

Praxisbericht zum Praxissemester

Melodieerkennung mit einem Mikrocontroller

Eric Gattinger, geb. 01.07.98

Matrikelnr. 866950

Abgabetermin: 01.03.2020

Technische Hochschule Rosenheim, Bachelorstudiengang Informatik

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich den vorliegenden Praxisbericht selbstständig und ohne Benutzung anderer als der angegebenen Quellen verfasst habe. Wörtlich übernommene Sätze oder Satzteile sind als Zitat belegt, andere Anlehnungen, hinsichtlich Aussage und Umfang, unter Quellenangabe kenntlich gemacht. Dieser Bericht hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen und ist nicht veröffentlicht. Sie wurde nicht, auch nicht auszugsweise, für eine andere Prüfungs- oder Studienleistung verwendet.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Ehrenwörtliche Erklärung.....	2
Abstract.....	4
Glossar.....	5
1 Einleitung.....	7
2 Hauptteil.....	7
2.1 Entwurf.....	7
2.2 Implementierung.....	8
2.2.1 Idee: Periodendauer zwischen zwei Nullpunkt durchgängen messen.....	10
2.2.2 Idee: Periodendauer zwischen zwei Peaks messen.....	11
2.2.3 Idee: 1. Harmonische durch Fouriertransformation berechnen.....	12
2.2.4 Idee: Bitweise Autokorrelation.....	12
2.2.5 Rhythmisik.....	15
2.3 Fertigung.....	15
2.4 Anwendung.....	17
3 Schlussfolgerung, Reflexion und Ausblicke.....	18
Literaturverzeichnis.....	19
Bestätigung der Inhaltlichen Richtigkeit.....	20

Abstract

Im folgenden Praxisbericht wird im Wesentlichen das Projekt vorgestellt, mit dem ich mich die meiste Zeit im Praktikum beschäftigte. Das Projektziel war ein Gerät zusammenzustellen, das eingesungene Melodien erkennen und auf bestimmte Melodien eine Reaktion auslösen würde.

Glossar

ADC (Analog-Digital-Converter): ein Modul, das die Höhe der Eingangsspannung in einen digitalen Wert umwandelt [1]

Amplitude: Größter Ausschlag einer Schwingung [2]

Arbeitsspeicher: der Ort, an dem die Variablen eines Programms hinterlegt sind

Arduino: eine programmierbare Hardware, bestehend aus einem einfachen I/O-Board und einem Mikrocontroller, das bekannt für seine Einsteigerfreundlichkeit ist [3]

Autokorrelation: beschreibt die Beziehung eines Signals mit sich selbst zu einem früheren Zeitpunkt [4]

Bestückung: der Prozess, in dem eine leere Leiterplatte ihre Bauteile aufgesetzt bekommt

CCS (Code Composer Studio): die Entwicklungsumgebung für den MSP [5]

Flash-Speicher: der Ort, an dem das Programm hinterlegt ist

Frequenz: die Anzahl der Zyklen einer Sinuskurve [6]

Leiterplatte: ein Träger für elektronische Bauteile, welche zur mechanischen Befestigung und elektrischen Verbindung dient

Lötpaste: dient zum Löten oberflächenmontierbarer Bauelemente

Mikrocontroller: ein Halbleiterchip, der einen Prozessor und zugleich auch verschiedene Module beinhaltet [7]

Modul: elektronisches Gerät, Bauteil oder Teil eines Bauteils mit einer bestimmten Funktion

MSP (Mixed-Signal Processor): ein Mikrocontroller der Firma Texas Instruments [8]

Oszilloskop: ein elektronisches Messgerät, das elektrische Spannungen im zeitlichen Verlauf am Bildschirm zeigt

Peak: Größter Ausschlag einer Periode

Periode: ein einziger Zyklus eines Signals

Periodendauer: die Dauer, bis sich das Signal selbst wiederholt

Referenzspannung: eine Spannung die man dem ADC als maximale Spannung vorgibt

Sample: eine einzelne gemessene und digitalisierte Spannung

Timer: ein Modul im MSP, das nach einer definierten Zeit einen bestimmten Vorgang auslöst
[9]

Toleranzwert: der Vergleichswert für einen noch zulässigen Wert

zeitdiskret: zu bestimmten (meist immer gleich weit entfernten) Zeitpunkten definiert [10]

1 Einleitung

Mein Praktikumsbetrieb war die HEMI GmbH in Raubling, eine Firma die kundenspezifische Elektronik entwickelt und fertigt. Ihre Kernkompetenz liegt in der Realisierung analoger, digitaler und gemischter Schaltungen sowie in kompakten Mikrocontroller-Systemen.

Die Firma ist in zwei Abteilungen aufgeteilt, der Fertigung und der Entwicklung. In der Entwicklung wird für ein Projekt unter Anderem Ideen gesammelt, die grobe Schaltung konzipiert und die Software geschrieben sowie debuggt. Die Fertigung hat die Aufgabe, die Schaltung zu konstruieren, also die Leiterplatte zu entwerfen und zu bestücken.

Mein Arbeitsplatz befand sich in der Entwicklung. In den ersten zwei Monaten lernte ich dort die Funktionen von Mikrocontrollern kennen, wie man auf dessen Register zugreift und diese ändert, erweiterte mein Wissen über elektrische Schaltungen und lernte, wie man im Team ein Projekt entwickelt.

Später hatte ich die Chance an einem eigenen Projekt zu arbeiten und konnte somit den Ablauf eines Projektes sowohl in der Entwicklung, als auch in der Fertigung, von Anfang bis Ende erfahren.

