Universidade de Aveiro
2013

Departamento de
Electrónica, Telecomunicações e Informática

Vitor Miguel
Melo Soares

Analisador de espectros portátil para RF usando Arduino

Handled RF spectrum analyser using Arduino

Universidade de Aveiro Departamento de
Electrónica, Telecomunicações e Informática
2013

Vitor Miguel
Melo Soares

Analisador de espectros portátil para RF usando Arduino

Handled RF spectrum analyser using Arduino

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Pedro Miguel Cabral, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática, da Universidade de Aveiro

o júri / the jury

presidente / president — **Professor Doutor Nuno Miguel Goncalves Borges de Carvalho**
Professor Catedrático da Universidade de Aveiro

vogais / examiners committee — **Professor Doutor Pedro Renato Tavares de Pinho**
Professor Adjunto, Area Departamental de Engenharia de Eletrónica e Telecomunicações e de Computadores do Instituto Superior de Engenharia de Lisboa (Arguente)

Professor Doutor Pedro Miguel da Silva Cabral
Professor Auxiliar da Universidade de Aveiro (Orientador)

agradecimentos / acknowledgements

Fazer acontecer reside na vontade prórpia de cada um, assim sendo, aproveito este espaço para expressar um pouco de emoção neste documento técnico. Naturalmente faço referência aos meus pais pela sua vontade de me possibilitar acesso a conhecimento. Aos meus colegas de estudo que se tornaram grandes amigos durante esta aventura, que da qual esta tese, assinala apenas uma etapa para nós.

Todas as pessoas que envolvidas directa ou indirectamente, possibilitaram a realização deste trabalho, como o meu prestável orientador Doutor Pedro Cabral, o hábil soldador e desenhista de placas para circuitos integrados Paulo Gonçalves ou o Sr. Pereira que nos aturou todos estes anos de curso e nós a ele.

É também normal referir que não é possível mencionar toda a gente com quem partilhamos apoio, mas para todos estes que serão recordados por mim, desejo muito amor, paz e dedicação nos altos e baixos desta vida energética, que são tão bem representados por uma onda sinusoidal.

Palavras Chave Analisador de espectros, RF, Rádio-frequência, Arduino

Resumo

Rádio-frequência é o termo utilizado para designar a gama de frequêcias utilizadas na transmissão de sinais eletromagnéticos, através do meio livre. O domínio deste fenómeno possibilita a que dispositivos eletrónicos possam comunicar sem fios.
A análise do espectro eletromagnético utilizado pelos diferentes dispositivos que coabitam no dia a dia, é essencial para possibilitar o correcto funcionamento de todos. Para tal, são utilizados analisadores de espectro, que com os quais se pode monitorizar algumas das características físicas das comunicações sem fios.
O objetivo deste trabalho é documentar o projeto, implementação e testes de validação de um analisador de espectros portátil para rádio-frequência utilizando tecnologia de controlo digital *opensource*, Arduino.

Key words Spectrum analyser, RF, Radio Frequency, Arduino

Abstract

Radio frequency term refers to the electromagnetic spectrum bandwidth used to transmission of electromagnetic signals trough free space.
The human knowledge about this phenomena turns possible to set wireless communications between electronic devices.
The analysis of that spectrum is demanding in order to achieve proper communications between those devices. Whit spectrum analysers is possible to observe some physical characteristics of the communication.
The aim of this document is to furnish information about the project, design and implementation of a spectrum analyser using open source digital control technology, Arduino.

Contents

List of Figures

List of Tables

Acronyms

ADC Analog to Digital Converter

ADNL Average Displayed Noise Level

ANACOM Autoridade Nacional de Comunicações

BGA Ball Grid Array

BPF Band Pass Filter

CRT Cathode Ray Tube

DAC Digital to Analogue Converter

DC Direct Current

DFT Discrete Fourier Transform

DSP Digital Signal Processor

DUT Device Under Test

ENIG Electroless Nickel/Immersion Gold

EEPROM Electrically Erasable Programmable Read Only Memory

FFT Fast Fourier Transform

FM Frequency Modulated

FPGA Field Programmable Gate Array

GSM Global System for Mobile Communications

HASL Hot Air Solder Leveling

HPF High Pass Filter

IC Integrated Circuit

IDE Integrated Development Environment

IF Intermediate Frequency

IMD Intermodulation Distortion

IMR Intermodulation Ratio

IO Input Output

IP3 Third-order Intercept Point

IT Instituto de Telecomunicações

ITIS Immersion Tin and Immersion Silver

LCD Liquid Crystal Display

LNA Low Noise Amplifier

LO Local Oscillator

LPF Low Pass Filter

MDS Minimum Detectable Signal

NF Noise Figure

OCXO Oven Controlled Crystal Oscillator

OSP Organic Solder Preservative

P1dB 1 dB Compression Point

PAE Power Added Efficiency

PCB Printed Circuit Board

PWM Pulse Width Modulation

QPSK Quadrature Phase-Shift Keying

RAM Random Access Memory

RBW Resolution Bandwidth

RF Radio Frequency

RTSA Real Time Spectrum Analyser

SAW Surface Acoustic Wave

SD Secure Digital

SDR Software Defined Radio

SMA SubMiniature Version A

SNR Signal to Noise Ratio

SPI Serial Peripheral Interface

SRAM Static Random Access Memory

TCXO Temperature Compensated Crystal Oscillator

TFT Thin-Film-Transistor

UART Universal Asynchronous Receiver Transmitter

USB Universal Serial Bus

VCO Voltage Controlled Oscillator

VSWR Voltage Standing Wave Ratio

VSA Vector Signal Analyser

Chapter 1

Introduction

1.1 Motivation

Wireless technology became a reality in our daily life. Electronic devices are able to exchange data through atmosphere and spreading for many different proposes. It may be transparent to human senses though there are physical constraints on data transmission over air and the comprehension of what is happening is key to turn all that possible. As for example, each established communication link between a transmitter and a receiver requires a bandwidth portion of electromagnetic spectrum.

Moreover the proliferation of radio broadcasts, cellular networks and domestic wireless applications would be interfering with each other if there were no restrictions on the electromagnetic spectrum usage. As the communications take place in a shared medium it is essential they can cohabit.

The state of the art of Radio Frequency (RF) technology restricted to such physical limitations, keeps looking to improve communications in a crowded spectrum. Engineers need to understand the conditions where wireless links are done and find solutions to come up to actual demanding standards. In practice this means to manipulate physical characteristics of electromagnetic signals to achieve transmission efficiency. There are necessary measurement tools to go ahead with telecommunications and one of them is denominated as spectrum analyser.

Since the begin of radio communications, spectrum analysers are used as an elementary instrument to inspect the used spectrum. Till today, they are a valued instrument for RF research and development laboratories, private companies operating in telecommunications business and governmental organizations which regulate wireless communications in *open air*.

Due to the quest for wireless communications, these instruments are needed to operate in indoor and outdoor environments. Benchtop versions may afford powerful processors and many other components for superior performance, turning to be heavier and bigger. Handled versions that can run on batteries, are developed to fill the need to identify interfering RF sources on field that may corrupt other wireless systems. It makes them useful tools to improve and troubleshoot wireless networks.

Any commercial version can be quite expensive. Describing the design and implementation of such instrument on a academic thesis, applying for open source software, may be a revision to the ones who need to access this kind of instruments.

1.2 Objectives

The aim of this dissertation is to project and implement a handled spectrum analyser prototype. The device should be capable to perform power measurements over frequency in a selectable bandwidth range from 880 to 950 MHz.

It is pretended to pick a Resolution Bandwidth (RBW) channel amongst different ones. Other parameters like sweeping times and measurement tools should be available to the user.

The utilization of these devices is done trough a graphic display and a keypad which also are contemplated in this document.

It will be described the necessary concepts to understand the hardware design, software control directives and practical tests to implement such device.

1.3 Dissertation outline

This document is formulated in 6 chapters and appendices, conducting the reader through the necessary subjects to project and understand a spectrum analyser.

Chapter 1 is this introduction where motivation and objectives are detailed

Background theory is introduced in chapter 2, elucidating the tackled topics to comprehend what is spectrum analysis and how to do it.

To choose an appropriate architecture in chapter 3 different approaches about collect radio signals are briefed.

The different components used to implement the spectrum analyser prototype are detailed in chapter 4.

The tests and results about the circuit functionality are in chapter 5

Concluding this document is chapter 6 with final considerations.

In appendices are additional informations to expand description details about radio circuits validation, tests results, the used PCB designs, calculus and the **Arduino** source code.

Chapter 2

Background theory

In telecommunications there is the objective to transmit data from point a to point b. When the channel between the two points is the atmosphere the transmitter and the receiver are named as radios.

The process consists to convert the intended data (audio, video, computer data, etc.) into an electromagnetic signal with proper characteristics to spread trough open space. This involves signal conditioning, modulation and frequency conversion. All these processes have physical requirements which often take place in harsh environments.

This requires very well optimized transmission techniques and spectral optimization to transmit maximum data with least bandwidth[1].

Spectrum analysers are a used tool to measure frequency utilization in communications. It is an essential piece to comprehend physical aspects about radio communications where the human senses are not able to perceive it.

Describing what is a signal, frequency spectrum, time and frequency domain measurements, operation theory of a spectrum analyser and main specifications of such instrument will introduce the reader to what is involved with spectra analysis.

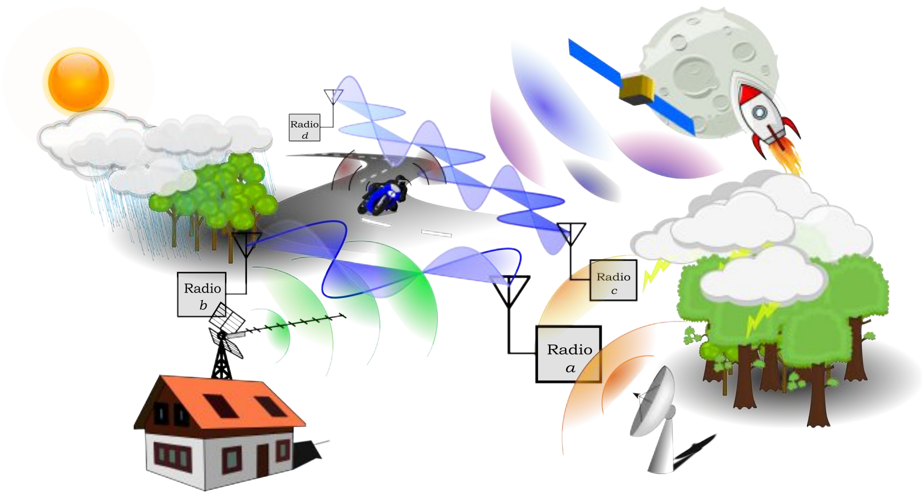


Figure 2.1: Radio communications environment

[1]There is a maximum data transmission limit over a fixed bandwidth which calculus are attributed to Claude Shannon [1].

2.1 Signals

Signals are temporal functions that may provide useful information. Traditionally represent physical characteristics of processes and can be either natural or synthesized. Furthermore they are modulated to encode data.

In electric systems a signal is a voltage or current variation [2]. It can be graphically represented as in figure 2.2. The signal present in 2.2a might be not predictable. It is not easy to mathematically describe it.

With the propose to evaluate signal transmission over networks, electronic science use controlled sinusoidal signals as depicted in figure 2.2b. These sinusoidal functions are mathematically represented by an the amplitude A (in volts) and period T (in seconds). A sinusoid signal may be mathematically expressed as:

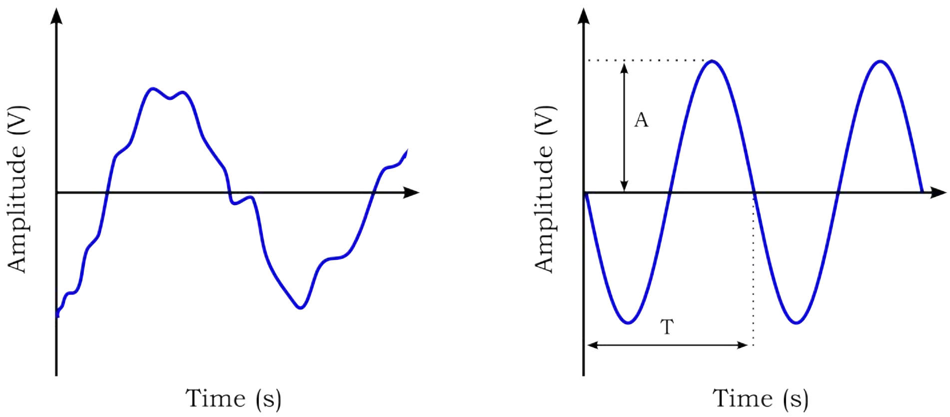
$$v(t) = A\sin(\omega t + \Phi) \tag{2.1}$$

The angular frequency of a sinusoid is represented with $\omega = 2\pi f(rad/s)$ and period is defined as the inverse of frequency stating $f = \frac{1}{T}Hz$. Φ represents the signal phase. It can be understood as a time shift in electromagnetic signals, but for now it will not be take in consideration.

More complex signals can be represented as a sum of diverse sinusoids. Equation 2.2 expresses the stated before.

$$v(t) = A_0\sin(\omega_0 t) + A_1\sin(\omega_1 t) + ... + A_n\sin(\omega_n t) \tag{2.2}$$

This signal description resulting from the sum of sinusoids was introduced by Jean Fourier. Fourier series and Fourier transform are mathematical tools widely used in signal characterization and processing. They have an important role in spectra analysis once they link time domain with frequency domain [2, 3, 4].



(a) Arbitrary signal (b) Sinusoidal signal

Figure 2.2: Signal representation

2.2 Frequency spectrum

As mentioned in equation 2.2 it is possible to perceive that a signal can contain n different frequency components represented as ω_n or f_n. This concept allows to modulate signals and is how telecommunications techniques controls and processes signals [5].

Considering three independent signals $v_0(t)$, $v_1(t)$ and $v_2(t)$

$$v_0(t) = \sin(2\pi f_0 t),\ v_1(t) = \frac{1}{3}\sin(2\pi f_1 t),\ v_2(t) = \frac{1}{5}\sin(2\pi f_2 t) \tag{2.3}$$

The sum of these signals leads to a new one, represented as $v(t)$:

$$v(t) = v_0(t) + v_1(t) + v_2(t) \tag{2.4}$$

Assuming $f_0 = 1f$, $f_1 = 3f$ and $f_2 = 5f$ it is possible now to say that signal $v(t)$ will have three different frequency components and this represents the signal's spectrum. This example relates to the first three frequency components which characterize a square wave.

Ideally the output spectrum for a single frequency oscillator may be represented as in figure 2.3a. Because of noise both frequency and amplitude are affected, turning the real world spectrum components to be seen with skirts as represented in figure.

Concepts like spectral occupancy start to emerge. In the case of signal $v(t)$, it requires $5fHz$ of bandwidth. As an example about the practical importance of this, adjacent communication links which operates at different frequency bands need to restrain their spectral emissions. Otherwise they can interfere with the surrounding channels. This situation it is not desired because it may corrupt transmitted data.

As a practical example of spectrum usage and regulation in Portugal, Autoridade Nacional de Comunicações (ANACOM) which is the governmental agency to legal conduct communications, sets the frequency band from 87,5 MHz to 108 MHz for Frequency Modulated (FM) radio broadcasting [6]. Another example is the regulated bandwidth for Global System for Mobile Communications (GSM) which fits in the 880 to 890 MHz band plus 925 to 930 MHz, which the intended prototype for this dissertation will focus.

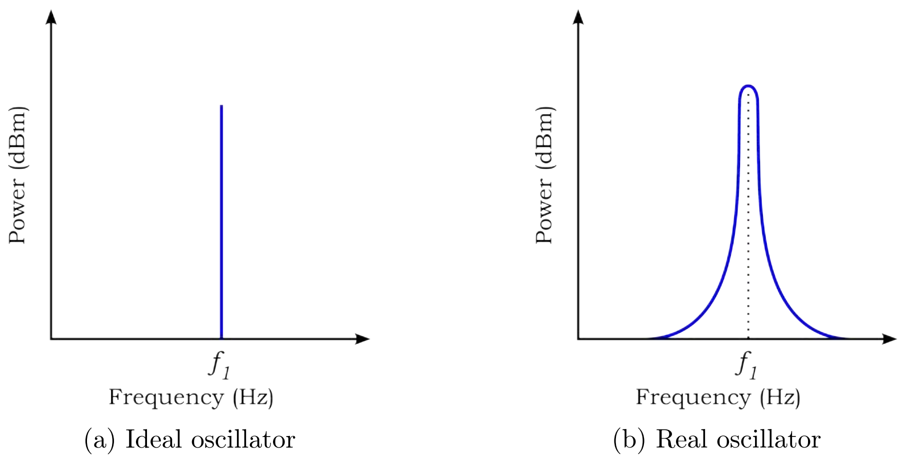


Figure 2.3: Ideal and real spectrum representations [7]

2.3 Time and frequency domain

In figure 2.4 is represented a three-dimensional coordinate system that will be used to understood signal characteristics. In the axes are represented amplitude, frequency and time. Signals $v_0(t)$, $v_1(t)$, $v_2(t)$ and $v(t)$ are also pictured.

With this perspective it is possible to analyse signals into two distinct domains, in time or frequency. Selecting one view leads to different measurements about them [3, 8].

While signal $v(t)$ is mathematically described in equation 2.4 and it may be produced by a signal generator. It is possible to analyse it in time domain using an oscilloscope[2] and a square wave approximation will be perceived (figure 2.4). In this case is feasible to characterize the signal in terms of amplitude and period.

Some details about a signal can be only known in time domain as pulse rise or fall times, overshoot and ringing [3]. Anyhow the three present signals which describe $v(t)$ are indistinguishable in a time domain analysis, yet they are in the frequency domain.

Spectrum analysers take place when it is necessary to analyse frequency content. These tools allow to analyse the signal composition and how much power exists over specific frequencies. More than that it turns to be a better approach to understand harmonic content or analyse the occupied bandwidth for a communication channel.

As an example of a frequency domain measurement, figure 2.5 represents a spectrum analyser view of a two tone signal[3].

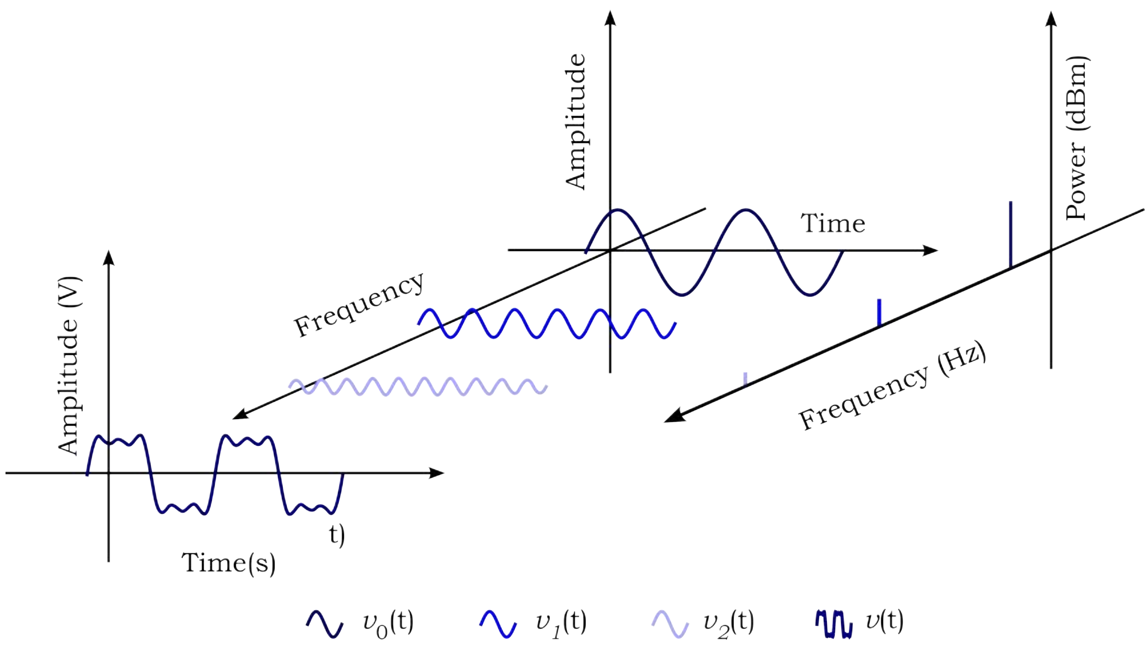


Figure 2.4: Frequency domain and time domain measurements

[2] An oscilloscope is a device used to perform electronic measurements. It allows to perceive voltage variation over time.

[3] A two tone signal is a test signal used in telecommunications. The main characteristics of particular signal, are two distinct spectral components. More about this signal will be detailed in later sections.

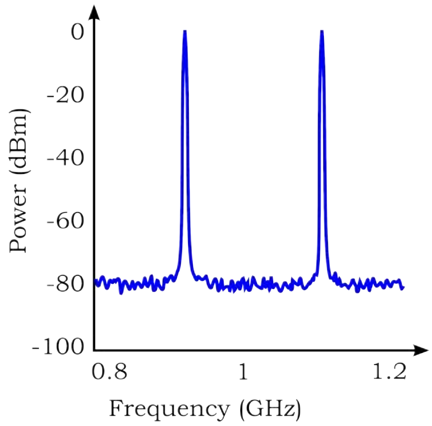


Figure 2.5: Frequency domain analysis of a two tone signal

2.4 Spectrum analyser theory of operation

To perform spectral analysis there are few distinct techniques. The essential function of these devices is to display power over frequency, though additional measurements can be performed pursuant the used method.

A spectrum analysis may be done under a swept-tune method or as a result of math calculations after the signal being acquired in time. Both provide the display of amplitude versus frequency.

Swept-tune methods interpret the power over frequency under successive tuning scans, while the other approaches use Fast Fourier Transform (FFT) techniques after an Analog to Digital Converter (ADC) had sampled in time the signal provided by a radio front-end.

For the new requirements of RF transmission techniques, where it can be mentioned complex digital modulations or spread spectrum techniques, spectrum analysis is commended to state of the art analysers where they can be commonly find with the name of signal analyser. These are the modern measurement equipments that can provide a more exhaustive signal analysis [9, 10].

2.4.1 Radio receiver

Before further details, any spectrum analyser has a RF receiver front-end. This hardware piece is essential for conditioning the electromagnetic signals and select the desired frequency band to process [11, 12, 13].

A RF front-end may be defined as anything between the antenna and the Intermediate Frequency (IF) stage [14]. It can be represent as plainly as figure 2.6.

The collected signals by the antenna are delivered to a variable attenuator to prevent high power levels to reach more sensible components. A Band Pass Filter (BPF) or a Low Pass Filter (LPF) selects the RF band, delivering the signal band to the respective interpretation process.

This simplicity might be not so efficient when treating RF or microwave frequency signals or for low power levels. Signal detection may be impractical even for sophisticated ADCs.

Turning a RF front-end as simple as possible is still object of study and research, especially for the new radio generation, the Software Defined Radio (SDR) [10].

Radio front-ends are also projected to provide frequency translation to a lower frequency level, easy to process. Special amplifiers with low noise characteristics are conjugated to improve powerless signals detection.

Concerning about interpret power over frequency, swept-tune methods have given proofs of functionality over the years and are widely used in spectrum analysers and many other wireless devices. Alongside there are the digital denominated spectrum analysers providing flexibility following up RF research needs.

A description of both methods will be presented next.

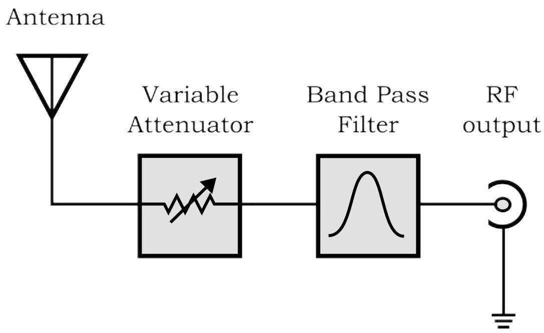


Figure 2.6: RF front-end receiver

2.4.2 Swept-tune method

Swept-tune spectrum analysers uses super-heterodyne radio architecture to sweep the input frequencies and display the energy present at each tune sweep step [13]. Super-heterodyne is a RF receiver architecture characterized by frequency conversion and its tuning ability. Figure 2.7 describes a generic super-heterodyne architecture.

A Low Noise Amplifier (LNA) receives the incoming RF signal amplifying the weak signals in the frequency band of interest to higher levels.

The mixer in the circuit is responsible to convert RF frequency to a IF one, with a controlled Local Oscillator (LO). This brings to the receivers tuning ability once varying the LO frequency, the output of the mixer will be directly related to the desired input frequency.

That conversion is intended to a fixed frequency for which the BPF is centred tuned. This rejects other conversion terms off the mixer selecting just the wanted channel. This is also known as IF filtering. The signal is again given into an amplifier to recover from previous attenuations and is ready to be interpreted.

Earliest spectrum analysers at the output of the IF, were plugged to a respective conditioning circuit to display amplitude over frequency in a Cathode Ray Tube (CRT) screen [3, 4, 13]. Figure 2.8 illustrates that IF interpretation section for swept-tune analysers.

A ramp generator guides the LO sweeping frequency as the same time promotes the electron beam deflection in the horizontal axis. The amplitude from the IF frequency caught in the envelope detector deflects the vertical axis. As result amplitude is displayed over to the frequency.

Whit the aid of digital era, microprocessors and other digital devices, turns possible other ways to interpret the IF frequency. This thesis project will have focus in another alternatives to those analogue components and these will be discussed later.

A relevant disadvantage relatively to other spectral analysis methods is the inability to measure the signal phase. Also this method may miss some quick events as pulses which skips to detection while LO is sweeping the input.

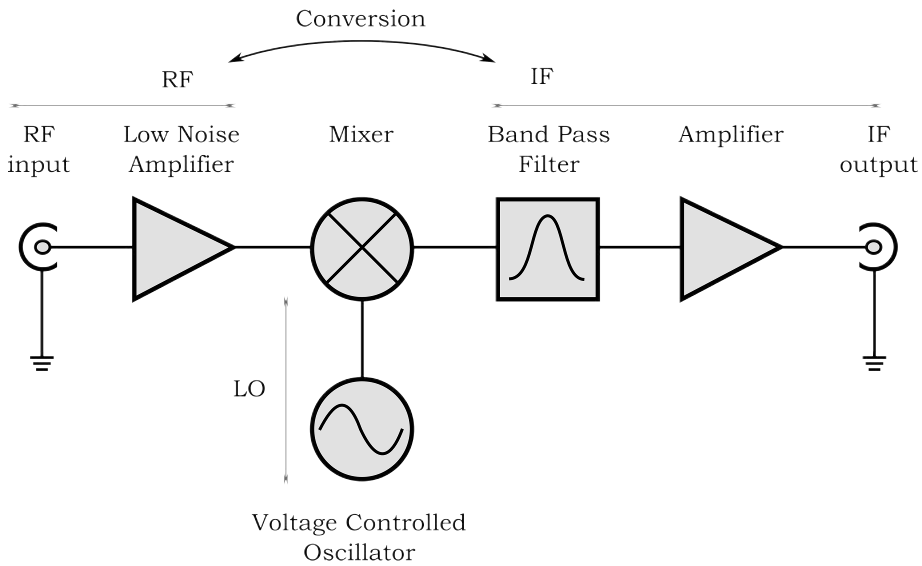


Figure 2.7: Super-heterodyne architecture

2.4.3 Fast Fourier Transform topology

This kind of spectrum analysers are also known as digital spectrum analysers in virtue to the calculation spectrum process. Spectrum calculations are sustained by Fourier transforms which involves serious mathematical theory. The most notable algorithm that converts a series of time domain samples into their spectral components is known as Fast Fourier Transform (FFT) which derives from the Discrete Fourier Transform (DFT) [15, 16, 17].

Figure 2.9 shows a block diagram with the signal acquisition principle of operation. The signal is collected by an ADC which samples and quantizes it in time domain. Furthermore powerful computing capability is required and devices as Digital Signal Processors (DSPs) or Field Programmable Gate Arrays (FPGAs) are often used.

Down-conversion, filtering and amplitude detection may be digitally computed. These operations are implemented with FFTs tools, converting the collected signal data to frequency components [4, 13, 18].

With this approach it is possible to retrieve phase information which brings a deeper knowledge about the signal. Though it has major limitations in the input frequency bandwidth which is limited to the ADC sampling rate. High sampling speeds can be obtained with less resolution bits, but this will restrict resolution, accuracy and dynamic range [19].

Vector signal and real-time analysers

More demanding signal analysis is required to detect modulation parameters and transient events. Engineers had to improve and adapt the analysis techniques in order to the dynamic RF panorama. Both Vector Signal Analyser (VSA) and Real Time Spectrum Analyser (RTSA) came to satisfy the new needs. Furthermore these models use Fourier math tools alongside with sweeping techniques and other technologies working all together to enhance spectrum analyse.

VSAs are able to perform measurements over modulation parameters. Also can display power versus time and spectrograms[4]. Specifications and features as the different perceived modulation types change from model to model. These exemplars are not indicated to perform real-time analysis once the signal needs to be in memory before calculations and the signal sampling process may not be done constantly [13, 19].

In RTSA models there is a real-time processing stage before signal post processing, granting the detection of quick changes in the input signal. Digital phosphor screens are used to implement functionalities as trace persistence and proportionality which are not natural features to digitally controlled Liquid Crystal Displays (LCDs) [20].

Measurements which requires to be digital processed as demodulation, are post executed and signal acquisition can be continuous. This conceives a spectral analysis coherent with the variations of the input signal.

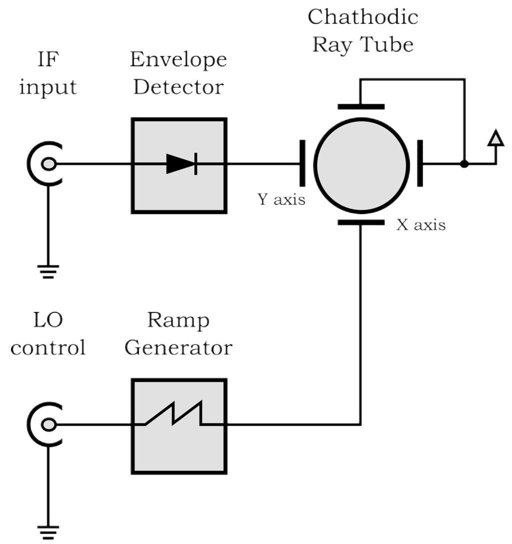


Figure 2.8: Swept-tune principle applied to a CRT screen [3]

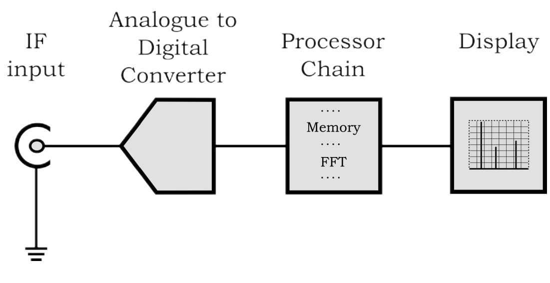


Figure 2.9: Simplified FFT topology [4, 13]

[4]A spectrogram is a color graphic representation of a spectrum. It displays frequency variation over time while color intensity changes to represent power.

