HOCHSCHULE MERSEBURG
University of Applied Sciences

FACHBEREICH INGENIEUR- UND NATURWISSEN-SCHAFTEN

Bachelorarbeit

zum Erlangen des Grades „Bachelor of Engineering“
im Studiengang Automatisierungstechnik

über das Thema

Verwendung eines Software Defined Radio (SDR) und der Schaltungsentwicklung in GNU Radio im Anwendungsfall wM-Bus

Eingereicht von:	Kevin Saalmann
Matrikelnummer:	21442
Erstbetreuer:	Prof. Dr. rer. nat. Uwe Heuert
Zweitbetreuer:	Dipl.-Ing. (FH) Oliver Punk
Abgabeort:	Merseburg
Abgabedatum:	03.11.2022

Inhaltsverzeichnis

Abbildungsverzeichnis

Formelverzeichnis

Abkürzungsverzeichnis

ADC	Analog-Digital-Converter
AM	Amplitude Modulation
ASK	Amplitude Shift Keying
BB	Base Band
bin	binary
CI	Control Information
CRC	Cyclic Redundancy Check
CRMM	Clock Recovery Mueller & Müller
dec	decimal
DSP	digitaler Signalprozessor
FFT (Size)	Fast Fourier Transform Size (im Kontext der Fenstereinteilung)
FIR	Finite Impulse Response
FM	Frequency Modulation
FPGA	Field Programmable Gate Array
FSK	Frequency Shift Keying
GFSK	Gaussian Frequency Shift Keying
GRC	GNU Radio Companion
GSM	Global System for Mobile Communication
GUI	Graphical User Interface
hex	hexadecimal
I/Q-Signal	In Phase/ Quadrature-Signal
I-Anteil	In Phase-Anteil eines I/Q-Signals
ID	Identification
IF	Intermediate Frequency

ISM-Band	Industrial, Scientific and Medical-Band
komplex	mathematisch komplexer Wert/ Ausdruck/ Form
LMN	Local Metrological Network
LNA	Low Noise Amplifier
LoRa	Long Range
LTE	Long Term Evolution
MCU	Microcontroller Unit
MIMO	Multiple Input Multiple Output
MSK	Minimum Shift Keying
GMSK	Gaussian Minimum Shift Keying
OOK	On Off Keying
PAM	Pulse Amplitude Modulation
Q-Anteil	Quadrature-Anteil eines I/Q-Signals
RF	Radio Frequency
RX	Receiver
SDR	Software Defined Radio
SINCGARS	Single Channel Ground and Airborne Radio System
SMGW	Smart Meter Gateway
SRD-Band	Short Range Device-Band
STM32	32-Bit Microcontroller von STMicroelectronics
STM32-Spirit	Funkadapter für den STM32
Transceiver	Transmitter und Receiver in einem
TX	Transmitter
UKW	Ultrakurzwelle
UMTS	Universal Mobile Telecommunications System
VHF AM	Very High Frequency Amplitude Modulation
wM-Bus	Wireless Meter-Bus

0 Einleitung

In der heutigen Zeit ist die digitale Umsetzung von Informationsmedien sämtlicher Art leicht realisierbar geworden und zu einem großen Vorteil unserer modernen Gesellschaft gewachsen. Durch eine gängige Vernetzung verschiedener Systeme wird es uns ermöglicht im stetigen Informationsfluss zu leben. Ein Teilgebiet der Nachrichtenübertragungstechnik digitaler Informationsgehalte befasst sich mit dem Anwenden „digitaler Funktechnik“.

Diese Verfahren sind Träger verschiedener Arten digitaler Medien. Besonderes Augenmerk unserer voranschreitend automatisierten Industrie ist dabei eine drahtlose Kommunikation zwischen mehreren Geräten, wie sie beispielsweise beim Erfassen von Messwerten digitaler Strom-, Wärme-, Wasser-, oder Gaszähler verwendet wird.

Um solche Übertragungsverfahren zu entwickeln, benötigt es, auf Grund von Software Defined Radio-Applikationen, keine hohe Anzahl an preisintensiven Gerätschaften mehr. Neben dem wirtschaftlichen Aspekt bietet das Thema SDR eine weitreichende und tragende Rolle in der Ausbildung einzelner Gesellschaftsgruppen im Umgang aber vor allen in der Entwicklung, Entdeckung und Erforschung der bereits gängigen aber auch neu zu entwickelnden Übertragungsverfahren. Durch die digitale Umsetzung der Schaltungsentwicklung und deren Erprobung, ist es zu dem ein Leichtes die gewonnenen Informationen weiterzutragen. Somit können andere Personen oder Programme an dem Entwickelten teilhaben und gegebenenfalls leicht zugänglich mitwirken. Aus diesem Grund soll diese Arbeit zeigen, wie es möglich ist beispielsweise die Funkkommunikation von digitalen Energiezählern, welche das Übertragungsprotokoll wM-Bus nutzen, in eine für den PC, und somit weiterführend dem Menschen und anderen Programmen, Form der Verwertbarkeit zu bringen.

1 Software Defined Radio

1.1 Was ist SDR? - Allgemeine Erläuterung des Begriffs „Software Defined Radio“

Um eine Schaltung oder ein Gerät für den Signalaustausch via Funkwellen zu entwerfen, bedurfte es vor geraumer Zeit eines recht hohen Ensembles entsprechender Gerätschaften, sowie die allgemeinen Bauteile in analoger Form und passende Fertigungswerkzeuge. Durch geschickte Planung und eine im angemessenen Rahmen variable Testschaltung konnten Anpassungen bzw. Fehlerkorrekturen mit mehr oder weniger erhöhtem Aufwand vorgenommen werden. Erhielt man das gewünschte Endresultat, wurde die Schaltung zur Fertigung in Produktion gegeben. Nachträgliche Änderungen an den Geräten waren kaum möglich. Der generell dafür zu betreibende Aufwand wird als normal verbucht und in die Gesamtkostenkalkulation für ein Projekt miteingeschlossen.

Eine zeitgemäße Alternative bietet hingegen das unter dem allgemeinen Begriff bekannte Thema „Software Defined Radio“.

Digital realisierbare Demodulation/ Modulation von Funksignalen, allgemeine Schaltungsentwicklung auf softwarebasierter Ebene äquivalenter Analogbauteile und enorm weiterführende Funktionen, hohe Flexibilität des entwickelten Endgeräts durch Protokollwechsel oder Softwareupdates (welche dadurch auf fast alle Bereiche des Geräts Wirkung zeigen können), direkte digitale Anbindbarkeit an weitere Programme oder Geräte auf Grund zum Beispiel einer möglichen Binärkodierung demodulierter Signale, Kosteneinsparung durch günstigere Gerätschaften und die allgemein geringere Anzahl dieser (im Vergleich zur reinen Analogfertigung), sowie Arbeitszeitersparnis. All diese und viele weitere Gegebenheiten sind Motivationsgründe sich mit dem SDR-Technologien zu befassen und diese weiterzuentwickeln.

Im Allgemeinen versteht man unter dem Thema Software Defined Radio eine zum großen Teil digital umgesetzte Variante eines Funkempfängers/ -senders. Eine dafür vorgesehene Hardware ist dennoch notwendig. Diese bildet die Schnittstelle von der zu sendenden oder empfangenen Antenne zum PC.

1.2 Historischer Aspekt zum Thema SDR

Eine erstmalige Erwähnung eines „digitalen Empfängers“ trat 1970 im Rahmen eines Forschungslabors (Gold Room) des US-Verteidigungsministeriums auf. Es entsprach einem Software-Basisband-Analysetools mit dem Namen „Midas“.

Die erste Erwähnung von „Software-Radio“ wurde 1984 von E-Systems Inc. getroffen. Signifikant für diesen digitalen Basisbandempfänger waren unter anderen eine programmierbare Interferenzunterdrückung und die Demodulation von Breitbandsignalen.

1990-91 baute E-Systems Melpar den Prototypen eines taktischen Terminals für Kommandeure und erfand unter Joe Mitola 1991 den Begriff „Software-Radio“ unabhängig neu. Dieser Prototyp fand allerdings nicht lange Bestand, denn man ersetzte seine C30-Karten durch herkömmlich HF-Filterung und SpeakEasy durch ein digitales Basisbandradio.

SpeakEasy war das erste militärische Programm zur Überwachung und Kontrolle von Bodenlufteinheiten der US-Luftwaffe. Mit diesem Projekt konnten diverse Protokolle und Modulationsarten (z.B. VHF AM, frequenzagiles UWK, SINCGARS) entsprechend verarbeitet werden. Das unterstütze Spektrum betrug dabei 2MHz bis 2GHz. Neben der Modulation und Demodulation von bereits implementierten Militärfunkprotokollen, gab es die entscheidende Möglichkeit neue Modulationen und Protokolle zu integrieren.

Die erste Erwähnung eines „Software Defined Radio“ trat 1996 durch Stephen Blust auf der ersten Sitzung des MMITS-Forums (Modular Multifunction Information Transfer Systems) auf.

1998 entwickelte Nutaq in Zusammenarbeit mit MathWorks die erste Entwicklungsumgebung für ausführbare Dateien aus einem Simulink-Modell. Für die damalige Entwicklung bot dieses Tool einen enormen Innovationsschritt für das Schreiben von Code für eingebettete Systeme. DSP und FPGA wurden auf einen sogenannten Signal Master zusammengefasst. Das Konzept stellte die erste kommerzielle SDR-Entwicklungsplattform für Fakultäten dar.

Im Jahr 2001 fand ein weiterer Meilenstein für die Entwicklung von SDR-Konzepten statt – die Gründung von GNU Radio durch Eric Blossom. Rasch entwickelte sich dieses Programm zu den beliebtesten PC-Entwicklungstool für SDR-Geräte, weil es, durch seine bereits implementierten Funktionseinheiten für verschiedenste Protokolle, eine breite Masse an Entwicklern begeistern konnte.

In den nachfolgenden Jahren erschufen Firmen wie Texas Instruments in Kooperation mit Xilinx oder Lime Microsystems immer leistungsfähigere Gerätschaften, einzelne Komponenten oder Softwarelösungen, um den rasant wachsenden Anforderungen an Funk und Gesellschafft gerecht zu werden. [1] [2]

1.3 Allgemeiner Aufbau eines SDR-Geräts

Die Hardware eines SDR wird eher schmal gehalten, da die definierbaren Parameter für gezielte Anwendungen softwareseitig passieren. Dennoch ist die Hardware der essenzielle Einstieg und Abschnitt für alle weiteren Signalverarbeitungsschritte. Analoge Komponenten eines solchen Geräts sind zum einen die Antenne, welche nach dem Frequenzbereich und der Art des Signals gewählt werden sollte. Diese dient als eine Art Bandpassfilter des Frequenzspektrums. Betrachtet man das Gerät als Empfänger, empfängt die Antenne die elektromagnetischen Funkwellen und leitet sie, nachdem diese einen Verstärker (Amplifier) durchlaufen haben, an einen Analog-Digital-Wandler (ADC genannt) weiter. Die gewonnenen/ abgetasteten Informationsstellen werden anschließend zur weiteren digitalen Signal-verarbeitung bereitgestellt.

Die Güte eines SDR-Geräts zeichnet sich unter anderen durch den möglichen adaptierbaren Frequenzbereich und der Abtastrate (Sampling-Rate) des ADCs aus.

Die eben erwähnte Funktionsweise beschreibt in abstrakter und verallgemeinerter Form die heutzutage gängige Variante von SDR-Geräten.

Ein wichtiges, dem Prinzip hinzuzufügendes, Arbeitsdetail, ist der Lokaloszillator. Damit sieht der gängige Aufbau komplettiert wie folgt aus:

Das zu untersuchen wollende breitbandige (mathematisch) komplexe Signal wird von der Antenne aufgefangen. Anschließend wird dieses durch einen Amplifier verstärkt und zum ADC weitergeleitet. Damit der ADC das komplexe Signal verarbeiten kann, bedarf es der Hilfe des Lokaloszillators und einer Phasenmodifikation. Dessen Aufgabe ist es ein I/Q-Signal (genauere Erläuterung dazu in den folgenden Kapiteln zu finden) zu erzeugen. Durch eine Aufteilung des den ADCs zugeführten komplexen Signals, können beide Kanäle (I und Q) separat abgetastet werden, unter der Voraussetzung, dass der zuvor agierende Lokaloszillator einen Anteil (Q-Anteil) phasenversetzt weiterführt. Gesamtgesehen ist es somit möglich den kompletten Informationsgehalt des komplexen Signals zu erhalten und weiterzuverarbeiten.

1.3.1 Unterschiedliche SDR-Funktionsarten

Neben der heutzutage gängigen Variante, werden SDRs generell in 3 Arten unterteilt, welche sich in ihren Funktionsprinzipien unterscheiden. [3]

1.3.1.1 Direkte Digitalisierung

Eine Möglichkeit ist die direkte Digitalisierung des Eingangssignals. Hierbei wird das Hauptaugenmerk auf die Abtastrate nach dem Nyquist-Shannon-Theorem gelegt, welches besagt, dass die Abtastrate größer sein muss als das doppelte der höchsten vorkommenden Frequenz des relevanten Spektrums. Die im Vorfeld passierte analoge Verarbeitung mit

Hinsicht auf Filterung, Verstärkung oder Dämpfung des Signals, wird eher schmal gehalten. Eine konkrete Analyse, eines für einen spezifischen Anwendungsfall relevanten Signalanteils des großen Frequenzspektrums, ist daher nur schwer möglich. Auch bedarf es, je nach Höhe des abzutastenden Spektrums, einer hohen Abtastrate, welche hardwarebedingt, je nach eingesetztem Gerät, an seine Grenzen kommt. Wird das reine Signal mit einer zu geringen Sampling-Rate quantisiert, kann es zu einer Verfälschung des Informationsgehalts kommen.

1.3.1.2 Digitalisierung einer Zwischenfrequenz

Die zweite Methode, nach den SDRs in ihre Funktionsprinzipien eingeteilt werden, ist die Digitalisierung auf einer Zwischenfrequenzebene. Diese Variante ähnelt sehr dem Prinzip eines Überlagerungsempfängers. Dabei wird, wie auch heute bei der gängigen Methodik üblich, ein Lokaloszillator eingesetzt, um das empfangene hochfrequente Signal auf ein geringeres Frequenzniveau, aber bei gleichem Informationsgehalt, herabzumischen. Die Zusammenführung dieser beiden Frequenzen ergibt eine resultierende Zwischenfrequenz, welche anschließend gefiltert, abgetastet und demoduliert werden kann. Somit ist auch die anzuwendende Verstärkung eines selektiven Bereichs, sinnvollerweise der Bereich des zu nutzen wollenden Informationsgehalts, effizienter zu gestalten. Auch sind keine immens hohen Abtastraten notwendig, da die höchste vorkommende Frequenz verringert wurde. Eine höhere Trennschärfe und ein damit verbunden leichterer und wohlmöglich höherer Gewinn des Informationsgehalts ist das Resultat. Auch eine verringerte Anforderung an eine Großsignalfestigkeit wird dadurch reduziert, dass die Filter nur auf die Nutzsignalbandbreite ausgelegt werden müssen.

1.3.1.3 Direktmischer nach I/Q-Verfahren

Bei der dritten Variante eines SDRs beschreibt das Prinzip einen Direktmischer nach dem I/Q-Verfahren. Dabei wird, ähnlich wie bei der Digitalisierung auf der Zwischenfrequenzebene, das empfangene hochfrequente Signal mit einem Lokaloszillatorsignal gemischt und somit der entstehende Bereich rein auf den Nutzbereich beschränkt, wenn das resultierende Signal in das Basisband verschoben wird. Die Lokaloszillatorfrequenz wird dabei so gewählt, dass sie den gleichen Wert wie die Trägerfrequenz des zu demodulieren wollenden Signals besitzt. Auch hierbei entsteht, auf Grund der Mischung, ein niederfrequentes Signal, welches ebenfalls keine hohen Anforderungen an eine Großsignalfestigkeit mehr besitzen muss. Ebenfalls fällt die aufzuwendende Abtastrate nach dem Nyquist-Shannon-Theorem deutlich geringer aus. Allgemein wird durch die NF-Verstärkung und die präzisere Selektion des zu nutzenden Bereichs im Spektrum eine hohe Trennschärfe erreicht. Um, wie bei herkömmlichen Direktmischern üblich, eine bedingte Spiegelfrequenz zu unterdrücken, wird das transformierte Signal mit dem I/Q-Verfahren bearbeitet. Eine Spiegelfrequenz entsteht bei der Mischung eines Empfangssignals mit einem Oszillatorsignal. Das große Problem dabei ist, dass ein

Signalabschnitt unterhalb der Oszillatorfrequenz den gleichen Anteil im positiven Bereich als Ausganssignal bildet. Lösung des Problems bietet das I/Q-Verfahren bei dem parallele Mischstufen, eine davon 90° phasenversetzt, angewendet werden. Eine genauere Erläuterung zu diesem Verfahren befindet sich im Abschnitt 1.4.

1.4 I/Q-Signale/-Verfahren und Quadraturmodulation

Um ein Signal gänzlich auszuwerten, reicht es in den meisten Demodulationsverfahren (äquivalent beim Erzeugen eines Signals, also bei einer Modulation) nicht aus allein die einfache Amplitude abzutasten oder den zeitlichen Versatz der Wellenberge oder -täler zu messen. Dieser alleinige Informationsgehalt wäre in den meisten Modulationsarten trügerisch und verfälschend. Für den vollen Umfang des Empfangssignals ist in den meisten Fällen die Phasenlage zusätzlicher Informationsgeber. Da es sich bei einem Funksignal immer um ein komplexes Signal, also bestehenden aus einem Real- und Imaginärteil, handelt, ist es sinnvoll beide Komponenten auszuwerten.

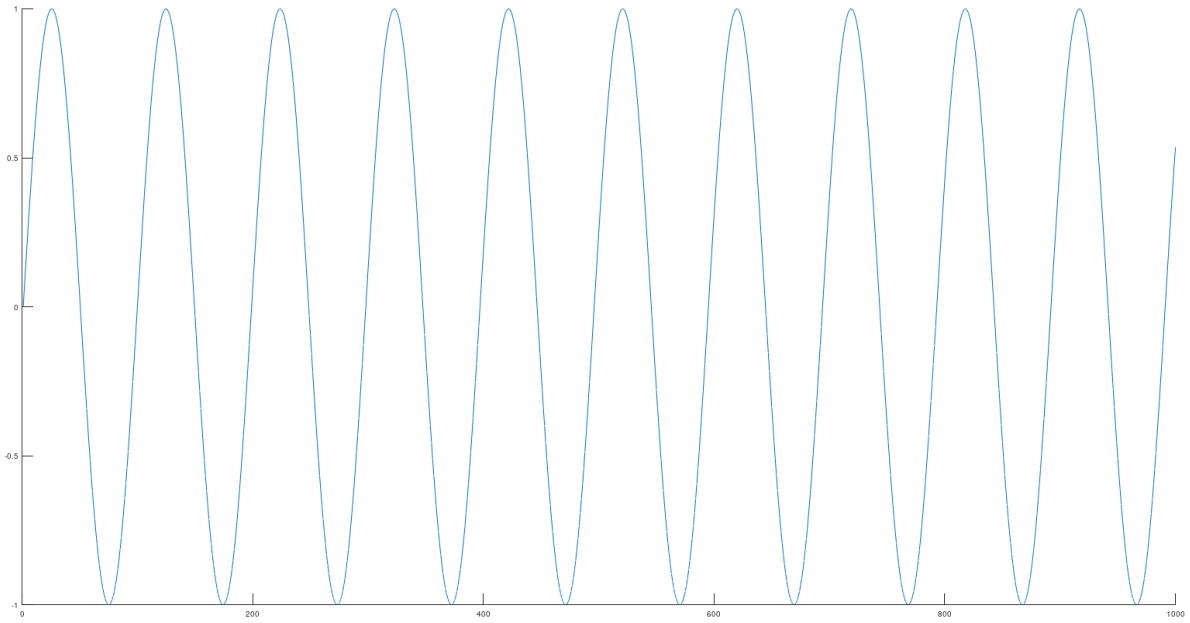


Abbildung 1: Sinusschwingung in 2-dimensionaler Darstellung mit Augenmerk auf Amplitude im zeitlichen Verlauf (Quelle: Eigene Darstellung)

In der gezeigten Grafik (Abbildung 1) sind die gesampelten Amplitudenwerte eines herkömmlichen Sinussignals dargestellt. Trügerischerweise könnte man hierbei die Annahme treffen Signale allein anhand der Periodenlänge, daraus resultierend dessen Frequenz, und dem Amplitudenverhalten bestimmen zu können. Eine dazu allerdings noch fehlende und wichtige Information, für viele Signalarten, ist die Frequenzrichtung, welche sich allein aus gesampelten Werten der zweidimensionalen Amplitude nicht gewinnen lässt. Eine in der

Praxis üblich vorgenommene Mischung zweier Signale, führt dadurch zu zwei Lösungen, also einem unklaren Ergebnis.

Das I/Q-Verfahren betrachtet demnach nicht nur die Perspektive im zweidimensionalen Bereich, sondern es wird noch eine weitere räumliche Komponente dazu genommen, wie in Abbildung 2 zu sehen ist.

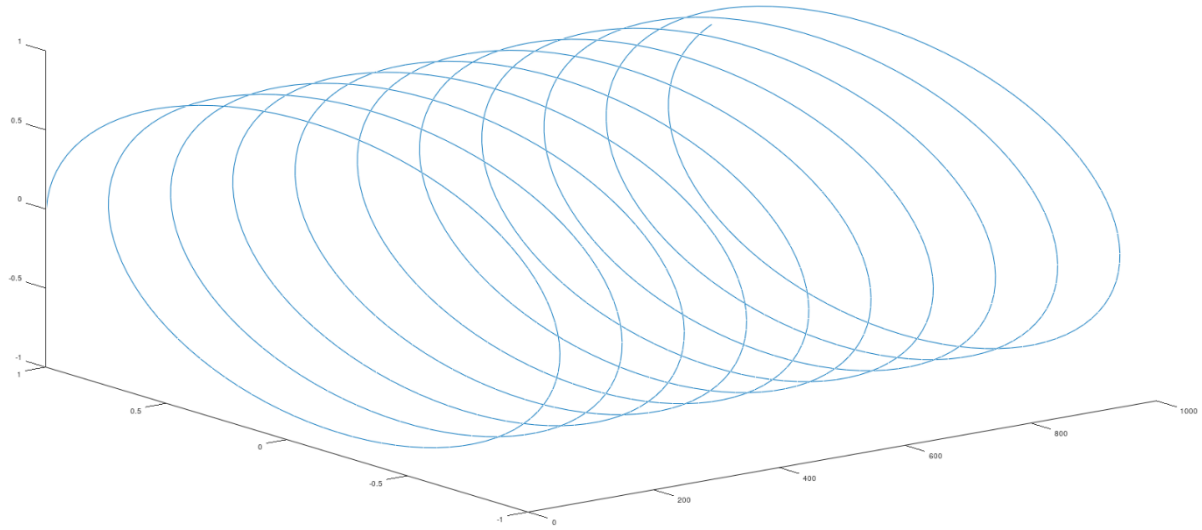


Abbildung 2: Sinusschwingung in 3-dimensionaler Darstellung mit Augenmerk auf Amplitude und Phasenlage im zeitlichen Verlauf
(Quelle: Eigene Darstellung)

Um dieses Prinzip zu veranschaulichen, wurde eine kleine Schaltung/ Flowgraph in GNU Radio erstellt:

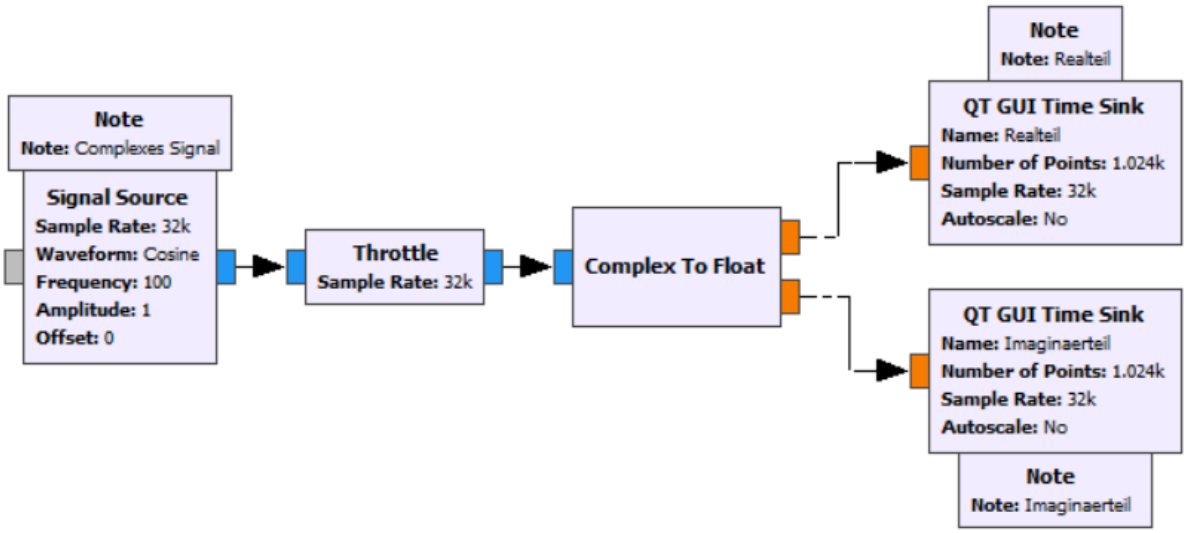


Abbildung 3: GRC-Flowgraph für separate Ansicht und vergleich des Real- und Imaginärteils
(Quelle: Eigene Darstellung)

In diesem wird durch einen Signalgenerator ein komplexes Kosinussignal mit einer positiven Frequenz von 100Hz erzeugt. Anschließend wird dieses Signal in seinen Real- und Imaginärteil separiert. Das Ergebnis im Zeitverlauf sieht wie folgt aus:

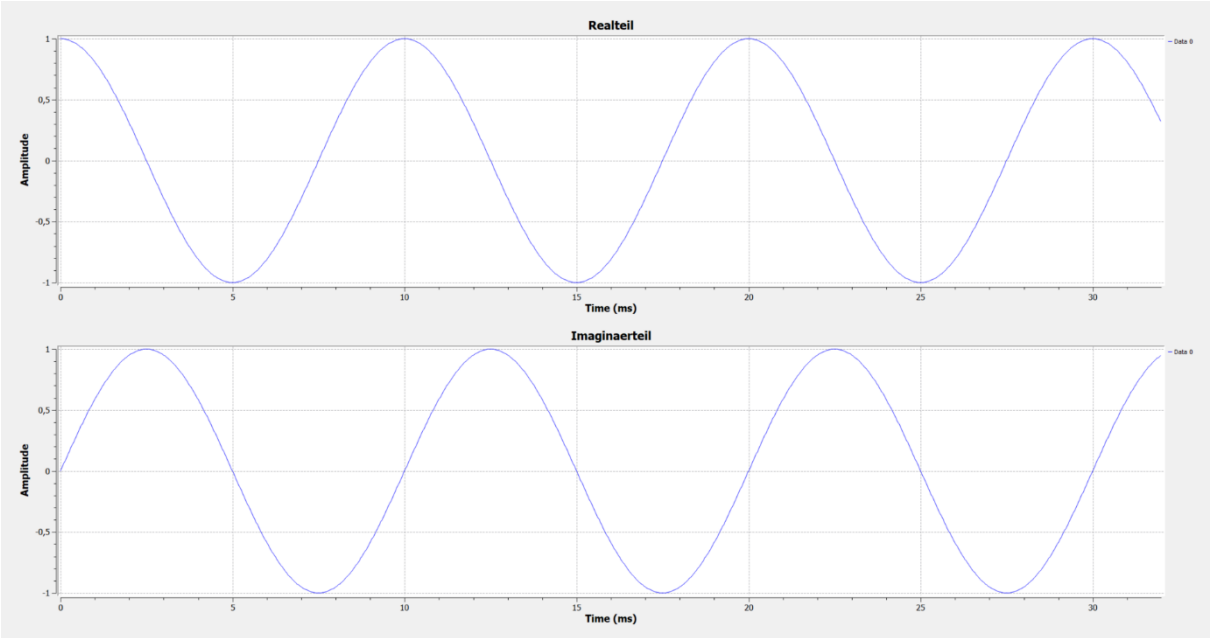


Abbildung 4: Phasenversatz zwischen Real- und Imaginärteil
(Quelle: Eigene Darstellung)

Man sieht hierbei deutlich die um 90° versetzte Phasenlage des Q-Anteils, also den Imaginärpart, wobei hingegen der I-Teil, also der Realpart des komplexen Signals, phasenseitig nicht verändert wurde.

Das Prinzip lässt sich ebenfalls veranschaulichen, wenn man sich die Spiraldarstellung des Signals (Abbildung 3) von vorn und oben vorstellt. Dabei zeigt die Ansicht aus der vorderen Perspektive das I-Signal und die Draufsicht das Q-Signal. Zusammengeführt ergibt dies die Helix.

Der I-Teil wird auch als In-Phase-Signal und der Q-Teil als Quadratur-Signal bezeichnet.

Um mit einem daraus hervorgehenden Signal arbeiten zu können, in dem man keinen geteilten Imaginär- und Realteil hat, wird die Quadraturmodulation angewendet, welche aus den beiden separaten Formen eine zusammengeführte komplexe Zahl bildet. Diese Umwandlung ist notwendig um mit diversen Modulationsverfahren wie FM, FSK, GMSK arbeiten und umgehen zu können.

Dabei wird das Produkt, aus dem um eine Abtastung verzögerten und konjugierten Signals (Q-Teil) und dem nicht verzögerten Signal (I-Teil) berechnet. Das Argument, als Winkel im Bogenmaß bekannt, bezeichnet einen resultierenden Part einer komplexen Zahl und gibt

Aufschluss über den Informationsgehalt der Phasenlage. Eine weiterführende Beschreibung dessen lässt sich im Praxisteil dieser Arbeit finden. [4]

1.5 Frequency Shift Keying (FSK)

Ein grundlegendes, und auch für diese Arbeit relevantes, Frequenzmodulationsverfahren in der elektronischen Nachrichtenübertragung beschreibt das Verfahren des Frequency Shift Keyings (FSK), zu Deutsch als Frequenzumtastung bekannt.

Da die Realisierung eines FSK-Signals vergleichbar einfach bewerkstelligt werden kann, wird dieses Verfahren gern für simple, rein analoge Geräte verwendet. Dazu sind keine komplizierten softwareseitigen Aufwendungen nötig.

Der Kerngedanke einer FSK beschreibt eine konstante Einhüllende, also eine konstante Amplitude während des Signalverlaufs, und lediglich von einer Trägerfrequenz ausgehend variierende Informationsgehaltfrequenzen. Der digitale Informationsgehalt wird der Trägerfrequenz aufmoduliert, wobei einer digitalen 1 und 0 jeweils eine bestimmte Frequenz zugeordnet wird. Diese sind symmetrisch zur Trägerfrequenz angesiedelt und werden auch im allgemeinen als FSK-Hub oder Frequency Deviation bezeichnet.

Innerhalb der FSK-Modulation gibt es Unterscheidungsfälle. Je nach Anwendung und Anforderung werden diese entsprechend gebraucht. Bei der einfachsten Form handelt es sich um die 2FSK. Dabei werden zwei feste Frequenzen, in symmetrischer Abweichung zur Trägerfrequenz, dem digitalen Informationsgehalt (0 oder 1) zugeordnet. Eine hohe Frequenz stellt dabei eine 1 und die niedrige Frequenz eine 0 dar. Eine weitere Methodik, um den Informationsgehalt in einer schnelleren und somit verbundenen effizienteren Form zu übertragen, ist die 4FSK. Das Prinzip der symmetrisch applizierten Informationsfrequenzen bleibt gleich. Hier werden allerdings 4 anstatt der bisher erwähnten 2 Frequenzen verwendet. Jede Frequenz überträgt dabei ein Dibit in Form von 00, 01, 10 oder 11. [5]

Durch die amplitudenirrelevante Variante einer elektrischen Nachrichtenübertragungs-modulation, ist dieses Verfahren störungsunempfindlicher als z.B. AM-Modulation. Schwankungen in der Amplitude können durch äußere Einflüsse, durch Störungen oder Gütegerade des Verstärkers auftreten. Es ist somit nicht notwendig einen linearen Verstärker zu verwenden. Dadurch können Kosten und Aufwand gespart werden.

GFSK ist eine weitere Variante des Frequency Shift Keyings und steht für eine Gaußsche Frequenzumtastung. Das Kernprinzip der regulären FSK bleibt dabei gleich, doch wird hierbei ein Gauß-Filter verwendet. Durch das direkte Modulieren digitaler Signale auf eine Trägerfrequenz können hohe Frequenzspitzen durch die steilen Flanken des

Informationssignals entstehen. Um damit ungewollten höheren Bandbreiten und ein unerwünschtes Nebensprechen (eingekoppeltes Signal von Nebenfrequenzen oder anderen Interferenzen) zu vermeiden, werden die digitalen Signale verschliffen und so hochfrequente Anteile rausgefiltert. [6]

Auch bei der GFSK-Methode gibt es wieder Unterscheidung in 2- bzw. 4-Frequenzabweichungen, also 2GFSK und 4GFSK.

Die Gegenüberstellung des digitalen Informationsgehalts und einem daraus generierten FSK-Signal lässt sich gut durch folgenden, in GNU Radio erzeugten, Flowgraph zeigen.

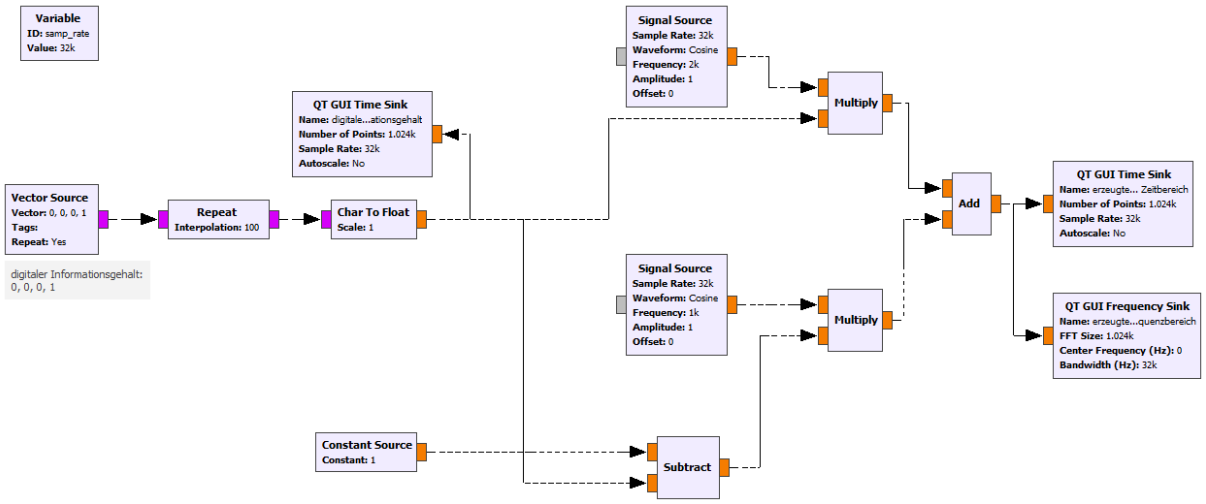


Abbildung 5: GRC-Flowgraph eines einfachen FSK-Signalgenerators
(Quelle: Eigene Darstellung)

In dieser Schaltung wird ein FSK-Signal mit einem Frequenzhub von ±500Hz erzeugt. Die Grundlagen der einzeln zusammengeführten Signale (unterer und oberer Frequenzbereich, welche repräsentativ für 0 und 1 stehen) werden hier verdeutlich.

Die darzustellen wollende Information steht in dem Block „Vector Source“. Diese ist eine Bitfolge von 0,0,0,1 und generell frei wählbar.

Die Ausgänge der Schaltung ergeben folgendes:

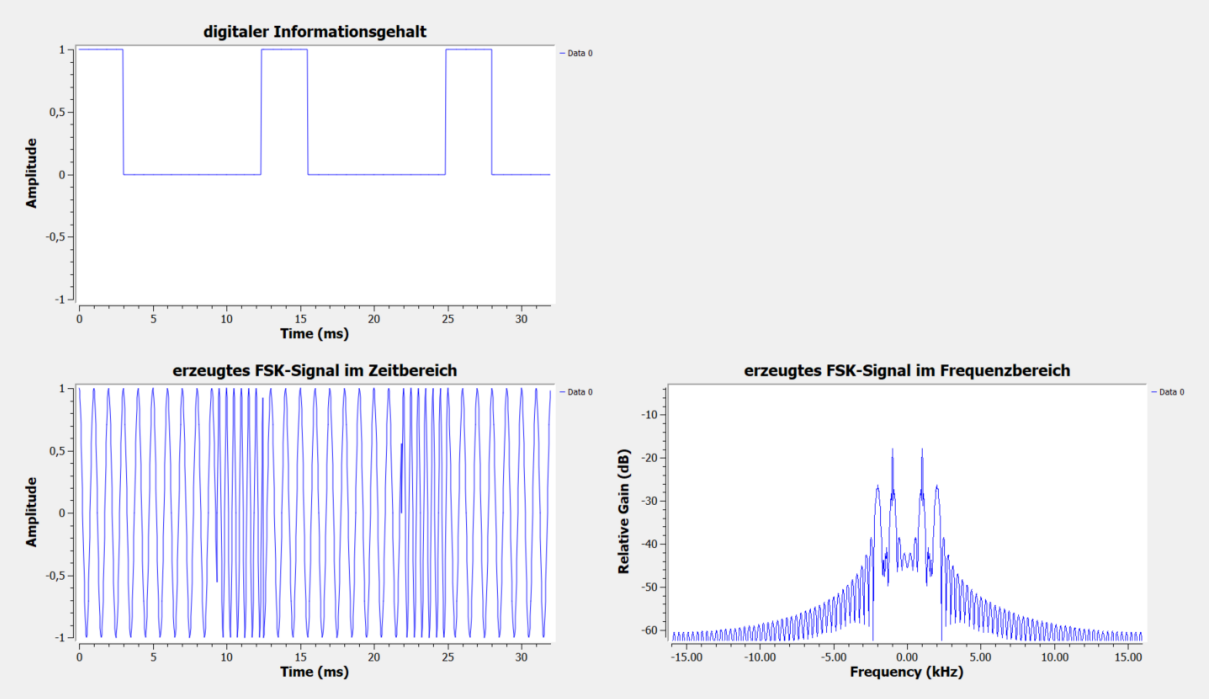


Abbildung 6: Ausführung und Anzeigeelemente des einfachen FSK-Signalgenerators unter Abb. 5 (Quelle Eigene Darstellung)

In Abbildung 6 ist der Informationsgehalt von 0,0,0,1 aus der Vector Source gut sichtbar als reines digitales Signal, also variierend zwischen 0 und 1, dargestellt, sowie zusätzlich die modulierten Frequenzen. Durch diese direkte Gegenüberstellung zeigt sich besonders deutlich, dass eine digitale 1 für einen Abschnitt höherer Frequenz steht und eine digitale 0 für einen Abschnitt niedrigerer Frequenz. Rechts unten im Bild sieht man das Signal im Frequenzspektrum. Dies ist ein typisches Erscheinungsbild für eine FSK. Erkennbar sind hier die beiden Trägerfrequenzen, deren Oberwellen und die Deviation.

Interessehalber wurde das Signal im Flowgraph zusätzlich mit einer „File Sink“ aufgenommen, um dieses mit dem externen Tool „Inspectrum“ anschauen und analysieren zu können. Hinsichtlich konkreterer Signalkomponenten wie: Detailanalyse, Symbolrate, Periodendauer, etc., ist dieses Tool, aus meiner Sicht, sehr effektiv, interessant und hilfreich gestaltet.

Die Darstellung in Inspectrum sieht wie folgt aus:

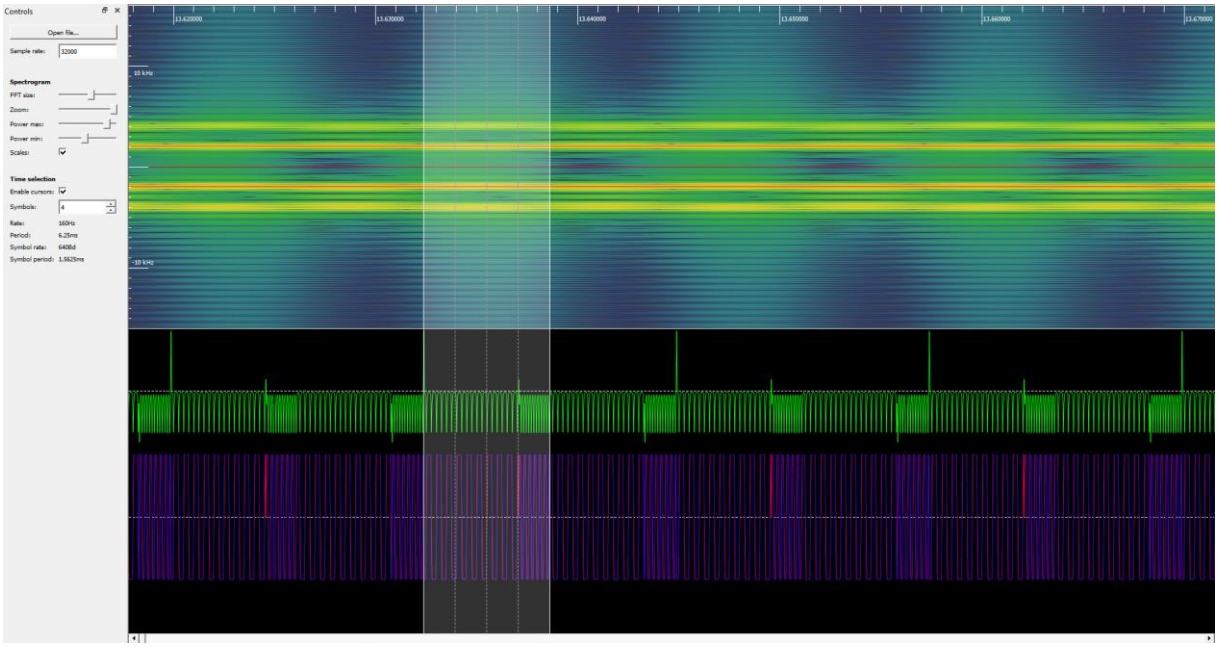


Abbildung 7: Aufgezeichnetes Signal des FSK-Signalgenerators mit Frequenz- und Phasenplot in Inspectrum (Quelle: Eigene Darstellung)

1.6 Im Rahmen der Arbeit genutzte Hardware

1.6.1 LimeSDR-USB

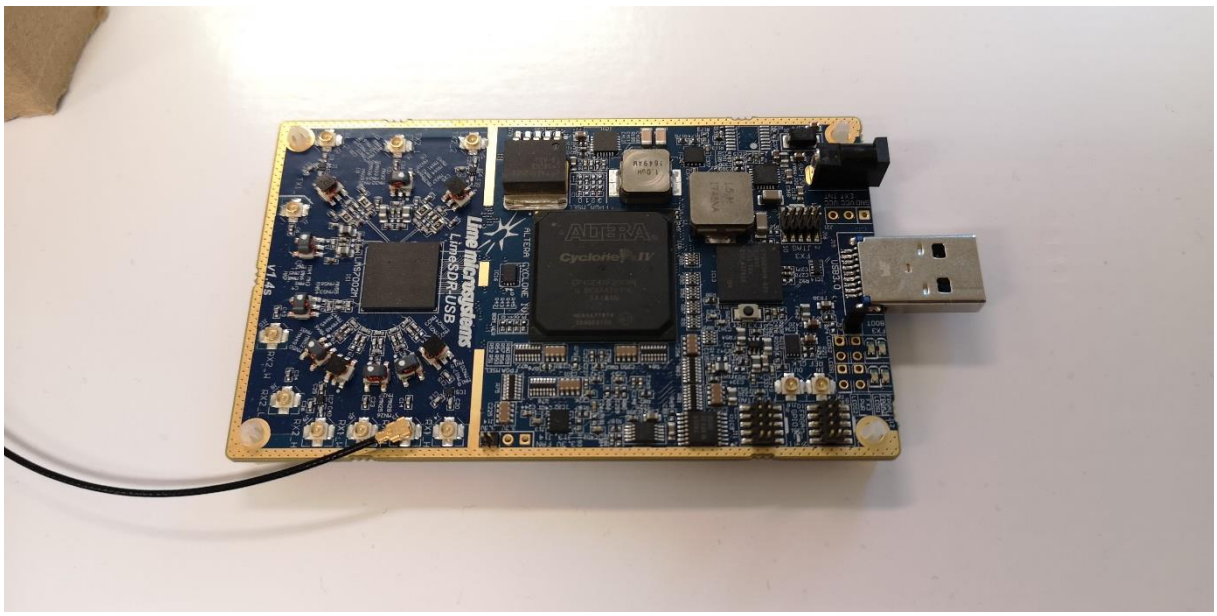


Abbildung 8: LimeSDR-USB
(Quelle: Eigene Darstellung)

Der LimeSDR-USB stellt das genutzte Hardwarekernstück dieser Arbeit dar. Das kraftvolle SDR-Gerät lässt RF-Signale in einem Spektrum von 100MHz bis 3,8GHz empfangen und

senden bei einer maximalen Bandbreite von 61,44MHz. Die Verbindung der gesampelten und quantisierten Daten und dem PC, stellt die Platine über eine USB3.0-Verbindung bereit. Auf dem Board stehen 10 verschiedene Antennenanschlüsse, welche in 6 RX- und 4 TX-Kanäle unterteilt sind, zur Verfügung. Diese sind wiederrum für verschiedene Frequenzbereiche vorgesehen. Ein Multiplexing-Verfahren von 2x2-MIMO ist zudem mit diesem Gerät realisierbar. [7]

Es ist mit diesem Board möglich, RF-Protokolle wie z.B. UMTS, LTE, GSM, LoRa, wM-Bus, ZigBee u.v.m. zu empfangen und zu senden.

Gewählt wurde der LimeSDR-USB für diese Arbeit, weil die Firma exceeding solutions GmbH diesen empfahl und er in den Frequenzbereich, sowie für künftige Arbeiten mit dem Thema Software Defined Radio passt und generell ein kraftvolles Entwicklungsboard für einen angemessenen Preis darstellt.

In diesem Versuchsausbau wurde dieses SDR als Empfänger genutzt.

