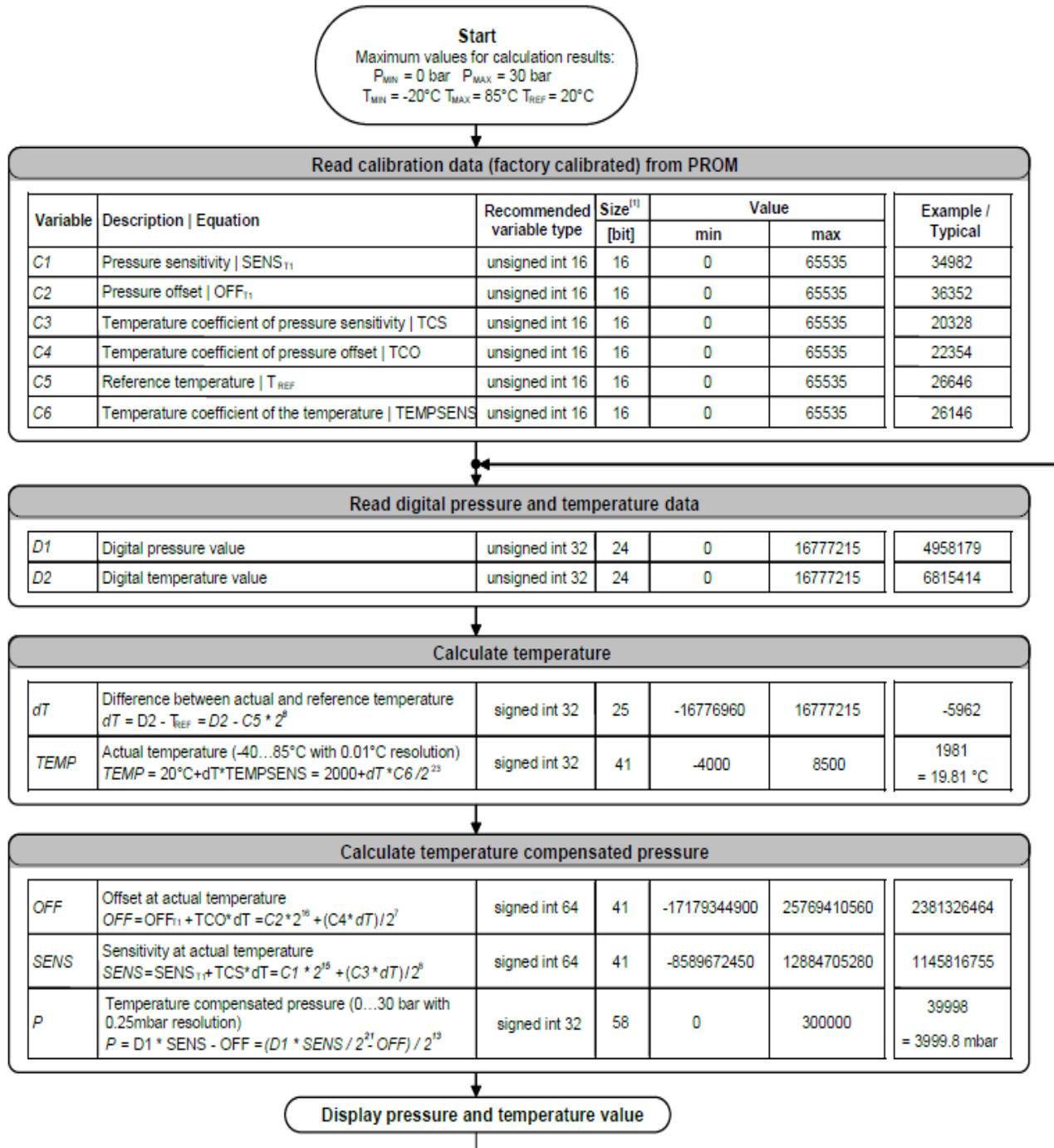
Table 11_MS5837-30BA slave address

The MS5837-30BA has five basic commands:

- reset;
- read PROM (128 bit of calibration words);
- D1 conversion;
- D2 conversion;
- read ADC result (24-bit pressure/temperature).

After ADC read commands, the device returns 24-bit result and after the PROM read 16-bit result. The read command for PROM shall be executed once after reset by the user to read the content of the calibration PROM and to calculate the calibration coefficients. There are in total 7 addresses resulting in a total memory of 112 bit. Addresses contain factory data and the setup, calibration coefficients, the serial code and CRC. The command sequence is 8 bits long with a 16-bit result which is clocked with the MSB first. The PROM Read command consists of two parts. First command sets up the system into PROM read mode. The second part gets the data from the system. The conversion command is used to initiate uncompensated pressure (D1) or uncompensated temperature (D2) conversion. After the conversion, using ADC read command the result is clocked out with the MSB first. When conversion is finished the data can be accessed by sending a Read command [20]. After reading the 24-bit digital value for pressure and temperature, some further computations are necessary to get the pressure value in bar and the temperature value in Celsius degree. A second order temperature compensation is implemented to obtain the optimum accuracy. The complete C++ code is reported in Appendix. In the figures below the algorithm from the sensor datasheet is illustrated.

PRESSURE AND TEMPERATURE CALCULATION



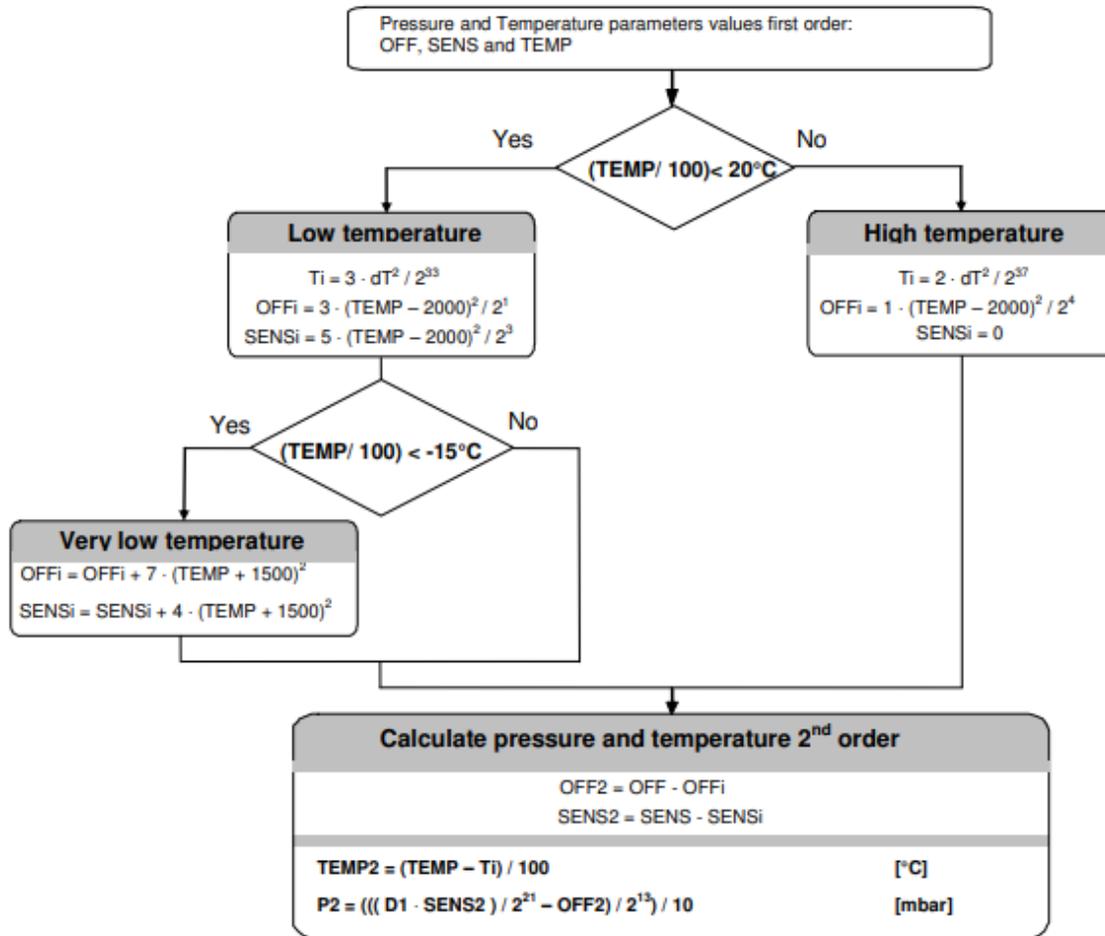


Figure 39_MS5837-30BA algorithm

To properly connect the sensor to the host controller, an electric circuit is implemented using two 10kOhm resistors and a 100nF capacitor. The detailed schematic of the circuit from the datasheet is shown in figure below.

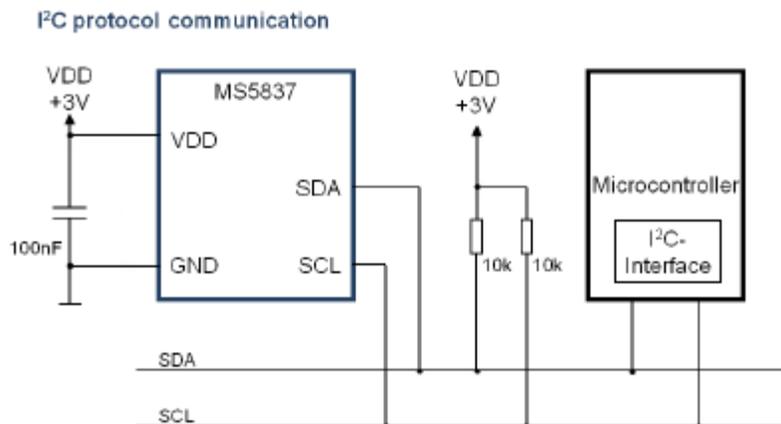


Figure 40_MS5837-30BA circuit

The circuit is mounted on a breadboard which is wired both to the sensor and to the LAUNCHXL board. The sensor is provided with a 3.3V power supply using the board. To realize the I2C communication we use DIO4 and DIO5 for SCL and SDA respectively and the 3.3V and the GND pins. To connect the sensor with the breadboard, 4 electrical wires are welded to the 4 pins of the sensor using tin as welding material. Each pin of the sensor is devoted to a specific function, the pin layout from the data sheet is shown in figure below.

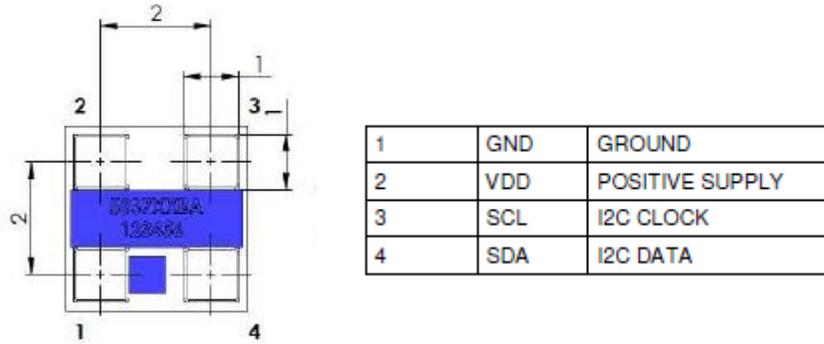


Figure 41_MS5837-30BA pin layout

A schematic of the connections between the board, the breadboard and the sensor is reported in the following figure.

