

Fachbereich Ingenieurwesen
Fachrichtung Elektro- und Informationstechnik

Masterarbeit

zur Erlangung des akademischen Grades

Master of Engineering

Aufbau und Inbetriebnahme eines Software Defined Radio am Zwischenfrequenzsystem des Radioteleskops Effelsberg zur Überwachung und Erkennung von Radio Frequency Interference

Verfasser: Björn Staufenbiel, B.Eng.
Unternehmen: Max-Planck-Institut für Radioastronomie
Betreuung: Herr Prof. Dr.-Ing. Uwe Gärtner
Herr Dr.-Ing. Reinhard Keller
Bearbeitungszeitraum: 15.09.2014 - 02.03.2015

Abstract

Der Einfluss von Radio Frequency Interference (RFI) auf radioastronomische Messungen am Radioteleskop Effelsberg ist ein zentraler Punkt in der Forschung des Instituts. Aufgezeigt wird in dieser Arbeit, wie mithilfe geeigneter Hardware und eigens entwickelter Software solche Störstrahler erkannt werden können, um Maßnahme zur Vorbeugung zu treffen. Speziell Störungen in den für die Radioastronomie geschützten Frequenzbändern werden so detektiert. Dazu wird ein FFT-Spektrometer des Instituts zur spektralen Erkennung von Störstrahlern genutzt. Mittels eines Software Defined Radio werden die so erkannten Störungen demoduliert und auf eine potenzielle Quelle geschlossen. Die Darstellung der Ergebnisse und die Steuerung des Software Defined Radio findet dabei über ein Webinterface statt.

Um eine geeignete Hardware zu finden, wurden verschiedene Software Defined Radios untersucht. Dazu zählen verschiedene DVB-T-Sticks und ein USRP von National Instruments. Diese wurden messtechnisch charakterisiert und mittels Software die Tauglichkeit für diese Anwendung erprobt. Zudem wurde ein FFT-Spektrometer, welches vom und für das Max-Planck-Institut entwickelt wurde, untersucht und zur spektralen Erkennung von Störstrahlern verwendet.

Die eingesetzten SDR-Interfaces haben vergleichbare Schwierigkeiten bereitet. So waren viele Eigenstörlinien im Spektrum zu finden und das Eigenrauschen war sehr hoch. Diese Punkte waren für die eindeutige Bestimmung von Kleinsignalanteilen im Spektrum von großem Nachteil. Die Ergebnisse mit dem FFT-Spektrometer waren dagegen als durchweg positiv zu bewerten. Neben der hohen Geschwindigkeit der Aufnahme von einer großen Signalbandbreite ist vor allem das geringe Eigenrauschen der entscheidende Vorteil. Das Spektrometer stellt jedoch nur das Betragsspektrum zur Verfügung, sodass eine Demodulation von Signalen mit diesem nicht möglich ist. Zur Signaldemodulation wurde aus diesem Grund ein DVB-T-Stick mit einem Elonics R820T-Tuner gewählt.

Während das Spektrometer rein für die spektrale Analyse der Signale verwendet wird, dient das Software Defined Radio ausschließlich zur Signaldemodulation. Der Zugang des Nutzers zu den so ermittelten Daten erfolgt über ein Webinterface. Dieses ermöglicht den einfachen und universellen Zugang zu den Daten von einer beliebigen Stelle aus. Über das Webinterface können Einstellungen am Software Defined Radio vorgenommen werden und detektierte Störung überwacht werden. Im Gesamten hat sich so ein praktisches Tool zur Detektion und Erkennung von Störern ergeben.

Inhaltsverzeichnis

Abbildungsverzeichnis	VI
Tabellenverzeichnis	XI
Listingverzeichnis	XI
1 Einleitung und Aufgabenstellung	1
2 Vorstellung des Max-Planck-Instituts für Radioastronomie	5
3 RFI und SDR - Begriffsdefinitionen und Grundlagen	7
3.1 Radio Frequency Interference - eine Definition	7
3.2 Definition des Begriffs Software Defined Radio	9
3.3 Allgemeine Aspekte zur Hard- und Software eines Software Defined Radio .	11
3.3.1 Idealisierter Aufbau eines SDR-Hardwareinterface	11
3.3.2 Parameter der technischen Komponenten eines Software Defined Radio	12
3.3.3 Praktische Anwendungen von Software Defined Radios	12
3.4 Übersicht über Softwareinterfaces für Software Defined Radios	13
4 Theoretische Grundlagen der spektralen Analyse	15
4.1 Einsatzgebiete der spektralen Analyse	15
4.2 Signaltheorie zur Berechnung eines Signalspektrums	15
4.3 IQ-Demodulation und Abtastung von Signalen	18
4.4 Anwendung der Theorie zur Berechnung breitbandiger Spektren	20
4.5 Berechnung der diskrete Fouriertransformation	24
5 Untersuchung von SDR-Hardwareinterfaces	27
5.1 Hardwareaufbau der verwendeten SDR-Interfaces	28
5.1.1 DVB-T-Sticks als SDR-Hardware	28
5.1.1.1 Spezifikationen der verwendeten DVB-T-Sticks	30
5.1.1.2 Aufbau und Funktionsweise des RF-Frontends	31
5.1.2 Hardwareaufbau des National Instruments USRP-2920	33
5.2 Bestimmung des Eingangsreflexionsfaktors	34
5.2.1 Theoretische Grundlagen zur Messmethode	34
5.2.2 Störemission und Eingangsreflexionsfaktor des E4000-Sticks	36
5.2.3 Störemission und Eingangsreflexionsfaktor des R820T-Sticks	39
5.2.4 Störemission und Eingangsreflexionsfaktors des USRP	41
5.3 Frequenzgang der Signalverarbeitungskette	46
5.4 Ermittlung des Dynamikumfangs	47
5.5 Bestimmung des Third-Order-Intercept-Punktes	48
5.5.1 Theoretische Grundlagen zur Bestimmungsmethode	48
5.5.2 Third-Order-Intercept-Punkt der RTL-SDRs	51
5.5.3 Third-Order-Intercept-Punkt des USRP-2920	54
5.6 Diskussion der ermittelten Messergebnisse	55
6 Einrichtung eines RTL-SDR-Servers	57

7	Entwicklung eines DFT-Spektrumanalysators mit einem SDR	61
7.1	Softwareentwicklung eines DFT-Spektrumanalysators	61
7.1.1	Ermittlung von breitbandigen Signalspektren mit einem RTL-SDR .	61
7.1.1.1	Erstellung eines GNU Radio-Flowgraphs zur Berechnung der DFT	61
7.1.1.2	Python-Softwareentwicklung eines breitbandigen Spektru- manalysators	65
7.1.1.3	Anpassung der Signalaufnahme	70
7.1.1.4	Verwendung einer zusätzlichen Zwischenfrequenz zur Fre- quenzkonversion	73
7.1.2	Nutzung eines USRP als DFT-Spektrumanalysator	77
7.1.2.1	Anpassung des GNU Radio-Flowgraphs	77
7.1.2.2	Einbindung des USRP in die Python-Software	78
7.1.2.3	Durchführung erster Testmessungen	80
7.1.2.4	Berechnung von breitbandigen Signalspektren	81
7.2	Anbindung der SDR-Hardware an das Zwischenfrequenzsystem	83
7.2.1	Inbetriebnahme des breitbandigen RTL-SDR-Spektrumanalysators an der Zwischenfrequenz	83
7.2.2	Anbindung des USRP an das Zwischenfrequenz-System	87
7.3	Testmessungen an der Zwischenfrequenz	88
7.3.1	Erfassung von breitbandigen ZF-Spektren mit einem RTL-SDR . .	88
7.3.2	Umfassender Test des USRP am ZF-System	93
7.3.2.1	Methode zur Aufnahme von zwei Kanälen mit einem USRP	95
7.4	Erweiterung des DFT-Spektrumanalysators zur Erkennung von RFI	97
7.4.1	Implementierung der Umrechnung zwischen Sky-Frequenz und Zwi- schenfrequenz	97
7.4.2	Integration eines Peak-Detektion-Algorithmus zur RFI-Erkennung .	102
7.4.2.1	Adaption der MATLAB-Funktion in Python	102
7.4.2.2	Test der Peak-Detektion anhand verschiedener Signalspek- tren	103
7.4.3	Bestimmung der vorgesehenen Frequenznutzung eines erfassten Stör- signals	106
7.4.3.1	Übertragung des Frequenznutzungsplans in eine MySQL- Datenbank	106
7.4.3.2	Verbindung zur Datenbank	107
7.5	Diskussion der Ergebnisse der Softwareentwicklung	108
8	Verwendung eines FFT-Spektrometers zur RFI-Detektion	110
8.1	Untersuchung des Hardwareaufbaus eines FFT-Spektrometers	110
8.2	Allgemeine Aspekte zur FFTS-Softwareentwicklung	112
8.2.1	Konfiguration des FFTS über UDP und SCPI	113
8.2.2	Empfang der spektralen Daten per TCP	115
8.3	Entwicklung eines Systems zur RFI-Detektion	115
8.3.1	Aufbau eines Kanalschalters zur Erfassung von zwei Kanälen . .	115
8.3.2	Programmentwicklung zur Aufnahme des Spektrums von zwei Kanä- len	117
8.3.3	Analoge Demodulation von Signalen zur Störerererkennung	119
8.4	Integration der Hardware in das ZF-System	120
8.4.1	Ermittlung des maximal möglichen Signal-Rauschverhältnisses . . .	120

8.4.2	Anpassung des Systems an unterschiedliche Eingangspegel	122
8.4.3	Bestimmung der Eingangsanpassung des FFTS	123
8.4.4	Darstellung des Systemaufbaus	127
8.5	Testmessungen an der Zwischenfrequenz mit einem FFTS	129
8.6	Implementierung einer Funktion zur spektralen Detektion von RFI über die Wahrscheinlichkeitsdichtefunktion von Signalen	133
8.6.1	Prinzipielle Darstellung der Methode	134
8.6.2	Implementierung des Prinzips in einem Python-Programm	135
8.7	Erweiterung des RFI-Monitoring-Backends auf vier ZF-Signale	139
8.7.1	Aufbau des Backend-Einschubs	139
8.7.2	Bestimmung der Streuparameter	141
8.7.3	Schutz der HF-Komponenten vor Störeinstrahlungen	142
8.8	Zusammenfassung der Arbeit mit dem FFT-Spektrometer	146
9	Entwicklung eines webbasierten Benutzerinterfaces	147
9.1	Spektrale RFI-Erkennung im Webinterface	147
9.2	Demodulation von Störstrahlern und Steuerung der SDR-Hardware	149
10	Fazit und Ausblick	153
10.1	Fazit	153
10.2	Ausblick	155
Anhang		A-1
A.1	Weitere Test-Messungen mit einem USRP	A-1
A.2	Streuparameter des Zweifach-Kanalumschalters	A-4
A.3	Weitere Beispiele detektierter Störsignale	A-7
A.3.1	GSM-Mobilfunksignale	A-7
A.3.2	Radarsignale des Fraunhofer-Instituts FKIE in Wachtberg	A-12
A.3.3	Dekodierung des Automatic Dependent Surveillance - Broadcast mittels "Dump1090" [36]	A-14
Quellenverzeichnis		XII
Eidesstattliche Erklärung		XVII

Abbildungsverzeichnis

2.1	Das Radioteleskop Effelsberg des Max-Planck-Instituts für Radioastronomie	5
2.2	Durchmusterung des Gesamthimmels bei 408 MHz [26]	6
3.1	RFI-Situation am Radioteleskop Effelsberg mit geschützten radioastronomischen Bändern (grüne Markierungen)	8
3.2	Frequenzabdeckung der Empfänger des Radioteleskops Effelsberg	9
3.3	Blockschaltbild eines idealisierten Software Defined Radio	11
3.4	Screenshot der grafischen Oberfläche von SDR#	13
4.1	Annäherung eines Rechteckimpulses durch eine endliche Fourierreihe	16
4.2	IQ-Demodulator zur Bildung des komplexen, äquivalenten Tiefpasssignals	19
4.3	Komplexes, äquivalentes Tiefpasssignal eines unsymmetrischen Bandpasssignals	19
4.4	Signalverarbeitungskette zur Bestimmung eines schmalbandigen Teilspektrums	20
4.5	Beispielhaftes Betragsspektrum eines bandpassgefilterten Nutzrauschsignals	21
4.6	Spektrum des IQ-demodulierten Nutzrauschsignals nach einer Filterung mit einem Tiefpass mit einer Grenzfrequenz von 1 MHz und flacher Flanke	22
4.7	Spektrum des digital gefilterten IQ-demodulierten Signals nach der Umsetzung der Abtastrate auf 2 MHz	23
4.8	Gegenseitiger Versatz von Teilspektren aufgrund einer Frequenzungenauigkeit des Lokaloszillators	24
4.9	Wirkung der Fensterung eines Signals auf das Spektrum; oben: zeitlicher Verlauf und Spektrum des ungefensterten sinusförmigen Signals; unten: zeitlicher Verlauf und Spektrum des gefenstereten sinusförmigen Signals	26
5.1	Vereinfachtes Blockschaltbild eines DVB-T-Sticks mit RTL2832U IC	29
5.2	Blockschaltbild des RTL2832U ICs [34]	29
5.3	Auf einen SMA-Anschluss umgebaute Terratec DVB-T-Stick	30
5.4	Blockschaltbild des Elonics E4000 Tuners [9, S. 4]	32
5.5	Blockschaltbild des Rafael Micro R820T Tuners [32, S. 7]	33
5.6	Blockschaltbild des National Instruments USRP-2920 [29]	34
5.7	Prinzip zur Bestimmung der Streuparameter [18, S. 15]	35
5.8	Schematischer Aufbau eines Netzwerkanalysators	35
5.9	Spektrum des Ausgangssignals des Eingangs des DVB-T-Sticks mit Elonics E4000 Tuner (Resolution-Bandwidth = 100 kHz; Video-Bandwidth = 100 kHz)	36
5.10	Betragsspektrum des Elonics E4000-Sticks bei einem ideal abgeschlossenen Eingang	37
5.11	$ S_{11} $ des DVB-T-Sticks mit Elonics E4000 Tuner	38
5.12	S_{11} des DVB-T-Sticks mit Elonics E4000 Tuner im Smithdiagramm	38
5.13	Spektrum des Ausgangssignals des Eingangs des DVB-T-Sticks mit Rafael Micro R820T Tuner zwischen 50 und 2000 MHz (Resolution-Bandwidth = 100 kHz; Video-Bandwidth = 100 kHz)	39
5.14	Spektrum des Rafael Micro R820T-Sticks mit 50 Ω -Abschluss	40
5.15	$ S_{11} $ des DVB-T-Sticks mit Rafael Micro R820T Tuner	40
5.16	S_{11} des DVB-T-Sticks mit Rafael Micro R820T Tuner im Smithdiagramm	41
5.17	Störemissionen des Ein- und Ausgangs TX/RX1 zwischen 50 und 2000 MHz (Resolution-Bandwidth = 100 kHz; Video-Bandwidth = 100 kHz)	42

5.18	Störemissionen des Eingangs RX2 zwischen 50 und 2000 MHz (Resolution-Bandwidth = 100 kHz; Video-Bandwidth = 100 kHz)	42
5.19	Spektrum der Kanäle TX/RX1 und RX2 bei einem 50 Ω -Abschluss	43
5.20	Spektrum eines Rauschsignals im Bereich zwischen 50 und 750 MHz mit unterschiedlicher, zusätzlicher Signaldämpfung	44
5.21	Betrag des Eingangsreflexionsfaktors des USRP-2920 zwischen 10 kHz und 2 GHz	45
5.22	Eingangsreflexionsfaktor des USRP-2920 im Smithdiagramm	45
5.23	Frequenzgang der Signalverarbeitungskette des USRP-2920 und zweier RTL-SDRs bei einem Eingangsträgersignalpegel von -80 dBm	46
5.24	Vergleich des Dynamikumfangs zwischen einem USRP-2920, einem E4000-RTL-SDR und einem R820-RTL-SDR	47
5.25	Blockschaltbild eines einfachen Übertragungssystems	49
5.26	Prinzipielle Darstellung der Verteilung der Intermodulationsprodukte zweiter und dritter Ordnung für ein beispielhaftes Eingangssignal	50
5.27	Prinzip zur Bestimmung des Third-Order-Intercept-Punktes	51
5.28	Blockschaltbild zur Messung des Third-Order-Intercept-Punktes eines DVB-T-Sticks	52
5.29	Darstellung des Third-Order-Intercept-Punktes für beide Tuner; gestrichelte Linie: mittlerer Rauschpegel; schwarze Geraden: Ausgangsprodukte dritter Ordnung; rote bzw. grüne Gerade: Ausgangsprodukte erster Ordnung	53
5.30	Intermodulationsprodukte am Ausgang des USRP-2920	54
5.31	Third-Order-Intercept-Punkt des USRP-2920 - gestrichelte Linie: mittlerer Rauschpegel; IIP3 = +15 dBm, OIP3: +18 dB, SFDR = 70 dB	55
6.1	Prinzip der Einbindung des Raspberry Pi in das Projekt	57
6.2	Raspberry Pi mit verbauten Kühlkörpern und einem RTL-SDR	58
6.3	Emittierte Störstrahlung eines Raspberry Pi mit einem eingesteckten R820T-RTL-SDR zwischen 0 und 1 GHz (normierte Messung, Video-Bandwidth = 100 kHz, Resolution-Bandwidth = 100 kHz	60
7.1	GNU Radio Flowgraph zur Berechnung der FFT der Empfangssignale eines RTL-SDR	61
7.2	Betragspektrum eines IQ-demodulierten HF-Bandpasssignals bei der Wahl einer festen Oszillatorfrequenz f_{LO}	62
7.3	Vergleich verschiedener Average-Einstellungen des LogPowerFFT-Blocks	64
7.4	Rahmenprogramm zur Berechnung der FFT der RTL-SDR-Zeitsignale	66
7.5	Breitbandige Spektromaufnahme des UKW-Rundfunkbandes zwischen zwischen 98 und 108 MHz	68
7.6	Erste Version des Programms zur Berechnung der FFT der RTL-SDR-Zeitsignale	69
7.7	Prinzip zur Aufteilung der Spektromaufnahme zur Vermeidung von Nulllinien und Schultern im Spektrum	70
7.8	Ablauf des Programms zur Berechnung des breitbandigen Spektrums mittels eines RTL-SDR	71
7.9	Versuchsaufbau zur Aufnahme des breitbandigen Spektrums eines einzelnen Trägers	72
7.10	Breitbandige Spektromaufnahme eines einzelnen Trägers im Softwareinterface (Bandbreite = 20 MHz)	73

7.11	Aufbau zur Aufnahme eines breitbandigen Spektrums mit einer Zwischenfrequenz	74
7.12	Breitbandige Spektrumsaufnahme eines einzelnen Trägers mit einer Zwischenfrequenz im Softwareinterface (Bandbreite = 20 MHz)	75
7.13	Ablauf des Programms zur Berechnung des breitbandigen Spektrums mittels eines RTL-SDR	76
7.14	Angepasster GNU Radio Flowgraph zur Einbindung eines National Instruments USRP-2920 in die RFI-Software	78
7.15	Angepasster Programmflussplan mit Einbindung eines National Instruments USRP-2920 in die RFI-Software	79
7.16	Breitbandige Spektrumsaufnahme von 50 bis 350 MHz an der Zwischenfrequenz beim Einsatz des 21 cm - 7 Beam Empfängers mit einem USRP-2920	81
7.17	Detailansicht der Spektrumsaufnahme mit einem USRP-2920 (Bandbreite = 30 MHz)	82
7.18	Detailaufnahme des abschnittsweise aufgenommenen Spektrums mit einem USRP-2920 (Bandbreite = 90 MHz)	82
7.19	Blockschaltbild zur Aufnahme des Spektrums der ZF-Signale	84
7.20	”First Light”-Aufnahme des Spektrums der Milchstraße (Bandbreite = 60 MHz)	85
7.21	Aufnahme des Spektrums des bandpassgefilterten Nutzsignals zwischen 50 und 350 MHz	85
7.22	Vergleichsaufnahme des Spektrums des bandpassgefilterten Nutzsignals zwischen 50 und 350 MHz mit einem Rohde & Schwarz Spektrumanalysator	86
7.23	Aufbau des Systems zur Erfassung des ZF-Signalspektrums mit einem National Instruments USRP-2920	87
7.24	Spektrumsaufnahme von Störsignalen (Bandbreite = 10 MHz)	89
7.25	Vergleichsaufnahme des Spektrums von Störsignalen mit einem Rohde & Schwarz Spektrumanalysator	89
7.26	3D-Wasserfalldiagramm über dauerhafte Messungen an der Schmalband-ZF zwischen 9 und 15 Uhr am 21 cm - 7 Beam Empfänger	91
7.27	Spektrum der Signale des 50-cm-Empfängers zwischen 395 - 415 MHz und 19:00 - 23:00 Uhr	94
7.28	Übersprechen der Signale des 6-cm-Sekundärfokus-Empfängers von Kanal ”TX/RX” auf den Kanal ”RX2” (Bandbreite = 700 MHz)	95
7.29	Programmflussplan zur Aufnahme beider Kanäle des USRP	96
7.30	Blockschaltbild der ZF-Mischung des 1,9-cm-Primärfokus-Empfängers	100
7.31	Detektion von Peaks über den kompletten Schmalband-ZF-Frequenzbereich (Aufnahme mit einem RTL-SDR; Bandbreite = 350 MHz)	104
7.32	Detaillierte Ansicht der Peaks im Bereich eines Radarsenders	105
7.33	Bildausschnitt aus der Detektion eines Störers im ZF-Spektrum mit Informationen zur Frequenznutzung der Bundesnetzagentur	108
8.1	FFTS-Board mit Xilinx Virtex-4 FPGA und 8192 spektralen Kanälen [31]	111
8.2	Blockschaltbild des FFTS mit 8192 spektralen Kanälen	111
8.3	Prinzipieller Aufbau der Software zur Konfiguration des FFTS und zum Empfang der spektralen Daten über TCP [12, S. 15]	113
8.4	Blockschaltbild des Umschalters zur Erfassung von zwei Kanälen mit einem FFTS und einem SDR	116
8.5	Programmflussplan zur Erfassung von zwei Kanälen mittels eines FFTS	118

8.6	Flowgraph zur AM- und FM-Demodulation mit einem RTL-SDR	119
8.7	Versuchsaufbau zur Bestimmung des maximal möglichen Signal-Rausch-Verhältnisses des FFTS im Nutzbandpassspektrum	120
8.8	Maximales Signal-Rausch-Verhältnis des FFTS oberhalb des Bandpasssignals der Zwischenfrequenz	121
8.9	Prinzip der dynamischen Anpassung des Signalpegels durch den Operateur	123
8.10	Betrag der Eingangsanpassung des FFTS	124
8.11	Eingangsreflexionsfaktor des FFTS im Smithdiagramm	124
8.12	Spektrum eines breitbandigen Rauschsignals ohne Eingangsanpassung und Anti-Aliasing-Filter	125
8.13	FFTS-Spektrum bei einem mit 50Ω abgeschlossenen Eingang	126
8.14	Spektrum eines breitbandigen Rauschsignals mit angepasstem FFTS-Eingang und Anti-Aliasing-Filter	127
8.15	Integration des FFTS und des SDR in das ZF-System des MPIfR	128
8.16	Aufnahme eines Signalspektrums des 18-21-cm-Empfängers	129
8.17	Versuchsaufbau zur Erfassung eines künstlich eingespeisten RFI-Signals .	130
8.18	Spektrum des erzeugten RFI-Signals mit deutlich erkennbarer Störlinie bei 12,18 GHz auf Kanal A	130
8.19	Spektrum des UKW-Signals mit überlagertem bandpassgefiltertem Rauschen in Kanal A	131
8.20	Spektrum des UKW-Signals am Eingang des SDR und der zeitliche Verlauf des demodulierten FM-Signals	132
8.21	Spektrum eines 433-MHz ASK-Signals (tiefpassgefiltert) und das demodulierte Zeitsignal	132
8.22	Detektierte Störungen mittels des Algorithmus zur Bestimmung von lokalen Maxima in der Schmalband-ZF	133
8.23	Verlauf der Wahrscheinlichkeitsdichtefunktion für ein Rauschsignal (blau) und ein Rauschsignal mit überlagertem Träger (rot)	134
8.24	Prinzip zur Ermittlung von RFI in einem Signalspektrum	136
8.25	Spektrum des 18-21-cm-Empfängers mit berechneter Differenz der Standardabweichung der einzelnen Frequenzblöcke	137
8.26	Störungsfreies Zwischenfrequenzsignal eines Sekundärfokus-Empfängers .	138
8.27	RFI-Monitoring Backend mit Anschluss von vier ZF-Signalen und Integration von vier RTL-SDR-Servern	140
8.28	Eingangsreflexionsfaktor des ZF-Eingangs "A" im Smith-Diagramm (orange) und in Betragsdarstellung (lila)	141
8.29	Vorwärtstransmission des ZF-Kanals A von der ZF-Signaleinspeisung hin zum FFTS (blau) beziehungsweise RTL-SDR (rot; Einstellung des regelbaren Dämpfungsglieds auf 0 dB, kein zusätzliches Dämpfungsglied)	142
8.30	Abgeschirmte HF-Bauteile zum Schutz vor Störeinstrahlungen (geöffnetes Gehäuse)	143
8.31	Einschub mit vier RTL-SDR-Servern und abgeschirmter HF-Signalverarbeitung	144
8.32	Frontplatte des Einschubs mit Netzschalter, FFTS-Ausgang und vier ZF-Eingängen	145
8.33	Störungsfreies Spektrum des Ausgangssignals des FFTS-Ausgangs (Video-Bandwidth = 30 KHz; Resolution-Bandwidth = 30 kHz)	145
9.1	Screenshots des Webinterfaces zur Darstellung der spektralen Erkennung von RFI	148

9.2	Screenshot des Webinterfaces zur Zuordnung der Frequenznutzung einer Störung	149
9.3	Screenshot des Webinterfaces zur Demodulation von RFI-Signalen und zur Einstellung eines SDR	150
9.4	Softwarestruktur zur Datenerfassung und Steuerung der Hardware	152
10.1	Idealer Signalfluss zur adaptiven Filterung von erkannten Störsignalen	155
A.1	Aufnahme des Spektrums des ZF-Signals des 21 cm - 7 Beam-Empfängers über mehrere Stunden	A-1
A.2	Aufnahme des Spektrums des ZF-Signals des 11-cm-Sekundärfokus-Empfängers über mehrere Stunden	A-2
A.3	Spektrum des ZF-Signals des 21 cm - 7 Beam-Empfängers mit detektierten Peaks in der grafischen QT-Oberfläche	A-3
A.4	Eingangsreflexionsfaktor $ S_{11} $	A-4
A.5	Vorwärtstransmission $ S_{21} $ der FFTS-Kanäle	A-5
A.6	Vorwärtstransmission $ S_{21} $ der SDR-Kanäle	A-6
A.7	GSM-Signal am Eingang des RTL-SDR mit AM-demoduliertem Zeitsignal	A-7
A.8	Breitbandige Spektrumsaufnahme des FFT-Spektrometers mit deutlich erkennbarer Störung im Frequenzband des GSM-Mobilfunks (Marker A14 und B13)	A-8
A.9	Mögliche GSM-Basisstation der detektierten Mobilfunksignale in einer Entfernung von 4,68 km [13, 16]	A-9
A.10	Mögliche GSM-Basisstation der detektierten Mobilfunksignale in einer Entfernung von 2,65 km [13, 2]	A-10
A.11	Mögliche GSM-Basisstation der detektierten Mobilfunksignale in einer Entfernung von 1,94 km [13, 15]	A-11
A.12	Spektrum der Signale der Radaranlage des Fraunhofer Instituts in Wachtberg bei 1330 MHz	A-12
A.13	Geografische Lage der Radaranlage in Wachtberg zum Radioteleskop Effelsberg mit einer Entfernung von 20,13 km [14]	A-13
A.14	Spektrum eines ADS-B-Signals; aufgenommen mit einem FFTS über das Radioteleskop Effelsberg	A-14
A.15	Spektrum eines ADS-B-Signals; aufgenommen mit einem Rohde & Schwarz Spektrumanalysator über das Radioteleskop Effelsberg	A-14
A.16	Darstellung der dekodierten ADS-B-Daten in Google-Maps	A-15

Tabellenverzeichnis

5.1	Vergleich zwischen einem RTL-SDR und einem NI-USRP 2920 [29]	28
5.2	Vergleich verschiedener Tuner-Spezifikationen [30]	30
5.3	Vielfache der Frequenz des Quarzoszillators eines RTL-SDR mit E4000-Tuner	36
6.1	Optionen für den Aufruf des "rtl_tcp"-Programms	59
7.1	Absolute Frequenzabweichung für verschiedene Frequenzen eines USRP	88
7.2	Messung der Programmlaufzeit zur Erfassung einer Bandbreite von 300 MHz mit einem RTL-SDR	92
8.1	Spezifikationen des verwendeten AFFTS [25, S. 5]	112
8.2	Übersicht über die wichtigsten Befehle zur Konfiguration eines FFTS	114

Listingverzeichnis

6.1	Vorbereitung zur Installation eines RTL-SDR Server auf einem Raspberry Pi	59
6.2	Installation eines RTL-SDR Server [21]	59
7.3	Fehler bei der Übertragung der USRP-Signale mit einer hohen Abtastrate	80
7.4	MATLAB-Skript zum dreidimensionalen Plotten von FFT-Daten	90
7.5	Funktion zur Umrechnung zwischen Zwischen- und Sky-Frequenz	101
7.6	Funktion zur Ermittlung von lokalen Peaks in einem Array	103
8.7	Beispiel zur Konfiguration eines AFFTS mittels eines UDP-Telnet Client	114

1 Einleitung und Aufgabenstellung

Diese Masterarbeit wurde im Rahmen des Masterstudiengangs Systemtechnik an der Hochschule Koblenz für das Max-Planck-Institut für Radioastronomie angefertigt. Die Arbeit befasst sich mit der Installation eines Software Defined Radios an der Zwischenfrequenz des Radioteleskops Effelsberg zur Detektierung von Funkstörungen. Die offizielle Aufgabenstellung ist nachfolgend dargestellt.

Offizielle Aufgabenstellung: Für die Datenaufnahme von breitbandigen, radioastronomischen Signalen im Radioteleskop Effelsberg wird die Erkennung von Störstrahlern immer wichtiger. In einem Umfeld von zunehmenden Funkdiensten in der naturgemäß beschränkten Frequenzdomäne wird es immer schwieriger das astronomische Nutzsignal von diesen ‚manmade‘ Signalen zu unterscheiden. Eine Möglichkeit dazu ist die Detektion und Demodulation dieser Signale mit üblichen Demodulationsverfahren, um daraus auf die Herkunft und somit auf mögliche Vermeidungsstrategien schließen zu können.

Zu diesem Zweck soll am Radioobservatorium Effelsberg ein Software Defined Radio (SDR) an das ZF-System der Empfangssysteme angeschlossen werden. Die Hardware dafür wird eines der breitbandigen COTS-Systeme sein, für die es im Internet verschiedene Anwendungssoftware gibt. Den Anfang macht ein DVB-T USB-Stick, der mit einer Eingangsfrequenz von 10 MHz bis 2,6 GHz für diesen Zweck gut geeignet ist. Dieses System soll mit einem Rechner ans Intranet des Instituts angebunden werden, sodass mit einem Internetbrowser auf die Daten zugegriffen werden kann. Dabei sollen sowohl die spektrale Verteilung als auch das detektierte Signal sichtbar und hörbar gemacht werden. Die Hardware muss über dieses Webinterface gesteuert werden können, entsprechende Zugriffsmechanismen müssen vorgeschlagen und implementiert werden. Bei Bedarf kann die Arbeit auf ein besseres System erweitert werden.

Ziel dieser Arbeit ist die Inbetriebnahme der oben beschriebenen Hardware und das Erstellen eines geeigneten Webinterfaces dafür. Bei der Inbetriebnahme kann auf die langjährige Erfahrung mit Hochfrequenztechnik und Störstrahlung im Hause zurückgegriffen werden. Grundlegende Erfahrungen zur Programmierung von webbasierten Anwendungen werden allerdings vorausgesetzt. Die Baugruppe wird im Observatorium in Effelsberg in Betrieb genommen und hier unter realen Bedingungen getestet. Anschließend ist der Einsatz am Radioteleskop Effelsberg geplant.

Zielsetzung der Arbeit ist aufgrund der gegebenen Aufgabenstellung, dass ein System implementiert wird, das spektrale Störungen erfasst und die Ergebnisse den Mitarbeitern, speziell den Astronomen und Operateuren des Max-Planck-Instituts für Radioastronomie, in einem allgemein zugänglichen Web-Interface zur Verfügung stellt. Dabei sollen Störstrahler im Spektrum erkannt werden und mittels verschiedener Methoden auf eine mögliche Quelle der Störung geschlossen werden. Durch die Demodulation von Signalen und die akustische Wiedergabe wird es dem Anwender zudem möglich die im Spektrum detektierte Störung einem Dienst zuzuordnen oder dem zeitlichen Verlauf zu folgen. Aufgrund der hohen Komplexität von Modulationsverfahren werden in dieser Arbeit ausschließlich analoge Modulationen betrachtet und in das System integriert. Der Schwerpunkt der Arbeit liegt daher auf der spektralen Erkennung von Störsignalen.

Um die angestrebten Ziele umzusetzen, befasst sich diese Arbeit auf der Grundlage der Berechnung von Spektren mit der praktischen Umsetzung dieser Methoden mit einem Software Defined Radio als Hardwareinterface. Die verwendeten Hardwareinterfaces werden mittels verschiedener Messmethoden charakterisiert und die Eignung der Hardware für diese Anwendung betrachtet. Dabei wird auf Methoden der HF-Messtechnik zurückgegriffen und im Zusammenhang mit den im Studium erlernten Grundlagen die Messergebnisse diskutiert.

Die Software für diese Arbeit wird in der Programmiersprache Python entwickelt. Dazu werden theoretische Methoden zur Softwareentwicklung angewandt, um ein reproduzierbares und wartbares System zu entwickeln. Das per Software ermittelte Spektrum der Empfangssignale des Radioteleskops Effelsberg wird zur Erkennung von Störungen genutzt. Dabei werden verschiedene Algorithmen entwickelt, die zur Detektion von Störungen in einem Signal dienen.

Die meisten Informationsquellen dieser Arbeit stammen aus dem Internet. Da die verwendeten Hardwareinterfaces teilweise sehr große Communities im Amateur- und Forschungsbereich haben, ist der Austausch von Informationen und Programmen über das Internet sehr verbreitet. Daher bezieht sich diese Arbeit häufig auf eben diese. Im zweiten Teil der Arbeit wird ein vom Institut entwickeltes Spektrometer verwendet. Dabei konnte auf die umfangreiche Erfahrung der Kolleginnen und Kollegen im Institut zurückgegriffen werden. Die allgemein theoretischen Grundlagen der Arbeit wurden vor allem aus Publikationen von Messgeräteherstellern entnommen.

Dieses erste Kapitel stellt die Aufgabenstellung, die verwendeten Methoden zum Erreichen der Zielsetzung und den Inhalt der einzelnen Kapitel dar.

Das zweite Kapitel beschreibt die Arbeit des Max-Planck-Instituts für Radioastronomie und gibt einen Überblick über die wichtigsten Arbeits- und Forschungsbereiche des Instituts. Das darauf folgende Kapitel befasst sich mit der Erklärung und der Definition der grundlegenden Begriffe dieser Arbeit. Besonders die Bedeutung des Begriffs "RFI" wird speziell für die Radioastronomie dargestellt. Zudem wird die von der ITU-R festgelegte Definition eines "Software Defined Radio" angeführt und die daraus resultierenden Anforderungen an Hard- und Software diskutiert. Anhand einer idealisierten Darstellung einer solchen Hardware werden Vor- und Nachteile von Software Defined Radios gezeigt und die zunehmende Bedeutung dieser in der Funktechnik herausgearbeitet.

Im vierten Kapitel werden die theoretischen Grundlagen zur Berechnung eines Signalspektrums betrachtet. Diese sind für diese Arbeit elementar, da die verwendete Hardware als Spektrumanalysator Einsatz findet, dessen Ergebnisse per Software weiterverarbeitet werden. Das fünfte Kapitel stellt die für diese Arbeit verwendete Hardware dar, einen herkömmlichen DVB-T-Stick, der per USB an einen Rechner angeschlossen wird sowie ein USRP-2920. Dabei wurden zwei verschiedene Sticks mit unterschiedlichem Tuner untersucht. Zudem wurden diese Hardwareinterfaces durch verschiedene Messungen charakterisiert und ein Vergleich zwischen beiden aufgebaut. Die Vor- und Nachteile, die sich aus diesen Messungen ergeben werden am Ende des Kapitels diskutiert.

Nach der umfangreichen messtechnischen Charakterisierung der Hardware wird der Einsatz eines DVB-T-Sticks als Software Defined Radio über einen Raspberry Pi im sechsten Kapitel gezeigt. Dieser dient als Server, der die Daten des mit einem USB-Anschluss ausgestatteten Sticks per Ethernet an eine beliebige Gegenstelle sendet. Mit diesem Hardwareaufbau wird im nachfolgenden Kapitel eine Software zur breitbandigen Aufnahme eines Signalspektrums mittels eines DVB-T-Sticks und eines USRP entwickelt. Dabei werden Probleme der Hardware gezeigt, die diese für die angestrebte Anwendung unbrauchbar machen. Zudem werden die wichtigsten Punkte in der Entwicklung der Software dargestellt.

Mit dieser Software werden im Laufe des Kapitels Messungen am Zwischenfrequenzsystem des Instituts durchgeführt. Da die Hardware an eben diesem System arbeiten soll, werden weitere Schwierigkeiten bei der Erfassung des Signalspektrums mit einem DVB-T-Stick gezeigt. Diese Probleme werden nachfolgend teilweise durch Softwarelösungen umgangen.

Zudem wird die Entwicklung der Software für ein National Instruments USRP-2920 diskutiert. Dabei wird die Integration der Hardware in die bisher entwickelte Software gezeigt. Das Ende des Kapitels stellt eine Zusammenfassung der Ergebnisse dar.

Das achte Kapitel zeigt die Arbeit mit einem FFT-Spektrometer, das vom Institut entwickelt wurde und zur Erfassung von Spektren über eine hohe Momentan-Bandbreite dient. Für diese Hardware wird ebenso ein Softwareinterface entwickelt, das die spektralen Daten erfasst und weiterverarbeitet. Dabei wird zudem ein Algorithmus entwickelt, der zur Detektion von Störsignalen in einem Signal dient und auf der Grundlage der Wahrscheinlichkeitsdichteverteilung von Signalen beruht. Weiterhin wird das Zusammenspiel des Spektrometers mit einem DVB-T-Stick zur Demodulation von Signalen gezeigt.

In diesem Kapitel wird zudem das entwickelte System auf vier ZF-Signale erweitert. Dazu wurde die Hardware entsprechend erweitert und die Software für die Verarbeitung der Signale angepasst. Die Hardware wurde in diesem Zusammenhang in ein EMV-konformes Gehäuse verbaut, um die eigentlichen Nutzsignale durch auftretende Störstrahlung nicht zu beeinflussen.

Im vorletzten Kapitel wird ein Webinterface entwickelt, das die Ergebnisse der Software der spektralen RFI-Erfassung in einem Webbrowser darstellt. Außerdem wird die Steuerung des DVB-T-Sticks über eine Website implementiert, um Parameter wie die Verstärkung und die Mittenfrequenz zu steuern. Über diese Website ist es zudem möglich zwischen verschiedenen Demodulationen zu wählen und das demodulierte Signal akustisch wiederzugeben.

Den Abschluss dieser Arbeit stellt ein Fazit dar, welches die Ergebnisse dieser Arbeit zusammenfassend darstellt. Dabei werden die Probleme zusammengefasst, die während der Entwicklung auftraten. Die entwickelten Lösungsansätze werden diskutiert und offene gebliebene Frage gezeigt. Zudem wird ein Ausblick gegeben, inwiefern dieses System weiterentwickelt werden kann und welche Möglichkeiten sich durch dieses System in Zukunft erschließen.

2 Vorstellung des Max-Planck-Instituts für Radioastronomie

Das Max-Planck-Institut für Radioastronomie arbeitet auf dem Gebiet der Radio- und Infrarot-Astronomie. Dabei geht es um die Erforschung der Physik von Sternen, Galaxien, Pulsaren und vielen weiteren Objekten des Weltraums. Für diese Beobachtungen nutzt das Institut das 1972 fertiggestellte Radioteleskop in Effelsberg. Mit einem Durchmesser von 100 Metern ist das Radioteleskop eines der größten der Erde. Aufgrund ständiger Weiterentwicklung der Technik des Teleskops gilt es bis heute als eines der modernsten Radioteleskope [26].

Die Radioastronomie ist seit ihren Anfängen in den 1930er Jahren zu einer der wichtigsten Methoden zur Erforschung des Universums geworden. Sie ergänzt dabei die Möglichkeiten der optischen Astronomie um die Beobachtung von durch Staubwolken verdeckten Objekten. So sind alle Astronomen, die bisher mit einem Nobelpreis ausgezeichnet wurde, ausschließlich Radioastronomen [26].



Abbildung 2.1: Das Radioteleskop Effelsberg des Max-Planck-Instituts für Radioastronomie

Mit einer Oberfläche von 7850 Quadratmetern ist das Radioteleskop ein höchst empfindlicher Empfänger für die äußerst schwachen Radiosignale aus dem Weltraum. Neben der radioastronomischen Forschung ist zudem die Entwicklung von Elektronik und Technologien ein wesentlicher Bestandteil der Arbeit des Instituts. Im Zusammenspiel mit internationalen Forschungsteams werden bei VLBI-Messungen mehrere Radioteleskope auf der Welt zusammengeschaltet. Daraus resultiert ein Radioteleskop, das einen Durchmesser hat, der dem Abstand der Radioteleskope entspricht [26].

Zudem betreibt das Institut ein LOFAR Radioteleskop. Dieses ist Teil von über 40 Stationen, die über ganz Europa verteilt sind. Das LOFAR Radioteleskop besteht dabei aus vielen einzelnen Dipolantennen. Verwendung findet dieses Teleskop bei der Messung in Frequenzbereichen von 10 bis 140 MHz. Auch beim in der chilenischen Atacamawüste erbauten APEX-Radioteleskop hat das Institut einen großen Teil zur Forschung und Entwicklung beigetragen [26].

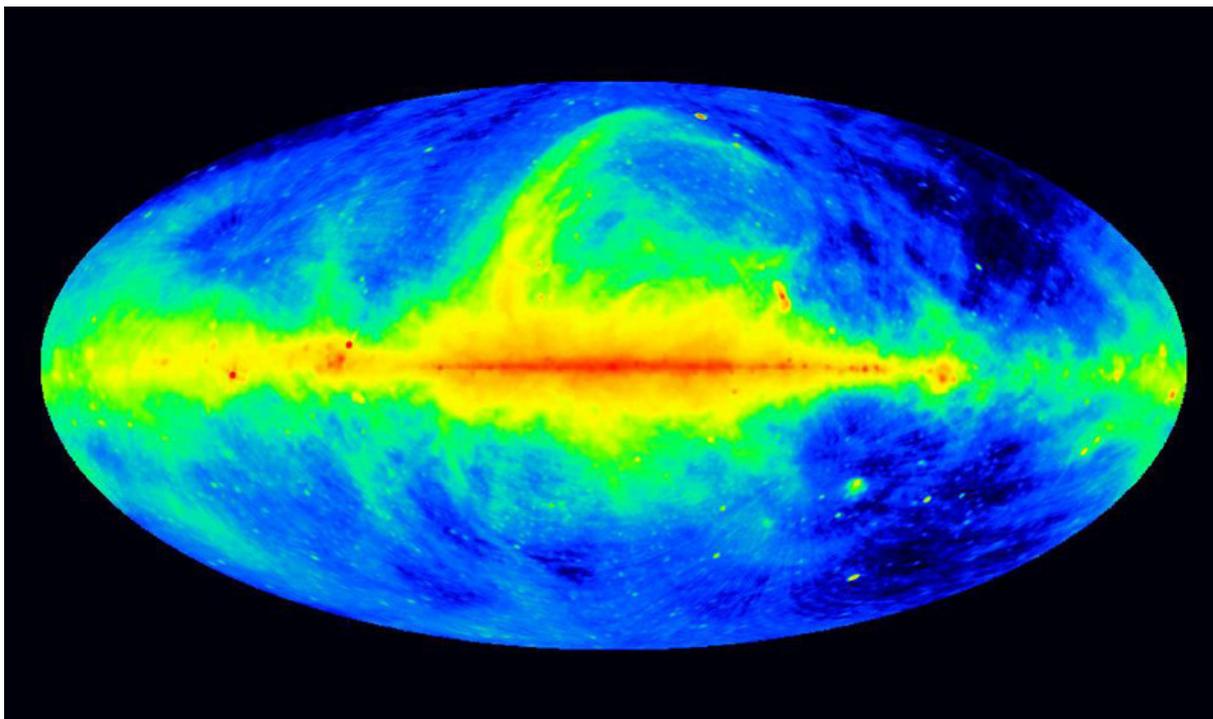


Abbildung 2.2: Durchmusterung des Gesamthimmels bei 408 MHz [26]

3 RFI und SDR - Begriffsdefinitionen und Grundlagen

Dieses Themen einleitende Kapitel gibt eine Übersicht über die in dieser Arbeit verwendeten Begriffe. Zu Beginn zeigt eine offizielle Definition von "Radio Frequency Interference" durch die ITU die große Bedeutung von RFI für die Radioastronomen und verstärkt den Nutzen dieser Arbeit. Im Anschluss wird die von der ITU-R herausgegebene Definition von "Software Defined Radio" angeführt und erläutert. In diesem Zusammenhang werden die Vor- und Nachteile eines Software Defined Radio diskutiert und die Anforderungen an die Hardware, damit ein Gerät im Sinne der Definition der ITU als Software Defined Radio gilt, dargestellt. Weiterhin wird ein kurzer Überblick über die am verbreitetsten Softwareinterfaces zum Betreiben von Software Defined Radios gegeben.

3.1 Radio Frequency Interference - eine Definition

Die ITU hat im wissenschaftlichen Bericht "Techniques for mitigation of radio frequency interference in radio astronomy" (RA 2126 – 1) folgende Definition von *RFI* (Radio Frequency Interference, deutsch: Funkstörungen) für Radioastronomen festgelegt:

Radio Frequency Interference (RFI): To a radio astronomer, RFI is any unwanted addition to the cosmic signal that has the potential to degrade or prevent the successful conduct of an observation. The term RFI will be used in this sense throughout this Report. Unlike thermal noise, which has stable temporal stochastic properties (white noise) and can be dealt with through radiometric detection (i.e. long integration times and on-source minus off-source subtraction), an RFI signal is temporally, spatially or spectrally structured and can obscure a deep-space signal or produce a false positive detection [20, S. 2] .

Aus dieser Definition folgt, dass ein RFI-Signal ein Signal ist, welches als ein Signal aus dem All oder als eine positive Entdeckung gedeutet werden kann, jedoch keines der beiden in Wirklichkeit ist. Allgemein ist RFI dabei jedes Signal, das sich dem kosmischen Signal überlagert und den Erfolg einer Beobachtung beeinträchtigt. In dieser Arbeit werden solche Störungen mittels eines Software Defined Radio erfasst und auf eine potenzielle Quelle dieses Störsignals geschlossen. Diese Störungen können beispielsweise durch Radarsignale oder auch mobile Kommunikationssysteme erzeugt werden. Eine Übersicht über die RFI-Situation am Radioteleskop Effelsberg ist in Abbildung 3.1 zu sehen.

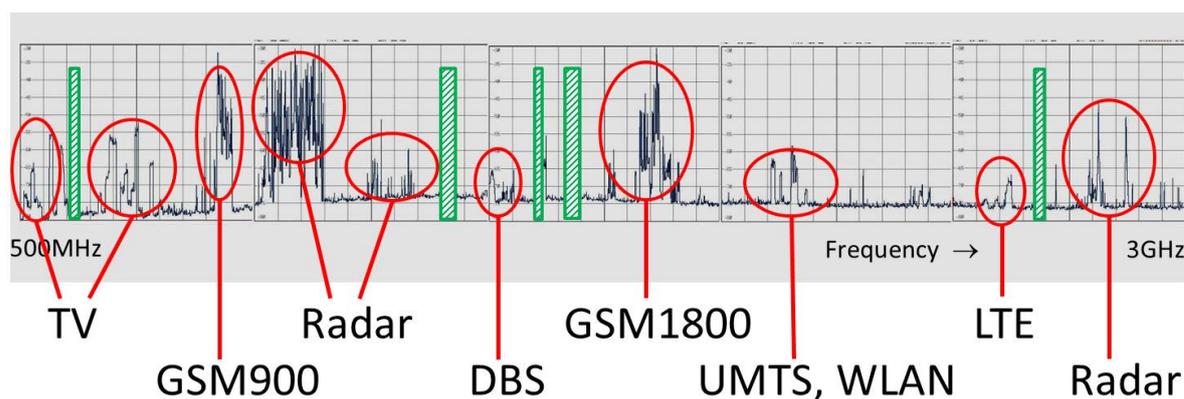


Abbildung 3.1: RFI-Situation am Radioteleskop Effelsberg mit geschützten radioastronomischen Bändern (grüne Markierungen)

Diese Darstellung zeigt, dass speziell Radarsignale und Mobilfunkdienste im Empfangsbereich zwischen 500 MHz und 3 GHz am Radioteleskop stark vertreten sind. Eben diese Störungen gilt es in dieser Arbeit zu erfassen. Da solche Störsignale den Erfolg einer Messung durch die Astronomen maßgeblich beeinflussen, ist die Detektion von solchen Signalen äußerst wichtig, da Messungen auch außerhalb der für die Radioastronomie geschützten Bänder, welche in Abbildung 3.1 als grüne Blöcke gekennzeichnet sind, stattfinden. Eine Übersicht über die Frequenzabdeckung der Empfänger des Instituts ist in Abbildung 3.2 zu sehen. Anhand dieser wird deutlich, dass die Empfänger nahezu nahtlos den kompletten Frequenzbereich zwischen 300 MHz und 100 GHz abdecken und in allen dazwischen liegenden Frequenzbändern Messungen stattfinden.

Daher wird ein System entwickelt, das den Mitarbeitern des Instituts die Erkennung von Störsignalen erleichtert. Diese können durch alle im Umkreis betriebenen Funkanwendungen hervorgerufen werden. Neben konstanten Störungsquellen wie Fernseh- und Radiostationen beeinflussen auch transiente Störquellen wie Radaranlagen die radioastronomischen Messungen. Störungen können statt extern auch durch das Observatorium an sich entstehen. So strahlen die Computer und verwendeten Geräte Störstrahlung ab, die Messungen negativ beeinflussen können. Für die Detektion solcher Störer wird ein Software Defined Radio eingesetzt. Die Definition des Begriffs und die Bedeutung werden im nachfolgenden Abschnitt gezeigt.

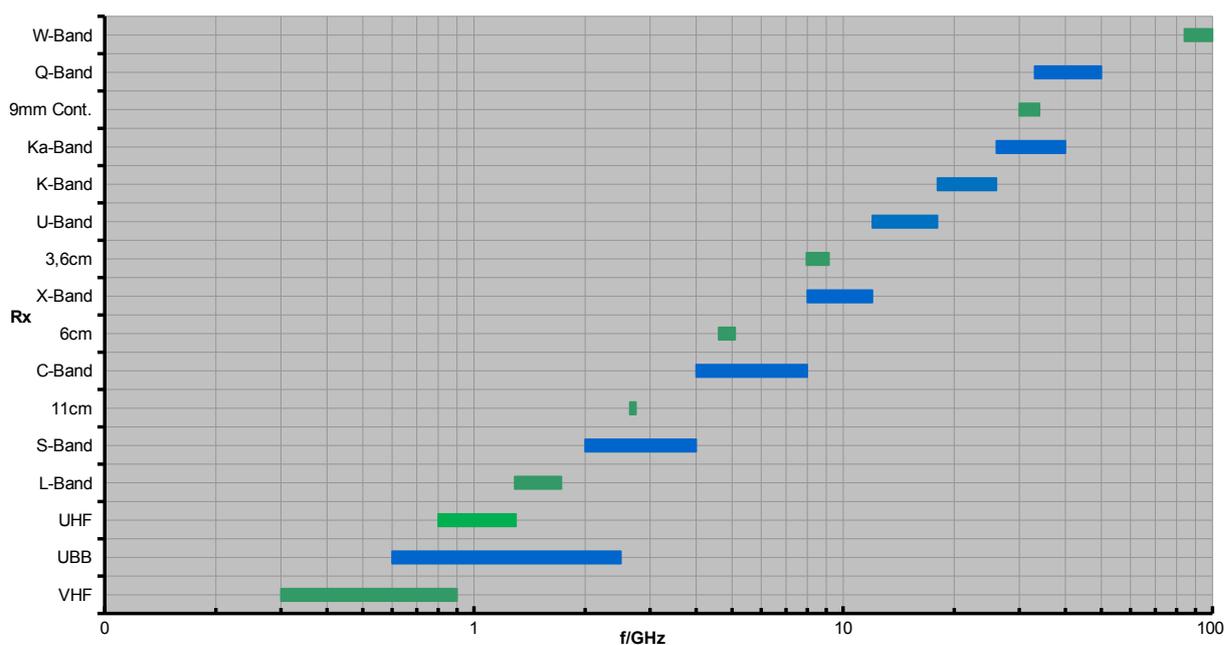


Abbildung 3.2: Frequenzabdeckung der Empfänger des Radioteleskops Effelsberg

3.2 Definition des Begriffs Software Defined Radio

Die ITU-R, die Unterabteilung der International Telecommunication Union, die sich mit Funkverkehr (Radiocommunication) befasst, hat im Bericht "Cognitive radio systems specific for International Mobile Telecommunications systems" eine offizielle Definition des Begriffs *Software Defined Radio*, kurz SDR, festgelegt.

Software-Defined Radio (SDR): A radio transmitter and/or receiver employing a technology that allows the RF operating parameters including, but not limited to, frequency range, modulation type, or output power to be set or altered by software, excluding changes to operating parameters which occur during the normal pre-installed and predetermined operation of a radio according to a system specification or standard [19, S. 5].

Aus dieser Definition ergibt sich, dass ein Funksystem dann als Software Defined Radio bezeichnet werden kann, wenn Betriebsparameter, wie der Frequenzbereich, aber auch die Verstärkung oder die Modulation, mittels Software während der Laufzeit geändert werden können. Ausgenommen sind jedoch Systeme, bei denen solche Veränderungen bereits im Voraus geplant und vorgesehen sind. Dies kann zum Beispiel eine Änderung der Modulationsart bei variierendem Eingangspegel sein, wie es beispielsweise bei verschiedenen Mobilfunkstandards vorkommt.

Auf Grundlage dieser Definition und in Anbetracht der Ziele dieser Arbeit kann das vorgesehene System per Definition als Software Defined Radio bezeichnet werden. Da durch die flexible Demodulation von empfangenen Signalen auf die mögliche Quelle der Signale geschlossen wird, ist eine ständige Anpassung an die entsprechende Anwendung notwendig. Dabei wird vor allem der Frequenzbereich geändert, in dem die Hardware arbeitet. Zudem sind die Anpassungen nicht vorhersehbar, da die Störungen nicht deterministisch auftreten.

Im Zusammenhang mit dem Thema Software Defined Radio ist häufig der Begriff "Cognitive Radio" zu finden. Um die beiden Begriffe gegeneinander abzugrenzen, ist nachfolgend die Definition des Begriffs durch die ITU-R aus demselben Bericht dargestellt.

Cognitive Radio System (CRS): A radio system employing technology that allows the system to obtain knowledge of its operational and geographical environment, established policies and its internal state; to dynamically and autonomously adjust its operational parameters and protocols according to its obtained knowledge in order to achieve predefined objectives; and to learn from the results obtained [19, S. 5].

Diese Definition sagt aus, dass ein Cognitive Radio System ein System ist, das sich aufgrund seiner bekannten Konfiguration und angestrebten Anwendung dynamisch und autonom an die gegebenen Bedingungen anpasst, um vorgegebene Ziele zu erreichen. Das angestrebte System als Cognitive Radio zu bezeichnen wäre daher nicht korrekt. Da eine autonome Anpassung des Systems an unbekannte Bedingungen vorausgesetzt wird, deckt sich diese Definition nicht mit der angestrebten Anwendung.

3.3 Allgemeine Aspekte zur Hard- und Software eines Software Defined Radio

Die nachfolgenden allgemeinen Betrachtungen zum Thema Software Defined Radio beziehen sich, sofern nicht anderweitig angegeben, auf den Bericht "Frequenzregulierung für Software Defined Radio und Cognitive Radio" des Instituts für Rundfunktechnik [6, S. 19-33]. Der Aufbau eines idealen Software Defined Radio wird diskutiert und Vor- und Nachteile dargestellt. Außerdem werden einige Anwendungen gezeigt, bei denen bereits heute Software Defined Radios verwendet werden.

3.3.1 Idealisierter Aufbau eines SDR-Hardwareinterface

Im Kapitel 3.2 wurde bereits die offizielle Definition eines Software Defined Radio der ITU-R diskutiert und die damit prinzipiell verbundenen Anforderungen an Hard- und Software. Dieser Abschnitt erläutert den Aufbau einer idealisierten SDR-Hardware. Aufgrund der angestrebten Anwendung eines SDR als Empfänger wird nur diese Richtung der Signalverarbeitung betrachtet. Ein ideales SDR ist allgemein betrachtet ein solches, welches hinter einer RF-Einheit, welche aus analogen Verstärkern und Filtern besteht, eine direkte digitale Signalverarbeitung durchführt und per Software steuerbar ist (siehe Abbildung 3.3).

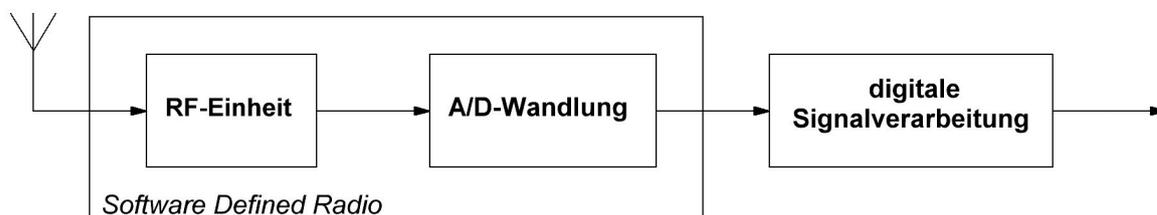


Abbildung 3.3: Blockschaltbild eines idealisierten Software Defined Radio

Die erste Stufe des idealen SDR verstärkt und filtert die Eingangssignale. Zudem kann in dieser Stufe eine Frequenzkonversion stattfinden. Entweder werden die Eingangssignale in eine Zwischenfrequenzlage oder direkt ins Basisband gemischt. Diese analogen Signale werden durch einen A/D-Wandler digitalisiert und danach digital weiterverarbeitet. Der Vorteil eines solchen Aufbaus ist, dass das Verfahren "frequenzagil" ist. Das bedeutet, dass das Software Defined Radio für jede Frequenz flexibel und gleichwertig genutzt werden kann. Der einzige Faktor, der dieses Verhalten einschränkt, ist die Abtastrate des A/D-Wandlers, die die höchste Frequenz der Signale begrenzt.

3.3.2 Parameter der technischen Komponenten eines Software Defined Radio

Im Zuge dieser Arbeit werden keine Freiraumwellen mit Hilfe einer Antenne empfangen. Daher wird die Charakteristik einer solchen Antenne und deren Anbindung an das System nicht von Bedeutung. Die erste Stufe eines SDR, die RF-Einheit, filtert und verstärkt die Eingangssignale. Die Anforderungen an die verbauten Verstärker sind eine hohe Linearität und eine möglichst geringe Rauschzahl. Die eingangsseitige Vorselektion durch ein flexibles Bandpassfilter dient vor allem zur Begrenzung der Bandbreite und zur Unterdrückung von Störsignalen. Diese Anforderungen schränken wiederum in einem gewissen Maß die Flexibilität ein.

Die anschließende A/D-Wandlung ist maßgeblich für die erfassbare Bandbreite des SDR. Dabei sind vor allem ein großer störungsfreier, dynamischer Bereich (SFDR) und ein hoher Signal-Rauschabstand, der durch die Abtastrate des A/D-Wandlers bestimmt wird, von Bedeutung. Diese Parameter sind maßgeblich für das Erreichen einer möglichst hohen Momentan-Bandbreite und einer zuverlässigen A/D-Wandlung. In heute üblichen SDRs werden bereits Momentan-Bandbreiten von 500 MHz erreicht. Die digitale Signalverarbeitung der abgetasteten Empfangssignale kann durch verschiedene Hardwarekomponenten realisiert werden. Dazu können herkömmliche CPUs, aber auch FPGAs und DSPs genutzt werden. Je nach Anwendung ist abzuschätzen, welcher Prozessor am besten geeignet ist. In dieser Arbeit wird die Software auf einem herkömmlichen Linux-Server laufen. Daher wird die entsprechend verbaute CPU zur digitalen Signalverarbeitung genutzt. Das hat den Vorteil, dass die Software flexibel verändert und angepasst werden kann.

3.3.3 Praktische Anwendungen von Software Defined Radios

Software Defined Radios werden aufgrund der Flexibilität und der Rekonfigurierbarkeit während der Laufzeit immer beliebter. Die Weiterentwicklung von Algorithmen kann durchgeführt werden, obwohl die Geräte schon auf dem Markt und in Betrieb sind. Diese nachträglichen Erweiterungen in der Signalverarbeitung in einem SDR können auch ohne direkten Kontakt mit dem Gerät erfolgen ("Over The Air Configuration"). Bereits heute gibt es einige Anwendungen und Bereiche, die Software Defined Radios im Sinne der Definition der ITU-R verwenden, zum Beispiel

1. Amateurfunkdienste,
2. Mobilfunkdienste,
3. Rundfunkdienste und die
4. Radioastronomie.

3.4 Übersicht über Softwareinterfaces für Software Defined Radios

Es gibt zwei weitverbreitete Softwareinterfaces für Software Defined Radios. Auf der einen Seite steht SDR#, welches eine umfangreiche Software darstellt und unter Windows läuft. Diese Software ist nahezu komplett abgeschlossen und bietet daher wenig Raum für eigene Entwicklungen. Auf der anderen Seite steht GNU Radio. GNU Radio ist ein Software Development Kit (SDK) mit einer umfangreichen Bibliothek an Funktionen und Methoden. Somit ist GNU Radio mehr an Entwickler und Programmierer gerichtet, da Anwendungen entweder über das grafische Tool "GNU Radio Companion" oder über die Programmiersprache Python entwickelt werden müssen. Es gibt jedoch bereits eine große Community, welche komplette Programme über das Internet zur Verfügung stellt. Beide Softwaretools werden nachfolgend erläutert und die wichtigsten Funktionen dargestellt.

SDR# ist ein unter Windows laufendes Tool, welches vor allem als Spektrumanalysator in Verbindung mit einem SDR dient (siehe Abbildung 3.4). Über die grafische Oberfläche kann neben der Einstellung des gewünschten Frequenzbereichs auch die Verstärkung der Signale durch das SDR geregelt werden. Zudem ist eine Demodulation der Signale in verschiedenen Formen, zum Beispiel AM und FM möglich. Über verschiedene Regler können Einstellungen zur FFT und zur Mittelwertbildung der Signale vorgenommen werden. In dieser Arbeit wurde diese Software vor allem zur Messung von Parametern zur Charakterisierung von SDRs verwendet. Wie zuvor angedeutet, ist diese Software in sich fast komplett geschlossen und bietet kein Programmierinterface. Daher ist sie für die Entwicklung von eigenen Programmen ungeeignet. Die Installation und Konfiguration ist sehr einfach und es werden keine weiteren Kenntnisse benötigt. Dadurch kann diese Software für eine schnelle Inbetriebnahme eines SDR genutzt werden.

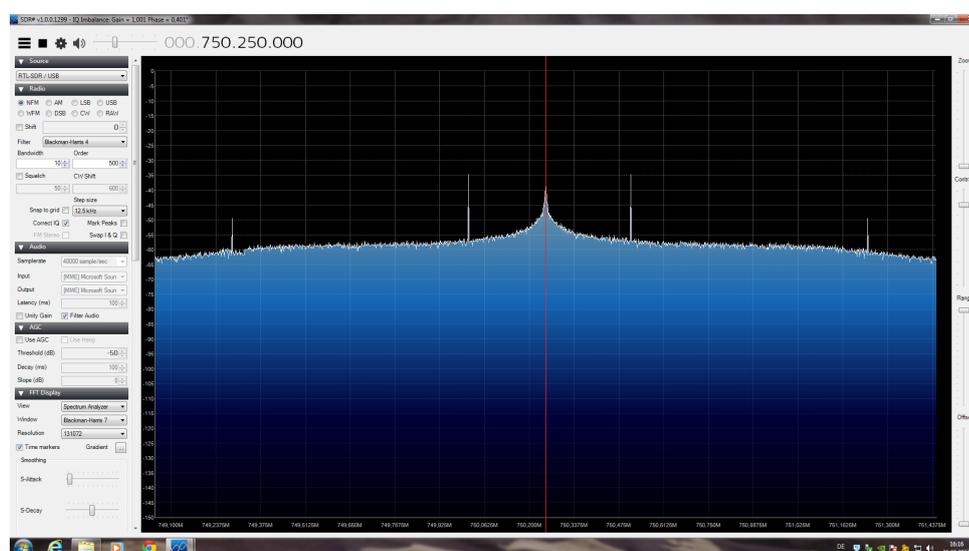


Abbildung 3.4: Screenshot der grafischen Oberfläche von SDR#

Aus den oben hervorgebrachten Gründen wird in dieser Arbeit zur Entwicklung der eigenen Software das SDK "GNU Radio" verwendet. GNU Radio bietet umfangreiche Bibliotheken mit einem großen Umfang an Funktionen zur Signalverarbeitung. Zudem gibt es eine große Community, welche ständig Erweiterungen entwickelt und eine hervorragende Dokumentation. Die Bibliotheken werden unter anderem von der Programmiersprache Python und C++ unterstützt.

GNU Radio bietet zudem eine grafische Oberfläche namens "GNU Radio Companion". Über diese Oberfläche können eigene Signalflussgraphen erstellt werden, sogenannte "Flowgraphs", welche als Grundlage für die eigenen Programme dienen können. Innerhalb dieser Oberfläche können Blöcke, welche verschiedene Klassen und Funktionen implementieren, zusammenschaltet und so der gewünschte Signalfluss umgesetzt werden. Die Software wandelt diese Flowgraphs in ein Python-Programm um, welches dann als Umgebung für die weitere Entwicklung genutzt werden kann. So wird das Handling mit den Funktionen und Objekten von GNU Radio dem Programmierer in großen Teilen abgenommen, beziehungsweise aufgezeigt. Da der Umfang von GNU Radio sehr groß ist, sei hier auf die Dokumentation von GNU Radio (Version 3.7) im Internet verwiesen [11]. Auf Einzelne für diese Arbeit relevante Bestandteile und Funktionen wird in den dazugehörigen Kapiteln eingegangen.

Der nächste Abschnitt befasst sich mit den Grundlagen zur Spektrumsanalyse. Da die Hardware im Zusammenspiel mit der entwickelten Software als Spektrumanalysator dienen soll, dessen Messergebnisse weiterverarbeitet werden, ist die Betrachtung der theoretischen Hintergründe sinnvoll für das Verständnis der Arbeit. Im darauf folgenden Kapitel wird eine umfangreiche Untersuchung verschiedener SDR-Hardwarekomponenten vorgenommen. Diese umfasst die Charakterisierung zweier DVB-T-Sticks mit jeweils unterschiedlichem Tuner und einem USRP von National Instruments. Dabei wird auch näher auf den Aufbau und die Funktion der Hardware eingegangen. Die Messungen umfassen unter anderem die Bestimmung der Eingangsreflexion, der Dynamikumfang und die Intermodulationsfestigkeit. Diese werden gegenübergestellt und die daraus resultierenden Folgen diskutiert.

4 Theoretische Grundlagen der spektralen Analyse

Dieses Kapitel stellt die Theorie der spektralen Signalanalyse dar sowie den prinzipielle Aufbau eines Spektrumanalysators. Zudem werden Anforderungen diskutiert, die an einen Spektrumanalysator gestellt werden, um eine optimale Darstellung des Signalspektrums zu erhalten. Aufgrund der Anwendung, die sich hauptsächlich mit der spektralen Analyse von Signalen beschäftigt, dient dieses Kapitel als theoretische Grundlage für die spätere praktische Anwendung. Die Ausarbeitungen beziehen sich dabei, sofern nicht anders angegeben, auf die Fachliteratur "Grundlagen der Spektrumanalyse" von Christoph Rauscher [33, S. 7-88].

4.1 Einsatzgebiete der spektralen Analyse

Häufig wird der zeitliche Verlauf von Signalen mittels eines Oszilloskops verfolgt. Doch vor allem in der Nachrichtentechnik ist die Betrachtung von Signalen im Frequenzbereich von Bedeutung. Um eine solche Analyse durchzuführen, wird ein Spektrumanalysator verwendet. Die theoretischen Grundlagen zur Bestimmung eines Signalspektrums werden in diesem Kapitel dargestellt. Hohe Ansprüche bestehen bei Spektrumanalysatoren vor allem in Bezug auf das Eigenrauschen, die Dynamik, den messbaren Frequenzbereich, den Funktionsumfang und die Messgeschwindigkeit. Inwiefern diese Anforderungen für die angestrebte Anwendung relevant sind und in welchem Maß diese durch die verwendete Hardware und die programmierte Software erfüllt werden, wird im Laufe der Arbeit gezeigt. Die theoretischen Grundlagen zur Softwareentwicklung basieren auf den nachfolgenden Ausführungen.

4.2 Signaltheorie zur Berechnung eines Signalspektrums

Allgemein wird ein Signal über die Zeit dargestellt. Diese Art der Darstellung liegt allgemein bei Oszilloskopen vor. Ein sinusförmiges Zeitsignal lässt sich aber auch als rotierender Zeiger in der komplexen Zahlenebene darstellen. Dabei wird die über die Zeit aufgetragene Amplitude des Zeitsignals auf die imaginäre Achse des Zeigers projiziert. Daraus ergibt sich für ein sinusförmiges Signal die komplexe mathematische Darstellung:

$$\begin{aligned}x(t) &= A \cdot \cos(2\pi f_0 t + \varphi) \\ &= \frac{A}{2} (e^{j(2\pi f_0 t + \varphi)} + e^{-j(2\pi f_0 t + \varphi)}) \\ &= \operatorname{Re}\{A \cdot e^{j(2\pi f_0 t + \varphi)}\}.\end{aligned}\tag{4.1}$$

Die Transformation zwischen Zeit- und Frequenzbereich leitet sich nach dem Theorem von *Fourier* ab, welches besagt, dass sich jedes im Zeitbereich periodische Signal durch eine unendliche Summe von Sinus- und Kosinustermen unterschiedlicher Amplitude und Frequenz darstellen lässt. Für periodische Signale mit der Periode $T_0 = 1/f_0$ ergibt sich die komplexe Fourierreihe

$$\underline{s}(t) = \sum_{n=-\infty}^{\infty} \underline{c}_n \cdot e^{j2\pi n f_0 t} \quad (4.2)$$

mit

$$\underline{c}_n = \frac{1}{T} \int_{-T/2}^{+T/2} \underline{s}(t) \cdot e^{-j2\pi n f_0 t} dt. \quad (4.3)$$

dem komplexen Fourier-Koeffizienten [17, S. 27].

Die Abbildung 4.1 zeigt die Annäherung eines Rechteckimpulses durch eine endliche Fourierreihe und deren einzelne Summanden, welche entsprechend der Formel 4.2 überlagert wurden.

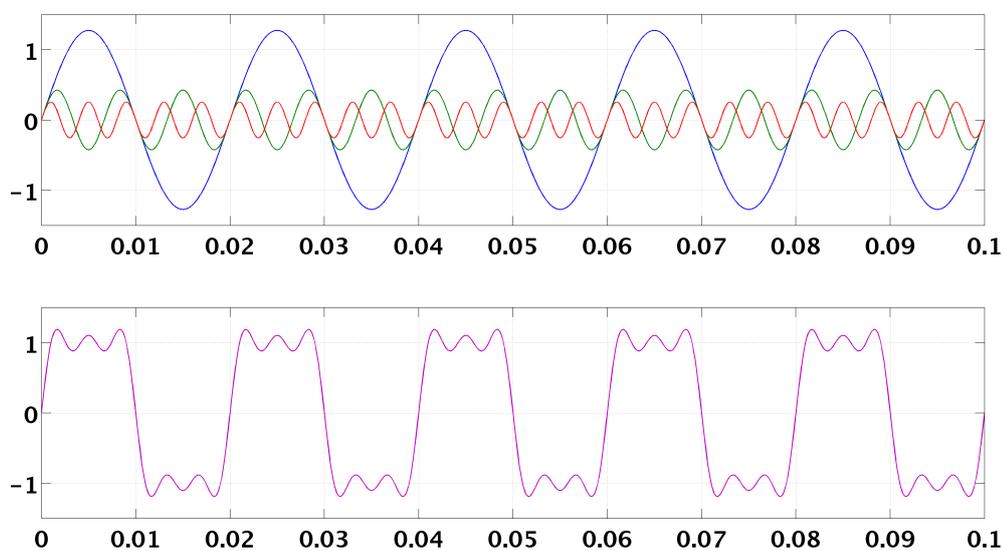


Abbildung 4.1: Annäherung eines Rechteckimpulses durch eine endliche Fourierreihe

Die Umrechnung zwischen Zeit- und Frequenzbereich für allgemeine Zeitsignale erfolgt mittels der Fouriertransformation. Diese stellt eine eindeutige Verknüpfung zwischen einem Zeitsignal und dem entsprechenden komplexen Fourierspektrum her. Für die Hinbeziehungsweise Rücktransformation gilt:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt \text{ bzw.} \quad (4.4)$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{j2\pi ft} df. \quad (4.5)$$

Für einige Signale lässt eine geschlossene Form der Fouriertransformierten finden. Mit Hilfe von Formel 4.4 ergibt sich beispielsweise für ein sinusförmiges Signal mit der Frequenz f_0 der Zusammenhang:

$$e^{j2\pi f_0 t} \xrightarrow{\mathcal{F}} \delta(f - f_0) \quad (4.6)$$

mit $\delta(f)$, dem Dirac-Impuls [23, S. 10].

Daraus folgt, dass für periodische Signale jeder einzelne Term der Summe aus Formel 4.2 eine Linie im Spektrum erzeugt. Es entsteht somit ein diskretes Linienspektrum. In der Praxis sind Signale jedoch nur selten periodisch. Für diesen Fall muss das Spektrum mithilfe der Formel 4.4 berechnet werden. Vor allem für stochastische Signale, wie zum Beispiel Rauschen, existiert keine Lösung. Das Spektrum muss dann durch numerische Verfahren aus konkreten Datensätzen berechnet werden.

