

Digitale Bestandserhaltung von Datenformaten und Speichermedien historischer Drum-Machines

Klaus Michael Indlekofer

Hochschule RheinMain (HS-RM), FB ING / Medientechnik

**Dieses Dokument unterliegt der Creative Commons Lizenz
CC BY-NC-ND 4.0 (Namensnennung-nicht kommerziell-Teilen-keine Bearbeitungen)**

Inhalt

1. Einleitung
2. Audio-Magnetbänder als Datenspeicher
3. Software-Defined Decoder-Konzept
4. Software-Defined Encoder-Konzept
5. Beispiel: LinnDrum
6. Zusammenfassung

1. Einleitung

Historische Audiosysteme: Drum-Machines, Synthesizer, Orgeln, ...

typ. Datenmengen ~1-10KB für Drum-Patterns, Sequenzen, Arrangements, Sound-Programme, Patch-Daten, Speicherinhalte ("Memory Dumps").



Typische historische Methoden der kostengünstigen Datenspeicherung:

- ROM- und RAM-Cartridges (meist proprietär)
- Audio-Magnetband, z.B. Compact Cassette (preiswert, wenig CPU-Belastung)
- Disketten ("Floppies", größerer Datenmengen, jedoch teuer 1970er-1980er)
- RS232-Datenpakete, gespeichert auf Host-Computer (jedoch teuer 1980er)
- MIDI System Exclusive Messages (mit MIDI Sequencer/Tool, erst ab Mitte 1980er)



Audio-Magnetband, z.B. Compact Cassette ("Datencassette") als Datenspeicher

war teilweise die einzige Möglichkeit einer externen Datenspeicherung/Backup

- "Brücke" zu moderner Technologie durch Decoder/Encoder-Tools erforderlich zur Wiederherstellung/Konvertierung historischer Kompositionen/Arrangements etc.
- Konvertierung in moderne Formate (WAV, SMF, ...) zur Nutzung von Patterns etc. auf modernen Systemen oder zur Wiederverwendung auf Original-Equipment

Herausforderung: Codierung und Protokolle sind meist proprietär und nicht dokumentiert

- Notwendigkeit der individuellen Analyse unter Verwendung alter oder neuer Aufzeichnungen, sowie Tests auf Original-Equipment
- Software-Defined Decoder- und Encoder-Konzepte
- implizite Dokumentation für die Zukunft durch eine Open-Source-Implementierung

2. Audio-Magnetbänder als Datenspeicher

Prinzip: „Töne für Bits“



verwendet geeignete **Pseudo-Audiosignale** zur Datenspeicherung auf Audio-Magnetband

Merkmale der analogen Audioaufzeichnung auf Magnetband:

- lineare Verzerrungen: sub-optimaler Frequenzgang (Amplitude und Phase)
 - nicht-lineare Verzerrungen (Rechtecksignale sind hier jedoch recht immun)
 - Rauschen und Brummen, Vor/Nachechos (Kopiereffekt)
 - Dropouts (Aussetzer), Degradation/Alterung
 - Geschwindigkeits-Abweichungen und -Schwankungen („Wow & Flutter“)
- mögliche Ursachen für Datenfehler

Anforderungen an geeignete Modulationsverfahren:

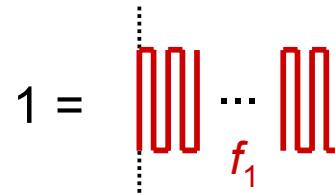
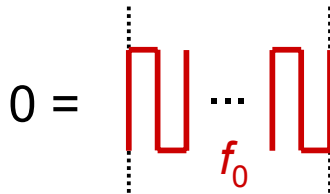
- Unabhängigkeit von der Signal-Polarität
- Bandbreitenbegrenzung (z.B. 100Hz...10kHz), DC-Freiheit (keine Gleichspannung)
- Toleranz gegenüber Störungen and Geschwindigkeits-Abweichungen etc.

