

# FACHBERICHT

## FS18 - PRO4E - TEAM 5

5. Juni 2018

AUFTAGGEBER:	H. GYSIN
	J. KALBERMATTER
BETREUER:	M. MEIER
	A. GERTISER
	R. DUBACH
	B. DOMENGHINO
	P. SCHLEUNIGER
PROJEKTLEITUNG:	SIMON ZOLLER
TEAMMITGLIEDER:	SEVERIN HUNZIKER
	MISCHA KNUPFER
	LUKAS LOOSLI
	JOSHA GIAMBONINI
	ELIAS VON DÄNIKEN
	GIANLUCA PICCIOLA
STUDIENGANG:	ELEKTRO- UND INFORMATIONSTECHNIK

## Abstract

Aimlessly wandering around a museum because of too little knowledge is one of the main negative reactions to a museum visit. That's why no one can remember an individual work of art. To counteract this, an audio guide called Dōjō was designed to provide visitors with key information about a work of art so they can enjoy their visit more. This project should design the circuits for this Dōjō, which electronically recognises the artwork the visitor is standing in front of. Audio information stored on an internal SD-Card over a Class-D Amplifier is provided through an integrated bone sound sensor produced by adafruit. Buttons for audio control have also been integrated. In addition, the Dōjō allows the visitor to „like“ an artwork by simply pressing a button. Information about these „liked“ works are made available to the visitors either by email or a hardcopy is provided at the end of their visit. Each work of art has a Bluetooth Low Energy (BLE) Beacon that continuously sends its ID to the Dōjō. The Dōjō's internal microcontroller (nrf52832) then scans for this BLE-Signals and plays the file that belongs to the ID with the strongest signal. The Dōjō can identify beacons at XX meters. The bone sound sensor plays the audio files with a loudness of XX dB without any distortion. The inductive rechargeable battery has a capacity of 800mAh and can provide energy for about 6 hours when the Dojo is playing audiofiles and be recharged in normal charging mode within 13 hours. To protect the battery from damage, a deep discharge protection turns off the Dōjō when the battery is under 3V of its nominal voltage.

Key Words: audio guide, Bluetooth low energy, inductive charge, bone sound sensor

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>4</b>
<b>2 Gesamtkonzept</b>	<b>5</b>
2.1 Funktionsweise . . . . .	5
2.2 Anwendung . . . . .	6
<b>3 Hardware</b>	<b>9</b>
3.1 Induktive Ladung . . . . .	9
3.2 Li-Ion-Batterie . . . . .	11
3.3 Ladestation . . . . .	13
3.4 Mikrocontroller . . . . .	13
3.5 Verstärkerstufe . . . . .	14
3.6 Knochenschallaktor . . . . .	14
<b>4 Software</b>	<b>15</b>
4.1 State-Machine . . . . .	17
4.2 Nordic Software Development Kit . . . . .	23
4.3 Bluetooth . . . . .	26
4.4 SD-Karte . . . . .	30
4.5 Audio, PWM . . . . .	33
4.6 Lizenzen . . . . .	35
<b>5 Validierung</b>	<b>37</b>
5.1 Testkonzept Hardware . . . . .	37
5.2 Testkonzept Software . . . . .	42
<b>6 Schlusswort</b>	<b>48</b>
<b>7 Ehrlichkeitserklärung</b>	<b>51</b>
<b>A Anhang</b>	<b>54</b>
A.1 Messresultate . . . . .	54

## 1 Einleitung

Museen bieten die Möglichkeit unterschiedlichste Ausstellungsobjekte unter einem Dach zu betrachten. Die Art der ausgestellten Kunst ist hier von Austellung zu Austellung unterschiedlich, was jedoch bleibt ist die kreative Wahrnehmung der Besucher. Wohl nirgends kann man so gut in seinen eigenen Gedanken versinken und sich Gedanken über ein Ausstellungsobjekt machen wie in einem Museum. Um Besucher anzulocken, sind Museen auf Innovation angewiesen, welche zum einen die Übergabe von Informationen möglichst benutzerfreundlich gestaltet, aber auch ein angenehmes Ambiente schafft. Hierbei kommt auch vermehrt der Einsatz von Smartphones zum Zuge, wobei die Problematik darin besteht, dass man der Aussenwelt gefährlich nahe kommt und dadurch abgelenkt wird. Um diesem Problem entgegenzuwirken, wird ein von Frau J. Kalbermatter designter Audioguide namens Dōjō entwickelt, welcher Kunstobjekte drahtlos erkennen und darüber gespeicherte Informationen via Körperschalltechnik zum Museumsbesucher bringen kann. Durch eine Vielzahl von weiteren Funktionen ist ein neuer Informationsaustausch an den Museumsbesucher möglich, und Ablenkungen zur Aussenwelt sind trotz modernster Technik in weiter Ferne.

Ziel im Projekt 4, Studiengang Elektro- und Informationstechnik an der Fachhochschule Nordwestschweiz, war es das funktionelle Konzept von Frau Kalbermatter durch die Verwendung von elektrotechnischen Bauteilen zu realisieren. Dazu wurde die drahtlose Erkennung der Kunstobjekte mittels Bluetooth Low Energy (BLE) Beacons erreicht. Genannte Beacons müssen in unmittelbare Nähe der Kunstobjekte angebracht sein. Die Informationen zu den Kunstobjekten wurden als Audiofiles auf einer herausnehmbaren SD-Karte gespeichert und werden zum Abspielen via PWM-Ausgang des Mikrocontrollers (nRF52832) über einen Klasse D Verstärker auf den Knochenschallaktor gegeben. Tasten für die Wiedergabekontrolle (Start, Stopp, Lauter, Leiser) wurden implementiert sowie die erwähnte „Like“-Taste. Außerdem verfügt der Dōjō über einen Li-Ionen-Akku, welcher induktiv geladen wird. Damit der Dōjō gänzlich drahtlos bleibt, erfolgen Datendownload und Konfiguration ebenso über Bluetooth. Der integrierte Mikrocontroller beinhaltet die Software und übernimmt somit die Erkennung, Ansteuerung und Koordination der Hardware.

Es wurde ein Prototyp der Elektronik realisiert. Er kann zu vier verschiedenen Sprachen XXX Stunden Audioausgabe speichern. Die Ansteuerung der Audiofiles erfolgt über Bluetooth-Beacons, welche bis zu einer Distanz von XXX m erkannt werden. Die eingebaute «Like»-Taste ermöglicht, favorisierte Kunstobjekte zu vermerken und die dazugehörigen Informationen am Ende des Besuches digital oder in Form einer Broschüre beim Ausgang als Erinnerung mitzunehmen. Außerdem besitzt der Dōjō einen integrierten Akku mit einer Kapazität von 800 mAh, welcher bei pausenloser Audioausgabe genug Energie für rund 6 Stunden liefert. Die Induktionsladung lädt den Akku im Normalbetrieb zu 100% innerhalb 13 Stunden. Außerdem sorgt ein Tiefenentladungsschutz der Batterie dafür, dass der Dōjō bei einer Betriebsnennspannung von unter 3V ausgeschaltet wird.

Der nachfolgende Bericht umfasst drei Hauptbereiche. Der erste Bereich (Kapitel 2) umfasst das Gesamtkonzept, welcher die gesamte Anwendung auslegt. Die nachfolgenden zwei Hauptbereiche sind in Hardware (Kapitel 3) und Software (Kapitel 4) gegliedert. Die Hardware teilt sich wiederum in die Themengebiete Energieübertragung (Kapitel 3.1), Energiespeicherung (Kapitel 3.2) und Audioausgabe über den Knochenschallaktor (Kapitel 3.5) auf. Die Software beinhaltet die Unterbereiche der State Machine (Kapitel 4.1), Bluetooth (Kapitel 4.3), sowie die gesamte Programmstruktur der SD-Karte (Kapitel 4.4) und Audioausgabe über PWM (Kapitel 4.5). In Kapitel 5 befindet sich die Validierung.

## 2 Gesamtkonzept

Bevor auf die einzelnen Komponenten des Dōjōs eingegangen werden kann, ist ein Einblick in das Gesamtkonzept notwendig. Hauptbestandteil werden in diesem Kapitel der Aufbau, die Funktionsweise des Gerätes wie auch die Anwendung im Museum sein. Beim Aufbau ist die Kommunikation der einzelnen Komponenten untereinander ersichtlich. Dieser Abschnitt wird durch die jeweiligen Funktionsweisen abgerundet. Der letzte Teil gibt einen Einblick in die Anwendung vor und während dem Museumsbesuch und welche technischen Möglichkeiten dem Besucher zur Verfügung stehen.

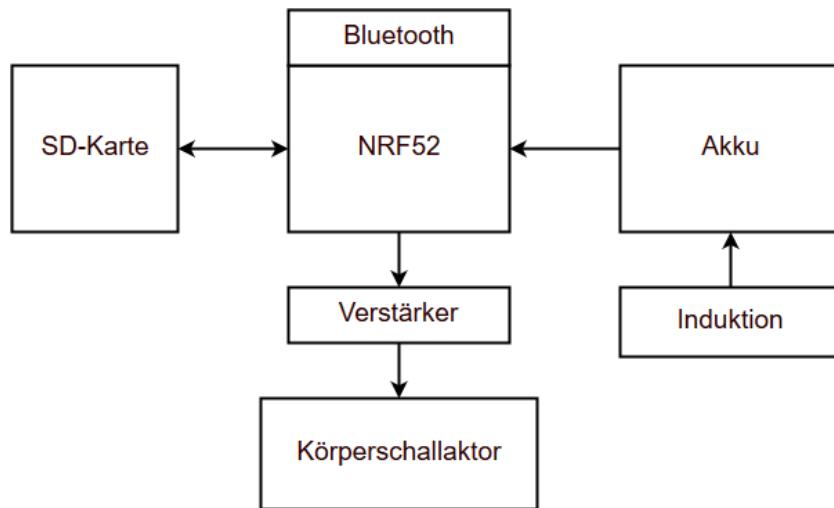
### 2.1 Funktionsweise

Der Dōjō ist eine Art Audioguide, welcher für Museen designt wurde und diverse Funktionen beinhaltet. Abbildung 1 zeigt den von der Auftraggeberin designten Prototypen. Einer der grössten Unterschiede zu einem herkömmlichen Audioguide ist die Sprachausgabe mittels einem Knochenschallaktor und nicht wie gewöhnlich mit einem Lautsprecher. Eine weitere Eigenheit ist der integrierte „Like“-Button, mit dem man die entsprechenden Ausstellungsstücke „liken“ kann. Dadurch lässt sich für jeden Besucher individuell eine persönliche Kunstobjektliste zusammenstellen, die nur noch die entsprechenden Objekte mit einem „Like“ enthält. Das ermöglicht es, am Ende des Besuchs jeder Person eine Liste mit den jeweiligen Informationen zum Ausstellungsobjekt auszuhändigen. Ansonsten verfügt der Dōjō über die selben Funktionen, die man von einem üblichen Audioguide erwarten würde, wie z.B. der Audiomeldung, dem Haltemodus oder der Lautstärkeregelung.



**Abbildung 1:** Dōjō als Modell

Das Herzstück des Dōjōs ist ein zentraler NRF52-Mikrocontroller von Nordic Semiconductor mit integriertem und low-Energy fähigem Bluetooth-Stack. Die verwendeten Daten werden auf einer SD-Karte gespeichert. Bei Bedarf werden die Audiodateien an den Verstärker weitergegeben und schliesslich über den Knochenschallaktor wiedergegeben. Die Energieversorgung wird mit einer Li-Ion-Batterie sichergestellt, welche über ein induktives Ladesystem versorgt wird. Abbildung 2 zeigt das technische Lösungskonzept.



**Abbildung 2:** Teilsysteme des Dōjōs

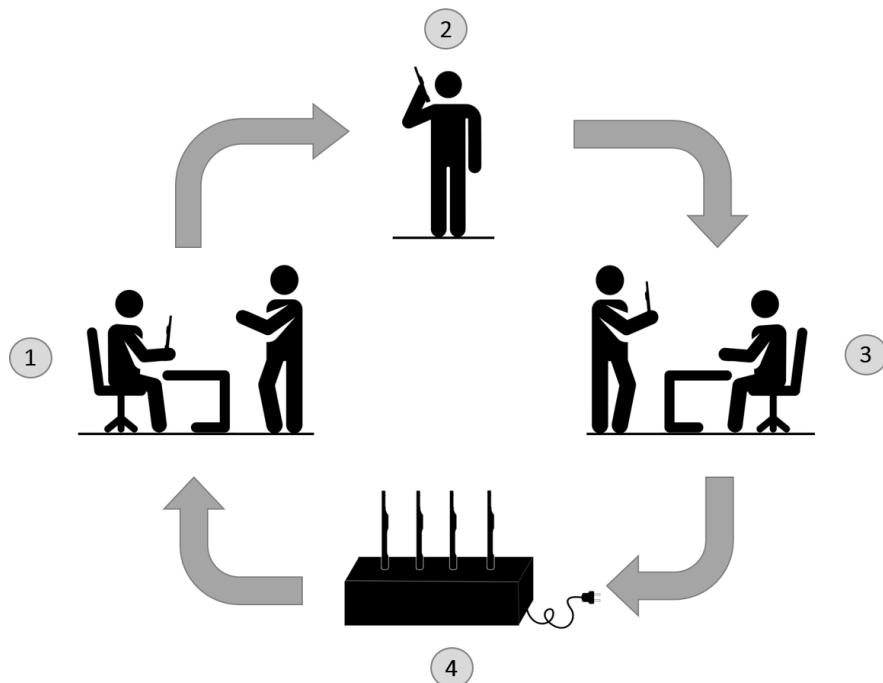
Die Funktionen des Dōjōs sind in zwei Bereiche unterteilt. Zuerst wird die Funktionalität aus Sicht des Users beschrieben. Anschliessend werden die relevanten Funktionen aus Sicht der Betreiber erläutert.

Der **User** geht mit dem Dōjō durch das Museum. Sobald die Bluetooth Beacons genügend nahe sind, erhält der Nutzer durch eine aufleuchtende LED ein Signal. Jetzt kann er entscheiden, ob er sich das zugehörige Audio-File anhören will. Trifft dies zu, kann er den Play-Button betätigen und die Audiodatei wird wiedergeben. Die Lautstärke kann über die Volume-Buttons justiert werden. Falls das Ausstellungsstück dem Nutzer gefallen hat, kann er die „Like“-Taste drücken. Diese speichert das Ausstellungsstück auf eine Liste in der internen SD-Karte. Am Ende des Museumsbesuches kann diese Liste ausgewertet werden, wobei die Verwendung dieser Daten nicht Bestandteil der Projektarbeit ist und aus diesem Grund nicht weiterverfolgt wird.

Der **Betreiber** hat die Aufgabe den Dōjō zu konfigurieren. Dies erfolgt über die dafür vorgesehene SD-Karte, welche mit dem Computer beladen wird. Anschliessend kann die Speicherkarte in den Dojō eingeführt werden. Die Anwendung wird über eine induktive Ladestation versorgt. Die nächsten beiden Funktionen sind Wunschziele, die vor allem mit Rücksicht auf die Laufzeit realisiert werden. Den Bluetooth-Receiver könnte man kurzzzeitig auf ein Bluetooth Beacon umschalten. Der Betreiber müsste nur noch einen Receiver pro Raum installieren. Damit könnte man die gewünschte Lokalisierung der Besucher umsetzen. Das zweite wäre die Möglichkeit per Bluetooth einzelne Audiofiles auf den Dōjō zu übertragen, um im Falle einer Änderung der Ausstellung, die Liste anzupassen und aktuell halten zu können.

## 2.2 Anwendung

Nachdem die Unterteilung zwischen Betreiber und User gemacht wurde, kann nun die Betriebsebene verdeutlicht werden. Um einen lückenlosen Betrieb zu gewährleisten, ist ein Konzept für den Ablauf von Vorteil. Nachfolgend werden die Hauptschritte erklärt. Abbildung 3 dient zur Übersicht.

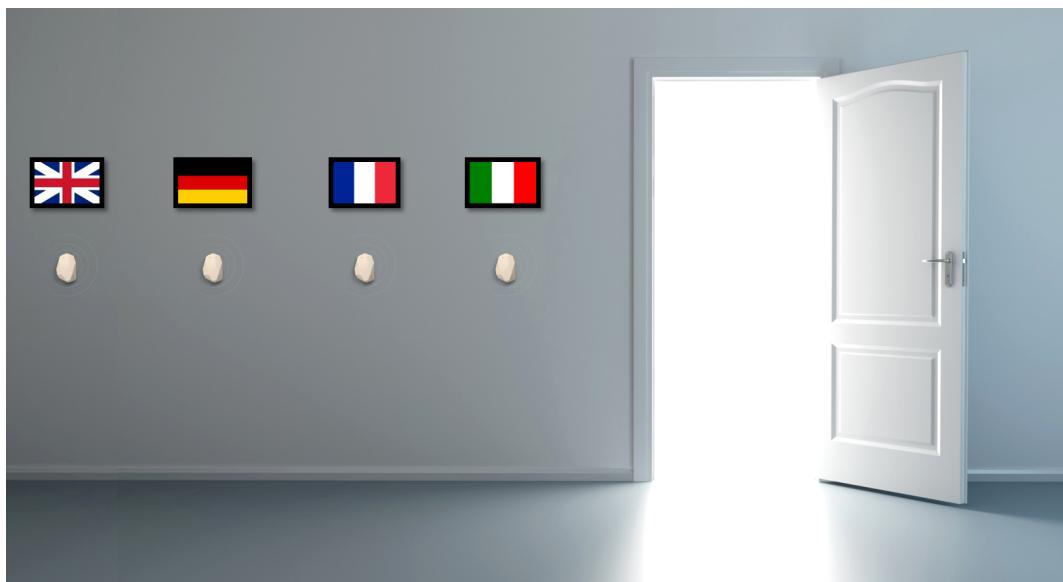
**Abbildung 3:** Anwendungsablauf

**Schritt 1** beinhaltet die Ausgabe des Dōjōs am Empfang. Es werden jeweils die Geräte mit dem höchsten Ladestatus abgegeben. Ein lückenloser Betrieb wird erreicht, wenn die Stückzahl der Audio-Guide Geräte in etwa der Anzahl der Besucher pro Tag entspricht. Die Zutrittsberechtigung wird gemäss dem Wunsch des Besuchers festgelegt. Anschliessend wird die Sprache durch den Besucher selbst gewählt. Hierbei stehen ihm vier Bluetooth-Beacons zur Verfügung, zu welchen er sein Dōjō hinkommen kann. Die gewünschte Sprache ist hierbei durch die Landesflagge gekennzeichnet. Abbildung 4 zeigt das Prinzip. Damit eine der vier Sprachen auf dem Dōjō aktiviert wird, muss das Gerät an den richtigen Beacon gehalten werden und gleichzeitig die Play-Taste gedrückt werden. Wurde die Sprache erfolgreich ausgewählt, wird ein kurzes Audio-Sample zur Überprüfung der gewählten Sprache abgespielt. Sobald die gewünschte Sprache geladen und getestet wurde, kann der Museumsbesuch gestartet werden.

In **Schritt 2** befindet sich der Besucher auf dem Rundgang mit dem Dōjō als Audio-Guide. Er hat dabei die Möglichkeit während dem Rundgang Bilder zu „liken“ und sich die zugehörigen Audio-Files der Kunstobjekte anzuhören.

Die Abgabe der Gerätes erfolgt in **Schritt 3**. Hier hat der Besucher die Möglichkeit die Kunstobjekte, welche mit einem „Like“ versehen wurden, als Broschüre zu erhalten oder per Mail zu erhalten. Das entgegengenommene Dōjō kann für den nächsten Besucher gereinigt werden.

In **Schritt 4** wird der Dōjō entgegengenommen und alle benötigten Informationen extrahiert. Danach wird das Gerät mit der induktiven Ladestation geladen. Für weitere Informationen zur Ladestation wird auf das Kapitel 3.1 verwiesen. Nun kann der Zyklus wieder von vorne beginnen.