4.3 IQ-Demodulation und Abtastung von Signalen

In der Praxis wird das Spektrum eines Zeitsignals mittels digitaler Signalverarbeitung berechnet. Dazu wird das analoge Eingangssignal abgetastet und anschließend quantisiert. Da die Quantisierung aus dem zeitdiskreten Abtastsignal ein zusätzlich wertdiskretes Signal erzeugt, kommt es hierbei zum Verlust von Informationen aufgrund der endlichen Bittiefe. Dieser Fehler schlägt sich in einem unvermeidbaren Quantisierungsrauschen nieder. Zudem begrenzt die endlich schnelle Abtastung die maximal erfassbare Bandbreite eines Signals. Um Aliasing-Effekte zu vermeiden, muss das Signal vor der Abtastung durch ein Filter in seiner Bandbreite begrenzt werden. Für die Abtastfrequenz eines Tiefpass-Signals gilt nach dem Theorem von *Shannon*, dass diese mindestens doppelt so groß sein muss, wie die maximale Frequenz des Eingangssignals:

$$f_{A, TP} \geq 2 \cdot f_{in,max}. \quad (4.7)$$

Für diese Annahme ist jedoch ein Filter mit einer unendlich steilen Flanke nötig, da das Spektrum im Bereich $|f| > f_{in,max}$ bandbegrenzt sein muss. Daher wird in der Praxis eine Abtastfrequenz gewählt, die deutlich höher als das Doppelte der maximalen Signalfrequenz ist. Die Signale, die in dieser Arbeit verarbeitet werden, sind jedoch keine Tiefpass-signale, sondern hochfrequente Bandpasssignale. Um eine Abtastung dieser Signale zu ermöglichen, wird mit Hilfe eines IQ-Demodulators (siehe Abbildung 4.2) das komplexe, äquivalente Tiefpasssignal der hochfrequenten Eingangssignale gebildet. Die Bandpass-mittelfrequenz wird dabei auf die Frequenz Null verschoben. Das durch dieses Verfahren gewonnene äquivalente Basisbandsignal wird auch Komplexe Einhüllende genannt. Besitzt das Bandpasssignal eine unsymmetrische spektrale Verteilung um den Frequenzursprung (siehe Abbildung 4.3), so wird das zum äquivalenten Tiefpasssignal gehörige Zeitsignal komplex und besitzt somit einen **I**n-Phase- und einen **Q**uadratur-Anteil [22, S. 25].

Die Abtastrate des IQ-demodulierten Signals muss entsprechend für den I- und Q-Kanal mindestens doppelt so hoch sein, wie die höchste Frequenz der Signale nach der analogen IQ-Tiefpassfilterung:

$$f_{A, \text{IQ-Demod.}} \geq 2 \cdot f_{\text{IQ,max.}} \quad (4.8)$$

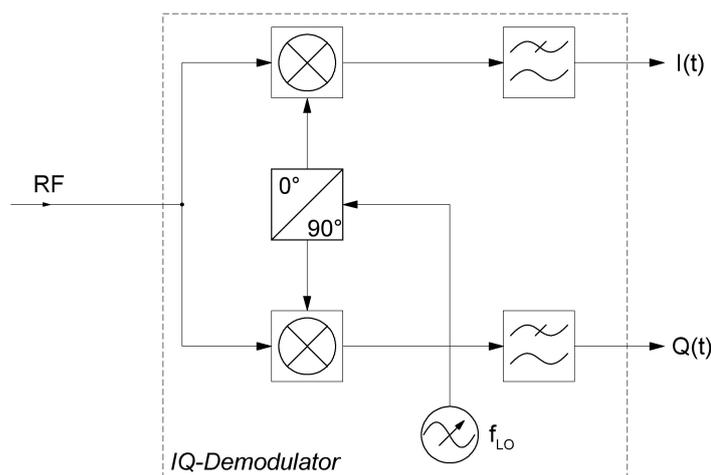


Abbildung 4.2: IQ-Demodulator zur Bildung des komplexen, äquivalenten Tiefpasssignals

Da die IQ-Tiefpassfilterung durch Filter mit endlicher Flankensteilheit und durchgeführt wird, muss die Abtastrate deutlich höher gewählt werden als das Doppelte der höchsten Frequenz des I- beziehungsweise Q-Signal. Beispielsweise wird in den meisten DVB-T-Empfängern eine Abtastrate von 28,8 MHz gewählt bei einer Ausgangsbandbreite von maximal 3,2 MHz. Entsprechend unerwünschte Signalanteile durch die endliche Flankens-teilheit der Tiefpassfilter werden anschließend digital gefiltert und durch ein Resampling die Abtastfrequenz an die Basisbandbreite von beispielsweise 3,2 MHz angepasst [10, S. 411].

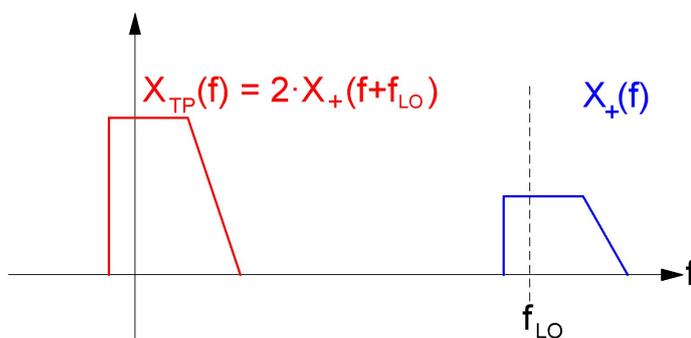


Abbildung 4.3: Komplexes, äquivalentes Tiefpasssignal eines unsymmetrischen Bandpass-signals

4.4 Anwendung der Theorie zur Berechnung breitbandiger Spektren

Um die oben beschriebenen Theorie zur Berechnung eines Spektrums zu untersuchen wird bei dieser beispielhaften Anwendung ein Bandpasssignal betrachtet, welches in einem Frequenzbereich von 200 bis 700 MHz liegt. Wie zuvor gezeigt wurde, begrenzt unter anderem die Flankensteilheit der Tiefpassfilter des IQ-Demodulators sowie die Abtastrate des A/D-Wandlers die maximal erfassbare Bandbreite eines System zum Beispiel auf 2 MHz (siehe Abbildung 4.4). Daraus folgt, dass das breitbandige Spektrum in einzelnen Teilspektren erfasst und anschließend zusammengesetzt werden muss. Aufgrund dieser Zusammensetzung des Spektrums ergeben sich Anforderungen an das System, welche nachfolgend diskutiert werden.

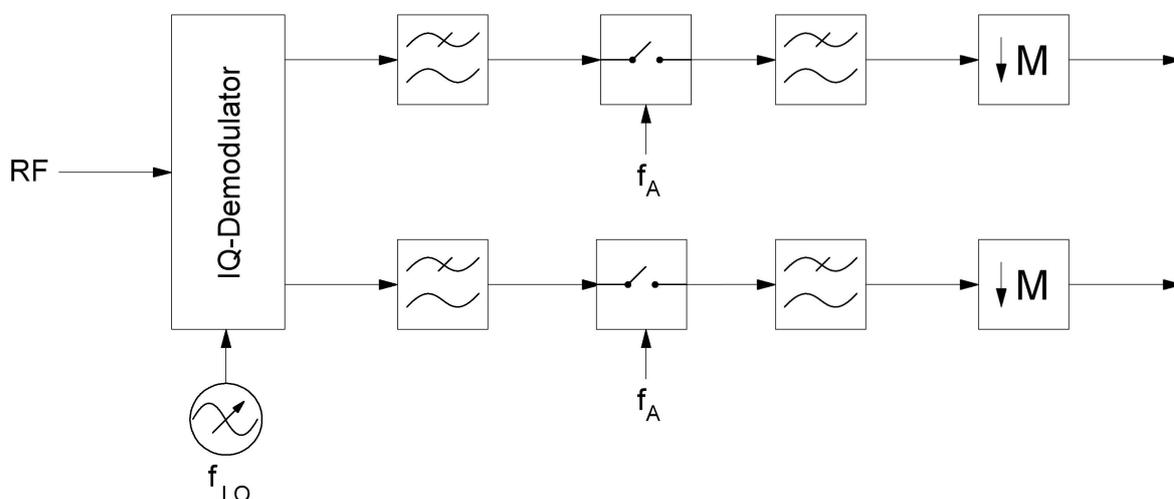


Abbildung 4.4: Signalverarbeitungskette zur Bestimmung eines schmalbandigen Teilspektrums

Die Abbildung 4.5 zeigt ein reelles Bandpassspektrum zwischen 200 und 700 MHz. Solche bandpassgefilterten Nutzrauschsignale werden auch in der späteren Anwendung als Eingangssignal dienen. Beispielhaft soll der rot markierte Bereich des Spektrums zwischen 500 und 550 MHz erfasst werden, da in diesem Bereich zum Beispiel eine Störung vermutet wird. Dieses Spektrum wird mit einem System entsprechend der Abbildung 4.4 ermittelt, welches am Ausgang eine maximale Bandbreite von 2 MHz erfasst.

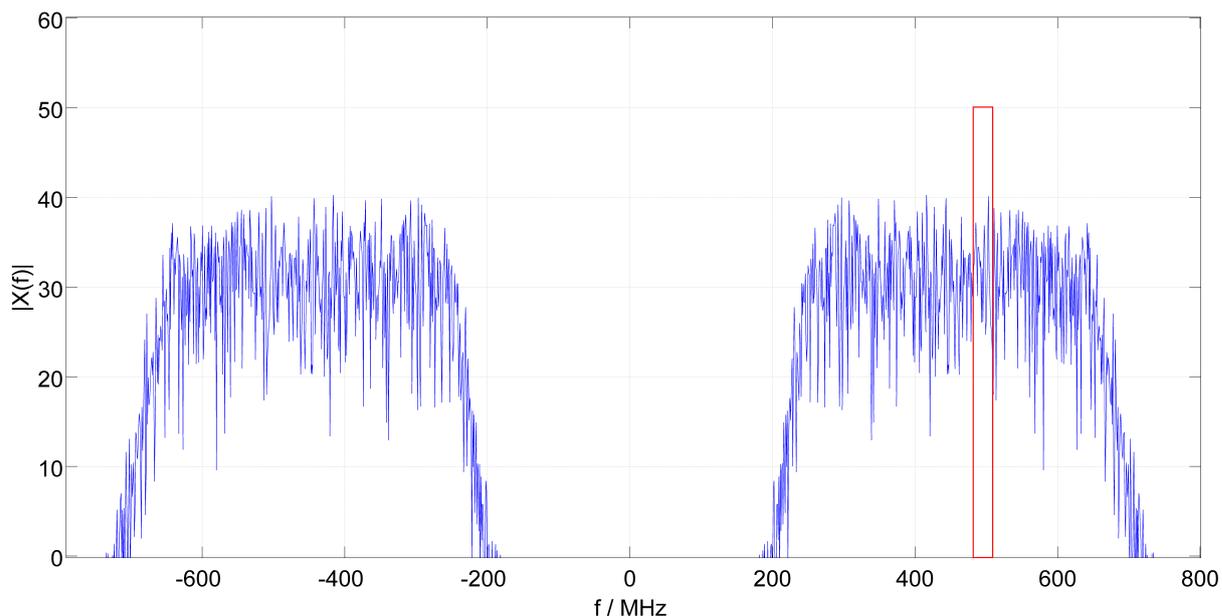


Abbildung 4.5: Beispielhaftes Betragsspektrum eines bandpassgefilterten Nutzrauschsignals

Die Signale werden dazu mittels eines IQ-Demodulators für eine feste LO-Frequenz ins Basisband gemischt. Für die Erfassung des angegebenen Frequenzbereichs liegt die LO-Frequenz für das erste Teilspektrum aufgrund der maximalen Bandbreite des Systems von 2 MHz bei 501 MHz. Die Grenzfrequenz der anschließenden Tiefpassfilter liegt beispielsweise bei 1 MHz mit einer endlichen Flankensteilheit sowie Sperrdämpfung. Das Ergebnis des IQ-demodulierten und analog tiefpassgefilterten Signals ist in Abbildung 4.6 zu sehen.

Die IQ-demodulierten Signale werden anschließend mit einer Abtastfrequenz von 28,8 MHz abgetastet. Die vorhergegangene Tiefpassfilterung hat die Signale soweit bandbegrenzt, dass bei dieser Abtastrate kein Aliasing entsteht, da das Signal oberhalb von etwa 1,5 bis $f_A/2 = 14,4$ MHz gleich Null ist. Die Abtastung führt zu einer Periodisierung des Spektrums mit der Abtastfrequenz von 28,8 MHz.

Das Spektrum enthält jedoch außerhalb der angestrebten Bandbreite von 2 MHz, also oberhalb von $|f| = 1$ MHz, weitere unerwünschte Spektralanteile, wie es deutlich in Abbildung 4.6 zu erkennen ist. Würde dieses Teilspektrum für eine Zusammensetzung des Gesamtspektrums genutzt werden, würde es zu einer unerwünschten Überlappung von spektralen Anteilen kommen und spektrale Fehler erzeugen.

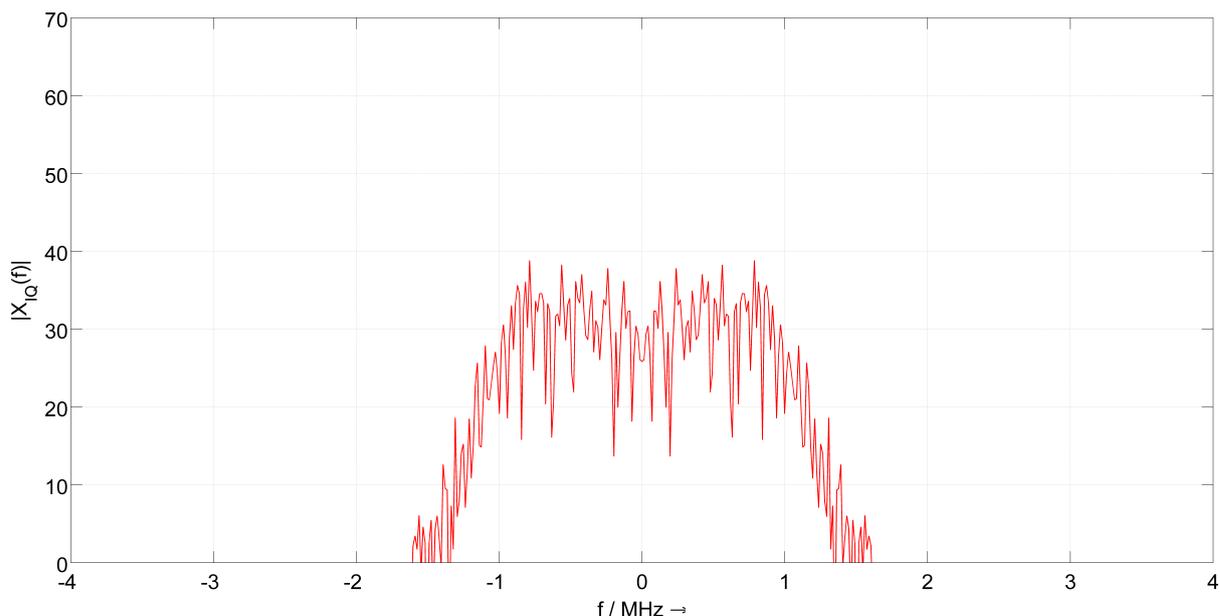


Abbildung 4.6: Spektrum des IQ-demodulierten Nutzrauschsignals nach einer Filterung mit einem Tiefpass mit einer Grenzfrequenz von 1 MHz und flacher Flanke

Damit sich die Teilspektren nicht überlappen, werden diese mittels digitaler FIR-Filter erneut gefiltert. Da digitale Filter deutlich steilere Flanken bei einer gleichzeitig hohen Sperrdämpfung erlauben, werden unerwünschte Spektralanteile, welche durch die endliche Flankensteilheit der analogen Tiefpassfilter zustande kommen, stärker unterdrückt. Durch ein Resampling wird anschließend die Abtastfrequenz auf 2 MHz umgesetzt. Das hat wiederum eine Periodisierung des Spektrums mit 2 MHz zur Folge (siehe Abbildung 4.7). Die Periodisierung ist durch die Punkte auf der rechten beziehungsweise linken Seite der Abbildung angedeutet. Wie zu erkennen ist, überlagern sich jedoch weiterhin Anteile, welche durch die digitale Filterung oberhalb von $|f| = 1$ MHz nicht entfernt werden konnten. Es kommt zu Aliasingfehlern.

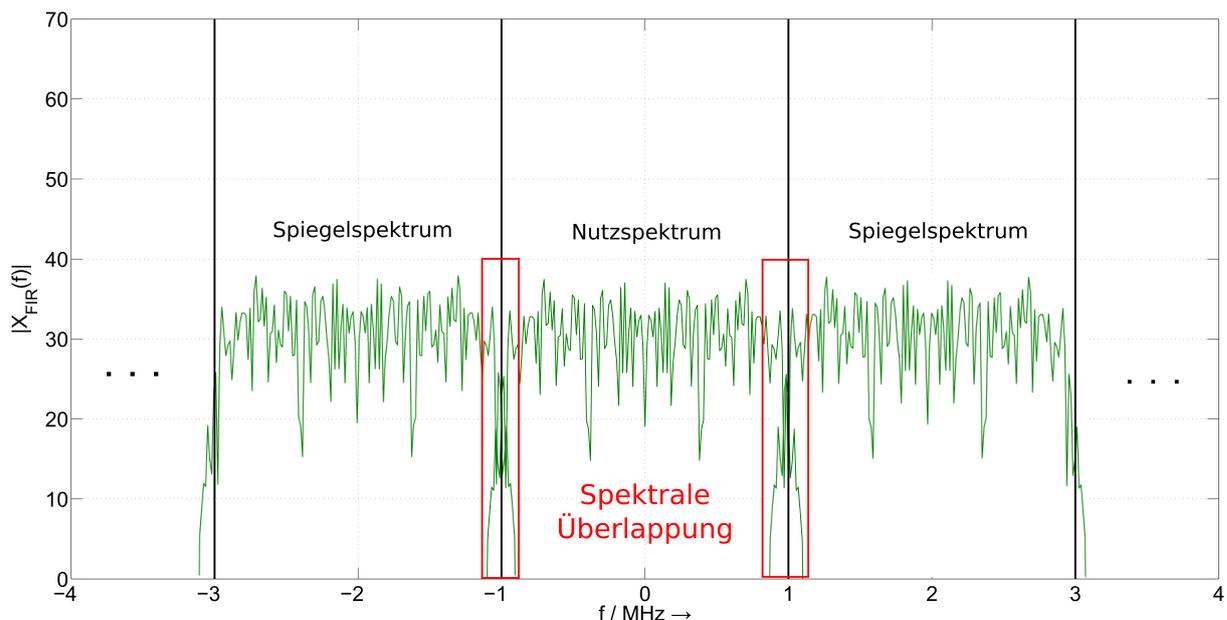


Abbildung 4.7: Spektrum des digital gefilterten IQ-demodulierten Signals nach der Umsetzung der Abtastrate auf 2 MHz

Dieses einzelne 2 MHz breite Teilspektrum, welches einen Teil des in Abbildung 4.5 rot markierten Spektrums darstellt, wird mit anderen Teilspektren zusammengeführt, welche nach der selben Methode bei unterschiedlichen LO-Frequenzen des IQ-Demodulators bestimmt wurden. Daraus ergibt sich, dass an den Bandkanten aufgrund des Aliasings bei der Umsetzung der Abtastfrequenz mit spektralen Fehlern zu rechnen ist. Zudem ist zu bedenken, dass der Lokaloszillator eine Frequenzgenauigkeit besitzt. Das heißt, dass Teilspektren nicht im exakten Abstand von beispielsweise 2 MHz ermittelt werden können (siehe Abbildung 4.8). Dabei können, wie im oberen Fall der Abbildung dargestellt, Überlappungen oder, bei einer Abweichung der Lokaloszillatorfrequenz nach oben, Lücken bei der Zusammenführung von Teilspektren entstehen.

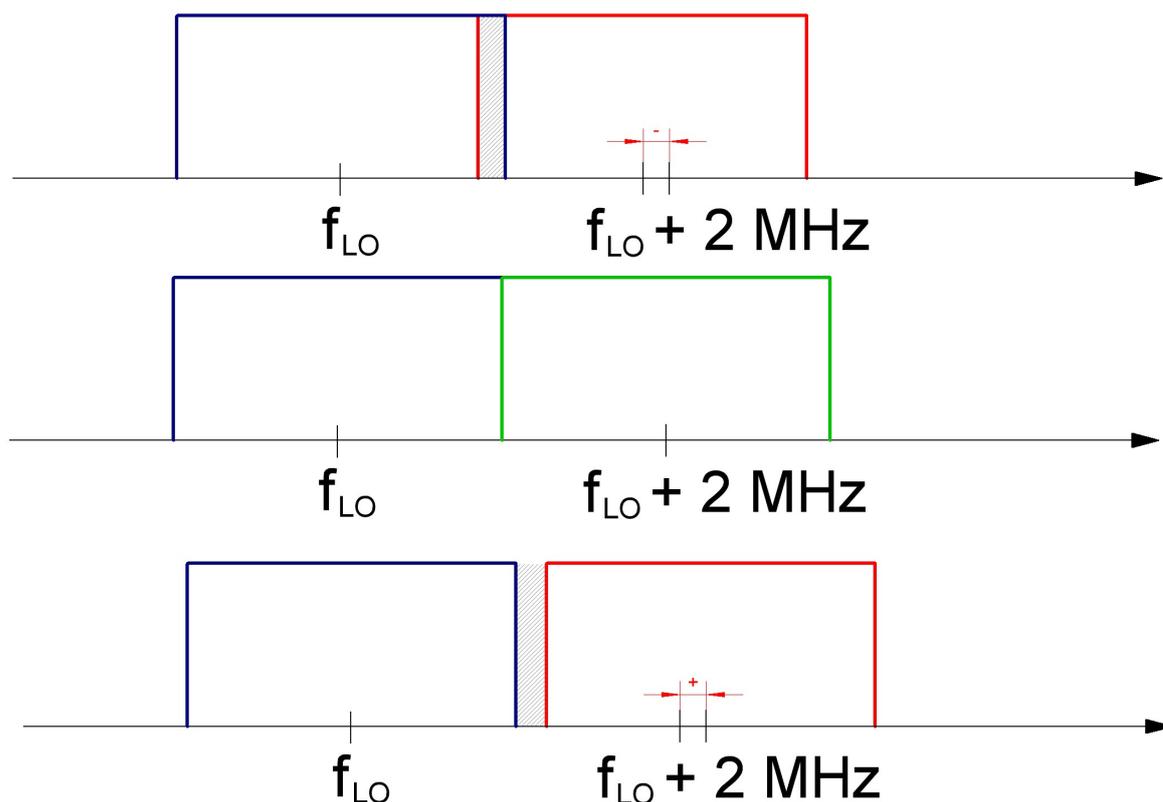


Abbildung 4.8: Gegenseitiger Versatz von Teilspektren aufgrund einer Frequenzungenauigkeit des Lokaloszillators

4.5 Berechnung der diskrete Fouriertransformation

Die Fouriertransformation ist eine Zuordnung eines reellen Zeitsignals $x(t)$ zu einer komplexen Funktion $X(f)$ im Frequenzbereich. Die Berechnungsformel 4.4 macht dabei ebenfalls deutlich, dass für eine exakte Berechnung des Fourierspektrums das Signal über eine unendliche Zeitdauer betrachtet werden muss. Praktisch steht jedoch nur eine begrenzte Beobachtungszeit zur Verfügung. Daher wird zur spektralen Analyse nur ein Ausschnitt des Zeitsignals betrachtet. Die Berechnung des Spektrums aus einem solchen kurzen Ausschnitt eines Zeitsignals wird Kurzzeit-Spektralanalyse genannt.

Um aus dieser begrenzten Anzahl an Abtastwerten des Zeitsignals das diskrete Signalspektrum zu berechnen, wird die diskrete Fouriertransformation (DFT) verwendet. Diese lässt sich aus der Fouriertransformation herleiten [17, S. 164]. Für ein mittels eines Rechteckfensters auf N Abtastwerte begrenztes Signal ergibt sich die Formel zur Berechnung der diskreten Fouriertransformierten zu:

$$\underline{X}_k = \sum_{n=0}^{N-1} x(n \cdot T_A) \cdot e^{-j2\pi k \frac{n}{N}}. \quad (4.9)$$

Die Auswertefrequenzen, also die diskreten Frequenzen, für die die DFT berechnet wird, liegen dabei um den Abstand

$$\Delta f = \frac{f_A}{N} = \frac{1}{T_A \cdot N} \quad (4.10)$$

auseinander. Der Abstand zwischen zwei spektralen Komponenten ist, um diese als getrennte Auswertefrequenzen wahrnehmen zu können, vom Betrachtungszeitraum $T_A \cdot N$ abhängig. Die spektrale Auflösung steigt daher mit der Beobachtungsdauer.

Die DFT liefert jedoch nur dann eine exakte Lösung liefert, wenn das Signal periodisch in $T_A \cdot N$ und auf $f_A/2$ bandbegrenzt sein. Das Ergebnis ist dann ein DFT-Vektor bestehend aus den exakten komplexen Fourierkoeffizienten des periodischen Signals mit dem Vorfaktor N .

Aus der zeitlichen Begrenzung der Signale folgt jedoch für eine Beobachtungsdauer, die nicht ein ganzzahliges Vielfaches der Periodendauer des Signals ist, eine Verschmierung des Spektrums, auch *Leakage* genannt. Wird ein Zeitsignal mit einem Rechteckfenster auf N Abtastwerte begrenzt, entspricht dies einer Faltung des Spektrums mit einer *si*-Funktion. Dieser Effekt wird in Abbildung 4.9 deutlich. Die obere Hälfte zeigt das Fourierspektrum, bei dem die Beobachtungsdauer ein ganzzahliges Vielfaches der Periodendauer der sinusförmigen Schwingung ist. Während für die Berechnung des Fourierspektrums in der unteren Hälfte eine Fensterbreite gewählt wurde, die nicht einem ganzzahligen Vielfachen der Periodendauer entspricht. Es ist deutlich zu erkennen, dass das Spektrum in diesem Fall nach der Fensterung verschmiert wird [17, S. 168].

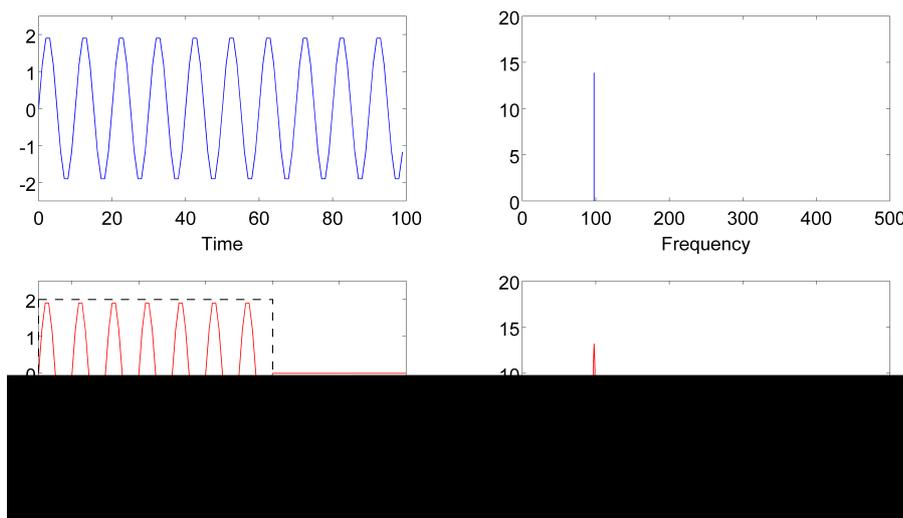


Abbildung 4.9: Wirkung der Fensterung eines Signals auf das Spektrum; oben: zeitlicher Verlauf und Spektrum des ungefensterten sinusförmigen Signals; unten: zeitlicher Verlauf und Spektrum des gefenstereten sinusförmigen Signals

Aufgrund dieser Verschmierung des Spektrums ist keine Bandbegrenzung mehr vorhanden. Die Diskretisierung dieses verschmierten Spektrums im Abstand f_A/N führt zum Ergebnis der diskreten Fouriertransformation. Aufgrund der Abtastung des kontinuierlichen Spektrums kommt es zu einer Periodisierung im Zeitbereich. Das diskrete Spektrum stellt somit eine Approximation der Fouriertransformierten im Bereich von 0 bis $f_A/2$ dar [17, S. 169].

Durch die Wahl eines anderen Fensters kann der oben beschriebene Leckeffekt minimiert werden. So blenden Fenster, wie zum Beispiel das Blackman-Harris- oder das Flattop-Fenster, die Zeitfunktion ein beziehungsweise aus. Aufgrund dieser Eigenschaft ist die spektrale Funktion der Fensterfunktion um die Mittenfrequenz sehr schmal und besitzt außerhalb eine hohe Dämpfung. Wie zuvor gezeigt wurde, wird das Spektrum des zeitlich begrenzten Signals mit dem Spektrum der Fensterfunktion gefaltet. Die Faltung mit einer solchen Fensterfunktion führt dazu, dass der Leckeffekt verkleinert und die Bandbegrenzung verbessert wird. Daraus folgt ein geringerer Aliasingfehler bei der Bildung der DFT und somit eine bessere Approximation der Fouriertransformierten.

Zur performanteren Berechnung des diskreten Fourierspektrums kann diese mit $N = 2^n$ Abtastwerten durchgeführt werden. Diese Art der Berechnung der DFT wird Fast Fourier Transformation genannt. Der Rechenaufwand fällt aufgrund der Berechnung mit $N = 2^n$ Abtastwerten mittels des Teile-und-herrsche Verfahrens von einer quadratischen Komplexität auf eine logarithmische Komplexität ab [43]. Aufbauend auf diesen Grundlagen und nach der Art des Beispiels in Abschnitt 4.4 wird in dieser Arbeit ein Spektrometer mittels eines Software Defined Radio zur spektralen Erkennung von RFI entwickelt [17, S. 172].

5 Untersuchung von SDR-Hardwareinterfaces

Dieses Kapitel untersucht die Eigenschaften eines DVB-T-Sticks sowie eines National Instruments USRP als RF-A/D-Einheit eines SDR. Dazu wird anfangs allgemein erläutert, warum ein DVB-T-Stick als SDR verwendet werden kann. Anschließend werden die technischen Eigenschaften von zwei verschiedenen DVB-T-Sticks und des USRP messtechnisch ermittelt und die Eignung für diese Anwendung betrachtet.

Im Jahr 2012 hat der finnische Student und Entwickler Antti Palosaari entdeckt, dass DVB-T-Sticks, die für den Empfang des europäischen Fernsehstandards am Computer konzipiert sind, auch als Software Defined Radio genutzt werden können. Ausgestattet mit einem RTL2832U IC des taiwanischen Herstellers Realtek kann ein solcher DVB-T-Stick neben dem demodulierten DVB-T-Signal auch reine digitale IQ-Daten über einen breiten Frequenzbereich über einen USB-Anschluss liefern. Der mögliche Frequenzbereich variiert jedoch je nach dem verbautem Tuner, welche die RF-Einheit eines solchen SDR darstellt. Als Bezeichnung für solche USB-Sticks hat sich, in Bezug auf den verbauten Realtek IC, der Begriff RTL-SDR gefestigt, welcher auch in dieser Arbeit verwendet wird [35].

Neben solchen DVB-T-Sticks wurde ein USRP-2920 (Universal Software Radio Peripheral) der Firma National Instruments verwendet. Dieses SDR-Interface bietet laut Hersteller eine breitbandige und schnelle Signalverarbeitung und kann dynamisch in vielen Bereichen der Forschung, Entwicklung und Messtechnik verwendet werden. Durch einen modularen Aufbau und den integrierten Xilinx FPGA sind je nach Konfiguration Anwendungen zwischen DC und 6 GHz möglich. Die verwendete Hardware deckt einen Frequenzbereich zwischen 50 und 2200 MHz ab. Die maximale Bandbreite von 20 MHz ist zudem von Vorteil für eine schnelle Berechnung eines breitbandigen Spektrums. Mit zusätzlichen Referenzeingängen zur Zeitsynchronisation kann außerdem mit einer höheren Frequenzgenauigkeit gerechnet werden. Die Quantisierung der Abtastsignale mit einer höheren Bittiefe bringt eine bessere Qualität der Abtastsignale und verkleinert das Quantisierungsrauschen aufgrund der genaueren Umsetzung. Der Preis des Gerätes ist mit 2500 € deutlich höher als der des RTL-SDR. Eine Gegenüberstellung zwischen einem RTL-SDR und dem USRP-2920 ist in der nachfolgenden Tabelle 5.1 zu finden. Im Folgenden wird untersucht, ob sich die Verwendung und die Investition in ein solches Gerät lohnt. Dazu werden verschiedene Hardwareparameter beider Interfaces untersucht und gegenübergestellt [29].

Parameter	RTL-SDR DVB-T-Stick	National Instruments USRP-2920
Frequenzbereich	E4000: 52 - 2200 MHz (Lücke zwischen 1100 und 1200 MHz) R820T: 24 - 1766 MHz	50 - 2200 MHz
Momentan-Bandbreite	3,2 MHz	20 MHz
ADC-Auflösung	8 Bit	14 Bit
theoretischer SNR_{\max}	49,92 dB	86,04 dB
Kanäle	1	2
Referenzeingänge	keine	10 MHz und 1pps
Frequenzgenauigkeit	20 - 100 ppm	2,5 ppm
Schnittstelle	USB 2.0	Gigabit-Ethernet
IIP3	E4000: -70 dBm R820T: -40 dBm	0 dBm
Programmierschnittstelle	GNU Radio	GNU Radio, LabView, MATLAB
Abmessungen	8 x 3 x 1,5 cm	22 x 16 x 5 cm
Preis	ca. 20 - 30 €	ca. 2500 €

Tabelle 5.1: Vergleich zwischen einem RTL-SDR und einem NI-USRP 2920 [29]

5.1 Hardwareaufbau der verwendeten SDR-Interfaces

5.1.1 DVB-T-Sticks als SDR-Hardware

Die Abbildung 5.1 zeigt den vereinfachten Aufbau eines DVB-T-Sticks mit einem RTL2832U IC. Die Eingangssignale werden über einen Antennenanschluss in den Stick eingespeist. Die erste Stufe im Empfänger (Tuner, RF-Einheit) wird für die analoge HF-Signalverarbeitung benötigt. Das bedeutet, dass dort unter anderem die Eingangssignale gefiltert und verstärkt werden. Zudem findet hier eine Frequenzumsetzung statt, welche aus dem HF-Eingangssignal das äquivalente, komplexe Tiefpasssignal (Direktkonversion) oder ein Zwischenfrequenzsignal bildet.

Die analogen, heruntergemischten Signale des Tuner-Ausgangs werden anschließend an den Realtek RTL2832U IC (siehe Abbildung 5.2) übertragen. Dieser wandelt die analogen Signale des Tuners mit einer Abtastrate von 28,8 MHz in 8-Bit-Samples um. Der IC unterstützt die Verarbeitung von Direktkonversionssignalen sowie Signalen bei einer Zwischenfrequenz von 36,125 oder 4,57 MHz mit einer Bandbreite von bis zu 8 MHz. Im Tuner auf eine Zwischenfrequenz gemischte Signale werden nach der A/D-Wandlung digital IQ-demoduliert und tiefpassgefiltert. Über einen Resampler und einen Dezimierer wird die Abtastrate auf maximal 3,2 MHz umgesetzt und die Bandbreite durch ein entsprechendes FIR-Filter begrenzt. Die Abtastrate ist dabei variabel per Software einstellbar.

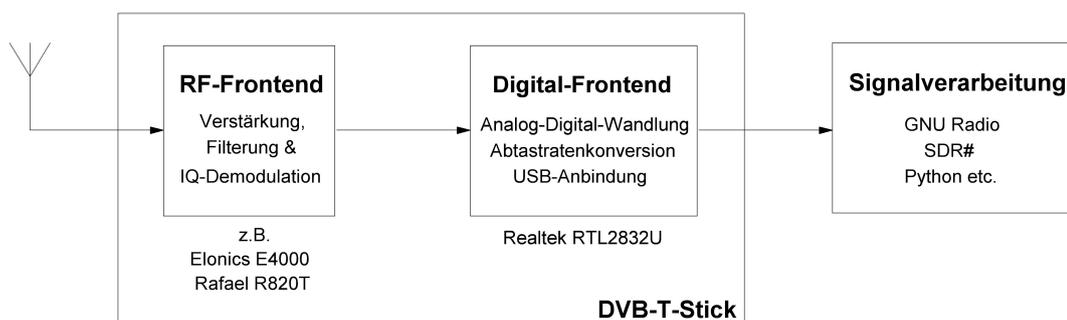


Abbildung 5.1: Vereinfachtes Blockschaltbild eines DVB-T-Sticks mit RTL2832U IC

Über einen USB-Controller werden die komplexe IQ-Samples an den Rechner übertragen. Die restlichen Blöcke werden nicht genutzt, da diese Teil der OFDM-Demodulation sind und zur Rekonstruktion der Video- und Audio-Signale des DVB-T-Streams dient.

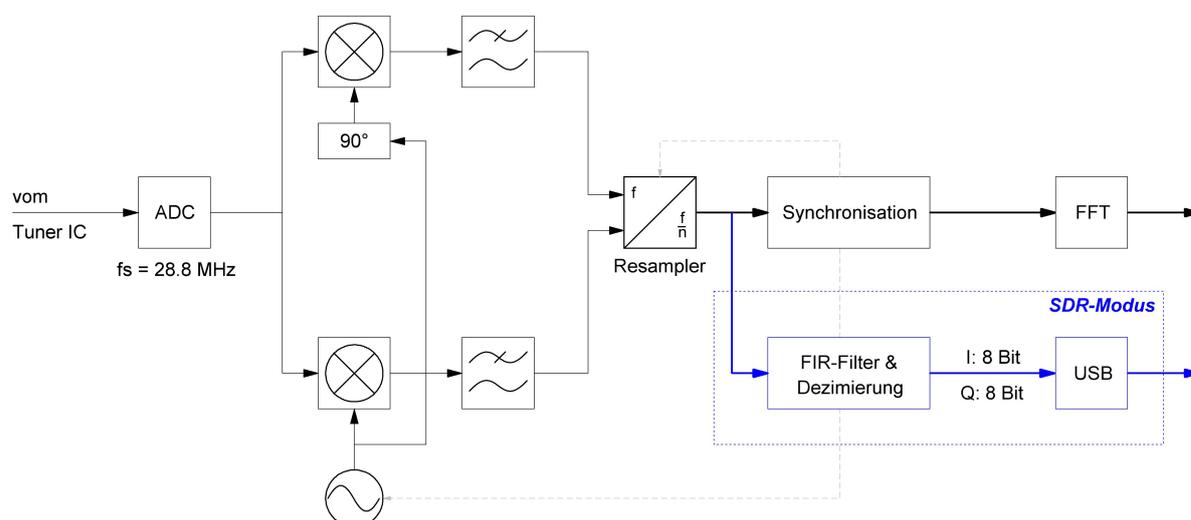


Abbildung 5.2: Blockschaltbild des RTL2832U ICs [34]

Die A/D-Wandlung begrenzt grundsätzlich den dynamischen Bereich des Empfängers. Bei voller Aussteuerung des A/D-Wandlers wird unter der Voraussetzung, dass die Quantisierungskennlinie linear verläuft, ein maximales Signal-Rauschverhältnis zwischen einem Nutzsignal und dem Quantisierungsrauschen von

$$\text{SNR}_{\max}/\text{dB} = 6,02 \cdot \text{Anzahl Bits/Sample} + 1,76 = 6,02 \cdot 8 + 1,76 = 49,92 \quad (5.1)$$

erreicht [37].

5.1.1.1 Spezifikationen der verwendeten DVB-T-Sticks

Dieser Abschnitt stellt die Hardware zweier verschiedener in dieser Arbeit verwendeten DVB-T-Sticks gegenüber, die sich hauptsächlich im verbauten Tuner unterscheiden. Im Stick des Herstellers Terratec ist ein Elonics E4000 Tuner verbaut, während in einem Stick von GiXa Technology ein Rafael Micro R820T die HF-Signalverarbeitung und Frequenzkonversion übernimmt. Die prinzipielle Funktionsweise und der Aufbau beider DVB-T-Sticks ist im Gesamten nahezu identisch und wurde im vorhergegangenen Kapitel bereits prinzipiell erläutert. Elementare Unterschiede sind aber vor allem im einstellbaren Frequenzbereich und in der Signalumsetzung zu finden (siehe Tabelle 5.2).

Tuner	Parameter	Frequenzbereich	Frequenzkonversion
Elonics E4000		52 - 1100 MHz, 1200 - 2200 MHz	Direktkonversion
Rafael Micro R820T		24 - 1766 MHz	Zwischenfrequenzmischung

Tabelle 5.2: Vergleich verschiedener Tuner-Spezifikationen [30]

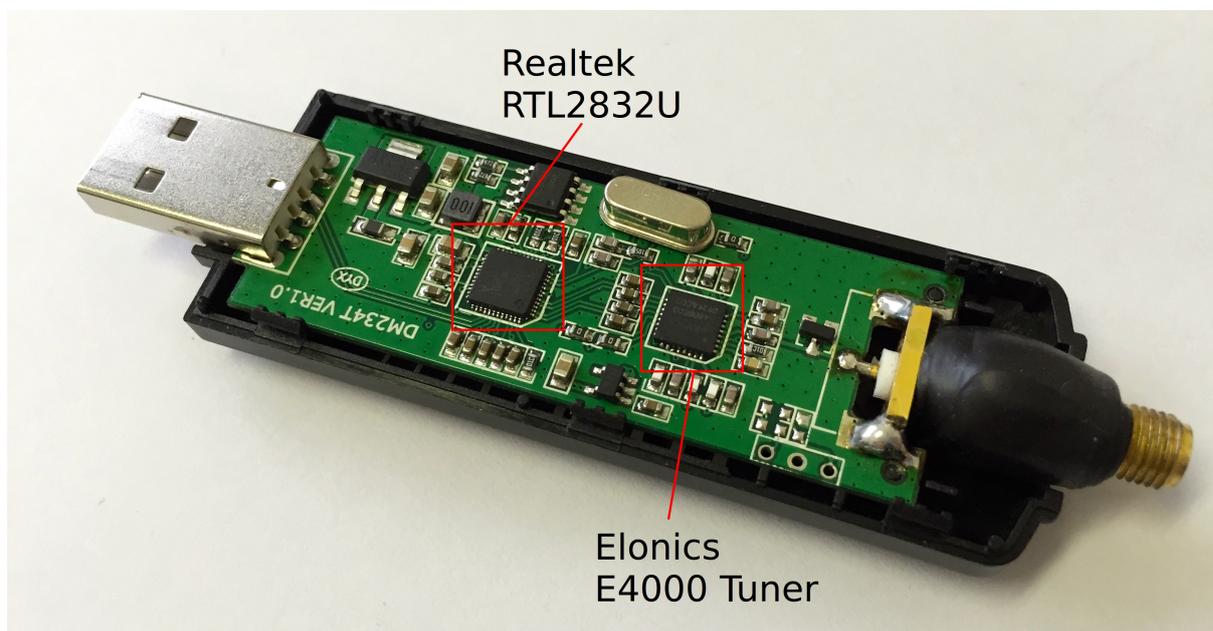


Abbildung 5.3: Auf einen SMA-Anschluss umgebauter Terratec DVB-T-Stick

5.1.1.2 Aufbau und Funktionsweise des RF-Frontends

Die bei beiden Sticks verbaute Antennenbuchse zur Einspeisung von Signalen wird durch eine SMA-Buchse ersetzt. Das erleichtert den Anschluss an Messgeräte und Signalgeneratoren sowie an das ZF-System, an dem der Stick später betrieben werden soll. Um die Buchse leichter erreichen zu können, wird zusätzlich eine SMA-Verlängerung aufgeschraubt (siehe Abbildung 5.3).

Die Abbildung 5.4 zeigt das Blockschaltbild des Elonics E4000 Tuners, welches aus dem Datenblatt des ICs entnommen wurde. Die Eingangssignale werden durch einen rauscharmen Vorverstärker verstärkt dessen Verstärkung später auch als "RF Gain" bezeichnet wird und in einem Bereich von 0 bis 35 dB per Software regelbar ist. Ein Leistungsdetektor bestimmt die Leistung des Ausgangssignals des Verstärkers. Dessen Ausgangssignal dient wiederum für eine automatische Verstärkungsregelungsschleife. Neben einer rauscharmen Verstärkung sollte diese Verstärkerstufe auch übersteuerungsfest sein, um eine optimale Vorverstärkung zu erzielen.

Um unerwünschte Störsignale bereits am Eingang zu eliminieren, wird das vorverstärkte Signal durch ein abstimmbares Bandpassfilter gefiltert. Dessen Grenzfrequenz wird mittels Software automatisch an die momentan gewählte Empfangsfrequenz angepasst. Auf der Grundlage der IQ-Demodulation wird aus dem Eingangssignal das komplexe, äquivalente Tiefpasssignal gebildet. Zur Erzeugung stabiler Mischsignale wird eine Phasenregelschleife, die ein 28,8 MHz Quarzoszillator zur Referenzsignalgenerierung nutzt, verwendet. Aufgrund der Direktkonversion entsteht am Ausgang der Mischer ein Gleichspannungsanteil. Eine Kompensation wird durch eine DC-Offset-Korrektur hinter jeder Mischstufe vorgenommen, deren Kompensationswerte digital eingestellt werden können.

Nach der IQ-Tiefpassfilterung wird das I- beziehungsweise Q-Signal durch eine weitere Verstärkerstufe verstärkt. Die analogen I- und Q-Signale werden anschließend an den RTL2832U IC übertragen. Wie bereits im vorhergegangenen Kapitel beschrieben, wird das IQ-Signal durch den RTL2832U digitalisiert, eine Abtastratenkonversion sowie eine Filterung vorgenommen. Die Samples werden dann über eine USB-Schnittstelle an den Rechner übertragen.

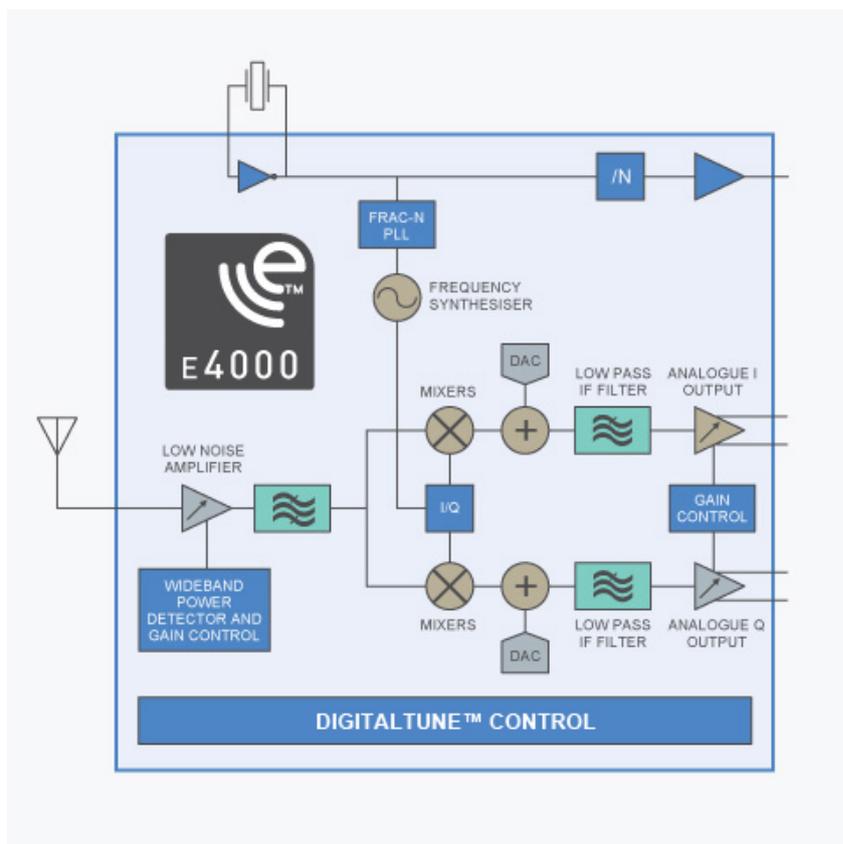


Abbildung 5.4: Blockschaltbild des Elonics E4000 Tuners [9, S. 4]

Das Blockschaltbild des Rafael Micro R820T Tuners aus dem Datenblatt (siehe Abbildung 5.5) zeigt, dass von der prinzipiellen Funktionsweise einige Unterschiede gegenüber dem E4000-Tuner erkennbar sind. Das RF-Signal wird auch hier durch einen LNA vorverstärkt. Ein erster Leistungsdetektor dient als Teil einer automatischen Verstärkungsregelung der Vorverstärkung. Nach einer HF-Bandpassfilterung wird das Eingangssignal auf eine feste Zwischenfrequenz umgesetzt. Diese liegt bei diesem Tuner bei 4,57 MHz mit einer Bandbreite von maximal 8 MHz. Das Signal wird durch ein entsprechendes Zwischenfrequenzfilter gefiltert. Nach einer weiteren Verstärkung der Signale wird das Zwischenfrequenzsignal an den RTL2832U IC übertragen. Zur exakten Frequenzumsetzung wird, wie auch beim E4000-Tuner, ein 28,8 MHz Quarzoszillator und eine Phasenregelschleife verwendet. Die Steuerung erfolgt über den I²C-Bus. Der RTL2832U IC bildet aus dem Zwischenfrequenzsignal nach der A/D-Wandlung das äquivalente, komplexe Basisbandsignal.

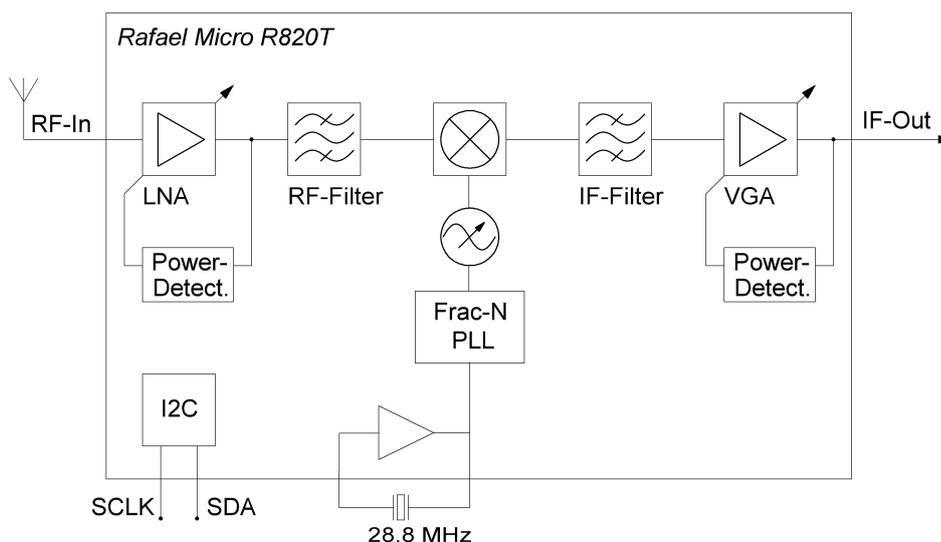


Abbildung 5.5: Blockschaltbild des Rafael Micro R820T Tuners [32, S. 7]

5.1.2 Hardwareaufbau des National Instruments USRP-2920

Die Abbildung 5.6 zeigt das Blockschaltbild des USRP-2920. Anhand dieser Darstellung werden nachfolgend die wichtigsten Elemente dieses Gerätes erläutert. Der prinzipielle Aufbau entspricht dabei dem im Themen einführenden Kapitel besprochenen idealen Aufbau eines SDR-Hardwareinterface in weiten Teilen und somit der Definition eines SDR durch die ITU-R.

Über einen der beiden Empfängereingänge werden die Empfangssignale über einen SMA-Anschluss in das USRP eingespeist. Mittels eines Umschalters kann der aktuelle Empfangskanal ausgewählt oder wahlweise der Sende-Modus geschaltet werden. Da für diese Anwendung nur der Empfang von Signalen von Bedeutung ist, wird nur dieser Zweig näher untersucht. Nach einer Vorverstärkung des Signals durch einen LNA werden die HF-Signale mittels eines IQ-Demodulators, der sich auf dem sogenannten Daughterboard des USRP befindet, in das Basisband umgesetzt. Nach einer Tiefpassfilterung werden die analogen IQ-Signale auf dem Motherboard digitalisiert. Im Anschluss findet eine Abstratenkonversion statt und die digitalen Signale werden über Ethernet an den PC gesendet. Dieser steuert die Hardware über diese Schnittstelle und verwendet ein spezielles Kontrollprotokoll.

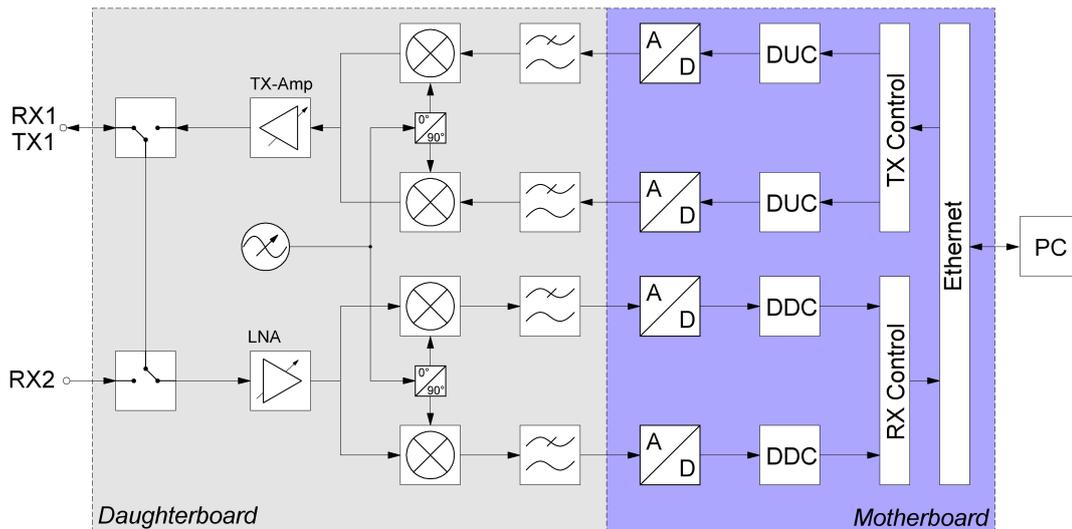


Abbildung 5.6: Blockschaltbild des National Instruments USRP-2920 [29]

5.2 Bestimmung des Eingangsreflexionsfaktors

Um eine optimale Abgabe der Leistung einer Quelle mit einem komplexen Innenwiderstand Z_Q an eine Last zu erreichen, muss die Lastimpedanz Z_L identisch dem konjugiert komplexen Innenwiderstand der Quelle sein. Das ZF-System des Max-Planck-Instituts und auch viele Signalgeneratoren und Messgeräte sind auf 50Ω angepasst. Das heißt, die maximale Leistungsübertragung ist gewährleistet, wenn die Eingangsimpedanz der SDR-Hardware ebenfalls 50Ω beträgt. Um die komplexe Eingangsimpedanz zu bestimmen, wird mithilfe eines vektoriellen Netzwerkanalysators das Verhältnis zwischen einer zum Eingang hinlaufenden Welle und der vom Eingang reflektierter Anteil gemessen. Daraus lässt sich auf die komplexe Impedanz des Eingangs schließen und auf die Güte der Anpassung. Doch zunächst werden einige Grundlagen zur Bestimmung des Eingangsreflexionsfaktors dargestellt.

5.2.1 Theoretische Grundlagen zur Messmethode

In der Hochfrequenztechnik wird zwischen der zu einem Bauteil *hinlaufenden Welle* a und der *rücklaufenden Welle* b unterschieden. Bei der Netzwerkanalyse breitet sich die hinlaufende Welle vom Analysator zum Bauteil hin aus. Die rücklaufende Welle bewegt sich entsprechend in die entgegengesetzte Richtung. Das Verhältnis aus hinlaufender und rücklaufender Welle wird als Reflexionsfaktor

$$\Gamma = \frac{b}{a} \quad (5.2)$$

bezeichnet.

Dieser Faktor ist im Allgemeinen komplex und kann direkt aus einer komplexen Impedanz z , welche auf den Wellenwiderstand Z_W normiert ist, berechnet werden:

$$\Gamma = \frac{z - 1}{z + 1}. \quad (5.3)$$

Dieser Faktor wird meist in einem Smithdiagramm eingetragen. Für die Vorwärtsmessung wird an Tor 2 eines Zweitorbauteils ein reflexionsfreier Abschluss angeschlossen. Das führt dazu, dass die von Tor 2 zu Tor 1 rücklaufende Welle $a_2 = 0$ wird. Für den Eingangsreflexionsfaktor S_{11} folgt damit

$$S_{11} = \frac{b_1}{a_1}. \quad (5.4)$$

Das Prinzip zur Messung des Eingangsreflexionsfaktors ist in Abbildung 5.7 für ein Zweitor dargestellt.

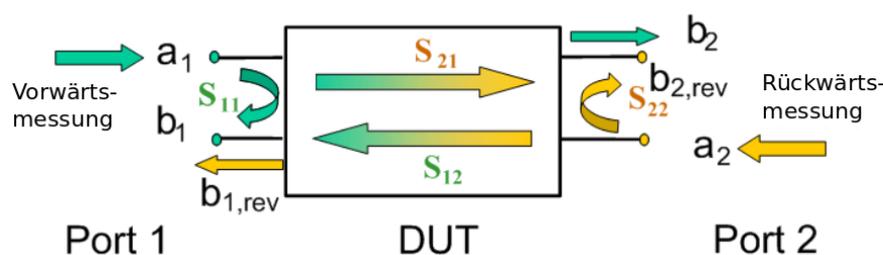


Abbildung 5.7: Prinzip zur Bestimmung der Streuparameter [18, S. 15]

Eine schematische Darstellung eines Netzwerkanalysators ist in Abbildung 5.8 für die Messung des Eingangsreflexionsfaktors eines Eintors zu sehen. Die Signalverarbeitung im Referenz- und Messkanal ist stichwortartig im Blockschaltbild festgehalten und gilt für beide Kanäle gleichermaßen. Nachfolgend wird die Messung dieses Reflexionsfaktors am Eingang der DVB-T-Sticks und des USRP gezeigt [18, S. 15-22].

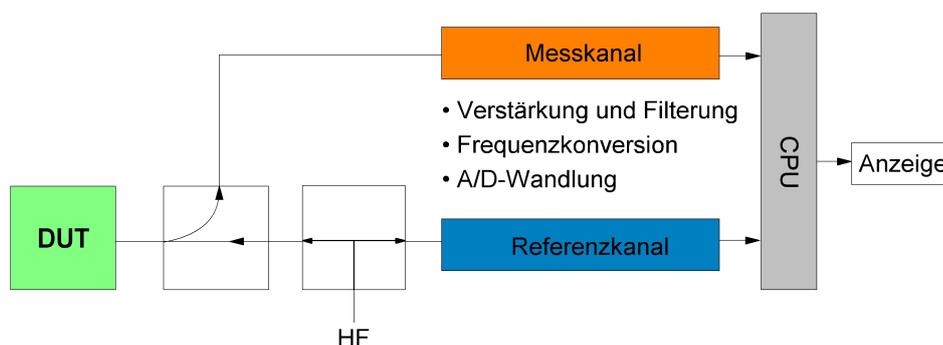


Abbildung 5.8: Schematischer Aufbau eines Netzwerkanalysators

5.2.2 Störemission und Eingangsreflexionsfaktor des E4000-Sticks

Vorbereitend zur eigentlichen Messung wird der Eingang des DVB-T-Sticks direkt an einen Spektrumanalysator angeschlossen. Das Ergebnis für den E4000-Stick ist in Abbildung 5.9 über einen Frequenzbereich von 0 bis 750 MHz zu sehen.

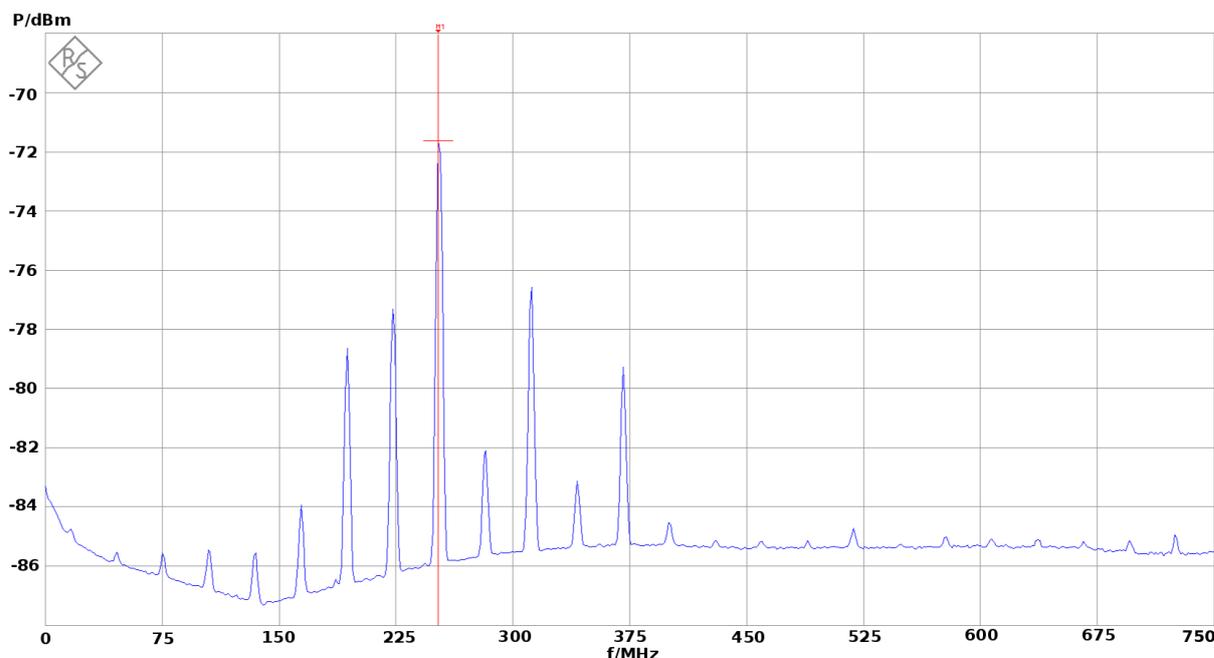


Abbildung 5.9: Spektrum des Ausgangssignals des Eingangs des DVB-T-Sticks mit Elnics E4000 Tuner (Resolution-Bandwidth = 100 kHz; Video-Bandwidth = 100 kHz)

Anhand dieser Darstellung ist deutlich erkennbar, dass der Stick starke Signale im Bereich von 260 MHz erzeugt und über den Eingang abstrahlt. Diese Signale sind ganzzahlige Vielfache der Frequenz des verbauten Quarzoszillators, wie es die Tabelle 5.3 verdeutlicht.

n	n · 28,8 MHz	n	n · 28,8 MHz
1	28,8 MHz	6	172,8 MHz
2	57,6 MHz	7	201,6 MHz
3	86,4 MHz	8	230,4 MHz
4	115,2 MHz	9	259,2 MHz
5	144 MHz	10	288 MHz

Tabelle 5.3: Vielfache der Frequenz des Quarzoszillators eines RTL-SDR mit E4000-Tuner

Besonders die neunte Oberwelle der Grundfrequenz fällt mit einem Pegel von knapp -70 dBm stark ins Gewicht. Diese Signale werden später im Spektrum als Störsignale erscheinen und Messungen verfälschen.

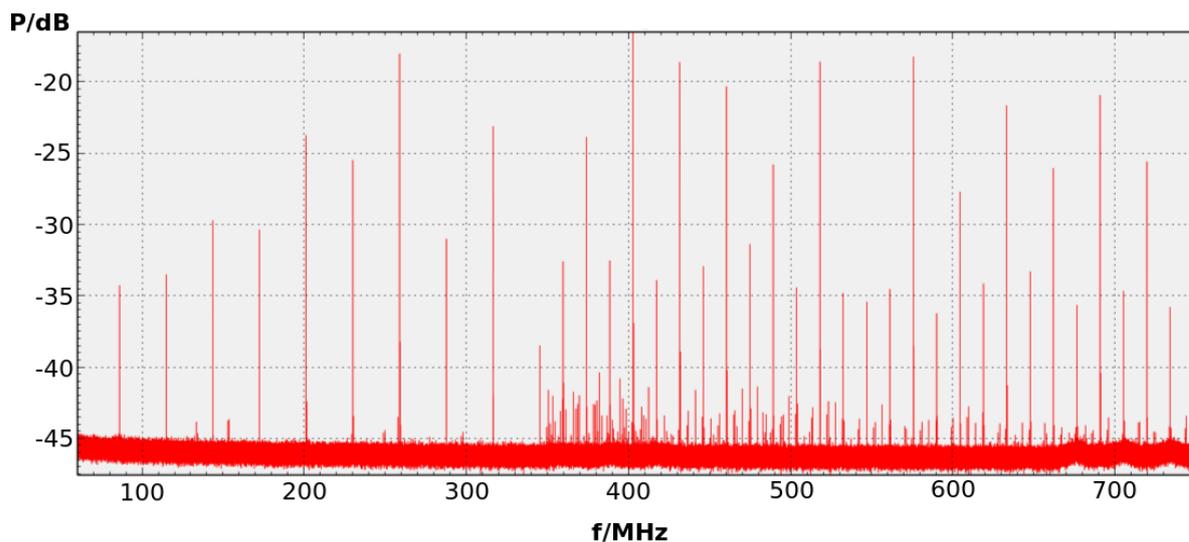


Abbildung 5.10: Betragsspektrum des Elonics E4000-Sticks bei einem ideal abgeschlossenen Eingang

Nimmt man das Betragsspektrum mit einem ideal abgeschlossenen Eingang auf, so bestätigen sich die oben angedeuteten Probleme. Wie die Abbildung 5.10 zeigt, entstehen im Frequenzbereich zwischen 50 und 400 MHz deutliche durch den verbauten Quarz entstandene Störlinien. Auch oberhalb der durch den Oszillator entstandenen Störungen sind weitere Störlinien zu erkennen die zudem sehr dicht beieinander liegen.

Für die eigentliche Messung des Eingangsreflexionsfaktors muss der Netzwerkanalysator kalibriert werden. In diesem Fall erfolgt die Kalibrierung mithilfe einer Kalibrationseinheit des Messgeräteherstellers. Diese kalibriert den Netzwerkanalysator automatisch nach dem TOSM-Prinzip. Das heißt, dass ein "Through"-, "Open"-, "Short"- und "Matched"-Abschluss zur Kalibrierung der Ports des Netzwerkanalysators genutzt wird.

Das "Device Under Test" (DUT) wird zur S_{11} -Messung mit Port 1 des Netzwerkanalysators verbunden. Das Ergebnis der Messung des Eingangsreflexionsfaktors für den E4000-Stick ist in Abbildung 5.11, beziehungsweise in Abbildung 5.12 im Smithdiagramm dargestellt. Da kein Anpassnetzwerk entwickelt werden soll, ist die komplexe Impedanz jedoch nicht von großer Bedeutung für diese Arbeit.

Es ist gut zu erkennen, dass der Eingang des DVB-T-Sticks eine, wie zu erwarten, schlechte Anpassung hat. Aus diesen Ergebnissen ergibt sich eine Reflexionsdämpfung von ungefähr 10 dB. Da die Pegel des Systems an dem das System später laufen soll stets hoch sind, ist der Verlust durch die Fehlanpassung jedoch nicht weiter von Bedeutung.

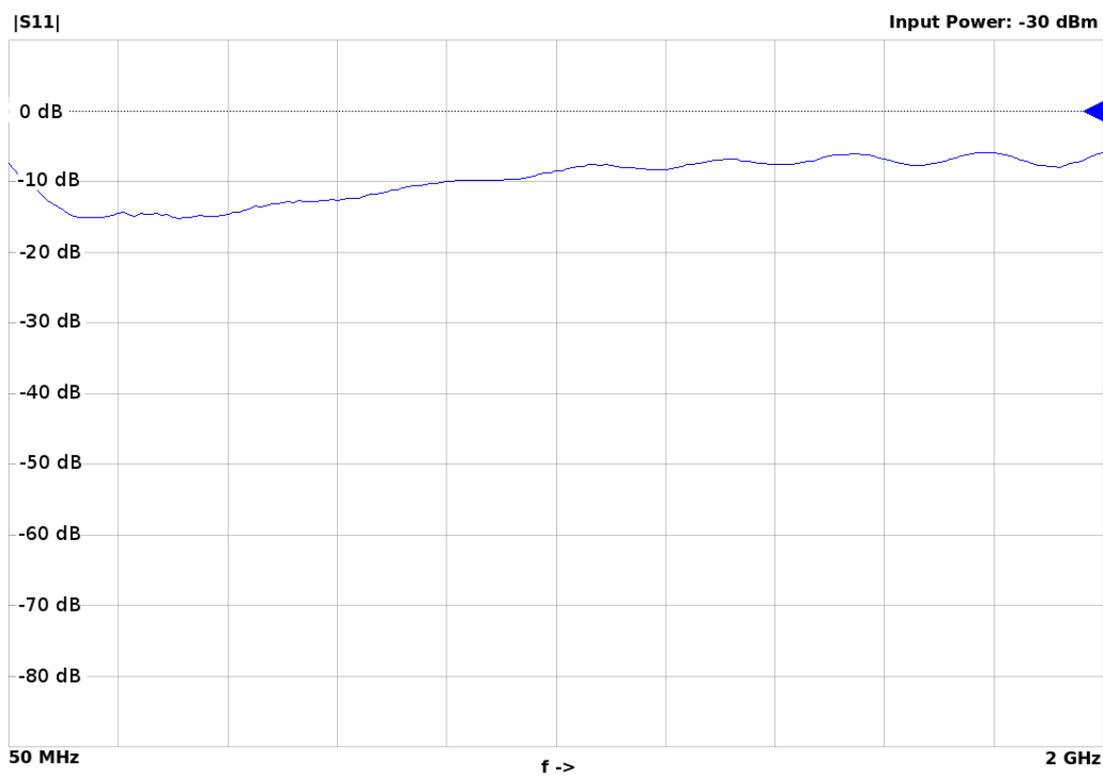


Abbildung 5.11: $|S_{11}|$ des DVB-T-Sticks mit Elonics E4000 Tuner

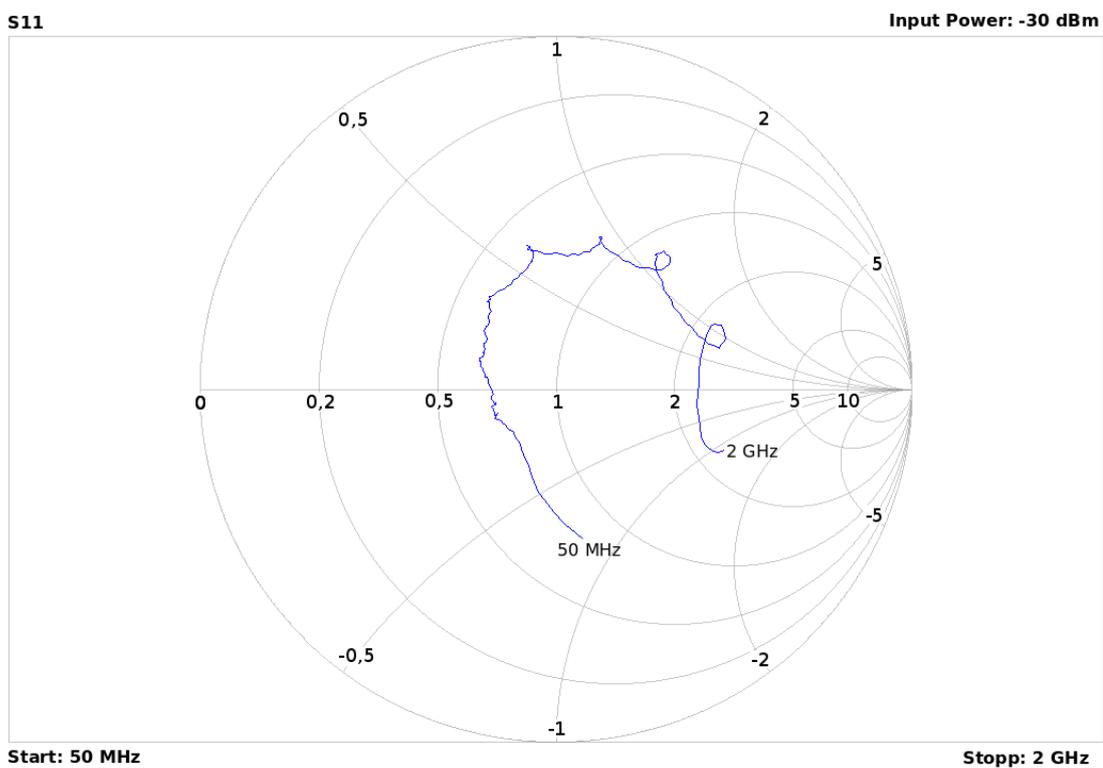


Abbildung 5.12: S_{11} des DVB-T-Sticks mit Elonics E4000 Tuner im Smithdiagramm

5.2.3 Störemission und Eingangsreflexionsfaktor des R820T-Sticks

Die Abbildung 5.13 zeigt das Ergebnis der Messung des Spektrums der vom R820T-Stick über den Eingang abgestrahlten Signale. Es ist erkennbar, dass dieser Stick deutlich keine Störungen über den Eingang abstrahlt.

