Universidade de Aveiro
2021

Samuel
Martins Simões

Sistema Adaptativo para Pré-distorção de Amplificadores

Amplifier Predistortion Adaption System

Universidade de Aveiro
2021

Samuel
Martins Simões

Sistema Adaptativo para Pré-distorção de Amplificadores

Amplifier Predistortion Adaptation System

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica de Doutor Telmo Reis Cunha, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dissertation presented to the University of Aveiro to fulfill the required to obtain the degree of Master in Electronics and Telecommunication Engineering, developed under the scientific guidance of Doctor Telmo Reis Cunha, Professor in the Department of Electronics, Telecommunications and Informatics of the University of Aveiro.

o júri / the jury

presidente / president	**Professora Doutora Susana de Jesus Mota** Professora Auxiliar, Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro
arguente Principal / main examiner	**Professor Doutor Manuel Cândido Duarte dos Santos** Professor Auxiliar, Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto
orientador / advisor	**Professor Doutor Telmo Reis Cunha** Professor Associado, Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos / acknowledgements

Em primeiro lugar gostaria de agradecer ao meu orientador, Professor Telmo Reis Cunha pelo apoio científico prestado, por toda orientação, sugestões e esclarecimento de dúvidas disponibilizado.

Gostaria também de agradecer ao Instituto de Telecomunicações, pelo acesso às instalações, laboratório, equipamentos de medida e componentes. Um obrigado ao Paulo Gonçalves e ao Hugo Mostardinha pelo apoio técnico prestado pela simpatia, profissionalismo e disponibilidade demonstrada.

Agradeço à Universidade de Aveiro e ao Departamento de Eletrónica, Telecomunicações e Informática pelas oportunidades e todo o ensino de qualidade.

A todos os meus amigos, em especial ao Rui Carvalho pelo apoio e ajuda imensurável ao longo dos anos, ao Bruno Brandão pelo apoio e conhecimento partilhado durante o design deste projeto. Por último e não menos importante aos meus amigos Eduardo Fernandes, Joana Castro, Joaquim Melim, José Nogueira e Rafael Almeida, a todos um grande obrigado pelo companheirismo durante dias a fio e noites adentro ao qual tornaram esta viagem inesquecível.

Finalmente um grande obrigado à minha família, pai, mãe e irmã pelo apoio incondicional ao longo da minha vida, nomeadamente nestes últimos anos, por acreditarem e não me deixarem desistir, a eles um enorme obrigado pelo que sou e consegui atingir.

Palavras-chave

Pré-distorção, Pré-distorção Analógica, Amplificador de Potência, Rádio Frequência.

Resumo

No mundo de hoje, com as exigências necessárias para cumprir os requisitos para o 5G, é fundamental que os amplificadores de Rádio Frequência (RF) funcionem de uma forma cada vez mais eficiente. Para cumprir esse objetivo, uma das opções existentes foca-se no uso de técnicas de linearização para melhorar o seu desempenho na sua zona de saturação. Uma das técnicas que mais se destaca, quer seja pela sua eficiência ou pelo nível de linearização que proporciona, é aquela que consiste na aplicação de pré-distorsores. Estes podem ser utilizados com uma configuração pré-programada ou podem adaptar a sua resposta em tempo real, recorrendo à leitura do sinal existente na saída do amplificador de potência.

Este trabalho foca-se na construção de um sistema adaptativo para pré-distorção de amplificadores em tempo real. Este sistema realiza a medição espetral de um sinal de RF com o objetivo de no futuro ser usado para controlar um pré-distorsor.

Tendo em consideração um cenário de *Multiple-Input and Multiple-Output (MIMO)*, onde um transmissor é construído através de vários amplificadores de baixa potência e com banda larga (em que cada um requer algum tipo de linearização), o sistema de medida espectral assume, desde o início, os requisitos de obter um produto de baixo custo e que exija o mínimo espaço possível.

Keywords

Predistortion, Analog Predistotion, Power Amplifier, Radio Frequency

Abstract

In today's world, with the demands needed to fulfil the challenges on 5G, it is important that Radio Frequency (RF) amplifiers work in a more efficient way. To meet this objective, one of the existing options focuses on the usage of linearization techniques to improve amplifier performance on its saturation zone. One relevant technique, which has great efficiency and linearization potential is the application of predistorters. They can be used with a pre-programed configuration or they may adapt their response in real-time, with the help of signal readings at the power amplifier's (PA) output.

This work focuses on the design and implementation of an amplifier predistortion real-time adaption system. This system measures the spectral power of an RF signal with the purpose of, in the future, incorporating a predistorter automatic adaption system.

Having the Multiple-Input and Multiple-Output (MIMO) transmitter scenario as background, where a single transmitter is built from several low-power and wideband amplifiers (each of which require linearization actions), the developed spectrum measurement system assumed, from the start, the requirements of achieving a very low cost and reduced space confinement.

List of contents

List of figures

List of tables

List of acronyms

4G	Fourth Generation
5G	Fifth Generation
AC	Alternating Current
ACPR	Adjacent Channel Power Ratio
ADC	Analog-to-Digital Converter
ADS	Advanced Design System
AP	Access Point
APD	Analog Predistorter
CCPR	Co-Channel Power Ratio
CP	Charge Pump
DAC	Digital-to-Analog Converter
DC	Direct Current
DPD	Digital Predistorter
ETSI	European Telecommunications Stander Institute
FCC	Federal Communications Commission
FSK	Frequency-Shift Keying
HID	Human Interface Device
I^{2}C	Inter-Integrated Circuit
I/O	Input/Output
IDE	Integrated Development Environment
IF	Intermediate Frequency
IMD	Intermodulation Distortion
IoT	Internet of Things
IT	Institute of Telecommunications

LDO	Low-Dropout Regulator
MIMO	Multiple-Input and Multiple-Output
MISO	Master In Slave Out
MOSI	Master Out Slave In
PA	Power Amplifier
PC	Personal Computer
PCB	Printed-Circuit Board
PD	Phase Detector
PEP	Peak Envelope Power
PLL	Phase Lock Loop
PPS	Peripheral Pin Select
RF	Radio Frequency
RMS	Root Mean Square
SAW	Surface Acoustic Wave
SPI	Serial Peripheral Interface
TCXO	Temperature Compensated Crystal Oscillator
USB	Universal Serial Bus
VCO	Voltage Controlled Oscillator
VGA	Variable Gain Amplifier
VSG	Vector Signal Generator

Chapter 1

Introduction

1.1 Motivation and Context

Nowadays, there is a steady increase on the number of equipment that require wireless communications. Not only there is a ubiquitous amount of the common cell phone, the Internet of Things (IoT) is having an explosive growth, which also will put an extra strain on the current infrastructure and some of the new technologies will have new requirements that are not possible with the current Fourth Generation (4G) cellular communication networks.

To solve this, the new wireless generation, the Fifth Generation, will need to address these issues, meaning that a 5G site will need not only to support a huge number of user equipment, but also be ready for any special constrains requested by them. Such common needs include very low latency, high data rate, reliability, security concerns and better cost- and energy-efficiency [1]. To achieve these obligations, the foundations of 5G will rely on edge technology that lay on millimetre waves, small cell, Multiple-Input and Multiple-Output (MIMO), beamforming, and full duplex. Millimetre waves refer to the usage of frequencies over 6 GHz allowing for wider channel bandwidths, small cells are used as Access Points (APs) used for extending coverage beyond regular base stations, massive MIMO allows the use of multiple transmitters to deliver a beam signal [2].

At the moment, the solutions used on 4G to obtain the aforementioned cost/energy-efficiency are not good enough for current stander because they use a higher efficiency Radio Frequency (RF) Power Amplifiers (PAs) that cause unacceptable distortion in form of Intermodulation Distortion (IMD), which produce Adjacent Channel Power Ratio (ACPR), Co-Channel Power Ratio (CCPR) [3][4]. These spurious frequencies outside of the desired bandwidth, need to be under the limitations impose by Federal Communications Commission (FCC) and European Telecommunications Stander Institute (ETSI).

To remedy this fault, a predistorter could be added to the transmitter chain to correct the signal. While this is a valid answer for the days before 5G, the components used for the predistorter are usually very expensive and use too much power [5]. This is a new problem caused by the use of millimetres waves and the surge of cell availability with its reliance on MIMO technology, which requires a massive number of transmitters, and a high number of low-power amplifiers per transmitter.

A predistorter is a device that modifies the input signal of the PA in a way that cancels the distortions introduced by the PA. There are two types of predistorters, the Digital Predistorter (DPD) and the Analog Predistorter (APD).

The APD uses an analog circuit, that relies on a predetermined open loop to cancel the distortion.

The DPD functions as an algorithm on a digital processor. Some of them use the output PA signal read through an Analog-to-Digital Converter (ADC) and is analysed in real time determine the distortion caused by the PA and corrects the input signal. The main drawback of this solution is that the processor used must have a high clock rate to be able to sample the signal, especially when operating in wideband applications.

In summary, the DPD is a better approach to obtain a higher quality signal, however the processor/ADC combination used is not as cost/energy efficient as an APD.

The target of this work is to develop a prototype of a simple (and low cost) circuit which measures the spectrum of the output signal of a PA, gathering the spectrum in a low grade microcontroller for further using such information to tune the parameters of the predistorter that compensates the PA. This prototype will not be based on a very fast sampler, but will consider a very sharp band-pass filter (followed by a power meter) through which the spectrum of the PA output signal is slided.

1.2 Objectives

The scope of this work was the development of a hardware platform that provided an adapting mechanism to a predistorter. This hardware should be capable of measuring an output signal from a PA, then it should calculate the distortion of the PA's output. With the calculation of the distortion completed, its value is used to act over a predistorter. This back-loop system optimizes the predistorter, improving its capability to reduce the distortion at the PA signal output. Figure 1.1 frames the created system on a transmitter chain.

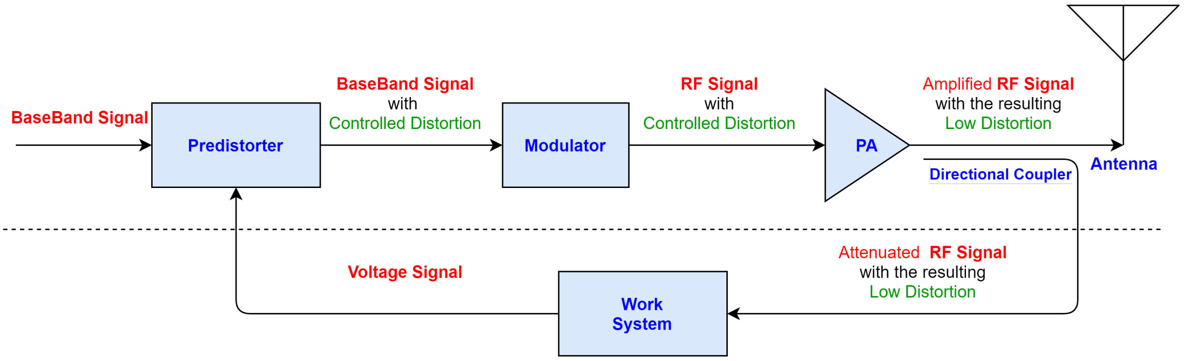


Figure 1.1: Work system framed in RF transmitter chain diagram.

The implementation of this adaptative system should be done using low complexity components and low complexity processing units (e.g. a microprocessor), both with a low rate of energy consumption.

Some of the most important requirements for these system's performance are as follow:

- Input Central Frequency of 2 GHz: The circuit must be capable of measuring signals with a central frequency around 2 GHz, for which some amplifiers available at the RF Laboratory at the Institute of Telecommunications (IT) were designed.

- Input bandwidth of 20 MHz: The input signal should have a bandwidth up to 20 MHz, which is a bandwidth already used in some mobile application.

- Input power level of -10 dBm: The PA signal output, which is the input signal of the designed adaptative system, would be acquired through a directional coupler, allowing a reliable copy of the signal without damaging the power of the original output signal. The signal obtained by the directional coupler is then much lower than the transmitting signal, with a level around -10 dBm.

- Power range of 90 dB: This value covers the regulation set by the entities such as FCC and ETSI, which state that for some mobile application, the adjacent channel power needs to be 65 dB lower than the transmitting signal [6].

- Output: The output of this system should be sampled by Digital-to-Analog Converters (DACs) and generate the necessary voltage signal to control the APD (predistorter).

Succinctly, the specifications and requirements of this system were set taking in consideration the state of the art for 5G technologies, but also the possibility of creating a prototype with the tools provided by IT.

1.3 Outline

The rest of this document is structured as follows:

Chapter 2: This chapter overviews the most important points relative to the objective of this work. The chapter begins explaining the reason why PAs suffer from signal distortion and the problems associated with it. Subsequently, some solutions to the distortion problem are presented. This chapter ends with the introduction of the proposed adaptive method to optimize a predistorter.

Chapter 3: In this chapter, it is explained the thought behind every component that was chosen, followed by a description of the design process considered for the implementation of the prototype circuit and sub-circuits. Afterwards, the Matching Networks between devices are presented and finally it is analysed the layout of the entire system.

Chapter 4: This chapter is focused on the software produced for this project. It begins with the introduction of the firmware developed to control the microcontroller, followed by the presentation of the user application to control the system over a Personal Computer (PC). This chapter ends with the overview of the Serial Peripheral Interface (SPI) communication used to control the devices, and communication between both software.

Chapter 5: This chapter contains the test and the results of the developed system. The final modifications on the Printed-Circuit Board (PCB), which led to the final working system discussed here. It also describes some tests focused on the crucial parts of the system.

Chapter 6: The last chapter presents the conclusions and ends with some suggestions for future work.

Chapter 2

State Of The Art

This chapter shows some of the existing technologies and techniques that are used in this work, along with some potential alternative examples.

2.1 Overview

A power amplifier is a circuit that converts a Direct Current (DC) power supply into an amplified Alternating Current (AC) output signal, whose main components are RF transistors, but since in the real world there are no perfect components, the amplifier is not linear in all of its conductive zone presenting, therefore, a nonlinear behavior [7]. In Figure 2.1 it can be seen a sketch of the gain of a regular power amplifier.

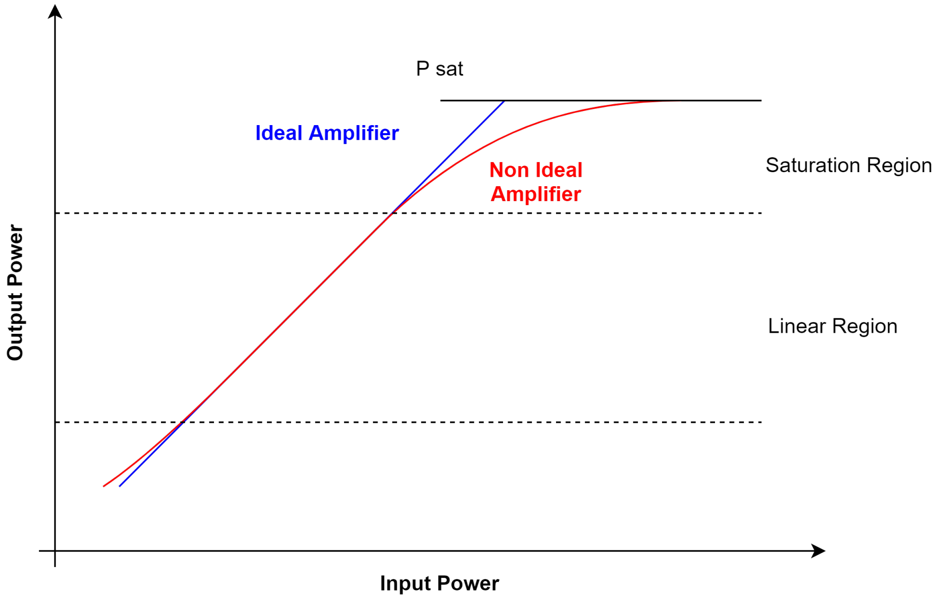


Figure 2.1: Gain regions of a Power Amplifier.

2.1.1 Amplifier Efficiency

The efficiency of a PA is determined by its output power divided by the power supplied to the PA (from the DC supply and from the input signal generator) Equation (2.1), therefore, since the output power achieves its maximum value on saturation zone and the DC power spent does not scale linearly, it can be concluded that this is the region where the PA has the most efficiency. Since this is out of the linear zone, it demonstrates that there is a trade-off between linearity and the PA efficiency, which is known as the power-spectrum efficiency trade-off [7].

$$\eta_t = \frac{P_{out}}{P_{dc} + P_{in}}. \tag{2.1}$$

There are multiple choices for the design of a PA. Each design will have a different impact on its efficiency and linearity, being the bias point of the transistors determinant for the overall behavior [8].

The PA bias defines the conduction period of the output current and will dictate the class of operation of the PA. Therefore, the bias point has direct impact on the PA efficiency, on its linearity, and also on the maximum output power [9]. The class A, whose bias allows the RF transistor to have a conduction angle of the output current of 360 °. This class is inherently the most linear but also the least efficient, with a maximum theoretical efficiency of 50 %, but due to real-world constraints, most applications usually cannot reach more than 25 % of efficiency [9].

There are other amplifier classes of operation, where their bias point is set to conduct on a lower conduction angle, as a result it is possible to achieve much better efficiency with a cost of lowering the linearity mainly due to the insertion of distortion on the signal [9]. Nowadays, these classes are a better choice than the class A, because there are techniques that can be used to easily remove the added harmonics and they keep the advantage of greater efficiency [4]. In the end, the root cause of the distortion seen on the output wave is the clipped signal caused by the saturation zone of the PA.

2.1.2 Amplifier Nonlinearities

The nonlinearity of the system comes mostly from the compression behaviour of the amplifier, meaning that the output power can no longer maintain the gain as the input power increases [9]. These nonlinearities cause Intermodulation Distortion (IMD) and Harmonic Distortion, producing a spectral transformation [7]. With this in mind, the PA has to be designed with a trade-off between power efficiency and spectral efficiency.

To illustrate the impact of nonlinearity on the signals frequency content, Figure 2.2 presents the response of a 5th-order polynomial to a two tone excitation signal (of frequency close to 100 Hz), and in Table 2.1 are represented the components for the distinct frequencies generated by a Polynomial Filter model, whose pth-order kernel is $H_p(.)$, when excited with a two-tone signal at frequencies ω_1 and ω_2, and amplitudes A_1 and A_2, respectively.

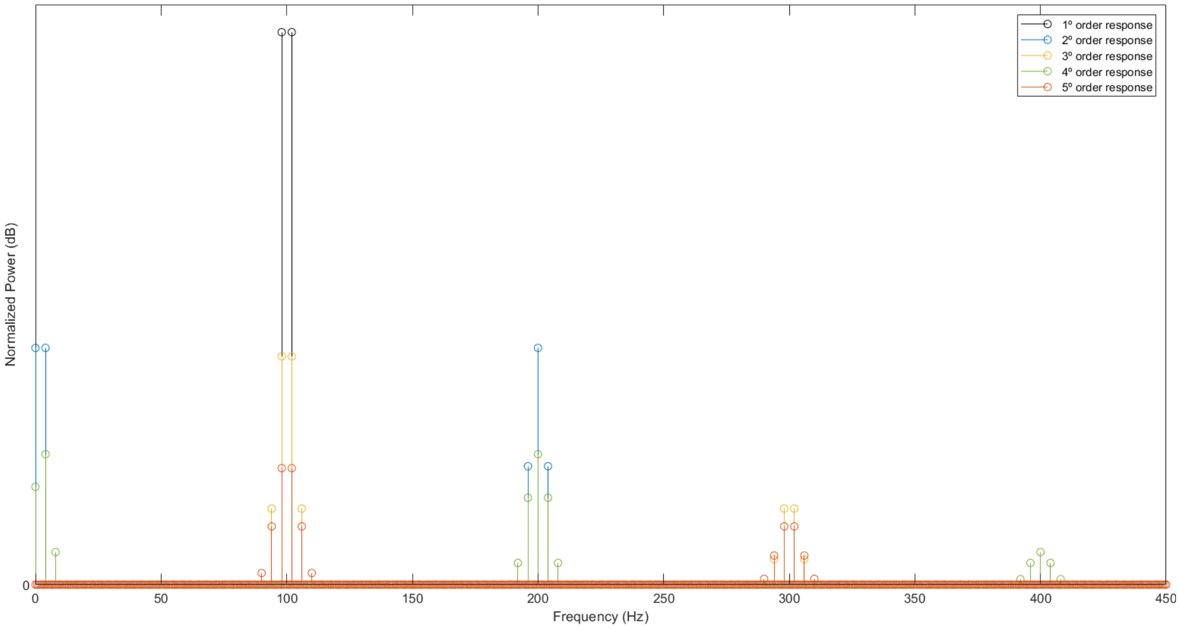


Figure 2.2: Response of a 5th-order polynomial to a two tone excitation signal.

In Figure 2.2 it can be seen that distortion appears at frequency bands distinct from the fundamental one (these can be removed by filtering), but distortion terms are also present within the fundamental band (these cannot be removed by a simple linear filter). One thing to note is that the even orders of the system response do not appear near the original frequencies so they are an issue easy to solve.

The odd order terms cause an increase both on the CCPR and on the ACPR, described in Equation (2.2) and represented in Figure 2.3 [7].

$$ACPR = \frac{IntegratedChannelSignalPower}{IntegratedAdjacent - ChanelPower_{I,S}}. \tag{2.2}$$

Table 2.1: Polynimial Filter model response to a two-tone signal at frequencies ω_1 and ω_2, and amplitudes A_1 and A_2.[7].

Response Frequency $(-\omega_m)$	Response Amplitude	Response Type
1st Order or Linear Response		
$-\omega_1$	$^1/_2 \cdot A_2^* \cdot H_1(-\omega_2)$	Linear Response
$-\omega_2$	$^1/_2 \cdot A_1^* \cdot H_1(-\omega_1)$	
ω_1	$^1/_2 \cdot A_1^* \cdot H_1(\omega_1)$	
ω_2	$^1/_2 \cdot A_2^* \cdot H_1(\omega_2)$	
2nd Order Response		
$-2\omega_2$	$^1/_4 \cdot A_2^{*2} \cdot H_2(-\omega_2, -\omega_2)$	2nd Order Harmonic Distortion
$-2\omega_1$	$^1/_4 \cdot A_1^{*2} \cdot H_2(-\omega_1, -\omega_1)$	
$2\omega_1$	$^1/_4 \cdot A_1^{*2} \cdot H_2(\omega_1, \omega_1)$	
$2\omega_2$	$^1/_4 \cdot A_2^{*2} \cdot H_2(\omega_2, \omega_2)$	
$\omega_1 - \omega_2$	$^1/_2 \cdot A_1 \cdot A_2^* \cdot H_2(\omega_1, -\omega_2)$	2nd Order Intermodulation Distortion
$\omega_2 - \omega_1$	$^1/_2 \cdot A_1^* \cdot A_2 \cdot H_2(-\omega_1, \omega_2)$	
$\omega_1 + \omega_2$	$^1/_2 \cdot A_1 \cdot A_2 \cdot H_2(\omega_1, \omega_2)$	
$-\omega_1 - \omega_2$	$^1/_2 \cdot A_1^* \cdot A_2^* \cdot H_2(-\omega_1, -\omega_2)$	
$\omega_1 - \omega_1$	$^1/_2 \cdot \lvert A_1\rvert^2 \cdot A_2 \cdot H_2(-\omega_1, \omega_1)$	Bias Point Shift
$\omega_2 - \omega_2$	$^1/_2 \cdot A_1 \cdot \lvert A_2\rvert^2 \cdot H_2(-\omega_2, \omega_2)$	
3rd Order Response		
$-3\omega_2$	$^1/_8 \cdot A_2^{*3} \cdot H_3(-\omega_2, -\omega_2, -\omega_2)$	3rd Order Harmonic Distortion
$-3\omega_1$	$^1/_8 \cdot A_1^{*3} \cdot H_3(-\omega_1, -\omega_1, -\omega_1)$	
$3\omega_1$	$^1/_8 \cdot A_1^{*3} \cdot H_3(\omega_1, \omega_1, \omega_1)$	
$3\omega_2$	$^1/_8 \cdot A_2^{*3} \cdot H_3(\omega_2, \omega_2, \omega_2)$	
$-2\omega_2 - \omega_1$	$^3/_8 \cdot A_1^* \cdot A_2^{*2} \cdot H_3(-\omega_2, -\omega_2, -\omega_1)$	3rd Order Intermodulation Distortion
$2\omega_2 + \omega_1$	$^3/_8 \cdot A_1 \cdot A_2^2 \cdot H_3(\omega_1, \omega_2, \omega_2)$	
$2\omega_1 + \omega_2$	$^3/_8 \cdot A_1^2 \cdot A_2 \cdot H_3(\omega_1, \omega_1, \omega_2)$	
$2\omega_1 - \omega_2$	$^3/_8 \cdot A_1^{*2} \cdot A_2^* \cdot H_3(-\omega_2, -\omega_1, -\omega_1)$	
$-2\omega_2 + \omega_1$	$^3/_8 \cdot A_1 \cdot A_2^{*2} \cdot H_3(-\omega_2, -\omega_2, \omega_1$	
$-2\omega_1 + \omega_2$	$^3/_8 \cdot A_1^{*2} \cdot A_2 \cdot H_3(-\omega_1, -\omega_1, \omega_2)$	
$2\omega_1 - \omega_2$	$^3/_8 \cdot A_1^2 \cdot A_2^* \cdot H_3(-\omega_2, \omega_1, \omega_1)$	
$2\omega_2 - \omega_1$	$^3/_8 \cdot A_1^* \cdot A_2^2 \cdot H_3(-\omega_1, \omega_2, \omega_2)$	
$-\omega_2$	$^3/_8 \cdot A_2^* \cdot \lvert A_2\rvert^2 \cdot H_3(-\omega_2, -\omega_2, \omega_2)$	AM/AM Conversion (Gain Compression) and AM/PM Conversion (Phase Shift)
ω_2	$^3/_8 \cdot A_2 \cdot \lvert A_2\rvert^2 \cdot H_3(-\omega_2, \omega_2, \omega_2)$	
ω_1	$^3/_8 \cdot A_1 \cdot \lvert A_1\rvert^2 \cdot H_3(-\omega_1, \omega_1, \omega_1)$	
$-\omega_1$	$^3/_8 \cdot A_1^* \cdot \lvert A_1\rvert^2 \cdot H_3(-\omega_1, -\omega_1, \omega_1)$	
$\omega_1 + \omega_2 - \omega_2$	$^3/_4 \cdot A_1^* \cdot \lvert A_2\rvert^2 \cdot H_3(-\omega_2, \omega_1, \omega_2)$	Response Sensitivity Loss
$-\omega_1 + \omega_2 - \omega_2$	$^3/_4 \cdot A_1^* \cdot \lvert A_2\rvert^2 \cdot H_3(-\omega_2, -\omega_1, \omega_2)$	
$-\omega_2 + \omega_1 - \omega_1$	$^3/_4 \cdot A_2^* \cdot \lvert A_1\rvert^2 \cdot H_3(-\omega_2, -\omega_1, \omega_1)$	
$\omega_2 + \omega_1 - \omega_1$	$^3/_4 \cdot A_2^* \cdot \lvert A_1\rvert^2 \cdot H_3(-\omega_1, \omega_1, \omega_2)$	

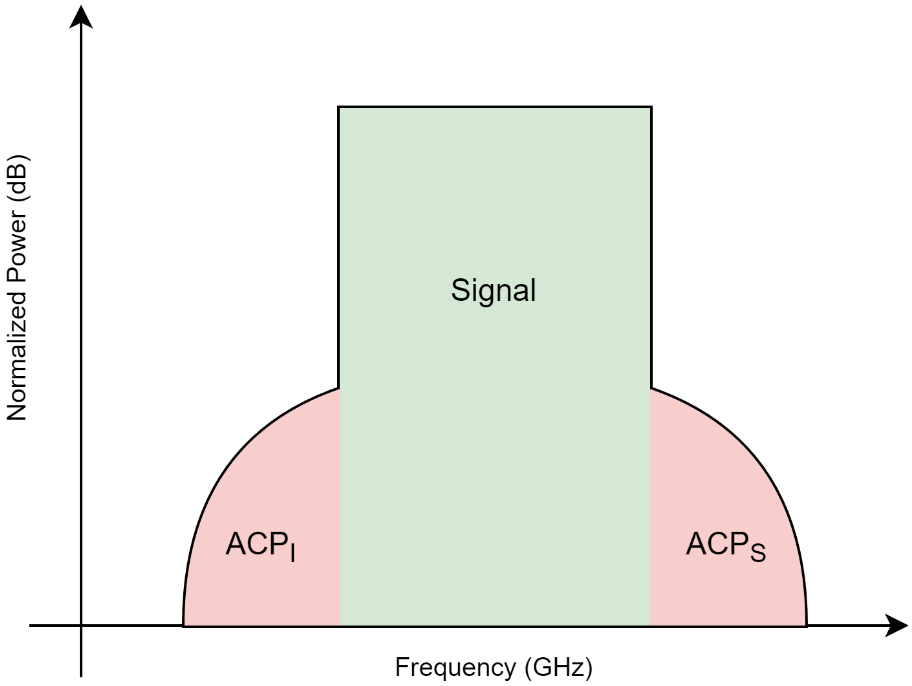


Figure 2.3: Spectrum Signal with ACPR.

Looking into the future, the 5G requirements will impose a high level of efficiency, and to achieve this the RF power transistor will have to operate in such a way that will inevitably create significant distortion [4].

There are techniques that can improve the overall performance of the PA. One well known is the linearization technique, as the name implies it uses elements that leads a PA to have a more linear behaviour. The main objective is to reduce the amplifier distortion to acceptable levels by adding low power consuming components, so that improving the linearity of the system does not cost too much of its efficiency.

2.2 Linearization Techniques

A simple way to linearize a PA is “backing off” the output power, but as this compromises the efficiency, it is not the most suitable technique to linearize the PA [4]. For that purpose, there are better linearization techniques, which can achieve a better spectral efficiency without losing too much power efficiency.

There are several linearization techniques, each with different approaches on how to linearize the PA, the three most common techniques are the Feedforward Linearization, the Analog Predistortion Linearization, and the Digital Baseband Predistortion Linearization.[9]

2.2.1 Feedforward Linearization

The feed-forward technique uses two amplifiers. The first to be considered, is the main amplifier where it creates an output signal composed by the amplified frequencies and the added distortion product obtained from the PA nonlinearity.

The second amplifier, the error amplifier, uses as input signal the distortions produced by the main PA. This correction signal is then used to cancel the distortions introduced by the previous PA. Theoretically, this approach allows the acquisition of a more precise output amplified signal and a simple design representation of this approach can be seen in Figure 2.6 [9].

Feedforward linearizers can achieve very high linearity and can support a wide bandwidth. However, it has a major drawback, since the error amplifier needs to be perfectly linear, it cannot operate with a high efficiency, leading to a significant decrease of the overall efficiency of the overall amplifier [10].

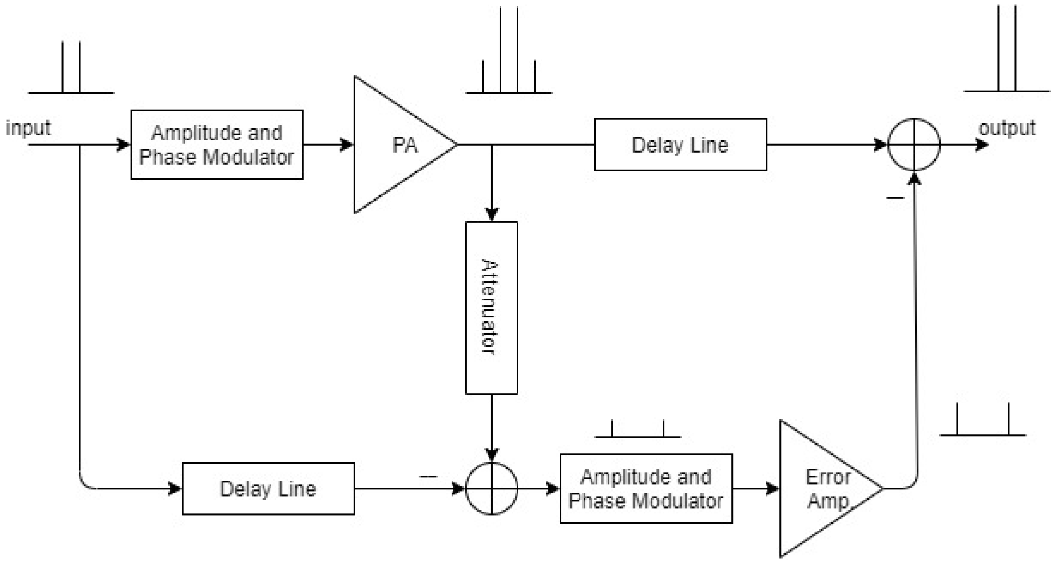


Figure 2.4: Feedforward linearization block diagram.

