

FACHBERICHT

FS18 - PRO4E - TEAM 5

31. Mai 2018

AUFTRAGGEBER:

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 XX mAh and can provide energy for XX hours when the Dojo is playing audiofiles and be recharged within XX hours. To protect the battery from damage, a deep discharge protection turns off the Dōjō when the battery is at the XX % of its power.

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

Inhaltsverzeichnis

1	Einleitung	4
2	Gesamtkonzept	5
2.1	Funktionsweise	5
2.2	Anwendung	6
3	Hardware	9
3.1	Energieübertragung	9
3.2	Li-Ion-Batterie	13
3.3	Mikrocontroller	14
3.4	Verstärkerstufe	15
3.5	Knochenschallaktor	15
4	Software	17
4.1	State-Machine	19
4.2	Nordic Software Development Kit	26
4.3	Bluetooth	27
4.4	SD-Karte	32
4.5	Audio, PWM	33
4.6	Lizenzen	35
5	Validierung	36
5.1	Testkonzept Hardware	36
5.2	Testkonzept Software	38
6	Schlusswort	39
7	Ehrlichkeitserklärung	41
A	Anhang	44
A.1	Messresultate	44

1 Einleitung

Zielloses Umherwandern und zu wenig Wissen über das Besondere an der Kunst sind nur einige Gründe weshalb bei einem Museumsbesuch oft der Besuch selbst mehr in Erinnerung bleibt als die eigentlichen Kunstobjekte. Um diesem Problem entgegenzuwirken, hat Frau J. Kalbermatter das Gehäuse eines modernen Audioguides entwickelt und sich dazu ein funktionales Konzept ausgedacht. Dieser Audioguide, Dōjō genannt, soll Kunstobjekte drahtlos erkennen und darüber gespeicherte Informationen via Körperschalltechnik zum Museumsbesucher bringen. Dazu muss sich der Benutzer das Ende des stabförmigen Audioguides hinter das Ohr halten. Dank der Informationsübertragung via Körperschall und dem Verzicht auf Kopfhörer, erfüllt der Dōjō höchste Hygieneansprüche, was die Benutzung angenehmer macht. Weiter soll der Audioguide auch als Ticket fungieren und die Zutrittsberechtigung regeln. Das Drücken einer implementierten „Like“-Taste soll vermerken, welche Kunstobjekte dem Besucher gefallen haben. Das ermöglicht es, die Informationen über ein interessantes Kunstobjekt am Ende des Besuchs in Form einer Broschüre oder per E-Mail an den Besucher zu übergeben. Mit diesen Funktionen wird der Dōjō den Museumsbesuch hygienisch und angenehmer gestalten. Ausserdem wird er helfen, das Allgemeinwissen zu verbessern. Dadurch werden die Kunstobjekte und nicht der Besuch selbst zu einem unvergesslichen Erlebnis werden.

Ziel im Projekt 4 im Studiengang Elektro- und Informationstechnik an der Fachhochschule Nordwestschweiz war es deshalb, 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 Mikrokontrollers (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 Mikrokontroller beinhaltet die Software und übernimmt somit die Erkennung, Ansteuerung und Koordination der Hardware.

Es wurde ein Prototyp der Elektronik realisiert. Er kann zu XXX 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 XXX mAh, welcher bei pausenloser Audioausgabe genug Energie für XXX Stunden liefert. Die Induktionsladung lädt den Akku zu 100% innert XXX Stunden. Ausserdem sorgt ein Tiefenentladungsschutz dafür, dass der Dōjō bei XXX% Akkuladestand ausgeschaltet wird um den Akku vor Schäden zu bewahren.

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 die Hardware und Software erklärt werden kann, ist ein Einblick in das Gesamtkonzept des Dōjōs 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 Abschnitt 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 Eigenheiten 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 Informationen auszuhändigen, oder in irgendeiner Form online zur Verfügung stellen. Ansonsten verfügt der Dōjō über die selben Funktionen, die man von einem üblichen Audioguide erwarten würde, wie z.B. der Audiowiedergabe, 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, welcher über ein induktives Ladesystem versorgt wird. Abbildung 2 zeigt das technische Lösungskonzept.

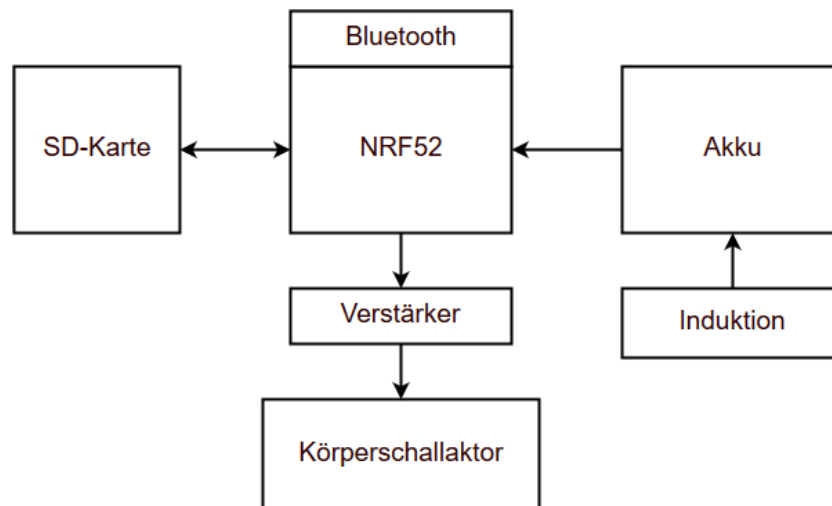


Abbildung 2: Teilsysteme des Dōjōs

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 Dōjō 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 kurzzeitig 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.

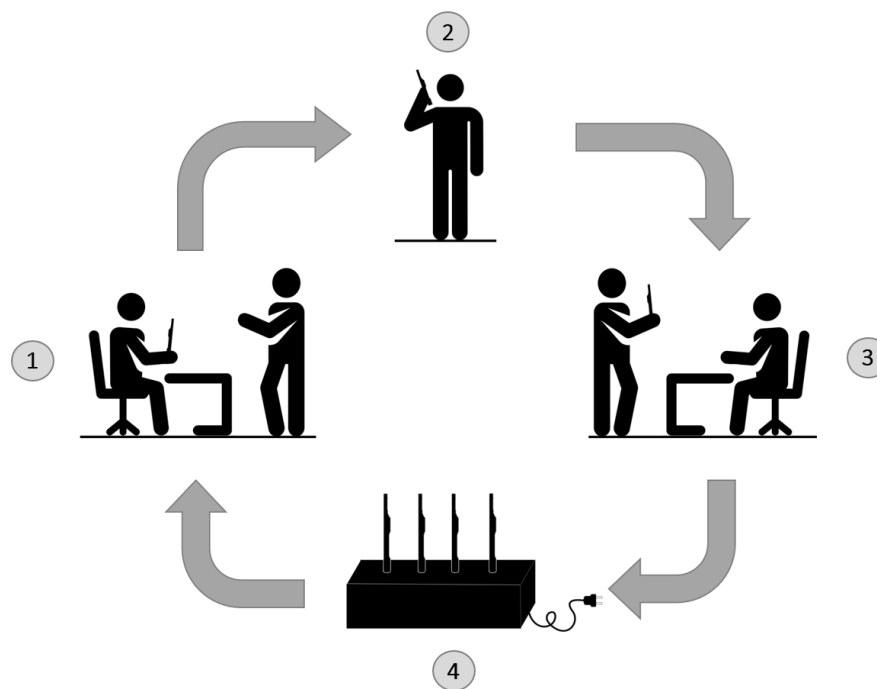


Abbildung 3: Anwendungsablauf

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 hinhalten 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.