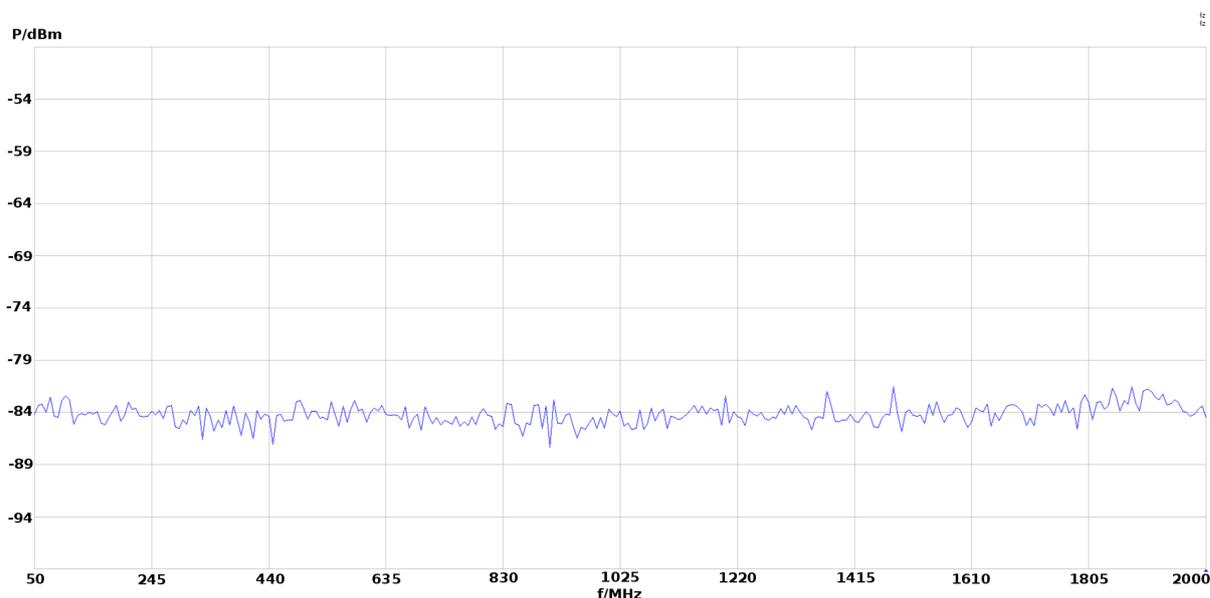
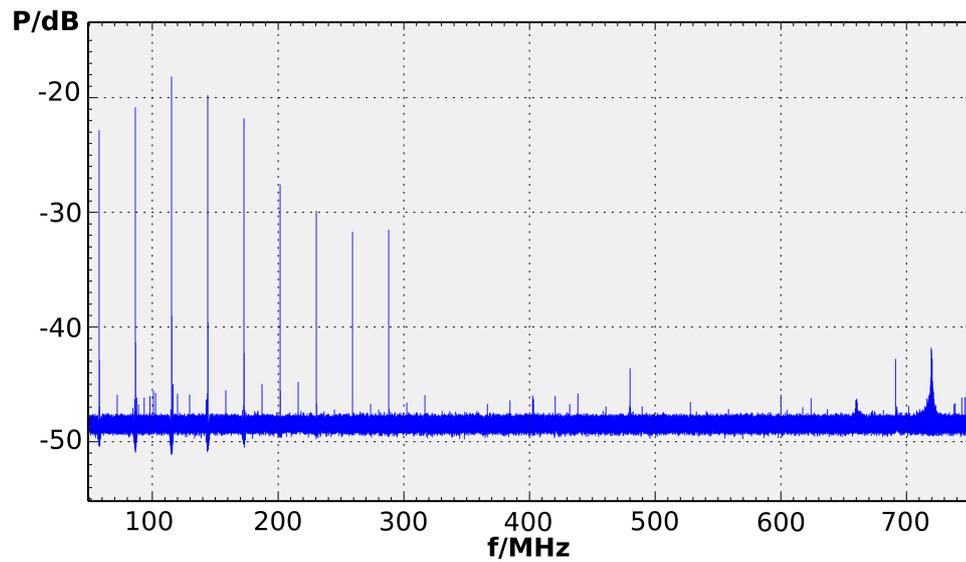
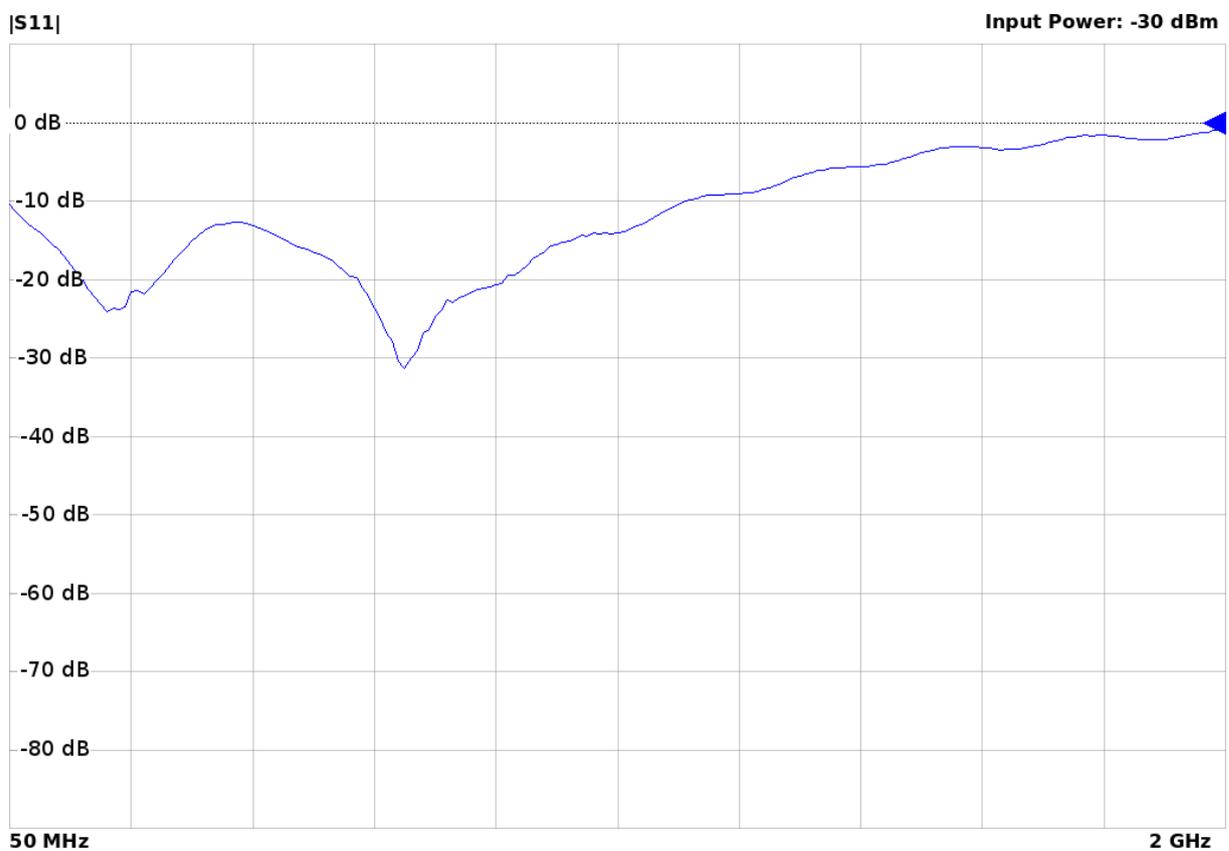


Abbildung 5.13: Spektrum des Ausgangssignals des Eingangs des DVB-T-Sticks mit Rafael Micro R820T Tuner zwischen 50 und 2000 MHz (Resolution-Bandwidth = 100 kHz; Video-Bandwidth = 100 kHz)

Ein anderes Bild ergibt sich, wenn das Spektrum bei einem Abschluss des Eingangs mit einem 50 Ω -Widerstand aufgenommen wird (siehe Abbildung 5.14). Auch in dem so aufgenommenen Spektrum sind die Oberwellen des verbauten Quarzes deutlich als Störlinien im Spektrum erkennbar. Diese stören bei der Detektion von RFI.

Für den R820T-Stick wurde eine äquivalente Messung des Eingangsreflexionsfaktors vorgenommen. Das Ergebnis ist in Abbildung 5.15 und 5.16 zu sehen. Auch hier ist eine verhältnismäßig schlechte Anpassung zu erkennen. Aus den gleichen Gründen wie zuvor wird jedoch auf die Verwendung eines Anpassnetzwerks verzichtet, da die Fehlanpassung aufgrund der hohen Eingangssignalpegel keinen direkten negativen Einfluss auf die angestrebte Anwendung hat.

Abbildung 5.14: Spektrum des Rafael Micro R820T-Sticks mit 50 Ω -AbschlussAbbildung 5.15: $|S_{11}|$ des DVB-T-Sticks mit Rafael Micro R820T Tuner

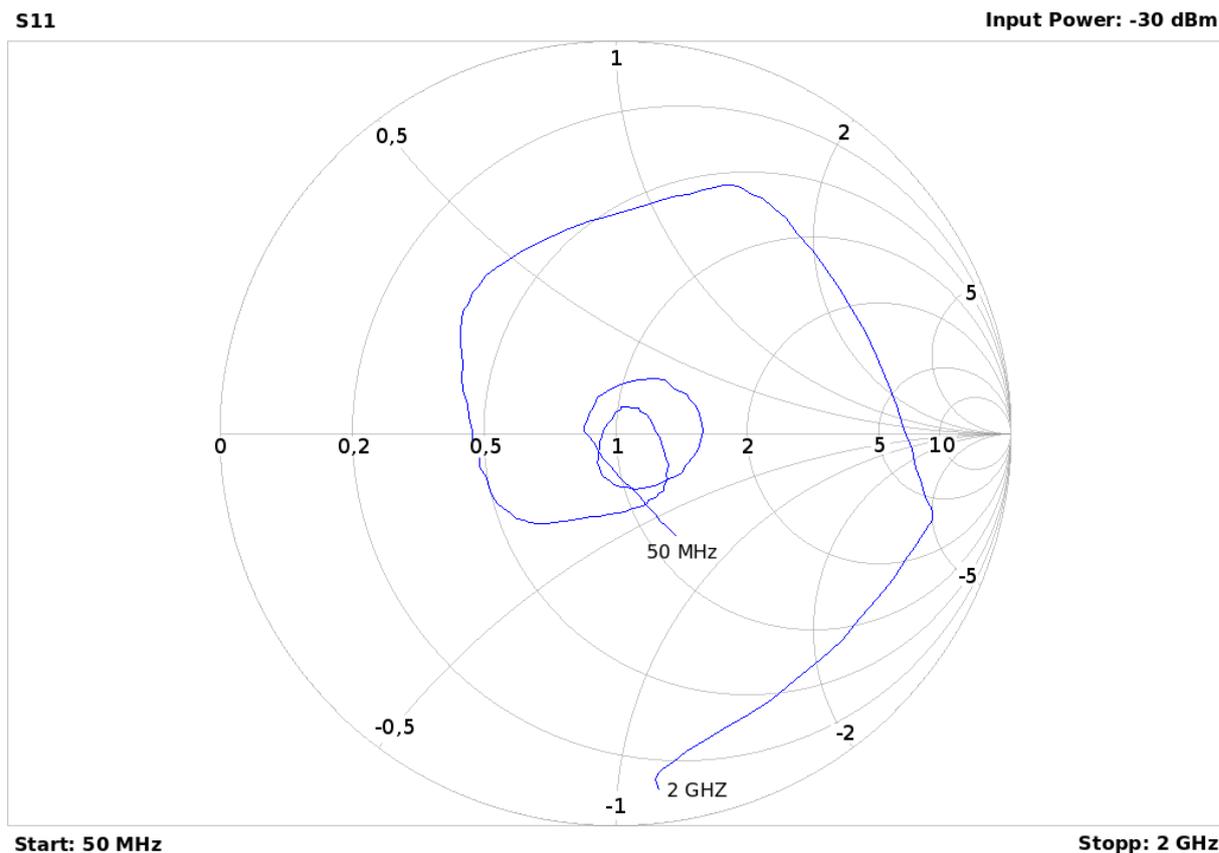


Abbildung 5.16: S_{11} des DVB-T-Sticks mit Rafael Micro R820T Tuner im Smithdiagramm

5.2.4 Störemission und Eingangsreflexionsfaktors des USRP

Zu Beginn der Messung des Eingangsreflexionsfaktors werden auch die Eingänge des USRP direkt an einen Spektrumanalysator angeschlossen. Das Ergebnis für den Ein- beziehungsweise Ausgang TX/RX1 ist in Abbildung 5.17 dargestellt. Das Ausgangsspektrum des Eingangs RX2 ist in Abbildung 5.18 zu sehen. Wie gut zu erkennen ist, senden die Ports zwischen 0 und 2 GHz keine Störsignale aus über den Anschluss aus.

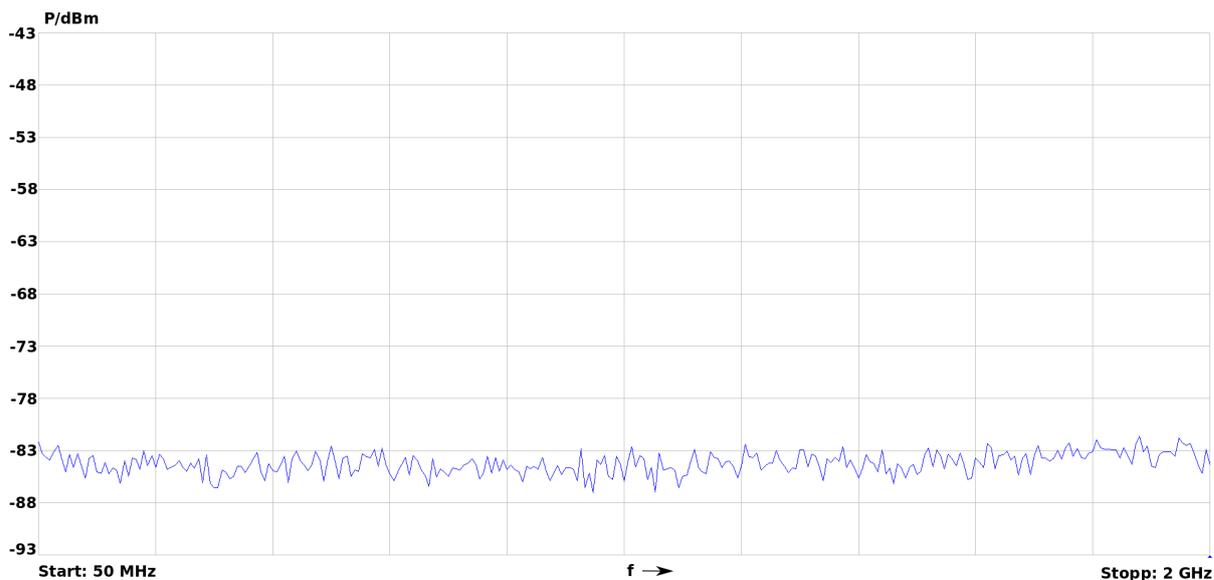


Abbildung 5.17: Störemissionen des Ein- und Ausgangs TX/RX1 zwischen 50 und 2000 MHz (Resolution-Bandwidth = 100 kHz; Video-Bandwidth = 100 kHz)

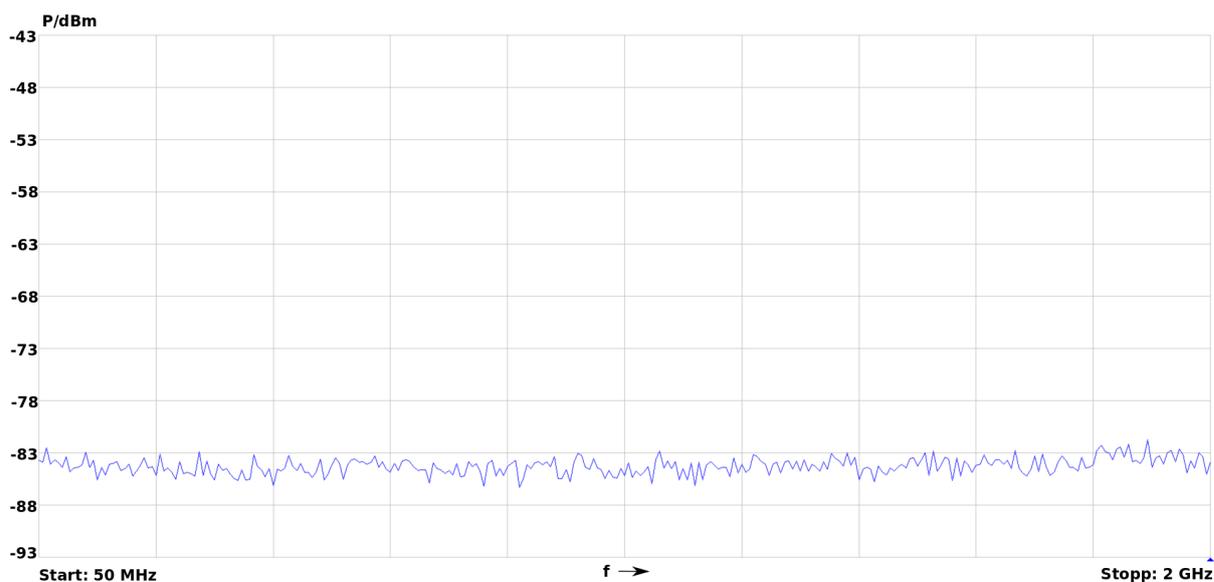


Abbildung 5.18: Störemissionen des Eingangs RX2 zwischen 50 und 2000 MHz (Resolution-Bandwidth = 100 kHz; Video-Bandwidth = 100 kHz)

Anders stellt sich die Sache dar, wenn beide Eingänge mit einem Abschlusswiderstand beschaltet werden und das Spektrum aufgenommen wird. Denn in diesem Fall sind auf beiden Kanäle deutliche Eigenstörlinien erkennbar, die aus dem Grundrauschen hervortreten (siehe Abbildung 5.19).

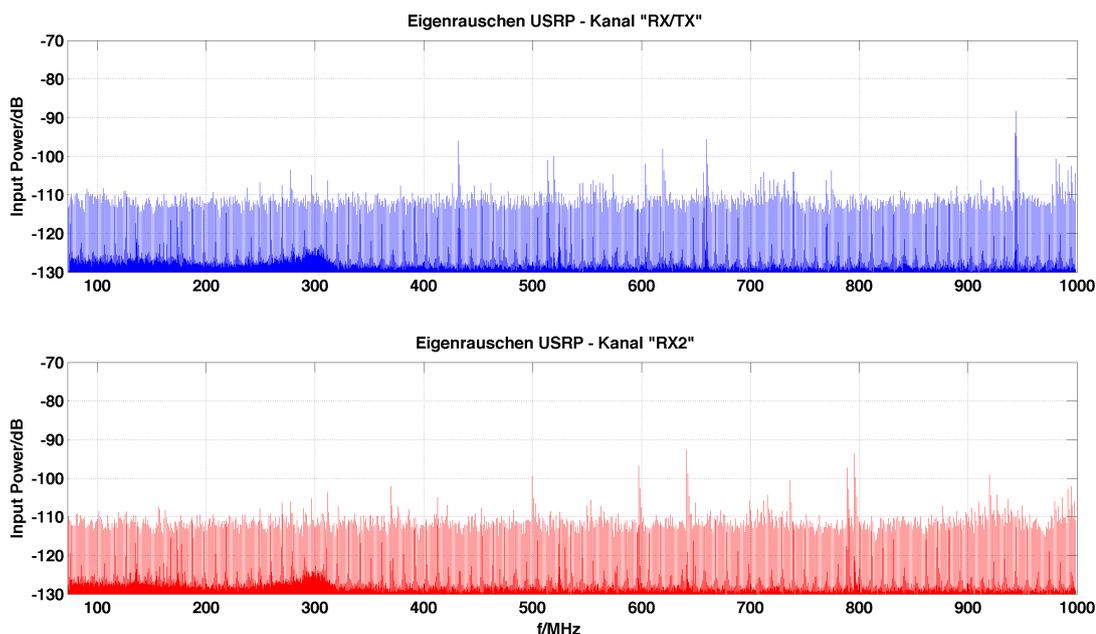


Abbildung 5.19: Spektrum der Kanäle TX/RX1 und RX2 bei einem 50 Ω -Abschluss

Diese Eigenstörungen sind für die sensible Detektion von RFI ein großer Nachteil. Denn je nach Anwendung ist ein Störsignal nicht von einer Eigenstörung durch das USRP unterscheidbar. Um festzustellen, inwiefern diese Eigenstörungen die Messung von Signalen beeinflusst, wird ein Rauschgenerator mit variabler Leistung an den Eingang des USRP geschaltet.

Das Signal wird über einen Splitter auf beide Eingänge des USRP aufgeteilt. Der Rauschgenerator erzeugt dabei ein Signal mit einer Gesamtleistung von $-11,5$ dBm. Das Signal wird durch den resistiven Splitter um je 6 dB gedämpft (siehe Abbildung 5.20). Anhand dieser Darstellung ist zu erkennen, dass bei einer zusätzlichen Dämpfung der Signale von 30 dB einige der Eigenstörlinien weiterhin aus dem Nutzrauschsignal hervortreten. Das ist exakt der Fall, der nicht erwünscht ist. Denn somit ist es nicht möglich, zwischen RFI und Eigenstörungen zu unterscheiden. Wird die Dämpfung verringert, verschwinden diese Anteile im Rauschen. Bei einer Dämpfung von 10 dB sind keine Eigenstörlinien mehr zu erkennen. Bei 5 dB Dämpfung stellt sich ein interessanter Effekt ein. Denn das Spektrum scheint zu oszillieren. Das kann dadurch erklärt werden, dass das Signal vom Eingang über den Splitter zurück auf den zweiten Eingang gekoppelt wird. Es entsteht eine stehende Welle. Diese Effekte beeinträchtigen die Tauglichkeit zur Ermittlung von RFI und sind bei der endgültigen Auswahl der Hardware zu bedenken.

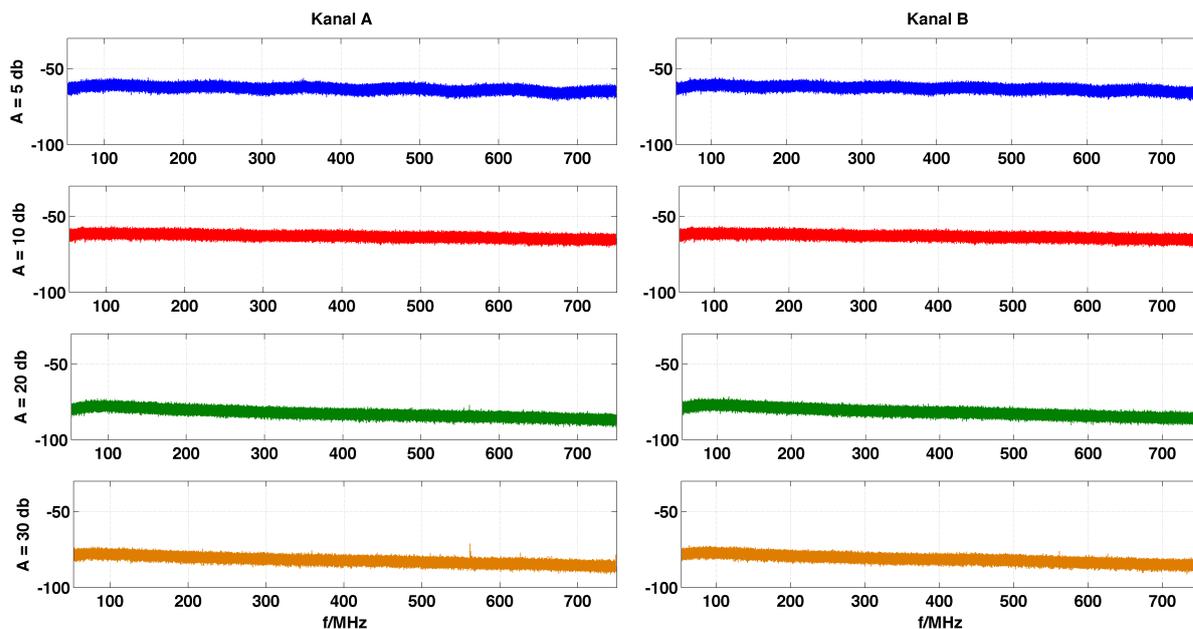


Abbildung 5.20: Spektrum eines Rauschsignals im Bereich zwischen 50 und 750 MHz mit unterschiedlicher, zusätzlicher Signaldämpfung

Die Messung des Eingangsreflexionsfaktors erfolgt, wie bereits zuvor dargestellt, mittels eines vektoriiellen Netzwerkanalysators. Die Darstellung des Betrags ist in Abbildung 5.21 zu sehen. Die komplexe Eingangsimpedanz lässt sich durch das in Abbildung 5.22 dargestellte Smithdiagramm bestimmen. Hier dargestellt ist dieser für den Kanal TX/RX1. Die Messungen haben für beide Kanäle ein vergleichbares Ergebnis geliefert.

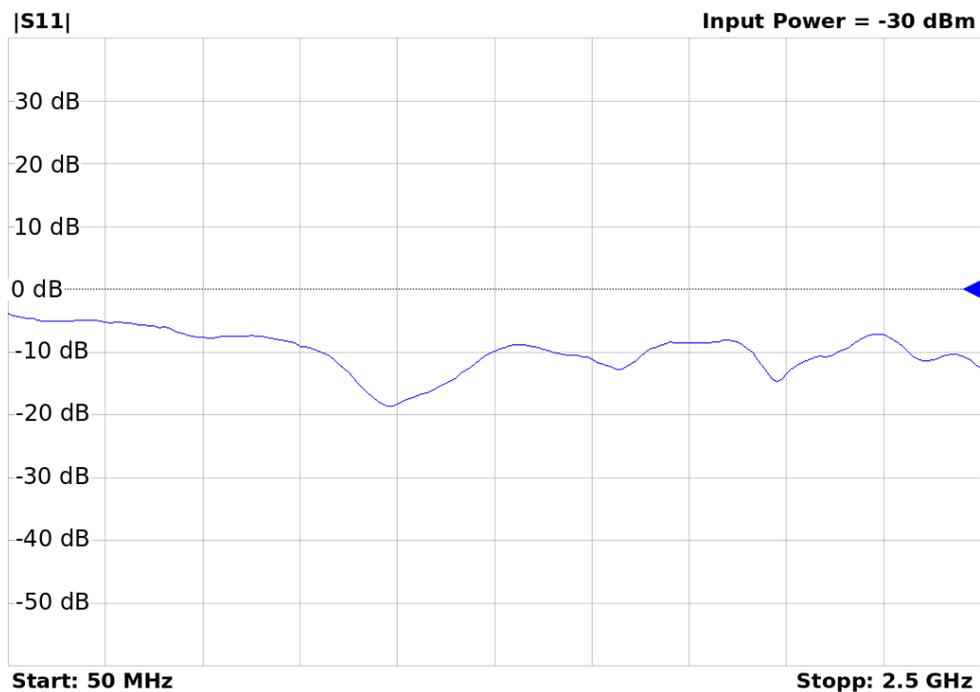


Abbildung 5.21: Betrag des Eingangsreflexionsfaktors des USRP-2920 zwischen 10 kHz und 2 GHz

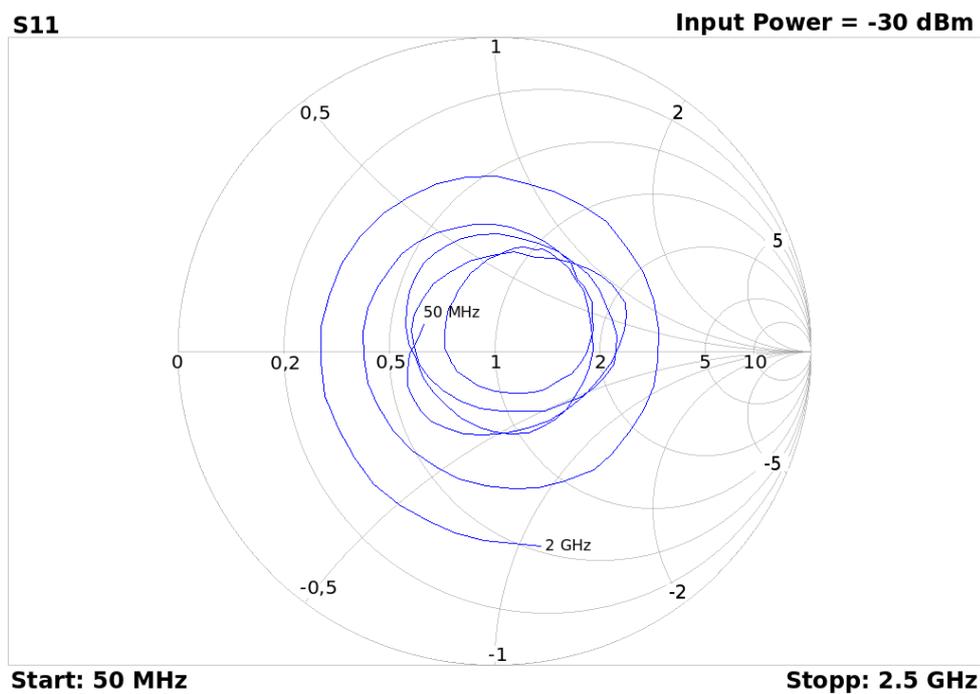


Abbildung 5.22: Eingangsreflexionsfaktor des USRP-2920 im Smithdiagramm

5.3 Frequenzgang der Signalverarbeitungskette

Um den Frequenzgang des Systems zu bestimmen, wurde über den einstellbaren Frequenzbereich bei einem konstanten Eingangspegel von -80 dBm der Pegel des Ausgangssignals bestimmt. Für die Messung wurde ein Signalgenerator verwendet, der ein sinusförmiges Zeitsignal erzeugt. Dessen Frequenz wurde bei einem konstanten Pegel über den einstellbaren Frequenzbereich des DVB-T-Sticks und des USRP variiert. Das Ergebnis der Messung ist in Abbildung 5.23 für die verwendeten SDR-Interfaces zu sehen. Die einstellbare Verstärkung wurde dabei auf 0 dB gestellt und die automatische Verstärkungsregelung für die Messung deaktiviert.

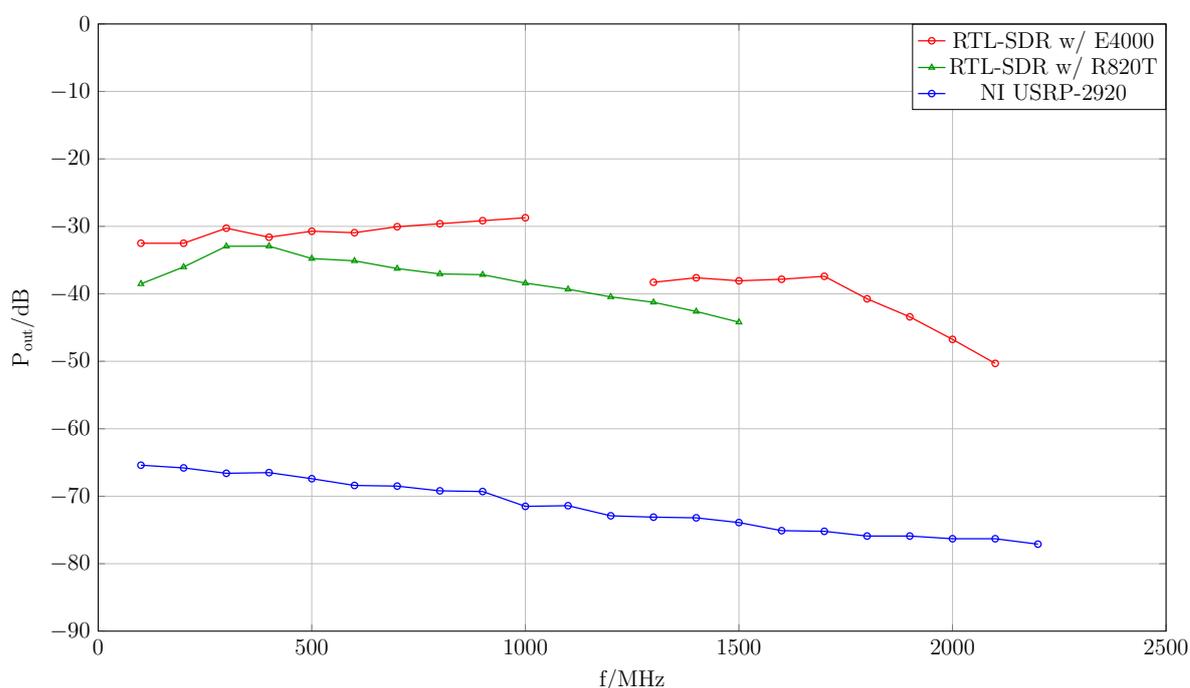


Abbildung 5.23: Frequenzgang der Signalverarbeitungskette des USRP-2920 und zweier RTL-SDRs bei einem Eingangsträgersignalpegel von -80 dBm

Anhand des Frequenzgangs der Signalverarbeitungskette des E4000-Sticks wird deutlich, dass bis zum Beginn einer Empfangslücke bei 1100 MHz der Verlauf nahezu konstant ist. Da die Phasenregelschleife zwischen 1100 und 1200 MHz nicht rastet kommt es zu der sichtbaren Empfangslücke. Zwischen einer Frequenz von 1700 MHz und der maximal einstellbaren Frequenz von 2100 MHz fällt der Frequenzgang stark ab.

Im Gegensatz zum Verhalten des E4000-Sticks fällt der Frequenzgang der Signalverarbeitungskette des R820T-Sticks konstant mit der Frequenz. Positiv zu bemerken ist, dass dieser Tuner keine Empfangslücke aufweist. Jedoch deckt sich die maximal einstellbare Frequenz nicht mit den Angaben, die auf der Website von Osmocom zu finden sind und am Anfang des Kapitels dargestellt wurden. Der verwendete Stick lässt sich nur bis 1500 MHz anstatt bis knapp 1700 MHz einstellen.

Der Frequenzgang des USRP dagegen ist über den kompletten einstellbaren Frequenzbereich sehr flach. Zudem weist dieser keine Lücken auf und die Messungen stimmen mit den Angaben im Datenblatt überein. Alle Frequenz zwischen 50 und 2200 MHz sind einstellbar.

5.4 Ermittlung des Dynamikumfangs

Um den Dynamikumfang zu bestimmen, wird ein konstantes Trägersignal mit einer Frequenz von 146 MHz auf das System gegeben. Das Ergebnis dieser Messung ist in der Darstellung 5.24 abgebildet. Die einstellbare Verstärkung wurde auf 0 dB eingestellt und die automatische Verstärkungsregelung deaktiviert. Der Pegel des Trägersignals wurde so lange erhöht, bis das Signal 3 dB über dem mittleren Grundrauschpegel liegt. Dieser Punkt wird als "Minimum Detectable Signal" (MDS) definiert. Der Trägersignalpegel wurde daraufhin solange weiter erhöht, bis der 1-dB-Kompressionspunkt erreicht wurde. Der Dynamikumfang ergibt sich aus der Differenz des MDS und dem 1-dB-Kompressionspunkt.

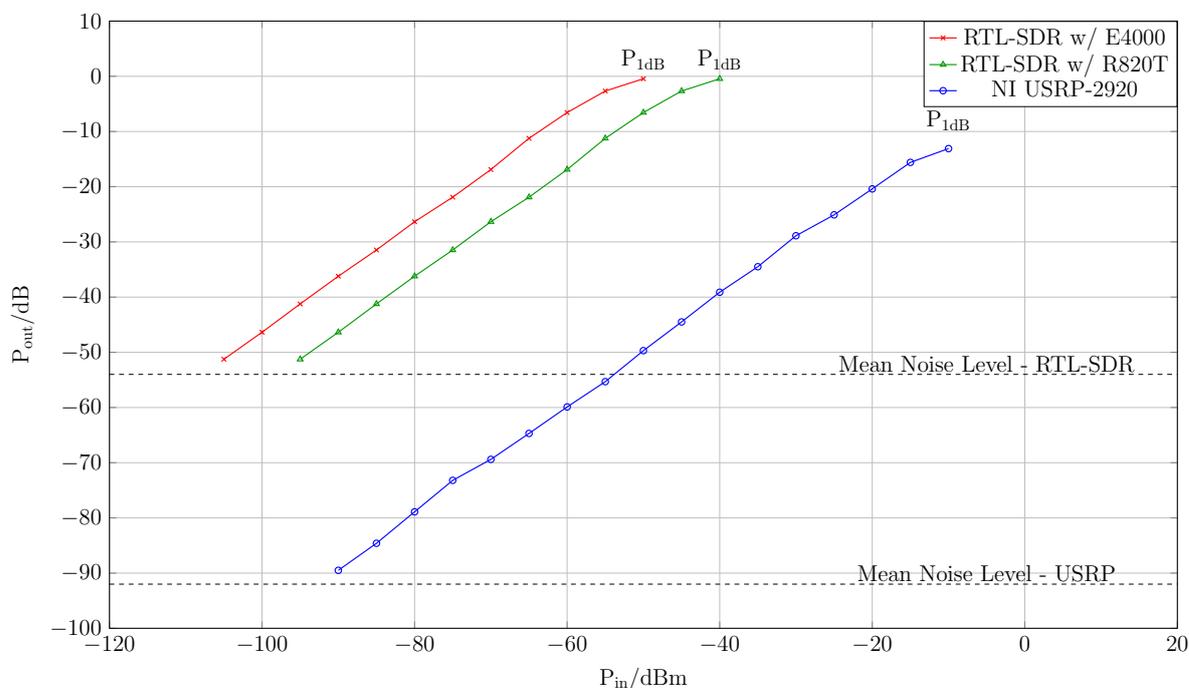


Abbildung 5.24: Vergleich des Dynamikumfangs zwischen einem USRP-2920, einem E4000-RTL-SDR und einem R820-RTL-SDR

Anhand der Abbildung 5.24 ist zu erkennen, dass das "Minimum Detectable Signal" beim R820T-Stick 10 dB höher liegt, als beim E4000-Stick. Das bedeutet, dass ein Eingangssignal einen Mindestpegel von -95 dBm beim R820T-Stick, beziehungsweise -105 dBm beim E4000-Stick, haben muss, um aus dem Grundrauschen hervorzutreten. Das deutet auf eine höhere Verstärkung der Signale beim E4000-Stick hin. Der 1-dB-Kompressionspunkt wird beim E4000-Stick bei einer Eingangsleistung von -55 dBm und beim R820T-Stick bei einer Eingangsleistung von -50 dBm erreicht. Zusammenfassend kann festgehalten werden, dass beide Sticks einen Dynamikumfang von knapp 50 dB aufweisen.

Das "Minimum Detectable Signal" des USRP liegt dagegen bei einem Eingangspegel von knapp -90 dBm. Der 1-dB-Kompressionspunkt wird bei einem Eingangspegel von -5 dBm erreicht. Daraus folgt, dass der Dynamikumfang beim USRP mit circa 85 dB fast doppelt so groß als bei einem RTL-SDR ist. Dies kommt besonders der exakten Bestimmung von kleinen sowie großen Störsignalen im Spektrum entgegen.

5.5 Bestimmung des Third-Order-Intercept-Punktes

Zur Messung des Third-Order-Intercept-Punktes wird ein über den Hersteller "Maxim Integrated" veröffentlichtes Tutorial zur Bestimmung dieses Punktes als Grundlage verwendet. Die nachfolgenden theoretischen Grundlagen beziehen sich, soweit nicht anders angegeben, auf diesen Artikel [8].

5.5.1 Theoretische Grundlagen zur Bestimmungsmethode

Intermodulationsprodukte entstehen als Mischprodukte von Eingangssignalen und werden durch nichtlineare Eigenschaften von zum Beispiel Verstärkern erzeugt. Diese Produkte sind unerwünscht, da sie im eigentlichen Signal nicht enthalten sind. Allgemein stellt der Intercept-Punkt einer beliebigen Ordnung einen Gütefaktor für die Linearität eines Bauteils oder eines Systems dar. Dieser ist dabei der Punkt, an dem die Intermodulationsprodukte einer gegebenen Ordnung den identischen Pegel haben, wie die eigentlichen Nutzsignale. Diese Punkte sind jedoch meist nicht messbar und werden durch Extrapolation von Messkurven erfasst, da die Systeme vor dem Erreichen dieses Punktes in die Kompression gehen. Ziel ist es, den Intercept-Punkt möglichst hochzuhalten, um so die bestmögliche Festigkeit gegen Intermodulation zu erzielen. Ganz besonders von Bedeutung ist der Intercept-Punkt dritter Ordnung, da Intermodulationsprodukte dieser Ordnung oft im zu erfassenden Frequenzbereich liegen. Auf die Messung dieses Punktes wird nachfolgend eingegangen.

Bei der Annahme eines Eingangssignals $x(t)$, ergibt sich das Ausgangssignal $y(t)$ über eine Approximation der nichtlinearen Übertragungseigenschaften zu:

$$y(t) = c_1 \cdot x^1(t) + c_2 \cdot x^2(t) + c_3 \cdot x^3(t) + \dots + c_n \cdot x^n(t) + \dots \quad (5.5)$$

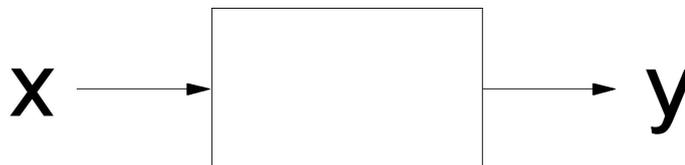


Abbildung 5.25: Blockschaltbild eines einfachen Übertragungssystems

Der Term $c_1 \cdot x^1(t)$ stellt den linearen Anteil des Ausgangssignals dar. Sei das Eingangssignal

$$x(t) = a_1 \cdot \cos(2\pi \cdot f_1 \cdot t) + a_2 \cdot \cos(2\pi \cdot f_2 \cdot t), \quad (5.6)$$

so erzeugt der Term erster Ordnung am Ausgang gemäß des Superpositionsprinzips exakt die selbe Summe, jedoch mit einem Vorfaktor c_1 gewichtet. Der Term zweiter Ordnung erzeugt hingegen Signale mit den Frequenzen $2 \cdot f_1$ und $2 \cdot f_2$ als Oberwellen und $f_1 + f_2$, $f_1 - f_2$ sowie $f_2 - f_1$ als Mischprodukte. Anhand beispielhafter Werte für die beiden Eingangsfrequenzen wird klar, dass diese Mischprodukte weit außerhalb des Frequenzbereichs liegen, der von Interesse ist. Problematisch sind jedoch Produkte dritter Ordnung. Denn diese erzeugen Produkte bei den Frequenzen

- $3 \cdot f_1$ und $3 \cdot f_2$ als Oberwellen,
- $2f_1 + f_2$, $2f_1 - f_2$ sowie
- $2f_2 + f_1$, $2f_2 - f_1$ als Mischprodukte.

Die Mischprodukte liegen deutlich dichter am eigentlichen Nutzsignalspektrum der beiden am Eingang anliegenden Trägersignale. Daher sind die Produkte dritter Ordnung die Produkte, die häufig Probleme in einem Empfänger bereiten. Eine beispielhafte Darstellung für dieses Verhalten ist in Abbildung 5.26 dargestellt. Die beispielhaften Trägersignale, dargestellt in blau, bei den Frequenzen 140 und 160 Hz, stellen dabei ein Eingangssignal entsprechend der Formel 5.6 dar. Aufgrund der zuvor dargestellten Theorie entstehen bei dieser Kombination des Eingangssignals Intermodulationsprodukte zweiter und dritter Ordnung. Diese in grün beziehungsweise rot dargestellten Signalanteile werden durch nichtlineare Eigenschaften des Übertragungssystem erzeugt.

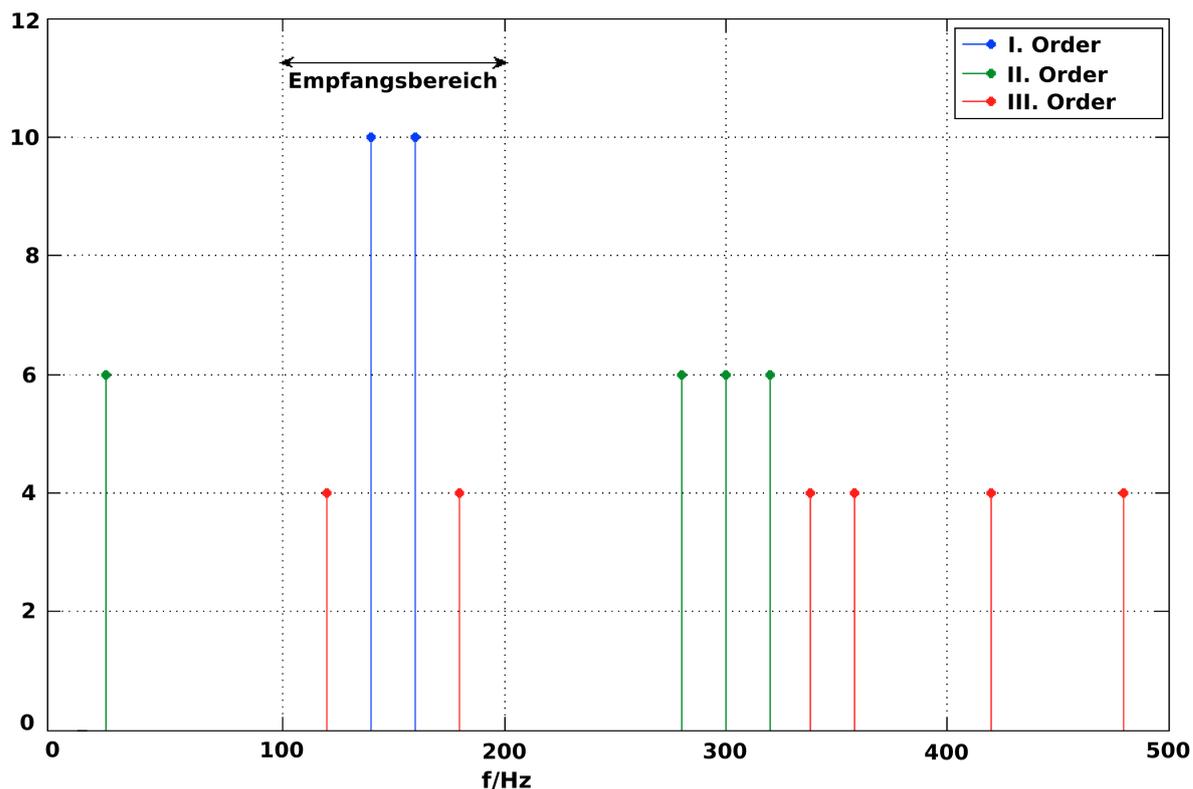


Abbildung 5.26: Prinzipielle Darstellung der Verteilung der Intermodulationsprodukte zweiter und dritter Ordnung für ein beispielhaftes Eingangssignal

Als Nächstes wird der Intercept-Punkt bestimmt, also der Punkt, an dem der Pegel der Intermodulationsprodukte dem Pegel der Eingangsträgersignale am Ausgang des Systems entspricht. Ausgehend von diesem Punkt kann der für dieses Verhalten notwendige Eingangspegel (IIP3) und der entsprechende Ausgangspegel (OIP3) bestimmt werden (siehe Abbildung 5.27). In der Regel geraten die zu testenden Systeme vor dem Erreichen des Intercept-Punktes in die Kompression. In diesem Fall wird eine Extrapolation des Verhaltens des Systems vorgenommen.

Die doppelt logarithmische Darstellung zeigt, dass die Produkte dritter Ordnung in dieser Darstellung als eine Gerade mit einer Steigung von drei eingehen. Die Produkte erste Ordnung folgen dagegen einer Geraden mit der Steigung eins. Der IP3 ist entsprechend im Schnittpunkt der beiden Geraden zu finden. Von diesem Schnittpunkt aus lässt sich der notwendige Eingangspegel zum Erreichen dieses Punktes ablesen. Der Versatz der Geraden erster Ordnung auf der Ausgangspegel-Achse kommt durch eine Verstärkung der Signale zustande.

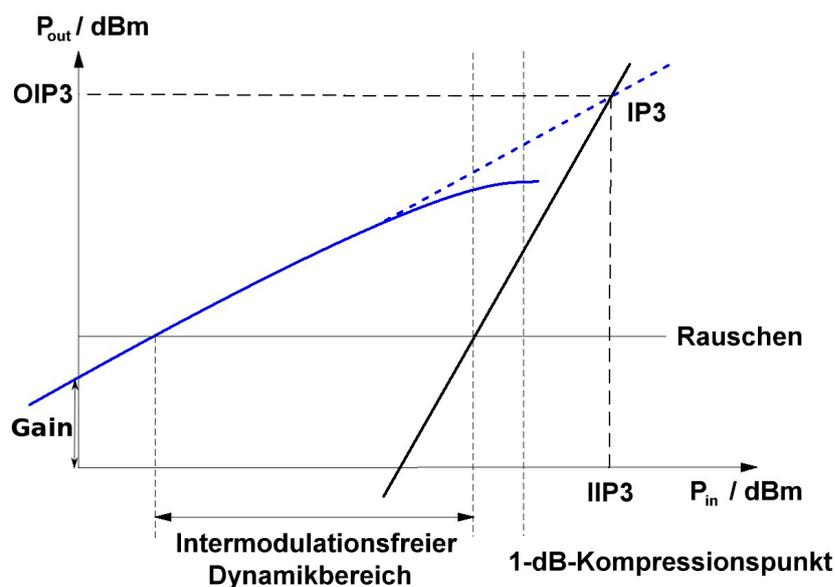


Abbildung 5.27: Prinzip zur Bestimmung des Third-Order-Intercept-Punktes

Es wird angenommen, dass ein Eingangssignal mit einer Leistung von P_{in} auf das System gegeben wird. Die Leistung P_{out} entspricht der gemessenen Leistung der Produkte erster Ordnung, die Leistung P_{out_3} hingegen der gemessenen Leistung der Produkte dritter Ordnung. Über eine grafische Herleitung ergibt sich der Zusammenhang:

$$\text{OIP}_3 = P_{\text{out}} + \frac{P_{\text{out}} - P_{\text{out}_3}}{2} \quad (5.7)$$

$$\text{IIP}_3 = P_{\text{in}} + \frac{P_{\text{out}} - P_{\text{out}_3}}{2}. \quad (5.8)$$

5.5.2 Third-Order-Intercept-Punkt der RTL-SDRs

Um den Third-Order-Intercept-Punkt zu ermitteln, werden zwei Signalgeneratoren benötigt (siehe Abbildung 5.28). Diese werden beispielhaft auf eine Frequenz von 750 MHz beziehungsweise 750,5 MHz eingestellt. Die Theorie besagt, dass so unter anderem Intermodulationsprodukte bei den Frequenzen 749,5 MHz und 751 MHz entstehen. Aufgrund der geringen Bandbreite der Hardwareinterfaces liegen ausschließlich diese Produkte im erfassbaren Frequenzbereich. Die beiden Trägersignale werden auf einen HF-Combiner gegeben und zur Kontrolle über einen Power Divider an einen Spektrumanalysator und die SDR-Hardware verteilt.

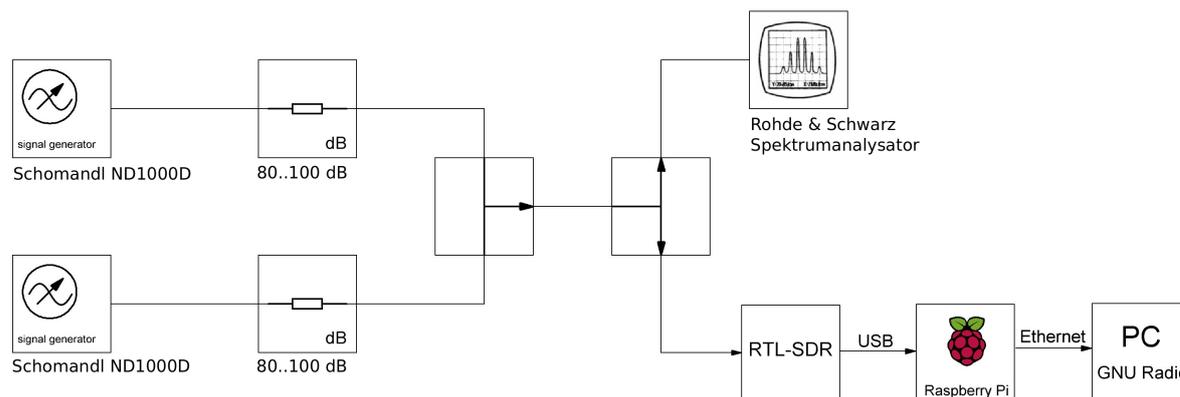


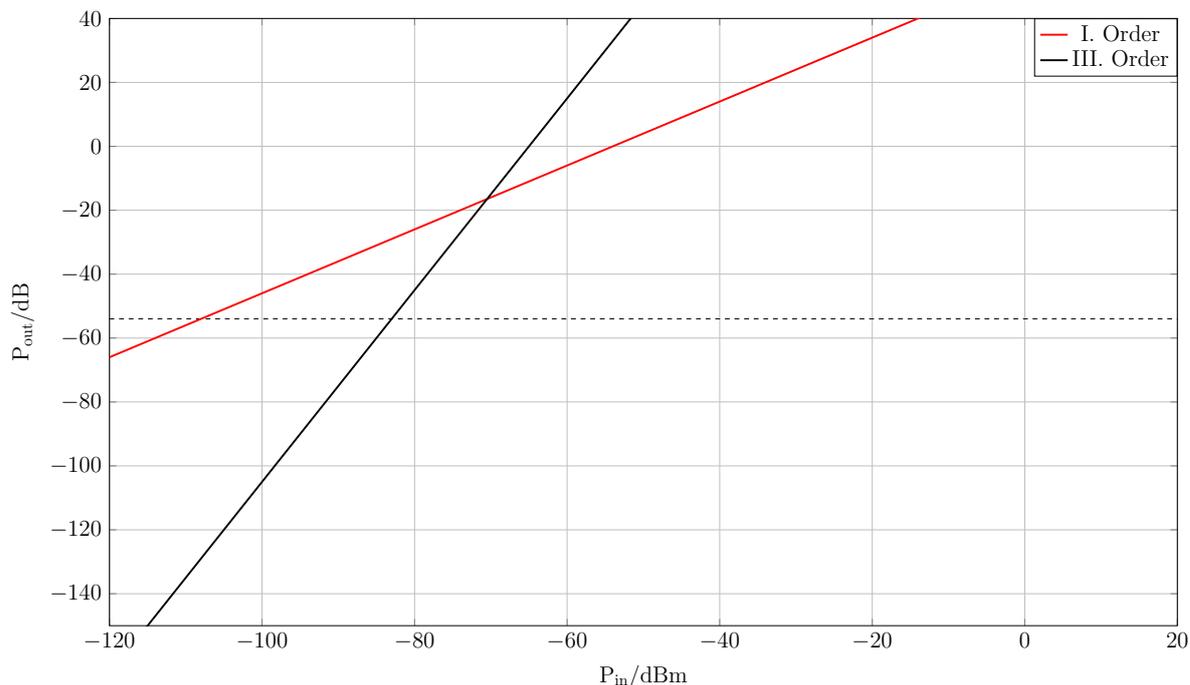
Abbildung 5.28: Blockschaltbild zur Messung des Third-Order-Intercept-Punktes eines DVB-T-Sticks

Der Pegel des Eingangssignals wird für die Messung variiert und der Ausgangspegel der Träger, sowie der Intermodulationsprodukte dokumentiert. Dies geschieht mit Hilfe eines auf GNU Radio basierenden Spektrumanalysators des Entwicklers Osmocom.

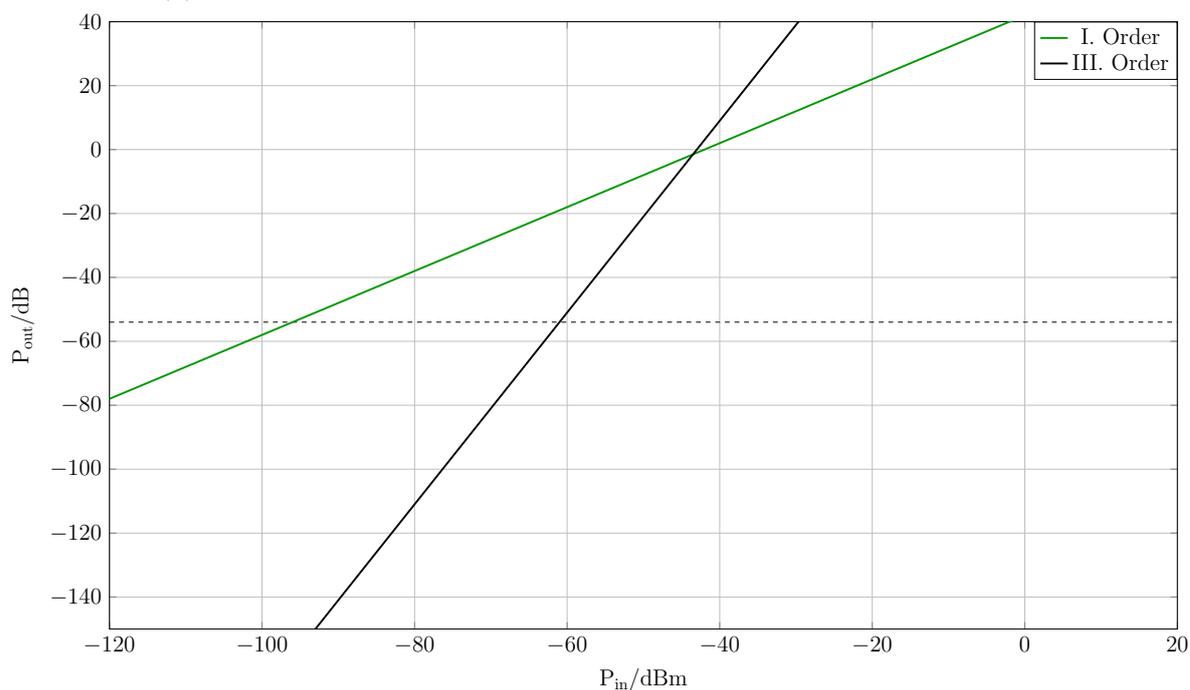
Aus den ermittelten Werten werden zwei Geraden extrapoliert und linearisiert, um den Third-Order-Intercept Punkt zu bestimmen. Diese werden über den Eingangspegel aufgetragen. Das Ergebnis ist für den E4000-Stick in Abbildung 5.29a dargestellt und für den R820T-Stick in Abbildung 5.29b.

Anhand dieser Darstellungen ist zu erkennen, dass der Third-Order-Intercept-Punkt des E4000-Sticks deutlich unter dem des R820T-Sticks liegt. Der Intercept-Punkt wird beim E4000-Stick bereits bei einem Eingangspegel von -74 dBm erreicht. Beim R820T-Stick liegt der Punkt bei einem Eingangspegel von knapp -42 dBm. Das macht einen Unterschied des IIP3 zwischen beiden Sticks von über 20 dB. Jedoch besitzt der E4000-Stick eine höhere Verstärkung, die dazu führt, dass der Intercept-Punkt früher erreicht wird. Unter Anbetracht dieses Aspektes ist die Intermodulationsfestigkeit des R820T-Sticks weiterhin um etwa 10 dB besser.

Mithilfe dieser Darstellungen ist es zudem möglich, den Spurious Free Dynamic Range zu bestimmen. Als untere Grenze dieses Bereiches wird dabei der Wert angenommen, bei dem der Pegel der Nutzsingale 3 dB über der mittleren Leistung des Grundrauschens liegt. Als obere Grenze wird der Punkt angenommen, bei dem die Intermodulationsprodukte aus dem Grundrauschen hervor treten. Dabei ergibt sich für den E4000-Stick ein störungsfreier, dynamischer Bereich von etwa 25 dB. Der R820T-Stick weist dagegen einen SFDR von etwa 35 dB auf.



(a) Elonics E4000-Stick; IIP3: -74 dBm, OIP3: -17 dB, SFDR: 25 dB



(b) Rafael Micro R820T-Stick; IIP3: -42 dBm, OIP3: -2 dB, SFDR: 35 dB

Abbildung 5.29: Darstellung des Third-Order-Intercept-Punktes für beide Tuner; gestrichelte Linie: mittlerer Rauschpegel; schwarze Geraden: Ausgangsprodukte dritter Ordnung; rote bzw. grüne Gerade: Ausgangsprodukte erster Ordnung

5.5.3 Third-Order-Intercept-Punkt des USRP-2920

Der Third-Order-Intercept-Punkt wurde auch für das USRP bestimmt. Die Abbildung 5.30 zeigt beispielhaft das Auftreten von Intermodulationsprodukten im Spektrum. Wie bereits anhand dieser Darstellung erkennbar ist, sind die Intermodulationsprodukte trotz des hohen Nutzsignalpegels sehr klein. Dies bestätigt sich, wenn die Extrapolation des linearisierten Verhaltens des USRP betrachtet wird (siehe Abbildung 5.31).

Der hier bestimmte IIP3 liegt bei einem Eingangspegel von +15 dBm und einem Ausgangspegel von knapp +20 dB. Daraus ergibt sich ein störungsfreier dynamischer Bereich von 70 dB. Somit führt eine geschickte Dämpfung der Signale am Eingang dazu, dass der Nutzsignalpegel zwar gesenkt wird, die Intermodulationsprodukte jedoch nicht mehr aus dem Grundrauschen hervortreten. So kann eine bessere Dynamik des Empfängers erzielt werden. Dieser Effekt wird insofern ausgenutzt, dass ein 10-dB-Dämpfungsglied vor den Eingang des USRP geschaltet wird. So ist es sicherer, dass im Spektrum keine Intermodulationsprodukte erscheinen und zur fehlerhaften RFI-Detektion führen.

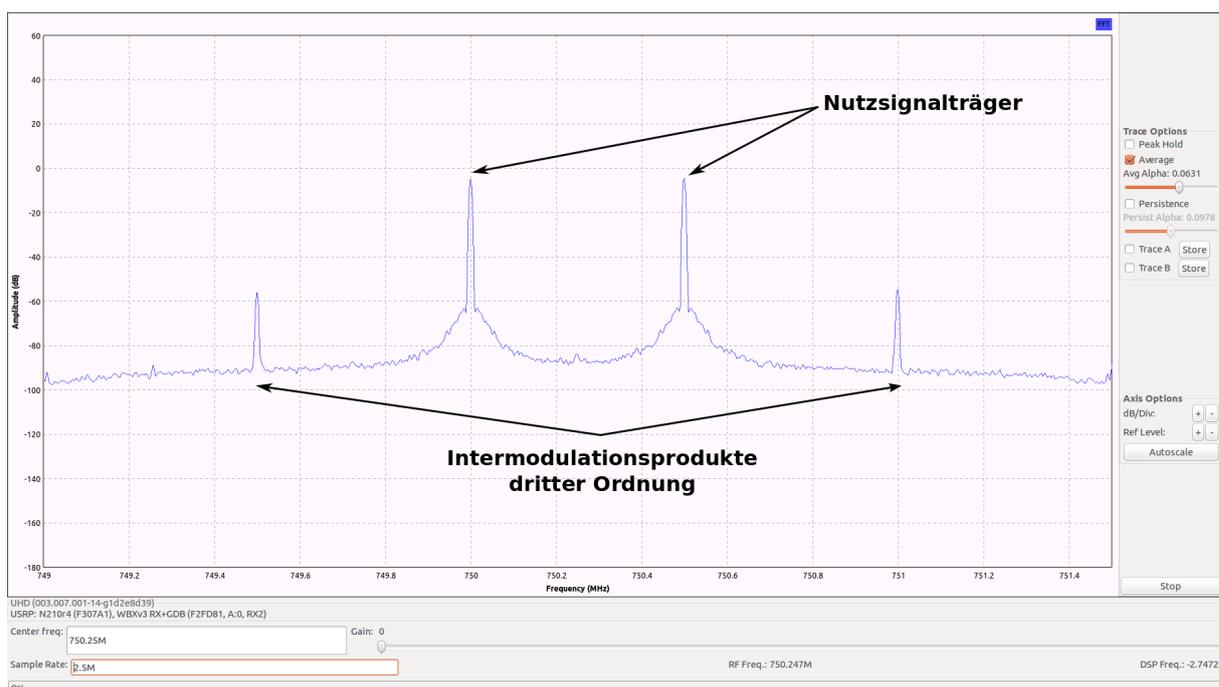


Abbildung 5.30: Intermodulationsprodukte am Ausgang des USRP-2920

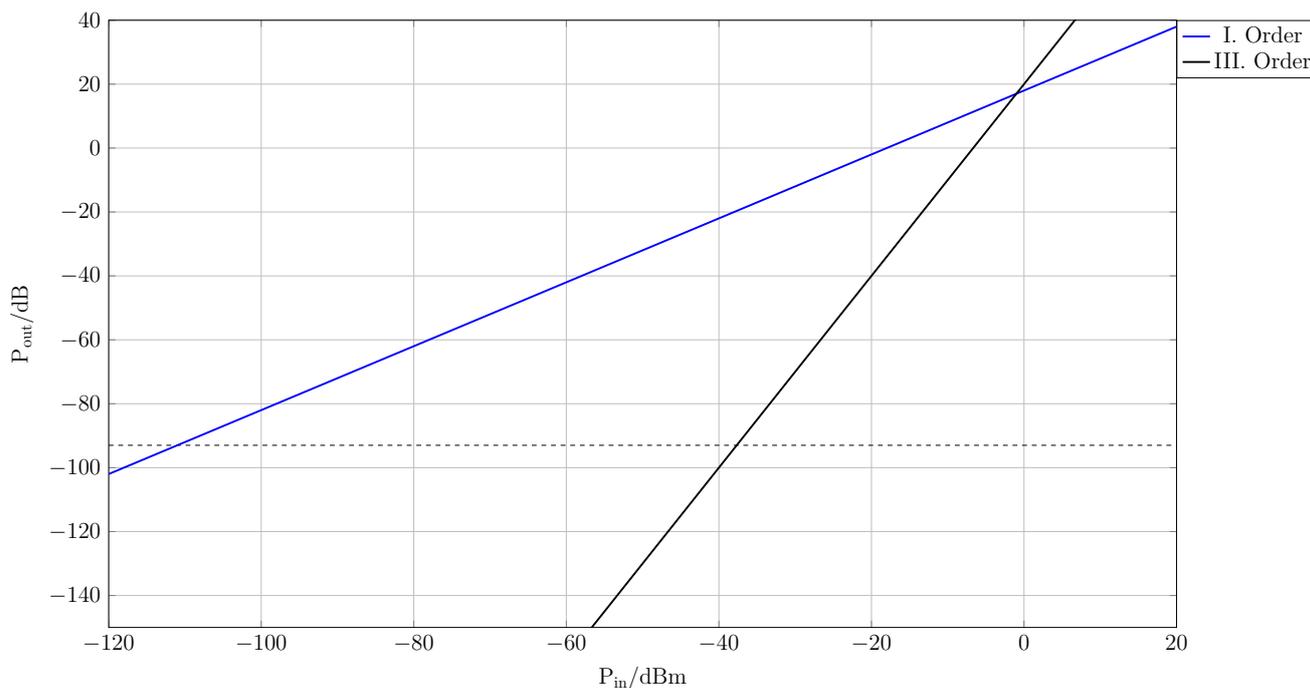


Abbildung 5.31: Third-Order-Intercept-Punkt des USRP-2920 - gestrichelte Linie: mittlerer Rauschpegel; IIP3 = +15 dBm, OIP3: +18 dB, SFDR = 70 dB

5.6 Diskussion der ermittelten Messergebnisse

Zu Beginn dieser Untersuchungen wurden der Aufbau und die Funktionsweise eines DVB-T-Sticks und eines USRP als SDR-Hardware gezeigt. Messtechnisch wurde zunächst der Eingangsreflexionsfaktor bestimmt. Die Anpassung aller drei Hardwareinterfaces bewegte sich in einem akzeptablen Bereich. Doch im Zusammenhang dieser Arbeit und der späteren Anwendung kann diese Fehlanpassung hingenommen werden, da die Eingangssignalpegel stets hoch genug sind und die Fehlanpassung so keine Probleme bereitet. In diesem Zusammenhang wurde das Störemissionsspektrum untersucht. Besonders der E4000 wies starke Störlinien über einen breiten Frequenzbereich auf. Beide Sticks sowie das USRP zeigten bei einem idealen Abschluss viele Eigenstörungen über den kompletten einstellbaren Frequenzbereich. Diese Störlinien erschweren eine Detektion von RFI im Signalspektrum, da nicht zwischen Eigenstörung und Fremdstörung unterschieden werden kann.

Die Untersuchung des Frequenzgangs der Signalverarbeitungskette hat gezeigt, dass der Frequenzbereich, den die Sticks abdecken, relativ stark begrenzt ist, beziehungsweise im Falle des E4000-Tuners eine 100 MHz breite Frequenzlücke aufweist. Der Dynamikumfang ist dagegen positiv zu bewerten, da beide Sticks einen Umfang von knapp 50 dB zwischen dem kleinsten, detektierbaren Signal und dem 1-dB-Kompressionspunkt abdecken. Das USRP hat mit einem Dynamikumfang von 85 dB jedoch einen deutlichen Vorteil gegenüber den DVB-T-Sticks.

Der anschließend untersuchte Third-Order-Intercept-Punkt hat weitere Probleme der DVB-T-Sticks aufgezeigt. Besonders der E4000-Stick ist aufgrund seines IP3 bei einem Eingangspegel von -74 dBm ungeeignet für die angestrebte Anwendung. Um auszuschließen, dass es an dem speziell verwendeten Stick lag, wurden die Untersuchungen mit einem baugleichen Stick wiederholt und konnten so verifiziert werden. Beide Sticks haben einen relativ schmalen intermodulationsfreien Dynamikbereich von 25 dB (E4000) beziehungsweise 35 dB (R820T). Das bedeutet, dass schon bei geringen Eingangspegeln mit dem Hervortreten von Intermodulationsprodukten aus dem Grundrauschen zu rechnen ist. Da diese als Linien im Spektrum erscheinen, wird eine Unterscheidung zwischen RFI und Eigenstörung weiter erschwert. Die Intermodulationsfestigkeit des USRP ist dagegen deutlich besser, da der IP3 erst bei einem Eingangspegel von 15 dBm erreicht wird. Der störungsfreie Dynamikbereich ist mit 70 dB zudem deutlich größer.

In den nachfolgenden Kapiteln wird mit Hilfe der hier dargestellten SDR-Interfaces ein DFT-Spektrumanalysator entwickelt. Die Softwareentwicklung für beide Interfaces getrennt dargestellt, auch wenn beide das gemeinsame Python-SDK GNU Radio nutzen. Im Zusammenhang der Softwareentwicklung werden durch die Ergebnisse dieses Kapitels entstehende Probleme betrachtet. Um eine flexible Entwicklung der Software zu ermöglichen wird im nächsten Kapitel die Anbindung der Hardware an das Netzwerk des Instituts betrachtet.

6 Einrichtung eines RTL-SDR-Servers

Der DVB-T-Stick zum Betreiben eines Software Defined Radios wird am Zwischenfrequenz-System des Max-Planck-Instituts für Radioastronomie Einsatz vorgesehen. Aufgrund der großen räumlichen Entfernung zwischen dem Server, der die Signalverarbeitung betreiben wird und der Anbindung an das ZF-System ist der Anschluss der Hardware per USB nicht möglich, da USB 2.0 eine maximale Kabellänge von fünf Metern erlaubt [42]. Aus diesem Grund wird ein Interface benötigt, das eine Übertragung per Ethernet ermöglicht. Dazu wurde in dieser Arbeit ein Raspberry Pi verwendet, um die vom DVB-T-Stick erzeugten Samples des abgetasteten IQ-Signals per TCP an einen entfernten Rechner zu streamen. Auf diesem wird später die Software laufen, die zur Signalverarbeitung benötigt wird. Nachfolgend wird schrittweise erläutert, wie der Raspberry Pi für diesen Einsatz konfiguriert wurde. Das Prinzip des Aufbaus ist in Abbildung 6.1 dargestellt.

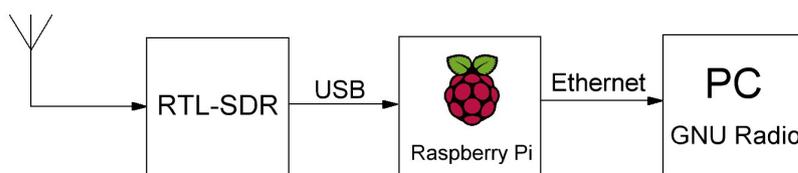


Abbildung 6.1: Prinzip der Einbindung des Raspberry Pi in das Projekt

Der Raspberry Pi hat nach seiner Markteinführung im Jahre 2012 für viel Aufsehen gesorgt. Aufgrund seines geringen Preises und der großen Community rund um die Programmierung dieses Einplatinencomputers, wurde dieser sehr beliebt bei Programmierern zum Experimentieren. Für diese Arbeit wurde das Modell B verwendet, welches einen 700-MHz-ARM-Prozessor sowie 512 MB RAM hat. Das verwendete UNIX-Betriebssystem "Rasbian" wird auf einer SD-Karte installiert und kann nach kurzer Konfiguration verwendet werden [24]. Die Programmierung des Raspberry Pi ist für die Programmiersprache Python ausgelegt (das "Pi" steht für "Python Interpreter"). Daher passt dieser gut in das Konzept dieser Arbeit, da GNU Radio eine Programmierung in Python unterstützt und die Softwareentwicklung ebenfalls in Python erfolgen wird [39].

Die Taktfrequenz von 700 MHz liegt exakt in der Zwischenfrequenz der beim MPIfR eingesetzten radioastronomischen Empfänger. Um ein Übersprechen des Taktsignals zu verhindern, beziehungsweise um keinen negativen Einfluss auf Messungen zu haben, wird die Frequenz auf 1 GHz hinaufgesetzt. Das hat eine deutliche Erhöhung der Wärmeentwicklung der CPU zur Folge. Um diesem entgegenzuwirken, werden Kühlkörper auf verschiedenen ICs des Raspberry Pi installiert.

Zudem begrenzt das auf dem Raspberry Pi laufende Programm zum Stream der digitalen IQ-Daten die maximal mögliche Abtastrate des RTL-SDR über Ethernet auf 2,4 MS/s und somit auch die erfassbare Bandbreite. Ein Bild des verwendeten Raspberry Pi ist in Abbildung 6.2 zu sehen.

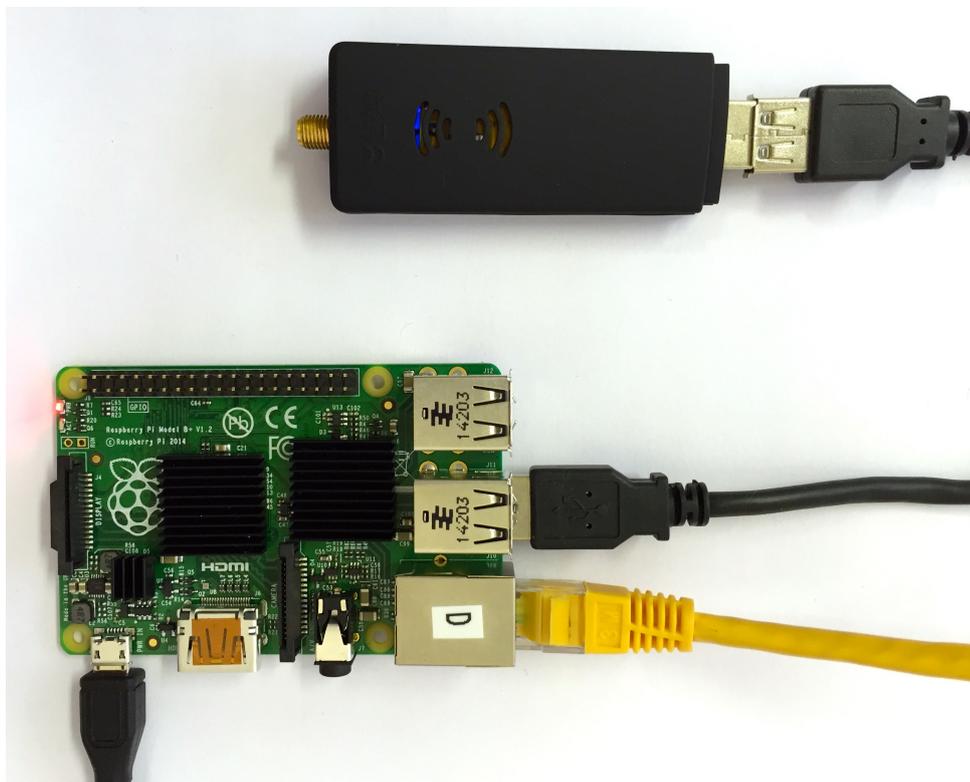


Abbildung 6.2: Raspberry Pi mit verbauten Kühlkörpern und einem RTL-SDR

Um einen DVB-T-Stick als Software Defined Radio über den Raspberry Pi zu betreiben, beziehungsweise um die Samples über TCP zu streamen, wird dem Raspberry Pi zu Beginn eine statische IP-Adresse zugewiesen. So kann ein ständiges Suchen des Gerätes im Netzwerk verhindert werden. Die auf dem Raspberry Pi installierte Software wird zu Beginn auf den aktuellsten Stand gebracht. Danach müssen einige Pakete installiert werden, die für den Betrieb der RTL-SDR Software notwendig sind. Die bis dahin notwendigen Terminal-Befehle sind im nachfolgenden Listing 6.1 dargestellt.

