



CHALMERS



Software Defined Radar System

Hardware application development

In partial fulfillment of the requirements for the
BACHELOR'S DEGREE in **ELECTRICAL ENGINEERING**

MARKO RUSSEGREN SLADIC

ENES FEHRATOVIC

Software Defined Radar System

Hardware application development

MARKO RUSSEGREN SLADIC

&

ENES FEHRATOVIC

Department Of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014

**Software Defined Radar System
Hardware application development
MARKO RUSSEGREN SLADIC
ENES FEHRATOVIC**

© MARKO RUSSEGREN SLADIC & ENES FEHRATOVIC, 2014

**Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000**

**Cover:
PXI System & SAAB Electronic Defence Systems (EDS)**

**Chalmers University of Technology
Göteborg, Sweden 2014**

Software Defined Radar System
Hardware application development
MARKO S. RUSSEGREN
ENES FEHRATOVIC
Department of Signals and Systems
Chalmers University of Technology

ABSTRACT

This research explores and describes the construction of a radar hardware application with the objective to locate targets within a predefined range. Radar systems are known to suffer from high expenses both in terms of hardware cost and system development. The development of new multitude radar system range implementations are pushing for new ground breaking solutions putting additional restraints on allocated project cost. Research for broader context applications naturally demand more affordable yet flexible hardware solutions.

This project explores the development of a Software Defined Radio (SDR) technology based radar system. The main reason for using SDR technology is to lower excessive production and development costs and make it more desirable in markets where sophisticated radars are too expensive.

The hardware adopted for this project is based on two FlexRio Field Programmable Gate Arrays (FPGA) cards combined with two Radio Frequency (RF) Frontend modules. These two FPGA cards will be used for signal processing and communication with the host PC – used for radar system control and data presentation.

The system has shown to be fully operational with successful echo verification and ranging on distances less than 100 meters. As the system has a limited range factor it could not be successfully used as a long range radar application. Although, the same construct and integrated system may be upgradeable to meet the requirements of a low cost long range radar system.

There is a strong correlation between hardware performance and short range distance measurements. And it also turns out that the energy requirement ie. the signal amplitude is crucial for long distance measurements. Future upgrades should therefore be considered in order to meet the requirements for these different range measurements.

Keywords: SDR, Software Defined Radio, Radar, Hardware Application Development, FPGA, FlexRio, PXI System, Antenna Array

ACKNOWLEDGEMENTS

We would like to express our gratitude to all whose support made this thesis work possible. Primarily, thanks to our thesis advisors Manne Stenberg from Chalmers University of Technology and Lennart Berlin from SAAB Electronic Defence Systems for their guidance and input during this work.

We would also like to thank our professor and mentor Sakib Sisteck and our examiner Bertil Thomas for making this thesis work possible.

Thanks to Jonas Mäki and Payman Tehrani from National Instruments for all their input regarding the LabVIEW software and the National Instruments hardware used in this work.

MARKO RUSSEGREN SLADIC
ENES FEHRATOVIC

TABLE OF CONTENTS

INTRODUCTION	1
1.1 BACKGROUND	1
1.2 PURPOSE AND OBJECTIVE	1
1.3 FRAME OF REFERENCE	2
1.4 PROJECT CONSTRAINT	2
1.5 BASELINE ASSESMENT FORMULATION.....	2
METHOD	3
2.1 HARDWARE DESCRIPTION.....	3
2.1.1 HOST COMPUTER	3
2.1.2 RF-FRONTEND.....	4
2.1.3 FPGA.....	5
2.1.4 LOG-PERIODIC ANTENNAS	6
2.1.5 LOW NOISE AMPLIFIERS	6
2.2 SOFTWARE TOOLS	7
2.2.1 LabVIEW	7
2.3 RADAR BASICS	8
2.4 SDR TECHNOLOGY	11
HARDWARE DEVELOPMENT	12
3.1 RF - FRONTEND MODULES	12
3.2 FPGA MODULES	15
3.2.1 INITIALIZATION & CONFIGURATION	16
3.2.2 SIGNAL PROCESSING.....	16
3.2.2.1 TX PROCESS	17
3.2.2.2 RX PROCESS	20
3.2.3 MIMO & NON-MIMO METHODS	22
3.3 ARRAY OF ANTENNAS	23
RESULTS	26
4.1 HARDWARE PERFORMANCE	26
4.2 DATA ANALYSIS.....	27
4.2.1 MEASURMENT TEST 1	30
4.2.2 MEASURMENT TEST 2	31
4.2.3 MEASURMENT TEST 3	32

PROBLEMS AND DISCUSSION	33
5.1 PROBLEMS.....	33
5.1.1 TRANSMISSION BREAKS	33
5.1.2 COMPILER ERROR	33
5.2 SOLUTIONS.....	34
5.2.1 TRANSMISSION BREAKS	34
5.2.2 COMPILER ERROR	34
SUMMARY AND CONCLUSIONS	35
6.1 ENVIROMENTAL & ETHICAL ASPECTS.....	35
REFERENCES	36
APPENDICES.....	38
APPENDIX I.....	38
APPENDIX II	39
APPENDIX III.....	40
APPENDIX IV.....	41

LIST OF ABBREVIATIONS

ASCII: American Standard Code for Information Interchange
ASIC: Application Specific Integrated Circuit
CPU: Central Processing Unit
DC: Direct Current
DSP: Digital Signal Processing
EDS: Electronic Defence Systems
FFT: Fast Fourier Transform
FIFO: First In – First Out
FIR: Finite Impulse Response
FMCW: Frequency-Modulated Continuous-Wave
FPGA: Field-Programmable Gate Array
GPS: Global Positioning System
I/O: Input - Output
ID: Identification Number
LNA: Low Noise Amplifier
LO: Local Oscillator
LP: Log-Periodic
MIMO: Multiple Input Multiple Output
NI: National Instruments
PCB: Printed Circuit Board
PRF: Pulse Repetition Frequency
PRT: Pulse Repetition Time
PXI: PCI eXtensions for Instrumentation
PXIe: PCI eXtensions for Instrumentation express
RF: Radio Frequency
RX: Receiver
SDR: Software Defined Radar
SMA: SubMiniature version A
TX: Transmitter
VHDL: VHSIC (Very High Speed Integrated Circuit) Hardware Description Language
VI: Virtual Instrument

INTRODUCTION

1.1 BACKGROUND

This project outlines collaboration between Chalmers University of Technology and SAAB Electronic Defence Systems (EDS). With more than 50 years of experience developing radar systems and offering solutions for surveillance, electronic warfare and self-protection systems, SAAB EDS is facing a new era of low-cost high precision radar system technology.

Products encompassing radar systems in present time suffer from being all too expensive for a multitude system range implementation. The main cause but also restraint on future development is mainly due to system hardware development costs. Research for broader context applications naturally demand more affordable yet flexible hardware solutions. Building multi-application radar technology with rather limited budget constraint set this project towards its first hurdle. Building a working prototype would thus require steering away from existing analog system based methodology and towards Software Defined Radio (SDR) technology.

1.2 PURPOSE AND OBJECTIVE

The purpose of this thesis is to create and develop a radar system application based on SDR technology with focus on hardware development. The main reason for using SDR technology is to lower the production and development costs and make it more desirable in markets where sophisticated radars are too expensive.

Another reason for pursuing this technology is due to its high flexibility. Hardware can thus easily be transformed into e.g. encrypted radio, telecommunication transceivers or even a Global Positioning System (GPS) just by the push of a button.

The goal is to create a working prototype that can detect objects at various distances using two FlexRio Field Programmable Gate Arrays (FPGA's), two Radio Frequency (RF) Frontend modules and one PXIe host computer. Some external components such as Log-Periodic antennas, power generators and amplifiers are also used in order to increase the power of the radar signals.

During the incremental development stages of the project, tests based on a multitude of measured data and range scenarios are subjects for analysis - setting a baseline for development of a more agile radar system.

1.3 FRAME OF REFERENCE

The main source of the technical information used in this thesis has been taken from our previous project at SAAB EDS that can be found in our report “Software Defined Radar” [1].

1.4 PROJECT CONSTRAINT

Both prior to and during the project work, limitations and boundaries were shaped so to make sure that main goals would be tangible within the specified scope and time frame. It was decided that the radar system should be adapted for both moving and non-moving targets without considering the Doppler Effect. In other words, the radar system should only be able to measure the distance of objects without considering their actual speed and bearing. The measurement of the Doppler Effect can easily be implemented in the future just by making some minor changes in the software.

The main system hardware is, for this project, limited to two sets of FlexRio FPGA’s, RF-Frontend modules and one PXIe host computer from National Instruments. This implies that the PXIe host computer will control the two FPGA’s and RF-Frontend modules used as transceivers, each with their own Log-Periodic antenna.

Since no other external components were a part of the project scope, except for the Log-Periodic antennas, amplifiers and power generators, the radar system is only measuring objects in a manually selected direction and within the antennas beam-width. No automatic or rotating systems are going to be added to control the physical directionality of the antennas.

1.5 BASELINE ASSESMENT FORMULATION

Does the hardware meet expectations and requirements for SDR radar system performance?

Are external components needed to make the system more powerful and sophisticated?

Within what range can objects be measured?

Is there a low cost alternative however performance wise preferable as a replacement for the system hardware that can be used in future developments?

Is the system hardware capable of bringing harm to humans and the environment?

METHOD

2.1 HARDWARE DESCRIPTION

The purpose of this chapter is to describe the hardware that has been used in this project. All of the hardware, except for the Low Noise Amplifiers (LNA's), belongs to National Instruments (NI). The radar application will be built on a PXI system that consists of one chassis, one mounted host computer, two mounted FPGA cards and two mounted RF-Frontend modules, see figure 1. The PXI system used in this project is shown in appendices III and IV. The purpose is to have all of the hardware mounted and implemented on one single chassis PXIe-1078 from NI.



Figure 1: Content of the NI PXI system used for this radar application

2.1.1 HOST COMPUTER

The host computer is an NI PXIe-8133 built in PC included in the chassis PXIe-1078 from NI. The purpose of the host computer is to display data from the radar system, such as the transmitted signals and the received signals or echoes. The host computer allows the user to control and monitor the whole radar system. The host computer is also used to program the hardware of the controllers. Table 1 shows the specification of the host computer model used in this project.

Table 1: Specification for the NI 5791R RF-Frontend module [2]

NI PXIe-8133 Host PC
Operating system: Windows 7 with LabVIEW 2013 SP1 and hardware drivers installed
Intel Core i7-820QM quad-core processor
2 GB dual-channel 1333 MHz DDR3 RAM
DVI-I video connector

2.1.2 RF-FRONTEND

The RF-Frontend modules used in this project are two NI 5791R modules. These RF-Frontend modules are products of National Instruments, and are specifically built to be used together with the NI 7966R FPGA card. The purpose of the Frontend modules is to convert all the digital signals from the FPGA card to analog form and transmit them over the air as RF signals within a specific frequency band with the help of antennas. The Frontend modules can also receive analog radio signals with the help of antennas and then convert them to digital form. These digital signals are then forwarded to the FPGA for signal processing and data analysis.

This specific RF-Frontend model has one transmitting output (Tx) and one receiving input (Rx). It operates within a frequency range from 200 MHz up to 4,4 GHz. The operating frequency is the carrier frequency of the transmitted radar signals. Table 2 shows the specification of the NI 5791R RF-Frontend module.

The Frontend modules are connected to the FPGAs through a custom Input Output (I/O) and the antennas are connected to the Frontend modules through SubMiniature Version A (SMA) connectors. Between the antenna and the Rx input on the Frontend module, two different LNA's are used for different operational frequencies to increase the power of the signals.

Table 2: Specification for the NI 5791R RF-Frontend module [3]

NI 5791R RF-Frontend Module
Adapted for the NI Flex Rio 7966R FPGA card
RF TX and RX with shared LO
200 MHz to 4,4 GHz operating frequency range
100 MHz instantaneous bandwidth
LO input and output for MIMO synchronization
12 bidirectional GP digital I/O channels
RX dynamic range greater than 105 dB

2.1.3 FPGA

Field-Programmable Gate Arrays, mostly known as FPGAs, are flexible semiconductor devices that can be programmed and reprogrammed after manufacturing. Since the FPGA's are programmed with a specific set of functional units, compared to e.g. microprocessors - their performance can be exceedingly higher if they are designed for one specific application.