**Abbildung 4:** Sprachauswahl mittels Bluetooth-Beacon

## 3 Hardware

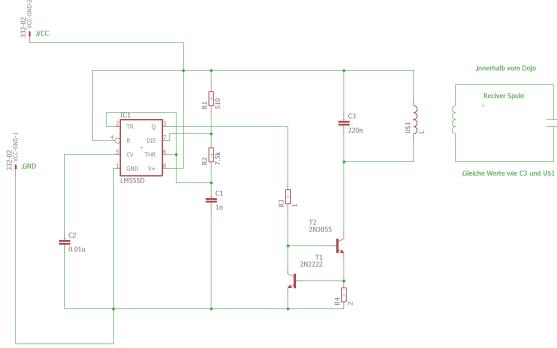
Nachdem die ganze Struktur und der Funktionsablauf der Anwendung detailliert erklärt wurde, soll nun an dieser Stelle der Dōjō in seine Komponenten zerlegt und genauer erläutert werden. Die einzelnen Systeme sind logisch geordnet und werden in den nachfolgenden Kapiteln erklärt. Zuerst wird die induktive Energieübertragung, die Energiespeicherung und die dazu notwendige Regelung erläutert. Anschliessend werden die Komponenten Mikrocontroller, Verstärkerstufen und Knochenschallaktor beschrieben.

### 3.1 Induktive Ladung

Der Dōjō wird mit Hilfe des Induktionsprinzips geladen. Der Ladeprozess wird gestartet, sobald der Dōjō auf die Ladestation gestellt wird. Beim Induktionsprinzip wird die Energie mithilfe von Spulen über eine kurze Distanz zwischen zwei Schaltungen transportiert. Die erste Schaltung, von welcher aus die Energie gesendet wird, wird Transceiver genannt. Diese Schaltung besteht aus einem Pulsgenerator, mit welchem das LC-Glied gepulst wird. Sie macht den Hauptanteil einer solchen Induktiven Ladeschaltung aus. Die zweite Schaltung, welche die Energie des Transceivers empfängt, wird Receiver genannt und besteht ebenfalls aus einem LC-Glied mit einem Gleichrichter. Nachfolgend werden beide Teile der Schaltung beschrieben.

#### Transceiver

Der Pulsgenerator wird einer Timer-Schaltung umgesetzt. Verwendet wird hierbei das elektronische Bauelement NE555. Dieses bietet den Vorteil, dass das notwendige Pulssignal für die Übertragung einfach erstellt und verändert werden kann. Der NE555 enthält eine monolithisch integrierte Zeitgeberschaltung, die sich aufgrund ihrer Eigenschaften als Taktgeber, Oszillator und für Zeitverzögerungen verwenden lässt. Bevor der Timer zu schalten beginnt, müssen verschiedene Spannungsschwellen erreicht werden. Diese lassen sich durch extern angeschlossene Widerstände und Kondensatoren einstellen. Massgebend für die Veränderung der Pulsdauer, Frequenz und des Duty-Cycles sind die Verhältnisse der Komponenten. Das entstandene Pulssignal wird schlussendlich an das LC-Glied gegeben. Dafür wird das LC-Glied an die Versorgungsspannung gehängt und in Serie dazu der Collectoranschluss eines 2N3055 Leistungs NPN-Transistor angeschlossen. An dessen Emitter wird nun ein Niederohmiger Widerstand auf *GND* gehängt, wobei die über ihm abfallenden Spannung von einem weiteren Transistor überwacht wird und somit den Strom begrenzt. Dieser „2N2222 Strombegrenzungs-Transistor“ wird zwischen dem Pulssignal, welches den Leistungs-Transistor steuert, und *GND* gehängt. Wird nun der Strom und somit auch die Spannung über dem Widerstand zu hoch, so schliesst der „2N2222 Strombegrenzungs-Transistor“ das Pulssignal kurz, wodurch der 2N3055 Leistungstransistor nicht mehr sauber durchgesteuert wird und dadurch den Strom begrenzt. Die Strombegrenzung ist abhängig von der verwendeten Peripherie des LC-Gliedes. Es gilt zu beachten, dass die Spule durch ihre kleine Bauform weniger Strom verträgt als der Leistungstransistor. In unserem Fall liegt der maximal zulässige Strom für die Spule bei 0.6A. Bei der Wahl von grösseren Spulen muss jedoch darauf geachtet werden, dass die Begrenzung von 15A nicht überschritten wird, da dies die Belastungsgrenze für den 2N3055 Leistungstransistor ist.



**Abbildung 5:** Die verwendete Timerschaltung als Pulsquelle

Das Pulssignal selber muss verschiedene Kriterien erfüllen. Zum einen sollte der Duty Cycle so nahe wie möglich an 50% sein. Um dies zu erreichen muss  $R_2 \gg R_1$  gelten. Das andere Kriterium ist die Erreichung der Resonanzfrequenz des LC-Gliedes. Um das Pulssignal optimal einzustellen, können folgende Richtlinien betrachtet werden:

- **C** beeinflusst die Zeiten (Frequenz/High-Time/Low-Time)
- **R<sub>1</sub>** beeinflusst die High-Time, lässt jedoch die Low-Time unverändert.
- **R<sub>2</sub>** beeinflusst die High- und Low-Time und beeinflusst somit den Duty Cycle.

Die verwendeten Komponenten  $C$ ,  $R_1$  und  $R_2$  wurden durch nachfolgende Formeln ?? bis 3.11 berechnet.

$$f = 1/T = 1.44/((R_1 + R_2 * 2) * C) \quad (3.1)$$

$$LowTime = 0.693 * R * C \quad (3.2)$$

$$HighTime = 0.693 * (R_1 + R_2) * C \quad (3.3)$$

$$D = DutyCycle = (R_1 + R_2)/(R_1 + 2 * R_2) \quad (3.4)$$

$$R_1 = 1.44 * (2 * D - 1)/(F * C) \quad (3.5)$$

$$R_2 = 1.44 * (1 - D)/(F * C) \quad (3.6)$$

$$T_L = 0.693 \cdot R \cdot C \quad (3.7)$$

$$T_H = 0.693 \cdot (R_1 + R_2) \cdot C \quad (3.8)$$

$$D = DutyCycle = \frac{R_1 + R_2}{R_1 + (2 \cdot R_2)} \quad (3.9)$$

$$R_1 = 1.44 \cdot \frac{2 \cdot (D - 1)}{F \cdot C} \quad (3.10)$$

$$R_2 = 1.44 \cdot \frac{1 - D}{F \cdot C} \quad (3.11)$$

Für die Berechnung der effektiven Werte, müssen die esten Werte angenommen werden. Unsere Berechnungen ergaben folgende für den Tranceiver am besten geeigneten Werte:

$$\begin{aligned} C &= 1nF \\ R1 &= 200\Omega \\ R2 &= 9k\Omega \end{aligned}$$

Die Frequenz (Formel ??) beträgt  $79.12kHz$ . Nach der Berechnung der Komponenten folgt die Implementation in das Gerät. Es ist zu beachten, dass Abweichungen sich unmittelbar auf die Frequenz, die Pulsdauer und den Duty Cycle des Pulssignales auswirken. Eine solche Abweichung kann auch durch eine leicht abweichende Resonanzfrequenz des LC-Gliedes auftreten. In Kapitel 3.3 wird ein Prototyp einer von uns entworfenen Ladestation vorgestellt, welche auch die gesamte Tranceiverschaltung implementiert.

Wie der Receiver aufgebaut ist, wird im nachfolgenden Abschnitt genauer erläutert.

### Receiver

Der Receiver besteht primär aus einem LC-Glied und einem Gleichrichter. Speziell ist, dass bei der Tranceiverschaltung das selbe LC-Glied verwendet wurde wie bei der Receiverschaltung. Dies aus dem Grund, weil das Energiefeld sehr klein ist und im Falle einer grösseren Spule das Receiver Glied nicht optimal ausgenutzt werden könnte. Das L-Glied (Flachspule) lässt sich mit einer Dimension von  $\varnothing 15mm$  Durchmesser und  $2mm$  Höhe gut im inneren des Dōjō's platzieren. Die Positionierung findet am Boden statt und ermöglicht dadurch die besten Übertragungswerte von Strom und Spannung. Der notwendige Kondensator für die Vervollständigung des LC-Gliedes kann direkt hinter der Spule montiert werden. Die hochfrequente Wechselspannung welche nun gemessen werden kann, muss für die Speisung der Batterie noch gleichgerichtet werden. Verwendet werden hierbei sowohl Kondensatoren als auch spezielle Gleichrichterdioden, welche eine Abfallspannung von lediglich  $0.1V$  aufweisen. Anschliessend wird die gesamte Ladeschaltung gespiesen, welche den gesamten Ladeprozess des Akkus übernimmt. Einen Einblick in diesen Ladeprozess gibt nachfolgendes Kapitel.

## 3.2 Li-Ion-Batterie

Die gesamte Energiespeicherung wird mit einem Lithium-Ionen-Akkumulator des Typs Emmerich LI14500 realisiert. Dieser weist eine Kapazität von  $800mAh$  bei einer Nominalspannung von  $3.7V$  auf und verfügt über interne Schutzeinrichtungen. Um eine Abschätzung über die Betriebszeit des Dōjōs zu erhalten, sind Faktoren wie maximaler Verbrauch, Nominalspannung und Kapazität notwendig. Die maximale Leistung des Dōjōs lässt sich durch die Leistung des Verstärkers/ Knochenschallgebers ( $P_{Kn}$ ) und die des Mikrocontrollers ( $P_{MC}$ ) beschreiben. Alle anderen Komponenten können aufgrund des geringen Betriebsstrom mit einem Sicherheitsfaktor ( $P_{zus}$ ) von  $0.1W$  dazu gerechnet werden. Die Leistung des Verstärkers weist eine maximale RMS-Leistung von  $471.9mW$  auf. Gerechnet wird mit 80% des RMS Wertes, da der Knochenschallgeber nicht permanent beansprucht wird. Die Leistungsaufnahme des Mikrocontrollers beträgt  $27.36mW$  und wurde durch Messungen ermittelt. Die Gesamtleistung wird wie folgt berechnet:

$$P_{max} = (0.8 \cdot P_{Kn}) + P_{MC} + P_{zus} = (0.8 \cdot 0.472W) + (0.02736W) + 0.1W = 0.506W \quad (3.12)$$

Die Gesamtleistung beträgt somit rund  $0.506W$ . Nun kann basierend auf der Gesamtleistung die Betriebszeit  $t_{max}$  berechnet werden:

$$t_{max} = \frac{W \cdot U}{P_{tot}} = \frac{800mAh \cdot 3.7V}{506mW} = 5.85h \approx 5h\ 51min \quad (3.13)$$

Es ist somit ersichtlich, dass bei permanentem Einsatz eine maximale Betriebszeit von knapp 6 Stunden erzielt werden kann. Allerdings kann ein lückenloser Betrieb durch einen geschickten Ladeprozess erzielt werden.

### Schutzeinrichtungen

Um den verwendeten Akkumulator zu schützen, sind diverse Schutzeinrichtung notwendig. Zum einen muss der Ladevorgang überwacht werden, so dass der maximale Ladestrom wie auch die Ladespannung nicht überschritten werden. Für die Laderegelung wurde ein Lade-IC von Microchip des Typs MCP73831 verwendet. Dieser übernimmt die gesamte Spannungs- und Stromregelung beim Ladeprozess und kann parallel zum Ladevorgang eine LED zur Ladesignalisation ansteuern. Der Ladezyklus ist in der Abbildung 6 ersichtlich. Hierbei wurde der Akku im Schnelllademodus mit einem maximalen Strom von  $400mA$  geladen. Dieser Strom ergibt sich aus dem Datenblatt der Batterie [1], wobei sowohl der Entladestrom, als auch der Ladestrom  $0.5 \cdot C$  pro Stunde beträgt.  $C$  entspricht der Batteriekapazität. Somit lässt sich der Strom folgendermassen berechnen:

$$I_{charge} = 0.5 \frac{1}{h} \cdot 800mAh = 400mA \quad (3.14)$$

Betrachtet man die Abbildung 6, so wird ersichtlich, dass die Spannung rund  $2.5h$  geregelt wird bis  $4.2V$  Grenze erreicht wird. Sobald der Spannungswert  $4.2V$  erreicht hat, beginnt der Lade-IC mit der Stromregelung. Für diesen Prozess wurden beim Versuch noch einmal rund 30 Minuten benötigt, wodurch die letzten rund 20% der Batteriekapazität geladen werden konnten.

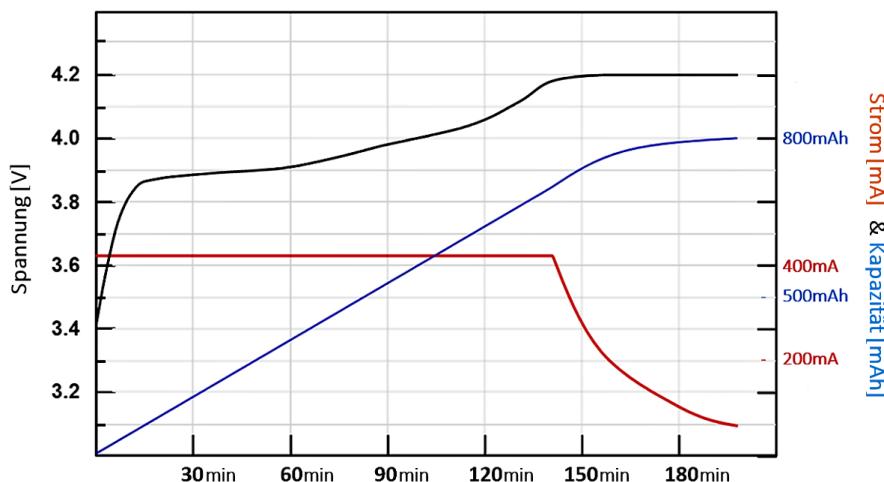


Abbildung 6: Laderegelung Li-Ion Akku

Für einen weiteren Schutz, hat die Emmerich LI14500 eine integrierte Schutzbeschaltung namens PCM (Protection Circuit Module). Dieser Schutz garantiert einerseits einen Überladeschutz von

$4.25V \pm 0.025V$ , aber auch einen Tiefentladungsschutz von  $2.5V \pm 0.063V$ . Weiter ist der Akku gegen Überströme ab einer Höhe von 4.8A geschützt.

### 3.3 Ladestation

Nachdem die gesamte induktive Ladeschaltung und Energiespeicherung beschrieben wurde, folgt nun die Beschreibung der Ladestation. Hierfür wurde ein Prototyp erstellt, welche zum einen die gesamte primäre induktive Ladeschaltung beinhaltet. Für den Prototypen wurde eine .stl Datei erstellt, welche mit einem 3D-Drucker gedruckt wurde. Wichtig ist hierbei zu erwähnen, dass es sich nur um einen Prototypen handelt und es bei einer Weiterentwicklung noch Anpassungen geben kann. Da die Ladestation für Versuchszwecke bereits erstellt wurde, ist das Design so gewählt, dass nur ein Dōjō geladen werden kann. Dies könnte in einem weiteren Schritt auf mehrere Ladeaussparungen erweitert werden, wobei mehrere Dōjōs gleichzeitig pro Ladestation geladen werden können. Abbildung 7 zeigt die Ladestation aus verschiedenen Perspektiven.



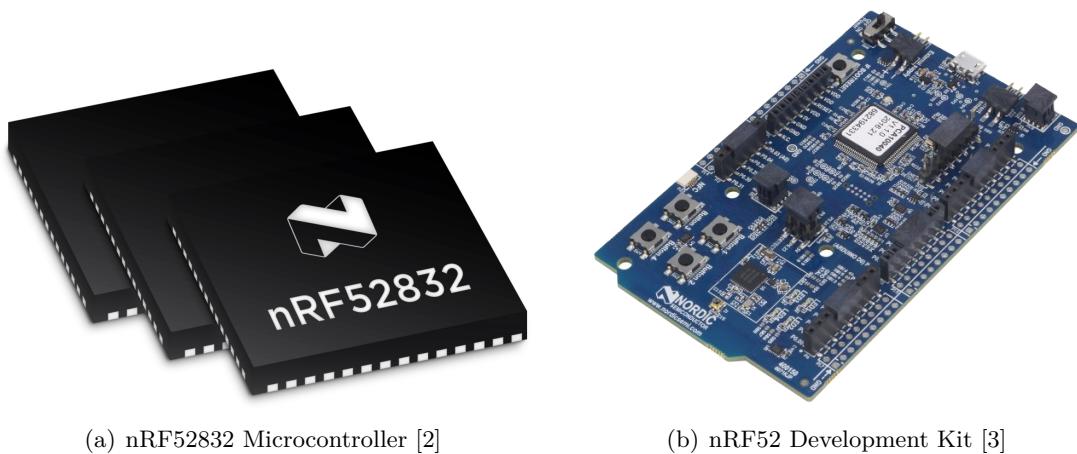
**Abbildung 7:** Prototyp der Ladestation aus verschiedenen Perspektiven

Die Öffnung dient zum einen als Standhalterung und zum anderen als korrekte Positionierung für die induktive Ladeschaltung. Die richtige Positionierung ist hierbei eines der wichtigsten Kriterien für einen optimalen Ladezyklus, da die Tranceiver- und Receiverspule direkt übereinanderliegend den besten Wirkungsgrad erzielen. Der Hohlraum im Inneren der Ladestation dient zur Platzierung des Primärkreises und hat die Abmessung (80 x 70.7 x 30)mm. Für den Prototypen wurde eine Lochrasterplatine mit allen nötigen Komponenten gefertigt, welche genau in diese Aussparung passt. Weiter sind runde Löcher (5mm Ø) ersichtlich, welche in die Kammer des Schriftzuges führen. Sie sind für LEDs vorgesehen, welche den Dōjō Schriftzug bei angeschlossener Versorgungsspannung zum Leuchten bringt.

### 3.4 Mikrocontroller

Für die Steuerung der Anwendung wird nun noch ein zentraler Controller benötigt. Die Wahl fiel auf den Mikrocontroller nRF52832 von Nordic Semiconductors. Seine hohe Performance ermöglicht es ein System aufzubauen, welches nur eine zentrale Schnittstelle beinhaltet. Abbildung 8 zeigt den Controller als Einzelbauteil und als Entwicklungskit-Kit.

Der nRF52832 weist eine Betriebsversorgungsspannung zwischen 1.7V und 3.6V mit einem Versorgungsstrom von 5.4mA auf. Diese niedrigen Werte ermöglichen einen dauerhaften Betrieb durch die integrierte Batterie. Die Speicherkapazität ist durch 512kB flash/64kB RAM Speicher gegeben. Für die Entwicklung des Prototypen wurde das Entwicklungsboard verwendet. Dies hat verschiedene Vorteile, wie z.B. die integrierte Bluetooth-Antenne, die einfache Verwendung von sogenannten Shields und das einfache Handling für Messungen und provisorische Verbindungen.



**Abbildung 8:** Mikrocontroller als Einzelement und als Entwicklungsanwendung

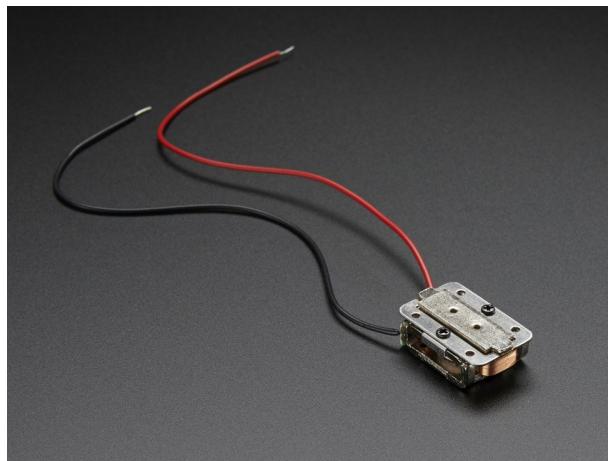
### 3.5 Verstärkerstufe

Nachdem die Steuerungseinheit gezeigt wurde, kann nun die Ausgangsstufe erläutert werden. Mit einer Verstärkerstufe lassen sich auf einfache Art und Weise Signale jeglicher Form verstärken. Sie eignen sich bestens, um den Ausgang eines Mikrocontrollers entsprechend aufzubereiten, da die Ausgangsseite meist sehr niedrige Ströme aufweist. Dadurch kann dem Knochenschallaktor genügend Energie zur Verfügung gestellt werden. Prinzipiell gibt es zwei Arten von Verstärkern. Entweder erfolgt die Umsetzung digital oder analog. Beide erfüllen die gleiche Aufgabe, weisen jedoch bezüglich Wirkungsgrad einen deutlichen Unterschied auf. Die digitale Variante weist ungefähr einen Wirkungsgrad von 90% auf [4], während die analoge Variante einen maximalen Wirkungsgrad im Bereich der Leistungsanpassung erzielt [5]. Aus diesem Grund wird ein digitaler Verstärker (Class-D-Verstärker) in der Anwendung implementiert. Die Wahl fiel auf den Stereo-Amplifier MAX 98306. Der Verstärker hat einen Stromverbrauch von  $143mA$  und eine Speisespannung von  $3.3V$ . Somit hat er einen Leistungsverbrauch von  $471.9mW$ . Im Standby benötigt er lediglich  $2mA$  und somit  $6.6mW$ [?].

