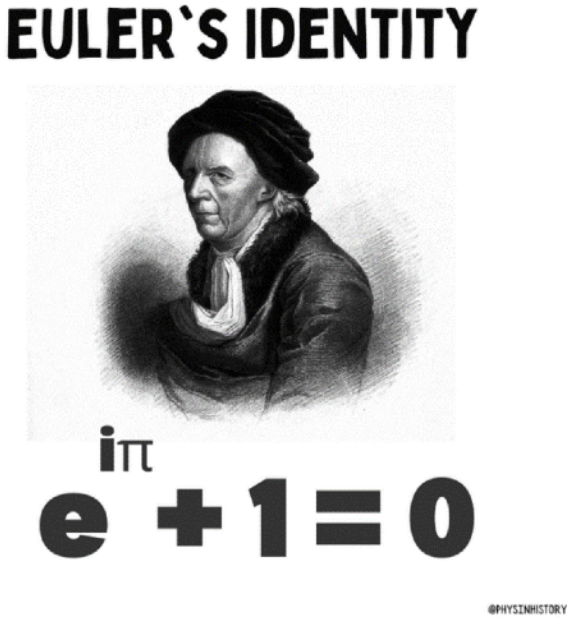
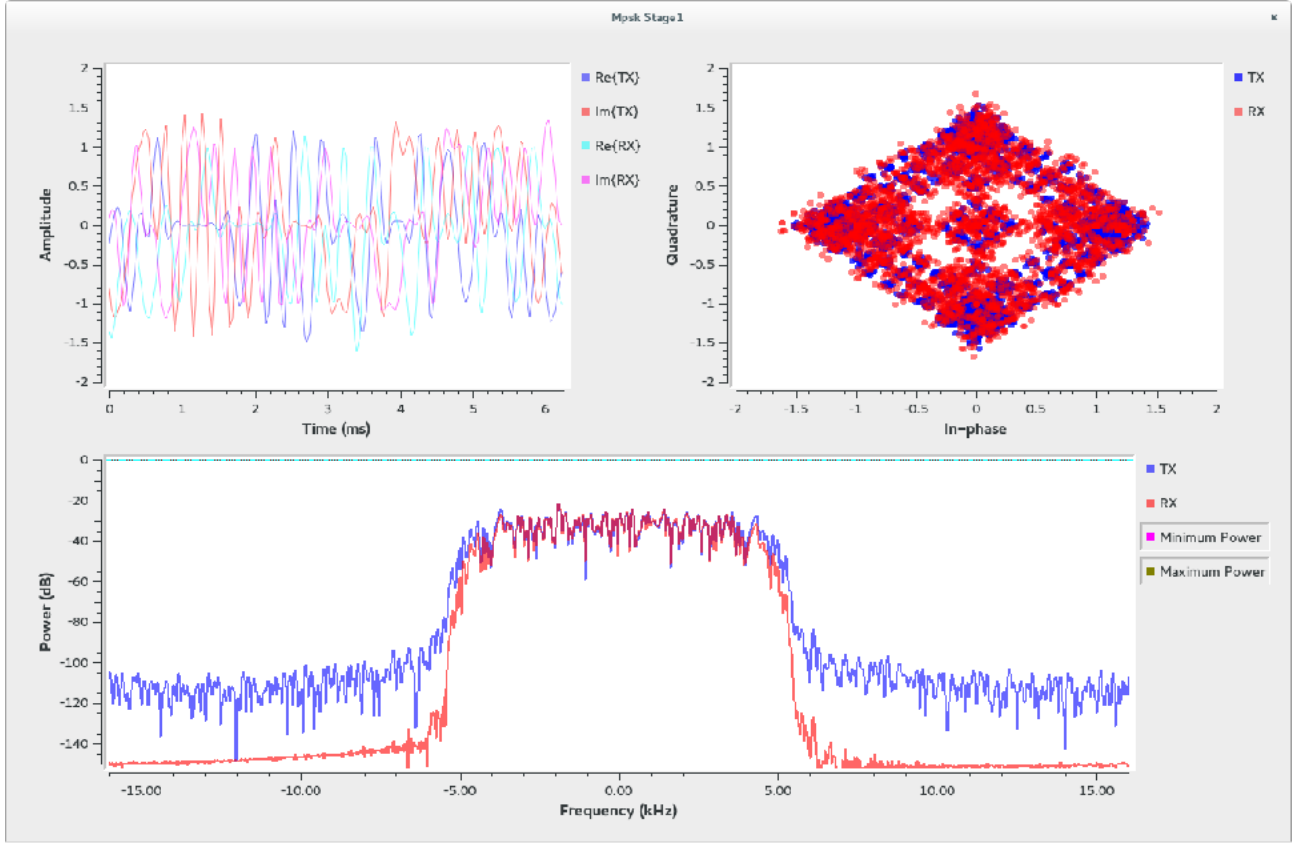


Software Defined Radio

Dr. Karsten Schmidt | Hamburg 37c3| Version 0.0.8



About me

Karsten Schmidt

- Studied Electrical Engineering
- Call sign DG1VS
- PhD at TU Dresden
- 4 years software development
  - Domain of mobile communication
  - Core network technology
- Now since many years in the automotive industry
- Different lectures since many years (Networking, Security, OS)
- Some weird hobbies

dg1vs@darc.de

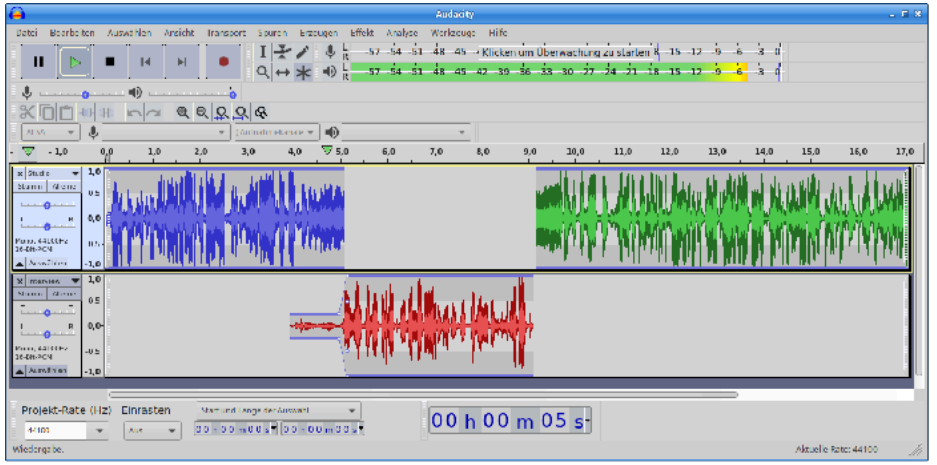
@dg1vs

Private pictures

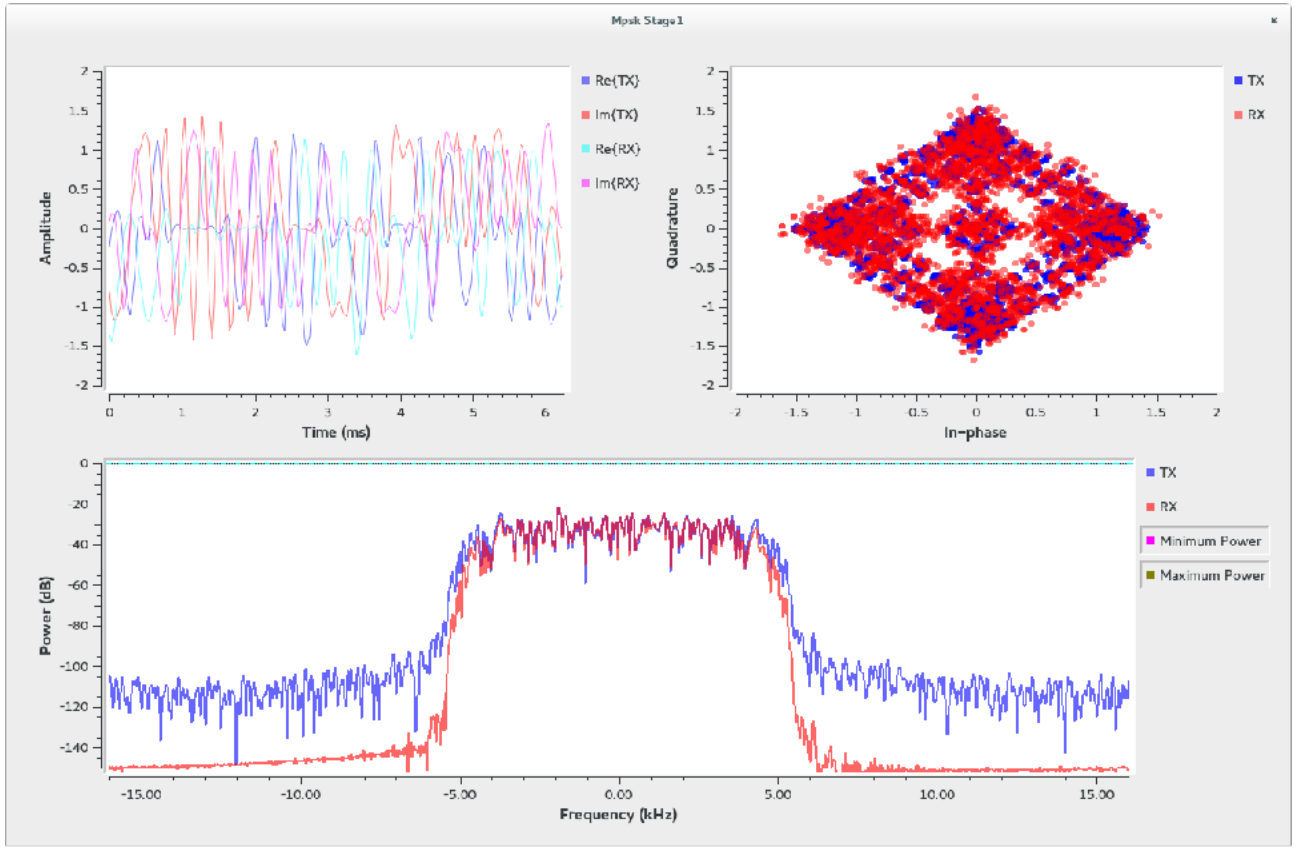
Why this presentation

1. If I do not understand something, I’m trying to prepare a presentation/lecture und I’m also trying to explain others.
2. I din’t understand why a Tayloe-Mixer has two ADC and my HiQSDR has only one, but both providing I/Q-Signals.
3. To help others to start.

SDR – What the fuck?



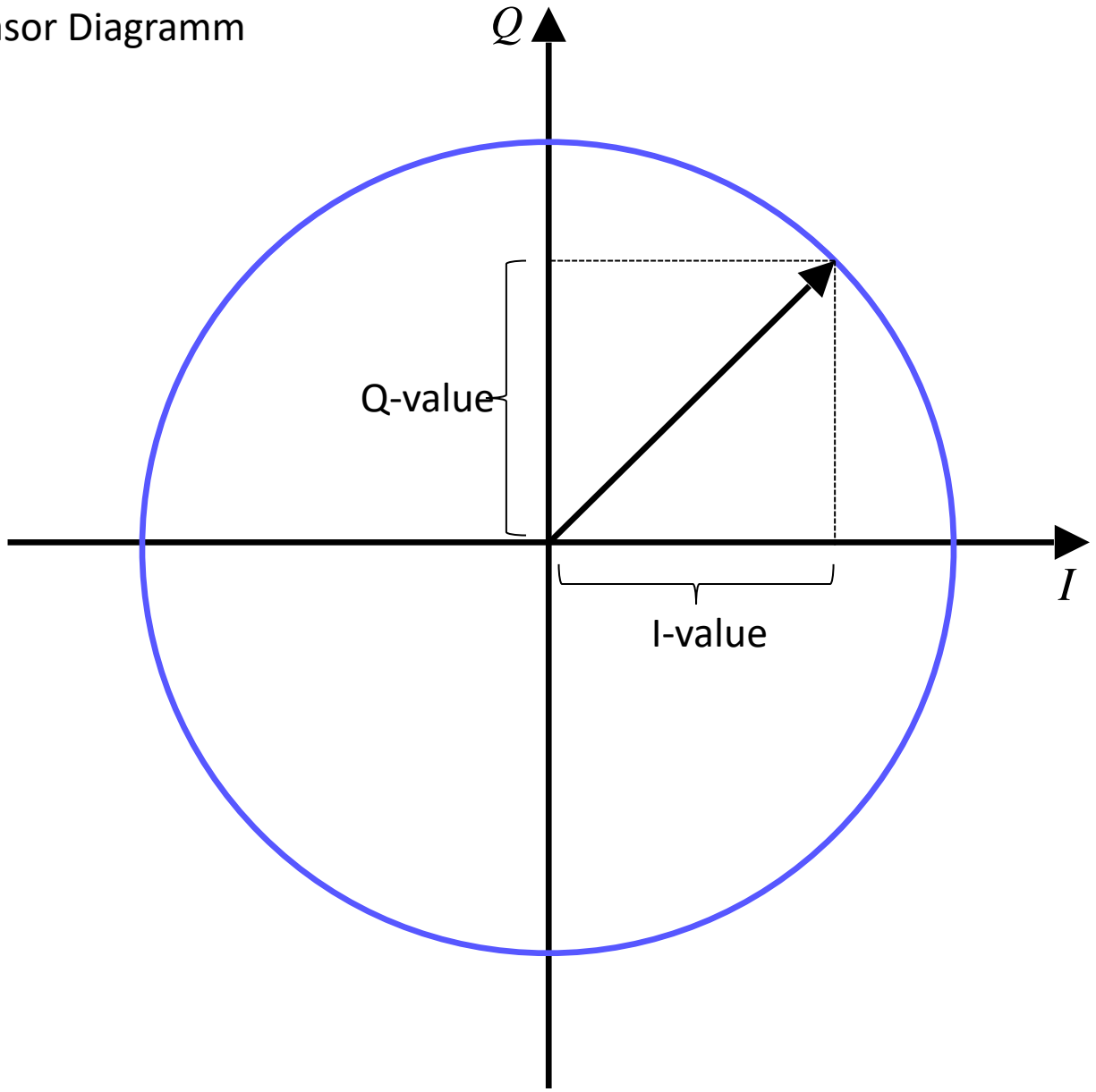
SDR = Hardware + Software



SDR = Hardware + Software

I/Q Signals

Phasor Diagramm



SDR

- Software Defined Radio (SDR) is a general term referring to any radio design that uses mostly software and some controlling software to “define” that radio’s operation.

