Hochschule Magdeburg-Stendal
Fachbereich Ingenieurwissenschaften und Industriedesign (IWID)
Institut für Elektrotechnik

Masterarbeit

zur Erlangung des Grades eines "Master of Engineering" im Studiengang Gebäudesystemtechnik

Thema: **"Erstellung von Interferenzprofilen mithilfe einer Software Defined Radio Plattform"**

Eingereicht von: **Marc Rabsilber**

Angefertigt für: Institut für Automation und Kommunikation Magdeburg eV.

Matrikel: 2012 3191

Ausgabetermin: 20. September 2017

Abgabetermin: 09. November 2017

Schulischer Betreuer: Herr Prof. Dr. Sebastian Hantscher

Betrieblicher Betreuer: Herr Dr. Lutz Rauchhaupt

....

1. Prüfer 2. Prüfer

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die Masterarbeit selbstständig angefertigt und alle von mir genutzten Quellen und Hilfsmittel angegeben habe.

Magdeburg, den 10.11.2017 ……

Abstract

The experimental research on wireless technologies demands full access to all network layers. Pre-built hardware solutions offer, if any, only limited access to these layers. Alternative choices are software defined radios, which feature RF-hardware and reprogrammable components for a wide array of applications.

In this thesis, the software defined radio USRP X310 from Ettus is used to emulate the occupation of the wireless transmission medium by communication systems like those specified by IEEE 802.11. Its ability to influence communication systems in a defined manner has been examined. Therefore, the occupation in frequency, time and transmission power has been tested and compared to the performance of a vector signal generator, which features designated hardware.

The used hardware in the USRP X310 is designed for a wider range of applications. In comparison to the designated and more expensive hardware of the vector signal generator, this range comes with generally poorer RF-properties. A few aspects to regard are shown in this master thesis.

Danksagung

Die Arbeit im Themenfeld Software Defined Radio ist spannend und an vielen Stellen herausfordernd. Ich möchte mich an dieser Stelle bei meinen Betreuern Marko Krätzig und Prof. Dr. Sebastian Hantscher für die großartige Betreuung und die zahlreichen, wertvollen Ratschläge bedanken.

Ebenso möchte ich mich bei der Open Source Community bedanken, die durch Projekte wie GnuRadio und Octave maßgeblich zu dieser Arbeit beigetragen haben.

Zuletzt möchte ich mich bei meinen Freunden und meinen Eltern für ihre Geduld und Unterstützung bedanken.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ADC	Analog-Digital Converter
AXI	Advanced eXtensible Interface
CE	Computation Engine
CHDR	Compressed Header
CR	Cognitive Radio
CS	Carrier Sensing
DAC	Digital-Analog Converter
DCF	Distributed Coordination Function
DiffServ	Differentiated Services
DIFS	DCF Interframe Space
DSP	Digital Signal Processing
ED	Energy Detection
EDCA	Enhanced Distributed Channel Access
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
GPSDO	GPS disciplined oscillator
HCCA	HFC Controlled Channel Access
HCF	Hybrid Coordination Function
IF	Intermediate Frequency
IP-Core	Intellectual Property Core
JTRS	Joint Tactical Radio System
LNA	Low Noise Amplifier
LO	Local Oscillator
MAC	Media access control
MSPS	Mega Samples Per Second
PCF	Point Coordination Function
PCIe	Peripheral Component Interconnect Express
PER	Packet Error Rate
PIFS	PCF Interframe Space

PPS	Pulse per second
PHY	Physical layer OSI-Referenzmodell
RBW	Resolution Bandwidth
RF	Radio Frequency
RTS/CTS	Request to send/Clear to send
SCA	Software Communication Architecture
SDR	Software-Defined Radio
SFP	Small Form-factor Pluggable
SIFS	Short Interframe Space
TTL	Time to live
UDP	User Datagram Protocol
UHD	USRP Hardware Driver
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
VBW	Video Bandwidth
VITA	VMEbus international trade association
VMEbus	Versa Module Eurocard bus
VSG	Vektor Signalgenerator

1 Einleitung

Die experimentelle Forschung an Funktechnologien erfordert den vollen Zugriff auf alle Netzwerkschichten. In bereits vorgefertigten Hardwarelösungen ist dieser uneingeschränkte Zugriff oft nicht oder nur im Rahmen teurer Softwareupdates möglich, welche ihrerseits wiederum in den Verwendungsmöglichkeiten begrenzt sind. Eine Alternative dazu bieten Software Defined Radio (SDR) Plattformen, welche mit universell einsetzbarer Hardware ein breites Spektrum an Anwendungen abdecken und gezielte Messungen möglich machen.

In dieser Arbeit wird eine SDR Plattform dazu verwendet, die Mediumsbelegung von WLAN Teilnehmern zu emulieren und dessen Einfluss auf das umgebene Funkmedium zu untersuchen. Die Konfiguration einiger wichtiger Parameter erfordert insbesondere Zugriff auf das Verhalten in der PHY- und MAC-Schicht des IEEE 802.11 Protokollstapel.

2 Aufgabenstellung

Das Thema der Masterarbeit ist die Erstellung von Interferenzprofilen mithilfe einer Software Defined Radio Plattform.

Bei den Interferenzprofilen handelt es sind um die Nachbildung der aktiven Umgebungseinflüsse, welche durch gleichzeitige und im gleichen Raum agierende Funkgeräte, gleicher oder anderer Funktechnologien und der Frequenzbelegung charakterisiert sind. Der Grad der Beeinflussung ist abhängig von der Mediumsnutzung und dem Mediumszugriffsmechanismus. Die Mediumsnutzung beschreibt die tatsächliche Belegung des Mediums in den Dimensionen Frequenz, Zeit und Leistung. Sie wird im Wesentlichen durch die Technologie des Funkkommunikationssystems und die Kommunikationsanforderung der Anwendung beeinflusst. Die Berücksichtigung aktiver Umgebungseinflüsse bei der Performancebewertung von industriellen Funksystemen soll den Grad des Einflusses auf das Zeit- und Fehlerverhalten der zu testenden Funksysteme aufzeigen.

Im Rahmen dieser Arbeit soll mit Hilfe einer Software Defined Radio (SDR) Plattform ein Konzept für die Erstellung von Interferenzprofilen erarbeitet werden. Damit sollen zum einen Interferenzprofile erstellt werden, welche die Mediumsbelegung von industriellen Funkanwendungen gezielt nachbilden. Zudem soll mithilfe der SDR Plattform der Mediumszugriffsmechanismus der verwendeten Funktechnologie berücksichtigt werden.

3 Grundlagen

3.1 Einleitung

Bereits in den späten 1970er Jahren wurden im militärischen Bereich Software Radio Systeme beschrieben, die aus einem Analog-Digital-Wandler (ADC) und einem Mikroprozessor bestanden. Wenig später, in den 1980er Jahren, kam die Idee auf, einen rekonfigurierbaren Empfänger zu entwickeln. Entsprechende Konzepte beinhalteten die automatische Erkennung der Modulation eines empfangenen Signals und Analyse des Bitstroms. Um die Rekonfigurierbarkeit zu erreichen, wird die Signalverarbeitung ab dem ADC durch flexible Software realisiert [2].

1999 haben Dr. Joseph Mitola und Prof. Gerald Q. Maguire in ihrem Artikel die Verwendung des SDR als Cognitive Radio (CR) vorgeschlagen [3], welches ein definiertes Frequenzspektrum überwacht, durch intelligente Algorithmen ungenutzte Frequenzbereiche erkennt und für die eigene Übertragung nutzt.

Eines der ersten öffentlichen SDR Projekte des US-Militärs war das SpeakEASY Projekt, welches mehr als 10 militärische Funkstandards in einem Frequenzbereich von 2 MHz bis 2 GHz abbildete. Das Projekt wurde in zwei Phasen aufgeteilt, aus der ersten Phase ging ein für den Laborbetrieb entwickelter Demonstrator hervor, welcher mit Bodenfunkgeräten, Satelliten, Luftwaffen- und Marinefunk kommunizieren konnte. In der zweiten Phase wurde dieses System in einem portablen Demonstrator realisiert [30].

Die Ergebnisse der zweiten SpeakEASY Phase gingen mit dem Projektende 1995 in die Entwicklung des Joint Tactical Radio System (JTRS) des US-Militärs über. Vier der größten Hersteller militärischer Funksysteme Raytheon, BAE SYSTEMS, Rockwell-Collins und ITT definierten im Auftrag des US Department of Defense den Software Communication Architecture (SCA) Standard für die Entwicklung von SDRs.

Da das Militär und der kommerzielle Markt bis dahin von den Aktivitäten des jeweils anderen nichts wussten, wurde 1996 das „Modular Multifunction Information Transfer System“ (MMITS) Forum gegründet, welches später in SDR-Forum und aktuell in Wireless Innovation Forum (WInnF) umbenannt wurde. Das Forum bringt weltweit mehr als 100 erfahrene Experten aus den Themengebieten SDR und CR zusammen und treibt die Entwicklung von Standards, Technologien und Produkten voran [13].

3.2 Software Defined Radio

3.2.1 Grundprinzip

Das SDR wird im Wesentlichen zwischen einer Antenne und einer Signalquelle bzw. –senke betrieben. Die Schnittstelle zur Antenne ist durch ein analoges Signal gegeben,

welches im Sendefall in das umgebene Funkübertragungsmedium abgestrahlt bzw. im Empfangsfall aus dem Funkübertragungsmedium über die Antenne am Eingang des SDR empfangen wird. Hostseitig sendet und empfängt die SDR-Plattform sogenannte Samples über Schnittstellen wie z.B. Ethernet oder PCIe, deren Bearbeitung vom Host übernommen wird.

Die Verarbeitung zwischen einem analogen Signal aus dem Funkübertragungsmedium und Samples, die vom Host bearbeitet werden, lässt sich in vier Bereiche unterteilen (siehe Abbildung 1).

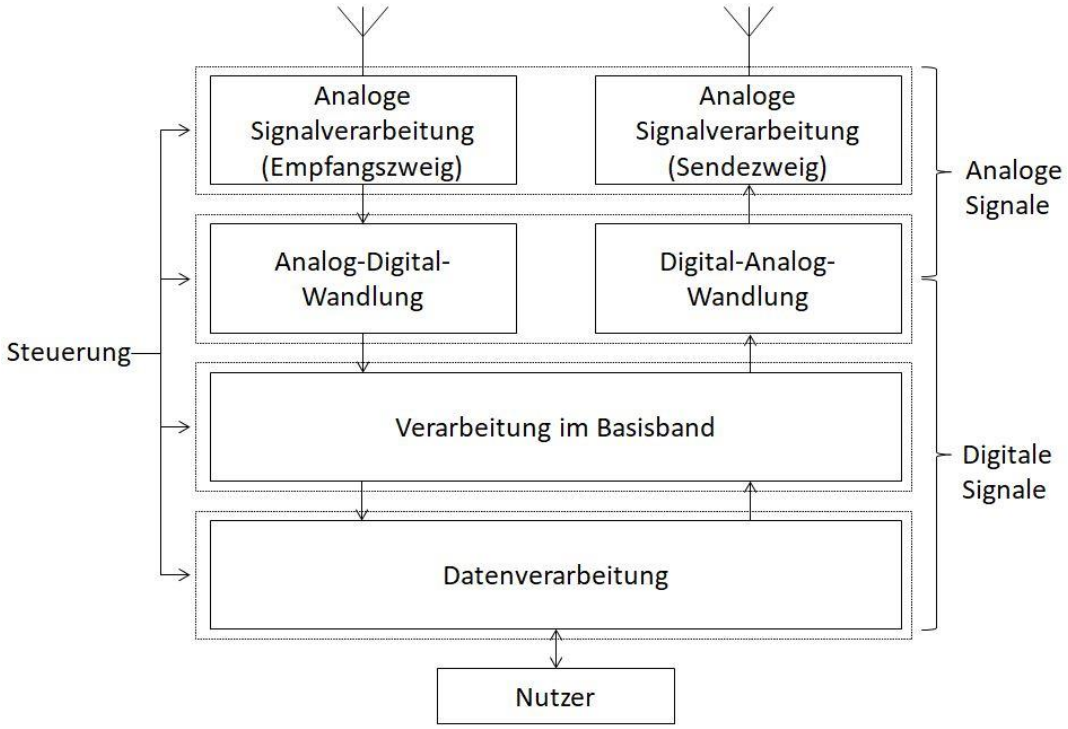

Abbildung 1: SDR Signalverarbeitungskette [2]

Die Hardwarekomponenten im analogen Bereich stellen die Schnittstelle zur Antenne dar. Deren Eigenschaften werden zwar durch die verwendete Software anwendungsabhängig gesteuert, sind aber ansonsten unabhängig von der jeweiligen Anwendung fest im SDR integriert.

Die Hardwarekomponenten im digitalen Bereich übernehmen die Signalverarbeitung der durch den Nutzer erzeugten Daten beziehungsweise der aus dem Funkübertragungsmedium erfassten und durch den Digital-Analog-Wandler digitalisierten Signale. Dazu wird ein Signalprozessor beziehungsweise ein Field Programmable Gate Array (FPGA) verwendet. Neben Komponenten, die die Rechenleistung beeinflussen, sind gegebenenfalls weitere Module als Schnittstellen zum Host vorhanden. Abhängig von der im

analogen Bereich angewandten Signalverarbeitung ist im digitalen Bereich aufgrund höherer zu verarbeitender Frequenzen gegebenenfalls mehr Aufwand nötig.

3.2.2 Hardwarearchitektur

Die Hardwarearchitektur eines idealen SDR-Transceivers verarbeitet das HF-Signal direkt. Der SDR-Empfänger digitalisiert also das empfangen HF-Signal nach der Antenne mit einem ADC und verarbeitet das digitalisierte HF-Signal in einem Signalprozessor (FPGA, GPU, CPU usw.). Der SDR-Sender wandelt das digitale HF-Signal mittels Digital-Analog-Wandler direkt in ein analoges HF Signal um. Allerdings scheitert der ideale Aufbau eines idealen SDR nicht nur an den technischen Schwierigkeiten der notwendigen Analog-Digital- und Digital-Analog-Wandler (DAC), sondern auch daran, dass hierbei ein breitbandiges Signalgemisch digitalisiert würde, von dem nur ein kleiner Teil vom Nutzsignal verwendet wird. Das wäre weder technische noch kommerziell sinnvoll [2]. Im Folgenden werden die möglichen Hardwarearchitekturen von SDR-Transceivern beschrieben.

Direktabtastung

Das dem idealen SDR am nächsten kommende Konzept ist die direkte Abtastung des empfangenen Signals. Abgesehen von einem Low Noise Amplifier, welcher die Signalleistung für die Abtastung durch den ADC erhöht, und einem Bandpassfilter findet keine weitere analoge Signalverarbeitung statt [5]. Das Konzept ist in seiner Anwendbarkeit insbesondere durch die Abtastrate des ADC und durch die verfügbare Rechenleistung des Signalprozessors bzw. FPGAs begrenzt. Mit derzeitigen ADCs können Frequenzen bis 2 GHz erfasst werden.

Superheterodynempfänger

Beim Überlagerungsempfänger (Superheterodynempfänger) wird das empfangene Signal vor der Abtastung mit einer durch einen Lokaloszillator (LO) erzeugten Trägerfrequenz gemischt, sodass die Zwischen-frequenz (IF) bei typischerweise wenigen MHz liegt. Bei der entsprechenden Implementierung im SDR ist die erforderliche Abtastrate des ADC damit deutlich geringer, allerdings können Probleme wie zum Beispiel Spiegelfrequenzen auftreten.

Direktmischempfänger (zero-IF)

Die Direktmischempfänger (Homodynempfänger) sind bei SDR die am weitesten verbreitete Form der analogen Signalverarbeitung. Das empfangene Signal wird nach der Verstärkung auf zwei Leitungen aufgeteilt und jeweils mit dem Cosinus (Inphase) bzw. dem Sinus (Quadratur) der Trägerfrequenz gemischt und tiefpassgefiltert (siehe

Abbildung 2). Dadurch entfallen Probleme bei der Mischung auf Zwischenfrequenzen, allerdings wird der DC-Offset durch die Mischung auf 0 Hz relevant [6].

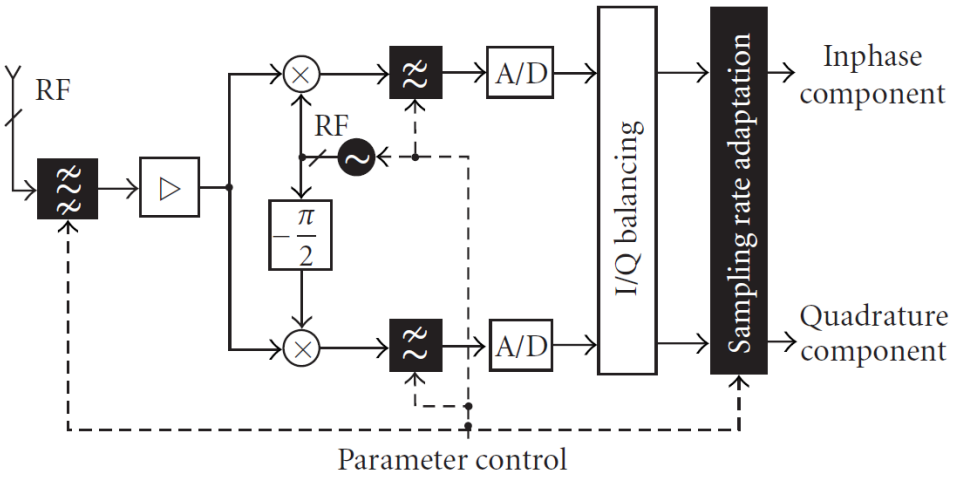

Abbildung 2: Blockschaltbild Direct Conversion Frontend [2]

3.2.3 Vorteile und Nachteile von Software Defined Radio

3.2.3.1 Vorteile

SDR Plattformen haben durch die anteilige Realisierung der Signalverarbeitung in Software ein erheblich breiteres Spektrum an möglichen Anwendungen als dedizierte Hardware. Die Veränderung des Funkverhaltens durch Modifikation der Software ist im Gegensatz zur teureren Hardwaremodifikation insbesondere im Forschungsumfeld und bei privatem Gebrauch durch z.B. Funkamateure von Vorteil.

Die für die digitale Signalverarbeitung auf SDR Plattformen vorgesehene Software nimmt insbesondere in der Entwicklung viel Zeit in Anspruch, ist aber in ihrer Anwendbarkeit nur durch den grundsätzlichen Aufbau der verarbeitenden Komponenten (FPGA, GPP, DSP) eingeschränkt und kann so auf eine Vielzahl an Geräten angewendet werden. Neben Python und Matlab sind insbesondere in der Programmiersprache C/C++ bereits umfangreiche Bibliotheken zur digitalen Signalverarbeitung vorhanden. Durch die Verwendung solcher Bibliotheken wie zum Beispiel Aquila [10] kann die Entwicklungszeit von Signalverarbeitungssoftware wiederum verkürzt werden.

Durch die Flexibilität der digitalen Verarbeitungskomponenten der SDR Plattformen ist es möglich, mehrere Anwendungen in der Entwicklung der Signalverarbeitungskette zu berücksichtigen und zwischen diesen Anwendungen während der Laufzeit zu wechseln. So können sich SDRs an diverse Funkprotokolle anpassen.

3.2.3.2 Nachteile

Durch die nötige Rechenleistung entsteht insbesondere bei den zur Signalverarbeitung eingesetzten Komponenten ein höherer Energieverbrauch als bei konventionellen Funkgeräten mit dedizierter Hardware. Ist die Energieaufnahme von Bedeutung, zum Beispiel wenn es sich um ein mobiles System handeln soll, so entsteht unter Umständen ein Balanceakt zwischen der Batterielaufzeit des Systems und der verfügbaren Rechenleistung.

3.3 Verwendete Hardwareplattform

3.3.1 Universal Software Radio Peripheral

Bei der in dieser Arbeit verwendeten SDR Plattform handelt es sich um das *Universal Software Radio Peripheral* (USRP) X310 mit zwei UBX-160 Erweiterungskarten von der Firma Ettus Research, siehe Abbildung 3 und Abbildung 4. Die Analog-Digital- beziehungsweise Digital-Analog-Wandler befinden sich zusammen mit einem FPGA auf dem USRP X310 und stellen die Schnittstelle zum Host und den Erweiterungskarten dar, welche die analoge Signalverarbeitung realisieren und als Module austauschbar sind.

Abbildung 3: USRP X310 mit zwei Erweiterungskarten UBX-160

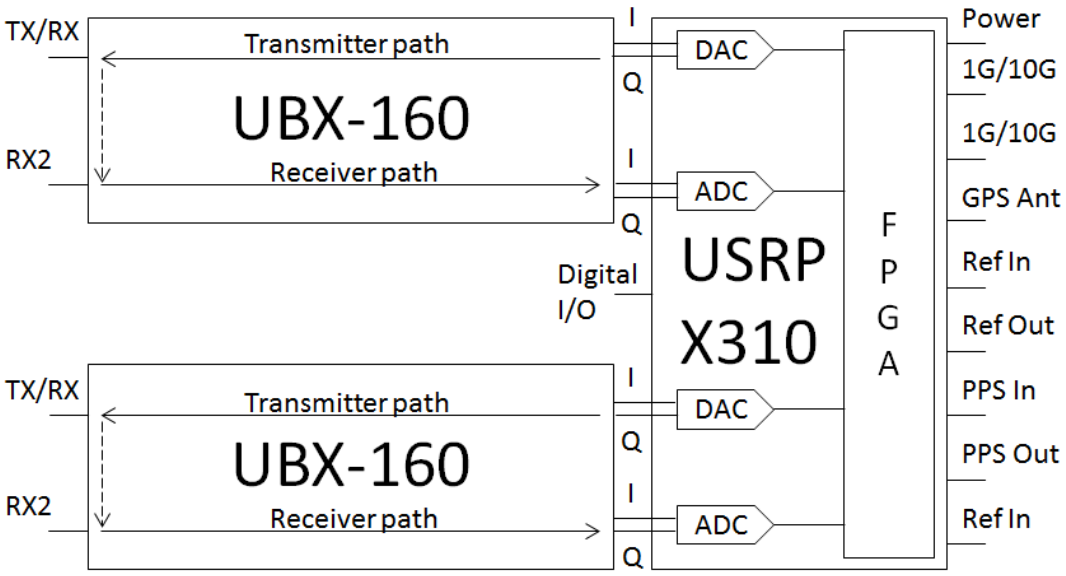

Abbildung 4: Skizze Aufbau USRP

Die USRP X310 Plattform hat insgesamt zwei Slots für Erweiterungskarten. Zur Abtastung der IQ-Daten sind pro Steckplatz jeweils ein Analog-Digital-Wandler mit 200 MS/s bei einer Auflösung von 14 Bit und ein Digital-Analog-Wandler mit 800 MS/s bei einer Auflösung von 16 Bit auf dem Mainboard vorhanden. Die A/D- und D/A-Wandler können über einen GPS Disciplined Oscillator (GPSDO) getaktet werden. Kernstück des USRP X310 ist der Kintex-7 FPGA mit 1 GB DDR3 RAM und EEPROM Speicher. Die vom FPGA bearbeiteten Samples können über die 1 GBit/s Ethernet Schnittstelle, die 10 GBit/s Ethernet Schnittstelle oder die PCIe x4 Schnittstelle an den Host übertragen werden [4].

Die im USRP X310 verwendeten Taktraten werden standardmäßig automatisch eingestellt, dies geschieht in Abhängigkeit von der gewählten Abtastrate. Wird beim Ansprechen des USRP durch den USRP Hardware Treiber (UHD) die Taktrate vorgegeben, so wird die automatische Einstellung mit dem angegebenen Wert überschrieben. UHD sieht dazu sowohl die Funktion set_master_clock_rate(double rate, size_t mboard) als auch ein Argument der Funktion uhd_usrp_probe (siehe Tabelle 1) vor. Da es nur einige mögliche Taktraten bei gegebener Abtastrate gibt, wird bei unpassender Taktrate die nächstmögliche genommen. Die aktuell verwendete Taktrate kann über get_master_clock_rate() abgefragt werden.

Die Konfigurationsparameter des USRP X310 sind auf dem internen EEPROM gespeichert. Diese können bei einer aktiven Verbindung durch mit UHD mitgelieferte Funktionen wie zum Beispiel rx_samples_to_file bearbeitet werden. In Tabelle 1 sind die mit diesen Funktionen verwendbaren Argumente aufgeführt.

Tabelle 1: Argumente rx_samples_to_file [11]

Argument	Beschreibung	Geräte	Beispiel
blank_eeprom	Löscht den EEPROM, wird nicht empfohlen	X3x0	blank_eeprom=1
fpga	Lädt alternatives Bitfile auf den FPGA	Alle USB Geräte, X3x0 (über PCIe), alle eingebetteten Geräte	fpga=/path/to/bitfile.bit
fw	Lädt alternative Firewall	Alle USB Geräte, X3x0	fw=/path/to/fw.bin
ignore-cal-file	Ignoriert vorhandene Kalibrierungsdaten	Alle Geräte mit cal-file Support	
master_clock_rate	Master Clock Rate in Hz	X3x0, B2x0, B1x0, E3x0, E1x0	master_clock_rate=16e6
dboard_clock_rate	Daughterboard clock rate in Hz	X3x0	dboard_clock_rate=50e6
mcr	Überschreibt master-clockcrate Einstellungen	USRP1	mcr=52e6
niusrprpc_port	RPC Port für NI USRP RIO	X3x0	niusrprpc_port=5445
system_ref_rate	Reference Clock Rate in Hz	X3x0	system_ref_rate=10e6
self_cal_adc_delay	Startet Übertragungszeit-Selbstkalibrierung des ADC.	X3x0	self_cal_adc_delay=1
ext_adc_self_test	Startet einen ausführlichen ADC Selbsttest	X3x0	ext_adc_self_test=1

Argument	Beschreibung	Geräte	Beispiel
recover_mb_eeprom	Schaltet Versionsüberprüfung aus, ist nur empfehlenswert, um Geräte mit fehlerhaftem EEPROM zurückzusetzen.	X3x0, N230	recover_mb_eeprom=1
skip_dram	Ignoriert DRAM FIFO Block. Verbindet TX Streamer direct mit DUC oder radio.	X3x0	skip_dram=1
skip_ddc	Ignoriert DDC Block. Verbindet Rx Streamer direkt mit radio.	X3x0	skip_ddc=1
skip_duc	Ignoriert DUC Block. Verbindet Rx Streamer oder DRAM direkt mit radio.	X3x0	skip_duc=1

3.3.2 Erweiterungskarten

Als Erweiterungskarten des USRP X310 kommen die UBX-160 Daughterboards zum Einsatz. Die analoge Signalverarbeitung wird vom Antennenanschluss bis zur Erzeugung des IQ-Signals im Direktmischverfahren durch die UBX-160 Daughterboards übernommen. Frequenzen, die über 500 MHz liegen, werden über den Direct Conversion Path geführt. Signale niedrigerer Frequenzen werden durch einen separaten LO auf bestimmte Zwischenfrequenzen gemischt (siehe Abbildung 5). Die verwendete Erweiterungskarte kann in einem Frequenzbereich von 10 MHz bis 6 GHz eingesetzt werden und hat eine maximale Bandbreite von 160 MHz.

