



Bachelorstudiengang Ingenieurpädagogik

Studiengang: BINGP19

Bachelorarbeit

# Realisierung eines Kommunikationsempfängers mittels Software Defined Radio und bladeRF

Hochschule Merseburg

Fachbereich INW

eingereicht von: Richard Dittrich

Hochschulbetreuer: Prof. Dr. Mückenheim

Zweitbetreuer: Prof. Dr. Klein

Abgabetermin: 06.09.2022

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>05</b>
<b>2</b>	<b>Hardware</b>	<b>06</b>
2.1	bladeRFx115	06
2.2	FPGA	09
2.3	Breitband-HF-Transceiver	10
2.4	Programmable Synthesizer	13
<b>3</b>	<b>Software</b>	<b>14</b>
3.1	Allgemeine Informationen	14
3.2	Aufbau der Plattform	15
3.3	Erste Schritte mit GNU Radio	16
<b>4</b>	<b>Signalübertragung mittels bladeRF</b>	<b>19</b>
4.1	Frequenzmodulation	19
4.2	Amplitudenmodulation	25
4.3	Digitale Modulation	29
<b>5</b>	<b>Erstellen einer Versuchsanleitung</b>	<b>33</b>
5.1	Versuchsziele	33
5.2	Vorbereitende Aufgaben	33
5.3	Grundlagen	33
5.3.1	bladeRFx115	33
5.3.2	GNU Radio	34
5.3.3	Frequenzmodulation (FM)	36
5.3.4	Amplitudenmodulation (AM)	37
5.3.5	Programmable Synthesizer HM8134	38
5.4	Versuchsdurchführung	39
5.4.1	Erster Versuch mit GNU Radio	39
5.4.2	Frequenzmodulation mit GNU Radio und bladeRF	40
5.4.3	Amplitudenmodulation mit GNU Radio und bladeRF	42
5.5	Auswertung	43
<b>6</b>	<b>Fazit</b>	<b>44</b>
<b>7</b>	<b>Literatur- und Quellenverzeichnis</b>	<b>45</b>
<b>8</b>	<b>Abbildungsverzeichnis</b>	<b>48</b>

# Abkürzungsverzeichnis

SDR	Software Defined Radio
USB	Universal Serial Bus
MHz	Mega-Hertz
GHz	Giga-Hertz
kHz	Kilo-Hertz
HF	Hochfrequenz
GPS	Global Positioning System
ATSC	Advanced Television System Committee
Wifi	Wireless Fidelity
FPGA	Field Programmable Gate Array
Tx	Transceiver exchange
Rx	Receiver exchange
ksps	Kilo-Samples per Second
MspS	Mega-Samples per Second
GPIF II	General Programmable Interface Generation II
SPI	Serial Peripheral Interface
JTAG	Joint Test Action Group
IQ	In Phase- & Quadraturverfahren
RXIQ	IQ-Empfangssignal
TXIQ	IQ-Sendesignal
VHDL	Very high Speed Integrated Circuit Hardware Description Language
DAC	Digital to Analog Converter
ADC	Analog to Digital Converter
LUT	Lookup-Tabellen
V	Volt
CIN	Carry/Control-Logic Input
COU	Carry/Control-Logic Output
LNA	Low Noise Amplifier
PA	Power Amplifier
LPF	Low Pass Filter
VGA	Variable Gain Amplifier

PLL	Phase Locked Loops
TXD	TX-Datenbus
IDAC	In-Phase-DAC
QDAC	Quadratur-DAC
RXD	RX-Datenbus
FM	Frequenzmodulation
NF-Signal	Niederfrequentes Signal
WBFM	Wideband FM
NBFM	Narrowband FM
DC	Direct Current
dB	Dezibel
AM	Amplitudenmodulation
MSK	Minimum Shift Keying
GMSK	Gaussian Minimum Shift Keying
GSM	Global System for Mobile Communications
ISI	Intersymbolinterferenzen
CVSD	Continuously variable Slope Delta Modulation
UKW	Ultrakurzwelle
cm	centimeter
MB	Mega-Byte

# 1. Einleitung

Der Austausch von Daten und Informationen ist für die heutige Zeit unerlässlich. Eine schnelle und korrekte Signalübertragung ist für die Nachrichtentechnik ein wichtiger Bestandteil. Seit der ersten Funkverbindung Ende des 19. Jahrhunderts hat sich die Kommunikationstechnik stetig weiterentwickelt und fortwährend selbst überformt. Eines der besten Beispiele für die Anwendung dieser Technik ist das Radio. Hier kommen Modulationsverfahren wie die Amplituden- und die Frequenzmodulation zum Einsatz. Neben diesen Verfahren gibt es noch weitaus mehr Verfahren in der Nachrichtentechnik.

Zur Erforschung und Veranschaulichung dieser Verfahren gibt es heutzutage unzählige Möglichkeiten. Das Ziel dieser Arbeit ist es, eine Kommunikation nachzubauen und dabei zwei analoge und ein digitales Modulationsverfahren anzuwenden. Das Resultat der Kommunikation soll die Übertragung eines Audiosignals sein. Dafür kommt das Konzept des SDR (Software Defined Radio) zum Einsatz. Mit einem sogenannten bladeRFx115 wird ein Kommunikationsempfänger nachempfunden. Der Sender für die Kommunikation wird in einer korrespondierenden Bachelorarbeit beschrieben [1]. Die Signalverarbeitung erfolgt mithilfe der Software von GNU Radio. Das Zusammenspiel dieser beiden Komponenten hat zum Ziel, das Audiosignal unter Anwendung der drei Modulationsverfahren zu übertragen. Im Vordergrund steht, die Audioübertragung zu testen und falls nötig so zu verbessern, damit eine gute Signalqualität erreicht wird.

Nach der Einleitung wird in Kapitel 2 auf den Aufbau und die Funktionsweise des bladeRFx115 eingegangen. Der Abschnitt beinhaltet die wichtigsten Komponenten im Inneren des bladeRF und ihre konkreten Aufgabenbereiche. Das nachfolgende Kapitel beschreibt die GNU-Radio-Software detaillierter. Neben den allgemeinen Informationen behandelt das Kapitel den Aufbau der Softwareoberfläche und gibt einen kleinen Einstieg zum Erstellen von Flussdiagrammen. Das Kapitel 4 ist der Praxisteil der Arbeit. Zu den durchgeführten Versuchen erfolgen ausführliche Dokumentationen und Auswertungen, die den Verlauf, die Implementierung und das Ergebnis beurteilen. Damit wird eine nachgehende Analyse der Versuche gewährleistet. In den drei verschiedenen Versuchen kommen drei unterschiedliche Modulationsverfahren zum Einsatz, die durch die Anwendung der GNU-Radio-Software realisierbar sind. Im fünften Kapitel wird zu den Experimenten eine Versuchsanleitung erstellt, die es Studenten ermöglichen soll, einen Einblick in die Arbeit mit dem bladeRF und die Kommunikationstechnik zu bekommen. Nach einem kleinen Einführungsversuch sollen die Studenten die verschiedenen Versuche an zwei Modulationsverfahren erproben. Für die Versuchsdurchführung sind einige Vorkenntnisse nötig, die durch das Stellen von verschiedenen vorbereitenden Fragen überprüft werden soll. Die Arbeit schließt mit einer kurzen Reflektion, wobei im Speziellen auf die Versuche und deren Ergebnisse eingegangen wird.

## 2. Hardware

### 2.1 bladeRF115

Das bladeRF ist ein von der Firma Nuand entwickeltes Funkgerät, welches als Sender, Empfänger oder Sendeempfänger verwendet werden kann. In Abbildung 1 ist das verwendete bladeRFx115 dargestellt. Die Eigenschaften des Funkgerätes werden durch eine Software bestimmt. Damit ist es möglich, die drahtlose Kommunikation genauer zu erforschen. Die Betriebssysteme von Linux, Windows und macOS



Abbildung 1 Darstellung des bladeRF [2]

unterstützen die Arbeit mit dem bladeRF. Die erforderlichen Bibliotheken und Firmware zur ordnungsgemäßen Ausführung des bladeRF sind online verfügbar. Das bladeRF verfügt über einen USB (Universal Serial Bus) 3.0-Anschluss. Somit kann das bladeRF eine Verbindung zum Laptop oder Computer herstellen, auf dem die bereitgestellten Treiber installiert sind [2].

Bei der Installation der Treiber für die Windows-Version kam es zu Komplikationen mit der Version bestimmter Bibliotheken. Deshalb wurde nicht die aktuelle Version von 2021 verwendet, sondern die Ausgabe von 2019. Somit wurden die Treiber für den bladeRF überspielt und die Kommunikation mit der verwendeten Software konnte hergestellt werden.

Zu den wichtigsten Eigenschaften des bladeRF zählt, dass er leistungsstark und mit einem Gewicht von 80 Gramm sehr leicht zu transportieren ist. Das Modul ist 8,7 cm (centimeter) lang, 13,1 cm breit und 1,8 cm hoch. Für den Betrieb des bladeRF muss keine herkömmliche Verbindung zur Steckdose erfolgen. Die Spannungsversorgung des Funkmoduls erfolgt über den USB 3.0-Anschluss. Dennoch besteht die Möglichkeit, den bladeRF über einen externen Anschluss mit 5 V (Volt) Gleichspannung zu versorgen. Das bladeRF arbeitet ab Werk im Frequenzbereich zwischen 300 MHz (Mega-Hertz) bis 3,8 GHz (Giga-Hertz) und deckt somit ein großes Frequenzspektrum ab. Aufgrund dieser technischen Gegebenheiten ist ein flexibler und vielseitiger Einsatz des bladeRF möglich. Daraus ergibt sich die Möglichkeit, dass es nicht nur als HF- (Hochfrequenz) Modem arbeitet, sondern auch als GPS- (Global Positioning System) Empfänger, ATSC-Sender (Advanced Television Systems Committee) oder Bluetooth/Wifi- (Wireless Fidelity) Client verwendet werden kann. Ebenso kann eine selbstständige Modifizierung des bladeRF erfolgen. Dazu gehört neben der Firmware für den USB 3.0 auch der Altera Cyclone IV FPGA (Field Programmable Gate Array) Chip, der neu bespielt werden kann [3, 4, 5].

Die unabhängigen Tx- (transceiver exchange) und Rx- (receiver exchange) Pfade ermöglichen dem bladeRF die Signalübertragung und den Signalempfang gleichzeitig durchzuführen und somit im Vollduplex-Kommunikationsmodus zu arbeiten [3, 4].

Für die Umwandlung der Daten von Analog zu Digital oder Digital zu Analog wird ein analoges zeitkontinuierliches Signal in zeitdiskrete digitale Abtastwerte umgeformt, wobei jeder der einzelnen aufgenommenen digitalen Abtastwerte die Amplitude des analogen Signals darstellt. Die Samplerate des bladeRF beginnt ab 160 ksps (Kilo-Samples per Second) und kann bis zu einem Wert von 40 Msps (Mega-Samples per Second) eingestellt werden. Dabei besitzt jedes digitale Sample eine Breite von 12 Bit. Das Funkmodul ist zudem in der Lage, eine Kanalbandbreite von 1,5 MHz bis hin zu 28 MHz umzusetzen [3].

In Abbildung 2 wird das bladeRF mit seiner inneren Architektur veranschaulicht. Die drei Hauptkomponenten des bladeRFx115 sind der Lime Micro LMS6002D RF Transceiver, der Altera Cyclone IV E FPGA und der Cypress FX3 USB 3.0 Controller, der den USB 3.0-Anschluss beinhaltet. Die Kommunikation zwischen den verschiedenen Bausteinen erfolgt über den USB-Anschluss komplett busbetrieben. Der USB-Anschluss ist zudem die Schnittstelle zwischen dem Computer und dem bladeRF. Des Weiteren ist der USB 3.0-Port auch abwärtskompatibel zu USB 2.0. Dies hat zur Folge, dass es bei der Abtastrate zu Einschränkungen kommt. Sollte der Benutzer-Computer nur USB 2.0 unterstützen, ist es für den bladeRF nicht mehr möglich, seine maximale Abtastrate von 40 Msps zu erreichen [3, 4].

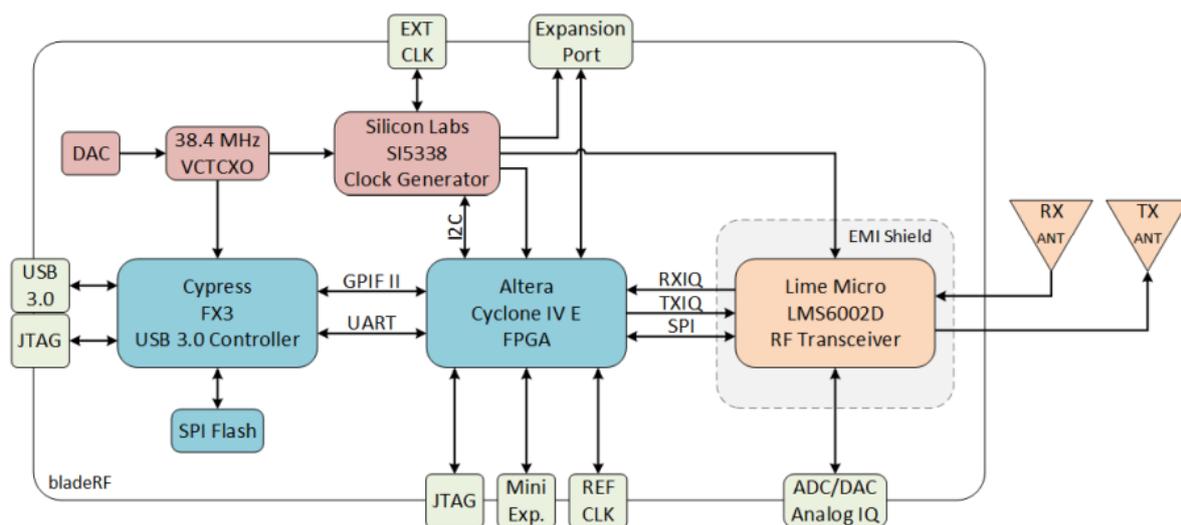


Abbildung 2 Aufbau des bladeRFx115 [5]

Das bladeRF ist mit einem Cypress FX3 USB 3.0 Controller ausgestattet. Der Mikrocontroller hat einen integrierten ARM926EJ-S Prozessor verbaut, welcher mit einer Taktfrequenz von 200 MHz arbeitet [3].

Dieser steuert den Datenaustausch zwischen dem USB 3.0-Port und der GPIF II- (General Programmable Interface Generation II) Schnittstelle. Die GPIF II- Schnittstelle ist dann mit dem FPGA-Chip von Altera verbunden. Der Cypress FX3 Controller besitzt eine SPI- (Serial Peripheral Interface) Schnittstelle, welche für die Speicherung der Firmware auf dem Mikrocontroller und auf dem FPGA-Chip zuständig ist [3].

Die SPI-Schnittstelle ist ein 4 MB (Mega-Byte) Flashspeicher, der den Firmware-Code der beiden Komponenten sichert. Damit wird beim Hochfahren des bladeRF die aktuelle Firmware in den Mikrocontroller und in den FPGA-Chip geladen. Die gespeicherte Firmware kann über den USB- oder den JTAG- (Joint Test Action Group) Anschluss aktualisiert werden [3].

Das wichtigste Bauteil des bladeRF ist der Altera Cyclone IV E FPGA. Der FPGA-Chip dient nicht nur als Bindeglied zwischen dem Cypress FX3 Controller und dem LMS6002D Breitband-HF-Transceiver, sondern auch zur Durchführung der Signalverarbeitung und zur Steuerung des HF-Transceiver. Zur Datenübertragung zwischen dem HF-Transceiver und dem FPGA werden die beiden unabhängigen RXIQ- (IQ- (In Phase- & Quadratur-Verfahren) Empfangssignal) und TXIQ- (IQ-Sendesignal) Pfade verwendet. Das bladeRF kann somit im Vollduplexmodus arbeiten. Aufgrund der vollständigen Programmierbarkeit des bladeRF, ist es möglich, dass Funkmodul mit den bereitgestellten Firmware-Codes für den Cypress FX3 Controller und den VHDL- (Very high Speed Integrated Circuit Hardware Description Language) Codes für den FPGA-Chip einfach an spezielle Signalverarbeitungsaufgaben anzupassen. Für die Neu-Programmierung der beiden Bauteile kann auch hier der USB- oder der JTAG-Anschluss verwendet werden [3].

Der spannungsgesteuerte Quarzoszillator (VCTCXO) arbeitet mit einer Frequenz von 38,4 MHz und liefert dem Cypress-Controller und dem SI5338 Taktgenerator eine stabile Frequenzreferenz. Für die genaue Spannungssteuerung des Quarzoszillator wird ein 16-Bit-DAC (Digital-to-Analog-Converter) vorgeschaltet. Die Samplingtiefe (Bittiefe) von 16 Bit erreicht eine hohe Genauigkeit, damit eine Rekonstruktion der analogen Amplituden möglich ist. Für die Synthetisierung des Taktsignals für den FPGA-Chip und für den HF-Receiver ist ebenfalls der SI5338 Taktgenerator verantwortlich. Ebenso wird der DAC und der ADC (Analog-to-Digital-Converter) die im HF-Transceiver verbaut sind, von den Takten des Taktgenerators angesteuert. Somit ist es möglich, dass die Abtastrate des DACs und des ADCs anpassungsfähig bleibt. Wie in Abbildung 1 zu sehen ist, wird für die Abschirmung von elektromagnetischen Interferenzen der HF-Transceiver mit einer silbernen Abschirmung versehen. Die Abschirmung schützt den Transceiver vor äußeren elektromagnetischen Störungen, dient aber gleichzeitig auch dazu die innere Abstrahlung des HF-Transceiver zu minimieren [3].

## 2.2 FPGA

Die FPGA-Chips zählen zu den integrierten Schaltungen. Die Vielzahl von logischen Bausteinen, die der Chip beinhaltet, führt dazu, dass viele Schaltungen realisiert werden können. Bei der Programmierung des Chips kommt es nicht nur vordergründig auf das Festlegen zeitlicher Abläufe, sondern auf die gewünschte Schaltungsstruktur des Benutzers an [3, 6].