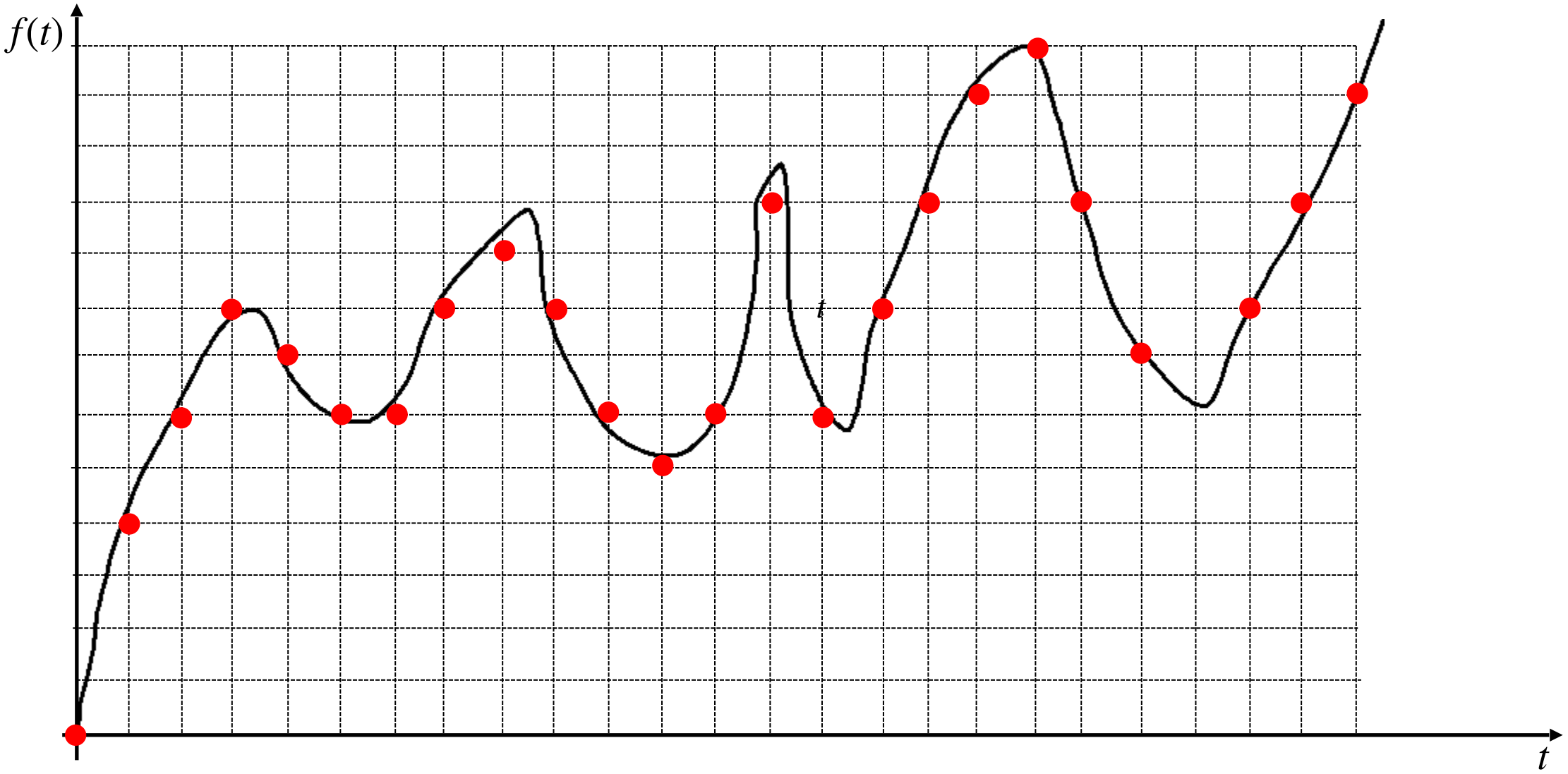
- The biggest reason to have a Software Defined Radio is the flexibility it offers the user.
- Works with I/Q Signals
- Modes of operation can be changed to accommodate new communications technologies
- All these functions are controlled in Software, rather than Hardware, making changes simpler (no new filters/hardware demodulators required- the code takes care of it)
- Filtering can easily be changed, depending on the needs

Motivation of SDR

- **Flexibility and adaptability:** SDR systems allow radios to be changed or improved through software upgrades rather than having to replace hardware. This enables rapid adaptation to new standards or operating modes.
- **Cost efficiency:** The ability to operate different radio standards and frequency bands with the same hardware reduces the need to own and maintain multiple specialized devices.
- **Multifunctionality:** An SDR can perform multiple functions simultaneously, such as voice communication, data transmission and signal analysis, which would not be possible in traditional hardwired systems.
- **Easier implementation of new technologies:** New transmission techniques and standards can be implemented through software updates without having to change the hardware, which is especially beneficial in the fast-paced world of wireless technologies.
- **Improved performance and capacity:** Digital signal processing techniques used in SDR systems can enable higher performance, better signal quality and more efficient use of spectrum.
- **Compatibility and interoperability:** SDR systems can be configured to be compatible with a variety of communication standards and networks, promoting interoperability between different systems and organizations.
- **Research and development:** SDR facilitates research and development of new communication techniques and algorithms, as ideas can be tested quickly and cost-effectively through software changes rather than having to build new hardware.
- **Expandability and future-proofing:** Because the technology is primarily software-based, SDR systems can be easily expanded and updated to meet future requirements and standards.

Signals

A Random Signal



Problems with Sampling

- An ADC/DAC is not "infinitely fast“
  - Introduction of the term sampling rate
- An ADC/DAC does not have an arbitrarily high resolution
  - Common resolutions: 8-bit - 16-bit (256 - 65536 steps related to reference voltage)

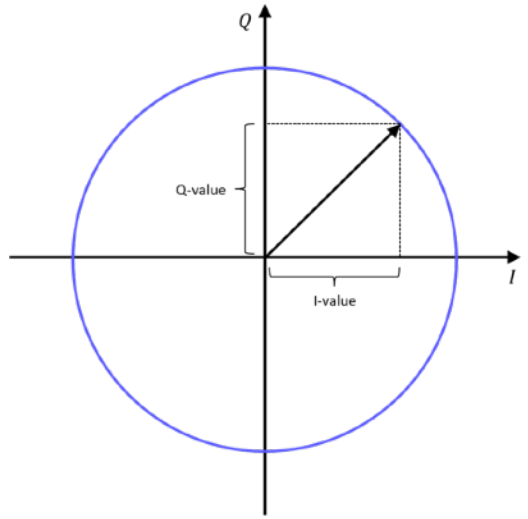
- Rule of thumb
  - Sample rate ↑ and resolution ↑ price ↑

- Depending on the solution of the SDR data needs to be transferred to the PC
  - Interface
  - Data rate
  - Latency

Problems with Sampling

- OK, an SDR converts signals between the analogue and digital domain using ADCs and DACs and does so at a specific sampling rate or
- But wait a minute! That's not all!
- Correct phase unknown:
  - e.g.: $sin(x)$ or $sin(-x)$?
- Signal power cannot be determined precisely
  - Problems in signal processing!

- Solution
- The I/Q converter:
- I/Q = In-phase/Quadrature
- Splitting of the input signals into two signals
- Phase information remains



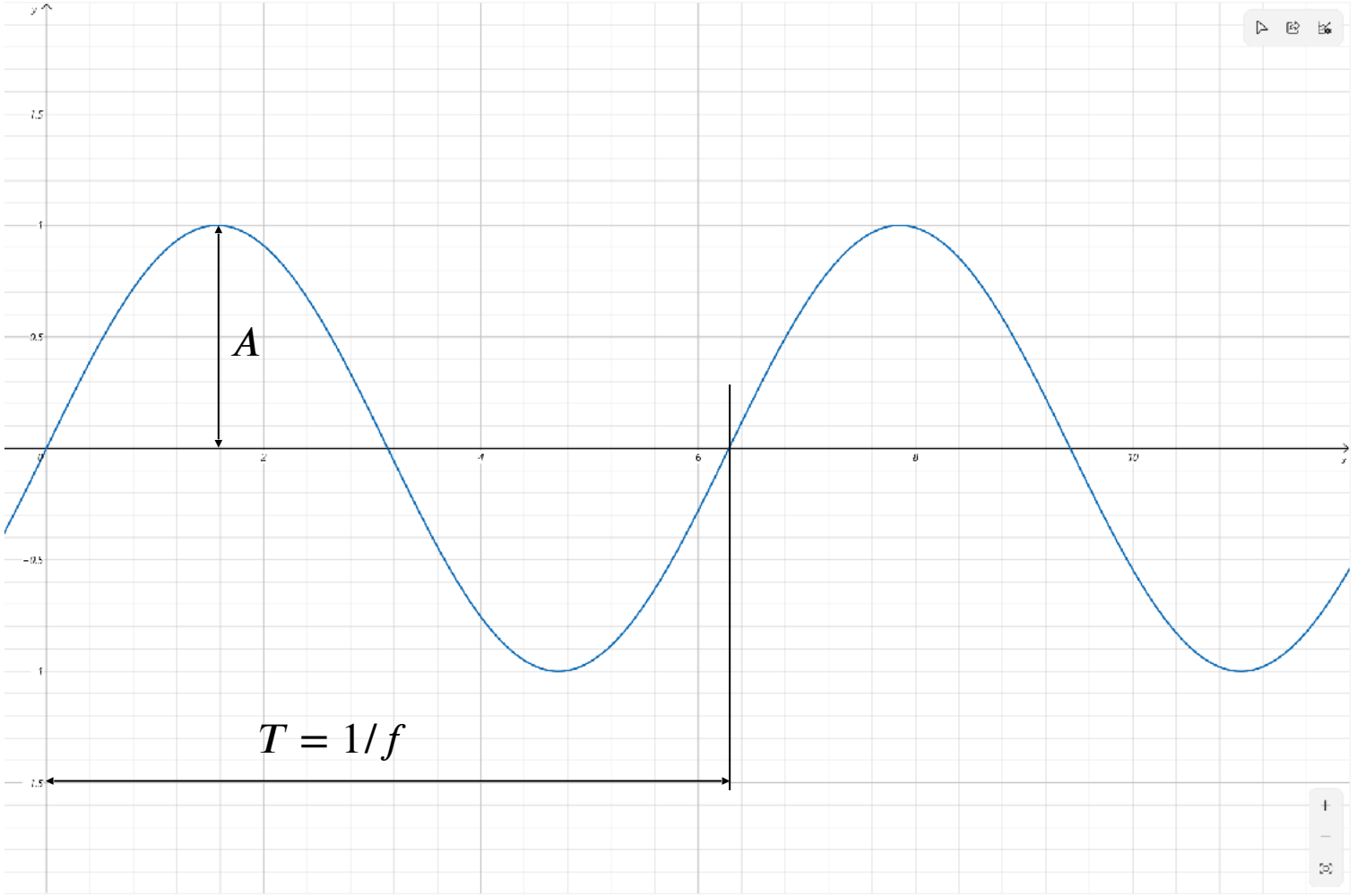
Basic of I/Q-Signals

Basic of I/Q-Signals

- Fundamental Concepts
- Just a Simple Sine Wave
- AM
- I/Q-Signals
- Modulation/Demodulation with I/Q-Signals

Sine Wave

- $s(t) = A \cdot sin(2\pi ft + \phi)$
- A - Peak Value
- f - Frequency
- t - Time
- ϕ - Phase Shift

