

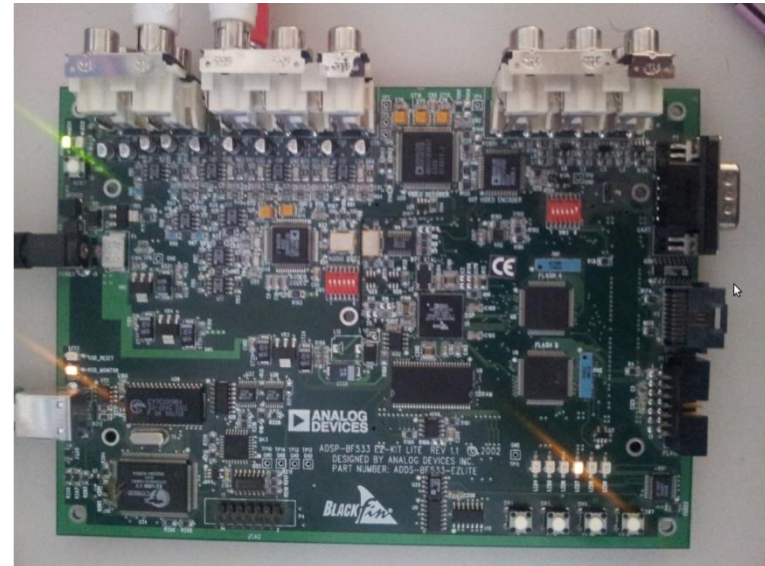
Signalprozessortechnik Projektarbeit – „Hall“

Hall Software für das EZ-Kit BF533

*Florian Grützmacher,
Simeon Wiedenmann*

Gliederung

- Aufgabenstellung
- Versuchsaufbau
- Das Prinzip digitaler Signalverarbeitung
- Was ist Hall
- Implementierung
- Demo



EZ-KIT BF533

Quelle: selfmade

Evaluation Board – Analog Devices

Aufgabenstellung

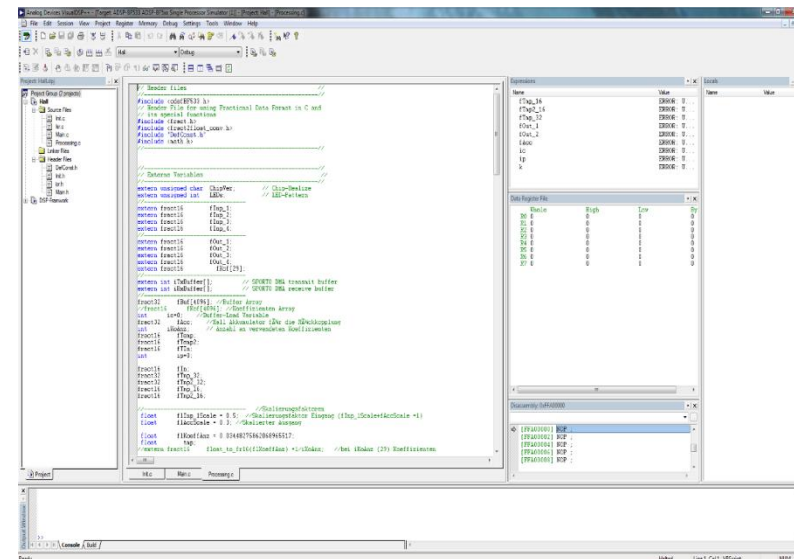
Hall-Software für das EZ-Kit BF533:

- Software zur erzeugung künstlichen Halls erstellen
- analoges Eingangssignal mit künstlichem Hall versehen
- analoge Ausgangssignal ausgeben



Quelle: selfmade

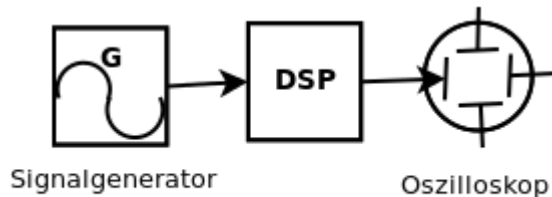
- Hausarbeit zur Projektarbeit erstellen
- Präsentation des Projektes



