



Politecnico di Torino

Department of Electronics and Telecommunications
Underwater communication system via VLF EM waves

Master degree in Electronic Engineering

Supervisor:
Prof. Claudio SANSOÈ

Candidate:
Giorgio FERRERO

October 2019

Abstract

In recent years the need to have submarine devices able to communicate with each other has arisen. This communication can take place in different ways. In the event that this occurs via a wired connection, a very fast, stable and reliable link is obtained, but it presents considerable limitations. The trade off mainly concerns the mechanical aspect of establishing a fixed connection. If the cable is left permanently connected, it could be damaged due to erosion or due to more traumatic events. These aspects could greatly increase maintenance costs and reduce the reliability of the entire system. Furthermore this solution is absolutely to be avoided if it is required to build a network of submarine mobile drones. A fixed connection between the different devices could get stuck during the movements damaging the entire system. On the other hand, if the wired connection is made only in the moment of need, for example when it is required to extrapolate data collected over a long period of time from a sensor anchored on the seabed other types of problems are present. In this case the device should arise to the sea surface in order to be connected, this procedure is safer for the data connection, but could wear out the anchoring system of the device. In fact, the retraction of the device in the sea depths causes considerable stress to the anchoring system, that also requires a huge amount of energy to operate. Otherwise, if the device can be reached by an operator, there is the problem of the risks associated with diving in hostile environments. Even these solutions are not efficient from an operational and economic point of view.

Another possible solution is to adopt a wireless communication system. Even in this case different technologies are available. The most used on the market right now are acoustic waves. Acoustic waves provide a very long range of communication (up to 20 km) although with a quite limited bandwidth. But it is fundamental to acknowledge under which factors and conditions their reliability drops. The first one is related to water level, if the height of water is not enough the wave has no sufficient room to propagate and it suffers from a considerable loss in range detectability. Acoustic waves are negatively affected also by turbidity, salinity, pressure gradients and reflections. This last flaw makes it very difficult to transmit thorough the barrier airwater. It must be also taken into account that these waves are also used by the marine fauna to communicate. An artificial interference in this link could cause problems for the animals.

The second technology considered for this project is transmission via optical waves. This kind of wireless link offers a very high transfer speed (in the order of Gbps). But it is very complex to exploit fully this speed because of the reliability of the link and due to the requirement to have the receiver and the transmitter in line of sight without anything between them. Optical waves are affected by water absorption, by scattering due to suspended particles and by the sun light that interferes with the data stream.

The third solution is the one chosen for the project. Electromagnetic waves are a reliable link which can provide an high bandwidth (Mbps) within a range of some tens of meters. EM waves are affected by salinity and temperature, but they are immune to turbidity, pressure gradients, sediment particles and acoustic noise. EM waves do not require in line of sight positioning and they can cross barrier between water/air and between water/seabed following the path of least resistance (multipath propagation).

This phenomenon makes transmission much easier in shallow water and makes communication between air and water possible. The project of this thesis is to investigate the performance of low-frequency electromagnetic waves in a marine environment.

In order to test the EM waves a system with two distinct voltage domains is developed. The system is powered by a 2.6 Ah Li-On 18650 battery whose voltage varies according to the charge level. So the two domains are powered using two different voltage regulators. The first domain is powered at 5 V and it is composed of a step up voltage regulator model TPS61032 (Texas Instruments) and a class D amplifier model TPA2008D2 (Texas Instruments) for the transmitting circuitry. The class D amplifier was chosen for its high efficiency and high output power. The second domain is regulated at 3.3 V and it is composed by a receiver, a wet contact, a pressure sensor and a development board equipped with a microcontroller that will manage all the mentioned hardware. The regulator is a buck regulator, model LTC3113 (Analog Devices), that provides a constant 3.32 V output voltage level. The LTC3113 was chosen for two reasons. First the LTC3113 is able to provide an extremely stable output voltage without any significant ripple. The regulator has the task of powering other devices as well, so it must be able to handle an high current output (up to 3 A). In order to amplify the received signal on the antenna a quad-operational amplified AD8618 (Analog Devices) was chosen. This integrated circuit has a very large GBW product and low offset, low noise structure. Since the system will deal with very low voltages as input and with an high gain (the goal is set to 60 dB) it is critical to have an IC with these characteristics.

These systems are printed on a PCB. The PCB has a shield shape compatible with an Arch Max 1.1 development board. The development board is produced by SeedStudio. The board is equipped with a STM32F407VET6 microcontroller and has some peripherals that are critical to the complete device. The first is a 12-bit ADC, it is connected to the output of the receiver to convert the output voltage of the receiver into the digital domain. Its output samples are copied in a circular buffer using a DMA channel. DMA is crucial for this application since, while the ADC is filling the one half of the circular buffer, the CPU must write the other half of the circular buffer on the SD card using CPU cycles. If the saving procedure was not fast enough with respect to the buffer filling time, part of the data would be lost due to overwriting of the circular buffer. The SD card is used as non volatile memory. The board has a SD card slot directly connected to the microcontroller through a SPI interface. The board has also a 12-bit DAC. This module is connected to the input of the class D amplifier. The DAC generates a sinusoidal wave at 10 kHz. The wave is alternated with periods of 1 s of silence (no sine) in order to get an OOK coded transmission. This kind of coding sends information through the presence or the absence of the carrier wave. The microcontroller sends a sinusoidal wave for 60 s then the coding begins with 1 s of silence. For the next 6 s are transmitted or 1 s of sine or 1 s of silence. The order in which a sine (1) or a silence (0) period is sent can be seen as a binary number. The encoded number changes dynamically every period since it represents the number of minutes elapsed from the beginning of the transmission. This information will be useful during the processing of the received data to maintain an easy to read temporal correlation of the data stream while two systems are moved in distance, height and angle in the sea depth. A pressure sensor is added to keep track of the depth in which the system is positioned. The sensor is able to detect temperature and pressure. These two raw data are used to complete a second order compensation algorithm in order to obtain an extremely precise pressure value. One pressure value is obtained every 25 s and it is stored on the SD card. The wet contact is used to trigger the microcontroller so the transmission and the reception can start as soon as the device is immersed in the water.

The time length of transmission and reception is regulated by a timer and can therefore be easily modified by software. When this period ends the microcontroller saves and permanently closes the data file on the SD and enters in a deep sleep state. In this way the analog circuit of the PCB is also deactivated.

The data present on the SD of the receiver are analyzed by a MATLAB script. Given the enormous

amount of data it was necessary to apply an algorithm to make the results easily analyzable by the user. The importance of having the entire sampling of the sine period available is in fact secondary, the algorithm only extracts the amplitude of each period. Once this procedure is finished, the amplitudes are averaged in subgroups of 400 periods and then these averages are plotted on the screen. The signal strength is on the vertical axis and the time is on the horizontal axis.

At this stage of the project the system aimed to test the EM wave in the marine environment is completed but it has been tested in free space. To test the system in the underwater environment a waterproof enclosure is required. Currently it is under development. When even this part of the project will be completed a test in the sea water will take place. As future development any type of complex modulation could be applied and the size of the system could be reduced by choosing other types of antennas with lower parasitic parameters and more compact dimensions.

Contents

Abstract	i
Table of contents	iii
1 Introduction	1
1.1 Underwater communication technologies	1
1.2 Thesis organization	4
2 Theoretical background	6
2.1 Sea water properties	6
2.1.1 Salinity	6
2.1.2 Conductivity	6
2.1.3 Permittivity	7
2.2 EM wave propagation theory	9
2.2.1 Summary of EM waves communication in underwater environment	11
2.3 Attenuation and choice of transmission frequency	13
2.3.1 Multipath Propagation	13
3 System Description	15
3.1 Required Interfaces	15
3.1.1 Receiving System Behaviour	17
3.1.2 Transmission System Behaviour	21
3.2 Hardware Description	25
3.2.1 Pressure sensor	25
3.2.2 μ SD	29
3.2.3 Wet Contact and μ C Low Power modes	31
3.2.4 Current absorption	36
4 Hardware Design	37
4.1 Transmitter and 5 V domain	37
4.1.1 Power Supply for 5 V domain	37
4.1.2 Transmitter	38
4.2 Power Supply for 3.3 V domain	38
4.2.1 Circuit design	39
4.3 Receiver	43
4.3.1 Circuit design	43
4.3.2 Stage analysis	44
4.3.3 Complete filter	51
4.3.4 Receiver Sensitivity	54

5	Analysis Software	55
5.1	Samples Extraction	55
5.2	Data Analysis	56
5.2.1	Averaging Function	56
5.2.2	Main Script	59
6	Final Hardware Testing	62
6.1	Sub-System testing	62
6.2	Complete system testing	66
6.2.1	Indoor testing	66
6.2.2	Outdoor testing	68
7	Conclusion and Future work	70
8	Bibliography	71

List of Figures

1.1	Block diagram of the RX TX chain	5
2.1	Dielectric permittivity and dielectric loss of water between 0 °C and 100 °C [10]	8
2.2	Attenuation of an EM wave as a function of frequency for different conductivities [12]	13
2.3	RF multipath propagation underwater [2]	14
3.1	ArchMax 1.1 Dev Board Pin OUT	16
3.2	Clock tree schematic STM32CubeMX	17
3.3	Single ADC block diagram	18
3.4	STM32F4xxx Family DAC structure [15]	22
3.5	Example of the OOK encoding. OSC settings vertical: 1V/div; horizontal: 2s/div	23
3.6	Main loop of the algorithm	24
3.7	IIC Reset Command [14]	26
3.8	IIC Reading 16 bits sequence [14]	27
3.9	IIC Reading 24 bits sequence [14]	27
3.10	IIC Pressure Sensor Algorithm. Green: dummy states. Blue: std states. Yellow: decisional stages.	29
3.11	μ SD pin schematic	30
3.12	SPI Module block diagram [15]	30
3.13	Wet Contact Circuitual Schematic [15]	32
3.14	V_{DDA} and V_{SSA} must be connected to V_{DD} and V_{SS} , respectively [15]	33
3.15	Backup domain diagram [15]	34
4.1	TPS61032 schematic	37
4.2	TPA2008D2 schematic	38
4.3	LTC 3113 schematic on LTSpice	39
4.4	LTC 3113 efficiency vs Load Current [13]	40
4.5	Error Amplifier with type III compensation	40
4.6	LTC 3113 startup transient on LTSpice $V_{in} = 1.8\text{ V}$ with a $10\ \Omega$ load	41
4.7	LTC 3113 startup transient on LTSpice $V_{in} = 3.5\text{ V}$ with a $10\ \Omega$ load	42
4.8	LTC 3113 startup transient on LTSpice $V_{in} = 5.5\text{ V}$ with a $10\ \Omega$ load	42
4.9	Multifeedback cell	43
4.10	Schematic of the first amplifier	44
4.11	Gain of the first stage	45
4.12	Closed loop gain of the first stage	46
4.13	Schematic of the second amplifier	46
4.14	Gain of the second stage	47
4.15	Closed loop gain of the second stage	47

4.16	Schematic of the third amplifier	48
4.17	Gain of the third stage	48
4.18	Closed loop gain of the third stage	49
4.19	Schematic of the fourth amplifier	49
4.20	Gain of the fourth stage	50
4.21	Closed loop gain of the fourth stage	50
4.22	Gain of the complete filter	51
4.23	Filter response to a 1 kHz 1 mV amplitude sine wave	52
4.24	Filter response to a 10 kHz 1 mV amplitude sine wave	52
4.25	Filter response to a 100 kHz 1 mV amplitude sine wave	53
4.26	Output noise of the filter	53
4.27	Receiver chain from analog to digital storage	54
5.1	Output plot of the MATLAB script. In blue the amplitude in percentage and in red the pressure in mbar separated by 25 s	60
6.1	Layout of the PCB. In blue metal layer one and in pink metal layer two.	63
6.2	Settling time of the output voltage of the 3.3 V regulator	64
6.3	Scheme of the circuit used to test the receiver amplifier.	65
6.4	Generated signal in red and amplifier output signal in blue. Settings 200 mV/div, 40 μ s/div.	65
6.5	Voltage signal across the transmitting antenna, measured using two probes. Settings 50 V/div, 20 μ s/div	66
6.6	Distance of 12 m. The orange line shows the pressure value [mbar], and the blue one is the signal strength [V/V] in percentage.	67
6.7	Distance of 19.2 m. The orange line shows the pressure value [mbar], and the blue one is the signal strength [V/V] in percentage.	67
6.8	Distance of 15.2 m. The orange line shows the pressure value [mbar], and the blue one is the signal strength [V/V] in percentage.	68
6.9	Distance of 20 m. The orange line shows the pressure value [mbar], and the blue one is the signal strength [V/V] in percentage.	68

List of Tables

1.1	Comparison of underwater wireless communication technologies	3
2.1	Changes in seawater salinity compared to temperature [9]	7
2.2	EM performance underwater.	10
2.3	EM performance underwater.	11
2.4	Summary of advantages of underwater RF-EM technology [2]	12
2.5	Example data rates for potential ranges of underwater EM [2].	13
3.1	ADC Pins [15]	19
3.2	DAC Pins	22
3.3	Command structure	26
3.4	Pressure sensor calibration coefficients	27
3.5	Current related to μC modes	36
3.6	Current related to μC peripherals	36
6.1	Indoor testing signal strength related to the distance summary	67
6.2	outdoor testing signal strength related to the distance summary	69

CHAPTER 1

Introduction

1.1 Underwater communication technologies

In recent years, sensor networks have seen a demand increase in the undersea environment. When it is required to install a sensor network on the seabed, it is vital to have an excellent link to extract the data collected from the sensor memory. Three possible solutions are commonly implemented [1].

The first involves the use of a wired link between the collector and the sensor. This solution is to be considered impractical in most cases since the sensors could be at high depths or in places that are difficult to reach.

The second solution provides that the sensor is equipped with a retractable buoy equipped with a communication device. At the time of need, the buoy is released and reaches the surface to transmit the data. Even this solution cannot be easily implemented since the energy required to retract the buoy in the sea depths is high and since the anchoring system to the seabed of the sensor would be put under heavy stress at every data collection.

The constraints of these two solutions are overcome by wireless communication systems. Research studies are in fact focused on the creation of sensors with an integrated communication system and on updating old cable-based infrastructures. Wireless communications can take advantage of different types of data transmission signals.

The most commonly used solution is communications via acoustic waves. This well established technology offers a very long range of communication, up to 20km, but it has drawbacks too. Acoustic waves suffer from a considerable loss of performance in low-depth underwater environments. Waves are in fact disturbed by turbidity, salinity, pressure gradients and reflections, this last flaw makes it very difficult to transmit thorough the barrier air–water. Moreover, the growing demand for these kind of system is focused on speed and acoustic waves are limited in bandwidth up to 20kb/s and chances of getting a performance improvement, in terms of speed, out of acoustic modems are quite remote [2].

Research has therefore focused on other types of emissions:

- Optical
- Electro Magnetic Waves

Optical waves: Communication systems based on optical waves are able to reach a band in the order of Gbps. However, the marine environment presents some problems for this type of waves. Optical waves are challenged by water absorption, by scattering due to suspended particles and by the sun light [3]. In any case even if it were possible to overcome these problems the receiver and the

transmitter must be in line of sight. These strong limitations do not make the system attractive for general applications.

EM Waves: This solution is probably the best one. EM waves are affected by salinity and temperature but they are immune to turbidity, pressure gradients, by sediment particles and acoustic noise. EM waves do not require line-of-sight positioning and they can cross barrier between water/air and between water/seabed following the path of least resistance, making the transmission much easier in shallow water and proving suitable for air/underwater communication. These barriers are responsible for a phenomenon called multipath propagation which can help the propagation of the useful signal toward the receiver. The main problem with this type of transmission is the attenuation due to sea water' salinity and temperature[4] . As shown in the article "Electromagnetic communications within swarms of autonomous underwater vehicles" the improvement in performance in relation to the increase in power at the transmitter has a reduced slope. This is caused by the high attenuation of sea water which reduces the power to the receiver exponentially compared to the distance [5] and therefore the communication range is limited to a few tens of meters.

A brief summary of the key advantages and limitations of technologies discussed above is presented on the Table 1.1.

Method	Benefits	Limitations
Acoustic	<ul style="list-style-type: none"> • Proven technology • Range: up to 20 km 	<ul style="list-style-type: none"> • Strong reflections and attenuation when transmitting through water/air boundary • Poor performance in shallow water • Adversely affected by turbidity, ambient noise, salinity, and pressure gradients • Limited bandwidth (0 b/s to 20 kb/s) • Impact on marine life
RF	<ul style="list-style-type: none"> • Crosses air/water/seabed boundaries easily • Prefers shallow water • Unaffected by turbidity, salinity, and pressure gradients • Works in non-line-of-sight; unaffected by sediments and aeration • Immune to acoustic noise • High bandwidths (up to 100 Mb/s) at very close range 	<ul style="list-style-type: none"> • Susceptible to EMI • Limited range through water
Optical	<ul style="list-style-type: none"> • Ultra-high bandwidth: gigabits per second • Low cost 	<ul style="list-style-type: none"> • Does not cross water/air boundary easily • Susceptible to turbidity, particles, and marine fouling • Needs line-of-sight • Requires tight alignment of nodes • Very short range

Table 1.1: Comparison of underwater wireless communication technologies

1.2 Thesis organization

The objective of the thesis is the design of a device able to transmit and receive signals at VLF and to use it for underwater communication. The thesis is focused on the hardware and software implementation of these two systems using a development board ARCHMAX 1.1 equipped with an STM32F407VET6 micro controller.

Two specific programs have been created for the different needs of the reception system and the transmission system.

The transmitter requests that the dev board generates the transmission signal with an ad-hoc coding using the built in DAC. While the receiving system requires to:

- Convert the analog output of the low noise band pass filter into a digital stream using the 12 BIT ADC
- Transfer the ADC output on a micro-SD card memory using the SPI interface allowing many hours of sampling

Both systems are equipped with a pressure sensor capable of detecting temperature and pressure. The sensor receives commands and transmits the readings through the IIC interface.

Software is designed to:

- Extracts temperature and pressure values from the sensor
- Adjust the read pressure value using a 2nd order compensation algorithm

The hardware system is designed to be powered by a rechargeable Li-On battery. Two DC-DC converters have been implemented to regulate the supply voltage on two different domains. The first one is at 3.3 V and it supplies the receiver, the micro-controller and the pressure sensor. The second domain at 5 V powers the transmitter.

The receiver has been designed to have a high amplification and a very narrow band around the transmitting frequency of 10 kHz.

The transmitter is a Class-D power amplifier.

Antennas loops have been chosen, their inductance has been coupled with a capacity so as to make them resonate at 10 kHz.

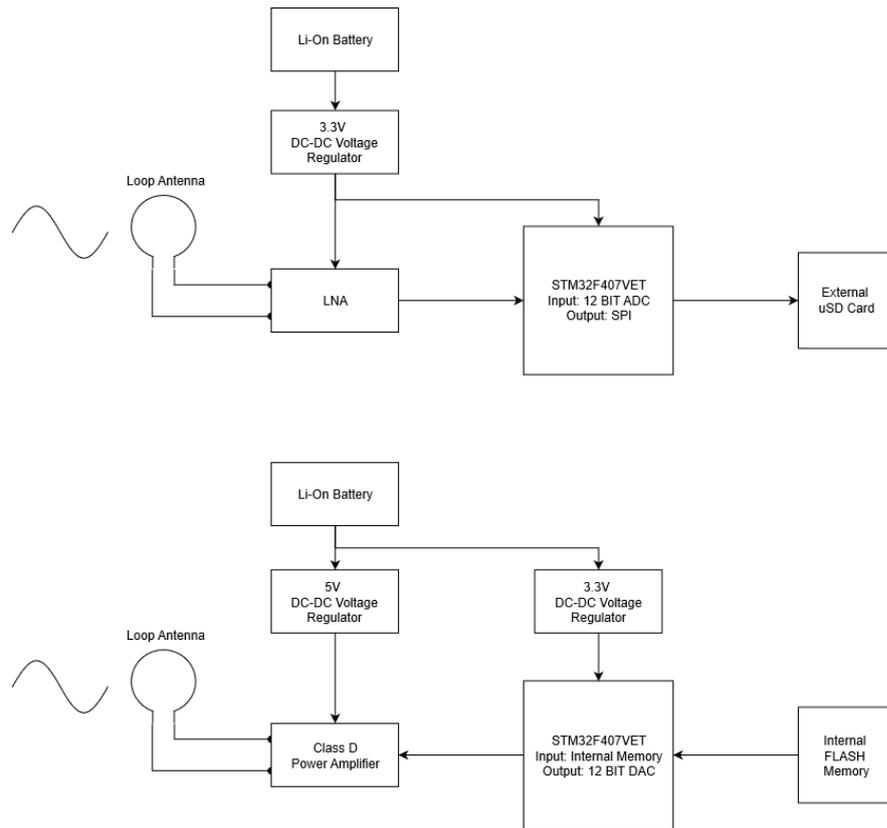


Figure 1.1: Block diagram of the RX TX chain

CHAPTER 2

Theoretical background

2.1 Sea water properties

Since the main environment for the application of the future system is sea water, it is necessary to study its physical properties and how these are influenced by any variation of seawater's chemical composition.