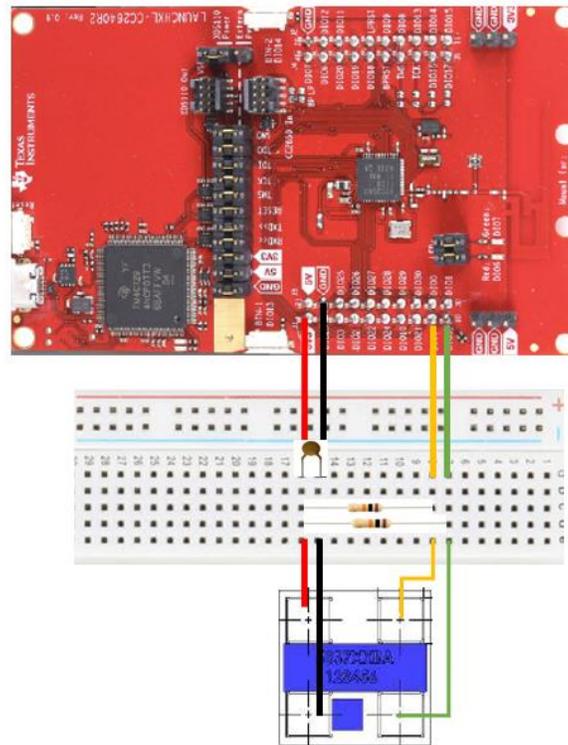
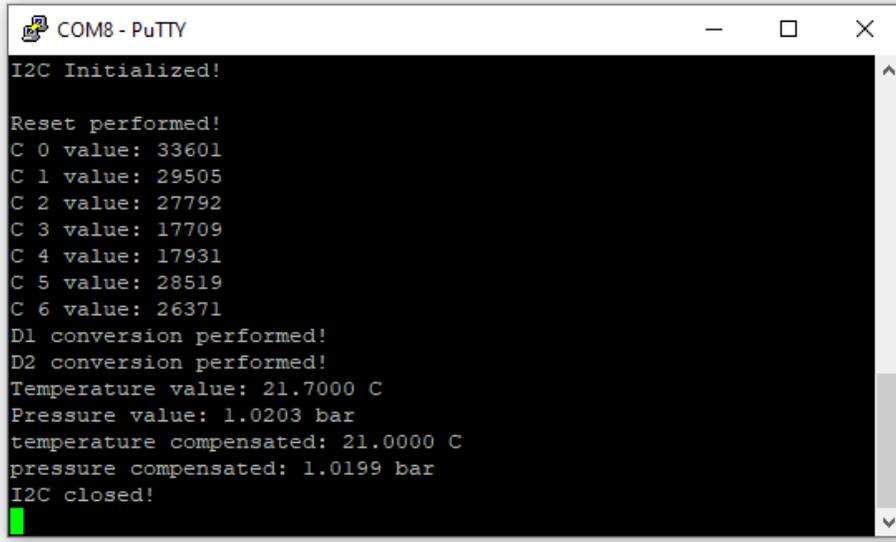


Figure 42_Schematic of hardware setup for MS5837-30BA sensor

The sensor testing is performed with the same procedure used for the other sensor. The C++ code is run on the board using Code Composer Studio and the output is displayed on the serial monitor, as shown in figure below.



```
COM8 - PuTTY
I2C Initialized!

Reset performed!
C 0 value: 33601
C 1 value: 29505
C 2 value: 27792
C 3 value: 17709
C 4 value: 17931
C 5 value: 28519
C 6 value: 26371
D1 conversion performed!
D2 conversion performed!
Temperature value: 21.7000 C
Pressure value: 1.0203 bar
temperature compensated: 21.0000 C
pressure compensated: 1.0199 bar
I2C closed!
```

Figure 43_ Output during sensor testing

4.1.3 Results and discussion

The code testing is performed with the same methodology for the two sensors: the pressure inside the reservoir is increased and decreased using the syringe and pressure readings from the sensors are displayed through the serial monitor. Moreover, a thermostat is used to verify the correctness of temperature readings. The aim of the activity is to develop the code for data acquisition from the sensor through I2C protocol and verify the consistency of pressure and temperature readings. Furthermore, it is possible to test the hardware set up with the electrical circuit and the wiring between the master and the slave device. Results obtained demonstrate the efficacy of the Arduino and C++ codes in acquiring pressure and temperature data, the algorithm and the hardware configuration are therefore validated. The second goal of the testing is to evaluate the optimal sensor solution for our application. The HSPPAD143A sensor results to be more precise in temperature measurement but it is able to detect a limited operating pressure range from 0.3 to 2.1 bar; on the other hand, the MS5837-30BA sensor shows less precise results regarding temperature but allows to obtain pressure measurements in a wide operating range from 0 to 30 bar. The main goal of this project is to study the osmotic pressure behavior; therefore the main interest is the

detection of the widest range of pressure possible and the precision of temperature readings is of secondary importance. For this reason, the optimal choice for the development of the implantable device results to be the MS5837-30BA absolute pressure and temperature digital sensor.

4.2 Bluetooth remote communication

The second challenge in the design of the device is the development of the Bluetooth low energy (BLE) communication in order to read pressure and temperature values with a wireless transmission. The microcontroller on the Printed Circuit Board is programmed with a C++ code, specifically developed to allow for remote data acquisition from the MS5837-30BA sensor using the I2C protocol. The main advantage of the BLE is the possibility to achieve remote data transmission with ultra-low power consumption: this characteristic is essential for future *in vivo* experimentation through implantation of the device in small animals. Data acquired from the sensor are saved in the scan response data. A BLE scan response is a packet that is sent by the advertising device upon the reception of scanning requests. The scan response usually has more data than the advertising packets. The central device sends scan requests to the advertising device in order to get additional user data through the scan response. The code has been built to manage also the case when the sensor is not connected: in this situation temperature and pressure readings display zero values. In the following figure is described the scan response data structure and output values for the sensor not connected case.

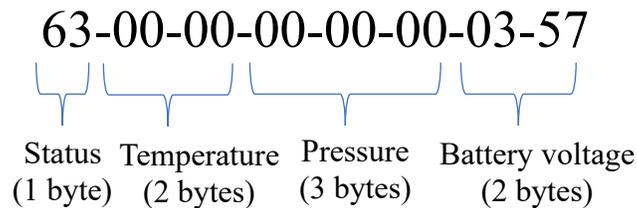


Figure 44_Scan Response Data structure

The Scan Response Data consists of the following parts:

- the status value is an additional variable that can be used in future applications to control the voltage applied to a particular type of nanofluidic membrane used to achieve tunable drug release;

- the temperature value corresponds to the temperature measured by the sensor;
- the pressure value corresponds to the pressure measured by the sensor;
- the battery voltage is used to control the charge level of the battery.

The values are displayed in hexadecimal system and can be read using a specific application called nRF Connect that is available both in the computer and in the mobile phone version. The following figure shows how the application interface looks like.

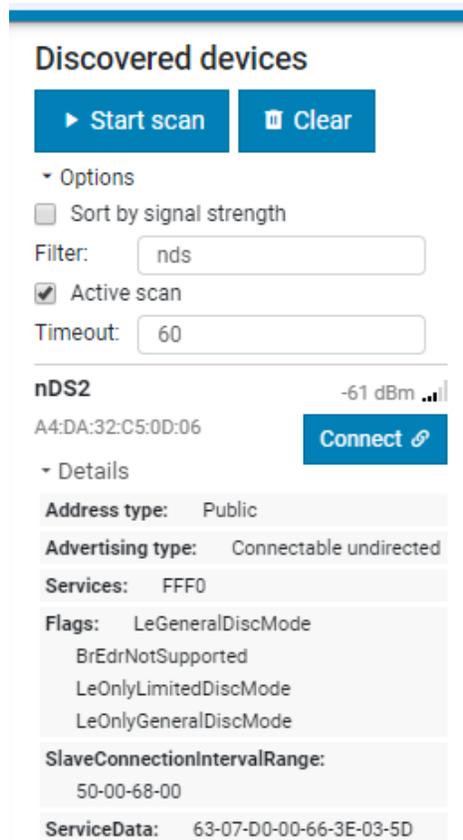


Figure 45_Example of data acquisition through BLE

In the example above data acquired are the following:

- Pressure: 00-66-3E (hex) → 26174 (dec) corresponding to 2,6174 bar
- Temperature: 07-D0 (hex) → 2000 (dec) corresponding to 20,00 °C

The wireless transmission based on the BLE communication protocol has been tested and proved to be reliable and stable and to allow continuous data acquisition.

4.3 Design and assembly procedure

This section illustrates the design and fabrication of the device, describing all its components and the overall assembly procedure. The body of the device is modeled using a CAD software and is then fabricated using the stereolithography (SLA) process. The body hosts the PCB, whose microcontroller is wired with the digital pressure and temperature sensor. The sensor detects the pressure inside a reservoir loaded with drug solution, a lid is used to seal the reservoir and allocate the membrane that allows for osmotic exchange.

4.3.1 Design and fabrication

The first step of the design activity is the modeling of the body structure to allocate the different elements of the device. The capsule is developed using Solidworks and then fabricated through stereolithography (SLA) 3D printing process using an implantable grade polymeric resin. The capsule is made of two chambers: one for the electronics and the other one for the drug reservoir. On the separation wall between the two compartments a U-shaped opening is modeled to allow the insertion of the sensor. To achieve pressure and temperature measurement, one end of the sensor is positioned inside the reservoir while the sensor board is located in the compartment devoted to electronics to allow wiring between the sensor pins and the PCB. The bottom part of the device hosts the battery and small holes are realized to allow the passage of the two wires to supply the PCB. The reservoir is sealed with a lid made of implantable grade polymeric resin that hosts the osmotic membrane. The lid presents some holes with a diameter of 0.8 mm to allow for fluidic exchange across the semipermeable membrane. A hole is located on one of the reservoir walls for reloading of the device. Table 12 summarizes the dimensions of the device, while the following figures show the Solidworks model of the device body and of the reservoir lid.

Device dimension	Value
Width	22 mm
Height	22 mm
Thickness	7 mm

Table 12_Device dimensions



Figure 46_Device drawings (SolidWorks)

4.3.2 Components and assembly procedure

The device is composed of the following elements:

- printed circuit board (PCB);
- discoidal battery;
- reservoir;
- osmotic membrane;
- pressure and temperature sensor.