2.2.2 Predistortion Linearization

The predistortion can be performed either in analog or digital domain, and can operate at baseband, Intermediate Frequency (IF) or RF frequency. This technique produces a calculated distortion on the input signal, therefore, when the signal is amplified in the main amplifier, a clean output will be provided. This is possible because the non-linearity of the PA is cancelled by the distortion added by the predistorter. An illustrative example is presented in Figure 2.5 [9].

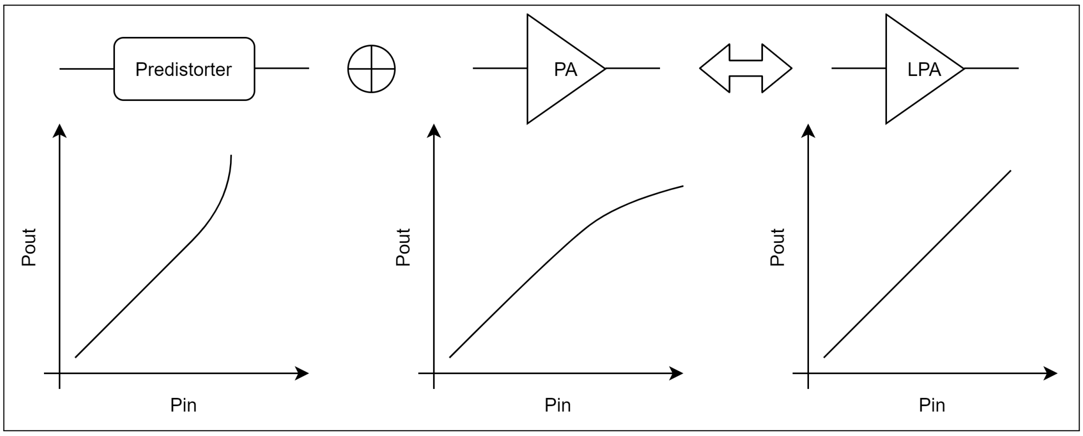


Figure 2.5: Predistotion linearization technique.

Analog Predistorter

The Analog Predistorter (APD) can be used in both RF frequencies and Baseband frequencies. Nowadays, this type of predistorters is mostly used for high power applications with bandwidths on the higher end of GHz [10].

A use case of APD is to split the input signal into a distortion path and a linear path. The signal on the distortion path is run through a "distortion creator" which will introduce a similar error as the main amplifier would create. The objective is that when later combining these two paths, they would cancel each other and provide an output signal without the usual distortion created by a standard PA [9].

The APD has low implementation cost, typically has high efficiency, and can perform on wide bandwidth signals, but lacks linearity performance when compared with other options [10].

Digital Predistortion Linearization

The Digital Predistorter (DPD) is based on processing the signal digitally, at baseband frequencies. The predistortion is done digitally by using the inverse characteristic of the transfer function of the PA [9].

This predistorter can be designed with either a close loop or an open loop, and some of the better options is to use a Lookup Table (LUT). A Lookup Table can be obtained through constellation mapping, signal mapping and the complex gain system of the amplifier. With it, the errors created by the PA can be considered beforehand and will be cancelled on the output signal [9].

A DPD has excellent linearity, typically good power efficiency, but falls a bit short on the bandwidth it can cover [10].

For the current state of technologies available, the preferable solution would be to have a predistorter capable of handling wide bandwidth signals, with the highest linearization performance possible, an additional feedback loop so that it could correct any environmental changes, low implementation costs and high efficiency.

2.3 Spectrum acquisition system

This section will expose two different approaches to measure the spectrum power of the signal, being the first approach to read the power at baseband frequencies, while the second approach at an IF frequency.

For the baseband method, the RF input signal passes through an envelope detector, resulting in its baseband signal, with this output signal being subsequently filtered by a variable band-pass filter. With this filter, it is possible to go through all of the bandwidth of the signal with the purpose of measuring the resulting power and, as a result, the power spectrum of the signal can be determined.

This method relies on an envelope detector and a filter. Even though an envelope detector was a device already used on IT, with good results, choosing a suitable filtering method is a problem, as the baseband working frequencies are from 0 MHz to 20 MHz, and finding components that allow a precise variable band-pass filter with high bandwidth at low frequencies is a difficult task.

The second method uses a down-converting mixer, this would convert the RF input signal to an IF signal where this can be filtered and then have its power measured. The quality of this system relies mainly in two components. First, it is required a frequency generator with the ability to provide frequencies with good precision and stability. The mixer will then use this frequency to shift the RF signal to IF frequencies. A Phase Lock Loop (PLL) is an example of a device that fills these two requisites. The second component, responsible to filter the signal on these new IF frequencies, is a Surface Acoustic Wave (SAW) filter, the reason for this choice was due to its great Q factor as it will provide a higher resolution in the spectrum measurement.

This last method has the advantage of being able to measure a larger bandwidth signal and having better accuracy, at the cost of having a slight increase in power consumption.

2.3.1 Components

- Mixer:

  A mixer is a non-linear circuit that multiplies two input signals that may be from different frequencies to produce a new signal, which has the sum and difference of the starting frequencies (Equation (2.3)). This component's main use is to shift an input signal into a new central frequency, this process is called heterodyning.

$$\begin{aligned} &cos(2\pi \cdot f_1 \cdot t) \cdot cos(2\pi \cdot f_2 \cdot t) = \\ &= \frac{1}{2} cos\big(2\pi \cdot (f_1 - f_2) \cdot t\big) + \frac{1}{2} cos\big(2\pi \cdot (f_1 + f_2) \cdot t\big) \end{aligned} \tag{2.3}$$

A down-converting mixer can be used to shift the input RF signal into a new IF frequency which is easier to work on with filters. This can be accomplished with the help of a PLL which provides the needed local oscillator to make this shift possible.

In the end, the resulting frequency from the mixer is $f_{RF} - f_{PLL} = F_{SAW}$, meaning that it will have to match the SAW filter's frequency.

- SAW Filter:

  A SAW filter is a type of a band-pass filter and can be used to eliminate the unneeded products of a mixer and provides the filtering quality required to isolate the frequency where the power of a signal has to be measured.

  An important feature of a SAW filter is the Q factor. The Q factor is defined by the ratio between the central frequency of the filter and its bandwidth. The SAW filter is an adequate band-pass filter, it is cheap, small, has low power consumption, and fulfils the requirement of a high Q factor.

- VGA:

  A Variable Gain Amplifier (VGA), as the name implies, is an amplifier where its gain can be controlled with an external input which allows the gain to be decided in real time, based on the working conditions necessities.

  Some applications require a high range of power readings, but for some of the values at the lower end of the scale, a normal power detector would be unable to detect them. Therefore, a VGA is a viable option when it is necessary to boost the signal to a level where it could be read.

- Power Detector:

  The power detector in this system takes the output signal from the VGA and calculates the Root Mean Square (RMS) power value, so it can be read through a microcontroller's ADC.

- PLL:

  A Phase Lock Loop (PLL) is a control device that generates a precise frequency. As represented in Figure 2.6, the PLL is based on a Voltage Controlled Oscillator (VCO), that generates the desired frequency based on an input voltage, at the cost of being inaccurate. To combat this, a feedback divider matches the output frequency (Fout) into the reference frequency (Fref), so that when closing the control loop to a Phase Detector (PD), the signals can be compared and therefore provide the error between the current output and the desired frequency. Then, this output signal error is amplified to the corresponding VCO's needed voltage through the Charge Pump (CP). Lastly, it has a Loop Filter so that sudden changes to the signal do not create massive ripples on the frequency, thus helping the VCO to converge to the desired frequency in a smoother way [11].

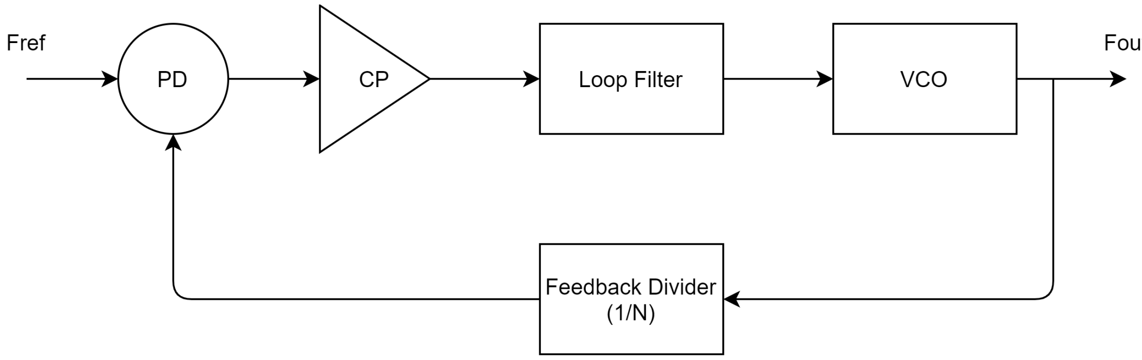


Figure 2.6: PLL Core Block Diagram.

- Microcontroller:

  A microcontroller can be used as the brain of any system. It has modules that can communicate with external devices and, using as examples the components discussed in this section, it can control both a VGA's gain and a PLL's frequency. Moreover, with the help of an ADC, a microcontroller can read measurements such as ones given by a power detector. This means that a microcontroller is able to adjust the control values as a system evolves over time.

Chapter 3

Spectrum Acquisition System Design

In this chapter, it will be presented the choices done during the conception of the design system and also the directives that led to purchase suitable components.

To obtain the final system, it was necessary to take into account the boundaries established beforehand: The end result has to be a working prototype, with the theoretical characteristics discussed on the state of art and needs to work with the constraints imposed by the equipment available on the IT.

Since the main objective of this work is to have a functional prototype, of the two methods discussed in the state of the art chapter, the one that used a down-converting mixer was followed, as it had the best chance of having results from a first research.

This system is based on a mixer, a PLL and SAW filters. To make it, it was also necessary to add a VGA, a power detector, and a microcontroller. A block diagram of the designed system can be consulted on Figure 3.1.

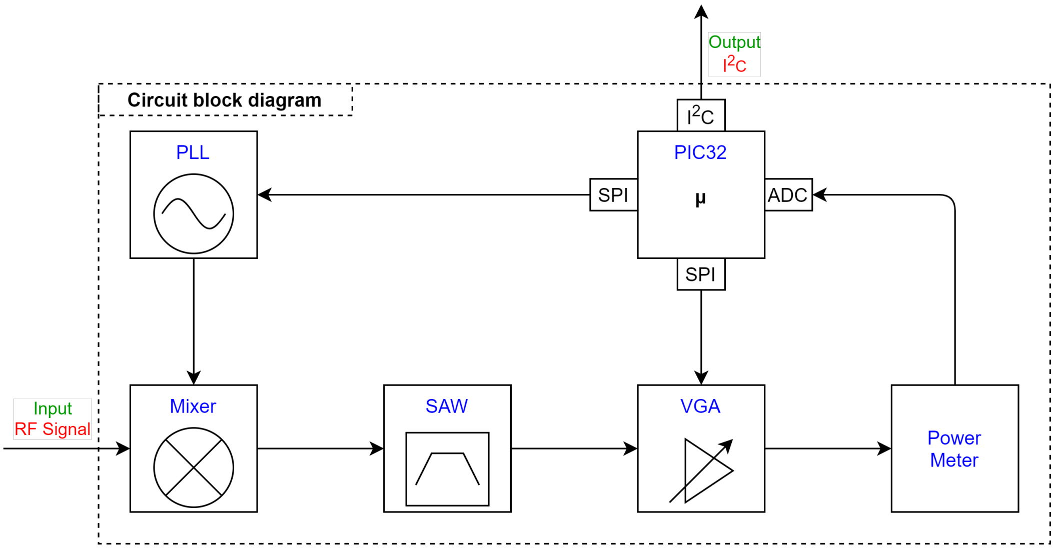


Figure 3.1: System block diagram.

3.1 Electronic Components

The main limitation related to the construction is imposed by the ability of the hardware machine(CNC) to make a PCB, as the actual technology on IT allows only the production of a PCB with two layers, the top and bottom layer. This constrain mainly impacts the design of the layout because all the components should have a good path placement so that no lane will overlap with each other during the implementation.

It is also a target trying to achieve the smallest prototype possible, as it will translate into less waste of substrate and so a cheaper product. However, this cannot conflict with the cleanliness and reliability of it.

Other constraints include the minimum size of the CNC's drill, 2mm, therefore the layout design must have, at least, the same distance between two copper lanes. One of the main drawbacks of this is that any device used must have at least two millimetres between the pads.

There are other limitation less relevant than the above mentioned, such as the soldering, this has to be done by the technician of IT, and therefore the components cannot be very small, or it will be very difficult to solder them by hand.

For the circuit elements properties, it should be taken into account their power consumptions, the signal power that they can handle and their noise figure, as these directly compromise the quality of the final product.

The last criterion was the price, the cheapest components were selected, and bought on a store approved by IT.

With the above requirements met, it was chose a component that could suit the theoretical and practical constrains.

For reason mentioned above, the circuit elements chosen were:

3.1.1 SAW Filter

The first element picked was the filter, as it is a central part of system circuit, that because this element dictates the quality of the measured signal, furthermore its central frequency will impose the other components frequencies.

Figure 3.2 presents the chosen SAW filter, the B39431B3790Z810 [12] from B3790 series made by RF360. As it can be seen in Figure 3.3, this filter has a theoretical centre frequency of 433,92 MHz, with a minimum insertion attenuation of 3,6 dB between 433,86 MHz and 433,98 MHz, which means it has a band-pass bandwidth of 0,12 MHz. Among the other seen filters, this filter was the one with the best Q factor at that moment.

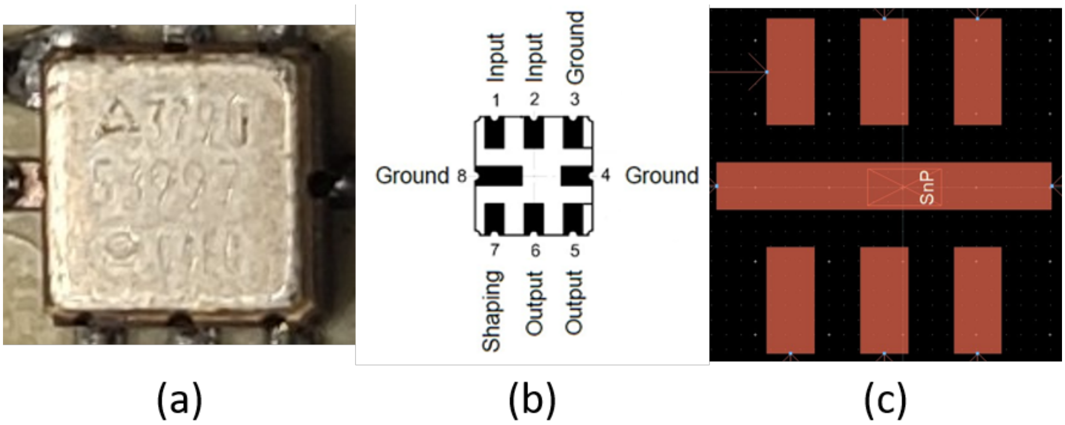


Figure 3.2: The SAW filter used represented as: (a) real Image, (b) bottom view [12], and (c) the used layout.

Another key factor for the filter choice was having a good attenuation along the band-stop, this important characteristic was not found on other available filters.

To achieve a better filter, it was considered the use of two different filters in cascade where one would have an excellent Q factor and the other would have very good attenuation on the band-stop, but no combinations were found that could be better than the B39431B3790Z810.

Furthermore, the filter alone cannot achieve the desired characteristics, as the signal should be 90dB below the transmitting signal, it is desired that the system should be able to measure a dynamic power range of 90dB, which for this case, the filter has to have the stopband with 90dB lower than the band-pass and to achieve that, two filters were used. In this case it was used the same filter two times in a cascade.

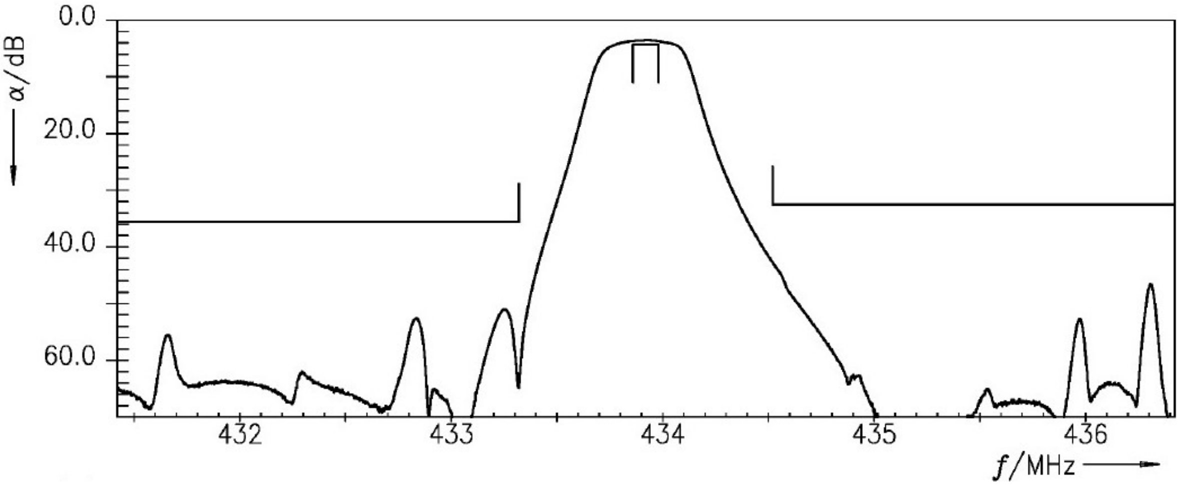


Figure 3.3: Transmission coefficient of used SAW filter [12].

3.1.2 PLL

The Phase Lock Loop generates the frequencies responsible to shift the RF signal into the frequencies that the system circuit works, which in this case is 434 MHz, since this is the central frequency of the SAW filter. Therefore, the PLL needs to generate frequencies around the 2000 MHz minus the 434 MHz, what resulting frequencies around the 1566 MHz.

The other requisite the PLL is also responsible for, is the ability to reach the 20 MHz bandwidth with a maximum step of 1 MHz between two frequencies.

With the feature mentioned above the PLL that suited better was the ADF4360-4 from Analog Devices, it is composed by an Integrated Synthesizer and an internal VCO, meaning that it just needs an external oscillator to work properly.

The ADF4360 series is composed by four devices, with each one operating in different frequencies. Among them, there is the ADF4360-4 [13] represented in Figure 3.4, with an output frequency range from 1450 MHz to 1750 MHz, which includes the desired frequencies around the 1566 MHz and more than the 20 MHz of bandwidth, furthermore this PLL can achieve a channel spacing with less than 1 MHz as it is able to get steps smaller than 0.1 MHz, this was the feature that led this PLL to be chosen among those available.

The ADF4360 series have similar characteristics, which is interesting for future projects, since the defining difference between them is the output frequency range of the PLL, the system could measure higher RF input frequencies with minimal changes to the whole system.

One last attribute to be mentioned is the PLL control's communication, the ADF4360-4 uses a compatible SPI communication, this is an advantage as it uses less lines when compared with other devices that use parallel communication.

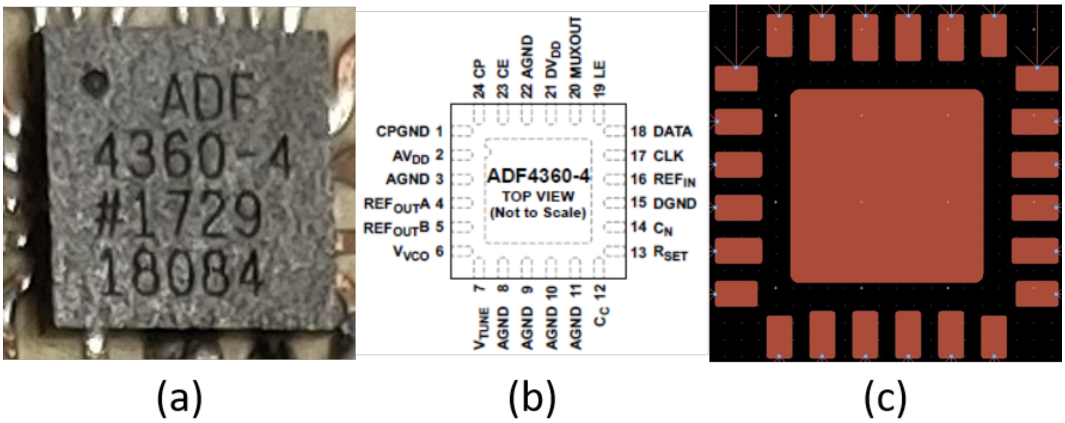


Figure 3.4: The PLL used represented as: (a) real Image, (b) top view [13], and (c) the used layout.

3.1.3 Mixer

The mixer is responsible to shift the input RF signal to the pretended 434 MHz. Its characteristics are predetermined by the PLL output signal, the RF signal that needs to be analysed and the SAW filter frequency.

By taking the above features into account, the mixer LT5557 [14] was chosen, made by Linear Technology.

This mixer is optimized for high linear down-conversion and depending on the output matching it can be capable of achieving a gain of 3 dB from RF input to IF output, furthermore it is already partially matched for frequencies up to 600 MHz (which includes the intended frequency of 434 MHz).

The datasheet proposes a few options for matching networks, with one of those being a perfect fit with all of the presented requirement above met. Thanks to this, the LT5557 was considered the best available choice.

Figure 3.5 presents the chosen mixer:

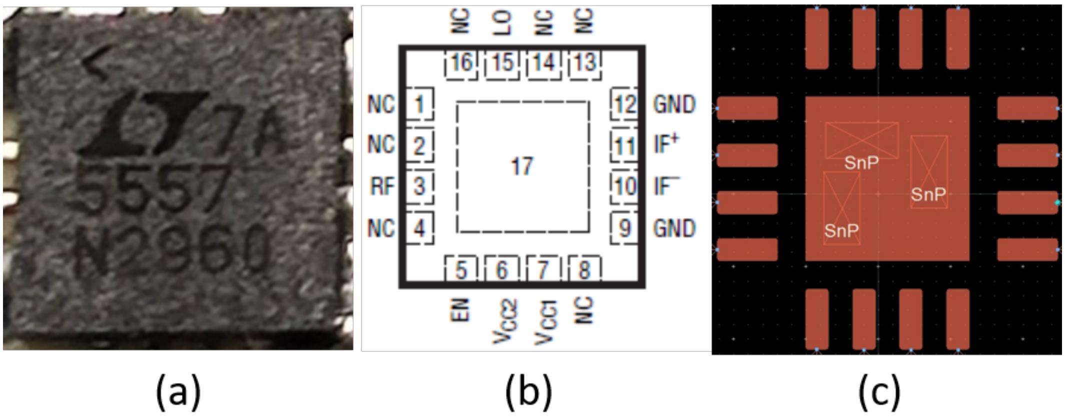


Figure 3.5: The Mixer used represented as: (a) real Image, (b) top view [14], and (c) the used layout.

3.1.4 RMS Power Detector

The LTC5582 [15] presented in Figure 3.6 is an RMS power detector, from Analog Devices, since the most important feature on the power meter is the dynamic range that can be measured, LTC5582 presents a theoretical range from -60 dBm to 2 dBm.

Another relevant feature of this power detector was its output, the LTC5582 generates an output between 0.5 V and 2.5 V, whose values are inside the ADC range of the microcontroller. Furthermore, it generates a linear output, which facilitates the conversion from voltage read on the ADC to the corresponding power measured, allowing less processing power usage during the conversion.

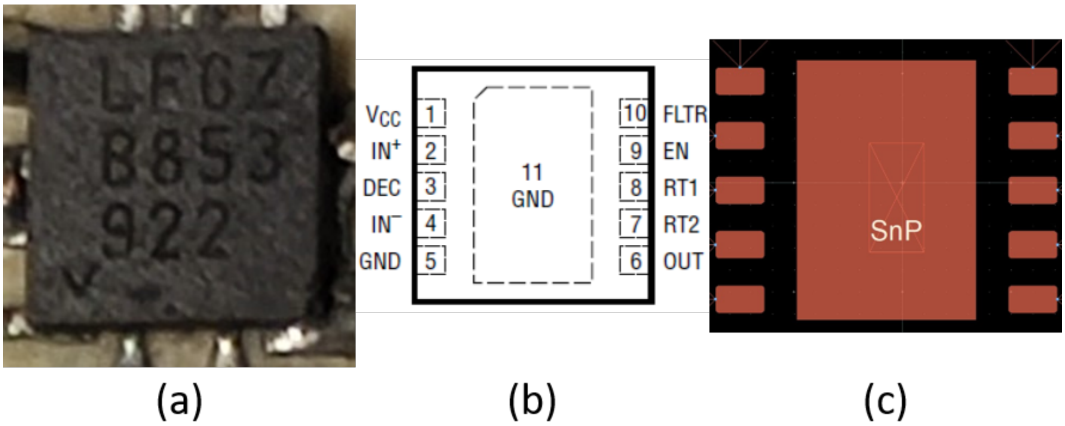


Figure 3.6: The RMS Power Detector used represented as: (a) real Image, (b) top view [15], and (c) the used layout.

3.1.5 VGA

The VGA has a great impact on the dynamic range of the system, therefore the main feature needed is the maximum gain at the intended frequency of 434 MHz. Most of the VGAs had a low-pass behaviour, so it was important to not only see the gain values presented on the datasheet, but also the gain response along frequencies.

The VGA chosen was the AD8370 [16] from Analog Devices, with a theoretical dynamic gain range from -11 dB to 37 dB, and a theoretical bandwidth of 750 MHz. An image of this VGA is presented in Figure 3.7.

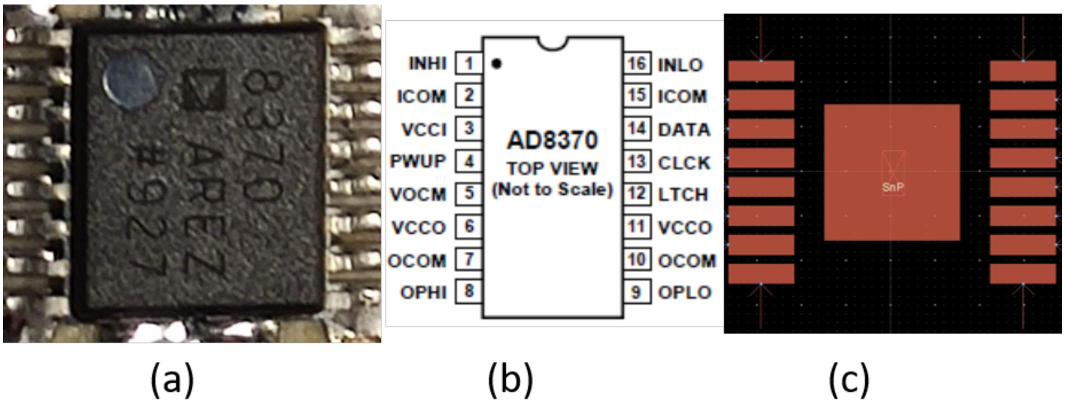


Figure 3.7: The VGA used represented as: (a) real Image, (b) top view [16], and (c) the used layout.

3.1.6 Microcontroller

The Figure 3.8 presents the microcontroller used, a PIC32MX274F256B [17] from Microchip. The Microchip microcontroller was chosen mainly because it was used previously in other projects, therefore it is a known microcontroller leading to an easy learning curve. This microcontroller requires another device to flash the program, but there was already a PICkit 3 [18] available, so it did not pose any problem. Among the Microchip controllers available, it was selected a microcontroller with a SPI, Universal Serial Bus (USB) and I^{2}C interfaces and that also respects the constrains referred at the beginning of this chapter. This microcontroller has the Peripheral Pin Select (PPS) technology on its most of Input/Output (I/O) pins, which allows configuration of digital peripherals to any I/O pin and also enables routing a peripheral to multiple pins, thus facilitating the layout. At last, it was decided to use a high-performance microcontroller, to guarantee that performance is not an issue on this prototype.

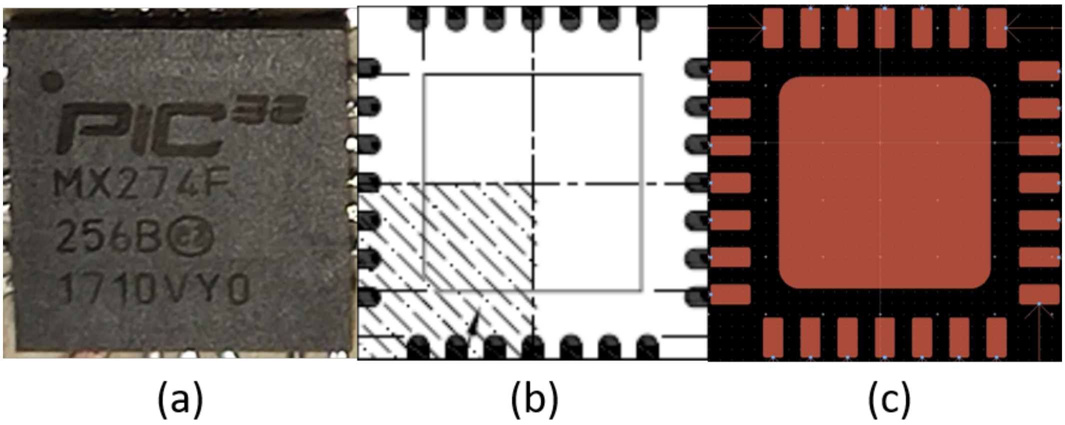


Figure 3.8: The microcontroller used represented as: (a) real Image, (b) bottom view [17], and (c) the used layout.

The PIC32MX274F256B has integrated interfaces and modules, for this thesis the relevant are the following:

- SPI:

  The SPI interface is used to communicate with PLL and VGA, the communication is done in parallel, with the device being selected through two pins, each one functioning as a chip select for the respective device. Furthermore, it was used only one pin port to source the clock signal (CKL) for the both devices, while for the output data signal (SDO) it was used two separates pins, this being possible by configuring the two pins as the same internal port through the PPS [19].

- I^{2}C:

The I^{2}C interface is used as an output communication, its objective is to use it to control a predistorter's DACs. This communication requires two pins, one for a clock signal (SCL) and another for the data signal (SDA), both pins were selected through PPS.

So that the system would have reliable communication it was added to the layout one ground pin and one 3.3 voltage pin, this can supply the DACs if needed and the ground pin can be used as a reference, ensuring the adequate voltage needed for the communication.

- USB:

  The USB was the communication opted to send the prototype results to the PC, in this way an application can easily send control commands to the microcontroller, and also receive a visual output.

  For the USB communications it was added a micro-USB port, this port not only allows communication but also it is used to power the entire system.

3.1.7 Oscillators

In this system there are two devices that require oscillators, the PLL and the microcontroller. They need different types of oscillators since they have different ways to attach to them. The first oscillator will generate the "base" signal for the PLL and it will need an external power supply. Also, this oscillator's goal is to aid the PLL in generating frequencies that are at GHz level, as these frequencies impact the accuracy of the system, this oscillator must be as precise as possible and with a low drift.

For that reason, it was selected the XNCLH10M000CHJA2P0 [20] from Murata, it is a precision Temperature Compensated Crystal Oscillator (TCXO), with a temperature compensated circuit, also has the pretended specifications, it has an operating temperature range from -40 °C to 85 °C, a frequency shift by temperature of 0.28 ppm, and a frequency aging of 0.28 ppm/year.

The XNCLH10M000CHJA2P0 oscillator, represented on Figure 3.9, needs a supply of 3.3 V and works at a frequency of 10 MHz.

