

SDR

Software-Defined-Radio

SDR, GNURadio & CO





SDR, GNURadio & Co



SDR SEMINAR TEIL 2

11.04.2024



SDR, GNURadio & Co



Inhalt Teil 2

- Wie kann man mit fertiger Software und SDR-Technik experimentieren
- Erste Schritte mit der Software GNURadio
 - Wie gehe ich mit der Blockstruktur in diesem nützlichen Tool um
 - Was muss ich beachten wenn ich Blöcke zu einem Signalweg zusammenfüge
 - Welche klassischen Source- und Sink-Blöcke stehen bei den ersten Entwicklungen zur Verfügung
- Erklärung von GNURadio am Beispiel eines Broadcast-Empfängers
- Erklärung von GNURadio am Beispiel eines einfachen 70cm FM-Transmitters
- Erster Überblick über notwendige Hardware, um einem UKW-Sende-Empfängers aufzubauen
- Erläuterung der nächsten Seminarinhalte laut Milestones



SDR, GNURadio & Co



SDR++ Software



SDR, GNURadio & Co



Wie kann man mit fertiger Software und SDR-Technik experimentieren

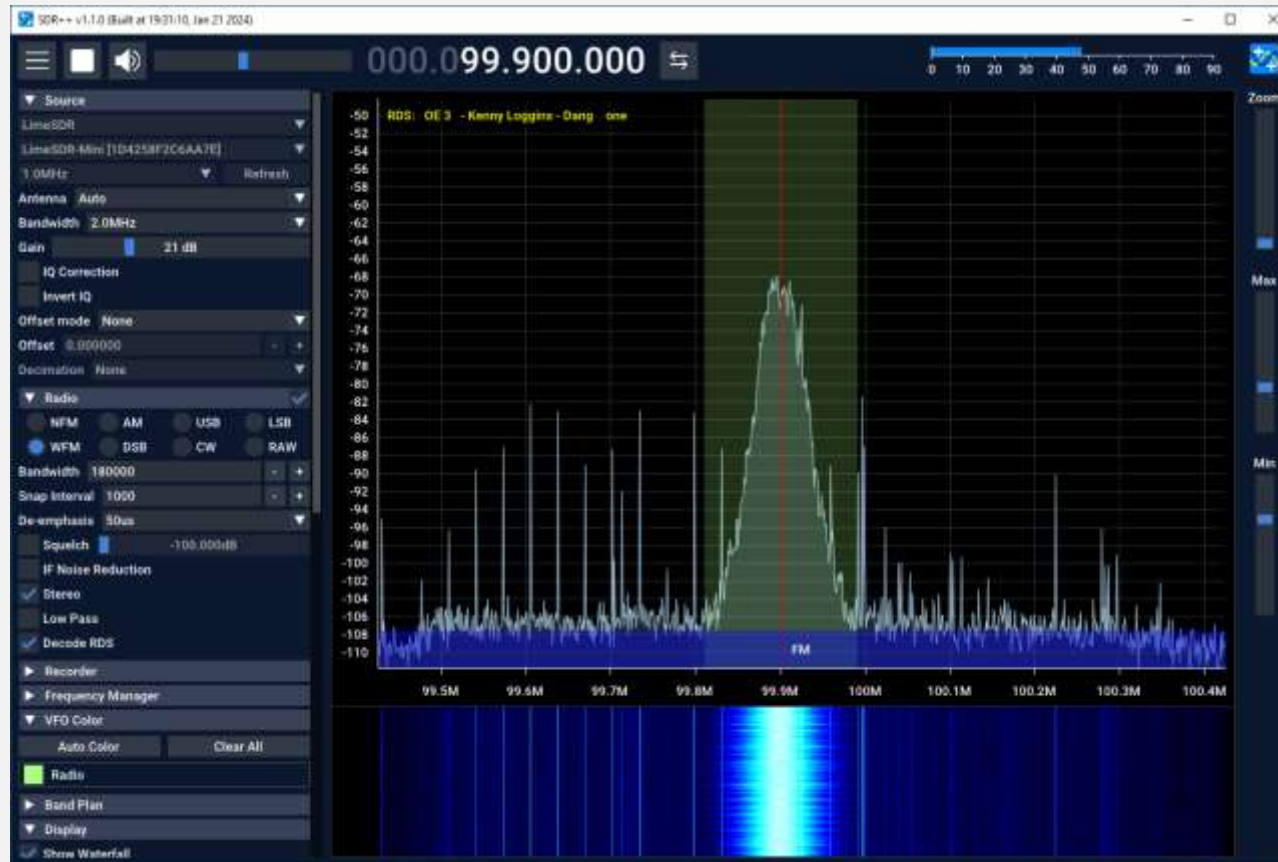
- Damit wir mit einer SDR-Hardware experimentieren können, ohne selbst Software zu programmieren bzw. Software mit modularen Werkzeugen wie GNURadio zu generieren, können wir SDR-Konsolsoftware verwenden.
- Es gibt diverse SDR-Konsolen, welche die gängigste SDR-Hardware abdecken. Häufig verwendet wird:
 - Windows/Linux/MAC:
 - SDR#, SDR++, SDRConsole, HSDR, SDRuno, SDR-RADIO.COM, Linrad, GQRX, u.v.m.
 - Android
 - SDR Touch, Wavesink Plus, RFAalyzer, u.v.m.
- Die verschiedenen SDR-Konsolen unterscheiden sich nicht nur für welches Betriebssystem diese verwendet werden können, sondern auch im Umfang der eingebetteten Funktionen und Module. Die Software **SDRConsole** hat z.B. sehr viele Funktionen, welche das Arbeiten mit dem QO-100 unterstützen mit an Port. Auch das Senden ist mit der **SDRConsole** möglich.
- Für dieses Seminar wurde **SDR++** ausgewählt da es unser Ziel, einen SDR-TRX zu entwickeln, sehr gut unterstützt. So gibt es dafür ein eingebettetes Modul welches M17 dekodieren kann. **SDR++** ist auch derzeit aktuell in guter Weiterentwicklung.