2.5 Spectrum analyser characteristics

This section explains the main characteristics of a spectrum analyser. These may be designated as figures of merit and are generally known as the device specifications. They are straight related with physical circuit characteristics and limitations. This is what distinguish one equipment from another and serve as guide selection to the user needs.

Any spectrum analyser has specifications as frequency range, accuracy, resolution, sensitivity, dynamic range and distortion parameters [3, 8, 10, 21].

2.5.1 Frequency range

This specification tells the minimum and maximum frequencies which the device is able to analyse. The difference of maximum and minimum tells the bandwidth of operation. It is also called as the span, and the term *full span* is used when it is performed a analysis over the full range.

This characteristic is dictated by the RF front-end, where the frequency responses of the antenna, LPF, LNA, the mixer and the LO ranges all combined together define the frequency range.

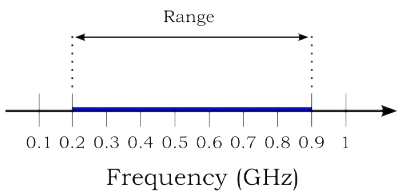


Figure 2.10: Frequency range example of 0.7 GHz

2.5.2 Accuracy

In a measurement context this figure of merit represents how close is the performed measurement to the true present value [22]. A device tend to be more accurate as closer to the original value it can read the input data.

Spectrum analysers specify different types of accuracy and they are listed as frequency, amplitude and phase. Figure 2.11 illustrates the lack of accuracy of a amplitude over frequency a measurement.

Note: It came in context to distinguish accuracy from precision. The last one is related to the capability to reproduce the same readout for the same input value.

Frequency accuracy

This figure of merit mainly relies on the accuracy of the local oscillator. Frequency references are subject of fluctuations due to temperature variations. For this reason manufacturers tend to implement internal frequency references with Temperature Compensated Crystal Oscillator (TCXO) and Oven Controlled Crystal Oscillator (OCXO) in spectrum analysers [4].

Another determining factor for frequency accuracy is the filter bandwidth accuracy.

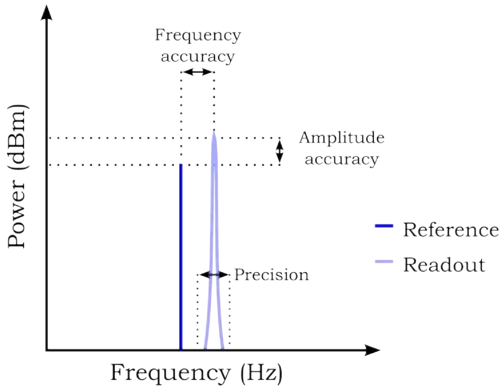


Figure 2.11: Accuracy of amplitude over frequency measurement

Amplitude accuracy

To correct identify the power level at the input of the spectrum analyser it is need to take in account the accuracy of all components in the circuit. All the losses and gains need to be accounted for subsequently correct the readout value.

2.5.3 Resolution

Frequency RBW[5] describes the spectrum analyser competence to distinguish adjacent frequency components. When at least two spectral elements are less spaced than the frequency resolution, they will be indistinguishable in the screen. Generally manufacturers provide different resolution bandwidths to satisfy distinctive needs.

Figure 2.12 shows a filter shape responsible for the RBW. Inside the BPF bandwidth exists two signals that wont be solved on the display. Narrower bandwidth filters will permit to distinguish closer signals.

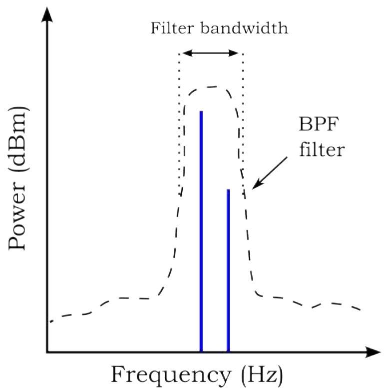


Figure 2.12: Two indistinguishable input signals

[5]This is also known as *Selectivity* in radio receivers[23].

2.5.4 Sensitivity

Any radio receiver has a noise floor level and therefore any signal with an inferior power level will be unrecognisable. Sensitivity informs the Minimum Detectable Signal (MDS) above the Average Displayed Noise Level (ADNL).

Noise floor is originated by thermal noise, inner components noise and is proportional to the bandwidth. Inspect wider bandwidths increasing RBW commit sensitivity.

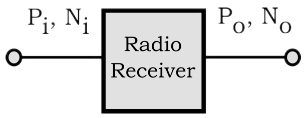


Figure 2.13: Input to output noise increase

Considering signal power P_i and noise N_i at input terminal. The output noise level is related in equation 2.5. This establish the receiver noise floor in Watts.

$$N_0 = k \cdot T_0 \cdot F \cdot B \tag{2.5}$$

Where k is the Boltzman constant($k = 1.38 \times 10^{-23} J/K$), T_0 denotes the temperature in kelvin (290 K as room temperature), NF is the noise figure and all th is related to the operating bandwidth B in Hz.

Expressing equation 2.5 in dBm it comes:

$$N_0(dBm) = -174(dBm/Hz) + NF(dB) + 10log_{10}(B) \tag{2.6}$$

A detectable signal needs to be greater than that. Combining the MDS with a minimum desired Signal to Noise Ratio (SNR) bring the figure of merit sensitivity[24]. It is usual to consider a minimum 3 dB stronger than noise floor [25].

$$S_i = -174 + NF(db) + 10log_{10}(B) + SNR_{min} \tag{2.7}$$

One characteristic of the MDS is the SNR which have to be high enough. This is depicted in figure 2.14.

2.5.5 Dynamic range

The dynamic range specifies the spectrum analyser capability to handle with weak and strong signals at the same time. It consists on a ratio and is expressed in dB. Figure 2.14 graphically illustrates the dynamic range relation in a measurement.

Maximum detectable signal is limited to components saturation levels while minimum signal detection is limited by the instrument noise floor and contrived signals. The last quoted problem are the second and third order intermodulation distortion products which may affect dynamic range [23].

2.5.6 Distortion

Spectrum analysers may be wanted to realize distortion measurements. For this reason it is necessary to prior now inner distortion levels. This is related to the internal spectrum

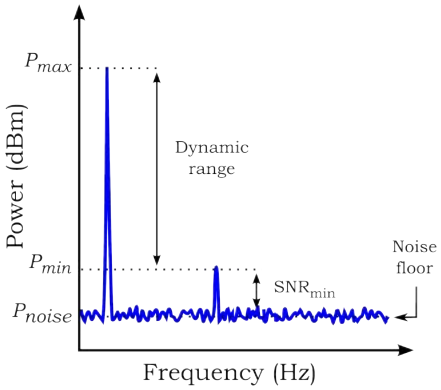


Figure 2.14: Sensitivity and dynamic range depiction

analyser components distortion. It needs to be take in account once measuring others devices distortion.

Distortion is an undesired but unavoidable effect of the non ideal components functionality. Moreover Intermodulation Distortion (IMD) is of great importance and will be detailed later together Third-order Intercept Point (IP3).

2.6 RF design basic concepts

In a RF design project there are useful concepts which helps understanding circuit blocks characteristics while performing. This notions are important to the component evaluation process. Some of them may have impact in the circuit functionality leading to respective spectrum analyser specifications. Special attention to RF electronic parts is demanding once high operating frequencies require some specific evaluation methods to ensure maximum power transfer.

Decibels, decibels to milliwatt and S-parameters are used to characterize a input-output device related to the enclosed power levels, are described in appendix A and B for whom are not familiarized with it.

This section provides an explanation about VSWR, 1 dB compression point, noise figure and intermodulation effects.

2.6.1 Voltage Standing Wave Ratio

In RF circuits, input and output VSWR tells how well the power transfers occurs for a frequency range with specific impedance value. It provides a way to analyse the impact of reflections in signal transmission, telling the ratio of the maximum to minimum values that voltage and current can ever get [26, 27].

When two waves with the same frequency travelling in same medium but in opposite directions they sum each other resulting into a composite standing wave[6] [26, 28, 29, 30].

In electric circuits either bad junctions or impedance variations over the network are responsible for signal reflections. Connecting one component to another results in some of the incident wave to be reflected. This leads to the two waves propagating in opposite directions.

[6] A pure standing wave forms only when the incident and reflected wave have the same amplitude and frequency[31].

As they sum each other occurs maximum constructive peaks and a minimums when both waves are in phase or opposition respectively (figure 2.15).

It is mathematically expressed in in equation 2.8, in order to the reflection coefficient provided by S_{11}.

$$VSWR = \frac{1 + |S_{11}|}{1 - |S_{11}|} \tag{2.8}$$

An ideal VSWR of 1 would mean that all energy is being transmitted as result of no reflections $S_{11} = 0$. This situation could be exemplified as an perfect impedance match. In case of a full signal reflection $S_{11} = 1$, would result on an infinite VSWR.

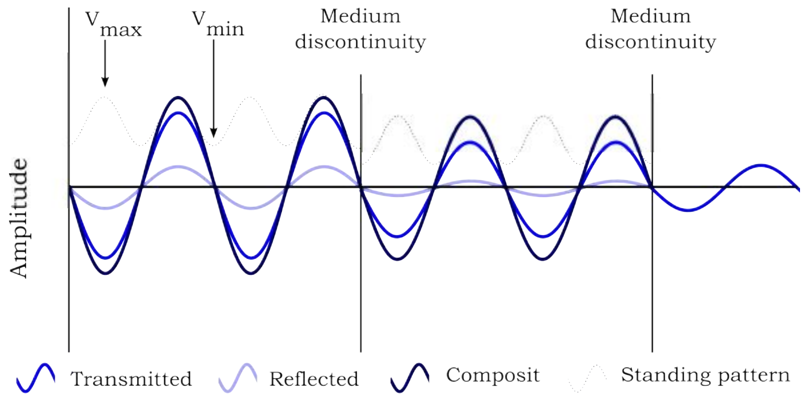


Figure 2.15: Voltage Standing Wave Ratio

2.6.2 Gain compression

Saturation effect in RF circuits is commonly designated as 1 dB Compression Point (P1dB). While systems are designed to operate in a linear region, they may change the desired behaviour under certain circumstances.

For example, an amplifier output power will increase 1 dB for each dB increment in the input signal. The device will operate in this linear region till the input power reach such high value where the output power stops to increase in the same 1 dB proportion as it did before.

Figure 2.16 illustrates an amplifier's 10 dB gain curve. The linear behaviour starts do change when input power reaches almost 10 dBm. In this example, when input power increases to 11 dBm the output remains on 20 dBm not keeping the 1 dB gain proportion.

This depicts the P1dB which is defined as the output power level where the the gain curve is 1 dB compressed in relation to the linear region. In this example the output power level for P1dB is 20 dBm.

Besides amplifiers, another component type that can be defined in terms of compression is the mixer which in contrast with amplifiers, P1dB tend to be defined as the maximum input level for each mixer starts to produce a non linear output curve.

2.6.3 Noise figure

As a figure of merit Noise Figure (NF) is a measurement to characterize the whole RF circuit and individual components with at least two ports as an input and output. Knowing

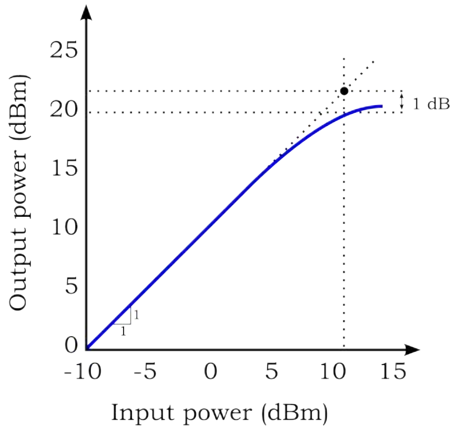


Figure 2.16: 1 dB compression point illustration

NF from individual circuits allows to calculate receptor sensitivity [32]. This figure of merit quantify the SNR degradation from the input to the output in a electric device.

Any component subjected to a temperature above zero kelvin superimpose inner noise as for example, the thermal noise in a resistor. That noise is added to a signal passing through the component with no added power gain to the signal[7].

In figure 2.17 is represented the input and output spectrum at respective amplifier's ports. In 2.17a it is perceptible a ground noise of -80 dBm and a major frequency component at 1 GHz with -20 dBm. Figure 2.17b depicts the output with the 1 GHz component pushed to 0 dBm and the noise floor raised to -30 dBm.

It is noticeable a 20 dB gain at 1 GHz although the ground noise level has been amplified 30 dB. This noise floor increase is related to the intrinsic device noise, being represented as the noise figure and in this example $NF = 10dB$. As already mentioned it represents a degeneracy in SNR [7, 32], and is mathematically expressed as in equation 2.9. The term NF is the representation in dB of the last relation which is known as noise factor (F).

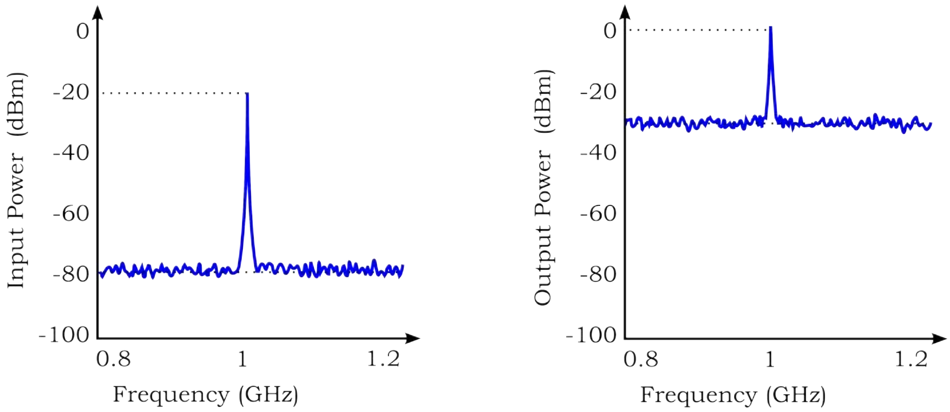
$$F = \frac{SNR_{input}}{SNR_{output}} \tag{2.9}$$

$$NF = 10log(F) \tag{2.10}$$

$$F_t = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + ... + \frac{F_n - 1}{G_1 G_2 ... Gn - 1} \tag{2.11}$$

As a RF receiver consists in a few cascaded devices, noise figure allows to to calculate the overall noise figure. Following equation 2.11 considers a series of components with noise factor F_n and respective numerical gains G_n which calculation method can be find in [33]. As this is intended to calculate the total system NF to find out sensitivity of the receiver, the first term is the most important [1, 10, 31]. Following terms are successively smaller due to the denominator increasing factor contributing less for the overall NF_t.

[7] In a noiseless system there are no degradation in SNR. Signal power and intrinsic noise are attenuated or amplified for the same factor [7]



(a) Input spectrum

(b) Output spectrum

Figure 2.17: Noise figure example [32]

2.6.4 Third order intercept point

This figure of merit, Third-order Intercept Point remarks relevant non linear characteristics in RF circuits. Something important to mention about non linear characteristics for a two port device is the creation of new frequency components at the output that are not present in input signal [1]. In addition any component has non linear characteristics even though they are designed to operate under a linear region. Special attention is taken into amplifiers and mixers.

Two tone test is a common way to measure some of the non linear characteristics. It allows to measure the system bandwidth impact[1].

This test consists in two different frequency components with the same amplitude used as input signal into the Device Under Test (DUT). It is expressed in equation 2.12.

$$v_i(t) = Acos(2\pi f_1 t) + Acos(2\pi f_2 t) \tag{2.12}$$

$$v_o(t) = a_1 v_i(t) + a_2 v_i^2(t) + a_3 v_i^3(t) \tag{2.13}$$

The output result may be represented mathematically in a power series as in equation 2.13. The two tone signal test is applied and at the output will appear a series of frequency components. The most important new components are represented in figure 2.18 and detailed in table 2.1 [1, 10].

One of the major problems of this kind of distortion is the inability to remove the new frequency components that fall into bandwidth of interest, very close to f_1 and f_2. In this case a BPF permit to attenuate the higher order harmonic components, but barely affects $2f_1 - f_2$ neither $2f_2 - f_1$.

In figure 2.20a it shown the output power relation of the fundamental frequency f_1 applied to the DUT and a third order component $2f_1 - f_2$ demonstrating the IP3 point. In opposition to a 1 dB/dB slope of the fundamental frequency, the third order component raises with a 3 dB/dB slope. Theoretically if a straight line is plotted long side to the gain curves, they will hypothetically intersect[8] That intersection point is known as IP3.

[8] This intersection does not occur in practice once both output curves start to compress due to higher order terms [10].

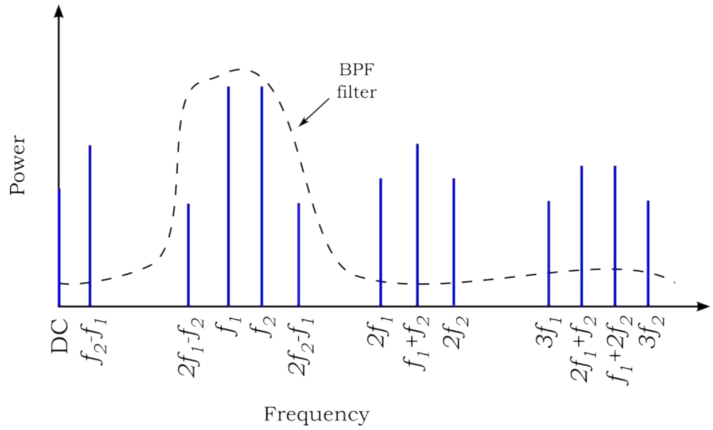


Figure 2.18: Distortion spectrum

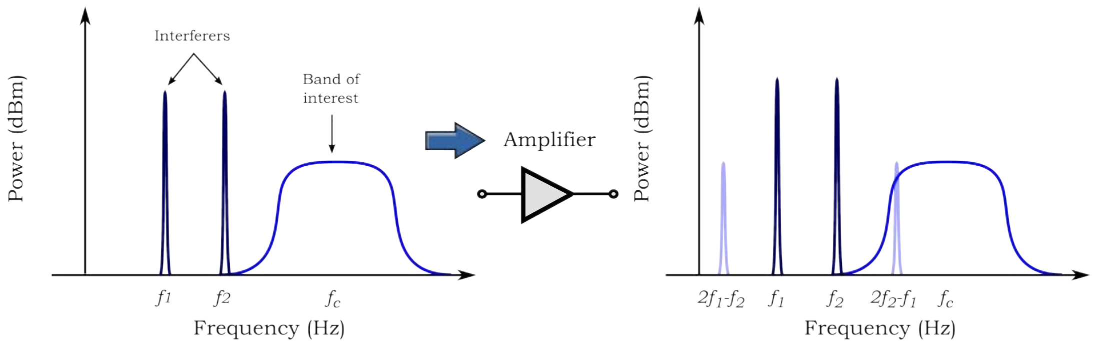


Figure 2.19: Non linear interference inside the desired band

This analysis allows to evaluate the Intermodulation Ratio (IMR) which is connected with intermodulation distortion. It describes the ratio of the fundamental output within IMD new components.

$$IMR = \frac{P_{f_1,o}}{P_{2f_1-f_2,o}} \tag{2.14}$$

$$= \frac{P_{f_2,o}}{P_{2f_2-f_1,o}} \tag{2.15}$$

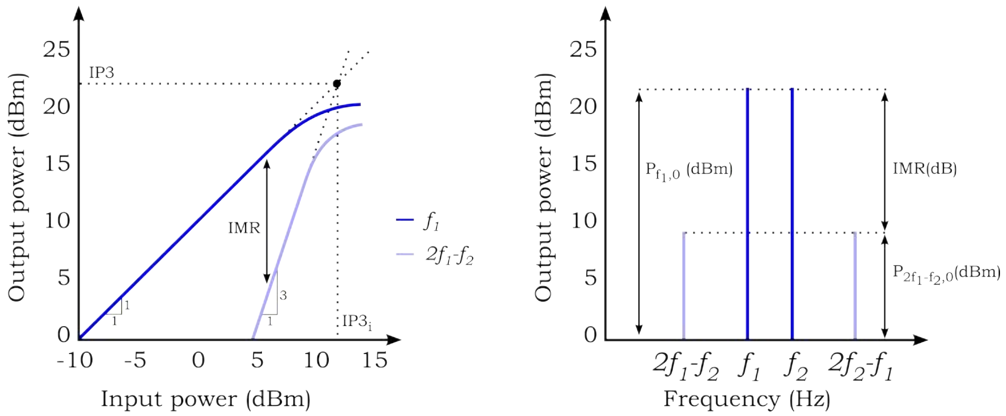
Equation 2.14 and 2.15 can be expressed in dB units as follow:

$$IMR(dB) = P_{f_1,o}(dBm) - P_{2f_1-f_2,o}(dBm) \tag{2.16}$$

$$= P_{f_2,o}(dBm) - P_{2f_2-f_1,o}(dBm) \tag{2.17}$$

Figure 2.20b represents a two tone test performed in a spectrum analyser. Using the spectrum analyser measurements it is possible to calculate IP3 with the following expression:

$$IP3(dBm) = P_{f_1,o}(dBm) + \frac{IMR(dB)}{2} \tag{2.18}$$



(a) Third order intercept point (b) IMR measure in a spectrum analyzer

Figure 2.20: IP3 and IMR examples

Practical meaning	❖	**Frequency components**
Component resulting from $f_1 - f_1$ and $f_2 - f_2$		Direct Current (DC)
f_1 fundamental frequency		f_1
f_2 fundamental frequency		f_2
f_1 second harmonic		$2f_1$
f_2 second harmonic		$2f_2$
f_1 third harmonic		$3f_1$
f_2 third harmonic		$3f_2$
Second order intermodulation products		$f_2 \pm f_1$
Third order intermodulation products		$2f_1 \pm f_2$
Third order intermodulation products		$2f_2 \pm f_1$

Table 2.1: Output components resulting from a two tone test

Chapter 3

Architectures

Subsequently to the introductory concepts involved with spectral analysis and circuit design, it is necessary to set the approach to implement the prototype. A spectrum analyser is mostly a radio receiver, which consists on a electric circuit. Even with the continuous development in the state of the art of Integrated Circuit (IC), aspects as cost, complexity, size number of external components and power dissipation are taken in preponderation to the final prototype solution[7].

The circuit control may be performed with digital technology. This brings advantages to the development stage due to flexibility to circuit control and human to interface.

Previously in subsection 2.4.2 was introduced one of the most commonly known spectrum analysers architectures. That was necessary to describe the theory of operation of such instrument. Regardless there are other architectures to extract to analysis, high frequency electromagnetic signals and a quick overview will be driven.

Therefore in this chapter will be detailed the chosen architecture for the radio circuit and the impact of the components in the overall system. Alongside is described the digital interface to control circuit under human direction .

3.1 Receiver architectures

A radio architecture can be understood as the design rules to organize the electronic components in order to perform a logical set of functions [34]. This piece of hardware is intended to collect the electromagnetic signals, reject the unwanted frequency bands, amplify small signals and all of this with minimum of distortion and added noise. As result the received electric signal will be conditioned and primed for interpretation, which usually includes at last human understanding.

A very important functionality in a receiver is the frequency down conversion. This takes place on the mixer which main function is to multiply the incoming RF signal with the LO controlled frequency. This creates two new frequency components at the output of the mixer, the sum and difference [23]. As result in the IF stage will be f_{IF} which is mathematically expressed as $f_{IF} = f_{RF} \pm f_{LO}$.

After the incoming signal passes through the mixer, the modulation present at f_{RF} is translated to the new f_{IF} components. This brings the *problem of image* and different architectures deal with it in a proper way [7].

3.1.1 Homodyne receivers

This architecture likewise known as *direct conversion* or *zero-IF*, is conceived to translate the input frequency of interest to a DC center. The frequency of the local oscillator matches the frequency of interest, $f_{LO} = f_{RF}$. Such method requires a high precision LO reference to stay in tune with the pretended input. As result of the intended down conversion it is possible to predict in the IF stage two spectral components, $f_{IF_0} = 0$ and $f_{IF_1} = 2f_{RF}$.

It is affordable to reject the unwanted component $f_{IF} = 2f_{RF}$ with a LPF for this situation. Figure 3.2 shows the two frequency components presents at the output of the mixer.

Some modulation techniques may not operate with this architecture[1]. A few modifications are available in the block diagram of figure 3.1 to solve this issue, though in a basic spectrum analyser there is no intention to interpret the modulated data.

More to say about some issues that occur in this architecture as DC offsets and power leakage from RF to LO and vice versa. This can be troublesome for signals integrity [7, 10, 23]. Direct conversion has wide use for single tuned frequency channels. The simplicity of the architecture avoid image problems, and provides effectiveness extracting audio signals straight forward from f_{IF} [23].

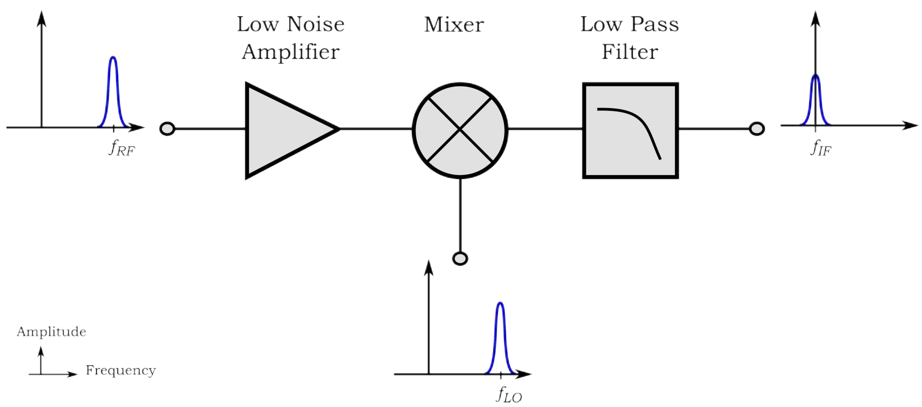


Figure 3.1: Simplified Homodyne receiver

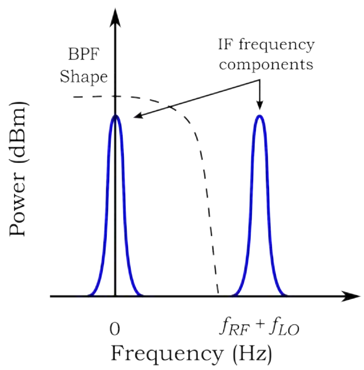


Figure 3.2: Image frequency rejected with a LPF

[1]Signal modulation like FM or Quadrature Phase-Shift Keying (QPSK) convey information that would be lost with this simplified process. [7].

3.1.2 Heterodyne receivers

In this architecture[2] the LO frequency does not equate the desired input frequency band. The new frequency conversion is intended for for other components that are not DC centred. Again, at the output of the mixer there are $f_{IF} = f_{RF} - f_{LO}$ and $f_{IF} = f_{RF} + f_{LO}$. Providing a fairly higher centred f_{IF} than in the homodyne architecture, allows to escape from obstacles previously described, although problem of image demands for solutions.

Choosing IF center frequency and LO tuning frequency takes some aspects in consideration. Incoming frequencies centred at $f_{RF} + 2f_{IF}$ will fall inside the IF filtering stage being denominated as image frequencies. Figure 3.3 illustrates the LPF output spectrum. The two incoming frequency components f_{RF} and f_{img} equally distant from f_{LO} fall inside the filter's pass band and that may invalidate the communication channel.

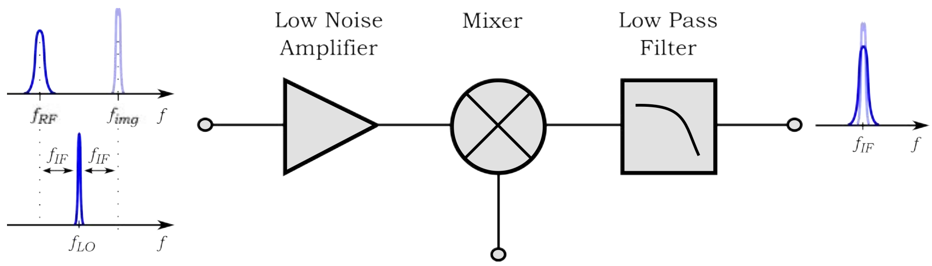


Figure 3.3: Image frequency problem in a heterodyne receiver

Image rejection and channel selection are conceivable with filters. Filtering in the RF stage removes the undesired images to reach the mixer. Anyhow, the mixer will produce both f_{IF_0} and f_{IF_1} and other spurious. A channel selection is performed with a LPF or BPF This selects the wanted conversion term to establish communication channel.

In the architecture project is necessary to decide which high or low f_{IF} side will be used to establish the communication channel. Some trade off consequences happens as for example, the added complexity to achieve good filters centred for high frequencies causing a loss on resolution when comparing with filters projected to operate at lower frequencies.

A low f_{IF} can compromise sensitivity, once a shorter $2f_{IF}$ means image frequencies closer to the f_{RF} pretended channel, and so image reject can be harder to implement. The receiver sensitivity decreases for a low f_{IF} once higher power levels from the image can reach the mixer lowering SNR. Figure 3.5 depicts the trade off between choosing a high or low IF side, with the same image reject filter.

Select the desired channel (f_{RF}) with a nearby interferer (f_n) is hard to achieve for a high centred f_{IF}. In opposition to a low f_{IF}, where narrower bandwidth filters are more effective, allows to remove the unwanted interferer.

Both image reject and channel select filters should afford a good attenuation outside the band of interest and low loss inside the pass band to minimize signal deterioration.

There are different adaptations from the basic architecture described in 3.3, as dual IF conversion, but advantages of one over the other can be relative [35]

[2]The terms *homo*, *hetero* and *dyne* means respectively: *same*, *different* and *to mix*.