Typische Methoden zur Erzeugung von Pseudo-Audiosignalen aus digitalen Daten:

- FSK (Frequency-Shift-Keying)
- Manchester Code
- Bi-Phase-Mark Code
- allgemeine Bit-Group-Coding-Ansätze

2. Audio-Magnetbänder als Datenspeicher

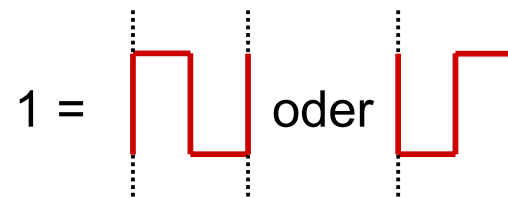
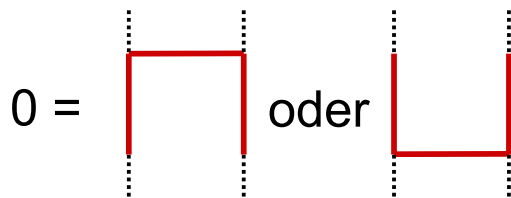
Frequency-Shift-Keying (FSK)



zum Beispiel:
 $f_0=2.4\text{kHz}$
 $f_1=4.8\text{kHz}$

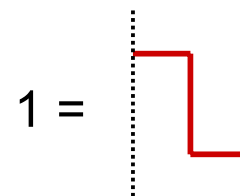
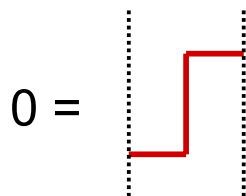
Beachte: invertiertes Signal auch möglich \rightarrow FSK ist unabhängig von Polarität.
Einfacher analoger FM-Demodulator verwendbar. Für niedrige Datenraten.

Bi-Phase-Mark Code (BMC)



Beachte: Bi-Phase-Mark Code ist unabhängig von Polarität.

Manchester Code = Phase Encoding (PE)

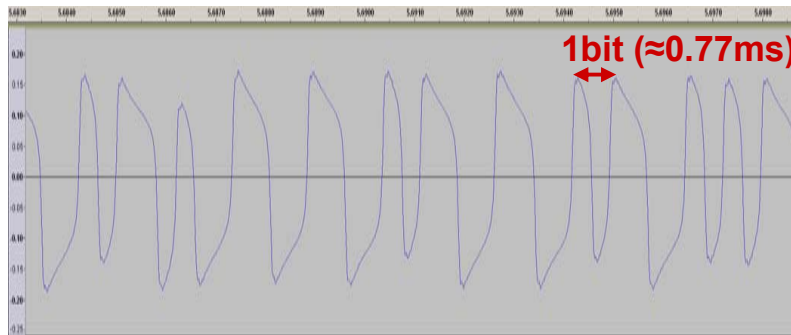


Bem.: Manchester Code ist abhängig von Polarität, Sync-Pattern erlaubt jedoch Polaritäts-Erkennung.
BMC ist äquivalent zu 1/2-Clock-verschobenem invertiert-differentiellem Manchester Code.

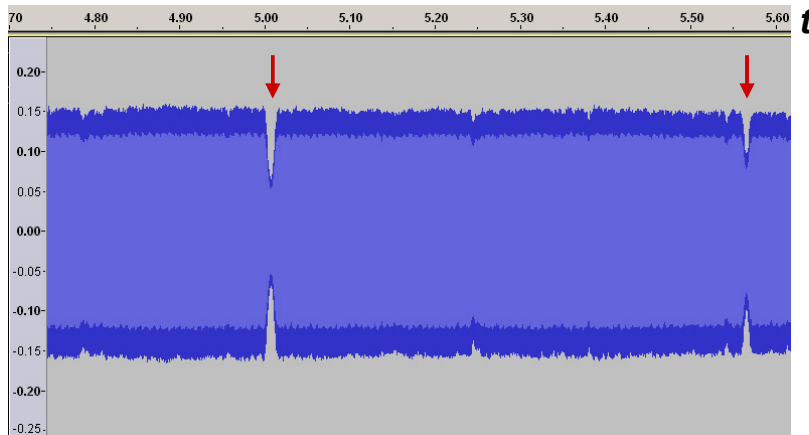
2. Audio-Magnetbänder als Datenspeicher

Beispiel: LinnDrum (Manchester Code, 1.3kbit/s)

