



TECHNISCHE  
UNIVERSITÄT  
DRESDEN

# Dekodierung eines Funkfernsehersignals mithilfe der Zynq-Plattform

Lehrstuhlseminar Benjamin Koch

Dresden, 27.08.2015



DRESDEN  
concept  
Exzellenz aus  
Wissenschaft  
und Kultur

## Gliederung

- Aufgabenstellung
- Funkfernschreiben (RTTY)
- Aufbau des Systems
- Fazit und Ausblick

## Aufgabenstellung

- Eingangsseitig gegeben 24 Bit Audiosamples
- Samplerate 94 kHz
- Aufbereitung/Verarbeitung des Signals
  - Filterung
  - FFT
- Dekodierung (zunächst max. 50 Baud)
- Ausgabe



## Aufgabenstellung

### Zynq-7000 von Xilinx

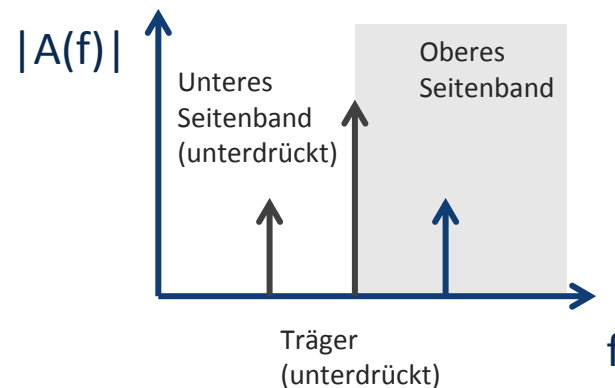
- Kopplung von programmierbarer Logik mit einem ARM Dualcore auf Cortex-A9-Basis
- System-on-a-Chip
- Optimiertes Hardware/Software-Codesign
- Entwicklungssystem: ZedBoard