### 3.6 Knochenschallaktor

Nachdem das vom Mikrocontroller ausgegebene Audio-File über die Verstärkerstufe entsprechend aufbereitet wurde, kann nun die Audiodatei über einen sogenannten Knochenschallaktor ausgegeben werden. Der Aktor arbeitet nach dem Prinzip der Weiterleitung von Schall-Schwingungen oder auch Vibrationen. Dadurch lässt sich der ursprüngliche Gehörgang umgehen und die Schwingungen werden über den Schädelknochen an das Innenohr übertragen. Dies verbessert auch die Hygiene der Anwendung, da kein direkter Kontakt mit dem Gehörgang stattfindet.[6] Für die Anwendung im Dōjō wird ein Knochenschallaktor des Herstellers Adafruit verwendet, welcher in der Abbildung 9 ersichtlich ist.

Das ausgewählte Bauteil eignet sich bestens für die Verwendung im Dōjō. Mit einem Gewicht von  $9.6\text{ g}$  und den Dimensionen  $14x21,5x7,9$  lässt sich der Aktor gut in das bestehende Gehäuse implementieren [4]. Weiter ist das Bauteil relativ kostengünstig im Handel erhältlich und kann  $1W_{RMS}$  Leistung liefern, was sich dann in der Lautstärke bemerkbar macht. Nach ausführlichen Recherchearbeiten konnten keine wirklichen Alternativen ausgemacht werden. Meist befindet sich die Technologie noch in der Entwicklungsphase oder fällt aufgrund des Preises aus der Auswahlmöglichkeit.



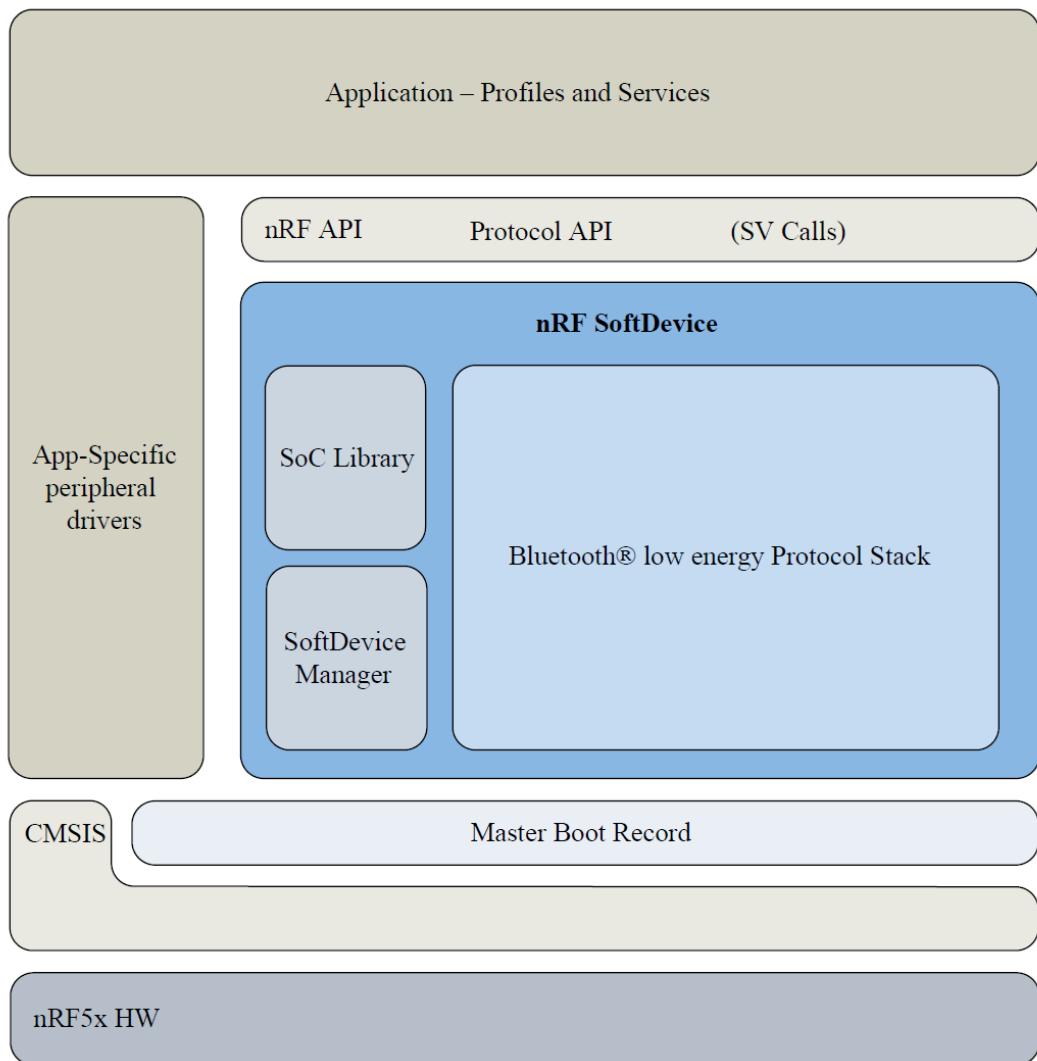
**Abbildung 9:** Knochenschallaktor von Adafruit

## 4 Software

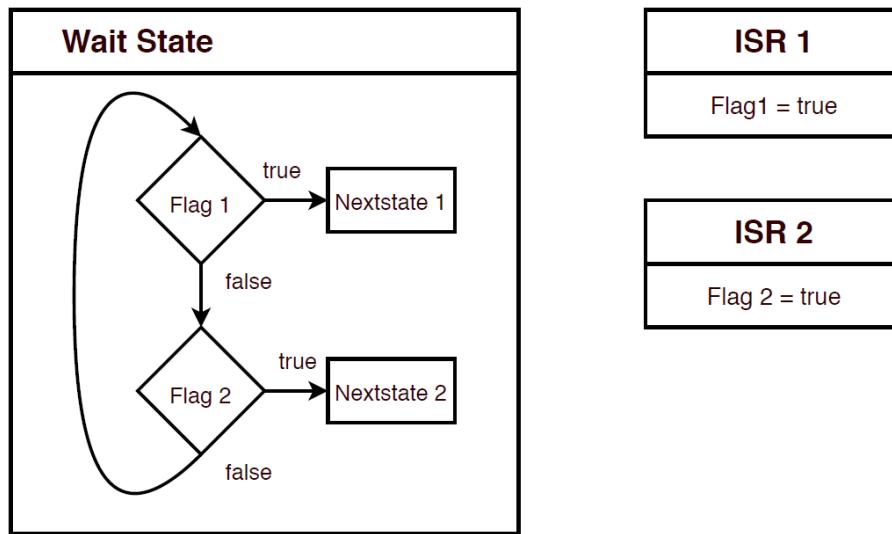
Nachdem das Kapitel Hardware ausführlich behandelt wurde, muss nun noch die Software dazu beschrieben werden. Der verwendete Mikrocontroller legt die Verwendung des Software Development Kit [7] (SDK) nahe. Dies ist eine Sammlung von Beispielen, Librarys und vorcompilierten Codes. Verwendet wurde die Version v12.3.0. Die SDK ist aufgrund des verwendeten integrierten Bluetooth-Stack wichtig. Dieser ist im sogenannten Softdevice enthalten. Um ihn nutzen zu können, verwenden wir den S132. Dieser und die nötige Initialisierungen des Bluetooth-Stacks waren in dem Beispielprojekt Uartc in der Central-Rolle vorhanden. Dadurch wurde das ganze Projekt auf diesem Beispiel aufgebaut. Der Softdevice und die SDK legen einige Abstraktions-Layer auf die Hardware. Diese sind in Abbildung 10 visualisiert. Die wichtigsten Module des SDK werden im Kapitel 4.2 erklärt, falls weitere Informationen gewünscht sind, wird auf die offizielle Dokumentation verwiesen [8]. Wichtig für dieses Kapitel ist, dass der beschriebene Software-Zustand dem Soll-Zustand entspricht. Die Abweichungen sind im Kapitel Validierung genauer beschrieben.

Der eigentliche Programmaufbau ist eine StateMachine, welche so aufgebaut ist, dass alle Events möglichst kurz gehalten wurden. Jedoch hat dies relativ viele Flags zur Folge. Anschliessend werden die Events entsprechend ihrer Priorität verarbeitet. Die Prioritäten ergeben sich aus der Else IF im Wait State, genaueres dazu wird im Kapitel 4.1 beschrieben. Dadurch entsteht ein pollendes Programm mit Prioritäten, wie in Abbildung 11 dargestellt.

Das Programm wurde in verschiedene Module unterteilt, um die Leserlichkeit zu verbessern. Die Unterteilung wurde entsprechend der Funktionen gemacht. Im Hauptmodul (Main) sind nur die StateMachine und die Interrupthandler vorhanden. Folglich wurde ein Modul für die BLE-Funktionalitäten geschrieben und eines für die SD-Karte. Ein weiters Modul existiert für die Batterie-Funktionalitäten. Dieses ist jedoch nicht implementiert und enthält nur Dummy-Funktionen.



**Abbildung 10:** Aufbau der Software auf der Hardware



**Abbildung 11:** Konzept der pollenden Software

## 4.1 State-Machine

Nachdem mit dem Einleitungskapitel ein kurzer Überblick geschaffen wurde, kann nun der Hauptteil der Software genauer betrachtet werden. Der gesamte Ablauf basiert auf einer klassischen StateMachine, die aufgrund von unterschiedlichen Parametern in die entsprechenden nächsten States springt. Das hat den Vorteil, dass sich das Programm stets in einem definierten Zustand befindet und mittels entsprechenden Parametern jeweils den nächsten Arbeitsschritt vordefiniert. Die Abbildung 12 zeigt das Gesamtkonzept der State-Machine. Anschliessend werden die einzelnen States genauer definiert und beschrieben.

### State: Lookup

In diesem State verschafft sich das Programm über die entsprechende Initialisierung Zugriff auf die SD-Karte der Anwendung. Falls der Mikrocontroller nicht auf die SD-Karte zugreifen kann, wird eine Fehlermeldung ausgegeben und die Funktion wird beendet. Andernfalls wird dem Mikrocontroller signalisiert, dass der Zugriff geglückt ist und die eigentliche Funktion wird gestartet. Dazu werden die beiden Minor- und Majorzahlen in ein hexadezimales Zahlensystem gewandelt, welche dann als Vergleichskriterium verwendet werden. Falls die Nummer gefunden wird, kann das entsprechend zugehörige Audio-File über den Mikrocontroller ausgegeben werden. Verglichen wird jeweils zeilenweise, weshalb auch ein Fehlerhandling eingebaut wurde. Damit wird erkannt, ob sich das Textfile am Ende befindet. Somit lässt sich dann die Suche wiederholen, oder einen Fehler ausgeben. Die nachfolgende Abbildung 13 zeigt den detaillierten Funktionsablauf im Lookup-State. Wichtig ist an dieser Stelle die Erkenntnis, dass der Lookup-State die Verbindung zwischen Bluetooth und dem Audio-File repräsentiert.

### State: Print „Likes“

Print „Likes“ wurde nicht implementiert und das Hauptprogramm sprint in den Wait State. Das ist eine optionale Möglichkeit, die in einem weiteren Entwicklungskonzept bearbeitet werden kann. Die Software ermöglicht es aber diese Funktion noch einzubetten. Dabei könnte die

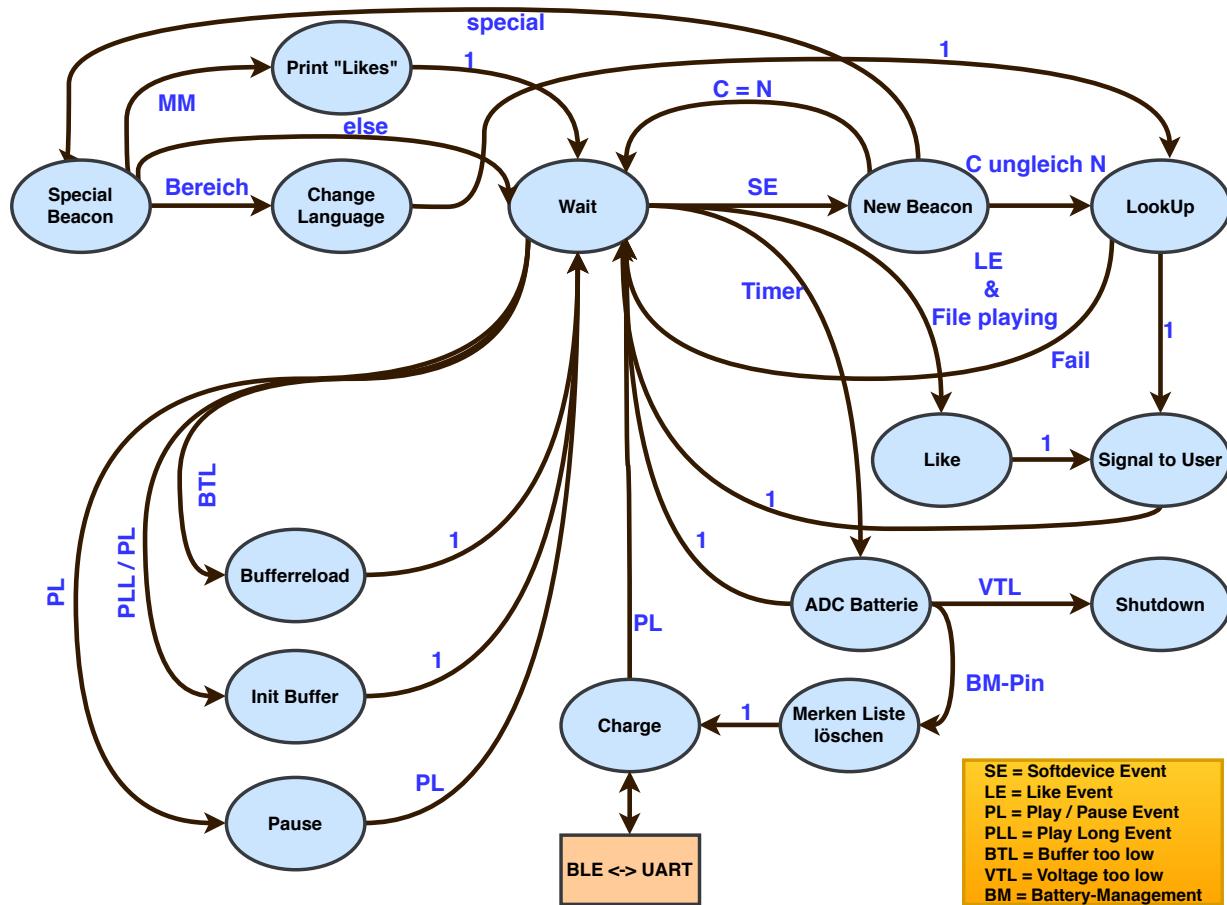
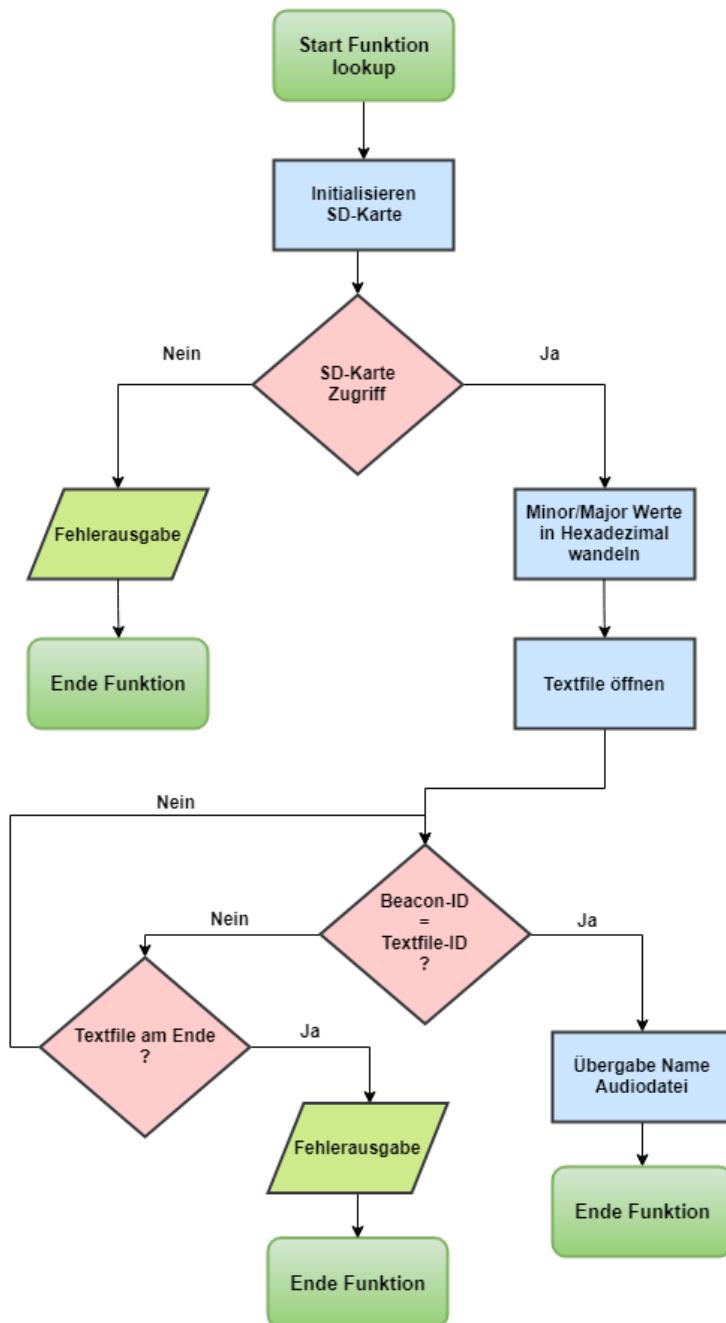


Abbildung 12: Statechart im Überblick mit den einzelnen States und den Parametern

Abbildung 13: Funktionsablauf im `lookup`-State

Broschüre mit den interessanten Objekten direkt gedruckt werden und bereits für den Besucher am Ausgang des Museums bereit liegen.

### State: Special Beacon

Special Beacon dient hauptsächlich zur Unterscheidung der verschiedenen Beacons für die Sprache, das Drucken und weitere Features die in einem weiteren Ansatz implementiert werden können. Aus diesem Grund wurde dieser State auch relativ einfach gehalten. Zuerst wird verglichen, ob es sich dabei um die Sprachkonfiguration handelt. Ist dies zutreffend, so springt das Programm in den Change Language State. Andernfalls wird überprüft, ob gerade der Like-Button gedrückt wird und entsprechend in den Print „Likes“ State gewechselt wird. Ist keine der beiden Zustände zutreffend, springt das Programm in den Wait State. Die nachfolgende Abbildung 14 zeigt den Ablauf der Funktion.

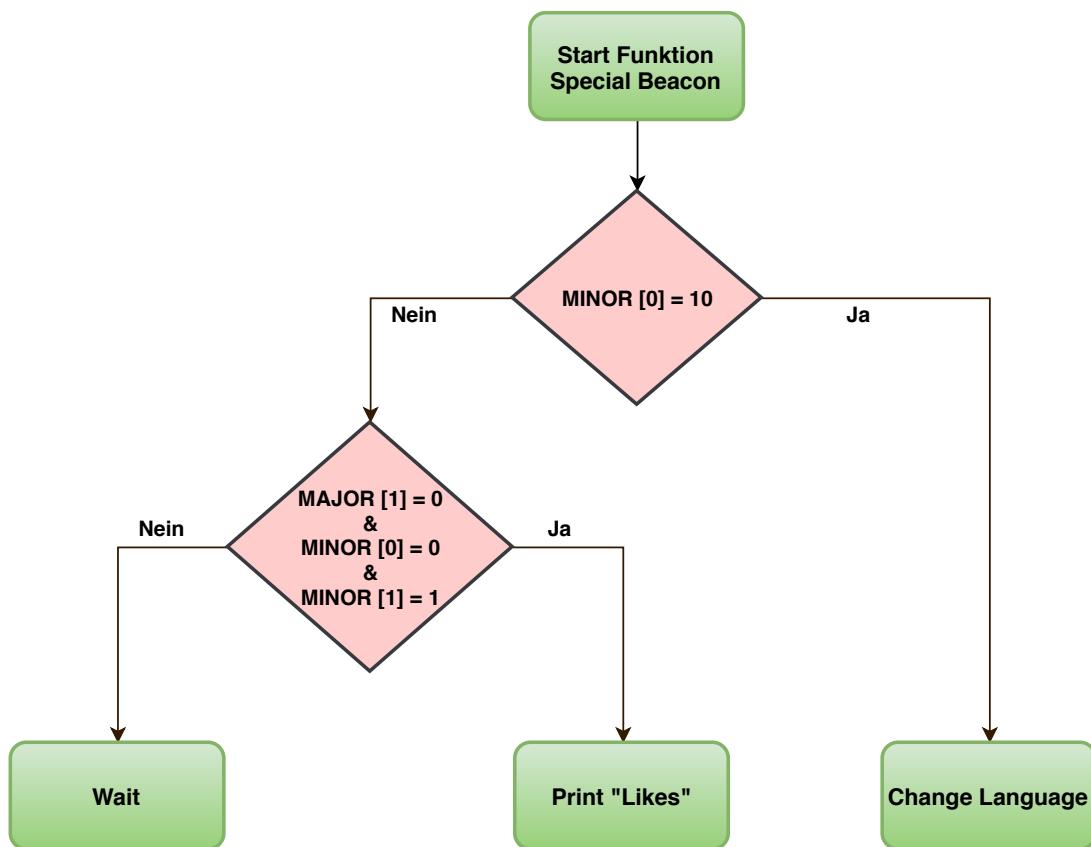


Abbildung 14: Funktionsablauf im Special Beacon State

### State: Pause

Dieser State pausiert das Sound-File und wartet auf ein weiteres Play-Event, um die Audiodatei wieder abzuspielen.