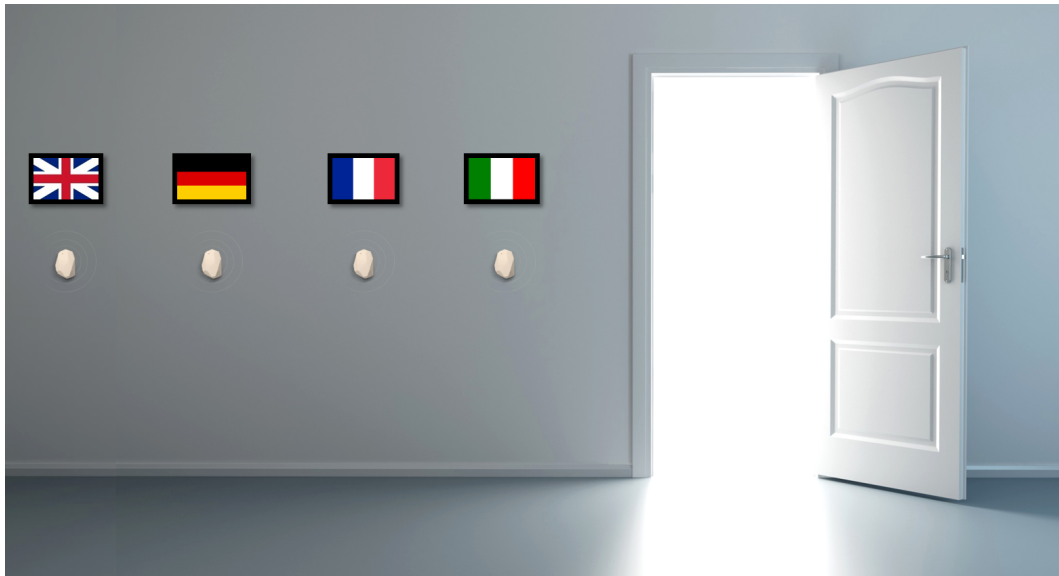


Abbildung 4: Sprachauswahl mittels Bluetooth-Beacon

3 Hardware

Nachdem die ganweize Struktur und der Funktionsablauf der Anwendung zuvor 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 Energieübertragung

Mit Hilfe des Induktionsprinzips wird der Dōjō geladen und zwar ohne, dass dieser angeschlossen oder in einer anderen Form vorbereitet werden muss. Dank der induktiven Ladeschaltung, erfolgt der Ladezyklus sobald der Dōjō in die dafür vorgesehene Ladebuchse eingeführt wird. Im folgenden Abschnitt wird die genaue Funktionsweise der Induktionsladeschaltung vorgestellt und analysiert:

Grundlegendes

Beim Induktionsladeprinzip wird 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 im Grundsatz aus einem Pulsgenerator, mit welchem das LC-Glied gepulst wird. Sie macht den Hauptanteil einer solchen Induktiven Ladeschaltung aus. Die zweite, kleinere Schaltung, welche die Energie des Transceivers empfängt, wird Receiver genannt und besteht ebenfalls aus einem LC-Glied, ergänzt mit einem Gleichrichter.

Tranceiver

Der bereits erwähnte Pulsgenerator wird durch eine Timer-Schaltung erreicht. 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 unterschiedlich einstellen. Massgebend für die Veränderung der Pulsdauer, Frequenz und des Duty-Cycles sind hierbei hauptsächlich unterschiedliche 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. Der 2N3055 Leistungstransistor wird nicht mehr sauber durchgesteuert und der Strom wird somit begrenzt. Die Strombegrenzung ist abhängig, welche Teile für das LC-Glied verwendet werden. 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. Würden grössere Spulen verwendet, so wäre spätestens eine Begrenzung bei 15A notwendig, da dies die Belastungsgrenze für den 2N3055 Leistungstransistor ist.

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 $R2 \gg R1$ 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_1 beeinflusst die High-Time, lässt jedoch die Low-Time unverändert.
- R_2 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 3.1 bis 3.6 berechnet.

$$F = 1/T = 1.44/((R1 + R2 * 2) * C) \quad (3.1)$$

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

$$TH = 0.693 * (R1 + R2) * C \quad (3.3)$$

$$D = DutyCycle = (R1 + R2)/(R1 + 2 * R2) \quad (3.4)$$

$$R1 = 1.44 * (2 * D - 1)/(F * C) \quad (3.5)$$

$$R2 = 1.44 * (1 - D)/(F * C) \quad (3.6)$$

Für die Berechnung der effektiven Werte, müssen die ersten Werte angenommen werden. Dadurch entstanden folgende für den Transceiver am besten geeigneten Werte:

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

Daraus ergibt sich bei Anwendung der Formel 3.1 eine Frequenz von 79.12kHz. Nachdem die Frequenz berechnet wurde, können die Elemente eingebaut werden. 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.

Speziell ist, dass für die Energieübertragung als Sender- und Empfängerspule das selbe LC-Glied verwendet wurde. Dies wurde so gewählt, da das Energiefeld sehr klein ist und im Falle einer grösseren Spule das Receiver Glied nicht optimal ausgenutzt werden könnte. Wie der Receiver aufgebaut ist, wird im nachfolgenden Abschnitt genauer erläutert.

Receiver

Der Receiver besteht primär aus einem LC-Glied und einem Gleichrichter. Für die Spule des LC-Gliedes wurde eine kleine Flachspule verwendet, welche eine Dimension von 15mm Durchmesser und 2mm Höhe aufweist, und somit im inneren des Dōjō's am Boden montiert werden kann. 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 0.1 V aufweisen. Anschliessend wird die gesamte Ladeschaltung gespiesen, welche den gesamten Ladeprozess des Akkus übernimmt. Einen Einblick in den gesamten Ladeprozess gibt das Kapitel 3.2.

Ladestation

Nachdem die gesamte induktive Ladeschaltung beschrieben wurde, folgt die Beschreibung der Ladestation. Hierfür wurde ein Prototyp erstellt, welche zum einen die gesamte primäre induktive Ladeschaltung beinhaltet, zum anderen eine Aussparung, welche als Ladeeinrichtung für den Dōjō dient, vorweist. Für den Prototypen wurde eine .stl Datei erstellt welche mit einem 3D-Drucker gedruckt wurde. Wichtig ist hierbei zu erwähnen, dass es sich bei nachfolgende Abbildungen 5 bis 7 nur um Prototypen handelt und es bei einer Weiterentwicklung noch Anpassungen geben kann. Da die Ladestation für Versuchszwecke betreffend der induktiven Ladeschaltung 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 5: Frontansicht - Ladestation Dōjō

Die oben gezeigte Abbildung gibt einen Einblick in das Design von vorne. Augenfällig ist die Öffnung für den Dōjō selbst, welche zum einen als Standhalterung und zum anderen als korrekte Positionierung für die induktive Ladeschaltung dient. Die richtige Positionierung ist hierbei eines der wichtigsten Kriterien für einen optimalen Ladezyklus, da die Tranceiver- und Receiverspule direkt übereinanderliegend den besten Wirkungsgrad erzielen. Weiter ist in der Abbildung der Dōjō Schriftzug ersichtlich, welcher bis in die dahinter liegende Kammer führt. Die Verwendung dieser Durchführung wird nachfolgend unter der Abbildung 7 weiter erklärt.

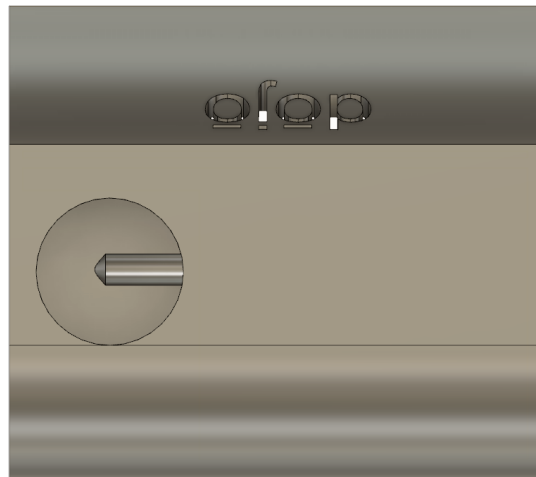


Abbildung 6: Draufsicht - Ladestation Dōjō

Die obige Abbildung fig: Prototyp Top zeigt den Prototypen von oben. In der Aussparung für den Dōjō ist ein Kanal für die Verkabelung der Primärspule ersichtlich. Diese Öffnung führt zur Kammer welche den Primär-Ladekreises beinhaltet. Die Abmessung (Länge x Breite x Höhe) des Prototypen ist (80 x 70.7 x 30)mm. Einen Einblick in die Kammer für die Elektronik des Primär-Ladekreises gibt nachfolgende Abbildung 7, welche den Prototypen von unten zeigt.



Abbildung 7: Ansicht von Unten - Ladestation Dōjō

Die grosse Kammer ist wie bereits oben beschrieben für den Primärkreis der induktiven Ladeschaltung vorgesehen. Für die ersten Tests 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 ersichtlich. Diese Durchführungen sind für LED vorgesehen, welche den Dōjō Schriftzug bei angeschlossener Versorgungsspannung zum Leuchten bringt.