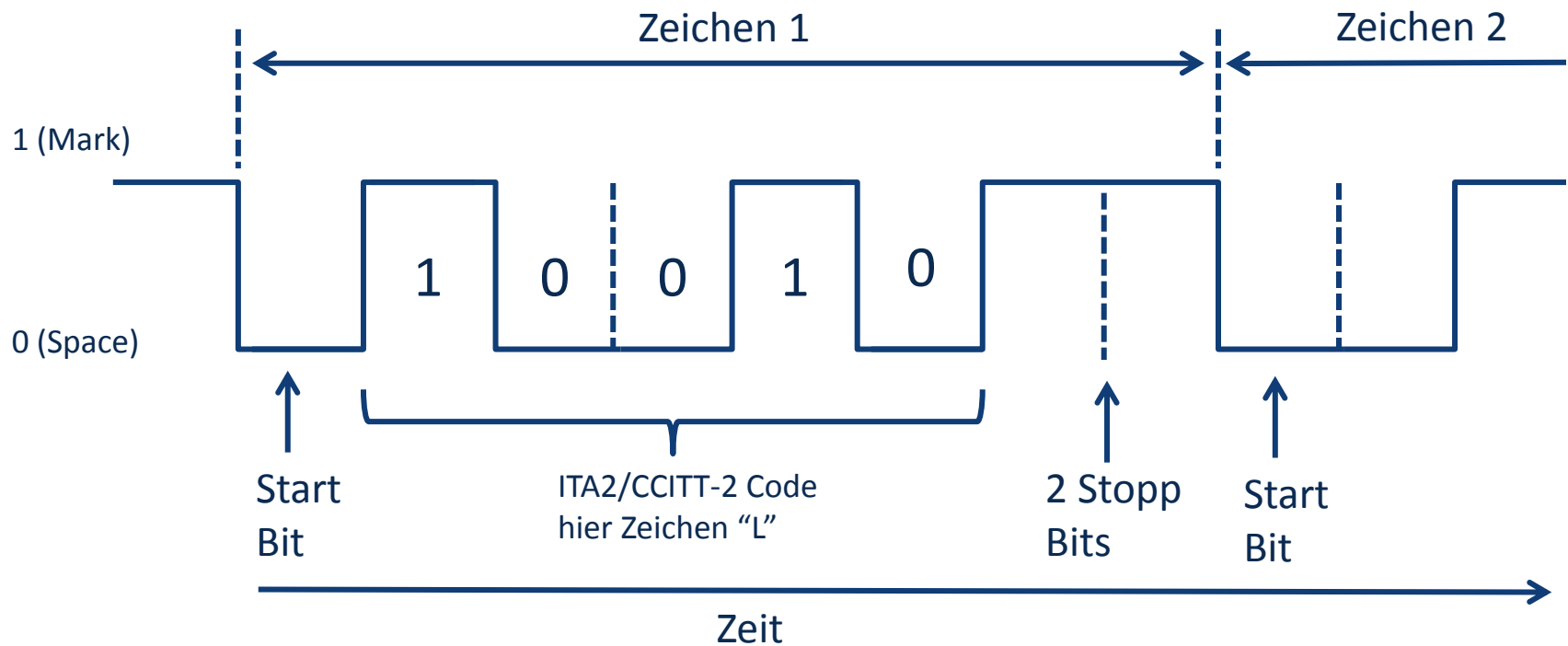
Abbildung 1  
ZedBoard

## Funkferschreiben (Radio TeleTYpe)

- Asynchroner serieller Bitstrom
- 1 und 0 durch 2 Töne signalisiert
- 5-Bit-Baudot Code
- 1 Startbit, 1, 1.5 oder 2 Stoppbits
- Baudrate üblicherweise zwischen 10 und 1000 Baud
- Modulationsart: Einseitenbandmodulation (SSB)



# Funkferschreiben (Radio TeleTYpe)



## Funkferschreiben (Radio TeleTYpe)

- CCITT-2 Code
- Seit 1932 als internationales Telegraphenalphabet Nr. 2 standardisiert

Tabelle 1

CODE	Buchstabe	Ziffern/Zeichen
00011	A	-
11001	B	?
01110	C	:
01001	D	Wer Da?
00001	E	3
01101	F	unbenutzt
11010	G	unbenutzt