Quelle: selfmade

Versuchsaufbau

Implementierung:

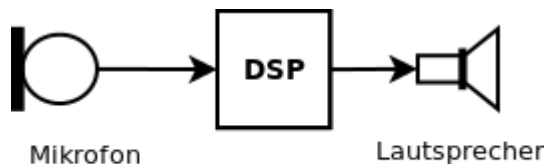


Versuchsaufbau - Implementation

Quelle: selfmade

- Signalgenerator erzeugt gewähltes Eingangssignal
- digitale Signalverarbeitung im DSP
- analoges Ausgangssignal im Oszilloskop darstellbar

Demonstration:

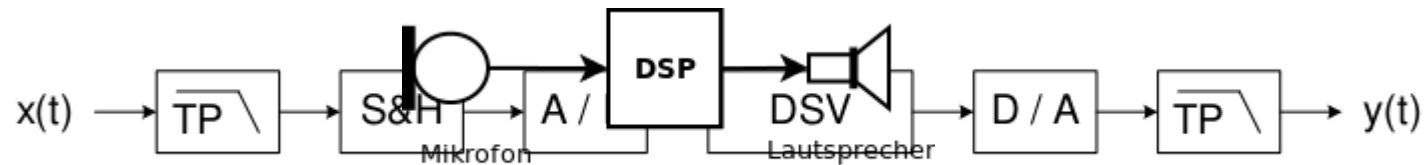


Versuchsaufbau - Demonstration

Quelle: selfmade

- analoges Eingangssignal vom Mikrofon
- mit künstlichem Hall versehen
- als analoges Ausgangssignal durch Lautsprecher ausgegeben