Die Einstellung der Daughterboards erfolgt über den UHD. Somit kann der Offset des Lokaloszillators dahingehend verändert werden, dass das Passband auf eine tiefe Zwischenfrequenz statt auf 0 Hz gemischt wird.

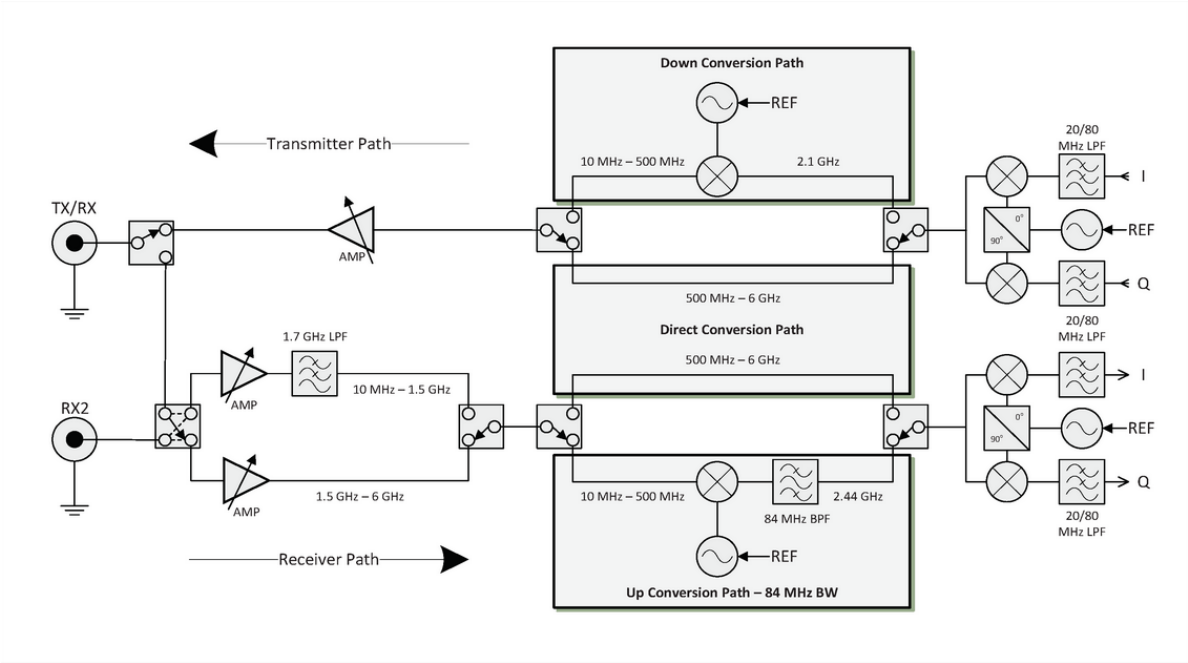

Abbildung 5: UBX160 Daughterboard [1]

3.3.3 Kommunikationsschnittstellen

Abhängig von den Anforderungen an die Nutzung der USRP Plattform können verschiedene Schnittstellen zum Host oder anderen externen Geräten genutzt werden. Zur Datenübertragung an den Host-PC stehen die in Abbildung 6 dargestellten Schnittstellen Ethernet und PCIe zur Verfügung.

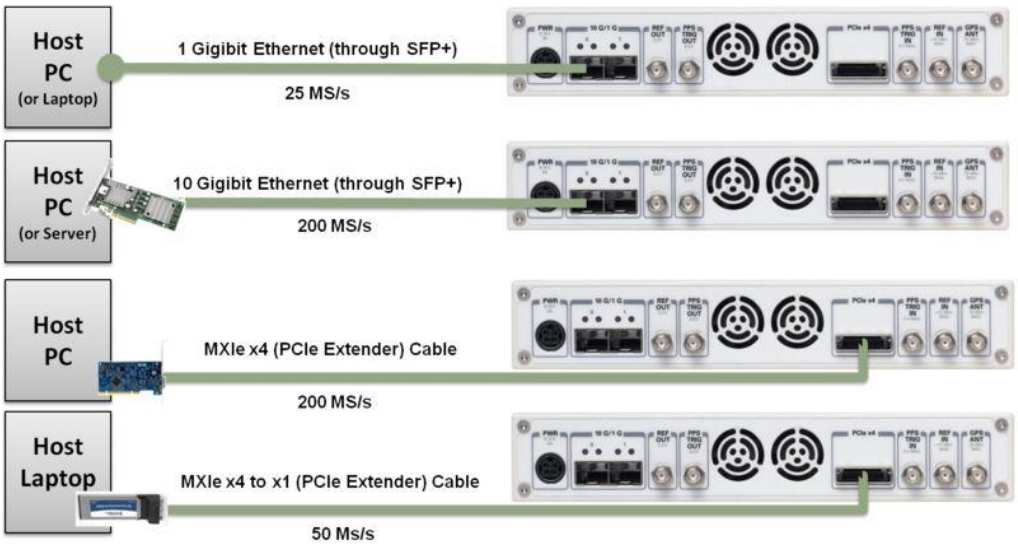

Abbildung 6: Schnittstellen USRP X310 [9]

Falls die USRP Plattform erstmalig konfiguriert oder durch die Anwendung nicht die volle Bandbreite und Geschwindigkeit genutzt werden muss, kann die Ethernetschnittstelle mit einer Datenübertragungsrate von 1 GBit/s genutzt werden. Der Host PC benötigt dazu

eine Netzwerkkarte, während für den SFP+ Slot Port 0 am USRP ein SFP-Adapter notwendig ist. Als Kommunikationsprotokoll über die Ethernetschnittstelle wird UDP verwendet. Die standardmäßig auf dem EEPROM unter ip-addr0 gespeicherte IP-Adresse ist die 192.168.10.2/24. Da sich mit 25 MS/s nur ein Bruchteil der ansonsten verfügbaren Leistung der USRP Plattform nutzen lässt, ist entweder die Verwendung einer leitungsfähigeren Verbindung mit dem Host PC sinnvoller oder eine kostengünstigere Alternative zum USRP X310 denkbar.

Neben der Verwendung der 1 GBit/s Ethernetschnittstelle sind die SFP+ Ports für die Nutzung als Ethernetschnittstelle mit einer Datenübertragungsrate von 10 GBit/s vorgesehen. Hostseitig wird dazu eine Netzwerkkarte für 10 GBit/s benötigt, welche über zwei SFP+ Kabel mit der USRP Plattform verbunden wird. Die Verbindung über die 10 GBit/s Ethernetschnittstelle ermöglicht die Übertragung von 200 MS/s im Vollduplex-betrieb, die IPs befinden sich im EEPROM unter ip-addr2 für Port 0 und ip-addr3 für Port 1. Standardmäßig sind die Adressen 192.168.30.2 und 192.168.40.2 für die beiden Ports vergeben.

3.3.4 Kommunikationsprotokoll

Die Software Communication Architecture (SCA) ist als offener Standard aus dem JTRS Projekt des US-Militärs entstanden und dient als ein an Interoperabilität, Portabilität und Austauschbarkeit orientiertes Framework und eine Richtlinie für Hard- und Software-designer. Als solche hat es sich als de-facto Standard für SDR-Systeme entwickelt. SCA definiert eine Software-Infrastruktur für die Steuerung, Konfiguration und das Management von Kommunikationssystemen und spezifiziert Schnittstellen, Abläufe, Waveform APIs und Security-Anforderungen.

Das Framework gibt vor, wie Hard- und Softwarekomponenten für die SCA-Konformität zusammenarbeiten müssen. [13]

Auf der SCA basierend hat die VMEbus International Trade Association (VITA) für SDRs den VITA Radio Transport (VRT) Standard entwickelt. Hauptmerkmal des Standards ist die Verknüpfung der durch den ADC erzeugten Samples mit Kontextdaten von den jeweiligen analogen Komponenten des SDR. Die Kontextdaten und Samples werden zusammen verpackt an die VRT-signalverarbeitende Hardware geschickt, welche die Kontextdaten und die zugehörigen Samples entsprechend des Kontextes verarbeitet. Beispiele für Kontextdaten sind in [14] aufgelistet, nennenswert sind an dieser Stelle die Streamidentifikation und Zeitstempel, die zur Zuordnung weiterer Kontextdaten wie Abtastrate, Bandbreite, Verstärkung und Antenneneigenschaften zum jeweiligen Sample nötig sind.

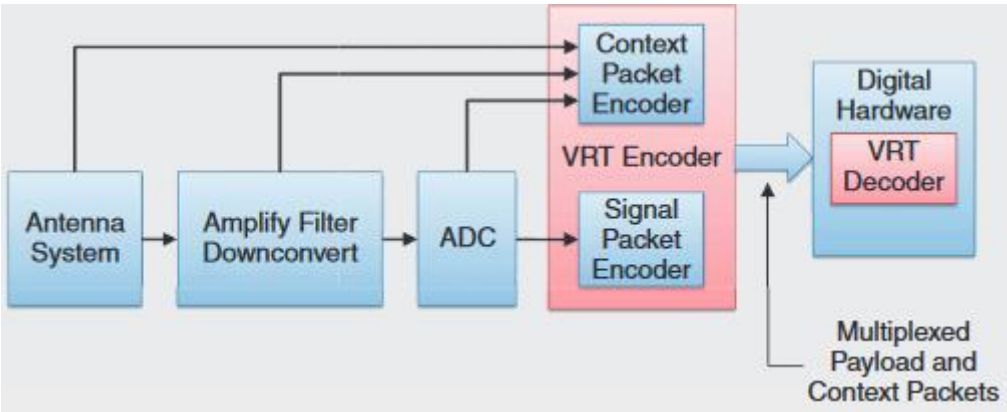

Abbildung 7: VITA Radio Transport [14]

In der dritten Generation der von Ettus vertriebenen USRPs wird das eigens dafür entwickeltes Übertragungsprotokoll CHDR (Compressed Header) verwendet, welches zwar auf VRT aufbaut, aber deutlich weniger Kontextdaten mitführt. Der Header hat, abgesehen von einem gegebenenfalls vorhandenen „fractional Timestamp“, grundsätzlich eine Gesamtlänge von 64 bit. Der Aufbau des CHDR ist in Tabelle 2 gegeben [11].

Tabelle 2: Aufbau CHDR [11]

Bits	**Bedeutung**
63:62	Packet Type
61	Fractional Timestamp Flag
60	End-of-Burst bzw. Error Flag
59:48	12-bit sequence number
47:32	Total packet length in Bytes
31:0	StreamID (SID)

Die 32 bit lange Streamidentifikation (SID) des Headers setzt sich aus den 8-bit langen Sender- und Empfängeradressen des Pakets und den jeweiligen, ebenfalls 8-bit langen Endpunkten zusammen. Der Pakettyp wird sowohl durch die ersten beiden Bits 63 und 62 als auch durch das End-of-Burst (EOB) Bit 60 definiert. In Tabelle 3 sind die einzelnen Pakettypen aufgelistet.

Tabelle 3: Kodierung des Pakettyps CHDR

Bit 63	Bit 62	Bit 60	Pakettyp
0	0	0	Daten
0	0	1	Daten (End-of-Burst)
0	1	0	Flusskontrolle
1	0	1	Steuerpaket
1	1	0	Steuerpaket – Nachfrage
1	1	1	Steuerpaket – Antwort

Über den von Ettus bereitgestellten CHDR-Dissector für Wireshark lassen sich die über Ethernet bzw. USB gesendeten UDP/CHDR Pakete anzeigen und analysieren. Beispielhaft sind die Mitschnitte einer Übertragung in Abbildung 8 für ein CHDR Paket vom Typ Data und in Abbildung 9 für ein CHDR Paket vom Type Command dargestellt.

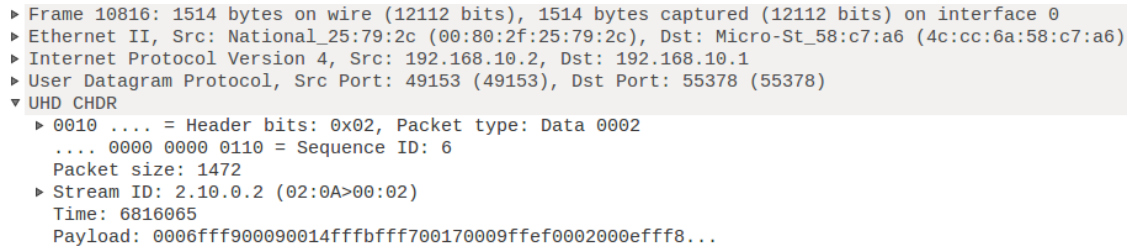

Abbildung 8: CHDR Paket vom Typ Data

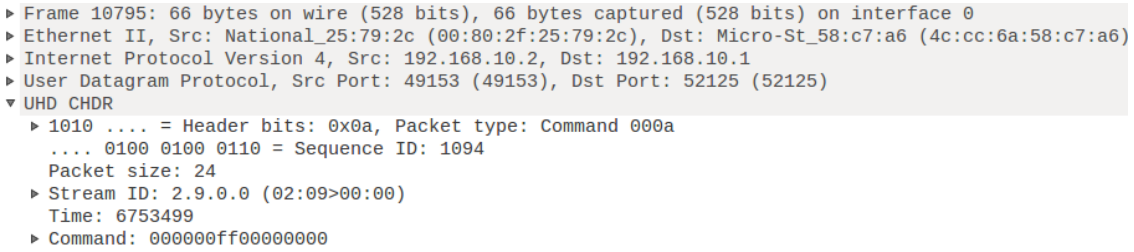

Abbildung 9: CHDR Paket vom Typ Command

Das Ethernetpaket wird als UDP Frame übertragen, der Frame unterteilt sich hierbei in die Header mehrerer Protokolle (Ethernet, IPv4, UDP, UHD CHDR) und Payload.

Der Ethernet-Header beinhaltet die MAC-Adressen der Paketquelle und des Paketziels und den Ethernet Typen des verwendeten Protokolls der nächsthöheren Schicht, in diesem Fall IPv4. Der Header ist insgesamt 14 Bytes lang, wovon jeweils 6 Bytes für jede der beiden MAC-Adressen und 2 Bytes auf das verwendete Ethernet Typenfeld fallen.

Der IPv4 Protokollheader ist 20 Bytes lang und enthält Informationen wie zum Beispiel die Quellen- und Ziel-IP-Adresse, Länge des Pakets, Time to Live (TTL - Lebenszeitangabe) und der Type des verwendeten Protokolls der nächsthöheren Schicht, in diesem Fall UDP.

Der UDP Header ist 8 Bytes lang und enthält die Portnummern der Quelle und des Ziels, die Länge und die Prüfsumme des Pakets, jeweils in 2 Bytes kodiert.

Der von Ettus bereitgestellte CHDR ist, je nach vorhandenem Zeitstempel, entweder 8 oder 16 Bytes lang und enthält weitere, inhaltsbezogene Kennzahlen wie die Sequenz-nummer, Nutzdatengröße, SID und gegebenenfalls einem auf den Takt basierenden Zeitstempel, wie er in Abbildung 8 und Abbildung 9 zu sehen ist.

Sind in dem Paket Nutzdaten enthalten, so werden insgesamt 364 32-Bit große Samples mit einer Gesamtlänge von 1456 Bytes übertragen. Je nach eingestellter Samplerate ist der Abstand zwischen zwei Paketen beziehungsweise die Anzahl der in einer Sekunde übertragenen Pakete variabel. Die mithilfe des bereitgestellten CHDR-Dissectors und Wireshark erfassten Übertragungen konnten dazu bei verschiedenen Sampleraten ausgewertet werden.

Die Peripheral Component Intercom Express (PCIe) Schnittstelle wurde überwiegend für die computerinterne Kommunikation von Erweiterungskarten konzipiert. Wesentlicher Unterschied zum Vorgänger PCI ist die Verwendung von Switches zur Verbindung der einzelnen Komponenten anstelle eines einzelnen Bussystems. Die Übertragung erfolgt über eine oder mehrere Lanes, die wiederum aus zwei differenziellen Leiterpaaren zur simultanen Übertragung in beide Richtungen bestehen [25].

Die Pakete durchlaufen, ähnlich wie bei Ethernet, mehrere Protokollschichten. Der durch die Transaction Layer erstellte Header enthält neben Inhalt den Pakettyp in den Feldern Fmt und Type, Adressen in den Feldern Requester ID und Address, cyclic redundancy check (CRC) und die Länge des Paketes im Feld Length [27]. Das in Abbildung 10 dargestellte Paket entspricht der verwendeten Version 2 des PCIe Protokolls.

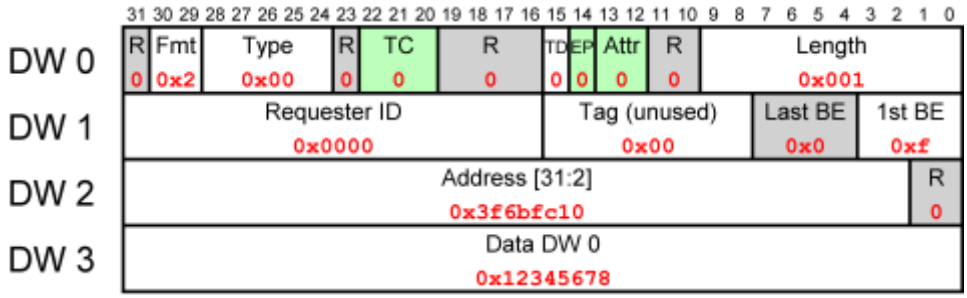

Abbildung 10: TLP write [26]

3.3.5 Hardwareschnittstellen und Konfiguration

Neben den SMA-Anschlüssen TX/RX- und RX2 für die Antennen sind weitere SMA-Anschlüsse für eine GPS Antenne, Ref In, Ref Out, PPS In und PPS Out vorhanden. Diese Anschlüsse sind insbesondere für die Verwendung eines externen Referenztaktes vorgesehen.

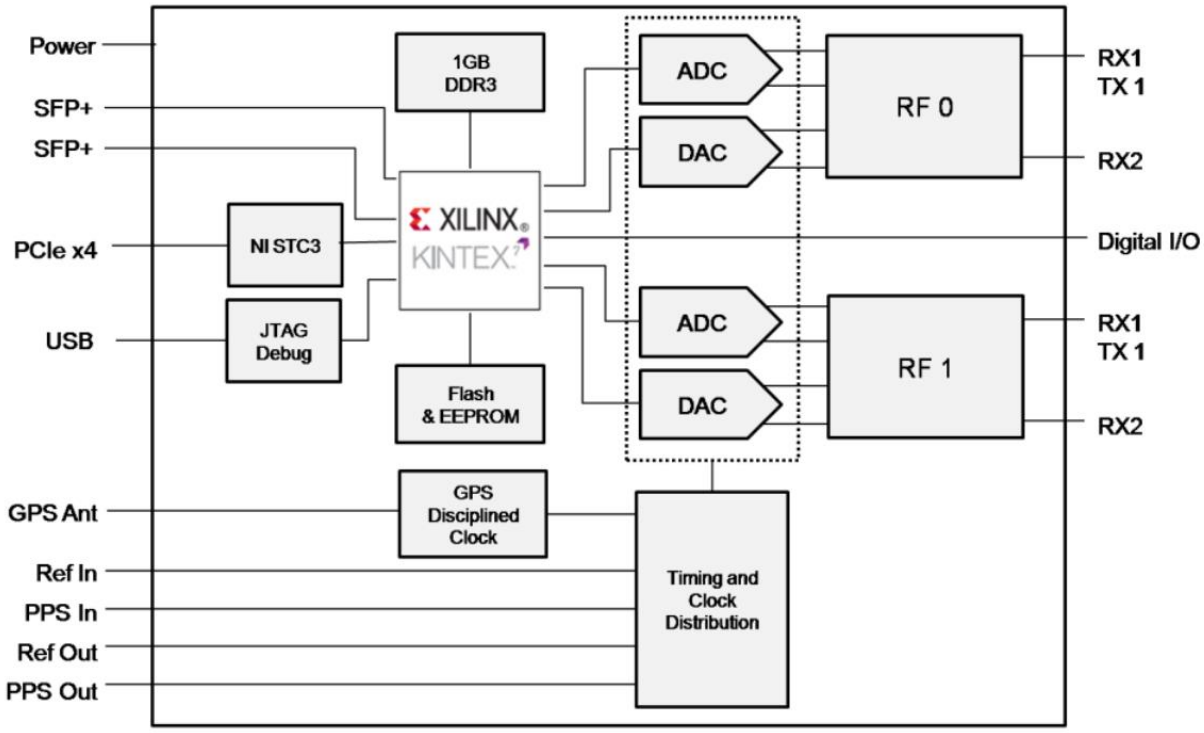

Abbildung 11: USRP X310

Tabelle 4: Schnittstellen USRP X310

Nr.	Schnittstelle	Anschlusstyp	Beschreibung
1	Power	KPJX	Gleichspannung 12 V
2	MXIe x4	MXIe	Schnittstelle für MXIe Verbindung
3	JTAG	USB 2.0 Typ B	Schnittstelle für USB-JTAG Programmierung
4	GPS Ant	SMA	Schnittstelle für GPS Antenne
5	Ref In	SMA	Referenztakt Eingang
6	PPS In	SMA	Eingang für PPS Signal
7	Ref Out	SMA	Referenztakt Ausgang
8	PPS Out	SMA	Ausgang für PPS signal
9	1G/10G ETH 0	SFP+	SFP+ Schnittstelle für Ethernet
10	1G/10G ETH 1	SFP+	SFP+ Schnittstelle für Ethernet

Nr.	Schnittstelle	Anschlusstyp	Beschreibung
11	RF A Tx/Rx	SMA	Tx/Rx Anschluss der Erweiterungskarte 1
12	RF A Rx	SMA	Rx Anschluss der Erweiterungskarte 1
13	RF B Tx/Rx	SMA	Tx/Rx Anschluss der Erweiterungskarte 2
14	RF B Rx	SMA	Rx Anschluss der Erweiterungskarte 2

3.4 Software

Neben der Konfiguration der verwendeten Hardware ist es notwendig, die weitere Signalverarbeitung in Software durchzuführen. In der einfachsten Form stehen hierzu die Inphase- und Quadratur-Werte aus dem USRP zur Verfügung, die von einer Auswahl an Programmen weiterverarbeitet werden können, um die Daten darzustellen, zu speichern oder wieder abzuspielen. Im Folgenden wird dazu das Programm GnuRadio und die in der jeweiligen Verwendung liegenden Möglichkeiten vorgestellt.

Bei GnuRadio handelt es sich um eine open-source Entwicklungsumgebung für die Implementierung verschiedener Verfahren aus der digitalen Signalverarbeitung. So können unterschiedliche Signale erzeugt, durch Modulations- bzw. Demodulations-verfahren und Filter verändert und als Zeitsignale, Frequenzspektren etc. dargestellt werden. Neben Anwendungen, die als Simulationen ausschließlich auf dem PC ausgeführt werden, ist die Verwendung zusätzlicher Hardware durch Schnittstellen in GnuRadio möglich. Entsprechende Peripheriegeräte können zum Beispiel Funkgeräte sein, die gesendete Inphase- und Quadraturdaten in das Funkübertragungsmedium abstrahlen. Anwendungsgebiete sind aufgrund günstig verfügbarer SDR-Hardware im Amateurfunk und in Forschung und Lehre verbreitet [7].

Eine intuitive Nutzung von GnuRadio ist durch den mitgelieferten GnuRadio Companion möglich, welcher eine Verkettung von Signalverarbeitungskomponenten als Blöcke in einem Flowgraph ähnlich wie in Simulink erlaubt. Hinter den Blöcken stehen Python bzw. C++ Funktionen, die beim Ausführen des Flowgraph generiert und ausgeführt werden. Eine weitreichende Bibliothek solcher Blöcke ist standardmäßig vorhanden, weitere Blöcke können entweder von Repositorien veröffentlichter Projekte kopiert und gegebenenfalls angepasst oder selbst in Python oder C++ erstellt werden. Für performancerelevante Blöcke bietet sich dabei eher C++ an, im Gegensatz dazu ist Python übersichtlicher [8].

Weiterhin ist der von Ettus Research veröffentlichte USRP Hardware Driver (UHD) in GnuRadio enthalten, welcher die Kommunikation mit USRP-Geräten über verschiedene Schnittstellen wie USB, Ethernet oder PCIe ermöglicht. Die entsprechenden UHD Versionen müssen dabei übereinstimmen und können dazu durch Auswahl der in

GnuRadio verwendeten Version aus dem Ettus Repository ausgewählt oder mit einer angepassten Installation von GnuRadio durch z.B. PyBOMBS festgelegt werden. Entsprechende Source- und Sink-Blöcke sind im GnuRadio Companion enthalten und können unter Angabe der USRP-Adresse und anderer Parameter wie Trägerfrequenz und Abtastrate als Schnittstelle für den Flowgraph verwendet werden.

Um die vom USRP gesendeten Daten speichern zu können, stellt GnuRadio Source- und Sinkblöcke zur Verfügung, die auf Dateien des Host PCs verweisen. Der Inhalt der erstellten Datei hängt von dem in GnuRadio gewählten Datentyp ab, standardmäßig werden sowohl I- als auch Q-Daten in 4 Byte große Werte codiert und nacheinander gespeichert. Weitere Datentypen sind in Tabelle 5 aufgeführt.

Tabelle 5: Datentypen in GnuRadio

Datentyp	**Beschreibung**
Complex64	Single-precision floats, 4 Bytes pro Sample, jeweils I und Q
Float32	Single-precision float, 4 Bytes pro sample
Int32	Signed Integer, 4 Bytes pro sample
Int16	Signed Integer, 2 Bytes pro sample
Int8	Signed Integer, 1 Byte pro sample

Nach der Erstellung und Generierung eines Flowgraphes steht neben der programmeigenen grc-Datei eine von Python lesbare, automatisch erstellte Datei zur Verfügung. Das so erstellte Python Programm kann unabhängig von Gnuradio verändert und ausgeführt werden. So ist es zum Beispiel möglich, die in Gnuradio verfügbaren Komponenten über Gnuradio Companion zu erzeugen und darauf folgend weitere Funktionen entweder in das Framework von Gnuradio einzubinden oder in den von Python lesbaren Code direkt zu integrieren.