2.1.1 Salinity

Seawater typically contains an average salt density of 3.47%, with fluctuations from 3.4 to 3.6%. Salt means not only sodium chloride but a complex mixture of salts. For instance Cl^- and Na^+ ions are just around 2.9% of the whole composition, many other ions are present like potassium, sulfate and magnesium.

Studies show that the overall percentages of water salinity are however relatively constant all over the world and that they are not influenced by atmospheric events such as rain or drought [6]. Evaporation, the addition of river or rain water change the total salt concentration in sea water, but the relative proportions of different salts remain unchanged [7].

The hydrogen and oxygen atoms in the water molecule are bound by covalent bonds. This type of bond, in which the electron density is unevenly shared between the two bonded atoms, is responsible for a strong dipole moment. The dipole moment represents the resistance that a molecule opposes to be excited and is tightly bound to the dielectric constant of the medium. When trying to transmit an EM signal inside a medium, a wave is generated. The wave excites the molecules. These are forced to rotate, the spin absorbs energy by subtracting it from the EM wave and then attenuating it. The addition of salts inside the solution reduces the dipole moment of pure water, in this way the effects of absorption of the dipole moment through the reduction of the dielectric constant are mitigated [8].

2.1.2 Conductivity

Conductivity is the property that describes how much is a material able to conduct current. Any apolar compound tend to ionize water increasing its conductivity. Insulators like pure water have a very low conductivity, 0 S/m ideally. Typical values for different types of water are:

- Ideal pure water: $\sigma = 0$ S/m
- Ultra purified water: $\sigma = 5.510^{-6}$ S/m
- Drinking water: $\sigma = 0.005-0.05$ S/m

- Sea water: $\sigma = 4\text{--}5.3$ S/m

This phenomenon has a negative effect on EM waves attenuation, as resistance is lowered by any chemical species losses increases. Temperatures affects also the conductivity of seawater. In Table 2.1 the variation is shown for different levels of salinity.

Temperature [$^{\circ}C$]	Salinity [$kppm$]		
	20	30	40
0	1.745	2.523	3.285
5	2.015	2.909	3.778
10	2.300	3.313	4.297
15	2.595	3.735	4.837
20	2.901	4.171	5.397
25	3.217	4.621	5.974

Table 2.1: Changes in seawater salinity compared to temperature [9]

2.1.3 Permittivity

Permittivity is the ability of a material to store electrical potential energy under the influence of an electric field resisting to it. This phenomenon is represented by an complex number.

$$\varepsilon = \varepsilon_r \varepsilon_0 - j \frac{\sigma}{\omega} \quad (2.1)$$

Where the quantities involved are:

- ε_r : relative permittivity of the medium, focus on 2.2 and 2.3 equations
- $\varepsilon_0 = 8.854 * 10^{-12} F/m$: vacuum permittivity
- $\omega = 2\pi f$: angular frequency of the E field
- σ : conductivity of the medium

In the waveguide cell method, the dielectric constant is measured by calculating the shift in minima of the standing wave pattern in a rectangular waveguide. This shift takes place due to the change in the guide wavelength when a dielectric material is introduced in waveguide.

$$\varepsilon_r = (\lambda_a/2a)^2 + (\lambda_a/\lambda_{g\epsilon})^2 \quad (2.2)$$

$$\frac{\tan(2\pi(d+L)/\lambda_{ga})}{2\pi L/\lambda_{ga}} = \frac{\tan(2\pi L)/\lambda_{g\epsilon}}{2\pi L/\lambda_{g\epsilon}} \quad (2.3)$$

Where variables are:

- a: Width of the waveguide
- λ_a = Wavelength in the free space,
- λ_{ga} = Guide wavelength filled with air,
- $\lambda_{g\epsilon}$ = Guide wavelength when filled with loss less dielectric material.,
- d= Displacement of the minima of air after insertion of the dielectric,

- L = Length of the plane position where the impedance to be measured.

From eq. 2.3 λ_{ge} is found and then used to computer ϵ_r in eq.2.2.

Even if temperature and frequency influence relative permittivity and dielectric losses, these are considered constant at low frequencies as can be seen in figure 2.1.

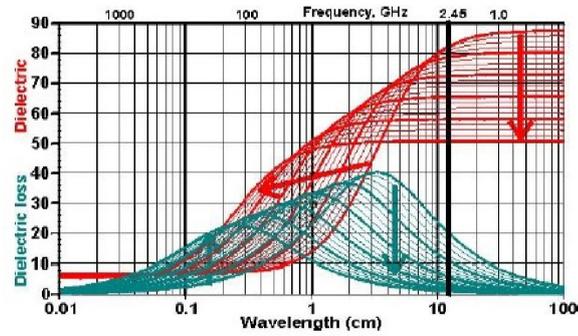


Figure 2.1: Dielectric permittivity and dielectric loss of water between 0 °C and 100 °C [10]

2.2 EM wave propagation theory

In the previous section it has been described how EM signals are affected by a lot of environmental factors such as temperature and salinity. In this section theoretical parameters will be examined for free space and in medium propagation.

In free space is an ideal condition so conductivity σ is 0 S m^{-1} and permittivity is constant and equal to 1, so EM waves are subjected to these parameters:

$$c = \lambda_0 f \quad (2.4)$$

$$k = k_0 = \frac{2\pi f}{c} \quad (2.5)$$

$$\gamma = \sqrt{(j\omega\epsilon)(j\omega\mu)} = jk_0 \quad (2.6)$$

$$\eta_{\text{free space}} = \frac{E}{H} = \sqrt{\frac{\mu}{\epsilon}} = Z_0 = 120\pi \quad (2.7)$$

Where:

- f is the frequency of the wave
- c is the speed of light in vacuum
- λ_0 is the wavelength in vacuum
- k_0 is phase constant of the wave in vacuum
- γ is the propagation constant of the wave
- μ is the magnetic permeability of the medium;
- η is the intrinsic impedance of the wave; Z_0 is the intrinsic impedance in free space

The wavelength is subject to a variation compared to the case in free space when the wave propagates in a ideal medium [11].

$$\frac{\lambda_0}{\lambda} = \frac{c_0 k}{\omega} \quad (2.8)$$

Even η , the characteristic impedance is modified:

$$\eta_{\text{water}} = \sqrt{\frac{\mu_{\text{water}}}{\epsilon_{\text{water}}}} = \sqrt{\frac{1.256 * 10^{-6} \text{ N/A}^2}{707 * 10^{-12} \text{ F/m}}} = 42\Omega \quad (2.9)$$

Despite these variations in physical parameters the EM waves maintain a considerable detachment in terms of propagation speed with respect to acoustic waves.

$$\nu = \frac{1}{\sqrt{\epsilon\mu}} \quad (2.10)$$

$$c = \sqrt{\frac{X}{\rho}} \quad (2.11)$$

Respectively from equation 2.10 and from equation 2.11 is possible to get propagation speed of EM waves and the speed of acoustic waves in any medium [9].

Parameter involved are:

- ϵ : medium permittivity [F/m]
- μ : medium permeability [H/m]
- X : bulk modulus of elasticity [psi]
- ρ : density of the medium [kg/m^3]

In case water this result are obtained:

- $\nu_{acoustic} = 1440 \text{ m s}^{-1}$

In the Table 2.2 it is possible to compare how EM signals at different frequencies behave in different mediums. In this table fresh water rows was calculated with $\sigma = 0.01 \text{ S/m}$ which is a typical value for fresh water. For sea water rows conductivity, σ was set to 3.2, 4.2, 4.3, 4.3, 4.3, and 5.4 S/m for the respective frequencies listed in the table. Parameters were calculated using these three expressions:

$$\lambda = 2\sqrt{\frac{\pi}{\eta_0 f \sigma}} \tag{2.12}$$

$$\nu = 2\sqrt{f\pi/(\mu_0\sigma)} \tag{2.13}$$

$$\delta_{skin} = 1/\sqrt{f\pi\sigma\mu_0} \tag{2.14}$$

		Frequencies (Hz)			
		100	1000	10,000	100,000
Propagation velocity (m/s)	Sea water	1.77×10^4	4.88×10^4	1.52×10^5	4.82×10^5
	Fresh water	3.16×10^5	1.00×10^6	3.16×10^6	1.00×10^7
	Free space	3.00×10^8	3.00×10^8	3.00×10^8	3.00×10^8
	Acoustic	1.50×10^3	1.50×10^3	1.50×10^3	1.50×10^3
Wavelength (m)	Sea water	1.76×10^2	4.88×10^1	1.52×10^1	4.82×10^0
	Fresh water	3.16×10^3	1.00×10^3	3.16×10^3	1.00×10^3
	Free space	3.00×10^6	3.00×10^5	3.00×10^4	3.00×10^3
Propagation distance (m) for 100 dB attenuation	Sea water	3.23×10^2	8.92×10^1	2.79×10^1	8.81×10^0
	Fresh water	5.78×10^3	1.83×10^3	5.78×10^2	1.83×10^2

Table 2.2: EM performance underwater.

So EM waves are faster than acoustic waves. It is interesting to notice that the electromagnetic propagation in water is only about 9 times slower than in free space. This has important advantages for command latency and networking protocols in underwater communications, where information has to be exchanged between different sensor[9]. However, it should be noted that EM waves in sea water suffer a very heavy reduction in wavelength compared to the free space case. This constrains the marine system to have an higher resolution.

In the third part of the table 2.2 [2] shows a quantitative analysis of the reduction in propagation distance within the frequency span from 100Hz to 10 MHz. Distances has been calculated with equations 2.14 and 2.15 considering an attenuation of 100 dB for a plane wave [2].

$$\frac{20 \log_{10}(1/e)}{\delta_{skin}} = \frac{100dB}{distance} \tag{2.15}$$

		Frequencies (Hz)	
		1 million	10 million
Propagation velocity (m/s)	Sea water	1.52×10^6	4.30×10^6
	Fresh water	3.16×10^7	1.00×10^8
	Free space	3.00×10^8	3.00×10^8
	Acoustic	1.50×10^3	1.50×10^3
Wavelength (m)	Sea water	1.52×10^0	4.30×10^{-1}
	Fresh water	3.16×10^3	1.00×10^3
	Free space	3.00×10^2	3.00×10^1
Propagation distance (m) for 100 dB attenuation	Sea water	2.79×10^0	7.87×10^{-1}
	Fresh water	5.78×10^1	1.83×10^1

Table 2.3: EM performance underwater.

2.2.1 Summary of EM waves communication in underwater environment

In [2] some examples of possible data rates at different distances were suggested. The focus of the project is to have a short-range communication system. Within a 10 m radius it should be possible to obtain at least an 8 ks data rate.

	Features	Details
Performance	Crosses water to air boundary	Long-range horizontal communication using air path, water-to-air, or land
	Multipath less of an issue	Can be advantageous in shallow water conditions
	Frequency-agile capability	No mechanical tuned parts as in an acoustic system
	Covert, localized communications	Using high-frequency carrier for high attenuation; close spatial frequency reuse
	High joules per bit efficiency	For short-range and high-bandwidth applications, high bit rate results in efficient system in terms of joules per bit, extending deployment times for battery operated equipment
	Potential for high data rates over small distances	Use of MHz carrier; does not require precise navigation for hard docking of connector-based systems. Improved reliability vs. connectors; avoids marine fouling, particulates, and alignment issues seen in laser-based systems
	High propagation speed	Low Doppler shift, low propagation delay especially important for networking protocols requiring multiple exchanges of information for handshake and error checking
Reliability	Unaffected by pressure gradients	Allows horizontal propagation
	Immune to acoustic noise	Operation unaffected by engine noise of heavy work, breaking waves, etc.
	Unaffected by low visibility	Sediment disturbed at the sea bed has no operational effect, while laser systems fail to operate
	Immune to aerated water	Operation in surf zone, communication at speed through cavitating propeller wash
Implementation	No need for surface repeater	Crosses water-to-air boundary for a long range without a surface repeater
	Distributed transducers	Radiating cables can deliver unique navigation and communications functions
	Compact, portable units	Small-to-medium antennas deliver acceptable performance
	No effects on marine animals	Effect of acoustic signals on marine mammals is becoming an issue

Table 2.4: Summary of advantages of underwater RF-EM technology [2]

2.3 Attenuation and choice of transmission frequency

Range	< 10 m	50 m	200 m	> 1 km
Sea water	> 8 kb/s	300 b/s	25 b/s	< 1 b/s
Fresh water	> 3 Mb/s	150 kb/s	9 kb/s	< 350 b/s
Application	AUV docking diver's personal network	Networks diver's conversation	AUV docking networking diver conversation	Deep water telemetry

Table 2.5: Example data rates for potential ranges of underwater EM [2].

In figure 2.2 it is possible to see the attenuation of an EM wave as a function of frequency and for different conductivities. These calculations were done in [12] following the Weyl model. The chosen frequency was 10 kHz in order to have a theoretical attenuation of 100 dB at about 30 m.

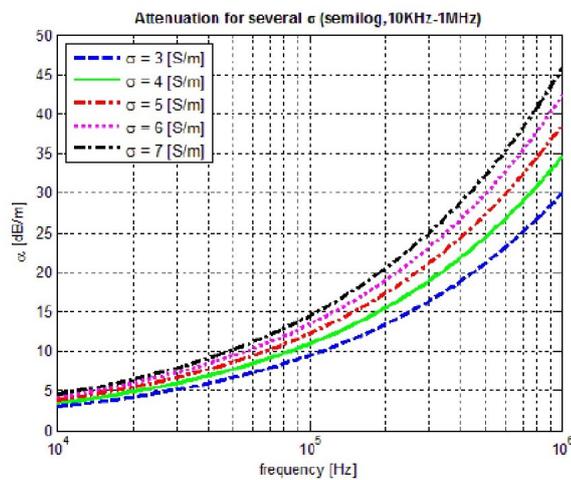


Figure 2.2: Attenuation of an EM wave as a function of frequency for different conductivities [12]

2.3.1 Multipath Propagation

Another physical phenomenon that must be considered is multipath propagation.

When a wave is generated and travels in space it is reflected, refracted and scattered by every hard object it encounters. This can cause the receiving device to transpose, in addition to the noise, a distorted version of the transmitted signal leading to inter symbolic interference (ISI) which in turn may result in erroneous demodulation. However in the case of communications between a transmitter under the sea and a receiver both submerged or on the surface this phenomenon helps a lot the transmission. To understand the phenomenon it is required to take into account two main factors. The first is the presence of some interfaces within the underwater environment: air–water, seabed–water. The second is the high permittivity of salt water. When a wave comes into contact with one of these interfaces it is scattered and thanks to the high permittivity the angle of reflection is practically parallel with the surface of the interface. This causes the transmitter to apparently emit the signal from higher than its actual position. This phenomenon is very useful as the seabed and the air has a markedly lower conductivity. The signal that then travels through them is much less subject to losses due to sea water. This phenomenon could lead to a significant improvement in the maximum transmission distance [13].

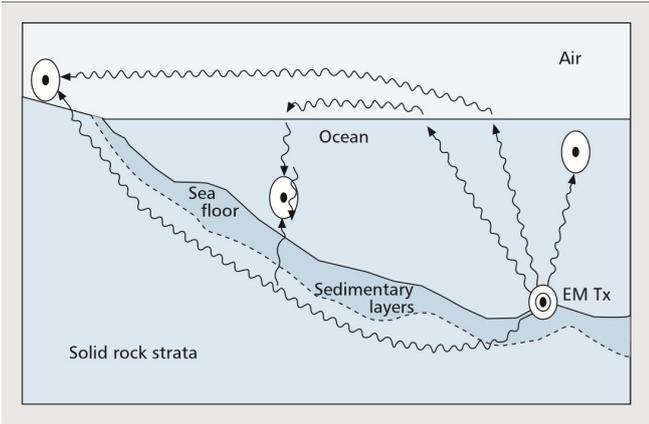


Figure 2.3: RF multipath propagation underwater [2]

CHAPTER 3

System Description

3.1 Required Interfaces

The complete system will require the integration of different hardware components. The main ones are:

- 12bits ADC
- 12bits DAC
- Pressure sensor
- Receiving system
- Storage on a μ SD
- Transmitting system
- Wet Contact

The core of the entire system is the ARCH MAX dev board. The board is based on an STM32 F407VET6 microcontroller with an ARM Cortex-M4F. The board provides Serial Wire Debug (SWD) and USB communication and fully supports the STM32CubeMX IDE and it was programmed including the HAL and LL driver libraries. To compile and load the project the development environment provided by ARM Ltd. KeilMDK μ Vision was used. The board pins are all programmable as general purpose input outputs and some are connected to particular microcontroller devices like the ADC or IIC interface. The board is equipped with an μ SD slot. The microcontroller interfaces with the μ SD via SPI. This device will be used as NVM for the storage of all the measurements taken from the system. As it is possible to see in the figure 3.1 the board has the typical structure suitable to support the use of shields. The reception and transmission systems will be realized with dedicated hardware on a PCB equipped with a pin out mirroring that of the Arch Max board. In this way the two components will be integrated in as little space as possible.

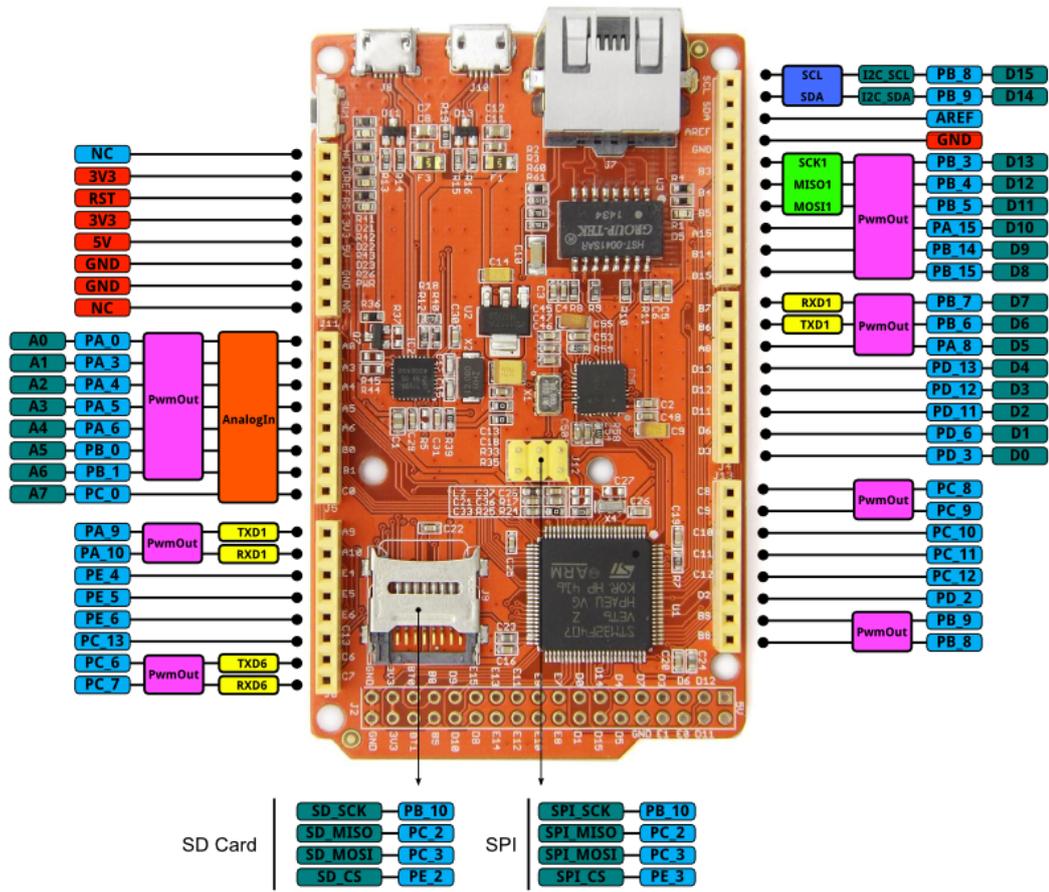


Figure 3.1: ArchMax 1.1 Dev Board Pin OUT

- PA0: Analog input for the ADC of the receiving system
- PA4: Analog output for the DAC of the transmission system
- PA5: Analog output for volume pin of the Class D amplifier
- PB1: Digital output for shutdown pin of the Class D amplifier
- PB6: Clock SCL of the IIC
- PB7: Data bus SDA of the IIC
- PA15: Enable Wet Contact
- PB14: Read Wet Contact
- PB10, PC2, PC3 and PE2 for the μ SD SPI interface

The clock tree has been configured selecting the HSE crystal oscillator as main source. The clock dividers and PLL configuration is reported here.