3.2 Li-Ion-Batterie

Die gesamte Energiespeicherung erfolgt durch einen Lithium-Ionen-Akkumulator des Typs Emmerich LI14500. Dieser weist eine Kapazität von $800mAh$ bei einer Nominalspannung von $3.7V$ auf. Ausserdem weist der Akkumulator integrierte Schutzseinrichtungen auf, welche später im Abschnitt Schutzseinrichtungen (Kapitel 3.2) weiter erläutert werden. 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 Knochenschallgebers und die des Microcontrollers beschreiben. Alle anderen Komponenten können durch ihren geringen Betriebsstrom durch einen Sicherheitsfaktor von $0.1W$ dazu gerechnet werden. Die Leistung des Verstärkers welcher den Knochenschallgeber speist, weist eine maximale RMS Leistung von $471.9mW$ auf. Die Rechnung erfolgt mit 80% des RMS Wertes, da der Knochenschallgeber nicht rund um die Uhr angesprochen wird. Die Microcontrollerleistung lässt sich durch den Radio Strom ($7.5mA$) und einigen Mikroampere Systemstrom (gesamthaft ca. $100\mu A$) multipliziert mit der Systemspannung von $3.6V$ bestimmen. Die Microcontrollerleistung wird durch die Nominalspannung multipliziert mit dem maximalen Microcontrollerstrom von $7.6mA$ berechnet. Somit wird maximal eine Leistung von $0.63W$ erreicht, wobei nachfolgend in Berechnung 3.7 die Herleitung der maximalen Leistung veranschaulicht wird.

$$P_{max} = (0.8 \cdot P_{Kn}) + P_{MC} + P_{zus} = (0.8 \cdot 0.472W) + (3.7V \cdot 7.6mA) + 0.1W = 0.506W \quad (3.7)$$

Die darausfolgende minimale Zeit t folgt durch nachfolgende Berechnung 3.8.

$$t_{max} = \frac{W \cdot U}{P_{tot}} = \frac{800mAh \cdot 3.7V}{506mW} = 5.85h \approx 5h \ 51min \quad (3.8)$$

Bei ständigem Gebrauch kann somit eine minimale Betriebszeit von knapp 6 Stunden erreicht werden. Hierbei gilt es zu erwähnen, dass durch einen geschickte Anwendungsablauf (ersichtlich in Kapitel ?? Anwendung) ein lückenloser Betrieb garantiert werden kann.

Schutzseinrichtungen

Um den verwendeten Akkumulator zu schützen, sind diverse Schutzseinrichtung notwendig. Zum einen muss der Ladevorgang überwacht werden, so dass der maximale Ladestrom wie auch die Ladespannung nicht überschritten werden. Für die Laderegulation wurde ein Lade-IC von Microchip des Typs MCP73831 verwendet. Dieser übernimmt die gesamte Spannungs- und Stromregelung beim Ladevorgang und steuert zu dem während dem Ladevorgang eine LED zur Ladesignalisation an. Der Ladevorgang für den oben erwähnten Li-Ion Akku ist in untenstehender Abbildung 8 ersichtlich. Hierbei wurde der Akku im Schnelllademodus mit einem maximalen Strom von $400mA$ geladen. Dieser Strom ergibt sich aus dem Datenblatt der Batterie, wobei sowohl der Entladestrom, als auch der Ladestrom $0.5C$ beträgt. Das C entspricht der Kapazität der Batterie, wodurch sich der Strom I_{max} gemäss der nachfolgenden Formel 3.9 berechnen lässt.

$$I_{charge} = \frac{0.5}{h} \cdot C = \frac{0.5}{h} \cdot 800mAh = 0.4A \hat{=} 400mA \quad (3.9)$$

Betrachtet man die Abbildung 8 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.

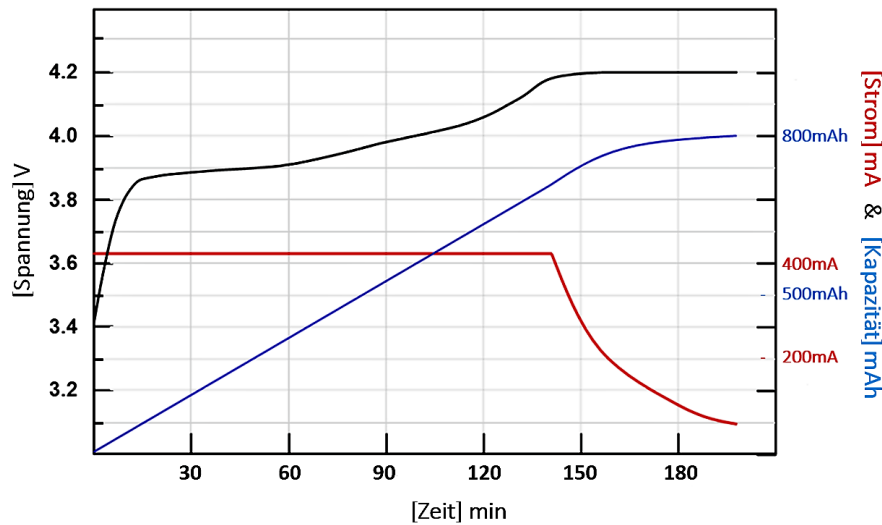


Abbildung 8: Blockschaltbild Energiespeicherung

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 und weist zudem einen Schutzschaltungswiderstand von $\leq 75m\Omega$ auf.

3.3 Mikrocontroller

Als Microcontroller wurde ein nRF52832 der Firma Nordic Semiconductors verwendet (ersichtlich in Abbildung 9). Seine hohe Performance ermöglicht es ein System aufzubauen, welches den Microcontroller als zentrale Schnittstelle beinhaltet. Der Microcontroller bildet wie in Kapitel 2 gemäss Abbildung 2 bereits beschrieben das Herzstück des Dōjōs.

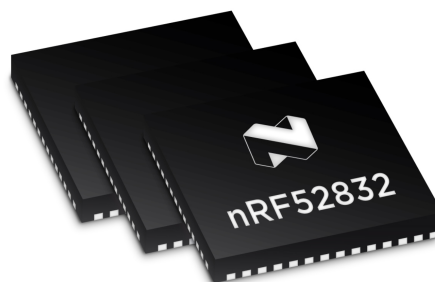


Abbildung 9: nRF52832 Microcontroller [1]

der nRF52832 weist eine Betriebsversorgungsspannung zwischen 1.7V und 3.6V mit einem Versorgungsstrom von 5.4mA auf. Der Speicher des nRF52832 ist durch 512kB flash/64kB RAM Speicher gegeben. Für die ersten Tests der Software wurde ein Development Kit (Abbildung 10) mit integrierem nRF52832 verwendet. Die Beschreibung der Software ist in Kapitel 4 ersichtlich.

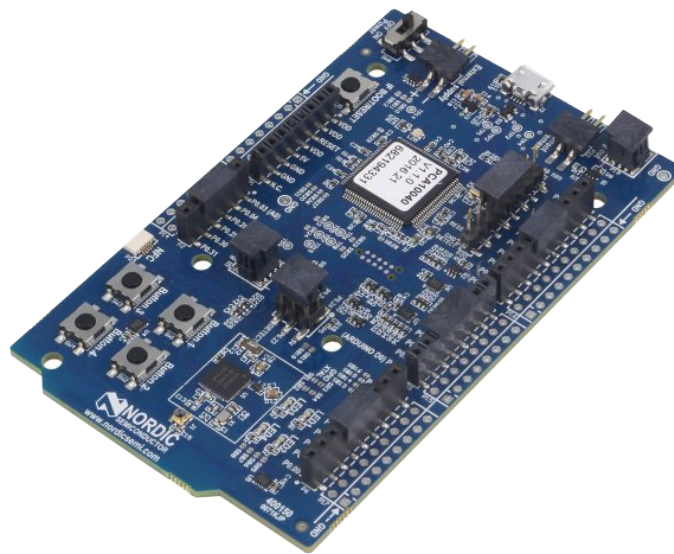


Abbildung 10: nRF52 Development Kit [2]

Die Schlüsselmerkmale dieses Boards sind zum einen wie bereits beschrieben der integrierte Microcontroller, zum anderen aber auch eine integrierte Bluetooth Antenne. Das Kit unterstützt proprietäre Bluetooth Smart, ANT und 2.4GHz Applikationen. Zudem ermöglicht es den Einsatz von Drittanbieter Shields. Dies ermöglicht das Beschreiben und Lesen der SD-Karte zu testen.

3.4 Verstärkerstufe

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 2 Arten von Verstärkern. Sie können digital oder auch analog umgesetzt werden. Beide erfüllen die gleiche Aufgabe, weisen jedoch bezüglich Wirkungsgrad einen deutlichen Unterschied auf. Die digitale Variante weist ungefähr einen Wirkungsgrad von 90% auf [3], während die analoge Variante maximal einen Wirkungsgrad im Bereich der Leistungsanpassung erzielt [4]. Aus diesem Grund wird ein digitaler Verstärker (Class-D-Verstärker) in der Anwendung implementiert. Die Wahl fiel auf den Stereo-Amplifier MAX 98306 (nicht ganz sicher was nacher wirklich implementiert wird).