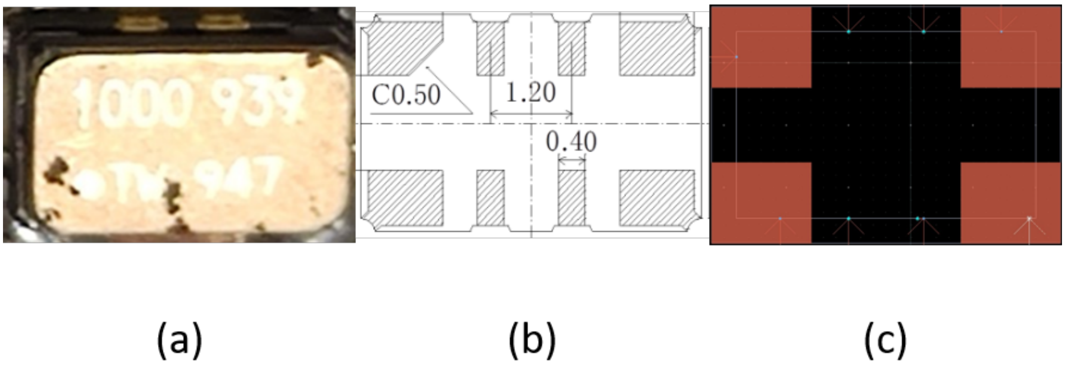


Figure 3.9: The PLL's oscillator used represented as: (a) real Image, (b) top view [20], and (c) the used layout.

In Figure 3.10 the second oscillator can be seen, this one is used to supply the microcontroller's clock, unlike the oscillator used on PLL, the power to supply this oscillator it is given by the microcontroller itself. Moreover, the main reason to use an oscillator attached to the microcontroller is to generate the clock for the USB communication, the USB communication uses a clock in the MHz order, so the oscillator does not need to be so good as the PLL clock.

The oscillator chosen is a ABM10-166-12.00MHz [21], which is a cheaper choice when compared with the PLL's oscillator. This oscillator, as the name implies, operates at a frequency of 12 MHz, and have a load capacitance of 8 pF.

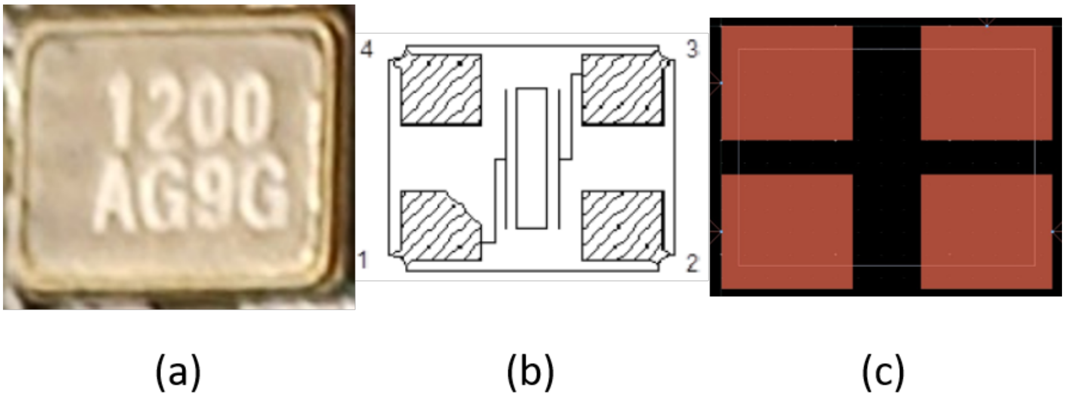


Figure 3.10: The microcontroller's oscillator used represented as: (a) real Image, (b) top view [21], and (c) the used layout.

3.1.8 Other Components

There were other components used to materialize a functional system, the most common were the capacitors, inductors, and resistances.

Beyond these there were other components worth to be mentioned:

- Headers / I/O Connectors:

  The headers pins were used to provide the I^2C communication with the external DAC's, and another group of I/O pins to attached to the programmer.

- USB Connector:

  For the USB connector, it was employed an USB type B inverted, the choice fell on an inverted connector because it facilitates the connection between the D+ and D- pins of the microcontroller and the connector, so that the lines do not cross over each other.

- SMA Connector:

  The SMA Connector is the physical port where the RF signal enters the system, this connector is matched to 50 Ω and works with frequencies around the 2 GHz.

- LDO:

  The Low-Dropout Regulator (LDO) ensures a stable voltage of 3.3 V converted from the 5 V provide by the USB. Several LDOs were used, one for each device, to ensure proper regulated voltage and a better isolation between the devices.

- Button:

  A small button was implemented to permit a physical reset on the microchip.

- PICkit 3 [18]:

  Figure 3.11 shows the used PICkit 3. As mentioned above, the PICkit 3 is an in-circuit Programmer and a Debugger, it was chosen to program the microcontroller.

  It is simple to integrate on the system, its software is already integrated on the microchip Integrated Development Environment (IDE) (MPLAB), with drivers automatically installed on Windows. As for the hardware required, a few copper lines and header pins were added to the prototype board so that the PICkit 3 can be easily connected to the microcontroller. A simple mini-USB type B cable is then used to connect to the computer to upload the code.

Figure 3.11: PICkit 3.

3.2 Designed Circuits

For a working system, the devices need an appropriate circuit design. The way to achieve that, it was by starting to follow the datasheet recommendations and when it was available, follow the schematics guidelines, always considering the differences between the presented projects and this thesis project.

Most of the devices used, have the enable chip feature, which can be controlled through an I/O pin, this would be preferable to use so that it could be controlled by the microcontroller, but due to constrains imposed by the two layers of the substrate, it was decided to not use it. Instead, the solution was to leave these pins be connect to 3.3 V turning its always on.

Another problem arose in some passive components, as the sought impedance, capacitance and inductance values were not always available to purchase. Usually, these values have a normalized reference value so, the solution was to trace the two nearest available components values and go for the one that had closest results.

Hereupon, it will follow specific for each device:

3.2.1 PLL

For the ADF4360-4, the Analog Devices provides, along with the datasheet [13], an evaluation board user guide [22], and a software, named ADIsimPLL, where a PLL loop filter can be simulated. The evaluation board user guide contains information about how to use their board and how to control the evaluation board through their software, moreover, another important factor is that it also contains a schematic of the evaluation board, which it was used to guide this thesis design.

To describe the decisions made during this circuit design, the approach will be to divide it into four separates parts, being the first the supplier circuits, the second is the input circuit composed by the oscillator, the third part is the PLL loop filter, and finally the fourth part, the output matching, which will be discussed in the next section.

The supplier circuits were designed by using a LDO to convert the 5V into to the required 3.3V, this LDO can supply a stable 3.3 V and can handle the maximum current of the PLL of 47.5 mA, furthermore, by following the datasheet [13] recommendations it was used two LDOs, one to power the PLL through its VDD pin, and another to supply the VCO through the VVCO pin and with the addition of two capacitors, one of 100nF and another with 10 pF, a reliable power supply was created with a good filter to prevent the non-wanted signal.

For the oscillator circuit, as the chosen oscillator has characteristics similar to the used on their evaluation board, the schematic proposed in the evaluation board user guide was used [22]. The important part of this schematic was edited and presented in Figure 3.12.

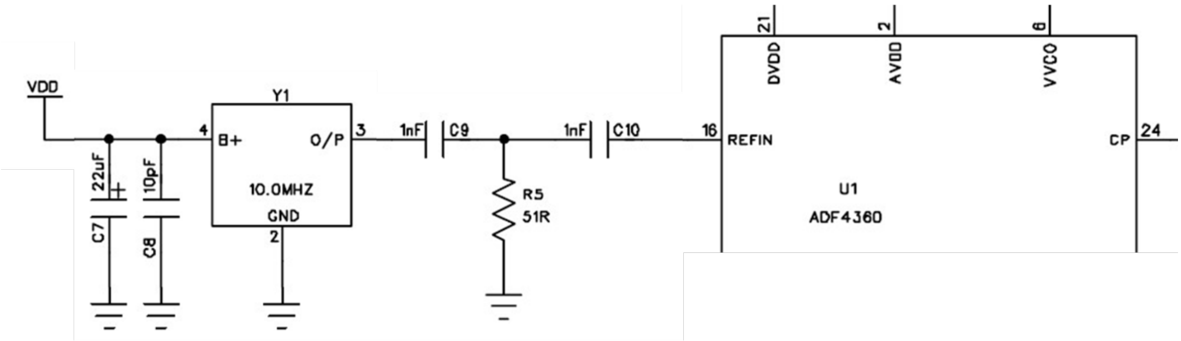
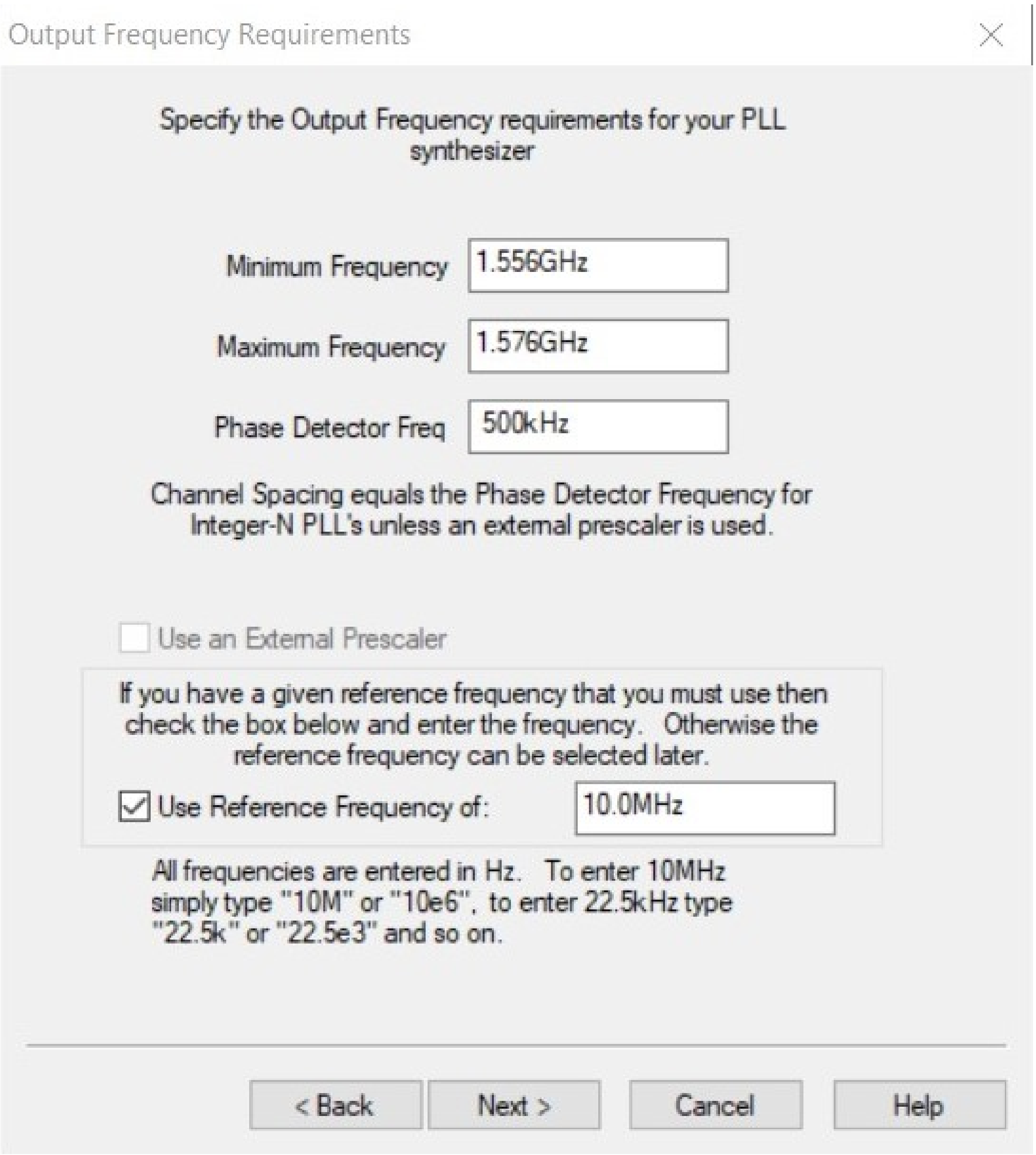


Figure 3.12: Reference input circuit for the PLL [22].

Finally, for the loop filter, it was used the ADIsimPLL software.

The ADIsimPLL was set with the predetermined PLL parameters, working with a central frequency of 1566 MHZ and 20 MHz of bandwidth, additionally as the Phase Detector Freq. parameter decides the channel spacing of the PLL, it was settled for 0.5 MHz, this value is well bellow 1 MHz which is the maximum spacing requirement for this thesis. If needed, the spacing can also be later changed through software to an even lower number as the PLL can support spacing lower than 0.1 MHz. The last parameter is the oscillator frequency which is 10 MHz.

The configurations mentioned above were set in the ADIsimPLL software and they can be seen in the following Figure 3.13.



Output Frequency Requirements

Specify the Output Frequency requirements for your PLL synthesizer

Minimum Frequency: 1.556GHz

Maximum Frequency: 1.576GHz

Phase Detector Freq: 500kHz

Channel Spacing equals the Phase Detector Frequency for Integer-N PLL's unless an external prescaler is used.

□ Use an External Prescaler

If you have a given reference frequency that you must use then check the box below and enter the frequency. Otherwise the reference frequency can be selected later.

☑ Use Reference Frequency of: 10.0MHz

All frequencies are entered in Hz. To enter 10MHz simply type "10M" or "10e6", to enter 22.5kHz type "22.5k" or "22.5e3" and so on.

Figure 3.13: ADF4360-4 working frequency configuring on the ADIsimPLL.

As a result, the software generates a schematic of the loop filter as seen in Figure 3.14, on another tab it shows a simulation of the noise and gain, and in the last tab a report is created, where it shows other characteristics of the generated sine signal, such as the noise present and the time needed to lock on the frequency.

The dimensioned components resulted from the simulation are not available to be purchased with their current values, so there was a need to adjust their values for the nearest purchasable components. These new values were added to the simulations and in Figure 3.15, it is possible to consult the new results.

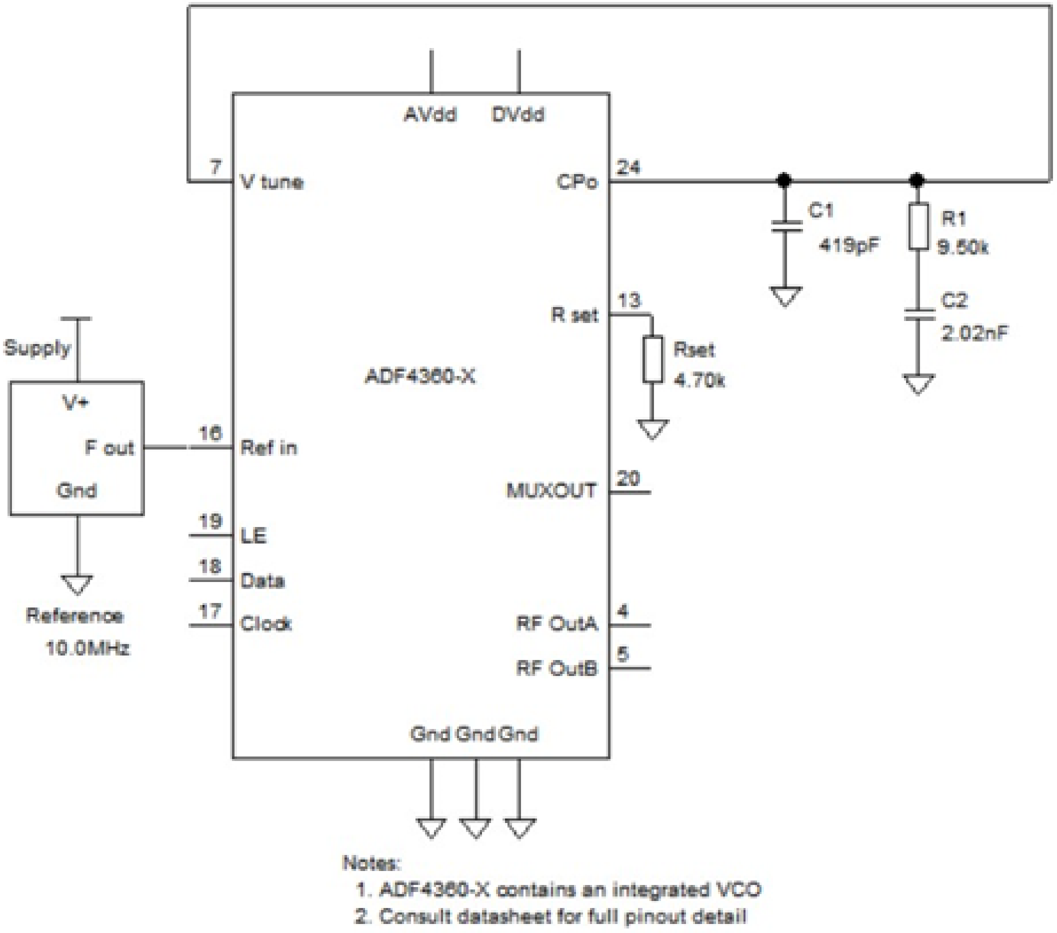
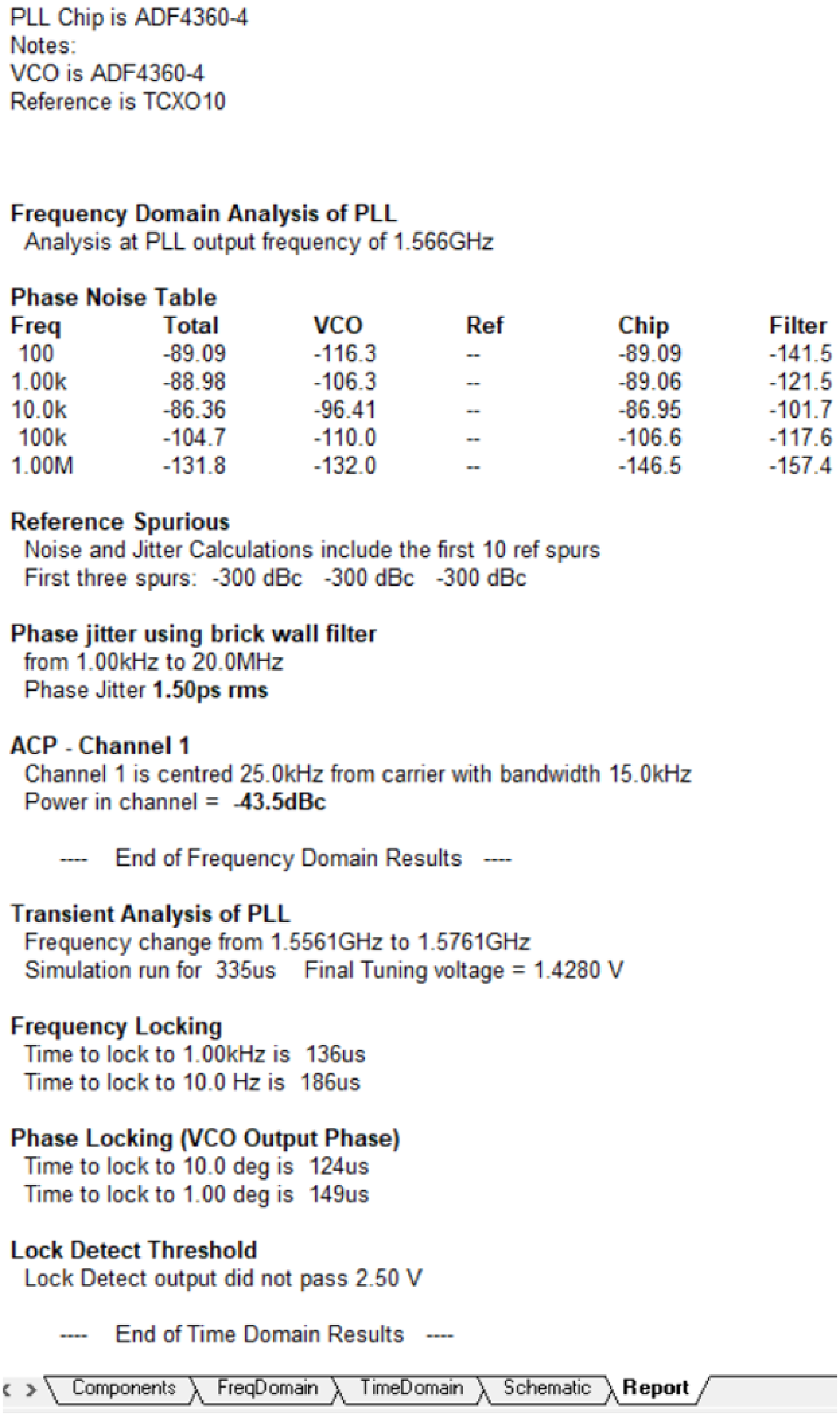


Figure 3.14: PLL loop filter generated by ADIsimPLL.



PLL Chip is ADF4360-4
Notes:
VCO is ADF4360-4
Reference is TCXO10

Frequency Domain Analysis of PLL
Analysis at PLL output frequency of 1.566GHz

Phase Noise Table

Freq	Total	VCO	Ref	Chip	Filter
100	-89.09	-116.3	--	-89.09	-141.5
1.00k	-88.98	-106.3	--	-89.06	-121.5
10.0k	-86.36	-96.41	--	-86.95	-101.7
100k	-104.7	-110.0	--	-106.6	-117.6
1.00M	-131.8	-132.0	--	-146.5	-157.4

Reference Spurious
Noise and Jitter Calculations include the first 10 ref spurs
First three spurs: -300 dBc -300 dBc -300 dBc

Phase jitter using brick wall filter
from 1.00kHz to 20.0MHz
Phase Jitter **1.50ps rms**

ACP - Channel 1
Channel 1 is centred 25.0kHz from carrier with bandwidth 15.0kHz
Power in channel = **-43.5dBc**

---- End of Frequency Domain Results ----

Transient Analysis of PLL
Frequency change from 1.5561GHz to 1.5761GHz
Simulation run for 335us Final Tuning voltage = 1.4280 V

Frequency Locking
Time to lock to 1.00kHz is 136us
Time to lock to 10.0 Hz is 186us

Phase Locking (VCO Output Phase)
Time to lock to 10.0 deg is 124us
Time to lock to 1.00 deg is 149us

Lock Detect Threshold
Lock Detect output did not pass 2.50 V

---- End of Time Domain Results ----

Figure 3.15: PLL behaviour result from a simulation made on the ADIsimPLL.

3.2.2 Mixer

The mixer circuit needs to take into account its two input ports, the RF input and the LO input, the IF output ports, which will be discussed in the next section, and at last the power supply.

The RF input is single-ended internally matched to 50 Ω for frequencies around 2 GHz, which are the intended work frequencies, furthermore the tests present in the datasheet use a input RF power between -15 dBm and 12 dBm at the same frequencies, which includes the intended -10 dBm. Since its impedance is already matched, there was no need to add any additional component, but for a better circuit protection, a capacitor was added so it would create a DC cut, by taking into consideration the 2 GHz of the input signal, it was added a capacitor of 1 nF.

The LO input is supposed to be fed by the PLL, the PLL generates frequencies around 1566 MHz where the mixer is internally matched with 50 Ω. Another important matter is the power delivered by the PLL, it can provide more than -3dBm which is what the mixer requires for its proper operations.

Lastly, like other devices, the power supply was provided by the LDO, with additional decoupling capacitors to improve the filtering, as recommended by the datasheet these capacitators have the value of 1 nF and 1 µF [14].

3.2.3 SAW Filter

The SAW filter is a passive component, which means it does not need any power source to work, its only requisite was to add an inductor of 56 nH on the shaping pin [12].

3.2.4 VGA

As can be seen in Figure 3.16, the VGA AD8370 has a bandwidth of 750 MHz where it has a gain drop of 3 dB. This component works with either the supply of 3 V or 5 V and if it is supplied with 3 V, it should only have a minor performance degradation. But according with the datasheet specification, the use of a low supply such as 3 V will reduce the bandwidth of the VGA by 100 MHz which is a problem. Since with a supply of 5 V the gain of VGA starts to drop when slightly above the 400 MHz, on 3 V a bandwidth reduction would mean an effective attenuation of the gain at the intended work frequency (434 MHz), for that reason and because the 5 V supply was readily available, it was decided to use the direct supply of the USB.

More than that, the VGA has three supply ports, therefore, to use 5 V to supply each port it was necessary to add two decoupling capacitors near of each port, one of 1 nF and other with 1 µF [16].

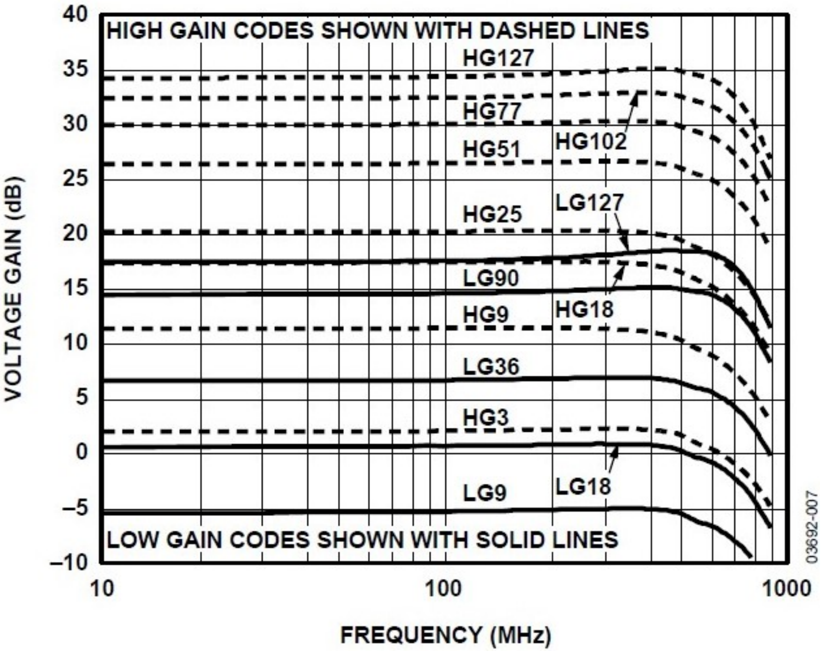


Figure 3.16: VGA frequency response vs. gain code [16].

3.2.5 RMS Power Detector

The circuit to the power meter LTC5582, like other devices was created by following the datasheet recommendation.

The power meter was supplied by the LOD with 3.3 V with a decoupling capacitor of 1 µF added. As recommended, the Input Common Mode Decoupling Pin (DEC pin), was left unconnected to facilitate the layout, this was possible since it was already internally decoupled by a 50 pF capacitor [15].

To obtain a stable operation, the LTC5582 needs an additional capacitor connect to the FLTR pin that when coupled with its internal components, creates a filter that stabilizes the AC average power measurement. The higher capacitor used, the less ripple with be present, but it will also increase the output transient times as a trade-off. For this project, it was used a 100 nF capacitor, as it meets the minimum value requirements and provides a smooth ripple.

Lastly, to reduce the residual ripple, it was added at the output pin a 1 nF capacitor, thus creating a stable voltage that can be read by the microcontroller ADC.

This power meter has two additional pins, the RT1 and RT2, whose function was used to compensate errors on the output signal caused by temperature fluctuations. Through this pin and by knowing the current temperature of the circuit it could be precisely corrected the output voltage deviation. They were set to the ground, so to disable them as its usage would require extra circuits to be developed and the errors caused by the temperature were not that severe that made this a priority.

3.2.6 Microcontroller

The Microcontroller PIC32 had four pins that needed to be connected to 3.3 V. A LDO regulator was used to provide that, as the power supply available to the system was through the USB port that had 5 V.

The VDD pin, the pin who supplies the microcontroller, and the VUSB3V3 pin, the USB internal transceiver supply, were both directly connected to the 3.3 V. The AVDD pin, the pin that supplies the ADC, it was added two decoupling capacitors, one 100 nF and another with 1 µF [17].

For the last pin, the MCRL pin, it was added a button that allows a quick hardware reset, this pin was attached to 3.3 V through a resistor, so that when it goes to ground it resets the pic. As it is recommended on the microcontroller datasheet, it was also added a capacitor of 10 µF to the Vcap pin, this capacitor works as an external filter and helps maintaining the stability of the internal regulator [17].

To program the microcontroller, as mentioned before, it was used a PICkit 3. As it can be seen in Figure 3.17, this programmer has six pins, but it only needs 5 to program, these five pins are all directly connected with the respective pins of the PIC32, its data pins with the same length for it to work properly. The MCLR/VPP are connected to the MCLR of the PIC, the VDD connected to the same supply voltage of 3.3V, the VSS is directly connected to the ground and for the last two used pins, the PGD pin was connected to the PGED1 of the pic, while the PGC was connected to the PGEC1, leaving the last pin, the PGM to float [18].

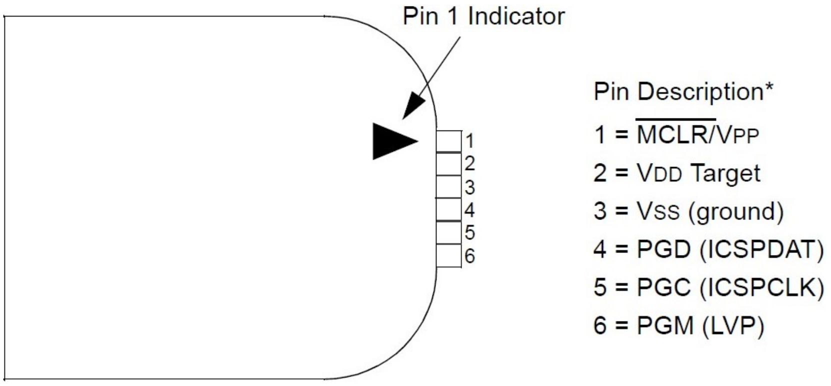


Figure 3.17: PICkit 3 ports [18].

The microcontroller is also connected to other devices, such as the PLL and the VGA through its SPI interface, they are all directly wired, while being careful that the connections all have the same length.

The PLL also needs an output connection, it comes from its mux, this is a lock detect, which will trigger a voltage when the PLL reaches the desired frequency, for that reason this pin is connected to a microcontroller pin, that allows external interrupts. On top of this, to limit the current on this lane, two resistors were added.

For the USB communication, it was connected directly to the respective pins of the PIC32. The VCC of the USB connector goes to the VBUS of the PIC32, so the PIC32 can detect that there is an USB is connected, moreover this pin is also used to power the entire system. The D+ and D- connect to the respective D+ and D- of the PIC32. The ID pin is connected USBID of the PIC32, this pin passes the ID of the system, so the PC can recognize the PIC32. Finally, the GND pin was connected to the ground of the system.

The last step regarding the microcontroller's circuit, is the oscillator connection to the OSC1 e OSC2 pins of the PIC32, for it to work properly, the oscillator needs two capacitors to oscillate at the intended frequency, by taking into account the datasheet recommendation it was used a capacitor of 10 pF in both ports.

3.3 Matching Networks

The matching networks have an important impact in the system since the component's impedances need to match each other so they can operate properly at the desired frequencies. The impedance matching impacts the power transfer or signal reflected from one device to another, a good matching will improve the efficiency and the power range of the system. Lastly, the matching of the SAW filter also conditions its properties.

Most devices have in their datasheets some recommended matching networks, mainly the 50 Ω matching at the work frequencies. If possible, these matchings are to be used whenever it suits the system implemented. As some devices' ports do not have 50 Ω impedances, a direct matching between the pair of ports was chosen instead of doing two 50 Ω network matching in a row. This approach decreases the number of elements needed to match the two ports, which decreases the size of the network match and also reduces the losses generated from using non-perfect components.

Some devices have alongside its datasheet, the port impedance values, which allows to do port simulation on a network matching [14][12][16].

The ADS software can simulate ports with the SP parameters acquired from the port impedance values. It can also optimize the values of the components that compose the network matching, this is made through an iterative optimization of the design parameters to achieve the proposed goals. For the optimization it was chosen to do a discrete step on the values and, since the components are sold with listed values, this step will make the optimization run faster without having any negative impact.

3.3.1 PLL to Mixer

The first network matching circuit is the PLL output to LO input of the mixer, this input is a single-ended port internally matched to 50 Ω and since the output frequencies are inside of the LO bandwidth the network is matching to 50 Ω.

The PLL's datasheet has three operation output matching to 50 Ω examples, and since there was no need to use a differential output, the third matching was chosen. In Figure 3.18 this matching is presented, and shows that this solution uses both the A and B output of the PLL, thus combining both outputs through a lumped-lattice-type LC balun to achieve a better power transfer to the LO [13].