1.6.2 STM32 Nucleo mit S2-LP expansion board als wM-Bus-Dummy

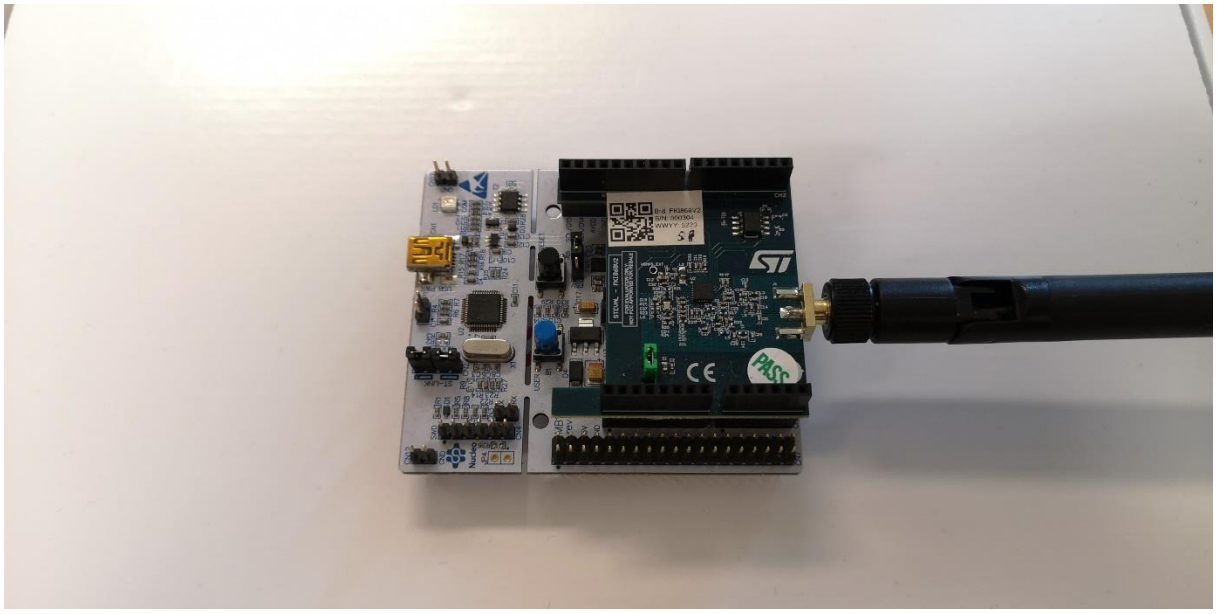


Abbildung 9: STM32 mit Funkaufsatz als wM-Bus-Dummy
(Quelle: Eigene Darstellung)

Bei diesem Gerät handelt es sich um ein Entwicklungsboard mit MCU und entsprechender Funkhardwareerweiterung mit SDR-Chip. Dieser wurde von exceeding solutions GmbH mit einem wM-Bus-Protokoll programmiert und dem Gesamtversuchsaufbau zur Verfügung gestellt.

Die dabei wichtigen Konfigurierungsparameter, um das im Versuch demodulierte Signal zu finden/kontrollieren, sind:

1. Modus: T1-T2 meter to other
2. Präambellänge: 0
3. Postambellänge: 2
4. Frequenzbasis: 868.95MHz
5. Datenrate: 100kBd
6. Freq.-Deviation: 50kHz
7. Bandbreite: 260kHz
8. Modulation: 2FSK

In diesem Versuchsaufbau wurde dieses Gerät als Sender genutzt.

1.6.3 STM32 Nucleo mit S2-LP expansion board als Sender und Empfänger mit selbstdefinierten Übertragungsparametern

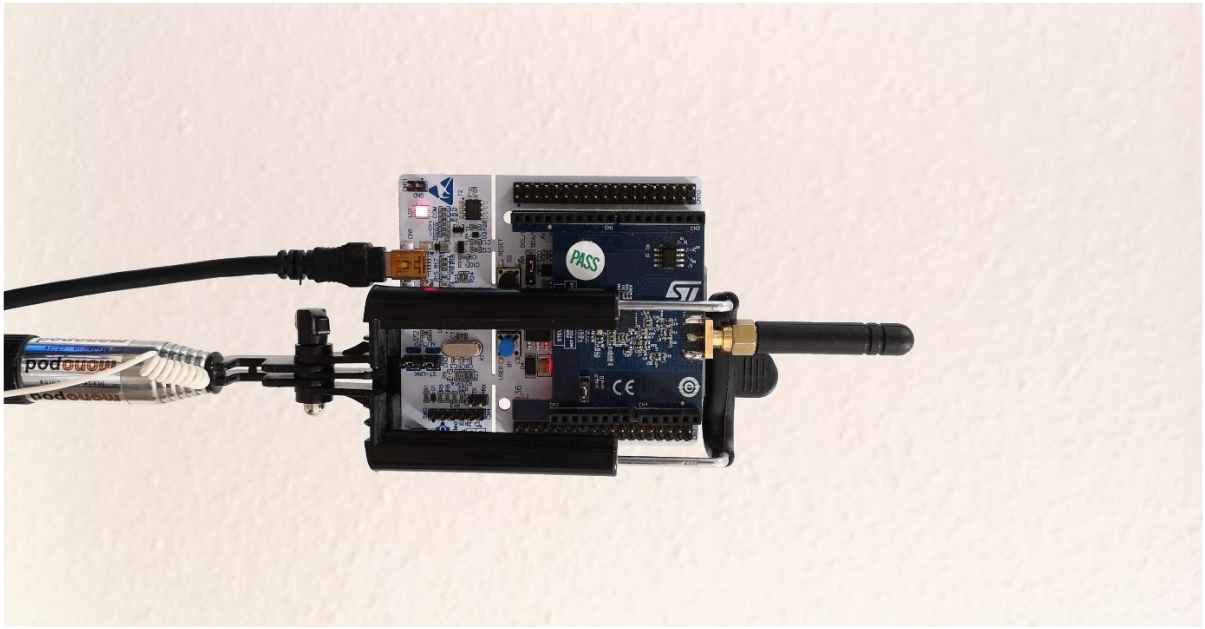


Abbildung 10: STM32 mit Funkaufsatz als selbstkonfigurierbarer wM-Bus-Transmitter und -Receiver (Quelle: Eigene Darstellung)

Ein weiterer STM32 mit entsprechender Funkerweiterung steht im Rahmen dieser Arbeit ebenfalls zur Verfügung. Dieser kann, mittels des Tools „STSW-S2LP-DK (S2-LP DK)“, frei als Sender oder Empfänger konfiguriert werden. Genutzt werden hierbei tatsächlich beide Varianten. Einmal zum Empfangen der Signale des wM-Bus-Dummy-Geräts, wobei der LimeSDR diesen Signalen mitlauschen kann und zum anderen zum Senden bestimmter Nachrichten im wM-Bus-Protokoll oder auch ohne bestimmte Protokollierung.

Dieses Gerät wurde ebenfalls von der Firma exceeding solutions GmbH dieser Arbeit zur Verfügung gestellt und dient, wie schon erwähnt, als Sender und Empfänger.

1.6.4 Fernbedienung zur Funksteckdose der Firma Heitech

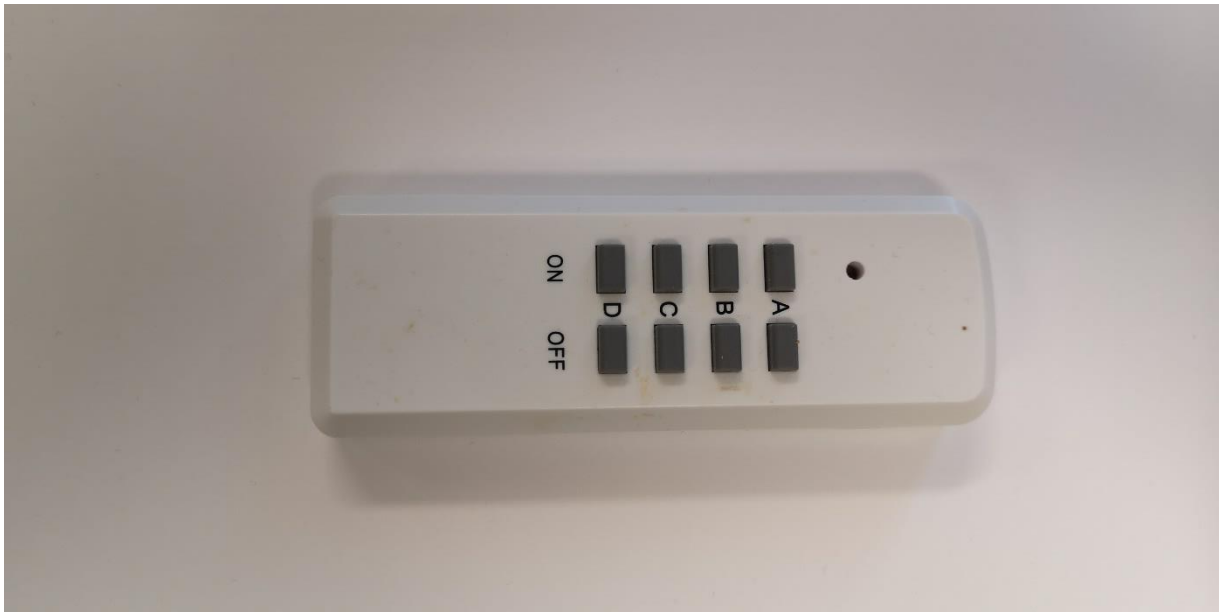


Abbildung 11: beliebige Funkfernbedienung als alternative Emitter-Quelle
(Quelle: Eigene Darstellung)

Um der wM-Bus-Protokollierung und -Modulierung eine Alternative und eine andere Form an Funksignalen zu bieten, wird eine Funkfernbedienung für entsprechende Steckdosen im Rahmen des Versuchsaufbaus verwendet. Auf der Rückseite lässt sich die Frequenz von 433,92MHz ablesen. Öffnet man das Batteriefach, erscheinen zudem 5 einstellbare Dip-Schalter. Mit diesen lassen sich die Adressaten der Signale festlegen. Generell besitzt die Fernbedienung 8 Taster (4 für ON und 4 für OFF) auf der Vorderseite.

1.6.5 Älterer Autoschlüssel

Als weiteres und letztes Zusatzversuchsobjekt wird ein etwas älterer Autoschlüssel für ein etwas älteres Auto genommen. Nach einer kurzen Recherche im Internet ließ sich die Frequenz finden.

Abbildung 12: beliebiger Autoschlüssel als alternative Emitter-Quelle
(Quelle: Eigene Darstellung)

1.7 Im Rahmen der Arbeit genutzte Software

1.7.0 Vorwort

Die Entwicklungsumgebung für das unter Windows10 verwendete SDR-Konzept stellte anfänglich diverse Probleme bereit. Das zur eigentlichen Umsetzung des Codes gedachte Programm „GNU Radio“ ist ursprünglich ein für den Linux-Kernel entwickeltes Tool. Für die Installation unter Windows wird eine Anleitung unter z.B. https://wiki.myriadrf.org/Gr-limesdr_Plugin_for_GNURadio bereitgestellt. Das auf dieser Seite unter Punkt 3.2 erwähnte Installationspaket „PhotosSDR“ stellte in diesem Projekt den letztendlichen Erfolg zur effektiven Nutzung der Programme her. Dabei gab es allerdings diverse Treiberprobleme, um den LimeSDR entsprechend der Anleitung einbinden zu können. Ein mehrmaliges Überprüfen und Ersetzen des Treibers im Gerätemanager bzw. mit dem Tool „Zadig“ brachte schlussendlich den Erfolg.

1.7.1 GNU Radio

Das ursprünglich für Linux entwickelte Tool GNU Radio ist ein freies open source-Programm für die Entwicklung von SDR-Schaltungen. Die Benutzeroberfläche stellt ein GUI (Graphical User Interface) dar und man „programmiert“ Code mit Hilfe von Informationsblöcken. Die Blöcke stellen bestimmte Funktionseinheiten dar und werden mit Linien verbunden. Im Hintergrund wird anschließend ein Gesamtkonzept/ ein einheitlicher Code aus dieser „Schaltung“ erzeugt. Es ist dennoch möglich über eine textbasierte, und damit für die meisten Anwender herkömmliche, Methode, Code zu erzeugen. Wichtig wird dieser Fakt bei der Erstellung und Implementierung eigens kreierter Funktionseinheiten.

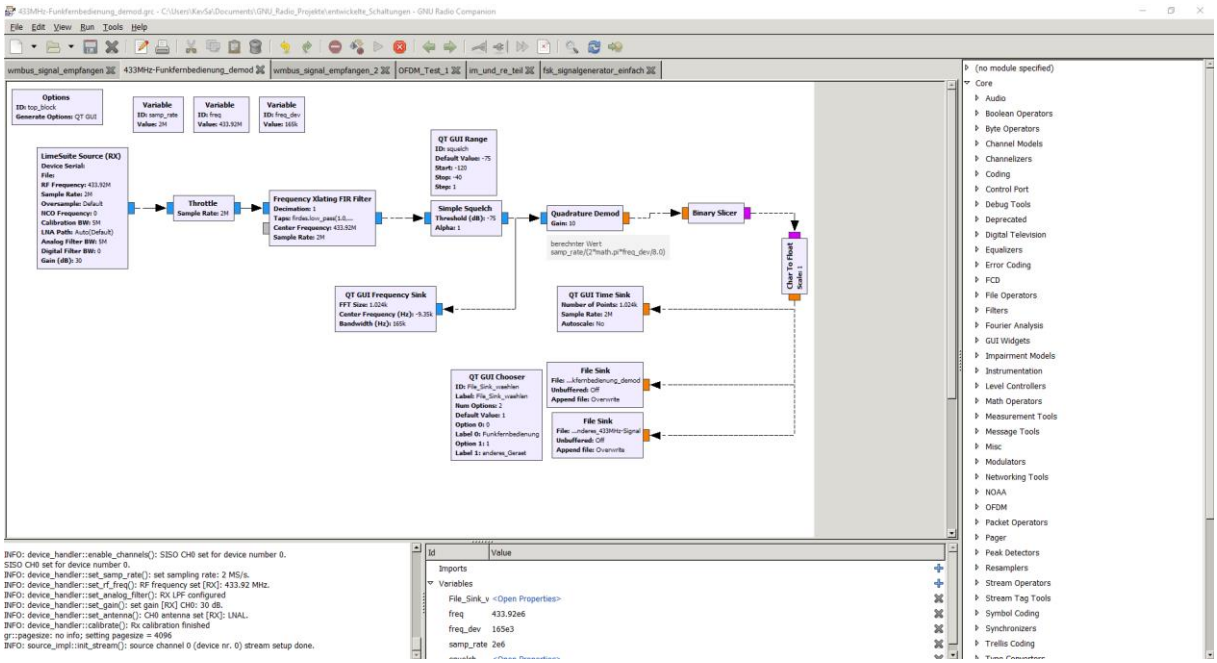


Abbildung 13: Beispielbild für GNU Radio Companion
(Quelle: Eigene Darstellung)

1.7.2 CubicSDR

CubicSDR stellt ein nützliches Tool für eine allgemeine SDR-Ausrüstung dar. Es ist mit diesem Programm möglich, ein breites, abhängig vom angeschlossenen Gerät, Frequenzspektrum abzusuchen. Plattformübergreifend bietet dieses Programm die Funktion der manuellen Suche und der automatischen Demodulation verschiedener Frequenzarten bzw. Signale. Somit lassen sich, je nach Gerät und verwendeter Antenne, diverse Funkkanäle auffangen/abhören, ohne dabei eine mehr oder minder komplizierte Schaltung entwerfen zu müssen. Diverse Tuningoptionen unterstützen dabei eine konkretere Demodulation.

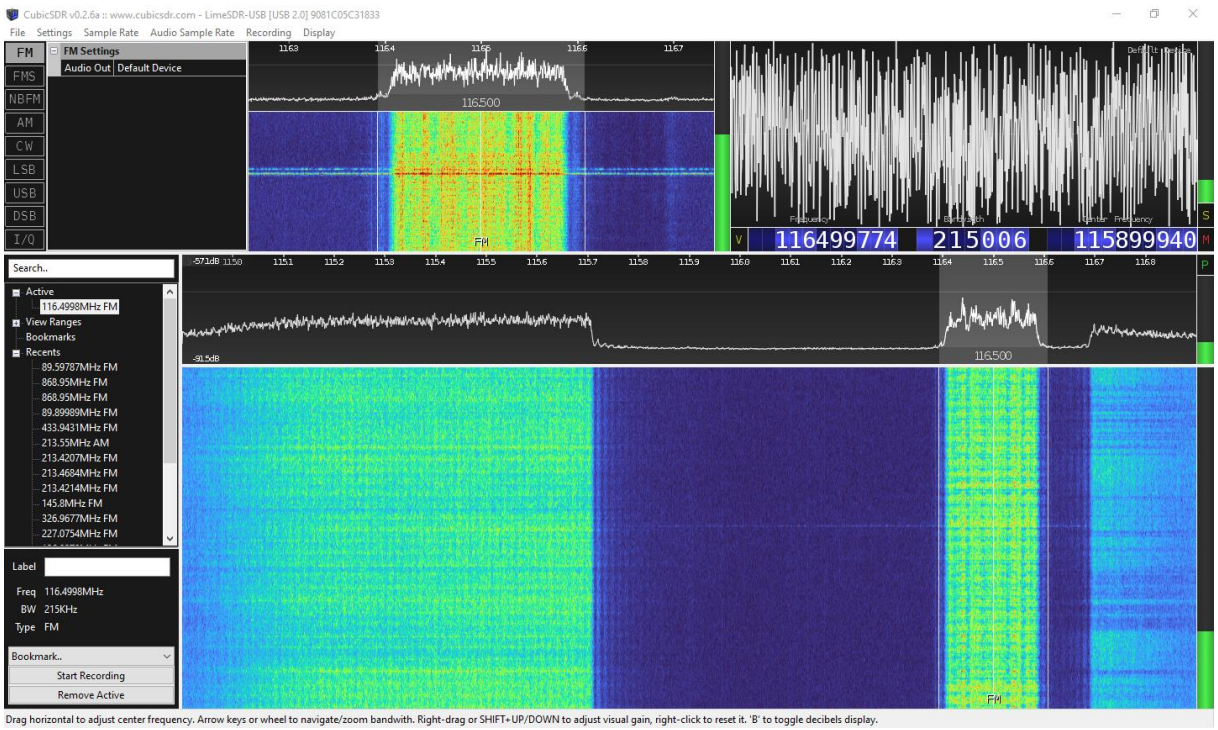


Abbildung 14: Beispielbild für CubicSDR
(Quelle: Eigene Darstellung)

1.7.3 Inspectrum

Für eine konkretere Signalanalyse eines aufgenommenen RF-Signals, wird in dieser Arbeit das Programm „Inspectrum“ verwendet. Mit diesem ist es möglich, Signale aus der aufgenommenen Probe zu selektieren, um diese genauer zu betrachten. Die Einstellungsmöglichkeiten basieren dabei auf rein optischen Werten. So ist es dem Anwender ermöglicht, manuell eine gewisse Verstärkung seiner Frequenzen vorzunehmen, um eine entsprechende Präselektion zu bewirken. Entdeckt der Anwender dadurch einen potenziell interessanten Signalbereich, kann mittels einer Zoom-Funktion das zeitliche Spektrum eingekesselt werden. Die Darstellung erfolgt aufgeteilt in 2 charakteristische Achsen – einen Frequenzbereich und einen Zeitbereich. Es wird somit ermöglicht, sich die in einer gewissen Bandbreite aufgenommenen Frequenzen in einem zeitlichen Verlauf anzuschauen. Für die genaue Analyse, hinsichtlich Symbolrate, Periodendauer, etc., kann ein Cursor eingesetzt werden, um einzelne Informationsbereiche passend abzugrenzen. Für die allgemeine Analyse des Signals stehen des Weiteren diverse Plots zur Verfügung. Diese stellen sich als Frequenzplot, Sampleplot, Amplitudenplot, Phasenplot, und zugehöriger Threshold für manche Plots, dar.

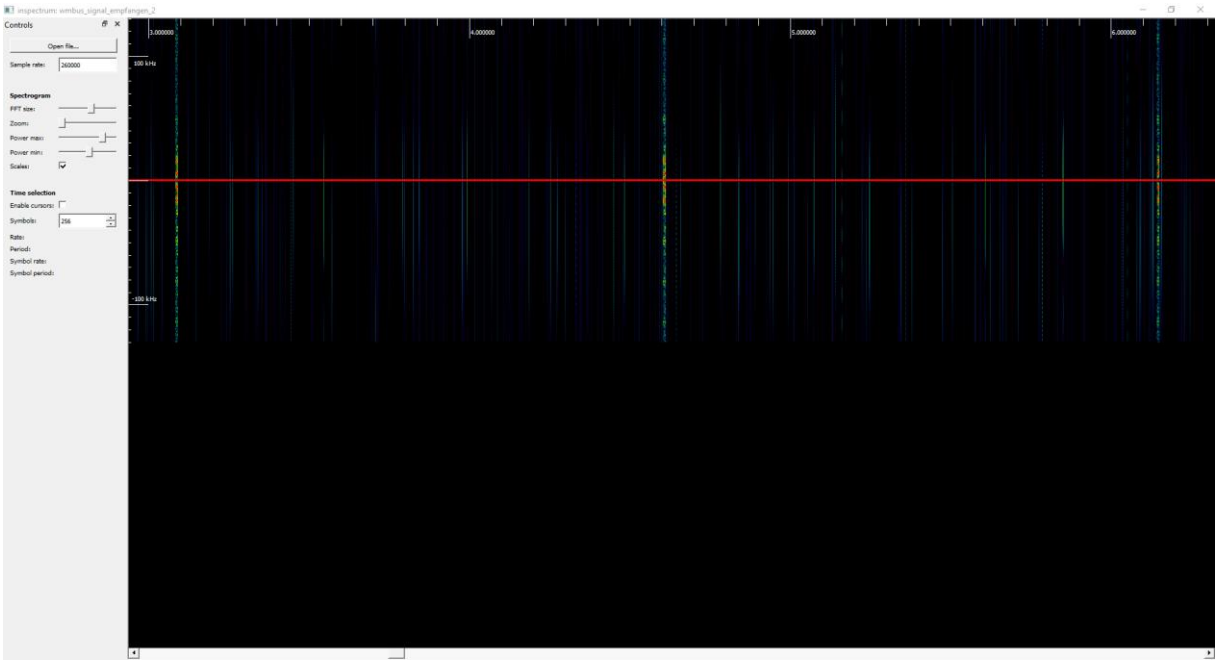


Abbildung 15: Beispielbild für Inspectrum mit Aufzeichnung in Vollansicht
(Quelle: eigene Signale)

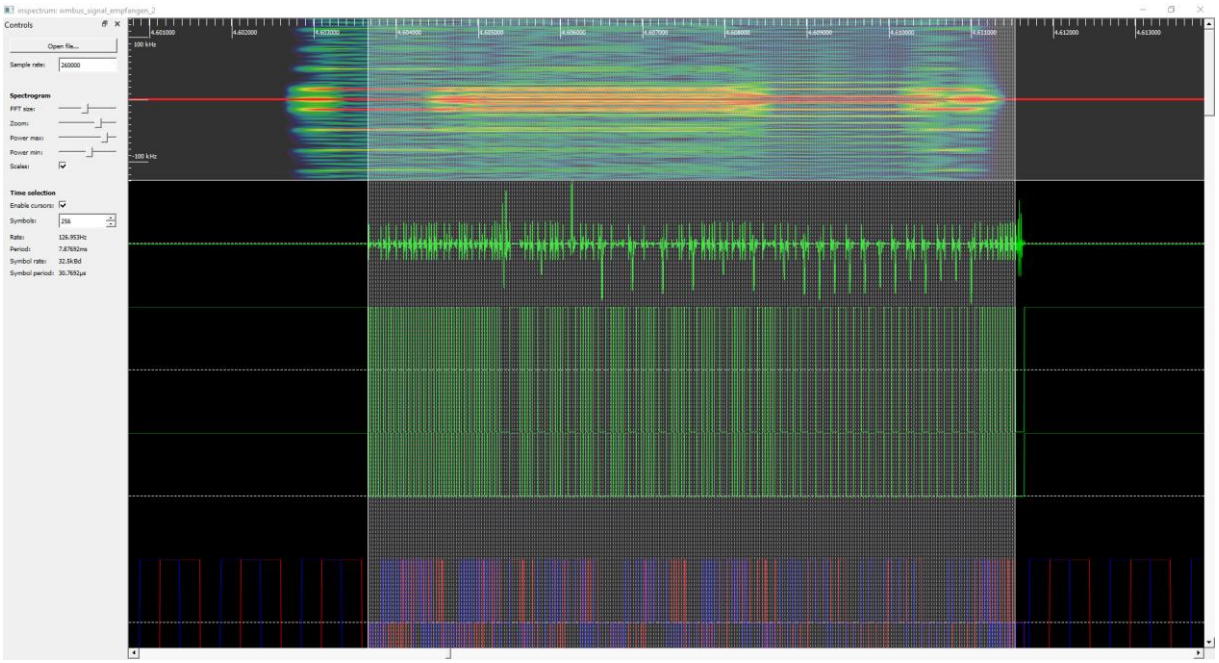


Abbildung 16: Beispielbild für Inspectrum mit Aufzeichnung in vergrößerter Ansicht
(Quelle: Eigene Darstellung)

1.7.4 Audacity

Zu Beginn dieser Arbeit wurde versucht, ein aufgefangenes und demoduliertes Signal mit dem mir im Vorfeld bereits geläufigen Programm Audacity auszuwerten. Ich persönlich empfehle dieses Tool eher für eine Auswertung von Signalen, die eine Varianz in der Amplitude mit sich bringen. Ein gewisses, in der Aufnahme enthaltenes, Rauschen lässt sich hier nachträglich schwierig bis gar nicht herausfiltern. Auch eine konkretere Analyse eines Signalabschnitts lässt sich nur recht mühselig realisieren. Dennoch kann sich wirkungsvoll ein Informationsgehalt Abbilden lassen.

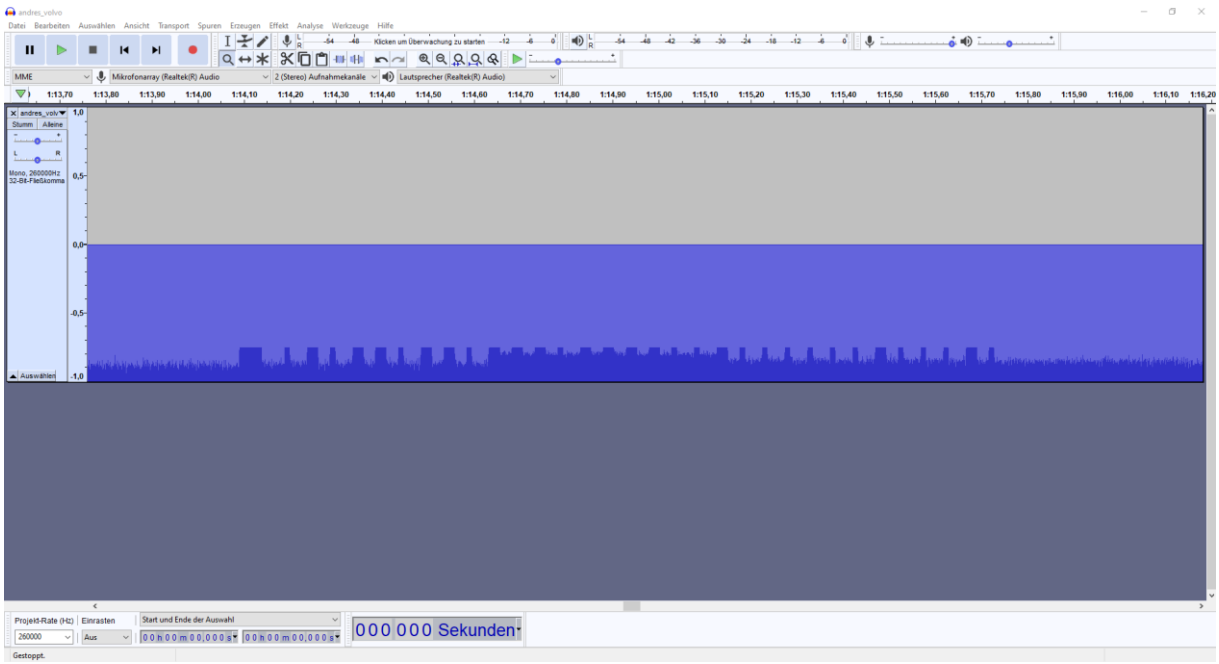


Abbildung 17: Beispielbild für Audacity
(Quelle: Eigene Darstellung)

1.7.5 STSW-S2LP-DK

Dieses evaluierte Softwarepaket der Firma STMicroelectronics wird im Zusammenhang mit dem bereits unter Punkt 1.6.3 erwähnten Gerät, welches ebenfalls von der Firma STMicroelectronics stammt, verwendet. Mit diesem Programm ist es möglich, speziell mit diesem Board Funksignale unter 1GHz zu senden oder zu empfangen. Genauer gesagt kann es dabei in den lizenzfreien ISM- und SRD-Frequenzbändern operieren, die bei 433MHz, 868MHz und 920MHz liegen oder in breiteren Bereichen (413-479MHz, 452-572MHz, 826-958MHz, 904-1055MHz). [8]

Dabei ist es möglich Optionen wie Trägerfrequenz, Datenrate, Freq.-Deviation und Bandbreite einzustellen. Ebenfalls besteht eine Auswahl an diversen Modulationsarten (z.B. 2FSK, 4GFSK, ASK, OOK). Die zu senden wollenden Informationen sind frei konfigurierbar. Die

Paketinformationsoptionen sind dabei preamble (max. 32 Bit), sync word (max. 32 Bit), Länge der Nachricht (max. 8 Bit), Payload und CRC-Kodierung (max. 8 Bit) unter der Paketformateinstellung „Basic“ im Reiter „Packet Settings“. Neben der Option Basic lässt sich das Übertragungsprotokoll „WMBUS“ auswählen. In diesem können diverse Submodi (z.B. T1, T2 meter to other) gesetzt, sowie eine optionale Prä- oder Postambel mit der einer jeweiligen festgeschriebenen Sequenz von „01“*n ausgewählt werden. Unter dem Reiter „Transmission Test“ → „TX“ lässt sich die eigentlich zu senden wollende Nachricht festlegen, sowie die Anzahl der zu sendenden Paketwiederholungen und die Intervallzeit. Unter „RX“ lassen sich empfangende Pakete anzeigen. Zusätzlich bietet dieses Programm diverse weitere Sende- und Empfangsoptionen, sowie Test-Signale ohne spezielle Codierung.

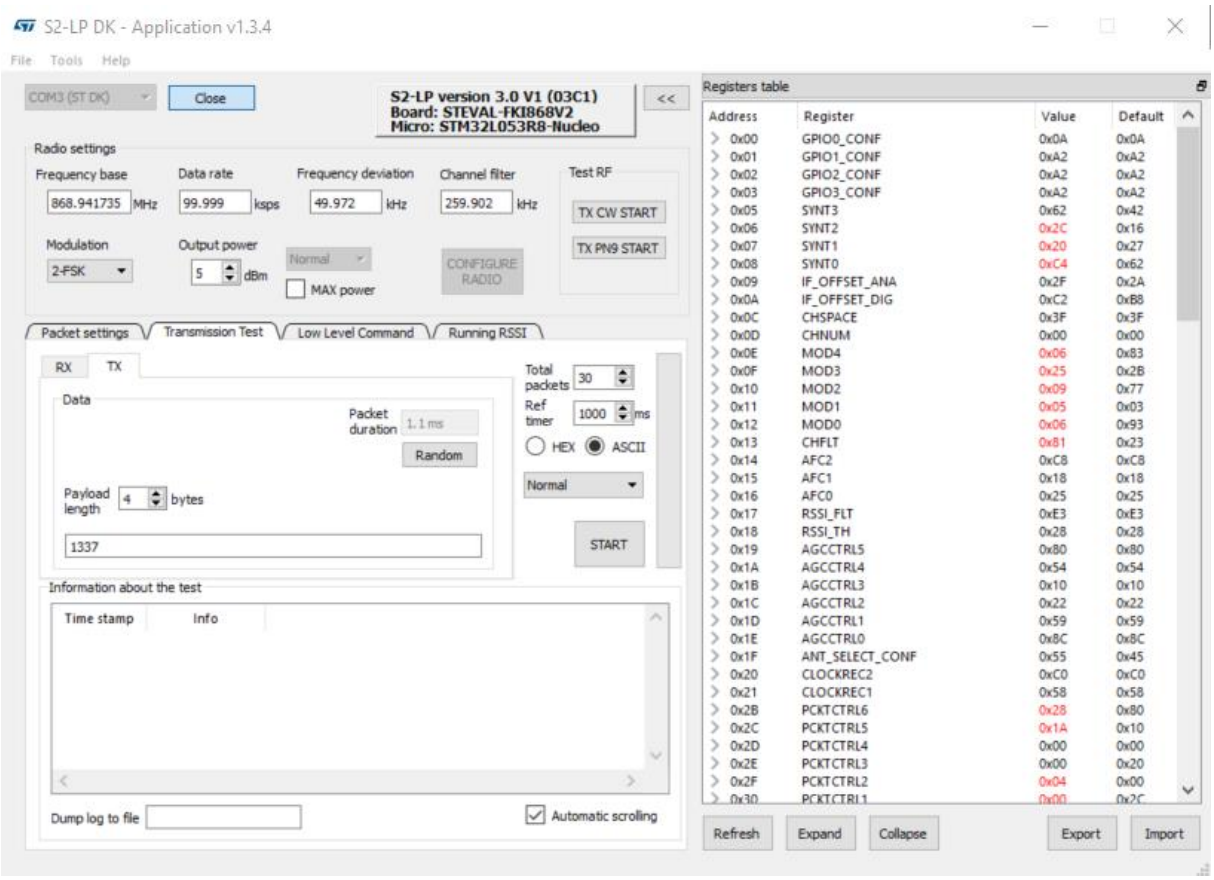


Abbildung 18: Beispielbild für S2-LP DK
(Quelle: Eigene Darstellung)

2 GNU Radio

2.1 Was ist GNU Radio?

Das ursprünglich für Linux entwickelte Tool GNU Radio ist ein freies open source-Programm für die Entwicklung von SDR-Schaltungen. Die Benutzeroberfläche stellt ein GUI (Graphical User Interface) dar und man „programmiert“ Code mit Hilfe von Informationsblöcken. Die Blöcke stellen bestimmte Funktionseinheiten dar und werden mit Linien verbunden. Im Hintergrund wird anschließend ein Gesamtkonzept/ ein einheitlicher Code aus dieser „Schaltung“ erzeugt. Es ist dennoch möglich über eine textbasierte, und damit für die meisten Anwender herkömmliche, Methode, Code zu erzeugen. Wichtig wird dieser Fakt bei der Erstellung und Implementierung eigens kreierter Funktionseinheiten/-blöcke.

Da dieses Werkzeug einen sehr weitreichenden Informations- bzw. Gestaltungsumfang besitzt, ist es ideal für die Entwicklung von SDR-Anwendungen geeignet.

Die generelle Intension von Software Defined Radio, in Bezug auf den gleichen Gewinn, im Gegensatz zur analogen Technik für die Entwicklungsumgebung, lässt sich hiermit leicht umsetzen. Ein dafür sehr breit gefächertes und bereits vorinstalliertes Portfolio an diversen Operatoren, Quellen, Senken, Filtern, Modulatoren, Demodulatoren und findige variable Gestaltungsmöglichkeiten der Funktionen setzen der eigenen Kreativität kaum Grenzen. Zudem lassen sich, wie bereits erwähnt, selbst Blöcke erstellen, falls das zur Verfügung gestellte nicht ausreicht oder man diverse, oft zusammen verwendete, Einheiten substituieren möchte.

GNU Radio in Verbindung mit einer SDR geeigneten Hardware zu verwenden ist kein Muss. Dieses Programm lässt sich auch „offline“ als reines Simulations-/Analyseprogramm betreiben. Anwendung findet diese Variante vor allen darin theoretische Signalgeneratoren/-modulatoren zu entwerfen oder eine bereits aufgenommene Probe zu bearbeiten.

Eine groß aufgestellte Community und viele Dokumentationsbeiträge bieten Einsteigern und Fortgeschrittenen die Möglichkeit des einfachgemachten und leicht zugänglichen Selbststudiums. Eine mögliche Informationsquelle, die sich auch für Themenneulinge eignet, ist die offizielle Enzyklopädie-Seite von GNU Radio – das GNU Radio Wiki (https://wiki.gnuradio.org/index.php/Main_Page).

2.2 Implementierung unter Windows

Das ursprünglich für den Linux-Kernel entwickelte Programm GNU Radio lässt sich durch zahlreiche Umsetzungen recht unkompliziert unter Windows installieren. Für die Realisierung gibt es verschiedene Wege, die je nach Anspruch an die SDR-Projekte und den eigenen Fortschritt geeignet sind.

Eine Möglichkeit bietet das Radioconda-Projekt, bei dem es sich um einen plattformübergreifenden Paketmanager handelt. Neben der eigentlichen GNU Radio-Entwicklungssoftware erhält man hiermit diverse weitere Softwareanwendungen. Eine Installationsanleitung lässt sich zum Beispiel direkt auf der Wiki-Seite von GNU Radio finden oder in einem auf GitHub bereitgestellten Projekt. Conda ist eine unabhängige Umgebung mit eigenen Paketen die unabhängig von der Systeminstallation und von dem Paketmanager sind.

Eine weitere Option ist die direkte Installation des GNU Radio Companion unter Windows von der offiziellen GNU Radio-Seite. Somit erhält man lediglich die reine GUI-Programmierumgebung für seine SDR-Projekte.

Die in dieser Arbeit verwendete Methode ist die Installation des sogenannten PhotosSDR-Pakets. Dieses beinhaltet alle wichtigen Softwarekomponenten für Windows, auch im Bezug auf die hier verwendete Hardware (LimeSDR). So kommt mit diesem Paket nicht nur der GNU Radio Companion, sondern auch die für Verwendung eines LimeSDR-Geräts erforderliche Einrichtungssoftware „LimeSuite“, dass für die Treiberinstallation wichtige Tool „Zadig“, die SoapySDR-Bibliothek, um diverse SDR-Geräte zu initialisieren/ konfigurieren, sowie zusätzliche und sehr nützliche Empfangs- oder auch Analysesoftware (CubicSDR, GQRX, Inspectrum). Generell bietet das Photos-Projekt eine weitere Vielzahl an Möglichkeiten für das Erstellen von Topologien unter grafischen Elementen. Diese können über ein Netzwerk bereitgestellt und getestet werden. Finden lässt sich dieses kostenlose open-source-Paket und eine detailliertere Beschreibung des Gesamtkonzepts von PhotosSDR unter anderen auf deren offiziellen Website (https://www.pothosware.com/#about).

2.3 Für diese Arbeit interessante Blöcke/Funktionen unter GNU Radio

2.3.1 Frequency Xlating FIR Filter

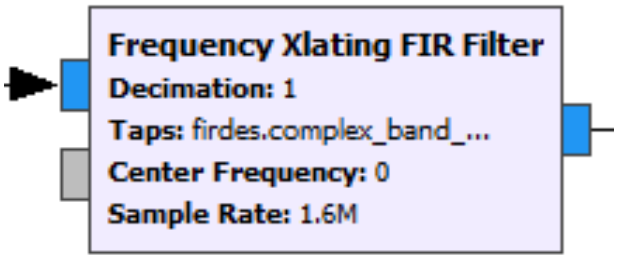


Abbildung 19: Frequency Translating FIR Filter als Block im GRC-Flowgraph
(Quelle: Eigene Darstellung)

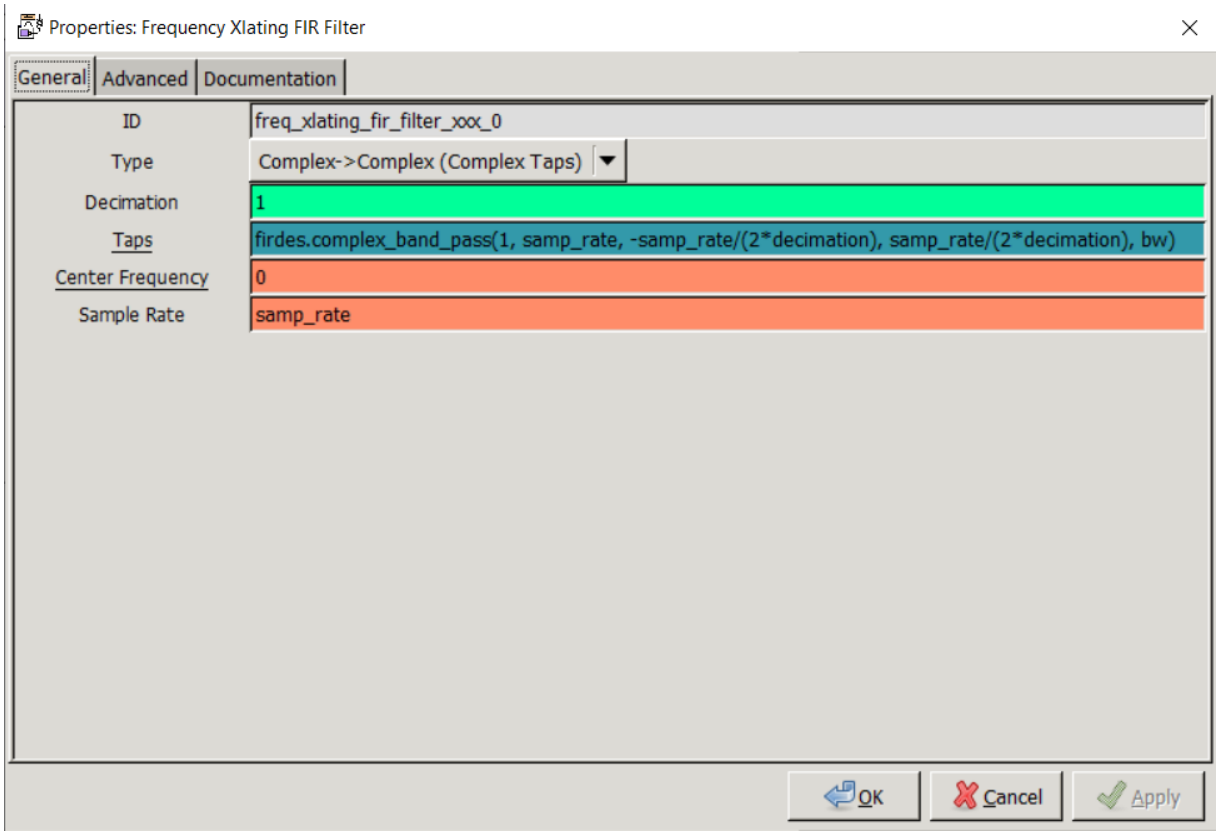


Abbildung 20: Frequency Translating FIR Filter in der Konfigurationsansicht
(Quelle: Eigene Darstellung)

Ein wichtiger Bestandteil eines, auch im analogen Sinne, Funkschaltungsaufbaus ist der einzusetzende Frequenzfilter. Allgemein lässt sich mit einem analogen oder auch digitalen Frequenzfilter das gesamte, dem Gerät möglichen, Frequenzspektrum begrenzen, indem Kriterien für den weiteren Durchlass getroffen werden. So bestehen die Möglichkeiten Frequenzen oberhalb (Hochpassfilter), unterhalb (Tiefpassfilter) oder zwischen Werten (Bandpassfilter) durchkommen zu lassen. Weitere Verarbeitungsschritte greifen demnach nur noch auf den letztendlich für den Prozess relevanten Bereich zu.

Für den Versuchsaufbau in dieser Arbeit, beim Empfangen von wM-Bus-Signalen, wird ein von GNU Radio bereitgestellter „Frequency Translating FIR Filter“ verwendet. Dieser besitzt die Eigenschaft ein empfangendes Signal auf eine einzustellende Frequenzmitte zu verschieben. Zudem wirkt intern ein Anti-Aliasing-Filter, um auftretende Spiegelfrequenzen zu unterdrücken/ auszublenden. Des Weiteren bietet dieser Block eine Decimation-Option. Mit dieser Möglichkeit lässt sich die im Vorfeld für das Signal eingestellte Sampling-Rate verringern – und zwar um den Faktor der Dezimierung, welcher einen resultierenden Teil der ursprünglichen Abtastrate bedingt. Diese kann somit nachfolgend verkleinert werden, da nur noch der Frequenzbereich von Interesse (die Bandbreite des Signals) abgetastet werden muss. [9]

2.3.2 Simple Squelch

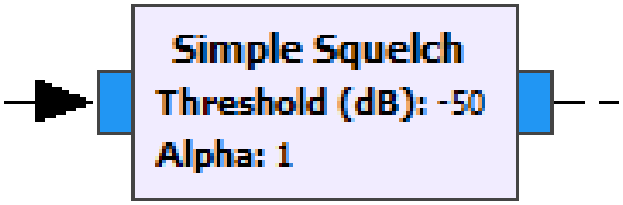


Abbildung 21: Simple Squelch als Block im GRC-Flowgraph
(Quelle: Eigene Darstellung)

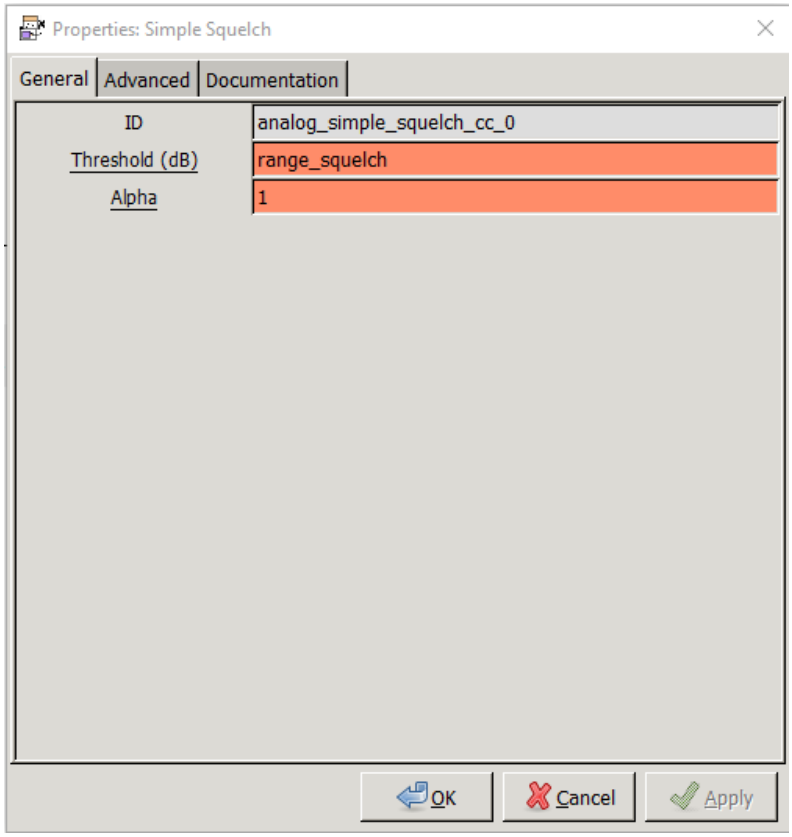


Abbildung 22: Simple Squelch in der Konfigurationsansicht
(Quelle: Eigene Darstellung)

Der Simple Squelch-Block dient ebenfalls zur Filterung. Nur filtert dieser nicht in der Frequenzachse, sondern in der der Frequenzachse orthogonal zugeordneten Leistungsstärkenachse. Das somit im Vorfeld, durch einen bereits eingesetzten Filter, verkleinerte Spektrum (und zwar nur auf den relevanten und für weitere Vorhaben interessanten Teil), wird durch die Threshold-Option ein weiteres Mal sondiert. Es werden nur Signale ab einer bestimmten Signalstärke durchgelassen und ein unerwünschtes Signalrauschen damit zusätzlich unterdrückt. Neben dem Threshold-Faktor lässt sich ein Verstärkungsfaktor für die durchgelassenen Frequenzen, unter „Alpha“, einstellen. [10]

Die Abbildungen 21 und 22 zeigen einmal die GUI-Ansicht für weitere Verknüpfungen und einmal die Detailansicht zum Einstellen der gegebenen Optionen. Dabei handelt es sich um ein und denselben Block. Für den Parameter Threshold wurde anstatt der einzustellenden Zahl (hier -50dB) eine Variable verwendet, mittels Range-Block, um beim Ausführen der Gesamtschaltung ein dennoch flexibles/variables System zu haben. Unterstrichene Parameter weisen in GNU Radio daraufhin, dass diese während des „Betriebs“ verändert werden können. Dies stellt sich, je nach Anwendungsfall, Kriterien für Untersuchungen, etc., als sehr hilfreich dar und spiegelt meiner Meinung nach einmal mehr den Kerngedanken von Software Defined Radio wider.

