Erfahrungen mit SDR und SDR-Programmierung

Alfred Wullschleger, HB9EPU

2. Februar 2011

USKA Sektion Winterthur

Inhalt

- Teil 1: Einleitung
  - SDR-Strukturübersicht
    - Direktkonverter
- Teil 2: DSP-Programm & Demonstration
  - Eigene Versuche: Empfängersoftware
    - Demonstration des DSP-Programms
    - Aktuelle Weiterentwicklungen
    - Diskussion
- Teil 3 aus Zeitgründen verschoben auf später...

Teil 1:
Einleitung

SDR-Strukturübersicht

Konzepte

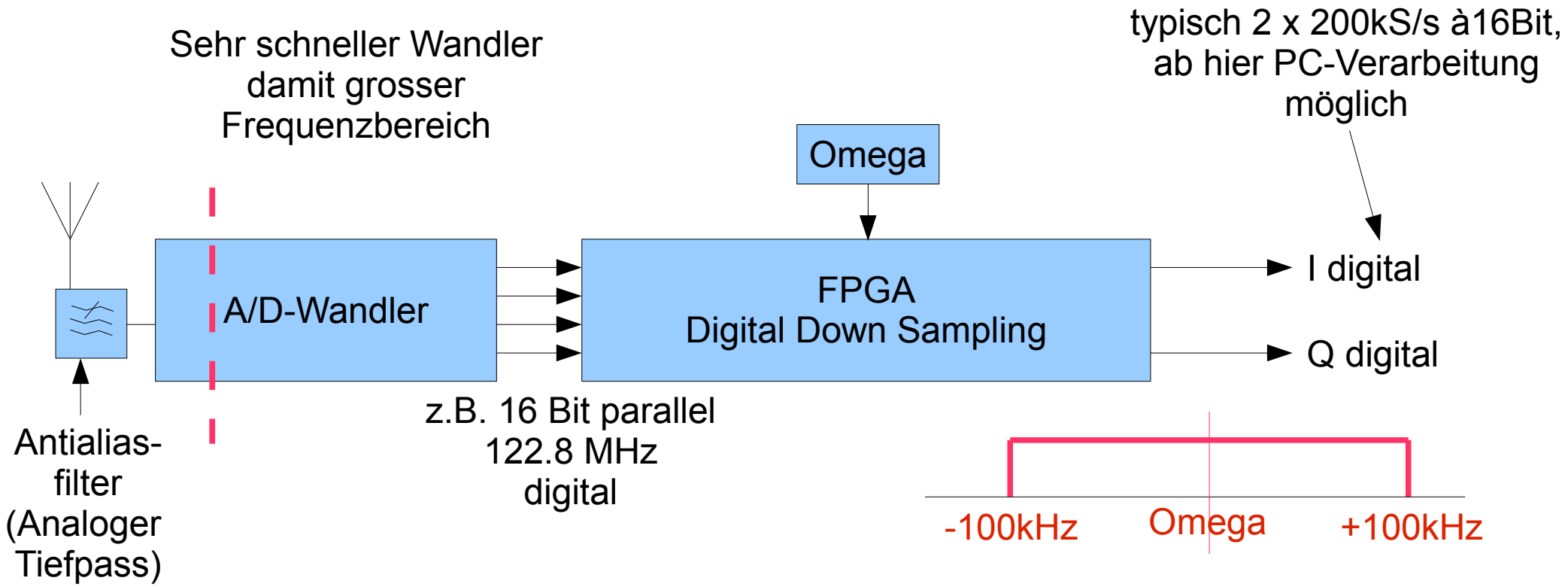
SDR-Strukturübersicht

- Grundsatz
  - Erzeugung eines digitalen Datenstroms durch Signalabtastung und A/D-Wandlung
- Zwei Empfangskonzepte
  - Schalt-Mischer (z.B. Flex-Radio, FA-SDR)
    - siehe Vortrag vom 1.9.2010 von HB9BGG und HB9MTN
  - Direktkonverter (z.B. Perseus, QuickSilver, HPSDR)
  - Ziel: Generierung von digitalen I- und Q-Signalen
    - Multiplikation mit cos(Omega*t) für I und sin(Omega*t) für Q
  - transformiert Frequenzband um Omega nach 0

Warum I- und Q-Signale?

- Zusammenhang zw. Zeit und Frequenz wird durch komplexe Zahlen beschrieben
  - 2 reelle Zahlen: Realteil und Imaginärteil
  - es braucht positive und negative Frequenzen
- Alle Modulationsarten können mit I- und Q-Signalen vollständig beschrieben werden
  - einfache Operationen für Selektion USB, LSB, CW oder AM durch entsprechende Filter
    - z.B. LSB durch Filtersetzung bei negativen Frequenzen
    - z.B. USB durch Filtersetzung bei positiven Frequenzen

Struktur Direktkonverter



A/D-Vorgang

- Gesamtsignal von 0 bis dSR/2 wird direkt gewandelt
  - sehr hohe Samplingrate dSR nötig
    - für z.B. 55 MHz Bandbreite: dSR > 110 MHz
  - elektrisch saubere Trennung Analogteil (Signale bis S0!) vom Digitalteil (mit Frequenzen jenseits von UKW!) sehr kritisch
  - Empfindlichkeit: hohe Auflösung (z.B. 16Bit) nötig
  - Antialiasfilter zur Elimination von Signalen mit f>=dSR/2
    - gemäss Abtasttheorem