This network matching can deliver 4dBm, which is more output signal power than the -3dBm that the mixer needs, but it can be adjusted with the help of software so to save some supply power[13].

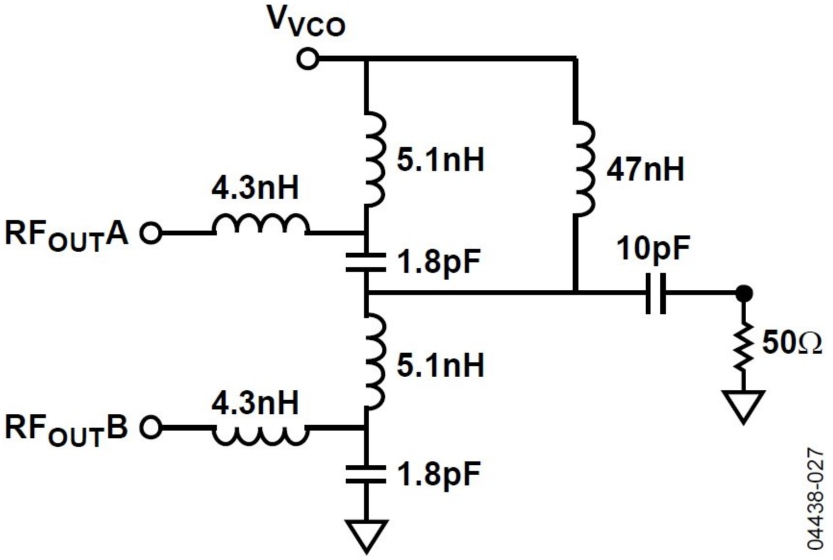


Figure 3.18: LC balun used to match the PLL output [13].

3.3.2 Mixer to SAW Filter and SAW Filter to VGA

The second network matching connects the mixer to the cascade SAW filter and then the second SAW filter to the VGA, this matching is not only important because of the impedance matching between the three components, but because it also defines the filter characteristics. For all these matching networks discussed in this sector, it was taken advantage of the device's ability to be differential ended as this type of output provides a better power transference.

The network matching between the IF output of the mixer and a SAW filter was already represented in the mixer datasheet, at the beginning it was used the same suggested design and, with the help of the optimizing tools available on Advanced Design System (ADS) software, tried to adjust the passive components in order to improve the matching.

In time, it was found that due to the high impedance of the SAW filter, the proposed network configuration would never achieve the desired impedance. As an alternative, a new matching network was created with the help of a smith chart.

Figure 3.19 represents the matching network between the mixer and the first SAW filter. This new network still maintains the parallel inductors, not only it cancels the internal capacitance of the mixer, but also it supplies power to the output signal. Furthermore, an extra capacitor was added to the SAW filters to obtain a DC cut so that the other circuits would not contaminate this matching network.

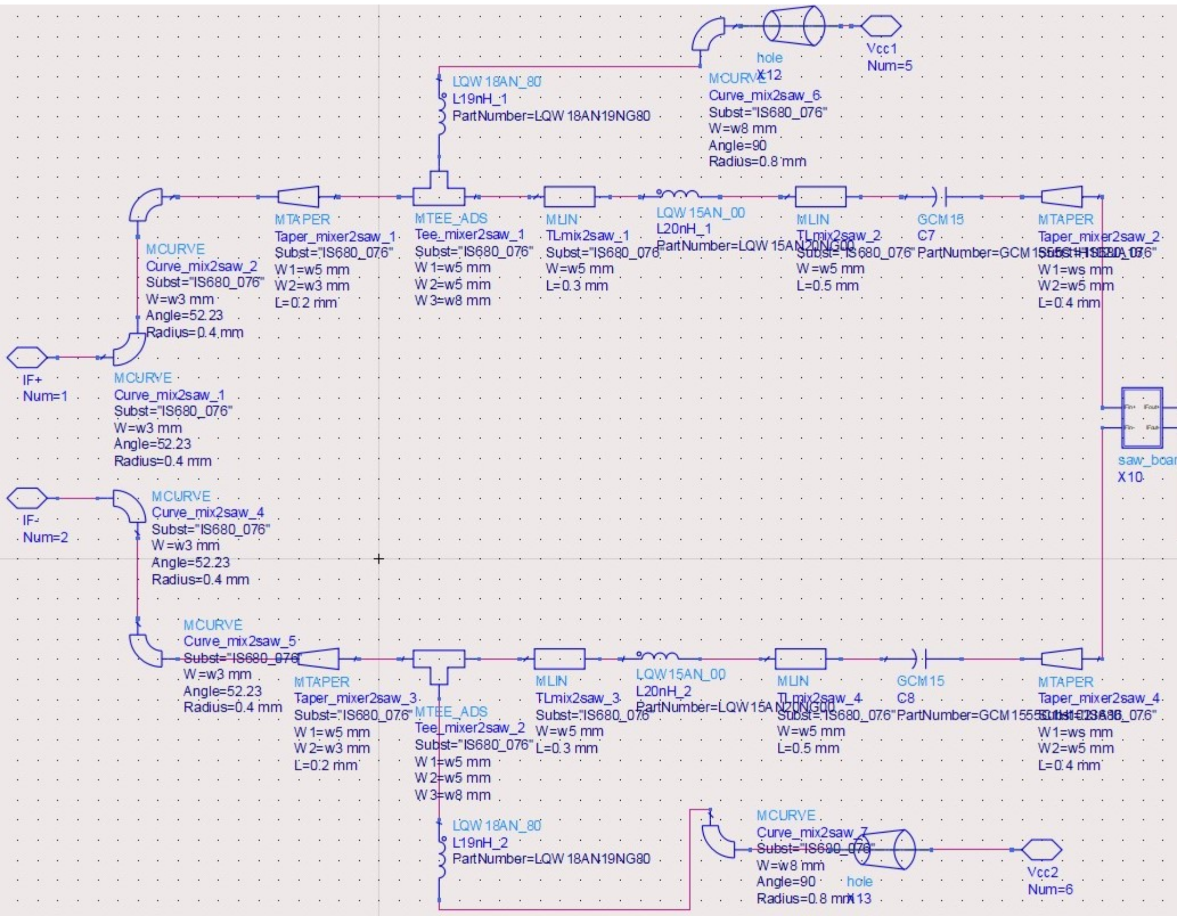


Figure 3.19: Matching network from mixer to first SAW filter Schematic.

To create the cascaded SAW filters represented in Figure 3.20, the output impedance of the first filter must match the input impedance of the second filter which in this case, it is achieved through a parallel inductance of 56 nH.

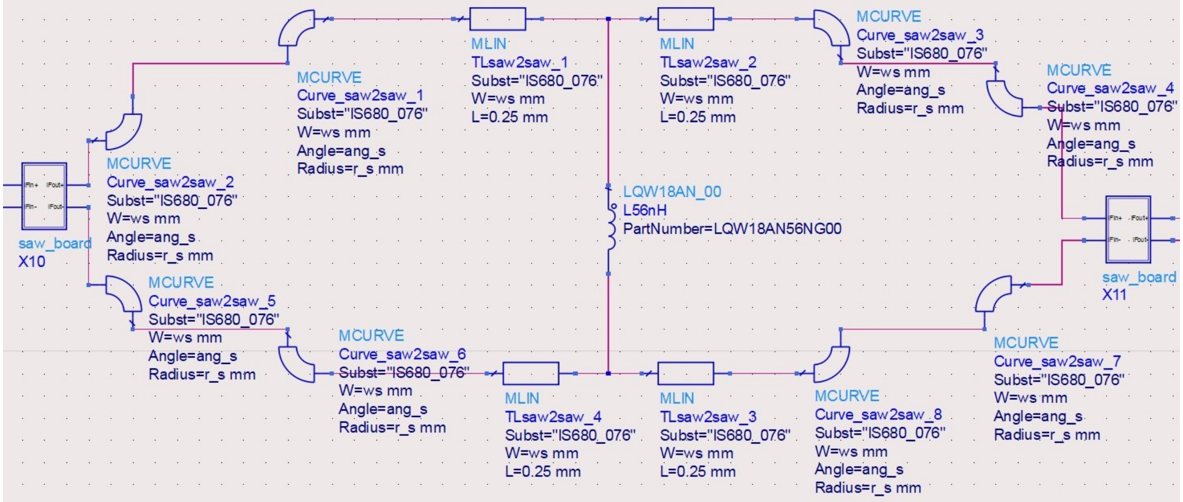


Figure 3.20: Cascade filter Schematic.

For the network going from the second SAW filter to the VGA, it was used the smith chart available on ADS to make the match at the frequency of 434 MHz, the filter's central frequency.

The impedances used were acquired through S-parameters, except for the VGA, since the impedance changes depending on the VGA's gain, it was used a 200 Ω impedance as it was recommended in the datasheet.

The above points resulted in the matching presented in the following Figure 3.21.

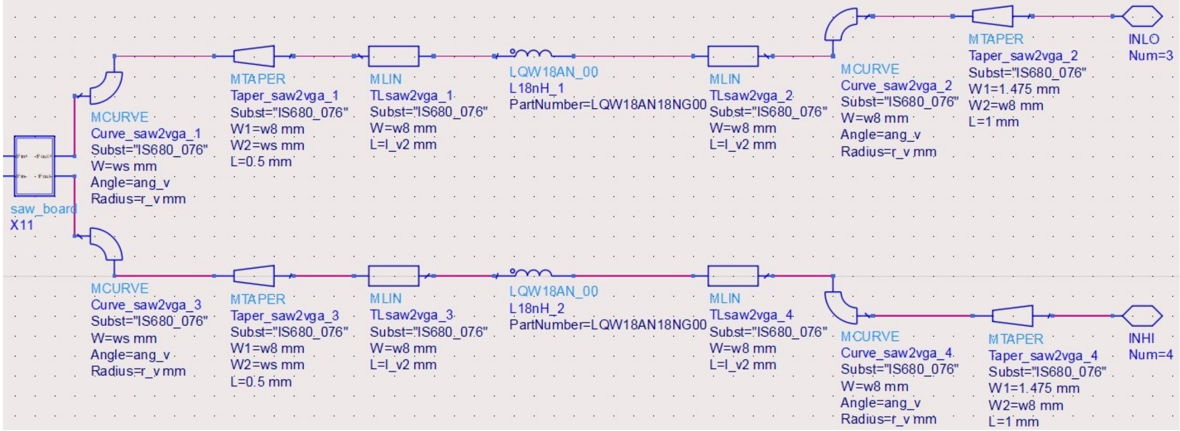


Figure 3.21: Schematic used to create the matching network between the second SAW filter and the VGA.

At last, to accomplish the characteristics needed for the system filter, a simulation was created in order to optimize the matching elements so to reach the filters goals. The goals are having power transferred to the VGA's port as close as possible to 0 dB when nearing the working frequency of the filter and having a power as low as possible on other frequencies.

The objective was achieved through the S-parameters. With S(2,1), the power transferred from the mixer to VGA has to be higher on the band-pass and lower on the stop-band, and with the help of S(1,1), which represents the power reflected, it was achieved lower power on band-pass and near to 0 dB on the stop-band.

Figure 3.22 shows the S(2,1) parameter achieved by the designed filter. In this graph it can be seen that the new filter has an attenuation on its passband around 7 dB, although provides a higher than 100 dB of attenuation on the stopband. With this cascade filter the desired 90 dB of power range can now be achieved.

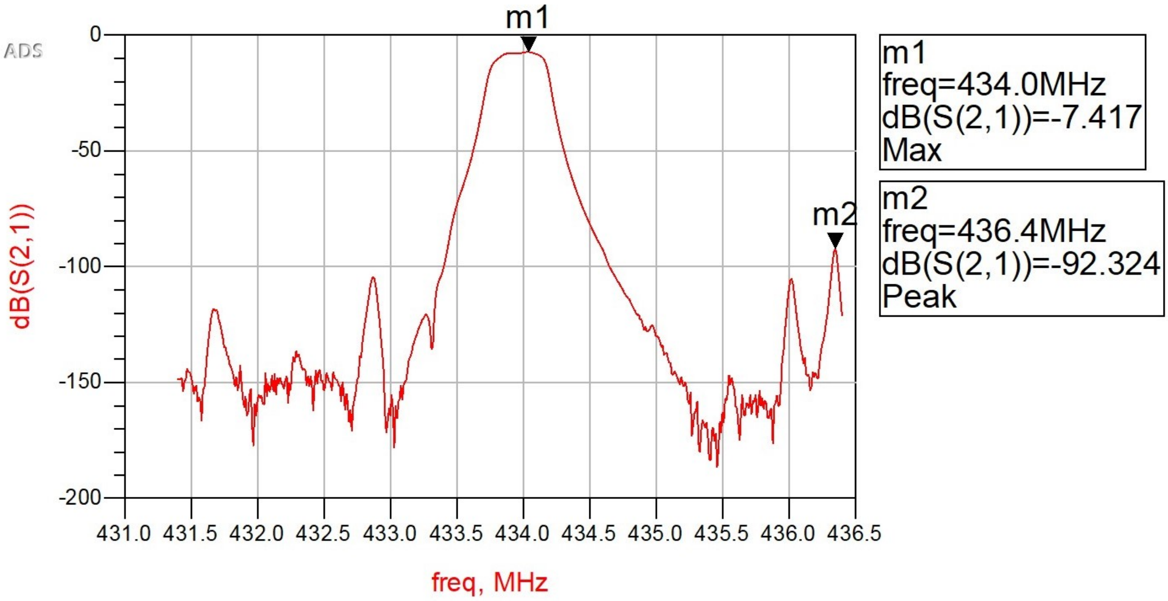


Figure 3.22: S21 parameters of the filter (matching network between mixer and VGA).

3.3.3 VGA to RMS Power Detector

This network matching was created through a smith chart. The process was to produce an optimization with the impedance table provided by the datasheet from the side of the power detector, and the 100 Ω impedance for the VGA output.

Adding this matching to the circuit completes the IF signal path, which allow a complete simulation of the system. In Figure 3.23 is present the simulation results taking into accont the maximum gain in the VGA S-Parameters.

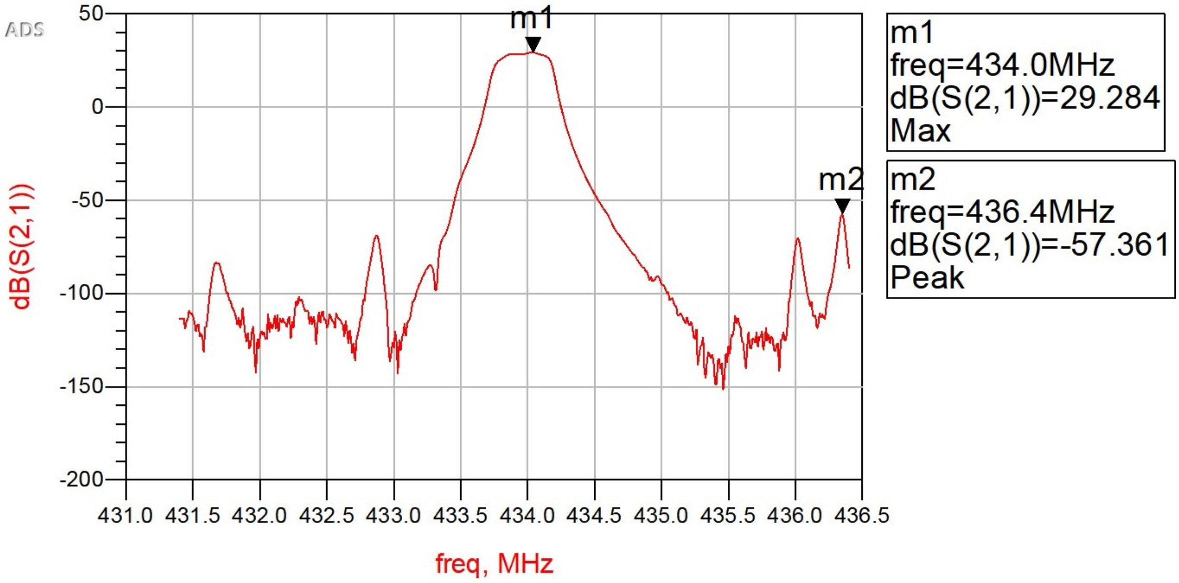


Figure 3.23: S21 parameters system maximum gain.

3.4 Layout

The layout had one of the main constrains of this project, since the PCB will be printed and soldered at the IT, the requisites previously mentioned on the electronic components section are applying here. While most of the layout's issues were taken in consideration during the components' selection, there are others, such as the two substrate layers, that will have a greater impact now.

The layout was built with the ADS software, apart from the devices' layout package, most of them were obtained through the Eagle software. It was added to Eagle the SamacSys PCB Library and the Snapeda library, as they are libraries with a huge number of schematics and layouts from most devices. The devices' layouts were exported from the Eagle, by Gerber files, and then imported into ADS. The few devices that were not found in these libraries were designed manually in ADS, with the package, provided by the datasheet.

In general, the circuit was designed to have the digital part separated from the RF/IF, and the bottom layer of the substrate was used for the ground only. The second point could not be fully accomplished, as some copper lines needed to pass over the bottom. These bottom lines were prioritized to be of DC over AC signal, being most of them lines supplying power for the devices. These features were considered so that the RF/IF path could be as clean as possible and not degrade the quality of the signal.

The lanes that the input signal travels when it is introduced into the system until it gets read by the power detector, were designed to be the straightest and shortest possible. Since the IF is a differential signal, the straight lines make the matching in the two lanes more similar. As for the advantage on the lines being shorter, is that since the frequencies are around 400

MHz, this means that the quarter length of the signal is around 0.18 mm, so the usage of small lines makes the matching rely mostly on just the matching components and the impedance from the lanes can be disregarded. Another benefit is that a small copper length prevents a magnetic field from interfering through cross lines. Nevertheless, after the copper lanes were placed, they were also added to the simulation, making the optimization more accurate and the matching more reliable.

Just like the schematics, the layout was made separately for each sector and after everything was completed, all sectors were connected as seen in Figure 3.24, thus creating a single system, with the supply power and the microcontroller with their communications the last added. The supply power and the communications were added like a puzzle, where the objective was to fit them with the minimum interference possible in the system and avoiding to do holes on the substrate layers, since these are hard to do them by hand.

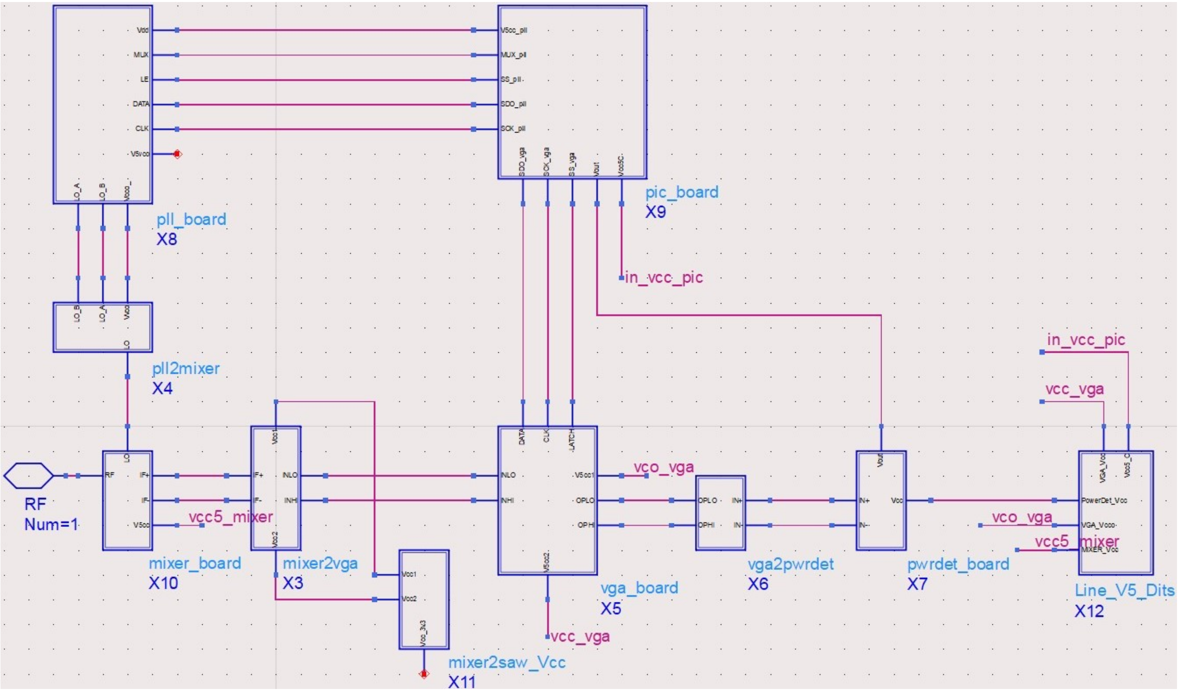


Figure 3.24: System schematic with links create between the devices layouts.

There were other copper lanes worth to be mentioned, such as the communication of SPI, I^{2}C, pipckit3, the USB and the microcontroller's oscillator lines. For the first three, each communication had the lanes with the same length, as this ensures the same delay of the signals and therefore a functional communication.

For the USB communication case, for an overall better system layout, the USB connector could not be connected with straight lanes to the microcontroller due to board constraints. This caused one of the lines to be longer, so for the communications to function properly, the max length difference for a USB speed interface was taken into account [23].

For the external communication, aside the USB connector, a SMA port was needed for the input RF signal, this requires a minimum 2 mm of copper line added to the input of the mixer, so to accomplish the SMA size and allowing it to be soldered. Also, there was a similar need for the I/O pins for the programmer input and for the I^{2}C output, in this case, it was added a hole with diameter of 0.4 mm and distance between them of 2.54 mm.

At last, a few copper test points were added so it would be possible to measure the signal with a spectrum analyzer.

And with that, the layout was ready to be printed (Figure 3.25).

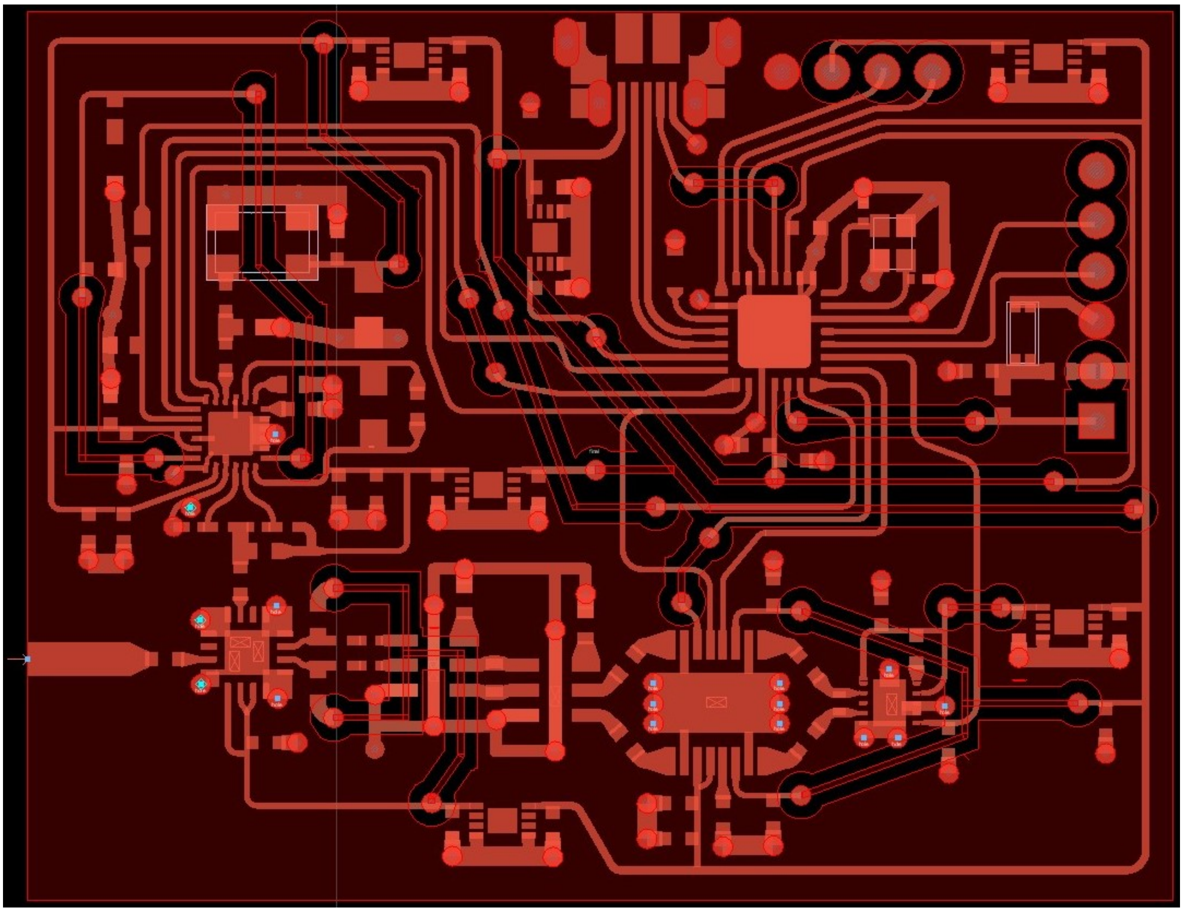


Figure 3.25: System layout.

3.5 System Specification

The system specification is composed by the combination of all elements, with each one contributing to the system specification. The maximum current consumption can be predicted by adding the maximum current of each active component. Table 3.1 represents the theoretical current consumption, the microcontroller maximum current was calculated with the consideration that the SPI is equivalent to three output pins. The results provided for this table also demonstrate that the system can be supplied by a typical laptop USB port.

Table 3.1: System consumption current.

	PLL[13]	Mixer[14]	VGA[16]	Power Det.[15]	Micro.[17]	Total
Max. Current (mA)	47.5	180.4	105	52	155	539.9
Typ. Current (mA)	47.5	163.2	82	41.6	32	366.3

The Noise Figure (NF) of the system can be also calculated through Equation (3.2), and with it was possible to achieve the results shown in Table 3.2. The importance of the noise figure in this thesis system is to understand the degradation of the measured signal. To read a signal with 90 dB on the VGA input, Equation (3.1) can be followed in conjunction with the NF of 12.32 shown in Table 3.2, it is possible to conclude that the input signal of the system needs to have at least 102.32 of SNR.

$$\begin{aligned} NF &= SNR_{i\,,dB} - SNR_{o\,,dB} \Leftrightarrow \\ SNR_{i\,,dB} &= NF + SNR_{o\,,dB} \Leftrightarrow \\ SNR_{o\,,dB} &= SNR_{i\,,dB} - NF. \end{aligned} \tag{3.1}$$

$$\begin{aligned} &NF_{Total} = 10 \cdot log_{10}\Big(n_1 + \sum_{i=2}^{M} \frac{n_i - 1}{\prod_{j=1}^{i-1} g_j}\Big). \\ &n_i = 10^{\frac{NF_i}{10}}, \quad g_i = 10^{\frac{G_i}{10}}. \end{aligned} \tag{3.2}$$

Table 3.2: System gain and noise figure.

	Mixer	Filter	VGA
Gain (dB)	3	-7.4	-11 / 37
Noise Figure (dB)	11.7	7.4	25 / 8.1
Cumulative Noise Figure (dB)	11.7	12.32	29.5 / 15.1

With all the techniques and components discussed in this chapter, this system can accomplish the work's objectives, and is able of measuring a bandwidth larger than the initial 20 MHz. This model can also be used for higher frequencies up to 3.8 GHz by only changing the PLL used and doing the necessary adjustments to the matching networks of the mixer so it can be scaled for current 5G applications if desired.

Chapter 4

Software Design

4.1 Microcontroller Project

In this project the microcontroller has three major responsibilities. It controls the PLL and the VGA devices through SPI protocol, it reads through an ADC the power spectrum measured by the power detector and it transmits the result to the PC through a USB communication.

The microcontroller can process the measurements and then control the predistorter by the I^{2}C protocol, while these features have been architected at the project design, there was no predistorter available to test, so no software was created for this matter, but it can be easily added afterwards for future projects.

Regarding the firmware development, it was created and tested by parts, thereafter the parts were integrated into a functional system with the help of state machine logic.

Initially, the firmware tests were performed in a different microcontroller, since the PIC32MX274F256B needed some extra time to arrive at IT and be built into the final PCB. For that reason, it was used the PIC32MX755F512H illustrated in Figure 4.1, since this microcontroller has similar specifications, and it was ready to use.

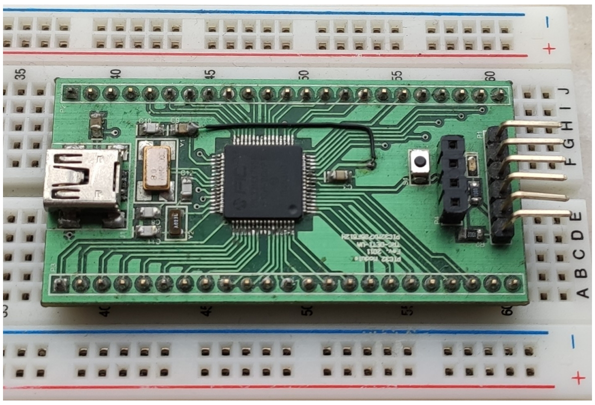


Figure 4.1: PIC32MX755F512H used on the development testing.

After the first test with PIC32MX755F512H, a PCB with PIC32MX274F256B was built alone without the other devices, in order to prove that the microcontroller circuit projected earlier would work as intended.

The Figure 4.2 demonstrate the workbench, where a new test was then designed with the help of a breadboard and Arduino Uno.

The Arduino is a basic single board microcontroller. This microcontroller has a simple interface with a large number of libraries and code available, for its simplicity and as this microcontroller was readily available, it was used to serve as an SPI slave, so it could simulate the PLL and VGA device, with the added advantage of the communication words could be presented in real time on a PC, by printing through the serial port.

With the Arduino and the breadboard, a practical simulation was performed, the PIC32MX274F256B performs a normal measure while, as a substitute, the Arduino receives the SPI communication there were originally for the PLL and VGA, a button is needed to simulate the trigger given by the PLL when the target frequency is Locked, and lastly the potentiometer value is read by the PIC32MX274F256B, thus simulating the resulting value of the power meter.

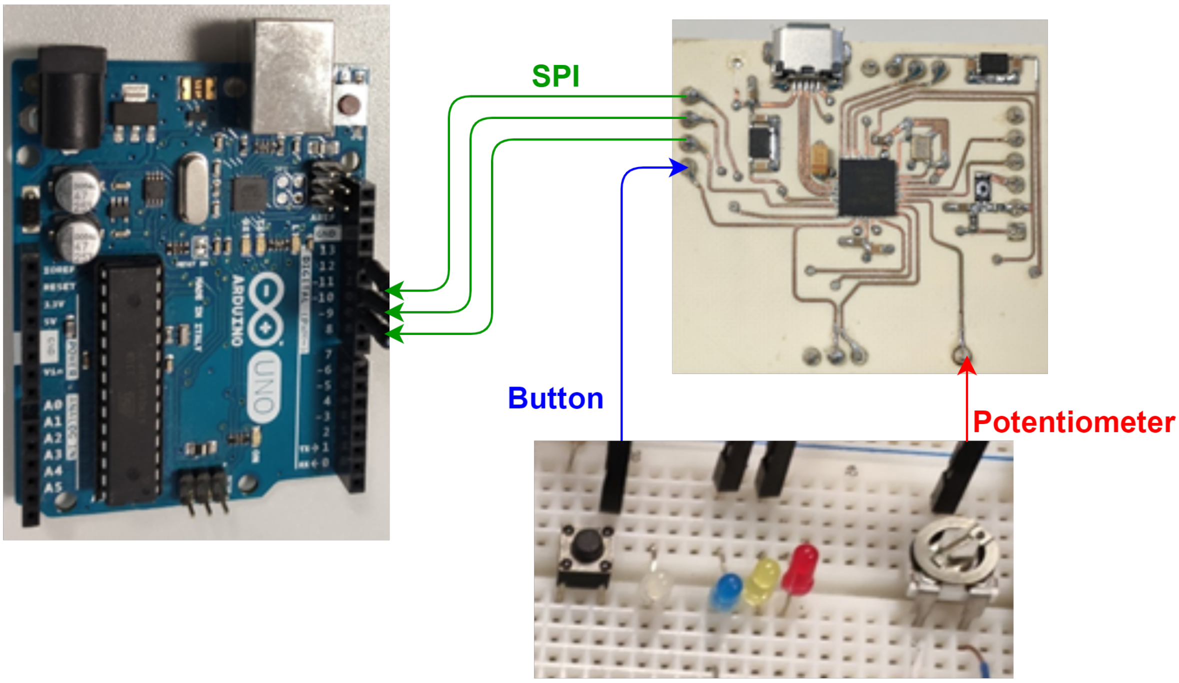


Figure 4.2: Workbench diagram used during the developement testing.