Neben der Signalverarbeitung auf dem Host-PC durch einen GPP können Prozesse auch auf dem USRP-internen FPGA realisiert werden. Da die Entwicklung des dazu notwendigen Programmcodes zeitaufwändig ist, wird von Ettus eine Schnittstelle für die Verwendung und Erstellung von entsprechenden Blöcken in GnuRadio bereitgestellt. So können insbesondere zeitkritische Funktionalitäten durch RF Network on Chip (RFNoC) auf dem FPGA realisiert werden [19].

Ein Beispiel für eine daraus resultierende Signalverarbeitungskette ist in Abbildung 12 gegeben, neben der Fourier Transformation können auf diese Weise noch weitere Funktionen wie beispielsweise OFDM Modulation beziehungsweise Demodulation vom FPGA im USRP übernommen werden.

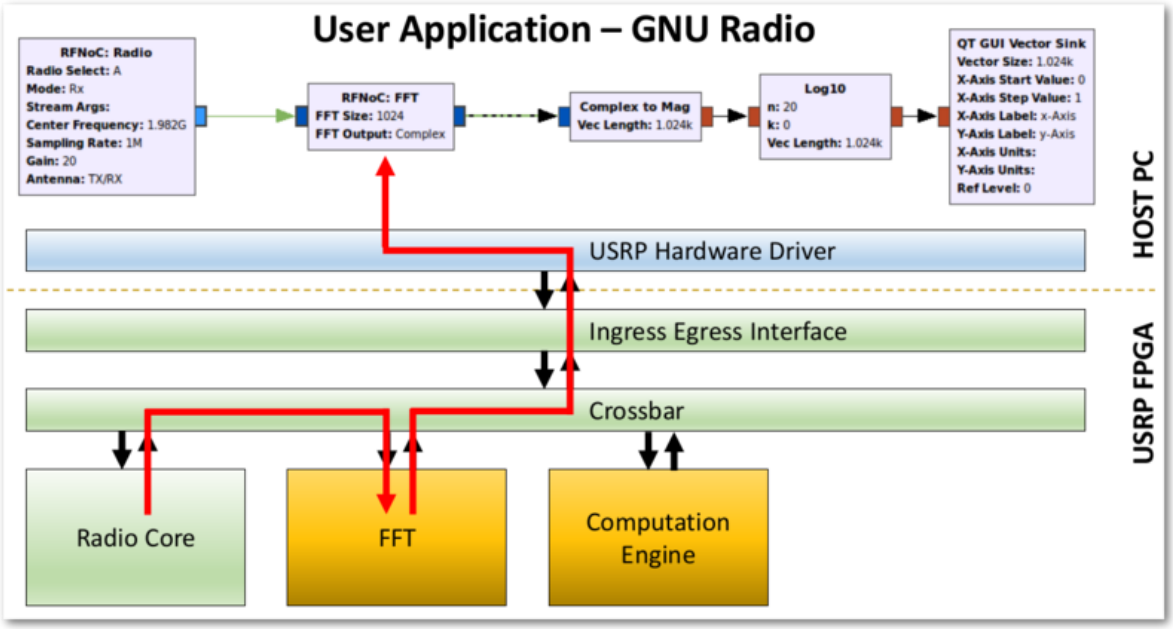

Abbildung 12: RFNoC Beispiel FFT [19]

3.5 Interferenzprofile

Bei den Interferenzprofilen handelt es sich um die Medienbelegung in Frequenz, Zeit und Leistung für typische Funkanwendung in der industriellen Automatisierungstechnik. Als Beispiel ist das Interferenzprofil für das Funksystem WISA in Abbildung 13 dargestellt.

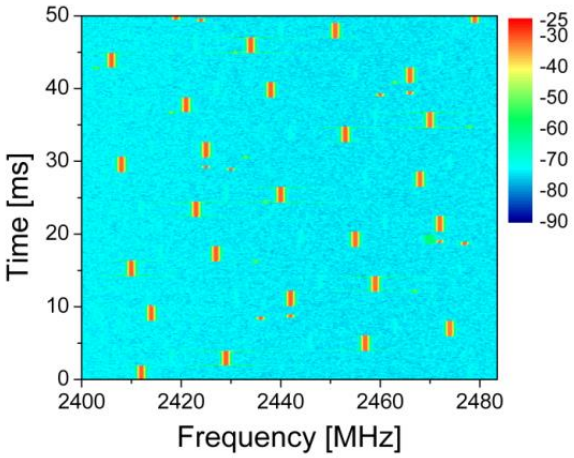

Abbildung 13: Interferenzprofil für das Funksystem WISA

In den Forschungsprojekten KoMe, HiFlecs und ReiCovAir wurden eine Vielzahl an Interferenzprofilen für einen Vektorsignalgenerator erstellt [34][35][36]. Dabei stehen die folgenden Interferenzprofile zur Verfügung:

- Industrielle Ethernetkommunikation mit WLAN im 2,4 GHz ISM Band (IEEE 802.11g) mit einer Bandbreite von 20 MHz, variabler Kanalbelegung (1-6-11; 1-7-13 usw.) und variabler zeitlicher Medienbelegung.

- Das industrielle Wireless Sensor Aktor Netzwerk (WISA) mit variablen Funkteilnehmern.
- Bluetooth ohne adaptive Frequenzsprungverfahren mit variablen Funkteilnehmern.
- Temporäre Störer wie zum Beispiel das Bluetooth Paging oder das Übertragen von WLAN Probe Request Nachrichten
- Funksysteme für Traffic for Transport and Telematics (TTT)
- Funksysteme Intelligent Transport Systems (ITS)

Die oben aufgeführten Interferenzprofile stehen in Form von WV-Dateien für den Vektorsignalgenerator (VSG) SMJ100A von Rohde und Schwarz zur Verfügung. Diese können mithilfe des VSG und der herstellerspezifischen Software (WinIQSim) oder Matlab erstellt werden. Neben Headerinformation wie die Samplerate und die Anzahl an Samples sind die IQ-Werte der Samples als 16-bit lange Integer-Werten in den WV-Files enthalten. Die Optionen zur Erstellung einer WV-Datei und die Inhalte des Headers sind in Tabelle 6 gegeben.

Tabelle 6: Header Inhalte [33]

Bezeichnung	**Beschreibung**
{TYPE: magic, xxxxxxxx}	magic SMU-WV: R&S SMJ waveform. SMU-MWV R&S SMJ multi-segment waveform. SMU-DL R&S SMJ data list. SMU-CL R&S SMJ control list. xxxxxxxx → ASCII-kodierte Checksumme 0 – für Datenlisten und Steuerungslisten Checksumme
{CLOCK: frequency}	Abtastrate der IQ-Daten in Hz.
{COMMENT: string}	ASCII String zum Kommentieren, welcher nicht analysiert wird.
{COPYRIGHT: string}	ASCII String mit Copyright Information, welcher nicht analysiert wird.
{DATE: yyyy-mm-dd;hh:mm:ss}	ASCII String mit Datums- und Uhrzeitangaben.
{EMPTYTAG-Length: #Empty-Sequence}	Der Empty-Tag ist leer, d.h. er enthält keine Daten und dient als Platzhalter.
{LEVEL OFFS: RMSOff-set_dB,PeakOffset_dB}	Der TAG “Level Offs“ bestimmt den Pegel des ARB Signals in der WV-Datei. Der Offset Pegel definiert den Offset des RMS und Peak Wertes relativ zur 16-bit Vollaussteuerung (-32767 to + 32767) = 0 dB.

Bezeichnung	Beschreibung
{SAMPLES: Samples}	Der TAG “Samples“ enthält die Anzahl an IQ-Werten, welche in der WV-Datei im ASCII Format enthalten sind.
{WAVEFORM-Length: #I0Q0I1Q1...IxQx...IN-1QN-1…}	Der TAG “Waveform” enthält die Längenangabe und die IQ-Werte der Waveform. Length 2 Byte: Anzahl der I/Q Paare * 4 (2 Byte pro I Wert und pro Q-Wert) + 1 byte (für “#”) IxQx: Binärdaten (16-bit signed integer) der I und Q Werter alternierend und mit I-Wert startend

3.6 Signalverarbeitung

Die Verarbeitung der erzeugten IQ-Daten durch die von Rohde & Schwarz bereitgestellte Software und weitere Programme, wie zum Beispiel Octave, Matlab und GnuRadio, ist von einigen Parametern abhängig.

Die Abtastrate F_s eines Signals, auch Sample Rate genannt, wird bei der Erzeugung des diskreten Signals festgelegt und muss bei Bearbeitung und Übertragung in andere Formate berücksichtigt werden. Zur Weiterverarbeitung von Abtastwerten in anderen Systemen ist gegebenenfalls eine Konvertierung der Abtastrate nötig. Diese Konvertierung kann in Form einer ganzzahligen Vervielfachung oder einer ganzzahligen Dezimierung der Abtastrate erfolgen. Durch die Ausführung einer ganzzahligen Vervielfachung, gefolgt von einer ganzzahligen Dezimation, kann eine Konvertierung durch einen rationalen Faktor realisiert werden.

Im Sinne der digitalen Signalverarbeitung wird die Erhöhung der Abtastrate auch als Interpolation bezeichnet. Der Faktor, um den sich die Abtastrate durch eine Interpolation erhöht, wird Interpolationsfaktor L genannt und ergibt sich aus.

$$L = \frac{Abtastrate\ am\ Ausgang}{Abtastrate\ am\ Eingang} = \frac{F_{sa}}{F_{se}} \tag{1}$$

In dem in Abbildung 14 dargestellten Beispiel wird die Folge x[n], n ∈ ℕ in zwei Schritten um den Faktor L = 3 interpoliert. Im ersten Schritt, der auch Upsampling oder zero padding genannt wird, sind nach jeder Stützstelle x[n] L-1 zusätzliche Punkte mit dem Wert 0 einzufügen. Die daraus resultierende Folge v[n] hat die L-fache Länge und nur jeder L-te Wert ist ungleich 0. Im Frequenzbereich entstehen durch diesen Schritt im Allgemeinen unerwünschte Abbilder bei Vielfachen von f_{se}, die das ursprüngliche Signal

verzerren. Diese Abbilder können im zweiten Schritt durch einen sogenannten Interpolationstiefpassfilter mit der Grenzfrequenz $f_G = \frac{F_S}{2}$ nach Nyquist herausgefiltert werden (siehe Abbildung 15 (b)). In Folge der Tiefpassfilterung entsteht die Ausgangsfolge y[n].[17]

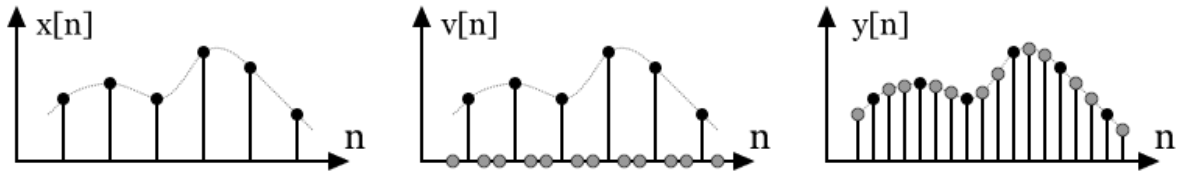

Abbildung 14: Beispiel Interpolation mit Faktor L=3 [16]

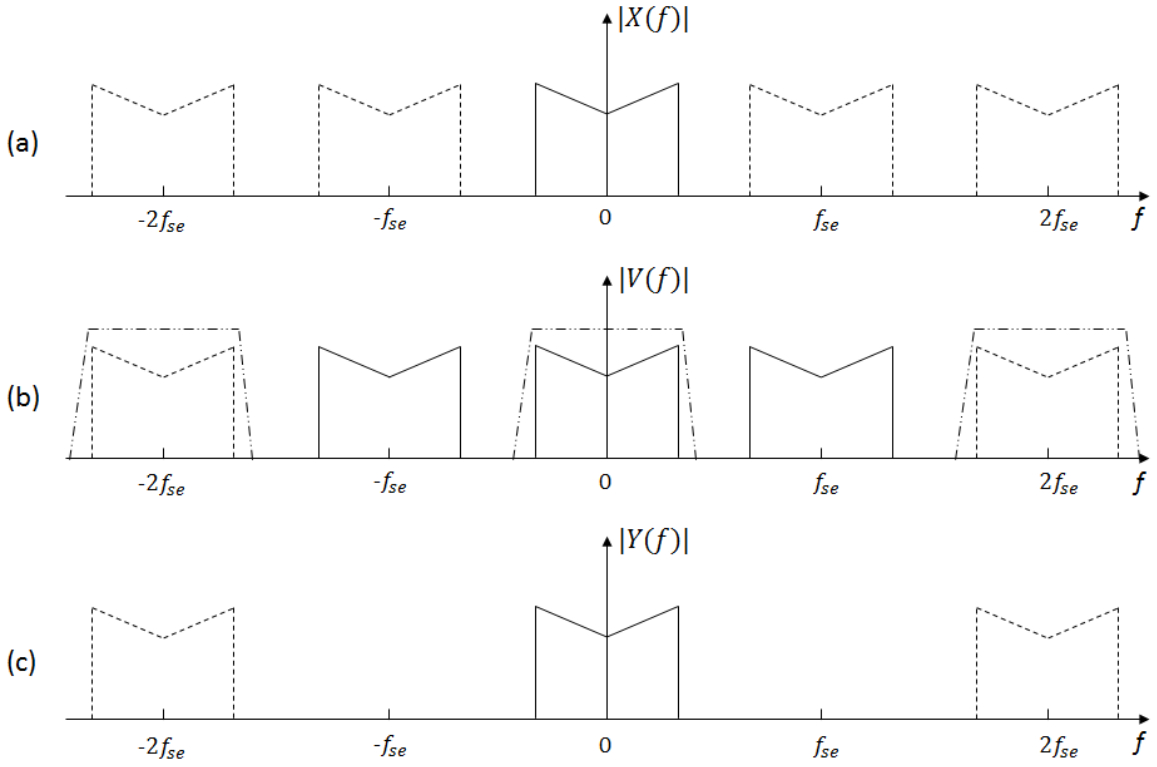

Abbildung 15: Upsampling und Interpolation im Frequenzbereich

Für den Fall, dass der Interpolationsfaktor L keine Primzahl ist, bietet sich eine Interpolation in mehreren Stufen an. So kann eine Interpolation um die Faktoren $L_1 = 2$ und $L_2 = 3$ zu einem geringeren Rechen- und Speicheraufwand als eine direkte Interpolation um den Faktor 6 führen. Grund hierfür ist die notwendige Anzahl an Filtertaps in den nachfolgenden Tiefpassfiltern. Bei der Interpolation in mehreren Stufen ist es empfehlenswert, höchstens zwei bis drei Stufen zu verwenden und vom kleinsten Faktor beginnend aufsteigend zu interpolieren.[17]

Die Reduzierung der Abtastrate eines Signals wird Dezimation genannt. Der Faktor, um den sich die Abtastrate durch die Dezimation verringert, wird Dezimationsfaktor M genannt und ergibt sich aus

$$M = \frac{Abtastrate\ am\ Eingang}{Abtastrate\ am\ Ausgang} = \frac{F_{se}}{F_{sa}} \tag{2}$$

Vergleichbar zu der Interpolation lässt sich die Dezimation in zwei Schritte aufteilen. Um Aliasing durch Verletzung des Nyquist Kriteriums zu vermeiden, wird das Eingangssignal x[n] mit einem Antialiasing Filter tiefpassgefiltert. Die Grenzfrequenz f_g des Tiefpassfilters ergibt sich aus der Abtastrate F_{sa} am Ausgang und der daraus folgenden maximalen Bandbreite B des Ausgangssignals.

$$f_g = B = \frac{F_{sa}}{2} \tag{3}$$

Da nach der Dezimation nur jeder M-te Abtastwert in der Ausgangsfolge y[n] vorkommt und dementsprechend die restlichen Werte nicht berechnet werden müssen, kann das Antialiasing Filter mit einem unter Umständen erheblich geringeren Aufwand realisiert werden.

Der auf den ersten Schritt der Filterung folgende zweite Schritt wird Downsampling genannt. Von der am Ausgang des Antialiasing Filters auftretenden Folge v[n] wird jeder M-te Wert in der Ausgangsfolge der Dezimation y[n] abgelegt.[17]

Um weiteren Rechenaufwand einzusparen, kann die Dezimierung stufenweise ausgeführt werden. Der Dezimationsfaktor darf dazu keine Primzahl sein, bei der Anordnung der mehrstufigen Dezimierungen bietet sich aufgrund der Rechenleistung eine absteigende Reihenfolge an.

Ist eine Konvertierung der Abtastrate um einen rationalen Faktor nötig, so bietet sich die Ausführung einer ganzzahligen Interpolation, gefolgt von einer ganzzahligen Dezimierung an. Da sowohl der letzte Schritt der Interpolation als auch der erste Schritt der Dezimation ein Tiefpassfilter ist, können die Filter zu einem einzigen Filter zusammengefasst werden, dessen Grenzfrequenz der kleineren der beiden ursprünglichen Filter entspricht.[17]

3.7 Medienzugriffsverfahren

3.7.1 Einleitung

Sobald sich mehrere Teilnehmer das Funkübertragungsmedium teilen, ist eine Regelung zur Vermeidung von Kollisionen und somit von Störungen notwendig. Weit verbreitete Funksysteme wie WLAN und Bluetooth nutzen entsprechende Medienzugriffsverfahren, um den Datendurchsatz im Funkübertragungsmedium zu erhöhen. Die realitätsnahe Nachbildung entsprechender Funksysteme mit einem SDR erfordert die Implementierung der entsprechenden Medien-zugriffsverfahren. Im folgenden Kapitel werden Medien zugriffsverfahren zur Kollisionsvermeidung und Störungsminderung (Interference Mitigation) vorgestellt.

3.7.2 Kollisionsvermeidung

Als Kollisionsvermeidungsmechanismus wird der Detect and Avoid (DAA) Mechanismus wie zum Beispiel Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) bei Funksystemen nach dem IEEE 802.11 [31] bezeichnet.

Clear Channel Assessment

Die Kollisionsvermeidung im CSMA/CA Verfahren durchläuft mehrere Schichten. Das Clear Channel Assessment als Teil der Physical Medium Dependant (PMD) und des Physical Layer Convergence Protocols (PLCP) beschäftigt sich mit der Belegung des umgebenen Funkmediums durch andere Stationen zu einem bestimmten Zeitpunkt. Die Beurteilung des Belegungszustandes kann entweder durch die Messung der empfangenen Leistung (Energy Detection, CCA-ED) oder durch die Auswertung der PLCP Präambel und daraus resultierend der Länge der aktuellen Übertragung (Carrier Sense, CCA-CS) ermittelt werden. Beide Verfahren nutzen eine Entscheidungsgrenze, die für CCA-ED 20 dB über der vom jeweiligen Gerät mindestens zu erkennenden Empfangsleitung für CCA-CS liegt.

3.7.3 Virtuelle Kollisonsvermeidung

Request to send/Clear to send

Das Request to send/Clear to send (RTS/CTS) beziehungsweise Multiple Access with Collision Avoidance (MACA) Verfahren ist in IEEE802.11 Systemen implementiert, die Ausführung ist allerdings optional. Ob das Verfahren in einem drahtlosen Netzwerk verwendet wird, hängt zum einen von der Medienbelegung und daraus folgend der Anzahl an Kollisionen und zum anderen von der Länge der zu sendenden Pakete ab. Eine Veranschaulichung des Ablaufes ist in Abbildung 16 gegeben.[18]

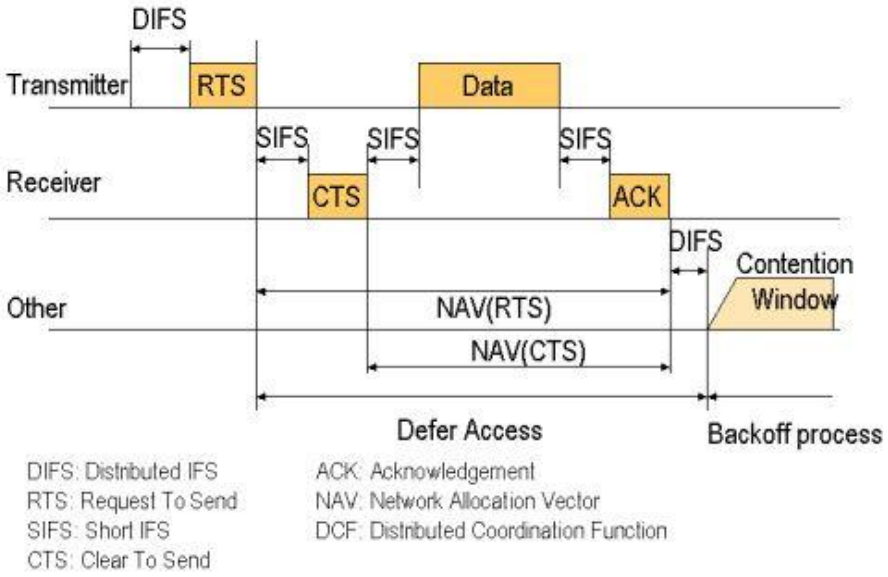

Abbildung 16: RTS/CTS Übertragung mit DCF [18]

Der in Abbildung 16 gezeigte Ablauf zeigt die Verwendung der dezentralen Distributed Coordination Function (DCF). Versucht ein Sender ein Paket an einen Empfänger zu senden, so wartet dieser eine durch den Distributed Control Function Interframe Space (DIFS) vorgegebene Mindestzeitspanne ab und lauscht im Medium, ob dieses frei ist. Erfolgt in dieser Zeitspanne keine Übertragung durch eine andere Station, so wird aus dem Intervall [0,T] eine zufällige Backoff-Zeit t gewählt, welche sich als Produkt aus einer durch die endliche Ausbreitungsgeschwindigkeit der Funkwellen ergebenen Zeit τ und einer Zufallszahl n ergibt. Das Intervall wird auch Contention Window genannt, da sich an dieser Stelle entscheidet, welche Station ihr Paket senden darf. Tritt zum Beispiel durch die Wahl der gleichen zufälligen Backoff-Zeit durch zwei Stationen eine Kollision auf, so wird wieder ein DIFS gewartet, und die Länge des Contention Windows verdoppelt. Bei mehreren aufeinander folgenden Kollisionen wächst die maximale Backoff-Zeit binär exponentiell an. Ein größeres Contention Window hat eine längere Wartezeit zur Folge, allerdings sinkt die Wahrscheinlichkeit gleich gewählter Backoff-Zeiten [25].

Sobald eine Station den DIFS und die Backoff-Zeit abgewartet hat, sendet sie eine Request-to-Send (RTS) Anfrage an die empfangende Station, die Anfrage enthält neben einer Frame Check Sequence (FCS), der Frame Control (FC) und der Sender- und Empfängeradresse die Dauer des zu sendenden Datenpakets. Da es an dieser Stelle möglich ist, dass sich zwei um das Funkübertragungsmedium konkurrierende Stationen während des Backoff-Prozesses aufgrund zu großer Entfernung oder Abschirmung nicht sehen konnten, sendet die empfangende Station nach einem Short Interframe Space (SIFS) ein Clear-to-Send (CTS), um zum einen der sendenden Station die Erlaubnis zum Senden zu geben und zum anderen allen Stationen in Reichweite des Empfängers die bereits im RTS enthaltende Dauer des Datenpakets mitzuteilen. Aus den Informationen

des RTS in Reichweite des Senders und des CTS in Reichweite des Empfängers können alle Stationen in Reichweite einen sogenannten Network Allocation Vector bilden, der die Belegungsdauer des Mediums durch die Übertragung angibt. Nach einem weiteren SIFS werden die Daten übertragen, bei erfolgreicher Übertragung erfolgt die Empfangsbestätigung durch ein Acknowledge (ACK) nach einem SIFS[25].

Point Coordination Function

Alternativ zu diesem dezentralen Medienzugriff mithilfe der DCF kann durch einen zentralen Koordinator wie zum Beispiel einen Access Point die Point Coordination Function (PCF), eine Art Polling Verfahren, durchgeführt werden. Der entsprechende PCF Interframe Space (PIFS) ist etwas länger als ein SIFS und etwas kürzer als ein DIFS und tritt an die Stelle des DIFS. Der Koordinator erhält damit trotz eines parallel laufenden DCF Verfahrens einen priorisierten Zugriff auf das Medium und kann über Polling Frames die einzelnen Stationen abfragen [25].

Hybrid Coordination Function

Der Standard IEEE 802.11e sieht zur Realisierung von Quality of Service (QoS) in WLAN Netzen die Hybrid Coordination Function (HCF) vor. Diese unterteilt sich in zwei Zugriffsmethoden, dem contention-freien Enhanced Distributed Channel Access (EDCA) und dem contention-basierenden HCF Controlled Channel Access (HCCA). Die zu sendenden Daten werden in Prioritätsklassen unterteilt, die sich insbesondere an der jeweiligen Anwendung orientieren. So wird eine Verzögerung in einer Sprachübertragung wie zum Beispiel VoIP kritischer eingestuft als die Übertragung zeitunkritischer Daten wie zum Beispiel Emails. Eine Übersicht der in IEEE802.11e definierten Klassen ist in Tabelle 7 gegeben, in Abbildung 17 ist ein Vergleich der aus den Klassen resultierenden Sendeverzögerung zu sehen.

Tabelle 7: Prioritätsklassen nach 802.1Q [28].