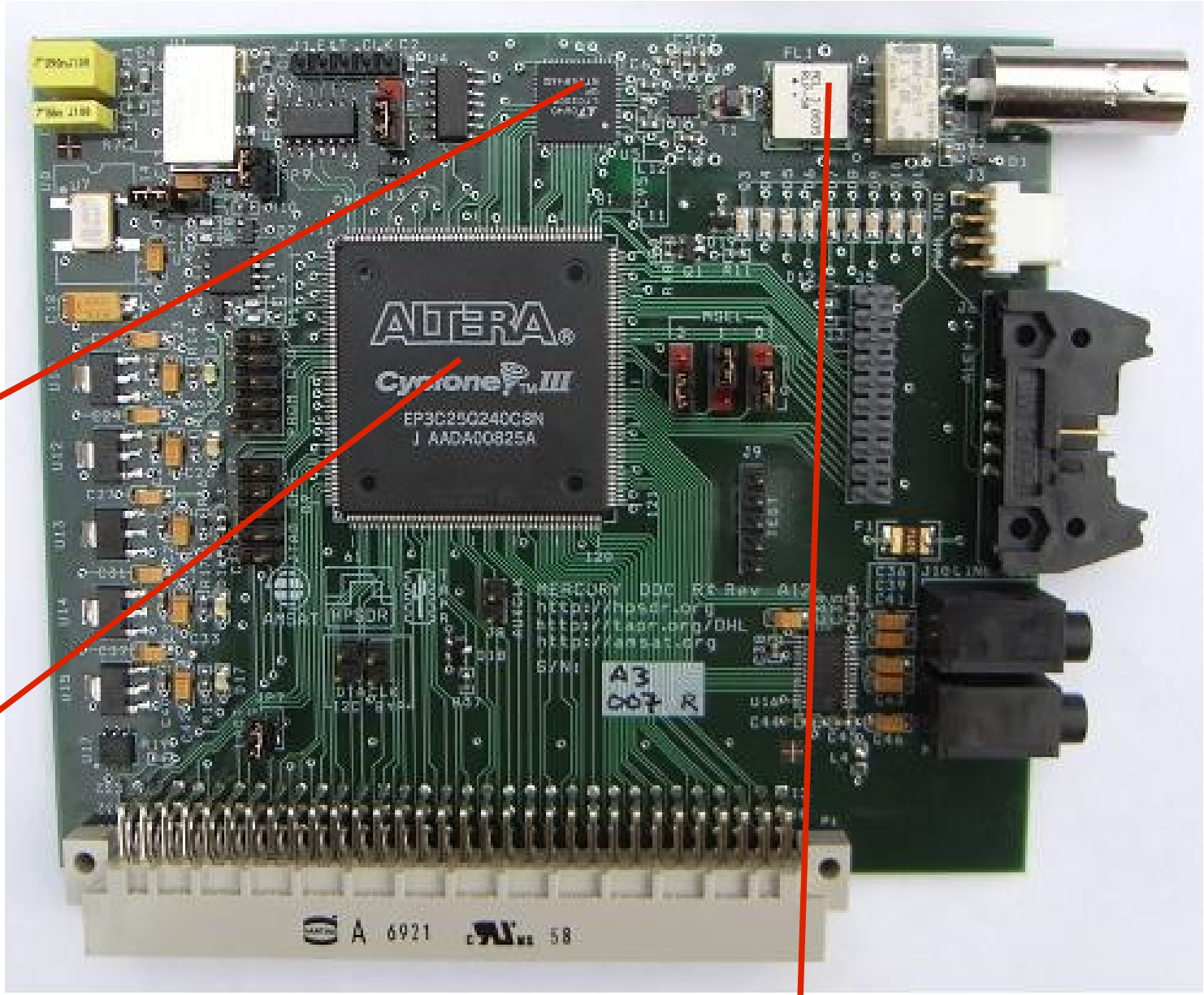
Digitale Verarbeitung

- nach dem A/D-Wandler ist alles digital
  - muss immer noch sehr schnell sein: FPGA
  - numerische Erzeugung von Sinus und Cosinus
    - keine Asymmetrie
    - Präzision nur durch Bitauflösung beschränkt
  - digitales Downsampling mit sog. CIC-Filtern
    - mehrstufiges Filtern und Elimination von Samples
    - am Ende ca. 200 kS/s, 2x16 Bit I,Q

Beispiel eines Direktkonverters

- "Mercury"-Platine aus dem HPSDR-Projekt
- A/D-Wandler LTC 2208 von Linear Technology
  - 122.88 MHz Abtastrate, 16 Bit Auflösung
- FPGA EP3C25Q240 von Altera
  - Quad Flat Pack 240 Pins
  - grösstes Altera-FPGA ohne Ball Grid Array
    - damit auch Selbstbau-Freaks unter den Amateuren das FPGA selbst einlöten können ;-)

