

digitales Theremin

Projektklärung

FS20, Projekt 6

Windisch, 06.03.2020

Auftraggeber:	Herr Prof. Dr. Hanspeter Schmid
Betreuung:	Herr Prof. Dr. Hanspeter Schmid Herr Prof. Karl Schenk
Experte:	Herr Dr. Jürg Stettbacher
Team:	Andreas Frei Dennis Aeschbacher
Studiengang:	Elektro- und Informationstechnik (EIT)

Inhaltsverzeichnis

1	Übersicht.....	3
1.1	Ausgangslage.....	3
1.2	Ziele.....	4
1.2.1	Sollziele.....	4
1.2.2	Wunschziele	4
2	Konzept.....	5
2.1	Blockschaltbild	7
3	Risiken	8
4	Projektstrukturplan	9
5	Quellenverzeichnis	9

1 Übersicht

1.1 Ausgangslage

Ein Theremin ist ein Musikinstrument, das mit "den Händen in der Luft" gespielt wird. Es kommt in vielen Musikstücken, aber auch in Filmen (z.B. "Inspector Barnaby"-Serie, Hellboy) zum Einsatz. Beim Theremin beeinflusst die elektrische Kapazität des menschlichen Körpers ein elektromagnetisches Feld. Dabei steuert die Position der Hände gegenüber zwei Antennen die Stärke der Veränderung von Tonhöhe und Lautstärke. Die sich ändernde Schwingung des Feldes wird verstärkt und als Ton über einen Lautsprecher ausgegeben [1].

Ziel des Projekts ist, basierend auf einem Bausatz eines analogen Theremins (z.B. Franzis) ein digitales Theremin zu bauen. Dabei soll die gesamte Signalverarbeitung in einem FPGA realisiert werden. Gewünschte Features beinhalten: Spielhilfen zur Bedienung des Instruments, Klangeffekte zur Annäherung anderer Instrumente in elektronischer Form und das Aufnehmen und Wiedergeben von kurzen Tonsequenzen. Das digitale Theremin soll in ein kompaktes ansprechendes Gehäuse gepackt werden.

Um diese Ziele zu erreichen wird die Arbeit auf das fünfte und sechste Semester verteilt.

Im fünften Semester wurde eine digitale Plattform für die Ansteuerung und Kalibrierung der Tonhöhenantenne entwickelt. Dazu wurde das DE1-SoC Entwicklungsboard mit einem Cyclone V verwendet. Im sechsten Semester wird zusätzlich die Lautstärkeantenne sowie die oben erwähnten Features implementiert. Um die Bedienung des Theremins möglichst benutzerfreundlich zu gestalten, wird dies über ein Touchscreen realisiert. Um die Verschiedenen Komponenten anzusteuern wird der Softcore NiosII implementiert.

1.2 Ziele

Die Ziele wurden nach ihrer Priorität geordnet und dann in Soll- und Wunschziele unterteilt.

1.2.1 Sollziele

Die Sollziele sind Ziele, welche erfüllt werden müssen.

Priorität	Ziel	Inhalt
1	Lautstärke Antenne	Zweite Antenne implementieren, um gleichzeitig die Lautstärke einstellen zu können.
2	Redesign des PCB	Neues PCB Design mit Lautstärkeantenne, Speisung und Audioverstärker erstellen.
3	Nios II und Menu auf Touchscreen	Das Theremin soll über einen Touchscreen bedienbar sein. Dies wird über den Nios II auf dem FPGA gesteuert.
4	Spielhilfe: Genaue Tongenerierung mit Glissando Effekt	Hilfe beim Spielen des Theremins durch annähern an diskrete Töne mit Glissando Effekt (keine Sprünge von Ton zu Ton sondern fließende Übergänge).
5	Automatisches Tuning der Tonhöhe und Lautstärke	Der digitale Referenzoszillator wird automatisch auf die Frequenz der externen Oszillatoren abgestimmt.
6	Spielhilfe: Anzeige der Spielgenauigkeit	Graphische Darstellung der Spielgenauigkeit auf dem Touchscreen.
7	ansprechendes Gehäuse	Ein ansprechendes Gehäuse bauen, bei welchem die verbaute Elektronik sichtbar ist.
8	Klangeffekte hinzufügen	Klang verändern, um weitere Instrumente in elektronischer Form anzunähern (Bsp. Violine) durch Manipulierung der Grund- und Oberwellen.

1.2.2 Wunschziele

Die Wunschziele können realisiert werden, wenn noch genügend Zeit vorhanden ist.

Priorität	Ziel	Inhalt
9	Weitere Instrumente annähern	Zusätzliche Instrumente annähern für weitere Klangmodi.
10	Sampling	Es soll möglich sein, während dem Spielen kurze Abschnitte abzuspeichern, um sie wiederholt abspielen zu können. Wie man das sonst mit Loopern macht.

2 Konzept

Die unten beschriebenen Komponenten können in Kapitel 2.1 Blockschaltbild gefunden werden.