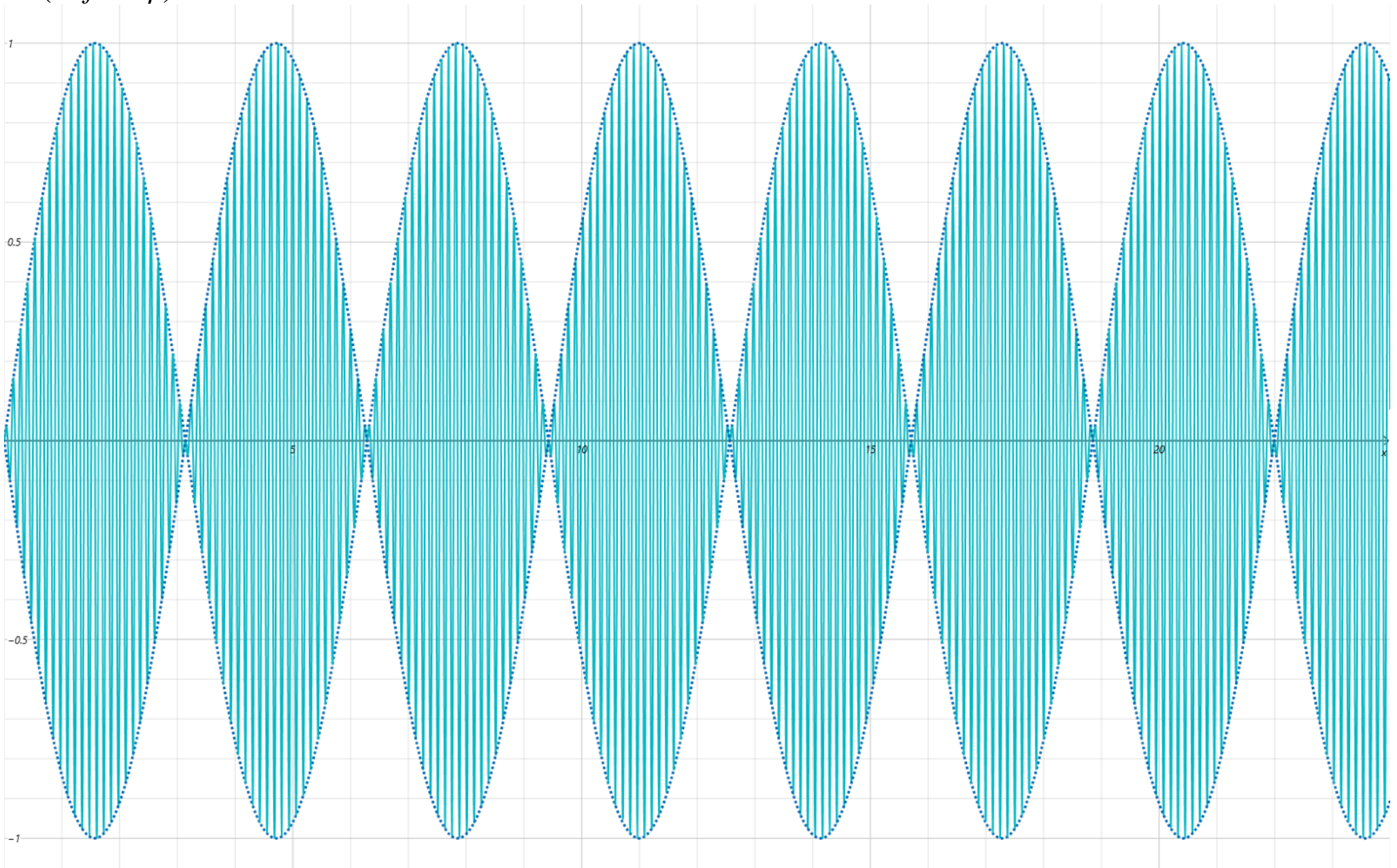


Modulation

- $s(t) = A \cdot sin(2\pi ft + \phi)$
- Modulation changes either
  - A → AM
  - f → FM
  - ϕ → PM
- or combination of three 3 values

AM

$$s(t) = A(t) \cdot sin(2\pi ft + \phi)$$

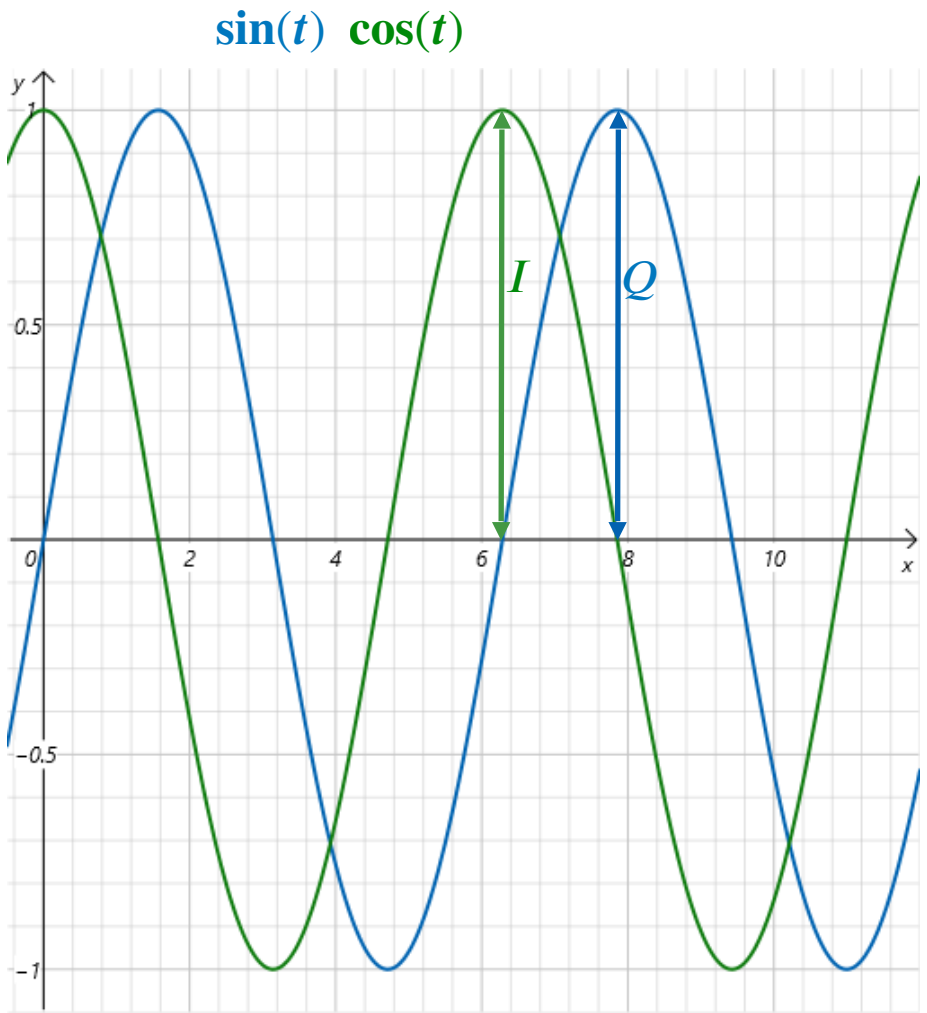


I/Q-Signals

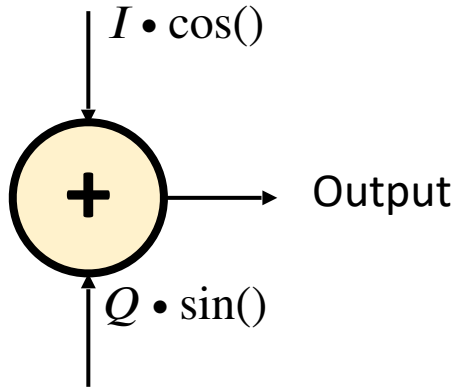
- I/Q signals, short for **I**n-phase and **Q**uadrature signals, are a fundamental concept in the field of signal processing and communications
- Two Signal are Quadrature signals, when between them a differences of 90° exist

- $sin(t)$and $cos(t)$ are in Quadrature

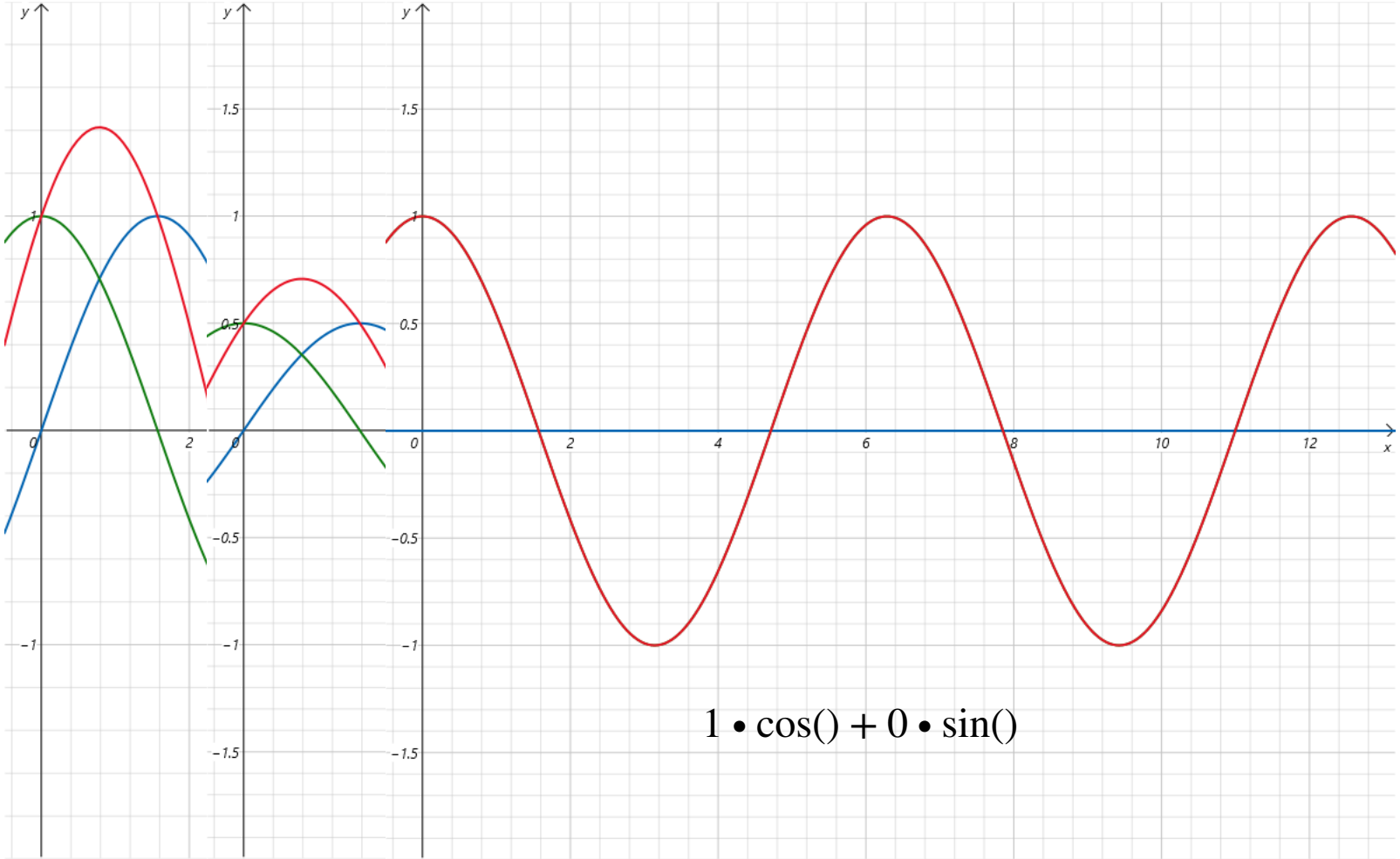
- Just as an rule
  - The amplitude of the In-Phase" Signal = I
    - $I = cos(2\pi ft)$
  - The amplitude of the In-Phase" Signal = Q
    - $Q = sin(2\pi ft)$