Two FPGA's are used in this project with the purpose to increase the speed and performance of the Digital Signal Processing (DSP) in the radar system. All RF radar signals received from the Log-Periodic (LP) antennas are converted to digital form by the RF-Frontend module. The digital signals are then processed by the FPGA and forwarded to the host computer for further analysis.

The FPGA modules used in this project are two NI FlexRio 7966R. They have an integrated DSP-focused Xilinx Virtex-5 FPGA that can be programmed with LabVIEW software. They have a maximum clock rate of 500 MHz and are adapted to communicate with the PXIe host computer through the PXI Express bus. They are also adapted for communication with the RF-Frontend modules NI 5791R through their custom I/O [4]. Table 3 shows the specification of the NI FlexRio 7966R FPGA cards.

These FPGA modules are improved to allow the new technology of Peer-to-peer streaming that enables direct data streaming among multiple FPGA modules without sending data back to the host processor [4]. This is very useful when multiple FPGA's are used, as in this project.

Table 3: Software and hardware specification for the NI Flex Rio 7966R [4]

NI Flex Rio 7966R
DSP-focused Xilinx Virtex-5 SX95T
LabVIEW Programming Software
500 MHz Maximum Clock Rate
1 Gbits/s Maximum Data Rate / Channel
Adapted for the PXI Express Bus
512 MB onboard DDR2 DRAM
16 DMA channels
High-speed data streaming at 800 MB/s
Customizable I/O with NI FlexRio Adapter MDK

2.1.4 LOG-PERIODIC ANTENNAS

The antennas used in this project, for transmitting and receiving the RF radar signals, are four Log-Periodic Printed Circuit Board (PCB) antennas. Log-Periodic antennas are directional, narrow-beam antennas that operate over a broad band of frequencies. Since radars typically operate at frequencies ranging from 1 GHz up to 100 GHz and the RF-Frontend 5791R modules have an operating frequency of 200 MHz up to 4,4 GHz, LP antennas capable of working within this frequencies are needed.

The purpose of using four LP antennas is to build an array of antennas. The main reason for this is to improve the directional characteristics and to obtain a higher antenna gain. Normally, an array consisting of N identical elements can achieve an increase in gain of up to a factor of N. This is very useful in radar applications in order to obtain a good response from the measured objects in form of a high echo reflection due to the increased gain. More about this theory is explained in chapter 3.3.

The specific LP antennas used in this project are the LP0965 PCB antennas manufactured by Kent Electronics. The antennas operate in frequencies from 850 up to 6500 MHz. The antennas are connected to the 5791R RF-Frontend modules through SMA connectors. Table 4 shows the specification of the LP0965 antenna.

Table 4: Specifications of the LP0965 antenna [5]

LP0965 Log-Periodic PCB Antenna
850 MHz to 6,5 GHz frequency range
Typical Forward Gain: 6 dBi
Size: 13,4 x 14,2 cm
120 deg. Horizontal and 160 deg. Vertical Beamwidth

2.1.5 LOW NOISE AMPLIFIERS

Since the RF-Frontend module transmits radio signals with an extremely low power compared to other radar systems, some types of amplifiers are needed. One source that also stand for most of the power losses is the coaxial cable feed line, the longer the feed line, the greater the loss. Therefore an LNA is used between the antenna and the Rx input on the RF-Frontend module to decrease those losses and to enhance the signal power. Two different LNA's are used in this project. One that works within frequencies from 5 MHz up to 1 GHz with a 15 dB gain [6] and the other one used for frequencies from 2 GHz up to 4 GHz with a 23 dB gain [7]. The reason for having two LNA's is because several tests of the radar system are made on different frequencies. Since these amplifiers work with an input voltage from 12 to 15 Vdc, an external Direct Current (DC) power generator is also used.

2.2 SOFTWARE TOOLS

The FPGA's and the RF-Frontend modules in this radar application were programmed with the graphical development platform from National Instruments, LabVIEW. This is a different approach to the hardware programming since the FPGA's are usually programmed in (Very High Speed Integrated Circuit Hardware Description Language) VHDL.

2.2.1 LabVIEW

LabVIEW stands for "Laboratory Virtual Engineering Workbench" and is a graphical development platform that can be used to program different types of hardware devices. LabVIEW is the software used in this project to program the RF-Frontend modules and the FPGA's devices. It was an obvious choice since it has all the necessary drivers for the given hardware.

LabVIEW belongs to a small group of programming languages called graphical programming. Different hardware can be programmed with the help of blocks and strings instead of American Standard Code for Information Interchange (ASCII) code as in e.g. C programming and VHDL, see figure 2. This programming language allows users to easily process data with built in filters, Fast Fourier Transforms (FFT's) and other mathematical algorithms. The processed data can also be analyzed with the help of built in graphs and indicators [8].

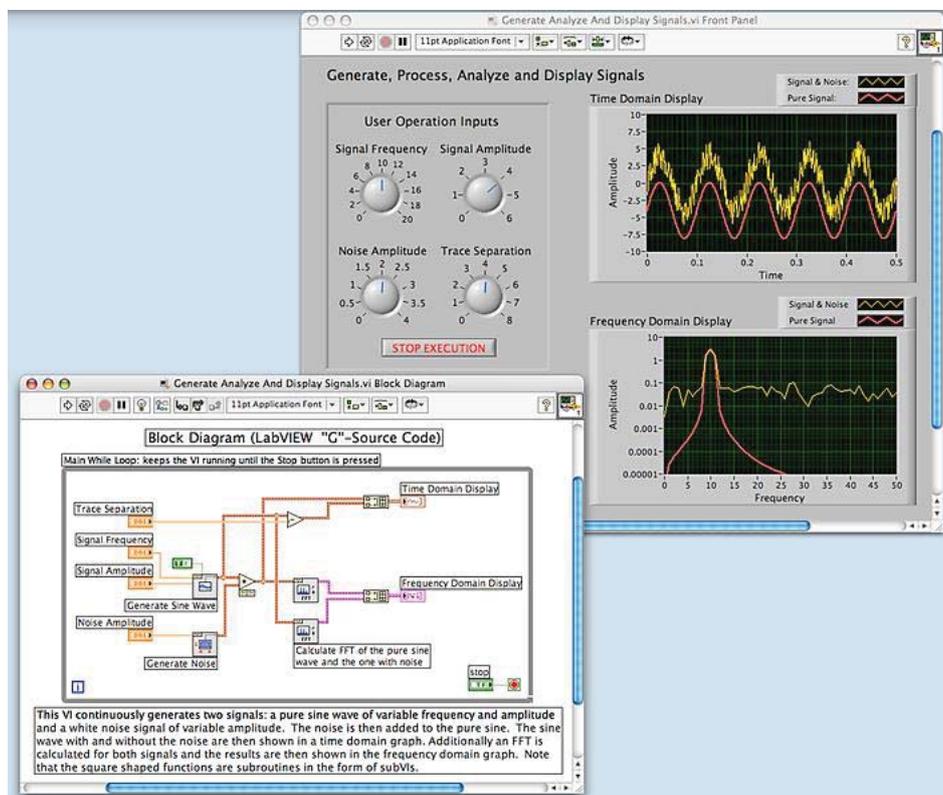


Figure 2: Sample of a LabVIEW program

The LabVIEW platform generates a file extension called Virtual Instrument (VI) that can be run by the user. The VI is primarily divided into two parts. The first part is the “block diagram”, where all the programming takes place with the help of different blocks and strings, and the second part is the “front panel”, where all the controls, indicators and graphs are located. One main VI can also consist of many sub-VI’s.

The user of the radar application will be able to control the radar system and set up different parameters, e.g. the operating frequency of the radar, from the front panel. All the graphical code in the block diagram and in the sub-VI’s will be hidden from future users.

2.3 RADAR BASICS

In order to use the RF-Frontend modules for radar applications, some basic knowledge about radar systems is needed. Radar systems are basically systems sending out electromagnetic wave pulses within the so called radio wave portion of the electromagnetic spectrum. Those wave pulses are propagated through the air and will subsequently hit different objects.

When one wave pulse hits an object, some of the radio energy from the pulse is going to reflect back to the radar system after a certain amount of time. Those reflections are called echoes, see figure 3. By knowing the time it takes for an echo to reach the radar receiving antenna, the system is then able to determinate the distance from the antenna to the measured object.

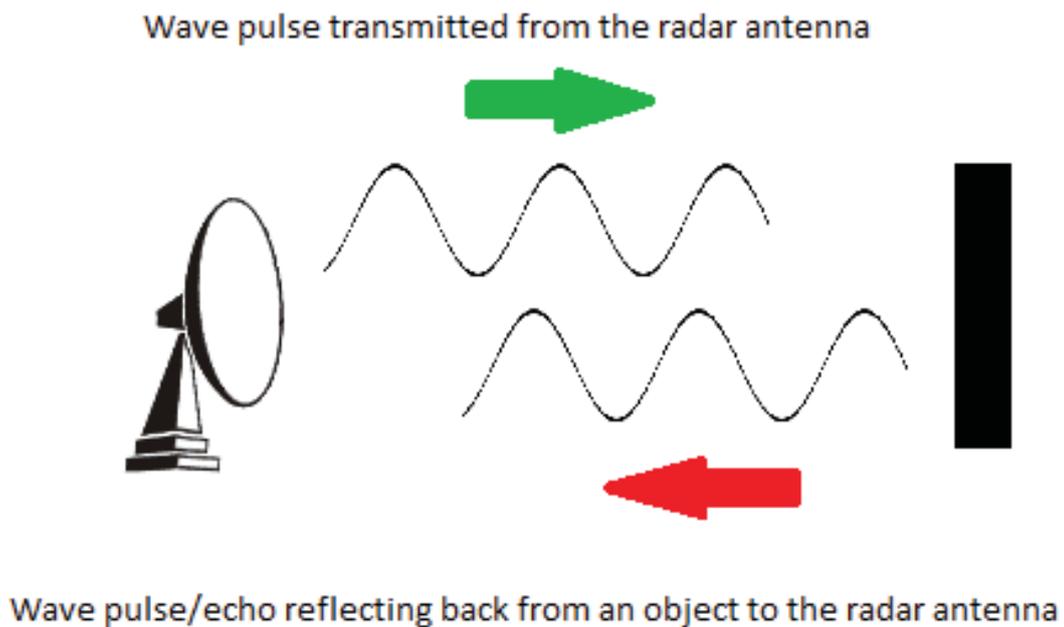


Figure 3: Propagated electromagnetic wave pulse and the reflecting echo from an object

Since the electromagnetic waves travel at the speed of light, the distance from an object to the antenna of a radar system can be determined with the following formula [9]:

$$(2.1) R = \frac{c_0 * t}{2}$$

Where:

R = the distance between the object measured and the radars receiver antenna in meters

t = the time measured for the echo to arrive at the receiver antenna in seconds

c_0 = the speed of light ($\sim 3 * 10^8$ m/s)

All echoes have to arrive at the receiving antenna after the wave pulse was transmitted and not during transmission. If an object is too close to the receiving antenna, the echo would neither be visible or valid on the radars receiver, since it would interfere with the transmitting signals. This means that the radar transceiver should be divided in two different parts. The first part would be the transmitting time and the second part would be the listening time.

Because of this, other important factors of the radar system are the Pulse Repetition Frequency (PRF) or Pulse Repetition Time (PRT) and the duty cycle, see figure 4.