Tonhöhenoszillator

In dem Block Tonhöhe kann die Frequenz des Ausgangssignals verändert werden, indem die Distanz zwischen dem Spieler und der angeschlossenen Antenne verändert wird und so der Tonhöhenoszillator verstimmt wird. Dies, da der Spieler des Instruments das elektrische Feld der Antenne so verändert, dass die Frequenz des Oszillators leicht ändert. Dieser Teil wird nicht im FPGA sondern analog realisiert. Der Schwingkreis wird mit dem Colpitts-Oszillator verwirklicht.

Referenzoszillator (Tonhöhe)

Der Referenzoszillator wird benötigt, um das Signal des Tonhöhenoszillators im Mischer zu bearbeiten. In Projekt 5 wurde für diesen Referenzoszillator der Cordic Algorithmus verwendet. Die Frequenz des Referenzoszillator kann über den Spielhilfe Modus und die Autokalibrierung verstellt werden.

Mischer (Tonhöhe)

Die Komponente Mischer ist dafür zuständig die Frequenz des Tonhöhenoszillators, welche nicht im hörbaren Bereich ist in den hörbaren Bereich zu bringen. Dabei multipliziert er die Signale des Tonhöhenoszillators mit der des Referenzoszillators. Der Mischer wurde bereits in P5 eingesetzt und kann für P6 übernommen werden.

Tiefpassfilter (Tonhöhe)

Da der zuvor erwähnte Mischer auch hochfrequente Signale generiert, muss das gemischte Signal noch mit einem Tiefpassfilter gefiltert werden. Dazu wird ein CIC Filter verwendet. Dieses wurde bereits in P5 verwendet und kann im P6 übernommen werden.

Lautstärkeoszillator, Referenzoszillator (Lautstärke), Mischer (Lautstärke), Tiefpassfilter (Lautstärke)

Im Block Lautstärke kann die Lautstärke des Ausgangssignals verändert werden, indem die Distanz zwischen dem Spieler und der angeschlossenen Antenne verändert werden. Der Lautstärkeoszillator funktioniert nach demselben Prinzip wie der Tonhöhenoszillator. Wie man im Blockschaltbild sehen kann verwandelt die Lautstärke Regelung dieselben Komponenten wie der Block Tonhöhe. Mit dem einzigen Unterschied, dass der Referenzoszillator nur von der Autokalibrierung verstellt werden kann.

Frequenzmessung

Die Frequenzmessung wird für verschiedenste Funktionalitäten implementiert. Zum einen wird sie benötigt, um die Kalibrierung der Tonhöhe durchzuführen. Zum anderen um die beiden Spielhilfemodi, welche in den Zielen beschrieben sind, zu ermöglichen. Für die beiden Spielmodi muss besonders auf die Genauigkeit der Frequenzmessung geachtet werden.

Klangeffekte

Um den Klang des Theremins abwechslungsreicher zu gestalten sollen Klangeffekte implementiert werden, um andere Instrumente in elektronischer Form anzunähern. Dazu können die Grund- und Oberwellen manipuliert werden. Mit welcher Methode dies gemacht werden kann muss noch recherchiert werden.

Lautstärkeeinstellung

Um die Lautstärke des Theremins zu steuern wird das Signal in Abhängigkeit der Frequenz des Lautstärkeblocks verstärkt. Dazu muss auch hier eine Frequenzmessung durchgeführt werden.

Sampling & Loop

Sampling & Loop soll es ermöglichen kurze Tonsequenzen aufzuzeichnen und wiederholt auszugeben, wie es sonst mit einem sogenannten Looper gemacht wird. Dieses Feature ist jedoch ein Wunschziel und wird nur bei genügender Zeit implementiert.

DAC

Wie der Name sagt wandelt der Digital-Analog Converter das digitale Signal in ein analoges um.