In the figure below the device assembly is illustrated. The body (a) features two chambers on the top part: the drug reservoir and the electronics chamber. The electronics chamber hosts the digital pressure sensor (b) wired to the PCB (c) which contains the microcontroller and the Bluetooth Low Energy antenna. The bottom part of the body hosts an implantable discoidal battery (d). The semipermeable membrane (e) is embedded in the lid (f) sealed on the top of the reservoir. The hole on one of the reservoir walls is used to load the implant and is sealed with a silicone cork and UV resin after reloading. Implantable grade epoxy has been used to completely embed battery and electronics.

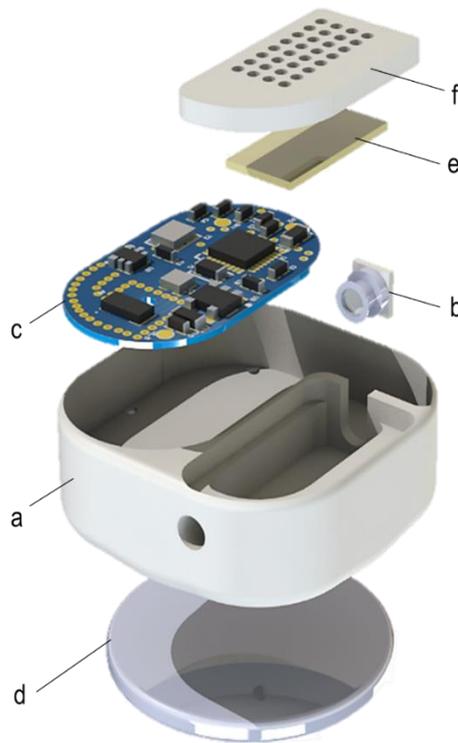


Figure 47 _Device assembly (SolidWorks rendering)

After the 3D printing of the implant body, it is possible to proceed with the assemble of the device. The steps for device assembly procedure are listed below.

- Welding the electrical wires on the PCB: using the microscope and the tin welder two wires are welded at the VDD pin of the PCB (one for the battery the other for the sensor supply), other two wires are connected to the GND pin of the PCB (one for the battery the other for the sensor pin), one wire soldered to the DIO_4 pin of the microcontroller, one wire soldered to the DIO_8 pin of the microcontroller. When the welding has been

performed, a drop of resin glue is positioned on all the welds to avoid unsoldering. The resin is cured with UV light. In figure 48b is shown the PCB after the electrical wires welding.

- Welding of the electrical circuit on the sensor: mount the 100nF capacitor and the two 10kOhm resistors on the sensor board according to the circuit described in the sensor datasheet and weld them to the sensor pins (figure 48c).
- Welding between the PCB wires and the pins of the sensor: under the microscope the PCB wires are welded with the sensor pins in the following way: DIO_4 with the SDA sensor pin, DIO_8 with SCL sensor pin, VDD and GND with their respective pins of the sensor. Figure 48a shows the set up during the welding procedure to connect the sensor with the PCB.
- Sealing of the sensor with the UV resin: the resin is positioned on the U opening of the 3D printed structure of the device, the sensor is inserted in the U opening and cured with UV light while pressing the sensor inside the case. The opening on the top of the sensor is closed with multiple layers of resins until complete sealing (figure 48d).
- Sealing of the two holes for the battery wires: the two holes for the battery wires are sealed using UV resin.
- Embedding of the PCB and the sensor using epoxy resin 302-3M. This is a bicomponent resin that is prepared mixing together the part A and part B components in the proportion of 145:100. The two components are mixed using a stick until a homogeneous color is achieved. The cuvette with the resin is positioned in the stirring machine at 4000 rpm for 2 minutes to eliminate air bubbles. A small amount of epoxy resin is, then, inserted into the PCB compartment using a pipette in order to cover all the bottom of the compartment. The following step consists in the stirring of the implant that is positioned in the stirring machine at a speed of 800-1000 rpm for 2 minutes to realize homogenous diffusion of the resin on the bottom of the compartment. Finally, the electronics cavity is completely filled with epoxy resin to achieve the embedding of the PCB and the sensor. In the end, the resin is cured in a container at 65 °C for 3 hours.
- Embedding of the battery: the same procedure previously explained for the electronics chamber embedding with curable bicomponent epoxy resin is used to achieve embedding of the discoidal battery positioned in the compartment on the bottom part of the device.

- Sealing of the lid on the top of the reservoir. The membrane is embedded in the lid by deposition of a layer of silicone that is cured at room temperature for 18 hours and then cured in the oven at 70°C for 2 hours. Lastly, the lid is positioned and sealed using UV resin. The final device obtained after the assembly procedure is shown in figure 48e.

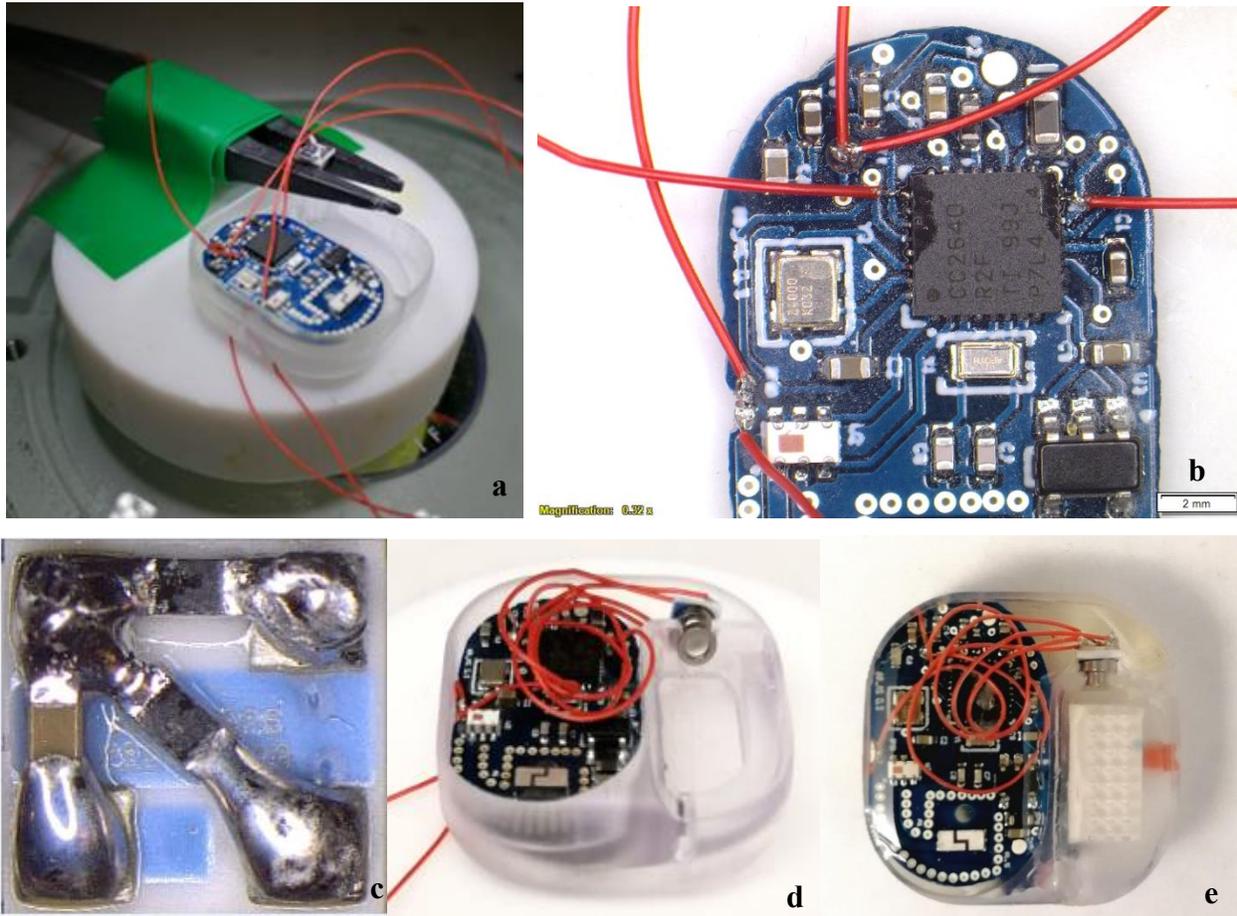


Figure 48_Device assembly procedure

5 Experimental validation of implantable device with digital pressure sensor

The purpose of this section is to describe the testing methodology for the experimental validation of the device and discuss the results obtained. The first subchapter illustrates the setup of the test bench and the testing methodology; the aim of the test bench is to pressurize the reservoir, providing an adjustable value of internal pressure. An external pressure sensor is used as a reference to validate the pressure readings obtained with the remotely controlled implantable device. The second subchapter illustrates the results with discussion and conclusions. The main objectives of the experimental activity can be resumed as follows:

- evaluate the accuracy of the pressure readings obtained with the digital sensor embedded in the device through the comparison with a reference pressure sensor;
- prove the stability of the wireless data transmission via BLE protocol;
- check the sealing of the device with the increase of pressure value and verify the absence of leakages;
- demonstrate the reliability and applicability of the device in performing continuous measure of pressure inside the reservoir.

5.1 Test bench and testing methodology

The goal of the testing activity is to pressurize the reservoir with different pressure values and compare pressure measurements obtained with the device to pressure values read with the reference external pressure sensor in order to validate the implant. The reference sensor used is a pressure transducer by Ashcroft with the following specifications:

- Operating pressure range (absolute): 0-60 psi
- Accuracy: 1.0 %
- Input: 10-36 Vdc
- Output: 1-5 Vdc
- Maximum pressure (absolute): 120 psi



Figure 49_Reference pressure transducer

An Arduino board and a Matlab script, reported in Appendix, are used in order to continuously acquire data from the reference pressure transducer. The pressure transducer is connected to the reservoir inside the device via a fluidic fitting. The reservoir is closed with a lid to allow for pressurization of the chamber through a syringe that provide an adjustable pressure value; a mechanical system is used to set the pressure value and keep it fixed. The syringe is also in communication with the reservoir thanks to another fitting inserted in one of the chamber walls. The device is supplied with 3.3V using the LAUNCHXL board; the board is also used to acquire data via the BLE communication using a customized Matlab script, reported in Appendix. The following figure shows the set up during the experimental validation.