Mercury-platine

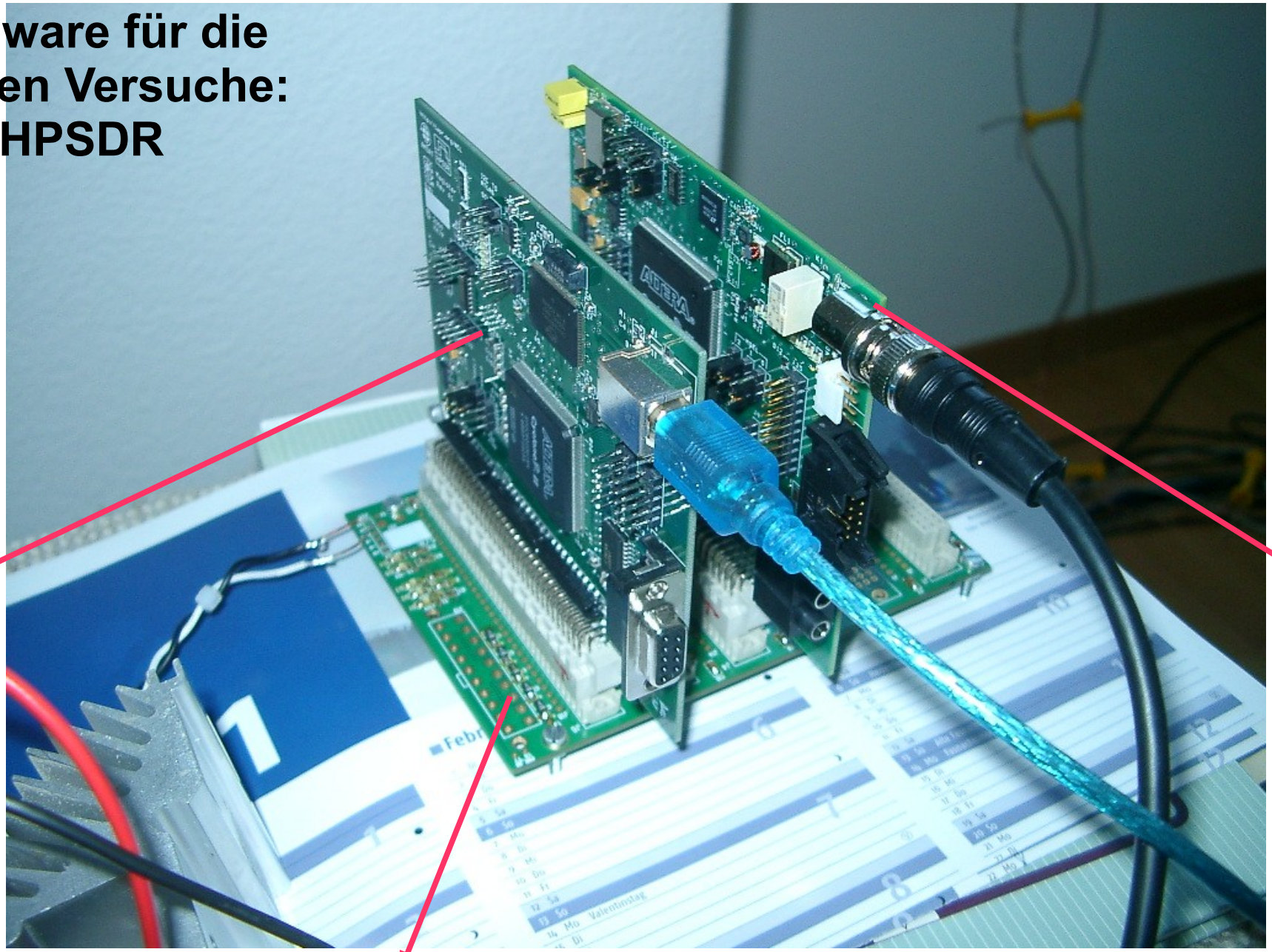


Teil 2:
DSP-Programm
und Demonstration

Eigene Versuche in DSP-Programmierung

Selbermachen ist lehrreich und sehr spannend...

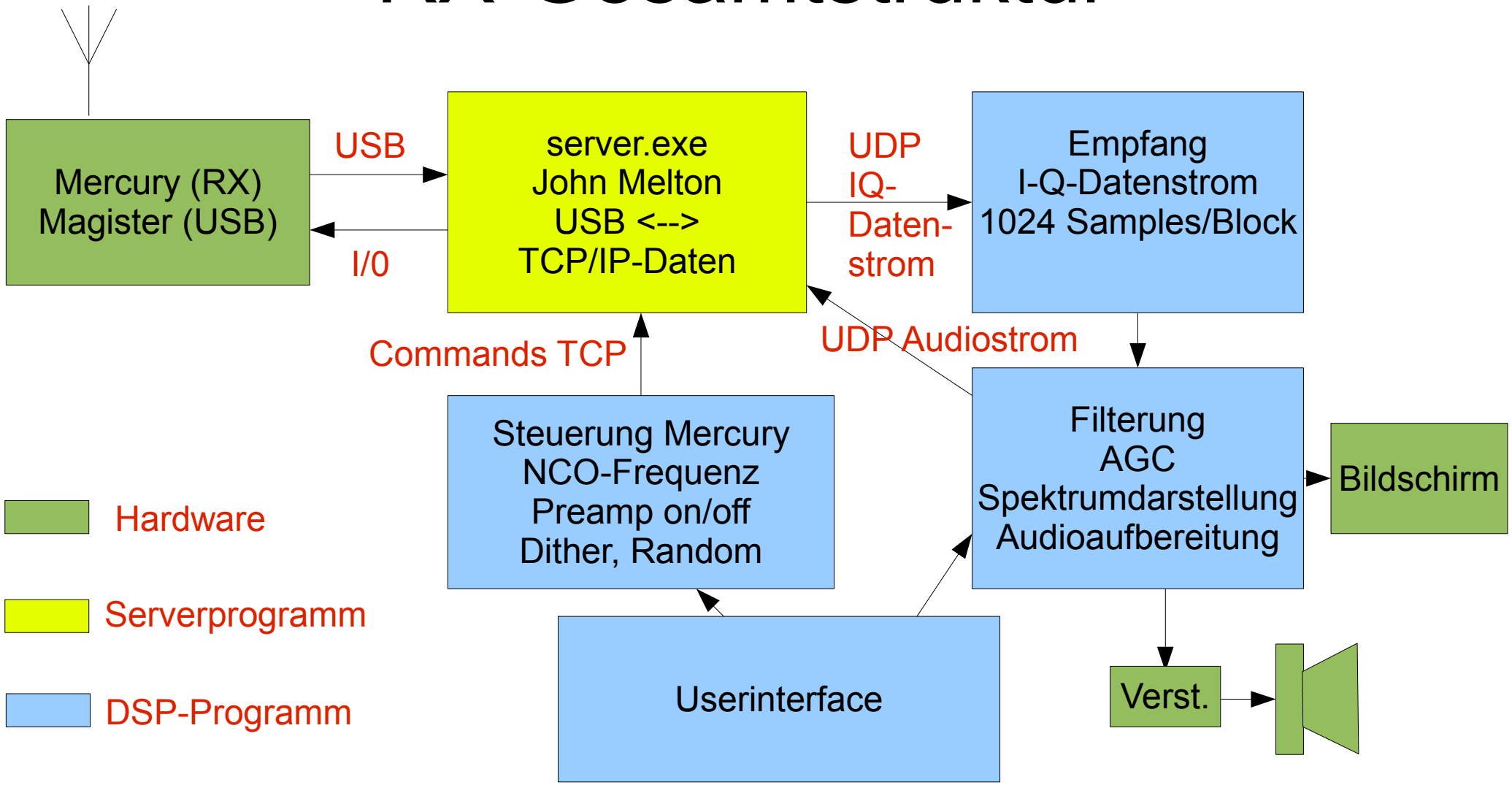
Hardware für die eigenen Versuche: HPSDR