## Funkferschreiben (RTTY)

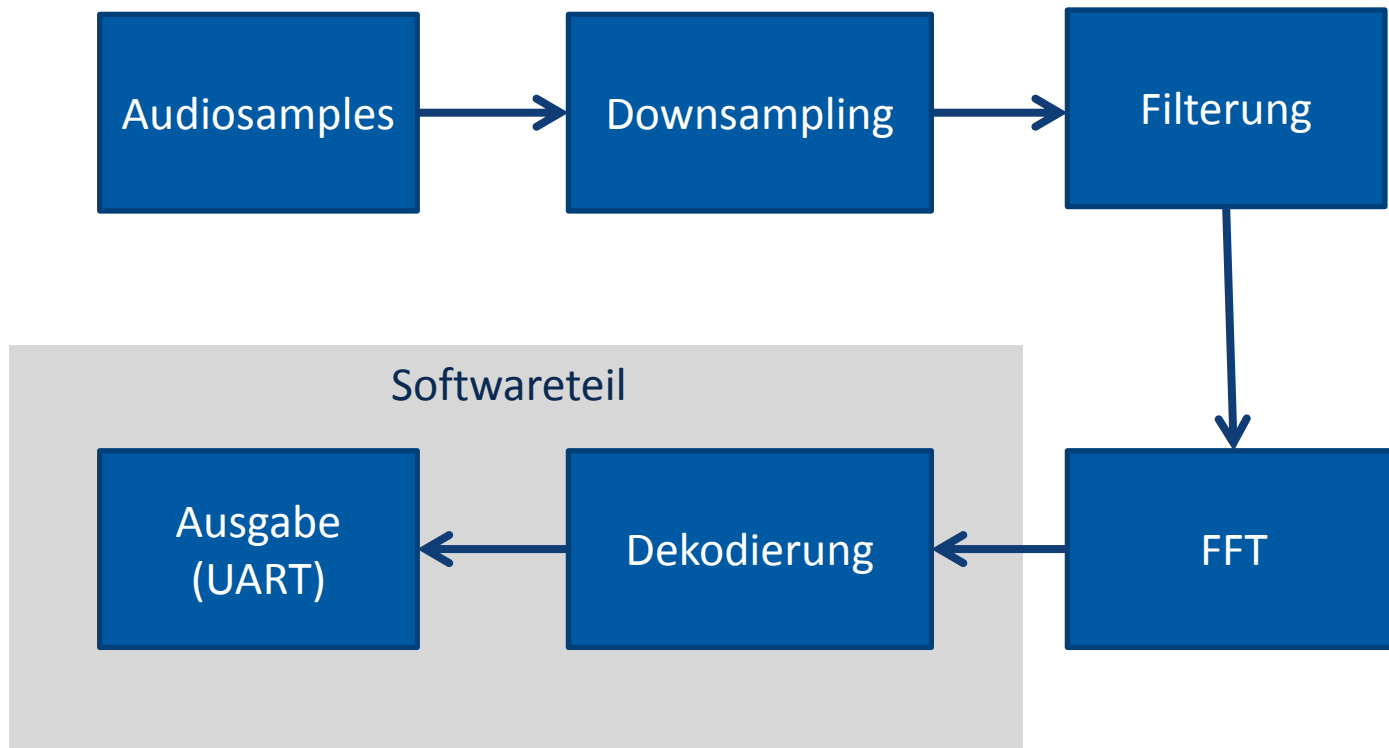
- Heute weitestgehend durch modernere Verfahren ersetzt
- Beliebt bei Funkamateuren
- Wetterdaten des deutschen Wetterdienstes
- Militärische Nutzung