Das Grundprinzip der FPGAs beruht auf einer tabellarisierten Implementierung. Der prinzipielle Aufbau einer FPGA ist in Abbildung 3 dargestellt. Der FPGA wird in viele kleine Logikbausteine unterteilt. Diese Logikblöcke enthalten sogenannte LUTs (Lookup-Tabellen). Der Benutzer kann die Ausgangswerte durch die Programmierung der verschiedenen Eingangsparameter der LUTs beeinflussen. Zur Sicherung der Ausgangswerte der Logikblöcke enthält dieser Flip-Flops. Die Anzahl der Flip-Flops ist genau so groß wie die Anzahl der vorhandenen LUTs. Für die Kommunikation zwischen den Blöcken sind diese über ein Verbindungsnetzwerk miteinander verknüpft. Für den Datenaustausch zwischen zwei Logikblöcken werden programmierbare Schalter im Verbindungsnetzwerk so gestellt, dass eine elektrische Verbindung zwischen den beiden Logikblöcken entsteht. Somit wird es ermöglicht, dass komplexe logische Schaltungen durch die Verknüpfung mehrerer Logikblöcke realisiert werden können. Die Anzahl der Logikblöcke bestimmt die Komplexität der logischen Schaltungen. Damit der Verdrahtungsaufwand

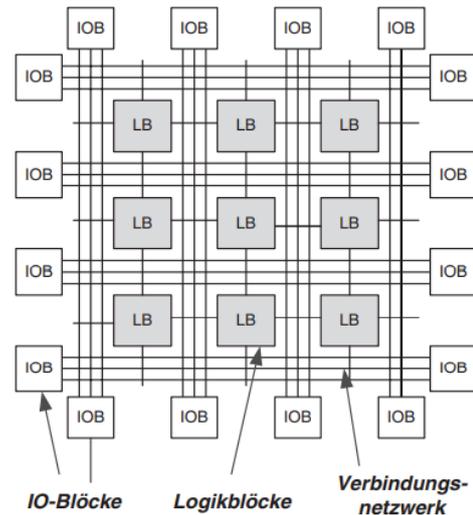


Abbildung 3 Aufbau des FPGA [7]

zwischen den Blöcken nicht zu groß wird, sind nicht alle Logikblöcke miteinander verbunden. Diese Tatsache ist ein weiterer limitierender Faktor für die Komplexität der logischen Schaltungen. Der dritte wichtige Bestandteil im Aufbau des FPGAs sind die Eingabe- und Ausgabe-Blöcke (I/O-Blöcke). Mithilfe dieser Bausteine ist der Benutzer in der Lage, bestimmte Parameter anzupassen. Ein Beispiel hierfür wäre die Anpassung des Logikpegels. Geht man davon aus, dass die interne Versorgungsspannung 1,8 V beträgt, kann der Signalpegel vom Anwender auf 3,3 V erhöht werden. Die Grundstruktur eines Logikblocks beinhaltet neben den LUTs und den Flip-Flops noch eine weitere Komponente. Wie in Abbildung 4 dargestellt ist, gehen die Anschlüsse CIN und COUT in die Carry/Control-Logic. Mithilfe der Anschlüsse und der Carry/Control-Logic wird ein schnellerer Durchlauf ermöglicht und die Verzögerungszeit bei der Verbindung der Logikblöcke reduziert [7].

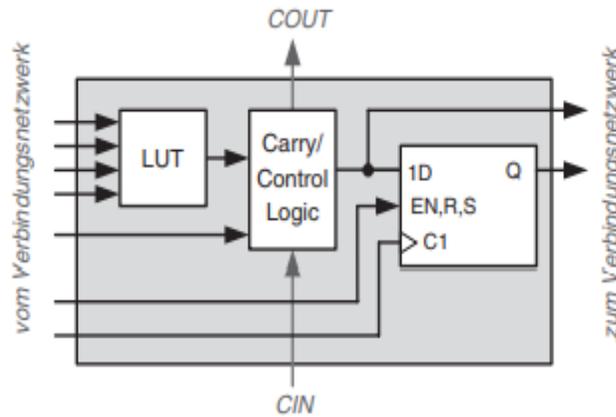


Abbildung 4 Aufbau eines Logikblocks [7]

Im bladeRFx115 ist ein Altera Cyclone IV E FPGA-Chip verbaut. Dieser ist vollständig programmierbar und ist die Schnittstelle zwischen dem Cypress FX3 Mikrocontroller und HF-Transceiver. Bei der Variante des Altera IV E FPGA umfasst der Chip bis zu 115000 Logikelemente. Nach der Anzahl der Logikelemente im FPGA-Chip werden auch die Varianten des bladeRF unterschieden [8].

## 2.3 Breitband-HF-Transceiver

Eine weitere wichtige Komponente im bladeRF ist der LMS6002D Breitband-HF-Transceiver. Im Deutschen wird der Transceiver als Sendeempfänger bezeichnet. Wie die Bauteilbezeichnung vermuten lässt, ist diese Komponente im bladeRF für das Senden und Empfangen der Signale zuständig. Diese Teilung des Sendeempfängers findet sich auch im Aufbau wieder. Die Eingänge und Ausgänge, die mit Tx versehen sind, gehören zu der Senderseite. Die Elemente, die zur Empfangsseite zugeordnet werden müssen, sind mit Rx versehen [9].

Der HF-Transceiver gehört ebenfalls mit zu den integrierten Schaltungen und besitzt verschiedene Bauelemente. Dazu gehören LNA (Low Noise Amplifier), PA (Power Amplifier), LPF (Low Pass Filter), Tx/Rx -Mischer, Tx/Rx -Verstärker mit VGA (Variable Gain Amplifier) und ADC/DAC. Die Funkübertragung erfordert es, ein digitales Breitbandsignal in ein analoges HF-Signal umzuformen oder ein analoges HF-Signal in ein digitales Breitbandsignal umzuwandeln, sodass die weitere Signalverarbeitung gewährleistet ist. Der Transceiver deckt ein breites Frequenzspektrum im Bereich von 300 MHz bis 3,8 GHz ab. Aufgrund der Feinabstimmung eignet sich der Transceiver gut für viele drahtlose Kommunikationen [3].

Interne Register steuern die Frequenz und die Kanalbandbreite des Transceivers. Es ist möglich, diese Register über den SPI-Port im Transceiver zu konfigurieren. Ebenso verfügt der Sendeempfänger über zwei unabhängige I/Q-Signalschnittstellen, die es ermöglichen, die beiden Signale (Rx-Signal und Tx-Signal) mit jedem Signalprozessor zu verbinden [3].

In Abbildung 5 wird der innere Aufbau des LMS6002D Breitband-HF-Transceiver genauer dargestellt. Dabei wird die Unterteilung des Sendeempfängers in einen Sende- und in einen Empfangsteil deutlich. Beide Signalpfade verfügen über ihren eigenen Mischer, der von den PLLs (Phase-Locked-Loops) (TxPLL und RxPLL) gesteuert wird. Die Synchronisation der einzelnen Takte erfolgt mithilfe der PLLs. Die beiden PLLs (TxPLL und RxPLL) teilen sich eine Referenztaktquelle (PLLCLK). Unabhängig davon können die PLLs verschiedene Trägerfrequenzen im Bereich zwischen 0,3 GHz bis 3,8 GHz für ihre eigenen Mischer erzeugen [3, 10, 11].

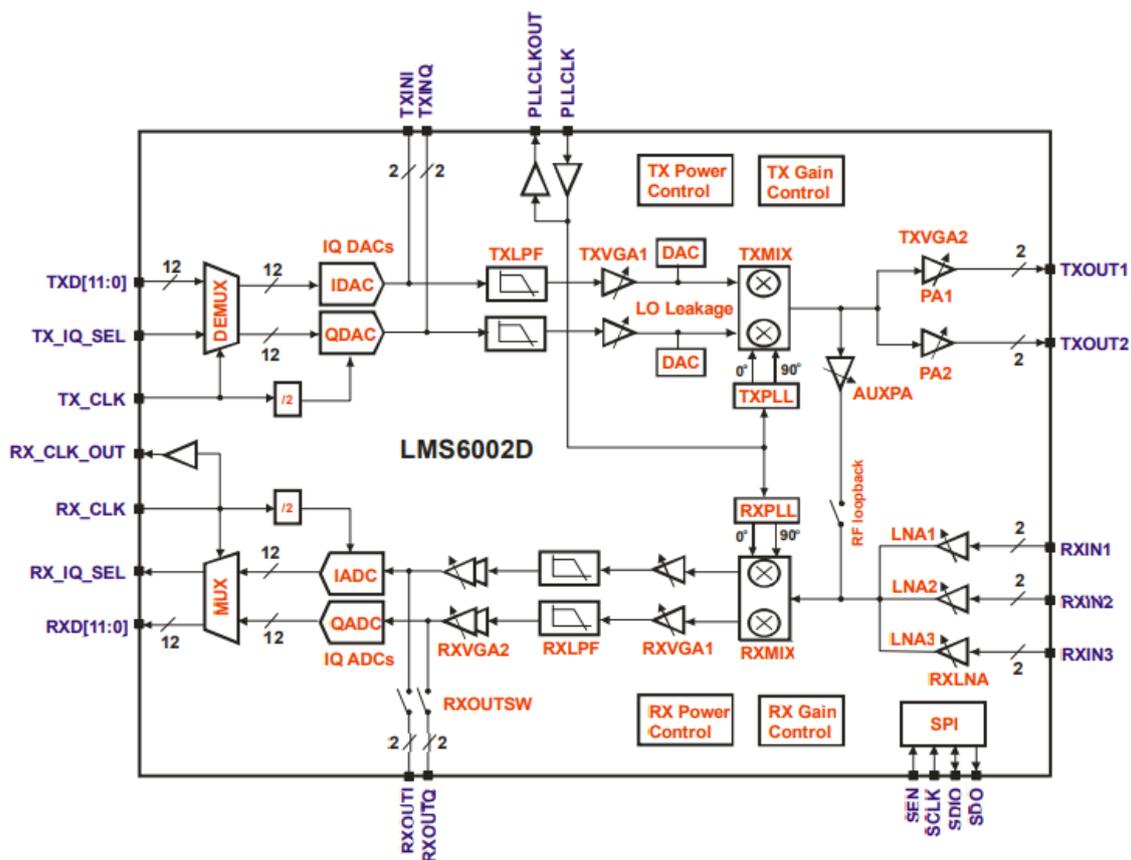


Abbildung 5 Aufbau des HF-Transceivers [11]

Der Tx-Pfad beginnt mit dem parallelen 12-Bit-TXD (Tx-Datenbus), der von einem digitalen Signalprozessor I/Q-Abtastwerte zur Verfügung gestellt bekommt. Die I/Q-Abtastwerte kommen abwechselnd in einem Strom aus I- und Q-Werten an. Die Identifizierung, welcher der beiden Werte gerade übergeben wurde, erfolgt durch das Tx-IQ-Signal Select-Flag-Bit (TX\_IQ\_SEL) [3, 11].

Sobald ein I-Abtastwert auf dem TXD ankommt, setzt sich das Select-Flag-Bit auf den Wert 1. Der De-Multiplexer wird angesteuert und dieser leitet den I-Abtastwert weiter zum IDAC (In-Phase-DAC). Kommt auf dem TXD ein Q-Abtastwert an, dann wird das Select-Flag-Bit auf den Wert 0 gesetzt. Erneut wird der De-Multiplexer angesteuert, der den Q-Abtastwert zum QDAC (Quadratur-DAC) übergibt [3].

Der IQ-DAC besteht damit aus einem 12-Bit-QDAC und einem 12-Bit-IDAC. Die Abtastrate des IQ-DAC liegt zwischen 0,16 MHz bis 40 MHz. Damit ist es möglich, dass 0,16 bis hin zu 40 Millionen I- und Q-Abtastwerte pro Sekunde umgewandelt werden können. Dabei muss beachtet werden, dass die Anzahl an IQ-Abtastwerten eine Kombination aus I- und Q-Abtastwerten ist. Daraus ergibt sich für den IDAC und den QDAC die halbierte Abtastrate im Bereich von 0,08MHz bis 20MHz. Nachdem die IQ-Werte den IQ-DAC durchlaufen haben, werden aus den digitalen Abtastwerten analoge Abtastwerte. Der Tx-LPF wird für die Begrenzung der Bandbreite von I- und Q-Signalen verwendet. Darüber hinaus glättet der Filter die Wellenform der beiden Signale. Der LPF ist flexibel einstellbar und hat verschiedene wählbare Bandbreiten zwischen 0,75 MHz bis 14 MHz. Das hat zur Folge, dass die Bandbreite des I- und Q-Signals begrenzt ist. Auf der Grundlage, dass die Kanalbandbreite das Doppelte der Bandbreite des I- oder Q-Signals ist, liegt die Durchlassbandbreite des Kanals im Bereich zwischen 1,5 MHz bis 28 MHz. Nach dem LPF befinden sich auf dem Tx-Zweig zwei variable Verstärker (TXVGA1 und TXVGA2). Die Verstärkungsfaktoren können bei beiden Verstärkern durch interne Register angepasst werden. Zwischen dem Verstärker TXVGA1 und TXVGA2 befindet sich der Tx-Mischer. Der Verstärker TXVGA1 liegt vor dem Mischer und hat die Aufgabe, die I- und Q-Basisbandsignale zu verstärken. Nach der Verstärkung der beiden Basisbandsignale geht es für die Signale weiter zu dem Mischer. Der Mischer vermischt die I- und Q-Signale mit den Ausgängen der Sende-PLL (TxPLL), damit das Signal in ein IQ-moduliertes HF-Signal umgewandelt werden kann. Am Ende des Tx-Pfades teilt sich das Signal auf zwei HF-Tx-Zweige auf. Der Verstärker TXVGA2, der als Leistungsverstärker (PA) verwendet wird, verstärkt erneut das HF-Signal. Als Ausgang stehen zwei HF-Ausgangs-Ports zur Verfügung (TXOUT1 und TXOUT2), jedoch ist immer nur ein Ausgang aktiv [3, 11].

Der Rx-Pfad beginnt mit drei verschiedenen HF-Eingangs-Ports (RXIN1, RXIN2, RXIN3). Die verschiedenen Eingänge haben unterschiedliche Frequenzbereiche, in denen sie das HF-Signal empfangen können. Der Eingang RXIN1 empfängt das Signal im Bereich zwischen den Werten 0,3 GHz bis 2,8 GHz. Beim zweiten Eingang RXIN2 wird das Signal zwischen 1,5 GHz und 3,8 GHz empfangen. Der dritte Eingangs-Port RXIN3 deckt den Frequenzbereich zwischen 0,3 GHz und 3,0 GHz ab. Danach wird das Signal zuerst durch einen der drei LNAs verstärkt, denn das empfangene Signal ist meist nur sehr schwach. Damit das IQ-modulierte HF-Signal in ein I- oder Q-Signal gewandelt werden kann, mischt der Mischer das HF-Signal mit der Ausgabe des RXPLL [3, 11].

Nach der Trennung der beiden Signale erfolgt dann die Verstärkung mit dem variablen Verstärker RXVGA1. Der Verstärker bringt das I- und Q-Basisbandsignal auf einen geeigneten Pegel. Im weiteren Verlauf passieren die I- und Q-Signale den LPF (RXLPF) [3].

Der Filter beschränkt die Bandbreite der beiden Signale, sodass die Signale, die innerhalb der ausgewählten Bandbreite liegen, weitergeleitet werden. Sollten die Signale außerhalb des gewünschten Bandbreitenbereiches liegen, werden diese Signale zurückgewiesen. Bei Kommunikationssystemen ist es normal, dass die Signalstärke sehr stark variieren kann. Diese Schwankungen sind für die Verwendung des ADC als sehr kritisch anzusehen. Für den Fall, dass die Signalstärke zu gering ist, könnte das Quantisierungsrauschen des ADC zum Überlagern des Signals führen. Sollte die Signalstärke zu groß sein, ist es möglich, dass der maximale Empfangspegel des ADC erreicht wird. Dies führt zur Sättigung des ADC. Zur Verhinderung dieser Situationen müssen die beiden Signale den nächsten Verstärker RXVGA2 durchlaufen. Der Verstärker begrenzt die I- und Q-Signalamplituden, damit diese innerhalb des Arbeitsbereiches des ADCs liegen. Ebenso wie der IQDAC besteht der IQADC auch aus einem 12-Bit-IADC und einem 12-Bit-QADC. Des Weiteren ermöglicht der IQADC Abtastraten zwischen 0,16 MHz und 40 MHz. Die digitalen I- und Q-Abtastwerte werden im Anschluss durch den Multiplexer abwechselnd an den 12-Bit-RXD (Rx-Datenbus) übergeben. Das Select-Flag-Bit teilt mit, ob ein I- oder ein Q-Abtastwert auf dem Datenbus zur Verfügung steht. Über den Rx-Datenbus gelangen die IQ-Abtastwerte weiter zum Signalprozessor [3, 11].

## 2.4 Programmable Synthesizer

Für die Überprüfung der blades kommt in den folgenden Versuchen ein Programmable Synthesizer HM8134 zum Einsatz. Dieser dient als Senderersatz. Der Synthesizer sendet ein bereits moduliertes Signal aus, das dann vom bladeRF empfangen wird. Weitere Informationen zum Umgang und zur Funktionsweise des Programmable Synthesizers sind in der Versuchsanleitung unter dem Punkt 5.3.5 zu finden.

## 3. Software

### 3.1 Allgemeine Informationen

Die Verwendung des bladeRFs erfordert eine dazugehörige Software. In diesem Fall kommt die Software von GNU Radio zum Einsatz. Die Software von GNU Radio ist eine kostenlose und freiverfügbare Entwicklungsumgebung zur Implementierung verschiedener Signalverarbeitungsprozesse, die in Funksystemen Anwendung finden. Die Simulationsumgebung bietet eine große Anzahl an Signalverarbeitungsblöcken, die es dem Benutzer ermöglichen, verschiedene SDR - Anwendungen zu erschaffen. Ebenso kommt die Software häufig in der Industrie und Forschung zum Einsatz. Diese soll zur Unterstützung der Erforschung zur drahtlosen Kommunikation und realen Funksystemen dienen. Die Software stellt viele Signalverarbeitungsblöcke, die auch in herkömmlichen Funksystemen zu finden sind, zur Verfügung. Dazu zählen Filter, Modulatoren und Demulatoren sowie Coder und Decoder. Für die Verwendung von GNU Radio wird dem Anwender eine grafische Benutzeroberfläche bereitgestellt. Diese erleichtert dem Benutzer die Entwicklung von verschiedenen Signalverarbeitungsprozessen mit einem Datenflussdiagramm. Diese Flussdiagramme werden in der GNU-Radio-Software mit der Programmiersprache Python ausgeführt. Generell sind für die Benutzung der Software keine Programmierkenntnisse nötig. Sollten aber spezielle Signalverarbeitungsblöcke für die Implementierung benötigt werden, können nur unter Anwendung der Programmiersprache neue Bausteine erstellt werden [3, 12].

Für die Arbeit mit dem bladeRF stellt die Software einen Schnittstellenblock zur Verfügung. Der am häufigsten verwendete Schnittstellenblock ist der nebenstehende, in der Abbildung 6

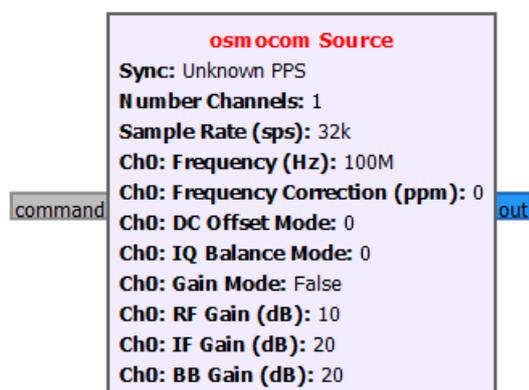


Abbildung 6 osmocom-Block

dargestellte osmocom-Baustein. Diesen Baustein gibt es jeweils als Quelle und als Senke. Somit sind das Senden und das Empfangen von Signalen möglich. Für die individuelle Anpassung können die verschiedenen Parameter im Block verändert werden. Neben der Abtastrate und der Channel-Frequenz sind auch Verstärkungs- und Korrekturfaktoren einstellbar. Bei Unklarheiten über die Funktion des Parameters bietet jeder Baustein eine Beschreibung zu den unterschiedlichen

Parametern an. Für die verschiedenen SDR-Anwendungen können neben den hier verwendeten bladeRF auch andere Hardwaregeräte Anwendung finden. Für die Verbindung der Hardware mit dem osmocom-Baustein muss das verwendete Gerät im Block angegeben werden.