2 Hauptteil

2.1 Entwurf

Zusammen mit einem weiteren Praktikanten, der Mechatronik an der TH Rosenheim studiert, durften wir uns ein eigenes Projekt überlegen. Wir setzten uns als Ziel, ein Gerät zu entwickeln, welches Audiosignale aufnimmt und zu einer bestimmten, selbst eingesungenen Melodie, eine gewünschte Aktion ausführt (z. B. ein Türschloss öffnet).

Dazu würden wir einen MSP430F5528 (MSP = mixed-signal processor) Mikrocontroller von Texas Instruments verwenden, welcher unter Anderem einen Analog-Digital-Converter (ADC), um das Audiosignal zu messen, Kommunikationsschnittstellen, um die Ergebnisse auf ein externes Anzeigegerät zu übertragen, mehrere Timer-Module, um Funktionsaufrufe zeitlich zu definieren, und genügend Flash- und Arbeitsspeicher für den Programmcode bzw. die Variablen zur Verfügung hat. Die MSP-Mikrocontroller sind für Anwendungen mit niedrigstem Stromverbrauch optimiert. [8][11]

Texas Instruments bietet eine eigene Entwicklungsumgebung an, namens Code Composer Studio (CCS), die auf Eclipse basiert. Als Programmiersprache wird C und/oder Assembler verwendet. Der Code für das Projekt wurde ausschließlich mit C geschrieben, während Assembler manchmal zum Debuggen hilfreich war.

Das Signal würde von einem Mikrofon und der eigenen Stimme produziert werden. Zudem gehört eine Verstärkerschaltung, um das Signal zu verstärken und auf den korrekten Arbeitspunkt zu setzen.

Wir hatten mehrere Ideen der Signalverarbeitung, die während der Implementierung entstanden und verworfen wurden. Ziel war es, ein funktionierendes Verfahren zu finden, das nicht zu viel Speicher benötigt und gleichzeitig schnell ausgeführt wird, um eine flüssige Bedienung zu ermöglichen.

Auf einem kleinen LCD-Display, sollen die Werte der Signalverarbeitung dargestellt werden. Da das Debugger-Tool im CCS die aktuellen Register- und Variablenwerte nur anzeigt, wenn das Programm pausiert wurde, dient das Display auch zum schnelleren und einfacheren Debugging, da die Werte während der Laufzeit vom Display abgelesen werden können.

2.2 Implementierung

Kurz vor unserer Projektidee, versuchten wir mit einem Piezo-Lautsprecher die Lautstärke eines akustischen Signals zu bestimmen. Ein Lautsprecher wird normalerweise als Ausgang von Tönen verwendet, allerdings brachte uns dieser zur Erkenntnis, dass jeder Lautsprecher auch ein mehr oder weniger gutes Mikrofon sein kann und umgedreht ebenso. Lautsprecher, sowie viele Mikrofone, haben einen Minus- und Pluspol. Werden diese in einer bestimmten Frequenz immer wieder bestromt, so wird ein akustisches Signal wiedergegeben. Benutzt man einen Piezo-Lautsprecher jedoch als Spannungsquelle, können Töne in Spannung umgewandelt und somit ein analoges Signal an einem verbundenen MSP geleitet werden.

Wenn man die Lautstärke mithilfe des ADC im MSP messen möchte, betrachtet man den größten Ausschlag des Signals, also die Amplitude. Verbildlicht man das Signal eines Klopfens an einem Oszilloskop, so kann man die Lautstärke am höchsten positiven oder negativen Peak auslesen. Je höher die Lautstärke, desto größer die gemessene Spannung.

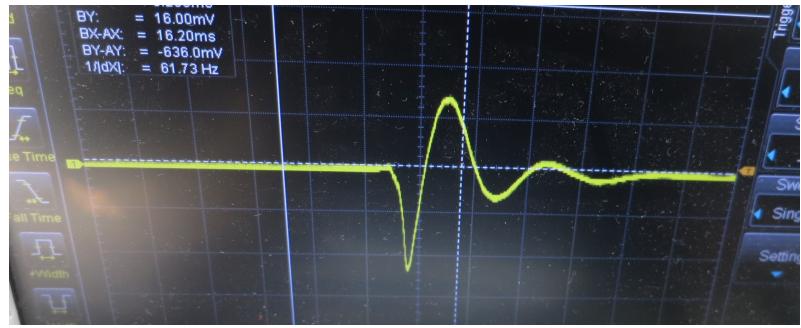


Abbildung 1: Signal eines Klopfens am Oszilloskop.

Der ADC generiert andauernd im Abstand von ein paar Millisekunden ein sogenanntes Sample. Samples entsprechen dem Wert der gemessenen Spannung gegenüber einer Referenzspannung und wandelt diese in einen digitalen Wert um. Je näher die gemessene Spannung an der Referenzspannung liegt, desto höher fällt der Wert aus. Liegt sie über oder auf der Referenzspannung, so entspricht der Wert dem maximal darstellbaren Digitalwert. Man nennt diesen Prozess auch Abtastung. [12]

Im Projekt möchten wir nun zusätzlich die Frequenz eines Tons bestimmen. Der Ton sollte eine kurze Zeit gehalten werden, um ein sinnvolles Ergebnis zu bekommen. Man sehe sich zuerst das Eingangssignal eines eingesungenen Vokals in Abbildung 2 an.