### State: Init Buffer und Bufferreload

Diese beiden States sind für das Initialisieren des Buffers und für den Start der PWM-Funktion verantwortlich. Für weitere Informationen dazu, wird auf das Kapitel 4.4 verwiesen.

**State: Merken Liste löschen**

Dieser State löscht die Likes, um für den nächsten User bereit zu sein. Anschliessend folgt der Charge State.

**State: Change Language**

Dieser State ist für die Sprachauswahl verantwortlich. Aufgrund des Zahlenwertes in Minor [1] wird zwischen den Landessprachen der Schweiz ausgewählt. Dabei wird die Vergleichstabelle in Form einer Datei kopiert und mit einem Kürzel entsprechend der Sprache versehen. Danach springt das Programm wieder in den lookup State.

**State: New Beacon**

Wird ein Beacon erkannt, dann wird der New Beacon State aufgerufen. Ist das aktuelle Beacon immer noch das gleiche wie das alte Beacon, springt das Programm in den Wait State. Wenn es sich nicht um das gleiche Beacon handelt, wird über den lookup State überprüft, ob eine entsprechende Übereinstimmung vorhanden ist.

**State: Like**

Wird in diesem State über einen Button ein Like-Ereignis ausgelöst, springt das Programm in den Signal to User State und speichert den entsprechenden Namen des Audio-Files auf die SD-Karte. Findet kein Ereignis statt, dann wird in den Wait State gesprungen.

**State: Signal to User**

Dieser State hat die Aufgabe, dem Benutzer mitzuteilen, dass ein neues Audio-File verfügbar ist, oder ob ein Kunstobjekt gespeichert wurde. Dabei blinkt eine dafür vorgesehene LED. Anschliessend springt die Software in den Wait State.

**State: Wait**

Dieser State ist die Ausgangslage des kompletten Ablaufs. Aufgrund von den in der Abbildung 17 ersichtlichen Bedingungen und sogenannten Flags, entscheidet das Hauptprogramm in welchen State zu switchen ist. Die Reihenfolge der Abfrage erfolgt nach dem Prioritätsprinzip.

**State: ADC Battery**

ADC-Battery wird zyklisch durch einen Softtimer im Wait-State aufgerufen, jedoch mit einer tiefen Priorität. Funktional wird hier die Batterie auf ihren Ladezustand überprüft und gemäss 18 entsprechend gehandelt. Wird die Batterie geladen, so wird die Like-Liste gelöscht. Andernfalls wird das Programm in den Wait-State wechseln, oder bei ungenügendem Ladezustand das System herunterfahren.

**State: Shutdown**

Ist der Ladezustand des Geräts zu tief, wird das System gezwungenermassen heruntergefahren. Somit sollen Schäden an der Hardware und an der Software vermieden werden.

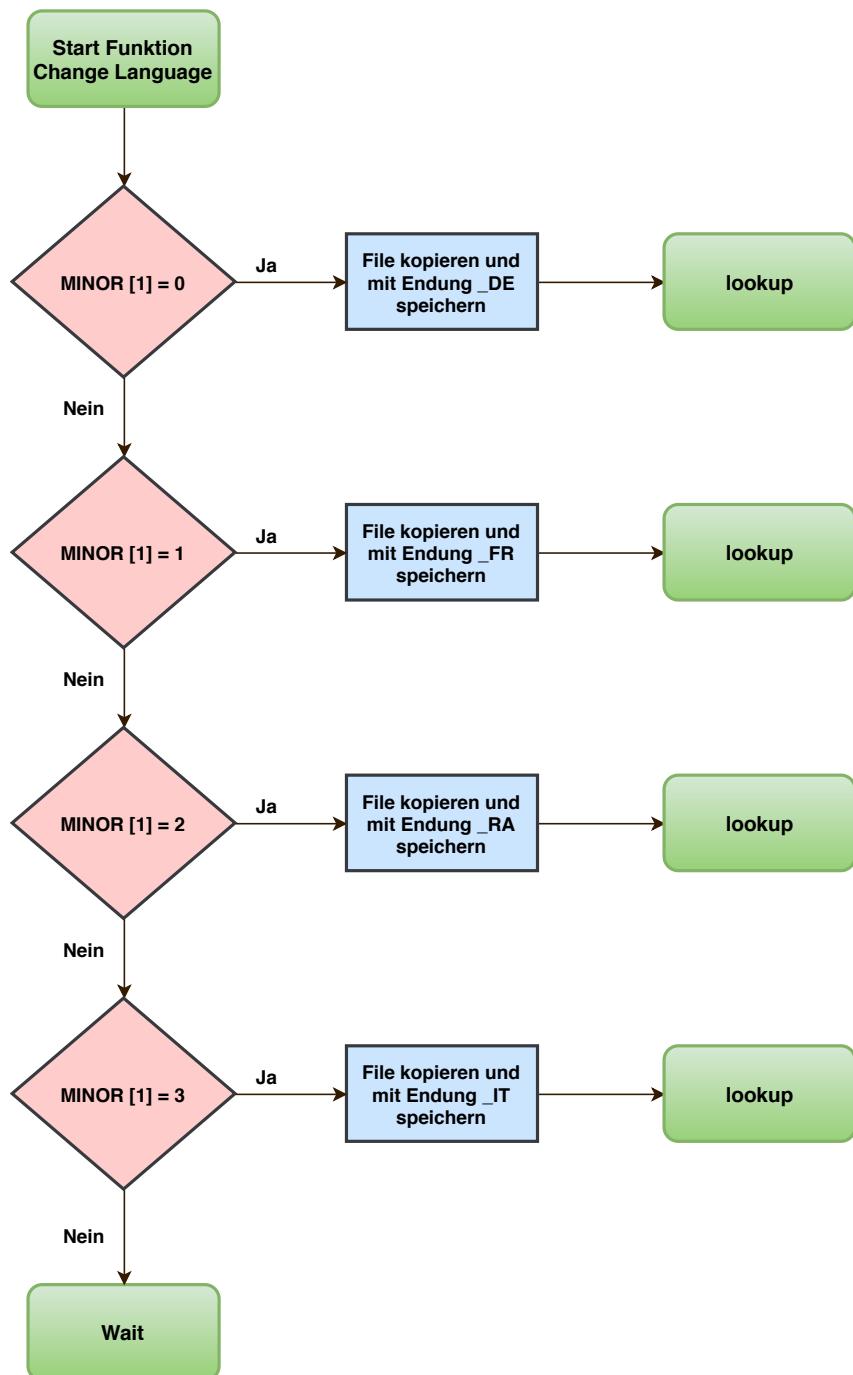
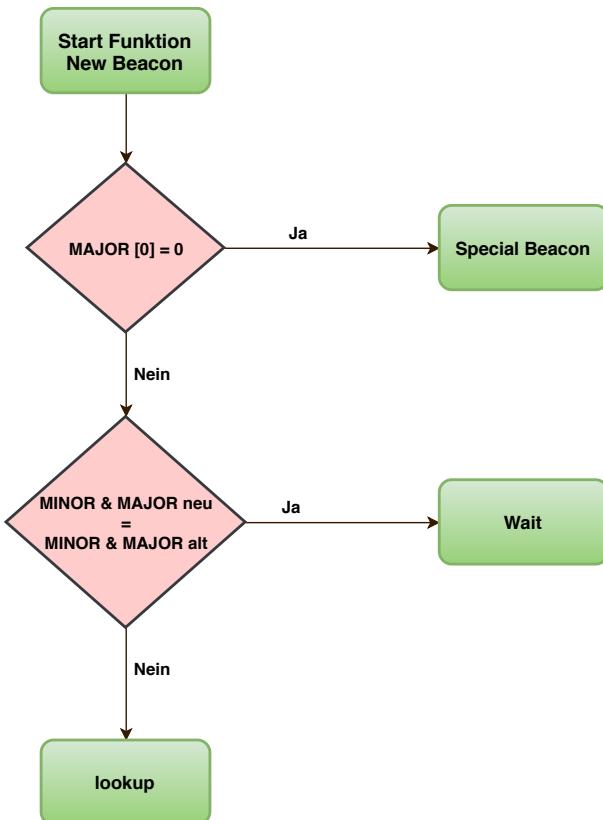


Abbildung 15: Funktionsablauf im Change Language State



**Abbildung 16:** Funktionsablauf im New Beacon State

### State: Charge

Hier wird überprüft, ob ein Pause- oder Playevent vorliegt. Falls ein Event vorliegt, springt die Software in den Wait-State und ist bereit für den nächsten User. Falls kein Event vorliegt, befindet sich das Programm in einer Schlaufe. Darin ist es möglich den Dojo mit der Ladestation zu verbinden und über UART Anpassungen auf der SD-Karte vorzunehmen.

## 4.2 Nordic Software Development Kit

Das Software Development Kit (SDK) ist eine Sammlung von nützlichem Code für die NRF-Chip Familie. Sie ist sehr umfangreich und ein absolutes Muss für unser Projekt. Hier wird jedoch nur auf ausgewählte Module innerhalb der SDK eingegangen, welche entscheidend sind für das Projekt. Für weitere Informationen wird auf die offizielle Dokumentation [8] Verwiesen.

### S132

Der Softdevice 132 ist ein Protokoll Stack nach Bluetooth 5. Er stellt alle Funktionen des Bluetooth-Low-Energy (BLE) Stacks zur Verfügung, was entscheidend war für das Projekt. Der Stack wird in der Central-Rolle verwendet. Jedoch ist der Softdevice nicht als Code erhältlich, es liegen nur die Binaries bei.

### Board Support Package (BSP)

Das Board Support Package ist eine abstraktions Schicht, die das Ziel hat die Hardware von der Software zu trennen und die Portierbarkeit zu erhöhen. Sie arbeitet wie schon der Bluetooth-

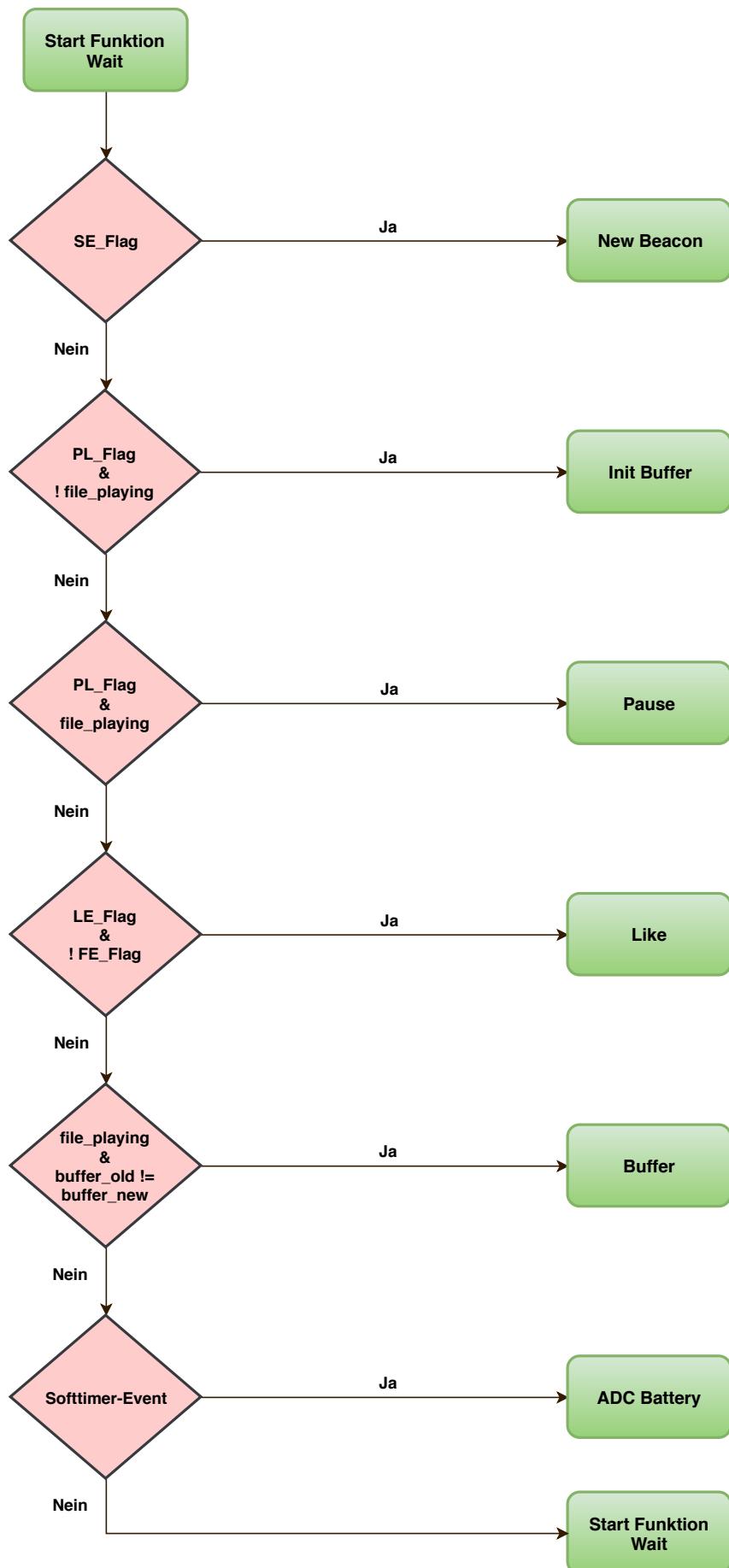


Abbildung 17: Funktionsablauf im Wait State

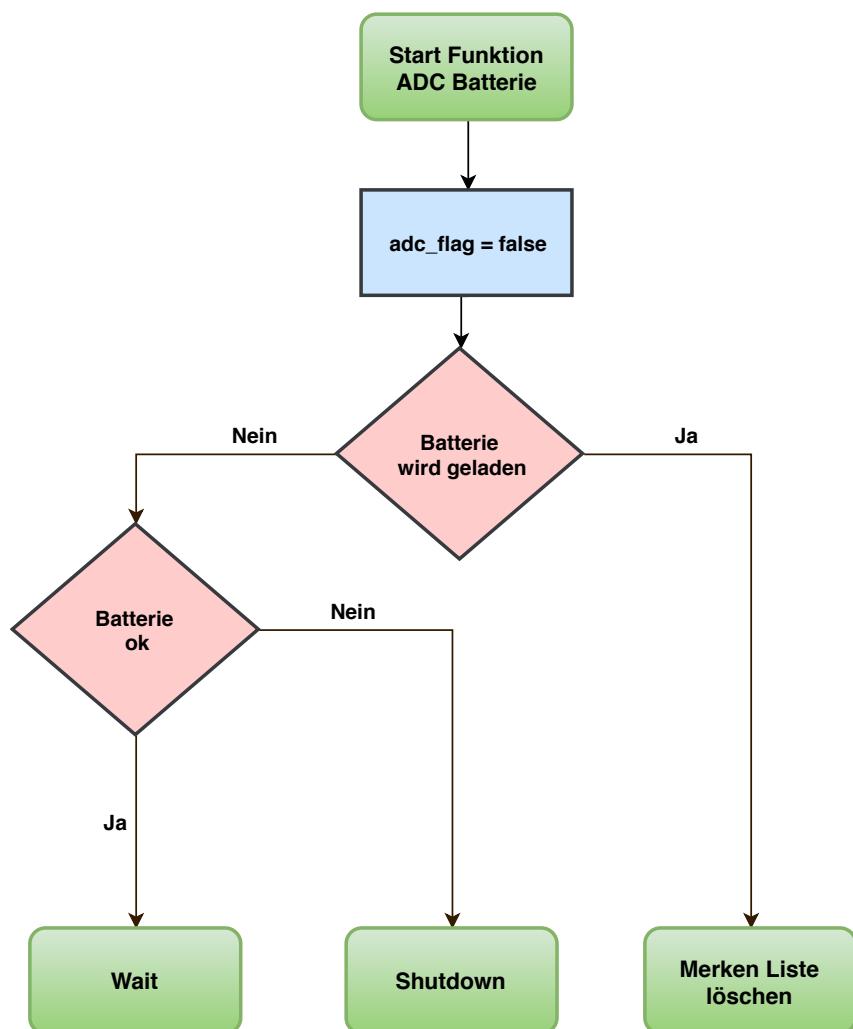
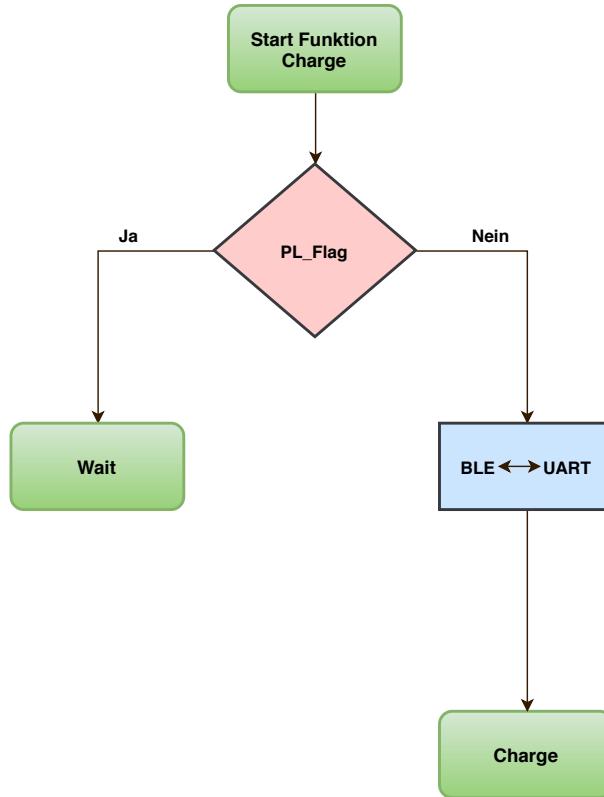


Abbildung 18: Funktionsablauf im ADC Battery State



**Abbildung 19:** Funktionsablauf im Charge State

Stack mit Callbacks. Sie bietet viele Funktionen für typische Hardware, zum Beispiel für Tasten. So kann für kurze und lange Betätigung der Taste ein verschiedenen Callback eingerichtet werden. Die Verlinkung zwischen der Hardware und den entsprechenden Defines findet in File boards.h statt. Dieses File muss angepasst werden für einen andern Print.

### NRF Log

Das Modul NRF Log ist ein mächtiges Debug-Werkzeug. Damit ist es möglich aus der laufenden Software auf eine Konsole zu schreiben. Diese kann frei gewählt werden. Es kann die Segger-Konsole verwendet werden, welche von Nordic empfohlen wird. Das Projektteam verwendete die Variante mit Putty. Weiter können auch Warnungs-Stufen festgelegt werden, was sehr hilfreich ist.

## 4.3 Bluetooth

### 4.3.1 Bluetooth Grundlagen

Standard-Bluetooth-Geräte senden in einem lizenzenfreien Band zwischen 2.402 und 2.480 GHz. Dabei können Störungen durch diverse andere Geräte auftreten, welche im selben Frequenzband arbeiten. Um eine Robustheit gegenüber Störungen zu erhalten, wird ein Frequenzsprungverfahren eingesetzt. Bei diesem Verfahren wird das Frequenzband in 79 Kanäle eingeteilt und bis zu 1600-mal in der Sekunde gewechselt [?].