2.3.3 Quadrature Demod

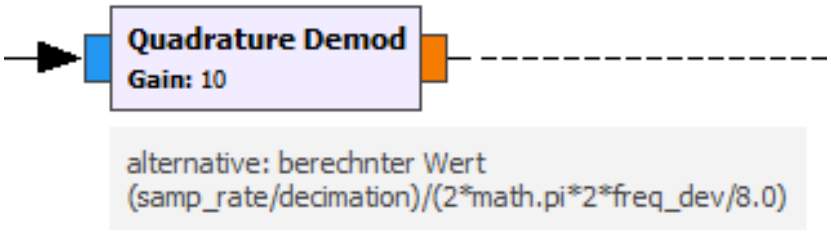


Abbildung 23: Quadrature Demod als Block im GRC-Flowgraph
(Quelle: Eigene Darstellung)

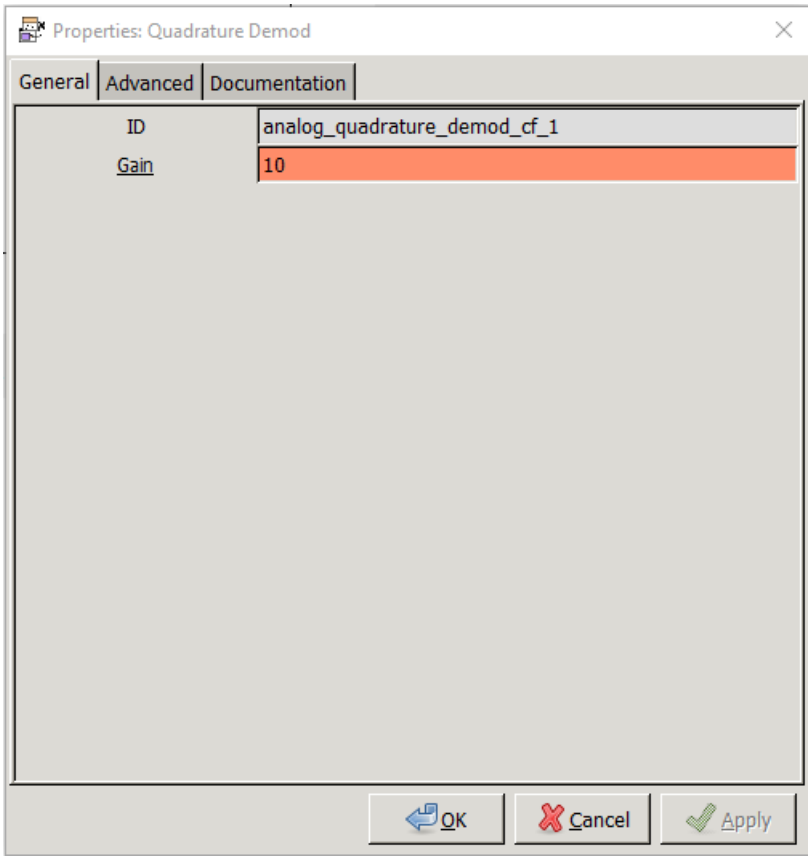


Abbildung 24: Quadrature Demod in der Konfigurationsansicht
(Quelle: Eigene Darstellung)

Der Quadrature-Demod-Block ist der vom GNU Wiki empfohlene Block zur Demodulation diverser frequenzabhängiger Modulationsverfahren (z.B. FM, FSK, GMSK). Zur Verstärkung des demodulierten Signals wird ebenfalls eine Berechnungsformel vorgeschlagen, welche hier als Kommentar unter dem GUI-Block zu sehen ist. Nach mehreren Praxisversuchen hat sich im Versuchsaufbau allerdings ein Verstärkungsfaktor von 10 als zielführend herausgestellt.

Dieser Block wandelt das empfangene komplexe Signal, bestehend aus 2 Komponenten, in eine komplexe Zahl um. Dazu wird hier das Produkt zwischen einen um eine Abtastung verzögerten und konjugierten Abtastwert und einen nicht verzögerten Abtastwert gebildet und

anschließend das Argument daraus ermittelt. Das resultierende Ausgangssignal des Quadrature-Demod-Blocks entspricht einer Pulsamplitudenmodulation (PAM), welche Abhängig vom Input (Frequenzwert größer oder kleiner Null) ihre Form entsprechend zeigt und einen Wert zwischen -1 und 1 annimmt. Der demodulierte Datenstrom lässt sich mit weiteren Verarbeitungsschritten in einen Bitstrom umwandeln. [11]

2.3.4 Clock Recovery MM

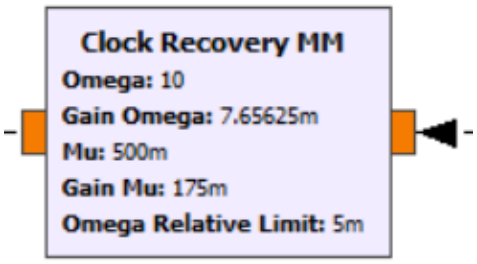


Abbildung 25: Clock Recovery MM als Block im GRC-Flowgraph
(Quelle: Eigene Darstellung)

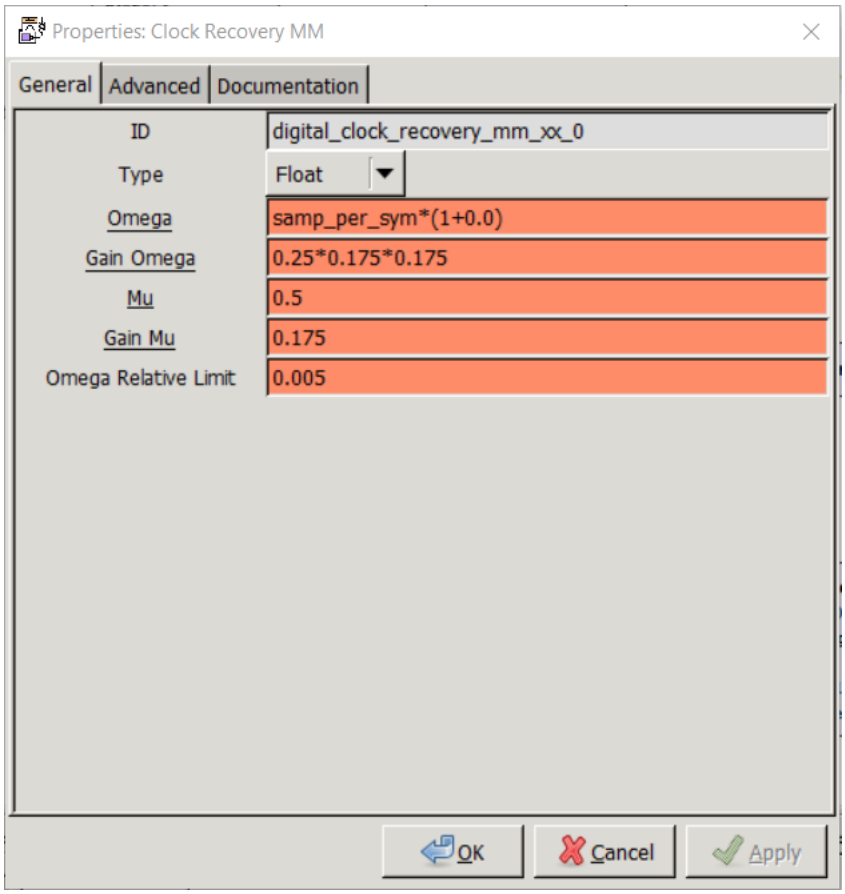


Abbildung 26: Clock Recovery MM in der Konfigurationsansicht
(Quelle: Eigene Darstellung)

Um das empfangende Signal in eine Form der für weitere Verarbeitungsprogramme oder -schritte zu bringen, ist es notwendig zu interpretieren wann eine Teilinformation, also ein Bit, anfängt und wann endet. In GNU Radio Companion (GRC) gibt es den bereits implementierten und fast vollständig parametrierten Block „Clock Recovery MM“. Es reicht demnach für die meisten Anwendungen aus hierbei lediglich die Variable „samp_per_sym“ unter Omega anzupassen. samp_per_sym steht für Sample pro Symbol, also Abtastwerte für jedes gewonnene Informationsteilchen. Diese Variable bildet dabei das Verhältnis von der Abtastrate zur Datenrate.

Die Detektion der High- und Low-Pegel eines gewonnenen sauberen Signals mit einer definierten zeitlichen Schrittweite findet also hier statt. Die in der Gleichung benötigte Abtastrate ist bekannt, denn diese ist im Vorfeld einzustellen. Eine Information über die Datenrate des Signals lieg in den meisten Fällen nicht vor. Um diese zu ermitteln, empfiehlt sich das im Vorfeld bereits erwähnte Tool Inspectrum. [12]

2.3.5 Binary Slicer

Abbildung 27: Binary Slicer als Block im GRC-Flowgraph
(Quelle: Eigene Darstellung)

Der Binary Slicer markiert die als solche erkannten High- und Low-Pegel als 1 und 0. Dazu steht am Eingang dieses Blocks ein Datenstrom aus Float-Werten an. Positive Werte werden zu einer 1 markiert/geschrieben, negative Float-Werte zu einer 0. Es entsteht am Ausgang ein Byte-Stream, welcher das, in den Block eingespeiste, demodulierte Signal repräsentiert. [13]

2.3.6 File Sink

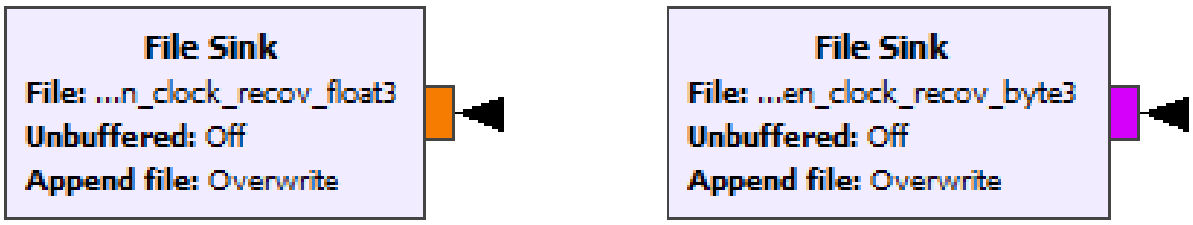


Abbildung 28: zwei File Sinks als Blöcke im GRC-Flowgraph
(Quelle: Eigene Darstellung)

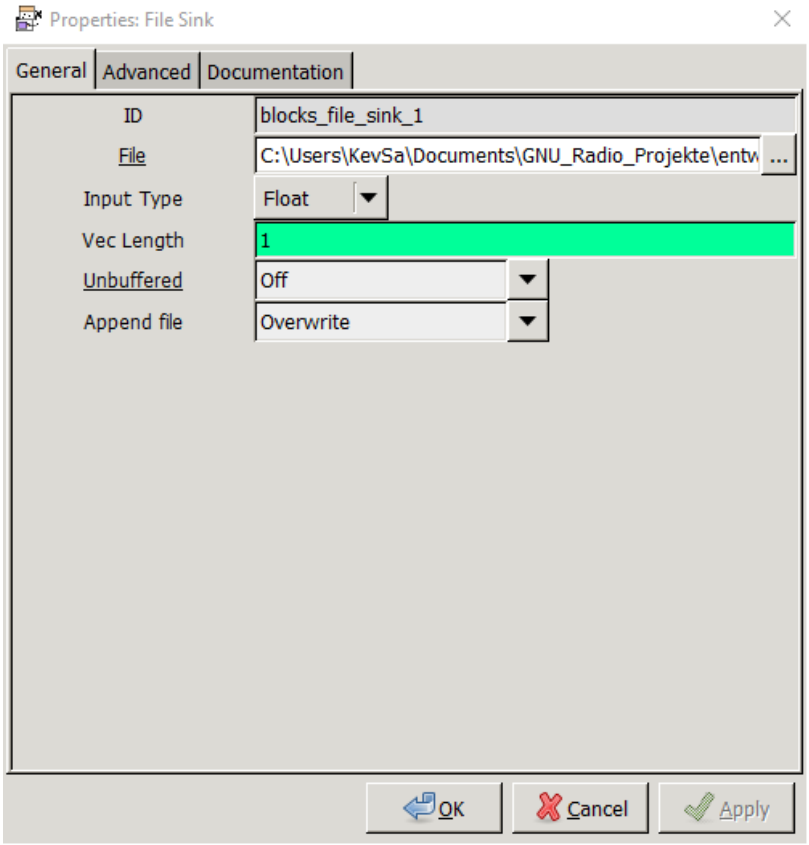


Abbildung 29: File Sink in der Konfigurationsansicht
(Quelle: Eigene Darstellung)

Um die aus einer Schaltung gewonnenen Daten aufzufangen und abzuspeichern, gibt es in GNU Radio „File Sinks“. In diesen Blöcken kann auf den eingehenden Datentyp, entsprechend der vorhergehenden Anwendung, reagiert werden (Input Type). Zulässige eingehende Datenformate sind Werte des Typs: Complex, Float, Int, Short oder Byte. Unabhängig vom anliegenden Datenstrom, werden die Werte in eine Binärdatei umgewandelt und können somit mit jeder Programmumgebung gelesen werden, die Binärdateien lesen kann. Das Binärformat entspricht einem 32-Bit-Float. Diese werden nacheinander angelegt. Ein besonderer Datentyp ist „Complex“, da diese Werte 2 Anteile beinhalten. Um dies zu händeln, schreibt die File Sink erst ein Float32 mit dem Realteil voll und fügt anschließend den zugehörigen Imaginärpart,

ebenfalls als Float32, an. Zurückgelesen wird das Ganze in gleicher Reihenfolge – zuerst Realteil, dann Imaginärteil. Generell ist es möglich und durchaus üblich ein bereits aufgenommenes und abgespeichertes Signal wiedereinzulesen. Dazu kann die der File Sink äquivalent zugehörige File Source verwendet werden. Diese Methode bietet sich vor allen an, wenn ein bestimmtes Signal nicht dauerhaft zur Verfügung steht oder es unerwünscht variiert. Beispielweise wurden in diesem Versuchsaufbau Signale parallel und mit entsprechenden Zwischenschritten einmal in einer File Sink des Typs Float (Ansicht der Signale mit z.B. Inspectrum) und in einer File Sink des Typs Byte (Ansicht in z.B. HxD (Binärdatei auslesen)) abgespeichert bzw. aufgefangen.

Weitere Einstellmöglichkeiten in diesem Block sind die, wie bei jedem Block, manuell zu vergebende ID (der Name des Blocks im aktuellen Programm), sowie der Dateipfad wo die letztendliche und zu benennende File-Sink-Datei gespeichert wird. Zudem die Vectorlänge des abzuspeichernden Streams und ob ein Buffering zugeschaltet werden soll oder nicht, welches die Performance des Flussdiagramms verbessern kann. Zu guter Letzt, ob beim mehrmaligen Beschreiben dieser Datei der neu aufgenommene Inhalt den alten ersetzt oder der neue Inhalt dem alten angefügt wird. [14]

2.3.7 GUI Hint

Es besteht die Möglichkeit bei allen Anzeige- und Steuerelementen ein „GUI Hint“ zu vergeben. Mit diesen werden die Anzeigen und/oder Regler selbstdefiniert angeordnet, um beispielsweise eine Zugehörigkeit von diversen Blöcken zueinander zu verdeutlichen oder eine Gegenüberstellung zu veranschaulichen. Dafür ist die Anzeigeseite in einer zweidimensionalen Matrix gegliedert, dessen Größe sich nach der am höchsten vergebenen Zahl (Zeile und Spalte separat) in einem GUI Hint richtet. In der Schreibform „x,y“ oder „x,y,a,b“ werden Anzeigen und Regler eingerichtet. Die erste Variante beschreibt dabei den Platz als „Zeile, Spalte“ und die zweite als „Zeile, Spalte, Zeilengröße, Spaltengröße“. [15]

Beispiel:

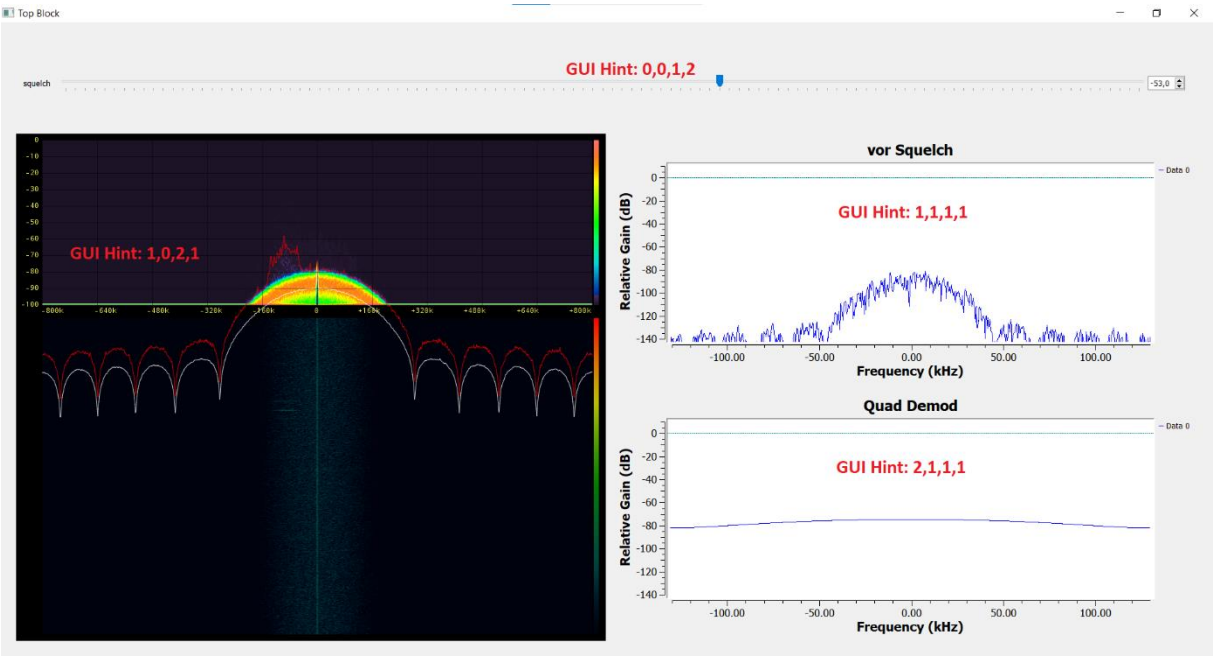


Abbildung 30: Beispiel der Anordnung diverser Anzeige- und Steuerelemente mit spezifischer Beschreibung des GUI-Hints

(Quelle: Eigene Darstellung)

2.3.8 Range-Variable

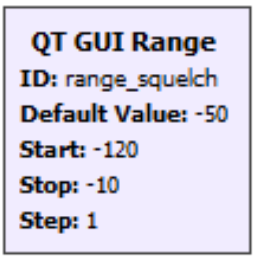


Abbildung 31: Variablentyp/ Steuereinheit Range als Block im GRC-Flowgraph

(Quelle: Eigene Darstellung)

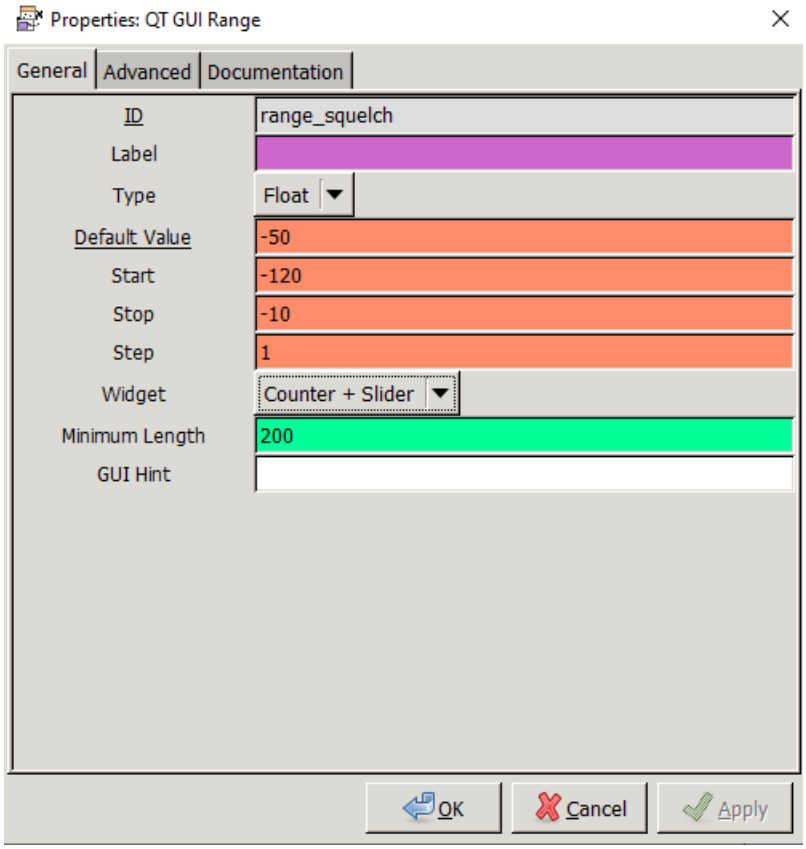


Abbildung 32: Range-Variable in der Konfigurationsansicht
(Quelle: Eigene Darstellung)

Die Möglichkeit während des laufenden Prozesses bestimmte Parameter gezielt zu manipulieren, bietet vor allen in der Entwicklungsphase einen enormen Komfortgewinn. Mit dem Range-Block ist diese Option gegeben. Er stellt eine Variable dar, die über dessen ID in andere Blöcke eingefügt und benutzt werden kann. Unter dem Parameter „Default Value“ wird der Wert eingetragen, den der ausgegebene Anfangswert des Blocks, also direkt nach dem Ausführen des Flowgraphs, betragen soll. Unter „Start“ und „Stop“ werden die obere und untere Grenze der Variable verzeichnet. Bei „Step“ trägt man das Inkrement, das pro Einmalbetätigung ausgeführt wird, ein. Da es sich beim Range-Block um ein optisches Element handelt und es beim Ausführen des Flowgraphs angezeigt wird, gibt es hier verschiedene Darstellungsvarianten unter „Widget“. Die Datentypen einer Range-Variable können Float oder Int sein. [16]

2.3.9 Anzeige- und Steuerelemente

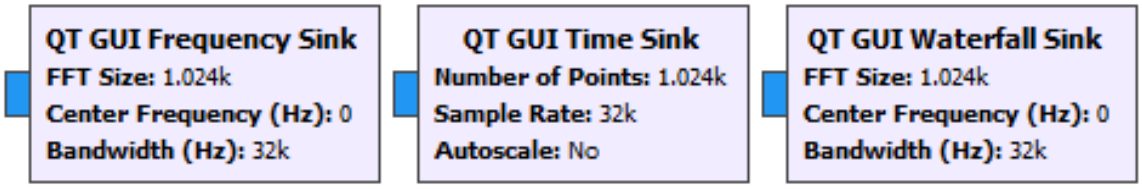


Abbildung 33: Collage diverser Anzeigeeinheiten als Blöcke im GRC-Flowgrah
(Quelle: Eigene Darstellung)

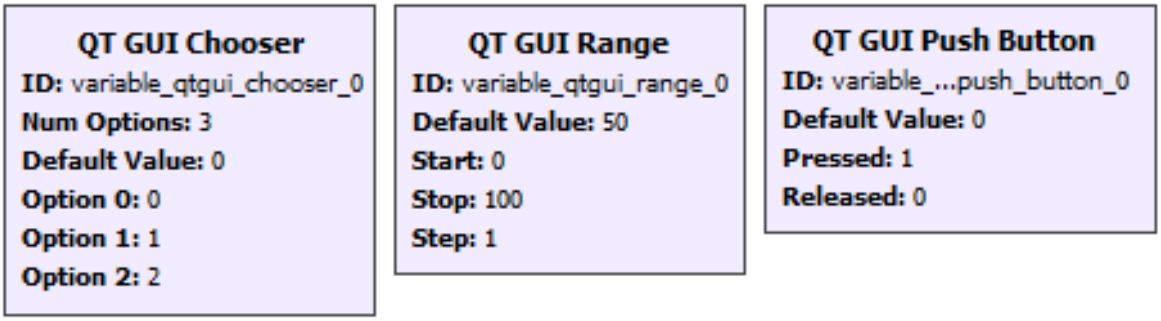


Abbildung 34: Collage diverser Steuereinheiten als Blöcke im GRC-Flowgraph
(Quelle: Eigene Darstellung)

Der GRC bietet an, momentane Signalzustände, beim Ausführen einer Schaltung, optisch darzustellen. Die in der oberen Abbildung (33) gezeigten „Sinks“, neben vielen weiteren, realisieren diese Option. Je nach Anwendungsfall können die Sinks entsprechend gewählt und parallel zum Verarbeitungsprozess oder als letztes Glied der Kette zum Anzeigen des Resultats verwendet werden. Jedes Element bietet dabei eigene Parametrieroptionen, welche im Vorfeld eingegeben werden können. Weitere Optionen lassen sich während des Ausführens der Schaltung im Anzeigefenster, per Klick der mittleren Maustaste in das Diagramm, auswählen.

Neben den darstellungsbietenden Analysewerkzeugen gibt es im Anzeigefenster Platz für Reglerbausteine (Widgets). Diese können je nach Ausführung bestimmte Aufgaben realisieren. Beispielsweise ist es mit einem „Chooser“ möglich eine Auswahl zwischen verschiedenen und separat gekennzeichneten Werten zu treffen.

Alle Elemente sind, wie im Abschnitt 2.3.7 beschrieben, manuell anordbar. GRC erstellt allerdings selbst eine automatisch generierte Anordnung der Elemente. Eine manuelle Einrichtung ist somit nicht zwingend erforderlich.

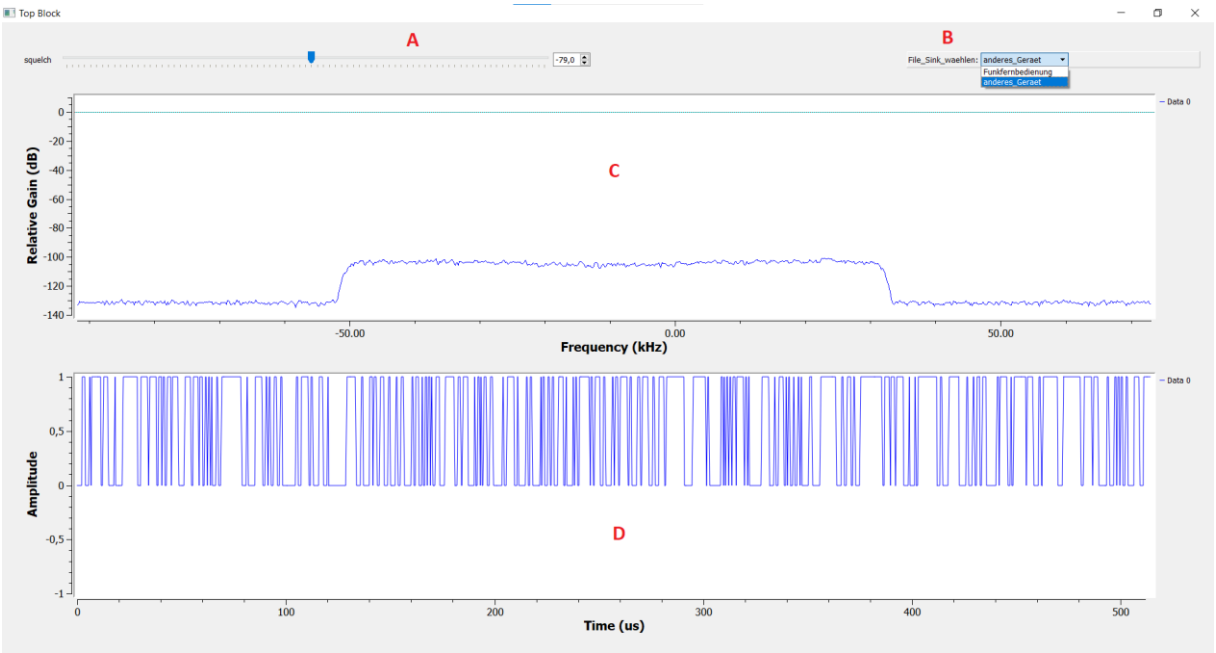


Abbildung 35: Beispiel für Anzeige- und Steuerelemente im GRC-Flowgraph
(Quelle: Eigene Darstellung)

Abbildung 35 soll als Beispiel für die Anordnung und Verwendung diverser Sinks und Widgets dienen. Dabei ist der mit A markierte Bereich ein Range-Block mit der Option „Counter + Slider“, Bereich B ein Chooser mit 2 auswählbaren Einheiten, Segment C ist seine Frequency Sink zur Anzeige des Signals im Frequenzbereich und der untere Teil (D) ist eine Time Sink und dient zur Anzeige des Signals im Zeitbereich. Alle Elemente im Beispielbild wurden manuell angeordnet über die Eingabe der Position und der Skalierung im jeweiligen GUI-Hint-Feld (A: 0,0,1,2; B: 0,3,1,1; C: 1,0,1,4; D: 2,0,1,4).

3 wM-Bus

3.1 Was ist wM-Bus?

Die Abkürzung „wM-Bus“ steht für „wireless Meter-Bus“ und bezeichnet die drahtlose Übertragung des Meter-Bus-Protokolls. Dieser Standard wurde im Zuge der Energiewende entwickelt und wird aktuell eingesetzt für Smart-Meter-Geräte. Bauelemente dieser Art können sowohl elektronische Zähler als auch die verwaltenden Smart-Meter-Gateways sein. Besonderes Augenmerk des Standards ist zum einen eine energieschonende Betriebsart, welche auch für den Batteriebetrieb geeignet ist. Je nach eingestellter Übertragungsart werden die gesammelten Daten nur sehr kurz gesendet und/oder empfangen. Auch die Zykluszeit zwischen den Kommunikationen kann sehr ausgeweitet sein. Ein anderer markanter Aspekt des Gesamtkonzept ist das stetige Übermitteln energierelevanter Messdaten in Echtzeit, welche für den Energieanbieter, aber auch für den Verbraucher von Interesse sind.

Prinzipiell herrscht in einem Smart-Meter-Netz (auch als Local Metrological Network (LMN) bezeichnet), in dem wM-Bus verwendet wird, das Master-Slave-Prinzip. Dabei stellt das SMGW (Smart Meter Gateway) den Master dar und die in verbindungstehenden Zähler/Sensoren die Slaves. Über dieses Gateway werden die Daten gesammelt und an den Stromanbieter weitergleitet. Für die Übertragung von Gateway zum Anbieter werden andere Übertragungsverfahren, wie z.B. Powerline oder Internet, verwendet. Der wireless M-Bus kann zwischen verschiedenen Betriebsmodi variieren, die sich in der Datenrate, Codierung, Frequenz, Übertragungsart, Frequenzhub, Präambelaufbau, Chiprate, u.v.m. unterscheiden.

3.2 Normen und Spezifikationen (DIN EN 13757-4)

3.2.1 Allgemein

Die Unterform des in Europa entwickelten M-Bus-Protokolls ist der wireless M-Bus. Entsprechend ist ein Teil der Normen diesem Funkprotokoll zugeordnet und konzentriert sich hierbei auf DIN EN 13757-4 (Kommunikationssysteme für Zähler – Teil 4: Drahtlose M-Bus-Kommunikation). Dort wird konkreter auf die einzelnen Betriebsarten, die Verbindungsschicht und die Verbindung zu höheren Protokollschichten eingegangen.

„Diese Norm legt die Anforderung an die Parameter der Bitübertragungsschicht und der Verbindungsschicht für Systeme fest, die zur Fernablesung von Zählern über Funk dienen. Das Hauptaugenmerk liegt hierbei auf der Verwendung von Funkanlagen mit geringer Reichweite (engl.: Short Range Devices, SRD) in freien Frequenzbändern. Diese Norm umfasst Vorbeigeh- und Vorbeifahrsysteme sowie feste Installationen. Sie kann weitgehend für unterschiedliche Anwendungsschichten angewendet werden.“ [17, p. 4]

3.2.2 Betriebsarten

Eine generelle Betriebsartenunterteilung findet auf Grund der unterschiedlichen Anforderungen an das Netz, dem Zähler oder dem Sammler statt. Die Arten unterscheiden sich dabei meist in nur wenigen Parametern. Die offizielle Kennzeichnung erfolgt durch einen Buchstaben, welcher für die angewendete Betriebsart steht, und einer Zahl, welche angibt, ob es sich bei der Kommunikation zwischen Zähler und Gateway um eine uni- (1) oder bidirektionale (2) Verbindung handelt.

Dabei können die relevanten Geräte nicht ausschließlich nur eine Betriebsart nutzen, sondern, je nach Auslegung des Geräts, mehrere oder gar alle.

3.2.2.1 Betriebsart S (stationärer Betrieb)

Hierbei wird eine uni- oder bidirektionale Verbindung zwischen einem Zähler und einem ortsfesten oder ortsveränderlichen Gerät genutzt. Es werden dabei spezielle und optimierte Gruppierungen für eine unidirektionale Verbindung von ortsfesten batteriebetriebenen Einrichtungen (S1) und für ortsveränderliche Einrichtungen (S1-m) angeführt. [17, p. 6]

3.2.2.2 Betriebsart T (häufiger Sendebetrieb)

Charakteristisch für diese Art ist das häufige Senden von kurzen Informationsinhalten, welche den Zähler im Abstand von wenigen Sekunden verlassen. Eine batteriebetriebene Einheit ist auf Grund des höheren Energieverbrauchs in diesem Modus eher unüblich. Generell ist diese Verbindungsart für das Erfassen der Daten im Vorbeigehen gedacht.

Auch hier wird, wie bei Betriebsart S, eine Unterteilung in mehrere Arten getroffen. So zeichnet T1 aus, dass die Daten vom Zähler lediglich gesendet werden. Dabei schickt dieser mindestens die Zähler-ID und den Messwert raus. T2 stellt dagegen eine bidirektionale Verbindung dar, bei der der Zähler mindestens seine ID und den Messwert liefert, danach aber auf eine Antwort des Empfängers wartet. Erhält der Zähler daraufhin eine Nachricht, wird ein Rückkanal geöffnet, welcher für besondere Dienste genutzt wird. [17, p. 6]

3.2.2.3 Betriebsart R (häufiger Empfangsbetrieb)

Diese Parametrierung dient zum Lauschen von Wecknachrichten und wird sinnvollerweise ausschließlich als bidirektionale Verbindung (R2) betrieben, um bei häufiger Beanspruchung eine dennoch energieschonende Performance zu erzielen. Nach Erhalt dieser Wecknachricht vom zu sendenden Gerät (meistens zu sammelndem Gerät wie Gateway), wird das Empfangsgerät auf einen Datenaustausch, der meist wenige Sekunden dauert, vorbereitet. Es besteht die Möglichkeit mehrere Zähler parallel abzufragen mittels „Mehrkanal-Empfangsbetrieb“. Mit diesem Verfahren bekommt jeder Zähler einen eigenen Frequenzkanal zugeteilt. [17, p. 7]

3.2.2.4 Betriebsart C (Kompaktbetrieb)

Bei dieser Betriebsart werden die wesentlichen Charakteristiken von „Betriebsart T“ wieder aufgegriffen, allerdings mit einer signifikanten Verbesserung des höheren Datendurchsatzes bei gleicher Einschaltdauer. Ansonsten bleibt ein häufiges Senden von kurzen Nachrichten gleich der Betriebsart T. Somit ist diese Parametrierung ebenfalls für das Auslesen von Zählern im Vorbeilaufen/ Vorbeifahren geeignet. [17, p. 7]

3.2.2.5 Betriebsart N (Schmalband-VHF)

Dieser Modus wurde für den Schmalbandbetrieb optimiert. Allerdings operiert dieser nicht auf dem in Deutschland sonst üblichen 868MHz-Frequenzband, sondern bei 169MHz. Betriebsart N ist ebenfalls zum Ablesen der Zähler gedacht. Dabei lassen sich dennoch verschiedene Unterbetriebsarten für reine unidirektionale Verbindung aber auch bidirektionale finden. [17, p. 7]

3.2.2.6 Betriebsart F (Häufiger Empfangs- und Sendebetrieb)

Generell ist dies der Weitbereichskommunikationsmodus und spricht dabei auf einem Frequenzband von 433MHz. Auch hier werden diverse Unterarten üblich gebraucht die sich in der Art der Verbindungsrichtung und den übermittelten Daten unterscheiden. So lauscht ein F2-m-Gerät einer Wecknachricht. Nach Erhalt dieser wird ein Informationskanal für wenige Sekunden geöffnet. Wobei hingegen die Unterbetriebsart F2 ein Telegramm sendet und für kurze Zeit auf eine Antwort wartet, welche eine bidirektionale Kommunikation initiiert. [17, p. 7]

3.3 Markante Merkmale dieses Funkprotokolls

Ein Aspekt der großen Charakteristika des wM-Bus-Protokolls sind zum einen die wesentlichen Unterschiede zwischen den einzelnen Betriebsmodi, welche diverse Differenzen in der Kanalzahl, Codierung oder Datenrate aufweisen. So reichen die genutzten Frequenzen von 868,33MHz bis 868,95MHz und nutzen dabei eine Kanalbreite von 100kHz. Die Datenrate reicht, je nach Parametrierung, von 300bit/s bis 66,66kbit/s. Die in diesem Protokoll verwendeten Codierungsarten sind die entweder die Manchester-Codierung oder die 3-aus-6-Codierung. [18]

Das zu übertragende Protokoll ist nach einem bestimmten Schema aufgebaut und dabei in mindestens 2 Nachrichtenblöcke unterteilt. Diese können sich, laut DIN EN 13757-4, in 2 generellen Formaten (A und B) unterscheiden. Das jeweilige Format muss anhand der Präambel und dem Synchronisationsmuster erkannt werden. Eine wichtige Gemeinsamkeit der beiden Formate ist, dass der erste Block die Verbindungsschicht enthält und eine feste Länge von 10 Bytes hat. Hier werden Informationen über die Länge des Telegramms, die Steuerinformationen und die Adresse des Senders übermittelt.

3.3.1 Erster Block im Telegrammformat A:

L-Feld	C-Feld	M-Feld	A-Feld			CRC
1 Byte	1 Byte	2 Byte	4 Byte	1 Byte	1 Byte	2 Byte

Erläuterung:

3.3.1.1 L-Feld

Das erste Feld des Blocks ist immer, egal ob es sich um Format A oder B handelt, das Längenfeld mit der Größe von 1 Byte. Dieses gibt die Länge des Telegramms an, abzüglich des CRC-Felds. Entsprechend wird hier die Anzahl der Nutzerbytes, einschließlich der Steuer- und Adressbytes, angegeben. Der Unterschied zu Format B besteht darin, dass dieses die Anzahl nachfolgenden Bytes einschließlich des CRC-Feldes festlegt.

3.3.1.2 C-Feld

Dieses Feld ist das Steuerfeld und gibt den Telegrammtypen an. Die Parametrierung dieses Bytes beinhaltet zum Beispiel ob die Nachrichten von einer Primär- oder Sekundärstation gesendet werden. Die dabei weitaus wichtigeren Informationen lassen sich im Unterblock „Funktionscode“ einstellen. Aus der DIN EN 13757-4 lassen sich die einzelnen Funktionscodes herauslesen. Diese beinhalten eine generelle Unterteilung in Primär- und Sekundärstation. Die Codes an sich geben beispielsweise an, ob eine Datenanforderung besteht oder ob eine Zugriffsanforderung von einem Zähler an einem anderen Gerät stattfinden soll.

3.3.1.3 M-Feld

Mit dem M-Feld wird die Hersteller-ID mitgegeben. Dieses Feld ist 2 Byte groß und bildet somit das dritte und vierte Byte des ersten Blocks. Für eine eindeutige Hersteller/-Anwendererkennung gibt es für den ID-Tag genaue Vorgaben die sich unter anderen aus einem 3-Zeichen-ISO/IEC-646-Code (A…Z) zusammensetzen (dieser ist in DIN EN 13757-3 genauer beschrieben). Für die Verwaltung des 3-Zeichen-Codes gibt es in DIN EN 13757-4 eine genauere Beschreibung im Anhang B.

3.3.1.4 A-Feld

Die hier mitgegebene Information ist die Adresse des Senders - es ist somit das Adressfeld. Für dieses sind insgesamt 6 Bytes vorgesehen, wobei hier eine vorgeschriebene Verkettung von „Identifikationsnummer“, „Versionsnummer“ und „Geräte-Identifikation“ bestehen muss. Beispiele dazu lassen sich in DIN EN 13757-4 Anhang C finden.

3.3.1.5 CRC-Feld

Das Akronym „CRC“ steht für „Cyclic Redundancy Check“, oder zu Deutsch „zyklische Redundanzprüfung“. Die Information dieses Feldes lässt sich als Checksumme/ Kontrollwert betrachten und dient somit einer Validität durch unveränderte Werte. Der spezifische Checksummenwert setzt sich dabei aus den vorhergehenden Informationsgehältern der entsprechenden Felder zusammen. Zur Kontrolle dieses Validitätswertes wird ein Abgleich durch das Ergebnis eines CRC-Polynoms vorgenommen. Das im wM-Bus verwendete Polynom lautet:

$$x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^{8} + x^{6} + x^{5} + x^{2} + 1$$

3.3.2 Zweiter Block im Telegrammformat A:

CI-Feld	Datenfeld	CRC-Feld
1 Byte	15 oder, wenn es der letzte Block ist, (((L-9) MOD 16)-1) Byte	2 Byte

Erläuterung:

3.3.2.1 CI-Feld

„CI“ steht für „Control Information“, zu Deutsch „Steuerungsinformation“. Dieses 1 Byte große Feld ist das erste Feld des zweiten Telegrammblocks. Aus den Informationen dieses Feldes wird der Protokolltyp festgelegt und somit die Art der nachfolgenden Informationen. Diese Deklarationseigenschaft beinhaltet zu dem eine Spezifizierung der Schichten (Anwendung-, Transport-, Vermittlungs- oder Verbindungsschicht).

3.3.2.2 Datenfeld

Im diesem Bereich sind die zu übermittelnden Nutzdaten des Geräts enthalten. Dabei werden zusätzlich die Länge und die Codierung der Daten, als wichtiger Parameter, mitübergeben.

[17, p. 33 ff.]

3.4 OMS-Group

Bei der OMS-Group e.V. handelt es sich um einen Interessenverbund verschiedener europäischer Firmen mit dem Ziel einen einheitlichen, herstellerübergreifen Protokollstandart für intelligente Messzähler, für die Erfassung von Strom-, Gas-, Wasser und Wärmeverbräuchen, zu entwickeln und als offene System- und Kommunikationsspezifikation anzubieten. Gegründet wurde der Verein 2007 von 6 Firmen, der Bundesvereinigung der Firmen im Gas- und Wasserfach (FIGAWA), KNX Association, sowie dem Zentralverband Elektrotechnik- und Elektroindustrie. Zum jetzigen Stand umfasst die OMS-Group ca. 70 europäische Firmen. Da die Arbeit dieses Verbundes auf den europäischen Normen und dem Interesse der Firmen basiert, bildet es ein zukunftssicheres System zur einheitlichen Kommunikation. Einige Vorteile diesen Standard zu verwenden sind zum Beispiel: eine freie Lieferantenwahl der Zählerhersteller aufgrund des einheitlichen Kommunikationsstandards, eine direkte Kommunikation mit den dafür ausgelegten Zählern zum Steuern oder Auslesen und die Datenvisualisierung und Anbindung an eine Gebäudeautomation mit KNX. [19] [20]

4 Praktische Durchführung

4.1 Aufgabenstellung

Die dieser Arbeit zugrundeliegende Aufgabenstellung lautet wie folgt:

Unter der Verwendung eines SDR sollen wM-Bus-Daten empfangen werden. Die empfangenen Daten sind zu demodulieren und im optimalen Fall zu dekodieren. Die softwareseitige Schaltungsentwicklung soll in GNU Radio erfolgen. Besonderes Augenmerk liegt dabei auf der Analyse der entwickelten Schaltung, sowie den allgemeinen Umgang mit SDR.

4.2 Ausgangssituation

Es soll sich im Rahmen dieser Arbeit in die Gebiete des „Software Defined Radios“, des Programms GNU Radio und etwaige weitere relevante Software, sowie wM-Bus im Allgemeinen eingearbeitet werden.

Zu diesem Zweck wird von exceeding solutions GmbH ein LimeSDR-USB dieser Arbeit zur Verfügung gestellt. Mit diesem leistungsfähigen SDR-Gerät steht dem kompletten Unterfangen eine solide Grundlage bereit. Für die theoretische Einarbeitung in die Themengebiete, wurden das Internet, diverse Sachbücher und relevante DIN-Normen genutzt. Um einen Kommunikationspartner für die Empfangsversuche gezielt auslösen zu können, wurde von exceeding solutions GmbH ebenfalls ein „wM-Bus-Dummy“ bereitgestellt. Dieser besteht aus einen Nucleo STM32 mit Funkmodulaufsatz. Auf diesem Gerät wurden wM-Bus-artige Nachrichten hinterlegt, welche sich per Tastendruck aussenden lassen. Die für den Empfang erforderlichen Parameter wie Frequenzmitte, Datenrate, Bandbreite, FSK-Abweichung, etc. standen im Vorfeld zur Verfügung. Im Laufe der Bearbeitung zu diesem Projekt wurde gelernt, solche Gegebenheiten selbst zu ermitteln. Es stellte jedoch für die anfängliche Testphase einen komfortablen Gegenpart dar. Des Weiteren wurden, um das Spektrum an Möglichkeiten der zu Untersuchenden Funkgeräte zu erweitern, diverse andere Gerätschaften, wie zum Beispiel eine Funkfernbedienung für Funksteckdosen und reale smarte Zähler hinzugefügt.