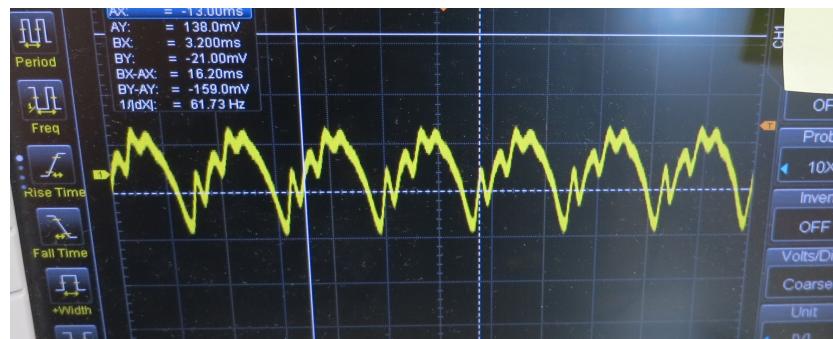


Abbildung 2: Signal eines eingesungenen „A“-Vokals.

Man erkennt zwei wichtige Fakten: Das Signal wiederholt sich periodisch und die Peaks (= höchster oder tiefster Ausschlag einer Periode) liegen immer an der selben Stelle, wenn man die Perioden miteinander vergleicht. Um die Frequenz zu erhalten, muss man die Anzahl der Perioden in einer Sekunde abzählen. Ziel ist es also, die Periodendauer herauszufinden. Dazu kamen folgende Ideen zustande, die in den folgenden Kapiteln einzeln erklärt werden:

- Die Zeit zwischen zwei Nullpunkten messen
- Die Zeit zwischen zwei Peaks messen
- Fourier-Transformation
- Autokorrelation

Bei jedem dieser Ideen, nimmt der MSP vor jeder Signalverarbeitung mehrere Samples auf, bis man eine maximale Anzahl von Werten hat. Das Signal ist somit diskretisiert abgespeichert, wie es das Oszilloskop ebenso macht. In anderen Worten: man hat ein abgetastetes Signal.

2.2.1 Idee: Periodendauer zwischen zwei Nullpunktdurchgängen messen

Vorteile: Schnell, genau und nachvollziehbar für Menschen.

Nachteil: Nur für perfekte Sinussignale geeignet. Somit ungeeignet für das Projekt.

Misst man zwei Nullpunktdurchgänge einer steigenden Flanke, bekommt man die Periodendauer zurück. Der Nullpunktdurchgang der sinkenden Flanke wird dabei ignoriert. Das Problem hierbei ist jedoch, dass es nur für Signale mit keinem weiteren Nullpunktdurchgang in einer Periode funktioniert, also funktioniert diese Methode nur bei Signalen, die wie ein Sinussignal aussehen. Signale bestehen aber oft aus mehreren interferierenden Sinusoiden und erfüllen diese Bedingung nicht, wie in Abbildung 3 ersichtlich.

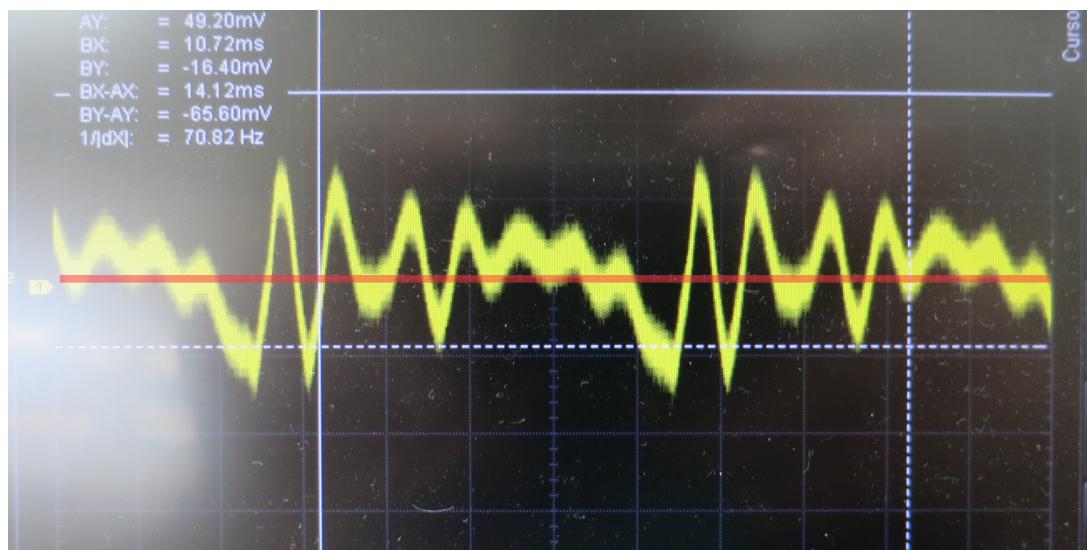


Abbildung 3: Signal eines anderen eingesungenen Tons. Die rote Linie kennzeichnet den Nullpunkt. Interessanter Nebenfakt: das Integral eines sinusartigen Signals ist immer 0.

2.2.2 Idee: Periodendauer zwischen zwei Peaks messen

Vorteile: Schnell und nachvollziehbar für Menschen.

Nachteile: Ziemlich ungenau und speicherintensiv wegen den vielen abgespeicherten Samples.

Sehr ähnlich zu der Idee mit den Nullpunktdurchgängen, messen wir hier die zeitlichen Abstände zweier nebeneinanderliegender Peaks, oder anders gesagt die Abstände der Minima oder Maxima aus zwei Perioden.

In jeder Periode eines Signals ist ein Peak vorhanden, der immer an der selben Stelle jeder Periode liegt, solange man diesen Ton hält. Die Software wird nun die minimalen und maximalen Amplitudenwerte aus einem Teil des abgetasteten Signals herausfinden, anhand derer sie im restlichen Teil des Signals die nächsten zwei Peaks findet und die Zeit dazwischen misst.