With the above strategies, it was possible to simplify and accelerate the cyclic process of programming, testing, and debugging the code, which lead to desired result.

4.2 Firmware Code

The code was written in C language using the MPLAB software, an IDE provided by Microchip, this software integrates the compiler, and it is also capable of program the microcontroller through the PICKIT3.

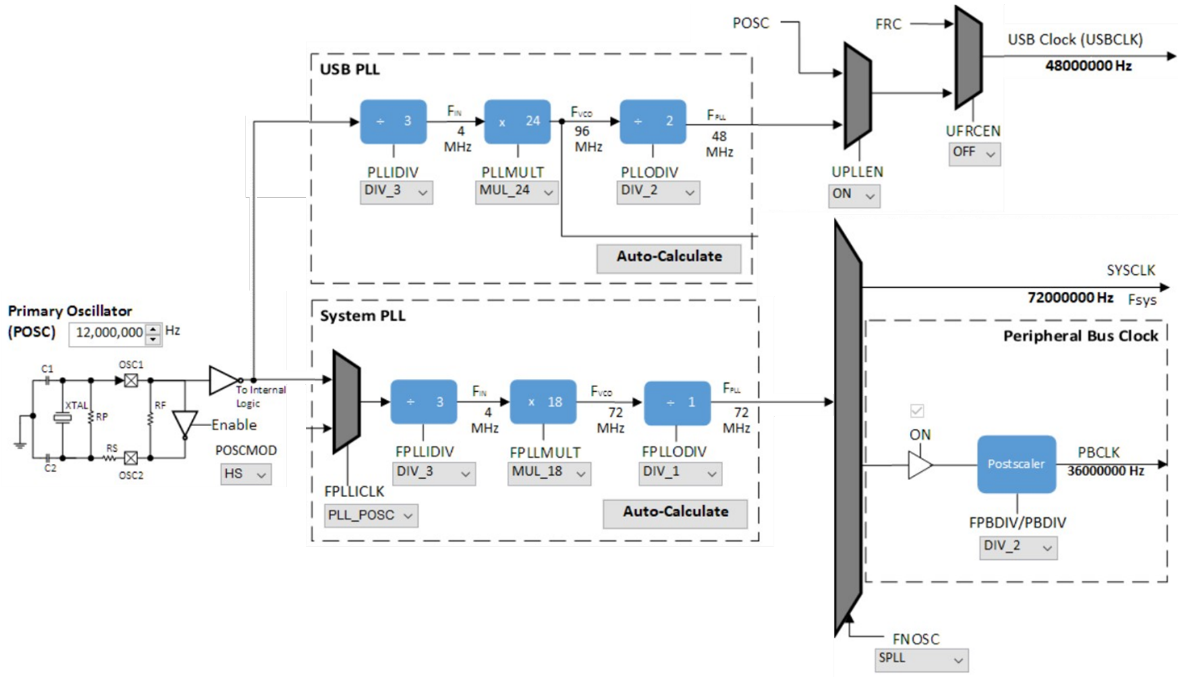
Microchip also provides the download of Harmony for MPLAB, the current Harmony v3 is a fully integrated embedded software development framework, this framework provides working and well documented drivers and complete test project applications, this framework is integrated in MPLAB with an intuitive a graphical interface.

At first, it was used on the PIC32MX755F512H a previous version of the Harmony, the version 2, this version provided a good experience, which gave a great boost when creating the firmware, especially because Harmony provides a full functional Human Interface Device (HID) drives, the class chosen for the communications between the microcontroller and the computer.

After moving to the PIC32MX274F256B, Harmony v2 no longer suited this new board, consequently forcing the move for Harmony v3, but this version presented some setbacks, as most of the previous drivers used no longer worked. For that reason, it was decided to nearly drop Harmony, at the end it was used only to configure the clocks and for its HID drivers.

4.2.1 Harmony

The clocks were configured in a graphic environment, the system clock is present in the Figure 4.3, and it can be seen that the primary oscillator of 12 MHz supplies the internal system PLL, and because the clock is divided by 3 to achieve the 4 MHz since that is the maximum stipulated frequency that FPLLMULT can have as an input, and then it is multiplied by 18 resulting in the system clock of 72 MHz which is the normal maximum supported the Pic without overclocking it. The peripheral bus provides the clock to the SPI, ADC and timers modules, and it uses the system clock divided by 2, resulting in 36 MHz, the maximum frequency the bus can handle. At last, for the USB clock the primary clock signal was processed through the dedicate USB PLL resulting into a frequency of 48 MHz.



Primary Oscillator (POSC) 12,000,000 Hz
POSCMOD HS
PLLIDIV DIV_3
PLLMULT MUL_24
PLLODIV DIV_2
UPLLEN ON
UFRCEN OFF
USB Clock (USBCLK) 48000000 Hz
FPLLICLK PLL_POSC
FPLLIDIV DIV_3
FPLLMULT MUL_18
FPLLODIV DIV_1
SYSCLK 72000000 Hz Fsys
PBCLK 36000000 Hz
FPBDIV/PBDIV DIV_2
FNOSC SPLL

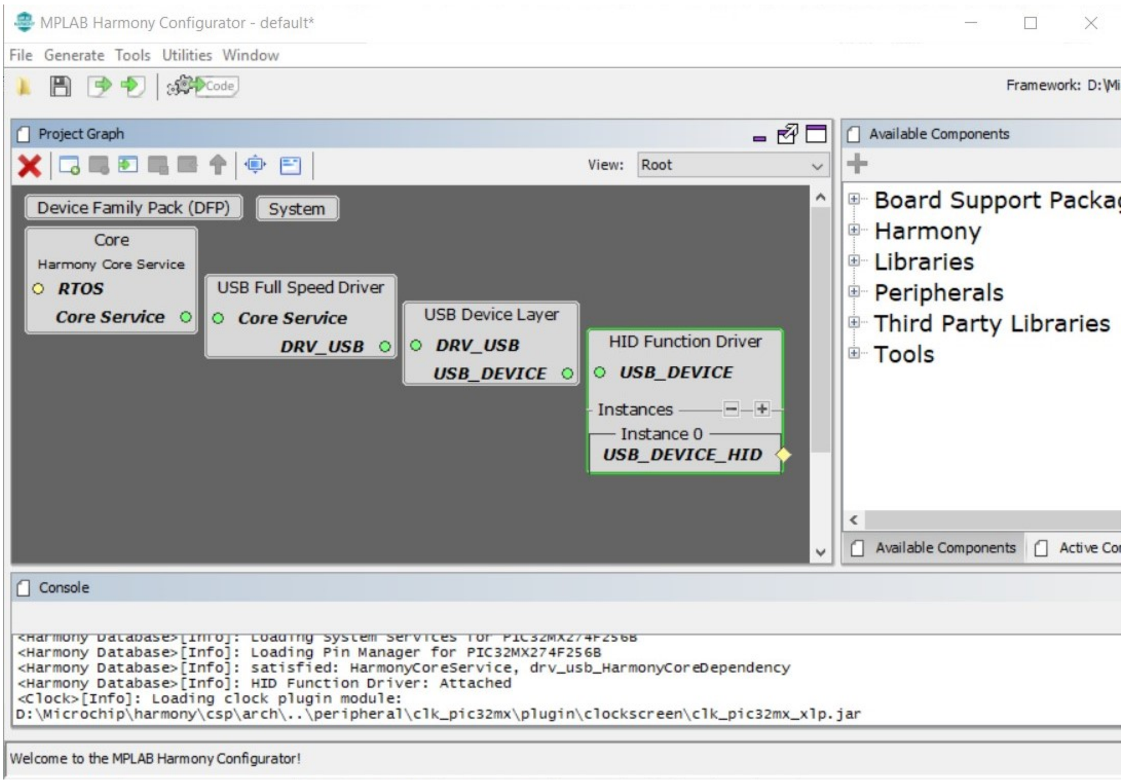
Figure 4.3: Harmony Software, edited system clock configuration.

The SPI code is an adaptation of the SPI peripheral library included in Harmony, to this one, it was added the capability to attach and use a chip select, this is an I/O port from the PIC selected by the user which allows the usage of SPI parallel slaves.

Furthermore, the target devices do not really use a SPI protocol, but a similar communication which does not have the Master In Slave Out bus. For that reason, it was removed from the Harmony SPI code the ability to generate dummy code and transmit data through the MISO bus, being this only able to transmit data from master to slave through the Master Out Slave In bus.

In the HID case, it was followed the microchip forum recommendations. First, the HID driver libraries from Harmony were imported, except the RTOS library, even though the harmony refer this library as a dependence to the HID drives, they can work without it. Then, so that the code would compile without errors, the drivers were adapted with the help of the old drivers from Harmony v2.

The above configurations are the final Harmony configuration that were used and they can be seen in the following Figure 4.4.



```
<Harmony Database>[Info]: Loading System Services for PIC32MX274F256B
<Harmony Database>[Info]: Loading Pin Manager for PIC32MX274F256B
<Harmony Database>[Info]: satisfied: HarmonyCoreService, drv_usb_HarmonyCoreDependency
<Harmony Database>[Info]: HID Function Driver: Attached
<Clock>[Info]: Loading clock plugin module:
D:\Microchip\harmony\csp\arch\..\peripheral\clk_pic32mx\plugin\clockscreen\clk_pic32mx_xlp.jar
```

Welcome to the MPLAB Harmony Configurator!

Figure 4.4: Harmony Software, system libraries dependency.

The remaining modules were programmed without the help of the Harmony drivers.

4.2.2 System State Machine

After all independent modules were working and tested, a state machine was created to merge them into a functional firmware.

The first state machine created was the “One Measure”, whose purpose was to measure the power of one frequency.

4.2.3 One Measure State Machine

In Figure 4.5, it can be observed a state machine diagram of “One Measure”. After the initial configuration of the peripherals, the intended starting gain is set on the VGA, with the communication microcontroller-VGA being through the channel 2 of SPI communication. The next step is to set the PLL configuration, this configuration includes the planned frequency, and it is made by sending three packages sequentially through the channel 1 of SPI. When this frequency gets locked on the PLL, the PLL triggers a signal at its output Mux pin, this is caught by the microcontroller through an external interrupt, which will prompt the next stage of the state machine to start.

On this next state, the ADC is read, and a decision is made based on the measured value. If the power read is above the minimum threshold limit, the current data is saved. But if the power read is below the threshold limit, a new package is sent to the VGA to increase its gain. In this new state two choices are available. If the gain was successfully increased, the state machine will return to the previous "ADC Read Power", but if the VGA's gain overtakes the established maximum, then the data is saved. At last, when the system finishes saving the results on the "Save Data" state or the program is cancelled, the VGA's gain is set to the idle gain.

This machine checks the HID communication at the beginning of each cycle, which allows the measure to be cancelled at any moment.

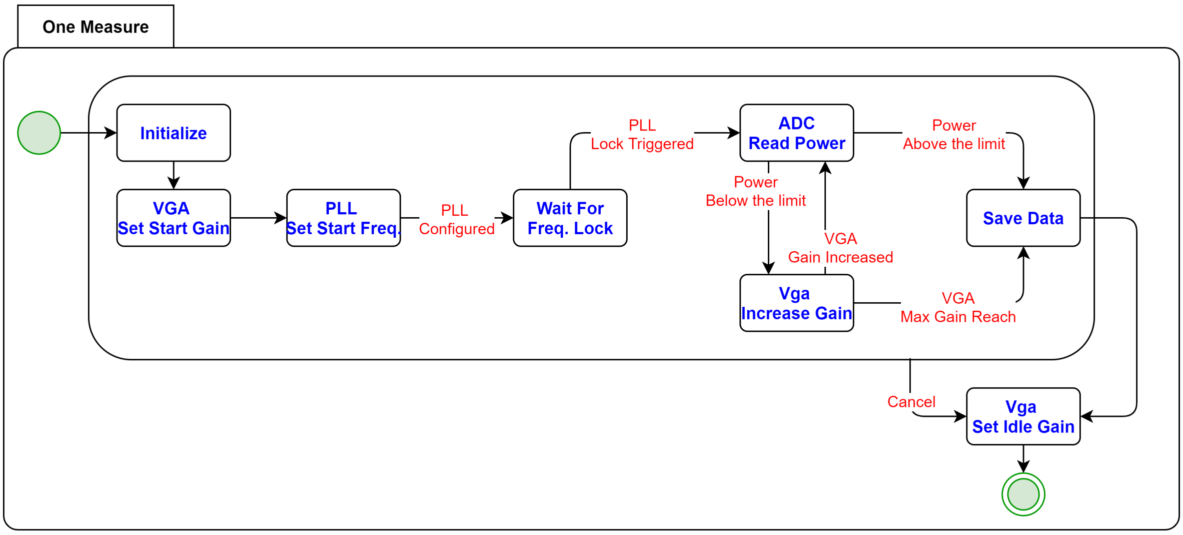


Figure 4.5: State diagram of the One Measure.

4.2.4 Normal Measure State Machine

After the "One Measure" state machine was tested, a more complex state machine was created. This "Normal Measure" state machine incorporates the previous "One Measure" state machine and adds the functionality of covering a range of frequencies, so with this state machine it is possible to obtain the intended measure of the signal spectrum.

In Figure 4.6, it is possible to see a state machine diagram of the "Normal Measure". As mentioned above, this machine runs exactly the same as the previous one until it arrives at the "Save Data" state and after that point, the state machine is continuously performing a new frequency measurement. First, the VGA's gain is reset back to the starting value and then, a new frequency is set. Since this new frequency is only an increase from the previous one, there is no need for additional configuration, therefore only one package is needed to be sent to the PLL. After this step, the cycle returns to the "Wait For Freq. Lock" state.

This cycle is broken in the “PLL increase freq.” once the system overtakes the established maximum frequency. When this occurs, the measurements are completed, and the state machine finishes by moving to the “VGA Set Idle Gain” state similarly to the “One State Measure”. At any time, the state machine can be cancelled through the HID communication at the beginning of every cycle too.

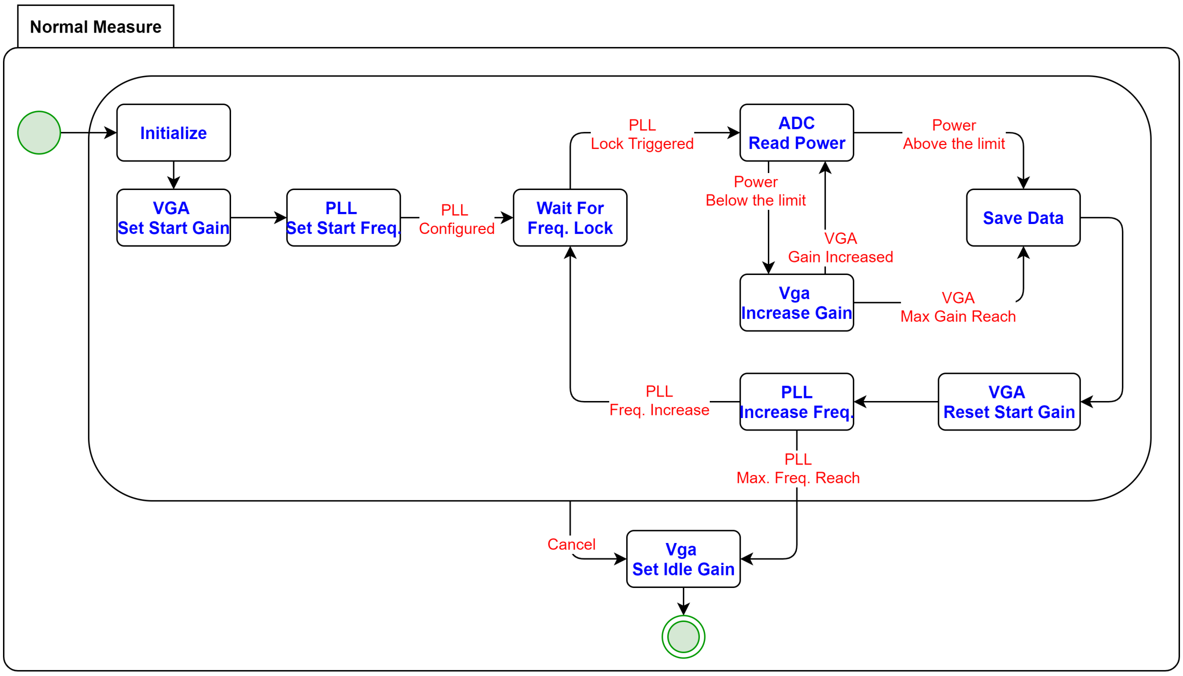


Figure 4.6: State diagram of the Normal Measure.

4.2.5 Final State Machine

Finally, it was created a “Super State machine”, that merges and allows access to the other state machines.

This super state, represented in Figure 4.7, starts with the initialize state where it makes the setup needed for the proper working conditions of every peripheral. After that, it goes to the “Idle” state, this state uses the HID communication. If a measure is requested by the HID module, the next state will correspond to request state.

Both the “One Measure State” and the “Normal State” return to “Idle” State when finished. Furthermore, at any moment these states can be cancelled, returning to the “Idle” state or they can move to each other by interrupting the current state, thus starting a new state.

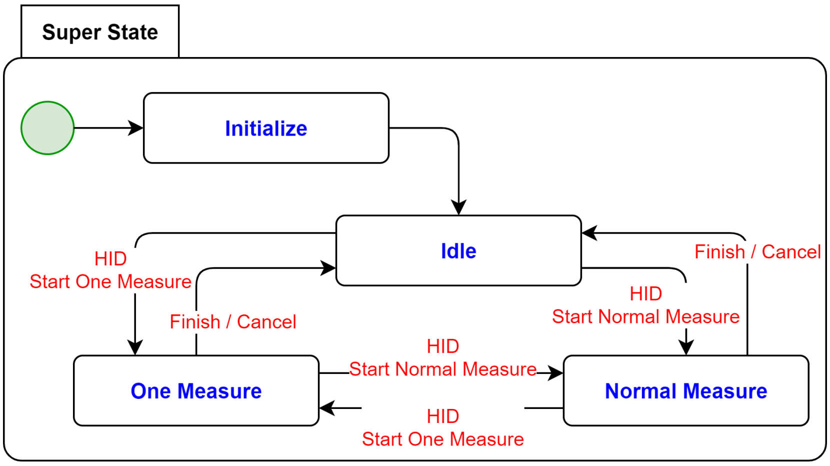


Figure 4.7: State diagram of the application firmware.

4.3 Application Interface

To control the system, it was created a computer application with a graphic interface, this provides a better platform to control and interpret the output of the PCB system. The application was made in MATLAB. MATLAB has a Design App where it is possible to easily create a graphic application, furthermore the programming language used, is known to be excellent when manipulating data and creating the corresponding graphics.

The design application programming is made through blocks, where it can be selected the graphic design and function of each object and after they can be customized with extra lines of code. To have a more organized program, various classes were created in separate files. Each class focus on a different subject and its methods can be called by the main function whenever needed.

In Figure 4.8 it is represented a use case diagram of the application created in MATLAB. When opened, the user has three main areas available. It has the “Menu” area containing “File” options which allows to both save and load any system configuration or measured results. “Menu” also includes a “Connection” option where the user may open or close a HID connection to the microcontroller system.

The second area is the “Central Panel” and has five different tabs. This area is the most important of them, as it controls both the configuration sends to the system and shows the results obtained from any experiment.

The last area is called “Instant Status”. At its left, it shows the last five log messages generate by the application and on the right, there is a switch button that shows the current status of an experiment and it can start a new test or cancel an existing one. There is also a box that allows the user to set the period of when the application requests the experiment results from the microcontroller’s system.

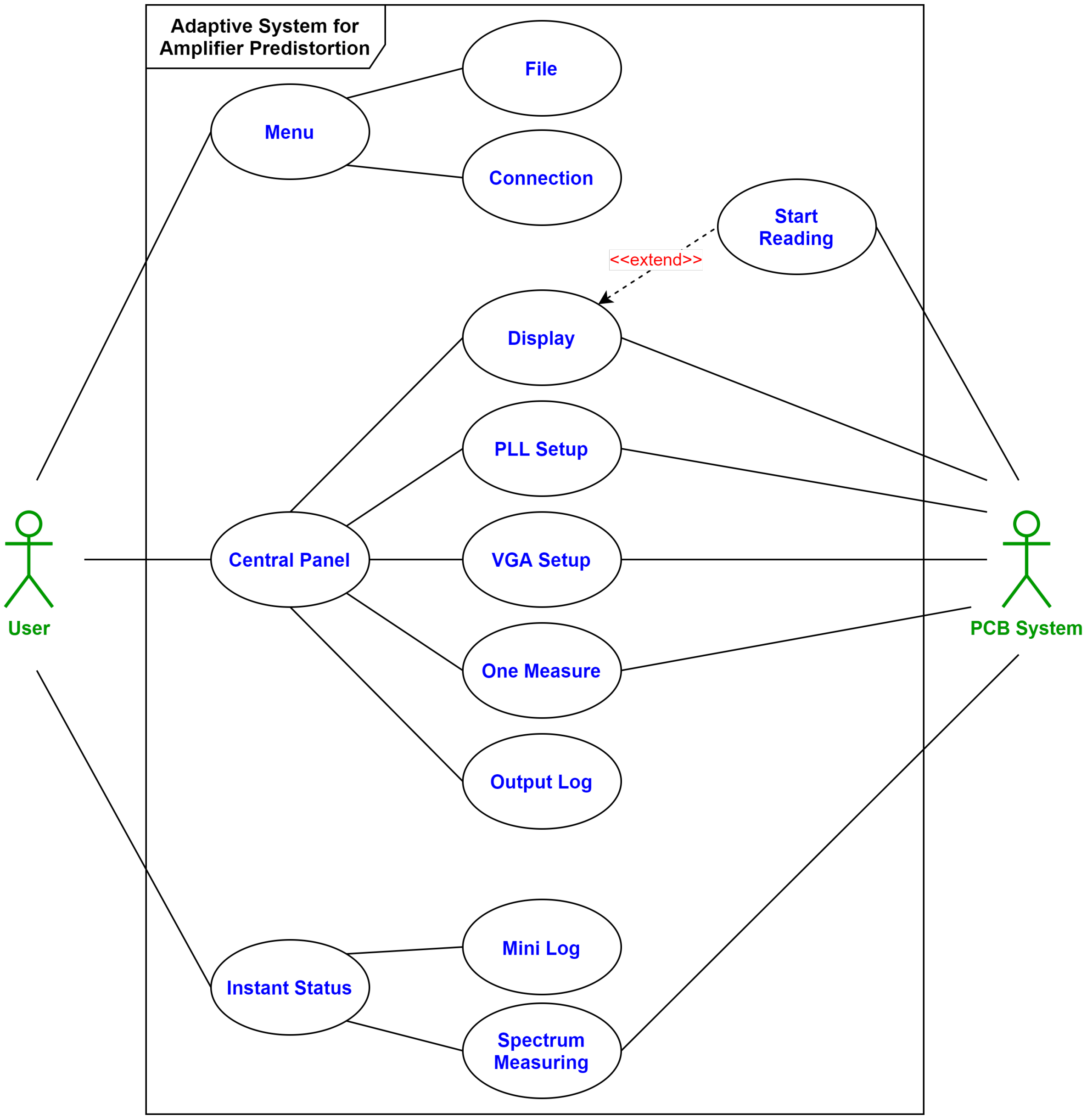
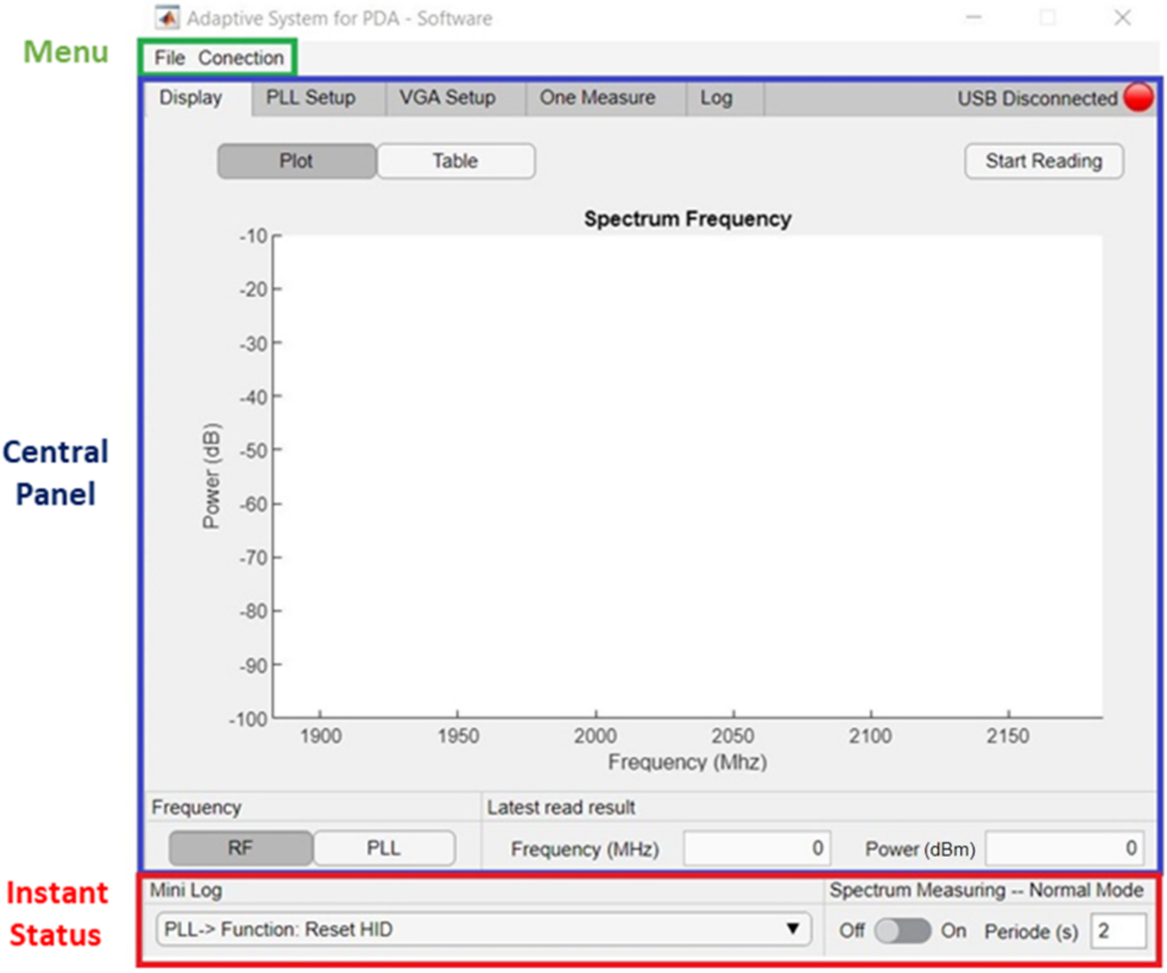


Figure 4.8: Use case diagram of the MATLAB application.



Frequency (MHz) 0

Power (dBm) 0

PLL-> Function: Reset HID ▼

Periode (s) 2

Figure 4.9: Display tab of the MATLAB aplication.

4.3.1 Central Panel

As mentioned before, the "Central Panel" has five different tabs, which are described below:

- Display:

  The first tab of the "Central Panel" is the display, it can be seen in Figure 4.9. In this panel it is presented the result received from a normal measure, the results can be shown with a graphic or as table values, with the current display being select through a button on the top-left of the panel. On the top-right, a button to start and stop a normal measure is available and this can be an alternative to the switch present on the "Instant Status" area.

  At last, at the bottom-left it is possible to select the type of frequency. There are two different frequency types, one that can be selected with the RF button, and the frequency generated by the PLL, which is controlled by the PLL button. These frequencies only differ 433.9 MHz between them since that is the frequency of the SAW filter. At the bottom-left is a box where the last value received is shown.

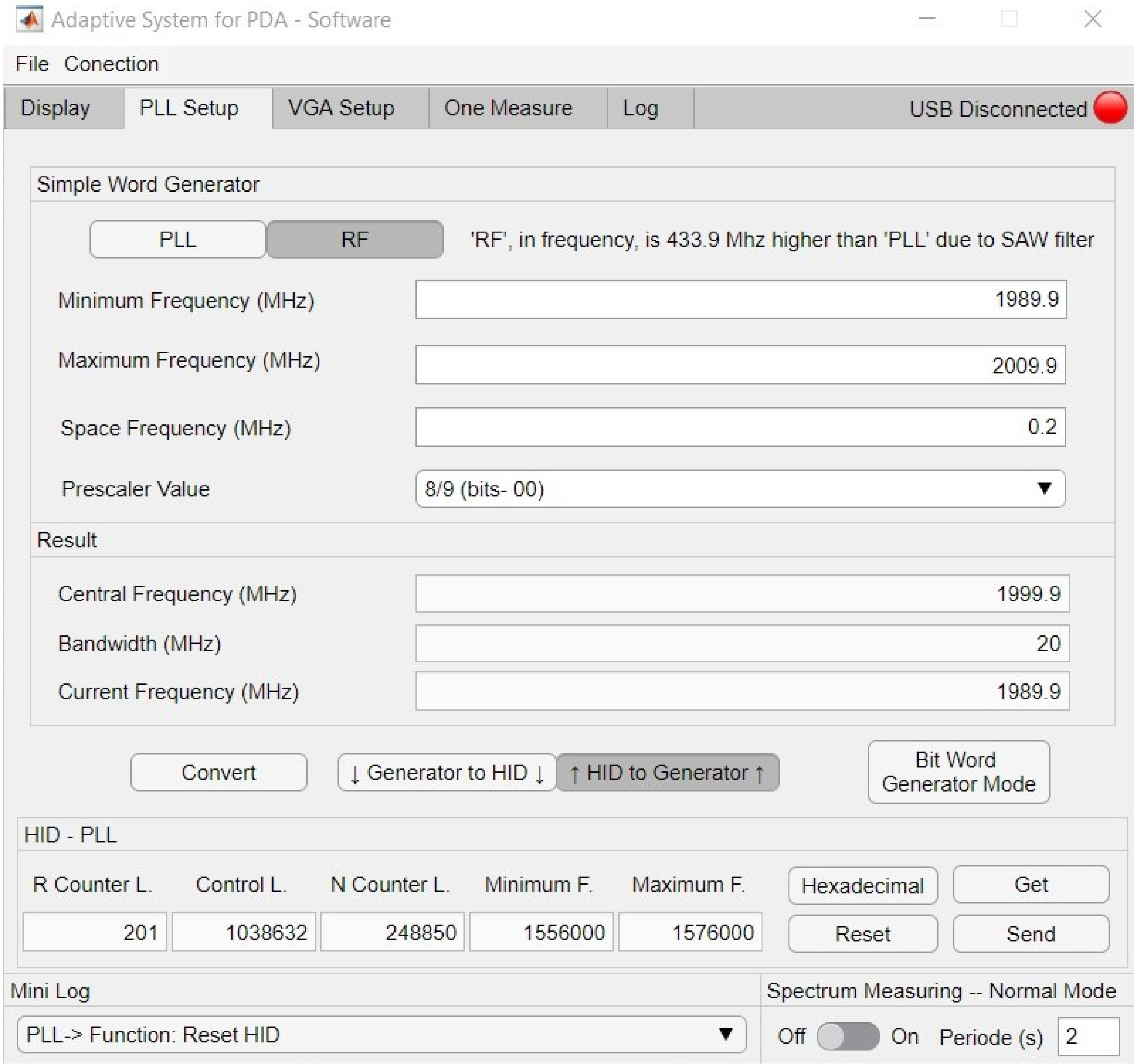
- PLL Setup:

  The “PLL Setup” is the second tab, which can be visualized on Figure 4.10. On the top part of this tab, it is possible to configure the PLL through two possible forms. The first is a simple word generator, where is used to update only the relevant data, while maintaining the other configurations unchanged. In this generator, the minimum and maximum frequencies can be set, as well as the space frequency of a normal measure. It is also possible to change the Prescaler value, while it usually does not need to be changed, depending on the frequency values above set, it may be necessary increase this value. After setting the mentioned values, the program generates the necessary data to send to the PLL and shows it in a simple form in the “Result” panel, where the user can confirm the values.