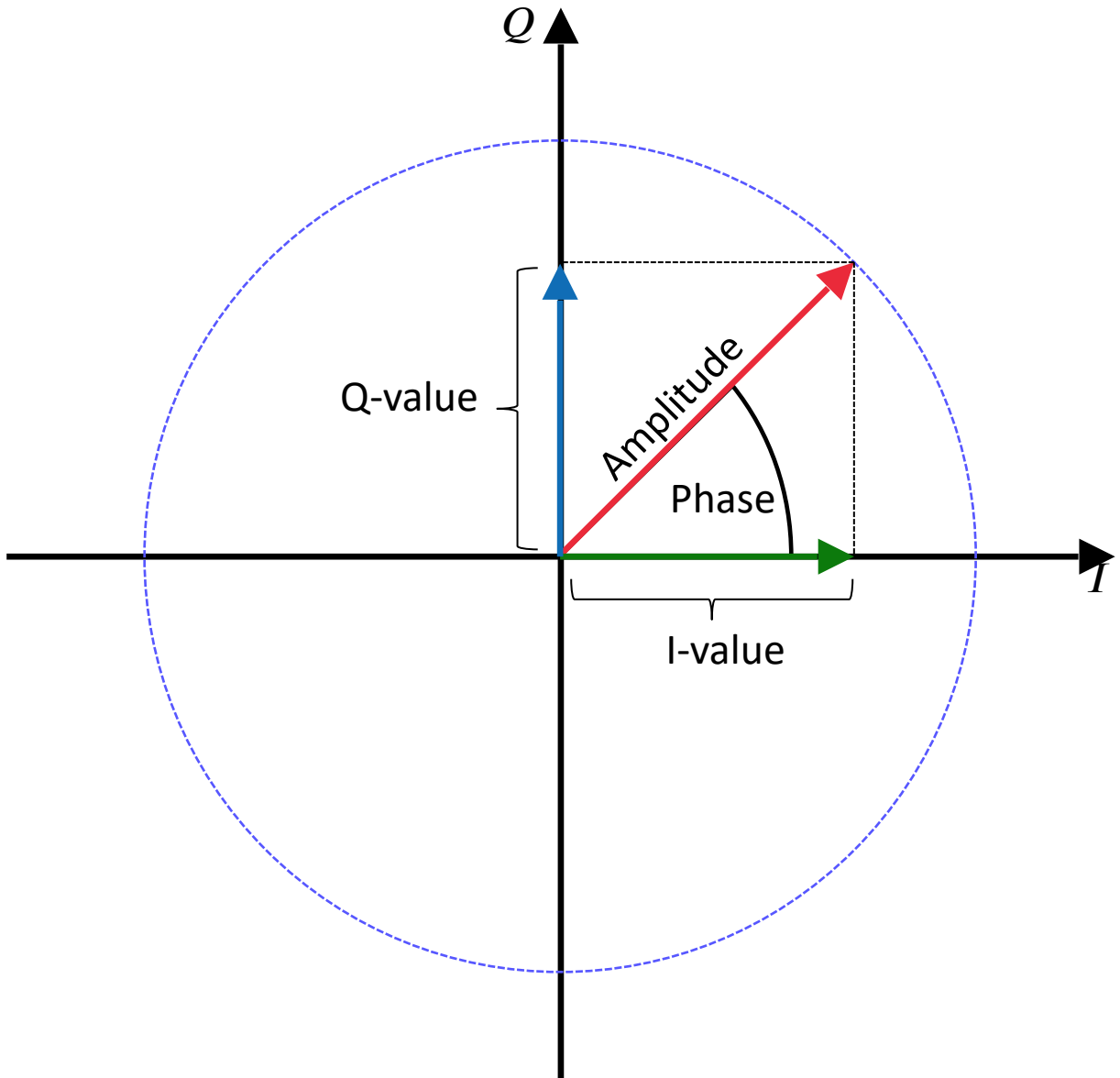
Adding with I/Q-Signals

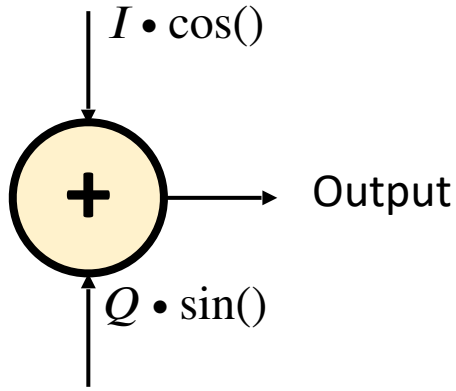


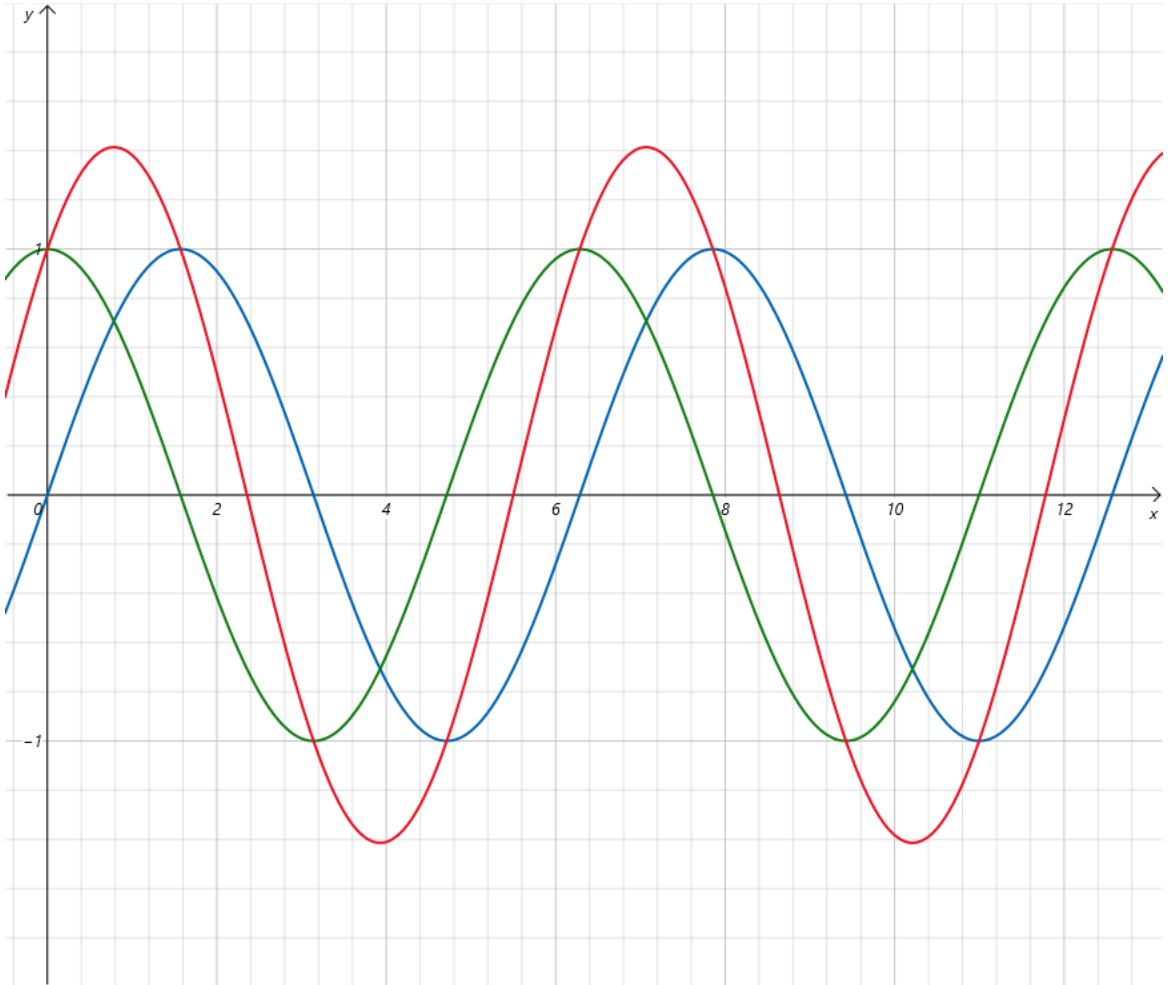
- $I(t) = \hat{I}(t) \cdot cos(2\pi ft)$ and $Q = \hat{Q}(t) \cdot cos(2\pi ft)$
- f constant
- Lets modify the amplitude and see what happens