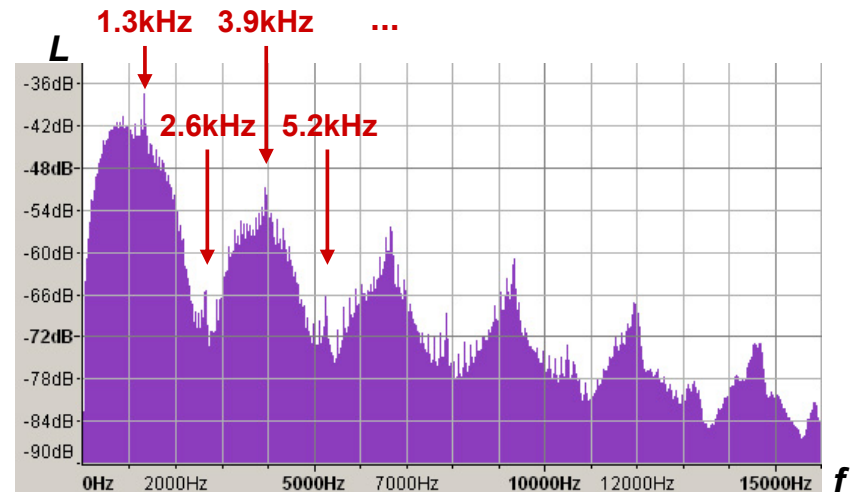
Signal vom Band (Compact Cassette):



Dropouts im Signal vom Band:



Frequenzspektrum vom Band:



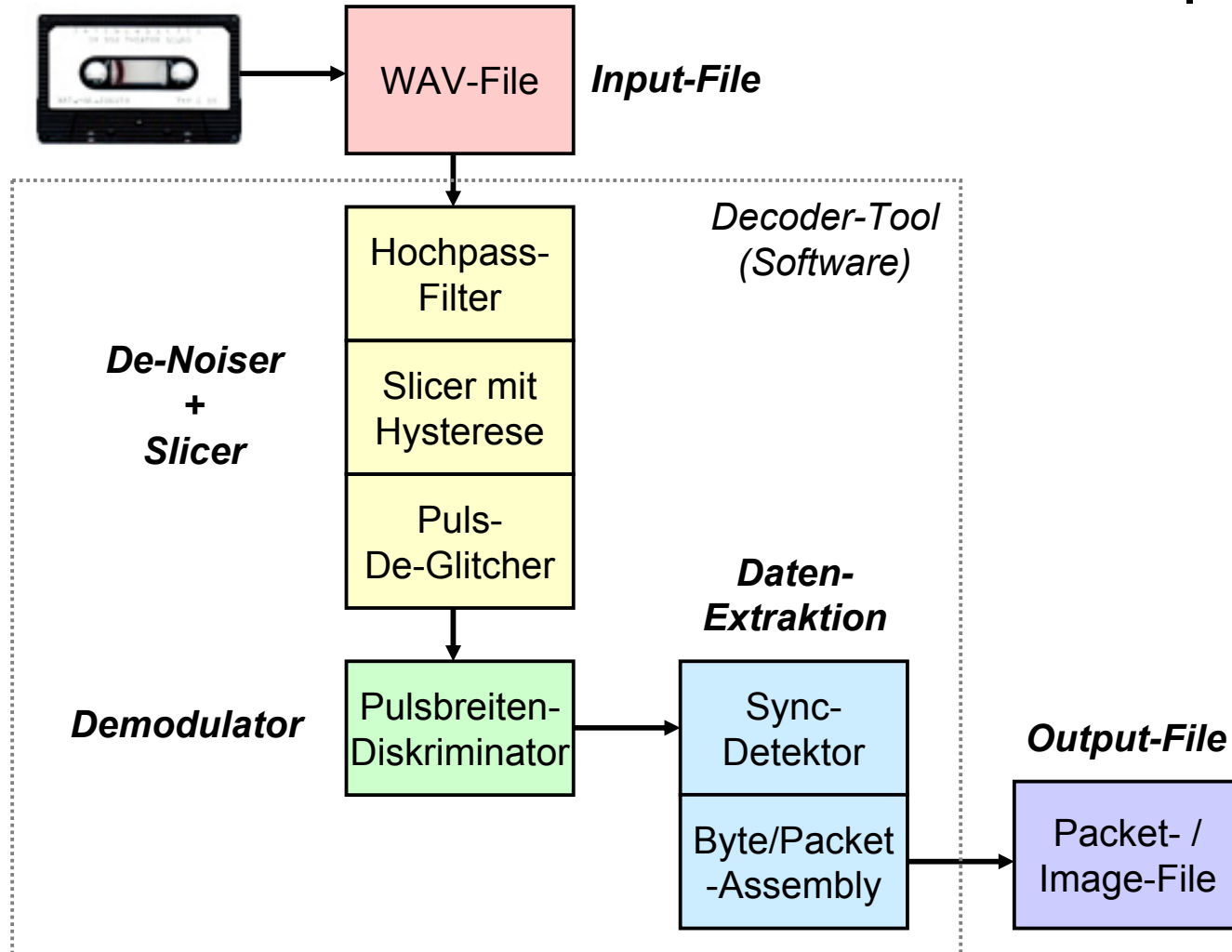
→ min. erforderliche Audiobandbreite $\approx 5.2\text{kHz}$
für Seitenbänder der 1. + 3. Harmonischen
eines Rechtecksignals