Eigene Versuche: Material

- Hardware:
  - HPSDR-Mercury+Magister+Atlas
- Software:
  - Windows XP
  - Programm server.exe von John Melton N6LYT (G0ORX)
    - Initialisierung Mercury+Magister
    - USB zu TCP/IP und umgekehrt
    - portiert von David McQuate WA8YWQ von Linux nach Windows
    - da als Source verfügbar, sehr leicht anpassbar
  - eigenes DSP-Programm

RX-Gesamtstruktur

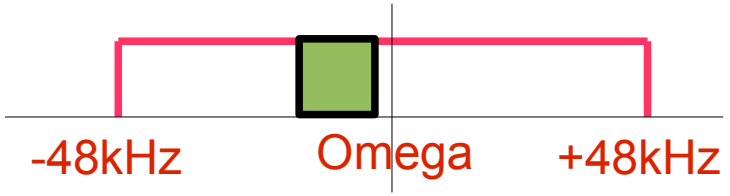


Aufbau DSP-Programm

- benutzt die TCP-Schnittstelle des server.exe
- Programmiersprachen:
  - C für Echtzeitteile
  - Smalltalk für das Userinterface
- benutzt das Bibliotheksprogramm fftw3 für die Diskrete Fouriertransformation
  - fftw3 heisst "Fastest Fourier Transform in the West"
  - fast überall im Einsatz (PowerSDR, Perseus...)

Aufgaben des DSP-Programms

- Spektrumdarstellung
  - mittels digitaler Fouriertransformation und Betragsbildung
- Demodulation
  - mittels digitaler Filterung

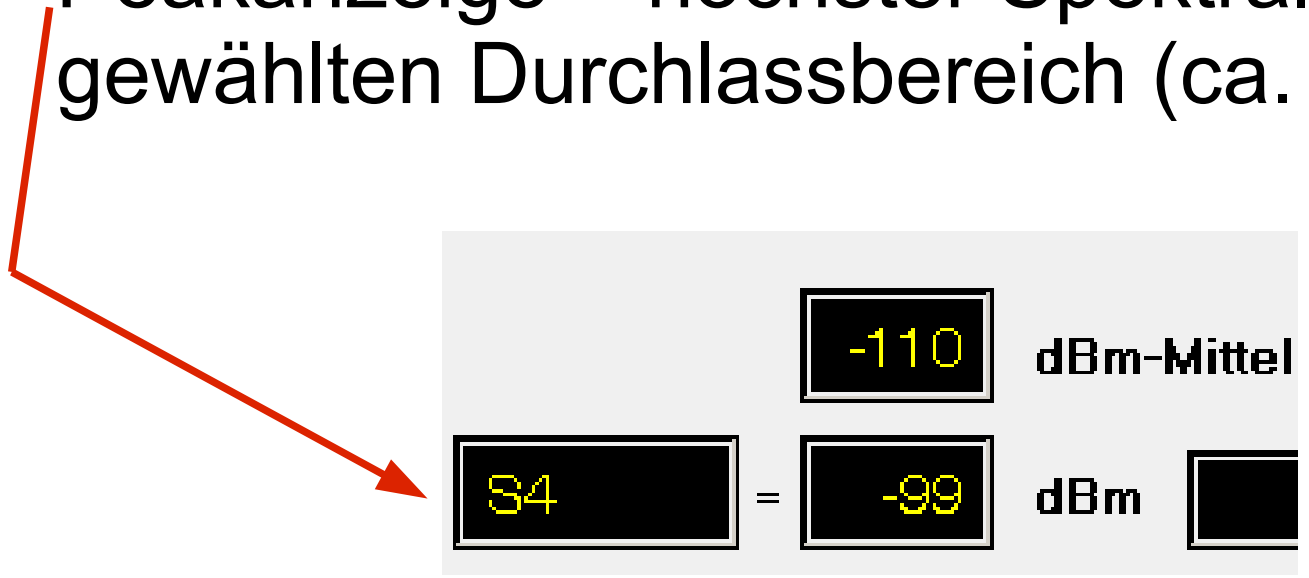


- S-Meter-Anzeige
  - von -127dBm bis 0dBm präzise, weil direkt gemessen
- AGC und Notchfilter