Listing 6.1: Vorbereitung zur Installation eines RTL-SDR Server auf einem Raspberry Pi

```

1 sudo apt-get update
2 sudo apt-get upgrade
3 sudo apt-get install git
4 sudo apt-get install cmake
5 sudo apt-get install libusb-1.0-0.dev
6 sudo apt-get install build-essential

```

Sind diese Paketabhängigkeiten installiert, kann die Software zum Streaming der IQ-Daten des RTL-SDR über TCP mittels GitHub installiert werden. Die notwendigen Befehle sind im Listing 6.2 zu sehen.

Listing 6.2: Installation eines RTL-SDR Server [21]

```

1 git clone git://git.osmocom.org/rtl-sdr.git
2 cd rtl-sdr/
3 mkdir build
4 cd build
5 cmake ../
6 make
7 sudo make install
8 sudo ldconfig

```

Nach erfolgreicher Installation wird bei eingestecktem DVB-T-Stick der Befehl "rtl_test -t" auf dem Terminal abgesetzt. Auf dem Bildschirm erscheint nun eine kurze Information über den Stick, zum Beispiel welcher Tuner verbaut ist und auf welche Verstärkung die einzelnen Verstärkerstufen eingestellt sind. Das Programm, das von eigentlichem Interesse ist, ist das Programm "rtl_tcp". Das Streaming der digitalen IQ-Daten des RTL-SDRs erfolgt über TCP/IP. Dazu muss im Terminal der Befehl "rtl_tcp -a", gefolgt von der IP-Adresse, über die die Daten gestreamt werden sollen, abgesetzt werden. Der Raspberry Pi streamt dann kontinuierlich die Daten über die eingegebene IP-Adresse und das empfangenseitige Programm ist nun in der Lage diese Daten auszuwerten. Zudem sind, wie in Tabelle 6.1 ersichtlich, weitere Optionen beim Aufruf des Programms möglich, um weitere Parameter einzustellen.

Option	Bedeutung
-a	IP-Adresse
-p	Port
-f	Mittenfrequenz
-g	Verstärkung
-s	Abtastrate
-b	Anzahl von Puffern

Tabelle 6.1: Optionen für den Aufruf des "rtl_tcp"-Programms

Da die durch elektrische Geräte emittierte Störstrahlung im Institut aufgrund des höchst empfindlichen Radioteleskops ein Problem ist, wurde in einer speziellen Messkammer die vom Raspberry Pi emittierte Strahlung gemessen. Dabei wurde ein DVB-T-Stick, wie er im späteren Verlauf verwendet wird, an den Raspberry Pi angeschlossen. Zudem wurde die Störstrahlung der Spannungsversorgung über das Netzteil erfasst. Das Ergebnis ist in Abbildung 6.3 zu sehen. Für diese Darstellung wurde eine Referenzmessung durchgeführt, um Einflüsse der Messkomponenten zu eliminieren. Der Grundrauschpegel des Spektrumanalysators beträgt bei dieser Bandbreite knapp -82 dBm.

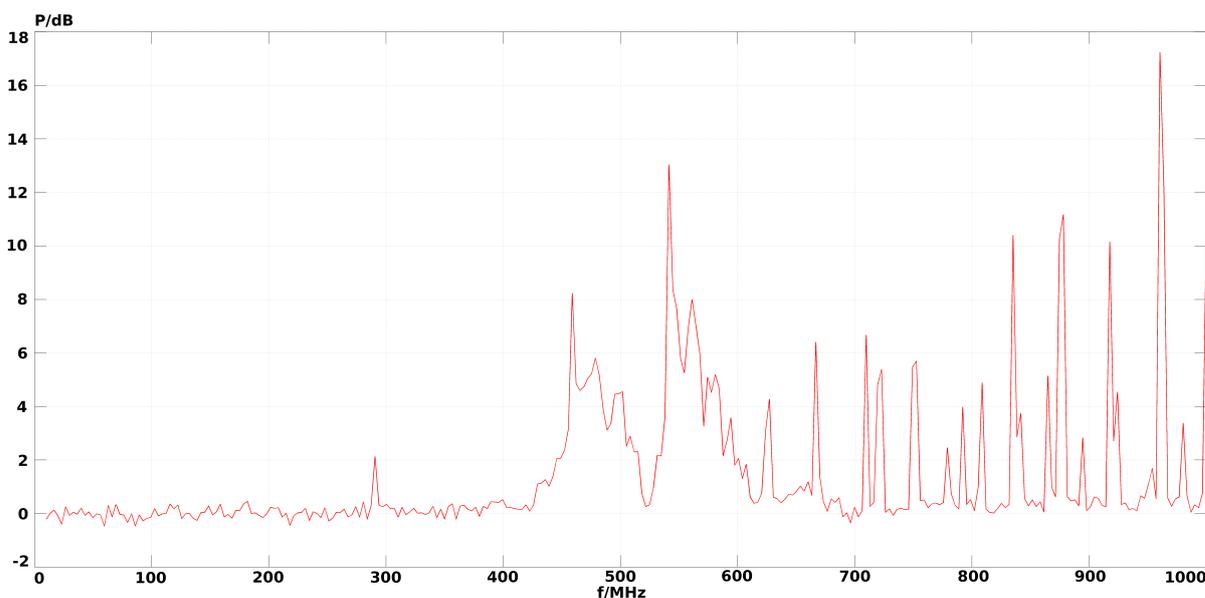


Abbildung 6.3: Emittierte Störstrahlung eines Raspberry Pi mit einem eingesteckten R820T-RTL-SDR zwischen 0 und 1 GHz (normierte Messung, Video-Bandwidth = 100 kHz, Resolution-Bandwidth = 100 kHz)

Über einen großen Frequenzbereich sind wie in Abbildung 6.3 zu sehen deutliche Störlinien vorhanden. Damit diese Störungen die Messungen des Instituts nicht negativ beeinflussen, muss beim späteren Einsatz eine entsprechende Abschirmung des Geräts vorgenommen werden.

Das nächste Kapitel befasst sich mit der Softwareentwicklung eines DFT-Spektrumanalysator mit einem RTL-SDR beziehungsweise einem USRP. Um das Spektrum über eine Bandbreite von bis zu 1 GHz aufzunehmen, wird ein Softwaretool entwickelt. Dabei wird auf weitere Schwierigkeiten und Probleme mit diesen Hardwareinterfaces eingegangen.

7 Entwicklung eines DFT-Spektrumanalysators mit einem SDR

Dieses Kapitel behandelt die Softwareentwicklung eines DFT-Spektrumanalysators auf Basis der in Kapitel 4 dargestellten Theorie zur Erfassung von Signalspektren. Zuvor wurde bereits die Hardware für den Empfang der Signale untersucht. Mit Hilfe dieser Hardware wird ein DFT-Spektrumanalysator entwickelt, der das Spektrum über eine hohe Bandbreite erfasst. Ziel ist es, das Spektrum über den kompletten Frequenzbereich der Zwischenfrequenz zu erfassen, also über eine Bandbreite von 1 GHz. Die Berechnung des Spektrums und die Verarbeitung der Daten werden komplett in Python implementiert. Dabei wird das Tool GNU Radio Companion zur Hilfe genommen und auf Basis eines Flowgraphs die entsprechende Software entwickelt. Nachfolgend werden die einzelnen Schritte in der Entwicklung für ein RTL-SDR sowie für das USRP dargestellt.

7.1 Softwareentwicklung eines DFT-Spektrumanalysators

7.1.1 Ermittlung von breitbandigen Signalspektren mit einem RTL-SDR

7.1.1.1 Erstellung eines GNU Radio-Flowgraphs zur Berechnung der DFT

Um mittels GNU Radio die Signale eines RTL-SDR zu empfangen und weiterzuverarbeiten, wird zu Beginn ein neuer Flowgraph erstellt. Ein leerer Flowgraph stellt grundsätzlich einen Block zur Konfiguration der grafischen Oberfläche (GUI) bereit. Dabei kann zwischen keinem, einer WX- und einem QT-GUI gewählt werden. Für erste Versuche und Entwicklungsschritte wird ein QT-GUI als grafische Oberfläche gewählt. Der für dieses Projekt erstellte Flowgraph ist in Abbildung 7.1 zu sehen.

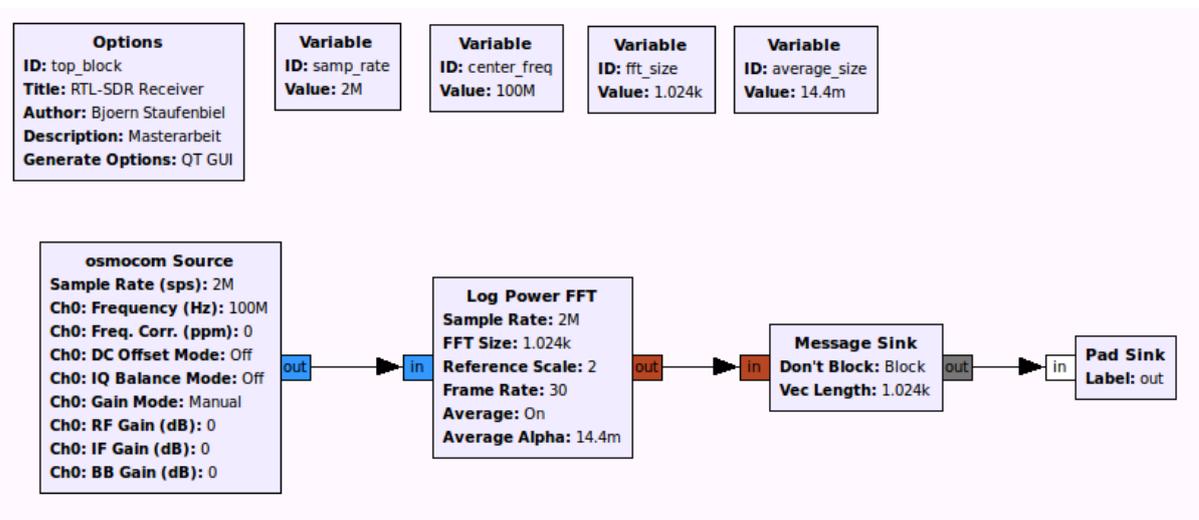


Abbildung 7.1: GNU Radio Flowgraph zur Berechnung der FFT der Empfangssignale eines RTL-SDR

Zwingend notwendig ist die Deklaration einer Variable, die die Abtastrate bestimmt. Diese stellt die Abtastrate ein, auf die die Signale im RTL2832U IC umgesetzt werden. Für dieses Programm wurde eine Abtastrate von 2 MHz gewählt. Bei der Berechnung eines Signalspektrums wird so 1 MHz unterhalb und 1 MHz oberhalb von 0 Hz des IQ-demodulierten Signals erfasst. Daraus resultiert eine Gesamtbandbreite von 2 MHz (siehe Abbildung 7.2).

Um die Signale eines RTL-SDR in GNU Radio zu verarbeiten, wird eine "osmocom Source" in den Flowgraph eingefügt. Dieser GNU Radio Block ist Teil einer Bibliothek des Entwicklers Osmocom. Wie bereits zuvor erwähnt, ist diese Entwicklergruppe die Erste gewesen, die ein Open-Source-Softwareinterface für RTL-SDRs entwickelt hat, welches hier genutzt wird. Der Block stellt einige Systemparameter bereit, die über die Blockeigenschaften eingestellt werden können. Zunächst werden diese auf eine Grundkonfiguration gesetzt und im später entwickelten Python-Programm verändert und an die Anwendung angepasst.

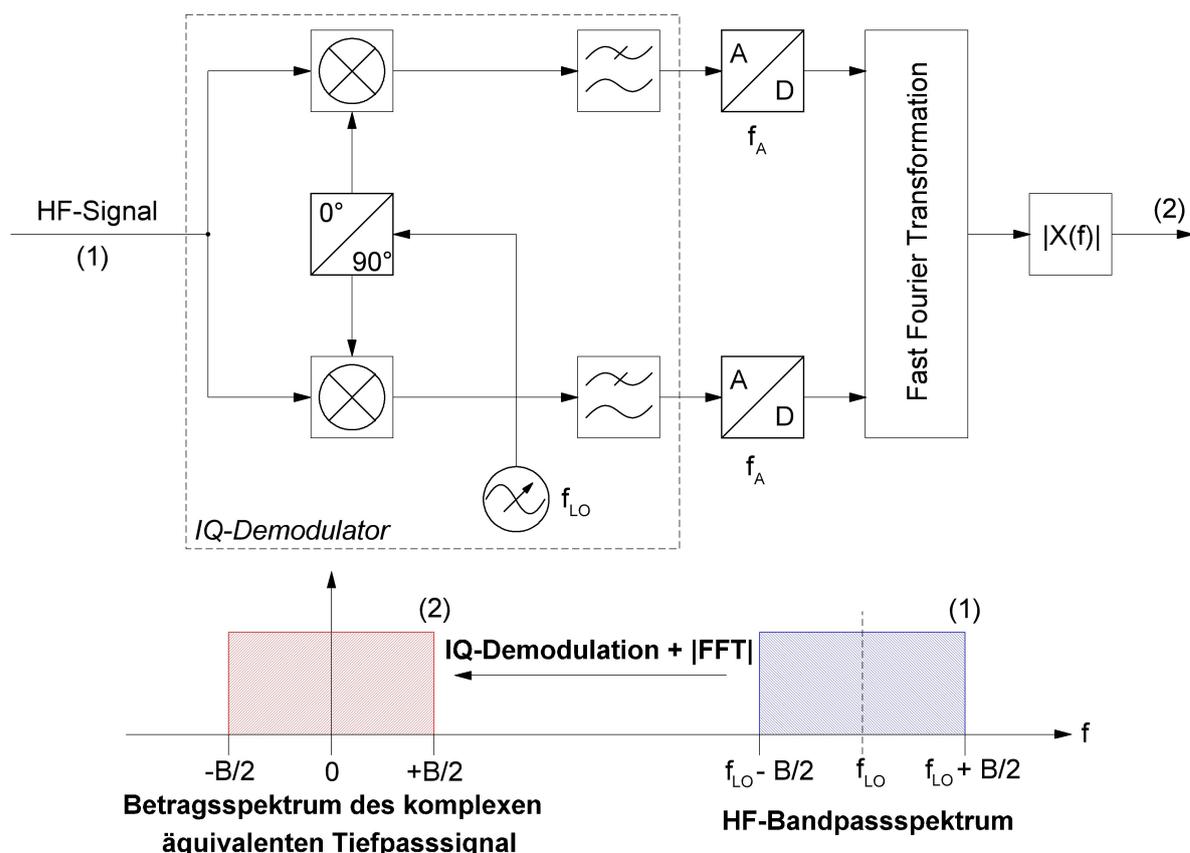


Abbildung 7.2: Betragsspektrum eines IQ-demodulierten HF-Bandpasssignals bei der Wahl einer festen Oszillatorfrequenz f_{LO}

Der Block bietet unter anderem Einstellungen für

- die Abtastrate,
- die zu verwendende Hardware (Device Arguments),
- die Mittenfrequenz,
- die Verstärkung und eine automatische Verstärkungsregelung.

Für die Abtastrate wird der Name des Variablenblocks der Abtastrate angegeben. Die Mittenfrequenz wird beispielhaft auf 100 MHz eingestellt. Die Verstärkung wird auf 0 dB gesetzt und die automatische Verstärkungsregelung deaktiviert. Da der Pegel bei der späteren Anwendung stets hoch ist (um -6 dBm), wird keine weitere Verstärkung benötigt. Alle anderen Parameter werden auf den voreingestellten Werten belassen. Unter dem Punkt Device Arguments wird folgender Eintrag eingefügt:

$$\text{rtLtcp} = 134.104.64.45:1234.$$

Mit diesem Eintrag weiß GNU Radio, dass die Daten über einen TCP-Stream empfangen werden müssen. Dabei wird die IP-Adresse und der Port des RTL-SDR-Servers angegeben. In diesem Fall ist das der zuvor erwähnte Raspberry Pi, an dem das RTL-SDR-Device hängt. Um das Spektrum der über den DVB-T-Stick empfangenen Signale zu berechnen, wird hinter die RTL-SDR-Quelle ein LogPowerFFT-Block geschaltet. Dieser Block berechnet das logarithmierte Betragsspektrum der RTL-SDR-Signale. Das ist besonders praktisch, da so eine zusätzliche Berechnung des Logarithmus entfällt und Zeit- und Rechenaufwand gespart werden kann. Dieser Block hat folgende einstellbare Parameter:

- die Abtastrate,
- die FFT-Größe und
- eine Mittelwertbildung.

Wie bereits bei der RTL-SDR-Quelle wird für die Abtastrate der Name des entsprechenden Variablenblocks eingefügt. Die FFT-Größe bestimmt, wie viele diskrete Punkte der Fouriertransformation für die eingestellte Bandbreite berechnet werden. Das bedeutet, dass bei einem hohen eingestellten Wert, mehr diskrete Frequenzpunkte innerhalb der erfassten 2 MHz Bandbreite berechnet werden, als bei einem kleinen Wert. Die FFT-Länge ist, wie in Kapitel 4 wurde, dabei ein wichtiger Parameter zur Bestimmung der Frequenzauflösung der DFT. Bei Werten ungleich 2^n wird das Signal mittels Zero Padding mit Nullen aufgefüllt. Wichtig ist zudem, dass die Auflösung hoch genug gewählt, sodass es bei der Zusammensetzung von Teilspektren zu keiner Überschneidung von diskreten Frequenzpunkten kommt [40].

Dieser Block ermöglicht zudem eine Mittelwertbildung des Spektrums. Das hat den Vorteil, dass der Einfluss von zufälligem Rauschen im Spektrum verkleinert wird. Hierbei sei vor allem auf das nicht vermeidbare Quantisierungsrauschen verwiesen. Dieses Rauschen entsteht durch die Quantisierung der Abtastwerte im RTL2832U IC. Im Falle eines RTL-SDR werden die Eingangssignale mit einer Auflösung von 8 Bit quantisiert, also ganzen Zahlen zwischen 0 und 255. Der dadurch resultierende Quantisierungsfehler führt zu Rauschen, das im realen Signal nicht vorhanden ist. Dieses Rauschen ist eine stochastische Größe. Durch eine Mittelwertbildung mehrerer Spektromaufnahmen kann dieses reduziert werden. Das erleichtert vor allem die Erkennung von kleinen Nutzsignalen im Rauschen. Der Nachteil einer solchen Mittelwertbildung ist, dass dadurch weiterer Rechenaufwand entsteht. Durch diesen zusätzlichen Rechenaufwand können Signale, die nur kurz im Spektrum auftauchen verloren gehen. Das ist für die spätere Anwendung ein großer Nachteil, da während der Bewegung des Radioteleskops Störsignale nur kurz im Spektrum auftauchen und wieder verschwinden können. Daher wird nur eine geringe Mittelwertbildung eingestellt, um das Quantisierungsrauschen im Spektrum zu minimieren und trotzdem schnell genug zu sein, um kurz auftauchende Signale zu erkennen. Die Abbildung 7.3 zeigt schematisch einen Vergleich verschiedener Einstellungen der Mittelwertbildung.

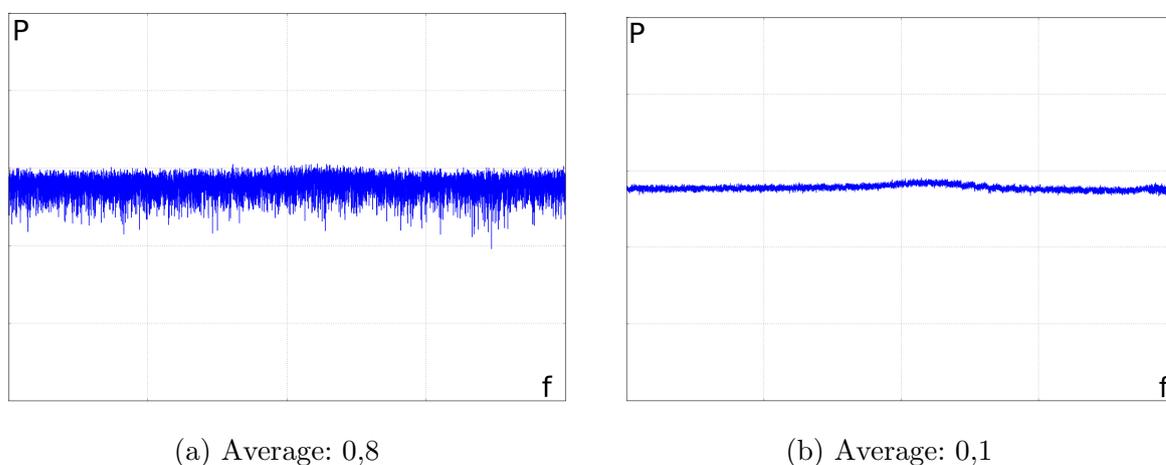


Abbildung 7.3: Vergleich verschiedener Average-Einstellungen des LogPowerFFT-Blocks

Der Vergleich verschiedener Einstellungen der Mittelwertbildung, welche in Form einer Zeitkonstanten zwischen 0 und 1 angegeben wird, zeigt, dass kleine Nutzschnale nur mit einer Mittelwertbildung der Signale erkennbar werden. Trotzdem ist das Eigenrauschen des RTL-SDR weiterhin hoch. Zudem verhindert begrenzt die A/D-Wandlung das maximale Signal-Rausch-Verhältnis auf maximal 50 dB.

Weiterhin ist im Flowgraph eine Message Sink notwendig. Diese ermöglicht es, den Vektor, den der FFT-Block erzeugt direkt in ein Array zu speichern. In Python gibt es den eigentlichen Typen Array nicht mehr, sondern sogenannte Listen. Diese Listen sind deutlich flexibler als zum Beispiel C-Arrays und bieten umfangreiche Methoden zur Manipulation an. Prinzipiell arbeiten beide Datentypen aber auf eine ähnliche Weise. Diese Senke macht es so möglich, die diskreten FFT-Werte direkt im Programm weiterzuverarbeiten. Da dieser Block einen Ausgang bietet, wird eine Pad Sink an die Message Sink geschaltet. Diese Senke dient nur als Abschluss des Flowgraphs und bietet keine weitere Funktion. Ohne diesen Block würde GNU Radio Companion bei der Kompilierung einen Fehler erzeugen. Im späteren Python-Programm wird dieser Block wieder gelöscht, da er keine für diese Anwendung nützliche Funktion bietet.

7.1.1.2 Python-Softwareentwicklung eines breitbandigen Spektrumanalysators

Um ein breitbandiges Spektrum mittels eines RTL-SDR zu berechnen und in einem QT-GUI anzuzeigen, sind mehrere Schritte notwendig. Doch bevor die eigentliche Berechnung des Spektrums betrachtet wird, wird ein Rahmenprogramm gezeigt, welches für die Entwicklung genutzt wird. Dieses implementiert auf der einen Seite die grafische Oberfläche zur Kontrolle der Ergebnisse, aber auch die Kommunikation mit dem Raspberry Pi und die Erstellung der GNU Radio Blöcke. Der prinzipielle Ablauf des Rahmenprogramms ist in Abbildung 7.4 zu sehen. Der orange markierte Block zur Berechnung des breitbandigen Signalspektrums wird im Nachfolgenden detaillierter betrachtet und wird durch verschiedene Programme immer wieder ausgetauscht, um eine zuverlässige spektrale Darstellung zu erzielen. Dieses Rahmenprogramm dient, sofern nicht anders angegeben, als Grundlage für alle Programme in diesem Kapitel.

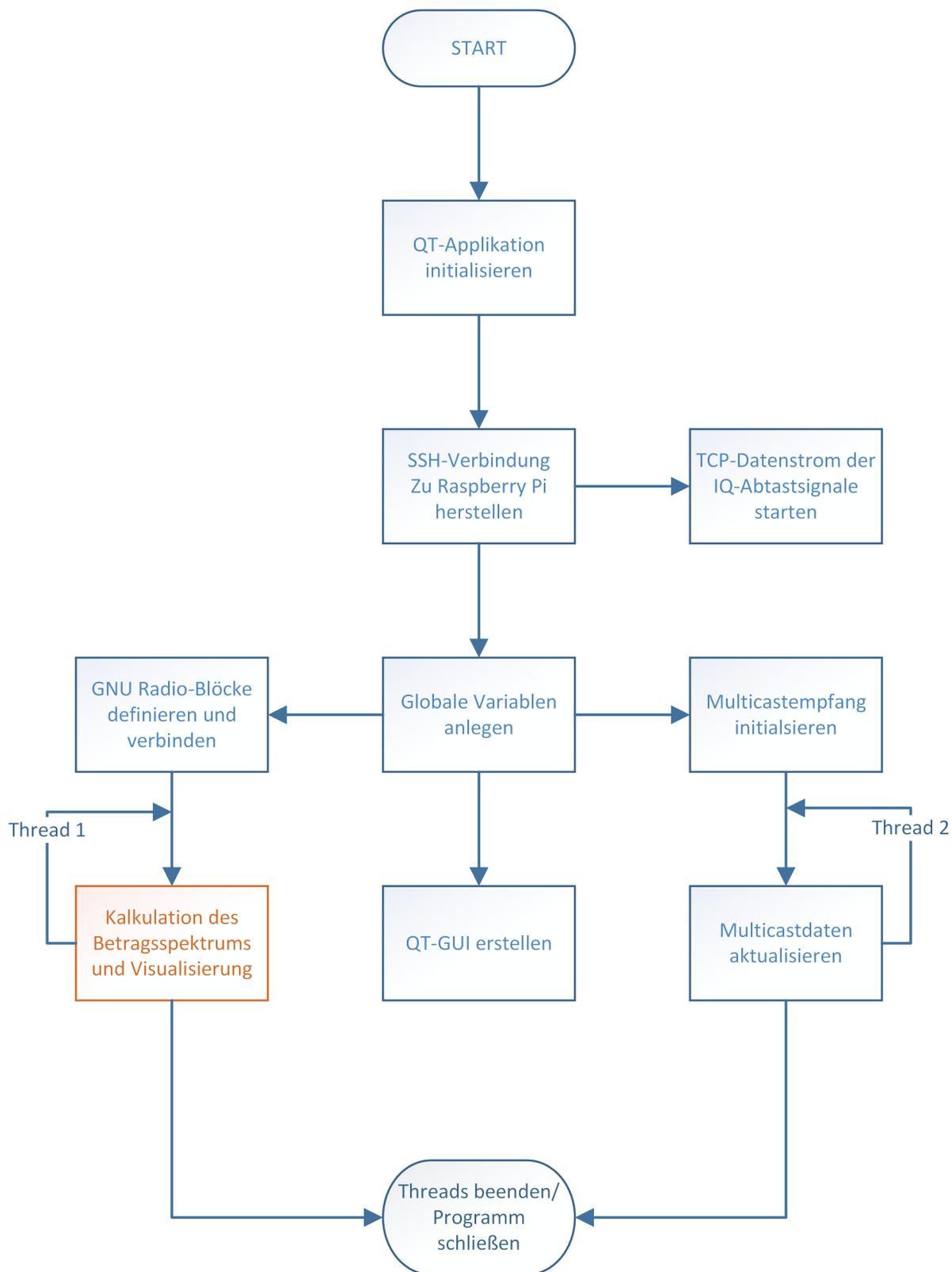


Abbildung 7.4: Rahmenprogramm zur Berechnung der FFT der RTL-SDR-Zeitsignale

Beim Start der Anwendung wird die eigentliche QT-Applikation initialisiert. Dies geschieht mithilfe einer Methode direkt nach dem Programmstart. Anschließend wird eine SSH-Verbindung zum Raspberry Pi aufgebaut. Über diese Verbindung wird das Programm "rtl_tcp" auf dem Raspberry Pi gestartet. Dadurch werden die Daten des RTL-SDR über TCP gestreamt. Nachfolgend werden verschiedene Variablen angelegt, zum Beispiel für die Abtastrate und die Mittenfrequenz des RTL-SDR. Danach werden drei voneinander getrennt betrachtbare Vorgänge gestartet.

Auf der einen Seite wird das QT-GUI erstellt. Dabei werden verschiedene Textfelder und Eingabefelder auf der grafischen Oberfläche platziert. Diese dienen als Interface für den Anwender. Auf der anderen Seite werden die Blöcke des zuvor erstellten Flowgraphs im Python-Programm erstellt und initialisiert. Dabei wird über eine Connect-Funktion die Verbindung zwischen den Blöcken hergestellt, um die Übertragung der Daten zwischen den Blöcken zu implementieren. Als Drittes wird eine Verbindung zum Multicast, welches über das interne Netzwerk aktuelle Informationen zum Status des Radioteleskops sendet. Darin sind verschiedene Parameter enthalten, wie zum Beispiel der aktuelle Elevationswinkel, der aktive Empfänger oder auch die aktuelle Frequenz der Lokaloszillatoren der Mischstufen zur Frequenzkonversion. Während des eigentlichen Programmablaufs werden verschiedene Parameter in der grafischen Oberfläche angezeigt, welche über einen parallel laufenden Thread jede Sekunde aktualisiert werden. So lässt sich zum Beispiel die Bewegung des Radioteleskops nachverfolgen.

Die Abbildung 7.6 zeigt den Programmflussplan des ersten Entwurfs für die Berechnung eines breitbandigen Spektrums mit einem RTL-SDR. Über Eingabefelder kann der Benutzer den gewünschten Frequenzbereich eingeben, der erfasst werden soll. Der Bereich wird in Form einer Start- und Stoppfrequenz angegeben. Aus der Startfrequenz wird die erste Mittenfrequenz des RTL-SDR berechnet. Wird zum Beispiel eine Startfrequenz von 50 MHz eingegeben, dann wird die Mittenfrequenz des RTL-SDR auf 51 MHz eingestellt. Dieses berechnete Spektrum wird in eine Liste gespeichert. Die Mittenfrequenz wird daraufhin um 2 MHz erhöht. Dabei ist zu beachten, dass die Frequenzungenauigkeit des RTL-SDR mit bis 100 ppm relativ hoch ist. Diese Ungenauigkeit hat zur Folge, dass sich spektrale Anteile beim Zusammenfügen wiederholen können oder Lücken entstehen. Da auf dieses Verhalten kein direkten Einfluss genommen werden kann und die absolute Frequenzabweichung für die eingestellten Frequenzen klein ist, wird dieser Umstand außen vor gelassen. Wenn die Stoppfrequenz noch nicht abgedeckt wurde, dann wird für die neue Mittenfrequenz erneut das Spektrum berechnet. Die Daten werden an die zuvor erstellte Liste mit den FFT-Daten angehängt. Dieser Vorgang wird so lange wiederholt, bis die Stoppfrequenz erreicht wurde. Ist das der Fall, wird die Liste, in der die FFT-Daten gespeichert sind, mittels Matplotlib geplottet.

Matplotlib ist eine Bibliothek für Python, die zum wissenschaftlichen Plotten von Daten bestimmt ist. Die Funktionsaufrufe ähneln dabei den Befehlen von MATLAB. Das breitbandige Spektrum wird so innerhalb dem QT-GUI angezeigt. Aufgrund der Nutzung dieser Bibliothek in vielen Programmen des Instituts wird diese hier verwendet.

Der erste Test bestand darin, einen einfachen Dipol an den DVB-T-Stick anzuschließen. Anschließend wurde mit dem entwickelten Breitband-Spektrumanalysator das Spektrum innerhalb eines Ausschnittes des UKW-Bandes berechnet. Dabei war vor allem der Empfang verschiedener Radiostationen von Interesse. Das Ergebnis ist in Abbildung 7.5 zu sehen.

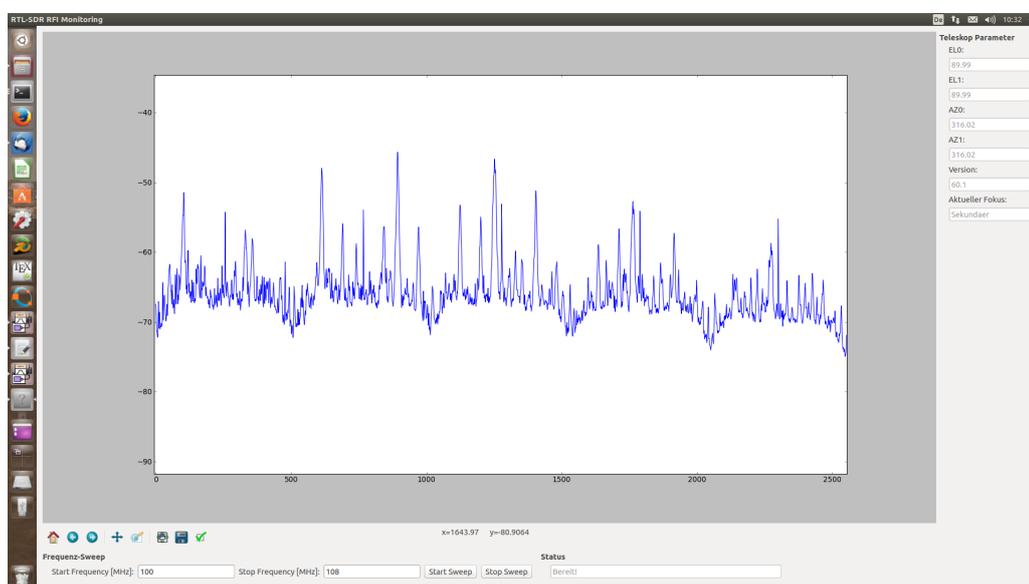


Abbildung 7.5: Breitbandige Spektromaufnahme des UKW-Rundfunkbandes zwischen 98 und 108 MHz

Die Abbildung zeigt Probleme, die bei der Berechnung des Spektrums mit einem RTL-SDR entstehen. Auf der einen Seite wäre der Abfall des Spektrums an den Flanken jedes Teilspektrums zu beachten. Das heißt, dass bei jedem 2 MHz breiten Teilspektrum der Pegel des Spektrums am Anfang und am Ende deutlich sichtbar abfällt. Ein Grund für dieses Verhalten ist die Überlappung von Spektren bei der Abtastratenkonversion aufgrund der endlich steilen Flanken der Tiefpassfilter (siehe Kapitel 4). Daraus resultiert eine Verzerrung des Spektrums an den Rändern. Außerdem gibt es ein starkes LO-Leakage. Das bedeutet, dass die Frequenz, auf der der Lokaloszillator schwingt, auf den Eingang zurückgekoppelt wird. Dadurch wird das Lokaloszillatorsignal im Spektrum sichtbar. Diese Linien sind zwischen dem eigentlichen Nutzsignal gut zu erkennen. Diese Probleme verlangten daher nach einer softwaremäßigen Lösung, welche im nachfolgenden Abschnitt dargestellt wird.

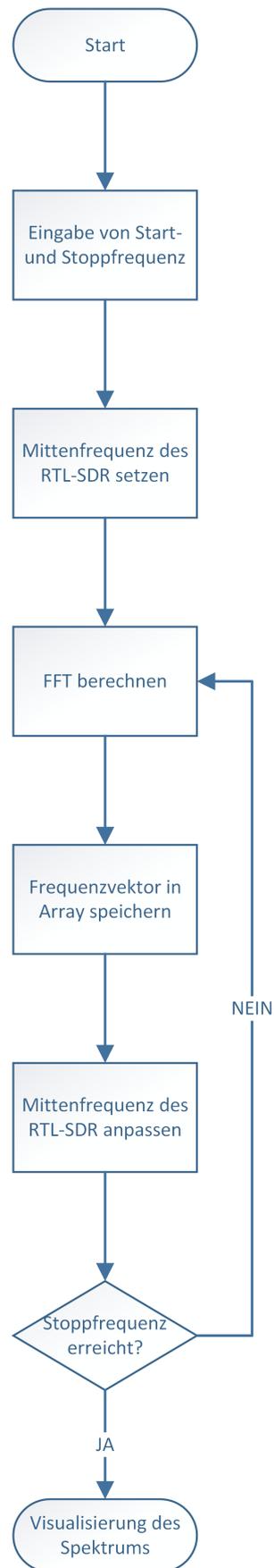


Abbildung 7.6: Erste Version des Programms zur Berechnung der FFT der RTL-SDR-Zeitsignale

7.1.1.3 Anpassung der Signalaufnahme

Die Problem in der Darstellung eines breitbandigen Spektrums werden softwaremäßig umgangen, da diese in der Hardware entstehen und aufgrund der hohen Integration der einzelnen Bauteile eine Einflussnahme auf die Schaltung nicht möglich ist. Die Idee ist, immer nur einen kleinen Teil der 2 MHz breiten Teilspektren auszuschneiden und zu speichern. Die Mittenfrequenz des RTL-SDR wird anschließend nur um die Breite dieses Ausschnitts erhöht und der nächste Ausschnitt gespeichert. Das Vorgehen ist in Abbildung 7.7 schematisch dargestellt.

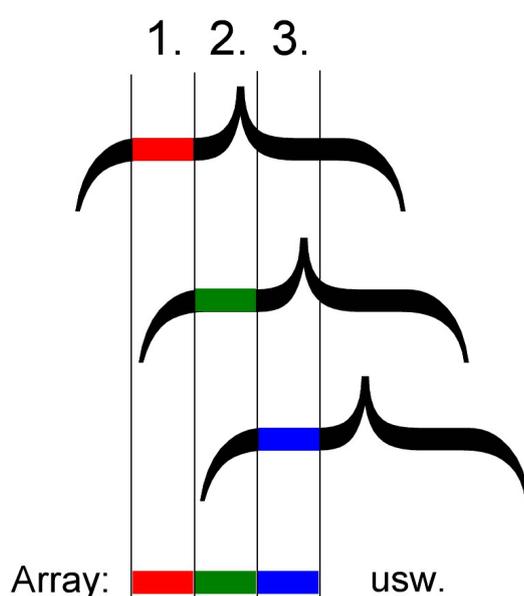


Abbildung 7.7: Prinzip zur Aufteilung der Spektromaufnahme zur Vermeidung von Nulllinien und Schultern im Spektrum

Die abfallenden Flanken und die Nulllinie in der Mitte eines Teilspektrums können abstrakt wie eine liegende, geschweifte Klammer betrachtet werden. Wie die Darstellung zeigt, wird aus diesem Teilspektrum nur der Teil ausgeschnitten, der linear verläuft und weit genug vom LO-Leakage entfernt ist. Dieser Ausschnitt wird wie zuvor in einer Liste gespeichert. Im nächsten Schritt wird die Mittenfrequenz des RTL-SDR so erhöht, dass wenn aus dem nächsten Teilspektrum ein Ausschnitt ausgeschnitten wird, dieser wiederum so an die Liste angehängt wird, dass kein spektraler Fehler entsteht. In diesem Fall wurde ein 500 kHz breiter Ausschnitt gewählt und durch Experimente als sinnvoll bestätigt. Dieser beginnt bei 250 kHz und endet bei 750 kHz, relativ vom niederfrequenten Anfang eines Teilspektrums gesehen.

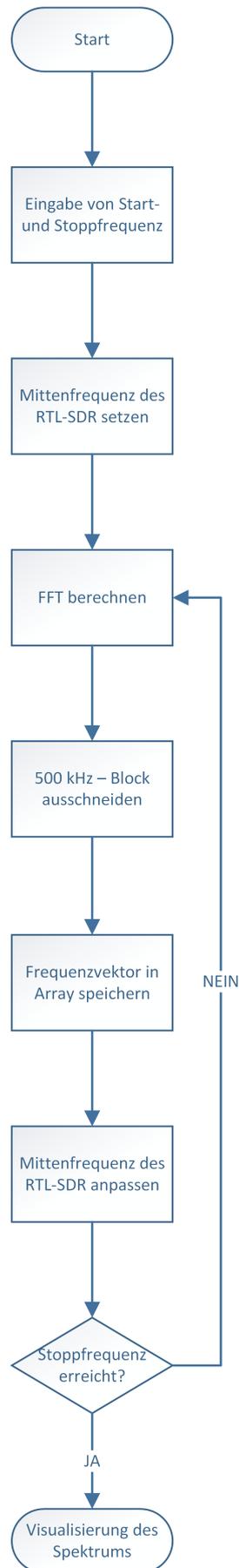


Abbildung 7.8: Ablauf des Programms zur Berechnung des breitbandigen Spektrums mittels eines RTL-SDR

Das Programm zur Berechnung der FFT muss entsprechend an die oben dargestellte Idee angepasst werden. Dazu wird nach der Berechnung eines Teilspektrums ein Ausschneiden eines 500 kHz breiten Ausschnitts hinzugefügt. Der angepasste Programmflussplan ist in Abbildung 7.8 zu sehen.

Um die Funktion des neuen Programms zu überprüfen, wird der in Abbildung 7.9 dargestellte Versuchsaufbau durchgeführt. Dazu wird ein einfaches Trägersignal auf den Eingang des RTL-SDR gegeben. Zur Kontrolle wird das Signal über einen Leisterteiler auf einem Spektrumanalysator betrachtet. Das Trägersignal hat eine Frequenz von 850 MHz und einen Pegel von -80 dBm am Eingang des RTL-SDR. Das Breitbandpektrum über einen Bereich zwischen 840 und 860 MHz ist in Abbildung 7.10 zu sehen.

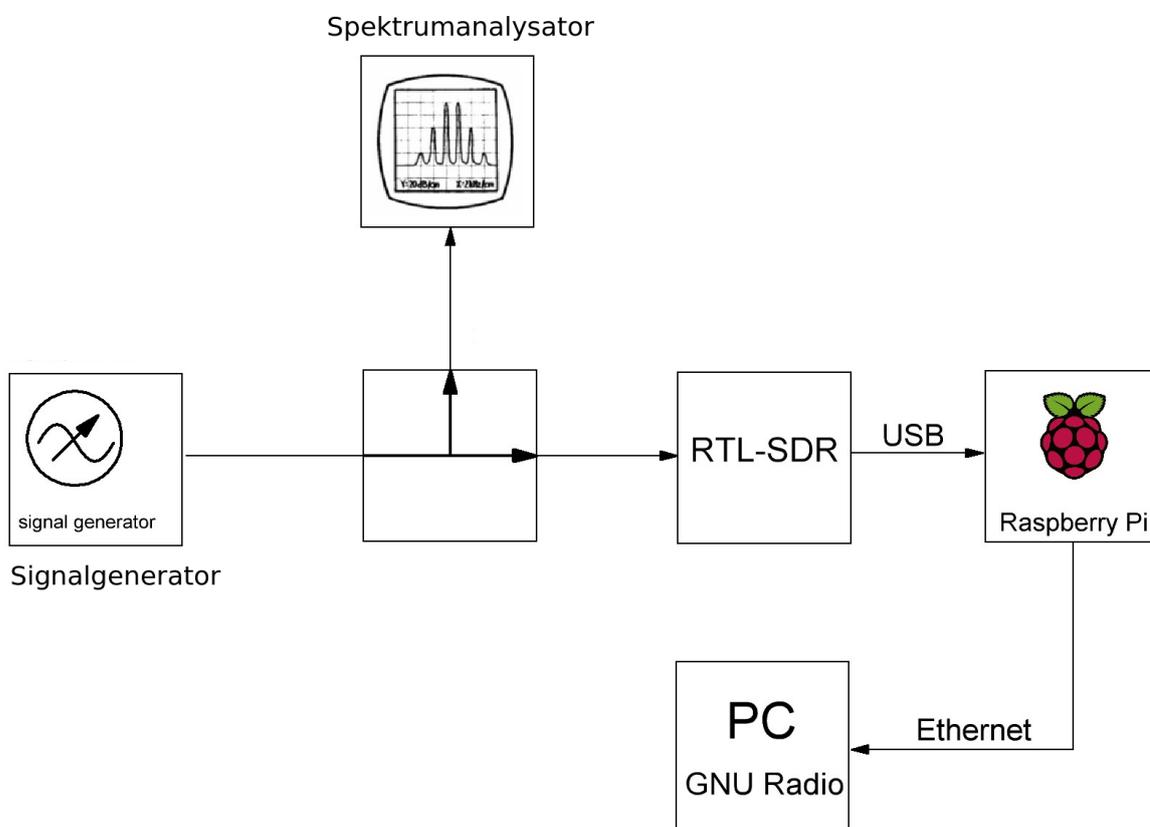


Abbildung 7.9: Versuchsaufbau zur Aufnahme des breitbandigen Spektrums eines einzelnen Trägers

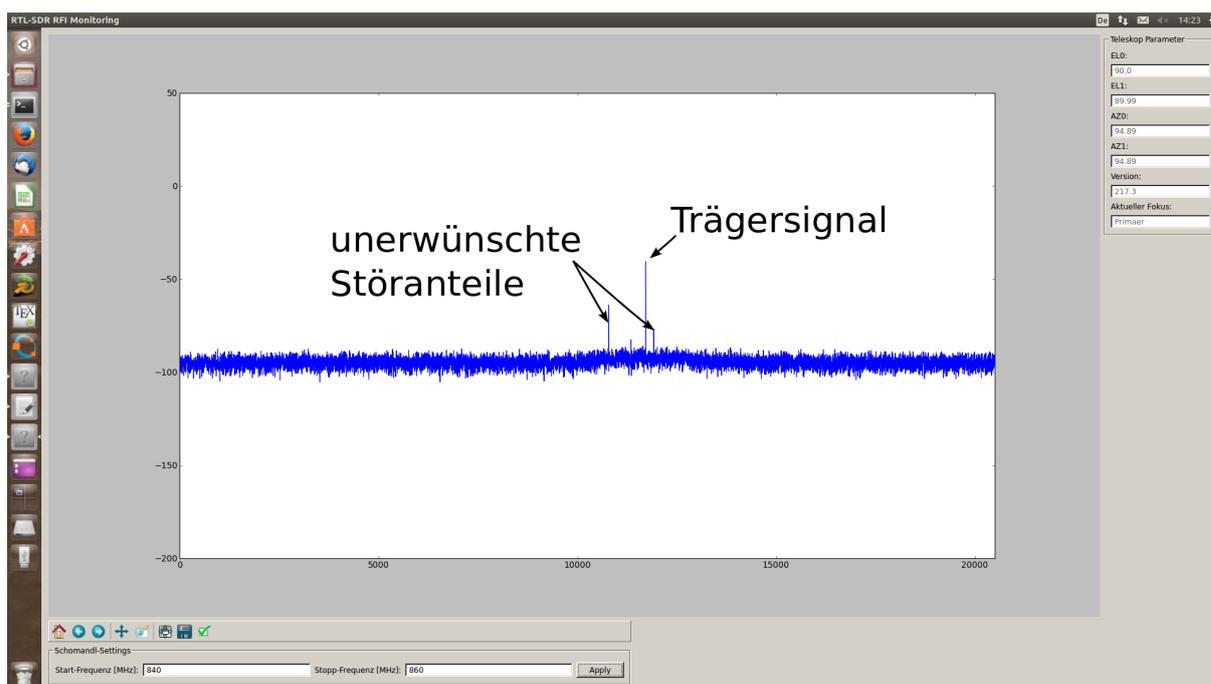


Abbildung 7.10: Breitbandige Spektromaufnahme eines einzelnen Trägers im Softwareinterface (Bandbreite = 20 MHz)

Die Aufnahme des breitbandigen Spektrums zeigt ein weiteres Problem bei der Berechnung. Denn in vielen Frequenzbereichen erzeugt der DVB-T-Stick eigene Störlinien. Dies führt dazu, dass nicht zwischen Eigenstörungen und Nutzsignalen unterschieden werden kann. Gerade im Zusammenhang mit der Erfassung von Störsignalen ist dies ein schwerwiegender Nachteil. Daher wird im nächsten Abschnitt eine weitere Methode dargestellt, um diese Schwierigkeiten zu umgehen.

7.1.1.4 Verwendung einer zusätzlichen Zwischenfrequenz zur Frequenzkonversion

Die zuvor gezeigten Probleme mit störungsbehafteten Frequenzbereichen führten dazu, dass eine weitere Methode zur korrekten und zuverlässigen spektralen Darstellung gefunden werden musste. Das Konzept hinter der nachfolgend dargestellten Idee ist, alle zu erfassenden Signale in einen Zwischenfrequenzbereich umzusetzen. Das heißt, die Eingangssignale werden alle auf eine feste Zwischenfrequenz gemischt. Das hat den Vorteil, dass ein störungsfreier Frequenzbereich des RTL-SDR ausgewählt werden kann und mittels eines Mischers und eines steuerbaren Lokaloszillators alle Signale in diesen Frequenzbereich gemischt werden können. Die Idee entspricht dem klassischen Aufbau eines Spektrumanalysators [33].

Die Abbildung 7.11 zeigt den prinzipiellen Aufbau dieser Methode. Das Eingangssignal, hier in Form eines Signalgenerators, wird zusammen mit einem Lokoszillatorsignal auf einen Mischer gegeben. Verwendet wurde dabei ein Anzac MD-141 Mischer. Wichtig ist, dass das Signal des Lokoszillators einen Pegel von +7 dBm haben muss. Das Eingangssignal darf höchstens einen Pegel von 0 dBm haben, da der Mischer sonst in die Kompression kommt [3, S. 72]. Das Ausgangssignal des Mixers ergibt sich zu

$$f_{ZF} = | f_{LO} \pm f_{IN} |, \quad (7.1)$$

wobei die ZF-Frequenz konstant bleibt und die Frequenz des Lokoszillators variiert wird. Verschiedene Frequenzanteile des Eingangssignals werden auf eine feste ZF-Frequenz gemischt. Hierbei sei angemerkt, dass in dieser Anwendung das obere Seitenband der Mischung gewählt wurde. Da sich das obere Seitenband nicht in Kehrlage befindet, ist die korrekte Berechnung und Darstellung in Python einfacher. Das gemischte Signal wird anschließend auf das RTL-SDR gegeben. Die Frequenz der Frequenzdekade, sprich des Lokoszillators, wird mithilfe eines Rechners über eine RS-232-Schnittstelle gesteuert.

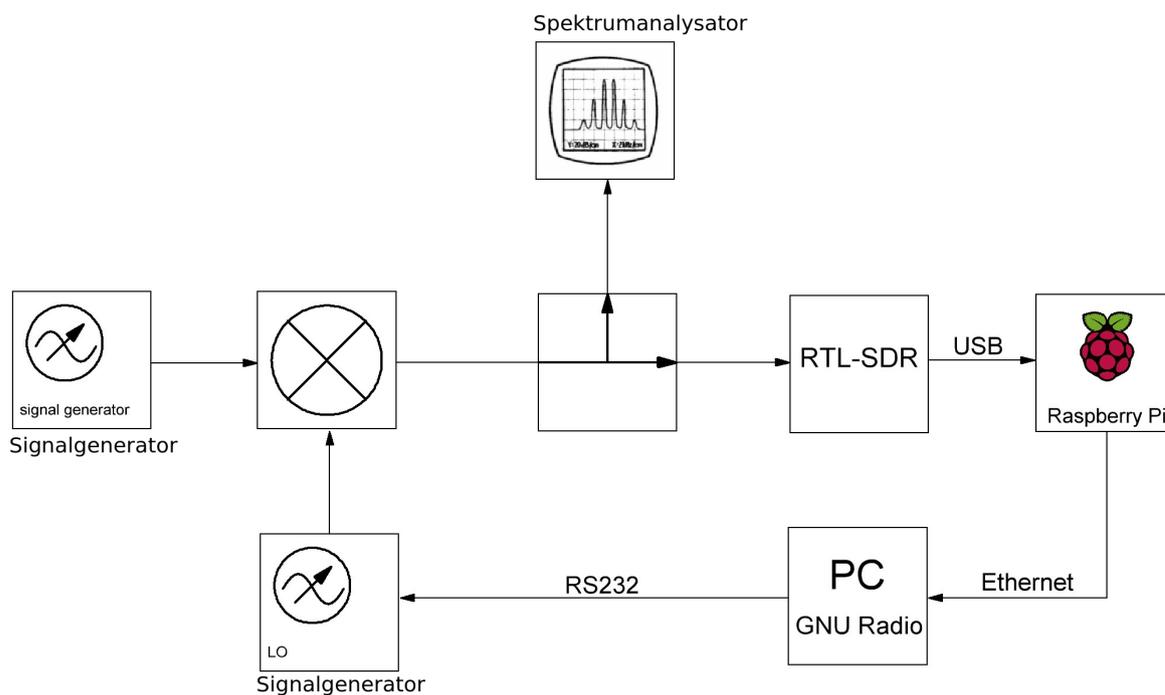


Abbildung 7.11: Aufbau zur Aufnahme eines breitbandigen Spektrums mit einer Zwischenfrequenz

Der Ablauf des neuen Programms ist in Abbildung 7.13 dargestellt und wird nachfolgend diskutiert. Wie zuvor hat der Benutzer die Möglichkeit eine Start- und eine Stoppfrequenz einzugeben. Die Mittenfrequenz des RTL-SDR wird auf eine feste Frequenz eingestellt, in diesem Fall auf 750 MHz. In dieser Arbeit wurde eine Schomandl Frequenzdekade als Lokaloszillator eingesetzt. Diese Frequenzdekade lässt sich über eine RS-232-Schnittstelle fernsteuern. Da diese Schnittstelle nicht praktisch für die Überbrückung von großen Strecken ist, wurde ein RS-232-Ethernet-Konverter eingesetzt, welcher die Befehle über TCP entgegennimmt und über RS-232 zum Gerät weiterleitet. Aus den Start- und Stoppfrequenzen werden die entsprechenden Mischfrequenzen berechnet, welche nötig sind, um den gewünschten Frequenzbereich in die feste Zwischenfrequenz zu mischen. Das Programm läuft dann ähnlich zu den vorhergegangenen ab. Doch anstatt nach jeder Berechnung eines Teilspektrums die Frequenz des RTL-SDR zu ändern, wird die Frequenz des Lokaloszillators entsprechend angepasst.

Der Versuch aus dem vorhergehenden Abschnitt mit einem Trägersignal als Nutzsignal wird mit dem erneuerten Programm ein weiteres Mal durchgeführt. Der Versuchsaufbau wurde bereits in Abbildung 7.11 gezeigt. Das Ergebnis der Messung ist in Abbildung 7.12 zu sehen.

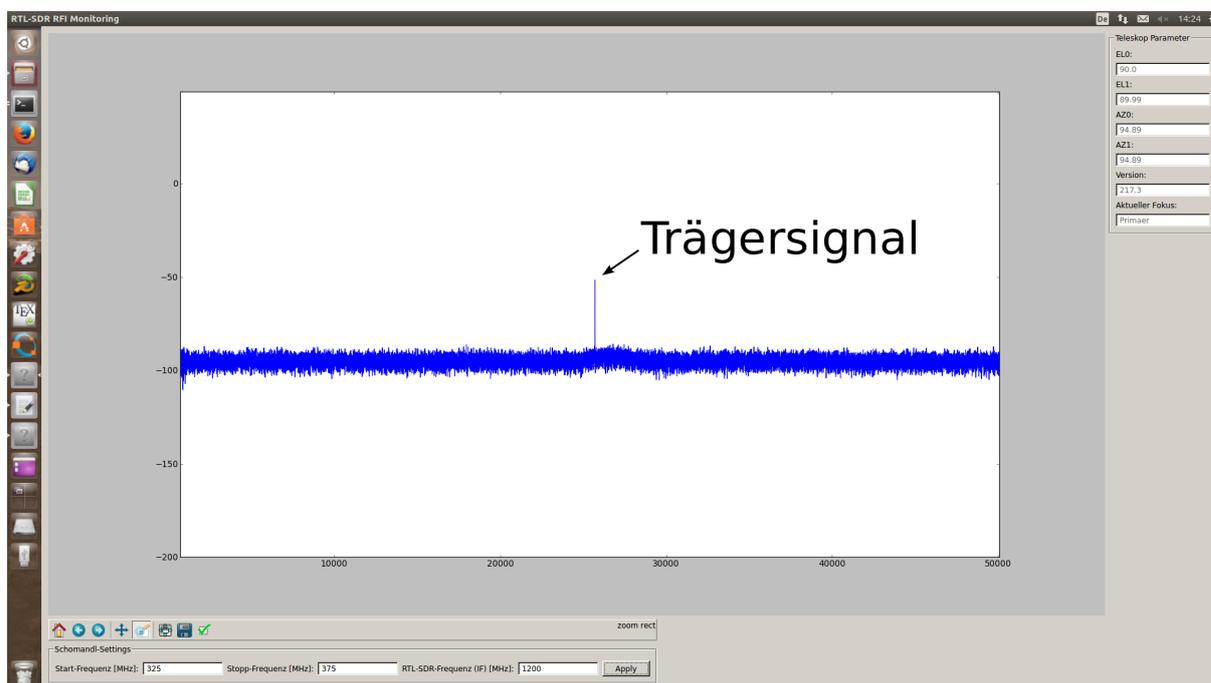


Abbildung 7.12: Breitbandige Spektrumsaufnahme eines einzelnen Trägers mit einer Zwischenfrequenz im Softwareinterface (Bandbreite = 20 MHz)

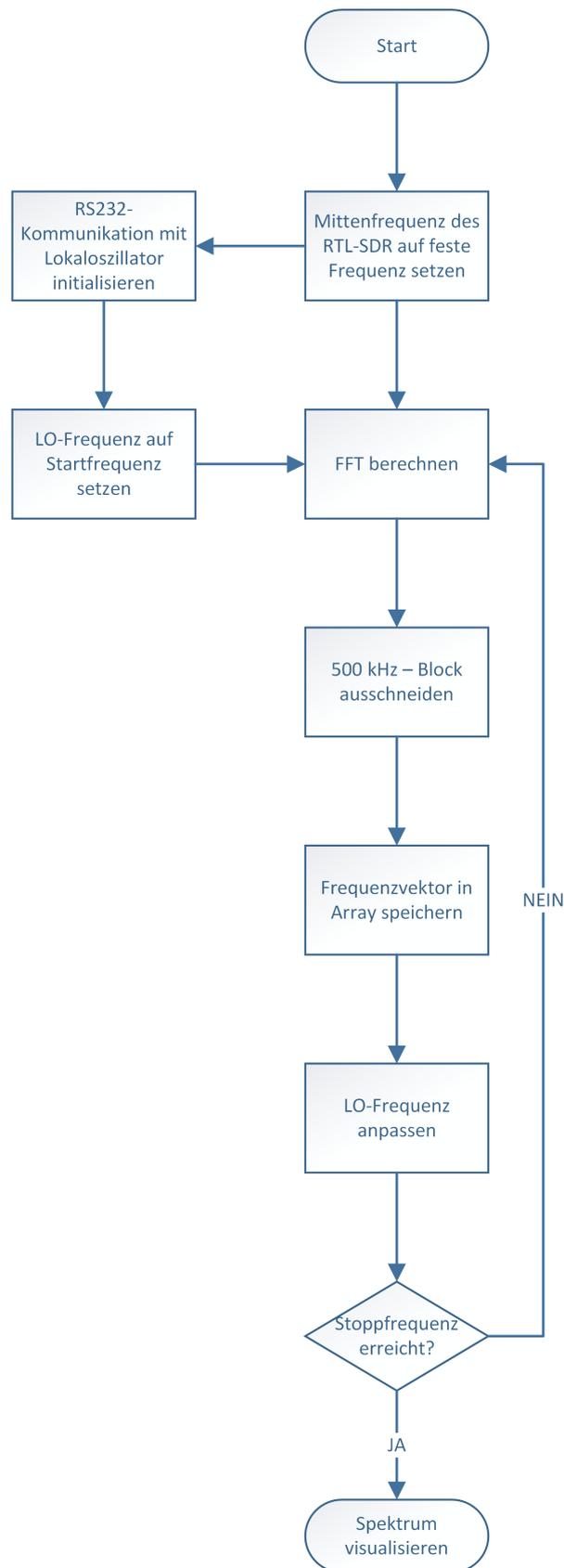


Abbildung 7.13: Ablauf des Programms zur Berechnung des breitbandigen Spektrums mittels eines RTL-SDR

Beim Vergleich der Darstellung mit der zuvor aufgenommenen (siehe Abbildung 7.10) ist zu erkennen, dass das Spektrum keine Störungen aufweist. Das heißt, dass neben dem eigentlichen Nutzsignal keine weiteren Störsignale zu erkennen sind. Das bestätigt die Annahme, dass mit diesem Versuchsaufbau eine deutlich bessere Darstellung des Spektrums über einen breiten Frequenzbereich möglich ist.

Die Versuche mit einer zusätzlichen Zwischenfrequenz haben gezeigt, dass mit diesem Mehraufwand eine saubere Darstellung eines breiten Frequenzbereichs möglich ist. Jedoch ist zu bedenken, dass der Aufwand an Hard- und Software deutlich gestiegen ist. Zudem sind die Kosten für eine Frequenzdekade sehr hoch, sodass über eine Hardware nachzudenken ist, die äquivalente Ergebnisse erzielt und keinen zusätzlichen Hardwareaufbau benötigt. Im nächsten Abschnitt wird die Softwareentwicklung eines äquivalenten Systems mit einem USRP betrachtet.

7.1.2 Nutzung eines USRP als DFT-Spektrumanalysator

7.1.2.1 Anpassung des GNU Radio-Flowgraphs

Die Abbildung 7.14 zeigt den für das USRP angepassten Flowgraph. Der einzige Block, der verändert wurde, ist die Signalquelle. Diese wurde durch einen USRP Source-Block ersetzt. Da die maximale Abtastrate des USRP deutlich höher ist, wurde diese entsprechend auf 20 MS/s angepasst. Da nun das Spektrum in einem größeren Frequenzbereich aufgenommen wird, muss die FFT-Länge angepasst werden, um eine entsprechende Erhöhung der Auflösung der diskreten Frequenzwerte zu erhalten. Das Verhältnis aus Abtastrate zu FFT-Länge war beim RTL-SDR 1 kHz pro diskretem Frequenzpunkt. Für diese Anwendung wurde die FFT-Länge N vorerst auf 4000 eingestellt. Das entspricht einem Verhältnis der Bandbreite zur FFT-Länge von 5 kHz pro diskretem Frequenzpunkt. Das heißt, dass ein diskreter Frequenzpunkt beim USRP fünfmal so viele echte Frequenzen abdeckt als beim RTL-SDR. Bei späteren Messungen ist abzuwägen, ob die FFT-Länge auf einen noch höheren Wert angepasst werden muss.

Wichtig ist zudem, dass bei diesem Block die externe Referenzsignalversorgung aktiviert wird. Nur so ist gewährleistet, dass das angeschlossene 10-MHz- und 1 pps-Referenzsignal genutzt wird. Die Verstärkung des USRPs wird, genau wie beim RTL-SDR, auf 0 dB eingestellt. Die Device-Adresse wird auf eine feste IP-Adresse eingestellt, die dem Gerät zuvor zugewiesen wurde. Da alle anderen Blöcke die FFT-Größe und die Abtastrate als Variable nutzen, nehmen diese den entsprechenden Wert an.

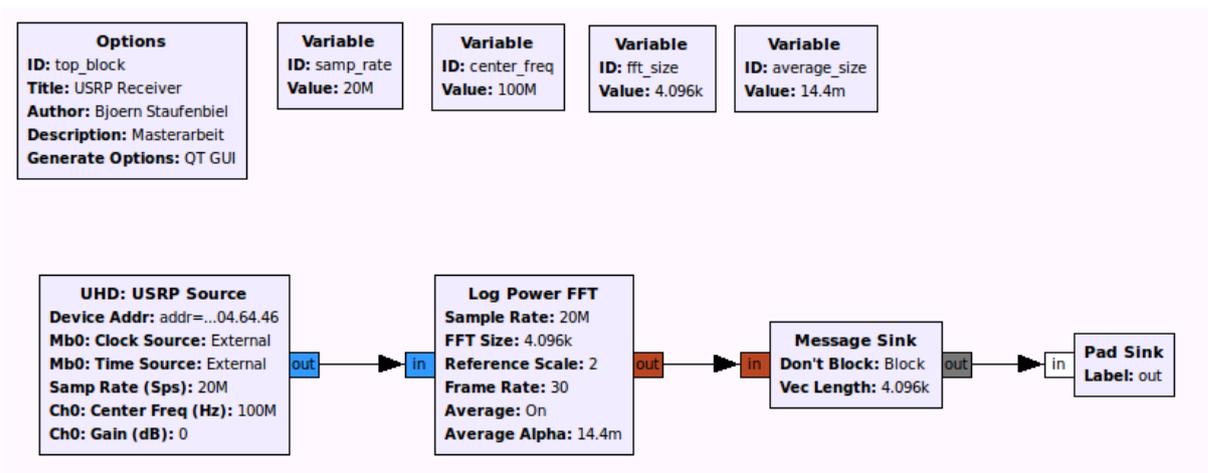


Abbildung 7.14: Angepasster GNU Radio Flowgraph zur Einbindung eines National Instruments USRP-2920 in die RFI-Software

7.1.2.2 Einbindung des USRP in die Python-Software

Der in Abbildung 7.14 gezeigte Flowgraph wird entsprechend in Python umgesetzt. Dazu wird das vorherige Programm weiterverwendet und der RTL-SDR-Block durch den USRP-Block ersetzt. Durch die Erstellung eines Referenzprogramms durch GNU Radio kann die Einbindung eines USRP direkt in das eigene Programm kopiert werden. In der Software müssen entsprechend die Variablen an den Namen der neuen Signalquelle angepasst werden und die neue Abtastrate und FFT-Länge eingetragen werden. Die Kommunikation mit der Frequenzdekade und die Berechnung von Mischfrequenzen entfallen dabei komplett und werden aus dem Programm gestrichen. Zudem muss die Erhöhung der Frequenz des USRP bei der Berechnung eines Spektrums angepasst werden. Der angepasste Programmflussplan ist in Abbildung 7.15 dargestellt.

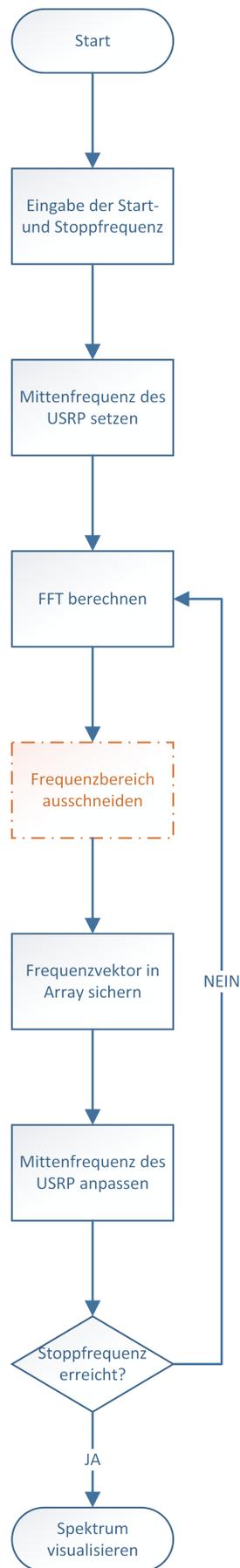


Abbildung 7.15: Angepasster Programmflussplan mit Einbindung eines National Instruments USRP-2920 in die RFI-Software

7.1.2.3 Durchführung erster Testmessungen

Erste Testmessungen führten mit den oben eingestellten Werten für die Abtastrate und die FFT-Länge zum im Listing 7.3 dargestellten Fehler.

Listing 7.3: Fehler bei der Übertragung der USRP-Signale mit einer hohen Abtastrate

```
1 UHD Error:
2     Control packet attempt 0, sequence number 2902:
3     RuntimeError: no control response, possible packet loss
4
5
6 UHD Error:
7     Control packet attempt 0, sequence number 3315:
8     RuntimeError: no control response, possible packet loss
9
10 ...
11
12 Traceback (most recent call last):
13   File "./usrp.py", line 171, in set_start
14     self.usrp.set_center_freq(self.freq_usrp*10e5, 0)
15   File "/usr/local/lib/python2.7/dist-packages/gnuradio/uhd/uhd_swig.py",
16     line 2881, in set_center_freq
17     return _uhd_swig.usrp_source_sptr_set_center_freq(self, *args)
18 DRuntimeError: RuntimeError: fifo ctrl timed out looking for acks
```

Aus diesem Fehler lässt sich ableiten, dass die Übertragung der IQ-Abtastsignale über die Ethernet-Verbindung nicht korrekt funktioniert. Das USRP verwendet ein spezielles Kontrollprotokoll zur Übertragung der Daten. Gehen zu viele Datenpakete im Netzwerk verloren, bricht die Verbindung zum USRP ab und das Programm wird beendet. Das USRP ist über einen Switch angeschlossen und so an das Netzwerk angebunden. Die hohe Abtastrate und das damit verbundene hohe Datenaufkommen führen dazu, dass diese Daten nicht über das Netzwerk aufgenommen werden können und Pakete verloren gehen. Als Konsequenz wird daher für das USRP die Abtastrate auf maximal 2 MS/s begrenzt. Das macht den großen Vorteil des USRP, nämlich die Erfassung von großen Bandbreiten in einem Firmennetzwerk zunichte. Im Nachfolgenden wird die Aufnahme von Spektren genauer betrachtet und einige weitere Probleme dargestellt.

7.1.2.4 Berechnung von breitbandigen Signalspektren

Wie bereits festgestellt, muss auch beim USRP das Spektrum mit einer geringeren Abtastrate als maximal möglich erfasst werden. Trotz dieses Einwandes wird das Spektrum für verschiedene Signale nachfolgend betrachtet und ausgewertet. Die Abbildung 7.16 zeigt, dass das USRP ähnliche Probleme wie das RTL-SDR bei der Berechnung des Betragspektrums hat. Denn wie in Abbildung 7.17 genauer zu erkennen ist, fällt das Spektrum aufgrund der in Kapitel 4 besprochenen Theorie ebenso an den Rändern ab, wie beim RTL-SDR. Das führt wiederum dazu, dass das Spektrum wieder in kleinen Einzelstücken aufgenommen werden muss. Der Vorteil, der sich aber anhand dieser Abbildung erkennen lässt, ist, dass es kein LO-Leakage gibt. Aus diesem Grund kann immer eine Bandbreite von 1 MHz in einem Schritt aufgenommen werden. Der Anfang und das Ende eines jeden 2-MHz-Teilspektrums wird dabei um je 500 kHz abgeschnitten. Das heißt, in einem Sweep wird im Vergleich zum RTL-SDR die doppelte Bandbreite erfasst.

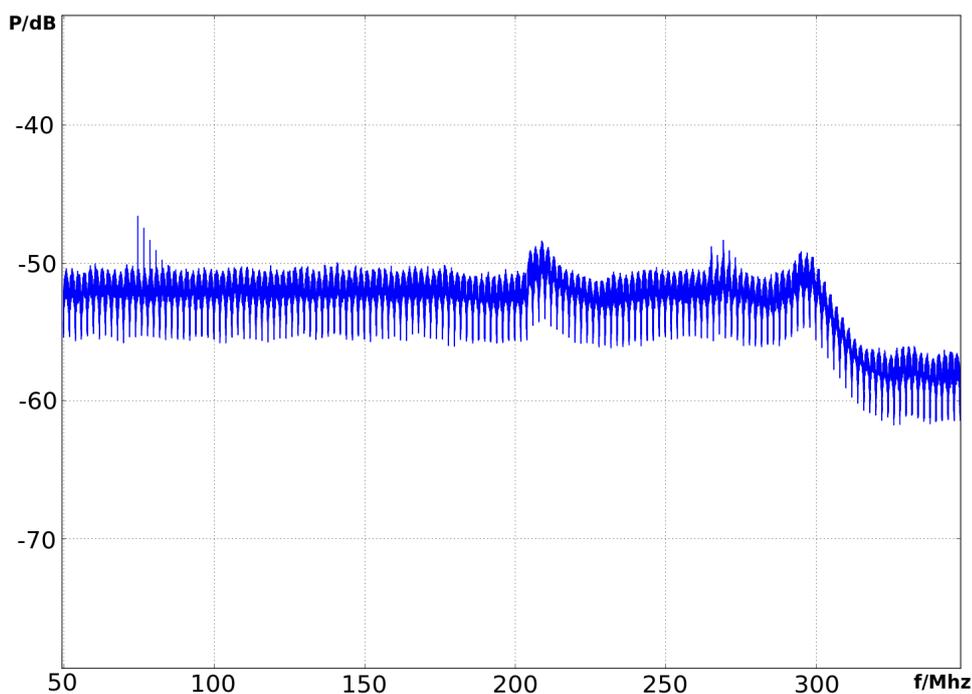


Abbildung 7.16: Breitbandige Spektromaufnahme von 50 bis 350 MHz an der Zwischenfrequenz beim Einsatz des 21 cm - 7 Beam Empfängers mit einem USRP-2920

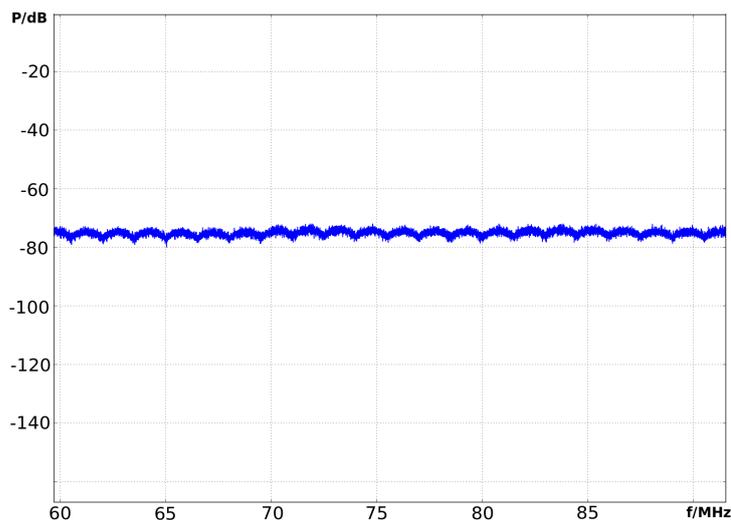


Abbildung 7.17: Detailansicht der Spektrumsaufnahme mit einem USRP-2920 (Bandbreite = 30 MHz)

Die Abbildung 7.18 zeigt eine detaillierte Spektrumsaufnahme mit einem abschnittsweise aufgenommenen Signalspektrum. Die abfallenden Flanken sind dadurch weggefallen. Durch die Aufnahme von 1 MHz pro Sweep hat sich die Programmlaufzeit von knapp 5,5 auf nur noch 1,5 Minuten gegenüber dem RTL-SDR verbessert. Die Totzeit zwischen jeder Spektrumsaufnahme konnte durch Experimente auf nur noch 0,2 Sekunden reduziert werden. Das liegt vor allem an der hohen Geschwindigkeit des verbauten FPGA und an der allgemein höherwertigen Hardware.