4.3 Herangehensweise und Zielsetzung

Zu Beginn der Recherche zu dieser Aufgabenstellung, zeigte sich ein mit kleinem Blick großer Horizont des SDR-Themas. Die weitreichenden Möglichkeiten eine Funkkommunikation mit einem kleinen Gerät und Software abzufangen, zu demodulieren, zu modulieren und/oder weiterzuverarbeiten lässt nur erahnen, wozu dieses Teilgebiet der Elektrotechnik in der Lage ist.

Nach der anfänglichen Einarbeitung in diverse theoretische Grundlagen, wurde das Programm GNU Radio installiert. Um diese Software zu verstehen und deren Potential besser kennenzulernen, wird zunächst mit kleinen Testprogrammen begonnen.

4.3.1 Erste Schaltung (Imaginär- und Realteil)

Um das Thema der wM-Bus-relevanten Modulation „FSK“ mit GNU Radio zu entdecken, besteht die erste Schaltung aus einem kleinen Funktionstest verschiedener Einheiten, um deren Zusammenspiel aber auch die letztendliche Signalanalyse dahinter zu zeigen.

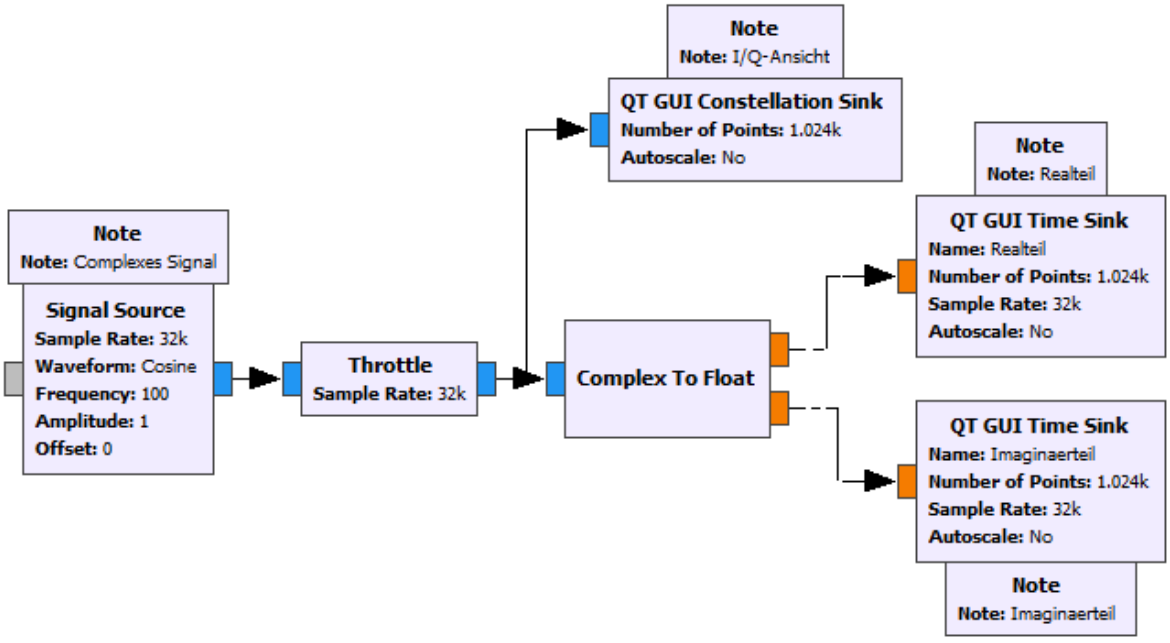


Abbildung 36: GRC-Flowgraph der ersten Schaltung über die Konstellation der einzelnen Signalanteile (Quelle: Eigene Darstellung)

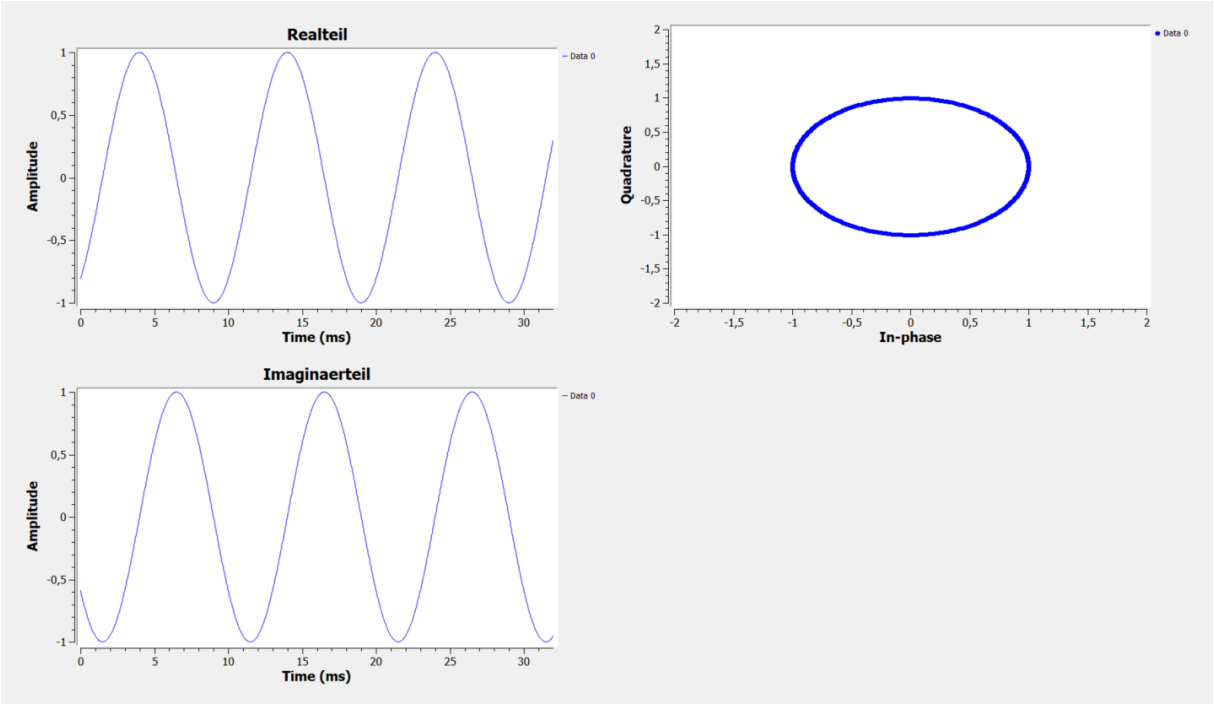


Abbildung 37: Ausführung der unter Abb. 36 gezeigten Schaltung
(Quelle: Eigene Darstellung)

Im Flowgraph dieses Programms, zu erkennen unter Abbildung 36, ist der Block „Signal Source“ zu sehen. Dieser generiert, wie in den angezeigten Parametern zu sehen, ein positives Kosinussignal bei einer konstanten Frequenz von 100Hz und einer Amplitude von 1. Abgetastet wird dieses Signal mit einer Sample-Rate von 32ksps (32000 samples per second). Wie bereits im Theorieteil erwähnt, lässt sich in GNU Radio mit verschiedenen Datentypen arbeiten. Diese werden für den Benutzer freundlicherweise optisch differenziert dargestellt.

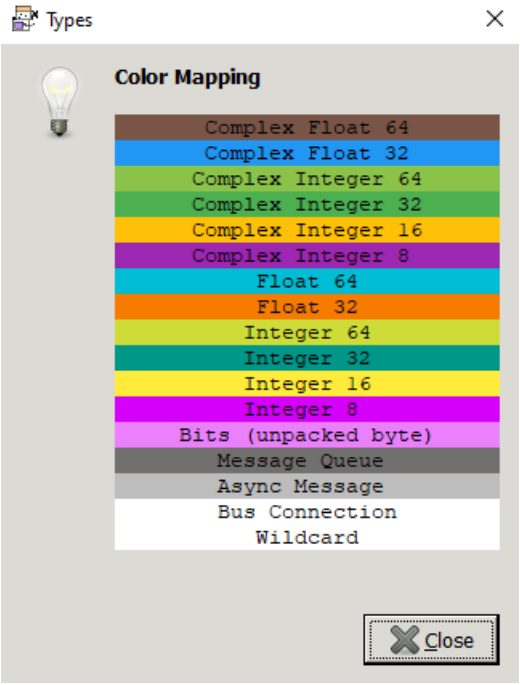


Abbildung 38: Liste der in GRC verwendbaren Datentypen und deren individuellen farblichen Kennzeichnung (Quelle: Eigene Darstellung)

Die Signalquelle hat einen blauen Ausgang, welcher auf den Datentyp eines komplexen 32Bit-Floats hinweist. Dies spiegelt auch die Datenbereitstellung eines Funksignals wider. Es wird hierbei ein komplexes Signal, also bestehend aus Real- und Imaginärteil, in jeweils nacheinander folgenden 32bit großen Fließkommawerten abgespeichert/aufgereiht. Sinn der Schaltung ist das Entdecken dieser Funktion. Der zum Detektieren notwendige Part des Phasenversatzes von 90° lässt sich in der Anzeige bzw. beim Ausführen des Flowgraphen erkennen. Die im Bild 37 gezeigten Elemente mit der Beschriftung „Realteil“ und der darunter dargestellte „Imaginärteil“ stellen verglichen diesen Versatz dar. Für die Verdeutlichung des Zusammenspiels und für die resultierende Ortskurve eines harmonischen Signals, in diesem Fall der Kosinus, wird die „Constellation Sink“ hinzugezogen. Diese wertet das, direkt aus der Signalquelle kommende, komplexe Signal aus. Die differenzierte Teilbeschreibung der generierten Frequenz wird im Zeitbereich über „Time Sinks“ angezeigt. Um dies zu realisieren wird das komplexe Signal zunächst mit dem Block „Complex to Float“ in seinen Real- und seinen Imaginärteil gesplittet. Dies lässt in die eben erwähnte Arbeitsweise des ADCs blicken. Der Throttle dient zur Lastminderung der CPU des Rechners. Er begrenzt den zu bearbeitenden Datenstrom auf die eingestellte Sampling-Rate. Eine nützliche aber für die Funktionsweise nicht relevante Komponente ist die „Note“-Einheit. Mit ihr lässt sich eine Notiz an geeigneter Stelle einbringen. Alternativ dazu gibt es die Möglichkeit Blöcke, unter dem jeweiligen Reiter „Advanced“, direkt mit einem Kommentar zu versehen.

4.3.2 Zweite Schaltung (einfacher FSK-Signalgenerator)

Um Themenrelevanz zu bewahren und sich generell mit dem Bereich Frequenzmodulation in GNU Radio zu befassen, wurde anschließend, nach dem erlangen elementarer Grundkenntnisse, ein FSK-Signalgenerator entworfen.

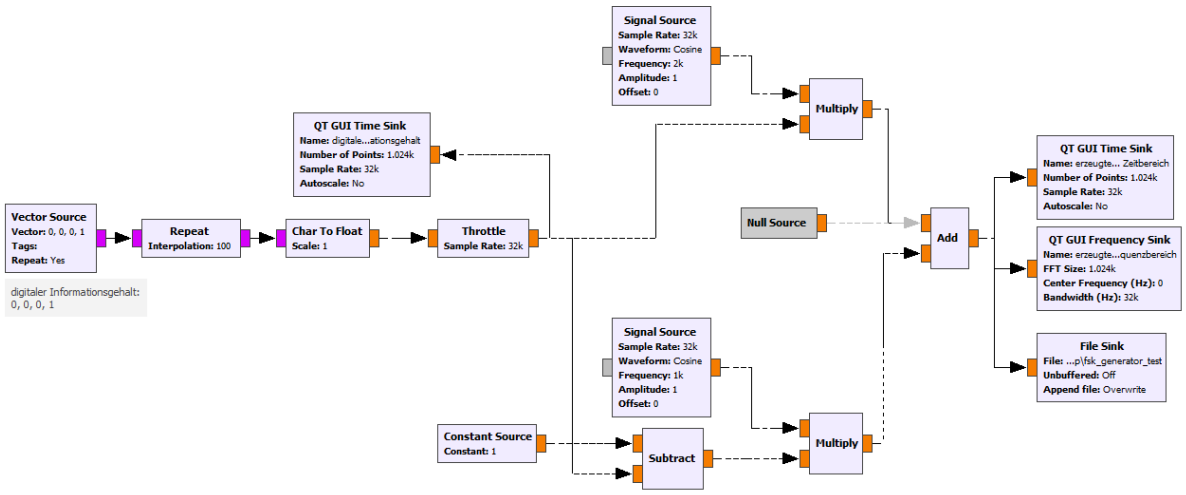


Abbildung 39: GRC-Flowgraph eines einfachen FSK-Signalgenerators
(Quelle: Eigene Darstellung)

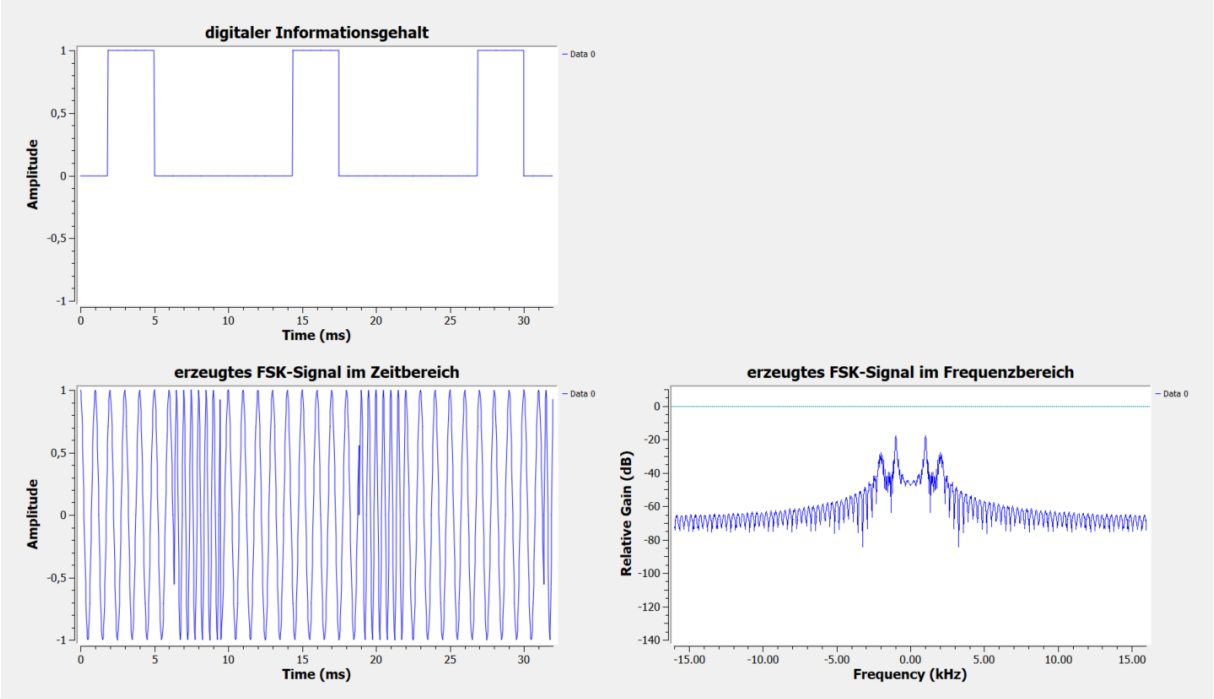


Abbildung 40: Ausführung der unter Abb. 39 gezeigten Schaltung
(Quelle: Eigene Darstellung)

Dieses Programm generiert ein frequenzmoduliertes Signal auf der Basis von 2 zusammengeführten Kosinusschwingungen und dem darzustellen wollenden digitalen Informationsgehalt. Letzteres befindet sich in der „Vector Source“ und beträgt 0,0,0,1

(Datentyp der Vector Source ist Byte). Um aus den Daten eine angenehme Anzeige zu erzeugen und um sich dabei ebenfalls viel Tipparbeit zu ersparen, wird ein „Repeat-Block“ eingefügt mit einem Interpolationswert von 100. Der Interpolationswert gibt an, wie oft ein eingehender Wert wiederholt wird. Ein anschauliches Beispiel dazu ist:

Inhalt Vector-Source = x, Interpolationswert = y, Ergebnis = z

x = 0,1; y = 1; z = 0,1

x = 0,1; y = 2; z = 0,0,1,1

x = 0,1; y = 5; z = 0,0,0,0,0,1,1,1,1,1

x = 0,0,1; y = 3; z = 0,0,0,0,0,0,1,1,1

Dieser Sachverhalt lässt sich in der Anzeige in Abbildung 40 unter „digitaler Informationsgehalt“ beobachten. Im Zeitgraph wird hier in gleichmäßigen Abschnitten die digitale 0 (Amplitude auf 0) und 1 (Amplitude auf 1) dargestellt.

Für die weitere Frequenzmodulation wird hierbei nun der Weg der Signale in die der 0 zugeordneten niedrigen Frequenzen und die der 1 zugeordneten hohen Fequenzen unterteilt, moduliert und anschließend zusammengeführt. Um dieses Prinzip besser veranschaulichen zu können, wurden separate Aufnahmen der Schaltung für den jeweiligen Strang getätigt.

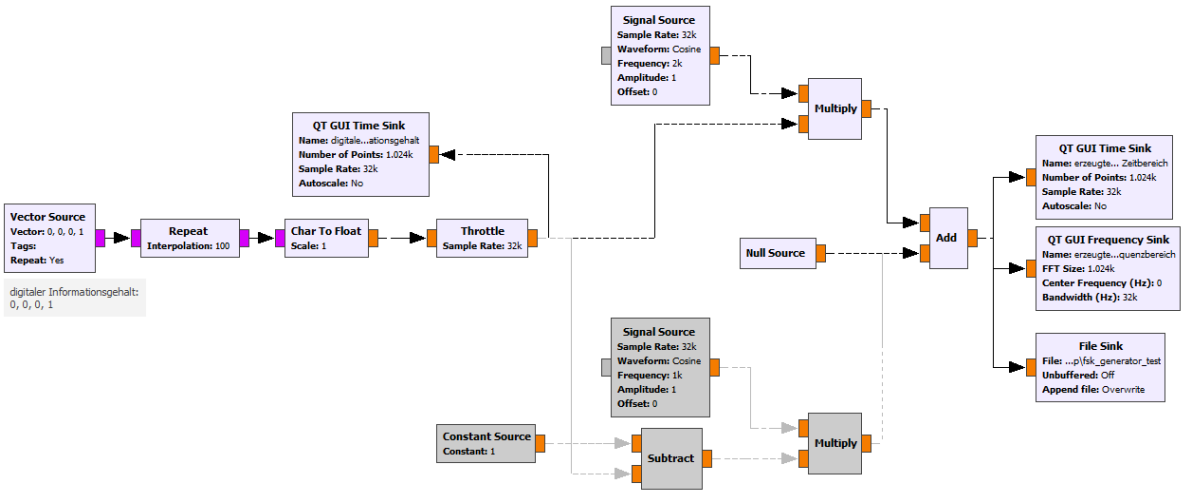


Abbildung 41: ausschließlich funktionierender Part der 1-Signale zur Teilanalyse (Quelle: Eigene Darstellung)

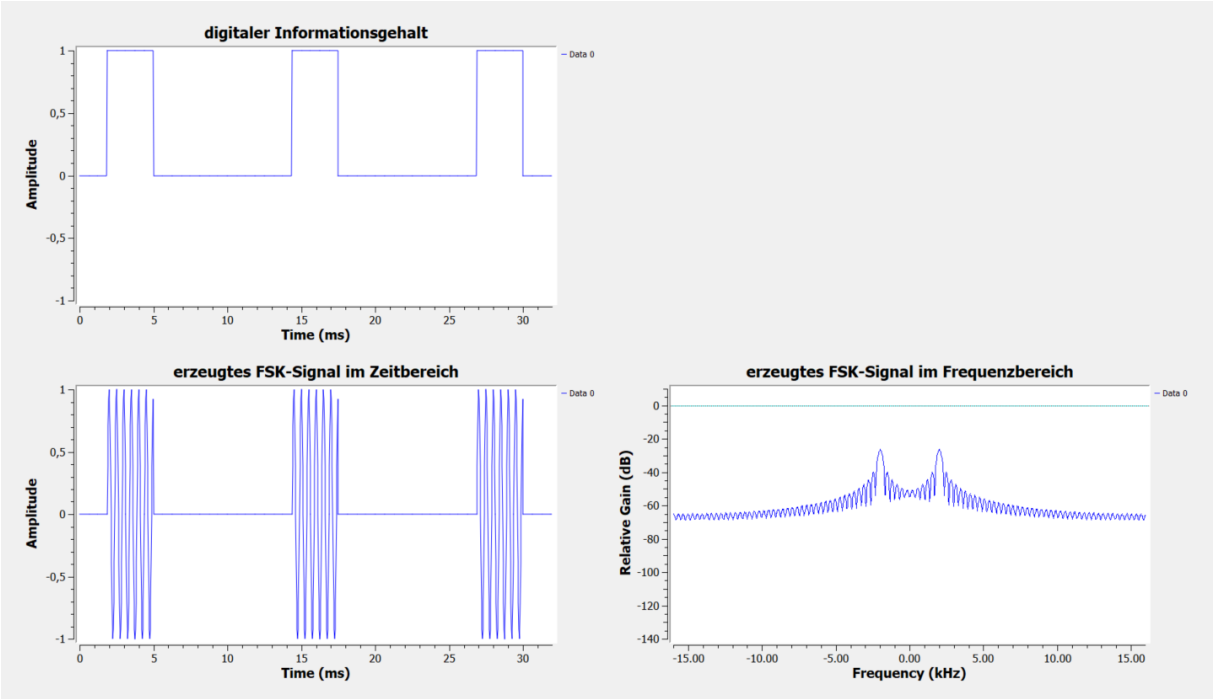


Abbildung 42: Ausführung der unter Abb. 41 gezeigten Schaltung
(Quelle: Eigene Darstellung)

Grauhinterlegte Blöcke in einer Schaltung sind funktionsunfähig gemacht. Es greift somit nur der „obere Pfad“. In diesem werden die Informationen aus der Vector Source mit dem aus einem Signalgenerator stammenden Kosinussignal von 2kHz multipliziert. Heißt, wenn eine digitale 0 anliegt, wird die Frequenz mit 0 multipliziert und entsprechend mit dem Ergebnis 0 dargestellt. Wenn eine 1 anliegt, wird für diese Dauer die Kosinusschwingung „durchgelassen“. Somit ist der obere Pfad für die partielle Zuteilung der obereren Frequenz eines FSK-Signal, in Verbindung mit der digitalen 1, zuständig.

Für eine modulierte 0 sieht es wie folgt aus:

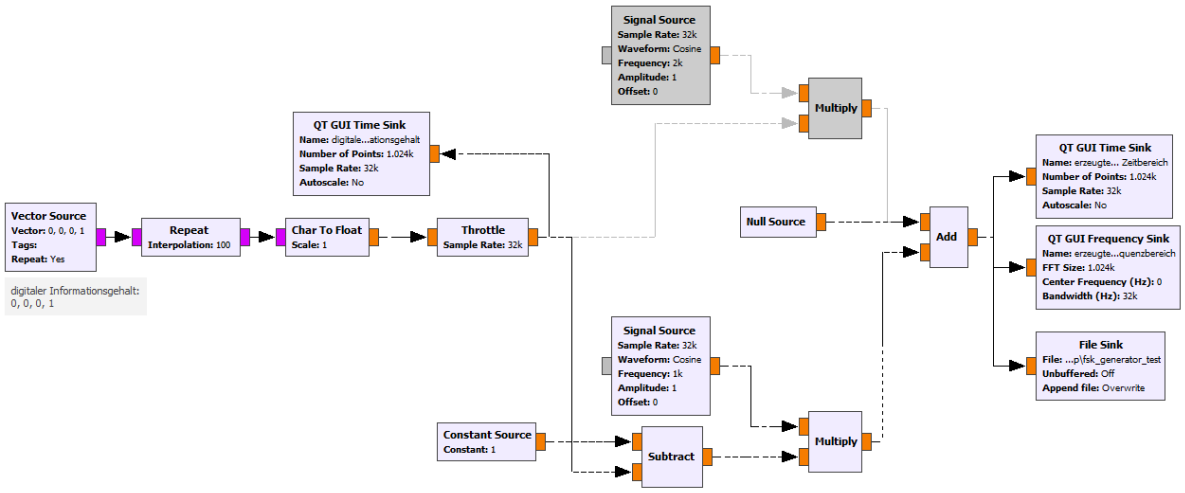


Abbildung 43: ausschließlich funktionierender Part der 0-Signale zur Teilanalyse (Quelle: Eigene Darstellung)

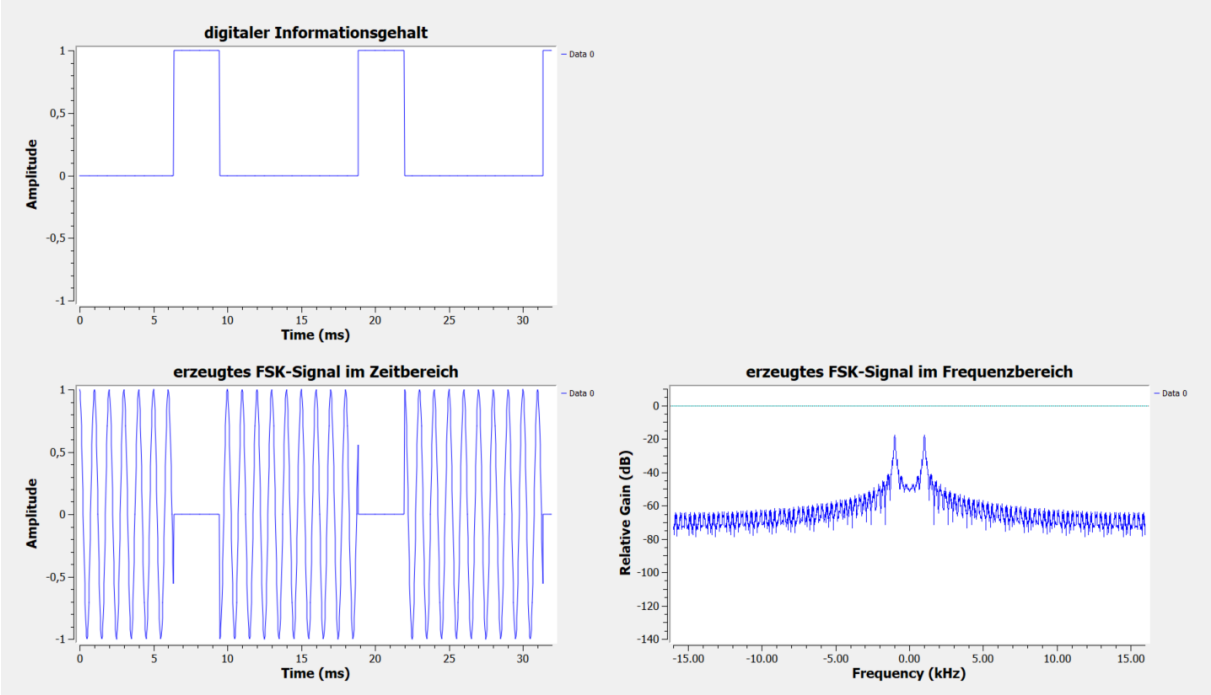


Abbildung 44: Ausführung der unter Abb. 43 gezeigten Schaltung (Quelle: Eigene Darstellung)

Das Prinzip ähnelt dem der digitalen 1. Jedoch handelt es sich hierbei um die konjugierte Form, welche in einer anderen, niedrigeren Frequenz dargestellt wird. Ebenfalls wird auch in diesem Pfad der digitale Informationsgehalt mit einer Kosinusschwingung (hier mit einer Frequenz von 1kHz) multipliziert. Da allein dieses Vorhaben eine Überlagerung der bereits im oberen Pfad erzeugten Modulation zur Folge hätte, werden hier die aus der Vector Source kommenden Informationen negiert. Diese Funktion übernimmt der „Subtract-Block“ (eine

mathematische Subtraktion) in Verbindung mit einer „Constant Source“. Die Constant Source, mit dem festen Wert von 1, stellt dabei den Minuenden dar. Die digitale Information (0,0,0,1) ist der Subtrahend. Rechnet man nun die Werte gegen, erzeugt letztendlich diese Subtraktion das invertierte Signal der Vector Source, wie in Abbildung 44, unter der Anzeige „erzeugtes FSK-Signal im Zeitbereich“, im Vergleich zur Anzeige „digitaler Informationsgehalt“, gut zu sehen ist. Eine digitale 0 wird einer harmonischen Schwingung einer, im Gegensatz zu dem oberen Pfad, niedrigeren Frequenz zugeordnet (siehe Bild 43), wobei eine digitale 1 keine Schwingung erzeugt.

Diese beiden erzeugten Signale (des oberen und des unteren Pfades) werden anschließend mit dem Add-Block (Addition) zusammengeführt. Es entsteht somit ein veranschaulichtes Beispiel eines FSK-Signals, wie in Abbildung 40 zu sehen ist.

4.3.3 Dritte Schaltung (wmbus_demod_2)

Um sich weiter an das Hauptthema, und somit verbunden auch reale Signale, zu tasten, entstand die dritte Schaltung. Hierbei wurde erstmalig das separate SDR-Gerät (hier der LimeSDR-USB) angeschlossen und verwendet.

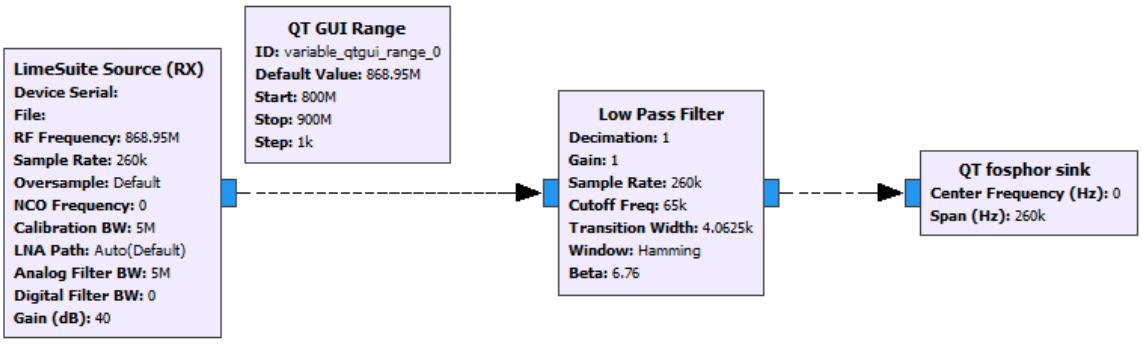


Abbildung 45: GRC-Flowgraph zum Entdecken erster realer Signale
(Quelle: Eigene Darstellung)

Der in Abbildung 45 gezeigte Block mit dem Namen „LimeSuite Source (RX)“ stellt genau dieses Gerät dar. Die Abkürzung RX steht auch hierbei, wie in der Technik üblich, für Receiver. Dies kennzeichnet, dass dieser Block dafür gedacht ist Signale zu empfangen (daher auch Source - Signalquelle). Diese Funktionseinheit wurde manuell in GNU Radio eingefügt und im Vorfeld mit dem Tool „Lime Suite“, entsprechend der Lime-Hardware, eingerichtet. Die 3 wichtigsten Parameter in den Einstelloptionen dieses Blocks sind zum einen die „RF Frequency“, welche das System auf die abzuzielende Trägerfrequenz einstimmt. Zum anderen

kommt es hierbei auf die Sample-Rate an. Die Verschiebung eines Signals in das Basisband bringt den großen Vorteil mit sich, dass die höchste vorkommende Frequenz in der Regel stark verringert wird. Die daraus resultierende Abtastrate, nach dem Nyquist-Shannon-Abtasttheorem, richtet sich nun nur noch nach der Bandbreite des spezifischen Spektrums. Entsprechend muss die Sample-Rate nicht endlos hoch sein. Sollte das empfangen zu wollende Signal zu schwach sichtbar sein, gibt es die Möglichkeit dieses unter „Gain“ zu verstärken. Ein weiterer wichtiger Punkt ist der „LNA Path“. Dieser steht standartmäßig auf autodetect und sucht sich somit den zu der eingestellten RF passenden Frequenzbereich. Neben dem autodetect-mode lassen sich die Parameter „H“, „L“ und „W“ manuell verwenden, welche für „High“, „Low“ und „Wide“ stehen. Die vollständige Bezeichnung einer Option wäre zum Beispiel „LNAL“. LNA steht dabei für Low Noise Amplifier, also für einen rauscharmen Verstärker. Dieser bewirkt ein Verstärken von Signalen in einem ausgewählten Bereich. Der Low-Bereich eignet sich laut Herstellerangaben für ein Spektrum von 700MHz bis 900MHz. Der wM-Bus-Bereich wird damit gegebenenfalls abgedeckt. Auch eine passende Antenne wurde für den Empfang gewählt. Es handelt sich hierbei um eine ALLNET-GSM-Antenne, die unter anderen für den Bereich 690MHz bis 960MHz ausgelegt ist. Somit lässt sich sagen, dass die Antenne eine erste Filterung der Frequenzen vornimmt. Um die einzustellende bzw. zu untersuchende Frequenz anfänglich varrieren zu können, wird hier eine Range-Variable unter „RF Frequency“ im LimeSuite-Source-Block verwendet. Somit ist ein Absuchen in einem bestimmten Spektrum (hier 800-900MHz) möglich. Optional ließe sich das auch mit einem externen Tool realisieren. Da die Performance dieser Tools in der Regel besser gelingt, empfiehlt sich diese Variante bei der Suche nach Frequenzen. Eine Möglichkeit ist die Verwendung des Programms „CubicSDR“.

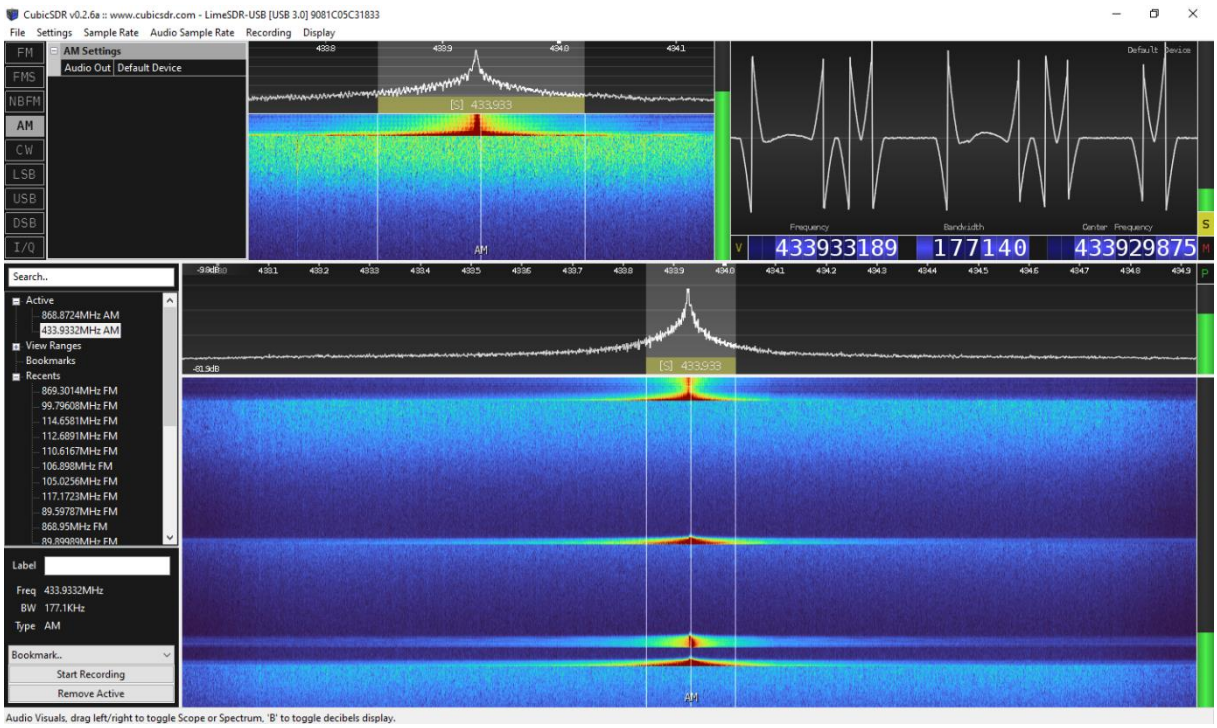


Abbildung 46: Beispiel der Signalfindung mit CubicSDR
(Quelle: Eigene Darstellung)

Wie in Abbildung 46 zu sehen ist, kann mit Hilfe solcher Tools ein erster Überblick möglich auftretender Signale erlangt werden. Hierbei ist zu sehen: auf welcher Trägerfrequenz das Signal operiert, erste Hinweise auf die verwendete Modulationsart und die ungefähr genutzte Bandbreite. Ein geschulter Blick erkennt sogar anhand dieser Informationen um welche Art von Signal (Sprechfunk über Radio, Sattelietenbilder, Datenaustausch von Geräten, etc.) es sich handeln könnte. Mit diesen Informationen geht man dann mit GNU Radio ins Rennen und versucht anhand der bereits gewonnen Sachlage, einen entsprechenden Flowgraph für die jeweiligen gesetzten Ziele zu entwerfen.

Neben der ersten „Filterung“/Selektion per Source-Block in Kombination mit einer zu dem Frequenzbereich passenden Antenne, lässt sich das Signal generell Filtern. Die gezeigte Methode ist die Verwendung eines Tiefpassfilters („Low Pass Filter“-Block). Wir wissen, dass sich die hier verwendete Sample-Rate auf die Bandbreite des Signals, durch die Transformation dessen in das Basisband, bezieht. So wird auch hier unter dem Parameterpunkt „Sample Rate“ diese übernommen. Ein Decimation-Faktor ist ebenfalls einstellbar. Dieser verringert die Abtastrate des nachfolgenden Systems um diesen Wert als Teiler. Hat man also eine anfängliche Sample-Rate von 2MSps und einen wirkenden Decimation-Faktor von beispielsweise 1000, hat das nachfolgende System eine effektive Abtastrate von 2kSps. Eine Decimation kann auch Verstanden werden als eine Art hineinzoomen in ein bestimmtes Frequenzspektrum. Es erzielt beispielweise in einer

Frequenzsenke eine höhere Frequenzauflösung, da diese eine bestimmte FFT Size, also eine Fenstergröße, besitzt.

Die FFT Size gibt dabei an, in wie viele gleich große Abschnitte (Kästchen/ Streifen) ein Fenster unterteilt wird. Man kann also sagen, dass eines dieser Kästchen ein bestimmtes Spektrum an gesampelten Werten beinhaltet und somit die Frequenzauflösung des Fensters darstellt. Generell ist die FFT eine wichtige Messmethode in der Frequenztechnik. Sie gibt die Spektralkomponenten eines Signals an und somit Aufschluss über dessen Zusammensetzung. Will man so eine Größe bestimmen, sind zunächst 2 Komponenten von Bedeutung. Zum einen die Abtastrate des Messsystems (f_s) und zum anderen die Anzahl der Samples die abtgetastet werden sollen, was hier als Blocklänge (L_B) bezeichnet wird und immer eine ganzzahlige Potenz zur Basis 2 ist. Betrachtet man nun die Bandbreite des Signalspektrums nach dem Nyquist-Shannon-Theorem (f_n), ergibt dieser Wert die theoretisch maximal zu bestimmende Frequenz durch die FFT an. Das Verhätlnis lässt sich wie folgt ausdrücken:

$$f_n = \frac{f_s}{2}$$

Formel 1: Nyquist-Frequenz/ Bandbreite und die Beziehung zur Abtastrate

Neben der Betrachtung des Abtastverhältnisses im Spektrum der Frequenz, ist die zeitliche Komponente, also die Messdauer (D_M), ebenfalls von belangen. Die Messdauer ist dabei das Verhältnis von Blocklänge zu Abtastrate.

$$D_M = \frac{L_B}{f_s}$$

Formel 2: Signalmessdauer eines FFT-Blocks

Um jetzt eine Aussage über die entscheidene Frequenzauflösung (d_f) treffen zu können, bildet man das Reziproke der Messdauer.

$$d_f = \frac{f_s}{L_B}$$

Formel 3: Kehrwert der Messdauer als Differenz zwischen 2 Messwerten (folglich Messauflösung)

Auf Grund dieser Abhängigkeiten kann man zu dem Schluss kommen, dass die Blocklänge, also die Anzahl der abzutastenden Samples, die Geschwindigkeit der Messwiederholung und somit die Frequenzauflösung bestimmt. Die Blocklänge (im Englischen auch Bins genannt)

bedingt also bei beispielsweise sinkender Sampling-Rate (f_s) eine höhere Frequenzauflösung (d_f).

Dadurch kann der Decimation-Faktor, der die Sampling-Rate für das nachfolgende System verringert, eine wichtige Rolle zur Signalanalyse spielen. Es werden Teile des Spektrums sichtbar gemacht, welche für die Analyse eine ausschlaggebene Wichtung haben könnten, wie in Abbildung 47 verdeutlichend dargestellt ist.

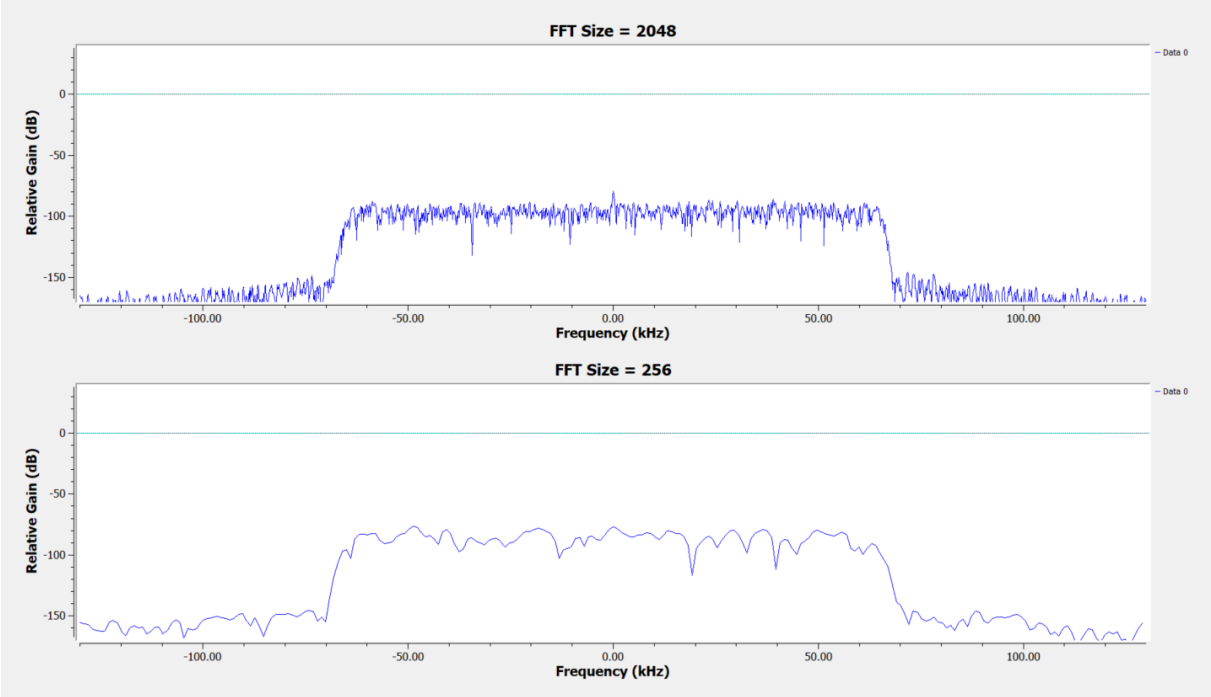


Abbildung 47: Demonstration zur Wirkung der FFT-Größe
(Quelle: Eigene Darstellung)

Folgende Grafik soll diesen Zusammenhang nocheinmal veranschaulichen (D = Decimation-Faktor):

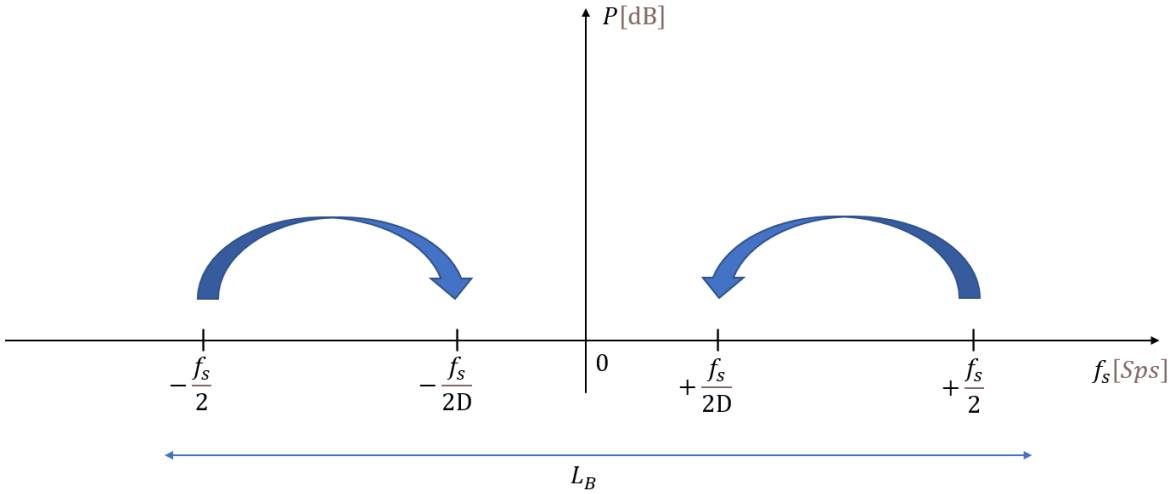


Abbildung 48: bildliche Darstellung eines wirkenden Dezimierungsfaktors
(Quelle: Eigene Darstellung)

Eine FFT (Fast Fourier Transform) eignet sich allerdings nur für periodische Signale. Demnach tauchen einzelne Signalabschnitte außerhalb des zu betrachten wollenden Spektrums in periodischen Abständen wieder auf. Um diesen Effekt, auch „aliasing“ genannt, zu verhindern, wirkt der Tiefpassfilter dem entgegen. Seine Eigenschaft besteht darin, nur einen bestimmten Frequenzbereich, der unter einer zu definierenden Schwelle liegt, passieren zu lassen. Alle weiteren, höheren Frequenzanteile des Spektrums werden dem Filter entsprechend stark gedämpft.