Demonstration DSP-Programm

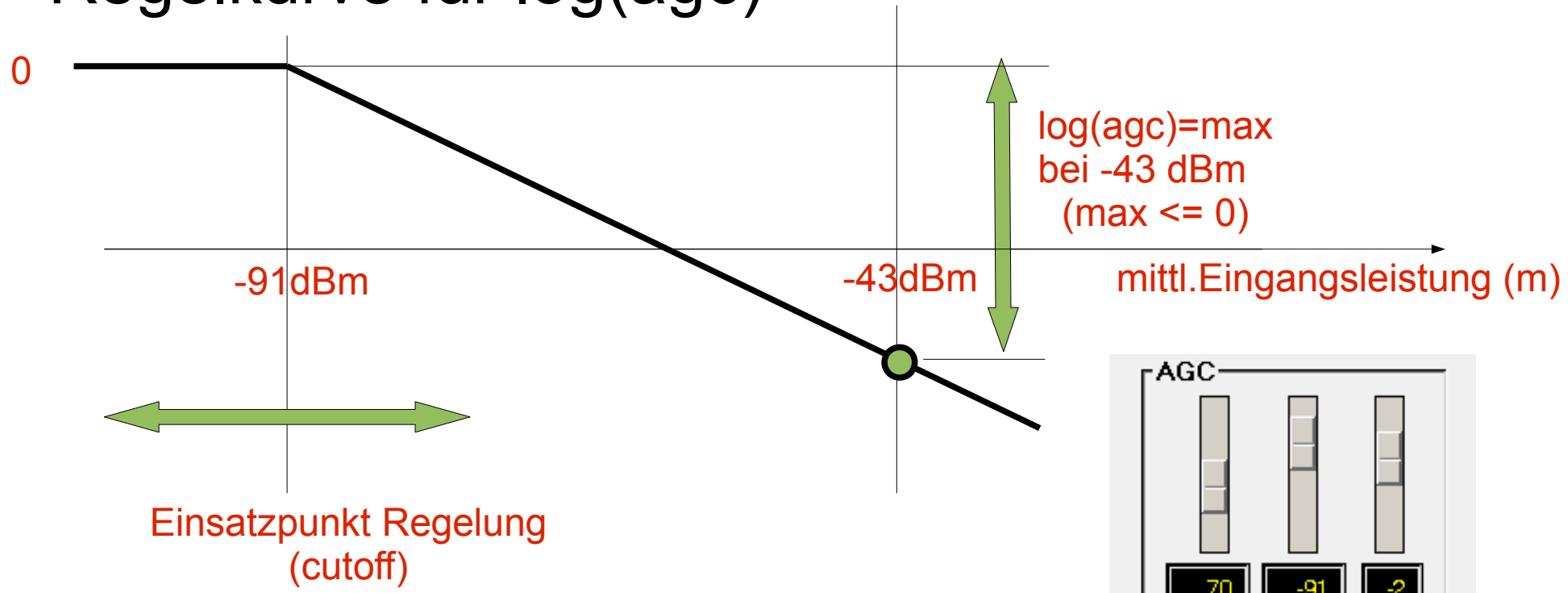
S-Meter

- Mittelwert = gemittelter Spektralbetrag über den gewählten Durchlassbereich
- Peakanzeige = höchster Spektralbetrag im gewählten Durchlassbereich (ca. 1 Mal pro s)



AGC (1)

- Regelkurve für log(agc)



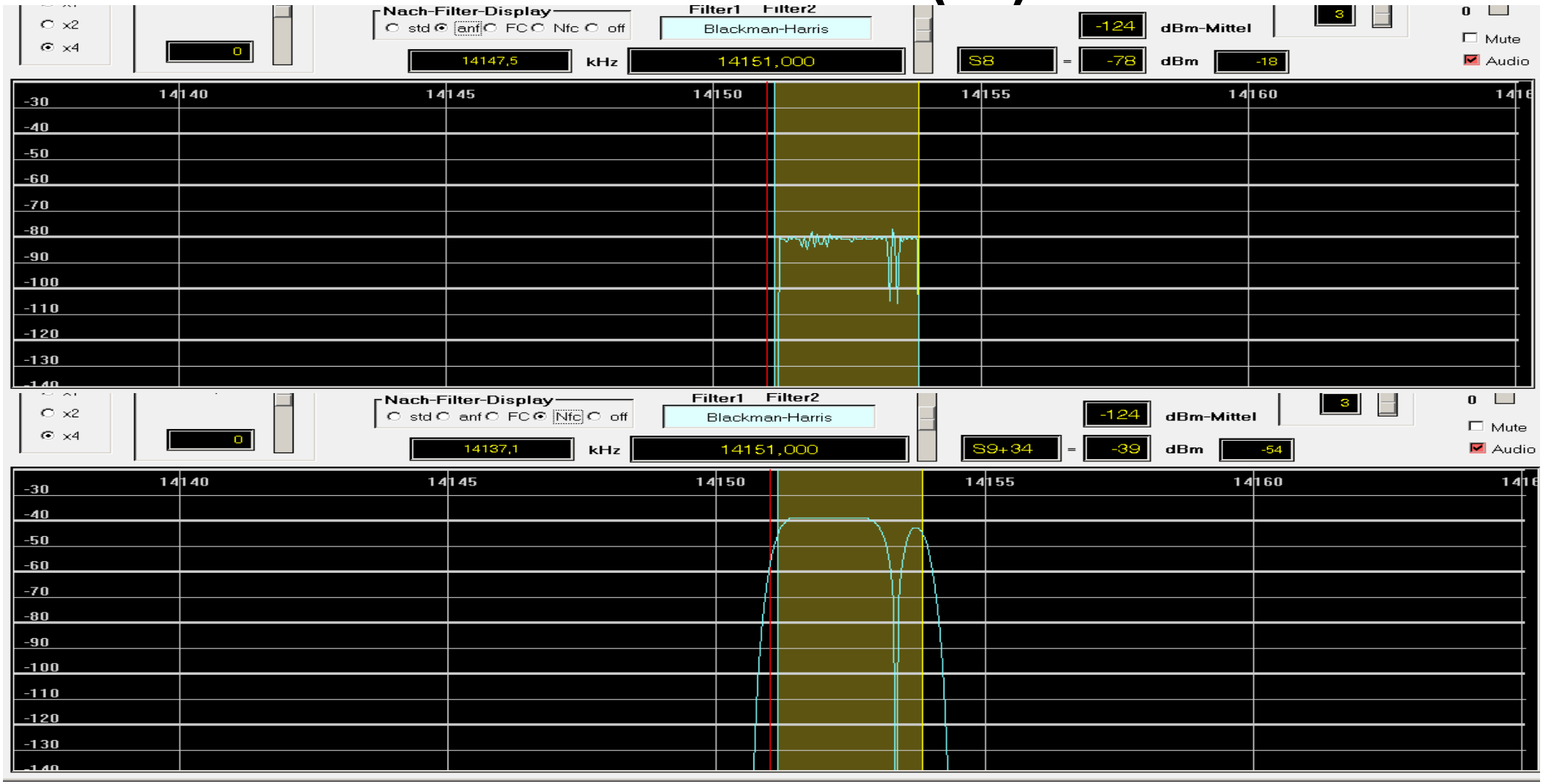
Mittelungszeit (count)

log(agc) = max * (1 – (m+43)/(cutoff+43)) f.m>cutoff
= 0 f.m <=cutoff

AGC (2)