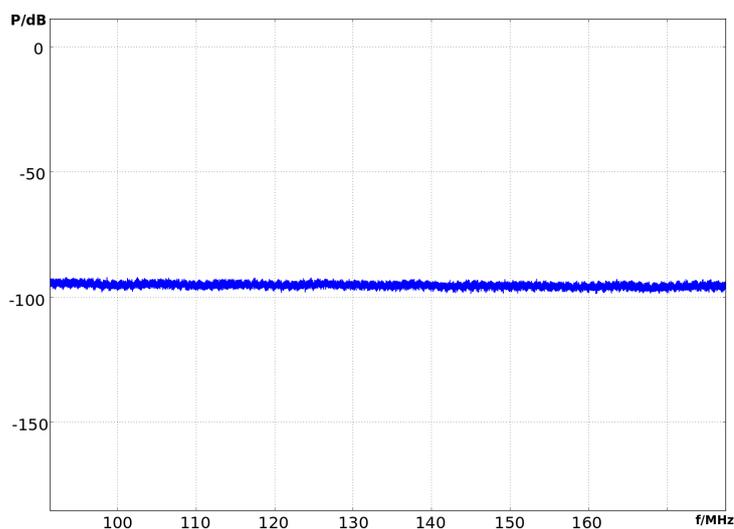


Abbildung 7.18: Detailaufnahme des abschnittsweise aufgenommenen Spektrums mit einem USRP-2920 (Bandbreite = 90 MHz)

7.2 Anbindung der SDR-Hardware an das Zwischenfrequenzsystem

7.2.1 Inbetriebnahme des breitbandigen RTL-SDR-Spektrumanalysators an der Zwischenfrequenz

Das Blockschaltbild in Abbildung 7.19 zeigt den Signalfluss vom Radioteleskop bis zum RTL-SDR. Die vom Radioteleskop empfangenen Signale, die zwischen 400 MHz und 86 GHz liegen, werden zu Beginn hochfrequent verstärkt. Die verstärkten Signale werden anschließend auf eine Zwischenfrequenz gemischt. Je nach Empfänger und Einsatzgebiet kann diese Zwischenfrequenz in verschiedenen Bändern liegen, zum Beispiel bei 150 MHz oder bei 750 MHz. Die Signalbandbreite beträgt zwischen 300 und 2500 MHz. Das Mischsignal wird anschließend verstärkt und entzerrt. Dieses Signal wird auf mehrere Räume und Rechnerschränke aufgeteilt. Auf den RX-Raum, den AK-Raum und den Faraday-Raum. Bei diesen Untersuchungen wird das RTL-SDR im RX-Raum über einen Leistungsteiler an die Zwischenfrequenz angeschlossen. Der nachfolgende Signalverlauf entspricht dem im Experiment zuvor dargestellten Aufbau. Über ein Dämpfungsglied wird das Zwischenfrequenzsignal auf einen Mischer gegeben. Dieser mischt die Signale in einen bestimmten Zwischenfrequenzbereich, auf welchen das RTL-SDR eingestellt ist. Über den Raspberry Pi werden die Signale per Ethernet an einen Rechner übertragen und ausgewertet.

Um eine Messung eines bekannten Signals durchzuführen, wurde der 21 cm - 7 Beam Primärfokus-Empfänger im Radioteleskop auf die Emissionsfrequenz der Milchstraße eingestellt. Anschließend wurde das Spektrum des Zwischenfrequenzsignals mit dem RTL-SDR ermittelt. Das Ergebnis ist in Abbildung 7.20 zu sehen. Bei einer Frequenz von knapp 160 MHz in Zwischenfrequenzlage, welches einer Sky-Frequenz von 1,420 GHz entspricht, ist das emittierte Radiosignal der Milchstraße zu erkennen. Als Sky-Frequenz wird dabei die Frequenz bezeichnet, die der echten Frequenz eines Signals am Himmel ohne Mischung in Zwischenfrequenzlage entspricht.

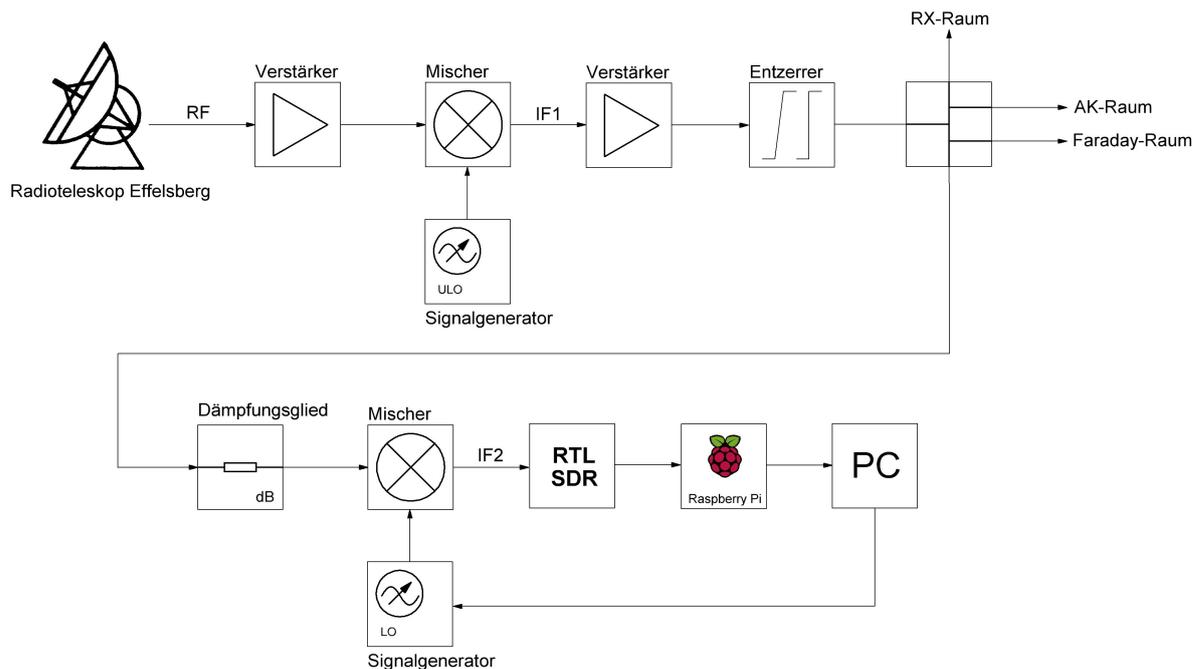


Abbildung 7.19: Blockschaltbild zur Aufnahme des Spektrums der ZF-Signale

Die Zwischenfrequenz des 21 cm - 7 Beam Empfängers liegt bei 150 MHz und hat eine Bandbreite von 300 MHz. Da dieser Empfänger während der Durchführung dieser Arbeit häufig im Einsatz war, beziehen sich viele der nachfolgenden Messungen auf diese Zwischenfrequenzlage und diesen Empfänger. Die Abbildung 7.21 zeigt eine Aufnahme über die komplette Bandbreite des Empfängers. Gut erkennbar sind der Bandpasscharakter des Spektrums und einige Störlinien bei verschiedenen Frequenzen. Die Abbildung 7.22 zeigt eine Referenzmessung des identischen Spektrums mit einem Rohde & Schwarz Spektrumanalysator. Die Spektren sind sich sehr ähnlich. Das spricht für die korrekte Funktion des RTL-SDR und der entwickelten Software. Zu beachten ist, dass der Spektrumanalysator auf Max-Hold stand. Dies wurde gewählt, um kurzzeitig auftretende Störlinie zu speichern. Daher sind in den Spektren leichte Unterschiede erkennbar.

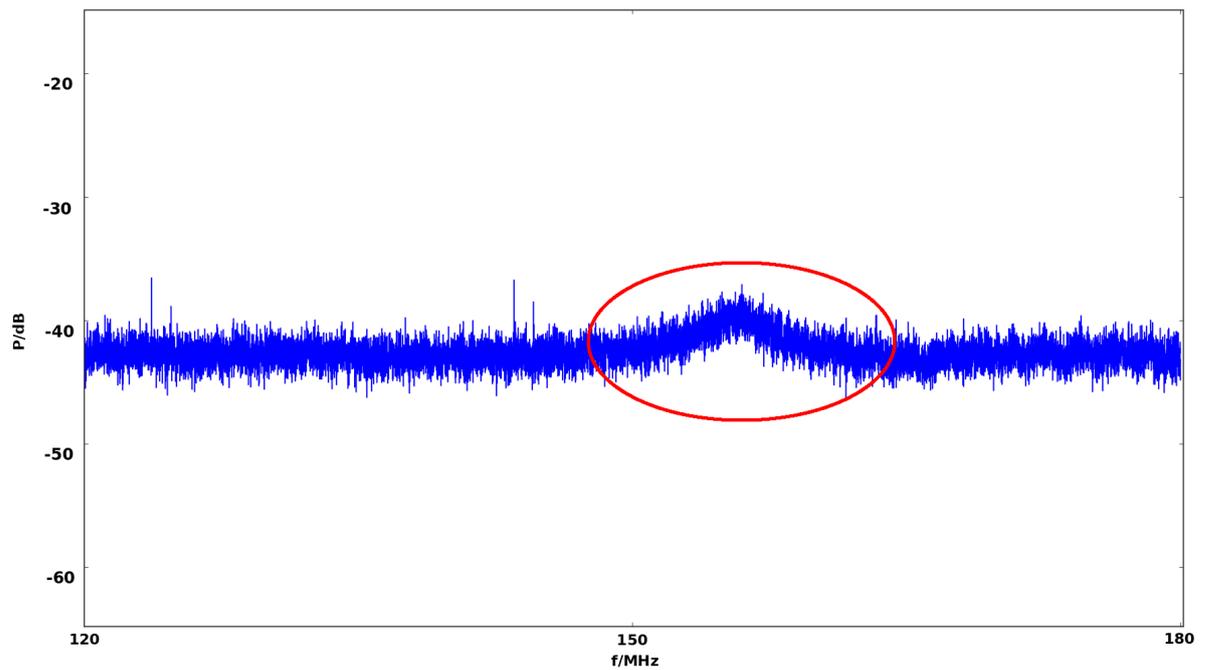


Abbildung 7.20: "First Light"-Aufnahme des Spektrums der Milchstraße (Bandbreite = 60 MHz)

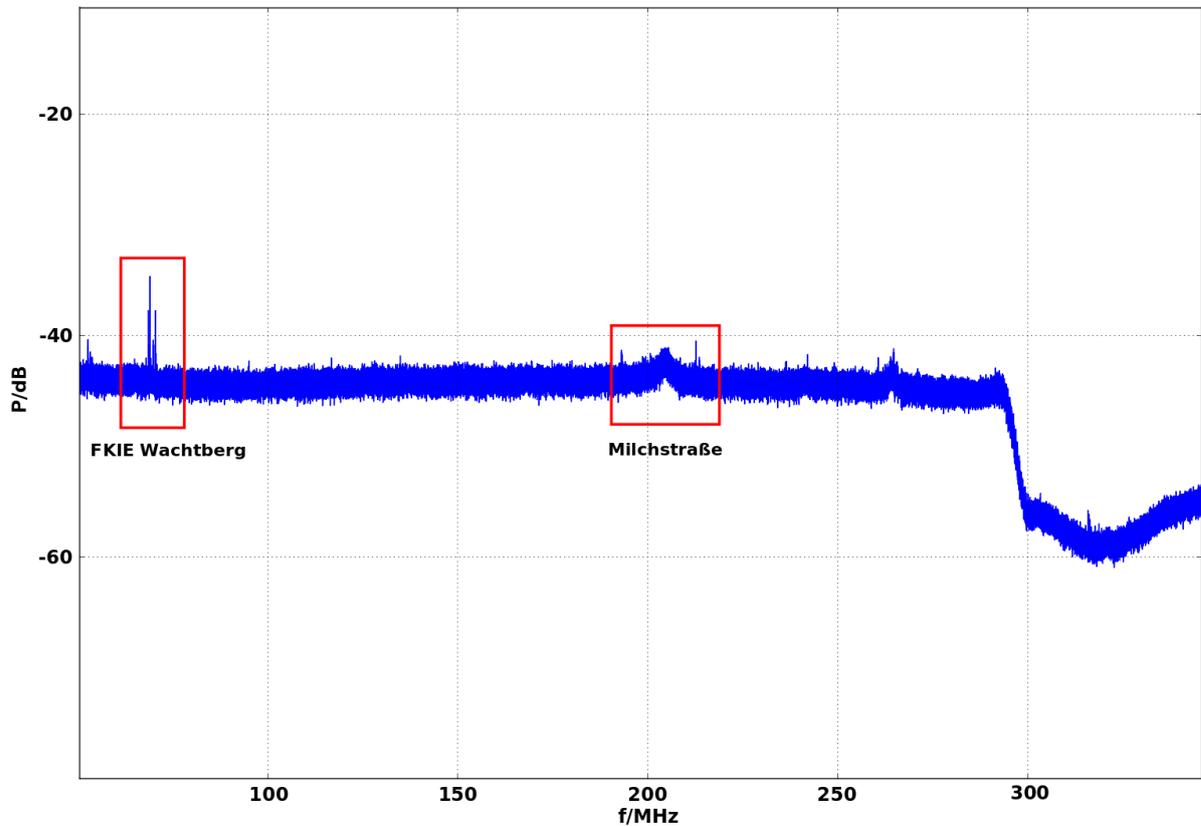


Abbildung 7.21: Aufnahme des Spektrums des bandpassgefilterten Nutzsignals zwischen 50 und 350 MHz

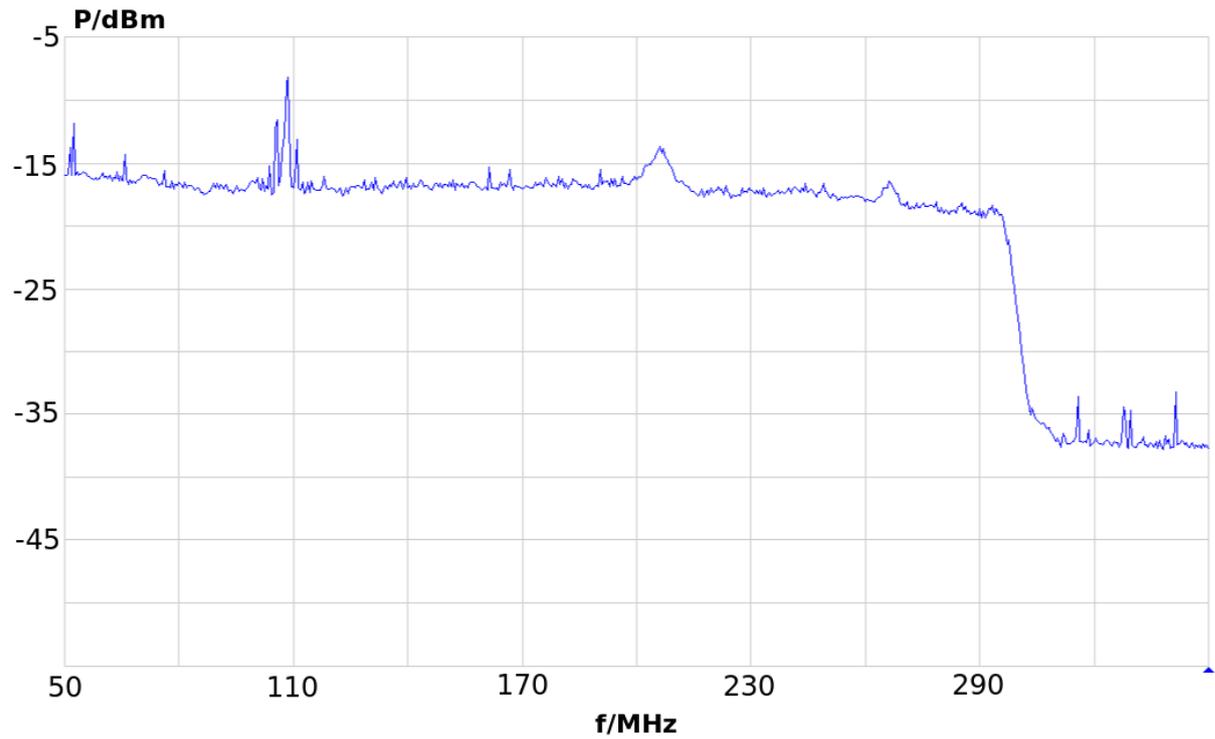


Abbildung 7.22: Vergleichsaufnahme des Spektrums des bandpassgefilterten Nutzsignals zwischen 50 und 350 MHz mit einem Rohde & Schwarz Spektrumanalysator

7.2.2 Anbindung des USRP an das Zwischenfrequenz-System

Der Anschluss des USRP an die Zwischenfrequenz erfolgt äquivalent zum Systemaufbau des RTL-SDR. Anstatt eines zusätzlichen Mixers und einer Frequenzdekade wird hier direkt das ZF-Signal auf den Eingang des USRP gegeben. Je nach Eingangspegel, der vom momentan aktiven Empfänger abhängt, wird eine zusätzliche Signaldämpfung vorgenommen. Über Ethernet wird das USRP mittels eines Switch direkt an das Netzwerk angeschlossen und so mit dem Rechner verbunden, der die digitale Signalverarbeitung betreibt.

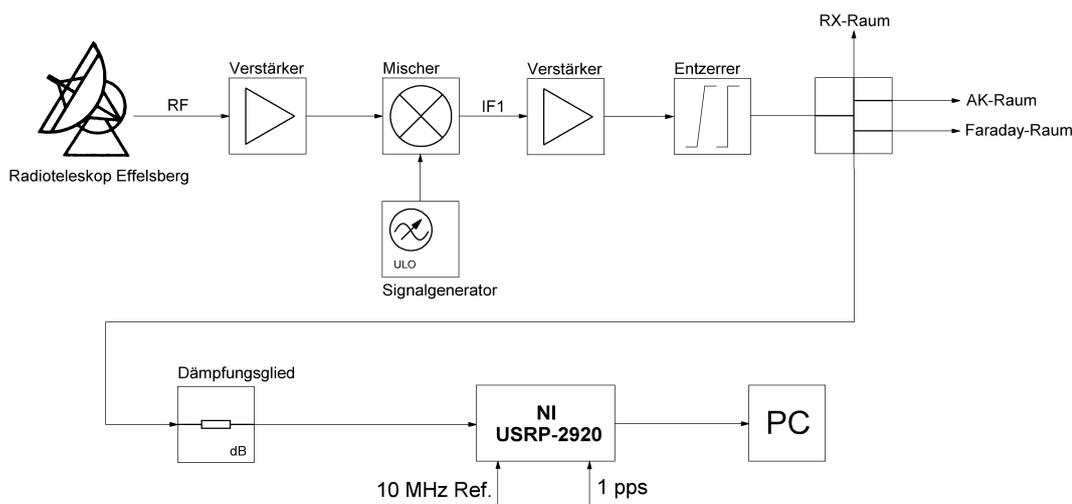


Abbildung 7.23: Aufbau des Systems zur Erfassung des ZF-Signalspektrums mit einem National Instruments USRP-2920

Besonders von Bedeutung bei diesem Aufbau ist die Einspeisung externer Referenzsignale zur Zeitsynchronisation. Wie in der Gegenüberstellung in Abschnitt 5 gezeigt wurde, beträgt die Frequenzabweichung bei der Einspeisung eines 10-MHz-Signals und eines Sekundensignals nur 2,5 ppm. Um auf die absolute Abweichung der Frequenz zu schließen, kann die Formel

$$\Delta f = \frac{f_0 \cdot \text{ppm}}{10^6}$$

verwendet werden. Die nachfolgende Tabelle 7.1 zeigt für verschiedene Frequenzen die entsprechende Frequenzgenauigkeit für eine Frequenzabweichung von 2,5 ppm.

f/MHz	$\Delta f/\text{Hz} @ 2,5 \text{ ppm}$
50	12,5
100	25
500	125
1000	250
2000	500

Tabelle 7.1: Absolute Frequenzabweichung für verschiedene Frequenzen eines USRP

Für die maximal einstellbare Frequenz ergibt sich eine Ungenauigkeit von gerade einmal 500 Hz. Das ist für die exakte Bestimmung von Störfrequenzen von großem Vorteil. Da die Referenzsignale von einem Wasserstoff-Maser erzeugt werden, der eine Frequenzstabilität im Femto-Bereich (10^{-15}) besitzt, ist die hohe Genauigkeit des USRP stets gegeben. Diese hohe Frequenzgenauigkeit erhöht daher die Zuverlässigkeit bei der Zusammensetzung von Teilspektren, da die Erhöhung der Mittenfrequenz in sehr exakten Abständen erfolgt.

7.3 Testmessungen an der Zwischenfrequenz

7.3.1 Erfassung von breitbandigen ZF-Spektren mit einem RTL-SDR

In diesem Abschnitt wird eine Messung eines Störsignals bei einer Frequenz von 1295 MHz genauer betrachtet. Bereits zuvor wurde im in Abbildung 7.21 dargestellten Signalspektrum ein Störer bei dieser Frequenz ermittelt. Mit Hilfe der in Matplotlib enthaltenen Zoom-Funktion für Plots wird dieses Spektrum detaillierter dargestellt (siehe Abbildung 7.24). Diese starken Störungen auf der Zwischenfrequenz sind diejenigen, die von diesem System erfasst und dokumentiert werden sollen. Als Referenzmessung ist in Abbildung 7.25 das gleiche Signalspektrum mit einem Rohde & Schwarz Spektrumanalysator dargestellt. Die Spektren sind nahezu identisch. Aufgrund der geringeren Frequenzauflösung des Rohde & Schwarz Spektrumanalysator und der Messungen mit der Max-Hold-Funktion sind wieder kleine Unterschiede erkennbar. Zudem wurden die Messungen nicht zum exakt identischen Zeitpunkt durchgeführt. Jedoch ist das Ergebnis insgesamt als gut zu bewerten und bestätigt die korrekte Funktion des SDR und der Software.

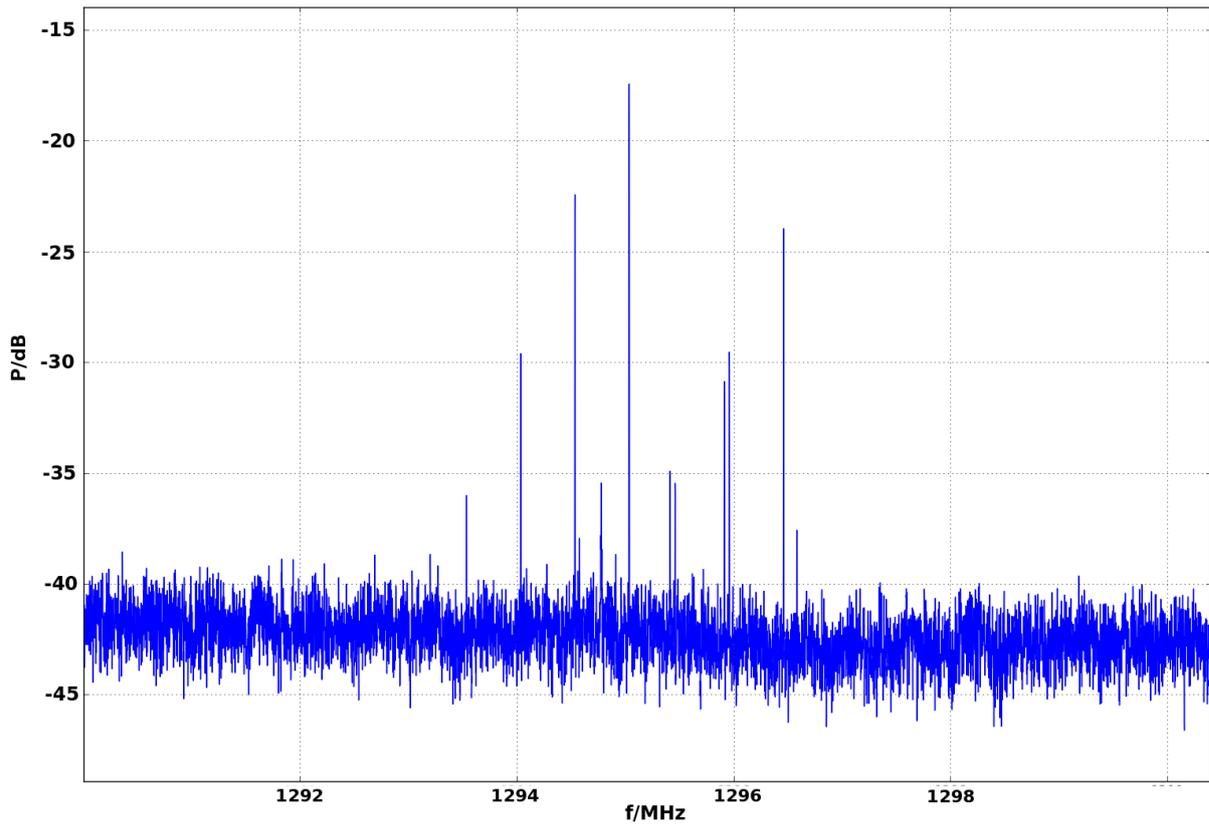


Abbildung 7.24: Spektromaufnahme von Störsignalen (Bandbreite = 10 MHz)

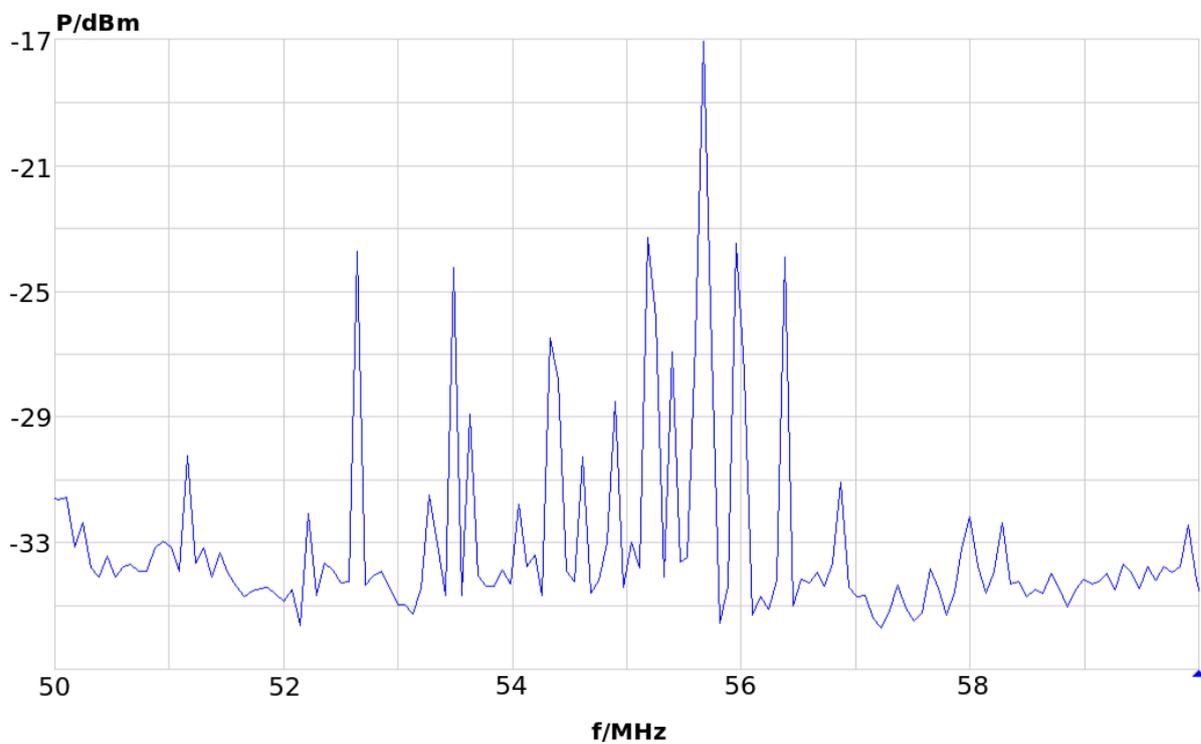


Abbildung 7.25: Vergleichsaufnahme des Spektrums von Störsignalen mit einem Rohde & Schwarz Spektrumanalysator

Die entwickelte Software zur Berechnung eines breitbandigen Spektrums mit einem RTL-SDR wurde für eine automatisierte Messung des Spektrums für einige Stunden eingesetzt. Dabei wurde das Signal des 21 cm - 7 Beam Empfängers auf Kanal B bei einer Sky-Frequenz von 1,367 GHz aufgenommen. Das 300 MHz breite Bandpasssignal wurde an der Zwischenfrequenz über 6 Stunden dokumentiert. In Abbildung 7.26 ist eine dreidimensionale Darstellung des Spektrums über die Zeit und die Frequenz zu sehen. Die einzelnen Messreihen des Spektrums wurden hintereinander zusammengefügt, um so einen dreidimensionalen Verlauf zu erhalten. Gut erkennbar ist, wie sich Störungen über die Zeit ändern. Speziell die Radarsignale bei 70 MHz in Zwischenfrequenzlage, was einer Sky-Frequenz von 1295 MHz entspricht, sind sehr dominant und ändern sich oft über die Zeit. Über die komplette Zeit ist jedoch zum Beispiel die konstante Resonanzfrequenz der im Empfänger verbauten Hohlleiter bei ungefähr 200 MHz in der Zwischenfrequenzlage zu erkennen. Um die Daten der FFT zu sichern wurden diese über eine einfache Python-Funktion in eine Textdatei geschrieben. Diese wurde anschließend in MATLAB eingelesen und der unten zu sehende Plot mittels der im Listing 7.4 dargestellten Befehle erstellt.

Listing 7.4: MATLAB-Skript zum dreidimensionalen Plotten von FFT-Daten

```
1 [frequency, power, time] = importdata('fft_file.txt');
2 figure;
3 surf(frequency, power, time);
4 xlabel('f/MHz \rightarrow');
5 ylabel('Input Power/dB');
6 zlabel('Time');
7 colormap default;
```

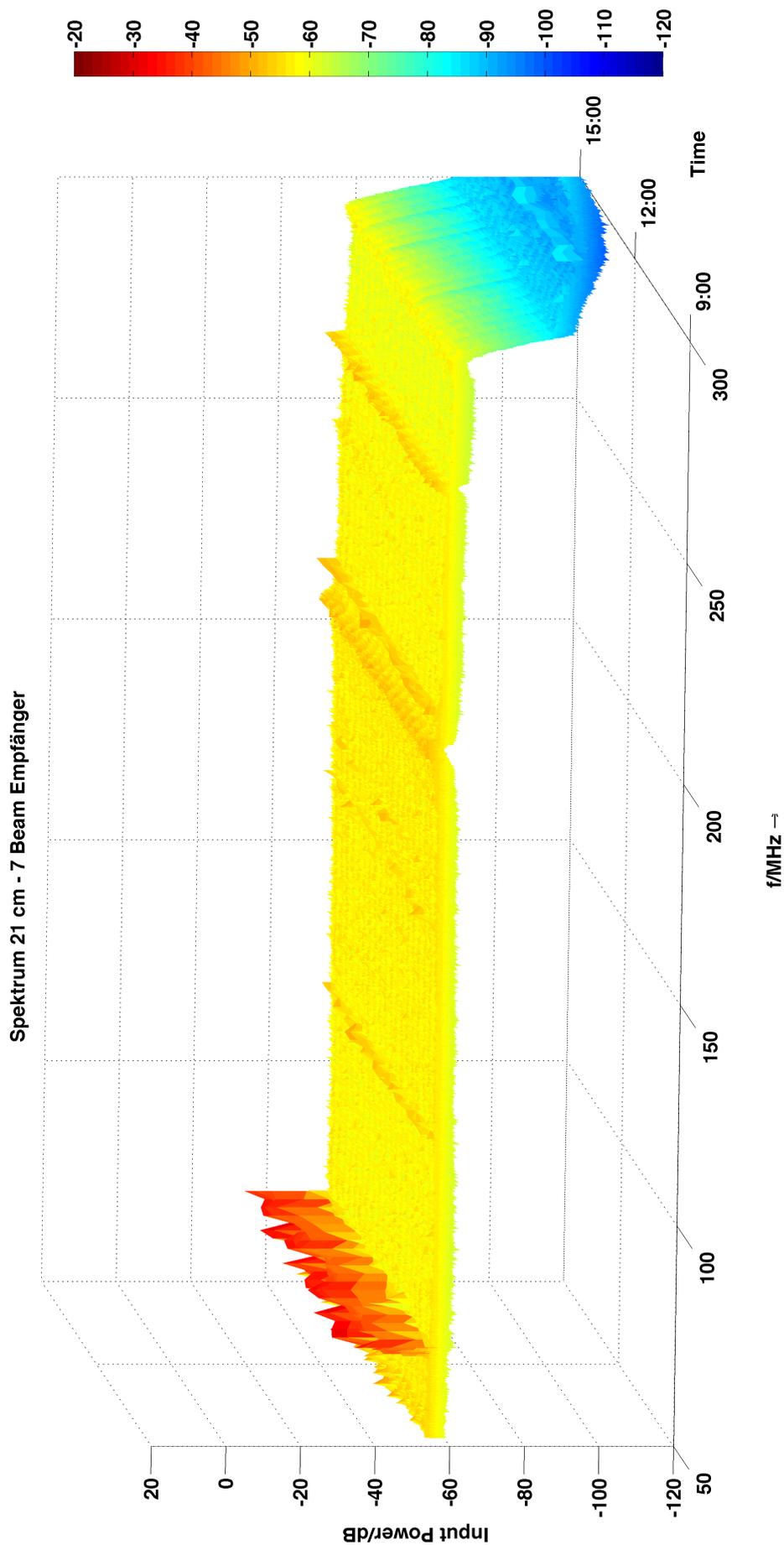


Abbildung 7.26: 3D-Wasserfalldiagramm über dauerhafte Messungen an der Schmalband-ZF zwischen 9 und 15 Uhr am 21 cm - 7 Beam Empfänger

Die Software zum Erstellen eines breitbandigen Spektrums erfasst in einem Sweep immer nur eine effektive Bandbreite von 500 kHz. Die Gründe dafür wurden bereits in den Abschnitten zuvor erläutert. Daraus folgt, dass für eine hohe Bandbreite entsprechend viele Teilspektren ermittelt werden müssen. Während der Digitalisierung des Eingangssignals entstehen viele Verzögerungen, zum Beispiel durch das Setzen des Empfangs-Synthesizers zwischen jedem Sweep und durch die Übertragung der Signale über Ethernet. Daher muss zwischen jedem Frequenzsweep eine Totzeit eingebaut werden, um eine korrekte Berechnung der FFT zu gewährleisten. Die Totzeit ist in diesem Experiment auf eine halbe Sekunde eingestellt. Dieser Wert wurde experimentell ermittelt, da so ein optimales Verhältnis aus zuverlässiger Berechnung und Laufzeit entsteht. Typischerweise müssen Bandbreiten von 300 MHz auf der Zwischenfrequenz erfasst werden. Für diese Bandbreite wurde nachfolgend die Laufzeit des Programms experimentell bestimmt. Dabei wurde nur die Zeit gemessen, die benötigt wird, um das Spektrum zu berechnen. Initialisierungen und das Plotten an sich wurden bei der Messung außen vor gelassen. Die Messreihe zur Laufzeit ist in der nachfolgenden Tabelle 7.2 dargestellt.

Messung	Zeit(sec)
1	324,85
2	326,38
3	329,51
4	325,76
5	325,91

Tabelle 7.2: Messung der Programmlaufzeit zur Erfassung einer Bandbreite von 300 MHz mit einem RTL-SDR

Die mittlere Laufzeit beträgt ungefähr 5 Minuten und 30 Sekunden. Daraus entstehen einige Probleme. Denn um sich schnell ändernde Störungen zu erfassen, ist es notwendig, dass das Spektrum schnell aktualisiert wird und eine möglichst große Bandbreite in möglichst wenigen Teilspektren erfasst wird. Da die Totzeit zwischen jedem Sweep nicht kleiner eingestellt werden kann, da sonst Fehler im Spektrum entstehen, kann die Laufzeit nicht weiter optimiert werden.

7.3.2 Umfassender Test des USRP am ZF-System

Im Rahmen eines Empfängertests mit einem neu eingebauten Bandpassfilter wurde das Betragsspektrum der Signale des 50 cm-Empfängers über mehrere Stunden aufgenommen. Das Filter dient zur Detektion der Signale der aktuell auf dem Mars befindlichen Rover. Das Filter hat einen Durchlassbereich zwischen 400 und 410 MHz. Die Signale des 50-cm-Empfängers werden aufgrund seines Frequenzbereichs nicht in eine Zwischenfrequenz umgesetzt, sondern direkt verarbeitet. Die Aufnahme des Signalspektrums für die Testreihe ist in Abbildung 7.27 in Form eines dreidimensionalen Wasserfalldiagramms dargestellt. Gut zu erkennen ist das Bandpassspektrum zwischen 400 und 410 MHz. Außerhalb des Bandpasses sind einige Störspektren zu erkennen. In diesem Frequenzbereich ist vor allem der digitale Funkstandard TETRA stark vertreten. Innerhalb des Bandpassspektrums sind einige Peaks zu erkennen, die in diesem Fall auf eine Detektion des Mars Rover schließen lassen oder auch als Folgen von RFI zu deuten sind. Im Kapitel A.1 des Anhangs sind Ergebnisse weiterer Testmessungen zu finden.

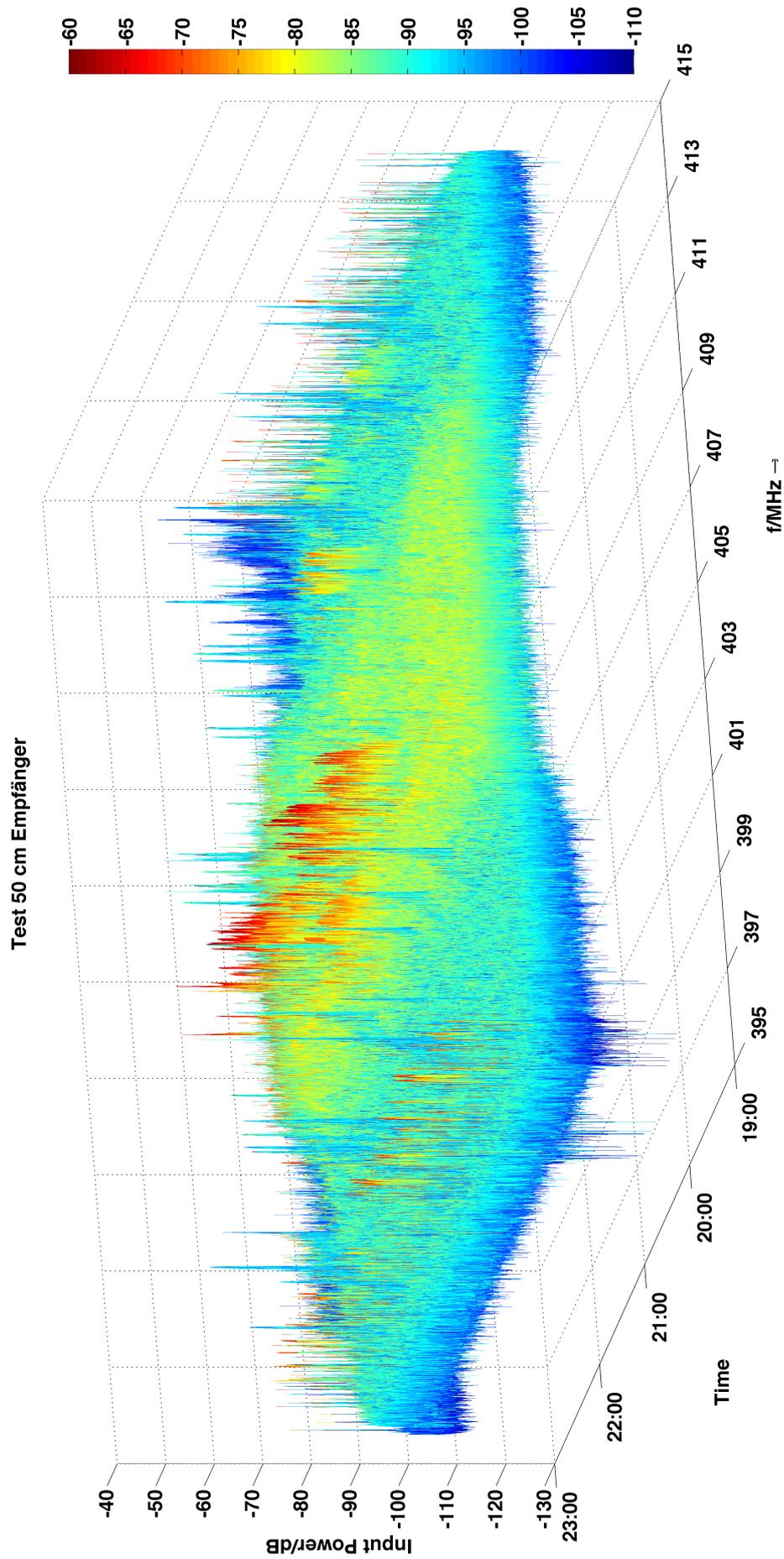


Abbildung 7.27: Spektrum der Signale des 50-cm-Empfängers zwischen 395 - 415 MHz und 19:00 - 23:00 Uhr

7.3.2.1 Methode zur Aufnahme von zwei Kanälen mit einem USRP

Da das USRP zwei Kanäle hat, ist es sinnvoll die Möglichkeit zu nutzen beide Kanäle zeitgleich aufzunehmen, da jeder Empfänger zwei Kanäle für zwei unterschiedliche Polarisationen besitzt. Da beide Kanäle des USRP nicht gleichzeitig ausgelesen werden können wird bei jedem Frequenzsweep der Kanal einmal umgeschaltet und anschließend wieder zurückgeschaltet. So werden die Signale beider Kanäle für jede eingestellte Mittenfrequenz des USRP ermittelt und gespeichert. Das Ergebnis wird in Form von zwei Subplots in der grafischen Oberfläche dargestellt. Der Programmflussplan zu dieser Methode ist in Abbildung 7.29 dargestellt.

Bei ersten Messungen fällt auf, wenn ein Eingang mit einem Signal belegt ist und der Zweite mit einem Abschlusswiderstand beschaltet ist, dass im Spektrum des zweiten Kanals Anteile der Signale des ersten Kanals enthalten sind. Das heißt, es kommt zu einem Übersprechen der Signale innerhalb des USRPs. Das ist ein Grund, warum die Nutzung von beiden Kanälen für diese Anwendung nicht geeignet ist. Da die voneinander unabhängigen Signale einen Einfluss aufeinander haben, ist eine zuverlässige RFI-Detektion nicht mehr möglich. Die Abbildung 7.28 zeigt ein Beispiel für die Signale des 6-cm-Sekundärfokus-Empfängers.

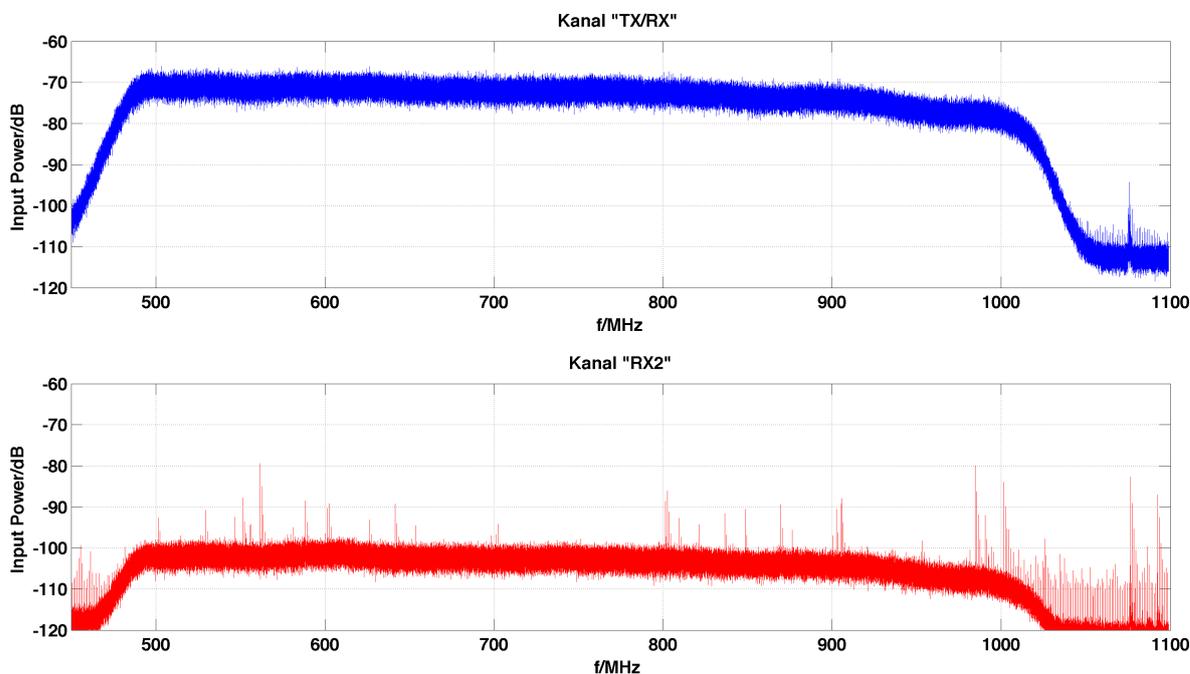


Abbildung 7.28: Übersprechen der Signale des 6-cm-Sekundärfokus-Empfängers von Kanal "TX/RX" auf den Kanal "RX2" (Bandbreite = 700 MHz)

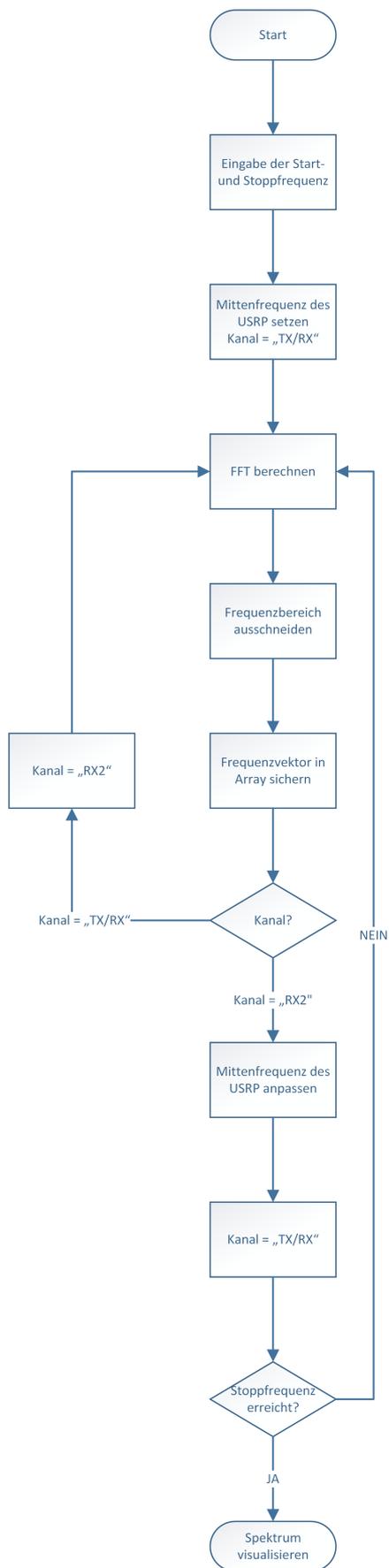


Abbildung 7.29: Programmflussplan zur Aufnahme beider Kanäle des USRP

Das nachfolgende Unterkapitel beschreibt Erweiterungen der zuvor entwickelten Software zur Erkennung von RFI mittels eines SDR. Dabei geht es vor allem um das automatische Zurückrechnen von einer Frequenz in der Zwischenfrequenzlage zur echten Sky-Frequenz. Aufgrund der zahlreichen Empfänger mit teils komplizierten Frequenzumsetzungen ist dies sehr umfangreich. Weiterhin wird eine Methode dargestellt, um Störungen in einem Signalspektrum zu erkennen und zu markieren.

7.4 Erweiterung des DFT-Spektrumanalysators zur Erkennung von RFI

7.4.1 Implementierung der Umrechnung zwischen Sky-Frequenz und Zwischenfrequenz

Um im weiteren Verlauf der Arbeit auf die echte Sky-Frequenz eines Störsignals zu schließen, muss diese über die aktuelle Zwischenfrequenz und die Einstellung der Frequenz der ULOs (Universal Local Oscillator) der Mischstufen berechnet werden. Diese Umrechnung wird mittels einer externen Python-Datei erledigt, sodass beim Hinzukommen von neuen Empfängern keine Änderungen im Hauptprogramm vorgenommen werden müssen. Zunächst wird anhand eines Beispiels die Methode hinter der Berechnung der aktuellen Sky-Frequenz untersucht.

Die hochfrequenten Empfangssignale des Radioteleskops werden bei der Übertragung in den Empfangsraum auf eine feste Zwischenfrequenz umgesetzt. Das hat vor allem Vorteile bei der Signalverarbeitung, da die empfangenen Signale Frequenzen bis zu 100 GHz enthalten können. Der durch die Abwärtsmischung entstehende Vorteil bei der Signalverarbeitung ist dabei von zentraler Rolle. Den Astronomen und Operateuren des Instituts steht dabei eine Übersicht zur Verfügung, um aus der aktuellen Sky-Frequenz die Frequenz der ULOs so zu berechnen, dass eben diese Sky-Frequenz auf die Mitte des Zwischenfrequenzbandes umgesetzt wird. Diese Berechnung kann umgekehrt werden und so von der Frequenz in Zwischenfrequenzlage auf die aktuelle Sky-Frequenz geschlossen werden. Dazu wird die aktuelle Frequenz der ULOs benötigt und der Empfängertyp sowie der aktuelle Messmodus. Da sich die Empfänger in der Berechnung aufgrund verschiedener Bandbreiten und Arbeitspunkte unterscheiden, ist das Wissen über den aktuellen Empfänger und den Modus für die korrekte Berechnung der Sky-Frequenz zwingend notwendig.

Anhand des 21 cm - 7 Beam Primärfokus-Empfängers wird diese Berechnung hier exemplarisch gezeigt. Um die ULO-Frequenz für eine bestimmte Sky-Frequenz so zu berechnen, dass diese in der Mitte der Zwischenfrequenz liegt, wird die Formel

$$f_{\text{ULO}} = \frac{f_{\text{Sky}} - 150 \text{ MHz}}{2} \quad (7.2)$$

benötigt. Diese kann nach der Sky-Frequenz umgestellt werden:

$$f_{\text{Sky}} = 2 \cdot f_{\text{ULO}} + 150 \text{ MHz}. \quad (7.3)$$

Die aktuelle ULO-Frequenz wird dabei über das Multicast extrahiert. Um für einen beliebigen Frequenzpunkt der Zwischenfrequenz die Sky-Frequenz zu berechnen, wird die konstante Frequenz von 150 MHz in der Formel 7.3 durch eine Variable ersetzt. Daraus folgt die Berechnungsformel:

$$f_{\text{Sky}} = 2 \cdot f_{\text{ULO}} + f_{\text{ZF}}. \quad (7.4)$$

Über diese einfache Formel ist so die Berechnung der Sky-Frequenz für jeden Punkt der Zwischenfrequenz möglich. Ist zum Beispiel bei 50 MHz in ZF-Lage eine Störung zu erkennen, folgt für die Sky-Frequenz bei einer ULO-Frequenz von 615 MHz:

$$\begin{aligned} f_{\text{Sky}} &= 2 \cdot f_{\text{ULO}} + f_{\text{ZF}} \\ &= 2 \cdot 615 \text{ MHz} + 50 \text{ MHz} = 1280 \text{ MHz}. \end{aligned} \quad (7.5)$$

Es gibt jedoch auch Empfänger die eine mehrfache Umsetzung der Empfangsfrequenz vornehmen. In diesen Fällen wird die Berechnung komplizierter, da mit zusätzlichen Mischfrequenzen umgegangen werden muss. Nachfolgend wird anhand des 6-cm-Sekundärfokus-Empfängers die Berechnung der Sky-Frequenz erläutert.

Für die zweifache Frequenzumsetzung der HF-Signale des 6-cm-Empfängers wird ein zweiter ULO benötigt. Dessen Frequenz wird ebenfalls über das Multicast ermittelt. Anhand der Empfängerbeschreibung lässt sich die erste Zwischenfrequenz mittels

$$f_{\text{ZF1}} = 10 \cdot f_{\text{ULO1}} - f_{\text{Sky}} \quad (7.6)$$

berechnen. Die zweite Zwischenfrequenz ergibt sich wiederum über

$$f_{\text{ZF2}} = 4 \cdot f_{\text{ULO2}} - f_{\text{ZF1}}. \quad (7.7)$$

Wird die Beziehung aus 7.6 in Formel 7.7 eingesetzt, ergibt sich die zweite ZF-Frequenz zu

$$f_{ZF2} = 4 \cdot f_{ULO2} - 10 \cdot f_{ULO1} + f_{sky}. \quad (7.8)$$

Wird die Formel 7.8 nach der Sky-Frequenz umgestellt, so folgt für diese

$$f_{sky} = f_{ZF2} - 4 \cdot f_{ULO2} + 10 \cdot f_{ULO1}. \quad (7.9)$$

Als Beispiel sei ein Störer bei einer Frequenz von 760 MHz in Zwischenfrequenzlage angenommen. Über die Formel 7.9 erhält man mit

$$f_{ULO1} = 746 \text{ MHz}$$

$$f_{ULO2} = 840 \text{ MHz}$$

eine Sky-Frequenz von

$$\begin{aligned} f_{sky} &= 760 \text{ MHz} - 4 \cdot 840 \text{ MHz} + 10 \cdot 746 \text{ MHz} \\ &= 760 \text{ MHz} - 3360 \text{ MHz} + 7460 \text{ MHz} \\ &= 4860 \text{ MHz}, \end{aligned} \quad (7.10)$$

welche exakt im Empfangsbereich des Empfängers liegt. Die Abbildung 7.30 zeigt das Blockschaltbild des 1,9 cm Empfängers zur Mischung der Eingangssignale in eine Zwischenfrequenzlage.

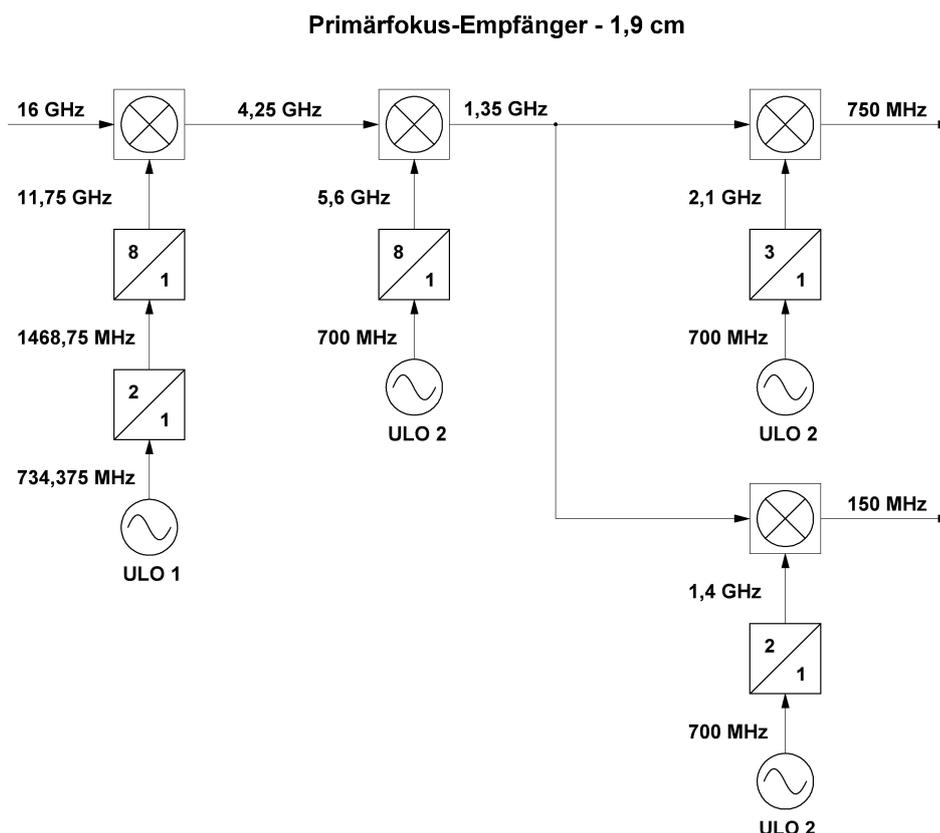


Abbildung 7.30: Blockschaltbild der ZF-Mischung des 1,9-cm-Primärfokus-Empfängers

Bei einigen Empfängern gibt es zudem verschiedene Messmodi. Je nach Messmodus werden die Signale auf eine andere ZF umgesetzt. Daher variiert die Berechnung der Sky-Frequenz für die Empfänger in Abhängigkeit des aktuellen Modus. Der aktuelle Empfänger wird zum Beispiel in der Form "20.3" über das Multicast gesendet. Das bedeutet, dass der 20 mm, beziehungsweise 2 cm Empfänger im Messmodus 3 betrieben wird. Mithilfe der Berechnungsübersicht wird so die korrekte Formel zur Berechnung ausgewählt.

In Python wird zur Berechnung der Sky-Frequenz eine externe Funktion definiert. Wie bereits oben erläutert ist für die korrekte Berechnung der Sky-Frequenz das Wissen über den aktiven Empfänger, den Messmodus und die Frequenzen der ULOs von Bedeutung. Daher werden diese Parameter der Funktion zur Berechnung übergeben. Zudem wird der Punkt der Zwischenfrequenz übergeben, für den die Sky-Frequenz berechnet werden soll. Als Rückgabewert wird die berechnete Sky-Frequenz geliefert. Um die korrekte Berechnungsformel auszuwählen, werden verschachtelte **if** Anweisungen verwendet. Als Beispiel ist ein Auszug aus der Funktion zur Berechnung der Sky-Frequenz für den 21 cm - 7 Beam Primärfokus-Empfänger und den 6 cm Sekundärfokus-Empfänger im nachfolgenden Listing dargestellt.

Listing 7.5: Funktion zur Umrechnung zwischen Zwischen- und Sky-Frequenz

```
1 def calc_freq(version, ulo0, ulo1, ulo2, freq, fokus):
2     """
3     @param: version(type=float) – Gibt den aktuellen ...
4             Empfaenger und den eingestellten Messmodus an
5             ulo0, ulo1, ulo2(type=float) – Gibt die ...
6             aktuelle Frequenz der ULOs an
7             freq(type=float) – Frequenzpunkt fuer ...
8             den die Sky-Frequenz berechnet ...
9             werden soll
10            fokus(type=string) – aktuell aktiver ...
11            Fokus
12
13    @return: sky_freq(type=float) – berechnete Sky-Frequenz
14    """
15    if(fokus=="Primaer"):
16        ...
17        # 21 cm – 7 Beam
18        if version == 217.1 or version == 217.2 or ...
19           version == 217.3 or version == 217.4:
20            f_sky = 2*ulo0 + freq
21        ...
22    if(fokus=="Sekundaer"):
23        ...
24        # 6 cm
25        if version == 60.1 or version == 60.2:
26            f_sky = freq + 5 * 2 * ulo0 - 4 * ulo2
27        ...
28
29    return f_sky
```

7.4.2 Integration eines Peak-Detektion-Algorithmus zur RFI-Erkennung

Um Peaks im Spektrum zu erkennen und diese in einem Plot zu markieren wird im folgenden Unterkapitel ein Algorithmus, der dies gewährleistet, in das Programm integriert. Diese Markierungen und die Ermittlung von Peaks, zusammen mit der Umrechnung auf die Sky-Frequenz dienen später dazu, auf die Quelle der Störung zurückzuschließen. Über eine Datenbank, die die Funkdienstverteilung enthält, wird so deutlich werden, welcher Dienst das Störsignal erzeugt.

Für die Implementierung einer solchen Funktion wird eine MATLAB-Routine zur Hilfe genommen, die eine Peak-Detektion in einem beliebigen Signal implementiert [5]. Diese MATLAB-Funktion wird in Python umprogrammiert und für die vorgesehene Anwendung angepasst.

7.4.2.1 Adaption der MATLAB-Funktion in Python

Der aus MATLAB adaptierte Algorithmus sucht nach lokalen Maxima in einem beliebigen Signal. Dazu werden die FFT-Daten benötigt, die in einer Liste gespeichert sind. Dieser Algorithmus sucht dabei nach lokalen Maxima in der Umgebung von Tälern im gegebenen Frequenzspektrum. Ein Wert wird als Peak gedeutet, wenn er einen gewissen Abstand zu den Werten in seiner Umgebung hat. Der Abstand zwischen dem zu suchenden Peak und den umgebenen Punkten wird in Form einer Variable *Delta* angegeben. So kann bestimmt werden, ab welchem Abstand ein Ausschlag in der Kurve als Peak gewertet wird. Das ist gerade bei dem starken Eigenrauschen der SDR-Hardware wichtig. Würde dieser Abstand nicht mit eingerechnet werden oder zu klein gewählt werden, würde es zu vielen Fehldetektionen kommen.

Zudem wird der Funktion ein Parameter übergeben, der angibt, in welchen Teilintervallen nach lokalen Maxima gesucht werden soll. So wird nicht global in der kompletten Datenliste nach Peaks gesucht, sondern immer nur für Ausschnitte und so lokale Maxima bestimmt. Das ermöglicht es, in beliebig kleinen oder großen Intervallen eine Peak-Detektion durchzuführen. Für diese Anwendung ist eine relative kleine Breite von Vorteil, da Störungen relativ dicht zusammenliegen können. Die aus MATLAB adaptierte Funktion ist im nachfolgenden Listing verkürzt dargestellt.

Listing 7.6: Funktion zur Ermittlung von lokalen Peaks in einem Array

```
1 def peakdetect(y_data, x_data, width, dlt):
2     max_peaks = [] # new empty List
3     length = len(y_data)
4     y_data = np.asarray(y_data) # convert FFT-List
5     max_value = -np.Inf # initialize max_value
6     for index, (x, y) in ...
7         enumerate(zip(x_data[:width], y_data[:width])):
8             # Find global Peaks
9             if y > max_value:
10                max_value = y
11                maxpos = x
12            # check if real maximum Peak
13            if y < max_value-dlt and max_value != np.Inf:
14                if y_data[index:index+width].max() < ...
15                    max_value:
16                    max_peaks.append((maxpos, max_value))
17                    max_value = np.Inf
18     return self.max_peaks
```

7.4.2.2 Test der Peak-Detektion anhand verschiedener Signalspektren

Anhand des ZF-Spektrums des 21 cm - 7 Beam Empfängers wird die zuvor implementierte Peak-Detektion getestet. In einer Aufnahme über die komplette Bandbreite des Nutzsignals wurde, wie in Abbildung 7.31 sichtbar, dieser Algorithmus angewandt. Einige Peaks wurden deutlich im Spektrum erkannt. Speziell bei der Sky-Frequenz von 1295 MHz sind viele Störungen zu erkennen, die auch korrekt detektiert wurden und mit einem roten Punkt markiert sind. Weiterhin wurden die Peaks im Spektrum die durch die Milchstraße und die Resonanzfrequenz der Hohlleiter im Spektrum entstehen detektiert. Daraus folgt ein weiteres Problem. Denn das Signal der Milchstraße ist per Definition nicht als RFI zu werten. Da das System aber nicht entscheiden kann, ob es sich um Man-Made-Noise oder um radioastronomische Strahlung handelt, wird dieses Signal als RFI gehandelt. Zwischen diesen Peaks wurden weitere Linien als Störungen detektiert, die teilweise im Rauschen liegen und somit als Fehldetektion gewertet werden können. Auffällig ist auch der Peak, der hinter der Flanke des Bandpasses liegt. Diese Abbildung zeigt, dass der Algorithmus funktioniert und sich bis auf wenige Ausnahmen für diese Anwendung gut eignet.

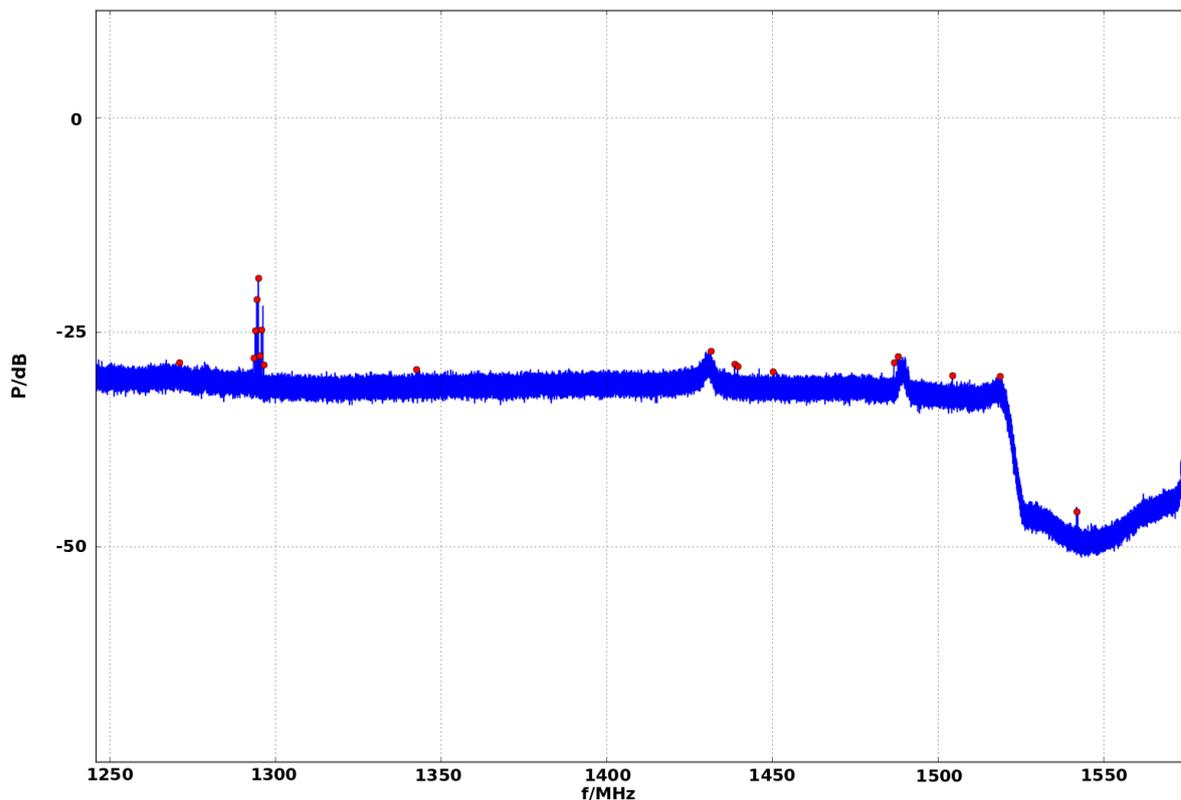


Abbildung 7.31: Detektion von Peaks über den kompletten Schmalband-ZF-Frequenzbereich (Aufnahme mit einem RTL-SDR; Bandbreite = 350 MHz)

Die Abbildung 7.32 zeigt eine detaillierte Darstellung der Radarstörsignale bei einer Sky-Frequenz von 1295 MHz. Diese werden durch die Radaranlage des Fraunhoferinstituts in Wachtberg erzeugt und werden häufig bei Messungen in diesem Frequenzbereich empfangen. Das Spektrum wurde über eine Bandbreite von 30 MHz aufgenommen. Hierbei ist gut zu erkennen, wie Störlinien, die sich nicht stark genug von den anderen Werten unterscheiden, nicht als Peaks detektiert werden. Die Hauptstörlinien werden korrekt mit einem roten Punkt an der Spitze markiert.

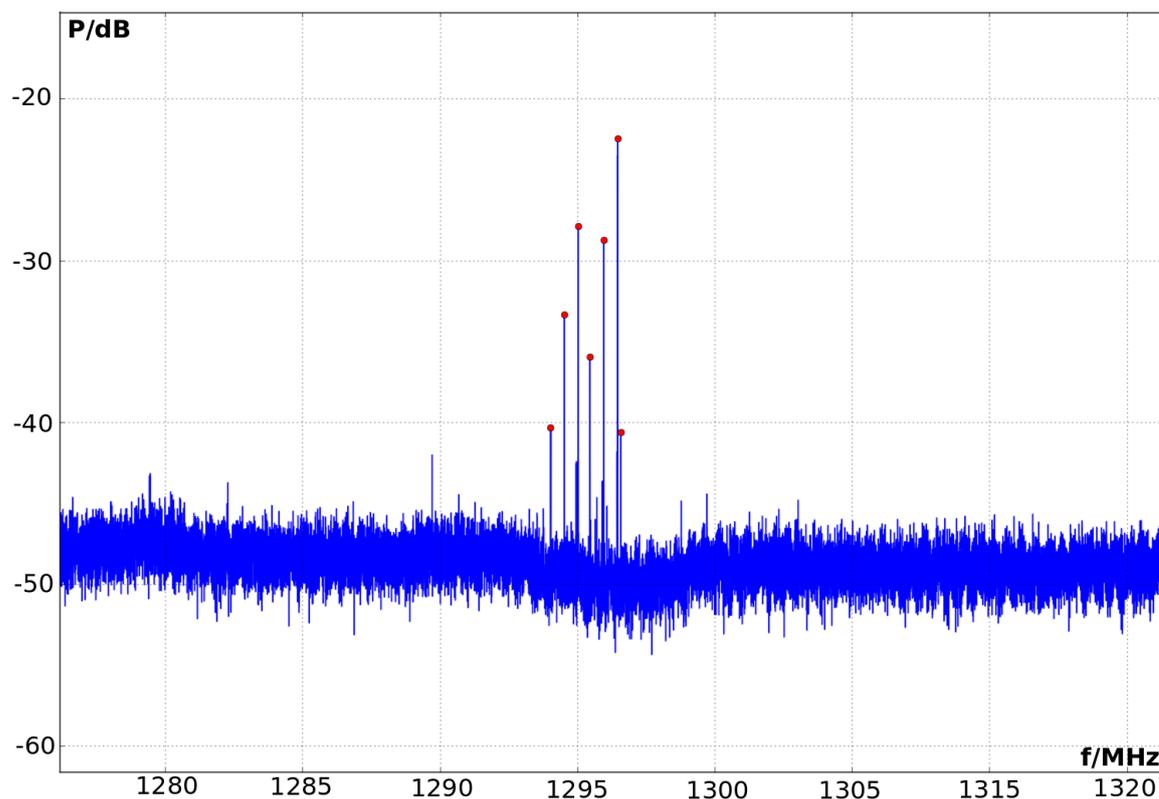


Abbildung 7.32: Detaillierte Ansicht der Peaks im Bereich eines Radarsenders

Die oben gezeigten Tests und weitere Messungen haben gezeigt, dass dieser Algorithmus zur Bestimmung von Peaks sehr gut funktioniert. Dabei ist aber auf eine korrekte Einstellung der Parameter *Delta* und *Width* zu achten. Die MATLAB-Funktion in Python zu konvertieren fiel dabei nicht schwer, da die Syntax teilweise ähnlich ist. Die Frage, die vorerst offen bleibt, ist, ob der detektierte Peak wirklich eine Funkstörung ist, die durch den Empfänger empfangen wurde oder durch beispielsweise Eigenstöranteile erzeugt wurde.

Im Zusammenspiel mit einer Datenbank wird die Funktion zur Ermittlung von lokalen Maxima Aufschluss über die potenzielle Quelle einer Störung geben. Diese Datenbank wird aus dem aktuellen Frequenznutzungsplan der Bundesnetzagentur erstellt und erfasst alle Frequenzen, die mit den Empfängern des Instituts empfangen werden können. Die Umsetzung dieses umfangreichen Plans in eine Datenbank wird im nachfolgenden Abschnitt betrachtet.

7.4.3 Bestimmung der vorgesehenen Frequenznutzung eines erfassten Störsignals

Die Bundesnetzagentur erstellt regelmäßig eine detaillierte Übersicht über die Verteilung der Frequenz in einem Bereich von 9 kHz bis 275 GHz. Dabei werden jedem einzelnen Frequenzbereich unter anderem eine Nutzungsbestimmung und eine Frequenznutzungsbedingung zugewiesen. Diese Zuweisung der Frequenzen ist für die Regulierung von Funkanwendungen elementar. Für diese Arbeit wird der Nutzungsplan dafür genutzt, um Störer im Spektrum einer Quelle, oder teilweise mehreren potentiellen Quellen, zuzuordnen. Dabei wird anhand der ermittelten Sky-Frequenz des Störsignals in einer Datenbank nachgesehen, welchem Funkdienst diese Frequenz zugeordnet ist [38].

7.4.3.1 Übertragung des Frequenznutzungsplans in eine MySQL-Datenbank

Der Frequenznutzungsplan ist sehr umfangreich. Daher wäre ein Eintragen der Daten per Hand sehr zeitaufwendig. Auf der Website des HF-Technik-Herstellers Aaronia ist der komplette Frequenznutzungsplan in Form einer HTML-Tabelle zu finden [1]. Über die Website *convertcsv.com* lassen sich solche HTML-Tabellen in CSV-Dateien konvertieren und speichern, welche sich wiederum mit Python handhaben lassen, wodurch das Füllen einer Datenbank sehr einfach wird und automatisiert abläuft. Auf der oben genannten Website zur Konvertierung von HTML-Tabellen wird der Link zur Frequenznutzungstabelle von Aaronia eingetragen. Nach einigen Einstellungen wandelt die Website die Tabelle so in eine herunterladbare CSV-Datei um.

Anschließend wird durch ein Python-Programm eine neue Tabelle in einer MySQL-Datenbank angelegt. Diese enthält unter anderem Einträge für:

- eine Eintragsnummer,
- die Startfrequenz und Stoppfrequenz des reservierten Frequenzbereichs,
- die Frequenzeinheit,
- den Funkdienst,
- die vorgesehene Nutzung (zivil oder militärisch),
- die vorgesehene Frequenznutzung und
- eine Frequenznutzungsbedingung.

In diese Tabelle, die Teil einer MySQL-Datenbank ist, wird anschließend der in der CSV-Datei gespeicherte Frequenznutzungsplan mit einem Python-Programm automatisiert geschrieben. Dazu wird in einer Schleife durch die Einträge in der CSV-Datei gegangen und diese zeilenweise direkt in die Datenbank geschrieben. Über die MySQL-Workbench, einem GUI-Tool für MySQL-Datenbanken, wird anschließend überprüft, ob die Eintragung der Daten korrekt funktioniert hat. In diesem Fall hat das Programm alle Einträge fehlerlos in die Datenbank geschrieben.

7.4.3.2 Verbindung zur Datenbank

Um bei der Detektion eines Störers Informationen über die Frequenznutzung aus der zuvor angelegten Datenbank zu erhalten, wird eine Funktion geschrieben, die es ermöglicht, für einen bestimmten Frequenzpunkt die entsprechenden Informationen aus der Datenbank abzurufen. In der Datenbank wurde je nach Frequenzbereich die Start- und Stoppfrequenz entweder mit der Einheit kHz, MHz oder GHz gespeichert. Um die korrekten Zahlenwerte zu ermitteln, wird der an die Funktion übergebene Frequenzpunkt, der stets in MHz angegeben ist, entsprechend seiner Größenanordnung an die passende Einheit angepasst. Danach wird über einen SQL-Befehl, in den der Frequenzpunkt als Obergrenze für die Startfrequenz und als Untergrenze für die Stoppfrequenz eingesetzt wird, an die Datenbank übertragen. Diese liefert Informationen über den Frequenzbereich zurück, in dem der Frequenzpunkt liegt.