Priority Code Point (PCP)	Acronym	Traffic Type	Access Category (AC)	Designation
1	BK	Background	AC_BK	Background
2	EE	Excellent Effort	AC_BK	Background
0	BE	Best Effort	AC_BE	Best Effort
3	CA	Critical Applications	AC_BE	Best Effort
4	VI	Video	AC_VI	Video <100 ms Verzögerung
5	VO	Voice	AC_VI	Voice <10 ms Verzö-

Priority Code Point (PCP)	Acronym	Traffic Type	Access Category (AC)	Designation
				gerung
6	IC	Internetwork Control	AC_VO	Netzwerk
7	NC	Network Control	AC_VO	Netzwerk

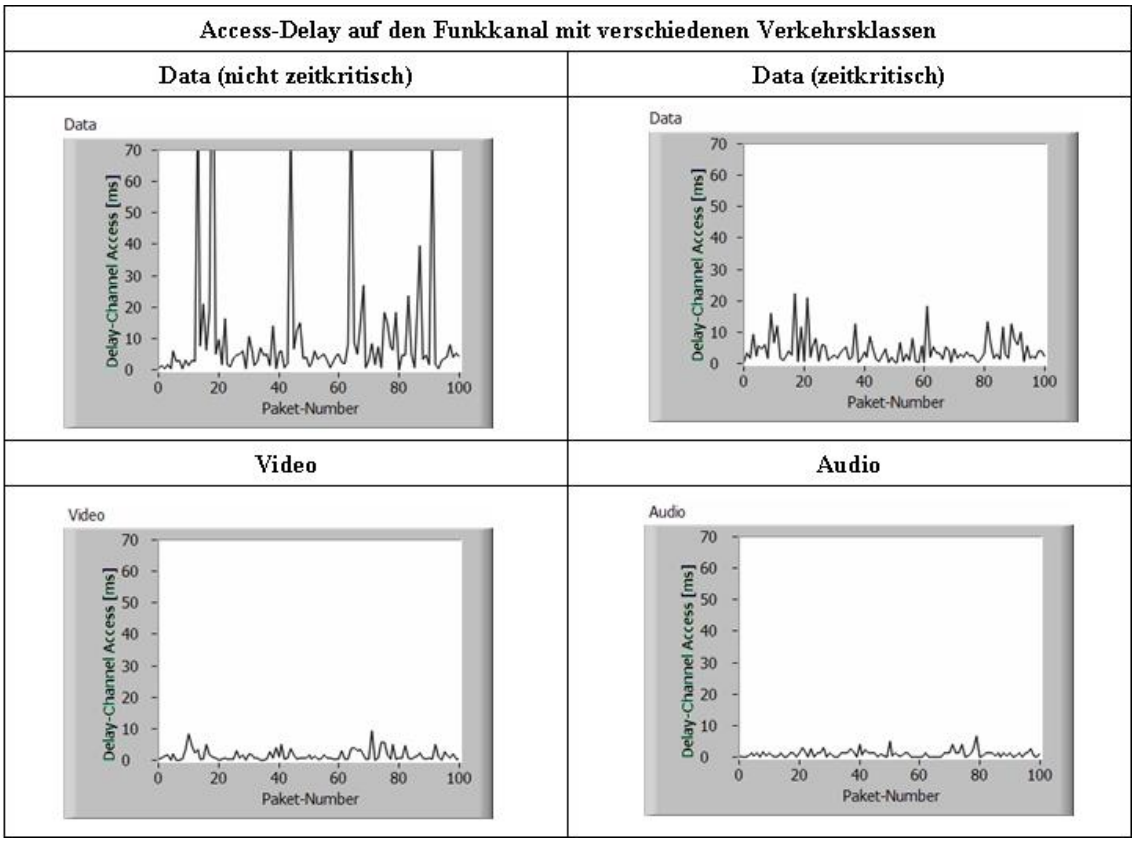

Abbildung 17: Sendeverzögerung Prioritätsklassen [22]

3.7.4 Kollisionserkennung

Ein typisches Beispiel für Kollisions- bzw. Interferenzerkennung verwendet die Funktechnologie Bluetooth mit dem Adaptive Frequency Hopping (AFH). Das Bluetooth Funksystem greift über das Frequency Hopping Spread Spectrum auf das Funkmedium im 2,4 GHz ISM Band zu, welches in 79 Kanäle mit einem Abstand von 1 MHz zwischen den einzelnen Kanälen aufgeteilt ist. Um die Interferenz mit anderen Funksystemen im 2,4 GHz Band gering zu halten, wird seit Version 1.2 das AFH verwendet. Die Kanäle, die sich zum Beispiel durch Interferenz mit anderen Funksystemen oder wegen anderer, frequenzstatischer Störer nicht zur Funkübertragung eignen, werden bei der Auswahl der für die Übertragung verwendeten Kanäle ignoriert. Beispiele für eine Bluetoothübertragung bei einer sendenden WLAN-Station sind in den Abbildungen Abbildung 18

(ohne AFH) und Abbildung 19 (mit AFH) gegeben. Die Aussparung der mit der WLAN-Übertragung zusammenfallenden Kanäle führt dazu, dass die Interferenz beider Funksysteme bei parallelem Betrieb deutlich reduziert wird.[24]

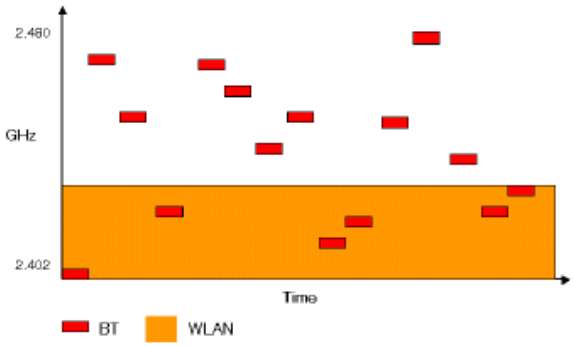

Abbildung 18: Bluetooth ohne AFH [23]

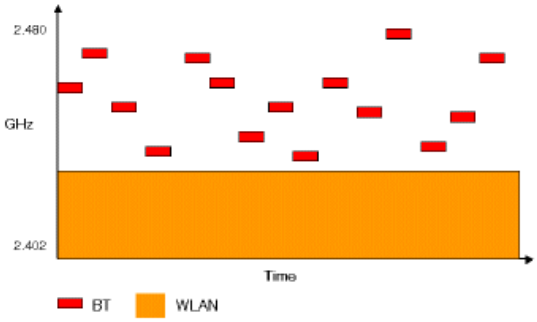

Abbildung 19: Bluetooth mit AFH [23]

Die Beurteilung, ob ein Kanal für die Übertragung geeignet oder durch weitere Sender gestört ist, ist nicht durch die Bluetooth Special Interest Group (SIG) vorgegeben, erfolgt aber vorwiegend anhand der Paketfehlerrate (PER), des Received Signal Strength Indicators (RSSI) oder des Signal-Rausch-Abstandes (SNR). Entsprechende Messungen werden fortlaufend vom Master und den Slaves des Pikonetzes durchgeführt und eine daraus resultierende Liste an nutzbaren Kanälen an die Teilnehmer des Netzes übermittelt.[24]

3.7.5 Medienzugriffsverfahren im Zusammenhang mit Cognitive Radio

Um die spärlich genutzten, aber von staatlichen Behörden reglementierten Frequenzbänder effektiver zu nutzen, sehen verschiedene Konzepte die Nutzung ansonsten ungenutzter Kanalkapazität vor, ohne dabei lizensierte Nutzer zu stören. Entsprechende unlizenzierte Nutzer würden mit Cognitive Radios, welche flexibel auf die Eigenschaften und die Belegung des Funkmediums reagieren, in dem lizenzierten Band senden. Neben dem Medienzugriff und Spectrum Sensing berücksichtigen die vorgeschlagenen Lösungen insbesondere die Einbindung höherer Schichten (siehe Abbildung 20).

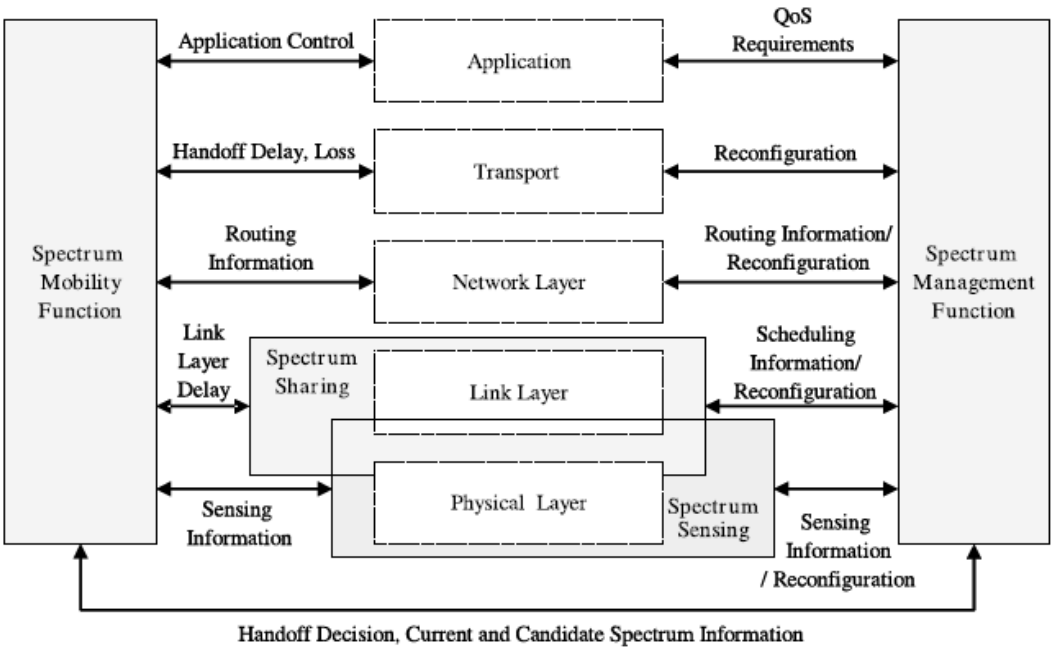

Abbildung 20: Cognitive Radio Funktionalitäten [29]

Die zur Erkennung anwesender lizenzierter Nutzer verwendeten Spectrum Sensing Verfahren lassen sich in verschiedene Kategorien unterteilen (siehe Abbildung 21). Neben der Transmitter Detection, die eigenständig durch das Cognitive Radio und in ähnlicher Weise bereits durch andere Funksysteme wie zum Beispiel WLAN durchgeführt wird, sehen einige Konzepte die Zusammenarbeit mehrerer Cognitive Radios zur Erkennung lizenzierter Nutzer vor. Ein Beispiel hierfür ist die Koordinierung mehrerer Cognitive Radios innerhalb eines Netzwerkes durch eine Basisstation, die in regelmäßigen Abständen Messungen veranlasst und die daraus errechneten nutzbaren Frequenzen an die Netzwerkteilnehmer sendet.

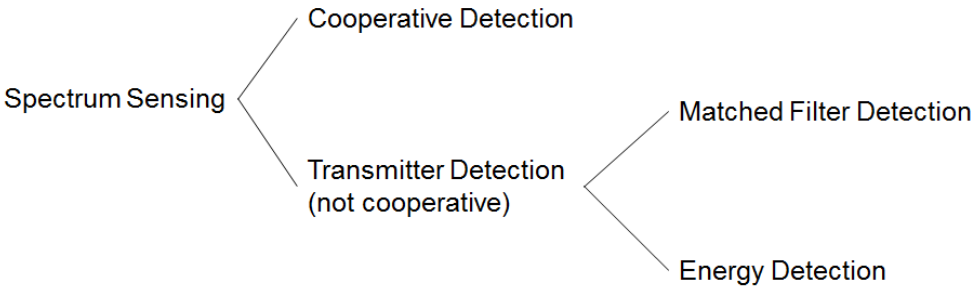

Abbildung 21: Gliederung Spectrum Sensing Verfahren

Neben unterschiedlichen Spectrum Sensing Verfahren sind in Abbildung 22 mit der grundlegenden Netzwerkarchitektur, der Zuweisung des zu nutzenden Spektrums und der Zugriffsmethode weitere Unterscheidungsmerkmale gegeben [29].

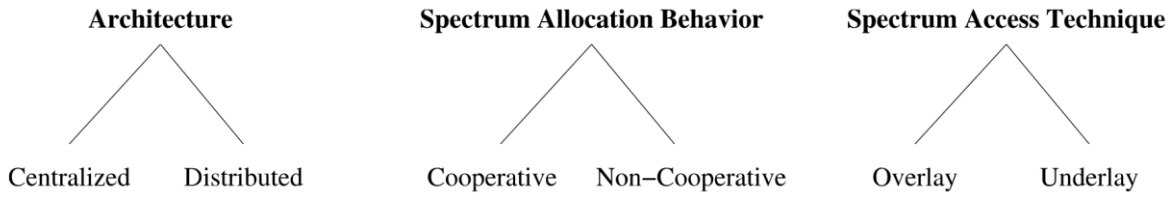

Abbildung 22: Unterscheidungskriterien Cognitive Radio Konzepte [29]

4 Implementierungskonzept

4.1 Ausgangssituation

Der verwendete Vektorsignalgenerator SMJ100A von Rohde &Schwarz kann für die Erzeugung verschiedener Signale, welche mit den jeweiligen Funksystemen konform sind, eingesetzt werden. Das Fehlen eines Empfängers und der eingeschränkte Frequenzbereich von unter 3 GHz grenzen die Verwendungsmöglichkeiten für bestimmte Szenarien jedoch ein. So kann zwar ein Datenstrom erzeugt und als Paket eines gewählten Protokollstapels abgestrahlt werden, allerdings können Mechanismen wie Medienzugriffsverfahren oder die Kollisionserkennung nicht emuliert werden. Dadurch werden beispielsweise Pakete nach 802.11 zyklisch ohne Abfrage der empfangenden Station oder durch FHSS modulierte Bluetooth Signale ohne zugehöriges Pikonetz abgestrahlt.

4.2 Konzepte

4.2.1 Umwandlung vorhandener Interferenzprofile

Die SDR-Plattform soll im Rahmen dieser Arbeit als Störer verwendet werden, welcher das Funkmedium mithilfe der verfügbaren Interferenzprofile belegt. Das Verhalten des Störers muss dazu dem Verhalten eines reellen Senders des entsprechenden Funkstandards möglichst ähnlich sein. Die verfügbaren Interferenzprofile werden mithilfe des VSG oder dem Programm WinIQSIM von Rohde & Schwarz als Wavefiles in Form von binären Daten erzeugt. Durch die Weiterverarbeitung der Daten in GnuRadio ist eine Realisierung der Interferenzprofile denkbar.

4.2.2 Erstellung von Interferenzprofilen mit GNU Radio

Eine weitere Möglichkeit, ein realitätsnahes Interferenzprofil zu erstellen, ist die Erzeugung und Verpackung von Nachrichten nach den Vorgaben der jeweiligen Protokollstapel in GNU Radio. Ein entsprechender Ablauf würde mit der Erzeugung von (zufälligem) Nutzdateninhalt durch eine Datenquelle beginnen und die für das Funksystem relevanten Protokolle in GnuRadio realisieren.

Die auf diese Weise erzeugten Interferenzprofile können im einfachsten Fall ein einzelnes, durch die in GNU Radio angegebenen Parameter definiertes Signal enthalten. Komplexere Interferenzprofile wie zum Beispiel die Belegung des Funkmediums durch mehrere Sender können beispielsweise durch die Addition mehrerer Interferenzprofile generiert werden.

Auf diese Weise erstellte Interferenzprofile können, wie in Abbildung 23 dargestellt, in GnuRadio direkt aus der entsprechenden IQ-Datei geladen und ausgeführt werden.

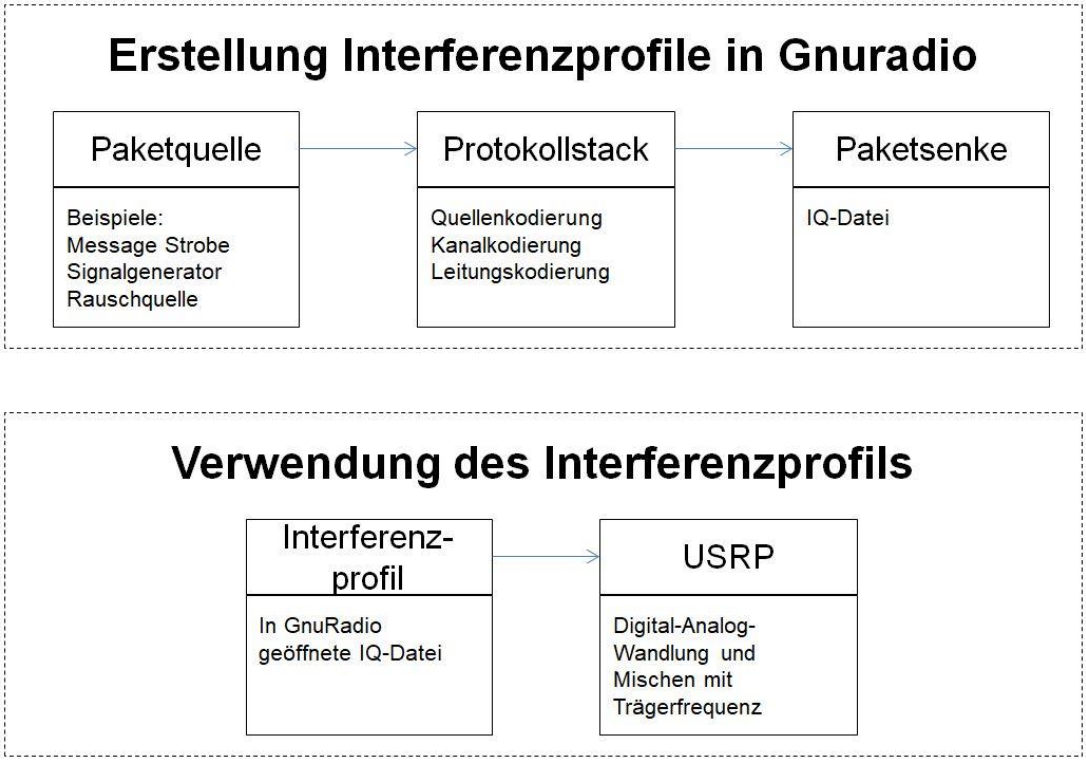

Abbildung 23: Konzept Interferenzprofil aus GnuRadio

4.2.3 Aufzeichnung realer Interferenzprofile

Die Erstellung von Interferenzprofilen kann durch das Aufzeichnen von Signalen mit der USRP-Plattform erfolgen. Dazu wird ein Frequenzband für ein bestimmtes Zeitintervall aufgezeichnet und direkt oder nach kurzer Bearbeitung in GnuRadio in eine Datei geschrieben. Die Quelle des aufgezeichneten Signales kann die mit einer geeigneten Antenne aufgenommene, unmittelbare Funkumgebung oder ein über ein Kabel angeschlossenes Funkgerät sein.

Die auf diese Weise erzeugten Interferenzprofile enthalten alle erfassten Signale im betrachteten Frequenzband und Zeitintervall der Messung.

4.3 Bewertung der Konzepte

Die Beurteilung, ob ein bestimmtes Konzept für die Umsetzung eines Interferenzprofiles geeignet ist, muss grundsätzlich von dem Erstellungsaufwand, der daraus resultierenden und der geforderten Qualität abhängig gemacht werden. So ist es beispielsweise unnötig, in einem Interferenzprofil einen kompletten Protokollstapel zu realisieren, wenn die dadurch erzeugten Daten bzw. Headerinhalte nicht von einem entsprechenden Funksystem ausgewertet werden.

Können aus bereits vorhandener Software IQ-Daten unter Rücksichtnahme bestimmter Parameter erzeugt und in ein für GnuRadio lesbares, binäres Format gebracht werden, so beschränkt sich die Komplexität der Realisierung in GnuRadio gegebenenfalls auf die noch nicht umgesetzten Parameter. Dem geringeren Realisierungsaufwand in GnuRadio steht der Übersetzungsaufwand der erstellten IQ-Daten in ein für GnuRadio lesbares Format entgegen.

Die Aufnahme eines Interferenzprofiles mithilfe der USRP-Plattform erfordert den geringsten Aufwand, beschränkt allerdings die Konfigurationsmöglichkeiten des entstandenen Profils. So entspricht das erfasste Signal zwar einem Funksystem, allerdings lassen sich Eigenschaften wie Modulation oder Inhalt nicht verändern.

Im Rahmen dieser Arbeit werden Interferenzprofile direkt in GnuRadio beziehungsweise durch eine Aufnahme des Funkübertragungsmediums der USRP-Plattform erzeugt. Dadurch sind Parameter und Eigenschaften des erzeugten Interferenzprofiles ausschließlich in den entsprechenden Konfigurationen in GnuRadio zu finden, Änderungen durch Parameter aus weiterer Software beziehungsweise der Übersetzung binärer Daten aus dieser Software können ausgeschlossen werden. Mithilfe einer Antenne oder durch weitere Funksysteme erzeugte und von der USRP-Plattform aufgenommene Profile werden in Zeit, Frequenz und Amplitude verändert und wieder abgespielt.

5 Implementierung

5.1 Einleitung

Die aus der Software Defined Radio Plattform und GnuRadio bestehende Verarbeitungskette hat mehrere Möglichkeiten zur Konfiguration und Anpassung an ein zu betrachtendes Funksystem. Neben den in Kapitel 4 vorgestellten Konzepten sollen im folgenden Abschnitt insbesondere Kriterien zur Auswahl der verwendeten Konfigurationsmöglichkeit vorgestellt werden.

5.2 Signalverarbeitungskette

Die Protokolle der untersten beiden Schichten des OSI-Referenzmodells, die Bitübertragungsschicht und die Sicherungsschicht, stellen zeitkritische Funktionalitäten zur Verarbeitung von Daten zur Verfügung, die einen Rahmen von nur einigen Mikrosekunden haben. Müssten entsprechende Funktionen vom PC übernommen werden, so würden die Daten durch das Ingress/Egress Interface zunächst in Pakete der Netzwerkschicht wie zum Beispiel TLP bei PCIe oder UDP bei Ethernet gepackt, an den verarbeitenden PC geschickt, dort ausgewertet und die Antwort über PCIe oder Ethernet wieder als Paket der Netzwerkschicht an die SDR-Plattform zurückgesendet werden. Die Verarbeitungszeit ginge je nach zu verarbeitenden Daten in den einstelligen Millisekundenbereich, was bei zeitkritischen Anwendungen wie beispielsweise der Berechnung des NAV nach Empfang einer RTS-Anfrage oder beim Medienzugriffsverfahren bei CSMA/CA nicht ausreicht.

Um den Zeitverlust durch das Ingress/Egress Interface zu vermeiden, können die Daten innerhalb des FPGA durch Implementierung von Intellectual Property (IP) Cores bzw. Softcores verarbeitet werden. Zusammen mit der NoC-Shell, welche Paketmanagement und Flusskontrolle zur Anbindung an den Crossbar Switch enthält und die Schnittstelle zum AXI Bus darstellt, werden diese Module Computation Engines (CE) genannt (siehe Abbildung 24). Neben den vorgefertigten Beispielen können eigene CE entwickelt und auf dem FPGA der SDR-Plattform implementiert werden.

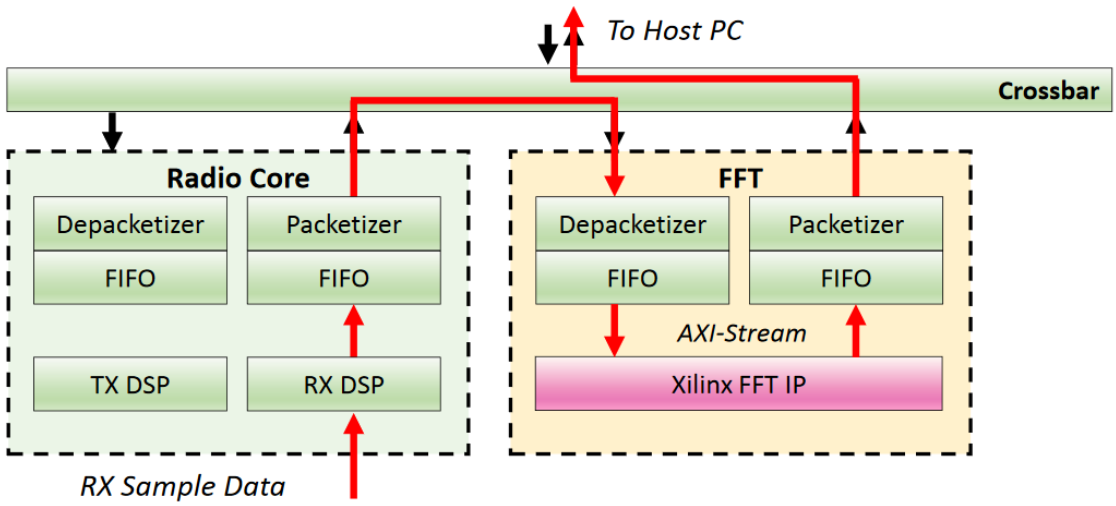

Abbildung 24: Computation Engines [21]

Abbildung 25 zeigt den Aufbau eines mithilfe des Modtools rfnocmodtool erzeugten IP Cores zur Verstärkung eines Signals. Neben wiederverwendbarem Verilog- bzw. VHDL-Code können eigene Funktionen wie in diesem Fall die Verstärkung des Signals durch

Multiplikation der einzelnen Inphase- und Quadratur-Komponenten mit einer Konstanten implementiert werden.

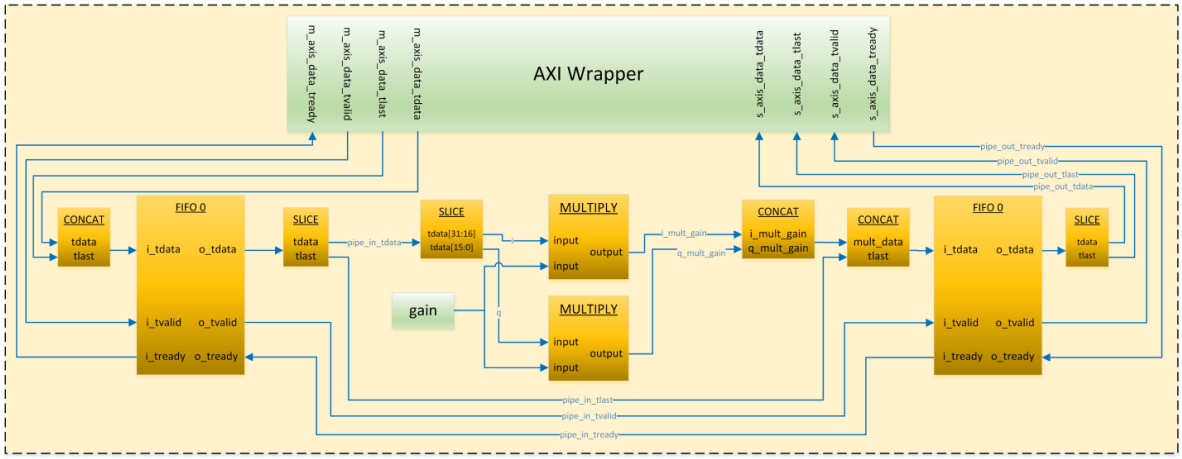

Abbildung 25: Beispiel Computation Engine Gain [20]

Wesentlicher Vorteil für die in dieser Arbeit vorgesehene Verwendung ist die ausreichend schnelle Auswertung auf ein empfangenes Signal, um zum Beispiel auf die Präambel oder den Header eines empfangenen Paketes rechtzeitig reagieren zu können. So wie Funksysteme in der Regel entweder nur auf Medienzugriffsmechanismen aus dem eigenen verwendeten Standard oder zusätzlich bei fremden Funkstandards auf Energy Detection zugreifen, so können die IP Cores für die Demodulation des Signals, das Auslesen eines bestimmten Datensatzes aus einem empfangenen Paket oder die Messung der Leistung in einem bestimmten Frequenzband genutzt werden. Für weiterführende Aufgaben, insbesondere die Ausführung von Protokollen der Sicherungsschicht, ist der Microblaze Softcore Prozessor besser geeignet.

