

## Conception of a telecommunication module for a nanosatellite

**Auteur :** Diepart, Morgan

**Promoteur(s) :** Redouté, Jean-Michel

**Faculté :** Faculté des Sciences appliquées

**Diplôme :** Master : ingénieur civil électricien, à finalité spécialisée en "electronic systems and devices"

**Année académique :** 2020-2021

**URI/URL :** <http://hdl.handle.net/2268.2/11544>

---

### *Avertissement à l'attention des usagers :*

*Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.*

*Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.*

---



University of Liège - Faculty of Applied Sciences

---

# Conception of a telecommunication module for a nanosatellite

---

Master's thesis submitted by  
Morgan Diepart  
in partial fulfilment of the requirements for the  
Master's degree in Electrical Engineering

Academic year 2020 - 2021

## Acknowledgements

It is no secret that during the creation of a work as large as a Master's thesis many person do get involved in a way or another.

At first I would like to thank Prof. Jean-Michel Redouté for his help and the motivation he provided me all along this work. I must also thank Hervé Pierre which first hinted me towards this thesis and helped me nearly everyday obtaining these results.

I also want to thank Dr. Fabrice Frébel for his punctual help in this work and that allowed me to understand a great deal of things when it comes to RF design.

The major part of this work was performed inside the labs of Microsys and so I want to thank their personnel. In particular thank to Samuel Dricot for the time he spent assembling the boards for this work, thank to François Piron and Gabriel Di Gregorio for their help and entertainment.

Finally thank to my room mates who made me cry and laugh when I needed it the most, thank you to Roxane Housen for the proofreading, thank to Léo for supporting me and thank to my parent to whom I owe so much.

This work would not be the same without all of you. Thank you.



University of Liège  
Faculty of Applied Sciences, Electrical Engineering

---

# Conception of a telecommunication module for a nanosatellite

MORGAN Diepart

Supervisor:  
Prof. JEAN-MICHEL Redouté

Academic Year 2020-2021

---

## Abstract

This work presents the development of a custom telecommunication module for a nanosatellite.

The study will start at the ground station and the limitations it imposed on the previous mission with a goal to get out of those. This will lead to a new proposition for the ground station. This new solution proposed will also be used to during the development of the telecommunication module facilitating the process by removing the need of bulky radio hardware.

The board will then be developed from the ground up. At first the components required will be selected. Several circuits will then be developed to perform measurements on every piece of hardware to confirm that their performances are compatible with the goal of the module. Different measurement methods will be used to better fit the specific pieces of hardware. Whenever needed the design will be tweaked to attain the best performances possible. Once all the measurements are performed, the performances of the components are known and the design are independently optimised a complete board will be created. It will include the transceivers, the power amplifier and the low noise amplifier. The required voltage regulators and other subsystems needed will be added on this last board.

A firmware will finally be developed on the same microprocessor and software architecture as what the on-board computer. This firmware contains a hardware abstraction layer for the transceiver and FreeRTOS tasks for both the emission and reception of AX.25 packets. When the software of the on-board computer will be written it will be able to use the telecommunication module with a small and concise set of functions.





Université de Liège  
Faculté des Sciences Appliquées, Ingénieur Civil Électricien

---

# Conception d'un module de télécommunication pour nanosatellite

MORGAN Diepart

Promoteur:  
Pr. JEAN-MICHEL Redouté

Année Académique 2020-2021

---

## Résumé

Ce travail présente le développement d'un module de télécommunication personnalisé pour nanosatellite.

L'étude va commencer par la station-sol et les limitations imposées à la première mission dans le but de s'en affranchir. Cela va mener à une nouvelle proposition pour la station-sol. Cette nouvelle solution proposée sera aussi utilisée durant le développement du module de télécommunication, facilitant le procédé en enlevant le besoin du matériel radio encombrant.

La carte sera ensuite développée depuis zéro. En premier lieu, les composants requis seront sélectionnés. Plusieurs circuits différents seront ensuite développés pour pouvoir effectuer des mesures sur chaque composant pour confirmer que leurs performances sont compatibles avec le but du module. Différentes méthodes de mesure seront utilisées pour mieux s'adapter aux composants spécifiques. Quand il y a besoin la conception sera adaptée pour pouvoir atteindre les meilleures performances possibles. Une fois toutes les mesures effectuées, les performances des composants connues et les designs indépendamment optimisés, une carte complète sera créée. Elle inclut les émetteurs-récepteurs, l'amplificateur de puissance et l'amplificateur de faible bruit. Les régulateurs de tensions requis ainsi que les autres sous-systèmes seront ajoutés à cette dernière carte.

Finalement un micrologiciel sera développé sur le même microcontrôleur et la même architecture logicielle que ce qui sera utilisé sur l'ordinateur de bord. Ce micrologiciel contient la couche d'abstraction matérielle pour les émetteurs-récepteurs et les tâches FreeRTOS pour l'émission et la réception des paquets AX.25. Quand le logiciel de l'ordinateur de bord sera écrit, il pourra utiliser le module de télécommunication à l'aide d'un ensemble de fonctions restreint et concis.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Nanosatellites . . . . .	1
1.2	OUFTI1 . . . . .	1
1.3	The Communication Subsystem . . . . .	1
1.4	What this work will be about . . . . .	1
1.5	Context . . . . .	3
<b>I</b>	<b>Terminal Node Controller and Software Defined Radio</b>	<b>4</b>
<b>2</b>	<b>Preliminary analysis</b>	<b>5</b>
2.1	The Ground Station . . . . .	5
2.2	The Communication Protocol . . . . .	5
2.2.1	AX.25 Protocol . . . . .	5
2.2.2	Terminal Node Controller . . . . .	6
2.2.3	G3RUH Modem . . . . .	7
2.2.4	Constraints from using the SCS Tracker / DSP TNC . . . . .	7
2.2.5	Constraints from the G3RUH modem . . . . .	7
<b>3</b>	<b>Available Solutions</b>	<b>9</b>
3.1	Data encoding and baseband signal generation . . . . .	9
3.1.1	Dire Wolf . . . . .	9
3.1.2	GNURadio . . . . .	9
3.1.3	PothosWare . . . . .	9
3.2	Frequency modulation and RF signal generation . . . . .	10
3.2.1	RF generation using analogue audio signal . . . . .	10
3.2.2	RF generation using digital signal . . . . .	10
3.3	Synchronisation word . . . . .	10
<b>4</b>	<b>Recommended Solution</b>	<b>12</b>
4.1	Software . . . . .	12
4.1.1	Data Generation . . . . .	12
4.1.2	Software TNC . . . . .	12
4.1.3	Frequency (de-)modulation . . . . .	14
4.2	Hardware . . . . .	14
4.3	Synchronisation Word . . . . .	16
4.4	Summary . . . . .	19
4.5	Results . . . . .	19
4.6	Conclusion . . . . .	21
4.7	To go further . . . . .	21

<b>II</b>	<b>Module Design</b>	<b>23</b>
<b>5</b>	<b>First Hardware design</b>	<b>24</b>
5.1	Components choice . . . . .	24
5.1.1	Criteria . . . . .	24
5.1.2	Power Amplifier . . . . .	25
5.1.3	Low Noise Amplifier . . . . .	25
5.1.4	Transceiver . . . . .	26
5.2	Schematic design . . . . .	27
5.2.1	Power Amplifier . . . . .	27
5.2.2	Low Noise Amplifier . . . . .	27
5.2.3	Transceiver . . . . .	27
5.2.4	Output Low Pass Filter . . . . .	32
5.3	Layout design . . . . .	32
5.3.1	Power Amplifier . . . . .	33
5.3.2	Low Noise Amplifier . . . . .	34
5.3.3	Transceiver . . . . .	34
5.4	Test of the components . . . . .	35
5.4.1	Power Amplifier . . . . .	35
5.4.2	Low Noise Amplifier . . . . .	37
5.4.3	Transceivers . . . . .	38
5.4.4	Output Low Pass Filter . . . . .	45
<b>6</b>	<b>Second Hardware Design</b>	<b>46</b>
6.1	Modifications . . . . .	46
6.1.1	Low Noise Amplifier . . . . .	46
6.1.2	Temperature Compensated Crystal Oscillators . . . . .	46
6.1.3	Main Connector . . . . .	46
6.2	New Subsystems . . . . .	47
6.2.1	Current measurement . . . . .	47
6.2.2	3.0 V Linear Regulator . . . . .	48
6.2.3	2.8 V Linear Regulator . . . . .	48
6.3	Tests . . . . .	48
<b>7</b>	<b>Firmware</b>	<b>50</b>
7.1	Overview of the ADF7021-N . . . . .	50
7.1.1	Hardware Block Diagram . . . . .	50
7.1.2	Interfaces . . . . .	50
7.1.3	Configuration Registers . . . . .	52
7.1.4	Image Rejection . . . . .	62
7.2	Software Architecture . . . . .	64
7.3	Written Code . . . . .	65
7.3.1	Code for the ADF7021-N . . . . .	66
7.3.2	AX.25 Packet . . . . .	67
7.3.3	FreeRTOS Tasks . . . . .	67
	<b>Conclusion</b>	<b>70</b>
	<b>A Link Budget</b>	<b>71</b>
	<b>B Configuration file used for Dire Wolf Software TNC</b>	<b>73</b>
	<b>C LNA Comparison</b>	<b>84</b>

<b>D</b>	<b>Test Circuits Schematics</b>	<b>85</b>
D.1	Power Amplifier Test Circuit . . . . .	86
D.2	Low Noise Amplifier Test Circuit . . . . .	87
D.3	Tx Transceiver Test Circuit . . . . .	88
D.4	Rx Transceiver Test Circuit . . . . .	89
D.5	Transceiver Test Circuit Connector . . . . .	90
D.6	Tx Low Pass Filter . . . . .	91
<b>E</b>	<b>Final Board Schematic</b>	<b>92</b>

# List of Figures

1.1	Block diagram of the telecommunication module used for OUFTI1[1]. . . . .	2
1.2	Block diagram of the telecommunication system to be designed for the new nanosatellite. . . . .	2
2.1	Radio chain of the ground station used for OUFTI1. The emission is performed at a frequency of 435.015 MHz. The reception is performed at a frequency of 145.95 MHz.[2, 3]	6
2.2	AX.25 Frame Construction. . . . .	6
3.1	Radio chain needed in order to use analogue audio output from <i>Dire Wolf</i> . The data is generated on the computer. A custom interface is needed to hook-up the signals to the ICOM IC-910 radio. The rest of the radio chain is omitted.[2] . . . . .	10
3.2	Radio chain needed in order to use a digital output from <i>Dire Wolf</i> to an SDR. The FM signal is generated on the computer and sent to the SDR via USB. The PTT signal must be connected to the amplifiers. The rest of the radio chain is omitted.[7] . . . . .	11
4.1	<i>GNURadio</i> flow graph used to interface Dire Wolf with the software defined radio. Some other blocks have been added to display some signals and spectrums while the flow graph is running. . . . .	16
4.2	Bode diagram of the filter used at the output of the SDR. The cutoff frequency is 634 MHz and the frequency of the signal at 435.015 MHz is located at the top of the first ripple and is not attenuated. Due to the way the filter is simulated in LTSpice it shows a 6 dB attenuation in the pass-band region instead of 0 dB. . . . .	17
4.3	Audio output of three successive transmissions performed just after starting <i>Dire Wolf</i> . The beginning of the three transmissions look different even though the data sent up to there is the same. This means that the LFSR is not reset between each transmission. . . . .	17
4.4	Audio output of five successive transmissions using <i>Dire Wolf</i> (modified to reset the LFSR and NRZI encoder after each transmission). The transmissions look identical except for their polarity which is not always the same across transmissions. This is because the phase of the audio tone generator is not reset between each transmission. . . . .	17
4.5	Audio output of five successive transmissions using <i>Dire Wolf</i> (modified to reset the LFSR, NRZI encoder and audio phase after each transmission). The transmissions look identical. . . . .	18
4.6	Proposed frames to be transmitted by the ground station to the nanosatellite. A synchronisation word have been added. . . . .	18
4.7	Impedance measurement of the filter. The input and output impedances of the filter are acceptable. The insertion losses do look like the simulation from figure 4.2. . . . .	19
4.8	Comparison of the SDR spectrum with and without the filter. A slight attenuation is noticeable with the filter. The shape stays the same. . . . .	20
4.9	Occupied bandwidth measurement on the signal transmitted by the SDR. The measured occupied bandwidth is 18.4 kHz. . . . .	20
4.10	Comparison of the harmonic distortions with and without the filter. . . . .	21
5.1	Output power and gain of the Qorvo RF5110G. The device will be supplied with 3.3 V. . . . .	26
5.2	$S_{21}$ (gain) parameter curve given by Infineon and measured using their recommended design. The curve to look at is the pink dashed curve. The gain at 435 MHz should be a little bit below 20 dB. . . . .	28
5.3	VCO external inductance value vs. RF output frequency. . . . .	29

5.4	Output matching Smith chart. The source impedance is the point DP1. The point DP2 represents the source impedance seen by the load after the inductor and before the capacitor. The point DP3 represents the source impedance seen by the load after the full matching network. . . . .	30
5.5	Lumped elements model of the RF input impedance of the ADF7021-N transceiver. At 435 MHz $C_A = C_C = 2.225$ pF, $R_A = R_C = 331.5 \Omega$ , $L_B = 2.85$ $\mu$ H, $R_B = 9.55$ k $\Omega$ . . . .	31
5.6	Balun design tool in <i>AppCAD</i> software. The design used is the "4-element Balun". . .	31
5.7	LTSpice Simulation of the Chebyshev low-pass filter. The frequency used of 145.95 MHz is located on top of the first ripple and is not attenuated. Due to the way the filter is simulated in LTSpice it shows a 6 dB attenuation in the pass-band region instead of 0 dB. . . . .	32
5.8	Test circuit built to measure the parameters of the RF5110G power amplifier. . . . .	34
5.9	Test circuit built to measure the parameters of the BGB741L7ESD low noise amplifier. . .	34
5.10	Test circuit built to measure the characteristics of the TX and RX transceivers. . . . .	35
5.11	Gain measurement of the RF5110G on the VNA. The gain measured saturates at about 20 dB while a gain of 34 dB is expected. . . . .	36
5.12	Setup used to measure the performances of the RF5110G Power Amplifier. The AWG generated a sinusoidal signal with a variable power and frequency. . . . .	36
5.13	Test setup to measure the parameters and performance of the low noise amplifier. . . .	37
5.14	$I_{CC}$ current consumption of the low noise amplifier versus the value of the $R_{ext}$ resistor used to set the value of $I_{CC}$ . The device operates at 3.0 V. . . . .	38
5.15	Caption . . . . .	39
5.16	VNA measurement of the input impedance of the receiving transceiver with the modified shunt inductor. The return loss at 435.015 MHz is $-14.45$ dB and the port impedance is $60 - 19i \Omega$ . . . . .	40
5.17	Comparison of the measured output power and the output power given in the datasheet. . . . .	42
5.18	Occupied Bandwidth measurement of the signal generated by the ADF7021-N transceiver. The bandwidth is measured to be 22.49 kHz . . . . .	42
5.19	Spurious emissions measurement at the output of the Tx ADF7021-N. The highest spurious emission recorded is 71 dB below the carrier power level. . . . .	43
5.20	Setup used to measure the sensitivity of the RX transceiver. . . . .	44
5.21	Spectrum analyser view of the signal generated by the SDR after a series of attenuators to test the sensitivity of the RX transceiver. . . . .	45
5.22	VNA measurement of the characteristics of the low-pass filter used to filter out the harmonics of the 145.95 MHz downlink signal. The impedance is close to $50 \Omega$ and the signal is located at the top of the attenuation curve. . . . .	45
6.1	Comparison of the PC/104 board mechanical specification versus the PC/104 plus board mechanical specification. . . . .	47
6.2	Pumpkin CubeSat Kit Mechanical Specification . . . . .	48
6.3	3D rendering of the Printed Circuit Board (PCB) of the second hardware design. . . .	49
7.1	Hardware block diagram of the ADF7021-N transceiver. . . . .	51
7.2	Content of the register 0 of the ADF7021-N. . . . .	52
7.3	Content of the register 1 of the ADF7021-N. . . . .	53
7.4	Content of the register 2 of the ADF7021-N. . . . .	54
7.5	Content of the register 3 of the ADF7021-N. . . . .	55
7.6	Content of the register 4 of the ADF7021-N. . . . .	56
7.7	Content of the register 5 of the ADF7021-N. . . . .	57
7.8	Content of the register 6 of the ADF7021-N. . . . .	58
7.9	Content of the register 7 of the ADF7021-N. . . . .	59
7.10	Content of the register 8 of the ADF7021-N. . . . .	59
7.11	Content of the register 9 of the ADF7021-N. . . . .	60
7.12	Content of the register 10 of the ADF7021-N. . . . .	60
7.13	Content of the register 11 of the ADF7021-N. . . . .	61
7.14	Content of the register 12 of the ADF7021-N. . . . .	62

7.15	Content of the register 15 of the ADF7021-N. . . . .	63
7.16	Representation of the source of the image signal problem. $\omega_{SIG}$ is the frequency of the signal of interest (435.015 MHz), $\omega_{LO}$ is the frequency of the local oscillator (434.915 MHz), $\omega_{IMG}$ is the frequency of the image signal (434.815 MHz). . . . .	63
7.17	Block diagram showing the RF input of the ADF7021-N and the system allowing to reject the signals at the image frequency, used for the image rejection calibration. . . .	64
7.18	Representation of the RSSI levels read for each pair of values (gain, phase) possible. A local minimum can be seen around (gain=-4, phase=0) while a lower value can be measured at (gain=-7, phase=-6). The vertical axis is in ADC counts. . . . .	65
7.19	<i>Harmony 3</i> project graph. . . . .	66

# Glossary

- 2FSK** 2-levels Frequency Shift Keying 5, 7, 13, 20, 26, 41, 54–56, 62
- ADC** Analogue to Digital Converter ix, 58, 59, 64, 65
- AFC** Automatic Frequency Correction 44, 45, 58, 60, 61, 67
- AGC** Automatic Gain Control 55, 60, 61, 67
- AWG** Arbitrary Waveform Generator viii, 35, 36
- CDR** Clock and Data Recovery 10, 18, 55
- COMM** Communication subsystem 1, 8, 12, 50
- CP** Charge Pump 32, 53, 54
- CPU** Central Processing Unit 14
- CRC** Cyclic Redundancy Code 67
- CSMA** Carrier Sense Multiple Access 13
- DCD** Data Carrier Detect 10, 14, 21, 22
- EMC** ElectroMagnetic Compatibility 33
- EPS** Electrical Power System 48
- FCS** Frame Check Sequence 5, 6
- FEC** Forward Error Correction 9
- GPIO** General Purpose Input/Output 22, 50, 51
- GUI** Graphical User Interface 9
- HAL** Hardware Abstraction Layer 65, 66
- HDLC** High-Level Data Link Control 5, 18, 68
- IF** Intermediate Frequency 52, 56–58, 61, 62, 67
- IO** Input/Output 46, 64, 65
- ISR** Interrupt Service Routine 66, 68
- KISS** Keep It Simple, Stupid 6, 13, 14, 19
- LDO** Low-Dropout regulator 48
- LFSR** Linear Feedback Shift Register vii, 7, 11, 16–18, 67
- LNA** Low Noise Amplifier 3, 27, 29, 30, 34, 45–48, 59, 60
- NRZI** Non Return to Zero Inverted vii, 7, 16–18
- OBC** On Board Computer 1, 8, 33, 46, 48, 50, 65
- OBW** Occupied BandWidth 20, 41, 71
- OIP3** Output Third Order Intercept Point 25, 26
- OOT** Out-Of-Tree 9, 14
- P1dB** Output 1dB Compression Point 26, 35
- PA** Power Amplifier 3, 27, 29, 32, 35, 38, 46–48, 55, 59, 67
- PCB** Printed Circuit Board viii, 28, 32, 33, 49
- PLL** Phase-Locked Loop 10, 18, 28, 32, 51–53, 67
- PTT** Push To Talk vii, 10, 11, 13, 14, 21, 22
- RSSI** Received Signal Strength Indicator ix, 40, 43, 44, 58–60, 62, 64, 65, 68
- SDR** Software Defined Radio vii, viii, 9–12, 14, 19, 20, 22, 40, 43, 45
- SNR** Signal to Noise Ratio 32, 45, 71
- SPI** Serial Peripheral Interface 51, 66
- SWD** Synchronisation Word Detection 6, 16, 18, 44, 51, 61, 67, 68
- TCXO** Temperature Compensated Crystal Oscillator 27, 46, 53, 54, 67
- TNC** Terminal Node Controller 5–14, 18, 19, 21, 41, 56
- UART** Universal Asynchronous Receiver Transmitter 51–53
- VCO** Voltage Controlled Oscillator viii, 28, 29, 32, 38, 40, 41, 53, 54, 67
- VNA** Vector Network Analyser viii, 35, 36, 38–41, 43, 45
- VSWR** Voltage Standing Wave Ratio 24, 29



# Chapter 1

## Introduction

### 1.1 Nanosatellites

A nanosatellite is a kind of satellite characterised by a weight of less than 10 kg and a length of a few tens of cm maximum. The most known type of nanosatellite is the CubeSat whose size is counted in units, or U. A unit is a cube of  $10\text{ cm} \times 10\text{ cm} \times 10\text{ cm}$ . The common sizes used are 1U, 3U and 6U. This work will develop a telecommunication module for a 1U CubeSat. A 1U CubeSat is thus contained in a  $10\text{ cm} \times 10\text{ cm} \times 10\text{ cm}$  cube with a weight of maximum 1.3 kg and an average power consumption of 1 W.

### 1.2 OUFTI1

OUFTI1 stands for Orbital Utility For Telecommunication Innovation and is the first satellite launched by the University of Liège. It is an educational 1U CubeSat developed by students from the University of Liège and the Haute École de la Province de Liège. It was launched on the 25<sup>th</sup> of April 2016 and reached orbit alive. However the communication could never be established and after 12 days the beacon stopped emitting. One of the payloads of the satellite was a D-Star relay and was supposed to allow very long range communication via the D-Star protocol for radio-amateurs all around the world.

All this left the university with a fully-functional ground station and good knowledge in the domain. Ever since the desire to re-iterate the experience while learning from the first mission is present and an attempt at developing a new satellite gave the project OUFTI2.

The OUFTI2 project sadly never came to an end despite the number of thesis already written. There thus exist a solid ground to develop a new project which must be exploited.

### 1.3 The Communication Subsystem

Both OUFTI1 and OUFTI2 had developed their own Communication subsystem (COMM). Those were both based on the same architecture and both included the D-Star relay which highly increased the complexity the design.

The figure 1.1 shows the block diagram of the telecommunication module used for OUFTI1. However it have been decided to change the architecture of the CubeSat and to simplify the telecommunication module. The first decision made is to remove the microcontroller of the COMM module and delegate its tasks to the On Board Computer (OBC). It was also decided to get rid of the D-Star repeater. A block diagram of the resulting design is displayed in figure 1.2.

### 1.4 What this work will be about

This work will develop a complete telecommunication module for a nanosatellite. However the development of a telecommunication module is tightly linked to the ground station because it tends to impose some constraints to what is feasible and can be implemented. This work will be split in two parts.

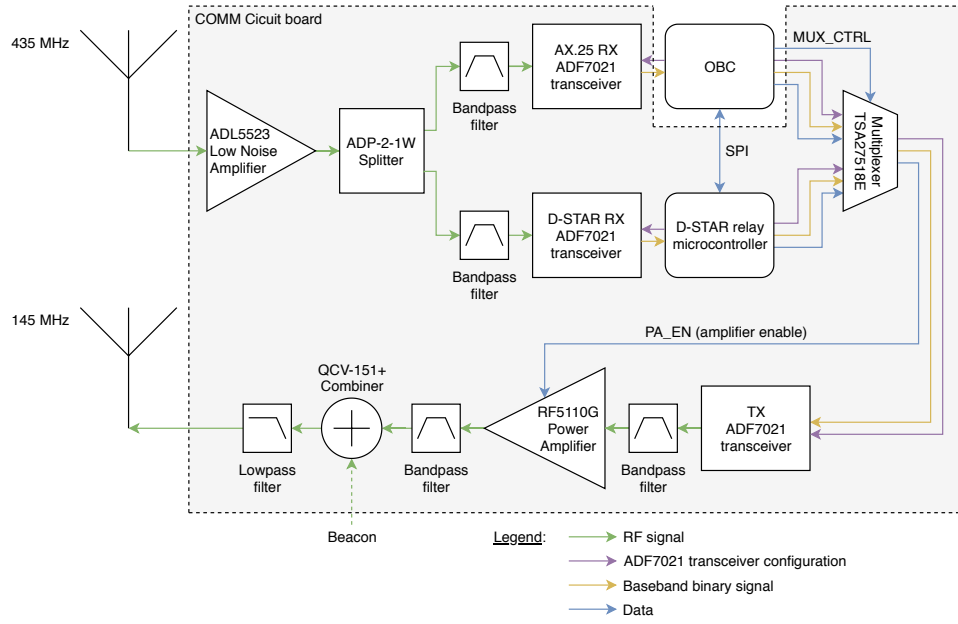


Figure 1.1: Block diagram of the telecommunication module used for OUFTI1[1].

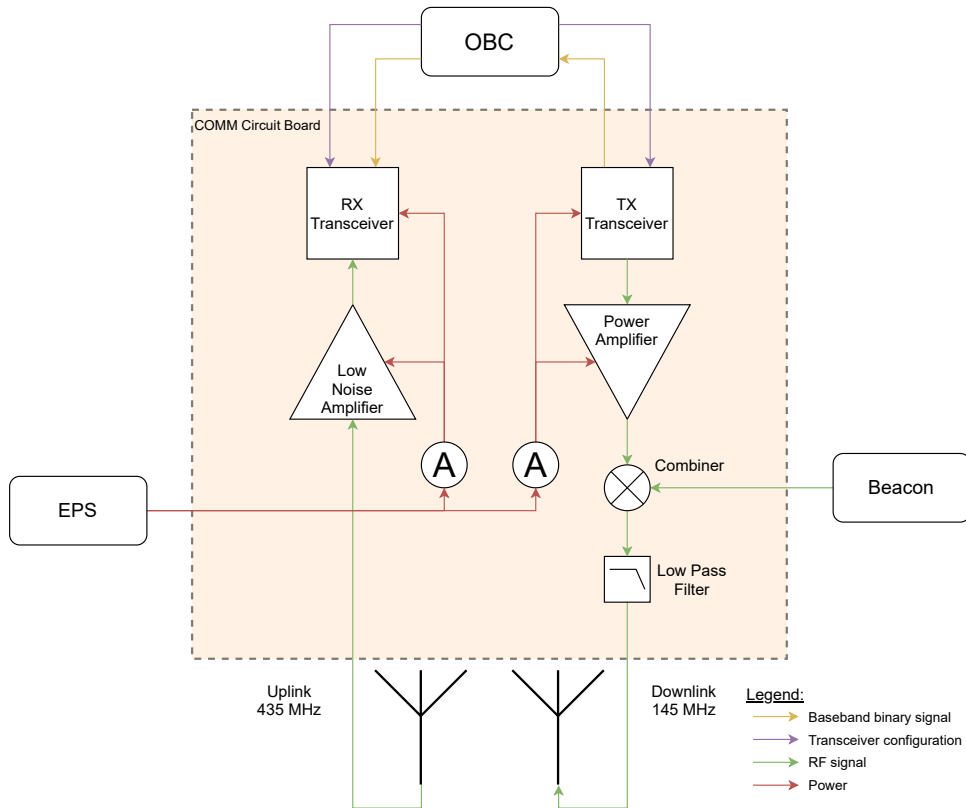


Figure 1.2: Block diagram of the telecommunication system to be designed for the new nanosatellite.

- Part I will analyse the current ground station and its hardware. The constraints will be analysed and solutions will be created to remove as many constraints as possible.
- Part II will then go through the needed steps to develop a fully-functional telecommunication module in three steps:
  - the first step in chapter 5 will focus on the selection of the components (Low Noise Amplifier (LNA), Power Amplifier (PA) and transceivers) and their individual tests;
  - the second step in chapter 6 will assemble the circuits of the previous step in a single module;
  - the third step in chapter 7 will go through the writing of the firmware needed to configure the telecommunication module and use it.

This work will however not re-discuss the topics that were already the topics of many previous works such as the management of which frequency to use on each link or the management of Doppler shift, the signals that will be present at the input of the CubeSat,... .

## 1.5 Context

This work have been done during the academic year 2020-2021. Everybody will remember the sanitary context of this year and the previous. While the students working on this kind of topic usually attend several activities about the development of satellite missions this was understandably not possible this year. This context also slowed down the deliveries and productions of circuits needed for this work. It was finally not possible to get the amateur radio licence required to perform radio telecommunications but the tests were performed via wired channels so this was not a major issue.

## Part I

# Terminal Node Controller and Software Defined Radio

# Chapter 2

## Preliminary analysis

### 2.1 The Ground Station

While it may be tempting to believe that the design of the communication module of a CubeSat is independent of the ground station, some of its technical characteristics will impact the design of the circuit.

The University of Liège already have a ground station set-up that have been used for the OUFTH project. It is able to transmit AX.25 packets at a radio frequency of 145.95 MHz and a power of 100 W. It is also able to receive AX.25 packets at a radio frequency of 435.015 MHz. The full chain, from the generation of the packet up to the emission of the 2-levels Frequency Shift Keying (2FSK) signal is described here and graphically explained in figure 2.1.

1. The data of the packet is generated via a dedicated user interface on a computer which is connected via a serial connection to the Terminal Node Controller (TNC).
2. The data sent by the computer is received on the TNC. Flags are prepended and appended to the data which is then converted to a binary stream (see section 2.2.3 for details). This baseband binary stream is then low-pass filtered and outputted on the mini-DIN connector (at a baud rate of 9600 Bd).
3. The analogue signal generated by the TNC is received by the radio which modulates the signal with a deviation frequency of 4800 Hz. This frequency modulated signal is then brought up to the carrier frequency of 435.015 MHz.
4. The radio-frequency signal produced by the radio is then amplified by the power amplifier up to a power of 100 W.
5. The amplified signal is then fed to an antenna whose heading can be controlled to be pointed towards the CubeSat.

### 2.2 The Communication Protocol

This ground station was designed to communicate using the AX.25 Protocol and for that it used the SCS Tracker / DSP TNC as its terminal node controller which is what allows someone to communicate via the AX.25 protocol.

#### 2.2.1 AX.25 Protocol

The AX.25 protocol is an amateur version of the X.25 protocol and uses High-Level Data Link Control (HDLC) frames syntax (i.e. flags, fields length, Frame Check Sequence (FCS),...)[4, 5]. The standards mainly defines the OSI layers 2 and 3 that are used in the protocol and there exist several types of AX.25 frames:

- information frames (I),

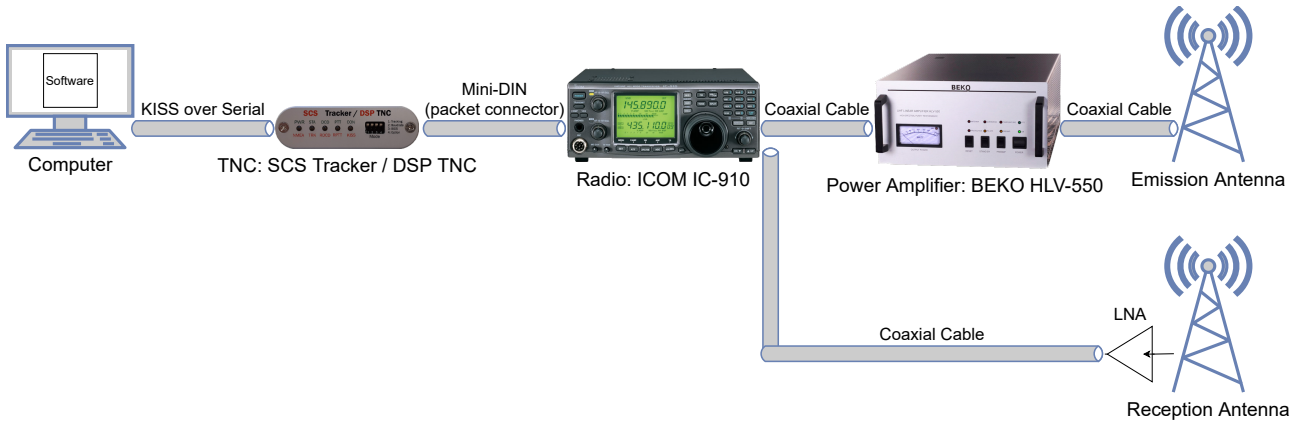


Figure 2.1: Radio chain of the ground station used for OUFTI1. The emission is performed at a frequency of 435.015 MHz. The reception is performed at a frequency of 145.95 MHz.[2, 3]

- supervisory frames (S),
- unnumbered frames (U).

The I frames are mainly used to transfer data while the S and U frames are mainly used to provide control over the established link (opening or closing a connection, requesting re-transmission,...). The structure of an AX.25 frame is displayed in table 2.2.

Flag	Address	Control	PID (I frames only)	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	8 Bits	N*8 Bits	16 Bits	01111110

Figure 2.2: AX.25 Frame Construction.

The flags can actually be repeated several times and when a transmitter is on but is not transmitting any frame it should continuously send flags.

### 2.2.2 Terminal Node Controller

Terminal node controllers can often be used in two main ways.

The first way is to let it manage the full protocol including starting and ending a transmission, acknowledgements, re-transmissions,... . This used to be a dedicated piece of hardware because the computational power required was too high to be managed by the same computer that is already managing the upper layers.

The second use case is to use the TNC in what is called Keep It Simple, Stupid (KISS) mode. This mode let the user manage the layer 3 of the AX.25 protocol, meaning that the TNC does not manage anymore the startings and endings of transmissions, acknowledgements, re-transmissions,... . Those are managed by the user. The TNC however still manages the encoding needed and the flags to be inserted.

In both operating modes the TNC usually selects specific modems depending on the communication speed selected. At 9600 Bd the modem is the G3RUH.

Since the creation of the first hardware TNCs however, the computational power of the computers have increased significantly. Nowadays, a single modest computer can easily handle much more than what a typical TNC used to perform. This lead to the apparition of software TNCs such as the *Dire Wolf software TNC*.

This piece of hardware was a major constraint in the design of the telecommunication modules and was one of the topics in the thesis of François Piron [1]. The goal of his work was to determine how the Synchronisation Word Detection (SWD) capability of the ADF7021 transceiver could be exploited to avoid interrupting the microcontroller 9600 times per second to detect the beginning of a transmission. This work concluded that a specific bit sequence was recognisable in the flags of the preamble of every packet transmitted by that TNC and that it could be used as a synchronisation word.

This major piece of hardware however, was nowhere to be found when this work started (it is still lost upon writing these words). It was thus required to find a workaround in order to at least be able to test the circuit that would be designed.

It was decided to not only find a workaround to test the hardware designed but also to perform researches on what the available solutions are currently so that the TNC could be replaced indefinitely by a more modern solution.

### 2.2.3 G3RUH Modem

The G3RUH modem is used when communicating using 9600 Bd 2FSK modulated AX.25 frames.

Several steps are used in the modem in order to improve the communication and are described here.

1. The preamble and postamble flags are prepended and appended to the data to form a frame.
2. The data in the frame is then bit-stuffed. That is, when there are five '1' in a row, a '0' is inserted. Those extra symbols will need to be discarded by the receiver. The flags are specifically not bit-stuffed such that the pattern stays recognisable by the receiver (six '1' in a row).
3. The bit-stuffed frame is then encoded using Non Return to Zero Inverted (NRZI) encoding. That means that the signal is now represented by transitions in the signal and not the actual symbols. A '1' is represented by no transition. A '0' is represented by a transition. Due to the bit-stuffing from the first step, the signal is guaranteed to have a transition at least every 5 symbols. It also guarantees that, at the reception, the decoded signal will be the same even if it is flipped.
4. The NRZI encoded frame is then scrambled using a 17 bits long Linear Feedback Shift Register (LFSR).

Those steps are used to reduce any DC-offset that may appear during the communication. It is also supposed to reduce the duration of times without any transition which could cause the receiver to de-synchronise from the transmitter data rate. It is also useful to specify that because of the way the data is scrambled (using a 17 bits LFSR) it can be unscrambled without having any knowledge of the state of the emitter's LFSR. After receiving 17 scrambled bits the unscrambled output of the receiver is guaranteed to be the same as the unscrambled input of the emitter (despite the receiver's scrambler not being in the same state as the emitter's scrambler). This makes it so that two identical frames emitted could be completely different from one another once scrambled. This is because the initial state of the scrambler was different upon starting the transmission but after having received 17 bits the receiver will still receive the same data in each case.

### 2.2.4 Constraints from using the SCS Tracker / DSP TNC

In order to orient the researches it is useful to list the constraints from working with that piece of hardware.

The first one is obvious and that is that this specific piece of hardware is needed to perform the tests. It requires an external power supply, communicates via a serial link to a computer and after this a radio is still required to transmit an actual RF signal.

The second one is that the operations it follows tends to be quite obscure and understanding some things required extensive reverse engineering.

Finally, at 9600 Bd it uses the G3RUH modem and nothing can be done about that. The operations can not be modified and the telecommunication module have to deal with what the TNC can and can not do.

### 2.2.5 Constraints from the G3RUH modem

Historically this modem was useful because it provides good transmission characteristics (reduced DC offset and regular signal transitions allowing the receiver to lock more easily to the sender's data rate, whatever the transmitted data may be). The scrambling and unscrambling were performed using

discrete components and thus did not require any computing. As the AX.25 protocol (and thus the detection of flags to signal the beginning of a frame) was furthermore performed by a TNC which is a dedicated piece of hardware the computer managing the communication did not have to be kept busy trying to recognise flags in the received bit stream. However, here the signal is frequency demodulated by the transceiver and then directly fed to the microprocessor of the OBC meaning that no processing is offloaded to the equivalent of a TNC. That combined with the fact that the received scrambled signal could be completely different for two identical unscrambled frames makes it hard to detect when a frame is actually being received or when the transceiver is outputting random data from the picked-up ambient noise. To solve this solution for OUFTI1 it was decided that the COMM microcontroller of OUFTI1 would continuously unscramble incoming data to check if a flag was received (this required interrupts at a rate of 9600 times per seconds).



## Chapter 3

# Available Solutions

Even though a new TNC could simply be bought it was decided to opt for a new solution that would give more control over the transmission.

### 3.1 Data encoding and baseband signal generation

#### 3.1.1 Dire Wolf

Decoded Information from Radio Emissions for Windows or Linux Fans, or *Dire Wolf* for short is a software TNC that is designed to use a sound card to output the baseband binary signal. It is an open-source project that have been existing since 2015 and is still in active development today. It implements a wide variety of functionalities that are commonly implemented in traditional TNC but that are not all of interest for this project. It also implements the FX.25 variant of AX.25 which includes Forward Error Correction (FEC) in the packet to allow for higher communication reliability while still being compatible with traditional AX.25 equipment.

One of the main problem that could arise from the use of this solution is the fact that the sound card used to output the generated signal needs to have a frequency response as flat as possible from DC up to 5000 Hz for a frequency deviation of 4800 Hz[6]. However the baseband signal does not have to leave the computer the TNC is running on if an Software Defined Radio (SDR) is used.