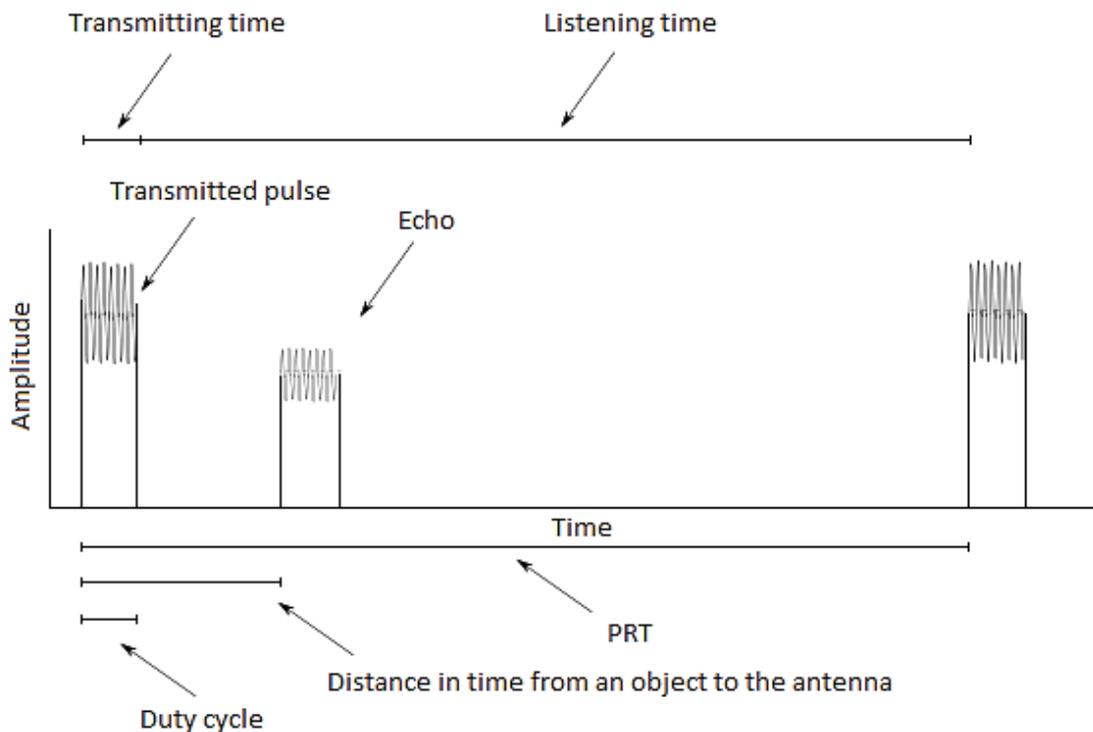


Figure 4: Example of an echo

The PRF is the number of pulses transmitted per second and the PRT is the time from the beginning of one transmitted pulse to the beginning of the next transmitted pulse. The duty cycle is the percentage of the transmitting pulse within the PRT. Radars usually have a duty cycle of 10% of the PRT time. By adjusting the duty cycle and the PRF or PRT, the minimum and maximum distance range of a radar system can be decided, see formula (2.2) and (2.3).

$$(2.2) R_{MIN} = \frac{c_0 * (PRT * \frac{\tau}{100})}{2}$$

$$(2.3) R_{MAX} = \frac{c_0 * (PRT * \frac{100 - \tau}{100})}{2}$$

Where:

R_{MIN} = the minimum distance range of a radar system in meters

R_{MAX} = the maximum distance range of a radar system in meters

PRT = the Pulse Repetition Time in seconds

τ = the duty cycle in %

The maximum distance range of a radar system is also decided by the transmitted power and the reflected power. If the reflected power is too low, the echo may be seen as noise on the radar receiver and will not be recognized by the radar system. Parameters that should be considered in this context are the reflected and transmitted powers, the wavelength of the transmitted signals, the radar cross section and the antenna gain [10].

If an object is located at a distance larger than the maximum distance decided by the selected PRF, there is a possibility that the echoes can be received by the radar system outside of the expected listening time which can result in measurement errors, see figure 5. These errors can be avoided by labelling the wave signals transmitted with changes in phase.

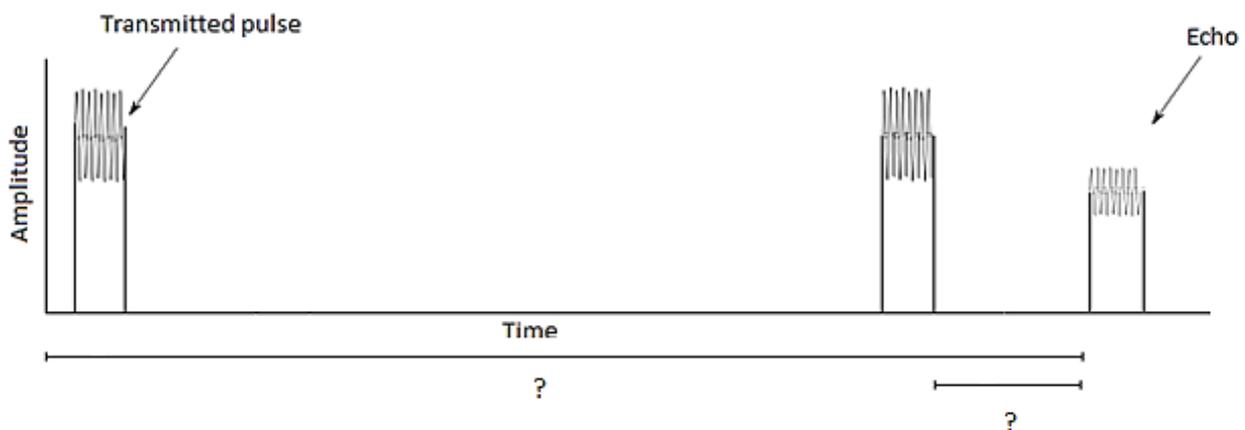


Figure 5: Example of a possible error during large distance measurements

2.4 SDR TECHNOLOGY

SDR stands for Software Defined Radio and it's a communication system where some or even all the physical layer functions and analog components as e.g. mixers, modulators, filters, amplifiers, etc., are implemented by means of software.

A device or hardware based on SDR technology can be used for multiple purposes where the communication between those devices is made within the RF spectrum. This means that the device based on SDR technology is incredibly flexible can be used for communication with cell phones, GPS, WIFI and other wireless computer systems and as in this case Radar Systems [11].

Not only are devices based on SDR technology very flexible, they can also provide efficient and inexpensive solutions compared to other technologies. As e.g. traditional radar systems are mostly built with analog parts which means that the system can only be modified through physical layer interventions. This means that the flexibility is very limited and the modifications can result in higher production costs.

Devices that are based on SDR technology usually have FPGA's or other types of DSP's that can easily be reprogrammed and modified. This means e.g. a radar application can easily be programmed and transformed into a GPS or an encrypted telecommunication system with the push of a button.

Because of its high flexibility and low production cost, SDR technology is also very efficient while building prototypes and test systems. The system can quickly be programmed and if necessary even reprogrammed allowing "bug fixes" in the system. This "bug fix" can even occur while the system is running and in service. This can also reduce possible future maintenance and transportation costs.

For the radar system in this project, SDR technology allows adding new future features if needed just by making some minor changes in the application software. Some examples of the features used in more sophisticated radar systems today are GPS, signal labeling with phase changes, velocity and bearing measurements, etc. All these features be added to the radar application in the future just by reprogramming it without the need to modify the analog components or physical layers used.

HARDWARE DEVELOPMENT

3.1 RF - FRONTEND MODULES

As mentioned in chapter 2.1.2, there are two main purposes for using the RF-Frontend modules. The first one is to convert the digital signals, from the corresponding FPGA cards, to analog form. Those analog signals are then broadcasted with the help of antennas as RF signals through air.

The second purpose is to receive the signals or echoes that are reflected back to the antennas from different objects. Those signals are then converted to digital form by the RF-Frontend modules and forwarded to the FPGA cards for signal processing.

The RF-Frontend modules are controlled and monitored by the host computer through the FPGA cards. The FPGA cards handle the communication between the host computer and RF-Frontend modules. This part is further explained in the chapter 3.2.

A special VI is made in LabVIEW, for the user on the host computer, to monitor and control the RF-Frontend modules. Two different methods are used for this radar application. The first method, named non-Multiple Input Multiple Output (non-MIMO), is to use one RF-Frontend module as a transmitter and the other one as a receiver where both modules work simultaneously. The second method, named Multiple Input Multiple Output (MIMO), is to have both modules working synchronously switching from transceivers to receivers.

Some of the following parameters can be set by the user to control the radar system for both methods in the LabVIEW front panel VI:

- Carrier Frequency (Hz)
- PRF (Hz)
- Duty Cycle (%)
- Sampling Rate (S/s)
- Input and Output Gain (dB)

Since RF-Frontend modules have an operating frequency from 200 MHz up to 4,4 GHz, the carrier frequency chosen by the user must be within those values. During the testing and analysis of the RF-Frontend modules, the carrier frequency was set to 1,3 GHz due to legal reasons, more about this can be found in the chapter 4.

The sampling rate is set to decide how good the sampling of the signal should be. The sampling rate is limited to the instantaneous bandwidth of 100 MHz from the RF-Frontend modules. The sampling rate limits also the minimum range of the radar system since the PRF cannot be higher than half of the sampling frequency, following the Nyquist-Shannon sampling theory.

The duty cycle can also be modified by the user of this radar application. The default value is 10% and it's recommend to keep the duty cycle unchanged and only modify the PRF to increase or decrease the distance range.

The input and output gain can also be set by the user. The default input and output gain values are 0 dB. Both can be increased to obtain a gain of up to +20 dB.

The received signals can be presented either in an A-Scope, see figure 6, or in an R-Scope, see figure 7. The difference between these two is that an A-Scope shows both the transmitting and listening time and the R-scope shows only the listening time or range.

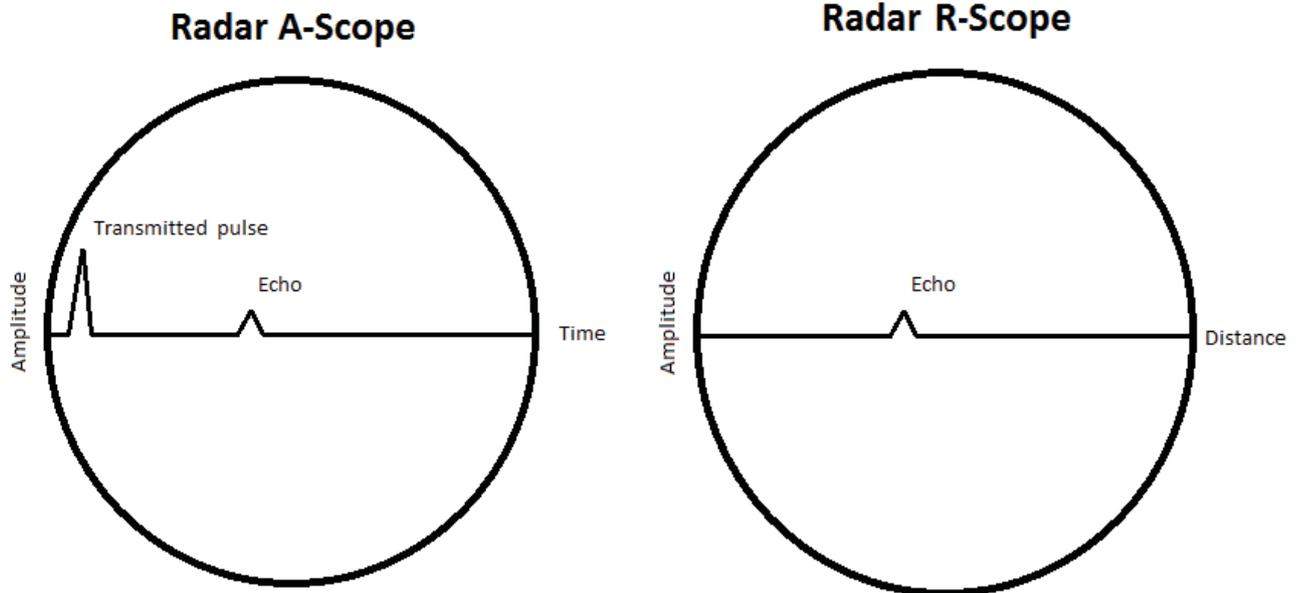


Figure 6: Example of an A-Scope

Figure 7: Example of an R-Scope

As mentioned earlier, the RF-Frontend modules can be controlled and monitored from the front panel of the radar VI built in this project. The VI can be opened with LabVIEW from the host computer. An example of the front panel for the non-MIMO method is shown in figure 8.

The front panel of the MIMO method looks almost the same as the one for the non-MIMO method. The only differences between these two methods are the codes that were made in the block diagram of the VI.