  The second form is called “Bit word generator” which allows the user to directly change bit by bit the data to be sent, and it is also displayed some information associated with each bit explaining their function.

  At the middle of this tab, it is possible to choose the generator, and with through the “Convert” convert button, the configuration set on the selected generator can be translate into a data set below so that it is ready to be sent.

  At the end of “PLL Setup” tab, the data that can be sent or received through the HID is presented, this action can be performed through the buttons available on the right. The “Hexadecimal” button only changes the shown data format, which can allow an easier view. The “Reset” button replaces the HID data to the initialize configuration. The “Get” button receives the current setup from the microcontroller and updates the shown HID values. Finally, the “Send” button sets the current configuration displayed on the microcontroller.



Simple Word Generator

'RF', in frequency, is 433.9 Mhz higher than 'PLL' due to SAW filter

Minimum Frequency (MHz): 1989.9

Maximum Frequency (MHz): 2009.9

Space Frequency (MHz): 0.2

Prescaler Value: 8/9 (bits- 00)

Result

Central Frequency (MHz): 1999.9

Bandwidth (MHz): 20

Current Frequency (MHz): 1989.9

HID - PLL

R Counter L.	Control L.	N Counter L.	Minimum F.	Maximum F.
201	1038632	248850	1556000	1576000

Mini Log: PLL-> Function: Reset HID

Spectrum Measuring -- Normal Mode

Periode (s): 2

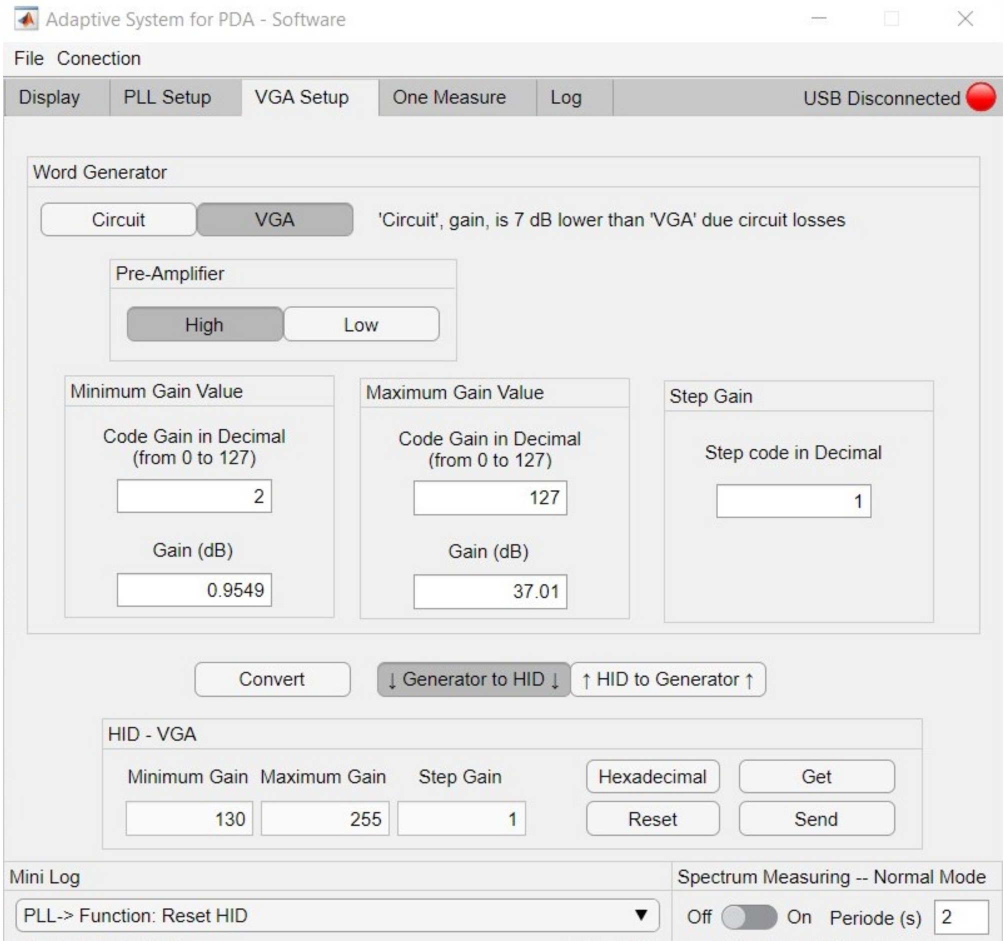
Figure 4.10: PLL Setup tab of the MATLAB application.

- VGA Setup:

  As the name implies, in the “VGA Setup” tab represented on Figure 4.11, the VGA can be configured.

  This tab is similar to the previous “PLL Setup” tab, but with only one generator. In this word generator, the start gain, stop gain and the step gain of the VGA can be configured. There are two options to obtain the values for this generator: the VGA gain code (how the microcontroller defines the gain, called “Code Gain in Decimal”) or by directly inputting the VGA gain in dB (called “Gain”). Other option in this tab include the “Circuit/VGA” button which changes the dB gain by 7, used to check the actual system/VGA gain, depending on the option selected.

  The “Pre-Amplifier” region with the options “High/Low” is useful to change the working mode of the VGA, whether on high range mode or low range mode. The “High” option was always selected so the system could enjoy a higher range of power available. The lower parts of this tab work similar to the previous “PLL Setup” tab.



Adaptive System for PDA - Software

USB Disconnected

Word Generator

'Circuit', gain, is 7 dB lower than 'VGA' due circuit losses

Pre-Amplifier: High

Minimum Gain Value
Code Gain in Decimal (from 0 to 127): 2
Gain (dB): 0.9549

Maximum Gain Value
Code Gain in Decimal (from 0 to 127): 127
Gain (dB): 37.01

Step Gain
Step code in Decimal: 1

HID - VGA
Minimum Gain: 130
Maximum Gain: 255
Step Gain: 1

Mini Log: PLL-> Function: Reset HID

Spectrum Measuring -- Normal Mode
Off / On
Periode (s): 2

Figure 4.11: VGA Setup tab of the MATLAB application.

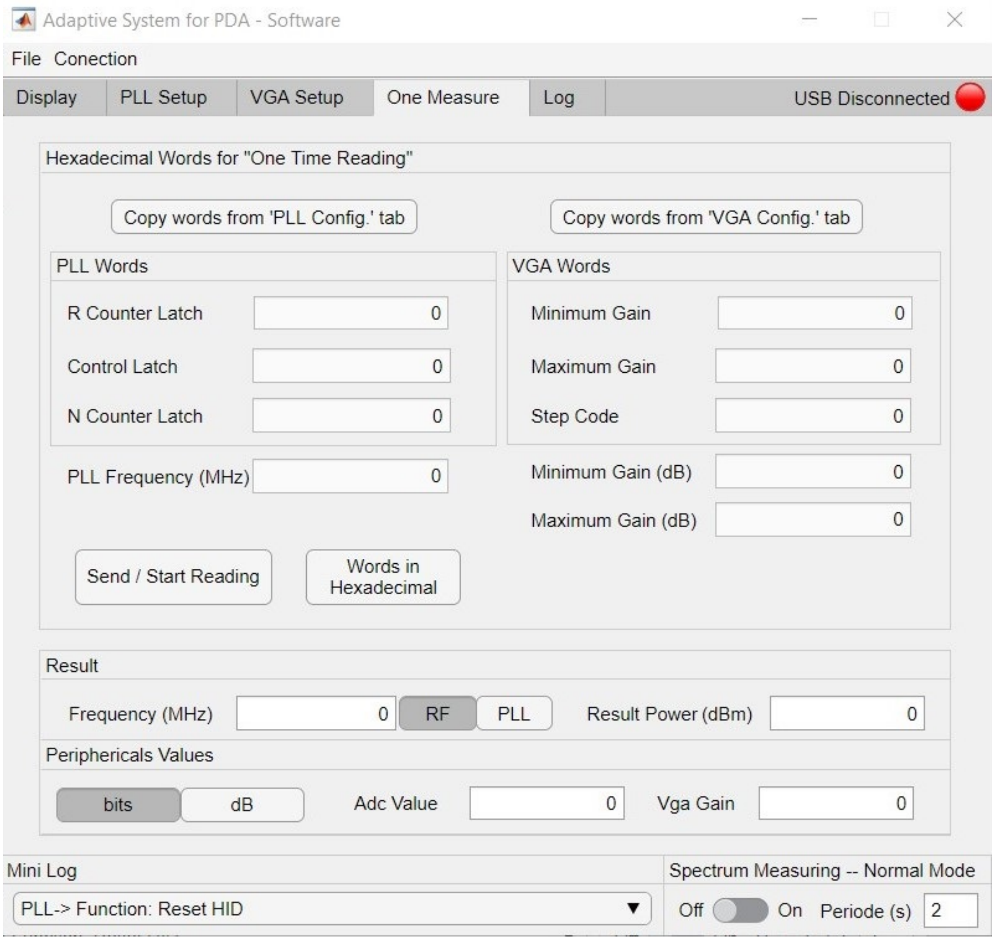
- One Measure:

  The “One Measure” tab is represented in Figure 4.12.

  The top panel present the configuration data, this that can be copied from both configurations tabs, the “PLL Setup” and the “VGA Setup” by pressing the corresponding buttons “Copy words from ‘PLL Config.’ Tab” and “Copy words from ‘VGA Config.’ Tab”. The only exception is the PLL data, where it does not need the maximum and minimum frequencies, the frequency copied will be the minimum, since this is the frequency set in the “N Counter Latch” by the program.

  After that, the “Send / Start Reading” button can be pressed to perform the one measure. When this measure finishes, the results are presented at the bottom panel.

  In this “Result” panel, it shows the total power measured and the components used to calculate it, which is the last gain set on the VGA and the voltage value given by RMS power detector read by the ADC. With these values and by subtracting the system losses, the resulting total power shown is obtained.



R Counter Latch 0

Control Latch 0

N Counter Latch 0

PLL Frequency (MHz) 0

Minimum Gain 0

Maximum Gain 0

Step Code 0

Minimum Gain (dB) 0

Maximum Gain (dB) 0

Frequency (MHz) 0

Result Power (dBm) 0

Adc Value 0

Vga Gain 0

PLL-> Function: Reset HID

Periode (s) 2

Figure 4.12: One Measure tab of the MATLAB application.

- Log:

  Finally, the last tab is the “Log”, in this tab all the application logs are saved and they can be seen, if the log is larger than the window, then a scroll appears to allow the user to see all the log.

4.4 Communication

As mentioned before, two different types of communication were used. First, there is the SPI, which is the interface that the microcontroller uses to control both the PLL and VGA devices. The second type is the USB, the interface that regulates the communications between the microcontroller and the PC.

4.4.1 SPI

The SPI is a full-duplex interface widely used on communications between a microcontroller and peripherals Integrated Circuits (ICs). This interface has the advantage of using fewer wirings when compared with other options such as the parallel communication, and it can handle greater speeds than most other synchronous options, like the I^2C. Further, and more important, the receiving hardware can be a simple shift register, which is the case with the PLL and VGA used.

As shown in Figure 4.13, a common SPI interface uses four wires, the first pin “CS” is a chip select, this pin allows the master to choose which device it will communicate with, this allows the use of the same SPI interface with multiple slaves. The second pin is called “SCLK”, it delivers to the slave the clock generated by the master, so that both devices are synchronous with each other. The last two wires connect the data transfer pins, there is the “MOSI” to “SDI” connection which governs the transmission from the master to the slave, and then there is the “SDO” to “MISO” connection where the slave can send information to the master.

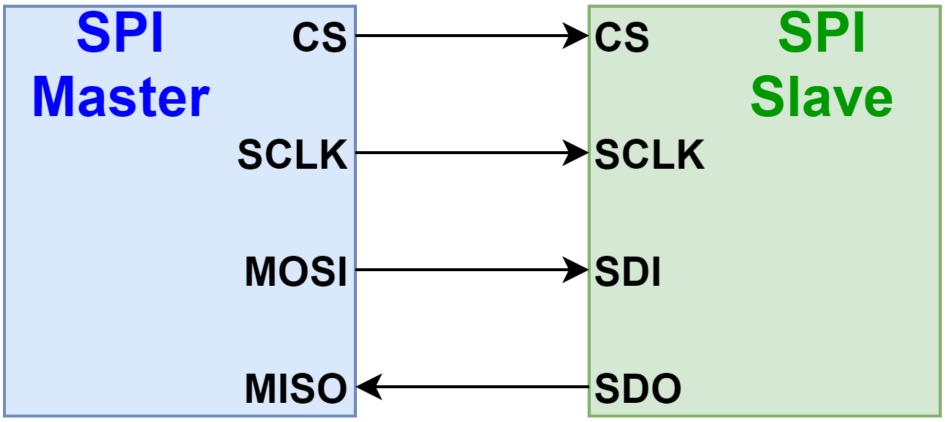


Figure 4.13: Common SPI bus conection from master to a single slave.

The SPI used in this project has some differences, since the slaves are devices that use a single shift register to communicate and there is no need for the microcontroller to receive information from them, the only wires needed are the CS-CS, SCLK-SCLK and the MOSI-SDI. An example of a standard communication using shift registers can be found below on Figures 4.14 and 4.15. The shift register are always active, so they receive new data whenever the microcontroller sends it.

Figure 4.14 shows the status of an 8-bit shift register during the reception of a byte of information, when the "SCLK" signal rises, the shift register stores the current bit read on the "MOSI" signal and shifts the other bits to the left. Figure 4.15 demonstrates the completed transmission of a byte, the data is transferred to the internal Latch of the device when the "CS" bit rises from 0 to 1.

An advantage of this approach is that when sending data, the shift register discards the oldest bits, which is very important for this project because the shift register present on the PLL is a 24-bit shift register and the SPI module of the microcontroller could only send batches of 8, 16 or 32 bits. This way 32 bits could be sent and the 8 leftmost bits would be ignored by the shift register.

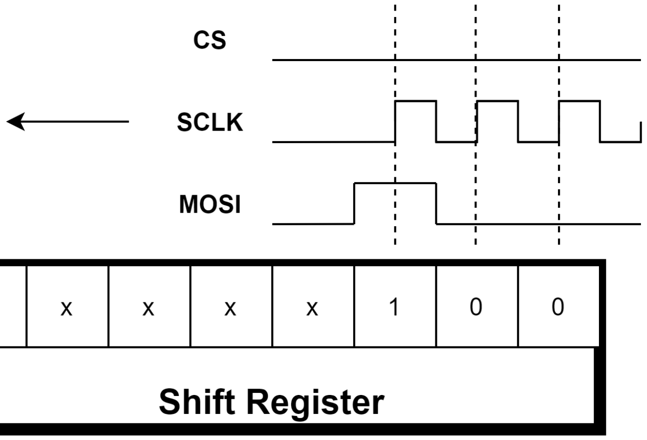


Figure 4.14: Running expample of a shift Register reciving a SPI word.

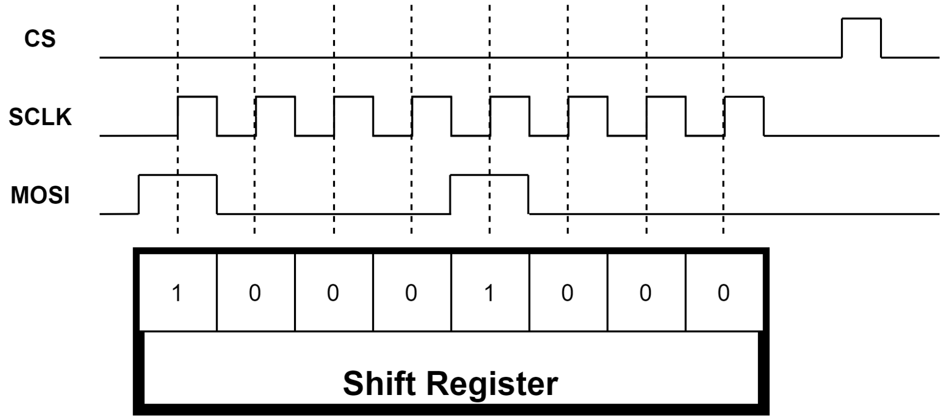


Figure 4.15: Complete SPI byte transmition to a shift Register.

- PLL Data:

  To control the PLL there are three latches that have to be filled with 24 bits. They are R Counter Latch, Control Latch, and N Counter Latch, with each having a different role to fulfil. The R counter Latch main task is to control the frequency step when doing a sweep and can also decide the PLL's precision when locking into a new frequency. The Control Latch is responsible for deciding the power usage and output, it enables the digital lock frequency detect feature and configures the Prescaler value. The N Counter stores information necessary to produce the intended frequency.

  As Figure 4.16 shows, the PLL requires three packets of data and each packet will be transferred to the corresponding Latch, to ensure proper configuration of this system. Once the initial configuration is done, if there is a need to make slight changes to working frequency, only the N counter latch needs to be updated, so just one new packet is required to be sent from the microcontroller.

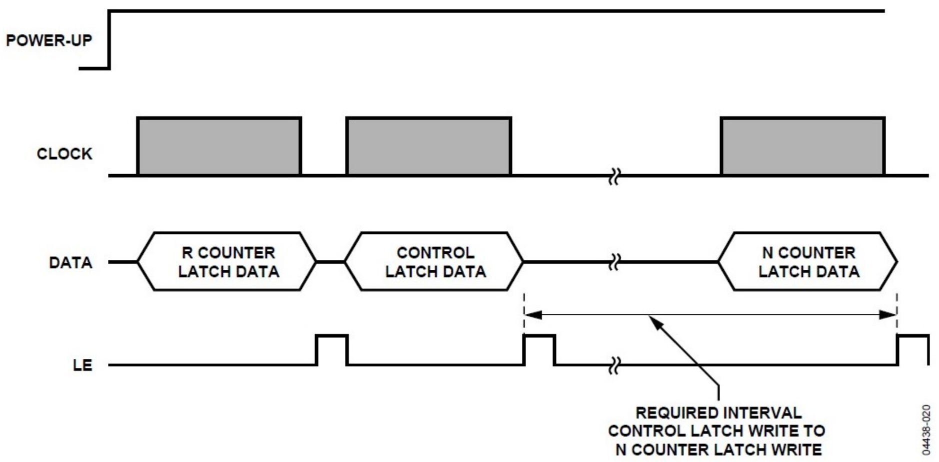


Figure 4.16: ADF4360-4 Power-Up Timing [13]

- VGA Data:

  The VGA only utilizes a single 8-bit packet to work. The leftmost bit chooses the preamplifier that VGA will use. There are two preamplifiers available so that the VGA can accommodate a wide range of input amplitude, because they play an important role when deciding the power step used. The remaining bits decide the power gain of this device.

4.4.2 USB

The communication was made through one open-source library, called matlab-hidapi. This HID protocol uses an array of 64 bytes to communicate, with it, the microcontroller can deliver the instructions to control the PCB system, and also to receive the data provided by the experiments.

In this project it was used 13 different types of data packets, and they can be distinguished by the first byte in each packet.

The types of pakets delivered from the Matlab application to the microcontroller are:

1. Send PLL's setup
2. Get PLL's setup
3. Send VGA' s setup
4. Get VGA' s setup
5. Start a normal measure
6. Request normal measure's results

7. Start a one measure

8. Request one measure's result

9. Stop current measuring

   The types from the microcontroller to the Matlab application are:

10. Send PLL's setup

11. Send VGA' s setup

12. Send normal measure's results

13. Send one measure's result

    Most of the communications are made by simply requesting and sending back a response, except for the Normal Measure. This is made with a set of packets, such as the ones represented in Figure 4.17. The normal measure requires an extra PLL setup and VGA setup and these should be set before starting a normal measure. After starting a frequency sweep with normal measure, the MATLAB application will keep requesting the results and the microcontroller system returns the information gathered so far, until it finishes the current task.

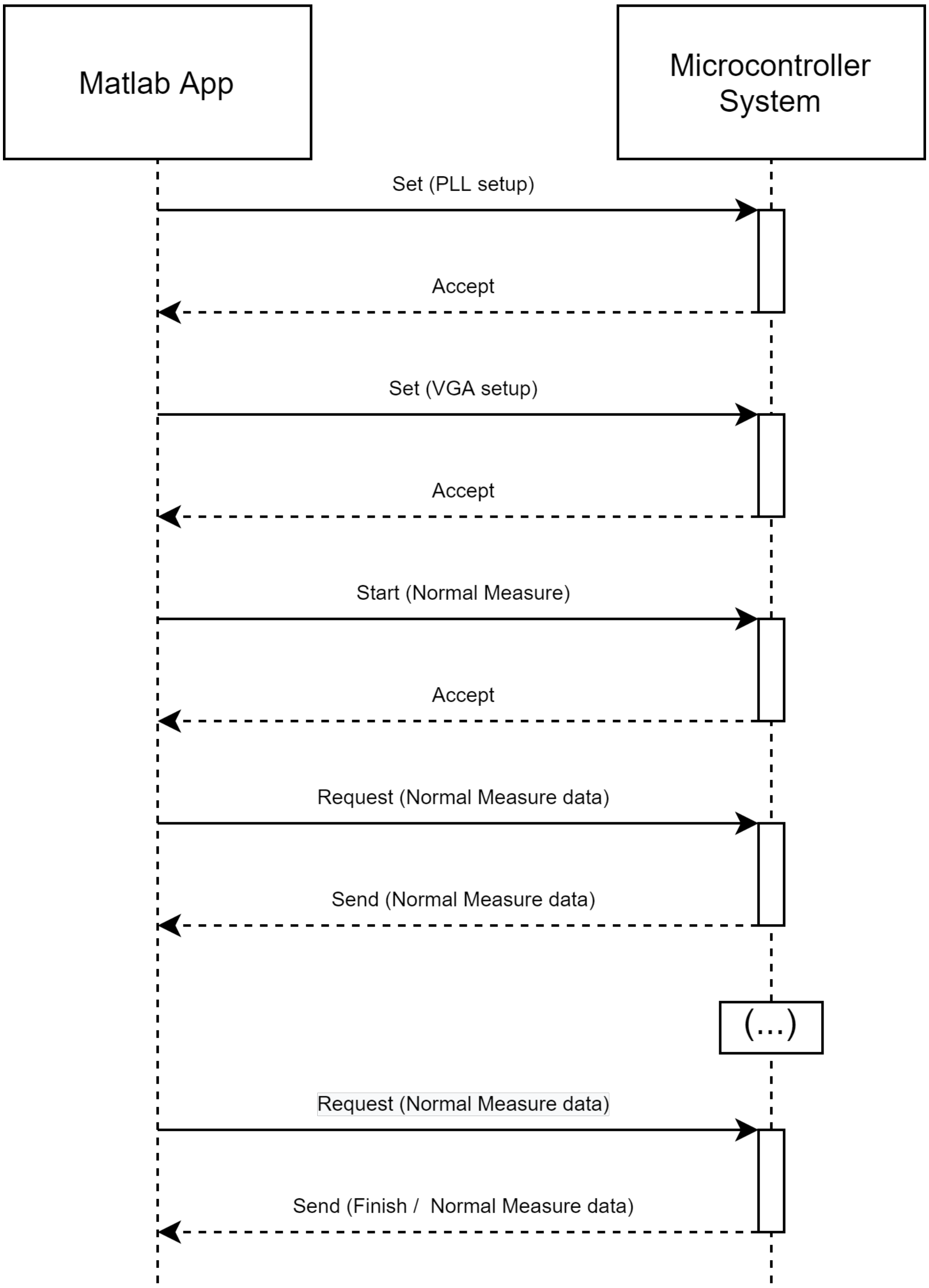

Matlab App
Microcontroller System
Set (PLL setup)
Accept
Set (VGA setup)
Accept
Start (Normal Measure)
Accept
Request (Normal Measure data)
Send (Normal Measure data)
(...)
Request (Normal Measure data)
Send (Finish / Normal Measure data)

Figure 4.17: Interaction diagram to proceed a Normal Measure.

Chapter 5

Measurement Results

In this chapter, it is presented the test and results of the two different measures, first the One Measure, and second the Normal Measure.

These tests were performed by using the MATLAB application to control the system board and show the results obtained from it. Moreover, it was also used a spectrum analyzer to obtain trustworthy measurements to further compare with the results obtained in the MATLAB test.

While the created MATLAB application has methods to save the received data, for the data obtained from the spectrum analyzer it was used an open-source software provided on Rohde & Schwarz's site called FSH4View. This application allows the saving of pictures from the spectrum analyzer as well as file data that can be imported and read on MATLAB.

There is also a demonstration of the signal's spectrum from PLL and SAW filter output, this confirms that both devices are working as expected.

5.1 Inicial Tests

The first test performed was the One Measure since this is simpler than the Normal measure and was used more often during the firmware development.

A representative diagram of the tools used on the first test can be seen in Figure 5.1 For this test, it was used a Vector Signal Generator (VSG) that provides the RF input signal, the thesis PCB system and a computer running the developed MATLAB software.

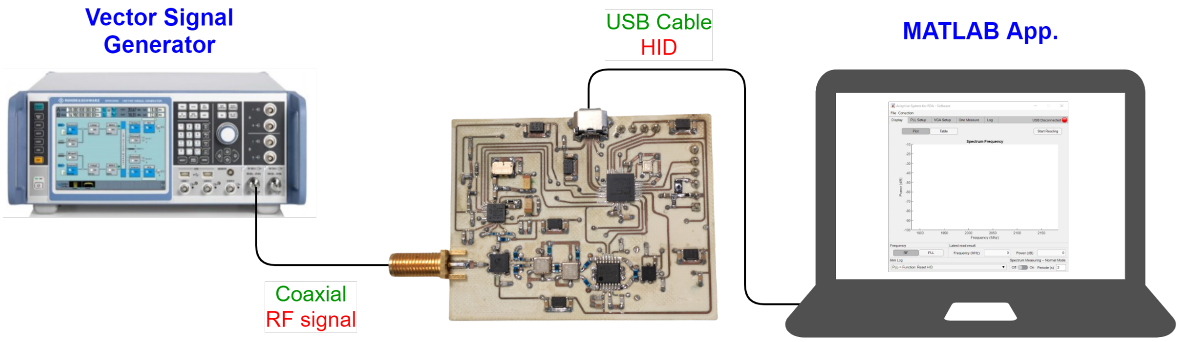
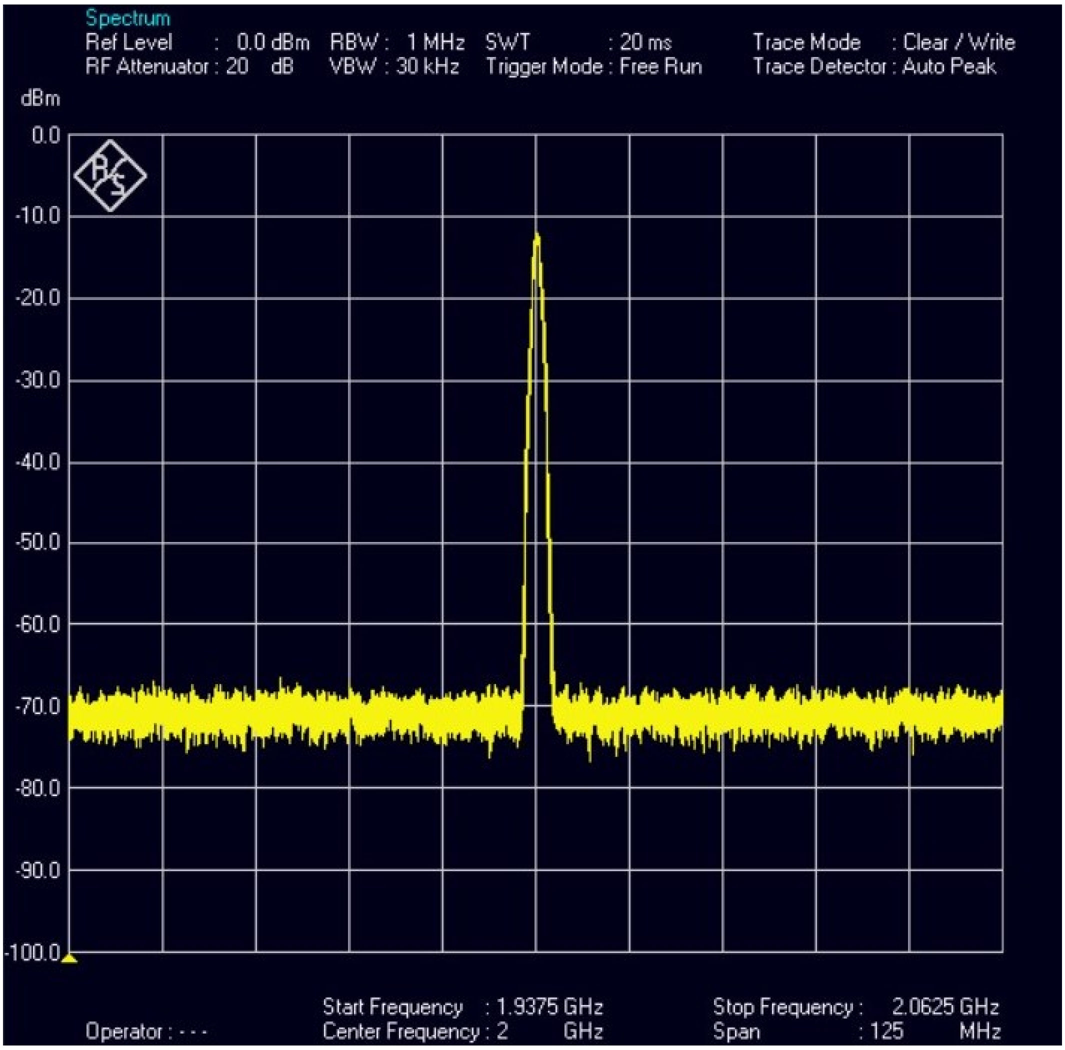


Figure 5.1: Workbench diagram.

In the initial test, the VSG was configured to generate a sine wave at 2 GHz with a Peak Envelope Power (PEP) power of -10dBm, which is represented in Figure 5.2.



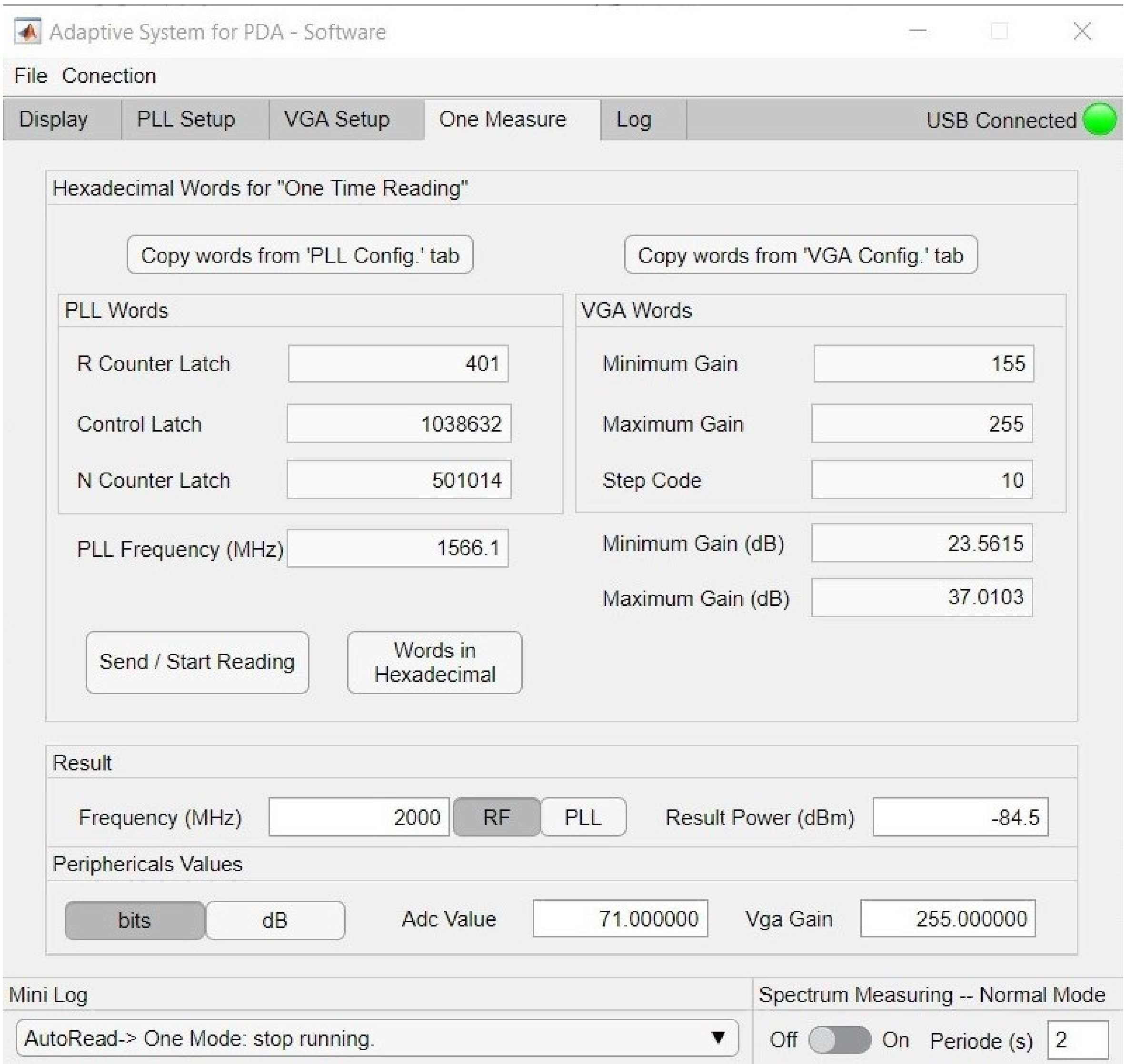
Spectrum
Ref Level : 0.0 dBm RBW : 1 MHz SWT : 20 ms Trace Mode : Clear / Write
RF Attenuator : 20 dB VBW : 30 kHz Trigger Mode : Free Run Trace Detector : Auto Peak

Start Frequency : 1.9375 GHz Stop Frequency : 2.0625 GHz
Operator : - - - Center Frequency : 2 GHz Span : 125 MHz

Figure 5.2: Sine RF signal at 2 GHz generated by VSG, photo taken on the FSH4View.