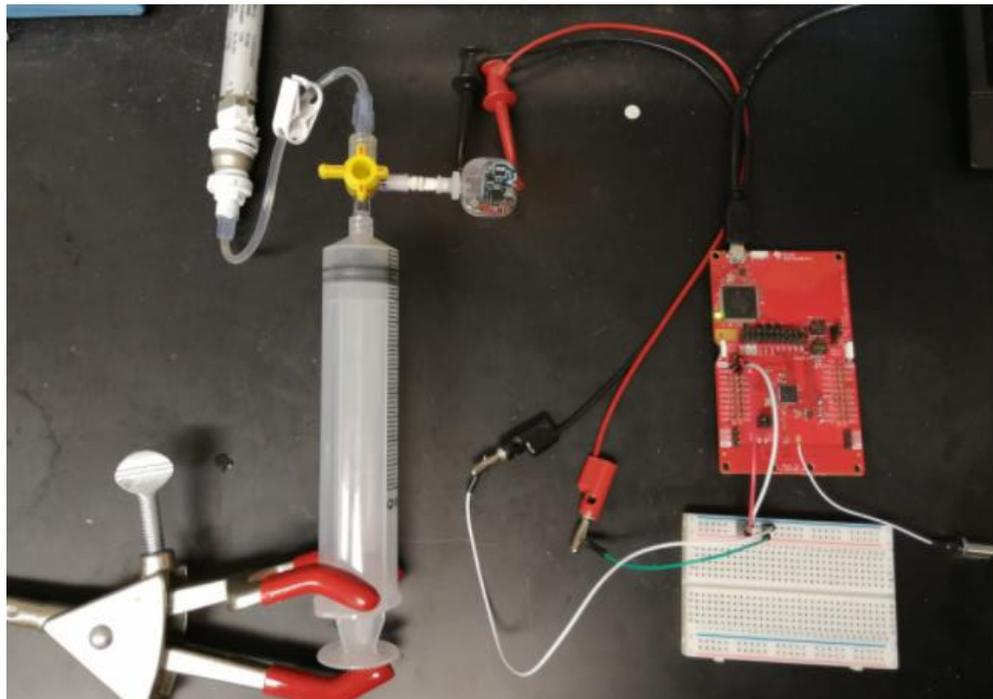


Figure 50 Experimental setup

5.2 Results and discussion

This section describes and discusses the results obtained from the experimental testing of the device. The two Matlab codes developed for data acquisition from the reference sensor and from the device provide the behavior of the pressure as a function of the time. The results are analyzed and compared to demonstrate the accuracy of the pressure readings and validate the new remotely controlled device. The pressure is gradually increased starting from atmospheric pressure and kept constant for the necessary amount of time to detect at least two pressure readings from the device. The validation testing is firstly performed with pressurized air and then with pressurized water to simulate the real working conditions with the reservoir loaded with aqueous drug solution. The figures below illustrate the results obtained from experimental validation of the device with pressurized air and pressurized water respectively.

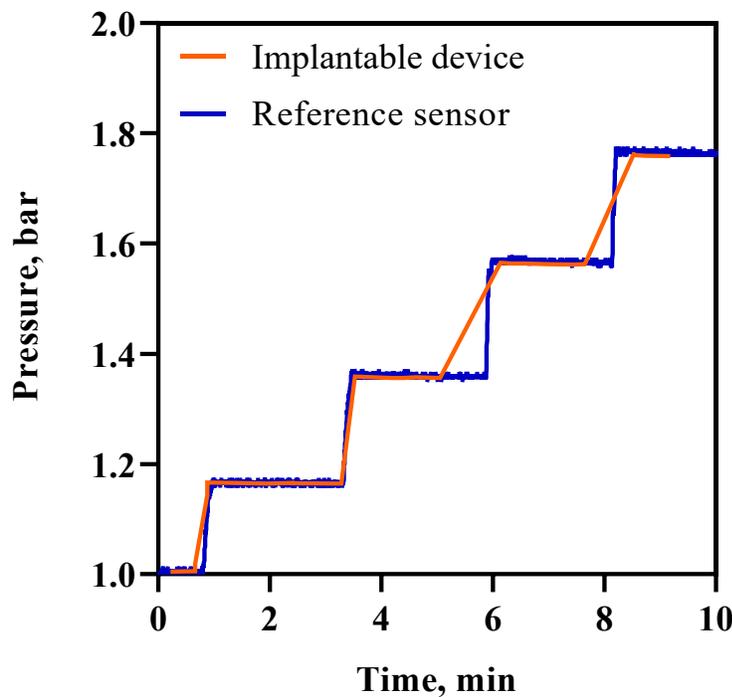


Figure 51 Experimental results for pressurized air test

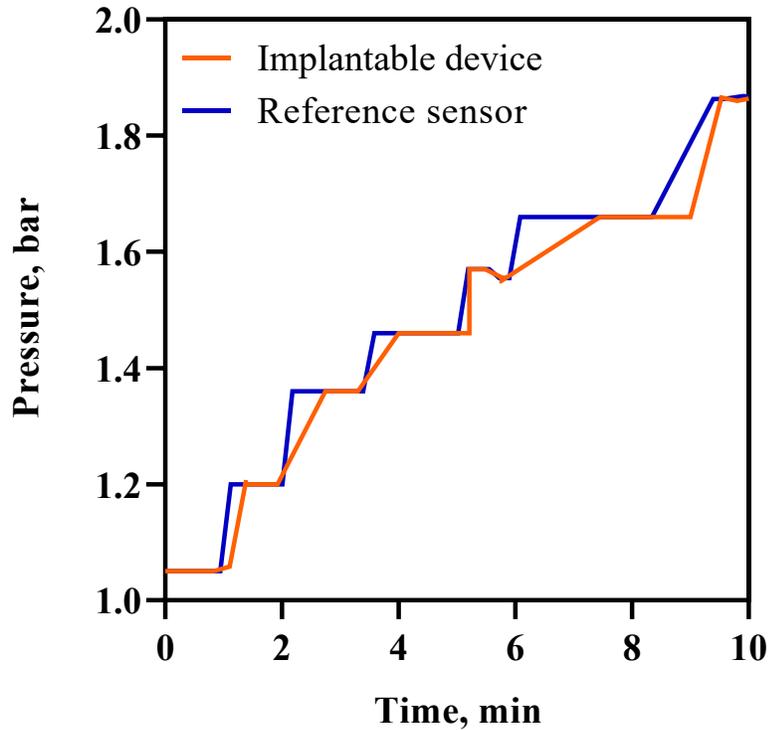


Figure 52_ Experimental results for pressurized water test

The small difference between the behaviors can be explained with the following consideration: the reference sensor acquires data in a continuous way, the BLE remote communication transmits data according to a wider time interval that is not exactly constant but may vary a little bit. This explains the discrepancy between the plots. For the purpose of our application the important aspect is the precision of pressure readings which is confirmed by the experimental results. In conclusion, the experimental testing activity demonstrates the accuracy and correctness of pressure readings obtained from the digital pressure sensor and validate the device for performing continuous pressure measurement. The pressurized water test is also used to check the sealing of the device and to confirm the absence of leakage. The device has been proved to reach a pressure of 6 bar without leakages.

6 In vitro experimentation

This chapter focuses on *in vitro* experimentation performed with the remotely controlled implantable device. A reverse osmosis semipermeable membrane is embedded in the device to allow for osmotic exchange. The first section describes the membrane testing procedure performed to achieve membrane fluidic characterization and to acquire preliminary information about osmotic pressure behaviour. The next subchapter focuses on preliminary experiments aiming at proving the stability of the Bluetooth remote communication and the accuracy of pressure measurement. After demonstrating accuracy and stability of the device, *in vitro* experimentation is conducted to investigate the osmotic pressure inside the reservoir as a function of different solutions with different molar concentrations. Therefore, the third subchapter is an overview of the methodology and the experimental design used for conducting the experiments. In the end, the last section describes the results obtained and the pressure data acquired reloading the device with aqueous solutions of glucose, sucrose, glycine and glycerol.

6.1 Reverse osmosis membrane testing

In order to investigate the pressure behaviour due to osmotic exchange across the reverse osmosis membrane, a specific testing procedure is developed and *in vitro* experiments are performed to test the membrane and achieve membrane fluidic characterization. The testing apparatus is designed in the following way: the RO membrane is inserted between two cylindrical components screwed together to provide sealing. A porous disc is used to provide support to the membrane and to avoid membrane deformation that can lead to variation in the chamber volume and consequently affect the internal pressure. The upper chamber is loaded with the solution and a pressure transducer is screwed inside the chamber to perform pressure measurement. The other side of the membrane is in contact with deionized water at atmospheric pressure. A Matlab code is run on a nearby computer to acquire pressure data continuously from the transducer that is connected to a multimeter. The membrane is prepared with a priming procedure that involves 20 minutes in ethanol and 30 minutes under vacuum. Experiments are performed with three glycine solutions with molar concentration of 0.1, 0.2 and 0.3 respectively to investigate the osmotic pressure as a function of the molar concentration. Each experiment lasts 24 hours and each solution is tested

twice to evaluate the repeatability of the procedure. A schematic of experimental apparatus is illustrated in figure 53.

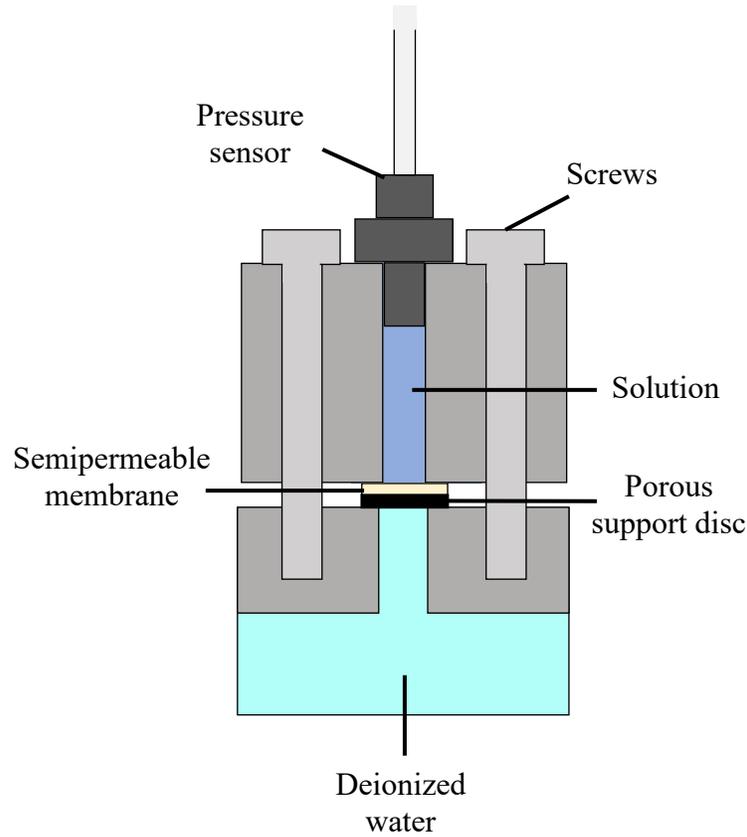


Figure 53_Experimental apparatus

The osmotic pressure of an ideal solution is described by the Van't Hoff equation as follows:

$$\pi = iMRT \quad (22)$$

where

π is the osmotic pressure [atm]

i is the van't Hoff factor [dimensionless]

M is the molar concentration of the solution [mol/L]

R is the gas constant [L atm mol⁻¹K⁻¹]

T is the absolute temperature [K]

Considering equation 22, the osmotic pressure results to be directly proportional to the molar concentration. The experimental results of membrane testing show, as expected, that the pressure inside the chamber increases due to the osmotic process and its increase is proportional to the

molar concentration of the solution. The pressure data acquired with membrane testing provide important information about the membrane characterization and the expected osmotic pressure curves inside the device. The following figure illustrates the osmotic pressure readings from the pressure transducer inside the solution chamber.