Ein Implementierungskonzept der Energy Detection auf der SDR-Plattform ist in Abbildung 26 dargestellt. Die zweite Erweiterungskarte kann dazu genutzt werden, Abtastwerte aus dem Funkmedium entweder direkt vor einem Sendeversuch oder unabhängig von einem Sendeversuch kontinuierlich zu erfassen. Der FPGA übernimmt mit dafür zugeschnittenen CE die Berechnung des Leistungsdichtespektrums und entscheidet mithilfe eines Softcore Prozessors, ob die zu sendenden Daten im Rahmen der Energy Detection abgestrahlt werden können.

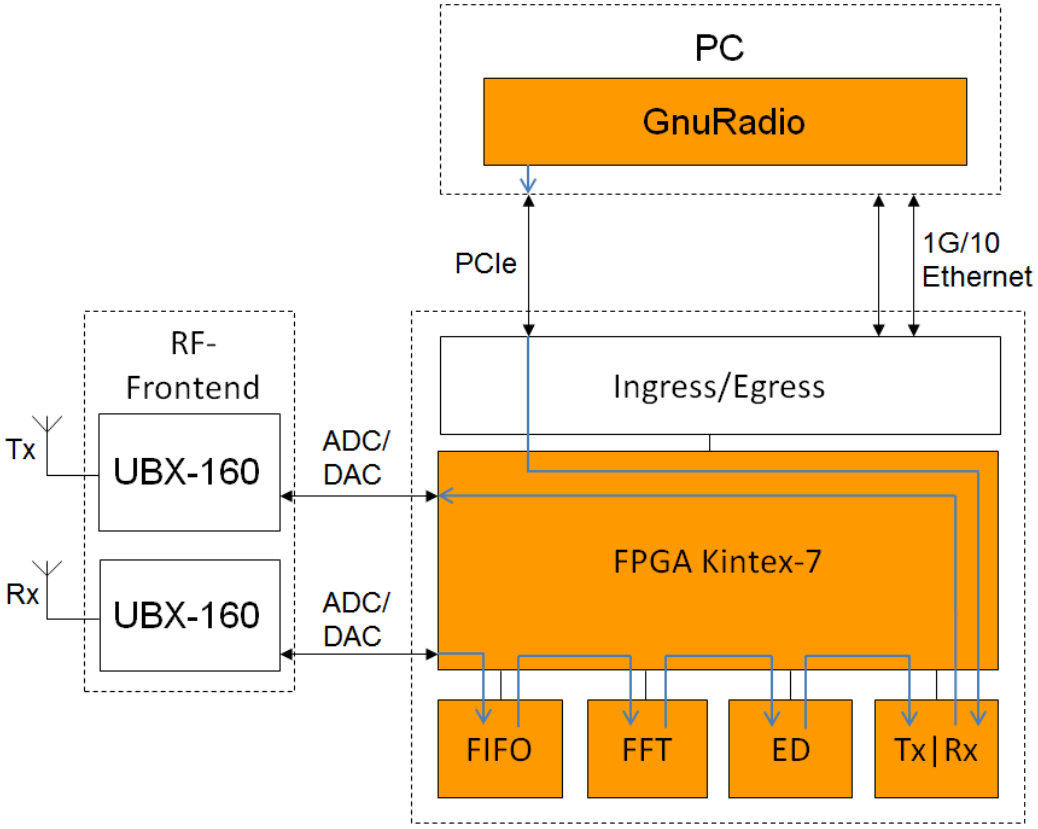

Abbildung 26: Konzept Störer mit Energy Detection

Auf ähnliche Weise kann das Carrier Sensing als Funktionalität für einzelne Funkstandards implementiert werden. Dies erfordert beispielsweise bei 802.11 eine Erkennung und Auswertung des Headers oder bei dem von Bluetooth verwendeten FHSS die Synchronisierung mit dem Master des betrachteten Pikonetzes.

5.3 Erstellung und Verwendung der Interferenzprofile

Mit Instrumenten und Software von Rohde & Schwarz können binäre Dateien erzeugt werden, die sich als eine Reihe von Inphase- und Quadraturwerten durch entsprechende Signalgeneratoren interpretieren lassen. Die so erzeugten Dateien können neben einzelnen Trägern und analogen Modulationsverfahren auch verschiedene digitale Modulationsverfahren und Übertragungsprotokolle enthalten. Insbesondere ist die Erzeugung von WLAN und Bluetooth Paketen unter Berücksichtigung bestimmter allgemeiner oder protokollspezifischer Parameter möglich.

Die so erzeugten Waveform Dateien sind, sofern sie auf einem PC oder von bestimmten Signalgeneratoren erzeugt wurden, verschlüsselt.

Als erste und aufwandsärmere Variante lassen sich die generierten Waveform-Dateien zwar direkt auslesen, die so abgestrahlte Leistung entspricht allerdings nur dem zeitlichen Verhalten des entsprechenden Signals. Die in Abbildung 27 dargestellte Alternative zeigt

die Verarbeitungskette, welche mithilfe des USRP und GnuRadio eine unverschlüsselte Datei mit IQ-Abtastwerten erzeugt. Der Signalgenerator wird über ein möglichst gut geschirmtes Koaxialkabel und ein geeignetes Dämpfungsglied an den USRP angeschlossen. Die vom Signalgenerator erzeugten Signale können im Rahmen der möglichen Bandbreite des USRP erfasst und als IQ-Datenstrom ohne speziell vorgesehene Signalverarbeitung durch den FPGA an den PC geschickt werden. Da sich in dieser Verarbeitungskette keine zeitkritischen Prozesse befinden, kann die gesamte Signalverarbeitung durch Programme wie GnuRadio oder Octave auf dem PC durchgeführt werden.

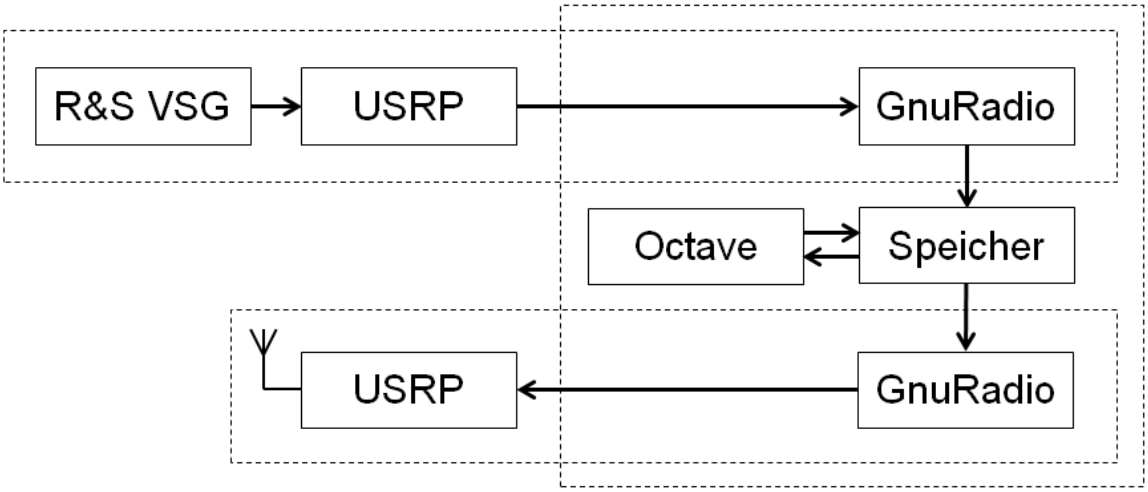

Abbildung 27: Verarbeitungskette Interferenzprofile

6 Validierung

6.1 Messaufbau

Zur Validierung der in 4.2 vorgestellten Konzepte werden im folgenden Abschnitt einige grundlegende Kriterien herangezogen, um die Qualität des USRP als Sender beziehungsweise als Empfänger zu beurteilen. Beispiele für diese Kriterien sind das Verhalten im Zeitbereich, im Frequenzbereich und die Signalleistung im Pass- und Sperrband. Der daraus resultierende Messaufbau sieht die USRP-Plattform entweder als Empfänger oder als Sender vor. Dementsprechend können beispielsweise ein VSG oder ein in das unmittelbar umgebene Funkmedium abstrahlendes Funksystem als Signalquelle für eine empfangende USRP-Plattform und ein Echtzeitspektrumanalysator (RSA) oder ein Signalanalysator (FSV 13) als Empfänger für eine sendende USRP-Plattform verwendet werden (siehe Abbildung 28).

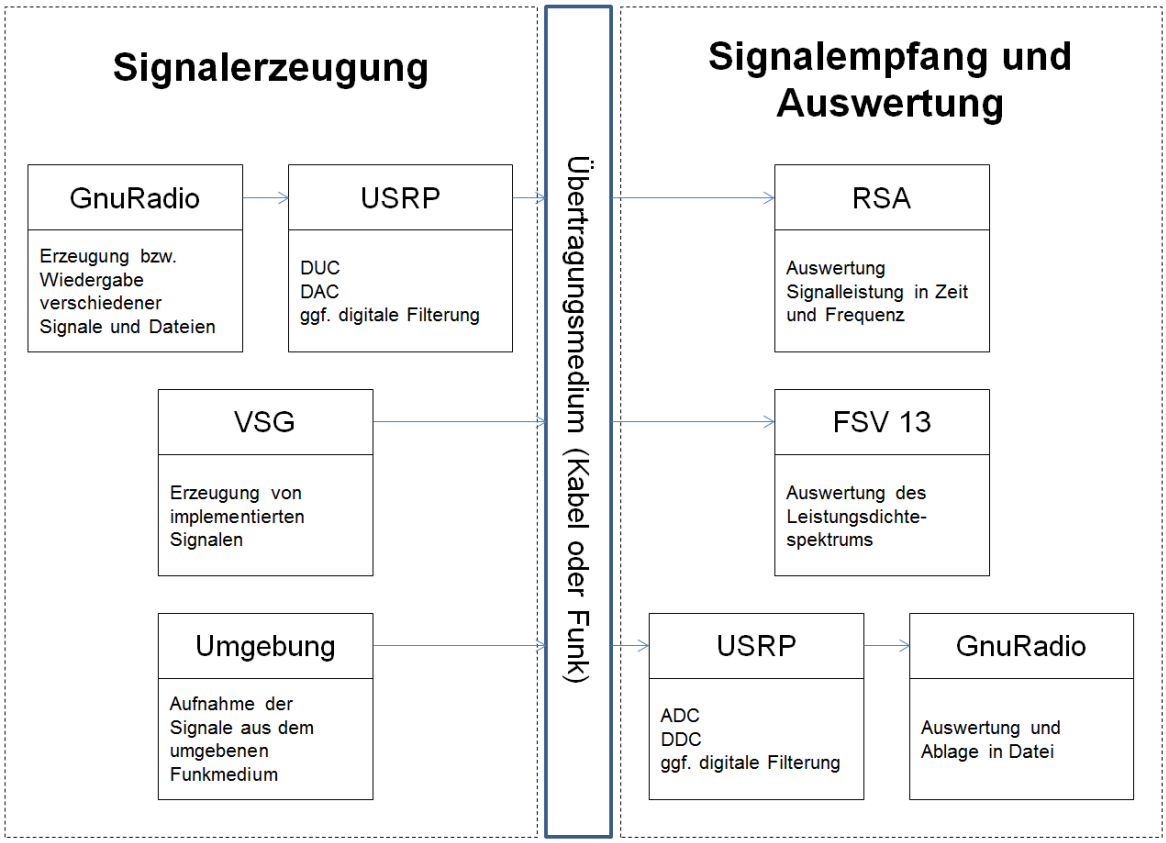

Abbildung 28: Messaufbau

6.2 Filtereigenschaften

Ein wesentliches Qualitätsmerkmal im Sendebetrieb ist die Unterdrückung der im Sperrbereich abgestrahlten Leistung. Als Konfigurationsmöglichkeiten für die USRP-Plattform stehen digitale Filter in GnuRadio und FIR-Filter als IP für den FPGA zur Verfügung. Die Filter auf den UBX-160 Erweiterungskarten sind nicht über Software konfigurierbar.

In Abbildung 29 ist die Signalverarbeitungskette in GnuRadio zu sehen. Die Random Source erzeugt einen zufälligen Wert zwischen 0 und 32766, mittels IFFT werden diese Zufallszahlen durch den OFDM Mod Block auf 200 orthogonale Träger moduliert. Die daraus entstehenden Inphase- und Quadraturwerte werden dann, je nach Messung, entweder in GnuRadio (Decimating FIR Filter) oder auf dem FPGA (RFNoC FIR) im Basisband tiefpassgefiltert. Da die Größe des FIR-Filters auf dem FPGA standardmäßig auf 41 Taps begrenzt ist, wurden die Filter dementsprechend berechnet. Hierzu fand für die verschiedenen Filtertypen die Fenstermethode Anwendung. Nach der Filterung wird das Signal digital-analog gewandelt und von der Erweiterungskarte entsprechend der Einstellungen analog weiterverarbeitet.

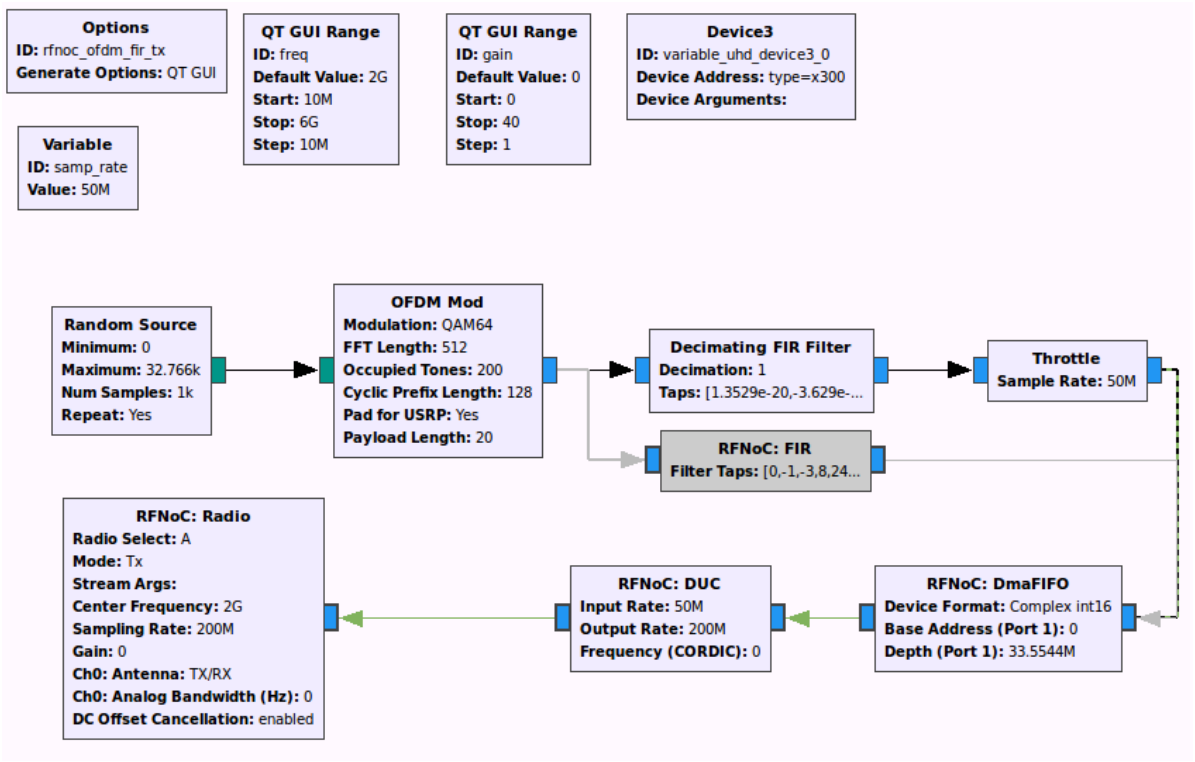

Abbildung 29: Generierung OFDM-Signal in GnuRadio

Die in den Abbildungen 30 bis 35 dargestellten Messungen wurden mit dem Signalanalysator FSV13 bei einer Mittenfrequenz von 2 GHz, einer Bandbreite von 100 MHz, einer Auflösebandbreite (RBW) von 50 kHz und einer Videobandbreite (VBW) von 50 kHz aufgenommen. Gemessen wurde der Maximalwert des Leistungsdichtespektrums über einen Zeitraum von 10 Minuten. Je Messung sind 24001 gleichverteilte Punkte mit einem Abstand von 4,2 kHz aufgenommen worden. Wird über den Eingang keine Leistung eingespeist, so liegt das Grundrauschen bei -67 dBm.

In Abbildung 30 ist ein in GnuRadio erzeugtes, ungefiltertes OFDM-Signal zu sehen. Mit etwa 30 dB Differenz zur Leistung im Sperrbereich hebt sich das Signal deutlich heraus, allerdings führt eine weitere Verstärkung ebenfalls zu einer höheren abgestrahlten Leistung im Sperrbereich. Idealerweise läge die von der USRP-Plattform im Sperrbereich abgestrahlte Leistung unterhalb des Rauschpegels.

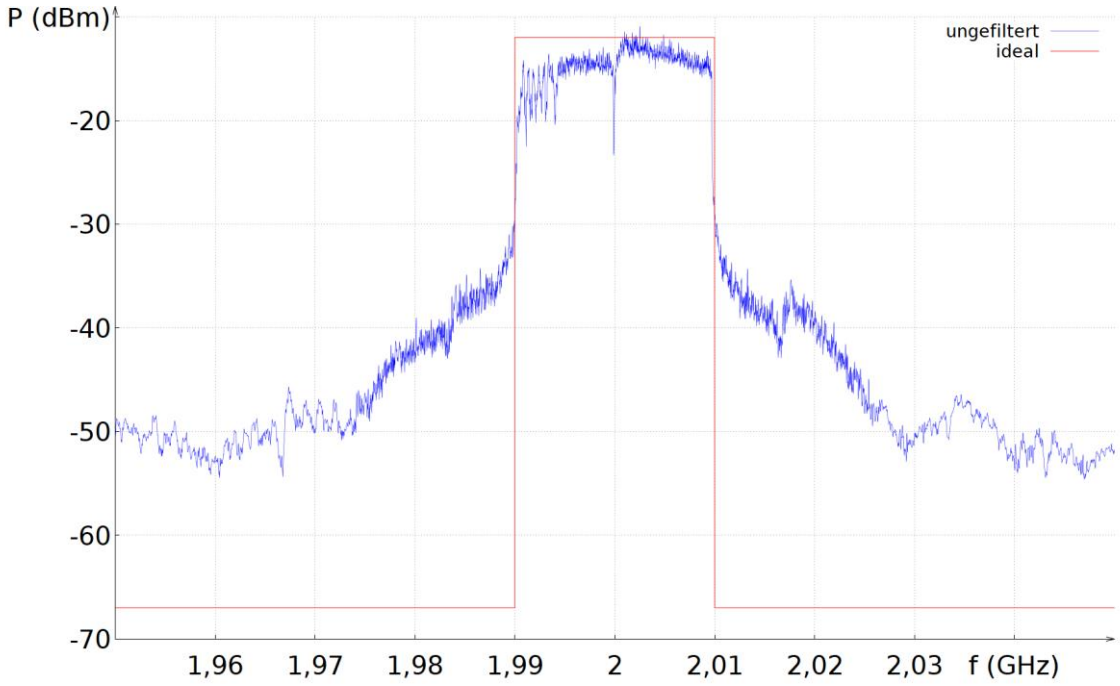

Abbildung 30: ungefiltertes OFDM Signal

Ein nahezu ideales Verhalten weist die in Abbildung 31 dargestellte Signalerzeugung des Vektorsignalgenerators auf. Das 17 MHz breite Signal weist, vergleichbar mit der Unterdrückung der Mischfrequenz von 2 GHz beim USRP einen Leistungseinbruch bei 2 GHz auf. Auffallend ist der mit 2 MHz sehr schmale und mit 40 dB steile Übergangsbereich, welcher bei der in den folgenden Abbildungen dargestellten USRP-Plattform trotz Filterung deutlich breiter und flacher ausfällt.

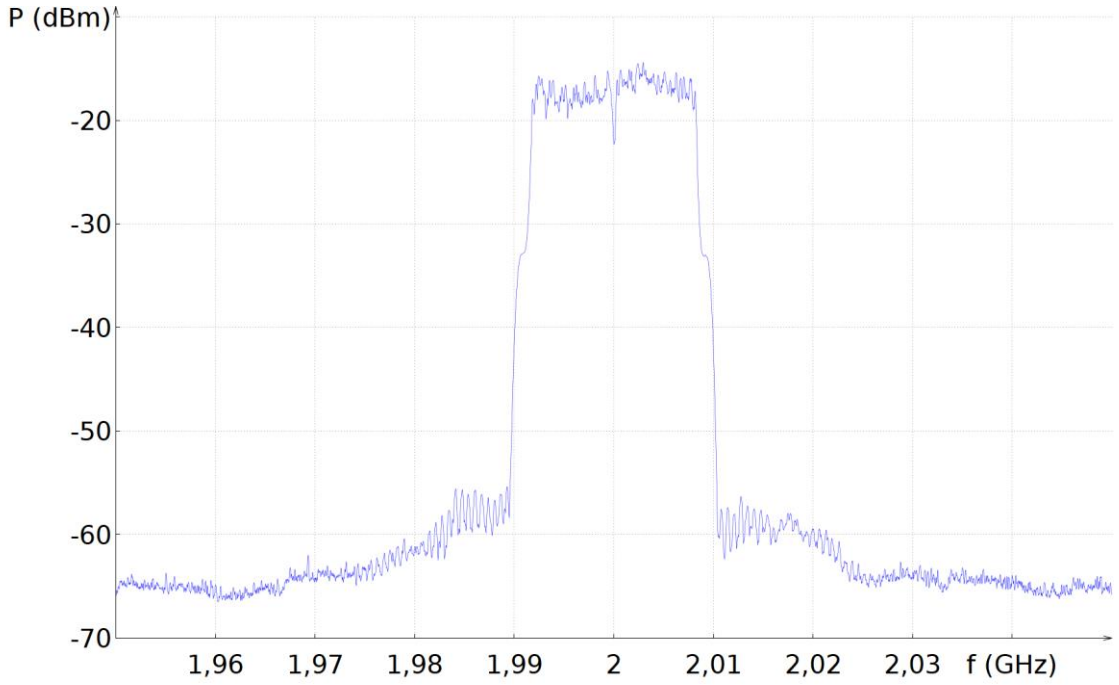

Abbildung 31: Durch VSG erzeugtes OFDM Signal

Die Filterkoeffizienten des in Abbildung 33 verwendeten Kaiserfilters wurden mithilfe der fir1-Funktion in Octave berechnet, der Amplitudengang in Abbildung 32 wurde durch die freqz-Funktion in Octave dargestellt. Neben Fenster-, Filtertyp und Anzahl der zu nutzenden Taps wurde für den Parameter der Grenzfrequenz w der Wert 0.4 gesetzt, welcher sich aus der Abtastrate und der gewünschten Grenzfrequenz ergibt:

$$w = \frac{2 \cdot f_G}{f_s} = \frac{2 \cdot 10\ MHz}{50\ MHz} = 0{,}4 \tag{4}$$

Die blau dargestellten Messwerte entsprechen dem gefilterten OFDM-Signal, das ungefilterte Signal ist zum Vergleich in rot mit enthalten. Durch die Verwendung des Kaiserfilters kommt mit einer um 5 dB höheren Dämpfung im Sperrbereich zwar eine leichte Verbesserung zustande, allerdings wird im Bereich außerhalb des von GnuRadio verarbeiteten und hochgemischten Basisbandes weiterhin Leistung oberhalb des Grundrauschens abgestrahlt.

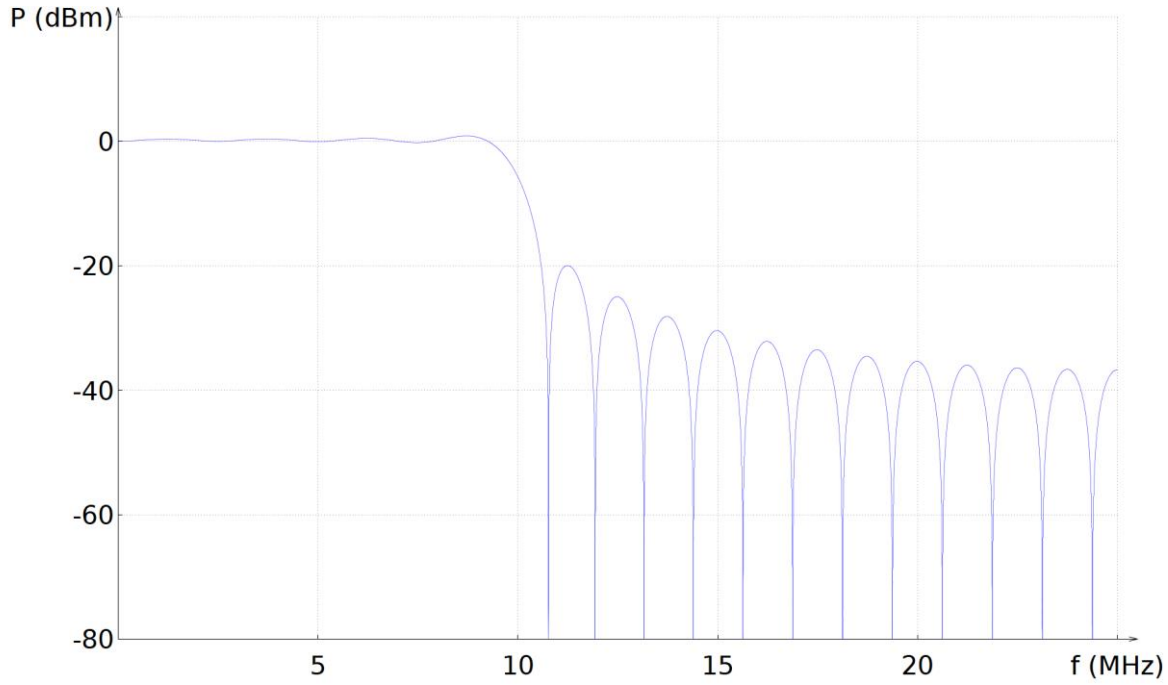

Abbildung 32: Amplitudengang Kaiserfilter

Nach dem in Abbildung 32 dargestellten Amplitudengang wäre eine Dämpfung der Leistung im Sperrbereich um mindestens 20 dB zu erwarten gewesen.