SDR, GNURadio & Co



Wie kann man mit fertiger Software und SDR-Technik experimentieren





SDR, GNURadio & Co



Erste Schritte mit GNURadio Companion



SDR, GNURadio & Co



Erste Schritte mit GNURadio

The screenshot shows the GNU Radio GUI interface. At the top, there is a menu bar with the following items: File, Edit, View, Run, Tools, and Help. Below the menu bar is a toolbar with several icons: a plus sign in a square, a folder icon, a download arrow, a close 'x' icon, a document icon, a monitor icon, a scissors icon, and a copy icon. Below the toolbar is a tab bar with three tabs: 'fmradio_rx_lime_rds', 'fmradio_rx_lime', and 'first'. The 'first' tab is highlighted in red. Below the tab bar are two configuration boxes. The first box is titled 'Options' and contains the following text: 'Title: Not titled yet', 'Output Language: Python', and 'Generate Options: QT GUI'. The second box is titled 'Variable' and contains the following text: 'ID: samp_rate' and 'Value: 32k'.



SDR, GNURadio & Co



Virtuelle Session zu GNURadio

- Aufruf via WEB-Browser
 - <https://workbench.sdr.oevsv.at/guacamole/#>
 - Username: student01
 - Passwort: sdr4oevsv
 - gnu desktop01 auswählen
 - Konsole starten: gnuradio companion (GRC) eingeben
 - Los geht's
- Die Möglichkeit einen RTL-Stick via TCP anzusprechen ist in OE1 in Arbeit.
- Danke an Robert OE6RKE@oevsv.at



SDR, GNURadio & Co



Erste Schritte mit GNURadio

The image shows the GNU Radio interface. In the background, the 'Options' panel displays:
Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

The 'Variable' panel displays:
Variable
Value: 32k

A blue arrow points from the 'Variable' panel to the 'Properties: Options' dialog box in the foreground. The dialog box has three tabs: 'General', 'Advanced', and 'Documentation'. The 'General' tab is selected and contains the following fields:

Property	Value	Type
ID	default	
Title	Audio Funktion	[string]
Author		[string]
Copyright		[string]
Description		[string]
Output Language	Python	
Generate Options	QT GUI	
Run	Autostart	[bool]
Max Number of Output	0	[int]

At the bottom of the dialog box are buttons for 'OK', 'Cancel', and 'Apply'.



SDR, GNURadio & Co



Erste Schritte mit GNURadio

The screenshot shows the GNURadio GUI. On the left, there are two property boxes: 'Options' and 'Variable'. The 'Variable' box shows 'ID: samp_rate' and 'Value: 32k'. A blue arrow points from this box to a larger 'Properties: Variable' dialog box on the right. This dialog box has three tabs: 'General', 'Advanced', and 'Documentation'. The 'General' tab is active and shows the following fields:

Property	Value
ID	samp_rate
Value	48000

At the bottom of the dialog box, there are three buttons: 'OK', 'Cancel', and 'Apply'.



SDR, GNURadio & Co



*untitled - GNU Radio Companion

File Edit View Run Tools Help

fmradio_rx_lime_rds x fmradio_rx_lime x first x **untitled x**

Options
Title: Audio Funktion
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

out

Search: signal

- Core
 - Waveform Generators
 - Signal Source**
 - Measurement Tools
 - Probe Signal Vector
 - Probe Signal
 - Digital Television
 - DVB-T
 - Demod Reference Signals
 - Reference Signals
 - Inspector
 - Conditioning
 - Signal Detector
 - Signal Extractor
 - Signal Separator
 - RADAR
 - Generators
 - Signal Generator CW
 - Signal Generator FMCW
 - Signal Generator FSK
 - Signal Generator Sync Pulse
 - Satellites
 - Deframers
 - SMOG-P Signalling Deframer

<<< Welcome to GNU Radio Companion 3.10.9.2 >>>

Block paths:
C:
.ProgramData\radioconda\Library\share\gnuradio\grc\blocks

..loading: "C:\GitHub\SDR-GNURadio-Seminar\Seminar Teil

ID	Value
Imports	
Variables	
samp_rate	48000



SDR, GNURadio & Co



*untitled - GNU Radio Companion

File Edit View Run Tools Help

fmradio_rx_lime_rds x fmradio_rx_lime x first x untitled x

Options
Title: Audio Funktion
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

Properties: Signal Source

General Advanced Documentation

Output Type: complex

Sample Rate: [real]

Waveform: [dropdown]

Frequency: [real]

Amplitude: [real]

Offset: [complex]

Initial Phase (Radians): [real]

Show Msg Ports: [dropdown] [bool]

Source - out(0):
Port is not connected.

OK Cancel Apply

ID	Value
Imports	
Variables	
samp_rate	48000

<<< Welcome to GNU Radio Companion 3.10.9.2 >>>

Block paths:
C:
ProgramData\radioconda\Library\share\gnuradio\grc\blocks

loading: "C:\GitHub\SDR-GNURadio-seminar\Seminar Teil



SDR, GNURadio & Co



*untitled - GNU Radio Companion

File Edit View Run Tools Help

fmradio_rx_lime_rds x fmradio_rx_lime x first x untitled x

Options
Title: Audio Funktion
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1.75k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

QT GUI Sink
Name:
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): 48k
Update Rate: 10

Core
Instrumentation
QT
fosphor sink (Qt)
QT GUI Bercurve Sink
QT GUI Constellation Sink
QT GUI Eye Sink
QT GUI Frequency Sink
QT GUI Histogram Sink
QT GUI Number Sink
QT GUI Sink
QT GUI Time Raster Sink
QT GUI Time Sink
QT GUI Vector Sink
QT GUI Waterfall Sink
GUI Widgets
QT
QT GUI App Background
QT GUI Fast Auto-Correlator Sink
QT GUI Az-El Plot
QT GUI Check Box
QT GUI Chooser
QT GUI Compass
QT GUI Dial
QT GUI Dial Gauge
QT GUI Distance Radar
QT GUI Message Edit Box
QT GUI Entry

<<< Welcome to GNU Radio Companion 3.10.9.2 >>>

Block paths:
C:
ProgramData\radioconda\Library\share\gnuradio\grc\blocks

loading: "C:\GitHub\SDR-GNURadio-

ID	Value
Imports	
Variables	
samp_rate	48000



SDR, GNURadio & Co



*untitled - GNU Radio Companion

File Edit View Run Tools Help

fmradio_rx_lime_rds x fmradio_rx_lime x first x **untitled x**

Options
Title: Audio Funktion
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1.75k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

QT GUI Sink
Name:
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): 48k
Update Rate: 10