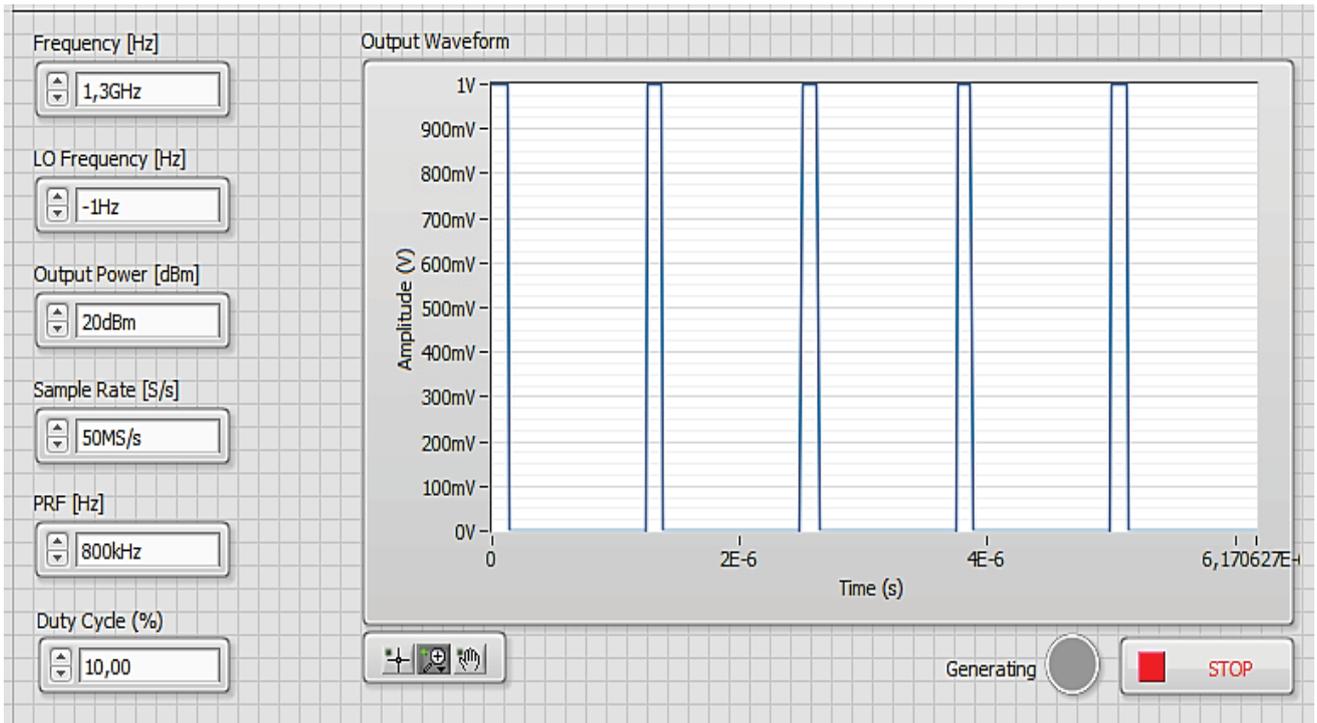


Figure 8: Example of Tx front panel for the non-MIMO method

The functions of this example VI can be changed in the future in order to adapt it to different radar technologies. For example different scopes can be added to this VI showing the bearing and speed of different objects. The Frequency Modulated Continuous Wave (FMCW) radar method can also be added to this VI as a possible option for the user.

Since this thesis project is divided in two different parts, the part of the RF-Frontend modules and how the radar application software was built, is further explained in the parallel thesis report “Software Development for a Radar Application Based on SDR Technology” [16].

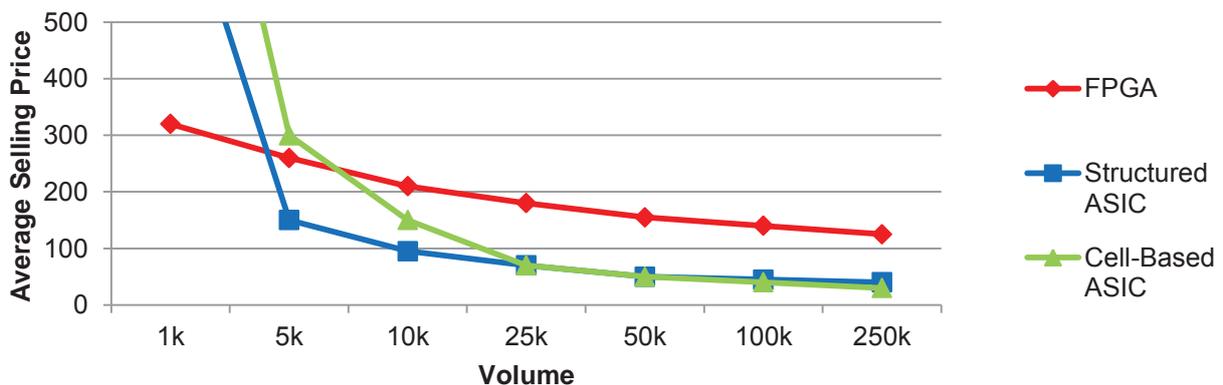
3.2 FPGA MODULES

As mentioned earlier, the main purpose of the FPGA modules in this application is to initialize and maintain the connection between the RF-Frontend modules and the host computer. They are also the ones processing the output and input signals of the radar system. All the data from the digital signal processing that communicates with the RF-Frontend modules and the host computer is stored in so called First In-First Out (FIFO) registers.

The FPGA's can also be used to run heavy calculations during the signal processing. Running heavy calculations on a host computer would require a Central Processing Unit (CPU) with an extreme performance in order to achieve the same power and force obtained by the FPGA's in this case.

The FPGA used in this application are programmed with the LabVIEW software tools. The advantage of using FPGA is that they can be reprogrammed at "run time" to do different processes and because of this their performance is much higher than most of the programmable devices on the market today.

A disadvantage with FPGA's is the high production cost per unit when used in high volume designs. If less than 5000 units of a product are manufactured with FPGA's, the production cost is still at a low marginal compared to other technologies as e.g. the Application Specific Integrated Circuit (ASIC). If the number of units manufactured with FPGA's exceeds 5000, it is more advantageous to use other technologies as ASIC's, see graph 1. Therefore, FPGA's are economically the best option when building a prototype, as in this project. If a future market analysis of this radar application shows that the number of units will exceed 5000 due to a higher demand, the FPGA cards can be replaced with e.g. ASIC cards.



Graph 1: Approximate cost per volume for FPGA's and ASIC's as shown in [12]

The big difference of the FPGA used in this application is that they cannot be programmed with the hardware description language VHDL, which is the most common way to program FPGA's. The only way to program the FPGA's is to use LabVIEW's graphical programming language, built specifically for FPGA programming. There are both advantages and disadvantages with this way of programming. These are further explained in chapter 4.

3.2.1 INITIALIZATION & CONFIGURATION

Two different clocks are used by the FlexRIO FPGA's. One clock of 40 MHz which is the Top-Level clock, and another clock of 200 MHz. The 40 MHz clock is the one steering the Register Bus and Configuration Loop, which is a required loop for the 7966R FPGA's combined with the 5791R RF-Frontend modules. This type loop is called Single-Cycle Timed Loop and it's very useful in LabVIEW FPGA programming, since it runs with the help of the clock selected, which in this case is the 40 MHz clock.

This required loop checks for errors and if the initialization of the RF-Frontend modules is done. The loop is also used to set-up the register bus and to retrieve and decode instructions from the host computer regarding the reading and writing of data in the FIFO's registers. The loop and its building blocks can be found in Appendix I.

3.2.2 SIGNAL PROCESSING

The FPGA code is divided in two clocked loops also known as Single-Cycle Timed Loops [13]. The first one is the required configuration loop mentioned in 3.2.1, and the second loop is the one built for digital processing, transmission and reception of the RF signals through the RF-Frontend modules.

This loop has two main processes, the Tx process for the transmission of the digital signals obtained from the host computer, and the Rx process for the reception of the digital signals obtained from RF-Frontend modules. These processes are both started with their respective trigger signals. The trigger signals are Boolean signals that start the transmission or reception on the RF-Frontend modules upon "rising edge". These signals should not be changed in order to have a functional synchronized system.

There are various trigger signals implemented in the code. All the trigger signals are connected to a Multiplexer and can be selected by the user. The trigger can be selected to manual mode, where the user can start the processes by manually switching a Boolean switch. The trigger can also be selected to automatic mode, where the processes are started automatically with the help of e.g. the implemented clocks or counters. An example of the trigger start method for the Tx transmission process is shown in figure 9.

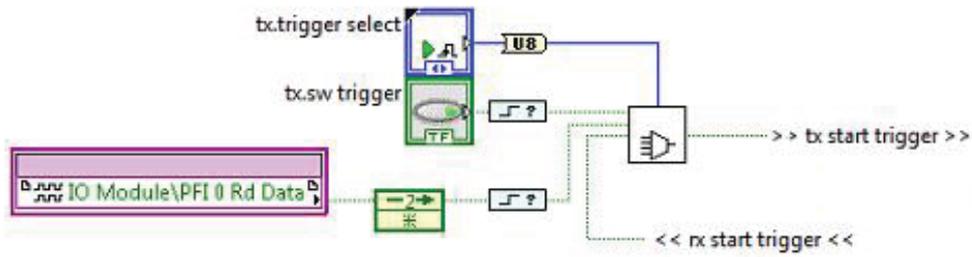


Figure 9: Example of the trigger start method for the transmission process

3.2.2.1 TX PROCESS

After the trigger for the transmission process is set, the Tx or transmission process starts. The purpose of the Tx process is to read the I/Q data from the host computer that's accumulated in a FIFO register. The I/Q data contains information in digital form about the changes in amplitude and magnitude of the transmission wave pulse signals obtained from the host computer software. This I/Q data is later digitally controlled and sent through different signal processing blocks. The I/Q data is finally sent to the RF-Frontend module for digital to analog conversion and transmission.

The first block after the trigger system in the Tx process is the Output Stream Control block. This block is the one controlling and regulating the whole Tx process after the trigger is set, see figure 10. The main purpose of this block is to enable and disable the FIFO stream data reading. It also keeps track for different errors, e.g. it aborts the reading if an underflow occurs in the stream generation.

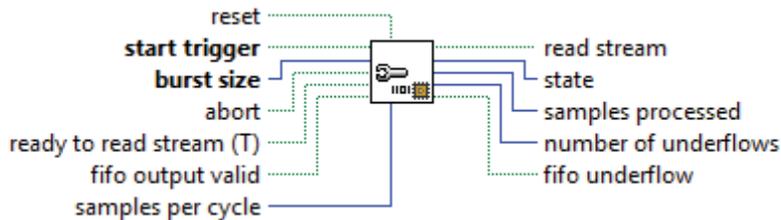


Figure 10: Example of the Output Stream Control VI block

After the Output Stream Control block checks for errors, a Boolean signal is sent to the next block FIFO Read to enable it to read data from the host computer that's stored in a FIFO register. The output data from this block is obtained in form of a 32-bit unsigned array, see figure 11. This 32-bit array is later converted to a special string of I/Q data that is needed for further Digital Signal Processing, see also figure 11.

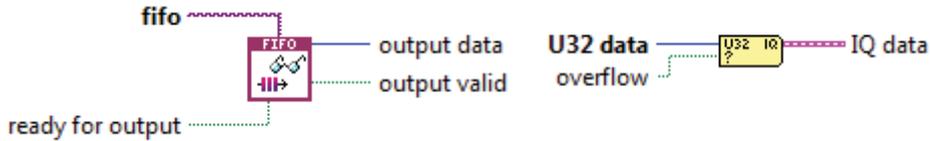


Figure 11: Example of the FIFO Read VI block and the I/Q data conversion VI block

After I/Q data is obtained from the FIFO register in form of I/Q data string, the DSP of this data starts. The first part of the DSP in this process is to control the gain from the measured I/Q data. This gain is the one selected from the user on the host computer application. The VI block used for this is called DSP Digital Gain and its purpose is to digitally control both I and Q signal levels found in figure 12. This block provides the following functionality where (*) represents scalar multiplication:

$$\text{DataOutI} \leq \text{DataInI} * \text{GainI}$$

$$\text{DataOutQ} \leq \text{DataInQ} * \text{GainQ}$$

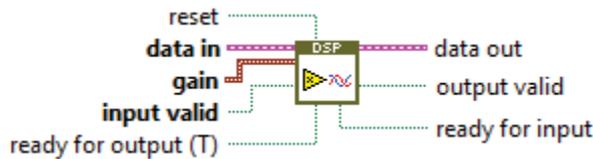


Figure 12: Example of the DSP Digital Gain VI block

After the Digital Gain processing, a block called DSP Fractional Interpolator is used to interpolate the input signal from the Digital Gain block according to the input sample rate and the delay, see figure 13. This block uses a Finite Impulse Response (FIR) filter in order to do this signal processing.