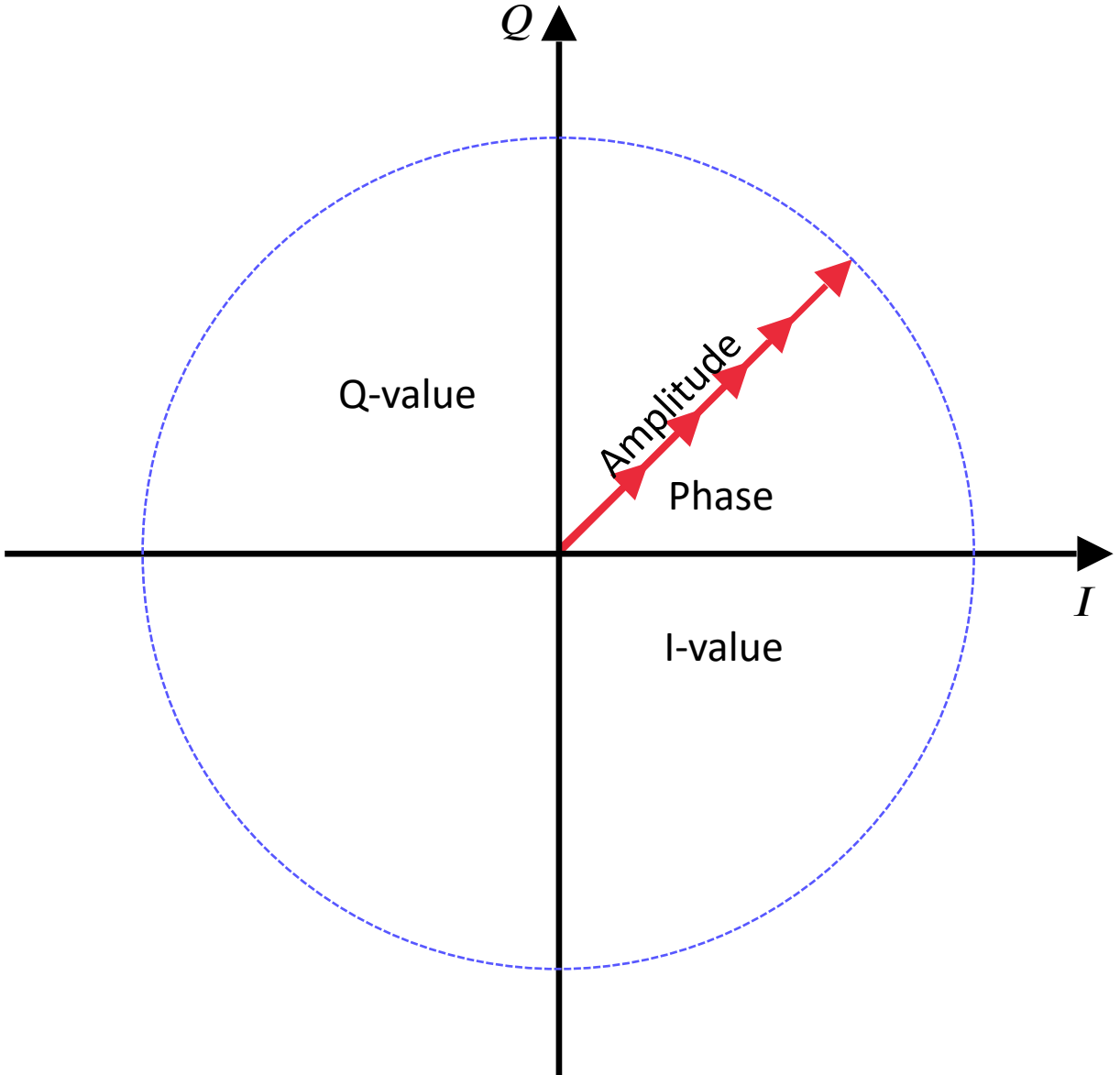
Adding with I/Q-Signals



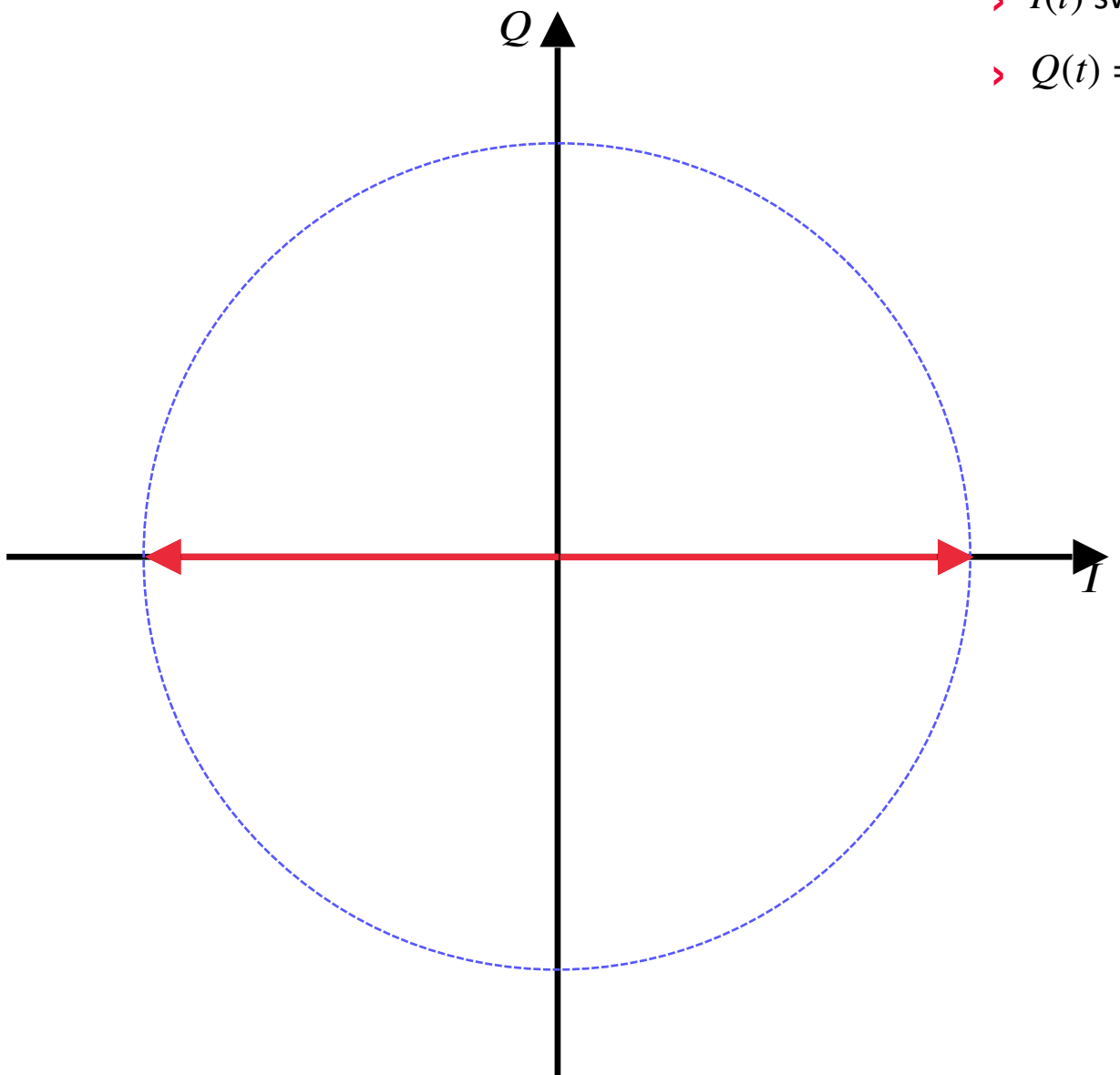




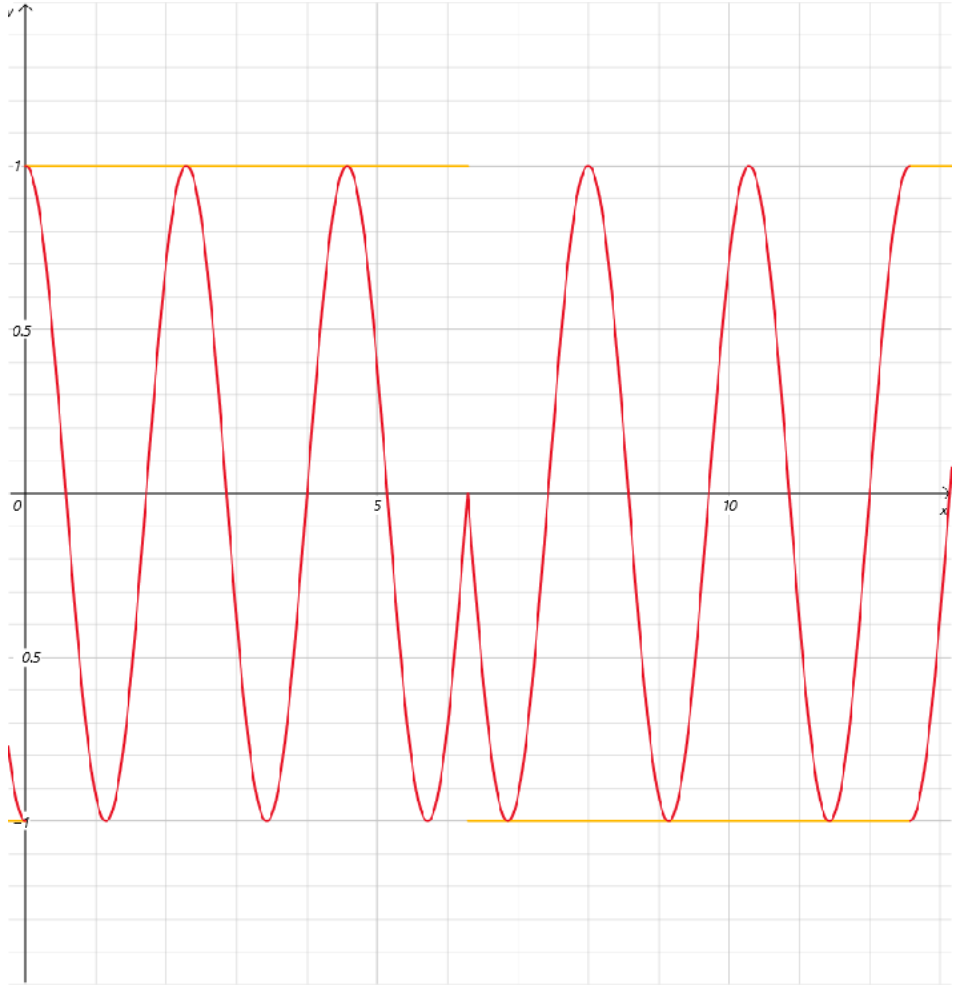
Again AM



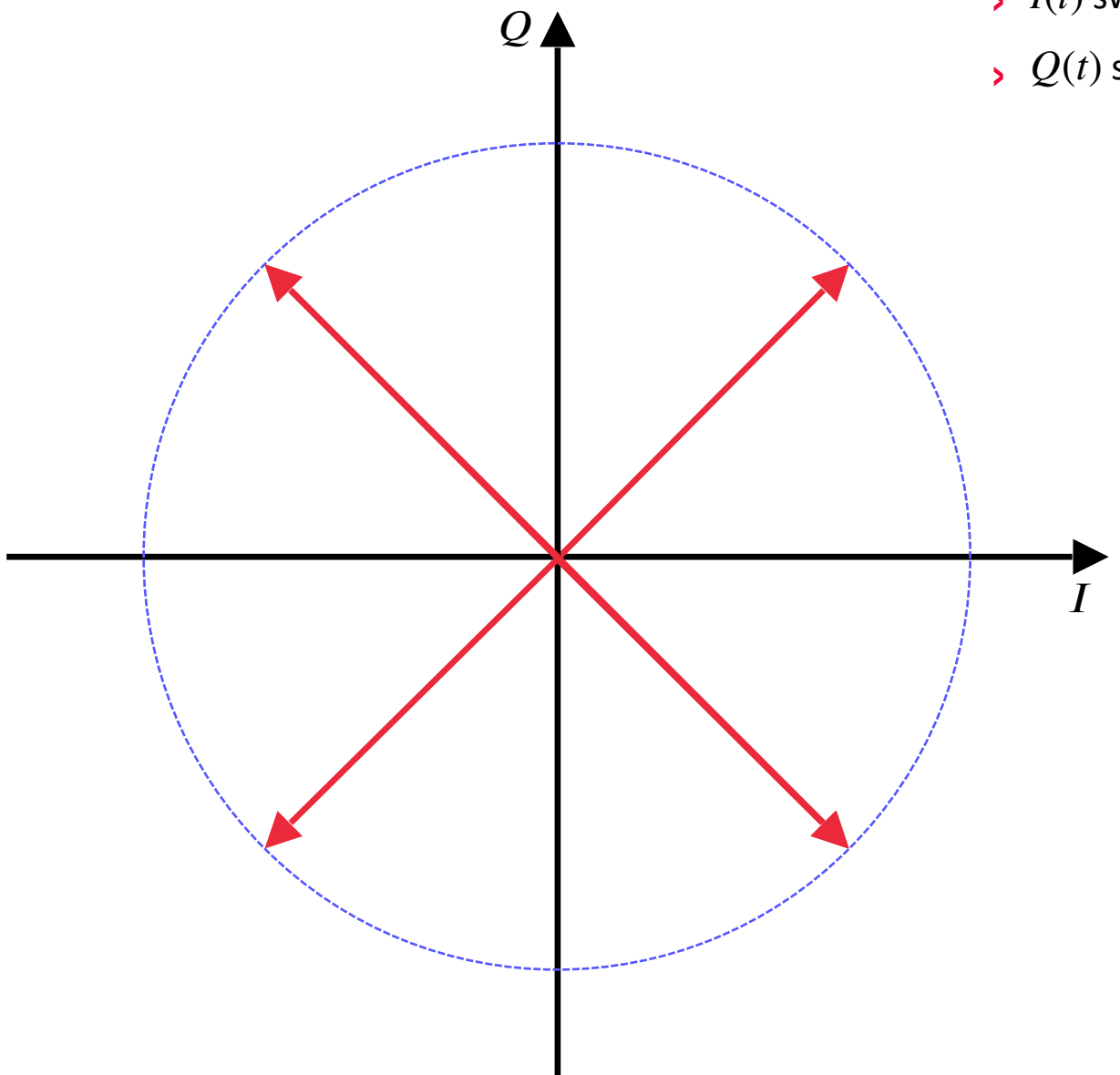
Binary Phase Shift Keying (BPSK)



- $I(t)$ switches between +1 and -1
- $Q(t) = 0$

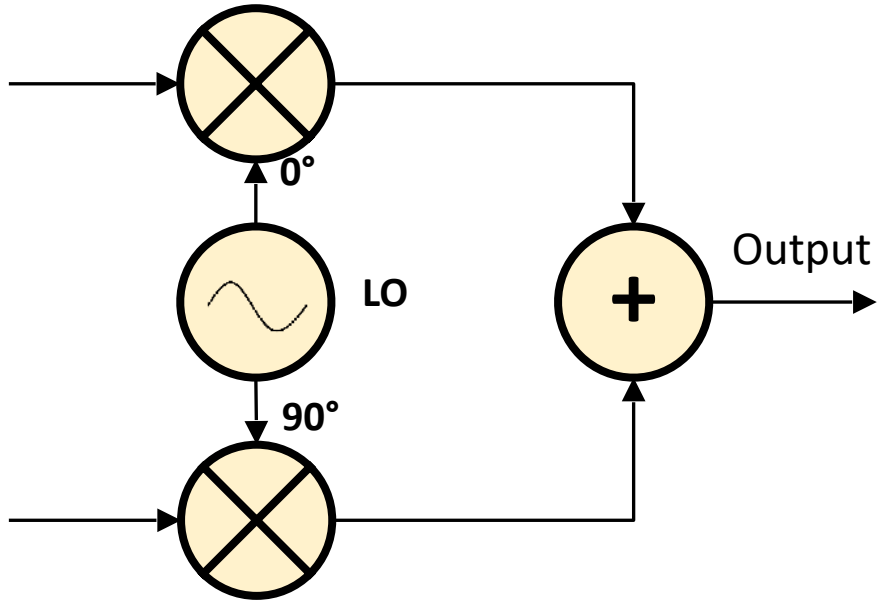


Quadrature Phase Shift Keying (QPSK)



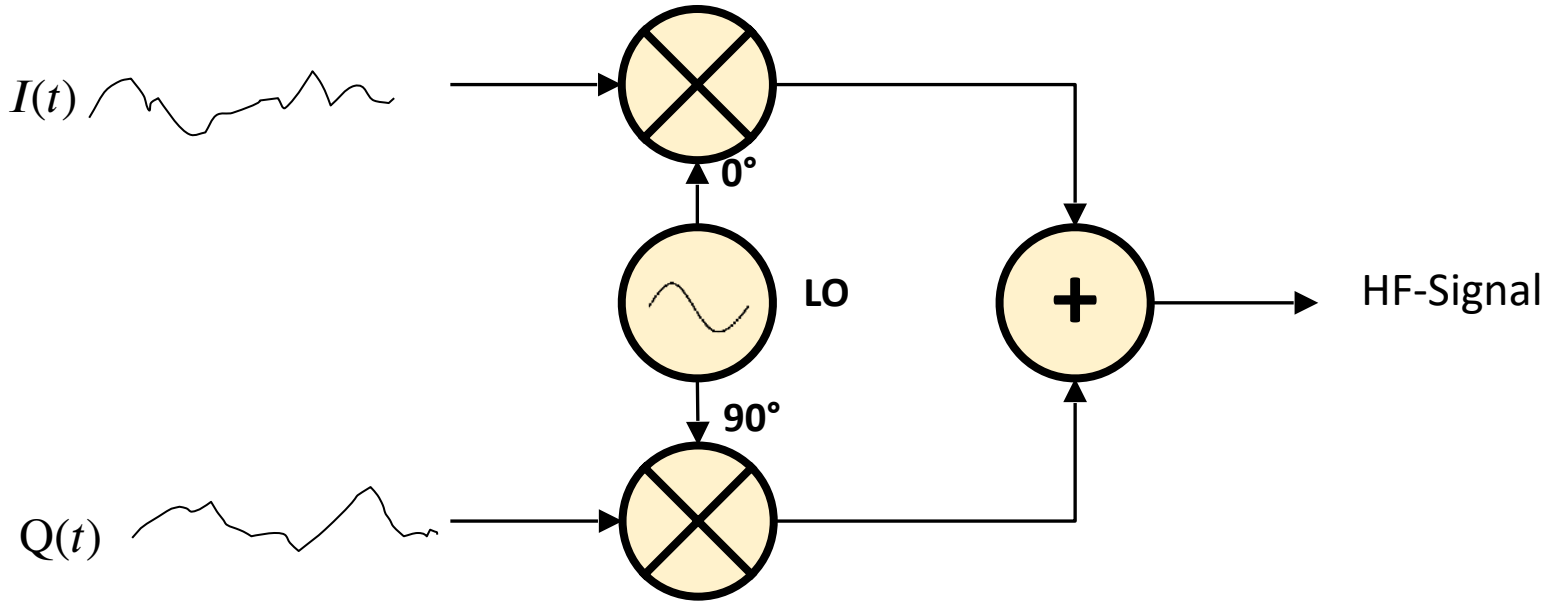
- $I(t)$ switches between +1 and -1
- $Q(t)$ switches between +1 and -1

I	Q	Output
+1	+1	45°
-1	+1	135°
-1	-1	225°
+1	-1	315°



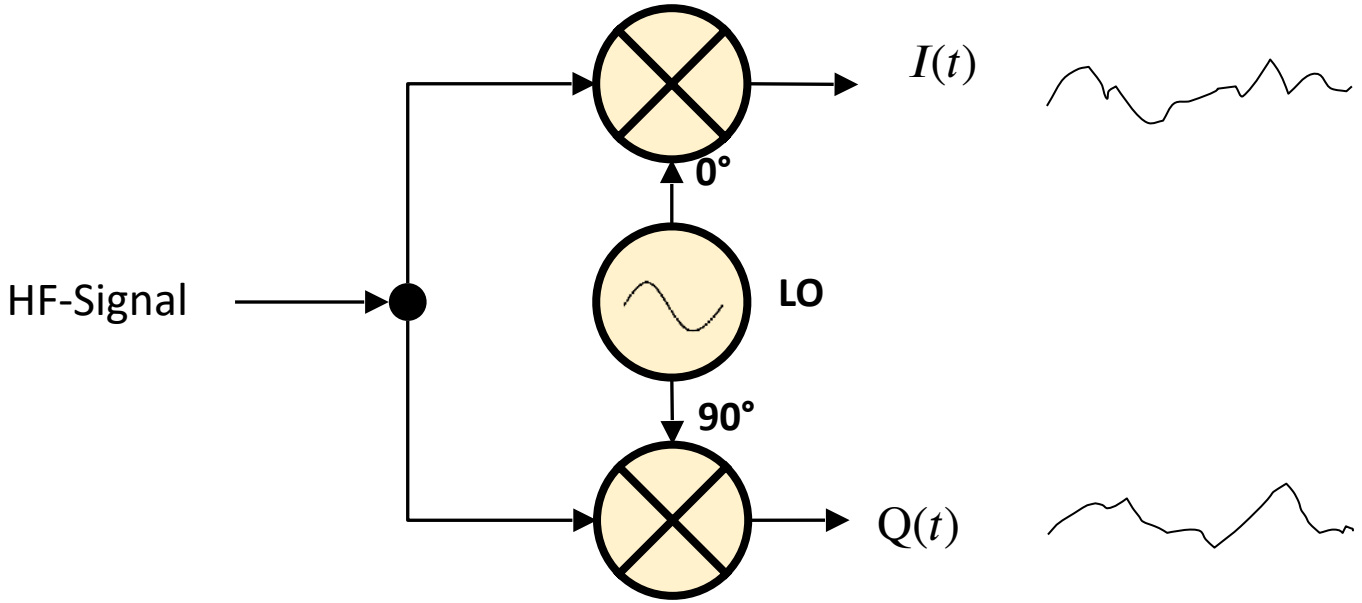
I/Q-Modulation

- With the correct I and Q signal any modulation (AM, FM, PM, SSB, QPSK…) can be generated