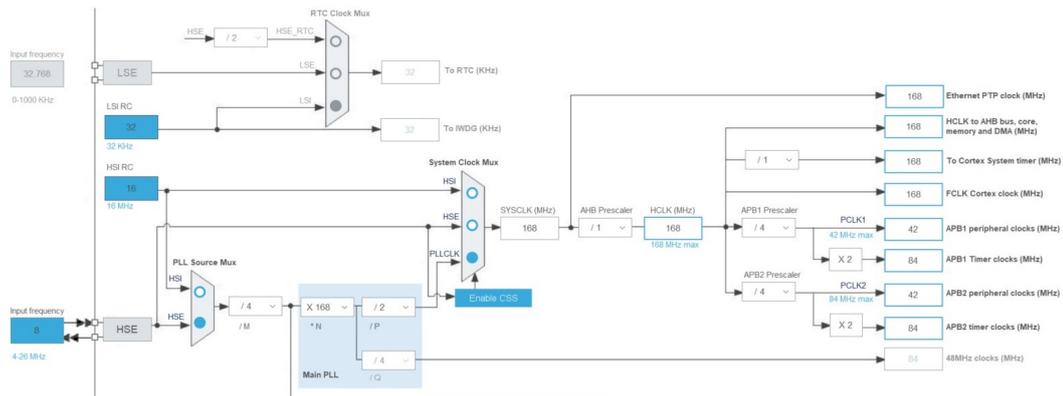


Figure 3.2: Clock tree schematic STM32CubeMX

```

1  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
2  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
3  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
4  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
5  RCC_OscInitStruct.PLL.PLLM = 4;
6  RCC_OscInitStruct.PLL.PLLN = 168; //160
7  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
8  RCC_OscInitStruct.PLL.PLLQ = 4;
9  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
10 {
11     _Error_Handler(__FILE__, __LINE__);
12 }
13
14 /**Initializes the CPU, AHB and APB busses clocks
15 */
16 RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYCLK
17                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
18 RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
19 RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
20 RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
21 RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV4;

```

Timers are based on the APB1 branch of the clock tree while the other peripherals are connected to the APB2 branch.

3.1.1 Receiving System Behaviour

The 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 19 multiplexed channels allowing it to measure signals from 16 external sources, two internal sources, and the VBAT channel. The A/D conversion of the channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored into a left- or right-aligned 16-bit data register. Here is reported the diagram of the ADC included.

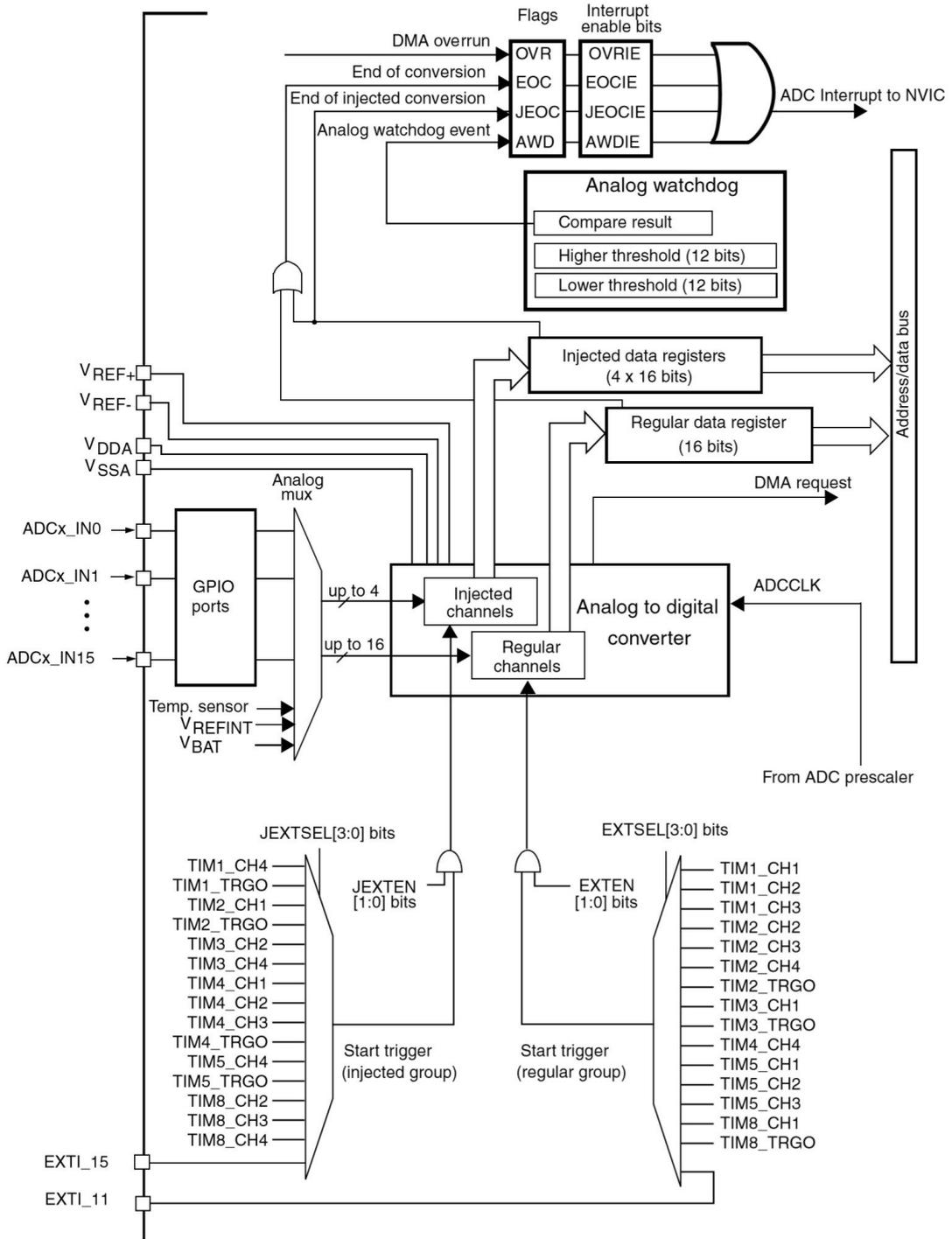


Figure 3.3: Single ADC block diagram

Name	Signal Type	Remarks
V_{ref+}	Input, analog reference positive	The higher/positive reference voltage for the ADC $1.8\text{ V} \leq V_{ref+} \leq V_{DDA}$
V_{DDA}	Input, analog supply	Analog power supply equal to V_{DD} and $2.4\text{ V} \leq V_{DDA} \leq V_{DD} 3.6\text{ V}$ (full speed) $1.8\text{ V} \leq V_{DDA} \leq V_{DD} 3.6\text{ V}$ (reduced speed)
V_{ref-}	Input, analog supply negative	The lower/negative reference voltage for the ADC, $V_{ref-} = V_{SSA}$
V_{SSA}	Input, analog supply ground	Ground for analog power supply equal to V_{SS}
ADCx_IN15:0	Analog input signal	16 analog input channels

Table 3.1: ADC Pins [15]

The clock source of the ADC is the APB2 domain and the peripheral has been configured with these parameters.

- Resolution of 12bit
- Continuous Conversion Mode enabled
- Data Aligned to right
- DMAC ontinuous Requests enabled
- ADC Sampling Time 144 cycles

```

1 static void MX_ADC1_Init(void)
2 {
3
4     ADC_ChannelConfTypeDef sConfig;
5
6     /**Configure the global features of the ADC (Clock, Resolution, Data Alignment and
7      *number of conversion)
8      */
9     hadc1.Instance = ADC1;
10    hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV8;
11    hadc1.Init.Resolution = ADC_RESOLUTION_12B;
12    hadc1.Init.ScanConvMode = DISABLE;
13    hadc1.Init.ContinuousConvMode = ENABLE;
14    hadc1.Init.DiscontinuousConvMode = DISABLE;
15    hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
16    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
17    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
18    hadc1.Init.NbrOfConversion = 1;
19    hadc1.Init.DMAContinuousRequests = ENABLE;
20    hadc1.Init.EOCSelection = DISABLE;
21    if (HAL_ADC_Init(&hadc1) != HAL_OK)
22    {
23        _Error_Handler(__FILE__, __LINE__);
24    }

```

```

25  /**Configure for the selected ADC regular channel its corresponding rank in the
    sequencer and its sample time.
26  */
27  sConfig.Channel = ADC_CHANNEL_0;
28  sConfig.Rank = 16;
29  sConfig.SamplingTime = ADC_SAMPLETIME_144CYCLES; //12 clk di offset;
30
31  if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
32  {
33    _Error_Handler(__FILE__, __LINE__);
34  }
35
36  }

```

An important feature available on the STM32 chip is a capability to use Direct Memory Access (DMA). Direct Memory Access allows to move data from periphery into the storage area or from a storage area to periphery without using Arithmetic and Logic Unit (ALU) registers. It allows to achieve greater performance and without wasting any CPU cycle. The ALU of the microcontroller only receives some information about how many information units should be sent and also the addresses of a receiver and transmitter.

The DMA was configured to save a continuous stream of data within a circular buffer of 1000 samples. The microcontroller supports two interrupts dedicated to circular buffers. The first is activated when the buffer filling reaches half of its capacity, the second when it is completely full. The ISRs of these interrupts consist in asserting a flag.

```

1  void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef* hadc1)
2  {
3    half_flag = 1;
4  }
5
6
7  void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc1)
8  {
9    full_flag = 1;
10 }

```

These two flags are read in the main loop, if they are asserted then the write_SD() function is invoked. The function to write to μ SD requires two parameters. The first is the pointer to the memory area that contains the data to be sent and the second is the number of bytes to be written. If the circular buffer appears to have been half filled, it is passed to the function the pointer to the beginning of the buffer and half the buffer capacity. In this way the first half of the buffer is written while the second half is filled. When the buffer is full then the writing of the second half begins while the first part is filled by the output of the ADC.

The receiving system is also equipped with the pressure sensor. Pressure values must be written on the μ SD too.

To avoid interrupting the aforementioned process when a new pressure value is available it is saved in a temporary variable and a flag is asserted. This flag is checked each time the first half of the circular buffer is filled. If this flag has been asserted, the pressure value replaces the ADC sample saved in position 0 and incremented by 4096. In this way only a sample of the ADC every 25s of sampling is lost, which is not a relevant loss. The pressure value is increased by 4096 to distinguish it from all other data saved on the SD. The ADC samples are in fact on 12 bits, therefore they range from a minimum of 0 to a maximum of 4095. Any sample greater than that value is therefore easily identified by the data analysis software, isolated from the ADC samples and reduced by 4096.

```

1  if (half_flag == 1)
2  {
3

```

```

4     if (new_pressure_value == 1)
5     {
6
7         new_pressure_value = 0;
8         HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_1);
9         values[0] = PRESSURE_value + 4096;
10
11    }
12
13    half_flag = 0;
14    write_SD(values, NSAMPLES); //(uint16_t*)
15
16    }
17
18
19    if (full_flag == 1)
20    {
21
22        full_flag = 0;
23        write_SD(values + NSAMPLES/2, NSAMPLES);
24
25    }

```

3.1.2 Transmission System Behaviour

The main task of this system is to send a sinusoidal signal. This signal is amplified via a class D amplifier. The base signal is generated through the DAC of the STM32F407VET microcontroller. The microcontroller is equipped with two DACs with dedicated channels. The resolution of the DAC is adjustable, it can be selected equal to 8 or 12 bits. Depending on the selected configuration mode on the x -th DAC channel, the data must be written into the specified register as described below:

- 8-bit right alignment: the software has to load data into the DAC_DHR8Rx [7:0] bits (stored into the DHRx[11:4] bits)
- 12-bit left alignment: the software has to load data into the DAC_DHR12Lx [15:4] bits (stored into the DHRx[11:0] bits)
- 12-bit right alignment: the software has to load data into the DAC_DHR12Rx [11:0] bits (stored into the DHRx[11:0] bits)

The content of the DHRx register is transferred to the DORx (*Date Output Register*) register which represents the actual input to the digital to analog converter circuitry. The content of the DORx register is converted into a voltage level with the following proportion.

$$DAC_{output_voltage} = V_{ref+} * \frac{DOR}{2^{resolution}} [V] \quad (3.1)$$

Where DOR is the digital value in LSBs and V_{ref+} is the reference voltage of the DAC system, equal to 3.3V in this case.

The sine wave was generated using the MATLAB software on a linear space between 0 and 2π of 60 values. The values were then converted to LSB on a 12bit scale, where 0 corresponds to -1 and 4095 corresponds to 1. This array of samples is stored into the memory of the microcontroller. An array of equal size containing only "2048" is also stored in memory. In this way it will be possible to generate either a sinusoid period or a period of constant voltage (1.65V) of the same duration. The system supports the use of a dedicated DMA system.

The DAC trigger can be selected between different sources as shown in the figure 3.4.

sinusoidal wave. This phase is useful to collect data about the amplitude of the signal received by the other device. The second is a phase of coding, this phase has a shorter duration than the first phase. This phase sends a modified version of the signal encoding, in a binary–Big Endian form, the number of cycles of the first phase elapsed since the beginning of the transmission.

```
1 #define NBIT_CODING 6
2 #define SECONDS_PER_PERIOD 2
```

The duration (in seconds) of the first phase is set by the constant SECONDS_PER_PERIOD. NBIT_CODING sets on how many bit can be encoded the second phase. In this case NBIT_CODING is set to 6 so the max number of first phase cycles that can be counted is 64. Then the coding is reset and restarts from 0. In figure 3.5 has been generated a short version of the typical transmission. In this test SECONDS_PER_PERIOD has been set equal to 2 and NBIT_CODING equal to 6. In yellow is reported the time duration of the first phase, which is followed by 1 second of silence (green). Then phase 2 begins (white) and terminates with a second of silence (green). In this case the coding represents the number "100010" since this was the 34th times that the first phase was repeated. Eventually the simple transmission for SECONDS_PER_PERIOD seconds is restarted.

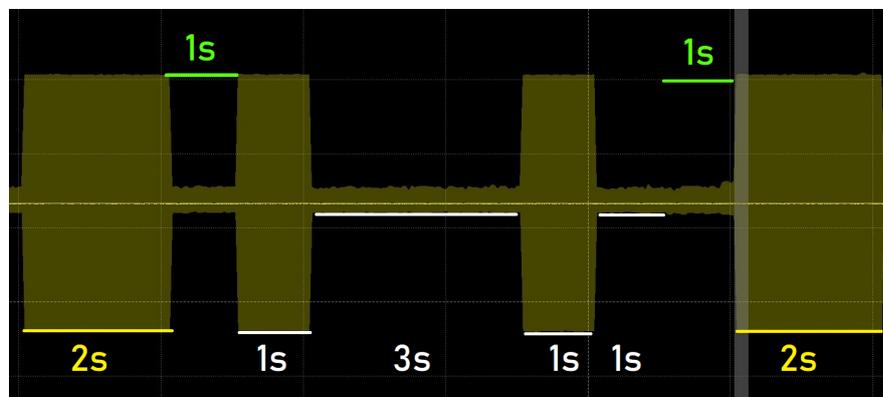


Figure 3.5: Example of the OOK encoding. OSC settings vertical: 1V/div; horizontal: 2s/div

In the first phase the DAC generates a sinusoidal signal for SECONDS_PER_PERIOD seconds, then the phase two begins. The first second of coding is a second of silence (no transmission), then are transmitted NBIT_CODING periods of sinusoidal signal or silence. Phase two ends with another period of silence.

Here is reported the while(1) loop of the program:

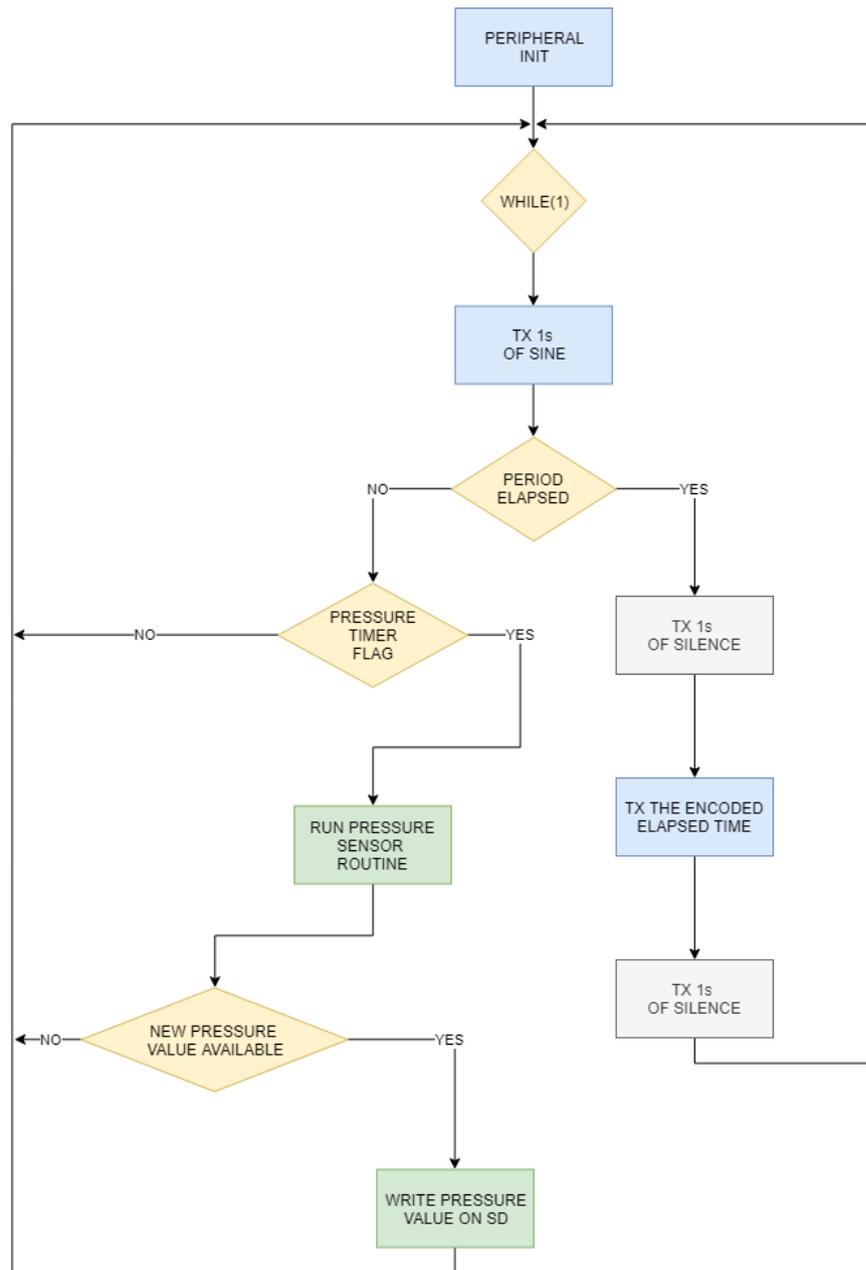


Figure 3.6: Main loop of the algorithm

Here is reported the interrupt callback routine the enabled timers:

```

1 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
2 {
3
4   if(htim == &htim2)
5   {
6
7     pressure_timer_flag = 1;
8
9   }
10
11  if(htim == &htim4)

```

```

12  {
13
14  tim4_counter ++ ;
15
16  if (tim4_counter % (SECONDS_PER_PERIOD+1) == 0)
17  {
18
19      if (tx_of_coding != 1)
20      {
21
22          minute_elapsed_flag = 1;
23
24          tim4_counter = 0; // avoid OVF
25
26          min_counter++;
27
28      }
29  }
30
31 }

```

Timer 4 is set to have a frequency of 1Hz, it is used to control the period of transmission of the first phase. The ISR of this timer simply updates a counter and compare it to the constant SECONDS_PER_PERIOD. If they are equal there is a chance that the ISR has to enable the transmission of the coded signal and has to update the number of elapsed period of transmissions. Since the coding takes $(2 + \text{NBIT_CODING})$ seconds to be completed this ISR could be called multiple times during this period (if SECONDS_PER_PERIOD is shorter than the coding period). To prevent the execution of the ISR, in particular the increment of *min_counter*, the *tx_of_coding* flag is checked. This flag is set in the main loop when the encoded transmission begins and dis-asserted when it ends.