In Figure 5.3, it is illustrated the One Measure configuration used to perform this test, and as it can be seen, the PLL frequency target was 1566.1 MHz, which when added to the 433.9 MHz of the SAW filter, will correspond to the RF signal of 2GHz. This frequency is where the input signal has the most power, with a value of -12 dBm.

This input signal is within the system's reading range, but as it can be seen on the result panel of the One Measure tab, even with the VGA at the max code gain, the voltage read on the ADC was lower than the minimum accepted value. The resulting -84.5 dBm represents the minimum value that the system can read, therefore power levels below this value will be always represented as this minimum value of -84.5 dBm.



Hexadecimal Words for "One Time Reading"

Copy words from 'PLL Config.' tab
Copy words from 'VGA Config.' tab

PLL Words

R Counter Latch: 401

Control Latch: 1038632

N Counter Latch: 501014

PLL Frequency (MHz): 1566.1

VGA Words

Minimum Gain: 155

Maximum Gain: 255

Step Code: 10

Minimum Gain (dB): 23.5615

Maximum Gain (dB): 37.0103

Result

Frequency (MHz): 2000 [✓ RF] PLL

Result Power (dBm): -84.5

Periphericals Values

[✓ bits] dB

Adc Value: 71.000000

Vga Gain: 255.000000

Mini Log

AutoRead-> One Mode: stop running.

Spectrum Measuring -- Normal Mode

[✓ Off] On Periode (s): 2

Figure 5.3: Initial test with the sine signal at 2GHz on the One Measure tab of the MATLAB application.

To detect the error, it was performed a test in pre-determined parts of the system. For this test, a spectrum analyzer and a passive probe, represented in Figure 5.4, was added to the workbench. With these two devices it should be possible to measure the power and frequencies of a signal at the testing points of the system.

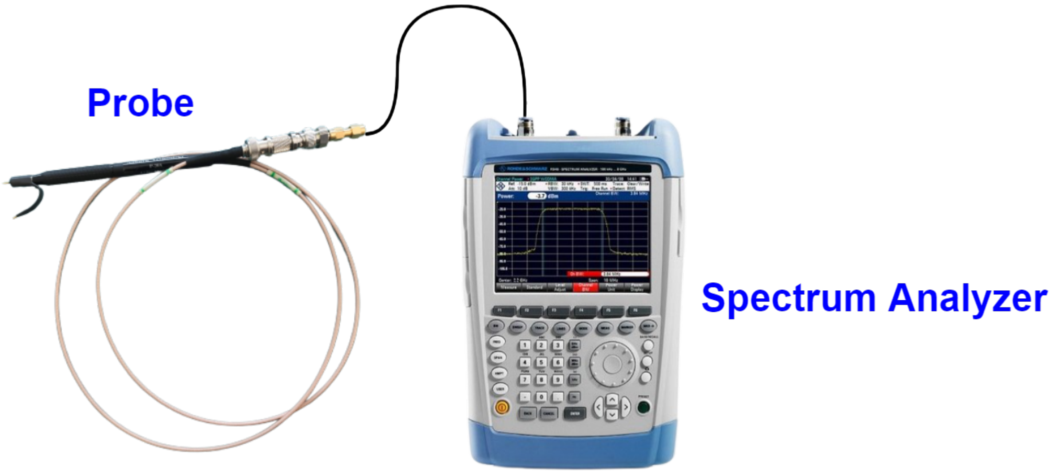


Figure 5.4: Spectrum Analizer with a passsive probe.

The used probe is capable of measuring signals below the 3 GHz frequency, with a 50 Ω impedance load.

There was other probe capable of measuring the IF frequencies at 433.9 MHz. This probe has a high impedance load, however this load presents a parallel capacitor, corresponding to the parasite capacitance. At the working frequencies, this capacitance component lowers the probe's load to values that are not acceptable.

The probe with 50 Ω of load was not the best solution, since this impedance does not match with the impedance of the system's test points.

As there was no another option available, the load with 50 Ω was used, and with that, it is not possible to measure the real power of the signal. However, this probe can still be used to measure the signal's frequencies on the testing points.

In Figure 5.5, it is represented the system board. The 3 red circles are the testing points used, where the first was used to measure the output PLL signal, the second the output from SAW filter and lastly the third point the VGA output.

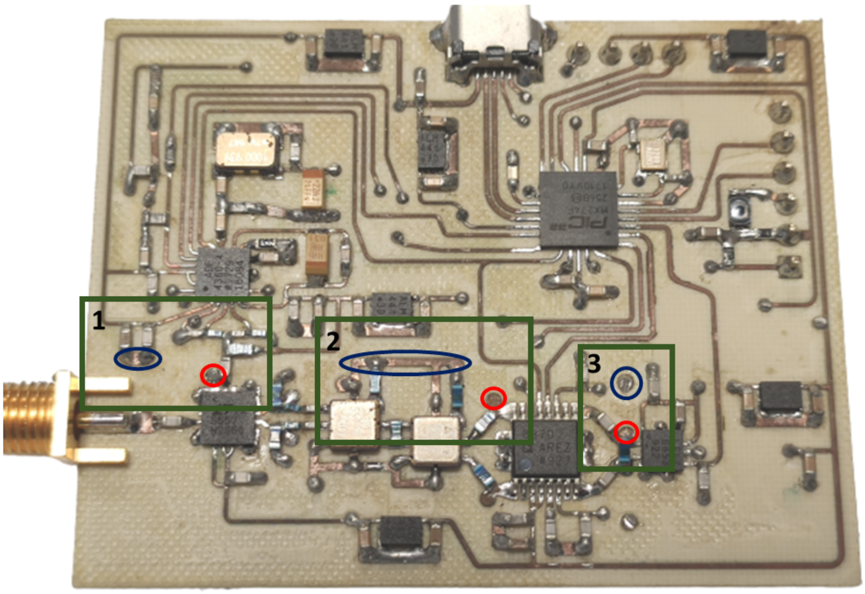


Figure 5.5: System testing points, (the reds circles represent the testing points and the blues the nearest grounds), to measure: 1- PLL output, 2- SAW filter output, 3- VGA output.

To perform these tests, it was used the same One Measure Configuration, which can be seen in Figure 5.3, and the same RF input signal, represented in Figure 5.2.

5.1.1 PLL Frequency

This test was made to see if the PLL works as expected, for that, it was used the first test point. The results can be seen on Figure 5.6.

In this figure, it can be observed that the PLL generated a signal with the central frequency at 1566.1 MHz, which is the intendent PLL frequency as it can be seen on the Figure 5.3.

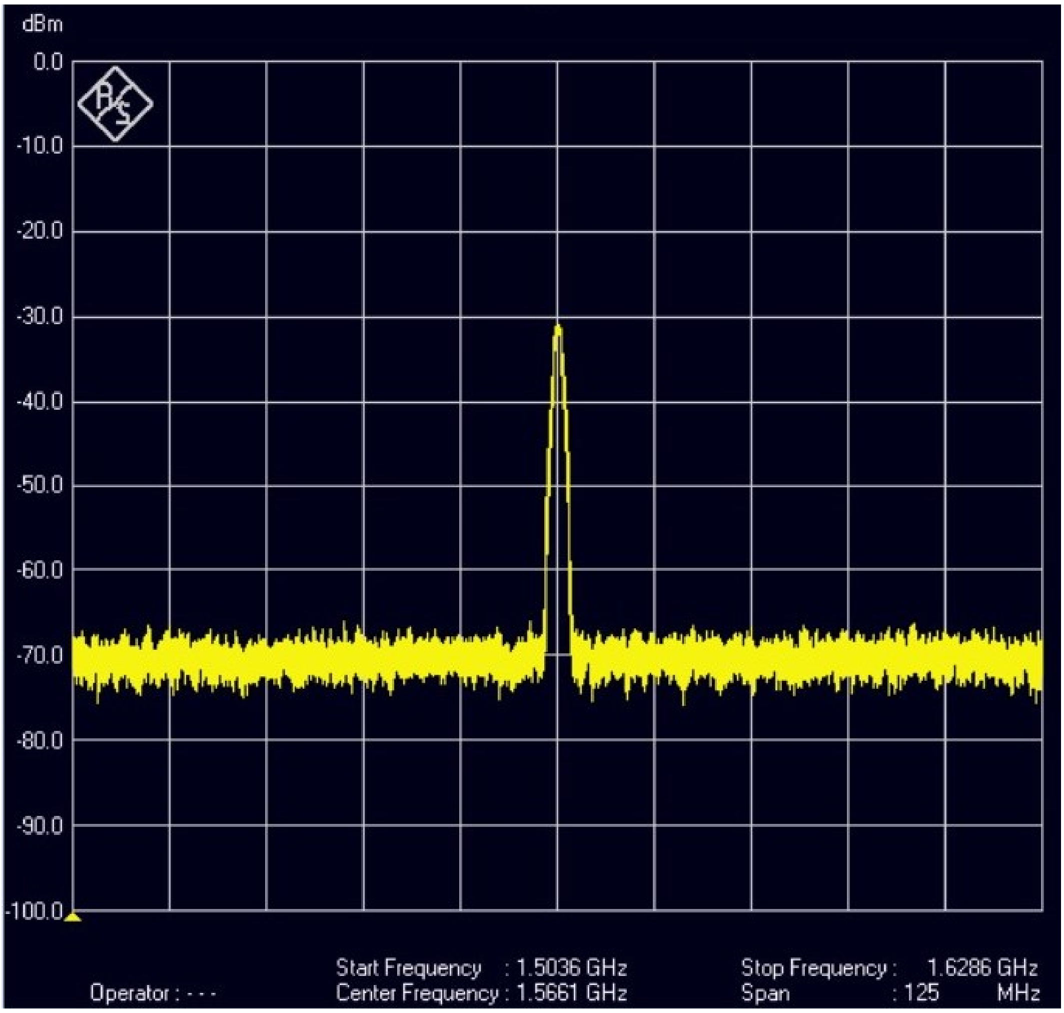


Figure 5.6: PLL output frequency, photo taken on the FSH4View.

5.1.2 SAW Filter

For the SAW filter, the second test point was used, on the spectrum analyzer side, the central frequency was changed to 433.9 MHz which corresponds to the central frequency of the filter.

As can be seen on the results presented in the Figure 5.7, the SAW filter is also working as intended.

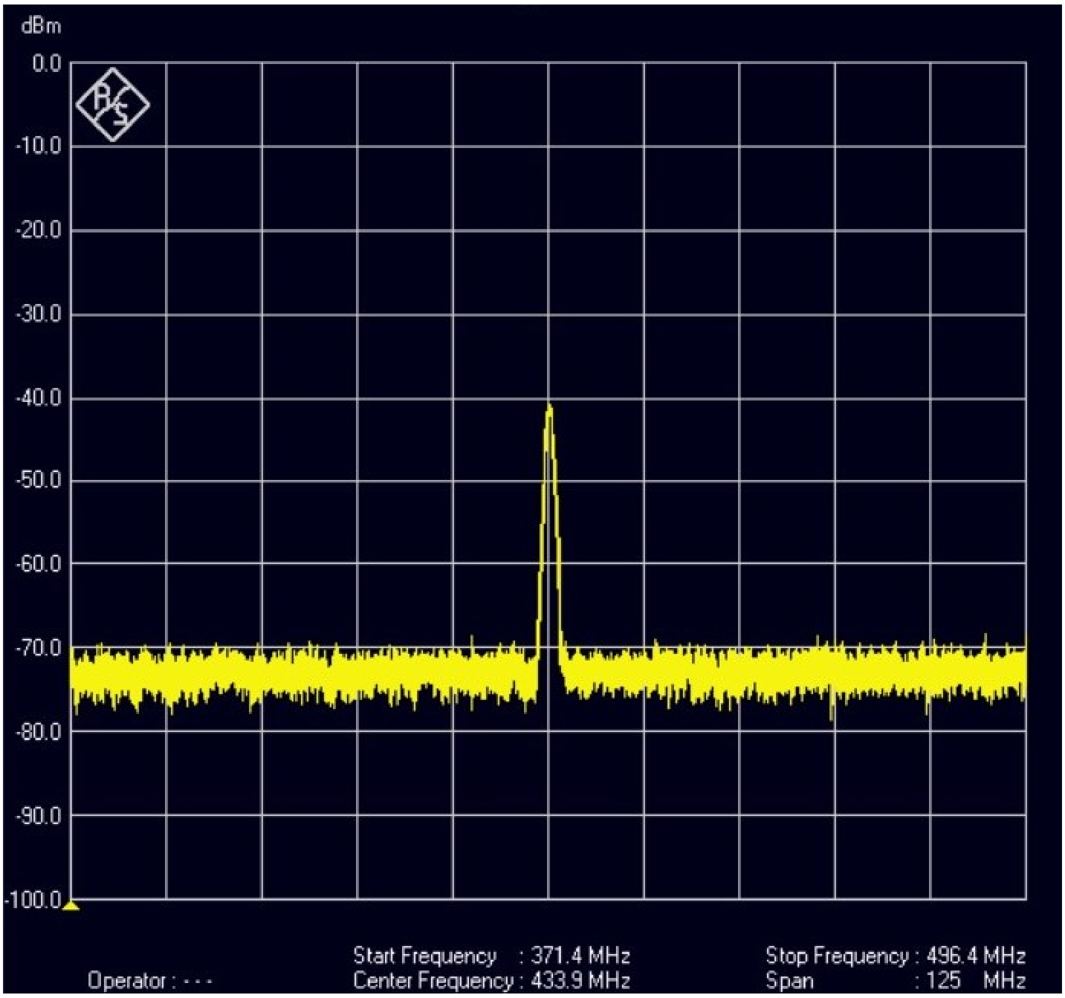


Figure 5.7: SAW filter output frequency, photo taken on the FSH4View.

5.1.3 VGA Gain

Finally, by using the third test point, it was possible to measure the output power of the VGA, which is represented on Figure 5.8.

In this test, the spectrum analyzer was set with the same configuration of the SAW filter, but the results demonstrate that the power in this test point was lower than on the previous test, which means that the VGA device attenuated the signal instead of adding gain.

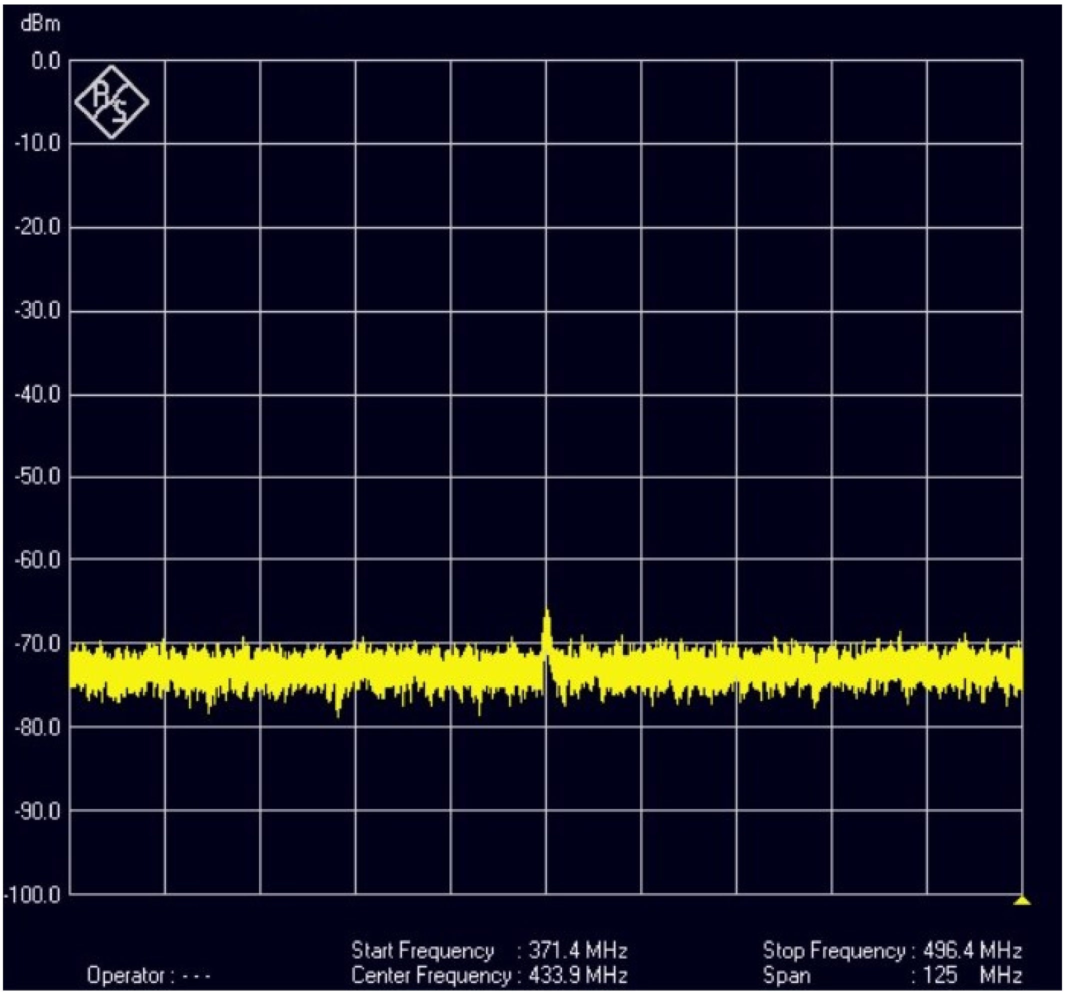


Figure 5.8: VGA output frequency, photo taken on the FSH4View.

By identifying the problem, it was possible to focus on what needs to be studied so that it could be understood what was causing this issue.

The problem was that the SAW filter is connected to the ground and it is also connected to the input of the VGA by the matching network. As the VGA needs a bias point to work properly, by connecting its input to the ground the internal bias point was being cancelled, resulting in its malfunction.

This problem was easily solved by adding two coupling capacitors, one to each input port of the VGA.

5.2 System Tests

With the initial tests, it was possible to repair and achieve the final system board, which it is presented in Figure 5.9.

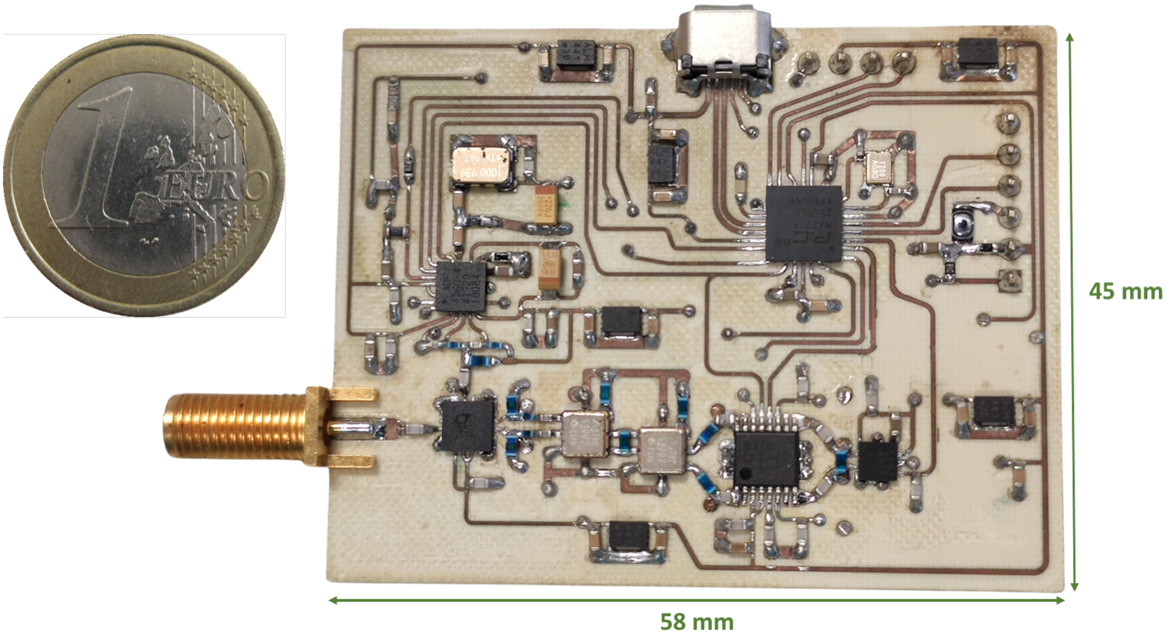
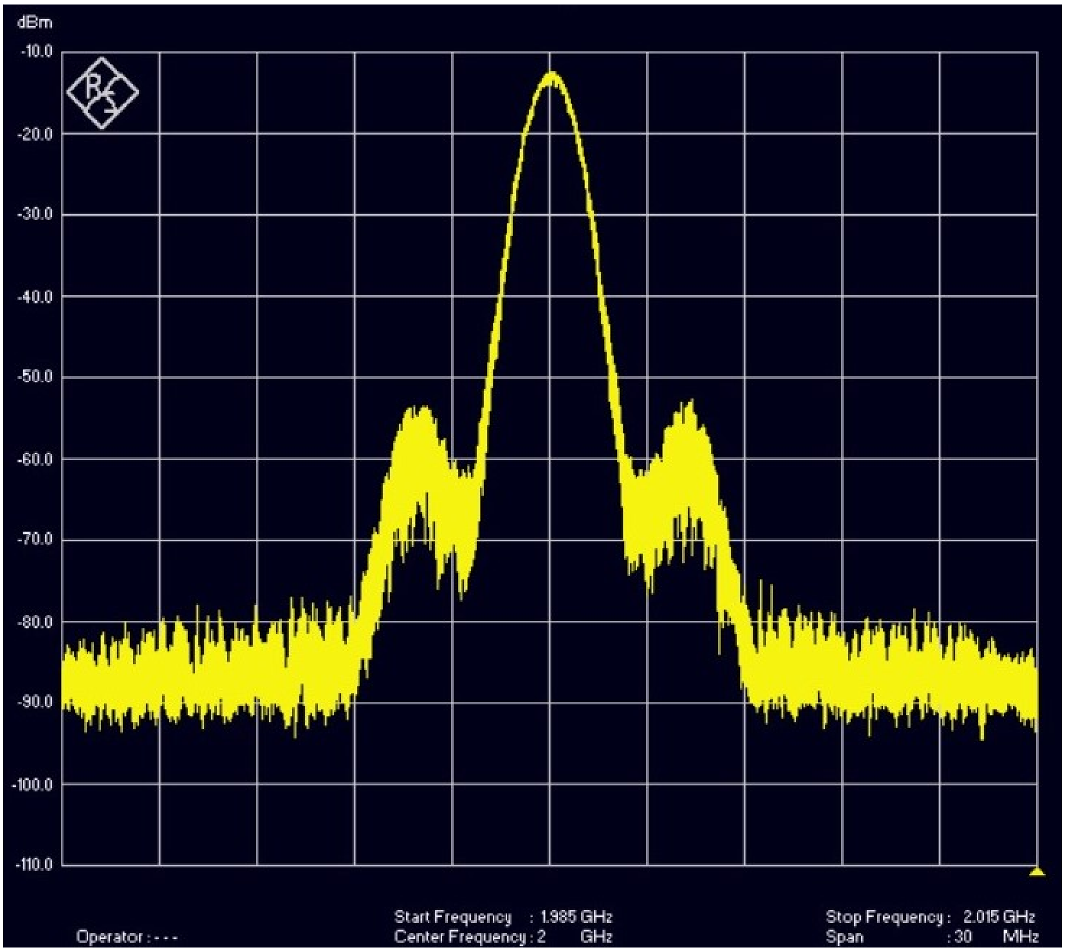


Figure 5.9: Final system board.

With this system board it was possible to advance to the final tests.

For these tests it was used the same workbench as represented in Figure 5.1, but instead of using only a sine signal it was chosen a more complex signal with a higher range and power variation over the signal bandwidth.

For the RF input signal, it was used a 64 Frequency-Shift Keying (FSK) signal generated by the VSG, this signal is available on the digital customize option of the base band signal and it is represented in Figure 5.10.



Operator : - - -

Start Frequency : 1.985 GHz
Center Frequency : 2 GHz

Stop Frequency : 2.015 GHz
Span : 30 MHz

Figure 5.10: 64 FSK signal at 2 GHz generated by VSG, photo taken on the FSH4View.

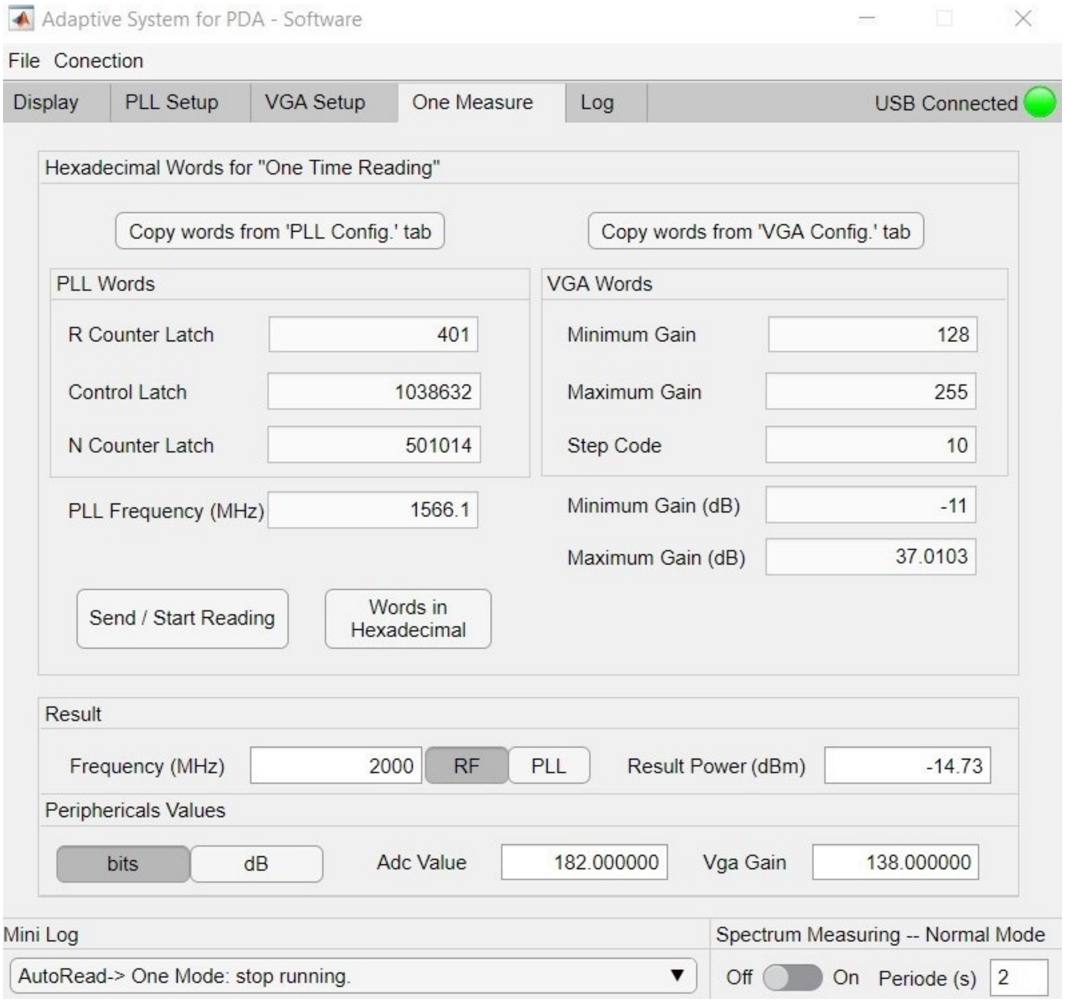
5.2.1 One Measurement

As it was done before on the initial test, it was chosen the One Measure as a first test, since this is simpler than the Normal Measure and can demonstrate if the devices are working as expected.

A One Measure test is represented in Figure 5.11, the results were as expected, and it demonstrated that the devices were all working fine.

Another aspect to reference is the resulting power value of -14.73 dBm. This test was performed with the previous mentioned RF signal as input, therefore the expected result should be -12 dBm at the frequency 2 GHz, as it can be seen on Figure 5.10.

A close, but not exact result was also anticipated, since both conversions, the VGA gain code to power, and the power detector's output voltage to power, were done using theoretical values only.



Hexadecimal Words for "One Time Reading"

PLL Words

R Counter Latch 401

Control Latch 1038632

N Counter Latch 501014

PLL Frequency (MHz) 1566.1

VGA Words

Minimum Gain 128

Maximum Gain 255

Step Code 10

Minimum Gain (dB) -11

Maximum Gain (dB) 37.0103

Result

Frequency (MHz) 2000

Result Power (dBm) -14.73

Peripherricals Values

Adc Value 182.000000

Vga Gain 138.000000

Mini Log

AutoRead-> One Mode: stop running.

Spectrum Measuring -- Normal Mode

Periode (s) 2

Figure 5.11: System test with the 64 FSK signal at 2GHz on the One Measure tab of the MATLAB application.