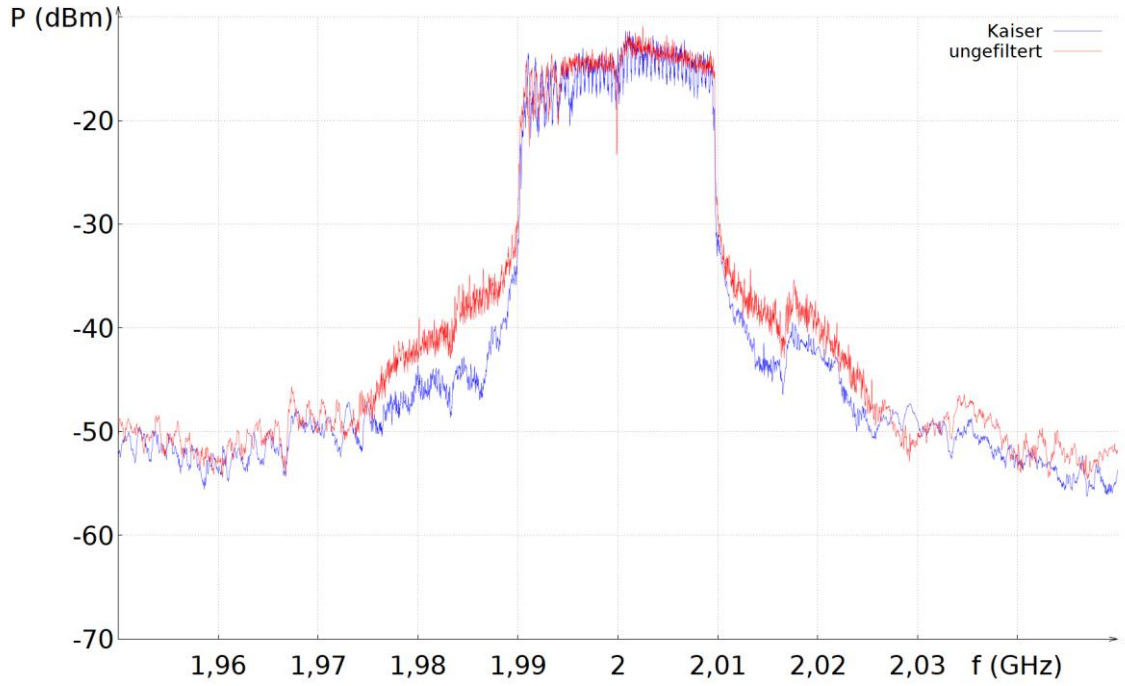

Abbildung 33: Kaiser Tiefpass bei 2 GHz

Zur Berechnung der Filterkoeffizienten eines Tschebyscheff Filters wurde der in Octave als remez-Funktion implementierte Parks-Mcclellan Algorithmus verwendet.

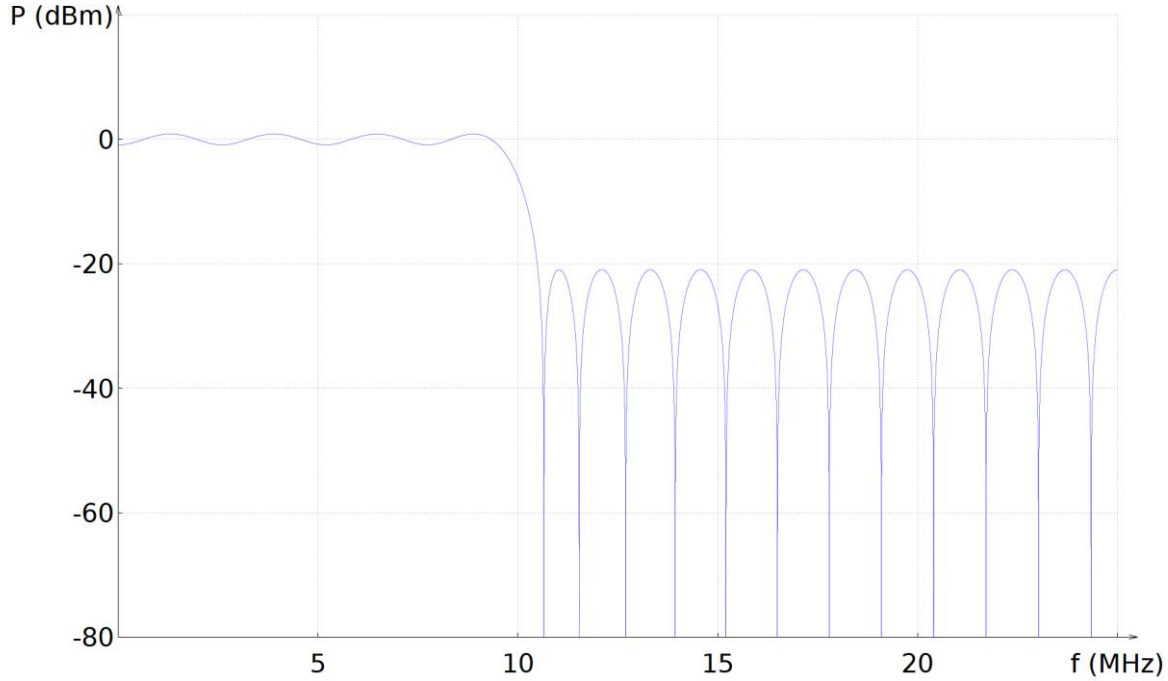

Abbildung 34: Amplitudengang Tschebyscheff-Filter

In Abbildung 35 ist das aus der Filterung resultierende OFDM-Signal in blau und das ungefilterte OFDM-Signal in rot zu sehen. Im Gegensatz zum Kaiserfilter ist nur eine

geringe Verbesserung gegenüber dem ungefilterten Signal zu sehen, dafür erscheint im Passband eine zusätzliche Welligkeit um die 3 dB.

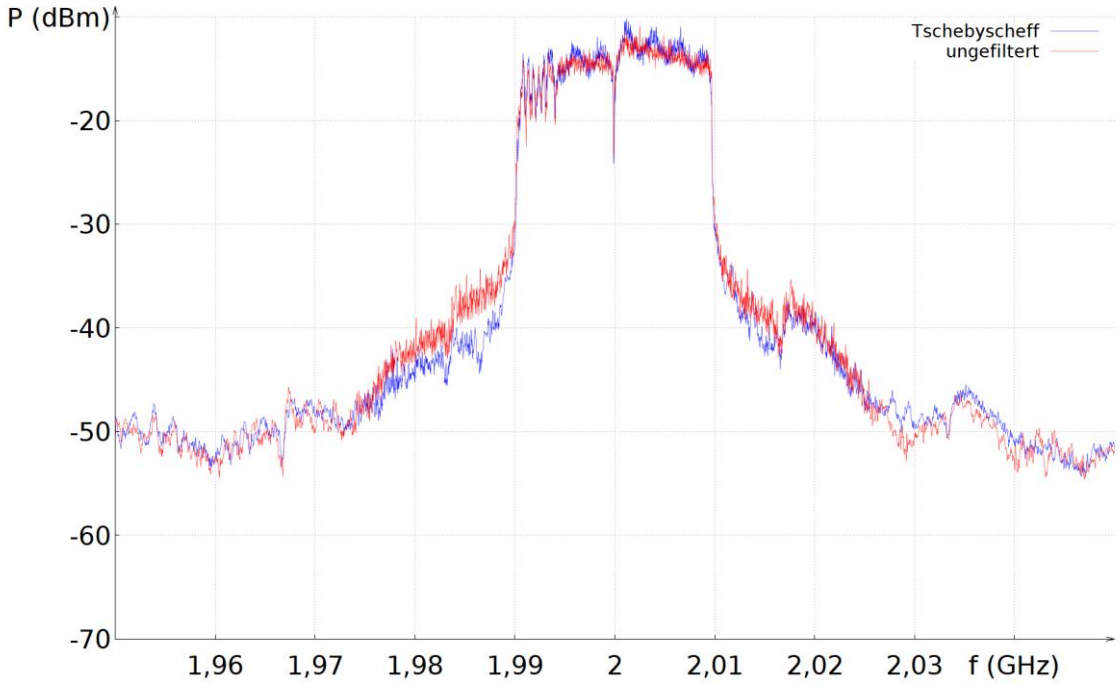

Abbildung 35: Tschebyscheff Tiefpass bei 2 GHz

6.3 Verhalten im Zeitbereich

Die parallele Ausführung der Signalerzeugung, Bearbeitung und Übertragung an die USRP-Plattform durch GnuRadio kann die Rechenkapazität des Prozessors vollständig auslasten. Ein daraus resultierender Effekt sind Unterbrechungen, wie sie in Abbildung 36 (Spektrogramm) und Abbildung 42 (Messung der Amplitude gegen Zeit) zu sehen sind. In GnuRadio wurde ein kontinuierliches OFDM-Signal bei 50 MSPS mit zufälligem Inhalt erzeugt und durch ein FIR-Filter mit 41 Taps gefiltert, die Messung erfolgte mit einem Echtzeitspektrumanalysator. Die in den Spektrogrammen in Abbildung 36, Abbildung 37 und Abbildung 38 dargestellte Leistung liegt zwischen -40 dBm (rot) und -75 dBm (blau), der Frequenzbereich liegt zwischen 1,96 GHz und 2,04 GHz. Das dargestellte zeitliche Intervall ist 10 ms lang. Für die Messungen in diesem Abschnitt wurde die USRP-Plattform mithilfe eines Koaxialkabels an den Echtzeitspektrumanalysator angeschlossen.

Abbildung 36 zeigt das Sendeverhalten bei paralleler Erzeugung und Filterung des OFDM-Signals in GnuRadio. Stehen die IQ-Werte nicht in der Taktrate zur Verfügung, wie sie von der USRP-Plattform erwartet werden, so wird anstelle des fehlenden Wertes keine Leistung abgestrahlt. Die Ausgabekonsole von GnuRadio gibt in diesem Fall ein „U“ aus, um den Underflow zu signalisieren. In dem dargestellten Fall wurde von idealerweise 100% Sendedauer nur in 23,8 % der Zeit Leistung abgestrahlt (vgl. Tabelle 8).

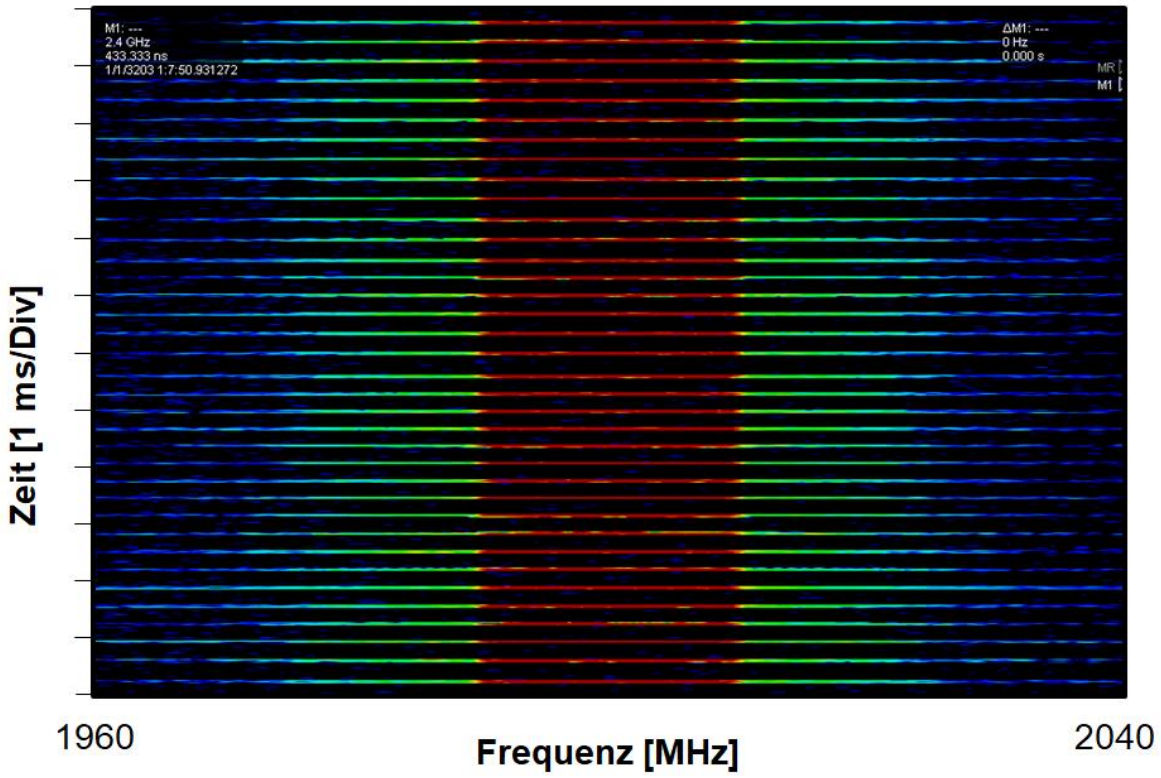

Abbildung 36: Filterung durch Software

Je nach Implementierung eines Filters können im schlimmsten Fall pro IQ-Wert 41 Multiplikationen und 41 Additionen anfallen, zusammen mit einer hohen Taktrate entsteht so eine erhebliche Last für die CPU. Wird das Filter hingegen nicht berücksichtigt, so kann der PC mit GnuRadio die geforderte Taktrate von 50 MSPS an die USRP-Plattform liefern. In Abbildung 37 ist das ohne FIR Filter erzeugte OFDM-Signal zu sehen. Relativ zur gesamten Sendezeit wird das Signal in diesem Fall nahezu durchgängig abgestrahlt.

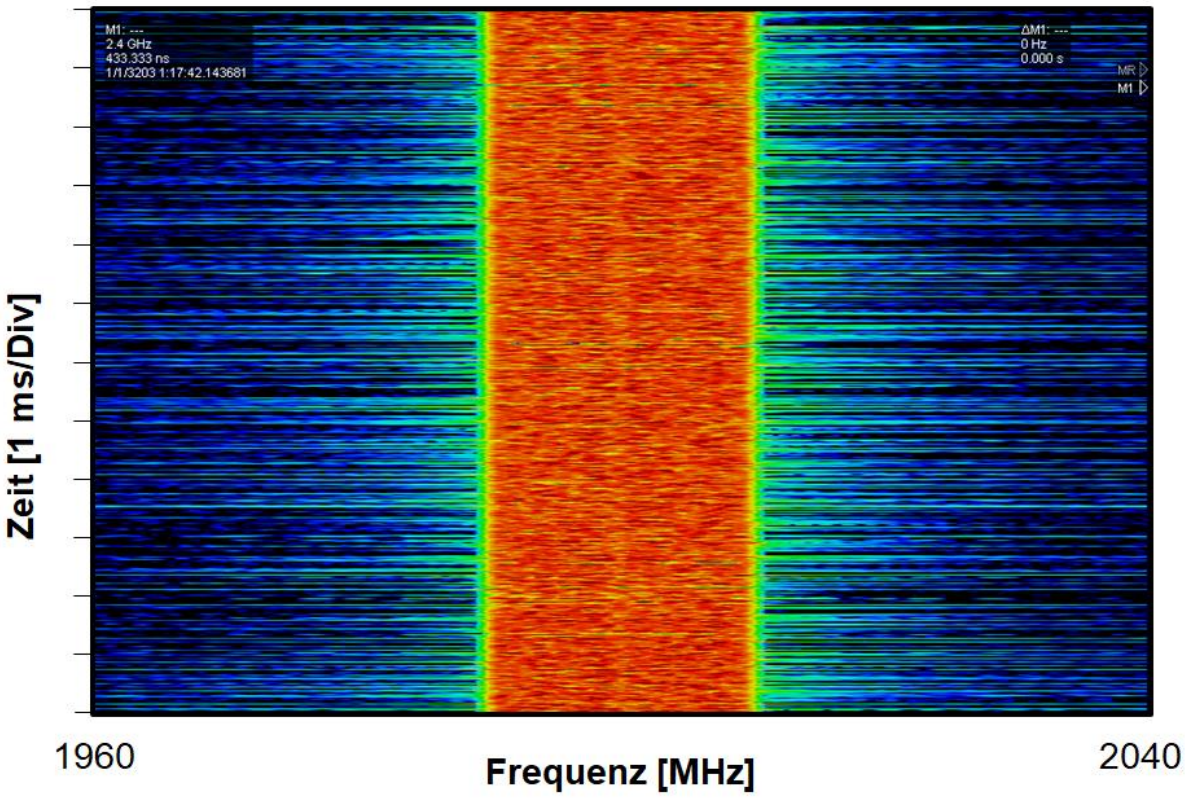

Abbildung 37: ohne Filter

Eine Möglichkeit, die Filterung trotz des erheblichen Aufwandes durchzuführen, ist die Implementierung des einzusetzenden FIR-Filters als Funktion auf dem FPGA. Auf diese Weise bleibt bei der Erzeugung der IQ-Werte für den ausführenden PC nur der Berechnungsaufwand durch die OFDM-Modulation. Das abgestrahlte OFDM-Signal in Abbildung 38 so nahezu durchgängig zu sehen, hierbei wurde die in einer Erweiterung von GnuRadio enthaltene RFNoC FIR-Filter Funktion verwendet.

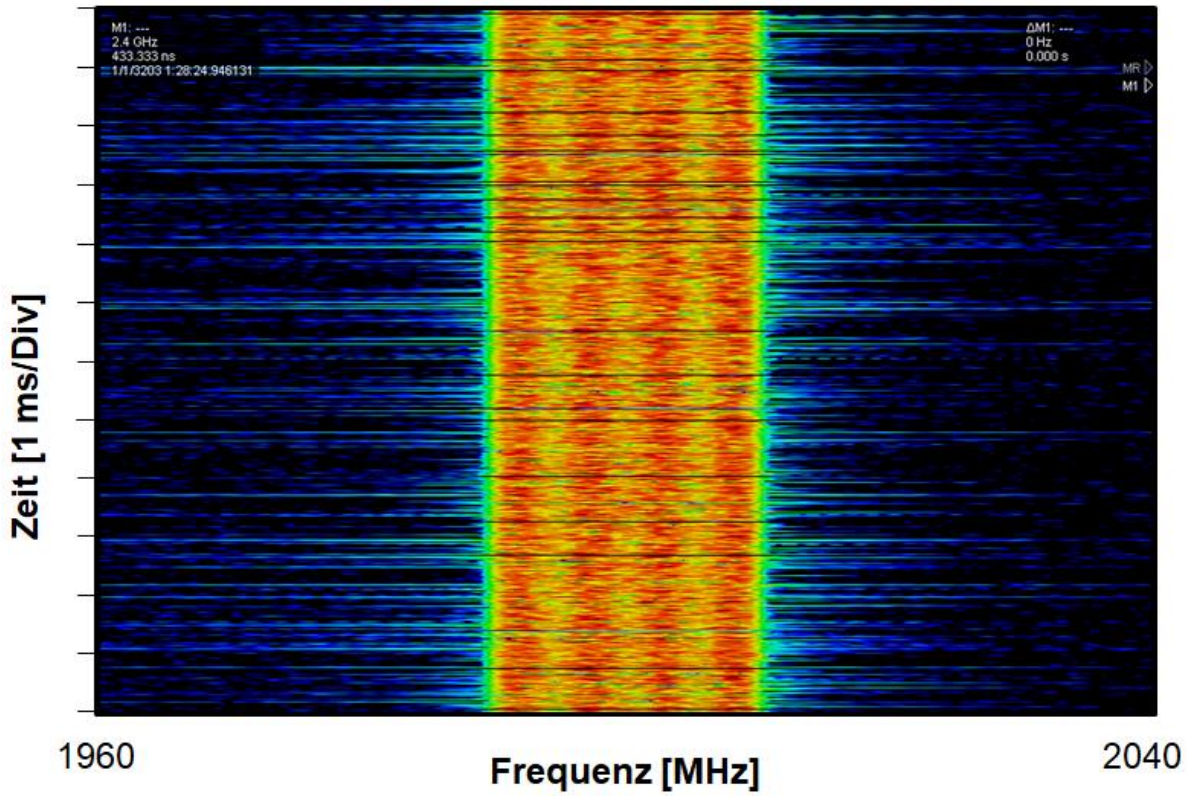

Abbildung 38: Filterung durch FPGA

Eine weitere Möglichkeit, die Filterung ohne Beeinträchtigung des abzustrahlenden Signals durchzuführen, ist die separate Bearbeitung des Signals als binäre Datei in GnuRadio. Hierzu wird die vorhandene Datei in GnuRadio abgespielt, gefiltert und wieder aufgenommen (siehe Abbildung 39).

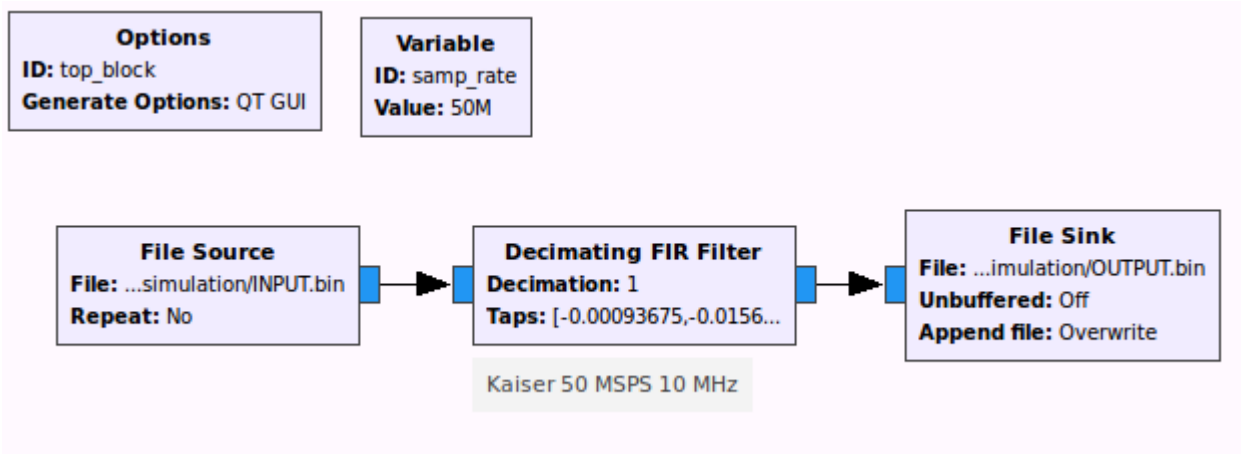

Abbildung 39: Filterung eines gespeicherten Interferenzprofiles

Das daraus resultierende Signal wurde in Abbildung 40 in einer sich wiederholenden Schleife aus einer File Source abgespielt. Der durch das Auslesen von IQ-Werten entstehende Aufwand ist gering genug, dass das Signal unterbrechungsfrei abgestrahlt

werden konnte. Die Messung der von GnuRadio bereitgestellten IQ-Werte ergab eine Taktrate von durchschnittlich 965 Millionen IQ-Werten pro Sekunde.

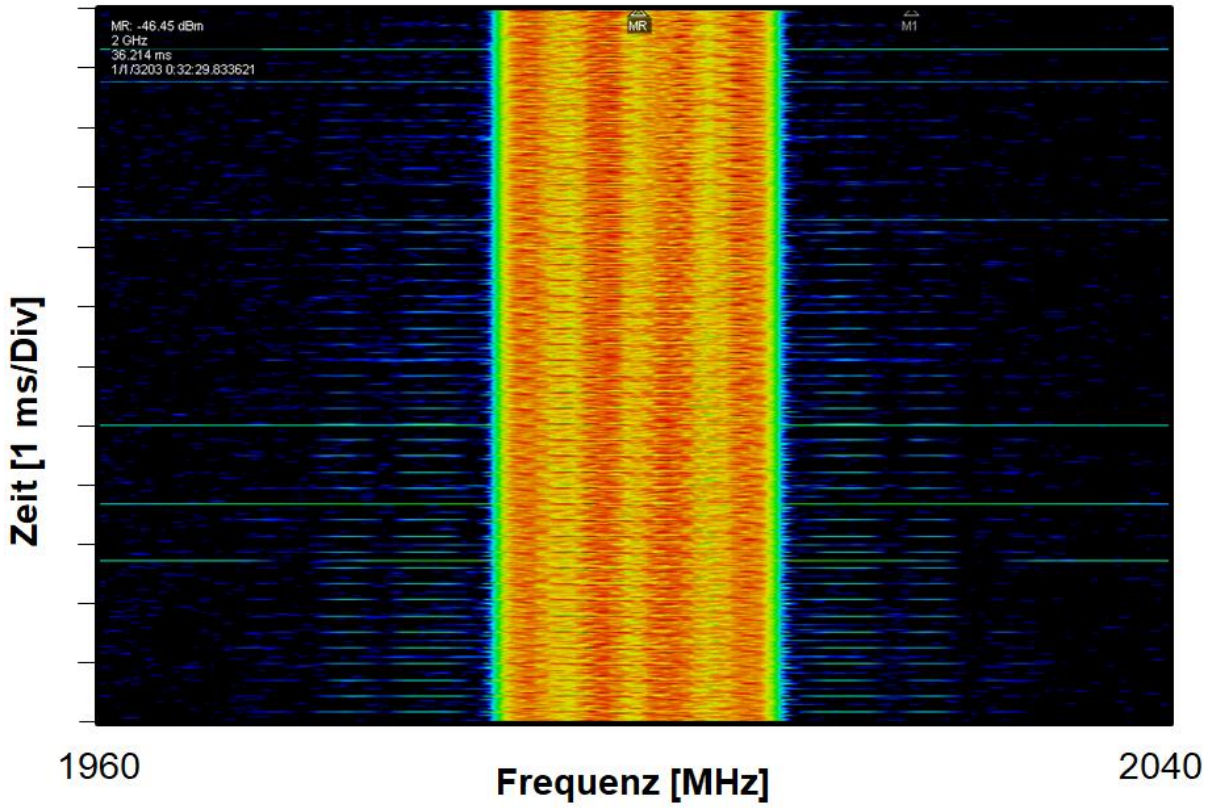

Abbildung 40: Vorbearbeitung durch Software

Die genutzte PCIe-Verbindung zwischen PC und USRP erlaubt die Übertragung von 200 MSPS, allerdings stellt die Verarbeitung in GnuRadio einen Engpass dar, der die Nutzung dieser Abtastrate einschränkt.

Um die Abhängigkeit der Unterbrechungen von der verwendeten Signalerzeugung, -bearbeitung und geforderten Abtastrate darzustellen, wurde mit dem RSA die abgestrahlte Leistung gegenüber der Zeit innerhalb eines 280 ms großen Intervalls aufgenommen. Die zeitliche Auflösung beträgt 2,8 µs, die in Abbildung 42, Abbildung 43, Abbildung 44, Abbildung 46 und Abbildung 47 dargestellten Ausschnitte zeigen einen 10 ms langen Ausschnitt der bei 2 GHz gemessenen Leistung.

In GnuRadio wurde die in Abbildung 29 gezeigte Konfiguration zur Messung verschiedener Filter wiederverwendet. Eine Änderung der Abtastrate führte zu einer Neuberechnung des verwendeten Filters. Die Filterung erfolgte auf dem PC in GnuRadio und wurde parallel zum Sendebetrieb durchgeführt.