Beachte: ist DC-freies Signal ($-\infty\text{dB}$ bei 0Hz)

3. Software-Defined Decoder-Konzept

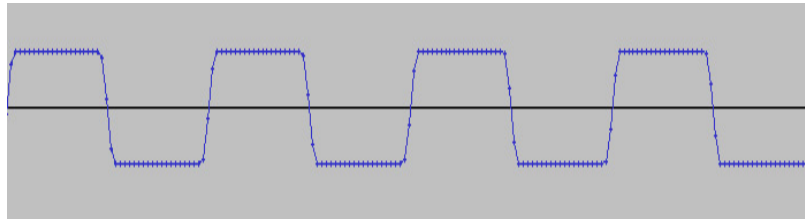
*Audio-Magnetband
z.B. Compact Cassette*

Decoder-Prinzip

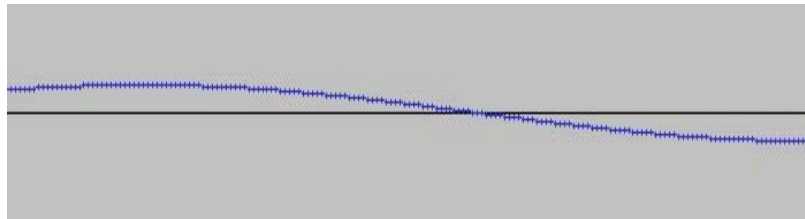


3. Software-Defined Decoder-Konzept

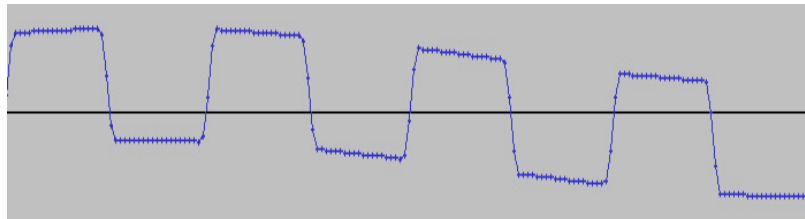
Hochpass-Filter → entfernt DC, reduziert NF-Störungen (z.B. Brumm)



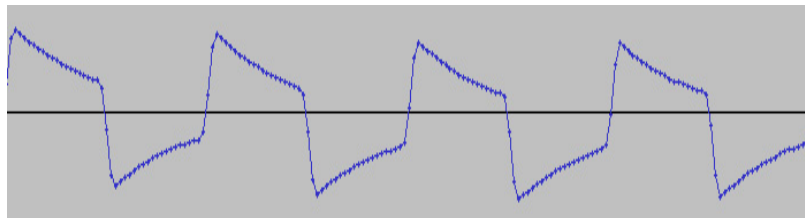
Original-Signal



NF-Störung (z.B. Brumm)



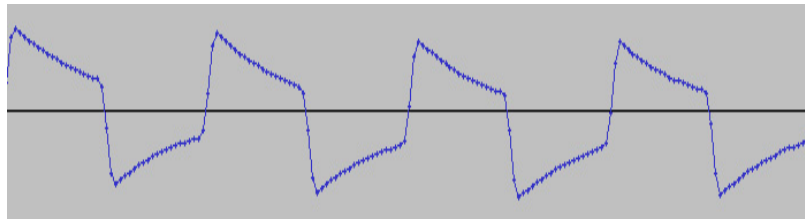
vom Band: Original + NF-Störung
(noch ohne Rauschen)