Da der ADC zeitdiskret misst, ist es sehr unwahrscheinlich, dass jemals der maximale oder minimale Ausschlag erfasst wird. Um Peaks zu erkennen, die den gemessenen Peak nur beinahe erreichen, muss man also eine ausreichende Toleranz zum Wert des gemessenen Peaks berücksichtigen. Die optimale Toleranz zu bestimmen ist sehr wichtig, da mit einer zu niedrigen Toleranz, der Vergleichswert zu hoch sein kann und Peaks somit übersehen werden können. Mit einer zu hohen Toleranz, werden wiederum Peaks gefunden, die gar keine sind. Viele Testversuche mit verschiedenen Toleranzwerten müssen also gemacht werden, um die optimale Toleranz zu finden.

Es kommt häufig vor, wie z. B. auch auf Abbildung 3, dass ein Signal in einer Periode zwei fast gleich hohe „Maxima“ und/oder „Minima“ hat. Theoretisch muss eines davon immer höher sein als der Rest. Durch die vorher bestimmte Toleranz und der Diskretisierung, erkennt das MSP jedoch beide Maxima oder Minima als Peak der Periode. In unserem Projekt entschieden wir uns deshalb beides, sowohl die Maxima als auch die Minima, zu messen, da eine der beiden Seiten eine höhere und somit die richtige Periodendauer ergibt. Solange es gilt, dass eines von beiden maximal ein Maximum oder ein Minimum in einer Periode hat, was bei den meisten Signalen der Fall ist, kann man so problemlos die Periodendauer berechnen. Bei Signalen mit mehreren ähnlich hohen Maxima und Minima innerhalb einer Periode, wie das auf Abbildung 3, besteht das Problem weiterhin.

Das letzte Problem löst sich folgendermaßen: Falls die nächste gemessene Zeit auf einmal deutlich kürzer, z. B. halb so kurz ausfällt wie die Vorherige, ergibt sich durch zusammenaddieren beider Zeiten entweder die tatsächliche Periodendauer, oder man wiederholt dieses Schema bis dahin. Diese Lösung ist nicht perfekt, aber ausreichend gut für unser Projekt, da es in sehr vielen Fällen die korrekte Periodendauer liefert.

2.2.3 Idee: 1. Harmonische durch Fouriertransformation berechnen

Vorteil: Präzises Ergebnis.

Nachteile: Langsam und hundertfach speicherintensiver.

Um die Frequenz des Eingangssignals zu berechnen, reicht es, die Frequenz der sogenannten Grundschwingung, bzw. ersten harmonischen Schwingung, zu kennen. Diese ist ein perfekter Sinus und somit leicht auslesbar. Mit der Fouriertransformation kann man die 1. Harmonische zwar berechnen, aber auf einem Prozessor würde die Berechnung zu lange dauern. Abhilfe schafft der Fast Fourier Transform Algorithmus (kurz: FFT), mit einer Komplexität von $O(n \cdot \log(n))$. Dieser Algorithmus behilft sich mit komplexen Zahlen, die Grund- und Oberschwingungen eines Signals schnell herauszufinden. [13][14]

Auch wenn wir nur die Grundschwingung brauchen, ist dieser Algorithmus sehr speicherintensiv, da eine komplexe Zahl aus 2 Gleitkommawerten besteht und man somit pro Sample mindestens 8 Bytes verbraucht.

Einen MSP mit genügend Arbeitsspeicher für diese Variablen wäre teurer als was in unserer Vorstellung lag. Um den Arbeitsspeicher nicht so stark zu belasten, gäbe es einen MSP mit einem integrierten FFT-Modul oder das FFT-Modul als Bauteil, was aber ebenso als überteuert galt. Somit wurde diese Idee verworfen.

2.2.4 Idee: Bitweise Autokorrelation

Vorteile: Sehr Speicherarm und sehr nachvollziehbar für Menschen, da der Mensch genauso vorgeht.

Nachteil: Etwas langsamer im Vergleich zu vorherige Ideen.

Bei der Autokorrelation vergleicht sich das Signal mit sich selbst zu einem früheren Zeitpunkt. Diese Idee ist in der aktuellen Software implementiert. [4]

Nachdem das Signal abgetastet wurde, unterteilt man es in zwei Hälften. Die Hälften werden stückchenweise (genauer gesagt: sampleweise) miteinander verglichen, bis sich beide Teile überdecken oder zumindest einer sehr hohen Ähnlichkeit – Toleranz muss erneut gut gewählt werden – entsprechen. Theoretisch überdecken sich die Hälften, wenn man eine Hälfte um genau eine (oder mehrere) Perioden verschiebt. Indem man eine Hälfte sampleweise verschiebt und die Zeit zwischen zwei Samples kennt, erhält man in der Praxis, sobald sich beide Hälften zum ersten mal überdecken, die Periodendauer anhand der Anzahl der Verschübe.

Die Signale können für dieses Verfahren stark quantisiert werden. Das macht es sehr tauglich für Mikrocontroller, die nur wenig Arbeitsspeicher zur Verfügung stellen können. Wir entschieden uns für eine 2-Bit Quantisierung, da diese zumindest auch die Amplituden noch in Betracht zieht. Die ADC-Werte, die im verwendeten MSP mit einer Genauigkeit von 12 Bits dargestellt werden, werden also

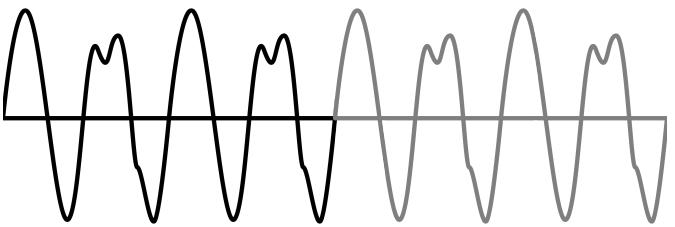


Abbildung 4.1: Das Signal wird in zwei Hälften

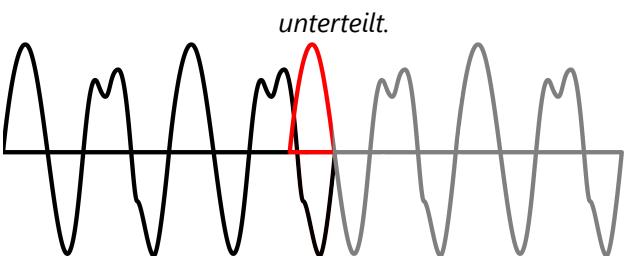


Abbildung 4.2: Die neuere Hälfte wird um einen Schritt nach links verschoben.