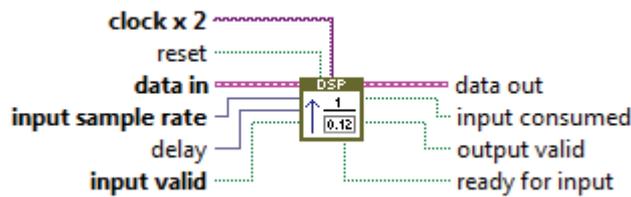


Figure 13: Example of the DSP Fractional Interpolator VI block

When the Fractional Interpolator finishes the processing, the data is sent to the DSP Frequency Shift block, see figure 14. The purpose of this block is to apply a digital frequency shift to the I/Q data. The block contains both cosine and sine signals which are multiplied with the I/Q data using a complex multiplier.

This block is useful in order to shift the complex spectrum to the left or to the right in the frequency domain with the following functionality where (*) represents scalar multiplication:

$$\begin{aligned} \text{DataOutI} &\leq \text{DataInI} * \text{Cosine} - \text{DataInQ} * \text{Sine} \\ \text{DataOutQ} &\leq \text{DataInI} * \text{Sine} - \text{DataInQ} * \text{Cosine} \end{aligned}$$

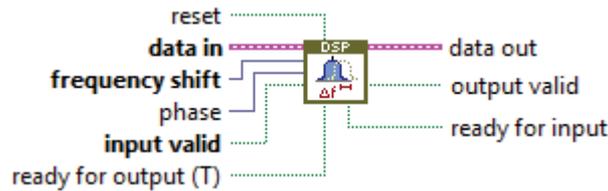


Figure 14: Example of the DSP Frequency Shift VI block

Data from the Frequency shift block goes through one last DPS block before it's forwarded to the RF-Frontend module. The last signal processing block is the DSP Impairments block. The purpose of this block is to modify the I/Q data in order to apply signal impairments. The block is shown in figure 15.

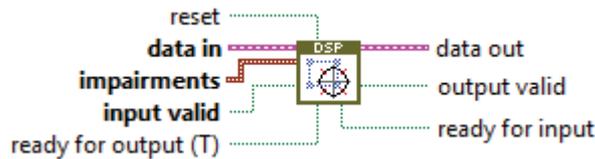


Figure 15: Example of the DSP Impairments VI block

After this final DSP block, I/Q data from data string needs to be divided in to two 16-bit arrays of integer data. One 16-bit integer array will consist of the Tx I data and the other 16-bit integer array will consist of the Tx Q data. Finally, if the whole Tx process and the digital signal processing went well, a Boolean signal is set and these two 16-bit arrays are forwarded to the RF-Frontend modules, see figure 16. Once the RF-Frontend module receives these arrays, the digital data is automatically converted to analogue form and then broadcasted over the air with the help of antennas.

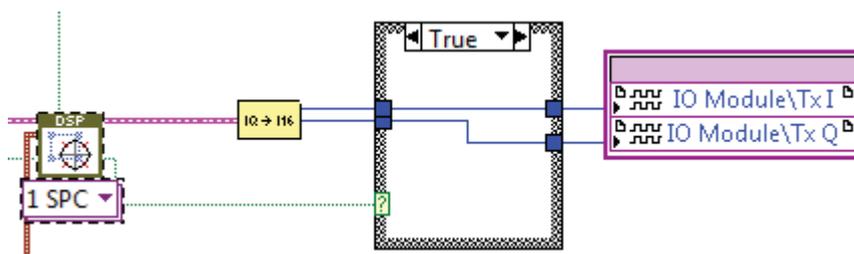


Figure 16: Final part in the Tx Process

3.2.2.2 RX PROCESS

The purpose of the Rx process is to read the obtained digital signals from the RF-Frontend module and after some basic DSP, forward these signals to the host computer for further analysis. Although the Rx process uses the same VI blocks as in the Tx process except for the DSP Digital Gain VI block, the Rx process is a little bit simpler. The difference is that the Rx process starts with the DSP after obtaining the signals from the RF-Frontend module, and at the end it checks for errors and underflows before forwarding the data to the FIFO register from where it communicates with the host computer.

The trigger in the Rx process works exactly as the one in the Tx process. After the trigger is set the Rx process starts reading I/Q data from the RF-Frontend module. The data from the RF-Frontend module are two 16-bit integer arrays, one for the I signal and one for the Q signal. These arrays are converted to a special string of I/Q data which is required in order to do the DSP, see figure 17.

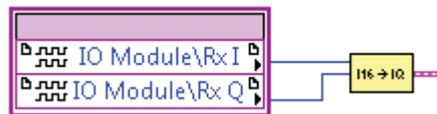


Figure 17: Reading and conversion of the I/Q signals from the RF-Frontend module

After I/Q signals are converted in to the I/Q string, the DSP starts. The DSP blocks used are the same ones used in the Tx process, except for the DSP Digital Gain VI block. The first block used is the DSP Impairments VI block with the purpose to modify the I/Q data in order to apply signal impairments. The VI block is shown in chapter 3.2.2.1 on figure 15.

The second DSP block used is the DSP Frequency Shift VI block, with the purpose to apply a digital frequency shift to I/Q data. This block is further explained in chapter 3.2.2.1 and is shown on figure 14.

The third and last DSP block is the DSP Fractional Decimator block. It works as the DSP Fractional Interpolator block used in the Tx process. The only difference is that this block is used with the purpose to decimate I/Q data from the previous DSP block according to the output sample rate and delay, see figure 18. This block uses also a Finite Impulse Response (FIR) filter.

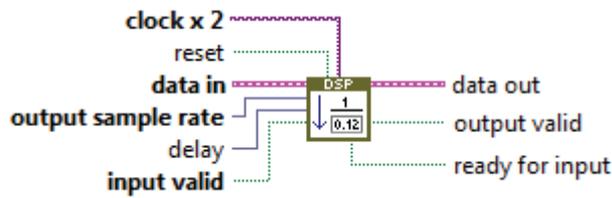


Figure 18: Example of the DSP Fractional Decimator VI block

After the DSP is finished, I/Q data from the string is converted to a 32-bit unsigned array in order to write it to the FIFO register that's also used by the host computer. After the conversion the Tx process uses a VI block called Input Stream Control in order to check for errors and overflows, see figure 19.

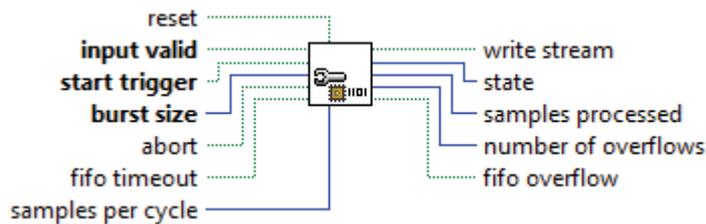


Figure 19: Example of the Input Stream Control VI block

After the Input Stream Control VI block checks for errors and overflows, a Boolean is set enabling I/Q data to be written to the FIFO register. The VI block used for writing the 32-bit unsigned array with I/Q data to the FIFO register is called FIFO Write VI block, see figure 20.

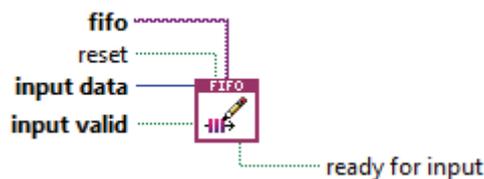


Figure 20: Example of the FIFO Write VI block

As mentioned earlier, this FIFO register can be accessed from the host computer. The host computer reads I/Q data from the FIFO register and presents it in different graphs. The data can be selected by the user to be shown in an A-Scope or R-Scope, see figures 6 and 7 for more information about these graphs.

3.2.3 MIMO & NON-MIMO METHODS

MIMO stands for Multiple Input Multiple Output and the purpose is to run the same processes synchronously on several inputs and outputs. Or as in this case several RF-Frontend modules. As mentioned earlier there are two ways to run this radar application. One way is to run the two RF-Frontend modules simultaneously where one module is the transmitter and the other module is the receiver. This method is the non-MIMO method, since each RF module runs a different process. The second way is to use both RF modules as transceivers, where they synchronously change from transmitters to receivers. This method is the MIMO method, since they synchronously run the same processes at the same time.

Since the purpose with this radar application is to use an array of antennas, the MIMO method is the only method suitable for this. In the non-MIMO method, one antenna is transmitting signals and the other one receiving signals at the same time. This means that that an array of antennas cannot be implemented in the non-MIMO method, see chapter 3.3.

In the non-MIMO method the user chooses the type of signal that should be broadcasted by the radar. This signal is constantly written to the FIFO register from the host computer in order for the FPGA to read it and forward it to the RF module. In the non-MIMO method, the signals written to the FIFO are basically arrays of 10% “ones” and 90% “zeros”, depending on the duty cycle. This means that no counter is used in the FPGA for the non-MIMO method. The FPGA continuously reads and forwards these signals.

In the MIMO method a constant signal of only “ones” is written to the FIFO by the host computer. In this case, the FPGA has an implemented counter that switches between transmission and reception after a specific amount of time depending on the PRF selected by the user. This means that 10% of the time in one PRT, the FPGA is reading signals from the FIFO and transmitting these signals through the RF module. The other 90% of the PRT, the FPGA works as a receiver writing signals to the FIFO. The host computer reads the signals from the FIFO and displays them in form of graphs. The length of the counter is decided with the PRT selected by the user on the host computer.

In the non-MIMO method, the transmission and reception processes are started and aborted simultaneously by the user. In the MIMO method, transmission and reception counts as one single process. The user starts and aborts the whole process, but the transmission and reception in this process are started and aborted automatically by the counter. This is done by the Output Stream Control VI block and by the Input Stream Control VI block, which can be found on figure 10 and figure 19 in the previous chapter.

3.3 ARRAY OF ANTENNAS

In this radar application, four Log-Periodic PCB LP0965 antennas are used to transmit and receive the radar signals. As mentioned before, the purpose of using four antennas is to increase the directional gain of the antennas. If several antennas are used as an array, the directional gain will increase up to a factor of N , where N is the number of identical elements in the array [14].

There are several advantages of using arrays of antennas in radar applications. One of the advantages is that the radiative signals may be enhanced and it may also be used to null the radiation pattern in one particular direction. This benefits the radar system to obtain higher echoes and if built properly, it may also be used to face of particular both hostile and non-hostile interfering sources.

Since no rotating system is used in this project to steer the antennas and their direction, another benefit with an antenna array is that the physical position may be fixed but the array may still be steerable. This can be obtained by changing the phase of the waves sent through particular elements in the array, see figure 21. This has not been implemented in this radar application, but this could be implemented in the future by making some changes in the software VI code.