Das Prinzip digitaler Signalverarbeitung



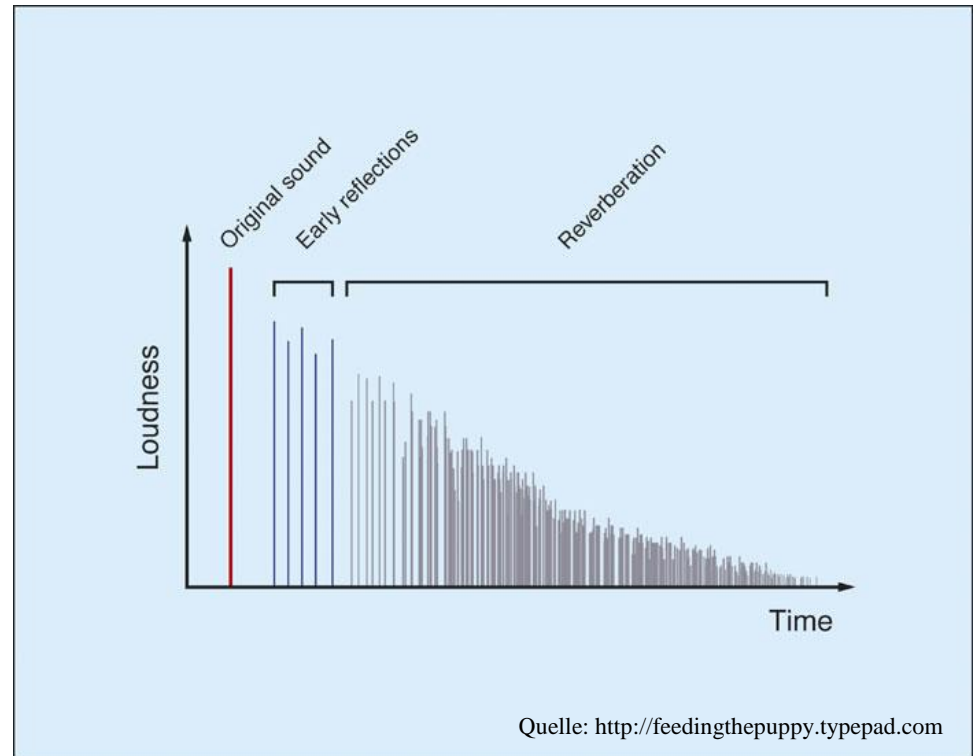
Versuchsaufbau - Demonstration

Quelle: Versuchsanleitung „Digitale Filter“

- analoges Eingangssignal $x(t)$
- Bandbegrenzung durch analogen Tiefpass
- Sample & Hold Glied
- Analog-Digitalwandler
- digitales Signal wird manipuliert
- Digital-Analogwandler
- analoger Tiefpass glättet Ausgangssignal $y(t)$

Was ist Hall

- kontinuierliche Reflexionen einer Schallwelle
- unterschiedliche Laufzeiten der reflektierten Schallwellen
- Überlagerung mehrerer, unterschiedlich stark gedämpfter Schallwellen
- Unterteilung in
 - frühe Reflexionen
 - diffuser Nachhall
- Dämpfung durch Wärmeverlust
 - Luftreibung
 - Reflektionen



Implementierung

benutzte Software:

- VisualDSP++ 5.0
- verteilte Versionsverwaltung mittels GIT
- GitHub – Webbasierter Hosting-Dienst für Software-Entwicklungsprojekte