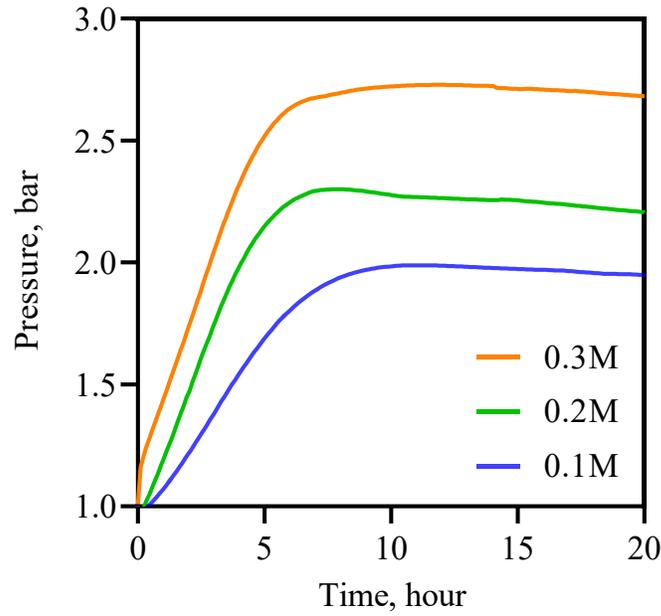


Figure 54_RO membrane testing results

6.2 Stability of the remote communication

Preliminary *in vitro* experimentation is performed in order to test the continuous pressure data transmission and the stability of the Bluetooth remote communication. The test is performed reloading the reservoir with a 1 molar glucose solution and lasted 20 hours. Furthermore, during the experiments, the RSSI index, that stands for Received Signal Strength Indicator, is recorded and monitored. This index provides an indication of strength and performance of a wireless connection. The values of the RSSI index swings in the range from -55 to -75 Decibel; these values prove the stability of the Bluetooth wireless transmission. The results obtained for the pressure inside the reservoir and for the RSSI index are shown in the figure below.

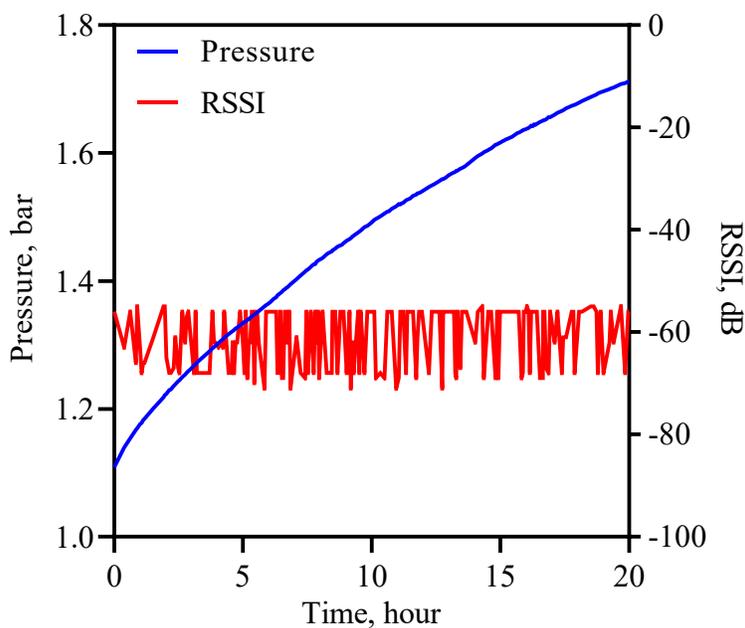


Figure 55_ Stability testing results

6.3 *In vitro* methodology

In vitro experimentation is performed using the remotely controlled implantable device to investigate the osmotic pressure inside the drug reservoir as a function of different solutes and molar concentrations. Experiments are conducted with different aqueous solutions of glucose, glycerol, sucrose and glycine. After reloading of the reservoir with the solution, the device is immersed in deionized water and a magnetic stirrer provides fluid constant homogenization to facilitate solvent exchange across the semipermeable membrane. The osmotic process involves the net movement of solvent from pure solvent solution, deionized water in the beaker, to a high concentrated one, which is the solution inside the reservoir. Pressure readings from the digital sensor are wirelessly transmitted and recorded to a nearby computer thanks to the Bluetooth remote communication. *In vitro* experiments are conducted using the reverse osmosis semipermeable membrane, previously tested and characterized. Several tests have been done to identify the proper membrane priming procedure. An issue that can affect the pressure behavior is the presence of air bubbles inside the reservoir; to address this problem the solution and the device itself have been positioned under vacuum. Another factor that directly affect the osmotic pressure is the temperature; for this reason, *in vitro* experiments have been performed inside the incubator at a constant temperature of 37 Celsius degrees. The duration of the experiments varies from hours to

several days; depending on the solution and on the concentration the osmotic process may require a different amount of time. A schematic of the methodology for *in vitro* experimentation and data acquisition is illustrated in the figure below.

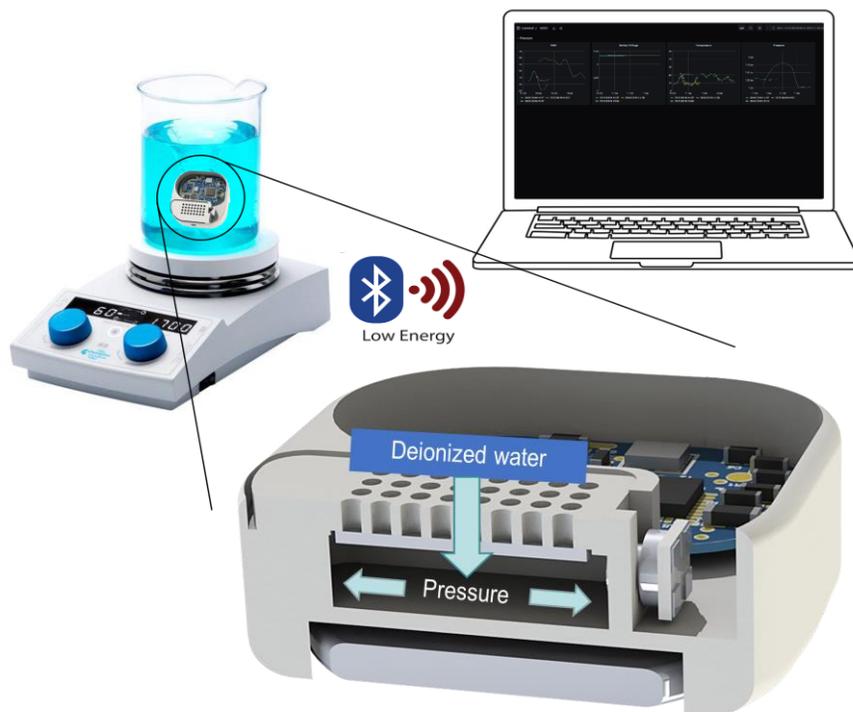


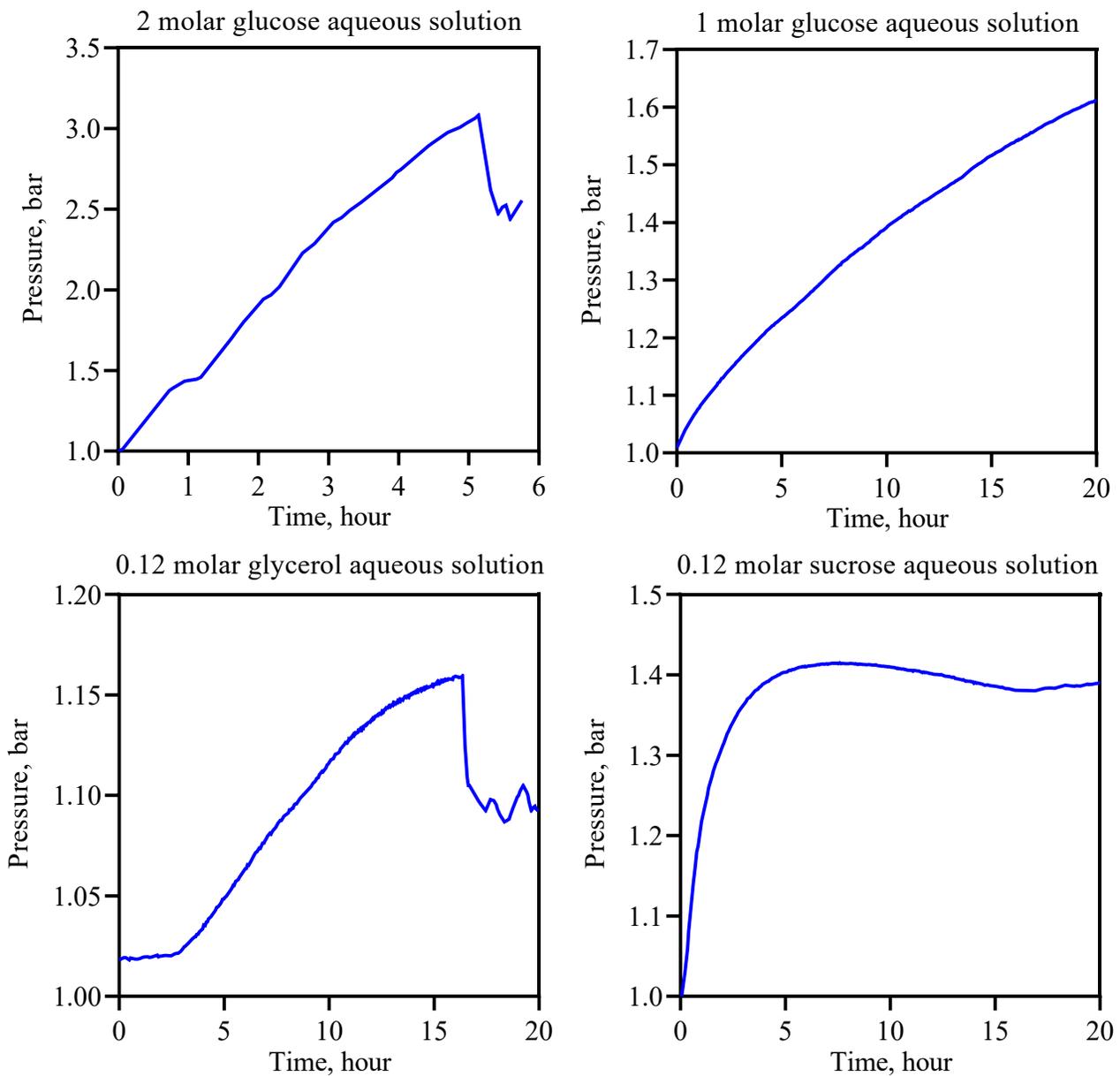
Figure 56 *In vitro* experimentation setup and methodology

6.4 Results and discussion

This section focuses on the presentation and discussion of the osmotic pressure data acquired through *in vitro* experimentation using the remotely controlled device. The tests are performed with the following solutions:

- glucose aqueous solution with molar concentration of 1 mol/L
- glucose aqueous solution with molar concentration of 2 mol/L
- sucrose aqueous solution with molar concentration of 0.12 mol/L
- glycerol aqueous solution with molar concentration of 0.12 mol/L
- glycine aqueous solution with molar concentration of 0.12 mol/L

The pressure inside the device reservoir is acquired continuously thanks to the wireless communication using a Python script running on a computer. Data analysis leads to obtain the pressure inside the device reservoir as a function of the time. The general osmotic pressure behavior is characterized by an initial pressure increase; after reaching the maxim value, the pressure shows an almost stable behavior or a small decrease. In the experiments performed with the 2-molar glucose solution and with the 0.12-molar glycerol solution it is possible to notice a sharp pressure decrease at a certain time; the pressure drop is probably caused by a leakage in the device. The osmotic pressure curves obtained with the *in vitro* experimentation with five different solutions are shown in the following figures.