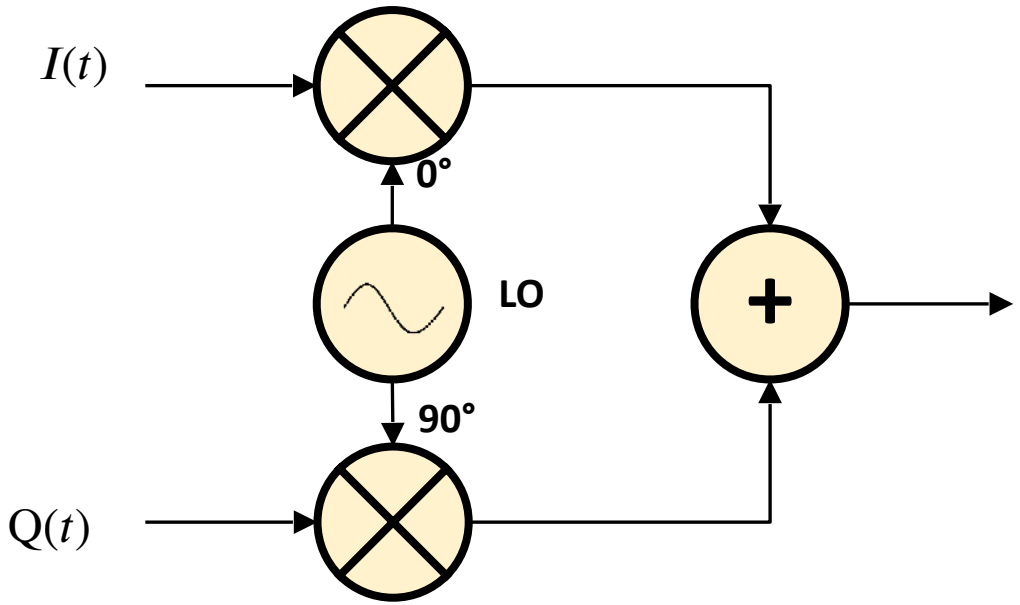
Demodulation of I/Q-Signals

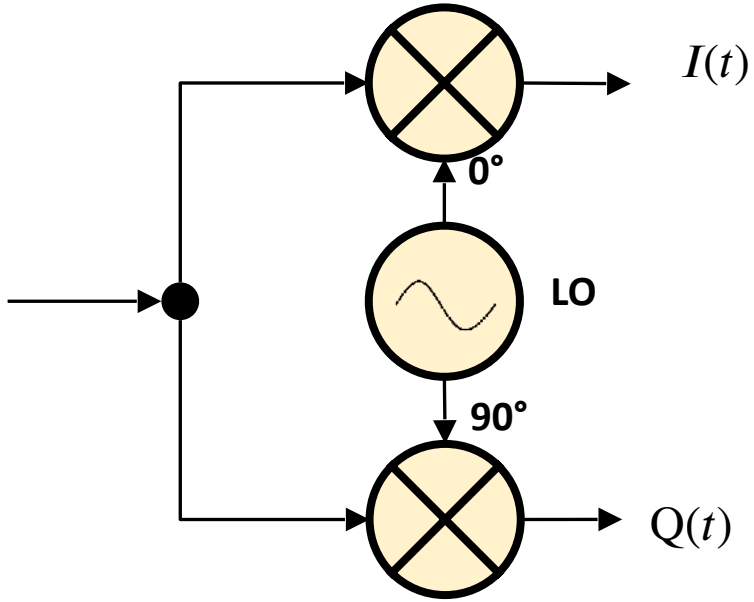
- Any kind of HF-Signal can be transformed into I/Q-Signals



Now we have the basis to build an SDR -Radio

- Basic blocks for modulation/demodulation
- I/Q Signals be now be easily analysed or generated in Software (at moderate requirements)
- We need ADC/DAC



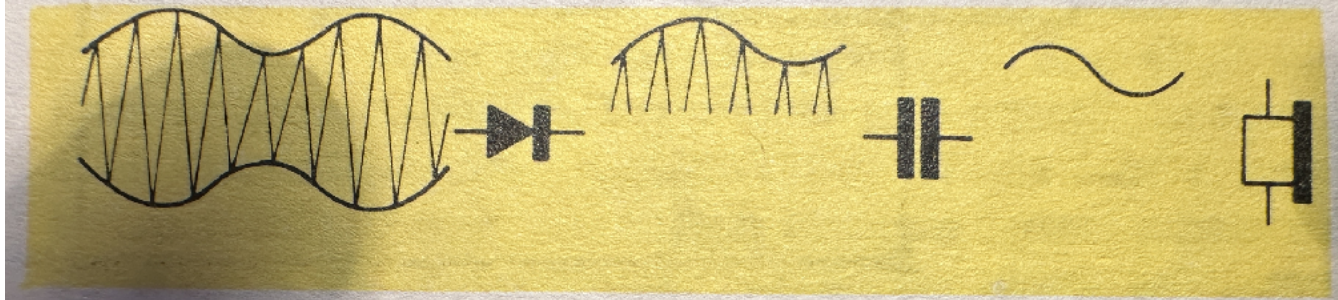


How SDR-Radio are Build?

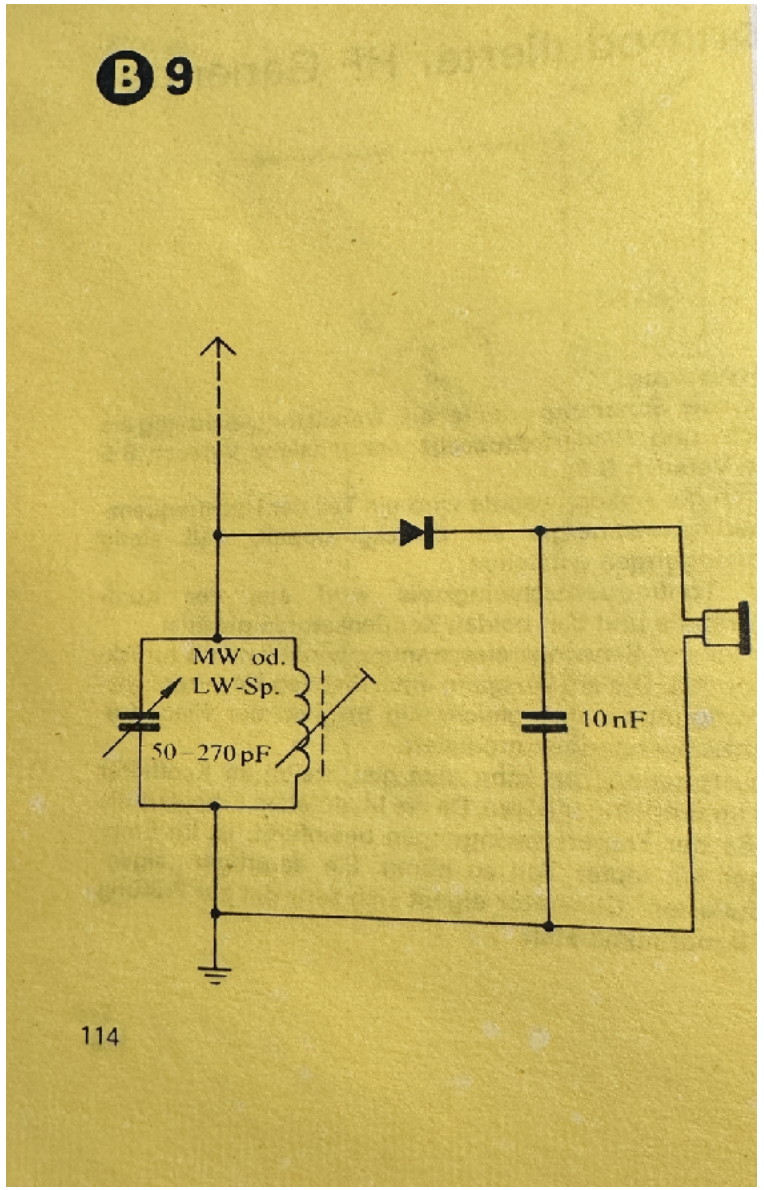
Evolution

Baukastensystem
POLYTRONIC
Elektrotechnik / Funktechnik / Elektronik
A B C

Die Membran des Kopfhörer folgt nur dem Wechsel der Spannung mit der niedrigen Frequenz, da sie für den schnellen Wechsel der Trägerfrequenz viel zu träge ist. Außerdem wird der Hochfrequenzanteil durch den Kondensator praktisch kurzgeschlossen.

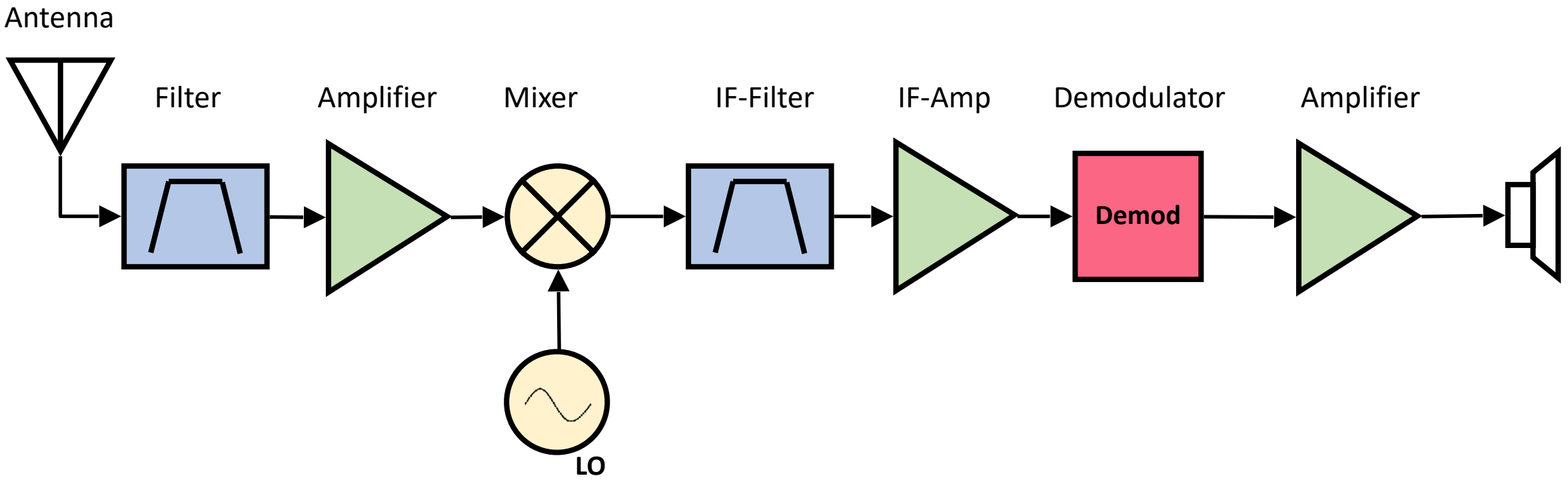


Die Gewinnung der Tonfrequenzschwingungen aus der Hochfrequenzschwingung bezeichnet man als Demodulation. Dabei spielt die Diode als Hochfrequenzgleichrichter eine große Rolle. Früher bezeichnete man diesen Gleichrichter als Detektor.