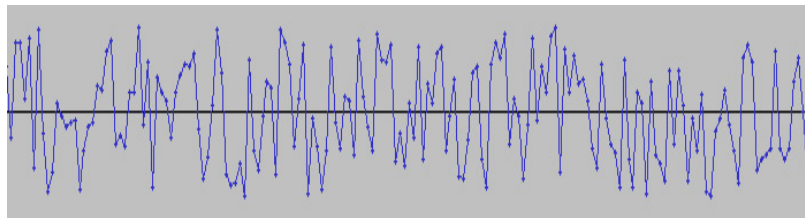
nach Hochpass-Filter
→ NF-Störung reduziert
und steilere Signal-Flanken

3. Software-Defined Decoder-Konzept

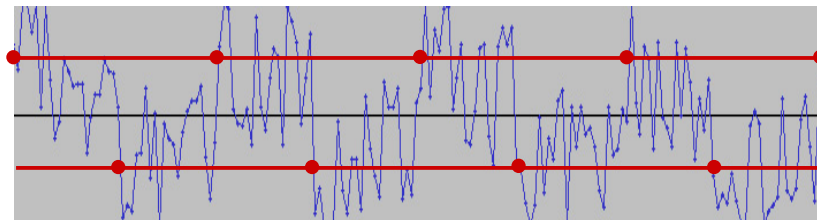
Slicer mit Hysterese → gewisse Immunität gegenüber Rauschen



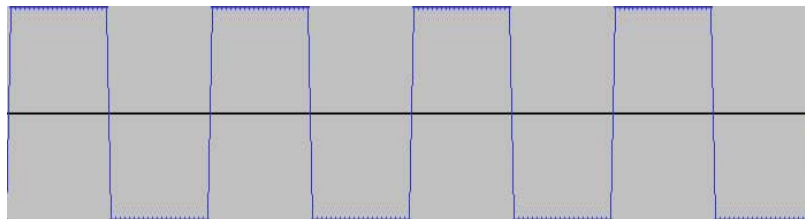
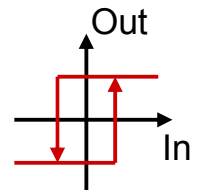
nach Hochpass-Filter
noch ohne Rauschen



Rauschen (wegen Band, Recorder, ...)



nach Hochpass-Filter
jetzt mit Rauschen

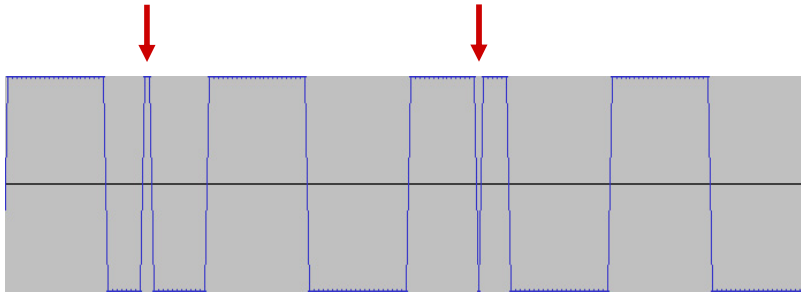


Ausgang des Slicers mit **Hysterese**
(Schaltschwellen s.o. rote Linien/Punkte)