In den Messdaten wurden die vollständig erfassten Sende- bzw. Unterbrechungsintervalle ausgewertet und ein Mittelwert über alle ausgewerteten Intervalle berechnet. Messwerte, die höchstens 10 dB unter der durchschnittlichen Sendeleistung liegen, werden hierbei als

vom USRP abgestrahlte Leistung interpretiert. Der Sendeanteil stellt den Zeitanteil des Sendebetriebs, gemessen an der abgestrahlten Leistung, zum gesamten Zeitintervall der Messung (siehe Tabelle 8) dar.

Tabelle 8: Zeitliche Abstände OFDM

Abtastrate	50 MSPS	100 MSPS	200 MSPS
Durchschnittliche Sendezeit	80,7 µs	39,5 µs	27 µs
Durchschnittliche Unterbrechungszeit	259,1 µs	615,4 µs	438,5 µs
Sendeanteil	23,8%	6 %	5,8 %

Zur Erzeugung eines Referenzwertes für die Taktrate steht in GnuRadio eine Rate Probe Funktion zur Verfügung, die die IQ-Datenrate an einer bestimmten Stelle in der Ausgabekonsole ausgibt. Die in Abbildung 41 dargestellte Konfiguration in GnuRadio zeigt die Messung der vom PC bereitgestellten IQ-Werte bei Erzeugung eines gefilterten OFDM Signals. Die Ausgabe wurde in einem Intervall von 500 ms aktualisiert, es konnten im Durchschnitt 26,5 MSPS bereitgestellt werden.

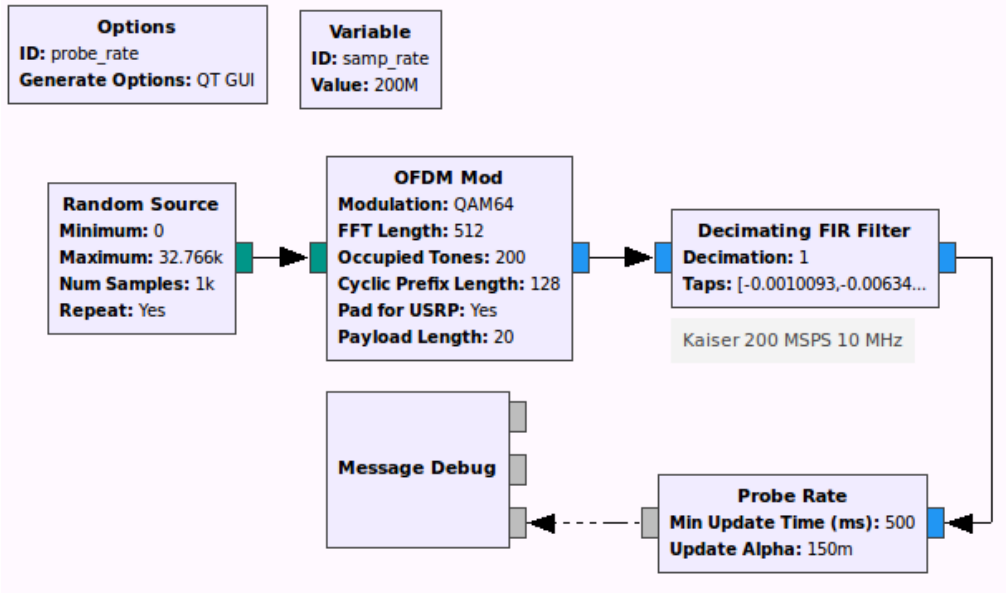

Abbildung 41: Taktrate der OFDM Signalerzeugung

Bei einem OFDM-Signal mit einer Taktrate von 50 MSPS zeigen sich deutliche, zyklisch auftretende Unterbrechungen. In Abbildung 42 ist ein 10 ms langer Ausschnitt der Messung zu sehen. Idealerweise wäre eine nahezu konstante Leistung bei -25 dBm zu sehen, stattdessen wird der Sendebetrieb bei fehlenden IQ-Daten bis zur folgenden Übertragung eingestellt.

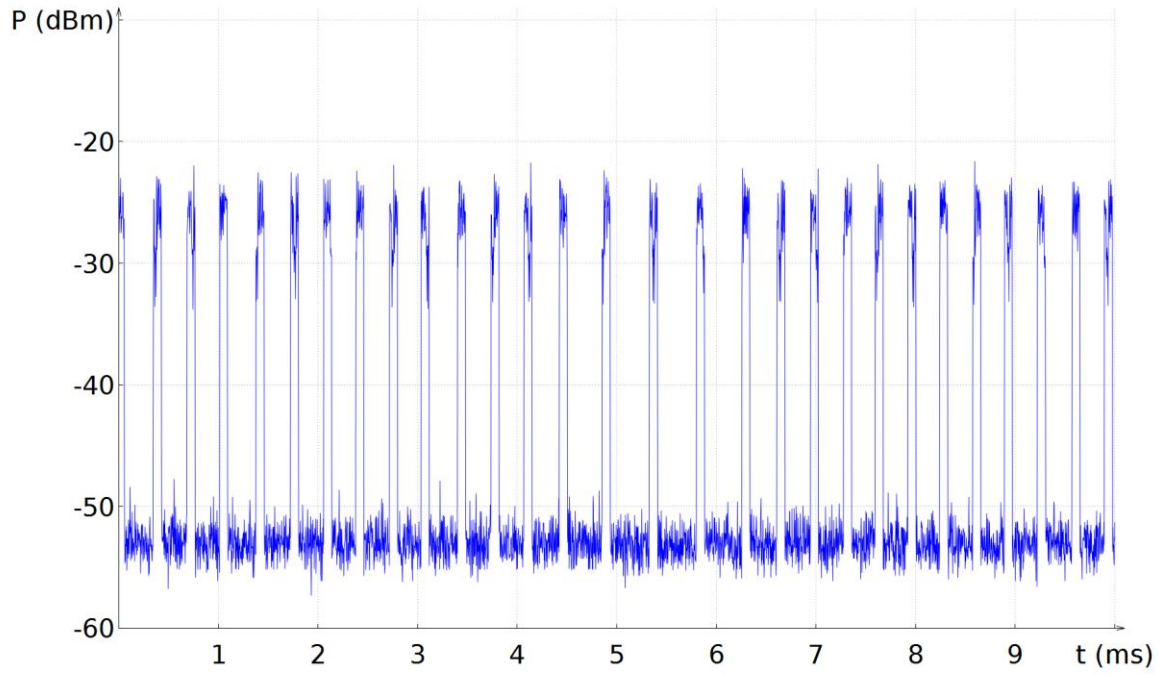

Abbildung 42: OFDM Signal bei 50 MSamples

Wird das OFDM Signal mit einer noch höheren Taktrate erzeugt, so nimmt der Sendeanteil weiter ab. In Abbildung 43 ist die gesendete Leistung bei 100 MSPS mit einem Sendeanteil von 6 % zu sehen, in Abbildung 44 ist die gesendete Leistung bei 200 MSPS mit einem Sendeanteil von 5,8 % dargestellt.

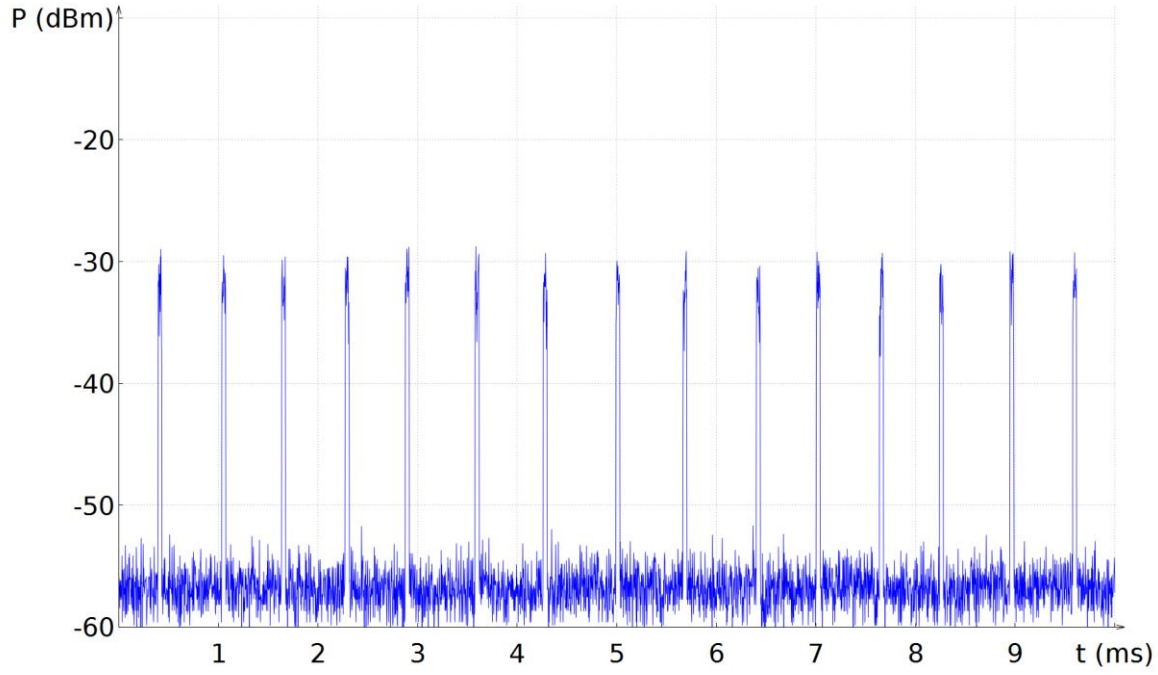

Abbildung 43: OFDM-Signal bei 100 MSamples

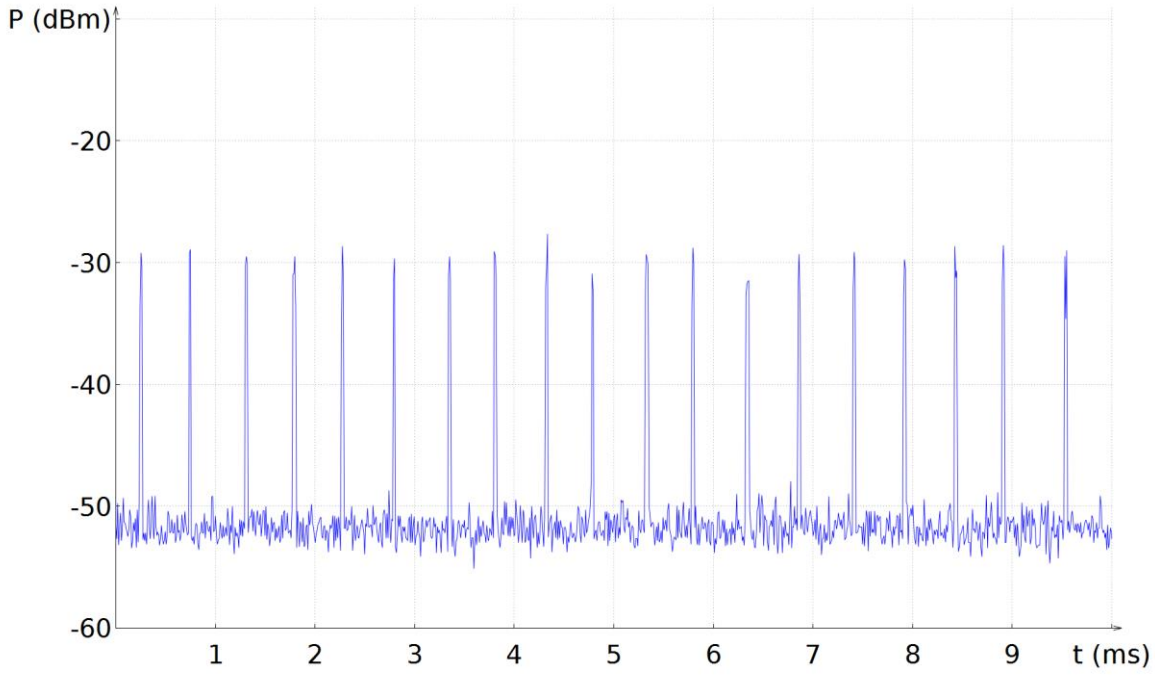

Abbildung 44: OFDM-Signal bei 200 MSamples

Unter der Annahme, dass der Berechnungsaufwand pro IQ-Wert und die Rechenleistung des verwendeten PC unverändert bleiben, wäre ein zur Abtastrate entgegengesetzt proportionaler Sendeanteil zu erwarten gewesen. Dieser Zusammenhang trifft auf die Messungen mit 50 MSPS und 200 MSPS zu, die Messung mit 100 MSPS pro Sekunde fällt hierbei mit einem nahezu identischen Sendeanteil wie bei der doppelten Abtastrate aus der Reihe. Grund dafür kann ein vom PC parallel ausgeführter Hintergrundprozess sein, welcher die für die Berechnung der Abtastwerte zur Verfügung stehende Rechenkapazität reduziert hat.

Um das Sendeverhalten bei einem deutlich reduzierten Berechnungsaufwand für die einzelnen IQ-Werte zu erfassen, wurde in GnuRadio ein einzelnes Sinussignal mit konstanter Amplitude erzeugt. Die Konfiguration zur Erzeugung des Signals ist in Abbildung 45 dargestellt.

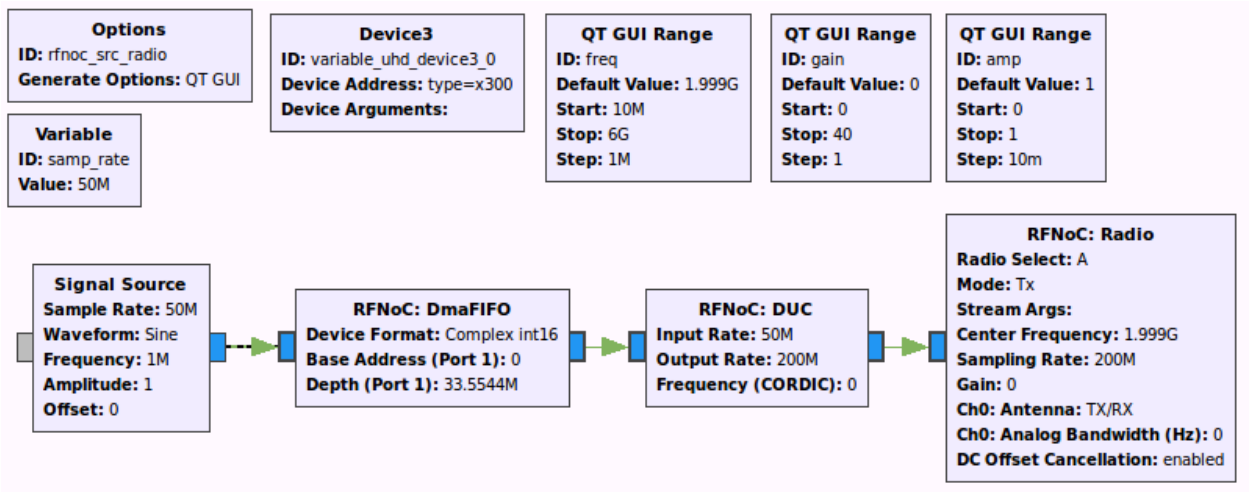

Abbildung 45: Generierung Sinussignal in GnuRadio

Die Länge des aufgenommenen Intervalls wurde auf 30 ms reduziert, um kurze Einbrüche der abgestrahlten Leistung erfassen zu können. Die resultierende zeitliche Auflösung beträgt 320 ns, das in Abbildung 46 und Abbildung 47 dargestellte Zeitintervall ist 300 µs lang. Alle weiteren Einstellungen des Echtzeitspektrumanalysators wurden von den Messungen des OFDM Signals übernommen. Tabelle 9 enthält die Zusammenfassung der Messergebnisse.

Tabelle 9: Zeitliche Abstände Einzelträger

Abtastrate	50 MSPS	200 MSPS
Durchschnittliche Sendezeit	79,37 µs	16,14 µs
Durchschnittliche Unterbrechungszeit	0,45 µs	3 µs
Sendeanteil	99 %	84,4 %

Entsprechend der in Abbildung 41 dargestellten Taktratenmessung bei der Erzeugung des gefilterten OFDM Signals wurde die Taktrate des Sinussignals gemessen. Es konnten durchschnittlich 109,5 MSPS bereitgestellt werden.

Das mit einer Taktrate von 50 MSPS erzeugte Sinussignal wurde, abgesehen von einigen zyklisch auftretenden Unterbrechungen, nahezu durchgängig von der USRP Plattform abgestrahlt (siehe Abbildung 46). Die errechnete durchschnittliche Unterbrechungszeit von 450 ns stellt bei einer Auflösung von 320 ns einen sehr ungenauen Wert dar.

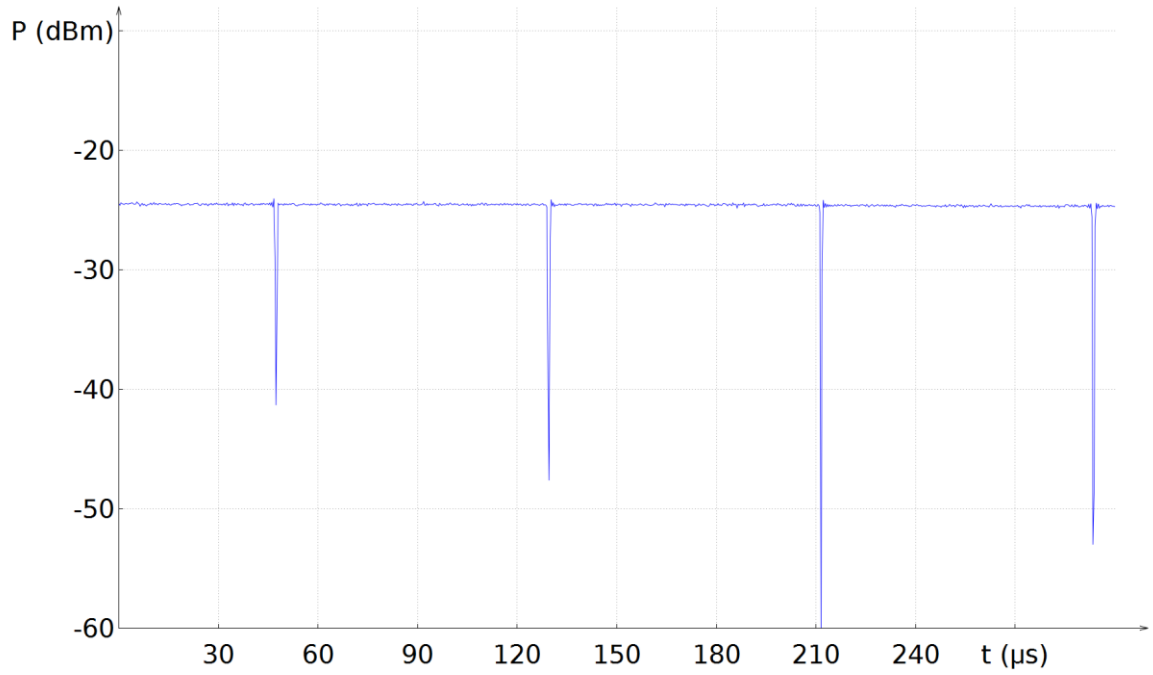

Abbildung 46: Sinus bei 50 MSamples

Bei der Erzeugung des Sinussignals mit einer Taktrate von 200 MSPS sinkt der Sendeanteil auf 84,4 %, die Unterbrechungen treten in deutlich kürzeren Intervallen als bei der Messung mit 50 MSPS auf. Die vom PC verwendete Näherungslösung zur Berechnung des Sinussignals erfordert bereits eine ausreichende Menge an Takten, um bei entsprechender Taktrate Unterbrechungen beim Sendebetrieb der USRP-Plattform herbeizuführen (siehe Abbildung 47).

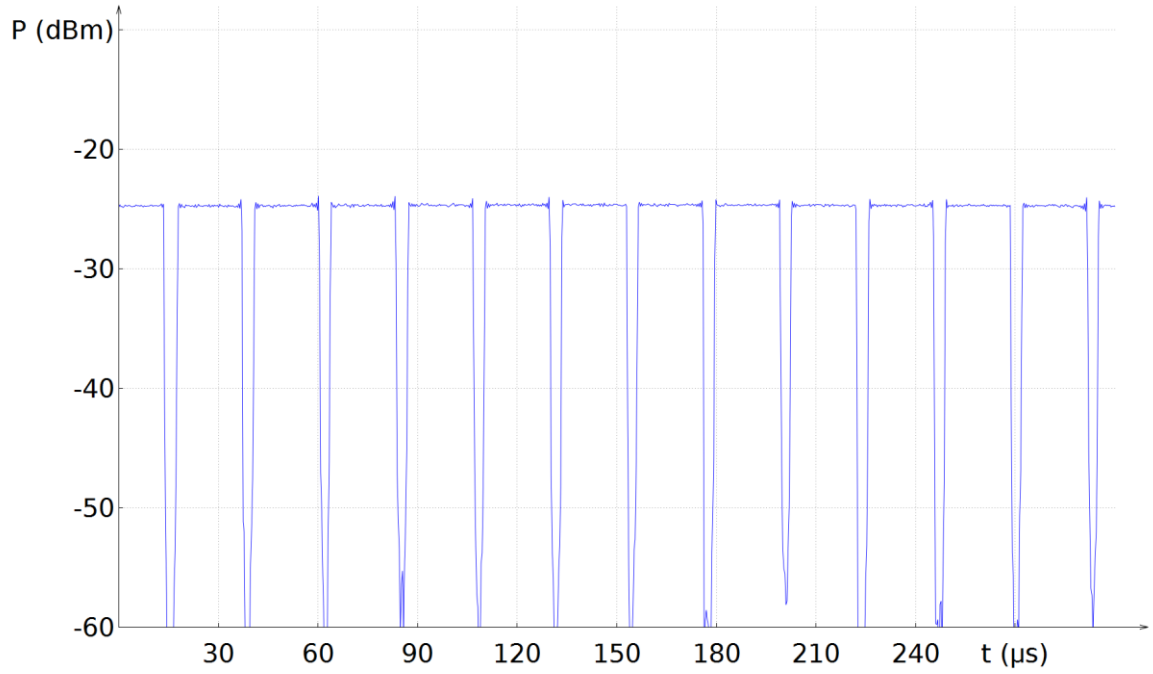

Abbildung 47: Sinus bei 200 MSamples

6.4 Oberwellen

Die Verarbeitung von Signalen durch die UBX-160 Erweiterungskarten ist von der eingestellten Trägerfrequenz abhängig. Trägerfrequenzen $f_N \geq 500\,MHz$ führen zu einer Verarbeitung im Direct Conversion Path und $f_N < 500\,MHz$ zu einer Verarbeitung im Down Conversion Path (vgl. Abbildung 48). Im folgenden Abschnitt werden die abgestrahlten Signale jeweils durch einen der Pfade verarbeitet und durch einen über ein Koaxialkabel angeschlossenen Spektrumanalysator in dem für die Erweiterungskarten zugelassenen Frequenzbereich von 10 MHz bis 6 GHz erfasst. Die RBW des Spektrumanalysators beträgt 20 kHz.

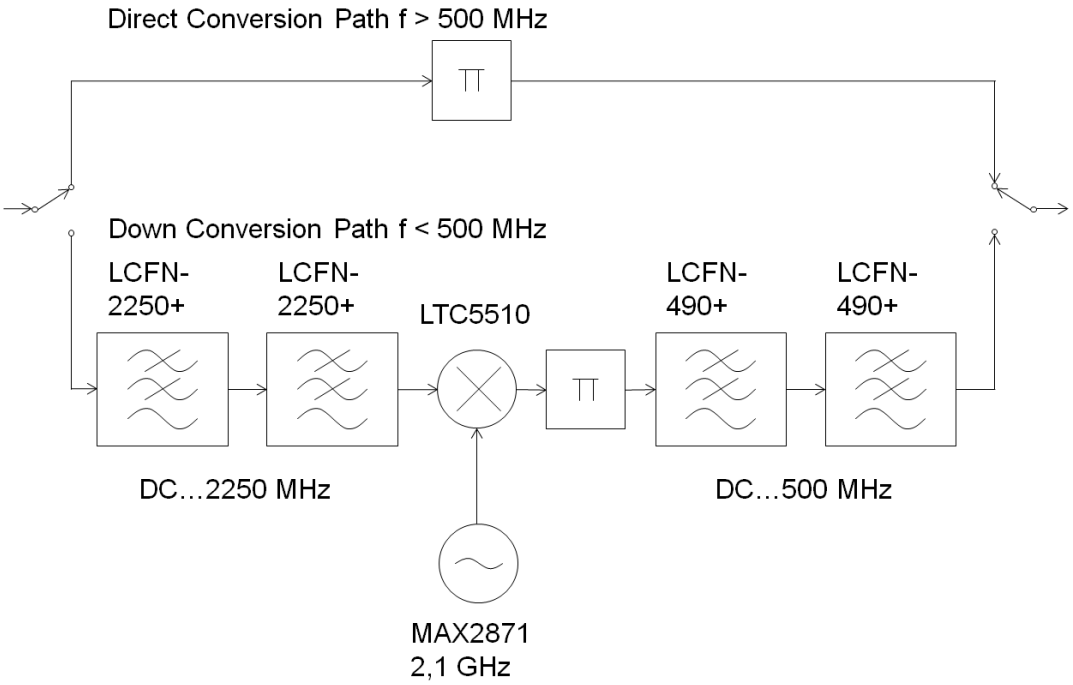

Abbildung 48: Darstellung Transmitter Conversion Paths

Abbildung 49 zeigt die Erzeugung eines Sinussignals, das von der Erweiterungskarte auf $f_N = 600\,MHz$ hochgemischt und durch den Direct Conversion Path verarbeitet wurde. Bei Vielfachen der Trägerfrequenz sind Abbilder des abgestrahlten Signals zu sehen. Auf der Erweiterungskarte findet nach der Verarbeitung im Direct- bzw. Down Conversion Path keine weitere Filterung statt.