Die charakterischen Merkmale eines solchen Filters sind vorallen die „Cut-Off-Frequency“ und die „Transition-Width“. Die Cut-Off-Frequency, oder auch Eck-Frequenz genannt, gibt an, bis zu welcher Frequenz, bei einem Tiefpassfilter, ein Signal ungedämpft durchgelassen werden kann. Die Transition-Width versteht sich als Maß für die „Steilheit“ eines Filters und gibt die Übergangsbreite zwischen dem Sperr- und Durchlassband an. Folgende Abbildung (49) demonstriert den Unterschied einer relativ hohen und einer relativ geringen Transition-Width.

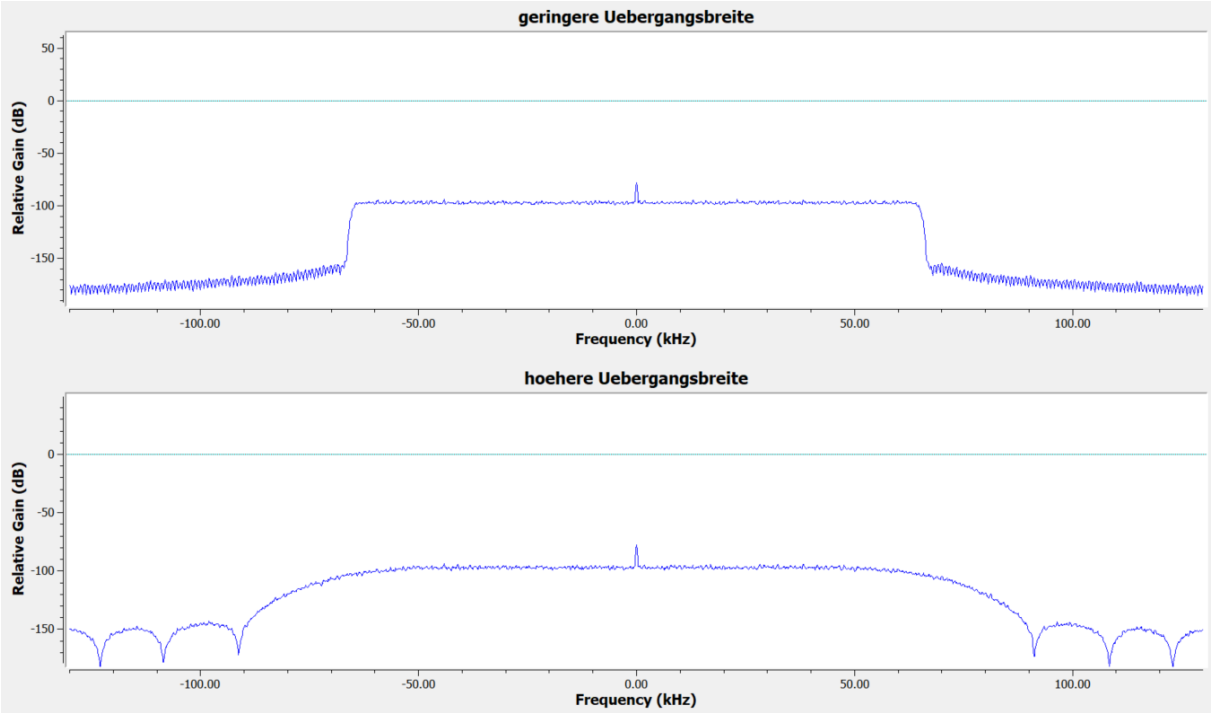


Abbildung 49: Gegenüberstellung verschiedener Übergangsbreiten eines Filters (Quelle: Eigene Darstellung)

Es handelt bei diesem Versuch um Frequenzanzeigen, welche mit ein und dem selben Signal gespeist wurden und auch sonst, bis auf die Übergangsbreite, die selben Parameterwerte besitzen. Wie man erkennen kann, stellt sie ein bedeutendes Maß für die Güte des Filters dar. [21] [22]

4.4 Konkrete Lösung und deren Ausarbeitung

4.4.0 Vorwort

Mit den bereits gewonnenen Erkenntnissen, im Umgang mit GRC und diversen Funktionseinheiten, wird ein Konzept entwickelt, welches es ermöglicht ein FSK-Signal zu empfangen und die demodulierten Frequnzen für weitere Verarbeitungsschritte bereitzustellen.

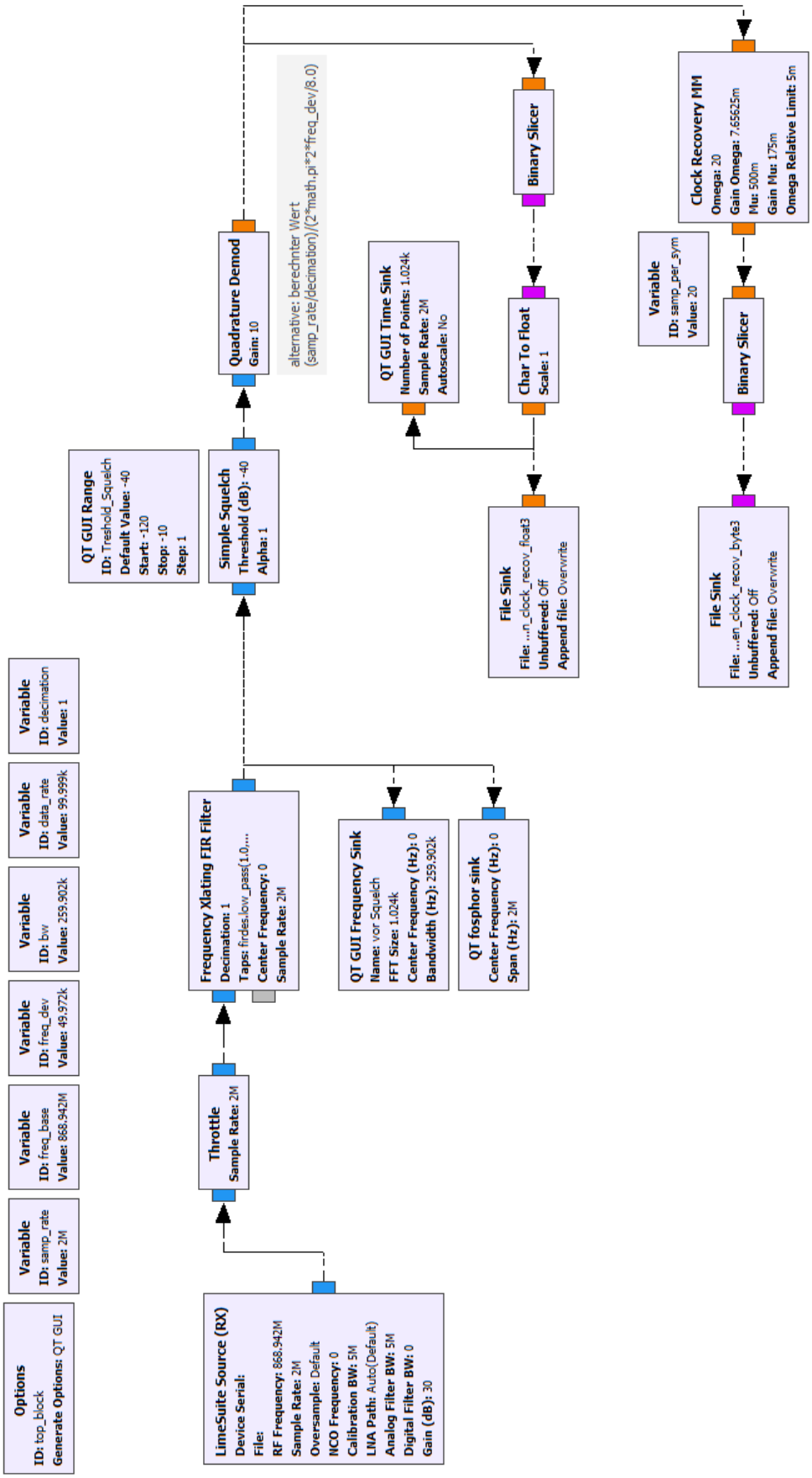


Abbildung 50: GRC-Flowgraph zum empfangen und demodulieren FSK-basierender wM-Bus-Signale (Quelle: Eigene Darstellung)

Abbildung 50 zeigt den in GNU Radio entwickelten Flowgraphen zu diesem Unterfangen. Für eine bessere Erklärbarkeit der einzelnen Strukturen, findet sich in Abbildung 51 eine detailierte Funktionsgruppengliederung wieder.

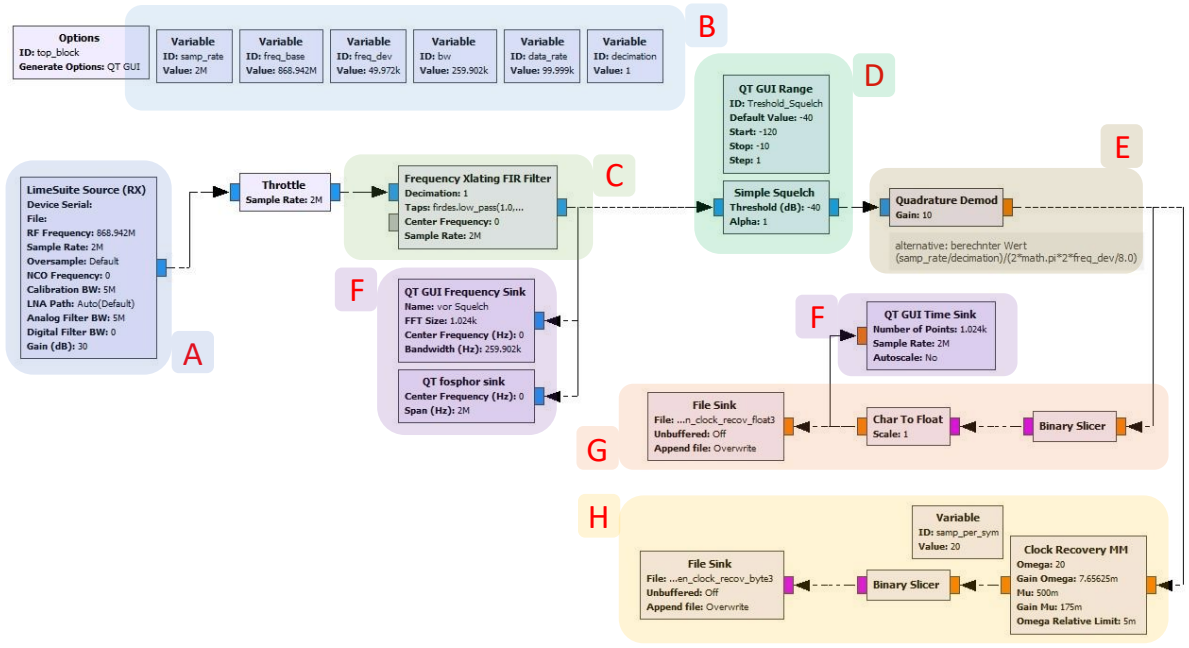


Abbildung 51: segmentierte Darstellung, zur besseren Erklärbarkeit, der in Abb. 50 gezeigten Schaltung (Quelle: Eigene Darstellung)

4.4.1 Angewendete Lime-Source (RX)

Der mit A markierte Bereich beinhaltet die LimeSuite-Source (RX), welcher das digitale Äquivalent zum in der Realität angeschlossenen LimeSDR-USB darstellt. Es wird hierbei auf die Trägerfrequenz des Versuchs-wM-Bus-Geräts abgezielt. Diese beträgt 868,942MHz und wurde hier über die Variable „freq_base“ als Parameter hinterlegt. Zu sehen ist zu dem eine eingestellte Sampling-Rate von 2MSps und eine gewählte Vorverstärkung von 30dB unter dem Paramter „Gain“. Der „LNA Path“ ist hier auf „Auto“ belassen wurden. Dieser sucht sich die angeschlossene und zur eingestellten Frequenz passende Buchse, welche hier Channel 0 (RX) ist und eine LNAL-Antenne stellt. Der LNAL ist laut Herstellerangaben der Bereich in dem die 868,942MHz liegen (700MHz bis 900MHz).

4.4.2 Verwendete Variablen und optionale Erfassung derer

Bereich B präsentiert die für die Schaltung relevanten Variablen. Diese wurden als solche gewählt, um zum einen eine Übersicht der signalbetreffenden Gegegbenheiten zu haben und zum anderen um diese an mehreren Stellen übersichtlich und effektiv verwenden bzw. ändern zu können. Unter der ID „samp_rate“ verbirgt sich die Abtastrate des Systems. Diese beträgt hier, wie bereits erwähnt, 2MSps und wird an diversen Stellen zur Parametrierung verschiedenener Abhängigkeiten genutzt. Unter „freq_base“ wird mit der Trägerfrequenz des Signals gearbeitet. Die einem FSK-Signal zugehörige und notwendige Abweichung (engl.: deviation) zum Carrier lässt sich unter „freq_dev“ finden. Mit dieser Charakteristik wird der digitale Informationgehalt in analoge Frequenzen umgesetzt. Eine Abweichung in die von der Trägerfrequenz untere Ebene bewirkt eine digitale 0. Ein Abschnitt mit einer kurzweilig konstant höheren Frequenz zum Carrier-Signal stellt eine digitale 1 dar. Die Deviation beträgt in diesem Fall ±50kHz. „bw“ steht hier für Bandbreite (engl.: bandwidth) und beträgt ca. 260kHz. Konkreter gemeint ist damit die analoge Bandbreite des Signals, also die Frequenzspektrumsbreite in dem sich die Informationen bewegen. Von hingegen einer digitalen Bandbreite spricht man, wenn es um die Datenrate geht. Diese wird in dem oben gezeigten Beispiel als „data_rate“ deklariert. Die Variable „decimation“ dient zum einstellen des Dezimierungsfaktors und beträgt, nach mehreren Versuchen, 1, da dies hier ein im Gesamtkonzept gut auswertbares Ergebnis liefert.

Dass diese wichtigen Eckdaten generell von vornherein gegeben sind, tritt vermutlich in den wenigsten Fällen auf. Die konkreten Werte für diese Schaltung wurden, aus den im Vorfeld definierten Parametern, aus dem Sende-Tool übernommen.

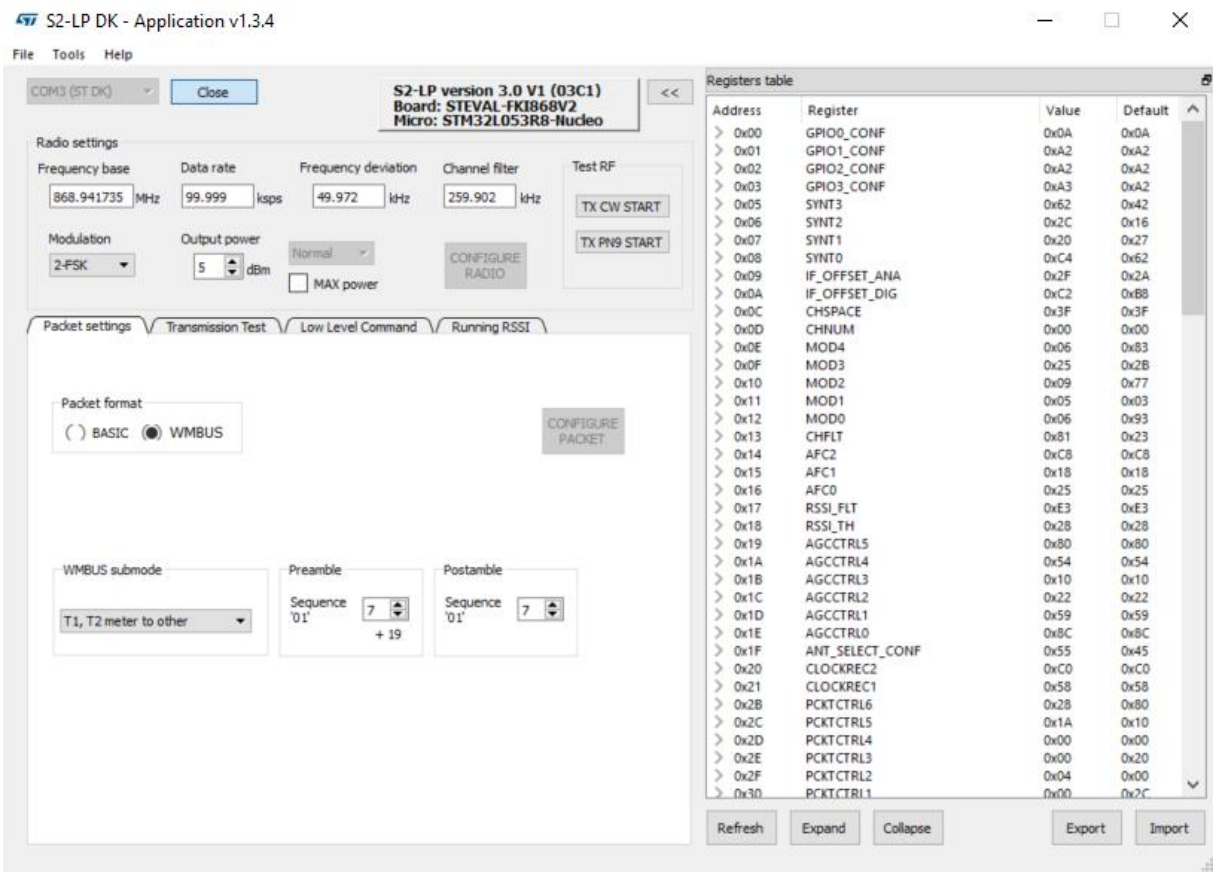


Abbildung 52: Konfigurationswerte eines Testtransmitters (Teil 1)
(Quelle: Eigene Darstellung)

Wie in Abbildung 52 zu sehen, entsprechen die konfigurierten Parameter den Werten der Variablen im GRC-Flowgraph. Ebenfalls sind dies die Konfigurationswerte des hier ebenfalls zur Erprobung verwendeten „wM-Bus-Dummys“.

Aus der Abbildung ist ebenfalls ersichtlich, dass es sich bei dem zu sendenen Signal um eine 2-FSK-Modulation handelt. Das Paketformat ist hierbei wM-Bus, genauer gesagt im Subtyp „T1, T2, meter to other“. Ebenfalls lässt sich eine Pre- und Postamble mit einer Sequenz von „01“ in einer definierbaren Wiederholzahl festlegen.

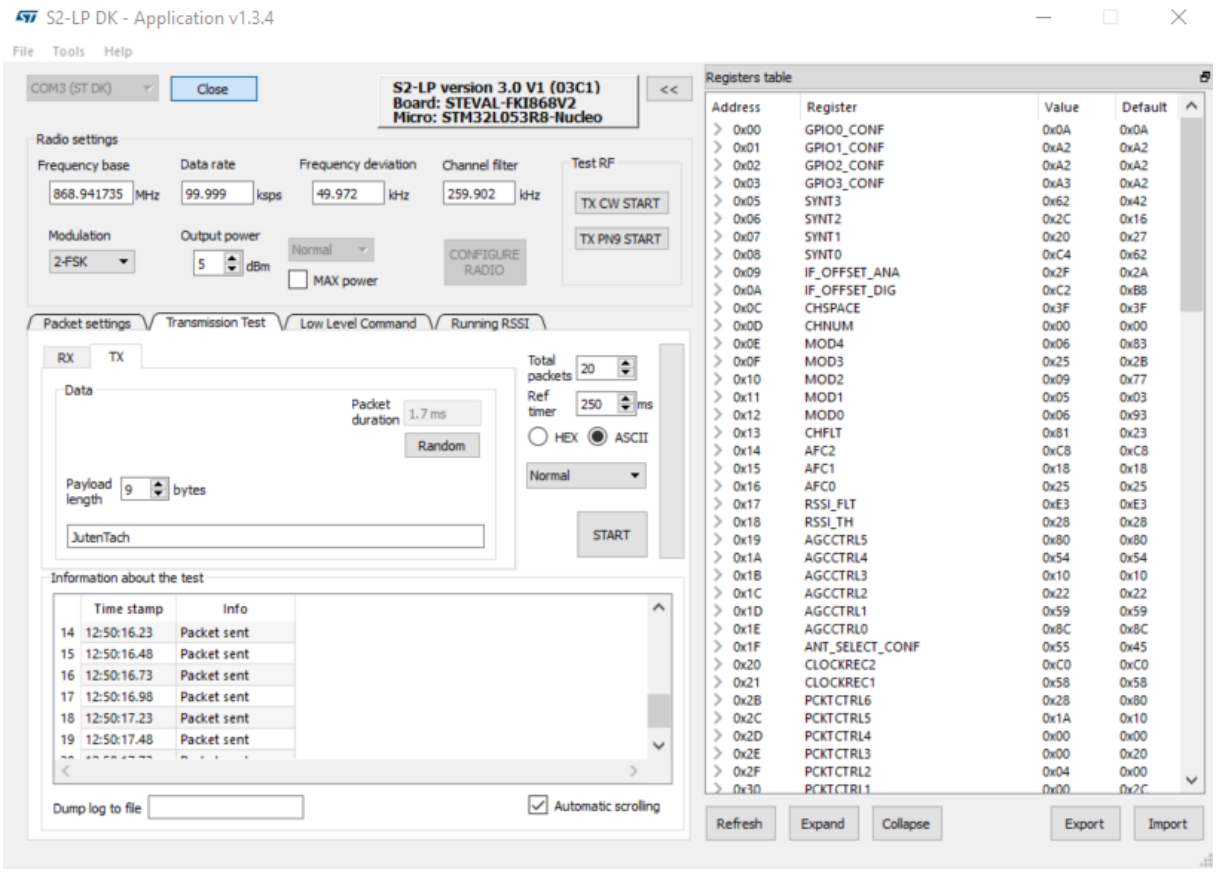


Abbildung 53: Konfigurationswerte eines Testtransmitters (Teil 2)
(Quelle: Eigene Darstellung)

In dem Reiter „Transmission Test“ wird die zu sendene Nachricht, im Format Hexadezimal oder ASCII, hinterlegt. Der Nachrichtentext ist dabei frei wählbar. Der aus dem eingegebenen Informationsgehalt entstehende Payload wird automatisch mitgezählt und in Byte angegeben. Dieser lässt sich jedoch manuell erweitern, also bei gleichen eingegebenen Text automatisch mit Leerstellen auffüllen, wenn man den Payload per Hand hochzählt. Einen wesentlichen Einfluss nimmt die Größe des zu übertragenden Informationsgehalts auf die Paketdauer (Packet duration). Desweiteren lässt sich die Anzahl der insgesamt zu senden wollenden Pakete unter „Total packets“ und die Intervallzeit zwischen den Nachrichten unter „Ref timer“ einstellen. Mit „Start“ wird die Übertragung mit den konfigurierten Werten umgesetzt.

Sollte es sich bei den auszuwertenden Signalen jedoch um „unbekannte“ Werte handeln, besteht die Möglichkeit sukzessiv sich diese zu erklären. Für eine anfängliche Signalfindung eignet sich beispielsweise das Programm „CubicSDR“.

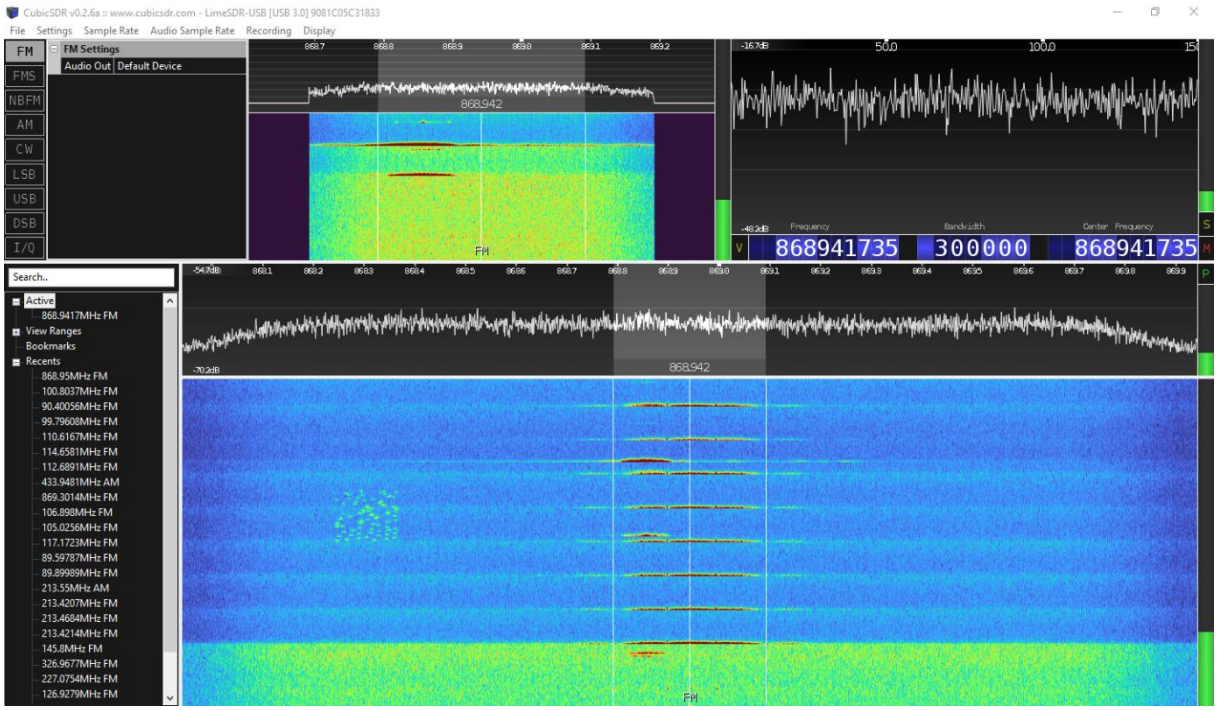


Abbildung 54: alternative Signalparametergewinnung in CubicSDR
(Quelle: Eigene Darstellung)

Um plausible Vergleichswerte zu haben, wird in Abbildung 54 das Testsignal, welches vom STM32-Spirit, über die in Abbildung 52 und 53 zu sehenden Parameter konfiguriert, ausgesendet wird, abgehört.

Die Trägerfrequenz lässt sich dabei optisch ermitteln. Sie befindet sich in der Mitte des Signals, also in der Mitte der Bandbreite. Die Bandreite bzw. das verwendete Spektrum ist hier als rote Linie zu sehen. Diese lässt sich hier für ein erstes grobes Abschätzen auf 300kHz ermitteln, jedoch noch etwas verringern, sodass die aus dem Spirit stammenden 260kHz Bandbreite optisch plausibel erscheinen. Die kurz aufeinander folgenden Nachrichten können, im Bezug auf die Intervallzeit, ignoriert werden. Relevant ist dabei lediglich ein einzelnes Signal. Da es sich um ein breites aber flaches Spektrum handelt, lässt sich eine frequenzabhängige Modulation mit kurzem Informationsgehalt erahnen. Mit diesen Informationen kann man in GNU Radio nun einen ersten Flowgraph für weitere Analysezwecke erstellen.

Diese Vorgehensweise ist bei Signalen verschiedenster Art anwendbar. Ebenfalls kann dabei die Lage der Trägerfrequenz oder die Streuung der Frequenzen Aufschluss über die Art/Verwendung der Information geben.

Eine erste Auswertung lässt sich bereits im GNU Radio mit diversen Anzeigeelementen vornehmen. Dies ist vor allen für eine Schaltungsentwicklung und Variablenfindung sehr zu empfehlen. Somit lassen sich einzelne Verarbeitungsschritte besser und gezielter

nachvollziehen. Für die Analyse eines Endresultats lässt sich beispielsweise mit dem Programm „Inspectrum“ arbeiten.

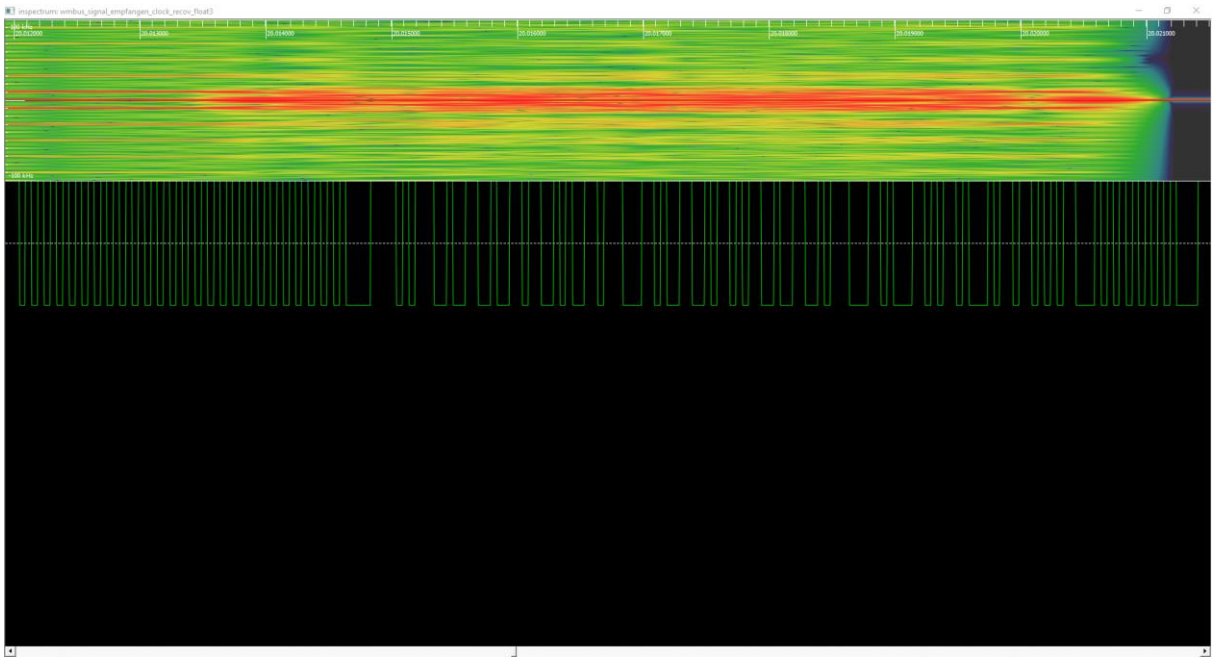


Abbildung 55: wM-Bus-Frame des Testtransmitters in Inspectrum
(Quelle: Eigene Darstellung)

Abbildung 55 zeigt den kompletten in Insprectrum dargestellten, mit dem Flowgraph aus Abbildung 51 aufgenommenen, wM-Bus-Frame. Für konkretere Analyseschritte lässt sich mit Hilfe der in den Abbildungen 56 und 57 gezeigten „Parameter- und Optionsleiste“ eine Auflistung gezielterer Werkzeuge darbieten.

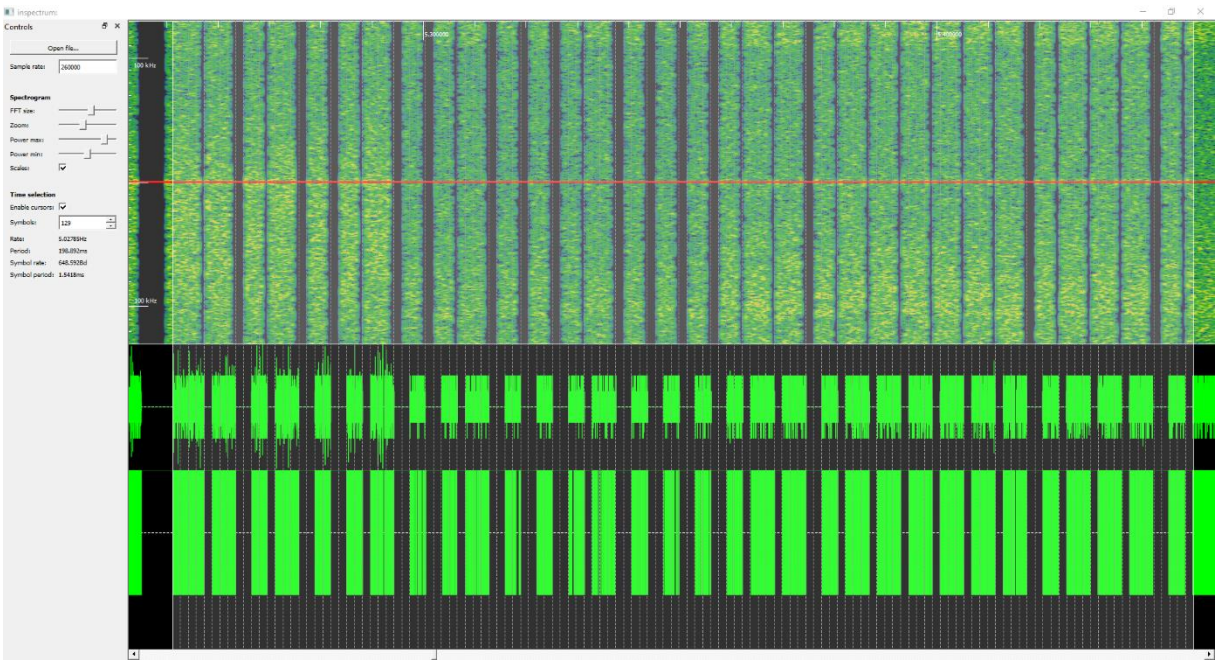


Abbildung 56: alternative Signalparametergewinnung in Inspectrum eines OOK-Signals mit unterteilten Einzelbits (Quelle: Eigene Darstellung)

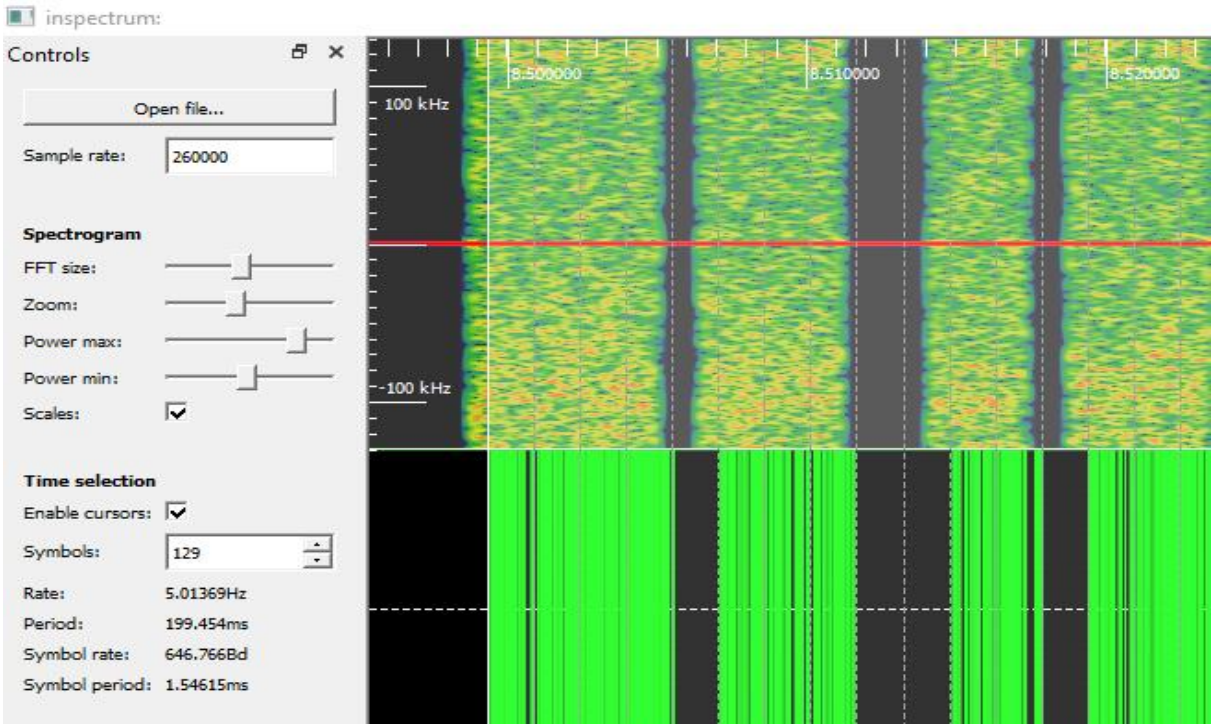


Abbildung 57: alternative Signalparametergewinnung in Inspectrum mit vergrößerter Optionsleiste (Quelle: Eigene Darstellung)

Nachdem ein Dateiabschnitt zur näheren Betrachtung herausgesucht und ein passender Plot eingestellt wurde, lässt sich mit „Enable cursors“ ein optische Segmentierung vornehmen. Dies soll dazu dienen, in den diversen Abschnitten, die Cursorbreite auf beispielweise 1 Bit zu skalieren. Ist die Breite passend, lässt sich mit Hilfe dieser unter anderen die Symbolrate

ablesen. Dabei ist darauf zu achten, dass die unter „Sample-Rate“ eingestellte Abtastrate der des Systems entspricht mit dem die Informationen aufgenommen wurden.

4.4.3 Angewendeter Frequency Translating FIR Filter

Bei dem unter Abschnitt C im Bild 58 markierten Bereich handelt es sich um die „zweite“ (neben der Antenne) maßgebliche Filtereinheit.

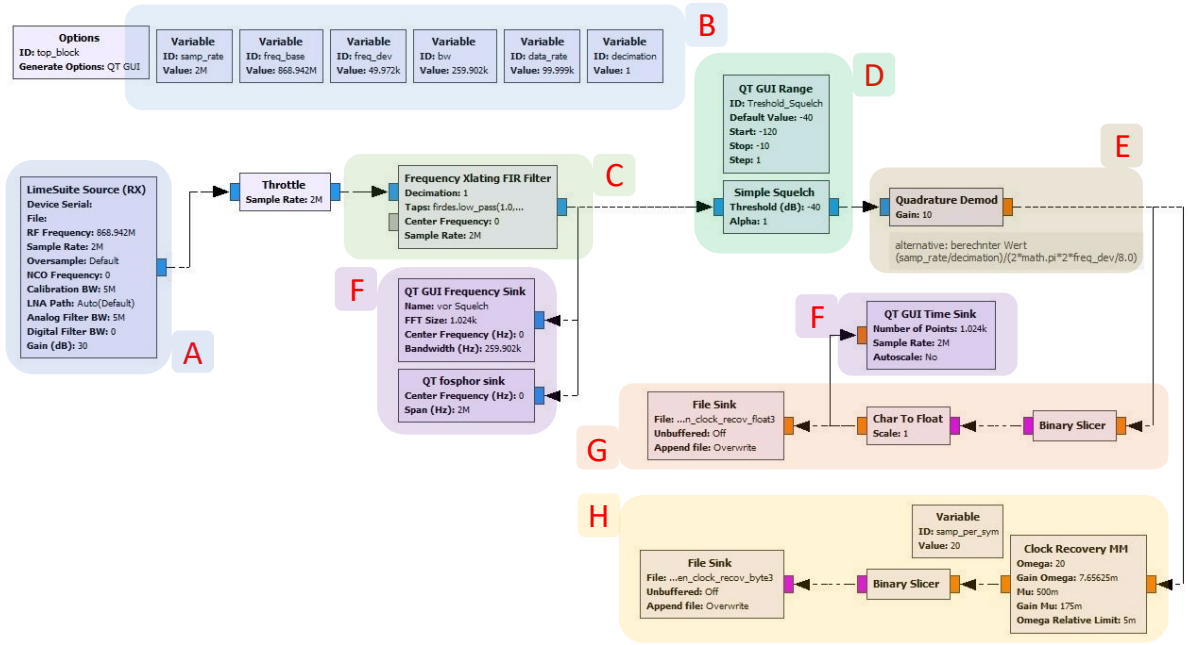


Abbildung 58: segmentierte Darstellung, zur besseren Erklärbarkeit, der in Abb. 50 gezeigten Schaltung (Quelle: Eigene Darstellung)

Der Frequency Translating FIR Filter bietet, wie bereits im Theoreiteil unter Kapitel 2.3.1 erwähnt, ein breites Maß an Funktionen und ist zudem kompakt designt. Die Verringerung der Sampling-Rate durch den Decimation-Parameter hat sich in diversen Testfällen in dieser Anwendung als nicht notwendig bis nicht sinnvoll erwiesen. Es wurde ohne Dezimierung ein besseres Anzeigeergebnis im Inspectrum erreicht. Aus diesem Grund wird der Wert hier auf 1 belassen. Generell ist das Grundprinzip der Dezimierung eine Verringerung der Sample-Rate durch eine Division dieser mit einem Divisor. Die rausgegebene Abtastrate entspricht einem Teil der eingehenden. Hintergrund der Anwendung dieses Verfahrens und zugleich großer Vorteil ist es, keine exorbitant hohen Abtastraten eines AD-Wandlers haben zu müssen. Die resultierende Sample-Rate wird dabei meist auf die Bandbreite des zu betrachtenden Signals/Spektrums angepasst, sodass nur relevante Werte gesampelt werden müssen und sich damit die Anforderung an die potenzielle Abtastrate des ADC verringert und somit kostengünstigere Varianten eingesetzt werden können. Man sollte allerdings darauf achten,

dass die dezimierte Abtastrate dennoch größer als die doppelte Bandbreite des Signals ist, um Datenverlust zu vermeiden (laut dem Nyquist-Shannon Abtasttheorem).

Wie auch andere Funktionseinheiten in GNU Radio besitzt dieser Filter-Block ebenfalls die Option einer Datentypenumwandlung. Der Output ist allerdings vorgegeben immer eine Auswahl aus diversen komplexen Werten. Es lässt sich dabei lediglich der Input-Datentyp und die weitere Datenverarbeitung auswählen. Für weitere Schritte kann in dem Abschnitt „Type“ entschieden werden, ob der Filter mit realen oder komplexen „Taps“ arbeitet. Taps kann in diesem Kontext aus dem englischen frei übersetzt werden mit „Abgriff“.

Unter dem Reiter „Taps“ wird der eigentliche Filter konfiguriert. Die hier verwendete Konfiguration sieht wie folgt aus:

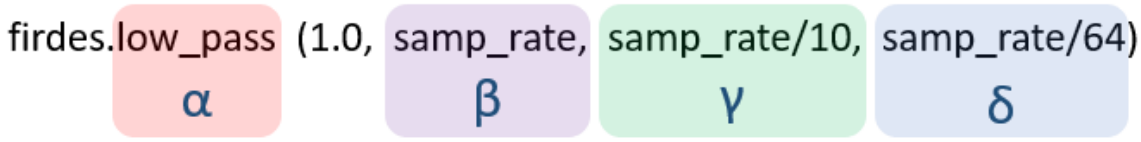


Abbildung 59: Filterkonfiguration mit unterteilten Abschnitten
(Quelle: Eigene Darstellung)

Im Bild 59 kennzeichnet Abschnitt α die allgemeine Initialisierung eines Tiefpassfilters. Seine Filtereigenschaften werden durch die in der Klammer stehenden Variablen definiert. Abschnitt β ist einer der drei wesentlichen Parametrierschächte. Hier wird die Sample-Rate eingegeben, welche allgemein im SDR-System verwendet wird. Auf der offiziellen GNU-Radio-Wiki-Seite zu diesem Block, wird die allgemeine Parametrierung wie folgt beschrieben:

*firdes.low_pass(1, samp_rate, samp_rate/(2*decimation), transition_bw)*

Hierbei sieht man deutlich, dass der Decimation-Faktor bei der Filterbreite zu berücksichtigen ist. Generell ist darauf zu achten, alle einer Dezimierung folgenden Blöcke, in der die Sample-Rate parametriert wird/ relevant ist, eine Berücksichtigung des Divisors beinhalten.

$$\frac{sample\ rate\ vor\ der\ decimation}{decimation\ value} = sample\ rate\ nach\ der\ decimation$$

Formel 4: zusammenfassende Beschreibung der Wirkung eines verwendeten Dezimierungsfaktors

Abschnitt γ ist der Bereich der Cut-Off-Frequency. Diese definiert im Filter den Grenzbereich des zu betrachtenden Spektrums. Sinnvollerweise sollte diese mit der maximalen Bandbreite des Signals enden, um unnötiges Rauschen auszuschließen aber auch keinen Informationsgehalt abzuschneiden. Im Bereich δ wird die Transition-Width definiert. Diese legt fest, wie steil ein Filter seine Hülle um das Spektrum des Interesses legt. Im Abschnitt 4.3.3 lässt sich ein Vergleich von Übergangbreiten (Transition-Width) finden. Da es auf Grund von

externen Einflüssen, wie beispielweise den aktuellen Wetterverhältnissen oder auftretende Störsignale, zu einer leichten Verschiebung der Frequenzmitte bzw. der Trägerfrequenz kommen kann, empfiehlt es sich die Übergangsbreite so zu wählen, dass ein gewisser Frequenzversatz toleriert wird. Meist ist es ratsam sich bei diesem Faktor langsam an eine dem Anwender und/oder dem System entsprechende Form anzunähern.

Des Weiteren bietet der Xalting FIR Filter die Option zum Einstellen einer Center Frequency. Dieser Punkt ist lediglich relevant, wenn eine Frequenzdifferenz zur eigentlichen Frequenzmitte, also einem Frequenz-Offset, besteht. Dieser Parameter bietet somit die Möglichkeit den Filter auf eine Zwischenfrequenz einzustellen. Sollte keine Differenz bestehen wird dieser Wert auf 0 belassen.

Zusammenfassend kann ein solcher Filter ein bestimmtes Spektrum detektieren und selektieren und dabei von anderen Signalen zu isolieren, sowie nur dieses Nutzsignal abtasten.

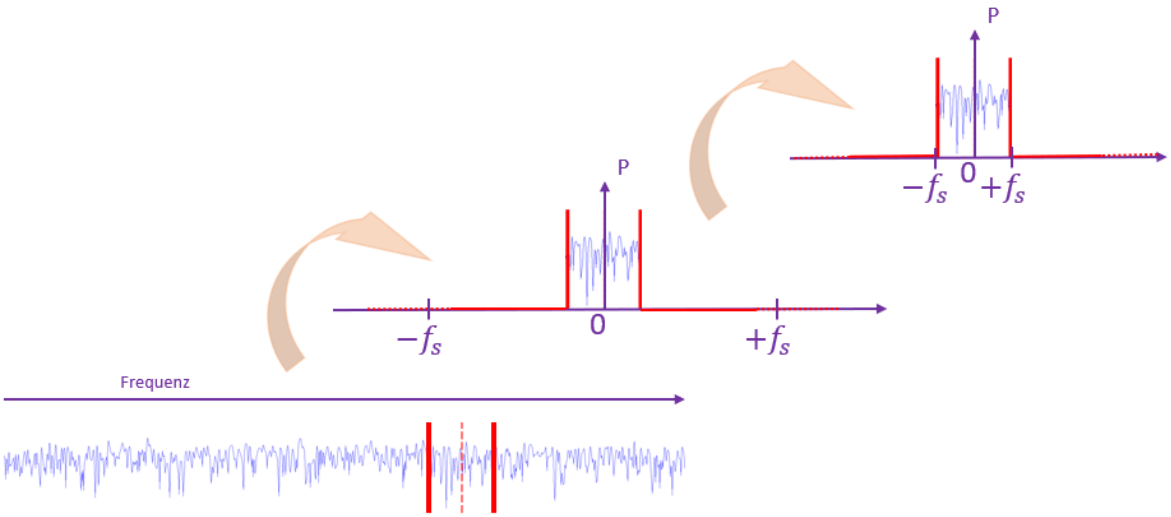


Abbildung 60: allgemeine Wirkung eines Frequency Translation FIR Filters grafisch zusammengefasst (Quelle: Eigene Darstellung)

Die interne Funktionsweise der Filterung eines allgemeinen FIR (Finite Impulse Response) - Filters ist dabei ebenfalls interessant und erwähnenswert. Dabei spielt ein „gleitender Mittelwert“ die Schlüsselrolle in der Realisierung dieser Filterfunktion. „In der Signaltheorie wird der gleitende Durchschnitt als Tiefpassfilter mit endlicher Impulsantwort (FIR-Tiefpass) beschrieben“. [23] Die Abbildungen 61 und 62 sollen das Prinzip des gleitenden Durchschnitts applizieren und verdeutlichen. In der Schaltung ist die Rechenoperation zu sehen. Dazu wird eine bestimmte Anzahl an Werten (Samples) aufgenommen, addiert und diese durch die Anzahl der Werte geteilt. Der somit entstehende Mittelwert ist gleitend, da die aufzunehmenden Samples fortlaufend und somit auch im Rechensystem weiter operierend sind.