## 3.2 Aufbau der Plattform

Dem Anwender wird für seine Implementierung eine grafische Benutzeroberfläche zur Verfügung gestellt. Die Software hat für die Realisierung der wichtigsten Schaltungen verschiedene Bausteine entwickelt. In der Abbildung 7 ist die Plattform, wie sie dem Benutzer angezeigt wird, dargestellt.

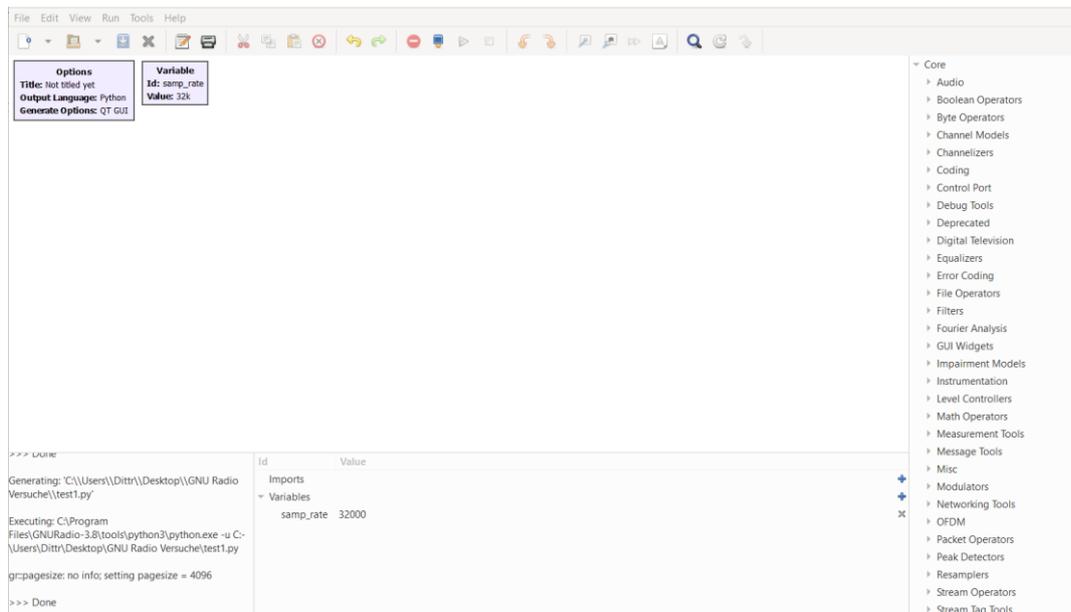


Abbildung 7 Benutzeroberfläche von GNU Radio

Bei dem Aufbau ist festzustellen, dass die Plattform in vier große Bereiche unterteilt ist. Der größte Bereich, in dem auch die beiden Blöcke „Optionen“ und „Variable“ bereits eingefügt sind, ist der Arbeitsbereich. In diesem Teil der Anwendung kann der Benutzer seine verschiedenen Signalverarbeitungsblöcke einfügen. Die Auswahl der von GNU Radio bereitgestellten Signalverarbeitungsbausteine befindet sich auf der rechten Seite. Um die vielen Bausteine grob zu ordnen, werden diese unter einem Oberbegriff zusammengefasst. Das heißt, dass zum Beispiel bei den Modulatoren verschiedene Blöcke für die Modulation und Demodulation von frequenzmodulierten und amplitudenmodulierten Signalen eingeordnet sind. Für eine schnelle Suche eignet sich auch die Suchfunktion aus der Menüleiste. Unter dem Symbol mit der Lupe können verschiedene Begriffe eingegeben werden. Die Menüleiste beinhaltet neben der Lupe noch weitere Symbole. Damit ist es möglich, dass neue Projekte erstellt oder bereits bestehende Projekte geöffnet werden. Weitere Funktionen sind das Ausschneiden, Kopieren und Einfügen von erstellten Schaltungen. Die beiden Pfeile ermöglichen es dem Benutzer, eine bestimmte Anzahl von Aktionen rückgängig zu machen. Nicht zu verwechseln sind diese mit den Pfeilen, die nach unten zeigen. Diese sind für die Rotation der Signalblöcke zuständig.

Für das Starten und das Beenden der Simulation befindet sich zwischen den beiden Pfeilsymbolen ein Play- und ein Stoppsymbol. Zudem können in der Menüleiste die eingefügten Signalverarbeitungsblöcke aktiviert und deaktiviert werden. Diese Funktion ist genauso über eine Tastatureingabe möglich. Die letzten beiden Bereiche befinden sich in Abbildung 7 unten. Im linken Bereich kann vom Benutzer die Ausführung des Programms beobachtet werden. Sollte bei der Kompilierung des Programms ein Fehler auftreten, wird dieser in diesem Fenster angezeigt. In dem zweiten Fenster sind die verwendeten Variablen aufgelistet. Je nach Initialisierung der Variable wird in diesem Bereich der Name (Id) und der Wert der erstellten Variablen angegeben.

Für die Dokumentation der Messergebnisse können die Werte in verschiedenen Diagrammen ausgegeben werden. Bei der Ausführung der Programme öffnet GNU Radio dann ein weiteres Fenster zur Darstellung der Messwerte in den unterschiedlichen Diagrammtypen. Zum Ablesen des genauen Messwertes reicht es, den Cursor auf den Diagrammverlauf zu schieben. Ebenso kann für die Anpassung von verschiedenen Werten Schieberegler erstellt werden. Dadurch ist es möglich, die Parameter während der laufenden Simulation zu ändern.

### 3.3 Erste Schritte mit GNU Radio

Nachgehend wird ein erstes Projekt mit der GNU-Radio-Software erstellt. Nach erfolgreicher Installation der Software auf dem Computer kann das Programm gestartet werden. Zum Erstellen eines neuen Projektes muss ein Klick auf „File“ und dann auf „New“ erfolgen. Eine weitere Möglichkeit zur Erstellung eines neuen Projektes ist das Klicken auf das erste Symbol in der Menüleiste. Somit ist die Entwicklungsumgebung zum Aufbau der Schaltung geschaffen. Standardmäßig sind zwei Bausteine im Arbeitsfeld eingefügt. Der Baustein „Optionen“ enthält die Angaben zum Namen des Projektes. Ebenso ist ein Baustein mit einer Variablen enthalten. Diese Variableneinstellung kann später angepasst oder entfernt werden.

Für die folgende Schaltung sollen drei Signalquellen miteinander verbunden und dann eine Ausgabe der Signale erfolgen. Für die Auswahl der benötigten Bausteine gibt es zwei Möglichkeiten. Zum einen die Schnellsuche, die sich in der Menüleiste befindet oder die Auswahl der Bausteine im rechten Fenster. Im Auswahlfeld gibt es den Oberbegriff „Waveform Generators“. Unter diesem Punkt befinden sich mehrere verschiedene Signalquellen zur Auswahl. In diesem Versuch soll ein Kosinus-Signal übertragen werden. Dafür ist die herkömmliche „Signal Source“ gut geeignet. Für die Kombination der Signale müssen diese miteinander „addiert“ werden. Auch dafür bietet die Software von GNU Radio einen Baustein. Der Unterpunkt „Math Operator“ beinhaltet einen „Add“-Baustein. Die Ausgabe erfolgt über mehrere verschiedenen Senken, die verschiedene Darstellungsweisen verwenden.

Die Darstellung des Signals erfolgt mit einer „Time Sink“, einer „Frequency Sink“ und einer „Waterfall Sink“. Diese Bausteine sind unter dem Punkt „Instrumentation“ zu finden. Der Punkt ist in zwei Unterpunkte gegliedert. Für die Auswahl der Senken muss der Unterpunkt „QT“ ausgewählt werden.

Diese Elemente müssen zur Realisierung der Schaltung ausgewählt und in das Arbeitsfeld eingefügt werden. Die Verbindung erfolgt durch das Klicken und Ziehen der Verbindung zwischen den beiden Bausteinen. Nach Fertigstellung des Schaltungsaufbaus kann die Anpassung der Parameter erfolgen.

Die verschiedenen Parameter sind durch den Benutzer im Block einsehbar. In dem Block der „Signal Source“ sind die Parameter Sample Rate, Waveform, Frequency, Amplitude, Offset

und Initial Phase (Radians) enthalten, wie in Abbildung 8 aufgezeigt wird. Die Sample Rate erhält in diesem Fall einen Wert von 32 kps. Daher, dass ein Kosinus-Signal übertragen werden soll, wird die Waveform auf „Cosine“ gestellt. Die Amplitude erhält den Wert 1 und die Parameter Offset und Initial-Phase bleiben auf 0. Die Signalquellen erhalten für den Wert der Frequenz unterschiedliche Initialisierungen. Die erste Signalquelle bekommt eine Frequenz von 1 kHz (Kilo-Hertz) zugewiesen, die zweite 2 kHz und die dritte 3 kHz. Der Wert der Abtastrate wird in dem vorliegenden Beispiel durch eine Variable repräsentiert. Diese wird in das Feld des gewünschten Parameters eingetragen. Aus diesen Vorbetrachtungen ergibt sich die Schaltung aus Abbildung 9.

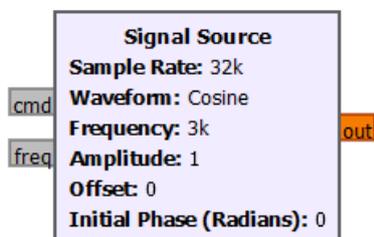


Abbildung 8 Signal-Source

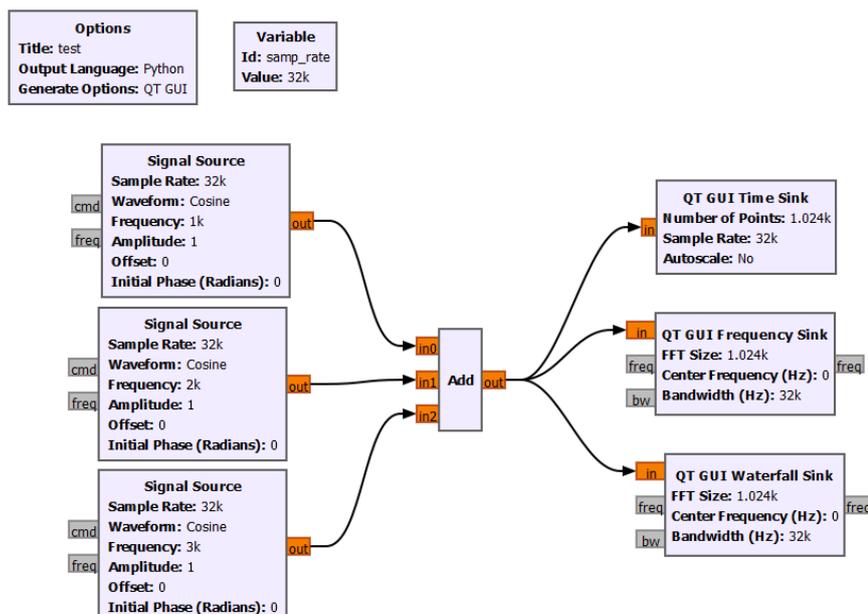


Abbildung 9 Flussdiagramm der Testschaltung

Wie in der Abbildung 9 zu sehen ist, werden die drei Signalquellen miteinander kombiniert und das Ergebnis mithilfe von verschiedenen Diagrammtypen ausgegeben. Zum Starten der Simulation muss ein Klick auf das Play-Symbol in der Menüleiste erfolgen. Zu jedem kompilierten Programm wird neben der normalen Datei eine zweite Datei mit dem Python-Code ausgegeben. Die Ausgabe des Codes ist mit unterschiedlichen Programmen möglich. Zudem kann die Veränderung des Programmcodes erfolgen. Sollte das Play-Symbol ausgegraut sein, liegt ein Fehler vor. Dann ist es wichtig, die Schaltung auf Fehler zu überprüfen. Diese Fehler können in der Verbindung zwischen den Signalverarbeitungsblöcken liegen, oder die Bausteine sind vom Benutzer falsch initialisiert. Im Fehlerfall sind die Namen der Bausteine rot eingefärbt. Somit ist es dem Benutzer möglich, den fehlerhaften Block schneller zu identifizieren. Sobald die Simulation gestartet wird, öffnet sich nach einer gewissen Verzögerungszeit ein neues Fenster. In diesem Fenster sind die Diagramme dargestellt. Im vorliegenden Beispiel werden die folgenden Diagramme aus Abbildung 10 ausgegeben.

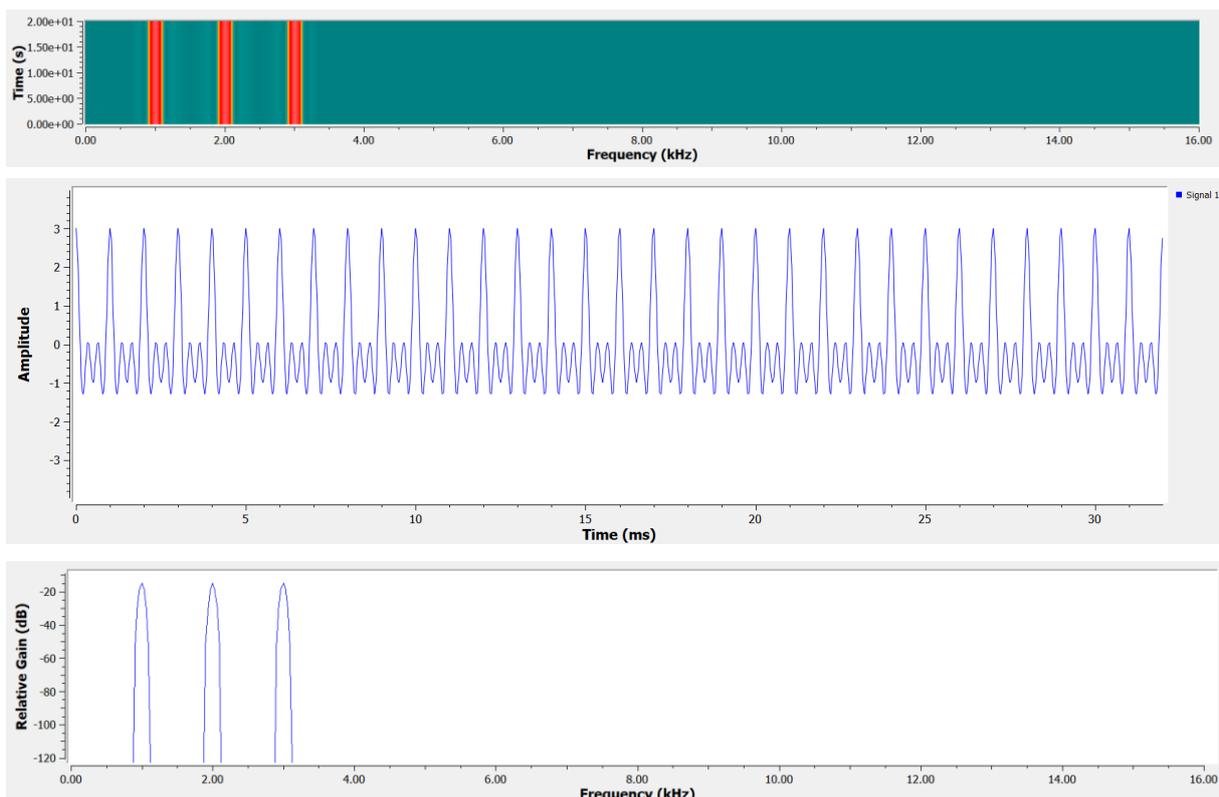


Abbildung 10 Ergebnisse der Testschaltung

Die Darstellung in den Diagrammen dient zur Dokumentation der Messergebnisse. Das erste und das letzte Diagramm zeigt jeweils einen Frequenzverlauf. Für die drei unterschiedlichen Signale werden jeweils die drei initialisierten Frequenzwerte angezeigt. Das mittlere Diagramm präsentiert den Amplitudenverlauf. Aufgrund der Addition der drei Signale entsteht eine Amplitudenhöhe mit dem Wert 3. Nach der Ausgabe der Werte ist das erste Projekt mit der GNU-Radio-Software abgeschlossen.

## 4. Signalübertragung mittels bladeRF

In den folgenden Versuchen wird eine Kommunikation zwischen zwei bladeRFs hergestellt. Die Ausführung und Dokumentation der Ergebnisse bezieht sich ausschließlich auf den bladeRF, der das Signal empfängt. Die Dokumentation der Senderseite finden Sie unter [1].

### 4.1 Frequenzmodulation

Zur Übertragung der Signale werden unterschiedliche Modulationsverfahren verwendet. Für den ersten Versuch wird zur Signalübertragung eine analoge Frequenzmodulation eingesetzt. Das Ziel ist es, ein frequenzmoduliertes Audiosignal zu empfangen und die Tonspur am Zielrechner ausgeben zu lassen.

Bei der FM (Frequenzmodulation) wird das zu übertragende Signal durch ein hochfrequentes Trägersignal modifiziert. Charakteristisch für FM ist, dass die Amplitude gleich bleibt. Die in Abbildung 11 dargestellten Signalverläufe zeigen das Verfahren der Frequenzmodulation. Das NF-Signal (Niederfrequentes Signal) ist das Signal, das die Information enthält. Um eine hochfrequente Übertragung zu ermöglichen, wird das niederfrequente Nutzsinal durch das hochfrequente Trägersignal verändert. Das Resultat ist das FM-Signal. Die Frequenz des FM-Signals variiert je nach Übertragungswert des Nutzsinals [13].

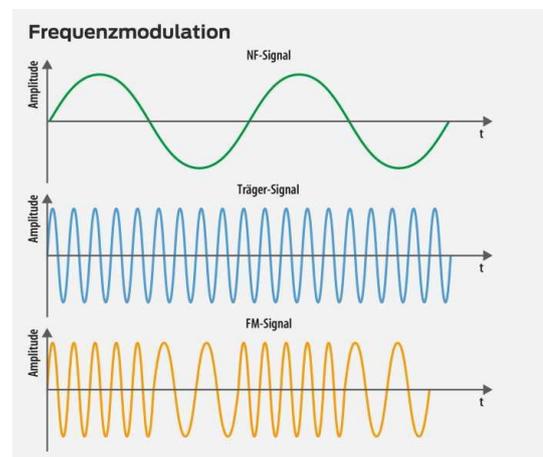


Abbildung 11 Vorgang der FM [23]

Für die Durchführung des Versuches bekommt der Empfänger ein frequenzmoduliertes Signal vom Sender übertragen. Die Aufgabe des Empfängers besteht darin, das empfangene Signal zu demodulieren, damit der Empfänger das ursprüngliche Signal korrekt ausgeben kann. Zu Beginn des Versuches wird überprüft, ob der bladeRF ordnungsgemäß funktioniert. Dies wird mithilfe eines Programmable Synthesizer HM8134 realisiert. Der Synthesizer überträgt an das Funkmodul ein frequenzmoduliertes 1 kHz- Ton. Somit kann überprüft werden, ob der bladeRF das empfangene Signal ordnungsgemäß demoduliert und den Ton am Ende gut hörbar ausgibt. Dabei ist es wichtig, dass am Synthesizer die richtigen Einstellungen vorgenommen werden. Der Synthesizer kann neben der Frequenzmodulation auch amplituden- und phasenmodulierte Signale aussenden. Des Weiteren muss die richtige Channel-Frequenz eingegeben werden, damit die Kommunikation zwischen Synthesizer und bladeRF nicht gestört wird.