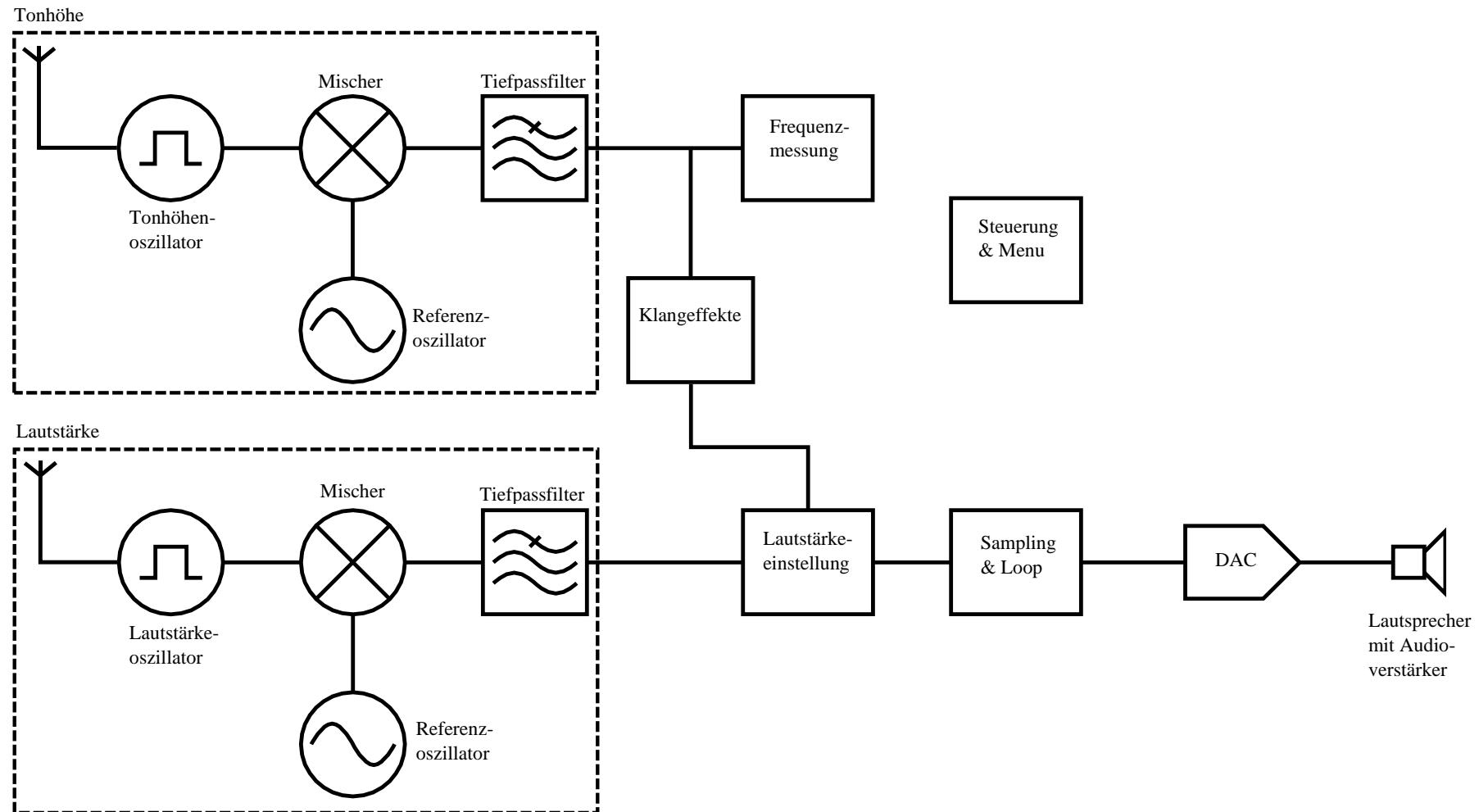
Lautsprecher mit Audioverstärker

Zum einen soll ein Lautsprecher fest in das Gehäuse des Theremins verbaut werden und zum anderen soll es auch möglich sein eine externe Audioquelle wie Kopfhörer an einen 3.5mm Klinkenstecker anzuschliessen. Für den eingebauten Lautsprecher wird auf dem bereits erwähnten PCB ein Audioverstärker implementiert.

Steuerung & Menu

Um das Theremin zu steuern soll ein Nios II System auf dem FPGA implementiert werden. Dieses verwaltet die verschiedenen Spielmodi und das Kalibrieren des Theremins. Weiter soll das Theremin über einen Touchscreen bedient werden können. Hier soll die Spielgenauigkeit grafisch ersichtlich sein.

2.1 Blockschaltbild



3 Risiken

Gewicht	Beschreib	Konsequenzen	Prävention
1	Die DC Spannung ist durch das Schaltnetzteil und das FPGA zu verrauscht.	Dies hat zur Folge das die Antennenoszillatoren nicht funktionieren. Das Theremin kann nicht gespielt werden.	Die Rippelspannung des vorhandenen Schaltnetzteil passend filtern.
2	Der Audioverstärker ist ungenügend	Die Audioqualität ist nicht genügend.	Bereits bestehende Theremin Bausätze konsultieren.
3	Die Implementierung des Nios II gelingt nicht.	Die Bedienung des Theremin ist eingeschränkt. Die Frequenzmessung für den Modus Spielhilfe kann nicht durchgeführt werden.	Bestehende Nios II Projekte konsultieren.
4	Die Frequenzmessung für den Spielhilfe Modus ist zu ungenau	Der Spielhilfe Modus funktioniert nicht richtig. Dies da das menschliche Ohr sehr empfindlich ist auf kleine Tonschwankungen.	Methoden für die Frequenzmessung recherchieren, welche eine hohe Genauigkeit ermöglichen.

4 Projektstrukturplan

Der Projektstrukturplan wurde auf Grund der Übersichtlichkeit im Excel File Projektstrukturplan.xlsx realisiert.

5 Quellenverzeichnis

- [1] T.Riegler, “Wie funktioniert ein Theremin? ,“ Theremin zum selberbauen. München, Franzis Verlag, 2018, 12. [Online]
Available:https://files.elv.com/Assets/Produkte/14/1448/144855/Downloads/65347_Theremin-selber-bauen_Druckdaten.pdf [09.10.2019]