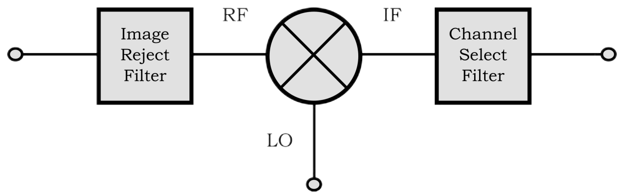


Figure 3.4: Image rejection and channel select in heterodyne receivers

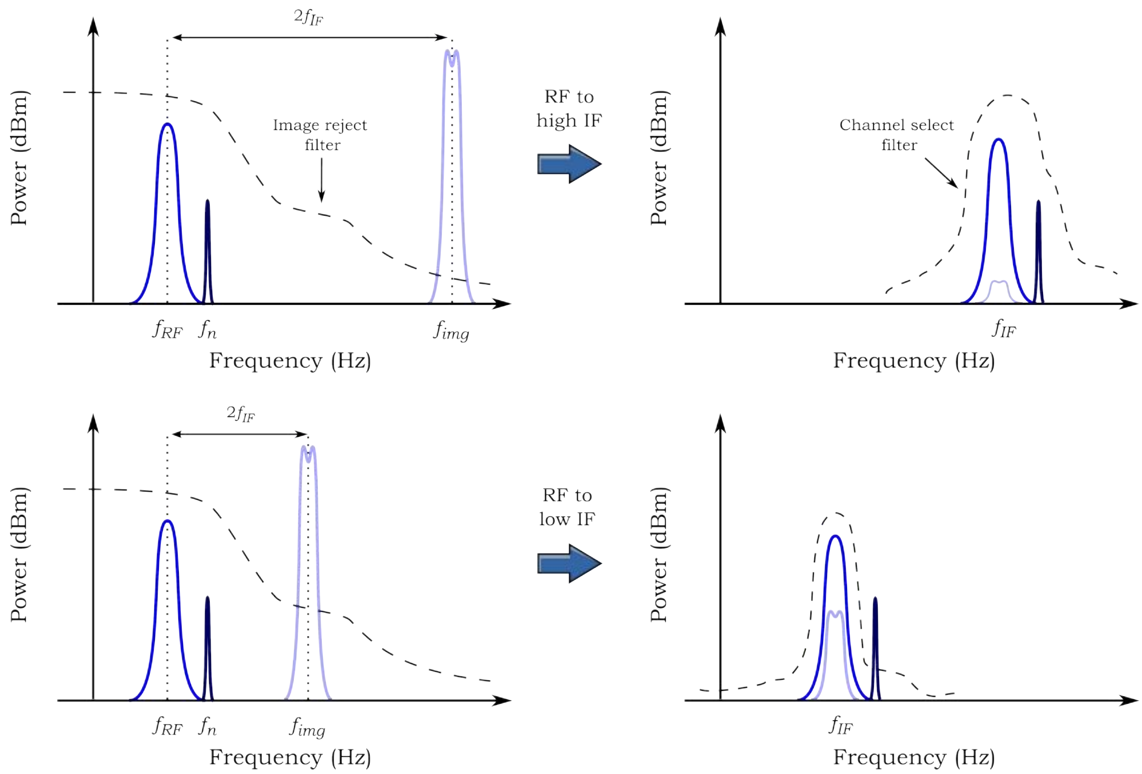


Figure 3.5: Rejection of image and channel selection differences for a high or low IF [7]

3.1.3 Super-heterodyne receivers

Super-heterodyne or heterodyne receivers are often described as only one type. There is no intention here to evidence differences although for coherency with literature it is presented in figure 2.7 a super-heterodyne block description.

In fact this will be the architecture used to implement the spectrum analyser prototype. The RF source signal used to the development stage is a controlled function generator. For that reason considerations about signal acquisition with an antenna, variable attenuator to protect sensible parts as the mixer, and image reject filtering will not be take in account. As so, the block description is present in diagram 3.6.

It will be used a LNA to increase overall sensitivity. A filter bank at the output of the mixer will provide variable RBW. A single high gain amplifier is placed at the end of the IF stage recovering the signal power to levels where a RF power detector can sense it.

This architecture provides incoming RF conversion and with a passive fixed tune BPF it is possible to select from the output of the mixer a specific incoming frequency. An important characteristic to mention is the tuning ability once LO oscillating frequency will be successively scanning the input range. This matches the needs of a spectrum analyser design, once it makes frequency sweeps to measure the existing power in each sweep step. A microcontroller is used as the control center holding the human interface while reading input user instructions and keeps the Voltage Controlled Oscillator (VCO) sweeping in frequency the input range. For each tune step it happens a DC voltage readout at the output of the power detector. The digital microcontroller will also drive a graphic LCD, under a swept tune method to perform power analysis over frequency.

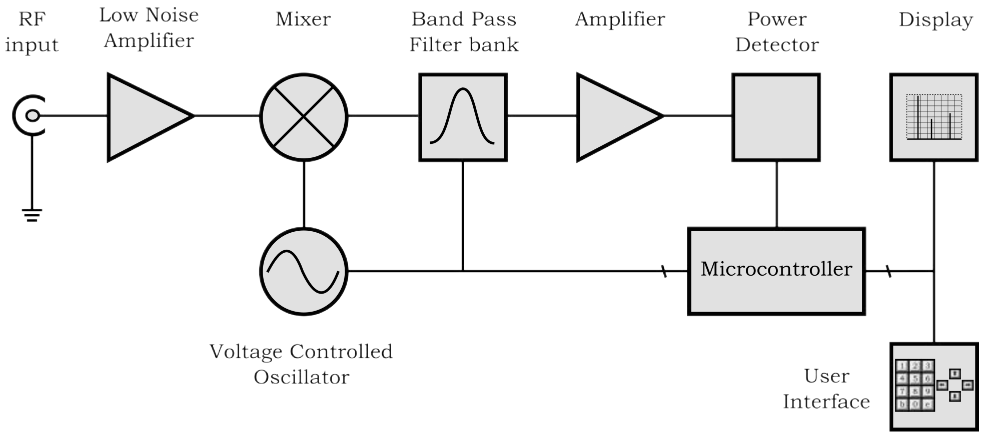


Figure 3.6: Spectrum analyser prototype block diagram

Recalling the theory of operation for a spectrum analyser and with intuit to understand the logical set of functions instructed by the chosen architecture, figure 3.7 illustrates what is spectrally happening in a swept tune analysis. It represents the input RF spectrum and the LO sweep range to perform down conversion. The signal at the output of the mixer passes through an IF filtering stage and the respective output spectrum is also represented.

Considering the mixer conversion loss of 10 dB and another 10 dB of attenuation inside the filter pass band. In total the signal channel is attenuated 20 dB from the RF input to the IF output. Outside of the filter passband attenuation is roughly 60 dB, allowing the receiver to be insensitive to nearby interferers.

For a BPF with a center frequency of $fc = 70MHz$, while pretending to scan the input range from 880 to 950 MHz the LO is set to perform a *low side injection* as consequence to the selected $f_{IF} = f_{RF} - f_{LO} = 70MHz$. This implies the LO oscillating frequency to keep the relation $f_{RF} - f_{LO} = 70MHz$. Evoking the problem of image, it should be mentioned that frequencies higher than 950 MHz should not reach the mixer with penalty of image frequency fall inside the passband.

Three snapshots were take to the output spectrum of the BPF while the LO sweeps the RF input range. The three moments take place when $f_{LO} = \{810, 840, 870\}\, MHz$ being each distinguished with the graph color.

Therefore is detected a power increase inside the 5 MHz pass band only for $f_{LO} = \{810, 840\}\, MHz$. When $f_{LO} = 870MHz$ there is no power over $f_{RF} = 940MHz$ resulting in a no power variation at the output of the filter.

This sweeping process is continuously done and kept by the microcontroller. Decisions on the sweep time can be adjusted as frequency range, controlling the LO tuning range.

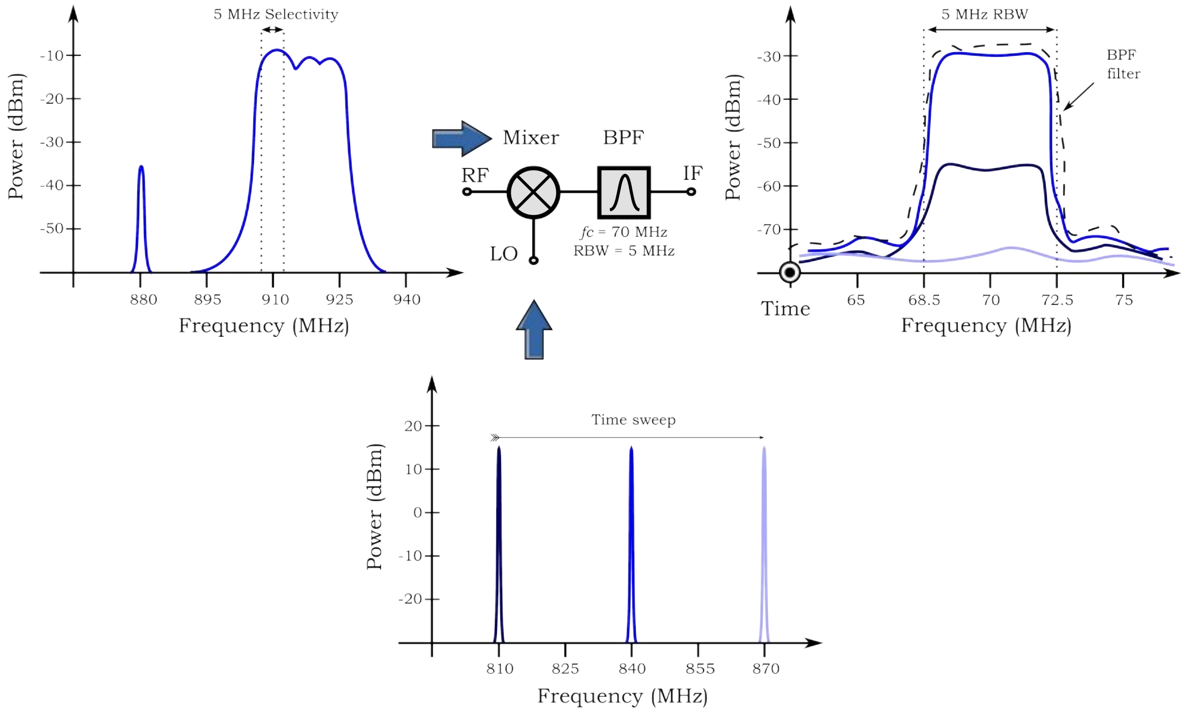


Figure 3.7: LO sweep

3.2 Components functionality

As result of the architectures description all the functions are implemented with RF electronic components. Reduced sizes and high integration technology brings to market many different IC styles as differences in performance specifications. Even though general behaviour of RF components can be described. To ensure the correct service of each pieces it is need to select them with a closer look to match design specifications. Concerns about physical dimension and package style will be take in next chapter.

In general, RF components acts as blocks functions with an input and a output. These blocks may be mentioned as networks and allows RF design projects to be schematized as it as been so far.

Here it will be described the performed function by each block and the effect they have on circuit.

3.2.1 Amplifiers

RF designed amplifiers act as gain blocks. They provide power amplification to the input signal [36]. There are distinct ways to express gain. The gain of an amplifier can be expressed in dB as follow equation 3.1. In this case, the gain factor G is added to the input signal power P_{in} expressing the output power as in equation 3.2. For example, if the gain block in figure 3.8 has a gain $G = 10dB$ and the input signal a power of $P_{in} = -20dBm$, the amplifier fit out $P_{out} = -10dBm$.

It is usual to find RF amplifiers already matched to be stable in 50 Ω systems. Amplifiers can be characterized by maximum power gain, noise figure, VSWR, DC power consumption and the emitted power at harmonics distortion. Operation bandwidth is also preponderant for the RF project. The importance of these figures of merit change in order to the finality which is intended the amplifier.

Another important figure of merit is the Power Added Efficiency (PAE). This parameter demonstrate the efficiency of DC to RF power conversion. How smaller is the result of equation 3.3, higher is the DC power consumption. This implies with batteries life time and in case of base stations turns into an overeating of the device [36].

$$G = 10\log\left(\frac{P_{out}}{P_{in}}\right) \tag{3.1}$$

$$P_{out} = P_{in} + G \tag{3.2}$$

$$PAE = \frac{P_{out} - P_{in}}{P_{DC}} \tag{3.3}$$

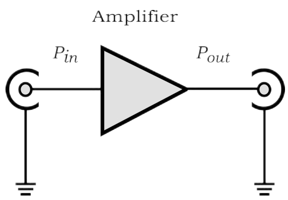


Figure 3.8: Gain block

Low noise amplifier

As the name implies, this is an amplifier with a low noise figure. These devices are usually used in the front of thed RF receiver chain determining the lowest signal possible to examine. This block in the spectrum analyser design is responsible for one of the figures of merit, sensibility.

Power amplifiers

General propose amplifiers, with higher power gain are used in the rest of the spectrum analyser schematic, but with the disadvantage of higher noise figure. This has no major consequences because the SNR levels are all ready high enough, and does not affect the power signal detection.

3.2.2 Mixer

Mixers are non linear designed devices that provide frequency translation. They can be active or passive and mixer's ports are described in block diagram 3.9. Theoretically signals phase and amplitude are not disturbed, making it available to work with modulated signals [36, 37, 38, 39].

The fundamental operation of this devices, the frequency conversion, is obtained as the sum or difference between the two input signal, generating new ones.

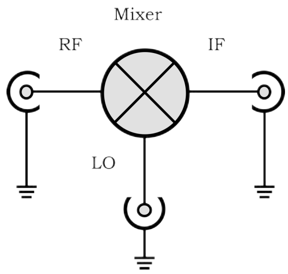


Figure 3.9: Mixer ports diagram

This device can transform RF to a lower IF frequency easy to process in receivers. This operation is usually designated as *down-conversion*. Additionally it may reciprocal convert IF as a base band signal to a higher RF frequency, for efficient wireless transmission. This method is described as *up-conversion*. For this prototype project which is a radio receiver, the mixer will be used as a *down-converter*.

The non linear mixer designed response creates a group of output signals containing multiplies of the input signals, with sums and differences all together. It is important to refer on this stage the importance of properly filtering the mixer output in order to remove the unwanted signals.

Considering v_{RF}, v_{LO} and v_{IF} as the signals presents at mixer's respective terminals. The signals v_{RF} and v_{LO} can be described as:

$$v_{RF}(t) = a_{RF}(t)\cos(\omega_{RF}t) \tag{3.4}$$

$$v_{LO}(t) = a_{LO}(t)\cos(\omega_{LO}t) \tag{3.5}$$

Where ω is the angular frequency denoted as in 3.6 and $a(t)$ represents the sinusoid amplitude.

$$\omega = 2\pi f \tag{3.6}$$

Ideal the signal v_{IF} which is the mixer's output, is described as a multiplication of the other two input signals.

$$v_{IF}(t) = v_{RF}(t)v_{LO}(t) \tag{3.7}$$

From equation 3.7 it cames:

$$v_{IF}(t) = \frac{a_{RF}(t)}{2}\cos(\omega_{RF}t + \omega_{LO}t) + \frac{a_{LO}(t)}{2}\cos((\omega_{RF}t - \omega_{LO}t) \tag{3.8}$$

$$= \frac{a_{RF}(t)a_{LO}(t)}{2}(\cos((\omega_{RF} + \omega_{LO})t) + \cos((\omega_{RF} - \omega_{LO})t) \tag{3.9}$$

Equation 3.9 reveals the sum and subtraction frequency terms. These are the second order responses [37].

$$\omega_{RF} + \omega_{LO} \tag{3.10}$$

$$\omega_{RF} - \omega_{LO} \tag{3.11}$$

Considering equation 3.6 where is defined the angular frequency it is possible to express the mixer sum and difference output frequency terms as mentioned in equation 3.12

$$f_{IF} = f_{RF} \pm f_{LO} \tag{3.12}$$

An important observation about equation 3.12 is it can be expressed as well in the way around:

$$-f_{IF} = f_{LO} \pm f_{RF} \tag{3.13}$$

In real world there are no difference in positive or negative frequencies. This is often miss explained in literature and can bring severe unwanted consequences to the system project if not considered.

For example, projecting the mixer to *down-convert* f_{RF} for $f_{IF} = 70MHz$[3]. At mixer IF output port it will appear a 70 MHz signal whenever $f_{RF} - f_{LO} = 70MHz$ or $f_{LO} - f_{RF} = -70MHz$.

So in practice the output interest terms to know about mixers are expressed as:

$$|f_{RF} \pm f_{LO}| \tag{3.14}$$

Therefore the receiver bandwidth is limited to the chose of the IF frequency, RF and LO range [3]. To note that mixers have RF and LO leakages to the IF output.

Another important parameters to know about a mixer on a practical design are the VSWR, conversion loss, ports isolation, LO power requirements, IP3 and noise factor.

Over the frequency translation operated by the mixer, it happens a power loss from the RF input signal to the IF output signal. This power loss is designated as conversion loss and mathematically expressed in equation 3.15.

$$L_c = 10 \log \left(\frac{P_{RF}}{P_{IF}} \right) \tag{3.15}$$

In opposition to a device gain expressed in equation 3.1, passive mixers attenuate the signal and L_c is often expressed as a negative quantity in dB.

3.2.3 Voltage controlled oscillator

The primary function of an VCO is to produce a oscillating electric signal at a specific frequency. Furthermore the signal frequency is variable and voltage controlled. These components are widely used in super-heterodyne architectures as the LO signal source. The figures of merit of a VCO are the tuning range, respective tuning sensitivity (MHz/V), frequency stability (PPM/°C), phase noise and output power.

3.2.4 Filters

Filters allow to reject some frequency bands and select others of interest. Typically in RF projects, filters are implemented in microstrip, resonant cavities or Surface Acoustic Wave (SAW) devices. The most important characteristics on filters are the VSWR, pass-band bandwidth, insertion loss inside band and attenuation outside band.

[3] Choosing the IF frequency it straight related to the center frequency of the IF filter bank[3]

Designations as Low Pass Filter, Band Pass Filter and High Pass Filter tells the frequency response characteristic. Cut off frequencies inform where the filter attenuation level changes. Filter attenuation inside the pass band should be low as opposition to the outside band of interest where it must attenuate other frequencies.

For this prototype is considered a filter bank in the IF section. This selects the frequency component of interest from the mixer providing different Resolution Bandwidths to spectra analysis. Therefore all filters in this bank should be centred to the same frequency to allow a constant LO sweep.

Each filter with a different pass bandwidth is selected to operation by the means of two additional blocks. One of then may be a RF switch digital controlled and the other a power combiner.

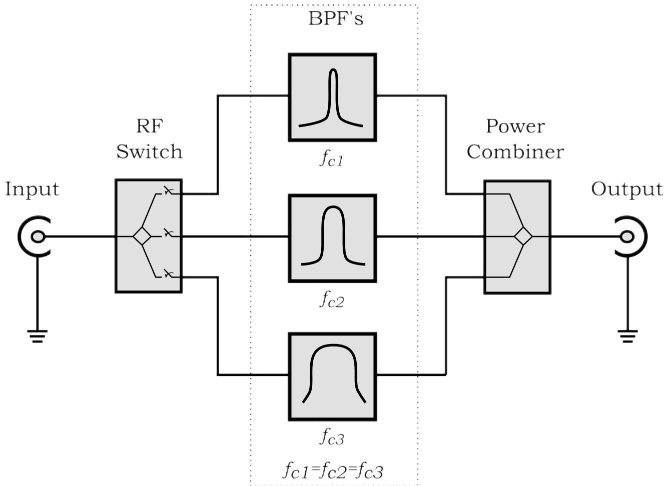


Figure 3.10: IF filter bank

3.2.5 RF switch

This device is able to route high frequency analogue signals trough different channels paths. It is an active device which may be digital controlled being suitable for automated systems. This devices are quite useful in laboratories environment where a signal can be conducted to different equipment tests.

As this project foresee an automatized digital circuit control, it turn to be an appropriate toll to be used in the filter bank, selecting the signal passage for the desired filter.

RF switches are characterized with VSWR, insertion loss, isolation between channels, P1dB and for exigent applications aspects as switching and settle times may also be detailed.

3.2.6 Power combiner

Power combiners are passive devices that can divide a single RF channel into different outputs or join together diverse transmission paths into one according to the propagation direction. It got a pre defined insertion loss for the frequency band which is designed. For the filter bank in this project is necessary to conduct the output of three filters to a single power detector, and this component turns to be serviceable, joining the signal paths.

The most important characteristics on a combiner are VSWR, insertion loss and ports isolation [36].

3.2.7 Power detector

This component is the last one in the receiver chain and it will be where the incoming RF signal is finally delivered. After the IF stage the signal's frequency will be centred at 70MHz, so the power detector needs to work in the same frequency range.

These devices are design for RF applications. In order to the input power, it sets an output DC voltage. Knowing the typical output voltage curves it is possible to measure the RF power in dBm.

3.2.8 Digital control - Microcontrollers

Behind the super-heterodyne architecture theory of operation it is need to control some variables in the RF system, to validate measurements and turn them available for users. By other means, translate the electric signals to human comprehension domain. For electronic projects, microcontrollers may be a very attractive option with reduced sizes, affordable prices and ease to use interfaces.

Microcontrollers likewise computers on chips have a processor unit provided with a variety of embedded peripherals such as non volatile flash memories, Electrically Erasable Programmable Read Only Memory (EEPROM), volatile memories as Random Access Memory (RAM) Static Random Access Memory (SRAM), input and output communication devices as Universal Asynchronous Receiver Transmitter (UART), Universal Serial Bus (USB) or even Ethernet. Alongside with ADCs, Input Output (IO) digital pins and Digital to Analogue Converters (DACs) these devices are able to sense and interact with external components and handle automated processes.

The programmed tasks can be done with a high-level abstraction[4] allowing to the code development with a programming language as C, avoiding the intrinsic native languages of each manufacturer.

In the architecture variables that require control are the electric DC voltage applied on the VCO, the channel selection in the IF filter bank and the incoming power interpretation. With an electric visual screen it is possible to bring electromagnetic graphics representations to sight.

[4]High-level programming languages provides abstraction to the minutiae details of the processor hardware while low-level languages turn to be keen in those specifications

Chapter 4

Implementation

In this chapter 4 are described the selected components, evaluation boards layouts, pictures of the used circuits and practical individual tests to accomplish the physic radio receiver. Next chapter discus tests of the entire electric circuit as a spectrum analyser. and respective evaluation process

Components were selected for each function block in diagram 3.6, under specific criteria designated as figures of merit. Therefore characteristics of the final spectrum analyser are straight related with the taken options for the individual RF blocks.

Each component will be take as an individual block to support the prototype implementation. Dividing the hardware into distinct evaluation boards helps to test and debug individually each device rather than assembling the whole circuit in only one Printed Circuit Board (PCB).

Following an initial set of specifications for the prototype as input frequency range from 880 MHz to 950 MHz with down conversion to a fixed IF of 70 MHz, three selectable resolution bandwidths and single power supply, turned to be the guidelines to select components.

Another requisite is the input impedance matching at 50 Ω. This minimizes external matching circuits. Also the DC power consumption and package style is taken into account.

The physic package dimensions need to be large enough to allow manual solder. It is common to find components with such small size that require solder techniques[1] that are not available at Instituto de Telecomunicações (IT) laboratories where the assembling process is executed.

To the RF input and output, SubMiniature Version A (SMA) connectors are used. These are coaxial interfaces to screw into leads which conduct signal from one device to another.

Alongside with component details, validation data is also plotted. Some measurements charts where computed with automatic processes as gain measurements (represented as S_{21}) and VSWR graphics which were calculated from the S_{11} values on the output datafile of the network analyser (Appendix B may provide additional information about this).

Other graphics where manually registered as mixer conversion loss, VCO tuning curve and power detector output curve. Function generators, spectrum analysers and network analysers from the IT laboratories where used for the validation process.

[1]Common soldering techniques used in PCB assembly process: Hot Air Solder Leveling (HASL), Electroless Nickel/Immersion Gold (ENIG), Immersion Tin and Immersion Silver (ITIS) and Organic Solder Preservative (OSP). Among distinct characteristics, some of this techniques as ENIG provides high levels of co-planarity allowing properly solder to Ball Grid Array (BGA) packages[40].

4.1 Amplifiers

For this prototype three amplifiers are required. One in the RF stage where it needs to increase weak signals adding minimum possible noise, another in the IF stage providing power gain and a last one, not specified in block diagram 3.6. This extra amplifier is required as consequence of the power needs in the LO port of the mixer which are described in section 4.2. The block diagram 4.7 expands the aforesaid block diagram 3.6 in the LO brunch.

As a practical recommendation to avoid damage while testing amplifiers, the following connections sequences should be take in consideration [41]:

- ♦ **1 -** Connect the output load
- ♦ **2 -** Turn on DC power supply
- ♦ **3 -** Connect input load

And in the reverse order to disconnect:

- ♦ **1 -** Disconnect input load
- ♦ **2 -** Turn off DC power supply
- ♦ **3 -** Disconnect output load

4.1.1 Low noise amplifier

As this is the first component in RF circuit, it is essential that operational bandwidth feat for the input range requirements. Low noise figure is desired to improve weak signals detection. Actually the NF of this component is the most representative in equation 2.11 and has direct implication in the receiver sensitivity expressed in 2.7. As consequence this component assumes responsibility for the spectrum analyser prototype sensitivity figure of merit.

The **HMC374** is a LNA from Hititte Microwave Corporation [42]. The components specifications that led to choose this gain block are in table 4.1 and electric diagram in figure 4.1.

Five external components are needed to turn this device operable with a single power supply. Two DC blocking capacitors $C_1 = C_2 = 150pF$ for RF input and output terminations. Two more capacitors $C_3 = 1nF$ and $C_4 = 4.7\mu F$ to decouple power supply and an inductance $L_1 = 27nH$. Pins 1 and 4 are not connected. All values are explicit by manufacturer. The used PCB layout is present in figure I.1

After soldering the component into the PCB, this block is ready for measurements tests to confirm the correct functionality. The plotted graphics in figure 4.3a show the device gain in the region of interest from 800 MHz to 1 GHz and figure C.1 in all operating range (C.1a). These experimental gain curves correspond to the typical gain curves present in component's data sheet.

In the frequency range of interest from 880 MHz to 950 MHz the amplifier gain is around 13.2 dB. LNA's presents a lower gain in opposite to power gain amplifiers, due to the necessary trade off in added noise to the signal.

It is also possible to analyse impedance at 50 Ω in the VSWR charts (C.1b,4.3b). It is noticeable for the frequency range of interest a VSWR of approximately 2.

Characteristic	❖	Value
Operation bandwidth		300 MHz to 3 GHz
Single positive supply (V_{DD})		2.75 to 5.5 V
Supply current (I_{DD})		90 mA
Typical gain (G)		15 dB
Noise Figure (NF)		1.5 dB
Output IP3		37 dBm
50 Ω Inpedance matched		Yes
P1dB		22 dBm

Table 4.1: LNA **HMC374** specifications

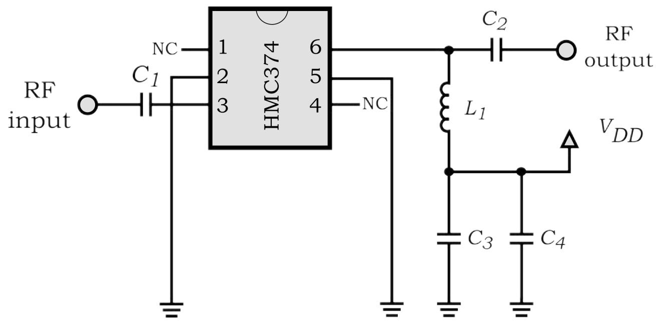
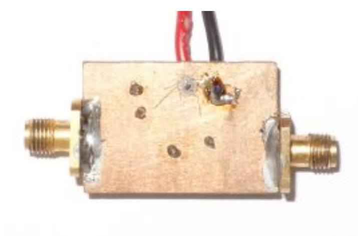


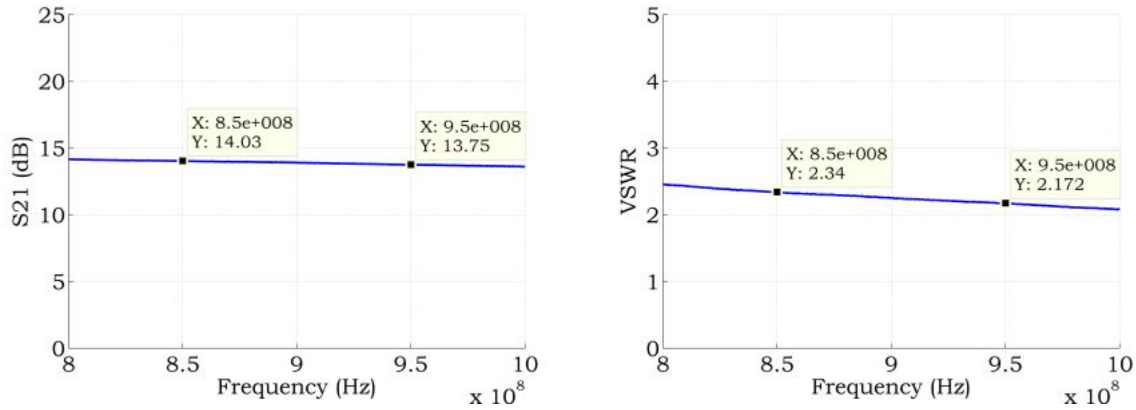
Figure 4.1: Electric diagram of **HMC374** evaluation board

(a) Top view



(b) Bottom view

Figure 4.2: Low noise amplifier **HMC374** evaluation board



(a) Gain from 800MHz to 1GHz

(b) VSWR from 800MHz to 1GHz

Figure 4.3: Low noise amplifier **HMC374** gain and VSWR

4.1.2 Power gain

As mentioned before two general propose RF amplifiers are needed. One for the IF stage and another one in the LO stage. These amplifiers are intended only to boost the respective input signal power so noise figure may be a more relaxed value than in case of LNA.

In the case of the amplifier needed to the IF section, it is necessary to restore power levels to a readable level to fed the power detector. A higher noise figure will bring a SNR degradation whit no major consequence to decipher the signal power.

In the LO brunch design is necessary to insert an amplifier to drive the VCO signal into the respective mixer port. Noise figure may affect the signal phase but for the objectives of this thesis, that will be tolerable.

IF gain

To choose the amplifier for the IF stage is necessary to ensure the device operating range which is considerably lower than the RF stage, fitting around 70 MHz. It will provide power gain at the output of the BPF bank, which is intended to allocate three filters centred at the same frequency but different in bandwidths. As the wider filter band has 20 MHz bandwidth leads to a frequency range of 60 to 80 MHz.