Schon kleine Abweichungen bei der Einstellung der Frequenz kann dazu führen, dass kein Signal empfangen werden kann. Wenn am Synthesizer die korrekten Werte eingestellt sind, kann eine Verbindung zum bladeRF mittels eines Verbindungskabels hergestellt werden. Das bladeRF ist mit dem Ausgaberechner verbunden, auf dem die Software GNU Radio installiert ist. Für die Durchführung des Tests wurde die folgende Schaltung aus Abbildung 12 verwendet.

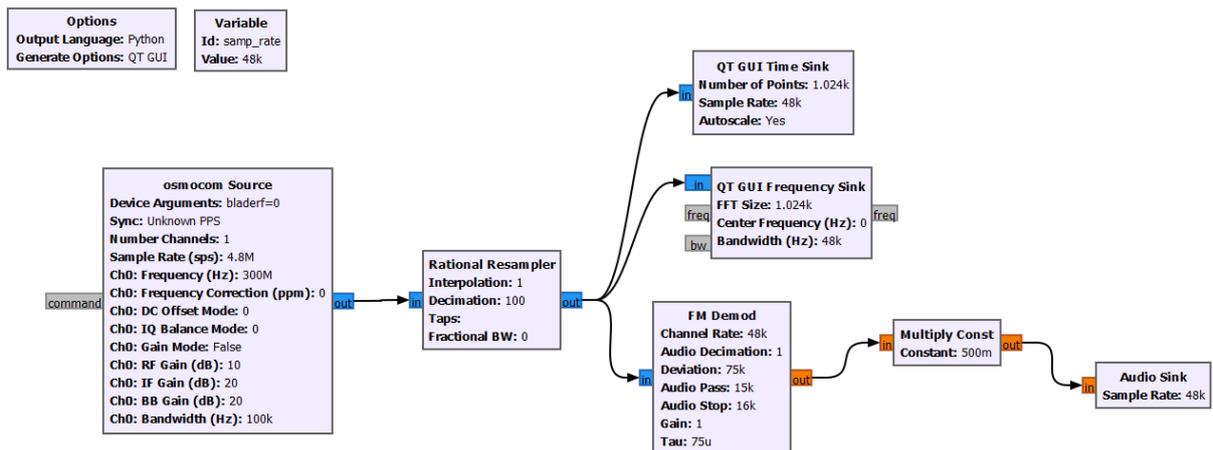


Abbildung 12 Flussdiagramm für FM-Testversuch

Die Schaltung beinhaltet verschiedene Blöcke, die von GNU Radio zur Verfügung gestellt werden. Dabei stellt die osmocom Source den angeschlossenen bladeRF dar. Die wichtigsten Einstellungen sind die Abtastrate mit 4,8 Msps und die Channel-Frequenz, die auch vom Synthesizer verwendet wird, mit 300 MHz. Der Rational Resampler wird mit dem Dezimationswert von 100 eingestellt, da bei dem FM-Demodulator und der Audiosenke am Ende der Schaltung die Abtastrate bei nur noch bei 48 ksps liegen soll. Die Ausgabe des Signals erfolgt mit zwei Grafiken. Der Zeitverlauf und Frequenzverlauf des Signals werden in Abbildung 13 dargestellt.

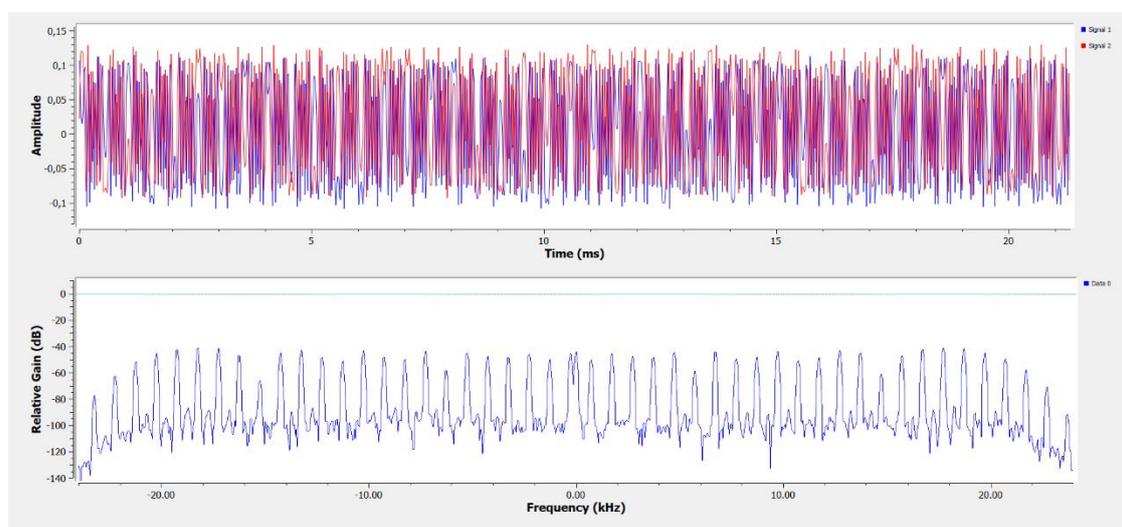


Abbildung 13 Messwerte für FM-Testversuch

Der Frequenzverlauf des Signals zeigt in 1 kHz-Abständen einen Ausschlag an. Das bedeutet, dass die Übertragung der Tonspur erfolgreich war. Bei der Ausführung des Tests sind dennoch Probleme aufgetreten. Die Übertragung des Audiosignals wurde durch Verzerrungen gestört. Für die reguläre Ausgabe des Tons wurde zwischen den Synthesizer und den bladeRF ein Dämpfungsglied geschaltet. Das Dämpfungsglied konnte das Signals so formen, dass der übertragene 1 kHz-Ton in guter Qualität am Zielrechner ausgegeben werden konnte.

Für die Demodulation des Signals kommt der FM-Demod-Block aus Abbildung 14 zum Einsatz. Dieser Block beinhaltet neben der Demodulation des Signals gleichzeitig eine Tiefpassfilterung. Für die Parameteranpassung wurde die Channel-Rate auf den Wert der Abtastrate gesetzt. Die Dezimation erfolgte bereits durch den Rational Resampler. Somit kann der Wert im Demod-Block auf 1 gesetzt werden. Die Initialisierung der Passbandfrequenz und der Stoppfrequenz erfolgt mit den Werten 15 kHz und 16 kHz. Diese Angaben legen den Frequenzbereich fest, indem der integrierte Audio-Tiefpassfilter arbeitet. Die Parameter Gain und Tau sind auf ihre Standardwerte eingestellt.

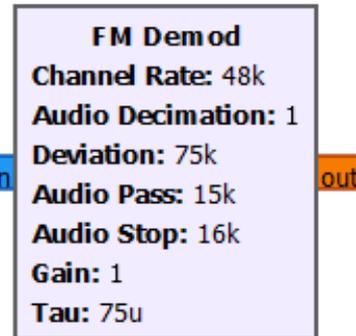


Abbildung 14 FM-Demod-Block

Nach der erfolgreichen Übertragung des 1-kHz-Tons, kann mit der Übertragung eines richtigen Audiosignals fortgefahren werden. Das Audiosignal wird vom entsprechenden Sender bladeRF übertragen [1]. Die GNU-Radio-Software bietet zwei verschiedene FM-Module zur Implementierung der FM-Modulation. In Abbildung 15 werden die beiden verschiedenen FM-Blöcke dargestellt. Abhängig vom zu übertragenden Audiosignal ist eine Auswahl zwischen Schmalband-FM (Narrowband) und Breitband-FM (Wideband) möglich. Schmalband-FM hat nur eine geringe Bandbreite. Im Gegensatz dazu ist bei Breitband-FM die Bandbreite nicht begrenzt. Ein weiterer Unterschied zwischen den beiden Verfahren ist der Frequenzbereich der Modulation. Schmalband-FM deckt nur den Bereich von 30 Hz bis 3 kHz ab. Breitband-FM beginnt ebenfalls bei 30 Hz, deckt jedoch ein Spektrum bis zu 15 kHz ab. Es gibt auch Unterschiede in der Frequenzabweichung. Die maximale Frequenzabweichung für Schmalband-FM beträgt etwa 5 kHz, während die Frequenzabweichung für Breitband-FM-Verfahren 75 kHz beträgt [14].

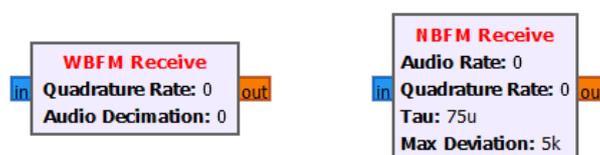


Abbildung 15 WBFM- und NBFM-Baustein

Für die Implementierung der Frequenzmodulation wurde sich für das Verfahren der Breitband-FM entschieden. Der Aufbau der Schaltung zur Frequenzmodulation wird in Abbildung 16 veranschaulicht.

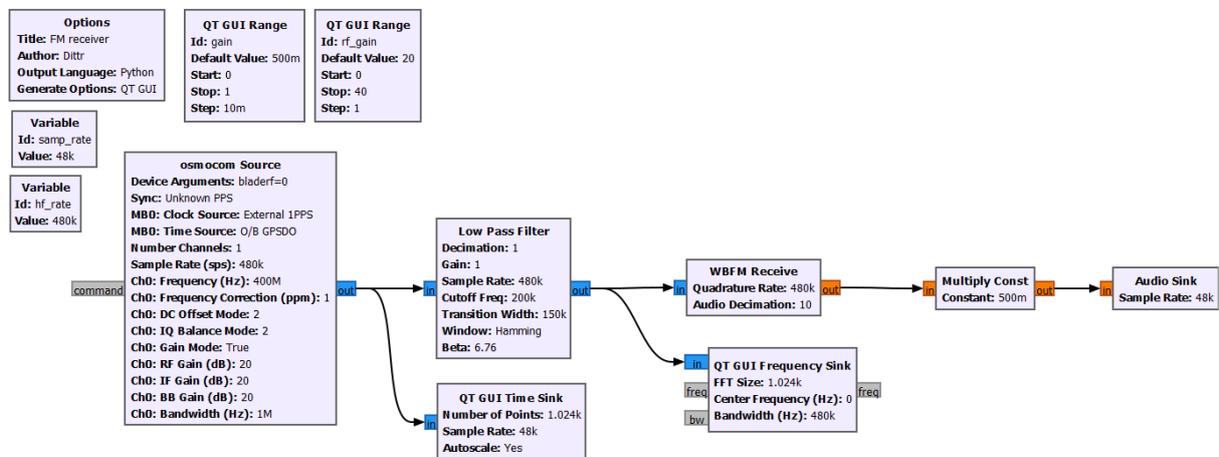


Abbildung 16 Flussdiagramm für FM-Versuch

Der bladeRF arbeitet in einem Frequenzbereich zwischen 300 MHz und 3,8 GHz. Damit der bladeRF nicht im unteren Grenzbereich arbeitet, wurde die Channel-Frequenz auf 400 MHz erhöht. Daher, dass im WBFM (Wideband FM) -Receive-Block kein integrierter Tiefpassfilter vorhanden ist, muss dieser vor den Demodulationsblock geschaltet werden. In der vorliegenden Schaltung ist die Abtastrate auf 480 kps eingestellt. Die Audio-Dezimation wird vom WBFM-Block vorgenommen, damit die Abtastrate an der Audiosenke bei 48 kps erreicht werden kann. Für die Dokumentierung der Messergebnisse ist nach der osmocom Source eine Zeitsenke eingebaut und für die Aufzeichnung des Frequenzverlaufs eine Frequenzsenke nach der Filterung.

Bei den ersten Versuchsdurchläufen war die Qualität des Audiosignals sehr schlecht. Das Signal war nur leise zu hören und es wurde durch Verzerrungen gestört. Die erste Problemlösung bestand darin, für das nur schwache und leise zu hörende Signal einen Verstärker zwischen die beiden bladeRFs zu schalten. Die Regulierung der Signalstärke erfolgte durch ein Dämpfungsglied. Mittels dieser zusätzlichen Bauelemente konnte eine Verbesserung der Signalübertragung erzielt werden. Ein weiterer Ansatz zur Verbesserung des Signals ist, die Parameter der einzelnen Module in der GNU-Radio-Software zu verändern. Zur besseren Bewertung der Parameteränderung wurden die zusätzlichen Bauelemente, der Verstärker und das Dämpfungsglied aus der Schaltung entfernt. Veränderungen an der Tiefpassfilterung und am WBFM-Block brachten keine Verbesserung der Signalqualität. Zudem wurden in der osmocom Source drei Parameter verstellt, die ebenfalls keine Verbesserung der Qualität des Signals erreichen konnten. Zu diesen Parametern zählt der Frequenzkorrekturfaktor, der DC (Direct Current) - Mode und der IQ-Balance-Mode.

Nachdem die Neuinitialisierung dieser Parameter nicht erfolgreich war, sind die Gain-Parameter verändert wurden. Die osmocomblock Source stellt insgesamt drei Gain-Parameter zur Verfügung. Eine Erhöhung der Gain-Parameter führt dazu, dass sich die Verstärkung des empfangenen Signals erhöht. Dabei ist zu beachten, dass im Umgang mit dem bladeRF nur ein Gain-Parameter eine Auswirkung auf die Verstärkung des Signals hat. Die in Abbildung 17 dargestellte Beschreibung der verschiedenen Parameter zeigt auf, dass die anderen beiden Werte nur Auswirkungen haben, wenn andere Hardwaremodule am Zielrechner angeschlossen sind. Um diese Beschreibung zu finden, muss auf den osmocomblock-Source-Block doppelt geklickt und anschließend der Unterpunkt „Documentation“ ausgewählt werden [15].

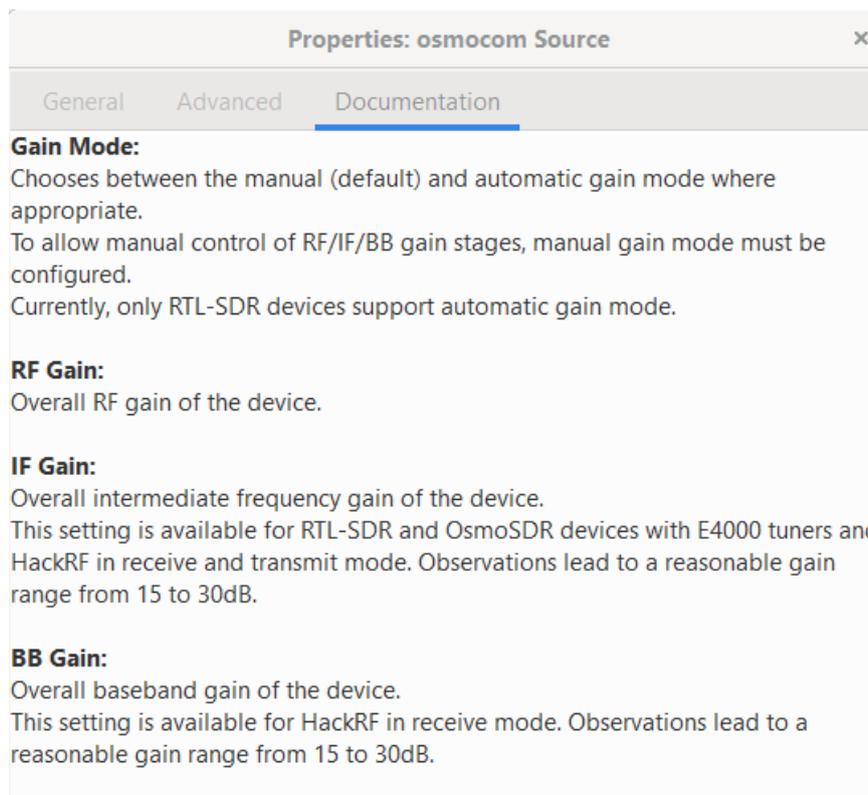


Abbildung 17 Parametererklärung für osmocomblock-Block

Damit das Signal verstärkt werden kann, reicht es aus, den RF-Gain in der osmocomblock Source neu zu initialisieren. Natürlich muss je nach verwendetem Audiosignal die Verstärkung individuell angepasst werden. In diesem Fall ist es zu empfehlen, dass eine Variable initialisiert wird. Im vorliegenden Beispiel ist ebenfalls eine Variable verwendet worden. Für die Erstellung der Variable gibt die GNU-Radio-Software bestimmte Werte vor, die eingetragen werden müssen. Dabei wird nach Start-, Stopp- und Standardwert gefragt. Zudem kann eine Schrittweite der Variablengröße festgelegt werden. In der nebenstehenden Abbildung 18 wird die Variableninitialisierung aus dem vorliegenden Beispiel gezeigt.

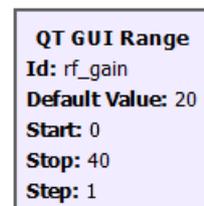


Abbildung 18 GUI-Range Block

Der Name der Variable lautet `rf_gain`. Die Auswahl des Verstärkungswertes liegt zwischen 0 und 40 db (Dezibel). Der Standardwert liegt bei 20 db mit einer Schrittweite von 1. Der Vorteil dabei ist, dass diese Variable bei der Ausgabe der Messergebnisse mit angezeigt wird. Damit kann während der laufenden Übertragung des Signals die Verstärkung mittels eines Schiebereglers individuell eingestellt werden.

Mit der Veränderung des RF-Gain konnte eine sichtliche Verbesserung des Empfangssignals festgestellt werden. Die größere Verstärkung bewirkte eine deutliche Aufwertung der Signalqualität. Somit konnte die Übertragung des Audiosignals erfolgreich unter Anwendung der Frequenzmodulation übertragen werden. Die aufgenommenen Messwerte werden in Abbildung 19 visualisiert.

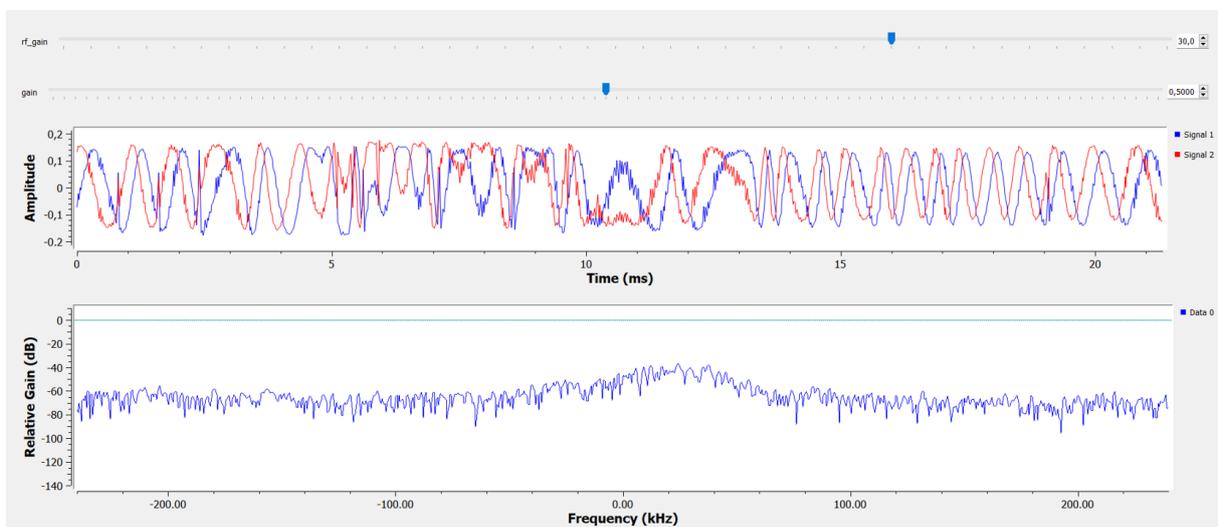


Abbildung 19 Messergebnisse für FM-Versuch

Das Resultat zeigt zum einem den Frequenzverlauf nach der Tiefpassfilterung und es wird das modulierte Signal im Zeitverlauf dargestellt. Im oberen Teil der Abbildung ist zu erkennen, dass der bereits beschriebene Schieberegler für den RF-Gain mit integriert ist und dass ein weiterer Schieberegler zur Anpassung der Lautstärke verwendet wurde. Es ist gut zu erkennen, dass im Zeitverlauf die Amplitude fast immer gleich bleibt. Zudem gibt es eine Frequenzänderung, je nachdem, welche Information vom Nutzsignal übertragen wurde. Trotz der verschiedenen Problemlösungsansätze und Verbesserungen der Signalqualität sind immer noch Störungen im Diagrammverlauf zu erkennen. Die sporadischen Störeinflüsse können mit der Verkabelung der beiden bladeRFs zusammenhängen. Ebenso ist es möglich, dass das ausgewählte Audiosignal durch das Herunterladen und Konvertieren eine Verschlechterung der Signalqualität zur Folge hat.