(Strom/Spannung/Leistung) ergänzen und begründen, Berechnung der Verstärkung

3.5 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.[5] Für die Anwendung im Dōjō wird ein Knochenschallaktor des Herstellers Adafruit verwendet, welcher in der Abbildung 11 ersichtlich ist.

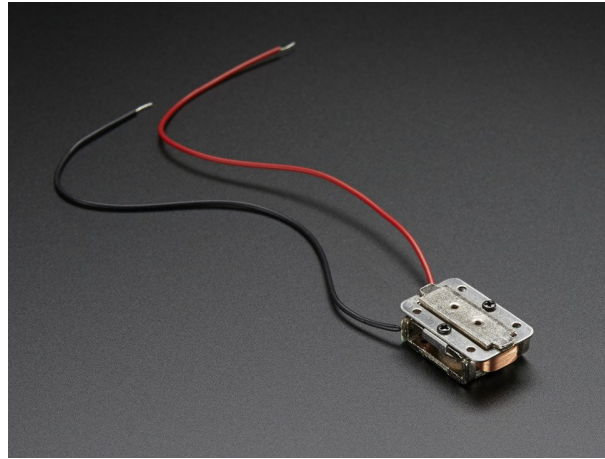


Abbildung 11: Knochenschallaktor von Adafruit

Das ausgewählte Bauteil eignet sich bestens für die Verwendung im Dōjō. Mit einem Gewicht von 9.6 g und den Dimensionen 14x21,5x7,9 lässt sich der Aktor gut in das bestehende Gehäuse implementieren [3]. Weiter ist Bauteil relativ kostengünstig im Handel erhältlich und kann $1W_{RMS}$ Leistung liefern, was sich dann in der Lautstärke bemerkbar macht. Nach ausführlichen Rechercharbeiten konnten keine wirklichen Alternativen ausgemacht werden. Meist befindet sich die Technologie noch in der Entwicklungsphase oder fällt aufgrund des Preises aus der Auswahlmöglichkeit.

4 Software

Wichtig für dieses Kapitel ist, dass der beschriebene Zustand der Software dem Soll-Zustand entspricht. Die Abweichungen sind im Kapitel Validierung genauer beschrieben.

Es wird ein NRF52832 verwendet von Nordic Semiconductor. Dadurch liegt die Verwendung des Software Development Kit [6] nahe. Dies ist eine Sammlung von Beispielen, Librarys und vorcompilierten Codes. Nachfolgend wird das Software Development Kit nur noch SDK genannt. Es wurde nRF5 SDK v12.3.0 verwendet. Wichtig ist die SDK für ihren integrierten Bluetooth-Stack, der verwendet wird. 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 12 visualisiert. Die wichtigsten Module des SDK werden im Kapitel 4.2 erklärt, falls weitere Informationen gewünscht sind wird auf die offizielle Dokumentation verwiesen [7].

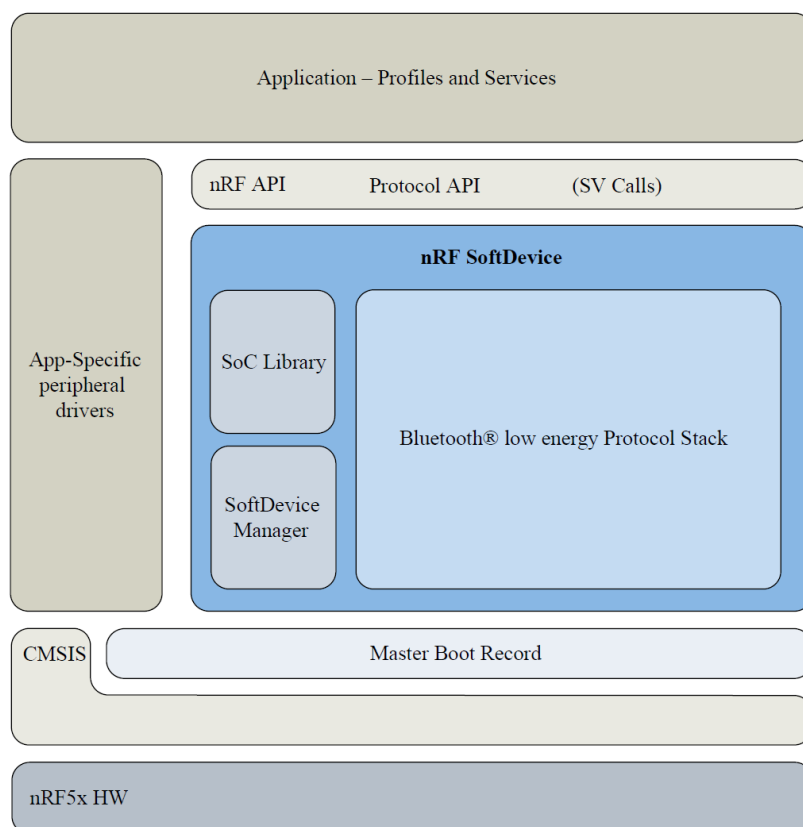


Abbildung 12: Aufbau der Software auf der Hardware

Der eigentliche Programmaufbau ist eine Statmaschine. Diese ist in einem separaten Kapitel erläutert. Das Programm ist so aufgebaut, 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, genauers im Kapitel 4.1. Dadurch entsteht ein pollendes Programm mit Prioritäten, was in Abbildung 13 dargestellt ist.

Das Programm wurde in verschiedene Module unterteilt, um die Leserlichkeit zu verbessern. Die Unterteilung wurde entsprechend der Funktion gemacht. Im Hauptmodul (Main) ist nur

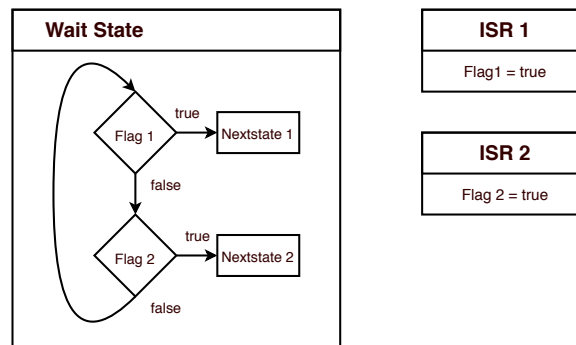


Abbildung 13: Konzept der pollenden Software

die Statmaschine und die Interrupthandler vorhanden. Folglich wurde ein Modul für die BLE-Funktionalitäten geschrieben und eines für die SD-Karte. Ein weiteres existiert für die Batterie-Funktionalitäten. Dieses ist jedoch nicht implementiert und enthält nur Dummy-Funktionen.

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 State-Machine, 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 14 zeigt das Gesamtkonzept der State-Machine. Anschließend werden die einzelnen States genauer definiert und beschrieben.

