



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

GPS/Galileo/GLONASS Software Defined Signal Receiver

Pedro Manuel Lourenço Marques Ferreira

Dissertation submitted to obtain the Master Degree in
Electronics Engineering

Jury

Chairman: Prof. Carlos Alberto Ferreira Fernandes
Supervisor: Prof. Gonçalo Nuno Gomes Tavares
Co-supervisor: Prof. Moisés Simões Piedade
Member: Prof. Rui Manuel Rodrigues Rocha

July 2012

Acknowledgments

First of all, I would like express my gratitude to my advisors Professor Gonalo Tavares and Professor Moiss Piedade, for the support and guidance throughout the elaboration of this dissertation, and for the knowledge and experience brought to me during the last years. I sincerely thank you.

I would also like to thank Professor Rui Rocha for the support and advice given.

I would also like to thank Artyom Gavrilov, who maintains the GNSS-related site <http://gnss-sdr.com> for his assistance, availability and willingness to share inside knowledge.

Also, I would like to thank my colleagues and friends at Instituto Superior Tcnico for their friendship and companionship and for helping me to achieve my academic goals. You are the best!

I must thank my long-time friends from Linda-a-Velha for all the support, understanding and unconditional friendship in good and bad times. I can’t describe how much I’ve grown with you all. Thank you very much.

Finally, I dedicate this work to my mother Maria do Carmo Ferreira, for all her love and comprehension and for always believing in me, opening all the doors in life for me. To her I owe everything. Thank you mother.

Abstract

Since ancient times, mankind has tried to find its bearings by using milestones and stars. A new era has begun, however, thanks to satellite communication. To take advantage of current and future Global Navigation Satellite Systems (GNSSs), new receiving and processing devices need to be developed.

GNSSs are the latest radio navigation systems in widespread use. The United States Global Positioning System (GPS), the Russian GLONASS and the European Union Galileo positioning system form the core of satellite navigation systems.

Traditional GNSS receivers use hardware based digital signal processing. This approach usually provides high performance with the advantage of low power consumption, although the flexibility of the receiver functionality is limited. By using open architectures, component replacement can occur more easily than in a closed system, significantly reducing total ownership costs in the future. Through software defined GNSS receivers, however, the whole signal processing is defined in software. By using such a system, the receiver can be reconfigured depending on the application, providing the receiver with enhanced adaptive capabilities.

This work focusses on the hardware development of a GNSS software radio receiver to work at the L1 band, also known as the civilian use band. The key points of the system are its versatility and flexibility and high configurability, offering the highest performance and integration, both for the expert and for the less acquainted user, making it a very competitive solution.

The design approach is presented and all the hardware and software developed for this work is explained in detail. The system validation was successfully performed in the field with real environment constraints, proving its capabilities.

Keywords

Global Navigation Satellite Systems (GNSS), Software-defined radio (SDR), Ethernet, CPLD

Resumo

Desde a antiguidade que a humanidade tenta definir as suas localizações através do uso de marcos miliários e estrelas. No entanto, uma nova era começou, graças à comunicação por satélite. Para tirar proveito dos actuais e futuros Sistemas Globais de Navegação por Satélite (GNSSs) é necessário o desenvolvimento de novos dispositivos de recepção e processamento.

Os GNSSs são os mais recentes sistemas de navegação por rádio de uso generalizado. O *Global Positioning System* (GPS) dos Estados Unidos, o GLONASS russo e da União Europeia, o sistema de posicionamento Galileo da União Europeia, formam o núcleo dos sistemas de navegação por satélite actuais.

Os receptores GNSS tradicionais efectuem o processamento de sinal baseado em *hardware*. Esta abordagem geralmente garante alto desempenho com a vantagem do baixo consumo, embora a flexibilidade do receptor seja limitada. Contudo, a utilização de arquitecturas abertas permite uma maior facilidade na substituição de componentes do que em sistemas fechados, minimizando significativamente o custo total do sistema a longo prazo. Através de receptores GNSS definidos por software, todo o processamento de sinal é realizado por software. Utilizando este sistema, o receptor pode ser reconfigurado para aplicações específicas, proporcionando ao receptor uma grande capacidade de adaptação.

Este trabalho concentra-se no desenvolvimento do *hardware* de um receptor GNSS definido por software para funcionar na banda de frequências L1, também conhecida por banda de uso civil. Os pontos-chave do sistema são a sua versatilidade e flexibilidade dada pela sua alta configurabilidade, oferecendo o mais alto desempenho e integração, tanto para especialistas como para os utilizadores comuns, fazendo desta uma solução bastante competitiva.

A abordagem ao projecto é apresentada, e todo o *hardware* e *software* desenvolvido para este trabalho é explicado em detalhe. A validação do sistema foi realizada com sucesso no terreno com as limitações de ambiente real, provando as suas capacidades.

Palavras-chave

Sistemas Globais de Navegação por Satélite (GNSS), Receptor definido por software (SDR), Ethernet, CPLD

Table of Contents

Acknowledgments	i
Abstract	iii
Resumo	v
List of Figures	ix
List of Tables	xiii
List of Abbreviations and Acronyms	xv
1 Introduction	1
1.1 Purpose and motivation.....	1
1.2 Goals, requirements and tasks	2
1.3 Outline	2
2 GNSS Fundamentals and Systems Overview	3
2.1 Introduction.....	3
2.2 Satellite navigation principles.....	3
2.3 Global Navigation Satellite Systems.....	4
2.3.1 GPS.....	5
2.3.2 Galileo.....	7
2.3.3 GLONASS.....	8
2.3.4 Navigation Message	9
2.3.5 GPS, Galileo and GLONASS Interoperability.....	12
2.4 Summary	14
3 GNSS Samplers and SDR receivers	15
3.1 Basic GNSS receiver Architecture	15
3.2 Software	16
3.2.1 Software signal acquisition	16
3.2.2 Software signal tracking	18
3.3 SDR Receivers	19
3.4 GNSS Software Receiver.....	22
3.5 Summary	24
4 GNSS Front-end Design.....	25
4.1 Introduction.....	25

4.2	EtherGNSS v3.....	25
4.2.1	GNSS RF Receiver	26
4.2.2	CPLD	28
4.2.3	Processor Core	32
4.2.4	Ethernet Interface	36
4.2.5	Power Supply	37
4.2.6	Other features.....	38
4.2.7	Impedance matching concerns and PCB layer stack-up	38
4.3	Budget and Prototyping	39
4.4	Summary	42
5	User Interface.....	43
5.1	Introduction.....	43
5.2	Embedded Webserver.....	43
5.3	EtherGNSS v3 GNSS front-end client.....	45
5.4	Summary	48
6	Measurements and results.....	49
6.1	Introduction.....	49
6.2	Characterisation of the EtherGNSS v3 system.....	49
6.3	Complete EtherGNSS v3 acquisitions validation.....	53
6.4	Overall EtherGNSS v3 Features and Specifications	61
6.5	Summary	61
7	Conclusions and Future Work.....	63
8	References	65
	Appendix I –EPCOS B9415 SAW Filter transfer function	67

List of Figures

Figure 2.1 – 1D simple model navigation principle [3].....	3
Figure 2.2 – 1D simple model with two transmitter [3].-.....	4
Figure 2.3 – Four satellites are required to determine a position in 3-D space.	4
Figure 2.4 - DSSS with BPSK modulation (Adaptation from [5]).	6
Figure 2.5 – Spectra of GPS Signals in L1 [1].....	6
Figure 2.6 – Spectra of Galileo Signals in E2-L1-E1 [1].	8
Figure 2.7 – Spectra of GLONASS signals in L1 [1].	9
Figure 2.8 - GPS navigation data structure [2].....	10
Figure 2.9 - Galileo Navigation Message Structure [1].....	11
Figure 2.10 – All GNSS frequency bands [1].	14
Figure 3.1 – Architecture of a basic software GNSS receiver.	15
Figure 3.2 – Block diagram of a receiver channel.	16
Figure 3.3 - Parallel Code Phase Search algorithm [2].	17
Figure 3.4 - Generic digital receiver channel [6].....	18
Figure 3.5 - GPS Radio front-end Design [5].....	19
Figure 3.6 - Sundance SMT8036 functional scheme [11].....	20
Figure 3.7 - Schematic of the hardware implementation [12].	21
Figure 3.8 - ipexSR architecture [13].....	22
Figure 3.9 – SoftGNSS v3.0 receiver flow diagram [2].	23
Figure 4.1 – Purposed GNSS SDR sampler architecture design.	25
Figure 4.2 - Typical application circuit for MAX2769.	26
Figure 4.3 – PLL frequency synthesizer.	27
Figure 4.4 – Maxim MAX2769 implementation and connections.	27
Figure 4.5 - MAX V 5M80ZE64C5N CPLD scheme.....	29
Figure 4.6 – CPLD Moore State-Machine Design flowchart.	30
Figure 4.7 – CPLD 1.023 MHz write strobe (Red). 16.368 MHz input (Yellow) divide by 16, for 1-bit input.	31
Figure 4.8 – CPLD 2.046 MHz write strobe (Red). 16.368 MHz input (Yellow) divide by 8, for 2-bit input.	31
Figure 4.9 – CPLD 4.092 MHz write strobe (Red). 16.368 MHz input (Yellow) divide by 4, for 4-bit input.	31
Figure 4.10 – PIC32MX340F512L in circuit serial programmer connections.....	32
Figure 4.11 – Microchips TCP/IP stack architecture.	33
Figure 4.12 – TCP/IP Stack Main application high-level flowchart.	34
Figure 4.13 – UDP Server Task high-level flowchart.	35
Figure 4.14 – DMA 5880-Byte buffer representation.	35

Figure 4.15 – UDP Server Application high-level flowchart.....	35
Figure 4.16 – DP83848M 10/100 Ethernet transceiver connections.....	37
Figure 4.17 – Texas Instruments TPS767D318 supply connections.....	38
Figure 4.18 - Coplanar Waveguide Cross-section.	38
Figure 4.19 – LenaCalc calculated Coplanar Waveguide over Ground line characteristics.....	39
Figure 4.20 – PCB Layer stack-up.....	39
Figure 4.21 – EtherGNSS v3 PCB (Real Size Topside & Underside).	40
Figure 4.22 – EtherGNSS v3 Layout (Topside).	41
Figure 4.23 – EtherGNSS v3 Layout (Underside).	41
Figure 5.1 – EtherGNSS v3 Embedded Webserver main page.....	44
Figure 5.2 – EtherGNSS v3 Embedded Webserver front-end configuration page.	44
Figure 5.3 – EtherGNSS v3 Embedded Webserver network settings page.	45
Figure 5.4 – EtherGNSS v3 JAVA GNSS IF Streamer Main application high-level flowchart.	46
Figure 5.5 – EtherGNSS v3 JAVA GNSS IF Streamer Menu.	46
Figure 5.6 – EtherGNSS v3 JAVA GNSS IF Streamer RF configuration.	47
Figure 5.7 – EtherGNSS v3 JAVA GNSS IF Streamer Capture mode submenu.	47
Figure 5.8 – EtherGNSS v3 Java capture procedure flowchart.	48
Figure 5.9 – EtherGNSS v3 JAVA GNSS IF Streamer Report.....	48
Figure 6.1 – Test setup used for characterization of the EtherGNSS v3 system.....	49
Figure 6.2 – Output signal spectrum of the 16.368 MHz TCXO.	49
Figure 6.3 – Case 1 – GPSL1 narrowband analog output signal spectrum.	50
Figure 6.4 – Case 1 – GPSL1 narrowband digital output signal spectrum.....	51
Figure 6.5 – Case 2 – GPS & Galileo L1 analog output signal spectrum.	51
Figure 6.6 – Case 2 – GPS & Galileo L1 digital output signal spectrum.....	52
Figure 6.7 – Case 3 – GLONASS wideband analog output signal spectrum.....	53
Figure 6.8 – Case 3 – GLONASS wideband digital output signal spectrum.....	53
Figure 6.9 – Probed signal acquisition 1. $f_s = 16.368$ MHz, $IF = 4.092$ MHz, Filter _{BW} = 2.5 MHz, $f_{VCO} = 1571.328$ MHz	54
Figure 6.10 – Acquisition result for acquisition 1.	54
Figure 6.11 – Tracking result for satellite PRN30 for acquisition 1.....	55
Figure 6.12 – Navigation solution for acquisition 1.	56
Figure 6.13 – Zoomed coordinate variation for acquisition 1.	56
Figure 6.14 – Orbitron [22] simulated Sky plot at acquisition 2.	57
Figure 6.15 – Probed signal acquisition 2. $f_s = 16.368$ MHz, $IF = 4.092$ MHz, Filter _{BW} = 2.5 MHz, $f_{VCO} = 1571.328$ MHz.	57
Figure 6.16 – Acquisition result for acquisition 2.	58
Figure 6.17 – Tracking result for satellite PRN30 for acquisition 2.....	58
Figure 6.18 – Navigation solution for acquisition 2.	59

Figure 6.19 – Zoomed coordinate variation for acquisition 2.	59
Figure 6.20 – Probed signal acquisition 3. $f_s = 16.368$ MHz, $IF = 2.046$ MHz, Filter _{BW} = 4.2 MHz, $f_{VCO} = 1573.374$ MHz.	60
Figure 6.21 – Probed signal acquisition 4. $f_s = 16.368$ MHz, $IF = 0$ MHz, Filter _{BW} = 18.8 MHz, $f_{VCO} = 1602$ MHz.	60

List of Tables

Table 2.1 – GPS Frequency plan.	5
Table 2.2 – Galileo signals main characteristics.	7
Table 2.3 – GLONASS legacy and modernization program frequency plan.....	9
Table 2.4 – Comparison of the most important properties of GPS, GALILEO and GLONASS (as of July 2012)	12
Table 3.1 - Execution time comparison between acquisition methods.	17
Table 4.1 - Receivers commercial IC front-ends, adapted from [15].	26
Table 4.2 - MAX V 5M80ZE64C5N Features.	28
Table 4.3 – CPLD design signal description.	30
Table 4.4 - Microchip PIC32MX795F512L main features for implementation.	32
Table 4.5 – Firmware specific UDP sockets.	34
Table 4.6 – EtherGNSS v3 overall prototyping cost (without assembly).....	40
Table 6.1 – Identification and coordinates of reference.....	53
Table 6.2 – Acquisition 1 solution coordinates.....	56
Table 6.3 – Acquisition 2 solution coordinates.....	59
Table 6.4 – EtherGNSS v3 Specifications.....	61

List of Abbreviations and Acronyms

ADC	– Analog-to-digital converter
AGC	– Automatic gain control
AJAX	– Asynchronous Javascript and XML
BOC	– Binary offset carrier modulation
BPSK	– Binary phase shift keying
C/A	– Coarse Acquisition code
CDMA	– Code Division Multiple Access
CPWG	– Coplanar Waveguide over Ground
DSSS	– Direct sequence spread spectrum
ECEF	– Earth-Centered, Earth-Fixed
ETRS89	– European Terrestrial Reference System 1989
FDMA	– Frequency Division Multiple Access
FFT	– fast Fourier transform
Galileo	– European Global Satellite Navigation System
GCC	– GALILEO Control Center
GLONASS	– Russian <i>Globalnaya Navigatsionnaya Sputnikovaya Sistema</i>
GMS	– GALILEO Ground Mission Segment
GNSS	– Global Navigation Satellite System
GPS	– Global Positioning System
GUI	– Graphical User Interface
HOW	– Handover word
HTML	– Hypertext Markup Language
IC	– Integrated circuit
IF	– Intermediate frequency
LE	– Logic Elements
LED	– Light Emitting Diode
LNA	– Low-noise amplifier
LO	– Local oscillator
MAC	– Media access control
MCU	– Microcontroller
NF	– Noise figure
PC	– Personal computer
PDOP	– Positional Dilution of Precision
PHY	– Physical Layer transceiver
PIPO	– Parallel-in, parallel-out
PLL	– Phase-locked loop

PoE	– Power over Ethernet
PPS	– Precise Positioning Service
PRN	– Pseudo Random Noise
PRS	– Public related service
PSD	– Power Spectrum Density
PVT	– Position, velocity, and time
RF	– Radio frequency
RMII	– Reduced Media Independent Interface
SBAS	– Satellite-based augmentation system
SDR	– Software defined radio
SIPO	– Serial-in, parallel-out
SPS	– Standard Positioning Service
TCXO	– Temperature-compensated crystal oscillator
TLM	– Telemetry

1 Introduction

1.1 Purpose and motivation

Global Navigation Satellite Systems (GNSSs) represent the satellite navigation systems that provide autonomous geo-spatial positioning with global coverage.

The core satellite navigation systems currently are GPS, Galileo and GLONASS, three systems developed by three political and economic powers, which are in various states of maturity and robustness. The development of the NAVSTAR GPS took nearly 20 years and cost more than \$10 billion. It is the first and currently the only fully operational GNSS, globally available since 1994, consisting on 32 medium Earth orbit satellites. Galileo and GLONASS are in development stages, with Galileo having 2 test bed satellites in orbit since 2005 and with the first two of four operational satellites to validate the system launched on October 2011. Full completion of the 30 satellites Galileo system (27 operational + 3 active spares) is expected by 2019 [1]. With the approval for development granted in 1970 by the Soviet Union, GLONASS reached full operation in 1995 by the hand of Russia Federation. After that there were no further launches until December 1999. By 2011, the full orbital constellation of 24 satellites was restored, enabling full global coverage. The GLONASS satellite designs have undergone several upgrades, with the latest version under implementation being GLONASS-K.

With GNSS, availability of two or more constellations, more than doubling the total number of available satellites in the sky, will enhance service quality, increasing the number of potential users and applications.

Traditional GNSS receivers designs are based on closed hardware implementations and signal processing. In an effort to take advantage of the rapid innovation of this technology, turning to open-architecture systems and software defined radios (SDRs) is the alternative to traditional receiver closed system designs. Compared with traditional receivers, the SDR has much more flexibility, especially in the GNSS receiver area [2]. This makes SDRs an ideal platform for development, test of algorithms and integration with other devices without the need of hardware replacement. This would likely be impractical to be done in traditional receivers and is one key element to keep pace with the changing technology without the cost of a complete redesign.

A SDR consists of two components, the hardware front-end implementation and the software signal processing, both presented and described throughout this document. A well-designed hardware ensures a proper functioning and a good foundation capable of being updated, extending the life-time of the product.

This document will introduce the design of a GNSS single frequency L1 band (civilian-use signal) SDR receiver, for the GPS, Galileo and GLONASS signals from the hardware point-of-view, although validation will be accomplished through the use of open-source signal-processing softwares.

1.2 Goals, requirements and tasks

The main objective of this work is to develop an open-architecture sampler for the combined GPS/Galileo/GLONASS L1 signals. As it will be shown along this document, this sampler will be a key component for accessing the potentialities offered by these systems. The main requirements for this project are:

1. Interoperability between GPS, Galileo and GLONASS systems;
2. Use of a flexible and efficient configurable front-end chipset;
3. Data transmission through an IP interface, optimized for maximum throughput to enable maximum acquisition rate performance;
4. The developed system should be platform independent, intended for all types of users;
5. Easy-to-use graphical user interface (GUI).

In order to achieve these objectives, the following set of key tasks were established and met:

1. Study GNSS characteristics and interoperability;
2. Research about current SDR implementations, algorithms and currently active projects;
3. Study of Ethernet Protocols and implementations;
4. Study hardware components characteristics and implementations to guarantee the best trade-off between price, efficiency and performance;
5. Implement glue logic Serial-in to Parallel-out (SIPO)/Parallel-in to Parallel-out (PIPO) by means of HDL language and map it into a Complex Programmable Logic Device (CPLD);
6. Research about best implementations and develop an application to interface with the receiver.
7. Design, develop and test the system printed circuit board (PCB);
8. Test and experimental characterize the developed system.

1.3 Outline

What follows is a brief summary of this document organization. Section 2 provides an overview of satellite navigation fundamentals and GNSS signal standards. Working frequencies and their modulations are presented as well as the navigation message format. The interoperability between the targeted systems is also discussed. In section 3, the basis of a GNSS receiver is presented with the hardware and software briefly described. Also in this section, existing SDRs and their conception are presented, namely the first complete SDR-GPS implementation, as well as other current implementations. Section 4 introduces the embedded system architecture. All the considered hardware is explained in detail along with the developed software. Fabrication concerns are described and the complete developed system is presented. In section 5, the user interfaces are demonstrated and described. Once the design and fabrication of the receiver has been carried out, validation is required. Section 6 deals with the characterization of the system and validation. Finally, some conclusions and future work suggestions are presented in section 7.

2 GNSS Fundamentals and Systems Overview

2.1 Introduction

In order to design a software-defined single frequency GNSS receiver, firstly the fundamentals of satellite navigation and its principles must be analysed. This section deals with the systems overview, also the characteristics of the signal and data transmitted from the satellites and received by the Radio Frequency (RF) front-end. These characteristics will impose steps to take in the course of the system development.

2.2 Satellite navigation principles

Satellite navigation systems are comprised of three functional segments: The *space segment*, corresponding to the operating satellites constellation; the *control segment*, corresponding to the ground stations involved in the monitoring of the system; and the *user segment*, represented by the receivers for the civilian and military use.

All satellite navigation systems determine coordinates using the same basic principles. Consider the simple model shown in Figure 2.1, where a transmitter and a receiver with synchronized clocks are located in a one dimensional plane. Knowing that a radio wave propagates at the speed of light of around 300000 km/s, by measuring the delay from when the signal is transmitted to when it is received, it is possible to assess the receiver distance from the transmitter. However, there can be a discrepancy between the clocks of the transmitter and the receiver, leading to a large error in the calculated distance.

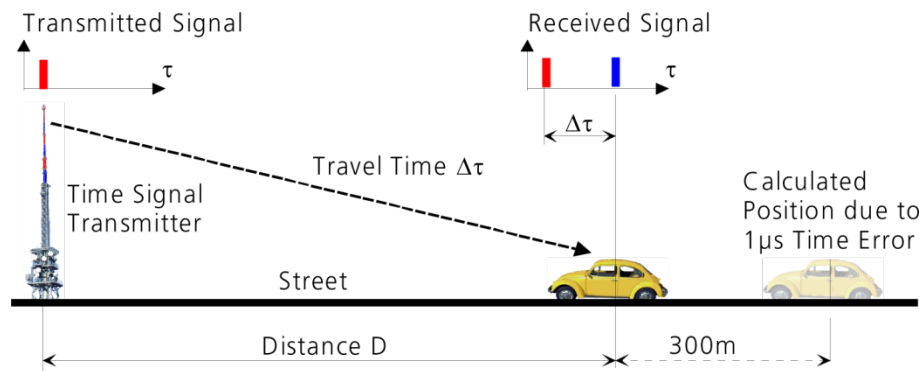


Figure 2.1 – 1D simple model navigation principle [3].

To overcome the local clock synchronization discrepancy, a second synchronized time signal transmitter where the separation to the first transmitter is known, can be used to outfit the exact same clocks, as shown in Figure 2.2.

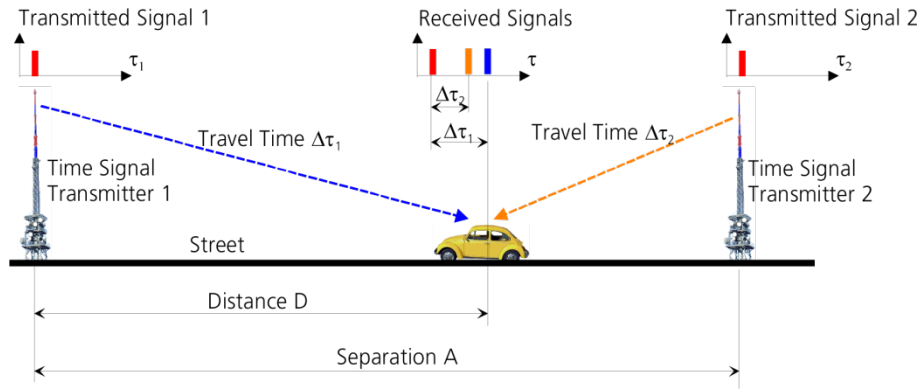


Figure 2.2 – 1D simple model with two transmitter [3].-

From this model a conclusion can be drawn: in order to exactly determine the position and time along a one dimension plane, at least two time signal transmitters are required. Now, this conclusion can be extrapolated to more dimensions. It is then possible to say that when an unsynchronized receiver clock is employed in calculating position, it is necessary that the number of time signal transmitters exceed the number of unknown dimensions by a value of one.

Satellite navigation systems use satellites as time-signal transmitters. In order for a GNSS receiver to determine its position, it must receive time signals from at least four separate satellites, represented in Figure 2.3, to calculate the signal travel times Δt_1 to Δt_4 .

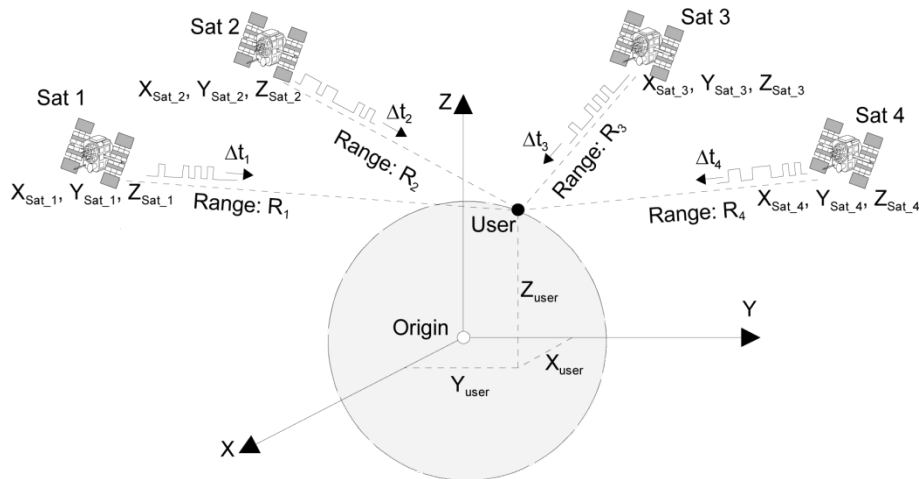


Figure 2.3 – Four satellites are required to determine a position in 3-D space.