Diese in Python implementierte Funktion wird in das Hauptprogramm eingebunden. Matplotlib bietet eine Funktion an, die es erlaubt, dass wenn ein im Plot eingetragener Punkt angeklickt wird, eine beliebige Funktion ausgeführt wird. Die Punkte, die die Peak-Detektion liefert, werden als Auslöser für dieses Klick-Ereignis verwendet. Wird auf einen der roten Punkte geklickt, wird die Funktion für die Bestimmung der Frequenznutzung ausgeführt und die zurückgelieferten Daten werden in einem Textfeld in dem QT-GUI angezeigt. Ein Bildausschnitt einer solchen Störeridentifizierung ist in Abbildung 7.33 dargestellt.

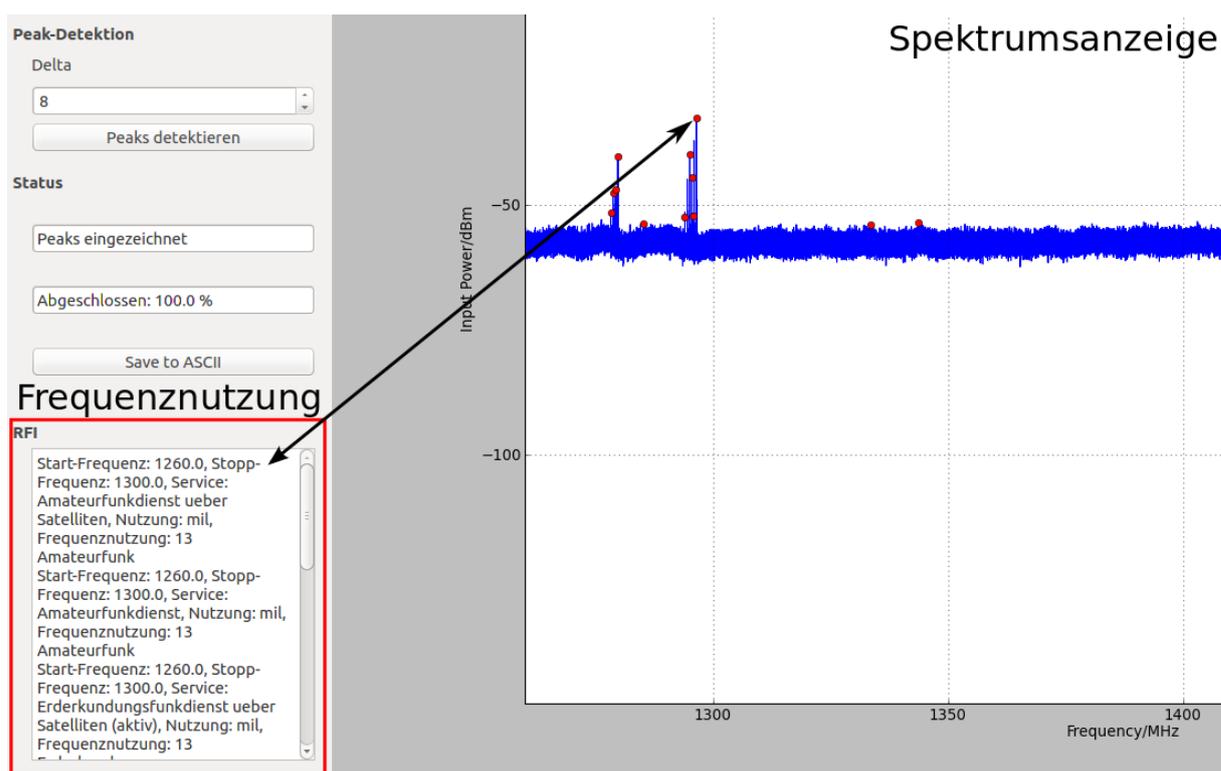


Abbildung 7.33: Bildausschnitt aus der Detektion eines Störers im ZF-Spektrum mit Informationen zur Frequenznutzung der Bundesnetzagentur

7.5 Diskussion der Ergebnisse der Softwareentwicklung

Die Experimente an der Zwischenfrequenz des Instituts haben aufschlussreiche Ergebnisse über die Verwendung eines RTL-SDR für diese Anwendung gegeben. Auf der einen Seite sprechen die in den Anfangskapiteln besprochenen hardwareseitigen Nachteile eines solchen SDR gegen die weitere Verwendung eines solchen Sticks. Auf der anderen Seite ist vor allem die lange Programmlaufzeit ein schwerwiegender Nachteil. Denn so wird es schwer, in einem breitbandigen spektralen Bereich sich schnell ändernde Störungen zu erfassen. Die Arbeit mit dem RTL-SDR hat aber großen Aufschluss mit dem Umgang mit einem SDR und der Signalverarbeitung in Python gegeben.

Dagegen bietet das USRP klare Vorteile bei der Dynamik und der Messgeschwindigkeit gegenüber einem RTL-SDR. Vor allem der doppelt so große Dynamikumfang und die sehr gute Intermodulationsfestigkeit bringen die entscheidenden Vorteile zur Überwachung von RFI. Durch die Erfassung einer höheren Bandbreite ist die Berechnung des Signalspektrums zudem deutlich schneller, auch wenn nicht in dem erwarteten Umfang. Es wurde jedoch auch klar, dass die Übertragung der IQ-Abtastdaten über das Netzwerk ein Element ist, das die erfassbare Bandbreite begrenzt. Daraus folgt, dass die Hardware direkt per Ethernet an den Signalverarbeitung betreibenden Rechner angeschlossen werden muss und keinen Umweg über einen Switch und das Firmennetzwerk haben darf.

Die Arbeit mit dem USRP ist im Ganzen aber als Erfolg zu werten, da die Möglichkeiten mit dieser Hardware weitreichender sind als mit einem RTL-SDR und deutlich bessere Ergebnisse erzielt werden konnten. Die untersuchten Eigenschaften und die Ergebnisse von Testmessungen machen deutlich, dass eine solche Hardware für den Einsatz als DFT-Spektrumanalysator prinzipiell geeignet sein kann. Doch vor allem die Eigenstörungen, die das USRP erzeugt und die im Spektrum erkennbar sind, sind für diese Anwendung ein großer Nachteil. Hinzu kommt das starke Übersprechen zwischen den beiden Kanälen. Die eindeutige Detektion von RFI wird so deutlich erschwert. Wiederum ist durch die hervorragende Softwareschnittstelle GNU Radio die Integration beider Interfaces in ein System und die Entwicklung von Software unkompliziert.

Insgesamt ist die verwendete Hardware aufgrund der in diesen Kapiteln dargestellten Ergebnisse für diese Anwendung als ungeeignet zu bewerten. Besonders die hohe Programmlaufzeit und Unsicherheiten bei der Zusammensetzung der Teilspektren führen zu unzuverlässigen Ergebnissen. Im Nachfolgenden wird daher der Einsatz eines FFTS zur Ermittlung des Signalspektrums zur spektralen RFI-Überwachung untersucht. Dieses FPGA-Board liefert das Betragsspektrum für eine Momentan-Bandbreite von 1,5 GHz direkt über Ethernet. Die Daten werden wie bisher in einem Python-Programm verarbeitet. Das Kapitel behandelt die Untersuchung der Hardware und den Vergleich zu der bisher verwendeten SDR-Hardware. Anschließend wird die Entwicklung der Software betrachtet, die sich in vielen Punkten von der bisherigen unterscheidet. Im Zusammenspiel mit einem RTL-SDR werden zudem durch das FFTS detektierte Störungen demoduliert. Die in diesem Kapitel entwickelte Software wird für die Erkennung von Störungen genutzt.

8 Verwendung eines FFT-Spektrometers zur RFI-Detektion

Dieses Kapitel befasst sich mit der spektralen Detektion von RFI mittels eines FFTS (Fast Fourier Transformation Spectrometer) im Zusammenspiel mit einem Software Defined Radio, welches zur Demodulation von Störsignalen genutzt wird. Einige Teile der zuvor entwickelten Software werden weiter genutzt, wie zum Beispiel die Umrechnung zwischen Sky- und Zwischenfrequenz und die Funktion zur Ermittlung der vorgesehenen Frequenznutzung durch die Bundesnetzagentur. Große Teile werden in diesem Kapitel jedoch verändert, beziehungsweise durch die Integration des SDRs und des FFTS in die Software zur spektralen RFI-Detektion umfassend erweitert.

Zu Beginn wird der Aufbau der Hardware diskutiert und auf die Funktionsweise eines FFTS eingegangen. Nachfolgend werden die Kommunikation und die Konfiguration dieses Gerätes über ein Python-Programm dargestellt. Die Anbindung des FFTS und des SDR in das ZF-System wird anschließend gezeigt. Für die Erfassung von zwei Kanälen mit einem FFTS wurde ein Umschalter gebaut, der mittels des Raspberry Pi gesteuert wird. Zur Bewertung der verwendeten Hardware wurden Testmessungen durchgeführt, die die Vorteile eines FFTS gegenüber einem USRP und einem RTL-SDR bei der Bestimmung des Spektrums zeigen. Am Ende dieses Kapitels wird ein alternativer Algorithmus zur Detektion von RFI in einem Spektrum diskutiert, der die zuvor implementierte Funktion zur Bestimmung von lokalen Maxima ergänzt. Zudem wird das entwickelte System auf insgesamt vier RTL-SDR Server erweitert.

8.1 Untersuchung des Hardwareaufbaus eines FFT-Spektrometers

Die Abbildung 8.1 zeigt ein Foto eines FFTS-Boards. Dieses ist der Hauptbestandteil eines FFTS. Gut zu erkennen ist das zentrale Xilinx FPGA. Dieses FPGA berechnet aus einem Eingangssignal das Betragsspektrum mit einer Bandbreite von 1,5 GHz über 8192 spektrale Kanäle. Über den 64-poligen Anschluss des Boards werden die Daten an die Backplane übertragen. Diese dient zur Kommunikation zwischen dem Rechner und dem FFTS-Board und überträgt die Daten mittels Ethernet an einen Rechner. Zudem verwaltet die Backplane die Ressourcenverteilung wenn mehrere FFTS-Boards in einem FFTS verwendet werden. Für diese Arbeit wurde ein FFTS des Typs AFFTS (Array Fast Fourier Transform Spectrometer) verwendet.



Abbildung 8.1: FFTS-Board mit Xilinx Virtex-4 FPGA und 8192 spektralen Kanälen [31]

Das Blockschaltbild in Abbildung 8.2 zeigt den schematischen Aufbau des FFTS. Der blaue Rahmen kennzeichnet die Bestandteile die direkt auf dem Xilinx FPGA realisiert sind. Grün eingrahmt sind die Teile des FFTS die zum Beispiel auf der Backplane oder dem FFTS-Board an sich implementiert sind.

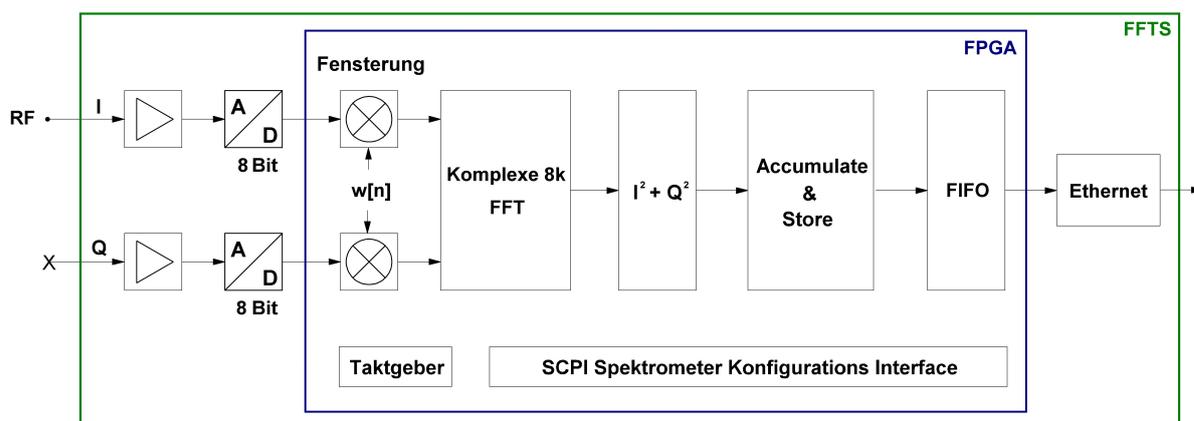


Abbildung 8.2: Blockschaltbild des FFTS mit 8192 spektralen Kanälen

Das RF-Signal, in diesem Fall das Signal der Zwischenfrequenz, wird über einen SMA-Anschluss in das FFTS eingespeist. Da nur reelle Bandpasssignale betrachtet werden, wird nur der I-Anschluss des FFTS verwendet. Nach einer Vorverstärkung der Signale werden diese mittels eines 8-Bit-A/D-Wandler digitalisiert. Auf dem zentralen FPGA werden die Signale gefenstert. Die Filterkoeffizienten der Fensterfunktion (standardgemäß ein Flat-Top-Fenster) können über eine externe Konfigurationsdatei eingestellt werden. Für diese Arbeit werden diese aber auf der Standardeinstellung belassen.

Nach der Fensterung der Signale wird eine komplexe FFT mittels einer Polyphase Filter Bank durchgeführt. Dabei wird eine FFT-Länge von $N = 8192$ für eine Momentan-Bandbreite von 1500 MHz verwendet. Mit 8192 Punkten über eine Bandbreite von 1,5 GHz wird daher eine spektrale Auflösung von 183 kHz pro diskretem Frequenzpunkt erreicht. Aus dem komplexen, diskreten Fourierspektrum wird dann das reine Betragsspektrum mit einer Auflösung von 72 Bit gebildet. Über einen Accumulate & Store-Block wird zudem eine Integration vorgenommen, um vor allem das Quantisierungsrauschen zu minimieren. Das Spektrometer integriert dabei Signale über eine maximale Zeitdauer von 5 Sekunden auf. Über einen FIFO-Speicher werden die ermittelten Daten an einen Ethernet-Controller übergeben, der die Daten an den Rechner sendet. Dabei wird zudem eine "Floating Point"-Konversion vorgenommen. Diese wandelt die 72-Bit-Integer-Zahlen des Betragsspektrums in 32-Bit-Float-Zahlen um. Dadurch wird die Fehlerfortpflanzung bei den numerischen Berechnungen des Spektrums verringert.

Das FPGA kann zudem mit einem IRIG-B-Signal gespeist werden, um aktuelle Zeitinformationen zu erhalten. Über ein spezielles SCPI-Interface wird das Spektrometer über UDP konfiguriert. Die Konfiguration wird im nachfolgenden Abschnitt näher betrachtet. Einige der wichtigsten Spezifikationen sind in Tabelle 8.1 dargestellt.

A/D-Wandler Abtastrate	3 GS/s
Eingangsleistung (Vollaussteuerung)	0 dBm
SFDR	> 43 dBc @ 1498 MHz
Spektrale Kanäle	8192
Dump-Zeit	20 ms @ 8192 Kanäle

Tabelle 8.1: Spezifikationen des verwendeten AFFT-S [25, S. 5]

8.2 Allgemeine Aspekte zur FFTS-Softwareentwicklung

Die Software zur Arbeit mit einem FFTS kann in mehrere Bereiche unterteilt werden (siehe Abbildung 8.3). Auf der einen Seite steht die Konfiguration und Steuerung des FFTS mittels eines UDP-Clients über ein SCPI-Interface. Zur Konfiguration wird dabei unter anderem auf mehrere Konfigurationsdateien zurückgegriffen. Diese enthalten Informationen zur Einstellung des FPGA oder zur Bestimmung der Filterkoeffizienten zur Fensterung der Signale. Auf der anderen Seite steht der Empfang der spektralen FFTS-Daten über TCP. Da diese Daten im Binär-Format übertragen werden, müssen diese zur weiteren Verarbeitung decodiert werden. Diese decodierten Daten stehen der Software dann zur digitalen Signalverarbeitung zur Verfügung. Die Konfiguration wird im nächsten Abschnitt anhand einer beispielhaften Konfiguration gezeigt.

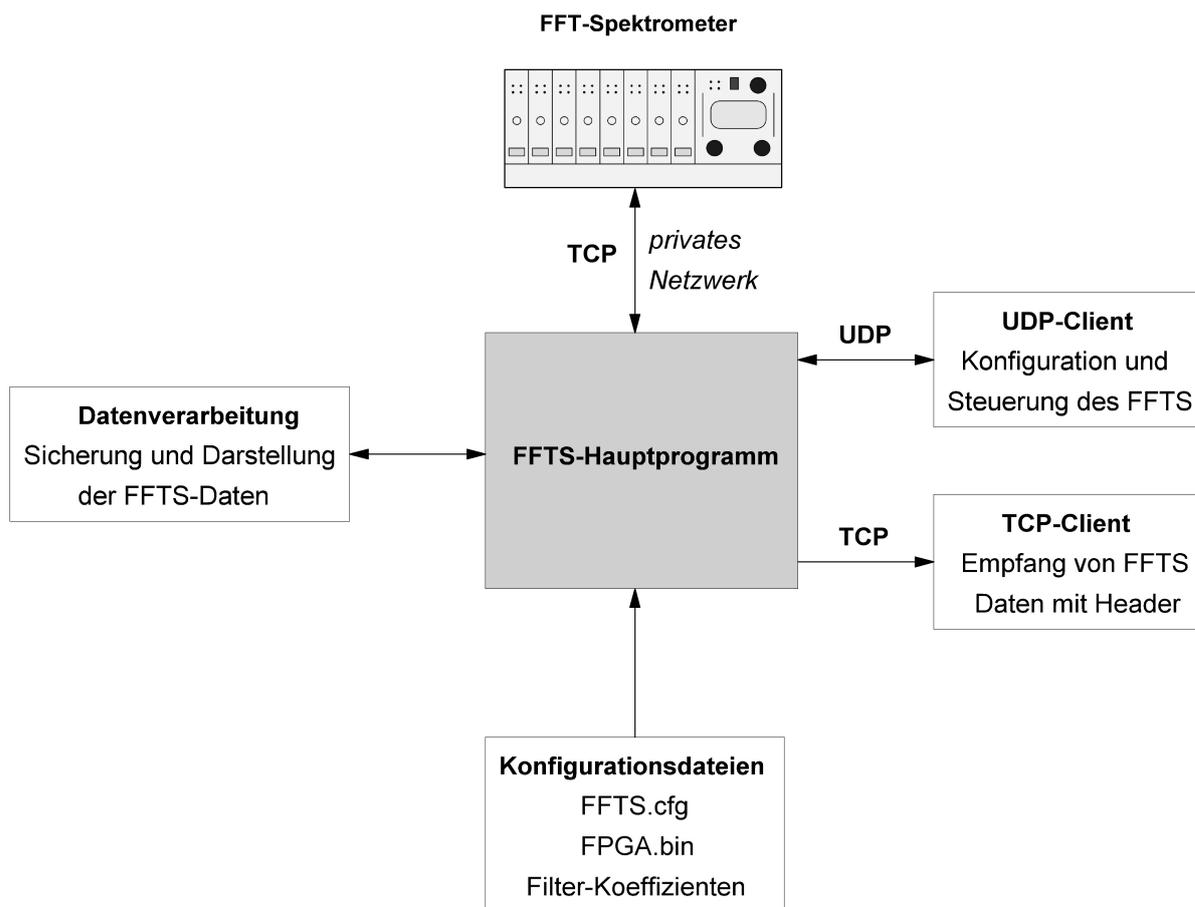


Abbildung 8.3: Prinzipieller Aufbau der Software zur Konfiguration des FFTS und zum Empfang der spektralen Daten über TCP [12, S. 15]

8.2.1 Konfiguration des FFTS über UDP und SCPI

Die Konfiguration des FFTS erfolgt über das UDP-Protokoll. Um die Konfiguration testweise durchzuführen, werden mittels eines einfachen UDP-Clients Befehle über das Terminal an das FFTS geschickt [12, S. 17]. Eine Übersicht der wichtigsten Befehle ist in Tabelle 8.2 zu finden. Wurde ein Befehl korrekt eingegeben und vom FFTS angenommen, wird eine Bestätigung des Befehls zurückgegeben. Im Fehlerfall wird eine Fehlerwarnung ausgegeben.

SCPI Befehl	Beschreibung
AFFTS:START	Startet eine Messung
AFFTS:STOP	Stoppt eine Messung nach der Beendigung der aktuellen Phase
AFFTS:ABORT	Bricht eine Messung ab
AFFTS:CONFIGURE	Schreibt die aktuelle Konfiguration in das Gerät
AFFTS:STATE	Liefert den aktuellen Status des FFTS
AFFTS:CMDBLANKTIME	Stellt die Totzeit zwischen jeder Messung ein
AFFTS:CMDUSEDSECTIONS	Auswahl des FFTS-Boards das zum Datenstream verwendet werden soll
AFFTS:CALADC	Kalibriert die A/D-Wandlung

Tabelle 8.2: Übersicht über die wichtigsten Befehle zur Konfiguration eines FFTS

Besonders wichtig sind die Befehle zum Starten und Stoppen von Messungen. Der "CONFIGURE"-Befehl dient zum Übernehmen der zuvor gesetzten Konfiguration. Weiterhin ist der Befehl "CALADC" von besonderer Bedeutung. Dieser Befehl kalibriert die A/D-Wandler, sodass diese optimal durch das Eingangssignal angesteuert werden. Das unten stehende Listing 8.7 zeigt einen Ausschnitt aus der Konfiguration des FFTS. Dabei wird zum Beispiel die eingestellte Bandbreite abgefragt sowie der Start einer Messung initialisiert.

Listing 8.7: Beispiel zur Konfiguration eines AFFTS mittels eines UDP-Telnet Client

```

1 AFFTS:BAND1:BANDWIDTH?
2 -> 1500.000
3 AFFTS:setconfigfile?
4 -> FFTS.cfg
5 FFTS.cfg
6 ('MPI:AFFTS:STOP 2014-09-23T08:36:14.260+0000',
7  ('134.104.79.80', 16210))
8 ('MPI:AFFTS:BAND1:CMDNUMSPECCHAN 8192
9  2014-09-23T08:36:16.019+0000',
10 ('134.104.79.80', 16210))
11 ('MPI:AFFTS:CMDUSEDSECTIONS 1 2014-09-23T08:36:16.120+0000',
12 ('134.104.79.80', 16210))
13 ('MPI:AFFTS:CALADC 2014-09-23T08:36:16.171+0000',
14 ('134.104.79.80', 16210))
15 ('MPI:AFFTS:CONFIGURE 2014-09-23T08:36:18.927+0000',
16 ('134.104.79.80', 16210))
17 ('MPI:AFFTS:START 2014-09-23T08:36:18.977+0000',
18 ('134.104.79.80', 16210))

```

8.2.2 Empfang der spektralen Daten per TCP

Die Daten des FFTS werden über TCP vom FFTS an den Rechner gestreamt. Um die Daten in Python empfangen zu können, wird ein TCP-Socket erstellt. Über diesen Socket horcht der Rechner auf der angegebenen IP-Adresse und schreibt die empfangenen Daten in ein Objekt. Da jeder einzelne Datenblock mit einem Header gesendet wird, ist es möglich, die Daten jeder einzelnen Messung exakt voneinander zu trennen.

Wie bereits zuvor erwähnt, werden die Daten im Binär-Format übermittelt. Über eine Funktion zur Dekodierung, die aus einer bereits vorhandenen Software des Instituts extrahiert wurde, werden diese spektralen Daten als Gleitkommazahlen in ein Numpy-Array geschrieben. Dieser Array-Typ ist Teil der Numpy-Bibliothek, welche umfangreiche Methoden und Funktionen zur wissenschaftlichen Arbeit mit Python bietet. Das Handling mit diesen speziellen Arrays ist ähnlich zum Listen-Datentyp von Python. Die Daten werden anschließend aus diesem Array heraus mittels Matplotlib geplottet und über ein QT-GUI dargestellt. Die genaue Vorgehensweise zur Entwicklung der Software wird im nächsten Abschnitt betrachtet.

8.3 Entwicklung eines Systems zur RFI-Detektion

8.3.1 Aufbau eines Kanalumschalters zur Erfassung von zwei Kanälen

Da die Empfänger im Institut zwei unterschiedlich polarisierte Wellen empfangen, werden diese Signale über zwei getrennte Kanäle übertragen. Da nur ein FFTS-Board und ein SDR zur Verfügung stehen, aber zwei Kanäle erfasst werden sollen, wurde ein Umschalter gebaut. Dieser wird über die GPIO-Pins des Raspberry Pi, der gleichzeitig auch als RTL-SDR Server dient, gesteuert und wird vor das FFTS, beziehungsweise die SDR-Hardware, gesetzt. Dieser Umschalter schaltet dabei die zwei Eingangssignale auf je einen Ausgang um. Das Eingangssignal des FFTS wird zudem um 22 dB verstärkt, während das Eingangssignal des SDR um 60 dB gedämpft wird, um die maximale Dynamik in beiden Hardwarekomponenten zu erreichen. Als Befehle werden einfache Steuersymbole über TCP gesendet, um die jeweiligen Relais per GPIO zu schalten. Die Relais sind dabei keine mechanischen Schalter, sondern arbeiten mit Pin-Dioden. Da die Umschaltung zwischen beiden Kanälen sehr schnell geschieht, wäre ein mechanisches Relais zu langsam. Die Abbildung 8.4 zeigt ein Blockschaltbild des Aufbaus des Umschalters. Die ermittelten Streuparameter des Umschalters sind im Anhang A.2 zu finden. Der Raspberry Pi wurde aufgrund der in Kapitel 6 ermittelten Störstrahlung in ein abgeschirmtes Gehäuse gebaut, um die Nutzsignale nicht negativ zu beeinflussen.

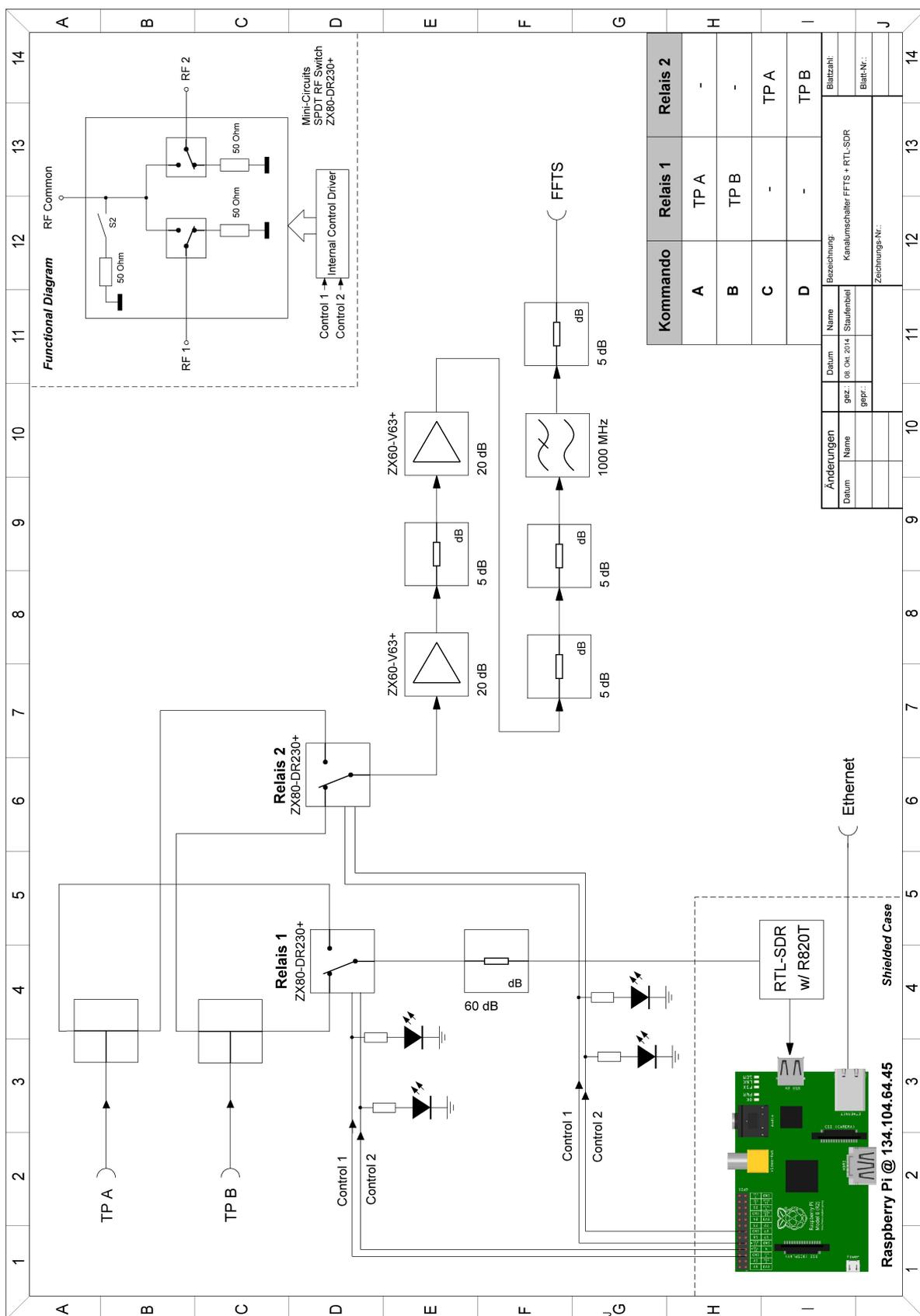


Abbildung 8.4: Blockschaltbild des Umschalters zur Erfassung von zwei Kanälen mit einem FFTS und einem SDR

Auf dem Raspberry Pi läuft ein Socket-Server, der in Python implementiert ist. Dieser horcht auf seiner IP-Adresse und dem Port 8888 und empfängt von einem Client gesendete Symbole. Je nach Symbol wird die Steuerspannung (3,3 V) für das entsprechende Relais geschaltet und so die Signale eines Kanals auf den Eingang des SDR und des FFTS gegeben. Für beide Hardwarekomponenten erfolgt die Umschaltung unabhängig voneinander. Der Client stellt ebenfalls über einen Socket eine Verbindung zum Server her und sendet entsprechend über diese Verbindung die festgelegten Steuerbefehle.

8.3.2 Programmentwicklung zur Aufnahme des Spektrums von zwei Kanälen

Zu Beginn des Programms (siehe Abbildung 8.5) zur Erfassung des Spektrums von zwei Kanälen wird das FFTS wie zuvor gezeigt konfiguriert. Wird eine Messung gestartet, wird im Probe-Zustand das Spektrum des ersten Kanals erfasst. Die Daten werden im Zustand "Decode" decodiert und anschließend wird das logarithmische Betragsspektrum berechnet.

Wenn der erste Kanal erfasst wurde, wird mittels des Umschalters das Signal des Kanals B auf den Eingang des FFTS geschaltet. Wieder wird das Spektrum erfasst, die Daten decodiert und das logarithmische Betragsspektrum berechnet. Wurde der zweite Kanal erfasst, werden die Daten in je einem Subplot dargestellt. Anschließend wird ein Algorithmus zur RFI-Detektion durchgeführt und die Daten gesichert. Der Umschalter wird auf Kanal A zurückgeschaltet und es wird auf den Beginn einer neuen Messung gewartet. Der große Vorteil dieser Software liegt darin, dass die Laufzeit zur Erfassung des Spektrums beider Kanäle und zum Plotten der Daten um die 10 Sekunden beträgt. So können schnell viele Spektren hintereinander berechnet werden und schnelle Änderungen im Signalspektrum auf beiden Kanälen erfasst werden.

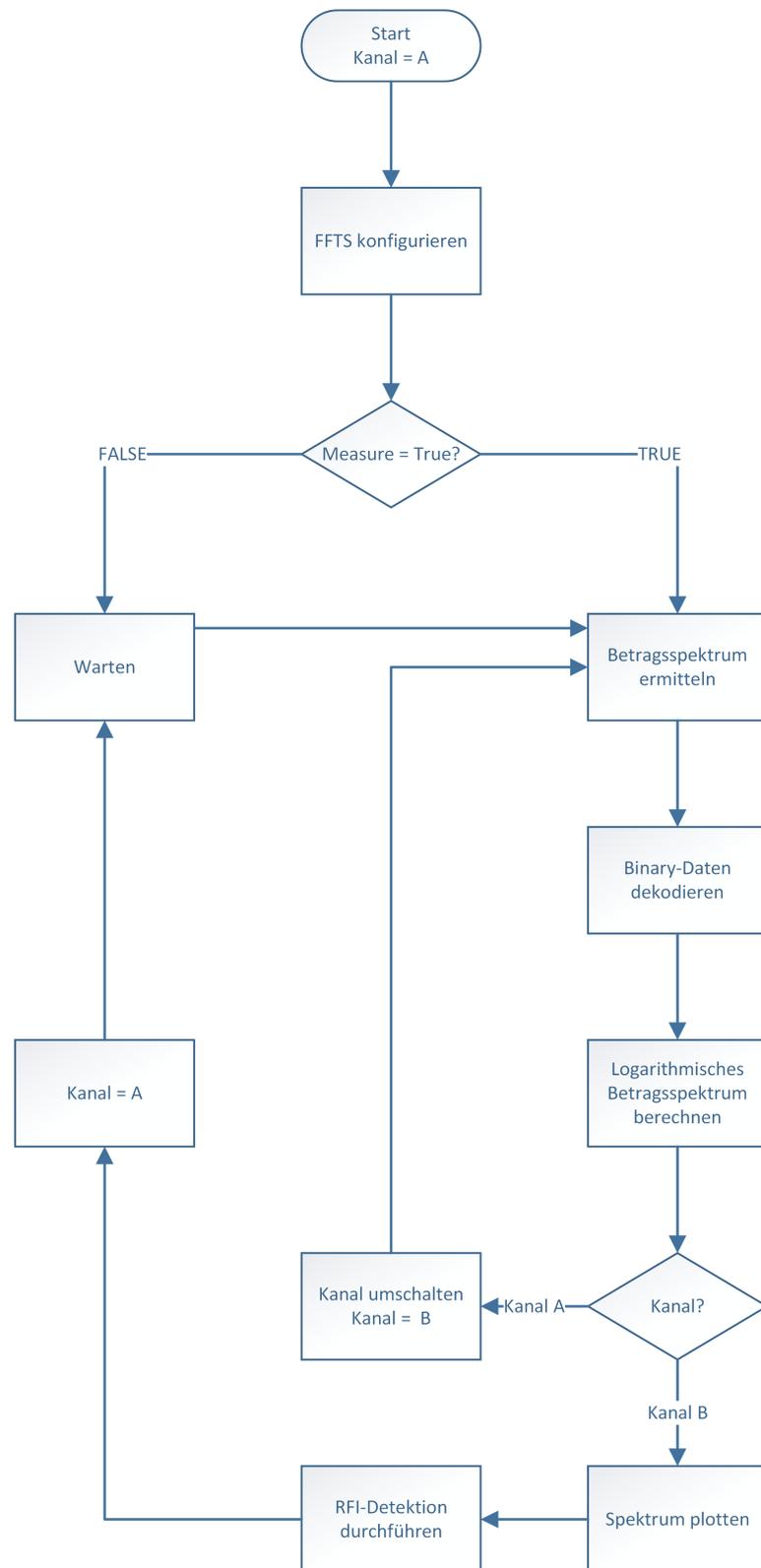


Abbildung 8.5: Programmflussplan zur Erfassung von zwei Kanälen mittels eines FFTS

8.3.3 Analoge Demodulation von Signalen zur Störerkennung

Das SDR dient nun ausschließlich zur Demodulation von Signalen. Dabei werden die empfangenen Signale mittels einer FM- oder AM-Demodulation demoduliert. Es wird nachfolgend ein Flowgraph in GNU Radio aufgebaut und zur Vorlage für die eigene Software verwendet (siehe Abbildung 8.6).

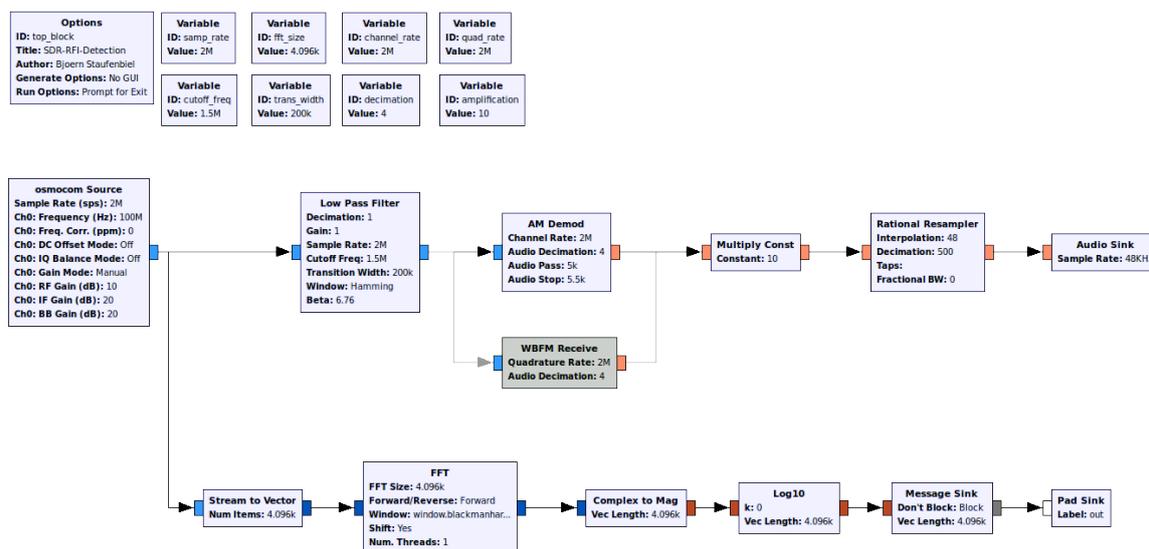


Abbildung 8.6: Flowgraph zur AM- und FM-Demodulation mit einem RTL-SDR

Die empfangenen Signale des SDR werden mittels eines digitalen Tiefpassfilters vorgefiltert. So wird das Rauschen im Signal minimiert und die Audiowiedergabe wird bei der erfolgreichen Demodulation deutlicher. Die Grenzfrequenz und die Breite des Übergangsbereichs sind im Softwareinterface einstellbar und an die aktuelle Anwendung anpassbar. Zudem wird im Filter eine Dezimierung der Abtastrate um den Faktor vier vorgenommen.

Die digitalen Signale werden anschließend durch entsprechende AM- oder FM-Blöcke demoduliert. Die Abtastrate des demodulierten Signals wird über einen Resampler auf 44,1 kHz umgesetzt. Da die Soundkarte diese Abtastrate nutzt, muss für die Audiowiedergabe die Abtastfrequenz korrekt eingestellt sein, da sonst Fehler bei der Wiedergabe entstehen. Die Art der Demodulation wird, wie auch die Filtereinstellungen, über das grafische Interface gesteuert. Im Programm selbst werden die Verbindungen zwischen den Blöcken beim Umschalten auf eine andere Modulation entsprechend neu konfiguriert.

Anhand einer einfachen Demodulation der UKW-Rundfunksignale konnte die korrekte Funktion der Demodulation der Signale bestätigt werden. Bei einer entsprechenden Einstellung konnten die Signale je nach Filtereinstellung klar und deutlich wiedergegeben werden. Die Demodulation beschränkt sich auf die zwei häufigsten analogen Modulationsarten und dient als Zusatz zur spektralen Detektion von Störsignalen.

Da für die Demodulation von digitalen Modulationssignalen das Wissen von vielen Parametern notwendig ist, um die Signale korrekt zu demodulieren, beschränkt sich diese Arbeit auf eben diese analogen Modulationsverfahren.

8.4 Integration der Hardware in das ZF-System

Das FFTS wird in diesem Abschnitt zusammen mit einem SDR, in diesem Fall einem DVB-T Stick mit R820T-Tuner, an das ZF-System angeschlossen. Dabei wird der zuvor gebaute Umschalter zur Erfassung beider ZF-Kanäle eingesetzt. Der Abschnitt diskutiert weitere Probleme, die bei der Durchführung einiger Testmessungen und der Inbetriebnahme aufgetreten sind.

8.4.1 Ermittlung des maximal möglichen Signal-Rauschverhältnisses

Um das maximal mögliche Signal-Rausch-Verhältnis des FFTS für ein Störsignal innerhalb eines ZF-Bandpassspektrums zu bestimmen, wurde der in Abbildung 8.7 zu sehenden Versuchsaufbau erstellt. Auf der einen Seite wird bandpassgefiltertes Rauschen im Frequenzbereich der Schmalband-ZF (100 - 200 MHz) erzeugt. Das Rauschen hat im Durchlassbereich des Filters dabei einen absoluten Pegel von -45 dBm. Diesem Rauschen wird auf der anderen Seite ein Nutzsinalträger überlagert, dessen Pegel variabel eingestellt werden kann. Das überlagerte Signal wird auf den Eingang A des entwickelten Kanalschalters gegeben. Der Kanal B wird als Referenz genommen, um den Pegel des bandpassgefilterten Rauschens konstant und nahe an einem Referenzspektrum zu orientieren.

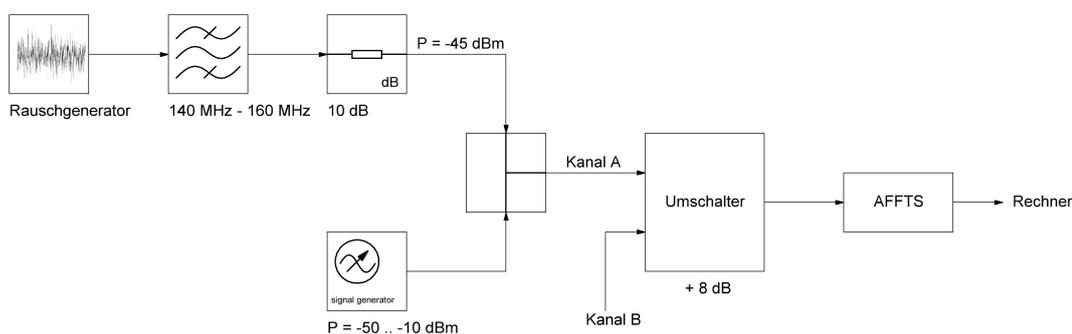


Abbildung 8.7: Versuchsaufbau zur Bestimmung des maximal möglichen Signal-Rausch-Verhältnisses des FFTS im Nutzbandpassspektrum

Am Ausgang des FFTS ergibt sich nun die Darstellung des bandpassgefilterten Rauschsignals mit einem relativen Pegel von -30 dB. Das Grundrauschen liegt bei ungefähr -50 dB.

Hier gehen also 20 dB an maximalem Signal-Rausch-Verhältnis durch das Bandpassspektrum des ZF-Signals verloren. Der Pegel des Signalgenerators wird nun so lange erhöht, bis das Nutzsinal aus dem Bandpassrauschen hervortritt. Der absolute Pegel wird dabei bis zum Erreichen des 1-dB-Kompressionspunktes des FFTS erhöht. Das Ergebnis des aufgenommenen Ausgangspegels für verschiedene Nutzsinalpegel ist in Abbildung 8.8 dargestellt. Trotz des ZF-Bandpassspektrums stehen weiterhin über 30 dB an Dynamikumfang zur Verfügung. Somit ergibt sich ein maximal mögliches Signal-Rausch-Verhältnis von knapp 50 dB.

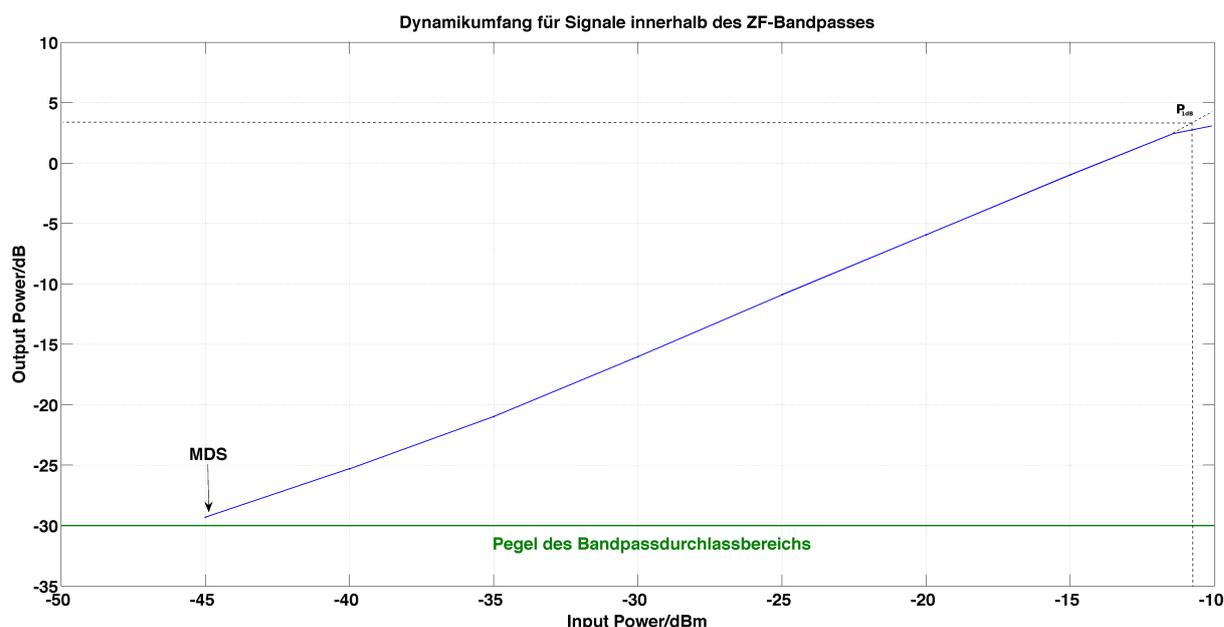


Abbildung 8.8: Maximales Signal-Rausch-Verhältnis des FFTS oberhalb des Bandpasssignals der Zwischenfrequenz

Das maximale Signal-Rausch-Verhältnis des A/D-Wandlers kann theoretisch über die Beziehung

$$\text{SNR}_{\max} = 6 \cdot b, \quad (8.1)$$

mit b , der Auflösung des A/D-Wandlers, berechnet werden. Für die A/D-Wandlung des FFTS ergibt sich so mit einer Auflösung von 8 Bit ein maximales Signal-/Rauschverhältnis von 48 dB. Das bedeutet, dass der Unterschied zwischen dem Pegel des Grundrauschens und des maximal erfassbaren Nutzsinalpegels 48 dB beträgt. Diese Berechnung kann mit Hilfe der Formel

$$\text{SNR}_{\max} = 6,02 \cdot b + 1,76 + 10 \cdot \log_{10} \left(\frac{f_S}{2 \cdot f_{\max}} \right) \text{ dB} \quad (8.2)$$

exakter durchgeführt werden.

In diesem Fall ergibt sich das maximale Signal-Rausch-Verhältnis für den A/D-Wandler des FFTS zu

$$\begin{aligned} \text{SNR}_{\max} &= 6,02 \cdot 8 + 1,76 + 10 \cdot \log_{10} \left(\frac{3 \text{ GS/s}}{2 \cdot 1,5 \text{ GHz}} \right) \text{ dB} \\ &= 49,92 \text{ dB.} \end{aligned} \quad (8.3)$$

Die Berechnung aus Formel 8.3 zeigt, dass das theoretisch berechnete, maximal mögliche Signal-Rausch-Verhältnis mit den in Darstellung 8.8 gezeigten Ergebnissen der Messung korreliert [37, S. 42-43].

8.4.2 Anpassung des Systems an unterschiedliche Eingangspegel

Da es dazu kommen kann, dass Empfänger ihre Signale auf unterschiedlich hohen Pegeln übertragen, muss die Leistung am Ausgang des sogenannten MultiFiBa stets konstant sein. Denn an den Ausgängen des MultiFiBa hängen die Backends, mit denen Astronomen ihre Messungen durchführen sowie das entwickelte RFI-Monitoring System. Da diese einen konstanten Pegel erwarten, muss die Leistung der Empfangssignale geregelt werden. Dies geschieht durch den Operateur. Zu deren Aufgabe gehört es, den Pegel zu beobachten und gegebenenfalls anzupassen. Die Anpassung erfolgt mittels eines Power Leveling Controllers. Um die korrekte Einstellung der Dämpfung zu finden, welche variabel in einem Bereich von 0 bis 31,5 dB ist, wird ein XFFTS zur Hilfe genommen und die Aussteuerung der A/D-Wandler betrachtet.

Die Darstellung 8.9 zeigt anhand eines Beispiels, wie eine solche Einstellung vorgenommen werden kann. Der Operateur erkennt anhand eines Softwaretools die Aussteuerung der A/D-Wandler für beide Kanäle eines XFFTS. Zentral dargestellt sind drei verschiedene Fälle für die Einstellung der Signaldämpfung. Der erste Fall zeigt eine zu geringe Aussteuerung. In diesem Fall ist die Dämpfung zu verringern. In der Mitte dargestellt ist die optimale Einstellung. Hier sind die A/D-Wandler gleichmäßig ausgesteuert und es ist eine gaußförmige Verteilung zu erkennen. Der letzte Fall zeigt eine zu geringe Dämpfung beziehungsweise gar keine Signaldämpfung. Anhand dieses Bildes wird deutlich, dass die A/D-Wandler übersteuert sind. Die Aufgabe des Operateurs besteht darin, den mittleren Fall stets einzuhalten. So ist es gesichert, dass am Ausgang des MultiFiBa ein konstanter Signalpegel zu erwarten ist und somit auch am Eingang des verwendeten FFTS.

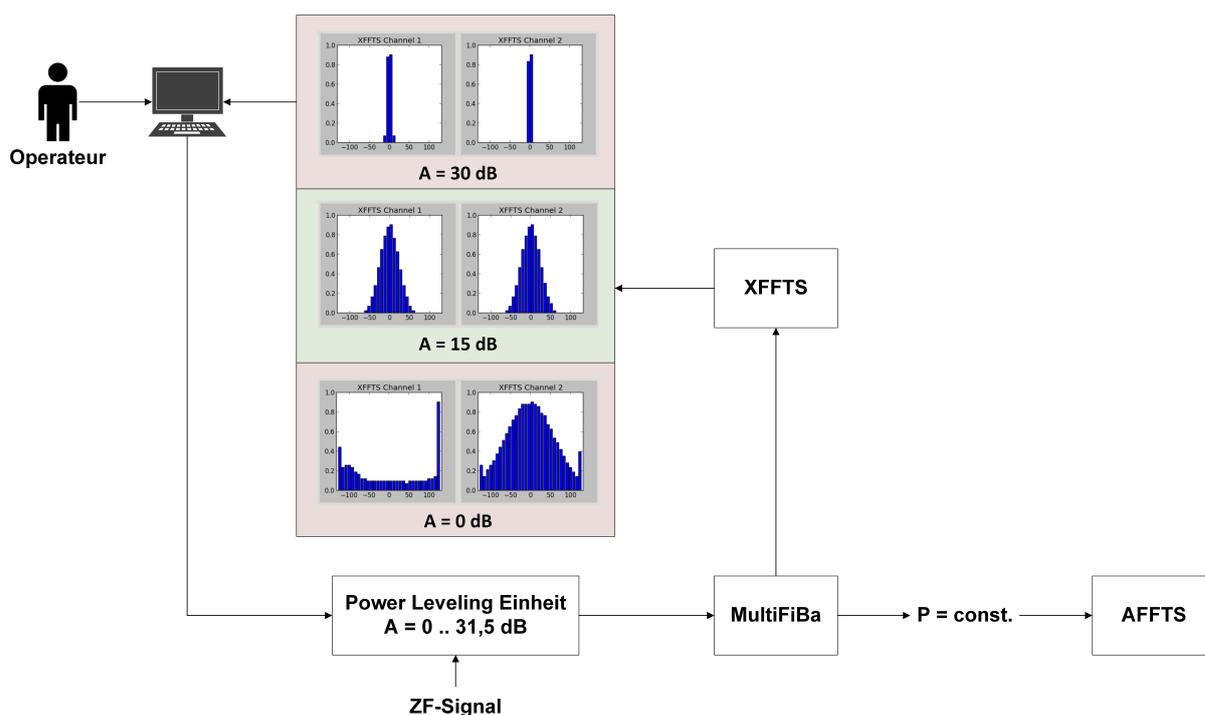


Abbildung 8.9: Prinzip der dynamischen Anpassung des Signalpegels durch den Operateur

8.4.3 Bestimmung der Eingangsanpassung des FFTS

Die Abbildungen 8.10 und 8.11 zeigen den Eingangsreflexionsfaktor des FFTS als Betragsdarstellung und im Smithdiagramm zwischen 100 kHz und 2 GHz. Es ist gut zu erkennen, dass die Anpassung für kleine Frequenzen sehr gut ist. Zwischen 400 und 500 MHz und oberhalb von 1 GHz ist die Anpassung jedoch relativ schlecht. Diese Fehlanpassung führt zu einigen Problemen die nachfolgend dargestellt werden.

In Abbildung 8.12 ist die Aufnahme des Spektrums eines breitbandigen Rauschsignals dargestellt. Durch die Fehlanpassung wird an einigen Stellen des Spektrums so viel Leistung vom Eingang reflektiert, dass sich deutliche Senken im Spektrum herausarbeiten. Zudem besitzt das FFTS kein Anti-Aliasing-Filter. Somit können einige Unebenheiten im Spektrum zudem durch das Auftreten von Aliasing aufgrund spektraler Anteile oberhalb von 1,5 GHz erklärt werden. Um diesem Effekt zu verhindern, wird ein einfaches Tiefpassfilter mit einer Grenzfrequenz von 1 GHz vor das FFTS geschaltet. So können Aliasing-Effekte einfach, aber effizient verhindert werden, da das Spektrum auf 1,5 GHz bandbegrenzt ist.

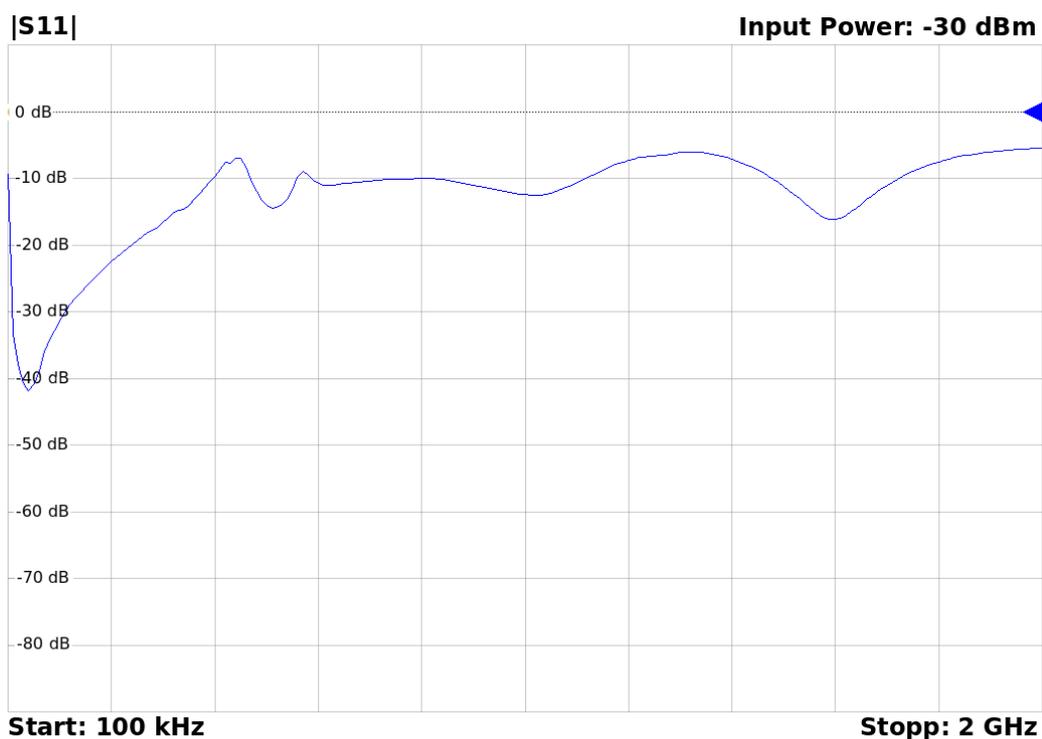


Abbildung 8.10: Betrag der Eingangsanpassung des FFTS

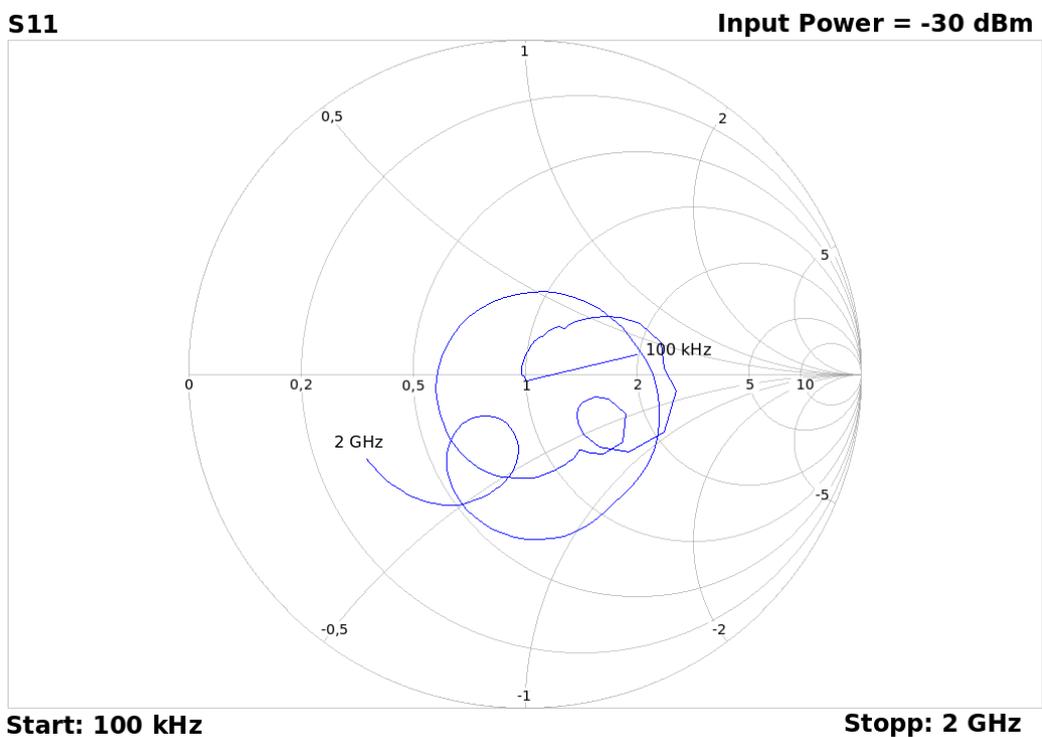


Abbildung 8.11: Eingangsreflexionsfaktor des FFTS im Smithdiagramm

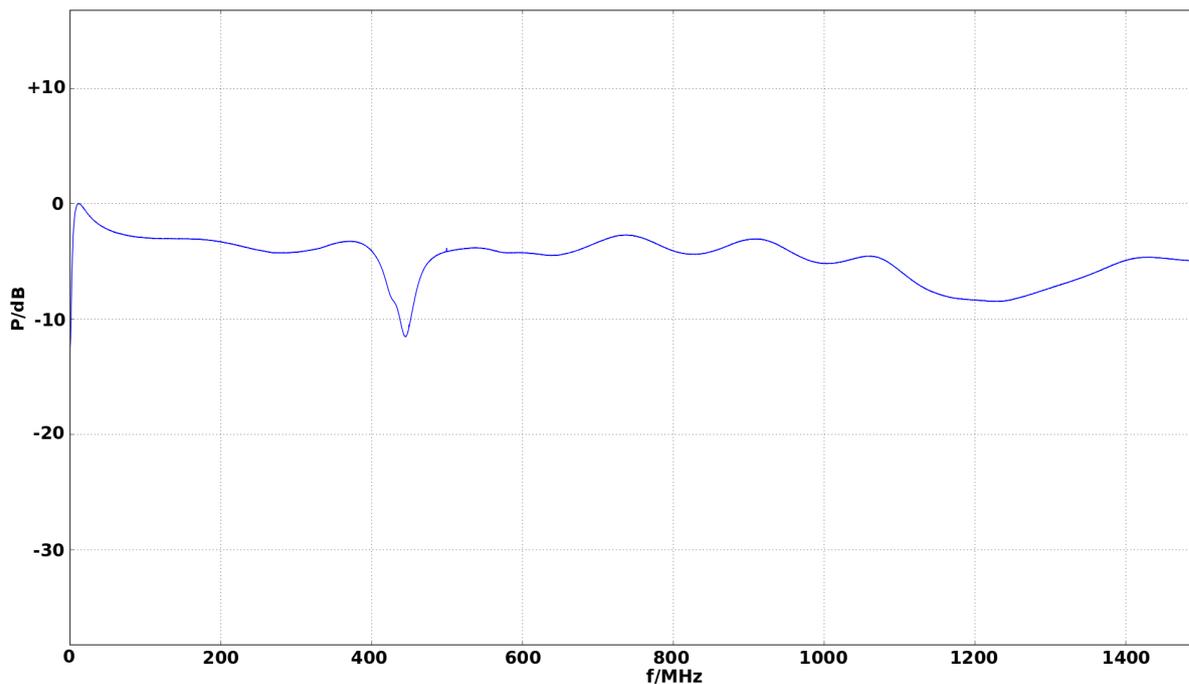


Abbildung 8.12: Spektrum eines breitbandigen Rauschsignals ohne Eingangsanpassung und Anti-Aliasing-Filter

Wird der Eingang des FFTS mit einem Abschlusswiderstand beschaltet, arbeiten sich einige Eigenstörlinien im Spektrum heraus (siehe Abbildung 8.13). Das liegt daran, dass das FFTS bei zu geringem oder gar keinem Eingangspegel Fehler bei der Berechnung des Spektrums macht. Daraus folgt, dass der Pegel am Eingang des FFTS immer hoch genug sein muss, damit diese Störungen nicht aus dem Spektrum heraustreten. Dies wird durch die in Abschnitt 8.4.2 beschriebene Methode stets gewährleistet.

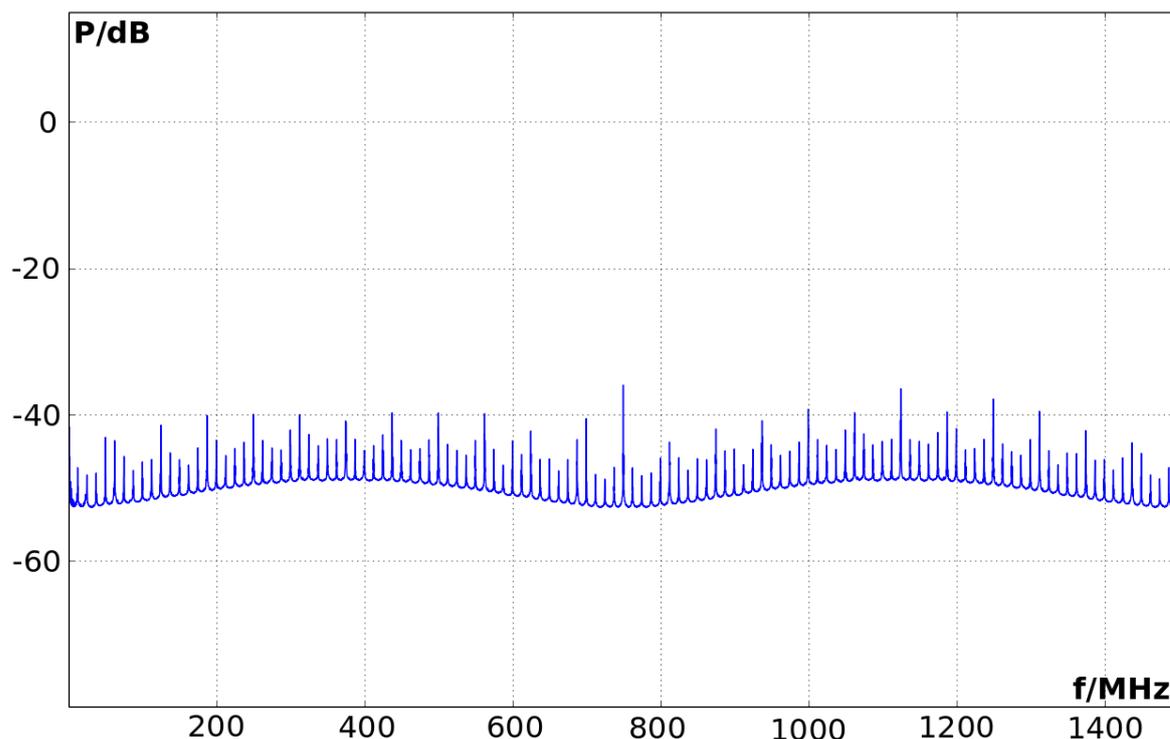


Abbildung 8.13: FFTS-Spektrum bei einem mit 50Ω abgeschlossenen Eingang

Um die Fehlanpassung des Eingangs des FFTS zu minimieren, wird ein einfaches Dämpfungsglied vor den Eingang geschaltet. In Kombination mit einem Tiefpassfilter ergibt sich das in Abbildung 8.14 zu sehende Spektrum. Dieses entspricht den Erwartungen und zeigt ein angelegtes Rauschsignal zwischen 0 und 1 GHz. Oberhalb wird das Signal durch das Tiefpassfilter bandbegrenzt. Daraus folgt ein Problem. Denn die verfügbaren 8192 spektralen Kanäle verteilen sich auf die komplette Bandbreite von 1,5 GHz. Durch diese Beschaltung werden Signale oberhalb von 1 GHz abgeschnitten und über 2700 spektrale Kanäle gehen verloren. Es wäre sinnvoll, diese nicht verwendbaren Kanäle auf die Nutzbandbreite aufzuteilen und die spektrale Auflösung so weiter zu erhöhen. Da dies jedoch nicht ohne Weiteres möglich ist und eine tiefgehende Umprogrammierung des FFTS-Kernels notwendig wäre, wird diese Einschränkung hingenommen, da die Frequenzgenauigkeit mit 183 kHz bereits sehr gut ist.

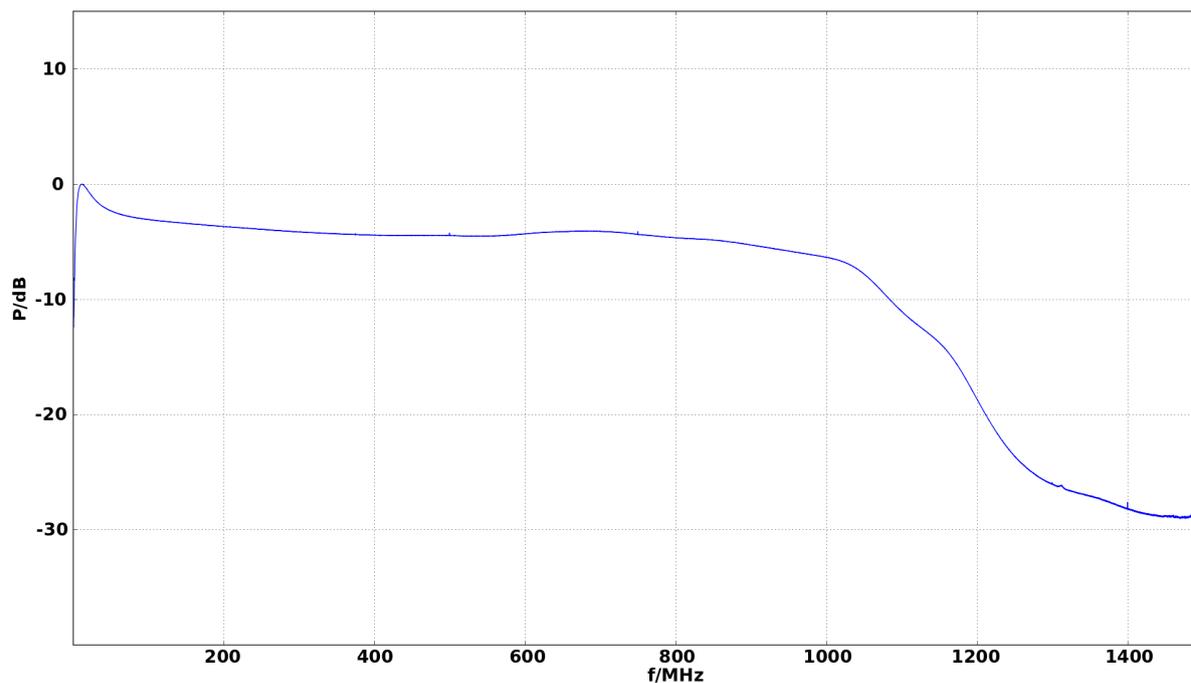


Abbildung 8.14: Spektrum eines breitbandigen Rauschsignals mit angepasstem FFTS-Eingang und Anti-Aliasing-Filter

8.4.4 Darstellung des Systemaufbaus

Die Abbildung 8.15 zeigt die Integration des FFTS und des SDR an das ZF-System. Das ZF-Signal, bestehend aus einer Leitung für Kanal A und einer für Kanal B, wird auf den bereits erwähnten Umschalter gegeben. Über je ein 5-dB-Dämpfungsglied wird eine einfache Anpassung des Filtereingangs und -ausgangs vorgenommen, der zur Vermeidung von Aliasing im FFTS dient. Die Datensignale des FFTS und des SDR werden per Ethernet an einen Rechner übertragen. Auf der einen Seite wird so das Spektrum dargestellt und RFI detektiert, auf der anderen Seite werden Signale demoduliert und über die Computerlautsprecher wiedergegeben.

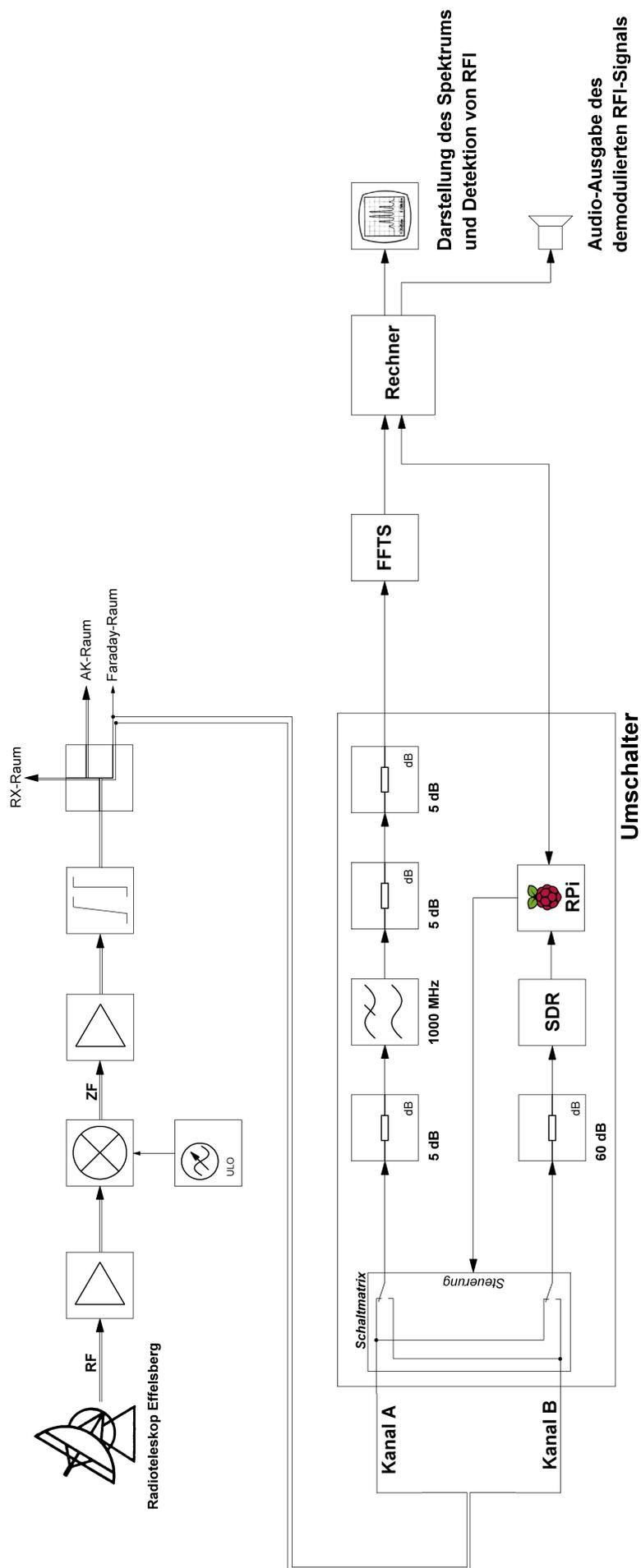


Abbildung 8.15: Integration des FFTS und des SDR in das ZF-System des MPIfR

8.5 Testmessungen an der Zwischenfrequenz mit einem FFTS

Mit dem oben gezeigten Systemaufbau und der entwickelten Software wurden einige Testmessungen durchgeführt. Einige der Ergebnisse werden nachfolgend dargestellt. Zu Beginn wurde das Spektrum des 18-21 cm Empfängers auf einem Kanal erfasst. Das Ergebnis ist in Abbildung 8.16 zu sehen. Im Empfangsbereich um 150 MHz in Zwischenfrequenzlage sind einige deutliche Störungen im Spektrum zu erkennen. Die Mitte des Bandpasses entspricht der Sky-Frequenz 1,666 GHz. Solche deutlich erkennbaren Störungen sind selten. Jedoch kann durch die kurze Programmlaufzeit ein kurzzeitig auftretendes Störsignal relativ einfach erfasst werden. In diesem Fall kommen die Signale durch ein in der Nähe des Radioteleskops aufgestelltes Radar zustande. Diese Störsignale wurden bereits zuvor mit den SDRs erfasst.

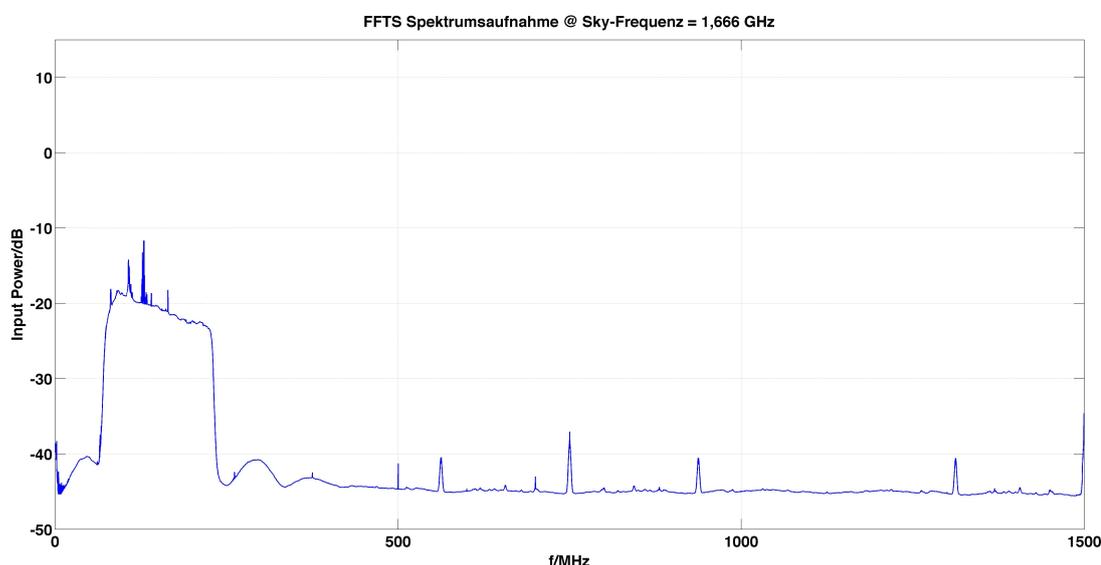


Abbildung 8.16: Aufnahme eines Signalspektrums des 18-21-cm-Empfängers

Um mit dem aufgebauten System eine bekannte Störung zu erfassen und zuzuordnen, wird ein mittels eines Signalgenerators erzeugtes Signal über eine Antenne an das Radioteleskop übertragen. Das Signal hat dabei eine Frequenz von 12,18 GHz. Als aktiver Empfänger dient der 2,2 cm Empfänger. Der Versuchsaufbau ist in Abbildung 8.17 dargestellt.