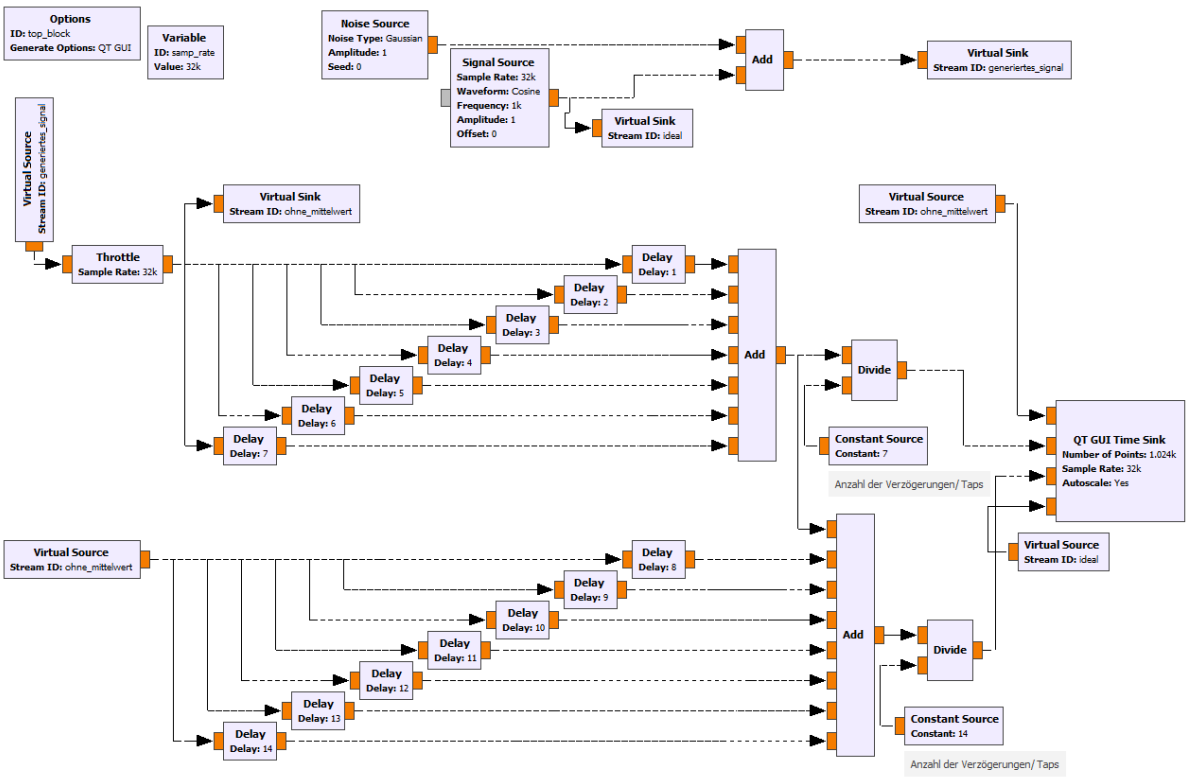


Abbildung 61: GRC-Flowgraph für gleitenden Mittelwert mit unterschiedlicher Breite
(Quelle: Eigene Darstellung)

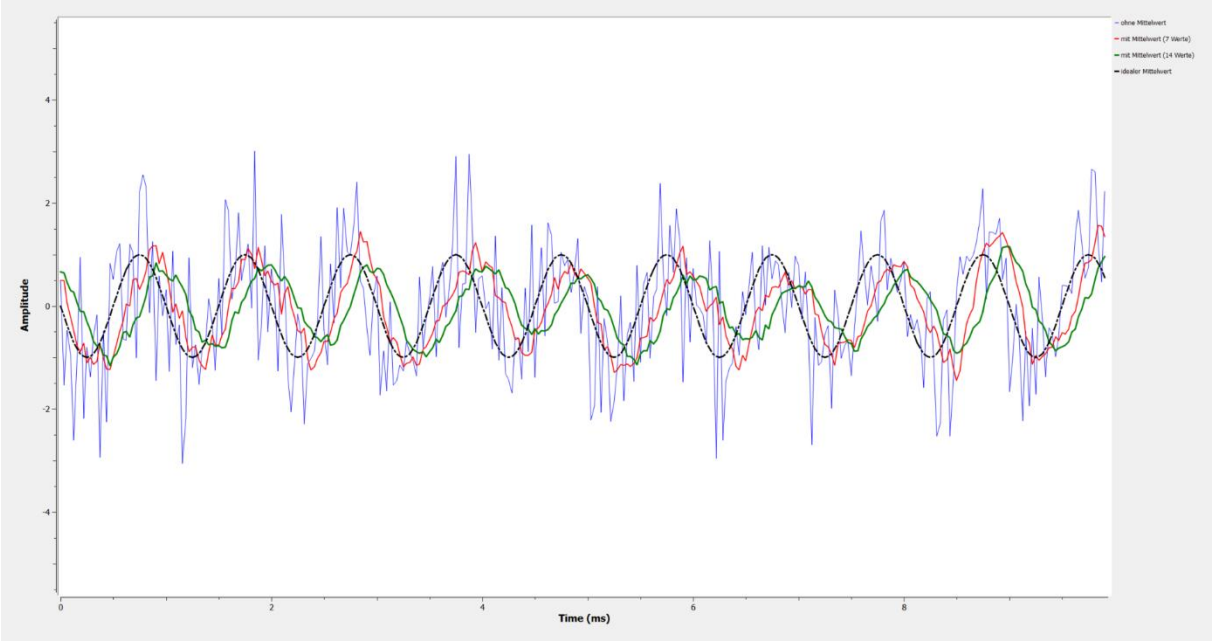


Abbildung 62: Ausführung der unter Abb. 61 gezeigten Schaltung
(Quelle: Eigene Darstellung)

Die hier vorgenommene Staffelung soll die Wirkung der Anzahl der untersuchten Werte – der Breite der Mittelwertoperation – zeigen. Als Signalquelle wurde eine Sinusschwingung addiert mit einer Rauschquelle. Das somit modulierte Signal weist deutliche Amplituden-

schwankungen auf. Je mehr Taps bei der Rechenoperation verwendet werden, desto glatter wird das Signal. Schwankungen werden herausgefiltert. Im Zeitbereich sind in Abbildung 62 diverse Graphen zu erkennen. Die dabei blau dargestellte Linie zeigt das modulierte Signal, welches es zu glätten gilt. Die Glättung soll dabei bedeuten unerwünschte Signalanteile herauszufiltern. Die rote Linie entsteht bei einer Rechenoperation des Mittelwerts unter der Verwendung von 7 Werten und die grüne unter Verwendung von 14 Werten. Die schwarze Linie ist ein direktabgriff der Sinusquelle und soll das ideal geglättete Signal symbolisieren. Je höher die Anzahl der Abgriffe ist, desto höher ist also die Annäherung an einen idealen Filterzustand. Dies bedingt demnach allerdings mehr Rechenzeit. Mathematisch kann dieses System durch folgende Gleichung ausgedrückt werden.

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x\,[n-k]$$

Formel 5: Beschreibung eines allgemeinen Mittelwertfilters

Das Eingangssignal sei dabei x in der diskreten Form x[n], das Ausgangssignal y[n], N die Anzahl der Taps, n der diskrete Wert und k ein Offset (hier auf 0).

Betrachtet man nun die realistische Anwendung des FIR-Filters oder eines generellen Frequenzfilters, bewegt man sich mit der immanenten Funktion im Frequenzbereich. Dabei werden ebenfalls Ausschläge in einem speziellen Spektrum durchgelassen oder geblockt. Bei der Betrachtung des Graphen fällt auf, dass auch die Frequenzanteile des Rauschsignal-anteils der modulierten Schwingung deutlich höher sind als die der reinen Sinusschwingung. Die durchgelassenen Frequenzen hängen also dabei von der Anzahl der verwendeten Abgriffe/Taps/Werte für die Mittelwertbildung ab. Dieses Verhalten stellt einen Tiefpassfilter dar, da die höheren Frequenzanteile rausgefiltert bzw. unterdrückt und die niedrigeren Anteile ungedämpft durchgelassen werden. Generell wandelt solch ein Filter also das Eingangssignal in ein vom Filter abhängiges Ausgangssignal um. Mathematische Grundlage eines FIR-Filters ist es, in dieser Mittelwertgleichung einem jeden verwendeten Eingangswert einen eigenen Koeffizienten zukommen (multipliziert) zu lassen und somit die allgemeine Filter-Gleichung zu bilden.

$$y[n] = \sum_{k=0}^{M} b_k x\,[n-k]$$

Formel 6: modifizierte Mittelwertfiltergleichung (Hinzufügen von Filterkoeffizienten) und die daraus resultierende allgemeine Filtergleichung eines FIR-Filters

b_k stellt dabei den bereits erwähnten hinzugefügten Filterkoeffizienten dar. Unter M versteht man im Allgemeinen die Ordnung des Filters, also die Anzahl der Werte, welche für die

Berechnung eingehen. Ein weiterer wesentlicher Faktor ist, neben der Anzahl der verwendeten Taps, die Parametrierung der Filterkoeffizienten.

Die Eigenschaft eines Filters lässt sich nach der Koeffizientenwahl durch seine Antwort auf einen Einheitsstoß zurückführen bzw. beobachten. Wichtiges Merkmal dieses Impulses ist

$$\delta_0[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases},$$

Formel 7: Definition des Einheitsimpulses für diskrete Systeme

also nur bei der Initialisierung einen Stoß in das System auszusenden und anschließend auf 0 zu verbleiben. Somit zeigt sich die Reaktion auf diesen einmaligen „Stupser“. Dabei wird also anstatt einer Folge von Eingangswerten (x[n]), der Einheitsimpuls δ[n] eingesetzt. Somit lassen sich rückführend die Filterkoeffizienten, welche maßgeblich bildend für das Filterverhalten sind, herauslesen. Aus diesem Verhalten ergibt sich auch der Name „Finite Impulse Response“, da die Anzahl der Filterkoeffizienten endlich ist und somit auch die Dauer der Impulsantwort. [9] [24]

4.4.4 Squelch-Arten (Threshold) in der Praxis

Nachdem das aufgefangene Spektrum nun so weit bearbeitet wurde, dass nur noch das informativ relevante Frequenzspektrum besteht, wird eine zusätzliche Selektion vorgenommen. Allerdings nicht auf der Frequenzachse, sondern auf der Leistungsseite. Abschnitt D im Bild 58 zeigt einen Simple-Squelch-Block und eine Rangevariable. Generell bewirkt der Simple Squelch eine Minderung des Durchlasses an Informationen mit einer bewusst eingestellten Grenze. Die Grenze wird als „Threshold“ bezeichnet und in der Einheit dB (Dezibel) angegeben. Da in dem im Vorfeld ausgewählten Frequenzspektrum diverse weitere und unerwünschte Schwingungen aufgenommen werden können, die allgemein als Rauschen bezeichnet werden, muss eine Unterscheidung zu dem eigentlich zu gewinnen wollenden Informationsgehalt erkannt werden. Dabei handelt es sich in den meisten Fällen um die Signalleistung. Ein konkretes Signal, vor allen wenn es bewusst ausgesendet werden kann und sich die Quelle nahe der Senke (zu empfangendes SDR-Gerät) befindet, lässt sich ein deutlicher Unterschied zwischen relevanten Nutzsignalen und diversen weiteren Schwingungen feststellen. Auch eine Struktur ist im Rauschen kaum bis garnicht zu erkennen, verglichen mit informationshaltigen Signalen. Um diese unerwünschten Anteile, auf Grundlage ihrer Signalstärke, filtern zu können, wird hier ein Simple-Squelch-Block verwendet. Die einstellbare dB-Grenze, unter dem Parameter Threshold, bewirkt ein Abschneiden der Frequenzen, welche unter diesem Leistungspegel liegen. Der Parameter „Alpha“ bietet dem Block einen internen Verstärkungsgrad. Für einen gewünschten Effekt, reicht es an dieser

Stelle, den Parameter Alpha auf dem Standartwert von 1 zu belassen und lediglich den Threshold zu variieren. Werden Daten nicht durchgelassen bewirkt der Simple Squelch keinen Aufnahmestopp des weiterenführenden Streams (Output). Bei der späteren Auswertung sind die durchgekommenden Schwingungen, welche mit einer File Sink aufgezeichnet wurden, entsprechend zeitlich von einander getrennt. Zwischen diesen relevanten Informationsgehältern lassen sich ab und an kleinere Signalabschnitte finden. Für das Herantasten an eine ausreichende Thresholdgrenze bietet es sich an den Vergleich zu diversen Resten des Rauschens zu sehen.

Um einen direkten Vergleich zwischen verschiedenen Threshold-Einstellungen zu erzielen, wurde eine kurze Testreihe erstellt und aufgezeichnet. Diese lässt sich im Anhang A finden. Dort zu sehen sind aufgezeichnete und sich wiederholende Signale. Dabei ist je Threshold-Wert einmal eine „Vollansicht“ der Aufzeichnung zu sehen, in der die sich wiederholenden und empfangenden Signale zu erkennen sind, und des Weiteren darunter eine vergrößerte Ansicht eines Signals, um den demodulierten Informationsghehalt zu präsentieren. Beim Vergleich der Aufzeichnungen ist ein deutliches sondieren der eigentlichen Signale erkennbar. Wichtig ist dabei der in den Kommentaren hinterlegte Threshold-Wert, um das herantasten an eine in dieser Situation optimale Einstellung zu verdeutlichen.

Möchte man eine Aufzeichnung lediglich der Signale, ohne zeitliche Auffüllung der „Pausenzeiten“, erzielen, bietet der „Power Squelch“-Block eine Option. Dieser besitzt unter „Gate“ die Auswahl, ob der weiterführende Stream unterbrochen wird oder nicht. Wählt man unter Gate „Yes“, werden lediglich die durchgelassenen Signale in der Aufzeichnung aufgereiht. In der Praxis hat sich herausgestellt, dass bei dem Power Squelch, neben dem Threshold-Wert, der Alpha-Wert eine wichtige Rolle spielt. Dieser kann variiert werden um ein konkretes Signal vom nachfolgenden Rauschen abzuschneiden. Bei richtiger Einstellung kann es so gelingen ausschließlich die gewollten Nutzsignale aufzuzeichnen.

Die Wirkung, im direkten Vergleich, von einen Simple Squelch zu einen Power Squelch lässt sich in den folgenden Abbildungen finden.

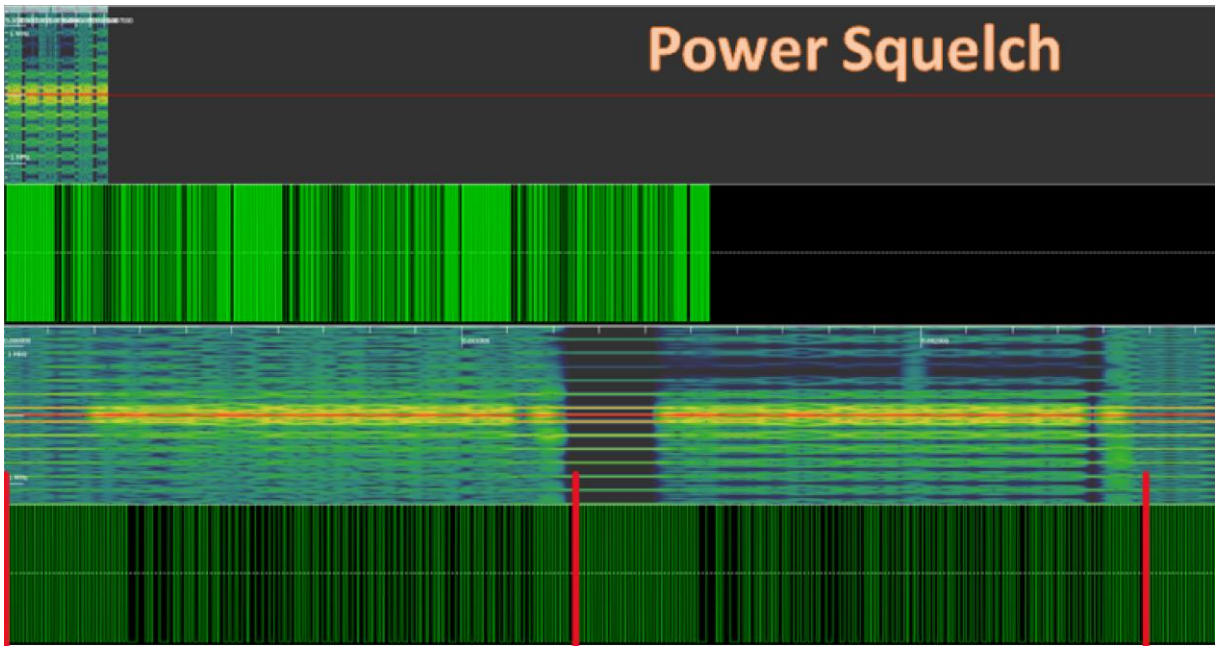


Abbildung 63: Beispiel in Inspectrum für eine mit einem Power Squelch aufgenommene Datei (Quelle: Eigene Darstellung)

Abbildung 63 zeigt in einer Collage den mit einem Power Squelch aufgenommenen und gespeicherten Inhalt einer File-Sink-Datei (Typ Float). In der oberen hälfte ist die Vollansicht der Datei und ein zugehöriger Amplitudenplot zu sehen. Es wurden mehrere Signale mit gleichen Informationsgehalt aufgenommen. Diese wurden ohne Leerstellen chronologisch aufgereiht. Für eine bessere Übersicht, wurde der untere Bildteil ergänzt, welcher einer vergrößerten Darstellung des Oberen entspricht. Die nachträglich hinzugefügten senkrechten roten „Trennlinien“ sollen verdeutlichen, wo ein wM-Bus-Frame aufhört und der nächste anfängt.

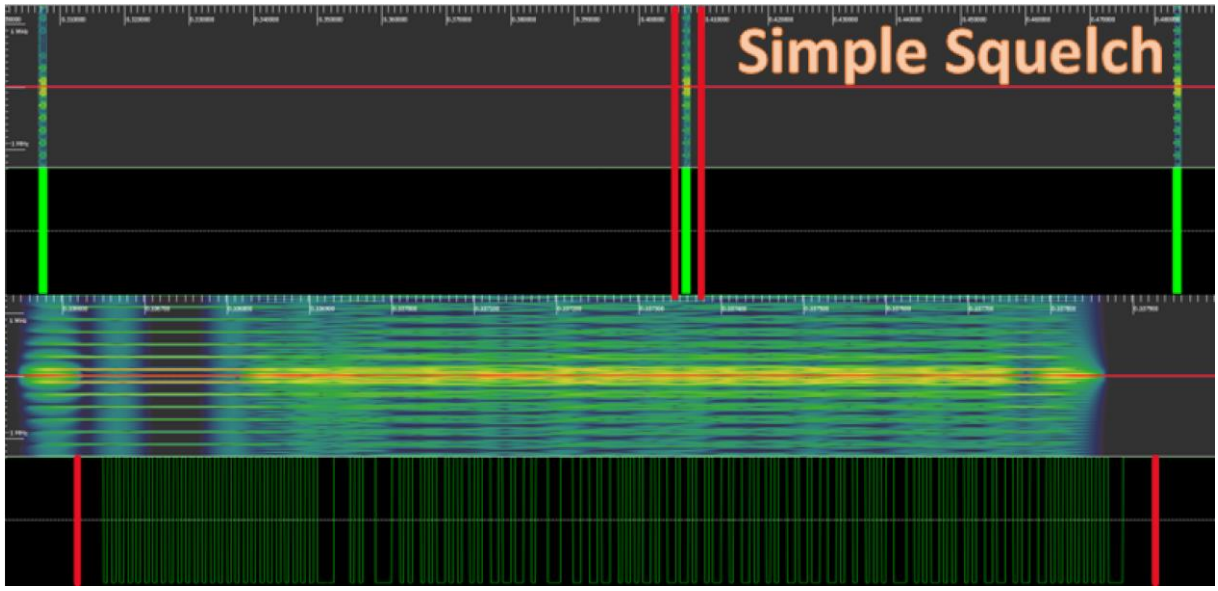


Abbildung 64: Beispiel in Inspectrum für eine mit einem Simple Squelch aufgenommene Datei (Quelle: Eigene Darstellung)

Abbildung 64 wurde zum Vergleich, als Auswertung mit den Simple Squelch, hinzugezogen. Deutlich erkennbar sind die im oberen Bildabschnitt schwarzen Bereiche. Diese Abschnitte sind die bereits erwähnten zeitlichen Leerstellen, welche für eine optische aber auch informative Trennung der durchgelassenen bzw. erreichten Signale dienen. Bei ungeigneter Threshold-Einstellung, würden diese Areale durch unerwünschte Rauschsignale ersetzt werden. Im oberen Bereich sind 3 Signale zu sehen, mit einem darunterliegenden Amplitudenplot. Im unteren Bereich des Bildes ist der oben mit senkrechten roten „Trennlinien“ markierte Bereich abgebildet. Dieser zeigt einen alleinstehenden, also keinen direkt von anderen Informationsgehältern umringten, wM-Bus-Frame.

Ein unerwünschtes Signalrauschen wurde in beiden gezeigten Varianten vollständig ausgeblendet/ unterdrückt, wie deutlich zu erkennen ist.

Ein weiterer, gegebenfalls relevanter, Punkt ist die entstehende Größe der Datei. Da mit einem Power Squelch wesentlich weniger Schwingungen zur weiteren Verarbeitung durchgelassen werden, wirkt sich diese Selektion erheblich auf die anfallende Datenmenge aus.

Beispiel aus einem Versuch:

Eine ca. 5 sekündige Aufzeichnung von Signalen mit einem Informationsgehalt von A, bei einer Sample-Rate von 2,6MSps und unter der Verwendung eines Simple Squelch, bedingte eine Dateigröße von gerundet 50MB.

Eine ca. 5 sekündige Aufzeichnung von Signalen mit einem Informationsgehalt von A, bei einer Sample-Rate von 2,6MSps und unter der Verwendung eines Power Squelch, mit aktivierter Gate-Funktion, bedingte eine Dateigröße von gerundet 0,2MB.

4.4.5 Angewendete Quadrature Demodulation

Nachdem die empfangenden Informationen „drei mal gefiltert wurden“ (1. Antenne als erste Selektion des Frequenzspektrum; 2. eigentlicher Filter für genauere Eingrenzung des Spektrums, ggf. Optimierung der Sample-Rate, ggf. Anti-Aliasing; 3. festlegen einer Signalleistungsgrenze (Threshold) als Durchlasskriterium), gelangt der Datenstream, wie in Abbildung 58 zu sehen, zum Funktionsabschnitt E – dem „Quadrature Demod“-Block. Um den hier ankommenden IQ-Datenstream in eine binäre und für die weitere Verarbeitung sinnvolle Form umzuwandeln, werden die separat aufgenommenen Real- und Imaginärwerte des Signals in dieser Funktionseinheit zusammengeführt. Dabei ist der Quadrature-Demod-Block eine generell von GNU Radio empfohlene Operationseinheit, wenn es um das Thema der FM-Demodulation geht. Dabei wird das Kernstück eines Software Defined Radios verarbeitet – das IQ-Signal.

Nachdem mit Hilfe des auf dem SDR-Gerät befindlichen lokalen Oszillator ein Frequenzspektrum in das Basisband verschoben wurde, also die Mitte seiner Bandbreite auf 0Hz liegt, muss für die negative Seite des Spektrums eine Lösung zum Erhalt gefunden werden. In der Natur gibt es keine negativen Frequenzen.

$$f_{RF} - f_{L0} = |f_{ZF}|$$

Formel 8: Entstehung einer Zwischenfrequenz aus der Beziehung von Lokaloszillatorfrequenz und der Frequenz des zu empfangenen Signals (Radio Frequency)

Die hier gezeigte Formel soll veranschaulichen, dass eine entstehende Zwischenfrequenz (f_{ZF}) abhängig von der eingestellten und dem System zugeführten Frequenz des lokalen Oszillators (f_{L0}) ist. Entspricht dabei die Trägerfrequenz des Signals (Radio Frequncy f_{RF}), welches von der Antenne aufgenommen wird, der Frequenz des lok. Oszi., dann ist die Zwischenfrequenz (f_{ZF}) gleich 0. Befindet sich ein Signal in diesem Bereich, spricht man vom Basisband (Frequenzmitte hier als f_{BB} bezeichnet). Auf Grund der Bandbreite des Signals, muss somit ein Teil im negativen Spektrum angesiedelt sein, wie in Abbildung 65 dargestellt.

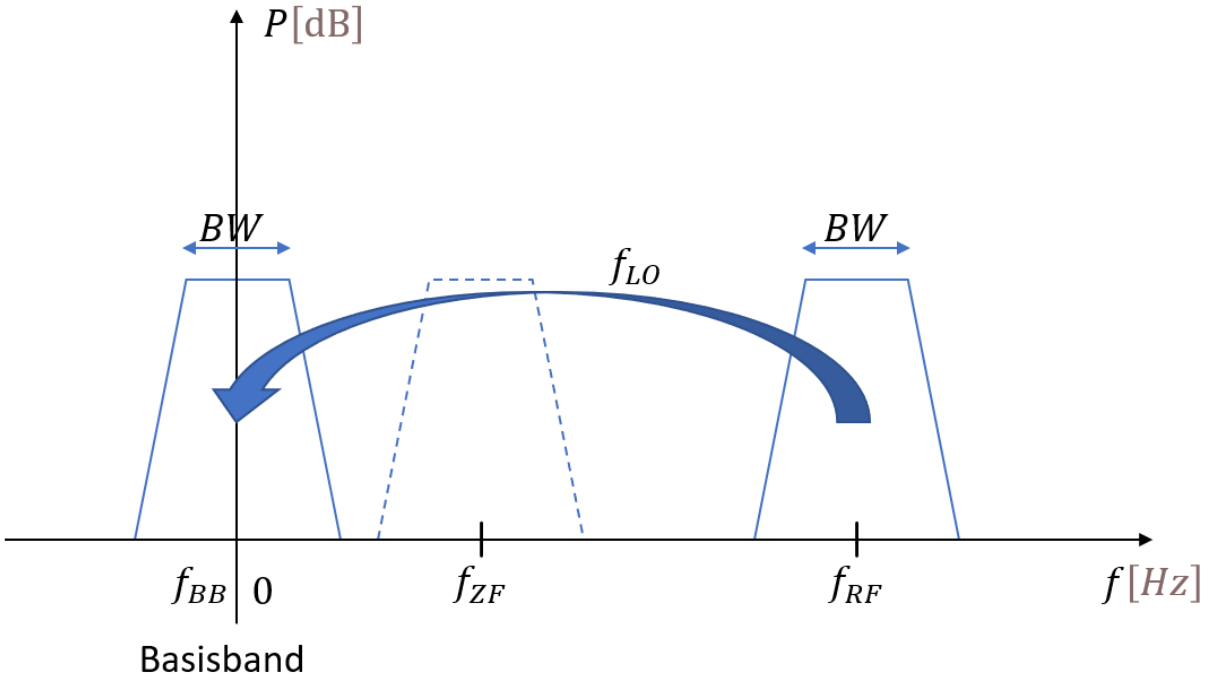


Abbildung 65: Signalverschiebung ins Basisband
(Quelle: Eigene Darstellung)

Würde man sich mit diesem Prinzip in der rein analogen Welt bewegen, ginge die Hälfte, also der negative Anteil des Spektrums, verloren.

Aus diesem Grund wurde das IQ-Signal in Verbindung mit dem Quadrature Demodulator eingeführt und findet hier Anwendung um dieses Problem zu lösen.

Dafür wird, wie bereits im Theorieteil unter Kapitel 2.3.3 erwähnt, ein Signal quadraturisiert. In diesem Kontext werden dabei aus einem Signal zwei Schwingungen erzeugt, welche im Bild 66 beispielhaft zu sehen sind.

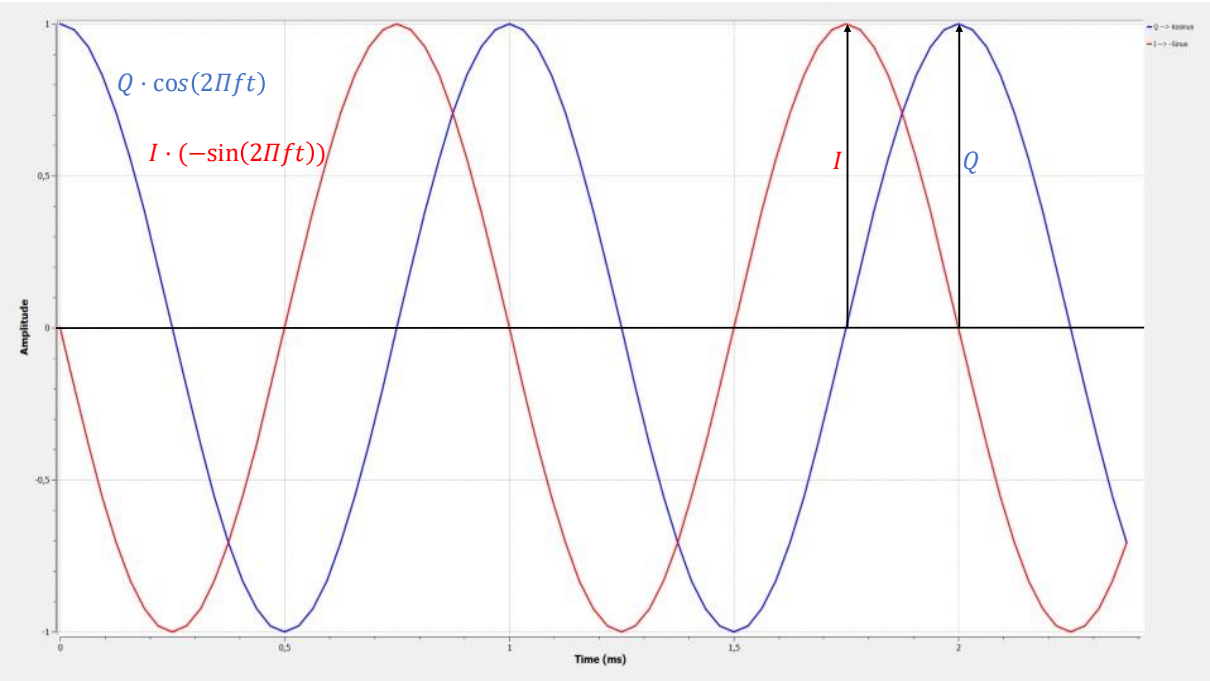


Abbildung 66: I- und Q-Anteil eines Signals
(Quelle: Eigene Darstellung)

Dabei wird ein Signal mit einem lokalen Oszillator vermischt und gesplittet, wobei eines davon um 90° Phasenversetzt wird. In den meisten Beschreibungen die während der Recherche zu dieser Arbeit gefunden wurden, wurde die Kombination aus einer Kosinusschwingung und einer negativen Sinusschwingung verwendet. „Grund für diese Verwendung anstatt der Sinus-Kosinus-Kombination ist, dass bei der Verwendung des Kosinus sich der Gleichanteil als Frequenz Null beschreiben lässt.“ [25]

$$Q \cdot cos(2\pi ft) = Q \cdot cos(0t) = Q \cdot 1$$

Formel 9: Verhalten des Kosinus bei einer Frequenz von 0Hz (Gleichanteil)

Das Blockschaltbild in Abbildung 67 soll das Prinzip der Aufteilung des emfpangenen Signals verdeutlichen und stellt dabei das Grundprinzip eines Software Defined Radios nach dem IQ-Verfahren dar.

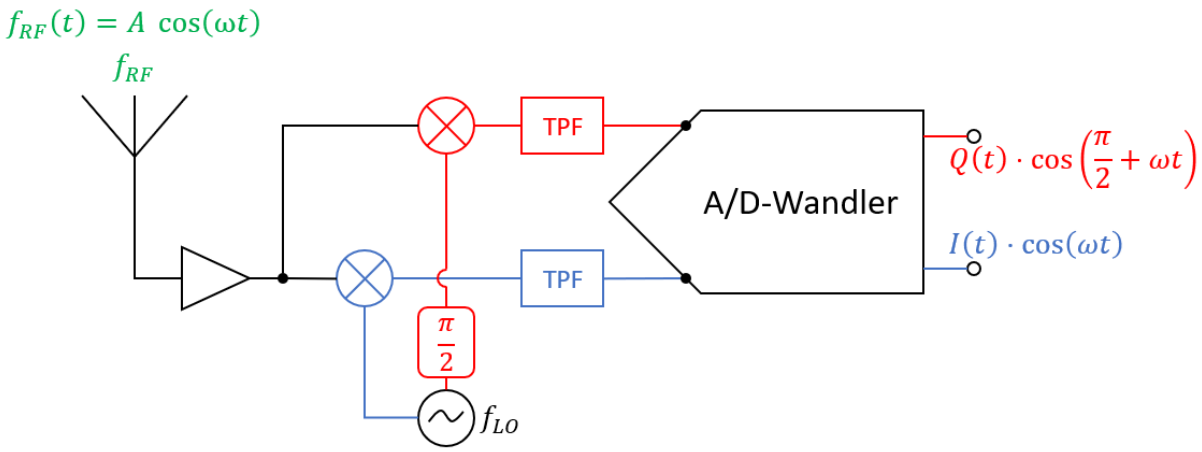


Abbildung 67: Grundschaltbild eines SDR nach IQ-Verfahren
(Quelle: Eigene Darstellung)

Auf Grund dieser Gegebenheit erhält man 2 Wellenformen, welche, durch den 90°-Phasenversatz, einer z.B. Sinus- und einer Kosinusschwingung oder Kosinus- und Minus-Sinuns-Schwingung entsprechen. Diese besitzen wiederum jeweils eine eigene Amplitude (hier als I und Q bezeichnet). Die elektromagnetische Schwingung f_{RF} wird von der Antenne aufgenommen und anschließend durch einen Amplifier verstärkt. Anschließend wird die Welle zu 2 Mischern geführt. Der hier blau dargestellte Part steht für den In-Phase-Anteil. Dieser besitzt keine zusätzliche Phasenverschieben. Es wird hier lediglich das ankommende Signal mit der Frequenz des lokalen Oszillators f_{L0} gemischt, um das Signal in das Basisband zu verschieben. Da auf Grund von beispielsweise thermischen Änderungen die Frequenz f_{L0} einen leichten Drift aufweisen könnte, wird dieses bereits Hardwareseitig berücksichtigt und ausgeglichen, sodass dieser Problematik, bei Umgebungsverhältnissen für die das SDR-Gerät ausgelegt ist, keine weitere Beachtung geschenkt werden muss. Der hier rot dargestellte Part stellt den Quadratur-Anteil dar. Wie auch bei dem In-Phase-Part wird hier das Signal f_{RF} mit f_{L0} gemischt, um das Basisband zu erreichen, doch zusätzlich wird eine Phasenverschiebung aufgezwungen, welche 90° beträgt.

$$I(t) \cdot cos(\omega t)$$

$$Q(t) \cdot cos\left(\frac{\pi}{2} + \omega t\right) = Q(t) \cdot (-sin(\omega t))$$

Formel 10: resultierende Signalteile nach der Mischung mit dem lokal. Oszi. und dem maßgeblich entscheidenden 90°-Phasenversatz

Da Aufgrund der Mischung/Verschiebung zusätzliche Mischprodukte mit einer vielfachen Frequenz entstehen, werden Tiefpassfilter eingesetzt, um die unerwünschten Anteile

herauszufiltern und somit rein das gewollte I(t)- oder Q(t)-Signal zu erhalten. Die Tiefpassfilter sind in Abbildung 67 mit TPF bezeichnet dargestellt.

Stellt man die beiden Amplituden in der IQ-Ebene dar, so lässt sich aus dem resultierenden Schnittpunkt und dem daraus hervorgehenden Zeiger eine Phaseninformation ablesen.

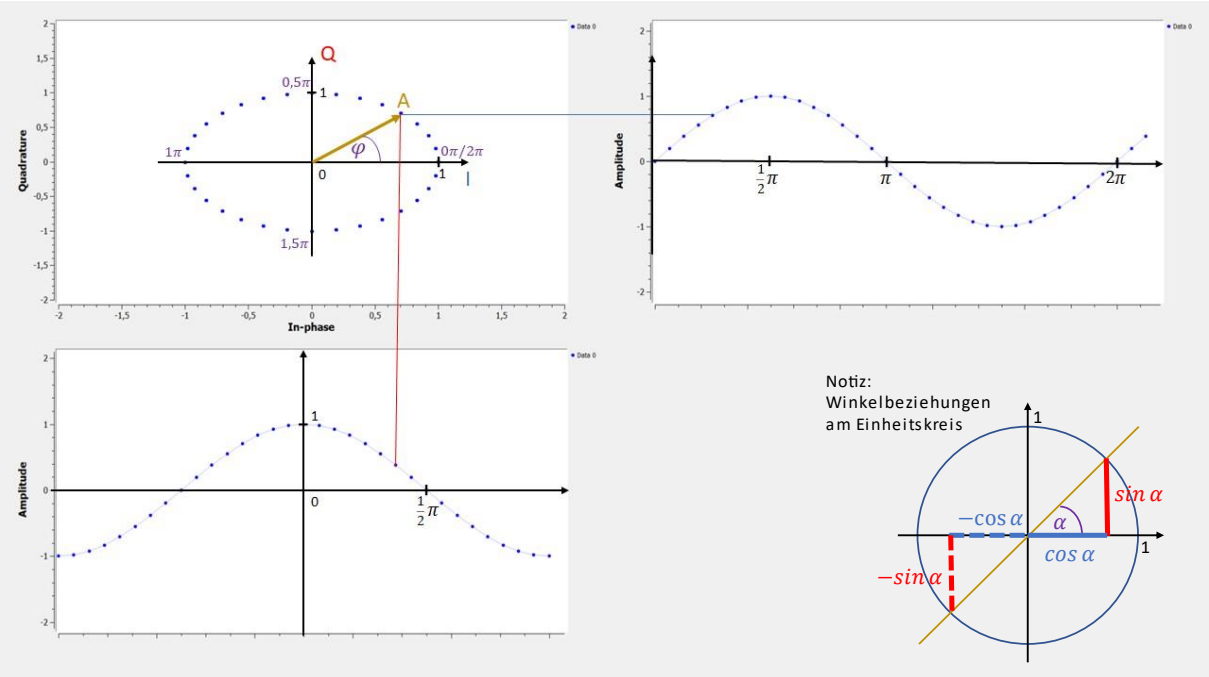


Abbildung 68: Erläuterung der Winkelbeziehungen und der Phasenlage in der IQ-Ebene (Quelle: Eigene Darstellung)

Die Amplitude des Signals f_{RF} wird hier mit A dargestellt und lässt sich mit der Formel

$$A(t) = \sqrt{I(t)^2 + Q(t)^2}$$

Formel 11: sich aus einem Wertepaar des IQ-Stroms ergebene Amplitude

ausdrücken. Der dabei entstehende Winkel in der komplexen Zahlenebene/ IQ-Ebene wurde hier mit φ bezeichnet. Dieser lässt sich ebenfalls über die Winkelbeziehungen ausdrücken.

$$\varphi(t) = arctan\left(\frac{Q(t)}{I(t)}\right)$$

Formel 12: sich aus einem Wertepaar des IQ-Stroms ergebener Phasenwinkel

Die Zusammenführung dieser beiden Anteile ist ein Kerngedanke der Quadratur Demodulation.

Folgendes Beispiel soll diesen Effekt verdeutlichen:

Werden 2 um 90° versetzte Schwingungen addiert, lässt sich eine Amplituden- und eine Phasenmodulation erzeugen, durch Änderung der Amplituden der beiden Eingangsignale (I und Q).

Dazu wurden ein Sinus- und ein Kosinussignal addiert und jeweils die Amplitude variabel gestaltet, sodass zu dem jeweiligen Standartwert von 1 ein Wert zwischen 0 und 10 hinzugefügt werden kann.

Abbildung 69 zeigt die Ausgangwerte.

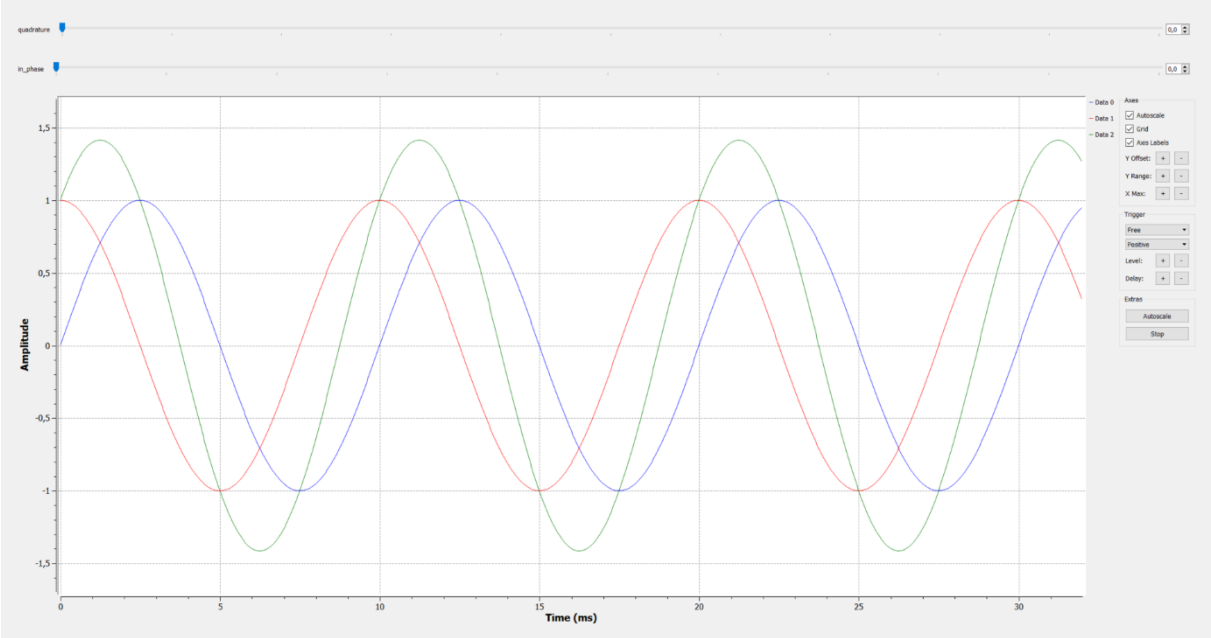


Abbildung 69: Addition zweier zueinander phasenversetzter Signale
(Quelle: Eigene Darstellung)

Rot und blau dargestellt sind das I- und Q-Signal. Das Resultat der Addition ist grün. Dieses ist phasenversetzt, in Abhängigkeit der jeweiligen Amplitudenwerte zum konkreten Zeitpunkt.

Variiert man nun die Amplituden, entsteht ein daraus resultierender Phasenversatz, wie in Abbildung 70 zu sehen ist.

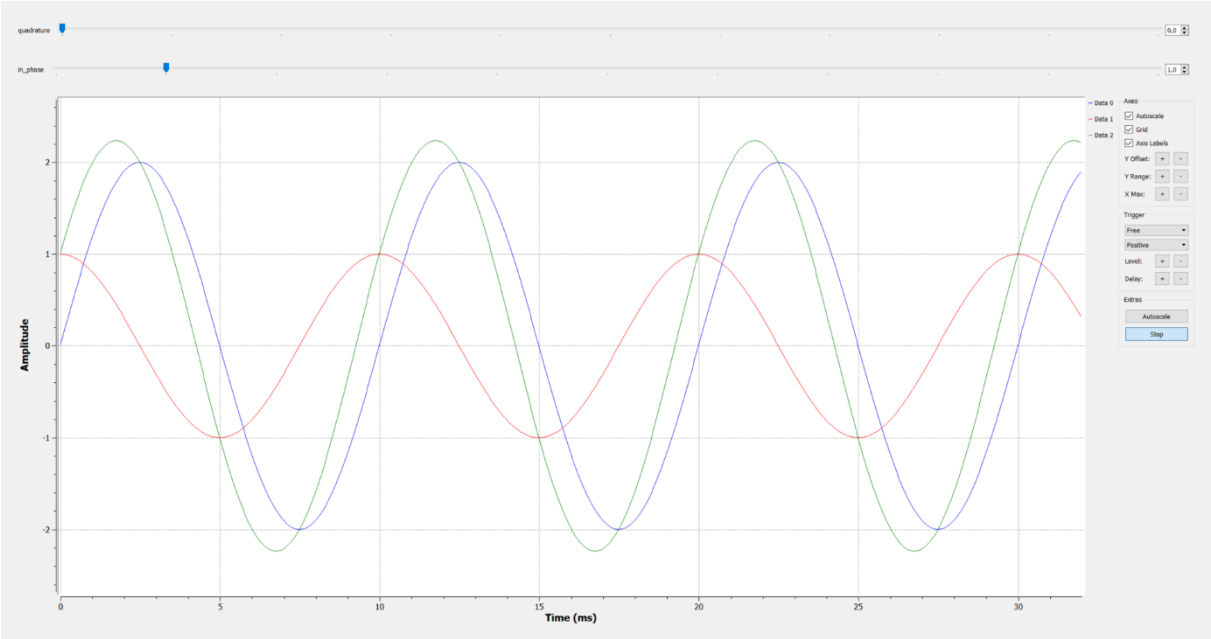


Abbildung 70: Addition zweier phasenversetzter Signale mit Amplitudenvariation und daraus resultierender Phasendifferenz
(Quelle: Eigene Darstellung)

Entsprechend entsteht das im Bild 68 erzeugte Konstellationsdiagramm mit resultierender Amplitude A und resultierenden Phasenwinkel φ.