Search: qt

- Core
 - Instrumentation
 - QT
 - fosphor sink (Qt)
 - QT GUI Bercurve Sink
 - QT GUI Constellation Sink
 - QT GUI Eye Sink
 - QT GUI Frequency Sink
 - QT GUI Histogram Sink
 - QT GUI Number Sink
 - QT GUI Sink**
 - QT GUI Time Raster Sink
 - QT GUI Time Sink
 - QT GUI Vector Sink
 - QT GUI Waterfall Sink
- GUI Widgets
 - QT
 - QT GUI App Background
 - QT GUI Fast Auto-Correlator Sink
 - QT GUI Az-El Plot
 - QT GUI Check Box
 - QT GUI Chooser
 - QT GUI Compass
 - QT GUI Dial
 - QT GUI Dial Gauge
 - QT GUI Distance Radar
 - QT GUI Message Edit Box
 - QT GUI Entry

<<< Welcome to GNU Radio Companion 3.10.9.2 >>>

Block paths:
C:
.ProgramData\radioconda\Library\share\gnuradio\grc\blocks

loading: "C:\GitHub\SDR-GNURadio-

ID	Value
Imports	
Variables	
samp_rate	48000



SDR, GNURadio & Co



*untitled - GNU Radio Companion

File Edit View Run Tools Help

fmradio_rx_lime_rds x fmradio_rx_lime x first x **untitled** x

Options
Title: Audio Funktion
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1.75k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

out → in

Name
FFT S
Cente
Band
Updat

Audio Funktion

Frequency Display Waterfall Display Time Domain Display **Constellation Display**

Quadrature

In-phase

■ Data 0

Display RF Frequencies

FFT Size: 1024

Window: Blackman-harris

```
<<< Welcome to GNU Radio Companion 3.10.9.2 >>>
Block paths:
C
ProgramData\radioconda\Library\share\gnuradio\grc\blocks
loading: "C:\Github\SDR-GNURadio-
```

ID	Value
Imports	
Variables	
samp_rate	48000



SDR, GNURadio & Co



*first_step.grc - C:\GitHub\SDR-GNURadio-Seminar\Seminar Teil 2

File Edit View Run Tools Help

fmradio_rx_lime_rds x fmradio_rx_lime x first x first_step x

Options
Title: Audio Funktion
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1.75k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

QT GUI Sink
Name: Analyse Grafik
FFT Size: 1024
Center Frequency (Hz): 1.75k
Bandwidth (Hz): 48k
Update Rate: 5

Audio Sink
Sample Rate: 48k

Properties: Audio Sink

General Advanced Documentation

Sample Rate: [int]

Device Name: [string]

OK to Block: [dropdown]

Num Inputs: [int]

Sink - in(0):
Port is not connected.

OK Cancel Apply

```
void CPU congestion.
```

ID	Value
Imports	
Variables	
samp_rate	48000

```
Executing: C:\ProgramData\radioconda\python.exe -u C:\GitHub\SDR-GNURadio-Seminar\Seminar Teil 2\first_step.py
```

```
>>> Done
```



SDR, GNURadio & Co



Signal Data Types

- Data Types

Types - Color Mapping

Complex Float 64	
Complex Float 32	←
Complex Integer 64	
Complex Integer 32	
Complex Integer 16	
Complex Integer 8	
Float 64	
Float 32	←
Integer 64	
Integer 32	
Integer 16	
Integer 8	
Bits (unpacked byte)	
Async Message	
Bus Connection	
Wildcard	

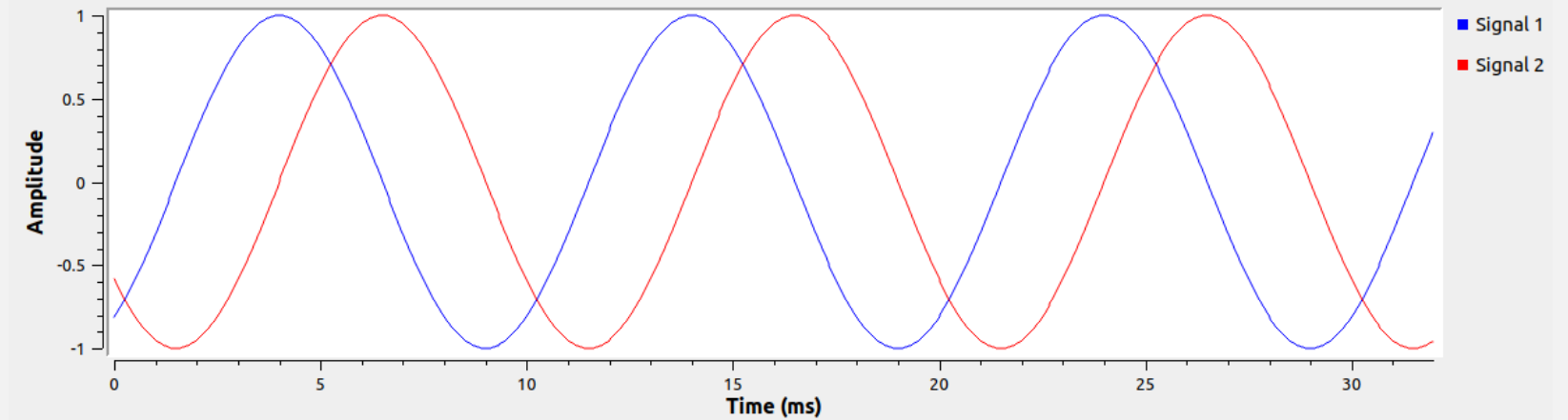
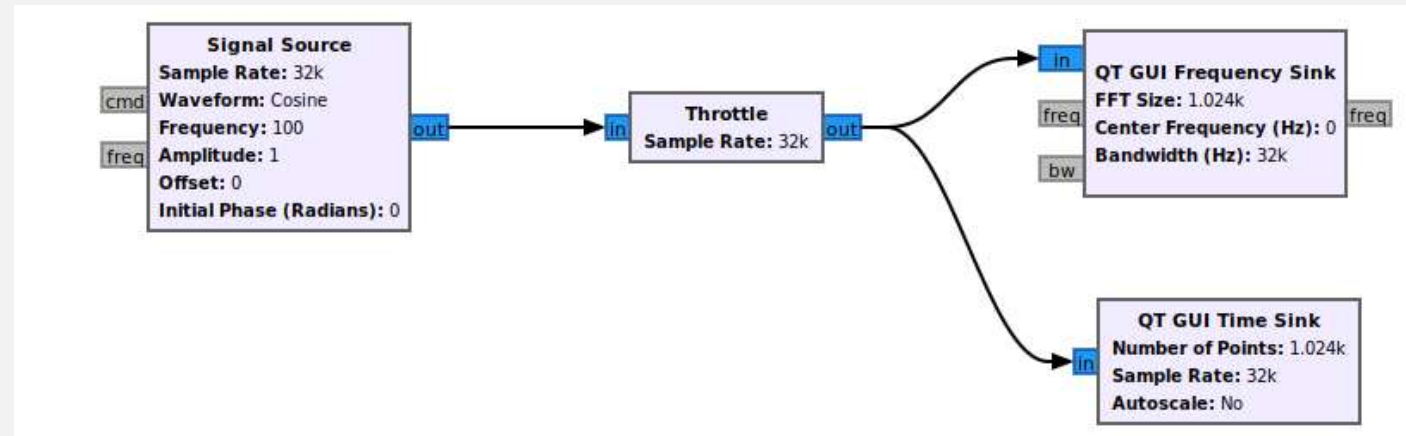