Derzeitiger Bluetooth-Standard ist Bluetooth 5. Bluetooth 5 besitzt im Gegensatz zu seinen Vorgängern eine höhere Reichweite (bis zu 100 m statt 25 m) und eine schnellere Datenrate

(bis zu 2 Mbit/s statt 1 Mbit/s). Ausserdem enthält dieser Standard ebenso den im Standard 4 bereits eingeführten «Low Energy»-Modus, welcher den schon moderaten Energieverbrauch zusätzlich senkt, was für dieses Projekt benötigt wird [?]. Im Low-Energy-Modus wird für das Frequenzsprungverfahren das Frequenzband nicht in 79- sondern in 40 Kanäle unterteilt. Ausserdem wird auch bei der Datenrate Energie gespart, indem eine Geschwindigkeit von 1- statt 2 Mbit/s erreicht werden kann. Die typische Reichweite im Low-Energy-Modus beträgt 40 m, was die Mindestanforderung von 5 Metern für das Projekt überschreitet und deshalb genügt [?]. Damit über Bluetooth Low Energy (BLE) Verbindungen aufgebaut werden können, benötigt es sogenannte Profile. Bei Bluetooth sind dies das GAP (Generic Access Profile) und das GATT (Generic Attribute Profile). Auf diese wird in den nächsten Kapiteln näher eingegangen.

### 4.3.2 Generic Access Profile

Das GAP kontrolliert Verbindungen und Authentifizierungen. Es beschreibt wie Geräte miteinander kommunizieren. Dazu definiert es zwei verschiedene Rollen für die Geräte. Zum einen die Rolle des zentralen Geräts und zum anderen die Rolle des Peripheriegeräts [?]. Damit eine Verbindung zustande kommt, muss das Peripheriegerät in einem bestimmten Intervall ein Datenpaket das Payload genannt wird, senden. Empfängt ein zentrales Gerät das Payload, kann es für mehr Daten zusätzlich ein Antwortpaket (scan response payload) anfordern, welches daraufhin vom Peripheriegerät gesendet wird [?]. Meistens senden die Peripheriegeräte ihre Authentifizierung, damit eine Verbindung eingegangen und das GATT für mehr Datenaustausch benutzt werden kann. Während diese Verbindung besteht, kann das Peripheriegerät keine weiteren Verbindungen eingehen. Wird jedoch nur eine kleine Datenmenge benutzt, kann diese in das Payload integriert werden, womit alle zentralen Geräte in der Nähe Zugriff auf die Daten haben. Dieses Verfahren nennt man Broadcasting und ist für dieses Projekt von zentraler Bedeutung. Dennoch wird im nächsten Kapitel erläutert, wozu das GATT nützlich ist [?].

### 4.3.3 Generic Attribute Profile

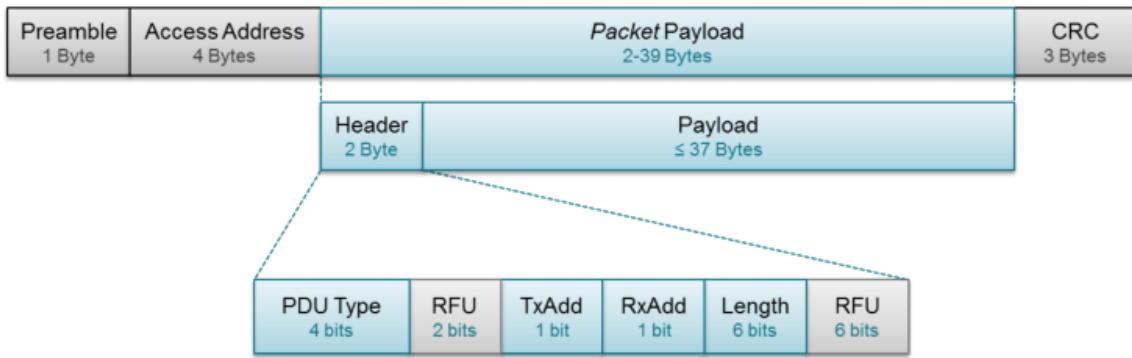
Das GATT definiert die Art und Weise, wie Peripherie und zentrales Gerät miteinander Daten austauschen. Dies macht es mit Hilfe des Attribute Protocol (ATT) [?]. Das ATT speichert Services, Characteristics und dazugehörende Daten. Services sind logische Sammlungen von zusammengehörenden Characteristics, wobei Characteristics als Datenpunkte angesehen werden können. Demzufolge besteht das ATT aus einer in Services geordneten Datensammlung [?].

Peripheriegeräte speichern das GATT und dienen somit als GATT-Server. Zentrale Geräte verbinden sich mit Hilfe des GAP mit dem GATT-Server und stellen eine Anfrage. Das zentrale Gerät wird somit zum GATT-Client. In einem Intervall werden vom Client gesendete Anfragen vom Server mit Datenpakete beantwortet. So können grössere Datenpakete ausgetauscht werden, jedoch kein Broadcasting stattfinden [?]. Das genannte zentrale Gerät bzw. GATT-Client ist der Dojo. Die genannten Peripheriegeräte sind BLE-Beacons, welche in der Nähe der Kunstobjekte angebracht sind. Im nächsten Kapitel wird näher auf die BLE-Beacons eingegangen [?].

### 4.3.4 BLE-Beacons

BLE-Beacons (Bluetooth Low Energy Beacons) sind kleine Peripheriegeräte, welche dazu da sind kleine Informationsmengen zu übertragen. Wird z.B. ein Temperaturverlauf über ein Jahr hinweg gemessen, so kommen BLE-Beacons zum Einsatz, da diese mit einer Knopfzellenbatterie über Jahre hinweg in Betrieb sein können und kleine Informationsmengen übertragen können [?].

Generell können BLE-Beacons vier Rollen einnehmen. Diese vier Rollen kann man unterteilen in zwei, welche Verbindungen eingehen können, und zwei weitere, welche nicht dazu in der Lage sind. Die Rollen, welche Verbindungen eingehen können, sind zum einen die eines Peripheriegeräts und zum anderen die eines zentralen Geräts. Die, welche keine Verbindungen eingehen können, sind die eines Senders und die eines Beobachters. Als Peripheriegerät fungiert das BLE-Beacon als Slave und wartet somit auf Input des zentralen Geräts (des Masters). Als zentrales Gerät fungiert das BLE-Beacon als Master und kann Verbindungen mit einem oder mehreren anderen Geräten eingehen. Als Sender (Broadcaster) kann das BLE-Beacon zwar keine Verbindungen eingehen, aber z.B. oben erwähnte gemessene Temperatur oder einen vordefinierten Wert senden. Wird das BLE-Beacon als Beobachter (Observer) benutzt, so kann dieser z.B. die von einem anderen BLE-Beacon gesendeten Werte empfangen und an einem angeschlossenen Display anzeigen, jedoch auch keine Verbindungen eingehen [?]. Die übermittelten Signale der BLE-Beacons werden gemäss Abbildung 20 formatiert [?].



**Abbildung 20:** PacketPayload Header

Die Preamble wird zur Synchronisierung und Zeitschätzung benötigt und ist für BLE-Beacons, welche in der Rolle eines Broadcasters sind, immer 0xAA. Auch die Access Address ist für Broadcaster immer gleich, nämlich 0x8E89BED6. Das Packet Payload beinhaltet Header und Payload. PDU bestimmt den Sendekanaltyp des sendenden Beacons, gemäss folgender Tabelle 1 [?]:

PDU Type	Packet Name	Description
0000	ADV_IND	Connectable undirected advertising event
0010	ADV_NONCONN_IND	Non-connectable undirected advertising event
0110	ADV_SCAN_IND	Scannable undirected advertising event

**Tabelle 1:** PDU Type

RFU (Reserved for Future Use) steht, wie der Name schon sagt, als Reserve für zukünftige weitere Implementationen und wird deshalb derzeit nicht gebraucht. TxAdd definiert, ob die Sendeadresse des Beacons öffentlich (TxAdd = 0) oder zufällig (TxAdd = 1) ist. Das RxAdd tangiert Beacons nicht und wird deshalb nicht weiter erwähnt. Das CRC (Cyclic Redundancy Check) dient der Erkennung von Fehlübertragungen. Das Payload ist gemäss Abbildung 21 aufgebaut [?].

Es ist zu sehen, dass es mit der Sendeadresse und den Sendedaten gefüllt ist. Die Sendeadresse kann öffentlich oder zufällig sein, wobei eine öffentliche Sendeadresse eine OUI (Organizationally Unique Identifier) nutzt, welche von der IEEE Registration Authority vergeben wird. Die Sendedaten können gemäss der Tabelle 2 formatiert werden [?].

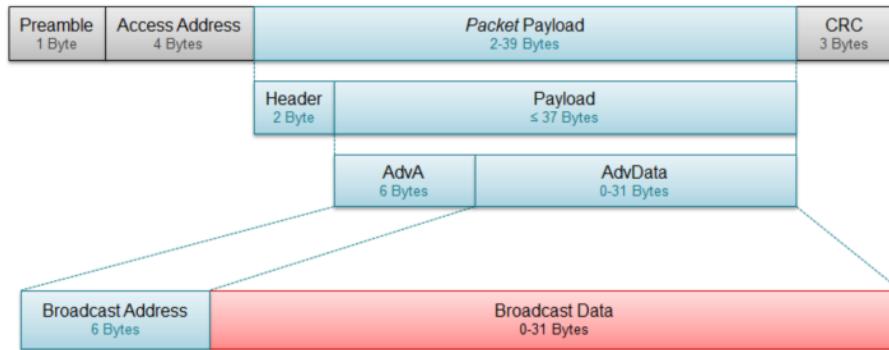


Abbildung 21: PacketPayload Payload

AD Data Type	Data Type Value	Description
Flags	0x01	Device discovery capabilities
Service UUID	0x02 - 0x07	Device GATT services
Local Name	0x08 - 0x09	Device name
TX Power Level	0x0A	Device output power
Manufacturer Specific Data	0xFF	User defined

Tabelle 2: AD Data Type

Die Flags definieren die Fähigkeiten des BLE-Beacons und sind gemäss der Tabelle 3 definiert [?].

Byte	Bit	Flag/Value	Description
0	•	0x02	Length of this data
1	•	0x01	GAP AD Type Flags
2	0	LE Limited Discoverable Mode	180 s advertising
•	1	LE General Discoverable Mode	Indefinite advertising time
•	2	BR/EDR Not Supported	•
•	3	Simultaneous LE and BR/EDR (Controller)	•
•	4	Simultaneous LE and BR/EDR (Host)	•
•	5-7	•	Reserved

Tabelle 3: Flags

Die Manufacturer Specific Data können gemäss Tabelle 2 vom Benutzer definiert werden. Hier können Werte gespeichert werden, die das BLE-Beacon senden soll, wie z.B. den RSSI-Wert. Dieser wird im nächsten Kapitel erläutert [?].

### 4.3.5 RSSI

RSSI steht für Received Signal Strength Indicator und ist ein Indikator für die Empfangsfeldstärke bei drahtloser Kommunikation. Je höher dieser Wert ist, desto stärker ist das empfangene Signal [?]. Die drahtlose Kommunikation erfolgt über das Senden von elektromagnetischen Wellen. Trifft eine solche Elektromagnetische Welle auf eine Antenne, so führt dies zu einer messbaren Selbstinduktion, welche dem RSSI-Wert entspricht. Dieser Wert muss abhängig von der

jeweiligen Anwendung interpretiert werden, da diverse Faktoren bezüglich Transmitter, Receiver und Übertragungsmedium den Wert beeinflussen können.

Im Falle des Museums mit gleichwertigen Beacons bedeutet dies, dass der Besucher sich eher in der Nähe des Beacons mit stärkerem RSSI Wert befindet. Dadurch kann das richtige Audiofile abgespielt werden. Zum Unterscheiden der Beacons werden UUID (Universal Unique Identifier), Major und Minor verwendet. UUID ist im Falle des Projekts eine gewählte Identifikationsnummer für das Museum. Major beschreibt eine eindeutig definierte Zimmernummer und Minor ist die definierte Nummer des Beacons im gleichen Raum. Im nächsten Kapitel wird die Minor- und die Majorvergabe beschrieben.

#### 4.3.6 Major Minor Vergabe

Um die Major Minor Nummern mit dem Dōjō zu benutzen, wurden sie im Verlaufe des Projektes standardisiert. Die Sprachauswahl funktioniert über solche Nummern. Beacons mit diesen speziellen Nummern lösen auf dem Dōjō entsprechende Funktionen aus. Abbildung 22 zeigt die Definitionen. Die Nummernräume sind so gestaltet, dass genug Platz für zusätzliche Funktionen vorhanden ist, wie zum Beispiel für weitere Sprachen oder Zugangskontrollen.



**Abbildung 22:** Default Definitionen von speziellen Major Minor Nummern.

In Abbildung 23 ist zu sehen, dass die erste Major Nummer 0x00 sein muss, damit ein spezieller Beacon erkannt wird. Das vereinfacht das Programm, da bei einem neuen Ble-Package nur der erste Eintrag im Array betrachtet werden muss, um die speziellen Beacons zu erkennen. Der Nummernraum für spezielle Beacons umfasst ca 16 Mio. Adressen, was für zusätzliche Funktionen reichen sollte. Somit bleiben ca 4.2 Mia. Adressen für Kunstwerke übrig.

### 4.4 SD-Karte

Um auf die SD-Karte zuzugreifen, wurde die Library fatfs verwendet. Sie ermöglicht einen einfachen Zugriff auf die Daten per SPI-Schnittstelle. Sie wird im ersten Kapitel beschrieben, anschliessend behandelt ein Kapitel die Bündelung des Codes in Zusammenhang mit der SD-Karte, weshalb ein eigenes Modul geschrieben wurde. Der letzte Teil beschreibt noch die benötigten Files auf der SD-Karte, um die Funktionen des Dōjō zu gewährleisten.

#### 4.4.1 FatFs

Für den Zugriff auf die SD-Karte wird ein Generic FAT Filesystem Module verwendet. In dieser Library sind Funktionen für die Initialisierung der SD-Karte und den Zugriff auf diese.

##### Lizenzen

Die FatFs Library untersteht einer 1-clause BSD Lizenz. Dies bedeutet sie darf verwendet werden, muss aber den Lizenztext beinhalten. Der Lizenztext ist im Anhang vorhanden.

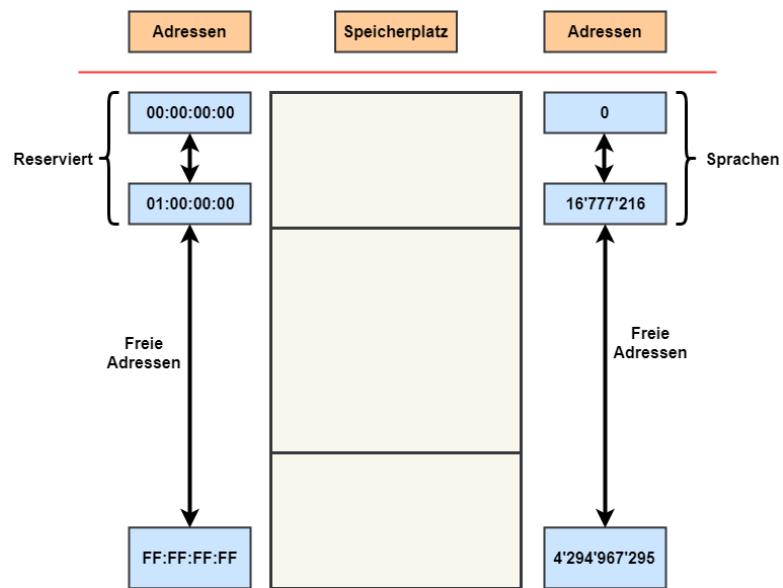


Abbildung 23: Definition der Majo Minor Vergabe

### **Spezifikationen FatFs**

Dateisystem Typ: FAT, FAT32(rev0.0) und exFAT(rev1.0)

Anzahl geöffneter Dateien: unlimitiert (hängt vom verfügbaren Speicher ab)

Anzahl Datenträger: bis zu 10

Datenträgergrösse: bis zu 2TB bei 512Bytes/Sektor

Dateigrösse: bis zu 4GB – 1 auf FAT-Volume und praktisch unbegrenzt auf exFAT-Volume

Clustergrösse: Bis zu 128 Sektoren auf FAT-Volume und bis zu 16 MB auf exFAT-Volume

Sektorgrösse: 512, 1024, 2048 und 4096 Bytes [9]

### **4.4.2 SD-Karte Modul**

Nachfolgend ist beschrieben welche Funktionalitäten im Zusammenhang mit dem SD-Karten Modul vorhanden sind.

#### **Merken**

Das Modul beinhaltet zwei Funktionen um die Funktionalität des Merken zu realisieren. Die Erste speichert das aktuelle Kunstwerk auf eine Liste. Auf dieser Liste werden die verschieden gemerkten Kunstwerken gesammelt. Die zweite Funktion löscht eben diese Liste wieder, um den Dōjō bereit für den nächsten Nutzer bereit zu machen. Den Dateinamen der Liste lässt sich im Header-File anpassen.

#### **SD-Karte Initialisieren**

Diese Funktion ruft eine Reihe fatfs-Befehle auf um die SD-Karte zu mounten. Diese wird über ein SPI-Bus an den Microkontroller angeschlossen. Die Pins für den Bus lassen sich im Header-File definieren.

### **4.4.3 Lookup**

Diese Funktion ist die eigentliche Implementierung des in Abbildung 13 beschrieben Algorithmus.

#### **next\_Value**

Die Funktion next Value dient dazu die nächsten Werte in den Buffer zu laden und ist wie folgt definiert:

```
static void next_value(int bufix)
```

Sie öffnet zuerst die entsprechende Audiodatei. Dies geschieht nur zu Beginn der Datei. Sie bleibt geöffnet bis die Audiodatei zu Ende ist, oder der Benutzer das Abspielen unterbricht. Ebenfalls wird der Lesezeiger beim ersten Aufruf der Funktion auf das 44 Byte geschoben werden, da die ersten 44 Bytes eines WAV Files keine Audiodaten enthalten. Die Funktion liest nun 4096 Bytes in einen Hilfsbuffer ein. Die Werte werden dann daraus skaliert und in die Sequenzen geladen. Um die eine invertierte Sequenz zu erzeugen, wird das 15 Bit auf 1 gesetzt.

#### **Sprachwechsel**

Diese Funktion wechselt die Sprache des Dōjō. Die Sprachauswahl geschieht über verschiedene Files. Es existiert für jede Sprache ein eigenes Lookup-File. Dieses verbindet die Majo-Minor-Kennzeichnungen mit den Wav-Files der jeweiligen Sprache. Dadurch muss für einen Sprachwechsel nur der Zeiger auf das Lookup-File geändert werden.

#### 4.4.4 Benötigte Files

Im Header-File des Moduls SD-Karte müssen verschiedene Files definiert werden. Diese müssen auch auf der SD-Karte vorhanden sein und im richtigen Format. Die Merken-Liste ist eine normale Textdatei (Endung .txt). Sie sollte leer sein. Des weiteren müssen die Lookup-Files als CSV (Comma separated Value) Dateien vorhanden sein. Sie sollten dem in Abbildung 24 definierten Format entsprechen. Zu beachten ist, dass die X Symbole für zwei stellige Hex-Zahlen stehen. Somit hat es eine Referenz auf ein Kunstwerk pro Zeile. Zu beachten ist, dass die Sprachkürzel nur zwei Zeichen beinhalten sollten.

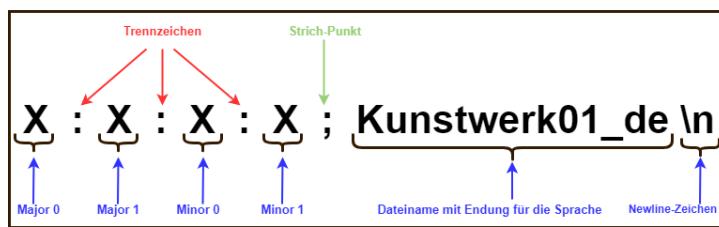


Abbildung 24: Formatdefinition Lookup-File

Die Audiodateien sind im Format WAV Unsigned 8-bit PCM auf der SD-Karte abzulegen. Dabei muss beachtet werden, dass die Sample Frequenz  $32kHz$  beträgt und es sich um eine Mono-Kanal Datei handelt.