Here high gains are needed to compensate insertion losses from previous components in the circuit. Noise figure is neglected once the first LNA had imposed a high enough SNR to the signal.

For the propose will be used a power gain block from Hittite Microwave Corporation manufacturer, the **HMC580ST89** [43] which has a appropriate package style and characteristics are present in table 4.2. It is noticeable higher gain in deterioration of the noise figure. This device is also cascadable in 50 Ω systems.

In figure 4.4, is represented the electric diagram and the required external biasing components. The specifications in data sheet recommend components values for best performance in the frequency range that will be operating on.

Two blocking capacitors $C_1 = C_2$ are advanced as $0.01\mu F$ in the RF input and output. More three capacitors for power supply decoupling are used being C_3, C_4 and C_5 respectively

$100pF$ and $2.2\mu F$. A resistor $R_1 = 13\Omega$ is used for biasing effect. The PCB schematic is present in figure I.2

After assembling the evaluation board as a individual PCB, the component is take into test for gain and VSWR measurements.

In figure D.1a is represented the overall gain circuit which appears as the ones shown in the data sheet. With more detail in band of interest, figure 4.6a traces the gain from 60 MHz to 90 MHZ where may be noticeable a gain factor of 22.5 dB in the band of interest.

Looking into the impedance mismatch at 50 Ω indicated by the D.1b graphic and in a closer look in 4.6b, it is obsevable an input VSWR of 1.25.

Characteristic	❖	**Value**
Operation bandwidth		DC to 1 GHz
Single positive supply (V_{DD})		5 V
Supply current (I_{DD})		110 mA
Typical gain (G)		22 dB
Noise Figure (NF)		2.8 dB
Output IP3		37 dBm
50 Ω Inpedance matched		Yes
P1dB		22 dBm

Table 4.2: General amplifier **HMC580ST89** specifications

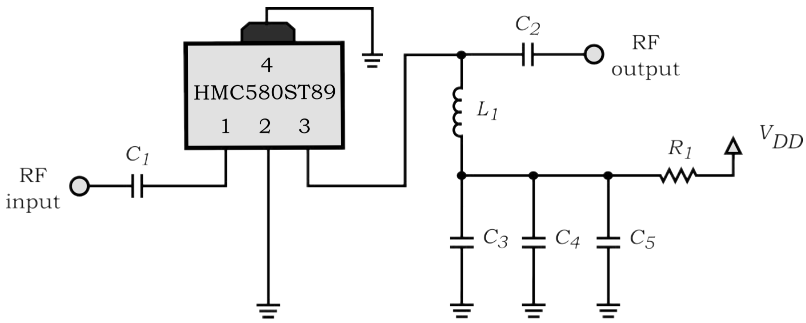
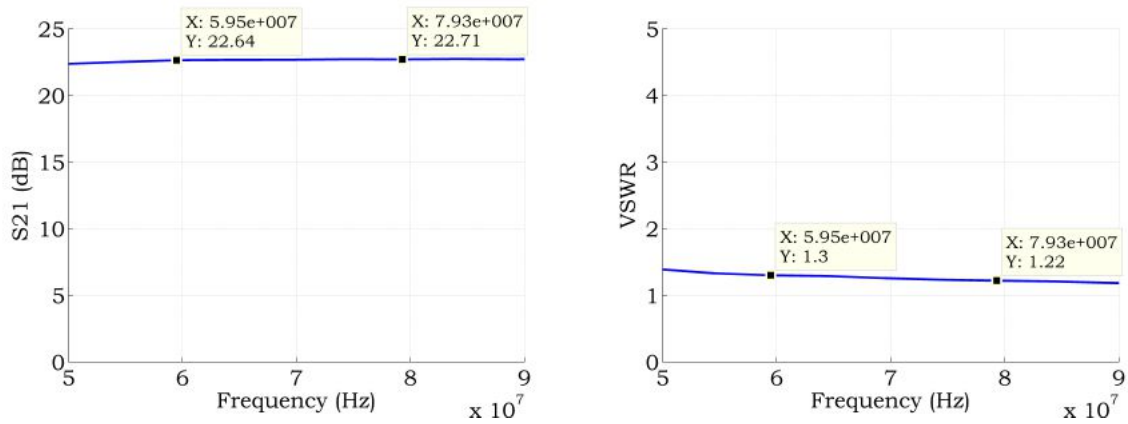


Figure 4.4: Electric diagram of **HMC580ST89** evaluation board

(a) Top view

(b) Bottom view

Figure 4.5: Amplifier **HMC580ST89** evaluation board



(a) Gain from 50MHz to 90MHz

(b) VSWR from 50 MHz to 90 MHz

Figure 4.6: Power gain and VSWR from **HMC580ST89**

LO stage

As mentioned earlier, in the LO stage will be used an amplifier as well. The block diagram 4.7 shows the insertion of an amplifier driving the VCO signal into the mixer LO port. This bridges the gap between the VCO output power and mixer LO power requirements, which if not matched, could lead to an inefficient frequency conversion.

The VCO is intended to sweep frequencies from 810 MHz to 880 MHz so has the amplifier operation bandwidth to be in the same band.

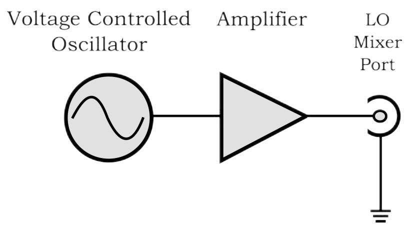


Figure 4.7: LO amplification stage

In IT laboratories there is an already assembled gain block evaluation board from Mini-Circuits manufacturer, the **ERA-5**+ [44]. The operational device specifications that make it suitable for the case are present in table 4.3.

Despite a higher power supply voltage, all the other characteristics fits the project needs. For prototype development and due to practical reasons as avoid extra cost, this situation may be tolerable.

The evaluation board electric circuit stands in figure 4.8. Blocking capacitors $C_1 = C_2 = 2400pF$ and $C_3 = 0.1\mu F$ for bypass effects. The component RFC is a TCCH-80+ from Mini-Circuitss and $R_1 = 110\Omega$.

Gain and VSWR tests were performed revealing the good functionality of this device and practical results are plotted in figure 4.10.

In the range of interest from 800 MHz to 900 MHz in figure 4.10a it is observable a gain factor of approximately 18.4 dB gain to the input signal. This will boost the VCO signal.

Looking into the VSWR plot 4.10b it is possible to analyse the component impedance match to 50 Ω. Is discernible a ratio value of 1.4.

Characteristic	❖	Value
Operation bandwidth		DC to 4 GHz
Single positive supply (V_{DD})		7 V to 20 V
Supply current (I_{DD})		65 mA
Typical gain (G)		20 dB
Noise Figure (NF)		3.5 dB
Output IP3		33 dBm
50 Ω Inpedance matched		Yes
P1dB		18.4 dBm

Table 4.3: General amplifier **ERA5+** specifications

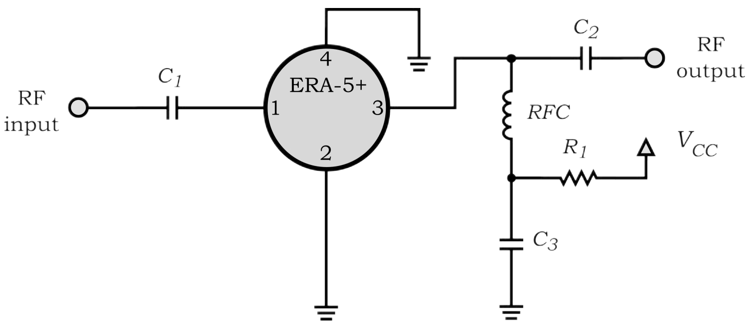
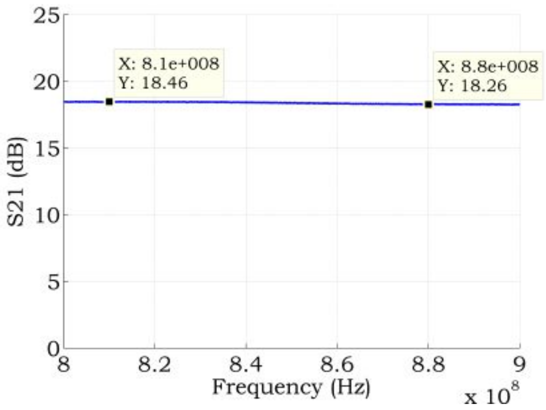


Figure 4.8: Electric diagram of **ERA-5+** evaluation board

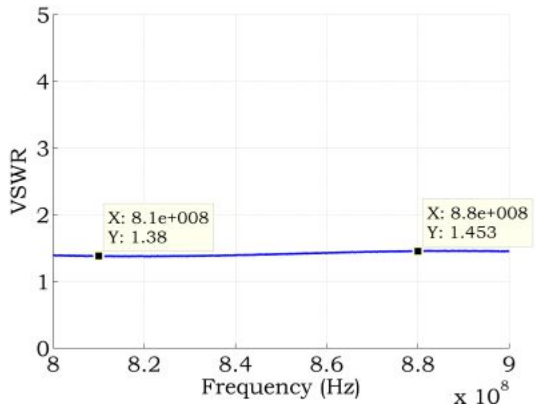
(a) Top view

(b) Bottom view

Figure 4.9: Amplifier **ERA-5+** evaluation board



(a) Gain from 800 MHz to 900 MHz



(b) VSWR from 800 MHz to 900 MHz

Figure 4.10: Experimental gain and VSWR from **ERA-5+**

4.2 Mixer

In this project the mixer is intended to perform down conversion from the RF input port to the IF port. The mixer needs to deal with a frequency range from 810 to 880 MHz in the RF port, allowing a IF centred in 70 MHz.

Mixers are available as passives or actives being the difference the power effect in signal. In case of passive mixers exists an attenuation from the input to the output as contrast in the active ones. As result passive mixers show conversion loss and active mixers have conversion gain.

The selected device is a passive mixer, **HMC207AS8** [45] from Hittite Microwave Corporation manufacturer. The characteristics which revealed this component are present in table 4.4. As a passive mixer it does not require DC power supply nor other components, and electric diagram is shown in figure 4.11 and PCB schematic in figure I.3.

It was performed a conversion loss test with two controlled RF generators to simulate the RF and LO mixer inputs. At the IF output was measured with a spectrum analyser the power over the frequency component of 70 MHz.

While RF power was kept constant, it were used four different power values for LO. The conversion loss experimental results are graphically expressed in 4.14. This results match the typical curves in data sheet.

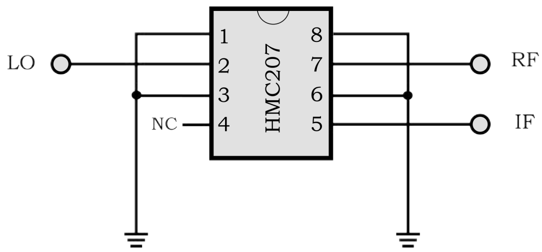


Figure 4.11: Electric diagram of **HMC207AS8** evaluation board

Characteristic ❖	**Value**
RF frequency range	700 MHz to 2 GHz
IF frequency range	DC to 300 MHz
Conversion loss	9 dB
Input IP3	17 dBm
LO power requirement	13 dBm
LO drive maximum input	27 dBm
Noise Figure NF	9 dB
50 Ω Impedance matched	Yes
Input P1dB	11 dBm

Table 4.4: Mixer **HMC207AS8** specifications

(a) Top view

(b) Bottom view

Figure 4.12: Mixer **HMC270AS8**

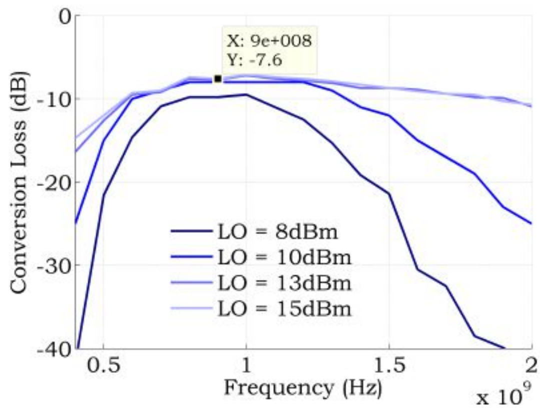


Figure 4.13: Experimental conversion loss

Figure 4.14: Mixer **HMC270AS8** conversion loss

4.3 Voltage controlled oscillator

In LO stage is necessary an variable oscillator. The frequency range for this device should be sweep-tuned from 810 to 880 MHz.

A model that fitted in the system requirements is the **CVCO55BE-0800-1600** [46, 47] from Crystek Microwave Corporation. The component characteristics are in table 4.5. It has a tuning range from 800 MHz to 1600 MHz and requires a power supply of 12 V.

The electric diagram is present in 4.15. This device is connected to a power supply in pin

14 with a bypass capacitor $C_1 = 0.1\mu F$ to ground. The tuning voltage is applied on pin 2 and an additional capacitor $C_2 = 1nF$ to ground as well. The RF signal is obtained from pin 10. In figure I.4 is present the PCB layout.

The applied tuning voltage while circuit operates as a spectrum analyser, will be performed by digital control and will be explained later together with the VCO experimental results in subsection 5.3.1.

Characteristic	❖	**Value**
Tuning range		m800 MHz to 1.6 GHz
Single power supply V_{DD}		12 V
Output power		6 dB
Tuning sensitivity		60 MHz/V
Phase noise @ 10 kHz offset		-100 dBc/Hz
Supply Current		15 mA
50 Ω Impedance matched		Yes

Table 4.5: Voltage controlled oscillator **CVCO55BE-0800-1600** specifications

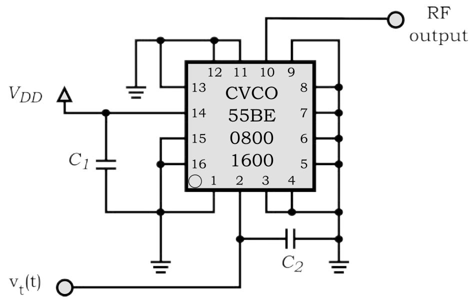


Figure 4.15: Electric diagram of VCO **CVCO55BE-0800-1600** evaluation board

(a) Top view

(b) Bottom view

Figure 4.16: VCO **CVCO55BE-0800-1600** evaluation board

4.4 Band pass filters

Following the signal split there are the BPF on the circuit chain. For the project three filters centred at the same center frequency fc and distinct in bandpass bandwidth.

The components search revealed three SAW[2] filters, from Vanlong Technology Co., Ltd. manufacturer. SAW filters have the advantages of small size, reliability and no tuning needs.

With an appropriate package to implement them on a PCB and easy to adapt impedance to 50 Ω, the three filters are introduced in respective order of 1, 5 and 20 MHz bandwidths. Those values dictate the spectrum analyser RBW options.

The three filters were soldered into one PCB, which schematic is depicted in figure I.5, as a evaluation board with RF connectors. Insertion loss and VSWR tests were conducted to confirm devices functionality.

With 1 MHz of bandpass bandwidth the **BP60370** [48] will be the narrower in the filter bank. As results the spectrum analyser will be able to solve signals that are distant more than 1 MHz. The filter characteristics are described in table 4.6.

This component requires inductors $L_1 = L_2 = 220nH$ in series with the input and output ports. As the test results of this filter, figure 4.18 display the insertion loss and VSWR measurements over the bandwidth range of interest, from 50 to 90 MHz. At the 1 MHz passband this filter introduces an attenuation of nearly 24 dB. In the stop bands the filter can reduce 50 dB the power of the non wanted frequencies. The VSWR chart in 4.18b shows a ratio of proximately to 3.8 inside of the device passband.

Characteristic	❖	**Value**
Center frequency (f_c)		70 MHz
Insertion loss at f_c		20.8 dB
1 dB Bandwidth		1 MHz
3 dB Bandwidth		1.2 MHz
40 dB Bandwidth		2.1 MHz
50 Ω Impedance matched		Yes

Table 4.6: BPF **BP60370**

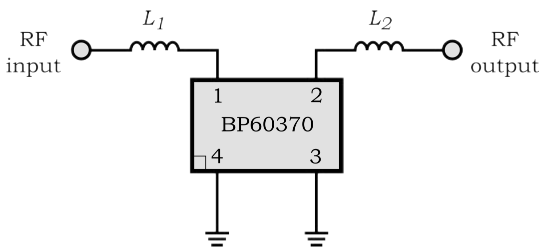
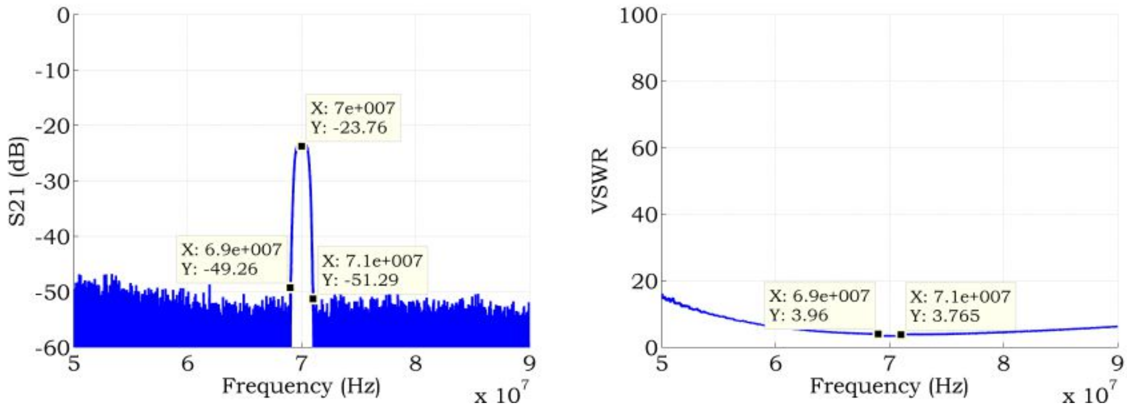


Figure 4.17: Electric diagram BPF **BP60370**

Rather wider the filter **BP60290** [49] has a passband of 2.5 MHz. The component characteristics are present in table 4.7. For instance this filter does not require any other component to operate in 50 Ω systems. The electric diagram is present in figure 4.19.

The test results of the insertion loss and VSWR measurements over the device are plotted in figure 4.20. It is possible to see an attenuation inside the passband of almost 23.1 dB and

[2]SAW filters are passive and integrated devices, characterized as a typical filter. Its operation theory is based on the propagation of mechanical surface waves and are constructed with piezoelectric material.



(a) Insertion loss (b) VSWR

Figure 4.18: BPF **BP60370** insertion loss and VSWR

50 dB outside. In the VSWR plot a ratio of 8.5 is recognizable inside the passband of interest.

Figures shows the results of insertion loss measure of the three filters. Despite an higher attenuation inside the pass band of **BP60370**, still respect the manufacturer specifications.

It is noticeable an attenuation of approximately 24 dB inside bandpass.

Characteristic ❖	**Value**
Center frequency (f_c)	70 MHz
Insertion loss at f_c	23.9 dB
1 dB Bandwidth	4.8 MHz
3 dB Bandwidth	5.2 MHz
40 dB Bandwidth	7.1 MHz
50 Ω Impedance matched	Yes

Table 4.7: BPF **BP60290**

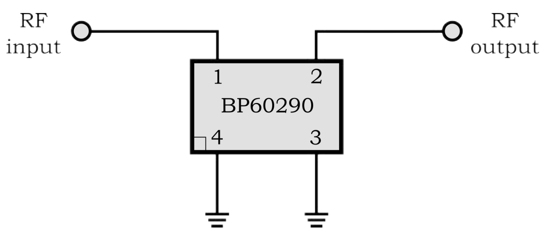
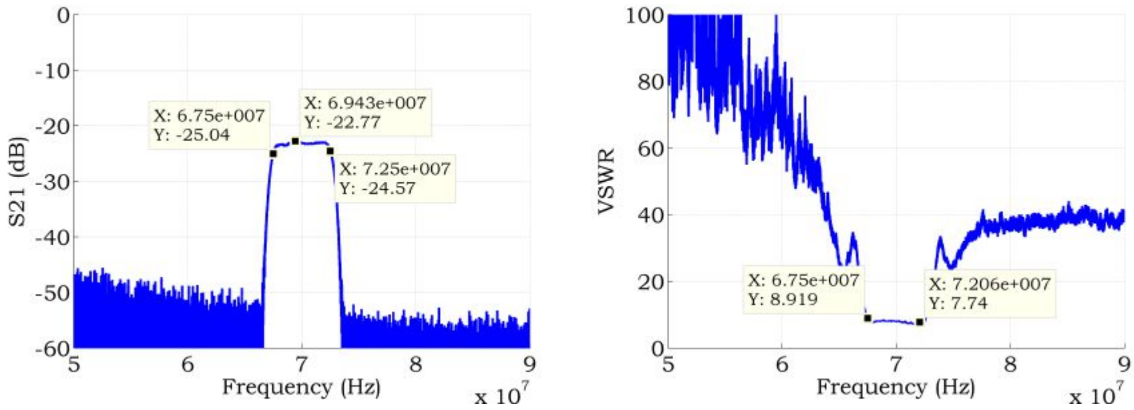


Figure 4.19: Electric diagram BPF **BP60290**

The last filter and the larger in bandwidth, from the same manufacturer is the **BP60110** [50]. It has a passband bandwidth of 20 MHz and general specifications are shown in 4.8.

Electric diagram of this device is depicted in figure 4.21. An inductance value of 276 nH is recommend to accomplish 50 Ω impedance match. Due to the commonly find inductors values, L_1 is defined by the manufacturer as 220 nH + 56 nH. Two inductors should be placed in series to do L_1.



(a) Insertion loss (b) VSWR

Figure 4.20: BPF **BP60290** insertion loss and VSWR

The obtained data from experimental evaluation is present in figure 4.22. In 4.22a is plotted the component insertion loss. Inside the passband from 60 to 80 MHz signal frequencies are attenuated by a factor of 25.5 dB. VSWR measurements are explicit in figure 4.22b.

Characteristic ❖	**Value**
Center frequency (f_c)	70 MHz
Insertion loss at f_c	23.9 dB
1 dB Bandwidth	19.4 MHz
3 dB Bandwidth	20.6 MHz
40 dB Bandwidth	25.1 MHz
50 Ω Impedance matched	Yes

Table 4.8: BPF **BP60110**

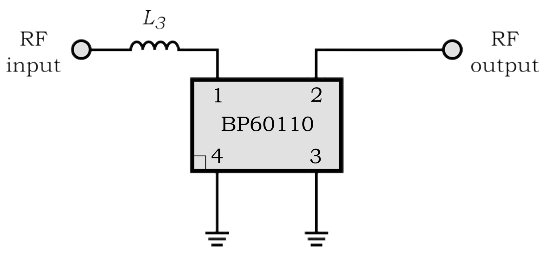
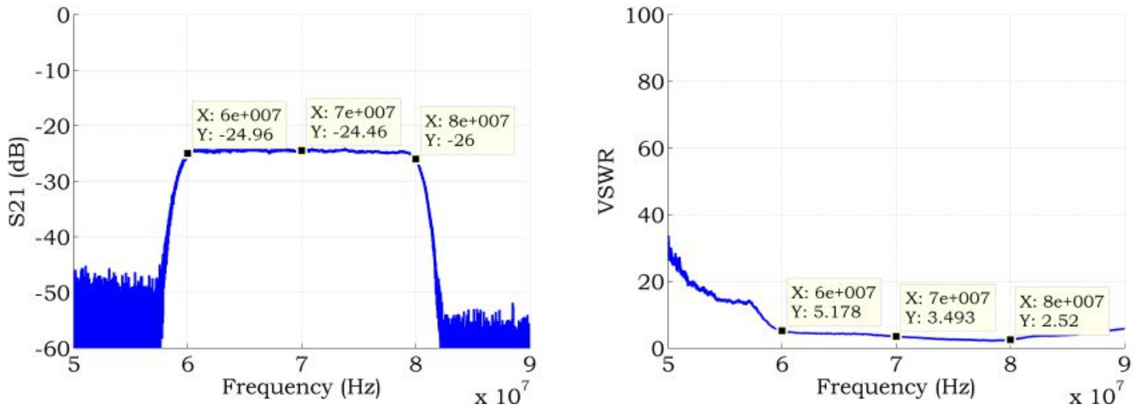


Figure 4.21: Electric diagram BPF **BP60110**

To conclude the filter implementation subsection it pictured in figure 4.23, an image of the PCB containing the three filters. From top to bottom there are the 5, 1 and 20 MHz filters.

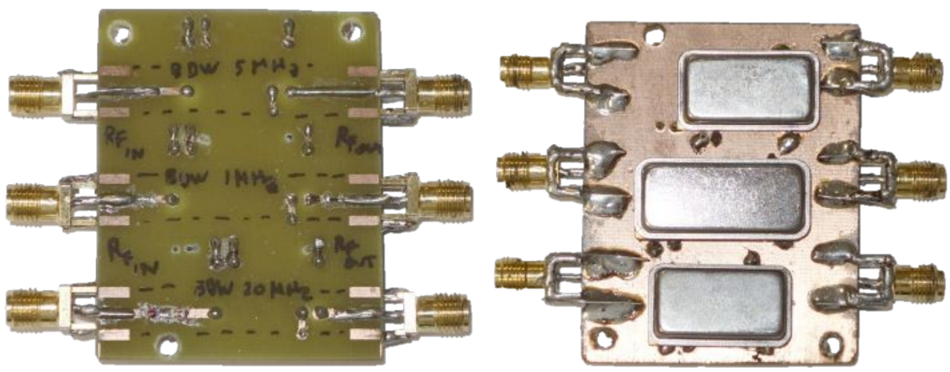
4.5 RF switch

To implement the filter bank in this project, it is necessary a circuit which is known as a RF switch. This device is digitally controlled and is suited to carry the selection of the RBW



(a) **BP60110** insertion loss (b) **BP60110** VSWR

Figure 4.22: BPF **BP60110** insertion loss and VSWR



(a) Top view (b) Bottom view

Figure 4.23: Filters evaluation board

in this spectrum analyser prototype.

It was used from IT laboratories a vacant RF switch evaluation board from Hittite Microwave Corporation. It is the **HMC252QS24** [51] and specifications of this circuit are in table 4.9. The evaluation board diagram is present on figure 4.24 where capacitors C_1 to C_7 are recommend by manufacturer to be $100pF$. Capacitors C_8 to C_{11} should be a capacitive value of $10nF$.

This device has 6 inputs and so it requires at least 3 control pins A, B and C, to select each channel to the RF COM port. For the bank filter in this project 3 channels are needed. The truth table 4.10 contains the logic high (⎍) and low (⊔) combinations to select channel 1, 2 and 3. As it can be observed, pin C can be fixed to a low level and therefore only pin A and B demand to be controlled.

On figure 4.26 is present the experimental results of insertion loss and VSWR for channel 1 and in E.1 for channel 2 and 3.

Characteristic	❖	Value
Frequency range		DC to 3 GHz
Single positive supply (V_{DD})		3.3 or 5 V
Typical insertion loss		0.9 dB
Isolation between ports		38 dB
Input IP3		46 dBm
Input P1dB		24
50 Ω Impedance matched		Yes

Table 4.9: RF switch **HMC252QS24** specifications

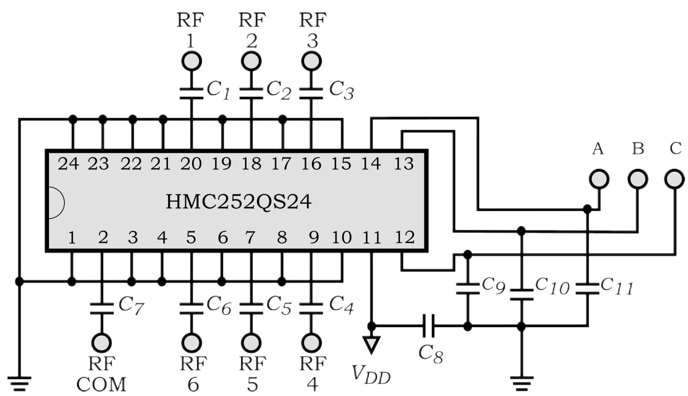
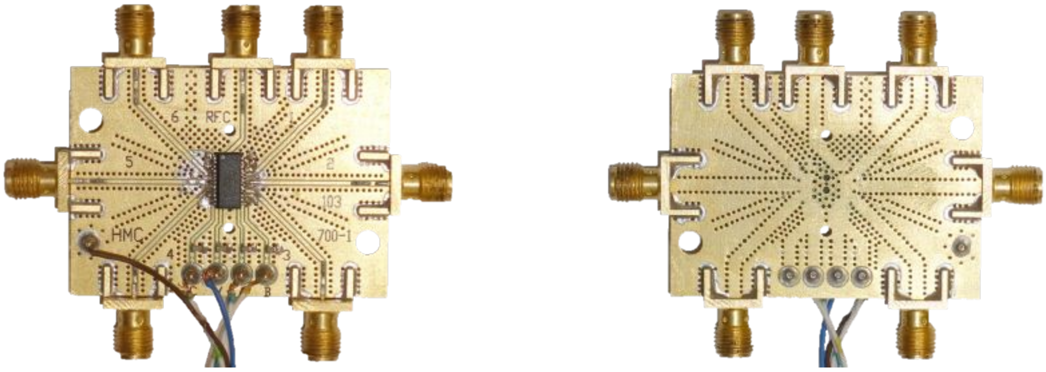


Figure 4.24: Electric diagram RF switch **HMC252QS24**

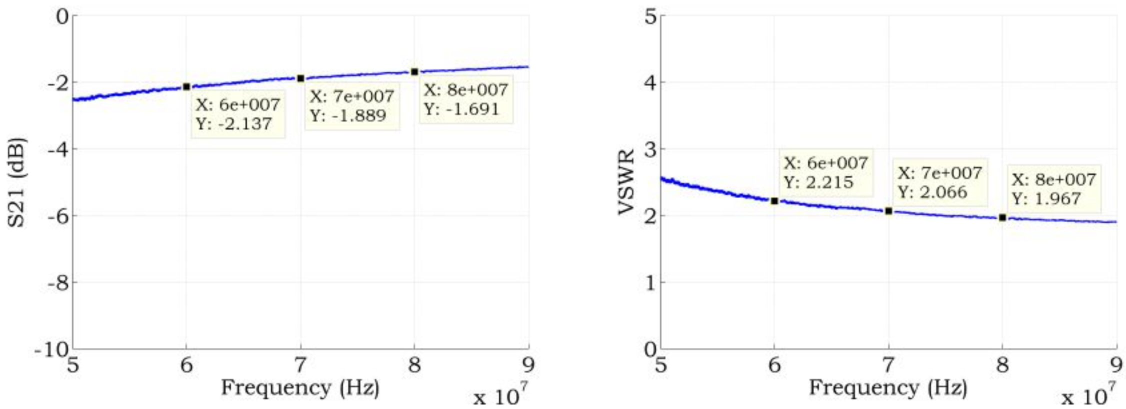
A	B	C	❖	RF COM to
‾|_|‾	‾|_|‾	‾|_|‾		RF 1
_|‾|_	‾|_|‾	‾|_|‾		RF 2
‾|_|‾	_|‾|_	‾|_|‾		RF 3

Table 4.10: RF switch **HMC252QS24** truth table



(a) Top view (b) Bottom view

Figure 4.25: RF switch **HMC252QS24** evaluation board



(a) Channel 1 insertion loss (b) Channel 1 VSWR

Figure 4.26: RF switch **HMC252QS24** insertion loss and VSWR

4.6 Power combiner

To collect the IF signal from the output filter bank is necessary to conduct the signal three different filters till the the power detector. This may be implemented with a power combiner.