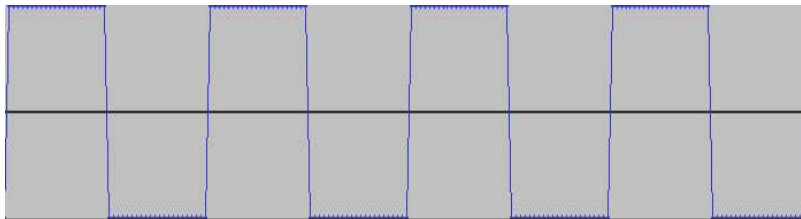
3. Software-Defined Decoder-Konzept

Puls-De-Glitcher → entfernt Pulsspitzen

Signal
Zeit

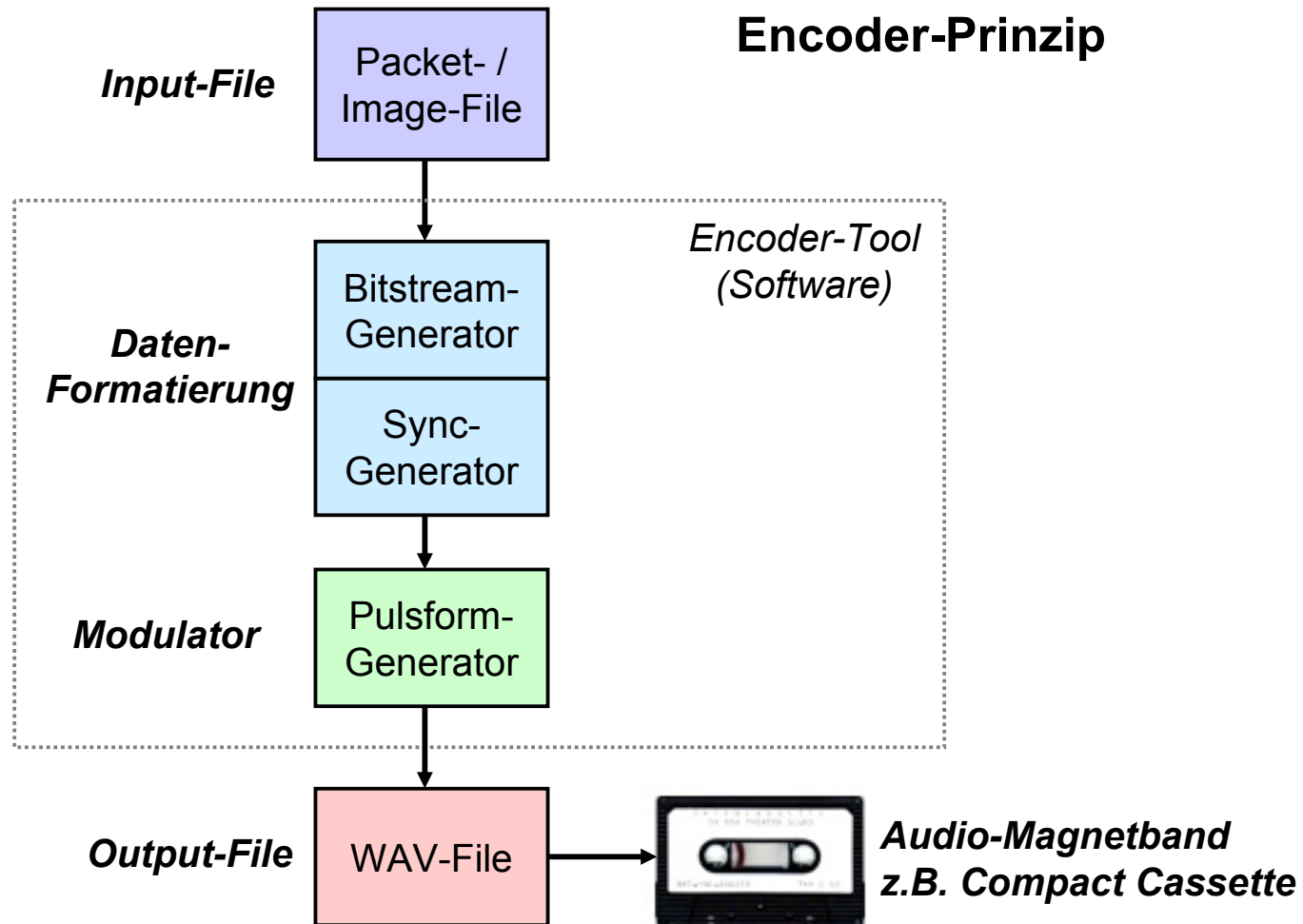


Ausgang des Slicers
mit Pulsspitzen bei zu starkem Rauschen



Ausgang des Puls-De-Glitchers
(entfernt Pulse zu kurzer Dauer)

4. Software-Defined Encoder-Konzept



5. Beispiel: LinnDrum

Linn Electronics: LinnDrum (1982)



Memory: 8KB (oder 16KB) RAM für System und 2x7x7 "Patterns" und 7x7 "Songs"

Sounds: 8bit μ -law Drum-Sound-Samples im ROM (nicht editierbar)

Timing: feste Auflösung von 48 PPQ (Pulse pro Viertelnote)

5. Beispiel: LinnDrum

Datenspeicherung auf Audio-Magnetband (z.B. Compact Cassette):

Manchester Code mit 2 Start-Bits (→ 10Bits pro Byte), 1305.5bit/s, Packet-Checksumme