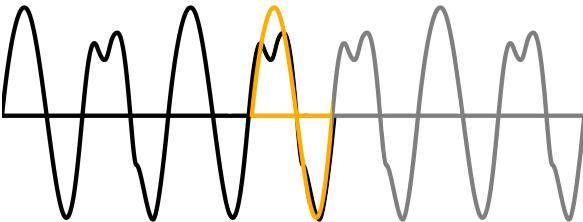


Abbildung 4.3: Ein weiterer Schritt nach links. Wenn eine zu hohe Toleranz für die Ähnlichkeit gewählt wird, kann man hier eine falsche Periodendauer erhalten.

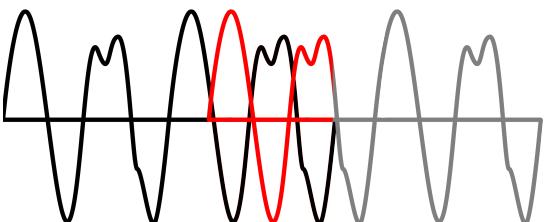


Abbildung 4.4: Ein weiterer Schritt nach links.

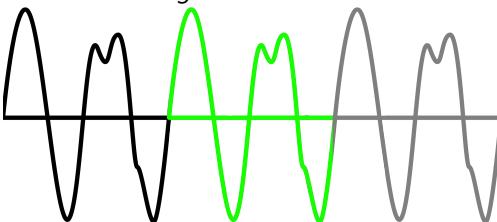


Abbildung 4.5: Nach einem weiteren Schritt überdecken sich die Teile der Hälften, da nun insgesamt ein Verschub von einer Periode durchgeführt wurde. Anhand der Anzahl der Schritte kennt man nun die Periodendauer.

von 4096 auf 4 Werte heruntergebrochen. Die Werte 0 bis 1023 entspricht dann 0, 1024 bis 2047 entspricht 1 usw.

Anders ausgedrückt, werden aus dem 12-Bit Wert die 10 niedrigstwertigen Bits ignoriert und nur die 2 höchstwertigen Bits abgespeichert.

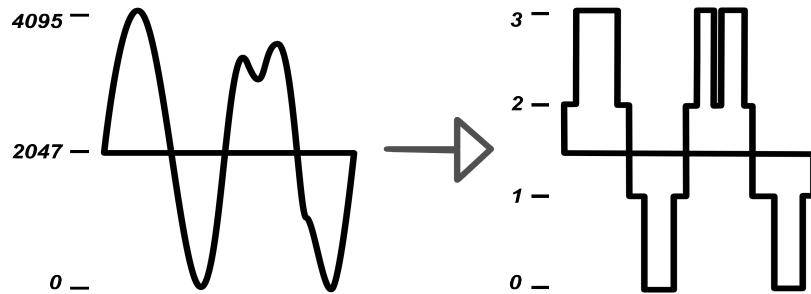


Abbildung 5: 2-Bit Quantisierung eines Signals.

Eine binäre Quantisierung wäre zwar halb so speicherintensiv, verliert aber jegliche Informationen über die Amplituden. Man weiß nur noch, ob der abgetastete Wert positiv oder negativ war. Bei manchen Signalen folgt daraus ein Problem, in dem das quantisierte Signal eine höhere Frequenz wie das Original aufweist, wie in Abbildung 6 zu sehen ist. Bei der 2-Bit Quantisierung kommt das Problem auch, aber viel seltener vor. In der Anwendung kann man das Problem trotzdem vereinzelt bemerken, wenn man beim Einsingen zufällig einen Ton findet, der auf dem Display eine doppelt (bzw. vervielfacht) so hohe Frequenz anzeigen lässt, als die in der der Ton tatsächlich liegt.

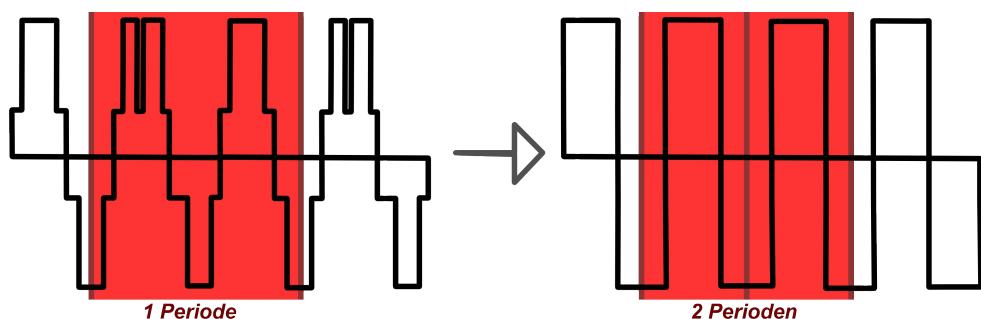


Abbildung 6: 2- und 1-Bit Quantisierung im Vergleich.