Calculations are effected in a Cartesian, three-dimensional coordinate system with a geocentric origin. The range of the user from each of the four satellites, R_1 to R_4 , can be determined with the help of the calculated signal travel times between the satellites and the receiver. As the locations X_{Sat} , Y_{Sat} and Z_{Sat} of the four satellites are known, the user coordinates can be calculated.

2.3 Global Navigation Satellite Systems

Having the satellite navigation principles briefly studied. In this sub-section, the GPS and Galileo and GLONASS standards are explained in more detail.

2.3.1 GPS

The NAVSTAR GPS is a continuously available, satellite-based RF positioning system providing highly accurate position, velocity and time (PVT). The baseline system is specified for 24 satellites, however the system currently employs 28-31 satellites, in a constellation of 6 orbital planes inclined 55 degrees to the equator, at 20,183 Km from the mean surface of the Earth. Each plane contains 4-7 active satellites.

Although other signals are being planned and studied for future implementation, the GPS, at present, transmits only on two carrier radio frequencies in the UHF band, referred to as L1 and L2 frequencies. Table 2.1 contains current and planned GPS carrier radio frequencies. As stated in [4], three signals are transmitted at the moment by GPS in L1: C/A Code and P(Y) Code. In the future, an additional new civil signal, known as L1C and military M-Code, will also be transmitted.

Table 2.1 – GPS Frequency plan.

Band	Frequency [MHz]	Bandwidth [MHz]	Description
L1	1575.420	~ 12	Coarse-acquisition (C/A) & encrypted precision P(Y) codes + L1 civilian (L1C) & military (M) codes on future Block III satellites.
[L1_{C/A}	1575.420	2.046]	Coarse-acquisition (C/A)
L2	1227.600	~ 12	P(Y) code + L2C & military codes on the Block IIR-M.
L3	1381.050		Used for nuclear detonation (NUDET) detection.
L4	1379.913		Being studied for additional ionospheric correction.
L5	1176.460	~ 12	Proposed for use as a civilian safety-of-life (SoL) signal.

The L1 carrier is modulated by two pseudo random noise (PRN) codes and navigation data, the C/A code and the P(Y) code. The L2 frequency is modulated by the P(Y) code whose availability is restricted to precise positioning service (PPS) military use, while C/A code aims for civilian Standard Positioning Service (SPS). Henceforth only the C/A code will be presented since this work focus on the acquisition and signal treatment of L1 signals only.

The C/A code is a 1023 bit deterministic sequence called pseudorandom noise which, when transmitted at 1.023 Mbit/s, repeats itself every millisecond. Each satellite transmits a unique PRN code, which is uncorrelated with any other satellite's PRN code, allowing them to not interfere significantly with each other. This type of transmission is called Code Division Multiple Access (CDMA), allowing the receiver to recognize multiple satellites on the same frequency.

GPS satellite transmissions utilize direct sequence spread spectrum (DSSS) modulation. DSSS provides the structure for the transmission of the ranging signal and the essential navigation data. The DSSS signal is used to modulate the carrier using binary phase shift keying (BPSK).

BPSK is a simple digital signalling scheme in which a carrier is either transmitted “as is” or with a 180° phase shift depending on whether a digital 0 or 1 is being conveyed.

The effect of modulation of the L1 carrier with the C/A code and the navigation data for one satellite can be seen in Figure 2.4. A C/A code is generated and multiplied (XORed) with the navigation data. The resulting signal modulates the carrier using BPSK which is then amplified and broadcasted.

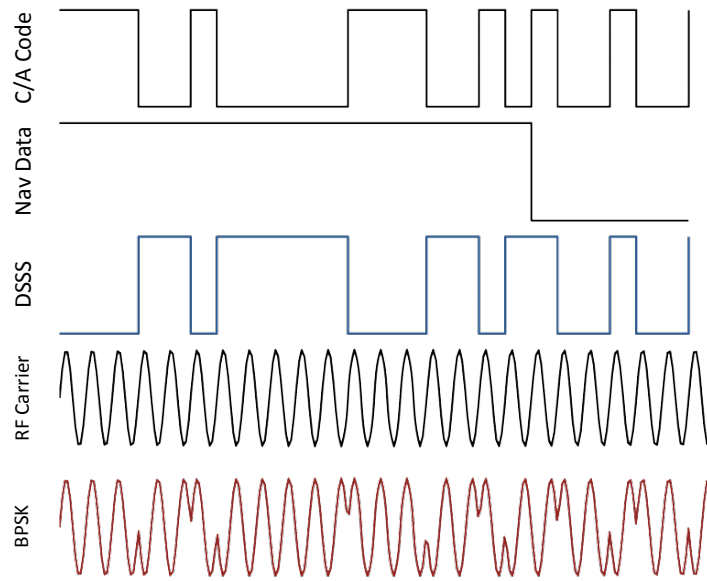


Figure 2.4 - DSSS with BPSK modulation (Adaptation from [5]).

There are two primary reasons why DSSS waveforms are employed for satellite navigation [6]. First and most importantly, the frequent phase inversions in the signal introduced by the PRN waveform enable precise ranging by the receiver, providing significant rejection of narrowband interference. Second, the use of different PRN sequences from a well-designed set enables multiple satellites to transmit signals simultaneously and at the same frequency band, CDMA. For this reason, the transmission of multiple DSSS signals having different spreading sequences is possible on a common carrier frequency.

The spectra of the GPS signals present in the L1 band, is presented in Figure 2.5, in accordance with Table 2.1 of the GPS frequency plan.

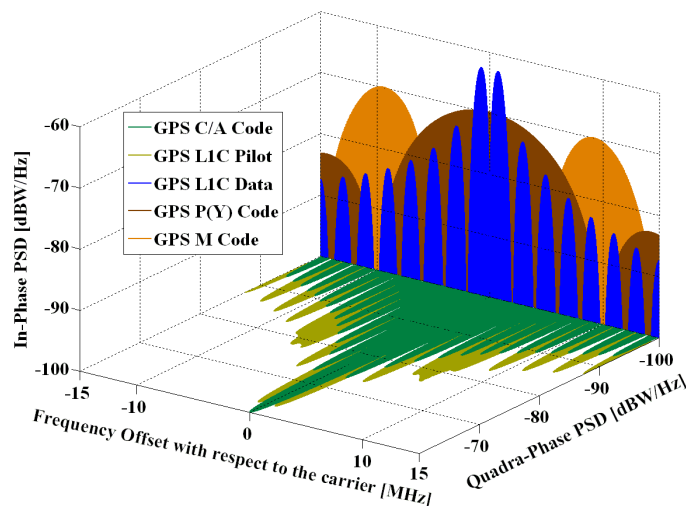


Figure 2.5 – Spectra of GPS Signals in L1 [1].

Of all the signals above presented, the C/A Code (dark green) is the best known as most of the receivers that have been built until today are based on it. It is important to note that although the GPS L1C pilot and data signals are shown in quadrature in the figure above, according to [4] the final phasing is still to be decided.

2.3.2 Galileo

The Galileo space segment constellation will comprise a constellation of a total of 30 Medium Earth Orbit satellites, of which 3 are spares. The Galileo satellite constellation has been optimised to have three equally spaced orbital planes, at an altitude of 23,222 km with an inclination of 56°. Each orbital plane would have 9 operational satellites, equally spaced and 1 spare satellite (also transmitting).

Each Galileo satellite will broadcast 10 different navigation signals in the frequency ranges 1164–1215 MHz (E5 band), 1260–1300 MHz (E6 band), and 1559–1592 MHz (E2-L1-E1 band), making it possible for Galileo to offer an open service (OS), safety-of-life service (SoL), commercial service (CS) and public regulated service (PRS). All satellites will make use of the same carrier frequencies with different ranging codes again through CDMA transmission.

Table 2.2 – Galileo signals main characteristics.

Band	Frequency [MHz]	Bandwidth [MHz]	Description
E2-L1-E1	1575.420	32.736	OS + SoL + CS + PRS
[L1F-d	1575.420	4.092]	OS + PRS
E5	1191.795	51.150	
[E5a	1176.450	20.460]	OS + SOL + CS
[E5b	1207.150	20.460]	OS + SOL + CS
E6	1575.420	40.920	CS + PRS

Six signals [1], including three data-less channels, so-called pilot tones (ranging codes not modulated by data), are accessible to all Galileo users on the E5a, E5b and L1 carrier frequencies for Open Services and Safety-of-life Services and their accessibility are free of direct charge. Two signals on E6 with encrypted ranging codes, including one dataless channel are accessible only to some dedicated users that gain access through a given Commercial Service provider.

Finally, two signals, one in E6 band and one in E2-L1-E1 band, with encrypted ranging codes and data are accessible to authorized users of the Public Regulated Service.

Moreover, several combinations for the open services provided by Galileo are also possible, such as a dual frequency service based on using L1 and E5a, for best ionospheric (atmospheric electricity) error cancellation, to single frequency services, at L1, E5a, E5b or E5a and E5b together, on which case the ionospheric error is removed using a model, and even triple frequency services using all the signal together which can be exploited for very precise centimetric type of applications. In GPS only one signal is available, which does not allow the kind of optimisation performed for Galileo. This will be overcome by the GPS modernisation programme, where more GPS signals will be made available.

However, the signal of interest for this work is the L1F-d (data), a signal transmitted in the E2-L1-E1 band as part of the Galileo open service (OS). The advantage of this signal in this work is that the L1 band of GPS signal is centered in the same carrier, making this signal a non-hardware changing signal to process in a GPS common SDR, as it can be seen in the spectra of Galileo signals in E2-L1-E1 Band below.

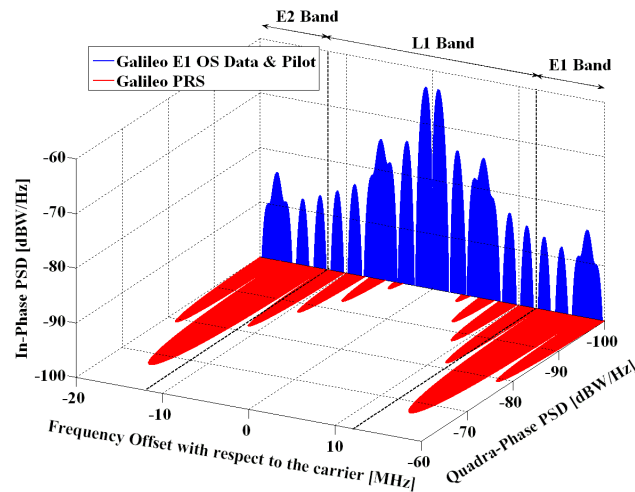


Figure 2.6 – Spectra of Galileo Signals in E2-L1-E1 [1].

The L1F signal has a data bandwidth of 4.092 MHz, twice that of the GPS system. In the figure above, the L1F signal is comprised in the two main lobes with zero offset from the carrier.

The L1 band modulation should combine three distinct signals associated to two different services into a phase modulated composite signal. The L1 signal consists of multiplexing three components which are:

- The L1P channel corresponds to the PRS.
- The L1F pilot channel corresponds to the pilot OS.
- The L1F data channel corresponds to the data OS.

For this last signal, L1F data channel, the carrier modulation corresponds to binary offset carrier modulation (BOC) by three components: the L1F navigation data stream, the L1F data channel PRN code sequence and the L1F sine-phased subcarrier.

The notation $\text{BOC}(f_s, f_c)$ is used, where f_s represents the square wave subcarrier frequency in units of 1.023 MHz, and f_c represents the chipping rate in units of 1.023 MHz.

The L1F-d signal is modelled through $\text{BOC}(1,1)$, therefore having a 4092 code length with a 1023 MHz chipping rate (bitrate) giving it a repetition rate of 4 ms.

2.3.3 GLONASS

Similarly to GPS, GLONASS satellites transmit signals centered on two discrete L-band carrier frequencies and have a program under development which will introduce new signals that will make its service more available, accurate, reliable and robust.

The space segment constellation is composed by 31 satellites of which 24 are operational plus 1 in flight test for the GLONASS modernization, located in middle circular orbit at 19,100 km altitude with a 64.8 degree inclination.

Table 2.3 – GLONASS legacy and modernization program frequency plan.

Band	Frequency [MHz]	Bandwidth [MHz]	Description
L1	1597 - 1605	1.023	Standard Positioning Service (SPS)
L1_[CDMA]	1575.420		Future SPS will use BOC(1,1) modulation
L2	1240 - 1260	10.23	Precise Positioning Service (PPS)
L2_[CDMA]	1242.000		Future PPS will use BOC(1,1) modulation
L3_[CDMA]	1207.140		Transmitted open signal, under test, uses QPSK(10) modulation
L5_[CDMA]	1176.450		Future open signal in the will use BOC(4,4) modulation + SOL

The signals use similar DSSS encoding and BPSK modulation as in GPS signals. However, the signal access scheme of this signal is Frequency Division Multiple Access (FDMA) where each satellite transmits the same PRN sequence on a different frequency using a 14-channel spanning either side from 1602.0 MHz, also known as the L1 band. The center frequency is $1602 \text{ MHz} + n \times 0.5625 \text{ MHz}$, where n is the satellite frequency channel number shared by two antipodal satellites. A representation of the spectra where the PSDs of the GLONASS signals in the L1 band is shown in Figure 2.7.

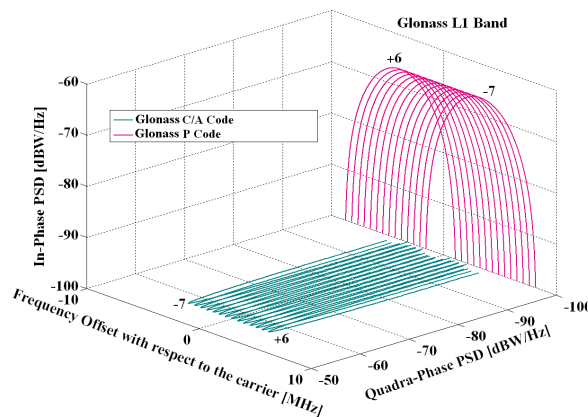


Figure 2.7 – Spectra of GLONASS signals in L1 [1].

In the figure above, each of the channels is filtered to only transmit the main lobe of the BPSK, thus the 14-channel spanning of the C/A code can be observed.

The choice of FDMA over CDMA is one of the design trade-offs for this work, more details on sub-section 2.3.5. FDMA typically leads to more expensive receivers because of the front-end components required to process multiple frequencies or a wider bandwidth, capable of acquiring them. This issue will be surpassed with the future of the GLONASS modernization program. Such an arrangement will allow easier and cheaper implementation of multi-standard GNSS receivers.

2.3.4 Navigation Message

The navigation data transmitted by the GNSS system provides the user with the information necessary to compute the precise locations of each visible satellite and time of transmission for each navigation signal. This sub-section outlines the main features of the GNSS navigation message format. A more complete description is available in [4] [7] [8].

The Navigation Message provides all the necessary information to allow the user to perform the positioning service. It includes the Ephemeris parameters, this parameters are needed to compute the satellite coordinates with enough accuracy, the Time parameters and Clock Corrections, to compute satellite clock offsets and time conversions, the Service Parameters with satellite health information (used to identify the navigation data set), Ionospheric parameters model needed for single frequency receivers, and the Almanacs, allowing the computation of the position of "all satellites in the constellation", with a reduced accuracy (1 – 2 km), which is needed for the acquisition of the signal by the receiver.

The GPS current Navigation Message contains 25 frames of 30 seconds each, forming the master frame that takes 12.5 minutes to be transmitted, as shown in Figure 2.8. Every frame is subdivided into 5 sub-frames of 6 seconds each consisting of 10 words, with 30 bits per word. The last 6 bits in each word of the navigation message are used for parity checking to provide the user equipment with a capability to detect bit errors during demodulation. This subframes always starts with the telemetry word (TLM) to assist the user equipment in locating the beginning of each subframe. Next, the transference word (HOW) appears. This word provides time information, allowing the receiver to acquire the week-long P(Y)-code segment [1].

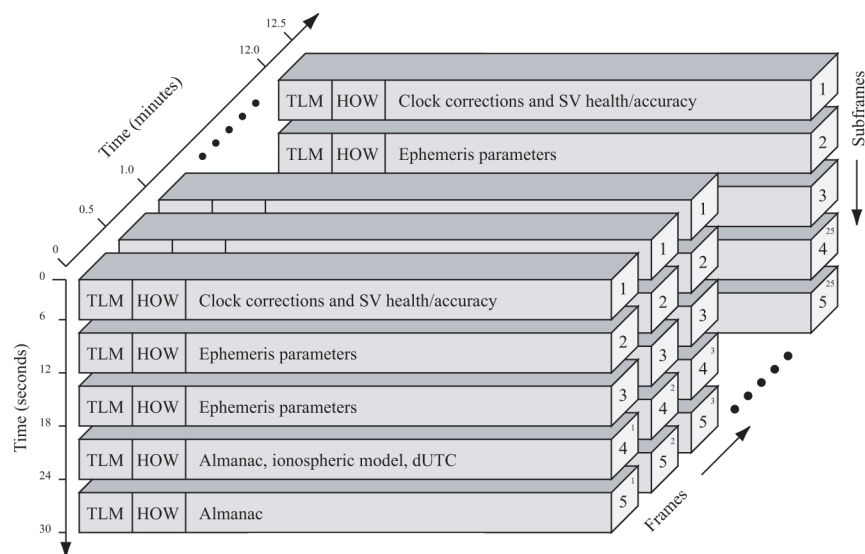


Figure 2.8 - GPS navigation data structure [2].

GPS modernisation introduces four new data messages; three civilian messages, and one military message. They provide more accurate and frequent message data than the legacy navigation message.

The Galileo satellites will broadcast five types of data (services) in four navigation messages: a freely accessible, an Integrity, the Commercial and the Governmental Navigation Messages.

The Galileo ephemeris parameters are *Keplerian*-like orbital elements as in GPS. The Almanac is also similar to the GPS and GLONASS ones.

The complete navigation message is transmitted on each data channel as a sequence of frames. A frame from the freely navigation message is composed by 12 sub-frames, and each sub-frame composed by 5 pages of 10s duration.

The page starts with a Synchronisation Word (SW) followed by the data field. After the data, a Cyclic Redundancy Check (CRC) parity bits are provided to detect the reception of corrupted data. The page ends with tail bits for the Forward Error Correction (FEC) encoding. With Block Interleaving on the resulting frames, allowing the reduction of the bit error ratio in the increased data rates, Galileo Message Data Streams have 3 levels of error coding applied.

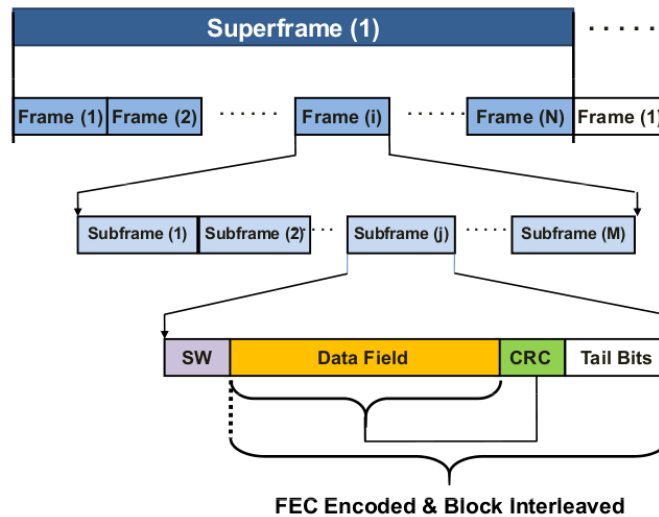


Figure 2.9 - Galileo Navigation Message Structure [1].

GLONASS satellites modulate two navigation messages onto the standard (C/A) and high accuracy (P) signals.

The navigation message of the standard accuracy signal (C/A) is broadcast as continuously repeating superframes with a duration of 2.5 minutes. Each superframe consists of 5 frames of 30 seconds, and each frame consists of 15 strings of 2 seconds duration.

The first 4 frames contain almanac data for 20 satellites, 5 per frame, and the 5th frame almanac data for 4 satellites.

The content of the messages is divided in immediate data of the transmitting satellite and non-immediate data for the other satellites. The immediate data is repeated in the first four strings of every frame, comprising ephemeris parameters, satellite clock offsets, satellite healthy flag and the relative difference between carrier frequency of the satellite and its nominal value. The non-immediate data is broadcast in the strings 5 to 15 of each frame (almanac for 24 satellites).

GLONASS ephemeris parameters differ from GPS and Galileo data, instead of Keplerian orbital elements, they are provided as Earth Centred Earth Fixed (ECEF) Cartesian coordinates in position and velocity. The almanac is quite similar to the GPS one, given as modified Keplerian parameters [1].

2.3.5 GPS, Galileo and GLONASS Interoperability

Before the study of the GNSSs interoperability, it is important to resume the characteristics of the three targeted systems studied in the previous subsections. Table 2.4 lists the most important properties of the systems.

Table 2.4 – Comparison of the most important properties of GPS, GALILEO and GLONASS (as of July 2012)

	GPS	Galileo	GLONASS
Start of development	1973	2001	1972
Status	Full operability	In development	Full operability Modernization program
Space Segment constellation	32 operational	2 In-Orbit Validation 2 Flight Model	24 operational
Orbitals	6	3	3
Inclination	55°	56°	64.8°
Altitude	20,180 km	23,222 km	19,100 km
Orbital Period	11 hours 58 min	14 hours 5 min	11 hours 15.8 min
Signal Access Scheme	CDMA		FDMA
L1 frequency band	1575.42 MHz		1602 + N*0.5625 MHz N = [-7 ; +6]
L1 band modulation	BPSK	BOC(1,1)	BPSK
Frequencies	2 frequencies, 5 planned	3 frequencies	2 frequency bands (on 14 frequencies), 5 planned
Services	2 (civilian + military), 5 planned	4 (OS, SoL, CS and PRS)	2 (civilian + military), 5 planned
Integrity Transmission	None, planned	Planned	None, planned

With the use of the above resumed table, now it is easier to study interoperability between systems. “Interoperability refers to the ability of global and regional navigation satellite systems and augmentations and the services they provide to be used together to provide better capabilities at the user level than would be achieved by relying solely on the open signals of one system.”, [9].

At system level, interoperability is viewed as the capability to provide the same navigation solution, within their respective system accuracy, by all systems. Thus, it is possible to say that in terms of system interoperability, the three systems (GPS, Galileo and GLONASS) are interoperable, although a solution of signal interoperability must be analysed.

Signal interoperability is achieved when the signals provided by different systems are similar enough to allow a GNSS receiver to use those signals with minor modification. In [10], the following factors were considered for signal interoperability:

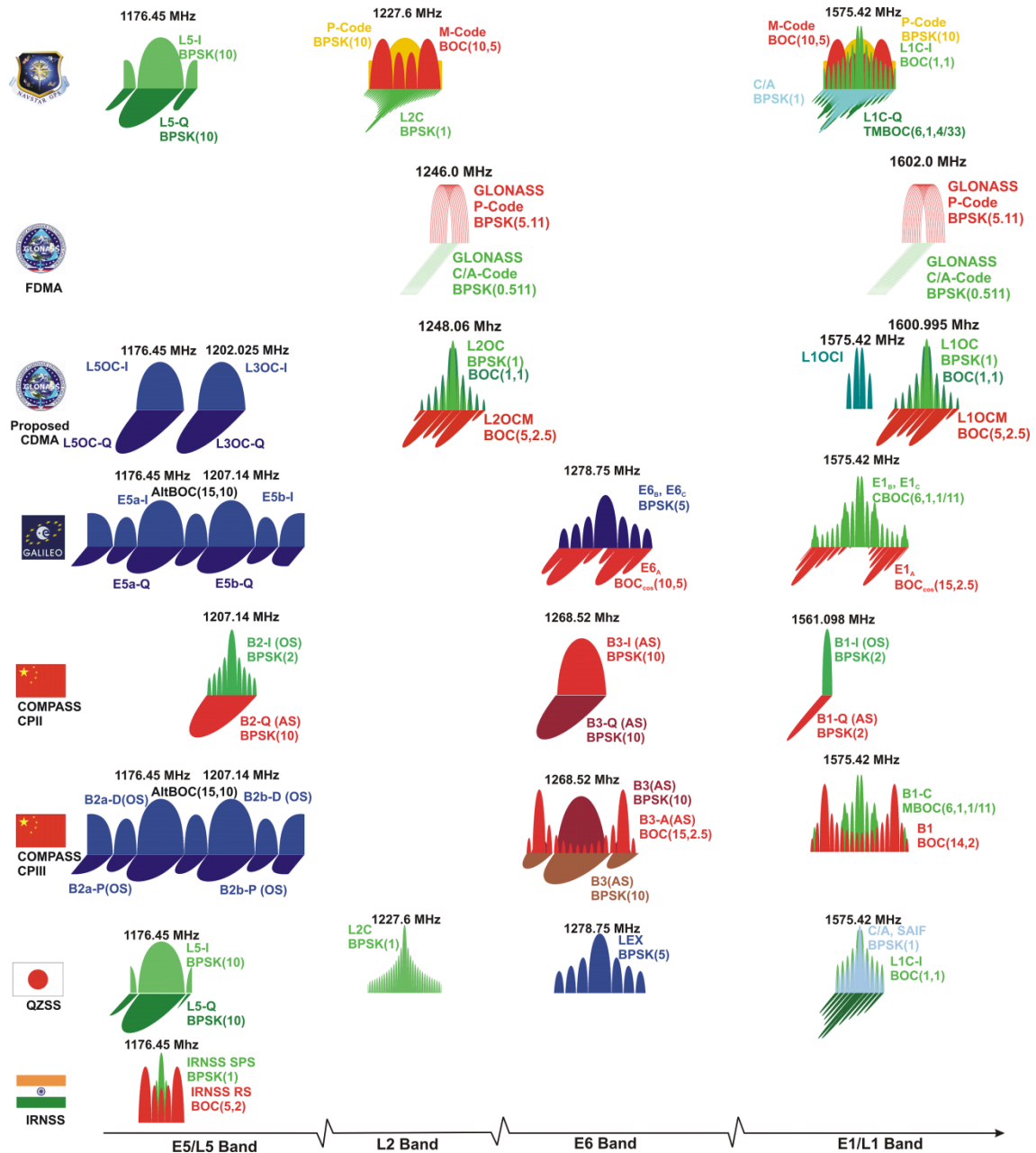
Characteristics, such as modulation, signal structure or selection of the codes that require only software modifications at the receiver can be considered to not affect interoperability. As a

consequence of this, only CDMA satellite systems can fulfil this requirement, which is not the case of GLONASS which uses FDMA. Thus, GPS and Galileo are Signal Interoperable with regard to the L1 and L5/E5a frequencies and are both free services. GLONASS is not Signal Interoperable with GPS or Galileo, but it is System Interoperable according to the definition given earlier. Once again it should be noted that the GLONASS modernization program will change this issue giving a huge step into full GNSS interoperability. As for now, different frequencies may introduce frequency biases and degrade accuracy. Front-ends with a larger bandwidth, or multiple front-ends, will be necessary.