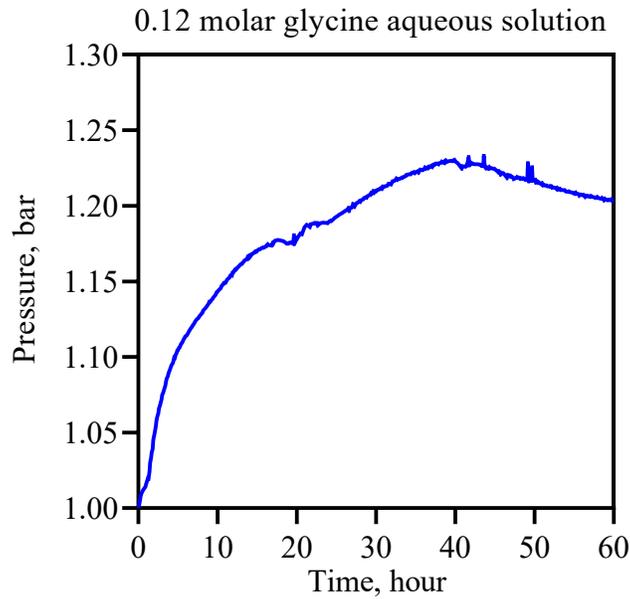


Figure 57 *In vitro* results for glucose, glycerol, sucrose and glycine solutions

The pressure increase results to be proportional to the molar concentration, as stated in equation 22. It is possible to notice that the slope of the pressure curve for the 2-molar glucose solution is way steeper compared to the 1 molar glucose solution, the osmotic pressure increase is faster due to the higher molar concentration. A difference of the osmotic pressure behavior according to the different solute has also been observed. The comparison of pressure characteristics as a function of the time for the five different solutions is reported in the following figure.

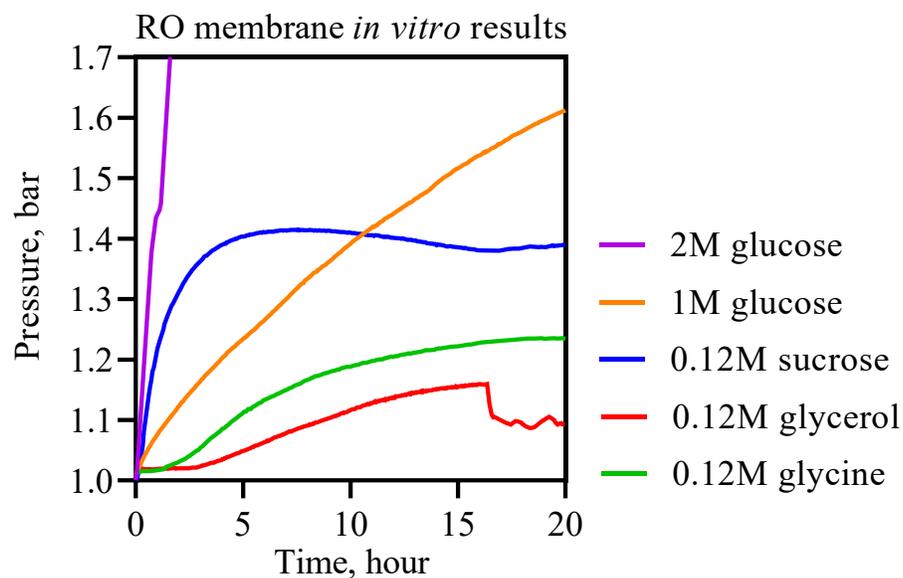


Figure 58 Comparison of *in vitro* results

In order to prove repeatability of the osmotic pressure characteristic, extended in vitro experiments with glycine are performed. The results obtained demonstrate that the three tests show a similar behavior; the pressure curves as a function of the time are reported in the figure below.

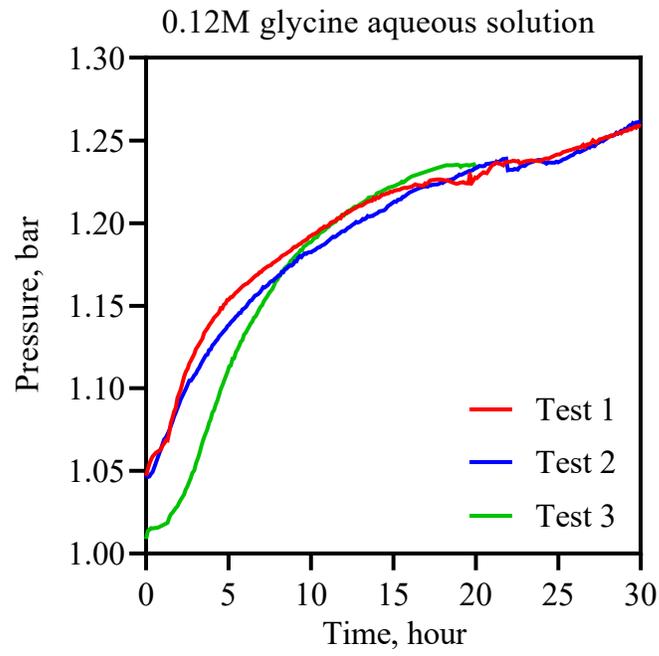


Figure 59_In vitro results with glycine solution

Conclusions and Future Directions

The main achievement of this thesis is the design, development and characterization of two implantable devices for direct and indirect remote measurement of osmotic pressure in long-acting drug delivery systems. The results obtained on metal and polymeric capsule prototypes equipped with strain sensors demonstrate the applicability to passive drug delivery systems. The resolution of the sensing method is comparable to the target resolution of osmotic pressure measurement, after filtering and decoupling the external disturbances. The materials used show peculiar advantages: higher dimensional accuracy and mechanical strength for titanium, low cost and fast processing for polyamide. The best sealing is obtained with titanium, the higher sensing sensitivity is provided by the flexibility of polyamide [10]. *In vitro* experimentation leads to the characterization of the drug reservoir in terms of strain-pressure relationship; moreover, extended tests confirm pressure stability over time and repeatability of the measurement chain. Additionally, the comparison with theoretical characteristics proves the accuracy of pressure values and validate the overall data acquisition strategy. The goal of this research is to accomplish the design and fabrication of multiple sensing solutions in order to provide implantable drug delivery devices able to perform not only indirect but also direct measurement of the pressure inside the reservoir. With this purpose, a second device prototype is modelled and fabricated based on a digital pressure sensor that results to be more expensive but also easier to be integrated with the microcontroller and the electronic board. Extended tests conducted on the implantable device for direct osmotic pressure measurement demonstrate the accuracy of pressure readings and the stability of the Bluetooth remote communication. A reverse osmosis semipermeable membrane is integrated in the device to allow for osmotic exchange; a membrane testing methodology is developed and proves to be suitable to achieve membrane fluidic characterization. Moreover, this research leads to *in vitro* experimentation conducted loading the device reservoir with aqueous solutions of glucose, sucrose, glycine and glycerol with different molar concentrations. This activity leads to the acquisition of new data about the osmotic pressure behavior inside the reservoir. The main achievement of the second device prototype is the implementation of a Bluetooth wireless remote communication that allows for simple reading and logging of pressure data just using a smartphone or a laptop. This technology is fundamental for future *in vivo* investigation of osmotic pressure through implantation in small animals, where the use of wired sensors is not applicable.

Considering the overall activities conducted in this thesis, future directions include the execution of *in vivo* experimentation through implantation of the device in rats in order to investigate the effect of osmotic phenomena on drug delivery. Additionally, another future step could be the development of a model that simulates the osmotic pressure behavior as a function of the molar concentration using pressure data acquired with *in vitro* experimentation. In conclusion, this research activity led to the publication of few scientific articles and the results obtained will assist further investigation of the effect of osmotic pressure on drug delivery both *in vitro* and *in vivo*. The outcome of this research will hopefully lead to a better understanding of osmotic phenomena, setting the foundation for a new generation of more effective and reliable drug delivery systems for chronic disease treatment.

References

- [1] Alwan A. et al., *Monitoring and surveillance of chronic non-communicable diseases: progress and capacity in high-burden countries*, The Lancet, 2010.
- [2] Park K., *Controlled drug delivery systems: Past forward and future back*, Journal of Controlled Release, 2014.
- [3] Pons-Faudoa F. P., Ballerini A., Sakamoto J. & Grattoni A., *Advanced implantable drug delivery technologies: transforming the clinical landscape of therapeutics for chronic diseases*, Biomed Microdevices, 2019.
- [4] Herrlich S. et al., *Osmotic micropumps for drug delivery*, Adv Drug Deliver, 2012.
- [5] Pons-Faudoa F. P. et al., *Preventive Efficacy of a Tenofovir Alafenamide Fumarate Nanofluidic Implant in SHIV-Challenged Nonhuman Primates*, Advanced Therapeutics, 2020.
- [6] Chua C. Y. X. et al., *Transcutaneously refillable nanofluidic implant achieves sustained level of tenofovir diphosphate for HIV pre-exposure prophylaxis*, Journal of Controlled Release, 2018.
- [7] Di Trani N. et al., *Electrostatically gated nanofluidic membrane for ultra-low power-controlled drug delivery*, Lab on a Chip, 2020.
- [8] Warreng C. Young, Richard G. Budynas, *Roark's Formulas For Stress And Strain*, McGraw-Hill, 7th edition.
- [9] Antonio Gugliotta, *Piastre circolari inflesse assialsimmetriche*, Machine building course, Politecnico di Torino.
- [10] Graziano M., Charig D. W., Di Trani N., Grattoni A. and De Pasquale G., *Design and characterization of AM sensorized capsules for drug delivery devices*, Infinite Science Publishing, AMMM 2021.
- [11] Kruth JP, Froyen L, Van Vaerenbergh J et al., *Selective laser melting of iron-based powder*, J Mater Process Technol, 2004.
- [12] De Pasquale G., Luceri F., Riccio M., *Experimental evaluation of selective laser melting process for optimized lattice structures*, Journal of Process Mechanical Engineering, 2018.
- [13] Wu H. et al., *Recent developments in polymers/polymer nanocomposites for additive manufacturing*, Progress in Material Science, June 2020.