2.2.5 Rhythmis

Mit den Timer-Modulen des MSP, werden die Programmfunctionen, die man geschrieben hat, ab einer definierten Zeitangabe aufgerufen. In diesem Projekt, ist ein Timer für den Takt zuständig, der genau eine halbe Sekunde, oder je nach dem, wie schnell man den Takt haben möchte, lang sein soll. Die Töne der Melodie ist in einzelne Vierteltakte abgespeichert. Singt man einen Ton ein, wird dieser zum erwarteten Ton des aktuellen Vierteltaktes verglichen und folglich entweder als korrekt angenommen oder abgelehnt.

2.3 Fertigung

Wir bereiteten uns ein Blockschaltbild vor. Mit diesem vereinfachten Schaltplan erkennt man deutlich, welche Bausteine gebraucht werden, wie sie miteinander verbunden sind und welche Spannungen benötigt werden.

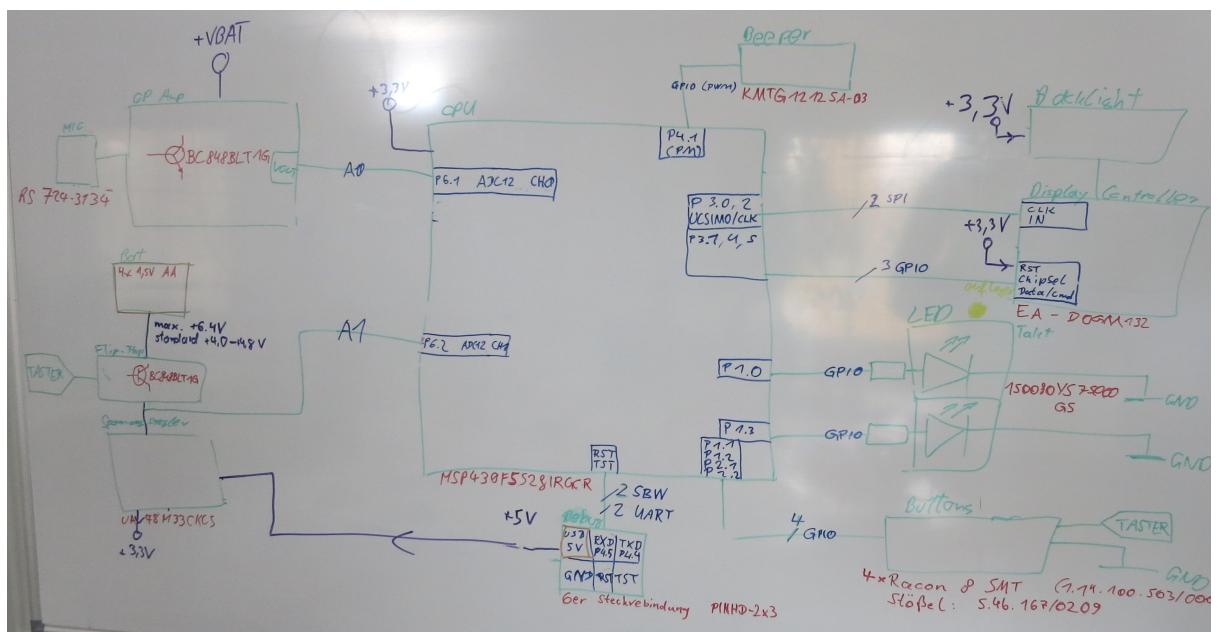


Abbildung 7: Das Blockschaltbild des Projekts.

Danach ging es zum Aufbau Schaltung. Während der Entwicklungsphase hat es sich zu Testzwecken bereits angeboten, eine nahezu komplettete Schaltung unseres Projekts mithilfe eines Steckbretts zu bauen.

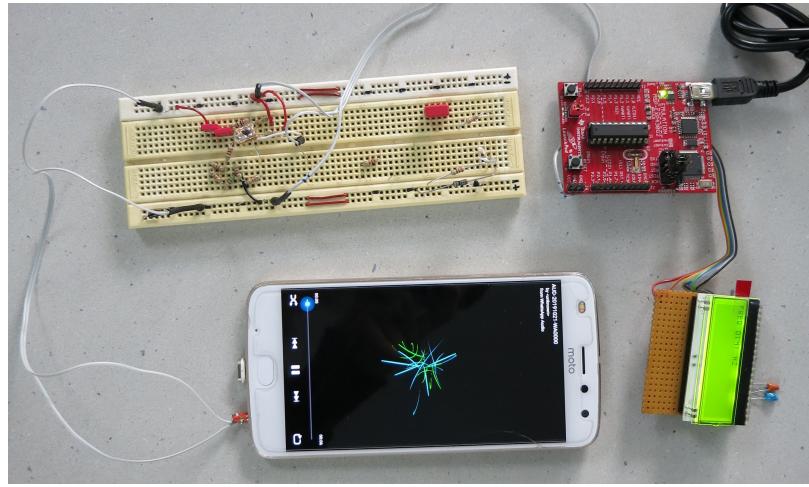


Abbildung 8: Der Prototyp der Schaltung. Anstelle eines Mikrofons, liegt hier ein Handy als Audioeingang an. Im Steckbrett ist eine Verstärkerschaltung, um innerhalb der ADC-Referenzspannung des MSP zu liegen. Das MSP sendet die berechneten Daten auf das LCD-Display.