Abbildung 2  
Sender Pinneberg des  
deutschen Wetterdienstes





## Aufbau des Systems

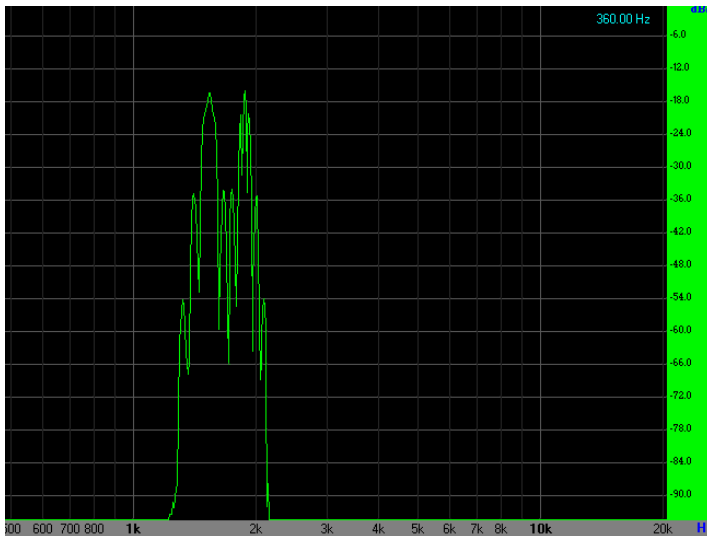


# Aufbau des Systems



$$y_1(t) = y_0 \cos(2\pi f_1 t)$$

$$y_2(t) = y_0 \cos(2\pi f_2 t)$$



Transformation in den Frequenzbereich

$$Y_1(f) = \frac{1}{2} (\delta(f + f_1) + \delta(f - f_1))$$

$$Y_2(f) = \frac{1}{2} (\delta(f + f_2) + \delta(f - f_2))$$

$\delta$  Diracsche Deltafunktion

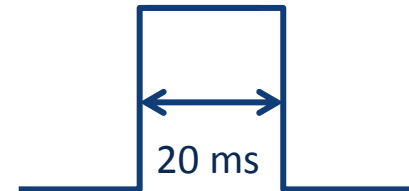
## Aufbau des Systems

- Eingangssignal 96-kHz-Audiodaten mit 24 Bit
  - Maximale Frequenz 48 kHz (Shannon-Theorem)
- Zu erkennende Signale aber nur zwischen 0 und 2 kHz
- Downsampling auf 16 kHz
  - Gelockerte Anforderungen an die FFT
  - Reduzierte Komplexität der Berechnung
- 5 Abtastwerte pro Bit

## Aufbau des Systems

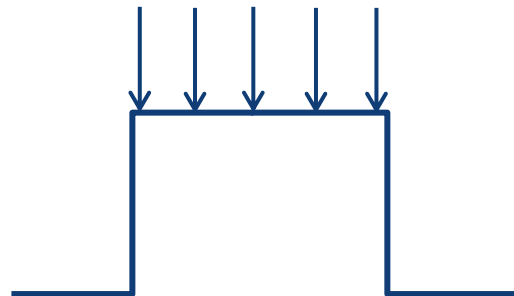
$$T_{\text{bit}} = \frac{1}{B} = \frac{1}{50 \times \text{s}^{-1}} = 20 \text{ms (Symboldauer)}$$

$$f_s = 16 \text{kHz (Samplerate)}$$



Festlegung Größe der FFT: 60 Punkte

$$T_{\text{st}} = 60 \times f_s = 3,75 \text{ms} \rightarrow \text{jedes Bit wird etwa 5-mal abgetastet}$$



## Aufbau des Systems

### Design Flow von Xilinx (Vivado)

- Entwurf eines IP-Cores für die programmierbare Logik
- Verdrahten der Systemkomponenten in einem Block Design
- Programmierung der Software
  