At IT laboratories there is available such device whit the necessary characteristics for this spectrum analyser prototype. It is one to four way plug in coaxial power combiner from Mini-Circuits manufacturer, the **ZSC-4-1**+ [52].

This device is optimized to operate in the frequency range of 100 kHz to 200 MHz in 50 Ω systems. This turns to be appropriate to handle the 70 MHz centred IF channel. The component characteristics are expressed in table 4.11.

In figure 4.28 is present the insertion loss measure for channel 1. Channel 2 and 3 tests are present in figure F.1 in appendix F due to the similarity of the data plots.

Frequency was swept in the range of 50 to 90 MHz in order to understand the component attenuation inside of the interest frequencies band.

Channel 1, 2 and 3 attenuate the input signal by a factor of approximately 6dB, as specified by the manufacturer data sheet.

Characteristic	**Value**
Number of channels	4
Frequency range	100kHz to 200 MHz
Insertion loss	6 dB
Isolation between ports	30 dB
50 Ω Impedance matched	Yes

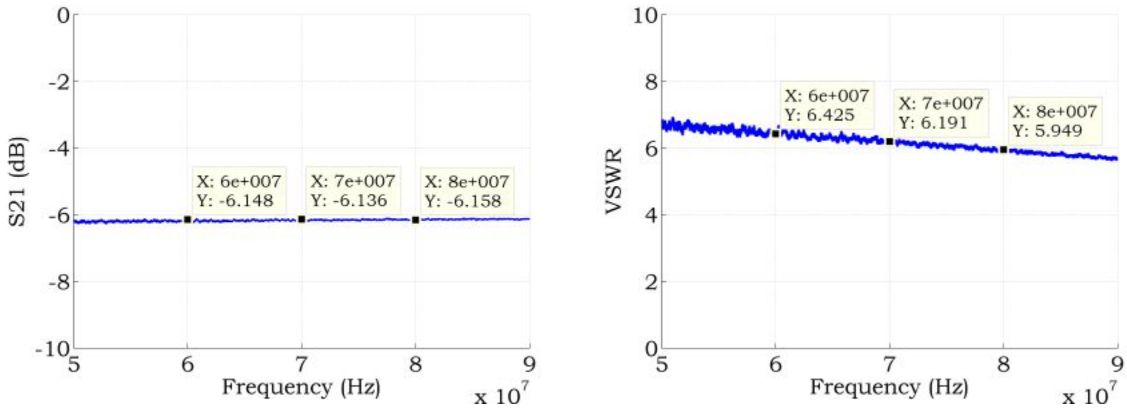
Table 4.11: Power splitter **ZSC-4-1**+ specifications

Figure 4.27: Power combiner **ZSC-4-1**+

4.7 Power detector

The power detector is the component responsible to convert incoming RF power to a DC voltage reference. The selected component is the **LTC5507** [53] from Linear Technology Corporation. Characteristics are in table 4.12.

This device covers a wide frequency range and so manufacturer recommend some adjustments to the evaluation board. In order to the lower incoming frequency two capacitors



(a) Channel 1 insertion loss

(b) Channel 1 VSWR

Figure 4.28: Power splitter **ZSC-4-1+** insertion loss and VSWR

should be projected. This will allow best performance in the frequency range of interest. As elucidated in component data sheet it follows:

$$C_1 = C_2 = \frac{1}{30f} \tag{4.1}$$

Where capacitors C_1 and C_2 are in μF units and f is the incoming frequency in MHz. As the result of equation 4.1 considering the lowest frequency of interest in this stage of $f = 60MHz$[3], capacitors C_1 and C_2 should match a value of 5.5 μF. To note in practice were used 5.6 μF capacitors. Incoming RF path is connected to ground with means of $R_1 = 68\Omega$ which combined with capacitor C_1 performs a high pass filter to attenuate lower frequencies into power detector. The used PCB schematic for this component is present in figure I.6.

With a RF generator were injected oscillating signals centred at 60, 70 and 80 MHz respectively. The signal power may be manually swept from -30 dBm to 20 dBm to trace an experimental DC output curve on pin 6. The registered results are present in figure 4.31. Whith this device is possible to distinguish RF signals with power levels above -22 dBm till 15 dBm. Signals with lower power levels produce a hardly distinguishable output DC voltage while it saturates for higher power levels then the referred.

Characteristic	**Value**
Input frequency range	100 kHz to 1 GHz
Single power supply V_{DD}	2.7 to 6V
Operating current	550 μA
Input power range	-30 dBm to 16 dBm
Logarithmic output	Yes
50 Ω Impedance matched	Yes

Table 4.12: Power detector **LTC5507** specifications

[3]Filter bank is centred at 70 MHz with a maximum of 20 MHz bandpass bandwidth, allowing to reach the power detector frequencies from 60 to 80 MHz.

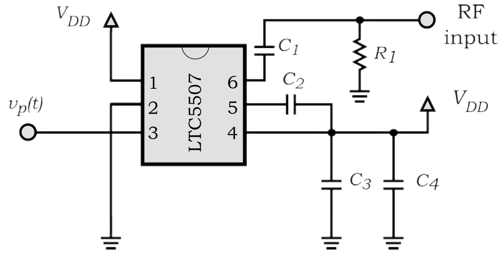


Figure 4.29: Electric diagram of **LTC5507** evaluation board

(a) Top view

(b) Bottom view

Figure 4.30: Power detector **LTC5507** evaluation board

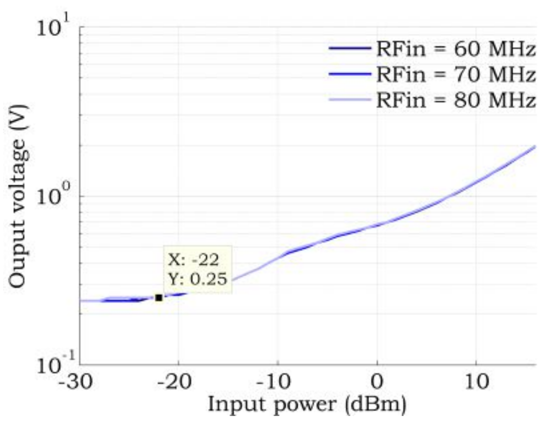


Figure 4.31: **LTC5507** Experimental output curves

4.8 Digital control - Arduino

To run the entire RF circuit as a spectrum analyser, automatic functions should take place to process and control electrical parameters. A very attractive option to implement the automatic control is with microcontrollers. There are several microcontrollers platforms and they can be found in market with an extensive set of specifications varying from manufacturer to manufacturer and from model to model. In the scope of this thesis the decided platform to use is the **Arduino**.

"*Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software* [54]".

Mainly this project provides a ready to use microcontroller platform board with power connections and a clean electronic interface to wire it up to other devices. Beside to the *open-source* characteristics [55], this project made simple the work process with microcontrollers providing a high-level programming abstraction and an efficient USB interface. Programming can be done in a own Integrated Development Environment (IDE) with a proper language based on Wiring which is "*an open-source programming framework for microcontrollers*" [54, 56].

Amongst distinct **Arduino** boards very different in specifications, the selected one was the **Arduino Uno R$_3$** which can be assumed as a basic product from the vast available gamut. This one has the following characteristics expressed in table 4.13.

This microcontroller works with 5 volts reference, turning the digital logic values as 0 and 5 V, for *low* and *high* respectively. In figure 4.32 is present the electric diagram of the **Arduino Uno R$_3$** connected to electric conditioning circuits to perform the following tasks:

- VCO sweep tuning control ($v_t(t)$).
- Read the DC voltage from power detector pin 3 ($v_p(t)$).
- Select resolution bandwidth channel actuating on pins A and B from RF switch.
- Update the screen with the collected data.
- Adjust measurement options under user requirement.

Characteristic ❖	**Value**
Microcontroller	ATmega328
Single power supply V_{DD}	5V
Total digital I/O pins	14
Pulse Width Modulation (PWM) pins	6
Analogue to digital input pins	6
Flash memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock speed	16 MHz

Table 4.13: **Arduino Uno R$_3$** board specifications

Furthermore this prototype is intended to work with batteries to operate as a handled version and so, they are represented as $V_M = 12V$. This will may be understood as the main power source for the circuit an it will be feeding a voltage regulator to provide a established source $V_{DD} = 5V$ to the RF components.

The used voltage regulator is an **UA7805** IC from Texas Instruments [57], which fixes an output voltage in 5 V and can be feed within 7 to 25 V. Recommended values for $C_1 = 0.33\mu F$ and $C_2 = 0.33\mu F$.

The **Arduino** board has an internal voltage regulator which can be plugged till 12 V and is possible to get from pin V_{in} $Vcc \approx V_M - 0.8$ and $V_{DK} = 5V$. This may be helpful to power up the RF components which require a higher supply voltage to V_{in}, and to power up the keyboard whit a stand alone voltage reference V_{DK}.

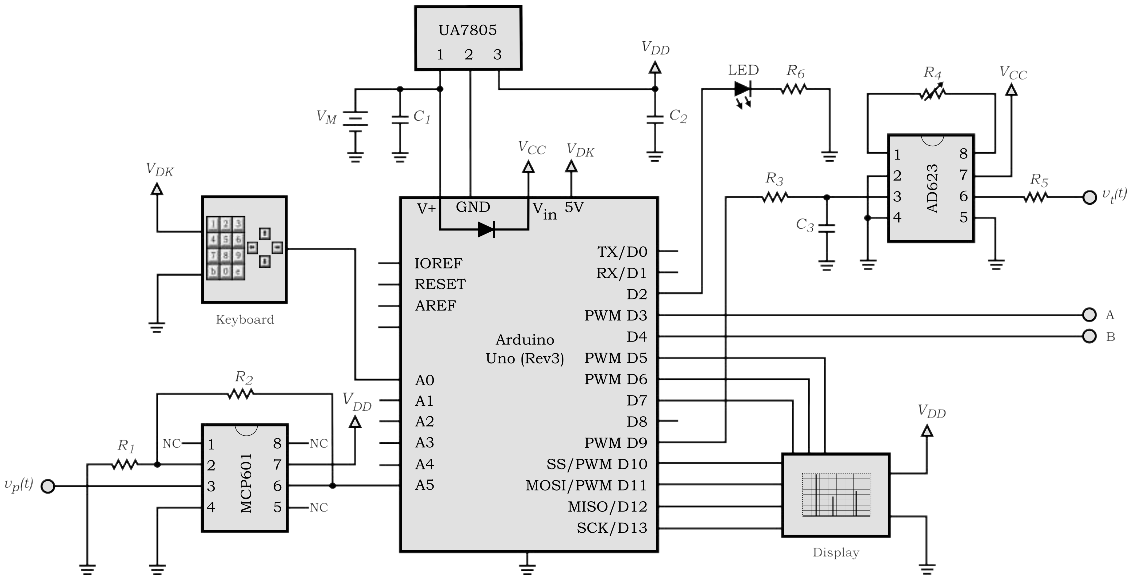
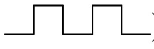
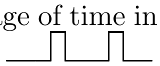


Figure 4.32: Electric diagram - Arduino

4.8.1 VCO control

To control the frequency sweep, the microcontroller should act over the VCO with a tuning DC tension. The selected **Arduino** board does not have any integrated DAC which would be the most intuitive way to accomplish that, though a PWM can be generated in respective pins, and with a complementary LPF (formed with $R_3 = 32.4k\Omega$ and $C_3 = 100nF$) to produce a DC voltage between 0 and 5V.

A PWM port may be programmed with internal timers to oscillate from low to high logic values, producing a square wave (50% duty-cycle ex: ) . Controlling the duty-cycle which is the percentage of time in relation to an oscillating period where the amplitude is high (25% duty-cycle ex: ), is possible to change the average power delivered to the LPF.

As result it is plausible to produce a sawtooth shaped DC voltage to successively tune the VCO output frequency. Figure 4.33 illustrates a PWM wave form, with duty-cycle being linearly incremented,to produce such voltage variation after the LPF.

The device **AD623** [58] is a single supply, rail-to-rail amplifier from Analog Devices Inc. The gain adjustment is manually executed with a precision potentiometer $R_4 = 50K\Omega$ extending the DC tuning range value as $v_t(t)$. Resistor $R_5 = 5.62K\Omega$ propose is limiting the

VCO current draw, which has a high input capacitance and if not there, the output frequency power would not be constant.

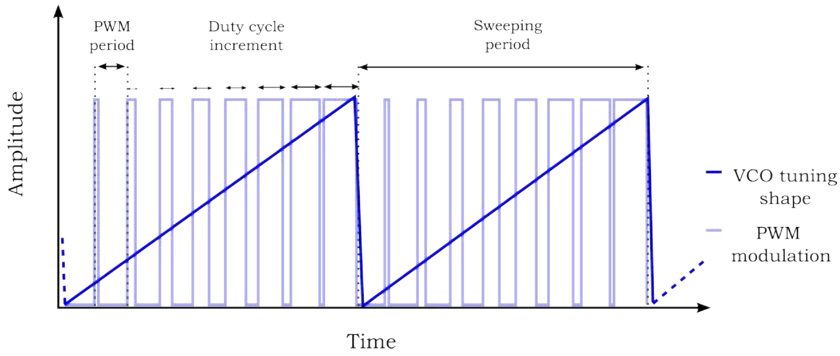


Figure 4.33: PWM wave form to produce a saw tooth

4.8.2 Power Read

The output DC voltage from the **LTC5507** power detector vary between 0.26 V till nearby 2V. With an amplifier is possible to extend this voltage range fulfilling the ADC extension. This mean bring the maximum output voltage level from **LTC5507** to 5V. This may provide a better differentiation about the incoming RF voltage levels captured by the ADC .

The used device to amplify the present voltage in pin 3 from the RF power detector is a rail-to-rail, single power supply the **MCP601** [59] from Microchip Technology Inc. With a non inverting configuration the gain expression can be stated as in equation 4.2. With $R_1 = 3.4K\Omega$ and $R_2 = 5.49K\Omega$ it is possible to get a gain factor of $Gain \backsimeq 2.6$.

$$Gain = 1 + \frac{R_2}{R_1} \tag{4.2}$$

4.8.3 Filter Selection

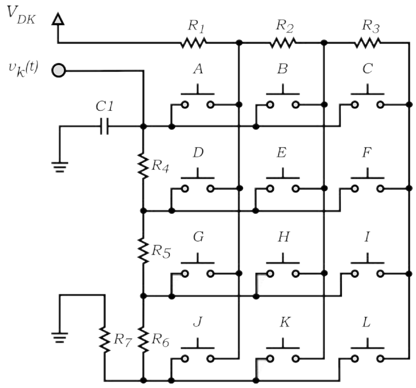
Two digital output pins are required to select one of the three BPFs in the filter bank. The truth-table 4.10 already mentioned indicates the combinations on pins A and B necessary to select the desired RBW.

4.8.4 Keypad

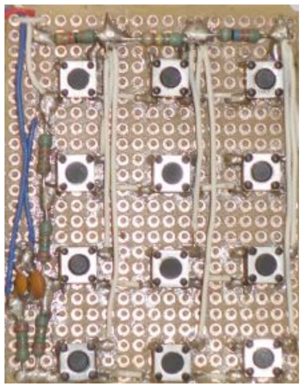
To human interaction, it was designed a resistive, following the electric diagram present in figure 4.34. This keyboard is intended to work with 3 connections being two of them the power supply and ground, and the left pin is connected to a **Arduino** ADCs pin (A0). The **Arduino** will get a voltage value from A to L (described in appendix G) whenever each switch is pulsed and 0 V when no interaction is happening. Capacitor C_1 couples the ADC pin to the ground to complete the circuit when no key is pressed.

4.8.5 Display

Over the years CRT screens have been kept away from electronic projects due to size and versatility of LCDs screens. This new technology is based on digital control over a pixel



(a) Electric diagram



(b) Implementation on a perforated board

Figure 4.34: Keyboard design

matrix. In figure 4.35 is pictured a upper left corner of such screen and the coordinate system do address each pixel.

Now it may be mentioned a benefit about work with the **Arduino** project, which is the emergence of commercial peripherals optimized to easy integration. Looking forward to an appropriate screen to this spectrum analyser project it was found an adaptation of a Color Thin-Film-Transistor (TFT) Shield from Adafruit Industries to **Arduino** [60], which come with a graphic library to be programmed. The specifications of this TFT screen are listed in table 4.14.

This shield module still have a micro Secure Digital (SD) slot card and a joystick resistive pad which will not be used for now.

Characteristic ❖	**Value**
Single power supply V_{DD}	5V
Screen size	128×160 pixels
Colour resolution	18 bit
Communication Protocol	Serial Peripheral Interface (SPI)
Required connection pins	6 pins.

Table 4.14: Color TFT Shield specifications

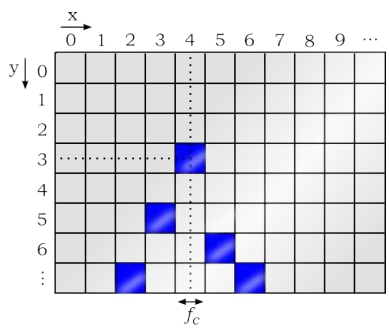
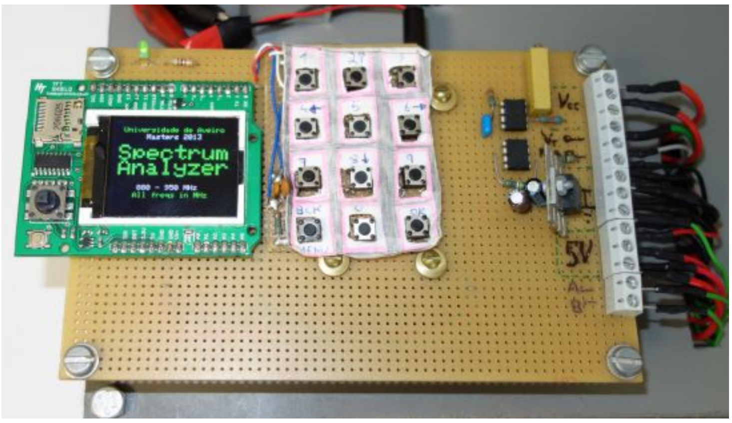


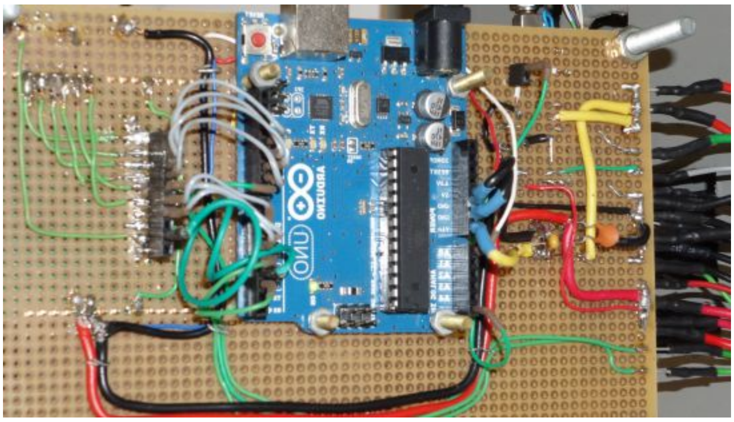
Figure 4.35: Screen coordinate system for pixels

4.8.6 Interface board

To accommodate the necessary electronic circuits to perform the digital control over all circuit, a perforated board was used and the final may be seen in figure 4.36. This interface board provides a support for the TFT screen, keypad, accommodation electronics and power distribution for the RF circuits. On the bottom side lies the **Arduino** board and connection wires.



(a) Top side - TFT, keypad and power supply



(b) Bottom side - **Arduino** connections

Figure 4.36: Interface board

Chapter 5

Tests and Results

The circuit parts that have been detailed in chapter 4, were assembled together and programmed to allow power measurements over frequency under user instructions. In this chapter are detailed the assemblage results and practical measurements.

Remembering the main goal of this master thesis, was to project and implement a device able to perform power measurements over frequency in electromagnetic spectrum, under a set of specifications like a selectable frequency range, RBW and sweeping times. As well important for such device are auxiliary functions which guide the user through the spectrum measurement process.

5.1 Prototype hardware

The photograph 5.1 shows the implemented spectrum analyser prototype in a test setup environment. The prototype itself consists on a physical support, where all circuits stay motionlessness and practical to transport. In the setup stands a power supply over a RF function generator feeding the prototype.

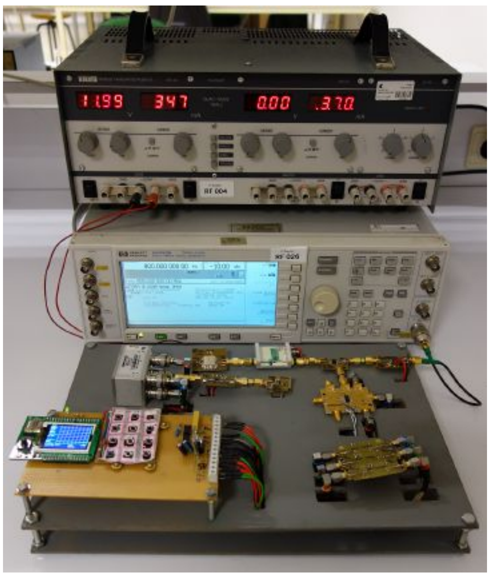


Figure 5.1: Prototype test setup

For additional detail, the shot moment had captured the circuit powered on and running performance tests for power measurements over the 880 to 950 MHz range.

In the prototype, different evaluation boards are connected with SMA male to SMA male cables and adapters. The RF input is situated in the top right corner of the prototype, where the first RF block in the circuit is connected to a single coaxial cable which delivers the RF signal reference from the RF function generator.

Still describing the last referred picture, there are two visible wires on the left side, a red and black, connecting the interface board to a DC power supply.

5.1.1 Current consumption

While trial tests goes one, it may be analysed the current consumption whit the lab power supply exemplar, a Thurlby Thandar **PL320QMT** which may be used fore current measurement reference.

In this test, the voltage is set to 12 V and approximately 350 mA are being drawn. These results provide a rough estimation to chose an adequate battery which may feed latter the prototype.

5.2 Specifications

To calculate receiver sensitivity theoretically with equation 2.7, it is necessary to use the NF (equation 2.10) from the entire circuit. From equation 2.11, where the receiver NF is considered, the first term is taken as the most significant, will be considered as the $NF = 1.5dB$ from the LNA.

The sensitivity value for this spectrum analyser is associated with each RBW. The noise floor power value increases alongside with the passband. Equations 5.1,5.2 and 5.3 expresses the sensitivity for the 1 MHz, 5 MHz and 20 MHz RBW channels respectively, whit a minimum SNR of 3 dB.

$$S_1 = -174 + 1.5 + 10log(1 \times 10^6) + 3 = -31.4dBm \tag{5.1}$$

$$S_2 = -174 + 1.5 + 10log(5 \times 10^6) + 3 = -24.4dBm \tag{5.2}$$

$$S_3 = -174 + 1.5 + 10log(20 \times 10^6) + 3 = -1.4dBm \tag{5.3}$$

Analysing illustration 5.2 where the RF chain is described with each component specifications it is possible to get some results about the circuit working as aradio receiver for spectrum analysis.

Also in the lower part of the image is represented the power budget in the circuit. From left to right it is considered the maximum allowed power on the input of the circuit, preventing power levels to reach input $P1dB = 11dBm$, of the mixer. It was get a maximum power in the RF input of $P_{max} = -2.2dBm$.

In the reverse order are calculated the minimum detectable signals for each RBW channel, considering the smallest power level detectable in the power detector of -22 dBm.

This last conclusion leads to the dynamic range of each RBW channel. The subtraction result from maximum with minimum allowed power levels for each channel is 24.2 dB which

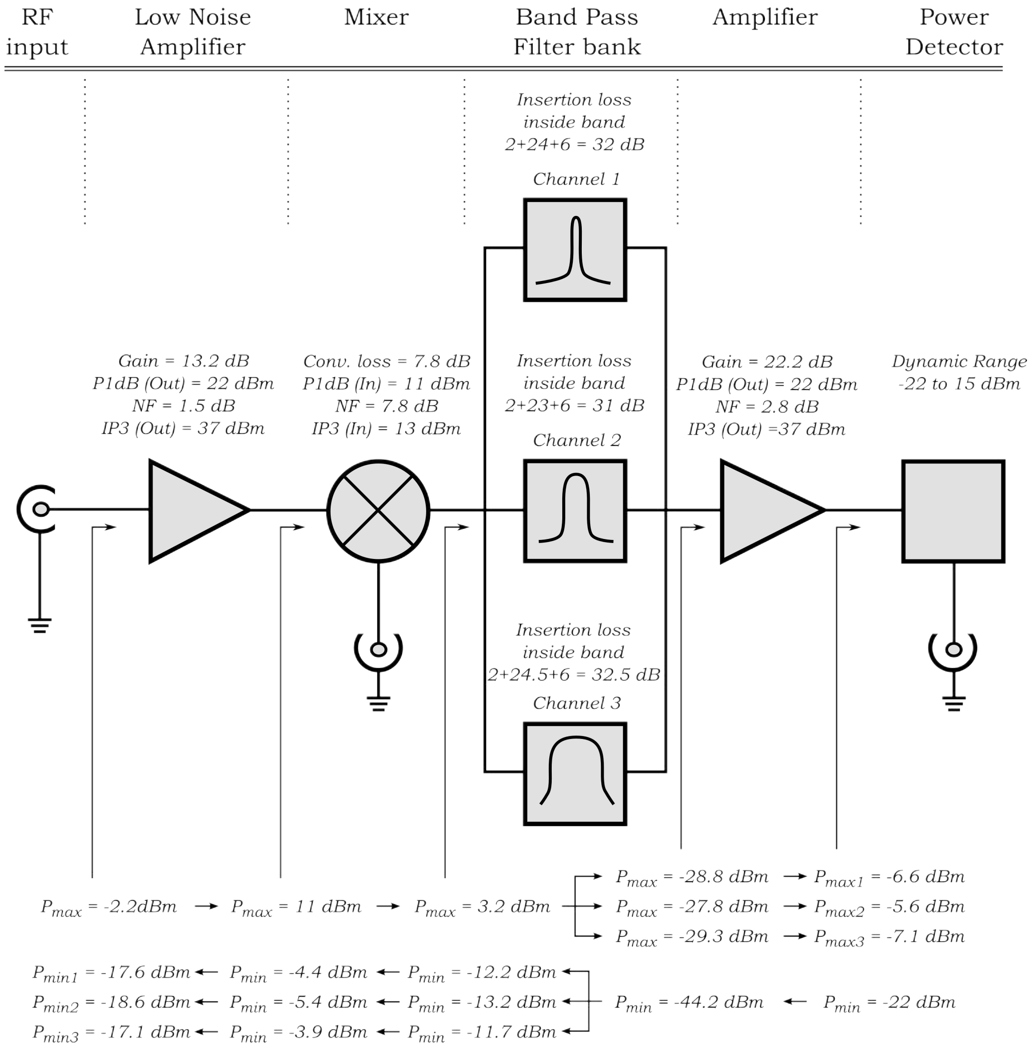


Figure 5.2: RF chain overview

turns to be the dynamic range reference

$$Dynamic\ Range = P_{max\mathbf{i}} - P_{min\mathbf{i}},\ \mathbf{i} \in 1,2,3 \tag{5.4}$$

5.3 Software control

The flowchart present in figure 5.3 illustrates the operational stages being executed by the microcontroller. It starts pre-loading the related information with circuit control and then checks the actual running time.

The read time value is compared with a reference which is used to control the number of data updates per second in the display. The reference time parameter may be adjusted by user instruction reflecting on different sweeping times.

Moreover, when time counter overlaps the time reference, the control sequence to update data into screen is initiated. It increment and the duty-cycle from the lower level (which represents the first frequency of the selected range), read the actual voltage value of the RF power detector, and update the graphic display.

The power levels are stored in volatile memory for posterior computation and this concludes the maintenance routine.

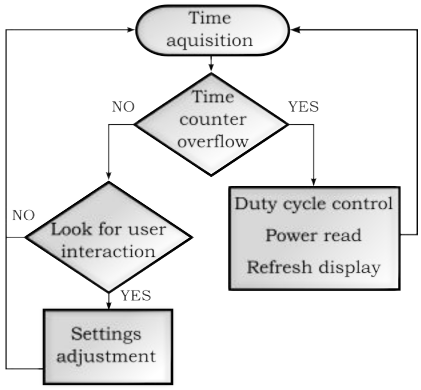


Figure 5.3: Software flowchart

While the the acquired time had not overlapped the time reference, it is looked for human interaction on the keyboard. When input user instructions have been detected the measurement process is suspended, and a menu interface (figure 5.4 will conduct user through settings adjustment.

The available options are described in the following list as:

- ♦ Frequency adjustment to the range specified from start to stop frequency.
- ♦ Select a RBW of 1, 5 and 20 MHz.
- ♦ Span time adjustment with trough time reference variance.
- ♦ Graphic markers to conduct user through measurements.