#### 3.1.2 GNURadio

*GNURadio* is an open-source software that is commonly used to interface with SDR hardware. Its main interface, called *GNURadio Companion* is a Graphical User Interface (GUI) that allows one to perform signal processing using a flow-graph based approach. It offers, out of the box, a pretty decent collection of blocks that are used to perform operations from sourcing a signal to outputting a signal, as well as alter it depending on the user's needs. One of its main strength is the large number of Out-Of-Tree (OOT) modules (developed by the community and designed to be easily added to the library of *GNURadio*). The blocks can be written either in Python or in C++. This makes it possible to implement a TNC that could for example take an input stream via a UDP or TCP port and generate baseband signals according to our needs. However no such blocks exists and it would need significant software development in order to create the required blocks.

#### 3.1.3 PothosWare

*PothosWare* is an alternative to *GNURadio*. It also is an open-source software. It have slightly different functionalities than *GNURadio*. It also have a module that allows for any block that was added in a *GNURadio* install to be used in *PothosFlow* (the GUI of *PothosWare*). It gives more control to the user over some functionalities and also seems to be more efficient than *GNURadio* to perform certain tasks. However the module supposed to import the blocks installed in *GNURadio* seems to not be compatible with the latest *GNURadio* versions which creates compatibility issues and in the end makes the solution very impractical to use.

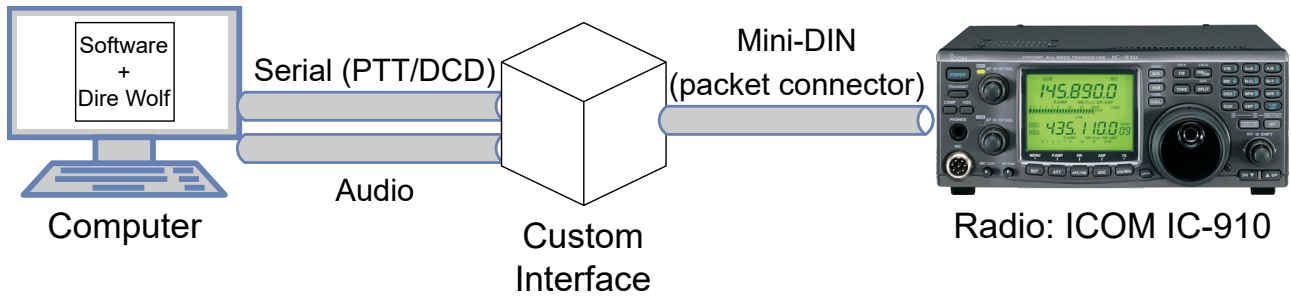


Figure 3.1: Radio chain needed in order to use analogue audio output from *Dire Wolf*. The data is generated on the computer. A custom interface is needed to hook-up the signals to the ICOM IC-910 radio. The rest of the radio chain is omitted.[2]

## 3.2 Frequency modulation and RF signal generation

Once the baseband signal is generated (no matter how) there are several possibilities in order to get a radio signal at the required carrier frequency. The available solutions involve either outputting an analogue audio signal or a digital signal to external hardware.

The configuration files of *Dire Wolf* allows to output a Push To Talk (PTT) and a Data Carrier Detect (DCD) signals. The PTT signal is used throughout the chain in order to power up the equipment only when there is a packet being sent while the DCD signal allows to wire an LED to indicate whenever the radio is transmitting.

### 3.2.1 RF generation using analogue audio signal

Once the baseband signal is generated by *Dire Wolf* it is outputted as an audio signal on the device selected in the configuration file. In this solution the signal would be outputted to a physical audio port of the computer (most probably a 3.5mm jack audio plug) while the PTT and DCD signals would be outputted via the data flow control lines of a serial adapter. It would require the choice of a good sound card with a frequency response as flat as possible up to (at least) 5000 Hz and the conception of a small interface device where these signals would be grouped together to finally be inputted in the ICOM-910 as it was with the original solution using a TNC. The reception path is such that the output of the ICOM IC-910 can be connected to an audio input of the computer which is then decoded by *Dire Wolf*. The figure 3.1 displays the resulting radio link.

### 3.2.2 RF generation using digital signal

Once the baseband signal has been generated by *Dire Wolf*, it is frequency modulated within a software (*GNURadio* or *PothosWare*) and then sent to an SDR which will bring the baseband signal up to the selected carrier frequency. For the PTT and DCD signals however there is no such solution and a connection like the one presented previously must still be used (see figure 3.2). The RF signal outputted by the SDR is about 10 dBm to 15 dBm and thus need to be pre-amplified before being amplified by the HLV-550 power amplifier. Such hardware also emits non-negligible levels of spurious and harmonic frequencies (see section 4.5). Thus band-pass filters will likely be required somewhere in the transmission path if there are none yet. The reception path is such that the received RF signal can be directly connected to the SDR (an adapter will likely be required). The signal will be received by the SDR software, frequency-demodulated and then passed to *Dire Wolf* as an audio signal.

## 3.3 Synchronisation word

The previous work from François Piron already explored a few solutions. One can either send a custom synchronisation word or look for a pattern formed by the scrambled preamble flags sent by the TNC[1]. It was then decided to choose a pattern in the scrambled preamble flags and choose that pattern near the end of the preamble such that the Phase-Locked Loop (PLL) of the Clock and Data Recovery

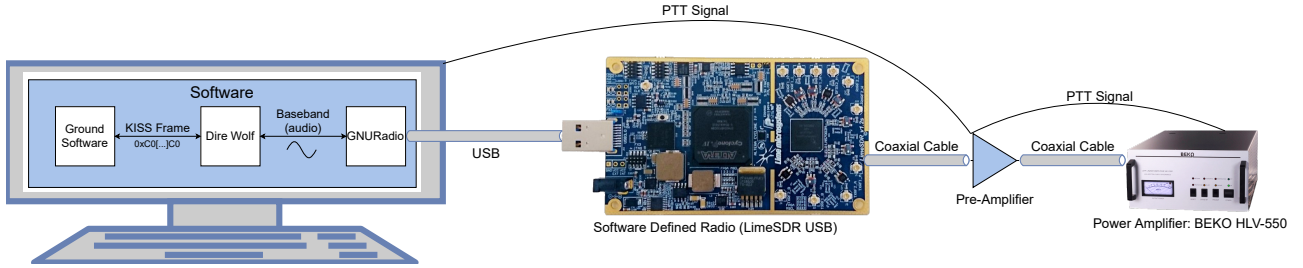


Figure 3.2: Radio chain needed in order to use a digital output from *Dire Wolf* to an SDR. The FM signal is generated on the computer and sent to the SDR via USB. The PTT signal must be connected to the amplifiers. The rest of the radio chain is omitted.[7]

(CDR) module would have the time to correct for any data rate mismatch between the emitter and the receiver. This solution was selected in particular because such a pattern was sent with each packet that the TNC would transmit and because the use of the proprietary TNC would not allow to insert a custom synchronisation word before the packet gets sent.

In the same work it was however found that in *Dire Wolf* the LFSR used for the scrambling was initialised with a value of 0 before a transmission and thus the same procedure could be performed in order to get a synchronisation word configured in the transceiver. As the solutions explored here will use the *Dire Wolf Software TNC* and because this software is open-source the solution of inserting a custom synchronisation word before the packet gets easier and must thus be reconsidered as a possible solution.

# Chapter 4

## Recommended Solution

The recommended solution is also the solution that was used in order to develop the telecommunication module developed in part II of this work. The solution selected is to use an SDR as this avoids the non-linearities that an analogue audio signal passing through a physical sound card would introduce. It is also way more compact and practical when performing tests in the lab as the hardware required is much smaller and can be powered via the USB port of a laptop. The main goal of this solution is to allow to transmit packets to the COMM board which will need to receive and decode the packet and then respond. The data from the response will only be received by the computer and pass the integrity check (the CRC must be valid) from the *Dire Wolf Software TNC*.

### 4.1 Software

The proposed software architecture was designed to be used on a *Linux* operating system and was more specifically designed using *Fedora 32* on an amd64 processor architecture. This might especially be relevant when it comes to compile software.

#### 4.1.1 Data Generation

At first the data needs to be generated. For this the ground software that was developed for the OUFTH1 project will simply be re-used. The source code will be edited such that it will be able to communicate with the *Dire Wolf Software TNC* via UDP instead of communicating with the old TNC via serial. The resulting source code is available on GitLab.<sup>1</sup>

#### 4.1.2 Software TNC

In order to get the signal out of *Dire Wolf*, the only supported way is through an audio device (it allows to use an UDP stream but only to receive a signal, not to transmit it). Using an internal software loop back it is thus possible to connect the sound output/input of a software (*Dire Wolf*) to the sound input/output of another software (*GNURadio*). The internal software loop back, being software, allows for a perfect signal transmission without distortion unlike what would happen with a sound card. The loop back can be added to the system by installing the Alsa package on the system and loading the `snd_aloop` kernel module using the command

```
1 sudo modprobe -a snd_aloop
```

. Once *Dire Wolf* is up and running a configuration file must be written to have the correct set of parameters corresponding to the previous setup using the old hardware TNC. From the source-code of the OUFTH1 ground software the following code can be extracted[8]:

```
1 public final void ConfigureInPacketRadio() throws DriversException
2 {
3     if(!isConnected())
4     {
```

---

<sup>1</sup>GitLab link: <https://gitlab.uliege.be/mdiepart/oufth1-gnd>

```

5         throw new TNCException("TNC.configureInPacketRadio: La
           connexion vers le TNC n'est pas etablie!");
6     }
7
8     // Configuration du TNC
9     sendToTNC("@D1");    // FULL duplex(1)/Half Duplex(0)
10    sendToTNC("@F1");    // Don't send flags during pause
11    sendToTNC("%B9600"); // 9600 baud FSK
12    //sendToTNC("%B1200");
13    sendToTNC("X1");      // PTT normal operations
14    sendToTNC("%X400");  // Set TNC Output level to 400mV
15
16    // Divers
17    sendToTNC("T100");    // Tx Delay (x10ms)
18    sendToTNC("P128");    // Persistence
19    sendToTNC("W20");     // Slot Time (ms)
20    sendToTNC("O2");      // Max Frames
21    sendToTNC("RO");      // Digipeating OFF
22
23    // Desable Call check
24    sendToTNC("@V0");     // Don't check CallSign
25
26    // KISS Mode ON
27    sendToTNC("@K");      // Set KISS mode ON
28    //sendToTNC("S2");
29 }

```

---

Listing 4.1: Sample of code extracted from the ground software of the OUFTI1 mission.

It mainly sends to the TNC the commands needed to switch it to KISS mode and set a few delays[9]. Here is the list of most important commands with a short explanation.

- `sendToTNC("@D1")` sets the TNC in full-duplex mode (it can transmit and receive simultaneously)
- `sendToTNC("%B9600")` configures the TNC for 9600 bauds, 2FSK configuration. This will automatically select the G3RUH modem.
- `sendToTNC("T100")` sets the delay between when the TNC powers on (and powers on the rest of the transmission chain with the PTT signal) and when the TNC actually sends the data it have to send. Those units represents 10 ms each.
- `sendToTNC("P128")` and `sendToTNC("W20")` are used for the Carrier Sense Multiple Access (CSMA) operation and determine the behaviour of the device before each transmission in order to avoid any collision.
- `sendToTNC("@K")` switches the TNC to KISS mode.

All these parameters have an equivalent in Dire Wolf and those are listed below.

1. SLOTTIME 20 is the equivalent of `sendToTNC("W20")`
2. PERSIST 128 is the equivalent of `sendToTNC("P128")`
3. TXDELAY 100 is the equivalent of `sendToTNC("T100")`
4. FULLDUP ON is the equivalent of `sendToTNC("@D1")`

A few instructions sent to the TNC however have no equivalent in Dire Wolf. This is most probably because some of the configurations that get sent to the TNC have no influence on the KISS mode used. It is also to be noted that the command in the line `sendToTNC("@F1")` despite having a comment saying it does not send flags during pause actually configures the TNC to send flags during pauses.

The full configuration file passed as an argument while starting Dire Wolf is available in appendix B. The sampling frequency chosen is 96 000 Hz because it seems to be the best sampling rate given a data rate of 9600 Bd[6]. It could be set higher, but that would only increase the Central Processing Unit (CPU) load with little improvements to the quality of the transmission/reception. *Dire Wolf* can then be started with the following command:

---

```

1      # a=60 => Audio statistics every 60 seconds
2      # r=96000 => 96kHz audio sampling rate
3      direwolf -a 60 -r 96000 -c ./direwolf.conf

```

---

Listing 4.2: Command used to start *Dire Wolf* with the selected configuration

### 4.1.3 Frequency (de-)modulation

**Transmission** The frequency modulation is performed in *GNURadio*. The signal enters the flow-graph using a standard audio input in which the audio loop back device must be selected. The signal then gets frequency modulated with a **Frequency Mod** block. It requires a sensitivity as argument. The value needs to be set according to the formula

$$k = 2\pi \times \frac{f_{dev}}{f_s}$$

where  $f_{dev}$  is the frequency deviation (4800 Hz) and  $f_s$  is the sampling frequency (96 kHz). Which gives a value of  $k = 0.314$ . The signal then gets sent to the SDR using the appropriate block.

**Reception** The signal first have to be received from the SDR at the correct frequency and is then squelched. That means that when the SDR does not receive a signal strong enough (in this case a strength in *GNURadio*  $< -50$  dBm) the signal is muted (set to 0). After that, the signal gets low-pass filtered to remove the out-of-band noise (the bandwidth of the SDR is much higher than the bandwidth of our signal). The signal is then frequency demodulated using a **WBFM Receiver** block.

Finally the signal is pre-emphasized. While usually a signal is pre-emphasized at the transmitter side it is needed here here after the **WBFM Receiver** because it automatically performs a de-emphasis (with a value  $\tau = 75 \times 10^{-6}$ ). This de-emphasis can not be disabled. The only way to compensate for this is to add a pre-emphasis block with  $\tau = 75 \times 10^{-6}$  after the receiver. This was discovered by looking at the source code of *GNURadio*. The file `wfm_rcv.py` (line 64, listing 4.3) shows that a de-emphasis block is instantiated (with default values) while the file `analog_fm_deemph.block.yml` (listing 4.4) shows that the default values for the de-emphasis block is  $\tau = 75 \times 10^{-6}$ .

The signal is then finally given to *Dire Wolf* using the audio loop-back device. The resulting flow graph is shown in figure 4.1.

## 4.2 Hardware

The software defined radio that will be used is the LimeSDR USB. No real selection was done, it just happens to be a device that was already available for this work. In order to interface it with *GNURadio* the OOT module `gr-limesdr` was installed. *GNURadio* was then able to interface to the SDR device connected to the computer via USB 3. In this configuration, the PTT and DCD signals are not used. That means that when no packet is being transmitted, the SDR transmits a tone at the carrier frequency (435.015 MHz).

A filter was also designed to remove any harmonics produced by the SDR. As the signal generated is very narrow-band the best suited filter is a Chebyshev filter. This type of filter typically shows steeper slope after the cutoff frequency when the pass-band ripple is higher. The filter was thus designed using the online tool *LC Filter Design Tool* (available online at <https://rf-tools.com/lc-filter/>) with

---

```

61 self.fm_demod = analog.quadrature_demod_cf(fm_demod_gain)
62
63 # input: float; output: float
64 self.deemph = fm_deemph(audio_rate)
65
66 # compute FIR filter taps for audio filter
67 width_of_transition_band = audio_rate / 32
68 audio_coeffs = filter.firdes.low_pass(1.0,          # gain
69                                     quad_rate,      # sampling rate
70                                     audio_rate / 2 -
71                                     width_of_transition_band,
72                                     width_of_transition_band,
73                                     filter.firdes.WIN_HAMMING)
74 # input: float; output: float
75 self.audio_filter = filter.fir_filter_fff(audio_decimation,
76                                           audio_coeffs)
77
78 self.connect (self, self.fm_demod, self.audio_filter, self.deemph,
79               self)

```

---

Listing 4.3: Code sample from file `wfm_rcv.py`

---

```

1 id: analog_fm_deemph
2 label: FM Deemphasis
3 flags: [ python ]
4
5 parameters:
6 - id: samp_rate
7   label: Sample Rate
8   dtype: real
9 - id: tau
10  label: Tau
11  dtype: real
12  default: 75e-6

```

---

Listing 4.4: Code sample from file `analog_fm_deemph.block.yml`

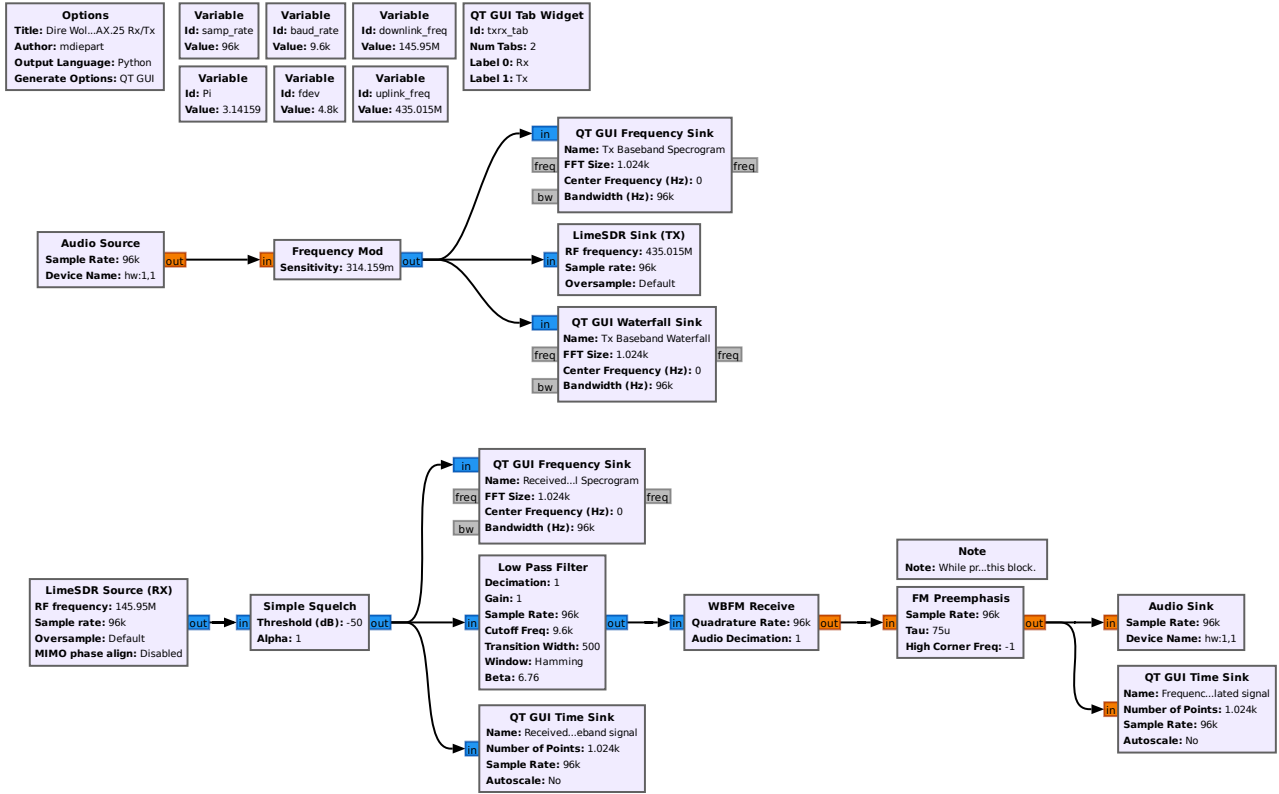


Figure 4.1: *GNURadio* flow graph used to interface Dire Wolf with the software defined radio. Some other blocks have been added to display some signals and spectrums while the flow graph is running.

a pass-band ripple of 1 dB and a cutoff frequency of 643 MHz such that the signal would be at the top of the first ripple (see figure 4.2 for an LTSpice simulation of the filter). This tends to reduce the attenuation at the selected frequency and makes the design less sensitive to slight deviation of the components from their nominal values.

### 4.3 Synchronisation Word

The question of the synchronisation word was not settled as quickly as expected. At first it was assumed that *Dire Wolf* did reset the LFSR used for the data scrambling before each packet transmission. However, a few tests showed that two successive data transmissions did not create identical outputs for the preamble part of the transmission (the flags sent during the Tx Delay of 1 second). Figure 4.3 shows the beginning of three successive data transmissions captured by recording the audio from the loop-back device with *Audacity*. After looking at the source code of Dire Wolf it was quickly found that the LFSR was only being initialised at 0 and was not reset before each transmission. The source code of the software was thus modified in order to reset the LFSR as well as the state of the NRZI encoder after each data transmission.

However, even when resetting the LFSR and NRZI encoder after each transmission the audio output would still be inverted from times to times between successive transmissions. This would highly degrade the SWD functionality as only the packets with the right polarity would be detected by the transceiver even though they contain the same data. By looking further in the source code, it was found that the culprit was the tone generation which would sometimes be left in a state where it is phase-shifted by 180° due to the last transmission (see figure 4.4). Resetting the tone phase between each transmission finally showed the expected behaviour. That is, two successive and identical transmissions produce the exact same audio output. The figure 4.5 shows five successive transmissions which produced identical outputs.

Despite the scrambling producing the same output for the same data it was decided to perform another modification. It was indeed decided to use a synchronisation word different from the data produced by the scrambled flags. The synchronisation word used is 0x00 or, once NRZI encoded, 0x55



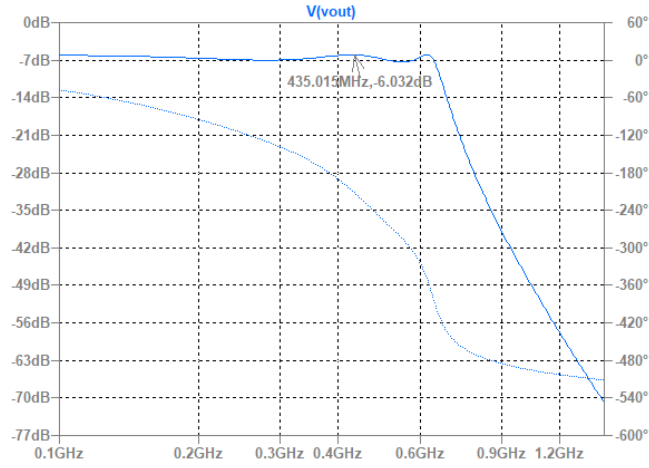


Figure 4.2: Bode diagram of the filter used at the output of the SDR. The cutoff frequency is 634 MHz and the frequency of the signal at 435.015 MHz is located at the top of the first ripple and is not attenuated. Due to the way the filter is simulated in LTSpice it shows a 6 dB attenuation in the pass-band region instead of 0 dB.

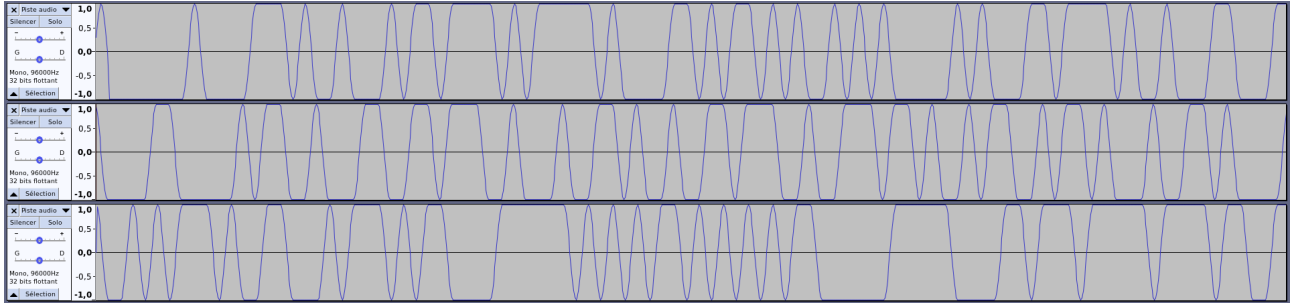


Figure 4.3: Audio output of three successive transmissions performed just after starting *Dire Wolf*. The beginning of the three transmissions look different even though the data sent up to there is the same. This means that the LFSR is not reset between each transmission.

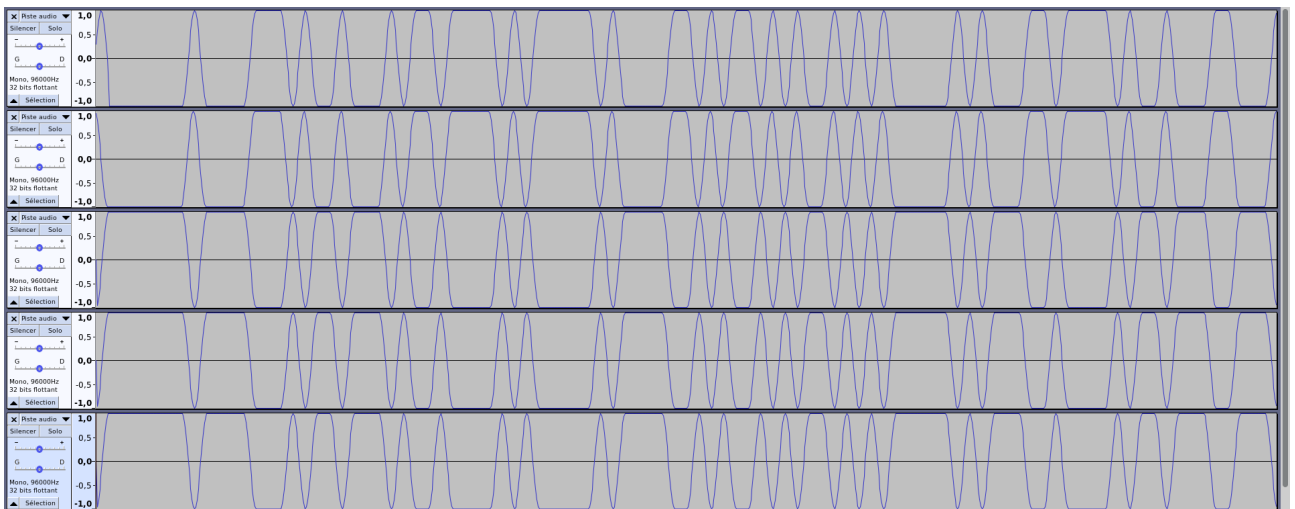


Figure 4.4: Audio output of five successive transmissions using *Dire Wolf* (modified to reset the LFSR and NRZI encoder after each transmission). The transmissions look identical except for their polarity which is not always the same across transmissions. This is because the phase of the audio tone generator is not reset between each transmission.

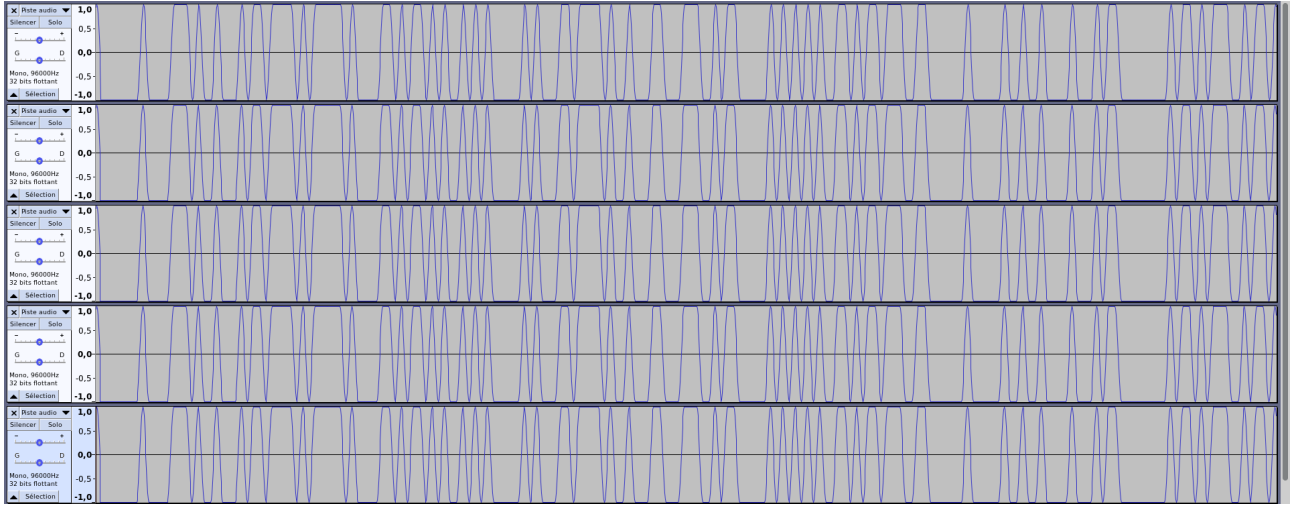


Figure 4.5: Audio output of five successive transmissions using *Dire Wolf* (modified to reset the LFSR, NRZI encoder and audio phase after each transmission). The transmissions look identical.

Tx Delay	Synchronisation Word	Flags	Frame	Flag
01111110 scrambled and repeated for Tx Delay seconds	01010101 (0x00 NRZI encoded) not scrambled repeated 5 times	01111110 scrambled repeated 5 times	AX.25 Frame	01111110 repeated for Tx Tail seconds

Figure 4.6: Proposed frames to be transmitted by the ground station to the nanosatellite. A synchronisation word have been added.

which in binary will produce the output 0b01010101. This is the highest frequency the baseband signal is able to represent. There are several advantages to doing so.

- Even if for some reasons the data was to have an inverted polarity the synchronisation word would still be detected because it would be equivalent to a one-bit shift in the data and if enough synchronisation words are sent in a row this will still trigger the SWD.
- This pattern represents the most signal transitions possible using a data rate of 9600 Bd. It will allow for the CDR PLL of the transceiver to be as accurate as possible.
- The synchronisation word generated by the scrambled flags is inherently linked to the number of flags that are sent in the Tx Delay encoded in the TNC and shortening this delay might remove the expected synchronisation word from the transmission and break the communication link.

The recommended transmission order would be as follow (see figure 4.6).

1. Send the HDLC flag 0x7E through the scrambler (as usual) for the set Tx Delay time.
2. Send the unscrambled (only NRZI encoded) 0x00 flag, 5 times for a total of 40 bits.
3. Send at least 4 successive HDLC flags. At least 4 flags are needed in order to have the 17 bits required to synchronise the emitter LFSR with the receiver LFSR.

Even though it is not really a concern in our case it can be noted that this mode of operation stays compatible with a standard AX.25 receiver because the 5 successive 0x00 flags are less than the minimum size of an AX.25 frame. For any AX.25 receiver that would receive this message it would look like two successive AX.25 frames being sent in a row with the first one being silently discarded because of its invalid length.

The source code of Dire Wolf have been modified to send the synchronisation word and is available on <https://github.com/mdiepart/direwolf><sup>2</sup>.

<sup>2</sup>Github link: <https://github.com/mdiepart/direwolf>

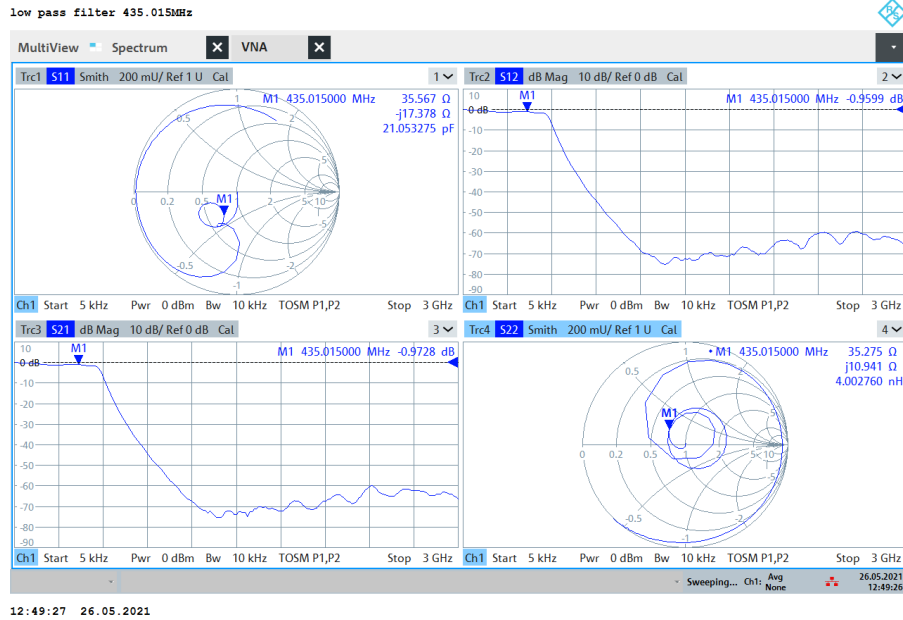


Figure 4.7: Impedance measurement of the filter. The input and output impedances of the filter are acceptable. The insertion losses do look like the simulation from figure 4.2.

## 4.4 Summary

The radio chain have been modified such that the ground software now sends its KISS frames to *Dire Wolf*, a software TNC, instead of the old hardware TNC. The output produced by *Dire Wolf* is sent to *GNURadio*, an SDR software via an internal software audio loop-back device. *GNURadio* then sends the frequency modulated data to a software defined radio. The *Dire Wolf* software was however modified such that it would reset its state fully between two transmissions and send a synchronisation word after the set Tx Delay.

For the reception, it is basically the same but in reverse. The signal sent by the satellite to the ground station is fed to the SDR. The signal is received in *GNURadio* which frequency demodulates it. The demodulated signal is then sent to *Dire Wolf* which performs the unscrambling and decoding operations.

## 4.5 Results

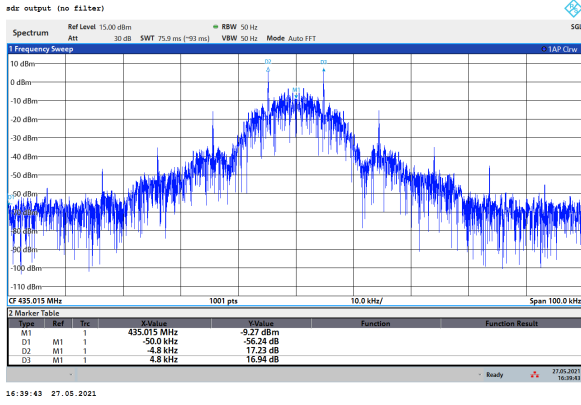
The resulting system successfully sends AX.25 frames at the frequency of 435.015 MHz from the ground software used for the OUFTI1 mission. But in order to go further in the analysis of the results it is appropriate to have a look at the spectrum of the signal emitted by the SDR.

The following measurements are performed using the ZNL3 Vector Network Analyser from the Microsys lab. It is able to perform impedance measurements and spectrum analysis from 5 kHz up to 3 GHz.

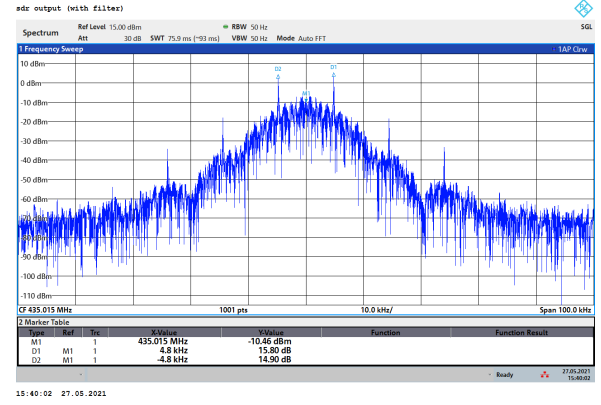
The first measurements to be performed concerns the filter used to attenuate the harmonics of the signal. The figure 4.7 shows the impedance measurement of the filter. The top-left graph shows the impedance seen from the input of the filter ( $S_{11}$ ), the bottom-right graph shows the impedance seen from the output of the filter ( $S_{22}$ ). Ideally both should be  $50\ \Omega$ . The top-right ( $S_{12}$ ) and bottom-left ( $S_{21}$ ) graphs are the same (because this is a passive device) and shows the fraction of the power that passes from one end to the other (the insertion losses) and should ideally be the same as the figure 4.2.

It can be seen that both the input and the output impedance are not exactly  $50\ \Omega$  but their value is close enough to not introduce significant reflection losses due to an impedance mismatch (the return losses are both below  $-11\ \text{dB}$  which means that more than 98% of the output power gets transmitted). The insertion losses are what was expected from the *LTSpice* simulation.

The second measurement shows the spectrum of the signal produced by the SDR centred around the frequency 435.015 MHz with a bandwidth of 100 kHz (see figure 4.8). As can be seen from the



(a) SDR output spectrum without output filter



(b) SDR output spectrum with output filter

Figure 4.8: Comparison of the SDR spectrum with and without the filter. A slight attenuation is noticeable with the filter. The shape stays the same.

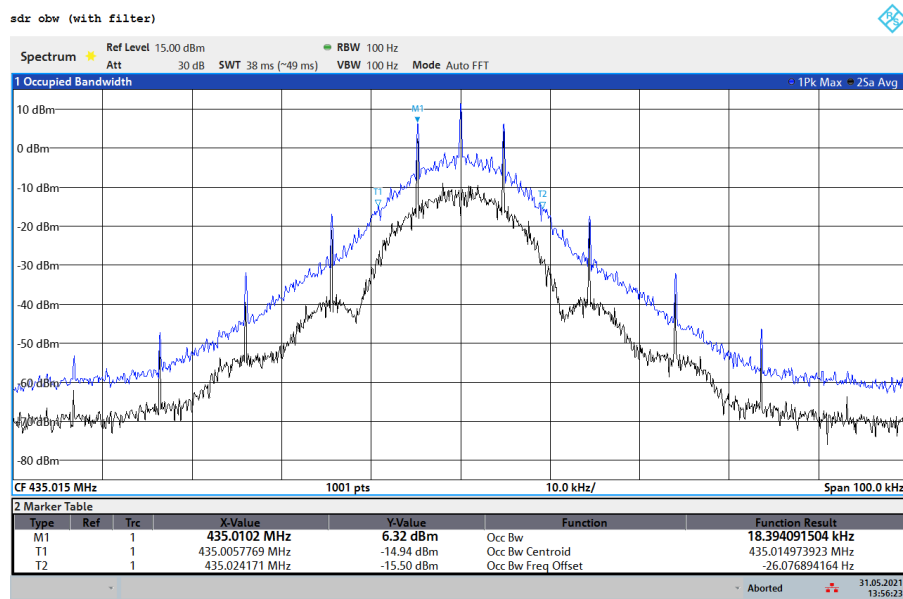
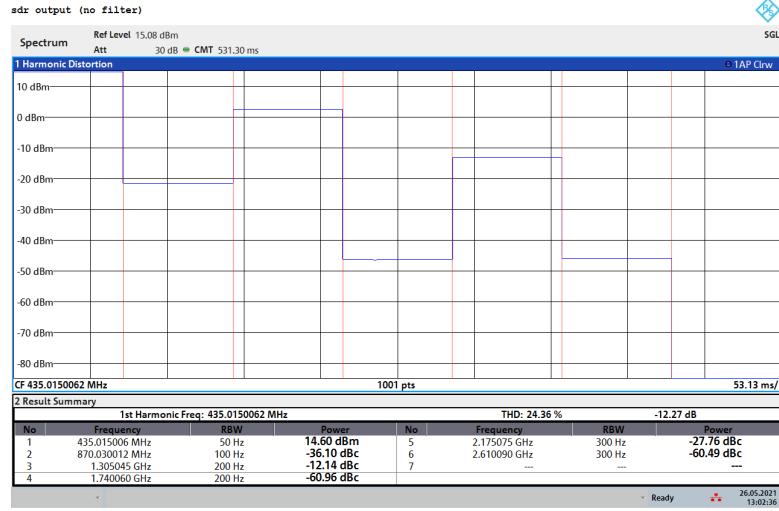


Figure 4.9: Occupied bandwidth measurement on the signal transmitted by the SDR. The measured occupied bandwidth is 18.4 kHz.