SDR, GNURadio & Co



Signal Data Types

- Data Types
 - Complex Data Type
 - Signal 1 ... real
 - Signal 2 ... complex



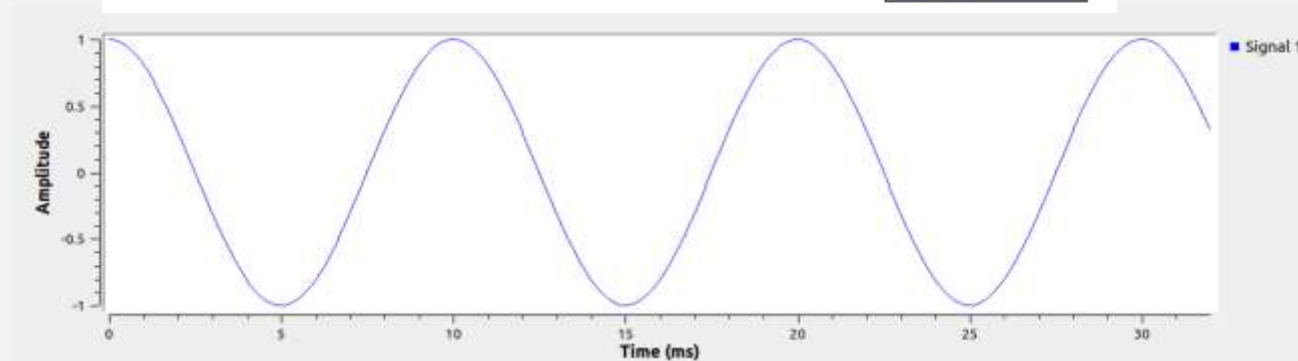
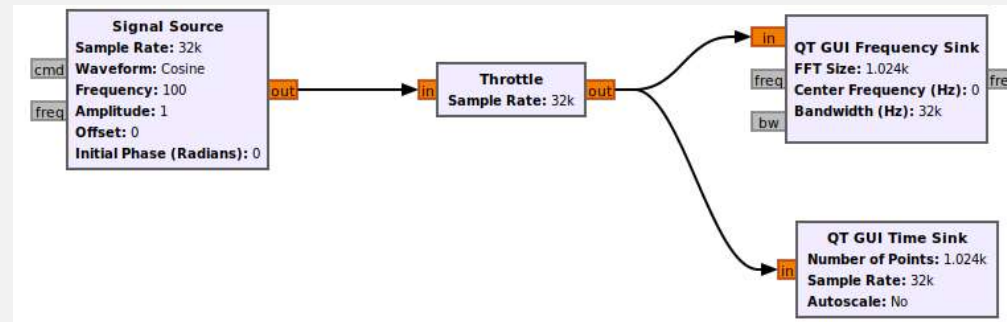
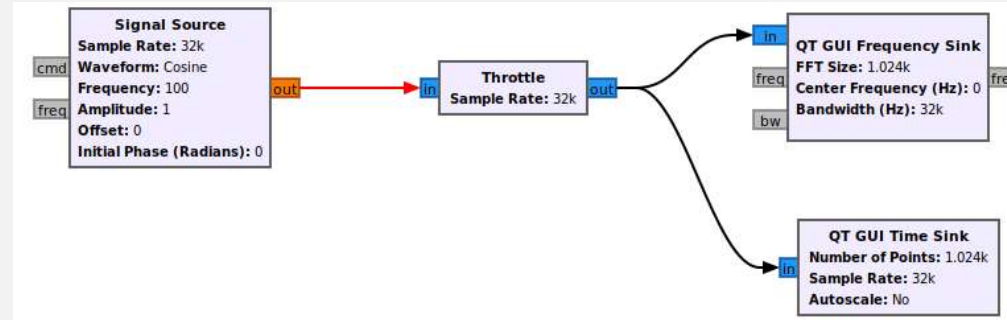


SDR, GNURadio & Co



Signal Data Types

- Data Types
 - Complex Data Type
 - Signal 1 ... real
 - Signal 2 ... complex
 - Float Data Type
 - Signal 1 ... real output