Interoperability is one of the mechanisms to achieve a Global Navigational Satellite System of systems. However, the independence of the systems also provides greater reliability and integrity contributing for a less vulnerable system. It also brings the advantage of being independently operated therefore providing redundancy to the GNSS user community, hence increasing the market confidence on the technology.

Figure 2.10 shows the frequency bands used by the different GNSS systems. In this figure the frequency plans of the GPS, Galileo and GLONASS are presented, as well as of the rest of the actual GNSS systems, some actually in development stages and others in full operability. The L1 band signal interoperability between the targeted systems can be observed by their spectra, with GPS and Galileo actual signals using the same carrier. The future modernization program signals of GLONASS also can be seen in the figure where a CDMA signal will be available on a 1575.42 MHz carrier. At this time it is possible to conclude that in a near future all the system will be fully interoperable in System and in Signals levels.

Given the open-architecture of the SDR developed in this work, it is expected that all the signals in the L1 band would be made possible to acquire, although since the Chinese COMPASS II as well as the Japanese QZSS space segment constellation is composed of geostationary satellites, its coverages are on the Asia-Pacific region. It is not possible to validate the interoperability with these two systems.



2.4 Summary

Satellite navigation fundamentals were explained giving an introduction to the study of the GNSS targeted system standards together with the most important properties of the various signals and data. System and Signal interoperability between systems was presented in order to better understand the scope of this work.

3 GNSS Samplers and SDR receivers

3.1 Basic GNSS receiver Architecture

A basic software-based GNSS receiver is composed of two main parts, as seen in Figure 3.1. The first part covers the hardware and can be divided in two functional blocks: RF section and IF section. The RF section consists of analog hardware modules responsible for RF to IF conversion, while the IF section is composed of digital hardware modules. The second part is the baseband software processing, which performs acquisition and tracking of the received signals, as well as the necessary algorithms for the computations of the receiver PVT.

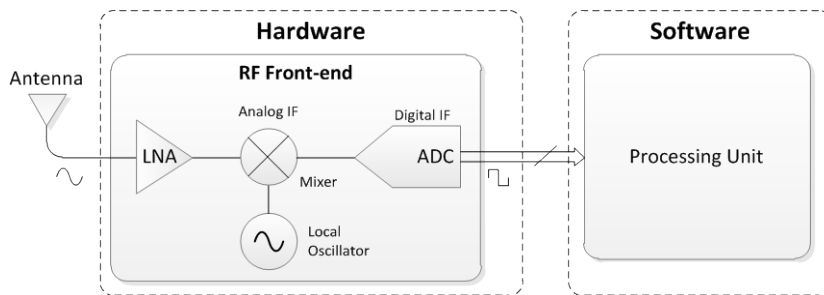


Figure 3.1 – Architecture of a basic software GNSS receiver.

The process starts with the signals transmitted from the satellites, propagating through space, and inciding on a GNSS's antenna. Due to the GNSS signal's low power upon reception, there is usually a set of filtering and low-noise amplification stages after the antenna. The antenna isn't usually considered part of the hardware design, although it is the first component in the signal path which will make the first signal conditioning. Thus, there are two possible solutions: making use of an antenna with a built-in amplifier, also known as active antenna, which means that the low-noise amplifier (LNA) is set to work in low-gain mode so as not to saturate the system. Or making use of a passive antenna; in this case the LNA is set to high-gain mode, avoiding the power consumption of the antenna and resulting in a low power consumption system.

The signal is then processed in the RF front-end. The RF front-end of a software-based GPS receiver, after amplification of the signal coming from the antenna, downconverts the input signal from RF to a low IF. This down-conversion is accomplished by mixing the input RF signal with the local oscillator (LO), which must be carefully chosen to avoid harmonics and image frequencies near IF, using one or two mixers followed by an appropriate bandpass filtering. The signal *Doppler* and the PRN codes are preserved after the mixing process, only the carrier frequency is lowered.

The resulting analog IF signal is then converted to a digital IF signal through the analog-to-digital converter (ADC). The ADC sampling frequency, f_s , should be carefully chosen. The requirements to prevent signal aliasing define a minimum sampling frequency, called *Nyquist frequency*, as $f_s = 2.B$, where B represents the analog (lowpass) signal bandwidth.

The IF digital signal is then transmitted to a processing unit which will handle all the baseband necessary computations to achieve the receivers PVT.

3.2 Software

Although the focus of this work is on the hardware required to convert the satellite signals into a digital data stream, it is important to consider the algorithms for software processing of the GNSS signals. The signal processing for satellite navigation systems is based on a channelized structure, as shown in Figure 3.2.

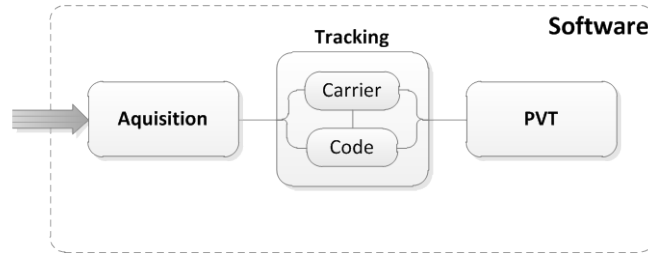


Figure 3.2 – Block diagram of a receiver channel.

The acquisition gives rough estimates of the signal parameters. These parameters are refined by two tracking blocks; after that the navigation message can be extracted and pseudoranges computed.

3.2.1 Software signal acquisition

The purpose of acquisition is to provide rough estimates of the carrier frequency and code phase from the visible satellites signals.

Doppler Effect is a consequence of the relative motion of the satellites, resulting in frequency deviations up to ± 10 kHz, in worst cases. This effect is important when generating a local signal to remove the incoming carrier from the signal.

One of the three standard methods of acquisition is *Serial Search Acquisition*. The algorithm is based on multiplication of generated PRNs and carrier signals. A PRN code is generated and multiplied by the incoming signal prior to be multiplied again by a generated carrier signal. This method performs one frequency sweep of IF in the range of ± 10 kHz in steps of 500 Hz and a code phase sweep over 1023 different code phases, performing a total of

$$Combinations = \left(2 \cdot \frac{10,000}{500} + 1 \right) \cdot 1023 = 41 \cdot 1023 = 41,943 ,$$

making this method very exhaustive.

The second method of acquisition parallelizes the search for the frequency, named *Parallel Frequency Space Search*. Initially this method is identical to the *serial search method*, a generated PRN code is multiplied with the incoming signal. After the code multiplication, the signal is transformed into the frequency domain through a Fourier transform. An efficient tool for that is the fast Fourier transform (FFT). If an alignment occurs within the PRN code, the output of the FFT will have a peak at the IF plus Doppler offset frequency. The absolute values of all components are calculated in order to determine the frequency of a possible peak.

The previous method parallelized the frequency space search eliminating the necessity of searching through the 41 possible frequencies. The goal of the Parallel Code Phase Search Acquisition is to parallelize the code phase.

Instead of multiplying the input signal with a PRN code with 1023 different code phases as done in the *serial search acquisition*, this method makes a circular cross correlation between the input and the PRN code without shifted code phase.

Primarily the incoming signal is multiplied with a generated cosine and sine carrier wave, obtaining an I and a Q signal. These two are combined as a complex input to the Fourier transform.

A PRN code is generated with zero code phase. Further, a Fourier transform is performed, and the result is complex conjugated.

This complex conjugate is multiplied with the result of the Fourier transform, finally this result is given to an inverse Fourier transform. As for the case of the FFT, for the output of the IFFT the absolute value also needs to be computed for all components. If a peak is present, the index of this peak symbolizes the PRN code phase of the incoming signal.

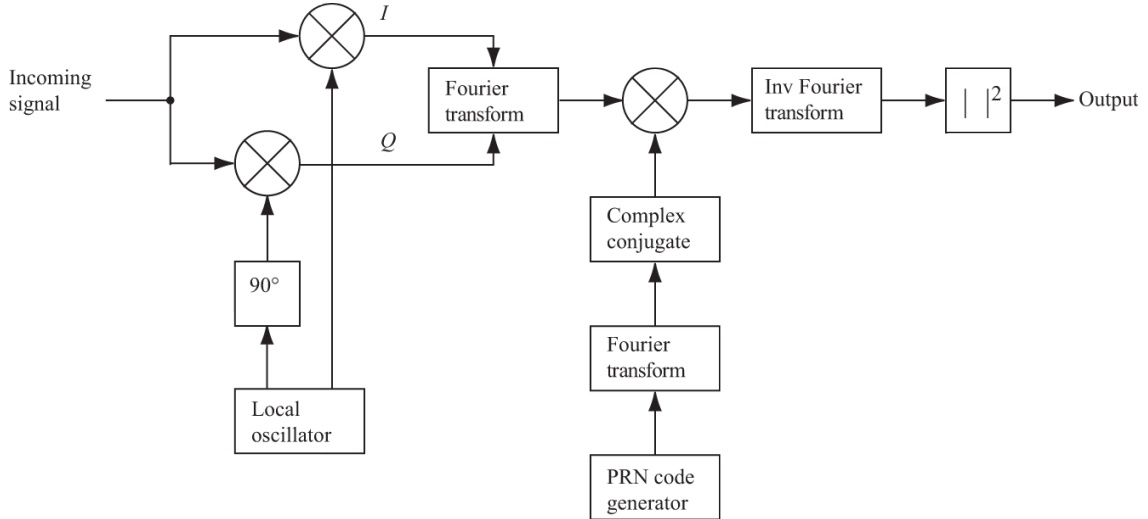


Figure 3.3 - Parallel Code Phase Search algorithm [2].

One difference in this method is that only one PRN code needs to be generated for each acquisition.

A comparison between the three methods execution time, taken from [2], can be seen in Table 3.1. The table also includes the number of combinations the algorithm has to perform.

Table 3.1 - Execution time comparison between acquisition methods.

Method	Execution time [time units]	Combinations	Complexity
Serial search	87	41943	Low
Parallel frequency space search	10	1023	Medium
Parallel code phase search	1	41	High

3.2.2 Software signal tracking

The acquisition provides rough parameters of the frequency and code phases. The main purpose of tracking is to refine these values, keep track, and demodulate the navigation data.

The receiver must first replicate the PRN code that is transmitted by the satellite, and then it must shift the phase of the replica code until it correlates with the satellites PRN code. It also must detect the satellite in the carrier phase dimension by replicating the carrier frequency plus Doppler induced effect. Figure 3.4 depicts a generic digital receiver channel.

The IF is removed from the carrier, plus carrier Doppler, by the replica carrier signals, also plus carrier Doppler, to produce in-phase (I) and quadrature (Q) sampled data. Any misalignment in the replica carrier phase with respect to the incoming satellite signal carrier phase produces a phase angle of the prompt I and Q vector magnitude, so that the amount and direction of the phase change can be corrected by the carrier tracking loop. When the phase-locked loop (PLL) is phase locked, I and Q signals are then correlated with early, prompt, and late replica codes. As a result, there are three replica code phases designated as early (E), prompt (P), and late (L). E and L are typically separated in phase by 1 chip and P is in the middle. When the prompt replica code phase is aligned with the incoming satellite code phase producing maximum correlation, the early phase is aligned a fraction of a chip period early, and the late phase is aligned the same fraction of the chip period later. Any misalignment in the replica code phase with respect to the incoming satellite code phase produces a difference in the vector magnitudes of the early and late correlated outputs so that the amount and direction of the phase change can be detected and corrected by the code tracking loop.

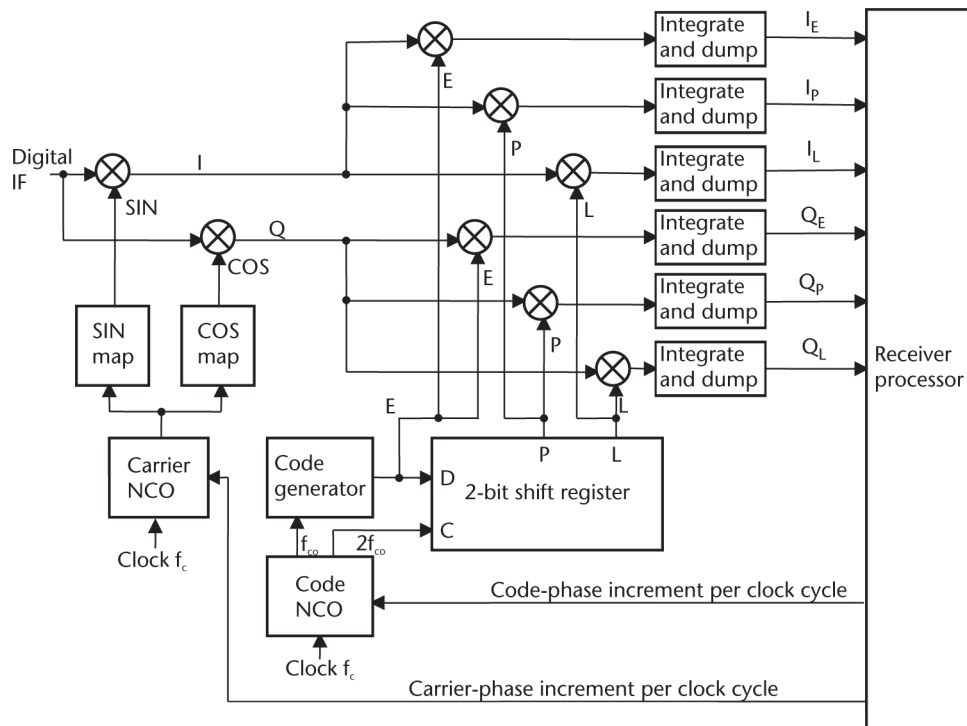


Figure 3.4 - Generic digital receiver channel [6].

With the demodulated navigation data, the subframes can be acquired, then the ephemeris are obtained, and finally the pseudoranges as well. This ephemeris data and the pseudoranges are used to determine the receiver's position on Earth using trilateration.

3.3 SDR Receivers

The GPS software radio has been firstly fully described by Dennis M. Akos in 1997, in his PhD dissertation [5]. In his work a GPS and GLONASS broadcast novel bandpass sampling front-ends have been proposed and successfully demonstrated, however the hardware and software implementation have been based solely on the GPS broadcast. The hardware front-end implemented is depicted in Figure 3.5.

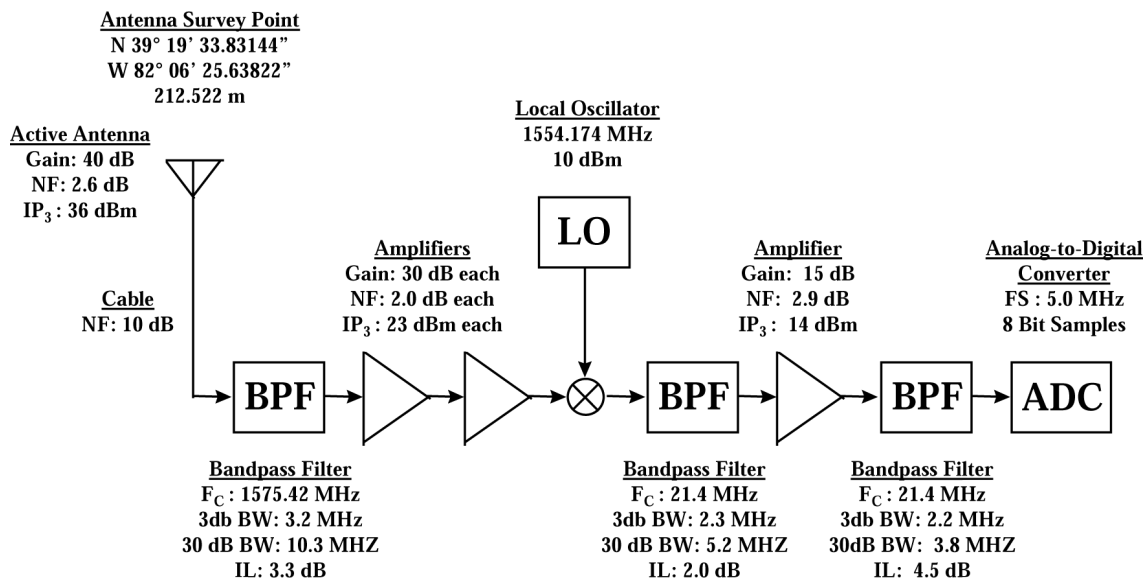


Figure 3.5 - GPS Radio front-end Design [5].

This design uses a downconversion stage with bandpass sampling. The 21.246 MHz IF is sampled at 5.0 MHz with 8 bit accuracy.

For signal acquisition, the technique used was chosen concerning the most rapid acquisition rate, making *parallel code phase search* the chosen technique, however other methods were described in this work.

Once knowing the acquisition parameters these were transferred to two tracking loops, consisting on a Costas (carrier tracking) loop followed by the code tracking loop early/late noncoherent delay lock loop. After the words were decoded and their validity was confirmed, a position solution was obtained. This position, when confronted with the actual reference coordinates, had an error of 59.6 m.

This implementation has been processed using the MATLAB programming language, providing the highest level of flexibility at the expense of reduced computational efficiency. One second of the GPS data samples required approximately 45 seconds to process on an Intel Pentium Pro 200 MHz microprocessor.

In 2007, in Italy, a SDR GPS and Galileo receiver has been presented [11]. The proposed architecture has been implemented on a hybrid FPGA/DSP board Sundance SMT8036, Figure 3.6.

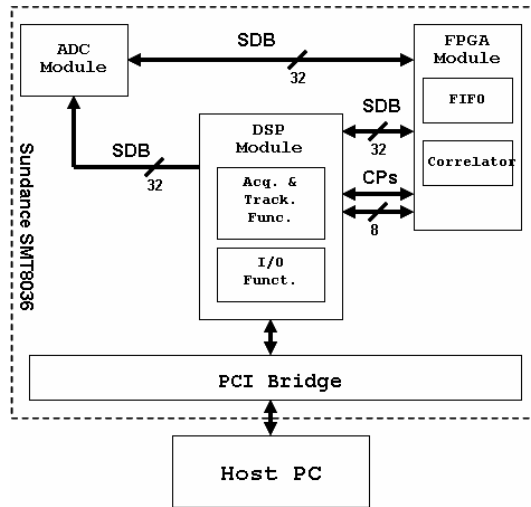


Figure 3.6 - Sundance SMT8036 functional scheme [11].

The calculation of correlations was made by means of dedicated hardware logic, FPGA, whereas discriminators and loop filters are software implemented, DSP.

Real-time constraint were taken in consideration, and a memory buffer was implemented between the analog-to-digital converter (ADC) and the correlators, to de-couple the time instants in sampling frequency of the ADC from the correlator, since the DSP isn't able to perform all the signal processing at the rate of the ADC sampling. The receiver had 24 channels, resulting in 24 correlators in parallel for the tracking of multiple satellites. This receiver had results with a low sampling rate, opening opportunities to extend it for E5 bandwidths where larger sampling rates are needed.

More recently, at the School of Electronic Information Engineering, BeiHang University in China, a real-time SDR GPS receiver was implemented using a Texas Instrument TMS320C6416 processor board [12].

In this architecture a Zarlink GP2015 IC was chosen as the RF front-end, due to its digital and analog Intermediate Frequency (IF) output support. Digital output wasn't used for any means, although its future purpose was to give support for multi-constellation. In front-end's considerations, the GPS L1 signal was down converted to 4.309 MHz prior to being sampled at 12 MHz by an ADC AD9283 from Analog Devices. The schematic of the hardware implementation is depicted in Figure 3.7.

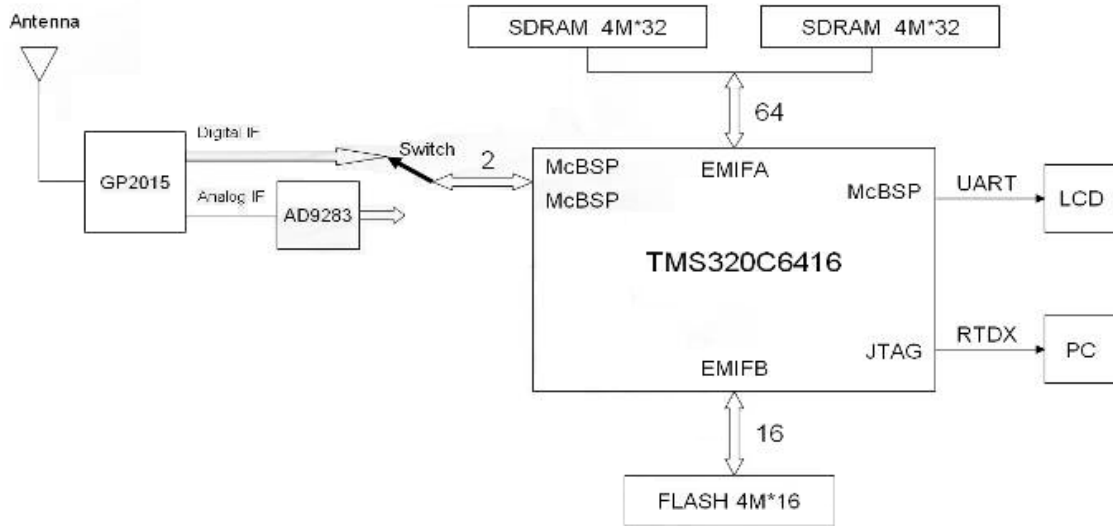


Figure 3.7 - Schematic of the hardware implementation [12].

The acquisition method implemented was also parallel code phase search for identical performance reasons.

However, a different approach to the software correlator was taken, intended to reduce the correlator complexity.

An equal-length C/A code method was implemented. Initially the unmatched length of the C/A code is recorded. The data transition bit might occur at CA code phase, so the division point is set at code phase plus unmatched length. The correlation is divided into two parts by a transition bit. If the sign of two parts is different, data transition occurs and correlation result is the minus result of two parts. If the sign of two parts is identical, the data transition has not occurred and correlation result is the sum result of two parts.

Tests with simulated data and real data were performed, although a comparison between these was not presented.

A real-time multi-frequency software GNSS receiver capable of receive and track signals from all-in-view GPS, Galileo, GLONASS and satellite-based augmentation system (SBAS) is in a third stage of development [13]. The receiver was developed at the *Institute of Geodesy and Navigation* at the *University FAF Munich* and is intended to run on a conventional PC, with the limitation of processing GPS L1 signals only. The receiver is implemented in C++ and the block diagram depicted in Figure 3.8.

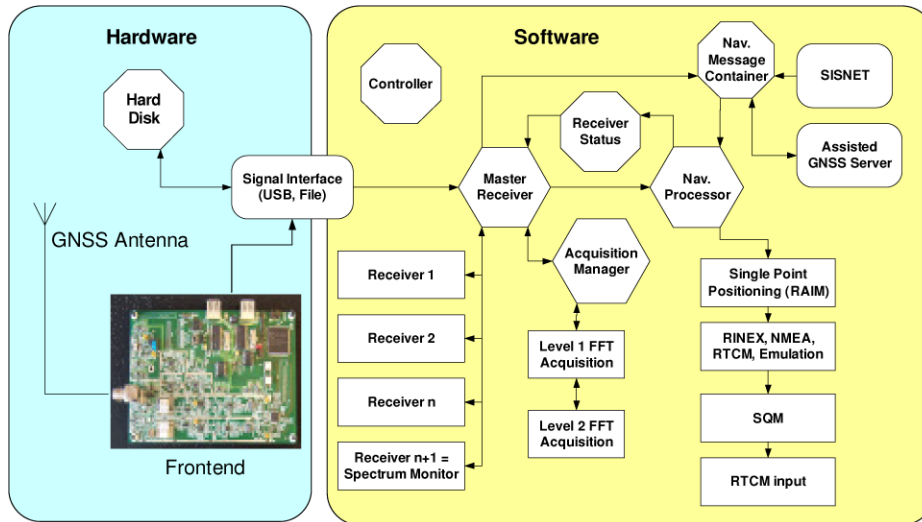


Figure 3.8 - ipexSR architecture [13].

This implementation has several receiver units, each tracking signals corresponding to a specific GNSS service. One single receiver unit comprises a number of different channels, each tracking on a satellite signal.

A specific unit has been dedicated to acquisition, composed of two fast Fourier transform levels. Level 1 is responsible for high-power signals while level 2 is responsible for high-sensitivity acquisition.

The tracking approach in this project is called *Code Continuous Reference Waveform*, this tracking method correlates the received signals with user definable reference functions derived from the pseudo-random noise (PRN) code sequence to optimize the tracking performance.

The receiver is provided with a graphical user interface with detailed information about acquisition, tracking and various monitoring tools.

3.4 GNSS Software Receiver

At this point, a complete combined GPS/Galileo/GLONASS SDR receiver is fully described. Thus execution of all the digital signal processing algorithms required for implementing the GNSS functionality in software is possible. The first open source software approached was the *SoftGNSS v3.0* [14], a complete GPS software receiver implemented in *MATLAB*, initiated by the *Danish GPS Center* and the *Aalborg University*, presented and described in [2]. The receiver is able to perform acquisition, code and carrier tracking, navigation bit extraction, navigation data decoding, pseudorange estimation and position computations for GPS navigation system. This software is able to process signal records with a various set of user defined settings. The data flow and the *MATLAB* functions used by *SoftGNSS* project receiver are depicted in Figure 3.9. Initially, a small section (few ms) at the start of the data file is read and is transferred to the acquisition algorithm. The acquisition searches for any presence of GPS signals. It evaluates the Doppler carrier shift and the C/A code phase for any satellite signal contained. After initialization of the channels (one channel per acquired signal, capable of up to 12 channels), block samples are read from the recorded file and passed to the tracking algorithm. The tracking function tracks signals for all enabled channels, detects bit boundaries, stores bits of the navigation data, and

decodes the data. A function identifies the start of a subframe, determines the signal transmission time, and estimates all pseudoranges. Then ECEF coordinates are computed and converted to a specified coordinate system. Finally, the results of acquisition, tracking, and positioning are plotted.