- zwei gleitende Mittel (g.M.) über die Samples im Durchlassbereich werden gerechnet
  - Maximal-dBmWert = g.M. grösster Spektralwert
    - bewirkt Störunterdrückung bei mehr als -13 dBm
  - Mittel-dBmWert (m) = g.M. des Mittelwertes im Durchlassbereich
- Mittelungszeit (count) einstellbar
  - ca. 0.3 s bis 1.5 s
- Audiosamples werden mit agc multipliziert

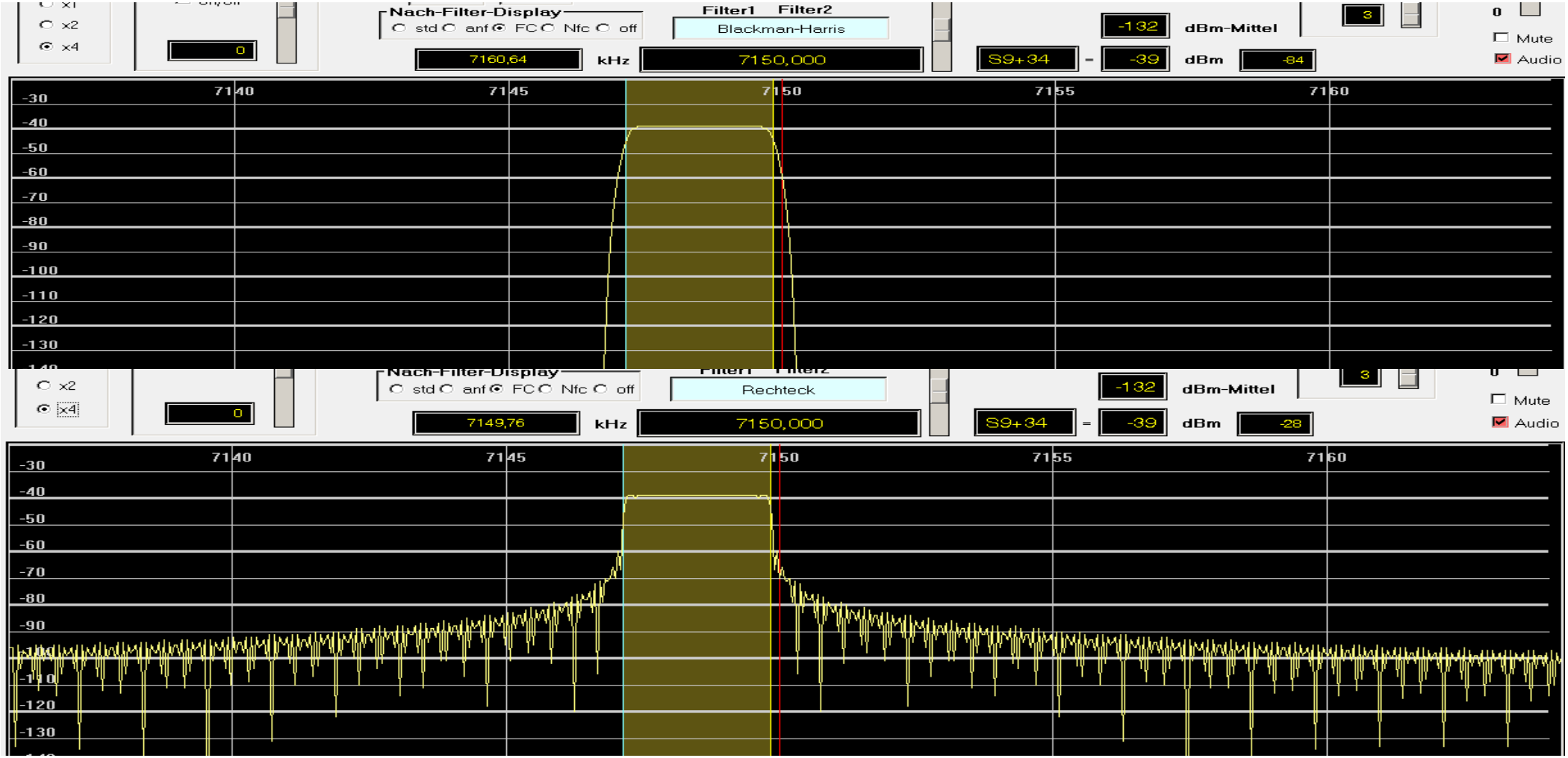
Notchfilter (1)



Notchfilter (2)