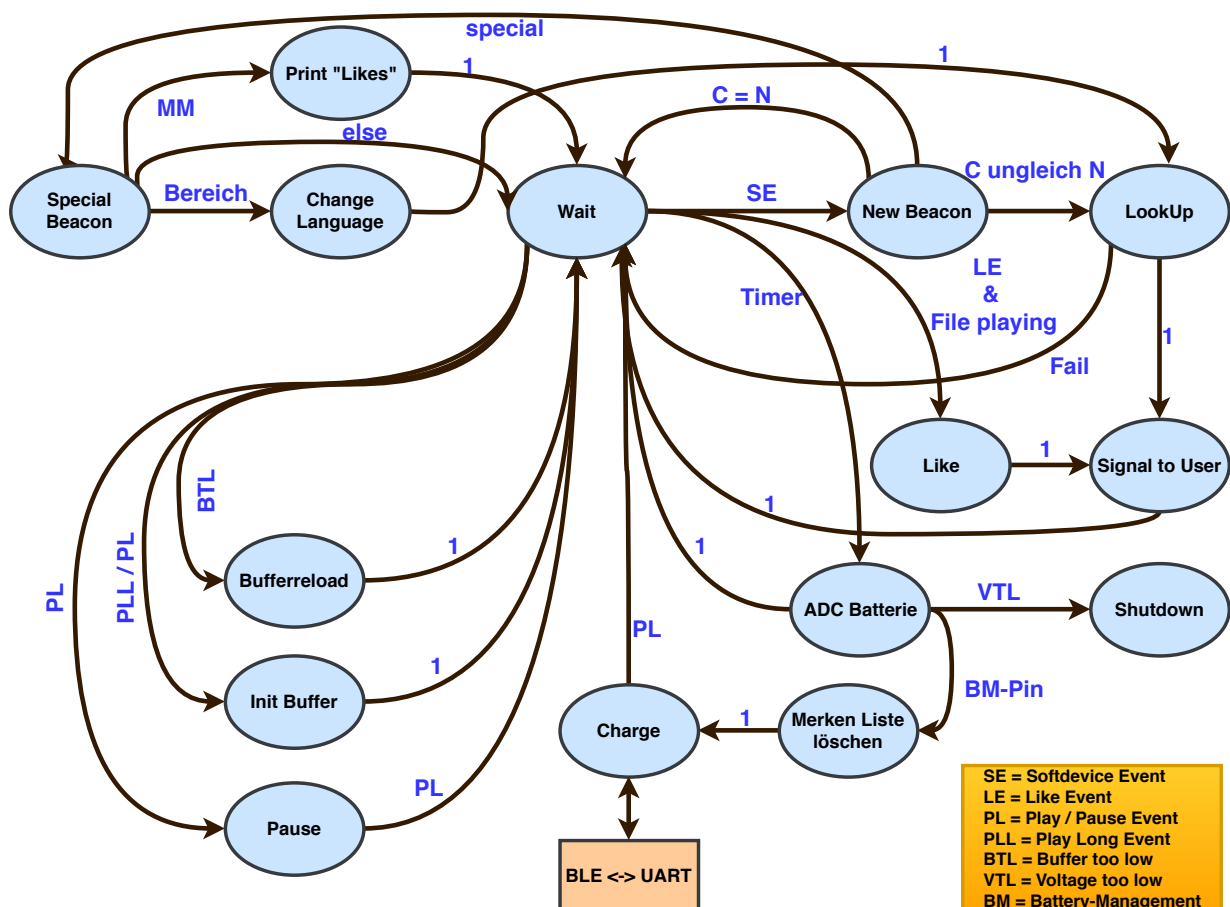


Abbildung 14: Statemachine im Überblick mit den einzelnen States und den Parametern

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. Anderenfalls wird dem Mikrocontroller signalisiert, dass der Zugriff geglückt ist und die eigentliche Funktion wird gestartet. Dazu werden die beiden Minor- und Majorzahlen in ein hexadezimalen 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 15 zeigt den detaillierten Funktionsablauf im Lookup-State.

!!!Verknüpfung Bluetooth und Sound File!!!

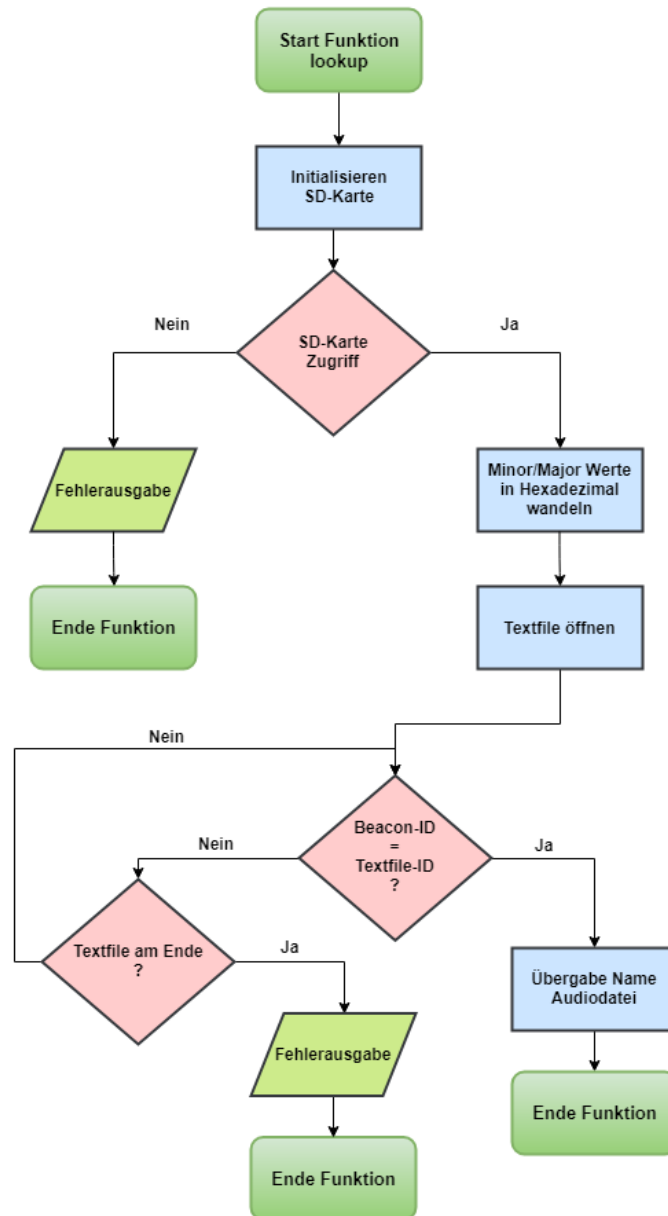


Abbildung 15: Funktionsablauf im lookup-State

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. Anderenfalls 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 16 zeigt den Ablauf der Funktion.

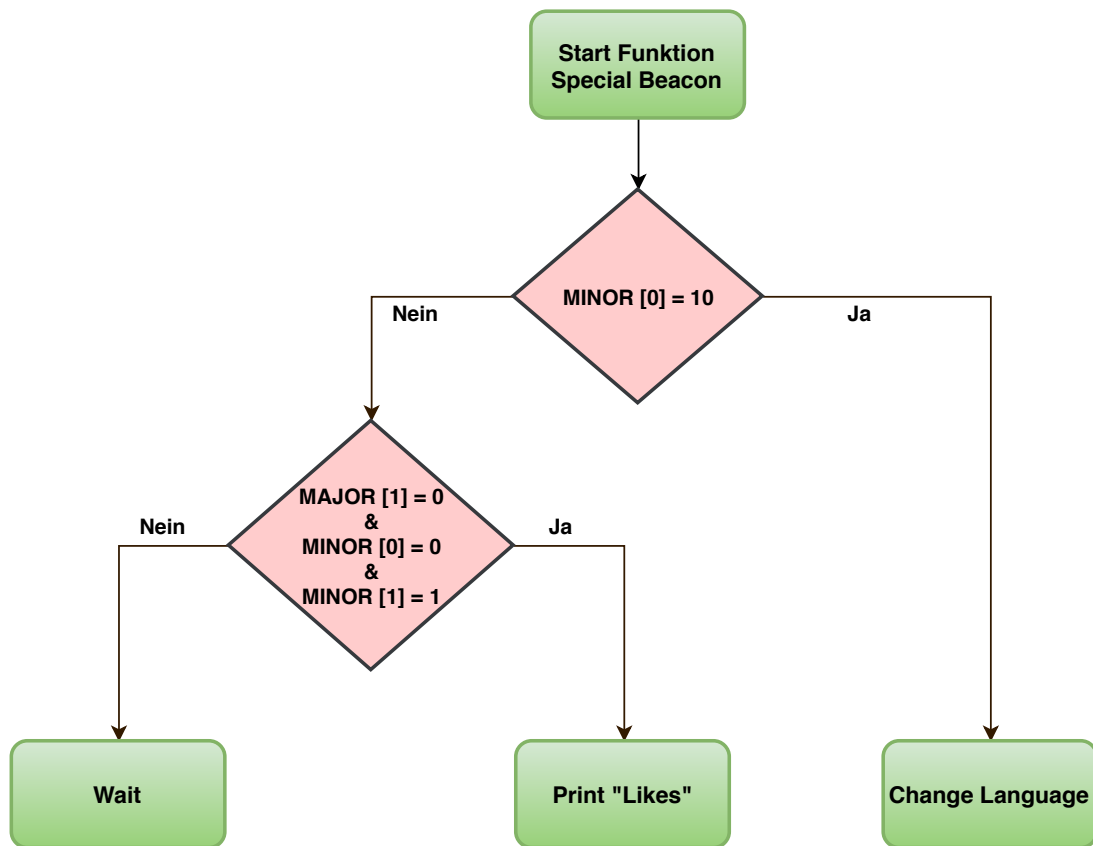


Abbildung 16: Funktionsablauf im Special Beacon State

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.