Timer 2 is used to control the pressure sensor and it is set to call its ISR every 2.5. This ISR is very basic, it asserts just one flag. This flag is read in the main loop. If it is found true (equal to 1) the main loop executes one step of the algorithm, when a valid pressure value is returned by the *pressure_sensor()* function it is written on the μ SD card. The management of the pressure sensor will be described in details in the Hardware Description section.

3.2 Hardware Description

3.2.1 Pressure sensor

The purpose of the system is to verify the maximum possible communication distance using a 10 kHz RF signal in an underwater environment. As previously analyzed, many factors can negatively or positively affect the quality of the transmission. If we consider the multipath propagation, for example, it is essential to know the positioning of the reception system with respect to the transmission system. During field tests the systems will be put in water at varying distances and depths. To keep track of the relative positioning of the two systems and to have a certain temporal correlation a IIC interface has been implemented. The interface will be responsible for communicating with a pressure sensor. The sensor management algorithm is divided into 10 steps. Every 10 steps a new pressure reading is available and it is written on the μ SD card. From the Stevino's equation 3.2 the position of the sensor can be evaluated.

$$p = \rho gh \quad (3.2)$$

Where parameters are:

- p : Pressure [mbar]
- ρ : Density [kg/m³]
- g : Gravitational acceleration [m/s²]
- h : Height/Depth [m]

The timing of the activation and execution of one step of the algorithm will be managed by a timer on the ARCHMAX dev board.

Having considered the type of application and the related requirements, the MS5837-30BA pressure sensor from TE Connectivity was chosen. The sensor module includes a high linearity pressure sensor and an ultra-low power 24 bit ΣΔ ADC with internal factory calibrated coefficients stored on a 112 bit PROM. These coefficients will be read by the microcontroller at the startup time and used in the program to convert the raw readings of temperature and pressure into compensated pressure and temperature values. The microcontroller clocks the peripheral using the SCL (Serial CLock) line (max 400 kHz). The sensor receives commands and transmits on the pin SDA (Serial DATA), which is bidirectional. So this interface type uses only 2 signal lines and does not require a chip select. The microcontroller sends commands on the SDA line, commands are coded on 8 bits.

Bit number from MSB	0	1	2	3	4	5	6	7	
Command name									Hex value
Reset	0	0	0	1	1	1	1	0	0x1E
Raw Pressure	0	1	0	0	1	0	1	0	0x4A
Raw Temperature	0	1	0	1	1	0	1	0	0x5A
ADC Read	0	0	0	0	0	0	0	0	0x00
PROM Read	1	0	1	0	X	Y	Z	0	0xA0 to 0xAE

Table 3.3: Command structure

Pressure sensor initialization

At sensor start-up the first command to be sent by the microcontroller is RESET (figure 3.7).

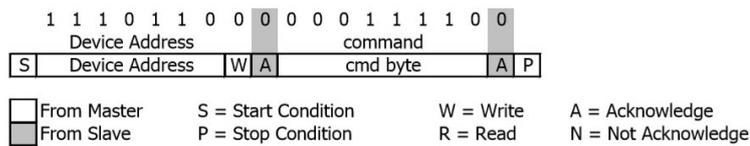


Figure 3.7: IIC Reset Command [14]

In idle conditions the lines of the IIC standard are at a high logic level. The start and the stop conditions and the acknowledge are periods of particular condition and are coded with a low logic level. In the first part of the command the peripheral address is sent. This command fails if the sensor does not respond forcing to a low level the SDA line in the acknowledge period.

When the sensor has been reset it is possible to request the transmission of the calibration constants. The constants are 16 bit coefficients. In the table 3.3 it can be noticed that the command of reading of the PROM has 3 undefined bits (XYZ). Those 3 bits are used to encode the request for a specific coefficient as shown in TABLE 3.4.

Cn	Description	Command (HEX)
C1	Pressure sensitivity	0xA2
C2	Pressure offset	0xA4
C3	Temperature coefficient of pressure sensitivity	0xA6
C4	Temperature coefficient of pressure offset	0xA8
C5	Reference temperature	0xAA
C6	Temperature coefficient of the temperature	0xAC

Table 3.4: Pressure sensor calibration coefficients

After the microcontroller has sent the command the sensor reads the value from its PROM and eventually is ready to transmit it on the SDA line. So the microcontroller writes on the SDA the sensor address and reads the value from the sensor. The IIC works transmitting packages of 8 bits, so for these coefficients are required two transmission periods (figure 3.8).

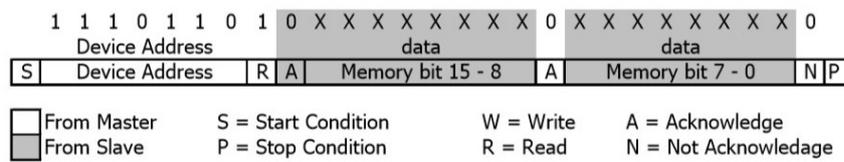


Figure 3.8: IIC Reading 16 bits sequence [14]

```

1 tx_buffer[0] = 0xA2;
2 HAL_I2C_Master_Transmit(&hi2c1, 0xEC, tx_buffer, 1, 1000);
3 HAL_I2C_Master_Receive(&hi2c1, 0xED, rx_buffer, 2, 1000); // recives 2 bytes of data
4 *C1 = (rx_buffer[0] << 8) | (rx_buffer[1]);
    
```

The 6 coefficients are stored on the microcontroller memory and ready to be used. This concludes the initialization phase of the sensor.

Second order compensation algorithm

The pressure sensor is equipped with a $\Sigma\Delta$ AD converter. The converter supports the selection of different levels of OSR (Over Sampling Rate). The best resolution is obtained with the maximum OSR of 8192. However, this type of reading is the slowest among all the available, this conversion can take up to 18.08 m.

Output values are coded on 24 bits. So in this case are required 3 transmission periods to receive the complete reading from the sensor. The command is a standard 8 bits command (Table 3.3). The receiving process is described in figure 3.9

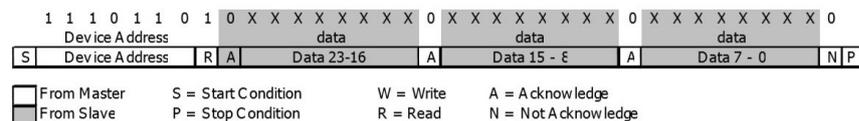


Figure 3.9: IIC Reading 24 bits sequence [14]

The algorithm was implemented in a FSM like structure using a switch statement. It is composed of 13 steps:

1. START PRESSURE CONV
2. READ PRESSURE
3. START TEMPERATURE CONV
4. READ TEMPERATURE
5. CALC TEMPERATURE
6. BRANCH LOW HIGH TEMPERATURE
7. HIGH TEMPERATURE
8. LOW TEMPERATURE
9. VERY LOW TEMPERATURE
10. DUMMY HIGH TEMPERATURE
11. DUMMY LOW TEMPERATURE
12. SECOND ORDER COMP
13. OUTPUT READY

The algorithm is activated by a microcontroller timer every 2.5 s, and at every activation it executes one step. To avoid problems related to the time required for an individual conversions, the sending phase of the conversion command and the reading phase have been separated into two different steps (states). The compensation process requires different operations depending on the ambient temperature. The different paths that the algorithm can follow have different number of steps. This is why some *dummy – state* have been inserted. These states with no operation inside are required to balance the execution time of the algorithm regardless of the ambient temperature. So every path is composed of 10 steps. In this way every 25 s under any environmental condition a pressure value is ready to be saved on the μ SD. This execution structure is useful for maintaining a time reference during field measurements.

The code is structured to be a simple function that is called in the `main()` in a transparent way. The `pressure_sensor()` function always returns -1 except if there is a pressure value ready. In that case the pressure value is returned. In this way the execution of the `main()` is completely blind to the execution of this algorithm, not knowing the precise state in which `pressure_sensor()` is. However the `main()` will be adapted to detect a new pressure value valid only when this function returns a value greater than zero.

Here are reported a diagram of the algorithm.

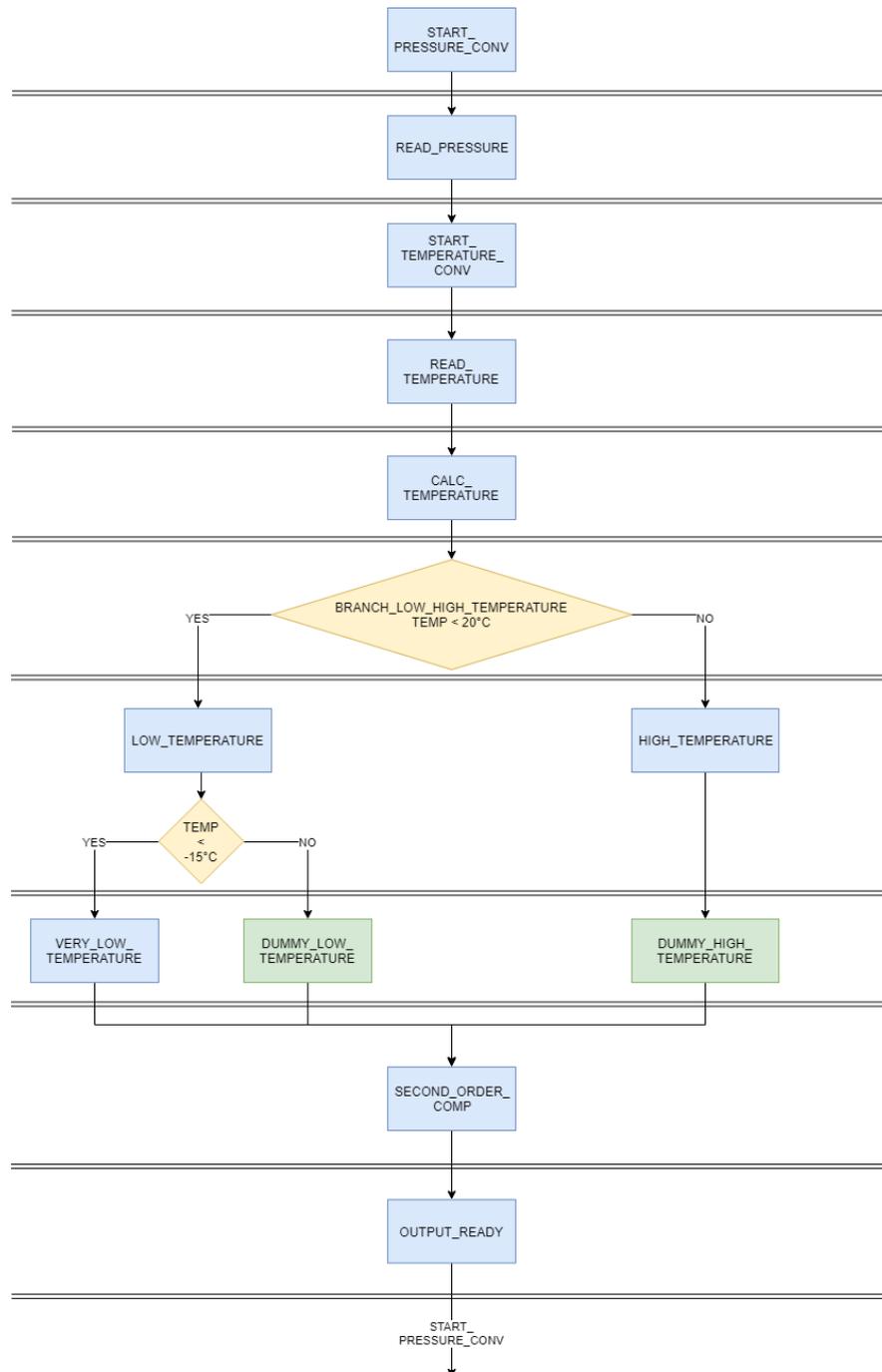


Figure 3.10: IIC Pressure Sensor Algorithm. Green: dummy states. Blue: std states. Yellow: decisional stages.

3.2.2 μ SD

On the board is installed SD card slot on its PCB and this slot is already connected to the SPI module. The connection diagram and the internal block scheme are reported here:

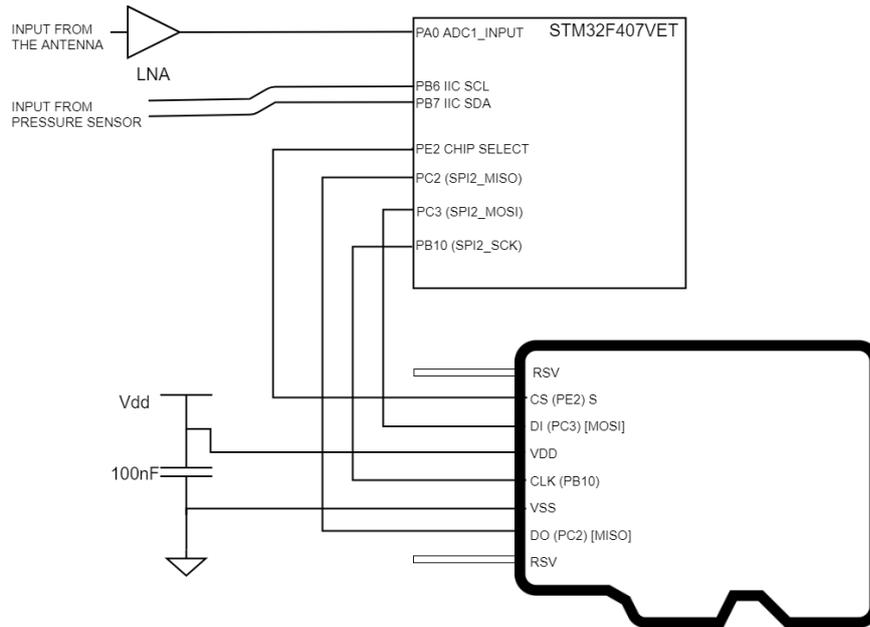


Figure 3.11: μ SD pin schematic

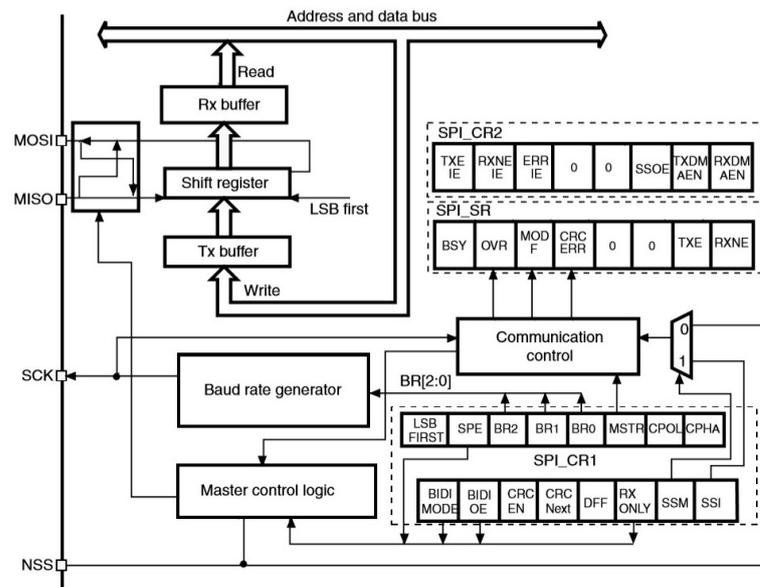


Figure 3.12: SPI Module block diagram [15]

Where pins are:

- MISO: Master In / Slave Out data. This pin can be used to transmit data in slave mode and receive data in master mode.
- MOSI: Master Out / Slave In data. This pin can be used to transmit data in master mode and receive data in slave mode.

- SCK: Serial Clock output for SPI masters and input for SPI slaves.
- NSS: Slave select. This is an optional pin to select a slave device. This pin acts as a ‘chip select’ to let the SPI master communicate with slaves individually and to avoid contention on the data lines

The μ SD card will receive a constant stream of data from the ADC, so it has been programmed to write data in large blocks of bytes. This mode of writing on flash memories is faster than repeated writing of small packets.

The writing function is called in the main loop of the receiver when a threshold of the ADC’s buffer is reached or by the main loop of the transmitter when a pressure value is ready. The function `write_SD()` receives two parameters. The first one is the pointer to the data that has to be written, the second one is the size in bytes of the data. After the writing process is terminated a check is performed. The function `f_write()` returns the number of written bytes and a status flag (`res`). If something has not worked properly the execution control system is passed to the ErrorHandler.

```

1 void write_SD(uint16_t *data, uint16_t NUMOFSAMPLES)
2 {
3
4     res = f_write(&MyFile, data, NUMOFSAMPLES, (void*)&byteswritten);
5
6
7     if((byteswritten == 0) || (res != FR_OK))
8         Error_Handler();
9
10 }
```

The stream of bits is not separated by any character, neither a "\n" nor a space. This choice was made for two reasons. The first is to save storage memory by reducing the amount of bits to be saved. The second is to speed up the data writing on the SD card to the maximum. At the end of the sampling the file present on the SD is not directly readable. The data is reworked by a software written in C language, which creates a dedicated text file for each group of 50 million samples, this is done to reduce the size of individual files and lighten the computational load of data analysis using a MATLAB algorithm. This aspect of the work will be discussed in details in the Software chapter.

3.2.3 Wet Contact and μ C Low Power modes

The device has 4 pins. The pins named PIN 0 and PIN1 are two pins exposed on the PCB. If these pins are connected by the presence of a low impedance conductive medium the output of the device changes state. The output of the device is `uC_WET_CONTACT`. This pin will be connected to an input pin of the microcontroller. In this way the microcontroller will be able to detect the immersion in sea water. The last pin `WET_CONTACT_ENABLE` is an enable pin that must be connected to the 3.3V source.

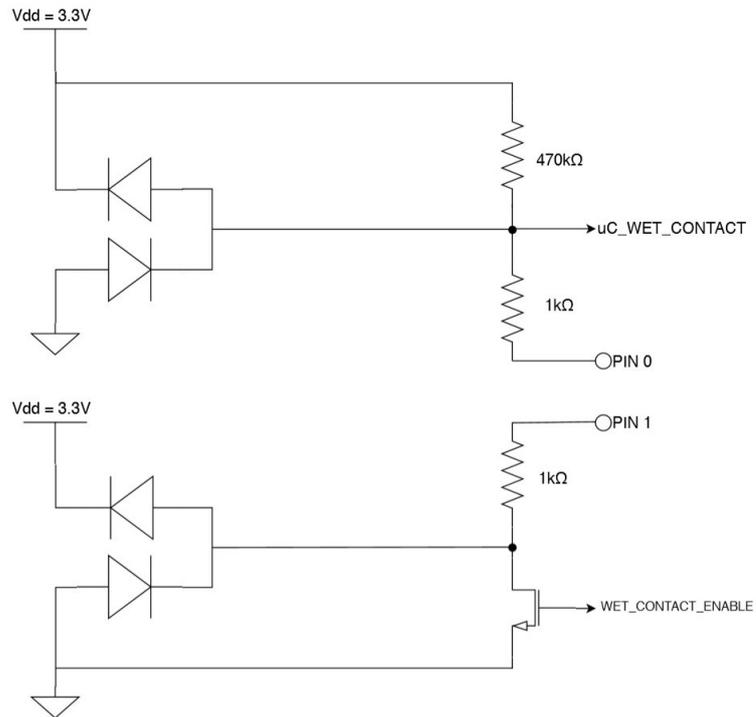


Figure 3.13: Wet Contact Circuitual Schematic [15]

When the system is powered up for the first time the initialization of the peripherals is started but they are not activated. The system then enters an energy saving mode.

The power supply scheme of the microcontroller is here reported Figure 3.14.

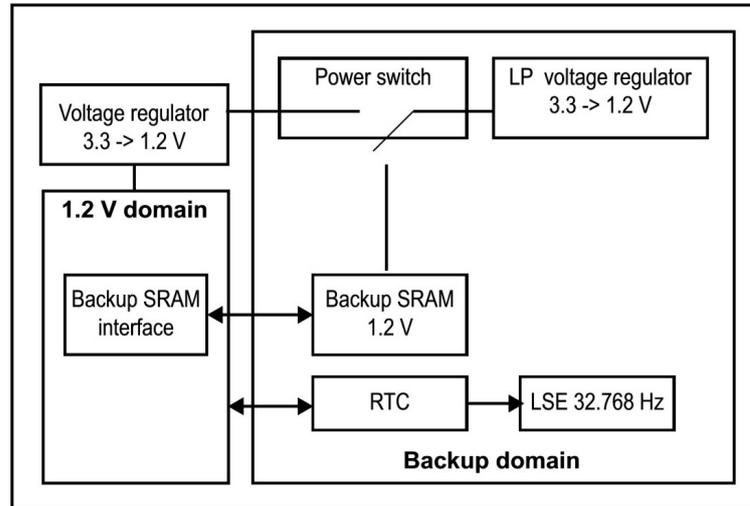


Figure 3.15: Backup domain diagram [15]

An embedded linear voltage regulator supplies all the digital circuitries except for the backup domain and the Standby circuitry. The regulator output voltage is around 1.2 V. When activated by software, the voltage regulator is always enabled after Reset. It works in three different modes depending on the application modes.