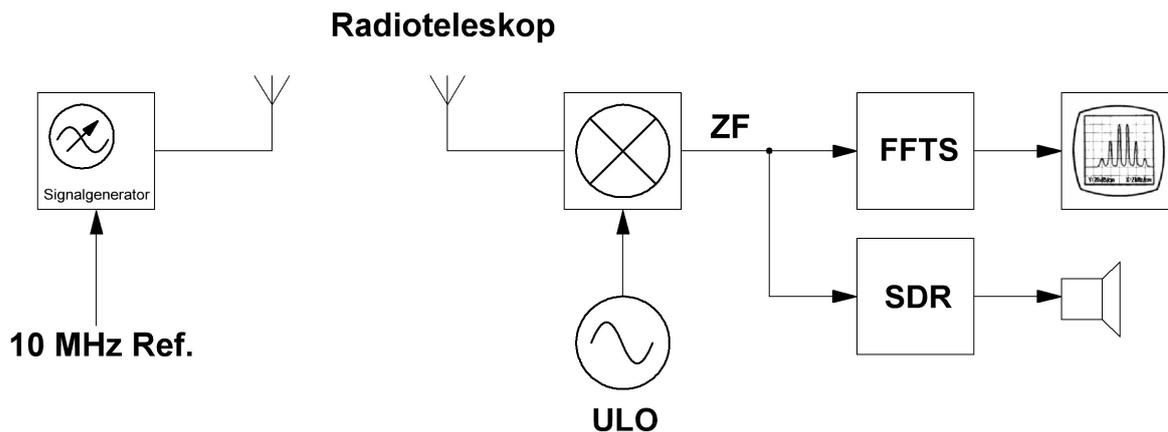


Abbildung 8.17: Versuchsaufbau zur Erfassung eines künstlich eingespeisten RFI-Signals

Die Abbildung 8.18 zeigt das Ergebnis der Messung. Das Störspektrum ist im Durchlassbereich des Bandpasses gut zu erkennen. Beim Überfahren der Störlinie im QT-GUI mit der Maus wird im unteren Bereich die Sky-Frequenz angezeigt. In diesem Fall war exakt die eingestellte Frequenz zu erkennen. Dieser Versuch bestätigt noch einmal die korrekte und zuverlässige Funktion des FFTS.

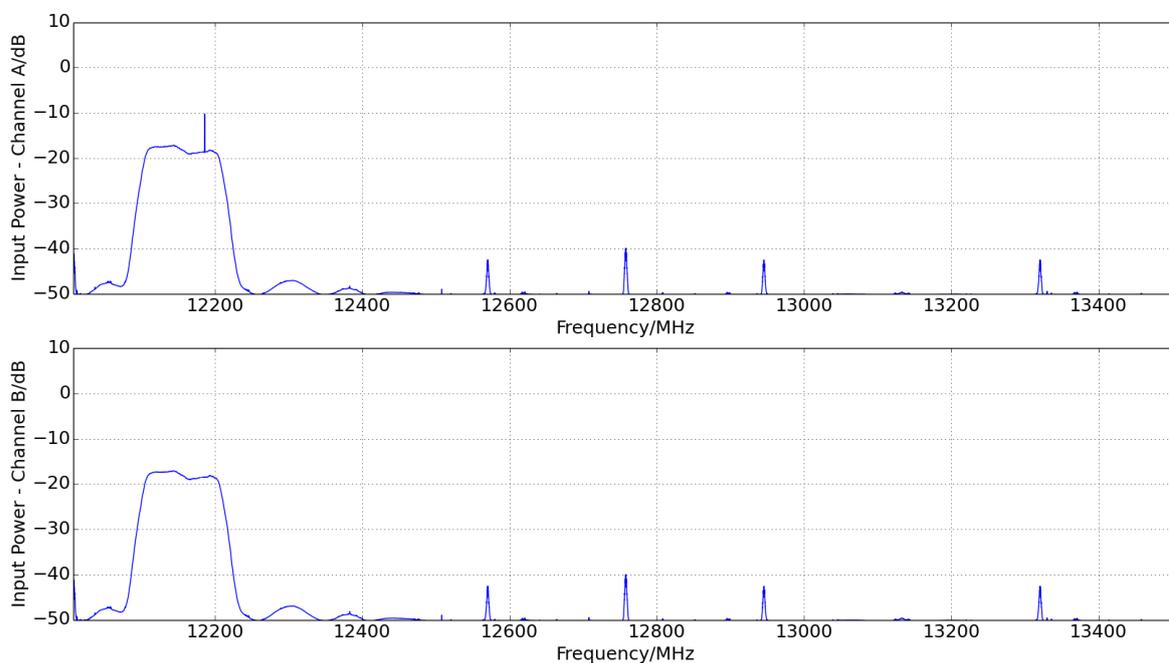


Abbildung 8.18: Spektrum des erzeugten RFI-Signals mit deutlich erkennbarer Störlinie bei 12,18 GHz auf Kanal A

Um ein bekanntes Signal über das aufgebaute System zu demodulieren, wird ein verstärktes UKW-Signal mit bandpassgefiltertem Rauschen überlagert, das im spektralen Bereich der Schmalband-ZF, also zwischen 100 und 200 MHz, liegt. Das für diesen Test verwendete Bandpassfilter hat jedoch einen etwas breiteren Durchlassbereich. Das mit einem Dipol empfangene UKW-Signal wird dem Rauschsignal überlagert und über den Kanal A an den Umschalter und so an das FFTS und das SDR gegeben. Das mit dem FFTS erfasste Spektrum ist in Abbildung 8.19 zu sehen.

Das SDR wird auf die entsprechende Frequenz eines UKW-Senders eingestellt. Die Abbildung 8.20 zeigt auf der einen Seite das Spektrum des am Eingang des SDR anliegenden Signals, auf der anderen Seite den zeitlichen Verlauf des demodulierten FM-Signals. Mit Hilfe dieses Versuchs konnte das Zusammenspiel zwischen beiden Komponenten überprüft und die korrekte Funktion bestätigt werden.

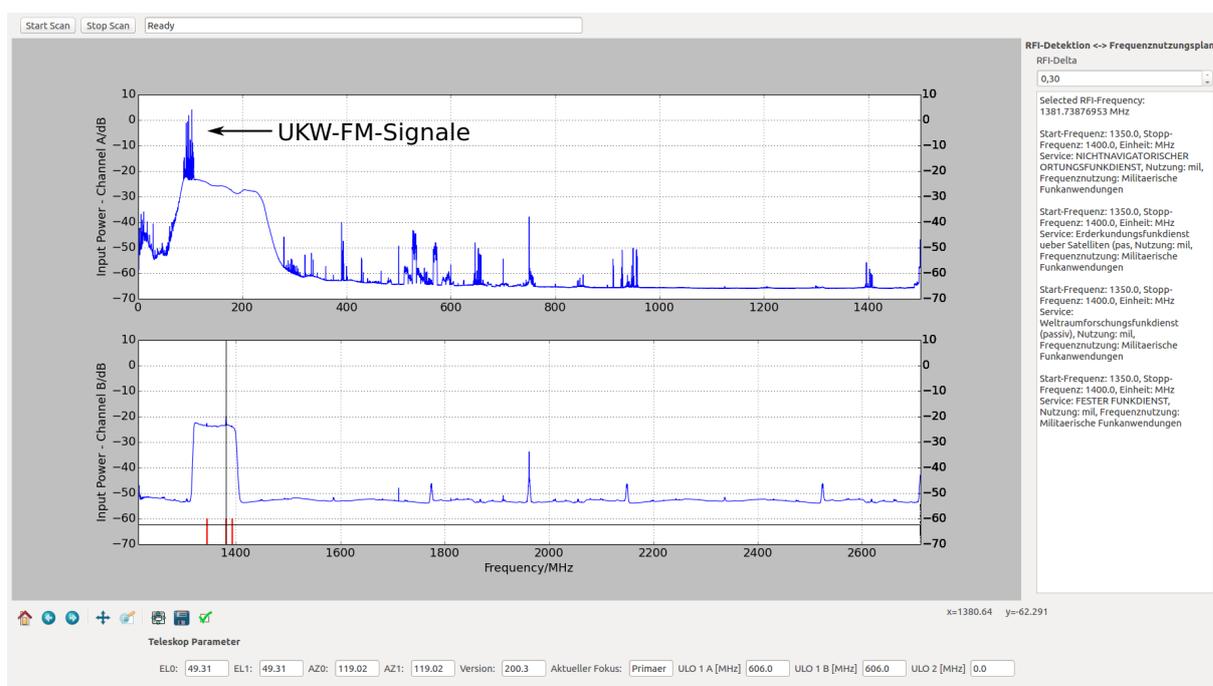


Abbildung 8.19: Spektrum des UKW-Signals mit überlagertem bandpassgefiltertem Rauschen in Kanal A

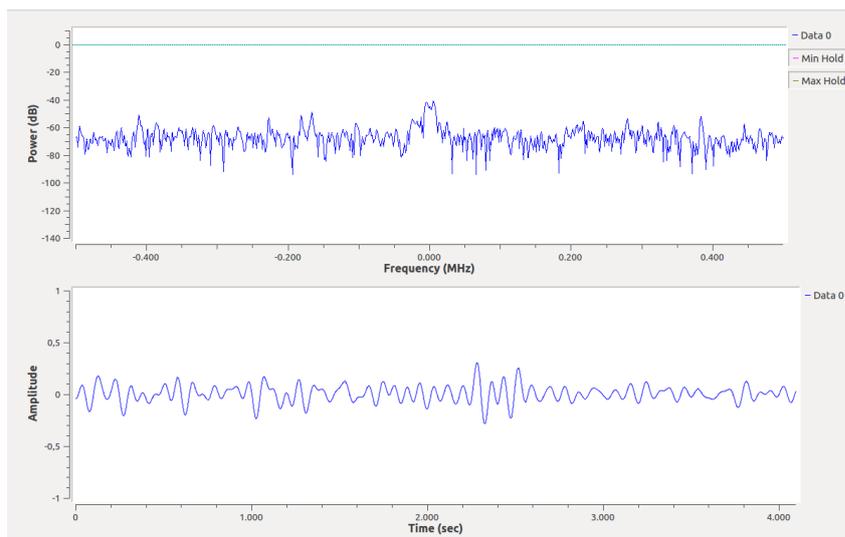


Abbildung 8.20: Spektrum des UKW-Signals am Eingang des SDR und der zeitliche Verlauf des demodulierten FM-Signals

Die nachfolgende Abbildung 8.21 zeigt das Spektrum eines 433-MHz Senders. Das eigentliche Nutzsignalspektrum liegt knapp unterhalb von 433 MHz. Der Ausschlag in der Mitte des Spektrums kommt durch das LO-Leakage zustande. In der unteren Hälfte der Abbildung ist das demodulierte ASK-Signal zu sehen. Da die Mischfrequenz nicht exakt der Signalfrequenz entspricht, ist dem Nutzsignal eine oszillierende Komponente überlagert. Zudem ist das Signal relativ stark verrauscht. Trotzdem sind die Rechteck-Impulse des ASK-Signals gut zu erkennen. Im Kapitel A.3 des Anhangs sind weitere Beispiele detektierter Störungen zu finden, die mithilfe des aufgebauten Systems über das Radioteleskop Effelsberg empfangen wurden.

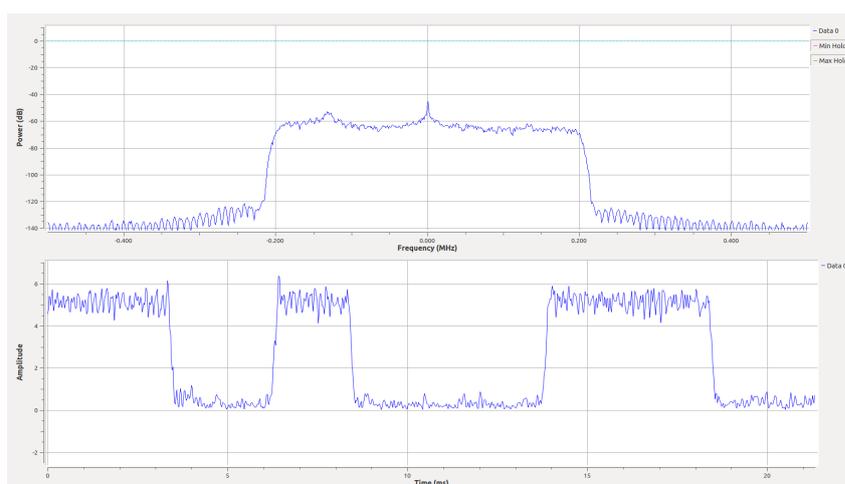


Abbildung 8.21: Spektrum eines 433-MHz ASK-Signals (tiefpassgefiltert) und das demodulierte Zeitsignal

8.6 Implementierung einer Funktion zur spektralen Detektion von RFI über die Wahrscheinlichkeitsdichtefunktion von Signalen

In den vorhergegangenen Kapiteln wurde zur spektralen Detektion ein von MATLAB adaptierter Algorithmus zur Bestimmung von lokalen Maxima verwendet. Dieser Algorithmus funktionierte gut und im Zusammenspiel mit dem in einer Datenbank gespeicherten Frequenznutzungsplan konnte so per Mausklick die mögliche Quelle der Störung einfach ermittelt werden. Die Abbildung 8.22 zeigt die Anwendung dieses Algorithmus auf ein mit dem FFTS ermittelten Spektrum. Dieser Abschnitt zeigt einen alternativen Algorithmus, der auf der Bestimmung der Standardabweichung von Ausschnitten des Spektrums beruht. Über die Veränderung der Standardabweichung wird dabei auf das Vorhandensein einer Störung geschlossen.

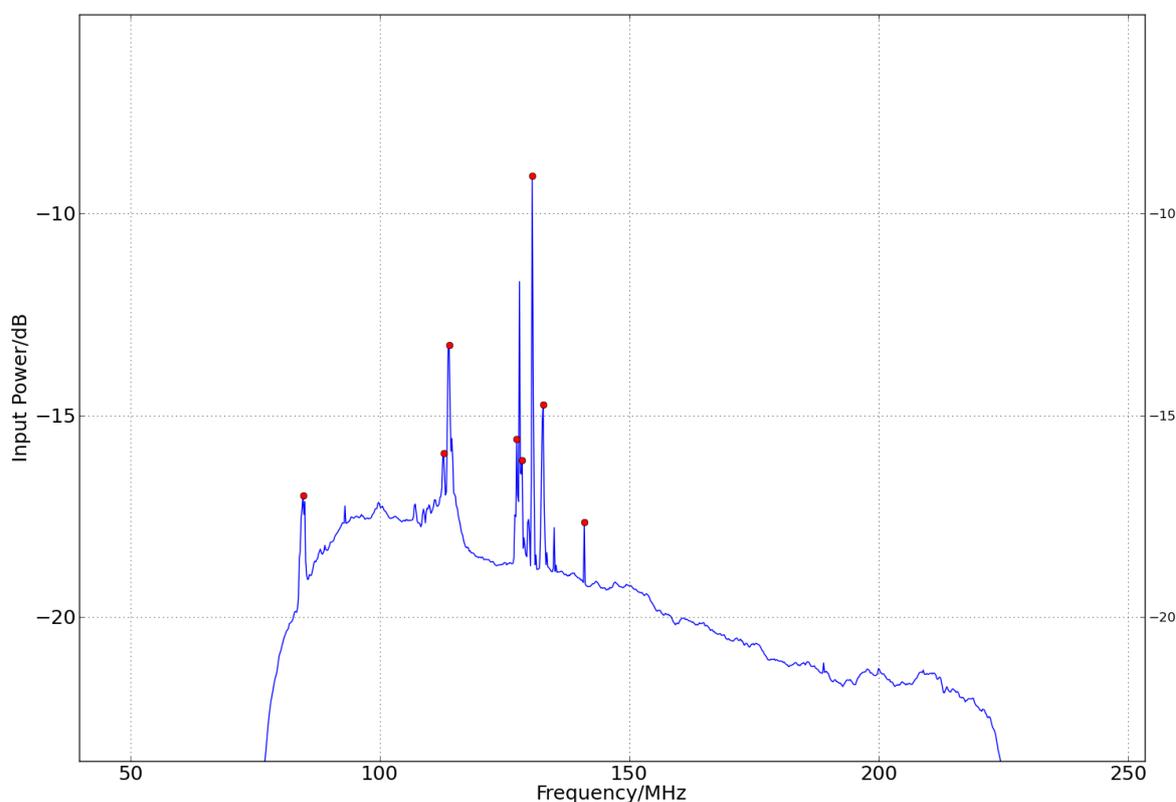


Abbildung 8.22: Detektierte Störungen mittels des Algorithmus zur Bestimmung von lokalen Maxima in der Schmalband-ZF

8.6.1 Prinzipielle Darstellung der Methode

Der im Nachfolgenden dargestellte Algorithmus verfolgt eine komplexe Idee zur Erkennung von RFI. Diese entstammt einem Bericht aus dem "Astronomical Journal" aus dem Jahre 2010 [4]. Bei dieser Methode wird die Veränderung der Standardabweichung durch in einem Rauschsignal vorhandene Signale ermittelt und als Maß für das Vorhandensein von Störungen angenommen. Dazu muss die Wahrscheinlichkeitsdichtefunktion (PDF - Probability Density Function) des Spektrums blockweise bestimmt werden und aus dieser wiederum die Standardabweichung. Diese Zahl ist ein Maß für die Breite der Streuung eines Zufallswertes um einen Mittelwert [41]. Den Einfluss eines Störsignals in einem überlagerten Rauschsignal auf diese Verteilung zeigt die Abbildung 8.23. Dieser Effekt wird nachfolgend näher untersucht.

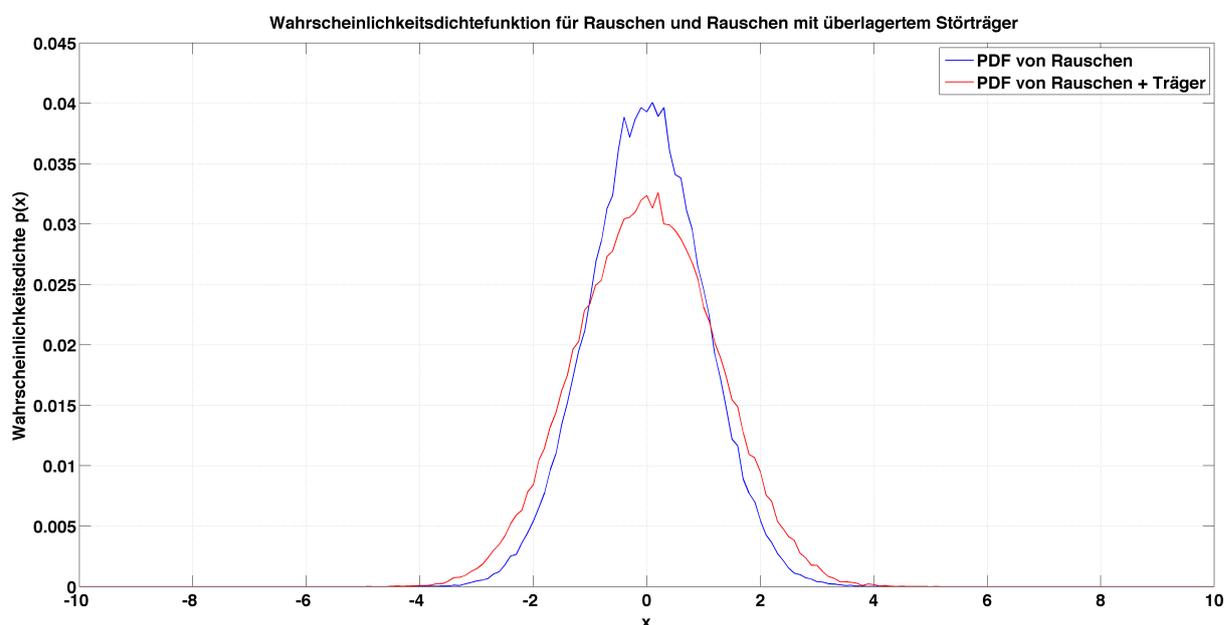


Abbildung 8.23: Verlauf der Wahrscheinlichkeitsdichtefunktion für ein Rauschsignal (blau) und ein Rauschsignal mit überlagertem Träger (rot)

Die oben stehende Abbildung zeigt den Verlauf der PDF für weißes Gaußrauschen. Die Funktion ist relativ schmal und die zufälligen Werte des mittelwertfreien Rauschens sind nahe um 0 verteilt. Das bedeutet, dass kleine Signalamplituden sehr wahrscheinlich sind, während große Werte der Amplitude nur sehr selten auftreten. Wird diesem Rauschen ein sinusförmiger Träger überlagert, so wird die Funktion erkennbar breiter. Die Standardabweichung wird somit größer und die Streuung der Werte um den Mittelwert nimmt zu. Wird davon ausgegangen, dass das Eigenrauschen des FFTS gaußförmiges Rauschen ist, kann über diese Methode das Vorhandensein eines Störers in einem Ausschnitt des Spektrums ermittelt werden. Der nächste Abschnitt erläutert, wie aufgrund dieser Feststellung mittels Software eine Störung in einem Rauschsignal ermittelt wird.

8.6.2 Implementierung des Prinzips in einem Python-Programm

Die Abbildung 8.24 zeigt detailliert die Vorgehensweise bei der Bestimmung von RFI mit dieser Methode. Im ersten Schritt wird das Betragsspektrum mittels des FFTS berechnet. Da je nach Empfängertyp immer nur ein Bereich der ZF von Interesse ist, wird das Spektrum auf den relevanten Frequenzbereich begrenzt. Anschließend wird das Spektrum in 8 MHz breite Teilspektren aufgeteilt, sogenannte Frequenzblöcke. Es gibt somit bei 8192 spektralen Kanälen exakt 1024 einzelne Frequenzblöcke. Für jeden dieser Blöcke wird die Standardabweichung ermittelt. Der Block mit der geringsten Standardabweichung wird als Referenz genommen. Denn im Block mit der geringsten Standardabweichung ist davon auszugehen, dass dort nur Rauschen vorhanden ist. Die Differenz der Standardabweichung von der so ermittelten minimalen Standardabweichung wird für jeden Block berechnet. Diese Differenz ist in der Grafik im grünen Plot für jeden Frequenzblock zu sehen.

Anschließend wird ein Threshold, also ein Schwellwert, festgelegt. Liegt die berechnete Differenz der Standardabweichung oberhalb dieses Wertes, wird für diesen Block davon ausgegangen, dass dort ein Störer vorhanden ist. Da das zufällige Rauschen des FFTS auch eine größere Standardabweichung haben kann als die minimale Standardabweichung, werden solche Blöcke durch diesen Schwellwert nicht als RFI detektiert. Alle Differenzen, die oberhalb dieses Wertes liegen, werden anschließend als "True" gekennzeichnet. Werte die unterhalb des Schwellwertes liegen, als "False". Für jeden Block ergibt sich so eine Darstellung, ob eine Störung vorhanden ist oder nicht. Da das Spektrum in 8 MHz breite Blöcke aufgeteilt wurde, hat das RFI-Array nur noch eine Länge von 1024 Werten. Mittels einer Interpolation mit dem Faktor acht wird das Signal wieder verbreitert. Ist ein Block mit "True" bewertet worden, so werden alle Werte des interpolierten Blocks ebenso gekennzeichnet. Bei einem Block mit einer "False" Bewertung werden die Interpolationswerte mit Nullen aufgefüllt. Das hat den Vorteil, dass das Spektrum und das RFI-Array in einem Plot dargestellt werden können. Markierungen im Plot deuten dann auf Frequenzblöcke mit detektierten Störungen hin.

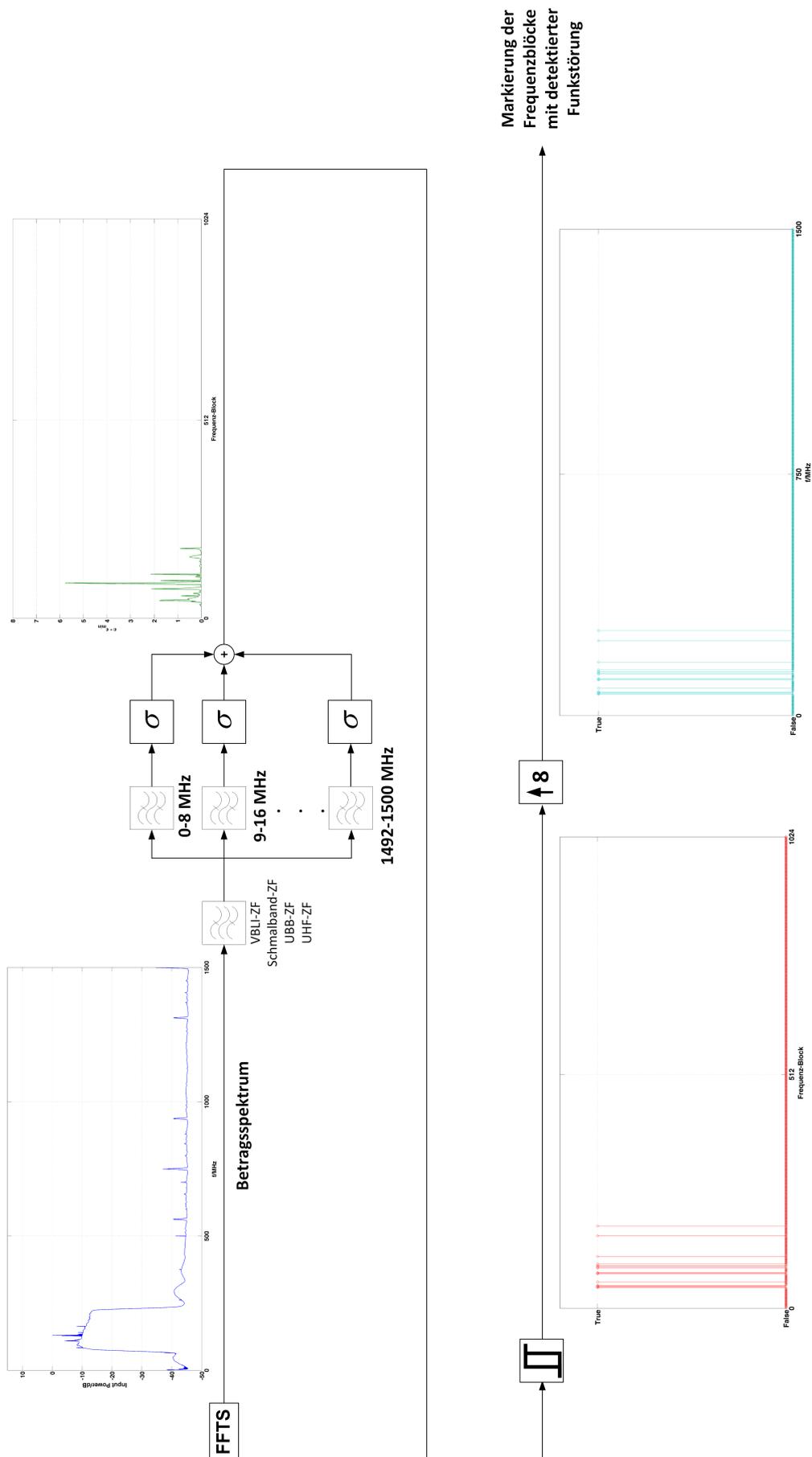


Abbildung 8.24: Prinzip zur Ermittlung von RFI in einem Signalspektrum

Das Ergebnis einer Messung und der Anwendung dieser Methode ist in Abbildung 8.25 zu sehen. Im Durchlassbereich des Bandpasses sind deutliche Störungen vorhanden. Die Bildung der Differenzen zwischen der minimalen Standardabweichung und der eines Blocks bestätigt diese Beobachtung. Die eigentliche Detektion validiert dieses Vorgehen weiterhin und markiert die gestörten Frequenzbereiche mit einem roten Peak.

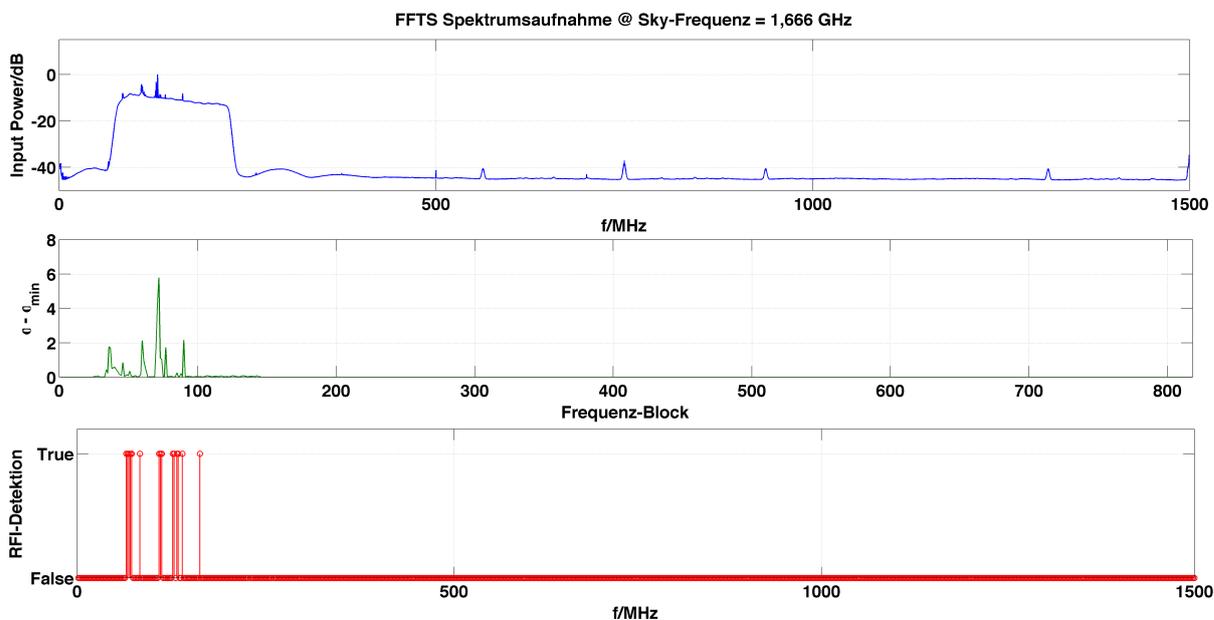


Abbildung 8.25: Spektrum des 18-21-cm-Empfängers mit berechneter Differenz der Standardabweichung der einzelnen Frequenzblöcke

Als Gegenprobe wurde das Signal eines Sekundärfokus-Empfängers betrachtet (siehe Abbildung 8.26). Im Spektrum sind keine Störungen im Nutzspektrum erkennbar. Die Differenzen zeigen ein entsprechendes Bild. Diese bewegen sich in einem Bereich zwischen 0 und maximal 0,5. Da keine Differenz den Schwellwert von 1 übersteigt, wird auch kein Peak zur RFI-Detektion eingezeichnet. Somit konnte die korrekte Funktionsweise des Algorithmus bestätigt werden.

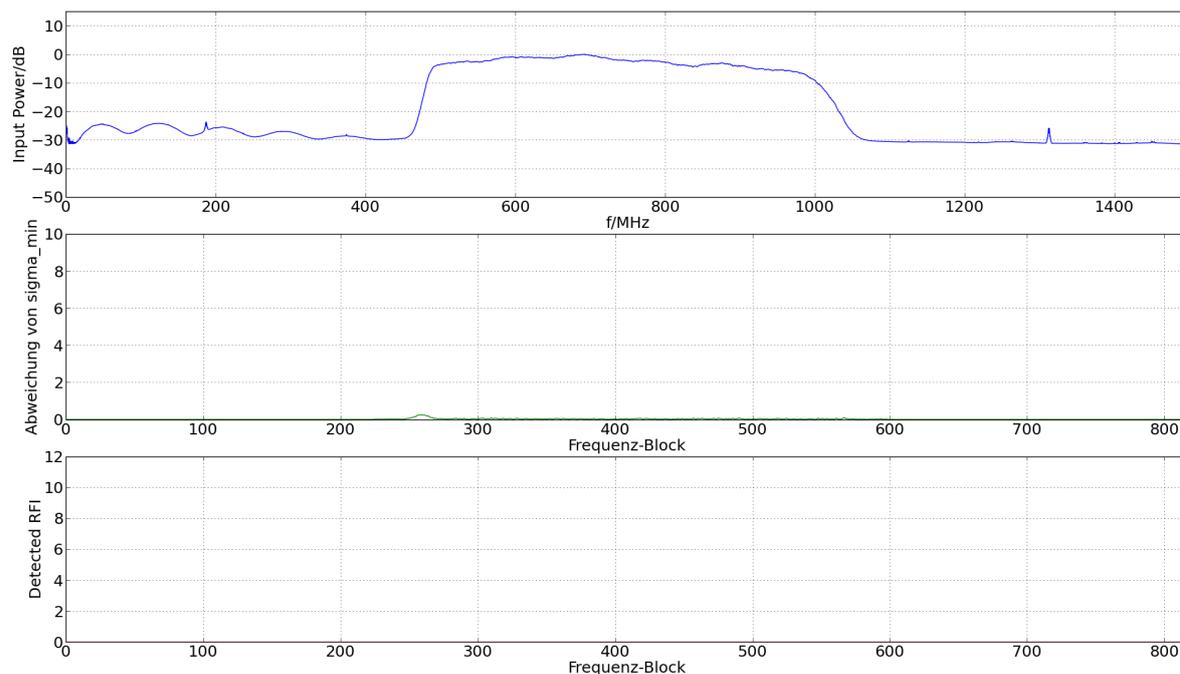


Abbildung 8.26: Störungsfreies Zwischenfrequenzsignal eines Sekundärfokus-Empfängers

Wird im Interface mit der Maus auf einen erkannten Störer geklickt, wird für die entsprechende Sky-Frequenz der passende Eintrag aus dem Frequenznutzungsplan der Bundesnetzagentur gezeigt, so wie es bereits zuvor implementiert wurde. Zudem wird bei dem Klick auf einen Störer die Mittenfrequenz des SDR auf die entsprechende Frequenz der Störung in der Zwischenfrequenzlage eingestellt. So ist es möglich, direkt eine AM- oder FM-Demodulation der Signale durchzuführen, ohne weitere Einstellungen vornehmen zu müssen. Gerade die AM-Demodulation erscheint dabei interessant, da so akustisch dem Verlauf des Störsignals gefolgt werden kann. Das macht es möglich, beispielsweise zu erkennen, ob es sich um ein gepulstes Signal handelt.

Die hier implementierte Funktion wird in Kombination mit der in Kapitel 7.4.2 geschriebenen Funktion zur Detektion von lokalen Maxima genutzt. So werden lokale Maxima nur dann eingezeichnet, wenn der Frequenzblock über den in diesem Kapitel vorgestellten Algorithmus als "True" gekennzeichnet wurde. Dies erhöht die Sicherheit, dass es nicht zu Fehldetektionen von RFI in störungsfreien Frequenzbereichen kommt, da diese durch den zusätzlichen Algorithmus gefiltert werden.

8.7 Erweiterung des RFI-Monitoring-Backends auf vier ZF-Signale

Bisher wurde ein RTL-SDR zur Signaldemodulation von zwei ZF-Kanälen genutzt, welche über einen Umschalter angewählt werden. Daraus ergibt sich der Nachteil, dass immer nur ein Kanal von allen Benutzern zur gleichen Zeit betrachtet werden kann. Da das Institut die Verwendung eines dritten und vierten Kanals plant, werden insgesamt vier RTL-SDRs in Betrieb genommen, um das System zukunftssicher zu machen. Diese sollen vor allem für Doppel-Horn-Systeme, wie beispielsweise dem 6-cm-Empfänger, genutzt werden.

8.7.1 Aufbau des Backend-Einschubs

Die Abbildung 8.27 zeigt den erweiterten Aufbau des RFI-Monitoring-Systems. Am Eingang dieses Systems werden die vier ZF-Eingangssignale über einen Splitter für die Einspeisung der Signale in das FFTS und in die SDR-Hardware aufgeteilt. Das einfache Pin-Dioden-Relais des zuvor aufgebauten Umschalters wird durch einen Vierfach-Umschalter des Herstellers Mini-Circuits ausgetauscht. Dieser wird durch ein im Institut entwickeltes μ C-Controller-Board gesteuert, welches entsprechend der über TCP gesendeten Symbole bis zu 8 Ausgänge eines μ C auf 'High' oder 'Low' setzt. Für diesen Umschalter werden jedoch nur zwei der Ausgänge benötigt. Die nachfolgende Signalverarbeitung für das FFTS hat sich nicht gegenüber dem ursprünglichen Aufbau verändert. Die Signale werden verstärkt und durch ein Tiefpassfilter zur Vermeidung von Aliasing gefiltert und an das FFTS übertragen.

Die Hardware wird zudem um drei Raspberry Pi des Typs B+ erweitert sowie der zuvor genutzte Raspberry Pi des Typs B durch einen Typ B+ ersetzt. Als SDR-Hardware kommen DVB-T-Sticks des Herstellers GiXa Technology mit einem R820T-Tuner zum Einsatz. Zur individuellen Einstellung der Eingangspegel an jedem SDR werden diesen zudem digital einstellbare Dämpfungsglieder vorgeschaltet. Um Schäden durch zu hohe Signalpegel zu vermeiden, ist außerdem ein festes Dämpfungsglied vorgesehen. Die Anbindung der Raspberry Pis und des Lantronix Controllers an das Netzwerk erfolgt über einen Gigabit-Ethernet-Switch.



Abbildung 8.27: RFI-Monitoring Backend mit Anschluss von vier ZF-Signalen und Integration von vier RTL-SDR-Servern

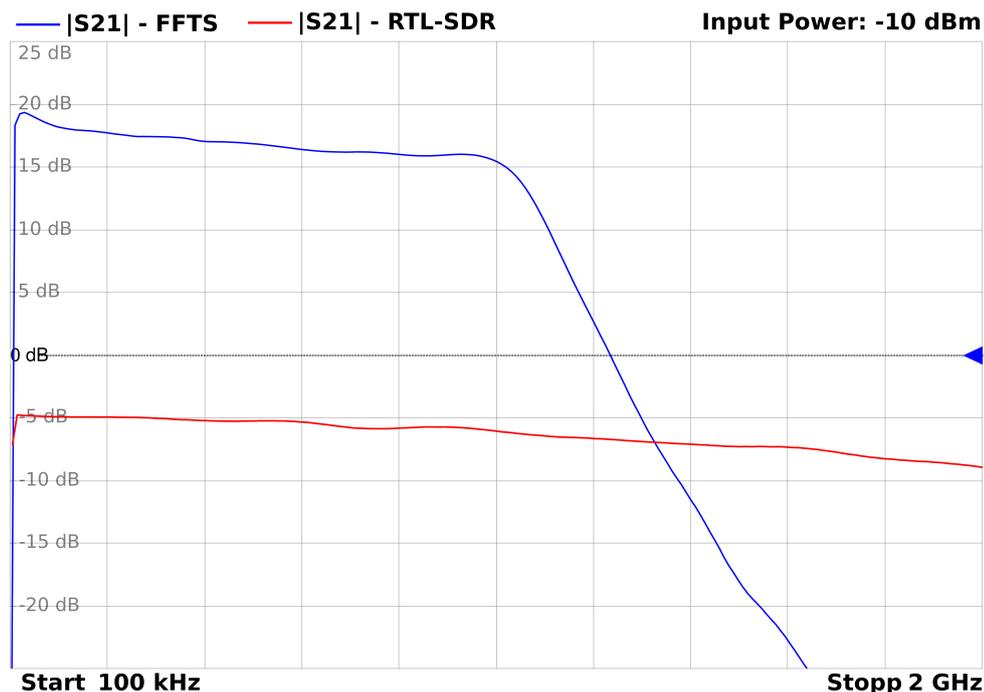


Abbildung 8.29: Vorwärtstransmission des ZF-Kanals A von der ZF-Signaleinspeisung hin zum FFTS (blau) beziehungsweise RTL-SDR (rot; Einstellung des regelbaren Dämpfungsglieds auf 0 dB, kein zusätzliches Dämpfungsglied)

8.7.3 Schutz der HF-Komponenten vor Störeinstrahlungen

Bereits zu Beginn der Arbeit wurde das Abstrahlverhalten eines einzelnen Raspberry Pi untersucht. Dabei waren starke Störungsemissionen zwischen 0 und 1 GHz festzustellen. Da diese exakt im Bereich der Nutzsignale des FFT-Spektrometers liegen, würden diese die spektrale RFI-Erkennung negativ beeinflussen. Aus diesem Grund wurden die HF-Verstärker, das Tiefpassfilter, die regelbaren Dämpfungsglieder und weitere HF-Komponenten in ein geschirmtes Gehäuse verbaut. Um ein optimales Ergebnis zu erzielen, wurden die Durchführung der Steuerleitungen der regelbaren Dämpfungsglieder durch gefilterte Steckverbinder abgeschirmt. Die Abbildung 8.30 zeigt das gebaute Gehäuse zur Abschirmung der HF-Komponenten, während in Abbildung 8.31 der komplette Einschub zu sehen ist. Dabei ist zu beachten, dass das HF-Gehäuse mit einem Deckel verschlossen ist. Um die Anschlüsse auf der Stirnseite zuordnen zu können, wurde dieser entsprechend beschriftet.

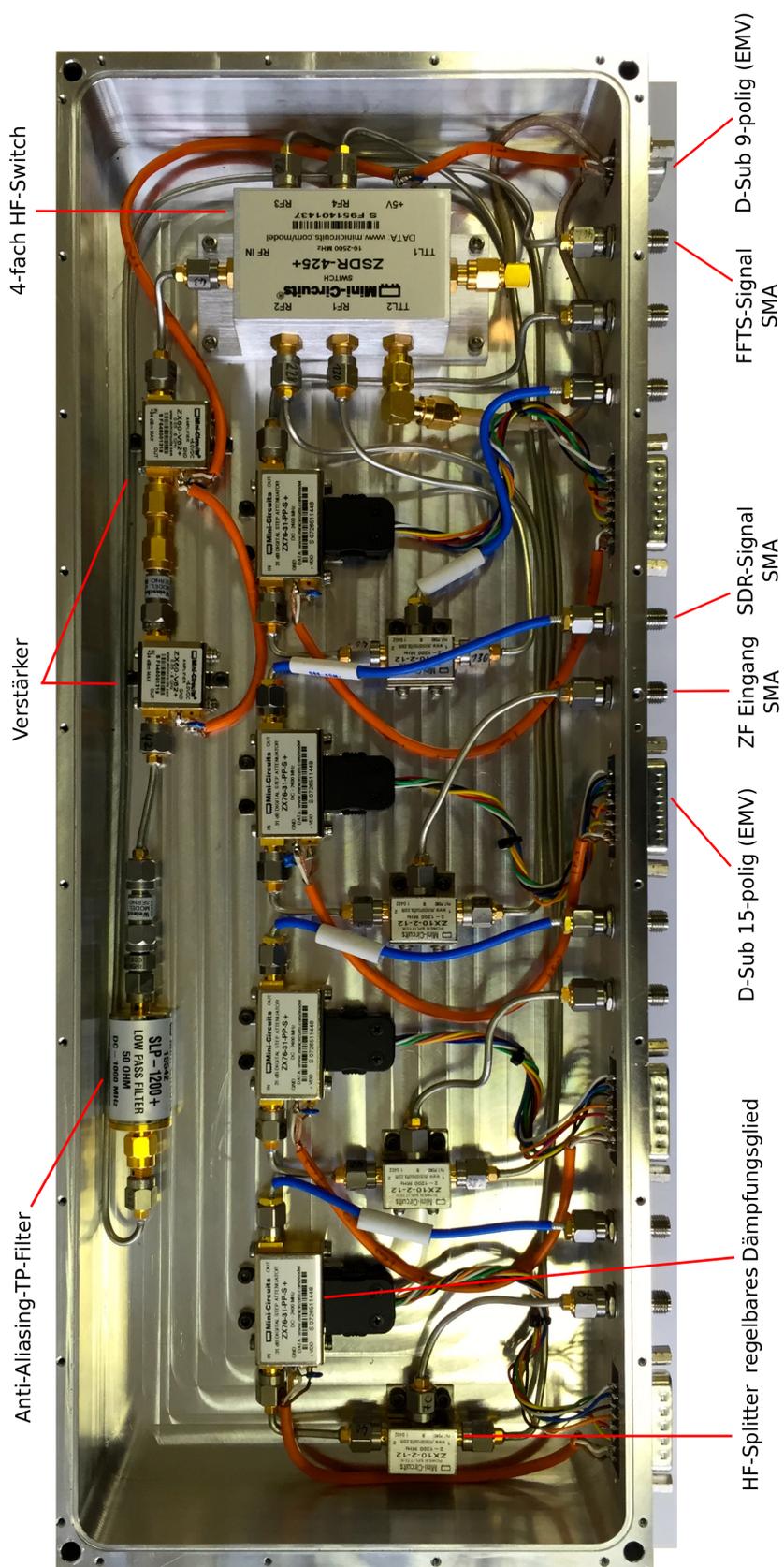


Abbildung 8.30: Abgeschirmte HF-Bauteile zum Schutz vor Störeinstrahlungen (geöffnetes Gehäuse)

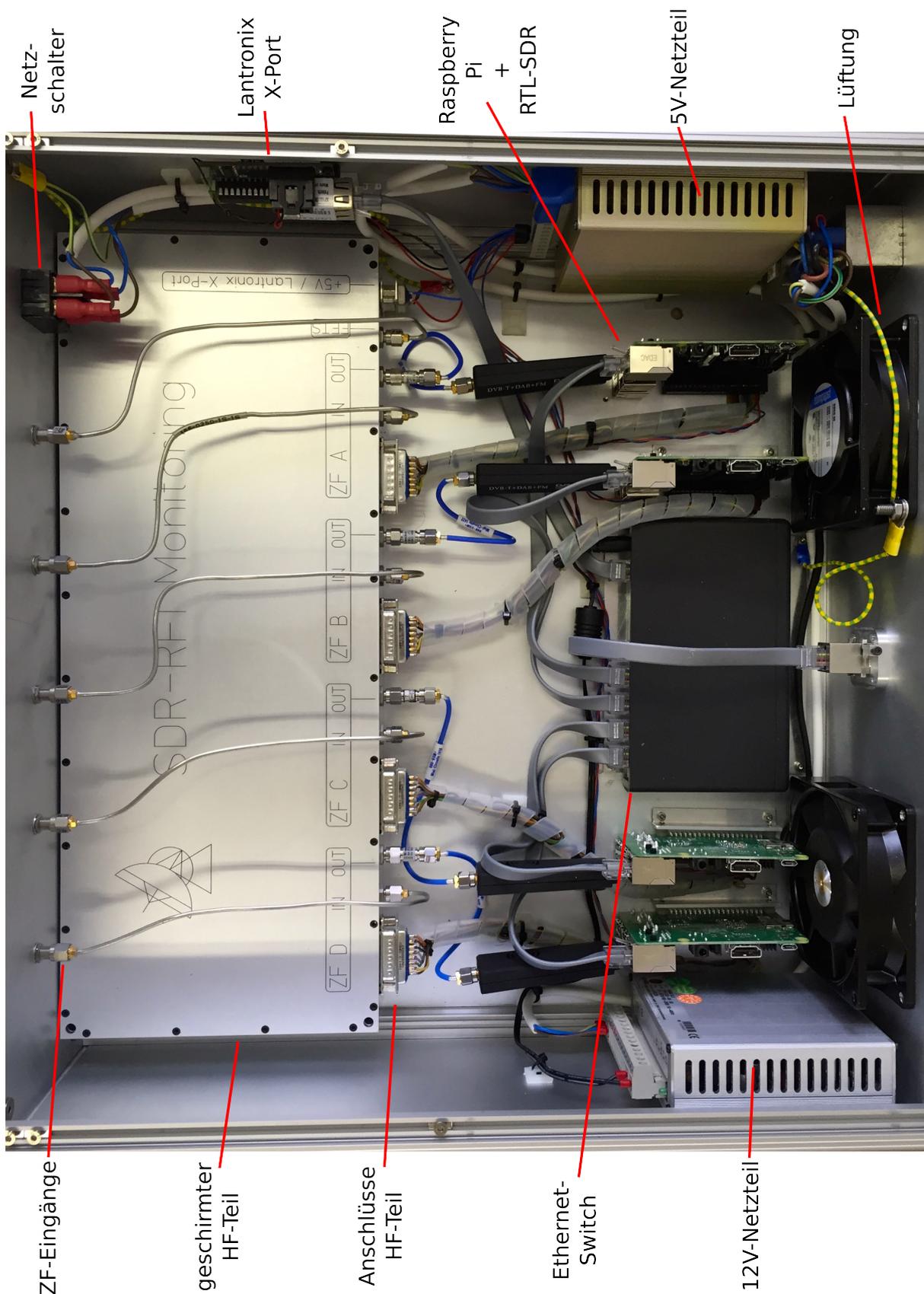


Abbildung 8.31: Einschub mit vier RTL-SDR-Servern und abgeschirmter HF-Signalverarbeitung

Die Abbildung 8.32 zeigt die Frontplatte des Einschubs. Gut erkennbar ist der Netzschalter auf der linken Seite und der Anschluss für das FFTS sowie die Eingänge der vier ZF-Signale. Zur Identifikation des Einschubs wurde dieser mit einer eindeutigen Nummer und dem Namen SDR-RFI Monitoring versehen.



Abbildung 8.32: Frontplatte des Einschubs mit Netzschalter, FFTS-Ausgang und vier ZF-Eingängen

Die Wirkung des abgeschirmten Gehäuses wird in Abbildung 8.33 deutlich, da das Betragspektrum des FFTS-Ausgangs keine Störemissionen aufweist, die durch die Raspberry Pis oder Netzteile entstanden sind. Der Schutz der HF-Komponenten vor Störeinstrahlung konnte daher erfolgreich umgesetzt werden.

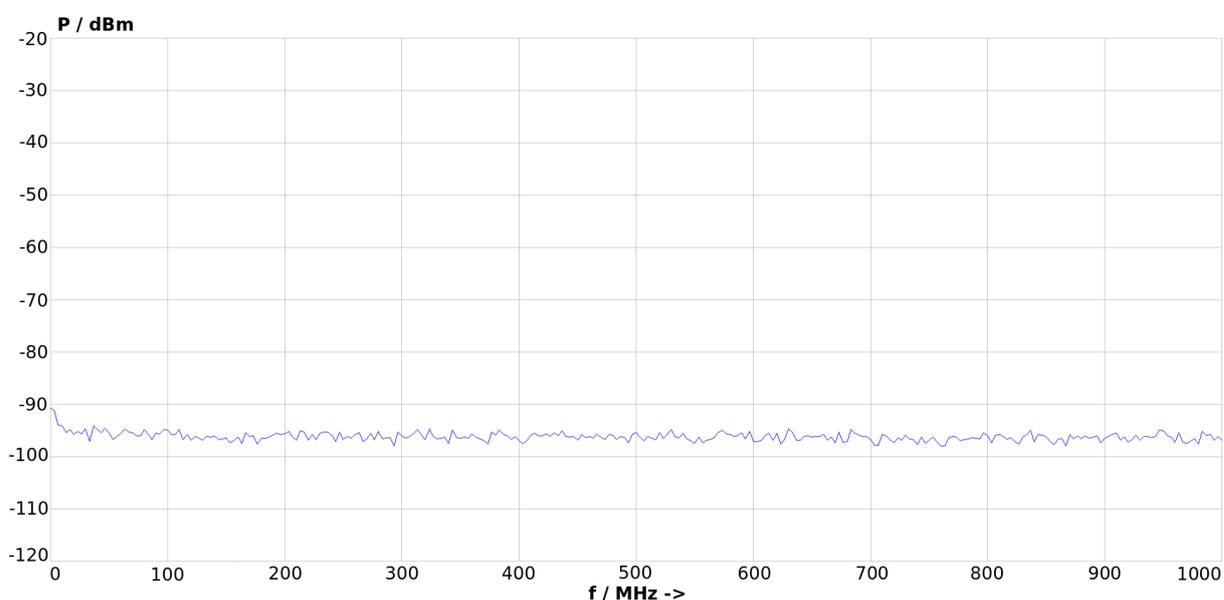


Abbildung 8.33: Störungsfreies Spektrum des Ausgangssignals des FFTS-Ausgangs (Video-Bandwidth = 30 KHz; Resolution-Bandwidth = 30 kHz)

8.8 Zusammenfassung der Arbeit mit dem FFT-Spektrometer

Die Konfiguration und die Einbindung des FFTS in die bereits implementierte Software waren zu Beginn sehr kompliziert und bereiteten einige Schwierigkeiten. Doch aufgrund des großen Know-hows der Kolleginnen und Kollegen des Instituts mit diesen Geräten konnten Lösungen für die Probleme schnell gefunden werden. Nach der erfolgreichen Integration des FFTS in die Software wurde deutlich, dass das Fourier-Betragsspektrum sehr schnell über einen breiten Frequenzbereich berechnet wird. Zur schnellen Erfassung von Störungen eignet es sich somit sehr gut.

Die Probleme der schlechten Eingangsanpassung und des Auftretens von Aliasing konnten mittels eines Tiefpasses und Dämpfungsgliedern erfolgreich gelöst werden. Die Spezifikationen des FFTS sprechen zudem für die Verwendung dieser Hardware zur spektralen RFI-Erkennung. Die entwickelte Software hat aufgrund der schnellen Berechnung des Betragsspektrums einen deutlichen Zuwachs an Performance bekommen. Die Darstellung von vier Kanälen und die Ermittlung von Störungen kürzester Zeit machen die Software im Zusammenspiel mit dem FFTS zu einem praxistauglichen Tool. Der alternative Algorithmus zur RFI-Detektion hat zudem gezeigt, dass mittels dieser Methode zuverlässig selbst kleinste Störsignale erfasst werden können. Das Problem, ob es sich bei einem detektierten Störer tatsächlich um RFI handelt bleibt aber weiterhin bestehen.

Im Zusammenspiel mit einem Software Defined Radio und der Demodulation der Signale kann diese Software bei der eindeutigen Detektion von Störquellen helfen. Das automatische Anpassen der Frequenz des SDR bei der Auswahl eines Störsignals und die Darstellung der Frequenznutzung erleichtern die Schlussfolgerung auf eine eventuelle Quelle des Signals. Die Wiedergabe der Signale durch eine FM- oder AM-Demodulation kann zur Identifikation eines Signals beitragen. Hierbei ist zu beachten, dass nur analoge Modulationsverfahren betrachtet wurden. Signale, die digitale Modulationsverfahren nutzen, sind meist so komplex, dass eine Demodulation nur unter der genauen Kenntnis der notwendigen Modulationsparameter möglich ist. Daher ist diese Art der Demodulation sehr umfangreich und wird in dieser Arbeit nicht weiter verfolgt. Zudem müssten entsprechende Interfaces zur Verfügung dargestellt werden, um zum Beispiel Videosignale darzustellen oder Datenpakete anzuzeigen.

Das nächste Kapitel befasst sich mit der Darstellung der Ergebnisse der spektralen RFI-Detektion über ein Web-Interface. Dabei werden die Plots der Spektren der ZF-Kanäle mit detektierten Störungen, die durch das FFTS erfasst wurden, dargestellt.

9 Entwicklung eines webbasierten Benutzerinterfaces

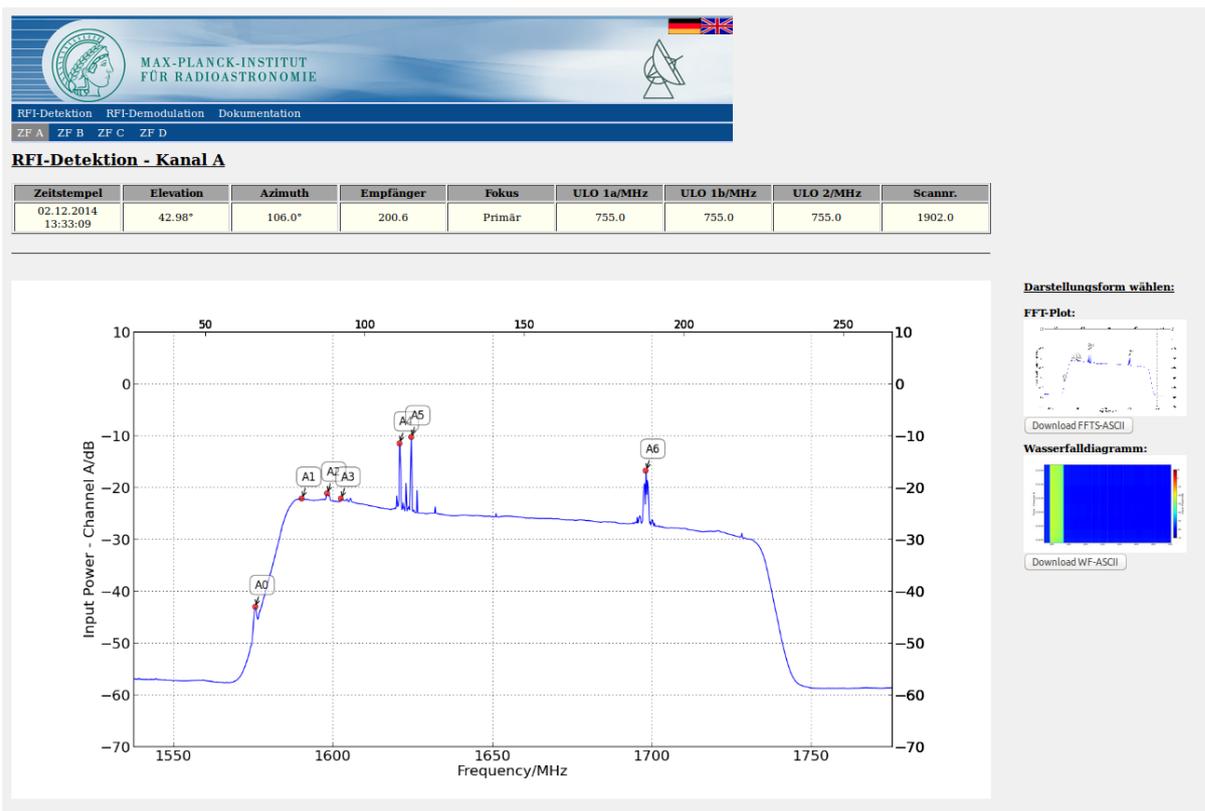
Die Detektion von RFI mittels eines FFT-Spektrometers und die Demodulation von Störungen durch ein Software Defined Radio werden in diesem Kapitel in einem Benutzerinterface dargestellt, welches über einen Webbrowser aufrufbar ist. Die Anforderungen der Aufgabenstellungen verlangen dabei, dass eine Steuerung der SDRs über dieses Webinterface implementiert wird. Dazu zählt die Einstellung verschiedener Parameter, wie der Verstärkung oder des Modulationsverfahrens. Die Entwicklung des Benutzerinterface in den Sprachen HTML, PHP und JavaScript wird in diesem Kapitel dargestellt.

9.1 Spektrale RFI-Erkennung im Webinterface

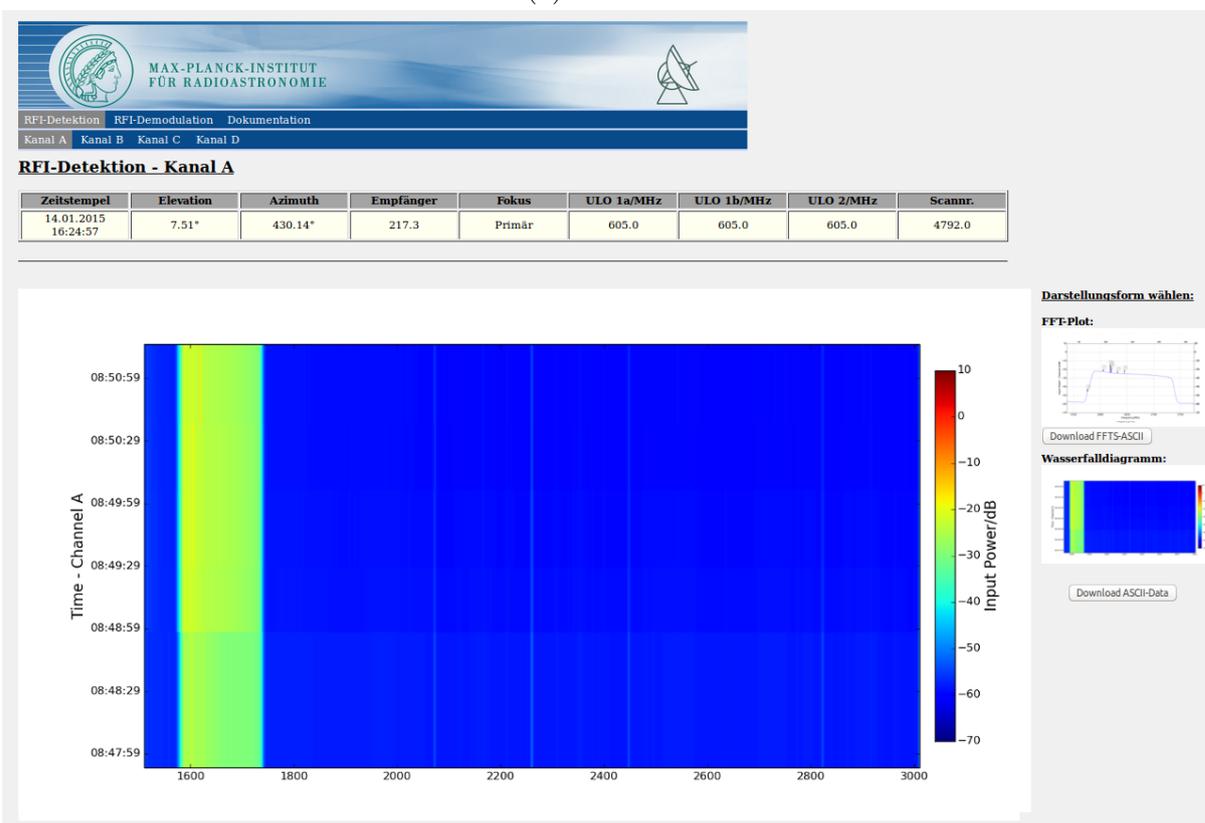
Die Verwendung eines Webinterfaces schränkt die Darstellung in einem gewissen Rahmen ein, da die Leistungsfähigkeit eines solchen Interface nicht mit einer in Python programmierten QT-Oberfläche mithalten kann. Der eindeutige Vorteil liegt jedoch im Mehrfachzugriff auf die Daten durch mehrere Benutzer zur gleichen Zeit. Zudem ist diese Art der Darstellung universell, da für den Aufruf nur ein Webbrowser benötigt wird. Ziel ist es, das Interface so benutzerfreundlich und so schnell wie möglich zu gestalten. Zunächst wird die Umsetzung der spektralen RFI-Detektion im Webinterface betrachtet.

Dazu wird die QT-Oberfläche der zuvor geschriebenen Python-Programme aus der Software entfernt. Statt der Darstellung über ein QT-GUI wird das geplottete Ergebnis der RFI-Detektion als Bild gesichert. Die detektierten Störungen erhalten einen Marker, um später der Störung den passenden Eintrag des Frequenznutzungsplans zuzuordnen. Neben der Darstellung als FFT-Plot wurde zudem eine Visualisierung in Form eines Wasserfall-Diagramms implementiert. Das Ergebnis wird ebenso als Bild gesichert.

Auf der Seite für die spektrale RFI-Detektion werden die gesicherten Plots dargestellt und mittels Java-Skript ständig aktualisiert, ohne dass die Website neu geladen werden muss. Der Benutzer kann einfach zwischen beiden Darstellungsformen hin und her schalten. Oberhalb des Plots werden die aktuellen Informationen aus dem Multicast angezeigt. Unterhalb werden die aus dem Frequenznutzungsplan extrahierten Daten zu einer Störung angezeigt. Die zuvor definierten Marker erleichtern dabei die exakte Zuordnung. Die Abbildung 9.1 zeigt einen Bildschirmausdruck dieser Seite.



(a) FFT-Plot



(b) Wasserfall-Plot

Abbildung 9.1: Screenshots des Webinterfaces zur Darstellung der spektralen Erkennung von RFI

9.2 Demodulation von Störstrahlern und Steuerung der SDR-Hardware

Um eine detektierte Störung zu demodulieren, bietet die Darstellung der Frequenznutzung die Möglichkeit, mittels des "Demodulieren"-Buttons das Software Defined Radio auf die passenden Einstellungen zu setzen (siehe Abbildung 9.2). Sprich, der entsprechende Kanal wird ausgewählt und die Mittenfrequenz des SDR wird auf den korrekten Wert der Zwischenfrequenz eingestellt.

Frequenznutzung detektiierter Störungen - Kanal A															
Marker	Demodulation	Eintrag	RFI-Frequenz/MHz	Start-/Stoppfrequenz/MHz	Service	Nutzung	Frequenznutzung	TB-Start-Stopp	Kanal-BW	Kanalraster	Duplexabstand	Duplex-Start-Stopp	Nutzungsdauer	Leistung	Frequenznutzungsbedingung
A0	Demodulieren	859	1575.55175781	1559-1610 MHz	FESTER FUNKDIENST D362B	mil	Funknwendungen der BOS	1590-1610 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	Richtfunkanlagen fuer Funknwendungen der BOS
A0	Demodulieren	860	1575.55175781	1559-1610 MHz	Welltraum) D329A	mil	Militaerische Funknwendungen	1559-1610 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	k.A.
A1	Demodulieren	859	1590.74951172	1559-1610 MHz	FESTER FUNKDIENST D362B	mil	Funknwendungen der BOS	1590-1610 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	Richtfunkanlagen fuer Funknwendungen der BOS
A1	Demodulieren	860	1590.74951172	1559-1610 MHz	Welltraum) D329A	mil	Militaerische Funknwendungen	1559-1610 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	k.A.
A2	Demodulieren	859	1598.07373047	1559-1610 MHz	FESTER FUNKDIENST D362B	mil	Funknwendungen der BOS	1590-1610 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	Richtfunkanlagen fuer Funknwendungen der BOS
A2	Demodulieren	860	1598.07373047	1559-1610 MHz	Welltraum) D329A	mil	Militaerische Funknwendungen	1559-1610 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	k.A.
A3	Demodulieren	859	1605.39794922	1559-1610 MHz	FESTER FUNKDIENST D362B	mil	Funknwendungen der BOS	1590-1610 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	Richtfunkanlagen fuer Funknwendungen der BOS
A3	Demodulieren	860	1605.39794922	1559-1610 MHz	Welltraum) D329A	mil	Militaerische Funknwendungen	1559-1610 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	k.A.
A4	Demodulieren	898	1698.04931641	1690-1700 MHz	WETTERFUNKDIENST UEBER SATELLITEN (Richtung W)	ziv	Meteorologischer Satellitennetz	1690-1698 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	Nutzerstationen zum direkten Datenempfang von
A4	Demodulieren	899	1698.04931641	1690-1700 MHz	WETTERHILFENFUNKDIENST 3	ziv	3 Militaerische Funknwendungen	1690-1700 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	Einzelfrequenzen fuer militaerische Nutzungen
A4	Demodulieren	900	1698.04931641	1690-1700 MHz	Fester Funkdienst 25	ziv	Militaerische Funknwendungen	1690-1700 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	Einzelfrequenzen fuer militaerische Nutzungen
A4	Demodulieren	901	1698.04931641	1690-1700 MHz	WETTERFUNKDIENST UEBER SATELLITEN (Richtung W)	ziv	Meteorologischer Satellitennetz	1698-1700 MHz	k.A. k.A.	k.A. k.A.	k.A. k.A.	k.A.-k.A. k.A.	k.A.	k.A.k.A.-k.A.	Nutzerstationen zum direkten Datenempfang von

Letzte Aktualisierung: 02.12.2014 - 13:35 Uhr

Abbildung 9.2: Screenshot des Webinterfaces zur Zuordnung der Frequenznutzung einer Störung

Man gelangt zur Seite zur RFI-Demodulation (siehe Abbildung 9.3). Auf dieser Seite bekommt der Nutzer neben einer Anzeige mit den aktuellen Einstellungen des SDR auch die Möglichkeit, die Parameter mittels Schieberegler selbst zu beeinflussen, um beispielsweise die Mittenfrequenz an die Störung anzupassen. Eine manuelle Einstellung entfällt jedoch, wenn über den Button "Demodulieren" auf der Seite zur spektralen RFI-Detektion der Seitenwechsel durchgeführt wurde. Neben den Einstellmöglichkeiten wird das aktuelle Spektrum am Eingang des SDR angezeigt, sowie das demodulierte Zeitsignal. Der Plot aktualisiert sich eigenständig alle jede Sekunde.

Zur Audiowiedergabe der demodulierten Zeitsignale wurde ein VLC-Client eingebunden. Die Wiedergabe startet dabei automatisch und so dem Anwender wird es möglich, dem zeitlichen Verlauf eines Signals zu folgen. Um die Demodulation der Signale und die korrekte Audiowiedergabe zu überprüfen, wurde ein UKW-Radiosignal eingespeist, demoduliert und über die Website erfolgreich wiedergegeben.

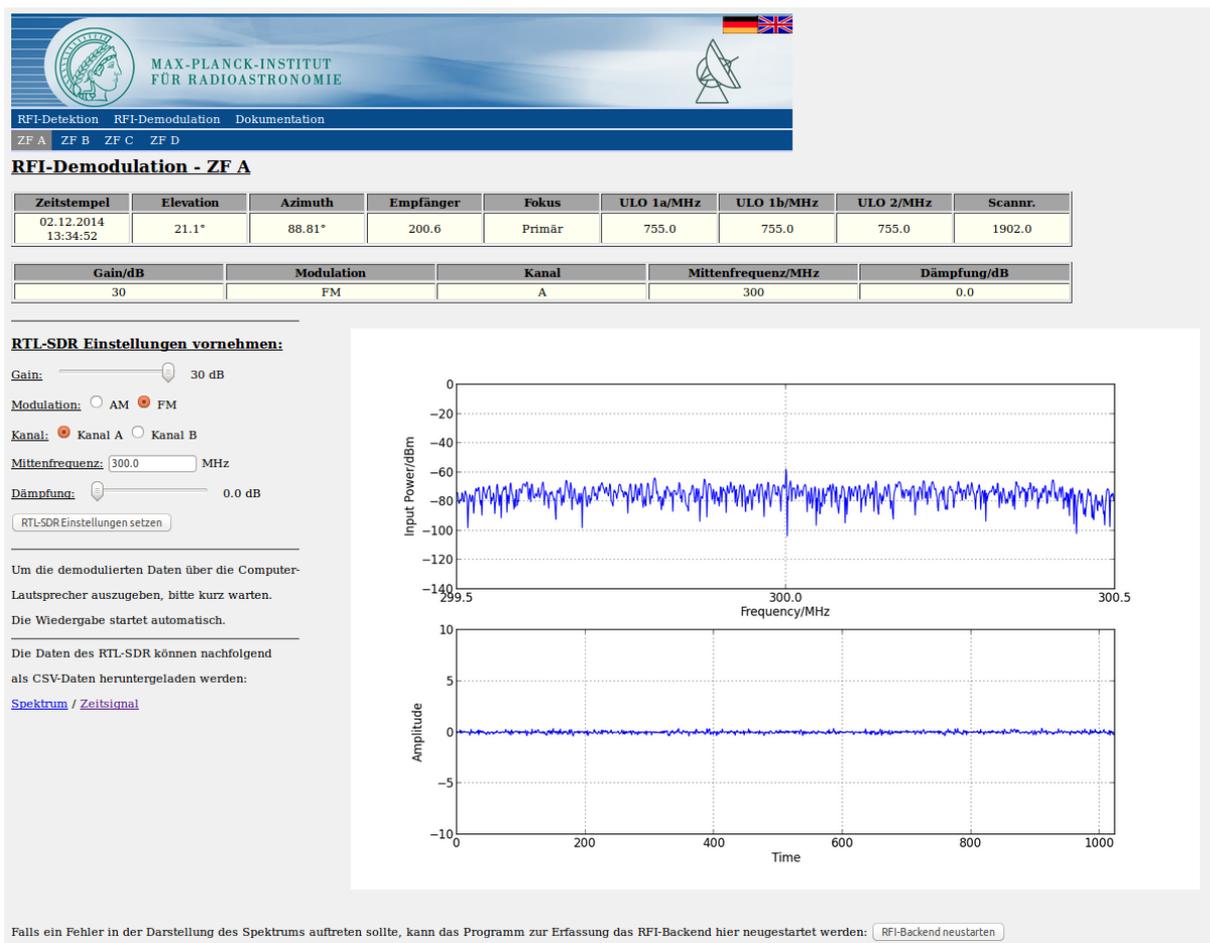


Abbildung 9.3: Screenshot des Webinterfaces zur Demodulation von RFI-Signalen und zur Einstellung eines SDR

Beide Websites bieten eine Möglichkeit, die im Hintergrund laufenden Python-Programme neu zu starten, falls ein System hängt oder fehlerhafte Ausgaben erzeugt. Dabei werden die Programme auf dem Server sauber beendet und komplett neugestartet. Zudem wurde eine Dokumentation der Python-Programme erstellt und über das Webinterface zugänglich gemacht. Damit werden die Wartung des Systems und das Verständnis über die genaue Funktion der Programme erleichtert. Die Abbildung 9.4 zeigt schematisch den Zusammenhang zwischen allen Programmen die auf dem Server, dem Webserver und einem exemplarischen Raspberry Pi laufen.

Ein Problem, das sich bei der Portierung des entwickelten Software auf einen der Instituts-Server ergab, war, dass die Performance der entwickelten Software aufgrund des sehr hohen Rechenaufwands auf diesen deutlich abgenommen hat. Die Laufzeit des Programms zur spektralen RFI-Detektion durch das FFTS hat sich dadurch merklich erhöht. Jedoch liegt diese weiterhin in einem akzeptablem Rahmen. Das hohe Datenaufkommen der durch die RTL-SDR-Server übermittelten Daten führte zu weiteren Problemen bei der Signaldemodulation. Da die auf den Raspberry Pi's laufende Software zur Übermittlung der IQ-Samples über TCP einen eingebauten Puffer hat, kommt es zu einer Verzögerung

der Darstellung des Spektrums von etwa 10 Sekunden, da der Puffer durch die Software nicht schnell genug geleert wird. So kann es dazu kommen, dass Störungen erst später im Spektrum der SDRs angezeigt werden als in der Darstellung durch das FFTS. Auch die Audiowiedergabe und Veränderungen der Einstellungen werden um entsprechend diesen Zeitraum verzögert. Dieser Nachteil stört jedoch nicht erheblich bei der praktischen Arbeit mit dieser Software und wird daher als eine Einschränkung akzeptiert.