This group of functions forms a basic set to perform power over frequency measurements. All are implemented in the source source code for the **Arduino**

Figure 5.4: Interface menu

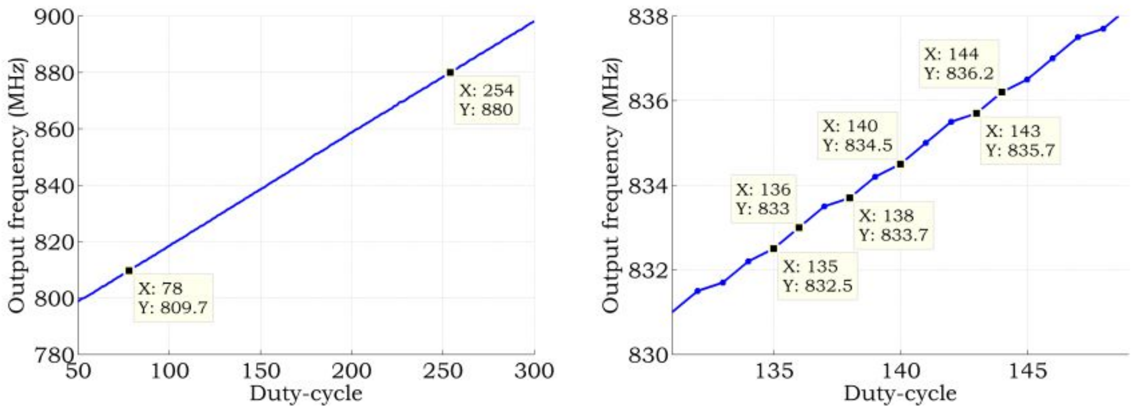
5.3.1 LO sweep range

The control of the VCO and subsequently the frequency tuning, is done via a digital PWM, varying the duty-cycle. To do this is necessary to obtain the digital duty-cycle references which turns to be the frequencies references to convert data to human domain.

The applied duty-cycle on the VCO was manually swept step by step, and registered the output frequency values with a benchtop spectrum analyser (**HP 8593E**). So The PWM signal generated by the **Arduino** board was tuned from 0 to 100% in a 10 bit base which allows 1024 discrete duty-cycle level steps.

The acquired results of interest are plotted in figure 5.5a, which represents the necessary sweeping range for the LO stage (from 810 to 880 MHz). This turns to be the frequency reference for the digital control and future calibrations require to recalculate the last referred graphic. Furthermore the registered output power of the VCO RF signal was grossly around 16.5 dBm.

In figure 5.5b is plotted in detail from figure 5.5a, 18 consecutive duty-cycle points (from 132 to 149) which will be used for comparison in next subsection.



(a) Tuning range of interest (b) Detail of 18 consecutive points

Figure 5.5: VCO output frequency over duty-cycle control

5.3.2 Interpolation algorithm

Interpolation are methods to extract an unknown value in the middle of two prior known points. In this project, linear interpolation is used to overcome limitations in **Arduino's** memory. As result the number of point for the lookup tables may be reduced to fit in the available programming resources.

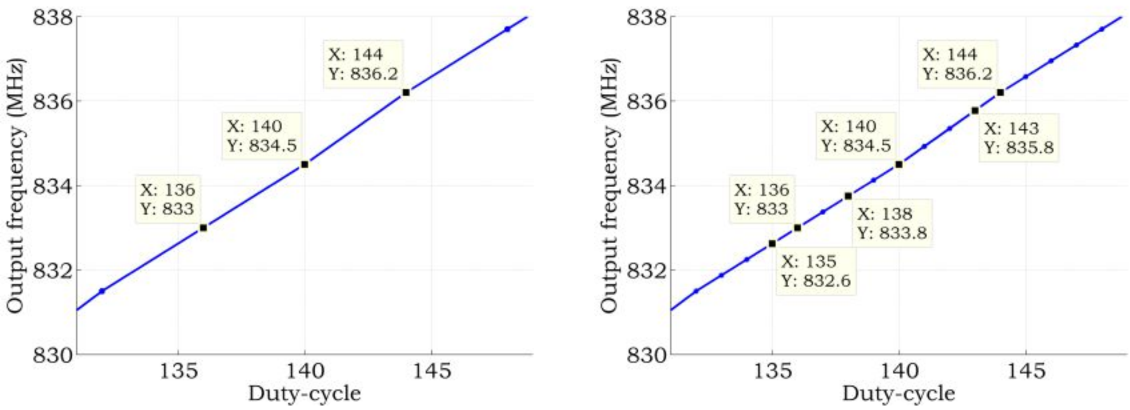
There are different orders of interpolation. Higher orders may introduce lower estimation errors, but require more computing resources. Any how, the first order or linear interpolation may be enough accurate for the requisites of this project.

Linear interpolation is mathematically expressed as in equation 5.5. Where y is the unknown and pretended value, x is the acquired value and y_a, x_a, y_b, x_b are the neighbours stored values.

$$y = y_a + (y_b - y_a)\frac{x - x_a}{x_b - x_a} \tag{5.5}$$

To validate this method and ensure the obtained error is acceptable, the duty-cycle table from figure 5.5a was compressed, discarding 3 in every 4 data points. The results of this may be observed in figure 5.6a, where the 5 data points between 132 and 149 duty-cycle values are represented.

Finally, in image 5.6b is the restored duty-cycle table from the previously compressed. It is possible to note from figure 5.5b to figure 5.6b where stands the reconstructed values, the error is inferior to 1 MHz. As consequence the compressed duty-cycle table is suitable to be used as frequency reference.



(a) Compressed duty-cycle table sample (5 data points)

(b) Restored duty-cycle table sample (18 data points)

Figure 5.6: Linear interpolation results

5.3.3 Amplitude correction factor

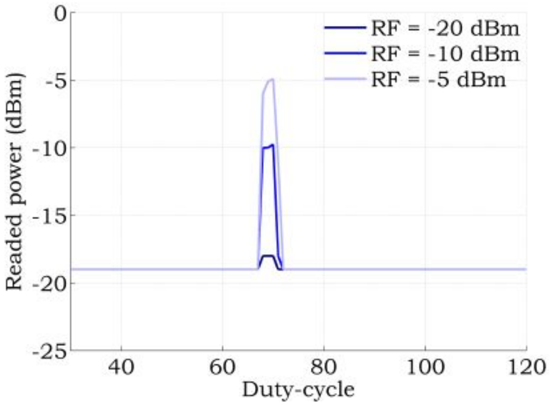
Since a RF signal reaches the first component in the circuit, till it is delivered into the power detector, various gains and losses happen. The three RBW channels have different power budgets due to different signal paths attenuation, as cables for example, and mainly for different filters characteristics. This means the signal power transformation from the RF input to the power detector is different, and to select one RBW is necessary to use an appropriate correction factor.

To calculate the practical correction coefficient it were taken in consideration the power budgets calculated in figure 5.2 for each RBW channel.

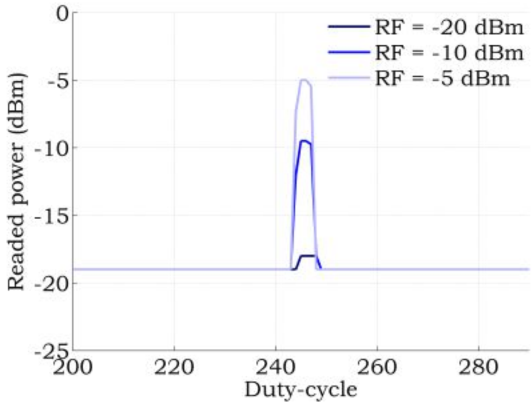
Then is was used a software routine which allowed manually control the duty-cycle, while the DC voltage from the power detector was converted to dBm, where the correction factor could be adjusted.

The read power levels for each tuning step were registered for a RF references of 880 and 950 MHz. The signal power was adjust for three trials as -20, -10 and -5 dBm. This process had been followed for the 3 RBW channels.

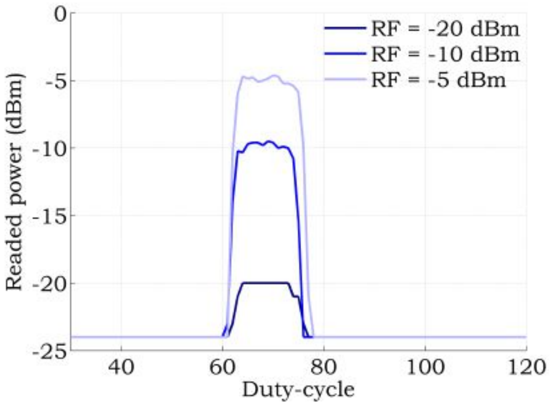
It is plausible now to have coherent power display over frequency as illustrated in figures 5.7. To remember the frequency will be addressed from the duty-cycle expressed in the X axis. Also is possible to note an amplitude accuracy of $\pm 2dBm$ related with the RF generator reference.



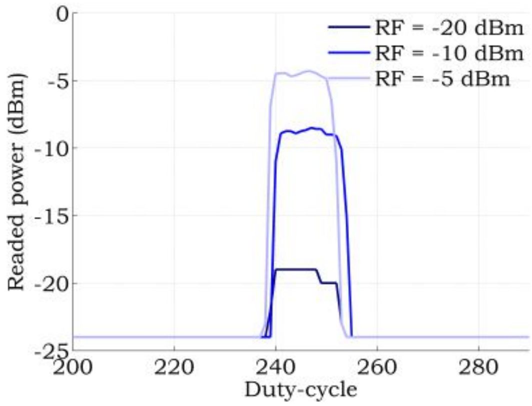
(a) RF in: 880 MHz, (1 MHz RBW)



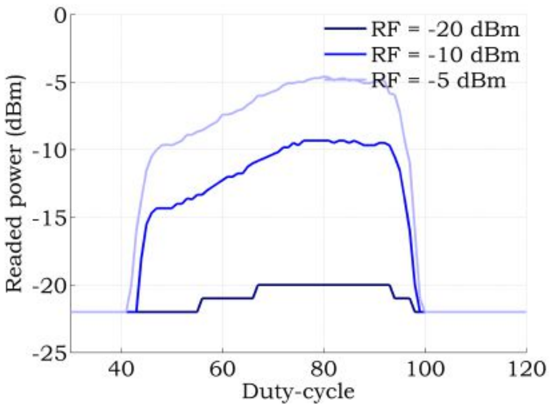
(b) RF in: 950 MHz, (1 MHz RBW)



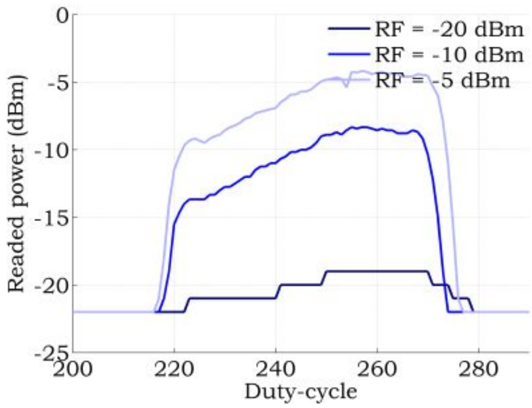
(c) RF in: 880 MHz, (5 MHz RBW)



(d) RF in: 950 MHz, (5 MHz RBW)



(e) RF in: 880 MHz, (20 MHz RBW)



(f) RF in: 950 MHz, (20 MHz RBW)

Figure 5.7: Amplitude over frequency addressed by duty-cycle manually controlled

5.3.4 Linear display of n data points

As an initial specification, the prototype should allow to the user to select a specific frequency range to spectral analysis. This leads to a variable and unpredictable number of duty-cycles points to sweep, which results in the number of points do be displayed in the TFT screen.

For the data display in this project, 104×120 pixels of the digital screen are reserved. The frequency is displayed in the horizontal axis which brings 120 pixels addressed by x, to plot the detected input RF power.

As so, the user may set the desired frequency range from start to stop. Those frequencies are converted to duty-cycles values, where $dutyMin$ and $dutyMax$ correspond to the minimum and maximum frequencies respectively. In the code flow it is created an array containing 120 values of duty-cycles.

The x value ($0 \leqslant x \leqslant 119$) is horizontally pointing to the pixel where the amplitude data will be displayed. For each x point, the value in the $dutyTable[x]$ is loaded into the **Arduino** PWM function.

$$\frac{dutyMax - dutyMin}{120} = dutyIncrement \tag{5.6}$$

$$\begin{gathered} dutyTable[0] = dutyMin \\ dutyTable[1] = dutyTable[0] + dutyIncrement \\ ... \\ dutyTable[119] = dutyTable[118] + dutyIncrement \Rightarrow dutyMax \end{gathered}$$

While programming special attention should relay with the appropriate data types to create the correct $dutyTable[x]$ vector. The $dutyIncrement$ must be a float type to first create the whole vector values with decimal precision, and only then convert it to integer type to apply it as a PWM modulation parameter.

5.4 Power over frequency measurements

To conclude the frequency spectrum analysis tests and validate the prototype as functional device, is presented now power over frequency practical measurements.

First was necessary to have a suitable frequency reference to compare with the experimental results. For that, with a benchtop spectrum analyser (**HP 8593E**) was registered the measurements of the output of a RF function generator (**HP E4433B**).

The function generator power was set to -20, -15, -10 and -5 dBm and frequency was swept from 875 to 955 MHz with 5 MHz increments. Some of the measurements with the benchtop spectrum analyser are certified in table 5.1 and a RBW of 1 MHz was used, for proximity with the prototype resources.

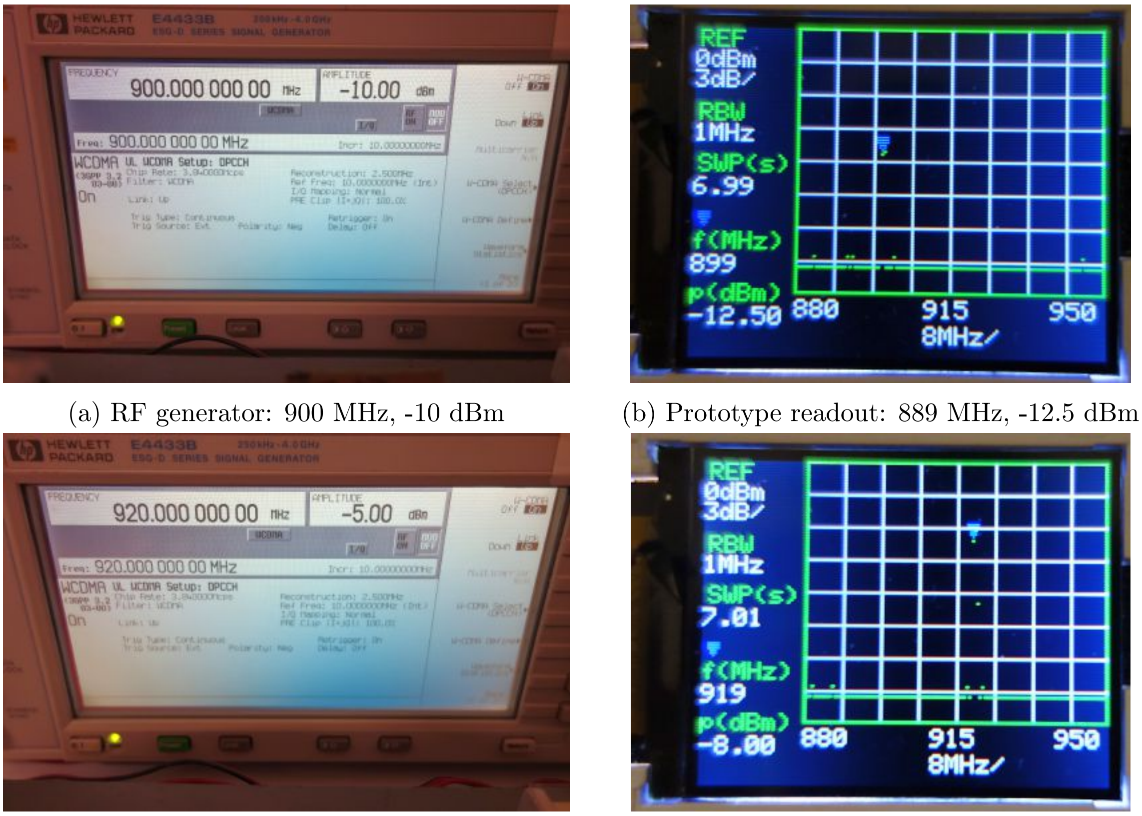
From the same RF function generator the output signal was injected into the prototype. Screen shots are pictured in 5.8, 5.9 and H.1.

In 5.8a and 5.8c is present the signal generator screen with the details of the configuration with the RF output on, and no modulation. In the other hand figures 5.8b and 5.8d demonstrate the screen visualisation of the prototype. To note marker function is on, and below the

Generator output		❖	Spectrum analyser readout	
Frequency	Power		Frequency	Power
895 MHz	-10 dBm		895.42 MHz	-11.19 dBm
900 MHz	-10 dBm		900.37 MHz	-11.15 dBm
905 MHz	-10 dBm		905.32 MHz	-11.20 dBm
		...		
915 MHz	-5 dBm		915.45 MHz	-6.36 dBm
920 MHz	-5 dBm		920.4 MHz	-6.35 dBm
920 MHz	-5 dBm		925.35 MHz	-6.39 dBm

Table 5.1: Frequency references

blue triangle (which represents the marker position), on the bottom left side of the screen, are the readouts.



(a) RF generator: 900 MHz, -10 dBm

(b) Prototype readout: 889 MHz, -12.5 dBm

(c) RF generator: 920 MHz, -5 dBm

(d) Prototype readout: 919 MHz, -8 dBm

Figure 5.8: Practical measurements, 1 tone

Another important test to validate the prototype is the modulation detection. A two tone signal must be applied in the RF input of the prototype, and the screen should reveal the spectral content. The used RF generator may feature such test signal (figure 5.9a) and the screen shot of the displayed data is shwon in figure 5.9b, where it is possible to notice the two spectral components raising above the noise floor.

(a) RF generator: two tone

(b) Prototype two tone readout

Figure 5.9: Practical measurements, 2 tone

At last in appendix H stands additional screen shots of the prototype display. The frequency range was adjusted and sweeping times setted to specific values. Also different resolutions bandwidths where tested.

Chapter 6

Conclusions

To accomplish such active instrument it was necessary a familiarization with subjects involved in the spectrum analysis process. Starting with a revision of signals and physical interpretation of a spectrum, time and frequency measurement domains were compared. A natural focus turned for what is necessary to perform spectrum analysis and how to achieve it, and so, RF electronic concepts to validate components were introduced.

Different receiver architectures behaviours were compared and the opted decisions were explained. The components functionality that fitted in the project requirements were detailed for choosing latter the physical devices.

The validation process of the used electronic circuits was described alongside with practical measurements. The result of the individual circuits programmed working as one, was described focusing on the practical implementation of the prototype.

Essential functions were implemented as the to selection of the frequency range under analysis, a selectable RBW and adjustment on sweeping times. A marker tool was implemented to identify frequencies and power levels accurately.

The main objective on this project was accomplished with the physical spectrum analyser prototype implemented and with the power over frequency measurements validation.

About the circuit performance it could be looked in detail to the dBm power measurement validations. It is possible to note a higher loss in relation to the frequency references in table 5.1 or even during the measurements tests results on figure 5.7 to figures 5.8b and 5.8d.

The correction coefficients used to the final measurements are the same as in the latter tests. Despite that, throughout final tests, the power level showed to be sporadically more accurate. This may be related with the evaluation method where circuits are connect with means of screwing coaxial connectors, or even bad soldering joints. Once useful for characterization and validation they may turn into signal leaks and reflections causing insertion loss to change with great ease.

It should be mentioned the filter shape of the 20 MHz resolution bandwidth is severe affected in the first 10 MHz. This may be the reflex of all frequency responses of previous components.

The **Arduino** board had shown to be practical and with enough computing capability to run such device.

The lack of space to store all data points present in graphic 5.5a, or for the latter conversion of the DC voltage from the power detector to a dBm scale, was surmounted with linear interpolation.

The TFT color display for sure brings more possibilities for data display as well a more comfortable environment for spectral analysis.

As an **Arduino** open source project, the code is entirely available and compilable.

Yet, this project as a handled spectrum analyser still have lot to improve as the occupied space, a necessary rugged out case and batteries project to feed the entire circuit.

6.1 Future Work

This prototype may be the base for research on this measurement tools, turning to be a point where much more is possible to test.

The incorporation of all circuits in one PCB circuit should be considered. This could lead for reduced sizes and reduce insertion losses in signal path.

It can be considered the introduction of a selectable attenuator, to allow the system to handle higher power levels and a respective image rejection filter in the RF stage. Also it is possible to consider the project of an antenna to collect electromagnetic signals from free space.

The batteries project should be incorporated in this design, to achive a properly handled spectrum analyser.

Introduce an extra gain amplifier in the IF section would lead a higher signal recovery from the attenuation associated with the RBW selection process. In the same IF section it could be tried the use of other IF center frequency higher than 70 MHz. This would bring a wider analysable frequency range. Also this could apply for filter project and so this spectrum analyser prototype could be a research source gathering different areas.

Appendices

Appendix A

Decibel (dB) and decibel to milliwatt (dBm)

In radio-frequency and other areas where such subject like wave propagation is drawn (acoustic science for example), power quantities are involved and dB and dBm tend to simplify the comparison about signals energy. It turns to be particular useful when power levels are too dissimilar to fit in a linear unit scale.

As a logarithm function this dB units can represent power levels in a more handy way, helping to quantify power gain or attenuation over a signal.

Looking first to the dB concept, it may be understood as a power ratio. Considering two power levels P_1 and P_2 in watt, if they have the same value, the result of equation A.1 is $0dB$. A positive result from the same equation indicates P_1 is greater than P_2 and that could represent a power gain. A negative dB quantity informs that P_2 is greater than P_1 and for instance represent a power loss.

$$dB = 10log\left(\frac{P_1}{P_2}\right) \tag{A.1}$$

In case of dBm, it is a conversion from power level in watts to a logarithmic scale, related to one milliwatt. A signal which has 0 dBm has 1 milliwatt.

$$dBm = 10log\left(\frac{P_1}{0.001W}\right) \tag{A.2}$$

Furthermore, dB can be added to dBm to indicate for example, the power level at the output of an amplifier or subtracted when the power level of signal is attenuated throughout a component.

Appendix B

S-parameters

Scattering parameters also known as S-Parameters provide practical scalar transmission measurements over a two port network device. In RF circuits it is necessary to transmit energy effectively over the network and the use of this parameters simplify the characterization of high frequency designed circuits[61, 65].

Figure B.1: Two port network diagram

About waves propagation it should be mentioned the phenomena of incident, reflected and transmitted energy. This happens whenever the medium where the wave is propagating abruptly changes. In case of electromagnetic waves spreading through electric circuits they find different impedances introduced by devices input and output impedance.

Figure B.1 describes a generic DUT with an input Port A and an output Port B. For example, when a electromagnetic wave reaches Port A, some of the energy will be absorbed by the DUT mentioned as a_1 and another portion is reflected as a_2 suggests. Furthermore, the absorbed energy will be transmitted as b_2 allusion.

In the way around electromagnetic energy that reaches Port B as b_1 will be transmitted to Port A joining a_2 and also reflected in Port B as represented by b_2.

S-parameters are related with the incident waves in both ports and expressed under the matrix equation B.1 [61, 65].

$$\begin{bmatrix} a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \tag{B.1}$$

Arranging in order to the S-parameters it comes:

$$S_{11} = \left.\frac{a_2}{a_1}\right|_{b_1=0} \quad S_{21} = \left.\frac{b_2}{a_1}\right|_{b_1=0} \quad S_{12} = \left.\frac{a_2}{b_1}\right|_{a_1=0} \quad S_{22} = \left.\frac{b_2}{b_1}\right|_{a_1=0} \tag{B.2}$$

Consequently S_{11} represents the reflected power in Port A, S_{12} mean the transmitted power from Port B to Port A, S_{21} designate the power transfer from Port A to port B and finally S_{22} denote the reflected power in Port B. All these quantities are expressed in dB.

Figure B.2 represents the same generic DUT as in figure B.1 but with the S-parameters depiction. They are denoted as S_{mn} representing the power transfer from port n to port m.

Figure B.2: Two port network diagram with S-Parameters representation

Network analysers are measure equipments that can compute S-parameters[66].

In RF Laboratories at IT there is available a microwave network analyser, the **E8361C** from Agilent Technologies. This device can perform an automatic full set of S-Parameters measures over a two port network. From now on, in this document, the S-parameters presented values where computed with the aid of this device.

With S_{11} it is possible to examine the impedance matching between devices and VSWR. In addition, S_{21} describe the component's gain also designated as insertion gain or quantify the introduced attenuation commonly called insertion loss.

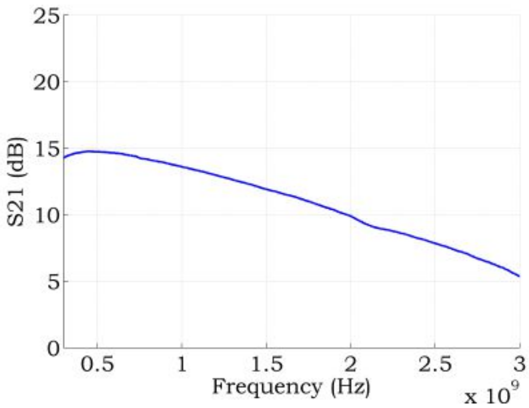
The S-parameters representation can be either positive or negative according its representation. For example, if S_{21} is a positive number it means that device produces a signal gain. As opposition, if S_{21} is negative it will inform that device attenuates the input signal.

From the four S-parameters S_{11} and S_{21} have greater importance in the description of this thesis work.

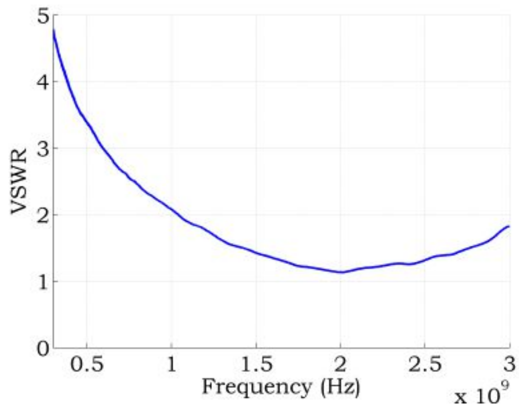
Appendix C

Amplifier HMC374 tests results

Additional test results for the LNA **HMC374** in figure C.1.



(a) Gain from 300MHz to 3GHz



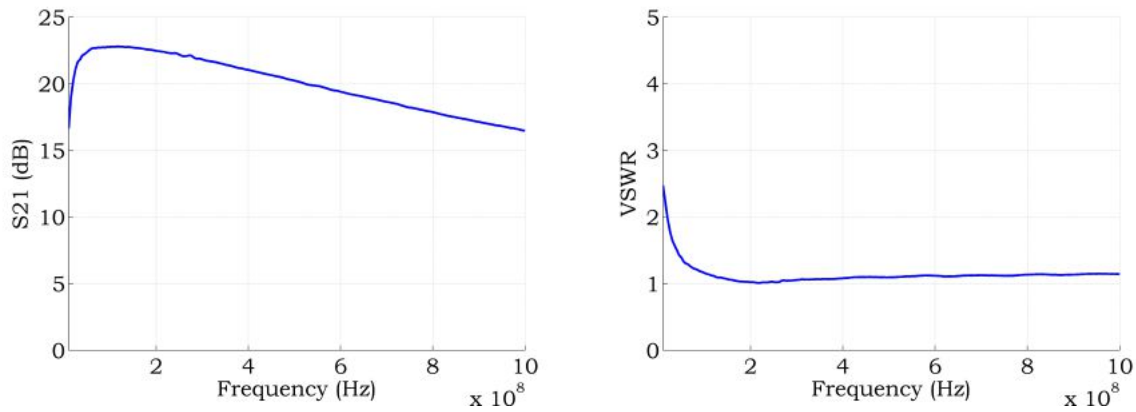
(b) VSWR from 300MHz to 3GHz

Figure C.1: Low noise amplifier **HMC374** gain and VSWR

Appendix D

Amplifier HMC580ST89 tests results

Additional test results for the power amplifier **HMC580ST89** in figure D.1.



(a) Gain from 10MHz to 1GHz

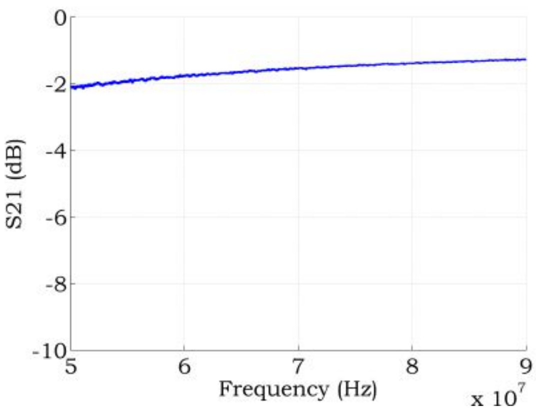
(b) VSWR from 10 MHz to 1 GHz

Figure D.1: Power gain and VSWR from **HMC580ST89**

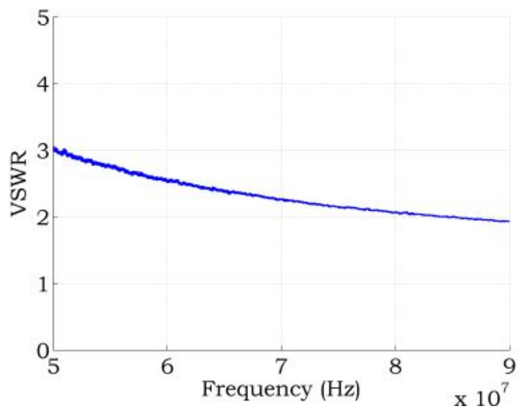
Appendix E

RF switch HMC252QS24 tests results

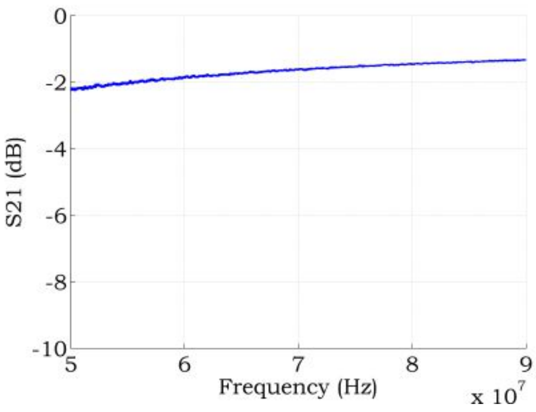
Additional test results for the RF switch **HMC252QS24** in figure E.1.