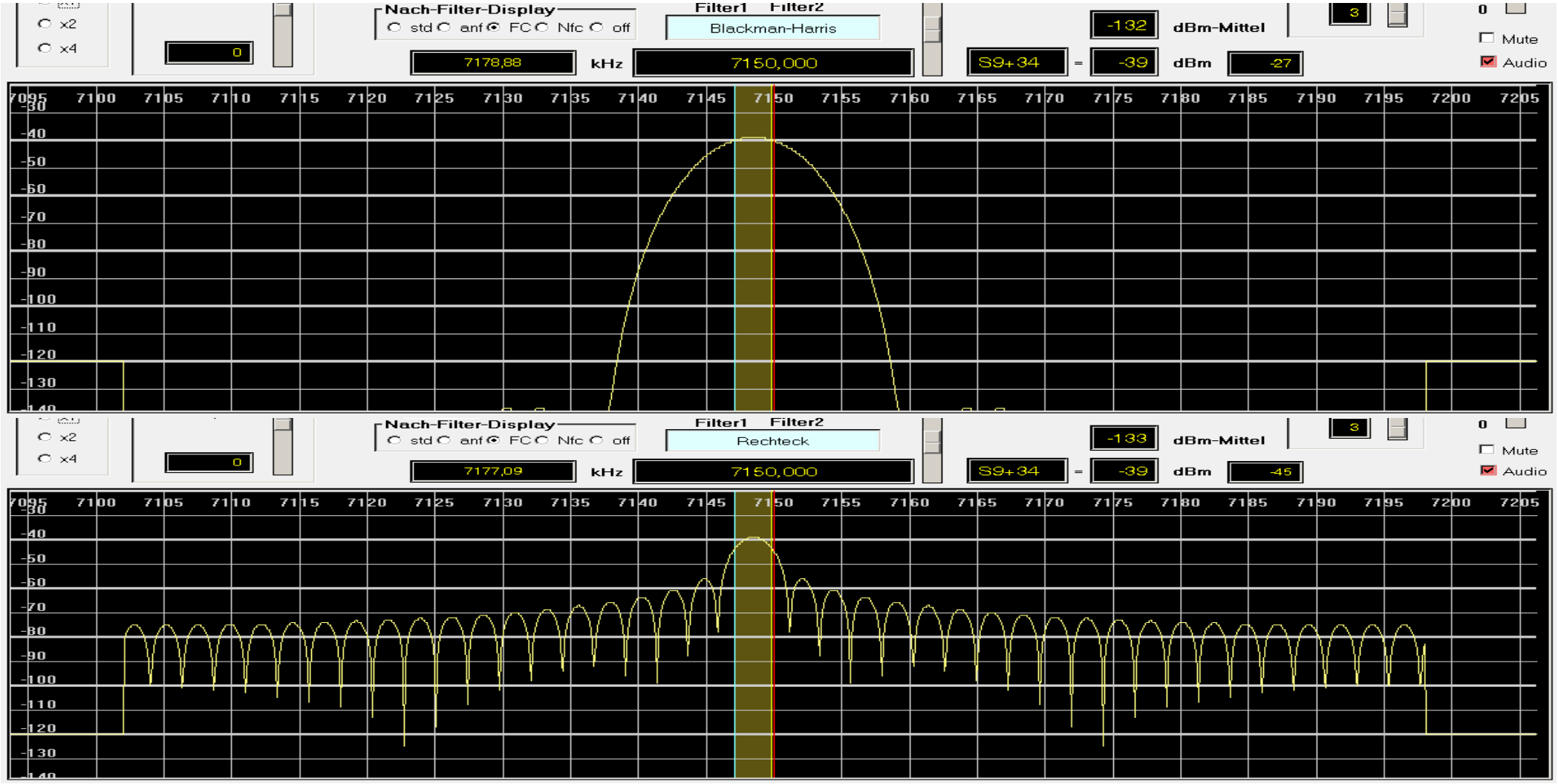
- 1)Detektion eines störenden Trägers im Durchlassbereich
  - durch Differenzenbildung Feststellung von Spitzen und dann Zählung der Spitzen über viele Buffer
  - gibt ein recht zuverlässiges Signal für die Notchfrequenz fn (in der Praxis sogar besser als bei PowerSDR)
- 2) schmalbandige Filterung am Ort des Störträgers
  - Benutze ein Tiefpassrechteckfilter H(z) mit c=0, Blackman-Harris, dann Komplementbildung 1 – H(z), N+1=501
  - Verschiebung an fn: gibt ein brauchbares Notchfilter

FIR im Frequenzraum N+1=901



Oben: Blackman-Harris, Unten: Rechteck, FIR-Länge 901 bei beiden!

FIR N+1=41



Oben: Blackman-Harris, Unten: Rechteck, FIR-Länge 41 bei beiden!

Erweiterungen:
ein Amateur bleibt nie stehen...