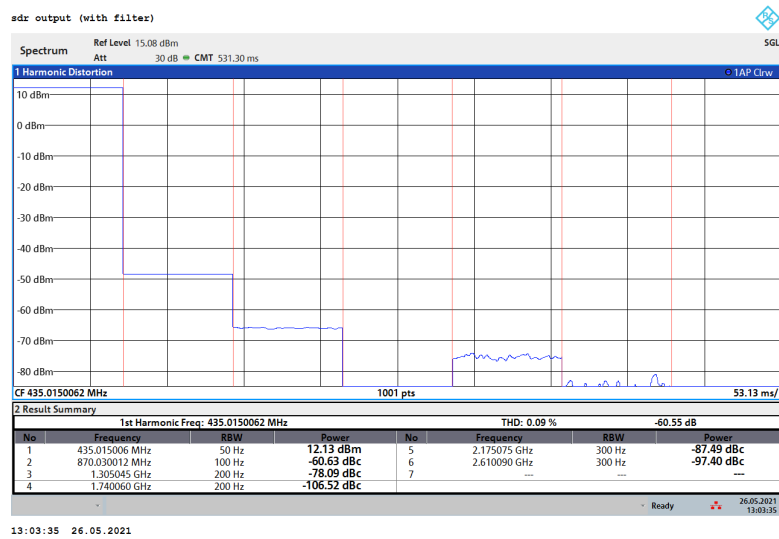
figures 4.3, 4.4 and 4.5, the baseband signal is not a pure square-wave signal which would have corresponded to a 2FSK modulation. The signal gets filtered and the overall shape of the graph in figure 4.8a seems to indicate that it is not a Gaussian filtering not a raised-cosine filtering. However, two spikes can be seen at  $\pm f_{dev}$  or  $\pm 4800$  Hz from the centre frequency. The filter have a slight attenuation effect on the power measured but the overall shape of the spectrum stays the same meaning that no non-linearities are introduced.

As the modulation scheme is a 2FSK modulation with some kind of pre-filtering it is interesting to perform a measurement of the Occupied BandWidth (OBW) (the bandwidth containing 99% of the signal's power). The spectrum analyser was thus setup in OBW measurement mode. However, when the measurement was performed on the raw input signal or an averaged input signal the measurement was varying a lot (between 16 kHz and 18 kHz). It was thus decided to perform the measurement on the maximum value of the signal (see figure 4.9). The measured occupied bandwidth is 18.4 kHz. This value will be exploited when designing the module in part II.

The last measurement performed shows the harmonic distortions measurements both with and without the filter at the output of the SDR. Those were performed by sending a single tone at 435.015 MHz and recording the levels at the frequencies of the harmonics up to 3 GHz. The harmonic distortions without the filter are very high with the most prominent being unexpectedly the third



(a) Harmonic distortions without the filter. Most noticeably the third harmonic is only 12.14 dB below the fundamental



(b) Harmonic distortions with the filter. The harmonics are now barely noticeable.

Figure 4.10: Comparison of the harmonic distortions with and without the filter.

harmonic only 12.14 dB below the carrier frequency (see figure 4.10a). Once the filter is added however the most powerful harmonic becomes the second harmonic (which is less attenuated than the third harmonic) with a power of  $-60$  dBc.

## 4.6 Conclusion

A working alternative to the old hardware TNC have been established using modern technologies. It will allow to test the hardware that will be developed in part II of this work. It also is much more flexible to use and can be modified to follow any change the project might be subject to in the future. All that using open-source solutions. The radiated spectrum have been analysed and a filter have been designed and tested to ensure the spectrum radiated will fit the different regulations.

## 4.7 To go further

The main goal of this setup being to perform transmission tests with the hardware that will be developed in the last part of this work the PTT and DCD signals were not used at all. However, in

order to get a fully-working radio chain that is also compliant with the regulations in place but also safe to use for the rest of the hardware making the full radio chain, some form of PTT signal will be required.

The filter designed however is not intended to be used in the ground station and as the regulations imposes a station to limit the power emitted in the spectrum that is not allocated to it, the harmonics produced by the SDR might not comply with the regulation. That means that if the ground station does not already have filters some will need to be added. It will most probably be a band-pass filter centred around the frequency of 435.015 MHz. If one is to be added to the ground station its position (between which elements of the chain it will be placed) must be carefully studied to have a correct signal to noise ratio while still not dissipating too large amounts of power (even though the power available on ground is not limited like on a satellite, filters dissipating a lot of power will likely be more expensive).

For the PTT signal (the DCD signal is not a real requirement, only a nice indicator to have in the station at best) it is required because the SDR constantly emits a tone at the carrier frequency if it is not sending a packet. So if the SDR is constantly plugged to the pre-amplifier and the HLV-550 power amplifier the ground station will constantly emit a tone at 435 MHz. If going on for too long this might damage some equipment (for example the HLV-550 datasheet states that the equipment should not be emitting at full power with a 100% duty cycle for too long[10]). While the user might manually switch the rest of the chain on and off before and after each transmission this is not a realistic option and thus the PTT signal outputted by *Dire Wolf* must be used. The easiest and probably cheapest option is to use a serial device and connect the selected re-purposed flow control output to the PTT inputs of the amplifiers. One could also install *Dire Wolf* and the *GNURadio* flow graph on a nano-computer having General Purpose Input/Outputs (GPIOs) available to the user such as a Raspberry Pi. *Dire Wolf* is able to output the PTT signal on a GPIO. The risk with that solution would be that those kind of computers are often very low power and it must be checked whether their processing capabilities are high enough to have both *Dire Wolf* and *GNURadio* running. Otherwise it would still be relatively inexpensive and the computer running the ground software would only have to connect to *Dire Wolf* via a local network.

# Part II

## Module Design

# Chapter 5

## First Hardware design

This part will go through the realisation of an entire module enabling AX.25 communication for a nanosatellite. Transceivers will be chosen as well as low noise amplifier for the reception and power amplifier for the emission. Separate circuits will be designed to test each component independently and to ensure that they work as expected. Finally a board will be built with all the components to ensure that they work once assembled together. Firmware will also be written to test the transceiver and exchange AX.25 frames with the system designed in part I.

### 5.1 Components choice

#### 5.1.1 Criteria

First and foremost the criteria that will be used to choose the components need to be known.

**Frequencies** Frequencies are one of the first criteria to be considered because it will highly limit the available components. Some like the transceiver might simply not be able to use the required frequencies while some components like the amplifiers might have very poor performances. Some other components might work at the frequencies used but their documentation might lack important data at the frequency at which they would be used, making the design of the circuit more difficult without giving any data against which the design could be checked.

Furthermore, the frequencies considered throughout this work are the exact frequencies that were used for the OUFTI1 mission. Even though it is highly unlikely that those exact same frequencies get attributed to the CubeSat that would use this telecommunication module, the changes that would be required to conform to the exact band allocated are so small that it does not constitute a concern. The frequency bands allocated for radio-amateur satellite telecommunications are 144 MHz to 146 MHz and 435 MHz to 438 MHz[11]. There are other bands that can be attributed to radio-amateur satellites but does not concern this module.

The out-of-band emissions levels are also regulated but will depend on the bandwidth allocated to the spacecraft. As this is not known it is difficult to tell the power below which an out-of-band emission must be. Through this work, those emissions will be reduced as much as possible to increase the chances that the device complies with such regulations.

**Power levels** The power levels of the different signals throughout the board are also of importance. At the reception will be very low-power signals. Caution must be taken not to loose those signals in the noise because of filtering, noisy amplification,...

At the emission side will be much higher power signals (above 30 dB or 1 W). The design must be done carefully such that the impedances are well matched and the Voltage Standing Wave Ratio (VSWR) are kept low enough. Otherwise this might damage some components on the board and render the CubeSat useless. The different losses (due to filters or low components efficiencies) should also be minimized as high losses will induce heating and high temperatures might also lead to components being damaged.



**Linearity** The linearity of the different components chosen is also of prime importance as non-linearities might affect the signal in unrecoverable ways or create unwanted emissions out of the allocated bandwidth possibly affecting other devices around.

**Power consumption** As the module will be aboard a CubeSat the power available is limited as is the available energy. The satellite being (by law) always listening for telecommands from the ground station, the components on the reception side will be always on and always consuming energy. The emission side will only be switched on when the satellite needs to send a telemetry but the higher the energy needed to send a telemetry the fewer it will be able to send.

## Out Of Band Emissions

### 5.1.2 Power Amplifier

The power amplifier is a device that amplifies a signal allowing it to reach higher power levels than other amplifiers (at the cost of increased noise). In order to reduce the footprint of the device and the number of external components and to simplify the design, more integrated solutions will be preferred. Those solutions generally include internal bias to be applied at the input and/or impedance matching for the input and/or output. The device should also be as efficient as possible in order to reduce its energy consumption. As a reminder the frequency at which the power amplifier will work is 145.95 MHz. The typical output power for a nanosatellite is 30 dBm. However, the output will have to go through a power combiner before the antenna so that a beacon can also be added later on and use the same antenna. This will add losses of at least 3 dB as this is the theoretical minimum losses introduced by a power combiner. In practice power combiner have insertion losses of about 0.9 dB which have to be added to the 3 dB of theoretical losses. The output power to aim for is thus above 30 dBm.

Very few devices fit our quite strict requirements and it seems that the best power amplifier available is the RF5110G from Qorvo. Even though the datasheet states that the frequency range for which the device is designed is from 150 MHz to 960 MHz no significant difference is expected between 145.95 MHz and 150 MHz[12]. Its output power is given to be at most 32 dBm for a gain of 32 dB. However, a power amplifier should not be used beyond its linear region. The limit of the linear region is considered to be the point at which the gain of the amplifier drops by 1 dB. For the RF5110G at 150 MHz this power level is 30.5 dB(see figure 5.1). At this output power the gain is 34 dB which is 1 dB below the gain at lower power which is 35 dB. That means that the required input power is about -3.5 dBm.

### 5.1.3 Low Noise Amplifier

For the low noise amplifier the frequency used is 435.015 MHz and a wider range of solutions is available. The main characteristics that will be taken into account are the gain provided, the noise figure as well as the energy consumption of the device. Once again, devices with integrated bias and matching will be preferred. Among the devices available a few were pre-selected and some features were listed. The table contains for each component

- the frequency range at which the device can operate,
- the frequencies at which the device was tested,
- the supply voltage,
- the supply current,
- the gain,
- the noise figure,
- the Output Third Order Intercept Point (OIP3),

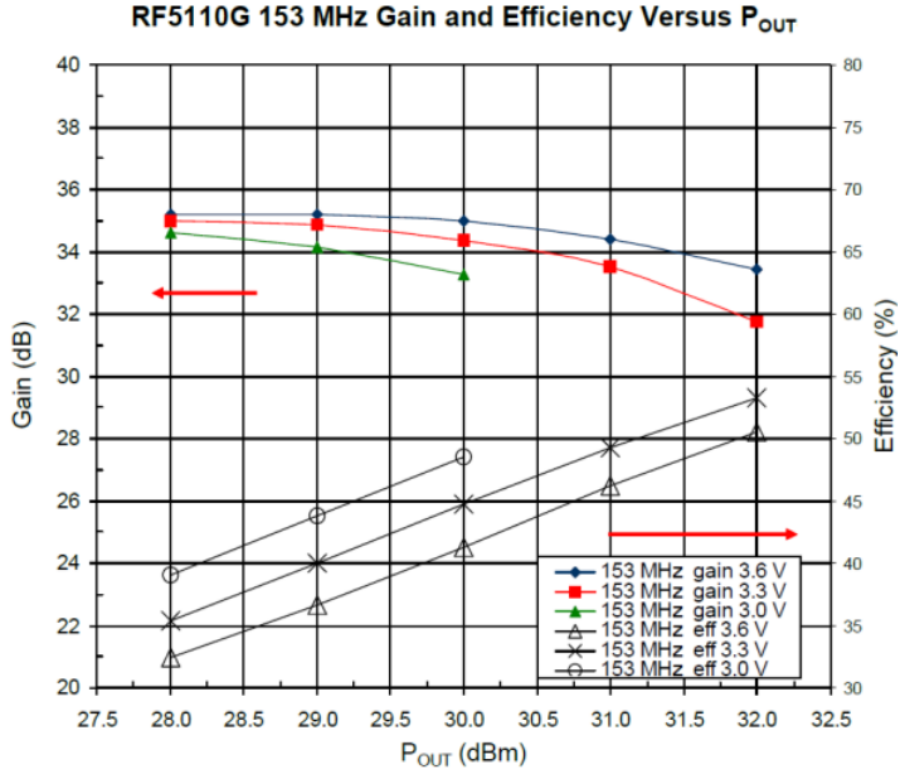


Figure 5.1: Output power and gain of the Qorvo RF5110G. The device will be supplied with 3.3 V.

- the Output 1dB Compression Point ( $P_{1dB}$ ),
- the maximum input power,
- the presence or not of an internal bias.

The frequency range must include the frequency of 435.015 MHz and the test frequency must be as close as possible to the frequency used. The supply voltage and supply current must give a low power consumption. The gain should be as large as possible and the noise figure should be as small as possible. The OIP3 and  $P_{1dB}$  are used as a measure of the linearity and should be as large as possible. The maximum input power is used as a measure of the robustness of the device and should be as large as possible, however, in this case, it will not be a major concern. The device selected should include an internal bias to facilitate the design. The table containing a summary of those values for all the pre-selected components is available in appendix C.

The component that will be used for the module is the BGB741L7ESD from Infineon. This amplifier contains an internal bias and an internal matching to  $50\Omega$  on both the input and the output. This is also a broadband amplifier meaning that a wide range of frequencies will be amplified. However, there is already some filtering which is done at the input of the system via the antenna's bandwidth and the selectivity of the internal filter of the transceiver should, in the end, make it so that only the frequency used will be demodulated.

#### 5.1.4 Transceiver

Because the uplink and downlink frequencies are different and because the CubeSat needs to be always listening for telecommands two transceiver will be needed. One for the reception (RX) and one for the transmission (TX). Using the same reference for both transceivers will make the development process much easier by allowing us to re-use schematics and code whenever possible. By looking through the available parts from different manufacturers it seems that there are few components that are able to perform arbitrary data transfers using 2FSK modulation at the frequencies and data rates used here. The main choices available were the ADF7021 from Analog Devices (used in the OUFTI1 mission) and the CC1120 from Texas Instrument. One can immediately notice that the Texas Instrument's

chip can be used with frequencies down to 136.57 MHz but that the 136.7 MHz to 160 MHz is lacking many information which can only be obtained by contacting the manufacturer. Even though using the designs given for 169 MHz would most probably work there are too many unknowns especially when compared to the ADF7021 whose data are fully available for the frequencies that will be used. The ADF7021 having been used in the past for OUFTI1 and OUFTI2 related works and it was decided that this would be the best choice as there already are several works already available. Several versions are available and the ADF7021-N will be used here as it is better suited to narrow-band applications[13].

## 5.2 Schematic design

Before designing a circuit containing all the components, each one must be tested independently. This will allow to measure their performances, power consumption, impedances,... much more easily than if they were all on the same circuit. The schematics representing the circuits designed in this section are all available in appendix D.

### 5.2.1 Power Amplifier

The design that will be used to test the RF5110G power amplifier is the design recommended by the manufacturer in the datasheet[12]. The input of the power amplifier is already matched to  $50\ \Omega$  and the input mainly consists in a DC blocking capacitor and a resistor. The output consists in a series of components whose goal is to match the  $50\ \Omega$  output load to the ideal impedance to be seen by the output of the amplifier. There are some inductors and some decoupling capacitors around the chip for the power supply. The schematic written to test the PA is available in appendix D.1.

### 5.2.2 Low Noise Amplifier

The design that will be used to test the BGB741L7ESD is the recommended design from the design guide of the component[14]. This design is made to work at the frequencies used by navigation systems (i.e. 1 GHz to 1.6 GHz approximately). However, this is a broadband amplifier whose input and output are matched to  $50\ \Omega$  and the figures shown by the manufacturer show that the performances at 435 MHz can be expected to be as good (see figure 5.2). The current setting resistor was chosen to have a current of 10 mA. The input mainly consists in an LC filter whose main goal is to apply the bias from the internal bias to the input of the amplifier via the inductor and block the DC component with the capacitor. The output also has an LC filter allowing to power the amplifier via the inductor and block the DC component with the capacitor. Some decoupling capacitors are also placed at the power input. The schematic written to test the LNA is available in appendix D.2.

### 5.2.3 Transceiver

The design of the ADF7021-N transceiver is based on one of the evaluation boards designed by Analog Devices. However, there is no specific design for the frequencies used. Moreover the way the devices will be used (that is with one being only in reception and one being only in emission) allows for design simplifications. Among the evaluation boards whose design is available the one that will be used is the evaluation board having separate input and output impedance matching. Indeed the device is a half-duplex device and when it is used as such with the emission and reception frequencies being the same it is possible to design a single impedance matching network who matches the  $50\ \Omega$  impedance of the antenna to the correct impedance for the RF input as well as the RF output. This however have two undesirable side effects. The first is that the matching network would be much more complicated to design and the impedance matching might be less accurate. The second is the fact that the RF input, RF output and antenna are all connected forces the use of an RF switch. Even though the device includes one this still induce a loss of 1 dB to 2 dB[15].

The device also requires a frequency reference to be provided. As the nanosatellite will be operating over a wide range of frequencies the source that will be used is a Temperature Compensated Crystal Oscillator (TCXO). Given the devices available at the different electronic components suppliers it was decided to opt for a crystal with an output of  $0.8V_{pp}$  clipped sinewave. It must thus be coupled to the

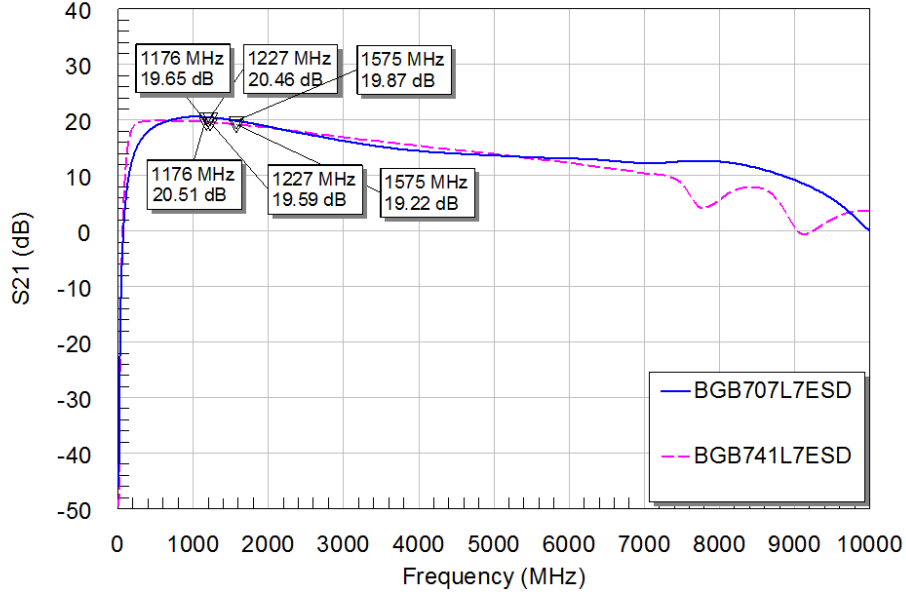


Figure 5.2:  $S_{21}$  (gain) parameter curve given by Infineon and measured using their recommended design. The curve to look at is the pink dashed curve. The gain at 435 MHz should be a little bit below 20 dB.

OSC1 input of the chip via a 22 pF capacitor. The datasheet also advises on using a crystal frequency such that the fractional part of the on-chip fractional-N PLL is close to the middle of the range as this helps in reducing the level of the spurious emissions (see section 7.1.3)[13]. It was found that a crystal at a frequency of 19.2 MHz would satisfy that criterion both for the emitter and the receiver.

Several main parts requires a specific design:

1. the voltage controlled oscillator of the TX transceiver,
2. the voltage controlled oscillator of the RX transceiver,
3. the output impedance matching for the TX transceiver,
4. the input impedance matching for the RX transceiver,
5. the PLL filter for the TX transceiver,
6. the PLL filter for the RX transceiver.

Each part will require a specific tool in order to dimension the required components. The schematics written to test the transceivers are available in appendix D.4 for the RX transceiver and appendix D.3 for the TX transceiver. The appendix D.5 shows which signals are available on the main connector.

**Voltage Controlled Oscillator (VCO) of the transmitter** The TX transceiver operating at a frequency of 145.95 MHz, the design have to use an external VCO inductor between the pins L1 and L2. Using the *ADIsimSDR Design Studio* software gave an inductor of 19.8 nH for the inductor to which the inductance due to the copper traces of the PCB must be subtracted. Anticipating a bit the layout design section it needs to be reduced by  $2 \times 1.65 \text{ mm} \times 0.57 \text{ nH mm}^{-1} = 1.88 \text{ nH}$  if considering an inductance of 0.57 nH per mm of trace[13]. This gives a final inductor value of 18 nH (using the standard inductor values).

**VCO of the emitter** The receiver can, by operating at 435.015 MHz, use either an internal inductor or an external inductor for its VCO. Using an external inductor consumes slightly less power and this was the selected option. *ADIsimSDR Design Studio* gave an ideal inductor value of 7.65 nH. Once again anticipating the layout design section that value will need to be reduced by the value of the inductance of the traces connecting the inductor to the chip on the PCB. The traces represent an

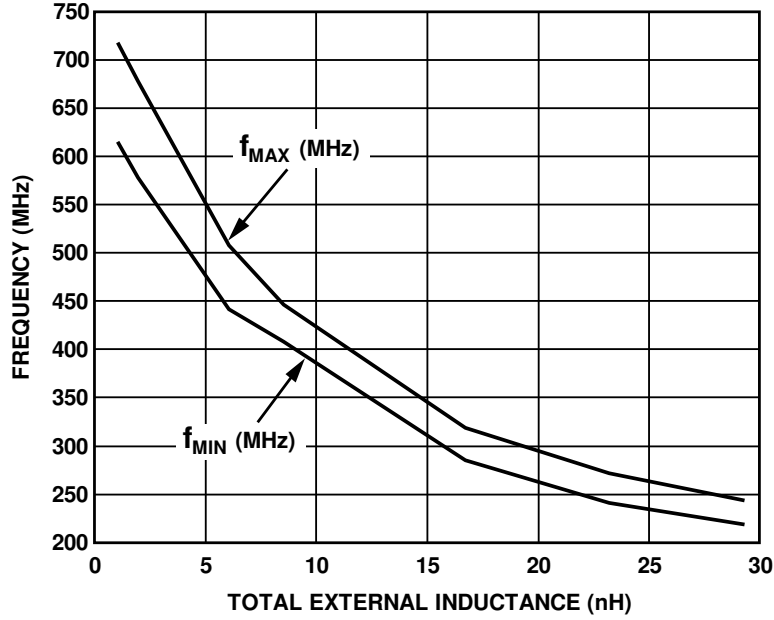


Figure 5.3: VCO external inductance value vs. RF output frequency.

inductance of  $2 \times 1.4 \text{ mm} \times 0.57 \text{ nH mm}^{-1} = 1.6 \text{ nH}$ . By selecting a standard inductor value and choosing among the stocks available an inductor of  $6.8 \text{ nH}$  was selected. This stays within the acceptable range of inductor values for a frequency of  $435 \text{ MHz}$  (see figure 5.3).

**Output impedance matching** The output impedance matching consists in a few components placed at the RF output of the chip whose goal is to match the impedance of the output port as close as possible to the standard  $50 \Omega$  of the rest of the system. This is needed because having an impedance mismatch will cause a fraction of the output power to be reflected back to the source and will thus reduce the output power. This reflected power create standing waves along the transmission line (the copper trace connecting the output to the load). If a standing wave appears and the ratio of the voltage between the nodes and the anti-nodes is too large (VSWR too high) it might damage either the source or the load.

The output impedance matching is quite easy to design. The impedance of the output RF port is available in an application note written by Analog Devices [15]. The section containing the impedance needed is the section "PA port impedance in Tx mode". It is given in the form of an optimal load impedance at the specified frequency. The lowest frequency for which an optimal impedance is given is  $160 \text{ MHz}$  but one can see that the impedance does not vary much below  $200 \text{ MHz}$  meaning that the matching network can be expected to work equally well at  $145 \text{ MHz}$ . The value of the optimal impedance is  $177.28 + i48.41 \Omega$ . To convert the optimal load impedance to an effective impedance, the conjugate value must be taken. The matching network must thus adapt an impedance of  $177.28 - i48.41 \Omega$  to an impedance of  $50 \Omega$ .

This network will be designed using the recommendations of Analog Devices in the device's datasheet. An inductor is used to power the PA and a capacitor is used to block DC current. Once the components to use are known their value must be set such that the impedance is properly matched. This will be done on a Smith chart on the [https://www.will-kelsey.com/smith\\_chart/](https://www.will-kelsey.com/smith_chart/) website. Once accounting for the standard components values it seems that an inductor of  $100 \text{ nH}$  and a capacitor of  $13 \text{ pF}$  are needed to match the source impedance of  $177.28 - i48.41 \Omega$  to an ideal load impedance of  $50 \Omega$  (see figure 5.4).

**Input impedance matching** The input impedance matching is more tricky to design because while matching the impedance of the LNA input to the ideal  $50 \Omega$  of the source, the unbalanced signal from the external LNA have to be converted to a balanced signal for the ADF7021-N's LNA differential input.

The application note referencing the ports impedance of the ADF7021-N gives the impedance needed

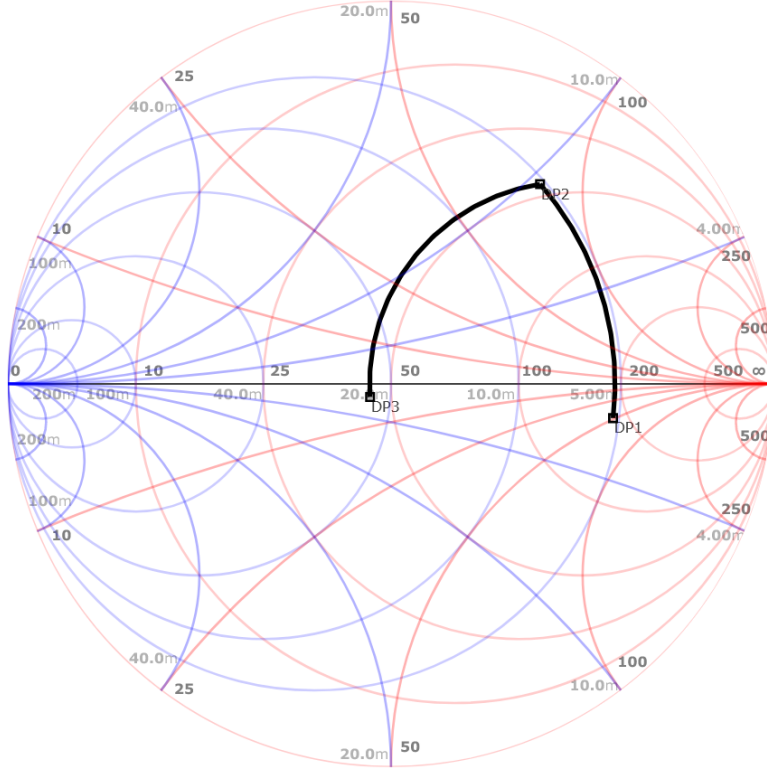


Figure 5.4: Output matching Smith chart. The source impedance is the point DP1. The point DP2 represents the source impedance seen by the load after the inductor and before the capacitor. The point DP3 represents the source impedance seen by the load after the full matching network.

in the "LNA port impedance in Rx Mode" section. It is given as a lumped-elements model with a parallel RC block between each input pin and the ground and a parallel RL block between the two input pins (see figure 5.5). The document gives the values at 430 MHz and 440 MHz so a simple interpolation will give the values at the desired frequency. At 435 MHz  $C_A = C_C = 2.225$  pF,  $R_A = R_C = 331.5 \Omega$ ,  $L_B = 2.85 \mu\text{H}$ ,  $R_B = 9.55 \text{ k}\Omega$ .

Before performing any impedance matching the given parameters must first be transformed to a  $R + iL\Omega$  impedance between the two input pins. The circuit is equivalent to a parallel LC block in parallel to two parallel RC blocks in series with the given values. This gives

$$\begin{aligned}
 Z_{diff} &= Z_B // (Z_A + Z_C) \\
 &= Z_B // (2 \times Z_A) \\
 &= \frac{1}{\frac{1}{Z_B} + \frac{1}{2 \times Z_A}} \\
 &= \frac{1}{\frac{1}{3820 + 4682i\Omega} + \frac{1}{2 \times 65.295 - 131.84i\Omega}} \\
 &= 145.15 - 262.54i\Omega
 \end{aligned}$$

In order to design the balun that will convert the unbalanced signal to a balanced signal, the software *AppCAD* will be used as it is free to use and have a balun design tool. Entering the value of  $Z_{diff}$  calculated previously gives a design with two 2.2 pF capacitors, two 64.4 nH inductors and one 125 nH inductor between the two inputs (see figure 5.6). However, in order to block any DC current between the inputs of the ADF7021-N's LNA (which is internally biased) and the ground or the signal source two 1 nF capacitors will be added. One is placed right before the balun. The second one is placed between the 64.4 nF capacitor and the ground (on the lower leg of the balun as in figure 5.6). Once rounded to standard values the 125.4 nH inductor is replaced by a 120 nH inductor while the two 64.4 nF inductors will be replaced by two 68 nF inductors. The two 2.1 pF capacitors are replaced by two 2.2 pF capacitors.

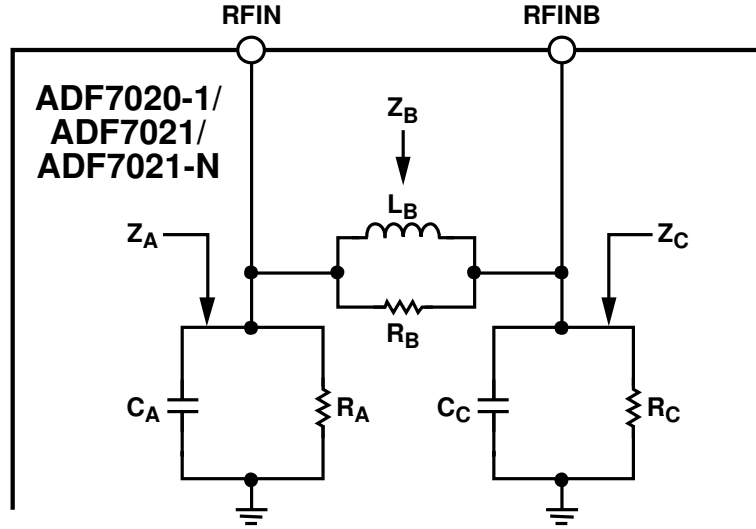


Figure 5.5: Lumped elements model of the RF input impedance of the ADF7021-N transceiver. At 435 MHz  $C_A = C_C = 2.225$  pF,  $R_A = R_C = 331.5$   $\Omega$ ,  $L_B = 2.85$   $\mu$ H,  $R_B = 9.55$  k $\Omega$ .

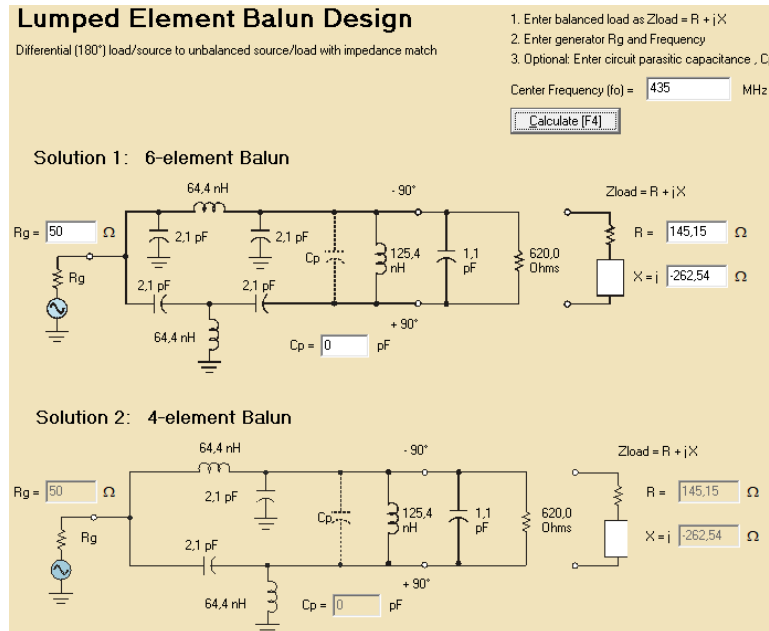


Figure 5.6: Balun design tool in AppCAD software. The design used is the "4-element Balun".

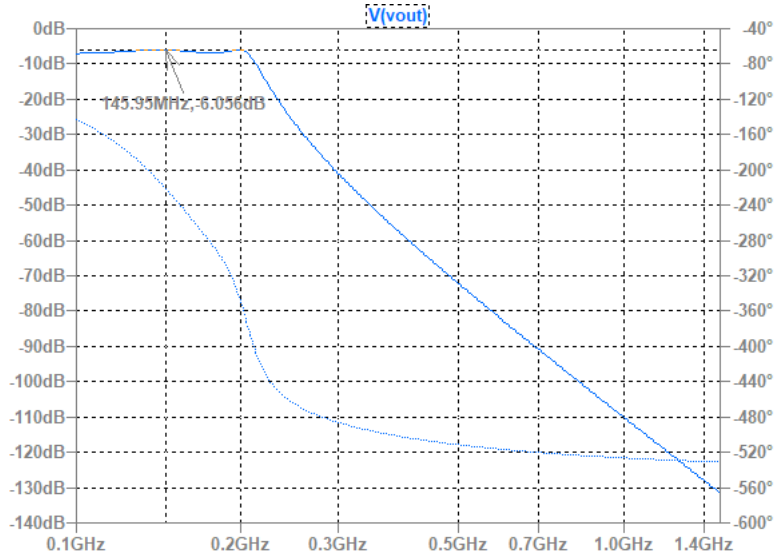


Figure 5.7: LTSpice Simulation of the Chebyshev low-pass filter. The frequency used of 145.95 MHz is located on top of the first ripple and is not attenuated. Due to the way the filter is simulated in LTSpice it shows a 6 dB attenuation in the pass-band region instead of 0 dB.

**PLL filter for the transmitter** The ADF7021-N chip uses a loop filter which integrates the current pulses from the Charge Pump (CP) to form a voltage that tunes the output of the VCO to the desired frequency[13]. The design of the filter is also closely related to out-of-band spurious emissions and in-band phase noise as well as the time required for the PLL to lock to the correct frequency. The datasheet recommends to use a filter with a bandwidth of 100 kHz. The *ADIsimSDR Design Studio* will be used to design the filter.

**PLL filter for the receiver** The same filter had to be designed for the RX transceiver using the same bandwidth and the same software.

### 5.2.4 Output Low Pass Filter

The transceiver is expected to emit non-negligible levels of harmonics. The datasheet states that unfiltered harmonics have a level of  $-35$  dBc for the second harmonic,  $-43$  dBc for the third harmonic and at most  $-36$  dBc for all the other harmonics. In order to suppress those harmonics a filter will be placed at the output of the circuit, after the external PA, to reduce its impact on the Signal to Noise Ratio (SNR). Our signal is very narrow-band and thus a Chebyshev filter was selected, allowing a high ripple in the pass-band region of the filter as this allows to have a steeper attenuation slope beyond the cut-off frequency. The filter was designed with a cutoff frequency of 210 MHz which is below the second harmonic frequency (291.9 MHz) but above the carrier frequency such that the carrier frequency would be at the top of the second ripple (see figure 5.7). The schematic of the low-pass filter is available in appendix D.6.

## 5.3 Layout design

On every board some design guidelines have been followed either specifically because this circuit carries radio frequency signals or because the circuit is designed to go in space.

Because the boards will carry RF signals it was decided to use 4-layers PCB. Even though the layouts were low complexity and did not require that many layers in order to route all the traces it was required to use 4 layers such that the RF traces could be designed as microstrips lines. This model adapts the width of the tracks with respect to its distance from the ground plane below such that  $Z_0 = \sqrt{\frac{L}{C}} = 50 \Omega$ . That means that if the track is too high above the ground plane the capacitance per unit length of track will be lower and thus the width of the track must be increased in order to compensate for that. With a 2-layers PCB the width of the track must be made so large that this



would be impractical to use. As 4-layers PCB are now not much more expensive than 2-layers PCB (depending on the manufacturer) it was decided to opt for 4-layers PCBs.

The stack-up used is the typical 4-layers stack-up with the layers arranged as such:

1. (top) signals, components, RF tracks,
2. ground,
3. power,
4. (bottom) signals (if needed).

However, on layers 1 and 4, the areas where no signal is routed were filled with ground planes.

The traces that will carry radio-frequency signals were thus designed such that their characteristic impedance would be  $50\Omega$ . The tracks were also shielded using vias to ground all around the traces. This ensures that the ground reference on the sides of the tracks are sufficiently coupled to the ground. The different ground planes of the circuit were then stitched together with vias placed at regular intervals. This also ensures that the ground reference is homogeneous all around the board. This also to a lesser extent helps with the ElectroMagnetic Compatibility (EMC) behaviour of the board because it creates a fine mesh through the board and will reduce the radiations of unwanted frequencies (for example the frequencies of the data signals connecting the transceivers to the OBC) and reduces the impact that external radiations may have on the signals of the board.

Because the board is intended to go in space the vias could not be capped (covered with solder-mask). However, when vias are placed in proximity to unrelated signals (such as the shielding vias that are close to the impedance matching components) the risk of a short circuit during assembly caused by a solder joint that spills on a via is high. The vias must thus be kept away from unrelated signals. The vias were also carefully placed such that an optical inspection of the board would be able to see all the vias and ensure that there is no short circuit of any kind.

The capacitors that are used in this board are, whenever possible, using the C0G dielectric. This ensures that their values will be kept stable when subject to temperature changes. However, such a stability is obtained at the cost of the value of the dielectric constant meaning that the maximum capacitance value of those components is low. Thus whenever capacitors with high capacitance were required other dielectrics such as X7R were used.

The size of the components was set to 0603 except for a few components that required larger footprints.