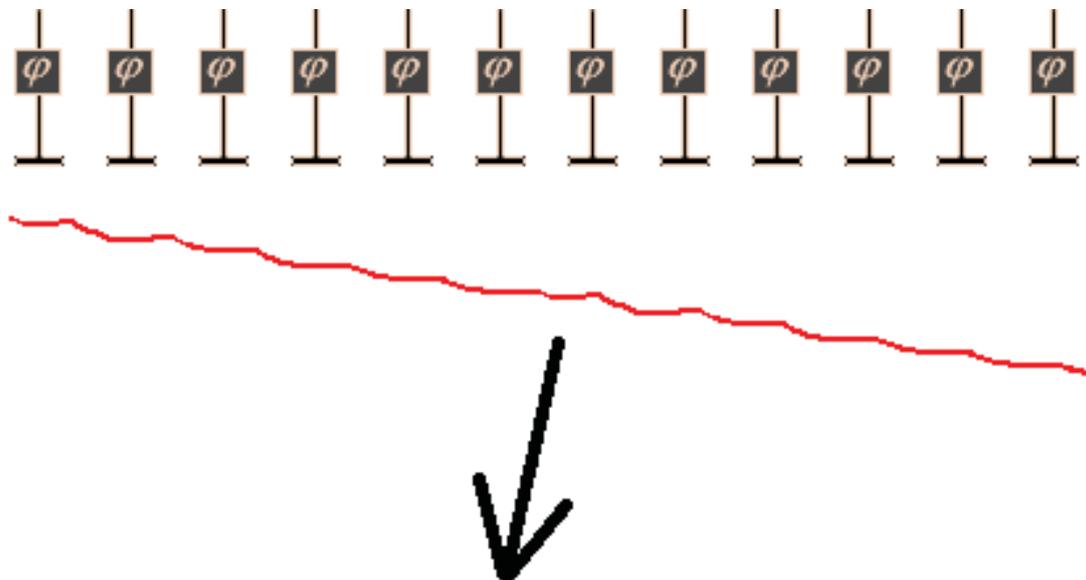


Figure 21: Result of particular element phase changing in antenna arrays

Since there is no shared I/O on the RF-Frontend modules to be used as both transmitter and receiver, two antennas are used in the non-MIMO method, one for Tx and one for Rx, and all four antennas are used in the MIMO method, two for Tx and two for Rx. This means that the array of antennas can only be implemented in the MIMO method.

There are two different arrays of antennas. The first one is the linear array of antennas, where all the elements in one vertical line have the same phase shifter. The second one is the planar array of antennas, where every element has its own phase shifter. The planar array of antennas is the one built in this project, although a linear array of antennas can also be used just by making a small change in the software VI application of the radar system.

In order to achieve a properly working antenna array, the elements or antennas must be placed on a specific distance from each other. When using two antennas as in the MIMO method, the proper distance between each antenna was calculated to be $\lambda/2$. Where λ is the wavelength of the operating frequency calculated with formula (3.1).

$$(3.1) \lambda = \frac{c_0}{f_{OP}}$$

Where:

c_0 = the speed of light ($\sim 3 * 10^8$ m/s)

f_{OP} = the operating frequency of the radar (Hz)

Since an operating frequency of 1,3GHz was primarily used in this project due to legal and security reasons, the distance between the two antennas was calculated to be $\approx 11,5$ cm. This helped reduce the beam-width and increase the directional gain, see figure 22.

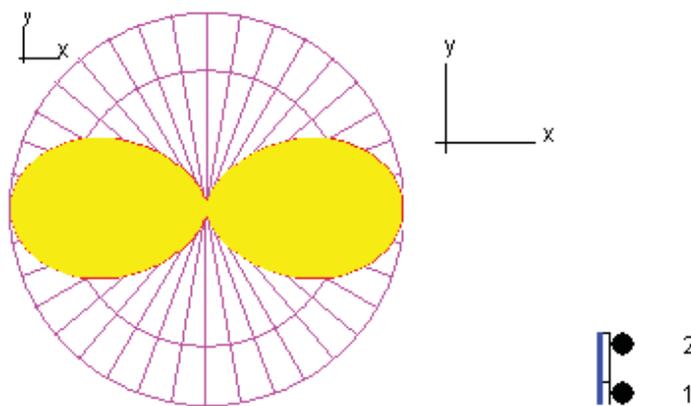


Figure 22: Beam-width for $\lambda/2$, with two elements and 0^0 phase difference between elements [15]

In the future, the number of elements can be increased which will result in an even smaller beam-width and a higher directional gain, see figure 23.

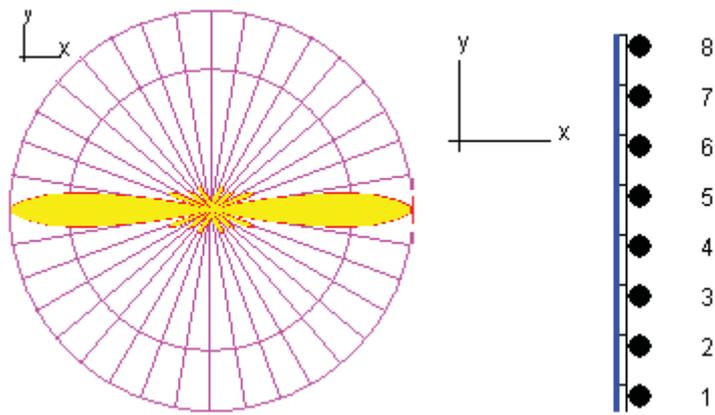


Figure 23: Beam-width for $\lambda/2$, with eight elements and 0° phase difference between elements [15]

RESULTS

4.1 HARDWARE PERFORMANCE

Since this radar application has to work at high speeds, the performance of the hardware is the key to obtain good results. As mentioned in earlier chapters, the maximum sampling rate is decided by the performance of the hardware. The maximum sampling rate then decides the minimum distance range of the radar system.

Since the radar system built in this project has a very low output power, the only way to receive back an echo is by looking at objects that are close to the antennas. In order to achieve this, a fast PRF is needed. This means that the sampling frequency or sampling rate should be twice as high as the PRF used, following the Nyquist-Shannon sampling theorem. And this requires a good and fast hardware performance.

The hardware had a different performance than the one expected. During the testing of the non-MIMO method, a carrier frequency of 2,3 GHz, a PRF of 800 kHz and a sampling rate of 50 MS/s were used. This setup generated and transmitted signals as expected but the only problem was that the system occasionally experienced some transmission breaks. These occasional transmission breaks were a couple of microseconds short and happened due to overflows and underflows in the FIFO registers. As fast as the system became stable and the overflows and underflows were eliminated, the system automatically returned to its idle mode and started transmitting signals again. These breaks partially affected the object measurement testing, but not up to the level that the system was unusable and showed false measurements values.

These transmission breaks happened due to the sampling rate. The maximum sampling rate that the system could handle was 50 MS/s. Selecting a higher sampling rate would result in an unusable system where the transmission doesn't even start. Using this maximum sampling rate of 50 MS/s means that the data is collected every 0,02 μ s.

The system was also tested with a lower PRF and a lower sampling rate. This resulted in a fully functional system, but selecting a low PRF resulted in a large minimum distance range where the output power was too low for the system to obtain an echo. As mentioned earlier, these transmission breaks did not affect the measurements but they can still not be tolerated in sophisticated radar systems.

The only solutions for these transmission breaks in order to obtain a fully functional system are either to lower the sampling rate and lower the PRF and also increase the output power with some external amplifiers. The other solution is to completely replace the FPGA cards and the RF-Frontend modules with devices capable of running at the higher speeds needed. This is further explained in chapters 5.1 and 5.2.

4.2 DATA ANALYSIS

As mentioned earlier, several distance measurements were made during the hardware and software programming. The carrier frequency which is the operational frequency of the radar system had to be selected within a specific frequency range. This was because the radar system should not interfere with any other already operating systems.

The distance measurements tests were made on different frequencies both with and without the help of LNA's. The only method used for the testing was the non- MIMO method. The reason for this is explained in chapter 5.1. For these tests, the LP antennas were attached to a styrofoam base and manually pointed at the measured objects, see figure 24.

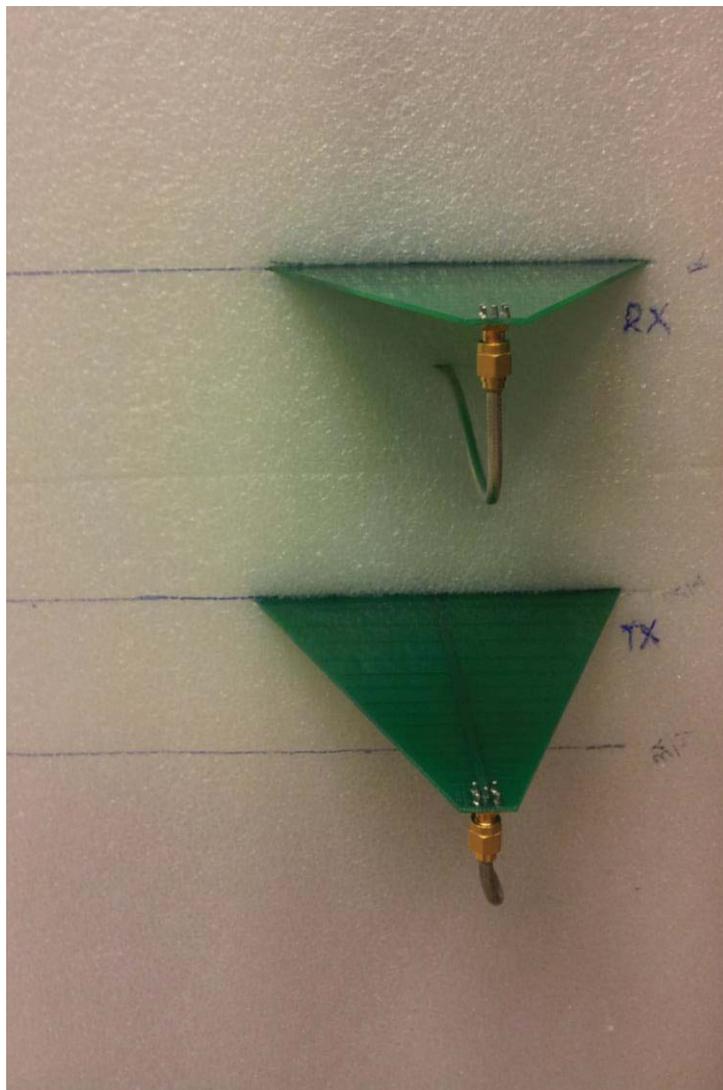


Figure 24: LP antennas used for the distance measurements

The measured object had to be a big metal object in order to receive back an echo. Therefore, the object used for all measurements in this project was a big metal garage door, see figure 25.



Figure 25: Metal garage door used for the distance measurements

The echoes from the distance measurements of the metal garage door are presented in A-Scope graphs on the front panel of the software VI on the host computer. The data on the graphs is later analyzed and the distance is manually calculated by looking at the time between the start of the transmission to the start of the reflected echo. The distance between the LP antennas and the garage door was also manually calculated with a meter measuring tool to about 70 meters. This distance corresponds to a time measurement of approximately $0,47 \mu\text{s}$.

Since the data is collected every $0,02 \mu\text{s}$ and the expected measured time is $0,47 \mu\text{s}$, the uncertainty of the measurement can be calculated to 4,3% using formula 4.1.

$$(4.1) \sigma = \frac{\Delta t}{T_M} * 100\%$$

Where

σ = the measurement uncertainty

Δt = the time between samples

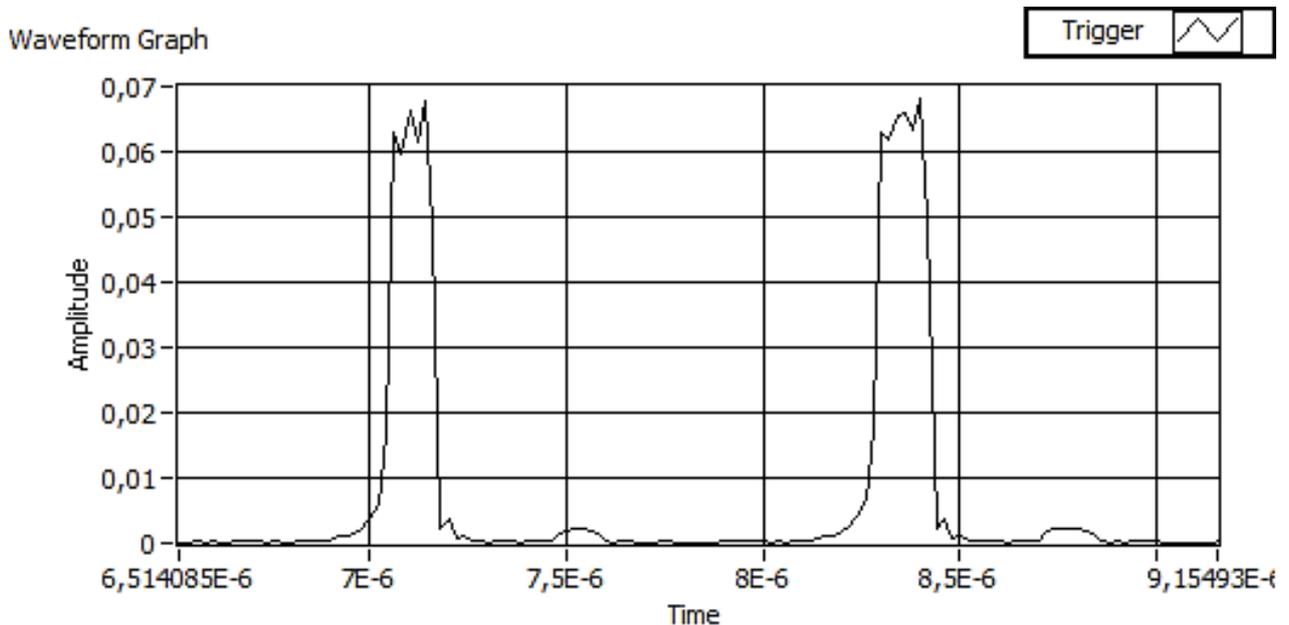
T_M = the measured time

With this, the uncertainty in the distance measurements is roughly estimated to 4,3% of 70 meters, which gives an uncertainty of about 3 meters.