## 4.2 Amplitudenmodulation

Im zweiten Versuch wird eine AM (Amplitudenmodulation) verwendet. Wie auch bei der Frequenzmodulation soll bei diesem Experiment ein Audiosignal durch die Amplitudenmodulation moduliert werden und beim Empfänger wieder demoduliert werden. Die Tonspur wird am Zielrechner ausgegeben.

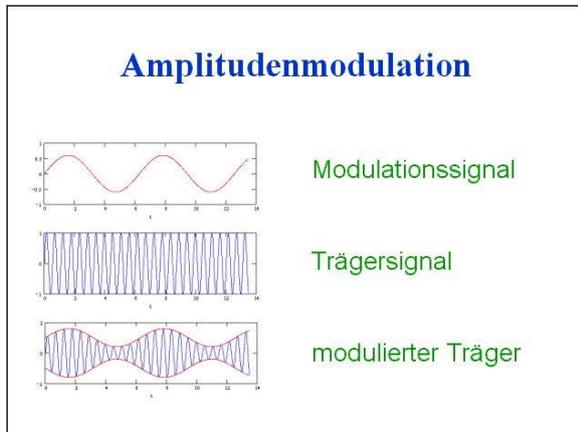


Abbildung 20 Vorgang der AM [24]

Bei der Amplitudenmodulation wird ein niederfrequentes Nutzsingal auf ein hochfrequentes Trägersingal moduliert. Die nebenstehende Abbildung 20 zeigt das Modulationssingal, das Trägersingal und das daraus resultierende modulierte Singal. Die Informationsübertragung wird durch die Veränderung der Amplitude im modulierten Singal realisiert. Je nachdem welchen Pegel und Frequenz das Nutzsingal aufweist, ändert

sich die Amplitude des resultierenden Singals. Ein weiterer wichtiger Aspekt der AM ist, dass bei der Modulation die Frequenz und Phase des Singals erhalten bleiben. Im Vergleich zur Frequenzmodulation ist die Amplitudenmodulation stör anfälliger. Die Ursache dafür ist, dass bei der AM die Information in der Amplitude steckt. Sollte die Amplitude des Singals beeinflusst werden, ist es möglich, dass die Information des Singals weg ist oder verfälscht wird. Die Störungen lassen sich durch Verzerrungen und durch eine schlechte Übertragungsqualität identifizieren [16].

Zu Beginn des Versuchs wird auch bei dieser Versuchsdurchführung vorher ein Testlauf gestartet. Dabei kommt wieder der Programmable Synthesizer HM8134 zum Einsatz. Dafür muss die Einstellung am Synthesizer auf Amplitudenmodulation gestellt werden. Wie auch im vorherigen Versuch wurde sich dazu entschieden, ein 1 kHz- Ton zu übertragen. Für den Testversuch wird in diesem Fall fast dieselbe Schaltung verwendet wie für die spätere Amplitudenmodulation mit dem Audiosingal. Die GNU-Radio-Software stellt für die Demodulation einen AM-Demod-Block zur Verfügung. Der AM-Demod-Block kann mit Beispieleinträgen aus dem vorliegenden Versuch in Abbildung 21 eingesehen werden. Der Block besitzt neben seinen Demodulationseigenschaften auch noch eine Audiofilterung. Somit ist in diesem Baustein bereits ein Audio-Tiefpassfilter integriert. Neben der Channel-Rate und der Audio-Dezimation kann die Passbandfrequenz und die Stoppfrequenz des Audio-Tiefpassfilters angegeben werden.

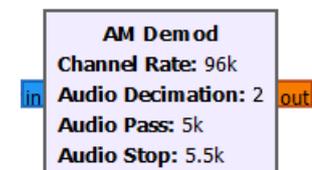


Abbildung 21 AM-Demod-Block

Zusätzlich zur Audio-Filterung wird vor dem AM-Demod-Block ein weiterer Filter in die Schaltung eingesetzt. Dabei handelt es sich um einen Bandpassfilter. Dieser kann beim Testlauf weggelassen werden. In Abbildung 22 ist der Bandpassfilter-Baustein aus der GNU-Radio-Software dargestellt. Dieser wird dafür benötigt, damit bestimmte Frequenzen aus bestimmten Frequenzbereichen isoliert werden können und um die Frequenzen, die im

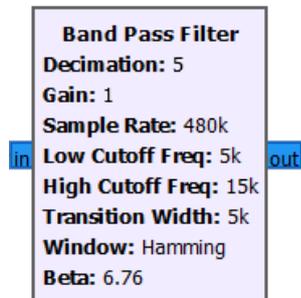


Abbildung 22 Band Pass Filter- Baustein

gewünschten Frequenzbereich liegen, herauszufiltern. Der Bandpassfilter bietet für die exakte Anpassung an das jeweilige Projekt verschiedene Parameter zu Initialisierung an. Neben der Dezimation und Verstärkung des Filters kann der untere und obere Grenzwert des Bandpasses angegeben werden. In diesem Fall liegt der Wert zwischen 5 kHz und 15 kHz, weil die Trägerfrequenz, bei der Übertragung des Audiosignals, vom Sender auf 10 kHz eingestellt wurde [1]. Für den Übergang zwischen Bandpass und Stoppfrequenz

wird eine Übergangsbreite bestimmt. Die Parameter Window und Beta sind für die Ausführung dieses Versuches nicht relevant und werden deshalb auf ihren Standardeinstellungen gelassen. Auf der Grundlage dieser Tatsachen wurde für die Durchführung des Versuchs der Amplitudenmodulation die folgende Schaltung aus Abbildung 23 entwickelt.

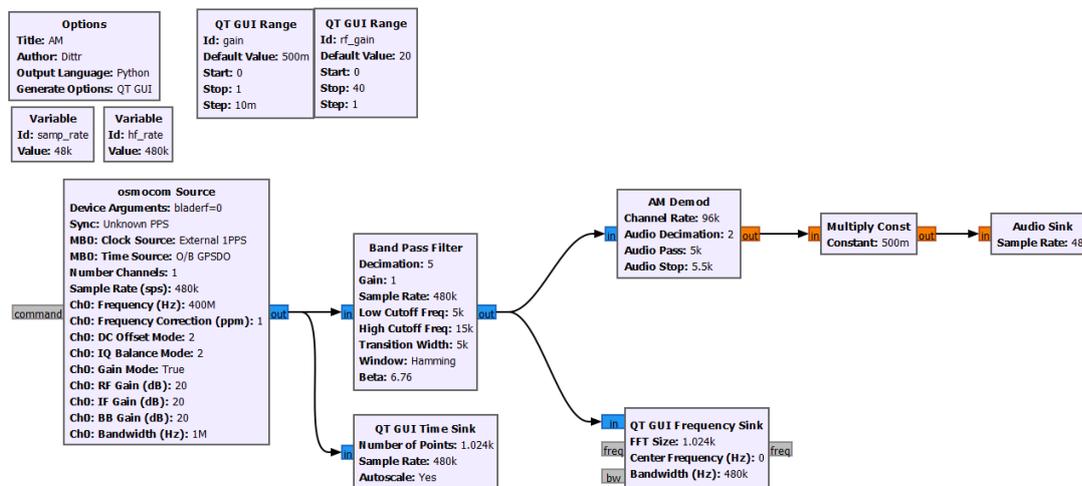


Abbildung 23 Flussdiagramm für AM-Versuch

In der Abbildung ist zu sehen, dass das Signal, nachdem es vom bladeRF empfangen wurde, zunächst den Bandpassfilter durchlaufen muss. Vor der Filterung wird der zeitliche Verlauf des Signals aufgenommen. Der Frequenzverlauf wird nach der Filterung durch den Bandpass in einem Diagramm festgehalten. Sobald das Signal vom Bandpassfilter herausgefiltert wurde, kann es den AM-Demodulator passieren. Im AM-Demodulator erfolgt dann die zweite Filterung des Signals. Hat das Signal diese Vorgänge durchlaufen, wird es an der Audiosenke ausgegeben. Die Einstellungen bei der osmocomb Source sind fast identisch mit den Einstellungen bei der Frequenzmodulation. Die Abtastrate wurde von 48 kps auf 480 kps erhöht.

Damit am Ende an der Audiosenke eine Abtastrate von 48 kps erreicht werden kann, wird die Abtastrate am Bandpassfilter das erste Mal dezimiert. Somit kommt beim AM-Demodulator nur noch eine Abtastrate von 96 kps an. Die zweite Dezimation findet dann im AM-Demodulator statt. Dabei wird der Wert um die Hälfte dezimiert, um an der Audiosenke eine Abtastrate von 48 kps realisieren zu können.

Für die Übertragung des Signals wurde das bladeRF zunächst an den Programmable Synthesizer HM8134 angeschlossen. Dieser überträgt das amplitudenmodulierte Signal an den Empfänger-bladeRF. Am Zielrechner wird das empfangene Signal aufgezeichnet, demoduliert und durch die Audiosenke ausgegeben. Die nachfolgende Abbildung 24 zeigt die Messergebnisse für die Übertragung des 1 kHz- Signals.

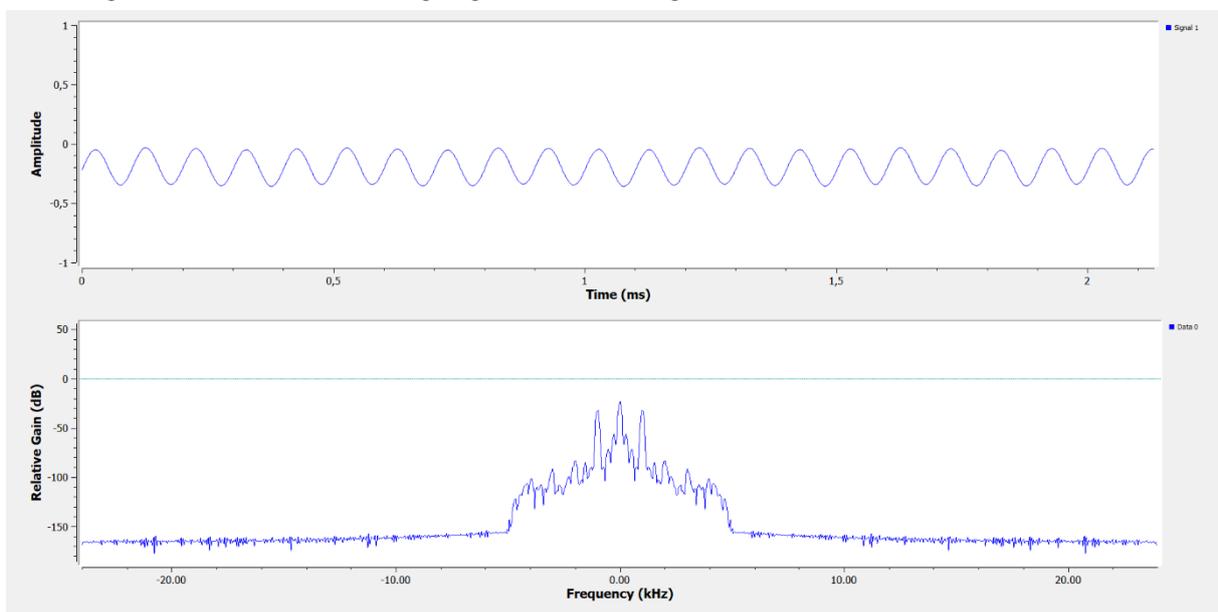


Abbildung 24 Messergebnisse für AM-Testversuch

Für die Übertragung zeigt sich im oberen Diagramm eine konstante Amplitude. Die Ursache dafür ist, dass immer wieder das gleiche Signal übertragen wird. Somit entstehen keine sichtbaren Veränderungen in der Amplitude. In der Aufzeichnung des Frequenzverlaufes ist gut zu erkennen, dass nur bestimmte Frequenzbereiche vom AM-Demod-Block durchgelassen werden. In den Frequenzbereichen, in denen der Demodulations-Block die Frequenzen passieren lässt, ist gut zu erkennen, dass die Übertragung des 1 kHz- Tons erfolgreich war. Bei auftretenden Problemen ist es sinnvoll, die Parameter der Signalverarbeitungsblöcke zu überprüfen. Ebenso sollte eine ausreichende Verstärkung des Signals durch den bladeRF gewährleistet werden.

Nachdem der Testlauf für die Versuchsdurchführung der Amplitudenmodulation erfolgreich war, kann mit der Übertragung des Audiosignals fortgefahren werden. Der Empfänger-bladeRF wird mit den mit dem Sender-bladeRF verbunden [1]. Die Versuchsdurchführung wird mit demselben Schaltungsaufbau durchgeführt.

Zu Beginn der Übertragung wurde das Signal am Empfänger nur sehr schlecht ausgegeben. Für die Problemlösung ist auch bei diesem Versuch ein Verstärker und ein Dämpfungsglied zum Einsatz gekommen. Mit diesen Bauelementen konnte keine Verbesserung der Signalqualität festgestellt werden. Der zweite Ansatz war es, die Parameter der Softwarebausteine zu verändern. Mit der Veränderung der Filterparameter konnte eine bessere Signalqualität erreicht werden. Für die Übertragung des Audiosignals ergaben sich folgende Messwerte, die in Abbildung 25 dokumentiert sind.

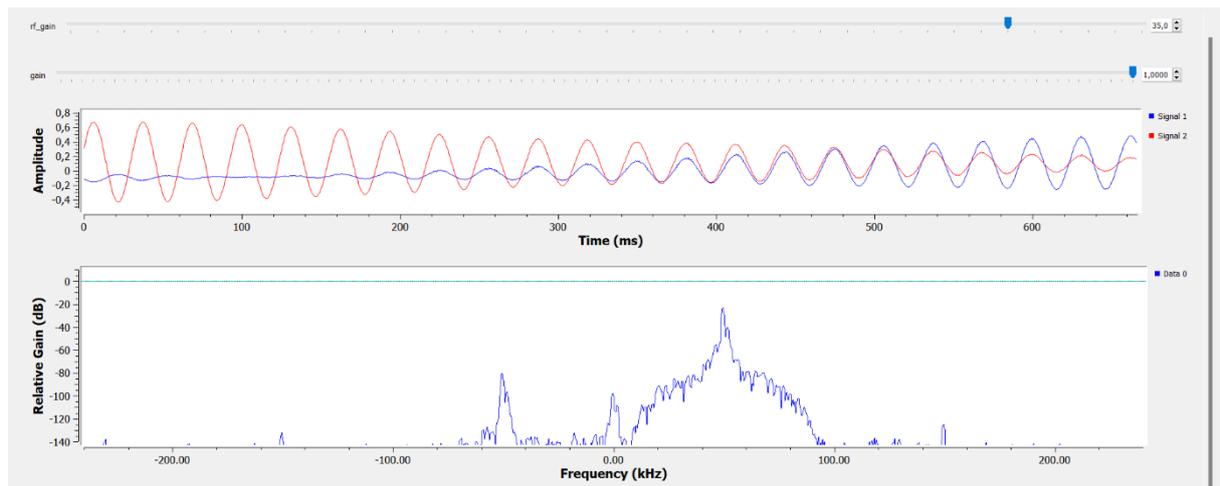


Abbildung 25 Messergebnisse für AM-Versuch

Bei der Auswertung für die Übertragung des Audiosignals ist die Amplitudenmodulation gut zu erkennen. Die obere Signaldarstellung weist eine Veränderung der Amplitude auf, die charakteristisch für die Amplitudenmodulation ist. Die Darstellung des Frequenzverlaufs zeigt auf, dass die Frequenzen bis fast 100 kHz durchgelassen werden. Für die Anpassung der Lautstärke und der Verstärkung sind zwei Schieberegler eingesetzt wurden. Es wurde eine Variable „rf\_gain“ initialisiert. Diese stellt die Verstärkung des Signals dar. Die Initialisierung der zweiten Variablen erfolgte mit der Bezeichnung „gain“. Diese Variable dient zur Regulierung der Lautstärke. Um eine gute Signalqualität zu gewährleisten, wurde die Verstärkung zuerst auf den Wert von 30 dB erhöht. Dies hatte eine deutliche Verbesserung des Signals zur Folge. Bei einer zu großen Verstärkung war eine Verschlechterung der Signalqualität zu verzeichnen. Für das vorliegende Beispiel wurde 35 dB als optimaler Wert für die Verstärkung ermittelt. Die Lautstärke der Ausgabe des Audiosignals wurde neben der normalen Lautstärkeregelung des Zielrechners mit dem zweiten Schieberegler angepasst. Dabei hat sich gezeigt, dass für eine optimale Klangqualität der Wert 1 zu empfehlen ist. Wie auch bei der Frequenzmodulation wird die Signalqualität sporadisch von Störeinflüssen beeinträchtigt. Diese Störungen können auf die Tonspur des Audiosignals, die Verkabelung der beiden bladeRFs oder auf die nicht optimale Einstellung der Filter zurückgeführt werden. Dennoch ist festzuhalten, dass die Übertragung erfolgreich war.

## 4.3 Digitale Modulation

Nachdem in den ersten beiden Versuchen jeweils zwei analoge Modulationsverfahren zur Anwendung kamen, wird im dritten Versuch ein digitales Modulationsverfahren zum Einsatz kommen. Bei der Durchführung wird eine Variante der MSK- (Minimum Shift Keying) Modulation verwendet, die GMSK- (Gaussian Minimum Shift Keying) Modulation. Dieses Modulationsverfahren findet in der heutigen Zeit Anwendung in dem Mobilfunkstandard GSM (Global System for Mobile Communications). In der Versuchsdurchführung soll erneut ein Audiosignal übertragen werden. Ebenso soll die Tonspur am Zielrechner ausgegeben werden.