- In Run mode, the regulator supplies full power to the 1.2 V domain (core, memories and digital peripherals). In this mode, the regulator output voltage (around 1.2 V) can be scaled by software to different voltage values: Scale 1 or Scale 2 can be configured on the fly through VOS (bit 15 of the PWR_CR register). The voltage scaling allows optimizing the power consumption when the device is clocked below the maximum system frequency.
- In Stop mode, the main regulator or the low-power regulator supplies to the 1.2 V domain, thus preserving the content of registers and internal SRAM. The voltage regulator can be put either in main regulator mode (MR) or in low-power mode (LPR). The programmed voltage scale remains the same during Stop mode.
- In Standby mode, the regulator is powered down. The content of the registers and SRAM are lost except for the Standby circuitry and the backup domain.

The HAL driver library, supported by the microcontroller, has three low power modes.

- Sleep mode: Cortex-M4 core stopped, peripherals kept running
- Stop mode: all clocks are stopped, regulator running, regulator in low power mode
- Standby mode: 1.2 V domain powered off

Sleep mode

In Sleep mode, only the CPU is stopped. All peripherals continue to operate and can wake up the CPU when an interrupt/event occurs.

The Sleep mode is entered by using the HAL_PWR_EnterSLEEPMode (PWR_MAINREGULATOR_ON, PWR_SLEEPENTRY_WFx).

PWR_SLEEPENTRY_WFx function can be:

- `PWR_SLEEPENTRY_WFI`: enter SLEEP mode with WFI instruction. WFI will wake up the microcontroller when any interrupt occurs.
- `PWR_SLEEPENTRY_WFE`: enter SLEEP mode with WFE instruction. WFE will wake up the microcontroller when a wakeup event occurs.

The Regulator parameter is not used for the STM32F4 family and is kept as parameter just to maintain compatibility with the lower power families (STM32L). Any peripheral interrupt acknowledged by the nested vectored interrupt controller (NVIC) or event can wake up the device from Sleep mode. In Sleep mode, all I/O pins keep the same state as in Run mode. In Sleep mode, the systick is stopped to avoid exit from this mode with systick interrupt when used as time base for Timeout. This is the least power saving mode.

Stop mode

In Stop mode, all clocks in the 1.2 V domain are stopped, the PLL, the HSI, and the HSE RC oscillators are disabled. Internal SRAM and register contents are preserved. The voltage regulator can be configured either in normal or low-power mode. To minimize the consumption in Stop mode, FLASH can be powered off before entering the Stop mode using the `HAL_PWREx_EnableFlashPowerDown()` function. It can be switched on again by software after exiting the Stop mode using the `HAL_PWREx_DisableFlashPowerDown()` function.

The Stop mode is entered using the `HAL_PWR_EnterSTOPMode(REGULATOR, WfX)` function. `WfX` sets if the Stop mode is entered with WFI or WFE instruction. `REGULATOR` parameter specifies the regulator state in Stop mode.

This parameter can be one of the following values:

- `PWR_MAINREGULATOR_ON`: Stop mode with regulator ON
- `PWR_LOWPOWERREGULATOR_ON`: Stop mode with low power regulator ON

In Stop mode, all I/O pins keep the same state as in Run mode. When the voltage regulator operates in low power mode, an additional startup delay is incurred when waking up from Stop mode. By keeping the internal regulator ON during Stop mode, the consumption is higher although the startup time is reduced. When exiting Stop mode by issuing an interrupt or a wake-up event, the HSI RC oscillator is selected as system clock. For the project the HSE clock was chosen, when the microcontroller exits the low-power mode it is necessary to consider an additional delay to re-configure the clock.

Any EXTI Line (Internal or External) configured in Interrupt/Event mode can wakeup the microcontroller and its peripherals.

Standby mode

The Standby mode is used to achieve the lowest power consumption. It is based on the Cortex-M4 deep sleep mode. The internal voltage regulator is switched off so that the entire 1.2 V domain is powered off. The PLL, the HSI RC and the HSE crystal oscillators are also switched off. After entering Standby mode, the SRAM and register contents are lost except for registers in the backup domain and the backup SRAM when selected. The device exits the Standby mode when an external reset (NRST pin), an IWDG reset, a rising edge on the WKUP pin, or an RTC alarm/wakeup/tamper/time stamp event occurs. The standby mode is not supported when the embedded voltage regulator is bypassed and the 1.2 V domain is controlled by an external power.

The Standby mode is entered using the `HAL_PWR_EnterSTANDBYMode()` function.

The microcontroller can exit from the standby mode using WKUP pin rising edge, RTC alarm (Alarm A and Alarm B), RTC wake-up, tamper event, time-stamp event, external reset in NRST pin, IWDG

reset.

In Standby mode, all I/O pins are high impedance except for: Reset pad (still available) RTC_AF1 pin (PC13) if configured for the output of the RTC alarm or calibration clock. WKUP pin 1 (PA0) if enabled.

3.2.4 Current absorption

Depending on the mode in which the microcontroller is located the maximum current absorption from the power supply network varies significantly. The current absorption is mainly related to the external clock of all peripherals $f_{HCLK} = 168 \text{ MHz}$.

Mode	Current I_{DD}
Run	93 mA
Sleep peripherals enabled	77 mA
Sleep peripherals disabled	27 mA
Stop with LP Regulator OFF	0.45 mA
Stop with LP Regulator ON	0.31 mA
Standby	4 μA

Table 3.5: Current related to μC modes

The current consumption related to the peripherals is proportional to the frequency of the domain that includes them. Both the APB1 domain and the APB2 domain are clocked at 42 MHz.

Domain	Peripheral	Current
APB2	ADC1	$1.6 \text{ mA} + 4.67 \mu\text{A}/\text{MHz}$
APB1	TIM2	$16.71 \mu\text{A}/\text{MHz}$
APB1	TIM4	$13.45 \mu\text{A}/\text{MHz}$
APB1	TIM6	$2.43 \mu\text{A}/\text{MHz}$
APB1	DAC	$1.67 \mu\text{A}/\text{MHz}$
APB1	IIC	$2.67 \mu\text{A}/\text{MHz}$

Table 3.6: Current related to μC peripherals

The μSD card following the datasheet's tables should absorb less than 100 mA even during the write operation [20].

This information will be useful for the choice of the 3.3 V switching regulator which will power all this devices.

CHAPTER 4

Hardware Design

4.1 Transmitter and 5 V domain

This hardware was previously designed by other colleagues, so in this section will be reported the final design in order to verify the design choices.

4.1.1 Power Supply for 5 V domain

Since the battery voltage is lower than 5 V a boost switching converter is required. The boost converter of choice is the Texas Instruments TPS61032 [18].

This converter outputs a fixed 5 V voltage and a maximum current above 2 A with an efficiency above 90%. The LBO control pin allows to check the state of the battery [18].

The suggested value for R_{15} is 390 k Ω . The value of R16 has been evaluated following the relation on the datasheet:

$$R_{16} = R_{15} \left(\frac{V_{bat}}{0.5 \text{ V}} - 1 \right) = 2.2 \text{ M}\Omega \quad (4.1)$$

The maximum inductor current can be calculated taking into account the maximum output current limited to 2A.

$$I_{inductor} = I_{out} \frac{V_{out}}{0.8 * V_{bat}} = 3.8 \text{ A} \quad (4.2)$$

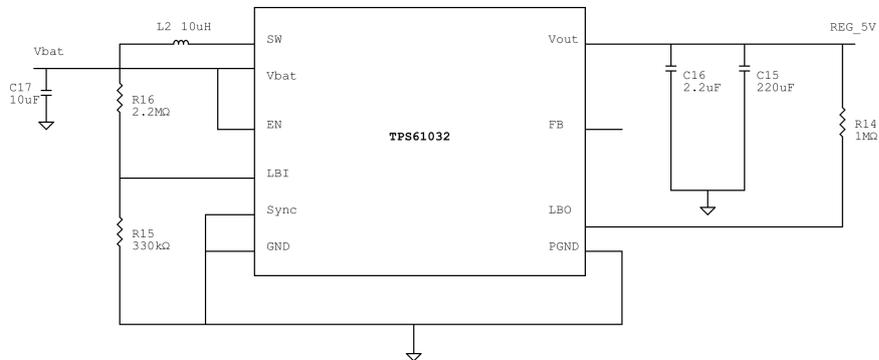


Figure 4.1: TPS61032 schematic

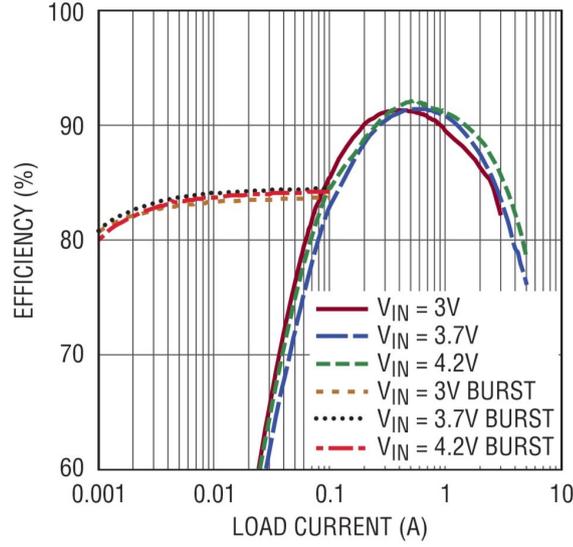


Figure 4.4: LTC 3113 efficiency vs Load Current [13]

Between the R_t pin and ground must be placed a resistor. The value of this resistor sets the switching frequency of the converter with the following equation.

$$f = \frac{90}{R_t[k\Omega]} [MHz] \quad (4.5)$$

The frequency range allowed is between 300 kHz and 2 MHz, the resistor was chosen equal to 100 k Ω to get a switching frequency of 900 kHz or the double-pole response of the output filter.

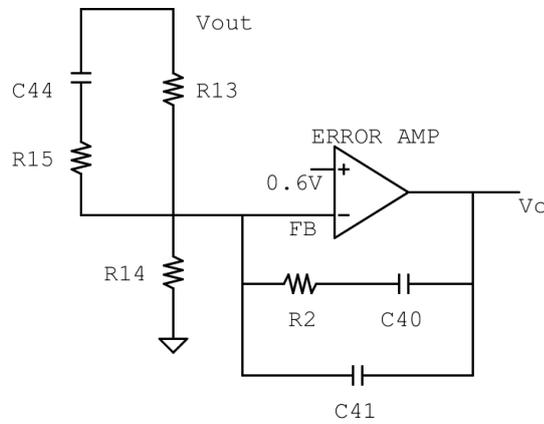


Figure 4.5: Error Amplifier with type III compensation

$$f_{filter_pole} = \frac{1}{2\pi\sqrt{LC_{out}}} = 5 \text{ kHz} \quad (4.6)$$

$$f_{pole1} = \frac{1}{10^5 * 2\pi R_{13} C_{40}} = 2.85 \text{ mHz} \quad (4.7)$$

$$f_{zero1} = \frac{1}{2\pi R_2 C_{40}} = 2.85 \text{ kHz} \quad (4.8)$$

$$f_{zero2} = \frac{1}{2\pi R_{13}C_{42}} = 1.06 \text{ kHz} \quad (4.9)$$

$$f_{pole2} = \frac{1}{2\pi R_2C_{41}} = 234.05 \text{ kHz} \quad (4.10)$$

$$f_{pole3} = \frac{1}{2\pi R_{15}C_{42}} = 82.34 \text{ kHz} \quad (4.11)$$

To FB is also connected to a voltage divider. This pin is the feedback voltage for the buck-boost converter. The buck-boost output voltage is given by the following equation.

$$V_{out} = 0.6 * \left(1 + \frac{R2}{R1}\right) [V] \quad (4.12)$$

The buck-boost converter has two current limit circuits. The primary current limit is an average current limit circuit which sources current into FB to reduce the output voltage. For this current limit feature to be effective, the Thevenin resistance from FB to ground should exceed 100 k Ω . Therefore R2 was chosen equal to 680 k Ω and R1 equal to 150 k Ω such that $R_{eq} = 121.9$ k Ω . The output voltage is set to 3.32 V.

The IC output voltage transient at start-up time is here reported.

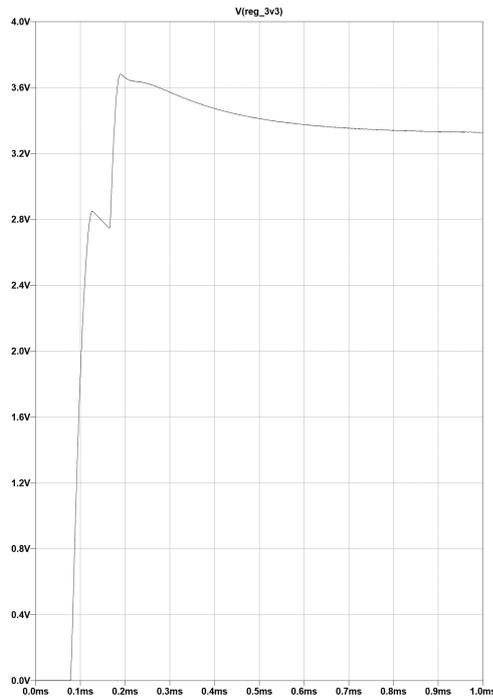


Figure 4.6: LTC 3113 startup transient on LTSpice $V_{in} = 1.8$ V with a 10 Ω load

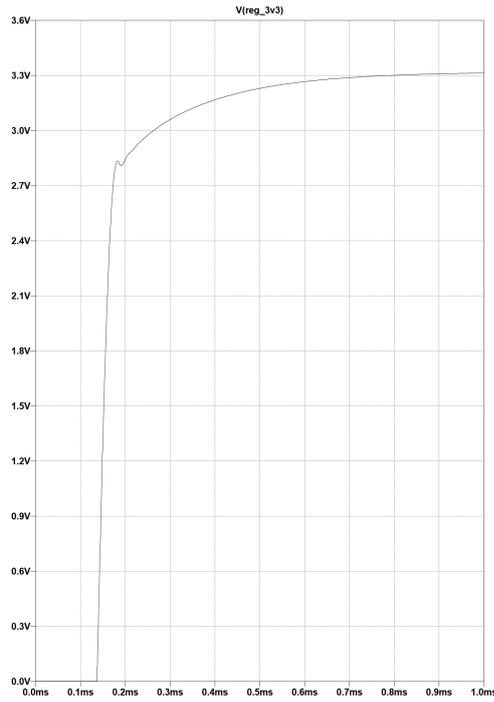


Figure 4.7: LTC 3113 startup transient on LTSpice $V_{in} = 3.5\text{ V}$ with a $10\ \Omega$ load

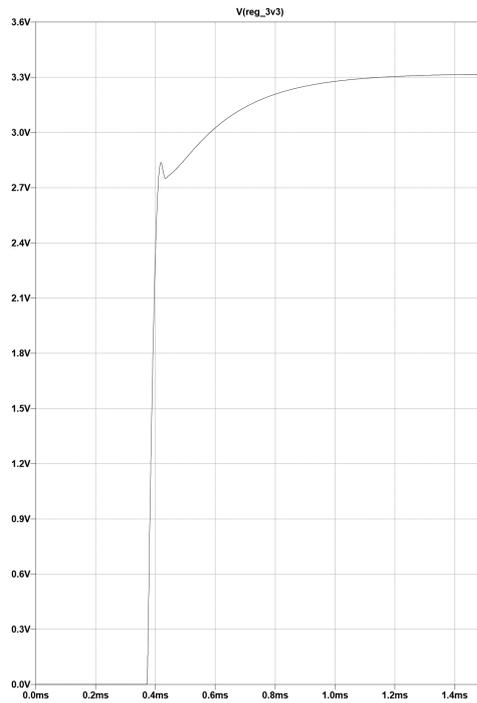


Figure 4.8: LTC 3113 startup transient on LTSpice $V_{in} = 5.5\text{ V}$ with a $10\ \Omega$ load

The two last components to be chosen are the inductor and the output capacitor. The peak-to-peak inductor current ripple can be calculated from the following formula, where f is the frequency in MHz and L is the inductance in μH :

$$\Delta I_{peak-to-peak} = \frac{V_{out}}{f * L} * \left(1 - \frac{V_{out}}{V_{in}}\right) \quad (4.13)$$

To ensure operation without triggering the reverse current comparator under no load conditions it is recommended that the peak-to-peak inductor ripple not exceed 800 mA taking into account the maximum reverse current limit of -400 mA. Utilizing this recommendation for applications operating at a switching frequency of 300 kHz requires a minimum inductance of 6.8 μ F, similarly an application operation at a frequency of 2 MHz would require a minimum of 1 μ F. Since the frequency is set to 900kHz it has been chosen a 10 μ F 3.9 A inductor. The output capacitor C_{out} is used to minimize the output voltage ripple. The capacitor should be chosen large enough to reduce the output voltage ripple to acceptable levels. Neglecting the capacitor ESR and ESL, the peak-to-peak output voltage ripple can be calculated by the following formulas, where f is the frequency in MHz, C_{out} is the capacitance in μ F, L is the inductance in μ H, V_{in} is the input voltage in volts, V_{out} is the output voltage in volts. ΔV_{ptp} is the output ripple in volts and I_{load} is the output current in amps.

$$C_{out} \geq \frac{1}{\Delta V_{peak-to-peak} * 8 * L * f^2} * \frac{(V_{in} - V_{out}) * V_{out}}{V_{in}} [\mu F] \quad (4.14)$$

With a ripple lower than 10 mV the minimum value for C_{out} is 82 μ F. The selected value is 100 μ F.

4.3 Receiver

The signal received by the device is low voltage and very subject to noise. A narrow band bandpass filter with a high gain and with an output dynamic compatible with the dynamics of the analogue to digital acquisition system has therefore been designed. The component selected for the built of the circuit is the AD8618 from Analog Devices. The AD8618 is a quad, rail-to-rail, single-supply amplifiers featuring very low offset voltage, wide signal bandwidth up to 24 MHz. The input voltage noise density is 7 nV/ $\sqrt{\text{Hz}}$ at 10 kHz and the current noise density is 50 pA/ $\sqrt{\text{Hz}}$ [16].

4.3.1 Circuit design

The reference circuit is a multifeedback cell shown in the figure 4.9. The transfer function is reported in equation 4.15 where the Y_n are the admittances of each component.

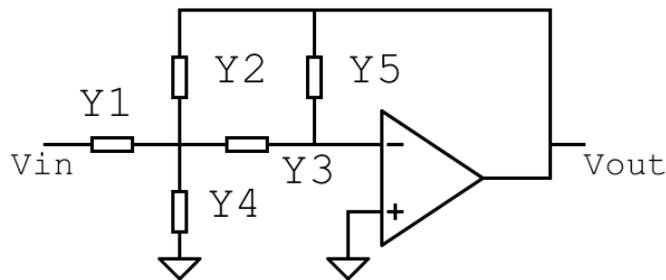


Figure 4.9: Multifeedback cell

$$\frac{V_{out}}{V_{in}} = -\frac{Y_1 Y_3}{Y_5(Y_1 + Y_2 + Y_3 + Y_4) + Y_2 Y_3} \quad (4.15)$$

The generic transfer function of a second order filter is:

$$H_{bandpass} = H_0 * \frac{\frac{\omega_0}{Q} * s}{s^2 + \frac{\omega_0}{Q} * s + \omega_0^2} \quad (4.16)$$

So to match the required behaviour components are selected in this way.

- $Y_1 = 1/R_1$
- $Y_2 = sC_2$
- $Y_3 = sC_3$
- $Y_4 = 0$
- $Y_5 = 1/R_5$

The final transfer function is here reported.

$$\frac{V_{out}}{V_{in}} = -\frac{sC_3R_5}{s^2C_2C_3R_1R_5 + sR_1(C_2 + C_3) + R_1} \quad (4.17)$$

To simplify the calculation of the parameters and the realization of the circuit we have chosen to select $C_2 = C_3 = C$. The key parameters of the cell: gain in band, central frequency and factor of merit can be obtained using the following equations.

$$H_0 = \frac{R_5}{2R_1} = 2Q^2 \quad (4.18)$$

$$f_0 = \frac{1}{2\pi C\sqrt{R_1R_5}} \quad (4.19)$$

$$Q = \frac{1}{2}\sqrt{\frac{R_5}{R_1}} \quad (4.20)$$

Only E12 series resistors and E6 series capacitors will be used for the construction of the circuit. The complete circuit consists of 4 MFB cells in cascade.