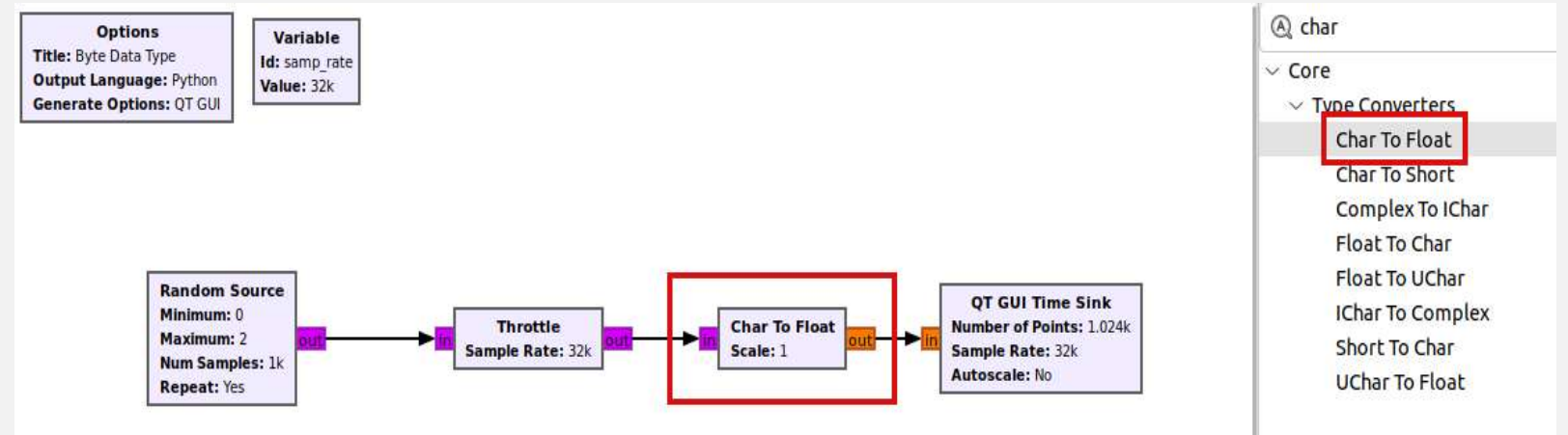


SDR, GNURadio & Co

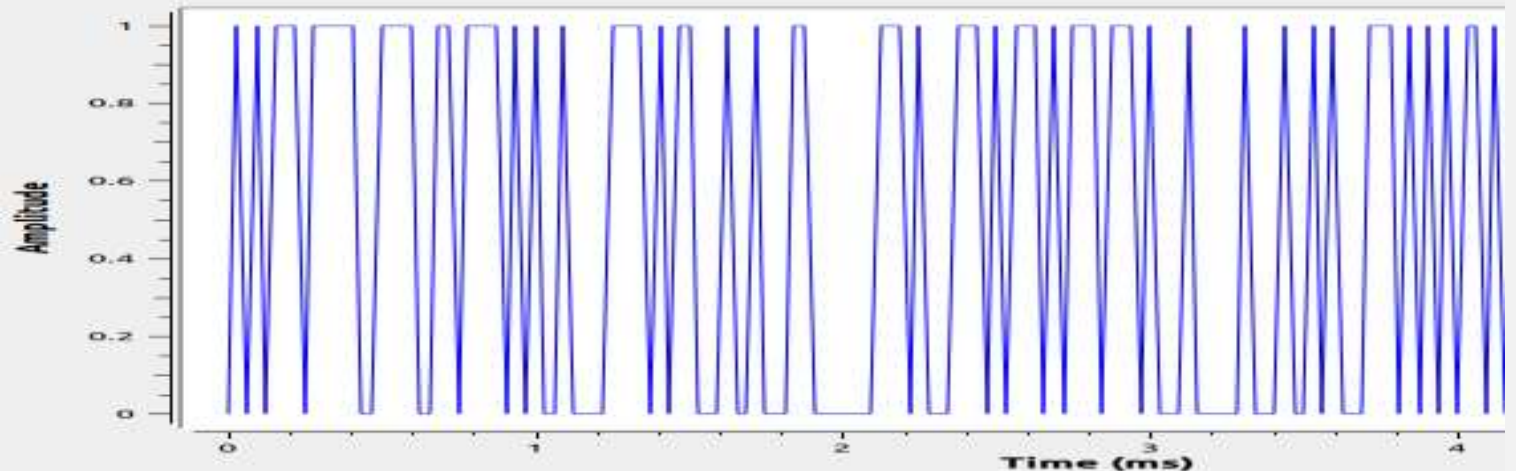


Signal Data Types

- Data Types
 - Complex Data Type
 - Signal 1 ... real
 - Signal 2 ... complex
 - Float Data Type
 - Signal 1 ... real output



- Converting Byte to Float 32





SDR, GNURadio & Co



*first_step.grc - C:\GitHub\SDR-GNURadio-Seminar\Seminar Teil 2

File Edit View Run Tools Help

fmradio_rx_lime_rds x fmradio_rx_lime x first x first_step x

Options
Title: Audio Funktion
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1.75k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

QT GUI Sink
Name: Analyse Grafik
FFT Size: 1024
Center Frequency (Hz): 1.75k
Bandwidth (Hz): 48k
Update Rate: 5

Audio Sink
Sample Rate: 48k
Device Name: default

Properties: Signal Source

General Advanced Documentation

Output Type: float

Sample Rate: samp_rate [real]

Waveform: Cosine

Frequency: 1750 [real]

Amplitude: 1 [real]

Offset: 0 [real]

Initial Phase (Radians): 0 [real]

Show Msg Ports: No [bool]

OK Cancel Apply

```
void CPU congestion.
```

ID	Value
Imports	
Variables	
samp_rate	48000

```
Executing: C:\ProgramData\radioconda\python.exe -u C:\GitHub\SDR-GNURadio-Seminar\Seminar Teil 2\first_step.py
```

>>> Done



SDR, GNURadio & Co



first_step.grc - C:\GitHub\SDR-GNURadio-Seminar\Seminar Teil 2

File Edit View Run Tools Help

fmradio_rx_lime_rds x fmradio_rx_lime x first x first_step x

Options
Title: Audio Funktion
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1.75k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

QT GUI Sink
Name: Analyse Grafik
FFT Size: 1024
Center Frequency (Hz): 1.75k
Bandwidth (Hz): 48k
Update Rate: 5

Audio Sink
Sample Rate: 48k
Device Name: default

Executing: C:\ProgramData\radioconda\python.exe - J C:\GitHub\SDR-GNURadio-Seminar\Seminar Teil 2\first_step.py

audio_windows_sink:warning: waveOut device 'default' was not found, defaulting to WAVE_MAPPER

ID	Value
Imports	
Variables	
samp_rate	48000

Audio Funktion

Frequency Display Waterfall Display Time Domain Display Constellation Display

Amplitude

Time (ms)