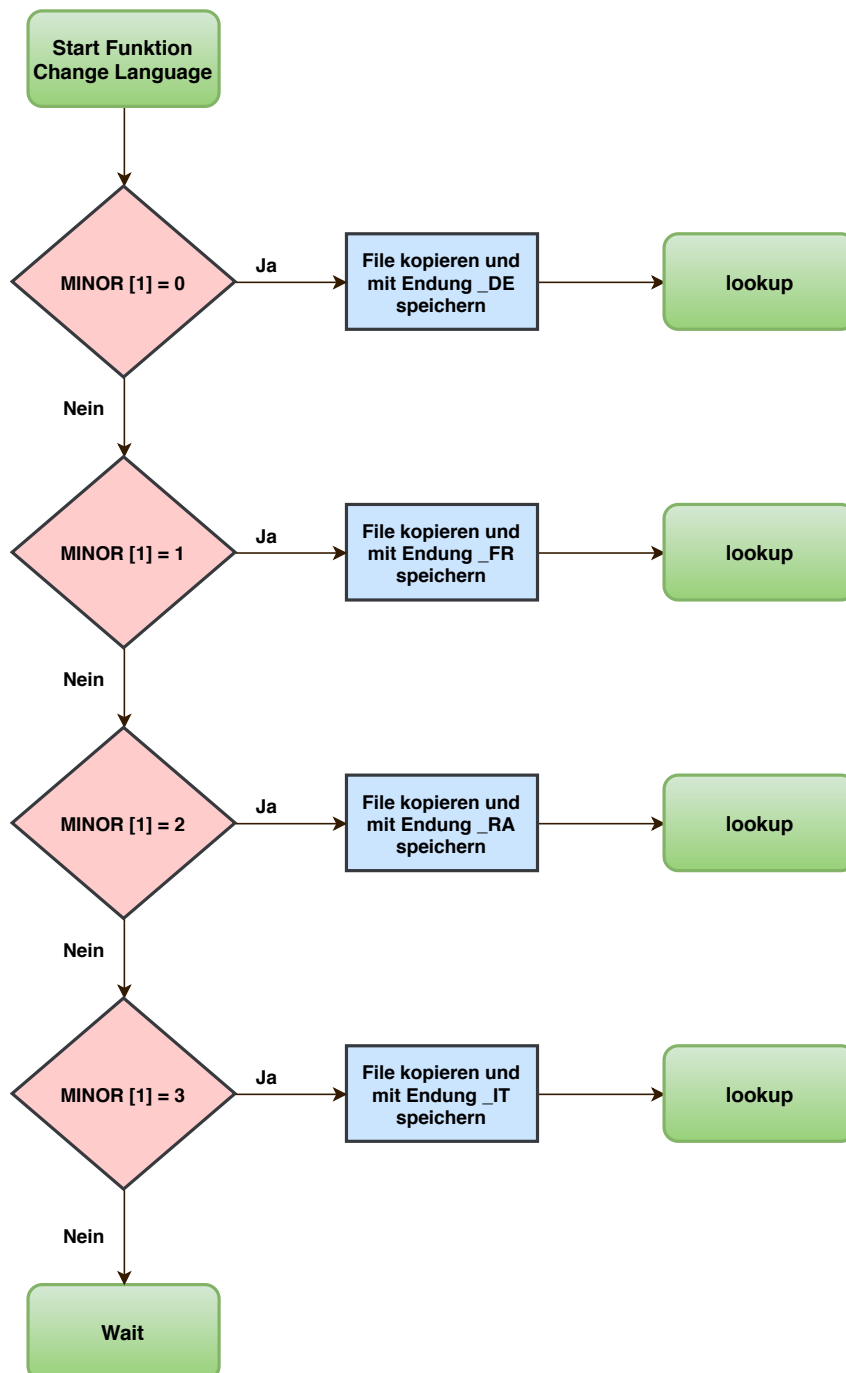


Abbildung 17: Funktionsablauf im Change Language State

State: Print „Likes“

Print „Likes“ wurde nicht implementiert und das Hauptprogramm springt 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 Broschüre mit den interessanten Objekten direkt gedruckt werden und bereits für den Besucher am Ausgang des Museums bereit liegen.

State: Wait

Dieser State ist die Ausgangslage des kompletten Ablaufs. Aufgrund von den in der Abbildung 18 ersichtlichen Bedingungen und sogenannten Flags, entscheidet das Hauptprogramm in welchen State zu switchen ist. Die Reihenfolge der Abfrage erfolgt nach dem Prioritätsprinzip.

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: 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 20 entsprechend gehandelt. Wird die Batterie geladen, so wird die Like-Liste gelöscht. Anderenfalls 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.