## 4.5 Audio, PWM

Um ein Audiosignal abspielen zu können, werden zwei PWM-Signale benötigt wie in Abbildung 25 ersichtlich. Das eine Signal ist die invertierte Variante des anderen Signals. Das PWM-Signal wird mit dem NRF52 generiert. Für den NRF52 stellt NORDIC SEMICONDUCTOR die PWM HAL and driver Bibliothek zur Verfügung. Die genutzten Funktionen sind in der Tabelle ?? aufgeführt. Diese ermöglicht es, ein PWM-Signal zu erzeugen. Der NRF52 bietet vier PWM Instanzen mit je vier Kanälen. Für die Audioausgabe wurden zwei PWM Instanzen mit je einem Kanal genutzt.

Beschreibung	Funktion	Argumente
PWM initialisiern	nrf_drv_pwm_init	nrf_drv_pwm_t const *const p_instance nrf_drv_pwm_config_t const *p_config nrf_drv_pwm_handler_t handler
Audio abspielen	nrf_drv_pwm_complex_playback	nrf_drv_pwm_t const *const p_instance nrf_pwm_sequence_t const *p_sequence_0 nrf_pwm_sequence_t const *p_sequence_1 uint16_t playback_count uint32_t flags

Tabelle 4: Funktionen der PWM HAL and driver Bibliothek

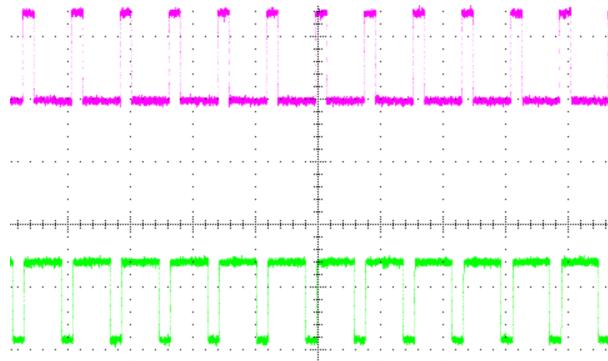


Abbildung 25: PWM-Ausgang des NRF52. Das violette Signal ist das invertierte grüne Signal.

In der Abbildung 26 ist der Ablauf des PWM-Unterprogramm aufgezeigt. Als erstes findet die Initialisierung statt. Nach der Initialisierung können die beiden Sequenzen 0 und 1 generiert werden. In diesen Sequenzen sind die Daten des Audio-Files abgelegt. Die Funktion complex\_playback generiert am Ausgang anhand dieser Sequenzen das PWM-Signal.

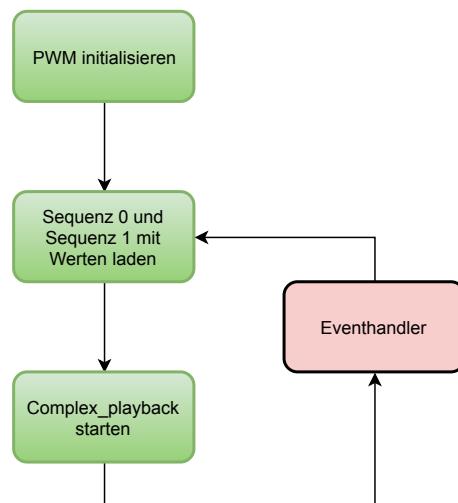


Abbildung 26: Ablauf PWM

## PWM Initialisieren

Bei der Initialisierung wurde die PWM Instanz, die Config und der Eventhandler mitgegeben. Die Config des PWM musste anhand der Angaben des Wavefiles generiert werden. Die Werte wurden wie in der Tabelle 5 gewählt.

« « < HEAD

Parameter	Gewählter Wert	Bemerkung
output_pins	Pin	
irq_priority	APP_IRQ_PRIORITY_LOWEST	Wurde aus dem Beispiel übernommen
base_clock	NRF_PWM_CLK_16MHz	Es wurde der höchste clock Frequenz gewählt
count_mode	NRF_PWM_MODE_UP	Wurde aus dem Beispiel übernommen
top_value	500	Siehe Berechnung top Value
load_mode	NRF_PWM_LOAD_COMMON	Wurde gewählt, da alle PWM Kanäle den selben Wert ausgeben
stepp_mode	NRF_PWM_STEP_AUTO	Wurde aus dem Beispiel übernommen

**Tabelle 5:** Gewählte Config Werte

### Berechnung top Value

Da die clock Frequenz des PWM Moduls auf  $16MHz$  gesetzt wurde, beträgt der top value für eine Audiodatei mit Abtastfrequenz von  $32kHz$  500. Dies berechnet sich wie folgt: ======  
Da die clock Frequenz (oben \*clock) des PWM Moduls auf  $16MHz$  gesetzt wurde, beträgt der top value für eine Audiodatei mit Abtastfrequenz von  $32kHz$  500. Dies berechnet sich wie folgt:  
»»»> bf25fa5cc30d7cbbed361af39edc0146dab9a6b3

$$32kHz = \frac{16MHz}{top\ value} \Rightarrow top\ value = \frac{16MHz}{32kHz} = 500 \quad (4.1)$$

### Sequenzen laden

Mit der Funktion next\_Value werden neue Werte in die Sequenzen geladen. Diese Funktion ist im Abschnitt ?? genauer beschrieben.

### Complex playback

Die Funktion generiert anhand der Sequenz ein PWM-Signal. Der Vorteil zu simple playback ist, dass zwei Sequenzen mitgegeben werden können. Wenn die Sequenz 0 fertig abgespielt wurde, startet automatisch die zweite Sequenz und der Eventhandler wird ausgelöst. In dieser Zeit kann die erste Sequenz wieder neu beladen werden. Die Beladung der Sequenzen wird mit dem Eventhandler dieser Funktion gesteuert.

## 4.6 Lizenzen

Die Lizenztexte sind alle in Englisch veröffentlicht. Dadurch kann das Projektteam nicht haftbar gemacht werden für unwissentlich gemachte Übersetzungsfehler.

### SDK

Nordic Semiconductor hat an Anfang eines jeden Files ihren Lizenztext [10] hinterlegt. In diesem ist Beschrieben was mit dem Code gemacht werden darf und was nicht. Einige wichtige Punkte:

- Am Anfang jedes Files muss der Lizenztext stehen.
- Weiterverteilung muss unter dem Selben Copyright erfolgen wie bisher.
- Mann soll nicht ungefragt Werbung machen für Nordic und Co.
- Die Software darf nur für NRF-Chips verwendet werden.

Auch werden alle Verantwortungen für Support, Sicherheit usw. abgestritten. Damit sind sie nicht Haftbar für Schäden, die die Software anrichtet.

Der S132 ist speziell, weil dieser nicht in Source-Form vorliegt. Er ist nur in Binary-Form enthalten. Er hat einen leicht anderen Lizenztext welcher in der SDK neben den Biarys gefunden werden kann. Er besagt zusätzlich, dass der Sourcecode der Softdevice nicht reverse engineered, decompiliert, modifiziert und disassembliert werden darf.

## 5 Validierung

Bei der Validierung werden nachfolgend die Testkonzepte der Software und Hardware erläutert. Die Hardware beinhaltet hierbei hauptsächlich das Testing der Batterie wie auch der Induktions-Ladeschaltung. Das Testkonzept der Software beinhaltet das Testing von .....

### 5.1 Testkonzept Hardware

Damit ein reibungsloser Betrieb möglich ist, müssen die einzelnen Hardware Komponenten auf Herz und Nieren geprüft werden. Nachfolgend werden die Testverfahren genauer beschrieben und die Testergebnisse aufgelistet.

#### Schutzmechanismen Batterie

Die Batterie weist einige Schutzmechanismen auf, welche alle getestet werden müssen. Als erstes wurde der Tiefentladungsschutz geprüft. Um dies zu testen wurde ein Widerstand der Dimension  $9\Omega$  angeschlossen, was gemäss Berechnung 5.1 einen Entladestrom von rund 411mA zur Folge hatte.

$$I_{discharge} = \frac{U}{R} = \frac{3.7V}{9\Omega} = 411mA \quad (5.1)$$

Während dem Entladevorgang wurde stets die Spannung überwacht, wobei die Spannung von 3.7V auf bis 2.5V absank. Nachdem die 2.5V Schwellenspannung unterschritten wurde, brach der integrierte Batterieschutz die Spannungsversorgung ab. Die Widerstände wurden abgehängt und der gesamte Vorgang wurde kurze Zeit danach mit dem selben Endergebnis wiederholt. Als nächstes wurde ein Kurzschlusstest durchgeführt, wobei hier der Schwellenstrom gemäss Datenblatt bei 4.8A liegt. Gemäss dem  $U = R \cdot I$  Gesetz, wurde ein Widerstand der Grösse von  $700m\Omega$  verwendet, um den Grenzwert zu überschreiten. Auch bei diesem Versuch riegelte das PCM den hohen Entladungsstrom ab und schaltete die Versorgungsspannung der Batterie ab.

#### Ladeschaltung der Batterie

Für die Ladeschaltung der Batterie wurde wie im Kapitel ?? bereits erwähnt ein Lade-IC verwendet. Dieser reguliert zuerst die Spannung wobei nach Erreichung des Schwellenwertes von 4.2V der Strom auf 0A herunter reguliert wird. Dieser Vorgang wurde während einem gesamten Ladevorgang der Batterie beobachtet und dokumentiert und ist in Abbildung 27 ersichtlich. Hierbei ist wichtig zu erwähnen, dass dieses Testing nicht die induktive Energieübertragung verwendete, sondern der Fokus auf der Funktionalität des Lade-ICs beschränkt und somit das Netzgerät „Power Supply“ der Firma „K. Witmer“ als Spannungsspeisung verwendet wurde. Aus diesem Grund ist auch ein Strom von 400mA wie auch eine Ladezeit von lediglich rund 270 Minuten ( $\cong 4.5h$ ) ersichtlich was die effektiven Ladewerte mittels induktiver Ladung deutlich unterbietet. Ersichtlich ist nachfolgend die Regulierung der Spannung (blaue Kurve) wie auch die Regulierung des Stromes (rote Kurve).

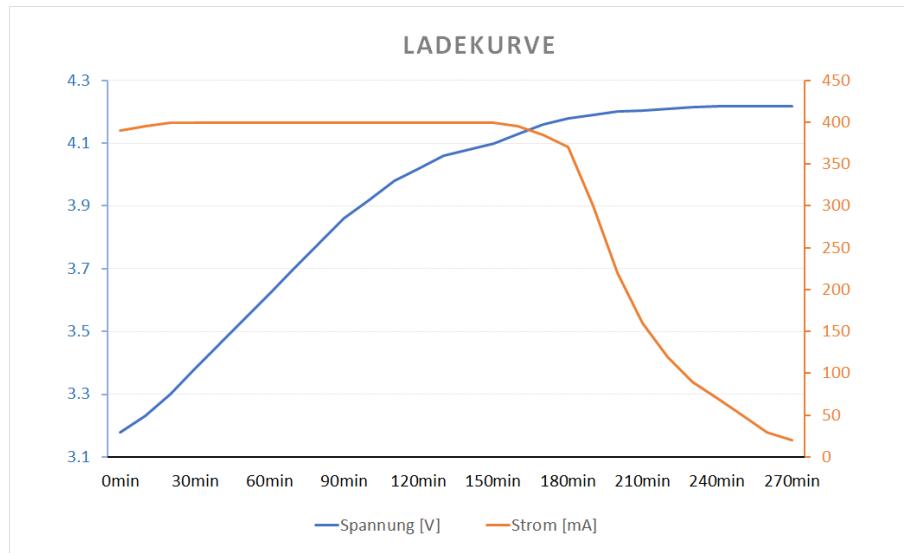


Abbildung 27: Ladekurve Emmerich LI14500

Die oben genannten Vorgänge der Spannungs- und Stromregulierung sind in dieser Grafik gut ersichtlich, wobei der Ladevorgang nach dem erreichen von rund 20mA als fertig betrachtet wurde.

### Induktive Ladeschaltung

In diesem Abschnitt werden die Ergebnisse der induktiven Ladeschaltung präsentiert. Nachfolgend zeigt Abbildung 28 die Abhängigkeit zwischen induziertem Strom und Spannung in Abhängigkeit zur Distanz z zwischen den Induktionsspulen.

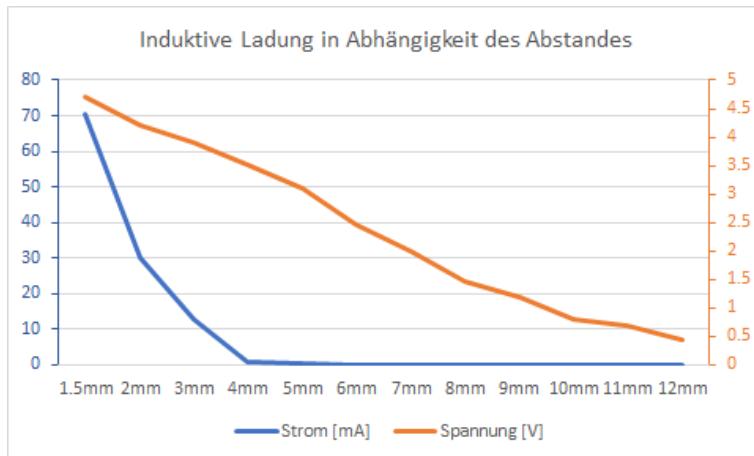


Abbildung 28: Induzierter Strom in Abhängigkeit der Distanz

In der Abbildung ist gut sichtbar, dass die Spannung fast linear zur Distanz abnimmt, hingegen der Ladestrom der Batterie extrem schnell klein wird. Aufgrund dieser Erkenntnissen sind wir gezwungen eine möglichst kurze Entfernung zwischen den Spulen einzuhalten weshalb wir eine Ladestation entworfen haben, welche die bestmöglichen Induktionswerte garantiert. Der minimale Abstand welcher beim Ladezyklus erreicht werden kann beträgt rund 1.5mm. Dieser Abstand entspricht ziemlich genau der Wanddicke des Dōjōs. Bei diesem Abstand resultiert ein

an die Batterie gelieferter Strom von maximal  $70mA$ , wobei die Ladezeit direkt von diesem Strom abhängt. Der Ladezyklus wird in zwei Etappen unterteilt. Bei der ersten Etappe wird die Batterie mit konstantem Strom geladen. Dies hat zur Folge, dass die Spannung von ihrem Minimalwert  $3.3V$  auf den Schwellenwert von  $4.2V$  reguliert wird. Um auf die notwendige Ladezeit des ersten Ladezyklus zu kommen, kann die Zeit während einer beliebig grossen Spannungsdifferenz gestoppt werden. Generell gilt: Umso grösser die Spannungsdifferenz, desto genauer die Approximation. Die Endzeit der Spannungsregelung kann linear hochgerechnet werden. Da jedoch der Ladezyklus nach vollständiger Spannungsregelung noch nicht abgeschlossen ist, folgt noch die benötigte Zeit für die Stromregelung. Da diese nicht einfach berechnet werden kann, wurde dieser Prozess im Labor durchgeführt und gestoppt. Diese zwei Ladezyklen zusammen gerechnet ergeben die gesamte Ladezeit. Für den Ladezyklus wurden drei unterschiedlich hohe Ströme definiert, welche die Zeit für den Ladezyklus vorgeben. Sie beinhaltet einen schonenden Ladezyklus ( $I_{charge} = 30mA$ ), einen normalen Ladezyklus ( $I_2 = 50mA$ ) und einen schnellen Ladezyklus ( $I_3 = 70mA$ ). Die Erreichung dieser drei Ladestufen wird in dieser Arbeit nicht weiter definiert, könnte jedoch durch drei unterschiedliche Spannungsstufen an die Primärspule reguliert werden. Nachfolgend sind die Berechnungen für die jeweiligen Ladezeiten  $t_{voltage}$  und  $t_{current}$  der drei Stufen, wie auch die gesamte Ladezeit  $t_{tot}$  ersichtlich.

#### Ladezyklus schonend

$$t_{voltage} = \frac{1.1V}{\left(\frac{0.006V}{10\text{Minuten}}\right)} = 1'833.3\text{ Minuten} = 30h 33min \quad (5.2)$$

$$t_{current} = 2h 7min \quad (5.3)$$

Die gesamte Ladezeit ergibt sich durch Addition der Berechnungen 5.2 und 5.3.

$$t_{tot} = t_{slow,voltage} + t_{current} = 32h 27min \sim 32.5h \quad (5.4)$$

#### Ladezyklus normal

$$t_{voltage} = \frac{1.1V}{\left(\frac{0.0016V}{10\text{Minuten}}\right)} = 687.5\text{ Minuten} = 11h 28min \quad (5.5)$$

$$t_{current} = 46min \quad (5.6)$$

Die gesamte Ladezeit des normalen Ladezyklus ist nachfolgend in 5.7 ersichtlich.

$$t_{tot} = t_{voltage} + t_{current} = 12h46min \sim 13h \quad (5.7)$$

#### Ladezyklus schnell

$$t_{voltage} = \frac{1.1V}{\left(\frac{0.0265V}{10\text{Minuten}}\right)} = 415.1\text{ Minuten} = 6h 55min \quad (5.8)$$

$$t_{current} = 1h\ 18min \quad (5.9)$$

Die gesamte Ladezeit gemäss Berechnung 5.10 ergibt sich durch Addition der Berechnungen 5.8 und 5.9.

$$t_{tot} = t_{voltage} + t_{current} = 7h\ 41min \sim 8h \quad (5.10)$$

Augenfällig ist die kurze Ladezeit von lediglich rund 8 Stunden beim schnellen Ladezyklus (Berechnung 5.10). Da sich jedoch bei einem Strom von  $70mA$  die Spule mehr erwärmt als beim Ladezyklus mit  $50mA$ , wird vorgeschlagen, dass bei genügend Zeit der normale Zyklus mit  $50mA$  gestartet wird. Dieser ist zwar um rund 5 Stunden langsamer, jedoch nachhaltiger für die eingebauten Materialien.

### **Inbetriebnahme der induktiven Ladeschaltung**

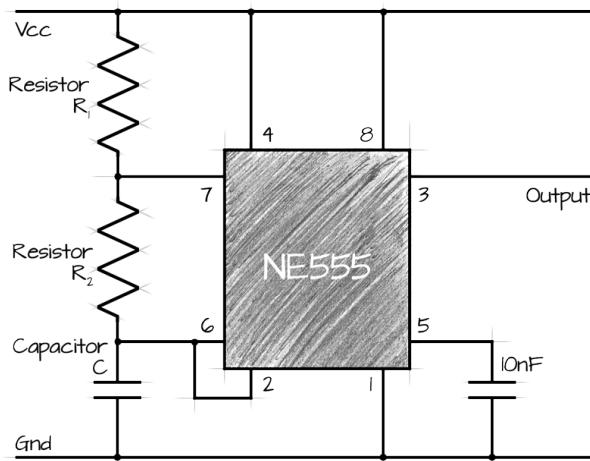
Der erste Prototyp der Pulsschaltung für die Induktion wurde mithilfe eines NE555 realisiert. Dabei wurden als Widerstände Potentiometer eingebaut, mit welchen die Resonanzfrequenz des LC-Gliedes gesucht werden sollte.

Der allererste Versuch ergab eine Übertragung von  $5mA$  bei einer Taktfrequenz von  $10kHz$ . Es stellte sich relativ schnell heraus, dass die Taktfrequenz viel zu tief war. Das konnte durch die Veränderung der Taktfrequenz der Pulsschaltung festgestellt werden. Bei höherer Taktfrequenz wurde die Übertragung besser.

Da die Taktfrequenz der NE555-Prototypenschaltung aufgrund der Wahl des Potentiometers auf  $10kHz$  begrenzt war, wurde nun in weiteren Schritten mit dem Frequenzgenerator der 2n3055 Leistungstransistor angesteuert. Des Weiteren wurde das C aus dem LC-Glied ausgebaut, um so das spezifische Maximum der Spule zu finden.

So wurde nun mit konstanter Eingangsspannung und variabler Taktfrequenz die Spule getestet. Mit höher werdender Frequenz wurde die Übertragung besser. Um diese noch weiter zu verbessern empfiehlt sich bei der Taktfrequenz einen Duty Cycle von möglichst 50% zu erzielen.

Es erwies sich, dass bei der Taktfrequenz von  $93kHz$  die beste Übertragung zustande kam. Bei höheren Frequenzen war diese wieder rückläufig. Da nun die optimale Frequenz gefunden und die Induktivität der Spule bekannt war ( $14.9\mu H$ ), konnte der ideale Kopplungskondensator berechnet werden. Dies ergab, unter Berücksichtigung der E-Reihen, einen  $220nF$  Kondensator als ideal. Damit die Spule nicht zu heiss werden und der Isolierlack nicht zu schmelzen beginnt, wurde der Strombegrenzungswiderstand des 2n2222 mit  $2\Omega$  definiert. Die Berechnung hierzu ist im oberen Text zu finden. Somit sind alle benötigten Bauteile bekannt.