- [14] P.S. Balaji and Karuppasamy Karthik Selva Kumar, *Introduction and Application of Strain Gauges. Applications and Techniques for Experimental Stress Analysis*, IGI Global, 2020.
- [15] John Whitefoot, *Use of Strain Gages to Determine the Strain in Cantilever Beams*, Mechanical Measurements course, October 2018.
- [16] R.B. Watson, *Influence of grid geometry on the output of strain gauge-based diaphragm Pressure Transducers*, 1986.
- [17] Joji P., *Pneumatic Controls*, Wiley-India Edition, 2008.
- [18] Di Trani N. et al., *Remotely controlled nanofluidic implantable platform for tunable drug delivery*, Lab on a Chip, 2019.
- [19] HSPPAD143A digital pressure and temperature sensor datasheet, AlpsAlpine.
https://tech.alpsalpine.com/prod/c/pdf/sensor/piezo/hsppa_wp/hsppad143a_data.pdf
- [20] MS5837-30BA digital pressure and temperature sensor datasheet, TE Connectivity.
<https://www.alldatasheet.com/datasheet-pdf/pdf/881483/TEC/MS5837-30BA.html>
- [21] Blum J, *Exploring Arduino®: Tools and Techniques for Engineering Wizardry*, Second Edition, John Wiley & Sons, 2020.

Appendices

Arduino code for HSPPAD143A sensor

```
#include <Wire.h>
#define addr8 0x48 //Unique bus address from the datasheet

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  Serial.println("Master is ready");
  Serial.println('\n');
}

void reset() //perform software reset
{
  Wire.beginTransmission(addr8);
  Wire.write(0x11); //move pointer to register address
  Wire.write(0x80); //sends 80h (hex) to register address 11h
  Wire.endTransmission();
}

void average() //average measured data of 2 times, AVG=001b
{
  Wire.beginTransmission(addr8);
  Wire.write(0x13); //move pointer to register address;
  Wire.write(0x39); //sends 39h (hex) to register address 13h
  Wire.endTransmission();
}

void mode() //set MODE to continuous mode
{
  Wire.beginTransmission(addr8);
  Wire.write(0x0F); // move pointer to CTL2.MODE address
  Wire.write(0xA1); // sends 0x1 (hex) 01 (binary)
  Wire.endTransmission();
}

void start() // start continuous measurements
{
  Wire.beginTransmission(addr8);
  Wire.write(0x0F);
  Wire.write(0xA5); // write A5h (hex) to register address 0Fh
  Wire.endTransmission();
}

void getdata(uint16_t *a, uint16_t *b, uint16_t *c)
{
  Wire.beginTransmission(addr8);
  Wire.write(4);
  Wire.endTransmission();
}
```

```

        Wire.requestFrom(addr,3); //Sends content of three registers that stores
POUT
    *a = Wire.read(); //first byte
    *b = Wire.read(); //second byte
    *c = Wire.read(); //third byte
}

void showdata()
{
    uint16_t aa,bb,cc;
    uint32_t p;
    double pressure;
    getdata(&aa, &bb, &cc);
    Serial.print("byte 1: "); Serial.println(aa,BIN);
    Serial.print("byte 2: "); Serial.println(bb,BIN);
    Serial.print("byte 3: "); Serial.println(cc,BIN);
    p = cc;
    //Serial.println(p<<16,BIN);
    p = p << 16 | bb << 8 | aa;
    //Serial.println(p,BIN);
    //Serial.println(p,DEC);
    pressure = (double)p*2/100000;
    Serial.print("pressure (bar): ");
    Serial.println(pressure,DEC);
    delay(1000);
}

void loop()
{
    reset();
    delay(500);
    average();
    delay(500);
    mode();
    delay(500);
    start();
    delay(500);
    showdata();
}

```

C++ code for HSPPAD143A sensor

```

#include <stdint.h>
#include <stddef.h>
#include <unistd.h>

/* Driver Header files */
#include <ti/drivers/GPIO.h>
#include <ti/drivers/I2C.h>
#include <ti/display/Display.h>

/* Example/Board Header files */
#include "Board.h"

```

```

#define TASKSIZE          640
/*
 * ===== TMP Registers =====
 */
#define ADDR              0x48;
#define REG_pressure      0x4;
#define REG_temp          0x9;
static Display_Handle display;
/*
 * ===== mainThread =====
 */
void *mainThread(void *arg0)
{
    uint8_t                txBuffer[2];
    uint8_t                rxBuffer[3];
    uint16_t               LB=0;
    uint16_t               MB=0;
    uint16_t               HB=0;
    uint32_t               p=0;
    double                 pressure=0;
    uint16_t               LB_t=0;
    uint16_t               HB_t=0;
    uint32_t               t=0;
    double                 temp=0;

    I2C_Handle             i2c;
    I2C_Params              i2cParams;
    I2C_Transaction         i2cTransaction;

    /* Call driver init functions */
    Display_init();
    GPIO_init();
    I2C_init();

    /* Open the HOST display for output */
    display = Display_open(Display_Type_UART, NULL);
    if (display == NULL) {
        while (1);
    }

    /* Turn on user LED */
    GPIO_write(Board_GPIO_LED0, Board_GPIO_LED_ON);
    Display_printf(display, 0, 0, "Starting the i2ctmp example.");

    /* Create I2C for usage */
    I2C_Params_init(&i2cParams);
    i2cParams.bitRate = I2C_400kHz;
    i2c = I2C_open(Board_I2C_TMP, &i2cParams);
    if (i2c == NULL) {
        Display_printf(display, 0, 0, "Error Initializing I2C\n");
        while (1);
    }
    else {
        Display_printf(display, 0, 0, "I2C Initialized!\n");
    }
    /*
     * ===== write: start continuous measurements =====

```

```

*/

i2cTransaction.writeBuf = txBuffer;
i2cTransaction.writeCount = 1;
i2cTransaction.readBuf = rxBuffer;
i2cTransaction.readCount = 0;

i2cTransaction.slaveAddress = ADDR;
if (!I2C_transfer(i2c, &i2cTransaction)) {
    Display_printf(display, 0, 0, "Unsuccessful I2C transfer");
    while(1);
}

// to start continuous measurements write A5h(165) on register 0Fh
initialized with A0h(160)

txBuffer[0] = 0x0F;//register location
txBuffer[1] = 0xA5;// register value to write
i2cTransaction.writeBuf = txBuffer;
i2cTransaction.writeCount = 2;
i2cTransaction.readBuf = rxBuffer;
i2cTransaction.readCount = 0;

i2cTransaction.slaveAddress = ADDR;
if (!I2C_transfer(i2c, &i2cTransaction)) {
    Display_printf(display, 0, 0, "Unsuccessful I2C transfer");
    while(1);
}

// read the value of register 0Fh
txBuffer[0] = 0x0F;
i2cTransaction.writeBuf = txBuffer;
i2cTransaction.writeCount = 1;
i2cTransaction.readBuf = rxBuffer;
i2cTransaction.readCount = 1;

i2cTransaction.slaveAddress = ADDR;
if (I2C_transfer(i2c, &i2cTransaction)) {
    Display_printf(display, 0, 0, "Reg value: %d ", rxBuffer[0]);
}

/*
* ===== Read the pressure value =====
*/

//the pressure value is a 17-bit number and is stored in registers 04h, 05h and
06h; the while cycle performs continuous pressure measurements

while (1){
    txBuffer[0]=REG_pressure;
    i2cTransaction.writeBuf = txBuffer;
    i2cTransaction.writeCount = 1;
    i2cTransaction.readBuf = rxBuffer;
    i2cTransaction.readCount = 3;

    if (I2C_transfer(i2c, &i2cTransaction)) {

```

```

        LB = rxBuffer[0];
        MB = rxBuffer[1];
        HB = rxBuffer[2];
        p = HB;
        p = p << 16 | MB << 8 | LB;
        pressure = (double)p * 2/100000;
        Display_printf(display, 0, 0, "Pressure value: %f bar", pressure);
    }

/*
 * ===== Read the temperature value =====
 */

//the temperature value is a 16-bit number and is stored in registers 09h and
0Ah
    txBuffer[0]=REG_temp;
    i2cTransaction.writeBuf = txBuffer;
    i2cTransaction.writeCount = 1;
    i2cTransaction.readBuf = rxBuffer;
    i2cTransaction.readCount = 2;

    if (I2C_transfer(i2c, &i2cTransaction)) {
        LB_t = rxBuffer[0];
        HB_t = rxBuffer[1];
        t = HB_t << 8 | LB_t;
        temp = (double)t /256;
        Display_printf(display,0,0,"Temperature value:%f C",temp);
    }

}

    I2C_close(i2c);
    Display_printf(display, 0, 0, "I2C closed!");
    return (NULL);
}

```

C++ code for MS5837-30BA sensor

```

#include <stdint.h>
#include <stddef.h>
#include <unistd.h>

/* Driver Header files*/
#include <ti/drivers/GPIO.h>
#include <ti/drivers/I2C.h>
#include <ti/display/Display.h>

/* Example/Board Header files*/
#include "Board.h"
#define TASKSTACKSIZE 640

/*
 * ===== TMP Registers =====
 */
#define ADDR 0x76;

```

```

static Display_Handle display;
/*
 * ===== mainThread =====
 */
void *mainThread(void *arg0)
{
    uint8_t      txBuffer[2];
    uint8_t      rxBuffer[3];
    uint16_t     LB=0;
    uint16_t     MB=0;
    uint16_t     HB=0;
    uint16_t     C[7];
    uint32_t     D1=0;
    uint32_t     D2=0;

    I2C_Handle   i2c;
    I2C_Params   i2cParams;
    I2C_Transaction i2cTransaction;

    /* Call driver init functions */
    Display_init();
    //GPIO_init();
    I2C_init();

    /* Open the HOST display for output */
    display = Display_open(Display_Type_UART, NULL);
    if (display == NULL) {
        while (1);
    }

    //Initialize the i2c transmission
    I2C_Params_init(&i2cParams);
    i2cParams.bitRate = I2C_400kHz;
    i2c = I2C_open(Board_I2C_TMP, &i2cParams);
    if (i2c == NULL) {
        Display_printf(display, 0, 0, "Error Initializing I2C\n");
        while (1);
    }
    else {
        Display_printf(display, 0, 0, "I2C Initialized!\n");
    }

    /*
     * ===== Start initializations with the sensor =====
     */

    //Send RESET Command to the sensor
    txBuffer[0] = 0x1E;
    i2cTransaction.writeBuf = txBuffer;
    i2cTransaction.writeCount = 1;
    i2cTransaction.readBuf = rxBuffer;
    i2cTransaction.readCount = 0;

    i2cTransaction.slaveAddress = ADDR;
    if (!I2C_transfer(i2c, &i2cTransaction)) {
        Display_printf(display, 0, 0, "Unsuccessful I2C transfer");
        while(1);
    }
}