Classic Analog Technologies

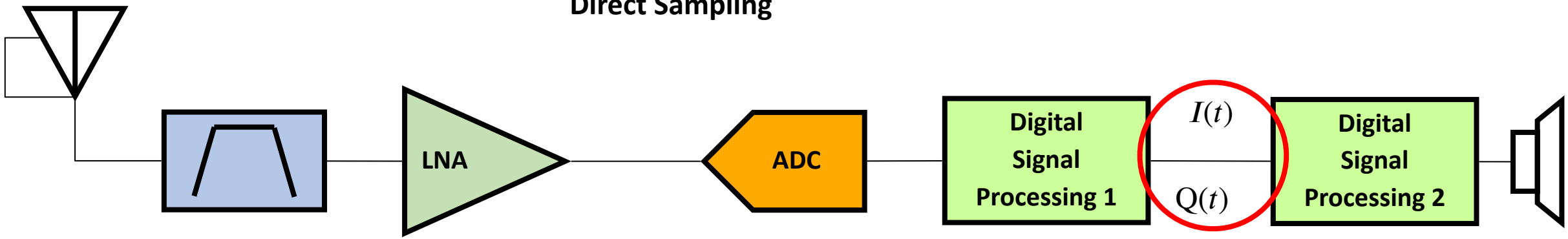
Receiver



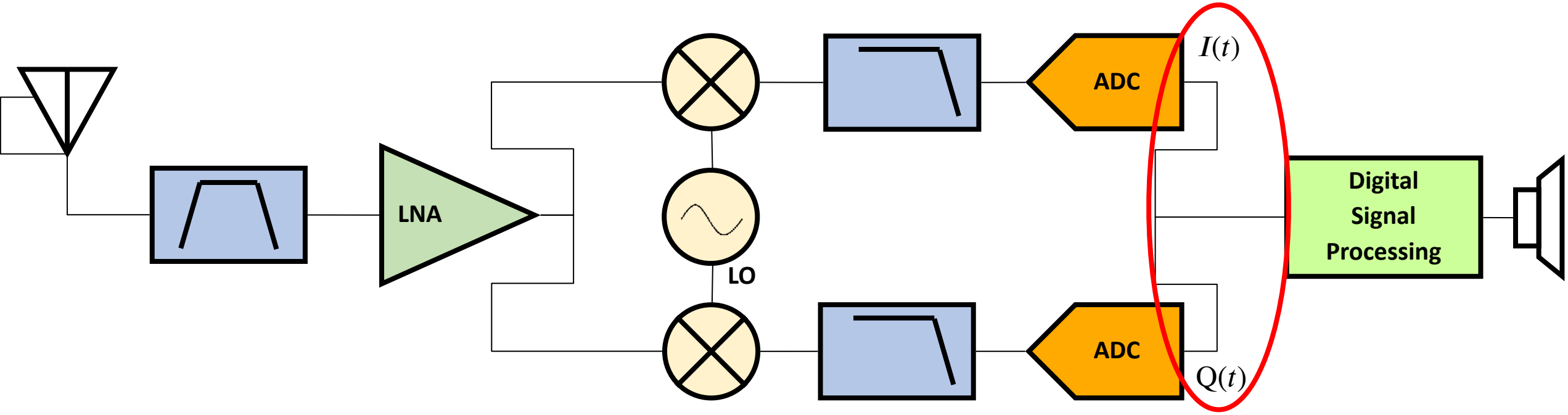
SDR

Variants

Direct Sampling



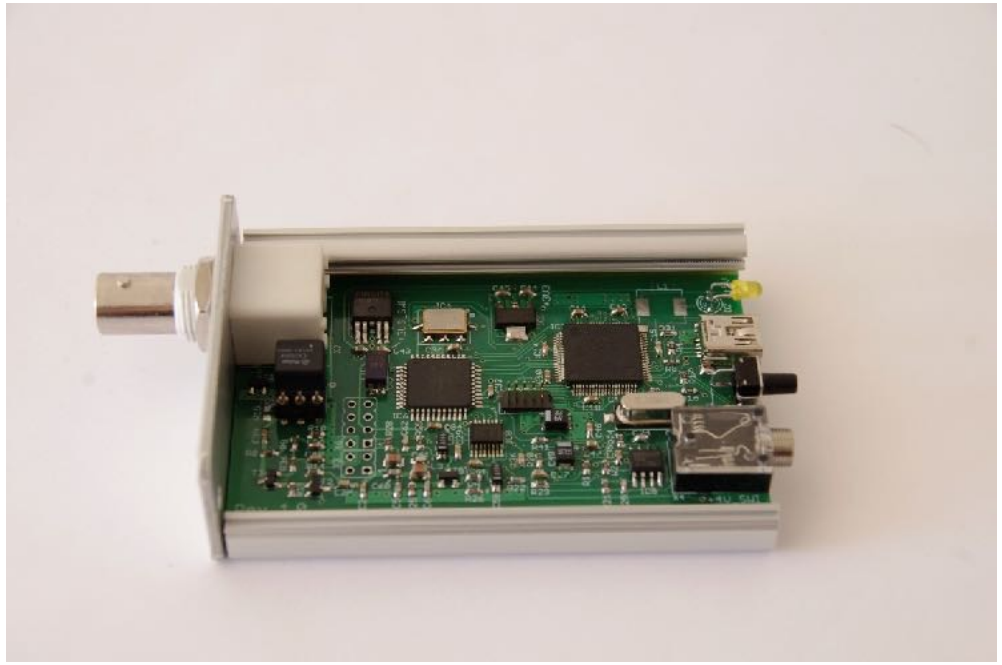
Direct Conversion



SDR Hardware

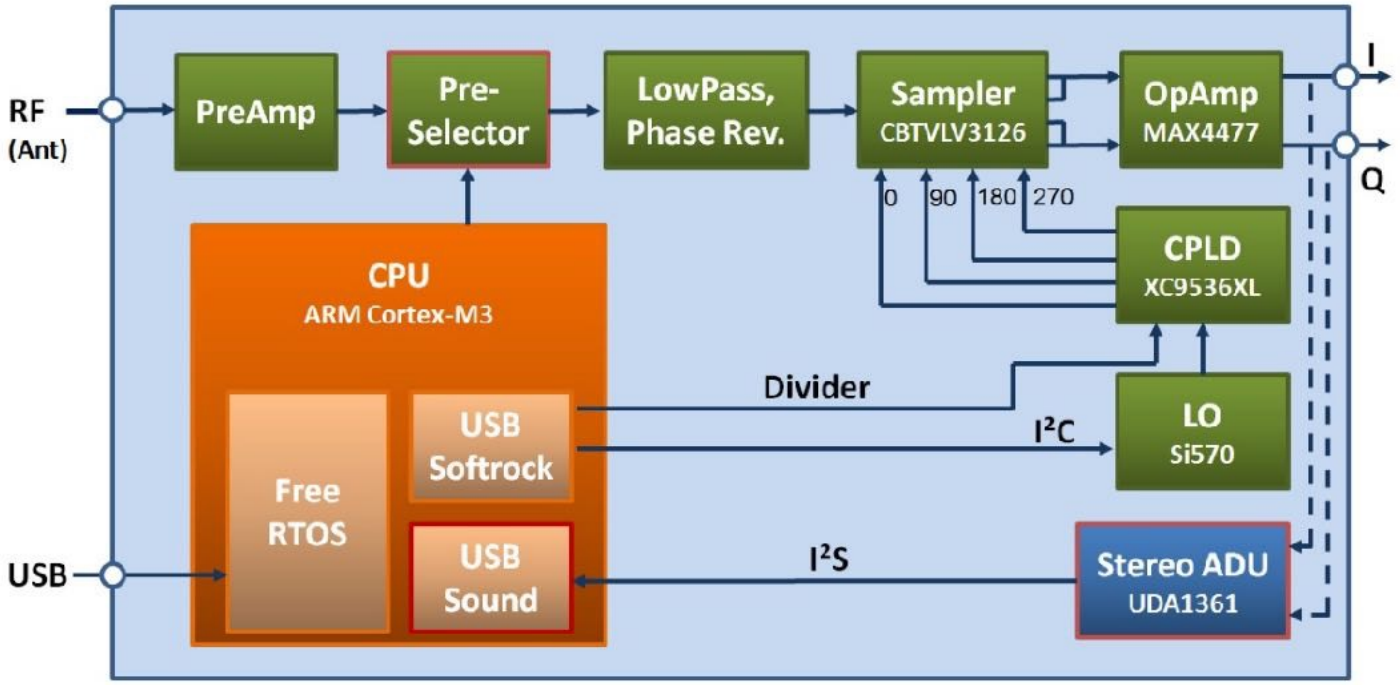
Some Examples

FiFi SDR



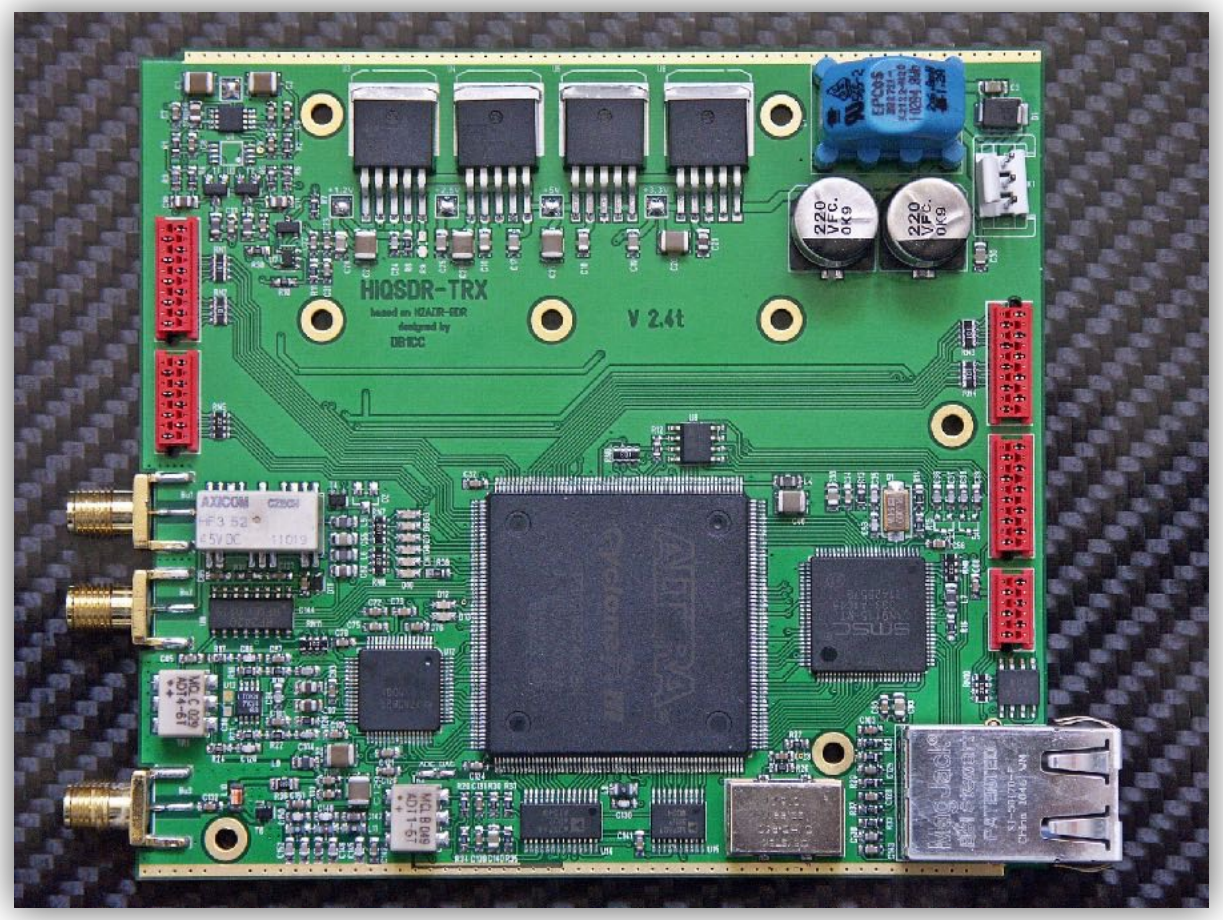
https://dh1tw.de/img/2011/04/IGP6607.jpg

- Frequency range: 0,1 .. 30 MHz
- ADC and DAC: UDA1361
- Channel bandwidth: 192 kHz
- CPLD



HiQSDR

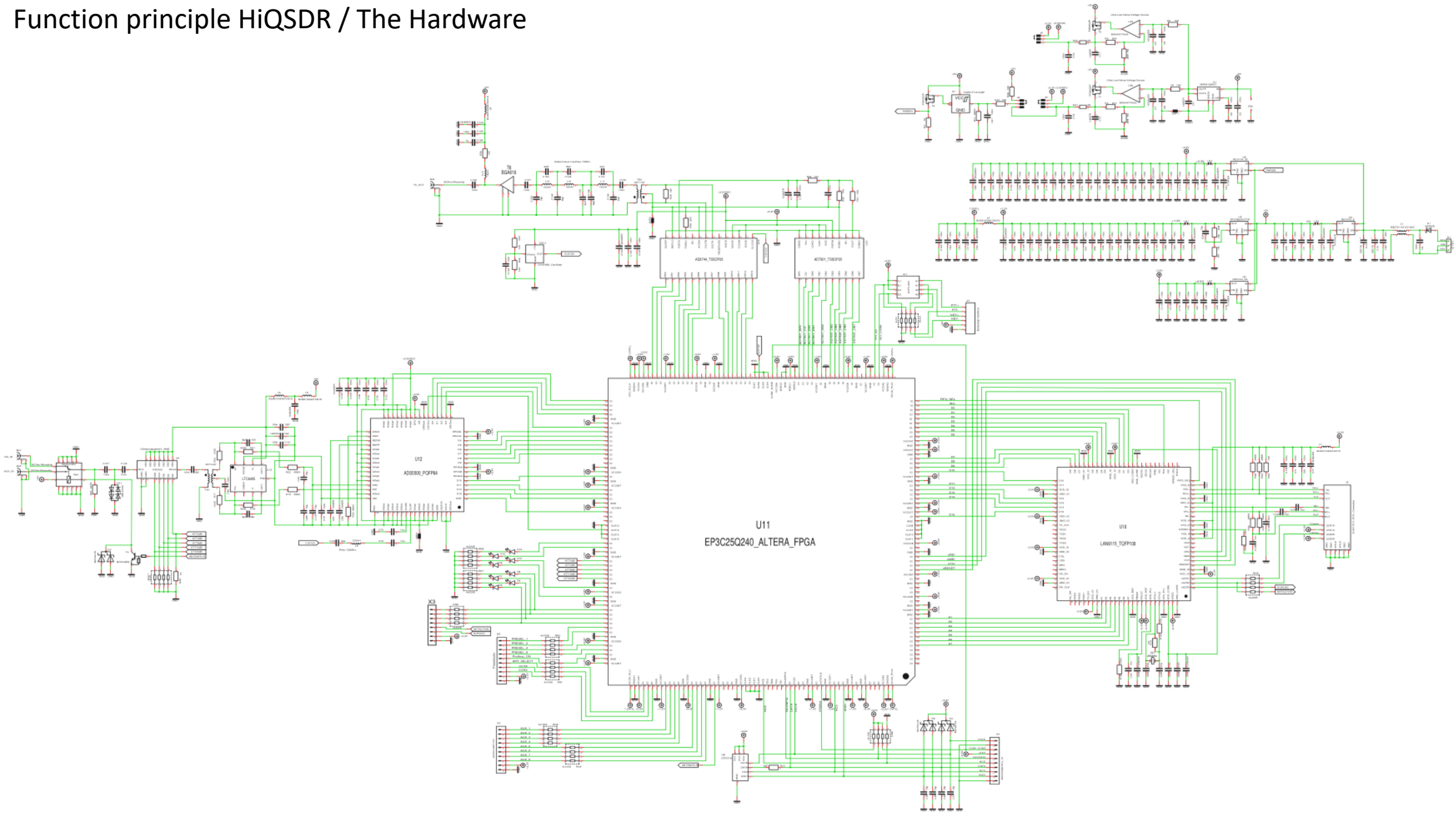
Direct Digital Conversion SDR Transceiver based on N2ADR-SDR rebuild by DB1CC