### 5.3.1 Power Amplifier

The layout for the test circuit of the power amplifier is also based on the recommended design by Qorvo. The footprint was reproduced on *Altium* with the *IPC Compliant Footprint Wizard*. The test board was designed such that three wires could be soldered to the board. One is the main power to the board and must be supplied with  $V_{CC} = 3.3\text{ V}$ . The second pad is connected to  $V_{APC}$  which is the voltage controlling the gain of the device. As the device will only be used at the maximum output power possible  $2.8\text{ V}$  must be applied to  $V_{APC}$ . The third pad is the ground pad. The board was fitted with SMA connectors to connect the board to the different measurement equipment that will be used later on. If no input signal is applied (but  $V_{APC} = 2.8\text{ V}$ ) it must consume a current  $I_Q \approx 200\text{ mA}$ . If a signal is applied the current at  $V_{CC}$  will be proportional to the output power. As the device must operate in the linear region the output power must not be greater than  $30.5\text{ dBm}$ [12]. At this point the efficiency is given to be 47%.  $30.5\text{ dBm}$  ( $1.12\text{ W}$ ) at the RF output means  $\frac{1.12}{0.47} = 2.38\text{ W}$  delivered by the power supply. With the power supply delivering  $3.3\text{ V}$  that means that the maximum current to be expected should be  $\frac{2.38\text{ W}}{3.3\text{ V}} = 0.720\text{ A}$ .

It is to be noted however that when designing the circuit, the inductor providing power at the RF output (L203 on the schematic in appendix D.1) was undersized. The footprint placed is a 0603 footprint and the current typically allowed through such an inductor ranges between  $400\text{ mA}$  and  $700\text{ mA}$  which is below the  $720\text{ mA}$  that could be expected from this design. Even if it is not much higher the current limit given for the inductor the temperature at which the circuit will be operating must be taken into account and this among other things mean that the current capacity of the components gets reduced.

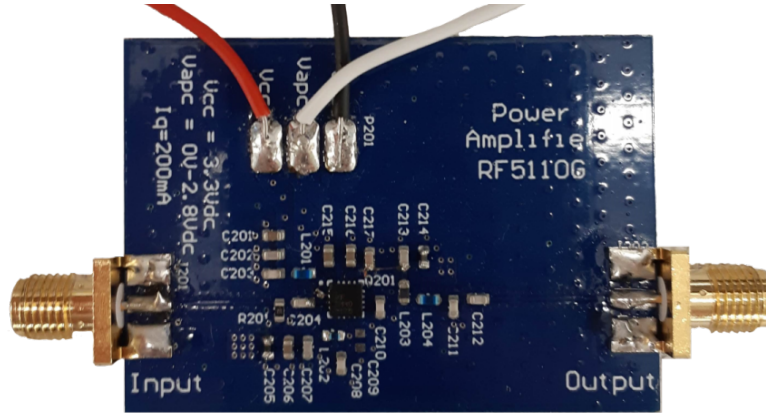


Figure 5.8: Test circuit built to measure the parameters of the RF5110G power amplifier.

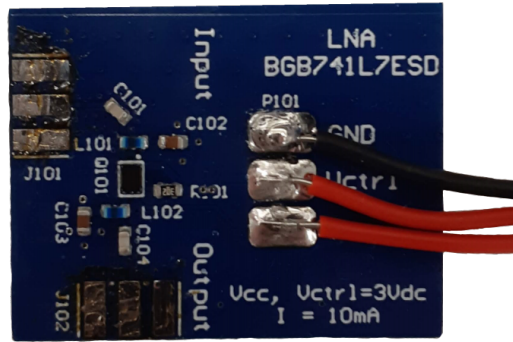


Figure 5.9: Test circuit built to measure the parameters of the BGB741L7ESD low noise amplifier.

### 5.3.2 Low Noise Amplifier

The layout of the test circuit for the low noise amplifier is copied from the design recommended by Infineon. The footprint for the chip was downloaded from their website. The very low components count made the design of the circuit very straightforward. The board was designed such that three wires could be soldered. The first pad is for the power supply and must be supplied with  $V_{CC} = 3.0\text{ V}$ . The second pad is connected to  $V_{ctrl}$  and turns the low noise amplifier on or off. It must be connected to  $3.0\text{ V}$  to power the LNA on and to ground to shutdown the LNA. The board was fitted with SMA connectors to connect the board to the different measurement equipment that will be used later on. The current expected on  $V_{CC}$  is  $10\text{ mA}$  no matter what the input power. The input power must be at all time below  $20\text{ dBm}$  and, in order to stay in the linear part of the gain curve, must stay below  $-8\text{ dBm}$ .

### 5.3.3 Transceiver

The layout of the test circuit for the transceivers was copied from the recommended design by Analog Devices and the footprint for the chip was downloaded from their website. As for the other circuits, care was taken while laying out the components that will be part of the RF signal tracks. Special care however was taken while laying out the balun as this kind of arrangement requires to be as symmetrical as possible.

The physical shape of the board is from the CubeSat Kit PCB Specification [16]. A connector was also placed at the specified location however this connector is not a PC104 connector nor the connector specified in the CubeSat Kit PCB Specification. It is a 120 pins connector with a pitch of  $2\text{ mm}$ . This will get fixed in chapter 6.

The layer 3 is divided in two planes. One plane is dedicated to the RX transceiver and the other one is dedicated to the TX transceiver. On each transceiver several signals were routed to the main connector. On the RX transceiver those are the configuration interface (SREAD, SDATA, SLE, SCLK), the data interface (TXRXCLK, TXRXDATA), the chip-enable signal (CE), the synchronisation word detection

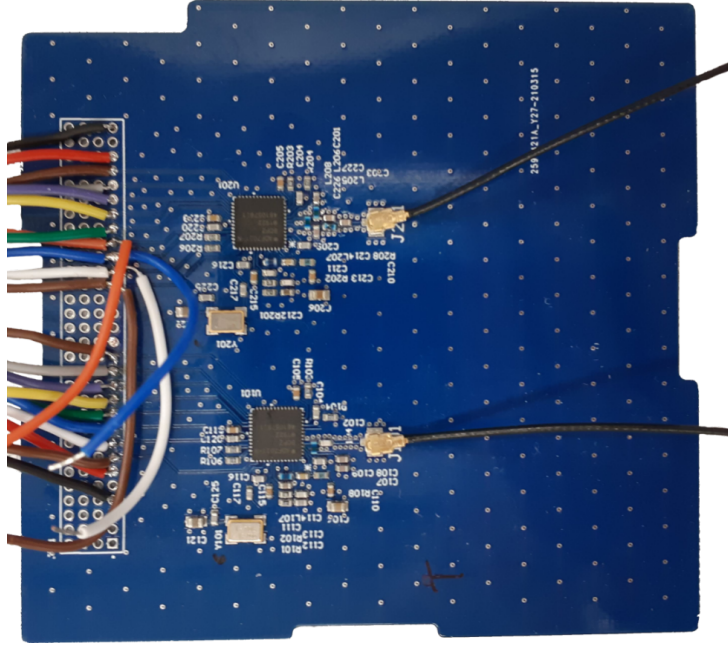


Figure 5.10: Test circuit built to measure the characteristics of the TX and RX transceivers.

signal (SWD) and the multiplexer output signal (MUXOUT). On the TX transceiver the same signals are also connected to the main connector except for the synchronisation word detection signal (SWD) which is of no use on the transmitter side.

The board was finally fitted with two U.FI connectors as those are the connectors that in the end will be on the satellite. The resulting board is shown in figure 5.10.

## 5.4 Test of the components

Once the boards are manufactured and the components assembled each component must be tested.

### 5.4.1 Power Amplifier

In order to test the parameters of the PA the three pads were connected to a power supply with  $V_{CC} = 3.3\text{ V}$  and  $V_{APC} = 2.8\text{ V}$ .

At first, it was attempted to measure the impedance and gain parameters with the ZNL3 Vector Network Analyser (VNA). However, the results showed a saturation in the gain measured at about 20 dB which is about 14 dB below the gain expected at the frequency used (see figure 5.11).

It turns out to be a limitation of the VNA and many different configurations were tested but none gave exploitable results. A new setup was thus established to test if the performances of the PA were correct.

The second method involves connecting an Arbitrary Waveform Generator (AWG) at the input of the power amplifier and the VNA in spectrum analyser mode at the output. The AWG is set to the frequency to use and its output power is adjusted until the output power reads what it is supposed to read. A hand-held digital multimeter was used to measure precisely the current consumed by the power amplifier while another hand-held digital multimeter was used to measure the voltages on the circuit, after the ampere-meter (see figure 5.12). This allowed to check if the efficiency of the device was indeed the power efficiency given in the datasheet and if the gain measured matched with the expected gain.

The first series of measurements was performed at 153 MHz which is a frequency for which the power efficiency and gains are given in the datasheet. The measurements were performed for an output power of 30 dBm (an output power of 1 W) and 30.5 dBm (the output power at the P1dB point). At  $P_{OUT} = 30\text{ dBm}$  the input power was  $P_{IN} = -4.7\text{ dBm}$  which gives a gain of  $30 - (-4.7) = 34.7\text{ dB}$ . The datasheet specifies a gain of 34.5 dB. The current measured was  $I_{CC} = 670\text{ mA}$  at a voltage

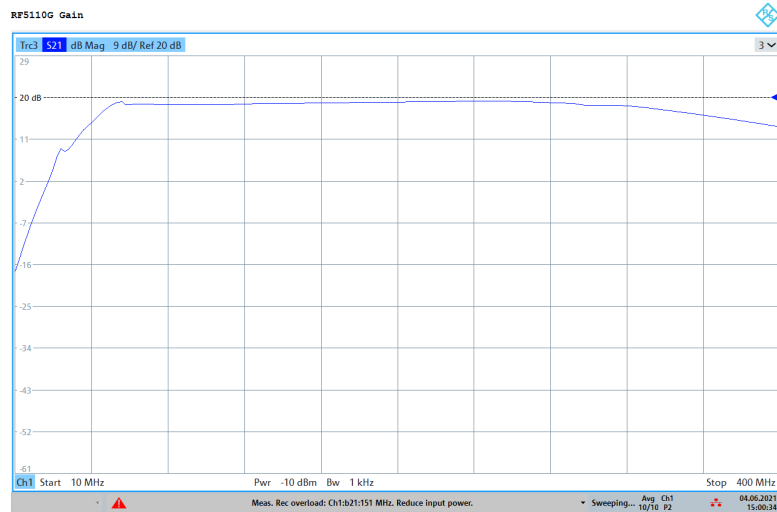


Figure 5.11: Gain measurement of the RF5110G on the VNA. The gain measured saturates at about 20 dB while a gain of 34 dB is expected.

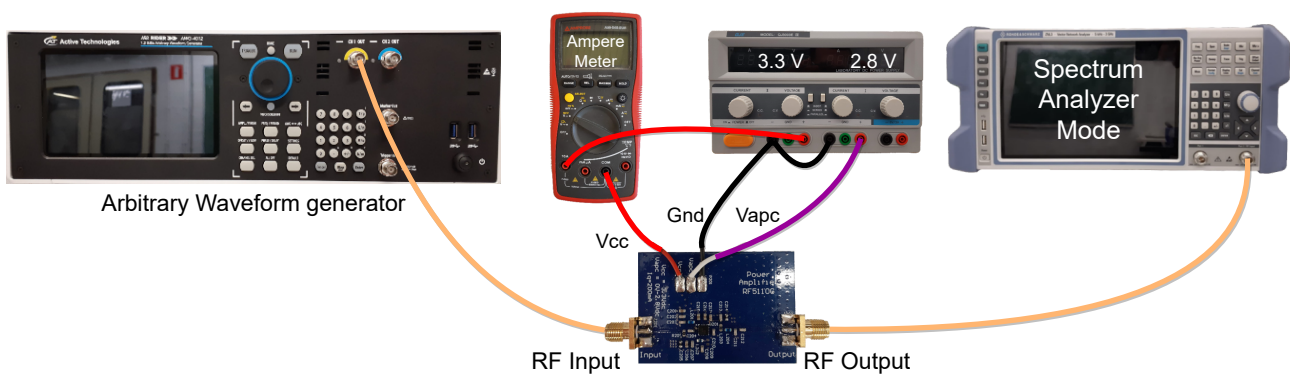


Figure 5.12: Setup used to measure the performances of the RF5110G Power Amplifier. The AWG generated a sinusoidal signal with a variable power and frequency.

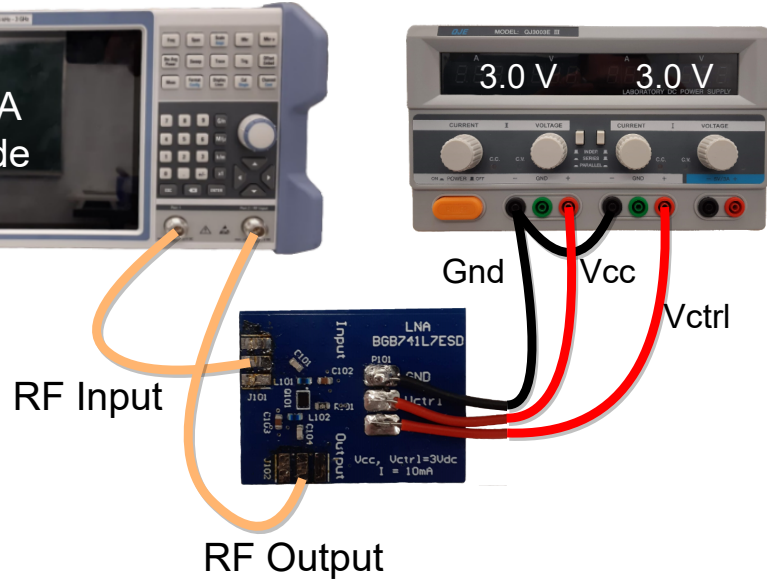


Figure 5.13: Test setup to measure the parameters and performance of the low noise amplifier.

$V_{CC} = 3.33 \text{ V}$  which gives an input power of  $3.33 \text{ V} \times 0.67 \text{ A} = 2.23 \text{ W}$  for an output power of  $30 \text{ dBm} = 1 \text{ W}$  or a power efficiency of  $\frac{1 \text{ W}}{2.23 \text{ W}} = 44.8\%$  while the datasheet specifies a power efficiency of  $45\%$ .

At  $P_{OUT} = 30.5 \text{ dBm}$  the input power was  $P_{IN} = -3.3 \text{ dBm}$  which gives a gain of  $30.5 - (-3.3) = 33.8 \text{ dB}$ . The datasheet specifies a gain of  $34 \text{ dB}$ . The current measured was  $I_{CC} = 710 \text{ mA}$  at a voltage  $V_{CC} = 3.32 \text{ V}$  which gives an input power of  $3.32 \text{ V} \times 0.71 \text{ A} = 2.36 \text{ W}$  for an output power of  $30.5 \text{ dBm} = 1.12 \text{ W}$  or a power efficiency of  $\frac{1.12 \text{ W}}{2.36 \text{ W}} = 47.45\%$  while the datasheet specifies a power efficiency of  $47\%$ .

Both those measurements indicate that the circuit behaves exactly as expected in the conditions specified by the manufacturer. Once the circuit is known to be working as expected the measurements at the frequency used can be performed.

At  $P_{OUT} = 30 \text{ dBm}$  the input power was  $P_{IN} = -4.2 \text{ dBm}$  which gives a gain of  $30 - (-4.2) = 34.2 \text{ dB}$ . The current measured was  $I_{CC} = 700 \text{ mA}$  at a voltage  $V_{CC} = 3.33 \text{ V}$  which gives an input power of  $3.33 \text{ V} \times 0.7 \text{ A} = 2.33 \text{ W}$  for an output power of  $30 \text{ dBm} = 1 \text{ W}$  or a power efficiency of  $\frac{1 \text{ W}}{2.33 \text{ W}} = 42.9\%$ .

At  $P_{OUT} = 30.5 \text{ dBm}$  the input power was  $P_{IN} = -3.1 \text{ dBm}$  which gives a gain of  $30.5 - (-3.1) = 33.6 \text{ dB}$ . The current measured was  $I_{CC} = 740 \text{ mA}$  at a voltage  $V_{CC} = 3.32 \text{ V}$  which gives an input power of  $3.32 \text{ V} \times 0.74 \text{ A} = 2.46 \text{ W}$  for an output power of  $30.5 \text{ dBm} = 1.12 \text{ W}$  or a power efficiency of  $\frac{1.12 \text{ W}}{2.46 \text{ W}} = 45.5\%$ .

It can be observed that the performances of the device are slightly reduced at the frequency of  $145.95 \text{ MHz}$  which was to be expected given that the design is created for a frequency of  $150 \text{ MHz}$ . This degradation however is sufficiently small as to not require the design of a new input and/or output matching networks to improve the performances of the power amplifier.

#### 5.4.2 Low Noise Amplifier

The test phase of the low noise amplifier was not as straightforward as expected. At first the circuit was attached to a power supply (as in figure 5.13). As the current expected is about  $10 \text{ mA}$  an external multimeter in ampere meter mode was used to get a more precise measurement of the current than what the power supply could provide. However, it turned out at first that the device was not consuming any current.

The datasheet was meticulously inspected and it was discovered that there was a confusion in the datasheet about where the pin 1 key was placed on the chip. Some drawings did indicate that the key was instead on the pin 6. It was thus asked to the technician if he could mount the chip on the board as if the key indicated the pin 6 instead of the pin 1. During this operation however he discovered



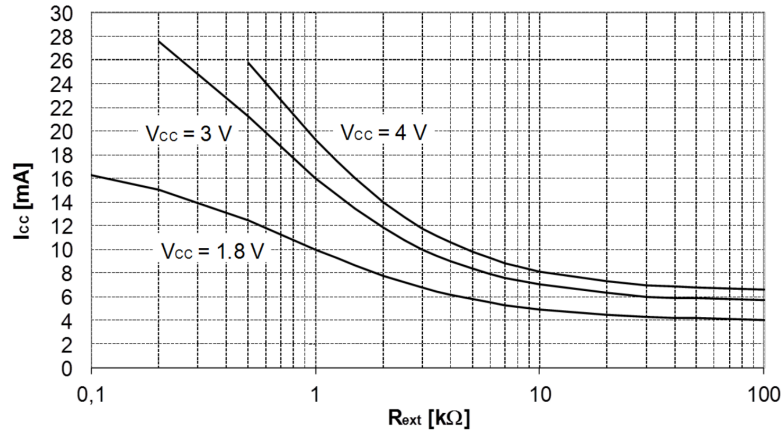


Figure 5.14:  $I_{CC}$  current consumption of the low noise amplifier versus the value of the  $R_{ext}$  resistor used to set the value of  $I_{CC}$ . The device operates at 3.0 V.

that the ground pad below the device was completely covered in soldermask. This came from the footprint downloaded from the manufacturer's website. The soldermask covering the ground pad was then carefully removed by scraping it under a binocular microscope until the copper underneath was fully visible. The pad was then tinned with a soldering iron and the circuit was sent back to the white room for a new assembly. Two assemblies were asked as the confusion about the pin key was not settled yet.

Upon receiving the newly assembled chips it turned out that the board where the amplifier was mounted in reverse (as if the key indicated the pin 6) was not functioning. It meant that the key was indeed indicating the pin 1 as first expected.

The circuit was then re-attached to a power supply (still as in figure 5.13). First the power supply is brought up to  $V_{CC} = 3.0$  V and then  $V_{ctrl}$  is brought up to 3.0 V too. At that time however the ampere meter showed a current of  $I_{CC} = 30$  mA. This is 3 times more than the 10 mA that were expected but this is not more than the maximum current the chip can handle so the next tests were performed anyway.

The last test to be performed is to attach the low noise amplifier to the VNA such that the ports impedance can be measured as well as the gain. The gain showed to be about 15 dB which is way below the 21.5 dB given in the datasheet. In order to verify that it was not again a limitation of the VNA a setup similar to what was finally used for the PA was used. It showed the same gain of 15 dB. This is probably linked to the higher than expected current. The device was then tested without any resistance to set the current which is supposed to reduce the current to about 6 mA but the result stayed the same.

By analysing the datasheet and looking at the curve of the current consumption versus the current setting resistor value (see figure 5.14) it could be guessed that a current consumption of 30 mA is due to a  $R_{ext}$  value of approximately  $0\Omega$ , or a short circuit between the pin and  $V_{CC}$ . The circuit was reassembled several times however the result stayed the same. Given the time needed for a re-assembly no more trials were performed and no further measurement were performed on the device.

### 5.4.3 Transceivers

In order to assess the performances of the transceivers, many measurements can be done in order to test all the subsystems. At first the general operation of the device can be assessed. Even though this requires some code to be written (this will be explained in chapter 7) it can already be said that the device works as expected and all the interfaces (data and configuration) are working as expected. The devices powers up properly and is able to receive and send signals at the requested frequencies.

The subsystems that require advanced measurements are mainly

1. the input impedance matching of the RX transceiver,
2. the VCO of both transceivers,

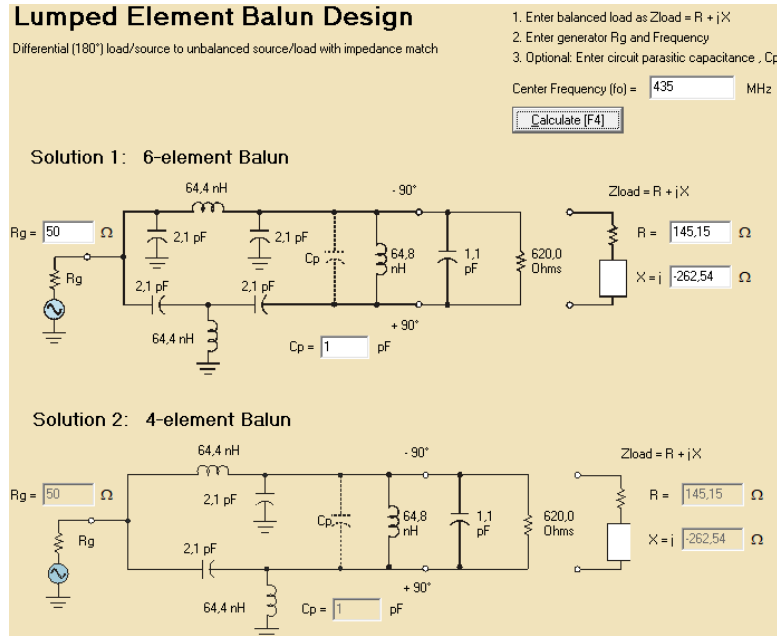


Figure 5.15: Caption

3. the output of the TX transceiver,
4. the sensitivity of the RX transceiver.

### Receiver's input impedance matching

The VNA was calibrated and then an impedance measurement ( $S_{11}$ ) was performed on the input of the transceiver. The first measurement showed a return loss of about  $-5$  dB at a frequency of 435.015 MHz. The graph however also showed a return loss of about  $-10$  dB at a frequency of about 360 MHz which is well below the frequency at which it is intended to be used. This seems to indicate that the input impedance matching is, in this state, incorrect. This was expected as impedance matching tends to be sensitive to parasitic impedances. In the *Lumped Element Balun Design* tool from *AppCAD* it is possible to give to the tool a parasitic capacitance between the two branches of the balun. This parasitic impedance originates from the proximity of the traces and components used for the rest of the balun and seems to be non-negligible. At the time of the theoretical design it was impossible to predict what this parasitic capacitance could be and so it was just ignored. By entering a parasitic value of 1 pF it can be seen that the design changes and the value of the shunt inductor placed between the two inputs drops to about 64.8 nH (see figure 5.15). As measuring this precise value could quickly become quite tedious it was decided to perform by trial-and-error and to change the value of the 120 nH inductor. It is to be noted that the power levels applied to that port are very low and any impedance mismatch (even a severe impedance mismatch) would not be able to damage the components so the trial-and-error method is not dangerous for the circuit.

It was expected that decreasing the value of the 120 nF shunt inductor would bring the dip in the return loss from 360 MHz closer to 435 MHz. If the dip was higher than 435 MHz that would mean that the new value of the inductor is too low. If it increased in frequency but stayed below 435 MHz that would mean that the value needs to be reduced again.

The first value to be tested is 68 nH. The inductor was changed and the board was connected to the VNA. It appears that the new matching network is actually very good and the return loss measured with the VNA is below  $-14$  dB (see figure 5.16). Return losses are expressed as negative number in decibels and correspond to the fraction of the emitted power that gets reflected back by the load to the source (due to an impedance mismatch). That is (if  $RL$  is the return loss,  $P_i$  the incident or transmitted power and  $P_r$  the reflected power)  $RL = 10 \log_{10} \frac{P_r}{P_i}$ . So in our case with the return loss of  $-14.45$  dB,

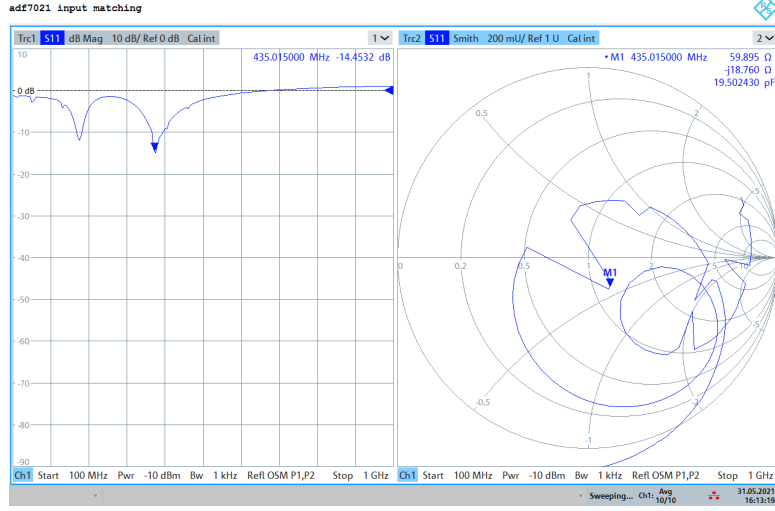


Figure 5.16: VNA measurement of the input impedance of the receiving transceiver with the modified shunt inductor. The return loss at 435.015 MHz is  $-14.45$  dB and the port impedance is  $60 - 19i\Omega$ .

$$\begin{aligned}
 -14.45 &= 10 \log_{10} \frac{P_r}{P_i} \\
 \iff -1.445 &= \log_{10} \frac{P_r}{P_i} \\
 \iff 10^{-1.445} &= \frac{P_r}{P_i} = 0.036
 \end{aligned}$$

. So 3.6% of the incident power gets reflected. Usually a return loss of  $-10$  dB (or a reflection of 10% of the incident power) is considered good.

In addition to the impedance and return loss measurements a Received Signal Strength Indicator (RSSI) measurement was also performed. The SDR was configured to emit a pure tone at the carrier frequency of 435.015 MHz. The level was then measured via the spectrum analyser mode of the VNA. The SDR must also be configured such that the signal emitted is below  $-47$  dBm as this is the upper limit the ADF7021-N's RSSI measurement can report. Any value higher than that will be read back as  $-47$  dBm. At the lower power setting the SDR emits a signal with a power of  $-52$  dBm recorded on the ZNL3. With the first balun (with a return loss of  $-5$  dB) the ADF7021-N reported an RSSI of  $-52$  dB or the same as what the spectrum analyser reported. However, with the corrected balun the ADF7021-N's RSSI measurement reported  $-49$  dBm or about twice the received power which is also higher than the power measured by the VNA. The fact that the ADF7021-N reported a higher received power than the spectrum analyser could be due to the filter's bandwidth used by the spectrum analyser during its measurement. It could also be an offset in the readings of the ADF7021-N which is given to have an absolute accuracy of  $\pm 3$  dB. The RSSI measurement must thus be used carefully. However, the 3 dB increase of the reading can be attributed to the gain of performances due to the modified balun.

Actually a return loss of  $-5$  dB corresponds to a reflected power of  $10^{-0.5} = 32\%$  or a transmitted power of 68%. Going from  $-5$  dB to  $-14$  dB means that the transmitted power increased from 68% to 97% or an increase of 42%. So the increase of the RSSI by 3 dB do contain measurement errors. The module is however not expected to be used to perform accurate measurement so this is not a real concern.

### Receiver's VCO inductor

As per the datasheet of the ADF7021-N the VCO's inductor must be selected such that the voltage at the VCOIN pin of the transceiver is at a voltage between 0.2 V and 2 V and should ideally be in



the center of the range (or 1.1 V)[13]. The voltage at the VCOIN of the receiver was measured using a hand-held digital multimeter. The value read was 1.03 V which is close enough to the 1.1 V specified in the datasheet. No further actions were taken.

### Transmitter's VCO inductor

The same procedure of measuring the voltage at the VCOIN pin was performed on the transmitting transceiver. It was measured at a voltage of 0.33 V which is a bit lower than the ideal value. Given that the circuit might be subject to large temperature changes the value of the inductor have to be changed such that the voltage at the VCOIN is closer to the ideal 1.1 V.

As the voltage read was lower than required that meant that the value of the inductor had to be increased (see figure 5.3). The circuit was originally designed with an inductor of 18 nH and the only inductor available with a higher inductance was a 22 nH inductor. Replacing the inductor gave a voltage at the VCOIN pin of 2.18 V which is higher than the maximum value acceptable. The datasheet however also states that the VCO center frequency can be adjusted by software (bits 23 and 24 in the configuration register 1, see section 7.1.3 for more details). The VCO center frequency was thus adjusted by the maximum allowable in the configuration register. The voltage at the VCOIN pin consequently dropped to 1.23 V which is within the range acceptable. There was in the lab no inductor with a value closer to what would be needed to operate without having to adjust the VCO center frequency so this will be done in chapter 6.

### Output of the transmitting transceiver

In the case of the transmitting transceiver the output impedance can not be simply measured using a VNA because the component emits a signal that is not related in any way to the signal emitted by the VNA. Also, when shut down, the output port goes in high-impedance mode which is not the same as when it is transmitting. In order to measure the output port impedance it would be required to perform what is called a load-pull impedance measurement. In this type of measurement the emitted power is measured while the load impedance changes. The impedance corresponding to the maximum recorded power corresponds to the conjugate value of the output impedance. This type of measurement is however not possible with the equipment available. Thus in order to assess the accuracy of the output impedance matching, the output power levels were compared against the values given in the datasheet [13]. The output power of the device can be adjusted via a field in a configuration register (bits 13 to 18 from configuration register 2, see section 7.1.3 for more details). The output power was thus recorded for several output power settings. Figure 5.17 shows a plot with both the measured output power and the output power given in the datasheet. The measured output power is at all points 1 dBm to 5 dBm higher than the output power given in the datasheet. This might be because the output matching used here does not contain any filter. This might also be because the output power measurements shown in the datasheet were not performed at the same frequency (no frequency is given in the datasheet) or because this design does not use the Rx/Tx switch that may have been used to perform the measurements for the datasheet. However, the fact that the output power is above what could be expected allows to conclude that the output matching is correct.

The output spectrum was also analysed to measure the OBW and to see the levels of the spurious emissions. The OBW of the transmitter in 2FSK was measured to 20 kHz (see figure 5.18). It is a bit higher than the OBW of the uplink signal because the TNC performs some low pass filtering of the signal while the ADF7021-N performs a 2FSK modulation with no filtering. It actually is a little bit below the theoretical bandwidth for a 2FSK modulated signal at 9600 Bd with a frequency deviation of 4800 Hz which have a theoretical OBW of

$$OBW = 2 \times deviation + baud\_rate = 2 \times 4800 + 9600 = 24 \text{ kHz}$$

[17].

The spurious emissions generated by the ADF7021-N were also measured and compared against the values given in the datasheet. The datasheet specifies spurious emission with a level of -65 dBc. The device was configured for an output power of 0 dBm. The levels were recorded with the VNA in spectrum analyser mode. The frequency span was configured to range from 5 kHz to 300 MHz. As

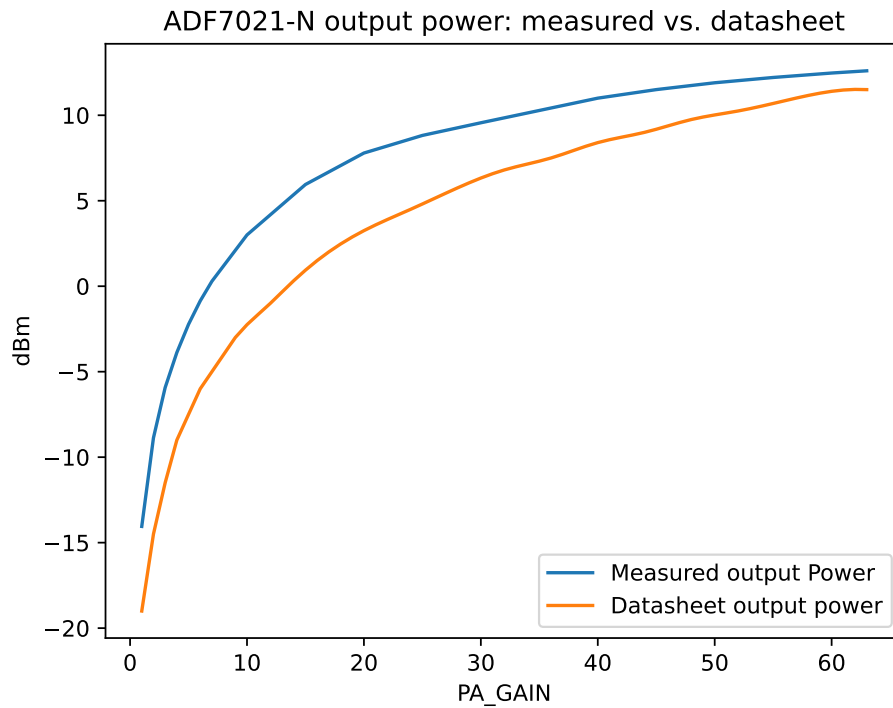


Figure 5.17: Comparison of the measured output power and the output power given in the datasheet.

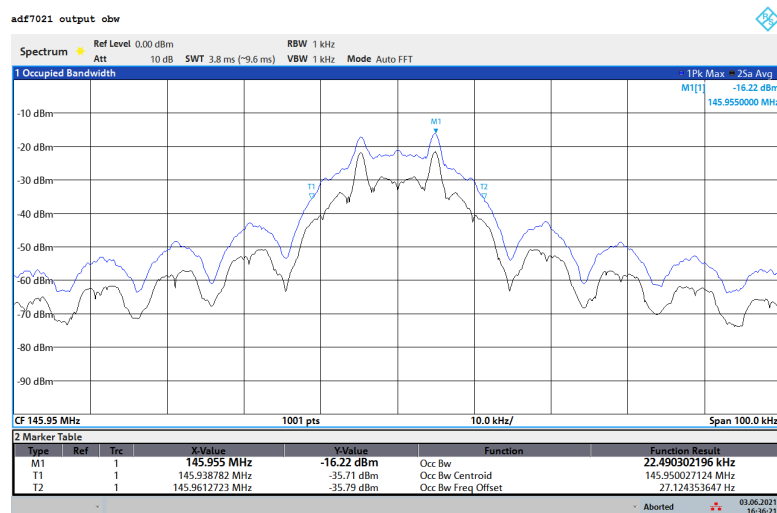
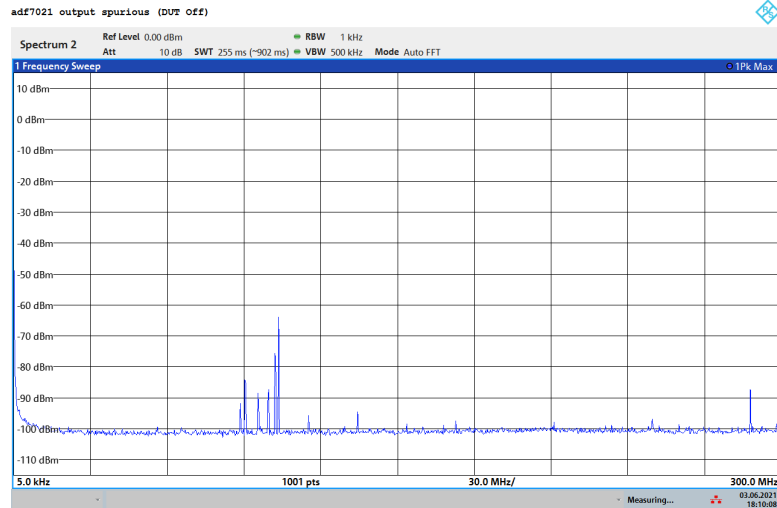
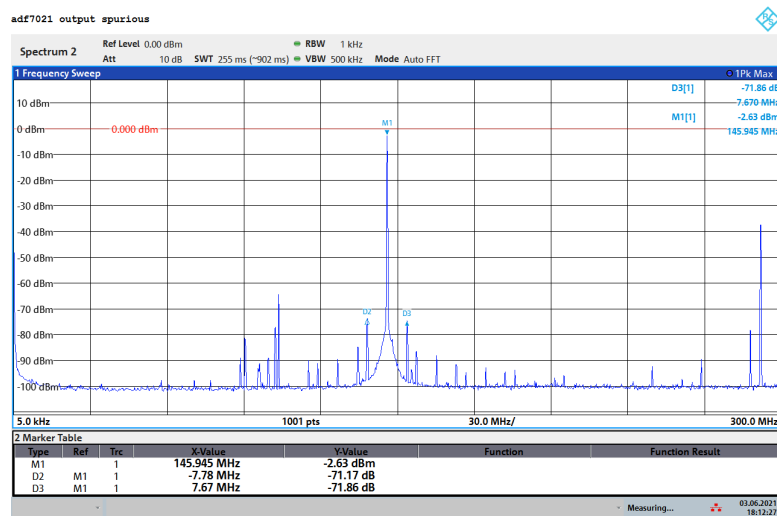


Figure 5.18: Occupied Bandwidth measurement of the signal generated by the ADF7021-N transceiver. The bandwidth is measured to be 22.49 kHz



(a) Measurement with the device turned off. Noise can be measured between 90 MHz and 100 MHz and should not be ignored when measuring the level of the spurious emissions.



(b) Measurement with the device turned on.

Figure 5.19: Spurious emissions measurement at the output of the Tx ADF7021-N. The highest spurious emission recorded is 71 dB below the carrier power level.

the spurious emissions are typically very sharp spikes in the spectrum the measurement device was configured with a narrow filter bandwidth. The trace displayed holds the maximum of the signal at any time. The spurious emissions are concentrated around the carrier frequency and above 200 MHz any emission will be attenuated by the filter placed at the output. The figure 5.19 shows the result of the measurement. However, as the signals measured have a very low power it is much more susceptible to external noise which could originate from outside of the system but still be measured by the spectrum analyser. In order to know what signals are noise and what signals are actual spurious emissions, a first measurement with the device turned off was performed before the actual measurement.