- Kommunikation der Komponenten via AXI Bus
- Datenaustausch über Block RAM

# Aufbau des Systems

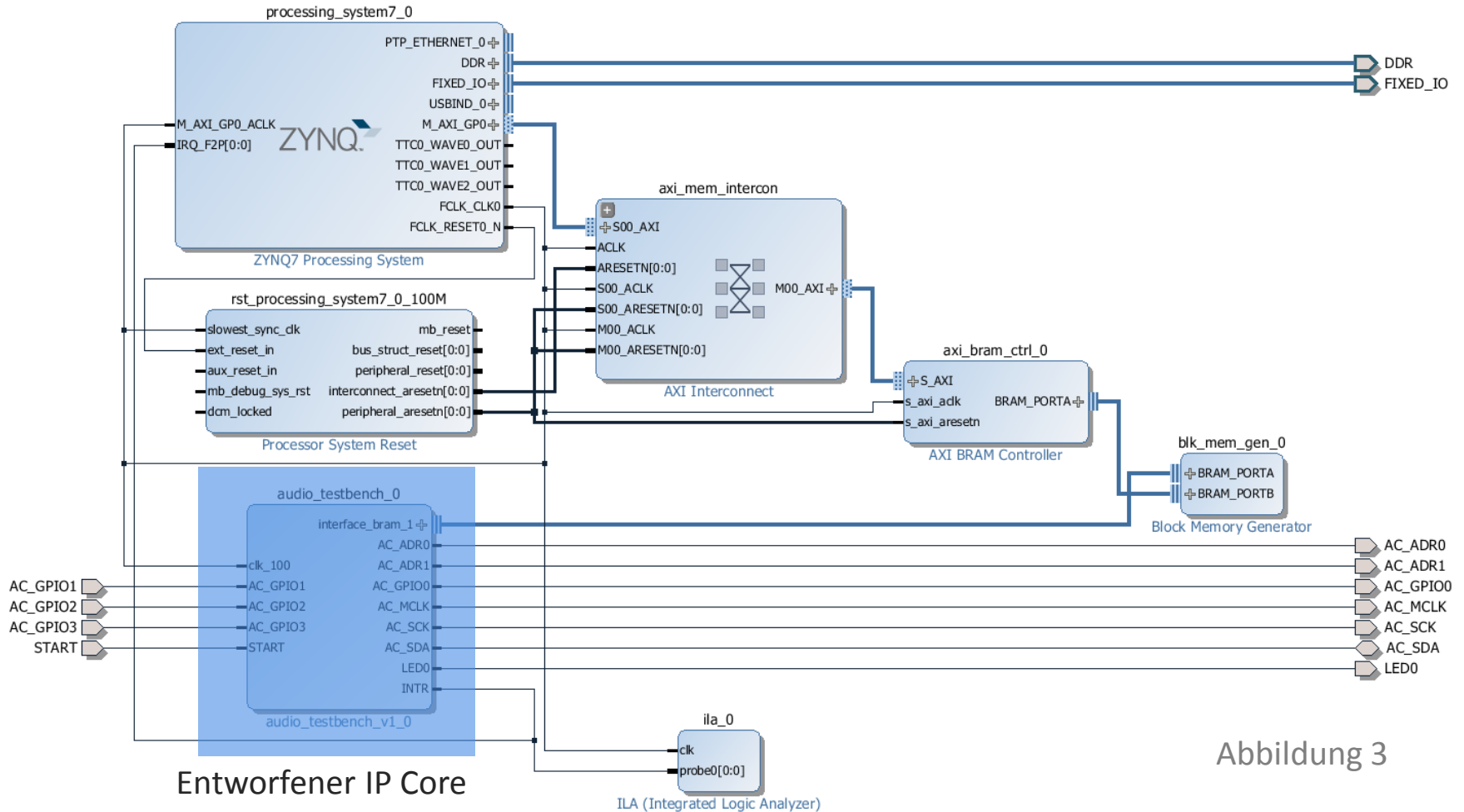
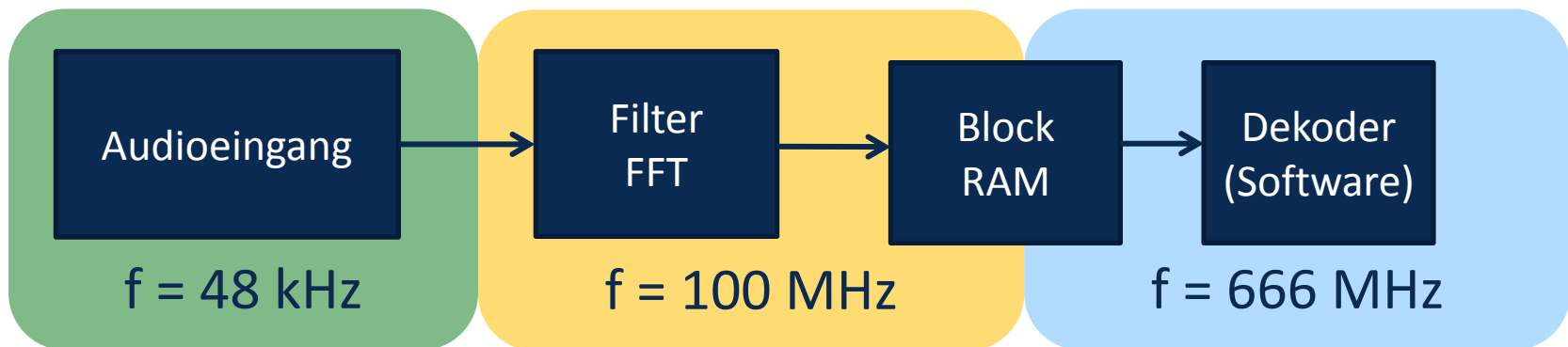