EAGLE (Einfach Anzuwendender Grafischer Layout-Editor) ist eine Entwurfssoftware für elektronische Systeme. In dieser haben wir unsere aufgebaute Schaltung eingezeichnet und daraus ein Layout für die Leiterplatte entworfen. Der Benutzer wird auf Kurzschlüsse, zu geringe Abstände, fehlende Anschlüsse, etc. gewarnt, um am Ende eine garantiert funktionierende Schaltung zu erhalten. Wenn man mit dem Layout fertig ist und keine Warnungen mehr vorhanden sind, schickt man dieses an einen Leiterplattenhersteller, der die Leiterplatte genau anhand dieser Daten, bestenfalls ohne Korrekturen, produziert.

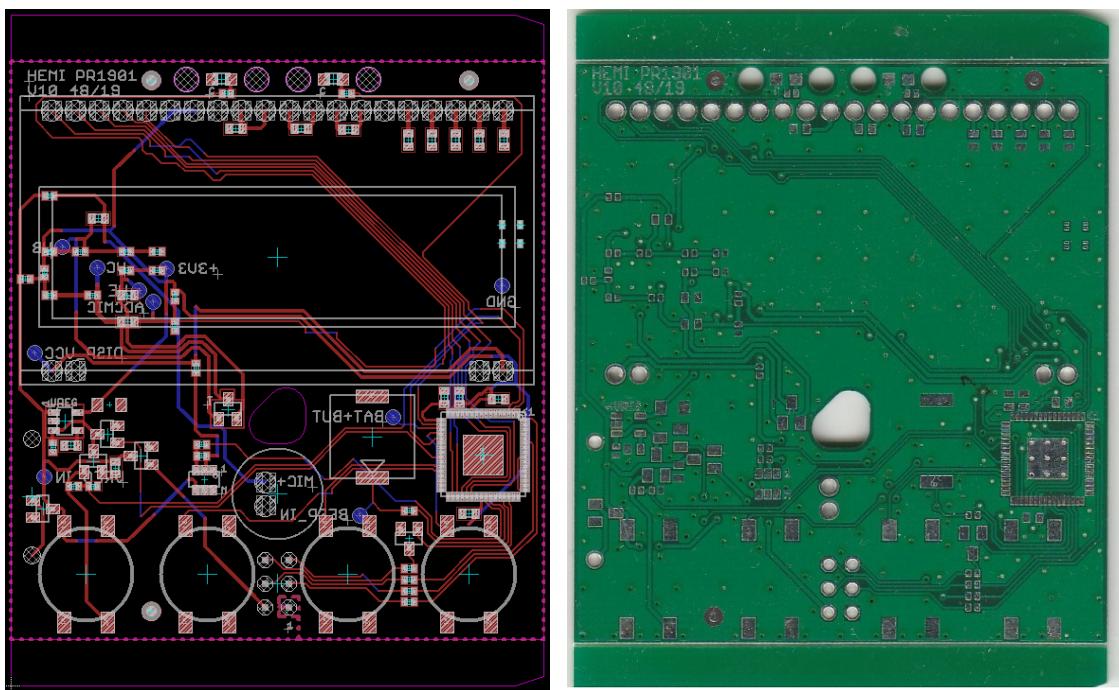


Abbildung 9: Das Leiterplatten-Layout und die daraus resultierte Leiterplatte.

Bevor die Leiterplatten zugestellt werden, programmiert man schon mal die Bestückungsmaschine, damit diese weiß, wo sie die Bauteile für diese Leiterplatte platzieren soll.

Als die Leiterplatten angekommen sind, wurden auf diese mithilfe der mitgelieferten Schablonen die Lötpaste aufgetragen. Dadurch bekommen die Bauteile später ihren Halt und den Kontakt zu den Kupferleitungen. Das Auftragen der Lötpaste hat deswegen Probleme verursacht, da wir dies per Hand gemacht haben. Falls die Schablone um 200 µm verrutschtet, hätte man mit der Lötpaste garantiert einen Kurzschluss geschaffen. Also wurde die Paste mehrere male abgeschabt und nochmal aufgetragen, bis keine potentiellen Kurzschlüsse mehr zu sehen waren. Eigentlich könnte man das Auftragen der Lötpaste einer Maschine überlassen, was aber bei unserer geringen Stückzahl von vier Stück keinen Sinn gemacht hätte.

Nach der Bestückung, kommen die Leiterplatten in die Dampfphase. In dieser werden die Leiterplatten erhitzt, wodurch die Lötpaste flüssig wird. Nach dem Abkühlen ist die Lötpaste fest.

2.4 Anwendung

Das fertige Produkt hat ein Gehäuse, vier Bedienknöpfe, eine Batterieversorgung für die übliche Benutzung sowie eine Netzversorgung zum Debuggen, zwei LEDs sowie ein LCD-Display, welches in Echtzeit die gemessene Tonfrequenz und Genauigkeit gegenüber dem Eingesungenen anzeigt. Falls die Melodie korrekt eingesungen wurde, wird auf dem Display eine vierstellige Pseudo-Zufallszahl angezeigt, die auf Basis der letzten gemessenen ADC-Werten generiert wird. Man hat die Möglichkeit, eine eigene Melodie einzusingen, die im Flash-Speicher abgespeichert wird, damit sie bei einem Programmneustart nicht verloren geht.

Theoretisch könnte man die angezeigte Zufallszahl als Kombination für ein elektronisches Zahlenschloss verwenden, indem sie verschlüsselt mit einem kabellosen Übertragungsverfahren an dieses Schloss mitgeteilt wird. Eine Minute lang ab Empfang der Kombination kann man das Schloss mit dieser aufschließen. Im Rahmen der Praktikumsdauer, konnte dieses Schloss aber nicht mehr entwickelt werden.