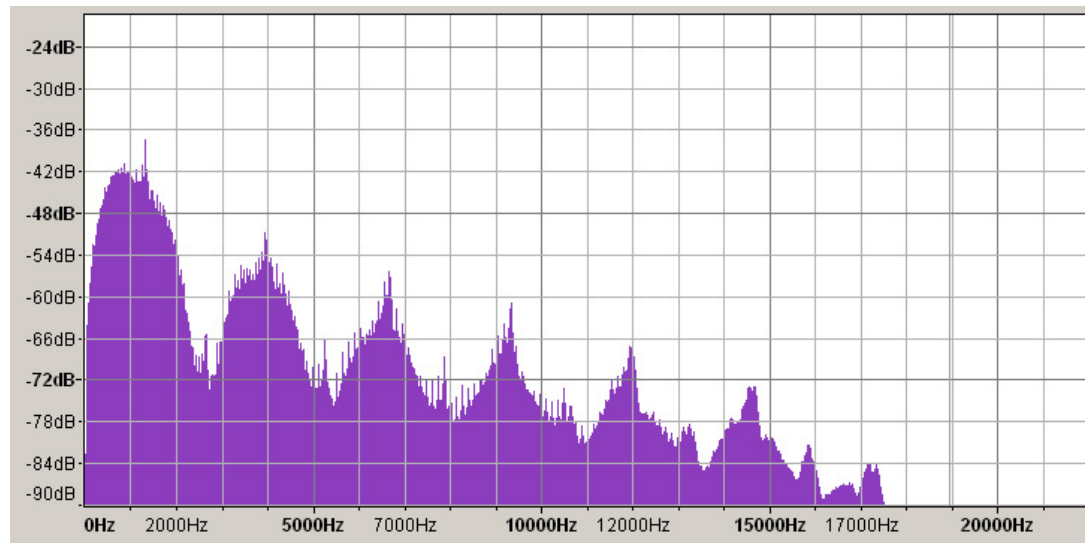
Dateninhalte: Pattern + Song Speicherinhalt ("EVERYTHING")

oder Song + zugehörige Patterns ("SONG")

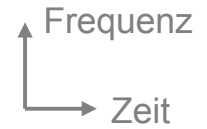
Software-Tools "linndrumcass": <https://sourceforge.net/projects/linndrumcass/>

Decoder, Encoder, Parser (zur Analyse der Daten), mit SMF (MIDI) File-Export

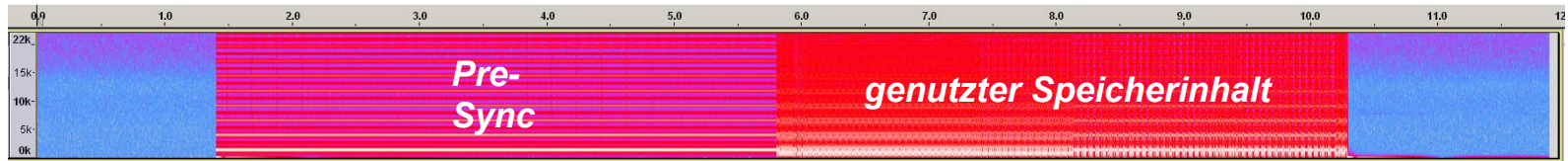
Beispiel: Signalspektrum vom Band (Compact Cassette)



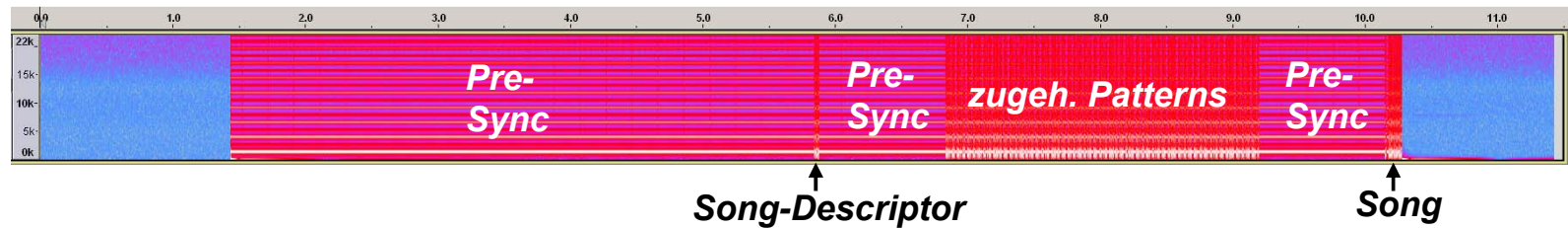
5. Beispiel: LinnDrum



Beispiel: Spectrogram für Backup-Typ „EVERYTHING“



Beispiel: Spectrogram für Backup-Typ „SONG“



Datenstrukturen:

- Song: Sequenz von Patterns
- Pattern: Sequenz von Event-Codes (inkl. Zeit-Informationen)
- Event-Code: Drum-Trigger-Code und zusätzliche Wartezeit (zum nächsten)
- Drum-Trigger-Code: 12 Drum-Sounds (teils exklusiv, einige Level/Pitch/Decay-Varianten)

Beispiel: Event-Code (3 Bytes)

0x0c 0x68 0x00 = 000000000110100000001100

	xxxxxxx	→ Δtime = 1 + 12
	xxxx	→ drums BASS2, SNAR2
	xx	→ drum HHAT2
	x	→ drum CLAP

6. Zusammenfassung

- Ziel:
Digitale Bestandserhaltung von Datenformaten und Speichermedien historischer Drum-Machines, Synthesizer, Orgeln, ...
- Pseudo-Audiosignal zur Datenspeicherung auf Audio-Magnetband
Prinzip: „Töne für Bits“
- Herausforderung:
proprietäre und undokumentierte Codierung/Protokolle
- Software-Defined Decoder und Encoder:
zur Daten-Extraktion, -Analyse, -Konvertierung
- Open-Source-Implementierung impliziert Dokumentation

Referenzen

- [1] IEC 60094 standard, parts 1-7: "Magnetic sound recording and reproducing systems"
- [2] siehe z.B. Wikipedia: "Compact Cassette", https://en.wikipedia.org/wiki/Compact_Cassette
- [3] siehe z.B. R.W.Middlestead: "Digital Communications with Emphasis on Data Modems", Wiley
- [4] MIDI Manufacturers Association: "Complete MIDI 1.0 Detailed Specification", "SMF" file format
- [5] Microsoft: "Resource Interchange File Format (RIFF)" file format, "WAVE" file format
- [6] Linn Electronics Inc.: LinnDrum Operating Instructions
- [7] Linn Electronics Inc.: LinnDrum Technical Information
- [8] E-mu Systems Inc.: Drumulator Owners Manual, Version 1.0, 1981
- [9] E-mu Systems Inc.: Drumulator Service Manual, Version 1.0, 1983
- [10] WERSI: "Drum Composer CX5 Bedienungsanleitung", BA 392
- [11] WERSI: "ALPHA DX350 Bedienungsanleitung", BA 328
- [12] WERSI: "ALPHA DX300 / ALPHA DX350 / CONDOR DX100 Technische Unterlagen", BA 329
- [13] WERSI: "ALPHA DX400/500 Bedienungsanleitung", BA 333
- [14] WERSI RS232 protocol: "WERSI, ALPHA DX350 Bedienungsanleitung", BA 328, pp.77-82
- [15] K.M.Indlekofer: "linndrumcass", <https://sourceforge.net/projects/linndrumcass/>
- [16] K.M.Indlekofer: "drumucass", <https://sourceforge.net/projects/drumucass/>
- [17] K.M.Indlekofer: "wersidatencassette", <https://sourceforge.net/projects/wersidatencassette/>
- [18] K.M.Indlekofer: "wersicx2datencassette", <https://sourceforge.net/projects/wersicx2datencassette/>
- [19] Audacity Team: "Audacity", <https://www.audacityteam.org/>
- [20] Z.Bacskó (Falcsoft): "Soundfont Midi Player", <http://falcsoft.hu/software.html#midiplayer>
- [21] K.M.Indlekofer: "Software-defined decoder and encoder for data cassette formats of historical audio systems", ISBN 978-3-9812830-9-9, 166 (2018)" 30. Tonmeister-Tagung, Köln

We are not affiliated in any way with companies/persons mentioned in this document. All brand names and trademarks are property of their respective owners. The contents of this file are intended for educational, non-commercial purposes only. The author does not take responsibility for the contents of any links referred to.