

Projektdokumentation: Frequenz-Analyse und Hall

David Yaman [2321570], Dennis Räk [2385799]

HAW HAMBURG – IT-Systeme – SS 2021

Inhaltsverzeichnis

1	Konzept	3
2	Funktionen	3
2.1	Halleffekt	3
2.2	Low-/High-Cut	3
2.3	Frequenzspektrum (FFT)	4
3	Umsetzung	5
3.1	GUI	5
3.2	Programmcode	6
3.2.1	main.cpp	6
3.2.2	Control.h	6
3.2.3	Display.h	6
3.2.4	Encoding.h	7
3.2.5	Shield.h	7
4	Probleme	9
4.1	Audio-Board	9
4.2	Einbinden von Effekten	9
4.3	Verlust eines Teensy Boards	9
5	Fazit	10

1 Konzept

Das entwickelte Gerät ist ein Hall-Effektgerät mit zusätzlicher Monitoring-Möglichkeit über Betrachtung des Frequenzspektrums des Ausgangssignals. Informationen über die Funktionsweise und Parameter werden über ein LCD Display dargestellt und können mit Hilfe von vier Dreh-Encodern angepasst werden. Der Halleffekt hat die Parameter Room und Damping. Zusätzlich begrenzen Filtermodule das verhallte Signal im Frequenzbereich. Die variablen Grenzfrequenzen sind auch während Darstellung des Frequenzspektrums veränderbar, um neben auditiver auch visuelle Kontrolle zu ermöglichen. Der bandbegrenzte Hallanteil im Ausgangssignal lässt sich über Dry/Wet kontrollieren.

2 Funktionen

Die realisierten Funktionen umfassen einen Halleffekt, Low-/High-Cut und ein visualisiertes Frequenzspektrum des manipulierten Signals. Die Funktionen werden als einzelne Menüpunkte dargestellt und können mit Hilfe des Encoders Eins durchblättert werden. Encoder Zwei bis Vier steuern die Effekt- und Filterparameter, die per Knopfdruck auf ihren Ursprungswert zurücksetzbar sind.

2.1 Halleffekt

Der Halleffekt wird nach Start als Menüpunkt Eins angezeigt. Hier lassen sich die Parameter Room und Damping, mit Encoder Zwei und Vier anpassen. Zusätzlich kann der bandbegrenzte Hallanteil im Ausgangssignal mit Dry/Wet über Encoder Drei kontrolliert werden.

2.2 Low-/High-Cut

Low- und High-Cut sind im Routing hinter dem Halleffekt angesiedelt und werden im Menü als zweites dargestellt. Das verhallte Signal wird automatisch bandbegrenzt. Die Start-Grenzfrequenz des Low-Cuts beträgt 400 Hz, die des High-Cuts 12kHz. Mit Encoder Zwei und Vier können in diesem Menüpunkt die Grenzfrequenzen verschoben werden. Mit Encoder Drei lässt sich weiterhin der bandbegrenzte Hallanteil im Ausgangssignal kontrollieren.

2.3 Frequenzspektrum (FFT)

Menüpunkt Drei ist eine Darstellung des Ausgangssignals im Frequenzbereich. Der hörbare Frequenzbereich wird in 15 Balken aufgeteilt. Die Aufteilung orientiert sich an dem Hörverhalten des Menschen. Daher sind hohe Frequenzen breiter zusammengefasst als tiefe. Balken Eins bis Vier stellen Frequenzen von 40 bis 860 Hz dar, Fünf bis Zehn Frequenzen von 860 bis 8,6 kHz und Elf bis Fünfzehn Frequenzen von 8,6 kHz bis 22 kHz. Encoder Zwei bis Vier sind auf dieselbe Weise belegt wie in Menüpunkt Zwei.

3 Umsetzung

Die Umsetzung erfolgt mit Hilfe eines Teensy 4.1 Micro-Controllers. Angegeschlossen ist außerdem das speziell für den Teensy 4.1 entwickelte Audio-board SGTL5000. Für die Anzeige des Menüs wird ein 2.8 Zoll TFT Display (ILI9341) genutzt und es werden vier KY-040 Dreh-Encoder der Marke Joy-It mit dem Micro-Controller verbunden. Für den Aufbau wird eine Lochrasterplatine verwendet.