## Sensitivity of the RX Transceiver

Measurements were performed to try to estimate the effective sensitivity of the RX transceiver. However, the lack of RF attenuators made it difficult to reduce the power of the signal below a certain power. Using all the available attenuators and using the SDR (see figure 5.20) at the lowest output possible it was possible to attenuate the signal down to an estimated power of about  $-110$  dBm (power measured via the RSSI read back of the RX transceiver). The signal was also measured with the VNA in spectrum analyser mode with a peak power of  $-116$  dBm (see figure 5.21). However, at power levels that low the appearance of the signal on a spectrum analyser is highly dependant on the bandwidth

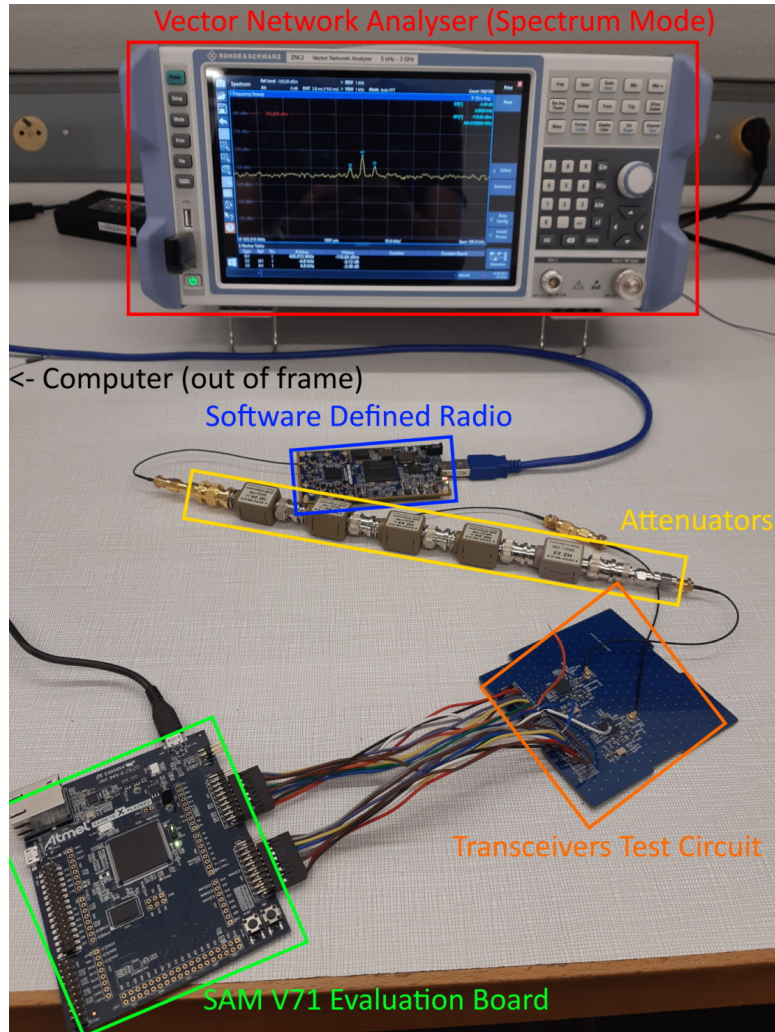


Figure 5.20: Setup used to measure the sensitivity of the RX transceiver.

of the filter at the input of the spectrum analyser. This combined with the absolute accuracy of the RSSI measurement of 3 dB for signals above  $-100$  dBm (and thus probably a worse absolute accuracy at power levels below  $-100$  dBm) make it hard to know the exact power of the signal applied at the input of the RX transceiver. As it was seen at the time of measuring the performances of the input network matching in section 5.4.3 it seems that the RSSI measurement of this particular transceiver tends to be on the high side, i.e. it tends to report an RSSI value a bit higher than the real power applied. Under that assumption it can reasonably be assumed that the signal applied was at most at  $-110$  dBm. The sensitivity given in the datasheet is  $-115$  dBm.

The tests performed were quite basic and consisted in sending packets to the module using the setup developed in part I. A total of 30 packets were sent to the module. All the 30 transmissions produced a valid SWD interrupt and the corresponding packet was successfully decoded every time and sent back to the computer. It is to be noted however that a few false SWD interrupts were triggered during that time but the module successfully discarded the data following the false SWD interrupts and returned to a valid listening state.

One last remark is that to enable these tests the Automatic Frequency Correction (AFC) (see section 7.1.3) had to be disabled otherwise the device would not be able to detect the incoming signal (and even reported an RSSI level of  $-124$  dBm or the minimum value that can be reported). This is most probably because the device was not able to lock its AFC on the signal sent due to its low power and was instead locked on another frequency. If the device was fully reset and the test was performed quickly after its initialisation it was possible to decode a few packets which would confirm the validity of this hypothesis.

Finally to make sense of these measurements it should be compared with the expected power level received by the nanosatellite during its mission. Providing that the link budget (see appendix A) is

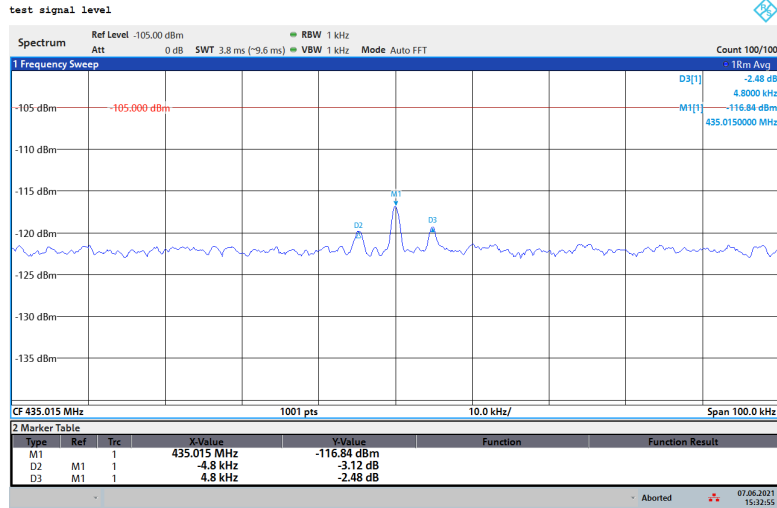


Figure 5.21: Spectrum analyser view of the signal generated by the SDR after a series of attenuators to test the sensitivity of the RX transceiver.

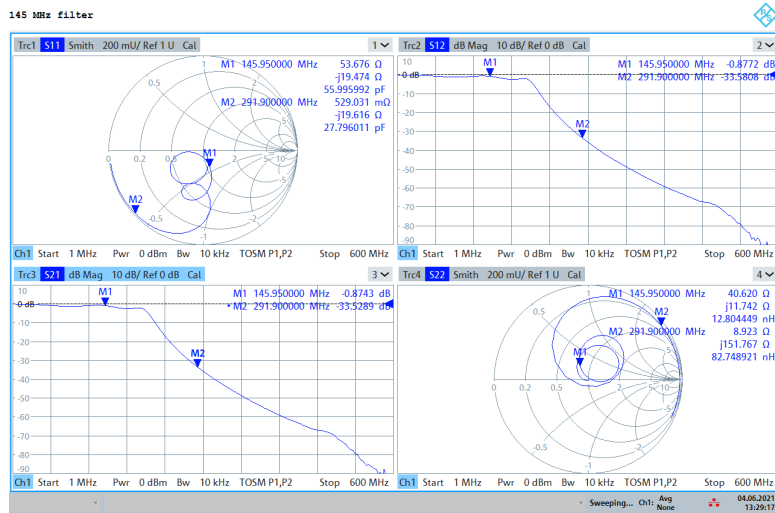


Figure 5.22: VNA measurement of the characteristics of the low-pass filter used to filter out the harmonics of the 145.95 MHz downlink signal. The impedance is close to  $50\Omega$  and the signal is located at the top of the attenuation curve.

correct the power received at the spacecraft should be about  $-97\text{ dBm}$ . The LNA should provide a gain of 20 dB bringing the power of the signal to  $-77\text{ dBm}$ . If the spacecraft indeed receive a signal as powerful as that, this test confirms that there will be no issue with the sensitivity of the transceiver. The problem with the AFC should even disappear. This however does not account for the SNR ratio of the signal but the link budget estimates the SNR to be above 15 dB and if that is correct this is a good enough ratio to be simply ignored.

#### 5.4.4 Output Low Pass Filter

The characteristics of the low-pass filter were measured with the ZNL3 VNA. The insertion losses were measured at about 0.8 dB and the input as well as the output impedance were measured to be close to  $50\Omega$  (see figure 5.22). An insertion loss of nearly 0.8 dB is non negligible when the signal have an output power of nearly 1 W and the size of the components used could be increased to reduce the dissipation losses due to the resistance of the series inductors in the filter.

# Chapter 6

## Second Hardware Design

With all the subsystems being independently tested, a board including all the components required can finally be designed. The board will include the two transceivers as well as the power amplifier and the low noise amplifier.

The design however does not simply consist in an interconnection of all the subsystems. Other changes (or fixes) were also included and will be presented throughout this section along with some smaller subsystems which had to be added such as current measurement circuits.

In this circuit the inductor placed at the output of the **RF5110G** PA is still undersized because this error was discovered too late. It should not influence the performances of the system but is a concern for prolonged use of the power amplifier. That mistake must be corrected if this work is to be re-used.

### 6.1 Modifications

#### 6.1.1 Low Noise Amplifier

Even though the design of the low noise amplifier could not be properly validated it was decided to include it on the board. If the fault in the circuit was due to handling of the board while removing the soldermask on top of the ground pad or due to repeated soldering attempts the circuit could very well be working in this version of the design. However an additional U.Fl connector was included on the board between the input of the receiving **ADF7021-N** transceiver and the output of the low noise amplifier. If the LNA appeared to be faulty again it could simply be bypassed. The footprint was also corrected so that no soldermask would be applied on the ground pad.

#### 6.1.2 Temperature Compensated Crystal Oscillators

The TCXO at the reception is, as in the first test circuit for the transceivers, simply connected to the 3.3V rail for the reception subsystem. This is because the spacecraft will always be listening for incoming telecommands and shutting down the TCXO would imply to shut-down the transceiver which is not possible.

The TCXO at the emission however will have its power supply provided by a pin of the OBC's microcontroller. The maximum current consumed by the TCXO is about 2 mA [18] which is below the maximum current that can be provided by an Input/Output (IO) pin of the **SAM V71** microcontroller[19]. This will allow to reduce slightly the power consumption of the board when not emitting.

#### 6.1.3 Main Connector

The main connector was also fixed from being a 120 pins connector to being a more standard 104 pins connector. This however is not a standard PC/104 connector and some explanations about the exact form factor used and standards followed is required.

The shape of the board itself is the shape of a PC/104 board with some slots cut out to accommodate for the other systems of the cubesat. The PC/104 standard however specifies two bus types, an 8-bits bus and a 16-bits bus with respectively a 64 pins or 104 pins connector on the lower part of the board (see figure 6.1a)[20]. However in this particular circuit none of those are placed.

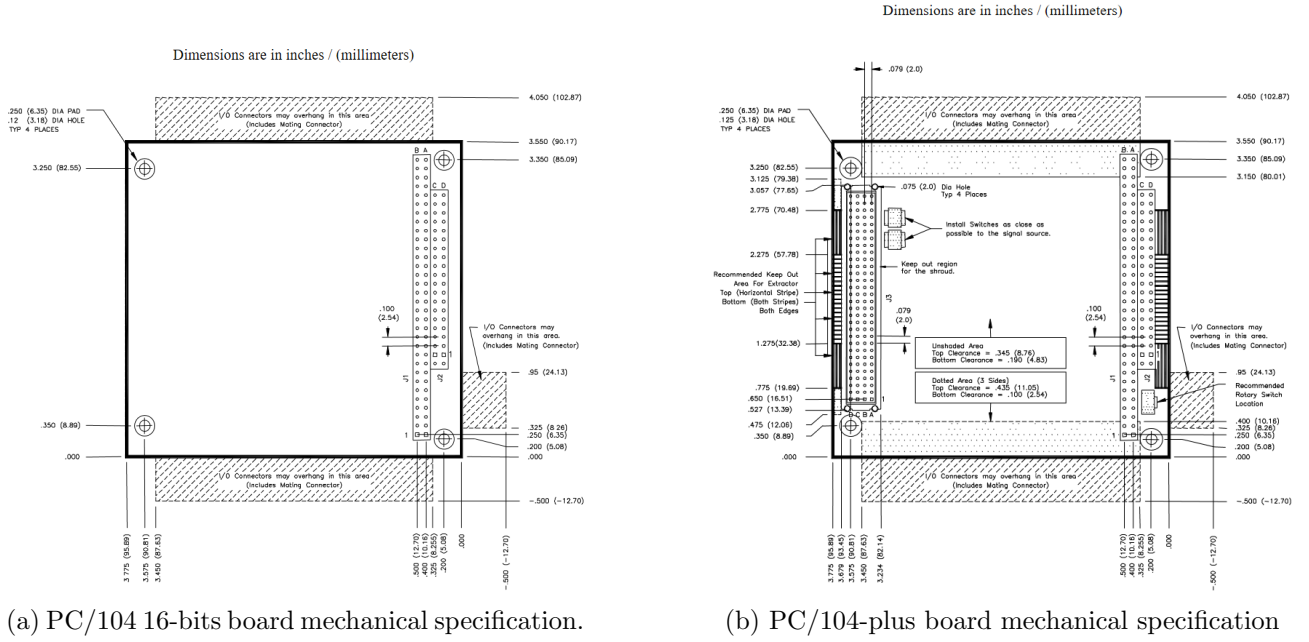


Figure 6.1: Comparison of the PC/104 board mechanical specification versus the PC/104 plus board mechanical specification.

The board might also look like a PC/104-*plus* board. That board have the same shape and connector as the PC/104 16-bits board, with an additional 104 pins PCI connector on the opposite side of the board (see figure 6.1b)[21]. This board however is not a PC/104-*plus* board either and the connector used is not a PCI connector as the PCI connector uses the same number of pins but with a pitch of 2mm while the connector used uses a 104 pins connector with a 2.54mm pitch.

The specification used for this board is actually the *Pumpkin CubeSat Kit PCB Specification* (see figure 6.2) without the PC/104 16-bits connector[16]. The connector used is thus a *CubeSat Kit Bus Connector* which looks like a PCI connector but is not compatible with an actual PCI connector.

## 6.2 New Subsystems

A few minor subsystems had to be added to the board to ensure everything would work as expected. This consists in voltage regulators and current measurement circuits. The power for the whole board will come from the main connector at 3.3V. It will then be split in two distinct rails: one 3.3V for the transmitter part and one 3.3V for the receiver part. The current on each part will be monitored.

### 6.2.1 Current measurement

The power on each rail needs to be measured. The best way to measure the current in this kind of configuration is to measure the voltage difference across a shunt resistor. The voltage generated across the shunt resistor is then amplified by a current shunt monitor which is basically a low noise operational amplifier (also called current-sense amplifiers). The shunt resistors will be placed between the common 3.3V rail coming from the main connector and each sub-rail for TX and RX. However the current on each rail will be different.

On the RX rail the devices consuming current are the ADF7021-N transceiver and the BGB741L7ESD LNA. The maximum current specified for the transceiver in reception is 19.5mA at 3.0V which is about  $\frac{3.3V}{3.0V} * 19.5mA = 21.45mA$  at 3.3V. The LNA consumes 10mA at 3.0V which is about  $\frac{3.3V}{3.0V} * 10mA = 11mA$  at 3.3V. The total is thus  $21.45 + 11 \approx 32.5mA$ . A shunt resistor of  $1\Omega$  was selected. It should produce a voltage drop of at most  $1\Omega \times 32.5mA = 0.0325V$  and dissipate a power of  $1\Omega \times (32.5mA)^2 \approx 1mW$ .

On the TX rail the devices consuming current are the ADF7021-N transceiver and the RF5110G PA. The maximum current specified for the transceiver in emission is 13.5mA at 3.0V (for a 0dBm output) which is about  $\frac{3.3V}{3.0V} * 13.5mA = 14.85mA$  at 3.3V. The PA consumes 720mA at 3.3V. The



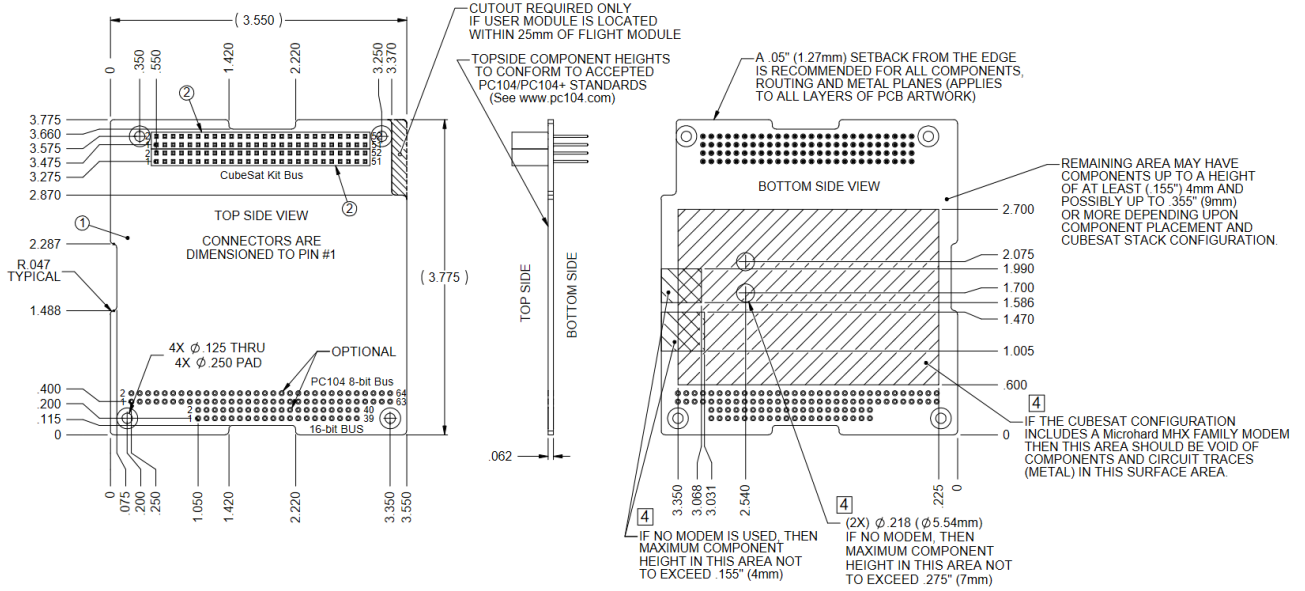


Figure 6.2: Pumpkin CubeSat Kit Mechanical Specification

total is thus  $14.85 + 720 \approx 735 \text{ mA}$ . A shunt resistor of  $52 \text{ m}\Omega$  was selected. It should produce a voltage drop of about  $52 \text{ m}\Omega \times 735 \text{ mA} = 0.038 \text{ mV}$ . The voltage drop produced by the shunt inductor can not be too large or the performances of the PA could be reduced by the consequently lower  $V_{CC}$ .

The current-sense amplifier selected is the INA210 from Texas Instrument[22].

### 6.2.2 3.0 V Linear Regulator

A 3.0 V linear Low-Dropout regulator (LDO) regulator is needed to power the LNA. As it needs to be powered in 3.0 V and requiring a dedicated 3.0 V rail from the Electrical Power System (EPS) would require too much work given the power consumption it was decided to use such a regulator. It is powered from the 3.3 V RX power.

### 6.2.3 2.8 V Linear Regulator

A 2.8 V linear LDO regulator was needed for the PA, the voltage required at the APC pins of the PA. While these pins are intended for power modulation at the output of the PA, an effective regulation of the power is only possible with a feedback of the output power which is not planned as the nanosatellite will always emit at the maximum output power. It must thus be set at a fixed 2.8 V for stable performances. This voltage will be generated via a LDO with a shutdown pin. This allows the OBC to fully shutdown the PA by shutting down the 2.8 V LDO (when  $V_{APC} = 0 \text{ V}$  the PA current is reduced to  $1 \mu\text{A}$ )[12].

## 6.3 Tests

The module could not be fully assembled in time and no test can thus be presented in this work. It is expected that the measurements performed on the test circuits will still hold. What could be seen is that the output power of the board is slightly lower than expected. This might come from the output filter. If needed the output power of the TX transceiver can be slightly changed to meet the output power required but the compression point of the PA should not be exceeded.

As an illustration of the resulting board the figure 6.3 shows a 3D rendering of the board.



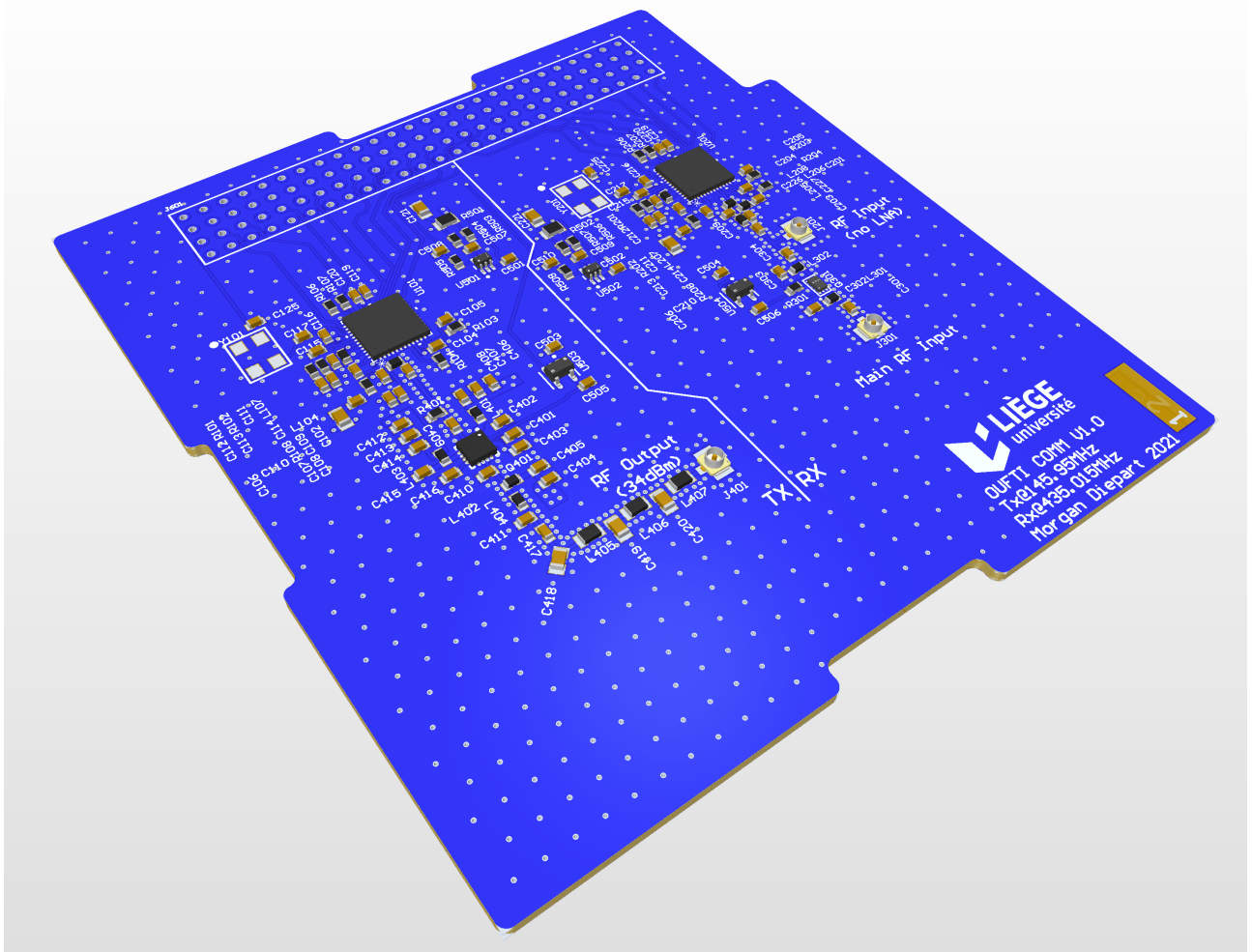


Figure 6.3: 3D rendering of the PCB of the second hardware design.

# Chapter 7

## Firmware

While doing this work the OBC of the nanosatellite was being developed as the topic of another thesis. That means that the choices of microcontroller and software architecture for the OBC are already settled. The microcontroller is a **SAM V71** from Microchip and is an ARM microcontroller clocked at 300MHz and equipped with a myriad of peripherals. The software will be architected around the FreeRTOS real-time operating system. The software will be written on the *MPLAB X IDE* software using the *MPLAB Harmony 3* framework. In order to write code for the COMM board on the OBC the same tools will be used. A development board will also be used. It conveniently provides a single USB port to power and interface with the board. It provides a debugger as well as a virtual serial port which will be used to interact with the firmware. Based on that some code will be written to interface with the COMM board, that is configure and interface with the transceivers, power on and off the components that do not require to be always powered on and send and receive data via the radio link. It will also enable to encode and decode data into and from AX.25 frames.

### 7.1 Overview of the ADF7021-N

In this section the ADF7021-N will be explored a bit more in-depth. The interfaces used to configure and transfer data via the chip will be presented and explained then the values selected for each configuration register will be given.

#### 7.1.1 Hardware Block Diagram

A hardware block diagram of the ADF7021-N transceiver is shown in figure 7.1.

#### 7.1.2 Interfaces

The ADF7021-N have two interfaces that the microcontroller will have to use. One is used for the configuration of the transceiver and the other one is used for the data exchange via the radio link. There also are a few more GPIO that are used to complete these interfaces.

#### Configuration Interface

The configuration interface is used to change the configuration of the transceiver which is composed of 16 registers. It consists in four signals. The first three are the data line, the clock and the latch (respectively **SDATA**, **SCLK** and **SLE**) and are generated by the microcontroller. The last signal is **SREAD** and is generated by the chip when providing a read back. The way it works is that the microcontroller generates the clock and writes the bit it wants to transfer on **SDATA** on each falling edge. The transceiver reads the value on the rising edge of the clock. Once the master have sent the required data it must write a '1' on the latch for the chip to load the value to the selected register (the last 4 bits sent are used as the register address). When the microcontroller wants to read a value it must first write to register 7 (see section 7.1.3) but at the time of writing a '1' on **SLE** it must keep the signal as a '1' for longer while the chips clocks the requested data via the **SREAD** signal. The chips write the value of the bit on the rising edge of the clock (which must be provided by the microcontroller reading).

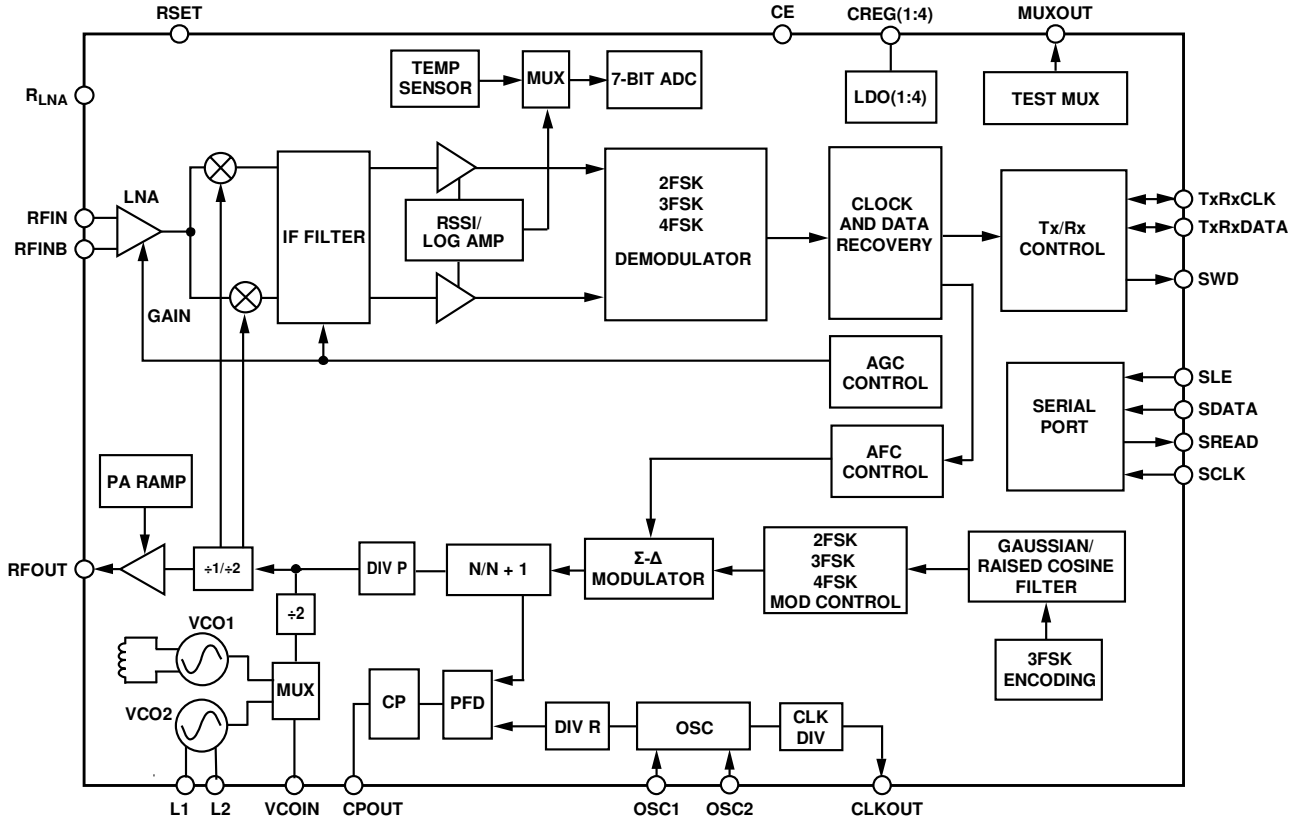


Figure 7.1: Hardware block diagram of the ADF7021-N transceiver.

The microcontroller must read the data on the falling edge of the clock. This is similar to an Serial Peripheral Interface (SPI) interface except for the behaviour of the SLE pin (and the lack of slave select signal).

## Data Interface

The data interface is the interface that allows to exchange data via the radio link configured on the chip. It consists in two signals. The first one is for the data (TXRXDATA) and the second one is for the clock (TXRXCLK). There are several ways to use that interface. The first way is as a Universal Asynchronous Receiver Transmitter (UART) bus with one line being used to send data and the other one being used to receive data. However this type of interface would require a UART peripheral to be used on the microcontroller. The interface could also be configured as an SPI interface with one signal providing the clock to the microcontroller while one signal is used to send data and a last signal is used to receive data. However the configuration that is used is the standard data interface proposed in the datasheet. One signal provides the clock while the other signal is used either to write or read data to or from the chip depending on the state it is configured in. As each transceiver will only work in one specific direction this interface is the simplest one to use as it is possible to simply attach an interrupt to be executed on specific transitions in the clock signal.

## General Purpose Input/Output

Several GPIO are also needed to interface with the transceiver.

The first one is the CE pin or Chip Enable pin. When a high level is written to that pin the ADF7021-N transceiver power-on and enables its subsystems.

The second one is the MUXOUT pin or Multiplexer Output pin. It can be configured via the configuration interface to output specific signals such as the status of the PLL (whether it is locked or not) or the state of the internal regulators (which indicates if the configuration interface is ready or not).

The third one is used only on the RX transceiver and is the SWD pin or Synchronisation Word Detect pin. Its exact behaviour can be configured but its role is to signal the microcontroller that

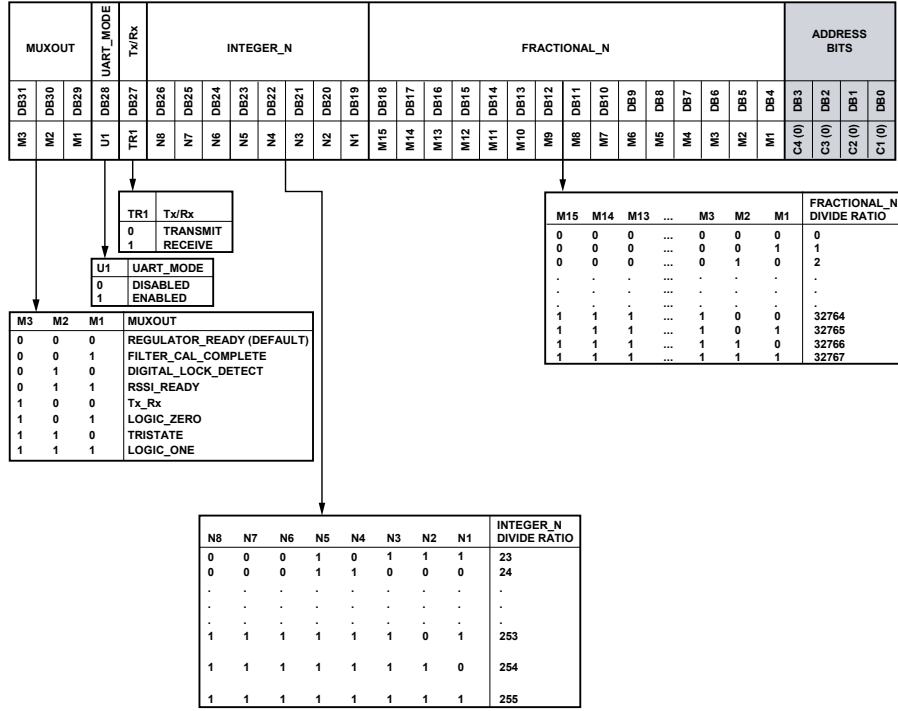


Figure 7.2: Content of the register 0 of the ADF7021-N.

a specific pattern was detected in the incoming RF signal. This allows the microcontroller to be interrupted to read the incoming data.

### 7.1.3 Configuration Registers

Sixteen different registers are available on each pin and dictate the behaviour of the pins. An explanation of the content of each of those registers will be made here. Some registers however only concern the RX or TX operations. Some other registers are intended to be used only when specific tests are being performed and are not used during regular operation of the transceiver. Most of the content of those registers was obtained via the *ADF7021-N Configuration Software* provided by Analog Devices and were checked afterwards for correctness.

#### Register 0 - N Register

Register 0 is used to configure the mode of operation of the MUXOUT pin, the mode of operation of the data interface, whether the transceiver is in Tx or Rx mode and holds the configuration of the fractional PLL.

**Rx** In RX the transceiver will update the mode of operation of MUXOUT according to its needs, the UART mode is disabled, the transceiver is in RX mode.

As the RF\_DIVIDE\_BY\_TWO (see register 1 below) is not enabled the configuration of the PLL is such that

$$RF_{OUT} = PFD \times \left( Integer_N + \frac{Fractional_N}{2^{15}} \right)$$

. However in order to account for the Intermediate Frequency (IF) frequency of 100 kHz it must be configured to be 100 kHz below the frequency at which the chip will be listening. It thus comes that  $Integer_N = 45$  and  $Fractional_N = 9950$  which gives

$$RF_{OUT} = 9.6 \times \left( 45 + \frac{9950}{32768} \right) = 434.91503 \text{ MHz}$$

The selected default value for register 0 in TX is thus 0x496A6DE0.

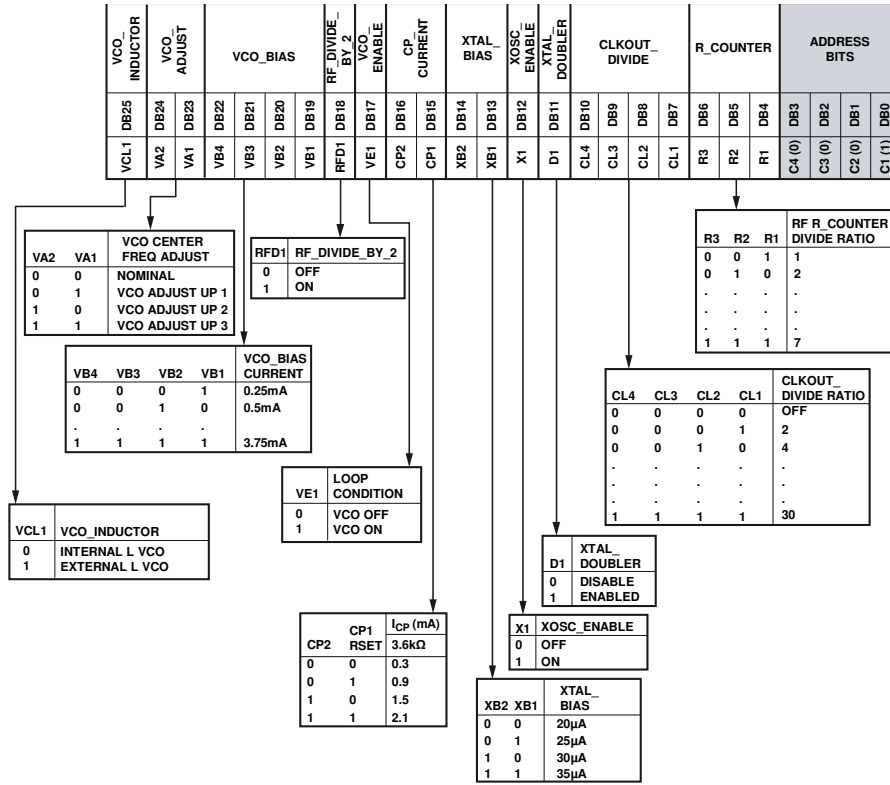


Figure 7.3: Content of the register 1 of the ADF7021-N.

**Tx** In RX the transceiver will update the mode of operation of MUXOUT according to its needs, the UART mode is disabled, the transceiver is in TX mode. As the RF\_DIVIDE\_BY\_TWO is enabled (see register 1 below) the configuration of the PLL is such that

$$RF_{OUT} = PFD \times \frac{1}{2} \times (Integer_N + \frac{Fractional_N}{2^{15}})$$

. It gives  $Integer_N = 30$  and  $Fractional_N = 13312$  for a frequency

$$RF_{OUT} = \frac{1}{2} \times 9.6 \times (30 + \frac{13312}{32768}) = 145.95 \text{ MHz}$$

.

The selected default value for register 0 in TX is thus 0x40F34000.

## Register 1 - Voltage Controlled Oscillator

Register 1 is used to configure the VCO whose goal is ultimately to generate the RF signal (or more exactly an RF signal at twice the actual RF frequency).

**Rx** In RX the transceiver operates with an external inductor for the VCO, at the nominal VCO center frequency and recommended VCO bias of 0.75 mA. RF\_DIVIDE\_BY\_TWO is disabled. The CP current is kept at the default value of 1.5 mA, the crystal bias is set at the recommended value of 30 μA. For the selected type of TCXO (0.8 Vpp clipped sinewave) the XOSC\_ENABLE setting must be set to 1. The crystal doubler is disabled. The CLKOUT signal is unused. R\_COUNTER sets the phase detector frequency PFD according to the formula

$$PDF[\text{Hz}] = \frac{XTAL[\text{Hz}]}{R\_COUNTER}$$

. It is recommended to use the lowest R\_COUNTER possible. In this case it must be set to 2 such that

$$PDF[\text{Hz}] = \frac{19.2 \text{ MHz}}{2} = 9.6 \text{ MHz}$$

.

The selected default value for register 1 in RX is thus 0x21B5021.