4.3.2 Stage analysis

First Stage

The first cell is connected to the antenna, so it has a differential input structure. In figure 4.10 is also reported the power supply organization. The main source is REG_3V3, the output of the DC-DC converter LTC3113. Also a voltage divider has been created to get Vpol. This voltage will be used in the other cells to translate the output signal in middle of the ADC dynamic.

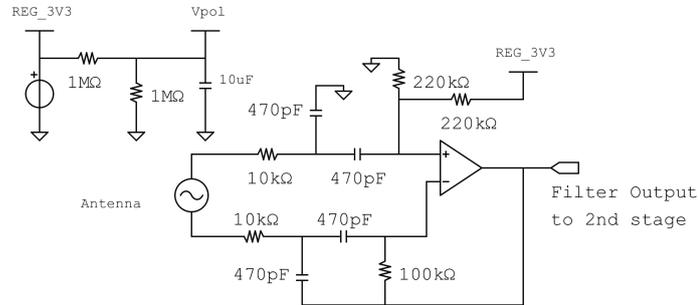


Figure 4.10: Schematic of the first amplifier

- $H_0 = 5$
- $f_0 = 10.7 \text{ kHz}$
- $Q = 1.58$
- $\text{phase margin} = 72.5^\circ$

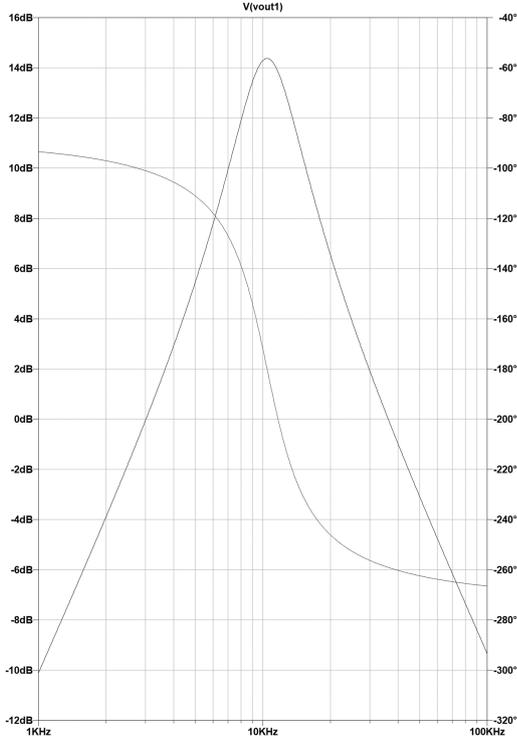


Figure 4.11: Gain of the first stage

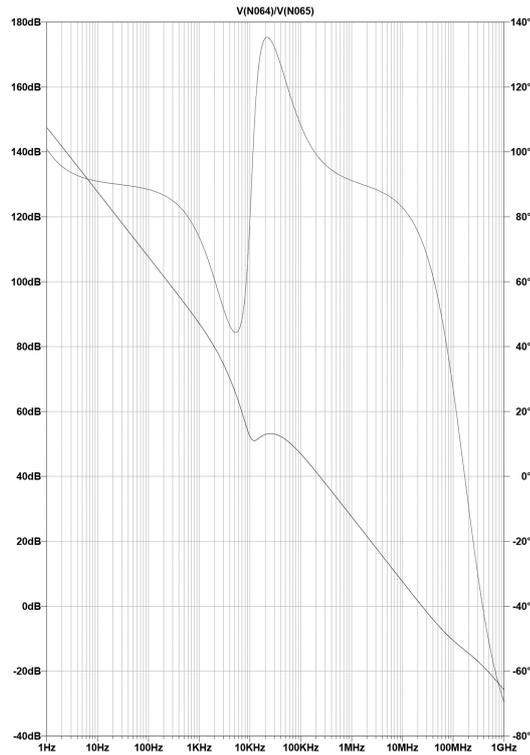


Figure 4.12: Closed loop gain of the first stage

Second Stage

The second stage has the main purpose of filtering out-of-band noise. The in band gain is just over 1 time.

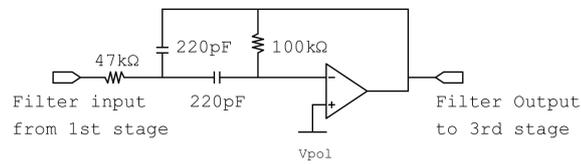


Figure 4.13: Schematic of the second amplifier

- $H_0 = 1.06$
- $f_0 = 10.55 \text{ kHz}$
- $Q = 0.73$
- $\text{phase margin} = 72^\circ$

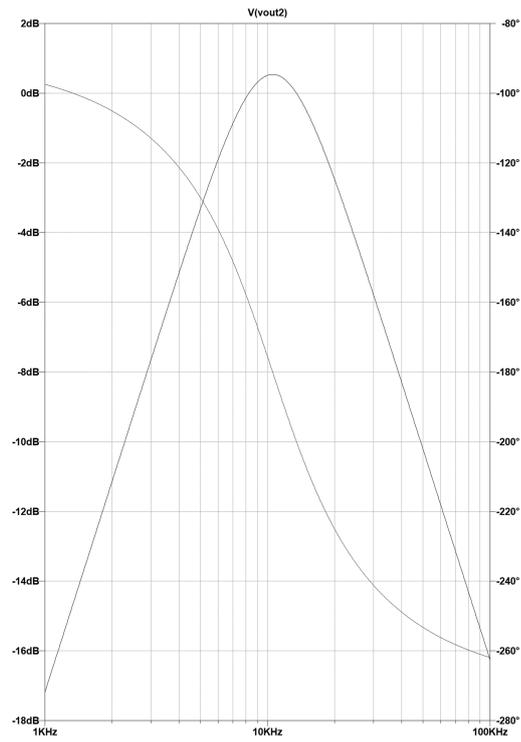


Figure 4.14: Gain of the second stage

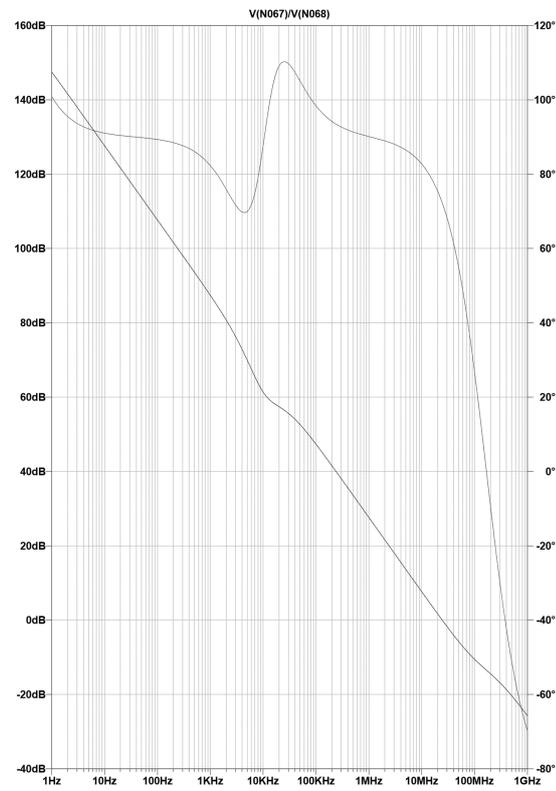


Figure 4.15: Closed loop gain of the second stage

Third Stage

This stage has the higher gain. The GBW of the AD8618 is 24 MHz so the maximum achievable gain for a 10 kHz signal amplification can be calculated assuming that the bandwidth of the circuit has to be at least one decade higher. However, it was decided not to achieve such a high gain in a single cell to obtain a cascade of cells with a narrower response possible.

$$G_{max} = \frac{GBW}{BW} = \frac{24 \text{ MHz}}{100 \text{ kHz}} = 240 \text{ times} = 47.6 \text{ dB} \quad (4.21)$$

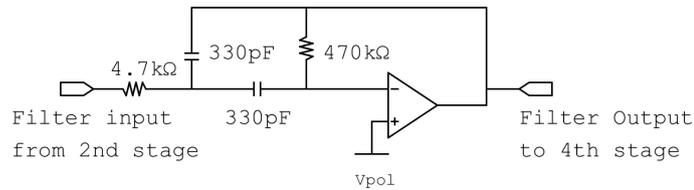


Figure 4.16: Schematic of the third amplifier

- $H_0 = 50$
- $f_0 = 10.26 \text{ kHz}$
- $Q = 5$
- $\text{phase margin} = 72.8^\circ$

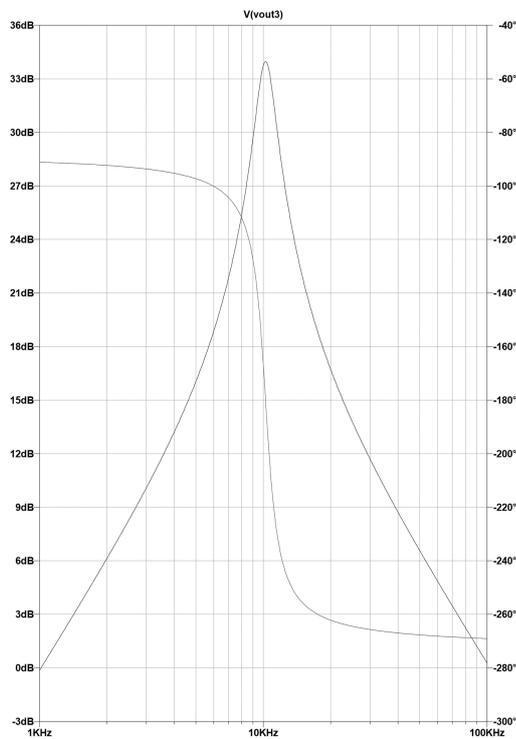


Figure 4.17: Gain of the third stage

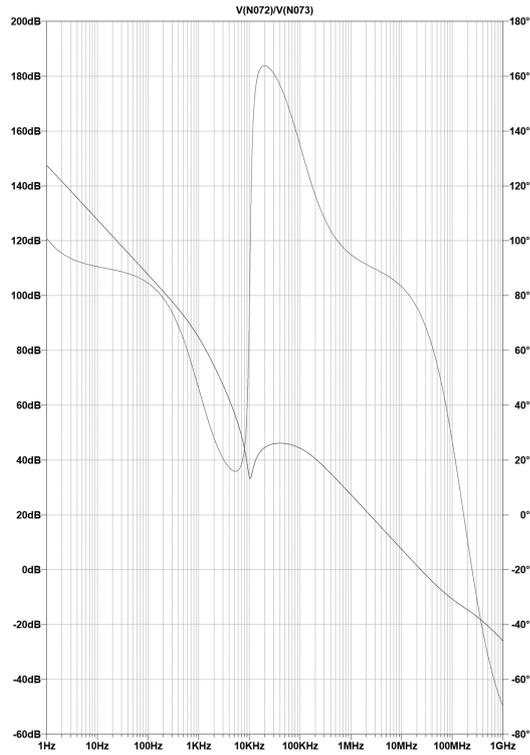


Figure 4.18: Closed loop gain of the third stage

Fourth Stage

The last cell has a lower gain, but sufficient to reach the 60 dB goal.

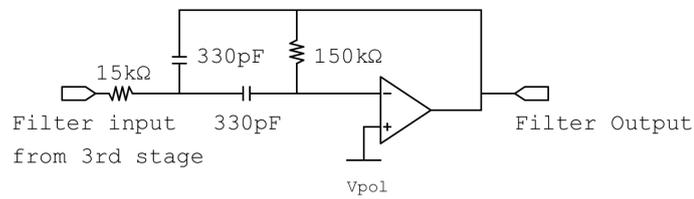


Figure 4.19: Schematic of the fourth amplifier

- $H_0 = 5$
- $f_0 = 10.17 \text{ kHz}$
- $Q = 1.58$
- $phase \ margin = 72.35^\circ$

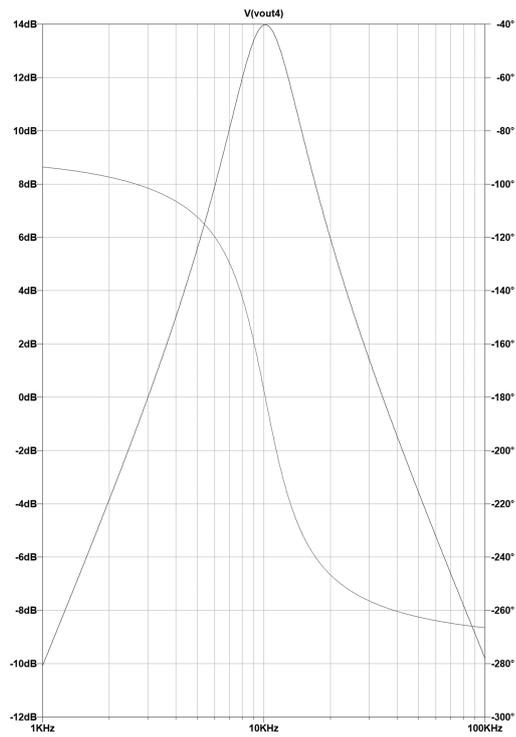


Figure 4.20: Gain of the fourth stage

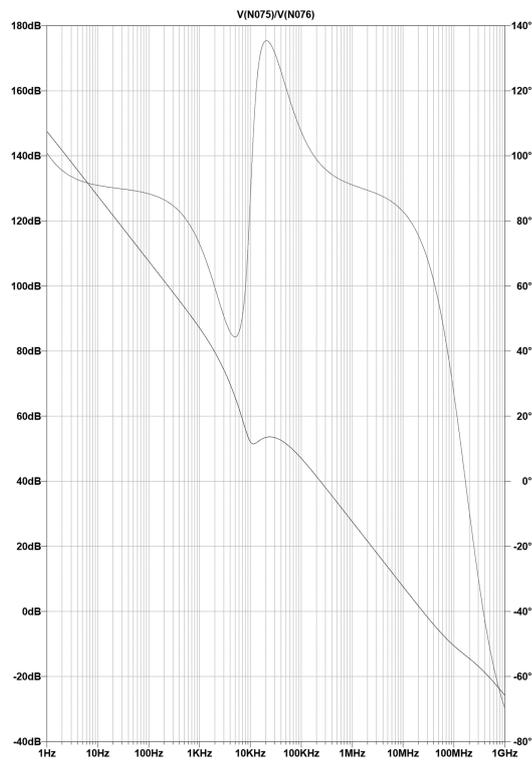


Figure 4.21: Closed loop gain of the fourth stage

4.3.3 Complete filter

The stability of the individual cells was verified by measuring the phase margin [17]. Each cell showed in figures 4.12 4.15 4.18 4.21 a margin of at least 72° . The complete filter has a peak gain of 62.85 dB at 10.25 kHz. The -3 dB bandwidth is 1.7 kHz. The gain at 10 kHz is 62.52 dB.

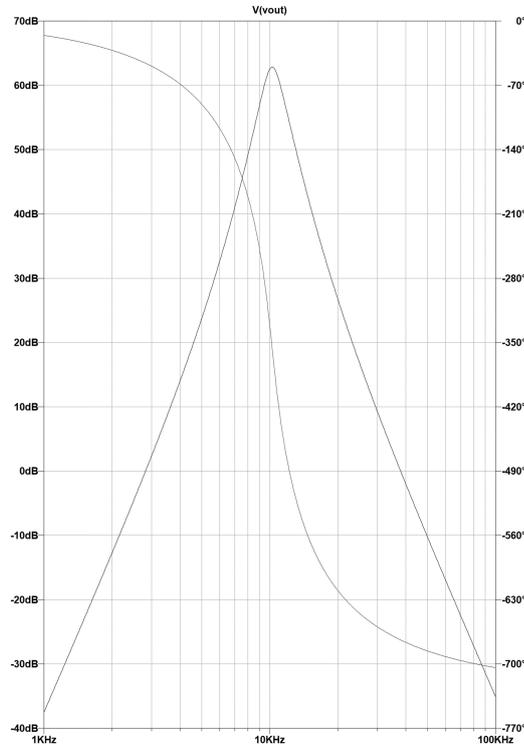


Figure 4.22: Gain of the complete filter

The filter response has been tested for an input sinusoidal waveform of 1 mV amplitude. The frequency of the input signal was selected at 1 kHz in figure 4.23 then at 100 kHz in figure 4.25 to test the filter response with out-of-band signals. In figure 4.24 the selected frequency is 10 kHz to test the filter in band settling time. This parameter could be critical when using a fast OOK coding.

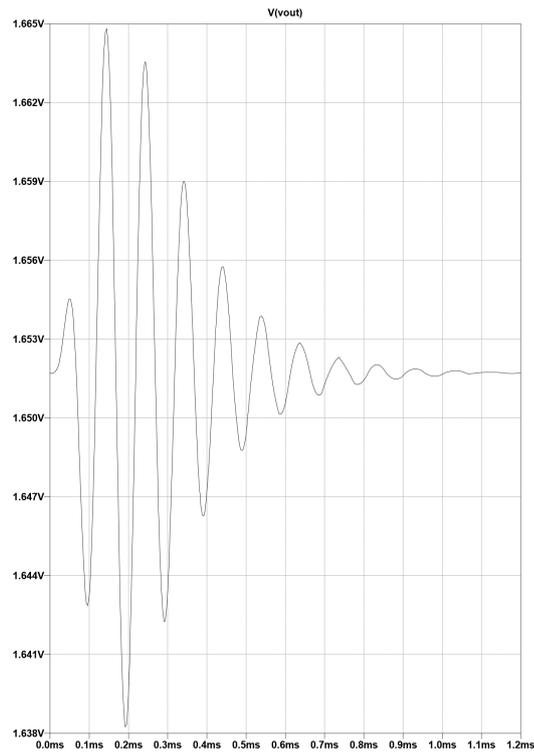


Figure 4.23: Filter response to a 1 kHz 1 mV amplitude sine wave

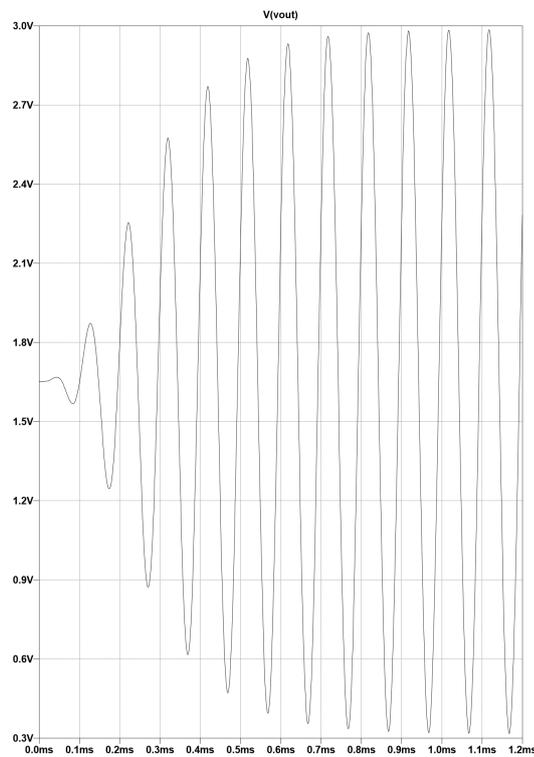


Figure 4.24: Filter response to a 10 kHz 1 mV amplitude sine wave

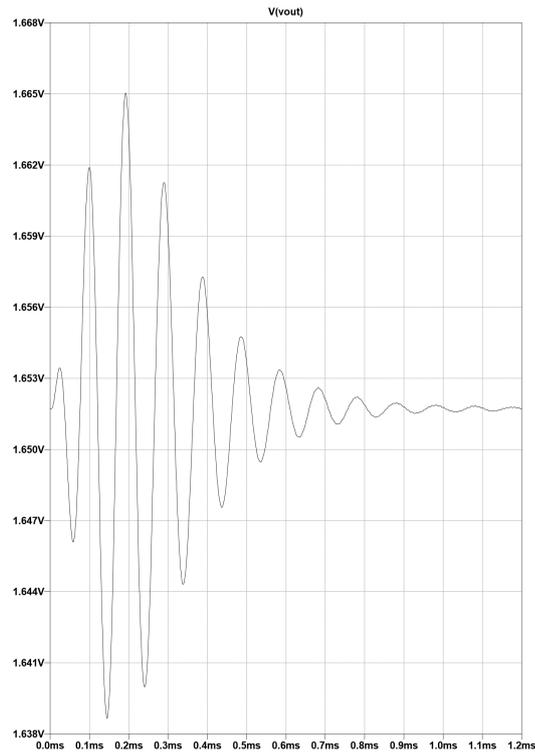


Figure 4.25: Filter response to a 100 kHz 1 mV amplitude sine wave

The output noise of the filter has been simulated too.