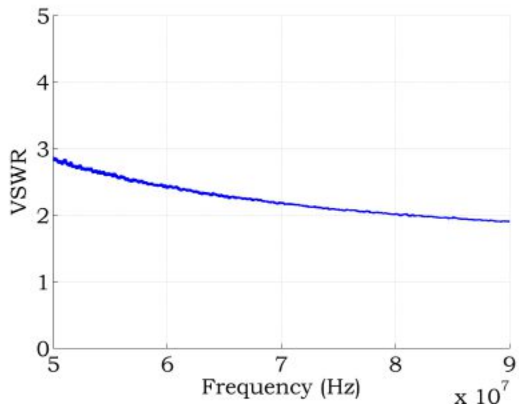
(a) Channel 2 insertion loss



(b) Channel 2 VSWR



(c) Channel 3 insertion loss



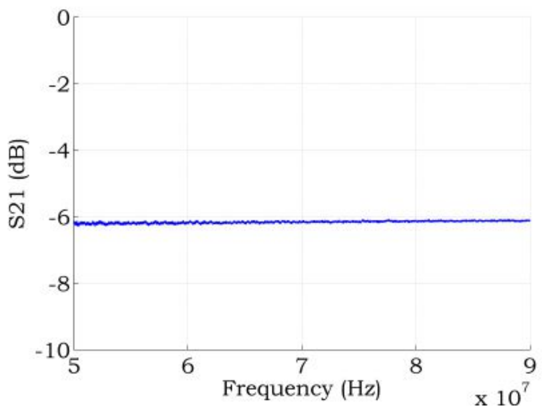
(d) Channel 3 VSWR

Figure E.1: RF switch **HMC252QS24** insertion loss and VSWR

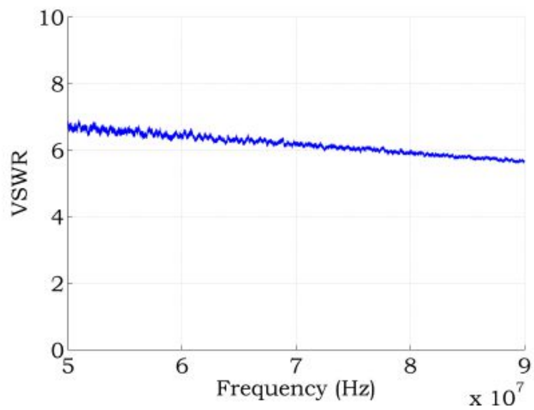
Appendix F

Power splitter ZSC-4-1+ test results

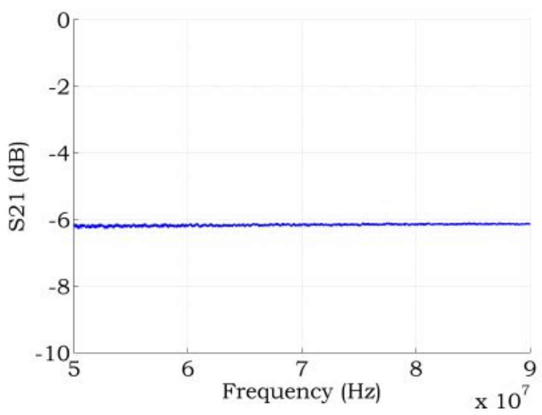
Additional test results for the power splitter **ZSC-4-1+**. Figure F.1 contains S21 and VSWR pratical results over channel 2 and 3.



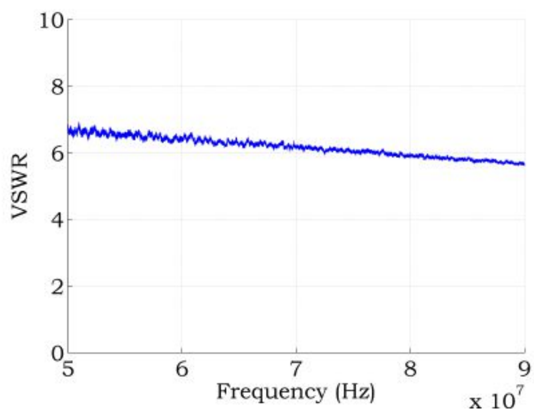
(a) Channel 2 insertion loss



(b) Channel 2 VSWR



(c) Channel 3 insertion loss



(d) Channel 3 VSWR

Figure F.1: Power combiner insertion loss and VSWR for channel 2 and 3

Appendix G

Keypad calculations

Voltage values read by the **Arduino** ADC. Calculus for each key pressed.

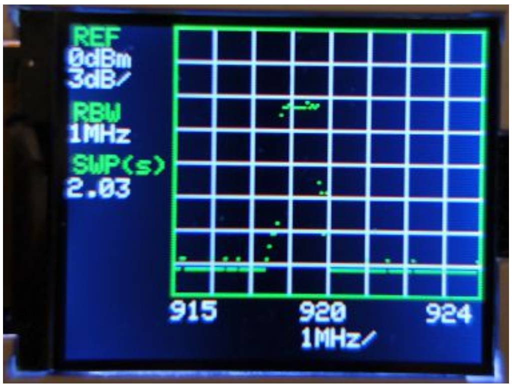
$$V_{DK} = 5V, R_1 = 332\Omega, R_2 = 3240\Omega, R_3 = 80600\Omega$$
$$R_4 = 12100\Omega, R_5 = 2150\Omega, R_6 = 2150\Omega, R_7 = 2150\Omega$$
$$A = V_{DK}\left(\frac{R_1}{(R_4 + R_5 + R_6 + R_7} + 1\right)^{-1} = 4.91V$$
$$B = V_{DK}\left(\frac{R_1 + R_2}{R_4 + R_5 + R_6 + R_7} + 1\right)^{-1} = 4.19V$$
$$C = V_{DK}\left(\frac{R_1 + R_2 + R_3}{R_4 + R_5 + R_6 + R_7)} + 1\right)^{-1} = 0.90V$$
$$D = V_{DK}\left(\frac{R_1}{R_5 5 + R_6 + R_7} + 1\right)^{-1} = 4.75V$$
$$E = V_{DK}\left(\frac{R_1 + R_2}{R_5 + R_6 + R_7} + 1\right)^{-1} = 3.21V$$
$$F = V_{DK}\left(\frac{R_1 + R_2 + R_3}{R_5 + R_6 + R_7} + 1\right)^{-1} = 0.35V$$
$$G = V_{DK}\left(\frac{R_1}{R_6 + R_7} + 1\right)^{-1} = 4.64V$$
$$H = V_{DK}\left(\frac{R_1 + R_2}{R_6 + R_7} + 1\right)^{-1} = 2.73V$$
$$I = V_{DK}\left(\frac{R_1 + R_2 + R_3}{R_6 + R_7} + 1\right)^{-1} = 0.24V$$
$$J = V_{DK}\left(\frac{R_1}{R_7} + 1\right)^{-1} = 4.33V$$
$$K = V_{DK}\left(\frac{R_1 + R_2}{R_7} + 1\right)^{-1} = 1.87V$$
$$L = V_{DK}\left(\frac{R_1 + R_2 + R_3}{R_7} + 1\right)^{-1} = 0.12V$$

Appendix H

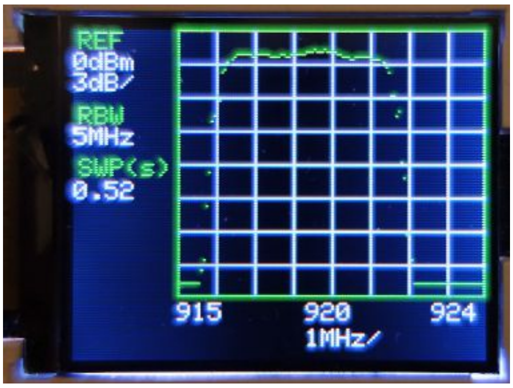
Prototype readouts

Additional screen shots from the prototype performing power over frequency measurements.

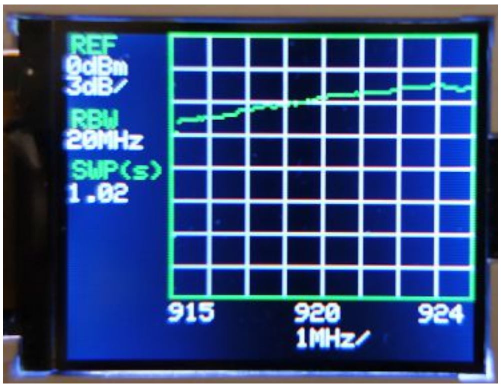
The two tone setup in figure 5.9a is being under analysis in figure H.1d.



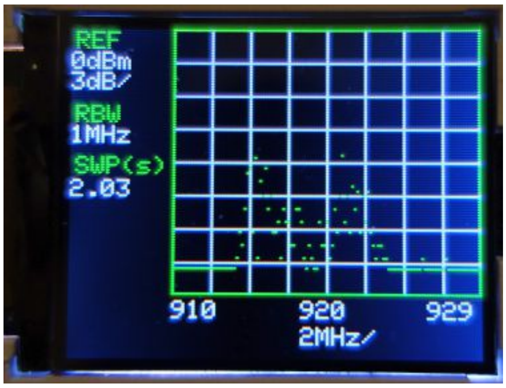
(a) RF in 920 MHz, -5 dBm; RBW 1 MHz, sweep time 2 seconds



(b) RF in 920 MHz, -5 dBm; RBW 5 MHz, sweep time 0.5 seconds



(c) RF in 920 MHz, -5 dBm; RBW 20 MHz, sweep time 1 seconds



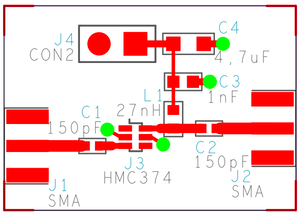
(d) RF in: two tone; RBW 1 MHz, sweep time 2 seconds

Figure H.1: Practical measurements

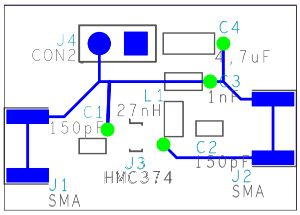
Appendix I

PCBs schematics

The used PCBs designs are presented here.

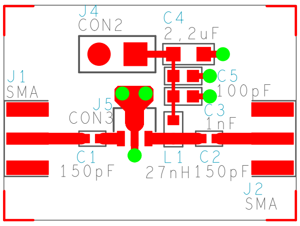


(a) Top layer

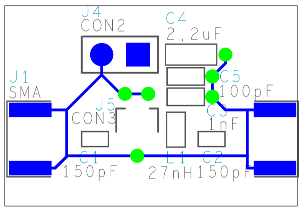


(b) Bottom layer - Ground plane

Figure I.1: LNA **HMC374** PCB schematic

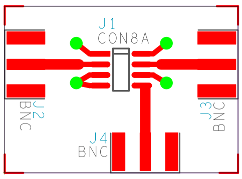


(a) Top layer

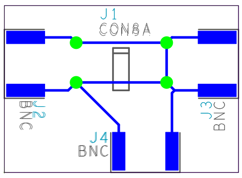


(b) Bottom layer - Ground plane

Figure I.2: Amplifier **HMC580ST89** PCB schematic

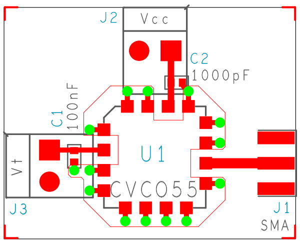


(a) Top layer

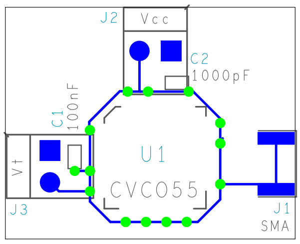


(b) Bottom layer - Ground plane

Figure I.3: Mixer **HMC207AS8** PCB schematic

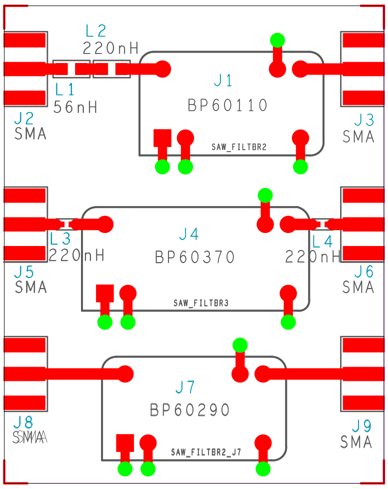


(a) Top layer

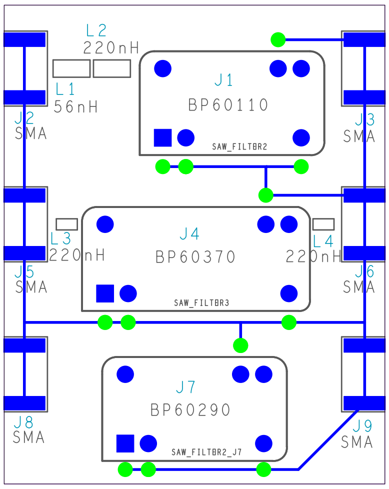


(b) Bottom layer - Ground plane

Figure I.4: VCO **CVCO55BE-0800-1600** PCB schematic

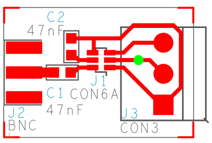


(a) Top layer

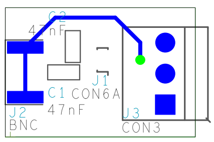


(b) Bottom layer - Ground plane

Figure I.5: Filter bank PCB schematic



(a) Top layer



(b) Bottom layer - Ground plane

Figure I.6: Power detector **LTC5507** PCB schematic

Appendix J

Arduino source code

```
#include <Flash.h>              // Flash library to store lookup tables
#include <TimerOne.h>           // TimerOne library to control timer1 (pin 10 and pin 9) as PWM generator
#include <SPI.h>                // SPI library to use adafruitdisplay
#include <Adafruit_ST7735.h> // Adafruit display graphics library
#include <Adafruit_GFX.h>     // Adafruit core graphics library
/*********************************************************/
#define TFT_CS   10            // Chip select line for TFT display
#define TFT_DC    7            // Data/command line for TFT
#define TFT_RST  NULL          // Reset line for TFT (or connect to +5V)
#define lcdBacklight  6        // LCD backlight controled with PWM

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

int ledPin = 4;                // presence led control pin
int pwmPin = 9;                // Timer1 pin - PWM generator
int sensorPin  = A5;           // Power sensor pin
int keyPadIn = A0;             // keypad sensor pin
int8_t resolutionBDWpins[2] = {3,2};  //Resolution bandiwth filter bank switch control pins resolutionBDWpins[0] == rf switch pin b; resolutionBDWpins[1] == rf switch pin a

boolean markerOp[2] = {0,0}; // Marker operator to control marker display and refresh. markerOp[0] == marker is on or off. markerOp[1] == marker need to be refreshed

int sensorValue;  //variable to read power detector
char k = '0'; //key detection char
int8_t RBW = 1; //variable to store the resolution bandwith in use

uint8_t freqSelect[4] = {0,0,0,0}; //array to process acquired data to select frequency range
int16_t freqMinLimit = 880, freqMaxLimit = 951; //absulete minimum and maximum range frequencies
int16_t dutyMin, dutyMax, duty; //variaables to control duty cycle
int16_t freqMin = freqMinLimit, freqMax = freqMaxLimit; //actual input frequency to display average power
int8_t menu_top = 14, menu_select; // Postition of first menu item from top of screen

```

```
int16_t x = 1, i = 0, length_rect = 120, width_rect = 104; //x is the initial horizontal position to display data pixels. length_rect and width_rect are the reserved space in screen for data plot
int16_t origin_x_rect = 40, origin_y_rect = 0; //screen references for display the reserved space do data plot
int16_t vLinesSpacing = 15, hLinesSpacing = 13; // references to plot grid in data plot
int16_t origin_y_pixel = origin_y_rect + width_rect - 1; //references to plot data inside the reserved space to data plot
uint16_t dutyXarray[120]; //array containning the duty-cycles values for each horizontal position in data plot
int16_t displayedVals[120]; //array containg information about the displayed values for posterior calculs (used by marker function)
int16_t xMarker = 1, xMarkerHold = xMarker, yMarkerHold; //markers variables
uint32_t readWait, sweepTime = 500; //readWait (in microsecond), sweepTime (in milliseconds): 500 milliseconds, 1000 milliseconds, 2000 milliseconds, 5000 milliseconds; readWait = sweeptime/(dutyMax-dutyMin) * 1000

float sensordBm; //
int8_t powerBudget[3] = {2,1,1}; //for channel 1,2 and 3 from rf switch
int8_t ampCorrection = powerBudget[0];

//look up table for duty-cycles
FLASH_ARRAY(int16_t, dutyTable, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104,
            108, 112, 116, 120, 124, 128, 132, 136, 140, 144, 148, 152, 156, 160, 164, 168,
            172, 176, 180, 184, 188, 192, 196, 200, 204, 208, 212, 216, 220, 224, 228, 232,
            236, 240, 244, 248, 252, 256, 260, 264, 268, 272, 276, 280, 284, 288, 292, 296, 300);

//look up table for frequency
FLASH_ARRAY(int16_t, freqTable,  7998, 8010, 8027, 8042, 8057, 8075, 8090, 8105, 8122, 8137, 8152,
            8170, 8185, 8202, 8217, 8235, 8250, 8265, 8282, 8297, 8315, 8330, 8345, 8362, 8377, 8392,
            8410, 8427, 8442, 8457, 8475, 8492, 8510, 8522, 8540, 8555, 8572, 8587, 8602, 8620, 8635,
            8652, 8665, 8682, 8697, 8715, 8730, 8745, 8762, 8777, 8792, 8807, 8825, 8840, 8855, 8870,
            8887, 8902, 8920, 8935, 8950, 8967, 8982);
//look up table for dBm conversion
FLASH_ARRAY(int16_t, adcValue, 138, 139, 140, 141, 142, 143, 144, 146, 149, 152, 156,
            160, 166, 172, 181, 191, 203, 216, 233, 251, 264, 274, 286, 299, 314, 327,
            337, 349, 362, 377, 394, 413, 434, 459, 486, 517, 552, 591, 635, 684, 740,
            804, 880, 1016); //to convert to dbm


int main(void)
{
  init();
  {
    pinMode(TFT_CS, OUTPUT); //set pin for tft as output
```

```
    pinMode(resolutionBDWpins[0],OUTPUT); //set pin for filter selection as output
    pinMode(resolutionBDWpins[1],OUTPUT); //set pin for filter selection as output
    digitalWrite(resolutionBDWpins[0],LOW); //activate filter selection
    digitalWrite(resolutionBDWpins[1],LOW); //activate filter selection
    pinMode(ledPin,OUTPUT); //set pin for presence led
    digitalWrite(ledPin,HIGH); //activate presence led
    tft.initR(INITR_REDTAB);   // initialize a ST7735R chip, red tab
    tft.setRotation(3);               // Set to landscape mode
    analogWrite(lcdBacklight, 255);   // Turn Backlight on full
    tftInitialDisplay(); //welcome screen display
    dutyMin = getDuty(freqMinLimit); //calculate absolute minimum duty-cycle
    dutyMax = getDuty(freqMaxLimit); //calculate absolute maximum duty-cycle
    videoSetup(); //prepare settings for correct data plot
    displayGrid(); //display grid in the data plot reserved space
    displayInfo(); //display information on screen
    Timer1.initialize(500); //initialize timer to control pwm
  }
  while(1)
  {
    uint32_t timeReadControl[2], timeSweepControl[2] = {0,0}; //actual time acquire
    if (x >= length_rect) //resest timer and x pointer for a new sweep
    {
      Timer1.initialize(500);
      x = 0;
    }
    timeSweepControl[0] = millis(); //time acquire for sweep control
    timeReadControl[0] = micros();  //time acquire for tuning setps control
    do
    {  //one full sweep display
        /*LOoOp*/
        Timer1.pwm(pwmPin,dutyXarray[x]); //load pwm function with the respective value
        timeReadControl[1] = micros(); //time acquire for tuning step control
        if (timeReadControl[1]-timeReadControl[0] >= readWait) //check for the time to execute a new read out
        {
            sensorValue = analogRead(sensorPin); //read the RF power detector output
            timeReadControl[0] = timeReadControl[1]; //reload the time till new comparison
            sensordBm = getDbm(sensorValue) + ampCorrection; //dBm amplitude conversion and correction
            uint16_t y = 104 + (sensordBm * (float)4.7); //framming the vertical power level to data display
            displayPixel(x,displayedVals[x],true); //clean old data display
            displayPixel(x,y,false); // uptade new data display
            displayedVals[x] = y; // store displayed amplitude value
            x = x++; //x increment for the new horizontal position
         }
         k = keyDetect(); //chek for user instructions (menu call or left/right marker instruction)
         if (k == 'j') // User had called the menu
         {
            menuCall(); //display options menu
```

```
            timeSweepControl[0] = millis(); // reset control time
        }
        if (markerOp[0] == 1) //if marker is set to on
        {
              callMarker(k); //update marker values
        }



    }while(x <= length_rect); //this loop is active till x reaches the end of the data sweep, then it starts a new sweeping
    timeSweepControl[1] = millis(); //time sweepcontrol for display
    tft.setCursor(0,59); //display sweep time
    tft.setTextColor(ST7735_WHITE, ST7735_BLACK); //display sweep time
    tft.print((timeSweepControl[1] - timeSweepControl[0])/(float)1000); // display sweep time
  }
}

/*****************************************************/
float getDbm(int16_t sensVal) //getDbm function receives the readed value from the rf power detector and converts it to dBm value by interpolation
{
  int16_t tablePosition = 0; //reference for the lookup table
  int16_t dbmLow, dbmHigh, dbmDelta, adcValueLow, adcValueHigh, adcValueDelta; //auxiliar variables for interpolation
  int16_t dbmMin = -22; //miminum usable value
  float dbmNew = 0;

  for (tablePosition; tablePosition <= adcValue.count(); tablePosition++) //finds values in lookup table
  {
    if ( sensVal <= adcValue[tablePosition])
    {
    break;
    }
    dbmMin = dbmMin++;
  }
  dbmLow = dbmMin - 1;
  dbmHigh = dbmMin;
   adcValueLow = adcValue[tablePosition-1];
   delay(1);
   adcValueHigh = adcValue[tablePosition];
   delay(1);
   dbmDelta = dbmHigh - dbmLow;
   adcValueDelta = adcValueHigh - adcValueLow;
   dbmNew =  dbmLow + dbmDelta*((sensVal - adcValueLow)/(float)adcValueDelta);
   return dbmNew;
}
/*****************************************************/
void displayMarkerInfo() //displayMarkerInfo function display information about the frequency and amplitude pointed by the marker
{
  int16_t freqMarker = freqMin;
  freqMarker = freqMarker + (xMarker/120.00)*(freqMax - freqMin);
  tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
  tft.setCursor(0,89);
```

```
  tft.print(freqMarker);
  if (freqMarker < 999)
   tft.print(" ");
  tft.setCursor(0,109);
  tft.print((float)(displayedVals[xMarker]-104)/4);
  if (((displayedVals[xMarker]-104)/4) > -10)
   tft.print(" ");
  /*if (displayedVals[xMarker] < 99)
   tft.print(" ");*/

}
/*************************************************************************/
void callMarker(char option) //callMarker function handles the marker positioning in the data display. The input value option is the pressed key to move the marker to left or right
{
     switch (option)
      {
         case 'd': //user instruction to move marker to left
         {
             if (xMarker <= 0) //if marker position is already in the lower position it keeps it there.
                 xMarker = 1;
             else
              {
                displayMarker(xMarkerHold,yMarkerHold,true); //delete old marker in screen
                xMarker--; //decrement marker position
              }
         }
         break;
         case 'f': //user instruction to move marker to right
         {
             if (xMarker >= length_rect)//if marker position is already in the top position it keeps it there.
              xMarker = length_rect-1;
             else
              {
                displayMarker(xMarkerHold,yMarkerHold,true); //delete old marker in screen
                xMarker++; //increment marker position
              }
         }break;
         default: //if no user instruction is set while marker is on, the marker should be refreshed to the mew amplitude values
         {
           if(xMarker != xMarkerHold || displayedVals[xMarker] != yMarkerHold)
           {
           displayMarker(xMarkerHold,yMarkerHold,true); //delete old marker in screen
           displayMarker(xMarker,displayedVals[xMarker],false);  //display new marker in screen from the stored values pointed by xMarker
           xMarkerHold = xMarker;
           yMarkerHold = displayedVals[xMarker];
           displayGrid();
           }
         }
```

```
        break;
      }

      displayMarkerInfo();
    }

/**************************************************************************/
void displayMarker(int16_t x0, int16_t y0, boolean del) //displayMarker function prints in screen the marker
{
  int16_t x1 = origin_x_rect + x0 - 2, y1 =  origin_y_pixel - y0 - 7, x2 = origin_x_rect + x0 + 2, y2 = origin_y_pixel - y0 - 7; //marker positioning limits
  if (del == true) //delete old marker
  {
    tft.fillTriangle(x0 + origin_x_rect, origin_y_pixel - y0 - 2, x1, y1, x2, y2, ST7735_BLACK);
  }
  else //print marker
  tft.fillTriangle(x0 + origin_x_rect, origin_y_pixel - y0 - 2, x1, y1, x2, y2, ST7735_BLUE);
}
/**************************************************************************/
void displayGrid() //displayGrid function forms the grid and box limit of the dataplot area
{
   tft.drawRect(origin_x_rect, origin_y_rect , length_rect, width_rect, ST7735_GREEN);
   for(int16_t i=origin_x_rect + vLinesSpacing; i <= tft.width() - vLinesSpacing; i+=vLinesSpacing)
   {
    tft.drawFastVLine(i, origin_y_rect +1, width_rect-2, 0xE71C);
   }
  for(int16_t i=origin_y_rect+hLinesSpacing; i < width_rect; i+=hLinesSpacing)
   {
    tft.drawFastHLine(origin_x_rect +1, i,length_rect,0xE71C);
   }
}
/****************************************************/
void displayPixel(int16_t xx, int16_t yy, boolean del) //displayPixel function plots the information in the data plot area
{
   if (del == true) //delete old pixel position
    {
        if ( xx == 0 || yy == 0 || xx == length_rect -1 || yy == width_rect -1) //cheks for position validation
         tft.drawPixel(origin_x_rect + xx, origin_y_pixel - yy, ST7735_GREEN); //if old pixel is in the border replace it with green
        else if ((xx) % vLinesSpacing == 0 || (yy+1)% hLinesSpacing == 0)
        {
          tft.drawPixel(origin_x_rect + xx , origin_y_pixel - yy, 0xE71C); //if old pixel is over a grid line replaces it for the grid color
        }
      else
        tft.drawPixel(origin_x_rect + xx , origin_y_pixel - yy, ST7735_BLACK); //if old pixel is over a black area replaces it for the black color
    }
```

```
    else
    tft.drawPixel(origin_x_rect + xx, origin_y_pixel - yy, ST7735_GREEN); // prints a green pixel with valid information
}
/*****************************************************/
void videoSetup() //videoSetup function prepares necessary variables to correct data display
{
  float stepSize;
  if (dutyMax-dutyMin == 0) //if selected start and stop frequencies are equal it grants a space between than.
  {
    dutyMax = dutyMin + 1; //increments duty cycle
  }
  else
  {
    stepSize = (dutyMax - dutyMin) / ((float)length_rect); //each pixel will contain info from nDuty points
  }
  dutyXarray[0] = dutyMin; //sets the minimum applyable duty cycle
  float aux = dutyXarray[0];
  for(uint8_t i = 1; i<=length_rect;i++) //creates the duty-cycle table addressed by horizontal x position
  {
    aux =  aux + stepSize;
    dutyXarray[i] =  aux ;
  }
   readWait = ((sweepTime) / (float)(length_rect))*1000; //establishes the reading times to grant the selected sweping times

}
/************************************************/
void menuCall() //menuCall function prints the options menu when called
{
      menu_select=1;// Select 1st menu item
      do
        {
          String menuMain[] = {" Spec Menu:", " Frequency Options", " Resolution Bandwidth"," Sweep Time"," Marker"," Return"};
          uint8_t numMenuMain = (sizeof(menuMain)/sizeof(String))-1;
          tftMenuInit(menuMain, numMenuMain);                          // Draw menu function
          switch(menu_select) //String menuMain[] = {"Spec Menu:", "Frequency Options", "Resolution Bandwd.","Span","Search Peak","Marker","Info"};
            {
            case 1: //frequency options
            {
              menu_select=1;
              do
              {
                String menuSetRange[] = {" Frequency Options"," Start Frequency"," Stop Frequency"};
                uint8_t numMenuSetRange =(sizeof(menuSetRange)/sizeof(String))-1;
                tftMenuInit(menuSetRange, numMenuSetRange);
                switch(menu_select)
```

```
          {
            case 1: //start frequency
            {
              dataAquire("Start Frequency (MHz):",freqSelect , freqMinLimit);
              dutyMin = getDuty(freqMin); //updates duty-cycle after user instruction for miminum frequency
              dutyMax = getDuty(freqMax); //updates duty-cycle after user instruction for maximum frequency
            }
            break;
            case 2: //stop freqeuncy
            {
              dataAquire("Stop Frequency (MHz):",freqSelect, freqMaxLimit);
              dutyMin = getDuty(freqMin);//updates duty-cycle after user instruction for miminum frequency
              dutyMax = getDuty(freqMax);//updates duty-cycle after user instruction for maximum frequency
            }
            break;
           }
        }while(menu_select != -1);
        menu_select = 1;
      }
      break;
      case 2: //Resolution bandwidth
      {
        String menuSetBand[] = {" Select RBW"," 1MHz"," 5MHz"," 20MHz"};
        uint8_t numMenuSetBand =(sizeof(menuSetBand)/sizeof(String))-1;
        menu_select = 1;
        tftMenuInit(menuSetBand, numMenuSetBand);
        if (menu_select != -1)
        {
          switch(menu_select)
          {
            case 1:
            {
              digitalWrite(resolutionBDWpins[0], LOW); //rf switch pin b to LOW
              digitalWrite(resolutionBDWpins[1], LOW); //rf switch pin a to LOW
              ampCorrection = powerBudget[0]; //select the appropriate amplitude correction factor for the selected RBW channel
             RBW = 1; //updates the selected channel
              ledBlink(); //led blinks to indicate succes of operation
            }
            break;
            case 2:
            {
              digitalWrite(resolutionBDWpins[0], LOW); //rf switch pin b to LOW
              digitalWrite(resolutionBDWpins[1], HIGH);  //rf switch pin a to HIGH
              ampCorrection = powerBudget[1];//select the appropriate amplitude correction factor for the selected RBW channel
             RBW = 5;//updates the selected channel
```

```
                        ledBlink();//led blinks to indicate succes of operation
                      }
                      break;
                      case 3:
                      {
                        digitalWrite(resolutionBDWpins[0], HIGH); //rf switch pin b
                             to LOW
                        digitalWrite(resolutionBDWpins[1], LOW);//rf switch pin a
                            to HIGH
                        ampCorrection = powerBudget[2];//select the appropriate
                            amplitude correction factor for the selected RBW channel
                        RBW = 20;//updates the selected channel
                        ledBlink();//led blinks to indicate succes of operation
                      }
                      break;
                    }
                  }
                  menu_select = 2;
                }
                break;
                case 3: // sweping time adjustment
                {
                  String menuSetSpan[] = {" Select Sweep Time", " 0.5 sec", " 1 sec
                      ", " 2  sec"," 5 sec"};

                  uint8_t numMenuSetSpan =(sizeof(menuSetSpan)/sizeof(String))-1;
                  menu_select = 1;
                  tftMenuInit(menuSetSpan, numMenuSetSpan);
                  if (menu_select != -1)
                  {
                    switch(menu_select)
                    {
                    case 1:
                    {
                        sweepTime = 500; //adjust sweping time
                        ledBlink(); //led blinks to indicate succes of operation
                    }
                    break;
                    case 2:
                    {
                      sweepTime = 1000;//adjust sweping time
                      ledBlink(); //led blinks to indicate succes of operation
                    }
                    break;
                    case 3:
                    {
                      sweepTime = 2000;//adjust sweping time
                      ledBlink();//led blinks to indicate succes of operation
                    }
                    break;
                    case 4:
                    {
                      sweepTime = 5000;//adjust sweping time
                      ledBlink();//led blinks to indicate succes of operation
                    }
                    break;
                  }
```

```
              menu_select = 3;
            }
          }
          break;
          case 4: // marker
          {
            String menuSetMarker[] = {" Select marker", " On", " Off"};
            uint8_t numMenuSetMarker =(sizeof(menuSetMarker)/sizeof(String))-1;
            menu_select = 1;
            tftMenuInit(menuSetMarker, numMenuSetMarker);
            switch(menu_select)
            {
              case 1: //turn marker on
              {
                if (markerOp[0] != 1){
                markerOp[0] = 1;//turn marker on
               }
                ledBlink();
              }
              break;
              case 2: //turn marker off
              {
                if (markerOp[0] != 0)
                {
                markerOp[0] = 0;//turn marker off
                }
                ledBlink();
              }
              break;
            }

          }
          break;
          case 5:
          {
            ledBlink();
            menu_select = -1;
          }
          break;
        }
    } while(menu_select != -1);
    displayInfo();
    videoSetup();
    displayGrid();
  }
/*****************************************************/
int16_t getDuty(int16_t freq) //getDuty function receives the pointed frequency to convert it to duty-cycle by interpolation
{
  freq = (freq-74)*10; //frequency calibration
  int16_t tablePosition = 0; //reference for lookup table
  int16_t freqLow, freqHigh, freqDelta, dutyLow, dutyHigh, dutyDelta;  //auxiliar variables for interpolation
  float dutyNew = 0; //new duty cycle value

