

Projektdokumentation: Frequenz-Analyse und Hall

David Yaman, Dennis Räk [2385799]

HAW HAMBURG – IT-Systeme – SS 2021

Inhaltsverzeichnis

1	Konzept	3
2	Funktionen	3
3	Umsetzung	3
3.1	Hardware	3
4	Umsetzung im Programmcode	4
4.1	main.cpp	4
4.2	Control.h	4
4.3	Display.h	4
4.4	Encoding.h	4
4.5	Shield.h	5
5	Probleme	6
5.1	Audio-Board	6
5.2	Einbinden von Effekten	6
5.3	Verlust eines Teensy Boards	6
6	Fazit	7

1 Konzept

Das Projekt ist ein Hallgerät mit eingebauter Frequenzanalyse. Für die Signalverarbeitung wird ein Teensy 4.1 Microcontroller mit Audio-Board Adapter verwendet. Dieses System lässt sich über 4 Drehencoder einstellen. Für die Darstellung eines GUI wird ein LCD Monitor benutzt. Das fertige Gerät ist auf einer Lochrasterplatine befestigt und in ein Gehäuse integriert. Ein- und Ausgang des Audio-Signals erfolgt über am Audio-Board befestigte Cinch-Buxen. Bei dem Hall ist es möglich Dämpfung, Raumgröße, Dry/Wet sowie Low-Cut und High-Cut einzustellen. In einem separaten Menü-Punkt lässt sich das Ausgangssignal per Frequenzanalyse beobachten.

2 Funktionen

Beim Start des Programmes erscheint ein kleines Intro

3 Umsetzung

3.1 Hardware

Übersicht der im Projekt verwendeten Hardware:

- 1x Teensy 4.1 Development Board
- 1x Teensy Audio Adaptor Rev D
- 1x 2.8 Zoll TFT Display (ILI9341)
- 4x KY-040 Drehencoder
- 1x Lochrasterplatine

Übersicht: - Audio Adaptor

4 Umsetzung im Programmcode

4.1 main.cpp

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

4.2 Control.h

Mit der Bibliothek "Control.h" wird die allgemeine Ansteuerung initialisiert. Über Switch-Cases wird je nach Status des Encoders eine Menu-Variable ausgewählt.

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 Encoder A auf dem Case 1, dann beeinflusst Encoder C die Grenzfrequenz des Low-Pass Filters.

4.3 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 gelöscht und durch den aktuellen ersetzt. Nach diesem Muster funktionieren Menüpunkte, Parameter sowie FFT. Beim Start des Programmes wird ein kleines Intro geladen.

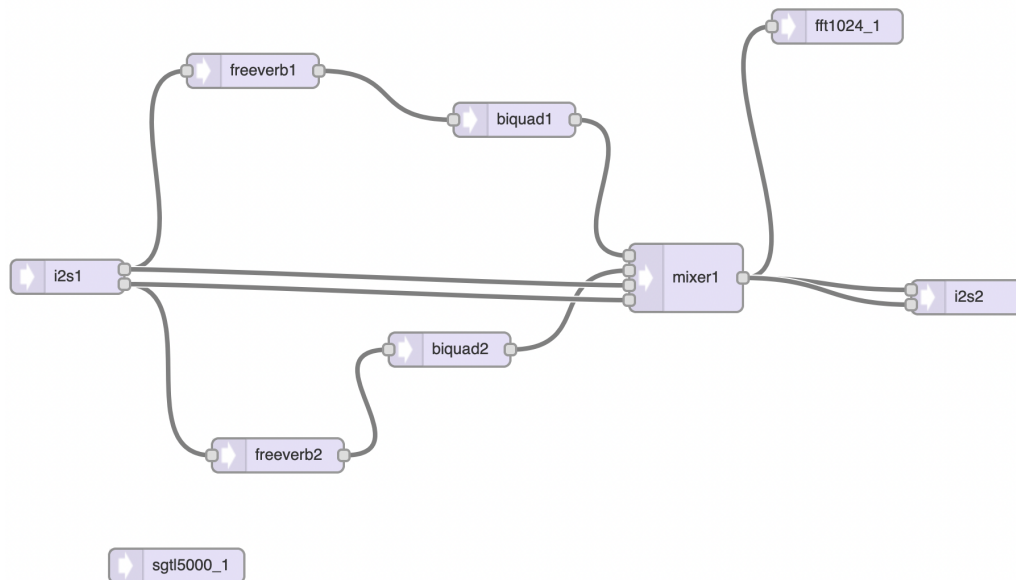
4.4 Encoding.h

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.

4.5 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:



5 Probleme

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

5.1 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.

5.2 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.

5.3 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.

6 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.