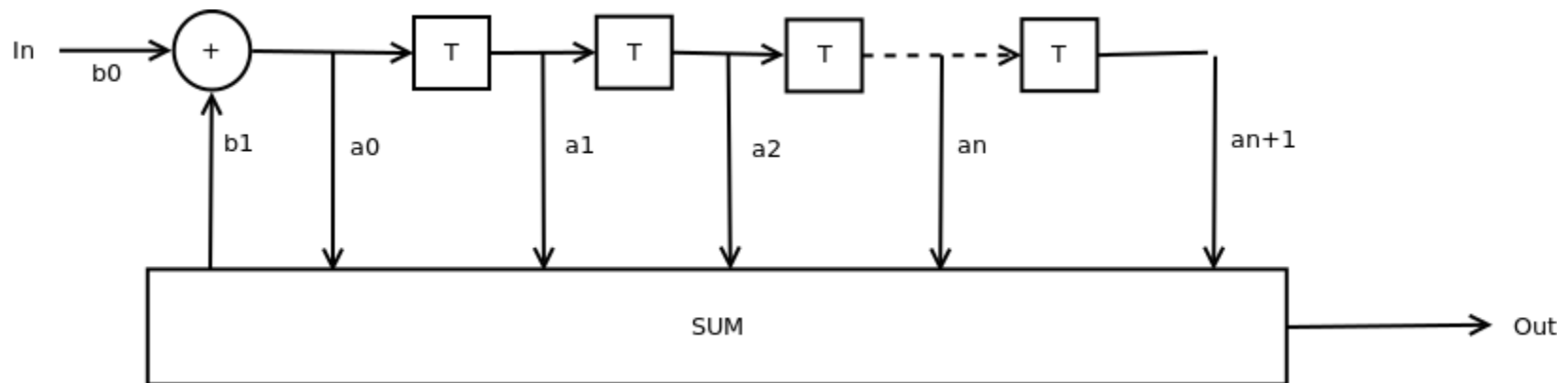
Quelle: <http://git-scm.com>



Quelle: <http://gregrickaby.com>

Implementierung

- künstlicher Hall durch Infinite-Impulse-Response Filterstruktur



Quelle: selfmade

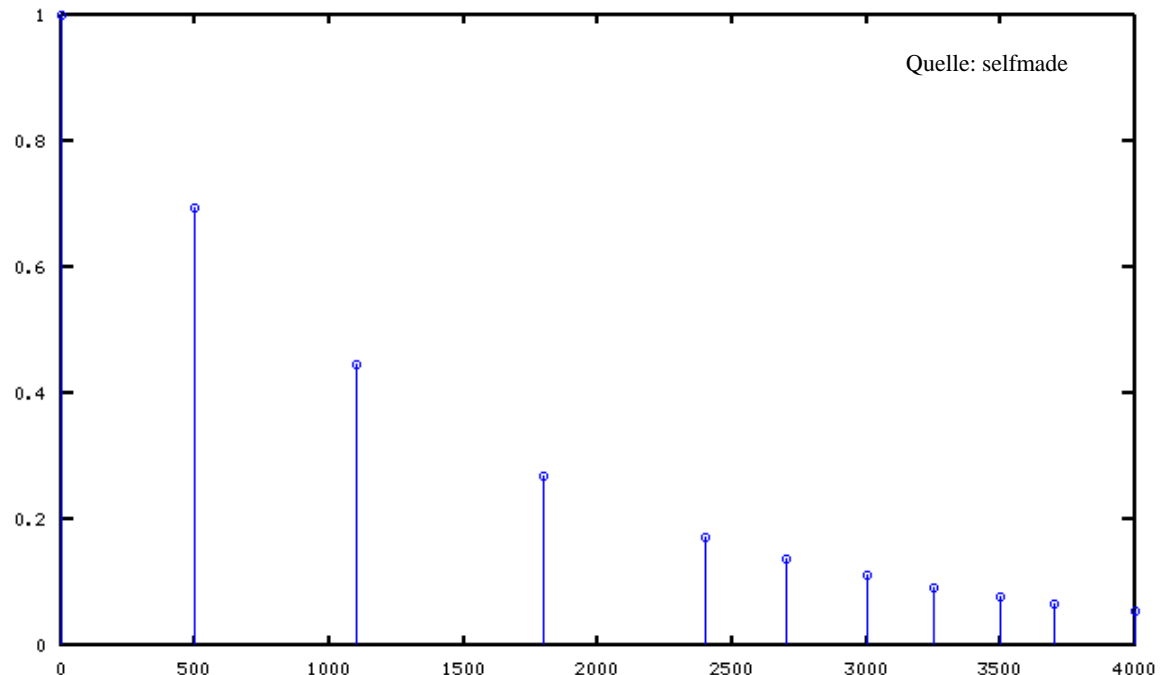
- Buffer mit 4096 Werten im fractional-32 Format
- Akkumulator im fractional-32 Format
- Rückkopplung des Akkumulators auf neuen Buffer-Wert, versehen mit faktor 0.25
- Neuer Buffer Wert aus dem AD-Wandler, versehen mit faktor 0.5
- Akkumulator wird an den DA-Wandler übergeben

Implementierung

- Akkumulator wird mit 11 Abgriffen vom Buffer geladen
- Koeffizienten für Abgriffe wurden mittels Exponentialfunktion berechnet

$$k = e^{-(3/4095)*i}$$

- k - koeffizient
- i - Index des Abgriffes



Implementierung

- Buffer als Ringspeicher mit laufendem Index
- Abgriff vom Buffer relativ zu neuestem Wert
- Ringspeicher mit 4096 Werten
 - inkrementierung des Index und AND mit 0x0fff

Dezimal: 4095 Binär: 0000 1111 1111 1111

- Koeffizienten für neuen Wert und Rückkopplung als 2er Potenz, daher reicht Schieben

Quellen

- Vorlesungsskript Signalprozessortechnik, SS2012
- Versuchsanleitung Nr. 2 „Digitale Filter“, Praktikum zur Veranstaltung „Digitale Signalverarbeitung“ Institut für Nachrichtentechnik, Universität Rostock
- <http://feedingthepuppy.typepad.com/.a/6a00e550f4976688340147e05708ca970b-800wi>
- <http://git-scm.com/downloads/logos>
- <http://gregrickaby.com/2012/03/how-to-use-github.html>

DEMO

GitHub