3.1 GUI

Das Userinterface ist in drei Menüpunkte aufgeteilt. Menüpunkt Eins gehört zum Halleffekt. Mittig wird Reverb angezeigt. Das untere Drittel des Displays ist in drei Kästen aufgeteilt die von links nach rechts Damping, Dry/Wet und Room und die jeweiligen Werte anzeigen. Damping und Room werden von 0 – 100, Dry/Wet von 0% - 100% angezeigt. Der Aufbau von Menüpunkt Zwei ist gleich. Die Parameter umfassen von links nach rechts Low-Cut, Dry/Wet und High-Cut. Low-Cut läuft zwischen 400 Hz und 12 kHz, High-Cut zwischen 12 kHz und 1,2 kHz, in einer Schrittweite von 100 Hz. Menüpunkt Drei zeigt die Frequenzanalyse an. Hier werden 15 Frequenzbänder mit ihrer momentanen Aussteuerung im Ausgangssignal angezeigt. Die Encoder sind gleich wie in Menüpunkt Zwei belegt. Die Ansteuerung erfolgt über die Encoder nach folgendem Schema:

Tabelle 1: Encoder Belegung

Encoder	Funktion Hall	Funktion Filter/FFT
1	Menü	Menü
2	Damping	Low-Cut
3	Dry/Wet	Dry/Wet
4	Room	High-Cut

3.2 Programmcode

Der Teensy 4.1 übernimmt mit seiner hohen Rechenleistung die Steuerung aller Funktionen. Das digitalgewandelte Audiosignal wird von ihm verhallt und bandbegrenzt. Währenddessen können Parameter und Grenzfrequenzen laufend verändert und über das Display ausgelesen werden. Auch die FFT wird vom Teensy durchgeführt und zur Visualisierung an das Display übergeben.

3.2.1 main.cpp

Die main ist das Herzstück des Programmcodes. Im Setup werden zuerst die Enocder initialisiert und anschließend das Display. Außerdem werden Audio-Board sowie zugehörige Effekte aktiviert. Durch die ‚encoder update‘ - Funktion wird innerhalb der Loop Funktion auf Änderungen im Status der Encoder gewartet.

3.2.2 Control.h

Mit der Bibliothek ‚Control.h‘ wird die allgemeine Ansteuerung initialisiert. Über Switch-Cases wird je nach Status des Encoders eine Menü-Variable und damit einer der drei Menüpunkte angewählt. Diese Information wird an ‚Display.h‘ übergeben und Parameterwerte der Effekte, sowie die Dry/Wet Zusammensetzung am virtuellen Audiomixer werden direkt angepasst.

Beispiel:

Encoder A beeinflusst die Menü-Auswahl und steht auf dem Case 0. Wird nun der Encoder C gedreht, dann beeinflusst dieser die Dämpfung des Hall-Effektes. Steht Enocder A auf dem Case 1, dann beeinflusst Encoder C die Grenzfrequenz des Low-Pass Filters.

3.2.3 Display.h

Dieser Programmabschnitt enthält die Display Klasse. In dieser werden Funktionen bereitgestellt, um die Änderung an Encoder oder FFT bildlich darzustellen. Wechselt der Nutzer zu einem anderen Menüpunkt oder verändert einen Parameter, dann wird der jeweilige Bildschirmbereich zurückgesetzt und durch den aktuellen ersetzt. Nach diesem Muster funktionieren Menüpunkte, Parameter sowie FFT. Beim Start des Programmes wird ein kleines Intro geladen. Wird Menüpunkt Drei aufgerufen, startet die FFT mit 1024 Sample-

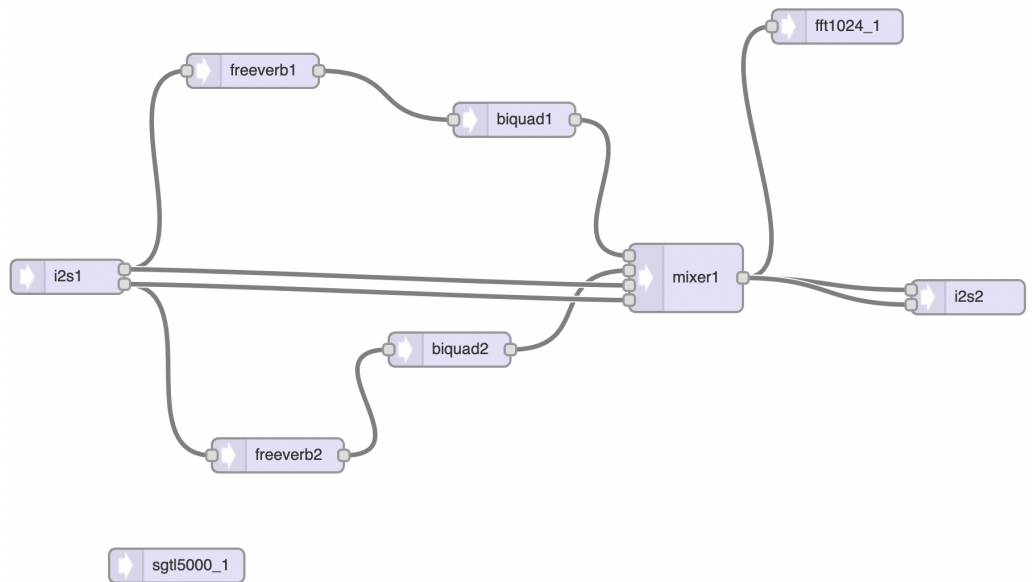
werten. Die FFT erzeugt 511 Frequenzbänder die den jeweiligen Frequenzanteil (0 – 1) enthalten. Die Werte werden normiert und begrenzt auf die 15 Balken aufgeteilt und vom Display angezeigt. Die Milliseconds-Funktion steuert den Aktualisierungsintervall. Die FFT, Halleffekt und die Filter wurden aus der Teensy Audio Library entnommen.^[1]