real imag

Display RF Frequencies

Window: Blackman-harris

FFT Size: 1024



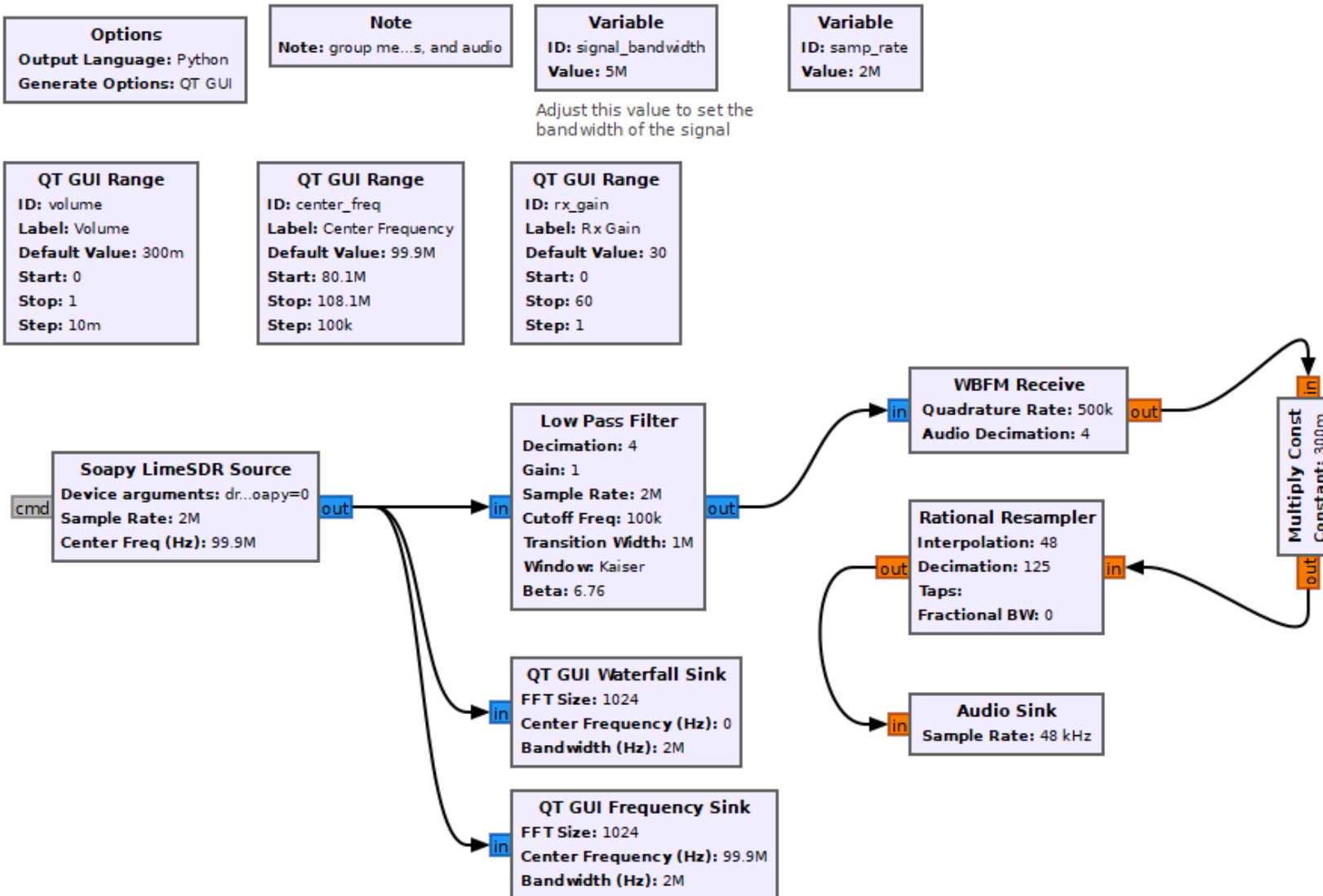
SDR, GNURadio & Co



Broadcast-Empfänger mit GNURadio



SDR, GNURadio & Co





SDR, GNURadio & Co



Options
Output Language: Python
Generate Options: QT GUI

Note
Note: group me...s, and audio

Variable
ID: signal_bandwidth
Value: 5M

Variable
ID: samp_rate
Value: 2M

Adjust this value to set the bandwidth of the signal

QT GUI Range
ID: volume
Label: Volume
Default Value: 300m
Start: 0
Stop: 1
Step: 10m

QT GUI Range
ID: center_freq
Label: Center Frequency
Default Value: 99.9M
Start: 80.1M
Stop: 108.1M
Step: 100k

QT GUI Range
ID: rx_gain
Label: Rx Gain
Default Value: 30
Start: 0
Stop: 60
Step: 1

Properties: QT GUI Range

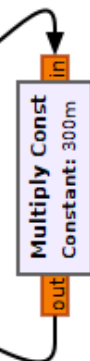
General	Advanced	Documentation
ID	volume	
Label	Volume	[string]
Type	float	
Default Value	0.3	[float]
Start	0	[float]
Stop	1	[float]
Step	0.01	[float]
Widget	Knob	
Minimum Length	200	[int]

OK Cancel Apply

Properties: QT GUI Range

General	Advanced	Documentation
ID	rx_gain	
Label	Rx Gain	[string]
Type	int	
Default Value	30	[int]
Start	0	[int]
Stop	60	[int]
Step	1	[int]
Widget	Counter + Slider	
Minimum Length	200	[int]