- Frequency range: 30 kHz - 62 MHz
- ADC and DAC: 14 bit
- Channel bandwidth: 48 kHz - 960 kHz
- FPGA: Altera Cyclone III
- Clock is a Crystek 122.88 MHz CVHD-950-122.880

HiQSDR

Function principle HiQSDR / The Hardware

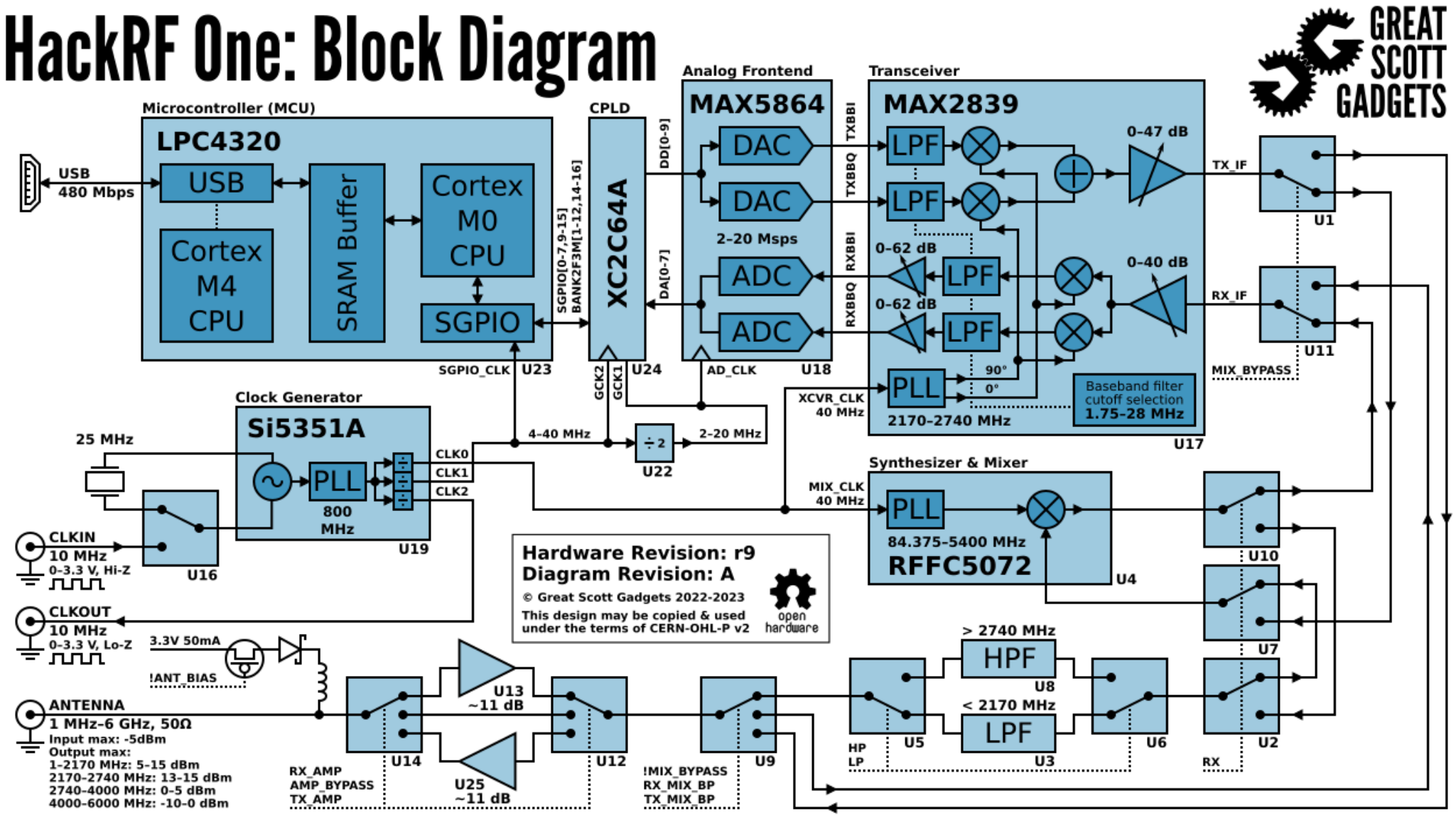


Pro/Contra Direct Conversion

- PRO:
  - Complete digital processing from antenna to demodulated signal output.
  - Full digital I/Q processing results in no amplitude / phase errors, no corrections needed
  - Full digital processing even for the RF part results in clean and exact signals and allows the maximum of possible linearity of the used ADC/DAC and their dynamic range.
  - Hi resolution of ADC/DAC allows high usable dynamic range and results in a good IP3
  - Digital processing in FPGA allows new functions / filters / functions loaded „on-the-fly“
  - Transfer of digital data via Network allows large operating range & speed, only limited on the range of the network and allows easy remote operated transceivers.
- CONTRA:
  - Higher cost due to FPGA und expensive ADC und DAC- converter

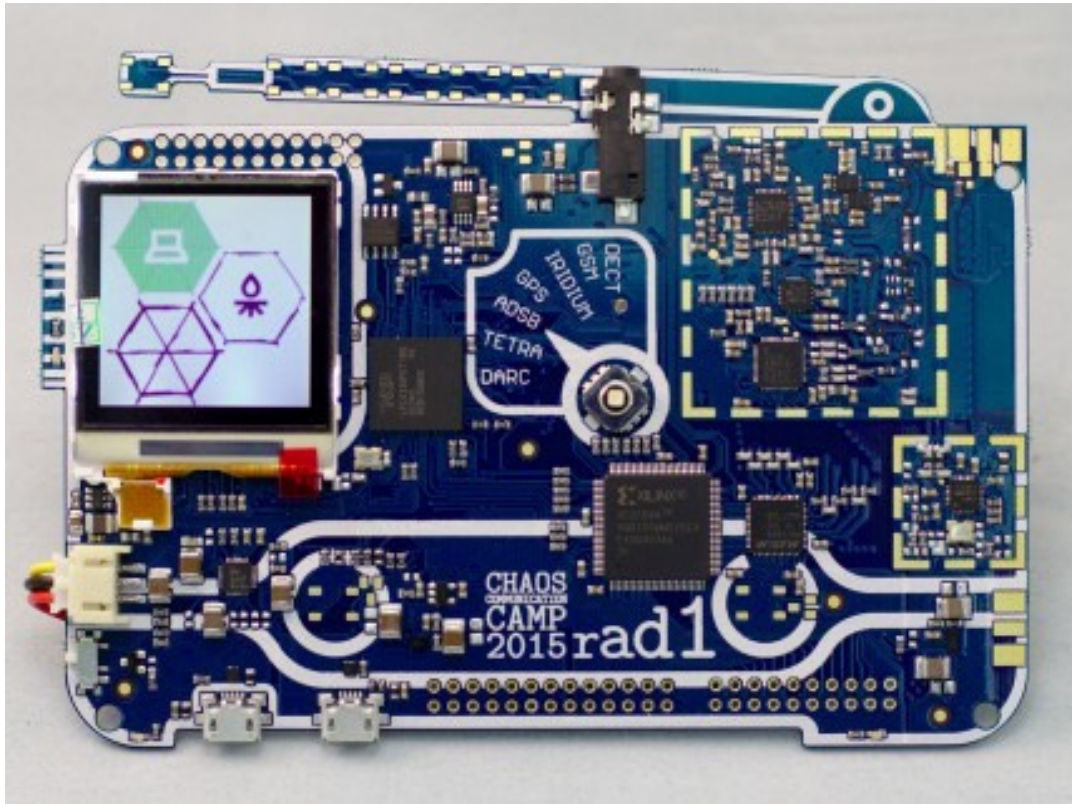
HackRF
(original)

- Frequency range: 1 MHz – 6 GHz
- ADC and DAC: 8 bits wide
- Channel bandwidth: Up to 20 MHz
- FPGA: None


HackRF One: Block Diagram
GREAT SCOTT GADGETS
Microcontroller (MCU)
LPC4320
USB
480 Mbps
USB
Cortex M4 CPU
SRAM Buffer
Cortex M0 CPU
SGPIO
SGPIO_CLK
U23
CPLD
XC2C64A
SGPIO[0-7,9-15]
BANK2F3M[1-12,14-16]
DD[0-9]
DA[0-7]
GCK2
GCK1
U24
Analog Frontend
MAX5864
DAC
DAC
2-20 Msps
ADC
ADC
AD_CLK
U18
TXBBI
TXBBQ
RXBBI
RXBBQ
Transceiver
MAX2839
LPF
LPF
LPF
LPF
0-47 dB
0-62 dB
0-62 dB
0-40 dB
PLL
90°
0°
XCVR_CLK
40 MHz
2170–2740 MHz
Baseband filter cutoff selection
1.75–28 MHz
U17
TX_IF
RX_IF
MIX_BYPASS
U1
U11
Clock Generator
Si5351A
25 MHz
PLL
800 MHz
CLK0
CLK1
CLK2
U19
4-40 MHz
÷2
2-20 MHz
U22
Synthesizer & Mixer
MIX_CLK
40 MHz
PLL
84.375-5400 MHz
RFFC5072
U4
U10
U7
U2
CLKIN
10 MHz
0-3.3 V, Hi-Z
U16
CLKOUT
10 MHz
0-3.3 V, Lo-Z
Hardware Revision: r9
Diagram Revision: A
© Great Scott Gadgets 2022-2023
This design may be copied & used under the terms of CERN-OHL-P v2
open hardware
3.3V 50mA
!ANT_BIAS
ANTENNA
1 MHz–6 GHz, 50Ω
Input max: -5dBm
Output max:
1–2170 MHz: 5–15 dBm
2170–2740 MHz: 13–15 dBm
2740–4000 MHz: 0–5 dBm
4000–6000 MHz: -10–0 dBm
U13
~11 dB
U25
~11 dB
U14
U12
RX_AMP
AMP_BYPASS
TX_AMP
U9
!MIX_BYPASS
RX_MIX_BP
TX_MIX_BP
> 2740 MHz
HPF
U8
< 2170 MHz
LPF
U3
HP
LP
U5
U6
RX

rad10 badge

HackeRF clone



- Frequency range: 50 MHz – 4 GHz
- ADC and DAC: 8 bits wide
- Channel bandwidth: Up to 20 MHz
- CoolRunner-II CPLD

Pluto

- Frequency range: 325 MHz – 3.8 GHz
  - (based on Analog Devices AD9363)
- ADC and DAC: 12 bits wide
- Channel bandwidth: Up to 20 MHz
- FPGA: Xilinx Zynq Z-7010

Gnuradio

What is GNURadio

- Framework
  - Currently C++ and Python supported
- Development tools
  - Tools to packages and interface your code with others
- Scheduler
  - Use all the cores
- Interface and abstraction of hardware
  - Both radio and some acceleration
    - Pluto-SDR
    - HackRF
    - ….
- Library of DSP-functionality
- Graphical interface
  - GNURadio Companion

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

Housekeeping

Symbols

- I is the In-Phase signal component
- Q is the Quadrature signal component
- i is the mathematical constant i such as $i^2 = 1$
- A is used for peak amplitude, the envelope, of the signal
- ϕ Phase shift
- e is the base for the natural logarithm $\approx$ 2.71828
- $\otimes$ is the operation mixing frequencies
- • is the operation multiplication
- $\mathbb{R}$ denotes real-numbers.
- $\mathbb{C}$ denotes complex numbers.
- DC for "direct current", the frequency zero.

Links

- https://www.cgran.org/
- https://wiki.gnuradio.org/index.php?title=Tutorials
- https://wirelesspi.com/i-q-signals-101-neither-complex-nor-complicated/
- https://medium.com/@nabanita.sarkar/simulating-amplitude-modulation-using-python-6ed03eb4e712
- https://github.com/MuSAELab/amplitude-modulation-analysis-module
- https://www.pe0sat.vgnet.nl/sdr/iq-data-explained/
- Quadraturamplitudenmodulation – Wikipedia
- Software-Defined Radios (arachnoid.com)
- SDR: Quadraturdemodulation vs. direkte HF-Abtastung (narkive.de)
- CRE087 Software Defined Radio | CRE: Technik, Kultur, Gesellschaft
- SDR: Wie werden I und Q aus dem eingehenden Signal bei der Quadraturabtastung auf der Empfängerseite bestimmt? (isolution.pro)
- https://youtu.be/h_7d-m1ehoY?list=PLhtYYpsE3LzVUVSNHUVhfcoaI3mLEKSIx
- https://youtu.be/5GGD99Qi1PA?list=PLhtYYpsE3LzVUVSNHUVhfcoaI3mLEKSIx
- https://f1atb.fr/index.php/2021/03/18/nbfm-transceiver-with-pluto-sdr-and-gnu-radio/

Many thanks