Um die beiden Modulationsverfahren referenzieren zu können, sind folgende theoretische Vorbetrachtungen nötig. Die MSK-Modulation bildet die Impulse im Basisbandsignal durch Sinus-Halbwellen ab. Dadurch wird eine kontinuierliche Phasenänderung erreicht und Phasensprünge werden vermieden. In Abbildung 26 ist der Phasenverlauf beider Modulationsverfahren abgebildet. Die linke Abbildung zeigt den Phasenverlauf der MSK-Modulation und die rechte Abbildung zeigt den Phasenverlauf der GMSK-Modulation. Es ist gut zu erkennen, dass im Phasenverlauf der MSK-Modulation Knicke entstehen. Diese Knicke sind bei der GMSK-Modulation nicht mehr gegeben. Der entscheidende Unterschied zwischen den beiden Verfahren liegt in der Verwendung eines Filters. Denn bei der GMSK-Modulation kommt ein vorgeschalteter Gauß-Filter zum Einsatz. Dieser Gauß-Filter beschreibt die Basisbandimpulse mithilfe von gaußförmigen Impulsen. Mit dem Einsatz des Filters wird eine Glättung der bei MSK auftretenden Knicke im Phasenverlauf erreicht [17].

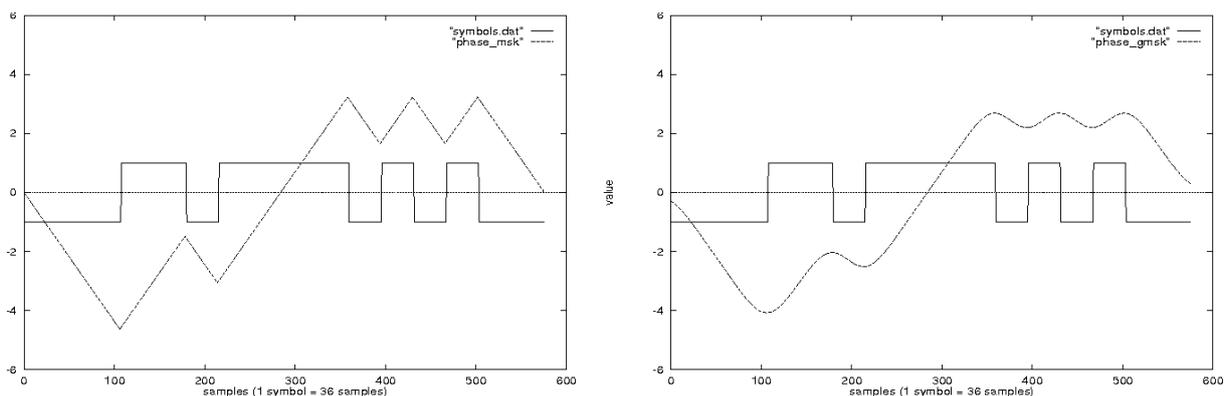


Abbildung 26 Vergleich MSK und GMSK [18]

Der Einsatz des Gauß-Filters bei der GMSK-Modulation ist von großer Bedeutung für die Bandbreite des Signals. Die Bandbreite bei der GMSK-Modulation weist einen noch geringeren Wert als bei der MSK-Modulation auf. Damit hat die GMSK-Modulation einen Vorteil zur MSK-Modulation zu verzeichnen. Dennoch ist zu erwähnen, dass durch die weitere Verringerung der Bandbreite ISI (Intersymbolinterferenzen) auftreten [17].

Für den vorliegenden Versuch wird das zu übertragende Audiosignal durch die GMSK-Modulation moduliert und somit an den Empfänger weitergeleitet. Die Signalübertragung erfolgt trotz der digitalen Modulation analog. Denn das bladeRF wandelt das Signal vor der Übertragung in ein analoges Signal um. Somit kommt das Signal beim Empfänger auch analog an. Das heißt, das digitale Signal muss am Empfänger rekonstruiert und dann analog ausgegeben werden.

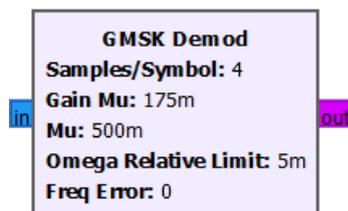


Abbildung 27 GMSK-Demod-Block

Zu Beginn der Versuchsdurchführung muss überlegt werden, welche Bausteine aus der GNU-Radio-Software zum Einsatz kommen. Die Software stellt für eine GMSK-Modulation einen GMSK-Demod-Block zur Verfügung. In Abbildung 27 wird der verwendete Demod-Block dargestellt. Der blau markierte Input des Bausteins bedeutet, dass in den Block ein komplex-moduliertes Eingangssignal eingeht. Der lila markierte Output

bedeutet, dass ein digitales Signal ausgesendet wird. In diesem Fall erfolgt die Ausgabe in einem Strom von Bits oder auch Bitstream genannt. Mithilfe des Bausteins kann die Signalrekonstruktion gewährleistet werden. Der Demodulationsblock enthält 5 verschiedene Parameter, die für die unterschiedlichen Signale individuell angepasst werden können [12].

Nachdem dem Ausgeben des Bitstroms durch den GMSK-Demod-Block müssen die Bits zusammengefasst und in Pakete verpackt werden. Dies wird durch den Baustein „Pack K Bits“ realisiert, der in Abbildung 28 abgebildet ist. Es erfolgt eine Konvertierung des Bitstroms. Die entpackten Bits werden in ein einzelnes Byte verpackt. Das Paket besteht aus K Bits und wird mit 8-K Nullen aufgefüllt. Die Anzahl der Bits der Pakete ist variabel einstellbar [19].



Abbildung 28 Pack K Bits-Block

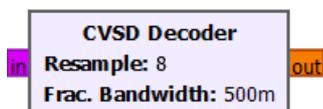


Abbildung 29 CVSD-Decoder-Block

Zum Abschluss der Rekonstruktion des digitalen Signals muss das Signal wieder in die analoge Darstellungsweise umgeformt werden. Dafür ist eine Decodierung notwendig. Für eine Audiocodierung stellt die GNU-Radio-Software ein CVSD- (Continuously variable Slope Delta Modulation) Decodierer zur Verfügung. Der verwendete Baustein für die Decodierung wird in der nebenstehenden Abbildung 29 dargestellt. Der CVSD-Coder hat die Aufgabe, das Signal für die digitale Übertragung auf die erforderliche Bandbreite zu reduzieren. Damit kann das Signal vom Analogen ins Digitale umgewandelt werden. Dieser Vorgang ist wichtig für die Endausgabe der Tonspur an der Audiosenke [20].

Nachdem die Überlegungen zur Umsetzung der GMSK-Modulation abgeschlossen sind, kann die Schaltung in der GNU-Radio-Software implementiert werden. Neben den drei beschriebenen Bausteinen und der osmocomb Source wird noch ein Tiefpassfilter mit in die Schaltung eingebaut. Die erstellte Schaltung für die GMSK-Modulation ist in Abbildung 30 dargestellt.

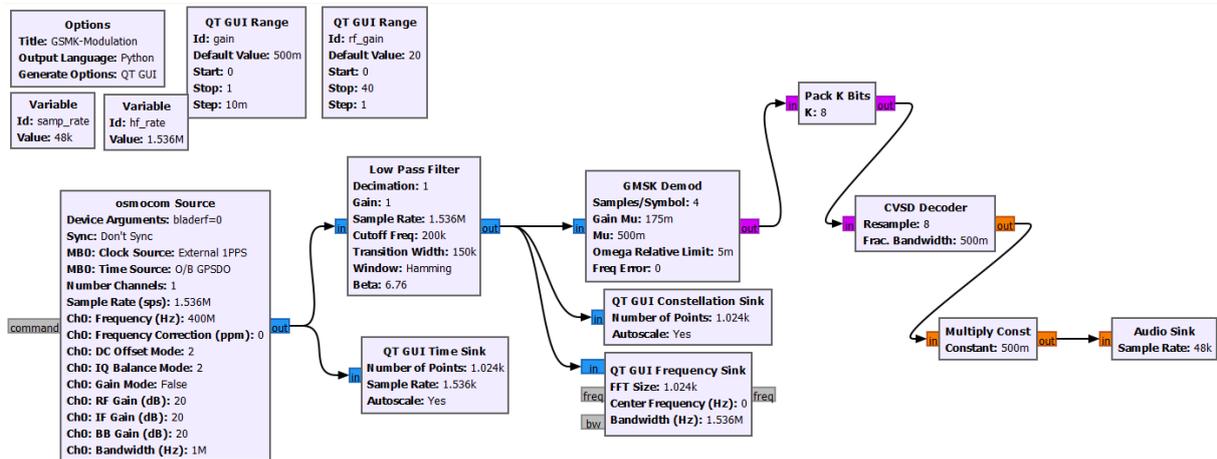


Abbildung 30 Flussdiagramm für GMSK-Versuch

Für die Kommunikation zwischen den beiden bladeRFs wird die Channel-Frequenz wieder mit 400 MHz initialisiert. Der Sender-bladeRF sendet das modulierte Eingangssignal [1]. Von der osmocomb Source abgehend wird zum einen der zeitliche Verlauf des modulierten Signals festgehalten, zudem passiert das Signal einen Tiefpassfilter. Das gefilterte Signal geht im nächsten Schritt in den GMSK-Demod-Block. Für die Auswertung nach der Filterung wird zum einen der Frequenzverlauf dokumentiert und ein Konstellationsdiagramm erstellt. Mit dem Parameter Samples/Symbol im GMSK-Demod-Block kann festgelegt werden, wie das analoge Signal abgetastet wird. Der „Pack K Bits“-Baustein verpackt die Abtastwerte in die Pakete. In diesem Fall werden 8 Bits in jedem Paket zusammengefasst. Für die Decodierung in eine analoge Ausgabe ist der CVSD-Decoder verantwortlich. Somit kann am Ende der Schaltung die Tonspur des Audiosignals an der Audiosenke ausgegeben werden. Ein weiterer wichtiger Punkt für die Implementierung ist die genaue Berechnung der Abtastrate. Die Audiosenke arbeitet mit einer Abtastrate von 48 kps. Für die Berechnung muss die Umwandlung des analogen zum digitalen Signal betrachtet werden. Dabei spielen die initialisierten Werte der Parameter Samples/Symbol und die Anzahl K der Bits eine wichtige Rolle. Der ausgegebene Bitstrom des GMSK-Demod-Blocks wird in Pakete zusammengefasst. Für die Berechnung muss die Abtastrate der Audiosenke, die Samples/Symbol und die Anzahl K der Bits miteinander multipliziert werden.

$$\frac{\text{Samples}}{\text{Symbol}} * \text{Anzahl } K \text{ der Bits} * \text{Abtastrate (Audiosenke)} = \text{Abtastrate (1)}$$

$$4 * 8 \text{ Bits} * 48 \text{ kps} = 1536 \text{ Msps (1)}$$

Nachdem die Parameter für die unterschiedlichen Bausteine der Schaltung berechnet und bestimmt wurden sind, kann die Funktionalität der entwickelten Schaltung getestet werden. Für die Auswertung des Versuchs sind drei verschiedene Diagramme zur Dokumentation in der Schaltung integriert. Diese Messwerte sind in Abbildung 31 festgehalten.

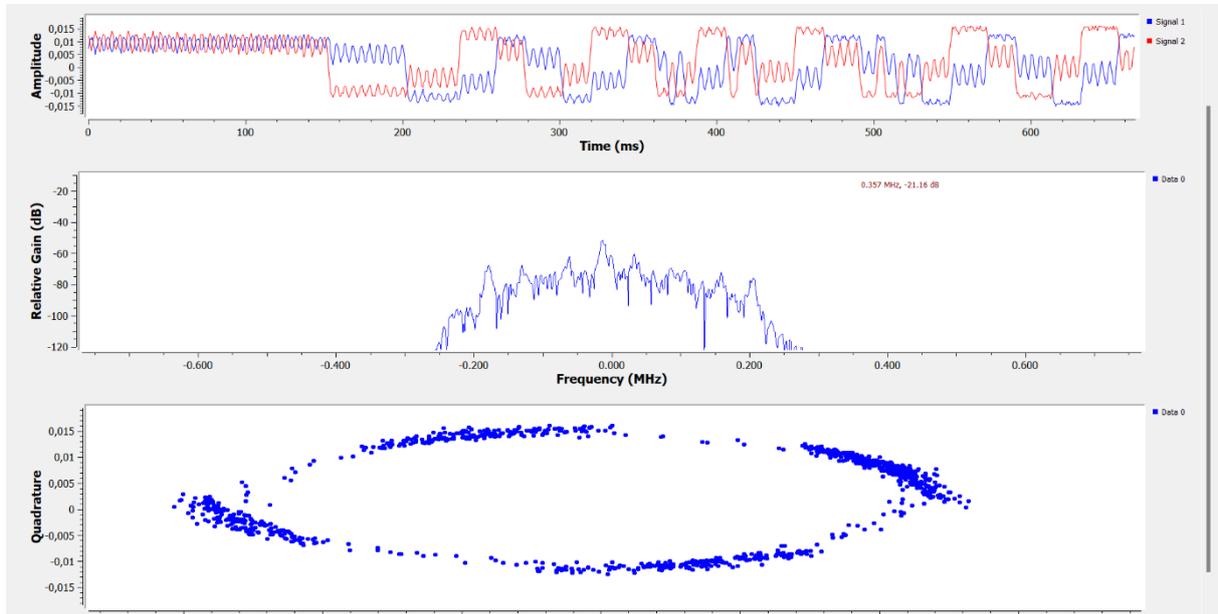


Abbildung 31 Messergebnisse für GMSK-Versuch

Das erste Diagramm zeigt den zeitlichen Verlauf des modulierten Signals. Im Abgleich mit den theoretischen Vorbetrachtungen ist gut zu sehen, dass das Signal keine Knicke, sondern die geglättete Kurvenform aufweist. Ebenso sind die umgewandelten binären Zustände im Signalverlauf gut zu erkennen. Nach der Begrenzung des Signals durch den Tiefpassfilter wurde der Frequenzverlauf und das Konstellationsdiagramm ausgegeben. Das zweite Diagramm zeigt den Frequenzverlauf auf. Dabei wurde die Frequenz durch die Filterung auf einen bestimmten Frequenzbereich begrenzt. Dieser Bereich liegt bei einem Tiefpassfilter zwischen 0 Hz und seiner Grenzfrequenz. Die Grenzfrequenz ist im vorliegenden Beispiel mit 200 kHz initialisiert. Im letzten Diagrammbild ist ein Konstellationsdiagramm zu sehen. Diese Diagrammart dienen zur Darstellung der übertragenen Symbole. Jedes der übertragenen Symbole wird in eine komplex dargestellte Zahl umgewandelt. Ein Punkt im Konstellationsdiagramm entspricht einer dieser komplex dargestellten Zahl und damit dem Symbol. Die Audioübertragung verlief somit erfolgreich. Das Signal konnte gut zwischen den beiden bladeRFs übertragen werden. Zudem ist das Signal durch eine gute Signalqualität gekennzeichnet. Dennoch sind auch bei diesem Versuch sporadische Störungen im Audiosignal zu hören. Mögliche Ursachen für die Störungen kann die Verkabelung zwischen den Funkmodulen oder Probleme bei der Rekonstruktion des Signals sein.

## 5. Erstellen einer Versuchsanleitung

Versuch: Nachbau eines Kommunikationsempfängers mittels bladeRF

### 5.1 Versuchsziele

- Kenntnisse zu Amplitudenmodulation und Frequenzmodulation erlangen
- Kennenlernen der Software von GNU Radio
- Kennenlernen des Aufbaus und Grundkomponenten des bladeRFx115
- Vertrautheit mit dem Umgang der GNU-Radio-Software im Zusammenhang mit dem bladeRF erlangen (Schaltungsaufbau, Simulationsversuche)

### 5.2 Vorbereitende Aufgaben

1. Erläutern Sie die Funktionsweise der Amplitudenmodulation und der Frequenzmodulation? Welche der beiden Modulationsverfahren ist anfälliger für Störungen?
2. Was ist der Unterschied zwischen Schmalband-FM und Breitband-FM?
3. Welche Funktion erfüllt ein Bandpassfilter und welche ein Tiefpassfilter?
4. Wie ist der bladeRFx115 aufgebaut und welchen Frequenzbereich deckt er ab?

### 5.3 Grundlagen

#### 5.3.1 bladeRFx115

Im vorliegenden Versuch kommt als Hardware der bladeRFx115 zum Einsatz. Der bladeRF ist ein von der Firma Nuand entwickeltes Funkmodul. Es kann als Sender, Empfänger und Sendeempfänger verwendet werden. Die Treiber des bladeRF müssen für die Inbetriebnahme auf dem entsprechenden Computer installiert sein, der ein Betriebssystem von Linux, Windows oder macOS verwendet [2].

Die Verbindung zwischen dem Computer und dem bladeRF erfolgt über ein USB-Kabel. Der USB 3.0 Port des blades ist abwärtskompatibel zu USB 2.0. Über diesen USB-Port erfolgt die Spannungsversorgung des Funkmoduls. Eine weitere Möglichkeit, den bladeRF mit Spannung zu versorgen, ist der externe 5 V-Gleichspannungsanschluss. Der bladeRF deckt einen Frequenzbereich von 300 MHz bis 3,8 GHz ab. Des Weiteren kann eine Kanalbandbreite in dem Bereich von 1,5 MHz bis 28 MHz umgesetzt werden. Zu den drei großen Hauptkomponenten im Aufbau des blades zählen der FPGA-Chip, ein HF-Transceiver und ein USB 3.0 Controller. Der FPGA-Chip ist die Steuerungseinheit im blade. Im Chip sind mehrere logische Bausteine vorhanden, die dazu beitragen, dass eine Vielzahl von Schaltungen realisiert werden kann. Der bladeRFx115 beinhaltet einen Altera Cyclone IV E FPGA-Chip [3, 4].

Dieser umfasst bis 115000 Logikelemente. Der FPGA-Chip ist die Schnittstelle zwischen dem USB-Controller und dem HF-Transceiver. Der HF-Transceiver im bladeRF besteht aus zwei verschiedenen Pfaden. Die unabhängigen Tx- (transceiver exchange) und Rx- (receiver exchange) Pfade ermöglichen dem bladeRF die Signalübertragung und den Signalempfang gleichzeitig durchzuführen und somit im Vollduplexmodus zu arbeiten [8, 11].

### 5.3.2 GNU Radio

Im folgenden Versuch wird die Software von GNU Radio verwendet. Die Software dient als Entwicklungsumgebung für Signalverarbeitungsprozesse. Sie kann die drahtlose Kommunikation und reale Funksysteme nachempfinden. Für die Implementierung von unterschiedlichen Anwendungen stellt GNU Radio eine Vielzahl von Signalverarbeitungsbausteinen bereit. Diese Bausteine werden auch in herkömmlichen Funksystemen angewendet. Die Software arbeitet mit der Programmiersprache Python. Solange vom Benutzer die von der Software bereitgestellten Bausteine verwendet werden, sind keine Kenntnisse der Programmiersprache nötig. Für die Neuerstellung von spezifizierten Bausteinen sollten gewisse Kenntnisse zu Python bekannt sein [3, 21].

In der GNU-Radio-Software können vom Benutzer Flussdiagramme, die die Signalverarbeitungsbausteine beinhalten, erstellt werden. Diese erleichtern die Realisierung der Signalverarbeitungsprozesse. In Abbildung 32 wird ein solches Flussdiagramm in der GNU-Radio-Software dargestellt.