**Abbildung 29:** Verwendete Timerschaltung NE555 als Pulsquelle

$$C = 1nF$$

$$R_1 = 200\Omega$$

$$R_2 = 9k\Omega$$

Mit diesen Werten konnte nach dem Gleichrichter, bei einer Übertragungsdistanz von einer Standard-Platinen dicke, eine Leerlaufspannung von ca. 17V und ein Kurzschlussstrom ca. 300mA gemessen werden. Es gilt jedoch zu beachten, dass diese Werte ohne Last gemessen wurden.

Da nun die Induktionsstufe funktioniert wird diese nun um die Ladeschaltung erweitert. Hier ergab sich folgendes Problem:

Die Eingangsspannung des Lade-IC's darf nicht mehr als 6.5 Volt betragen. Da die Receiverspule sich im inneren des Dōjōs befindet und dadurch nur eine sehr geringe Baugröße besitzen darf, kann diese zwangsläufig keine grossen Leistungen erbringen. Dies bedeutet, dass akzeptabel Ladeströme nur mit genügend hohen Spannungen erzeugt werden können.

Um dies zu erzielen wurde versucht, einen Spannungsregler einzubauen, um den übertragenen Strom bei zu halten, jedoch die Spannung zu begrenzen. Das Ergebnis davon war jedoch ziemlich bescheiden. Der so entstehende Energieverlust ist ziemlich gross und ausserdem kam es vereinzelt vor, dass der Lade-IC trotzdem kaputt ging, da der Spannungsregler nicht sauber regelte.

Die Ideale Lösung wäre natürlich einen geeigneteren Lade-IC zu nehmen. Da jedoch die Validierung desjenigen bereits durchgeführt wurde und nicht genügend Zeit vorhanden war diesen erneut zu bestellen wurde sich entschieden die Transceiver Schaltung ein wenig anzupassen.

Dabei gibt es folgende Möglichkeiten. Man kann die Eingangsspannung absenken. Dies beeinflusst jedoch auch den übertragenen Strom exponentiell, da einerseits eingangsseitig mit weniger Spannung auch weniger Strom durch die Spule fließt und andererseits auch der Leistungstransistor nicht sauber durchgesteuert wird da dieser ebenfalls an der gleichen Versorgungsspannung hängt und somit sein Pulssignal beeinflusst. Die zweite Möglichkeit ist die Taktfrequenz verringern. Dadurch wird die Induzierte Spannung ebenfalls abgesenkt was ebenfalls den Strom exponentiell beeinflusst. Die dritte Möglichkeit wäre die Übertragungsdistanz zu verringern. Da aber sichergestellt werden wollte, dass der Dōjō-Boden dennoch eine gewisse Stabilität aufweisen sollte wurde diese Distanz der Platinendicke beihalten.

Somit wurde die Schaltung lange ausgetestet um die beste Kombination zu erzielen. Das beste Resultat ergab die Kombination zwischen einer Taktfrequenz von ca 80kHz und einer Eingangs Spannung von 8.3V. Dies lässt sich damit erklären das leichtes Absenken der beiden Möglichkeiten den Strom nicht so stark beeinflusst. Des Weiteren wurde der Strombegrenzungswiderstand auf  $1.5\Omega$  verringert da sich die Eingangs Spannung verringert hat und ausserdem der Stromfluss durch die Spule damit erhöht wurde.

Diese Anpassungen fielen ebenfalls zu Gunsten des Energiemanagements, jedoch ist dies eine stabile Variante für die ersten Prototypen. Durch die Anpassungen wird die überschüssige Energie in Form von Wärme an dem Transceiver Spule abgegeben. Die Test ergaben, dass so die Batterie bei einer Spannung von 4.7V mit 70mA geladen wird. Es ist ebenfalls möglich kurzzeitig mit einem höheren Strom zu laden, wenn die Eingangs Spannung nur geringfügig erhöht wird. (bis zu 120mA). Jedoch ist die Hitzeentwicklung hierbei so gross, dass nach einer gewissen Zeit die Übertragung zusammenbricht. Deshalb ist zu empfehlen die angegebenen Werte nicht zu überschreiten um eine dauernde Funktionalität zu gewährleisten.

Die Werte der oben beschriebenen angepassten Pulsschaltung sind folgende:

C:  $1nF$

R1:  $200 \Omega$

R2:  $9 k\Omega$

## LC-Filter

Die Validierung des LC-Filters wurde mittels eines  $500Hz$  Sinus Signals erledigt. Dabei wurde sowohl ohne Filter wie auch mit Filter eine KO Bild aufgenommen. Abbildung ?? zeigt das Signal ohne Filter. Rot angezeigt ist die FFT des Signals. In Abbildung ?? ist das Signal und die FFT davon mit dem LC-Filter.

Durch die beiden KO Aufnahmen wurde ersichtlich, dass durch den Filter ein deutlich besseres und weniger verrauchtes Signal rekonstruiert wird. Somit ist der LC-Filter notwendig, da mit ihm die Schaltfrequenzen des PWM Ausgangs unterdrückt werden.

## 5.2 Testkonzept Software

Erklärt, welcher Teil wie getestet wird und führt die Ergebnisse mit Auswertung auf.

### 5.2.1 PWM Signal

Das PWM Signal wurde mit dem Gleichen Signal wie das LC-Filter validiert. Dabei wurden sowohl die beiden PWM Kanäle mit dem KO aufgezeichnet wie in ?? zu sehen ist, sowie das gefilterte Signal.

Aus Figure 32 ist ersichtlich, dass die beidem PWM Signale zwar invertiert zu einander sind aber auch Zeitverschoben. Die Zeitverschiebung beträgt knapp  $4\mu s$ . Dies ergibt keinen hörbaren Effekt und ist somit zu vernachlässigen.

Um die Einstellungen des PWM-Moduls aus 4.5 zu validieren wurde wieder ein  $500Hz$  Sinus Signal mit dem KO abgebildet.

In Figure 33 ist ersichtlich, dass die Frequenz des Signals nun nur  $127Hz$  anstatt  $500Hz$  beträgt. Dies entspricht einem Faktor von ungefähr 4. Um die  $500Hz$  zu erreichen wurde nun der top value aus Kapitel 4.5 von 500 auf 125 gesetzt. Dies entspricht dem Faktor 4. In Figure 34 ist ersichtlich, dass nun das Sinus Signal mit  $500Hz$  schwingt.

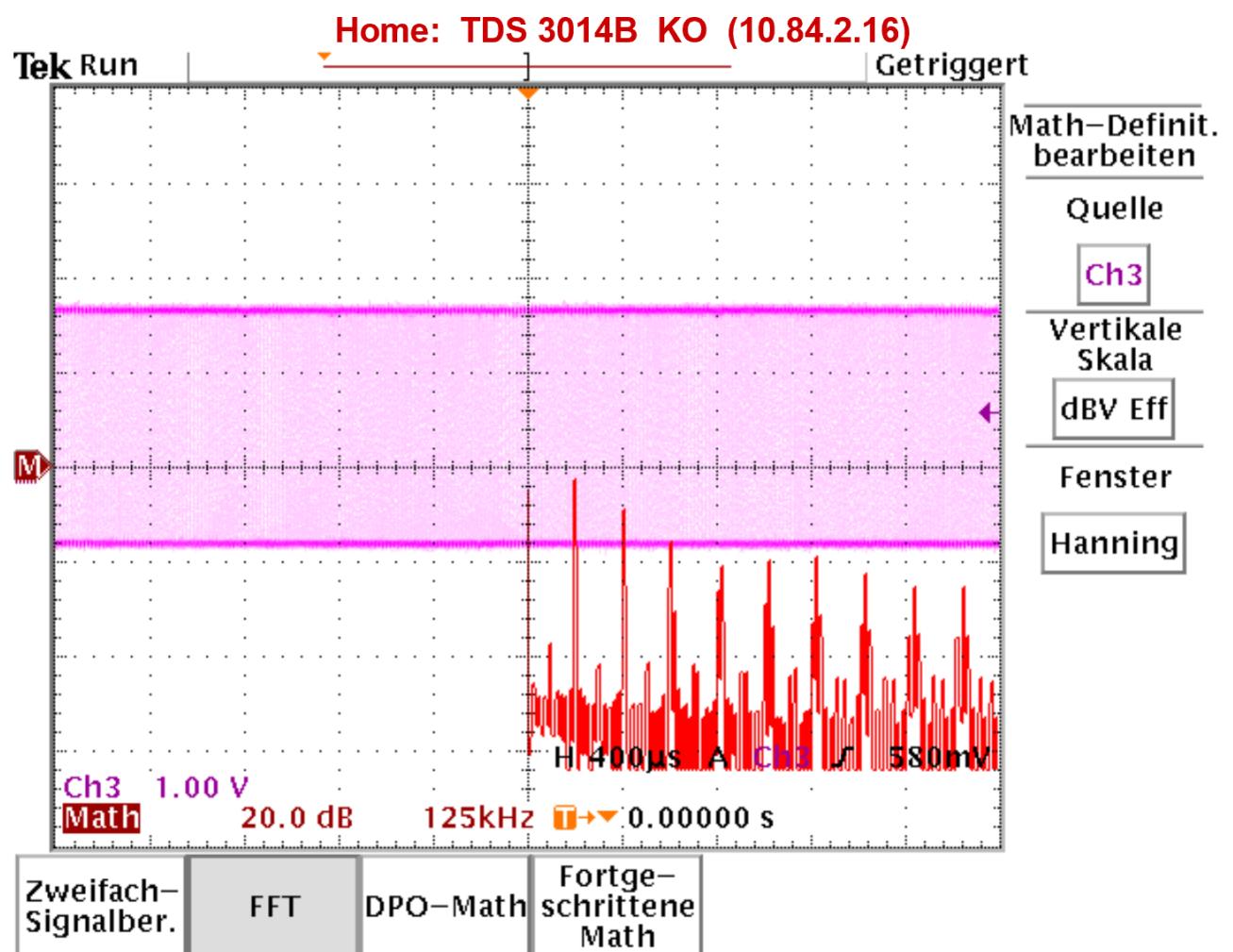


Abbildung 30: 500Hz Sinus Signal und FFT ohne LC-Filter

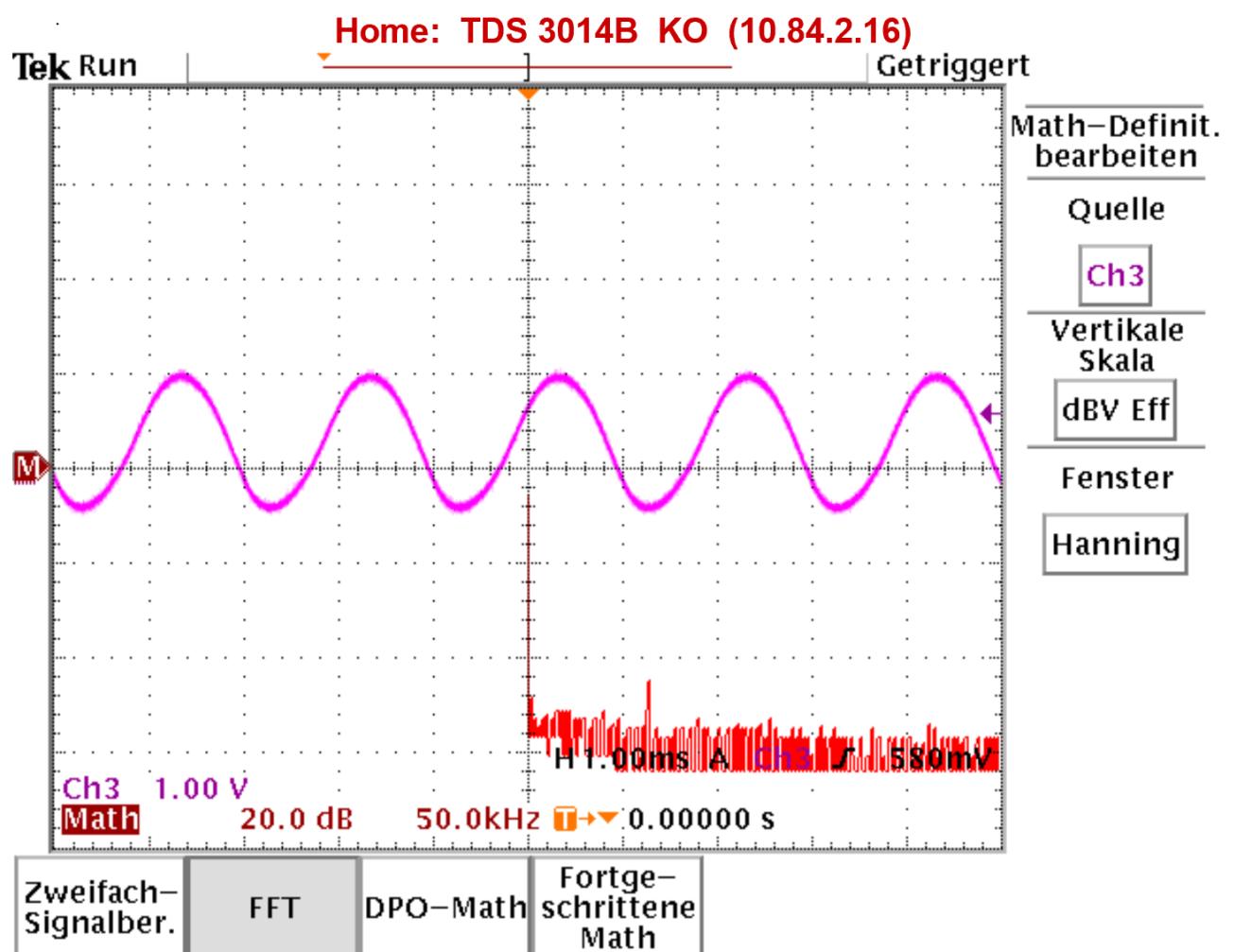


Abbildung 31: 500Hz Sinus Signal und FFT mit LC-Filter

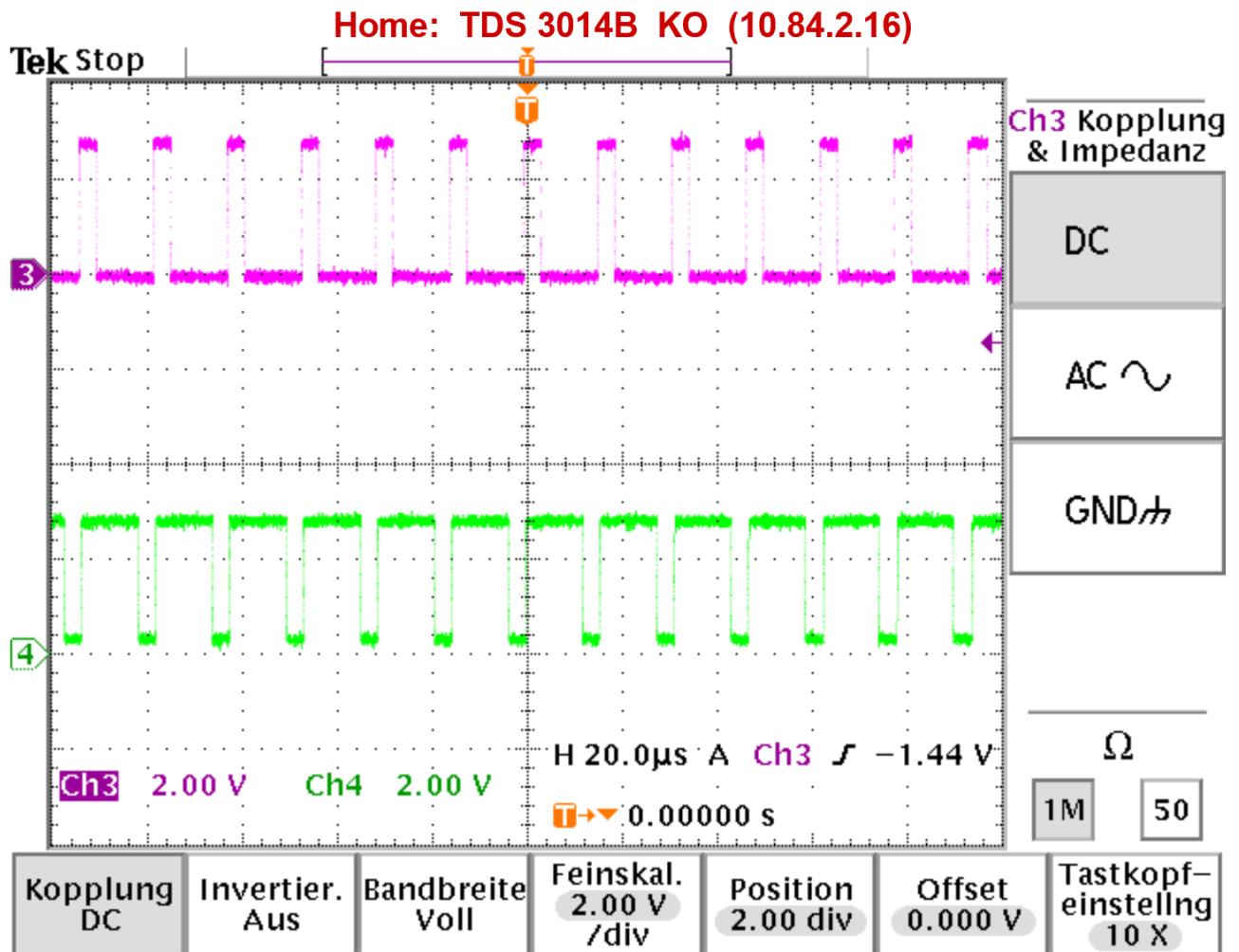


Abbildung 32: PWM Signal beider PWM Ausgänge

Da die Berechnungen einen top value von 500 ergeben, deutet dies auf einen Fehler im Code, in der Audiodatei oder einen Überlegungsfehler hin. Fälschlicherweise wurden diese Tests mit einer stereo Audiodatei durchgeführt. Da nur ein Kanal angesteuert wird, ergibt sich ein Faktor 2 unterschied zu den Berechnungen. Dieser Test wurde folglich noch einmal mit einer mono Audiodatei durchgeführt. Dabei ergab sich das der top value, wie erwartet, noch um Faktor 2 vom berechneten Wert abweicht.

### 5.2.2 Statemachine

Die Statemachine wurde auf ihr Verhalten hin getestet. Es wurden alle States ein mal erzwungen. Dies geschah mithilfe der Flags. Dabei wurde sichtbar, dass folgende States nicht korrekt funktionieren.

- ADC-Batterie
- Shutdown
- Merken Liste Löschen
- Charge

Anzumerken ist, dass diese States in der Statemaciene funktionieren. Aber ihre Aufgabe nicht erfüllen können, da sie aus zeitlichen Gründen nicht implementiert sind.