Bei einer Frequenzmodulation, wie die FSK eine ist, lässt sich ein Effekt ebenfalls in der IQ-Ebene darstellen. Gehen wir davon aus, dass bei einer frequenzabhängigen Modulation die Amplitude, also I und Q und die daraus resultierende Länge des Zeigers für A, nicht verändert wird, erreichen wir eine besonders gute Veranschaulichung über den Phasenwinkel. Abbildung 71 zeigt einen Phasenversatz auf Grund einer Frequenzdifferenz zwischen der roten und der blauen Schwingung.

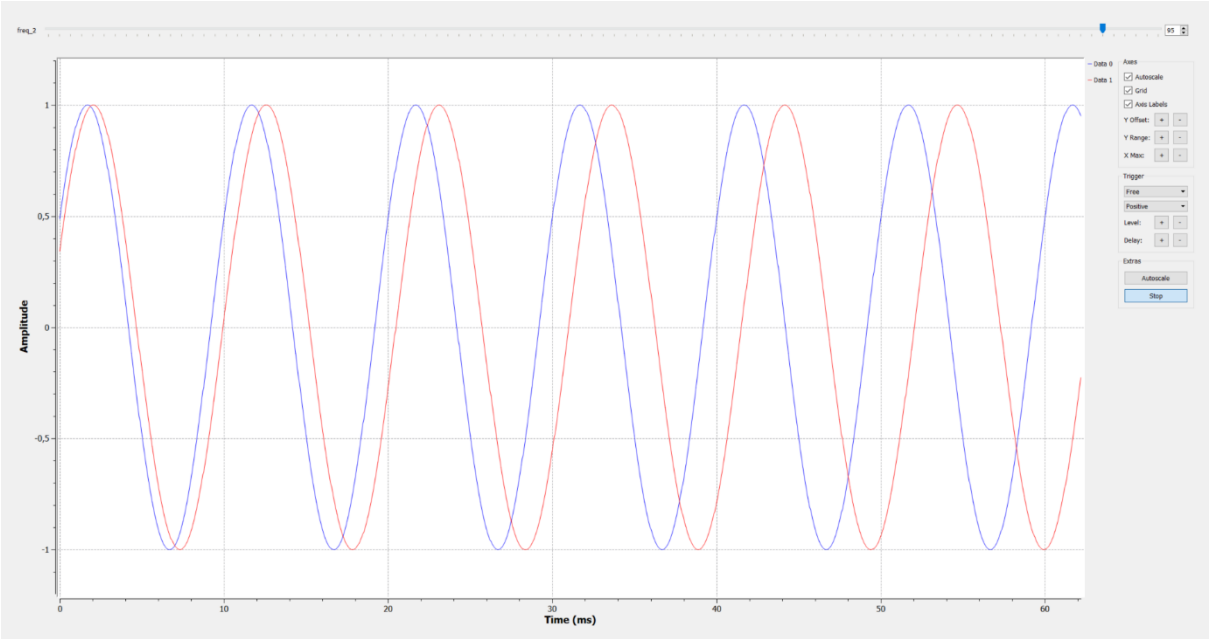


Abbildung 71: Phasenversatz zweier Signale aufgrund einer Frequenzdifferenz
(Quelle: Eigene Darstellung)

Betrachtet man nun zusätzlich Abbildung 68, lässt sich das Prinzip des abhängigen Phasenwinkels verdeutlichen. Die Amplituden I und Q sind gleich und über die Zeit konstant. Die Drehgeschwindigkeit des Phasenwinkels φ ist dabei Indikation der Frequenzdifferenz.

Neben der Geschwindigkeit des Phasenwinkels lässt sich daher ein weiterer wichtiger Punkt feststellen. Die Differenz der beiden Frequenzen ist nur ein Faktor. Welche Frequenz des IQ-Signals oder auch die des empfangenden Signals schneller ist, gibt dabei die Drehrichtung des Phasenwinkels an. Wäre die modulierte Trägerfrequenz oberhalb der Frequenz des lokalen Oszillator, würde sich das Signal in eine positive Richtung drehen (gegen den Uhrzeigersinn). Bei entgegengesetzter Konstellation entsprechend anders herum.

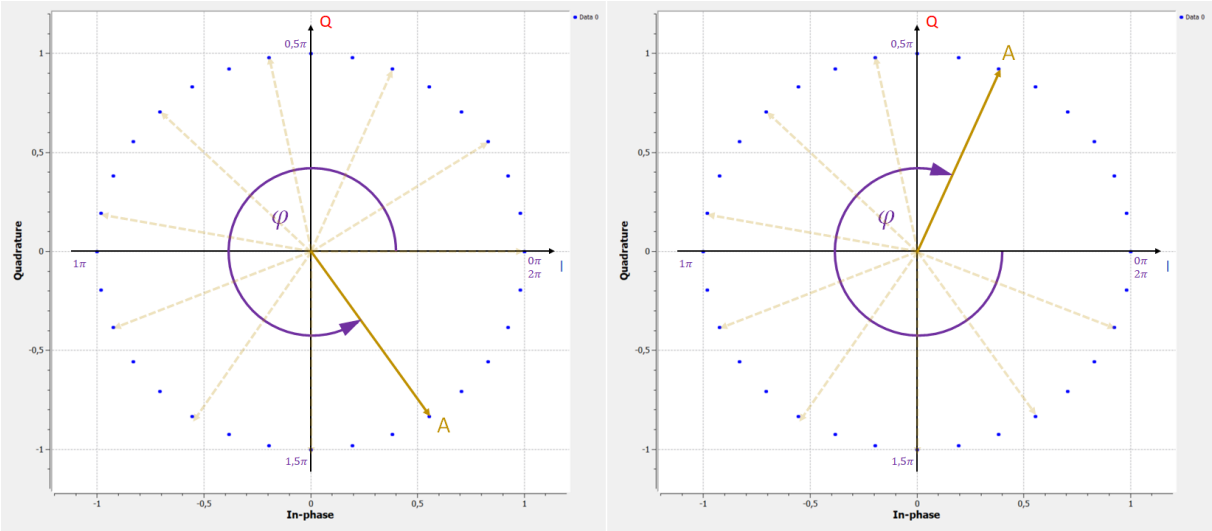


Abbildung 72: aus Frequenzdifferenz zum Referenzwert resultierende Drehrichtungsänderung der Phasenlage (Quelle: Eigene Darstellung)

Und das ist das Kernprinzip des Quadratur Demodulators. Die Detektion des Informationsgehalts eines Basisbandsignals in Abhängigkeit der aufmodulierten Frequenzdifferenz zur ursprünglichen Frequenzmitte, welche der Frequenz des lokalen Oszillators entspricht, über die Drehrichtung des Phasenwinkels.

Beispielsweise mathematisch darstellbar als:

Für $f_{RF} > f_{L0}$ (also bei FSK einer aufmodulierten 1 entsprechend):

$$A(t) = IQ(t) = cos(2\pi ft) + j \cdot sin(2\pi ft) = e^{j2\pi ft}$$

Für $f_{RF} < f_{L0}$ (also bei FSK einer aufmodulierten 0 entsprechend):

$$A(t) = IQ(t) = cos(2\pi ft) + j \cdot (-sin(2\pi ft)) = e^{-j2\pi ft}$$

Formel 13: Drehrichtungen des Zeigers in der IQ-Ebene als Ergebnis der in das Basisband verschobenen FSK-Modulation (grafischer Bezug in Abb. 72)

Um das Prinzip zu verdeutlichen, wird in Abbildung 73 die vorgehensweise eines dem FSK verwandten Modulationsprinzips, das MSK (Minimum Shift Keying), im Bezug des Datenflusses zur IQ-Ebene, wie auch in Abbildung 72, hier mit $\Delta\varphi = \frac{\pi}{2}$ dargestellt.

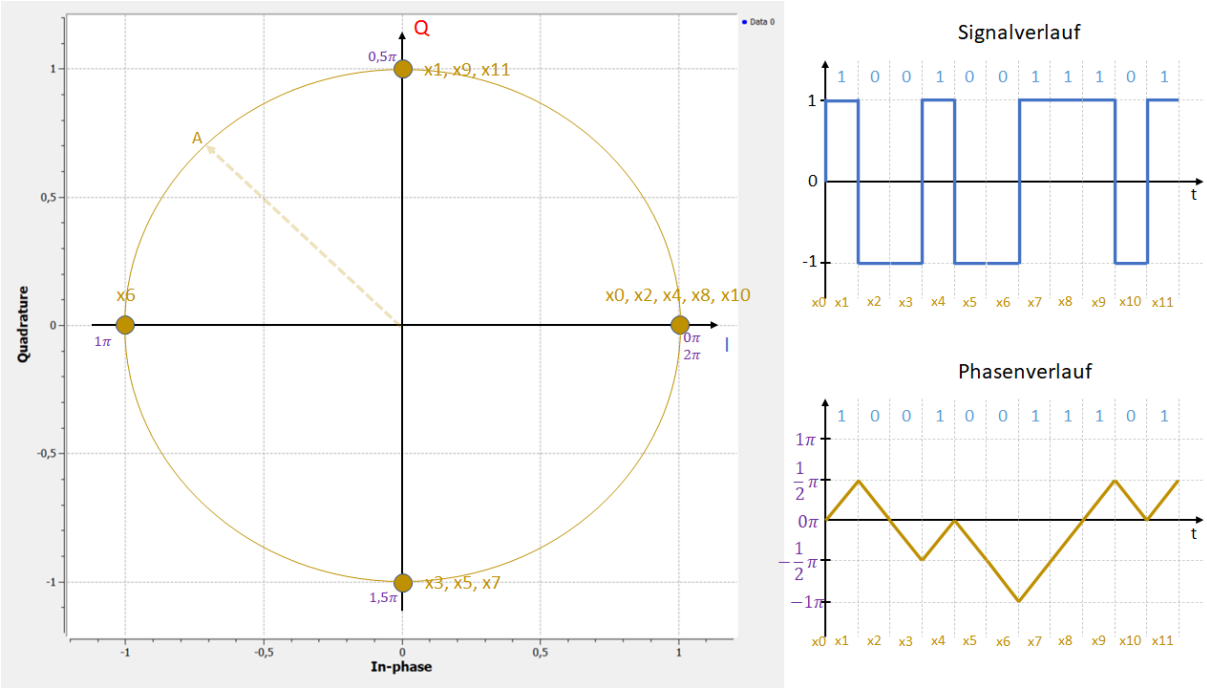


Abbildung 73: Veranschaulichung des Zusammenhangs zwischen Signal- und Phasenverlauf und der daraus resultierenden Drehrichtungsänderung anhand eines MSK-Beispiels mit striktem Pi/2-Phasenversatz (Quelle: Eigene Darstellung)

Auf der offiziellen GNU-Radio-Wiki-Seite wird die Funktionsweise dieses Blocks, hinsichtlich konkreter Sample-Werte, beschrieben. Dabei heißt es, dass hierbei das Produkt eines um einen Abtastwert verzögerten und konjugierten Wertes des Eingangs und der Wert des nicht verzögerten Signals berechnet wird. Zusätzlich wird das Argument, also der Winkel der resultierenden komplexen Zahl aus diesen beiden Werten, betrachtet und kalkuliert. Auf der Seite wird diese beziehung als

$$y[n] = arg(x[n] \cdot \bar{x}[n-1])$$

Formel 14: Ausgangspunkt der Funktionsweise der Ausgangswertbildung des Quadrature-Demod-Blocks

dargestellt. Dabei entsprechen y[n] dem hier im Vorfeld verwendeten A(t), sowie x[n] dem I(t) und $\bar{x}[n-1]$ dem Q(t). Desweiteren wird x als komplexer Sinuid mit einer Amplitude größer 0 (A), einer reellwertigen Frequenz (f) sowie einer Phase (φ_0) zwischen 0 und 2π definiert. f_S steht dabei für die Samplingfrequenz.

$$x[n] = Ae^{j2\pi\left(\frac{f}{f_S}n+\varphi_0\right)}$$

Formel 15: Einführung des Eingangssignals des Quadrature-Demod-Blocks als komplexe Form

Anschließend wird diese Beziehung ausführlich und schrittweise umgesetzt.

$$y[n] = arg\left(Ae^{j2\pi\left(\frac{f}{f_s}n+\varphi_0\right)} \cdot \overline{Ae^{j2\Pi\left(\frac{f}{f_s}(n-1)+\varphi_0\right)}}\right)$$

Formel 16: erstes Zusammenführen der in den Formeln 14 und 15 grundlegenden Gegebenheiten

In der ersten Form erkennt man gut das im Vorfeld textlich beschriebene. Genauer gesagt das Produkt aus dem aufgenommenen unverzögerten Wert und dem um eine Stelle versetzten und komplex konjugierten Wert.

$$y[n] = arg\left(A^2e^{j2\pi\left(\frac{f}{f_s}n+\varphi_0\right)} \cdot e^{-j2\Pi\left(\frac{f}{f_s}(n-1)+\varphi_0\right)}\right)$$

Formel 17: Auflösen der aus Formel 16 stammenden komplex konjugierten Schreibweise, sowie ein anfängliches Zusammenfassen

In den weiteren Zeilen wird dieser Ausdruck zusammengefasst und umgestellt, sodass zum Schluss eine möglichst handzahme Form entsteht. Der Bezeichner „Quadratur“ wird in diesem Kontext im allgemeinen definiert durch eine um 90° versetzte Phase des einen Signals (Q - Quadratur) zum Referenzsignal, welches unverändert ist (I – In Phase). Man stelle sich dabei die Seitenlängen- und Winkelberechnung eines rechtwinkligen Dreiecks vor.

$$y[n] = arg\left(A^2e^{j2\pi\left(\frac{f}{f_s}n+\varphi_0-\frac{f}{f_s}(n-1)-\varphi_0\right)}\right)$$

$$y[n] = arg\left(A^2e^{j2\pi\left(\frac{f}{f_s}n-\frac{f}{f_s}(n-1)\right)}\right)$$

$$y[n] = arg\left(A^2e^{j2\pi\left(\frac{f}{f_s}(n-(n-1))\right)}\right)$$

$$y[n] = arg\left(A^2e^{j2\pi\frac{f}{f_s}}\right)$$

Formel 18: zusammengefasste Form des resultierenden Ausgangssignals des Quadrature-Demod-Blocks

Diese entstandene kurze Form beschreibt das Gesamtverhältnis des Resultats in der IQ-Ebene.

Auf der GNU-Radio-Wiki-Seite wird ebenfalls die Amplitude A und die daraus entstehende endgültige Schlüsselform erwähnt. A wird dabei als Real erkannt und somit auch A^2, welches nur skaliert. Dafür ist das Argument $arg\left(A^2e^{j2\pi\frac{f}{f_s}}\right)$ dabei invariant, woraus geschlussfolgert

wird, dass $arg\left(A^2 e^{j2\pi\frac{f}{f_s}}\right) = \frac{f}{f_s}$ ist. Die Kernaussage der Quadratur Demodulation ist demnach also das Verhältnis der Frequenz des modulierten Signals zur Sampling-Rate. Auf Grund der Verschiebung in das Basisband variiert dabei die informationshaltige Schwingung um den Nullpunkt, definiert nach der Frequenzabweichung der FSK, – also von negativ zu positiv und andersrum. Generell kommt es dabei aber auf die Differenz zwischen der Frequenz des lokalen Oszillators und der aktuellen Frequenz des Signals an.

Dabei sieht der Input am Quad-Demod-Block beispielsweise wie in Abbildung 74 aus.

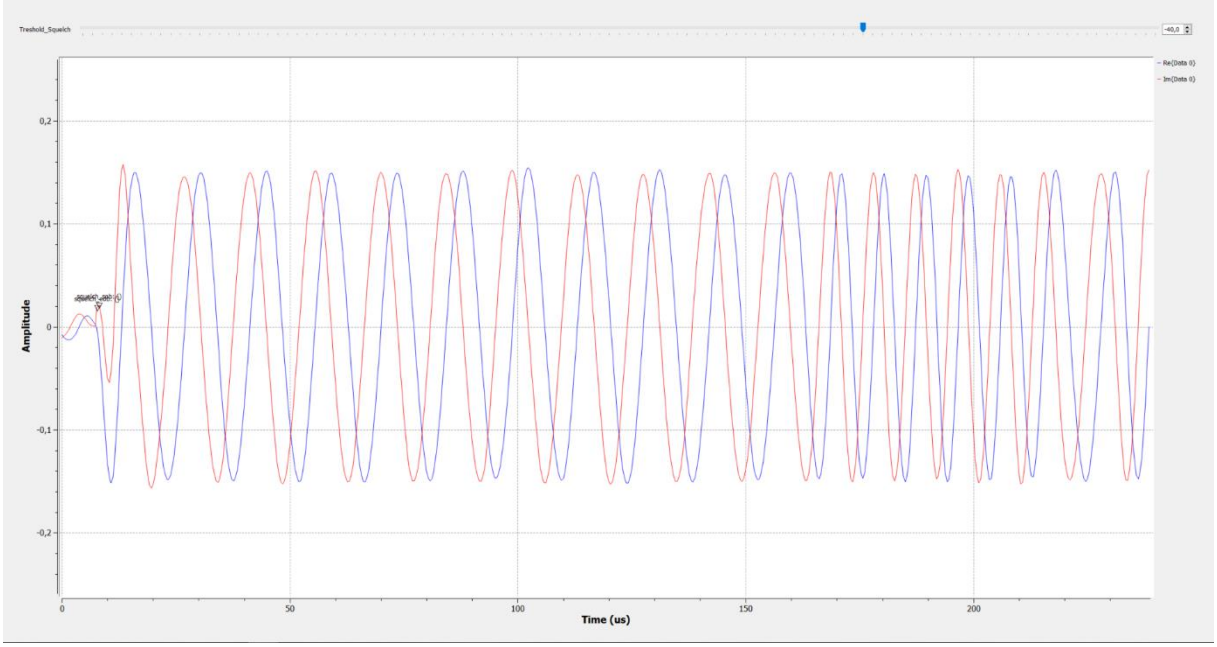


Abbildung 74: Inputsignalbeispiel des Quadrature-Demod-Blocks
(Quelle: Eigene Darstellung)

Die finale Eigenschaft dieser Funktionseinheit ist es, den detektierten frequenzabhängigen Informationsgehalt für die nachfolgende Verarbeitung als Datemstrom aus zusammengefassten (Real- und Imaginärteil, also einer komplexen Zahl) positiven und negativen Werten zur Verfügung zu stellen. Je Abtastpunkt wird dabei entschieden ob der resultierende, also am Output des Blocks anliegende, Pegel auf einen von der eingestellten Verstärkung abhängigen Wert über oder unter Null gezogen wird. Zur Demonstration zeigt Abbildung 75 den Output des Quad-Demod-Blocks.

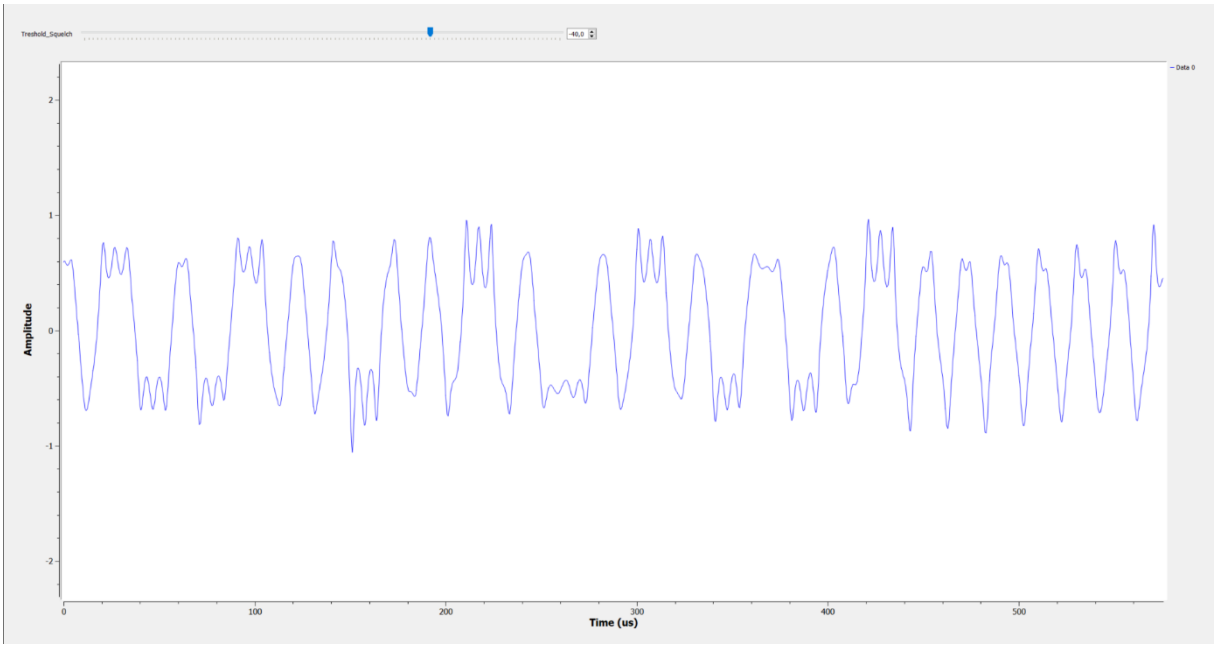


Abbildung 75: Outputsignalbeispiel des Quadrature-Demod-Blocks
(Quelle: Eigene Darstellung)

Das somit im Vorfeld frequenzmodulierte, und aufgrund der Verarbeitungsweise eines SDR-Geräts separat gesplittete, Signal, entspricht nun einem Rechtecksignal einer gepulsten Amplitudenmodulation. Dieses zusammengeführte Signal kann nun zur weiteren Datenverarbeitung an die entsprechenden Stellen geleitet werden. [11] [26] [27] [28]

4.4.6 Diverse Anzeigeelemente

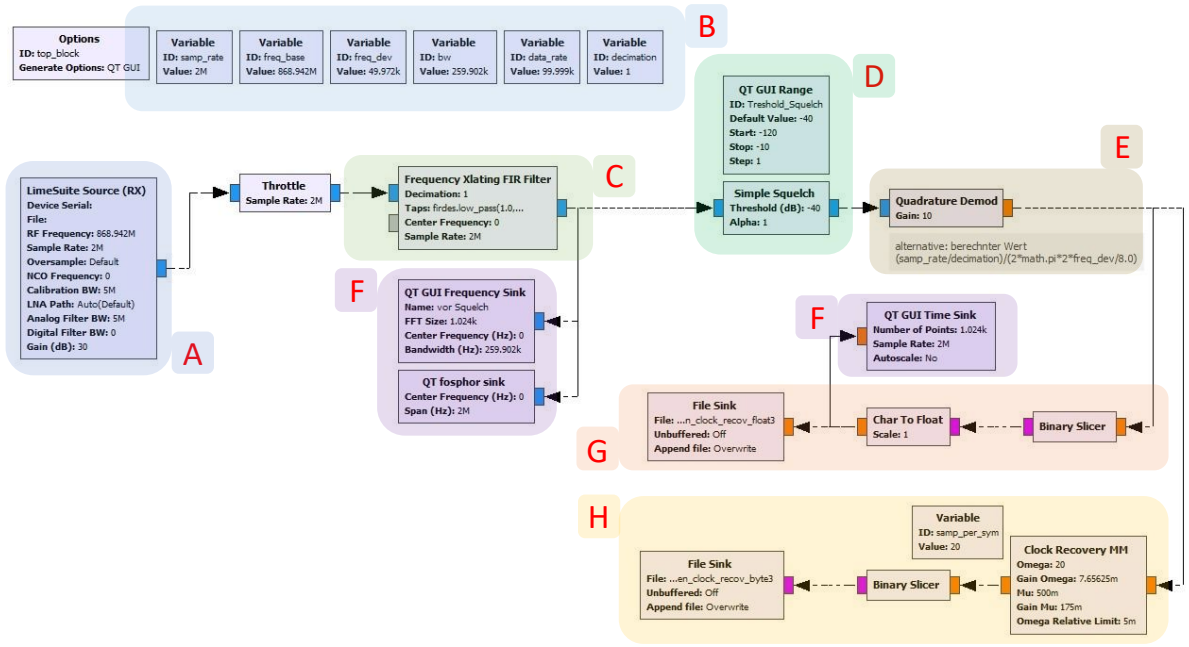


Abbildung 76: segmentierte Darstellung, zur besseren Erklärbarkeit, der in Abb. 50 gezeigten Schaltung (Quelle: Eigene Darstellung)

Die im Bild 76 mit F gekennzeichneten Abschnitte zeigen diverse „Sinks“. Diese letztendlichen Anzeigeelemente sind generell empfehlenswert, um partiell schauen zu können, wie sich bestimmte Werte verhalten. Sowohl bei der Schaltungsentwicklung, als auch bei einer allgemeinen Signalanalyse ist dies von großem Vorteil. In der analogen Welt würde man für solche Fälle, entsprechend ihrer Auslegung und dem Anwendungsfall, beispielsweise ein Oszilloskop einsetzen. Somit lassen sich Signale unter anderen im Frequenz- und Zeitbereich untersuchen, ein spezielles Frequenzspektrum mit Signalverlauf betrachten oder eine Konstellation der I- und Q-Anteile in der IQ-Ebene analysieren.

Beispielsweise wurde die in Abbildung 74 gezeigte Information mit einer Time Sink des Typs Complex vor dem Quadrature-Demod-Block abgegriffen und das in Abbildung 75 gezeigte Signal nach der Quad. Demod. ebenfalls mit einer Time Sink aber des Typs Float visualisiert. Für eine Analyse der empfangbaren Werte, wurde eine „Frequency Sink“ und eine „Fosphor Sink“ nach dem Xlating FIR Filter verwendet. Um den abschließenden reinen Signaldurchlass zu sehen, wurde nach dem Simple Squelch oder dem Power Squelch ebenfalls eine entsprechende Sink zur Anzeige eingesetzt. Generell wurde hier an verschiedensten Stellen mit verschiedensten Anzeigeelementen gearbeitet. Abbildung 77 zeigt die in Bild 76 verwendeten Sinks.

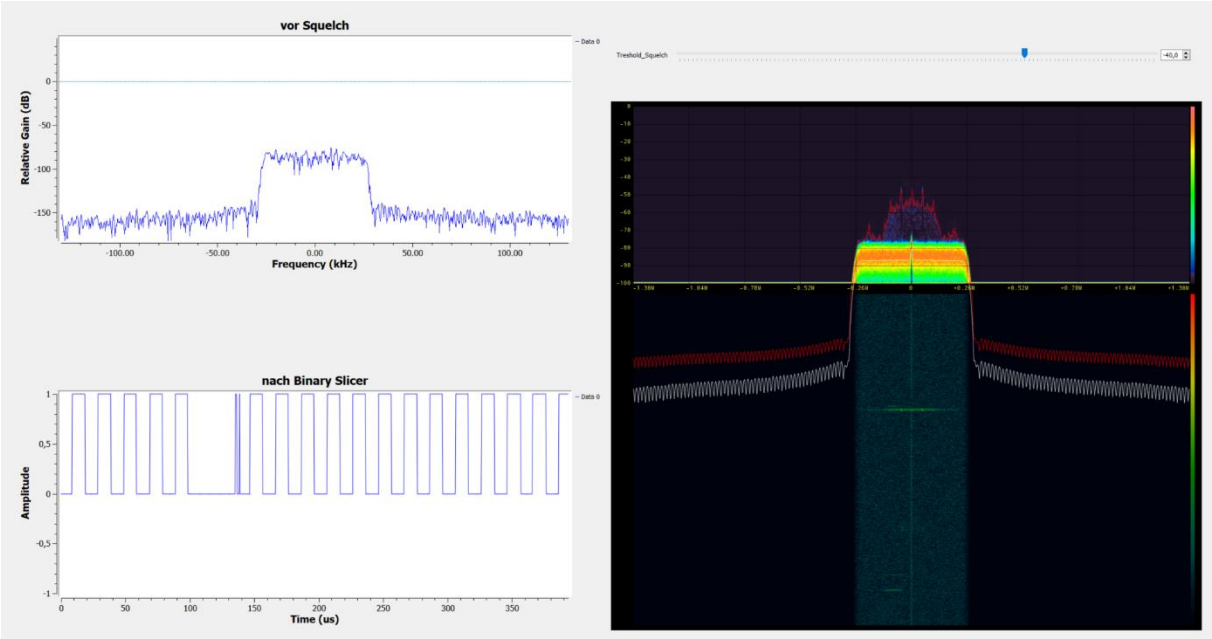


Abbildung 77: ausgewählte Anzeige- und Steuerelemente zur Signalanalyse
(Quelle: Eigene Darstellung)

4.4.7 Ausgabewerte im Formattyp Float

Nachdem nun das Signal empfangen, gesplittet, transformiert, gefiltert, generell begrenzt und seine ursprüngliche Modulationsart umgewandelt wurde, wird es in eine für den PC sinnvoller auswertbare Form, der Binärform, gebracht. Bereich G im Bild 76 stellt den ersten von hier zwei Abschnitten, welcher zu einer speichernden Sink, der File Sink, führt, dar. In diesem Part werden die Daten als Float-Werte gespeichert, um diese optisch mit dem Programm Inspectrum auswerten zu können. Für eine saubere Form des umgewandelten Signals und eine Unterscheidung der Werte in lediglich 0 und 1, wird ein Binary Slicer eingesetzt. Dieser markiert positive und negative Werte. Der Output des Quadtrature-Demod-Blocks zeigt, dass genau solch ein Format entsteht. Entsprechend ist es für den Binary Slicer möglich diesen Output auszuwerten und umzuwandeln. Für die optische Analyse des demodulierten Signals, wird hierbei der entstehende Bitstrom wieder in den Datentyp Float gewandelt, um ihn anschließend in der File Sink des Typs Float speichern zu können. Einen direkten Vergleich nach der Umwandlung, mit und ohne einen verwendeten Binary Slicer in der Betrachtungsanalyse, wird in Abbildung 79 gezeigt. Bild 78 zeigt eine Time Sink direkt nach der Konvertierung des Binary-Slicer-Outputs in Float-Variablen.

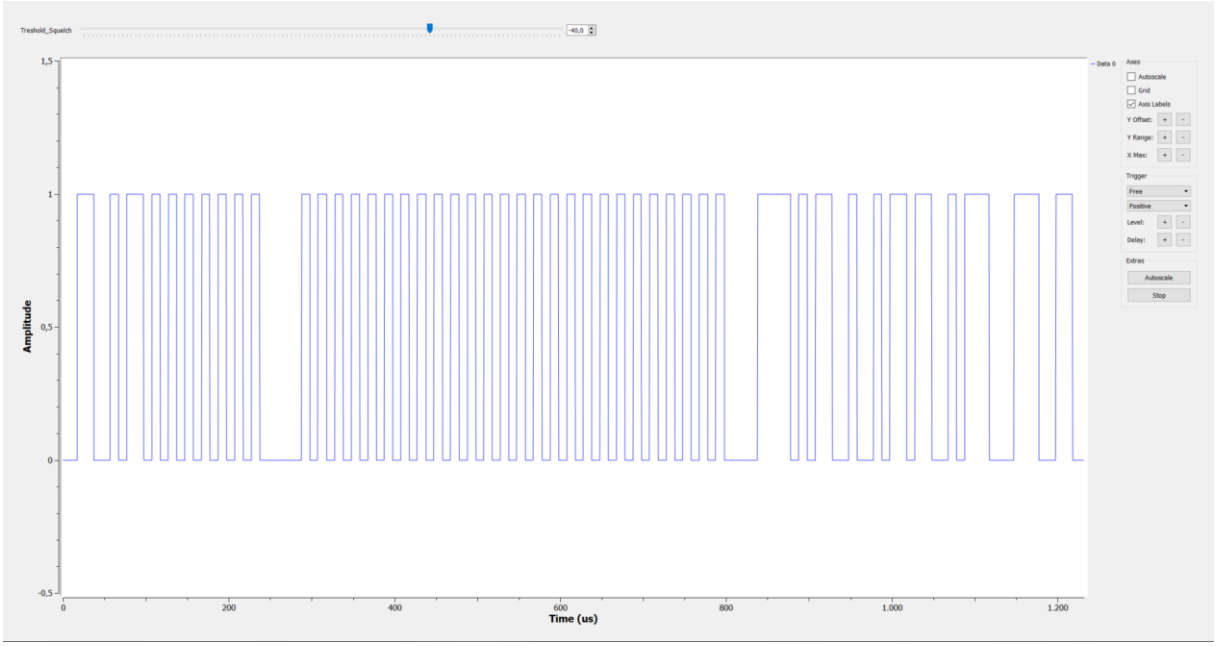


Abbildung 78: zeitlicher Signalverlauf nach erfolgter Demodulation und Umwandlung in qualitativ aussagekräftigen Float-Werten in GNU Radio
(Quelle: Eigene Darstellung)

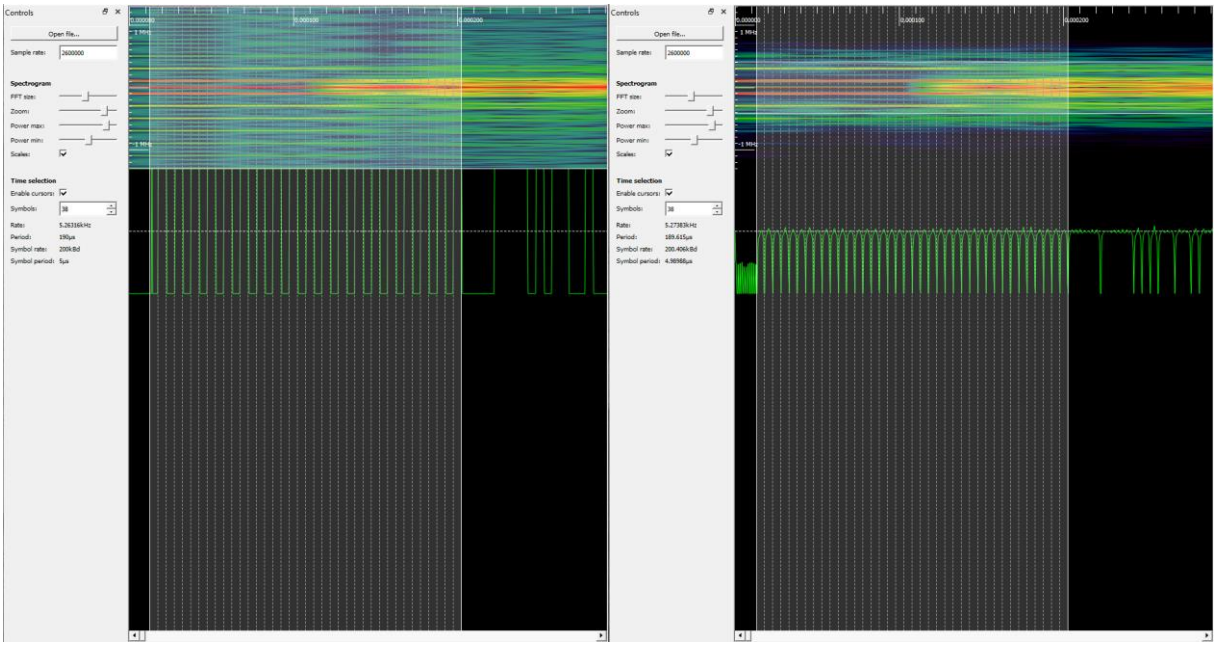


Abbildung 79: Gegenüberstellung Amplituden- (links) und Frequenzplot (rechts) des demodulierten Signals in Inspectrum
(Quelle: Eigene Darstellung)

Zu sehen ist auf der linken Seite die aufgenommene Informationsmenge, mit einem Binary Slicer, zwischen der File Sink und dem Quad-Demod-Block. Auf der rechten Seite die unveränderte Aufzeichnung, also ohne Binary Slicer. Bei der Plotauswahl in Inspectrum wird links ein Amplitudenplot verwendet. Wie bereits im Vorfeld beschrieben, wurde aus den ankommenden I- und Q-Strömen ein einheitliches Signal generiert in Form einer gepulsten

Amplitudenmodulation. Wie in Abbildung 75 zu sehen, weisen diese geformten und dem binärformat ähnelnden Segmente kleinere „Sub-Frequenzanteile“ auf (zu erkennen an den nicht geradlinig verlaufenden Bergen und Tälern). Diese werden durch den Binary Slicer „geglättet“, denn dieser unterscheidet bei der Umschaltung von positiven zu negativen und negativen zu positiven Werten (Amplitudenbereich und Wechsel der Signale im Output-Diagramm zwischen ca. -1 und +1). Es ist dabei jedoch ebenfalls wichtig diese kleineren Wellen zu detektieren, um eine Unterteilung der Bits treffen zu können. Würde diese Funktionseinheit lediglich auf den Wechsel von positiven und negativen und nicht auf den Wechsel von positiveren und negativeren Werten schauen, wäre eine Kombination/Bitreihenfolge aus 0*n oder 1*n eine einzelne 0 oder 1, da in diesem Block kein Takt einstellbar ist. Die rechte Seite in Abbildung 79 zeigt dabei einen Frequenzplot, da die Auswahl eines Amplitudenplots keinen Erfolg hatte bzw. keine Anzeige erbrachte. Hier sind lediglich Signalwechsel zu sehen.

4.4.8 Ausgabewerte im Formattyp Byte

Abschnitt H ebnet die Auswertung für weitere diverse Programme, welche eine Analyse über das Binärformat vornehmen. Beispielsweise lässt sich dieses Binaryfile mit dem Programm HxD lesen (oder einem anderen Hex-Editor). Um nun aber die analog angehafteten Informationen in reine 1 und 0 umzuwandeln und diese in dem Format Byte anstatt Float zu speichern, ist hierbei eine Detektion der einzelnen Bits notwendig.

Um solch eine Form zu erreichen sind in GRC zwei bereits implementierte Blöcke notwendig. Der ebenfalls im Float-Part verwendete und erwähnte Binary Slicer und zusätzlich die Funktionseinheit Clock Recovery MM. Die Taktwiederherstellung von Mueller und Müller kommt in GNU Radio, wie bereits im Theorieteil unter Abschnitt 2.3.4 erwähnt, mit vordefinierten Parametern, die nur leicht modifiziert werden müssen, um bereits dadurch ein vorzeigbares Ergebnis zu erzielen. Ein wichtiger Punkt ist dabei der Parameter Omega, welcher durch „samp_per_sym“ ausgedrückt wird. Der Variablenname samp_per_sym steht dabei für Samples per Symbol. Mit Sample ist die Sample-Rate gemeint und mit Symbol, wie man aus der 2FSK-Modulation weiß, ein Bit pro Frequenzvariationsabschnitt. Die Sample-Rate ist bekannt, da diese in GRC definiert werden muss, bevor diverse Blockeinheiten richtig arbeiten können. Die Datenrate oder auch Baudrate oder auch Symbols per Second lässt sich aus den selbst eingetragenen Paramtern des Transmitter-Programms auslesen. Sollte dieser Luxus im Vorfeld nicht gegeben sein, so gelingt die Erkundung der Baudrate beispielsweise mit dem Programm Inspectrum, in dem ein Float-File der Signale eingelesen und optisch analysiert wird, mittels setzen des Cursers. Die Baudrate wird hierbei als Symbol-Rate bezeichnet und in der Einheit Baud (Bd) angegeben. Sie drückt aus wie viele Symbole pro

Sekunde übertragen werden. Daraus ergibt sich Abtastpunkte pro Sekunde (f_{SAM}) im Verhältnis zu Symbolen pro Sekunde (f_{SYM}), also

$$T_{SAM} = 1s;\ f = \frac{1}{T_{SAM}} \rightarrow \frac{Abtastpunkte}{1\ Sekunde} = f_{SAM}$$

$$T_{SYM} = 1s;\ f = \frac{1}{T_{SYM}} \rightarrow \frac{Symbole}{1\ Sekunde} = f_{SYM}$$

$$\frac{f_{SAM}}{f_{SYM}} \rightarrow \frac{Abtastpunkte}{1\ Sekunde} : \frac{Symbole}{1\ Sekunde} \rightarrow \frac{Abtastpunkte}{Sekunde} * \frac{Sekunde}{Symbol}$$

$$\rightarrow \frac{Abtastpunkte}{Symbol}$$

Formel 19: zusammenfassende Beschreibung der Variable "samp_per_sym", welche unter dem Parameterwert "Omega" der M&M-Clock-Recovery verwendet wird

Diese Information, welche unter Parameter Omega übergeben wird, hilft diesem Block dabei die Höhen und Tiefen des Quad-Demod-Outputsignal im richtigen Takt zu erkennen. Das Ausgangssignal des Mueller&Müller-Blocks sieht dabei wie in Abbildung 80 aus.

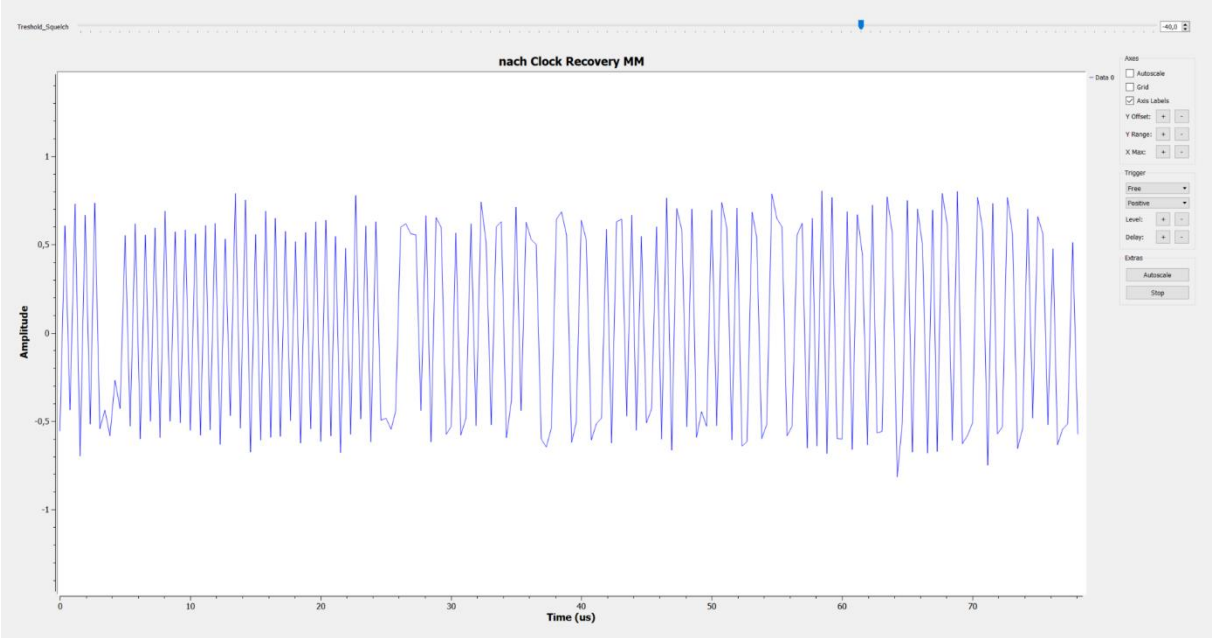


Abbildung 80: zeitlicher Signalabgriff am Clock Recovery Output
(Quelle: Eigene Darstellung)

Da hier nun ebenfalls eindeutig die Wellenberge und -täler detektierbar angezeigt werden und zwischen positven und negativen Amplitudenwerten variieren, kann die nachfolgende Einheit, der Binary Slicer, diese Werte nehmen und sie in einheitliche Einsen und Nullen umwandeln.

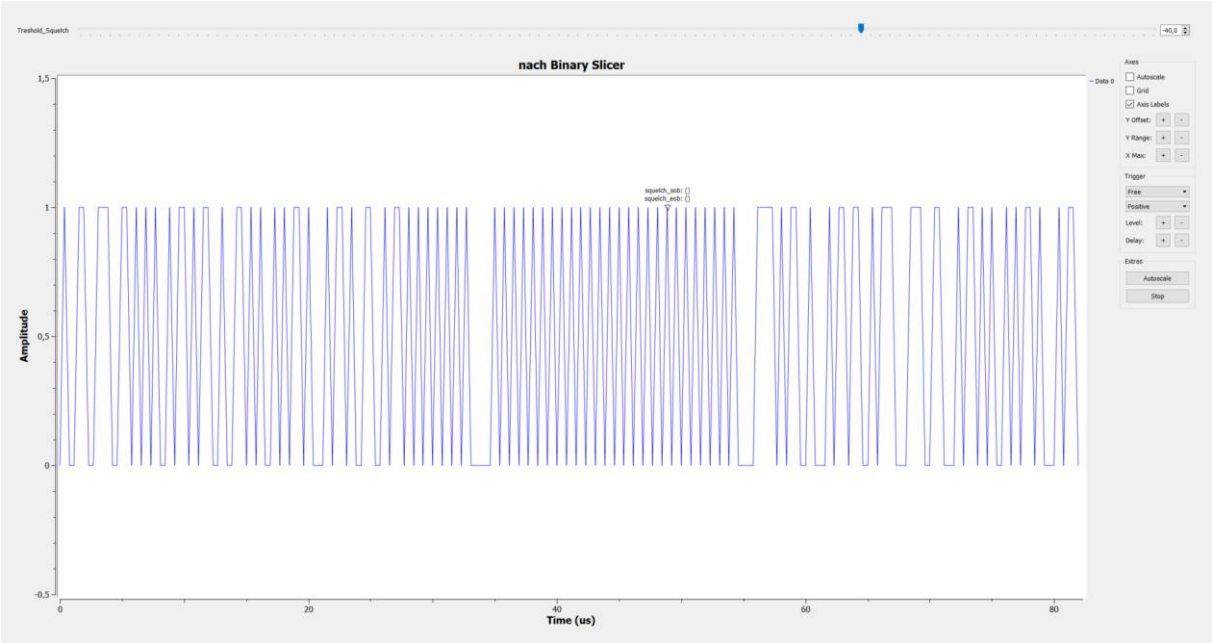


Abbildung 81: zeitlicher Signalabgriff nach Clock Recovery Output und Binary Slicer (Quelle: Eigene Darstellung)

Die anschließende Speicherung für eine beispielsweise weitere, dem GRC externe, Verarbeitung, erfolgt durch eine File Sink des Typs Byte. Man erkennt hier gut die weniger steilen Flanken und eine Interpolation von Verbindungslinien zwischen wenigen Punkten. Die wenigen Punkte entsprechend dabei den Rohdaten, den detektierten Einsen und Nullen. Eine Ähnlichkeit ist zwischen dem CRMM-Output und dem Output des nachfolgenden Binary Slicers unverkennbar. Der optische Vergleich ist hierbei konkreter gesehen nicht relevant. Es kommt in diesem Part auf die reinen Rohdaten für eine Weiterverarbeitung an.

Beispielsweise sieht der Inhalt solch einer Datei wie in Bild 82 gezeigt aus.

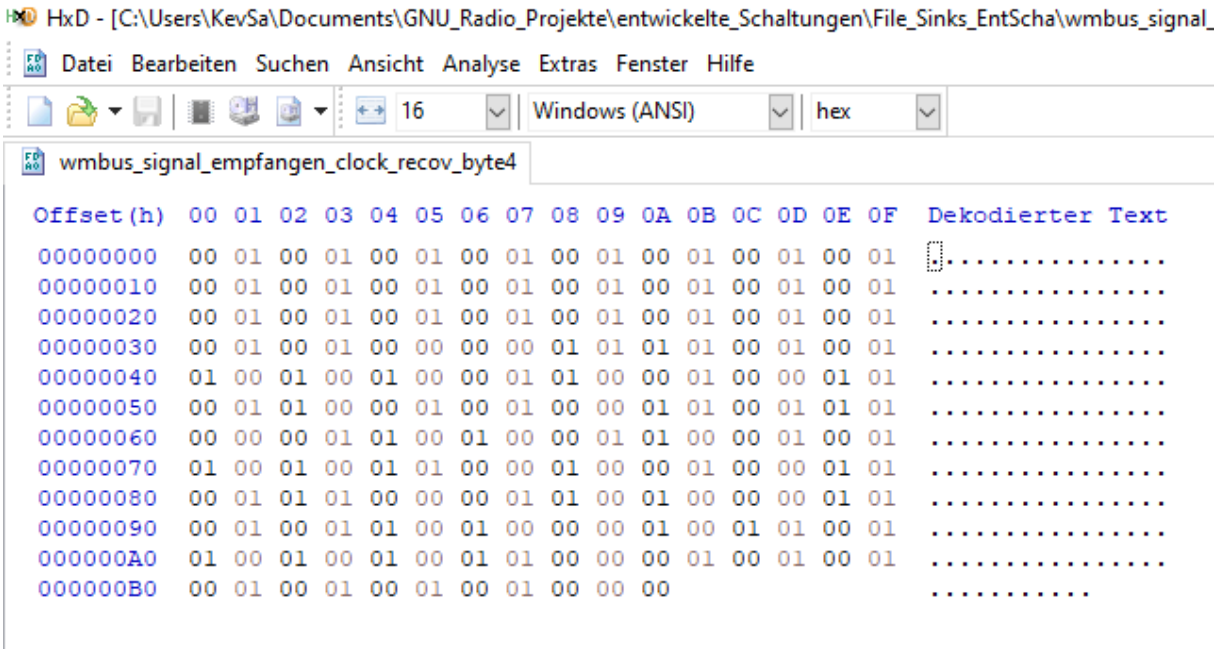


Abbildung 82: Darstellung eines demodulierten wM-Bus-Frames in binärer Form mit dem Programm HxD (Quelle: Eigene Darstellung)

5 Dekodierung der empfangenen Daten

Schaut man sich die empfangene Datenfolge an, erkennt man nach ein paar Zeichen, dass eine gewisse Unregelmäßigkeit vorhanden ist oder dass keine taktabhängig gezwungene Varianz der Bitfolgen stattfindet. Auch erkennt man unter mehreren und unterschiedlichen empfangenen und demodulierten Nachrichten, zu beginnn des Signals, ein gewisses sich wiederholendes Muster. Eine lange Folge von aneinander gereihten „01“-Stellen ist dabei unverkennbar. Gefolgt wird dieses Muster immer von einer etwas längeren Serie aus Nullen, 3 bis 4 Stück, und einer ebenso langen Folge von Einsen. Danach schließt sich wieder ein kurzer 01-Wechsel an. Dieses Muster ist bei allen wM-Bus-Frames, die während des Bearbeitungszeitraums aufgezeichnet wurden, zu finden. Ein Blick in die wM-Bus-Norm DIN EN 13757-4 bestätigt dabei diese Beoachtungen und lässt sie plausibel erscheinen. [17]

5.1 Vergleich der Gegebenheiten

In dieser Arbeit wurden Frames auf der Frequenz 868,95MHz empfangen. Eine Frequenzabweichung von ca. 50kHz, bei einer 2FSK-Modulation, ist dabei ebenfalls markant. Die anfänglichen Muster der Nachrichten stellen immer eine gewisse Regelmäßigkeit dar. Vorn angestellt ist dabei immer eine Folge von n * 01, also 01010101010101010101.... Danach folgt immer direkt 0000111101. Der nachstehende Informationsgehalt weißt keine gezwungene taktabhängige Regelvarianz auf, sondern wirkt eher wie eine bunte Mischung aus Nullen und Einsen.

All diese Gegebenheiten lassen darauf deuten, dass es sich bei den empfangenen wM-Bus-Frames um deren Subtyp T handelt. In DIN EN 13757-4 ist unter Abschnitt 6.4.2.3 – „Betriebsart T1 und T2 – Zähler sendet: Präambel und Synchronisationsmuster“ die anfängliche Datenfolge beschrieben, welche auch zu den Beobachtungen passt. Genauer einkreisen lässt sich damit der Subtyp auf T1/T2 eines sendenen Zählers. Dabei wird beschrieben, dass die Folge aus n*01 den Nachrichtenkopf bildet. Das angrenzende Muster „0000111101“ ist die eindeutige Synchronisationsstruktur. Beides zusammegefasst wird als Präambel bezeichnet und bildet somit immer die Vorhut eines wM-Bus-Frames. Die unter Abschnitt 6.2 – „Betriebsart T: Sender“ aufgeführte Tabelle 8 lässt ebenfalls auf diesen Verdacht des spezifizierten und ermittelten Subtyps schließen. Sie enthält eine Auflistung diverser charakteristischer Mittenfrequenzen, Frequenzhübe, etc., deren Toleranzen und die zugehörigen Subtypen. Laut dieser Tabelle liegt bei einer Mittenfrequenz von 868,95MHz und einem FSK-Hub von ca. 50kHz ebenfalls die Unterart „T1,T2-meter to other“ vor. Abbildung 83 zeigt dabei den in Inspectrum dargestellen Datenstrom.

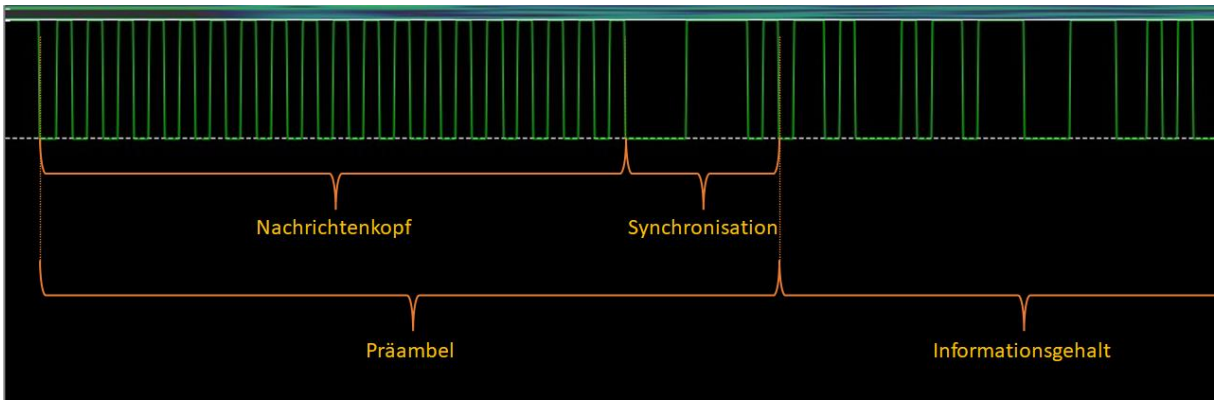


Abbildung 83: Präambel eines wM-Bus-Frames des Subtyps T1, T2-meter to other (Quelle: Eigene Darstellung)

Die im Vorfeld erwähnte ausgeprägte Unregelmäßigkeit an verteilten Nullen und Einsen im Imformationsgehaltsabschnitt, deutet nicht auf eine Manchestercodierung hin. Dies wird bestätigt durch Abschnitt 6.4.2 – „Betriebsart T – Zähler sendet: „3-aus-6“-Datencodierung“, welche angibt, bei einen sendenen Zähler im Subtyp „T1,T2-meter to other“, eine 3-out-of-6-Codierung zu verwenden. [17]

5.2 Anwendung der Dekodierung „3-out-of-6“

Bei dieser, im Gegensatz zum Manchestercode, effektiveren Form der Datenverschlüsselung, geht um es eine eindeutige Zuordnung bestimmter Werte, welche den eigentlichen Informationsgehalt ausdrücken sollen. Jedes Halbbyte, oder auch Nibble genannt, also bestehend aus 4 Bit, wird einer eindeutigen 6-Bit-starken Zeichenfolge zugeordnet. Diese wurden so ausgewählt, dass sie jeweils 3 Nullen und 3 Einsen haben und mindestens 2 Zustandswechsel beeinhalten. Aus den möglichen Kombination (64, da 2^6) treffen diese Kriterien auf 16 eindeutige Bitfolgen zu. Eine weitere Regel dieser Codierungsart legt fest, wie rum ein 6-Bit-Code übertragen wird. Dabei gelangt immer zuerst das MSB, die linke Seite der sechsstelligen Zeichenfolge, und das höchstwertige Nibble (selbstverständlich als 6-Bit-Code ausgedrückt) auf den Übertragungskanal. Im Anhang B ist die 3-out-of-6-Kodierungs/-Dekodierungstabelle zu finden.

Nachdem nun die Rohdaten des übertragenen Signals unverändert darliegen, ist es möglich, eine Dekodierung dieser vorzunehmen. In der Betriebsart, der die hier aufgefangenen Daten unterliegen, entspricht dieser der erwähnten 3-aus-6-Kodierung.

Um provokant zu demonstrieren, dass man unter der Verwendung eines Software Defined Radio, und der vorher erwähnten Schaltung zur Demodulation, rein mit den Rohdaten des Signal arbeitet und diese komplett in der Hand hat, wird eine für diesem Zweck entwurfene Exceltabelle verwendet. In dieser werden die in die Binärform konvertierten Radiowellen

eingelesen und zunächst chipweise aufgereiht, wie in Abbildung 84 zu sehen. Die Rohdaten können in diesem Aufbau direkt aus dem Byte-File entnommen oder aus Inspectrum, über den Threshold-Plot, extrahiert werden.

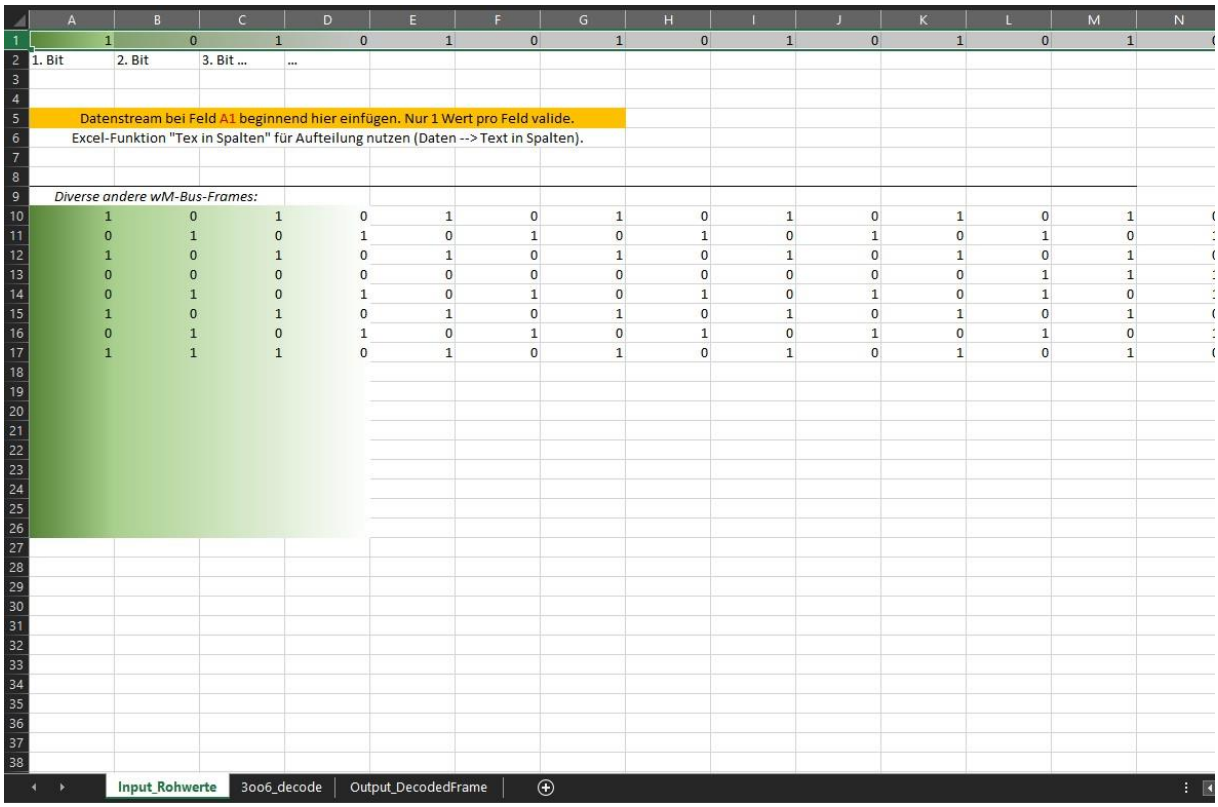


Abbildung 84: Input der Dekodierungs-Exceldatei
(Quelle: Eigene Darstellung)

In diesem Tabellenblatt ist lediglich die spaltenweise Trennung der Zeichen wichtig und der Beginn bei Zelle A1. Von dieser Anordnung ausgehend, kann das folgende Tabellenblatt funktionsgerecht agieren. Abbildung 85 zeigt die 3 Stadien der eigentlichen Dekodierung. Sinn und Zweck dieser Excel-Tabelle ist es, neben dem provokanten demonstrieren der vollumfänglichen Verfügbarkeit der Rohdaten, einmal mehr zu zeigen, wie die Daten selbst entdeckt werden können. Aus diesem Grund sind 2 Wahlschalter hinterlegt. Der Erste bestimmt eine Variable, welche einen Versatz der Maskenmatrix zur Folge hat. Dies ist notwendig und zugleich interessant zu entdecken, da der Anfangswert einer gültigen 6-bit-Folge von vornherein, auf Grund der Präambel und des zufällig kopierten Rahmens, mit hoher Wahrscheinlichkeit nicht passt. Es wird somit der Anfang des Frames, und somit auch der Anfangswert der zu suchenden Zeichenkette verschoben. Der Anwender sieht dabei, mit einem direkten optischen Feedback, welches Verhalten diese Varianz zur Folge hat.

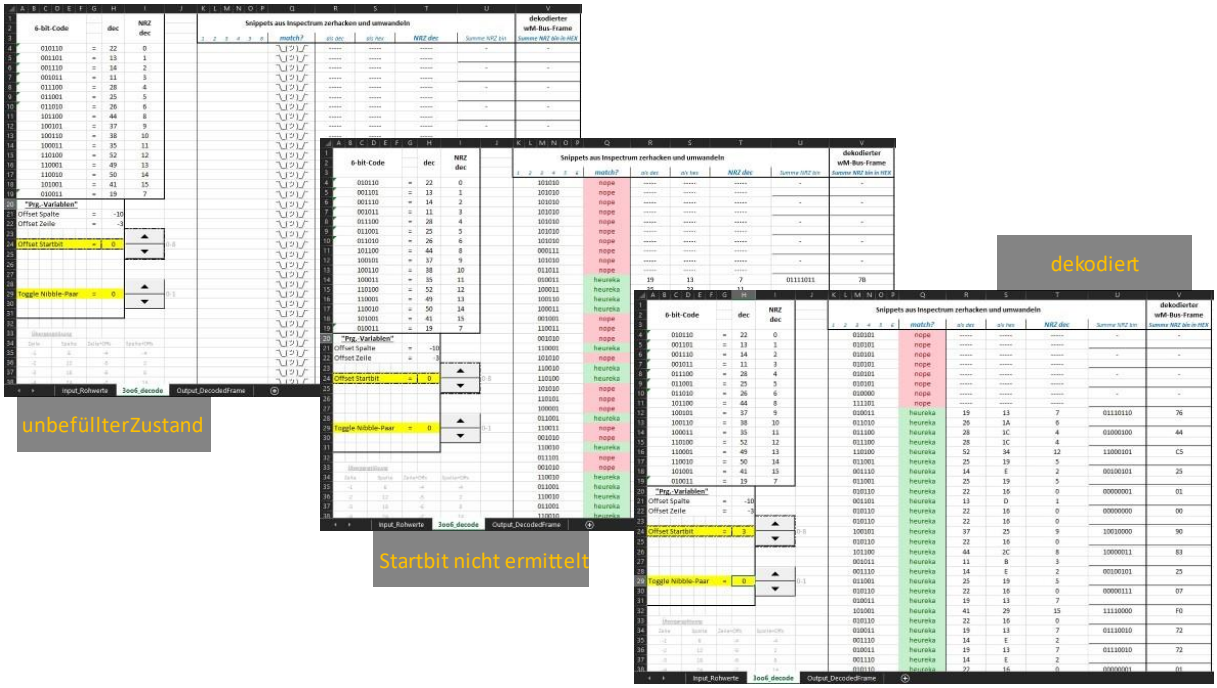


Abbildung 85: Stadien der eigentlichen Dekodierung
(Quelle: Eigene Darstellung)

Die mittlere Tabelle in Abbildung 85 zeigt dabei den noch nicht erreichten korrekten Startwert für die 6-bit-Dekodierung-Sequenz. Rotmarkierte Kästchen deuten dabei auf eine Nichtübereinstimmung mit Werten aus der vorgegebenen Dekodierungstabelle hin, wohingegen ein grünes Kästchen auf ein gefundenes Match hinweist. Die rechte Darstellung in Abbildung 85 zeigt den finalen Zustand, also den Zustand der gänzlich erfolgreichen Suchmatrix. Der zweite Wahlschalter soll dabei nun die Entdeckung der richtigen Wertepaare ermöglichen. Der Wert hinter der Variable des Schalters nimmt dabei 0 oder 1 an. Der Anwender kann somit variieren, welche Nibble zusammengezogen werden sollen, für eine finale Dekodierung des Signals. Er wird dabei von automatisch entstehenden Trennstrichen unterstützt. Sollten diese mit der vorhergehenden Spalte in der jeweiligen Zeile übereinstimmen, ist das richtige Halbbytepaar gefunden. Diese werden in der angrenzenden Auswertungsspalte in eine hexadezimale Form gewandelt.

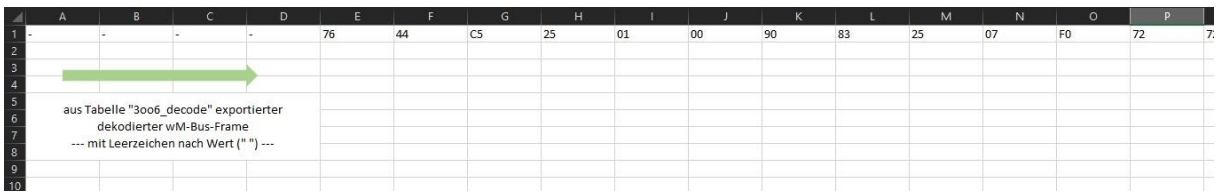
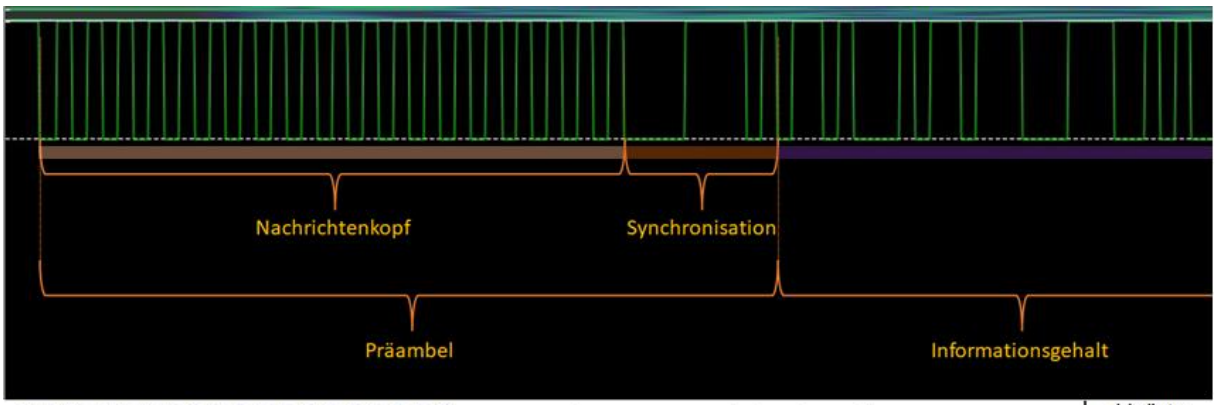


Abbildung 86: Output der Dekodierungs-Exceldatei
(Quelle: Eigene Darstellung)

Diese Werte werden in das nächste Tabellenblatt extrahiert, spaltenweise aufgereiht und ihnen wird zusätzlich ein Leerzeichen angehangen, welches dienlich für die sinngemäße Zeichen-

trennung weiterer externer Verarbeitungsschritte ist. Eine Darstellung dieses Vorgangs ist in Bild 86 zu erkennen.

Abbildung 87 zeigt eine Collage einer demodulierten und dekodierten Nachricht, mit Augenmerk auf die sichtbare Unterteilung in die Abschnitte Präambel und Informationsgehalt.



Snippets aus Inspectrum zerhacken und umwandeln

1 2 3 4 5 6	match?	als dec	als hex	NRZ dec	Summe NRZ bin	dekodierter wM-Bus-Frame: Summe NRZ bin in HEX
010101	nope	-----	-----	-----	-	-
010101	nope	-----	-----	-----		
010101	nope	-----	-----	-----	-	-
010101	nope	-----	-----	-----		
010101	nope	-----	-----	-----	-	-
010101	nope	-----	-----	-----		
010000	nope	-----	-----	-----	-	-
111101	nope	-----	-----	-----		
011010	heureka	26	1A	6	01100011	63
001011	heureka	11	B	3		
011100	heureka	28	1C	4	01000100	44
011100	heureka	28	1C	4		
101001	heureka	41	29	15	11111010	FA
100110	heureka	38	26	10		
001011	heureka	11	B	3	00110000	30
010110	heureka	22	16	0		
010110	heureka	22	16	0	00000011	03
001011	heureka	11	B	3		
010110	heureka	22	16	0	00000000	00
010110	heureka	22	16	0		
010110	heureka	22	16	0	00000000	00
010110	heureka	22	16	0		
010110	heureka	22	16	0	00000000	00
010110	heureka	22	16	0		
010110	heureka	22	16	0	00000000	00
010110	heureka	22	16	0		
010110	heureka	22	16	0	00000010	02
001110	heureka	14	E	2		
001110	heureka	14	E	2	00101110	2E

Abbildung 87: Zusammenfassende Collage der demodulierten und der dekodierten Daten (Quelle: Eigene Darstellung)

Laut der Norm folgt dieser Präambel immer ein Telegramm des Typs A. Dieses ist ebenfalls in dem Dokument beschrieben. Damit ließen sich die dekodierten spezifischen Daten korrekt interpretieren. [17]

6 Zusammenfassung

Mit dem hier entwickelten Programm ist es möglich, die Daten einer FSK-Übertragung, so wie wM-Bus eine nutzt, zu empfangen, zu demodulieren und in verschiedenen Datentypen aufzuzeichnen. Diese können anschließend zur weiteren Datenverarbeitung genutzt werden. In erster Linie wird dazu eine Dekodierung des aufgezeichneten Informationsgehalts notwendig sein. Zusammenfassend kann man aus den kreierten und untersuchten Anwendungen sagen, dass der Empfang von wM-Bus-Daten, unter der Verwendung von GNU Radio und einem geeigneten SDR, problemlos umsetzbar ist. Die Realisierung gelingt dabei mit den bereits im GRC implementierten Funktionseinheiten. Die Erprobung einzelner Teilabschnitte und die Realisierung des Konzepts, lässt sich mit den eingebauten Gegebenheiten bewerkstelligen. Der Kreativität sind dabei kaum Grenzen gesetzt, was eine möglichst allumfängliche Auswertung und Entwicklung bedingt.

7 Fazit

Generell ist die Entwicklung eines funkbasierenden Konzepts mit einem SDR-fähigen Gerät eine praktikable und angenehme Methode. Die für Änderungen in der Funktionsweise notwendige Anpassungen sind mehr oder minder schnell umgesetzt, im Vergleich zu überwiegend analogen bzw. allgemeinen Hardwarekomponenten. Wiederum wird dadurch eine allgemeine Kostenreduzierung bedingt. Für die Erprobung, Entwicklung oder Untersuchung von diversen Konzepten ist die SDR-Technologie unschlagbar einsatzfähig. Die immanente Vielfältigkeit macht dieses Gebiet zu einem unschlagbaren Faktor in der Entwicklung. Das Thema Software Defined Radio lässt sich dabei einsteiger- und fortgeschrittenenfreundlich mit den hier gezeigten Mitteln entdecken.

8 Ausblick

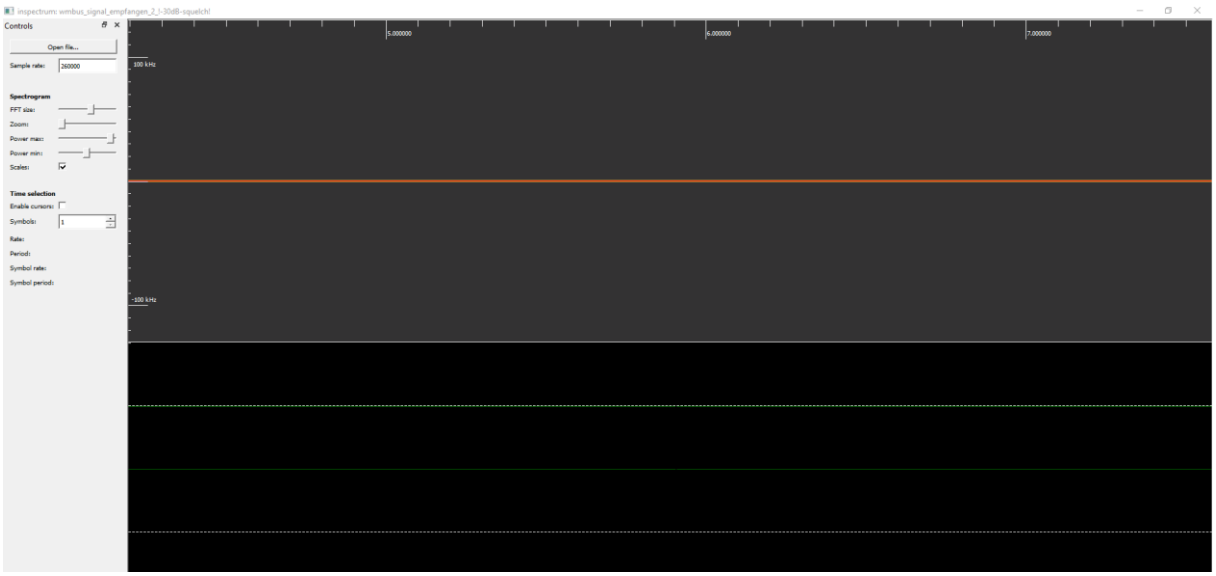
Diese Arbeit bietet einen kleinen Einblick in das mannigfaltige Einsatzspektrum der SDR-Technik. Es ist zugleich Erprobungswerkzeug in einer kostengünstigen aber dennoch qualitativ hochwertigen Form und Verlängerungsarm der Kreativität auf den Einsatzgebieten der Nachrichtenübertragungstechnik und der digitalen Signalverarbeitung.

Diese schier endlos erscheinenden Optionen werden vorallen durch den Aspekt der Programmierbarkeit bestimmt. Reichen beispielsweise die bereits in GRC vorhandenen Blöcke nicht aus, um eine spezifische Anwendung zu kreieren, lässt sich kurzer Hand eigener Code implementieren und umwandeln. Selbstverständlich beschränkt sich die Verarbeitung der Informationen nicht nur auf GNU Radio. Das SDR-Gerät stellt dabei den Datenstrom dem PC zur Verfügung, ungeachtet der verwendeten Software.

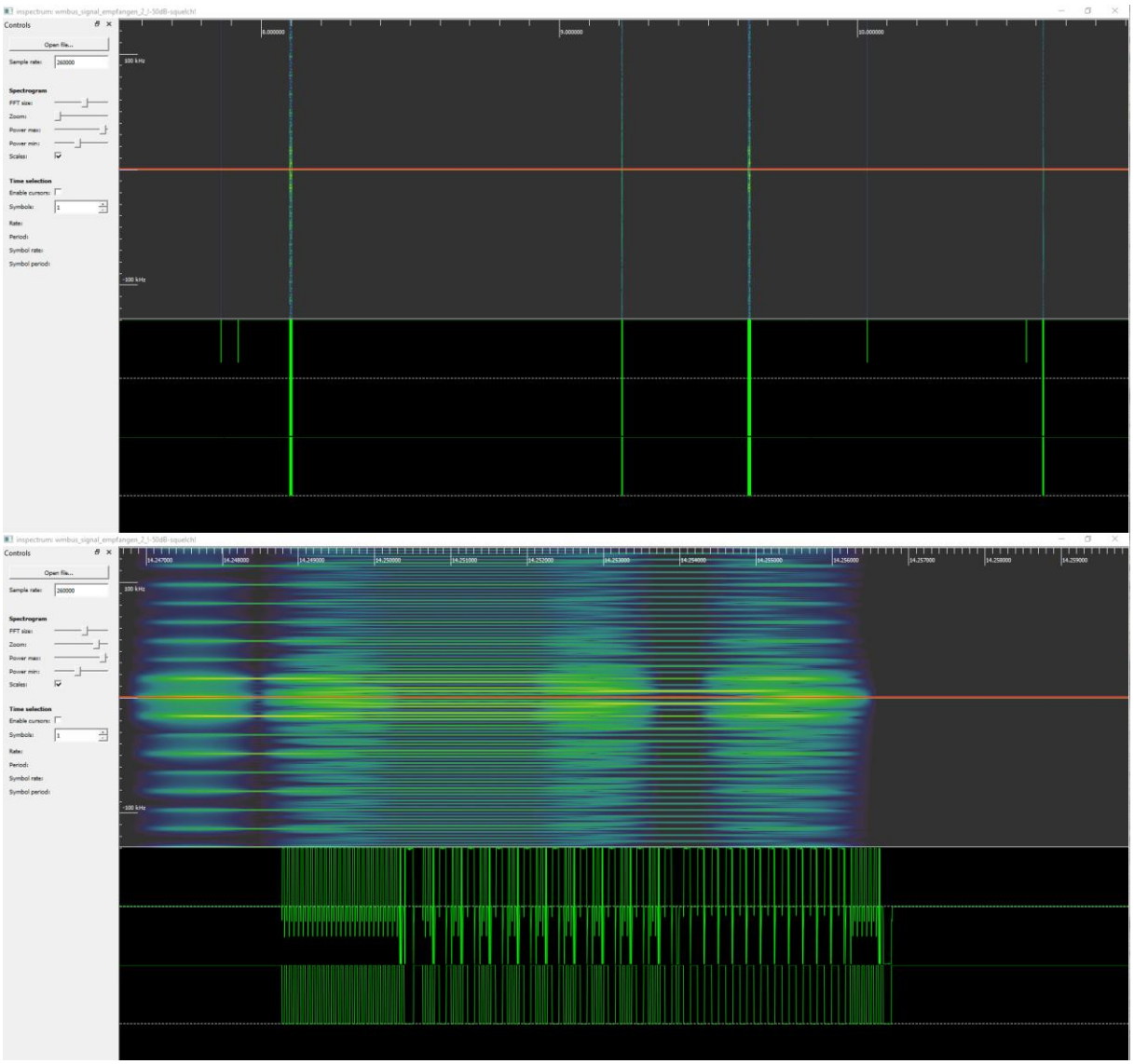
Die Verarbeitung von wM-Bus stellt hier nur eine Möglichkeit dar. So lassen sich Frequenzen sämtlicher Art, einzig allein durch den Empfangsbereich der verwendeten Hardware bedingt begrenzt, empfangen, demodulieren und weiterverarbeiten. Andere zertifizierte Übertragungstechniken wie beispielweise LoRa oder 5G lassen sich demnach ebenfalls mit einem mehr oder minder großen Aufwand empfangen und auswerten. Die Verfügbarkeit einer generellen Transmitterlizenz ist allerdings nicht automatisch gegeben. Generell ist im Funkverkehr darauf zu achten, unter welchen Frequenzen frei gefunkt werden darf oder lizensiert gefunkt werden soll. Der Entwurf einer SDR-Software für einen bestimmten Transceiver oder einen reinen Transmitter ist deshalb mit diversen Auflagen der Zertifizierbarkeit verbunden, um sich im gesetzlich legalen Rahmen zu bewegen.

Für die Erforschung und Entwicklung im privaten aber auch im professionellen kommerziellen Bereich lässt sich diese Technik vorteilhaft für die gedachten Anwendungsgebiete einbringen. GNU Radio bietet dabei eine große Open Source Platform, welche auf Interessentenzuwachs und deren Beiträgen basiert.

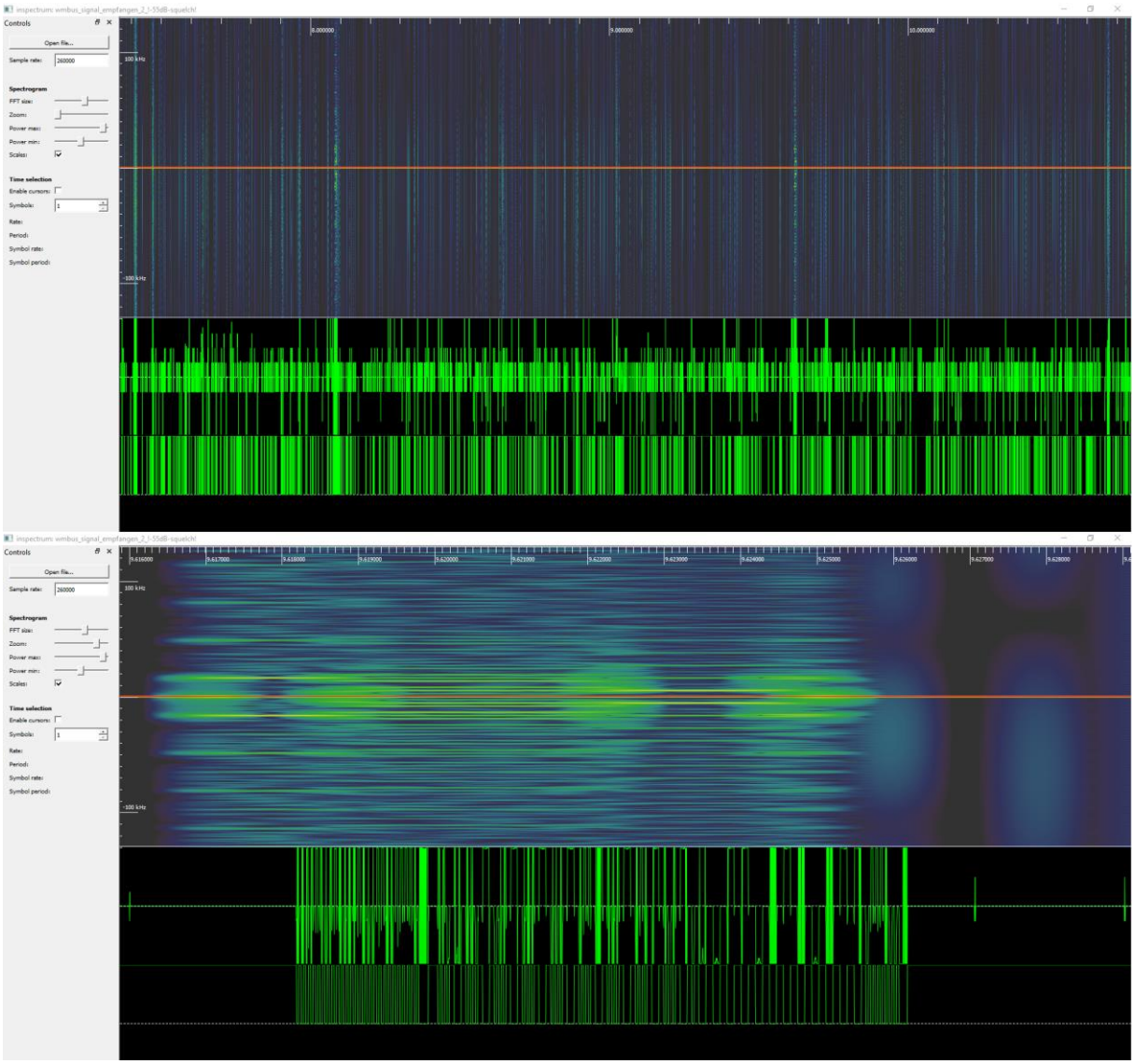
Anhang A – Testreihe Threshold-Wert



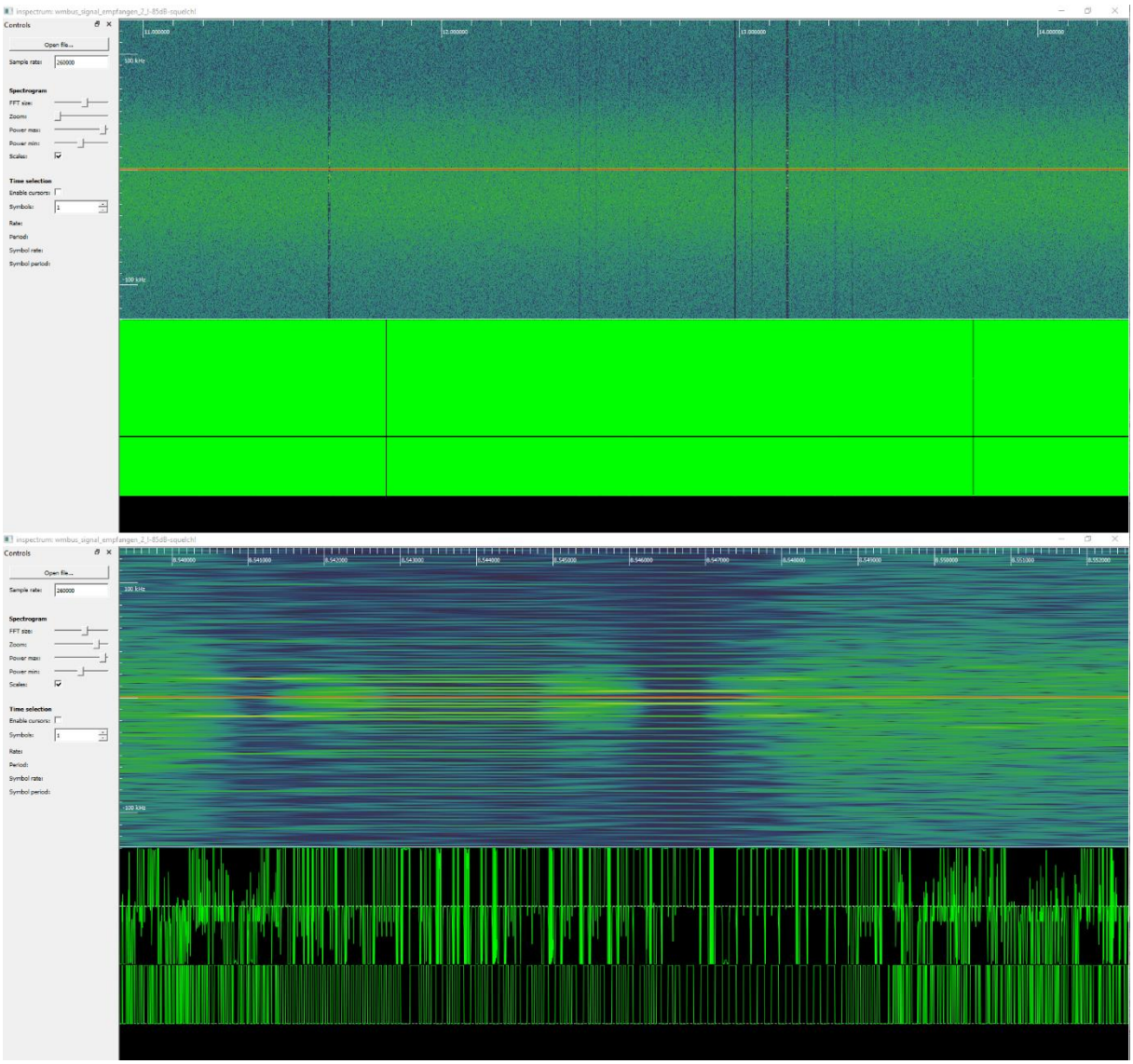
Anhang A - Abbildung 1: Threshold-Wert des Simple Squelch bei -30dB mit unveränderter Config/ Setup (Quelle: Eigene Darstellung)



Anhang A - Abbildung 2: Threshold-Wert des Simple Squelch bei -50dB mit unveränderter Config/ Setup (Quelle: Eigene Darstellung)



Anhang A - Abbildung 3: Threshold-Wert des Simple Squelch bei -55dB mit unveränderter Config/ Setup (Quelle: Eigene Darstellung)



Anhang A - Abbildung 4: Threshold-Wert des Simple Squelch bei -85dB mit unveränderter Config/ Setup (Quelle: Eigene Darstellung)

Anhang B – „3-aus-6“-Datencodierung

NRZ-Code	Dezimal	6-Bit-Code	Dezimal2	Zustandswechsel
0000	0	010110	22	4
0001	1	001101	13	3
0010	2	001110	14	2
0011	3	001011	11	3
0100	4	011100	28	2
0101	5	011001	25	3
0110	6	011010	26	4
0111	7	010011	19	3
1000	8	101100	44	3
1001	9	100101	37	4
1010	10	100110	38	3
1011	11	100011	35	2
1100	12	110100	52	3
1101	13	110001	49	2
1110	14	110010	50	3
1111	15	101001	41	4

Anhang B - Abbildung 1: eine Darstellung der „3-aus-6“-De-/Kodierungstabelle

(Quelle: Eigene Darstellung – Informationsgehalt aus DIN EN 13757-4)

Literatur- und Quellenverzeichnis

[1] lordneo, „Software-definiertes Radio - Wikipedia,“ [Online]. Available: https://wiki.edu.vn/wiki15/2020/11/19/software-definiertes-radio-wikipedia/. [Zugriff am 31 08 2022].

[2] Bremerfunkfreunde, „Bremerfunkfreunde,“ [Online]. Available: https://bremerfunkfreunde.de/index.php/sdr/geschichte-der-sdr-technik. [Zugriff am 31 08 2022].

[3] Frantsch, „Wikipedia,“ [Online]. Available: https://de.wikipedia.org/wiki/Software_Defined_Radio. [Zugriff am 31 08 2022].

[4] Marcindsp, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php?title=Quadrature_Demod. [Zugriff am 31 08 2022].

[5] D. Mietke, „Elektroniktutor,“ [Online]. Available: https://www.elektroniktutor.de/signalkunde/fsk.html. [Zugriff am 01 09 2022].

[6] „IT Administrator - Das Magazin für professionelle System- und Netzwerkadministration,“ [Online]. Available: https://www.it-administrator.de/lexikon/gaussian_frequency_shift_keying.html. [Zugriff am 01 09 2022].

[7] L. M. Ltd., „LimeMicro,“ [Online]. Available: https://limemicro.com/products/boards/limesdr. [Zugriff am 01 09 2022].

[8] STMicroelectronics, „ST - embedded software,“ [Online]. Available: https://www.st.com/en/embedded-software/stsw-s2lp-dk.html. [Zugriff am 01 09 2022].

[9] MubashiraZaman, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php/Frequency_Xlating_FIR_Filter. [Zugriff am 01 09 2022].

[10] MubashiraZaman, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php?title=Simple_Squelch. [Zugriff am 01 09 2022].

[11] Marcindsp, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php?title=Quadrature_Demod. [Zugriff am 01 09 2022].

[12] Keely, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php/Clock_Recovery_MM. [Zugriff am 01 09 2022].

[13] Duggabe, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php?title=Binary_Slicer. [Zugriff am 01 09 2022].

[14] MarcusMueller, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php?title=File_Sink. [Zugriff am 01 09 2022].

[15] 777arc, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php?title=GUI_Hint. [Zugriff am 01 09 2022].

[16] MubashiraZaman, „GNU Radio Wiki,“ [Online]. Available: https://wiki.gnuradio.org/index.php?title=QT_GUI_Range. [Zugriff am 01 09 2022].

[17] D. I. f. N. DIN, Kommunikationssystem für Zähler und deren Fernablesung - Teil 4: Zählerauslesung über Funk (Fernablesung von Zählern im SRD-Band); Deutsche Fassung EN 13757-4:2013, Berlin: Beuth Verlag GmbH, 2014.

[18] „IT Wissen,“ [Online]. Available: https://www.itwissen.info/Wireless-M-Bus-wireless-M-bus-wM-Bus.html. [Zugriff am 01 09 2022].

[19] D. W. Domschke, „oms-group,“ [Online]. Available: https://oms-group.org/fileadmin/files/press/in_der_presse/OMS_Lexikon.pdf. [Zugriff am 01 09 2022].

[20] „oms-group,“ [Online]. Available: https://oms-group.org/open-metering-system/warum-oms. [Zugriff am 01 09 2022].

[21] „IRCAM,“ [Online]. Available: https://support.ircam.fr/docs/AudioSculpt/3.0/co/FFT%20Size.html. [Zugriff am 01 09 2022].

[22] „nti-audio,“ [Online]. Available: https://www.nti-audio.com/de/service/wissen/fast-fourier-transformation-fft. [Zugriff am 01 09 2022].

[23] „Wikipedia,“ [Online]. Available: https://de.wikipedia.org/wiki/Gleitender_Mittelwert. [Zugriff am 01 09 2022].

[24] „TU Wien,“ [Online]. Available: https://ti.tuwien.ac.at/cps/teaching/courses/dspv/files/FIRFilter.pdf. [Zugriff am 01 09 2022].

[25] „TU Wien,“ [Online]. Available: https://ti.tuwien.ac.at/cps/teaching/courses/dspv/files/DSP_5-Signale.pdf. [Zugriff am 01 09 2022].

[26] „Wikipedia,“ [Online]. Available: https://de.wikipedia.org/wiki/Quadraturamplitudenmodulation. [Zugriff am 01 09 2022].

[27] w2aew, „#170: Basics of IQ Signals and IQ modulation & demodulation - A tutorial,“ Youtube, [Online]. Available: https://www.youtube.com/watch?v=h_7d-m1ehoY. [Zugriff am 02 09 2022].

[28] w2aew, „#171: IQ Signals Part II: AM and FM phasor diagrams, SSB phasing method,“ Youtube, [Online]. Available: https://www.youtube.com/watch?v=5GGD99Qi1PA. [Zugriff am 02 09 2022].

Selbstständigkeitserklärung

Ich, Kevin Saalmann, geboren am 20.10.1991 in Dessau-Roßlau, erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

_____ _____

(Ort, Datum) (Unterschrift)