3.2.4 Encoding.h

Die Encoding Bibliothek ist einer der wichtigsten Teile des Codes, da eine Bedienung ohne funktionierende Encoder nicht möglich ist. Um akkurate Veränderung am Encoder wahrzunehmen, müssen diese Entprellt werden. Bei der Wahl einer geeigneten Entprellung zeigte das Quadrature Phase Shift Encoding von John Main gute Ergebnisse. Mit hoher Genauigkeit lassen sich kleine Veränderung am Encoder als +1 oder -1 wahrnehmen. Über die im Loop eingebettete encoder update Funktion wird eine Änderung am Encoder erkannt und in Form einer In- oder Decrementation an die Control Bibliothek weitergegeben. Die Knöpfe der Encoder werden durch die Bounce Library entprellt und ebenfalls über die Update Funktion überwacht.

3.2.5 Shield.h

Dieser Teil des Codes ist mithilfe des Teensy Audio System Design Tools generiert. Hier wird das Routing des Eingangssignals durch die verschiedenen Effekte festgelegt. Aus einem Screenshot des Audio System Design Tools wird das im Code erzeugte Routing deutlich:



4 Probleme

In diesem Abschnitt werden die Hindernisse und Probleme bei der Realisation des Projektes beschreiben.

4.1 Audio-Board

Anfänglich kam es zu Schwierigkeiten beim Anschluss des Audio-Boards. Über normale Jumper Wire ist eine störfreie Signalwiedergabe nicht möglich. Erst durch auflöten auf eine Lochrasterplatine ist eine Interferenzfreie Verbindung möglich und das Audio-Board funktioniert planmäßig.

4.2 Einbinden von Effekten

Im ursprünglichen Konzept sollte die eigenständige Programmierung von Audio Effekten erfolgen. Dieses Verfahren ist an mehreren Stellen gescheitert.

Viele Effekte lassen sich leicht in höheren Programmiersprachen wie Python oder MATLAB realisieren, aber für eine effiziente Programmierung in Echtzeit werden viele erweiterte C Kenntnisse benötigt. Es ist zwar leicht das Konzept nachzubauen, aber ohne stabiles DSP Framework ist eine gute Performance nur schlecht möglich.

Der Teensy erweist sich bei diesem Verfahren ebenfalls als eine Herausforderung. Bei Überwindung der ersten Herausforderung ist eine Einbindung in die bestehende Teensy Bibliothek notwendig. Dies ist sehr umständlich und ohne ein absolutes Verständnis nicht möglich. Leider gibt es von Seiten der Entwickler keine Anleitung eigene Effekte in die bestehende "Teensy Audio Library" einzubauen.

Alles in allem führten diese Hindernisse zu einer Abkehr vom Bau eigener Effekte.

4.3 Verlust eines Teensy Boards

Beim Anschluss der Drehencoder ist eine fehlerhafte Anbindung an die 5V Stromverbindung des Teensy entstanden. Da die Teensy PINs nicht 5V to-

lerant sind, führte dies zu einer Zerstörung eines Teensy Boards. Für dieses musste ein Ersatzgerät erworben werden.

5 Fazit

Trotz einiger Planänderungen ist das Projekt erfolgreich gewesen. Es ist ein funktionstüchtiger Halleffekt mit eingebauter Fast Fourier Transformation entstanden. Da einige Dinge wie das Einbinden von Effekten zum jetzigen Zeitpunkt noch nicht erreicht werden konnte, bietet das Projekt auch nach der IT-Systeme Veranstaltung Möglichkeiten der Weiterarbeit.