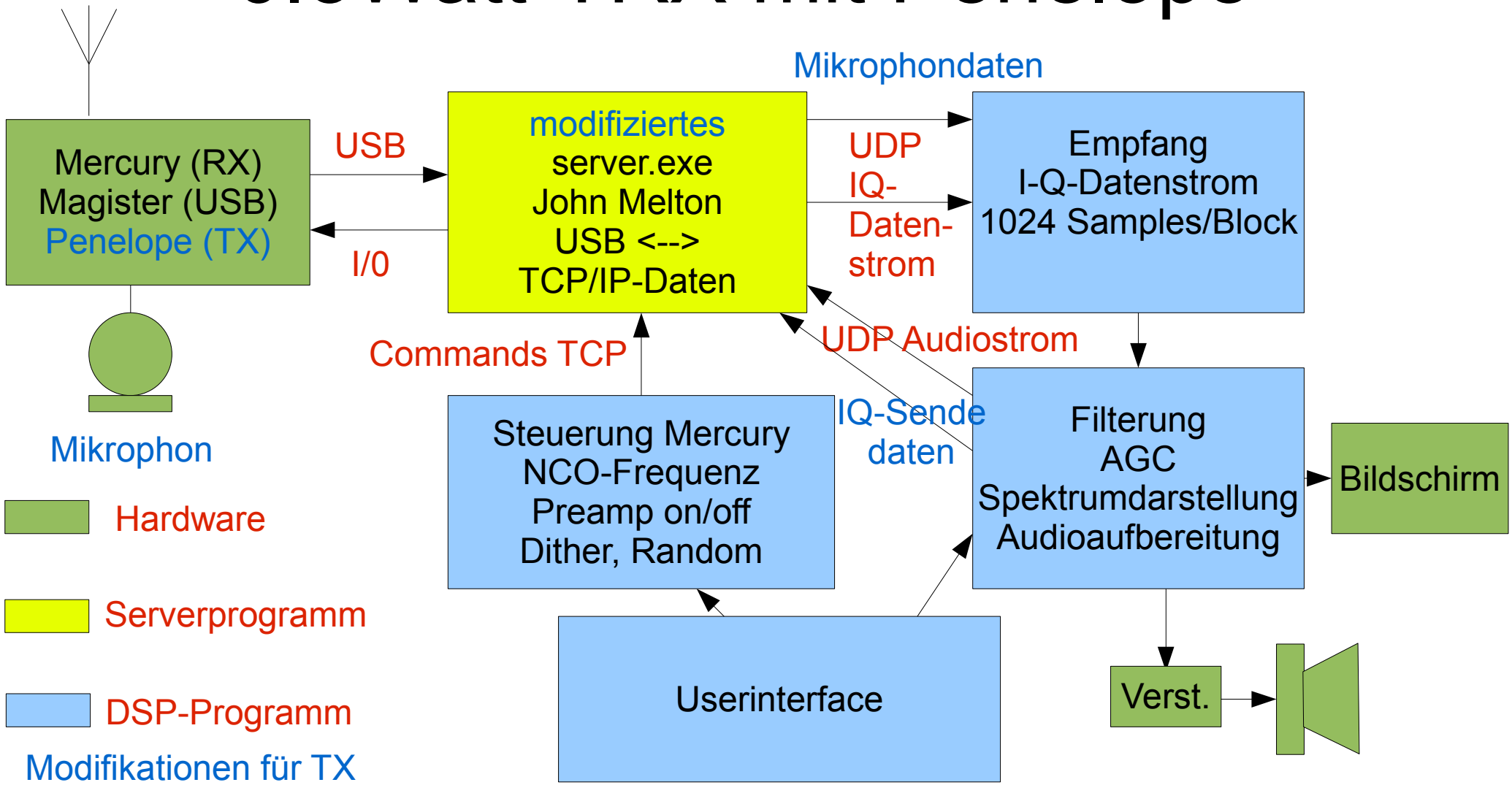
Erweiterungen DSP-Programm (1)

- Senden mit Penelope
  - Penelope ist ein 0.5-Watt-Digital-Upconverter-Steuersender, ebenfalls auf dem ATLAS-Bus
  - Digitalisiertes Eingangssignal vom Mikrophon ist reell (Imaginärteil =0), kommt von Penelope
  - Mittels USB- oder LSB-FIR-Filter daraus I und Q erzeugen
    - braucht also keine neuen Filter-Rechnungen
    - der RX liefert schon alles! (Siehe Demo)
  - via server.exe an Penelope durchreichen
  - das läuft schon, Bausatz GH-01-PA bis 50 Watt in Arbeit

Penelope


HF-out
Mikrophon, PTT
PTT out,
Open
Collector
Outputs
etc.
z.B. f. Antennen-
umschaltung
ALTERA
Cyclone II
PENELOPE DUC TX
http://hpsdr.org
http://tapr.org/OHL
HPSDR
Rev A

0.5Watt-TRX mit Penelope



Erweiterungen DSP-Programm (2)

- Mehrwegeempfang
  - mehrere Mercury zusammen
  - Antennenauswertung phasensynchron
  - primäres Ziel bei mir: Störunterdrückung
  - das ist derzeit in Arbeit
- Erweiterung auf sr=192kHz und sr=48kHz
  - Mercury kann 48, 96 und 192 kHz
  - DSP-Programm derzeit aber fix auf 96kHz

Erweiterungen DSP-Programm (3)

- Weitere Ideen und Möglichkeiten
  - Predictionfilter zur Korrelation von Störsignalen
    - Korrelation zwischen Durchlassbereich und ausgewähltem Störsignalbereich mit identischem Filter
    - Ziel: Unterdrückung systematischer Störsignale im Durchlassbereich
  - A/D-Wandler-Auswertungen
    - statistische Messungen von statischen und dynamischen Linearitätseigenschaften
    - braucht gut definierbares, sehr stabiles Eingangssignal

SDR: Diskussion

Vielen Dank für Ihr Interesse

Anhang

Literaturangaben

Literatur (1)

- Steven W. Smith
  - „The Scientist and Engineer's Guide to Digital Signal Processing“
    - ausgezeichnete Einführung in DSP, auch für Nichtmathematiker
    - http://www.dspguide.com
- Richard Lyons
  - "Quadrature signals: Complex, but not complicated"
    - Einführung in die DSP-Arbeit mit komplexen Zahlen
    - http://www.dspguru.com/dsp/tutorials/quadrature-signals

Literatur (2)

- Wikipedia-Artikel
  - http://de.wikipedia.org/wiki/Fensterfunktion
    - Erklärung und Beispiele von Fensterfunktionen
  - http://de.wikipedia.org/wiki/CIC-Filter
    - Downsampling
  - http://de.wikipedia.org/wiki/Cordic
    - Berechnung von sin(*) und cos(*) für FPGAs
  - http://de.wikipedia.org/wiki/Sinc-Funktion
    - Fourier-Transformierte des Rechteckfilters

Weitere Adressen (1)

- HPSDR-Projekt:
  - http://openhpsdr.org
- Octave: Numerikprogramm
  - freies Programm (GNU-Lizenz)
  - gleiche Syntax und Funktionen wie MATLAB
    - somit laufen MATLAB-Programme
      - wenige Ausnahmen
  - gut geeignet für die Berechnung von Filterbeispielen
  - http://www.octave.org

Weitere Adressen (2)

- Altera-Application Note Nr. 455
  - "Understanding CIC compensation filters"
    - letzte Filter-Stufe im FPGA der Mercury ist ein CFIR
    - für die Kompensation des Frequenzgangs der CIC-Stufen
    - http://www.altera.com
- Diskrete Fourier-Transformation
  - "Fastest Fourier Transform in the West"
    - http://www.fftw.org