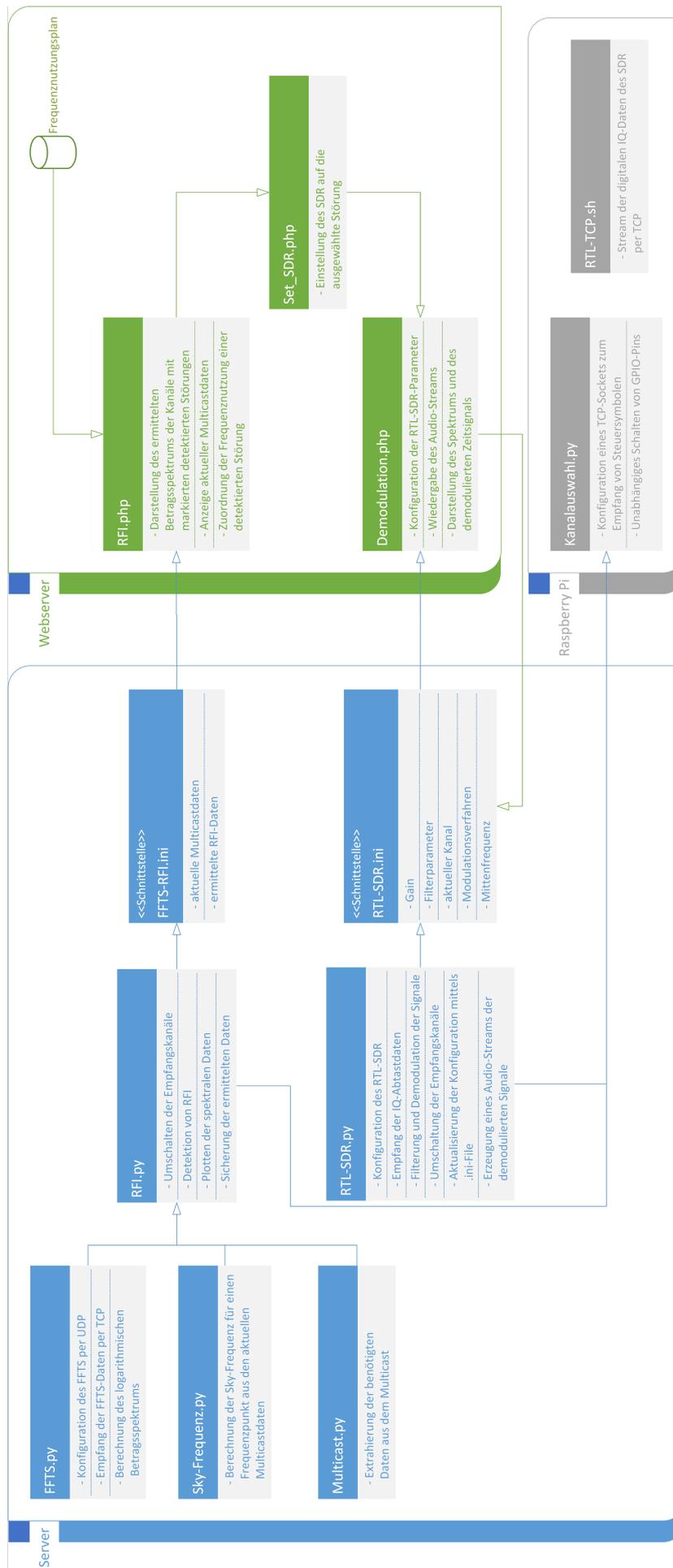


Abbildung 9.4: Softwarestruktur zur Datenerfassung und Steuerung der Hardware

10 Fazit und Ausblick

Dieses abschließende Kapitel fasst die Ergebnisse dieser Arbeit zusammen. Dabei werden die Probleme die während dieser Arbeit auftraten diskutiert und die gefundenen Lösungen zusammenfassend dargestellt. Im Anschluss an das Fazit folgt ein Ausblick, welcher sich mit weiterführenden Überlegungen in Bezug auf offengebliebene Fragen und Erweiterungen des Systems befasst.

10.1 Fazit

Die ersten Kapitel dieser Arbeit haben die große Bedeutung von Radio Frequency Interference für die Arbeit der Astronomen des Max-Planck-Instituts für Radioastronomie aufgezeigt. Auf Grund der immer weiter wachsenden Anzahl an Funkdiensten, die die Messungen mit dem Radioteleskop Effelsberg beeinträchtigen können, ist eine Erkennung dieser Störer für das Institut sehr wichtig. Diese Detektion sollte mit einem Software Defined Radio durchgeführt werden. Doch inwiefern eignen sich COTS-Software Defined Radios für diese Anwendung?

Ziel der Arbeit war es, aus verschiedenen SDR-Hardwareinterfaces das passende für eine RFI-Detektion zu finden und entsprechende Software zu entwickeln. Die Definition des Begriffs Software Defined Radio durch die ITU-R wurde erörtert und der Grund, warum diese Hardware eine zunehmende Rolle in der modernen Funktechnik spielt. Zudem wurde der Begriff der RFI für Astronomen definiert und Merkmale herausgearbeitet. Zu Beginn der Untersuchungen wurde ein handelsüblicher DVB-T-Stick zum Fernsehempfang am Computer verwendet. Nach der Darstellung, warum ein solcher Stick als SDR-Hardware verwendbar ist, wurden zwei Sticks mit jeweils unterschiedlichem Tuner messtechnisch untersucht. Die Ergebnisse haben gegen die Verwendung einer solchen Hardware gesprochen, da die Intermodulationsfestigkeit, der Dynamikumfang sowie die Bandbreite unzureichend für diese Anwendung sind. Die für diese Hardware entwickelte Software hat weitere schwerwiegende Probleme, wie starke Eigenstörlinien über einen breiten Frequenzbereich, hervorgebracht. Eine spektrale Detektion mit einem DVB-T-Stick ist daher nicht möglich.

Um die Nachteile dieser Hardware zu beseitigen, wurde testweise ein USRP von National Instruments eingesetzt. Die Arbeit mit diesem Gerät hat viele der Probleme mit einem DVB-T-Stick als SDR behoben, jedoch auch weitere hervorgebracht. So hatte das USRP zwar den Vorteil zwei Kanäle zu haben, jedoch war das Übersprechen zwischen diesen beiden so stark, dass dieser Vorteil nicht praktisch genutzt werden konnte. Insgesamt waren die Nachteile zu gravierend, um dieses für diese Anwendung zu nutzen.

Im Anschluss an die Arbeit mit einem USRP wurde ein FFT-Spektrometer eingesetzt. Dieses ermöglicht die Aufnahme des Betragsspektrums über eine hohe Bandbreite in sehr kurzen Abständen. Das geringe Eigenrauschen macht es möglich, das so berechnete Betragsspektrum zur spektralen RFI-Detektion zu verwenden. Weitere messtechnische Untersuchungen haben die Eignung dieser Hardware bestätigt. Für die Aufnahme von zwei Kanälen wurde ein Umschalter gebaut, welcher über einen Raspberry Pi gesteuert wird und abwechselnd beide Kanäle auf den Eingang des FFT-Spektrometers schaltet.

Eine weitere Aufgabe dieser Arbeit war die Demodulation von Signalen. Da sich das USRP sowie die DVB-T-Sticks als ungeeignet für die spektrale Signalanalyse erwiesen haben, wurde ein DVB-T-Stick für die Demodulation von Signalen eingesetzt. Dabei wurden die zwei häufigsten analogen Modulationsverfahren umgesetzt, nämlich AM und FM, um detektierte Störsignale zu demodulieren. Trotz der schlechten Eigenschaften der Hardware konnten diese erfolgreich für die Demodulation von Signalen eingesetzt werden.

Im nachfolgenden Abschnitt wurde eine zusätzliche Erweiterung des Systems um zwei weitere ZF-Signale dargestellt. Dabei werden insgesamt vier Raspberry Pis eingesetzt, um für jeden ZF-Kanal eine separate Signaldemodulation durchzuführen. Das FFTS nimmt ebenso vier statt zwei Kanäle auf. Die zusätzlichen Kanäle dienen vorerst nur als Reserve und werden in naher Zukunft mit Signalen belegt. Die Software wurde dementsprechend erweitert und angepasst. Das Prinzip des Systems ist jedoch unverändert geblieben.

Ein weiteres Ziel war es, die durch die oben beschriebenen Hardwarekomponenten erzielten Ergebnisse benutzerfreundlich in einem Webinterface darzustellen. Dazu wird das Ergebnis der spektralen RFI-Detektion auf einer Website dargestellt. Über die zugeordnete Frequenznutzung aus dem Frequenznutzungsplan der Bundesnetzagentur ist eine direkte Zuordnung einer Störung zu einem Dienst möglich. Weiterhin kann auf Knopfdruck das für den Kanal entsprechende SDR auf die passenden Einstellungen eines detektierten Störsignals gesetzt werden. Dabei wird das demodulierte Zeitsignal dargestellt und akustisch über die Computerlautsprecher wiedergegeben. Das aktuell laufende System umfasst dabei etwa 17000 Zeilen Programmcode in den Sprachen Python, PHP, HTML und JavaScript.

Mit den so erzielten Ergebnissen wurde ein einfach zu bedienendes Tool zur Überwachung und Detektion von Störsignalen gestaltet. Die Verwendung und die Untersuchung verschiedener Hardwarekomponenten hat viel zu der erfolgreichen Entwicklung des Projektes beigetragen, da so der Sinn für die wichtigen Aspekte geschärft wurde. Der Einsatz des fertigen Systems wurde am Radioteleskop Effelsberg erfolgreich erprobt.

10.2 Ausblick

Die Demodulation von Signalen wurde in dieser Arbeit auf analog modulierte Signale beschränkt. Eine weitere denkbare Aufgabe wäre es, das System um die Demodulation von digital modulierten Daten zu erweitern. Da diese aber meist sehr komplex sind, ist der Aufwand für die Implementierung sehr umfangreich.

Ein weiterer Punkt ist die Ablage von detektierten Störungen in einer Datenbank. Dies würde die nachträgliche Verfolgung des Auftretens einer Störung ermöglichen. Ein entsprechendes Interface müsste in das Webinterface integriert werden.

Zuletzt bleibt die Frage offen, die sich durch die komplette Arbeit gezogen hat. Wann ist eine detektierte Störung wirklich eine durch eine fremde Quelle erzeugte Störung? Denn mit den entwickelten Algorithmen wurde beispielsweise die Radioemission der Milchstraße als Störung erkannt. Per Definition ist dies aber kein RFI. Daher wäre es denkbar, das System mit einer solchen Intelligenz auszustatten, dass dieses tatsächlich erkennt, ob es sich um ein echtes RFI-Signal handelt. Zudem wäre eine automatische Erkennung des Modulationsverfahrens eines Signals denkbar, sodass eine manuelle Einstellung zur Demodulation entfällt. Das ideale Ergebnis dieser Anwendung würde in einer adaptiven Filterung der erkannten Störsignale aus den ZF-Signalen liegen, um deren Einfluss auf Messungen zu eliminieren (siehe Abbildung 10.1).

Das System wurde erfolgreich getestet und erfüllt alle gestellten Anforderungen. Jedoch sind, wie oben ausgeführt, viele Erweiterungen denkbar und einige offengebliebene Fragen zu beantworten. Es hat sich gezeigt, dass das Thema RFI nicht nur von großer Bedeutung, sondern auch äußerst komplex ist.

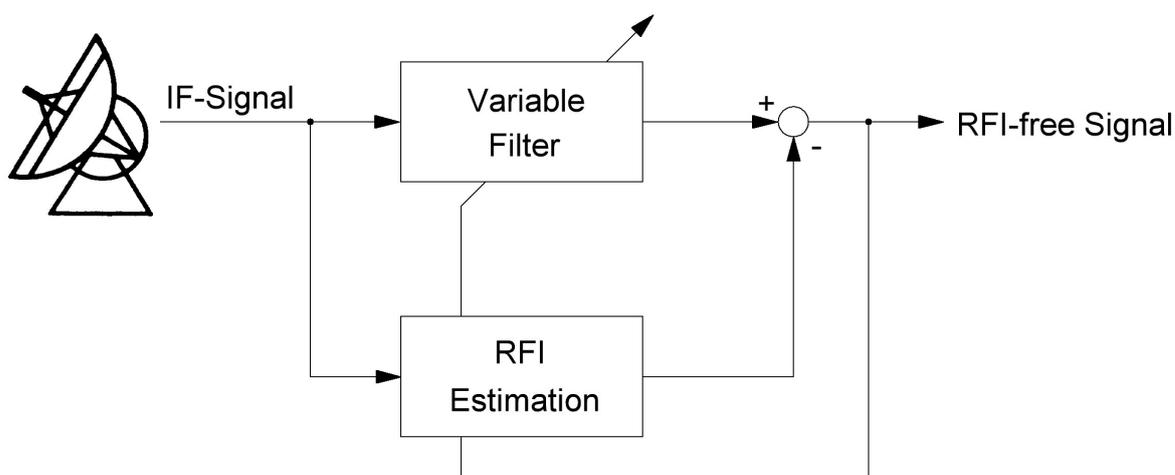


Abbildung 10.1: Idealer Signalfluss zur adaptiven Filterung von erkannten Störsignalen

Anhang

A.1 Weitere Test-Messungen mit einem USRP

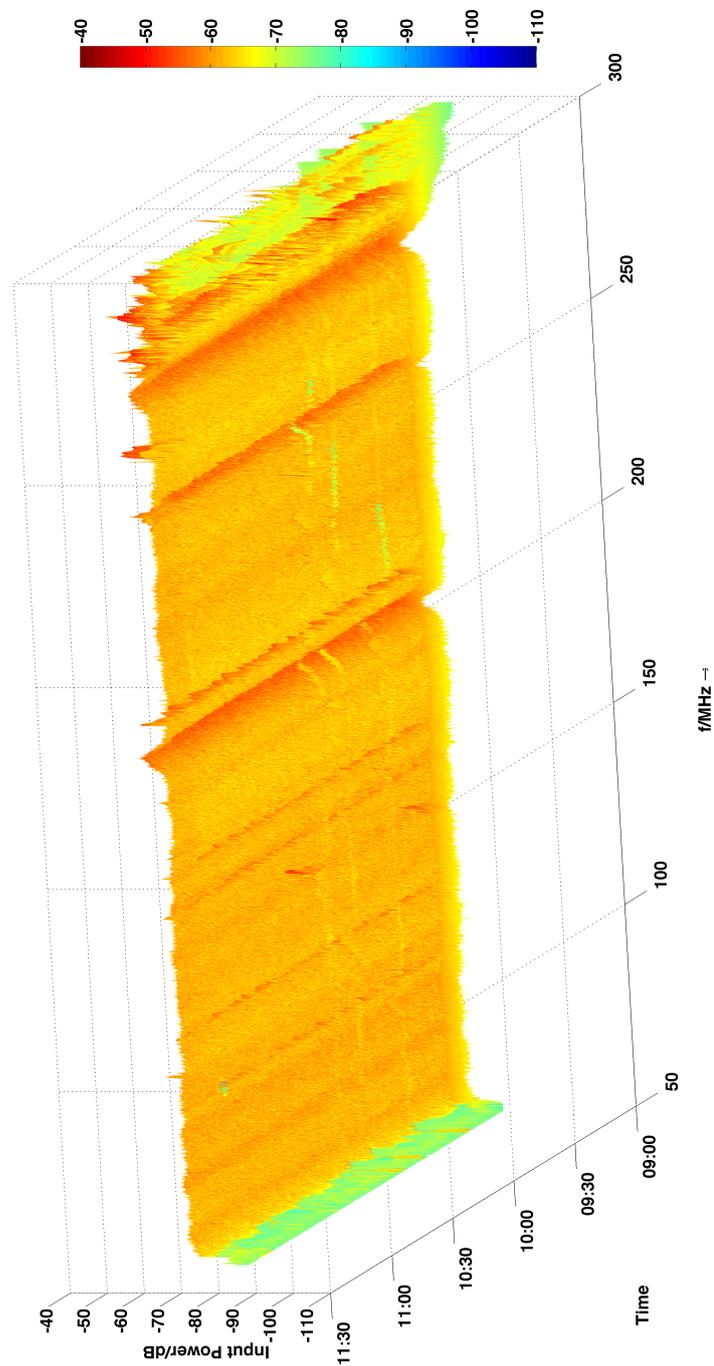


Abbildung A.1: Aufnahme des Spektrums des ZF-Signals des 21 cm - 7 Beam-Empfängers über mehrere Stunden

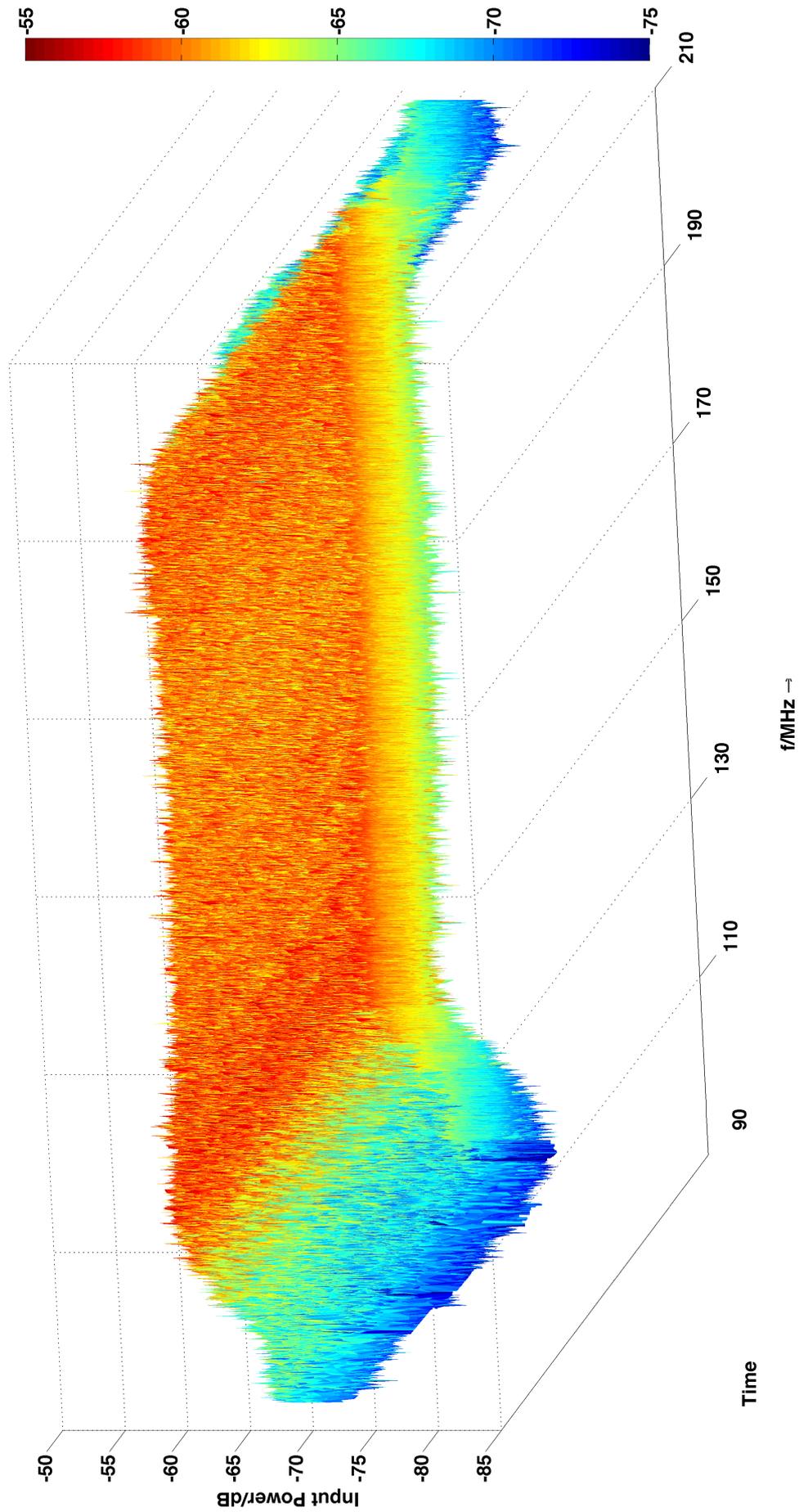


Abbildung A.2: Aufnahme des Spektrums des ZF-Signals des 11-cm-Sekundärfokus-Empfängers über mehrere Stunden

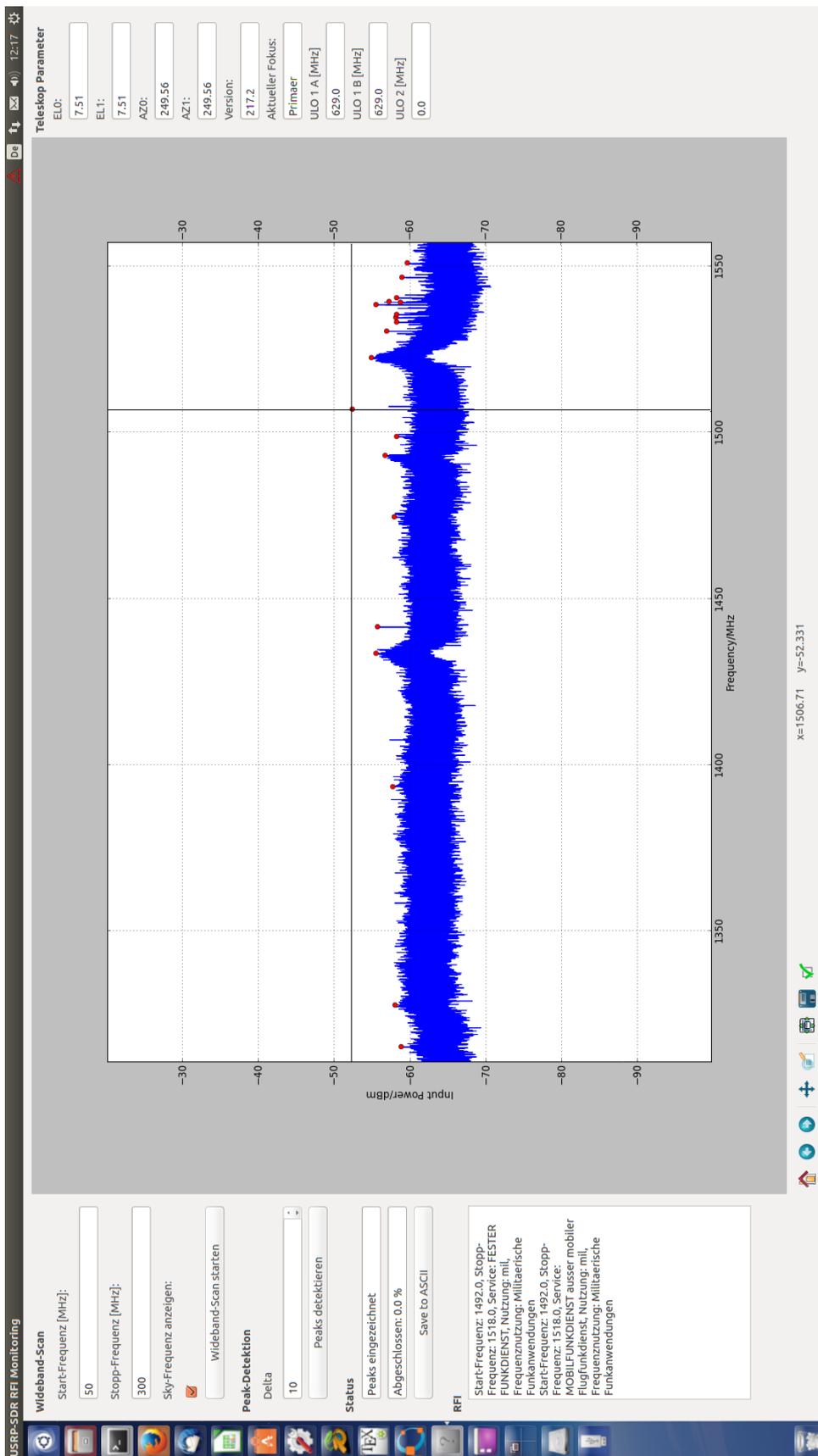
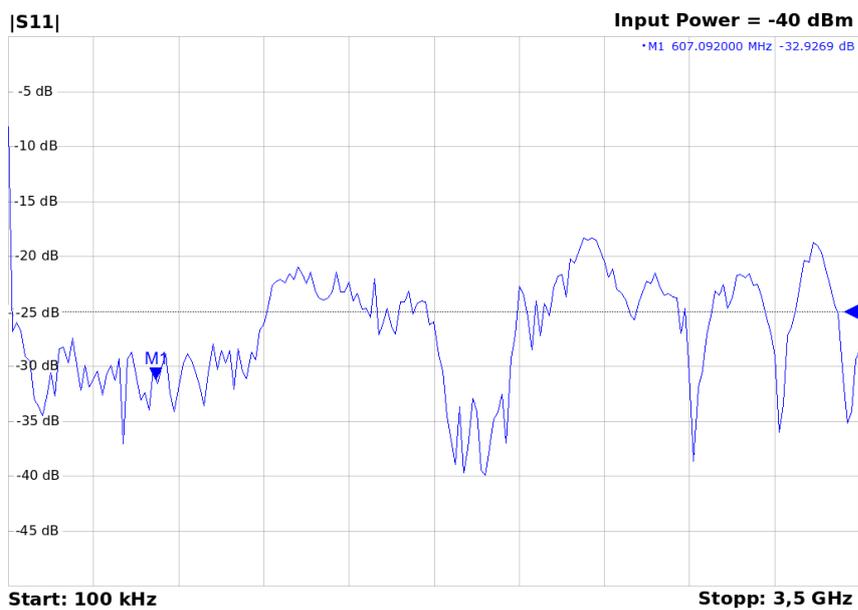
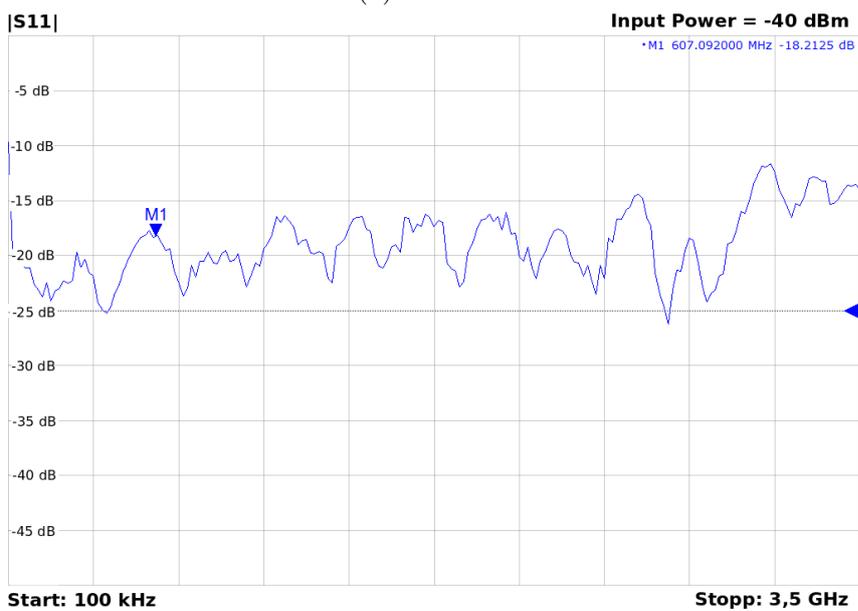


Abbildung A.3: Spektrum des ZF-Signals des 21 cm - 7 Beam-Empfängers mit detektierten Peaks in der grafischen QT-Oberfläche

A.2 Streuparameter des Zweifach-Kanalumschalters

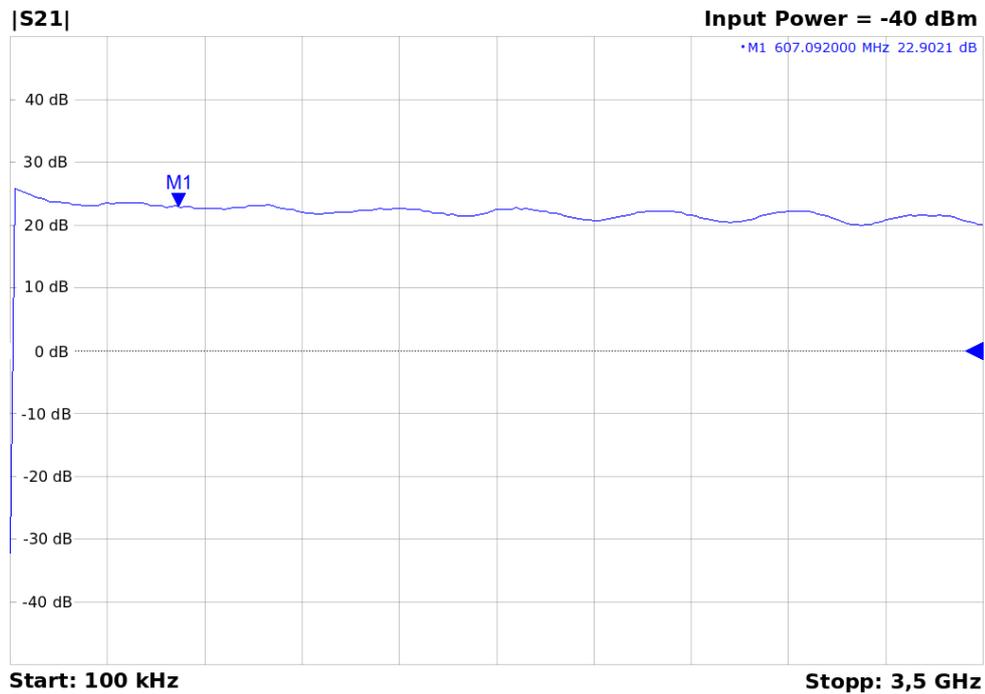


(a) Kanal A

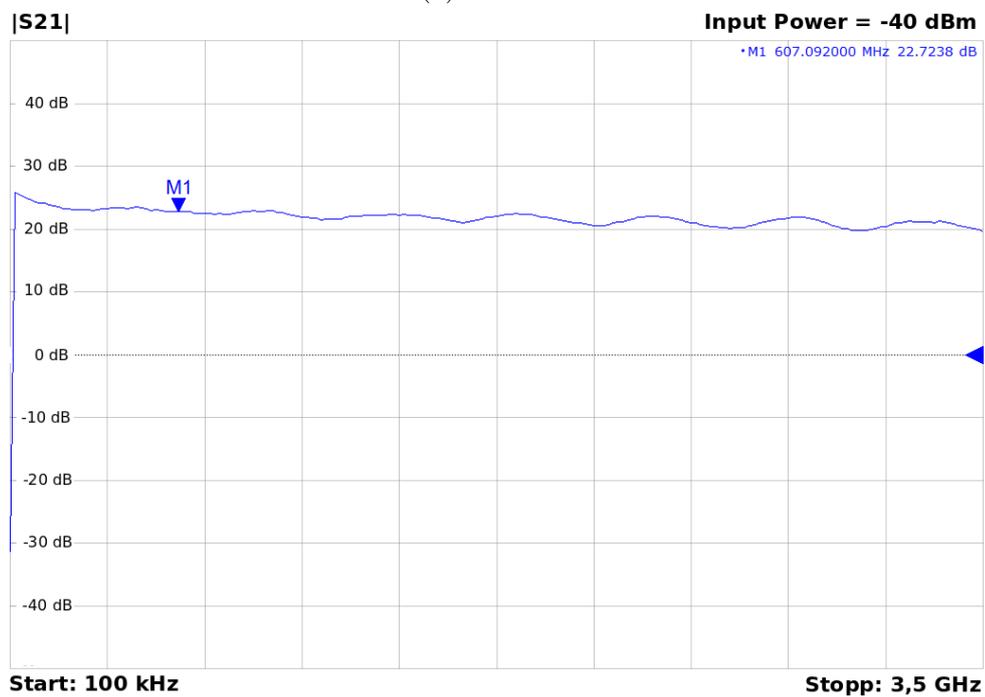


(b) Kanal B

Abbildung A.4: Eingangreflexionsfaktor $|S_{11}|$

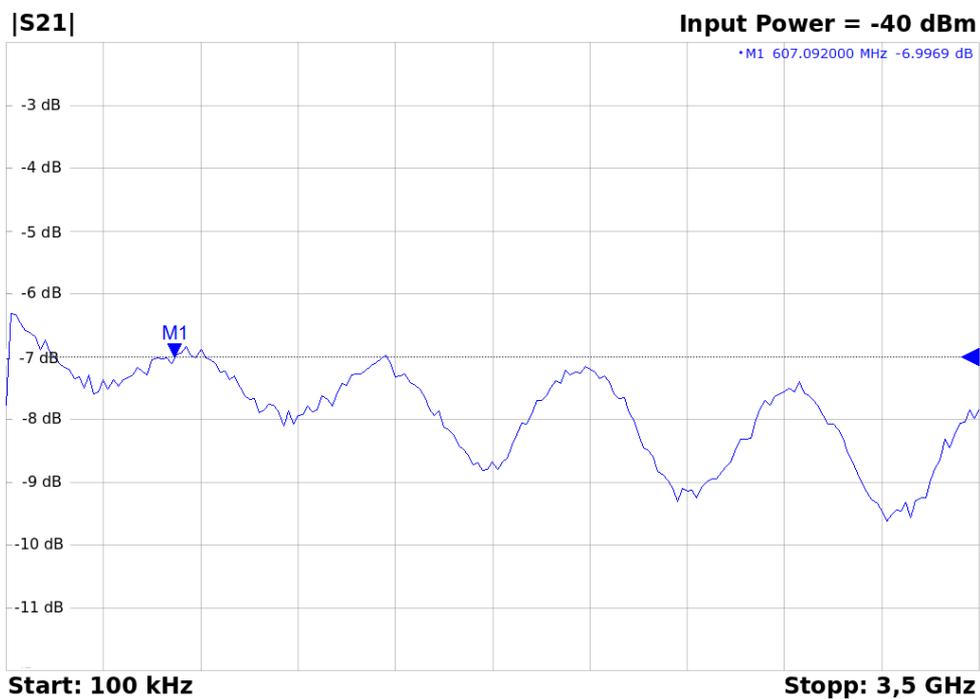


(a) Kanal A

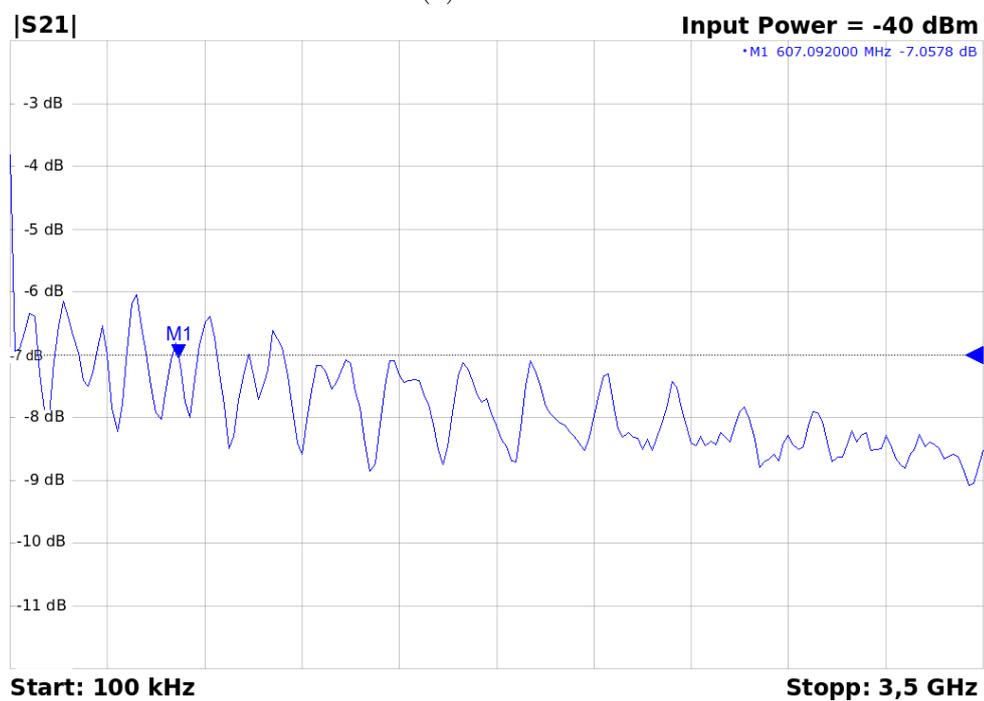


(b) Kanal B

Abbildung A.5: Vorwärtstransmission $|S_{21}|$ der FFTS-Kanäle



(a) Kanal A



(b) Kanal B

Abbildung A.6: Vorwärtstransmission $|S_{21}|$ der SDR-Kanäle

A.3 Weitere Beispiele detektierter Störsignale

A.3.1 GSM-Mobilfunksignale

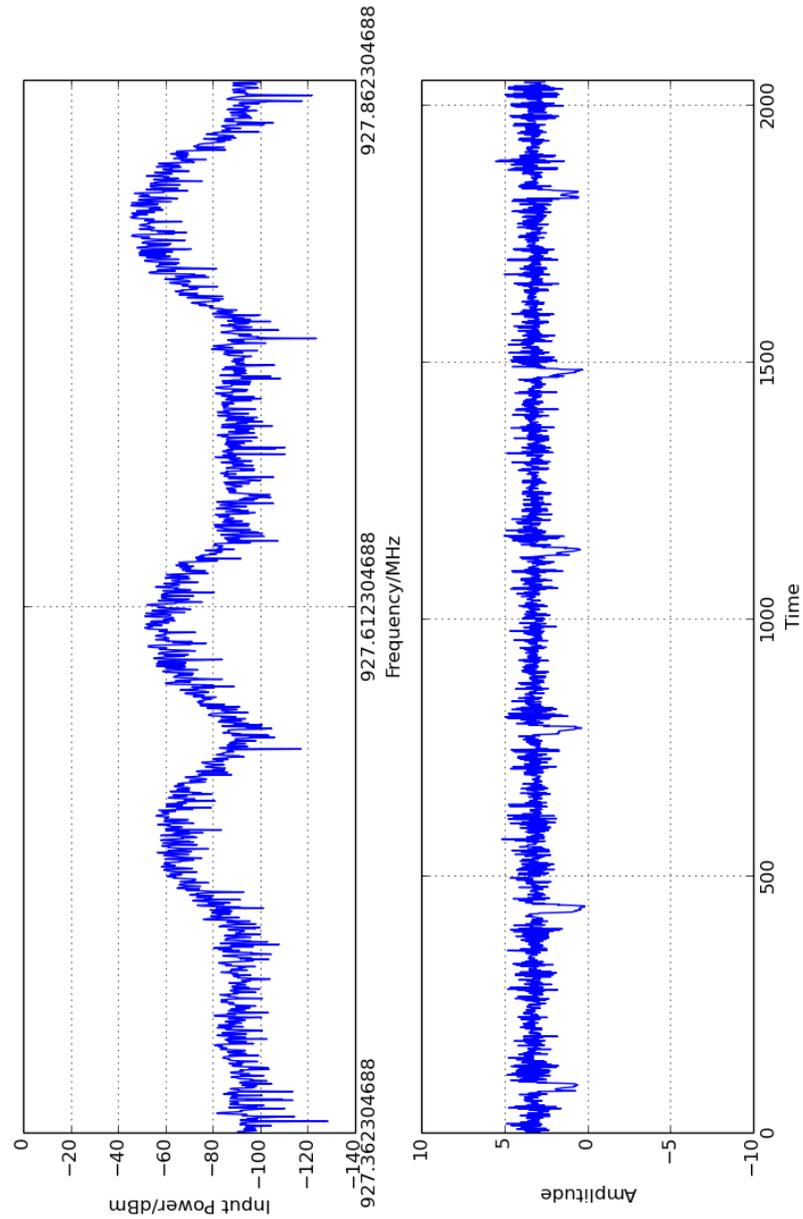
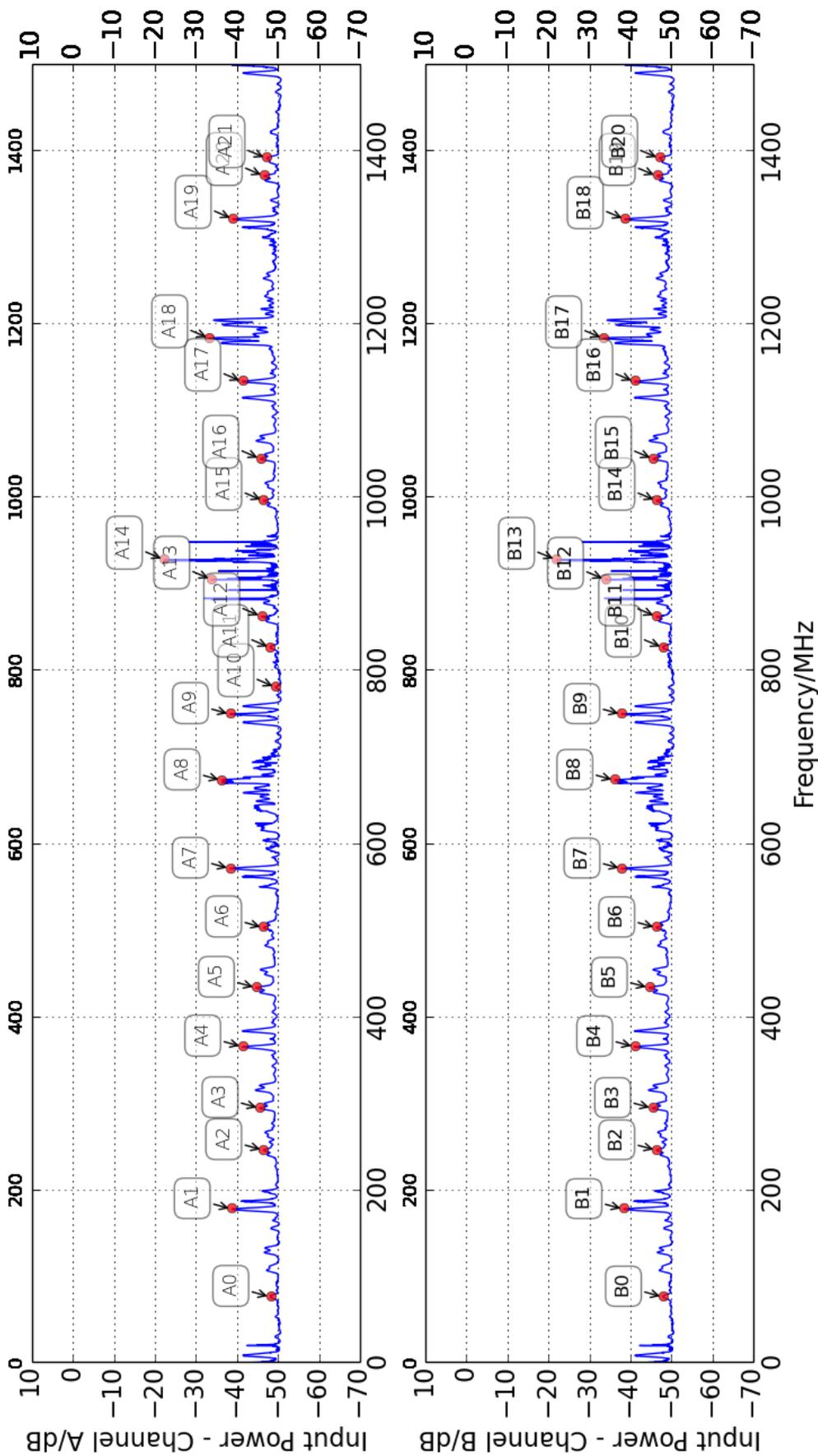


Abbildung A.7: GSM-Signal am Eingang des RTL-SDR mit AM-demoduliertem Zeitsignal



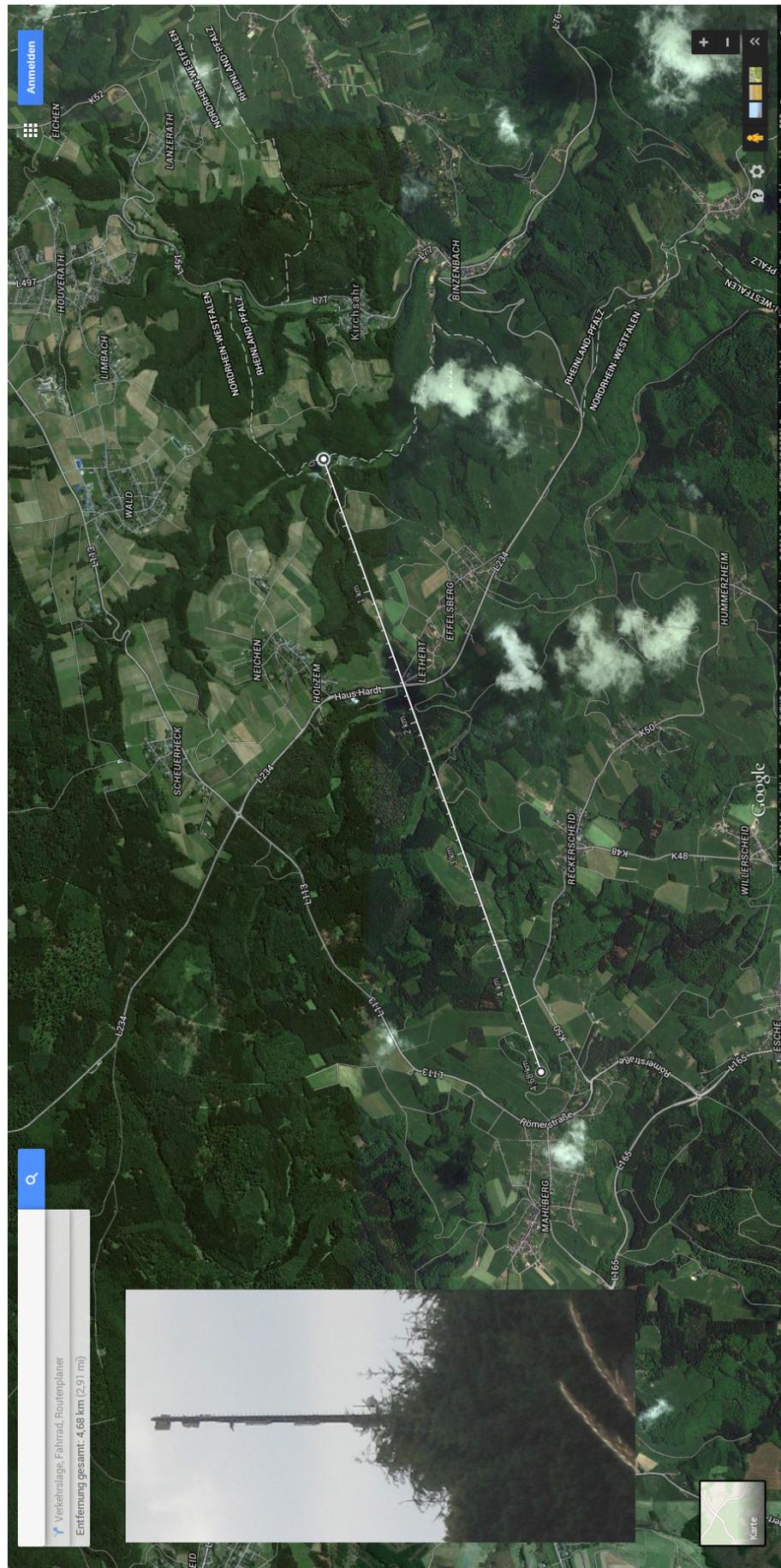


Abbildung A.9: Mögliche GSM-Basisstation der detektierten Mobilfunksignale in einer Entfernung von 4,68 km [13, 16]

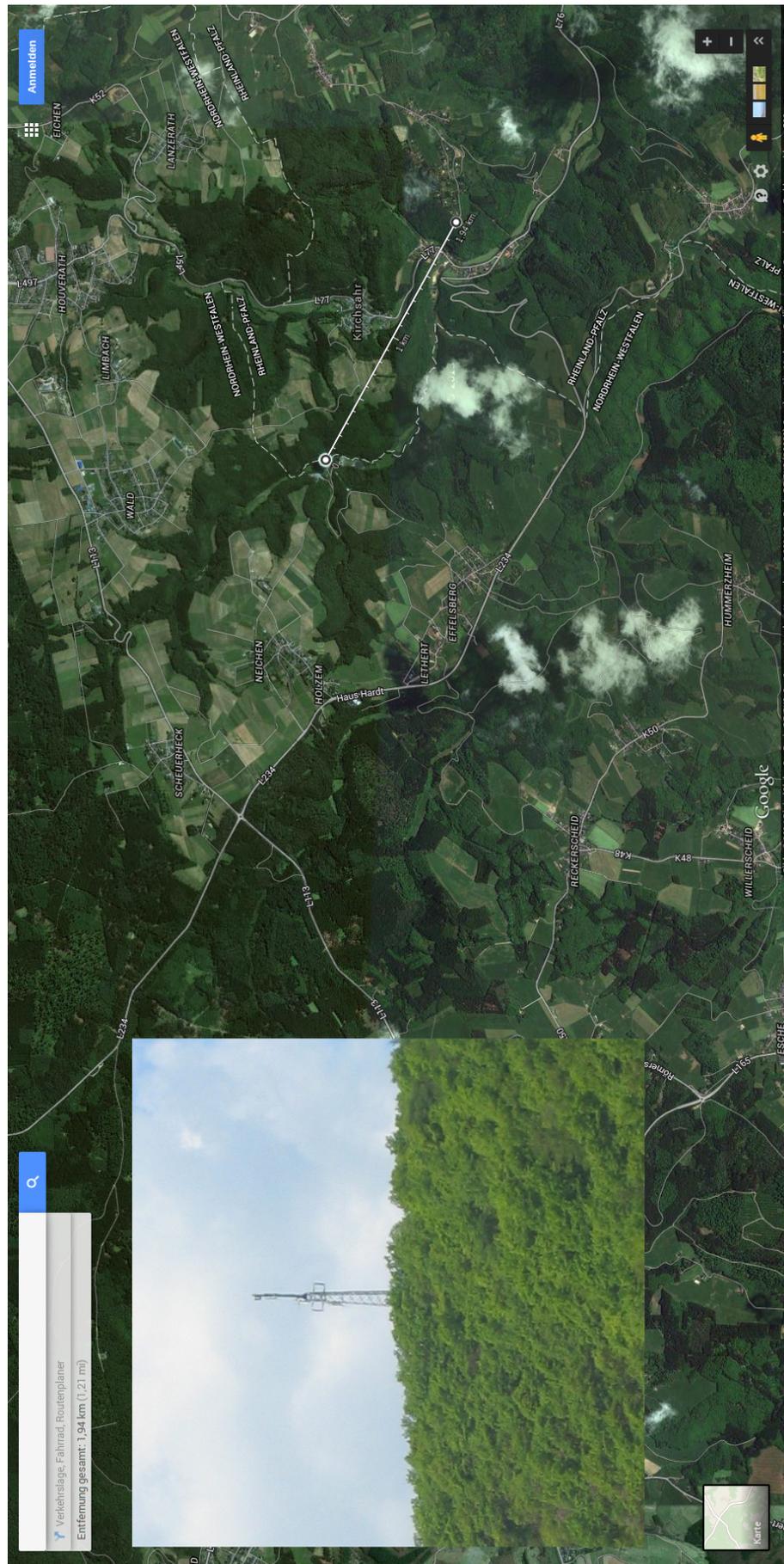


Abbildung A.11: Mögliche GSM-Basisstation der detektierten Mobilfunksignale in einer Entfernung von 1,94 km [13, 15]

A.3.2 Radarsignale des Fraunhofer-Instituts FKIE in Wachtberg

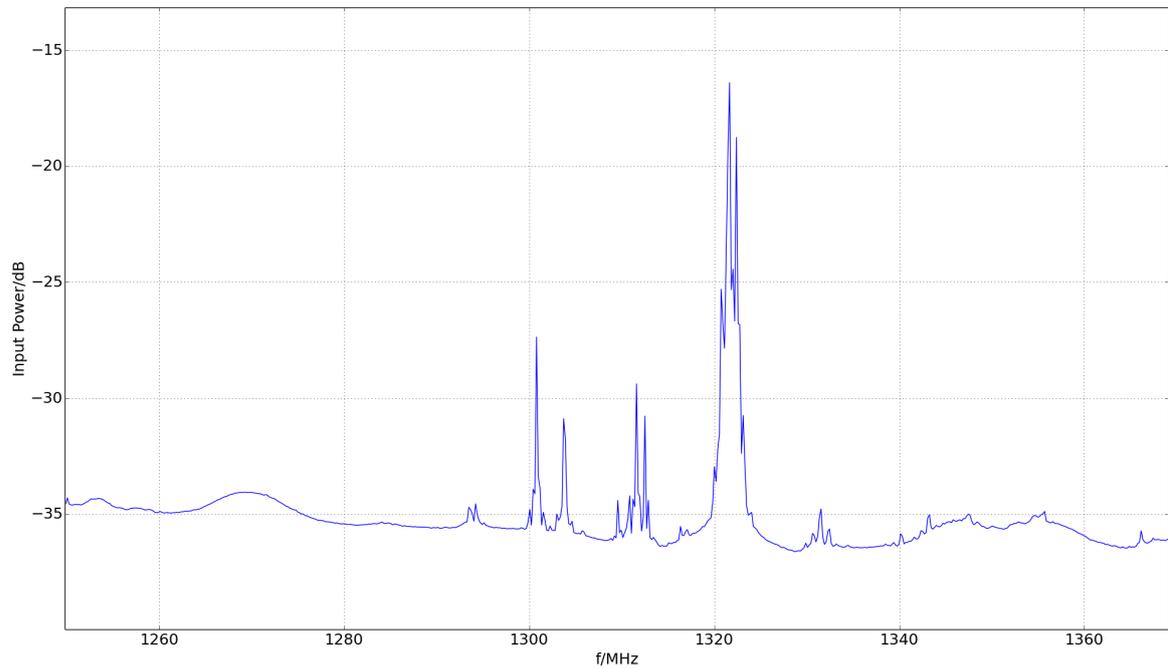


Abbildung A.12: Spektrum der Signale der Radaranlage des Fraunhofer Instituts in Wachtberg bei 1330 MHz

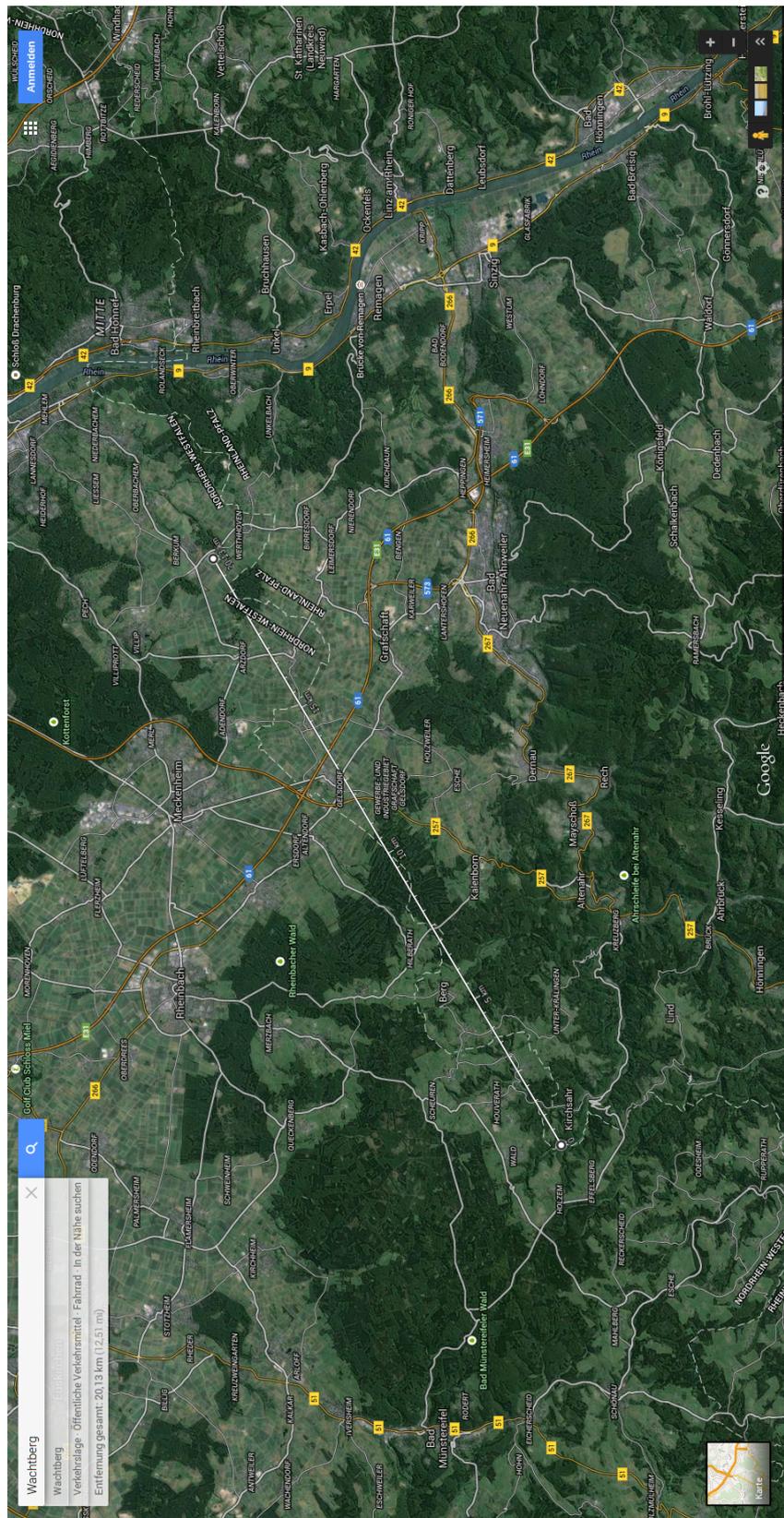


Abbildung A.13: Geografische Lage der Radaranlage in Wachtberg zum Radioteleskop Effelsberg mit einer Entfernung von 20,13 km [14]

A.3.3 Dekodierung des Automatic Dependent Surveillance - Broadcast mittels "Dump1090" [36]

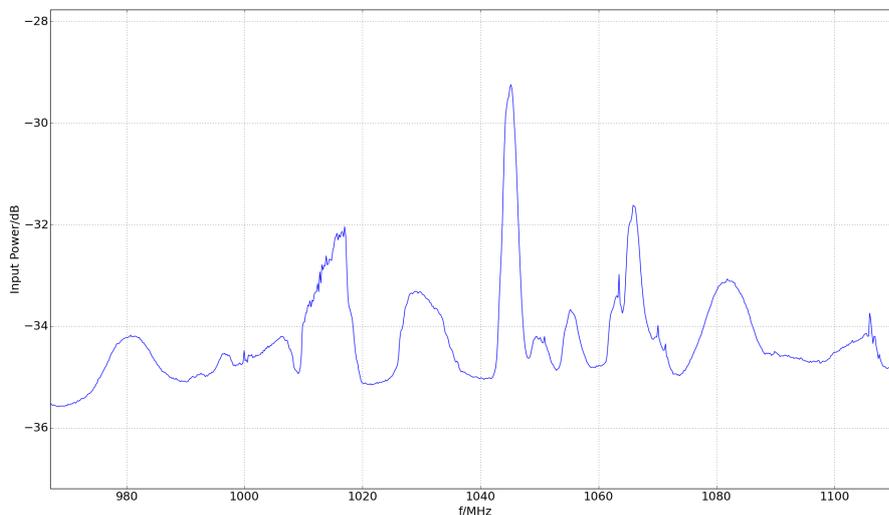


Abbildung A.14: Spektrum eines ADS-B-Signals; aufgenommen mit einem FFTS über das Radioteleskop Effelsberg

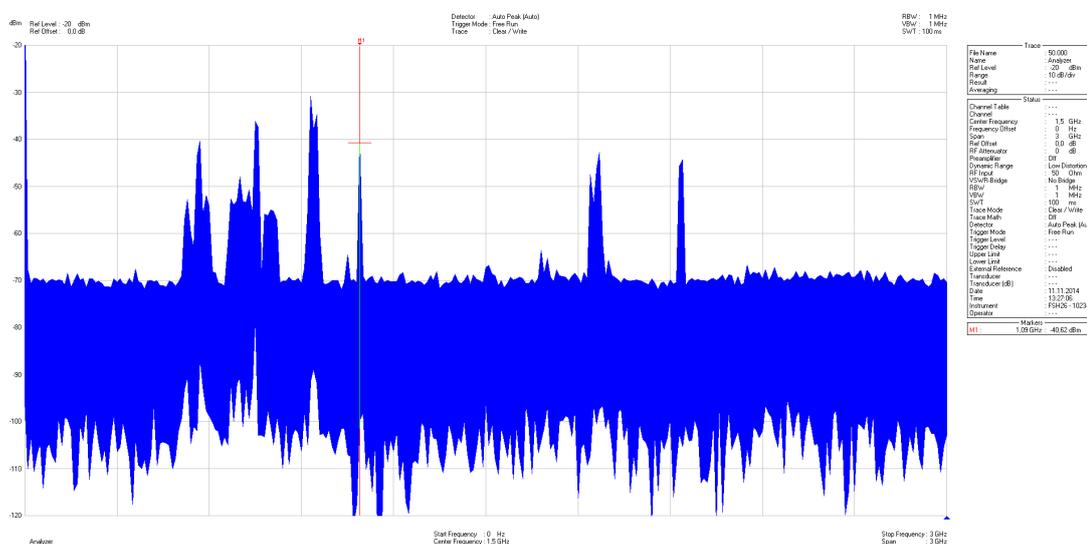


Abbildung A.15: Spektrum eines ADS-B-Signals; aufgenommen mit einem Rohde & Schwarz Spektrumanalysator über das Radioteleskop Effelsberg

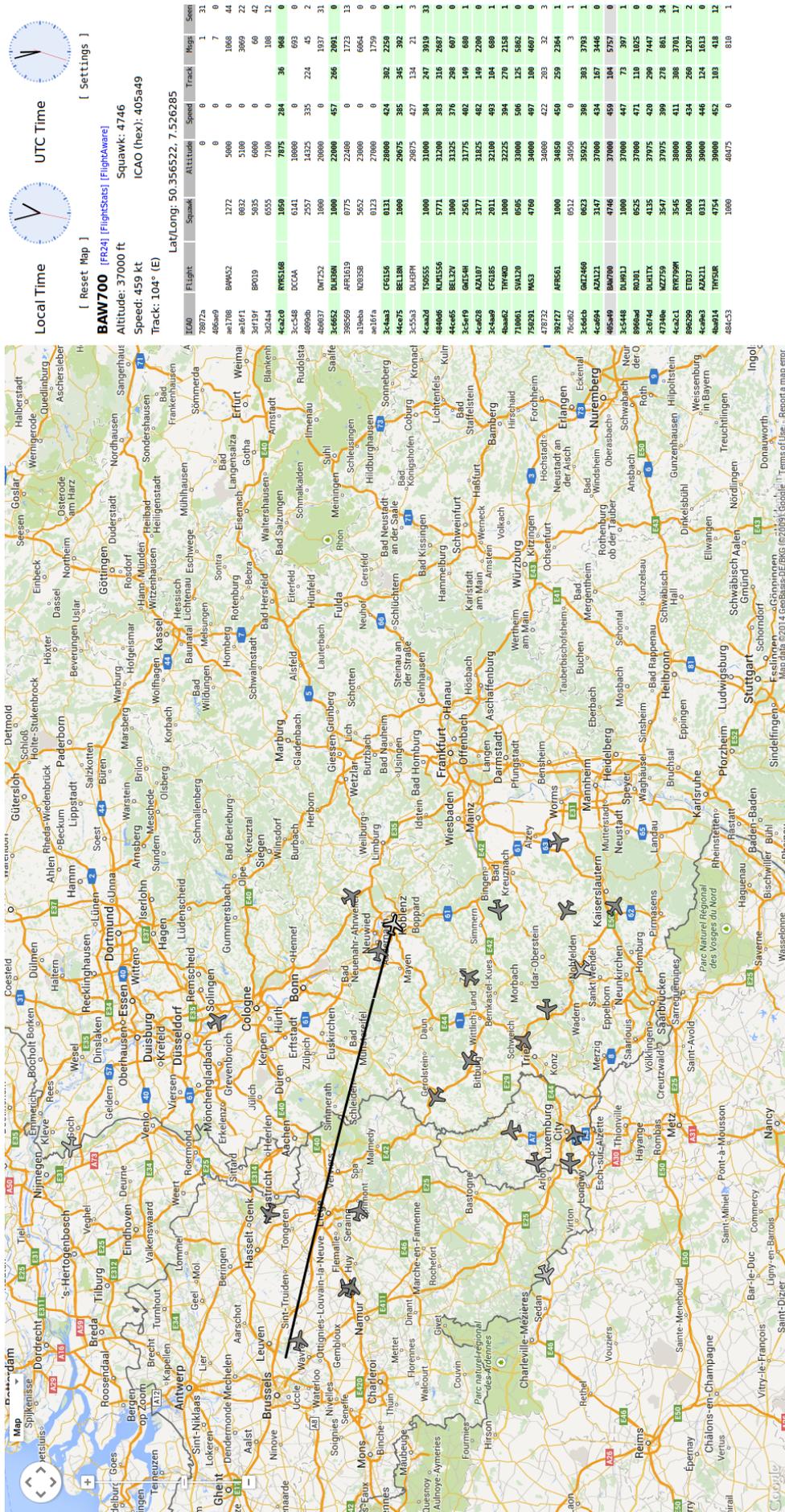


Abbildung A.16: Darstellung der dekodierten ADS-B-Daten in Google-Maps

Quellenverzeichnis

- [1] AARONIA AG: *Frquenzpläne*. <http://www.aaronia.de/grundlagen/frequenzplaene/>. Version: 2014. – [Online; Stand 5. November 2014]
- [2] AHLBRECHT, Horst: *Scheuerheck_Kreuzung.jpg*. http://www.ahlbrecht-su.de/bn/gsm/BILDER/SCHEUERHECK_KREUZUNG.JPG. Version: 2014. – [Online; Stand 13. November 2014]
- [3] ANZAC ELECTRONICS: *Anzac RF and Microwave Components*. 1978
- [4] BADDI, Raju R.: *Interference detection in gaussian noise / Raman Research Institute*. 2010. – Forschungsbericht
- [5] BILLAUER, Eli: *peakdet: Peak detection using MATLAB*. <http://billauer.co.il/peakdet.html>. Version: 2014. – [Online; Stand 30. September 2014]
- [6] BRUGGER, Dr. R. ; GOROL, Reinhard ; OUTTERS, Jan ; SCHERTZ, Alexander: *Frequenzregulierung für Software Defined Radio and Cognitive Radio / Institut für Rundfunktechnik GmbH*. 2010. – Forschungsbericht
- [7] CASS, Stephen: *A 40 Dollar Software-Defined Radio*. <http://spectrum.ieee.org/geek-life/hands-on/a-40-softwaredefined-radio>. Version: Juni 2013. – [Online; Stand 20. September 2014]
- [8] CHAN, Kuo-Chang: *The IP3 Specification - Demystified*. <http://www.maximintegrated.com/en/app-notes/index.mvp/id/5429>. Version: März 2013. – [Online; Stand 18. September 2014]
- [9] ELONICS LTD.: *E4000 Datasheet*. <http://www.nooelec.com/files/e4000datasheet.pdf>. Version: August 2010
- [10] FISCHER, Walter: *Digitale Fernseh- Und Hörfunktechnik in Theorie und Praxis - Mpeg-Basisbandcodierung, Dvb-, Dab-, Atsc-, Isdb-T-Übertragungstechnik, Messtechnik*. 3. Aufl. Berlin Heidelberg New York : Springer-Verlag, 2010. – ISBN 978-3-642-15047-0
- [11] FREE SOFTWARE FOUNDATION: *GNU Radio Documentation*. <http://gnuradio.org/doc/sphinx-3.7/index.html>. Version: 2014. – [Online; Stand 23. September 2014]
- [12] GMBH, Radiometer P.: *RPG - FFTS, Fast Fourier Transform Spectrometer v2.0*
- [13] GOOGLE MAPS: *Radioteleskop Effelsberg*. <https://www.google.de/maps/@50.5208816,6.8505652,5783m/data=!3m1!1e3?hl=de>. Version: 2014. – [Online; Stand 13. November 2014]

- [14] GOOGLE MAPS: *Wachtberg*. <https://www.google.de/maps/place/Wachtberg/@50.567483,7.0418704,23108m/data=!3m1!1e3!4m2!3m1!1s0x47befcae86613691:0x042760fc4a2a7de0?hl=de>. Version: 2014. – [Online; Stand 13. November 2014]
- [15] GOTTSCHLICH, Bernhard: *53505_6AD2.jpg*. http://home.arcor.de/bernhard.gottschlich/53505_6AD2.jpg. Version: 2014. – [Online; Stand 13. November 2014]
- [16] GOTTSCHLICH, Bernhard: *53902_A074.jpg*. http://home.arcor.de/bernhard.gottschlich/53902_A074.jpg. Version: 2014. – [Online; Stand 13. November 2014]
- [17] GRÜNIGEN, Daniel v.: *Digitale Signalverarbeitung* -. 3. Auflage. München; Wien : Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 2004. – ISBN 978–3–446–22861–0
- [18] HIEBEL, Michael: *Grundlagen der vektoriellen Netzwerkanalyse*. 1. Auflage. Rhode und Schwarz, 2006. – ISBN 978–3–939–83705–3
- [19] ITU-R: Report ITU-R M.2242. In: *Cognitive radio systems specific for International Mobile Telecommunications systems*, 2011
- [20] ITU-R: Report ITU-R RA.2126-1. In: *Techniques for mitigation of radio frequency interference in radio astronomy*, 2013
- [21] JANOVSKY, Anton: *Setting up my Raspberry Pi as a SDR Server with RTL-2832U USB dongle*. <http://zr6aic.blogspot.de/2013/02/setting-up-my-raspberry-pi-as-sdr-server.html>. Version: 2013. – [Online; Stand 12. November 2014]
- [22] KAMMEYER, Karl-Dirk: *Nachrichtenübertragung* -. Berlin Heidelberg New York : Springer-Verlag, 2013. – ISBN 978–3–322–94062–9
- [23] KAMMEYER, Karl-Dirk ; KROSCHEL, Kristian ; BOSS, Dieter ; DEKORSY, Armin: *Digitale Signalverarbeitung - Filterung und Spektralanalyse ; mit MATLAB-Übungen*. 4, vollst. überarb. erw. Aufl. 1998. Wiesbaden : Teubner, 1998. – ISBN 978–3–519–36122–0
- [24] KARRES, Jan: *Raspberry Pi: Raspbian installieren*. <http://jankarres.de/2012/08/raspberry-pi-raspbian-installieren/>. Version: April 2014. – [Online; Stand 19. September 2014]
- [25] KLEIN, B. ; HOCHGÜRTEL, S. ; KRÄMER, I. ; BELL, A. ; MEYER, K. ; GÜSTEN, R.: *High-resolution wide-band Fast Fourier Transform Spectrometers / Max-Planck-Institut für Radioastronomie*. 2012. – Forschungsbericht
- [26] MAX-PLANCK-INSTITUT FÜR RADIOASTRONOMIE: *Homepage*. <http://www.mpifr-bonn.mpg.de/>. Version: 2014. – [Online; Stand 30. September 2014]

- [27] MINI CIRCUITS: *SPDT RF SWITCH*. <http://217.34.103.131/pdfs/ZX80-DR230+.pdf>. Version: 2014. – [Online; Stand 5. November 2014]
- [28] MINI CIRCUITS: *Wideband Amplifier*. <http://217.34.103.131/pdfs/ZX60-V63+.pdf>. Version: 2014. – [Online; Stand 5. November 2014]
- [29] NATIONAL INSTRUMENTS: *NI USRP-292x/293x Datasheet*. <http://sine.ni.com/ds/app/doc/p/id/ds-355/lang/de>. Version: 2014. – [Online; Stand 1. Oktober 2014]
- [30] OSMOCOM: *OsmoSDR*. <http://sdr.osmocom.org/trac/wiki/rtl-sdr>. Version: 2014. – [Online; Stand 23. September 2014]
- [31] RADIOMETER PHYSICS GMBH: *Fast Fourier Transform Spectrometer*. http://www.radiometer-physics.de/rpg/html/Products_Spectrometers.html. Version: 2011. – [Online; Stand 23. September 2014]
- [32] RAFAEL MICRO: *R820T Datasheet*. http://superkuh.com/gnuradio/R820T_datasheet-Non_R-20111130_unlocked.pdf. Version: November 2011
- [33] RAUSCHER, Christoph: *Grundlagen der Spektrumsanalyse*. 4. Auflage. Rohde und Schwarz, 2008. – ISBN 978-3-939-83700-8
- [34] RTL1090 ADS-B SOFTWARE: *RTL2832 Schematic*. <http://www.globe-s.eu/images/rtl2832.jpg>. Version: 2014. – [Online; Stand 01. Dezember 2014]
- [35] RTLSDR.ORG: *Wiki*. <http://rtlsdr.org>. Version: 2014. – [Online; Stand 18. September 2014]
- [36] TAYLOR, David: *ADS-B dump1090*. <http://www.satsignal.eu/raspberry-pi/dump1090.html>. Version: 2014. – [Online; Stand 13. November 2014]
- [37] WEPMAN, Jeffrey A.: Analog-to-Digital Converters and Their Applications in Radio Receivers. In: *IEEE Communications Magazine* (1995), S. 39–45
- [38] WIKIPEDIA: *Frequenznutzungsplan* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Frequenznutzungsplan&oldid=124996686>. Version: 2013. – [Online; Stand 1. Oktober 2014]
- [39] WIKIPEDIA: *Raspberry Pi* — *Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=133018228. Version: 2014. – [Online; Stand 17. September 2014]
- [40] WIKIPEDIA: *Schnelle Fourier-Transformation* — *Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Schnelle_Fourier-Transformation&oldid=133508390. Version: 2014. – [Online; Stand 3. Oktober 2014]

-
- [41] WIKIPEDIA: *Standardabweichung* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Standardabweichung&oldid=134294235>. Version: 2014. – [Online; Stand 1. Oktober 2014]
- [42] WIKIPEDIA: *Universal Serial Bus* — *Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Universal_Serial_Bus&oldid=134497303. Version: 2014. – [Online; Stand 1. Oktober 2014]
- [43] WIKIPEDIA: *Schnelle Fourier-Transformation* — *Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Schnelle_Fourier-Transformation&oldid=138918932. Version: 2015. – [Online; Stand 17. Februar 2015]

Eidesstattliche Erklärung

Erklärung zur Masterarbeit

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt beziehungsweise in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (CD-ROM).

Unterschrift

Ort, Datum