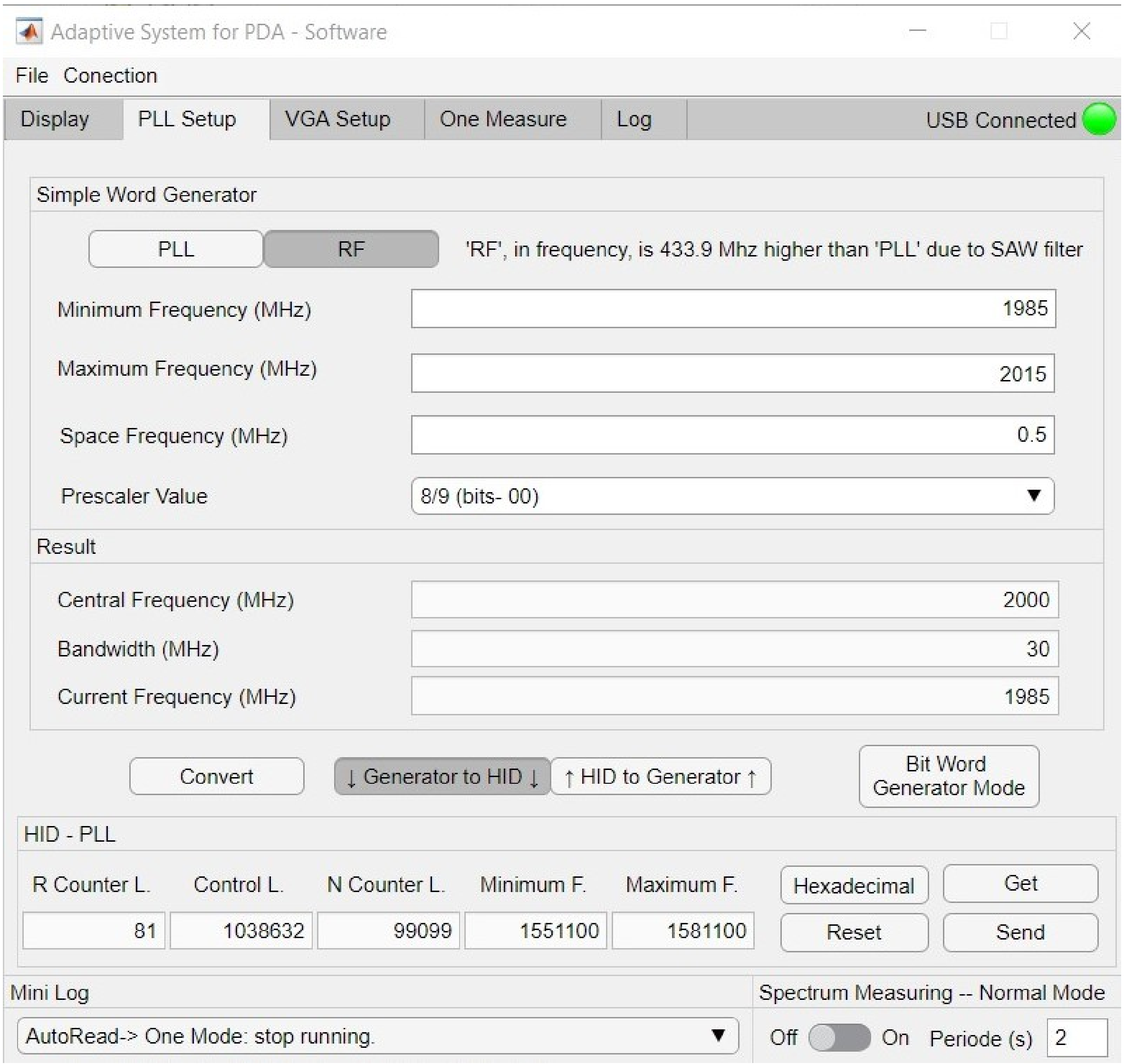
5.2.2 Normal Measurement

To perform a Normal Measure test, first the user needs to configure the PLL Setup and the VGA Setup and then send them to the system board.

The PLL setup was configured to work from 1985 MHz to 2015 MHz, these frequencies are the same used by the spectrum analyzer to measure the input signal, so that it is possible compare both results later.

The space frequency was set to 0.5 MHz, this step is small enough to study the signal behaviour, and big enough to perform a quick measure.

These configurations were set by using the simple configuration of the PLL Setup tab of the MATLAB, which is represented in Figure 5.12.



'RF', in frequency, is 433.9 Mhz higher than 'PLL' due to SAW filter

Minimum Frequency (MHz): 1985

Maximum Frequency (MHz): 2015

Space Frequency (MHz): 0.5

Prescaler Value: 8/9 (bits- 00)

Central Frequency (MHz): 2000

Bandwidth (MHz): 30

Current Frequency (MHz): 1985

R Counter L.	Control L.	N Counter L.	Minimum F.	Maximum F.
81	1038632	99099	1551100	1581100

AutoRead-> One Mode: stop running.

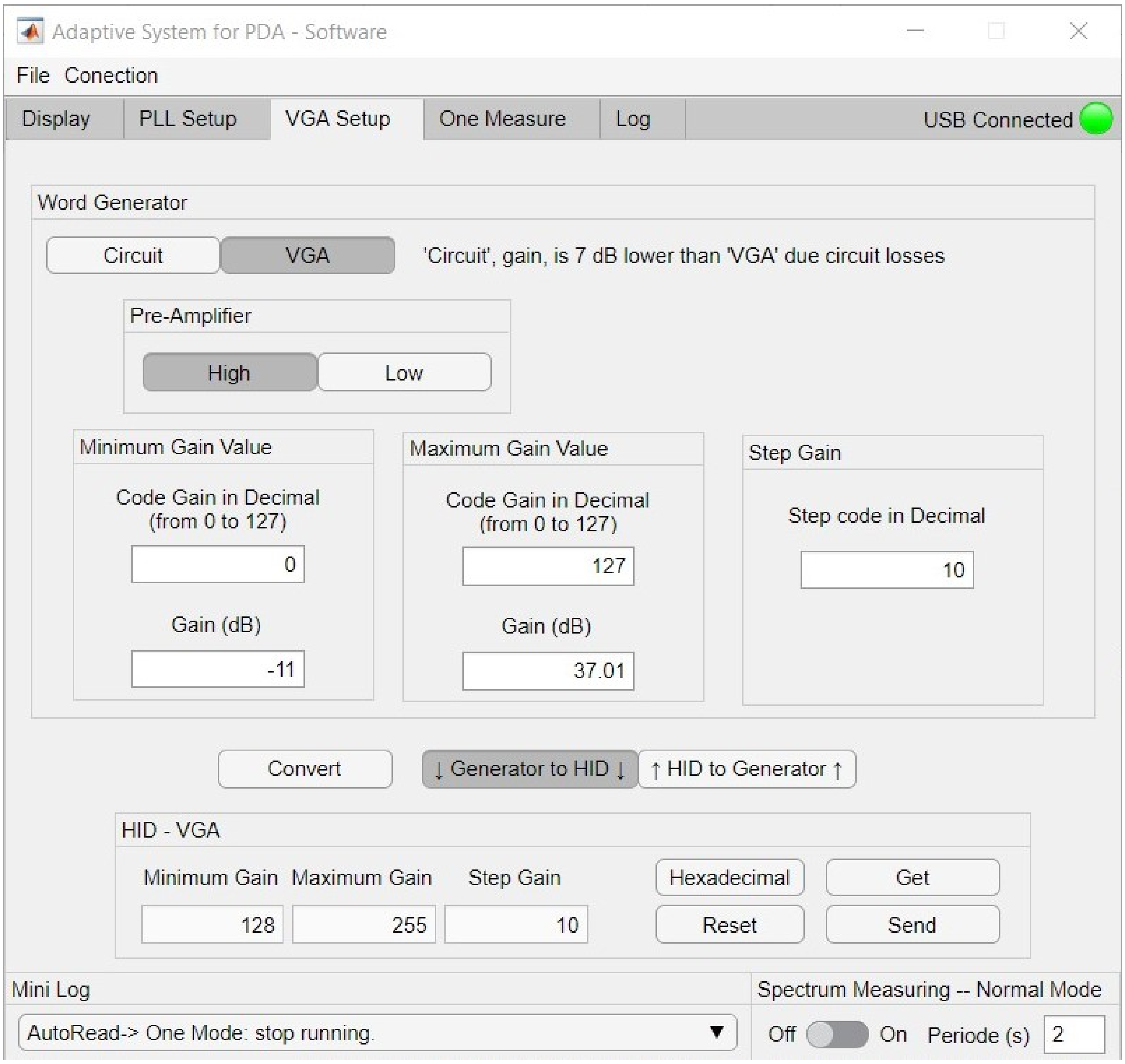
Periode (s): 2

Figure 5.12: PLL configuration used on the Normal Measure test.

The Figure 5.13 illustrates the VGA setup, which was set to use the high gain mode, thus providing a higher gain range, and allowing to reach the maximum gain of 37 dB.

The gain code was set to start at 0, when the VGA behaves as an attenuator, and to finish at its maximum code of 127, that corresponds to 37 dB of gain.

The step code chosen was 10, this step corresponds to a gain of approximately 4 dB. This step is limited by the power detector's reading range and it can be set to higher values for faster tests.



'Circuit', gain, is 7 dB lower than 'VGA' due circuit losses

Minimum Gain Value
Code Gain in Decimal (from 0 to 127): 0
Gain (dB): -11

Maximum Gain Value
Code Gain in Decimal (from 0 to 127): 127
Gain (dB): 37.01

Step Gain
Step code in Decimal: 10

HID - VGA

Minimum Gain	Maximum Gain	Step Gain
128	255	10

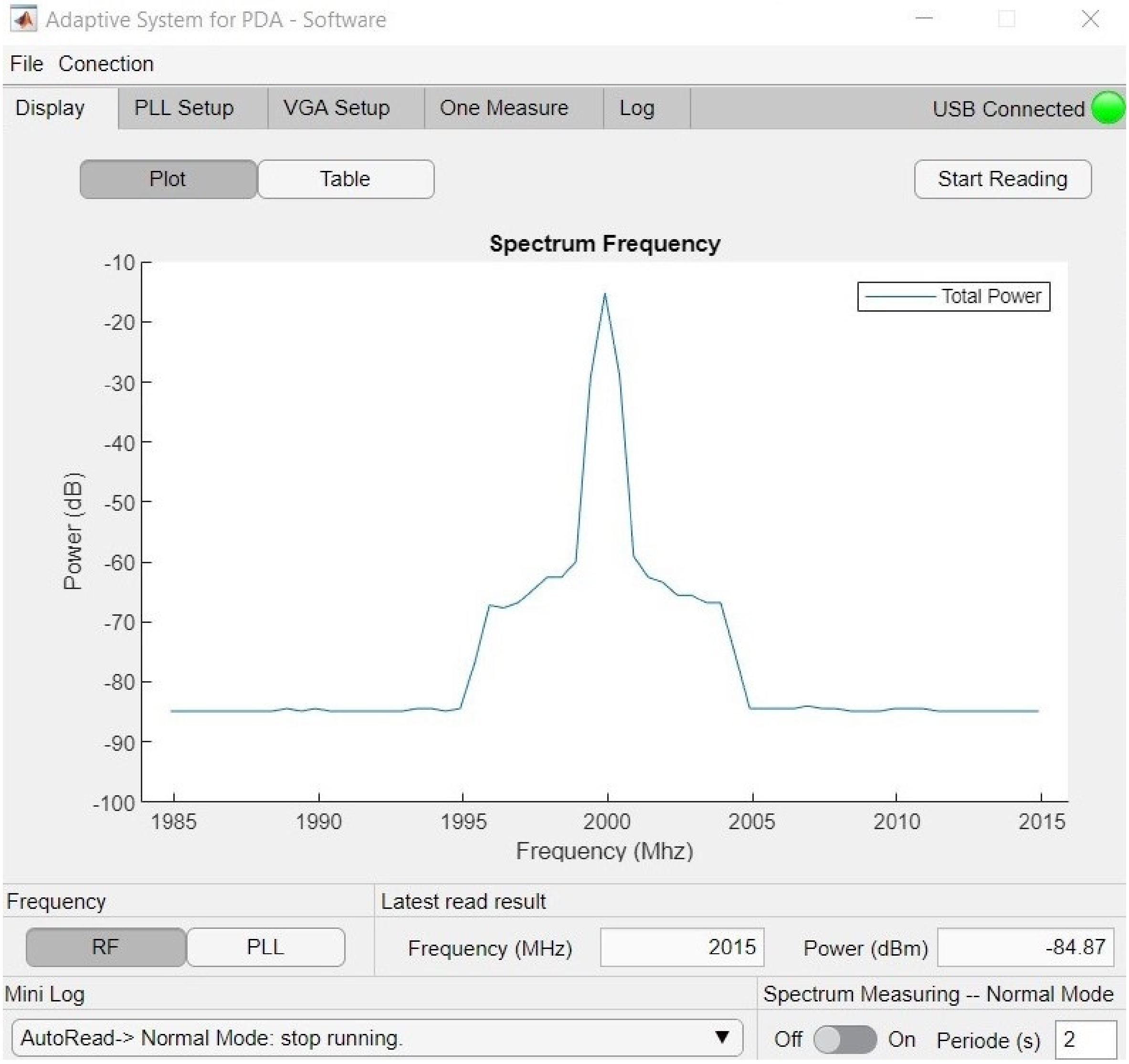
Mini Log
AutoRead-> One Mode: stop running.

Spectrum Measuring -- Normal Mode
Off On Periode (s) 2

Figure 5.13: VGA configuration used on the Normal Measure test.

To run a Normal Measure, it was used the Start Reading button on the display tab. In this tab the user can choose two different ways of displaying the real time result, either by a graph or by a table.

Figure 5.14 represents the tab with the results displayed in the graph mode, the performed Normal Measure took 40 seconds to fully complete and when finished, it presented the results in a graph similar to the input signal.


Adaptive System for PDA - Software

USB Connected

Latest read result

Frequency (MHz): 2015

Power (dBm): -84.87

Mini Log: AutoRead-> Normal Mode: stop running.

Spectrum Measuring -- Normal Mode

Periode (s): 2

Figure 5.14: Normal Measure result for the input 64 FSK signal on the Display the tab of the MATLAB application.

Finally, it was possible to save the measurements obtained with the Normal Measure method to subsequently compare them with the results from the spectrum analyzer.

To compare the results, a script on MATLAB was created. This script creates two graphs, one with the Normal Measure files and the other one with the files obtained from the spectrum analyzer, these resulting graphs can be seen overlapped on Figure 5.15.

By comparing the graphics, the system is working as predicted, with only two problems.

The first problem is due to the system's results being calculated through theoretical values, for better results it should be done a calibration on the system.

The second problem is the power range of the system, as it can be seen on the graph, the system only can reach -85 dBm, a problem referred before.

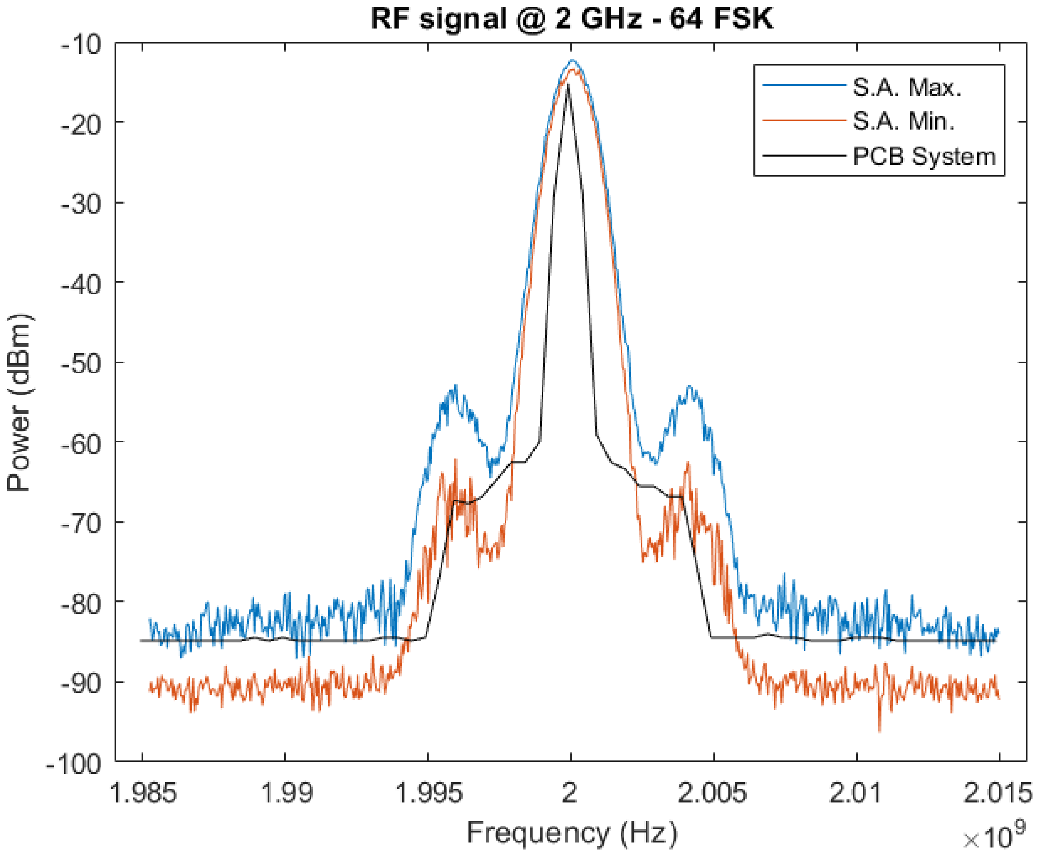


Figure 5.15: Overlap result for the input 64 FSK signal tests.

5.2.3 SAW Filter

A last test was created to characterize the SAW filter and understand the filter's behaviour. Using the same input signal, one more One Measure tests was performed at a frequency of 1991 MHz, which is a frequency with no signal power. With this test, represented on Figure 5.16, it can be seen how the filter attenuates the peak power at 2 GHz and how the not wanted power affects the results on the reading frequency.

According to the designed cascade filter theoretical characteristics given by the simulation made on ADS and represented in Figure 3.22, the SAW filter should be capable of attenuating more than 90 dB on its stopband. But these values do not represent the reality, since there is signal power outside of the bandpass of the filter, as it can be seen in Figure 5.16.

In this figure, the measures obtained with 1991 MHz should not contain any signal power, but as it can be seen, the filter did not attenuate all of the signal at 2000 MHz, where it originally had a power of -12 dBm, this could lead the system to measure the residual power at the stopband zone of the filter instead of measuring the intended floor level power on the current frequency.

On Figure 5.17, the one measure tab of this test, it can be seen that the result power was -84.44 dBm which indicates that even though the filter was not able to completely attenuate the signal, it indicates that it did not interfere with the test since the peak power of the signal had power lower than the minimum value that can be read by the system.

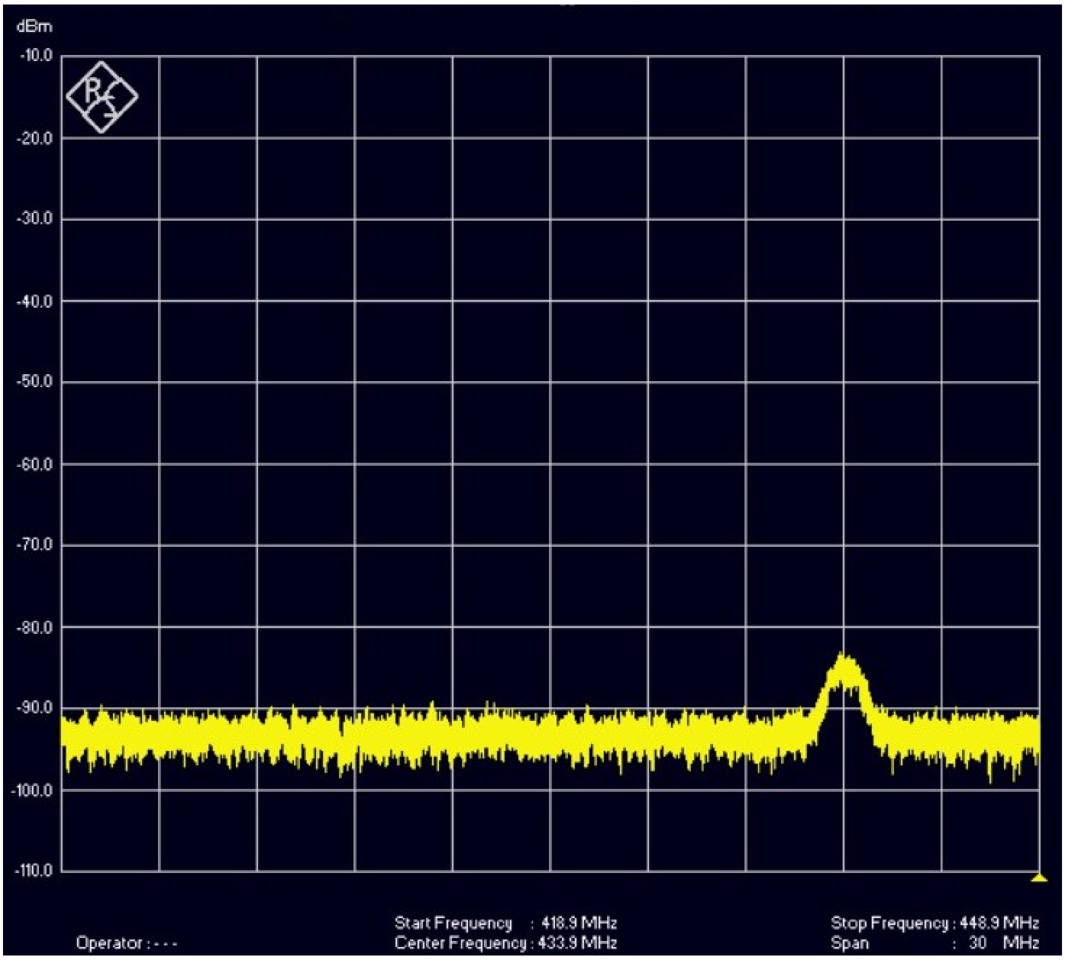
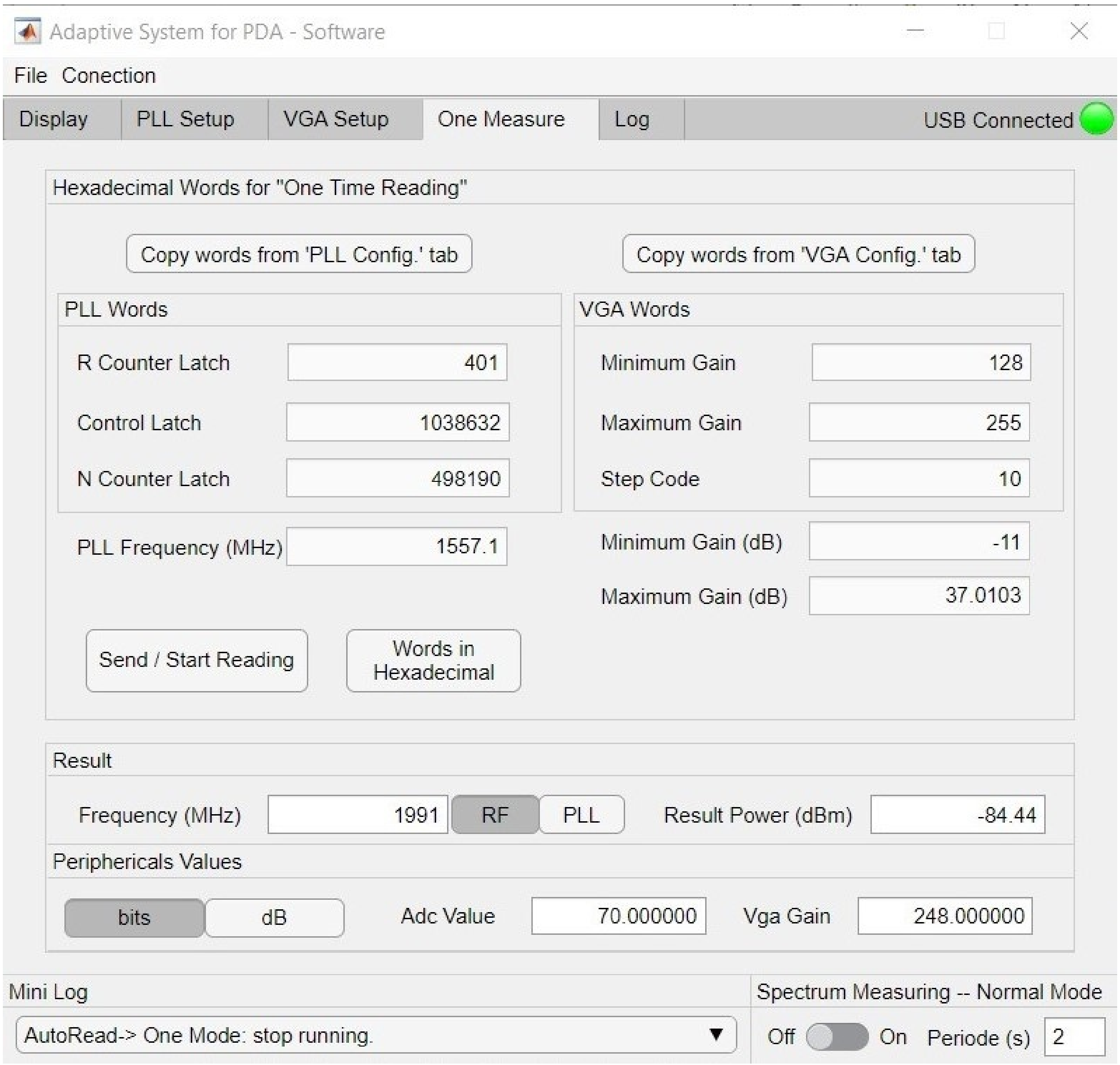


Figure 5.16: SAW filter output with a central frequency of 1991 MHz, photo taken on the FSH4View.



Hexadecimal Words for "One Time Reading"

PLL Words

R Counter Latch: 401

Control Latch: 1038632

N Counter Latch: 498190

PLL Frequency (MHz): 1557.1

VGA Words

Minimum Gain: 128

Maximum Gain: 255

Step Code: 10

Minimum Gain (dB): -11

Maximum Gain (dB): 37.0103

Result

Frequency (MHz): 1991 (RF)

Result Power (dBm): -84.44

Periphericals Values (bits)

Adc Value: 70.000000

Vga Gain: 248.000000

Mini Log

AutoRead-> One Mode: stop running.

Spectrum Measuring -- Normal Mode

Off / On

Periode (s): 2

Figure 5.17: SAW filter test with the 64 FSK signal at 1991 MHz on the One Measure tab of the MATLAB application.

Chapter 6

Conclusive Remarks

6.1 Discussion and Conclusions

The aim of this work was to create a system capable of measuring the spectrum signal of an RF Amplifier's output. In a future work, this system should be used in the tuning process of a predistorter, therefore it was used a microcontroller capable of processing the necessary calculations to generate the control signal, and then send it through ADCs to the predistorter.

The built system was mainly composed by a microcontroller, a PLL, a mixer, two SAW filters, one VGA and an RMS power detector. The combination of these devices created a reliable system capable of achieving the proposed goals.

At the beginning, another method was studied, but when compared with the used method, the chosen solution was deemed to be a safer option to achieve a fully working system.

Various objectives were met with the resulting system:

The system was able to measure signals with a central frequency of 2 GHz and this was achieved through a matching of the four devices, the PLL, mixer and two SAW filters. The PLL provides a high bandwidth to the system, allowing it to achieve 300 MHz of bandwidth, which is higher than the 20 MHz established as objective. Moreover, the PLL also offers the step between frequency from 0.1 MHz to 1 MHz, which also reaches the objective of 1 MHz of space frequency.

The final system can measure a power signal from around -85 dBm to 8 dBm, this power range is achieved by joining the VGA with a RMS power detector. This accomplishes the objective of having an input signal with a peak power of -10 dBm. However, the desired power range of 90 dB is not met due to the developed cascade filter as it can only attenuate 75 dB on its stopband. Therefore, the system can only provide a maximum 75 dB of power range.

Finally, for the last objective, no DAC was added, since there was no predistorter to control, the solution was provided by I^{2}C, which is a common interface used by microcontrollers to communicate with DACs, to be able to control the future predistorter's DACs.

The developed MATLAB application provides an intuitive interface to control the system, and also produce real-time graphs and tables, which allows the user to look at the system results in a fast and intuitive way.

The PCB has fully functional firmware, that can easily be adapted to operate without depending on the MATLAB application. Furthermore, its state machines provide a simple way to further add any necessary code to control the predistorter.

Concluding, most of the work objectives were accomplished, finishing with a functional system that could be integrated on a future predistorter, leading to a better linearization and consequently allowing an RF amplifier to be more efficient.

6.2 Future Work

The final system has some points to improve.

First, a new layout should be printed with no test points, and if possible, a better PCB print, in this situation, a new layout should be designed to take into consideration the new limitations, as these would ensure a better filter and therefore a better power range.

Secondly, if even with the previous point, the desired power range could not be achieved, a new filter design should be considered.

Finally, more tests should be done to better characterize the system and improve its efficiency. There should be tests that measure the power that the system consumes and a new test case where the output power of the PLL is reduced. The VGA can be powered by 3.3 V, a supply voltage not tested on this thesis. Lastly, another type of microcontroller could be used, since power consumption is a big factor on the goals of the project, an alternative that consumes less power and be able to process the same information can be explored.

Beyond these improving points, there is some follow up work for this project: Adding a predistorter to the project, this will allow the creation of a control signal and the implementations of DACs.

Finally, adding a RF amplifier will provide more practical tests which leads to the final objective of creating a more efficient and linear RF amplifier.

References

[1] T. E. Bogale and L. B. Le, “Massive MIMO and mmWave for 5G Wireless HetNet: Potential Benefits and Challenges”, *IEEE Vehicular Technology Magazine*, vol. 11, no. 1, pp. 64–75, 2016.

[2] *Everything You Need to Know About 5G.* [Online]. Available: `https://spectrum.ieee.org/video/telecom/wireless/everything-you-need-to-know-about-5g`.

[3] D. Lie, J. C. Mayeda, and J. Lopez, “Highly efficient 5G linear power amplifiers (PA) design challenges”, *2017 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2017.

[4] P. M. Lavrador, T. R. Cunha, P. M. Cabral, and J. Pedro, “The Linearity-Efficiency Compromise”, *IEEE Microwave Magazine*, vol. 11, no. 5, pp. 44–58, 2010.

[5] S. N. Ali, P. Agarwal, S. Gopal, and D. Heo, “Transformer-Based Predistortion Linearizer for High Linearity and High Modulation Efficiency in mm-Wave 5G CMOS Power Amplifiers”, *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 7, pp. 3074–3087, 2019.

[6] “RECEIVER SPECTRUM STANDARDS Phase 1 – Summary of Research into Existing Standards”, NTIA. [Online]. Available: `https://www.ntia.doc.gov/legacy/osmhome/reports/ntia03-404/NTIAREPORT03-404.pdf`.

[7] J. C. Pedro and N. B. Carvalho, *Intermodulation distortion in microwave and wireless circuits.* Artech House, 2003.

[8] A. S. Sedra and K. C. Smith, *Microelectronic circuits.* Holt, Rinehart and Winston, 1982.

[9] F. U. Dowla, *Handbook of RF and wireless technologies.* Newnes, 2004.

[10] A. Mohammadi and F. M. Ghannouchi, *RF Transceiver Design for MIMO Wireless Communications.* Springer Berlin, 2014, pp. 64–75.

[11] I. Collins, *Phase-Locked Loop (PLL) Fundamentals.* [Online]. Available: `https://www.analog.com/en/analog-dialogue/articles/phase-locked-loop-pll-fundamentals.html`.

[12] "SAW RF filter Short range devices, B39431B3790Z810 (Data sheet)", RF360 - Qualcomm. [Online]. Available: https://pt.mouser.com/datasheet/2/842/B3790-1092449.pdf.

[13] "Integrated Synthesizer and VCO, ADF4360-4 (Data sheet)", Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADF4360-4.pdf.

[14] "400 MHz to 3.8 GHz 3.3V, Active Downconverting Mixer, LT5557 (Data sheet)", Linear Technology - Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/5557fc.pdf.

[15] "40 MHz to 10 GHz, RMS Power Detector, with 57dB Dynamic Range, LTC5582 (Data sheet)", Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/LTC5582.pdf.

[16] "LF to 750 MHz, Digitally Controlled VGA,LF to 750 MHz, Digitally Controlled, AD8370 (Data sheet)", Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8370.pdf.

[17] "PIC32MX1XX/2XX28/44-PIN XLP FAMILY", Microchip. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/PIC32MX1XX-2XX-28-44-Pin-XLP-Data-Sheet-60001404D.pdf.

[18] "PICkit™3In-Circuit Debugger/ProgrammerUser's Guide For MPLAB®X IDE", Microchip. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/PIC32MX1XX-2XX-28-44-Pin-XLP-Data-Sheet-60001404D.pdf.

[19] L. D. Jasio, *This is (Not) rocket science*. Lulu Enterprises, 2015.

[20] "Specification of Crystal Oscillator, XNCLH10M000CHJA2P0 (Data sheet)", Murata. [Online]. Available: https://www.murata.com/en-global/products/productdata/8801053343774/SPEC-XNCLH10M000CHJA2P0.pdf?1517839211000.

[21] "IoT OPTIMIZED LOW PROFILE QUARTZ CRYSTAL, ABM10-166-12.00 MHz (Data Sheet", Abracon. [Online]. Available: https://abracon.com/Resonators/ABM10-166-12.000MHz.pdf.

[22] "Evaluation Board User Guide, UG-099", Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/user-guides/UG-099.pdf.

[23] "PICkit™3In-Circuit Debugger/ProgrammerUser's Guide For MPLAB®X IDE", Texas Instruments. [Online]. Available: https://www.ti.com/lit/an/spraar7h/spraar7h.pdf.