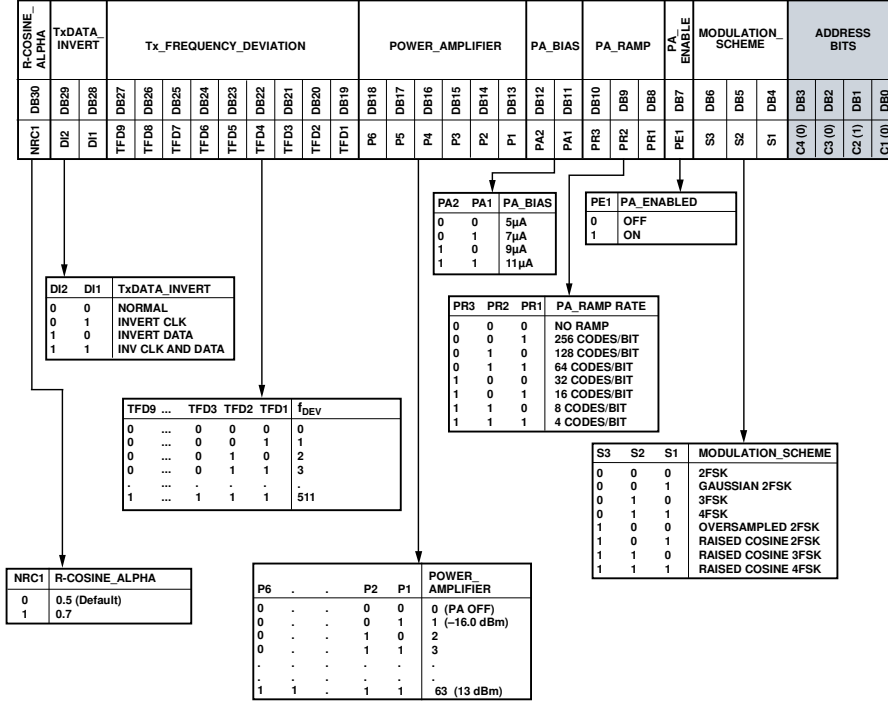


Figure 7.4: Content of the register 2 of the ADF7021-N.

**Tx** In TX the transceiver operates with an external inductor for the VCO. The nominal VCO center frequency was adjusted up to 3 when the VCO inductor was replaced by a 22 nH inductor (see section 5.4.3). The recommended VCO bias must be changed from the 0.75 mA value recommended by the *ADF7021-N Evaluation Software* down to a value of 0.5 mA (value given in the datasheet). RF\_DIVIDE\_BY\_TWO is enabled in order to access the lower RF frequency range of the transceiver. The CP current is kept at the default value of 1.5 mA, the crystal bias is set at the default value of 30 μA. For the selected type of TCXO (0.8 Vpp clipped sinewave) the XOSC\_ENABLE setting must be set to 1. The crystal doubler is disabled. The CLKOUT signal is unused.

R\_COUNTER sets the phase detector frequency PFD according to the formula

$$PDF[\text{Hz}] = \frac{XTAL[\text{Hz}]}{R\_COUNTER}$$

. It is recommended to use the lowest R\_COUNTER possible. In this case it must be set to 2 such that

$$PDF[\text{Hz}] = \frac{19.2 \text{ MHz}}{2} = 9.6 \text{ MHz}$$

The selected default value for register 1 in TX is thus 0x3975021.

## Register 2 - Tx Modulation

Register 2 contains the configuration for the modulation in transmission mode.

**Rx** This register is not used in reception.

**Tx** The transmitted signal is 2FSK modulated with a frequency deviation of 4800 Hz. Thus the R-CosineAlpha value does not matter. The data and clock signals are kept at their normal polarity. The frequency deviation must be configured such that

$$FrequencyDeviation[\text{Hz}] = \frac{1}{2} \times \frac{TX\_FREQUENCY\_DEVIATION \times PFD}{2^{16}}$$

thus with a value of 66 the frequency deviation is

$$FrequencyDeviation[\text{Hz}] = \frac{1}{2} \times \frac{66 \times 9.6 \text{ MHz}}{65536} = 4834 \text{ Hz}$$

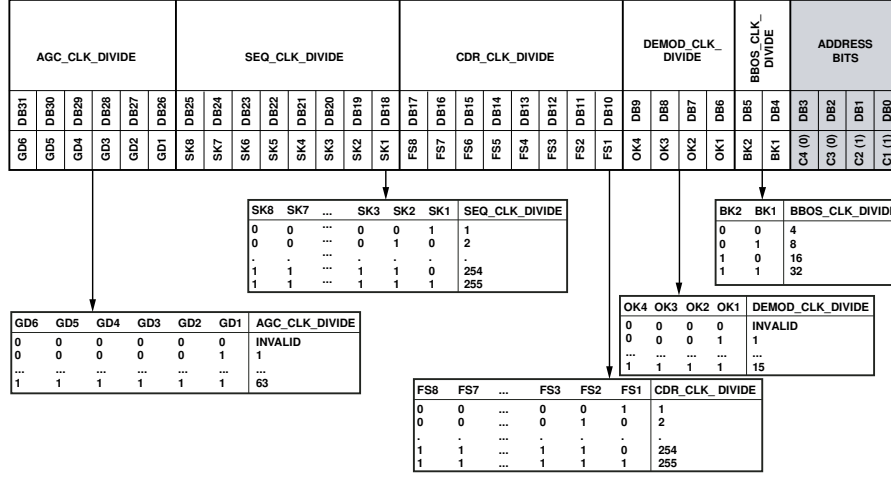


Figure 7.5: Content of the register 3 of the ADF7021-N.

In order to have a power of 30.5 dB at the output of the external PA the power at the output of the ADF7021-N must be  $-3.1$  dBm. This implies choosing a setting for the power amplifier of 8 and a bias of  $5\mu\text{A}$  (the bias allows finer control over the gain of the PA of the ADF7021-N). The PA ramping time will be set to the maximum as it allows to reduce any transient phenomenon or spurious emissions that could occur while turning it on or off. The PA is of course enabled and the modulation scheme is 2FSK.

The selected default value for register 2 in TX is thus 0x2110782.

### Register 3 - Tx/Rx Clock

Register 3 contains the configuration for a number of clocks which are then used by other subsystems. Because the two transceivers use the same reference frequency crystals and the same baud rates, the configuration of these registers will be the same for both transceivers.

SEQ\_CLK\_DIVIDE sets the frequency of the sequencer and is used in the receiver part of the transceiver. It should be set such that the frequency is as close as possible to 100 kHz following the formula

$$SEQ\ CLK[\text{Hz}] = \frac{XTAL[\text{Hz}]}{SEQ\_CLK\_DIVIDE}$$

. SEQ\_CLK\_DIVIDE is thus set to 192 for a SEQ CLK clock frequency of exactly 100 kHz (the crystal frequency XTAL is 19.2 MHz).

AGC\_CLK\_DIVIDE sets the frequency of the clock used in the Automatic Gain Control (AGC) subsystem such that the frequency must be as close as possible to 8 kHz following the formula

$$AGC\ CLK[\text{Hz}] = \frac{SEQ\ CLK[\text{Hz}]}{AGC\_CLK\_DIVIDE}$$

. AGC\_CLK\_DIVIDE is thus set to 13 for an AGC CLK frequency of 7692 Hz.

DEMOD\_CLK\_DIVIDE sets the frequency of the clock used in the demodulation subsystem such that the frequency must be within the range 2 MHz to 15 MHz following the formula

$$DEMOD\ CLK[\text{Hz}] = \frac{XTAL[\text{Hz}]}{DEMOD\_CLK\_DIVIDE}$$

. DEMOD\_CLK\_DIVIDE is set to 2 for a DEMOD CLK frequency of  $\frac{19.2\text{MHz}}{2} = 9.6$  MHz.

CDR\_CLK\_DIVIDE sets the frequency of the CDR subsystem and must be set to be within 2% of  $(32 \times \text{data rate})$ . With the data rate being  $9600\text{ bs}^{-1}$  it must be in the range 301 056 Hz to 313 344 Hz. The formula to follow is

$$CDR\ CLK[\text{Hz}] = \frac{DEMOD\ CLK[\text{Hz}]}{CDR\_CLK\_DIVIDE}$$

. CDR\_CLK\_DIVIDE is thus set to 31 for a CDR CLK frequency of 309 677 Hz.





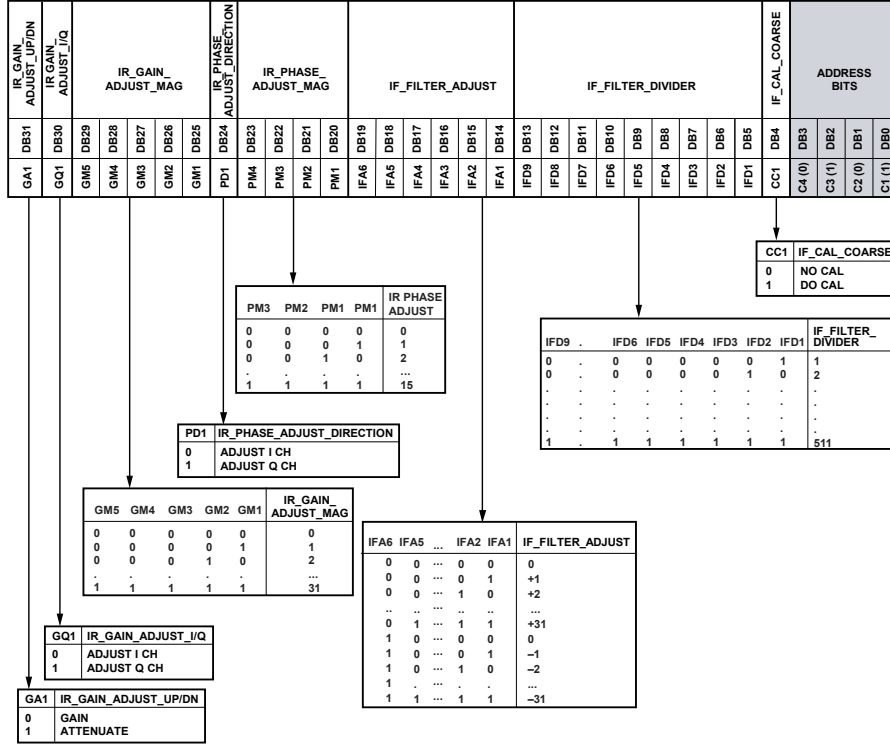


Figure 7.7: Content of the register 5 of the ADF7021-N.

## Register 5 - IF Filter Setup

The register 5 contains the configuration for the IF filter including the configuration allowing a good image rejection.

**Rx** The parameters `IR_GAIN_ADJUST_UP/DN`, `IR_GAIN_ADJUST_I/Q` and `IR_PHASE_ADJUST_DIRECTION` are used to improve the performances of the image rejection (see section 7.1.4 for more details on image rejection).

The parameter `IF_FILTER_ADJUST` is not used here because a fine calibration will need to be performed at regular intervals given the temperature range under which the nanosatellite will be operating.

`IF_CAL_COARSE` will be written to 1 as this is what starts the internal IF calibration procedure. `IF_FILTER_DIVIDE` must be chosen such that

$$\frac{XTAL[Hz]}{IF\_FILTER\_DIVIDE} = 50 \text{ kHz}$$

. This gives an `IF_FILTER_DIVIDE` value of 384 for a frequency of  $\frac{19.2\text{MHz}}{384} = 50 \text{ kHz}$  exactly.

The selected default value for register 5 in RX is thus 0x3015.

**Tx** The IF filter is not used in transmission and this register is thus not used in TX.

## Register 6 - IF Fine Calibration

Register 6 contains configuration bits which are used to configure the fine calibration of the IF filter as well as perform the image rejection calibration.

**Rx** `IR_CAL_SOURCE_DIV_2` and `IR_CAL_SOURCE_DRIVE_LEVEL` are used during the image rejection calibration procedure. More details about the image rejection procedure are given in section 7.1.4.

`IF_FINE_CAL` controls whether the device will perform only a coarse or a coarse and a fine calibration the next time it will perform an IF filter calibration (that is the next time the parameter `IF_CAL_COARSE` from register 5 is set to 1). It is set to 1 to perform a fine calibration.

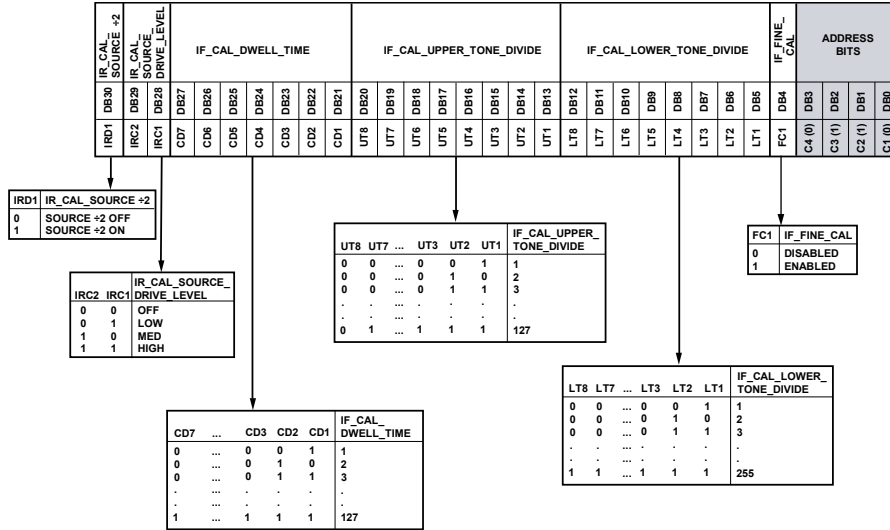


Figure 7.8: Content of the register 6 of the ADF7021-N.

IF\_CAL\_DWELL\_TIME sets the time given to the transceiver to perform its calibration and must be set such that the time is grater than 500  $\mu$ s. The formula to use is

$$IF \text{ Tone Calibration Time} = \frac{IF\_CAL\_DWELL\_TIME}{SEQ\_CLK}$$

- The value used is 80 for a calibration time of  $\frac{80}{100\text{kHz}} = 800 \mu\text{s}$ .

IF\_CAL\_UPPER\_TONE\_DIVIDE controls the higher tone frequency for the IF fine calibration and given that the IF bandwidth used is 18.5 kHz must be chosen such that the tone is at 119 kHz. The formula is

$$Upper \text{ Tone Frequency}[\text{Hz}] = \frac{XTAL[\text{Hz}]}{IF\_CAL\_UPPER\_TONE\_DIVIDE \times 2}$$

- The value is thus set at 81 for an upper tone frequency of  $\frac{19.2\text{MHz}}{2 \times 81} = 118.52 \text{ kHz}$ .

IF\_CAL\_LOWER\_TONE\_DIVIDE controls the lower tone frequency for the IF fine calibration and given that the IF bandwidth used is 18.5 kHz must be chosen such that the tone is at 78.1 kHz. The formula is

$$Lower \text{ Tone Frequency}[\text{Hz}] = \frac{XTAL[\text{Hz}]}{IF\_CAL\_LOWER\_TONE\_DIVIDE \times 2}$$

- The value is thus set at 123 for an upper tone frequency of  $\frac{19.2\text{MHz}}{2 \times 123} = 78.05 \text{ kHz}$ .

The selected default value for register 6 in RX is thus 0xA0A2F76.

**Tx** The IF filter is not used in transmission and this register is thus not used in TX.

## Register 7 - Readback

Register 7 is not a typical configuration register in that its content does not modify the behaviour of the transceiver. The value must be written to the register as for any other register but at the time of latching the data in the register the SLE signal must be kept for a time during which the transceiver will clock the requested data out. The read back can contain several types of data: the AFC word which contains information on the AFC, the output of the internal Analogue to Digital Converter (ADC), the value the chip found while calibrating the IF filter or the silicon revision. The ADC can be configured to measure either the RSSI, the battery voltage (which is the voltage supplied to the chip), the temperature of the chip or the voltage applied on the external ADC pin. This register is mostly used to measure the RSSI.

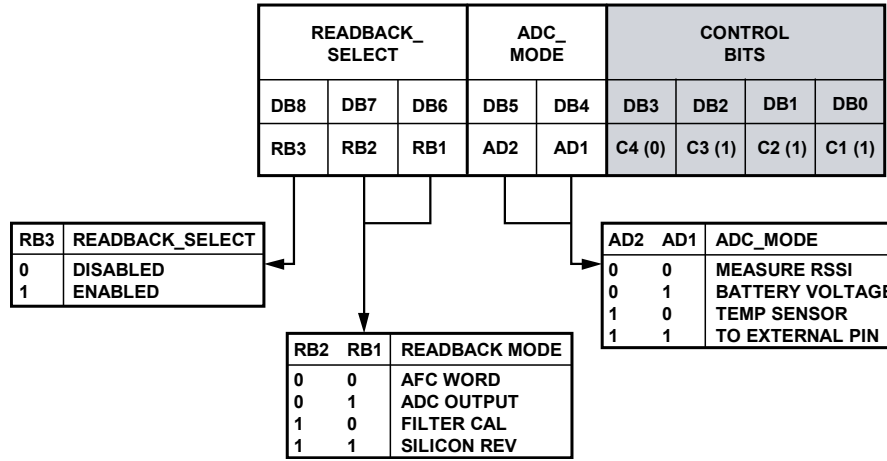


Figure 7.9: Content of the register 7 of the ADF7021-N.

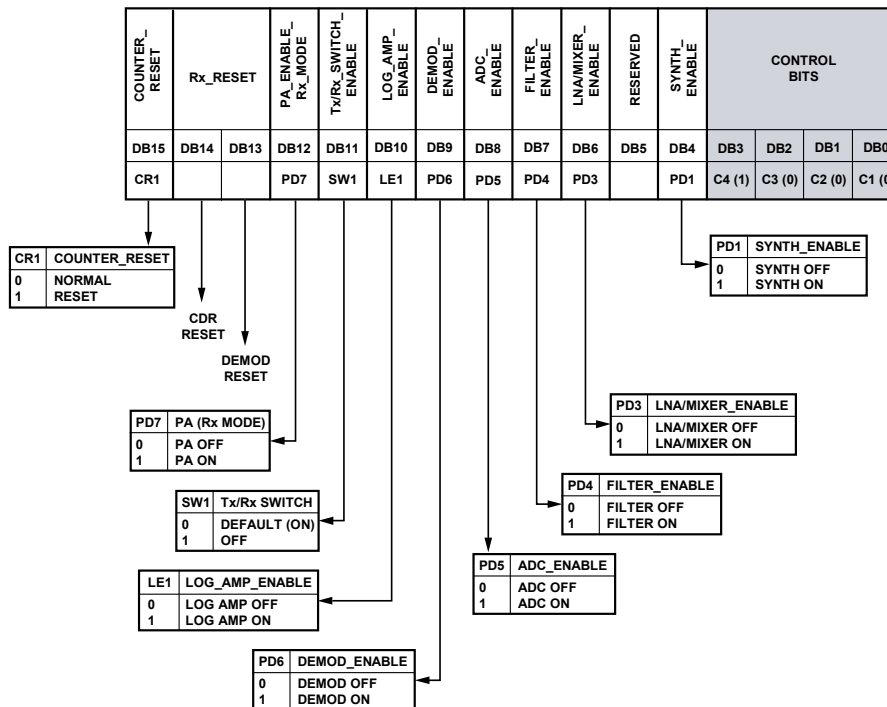


Figure 7.10: Content of the register 8 of the ADF7021-N.

## Register 8 - Power Down Test

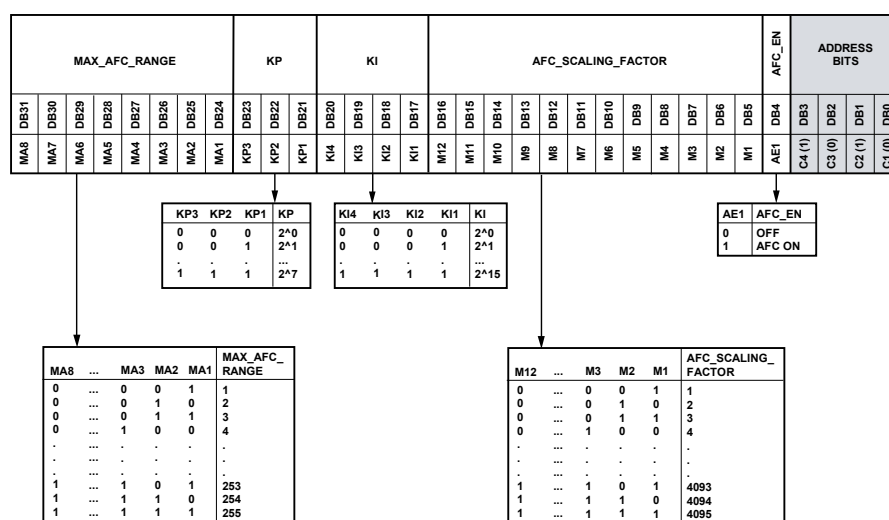
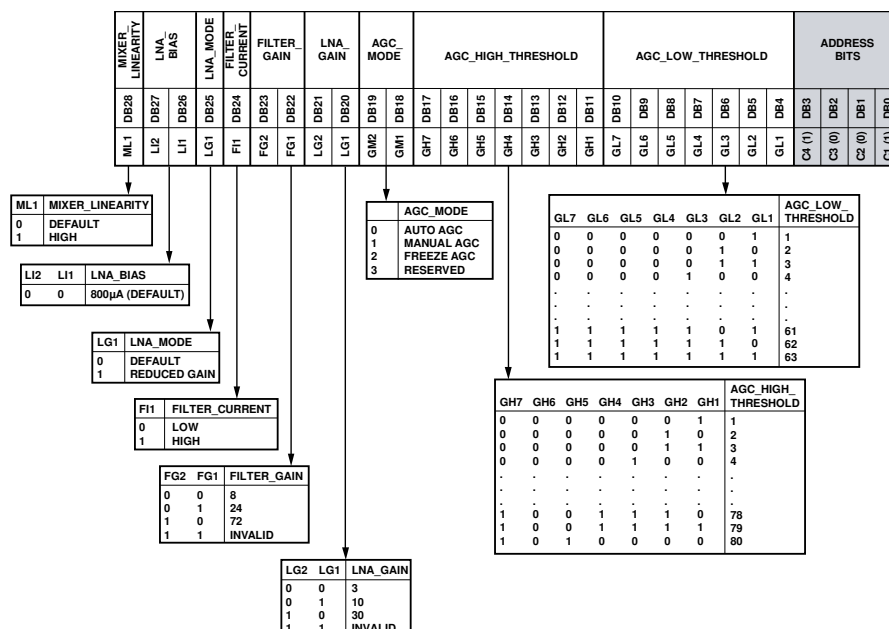
Register 8 contains configuration bits which can be used to power down some subsystems of the transceiver.

**Rx** In RX mode the subsystems that can be powered down are the PA and the Tx/Rx switch (the LNA and PA are matched separately).

The selected default value for register 8 in RX is thus 0xFD8.

**Tx** In TX mode the subsystems that can be powered down are the Tx/Rx switch (the LNA and PA are matched separately), the log amplifier (used to measure the RSSI in reception), the demodulator, the ADC (which is powered-down by default in TX mode and must be powered-on to be used in TX mode), the filter and the LNA/Mixer.

The selected default value for register 8 in TX is thus 0x1818.



## Register 9 - Automatic Gain Control

**Rx** The automatic gain control is left in the default mode. However this registers allow to fine-tune AGC behaviour and may be used if for example it was decided to increase the linearity of the transceiver by lowering the maximum LNA gain (that is the sensitivity).

## Register 10 - Automatic Frequency Correction

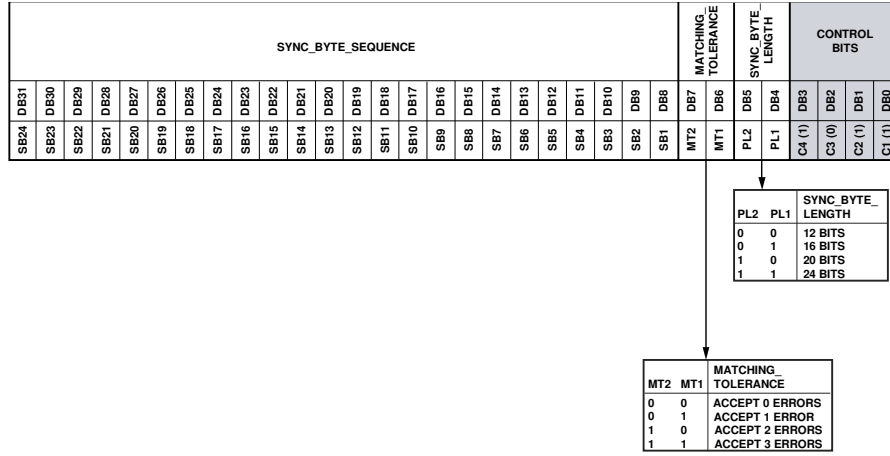


Figure 7.13: Content of the register 11 of the ADF7021-N.

**Rx** The values used in this register are the values recommended by Analog Devices.

The MAX\_AFC\_RANGE is set to 50 meaning that the AFC is able to correct a frequency error in a range of  $50 \times 500 \text{ Hz} = 25 \text{ kHz}$ <sup>1</sup> (the datasheet recommends a range of  $1.5 \times IF \text{ bandwidth}$ ). That means that the incoming signal can have an error of  $\frac{25 \text{ kHz}}{2} = \pm 12.5 \text{ kHz}$ .

The values for KI and KP are the recommended values of 11 and 4.

AFC\_SCALING\_FACTOR must be set according to the formula

$$AFC\_SCALING\_FACTOR = Round\left(\frac{2^{24} \times 500}{XTAL[Hz]}\right)$$

. This gives an AFC\_SCALING\_FACTOR value of 437.

AFC\_EN must be set to 1 to enable the AFC loop.

The selected default value for register 10 in RX is thus 0x329636BA.

**Tx** This register is not used in emission.

### Register 11 - Sync Word Detection

Register 11 contains the configuration bits for the SWD.

**Rx** The synchronisation word that is used in the chip is 0x555555. No error will be tolerated on the synchronisation word and its length is 24 bits.

The selected default value for register 11 in RX is thus 0x5555553B.

**Tx** This register is not used in emission.

### Register 12 - SWD Setup

Register 12 contains the configuration bits that configures the way the transceiver must behave upon detection of a synchronisation word.

**Rx** When the device will detect a synchronisation word it must set the SWD pin to high until it gets reset. The DATA\_PACKET\_LENGTH is not used so any value can be used. The device should also lock its AGC loop upon detection of a synchronisation word.

The selected default value for register 12 in RX is thus 0x15C.

**Tx** This register is not used in emission.

<sup>1</sup>However the IF filter have a bandwidth of 18.5 kHz meaning that the AFC may not be able to correct for the frequency error if the signal is at a level so low that after being attenuated by the IF filter it is under the sensitivity level.

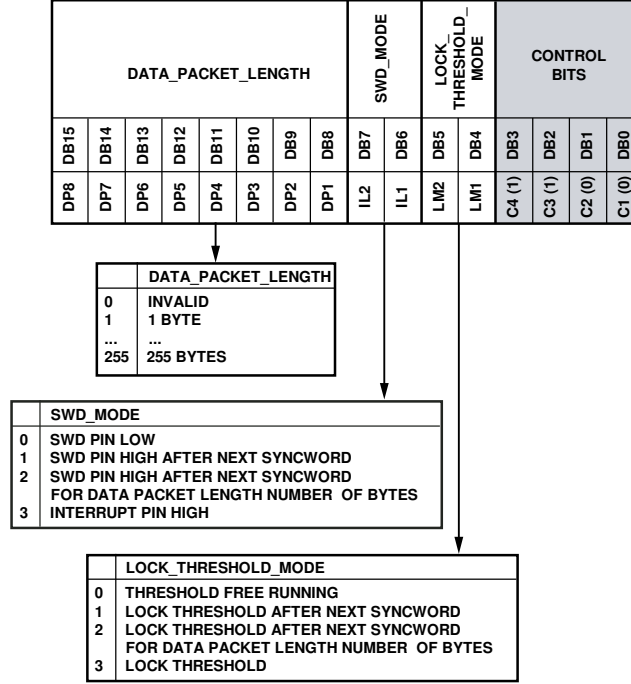


Figure 7.14: Content of the register 12 of the ADF7021-N.

### Register 13 - 3FSK/4FSK demodulation

As the module is designed to operate in 2FSK this register is not used neither in reception nor in emission.

### Register 14 - Test DAC

This register is only used for very specific tests or debugs situation and was not used in this work, neither in reception nor in emission.

### Register 15 - Test Mode

This register is not intended for use in normal operation of the transceiver. However the TX\_TEST\_MODES was used while testing the performances of the transceiver to request that it sends a pure tone at the carrier frequency, a "101010" data pattern or PN9<sup>2</sup> data.

#### 7.1.4 Image Rejection

The ADF7021-N is, in RX mode, a superheterodyne receiver. That means that it brings the signal at the listening frequency ( $f_{sig}$ ) down to an intermediate frequency ( $f_{if} = 100\text{kHz}$  in the case of the ADF7021-N) by mixing it with a reference frequency ( $f_{lo}$ ) which is  $f_{if}$  below  $f_{sig}$ . This however creates a situation where a signal that is present in the spectrum at  $f_{img} = f_{lo} - f_{if}$  will also be present at  $f_{if}$  once mixed with the reference frequency. The frequency at  $f_{img}$  is called the image frequency and for a frequency  $f_{sig} = 435.015\text{ MHz}$  its value is  $f_{img} = 434.815\text{ MHz}$  (see figure 7.16).

This transceiver however uses different I/Q mixers which enables it to then perform an image rejection filtering with a polyphase IF filter (see figure 7.17). This kind of filter however is reliant on matching I and Q paths. As a perfect match between the two is never possible a mechanism is implemented in the chip to correct for any mismatch [23].

To compensate for that the procedure consists in generating a tone inside the transceiver at the image frequency (434.815 MHz) and then adjust a series of parameters (in register 5) so that the RSSI read back is as small as possible (meaning that the signal is attenuated as most as possible).

<sup>2</sup>PN9 data is a specific bit-sequence that is used in some radio systems to asses its performances using an identical data sequence from test to test.

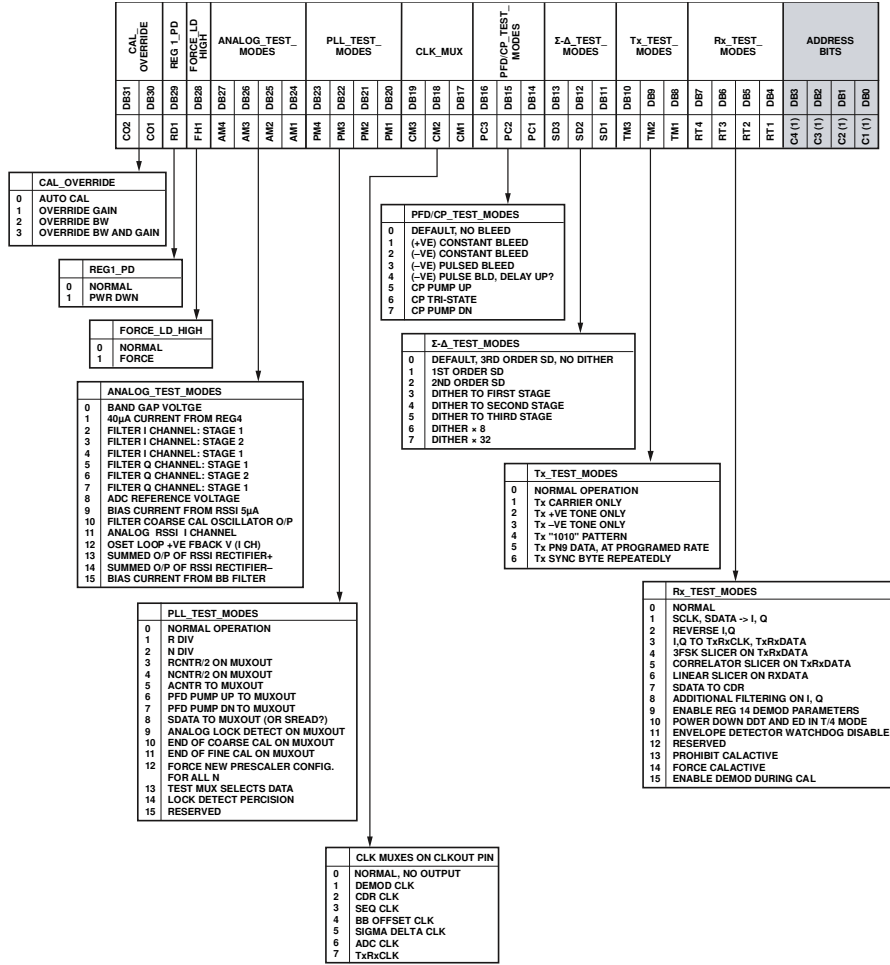


Figure 7.15: Content of the register 15 of the ADF7021-N.

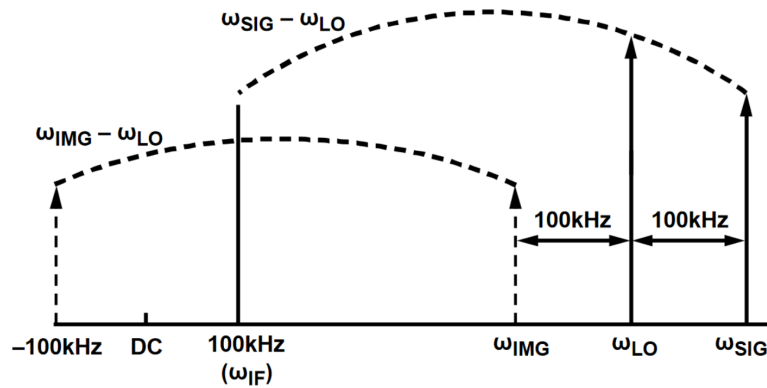


Figure 7.16: Representation of the source of the image signal problem.  $\omega_{SIG}$  is the frequency of the signal of interest (435.015 MHz),  $\omega_{LO}$  is the frequency of the local oscillator (434.915 MHz),  $\omega_{IMG}$  is the frequency of the image signal (434.815 MHz).





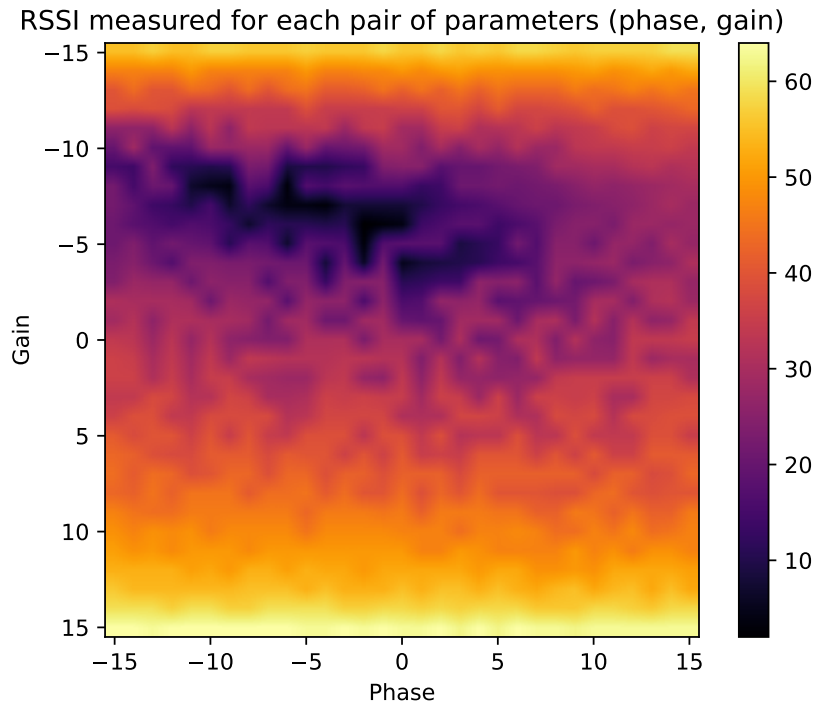


Figure 7.18: Representation of the RSSI levels read for each pair of values (gain, phase) possible. A local minimum can be seen around (gain=-4, phase=0) while a lower value can be measured at (gain=-7, phase=-6). The vertical axis is in ADC counts.

- CMSIS Pack, System, Device Family Pack, Board Support Package. Those 4 components allow to configure the IO, interrupts, clocks and peripherals of the chip,...
- FreeRTOS. This component represent the FreeRTOS real-time operating system and allows to configure many of its parameters.
- Core. This component is required by other components and allow them to register their services in FreeRTOS.
- Debug. This component allows to print debug messages to the virtual serial port during execution of the code.
- Command. This component allows to input commands from the serial port basically transforming it into a small shell. Commands can later be registered in the code to trigger some actions from the serial terminal.
- USART1 and Console. Those two components are required for the debug and command components and allows those components to interface properly with the hardware serial peripheral of the SAM V71 chip.
- TC3. This components implements some code to interface more easily with the timer and counter number 3 of the SAM V71 chip. It will be used to create short (sub-millisecond) delays during execution of the code.

### 7.3 Written Code

The code that have been written in order to implement the AX.25 communication protocol on the OBC consists of three main parts.

The first part is the ADF7021-N Hardware Abstraction Layer (HAL) and the goal of this code is to provide an easy-to-use interface to the ADF7021-N transceiver.

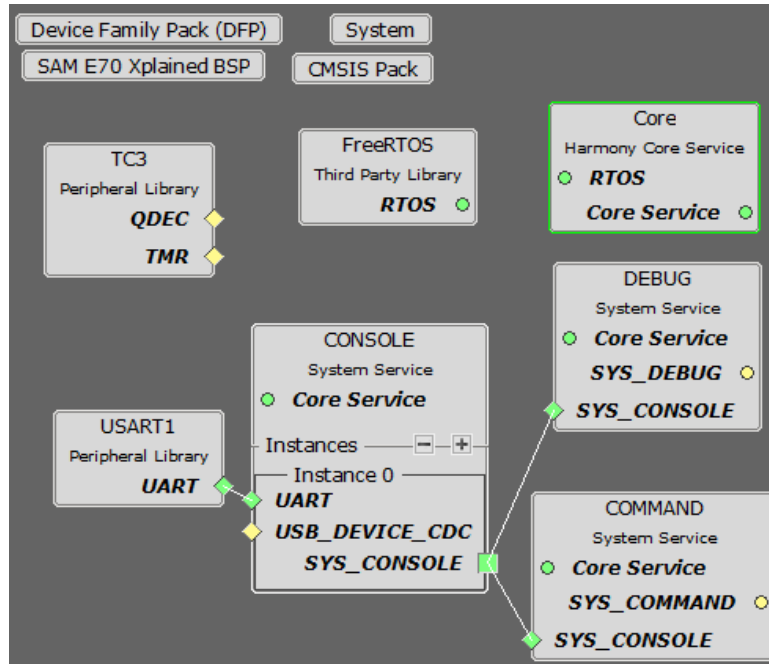


Figure 7.19: *Harmony 3* project graph.

The second part contains the data structure representing an AX.25 packet as well as the functions needed to interact with those packets and perform the required encodings, bit stuffing, scrambling,...

The third part is composed of several FreeRTOS tasks which perform the required operations to send and receive data via the radio link.

The written code is available on GitLab<sup>3</sup>

### 7.3.1 Code for the ADF7021-N

One of the role of the HAL is to provide an interface to the configuration interface of the transceiver. It is implemented via two functions for each transceiver. One is used to send a configuration register and the other one is used to perform a read back.