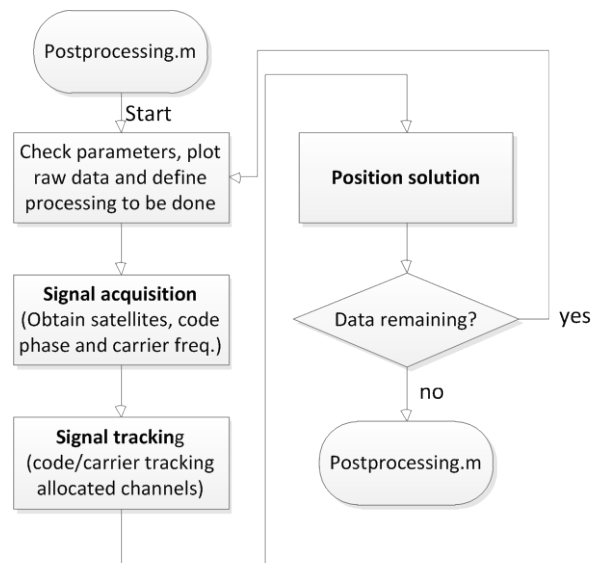


Figure 3.9 – SoftGNSS v3.0 receiver flow diagram [2].

Based on the *softGNSS v3.0* software, *softGNSS v3.0* GLONASS version is a port from the first software, updated and converted to *Scilab*, an open-source implementation analogue of *Matlab*, by Artyom Gavrilo owner of the website <http://gnss-sdr.com>. The structure of the software is maintained.

Other open source software GNSS receivers are available, although they weren't used since none of them brings any new features. Some of most notable are:

- *GPS-SDR* is a GPS L1 C/A code receiver, highly modular, multithreaded enabled, C++ application.
- *SoftOSGPS* project is a 12 channel GPS L1 receiver. The software goes from acquisition all the way to processing the navigation message and producing position and velocity fixes. The source code is written in C.
- *GPSTk* is sponsored by *Space and Geophysics Laboratory*, within the Applied Research Laboratories at the *University of Texas at Austin*. It's a GPS only platform independent software receiver. Platform independency is achieved through the use of the ISO-standard C++ programming language.

3.5 Summary

The circuit design flow of a GNSS SDR receiver and an overview of its functionality, as well as a brief description of the required signal processing has been presented. SDR receivers were presented since its first proposal and successful implementation up to nowadays, with the purpose of better understanding the progress and diverseness of satellite SDR implementations. Digital processing softwares used to test and validate the system were briefly described. The next section describes the development of the GNSS hardware front-end.

4 GNSS Front-end Design

4.1 Introduction

Given the basic GNSS receiver architecture in the previous section, and considering the objective requirements of an open-architecture, the following architecture design was approached for the system.

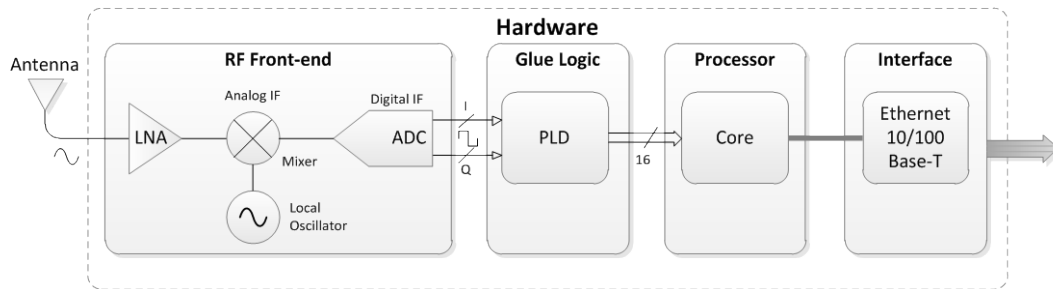


Figure 4.1 – Purposed GNSS SDR sampler architecture design.

Since the digitalization is done at a high rate, some glue logic must be used to interconnect the Front-End to processor core, the best choice is the implementation of a data packer in a PLD. A PLD is a programmable logic device, i.e., it is a digital logic IC which can be programmed with custom internal logic circuits.

The processor is the core of the hardware since it has to implement the TCP/IP Stack, manage the interface with the user, capture and transmit the digitized samples, etc.

At the final stage, there is the Ethernet interface, with an Ethernet PHY (physical layer) to interconnect the processor with a modular RJ45 connector in order to transmit capture data to a processing unit which will handle all the necessary computations to achieve the receivers PVT.

This section will explain in detail the considerations and design of every block of this architecture.

In order to perform an experimental validation of the proposed hardware architecture, a preliminary version was developed which showed a full potential for the proposed work and even enabled the implementation of additional features. This preliminary system was manufactured and assembled in the IST Taguspark rapid prototyping facilities. After system validation, a reworked prototype named EtherGNSS v3 was designed and manufactured.

4.2 EtherGNSS v3

A review of current state-of-the-art GNSS RF front-end chipsets is required to establish the starting point in the creation of any RF receiver.

Many semiconductor companies offer a RF Front-end chipset. Not all systems are fully integrated, since the LNA is external in some cases. A comparison between five commercial receivers was made. A comparison of characteristics for the most suitable front-ends from different manufacturers is summarised in Table 4.1.

Table 4.1 - Receivers commercial IC front-ends, adapted from [15].

Reference	VCC [V]	Power [mW]	Gain [dB]	NF [dB]	Bit nr.	Avg. Price [€]	Comments
MAXIM MAX2741	3	90	80	4.7	2/3	5.20	Internal LNA, double conversion, AGC
MAXIM MAX2769	3	54	96	1.4	2/3	3.00	Two internal LNA, single conversion, ready for Galileo and GLONASS
SiGe SE4120S	3	30	18.5	1.65	2	2.90	Internal LNA, ready for Galileo, multibit serialized I/Q output
ST STA5630	1.8	25	?	?	3	n/a; <10	N/A. Internal and external LNA, ready for Galileo
Zarlink GP2015	3	173	120	9	2	?	External LNA, triple conversion, AGC

The Zarlink GP2015 front-end presents the best gain-to-noise figure ratio with a gain of 120dB at a cost of a high noise figure (NF) of 9 dB and high power consumption of 173 mW. The MAXIM MAX2769 also has a high gain of 96 dB although with a low NF of 1.4 dB with a consumption of 54 mW.

The MAXIM MAX2741 has high power consumption and a high price per unit, which dismisses its implementation viability.

The SiGe SE4120S, ST STA5630 and MAXIM MAX2769 are dual front-ends currently ready for GNSS applications. Unfortunately the ST STA5630 doesn't have more information available to the public, however according to the manufacturer website the IC is at volume production.

Through this analysis, it is possible to ascertain that the suitable solution to implement in this work is MAXIM MAX2769, with a good gain-to-noise figure ratio, reasonable power consumption and the compatibility to the three satellite navigation systems, at a relatively low price.

4.2.1 GNSS RF Receiver

The Maxim MAX2769 [16] is a versatile and flexible GNSS receiver chip offering the highest performance and integration, making it one of the best solutions on the market. It integrates all the required RF to low-IF digital conversion, as depicted in Figure 4.2.

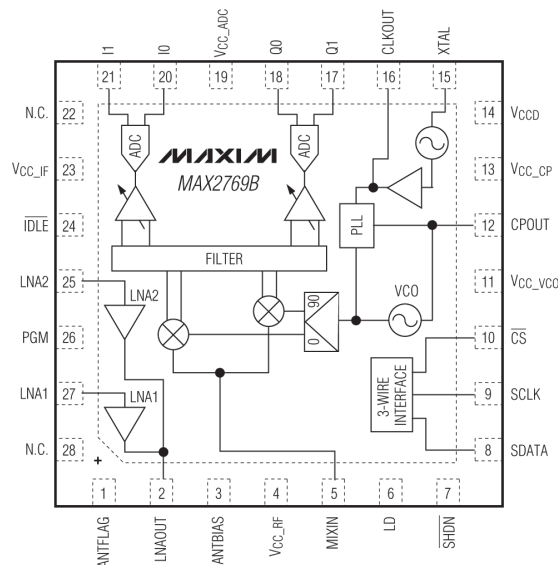


Figure 4.2 - Typical application circuit for MAX2769.

Current consumption at 2.8 V ranges from 13 mA to 18 mA depending on the configuration, with a total cascade NF as low as 1.4dB.

The device integrates two internal LNAs, one for use with passive antennas set to high-gain mode and the other LNA for use with an active antenna, set to work in low-gain mode. It also integrates a low-dropout regulator to bias an active antenna, this regulator incorporates a sensor that gates the LNA selection, depending on the type of antenna connected. The LNA1 is typically used with the passive antenna, in default mode, has a Power Gain of 19dB and a NF of 0.83dB. On the other hand, the LNA2 is typically used with the active antenna with a power gain of 13dB and a NF 1.14dB. Also in default mode the bias current supplied is 4 mA.

The signal is then downconverted directly by a quadrature mixer, using the integrated 20 bit, sigma-delta, fractional-N frequency synthesizer, represented in Figure 4.3, together with a 15 bit integer divider to achieve virtually any desired IF up to 12MHz. The PLL filter is a classic C-R-C network at the charge pump output, the filter component values used aimed at a comparing frequency of 1.023 MHz.

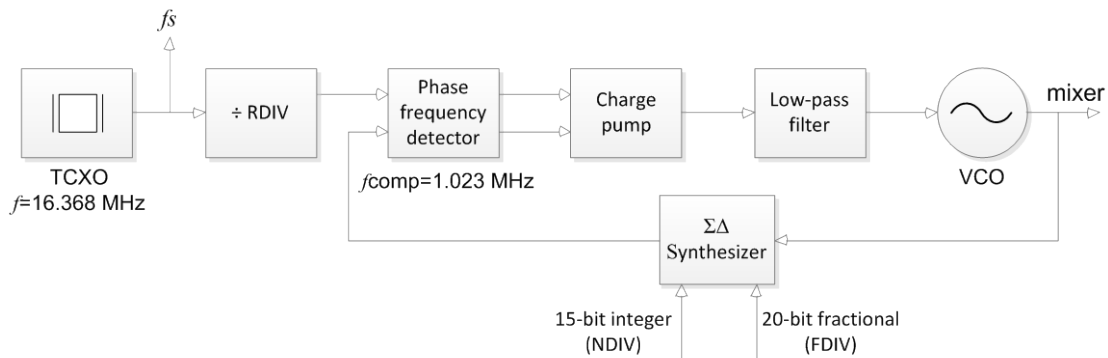


Figure 4.3 – PLL frequency synthesizer.

Then a baseband filter can be programmed to be a complex bandpass filter, with a configurable center frequency (6bits register) and a bandwidth of 2.5MHz, 4.2MHz or 8MHz, or a low-pass filter with a configurable bilateral bandwidth of 2.5MHz, 4.2MHz, 8MHz or 18 MHz.

The internal ADC has a selectable output of one to three Real and Complex (I&Q) bits, with a maximum sampling rate of approximately 50Msps. All internal configurations are made via 3-wire synchronous serial SPI protocol.

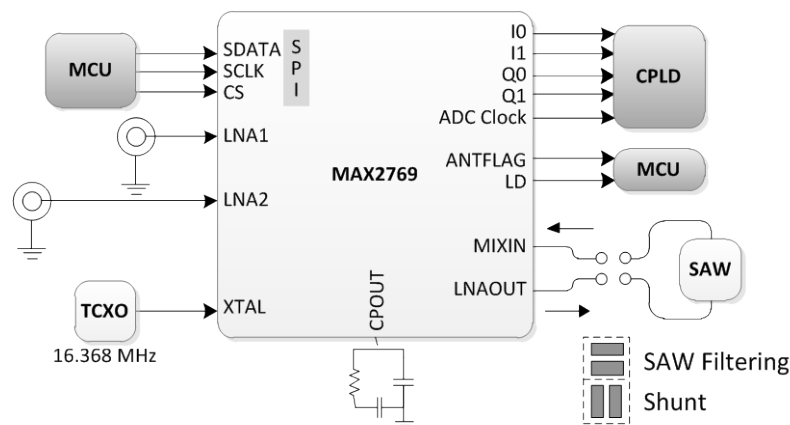


Figure 4.4 – Maxim MAX2769 implementation and connections.

The integrated reference oscillator enables operation with either a crystal or a temperature-compensated crystal oscillator (TCXO). A 16.368 MHz reference clock for down-conversion and sampling is used. 16.368 MHz is a commonly used frequency since it is 16 times the 1.023 MHz C/A chipping rate for GPS, multiplied by 96.25 to get the 1575.42 MHz L1 frequency or multiplied by 75 to get the 1227.60 MHz L2 frequency.

External filtering can be done on the amplified RF signal before the mixer downconversion. Setting the filtering mode is done by asserting the jumper in horizontal position driving the RF signal to a narrow band GNSS SAW filter, useful specially when using passive antennas which don't have signal conditioning. Asserting the jumper in vertical position will shunt the LNA to the mixer's input directly.

Since the output load of the on-chip LNA is 50 Ω and the SAW Filter is expecting a 50 Ω source, there is no need for a matching network between LNA output and SAW filter input. Its nominal transfer function is included in Appendix I.

4.2.2 CPLD

The purpose of the use of a programmable logic device is to handle the high rate serialized output digital samples handed over by the front-end module, to a parallel format prior to being transmitted to a communication module. This glue logic implementation is a switchable 1 to 4 input, serial-in, parallel-out (SIPO) / parallel-in, parallel-out (PIPO), depending on the number sampling bits, where the CPLD is used to buffer the ADC data. This adaptation is necessarily made by a PLD due to the high sampling rate of the output in both the I and Q channels.

A CPLD is an electronic component used to build reconfigurable digital circuits, and has highly flexible routing resources. CPLDs are arranged as an array of clusters, linked by horizontal and vertical routing channels. With their easy-to-understand AND-OR structure, CPLDs offer a single-chip solution with low pin-to-pin delays. Once programmed, the design can be locked providing some security to the design.

In-system-programmability is a must for nowadays designs, and the ability to maintain pin-outs during design modifications ("pin-locking") is crucial. The limited capacity (<500 flip-flops) means that most CPLDs are used for "glue logic" functions.

The EtherGNSS CPLD device is an Altera MAX V 5M80ZE64C5N. His architecture includes an array of logic elements (LE) grouped in logic array blocks, memory resources (non-volatile flash and LE RAM), digital PLLs, global signals (clocks or control signals), and a generous amount of user I/O. It is designed to maximize performance and minimize power by using the most efficient, direct connection from input to logic to output. Table 4.2 lists MAX V 5M80ZE64C5N features.

Table 4.2 - MAX V 5M80ZE64C5N Features.

LEs	Flash Memory Size [bits]	Global Clocks	Internal Oscillator	Max IO pins
80	8,192	4	1	64

The CPLD supports two I/O banks, each bank supporting all the LVTTTL, LVCMOS, LVDS and RSDS standards. Also, the output buffer for each I/O pin has a programmable output slew-rate control that can be configured for low noise or high-speed performance. A low slew-rate reduces system noise, however adds a nominal output delay to rising and falling edges.

Input pins “Clock”, “Reset” and the 4 ADC sampled bits from the RF Front-end are located in bank 1, and two additional input bits to define frequency division upon the configured sampled bits (from 1 to 4), are located in bank 2. “Clock” and “Reset” are routed to the global clock network. The global clock network can provide clocks for all resources throughout the entire device. Output pins are configured with Slow Slew-Rate in order to minimize the system noise, are set to 3.3 V-LVTTTL/LVCMOS, and are distributed in bank 1 and bank 2, giving use of the flexible pinout assignment of the CPLDs with the purpose of simplifying the design layout.

A write strobe signal, named “CPLD Clock” is generated by the division of the input frequency, a logic “high” indicates that a new packet 16-bit array is ready to be captured from the CPLD.

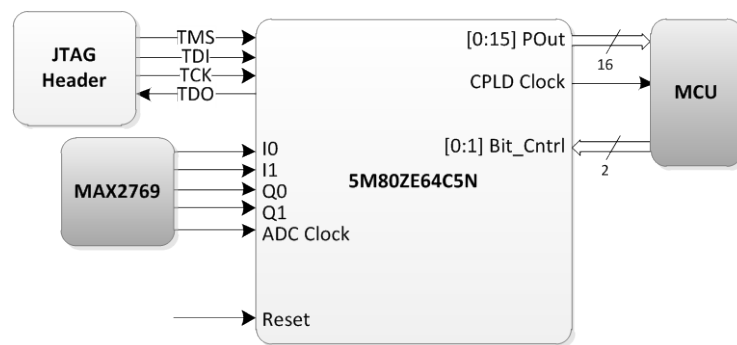


Figure 4.5 - MAX V 5M80ZE64C5N CPLD scheme.

The MAX V 5M80ZE64C5N is programmed in-system through the industry standard Joint Test Action Group (JTAG) 4-pin IEEE Std. 1149.1 interface, that also provides the Boundary-Scan Support for testing. A JTAG header is available for direct download of designs with Altera download cables or compatible programmers.

4.2.2.1 CPLD Glue Logic

Designing digital logic circuits for a CPLD can be done through schematic design of conventional logic symbols and circuits, as well as descriptor languages, like VHDL or Verilog HDL, being the descriptor languages a more efficient way of describing logic behaviour.

For the designing, every manufacturer provides their design software. Quartus II Web Edition is a free version of Quartus II, a development environment produced by Altera for analysis and synthesis of HDL designs, which enables the compilation of designs, performs timing and stimuli analysis, in addition to configuring the target device with the programmer.

A Verilog HDL has been designed. This design implements a SIPO/PIPO glue logic to interface the RF front-end outputs to the microcontroller. Due to the front-end configurable multi-bit output and the microcontroller fixed 16-bit input, there was the need to develop a dynamic design, with independent

inputs although with a fixed output. To achieve this interface a Moore state-machine was designed, see Figure 4.6.

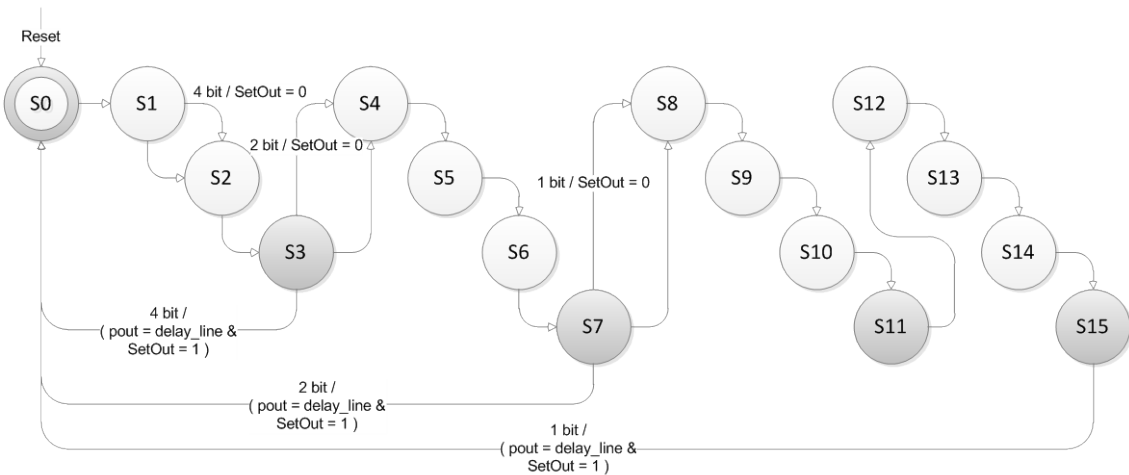


Figure 4.6 – CPLD Moore State-Machine Design flowchart.

The state-machine input capture bits are controlled by a 2-bit control *wire input*. This *wire input* controls the number of bits (1, 2 or 4) shifted into a 16-bit delay line, as well as reinitializes the State-Machine when the delay line is full. With this configuration, four working modes were designed: 1-bit only I sampling; 2-bit only I sampling; 1-bit I & 1-bit Q sampling and 2-bit I & 2-bit Q sampling.

Whenever the state-machine is reinitialized to state “S0”, the delay line is assigned to the parallel output and a write strobe is set High (1) indicating that there is a new data. The write strobe is set to Low (0) in the mid-cycle of the state-machine creating a write strobe with 50% duty cycle. The function of each signal is described in Table 4.3.

Table 4.3 – CPLD design signal description.

I/O	Type	Bits	Name	Function
Input	Wire	4	pin	Parallel Input
Input	Wire	2	bits	Number of capture bits
Input	Wire	1	clk	Clock signal
Input	Wire	1	reset	Reset signal
-	Reg	16	delay_line	Shift-Register array
Output	Reg	16	pout	Parallel Output
Output	Reg	1	setOut	Write Strobe

Using the 16.368 MHz TCXO signal, which corresponds to the ADC sampling frequency of the RF front-end, the frequency conversion of the write strobe, for 1, 2 and 4 input bits was verified.

Figure 4.7 shows the output write strobe signal (RED), for a 1-bit sampling corresponding to an input frequency division of 16, that is 1.023 MHz. For 2-bit sampling, as seen in Figure 4.8, the frequency division ratio is 8, corresponding to a write strobe of 2.046 MHz. Lastly, for 4-bit sampling, as seen in Figure 4.9, the frequency division ratio is 4, corresponding to a write strobe of 4.092 MHz.

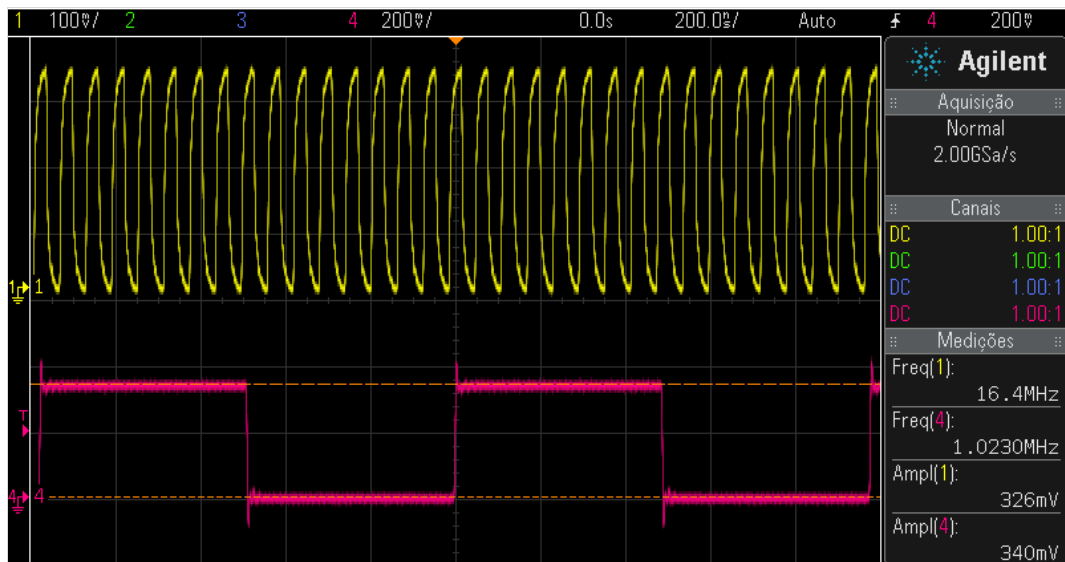


Figure 4.7 – CPLD 1.023 MHz write strobe (Red). 16.368 MHz input (Yellow) divide by 16, for 1-bit input.

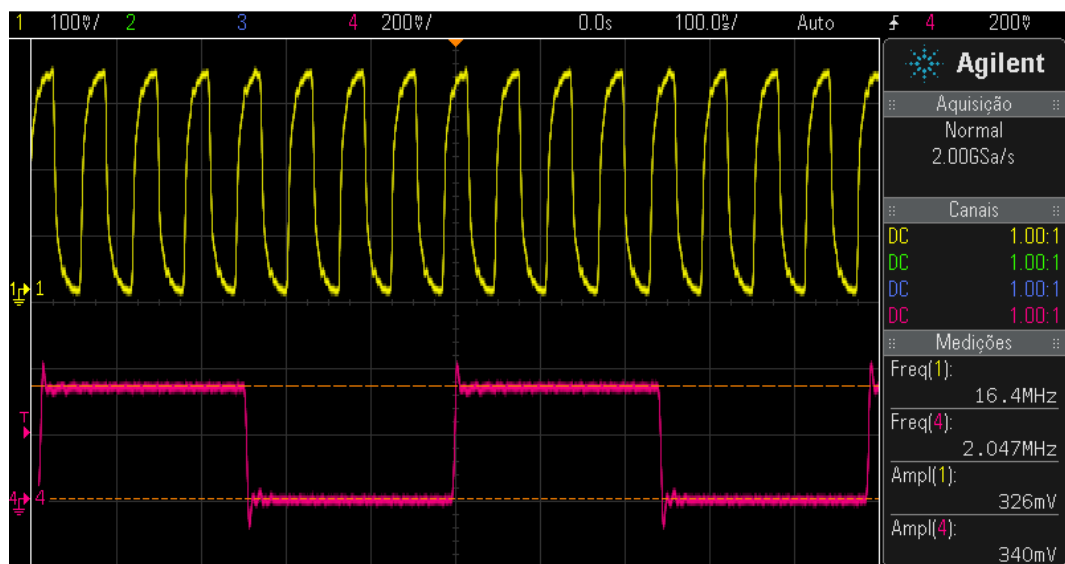


Figure 4.8 – CPLD 2.046 MHz write strobe (Red). 16.368 MHz input (Yellow) divide by 8, for 2-bit input.