```

```
  for (tablePosition; tablePosition <= freqTable.count(); tablePosition++)  // looks for values in look up tabble
  {
    if  ( freq <= freqTable[tablePosition])
    {
    break;
    }
  }
  //interpolation calculos
  freqLow = freqTable[tablePosition -1];
  delay(1);
  freqHigh = freqTable[tablePosition];
  delay(1);
  freqDelta = freqHigh - freqLow;
  dutyLow = dutyTable[tablePosition  -1];
  delay(1);
  dutyHigh = dutyTable[tablePosition];
  delay(1);
  dutyDelta = dutyHigh - dutyLow;

  dutyNew =  dutyLow + dutyDelta*((freq - freqLow)/(float)freqDelta); // conversion for a discrete duty-cycle value

  ledBlink();
  return (int)dutyNew;
}
/*****************************************************************/
int16_t getFreq(int16_t dutyx) //getFreq function receives the pointed duty-cycle to convert it to frequency by interpolation
{
  int16_t tablePosition = 0; //reference for lookup table
  int16_t freqLow, freqHigh, freqDelta, dutyLow, dutyHigh, dutyDelta; // auxiliar variables for interpolation
  float freqNew = 0; //newfrequency value

  for (tablePosition; tablePosition <= dutyTable.count(); tablePosition++) // looks for values in lookup table
  {
    if  ( dutyx <= dutyTable[tablePosition])
    {
    break;
    }
  }
  //interpolation calculus
   freqLow = freqTable[tablePosition -1];
   delay(1);
   freqHigh = freqTable[tablePosition];
   delay(1);
   freqDelta = freqHigh - freqLow;
   dutyLow = dutyTable[tablePosition  -1];
   delay(1);
   dutyHigh = dutyTable[tablePosition];
   delay(1);
   dutyDelta = dutyHigh - dutyLow;
   freqNew =  freqLow + freqDelta*((dutyx - dutyLow)/(float)dutyDelta);
   freqNew = freqNew/10;
   return (int)freqNew;
```

```
}
/*****************************************************/
void dataAquire(char str[], uint8_t arr[], int maxMin) //dataAquire function handles the user input to select minimum or maximum frequencies
{
  //receives a string to prin (str[]), an int array to store input data(arr[]), and a maxMin flag to identify min or max freq validation
  char c;
  int8_t i =0;
  boolean done = false; //flag to detect if operation had finished
  tft.fillScreen(ST7735_BLACK);
  tft.setCursor(0,0);
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
  tft.println(str);
  tft.drawFastHLine(0, 9, tft.width()-1, ST7735_GREEN);
  tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
  //print the acquired data so far
  while( done == false )
  {
  tft.setCursor(0,menu_top);
  switch (i)
  {
    case 0:
    {
      tft.print("_   ");
    }
    break;
    case 1:
    {
      tft.print(arr[0],DEC);
      tft.print("_  ");
    }
    break;
    case 2:
    {
      tft.print(arr[0],DEC);
      tft.print(arr[1],DEC);
      tft.print("_ ");
    }
    break;
    case 3:
    {
      tft.print(arr[0],DEC);
      tft.print(arr[1],DEC);
      tft.print(arr[2],DEC);
      tft.print("_");
    }
    break;
    case 4:
    {
      tft.print(arr[0],DEC);
      tft.print(arr[1],DEC);
      tft.print(arr[2],DEC);
      tft.print(arr[3],DEC);
     }
    break;
  }
```

```

  do //wait for user instruction
  {
    c = keyDetect();
    delay(150);
  } while(c=='0');

  //deals with user instruction
  if (c != '0')
  {
    c = numericKeypadHandle(c);
    switch (c)
      {
        case 'e': //when user finish to insert data it presses enter and saves the values
        {
          dataValidation(i, arr, maxMin); //cheks for validation of data
          done = true;
        }
        break;
        case 'b': //if user wants to delete the last entered value for correction
        {
          if (i<=0)
            {
            i=0;
            arr[i]=0;
            done=true;
            }
          else
            {
              i--;
              arr[i]=0;
            }
        }
        break;
        case 'o':
        {
          i=i;
        }
        break;
        default:
        {
          arr[i] = c -'0';
          i++;
        }
      }
    }
  if (i >4)
    i=4;
  }
  tft.fillScreen(ST7735_BLACK);
}
/***********************************************/
void dataValidation(int quantity, uint8_t arr[], int maxMin) //dataValidation validates the acquired data from user to start and stop frequencies. It recveives the number of insert digits, the array wich keeps the inserted
```

```
     data and maxMin is a flag for validation of the start or stop frequency
{

  int aux = 0;

   if (quantity != 0) //checks if data array array contains information
   {
     if (arr[aux] == 0)
     {
       while(arr[aux] == 0)
       {
           for(int j = 0; j <= sizeof(arr); j++)
           {
             arr[j] = arr[j+1];
           }
         quantity--;
         aux++;
       }
     }
     aux =1 ;
     for(int i =0; i < quantity; i++) //converts the data in array to a integer value to procees
     {
      aux = arr[i] * pow(10,quantity - 1 - i) + aux;
     }
   }
   else //cheks for limit conditions for frequencies
     {
       aux = 0;
     }
     if (maxMin == freqMinLimit && aux < freqMinLimit)
      freqMin = freqMinLimit;
     if(maxMin == freqMinLimit && aux >= freqMinLimit && aux <= freqMax)
      freqMin = aux;
     if (maxMin == freqMinLimit && aux > freqMax &&  aux < freqMaxLimit)
     {
       freqMin = aux;
       freqMax = aux;
       freqChangeWarning("Stop frequency");
     }
     if (maxMin == freqMinLimit && aux > freqMaxLimit)
     {
      freqMin = freqMaxLimit;
      freqMax = freqMaxLimit;
      freqChangeWarning("Stop frequency");
     }
     if (maxMin == freqMaxLimit && aux >= freqMaxLimit)
      freqMax = freqMaxLimit;
     if(maxMin == freqMaxLimit && aux <= freqMaxLimit && aux > freqMin)
      freqMax = aux;
     if (maxMin == freqMaxLimit && aux <= freqMin && aux >= freqMinLimit)
     {
      freqMin = aux;
      freqMax = aux;
      freqChangeWarning("Start frequency");
     }
     if (maxMin == freqMaxLimit && aux < freqMinLimit)
```

```
      {
        freqMin = freqMinLimit;
        freqMax = freqMinLimit;
        freqChangeWarning("Start frequency");
      }

    for(int8_t i = 0; i <= (sizeof(arr)/sizeof(uint8_t)) - 1; i++)
    {
    arr[i] = 0;
    }
 }
/*****************************************************/
void freqChangeWarning(String str) //freqChangeWarning allerts the user when start frequency is higher than stop frequency, and as consequece the stop frequency must be adjusted or vice versa
{
       tft.setCursor(0,50);
       tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
       tft.println(str);
       tft.print("has changed");
       delay(1500);
}
/*****************************************************/
void tftMenuInit(String menu[], int8_t numMenu) //tftMenuInit function handles the menu printings
{
  // Clear screen and display the menu
  int8_t i;
  tft.fillScreen(ST7735_BLACK);
  boolean done = false;
  do{
  tft.setTextWrap(false);
  tft.setTextSize(1);
  tft.setCursor(0, 0);
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
  tft.println(menu[0]);

  tft.drawFastHLine(0, 9, tft.width()-1, ST7735_GREEN);

  tft.setTextColor(ST7735_WHITE, ST7735_BLACK);

  for(i=1;i<=numMenu;i++)
  {
      if (menu_select == i)
      {
        tft.setTextColor(ST7735_BLACK, ST7735_GREEN);
      }
      else
      {
        tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
      }
      tft.setCursor(0, ((i-1)*10)+menu_top);
      tft.println(menu[i]);
  }

  do
  {
```

```
        k = keyDetect();
        delay(100);
    } while (k == '0');
      switch(k)
      {
        case 'l':
        {
          //Select (enter was pressed)
          done = true;
        }
        break;
        case 'b':
        {
          // Up
          // move up one menu item, if at top wrap to bottom
          if (menu_select <= 1 )
          {
          menu_select = numMenu; //set to last position
          }
          else
          {
          menu_select--;
          }
        }
        break;
        case 'h':
        {
              if (menu_select >= numMenu)
              {
                menu_select = 1;
              }
              else
              {
                menu_select++;
              }
        }
        break;
        case 'j':
        {
        menu_select = -1;
        done = true;
        }
        break;
      }
    }while(!done);
    tft.fillScreen(ST7735_BLACK);
}
/*************************************************************/
void displayInfo() //displayInfo function updates the data in the screen and
    information values for user interpretation
{
  int16_t  perDiv, space, freqCenter = freqMin + (freqMax - freqMin) / 2;
  perDiv = (freqMax - freqMin) / 8;
  uint8_t x = 40, y = 106;

  //display info left side display
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
```

```
  tft.setCursor(2,1);
  tft.println("REF");
  tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
  tft.println("0dBm");
  tft.print("3dB/");
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
  tft.setCursor(2,30);
  tft.println("RBW");
  tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
  tft.print(RBW);
  tft.print("MHz");
  tft.setCursor(2,50);
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
  tft.print("SWP(s)");

  //display freqs under grid
  tft.setTextSize(1);
  tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
  tft.setCursor(x,y);
  tft.print(freqMin);
  tft.setCursor(x+50,y);
  tft.print(freqCenter);
  tft.setCursor(x+99,y);
  tft.print(freqMax-1);
  tft.setCursor(x+50,y+10);
  tft.print(perDiv);

  if (perDiv < 9)
  space = 6;
  else if (perDiv < 99)
  space = 12;
  else if (perDiv < 999)
  space = 18;

  tft.setCursor(x+50+space,y+10);
  tft.print("MHz/");


  //if marker is on displays the information about the marker
  if (markerOp[0] == 1)
  {
    tft.fillTriangle(4, 77, 2, 72, 6, 72, ST7735_BLUE);
    tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
    tft.setCursor(0,80);
    tft.print("f(MHz)");
    tft.setCursor(0,100);
    tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
    tft.print("p(dBm)");
  }
}
/***********************************************/
char keyDetect() //keyDetect function handles the input of the keypad
{
  char key;
  int16_t sensorKeypad = analogRead(keyPadIn); //read the value from the sensor
   if (sensorKeypad < 15)
     {
```

```
      key = '0';
        return key;
      }
      else{

        if ((sensorKeypad < 1010) && (sensorKeypad >1000))
        {
          key = 'a';
          return key;
        }
        if ((sensorKeypad < 870) && (sensorKeypad >850))
        {
          key = 'b';
          return key;
        }
        if ((sensorKeypad < 190) && (sensorKeypad >170))
        {
          key = 'c';
          return key;
        }
        if ((sensorKeypad < 990) && (sensorKeypad >960))
        {
          key = 'd';
          return key;
        }
        if ((sensorKeypad < 670) && (sensorKeypad >640))
         {
          key = 'e';
        return key;
         }
        if ((sensorKeypad < 80) && (sensorKeypad >60))
         {
          key = 'f';
     return key;}

        if ((sensorKeypad < 960) && (sensorKeypad >930))
        {
          key = 'g';
         return key;
        }
         if ((sensorKeypad < 570) && (sensorKeypad >540))
      {
          key = 'h';
        return key;
        }
        if ((sensorKeypad <55) && (sensorKeypad >40))
        {
          key = 'i';
        return key;
        }
         if ((sensorKeypad < 895) && (sensorKeypad >870))

          {
            key = 'j';
            return key;
          }
        if ((sensorKeypad < 390) && (sensorKeypad >370))
```

```
          {
          key = 'k';
          return key;
          }
        if ((sensorKeypad <30)  && (sensorKeypad >15))
          {
          key = 'l';
        return key;
        }
      key = '0';
      return key;
      }
}
/***********************************************/
char numericKeypadHandle(char input) //numericKeypadHandle function handle the
    output of the keypad to convert it to decimal values for numeric input
{
  char output;
  switch (input)
  {
    case 'a':
    {
      output = '1';
    }
    break;
    case 'b':
    {
      output = '2';
    }
    break;
    case 'c':
    {
      output = '3';
    }
    break;
    case 'd':
    {
      output = '4';
    }
    break;
    case 'e':
    {
    output = '5';
    }
    break;
    case 'f':
    {
      output = '6';
    }
    break;
    case 'g':
    {
      output = '7';
    }
    break;
    case 'h':
    {
```

```
        output = '8';
      }
      break;
      case 'i':
      {
        output = '9';
      }
      break;
      case 'j':
      {
        output = 'b';
      }
      break;
      case 'k':
      {
        output = '0';
      }
      break;
      case 'l':
      {
        output = 'e';
      }
      break;
      default:
      {
      output = 'o';
      }
  }
  return output;
}
/************************************************/
void tftInitialDisplay() //tftInitialDisplay function displays welcome menu for the spectrum analyser
{
  tft.fillScreen(ST7735_BLACK);      // Fill screen with black
  tft.setCursor(15,10);
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
  tft.println("Universidade de Aveiro");
  tft.setCursor(45,20);
  tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
  tft.println("Masters 2013");
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
  tft.setTextSize(3);
  tft.setCursor(10,35);
  tft.print("Spectrum");
  tft.setCursor(10,55);
  tft.println("Analyzer");
  tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
  tft.setTextSize(1);
  tft.setCursor(35,90);
  tft.println("880 - 950 MHz");
  tft.setCursor(30,100);
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
  tft.println("All freqs in MHz");
  uint16_t i = millis(), ii;
  do
  {
```

```
      k = keyDetect();
      ii = millis();
   }while (k == '0' && (ii - i <= 2500));
   tft.fillScreen(ST7735_BLACK);        // Fill screen with black

}
/*********************************************/
void ledBlink(void) //ledBlink function turns the presence led on and off in a
      blink pattern
{
   for(int8_t i = 0; i <=10; i++)
   {
      digitalWrite(ledPin,LOW);
      delay(10);
      digitalWrite(ledPin,HIGH);
   }
}
```

Bibliography

[1] Nuno Borges de Carvalho. (2013) Conceitos Fundamentais de Rádio Comunicações. Universidade de Aveiro

[2] Adel S. Sedra, Kenneth C. Smith. (2011) *Microelectronic Circuits.* Oxford University Press

[3] Agilent Thechnologies. (2000) Application note 150, *Spectrum Analysis Basics* [Available] `http://cp.literature.agilent.com/litweb/pdf/5952-0292.pdf` [Accessed] 7 August 2013

[4] Rohde & Schwarz. (2008) *Fundamentals of Spectrum analysis.* [Available] `http://www.electron.frba.utn.edu.ar/~jcecconi/Bibliografia/05%20-%20Analizador%20de%20Espectro/Fundamentals_of_Spectrum_Analysis_Rauscher2008.pdf` [Accessed] 5 September 2013

[5] Agilent Thechnologies. (2001) Application note 150-1, *Spectrum Analysis Amplitude and Frequency Modulation* [Available] `http://literature.agilent.com/litweb/pdf/5954-9130.pdf` [Accessed] 7 August 2013

[6] ANACOM. (2013) *Quadro Nacional de Atribuição de frequencias, 2010-2011, versão 2* [Available] `http://www.anacom.pt/render.jsp?contentId=1172786` [Accessed] 13 September 2013

[7] Behzad Razavi. (1998) *RF Microelectronics.* Prentice Hall PTR

[8] R. E. Ham. (2001) Microwave Measurements, *The RF and Microwave Handbook*, CRC Press LLC. [Available] `http://vega.unitbv.ro/~olteanu/Tehnici%20de%20masurare%20in%20tc/RF%20Measurment.pdf` [Accessed] 29 August 2013

[9] Agilent Technologies (2012) *Spectrum Analyzer and Signal Analyzer Selection Guide* [Availble] `http://literature.agilent.com/litweb/pdf/5968-3413E.pdf` [Accessed] 30 August 2013

[10] João Carreira. (2010) *Hardware implementation of a spectrum anlyzer based in SDR* [Available] `http://ria.ua.pt/bitstream/10773/4395/1/tese_joao_carreira.pdf` [Accessed] 22 August 2013 Master's Thesis, Universidade Aveiro

[11] Wade Lowdermilk. (2009) Wide Spectral Span Spectrum Analysis with an Analog Step and Dwell Translation Pre-Processor to a High Dynamic Range FFT Based Spectrum Analyzer, *IEEE Conference Publications*

[12] Mark Rodwell. (2013) *Introduction to Receivers.* [Available] http://www.ece.ucsb.edu/Faculty/rodwell/Classes/ece218b/notes/Introduction_to_Receivers_w11.pdf [Accessed] 2 September 2013. University of California

[13] Tektronix. (2009) *Fundamentals of Real-Time Spectrum Analysis* [Available] www.tektronix.com/rsa [Accessed] 9 August 2013

[14] Lioe de Xing, Sushaidi Shafie, Izhal Abdul Halin and S. S. Jamuar. (2009) Design of Front End of a RF Receiver, *Proceedings od 2009 IEEE Student Conference on Research and Development*

[15] WolframAlpha Website. (2013) *Fourier transform.* [Online] http://www.wolframalpha.com/input/?i=Fourier+transform&lk=1&a=ClashPrefs_*MathWorld.FourierTransform- [Acessed] 5 September 2013

[16] WolframAlpha Website. (2013) *discrete Fourier transform.* [Online] http://www.wolframalpha.com/input/?i=discrete+Fourier+transform&lk=1&a=ClashPrefs_*MathWorld.DiscreteFourierTransform- [Acessed] 5 September 2013

[17] WolframAlpha Website. (2013) *fast Fourier transform.* [Online] http://www.wolframalpha.com/input/?i=fast+Fourier+transform&lk=1&a=ClashPrefs_*MathWorld.FastFourierTransform- [Acessed] 5 September 2013

[18] Matthew T. Hunter, Achilleas G. Kourtellis, Christopher D. Ziomek and Wasfy B. Mikhael. (2010) Fundamentals of Modern Spectral Analysis *IEEE Conference Publications*

[19] Matteo Bertocco, Andrea Dalla Chiara, Giovanni Gamba and Allessandro Sona. (2009) Experimental comparison of spectrum analyzer architectures in the diagnosis of RF interference phenomena, *IEEE - I2MTC 2009 - International Instrumentation and Measurement Technology Conference*

[20] Tektronix. (2007) *Fundamentals of Digital Phosphor^{TM}Technology in Real-Time Spectrum Analyzers* [Available] http://www.scopeshop.de/(S(alt0xq453fazx455zwhwnj45))/PRODUKTINFO/Spektrumanalyzer/File_2328/Fundamentals%20of%20Digital%20Phosphor%E2%84%A2%20Technology%20in%20Real-Time%20Spectrum%20Analyzers%20Primer_DPX_Primer_37W_19638_2.pdf [Accessed] 25 September 2013

[21] Anritsu. (2008) *Guide to Spectrum Analyzers.* [Available] http://downloadfile.anritsu.com/RefFiles/en-US/Services-Support/Downloads/Technical-Notes/White-Paper/GuideToSpectrumAnalyzers.pdf [Accessed] 19 Septembre 2013

[22] Joint Committee for Guides in Metrology. (2012) International vocabulary of metrology *Basic and general concepts and associated terms (VIM).* [Available] http://www.bipm.org/utils/common/documents/jcgm/JCGM_200_2012.pdf [Accessed] 16 September 2013 Jim Stiles. (2006) *Minimum Detectable Signal* [Available] http://www.ittc.ku.edu/~jstiles/622/handouts/Minimum%20Detectable%20Signal.pdf [Accessed] 24 August 2013 The University of Kansas

[23] The American Radio Relay League, Inc. (2013) *The ARRL Handbook for Radio Communications 2013 Edition.* ARRL, the national association for Amateur Radio

[24] Jim Stiles. (2006) *Minimum Detectable Signal* [Available] http://www.ittc.ku.edu/~jstiles/622/handouts/Minimum%20Detectable%20Signal.pdf [Accessed] 24 August 2013 The University of Kansas

[25] Devendra K. Misra. (2001) Analysis and Design *Radio-Frequency and Microwave Communication Circuits.* Wiley Inter-Science

[26] BesserNet Website. (2013) Reflectometer Calculator [Online] http://www.bessernet.com/Ereflecto/tutorialFrameset.htm [Accessed] 17 August 2013

[27] Amanowaga. (2013) Standing Wave Patters and VSWR, *Tutorial F* [Available] http://www.amanogawa.com/archive/docs/F-tutorial.pdf [Accessed] 17 August 2013

[28] The Physics Classroom. (2013) *Travelling Waves vs. Standing Waves* [Online] http://www.physicsclassroom.com/class/waves/u10l4a.cfm [Accessed] 16 August 2013

[29] Antenna Theory Website (2013) *Voltage Standing Wave* [Online] http://www.antenna-theory.com/definitions/vswr.php [Accessed] 17 August 2013

[30] Microwaves101 Website. (2013) Visualizing VSWR [Online] http://www.microwaves101.com/encyclopedia/vswr_visual.cfm [Accessed] 17 August 2013

[31] Jon B. Hagen. (1996) *Radio-frequency electronics: circuits and applications.* Cambridge University Press

[32] Agilent Technologies. (2010) Application Note 57-1, *Fundamentals of RF and Microwave Noise Figure Measurements.* [Available] http://literature.agilent.com/litweb/pdf/5952-8255E.pdf [Accessed] 21 August 2013

[33] Mini-Circuits. (2009) Understanding noise parameter measurements,*AN-60-040.* [Online] http://217.34.103.131/app/AN60-040.pdf [Accessed] 16 October 2013

[34] Joe Mitola (1995) The Software Radio Architecture, *IEEE Communications Magazine May 1995* [Available] http://www.freewebs.com/mabilloo/softwareRADarch.pdf [Accessed] 23 Septembre 2013

[35] Peter Berg. (2002) *Dual Conversion Receivers Are Better Than Single Conversion Receivers... Fact or Fiction??* [Available] http://www.bergent.net/SC-DC.pdf [Accessed] 11 October 2013

[36] Nuno Borges de Carvalho. (2013) Arquitecturas de Recepção/Transmissão. Universidade de Aveiro

[37] Gary Breed. (2011) High Frequency Electronics, *The Mathematics of Mixers: Basic Principles* [Available] http://www.highfrequencyelectronics.com/Archives/Jan11/HFE0111_Tutorial.pdf [Accessed] 6 August 2013

[38] Analog Devices. (2011) MT-080, *Mixers and Modulators* [Available] www.analog.com/static/imported-files/tutorials/MT-080.pdf [Acessed] 6 August 2013

[39] Jim Stiles. (2005) *Mixer Conversion Loss* [Available] http://www.ittc.ku.edu/~jstiles/622/handouts/Mixer%20Conversion%20Loss.pdf [Accessed] 6 August 2013 The University of Kansas

[40] Indium Corporation. (2012) Application note, *Soldering 101 - A Basic Overview.* [Available] www.indium.com/_dynamo/download.php?docid=27 [Accessed] 13 August 2013

[41] Mini-Circuits. (2010) Amplifier Terms Defined,*AN-60-038.* [Available] http://217.34.103.131/app/AN60-038.pdf [Accessed] 6 November 2013

[42] Hittite Microwave Corporation. (2013) Data sheet, *HMC374/374E.* [Available] http://www.hittite.com/content/documents/data_sheet/hmc374.pdf [Accessed] 3 October 2013

[43] Hittite Microwave Corporation. (2013) Data sheet, *HMC580ST/580ST89E.* [Available] https://www.hittite.com/content/documents/data_sheet/hmc580st89.pdf [Accessed] 3 October 2013

[44] Mini-Circuits. (2013) Data sheet, *ERA-5+.* [Available] http://217.34.103.131/pdfs/ERA-5+.pdf [Accessed] 3 October 2013

[45] Hittite Microwave Corporation. (2013) Data sheet, *HMC207AS9/2077AS8E.* [Available] http://www.hittite.com/content/documents/data_sheet/hmc207as8.pdf [Accessed] 3 October 2013

[46] Crystek Microwave. (2013) Data sheet, *CVCO55BE-0800-1600.* [Available] http://www.crystek.com/microwave/admin/webapps/welcome/files/vco/CVCO55BE-0800-1600.pdf [Accessed] 3 October 2013

[47] Crystek Microwave. (2013) Data sheet, *VCO Evaluation Boards.* [Available] http://www.crystek.com/microwave/spec-sheets/vcoeval/CEVAL-033_055.pdf [Accessed] 3 October 2013

[48] Vanlong Technology Co., Lta. (2013) Data sheet, *BP60370.* [Available] http://www.vanlong.com/download/BP60370.pdf [Accessed] 3 October 2013

[49] Vanlong Technology Co., Lta. (2013) Data sheet, *BP60290.* [Available] http://www.vanlong.com/download/BP60290.pdf [Accessed] 3 October 2013

[50] Vanlong Technology Co., Lta. (2013) Data sheet, *BP60110.* [Available] http://www.vanlong.com/download/BP60110.pdf [Accessed] 3 October 2013

[51] Hittite Microwave Corporation. (2013) Data sheet, *HMC252QS24/252QS24E.* [Available] http://www.hittite.com/content/documents/data_sheet/hmc252qs24.pdf [Accessed] 3 October 2013

[52] Mini-Circuits. (2013) Data sheet, *ZSC-4-1+.* [Available] http://www.minicircuits.com/pdfs/ZSC-4-1+.pdf [Accessed] 3 October 2013

[53] Linear Technology Corporation. (2013) Data sheet, *LTC5507.* [Available] http://www.linear.com/docs/2368 [Accessed] 3 October 2013

[54] Arduino Website. (2013) [Online] http://www.arduino.cc [Accessed] 7 August 2013

[55] The Open Source Initiative Website. (2013) [Online] http://opensource.org/osd [Accessed] 14 October 2013

[56] Wiring Website. (2013) [Online] http://wiring.org.co/ [Accessed] 14 October 2013

[57] Texas Instruments Incorporated. (2013) Data sheet, *µA7800 SERIRES*. [Available] http://www.ti.com/lit/ds/symlink/ua7805.pdf [Accessed] 13 October 2013

[58] Analog Devices, Inc. (2013) Data sheet, *AD623*. [Available] http://www.analog.com/static/imported-files/data_sheets/AD623.pdf [Accessed] 13 October 2013

[59] Microchip Technology, Inc. (2013) Data sheet, *MCP601/1R/2/3/4*. [Available] http://ww1.microchip.com/downloads/en/devicedoc/21314g.pdf [Accessed] 13 October 2013

[60] Hobbytronics. (2013) Arduino Colour TFT Shield http://www.hobbytronics.co.uk/arduino-tft-shield?keyword=tft%20arduino [Accessed] 7 November 2013

[61] Nuno Borges de Carvalho. (2011) Antenas e Guias de Onda, *Aula 3* Universidade Aveiro

[62] Agilent Thechnologies. (2004) Application note, *Network analyser Bascis* [Available] http://cp.literature.agilent.com/litweb/pdf/5965-7917E.pdf [Accessed] 15 August 2013

[63] Agilent Technologies. (2008) Application note, *Measurement of Return Loss and Transmission Loss or Gain Using the Agilent N9340B*. [Available] http://www.home.agilent.com/en/pd-1348735-pn-N9340B/handheld-spectrum-analyser-hsa-3-ghz [Accessed] 7 August 2013

[64] Tektronix. (1983) *Fundamentals of Spectrum analysis* [Available] http://72.29.85.210/~radioama/guides/Tektronix%20fundamentals%20of%20spectrum%20analysis.pdf [Accessed] 19 September 2013

[65] Pedro Miguel Correia (2004) Final Project Report, *Amplificador de Potência.* Instituto Superior de Engenharia de Lisboa [Available] http://www.deetc.isel.ipl.pt/sistemastele/Projecto/2003_2004/PA/Relat%C3%B3rio_Pedro_Correia.pdf [Accessed] 15 August 2013

[66] Tiiti Kellomáki. (2011) Crash Course on Network Analysers & How to choose Power. http://www.ele.tut.fi/teaching/ele-6100/luennot_cc/network_analyser_quick.pdf Tampere University Of Technology

[67] Nasa Website. (2013) *Electromagnetic Spectrum* [Online] http://imagine.gsfc.nasa.gov/docs/science/know_l1/emspectrum.html [Accessed] 31 August 2013

[68] Kevin Faison. (2006) *Understanding Noise Figures in Radio Receivers* [Online] http://www.eetimes.com/document.asp?doc_id=1272302 [Accessed] 24 August 2013