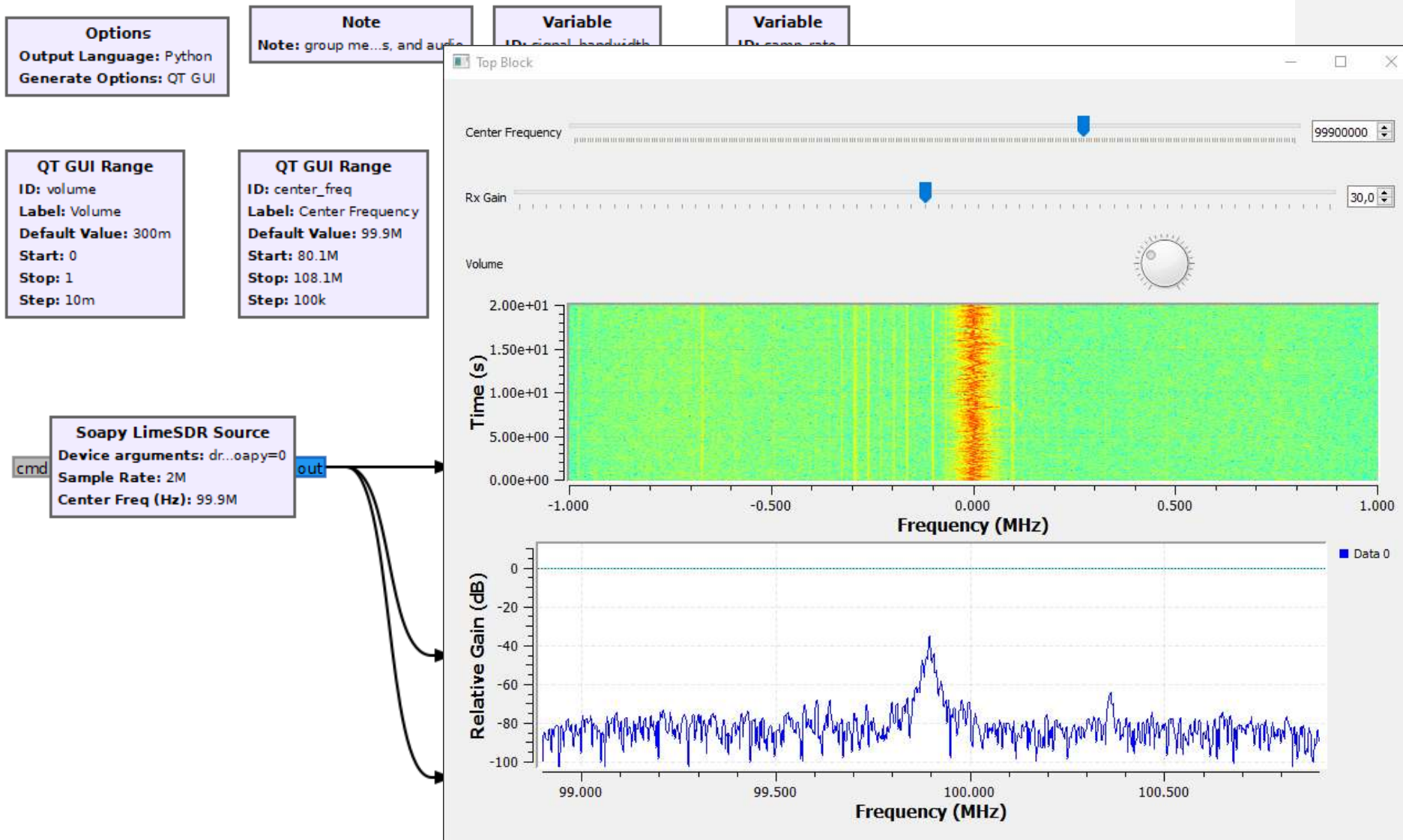
OK Cancel Apply



Center Frequency (Hz): 99.9M
Bandwidth (Hz): 2M

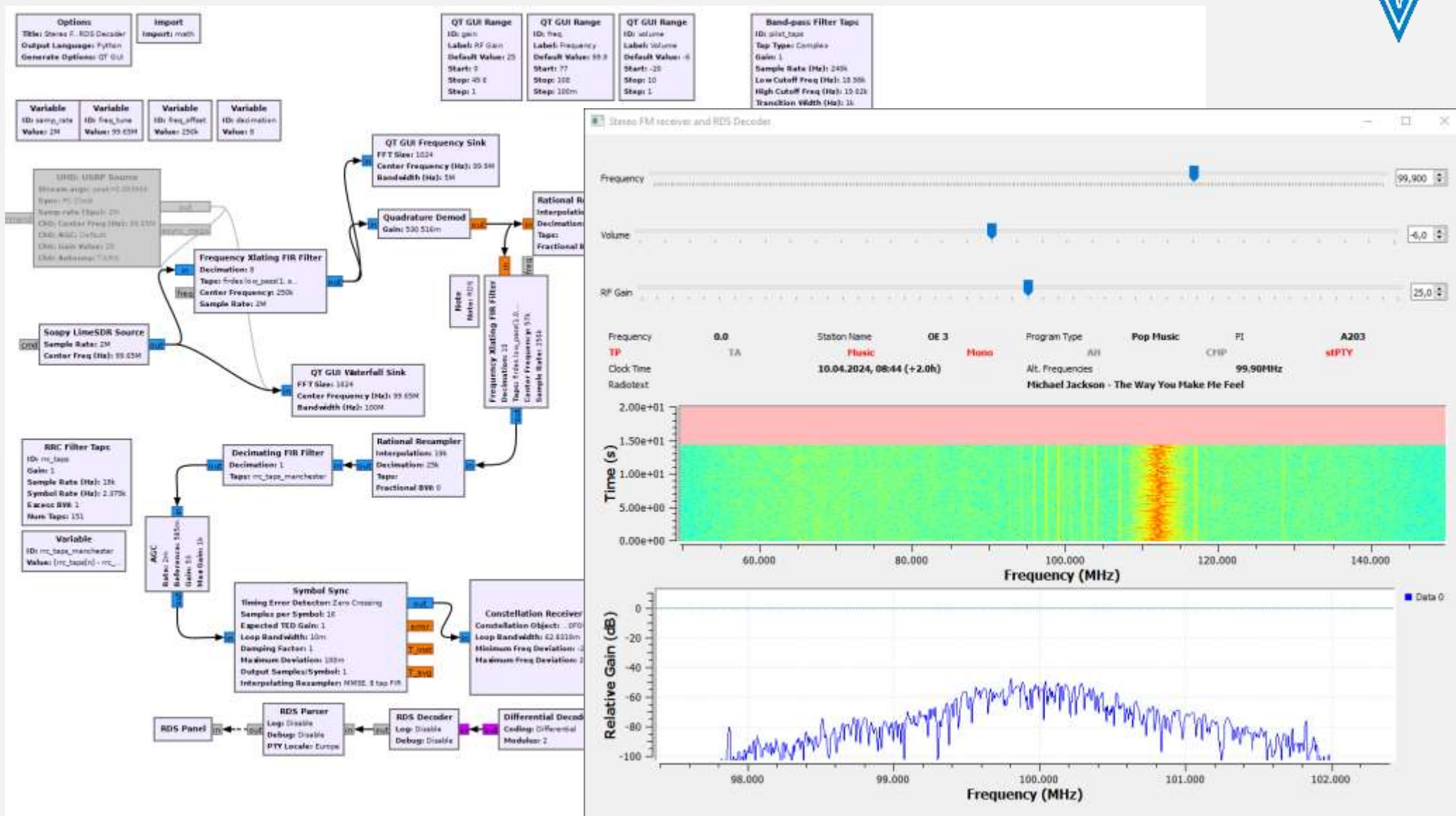


SDR, GNURadio & Co





SDR, GNURadio & Co





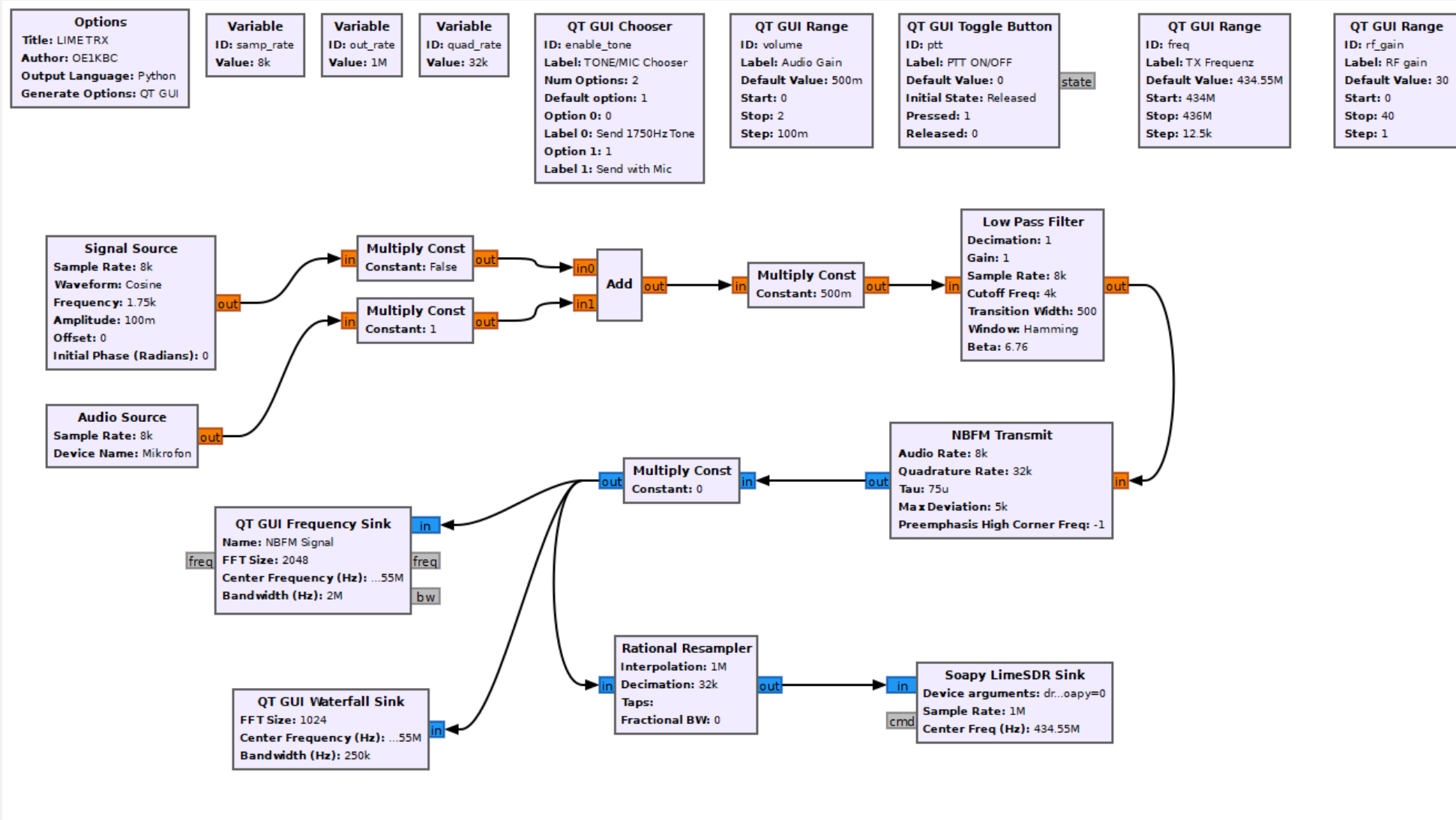
SDR, GNURadio & Co



FM-Transmitter für 70cm mit GNURadio



SDR, GNURadio & Co





SDR, GNURadio & Co



Options
Title: LIMETR
Author: OE1KBC
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 8k

Variable
ID: out_rate
Value: 1M

Variable
ID: quad_rate
Value: 32k

Signal Source
Sample Rate: 8k
Waveform: Cosine
Frequency: 1.75k
Amplitude: 100m
Offset: 0
Initial Phase (Radians): 0

Audio Source
Sample Rate: 8k
Device Name: Mikrofon

Multiply Const
Constant: False

Multiply Const
Constant: 1

QT GUI Frequency Sink
Name: NBFM Signal
FFT Size: 2048
Center Frequency (Hz): ...55M
Bandwidth (Hz): 2M

QT GUI Waterfall Sink
FFT Size: 1024
Center Frequency (Hz): ...55M
Bandwidth (Hz): 250k

LIME TRX

NBFM Signal

Relative Gain (dBm)

Frequency (MHz)

Audio Gain

RF gain: 30.0

PTT ON/OFF

TX Frequenz: 434550000.0

TONE/MIC Chooser:
 Send 1750Hz Tone
 Send with Mic

Time (s)



SDR, GNURadio & Co



UKW-Sende-Empfänger Konzept



UKW-Sende-Empfänger Konzept

- RX
 - Bandfilter 2m/70cm
 - LNA (Vorverstärker) ca. 10-20 dB
- TX
 - Bandfilter 2m/70cm oder schaltbar
 - PA (Treiberstufe) ca. 20-30 dB



SDR, GNURadio & Co



UKW-Sende-Empfänger Konzept

Beispiele:





SDR, GNURadio & Co



Vorschau SDR, GNURadio & Co Teil 3