State: Merken Liste löschen

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

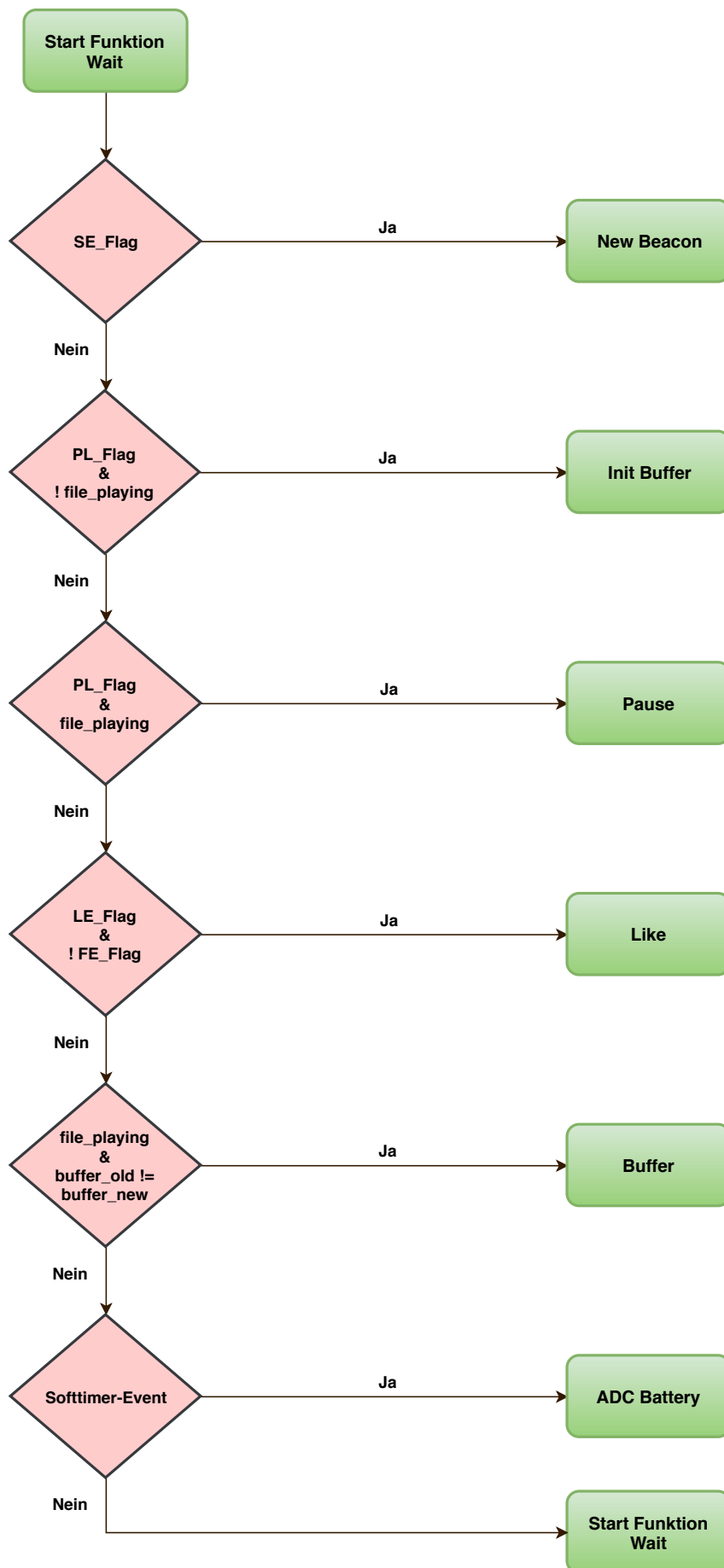


Abbildung 18: Funktionsablauf im Wait State

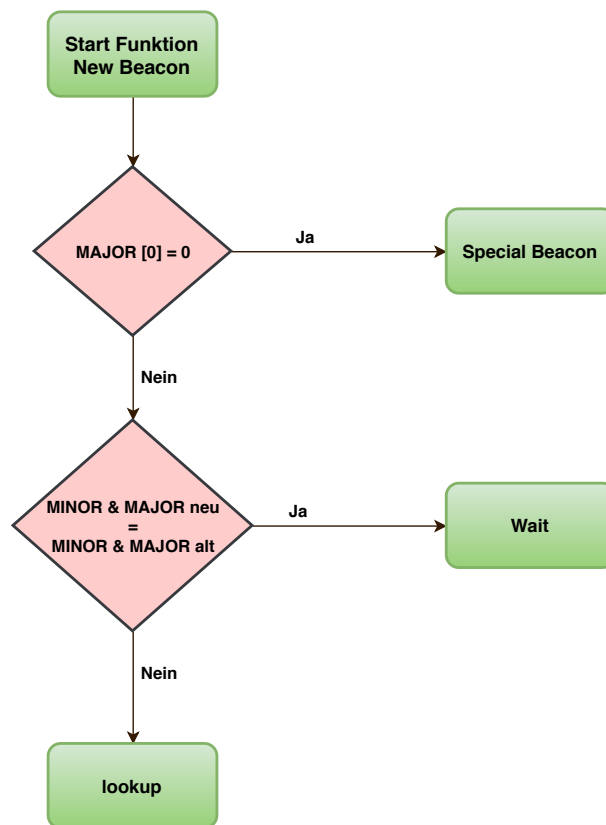


Abbildung 19: Funktionsablauf im New Beacon State

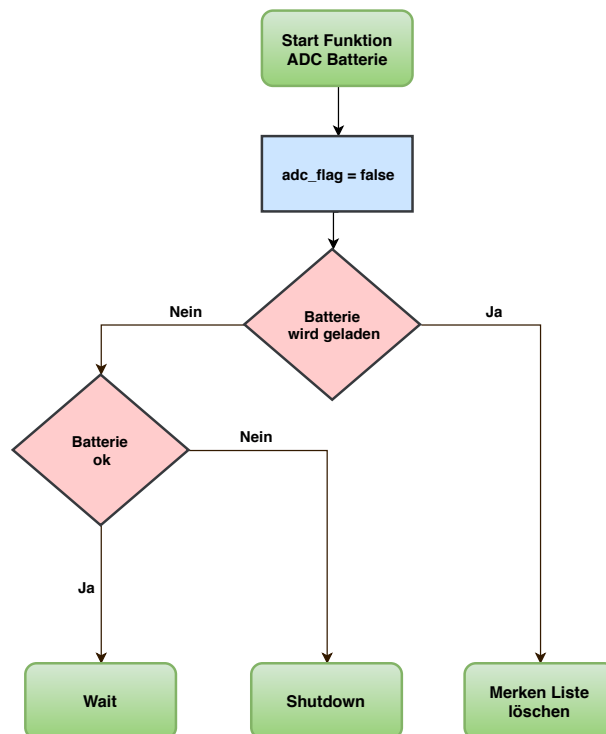


Abbildung 20: Funktionsablauf im ADC Battery State

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 Schleife. Darin ist es möglich den Dojo mit der Ladestation zu verbinden und über UART Anpassungen auf der SD-Karte vorzunehmen.

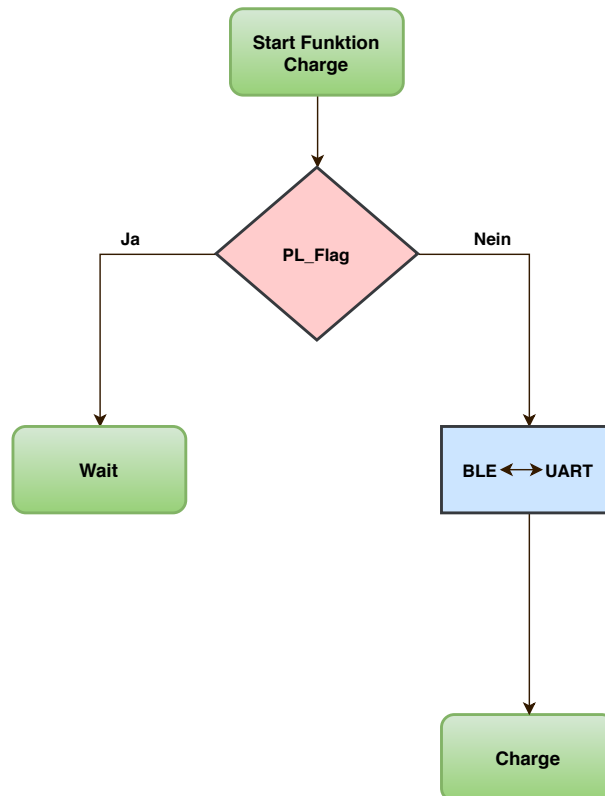


Abbildung 21: Funktionsablauf im Charge State

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.

4.2 Nordic Software Development Kit

Beschreibt die Aufgabe des MCs mittels Ablaufdiagramm, sowie welche Funktionen dazugehören und deren Wirkung. Funktionen, Prinzip des Kits, Example des ursprünglichen Aufbaus

4.3 Bluetooth

4.3.1 Bluetooth Grundlagen

Standard-Bluetooth-Geräte senden in einem lizenzfreien 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 Meter 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. Genanntes Verfahren nennt man Broadcasting. Das Broadcasting ist für dieses Projekt von zentraler Bedeutung. Dennoch wird im nächsten Kapitel erläutert, wozu das GATT dienlich 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 dazugehörige 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 Kopfbatterien über Jahre hinweg in Betrieb sein und diese kleine Informationsmenge 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 in der Lage sind dazu. 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 [?].

Von BLE-Beacons übermittelte Signale werden formatiert wie in Abbildung 22 gezeigt [?].

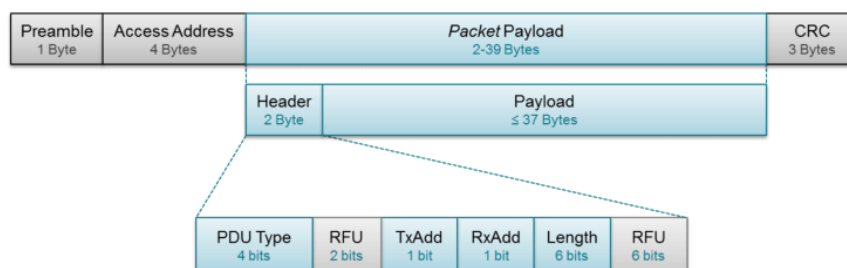


Abbildung 22: PacketPayload Header

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 [?]:

RFU (Reserved for Future Use) steht, wie der Name schon sagt, als Reserve für zukünftige

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

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

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 aufgebaut gemäss Abbildung 23 [?].

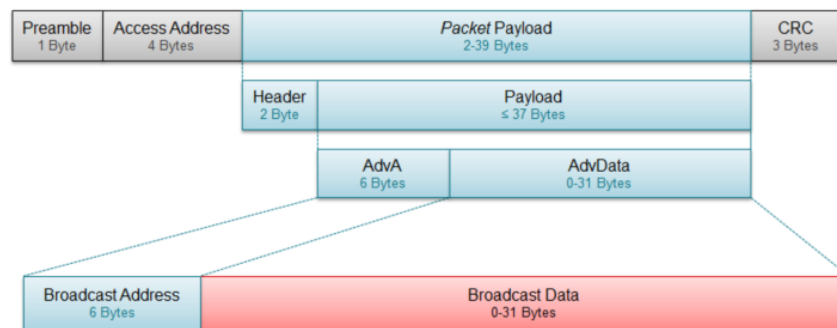
**Abbildung 23:** PacketPayload Payload

Abbildung 23 zeigt den Aufbau des Payloads. Es ist zu sehen, dass es aus 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 folgender Tabelle 2 formatiert werden [?]:

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

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. der RSSI-Wert. Was dieser RSSI-Wert ist und wie er zustande kommt wird im nächsten Kapitel erläutert [?].

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

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 [?].

Drahtlose Kommunikation erfolgt über das Senden von Elektromagnetischen Wellen. Trifft eine solche Elektromagnetische Welle auf eine Antenne, so führt dies zu einer Selbstinduktion, welche gemessen werden kann. Die Messung dieser Selbstinduktion führt zum RSSI-Wert. Dieser Wert muss abhängig von der jeweiligen Anwendung interpretiert werden, da diverse Faktoren bezüglich Transmitter, Receiver und Übertragungsmedium diesen 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 in selbigem Raum. Wie Major und Minor vergeben wurden, wird im nächsten Kapitel beschrieben.

4.3.6 Major Minor Vergabe

Um die Major Minor Nummern mit dem Dōjō zu benutzen, wurden sie im Verlaufe des Projektes standardisiert. Es sind Bereiche definiert für die speziellen Funktionen der Software. Die Sprachauswahl funktioniert über solche Nummern. Beacons mit diesen Speziellen Nummern lösen auf dem Dōjō entsprechende Funktionen aus. Auf Abbildung 24 sind die ersten Definitionen beschrieben. Die Nummernräume sind so gestaltet, dass es genug Platz für zusätzliche Funktionen vorhanden ist, wie zum Beispiel für weitere Sprachen oder Zugangskontrollen.