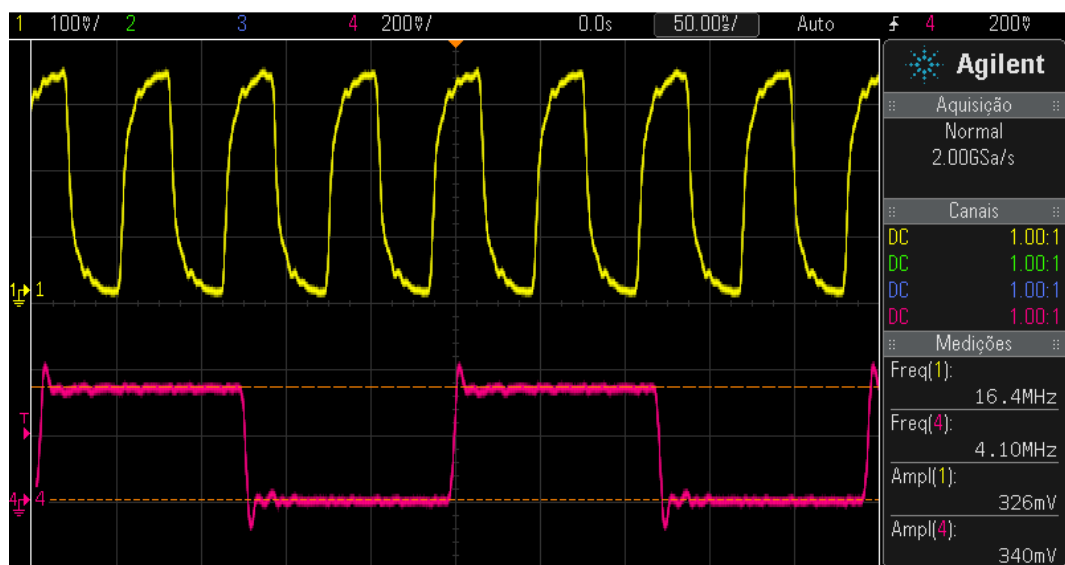


Figure 4.9 – CPLD 4.092 MHz write strobe (Red). 16.368 MHz input (Yellow) divide by 4, for 4-bit input.

4.2.3 Processor Core

For the choice of the processor core some consideration needed to be taken. Since there is a data transmission requirement of an IP interface, a processor with MAC protocol is required. In addition, the processor may serve performance-critical role due the high throughput of data. Therefore the right processor for performance and cost developed for embedded systems a microcontroller (MCU).

A MCU is a small computer on a single integrated circuit designed for embedded applications which incorporates a processor core, memory, clock and programmable I/O peripherals.

The EtherGNSS MCU core is a Microchip PIC32MX795F512L featuring a 10/100 Mbps Ethernet MAC controller and high-performance 32-bit RISC (Reduced Instruction Set Computing) CPU offering speed and performance at low cost. The MIPS32® M4K® Processor core is the heart of the PIC32MX795F512L MCU, fetches instructions, decodes each instruction, fetches source operands, executes each instruction and writes the results of instruction execution to the proper destinations [17].

Some of the main features of this MCU are listed in Table 4.4.

Table 4.4 - Microchip PIC32MX795F512L main features for implementation.

Max Speed [MHz]	Memory Size [KB]	RAM [KB]	DMA	SPI	Ethernet MAC	Internal Oscillator	IO Pins
80	512	128	8	4	10/100Base-T	8 MHz, 32KHz	85

The PIC32MX340F512L MCU has 512 KB of Flash and 32 KB of integrated RAM. In addition to the 10/100 Base-T controller, it features 4 channels of the synchronous serial SPI module used to program the RF front-end and 8 channels for the direct memory access (DMA) controller, used for data transfers between any memory mapped addresses without CPU intervention, for this work it is used to transfer the 16-bit samples from the CPLD. Furthermore it is capable of working with an 80 MHz clock, a considerable advantage over its competitors.

The programming and debugging of the MCU is done through the In Circuit Serial Programming pins, ICSP™, of the PIC32.

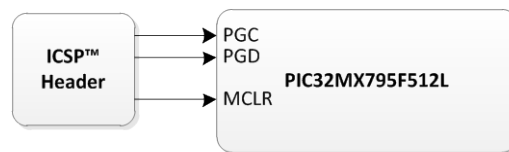


Figure 4.10 – PIC32MX340F512L in circuit serial programmer connections.

Microchip also offers a free licensed TCP/IP stack optimized for the PIC32 microcontrollers. The TCP/IP stack is modular in design and written in the 'C' programming language [18]. It includes a light-weight and high-performance implementation of the TCP and UDP transport layers, as well as other supporting modules, as depicted in Figure 4.11.

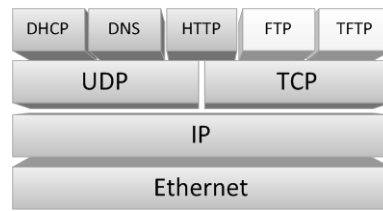


Figure 4.11 – Microchips TCP/IP stack architecture.

Furthermore its purpose is to serve as a reference guide to the features and APIs available in the TCP/IP Stack. Figure 4.11 also highlights some of the modules available, they are:

- TCP and UDP transport layer;
- ICMP, giving Ping query and response capability;
- Dynamic Host Configuration Protocol (DHCP) client for obtaining IP address and other parameters;
- Domain Name Service (DNS) client for resolving hostname strings to IP addresses;
- Hypertext Transfer Protocol (HTTP) server, for the Webserver implementation;
- Announce, Microchip Embedded Ethernet Device Discoverer server/client;
- Reboot Server, module for resetting the board remotely.

4.2.3.1 MCU software

Making use of the TCP/IP stack, an application was developed, see Figure 5.1. This application is responsible for performing stack specific operations including handling the transmission and reception of data packets, along with application specific operations. To ensure that no single task monopolizes control of the microprocessor, the code implements a dispatcher in a *cooperative multitasking* approach, in other words, with a preemptive scheduler, making periodic calls to task functions in order to prevent conflicts with the stack and guarantee maximum performance.

The application starts with the sequence of initializations, which includes initialization of application specific hardware followed by stack and application related components, core stack layers and application modules initialization. After the initialization is finished, the application enters an infinite loop which handles the application tasks. Within this loop a sequence of functions comprise the nucleus of the application. Primarily, any timed operations that the stack requires are performed, and the transmission and reception of data packets are handled or routed to the appropriate application protocol-level function. Then protocol application modules are called to process any tasks that are in queue. These protocol applications modules are paused during captures to maximize performance and guarantee zero packet loss. EtherGNSS's specific functions follow.

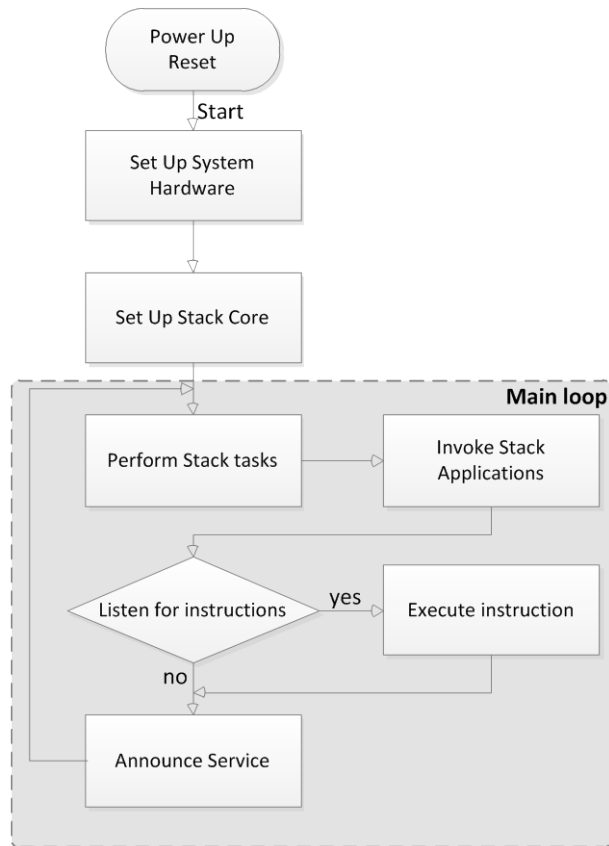


Figure 4.12 – TCP/IP Stack Main application high-level flowchart.

Since a high speed throughput of data is required, the application creates a UDP Server, see Figure 4.13. The UDP protocol is a simpler connectionless protocol that doesn't provide error checking and correction, avoiding the overhead of such processing at the network interface level. Three specific UDP sockets are used in this firmware, they are listed in Table 4.5.

Table 4.5 – Firmware specific UDP sockets.

Purpose	Communication Service	Access	Port	Description
Announce	Broadcast	In/Outbound	30303	Reply Announce requests and transmit IP changes
Instruction	Unicast	In/Outbound	32333	Capture instructions and retrieve acknowledges
Data	Unicast	Outbound	32334	Transmission of sampled data

Thus a UDP socket for inbound and outbound transmission of instructions is opened. Whenever a valid instruction is acquired their parameters are saved. If the type of instruction is a RF front-end configuration request, the correct parameters are transmitted via SPI protocol to the device. If a data acquisition is requested, the server socket is directed to the destination IP and the execution is permitted. This function is followed by an acknowledge return to the destination.

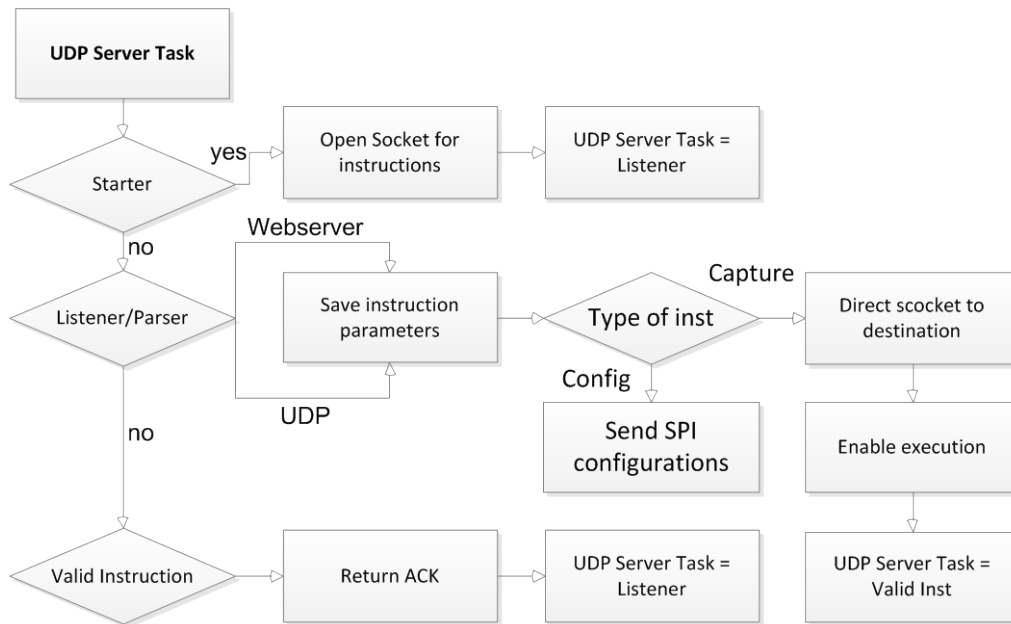


Figure 4.13 – UDP Server Task high-level flowchart.

The UDP Server Application executes instruction orders, see Figure 4.15. If the instruction is for data acquisition, a unicast socket is open for outbound transmission and DMA is enabled. The DMA channel captures 16-bit of data triggered by the CPLD clock in a circular buffer. The DMA buffer has the size of four 1470-byte UDP packets, this size guarantees that no data overlap occurs during data transmission.



Figure 4.14 – DMA 5880-Byte buffer representation.

An interruption is triggered when half and full buffer are reached. This set a *Data ready* flag and the firmware starts the application which creates a UDP packet and routes it for transmission. The second UDP packet of the half buffer is rapidly followed and the *Data ready* is cleared. The execution is finished with the disable of the DMA channel and the release of the socket for future communications.

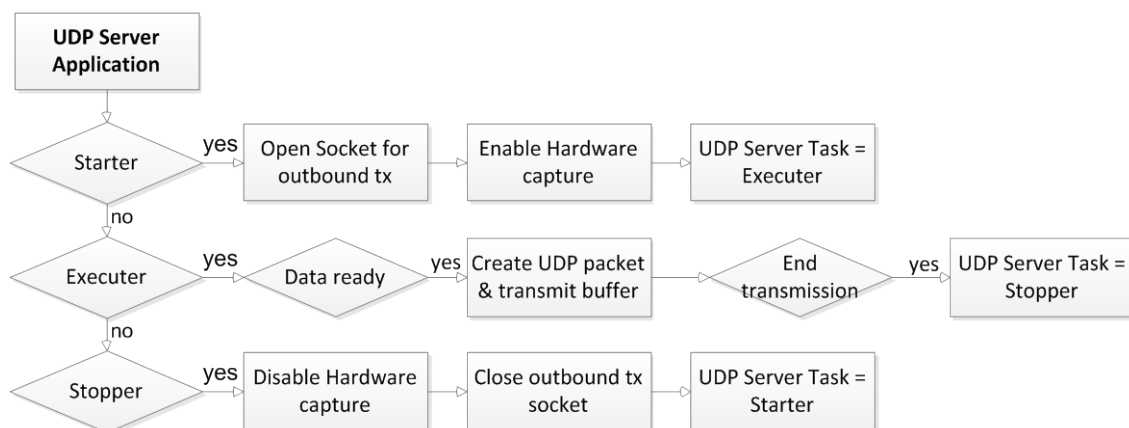


Figure 4.15 – UDP Server Application high-level flowchart.

The main loop is ended with an Announce Service, which detects changes in IP address and in that case broadcasts it for the listening clients.

Three data captures are supported by EtherGNSS v3. Timed capture, where capture is performed for a submitted period of time, in seconds. Size capture, a submitted size value, in KB, is captured. And direct stream to a destination IP.

4.2.4 Ethernet Interface

Data transmission is handled through the Ethernet protocol, also known by standard IEEE 802.3, the most common protocol used at the physical layer. Ethernet is the most popular and considered the networking topology standard for most computer connections.

The Ethernet protocol is made up of a number of components, such as the structure of Ethernet frames, the media access control (MAC) operation and its Physical Layer.

The MAC sub-layer acts as an interface to the network's physical layer. It emulates a full-duplex logical communication channel in a multi-point network, capable of providing unicast, multicast or broadcast communication service. The PIC MCU integrates an Ethernet Controller that provides a MAC block, responsible for the implementation of the MAC functions.

An integrated off-the-shelf solution was considered, given the obvious benefits of being a low-cost solution with implemented TCP/IP and MAC protocol layer and physical layer. This solution offers simpler implementation of the Ethernet interface, although with the cost of the reduced flexibility and performance. So this option was excluded, and a most flexible solution was approached with the use of an Ethernet physical layer transceiver (PHY).

An Ethernet PHY is the device that operates at physical layer to connect a link layer device, MAC, to a physical medium. It implements the hardware send and receive function of Ethernet frames. This industry standard interface is implemented based on the National Semiconductor DP83848M single 10/100 Ethernet transceiver. The DP83848M incorporates the Reduced Media Independent Interface (RMII). This interface is used to connect the device to a MAC in 10/100 Mb/s systems using a reduced number of pins. In this mode, data is transferred 2-bits at a time using the 50 MHz RMII REF_CLK for both transmit and receive.

Besides the RMII interface pins, there is the need to have a RMII management interface, since there is no signal to define whether the interface is in full or half duplex mode or whether the interface is in 10 or 100 Mbit/s mode, although both the MAC and the PHY need to be concordant. RMII management interface consists on, serial MDIO data I/O line and MDC data clock line signals. RMII interface is represented in Figure 4.16.

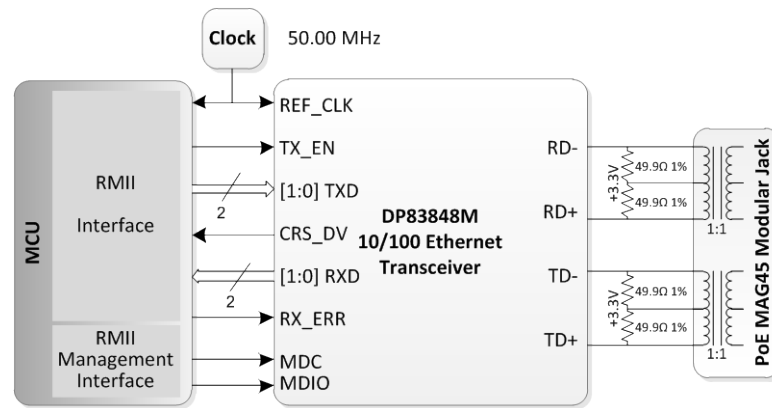


Figure 4.16 – DP83848M 10/100 Ethernet transceiver connections.

A single port RJ45 connector with integrated magnetics for 10/100Base-T PoE enabled applications is used. This modular jack has incorporated transformers (magnetics) to interface the Ethernet twisted pairs data lines, and a PoE polarity insensitive implementation using a diode bridge in a standard RJ45 jack.

To avoid reflections, termination resistor must be with the same value as characteristic impedance of cable. Since characteristic impedance of cables used in 10base and 100base is 50 Ohm, a 49.9 Ohm resistor is used to guarantee matching.

4.2.5 Power Supply

EtherGNSS board requires two voltage supplies. The low power MAX V CPLD accepts a 1.8 V external supply to power the device core directly. It also supports a multivolt I/O interface, set to 3.3 V-LVTTL/LVCMOS. All the remaining devices of the board use a 3.3 V supply voltage.

The Texas Instruments TPS767D318 [19] is a dual voltage regulator offering a fast transient response, low dropout voltages and dual outputs, each supporting up to 1 A, for split-supply applications. It also features a very low quiescent current and the output voltage tolerance is specified as a maximum of 2% over line, load, and temperature ranges.

It also provides internal current limiting, in normal operation approximately 1.7 A, and thermal shutdown protection if the device temperature exceeds 150°C. This protection prevents gross device failures.

EtherGNSS power can be supplied in the range +3.3 – +10 V in three ways:

- A regulated DC power supply connected to the DC jack;
- Two test-points for external supply of “+5 V” and “GND”;
- Power over Ethernet (PoE) technology.

PoE describes a system to pass electrical power safely, along with data, on Ethernet cabling, so the board doesn't need a separate cable for each need.



Figure 4.17 – Texas Instruments TPS767D318 supply connections.

In order to minimize coupling between different sections of the ICs, a star power-supply routing configuration was devised. There is a central V_{CC} node for both the output voltage supplies, where IC specific V_{CC} traces branch out from this node with a bypass capacitor as close as possible to each supply pin. This arrangement provides local decoupling at each V_{CC} pin.

4.2.6 Other features

EtherGNSS board also includes additional features. Three Light Emitting Diodes (LEDs) are present on the board: a green LED representing board's heart beat; a red LED for error and exceptions notification; a green LED debug indicator.

The board features a push-button for global board reset, this reset only applies to the digital circuits, leaving the RF Front-end in operation and preventing it to lose non-default or user defined configurations.

Moreover a 5-pin micro-pitch header is located in the outputs of the RF Front-end giving the ability to extract the signals, digital or analog, for external measurements or signal manipulation.

4.2.7 Impedance matching concerns and PCB layer stack-up

In order to establish a good signal path for the RF signals, general RF layout rules are needed to be considered during the PCB design. Regarding trace widths, *LenaCalc*, a tool of the RF design software "LENA – Line Network Analysis" [20], that allows the calculation of trace widths considering for a given PCB stack-up, Coplanar Waveguide over Ground (CPWG) for this implementation.

A CPWG provides better implementation isolation. This medium consists of a center conductor with ground planes on either side and below, see Figure 4.18. Via "fences" are recommended on both sides of the center conductor. This implementation causes the return currents induced on the top layer to be shorted to the underlying ground layer.

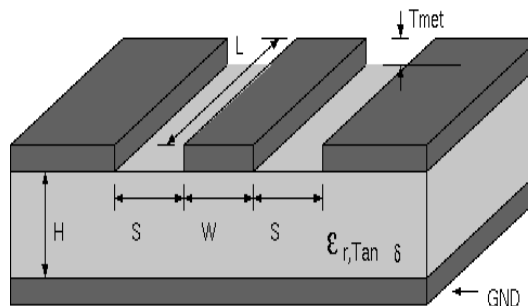


Figure 4.18 - Coplanar Waveguide Cross-section.

The LNA and Mixer input are matched to 50 Ω . Considering the pre-set characteristics of a 4-layer prepreg substrate, a resulting 0.55 mm center conductor width with 0.25 mm of clearance was defined on the RF PCB layout design.

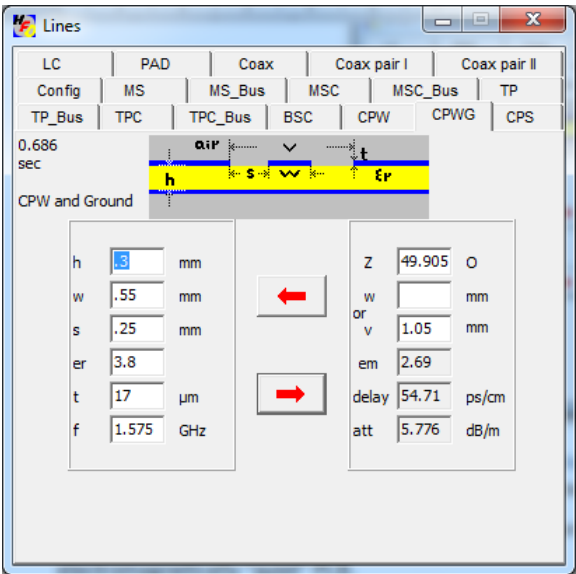


Figure 4.19 – LenaCalc calculated Coplanar Waveguide over Ground line characteristics.

Even more, via stitching was applied between ground layers, in order to maintain low impedance connections and short return loops through the board structure for developing high-performance, electromagnetically “quiet” PCB. The implemented layer stack-up is represented in Figure 4.20.

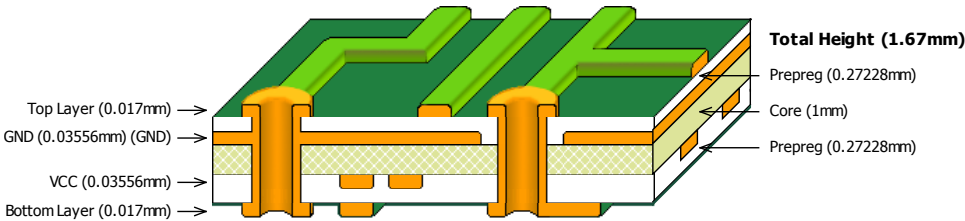


Figure 4.20 – PCB Layer stack-up.

Components are placed in the top and bottom layers as well as signal traces. The simplest way to improve the Electromagnetic Compatibility (EMC) performance of a four-layer board is to space the signal layers as close to the planes as possible, and use a large core between the power and ground planes [21]. This stack reduces signal loop areas and plane impedance and also helps to decrease the crosstalk between traces.

4.3 Budget and Prototyping

A market prospection was made with both European and Non-European PCB manufacturers, with the final choice considering the lowest price, and lowest price upon assembly. The final prototyping cost including the PCB and components is show in Table 4.6.

Table 4.6 – EtherGNSS v3 overall prototyping cost (without assembly).

	Unit price [€]	UN	Sub-total [€]
PCB	13.04	4	52.16
Tooling cost (one-time)	153.54		153.54
Parts & components	-	-	271.62
Total			477.32

In general for all the components and parts considered, the trade cost/quality was imperative.

The EtherGNSS v3 board is a 50x50 mm square and can be seen in Figure 4.21.

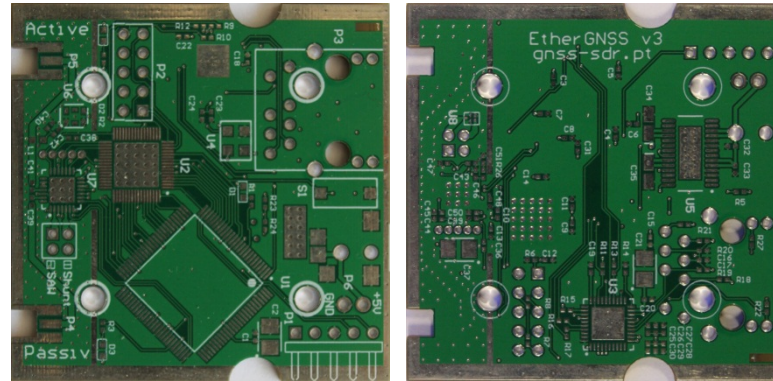


Figure 4.21 – EtherGNSS v3 PCB (Real Size Topside & Underside).

Representations of the layout of the EtherGNSS v3 are shown in Figure 4.22 and Figure 4.23.

The top assembly of the board includes these key features, as indicated in Figure 4.22:

1. MAXIM MAX2769 RF Front-End;
2. 16.368 MHz Front-End reference oscillator;
3. CPLD device;
4. PIC32MX795F512L 32-bit microcontroller;
5. 50 MHz Ethernet PHY oscillator;
6. PoE MAG45 Ethernet port;
7. Push-button for boards global reset;
8. 1.3mm pin DC socket for power supply;
9. Test-points for external power supply;
10. ICSP™ port for debugging;
11. JTAG CPLD in-system programmer;
12. External filter select jumper;
13. Header for Front-End outputs extraction;
14. MMCX Edge Connect antenna connectors;
15. Three indicator LEDs. Green Heart LED, Green debug indicator and Red exception indicator.

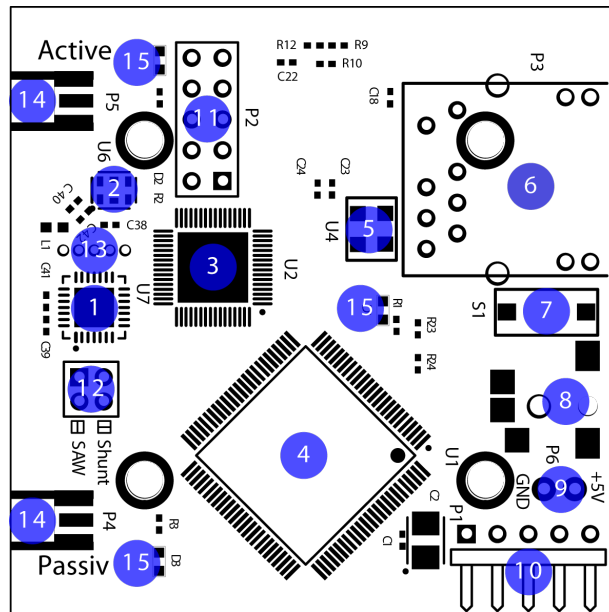


Figure 4.22 – EtherGNSS v3 Layout (Topside).