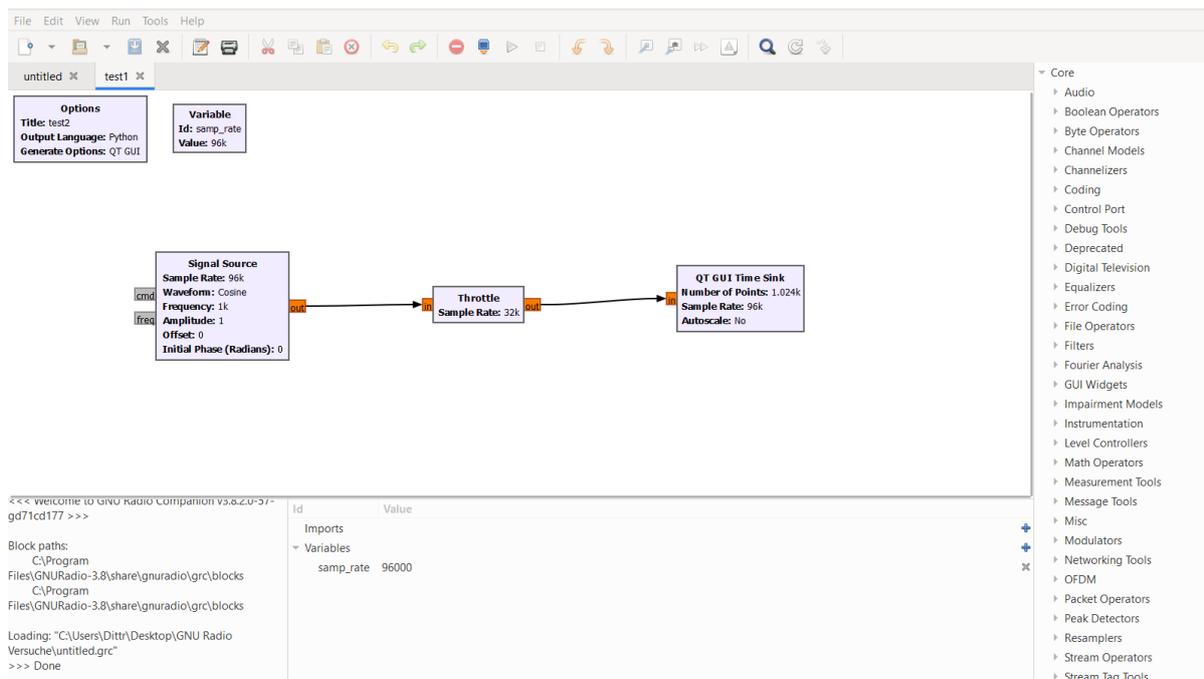


Abbildung 32 Flussdiagramm Probeversuch

Die Abbildung 32 zeigt eine Schaltung, die ein Kosinus-Signal ausgibt. Zwischen der Signalquelle und der Ausgabe befindet sich ein Block, der die Abtastrate drosselt. Der Optionen-Block beinhaltet den Namen des Projektes. Zudem ist ein Variable initialisiert, die den Wert der Abtastrate repräsentieren soll.

Folgende Schritte sind für die Entwicklung dieser Schaltung nötig:

- Klicken auf das Blattsymbol, um ein neues Projekt zu erstellen
- Doppelklick auf den Optionen-Block und unter Id einen Titel hinzufügen
- in der Menüleiste auf das Lupen-Symbol klicken (Schnellsuche)
- im Suchfeld die drei Bausteine „Signal Source“, „Throttle“ und „QT GUI Time Sink“ eingeben
- den gesuchten Baustein mit einem Doppelklick auswählen oder auf den Baustein einmal klicken und dann in das Arbeitsfeld hineinziehen
- Verbinden der Blöcke durch Ziehen von einem zum anderen Block

Sollten bestimmte Blöcke rot markiert sein, sind diese fehlerhaft. Zum Beheben der Fehler müssen die Verbindungen der Blöcke überprüft werden. Zudem können Fehler bei der Einstellung der Signalverarbeitungsbausteine entstanden sein. Sind Fehler in der Schaltung vorhanden, ist keine Simulation der Schaltung möglich. Somit ist auch das Play-Symbol aus der Menüleiste, welches die Simulation startet, ausgegraut.

Sind in der Schaltung keine Fehler vorhanden, kann die Simulation gestartet werden. Bei der Ausgabe von Diagrammen öffnet sich zur Anzeige der Messwerte ein neues Fenster. Im vorliegenden Beispiel wird der Zeitverlauf des Signals aufgezeichnet. Die Ausgabe wird in Abbildung 33 angezeigt. Dabei wird die Amplitude mit dem eingestellten Wert von 1 dargestellt.

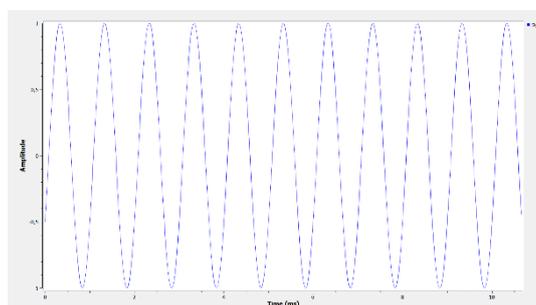


Abbildung 33 Messergebnis Probeversuch

Zum Beenden der Simulation kann das Fenster mit dem Diagrammverlauf geschlossen werden. Eine weitere Möglichkeit ist es, in der Menüleiste das Stopp-Symbol auszuwählen. Danach kann die Schaltung vom Benutzer wieder verändert werden. Für die Veränderung bestimmter Parameter während der Simulation ist eine Integrierung von Schiebereglern möglich. Für diesen Fall muss eine Variable angelegt werden, deren Wert verstellbar ist. Dies ist mit der GNU-Radio-Software möglich, indem der Baustein „QT GUI Range“ verwendet wird.

Dieser Baustein kann die Variablen mit unterschiedlichen Werten initialisieren. Das Anlegen der Variable und deren Wertespektrum ist vom Benutzer vorzunehmen.

In der folgenden Schrittanleitung wird das Anlegen einer Variablen mit mehreren Werten beschrieben. Damit soll es möglich sein, die Amplitude des Kosinus-Signals während der laufenden Simulation zu verändern. Diese Schritte sind notwendig:

- In der Menüleiste auf das Lupensymbol klicken und in Suchfeld den Baustein „QT GUI Range“ suchen
- den Baustein in das Arbeitsfeld ziehen und mit einem Doppelklick auswählen, um die Parameter anzupassen
- im Feld Id einen Variablennamen eingeben
- der Standardwert wird auf 1 gesetzt
- der Startwert wird mit 0 und der Stoppwert mit einer 5 initialisiert
- die Schrittweite bekommt den Wert 1 zugewiesen
- Eingabe mit „OK“ bestätigen

Sobald der Benutzer jetzt die Simulation startet, wird ein Schieberegler oberhalb des Diagramms abgebildet. Während der laufenden Simulation kann der Wert der Amplitude verändert werden und es ist nicht mehr nötig, die Simulation immer wieder zu beenden.

### 5.3.3 Frequenzmodulation (FM)

Die Frequenzmodulation ist ein analoges Modulationsverfahren. Eine der ersten Anwendungen der Frequenzmodulation war der UKW- (Ultrakurzwellen) Hörfunk. Charakteristisch für dieses Modulationsverfahren ist die gleichbleibende Amplitude des Signals. In der unten stehenden Abbildung 34 ist eine vereinfachte Darstellung der Frequenzmodulation dargestellt. Darauf ist ein Nachrichtensignal oder auch Nutzsignal zu erkennen. Gleichzeitig ist das dazu passend modulierte Signal abgebildet. Dabei wird deutlich, dass sich beim modulierten Signal nur die Frequenz ändert [13].

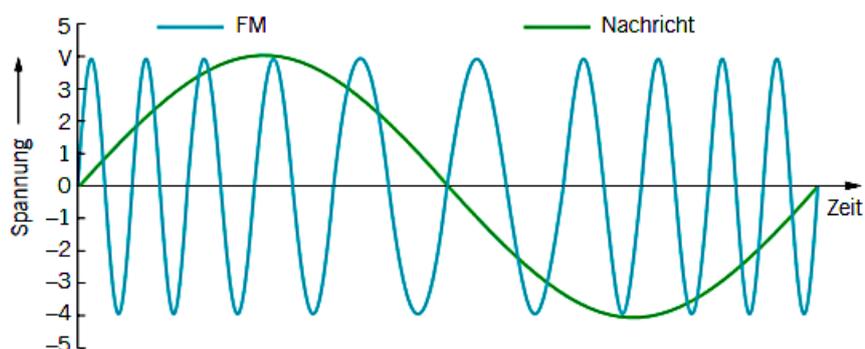


Abbildung 34 Darstellung der Frequenzmodulation [25]

Die Funktionsweise der Frequenzmodulation beginnt mit einem Nutzsignal. Das Nutzsignal enthält die zu übertragende Information. Damit das Signal moduliert werden kann, wird es durch ein hochfrequentes Trägersignal modifiziert. Somit kann eine hochfrequente Übertragung des Signals gewährleistet werden. Das daraus resultierende Signal ist das frequenzmodulierte Signal [13].

### 5.3.4 Amplitudenmodulation (AM)

Ein weiteres analoges Modulationsverfahren ist die Amplitudenmodulation. Sie wird heute noch für Lang-, Mittel- und Kurzwelle verwendet. Die Amplitudenmodulation ist ein sehr preiswertes Verfahren, da es einfach zu realisieren ist. Für die Amplitudenmodulation gilt, dass ein niederfrequentes Nutzsignal auf ein hochfrequentes Trägersignal moduliert werden soll. In der nebenstehenden Abbildung 35 wird im ersten Diagramm das Nutzsignal gezeigt. Im zweiten Diagrammausschnitt ist das hochfrequente Trägersignal abgebildet. Durch die unterschiedlichen Pegel des Nutzsignals ändert sich die Amplitude des modulierten Signals, wie im dritten Diagrammausschnitt zu sehen ist. Somit enthält bei der Amplitudenmodulation die Amplitude die Information. Die Frequenz und die Phase bleiben bei der Modulation erhalten. Ein Beispiel für diesen Vorgang ist die Übertragung eines Audiosignals. Damit die Lautstärke des Signals übertragen werden kann, wird die Amplitudenhöhe verändert. Die Frequenz der Hüllkurve gibt die Information der Tonhöhe des Signals weiter. Die Amplitudenmodulation ist im Gegensatz zur Frequenzmodulation störanfälliger.

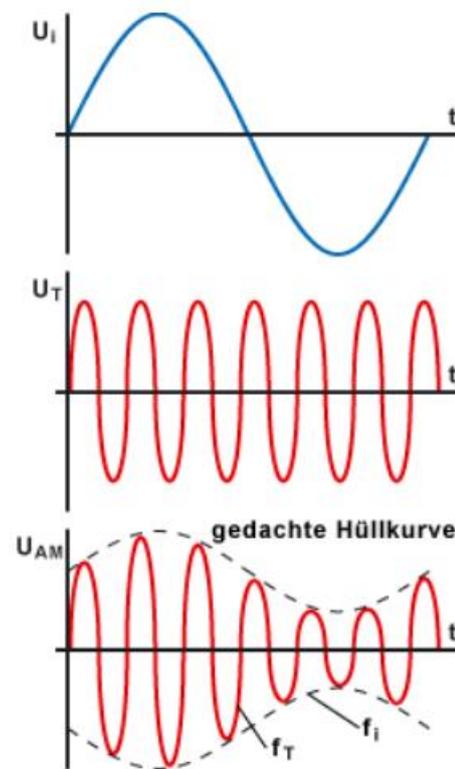


Abbildung 35 Ablauf der AM [16]

Dies begründet sich darauf, dass bei der Amplitudenmodulation die Information in der Amplitude enthalten ist. Folgende Einflüsse können Störungen verursachen:

- Übertragungen im Stromnetz mit hoher Frequenz, daraus entsteht eine hochfrequente Abstrahlung
- Ein- und Ausschalten von Geräten im Stromnetz
- Wetterbedingte Einflüsse [16]

Die Folge aus diesen Störeinflüssen ist, dass die Information verloren geht oder verfälscht wird. Dies äußert sich beim Übertragen des Signals durch Verzerrungen und einer schlechten Signalqualität [16].

### 5.3.5 Programmable Synthesizer HM8134

Im vorliegenden Versuch wird ein weiteres Gerät benötigt. Dabei handelt es sich um den Programmable Synthesizer HM8134. Der Synthesizer zeichnet sich durch seine hohe Leistung und seine schnelle und einfache Bedienung aus. Dabei wird ein Frequenzbereich von 1 Hz bis hin zu 1,2 GHz abgedeckt. Der breite Bereich enthält das am häufigsten benötigte Spektrum an Audio-, Video- und HF-Frequenzen. Diese Frequenzen werden in Vielzahl von den Empfängern und Sendern in den verschiedenen Kommunikationssystemen verwendet. Die Frequenzauflösung des Synthesizers beträgt 1 Hz. Eine weitere Eigenschaft des Synthesizers ist das dieser über unterschiedliche Modulationsverfahren verfügt. Dazu zählen unter anderem die Amplitudenmodulation, die Frequenzmodulation und die Phasenmodulation [22].

Im folgenden Schritt wird der Aufbau des Programmable Synthesizer HM8134 dargelegt. Die Abbildung 36 zeigt den Synthesizer mit seinen verschiedenen Komponenten.

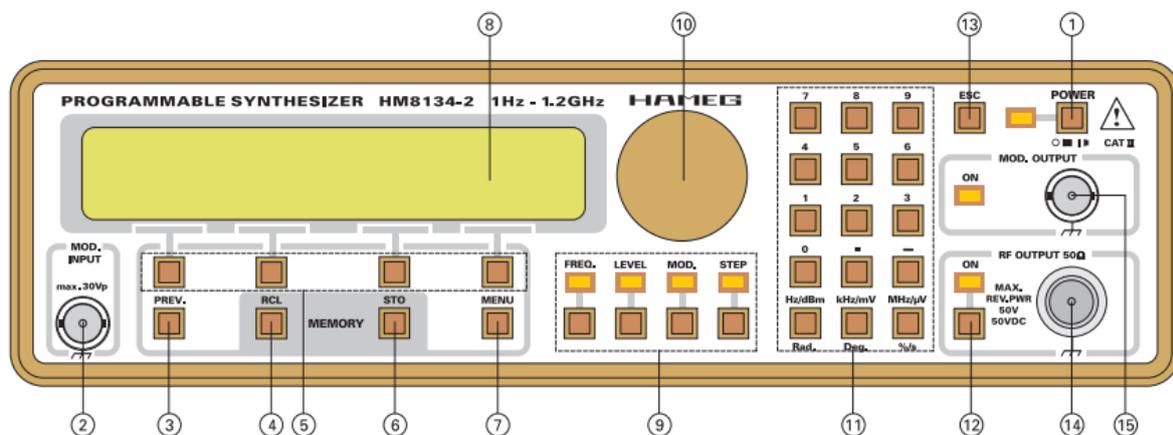


Abbildung 36 Aufbau des Synthesizers [22]

1. Power Schalter (Ein-/Ausschalter mit LED)
2. MOD.IN (Eingang für externe Modulation)
3. PREV. (Auswahl des vorherigen Menüs)
4. RCL Memory (Tasten zum Abrufen des Speichers)
5. Kontextsensitive Tasten
6. STO (Taste zum Sichern im Speicher)
7. MENU (Eingabetaste im Konfigurationsmenü)
8. Display
9. Aktionstasten mit LED
10. Drehsteuerung

11. Nummerneingabefeld
12. Taste zur Aktivierung des Ausgangs
13. ESC. (bricht aktuelle Anzeige ab)
14. Signalausgabe
15. Ausgabe des modulierten Signals

## 5.4 Versuchsdurchführung

### 5.4.1 Erster Versuch mit GNU Radio

Für einen ersten Einblick in den Umgang mit der GNU-Radio-Software soll die Schaltung aus der Abbildung 32 nachgebaut werden. Die Ergebnisse des Versuches sollen ausgegeben werden.

- dazu muss die Anwendung GNU Radio Companion am verwendeten Computer aufgerufen werden
- Unter dem Punkt 5.3.2 wird in der ersten Schrittanleitung gezeigt, wie ein neues Projekt erstellt wird und welche Signalverarbeitungsblöcke Anwendung finden
- ist die Schaltung korrekt aufgebaut, kann die Simulation beginnen
- Simulation mit Play-Symbol starten
- das angezeigte Diagramm ausgeben und mithilfe eines Bildschirm-Screenshots abspeichern

Nachdem das Kosinus-Signal erfolgreich ausgegeben wurde, soll ein Schieberegler zur Änderung der Amplitude integriert werden. Der Benutzer soll das Ergebnis in einem Diagramm ausgeben.

- Unter dem Punkt 5.3.2 wird in der zweiten Schrittanleitung gezeigt, wie ein Schieberegler integriert wird
- bei korrekter Implementierung des Schiebereglers kann die Simulation mit dem Play-Symbol gestartet werden
- nach der Ausführung den Schieberegler bewegen, sodass die Amplitude unterschiedliche Werte annimmt
- das angezeigte Diagramm ausgeben und mithilfe eines Bildschirm-Screenshots abspeichern (Amplitude soll ungleich 0 und 1 sein)

Anschließend soll ein zweiter Schieberegler hinzugefügt werden. Dieser soll die Frequenz des Kosinus-Signals verändern. Der Benutzer soll des Kosinus-Signal mit einer geänderten Frequenz ausgeben.

- Schieberegler für Frequenz integrieren (nach Schrittanleitung in Punkt 5.3.2)

- Simulation starten
- Diagramm ausgeben lassen
- Angezeigtes Diagramm mit Bildschirm-Screenshot sichern
- Flussdiagramm mit Bildschirm-Screenshot sichern

**Wichtig!** Beim Abspeichern der Diagramme sollen die verwendeten Schieberegler mit abgebildet werden.

#### 5.4.2 Frequenzmodulation mit GNU Radio und bladeRF

Die GNU-Radio-Software eignet sich dazu, Schaltungen aufzubauen, um eine Frequenzmodulation durchzuführen. Damit vor dem Versuch die technische Funktionalität des Computers und des bladeRFs geprüft werden kann, soll vor dem eigentlichen Versuch eine Testübertragung erfolgen. Dies äußert sich durch die Übertragung eines 1 kHz -Tons. Dabei kommt der in den Vorbetrachtungen beschriebene Programmable Synthesizer HM8134 zum Einsatz. Bevor der Testvorgang starten kann, muss eine Schaltung im GNU Radio entwickelt werden.

##### a) Testübertragung

Überlegen Sie, welche Komponenten für die Schaltungsentwicklung gebraucht werden. Für die Implementierung wird in jedem Fall der Baustein „osmocom Source“ benötigt, damit eine Kommunikation zum bladeRF aufgebaut wird. Zudem sollte die Schaltung einen Baustein zur Demodulation des frequenzmodulierten Signals beinhalten. Des Weiteren soll der zeitliche Verlauf, der Frequenzverlauf und der 1 kHz- Ton ausgegeben werden.

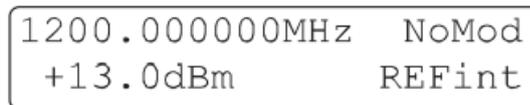
Sobald die Schaltung entwickelt wurde, können die Parameter der einzelnen Signalverarbeitungsbausteine bestimmt werden. Als Hilfestellung werden folgende Werte für die „osmocom Source“ vorgegeben:

- Im Parameter „Device Arguments“ den bladeRF mit „bladerf=0“ initialisieren
- Ch0: Frequency muss mindetens 300 MHz betragen
- Für eine bessere Verstärkung des Signals kann der Parameter „Ch0: RF Gain“ verändert werden (empfohlen: zwischen 20 bis 35 dB)

Die Parameter der anderen Bausteine können je nach Signalqualität individuell angepasst werden. Im Anschluss muss der bladeRF mit dem Programmable Synthesizer HM8134 verbunden werden. Dazu muss vom Synthesizer (Ausgang moduliertes Signal) ein Kabel an den RX-Pfad des blades gehen. Zwischen dem blade und dem Computer, auf dem GNU Radio ausgeführt wird, besteht ebenfalls eine Verbindung.