SDR, GNURadio & Co



- Milestones der Seminar-Reihe
 - Wir werden gemeinsam
 - ein einfaches Sende-/Empfangsgerät mit einer digitalen Modulation und Verwendung von GNURadio designen
 - Front-Ends aussuchen, welche das Umsetzen eines Transceivers auf einem Raspberry-Pi ermöglichen
 - Das Ziel, einen in der Praxis einsetzbaren UKW-SDR-Transceiver aufzubauen, weiter anpeilen.



SDR, GNURadio & Co



DOWNLOAD



SDR, GNURadio & Co



Index

- GNURadio Flow Charts (Beispiele aus dem Vortrag)
 - SDR, GNURadio & Co Seminar
<https://github.com/oe1kbc/SDR-GNURadio-Seminar/tree/main/Seminar%20Teil%202>
- SDR-Konsolen
 - SDR++ (Nightly build)
<https://github.com/AlexandreRouma/SDRPlusPlus/releases/tag/nightly>
 - SDRConsole (3.3)
<https://www.sdr-radio.com/download#Release>
- GNURadio (v3.10)
<https://wiki.gnuradio.org/index.php/InstallingGR>



SDR, GNURadio & Co



Literatur

The perfect HF Receiver. How would it look like today?

Prof. Dr. Ing. habil. Dr. h. c. mult. Ulrich L. Rohde, Univ. of Armed Forces Munich
Dipl. Ing. Univ. Thomas Boegl, Rohde & Schwarz Munich (2022)

SDR Grundlagen und Anwendungen im Amateurfunk

Ing. Reinhold Autengruber OE5RNL (2010)



Amateurfunk und Citizen Science

Wir wollen es wissen! Wir alle sind Forschung!



- **IceBird-Talk**
ÖVSV Landesverbandes Wien
Ing. Kurt Baumann, OE1KBC
 - Unterstützung von Forschung und Entwicklung
 - Aus- und Weiterbildung im Funkwesen
 - Projekte planen und verwirklichen

<https://oe1.oevsv.at>