Abbildung 3

## Aufbau des Systems

- FFT, FIFO, BRAM, BRAM Controller IP von Xilinx
- Audio Interface von Stefan Scholl, Lehrstuhl Entwurf Mikroelektronischer Systeme, TU Kaiserslautern
- Takt: 100 MHz, Steigerung möglich
- Verwendung von Cross Clock FIFOs/BRAM zur Kopplung der Clock Domains



## Fazit und Ausblick

### Performance

- Bei aktueller Anwendung (RTTY, 50 Baud) noch keine Beschleunigung durch Hardware-Implementierung notwendig
- Bei komplexeren Betriebsarten oder hohen Datenraten durchaus sinnvoll
- Nutzung als Software Defined Radio denkbar

Benchmark 4096 point FFT – Complex 32 bit floating point (Zynq 7020)

ARM processor alone	NEON SIMD engine	Hardware in PL fabric
830 $\mu$ S	571 $\mu$ S	129 $\mu$ S

Tabelle 1



## Fazit und Ausblick

- Bisher nur sehr grobe Filterung des Signals
- Implementierung weiterer Betriebsarten denkbar (CW, SSTV, Digital)
- Peripherie des ZedBoards besser nutzen
  - Bildausgabe (VGA/HDMI)
  - Ethernet
- Einsatz unter Linux

## Fazit und Ausblick

- Arbeit mit Vivado aufgrund von Bugs anstrengend
- Schlechte Dokumentation für grundlegende Schritte
- Sehr guter Support im Xilinx Forum
- C-Modelle von Xilinx zur Simulation sehr hilfreich

# Quellen

## Abbildung 1

[http://zedboard.org/sites/default/files/styles/product\\_slider/public/product/ZedBoard\\_RevA\\_sideA\\_0\\_0.png?itok=lsIF6leb](http://zedboard.org/sites/default/files/styles/product_slider/public/product/ZedBoard_RevA_sideA_0_0.png?itok=lsIF6leb)

## Abbildung 2

[https://de.wikipedia.org/wiki/DDH47#/media/File:DDH47\\_25112012\\_1.JPG](https://de.wikipedia.org/wiki/DDH47#/media/File:DDH47_25112012_1.JPG)

## Abbildung 3

Screenshot des Block Designs aus Vivado

## Tabelle 1

[https://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-S.1-198811-S!!PDF-E&type=items](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-S.1-198811-S!!PDF-E&type=items)

## Tabelle 2

<http://www.wiki.xilinx.com/Zynq-7000+AP+SoC+Spectrum+Analyzer+part+1+-+Accelerating+Software+%26+More+-+Installing+and+Running+the+Spectrum+Analyzer+Demo+Tech+Tip>

# Quellen

**Rumpf, Karl-Heinz**

**Trommeln, Telefone, Transistoren. Ein Streifzug durch die elektrische Nachrichtentechnik.  
Verlag Technik, Berlin 1971**

**Professor Fettweiß**

**Vorlesung Nachrichtentechnik**

[https://mns.ifn.et.tu-dresden.de/Teaching/Courses/NT\\_Documents/Nachrichtentechnik\\_Skript\\_2015\\_De.pdf](https://mns.ifn.et.tu-dresden.de/Teaching/Courses/NT_Documents/Nachrichtentechnik_Skript_2015_De.pdf)

**Stefan Scholl**

**Audio Interface for ZedBoard**

[https://kluedo.ub.uni-kl.de/frontdoor/deliver/index/docId/4034/file/zedboard\\_audio\\_doc.pdf](https://kluedo.ub.uni-kl.de/frontdoor/deliver/index/docId/4034/file/zedboard_audio_doc.pdf)

[https://github.com/ems-kl/zedboard\\_audio](https://github.com/ems-kl/zedboard_audio)