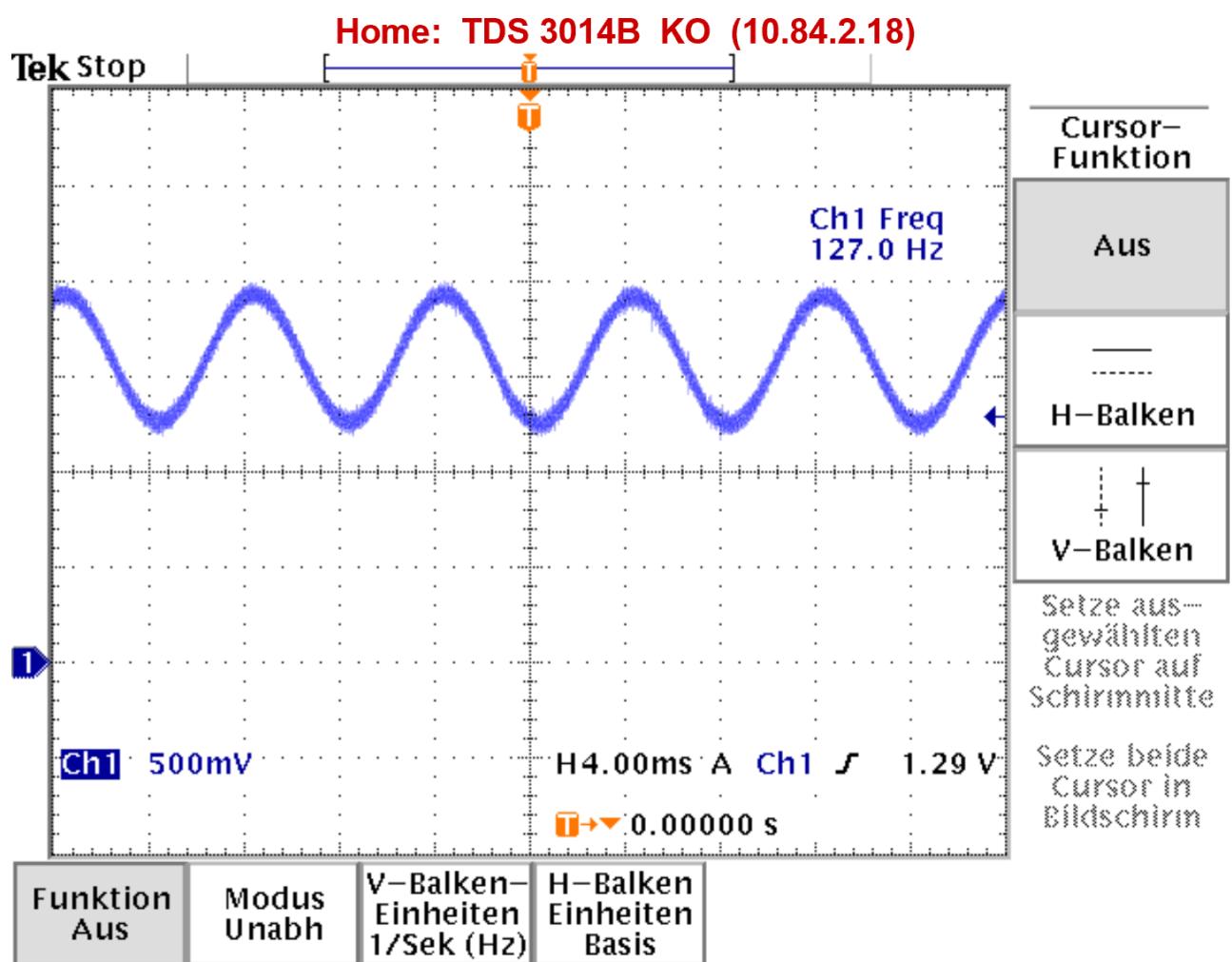


Abbildung 33: PWM auf 32kHz eingestellt

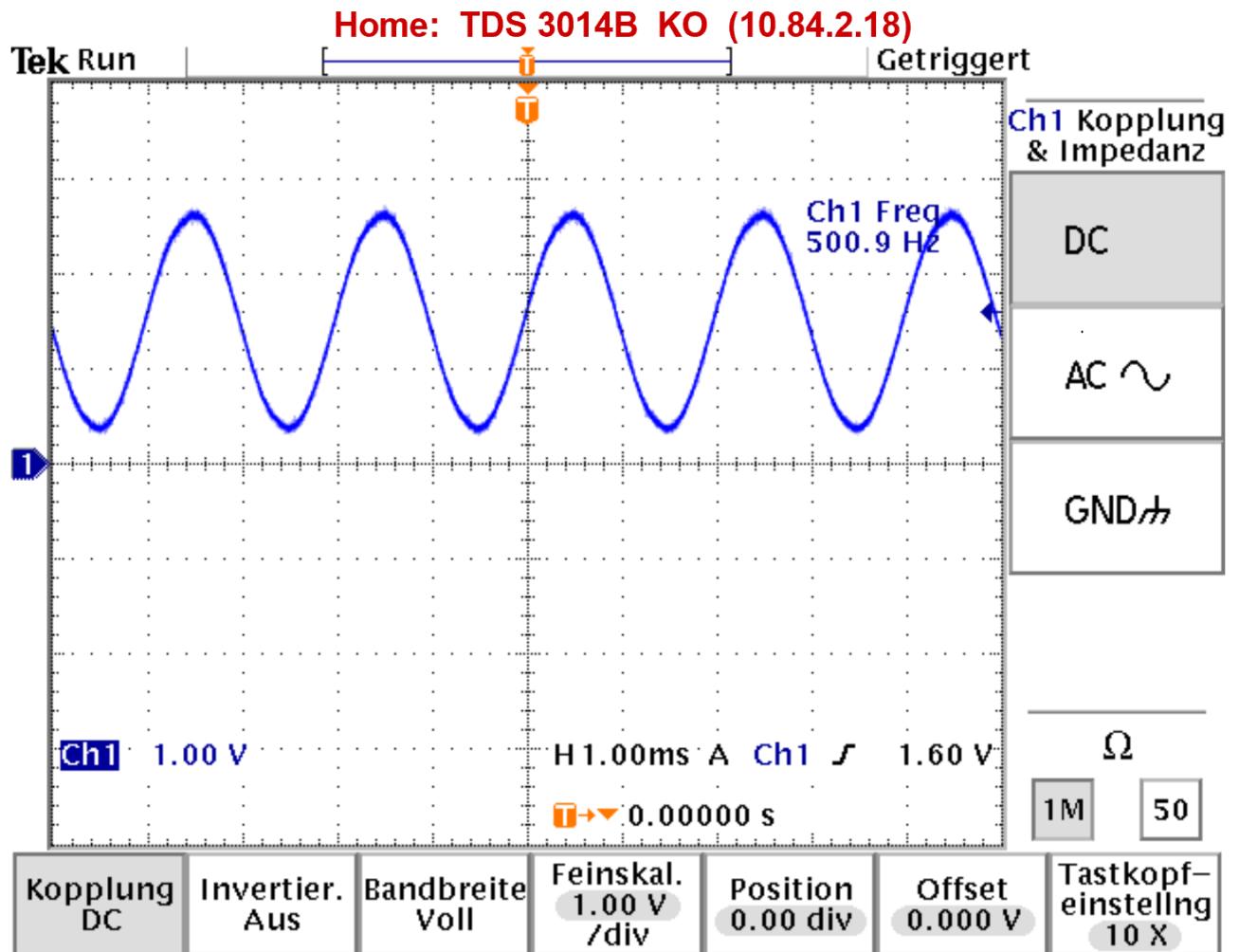


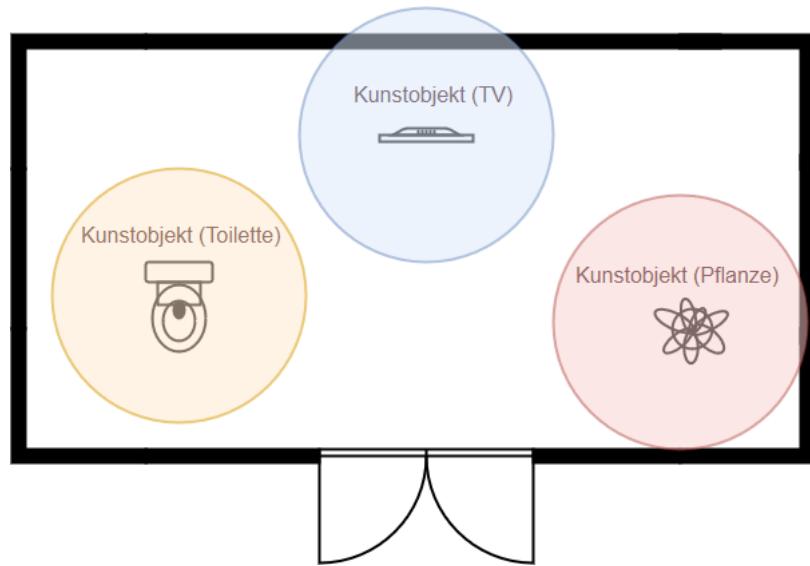
Abbildung 34: PWM auf 128kHz eingestellt

### 5.2.3 Bluetooth

Das Bluetooth Modul wurde mit verschiedenen Beacon's getestet. Dabei ist aufgefallen, dass wenn mehrere Beacons mit verschiedenen Sendeintervallen senden, dies zu grösseren Problemen führt. Dann dominieren die schnelleren Sendenden Beacons die langsameren. Darum ist es zu empfehlen, dass sich die Beacons nicht näher aneinander befinden als ihre Referenz Felder wie in Abbildung 35.

### 5.2.4 SD-Karte

Dieses Modul hat erfüllt alle Anforderungen. Jedoch gewisse nur knapp. Die Funktion des Merken hat einige Makel. Es ist nicht möglich ein gemerktes Kunstobjekt wieder zu löschen und es ist möglich ein Kunstobjekt mehrmals zu merken.



**Abbildung 35:** Überlappung der Referenz-Felder

## 6 Schlusswort

Dieses Kapitel besteht noch aus einzelnen Argumenten und Abschnitten und ist noch nicht miteinander verknüpft

**a) Was läuft? Was wurde im Projekt erreicht? Welche in der Aufgabenstellung gestellten Anforderungen wurden realisiert? Wo wurden diese übertroffen?**

Dōjō wurde ursprünglich in einem kleinen hohlen Stab konzeptioniert. In einer ersten Entwicklungsphase wurde ein Prototyp realisiert, welcher optisch an und für sich nichts mit dem Dōjō zu tun hat. Das hat den Vorteil, dass sich einzelne Komponenten wie z.B. eine Ladeschaltung für die Energiespeicherung, Mikrocontroller Evaluation Board oder der Knochenschallaktor als einzelne Teilsystem einfach zu einem Gesamtsystem zusammensetzen lassen. Dadurch lassen sich die Arbeiten voneinander entkoppeln und Messungen können mit deutlich weniger Aufwand durchgeführt werden. Das Evaluation Board des Mikrocontrollers bringt den grössten Vorteil mit sich. Steckplätze für Verbindungen und weitere Komponenten sind bereits vorhanden und können ohne weiteres genutzt werden. Ebenfalls kann dadurch die Programmierung der einzelnen Teilbereiche durch das Verwenden mehrerer Evaluation Boards unabhängig voneinander durchgeführt und getestet werden. Zusätzlich wurde dabei darauf geachtet, dass sich die einzelnen Komponenten auf einem kleinen PCB implementieren lassen, das sich dann wiederum in den Dōjō einbauen lässt.

Die Erkennung der Bluetooth-Beacons funktioniert einwandfrei, solange nicht mehr als 10 Beacons in unmittelbarer Nähe sind. Das Detektieren erfolgt über die Software. Mit einem geeigneten Algorithmus wird jeweils das Beacon mit dem stärksten Signal als das „korrekte“ Kunstobjekt identifiziert. Das integrierte Bluetooth-Modul im Mikrocontroller ermöglicht das Empfangen der entsprechenden Beacon-ID und kann diese über eine geeignete Funktion in eine Hexadezimal-Zahl wandeln. Anschliessend wird mit einer ID-Liste auf der SD-Karte verglichen und bei Übereinstimmung das entsprechende Audio-File abgespielt.

Die Energieversorgung wird über eine Li-Ion Batterie im Boden des Dōjō garantiert. Der Ladezyklus erfolgt mittels induktiver Ladung, welches eine mühsame Ladevorbereitung des Gerätes verhindert. Der gesamte Ladevorgang wird gestartet sobald Ein kleine Einbuchtung am Bo-

den des Dōjō ermöglicht das einfache aufstecken und laden. Aus diesem Grund wurde auf eine Implementierung einer USB-Connection verzichtet.

Die induktive Ladung funktioniert einwandfrei und ermöglicht eine Stromübertragung von rund 70mA. Für nachfolgende Weiterentwicklungen der Ladestufe empfiehlt es sich, einen Lade-IC zu verwenden, welcher eine grössere Eingangsspannung verarbeiten kann. Der LT1512 weist eine Eingangsspannung zwischen 2.7V – 25V auf. Dies hat zur Folge dass die Grösse des Bauteils ( $L \times B \times H$ ) beträgt  $(6.2 \times 5 \times 1.7)mm$ . Trotz seiner grossen Bauform überwiegt der Vorteil mit der hohen Versorgungsspannung.

Desweiteren würde es sich lohnen die Spulen zu optimieren. Die eingebauten Flachspulen sind zwar platzsparen, kommen jedoch aufgrund ihrer Bauform ziemlich schnell an ihr Leistungsmaximum. Hier würden sich solche Spulen sich empfehlen, wie sie z.B. auch in modernen elektrischen Zahnbürsten verwendet werden. Hierbei handelt es sich um zwei unterschiedlich grosse Ringspulen, von welchem die kleinere Tranceiver-Spule in der Ladeschaltung in die grössere Receiverspule im Dojo gesteckt wird. Diese Ringspulen wären zwar ein wenig grösser, doch durch die grössere Bauform könnte die Ladeschaltung mit grösseren Strömen betrieben und damit auch der Dojo noch schneller geladen werden.

#### Like-Button

Mit der Verwendung des komplexen und leistungsfähigen Mikrocontrollers NRF52832 können alle Prozesse zentral gesteuert werden und dadurch entfallen weitere Controller. Ebenfalls kann dadurch der Energieverbrauch gesenkt werden und die Verwendung eines zusätzlichen Bluetooth-Moduls entfällt.

**b) Was läuft nicht? Wo besteht Verbesserungsbedarf? Was konnte nicht realisiert werden? Hier ist eine kurze Angabe der Gründe sinnvoll.**

Mit dem Verzicht einer USB-Schnittstelle besteht keine Möglichkeit die SD-Karte ohne weiteres zu aktualisieren oder den Inhalt im Allgemeinen zu verändern. Daher sieht das Konzept vor, die SD-Karte manuell zu entnehmen und dann über einen extra dafür vorgesehenen SD-Karten-Hub zu aktualisieren. Dadurch lassen sich mehrere SD-Karten gleichzeitig bearbeiten und führt damit zu einer signifikanten Zeiteinsparung für die Betreiber. Weiter wird die SD-Karte im Gehäuse auch so angebracht, dass eine einfache Entnahme ohne Werkzeug oder Ähnliches möglich ist. Theoretisch wäre es möglich gewesen eine USB-Schnittstelle zu implementieren. Allerdings muss man sich dann auch unmittelbar die Frage stellen ob es dann noch Sinn macht, den Dōjō induktiv zu laden. Beide Eigenschaften sind unmittelbar miteinander verknüpft. Die Entscheidung fiel dann auf die induktive Ladung der Anwendung.

**c) Optimierungs- / Weiterentwicklungsmöglichkeiten: Welche Optimierungsmöglichkeiten bestehen? Was könnte wie besser gemacht werden? Was muss bei einer Weiterentwicklung bedacht werden?**

Eine Weiterentwicklung des aktuellen Konzepts ist durchaus sinnvoll. Nachdem die Basis des Dōjō umgesetzt und getestet wurde, ist viel technisches „Know-How“ vorhanden. Einerseits kann eine Aktualisierung der SD-Karte über ein drahtloses Netzwerk wie z.B. Bluetooth oder W-LAN umgesetzt werden. Dadurch liesse sich jedglicher Kontakt mit der SD-Karte durch einen Mitarbeiter vermeiden. Fraglich ist an dieser Stelle jedoch in welchem Umfang Aktualisierungen vorgenommen werden, da die drahtlose Datenübertragung nicht unbegrenzt Daten übermitteln kann.

Ein weitere Möglichkeit ist die Umsetzung der Sprachwahl. Konzeptionell wurde diese im Verlaufe des Berichts erläutert aber nicht umgesetzt. Durch die automatische Konfiguration braucht es keinen Mitarbeiter, der die jeweilige Sprache konfiguriert. Softwaremässig kann dies mit bescheidenem Aufwand umgesetzt werden. Dabei werden alles Sprachen auf der SD-Karte platziert

und mit einer entsprechenden Endung, wie z.B en, de oder it versehen. Durch die zugehörige Beacon-ID kann dann jeweils in allen Suchalgorithmen die Namensgebung hinzugefügt werden. Somit wird dann automatisch die richtige Audio-Datei mit der richtigen Sprache ausgegeben.

Like Button mit Netz.

## 7 Ehrlichkeitserklärung

Der Projektleiter bestätigt mit der Unterschrift, dass der Bericht selbst verfasst und alle Quellen sauber und korrekt deklariert wurden.

---

Ort, Datum

---

Unterschrift Projektleiter

## Literatur

- [1] Emmerich, "Data Sheet Rechargeable Li-Ion Cell with PCM and Wire," Website, 2014, online erhältlich unter [http://www.produktinfo.conrad.com/datenblaetter/1200000-1299999/001221199-da-01-en-EMMERICH\\_LIION\\_AKKU\\_14500\\_MIT\\_KABELN.pdf](http://www.produktinfo.conrad.com/datenblaetter/1200000-1299999/001221199-da-01-en-EMMERICH_LIION_AKKU_14500_MIT_KABELN.pdf); abgerufen am 04. Mai 2018.
- [2] Nordic Semiconductor, "nRF52832 Chip," Website, unknown, online erhältlich unter [http://www.nordicsemi.com/node\\_206/node\\_305/Product-Related-News/Nordic-nRF52832](http://www.nordicsemi.com/node_206/node_305/Product-Related-News/Nordic-nRF52832); abgerufen am 27. April 2018.
- [3] ——, "nRF52 Development Kit," Website, unknown, online erhältlich unter <https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52-DK>; abgerufen am 27. April 2018.
- [4] Adafruit, "Bone Conductor Transducer With Wires - 8 Ohm 1 Watt," Website, unknown, online erhältlich unter <https://www.adafruit.com/product/1674>; abgerufen am 26. April 2018.
- [5] N. Peter, "Allgemeine Elektrotechnik," 2009, unpublished thesis.
- [6] Wikipedia, "Knochenleitung," Website, 2018, online erhältlich unter <https://de.wikipedia.org/wiki/Knochenleitung>; abgerufen am 26. April 2018.
- [7] NORDIC Semiconductor, "Software development kit," Download, Unknown, online erhältlich unter [https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF5-SDK/\(language\)/eng-GB](https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF5-SDK/(language)/eng-GB); abgerufen am 10. März 2018.
- [8] ——, "Dokumentation," Website, Unknown, online erhältlich unter <http://infocenter.nordicsemi.com/index.jsp>; abgerufen am 10. März 2018.
- [9] ChaN, "Fatfs - generic fat filesystem module," Website, Unknown, online erhältlich unter [http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html); abgerufen am 10. März 2018.
- [10] NORDIC Semiconductor, "Software development kit common license," Download, Unknown, online erhältlich unter [http://developer.nordicsemi.com/nRF5\\_SDK/nRF5\\_SDK\\_v15.x.x/doc/15.0.0/nRF5\\_Nordic\\_license.txt](http://developer.nordicsemi.com/nRF5_SDK/nRF5_SDK_v15.x.x/doc/15.0.0/nRF5_Nordic_license.txt); abgerufen am 10. März 2018.

## Abbildungsverzeichnis

1	Dōjō als Modell . . . . .	5
2	Teilsysteme des Dōjōs . . . . .	6
3	Anwendungsablauf des Dōjōs . . . . .	7
4	Sprachauswahl mittels Bluetooth-Beacon . . . . .	8
5	Ne555 . . . . .	10
6	Laderegelung Li-Ion Akku . . . . .	12
7	Prototyp Ladestation . . . . .	13
8	Mikrocontroller NRF52832 . . . . .	14
9	Knochenschallaktor [4] . . . . .	15
10	Software:Layers . . . . .	16
11	Software:Pollend . . . . .	17
12	Statemachine-Diagramm . . . . .	18
13	Statemachine: lookup . . . . .	19
14	Statemachine: Special Beacon . . . . .	20
15	Statemachine: Change Language . . . . .	22
16	Statemachine: New Beacon . . . . .	23
17	Statemachine: Wait . . . . .	24
18	Statemachine: ADC Battery . . . . .	25
19	Statemachine: Charge . . . . .	26
20	PacketPayload Header . . . . .	28
21	PacketPayload Payload . . . . .	29
22	Software:Definierte MM . . . . .	30
23	Software:MM Vergabe . . . . .	31
24	Formatdefinition Lookup-File . . . . .	33
25	PWM-Ausgang des NRF52 . . . . .	34
26	Ablauf PWM . . . . .	34
27	Ladekurve Emmerich LI14500 . . . . .	38
28	Induzierter Strom in Abhängigkeit der Distanz . . . . .	38
29	Verwendete Timerschaltung NE555 als Pulsquelle . . . . .	41
30	500Hz Sinus Signal und FFT ohne LC-Filter . . . . .	43
31	500Hz Sinus Signal und FFT mit LC-Filter . . . . .	44
32	PWM Signal beider PWM Ausgänge . . . . .	45
33	PWM auf 32kHz eingestellt . . . . .	46
34	PWM auf 128kHz eingestellt . . . . .	47
35	Überlappung der Referenz-Felder . . . . .	48

## A Anhang

### A.1 Messresultate

bcsjdbcsjbvsvb