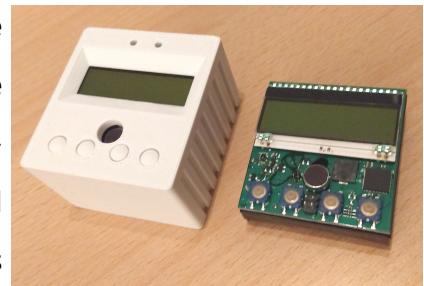


Abbildung 10: Das fertige Produkt mit und ohne Gehäuse.

3 Schlussfolgerung, Reflexion und Ausblicke

Gerade noch rechtzeitig am vorletzten Tag im Praktikum, sind wir mit unserem Projekt fertig geworden und haben es vor allen anwesenden Mitarbeitern präsentiert. Die Präsentation ist gut angekommen und wir wurden für das, wie weit wir in dem Projekt gekommen sind, gelobt.

Mit der Arbeit in meinem Praktikumsbetrieb war ich mehr als nur begeistert. Bis zum Anfang des Praktikums kam ich maximal dazu, ein Arduino zu programmieren. Die Programmierung eines MSP geht natürlich noch viel tiefer in die Maschinenprogrammierung und es hat mich fasziniert, wie viele Möglichkeiten man damit hat. Auch war mein Verständnis über die Elektrotechnik nie wirklich gut, was sich nun durchaus verbessert hat.

Das Arbeitsklima war prima. Bei Fragen konnte man sich an jeden wenden. Man durfte nur nicht immer eine simple Antwort erwarten, denn damit man dabei auch was lernt, sollte man selbst die Lösung für ein Problem finden.

Obwohl ich im Studium den Zweig Software Engineering gewählt habe, habe ich nun sogar mehr den Drang, zukünftig etwas in Richtung Embedded Systems zu tätigen. Empfehlen kann ich diesen Praktikumsbetrieb jedem Informatikstudenten, egal welchen Zweig er gewählt hat.

Literaturverzeichnis

- [1] Walter, Marian & Tappertzhofen, Stefan: Das MSP430 Mikrocontroller Buch. 5.3 Analoche Eingänge. [Online]. Verfügbar: https://www.medit.hia.rwth-aachen.de/fileadmin/MSP430Buch/msp_html_buchse21.html#x34-1120005.3 [Zugriff am 10.02.2020]
- [2] Duden: Duden [Online]. Verfügbar: <https://www.duden.de/rechtschreibung/Amplitude> [Zugriff am 10.02.2020]
- [3] Wikipedia: Arduino (Plattform). [Online]. Verfügbar: [https://de.wikipedia.org/wiki/Arduino_\(Plattform\)](https://de.wikipedia.org/wiki/Arduino_(Plattform)) [Zugriff am 10.02.2020]
- [4] Wikipedia: Autokorrelation. [Online]. Verfügbar: <https://de.wikipedia.org/wiki/Autokorrelation> [Zugriff am 10.02.2020]
- [5] Texas Instruments Incorporated: Code Composer Studio (CCS) Integrated Development Environment (IDE). [Online]. Verfügbar: <http://www.ti.com/tool/CCSTUDIO> [Zugriff am 10.02.2020]
- [6] Fluke Corporation: Was versteht man unter „Frequenz“?. [Online]. Verfügbar: <https://www.fluke.com/de-de/mehr-erfahren/bewaehrte-methoden/grundlagen-der-messtechnik/elektrizitaet/was-versteht-man-unter-frequenz> [Zugriff am 10.02.2020]
- [7] Walter, Marian & Tappertzhofen, Stefan: Das MSP430 Mikrocontroller Buch. 1.1 Mikrocontroller-Grundlagen. [Online]. Verfügbar: https://www.medit.hia.rwth-aachen.de/fileadmin/MSP430Buch/msp_html_buchse1.html#x7-40001.1 [Zugriff am 10.02.2020]
- [8] Wikipedia: TI-MSP430. [Online]. Verfügbar: https://de.wikipedia.org/wiki/TI_MSP430 [Zugriff am 10.02.2020]
- [9] Walter, Marian & Tappertzhofen, Stefan: Das MSP430 Mikrocontroller Buch. 4.1 Timer A/B. [Online]. Verfügbar: https://www.medit.hia.rwth-aachen.de/fileadmin/MSP430-Buch/msp_html_buchse15.html#x27-700004.1 [Zugriff am 10.02.2020]
- [10] Wikipedia: Zeitdiskretes Signal. [Online]. Verfügbar: https://de.wikipedia.org/wiki/Zeitdiskretes_Signal [Zugriff am 10.02.2020]

- [11] Texas Instruments: MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet (Rev. N): Seite 1. [Online]. Verfügbar: <http://www.ti.com/lit/ds/symlink/msp430f5528.pdf> [Zugriff am 03.01.2020]
- [12] Wikipedia: Abtastung (Signalverarbeitung). [Online]. Verfügbar: [https://de.wikipedia.org/wiki/Abtastung_\(Signalverarbeitung\)](https://de.wikipedia.org/wiki/Abtastung_(Signalverarbeitung)) [Zugriff am 03.01.2020]
- [13] McGee, Kevin J.: An Introduction to Signal Processing and Fast Fourier Transform (FFT): Seite 6-1. [Online]. Verfügbar: <http://fftguru.com/fftguru.com.tutorial.pdf> [Zugriff am 03.01.2020]
- [14] Wikipedia: Schnelle Fourier Transformation: Komplexität. [Online] Verfügbar: https://de.wikipedia.org/wiki/Schnelle_Fourier-Transformation#Komplexit%C3%A4t [Zugriff am 03.01.2020]

Bestätigung der inhaltlichen Richtigkeit

Unterschrift des Praktikanten

Ort, Datum

Unterschrift

Unterschrift der Firma

Ort, Datum

Unterschrift