```

```

    }
    sleep(0.02); // Wait for reset to complete

//Send READ PROM Command: read the sensor calibration data from the rom
uint8_t i,j;
for (i=0; i<8; i++){
    j=i;
    txBuffer[0] = 0xA0 + (j<<1);
    i2cTransaction.writeBuf = txBuffer;
    i2cTransaction.writeCount = 1;
    i2cTransaction.readBuf = rxBuffer;
    i2cTransaction.readCount = 2;

    i2cTransaction.slaveAddress = ADDR;
    if (I2C_transfer(i2c, &i2cTransaction)) {
        HB = rxBuffer[0];
        LB = rxBuffer[1];
        C[i] = HB << 8 | LB;
        //Display_printf(display, 0, 0, "C %d value: %d ", i, C[i]);
    }
}

//Start the sensor pressure conversion D1
txBuffer[0] = 0x48;
i2cTransaction.writeBuf = txBuffer;
i2cTransaction.writeCount = 1;
i2cTransaction.readBuf = rxBuffer;
i2cTransaction.readCount = 0;

i2cTransaction.slaveAddress = ADDR;
if (!I2C_transfer(i2c, &i2cTransaction)) {
    Display_printf(display, 0, 0, "Error1");
}
else{
    Display_printf(display, 0, 0, "D1 conversion performed!");
}

//Send ADC READ Command: read the previous conversion result (pressure)
sleep(1); // delay to perform the conversion
txBuffer[0] = 0x00;
i2cTransaction.writeBuf = txBuffer;
i2cTransaction.writeCount = 1;
i2cTransaction.readBuf = rxBuffer;
i2cTransaction.readCount = 3;

i2cTransaction.slaveAddress = ADDR;
if (I2C_transfer(i2c, &i2cTransaction)) {
    HB = rxBuffer[0];
    MB = rxBuffer[1];
    LB = rxBuffer[2];
    D1 = HB;
    D1 = D1 << 16 |MB << 8 | LB;
}
else {
    Display_printf(display, 0, 0, "Error2");
}

```

```

//Start the sensor temperature conversion D2
txBuffer[0] = 0x58;
i2cTransaction.writeBuf = txBuffer;
i2cTransaction.writeCount = 1;
i2cTransaction.readBuf = rxBuffer;
i2cTransaction.readCount = 0;

i2cTransaction.slaveAddress = ADDR;
if (!I2C_transfer(i2c, &i2cTransaction)) {
    Display_printf(display, 0, 0, "Error1");
}
else{
    Display_printf(display, 0, 0, "D2 conversion performed!");
}

//Send ADC READ Command: read the previous conversion result
(temperature)
sleep(1);
txBuffer[0] = 0x00;
i2cTransaction.writeBuf = txBuffer;
i2cTransaction.writeCount = 1;
i2cTransaction.readBuf = rxBuffer;
i2cTransaction.readCount = 3;

i2cTransaction.slaveAddress = ADDR;
if (I2C_transfer(i2c, &i2cTransaction)) {
    HB = rxBuffer[0];
    MB = rxBuffer[1];
    LB = rxBuffer[2];
    D2 = HB;
    D2 = D2 << 16 | MB << 8 | LB;
}
else {
    Display_printf(display, 0, 0, "Error2");
}
/*
* ===== Reading pressure and temperature values from the sensor =====
*/
int64_t dT=0;
int64_t OFF=0;
int64_t SENS=0;
int64_t SENSi = 0;
int64_t OFFi = 0;
int64_t Ti = 0;
int64_t OFF2 = 0;
int64_t SENS2 = 0;
int64_t t=0;
int64_t p=0;
float temp=0;
float press=0;

// Calculation according MS5837-01BA data sheet
dT = D2 - (int64_t)C[5]* 256;
OFF = (int64_t)C[2] * 65536 + ((int64_t)dT * (int64_t)C[4])/128;
SENS = (int64_t)C[1] * 32768 + ((int64_t)dT * (int64_t)C[3])/256;
t = 2000 + ((int64_t)dT * (int64_t)C[6]) / 8388608;
p = ((int64_t)D1*SENS/2097152 - OFF) / 8192;

```

```

// Second order temperature compensation
if((t/100)<20){
    Ti = 3*(int64_t)dT*(int64_t)dT/8589934592;
    OFFi = (3*(t-2000)*(t-2000))/2;
    SENSi = 5*(t-2000)*(t-2000)/8;
}
else{
    Ti = 2*(int64_t)dT*(int64_t)dT/137438953472;
    OFFi = ((t-2000)*(t-2000))/16;
    SENSi = 0;
}
if((t/100)<15){
    OFFi = OFFi + 7*((t+1500)*(t+1500));
    SENSi = SENSi + 4*(t+1500)*(t+1500);
}
//Calculate pressure and temperature second order
OFF2 = OFF-OFFi;
SENS2 = SENS-SENSi;
temp = (t-Ti)/100; //temperature in Celsius
press = (((int64_t)D1*SENS2/2097152-OFF2)/8192)/10000; //press in bar
Display_printf(display, 0, 0, "temperature compensated: %f C", temp);
Display_printf(display, 0, 0, "pressure compensated: %f bar", press);

I2C_close(i2c);
Display_printf(display, 0, 0, "I2C closed!");
return (NULL);
}

```

Matlab code for data acquisition from pressure transducer

```

try
    fclose(Arduino);
    delete(Arduino);
    clear Arduino;
    close(h2);
end

Arduino = serial('COM3', 'BaudRate', 9600, 'Terminator', 'CR');
fopen(Arduino);
StartTime = datestr(datetime('now'));
name=['PressureSensor'];
%Variables
duration=30*60;
PS_A_Range=60;
p_atm=14.7;
init=0;
h=figure(1);
count=1;
plot2_sample=100;
t0=now();
t(1)=0;
V=[];
i=0;
last_saved=0;

```

```

while t(end)<duration
    i=i+1;
    t(i)=(now()-t0)*24*3600;
    flushinput(Arduino)
    V(i) = str2double(fscanf(Arduino))/1024*5;
    if i>5 && abs(V(i)-mean(V(i-5:i-1)))>V(i-1)*0.4
        i=i-1;
    else
        Pressure(i)=(V(i)-1)/4*PS_A_Range+p_atm;      %pressure in psi
        Pressure(i)=Pressure(i)*0.06894757;          %pressure in bar
    end

    if i==1
        figure(1)
        set(gcf, 'Units', 'Normalized', 'Outer Position', [0,0.49,0.5,0.50]);
        subplot(1,2,1)
        c=plot(t/60, Pressure, 'LineWidth',2);
        xlabel('Time, min'),ylabel('Pressure, bar'),grid on
    end

    if mod(i,5)
        c.XData=t/60;
        c.YData=Pressure;
        if i>plot2_sample+1 d.XData=t(end-plot2_sample:end);
            d.YData=Pressure(end-plot2_sample:end);
            legend(d, sprintf('Average: %.2f , Slope:
%.2f', mean(Pressure(end-plot2_sample:end)), (Pressure(end)-Pressure(end-
plot2_sample))/(t(end)-t(end-plot2_sample)) ), 'Location', 'NorthWest')
        else
            d.XData=t;
            d.YData=Pressure;
        end
        drawnow
    end

    if (t(end)/60-last_saved)>5
        save([datestr(now, 'mmmdd_HHMM_') name], 'V', 'Pressure', 't');
        last_saved=t/60;
        fprintf([datestr(now, 'mmmdd_HHMM_') name] ' saved!\n'])
    end

    if init==0, init=1; end
end
save([datestr(now, 'mmmdd_HHMM_') name], 'V', 'Pressure', 't');
last_saved=t/60;
fprintf([datestr(now, 'mmmdd_HHMM_') name] ' saved!\n'])
figure
plot(t(1:length(Pressure))/60, Pressure, 'LineWidth',2)
title('Refilling Pressure'),xlabel('Time, min'),ylabel('Pressure, psi')
legend('Pressure'),grid on, axis tight
try
    fclose(Arduino);
    delete(Arduino);
    clear Arduino;
    close(h2);
end

clipboard('copy', sprintf([StartTime '\t' '%.f'], t(end)))

```

Matlab code for data acquisition from the device

```
clear all
close all
clc

Addr = {'A4','DA','32','C5','0C','FB'};
load commands
MSGID='MATLAB:serial:fread:unsuccessfulRead';
warning('off', MSGID)
try
    fclose(instrfind)
end

usb=Inizialize();
fprintf('All initialization performed!\n');

NOW=now(); %Current date and time as date number
a=now;
iteration=1;
time=30; % seconds
flag_filter=1;
j=1;
Pressure=[];
Temperature=[];
TimeMin=[];
last_saved=0;
name=['PressureSensor'];

while iteration~=10000
    [DiscList Discovery_list Discovery_extList]=DiscoverProtocol(usb, Addr,
iteration, NOW,time,flag_filter);
    %save pressure and temperature values inside a variable
    for i=1:length(Discovery_extList)
        if Discovery_extList{1,i}.AdvRptEventType==27
            p=Discovery_extList{1,i}.Data(18:20);
            p=p(1)*256^2+p(2)*256+p(3);
            press=p/10000; %pressure value in bar
            fprintf(1,'pressure [bar]: %d\n',press);
            t=Discovery_extList{1,i}.Data(16:17);
            t=t(1)*256+t(2);
            temp=t/100; %temperature value in Celsius degree
            fprintf(1,'temperature [Celsius]: %d\n',temp);
            time_min=(now-a)*24*60;
            Pressure(j)=press;
            Temperature(j)=temp;
            TimeMin(j)=time_min;

            %plot pressure and temperature as function of time
            if j==1
                subplot(1,2,1);
                pline=plot(TimeMin,Pressure,'-o');
                grid on;
                xlabel('time [min]');ylabel('pressure [bar]');

                subplot(1,2,2);
```

```

        tline=plot(TimeMin,Temperature,'-o');
        grid on;
        xlabel('time [min]');ylabel('temperature [Celsius]');
    else
        pline.XData=TimeMin;
        pline.YData=Pressure;
        tline.XData=TimeMin;
        tline.YData=Temperature;
    end
    drawnow;

    if (TimeMin(j)-last_saved)>60    %save the variables every
hour
        save([datestr(now,'mmmdd_HHMM_')name],'Pressure',
'Temperature','TimeMin');
        last_saved=TimeMin(j); %min
        fprintf([datestr(now,'mmmdd_HHMM_') name ' saved!\n'])
    end
    j=j+1;

    end
    end
    iteration=iteration+1;
end

```