The bottom assembly of the board includes these key features, as indicated in Figure 4.23:

1. Regulated dual output +3.3V/+1.8V power supply for powering the board;
2. External Ethernet PHY;
3. GNSS SAW filter.

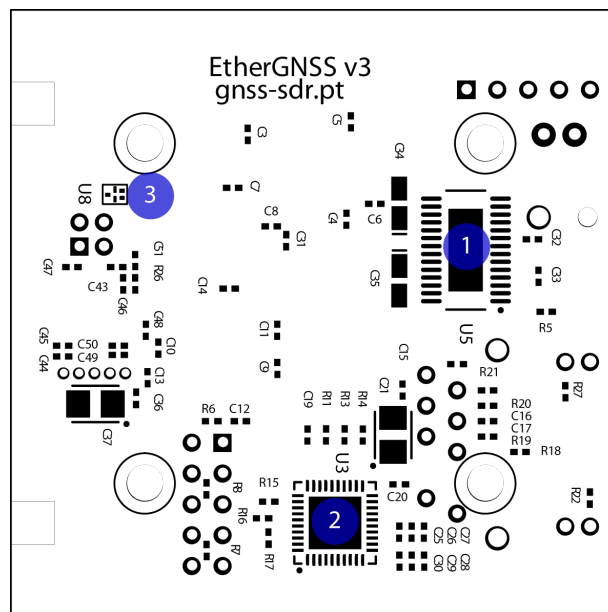


Figure 4.23 – EtherGNSS v3 Layout (Underside).

4.4 Summary

In this section, the hardware circuit design of all blocks was presented as well as the development and prototyping of the EtherGNSS v3. This board was designed with general layout considerations for RF and mixed-signal PCBs with the purpose of minimizing RF interference and coupling between ICs as well as impedance matching. Overall budget and prototyping costs were presented, however it should be taken into account that the values can be reduced in volume manufacturing. Since the EtherGNSS v3 system hardware is already described, the next phase is to describe the user interfaces developed to interact with the system.

5 User Interface

5.1 Introduction

In this section the user interfaces developed for EtherGNSS v3 are described. The microcontroller firmware implements a direct interface with the EtherGNSS v3 system embedded in a webserver which, together with a developed JAVA PC application, gives the user two possible interfaces with the developed system. Other interfaces can be designed or even adapted in current signal processing software in order to improve EtherGNSS v3 functionalities and capabilities.

5.2 Embedded Webserver

The advantage of using the HTTP server in the embedded system is that no special client application is required to communicate with the system, since the interaction with the board is done over any standard web browser.

A webpage was designed to provide an easy method to view and control status information and configurations. The webpage uses a Hypertext Markup Language (HTML) and Asynchronous Javascript and XML (AJAX) technology to interact with the browser (client application) through the TCP/IP protocol. Using asynchronous JavaScript code executing in the web browser, the status information of the page is updated at virtually real time from the web server without doing a full page refresh. To prevent overhead due to the constant use of TCP sockets, the Webserver module is temporarily paused during data acquisition periods. The EtherGNSS v3 Webserver main page can be seen in Figure 5.1.

The main page presents the user with real-time status information of EtherGNSS board. This information includes the board current status and a counter of data packets sent in the last data transmission. For the MAX2769 front-end, an indicator of the active antenna presence is featured, as well as a PLL lock indicator. As for the CPLD, the input mode is presented and can be asserted; also the division ratio resulting by the input mode is presented.

In the body of the page, the user has the possibility to carry out a data transmission to a destination IP, asserting the desired acquisition parameters.

There are two buttons for settings configuration: one for Maxim's MAX2769 RF front-end as can be seen in Figure 5.2, and another for network settings as can be seen in Figure 5.3.

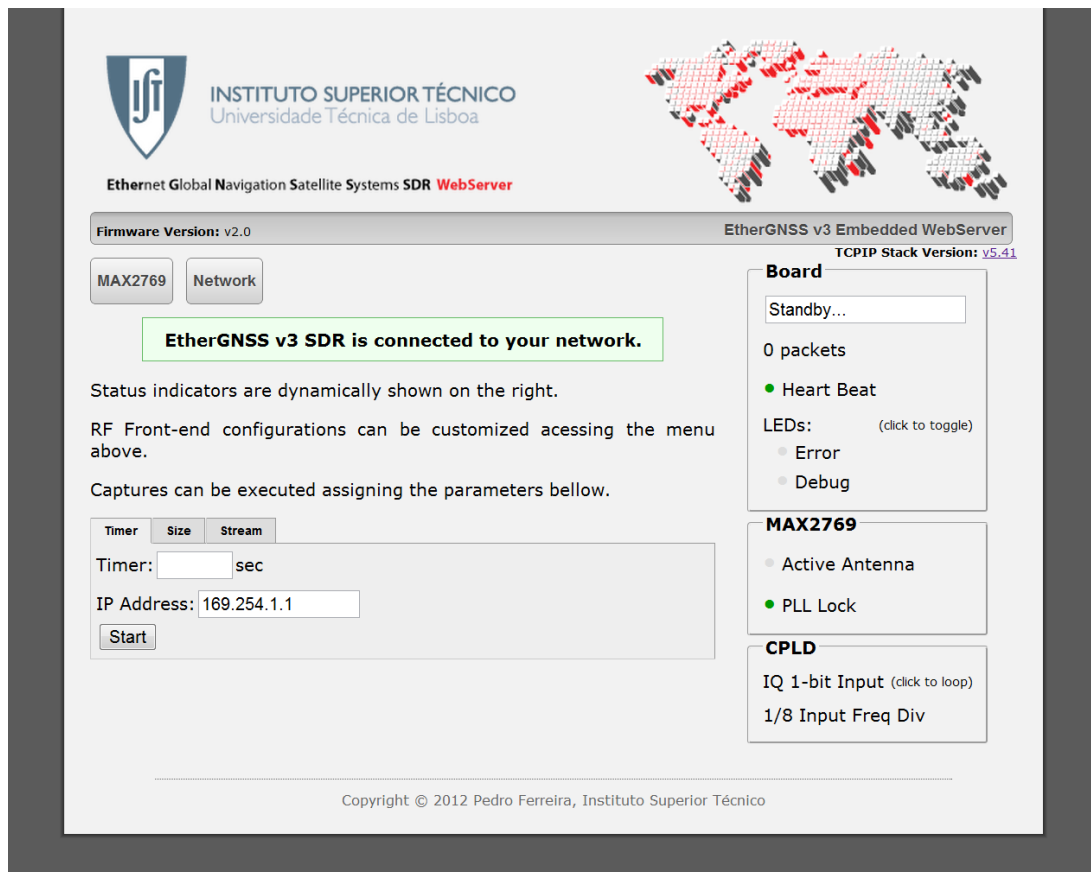


Figure 5.1 – EtherGNSS v3 Embedded Webserver main page.

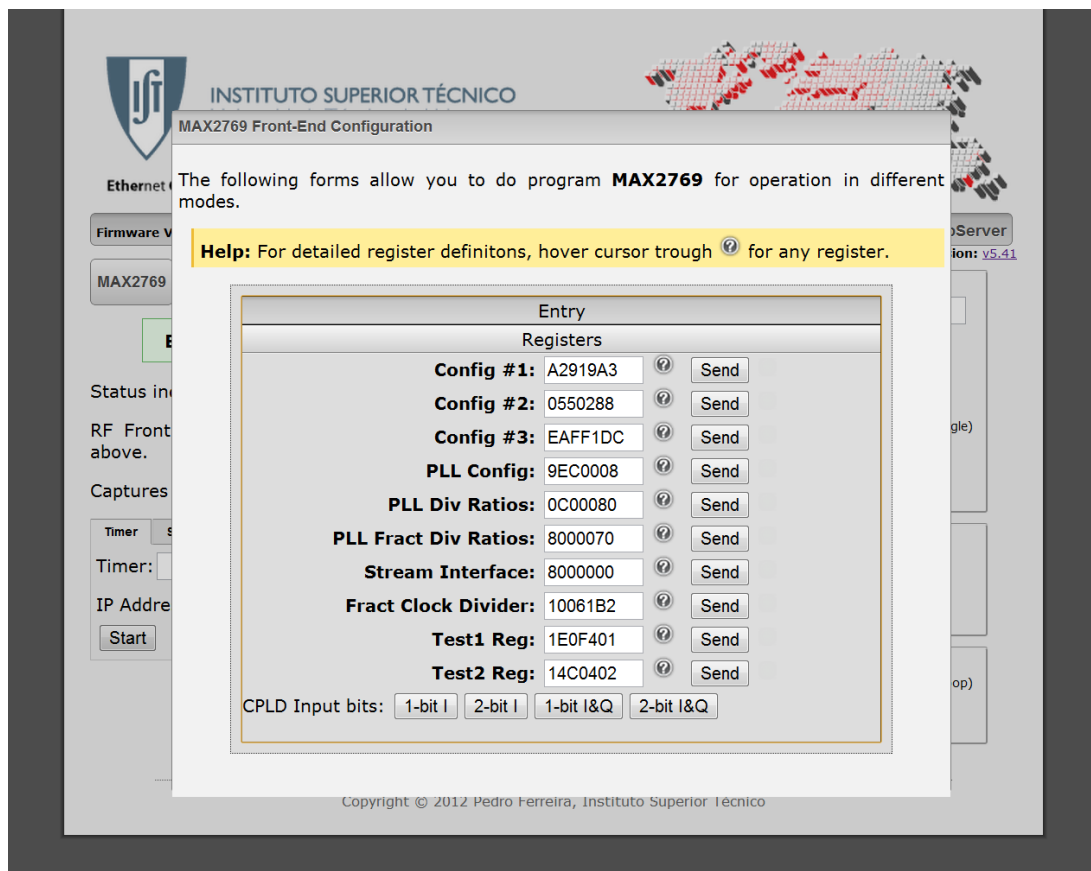


Figure 5.2 – EtherGNSS v3 Embedded Webserver front-end configuration page.

The RF front-end configuration window is composed by various HEX inputs for each register, loaded with their default values. Users can insert different customized settings for each register giving the user the control of the PLL, AGC, ADC, Filter, etc...

The CPLD input mode can also be asserted in this window, to be concordant with the configurations made.

The screenshot shows the 'Network Configuration' page of the EtherGNSS v3 Embedded Webserver. The page has a header with the 'Instituto Superior Técnico' logo and a decorative graphic. A caution box states: 'CAUTION: Incorrect settings may cause the board to lose network connectivity. Recovery options will be provided after save.' Below this, a section titled 'Enter the new settings for the board below:' contains the following fields:

- MAC Address:** 00:04:A3:5F:95:3A
- Host Name:** ETHERGNSS
- ☒ **Enable DHCP**
- IP Address:** 169.254.1.1
- Gateway:** 169.254.1.1
- Subnet Mask:** 255.255.0.0
- Primary DNS:** 169.254.1.1
- Secondary DNS:** 0.0.0.0

A 'Save Config' button is located at the bottom of the settings section. The footer of the page reads: 'Copyright © 2012 Pedro Ferreira, Instituto Superior Técnico'.

Figure 5.3 – EtherGNSS v3 Embedded Webserver network settings page.

The network settings window allows the configuration of EtherGNSS network settings. MAC address and hostname have predefined values that can be modified in this page. As default, the DHCP module is activated for obtaining IP address from the network, although it can be deactivated and a fixed IP address can be assigned.

Changes in any of these settings will cause the EtherGNSS board to reboot for the changes to take effect, although manual reboot will return pre-programmed configurations to take place.

5.3 EtherGNSS v3 GNSS front-end client

Java is an industry-standard, concurrent, class-based, object-oriented language, designed to have as few implementation dependencies as possible. Java cross-platform "write once, run anywhere", denotes that the code that runs on one platform does not need to be recompiled to run on another.

These benefits make Java the most suitable language to develop and application for all users, platform independent.

Thus, a Java application was developed, Figure 5.4 and Figure 5.8, to allow the user to configure and capture raw data from EtherGNSS v3 board. The application requires Java Runtime Environment (JRE) 1.6 or later to be installed on the target computer.

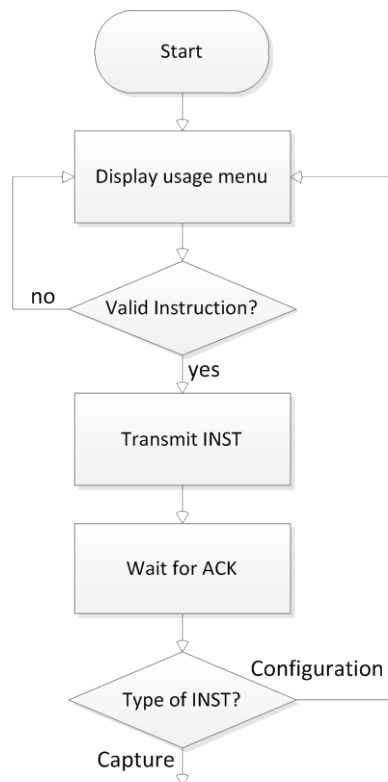


Figure 5.4 – EtherGNSS v3 JAVA GNSS IF Streamer Main application high-level flowchart.

The application starts with the usage menu, Figure 5.5, giving the user the possibility to make front-end configuration changes or initiate a capture. Depending on the instruction transmitted, an acknowledge response is retrieved from the device. In the case of RF configuration modifications, the instruction is directly transmitted to the device and the application returns to the usage menu for new instructions, as can be seen in Figure 5.6.

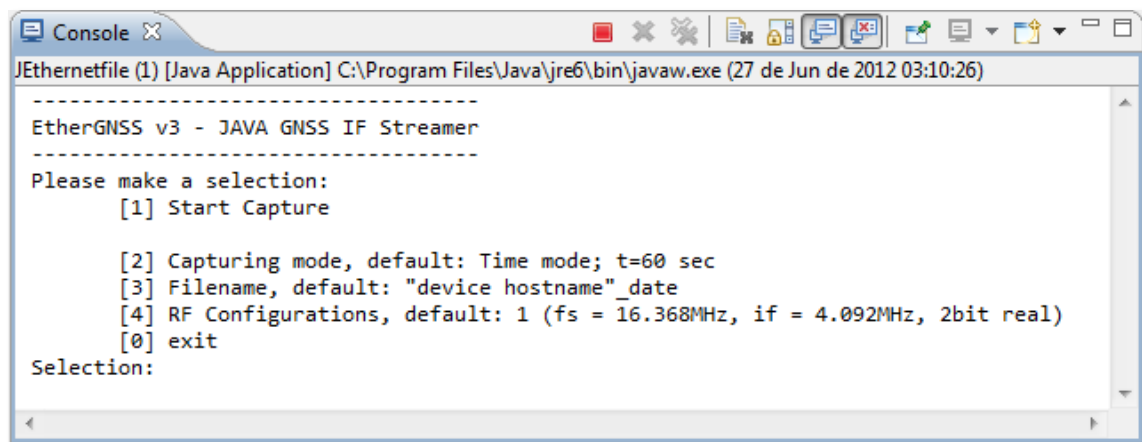


Figure 5.5 – EtherGNSS v3 JAVA GNSS IF Streamer Menu.

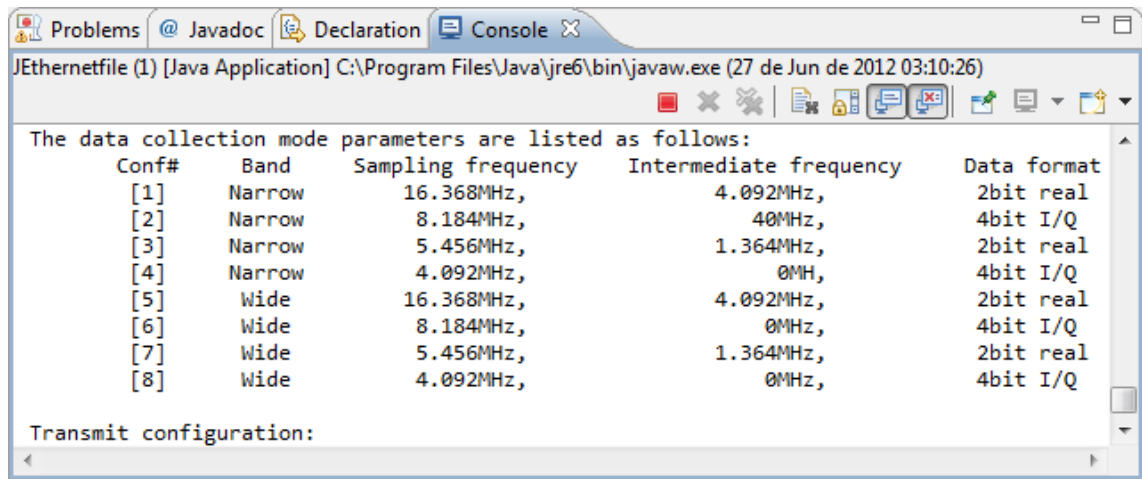


Figure 5.6 – EtherGNSS v3 JAVA GNSS IF Streamer RF configuration.

The RF configuration submenu has already some of the most common RF configurations for data acquisition aiming for an easier user configuration. Both narrow and wide band are included in this configurations.

On the other hand, the application has the ability to customize and initiate capture.

The “capturing mode” submenu, Figure 5.7, presents the user with the three possible acquisition methods, and in each of them the corresponding parameter value can be set.

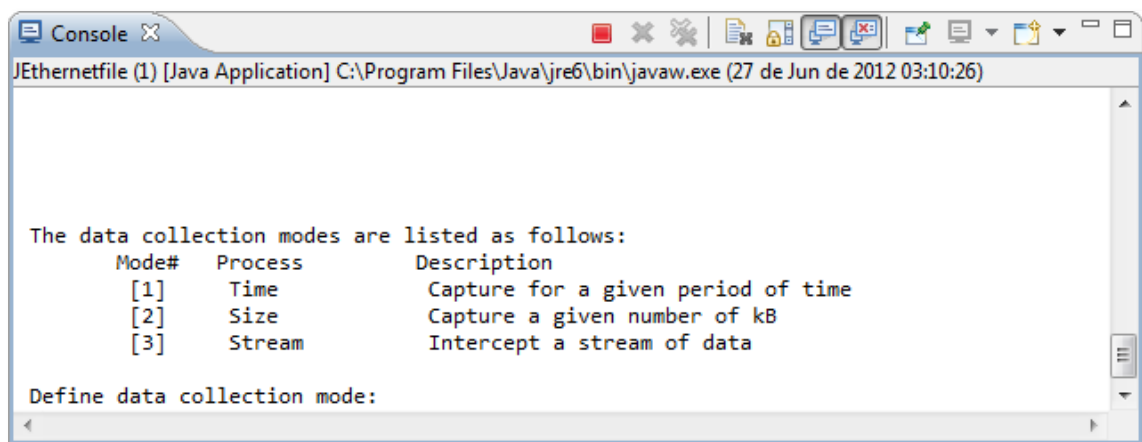


Figure 5.7 – EtherGNSS v3 JAVA GNSS IF Streamer Capture mode submenu.

The acquisition starts with the detection of EtherGNSS v3 board on the network. The application will transmit a broadcast UDP packet containing the message, "Discovery: Who is out there?" on the local network. If any embedded devices with the Announce protocol enabled are connected to the network, they will respond with a UDP packet containing their host name and MAC address. If an Announce packet is received, his IP, Hostname and Mac address is saved and two unicast sockets are opened for instruction and data communication.

Following, a destination file is initially created on the application folder. Then the network UDP packet sniffing starts in the data transmission port.

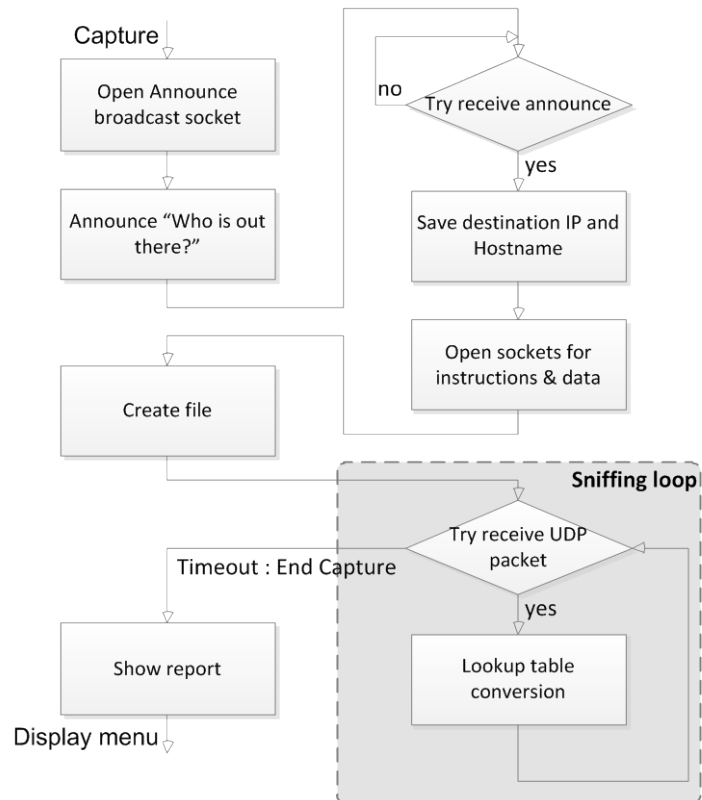


Figure 5.8 – EtherGNSS v3 Java capture procedure flowchart.

The sniffing loop is a permanent packet receiver. Every sniffed packet is indexed by a lookup table containing pre-computed bipolar values, converting every sample of 1 or 2-bits to one byte.

When the capture is ended, a report is shown, as seen in Figure 5.9 and the application returns to the usage menu for new instructions.

```

<terminated> JEthernetfile [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (29 de Jun de 2012 23:19:32)
Timeout: Capture finished!

-----
Report
-----
Time taken: 40999 ms
111222 Packets received
163718784 Bytes received
40009 ms captured

-----
Press any key to return to the menu.
  
```

Figure 5.9 – EtherGNSS v3 JAVA GNSS IF Streamer Report.

5.4 Summary

The system software developed has been explained. Verilog, C, Html, AJAX and Java are the programming languages used in the software development, however a user only needs to use a GUI to operate the *EtherGNSS v3* system hardware and perform GNSS signal captures.

6 Measurements and results

6.1 Introduction

Once the software and hardware design and fabrication of the receiver have been carried out, measurements and validation is required. The validation of some blocks have already been made and presented. This section deals initially with the characterisation of the system with different configurations. Moreover, the performance of the entire system is tested in the field.

6.2 Characterisation of the EtherGNSS v3 system

The test setup used to characterise the complete EtherGNSS v3 system is illustrated in Figure 6.1.

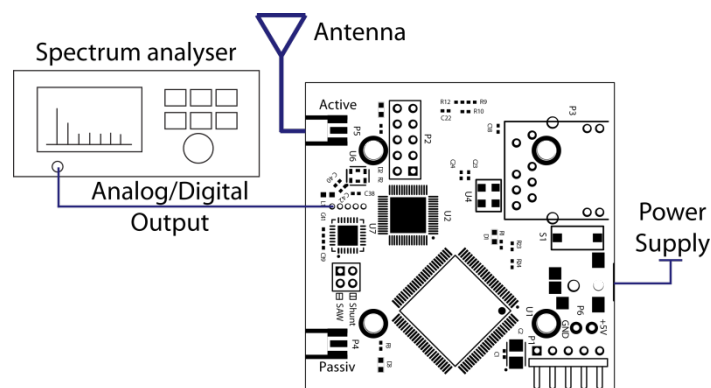


Figure 6.1 – Test setup used for characterization of the EtherGNSS v3 system.

The registers were set to the default power-up states. LNA1 input was driven from a 50 Ω active antenna. RF measurements were done in both analog output mode with the ADC bypassed and also in digital mode, by observing the ADC most significant bit. The PGA was set to Automatic Gain Control (AGC).

To validate future measures, the TCXO output was analysed, as seen in Figure 6.2, proving its precision and stability with a measured value of 16.367925 MHz.

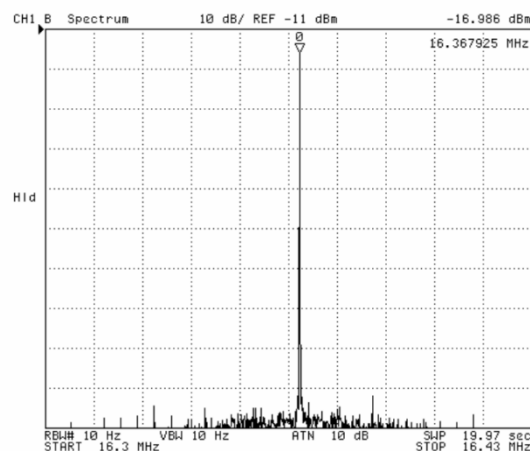


Figure 6.2 – Output signal spectrum of the 16.368 MHz TCXO.

For the following cases, some previously described circuits should be recalled, particularly the frequency synthesizer, represented in Figure 4.3.

Case 1:

For the usage of the system on the L1 frequency band to acquire the GPS L1 Narrowband constellation signal, the following configurations were made.

Considering the TCXO frequency of 16.368 MHz as the reference frequency; reference divider, RDIV, was set to 16 imposing a PLL comparing frequency of 1.023 MHz.

$$f_{Comp} = \frac{f_{TCXO}}{RDIV} = \frac{16.368}{16} = 1.023 \text{ MHz}$$

The IF filter center frequency was set to 4.092 MHz with a narrow bandwidth of 2.5 MHz.

For a IF frequency of 4.092 MHz at the quadrature mixer output, a VCO frequency of 1,571.328 MHz was obtained setting the feedback loop divider, NDIV, to 1536.

$$f_{VCO} = f_{GPS_{L1}} - f_{IF} = 1575.42 - 4.092 = 1571.328 \text{ MHz}$$

$$VCO_{NDIV} = \frac{f_{VCO}}{f_{Comp}} = \frac{1571.328}{1.023} = 1536$$

Figure 6.3 shows the spectrum at the mixer output analog signal. A clearly defined narrow pass-band filter spectrum can be seen, which validates the configuration settings above.

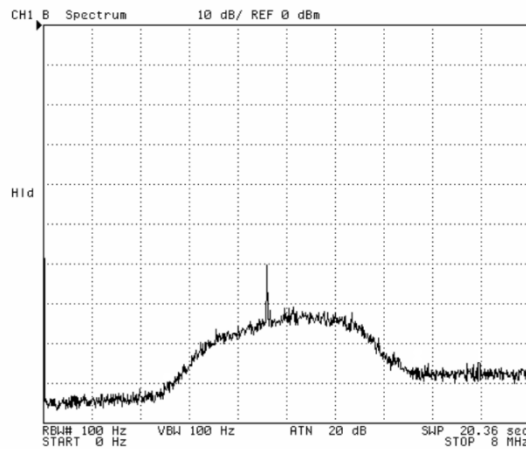


Figure 6.3 – Case 1 – GPSL1 narrowband analog output signal spectrum.

Figure 6.4 shows the digital spectrum of the output digital signal most significant bit. Four tones can be clearly identified: the first one, at 4.092 MHz, corresponds to the IF navigation data frequency; the third tone is the sampling frequency at 16.368 MHz; the other two tones correspond to the sampling components $16.368 \pm 4.092 \text{ MHz}$.

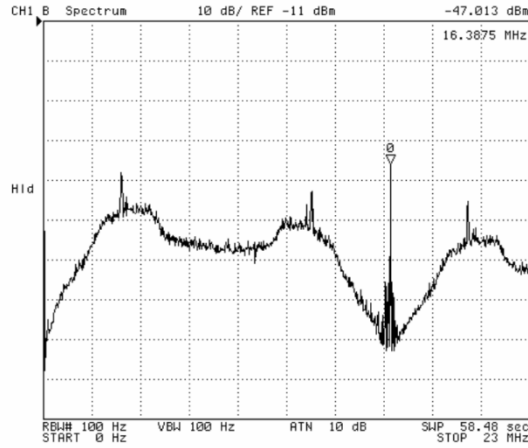


Figure 6.4 – Case 1 – GPSL1 narrowband digital output signal spectrum.

Case 2:

For the usage of the system on the L1 frequency band to acquire the GPS L1 and Galileo L1 constellations signals, the following configurations were made.

Also, considering the TCXO frequency of 16.368 MHz as the reference frequency; reference divider, RDIV, was set to 16 imposing a PLL comparing frequency of 1.023 MHz.

$$f_{Comp} = 1.023 \text{ MHz}$$

The IF filter center frequency was set to 2 MHz with a bandwidth of 4.2 MHz.

For a IF frequency of 2.046 MHz at the quadrature mixer output, a VCO frequency of 1,573.374 MHz was obtained setting the feedback loop divider, NDIV, to 1538.

$$f_{VCO} = f_{GPS/Galileo_{L1}} - f_{IF} = 1575.42 - 2.046 = 1573.374 \text{ MHz}$$

$$VCO_{NDIV} = \frac{f_{VCO}}{f_{Comp}} = \frac{1573.374}{1.023} = 1538$$

Figure 6.5 shows the spectrum at the mixer output analog signal. In this spectrum a 4.2 MHz pass-band filter can be seen, for the IF of 2.046 MHz. Once again configuration settings are validated.

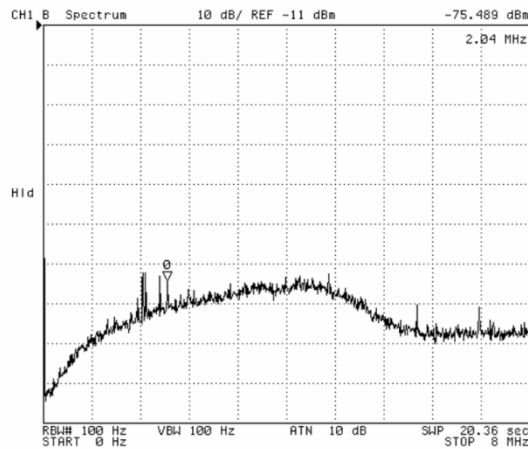


Figure 6.5 – Case 2 – GPS & Galileo L1 analog output signal spectrum.

Figure 6.6 shows the digital spectrum of the output digital signal for the GPS and Galileo L1 frequencies. Again, the spectrum has four tones: the first one, at 2.046 MHz , corresponds to the IF navigation data frequency; the third tone is the sampling frequency at 16.368 MHz ; the other two tones correspond to the sampling components $16.368 \pm 2.046\text{ MHz}$.

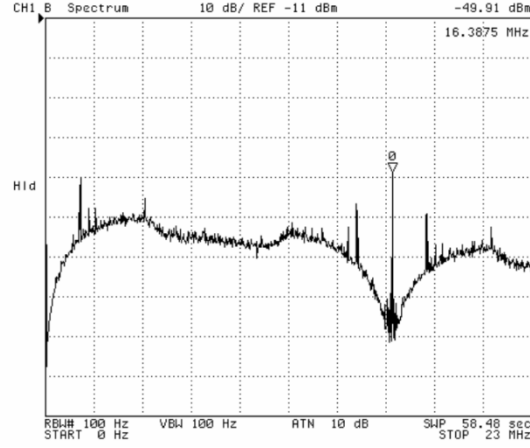


Figure 6.6 – Case 2 – GPS & Galileo L1 digital output signal spectrum.

Case 3:

For the usage of the system on the L1 frequency band to acquire the GLONASS FDMA Wideband constellations signals, the following configurations were made.

As GLONASS uses FDMA each satellite uses its own frequency and hence has its own IF frequency.

$$f_{Comp} = 1.023\text{ MHz}$$

A zero-IF configuration is used in which the IF filter is set to low-pass with a two-sided bandwidth of 18 MHz and a corner frequency of 9 MHz . For zero-IF, a VCO frequency of 1602 MHz , GLONASS channel 0, obtained setting the feedback loop divider, NDIV, to 1565.

$$f_{VCO} = f_{GLONASS} - f_{IF} = 1602.00 - 0 = 1602.00\text{ MHz}$$

$$VCO_{NDIV} = \frac{f_{VCO}}{f_{Comp}} = \frac{1602.00}{1.023} = 1565$$

Figure 6.7 shows the spectrum of the output analog signal.

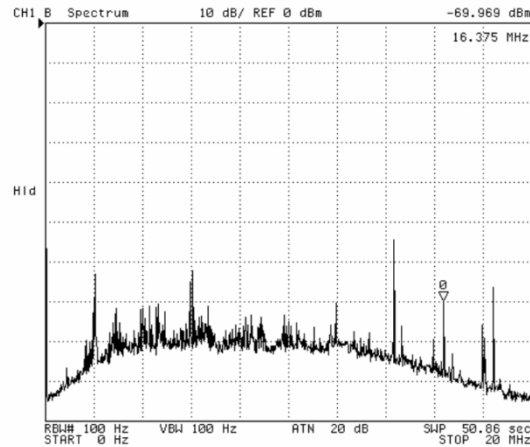


Figure 6.7 – Case 3 – GLONASS wideband analog output signal spectrum.

Figure 6.8 shows the digital spectrum of the output digital signal for the GLONASS frequency band. In this case, distinct from the previous, the digital spectrum results on low-pass configuration up to sampling frequency spur.

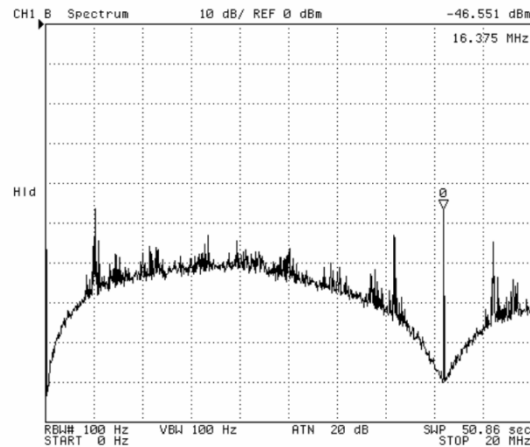


Figure 6.8 – Case 3 – GLONASS wideband digital output signal spectrum.

6.3 Complete EtherGNSS v3 acquisitions validation

For the validation of the system, it was required to go to the field to test the whole system with real GNSS signals. In order to attain this goal, a one set of signal acquisitions where performed in a geodetic marker. Thus, it is also possible to derive the deviation of the navigation solution comparing to a well know coordinates.

The geodetic marker chosen was the geodetic marker COTÃO in Porto Salvo, Sintra, Portugal. The respective coordinates are the reference for comparison with EtherGNSS system.

Table 6.1 – Identification and coordinates of reference.

Geodetic marker	European Terrestrial Reference System 1989 (ETRS89) Coordinates	Height
COTÃO	-9° 18' 10.6789" W	223.95 m

Initial signal acquisition was made with configuration settings of narrowband Case 1 in a 2-bit only I channel mode.

Figure 6.9 shows the characteristics of the acquired signal. In the frequency domain plot it is clearly visible the shape of the IF filter in a well-defined signal. On the left it is visible the quantization of the acquired signal, followed by the histogram of the signal. This histogram, as it is expected, bears a strong resemblance to the probability function for a Gaussian random variable. This is caused by the random nature of the signal being sampled.

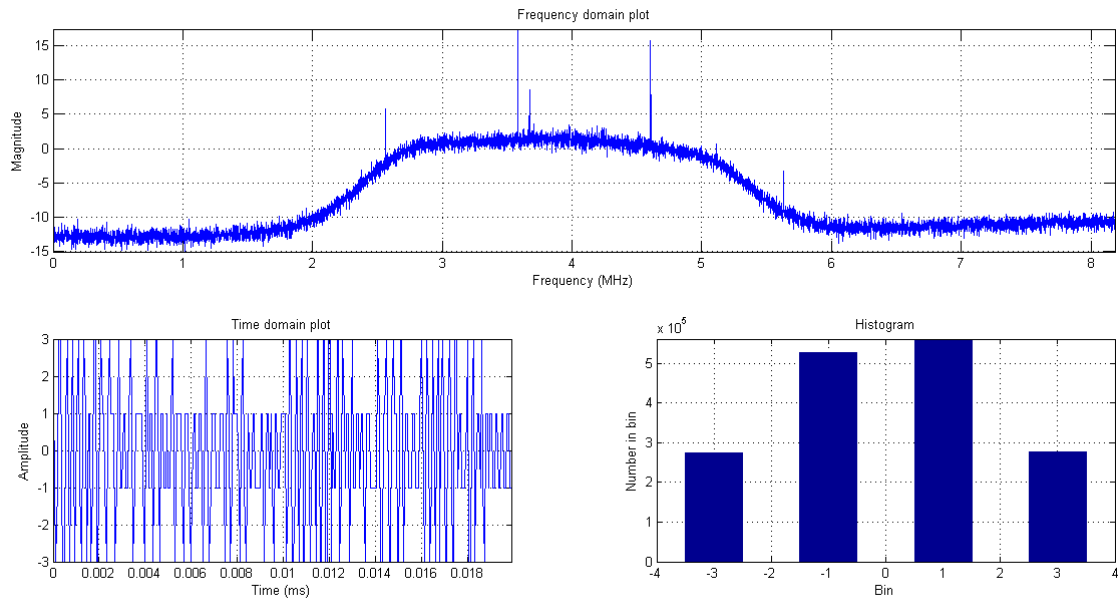


Figure 6.9 – Probed signal acquisition 1. $f_s = 16.368$ MHz, $IF = 4.092$ MHz, $Filter_{BW} = 2.5$ MHz, $f_{VCO} = 1571.328$ MHz

Figure 6.10 shows the acquisition result for the acquired signal. A signal is considered acquired when the correlation value exceeds 2.5. In this acquisition a total of 5 satellites were acquired.

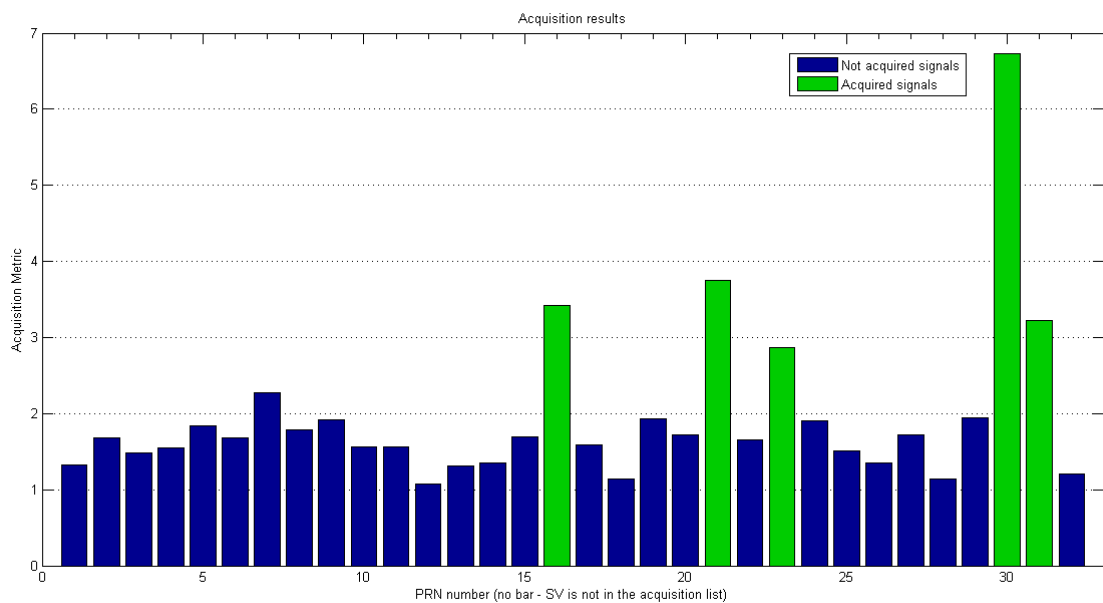


Figure 6.10 – Acquisition result for acquisition 1.

In Figure 6.11, the result from the tracking of one of the five channels is plotted. In this figure, the I and Q component of the baseband signal are plotted. The quality of the tracking is revealed by the

concentration groups formed that needs to be symmetric. Navigation data was successfully extracted as can be seen in its representation with clear transitions of bits. Bellow, the plot for the Early, Prompt and Late code correlation value is showed. It is intended to always have the Prompt value higher than the Early and Late, and that is the case as can be seen.

The plot of the raw PLL discriminator and raw DLL discriminator are show in red. They represent the error in the NCOs, and their filtered values are represented by the plots in blue.

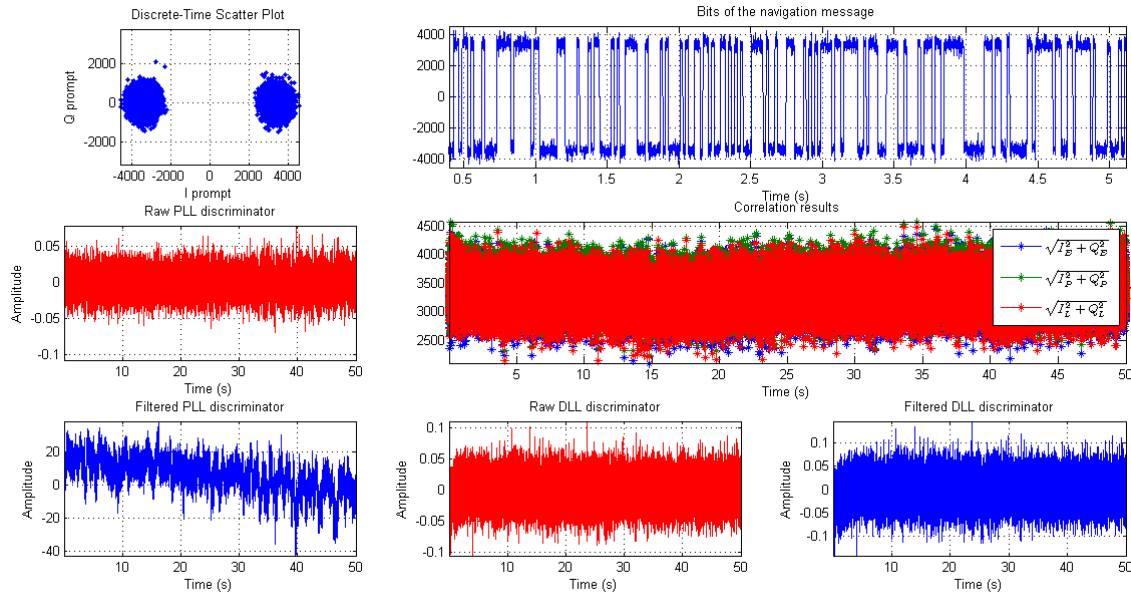


Figure 6.11 – Tracking result for satellite PRN30 for acquisition 1.

The results of the navigation solution are presented in Figure 6.12. The navigation solution shows the coordinates variations. The coordinates variation is self-explanatory representing the variation, in the Universal Transverse Mercator (UTM) coordinate system, of the easting, northing and up coordinate. Lower variation means better tracking. Figure 6.13 enables a better understanding of the variation being of around ± 2 m for easting and northing coordinates, demonstrating a good acquisition. The up variation shows an increased variation, this is due to the location of the satellites above which leads to a decrease of information in the up dimension. This effect is described as the dilution of precision.

A two dimensions plot indicates the calculated positions for the processed signal, with a variation of about 3 meters. The red cross indicates the mean position which is considered the result position.

The final plot in the right shows the relative position of the acquired satellites in the sky. A value of Positional Dilution of Precision (PDOP) of 7.0128 was obtained which measures the distribution of the acquired satellites in the sky. The precision of the navigation solutions can be degraded by a non-even distribution of the satellites. A smaller PDOP value indicates a better distribution, and the value in this acquisition is fair.

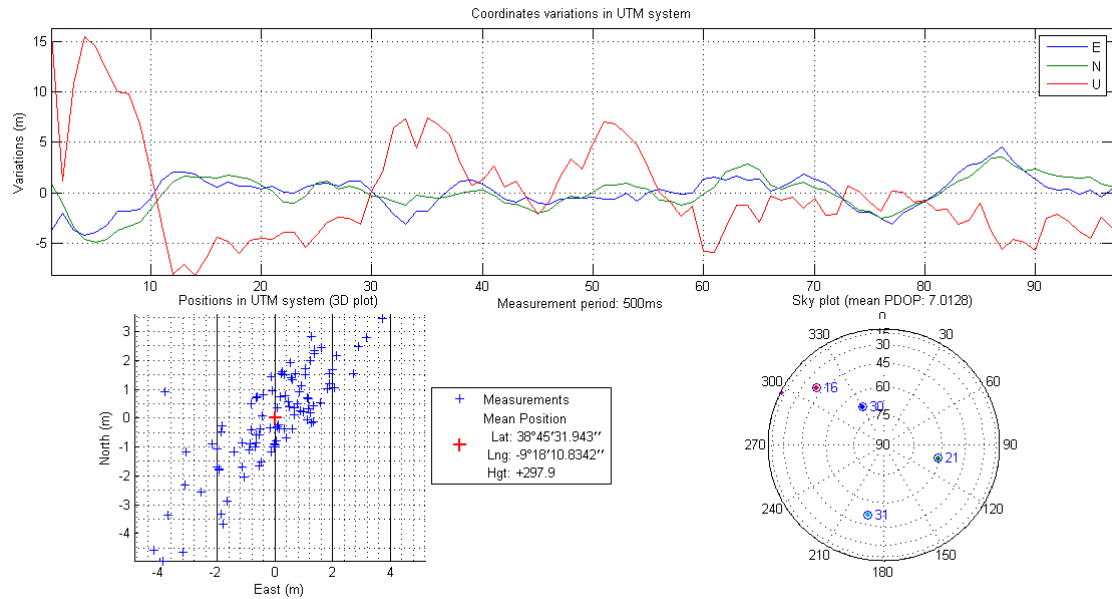


Figure 6.12 – Navigation solution for acquisition 1.

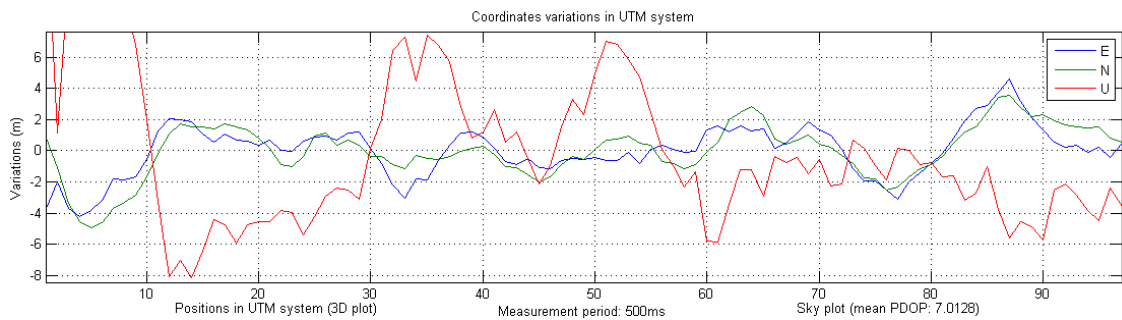


Figure 6.13 – Zoomed coordinate variation for acquisition 1.

The resulting position is given in ETRS89 Coordinates with values bellow.

Table 6.2 – Acquisition 1 solution coordinates.

	Latitude	Longitude	Height
	38° 45' 31.943" N	-9° 18' 10.8342" W	297.9 m
Deviation	+0.44 m	+4.80 m	+73.95 m

As it was expected, the height coordinate deviation from the reference is high. This could be explained by the value of PDOP. North coordinate shows a very accurate position value, with a deviation of +0.44 m, although West coordinate demonstrate a diversion of +4.80 m from the know reference coordinate. This deviation could also be explained by the satellites relative position in the sky with the respect to the receiver.

A second acquisition test was made with configuration setting of narrowband Case 1 now with a 1-bit I and Q channel mode. IQ sampling holds twice the spectral information than real sampling, thus a more accurate acquisition is expected.

At the time of the acquisition the visible GPS constellation was as represented in the Figure 6.14 sky plot.

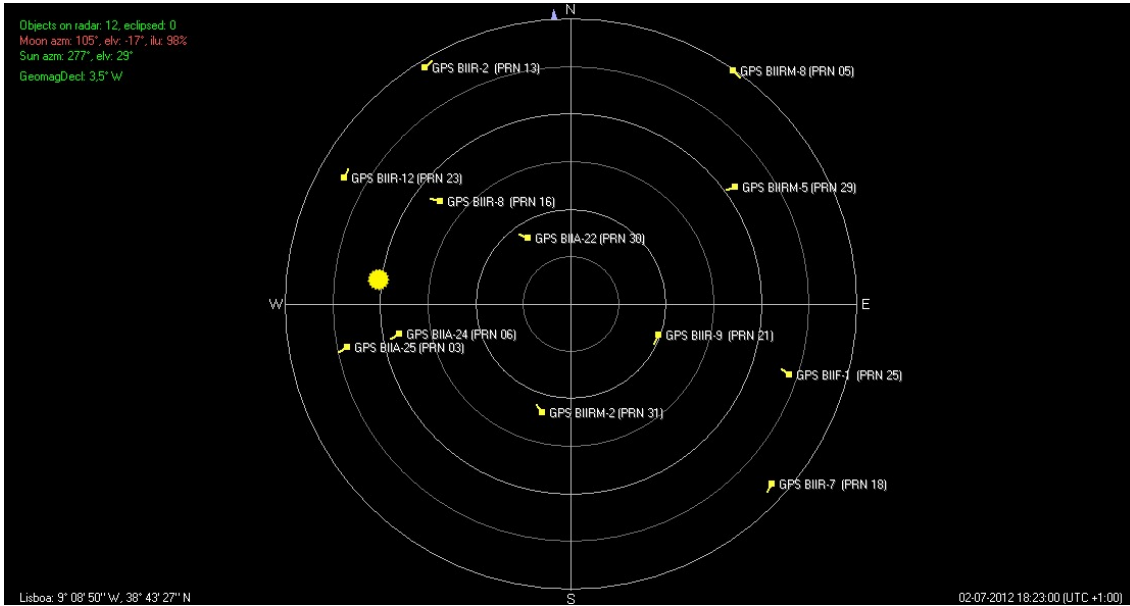


Figure 6.14 – Orbitron [22] simulated Sky plot at acquisition 2.

Figure 6.15 shows the characteristics of the new acquired signal. Now the signal has both positive and negative frequency content. Since sampling is done with 1-bit sampling, the histogram has two values -1 and +1, both with approximately the same distribution.

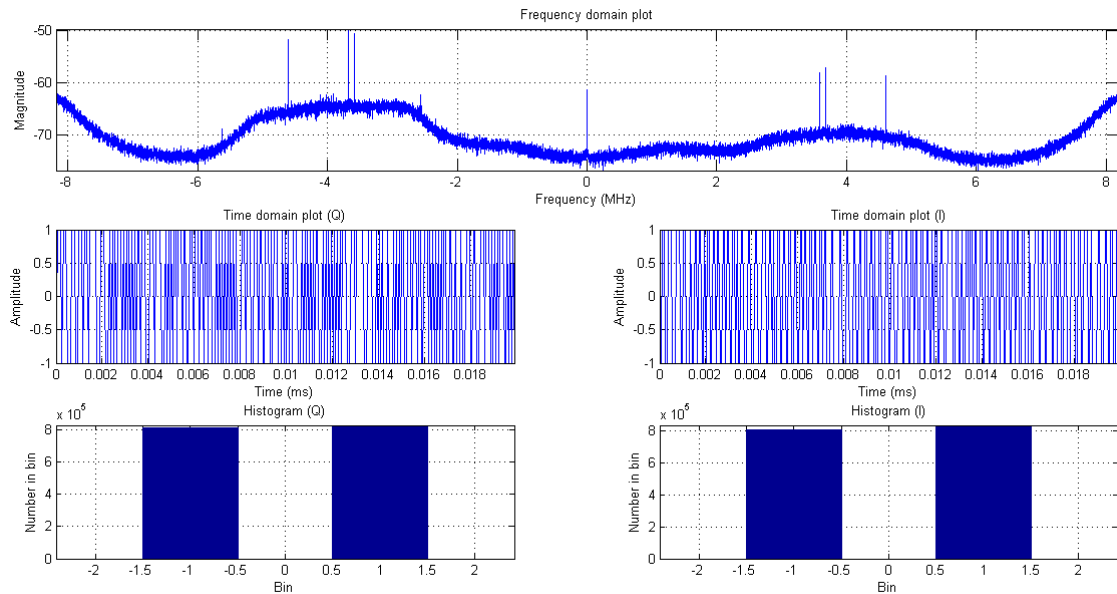


Figure 6.15 – Probed signal acquisition 2. $f_s = 16.368$ MHz, $IF = 4.092$ MHz, $Filter_{BW} = 2.5$ MHz, $f_{VCO} = 1571.328$ MHz.

Figure 6.16 shows the acquisition result with 5 satellites acquired. The 5 acquired satellites according to Figure 6.14 sky plot, are approximately the most centered satellites. Satellite PRN21 was not acquired although his correlation result was near the 2.5 threshold.

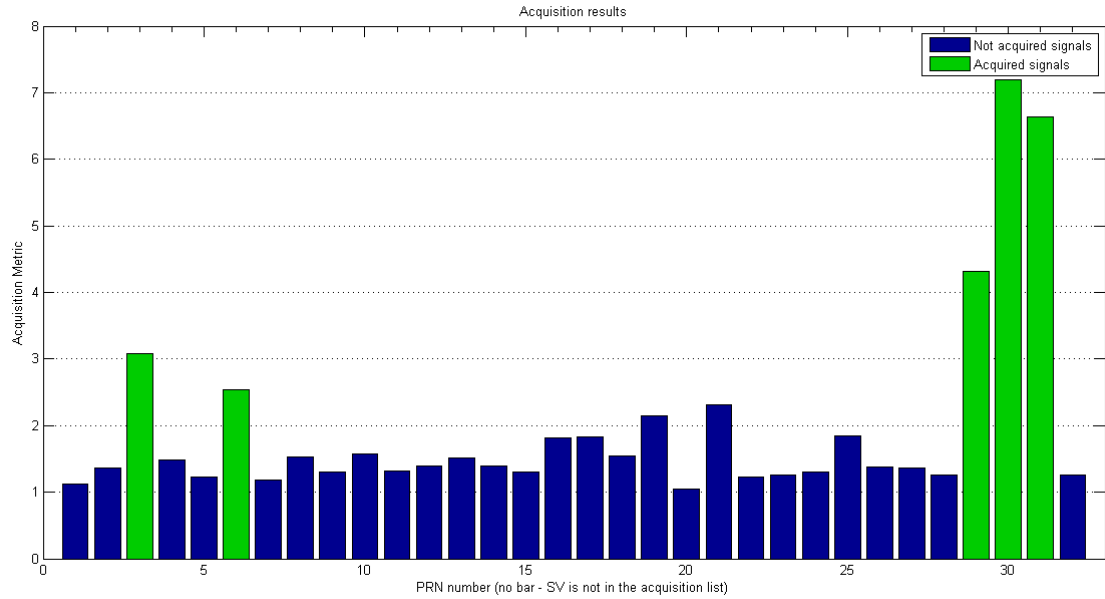


Figure 6.16 – Acquisition result for acquisition 2.

Figure 6.17 illustrates the tracking result plot from the PRN30 satellite. I and Q baseband components demonstrates good tracking quality.

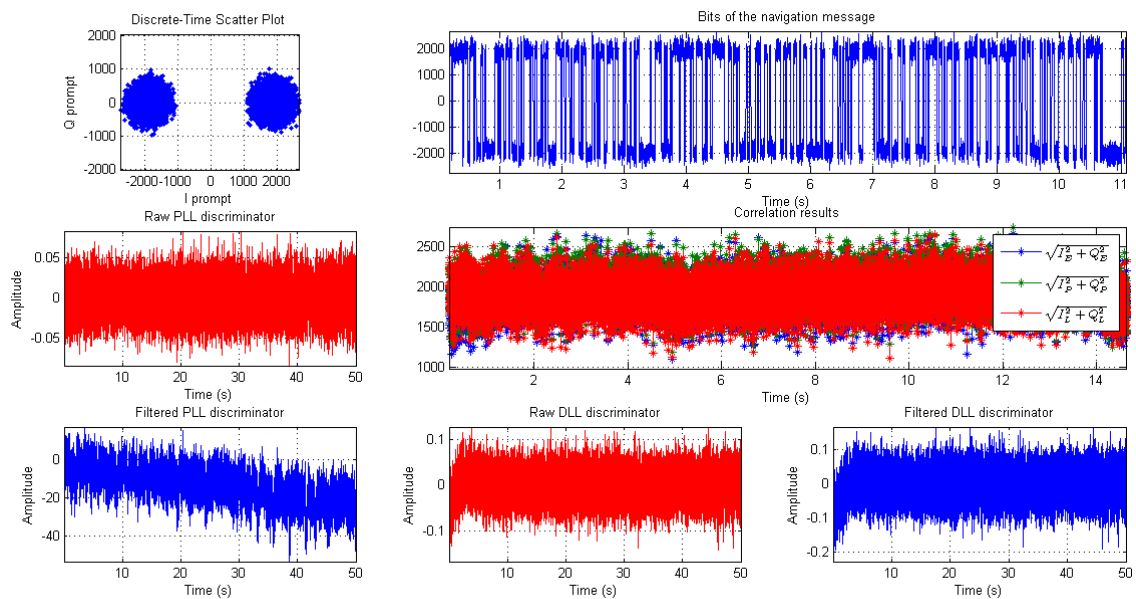


Figure 6.17 – Tracking result for satellite PRN30 for acquisition 2.

Navigation solution coordinates variation for easting and northing coordinates are below ± 2 m revealing an improvement, even for the up coordinate. The variation improvement can also be explained by the PDOP of 2.7479. The extracted sky plot is in concordance with the simulated one.

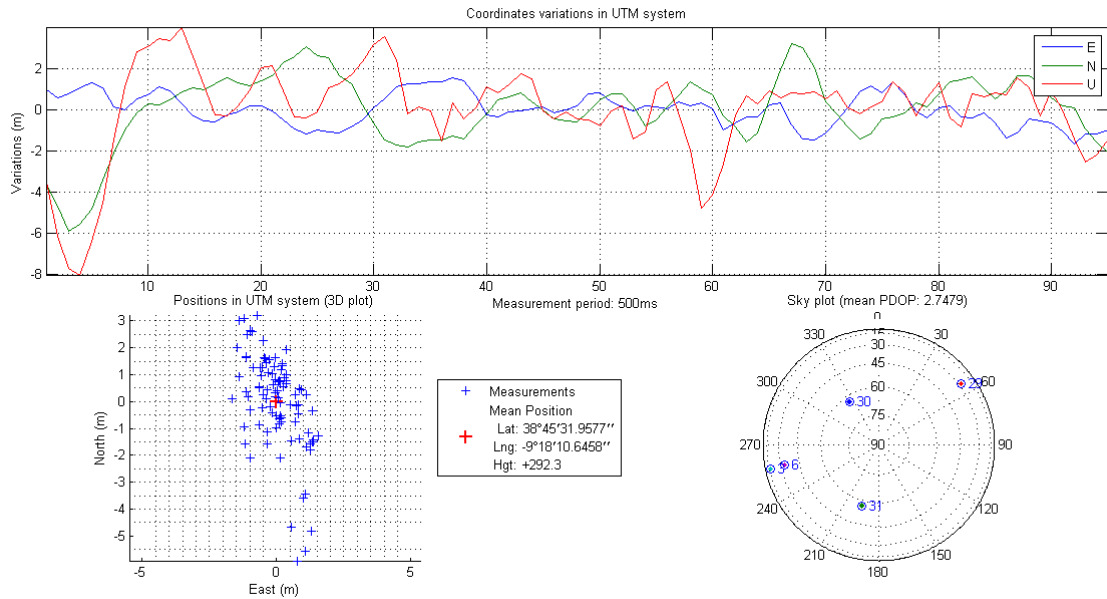


Figure 6.18 – Navigation solution for acquisition 2.

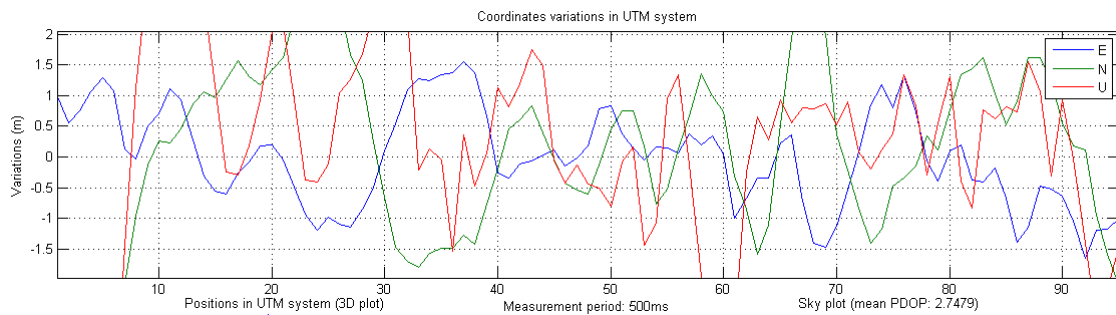


Figure 6.19 – Zoomed coordinate variation for acquisition 2.

Comparing coordinates a real improvement exists. The North coordinate demonstrates a deviation of +0.89 m, whereas the West coordinate indicates a deviation of -1.02 m. The Up coordinate has a higher deviation of +68.35 m.

Table 6.3 – Acquisition 2 solution coordinates.

	Latitude	Longitude	Height
	38° 45' 31.9577" N	-9° 18' 10.6458" W	292.3 m
Deviation	+0.89 m	-1.02 m	+68.35 m

Since Galileo current on-sky satellites have two spreading codes, both codes are truncated gold codes, no real acquisition could be made for Galileo navigation system. Nevertheless an acquisition test was made to prove the system compatibility with Galileo.

Figure 6.20 illustrates the acquisition characteristics of the acquired signal. Acquisition was made with configuration setting of Case 2 in 1-bit I and Q channel mode.

The 4.2 MHz pass-band IF filter is well defined with an IF frequency of 2 MHz. Acquisition of mixed GPS+Galileo can be executed by EtherGNSS v3 system.

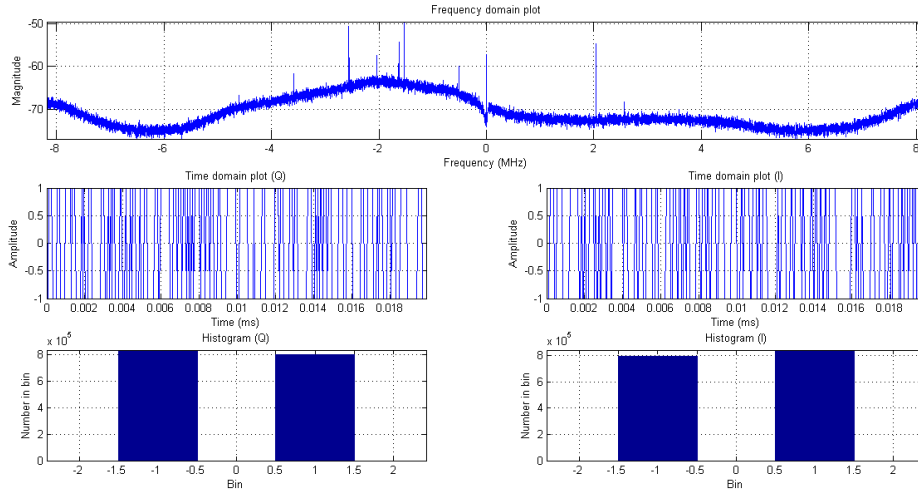


Figure 6.20 – Probed signal acquisition 3. $f_s = 16.368$ MHz, $IF = 2.046$ MHz, $Filter_{BW} = 4.2$ MHz, $f_{VCO} = 1573.374$ MHz.

Finally a GLONASS acquisition was tested. Once again, since GLONASS uses FDMA, each satellite uses its own frequency and hence has its own IF frequency. Thus an acquisition test was made with configuration setting of wideband Case 3 in 1-bit I and Q channel mode.

Figure 6.21 presents the recorded signal spectrum. Zero frequency corresponds to GLONASS channel 0 at 1602 MHz.

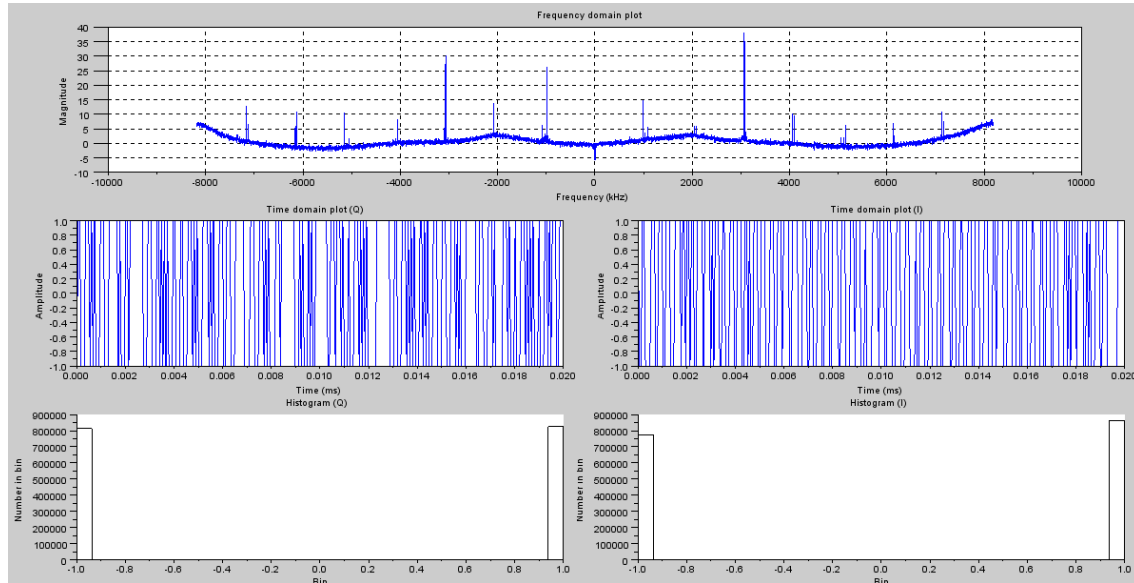


Figure 6.21 – Probed signal acquisition 4. $f_s = 16.368$ MHz, $IF = 0$ MHz, $Filter_{BW} = 18.8$ MHz, $f_{VCO} = 1602$ MHz.

This signal acquisition did not return any acquired satellite. This result is probably caused either by the use of a GPS active antenna with insufficient bandwidth to accommodate the GLONASS signal spectrum or a possible inadequate programming of the front-end frequencies. Unfortunately, the MAX2769 datasheet lacks much of the information required to adequately program it for GLONASS reception. The chipset manufacturer (MAXIM) has been contacted about this but no response has yet been received. More testing needs to be made in order to adequate the configurations on the front-end

to guarantee GLONASS acquisition. Nevertheless the characterisation of the system for GLONASS navigation system proves the ability to acquire in the Galileo L1 band.

6.4 Overall EtherGNSS v3 Features and Specifications

The overall performance of the receiver was obtained. These features are summarized in Table 6.4.

Table 6.4 – EtherGNSS v3 Specifications

Power supply	+3.3 V to +10 V
Current consumption	205 mA
Antenna Connectors	2x MMCX
Support of Active Antenna	Yes, maximum 50 mA @ 4.5 VDC
RF Frequency Range	1530 < 1575.42 < 1670 MHz
Bandwidth	2.5 / 4.2 / 8 MHz / 18 MHz (low-pass)
Intermediate Frequency (IF)	4.092 MHz (Configurable)
Analog IF Output	Available
Internal TCXO	Yes, @ 16.368 MHz
Sample Rate	4.092 / 8.184 / 16.368 MHz
Sampling Resolution	Up to 4 bit
Clock Output	Available
Digital Data Interface	Ethernet 10/100 Base-T
Digital sampling modes	Per Time / Per Size / Stream

6.5 Summary

In this section, the fabricated EtherGNSS v3 receiver has been characterized and validation was performed. First, with a set of different configurations from narrow band (2.5 MHz) to wide band (18 MHz), the characterization of system was presented. Then, a method for the complete system validation in real environment was carried out and the obtained results presented. The validation results are considered very positive, although they could not be obtained for all the configurations. However, this is no drawback since the system global working operation was successfully verified and characterized. Finally a summary of the overall EtherGNSS v3 system features and specifications was presented.

7 Conclusions and Future Work

During the elaboration of this work, many obstacles were overtaken leading to a fully functional, tested and validated GNSS signal acquisition system. This work was divided in the study, design, manufacture, test/validate of a GNSS receiver. To accomplish the proposed objectives, it was imperative to acquire the necessary knowledge in the areas of network, programming, embedded systems, programmable logic and RF, broadening the scope of the dissertation. Some conclusions could be taken based on the work developed and the results achieved. GNSS SDRs solutions on the market are rather similar, each taking different hardware or software approaches, although none providing the user with full customization and ease to use. It was necessary to develop a fully customizable and user friendly GNSS receiver. Some designing and implementation considerations were made based on related works, concerning better performance.

The MAX2769 is a low cost, versatile, high performance, small size IC. The MAX2769 high integration chip, provides a convenient design of the software GNSS receiver. The characteristics of the adjustable parameters can provide a wide range of outputs. The interface chosen for transmission of the data was Ethernet. The implementation of the Ethernet protocol on an embedded system can be challenging, although having a networking topology standard for most computers along with the integration of a user interface in the embedded system, makes it an added value and a great benefit to the user.

Overall, the elaboration of EtherGNSS v3 was a success, proving the concept and obtaining very good results. In fact the developed EtherGNSS is at this time the only GNSS IF sampler in the world capable of acquiring GPS, Galileo and GLONASS signals and offering full system programmability through a standard Ethernet interface. Other commercially available samplers are customized for only one of the systems and do not allow any kind of programmability (like filter bandwidths and center frequencies, IF frequency or sampling rate). In addition, all these systems are USB-based and do not offer the flexibility of the Ethernet interface. One unit of the EtherGNSS is planned to be installed at a known location at IST Taguspark campus and permanently connected to the internet via Ethernet. In this way it will be possible for users to perform custom signal acquisitions (either GPS/Galileo or GLONASS) any time, from anywhere in the world.

It should also be noted that the future of the modernization programs of already implemented navigation systems is to increase use of the civilian band and decrease use of more specific frequency bands, thus making the developed EtherGNSS system very attractive as a multisystem general IF sampler. For example, the GLONASS-K will also transmit on the 1575.42 MHz L1 band using CDMA (see Figure 2.8). Further developments of the EtherGNSS system are possible, namely:

- Improve EtherGNSS webserver for additional configurations and interaction with the user.
- Develop a GUI for Java application enabling a more user friendly interaction.

- Improve radio frequency interference (RFI) on EtherGNSS system in different ways like modifying the PCB substrate, gold coating, EMI/RFI shielding, among others, aiming for a better RF immunity.
- Enable real-time processing software to work with the EtherGNSS system. This can be rather challenging since the current software solutions that accept data streams, do it through serial USB, and EtherGNSS, as well as any other Ethernet based streamer do it by the transmission of packets of data.
- Step from a CPLD+MCU towards only FPGA implementing a SoC where, besides signal sampling, some processing can be made, like acquisition correlation, determination of Doppler shifts and delay on each channel. Some information is already available on the internet [23] [24].

There are many other improvements as well as new developments that can be made in the radio navigation satellite area which, at the same time is evolving to new horizons. The creation of a study and development group on this topic would be of interest.

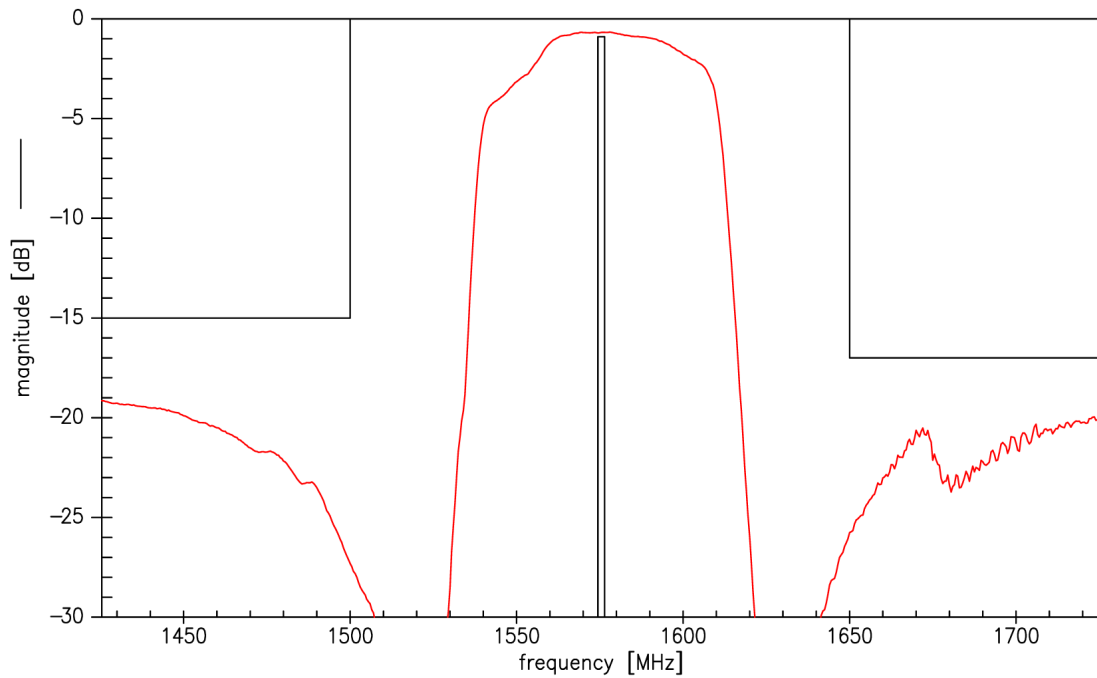
8 References

- [1] European Space Agency. (2012, June) NAVIPEDIA. [Online]. <http://www.navipedia.net/>
- [2] Dennis M. Akos, Nicolaj Bertelsen Kai Borre, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*, 1st ed.: Birkhäuser Boston, 2006.
- [3] Jean-Marie Zogg, *GPS - Essentials of Satellite Navigation*, Revision C ed. Thalwil, Switzerland: u-blox AG, February 2009.
- [4] GLOBAL POSITIONING SYSTEM DIRECTORATE, "NAVSTAR GPS Space Segment/Navigation User Segment Interfaces," IS-GPS-200F, 2011.
- [5] Dennis M. Akos, "A software radio approach to Global Navigation Satellite System receiver design," *Doctoral Dissertation*, August 1997.
- [6] Christopher Hegarty Elliott D. Kaplan, Ed., *Understanding GPS: Principles and Applications*, 2nd ed.: Artech House Publishers, 2005.
- [7] "European GNSS (Galileo) Open Service Signal In Space Interface Control," European Union, OS SIS ICD, Issue 1.1, 2010.
- [8] "GLONASS INTERFACE CONTROL DOCUMENT, Navigational radiosignal In bands L1, L2 (Edition 5.1)," Russian Institute of Space Device Engineering, MOSCOW, 2008.
- [9] (2012, March) International Committee on Global Navigation Satellite Systems (ICG). [Online]. <http://www.unoosa.org/oosa/SAP/gnss/icg.html>
- [10] Prof. Dr.-Ing. Günter Hein, "GNSS Interoperability - Achieving a Global System of Systems or "Does Everything Have to Be the Same?,"" *Inside GNSS magazine*, pp. 57-60, January/February 2006.
- [11] G. Girau, A. Tomatis, F. Dosis, and P. Mulassano, "Efficient Software Defined Radio Implementations of GNSS Receivers," *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pp. 1733-1736, May 2007.
- [12] Jin Tian, Wang Ye, Shi Lin, and Zhang Hua, "Software Defined Radio GNSS Receiver Design over Single DSP Platform," *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pp. 1-4, October 2008.
- [13] C. Stöber et al., "ipexSR: A real-time multi-frequency software GNSS receiver," *ELMAR, 2010 PROCEEDINGS*, pp. 407-416, September 2010.
- [14] Danish GPS Center. (2009, April) SoftGPS project. [Online]. <http://kom.aau.dk/project/softgps/>
- [15] Juan Melendez Lagunilla, Roc Berenguer Perez Jaizki Mendizabal Samper, *GPS and Galileo: Dual RF Front-end Receiver and Design, Fabrication, & Test*, 1st ed.: McGraw-Hill Professional, 2008.

- [16] Maxim Integrated Products, Inc. (2010, June) MAX2769 Universal GPS Receiver Rev 2. [Online]. <http://datasheets.maxim-ic.com/en/ds/MAX2769.pdf>
- [17] MIPS Technologies, Inc. (2008, March 4) MIPS32® M4K® Processor Core. Datasheet. [Online]. <http://www.mips.com/media/files/MD00247-2B-M4K-DTS-02.01.pdf>
- [18] Microchip Technology Inc. (2012, January) Microchip TCP/IP Download & Support. [Online]. http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1489
- [19] Texas Instruments Incorporated. (2011, September) TPS767D318 Dual-Output Low-Dropout (LDO) Voltage Regulators. Website. [Online]. <http://www.ti.com/product/tps767d318>
- [20] LENA. (2012, March) Programm zur Analyse elektronischer, linearer und nichtlinearer Schaltungen im Frequenz- oder Zeitbereich. Website. [Online]. <http://www.hfdollinger.de/14132.html>
- [21] Henry Ott Consultants. (2012, March) PCB Stack-Up - Part 2. [Online]. <http://www.hottconsultants.com/techtips/pcb-stack-up-2.html>
- [22] Sebastian Stoff. (2012, June) Orbitron - Satellite Tracking System. [Online]. <http://www.stoff.pl/>
- [23] Artyom Gavrilov. (2012, July) GNSS-SDR software defined radio technique in GPS/GLONASS receivers. [Online]. <http://gnss-sdr.com/>
- [24] General Dynamics Corporation. (2012, July) Namuru. [Online]. http://dynamics.co.nz/index.php?main_page=page&id=9
- [25] WIZnet Inc. (2009, January) WIZ812MJ Datasheet (Ver. 1.1). [Online]. [http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/WIZ812MJ_Datasheet_V_1.1\[1\].pdf](http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/WIZ812MJ_Datasheet_V_1.1[1].pdf)
- [26] Russian Federation. (2011, February) Federal Space Agency, Information-Analytical Center. [Online]. <http://www.glonass-ianc.rsa.ru/>

Appendix I –EPCOS B9415 SAW Filter transfer function

Transfer function (passband detail)



Transfer function