In all tests performed, the input gain was selected on the software VI to its default value of 0 dB. The output gain was selected to its maximum value of 20 dB. The reason for this was because the transmitted signals should be selected to the highest power possible in order to obtain echoes, since the default values have a very low output power. But increasing the input gain will only end up increasing the background noise which will mix with the echoes and give results that would be difficult to read. Therefore the input gain is set to its default value.

4.2.1 MEASUREMENT TEST 1

The first test at a distance of about 70 meters was made with a carrier frequency of 900 MHz and with a PRF of 800 kHz. An LNA working within the frequency range from 5 MHz to 1 GHz is also used during this first test. The LNA was connected between the receiving LP antenna and the Rx input on the RF-Frontend module. The result of this first test is shown in graph 3.



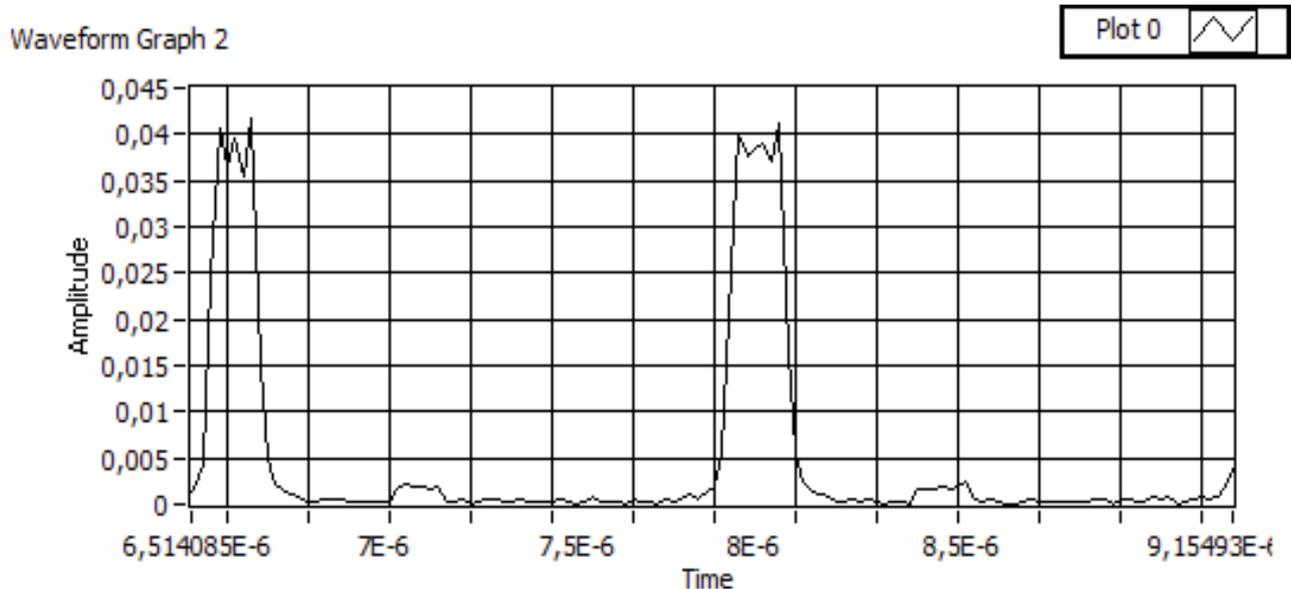
Graph 3: Result of the data analysis from the first measurement test

An echo from the garage door, with a very low reflected power, is seen in the A-Scope graph. The distance results from this graph are manually measured by looking at the time from the start of the transmission to the start of the echo, which in this first test is about 0,46 μ s.

Now with the help of formula (2.1), the distance between the antennas and the garage door can be calculated to about 69 meters, giving the final result within 69 ± 3 meters. Knowing that the distance between the antennas and the garage door was calculated to about 70 meters with the help of a meter measuring tool, the results from this first test can be considered successful.

4.2.2 MEASUREMENT TEST 2

The second test at a distance of about 70 meters was made with a carrier frequency of 1,3 GHz and with a PRF of 800 kHz. For this test no LNA was used. The reason for this is because the two LNA's used in this project cannot operate on a frequency of 1,3 GHz. This test is primarily made to see how good the result would be without an LNA. The result of this first test is shown in graph 4.



Graph 4: Result of the data analysis from the second measurement test

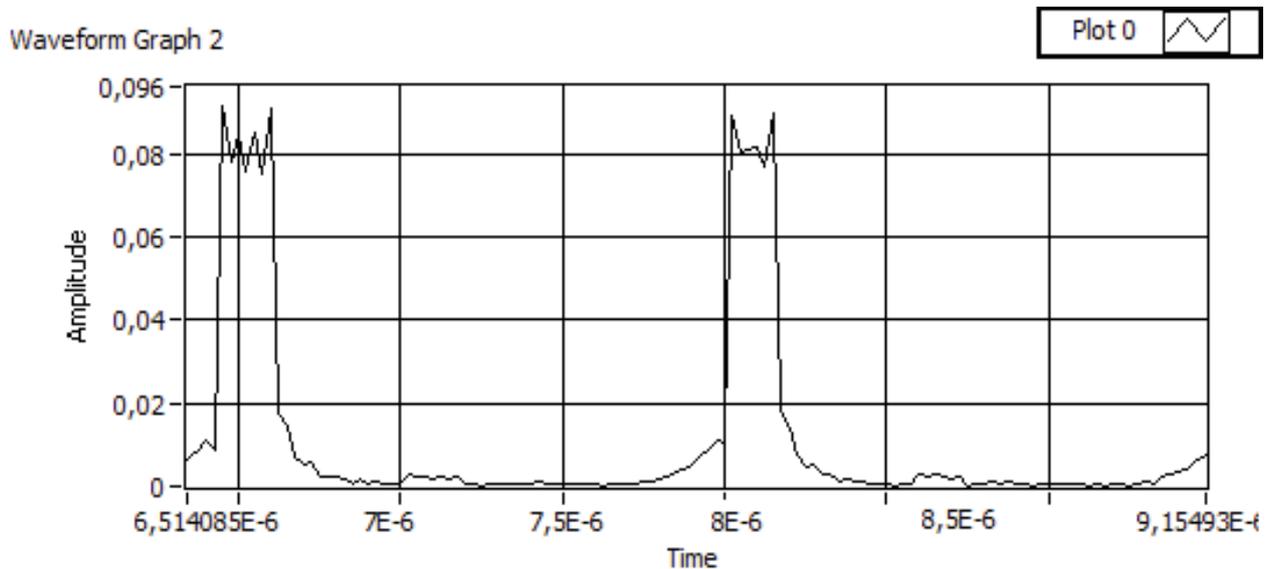
An echo from the garage door with a very low reflected power compared to the one in graph 3 can also be seen in this A-Scope graph. The distance results from this graph are manually measured by looking at the time between the start of the transmission and the start of the echo, which is the same as in the first test, about 0,46 μ s.

As in the first test, the distance is manually calculated to about 69 meters with the formula (2.1). The second test was also successful since the final result is within 69 ± 3 meters.

One of the differences in the results, from the first and the second test, is that the signals have a lower power. The signal power is almost twice as low in this test compared to the result from the measurement test 1. This difference is because an LNA is not used in this second test. This implies also a higher background noise mixed with the echo signals. By comparing the two graphs, the echoes in graph 4 have a higher mixed noise than the ones in graph 3.

4.2.3 MEASUREMENT TEST 3

The third test at a distance of about 70 meters was made with a carrier frequency of 2,3 GHz and with a PRF of 800 kHz. For this test an LNA working within the frequency range from 2 GHz to 4 GHz is also used during this third test. The LNA was connected between the receiving LP antenna and the Rx input on the RF-Frontend module. The result of this first test is shown in graph 5.



Graph 5: Result of the data analysis from the third measurement test

An echo is also seen in this A-Scope graph. The power of the signals is twice as high compared to the signals from graph 4. This is due to the use of the second LNA. The distance results from this graph are also manually measured by looking at the time between the start of the transmission and the start of the echo, which is the same as in the first and second test, about 0,46 μ s.

As in the other measurement test, the distance is manually calculated to about 69 meters with the formula (2.1). The third test was also successful since the final result is within 69 ± 3 meters.

The difference between this test and the others, except for the higher output power, is that the rise and fall time for the transmitted pulses is definitely longer. The reason for this is because the carrier frequency has increased. The RF-Frontend modules need more time to build these pulses when a higher operating frequency is used.

PROBLEMS AND DISCUSSION

5.1 PROBLEMS

During the programming and testing, two severe problems were detected apart from other small problems. The first severe problem that was detected during the measurement testing is that the system experienced short breaks during the signal transmission. These breaks occurred because of the underflows and overflows that happened due to the high PRF and sampling rate that were used. The second severe problem happened during the programming of the MIMO method on the FPGA cards and it's the reason to why results are only shown for the non-MIMO method. The problem was that the LabVIEW FPGA code compiler suddenly crashed and didn't start working again.

5.1.1 TRANSMISSION BREAKS

The transmission breaks that occurred due to the underflows and overflows happened because of the high sampling rate and PRF selected. This transmission break problem disappeared when the sampling rate and the PRF were reduced. This problem happened when the maximum sampling rate of 50 MS/s was selected. The reason for choosing this sampling rate was because of the high PRF of 800 kHz. The better the sampling rate was in this case the better the transmitted signal was going to be. This high PRF was needed in order to measure objects at a close distance range, since the systems output power is too low in order to measure objects at large distances.

These transmission breaks did affect partially the measurement testing, but not up to the level where the system was completely unusable. Since the live data stream on the graphs had to be paused in order to measure the time between the transmitted signals and the echoes, sometimes the pause could occur during a transmission break. When this happened, the only thing visible in the graphs was the background noise.

5.1.2 COMPILER ERROR

The compiler error was the most severe problem during this project. The compiler suddenly crashed and stopped working. The compiler never showed an error ID. Appendix II shows how the compiler looked after the crash. Every time the compiling was restarted, the compiler looked the same. Even a whole system reboot was tried but without success the compiler looked still completely dead and not responding.

A lot of help has been obtained from the National Instruments support team regarding this problem, but a complete solution to this problem was never found. This problem affected the MIMO method programming. Sadly, due to this "bug" in the compiler system the programming and testing of the MIMO method never went through. The reason for this problem is still unknown.

5.2 SOLUTIONS

A lot of small problems were detected during the programming and every problem had a solution. The only two severe problems without specific solutions were the transmission breaks and the compiler error.

5.2.1 TRANSMISSION BREAKS

One solution to the transmission break problem is to decrease the sampling rate and the PRF selected by the user in the host computer. For this solution an external amplifier is needed since the default output power is too low for large distance measurements.