Because this interface is mainly used when starting and configuring a transceiver this code will not get executed often and given that the SPI interface may be already attributed to other subsystems of the nanosatellite and that the behaviour of the SLE pin is not an SPI feature it was decided to implement this interface in a sequential fashion. That means that the microcontroller will successively write a '0' on the clock signal and write the value of the bit on the SDATA signal, wait a fixed delay, write a '1' on the clock signal and wait a fixed delay then write a '0' on the clock signal and so on until all the bits have been sent. Once all the bits of the registers have been sent it writes a '1' on the SLE signal, wait a fixed delay, and resets the interface. While doing so may be seen as timely inefficient, it is not performed often by the microcontroller. Furthermore the actual time taken to send a complete configuration register is very small. The clock signal was chosen to be 5 MHz (corresponding to a fixed delay of 100 ns) (the datasheet of the transceiver specifies a maximum frequency of 20 MHz and does not specify a minimum frequency) [13]. That means that for writing a 32-bits register (the maximum length for a register) the time required is  $33 * 200 \text{ ns} = 6600 \text{ ns} = 6.6 \mu\text{s}$ . Furthermore the fact that the chip does not specify a minimum frequency allows the task sending the configuration register to be preempted while sending data. This just means that the signals will be kept in a frozen status for longer than needed but it will not have any side-effect. This was implemented in a specific function for each transceiver (as there are no slave-select signal the same pins could not be shared between the transceivers).

For the data interface there is not a lot that can be implemented. The code will just attach an interrupt to the clock pin allowing it to sample or write to the TXRXDATA line when needed. The microcontroller can then enable or disable the execution of the Interrupt Service Routine (ISR) so

<sup>3</sup>The GitLab repository is available at <https://gitlab.uliege.be/mdiepart/oufti2-commboard>

that it can simply ignore the clock signal when it should not decode the incoming data or when no data must be sent.

Finally functions to start and configure each transceiver were implemented. In order to start the RX transceiver the microcontroller must first power it on (via the **CE** pin) and wait for the **MUXOUT** pin to indicate that the internal regulators are ready. Once the configuration interface become available register 1 is programmed and the task waits for 1 ms for the VCO to settle. Registers 3, 6 and 5 are then programmed for the clocks and calibrations, and the task once again waits (for 10 ms for the IF calibration to complete. After that the registers 11 and 12 are programmed to configure the SWD. Register 01 is programmed and the task waits for **MUXOUT** to signal that the PLL is locked. Finally the registers 4, 10, 8 and 9 are programmed to configure the demodulator, the AFC, the AGC and power-down the unused subsystems.

In order to start the TX transceiver a similar procedure is followed. The microcontroller first turns on the chip (via the **CE** pin) and wait for the **MUXOUT** pin to indicate that the internal regulators are ready. Once the configuration interface become available register 1 is programmed and the task waits for 1 ms for the VCO to settle. Register 3 and 0 are programmed for the clocks and the PLL and the task waits for **MUXOUT** to signal that the PLL is locked. Finally the registers 2 is programmed for the TX modulation. On the second board, the TCXO and the external PA would need to be enabled too.

### 7.3.2 AX.25 Packet

In order to implement the AX.25 protocol many steps have to be followed such as encoding, scrambling, perform bit-stuffing,... (see section 2.2.1). It was decided to create a data structure that would perform all those operations seamlessly for the rest of the code. The data structure contains an array of bytes which represent the data the packet is supposed to hold. This is the data, byte-by-byte, without any processing or Cyclic Redundancy Code (CRC) appended. The structure then contains a number of internal states, counters, flags and registers to keep track of the actual state of the transmission of that packet. Thanks to this approach a very limited number of functions must be used to interact with an AX.25 packet.

In transmission, the packet must first be created with the dedicated function and then be given the data it must contain. The code then have to call a function to deserialize the packet bit-by-bit which forms a stream that represents the packet once all the encoding, scrambling, flags operations, bit-stuffing,... are performed. This function must be called successively until all the bits were extracted from the packet at which point the function simply returns a special value which basically represents an "end of stream" condition. The packet must then be deleted using the appropriate function.

In reception the packet must also be created using the same function as in the transmission case. The incoming stream of bits can then be deserialized by inserting the raw bits into the data structure. The function will automatically synchronise the LFSR register, wait for the preamble flag and perform all the unscrambling, decoding, destuffing,... required to get the data contained in the packet. The status of the packet is updated in real-time allowing to interrupt the transmission if an error was detected. Once the postamble flag is detected the packet is complete and the code can get the content of the packet by calling a single function which will return the contained data if the CRC check completes successfully. If the CRC cannot be verified no data is returned.

### 7.3.3 FreeRTOS Tasks

The FreeRTOS tasks represent the core of the business. There are responsible for using the components presented previously. One exist for each direction (TX and RX). One last task was implemented to allow to test the data in both direction with minimal effort.

#### Reception

The reception tasks is responsible for the RX transceiver and the handling of the AX.25 protocol in reception.

The internal data structure of the task contains a queue with received packets and a pointer to a callback function to be called upon reception of a valid AX.25 packet.

At first the tasks must initialise. For that it starts by powering-up and configuring the RX transceiver (the nanosatellite must always be in reception) then performs a calibration of the image rejection parameters of the ADF7021-N. After that it enables interrupts on the SWD pin.

In the main loop of the task is first waits until a packet gets pushed on the internal queue. When a packet is available in the queue it pops it and checks if the packet is valid. If it is not it silently discards it and waits for the next packet. If the packet is valid it checks if a callback is registered to be called upon reception of a valid packet. If there is one it calls it otherwise it silently discards the packet.

The ISR that gets executed upon a rising edge of the SWD pin first checks if the task is ready to receive a packet and if so creates an active AX.25 packet, enables interrupts on the TXRXCLK pin and disables interrupts on the SWD pin.

The ISR that gets executed upon a rising edge of the TXRXCLK first samples the TXRXDATA and then inputs that bit into the active AX.25 packet. It then checks if the current packet is in an error state or an end of packet state. If there is an error it discards the packet. If the packet is finished it pushes it on the queue and starts listening for a new packet (it could be that two or more packets are sent in a row being separated only by two HDLC flags and no synchronisation word).

The tasks also registers two commands to the command task implemented by *Harmony 3*.

The first commands performs an RSSI measurement averaged over a given number of samples and prints the result to the serial terminal.

The second command is specifically used to test the internal algorithms used for the reception, decoding, unscrambling, de-stuffing of the incoming data. After having executed the command the user will be able to send a stream of ones and zeros as if it was data being sent via the radio link. This stream of data will be decoded using the same functions as an incoming AX.25 packet would. If the packet is valid its data will be displayed in the terminal.

## Transmission

The transmission tasks is responsible for the TX transceiver.

The internal data structure of the task contains a queue with the packets to send as well as some parameters such as the TX\_DELAY parameter.

The main function that must be used by external code is a function that allows to insert data in the transmission queue. The function takes a byte array and creates a corresponding AX.25 packet to be placed in the transmission queue. Another function allows to configure the delays between the power-on of the radio and the start of the first transmission as well as between the last bit of the last transmission and the power-off of the radio.

The main loop of the task wait until there is a packet in the transmission queue. When there is a packet it powers the radio on and starts transmitting the first packet. If after having transmitted the packet there are other packets in the queue it will transmit them. When the queue is empty it will shutdown the radio and return to the state where it wait for a packet to be pushed in the queue.

The task also registers four commands to the command task implemented by *Harmony 3*.

The first two functions are to turn the transmission chain on and off (including the power amplifier).

Another function allows to send a packet via the radio. It is mainly used to test the algorithms used to encode, scramble, bit-stuff the data to send and will either write the resulting bit stream in the terminal or will send the data via the radio link.

The last function allows to configure the delays before and after the transmission.

## Parrot

The Parrot tasks is not really a tasks required in order to use the AX.25 radio link. This task was simply written such that the radio could be tested very simply. It receives incoming AX.25 packet from the RX transceiver and sends them back via the TX transceiver.

The architecture of the task is quite simple. At the initialisation it creates a queue holding AX.25 packets and registers an internal function to be called by the Reception task upon reception of a valid AX.25 packet. This function thus pushes the data received on the queue when a valid packet is received.

The main loop of the task simply wait until there is a packet in the queue. When there is, it pops the packet from the queue and asks to the Transmission task to send an AX.25 packet containing that data.

# Conclusion

It is finally time to conclude what have been performed before and through this work. Even though a lot of practical analysis and results are presented here, many things went on behind the scenes.

It have been an excellent opportunity to learn about the specificities of radio systems, the methods of modelisation and analysis as well as the measurements tools. This is not covered in the typical cursus of an electrical engineer at the university of Liège but is yet primordial when it comes to radio systems design.

It was also an opportunity to perform a screening of the current states of satellite projects in the university of Liège. This allowed to apprehend what is needed to manage a project as large as building a full CubeSat. There is still a long way to go before another spacecraft gets launched by students of the university but now the picture gets clearer about the legacy knowledge that can be relied upon, as well as the topics that will need more exploration.

In what concern the telecommunication system, the analysis of the ground station highlighted the main problems that were met during the design of a the first satellite in a hope to address them. Solutions combining flexible software components and versatile hardware were imagined and assembled to a solid proof-of-concept. This solution even allowed to develop a new telecommunication module and test it.

The module itself have been redesigned from the ground up. All the components were analysed. The transceivers and power amplifier were confirmed to be good choices. Their performances were measured and match perfectly with the result that can be expected from them. Even though the low noise amplifier could not be confirmed to be working the complete module have been designed anyway and is yet to be tested. If it turned out that the low noise amplifier still was not operational the work needed to replace it with another solution is small and would not be a major obstacle.

To go further and with the rigour that the development of a spacecraft requires several other measurements that could not be performed during this work should be done. It includes testing the sensitivity of the reception with higher precision as well as the performances of the system with varying signal to noise ratios. The board must also be tested in a space-like environment especially in what concerns the pressure and temperature. It is also advised to stress-test the module with many successive transmissions to ensure that the design will not be damaged by repeated power cycles. Especially for the most sensitive parts such as the power amplifier. Other tests including very long continuous transmissions could also be performed to ensure that the module will not be damaged by its own dissipated power once in space-like conditions.

## Appendix A

# Link Budget

An important set of parameters to know before designing a telecommunication system is the expected link budget.

The link budget available from the previous OUFTI1 mission is based on an Excel spreadsheet made by the AMSAT Radio Amateur Satellite Corporation [24]. This was discovered quite late in this work and some parameters used for the calculations are identical to the default data in the spreadsheet which could indicate that some parameters were not adjusted to the hardware used for the mission.

Among those parameters is the bandwidth used to compute the uplink noise margin. It was set to 9600 Hz which correspond to twice the frequency deviation used in the modulation scheme. However the calculation of the link budget according to the SNR method must use the bandwidth of the filter at the input of the transceiver as this is filter that will limit the noise input. This filter must be selected according to the OBW of the emitted signal. In section 4.5 the OBW of the uplink signal will be measured, setting the value of the input filter of the transceiver. The new bandwidth to use is thus 18.5 kHz. This is about twice the previously used bandwidth which means that twice the input noise level is expected which should translate to a SNR decreased by about 3 dB. The SNR decreased from 17.9 dB to 15 dB as expected. The result stays very good.

It is also to be noted that the Ground Station Transmitter Output Power is set at 100 W and it is not known if this is the power that is used at the ground station or not. If it is the power used it is not known what is the limiting factor in the transmission chain. However the maximum transmission power allowed by the Institut Belge des Postes et Télécommunications is set to 1500 W and the power amplifier used is capable of outputting a signal at a maximum power of 550 W.

Other parameters such as the noise temperature at the spacecraft's receiver were oddly selected and might not reflect the actual performances of the system.

However a summary of the link budget is still made available here. The bandwidth in the uplink budget (SNR method) is updated. The rest of the parameters is kept intact.

OUFTI		NOTE:
Uplink Command Budget:		
Parameter:	Value:	Units:
<b>Ground Station:</b>		
Ground Station Transmitter Power Output:	100,0 watts	
In dBW:	20,0	dBW
In dBm:	50,0	dBm
Ground Stn. Total Transmission Line Losses:	2,8 dB	
Antenna Gain:	15,0 dBi	
Ground Station EIRP:	32,2	dBW
<b>Uplink Path:</b>		
Ground Station Antenna Pointing Loss:	0,2 dB	
Gnd-to-S/C Antenna Polarization Losses:	3,0 dB	
Path Loss:	153,3 dB	
Atmospheric Losses:	2,1 dB	
Ionospheric Losses:	0,7 dB	
Rain Losses:	0,0 dB	
Isotropic Signal Level at Spacecraft:	-127,1	dBW
<b>Spacecraft (Eb/No Method):</b>		
----- Eb/No Method -----		
Spacecraft Antenna Pointing Loss:	0,0 dB	
Spacecraft Antenna Gain:	0,0 dBi	
Spacecraft Total Transmission Line Losses:	0,9 dB	
Spacecraft Effective Noise Temperature:	239 K	
Spacecraft Figure of Merit (G/T):	-24,7 dB/K	
S/C Signal-to-Noise Power Density (S/No):	76,9	dBHz
System Desired Data Rate:	9600	bps
In dBHz:	39,8	dBHz
Command System Eb/No:	37,1	dB
Demodulation Method Seleted:	G3RUH FSK	
Forward Error Correction Coding Used:	None	
System Allowed or Specified Bit-Error-Rate:	1,0E-04	
Demodulator Implementation Loss:	1,0	dB
Telemetry System Required Eb/No:	16,7	dB
Eb/No Threshold:	17,7	dB
<b>System Link Margin:</b>	<b>19,4</b>	<b>dB</b>
<b>Spacecraft Alternative Signal Analysis Method (SNR Computation):</b>		
----- SNR Method -----		
Spacecraft Antenna Pointing Loss:	0,0 dB	
Spacecraft Antenna Gain:	0,0 dBi	
Spacecraft Total Transmission Line Losses:	0,9 dB	
Spacecraft Effective Noise Temperature:	239 K	
Spacecraft Figure of Merit (G/T):	-24,7 dB/K	
Signal Power at Spacecraft LNA Input:	-127,9	dBW
Spacecraft Receiver Bandwidth:	18 500	Hz
Spacecraft Receiver Noise Power (Pn = kTB)	-162,1	dBW
Signal-to-Noise Power Ratio at G.S. Rcvr:	34,2	dB
Analog or Digital System Required S/N:	19,0	dB
System Link Margin	15,2	dB

(a) Uplink Budget

OUFTI		NOTE:
Downlink Telemetry Budget:		
Parameter:	Value:	Units:
<b>Spacecraft:</b>		
Spacecraft Transmitter Power Output:	0,7 watts	
In dBW:	-1,5	dBW
In dBm:	28,5	dBm
Spacecraft Total Transmission Line Losses:	0,7 dB	
Spacecraft Antenna Gain:	0,0 dBi	
Spacecraft EIRP:	-2,3	dBW
<b>Downlink Path:</b>		
Spacecraft Antenna Pointing Loss:	0,0 dB	
S/C-to-Ground Antenna Polarization Loss:	3,0 dB	
Path Loss:	143,8 dB	
Atmospheric Loss:	2,1 dB	
Ionospheric Loss:	0,7 dB	
Rain Loss:	0,0 dB	
Isotropic Signal Level at Ground Station:	-151,9	dBW
<b>Ground Station (Eb/No Method):</b>		
----- Eb/No Method -----		
Ground Station Antenna Pointing Loss:	0,1 dB	
Ground Station Antenna Gain:	12,7 dBi	
Ground Station Total Transmission Line Losses:	1,6 dB	
Ground Station Effective Noise Temperature:	209 K	
Ground Station Figure of Merit (G/T):	-12,1 dB/K	
G.S. Signal-to-Noise Power Density (S/No):	64,4	dBHz
System Desired Data Rate:	9600	bps
In dBHz:	39,8	dBHz
Telemetry System Eb/No for the Downlink:	24,6	dB
Demodulation Method Seleted:	User Defined	
Forward Error Correction Coding Used:	None	
System Allowed or Specified Bit-Error-Rate:	1,0E-03	
Demodulator Implementation Loss:	1	dB
Telemetry System Required Eb/No:	9,6	dB
Eb/No Threshold:	10,6	dB
<b>System Link Margin:</b>	<b>14,0</b>	<b>dB</b>
<b>Ground Station Alternative Signal Analysis Method (SNR Computation):</b>		
----- SNR Method -----		
Ground Station Antenna Pointing Loss:	0,1 dB	
Ground Station Antenna Gain:	12,7 dBi	
Ground Station Total Transmission Line Losses:	1,6 dB	
Ground Station Effective Noise Temperature:	209 K	
Ground Station Figure of Merit (G/T):	-12,1 dB/K	
Signal Power at Ground Station LNA Input:	-141,0	dBW
Ground Station Receiver Bandwidth (B):	10 000	Hz
G.S. Receiver Noise Power (Pn = kTB)	-165,4	dBW
Signal-to-Noise Power Ratio at G.S. Rcvr:	24,4	dB
Analog or Digital System Required S/N:	16,6	dB
System Link Margin	7,8	dB

(b) Downlink Budget



## Appendix B

# Configuration file used for Dire Wolf Software TNC

---

```
1 #####
2 #
3 #           Configuration file for Dire Wolf           #
4 #
5 #           Linux version                               #
6 #
7 #####
8 #
9 # Extensive documentation can be found here:
10 # Stable release -      https://github.com/wb2osz/direwolf/tree/
11 #                     master/doc
12 # Latest development -  https://github.com/wb2osz/direwolf/tree/dev/
13 #                     doc
14 #
15 # The complete documentation set can also be found in
16 # /usr/local/share/doc/direwolf/ or /usr/share/doc/direwolf/
17 # Concise "man" pages are also available for Linux.
18 #
19 # This sample file does not have examples for all of the possibilities
20 #
21 # Consult the User Guide for more details on configuration options.%C
22 #
23 # These are the most likely settings you might change:
24 #
25 #           (1)          MYCALL - call sign and SSID for your station.
26 #
27 #                               Look for lines starting with MYCALL and
28 #                               change NOCALL to your own.
29 #
30 #           (2)          PBEACON - enable position beaconing.
31 #
32 #                               Look for lines starting with PBEACON and
33 #                               modify for your call, location, etc.
34 #
35 #           (3)          DIGIPEATER - configure digipeating rules.
36 #
37 #                               Look for lines starting with DIGIPEATER.
38 #                               Most people will probably use the given
39 #                               example.
40 #
41 #                               Just remove the "#" from the start of the line
```

```

37 #                                     to enable it.
38 #
39 #          (4)          IGSERVER, IGLOGIN  - IGate server and login
40 #
41 #                                     Configure an IGate client to relay messages
         between
42 #                                     radio and internet servers.
43 #
44 #
45 # The default location is "direwolf.conf" in the current working
         directory.
46 # On Linux, the user's home directory will also be searched.
47 # An alternate configuration file location can be specified with the
         "-c" command line option.
48 #
49 # As you probably guessed by now, # indicates a comment line.
50 #
51 # Remove the # at the beginning of a line if you want to use a sample
52 # configuration that is currently commented out.
53 #
54 # Commands are a keyword followed by parameters.
55 #
56 # Command key words are case insensitive.  i.e. upper and lower case
         are equivalent.
57 #
58 # Command parameters are generally case sensitive.  i.e. upper and
         lower case are different.
59 #
60
61
62 #####
63 #                                     #
64 #          FIRST AUDIO DEVICE PROPERTIES          #
65 #          (Channel 0 + 1 if in stereo)          #
66 #                                     #
67 #####
68
69 #
70 # Many people will simply use the default sound device.
71 # Some might want to use an alternative device by choosing it here.
72 #
73 # Linux ALSA is complicated.  See User Guide for discussion.
74 # To use something other than the default, generally use plughw
75 # and a card number reported by "arecord -l" command.  Example:
76
77 #N.B. the software loopback device does not have to be hw:1,0 on every
         computer.
78 ADEVICE hw:1,0
79
80 # You can also use "-" or "stdin" to pipe stdout from
81 # some other application such as a software defined radio.
82 # "stdin" is not an audio device.  Don't use this unless you
83 # understand what this means.  Read the User Guide.
84 # You can also specify "UDP:" and an optional port for input.

```

```

85 # Something different must be specified for output.
86
87 #
88 # Number of audio channels for this soundcard: 1 (mono) or 2 (stereo).
89 # 1 is the default so there is no need to specify it.
90 #
91
92 #ACHANNELS 2
93
94
95 #####
96 #
97 #          SECOND AUDIO DEVICE PROPERTIES
98 #          (Channel 2 + 3 if in stereo)
99 #
100 #####
101
102 #ADEVICE1 ...
103
104
105 #####
106 #
107 #          THIRD AUDIO DEVICE PROPERTIES
108 #          (Channel 4 + 5 if in stereo)
109 #
110 #####
111
112 #ADEVICE2 ...
113
114
115 #####
116 #
117 #          CHANNEL 0 PROPERTIES
118 #
119 #####
120
121 CHANNEL 0
122
123 #
124 # The following MYCALL, MODEM, PTT, etc. configuration items
125 # apply to the most recent CHANNEL.
126 #
127
128 #
129 # Station identifier for this channel.
130 # Multiple channels can have the same or different names.
131 #
132 # It can be up to 6 letters and digits with an optional ssid.
133 # The APRS specification requires that it be upper case.
134 #
135 # Example (don't use this unless you are me): MYCALL WB2OSZ-5
136 #
137
138 MYCALL ON4ULG

```

```

139
140 ##
141 # Options added to suit channel settings from OUFTI1 TNC's
    configuration
142 ##
143
144 #DWAIT 10 msec per unit. First wait for DWAIT time. Normally zero.
    Used in very specific situations
145 DWAIT 0
146 #SLOTTIME 10 msec per unit
147 SLOTTIME 20
148 #PERSIST propability for transmitting after each slottime
149 PERSIST 128
150 #TXDELAY 10 msec per unit
151 TXDELAY 100
152 #TXTAIL 10 msec per unit (TNC actually transmits 2 bytes. 10 msec is
    12 flags
153 TXTAIL 1
154 #FULLDUP OFF=half-duplex ON=full-duplex.
155 #We work on full duplex because LimeSDR is able to
156 FULLDUP ON
157
158
159 # Pick a suitable modem speed based on your situation.
160 #      1200      Most common for VHF/UHF. Default if not specified.
161 #      2400      QPSK compatible with MFJ-2400, and probably PK232-2400
    & KPC-2400.
162 #      300      Low speed for HF SSB. Default tones 1600 & 1800.
163 #      EAS      Emergency Alert System (EAS) Specific Area Message
    Encoding (SAME).
164 #      9600      G3RUH style - Can't use Microphone and Speaker
    connections.
165 #      AIS      International system for tracking ships on VHF.
166 #                  Also uses 9600 bps so Speaker connection won't
    work.
167 #
168 # In most cases you can just specify the speed. Examples:
169 #
170
171 MODEM 9600
172
173 #
174 # Many options are available for great flexibility.
175 # See User Guide for details.
176 #
177
178 #
179 # Uncomment line below to enable the DTMF decoder for this channel.
180 #
181
182 #DTMF
183
184 #
185 # If not using a VOX circuit, the transmitter Push to Talk (PTT)

```

```

186 # control is usually wired to a serial port with a suitable interface
    circuit.
187 # DON'T connect it directly!
188 #
189 # For the PTT command, specify the device and either RTS or DTR.
190 # RTS or DTR may be preceded by "-" to invert the signal.
191 # Both can be used for interfaces that want them driven with opposite
    polarity.
192 #
193 # COM1 can be used instead of /dev/ttyS0, COM2 for /dev/ttyS1, and so
    on.
194 #
195
196 #PTT COM1 RTS
197 #PTT COM1 RTS -DTR
198 #PTT /dev/ttyUSB0 RTS
199
200 #
201 # On Linux, you can also use general purpose I/O pins if
202 # your system is configured for user access to them.
203 # This would apply mostly to microprocessor boards, not a regular PC.
204 # See separate Raspberry Pi document for more details.
205 # The number may be preceded by "-" to invert the signal.
206 #
207
208 #PTT GPIO 25
209
210 # The Data Carrier Detect (DCD) signal can be sent to the same places
211 # as the PTT signal. This could be used to light up an LED like a
    normal TNC.
212
213 #DCD COM1 -DTR
214 #DCD GPIO 24
215
216
217 #####
218 #                                                                 #
219 #             CHANNEL 1 PROPERTIES                               #
220 #                                                                 #
221 #####
222 #Channel 1 is not used.
223 #CHANNEL 1
224
225 #
226 # Specify MYCALL, MODEM, PTT, etc. configuration items for
227 # CHANNEL 1. Repeat for any other channels.
228
229
230 #####
231 #                                                                 #
232 #             TEXT TO SPEECH COMMAND FILE                         #
233 #                                                                 #
234 #####
235

```



```

289 #           OBEACON           - for an object report (usually some other
    entity)
290 #
291 # Each has a series of keywords and values for options.
292 # See User Guide for details.
293 #
294 # Example:
295 #
296 # This results in a broadcast once every 10 minutes.
297 # Every half hour, it can travel via two digipeater hops.
298 # The others are kept local.
299 #
300
301 #PBEACON delay=1  every=30 overlay=S symbol="digi" lat=42^37.14N long
    =071^20.83W power=50 height=20 gain=4 comment="Chelmsford MA" via=
    WIDE1-1,WIDE2-1
302 #PBEACON delay=11 every=30 overlay=S symbol="digi" lat=42^37.14N long
    =071^20.83W power=50 height=20 gain=4 comment="Chelmsford MA"
303 #PBEACON delay=21 every=30 overlay=S symbol="digi" lat=42^37.14N long
    =071^20.83W power=50 height=20 gain=4 comment="Chelmsford MA"
304
305
306 # With UTM coordinates instead of latitude and longitude.
307
308 #PBEACON delay=1  every=10 overlay=S symbol="digi" zone=19T easting
    =307477 northing=4720178
309
310
311 #
312 # When the destination field is set to "SPEECH" the information part
    is
313 # converted to speech rather than transmitted as a data frame.
314 #
315
316 #CBEACON dest="SPEECH" info="Club meeting tonight at 7 pm."
317
318 # Similar for Morse code.  If SSID is specified, it is multiplied
319 # by 2 to get speed in words per minute (WPM).
320
321 #CBEACON dest="MORSE-6" info="de MYCALL"
322
323
324 #
325 # Modify for your particular situation before removing
326 # the # comment character from the beginning of appropriate lines
    above.
327 #
328
329
330 #####
331 #
332 #           APRS DIGIPEATER PROPERTIES
333 #
334 #####

```

```

335
336 #
337 # For most common situations, use something like this by removing
338 # the "#" from the beginning of the line below.
339 #
340
341 #DIGIPEAT 0 0 ^WIDE[3-7]-[1-7]$|^TEST$ ^WIDE[12]-[12]$ TRACE
342
343 # See User Guide for more explanation of what this means and how
344 # it can be customized for your particular needs.
345
346 # Filtering can be used to limit was is digipeated.
347 # For example, only weather weather reports, received on channel 0,
348 # will be retransmitted on channel 1.
349 #
350
351 #FILTER 0 1 t/wn
352
353 # Traditional connected mode packet radio uses a different
354 # type of digipeating. See User Guide for details.
355
356 #####
357 #
358 # INTERNET GATEWAY #
359 # #
360 #####
361
362 # First you need to specify the name of a Tier 2 server.
363 # The current preferred way is to use one of these regional rotate
    addresses:
364
365 # noam.aprs2.net - for North America
366 # soam.aprs2.net - for South America
367 # euro.aprs2.net - for Europe and Africa
368 # asia.aprs2.net - for Asia
369 # aunz.aprs2.net - for Oceania
370
371 #IGSERVER noam.aprs2.net
372
373 # You also need to specify your login name and passcode.
374 # Contact the author if you can't figure out how to generate the
    passcode.
375
376 #IGLOGIN WB2OSZ-5 123456
377
378 # That's all you need for a receive only IGate which relays
379 # messages from the local radio channel to the global servers.
380
381 # Some might want to send an IGate client position directly to a
    server
382 # without sending it over the air and relying on someone else to
383 # forward it to an IGate server. This is done by using sendto=IG
    rather

```



```

384 # than a radio channel number. Overlay R for receive only, T for two
    way.
385
386 #PBEACON sendto=IG delay=0:30 every=60:00 symbol="igate" overlay=R lat
    =42^37.14N long=071^20.83W
387 #PBEACON sendto=IG delay=0:30 every=60:00 symbol="igate" overlay=T lat
    =42^37.14N long=071^20.83W
388
389
390 # To relay messages from the Internet to radio, you need to add
391 # one more option with the transmit channel number and a VIA path.
392
393 #IGTXVIA 0 WIDE1-1
394
395
396 # Finally, we don't want to flood the radio channel.
397 # The IGate function will limit the number of packets transmitted
398 # during 1 minute and 5 minute intervals. If a limit would
399 # be exceeded, the packet is dropped and message is displayed in red.
400
401 IGTXLIMIT 6 10
402
403
404 #####
405 #
406 # APRStt GATEWAY #
407 #
408 #####
409
410 #
411 # Dire Wolf can receive DTMF (commonly known as Touch Tone)
412 # messages and convert them to packet objects.
413 #
414 # See separate "APRStt-Implementation-Notes" document for details.
415 #
416
417 #
418 # Sample gateway configuration based on:
419 #
420 # http://www.aprs.org/aprstt/aprstt-coding24.txt
421 # http://www.aprs.org/aprs-jamboree-2013.html
422 #
423
424 # Define specific points.
425
426 TTPOINT B01 37^55.37N 81^7.86W
427 TTPOINT B7495088 42.605237 -71.34456
428 TTPOINT B934 42.605237 -71.34456
429
430 TTPOINT B901 42.661279 -71.364452
431 TTPOINT B902 42.660411 -71.364419
432 TTPOINT B903 42.659046 -71.364452
433 TTPOINT B904 42.657578 -71.364602
434

```

```

435
436 # For location at given bearing and distance from starting point.
437
438 TTVECTOR B5bbbdd 37^55.37N 81^7.86W 0.01 mi
439
440 # For location specified by x, y coordinates.
441
442 TTGRID Byyyxxx 37^50.00N 81^00.00W 37^59.99N 81^09.99W
443
444 # UTM location for Lowell-Dracut-Tyngsborough State Forest.
445
446 TTUTM B6xxxxyy 19T 10 300000 4720000
447
448
449
450 # Location for the corral.
451
452 TTCORRAL 37^55.50N 81^7.00W 0^0.02N
453
454 # Compact messages - Fixed locations xx and object yy where
455 # Object numbers 100 - 199 = bicycle
456 # Object numbers 200 - 299 = fire truck
457 # Others = dog
458
459 TTMACRO xx1yy B9xx*AB166*AA2B4C5B3B0A1yy
460 TTMACRO xx2yy B9xx*AB170*AA3C4C7C3B0A2yy
461 TTMACRO xxyyy B9xx*AB180*AA3A6C4A0Ayyy
462
463 TTMACRO z Cz
464
465 # Receive on channel 0, Transmit object reports on channel 1 with
    optional via path.
466 # You probably want to put in a transmit delay on the APRStt channel
    so it
467 # it doesn't start sending a response before the user releases PTT.
468 # This is in 10 ms units so 100 means 1000 ms = 1 second.
469
470 #TTOBJ 0 1 WIDE1-1
471 #CHANNEL 0
472 #DWAIT 100
473
474 # Advertise gateway position with beacon.
475
476 # OBEACON DELAY=0:15 EVERY=10:00 VIA=WIDE1-1 OBJNAME=WB20SZ-tt SYMBOL=
    APRStt LAT=42^37.14N LONG=71^20.83W COMMENT="APRStt Gateway"
477
478
479 # Sample speech responses.
480 # Default is Morse code "R" for received OK and "?" for all errors.
481
482 #TTERR OK SPEECH Message Received.
483 #TTERR D_MSG SPEECH D not implemented.
484 #TTERR INTERNAL SPEECH Internal error.
485 #TTERR MACRO_NOMATCH SPEECH No definition for digit sequence.

```

486	#TTERR	BAD_CHECKSUM	SPEECH	Bad checksum on call.
487	#TTERR	INVALID_CALL	SPEECH	Invalid callsign.
488	#TTERR	INVALID_OBJNAME	SPEECH	Invalid object name.
489	#TTERR	INVALID_SYMBOL	SPEECH	Invalid symbol.
490	#TTERR	INVALID_LOC	SPEECH	Invalid location.
491	#TTERR	NO_CALL	SPEECH	No call or object name.
492	#TTERR	SATSQ	SPEECH	Satellite square must be 4 digits.
493	#TTERR	SUFFIX_NO_CALL	SPEECH	Send full call before using suffix.

---

## Appendix C

# LNA Comparison

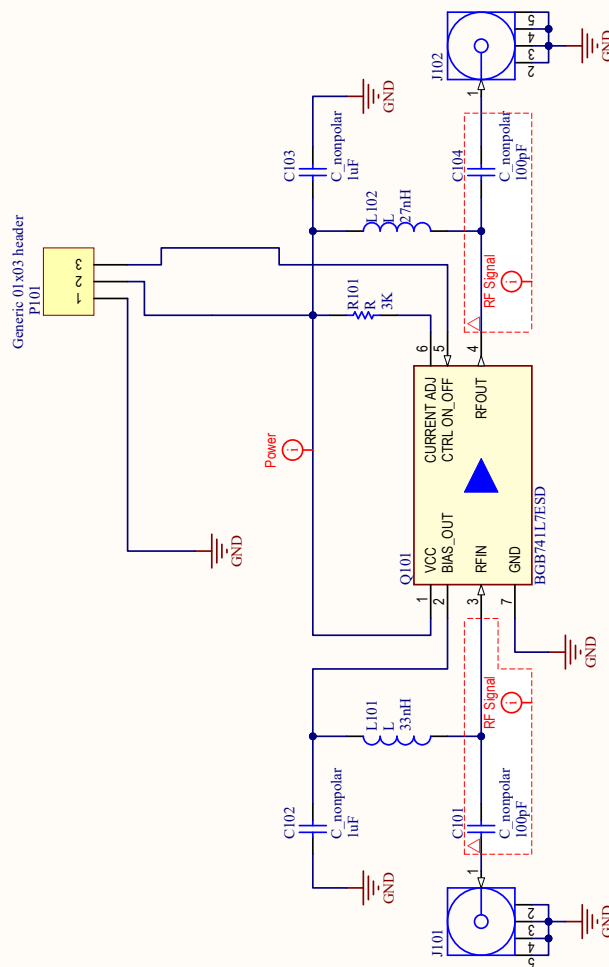
Low Noise Amplifier Comparison Sheet						
Manufacturer	Analog Devices	RFMD	Infineon	NXP	Mini-Circuits	Notes:
Part Number	ADL5523	RF2878	BGB741L7ESD	BGU6104	TSY-13LNB+	
Bandwidth	400-4000 MHz	150-2500 MHz	50-3500 MHz	40-4000 MHz	30-1000 MHz	
Test Frequency	900 MHz	350 MHz	450 MHz	450 MHz	500 MHz	
Supply Voltage	3 V	3,5 V	3 V	3 V	2,7 V	
Supply Current	30 mA	7 mA	10 mA	12 mA	7,7 mA	
Supply Power	90 mW	24,5 mW	30 mW	36 mW	20,79 mW	Important criteria: LNA is expected to be powered on at all times
Gain	21 dB	25,5 dB	20,5 dB	27,5 dB	14,7 dB	
Noise Figure	0,8 dB	1,75 dB	0,95 dB	0,8 dB	1,2 dB	The SNR margin in the link budget allows even for the worse device NF
Input Third Order Intercept (IIP3)	Higher is better	4,5 dBm	2,5 dBm			Was used to compute OIP3 in case only IIP3 was given in the datasheet
Output Third Order Intercept (OIP3)	Higher is better	30 dBm	23 dBm	17 dBm	26,4 dBm	
Input 1dB Compression Point	Higher is better		-7,5 dBm			Was used to compute P1dB in case only input 1dB compression point was given in datasheet
Output 1dB Compression Point	Higher is better	15,6 dBm	12 dBm	5,5 dBm	17,1 dBm	All expected power levels should be way below the lowest P1dB
Maximal RF Input Level	Higher is better	10 dBm	20 dBm	NA	10 dBm	
Internal Bias	YES is better	Yes	Yes	Yes	Yes	Will simplify the design
Choice		No	Yes	Yes	Yes	