Die folgende Schrittanleitung erklärt die Einstellung am Synthesizer und erfolgt am Beispiel der Frequenzmodulation. Das Vorgehen bei anderen Modulationsverfahren bleibt gleich.

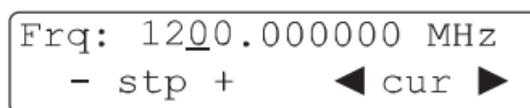
- Synthesizer mit POWER-Knopf einschalten
- Startbildschirm wird angezeigt (Abbildung 37)



```
1200.000000MHz  NoMod
+13.0dBm        REFint
```

Abbildung 37 Startbildschirm [22]

- Einstellen der Frequenz: **FREQ.** drücken und die Frequenz mithilfe der Drehsteuerung oder des Nummerneingabefeld eingeben (Abbildung 38)



```
Frq: 1200.000000 MHz
- stp +    < cur >
```

Abbildung 38 Frequenzmenü [22]

- Einstellen der Modulationsart: **MOD.** drücken und das gewünschte Modulationsverfahren mithilfe der kontextsensitiven Tasten auswählen (Abbildung 39)



```
* MODULATION MENU *
AM  FM  PM  GATE
```

Abbildung 39 Modulationsmenü [22]

- im Modulationsmenü können dann die entsprechenden Parameter der Modulation angepasst werden (Abbildung 40)



```
***** FM MENU *****
Shape Fmod Dev >Off
```

Abbildung 40 FM-Menü [22]

- Für die Einstellung des 1 kHz- Tons muss auf „Fmod“ geklickt werden, die Eingabe erfolgt mit der Drehsteuerung oder dem Nummerneingabefeld
- Sobald alle Einstellungen erfolgt sind kann die Taste zur Aktivierung des Ausgangs betätigt werden

Nachdem die Einstellungen am Synthesizer abgeschlossen sind, kann mit dem Start der Simulation fortgefahren werden. Testen Sie, ob die Schaltung funktioniert und welche Ausgabe erfolgt. Schätzen Sie anhand der angezeigten Diagramme ein, ob die Übertragung erfolgreich war und beurteilen Sie am ausgegebenen Ton die Qualität des Audiosignals. Sollte die Qualität des Signals nicht ausreichend gut sein, kann die Änderung der Parameter für eine Verbesserung sorgen.

Ist die Übertragung erfolgreich gewesen, sind folgende Schritte vorgesehen:

- machen Sie ein Bildschirm-Screenshot von der erstellten Schaltung in der GNU-Radio-Software und speichern Sie diese ab
- machen Sie ein Bildschirm-Screenshot von den ausgegebenen Diagrammen und speichern Sie diese ab
- danach kann die Simulation gestoppt werden
- Signalausgabe des Synthesizers kann beendet werden

#### b) Übertragung eines Audiosignals

Nach dem erfolgreichen Testversuch kann die eigentliche Übertragung starten. Dazu wird vom Sender-bladeRF ein Audiosignal übertragen. Abgehend vom TX-Pfad des Sender-bladeRF wird ein Kabel an den RX-Pfad des Empfänger-bladeRF angeschlossen. Bei der Frequenzmodulation bietet GNU Radio zwei Blöcke an. Zum einen die Schmalband-FM (NBFM Receive) und zum anderen die Breitband-FM (WBFM Receive). Entscheiden Sie, welche der beiden Blöcke zum Einsatz kommen soll. Zudem sollten Sie überlegen, ob ein Filtereinsatz für die Durchführung des Versuches sinnvoll wäre. Falls ja, welche Filterart sollte gewählt werden? Zur Aufnahme der Messergebnisse soll erneut der zeitliche Verlauf und der Frequenzverlauf bestimmt werden. Des Weiteren soll die Ausgabe des Audiosignals erfolgen.

Sobald die Schaltung korrekt aufgebaut wurde, kann die Übertragung vom Sender-bladeRF starten. Danach kann der Empfänger-bladeRF die Simulation ebenfalls zur Ausführung bringen. War die Übertragung erfolgreich, sind folgende Schritte durchzuführen:

- Ergebnis beurteilen
- Flussdiagramm (Schaltung) und Messwerte mit Bildschirm-Screenshot sichern
- Simulation beenden

#### 5.4.3 Amplitudenmodulation mit GNU Radio und bladeRF

Im dritten Versuch soll eine Schaltung zur Realisierung einer Amplitudenmodulation implementiert werden. Wie auch bei der Frequenzmodulation, soll vor der Übertragung des eigentlichen Audiosignals ein Testvorgang erfolgen.

##### a) Testübertragung

Entwickeln Sie eine Schaltung zur Durchführung einer Amplitudenmodulation. Als Testübertragung soll ein 1 kHz- Ton vom Programmable Synthesizer HM8134 gesendet werden. Überlegen Sie, welche Demodulations- und Filterbausteine die Schaltung enthalten muss. Zudem soll der Frequenzverlauf, der zeitliche Verlauf und die Tonspur des Signals ausgegeben werden.

Folgende Aufgaben sind zu erfüllen:

- Synthesizer nach Anleitung bei Frequenzmodulation einstellen und modulierte Signal ausgeben lassen (Umstellung auf Amplitudenmodulation erforderlich)
- Ergebnis beurteilen
- Flussdiagramm (Schaltung) und Messwerte mit Bildschirm-Screenshot sichern
- Simulation beenden

### b) Übertragung eines Audiosignals

Im folgenden Versuch wird erneut ein Audiosignal vom Sender-bladeRF übertragen. Für die Durchführung kann die Schaltung aus der Testübertragung verwendet werden, insofern diese fehlerfrei funktioniert.

Folgende Aufgaben sind zu erfüllen:

- nach erfolgreicher Simulation einen Schieberegler für den RF-Gain integrieren
- verschiedene Verstärkungen für Amplitudenmodulation ausprobieren
- Flussdiagramm (Schaltung) und Messwerte mit Bildschirm-Screenshot sichern
- Simulation beenden

Im Anschluss kann die Software geschlossen werden. Bei Bedarf kann sich die Python-Datei der verschiedenen erstellten Projekte angezeigt werden lassen. Diese wird zusammen mit der GNU-Radio-Datei abgespeichert.

## 5.5 Auswertung

1. Stellen Sie ihre Diagramme aus dem Versuch von Punkt 5.4.1 dar.
2. Stellen Sie den Zeitverlauf und den Frequenzverlauf der beiden durchgeführten Modulationsverfahren auf der Empfängerseite dar.
3. Zeigen Sie auf welche Flussdiagramme Sie in der GNU-Radio-Software entworfen haben und begründen Sie die Auswahl ihrer Signalverarbeitungsblöcke.
4. Bewerten Sie, ob die Übertragung des Signals erfolgreich gewesen ist!  
Begründen Sie ihre Aussage!
5. Welche Störungen können bei den angewendeten Modulationsverfahren auftreten?

## 6. Fazit

Bei der Durchführung der einzelnen Versuche hat sich gezeigt, dass die Übertragung der Audiosignale erfolgreich war. In den unterschiedlichen Untersuchungen haben sich alle drei Modulationsverfahren bewähren können. Rückblickend war die Realisierung der Frequenzmodulation, der Amplitudenmodulation und der GMSK-Modulation mit wenigen Signalverarbeitungsblöcken aus der GNU-Radio-Software möglich.

Bei der Entwicklung der Schaltungen bzw. Flussdiagramme gab es anfängliche Probleme. Die Übertragungen zwischen den blades war zwar erfolgreich, aber die Signalqualität war schlecht. Bei der Frequenzmodulation wurde das Signal meistens von Verzerrungen überlagert und bei der Amplitudenmodulation konnte die Tonspur in den ersten Versuchen gar nicht herausgehört werden. Deswegen sind für die Prüfung des blades Testübertragungen erfolgt. Diese sollten Gewissheit geben über die einwandfreie Funktion des blades und des verwendeten Computers. Mit der Anpassung der Parameter der Signalverarbeitungsblöcke war es möglich, die Signalqualität immer weiter zu verbessern. Besonders ausschlaggebend war die Anpassung des RF-Gain-Parameter im osmocomb-Source-Block. Dieser hatte zu Folge, dass das Signal verstärkt werden konnte und es war ein deutlicher Fortschritt zu erkennen.

Anfängliche Probleme gab es ebenfalls bei der Verbindung mit dem bladeRF. Aufgrund der Tatsache, dass die neusten Treiber des bladeRF installiert waren, konnte aus unbekanntem Gründen keine Verbindung zwischen Computer und dem blade erfolgen. Insgesamt ist festzuhalten, dass bei der Arbeit mit neueren Computern und dem entsprechend neueren Betriebssystemen es zu Komplikationen bei der Installation und der Kommunikation zwischen den Komponenten kommen kann.

Zusammenfassend lässt sich sagen, dass nach einer gewissen Einarbeitungsphase in den Umgang mit der Software und dem bladeRF die Versuche erfolgreich abgeschlossen werden konnten. Die Software lässt sich intuitiv bedienen und ist für Neuanwender einfach zu verstehen. Bei der Erstellung der Flussdiagramme müssen auf jeden Fall Vorüberlegungen erfolgen, wie das Projekt zu realisieren ist. Weiterhin festzuhalten ist, dass der bladeRF und auch die Software weitaus mehr als die verwendeten Modulationsarten bereithält. Für nachfolgende Studenten erfolgte die Entwicklung einer Versuchsanleitung, in der die ersten Schritte für die Arbeit mit GNU Radio und dem bladeRF detailliert erläutert sind. Zudem ist das Nachbauen der Schaltungen und der dazugehörigen Testübertragungen möglich. Abschließend ist zu erwähnen, dass durch die Durchführung der einzelnen Versuche eine Erweiterung der Kenntnisse im kommunikationstechnischen Bereich erreicht werden konnte.

## 7. Literatur- und Quellenverzeichnis

- [1] M. Fox, Realisierung eines Kommunikationssenders mittels Software Defined Radio und bladeRF, 2022.
- [2] „Nuand,“ Nuand , [Online]. Abgerufen unter: <https://www.nuand.com/bladerf-1/>. [Zugriff am 01 06 22].
- [3] Z. Wang, „Semantic Scholar,“ 26 11 17. [Online]. Abgerufen unter: [https://essay.utwente.nl/74086/1/Wang\\_MA\\_EEMCS.pdf](https://essay.utwente.nl/74086/1/Wang_MA_EEMCS.pdf). [Zugriff am 07 06 22].
- [4] „Nuand,“ Nuand, [Online]. Abgerufen unter <https://www.nuand.com/product/bladerf-x115/>. [Zugriff am 02 06 22].
- [5] Nuand, „Nuand,“ [Online]. Abgerufen unter: <https://nuand.com/bladeRF-brief.pdf>.
- [6] „mikrocontroller.net,“ [Online]. Abgerufen unter: <https://www.mikrocontroller.net/articles/FPGA>. [Zugriff am 20 06 22].
- [7] P. D.-I. W. Gehrke, P. D.-I. M. Winzker, P. D.-I. K. Urbanski und P. D.-I. R. Weitowitz, „SpringerLink,“ 2016. [Online]. Abgerufen unter: <https://link.springer.com/content/pdf/10.1007/978-3-662-49731-9.pdf>. [Zugriff am 23 06 22].
- [8] „Intel,“ [Online]. Abgerufen unter: <https://www.intel.de/content/www/de/de/products/sku/210467/cyclone-iv-ep4ce115-fpga/specifications.html>. [Zugriff am 24 06 22].
- [9] „Technologie im Wandel,“ 18 06 21. [Online]. Abgerufen unter: <http://semiconductorsportal.com/2021/06/18/wie-funktioniert-ein-transceiver/>. [Zugriff am 25 06 22].
- [10] TU Berlin, „Phase Lock Loop,“ [Online]. Abgerufen unter: <https://kn-pr.tkn.tu-berlin.de/script/html/BlockB/add/pll.php>. [Zugriff am 06 22].
- [11] Lime Microsystems , „limemicro.com,“ [Online]. Abgerufen unter: <https://limemicro.com/app/uploads/2015/10/LMS6002Dr2-DataSheet-1.2r0.pdf>. [Zugriff am 06 22].
- [12] GNU Radio, „Wiki GNU Radio,“ [Online]. Abgerufen unter: [https://wiki.gnuradio.org/index.php/GMSK\\_Demod](https://wiki.gnuradio.org/index.php/GMSK_Demod). [Zugriff am 08 22].

- [13] P. Schnabel, „Elektronik Kompendium,“ [Online]. Abgerufen unter: <https://www.elektronik-kompendium.de/sites/kom/0401251.htm>. [Zugriff am 08 22].
- [14] P. D.-I. W. Pläßmann und U.-P. D.-I. h. D. Schulz, „SpringerLink,“ 2016. [Online]. Abgerufen unter: <https://link.springer.com/book/10.1007/978-3-658-07049-6>. [Zugriff am 08 22].
- [15] R. Kharadi und H. Mehta, 2016. [Online]. Abgerufen unter <https://academic.csuohio.edu/yuc/mobile/mcproj/8d-Audio%20File%20Transmission%20using%20GNU%20RADIO%20and%20USRP%20.pdf>. [Zugriff am 08 22].
- [16] P. Schnabel, „Elektronik Kompendium,“ [Online]. Abgerufen unter: <https://www.elektronik-kompendium.de/sites/kom/0401181.htm>. [Zugriff am 08 22].
- [17] M. Lenzhofer und P. Meissner. [Online]. Abgerufen unter: [https://www2.spsc.tugraz.at/www-archive/downloads/skriptum\\_nt\\_lab1.pdf](https://www2.spsc.tugraz.at/www-archive/downloads/skriptum_nt_lab1.pdf). [Zugriff am 08 22].
- [18] „Semantic Scholar,“ 1996. [Online]. Abgerufen unter: <https://www.semanticscholar.org/paper/Gmsk-in-a-Nutshell-Turletti/2980144cff3336f5643f4a46ad6e48730c9ffa42>. [Zugriff am 08 22].
- [19] GNU Radio , „Wiki GNU Radio,“ [Online]. Abgerufen unter: [https://wiki.gnuradio.org/index.php?title=Pack\\_K\\_Bits](https://wiki.gnuradio.org/index.php?title=Pack_K_Bits). [Zugriff am 08 22].
- [20] GNU Radio , „GNU Radio Manual and C++ API Reference,“ [Online]. Abgerufen unter [https://www.gnuradio.org/doc/doxygen/v3.8.1.0/classgr\\_1\\_1\\_vocoder\\_1\\_1\\_cvds\\_\\_encoder\\_\\_sb.html](https://www.gnuradio.org/doc/doxygen/v3.8.1.0/classgr_1_1_vocoder_1_1_cvds__encoder__sb.html). [Zugriff am 08 22].
- [21] GNU Radio, „GNU Radio,“ [Online]. Abgerufen unter: <https://www.gnuradio.org/about/>. [Zugriff am 08 22].
- [22] Rohde & Schwarz, „Rohde & Schwarz,“ [Online]. Abgerufen unter: <https://docs.rs-online.com/3633/0900766b80193047.pdf>. [Zugriff am 08 22].
- [23] D. Bachfeld, „heise online,“ [Online]. Abgerufen unter: <https://www.heise.de/ratgeber/Arduino-ESP32-und-Pi-als-Radiosender-4521173.html?seite=2>. [Zugriff am 08 22].

- [24] Uni Kiel, [Online]. Abgerufen unter: [https://www.ieap.uni-kiel.de/plasma/ag-piel/elektronik/f17\\_3.jpg](https://www.ieap.uni-kiel.de/plasma/ag-piel/elektronik/f17_3.jpg). [Zugriff am 08 22].
- [25] D.-I. R. Tuzinski, „Elektro Praktiker,“ [Online]. Abgerufen unter: <https://www.elektropraktiker.de/nachricht/modulationen-frequenzmodulation-4/>. [Zugriff am 08 22].

## 8. Abbildungsverzeichnis

Abbildung 1 Darstellung des bladeRF [2].....	6
Abbildung 2 Aufbau des bladeRFx115 [5].....	7
Abbildung 3 Aufbau des FPGA [7].....	9
Abbildung 4 Aufbau eines Logikblocks [7] .....	10
Abbildung 5 Aufbau des HF-Transceivers [11] .....	11
Abbildung 6 osmocom-Block .....	14
Abbildung 7 Benutzeroberfläche von GNU Radio .....	15
Abbildung 8 Signal-Source .....	17
Abbildung 9 Flussdiagramm der Testschaltung .....	17
Abbildung 10 Ergebnisse der Testschaltung.....	18
Abbildung 11 Vorgang der FM [23].....	19
Abbildung 12 Flussdiagramm für FM-Testversuch.....	20
Abbildung 13 Messwerte für FM-Testversuch.....	20
Abbildung 14 FM-Demod-Block.....	21
Abbildung 15 WBFM- und NBFM-Baustein.....	21
Abbildung 16 Flussdiagramm für FM-Versuch.....	22
Abbildung 17 Parametererklärung für osmocom-Block.....	23
Abbildung 18 GUI-Range Block.....	23
Abbildung 19 Messergebnisse für FM-Versuch .....	24
Abbildung 20 Vorgang der AM [24].....	25
Abbildung 21 AM-Demod-Block.....	25
Abbildung 22 Band Pass Filter- Baustein.....	26
Abbildung 23 Flussdiagramm für AM-Versuch.....	26
Abbildung 24 Messergebnisse für AM-Testversuch .....	27
Abbildung 25 Messergebnisse für AM-Versuch .....	28
Abbildung 26 Vergleich MSK und GMSK [18].....	29
Abbildung 27 GMSK-Demod-Block.....	30
Abbildung 28 Pack K Bits- Block .....	30
Abbildung 29 CVSD-Decoder-Block .....	30
Abbildung 30 Flussdiagramm für GMSK-Versuch.....	31
Abbildung 31 Messergebnisse für GMSK-Versuch .....	32
Abbildung 32 Flussdiagramm Probeversuch .....	34
Abbildung 33 Messergebnis Probeversuch.....	35
Abbildung 34 Darstellung der Frequenzmodulation [25] .....	36
Abbildung 35 Ablauf der AM [16] .....	37

Abbildung 36 Aufbau des Synthesizers [22] .....	38
Abbildung 37 Startbildschirm [22] .....	41
Abbildung 38 Frequenzmenü [22].....	41
Abbildung 39 Modulationsmenü [22].....	41
Abbildung 40 FM-Menü [22] .....	41

# Selbstständigkeitserklärung

Arbeit: Bachelorarbeit

Name, Vorname: Dittrich, Richard

Hiermit versichere ich, dass die vorliegende Arbeit selbstständig verfasst wurde, dass keine anderen Quellen und Hilfsmittel als die angegebenen verwendet wurden und dass die Stellen der Arbeit, die aus fremden literarischen Werken oder Darstellungen wissenschaftlichen oder technischer Art übernommen wurden, einschließlich der in den elektronischen Medien veröffentlichten Quellen, unter Hinweis auf die Quelle gekennzeichnet wurden.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht, Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann.

Unterschrift: