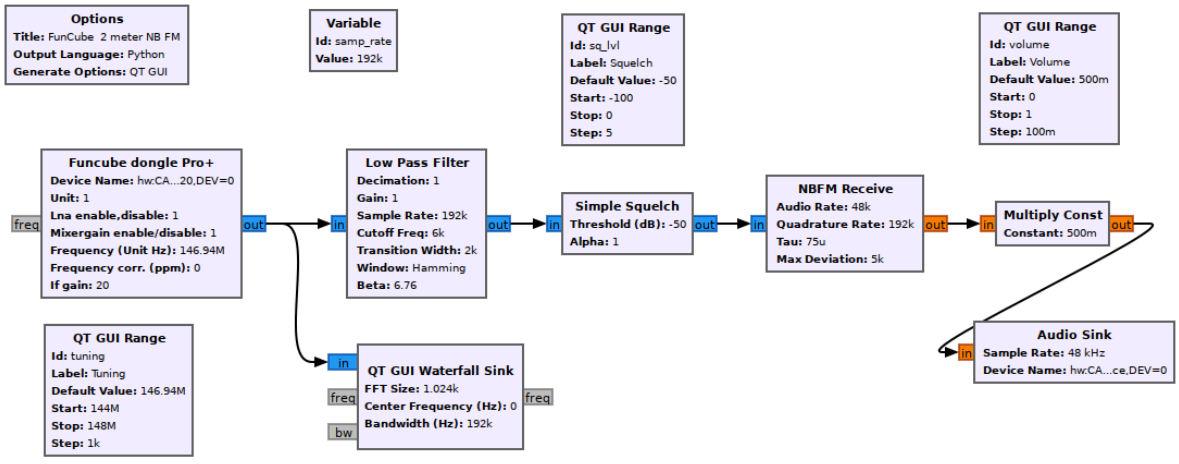
GNU-Radio Einführung

(Google-Übersetzung aus: https://wiki.gnuradio.org/index.php?title=What_Is_GNU_Radio

GNU Radio ist ein kostenloses Open-Source-Toolkit für die Softwareentwicklung, das Signalverarbeitungsblöcke zur Implementierung von Softwareradios bereitstellt. Es kann mit leicht verfügbarer, kostengünstiger externer HF-Hardware zur Erstellung softwaredefinierter Funkgeräte oder ohne Hardware in einer simulationsähnlichen Umgebung verwendet werden. Es wird häufig in Forschung, Industrie, Wissenschaft, Regierung und Hobbyumgebungen eingesetzt, um sowohl die drahtlose Kommunikationsforschung als auch reale Funksysteme zu unterstützen.

Hier sehen Sie ein Beispiel für ein Flussdiagramm im visuellen Editor von GNU Radio Companion:



GNU Radio ist ein Framework, das es Benutzern ermöglicht, hochleistungsfähige reale Funksysteme zu entwerfen, zu simulieren und bereitzustellen. Es handelt sich um ein hochmodulares, „Flowgraph“-orientiertes Framework, das über eine umfassende **Bibliothek von Verarbeitungsblöcken** verfügt, die problemlos zu komplexen **Signalverarbeitungsanwendungen** kombiniert werden können. GNU Radio wurde für eine Vielzahl realer Radioanwendungen verwendet, darunter Audioverarbeitung, Mobilkommunikation, Satellitenverfolgung, Radarsysteme, GSM-Netzwerke, Digital Radio Mondiale und vieles mehr – alles in Computersoftware. Gnu radio konzentiert sich nicht auf eine bestimmte Hardware. Es bietet auch keine sofort einsatzbereiten Anwendungen für bestimmte Funkkommunikationsstandards (z. B. 802.11, ZigBee, LTE usw.), aber es kann (und wurde) verwendet werden, um Implementierungen grundsätzlich aller bandbegrenzten Kommunikationsstandards zu entwickeln.

Warum sollte ich GNU Radio verwenden?

Früher musste der Ingenieur bei der Entwicklung von Funkkommunikationsgeräten eine spezielle Schaltung zur Erkennung einer bestimmten Signalklasse entwickeln, eine spezielle integrierte Schaltung entwerfen, die diese bestimmte Übertragung dekodieren oder kodieren konnte, und diese mithilfe kostspieliger Geräte austesten.

Software-Defined Radio (SDR) übernimmt die analoge Signalverarbeitung und verlagert sie, soweit physikalisch und wirtschaftlich machbar, auf die Verarbeitung des Funksignals auf einem Computer mithilfe von Software-Algorithmen.

Sie können Ihr mit dem Computer verbundenes Funkgerät natürlich in einem Programm verwenden, das Sie von Grund auf neu schreiben, Algorithmen nach Bedarf verketten und Daten selbst ein- und auslagern. Doch das wird schnell umständlich: Warum implementiert man einen Standardfilter neu? Warum müssen Sie sich darum kümmern, wie Daten zwischen verschiedenen Verarbeitungsblöcken übertragen werden? Wäre es nicht besser, hochoptimierte und von Experten überprüfte Implementierungen zu verwenden, als Dinge selbst zu schreiben? Und wie erreichen Sie, dass Ihr Programm auf Multi-Core-Architekturen gut skaliert, aber auch auf einem eingebetteten Gerät gut läuft, das nur ein paar Watt Strom verbraucht? Möchten Sie wirklich alle GUIs selbst schreiben?

Hier kommt GNU Radio ins Spiel: Ein Framework zum Schreiben von Signalverarbeitungsanwendungen für Standardcomputer. GNU Radio verpackt Funktionalität in benutzerfreundliche, wiederverwendbare Blöcke, bietet hervorragende Skalierbarkeit, stellt eine umfangreiche Bibliothek von Standardalgorithmen bereit und ist stark für eine Vielzahl gängiger Plattformen optimiert. Es enthält außerdem zahlreiche Beispiele, die Ihnen den Einstieg erleichtern.

Der Rest dieser Seite bietet eine kurze Einführung in DSP. Wenn Sie bereits mit DSP vertraut sind, können Sie gerne mit dem nächsten Tutorial fortfahren.

Digitale Signalverarbeitung

Als Software-Framework arbeitet GNU Radio mit digitalisierten Signalen, um Kommunikationsfunktionen mithilfe von Allzweckcomputern zu generieren.

Eine kleine Signaltheorie

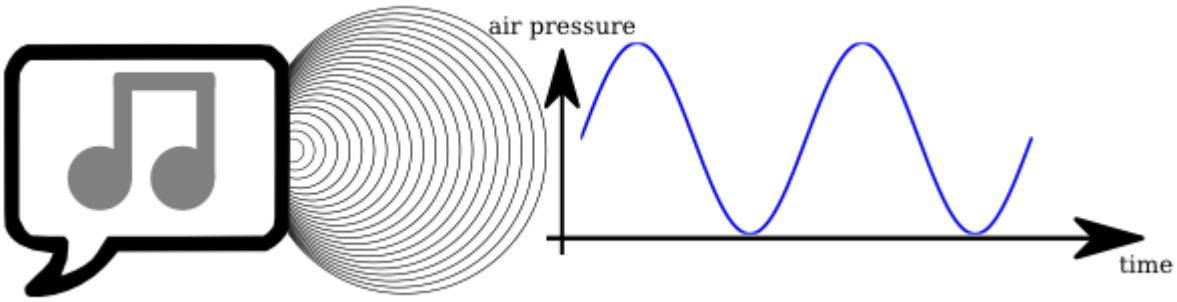
Für die Signalverarbeitung in Software muss das Signal digital sein. Aber was ist ein digitales Signal?

Um es besser zu verstehen, schauen wir uns ein häufiges „Signal“-Szenario an:

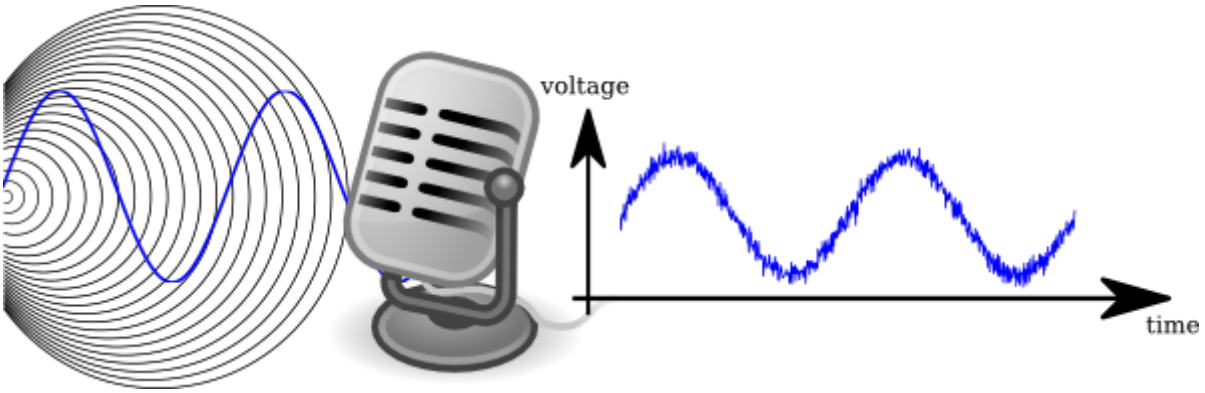
Sprachaufzeichnung zur Übertragung mit einem Mobiltelefon.

Eine physisch sprechende Person erzeugt ein Tonsignal – das Signal besteht in diesem Fall aus Wellen unterschiedlichen Luftdrucks, die von den Stimmbändern

eines Menschen erzeugt werden. Ein Signal ist eine zeitlich veränderliche physikalische Größe, wie zum Beispiel der Luftdruck.



Da das Signal nun elektrisch ist, können wir damit arbeiten. Das Audiosignal ist zu diesem Zeitpunkt analog – ein Computer kann damit noch nicht umgehen; Für die rechnerische Verarbeitung muss ein Signal digital sein, was zweierlei bedeutet:

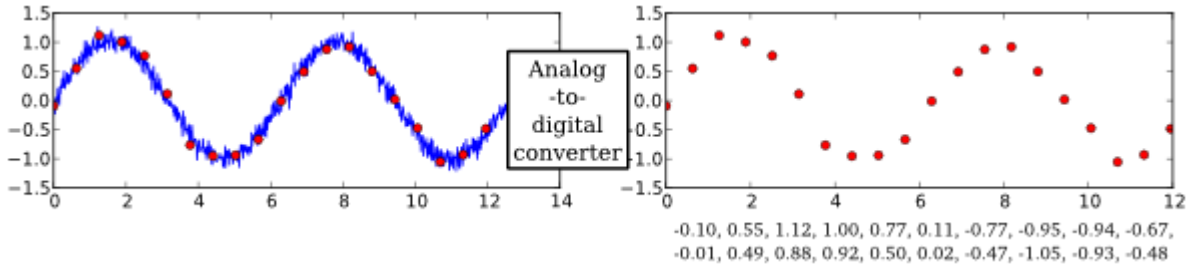


1. Es kann nur einer von einer begrenzten Anzahl von Werten sein.

Das Signal kann sich im Laufe der Zeit ändern, nimmt aber für jeden Moment nur einen Wert an – und dieser Wert stammt nicht aus einem „Kontinuum“ (wie [− 1,5 ; + 1,5], aber aus einer endlichen Menge (wie { − 1,50 , − 1,49 , … , + 1,49 , + 1,50 }.

2. Es existiert nur für eine diskrete Menge von Zeitpunkten

Das Signal ist nicht für irgendeinen Zeitpunkt definiert – die Zeitpunkte sind getrennt und zählbar. Man kann sagen: „Dies ist der erste Zeitpunkt, zu dem das Signal einen bestimmten Wert annimmt, dies ist der zweite Zeitpunkt…“.



Dieses digitale Signal kann somit durch eine Folge von Zahlen, sogenannte **Samples**, dargestellt werden. Ein festes Zeitintervall zwischen den Abtastungen führt zu einer **Signalabtastrate**.

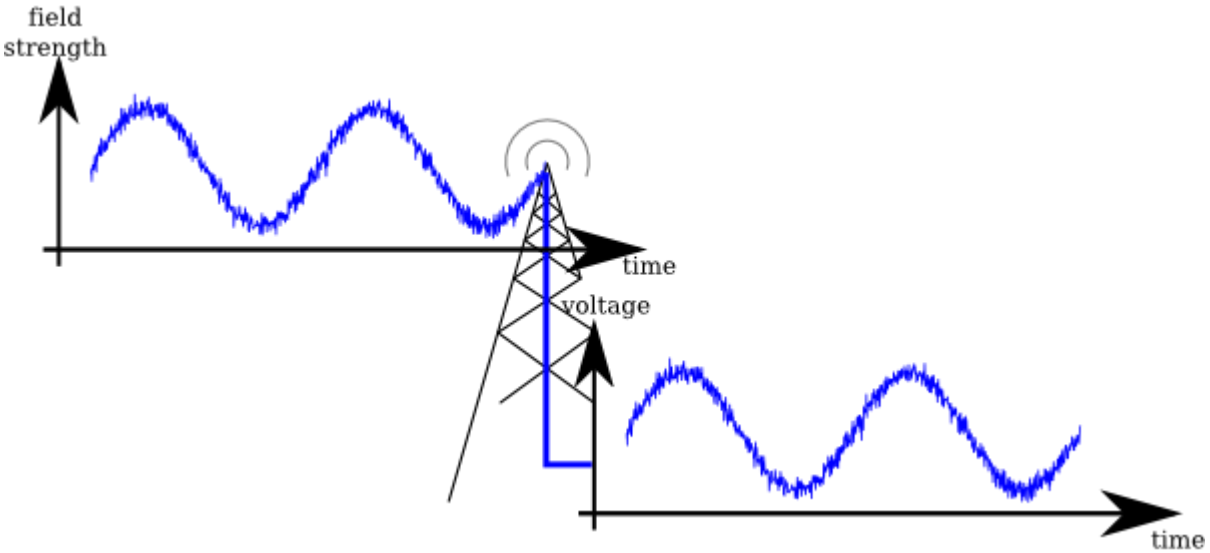
Der Prozess der Erfassung einer physikalischen Größe (Spannung) und deren Umwandlung in digitale Abtastwerte erfolgt durch einen **Analog-Digital-Wandler** (ADC). Das ergänzende Gerät, ein **Digital-Analog-Wandler** (DAC), übernimmt Zahlen von einem digitalen Computer und wandelt sie in ein analoges Signal um.

Da wir nun eine Zahlenfolge haben, kann unser Computer alles damit machen. Es könnte beispielsweise digitale Filter anwenden, es komprimieren, Sprache erkennen oder das Signal über eine digitale Verbindung übertragen.

Anwendung digitaler Signalverarbeitung auf Funkübertragungen

Auf Radiowellen können die gleichen Prinzipien wie für Töne angewendet werden:

Ein Signal, hier elektromagnetische Wellen, kann mit einer Antenne in eine variierende Spannung umgewandelt werden.

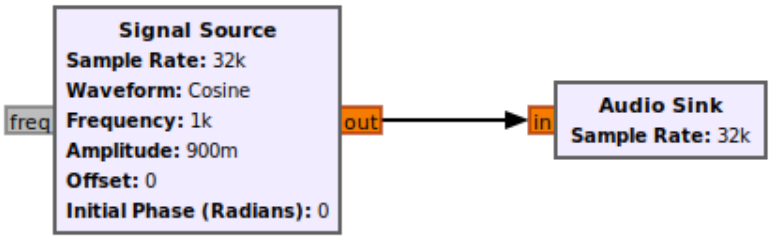


Dieses elektrische Signal liegt dann auf einer Trägerfrequenz, die meist mehrere Megahertz oder sogar Gigahertz beträgt.

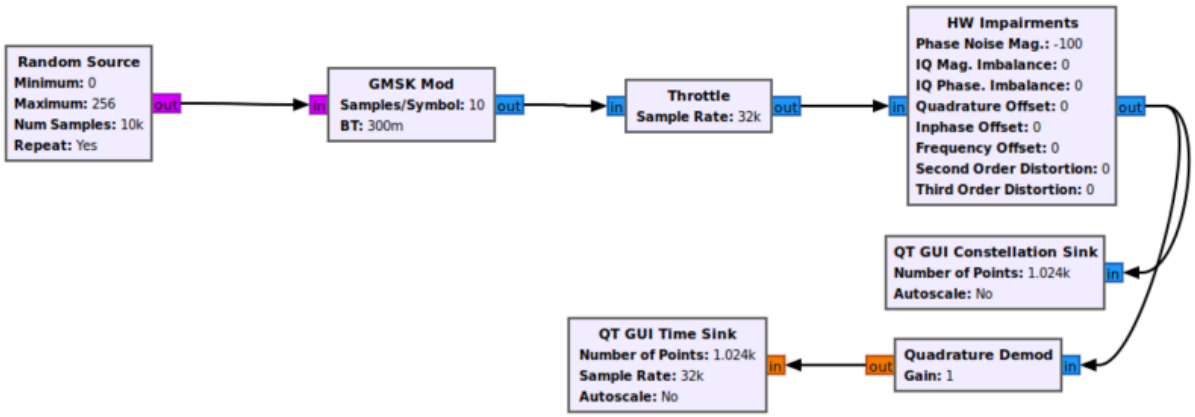
Verschiedene Arten von Empfängern (z. B. Superheterodyne-Empfänger, Direct Conversion, Low Intermediate Frequency Receiver), die als dedizierte Software-Funkperipherie im Handel erhältlich sind, sind für Benutzer bereits verfügbar (z. B. an Soundkarten angeschlossene Amateurfunkempfänger) oder können durch die Umnutzung billig erhältlicher digitaler Fernsehempfänger für Verbraucher erhalten werden (das berüchtigte RTL-SDR-Projekt).

Ein modularer, auf Flussdiagrammen basierender Ansatz zur digitalen Signalverarbeitung:

Um digitale Signale zu verarbeiten, kann man sich die einzelnen Verarbeitungsstufen (Filterung, Korrektur, Analyse, Erkennung ...) einfach als Verarbeitungsblöcke vorstellen, die durch einfache Flussanzeigepfeile verbunden werden können:



Beim Erstellen einer Signalverarbeitungsanwendung wird ein vollständiger Blockgraph erstellt. Ein solcher Graph wird in GNU Radio **Flowgraph** genannt.



GNU Radio ist ein Framework zur Entwicklung dieser Verarbeitungsblöcke und zur Erstellung von Flussdiagrammen, die Funkverarbeitungsanwendungen umfassen.

Als GNU Radio-Benutzer können Sie vorhandene Blöcke zu einem High-Level-Flussdiagramm kombinieren, das etwas so Komplexes wie den Empfang digital modulierter Signale ausführt. GNU Radio verschiebt die Signaldaten automatisch zwischen diesen und veranlasst die Verarbeitung der Daten, wenn sie zur Verarbeitung bereit sind.
GNU Radio wird mit einem großen Satz vorhandener Blöcke geliefert. Einen Index zu allen finden Sie in ***Block Docs***.

https://wiki.gnuradio.org/index.php/Category:Block_Docs

Um Ihnen nur einen kleinen Auszug dessen zu geben, was in einer Standardinstallation verfügbar ist, sind hier einige der beliebtesten Blockkategorien und einige ihrer Unterkategorien aufgelistet:

Waveform Generators

-
  - Constant Source
  - Noise Source
  - Signal Source (e.g. Sine, Square, Saw Tooth)

- **Modulators**
  -
    - AM Demod
    - Continuous Phase Modulation
    - PSK Mod / Demod
    - GFSK Mod / Demod
    - GMSK Mod / Demod
    - QAM Mod / Demod
    - WBFM Receive
    - NBFM Receive

- **Instrumentation** (i.e., GUIs)
  -
    - Constellation Sink
    - Frequency Sink
    - Histogram Sink
    - Number Sink
    - Time Raster Sink
    - Time Sink
    - Waterfall Sink

- **Math Operators**
  -
    - Abs
    - Add
    - Complex Conjugate
    - Divide

    - Integrate
    - Log10
    - Multiply
    - RMS
    - Subtract

- **Channel Models**

    -
    - Channel Model
    - Fading Model
    - Dynamic Channel Model
    - Frequency Selective Fading Model

- **Filters**

    -
    - Band Pass / Reject Filter
    - Low / High Pass Filter
    - IIR Filter
    - Generic Filterbank
    - Hilbert
    - Decimating FIR Filter
    - Root Raised Cosine Filter
    - FFT Filter

- **Fourier Analysis**

- FFT
- Log Power FFT
- Goertzel (Resamplers)
- Fractional Resampler
- Polyphase Arbitrary Resampler
- Rational Resampler (Synchronizers)
- Clock Recovery MM
- Correlate and Sync
- Costas Loop
- FLL Band-Edge
- PLL Freq Det
- PN Correlator
- Polyphase Clock Sync

Mit diesen Blöcken können viele Standardaufgaben wie Normalisierung von Signalen, Synchronisierung, Messungen und Visualisierung erledigt werden, indem einfach der entsprechende Block mit Ihrem Signalverarbeitungsflussdiagramm verbunden wird.

Sie können auch Ihre eigenen Blöcke schreiben, die entweder vorhandene Blöcke mit etwas Intelligenz kombinieren, um zusammen mit etwas Logik neue Funktionen

bereitzustellen, oder Sie können Ihren eigenen Block entwickeln, der mit den Eingabedaten arbeitet und Daten ausgibt.

Somit ist GNU Radio hauptsächlich ein **Framework für die Entwicklung von Signalverarbeitungsblöcken und deren Interaktion**. Es verfügt über eine umfangreiche Standardbibliothek an Blöcken und es stehen zahlreiche Systeme zur Verfügung, auf denen ein Entwickler aufbauen kann. Allerdings ist GNU Radio selbst keine Software, die bereit ist, etwas Bestimmtes zu tun – es ist die Aufgabe des Benutzers, daraus etwas Nützliches zu bauen, obwohl es bereits viele nützliche Arbeitsbeispiele enthält. Betrachten Sie es als eine Reihe von Bausteinen.

https://wiki.gnuradio.org/index.php?title=What_Is_GNU_Radio

Anhang:

Der Name GNU ist ein rekursives Akronym von „GNU's Not Unix“ („GNU ist Nicht Unix“) und soll, um Verwechslungen zu vermeiden, wie das Tier Gnu im Deutschen ausgesprochen werden, nicht wie im Englischen (also nicht wie new). Auch als Logo wurde der Kopf einer afrikanischen Gnu-Antilope gewählt.

Das erste für GNU geschriebene Programm war der Texteditor GNU Emacs von Richard Stallman. Die Arbeit daran begann im September 1984.[22] Anfang 1985 wurde es von Stallman selbst erstmals als benutzbar eingestuft. In dieser Zeit war der Softwarevertrieb über das Internet noch nicht üblich, da Zugänge selten waren. Software wurde stattdessen auf Disketten verkauft.

1991 entwickelte Linus Torvalds, inspiriert durch GNU, einen neuen Kernel: Linux. Dieser wurde 1992 unter der GNU General Public License freigegeben und wurde von einigen Distributoren als Variante zum noch nicht fertiggestellten Systemkernel GNU Hurd eingesetzt.[23] Es ist Linux zu verdanken, dass heute tatsächlich eine Version des GNU-Systems ausführbar ist.[24] Im Zuge zunehmender Popularität wurde diese Variante GNUs fälschlicherweise „Linux“ genannt. Richard Stallman legt daher auf die Bezeichnung **GNU/Linux** wert.[25] (Siehe auch GNU/Linux-Namensstreit.)

https://de.wikipedia.org/wiki/GNU

Als **GNU/Linux-Namensstreit** wird eine Debatte zwischen den Anhängern der Freie-Software-Bewegung und jenen des Open-Source-Lagers darüber bezeichnet, ob Betriebssysteme, die auf dem Linux-Betriebssystemkernel einerseits und den Paketen des GNU-Projekts andererseits basieren, als *Linux* oder als *GNU/Linux* zu bezeichnen sind.

Der Linux-Kernel besteht aus hardwarenaher Software für Scheduling, Multitasking, Gerätetreiber, Speicherverwaltung usw.,[1] die GNU-Pakete bestehen u. a. aus der Shell Bash, den Core Utilities, Compilern wie gcc, Bibliotheken wie glibc und der

Umsetzung sämtlicher Funktionen des Portable Operating System Interface (POSIX) System Application Program Interface (POSIX.1) usw.[2] Gemeinsam ergeben Kernel und GNU-Pakete ein unixartiges Betriebssystem.

Während sich zumeist die kürzere Bezeichnung *Linux* für das Betriebssystem durchgesetzt hat, verwenden manche Projekte, z. B. Debian und Knoppix, die Bezeichnung *GNU/Linux*.

https://de.wikipedia.org/wiki/GNU/Linux-Namensstreit