## Appendix D

# Test Circuits Schematics

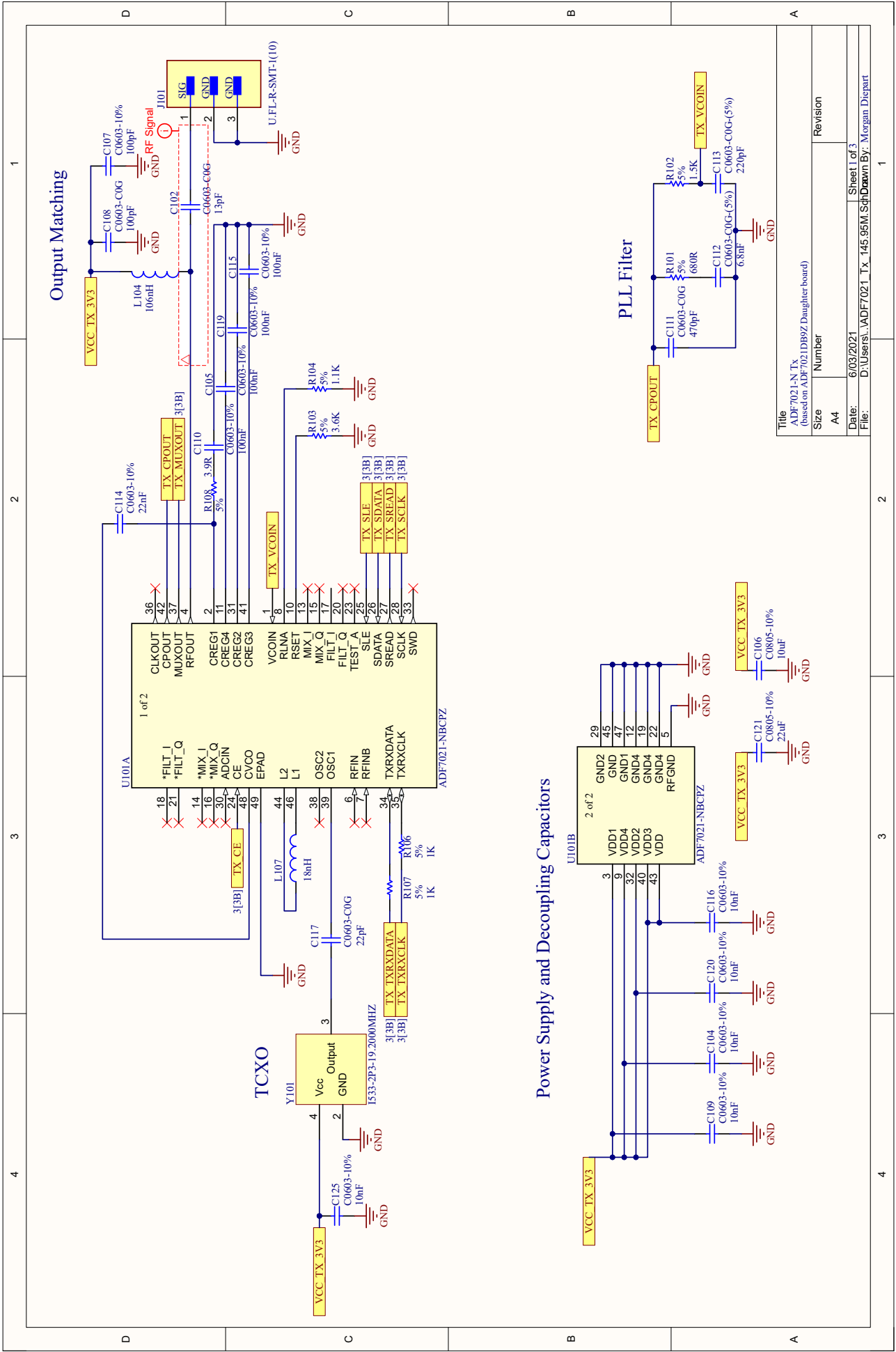


## D.2 Low Noise Amplifier Test Circuit



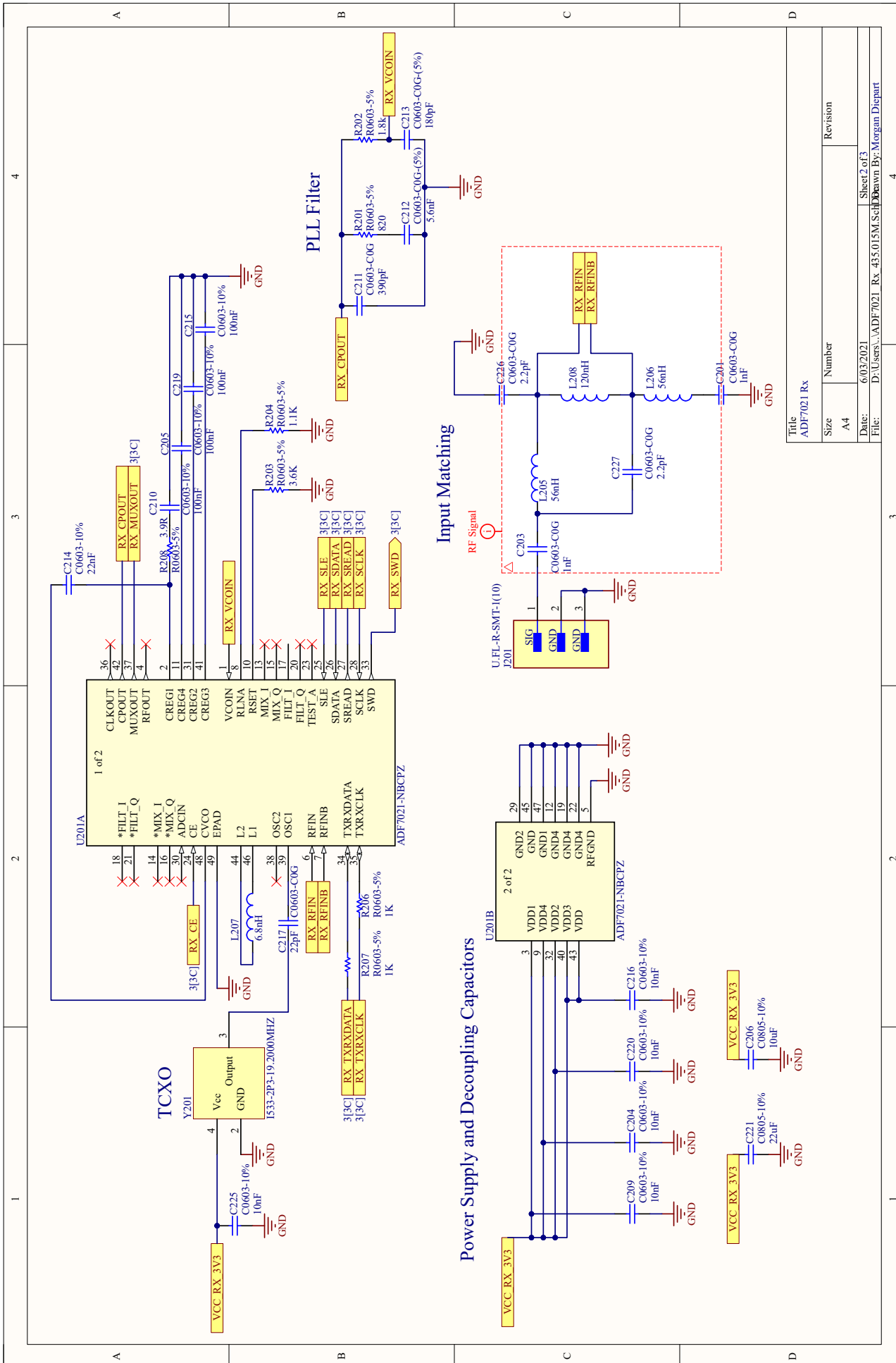
Title	
BGB741L7ESD Test Board	
Size	Number
A4	
Date:	6/03/2021
File:	D:\Users\...BGB74

# D.3 Tx Transceiver Test Circuit

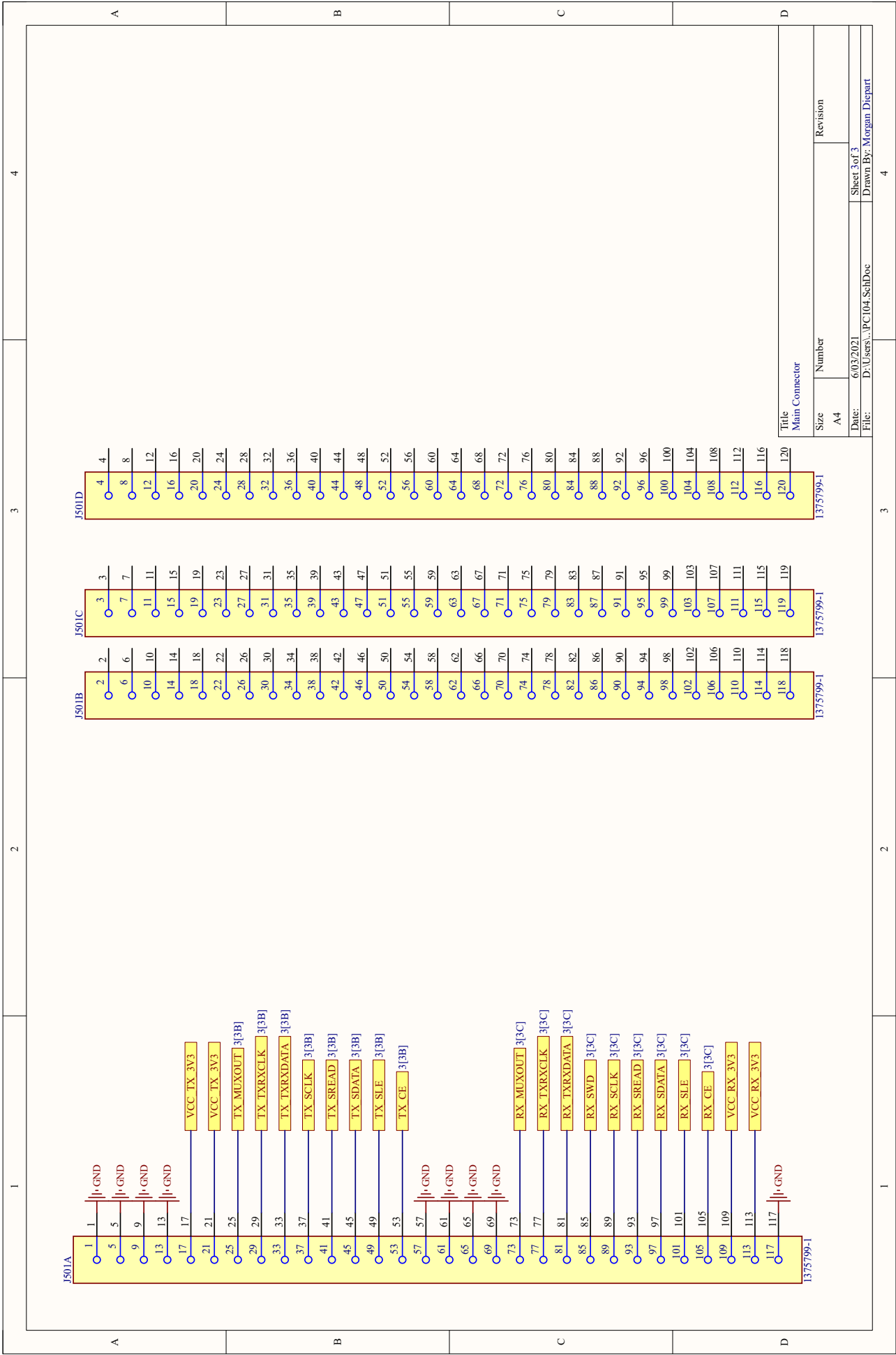




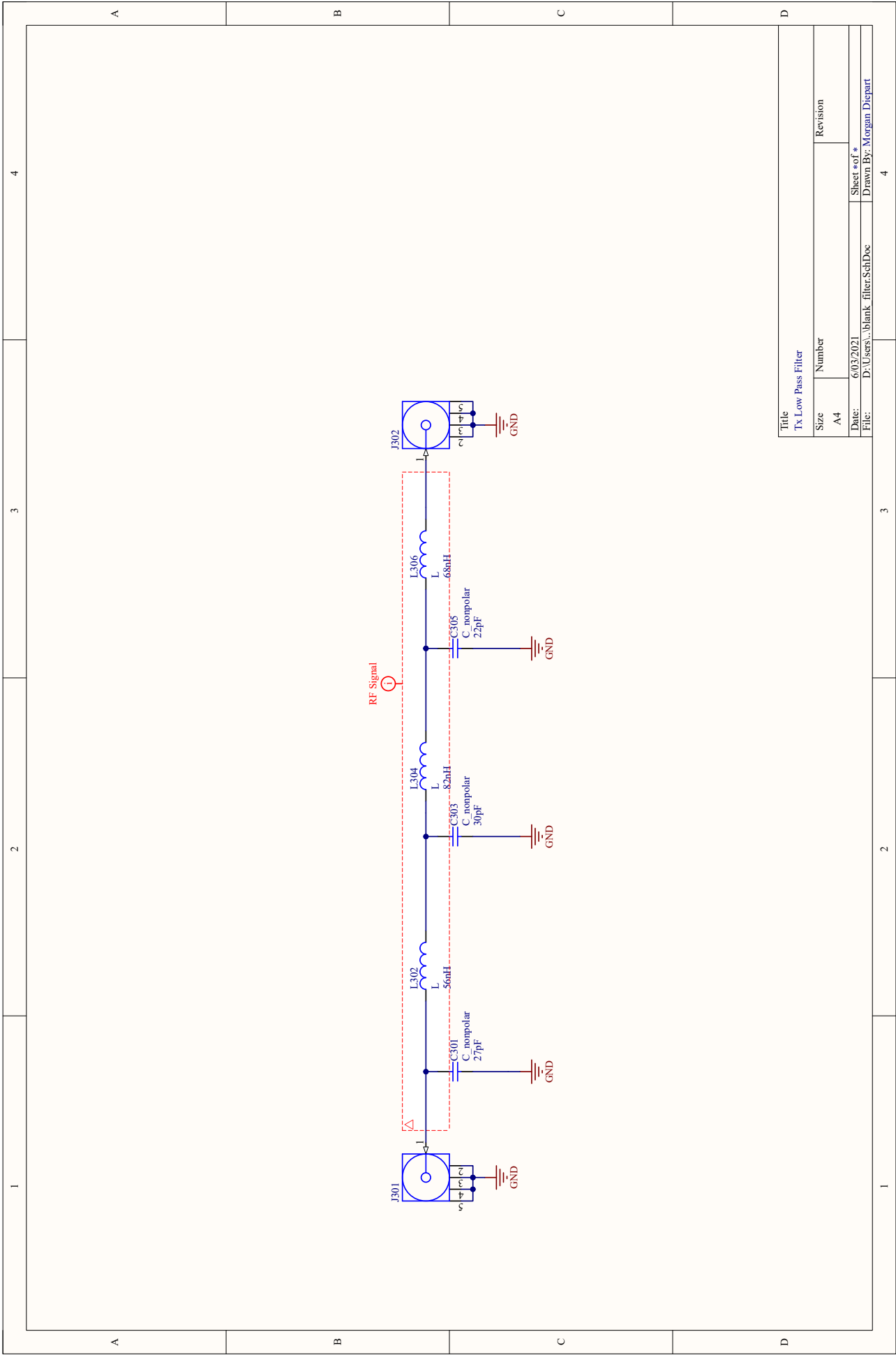
## D.4 Rx Transceiver Test Circuit



D.5 Transceiver Test Circuit Connector



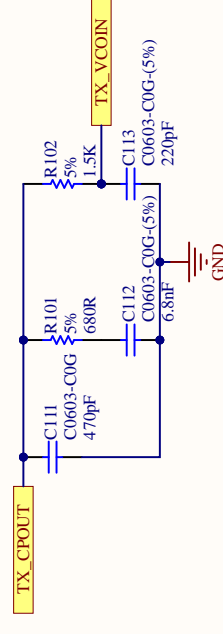
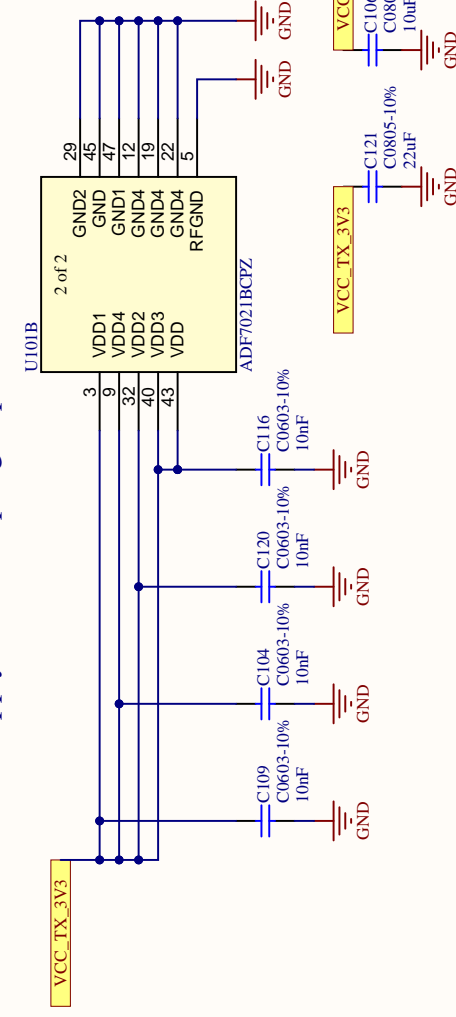
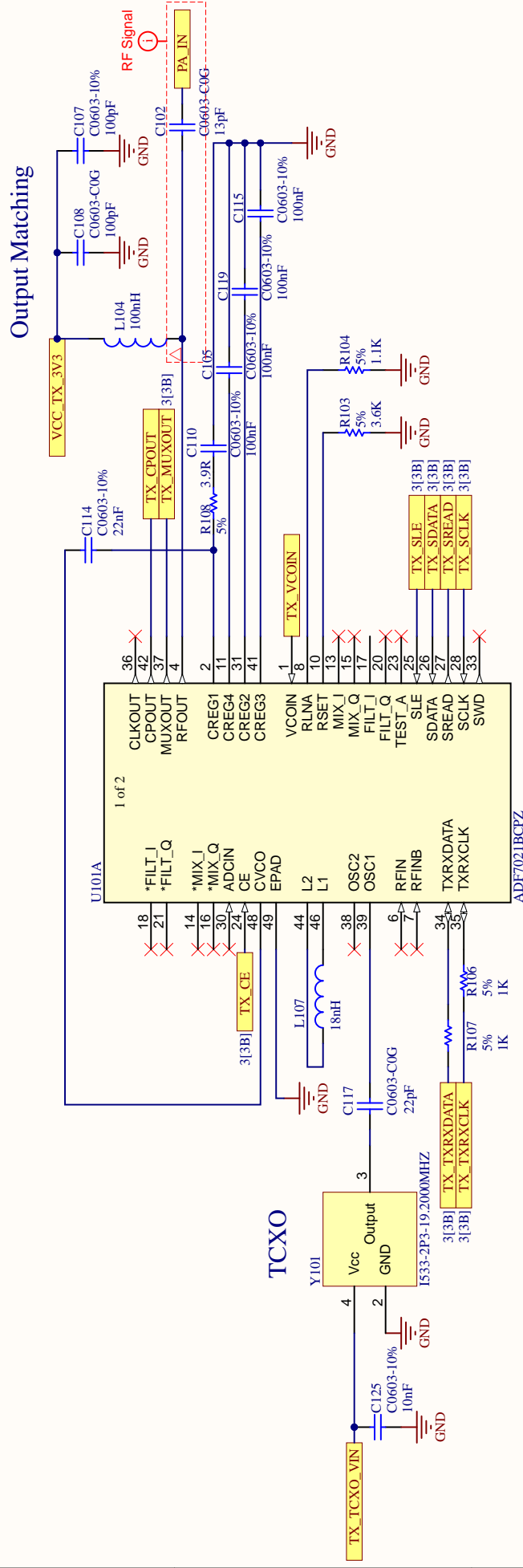
D.6 Tx Low Pass Filter



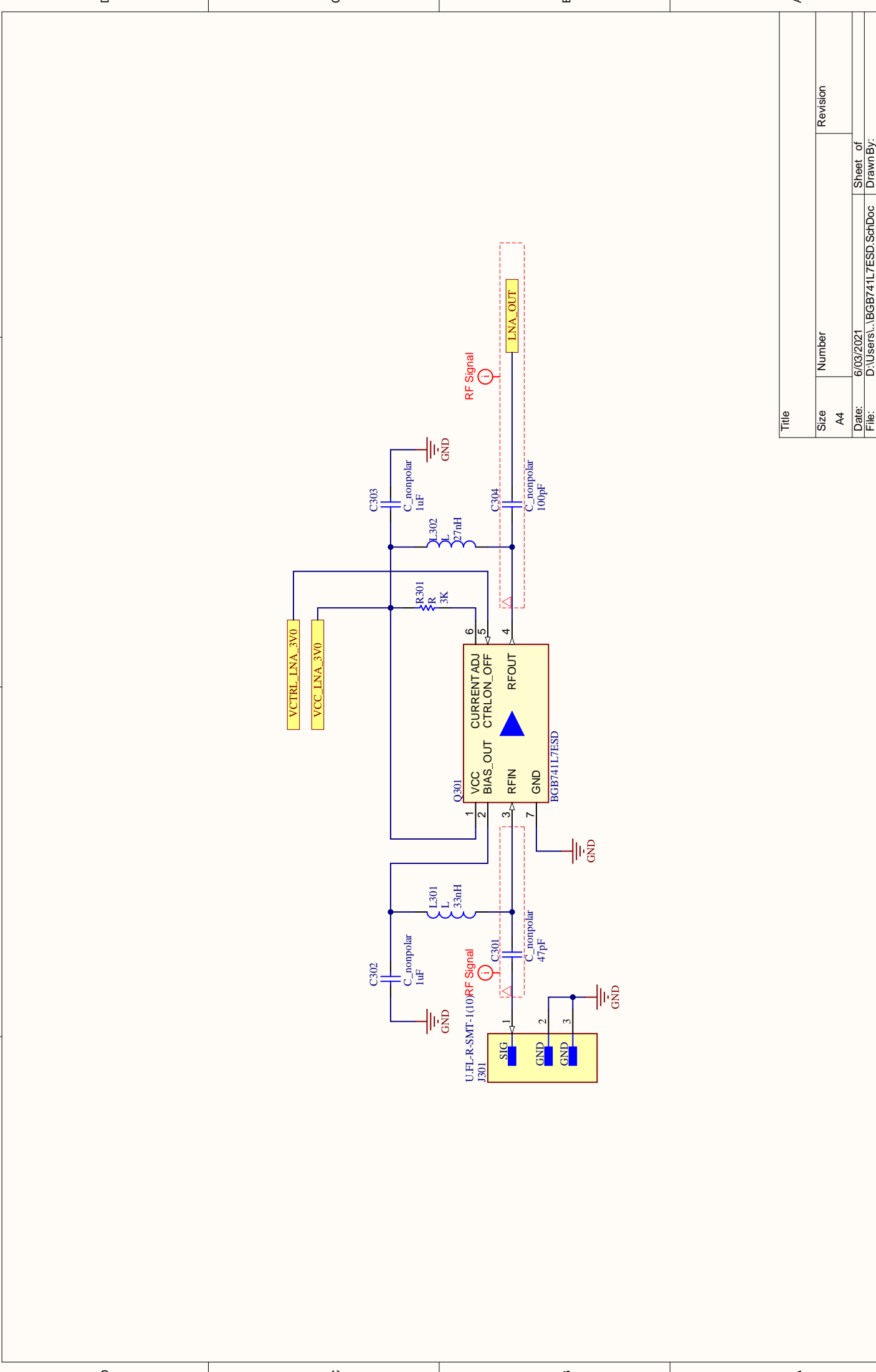
Appendix E

Final Board Schematic

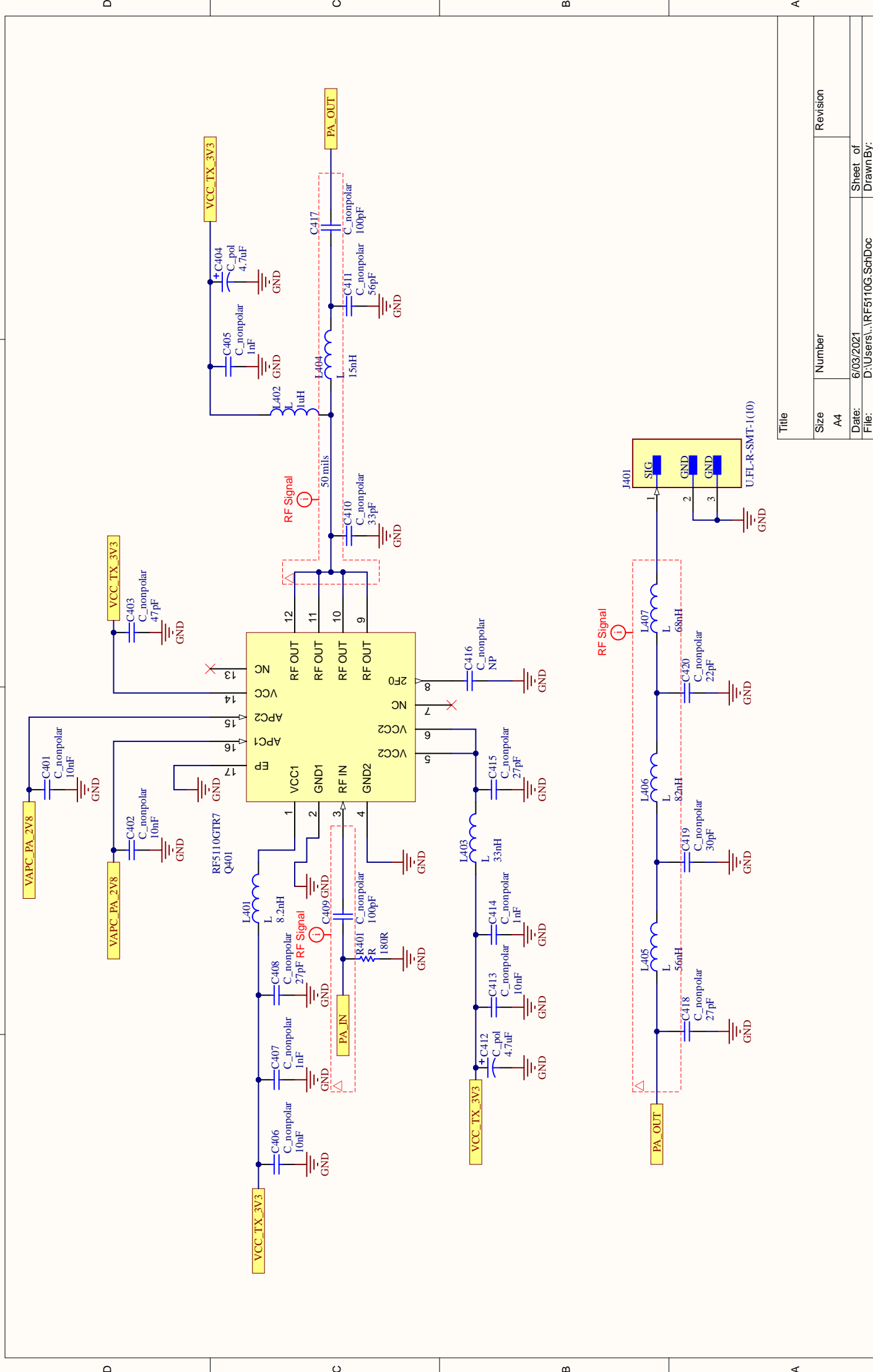




Title		
Size A4	Number	Revision
Date: File:	6/03/2021 D:\Users\...\ADF7021_Tx.SchDoc	Sheet of Drawn By:



Title		
Size	Number	Revision
A4		
Date:	6/03/2021	
File:	D:\Users\...\BGB741L7ESD.SchDoc	
Sheet of		1
Drawn By:		



Title		Revision	
Size	Number		
A4			
Date:	6/03/2021	Sheet of	
File:	D:\Users\...\RF5110G.SchDoc	Drawn By:	

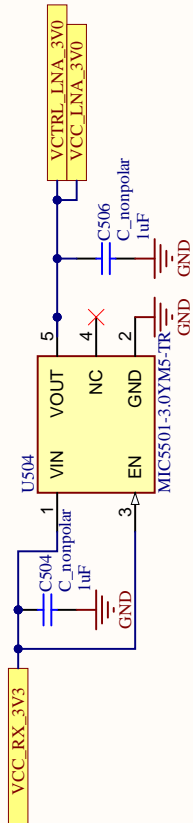
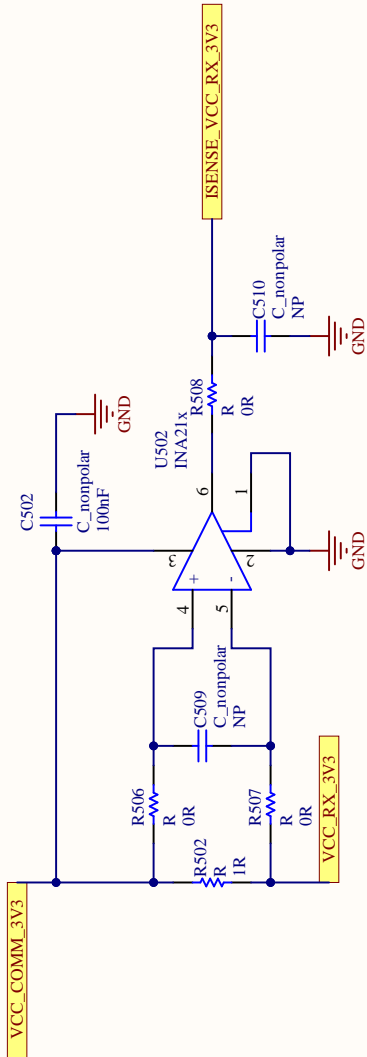
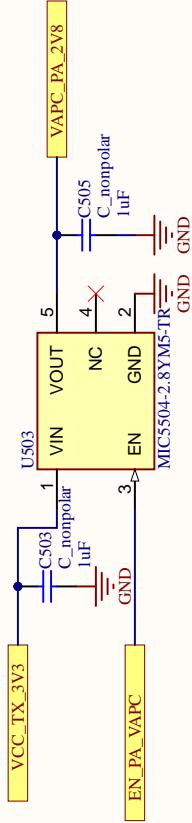
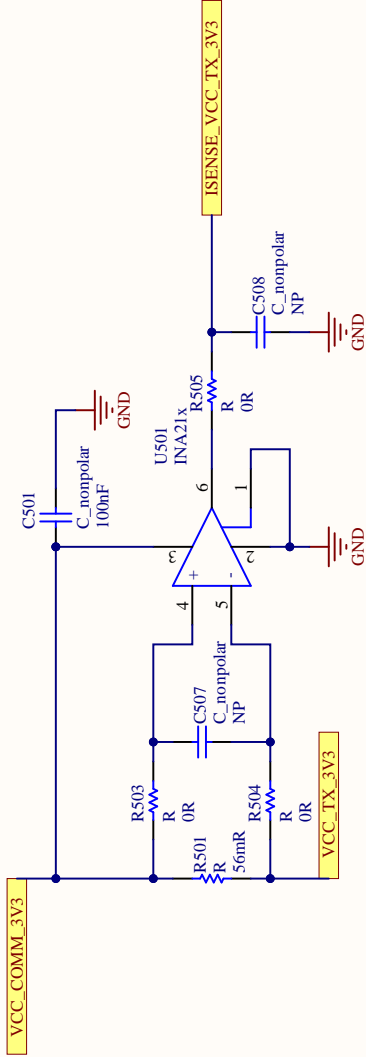


D

C

B

A



Title

Size

Number

Revision

Date:

File:

Sheet of

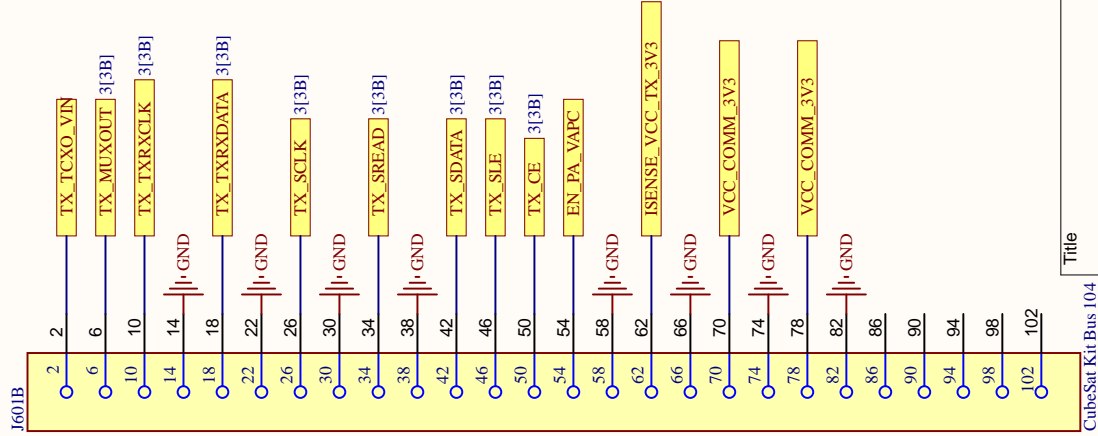
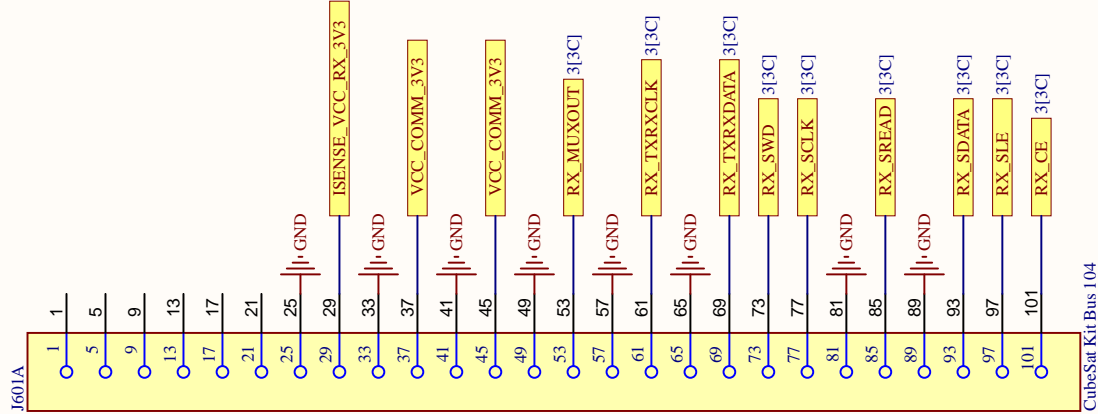
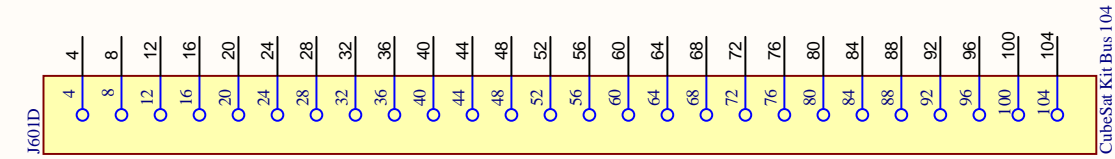
Drawn By:

D

C

B

A



Title

CubeSat Kit Bus 104

CubeSat Kit Bus 104

CubeSat Kit Bus 104

Revision

Size

A4

Date:

6/03/2021

Sheet of

D:\Users\...\PC104.SchDoc

Drawn By:

# Bibliography

- [1] François Piron. *Optimization of the AX.25 and D-STAR telecommunications systems of the OUFTE-2 nanosatellite*. Available at <https://matheo.uliege.be/handle/2268.2/6751>. June 2019.
- [2] ICOM. *IC-910H*. [Online; accessed 09-june-2021]. URL: <http://www.icomamerica.com/en/products/amateur/satellite/910h/options.aspx>.
- [3] Beko Elektronik. *HLV-550*. [Online; accessed 09-june-2021]. URL: <https://www.beko-elektronik.de/index.php/de/amateurfunk/70cm/hlv-1101>.
- [4] Wikipedia contributors. *X.25* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/w/index.php?title=X.25&oldid=1019637164>. [Online; accessed 2-June-2021]. 2021.
- [5] Wikipedia contributors. *AX.25* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/w/index.php?title=AX.25&oldid=1014823710>. [Online; accessed 2-June-2021]. 2021.
- [6] wb2osz. *Going Beyond 9600 Baud*. Available at <https://github.com/wb2osz/direwolf/blob/master/doc/Going-beyond-9600-baud.pdf>. May 2016.
- [7] Lime Microsystems. *LimeSDR*. [Online; accessed 09-june-2021]. URL: <https://www.crowdsupply.com/lime-micro/limesdr/updates/usb-boards-now-in-stock>.
- [8] Sebastien De Dijker. *Oufte 1 Ground Software*. Unpublished.
- [9] *Instruction Manual / Command Description for the Tracker/ DSPTNC Firmware Version 1.7*. Available at [https://www.p4dragon.com/download/SCS\\_Manual\\_DSPTNC\\_1.7.pdf](https://www.p4dragon.com/download/SCS_Manual_DSPTNC_1.7.pdf). SCS GmbH & Co. KG. Jan. 2018.
- [10] *HLV-550 Instruction & Operations Manual*. Available at [https://www.beko-elektronik.de/index.php/en/services-2/downloads/item/download/34\\_ff928cbbc4f5ae7d957b7928bf2418df](https://www.beko-elektronik.de/index.php/en/services-2/downloads/item/download/34_ff928cbbc4f5ae7d957b7928bf2418df). BEKO-ELEKTRONIK. 2014.
- [11] IBPT. *Plan des fréquences*. Aug. 2020. URL: <https://www.ibpt.be/consommateurs/plan-des-frequences>.
- [12] *RF5110G Data Sheet*. Rev. F, available at <https://www.qorvo.com/products/d/da000459>. Qorvo. Dec. 2020.
- [13] *ADF7021-N Datasheet*. Rev. B, available at <https://www.analog.com/media/en/technical-documentation/data-sheets/ADF7021-N.pdf>. Analog Devices. Mar. 2016.
- [14] *Design guide for RF low-noise transistors in global navigation satellite systems*. V X.Y, available at [https://www.infineon.com/dgdl/Infineon-AN\\_1805\\_PL32\\_1806\\_113119\\_design%20guide%20for%20low%20noise%20transistors%20-AN-v01\\_00-EN.pdf?fileId=5546d46265f064ff016643d954321028](https://www.infineon.com/dgdl/Infineon-AN_1805_PL32_1806_113119_design%20guide%20for%20low%20noise%20transistors%20-AN-v01_00-EN.pdf?fileId=5546d46265f064ff016643d954321028). Infineon. May 2018.
- [15] *AN-859*. Rev. A, available at <https://www.analog.com/media/en/technical-documentation/application-notes/AN-859.pdf>. Analog Devices. 2009.
- [16] *CubeSat Kit PCB Specification*. Rev. A5, available at [http://www.cubesatkit.com/docs/CSK\\_PCB\\_Spec-A5.pdf](http://www.cubesatkit.com/docs/CSK_PCB_Spec-A5.pdf). Pumpkin Inc. June 2007.
- [17] dasimon. *Calculation of the occupied bandwidth for digital frequency modulation*. [Online; accessed 3-June-2021]. Feb. 2015. URL: [https://www.silabs.com/community/wireless/proprietary/knowledge-base.entry.html/2015/02/17/calculation\\_of\\_theo-S9wI](https://www.silabs.com/community/wireless/proprietary/knowledge-base.entry.html/2015/02/17/calculation_of_theo-S9wI).

- [18] *I533 Series*. Rev. I, available at <http://www.farnell.com/datasheets/1998851.pdf>. ILSI America. Nov. 2020.
- [19] *SAM V71 Datasheet*. Rev. 44003E, available at [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-44003-32-bit-Cortex-M7-Microcontroller-SAM-V71Q-SAM-V71N-SAM-V71J\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-44003-32-bit-Cortex-M7-Microcontroller-SAM-V71Q-SAM-V71N-SAM-V71J_Datasheet.pdf). Atmel. Oct. 2016.
- [20] PC104 Consortium. *PC/104*. [Online; accessed 2-June-2021]. Oct. 2008. URL: [https://pc104.org/wp-content/uploads/2015/02/PC104\\_Spec\\_v2\\_6.pdf](https://pc104.org/wp-content/uploads/2015/02/PC104_Spec_v2_6.pdf).
- [21] PC104 Consortium. *PC/104-plus*. [Online; accessed 2-June-2021]. Oct. 2008. URL: [https://pc104.org/wp-content/uploads/2015/02/PC104\\_Plus\\_v2\\_32.pdf](https://pc104.org/wp-content/uploads/2015/02/PC104_Plus_v2_32.pdf).
- [22] *INA21x Voltage Output, Low- or High-Side Measurement, Bidirectional, Zero-Drift Series, Current-Shunt Monitors*. Rev. J, available at <https://www.ti.com/lit/ds/symlink/ina210.pdf>. Texas Instrument. Feb. 2017.
- [23] *AN-1258*. Rev. 0, available at <https://www.analog.com/media/en/technical-documentation/application-notes/AN-1258.pdf>. Analog Devices. Nov. 2013.
- [24] Radio Amateur Satellite Corporation. *AMSAT-IARU Link Model*. Version 2.5.3. Dec. 2008. URL: [www.amsat.org/wordpress/xtra/AMSAT-IARU\\_Link\\_Model\\_Rev2.5.3.xls](http://www.amsat.org/wordpress/xtra/AMSAT-IARU_Link_Model_Rev2.5.3.xls).