The other solution to this problem is to upgrade the FPGA cards and the RF-Frontend modules to devices that could handle these speeds without giving underflow and overflow errors.

5.2.2 COMPILER ERROR

Since the reason behind the compiling error is still unknown, a specific solution to the problem has not yet been found. The only possible solution is to wait until the next LabVIEW FPGA update from National Instruments is released.

A lot of help has been obtained from the National Instruments support team. Their suggested to try the following solutions:

- To compile an FPGA code only on the FlexRIO card, this means making a new project without the so called CLIP nodes
- Installing LabVIEW on another pc and try compiling the code with the new installed compilation
- If the above solutions do not work, try and reinstall everything on the host pc and start from the beginning

Sadly, none of these solutions were successful. The only option left is to wait for the LabVIEW update in order to finish the MIMO method and to continue with the FPGA programming. Unfortunately, this update from National Instruments is expected after the thesis submission.

SUMMARY AND CONCLUSIONS

Based on results, the hardware did partially meet the expectations and requirements for SDR radar system performance. Although the system experienced transmission breaks as discussed in chapter 5.1, the system has shown to be fully operational with successful echo verification and ranging on distances greater than 50 meters and less than 100 meters.

In order to measure objects at distances above 100 meters, external components such as amplifiers are needed in order to increase the power of the signals.

Low cost alternatives and performance wise preferable hardware can be used in future developments. The FPGA average selling price is drastically decreasing as the order quantity increases every year. Therefore, SDR systems with high performance FPGAs can be used as fully functional radar systems in the near future.

6.1 ENVIRONMENTAL & ETHICAL ASPECTS

The system in itself uses frequencies and signal amplitudes that are commonly found in most households and offices in technologies such as WIFI routers and any mobile radio transmitter units, one cannot with any certainty claim that the system is harmful to human health and our environment.

This leaves room for future arguments given that long-term studies (15-20 years) have yet to show the results for these, today widely debated, frequencies vs. side effects.

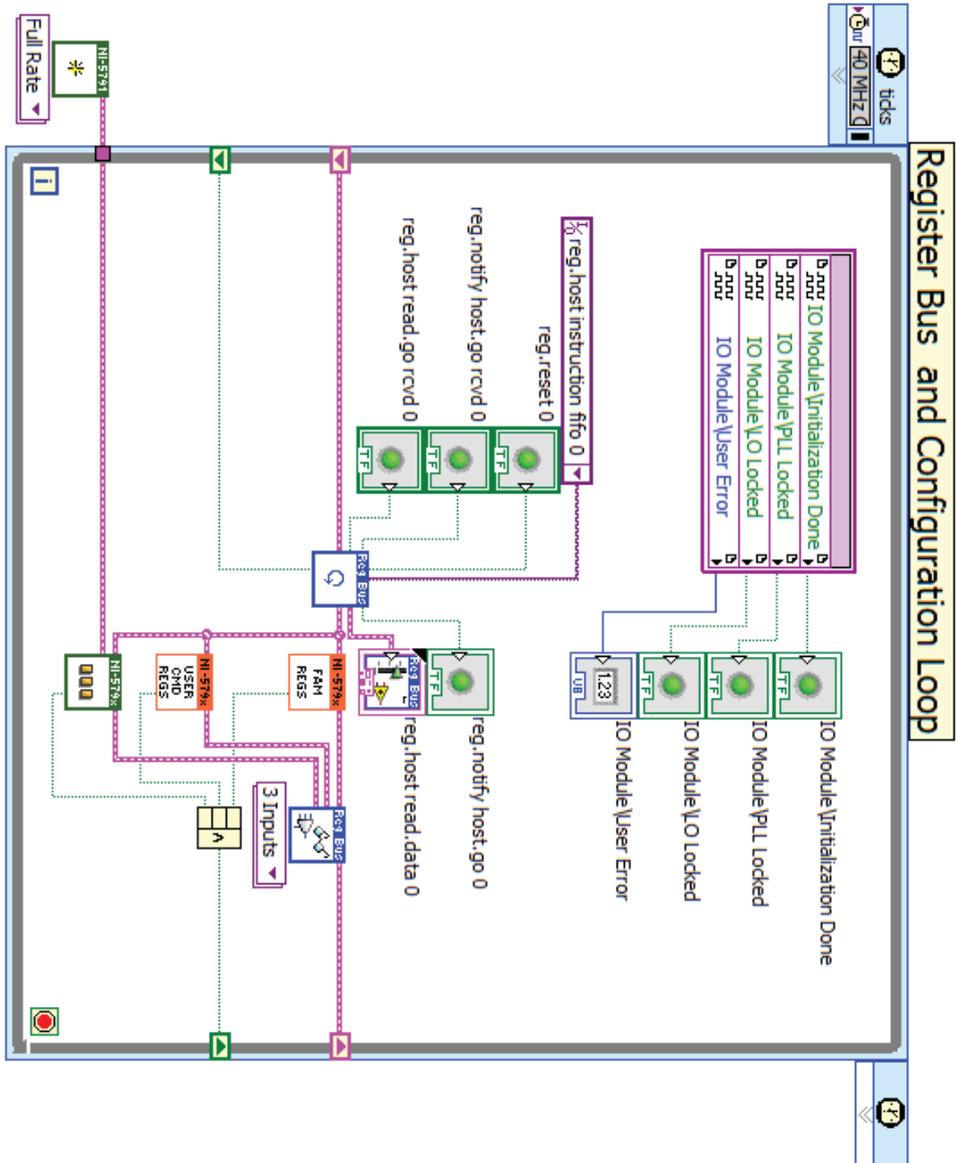
REFERENCES

- [1] M. S. Russegren, E. Fehratovic, et al. (2014), *Software Defined Radar*. Göteborg: Chalmers University of Technology
- [2] National Instruments Corporation (2010), *NI PXIe-8133 User Manual*. Austin, Texas, USA: National Instruments Corporation
- [3] National Instruments Corporation (2012), *NI 5791R User Manual and Specifications*. Austin, Texas, USA: National Instruments Corporation
- [4] National Instruments Corporation (2011), *NI FlexRIO FPGA Module Installation Guide and Specifications*. Austin, Texas, USA: National Instruments Corporation
- [5] Kent Electronics (2012), *PCB-LP Antenna*. Texas, USA: Kent Electronics
- [6] Adams-Russell Co., *Models AM-/AMC-174 5-1000 MHz Specifications and Operating Characteristics*. Burlington, USA: Adams-Russell Co.
- [7] Hewlett-Packard, *Wideband Small Signal Amplifiers 2.0 to 4.0 GHz*. Palo Alto, USA: Hewlett-Packard
- [8] National Instruments Corporation (2010), *Introduction to LabVIEW Graphical Programming Hands-On Seminar*. Austin, Texas, USA: National Instruments Corporation
- [9] Gaspare Galati (1993), *Advanced radar techniques and systems*. London, United Kingdom: Peter Peregrinus Ltd.
- [10] S. Kingsley, S. Quegan (1992): *Understanding Radar Systems*. McGraw-Hill Book Company

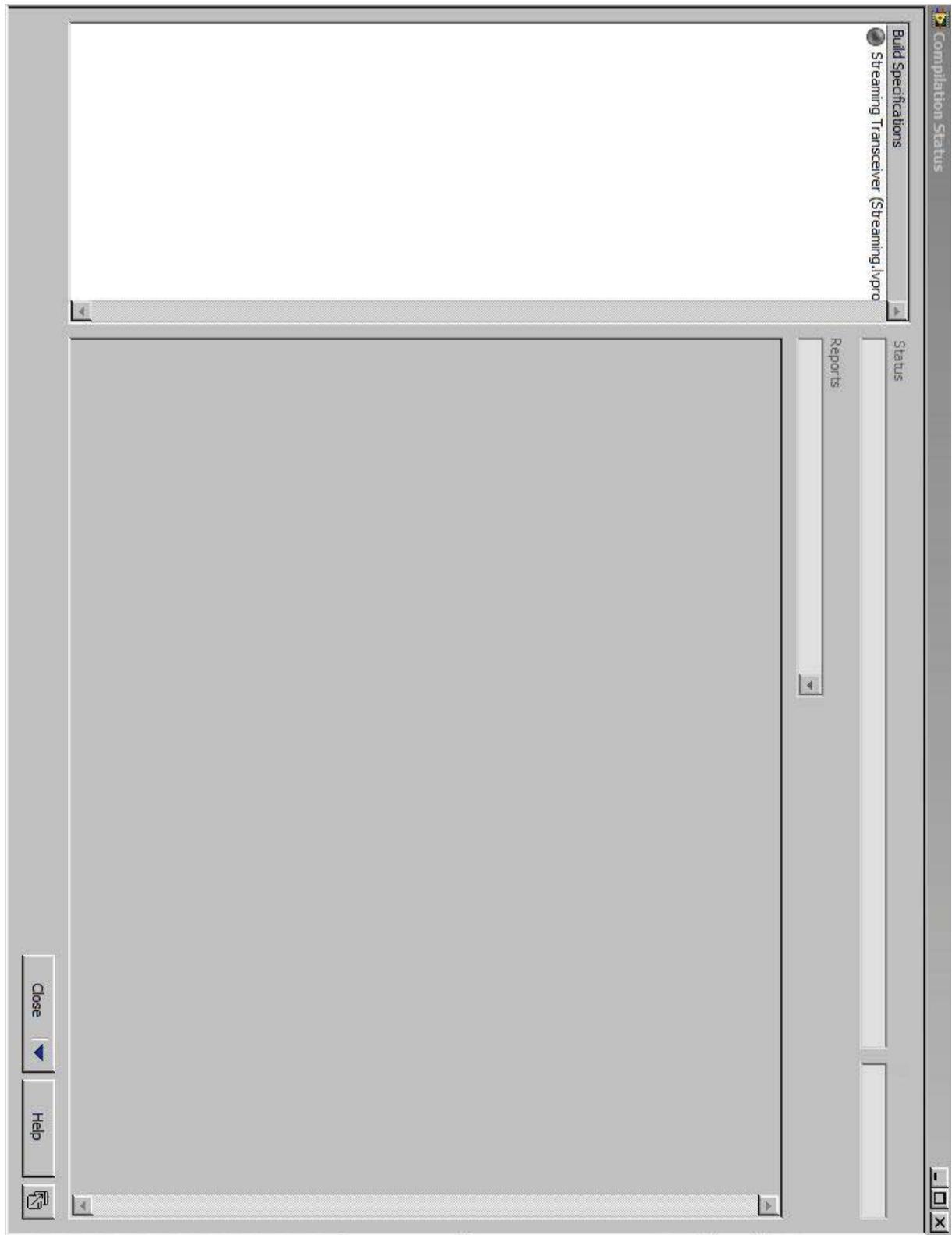
- [11] Hosking, R. (2014) Software-Defined Radio Handbook. *EDN Europe*. (2014-02-01)
http://www.edn-europe.com/en/software-defined-radio-handbook.html?cmp_id=34&news_id=10003220&vID=1565&from_mail=1#.U42lpPI_tch
(2014-05-23)
- [12] Mosher, R. FPGA Prototyping to Structured ASIC Production to Reduce Cost, Risk & TTM. *Design & Reuse*. <http://www.design-reuse.com/articles/13550/fpga-prototyping-to-structured-asic-production-to-reduce-cost-risk-ttm.html> (2014-05-25)
- [13] National Instruments Corporation (2004), *LabVIEW FPGA Module User Manual*. Austin, Texas, USA: National Instruments Corporation
- [14] Wulf Dieter (2001): *Radar techniques using array antennas*. London, United Kingdom: The Institution of Electrical Engineers
- [15] Tony Townsend, T.T. Linear Arrays. *Università di Catania*
<http://www.diit.unict.it/users/campi/antennas/Antenna.html> (2014-05-20)
- [16] A. Stockman, N. Glavmo (2014), *Software Development for a Radar Application Based on SDR Technology*. Göteborg: Chalmers University of Technology

APPENDICES

APPENDIX I



APPENDIX II



APPENDIX III

PXI System comprising of two sets of FPGAs and RF-Frontend modules mounted on PXIe computer chassis



APPENDIX IV

Shows the PXI system when operational