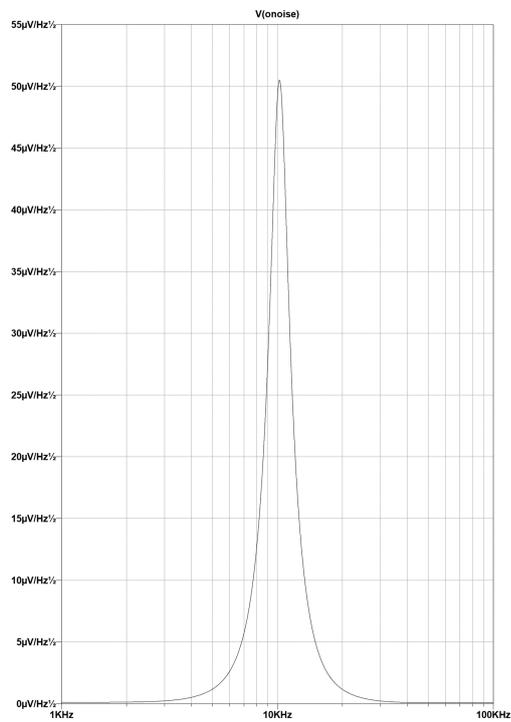


Figure 4.26: Output noise of the filter

4.3.4 Receiver Sensitivity

The minimum signal detectable by the system is related to the accuracy of the ADC. The STM32 microcontroller is equipped with a 12bit ADC and it is powered with the 3.3V line. So the voltage related to 1 LSB is:

$$V_{LSB} = \frac{V_{ref}}{2^{12} - 1} = \frac{3.3 \text{ V}}{4095} = 0.81 \text{ mV} \quad (4.22)$$

This voltage can be divided by the gain of the analog receiver in order to get the minimum voltage detected by the antenna which is actually capable of modify the output of the ADC.

$$V_{antenna} = \frac{V_{LSB}}{Gain} = \frac{0.81 \text{ mV}}{10^{\frac{62.85 \text{ dB}}{20}}} = 0.6 \mu\text{V} \quad (4.23)$$

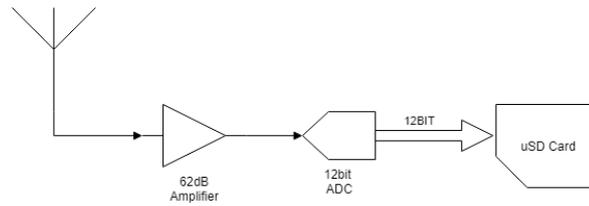


Figure 4.27: Receiver chain from analog to digital storage

CHAPTER 5

Analysis Software

The file generated by the receiving system includes the values sampled by the ADC and the values of the pressure sensor. The file generated by the transmitter includes only pressure values. Since the data is written as bit streams not spaced out by anything, both files are not directly readable. First it is necessary to separate the individual samples so that they can be analyzed.

5.1 Samples Extraction

The first part of software dedicated to the analysis of samples is a C program. The program reads the raw file coming from the communication system and creates up to 8 text files. The first 4 contain the ADC samples. These files contain a maximum of 250 million samples for reasons of file size. The remaining 4 files contain the pressure values. These values before being written on the μ SD are increased by 4096 in order to make them easily distinguishable from the ADC samples. The C program recognizes values greater than 4096 (outside the 12bit range) and inserts them in ad hoc files. The temporal correlation between samples of the ADC and pressure values is maintained since the files 1, 2, 3 and 4 of both concern the same sets of values saved on the μ SD. These files will be combined and plotted by a MATLAB script.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 #define sam 150000000
6 #define SIZE_FILE_LIMIT 25000000
7
8 short samples[sam];
9
10
11 int main()
12 {
13     long int i;
14     FILE *fin;
15     FILE *foutadc1, *foutadc2, *foutadc3, *foutadc4;
16     FILE *foutpress;
17
18     fin = fopen("samples.txt", "rb");
19
20     fread(samples, sizeof(samples), 1, fin);
21
22     fclose(fin);
23
```

```

24  foutadc1 = fopen("out_adcsamples1.txt","wt");
25  foutadc2 = fopen("out_adcsamples2.txt","wt");
26  foutadc3 = fopen("out_adcsamples3.txt","wt");
27  foutadc4 = fopen("out_adcsamples4.txt","wt");
28
29  foutpress = fopen("out_senspressures.txt", "wt");
30
31  for(i = 0; i < sam; i++)
32  {
33      if (samples[i] > 4096)
34      {
35          fprintf(foutpress, "%d\n", samples[i] - 4096);
36          printf("index %d : %d\n", i, samples[i] - 4096);
37      }
38
39      else
40      {
41          if (i < SIZE_FILE_LIMIT)
42              fprintf(foutadc1, "%d\n", samples[i]);
43
44          else if ((i >= SIZE_FILE_LIMIT) && (i < 2*SIZE_FILE_LIMIT))
45              fprintf(foutadc2, "%d\n", samples[i]);
46
47          else if ((i >= 2*SIZE_FILE_LIMIT) && (i < 3*SIZE_FILE_LIMIT))
48              fprintf(foutadc3, "%d\n", samples[i]);
49
50          else
51              fprintf(foutadc4, "%d\n", samples[i]);
52      }
53  }
54
55  fclose(foutadc1);
56  fclose(foutadc2);
57  fclose(foutadc3);
58  fclose(foutadc4);
59
60  fclose(foutpress);
61
62
63
64
65  return 0;
66 }

```

5.2 Data Analysis

5.2.1 Averaging Function

The averaging function receives the data file as the only parameter and returns an array with the data averaged according to the following algorithm. The realization of this analysis algorithm was necessary given the large amount of samples that are generated. The direct analysis of these samples is very difficult and heavy from the point of view of computational load. It was therefore necessary to find a sample decimation device that was not going to reduce the quality of the data but to accentuate the feature currently being analyzed or the amplitude of the received signal.

The first action performed by the function is to apply an interpolation on the ADC samples using the PCHIP method to smooth the sampled signal in order to reduce noise and spikes. This method reconstructs data in exactly the same way as a SPLINE, except that the slopes are chosen differently. SPLINE is more accurate if the data are values of a smooth function. PCHIP has no overshoots and

less oscillation if the data are not smooth. PCHIP is less expensive to run [21]. Since it is mandatory to keep the original amplitude of the sampled signal the PCHIP method is suitable for this purpose. After that the function generates two arrays. In these arrays are stored local minimums and local maximums of the received signal. In the for cycle a sample is read and compared to the previous one. If there the i -th sample is greater than the $(i-1)$ -th one the slope is positive, so the i -th sample is saved in a temporary variable. When an inversion of slope is detected the last sample stored in the temporary variable is a local maximum. The algorithm for the minimum detection is specular to this one.

The following section is used to calculate the actual size of the arrays of local minimums and maximums. These arrays are initialized with a size equal to that of the input file. It is therefore necessary to determine how many memory locations have actually been occupied by the previous piece of code. Then is filled the amplitude array following this equation for every i -th location.

$$amplitude(i) = \frac{max(i) - min(i)}{4096} * 100 \quad (5.1)$$

In this way we obtain the amplitude of each single sampled sine wave period in percentage from 0 to 100.

The last operation performed by the function is to fill the return array with the average of the amplitudes. The amplitude array is accessed in windows of 400 values and from every window is generated the mean value of those amplitudes. This operation is the strongest reduction samples' number of the algorithm.

The function is here reported.

```

1 function avg_of_amplitude = exe_avg(file)
2
3 [rows, cols] = size(file);
4
5 i = 1;
6 j = 1;
7 k = 1;
8
9 xspace_mult = 10;
10
11 linspace_of_samples = linspace(1, rows, xspace_mult*rows);
12
13 interped = interp1(linspace(1, rows, rows), file, linspace_of_samples, 'pchip');
14
15 % nearest
16 % makima
17 % pchip
18 % spline
19 % linear
20
21 % figure(1)
22 % plot(file, ':. ')
23 % hold on
24 % plot(linspace_of_samples, interped, 'red')
25 % xlim([10000000 10003000])
26 % grid on
27
28 new_rise = 0;
29 new_fall = 0;
30
31 min = 0;
32 max = 0;
33
34 min_array = zeros(1, xspace_mult*rows+1);
35 max_array = zeros(1, xspace_mult*rows+1);

```

```

36
37 for i = 2:xspace_mult*rows
38
39     if (interped(i-1) < interped(i))
40
41         new_rise = 1;
42         max = interped(i);
43         if new_fall == 1
44
45             min_array(k) = min; %% save local min when slope inversion
46             k = k+1; %% update index of mins
47             new_fall = 0;
48
49         end
50
51     end
52
53
54     if (interped(i-1) > interped(i))
55
56         new_fall = 1;
57         min = interped(i);
58         if new_rise == 1
59
60             max_array(j) = max; %% save local max when slope inversion
61             j = j+1; %% update index of max
62             new_rise = 0;
63
64         end
65
66     end
67
68 end
69
70 k = 0;
71 num_of_max = 0;
72 num_of_min = 0;
73
74 for i = 1:xspace_mult*rows
75
76     if ((max_array(i) == 0) && (k == 0))
77
78         num_of_max = i;
79         k = 1;
80
81     end
82
83 end
84
85 temp_max = max_array(1, 1:num_of_max);
86 temp_min = min_array(1, 1:num_of_max);
87
88 % figure(2)
89 % plot(temp_max, '.')
90 % hold on
91 % title('Graph MAX')
92 % grid on
93 %
94 % figure(3)
95 % plot(temp_min, '.')
96 % hold on
97 % title('Graph MIN')

```

```

98 % grid on
99
100 amplitude_array = zeros(1, num_of_max+1);
101
102 for i = 1:num_of_max
103     amplitude_array(i) = ((max_array(i)) - (min_array(i) + 1))/40.96;
104
105
106 end
107
108 % figure(4)
109 % plot (amplitude_array)
110 % hold on
111 % title('Graph Amplitude')
112 % xlabel('Time [s]')
113 % ylabel('Amplitude in percentage')
114 % grid on
115
116 i = 0;
117 k = 1;
118 j = 0;
119
120
121 avg_period = 400;
122
123 temp_ampl = 0; %zeros(1, avg_period);
124 avg_of_amplitude = zeros(1, floor(num_of_max/avg_period));
125
126 for i = 1:floor(num_of_max/avg_period)
127
128     temp_ampl = amplitude_array(1, k:(avg_period-1+k));
129
130     k = k + avg_period;
131
132     avg_of_amplitude(i) = mean(temp_ampl);
133
134 end
135
136 end

```

5.2.2 Main Script

The main script takes care of the opening of the files, then the averaging function is invoked. The main script receives the array of averaged amplitudes and calculates some parameters in order to correctly represent the final values on plot which has the x axis in seconds. Taking into account the sampling frequency of the ADC, the decimation algorithm of the function now 24 values correspond to 1 second of sampling. On the output figure are also plotted the pressure values. These values are not modified by any means in the algorithm.

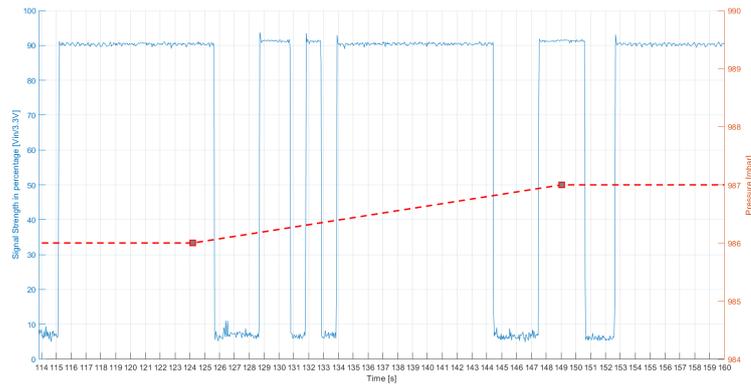


Figure 5.1: Output plot of the MATLAB script. In blue the amplitude in percentage and in red the pressure in mbar separated by 25 s

```

1  clc
2  clear all
3  close all
4
5  % OPEN FILES
6  file = importdata('out_adcsamples1.txt');
7  pressures = importdata('out_senspressures1.txt');
8
9  % CALC SIZE
10 [rows, cols] = size(file);
11 [press_samples, dump] = size(pressures);
12
13 % RUN SCRIPT
14 avg_of_amplitude = exe_avg(file);
15
16 % CALC PARAMS FOR ADC PLOT
17 [dump, avg_samples] = size(avg_of_amplitude);
18 time_base = floor(avg_samples/24); % 24 averaged samples corresponds to 1s
19 avg_samples = time_base * 24;
20 avg_of_amplitude(1,1:avg_samples);
21 xpoints_adc = (0 : 24 : avg_samples) / 24;
22
23 % CALC PARAMS FOR PRESSURE SENSOR PLOT
24 press_step = avg_samples / press_samples;
25 xpoints_pres = (press_step : press_step : avg_samples);
26
27 lowest_press = pressures(1);
28 highest_press = pressures(1);
29
30 for i = 1 : press_samples
31
32     if lowest_press > pressures(i)
33
34         lowest_press = pressures(i);
35
36     end
37
38     if highest_press < pressures(i)
39
40         highest_press = pressures(i);
41
42     end

```