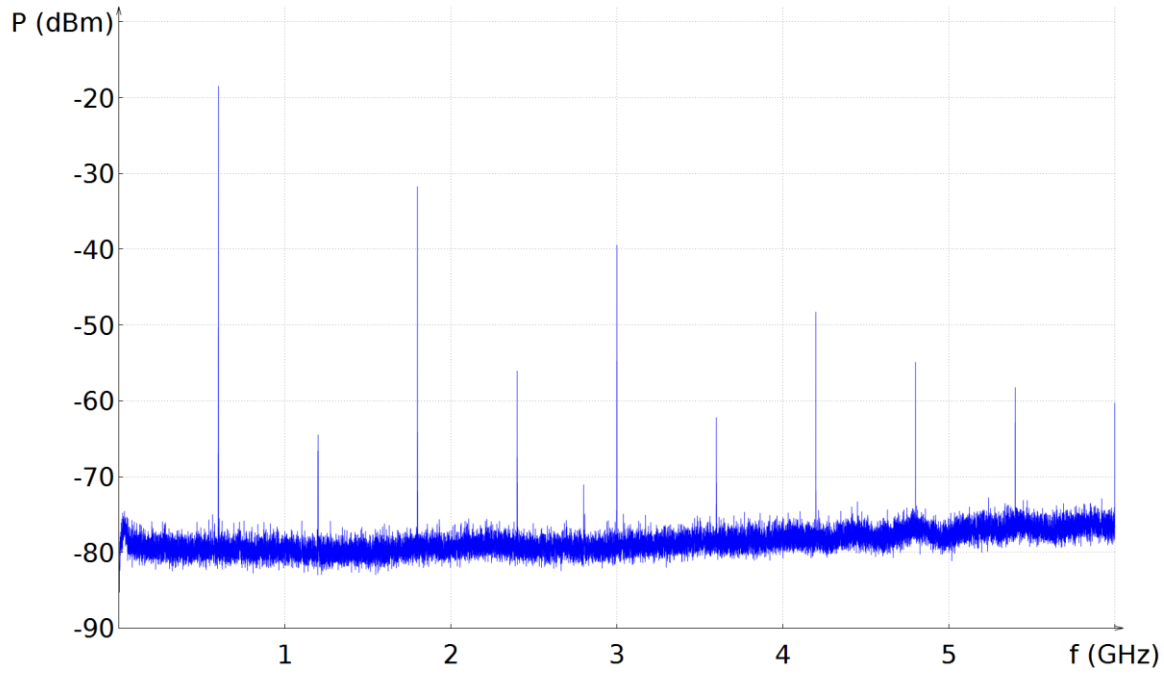

Abbildung 49: Trägerfrequenz bei 600 MHz

Durch die Erzeugung eines OFDM-Signals anstelle des Sinussignals lassen sich bei einzelnen Frequenzen Abbilder des originalen OFDM Signals bei 600 MHz erkennen.

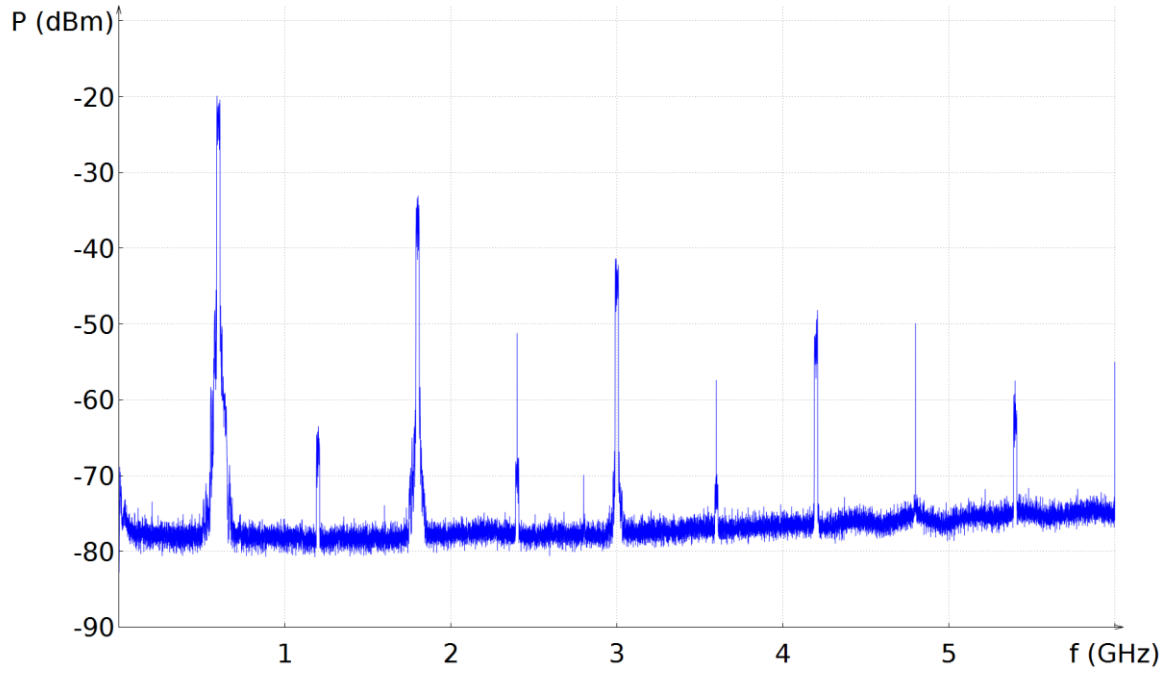

Abbildung 50: OFDM Signal bei 600 MHz

Das in Abbildung 51 dargestellte Sinussignal wurde von der Erweiterungskarte auf $f_N = 400\,MHz$ hochgemischt und durch den Down Conversion Path verarbeitet. Das Basisbandsignal wird hierbei auf 2,1 GHz gemischt, nach einer Tiefpassfilterung auf die

Nutzfrequenz f_N heruntergemischt und nochmals tiefpassgefiltert. Oberhalb der vorgesehenen Nutzfrequenz sind keine weiteren Leistungsanteile sichtbar, die vom Nutzsignal abhängen. Bei 2,1 GHz ist ein um 45 dB gedämpftes Abbild des Signals bei 400 MHz zu sehen, dessen Frequenz sich bei Veränderung der Trägerfrequenz im Bereich $f_N \in [10\,MHz, 500\,MHZ)$ nicht verschiebt.

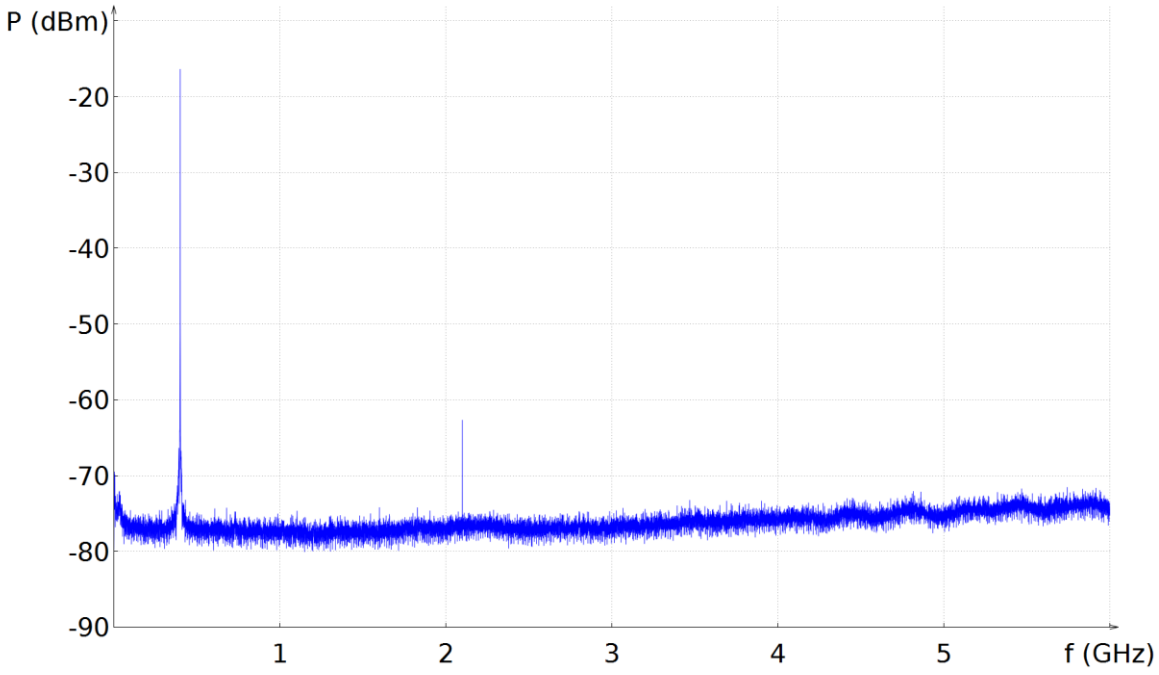

Abbildung 51: Trägerfrequenz bei 400 MHz

In Abbildung 52 wurde anstelle des Sinussignals ein OFDM-Signal mit $f_N = 400\,MHz$ erzeugt, das Signal ist als Abbild bei 2,1 GHz wiederzuerkennnen. Da das Abbild in dieser Form nur nach dem ersten Lokaloszillator bei Verwendung des Down Conversion Paths vorkommt, ist eine Übertragung durch den ungenutzten Direct Conversion Path denkbar.

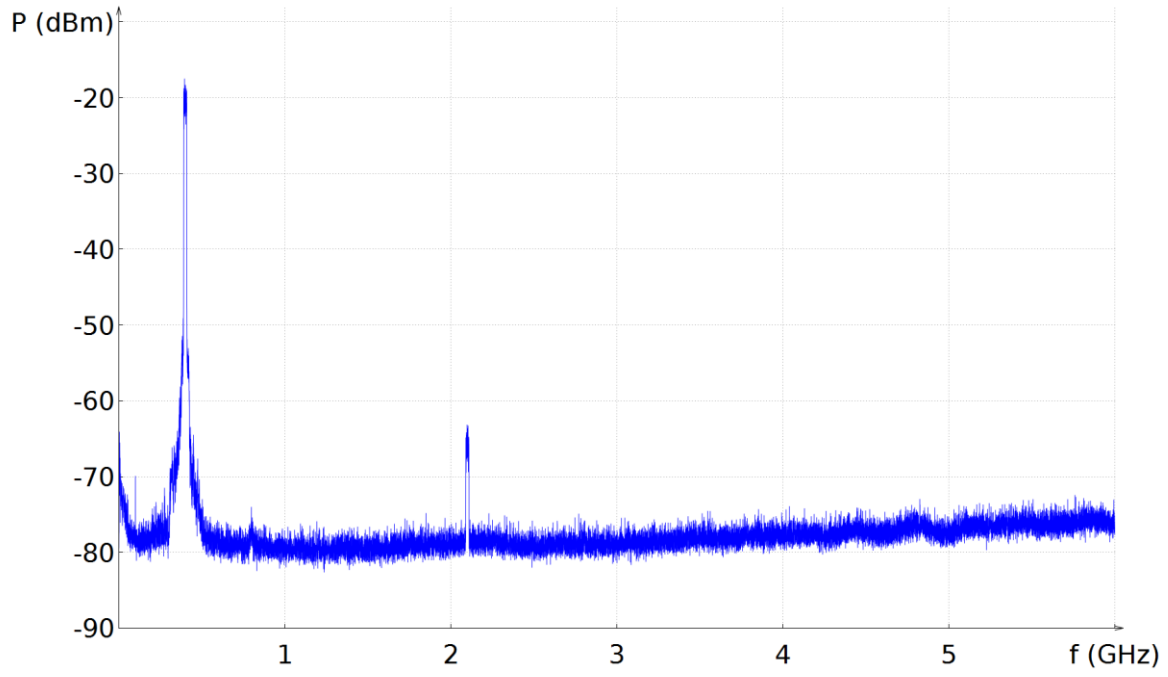

Abbildung 52: OFDM-Signal bei 400 MHZ

6.5 Trägersignal

Eine Eigenschaft der verwendeten USRP-Plattform im Sendebetrieb ist der deutlich sichtbare Träger bei f_N. In Abbildung 54 ist ein QAM64 OFDM Signal zu sehen, welches im ersten Schritt vom VSG bei einer Frequenz von 2 GHz erzeugt und von der USRP-Plattform aufgenommenen wurde. Im zweiten Schritt wurde das Signal mit der in Abbildung 53 gezeigten Konfiguration softwareseitig auf eine 20 MHz höhere Frequenz gemischt und im folgenden dritten Schritt an den über ein Koaxialkabel angeschlossenen Echtzeitspektrumanalysator gesendet.

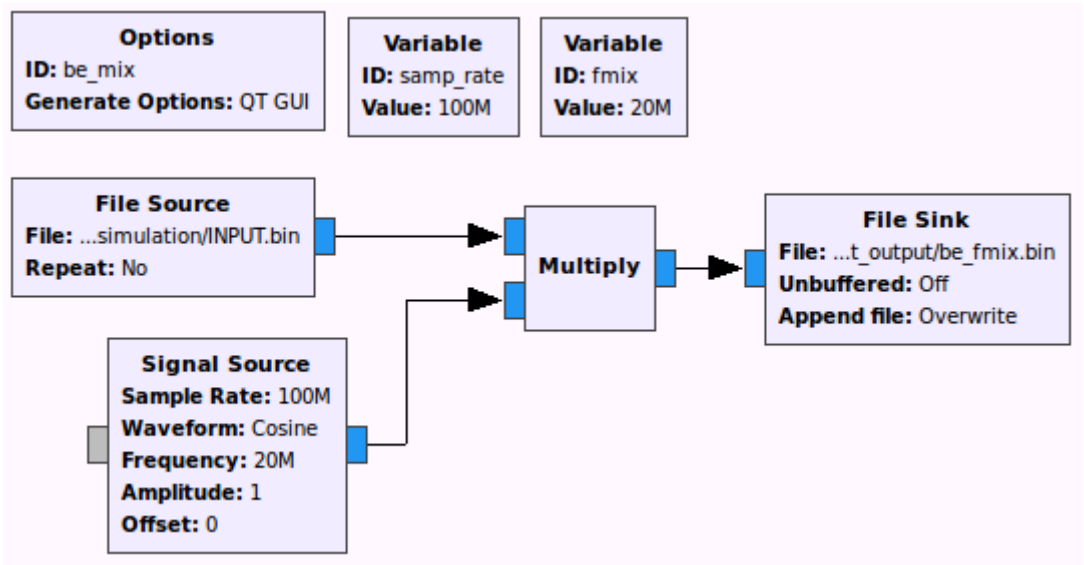

Abbildung 53: Mischen eines vorhandenen Signals

Das im ersten Schritt aufgenommene und verschobene Signal wurde ohne Verzerrungen in Zeit, Frequenz und Amplitude wiedergegeben. Somit ist eine Wiedergabe sowohl erfasster als auch erzeugter Signale grundsätzlich möglich. Allerdings stellt der abgestrahlte Träger mit einer Leistung von -45 dBm eine Teilung der ansonsten nutzbaren 160 MHz Bandbreite in zwei 80 MHz Bänder dar, zudem ist im Sendebetrieb unabhängig von der Lage des Nutzsignals auch das Trägersignal vorhanden.

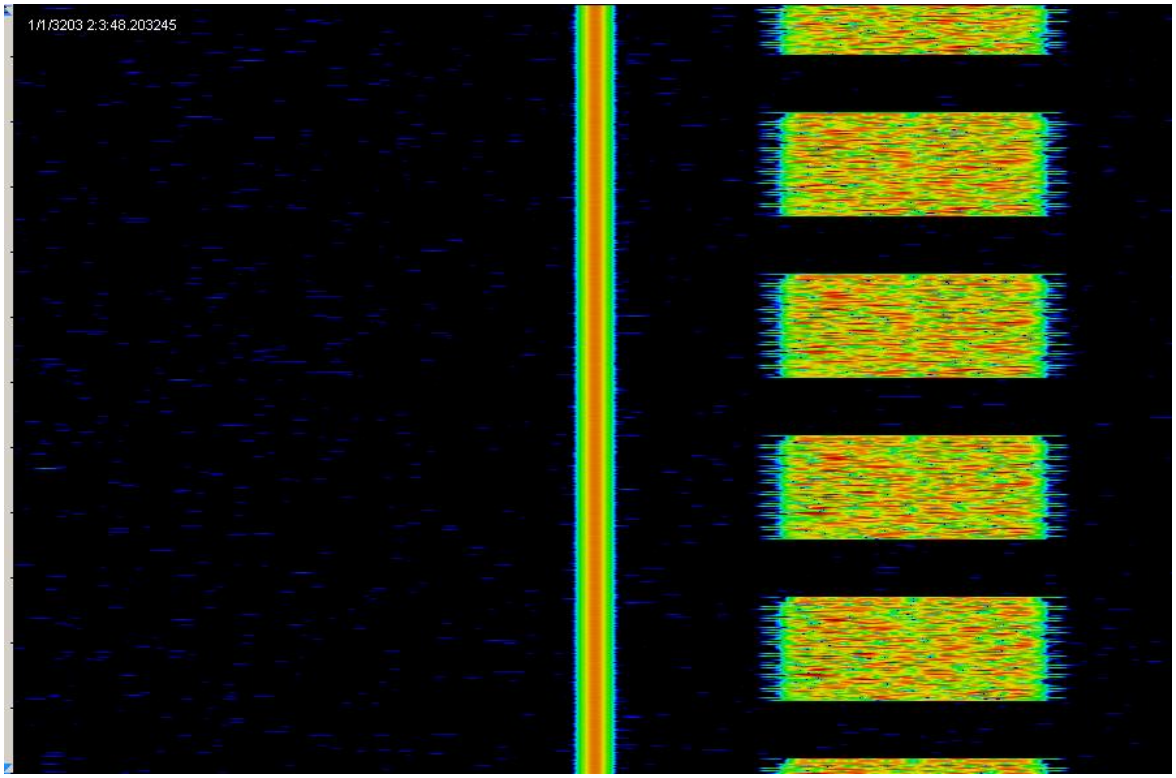

Abbildung 54: Aufgezeichnetes OFDM Signal mit Träger

7 Zusammenfassung

In dieser Arbeit wurde die Eignung der Software Defined Radio Plattform USRP X310 mit zwei UBX-160 Erweiterungskarten als rekonfigurierbares Funksystem für die Ausführung von Interferenzprofilen untersucht, die Signalverarbeitung erfolgte hierbei mit GnuRadio. Als vergleichbares Gerät zur Signalerzeugung wurde ein Vektorsignalgenerator verwendet. Im Rahmen der Konzepte und Implementierung wurde eine mögliche Verwendung von Medienzugriffsverfahren in zu betreibenden Interferenzprofilen vorgestellt. Zudem wurde die USRP Plattform im Sendebetrieb validiert und mit dem Vektorsignalgenerator verglichen.

Den recht genauen, zweckgebundenen Hardwarekomponenten des Vektorsignalgenerators steht die Rekonfigurierbarkeit des Software Defined Radio gegenüber, welches für einen deutlich breiteren Anwendungsbereich ausgelegt ist. Der Betrieb von Softwarekomponenten, die durch Berechnung von IQ-Werten in Echtzeit ihre Gegenstücke in

Hardware ersetzen, erfordert eine erhebliche Rechenleistung. Die in dieser Arbeit verwendete USRP Plattform stellt mit einem FPGA eine Möglichkeit zum Auslagern bereits vorgefertigter Funktionen zur Verfügung, zudem kann Rechenleistung anwendungsspezifisch eingespart werden. Sobald die erforderte Bandbreite die Rechenleistung des Rechners übersteigt, entstehen deutliche Lücken im Sendebetrieb, welche die durch das verwendete Interferenzprofil beabsichtigte Medienbelegung verfälschen.

8 Ausblick

Die Einsatzmöglichkeiten der durch Software Defined Radios ausgeführten Interferenzprofile beschränken sich nicht auf die in dieser Arbeit aufgezeigten Konzepte. Durch die Realisierung vollständiger Funksysteme als Interferenzprofile mit Medienzugriffsverfahren und die Verwendung mehrerer USRP Plattformen sind zum Beispiel Untersuchungen verschiedener Kenngrößen in Übertragungskanälen oder das Verhalten von Funksystemen in heterogenen Netzen möglich. Die Verwendung der USRP Plattformen beschränkt sich hierbei nicht auf die Nutzung in einer Laborumgebung, sondern kann unter Berücksichtigung entsprechender Vorkehrungen außerhalb von Gebäuden und in mobilen Anwendungen eingesetzt werden.

Zur Kompensierung der fehlenden Rechenleistung können Funktionen wie zum Beispiel die Filterung durch einen FIR-Filter mithilfe eines vorhandenen Grafikprozessors beschleunigt werden. Zudem werden kontinuierlich vorgefertigte, auf dem FPGA implementierbare Funktionen im Rahmen von RFNoC zur entsprechenden Erweiterung von GnuRadio hinzugefügt. Neben GnuRadio können weitere, gegebenenfalls proprietäre Softwarelösungen in Betracht gezogen werden.

Im Rahmen späterer Arbeiten können die in GnuRadio und VHDL erstellbaren Softwarekomponenten von Interferenzprofilen entwickelt werden. Insbesondere die auf dem FPGA implementierbaren Medienzugriffsverfahren stellen eine wichtige Komponente zum Betrieb eines Interferenzprofiles dar.

Um die USRP Plattform effektiver nutzen zu können, sollten gegebenenfalls Wege zur Trägerunterdrückung und zur Anpassung des gesendeten Signals berücksichtigt werden.

Zur Eingliederung in einen Messaufbau zur Messung von Übertragungseigenschaften sind Schnittstellen zu weiteren Komponenten sinnvoll, um beispielsweise die Konfiguration durch eine zentrale Steuerung und die Erzeugung beziehungsweise Verarbeitung von IQ-Daten durch ein dafür vorgesehenes System zu realisieren.

9 Referenzen

[1] UBX 10-6000 MHz Rx/Tx: https://www.ettus.com/product/details/UBX160 (Stand 10.04.2017)

[2] Friedrich K. Jondral: Software-Defined Radio: Basics and Evolution to Cognitive Radio, EURASIP Journal on Wireless Communications and Networking 2005:3, 275—283, 04.2005

[3] J Mitola, GQ Maguire: “Cognitive Radio: Making software radios more personal”, IEEE personal communications, 1999

[4] Datenblatt USRP X310 https://www.ettus.com/content/files/X300_X310_Spec_Sheet.pdf (Stand 12.04.2017)

[5] Dennis M. Akos, Michael Stockmaster, James B. Y. Tsui, Joe Caschera: Direct Bandpass Sampling of Multiple Distinct RF Signals, IEEE transactions on communications, Vol. 47, No 7, July 1999

[6] Behzad Razavi: Design Considerations for Direct-Conversion Receivers, IEEE Transactions on circuits and systems- II: Analog and digital signal processing, Vol. 44, No 6, June 1997

[7] What is GNU Radio? https://wiki.gnuradio.org/index.php/What_is_GNU_Radio%3F (Stand 19.04.2017)

[8] Guided Tutorial GNU Radio in C++: https://wiki.gnuradio.org/index.php/Guided_Tutorial_GNU_Radio_in_C%2B%2B (Stand 19.04.2017)

[9] X300/X310 https://kb.ettus.com/X300/X310#Choosing_USRP_X310_vs_USRP_X300 (Stand 24.04.2017)

[10] Aquila https://github.com/zsiciarz/aquila (Stand 25.04.2017)

[11] Radio Transport Protocols: https://files.ettus.com/manual/page_rtp.html (Stand 25.10.2017)

[12] Device configuration through address string, http://files.ettus.com/manual/page_configuration.html#config_devaddr (Stand 02.05.2017)

[13] A.Gnad, M.Krätzig, J.Schade, R.Schönrock, S. Trikaliotis, Dr. L.Rauchhaupt: „Software Defined radio und Cognitive Radio in der industriellen Automation“, DFAM Studie Nr.7/2011

[14] Todor Cooklev, Robert Normoyle, David Clendenen: “The VITA 49 Analog RF-Digital Interface”, IEEE CIRCUITS AND SYSTEMS MAGAZINE, 29.11.2012

[15] Rolf Rannacher: Einführung in die Numerische Mathematik, Institut für angewandte Mathematik, Universität Heidelberg, 27.4.2016

[16] Vorgang Upsampling um den Faktor L=3 mit beispielhaften Signalverlauf: https://commons.wikimedia.org/wiki/File:Upsampling_Example.svg

[17] P. Vaidyanathan: Multirate Systems and Filter Banks, Prentice-Hall, 1992. ISBN 0-13605-718-7

[18] Lawrence Landweber,Jun Murai: http://www.soi.wide.ad.jp/class/99007/slides/09/23.html, Introduction to Computer Networks, SOI WIDE University

[19] RFNoC, https://kb.ettus.com/RFNoC , (Stand 30.06.2017)

[20] Getting Started with RFNoC Development, https://kb.ettus.com/Getting_Started_with_RFNoC_Development (Stand 24.07.2017)

[21] RFNoC: RF Network on Chip, https://www.ettus.com/content/files/RFNoC_Wireless_at_VT_Intro.pdf, (Stand 25.07.2017)

[22] Hermann Pommer: Roaming zwischen Wireless Local Area Networks, 2008, ISBN 9783836487085, S.112

[23] Adaptive Frequency Hopping for Reduced Interference between Bluetooth and Wireless LAN, https://www.design-reuse.com/articles/5715/adaptive-frequency-hopping-for-reduced-interference-between-bluetooth-and-wireless-lan.html (Stand 20.07.2017)

[24] Bluetooth Adaptive Frequency Hopping on an R&S CMW, https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1c108/1C108_0e_Bluetooth_BR_EDR_AFH.pdf , (Stand 20.07.2017)

[25] Patrick Schnabel: Netzwerktechnik-Fibel, ISBN 978-3833416811, 4. Auflage

[26] Down to the TPL: How PCI express devices talk: http://xillybus.com/tutorials/pci-express-tlp-pcie-primer-tutorial-guide-1 , (Stand 27.07.2017)

[27] PCI Express 2.0 Base Specification, https://de.scribd.com/document/195991550/Pci-Express-Base-Specification-v2-0, (Stand 07.2017)

[28] IEEE Standard for Local and metropolitan area networks-Bridges and Bridged Networks 802-1Q-2014, http://standards.ieee.org/getieee802/download/802-1Q-2014.pdf, IEEE Computer Society

[29] Ian F. Akyildiz, Won-Yeol Lee, Mehmet C. Vuran, Shantidev Mohanty:
Next generation/dynamic spectrum access/cognitive Radio wireless networks: A survey, Journal Computer Networks Vol 50 Issue 13, 15.09.2006

[30] Jeffrey Hugh Reed: Software Radio: A Modern Approach to Radio Engineering, Prentice Hall Professional, 2002, ISBN 9780130811585

[31] Marcus Burton: 802.11 Arbitration, https://www.cwnp.com/wp-content/uploads/pdf/802.11_arbitration.pdf, 09.2009

[32] MATLAB Toolkit for R&S Signal Generators, https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1gp60/1GP60_9E.pdf
(Stand 21.08.2017)

[33] R&S WinIQSIM2 Signal Generation Software User Manual 1177.5533.02-02

[34] HiFlecs: https://www.ifak.eu/de/projekt/hiflecs-hochperformante-sichere-funktechnologien-deren-systemintegration-zuk%C3%BCnftige-industr (Stand 27.10.2017)

[35] KoMe: https://www.ifak.eu/de/content/kome-kognitive-mediumszugangsalgorithmen-f%C3%BCr-industrielle-funkanwendungen-startet-am-ifak (Stand 27.10.2017)

[36] ReiCovAir: https://www.ifak.eu/de/projekt/reicovair-reliable-industrial-communication-over-air (Stand 27.10.2017)