```
43
44 end
45
46 low_y_lim = lowest_press - 2 ;
47 high_y_lim = highest_press + 2;
48
49
50 % BEGIN FIGURE
51 figure(1)
52 hold on
53 grid on
54
55 % ADC SAMPLES
56 yyaxis left
57
58 plot(avg_of_amplitude)
59 xticks(0 : 24 : avg_samples);
60 xticklabels(xpoints_adc)
61 xlabel("Time [s]")
62 ylabel("Signal Strength in percentage [Vin/3.3V]")
63
64 % PRESSURE SENSOR
65 yyaxis right
66
67 plot(xpoints_pres, pressures, '--gs',...
68      'LineWidth', 2, 'MarkerSize', 10, 'MarkerEdgeColor', 'g', 'MarkerFaceColor',
69      [0.5,0.5,0.5])% 'b-o')
70 ylim([low_y_lim high_y_lim])
71 ylabel("Pressure [mbar]")
```

CHAPTER 6

Final Hardware Testing

In this section will be checked and described the behaviour of each sub-system and the experimental results of the complete device in different testing environments.

6.1 Sub-System testing

PCB Layout

The Mentor Graphics Expedition software suite was used to build the final circuit. The PCB has two layers, the first one is used for the majority of the electrical interconnections on the front side, the second one is mainly used for the ground connection and for some electrical connection on the back side. The back side of the PCB is the one will be accessible to the user when the shield is attached to the ArchMax. Having revised the work done previously concerns have arisen about the susceptibility of some components such as the receiving filter. For this reason it has been chosen to connect to ground all the layer two not strictly necessary for the realization of interconnections. This arrangement should significantly reduce both the susceptibility to noise of the components. The dimensions were chosen so that the PCB was complementary to the ArchMax board creating a shield. The anchoring between the two boards is guaranteed by two rows of pins on the sides of the PCB. The passive hardware components and the integrated circuits have been placed on the side of the PCB that is closed towards the ArchMax. On the opposite side three screw connectors have been positioned, two of those are used to secure the antennas and the other one can be used to connect an external battery with wires. In parallel to this last vertical connector is placed a container slot for 18650 type lithium-ion batteries (2.6 Ah of capacity). A four-pin connector was also added to connect the pressure sensor. Finally, two LEDs and two $370\ \Omega$ resistors have also been added to the back of the PCB. These two LEDs are used to provide an easily visible feedback to the user about the operating status of the device.

When the system is power on for both LEDs start to blink in opposite phase (just one is turned on per time) every 1 s, the user must press the hardware reset button to initialize the dev board. When the user presses the button the system perform the initialization of the peripherals and it is ready to be immersed in water. Both LEDs are turned on.

When the system is in the running state LEDs start to blink. Both the receiver are programmed to invert the state of the LED on the right every 2.5s, every time the pressure sensor routine is called and executes one step. The LED on the left is toggled every time a pressure value is written on the micro-SD.

When the execution terminates and the file containing all the samples is saved both LEDs are turned off. Then they start to blink every 1 s in phase (both turned on or off at the same time).

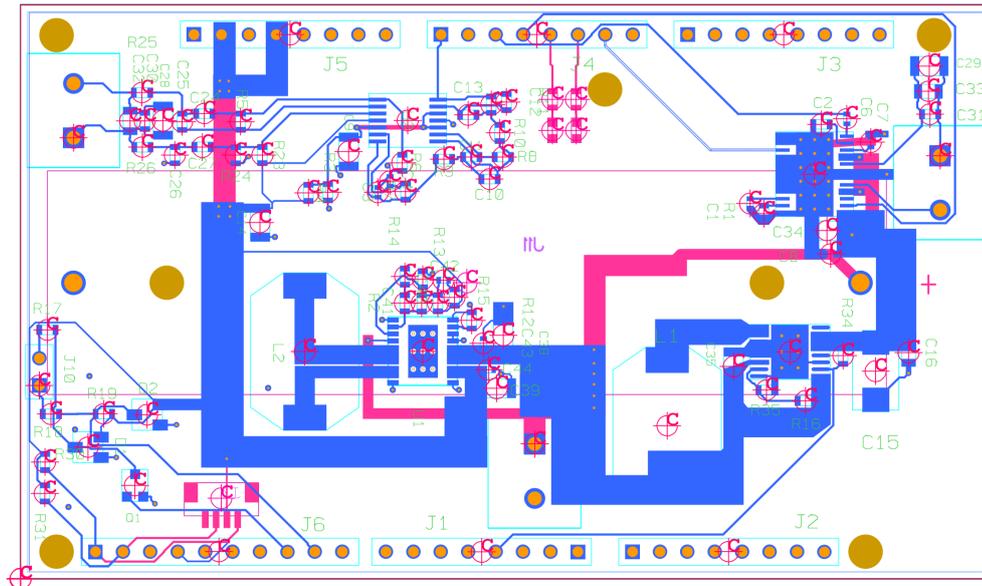


Figure 6.1: Layout of the PCB. In blue metal layer one and in pink metal layer two.

3.3 V Power Supply testing

The power supply system has been tested to verify the power up transient and voltage stability. The voltage is stable at 3.32 V. The previously developed circuitry suffered from a sawtooth oscillation of about 10 mV. This oscillation was negatively affecting the amplifier of the reception system. To eliminate this problem an integrated circuit with low noise architecture was chosen. By analyzing the output voltage of the new circuit, by setting the oscilloscope in AC, no significant oscillations are visible.

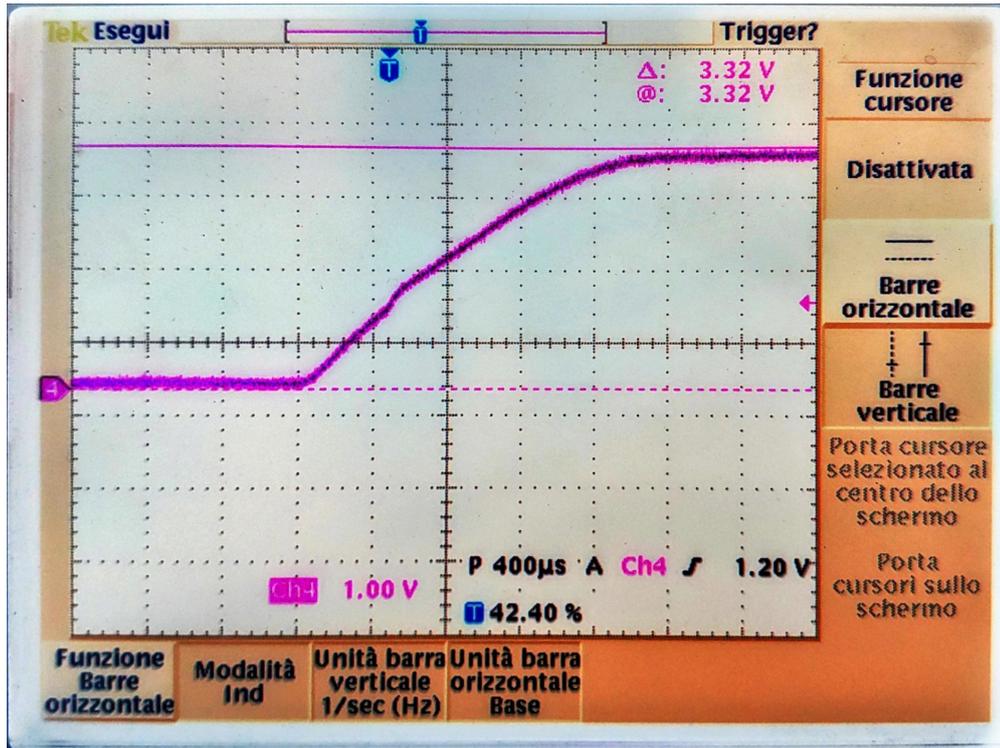


Figure 6.2: Settling time of the output voltage of the 3.3 V regulator

Receiver testing

The test of the receiving amplifier was carried out by applying a sinusoidal signal with an amplitude equal to 1 V with frequency of 10 kHz to a voltage divider composed of two resistors as shown in figure 6.3. The sine is provided using a signal generator. The first with a value of 99.8 kΩ and the second of 101.2 Ω. The amplifier input is connected across the R2 resistor. In this configuration the amplifier is receiving an input voltage with amplitude equal to a thousandth of the sinusoidal signal. This testbench circuitry is needed in order to have a sufficiently low amplitude input signal such that the voltage output of the amplifier is not saturating to 3.32 V. The 10 kHz measurement is shown in the figure 6.4. The precise voltage can be calculated as:

$$V_{R2} = V_s \frac{R2}{R1 + R2} = 1 \text{ V} \frac{101.2 \Omega}{99.8 \text{ k}\Omega + 101.2 \Omega} = 1.01 \text{ mV} \quad (6.1)$$

The voltage output of the filter is measured using an Analog Discovery. The amplitude is equal to 1.1 V. Therefore the gain can be evaluated as:

$$Gain_{dB} = 20 \text{Log} \left(\frac{V_{out}}{V_{R2}} \right) = 20 \text{Log} \left(\frac{1.1 \text{ V}}{1.01 \text{ mV}} \right) = 60.67 \text{ dB} \quad (6.2)$$

The gain peak is 61.06 dB at 10.33 kHz. The -3 dB bandwidth was verified too and resulted equal to 1.82 kHz. The system was tested with outband signals at 1 kHz and 100 kHz, no appreciable voltage at the output was observed.

The antenna requires to be coupled with a capacitance. On the PCB layout three pads of different sizes have been created in order to perform an empirical tuning. The best result was obtained with a parallel capacitance of 27 nF.

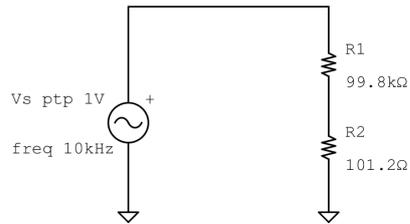
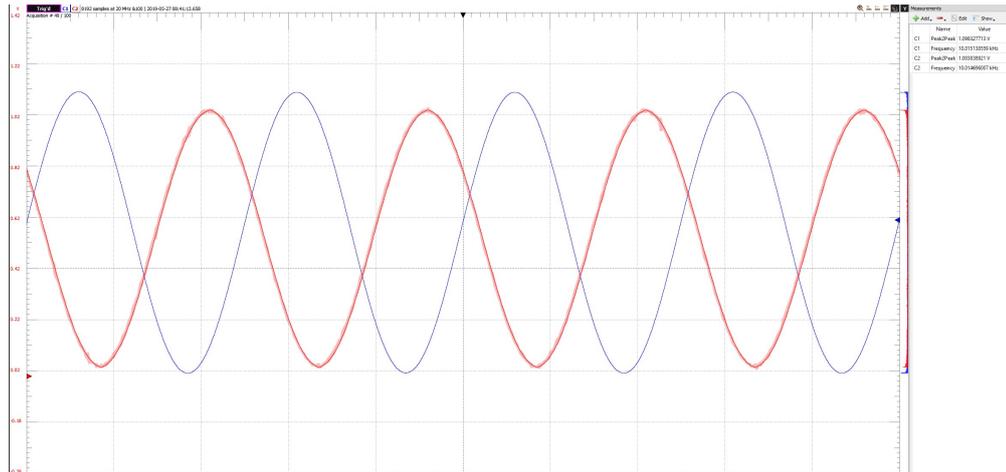


Figure 6.3: Scheme of the circuit used to test the receiver amplifier.

Figure 6.4: Generated signal in red and amplifier output signal in blue. Settings 200 mV/div, 40 μ s/div.

Transmitter testing

The transmitting antenna is coupled with a 62.95 nF series capacitance. The input of the class d amplifier is connected to the output of the dev board's DAC. In figure 6.5 is shown the voltage signal measured using two probes differentially across the transmitting antenna.

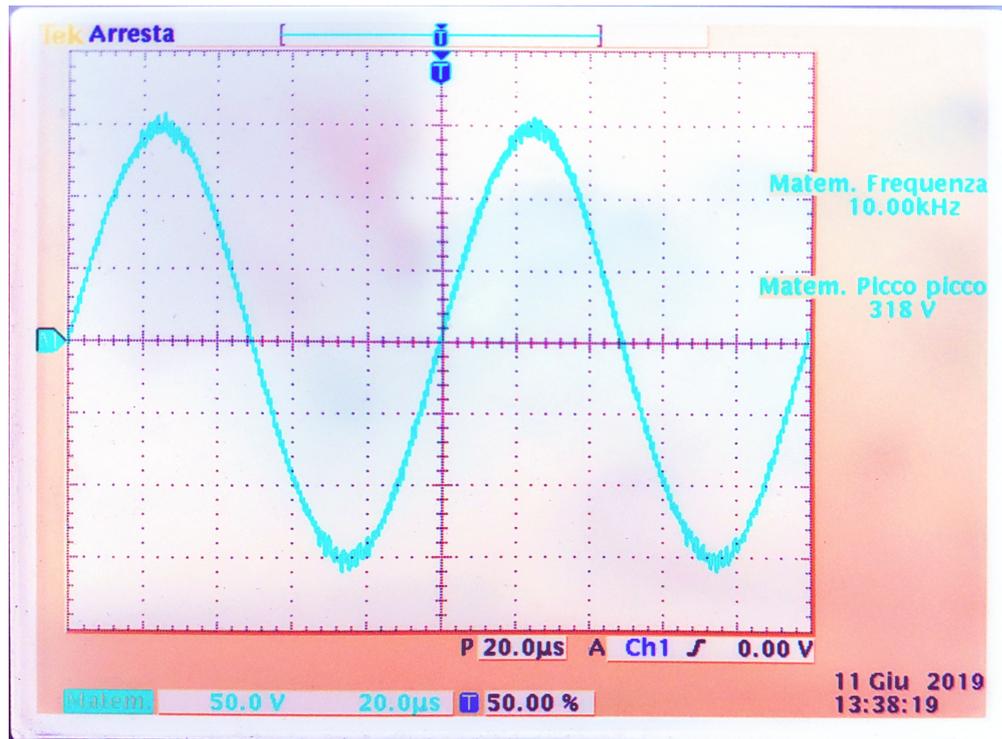


Figure 6.5: Voltage signal across the transmitting antenna, measured using two probes. Settings 50 V/div, 20 μ s/div

6.2 Complete system testing

The shield is attached to the dev board. The SD card, the pressure sensor and the wet contact are connected and enabled. The system is now powered using a 18650 Li-On battery.

6.2.1 Indoor testing

The first test takes place in a closed environment, inside the DET. A timed program is started to last 25 min. The OOK coding is turned off during this test since it would be not useful, the transmitting system generates a continuous stream of sines. During this time the transmitter is left in the same position, while the receiver is moved. Both devices are kept at the same height (ground floor) and the antennas maintain, as far as possible, a perpendicular orientation on a straight line. The output file is analyzed using the MATLAB script, a couple of samples are here reported. In figure 6.6 the receiver is placed at 12 m from the transmitter and in figure 6.7 the two systems are separated by 19 m. The receiver was then moved to 23.2 m where the signal strength dropped to 15% with significant oscillations (around 5%) and to 30 m where the signal received was mainly environmental noise.

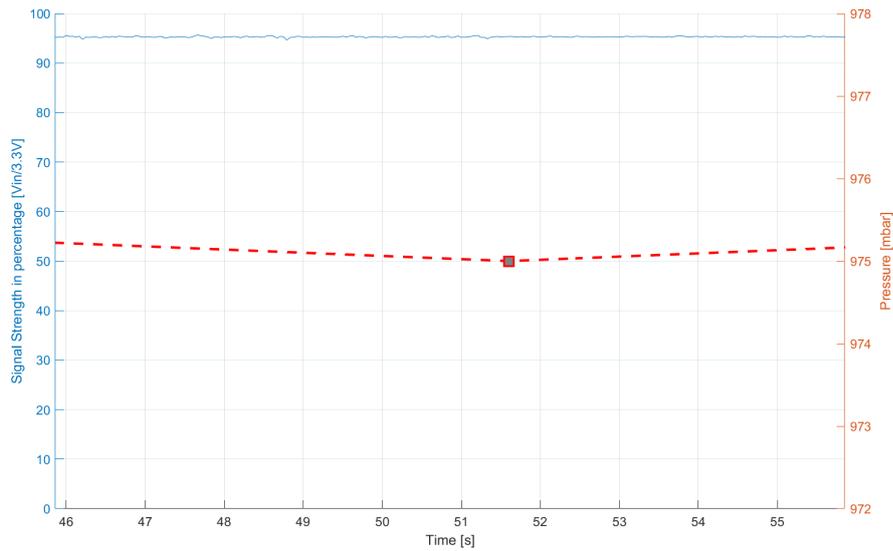


Figure 6.6: Distance of 12 m. The orange line shows the pressure value [mbar], and the blue one is the signal strength [V/V] in percentage.

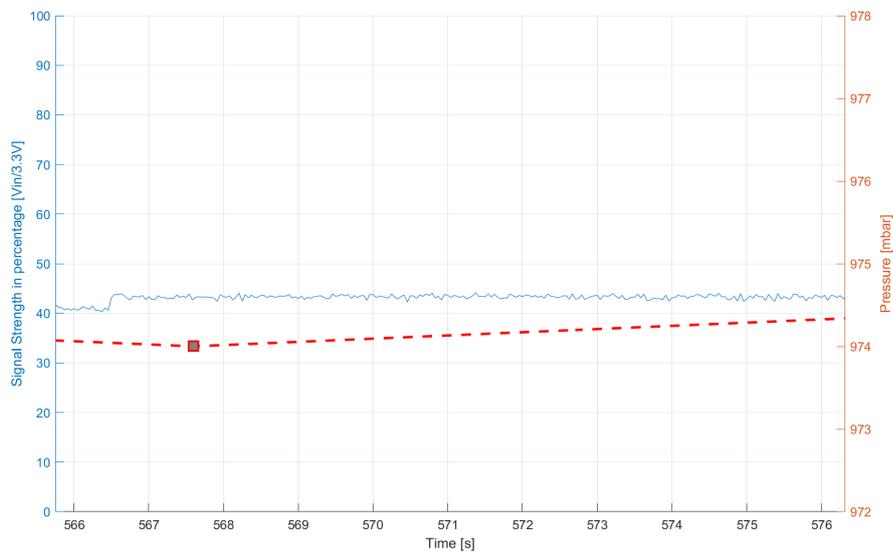


Figure 6.7: Distance of 19.2 m. The orange line shows the pressure value [mbar], and the blue one is the signal strength [V/V] in percentage.

Distance [m]	Signal Strength [V/V]
12	95%
19.2	43%
23.2	15%
> 30	< 5%

Table 6.1: Indoor testing signal strength related to the distance summary

The discrepancy between the measurement at 19.2 m and that at 23.2 m is relevant. This is probably due to the presence of some fire doors, made entirely of metal which could interfere with the

transmission.

6.2.2 Outdoor testing

A fundamental aspect to consider in this type of test concerns the environmental factor. To perform a test at such a long distance in a closed environment it was necessary to use a corridor. This corridor could be responsible for an effect similar to a waveguide. To check the validity of the test, a second measurement was performed outdoors. This test produced the following results at 15.2 m distance in figure 6.8 and at 20 m in figure 6.9.

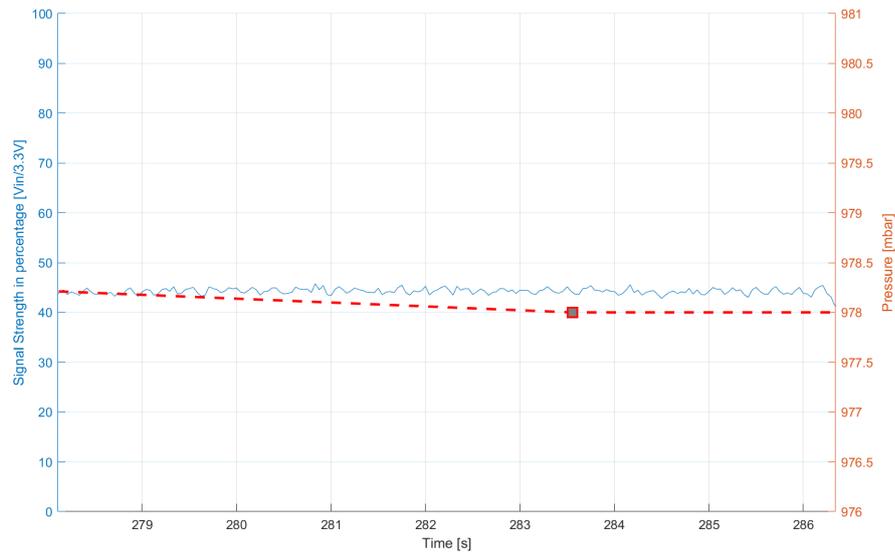


Figure 6.8: Distance of 15.2 m. The orange line shows the pressure value [mbar], and the blue one is the signal strength [V/V] in percentage.

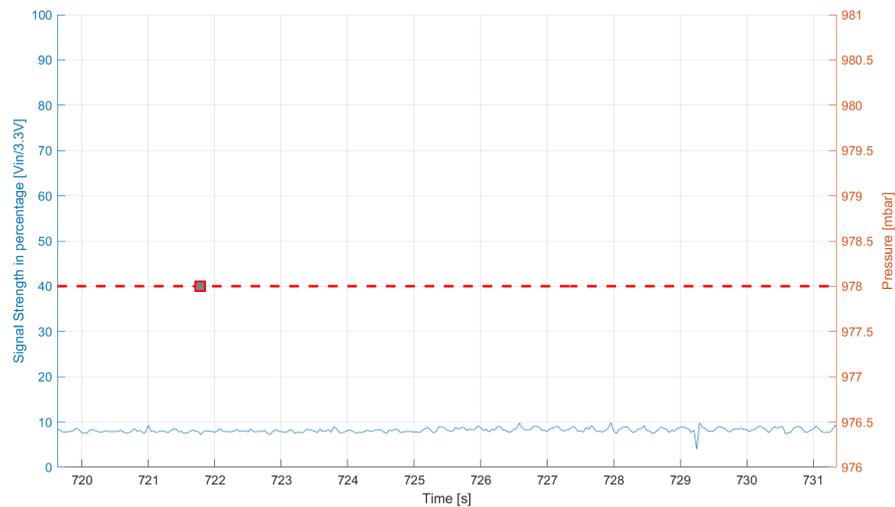


Figure 6.9: Distance of 20 m. The orange line shows the pressure value [mbar], and the blue one is the signal strength [V/V] in percentage.

Distance [m]	Signal Strength [V/V]
< 10	> 95%
15.2	44%
> 20	< 10%

Table 6.2: outdoor testing signal strength related to the distance summary

CHAPTER 7

Conclusion and Future work

The aim of the thesis was to develop a system capable of allowing the evaluation of the efficiency and performance of very low frequency (VLF) EM waves for communication in the marine environment. The system involves the use of a dedicated PCB and a microcontroller for which some functions have been written that can fully exploit the hardware and the main peripherals. The transmitter was able to generate a sinusoidal signal of about 300 V and the reception system proved to be capable of producing an output voltage without significant noise despite the high gain. With two voltage regulators it was possible to power all the analog circuitry and the development board. The stability of current absorbed by the board is very critical for the writing process of a flash memory.

Both indoor and outdoor results were encouraging and showed that it is possible to transmit up to a few dozen of meters while still with good signal strength. Since the transmission is based to resonant circuits, it is possible to hypothesize that by exploiting oscillators with higher factors of merit better results could be obtained. The antennas used in these tests are simple copper coils that have a series resistance of about 30Ω and have a diameter of about 15 cm. Using antennas with better parasitic parameters it is assumed that overall performance could improve. Also a coupling procedure should lead to a better power transfer. A second possibility could be to directly include an oscillator in the transmission circuit in order to avoid the continuous use of a DAC. These types of circuits in fact require only to be fed to produce an oscillation. This aspect could significantly reduce the computational load of the card. Once established the goodness of the communication system via EM waves in the marine environment it will be possible to establish the actual possibilities of system development. In case of excellent performance the system could be used to transmit information from a boat to a submerged device, crossing the water air barrier. If the performances turn out to be more limited, the project could be used to create a communication system between a wrist computer for divers who are in proximity during the dives. These types of applications would require the application of a much more complex coding and the formation of a half duplex channel for the transmission of information. Once the potentialities of the medium have been established and the transmission channel is refined, peripherals dedicated to the immediate vision of the information exchanged as an LCD screen can be included in the final device, for which routines already exist within the project archive.

CHAPTER 8

Bibliography

[1] Dale Green, "In Situ data extraction from ocean sensors", OCEANS, IEEE Conference, pp. 1-4, 2010.

[2] X. Che, I. Wells, G. Dickers, P. Kear, X. Gong, "Re-evaluation of RF electromagnetic communication in underwater sensor networks," IEEE Commun. Mag., vol. 48, no. 12, pp. 143-151, December 2010.

[3] Hemani Kaushal and Georges Kaddoum. Underwater optical wireless communication. IEEE Access, 4:1518–1547, 2016.

[4] A Hales, G Quarini, G Hilton, L Jones, E Lucas, D McBryde, and X Yun. The effect of salinity and temperature on electromagnetic wave attenuation in brine. International Journal of Refrigeration, 51:161–168, 2015.

[5] Michael R Frater, Michael J Ryan, and Robin M Dunbar. Electromagnetic communications within swarms of autonomous underwater vehicles. In Proceedings of the 1st ACM international workshop on Underwater networks, pages 64–70. ACM, 2006.

[6] H. U. Sverdrup, Martin W. Johnson, Richard H. Fleming. The Oceans, Their Physics, Chemistry, and General Biology. New York: Prentice-Hall, c1942 1942. <http://ark.cdlib.org/ark:/13030/kt167nb66r/>

[7] Peter KPK Weyl. Oceanography: an introduction to the marine environment. Number 551.46 WEY. 1970.

[8] D.H.Gadani,V.A.Rana,S.P.Bhatnagar,A.N.Prajapati,A.D.Vyas,"Effect of salinity on the dielectric properties of water,"IndianJ.PureAppl.Phys.vol. 50, pp. 405-410, 2012.

[9] A. G. Miquel, "UWB antenna design for underwater communications," Universitat Politecnica de Catalunya, May 2009.

[10] O.P.N. Calla, Dinesh Bohra and Rajesh Vyas. MEASUREMENT OF DIELECTRIC CONSTANT OF SALINE WATER AND ESTIMATION OF EMISSIVITY AND SCATTERING COEFFICIENT AT DIFFERENT PHYSICAL TEMPERATURES AT MICROWAVE FREQUENCIES. International Centre for Radio Science, 'OM NIWAS'A-23 Shastri Nagar Jodhpur – 342003

- [11] O. Aboderin, L. M. Pessoa, H. M. Salgado, "Performance Evaluation of Antennas for Underwater Applications," *Wireless Days 2017*, p. 4, March 2017.
- [12] A. Massaccesi, "Analysis and design of underwater antennas for diving applications," *Politecnico di Torino*, 2015.
- [13] Söderström, Emma. "Reducing Effects of Multipath Propagation With a Blind Equalizer". Linköping University, 2018.
- [14] TE Connectivity. MS5837-30BA datasheet. Rev 03/12/2015
- [15] STMicroelectronics. Reference Manual STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced ARM®-based 32-bit MCUs, July 2017. Rev. 15.
- [16] Analog Devices, "LTC3113 3A Low Noise Buck-Boost DC/DC Converter", LTC3113 datasheet
- [17] Sergio Franco, "Loop gain measurements". *EDN Network*, Sept 2014
- [18] Texas Instruments, "TPS6103x 96% Efficient Synchronous Boost Converter With 4A Switch", TPS6103x datasheet, September 2002, revised March 2015.
- [19] Texas Instruments, "TPS6305x Single Inductor Buck-Boost With 1-A Switches and Adjustable Soft Start", TPS6305x datasheet, July 2013, revised July 2015.
- [20] SD Group (Panasonic, SanDisk, Toshiba) and SD Card Association. *SD Specifications. Physical Layer Simplified Specification*, September 2006. Ver. 2.00.
- [21] Mathworks. PDF Documentation for MATLAB. *MATLAB Mathematics*.
https://it.mathworks.com/help/pdf_doc/matlab/math.pdf