In Abbildung 25 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.

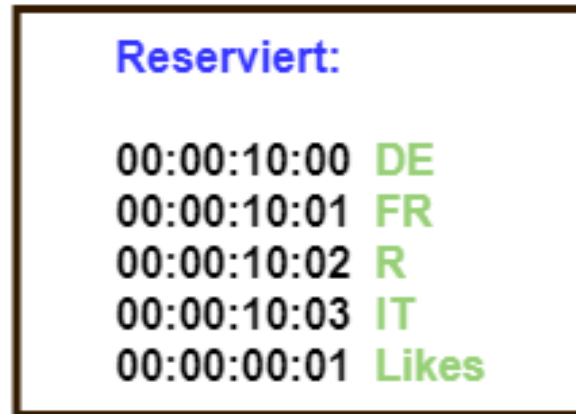


Abbildung 24: Default Definitionen von speziellen Major Minor Nummern.

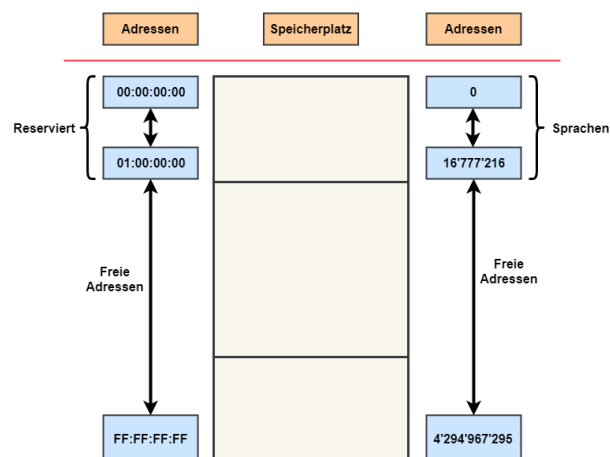


Abbildung 25: Definition der Major Minor Vergabe

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. Darum wurde ein eigenes Modul geschrieben. Das letzte Kapitel beschreibt noch die benötigten Files auf der SD-Karte um die Funktionen des Dōjō zu gewährleisten.

4.4.1 `fatfs`

Loosli?

4.4.2 SD-Karte Modul

Anschliessend sind die verschiedenen Funktionalitäten aufgelistet.

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 Mikrokontroller angeschlossen. Die Pins für den Bus lassen sich im Header-File definieren.

Lookup

Diese Funktion ist die eigentliche Implementierung des in Abbildung 15 beschrieben Algorithmus.

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.

4.4.3 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 `.txt` Datei. Sie sollte leer sein. Des weiteren müssen die Lookup-Files vorhanden sein. Sie sollten dem in Abbildung 26 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.

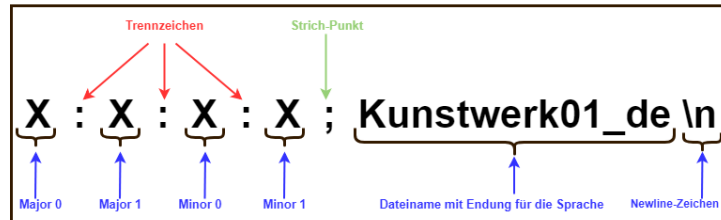


Abbildung 26: Formatdefinition Lookup-File

Wav files Loosli?

4.5 Audio, PWM

Um ein Audiosignal abspielen zu können, werden zwei PWM-Signale benötigt wie in Abbildung 27. 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 zu Verfügung, die es ermöglicht ein PWM zu erzeugen. Der NRF52 bietet vier PWM Instanzen mit je vier Kanälen. Für die Audioausgabe wurden zwei PWM Instanzen mit je einem Kanal genutzt. Im folgenden Abschnitt wird erklärt wie ein PWM Signal eines Kanales generiert wurde.

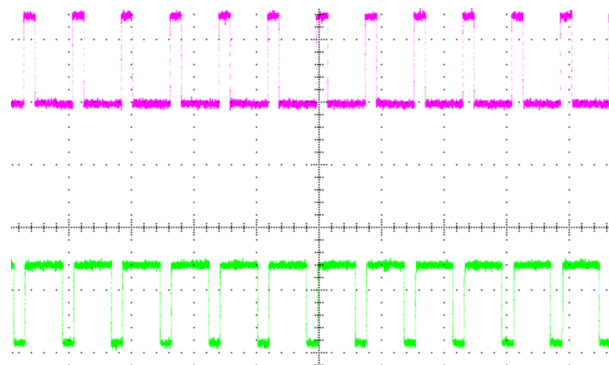


Abbildung 27: PWM-Ausgang des NRF52. Das violette Signal ist das invertierte grüne Signal.

In der Abbildung 28 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 wav file Daten abgelegt. Die Funktion `complex_playback` generiert am Ausgang anhand diesen Sequenzen ein PWM Signal.

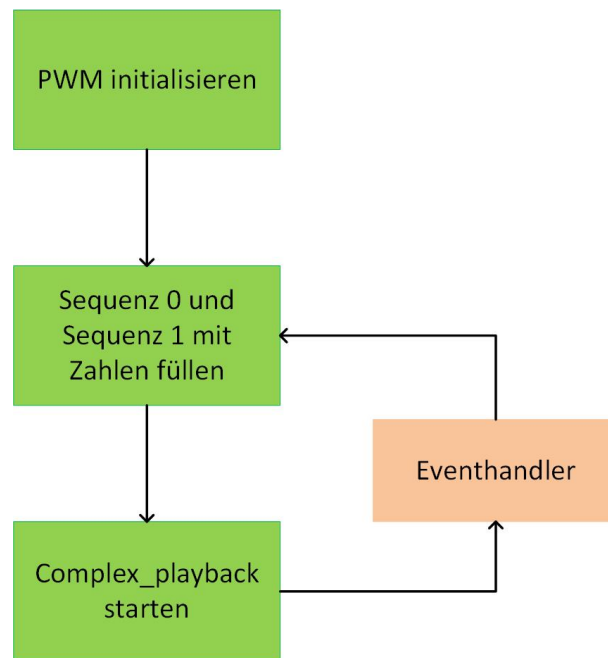


Abbildung 28: Ablauf PWM

PWM Initialisieren

Definieren der Abtastfrequenz -Welche PWM Instanz

-Config

*ouputPins

*clock

*zählermodus

*top value

*Kanal modus

*modus

-event handler

Sequenzen befüllen

Daten in sequenzen füllen, sequenz 1 muss noch ein Datenwert dazugerechnet werden. verweisen auf die SD Kart lesefunktion.

Complex playback

Die Funktion generiert anhand der Sequenz eine PWM-Signal. Der Vorteil zu simple playback ist das 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.

Beschreibung	Funktion	Argumente
PWM initialisieren	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

4.6 Lizenzen

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 9Ω angeschlossen, wobei gemäss Berechnung 5.1 ein Entladestrom von rund $400mA$ resultierte.

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

Während dem Entladevorgang wurde stets die Spannung überwacht, wobei die Spannung von $3.7V$ auf bis $2.5V$ absank. Nach dem 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 Kurzschluss test durchgeführt, wobei hier der Schwellenstrom gemäss Datenblatt bei 4.8 liegt. Gemäss dem $U = R \cdot I$ Gesetz, wurde ein Widerstand der Grösse von $700m\Omega$ verwendet damit der Grenzwert überschritten wird. Auch bei diesem Versuch, regelte das PCM den hohen Entladungsstrom ab und schaltete die Versorgungsspannung der Batterie ab.

Ladeschaltung der Batterie

Für die Ladeschaltung der Batterie wurde (wie bereits im Kapitel ?? Abschnitt Schutz einrichtungen erwähnt) ein Lade-IC verwendet. Dieser reguliert zuerst die Spannung wobei nach Erreichung des Schwellenwertes von $4.2V$ den Strom auf $0A$ herunter reguliert. Dieser Vorgang wurde während einem gesamten Ladevorgang der Batterie beobachtet und dokumentiert. 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 ($\hat{=}4.5h$) ersichtlich was die effektiven Ladewerte mittels induktiver Ladung deutlich unterbietet. Die nachfolgende Abbildung 29 zeigt die Regulierung der Spannung (blaue Kurve) wie auch die Regulierung des Stromes (rote Kurve) in Abhängigkeit der Zeit in Minuten.

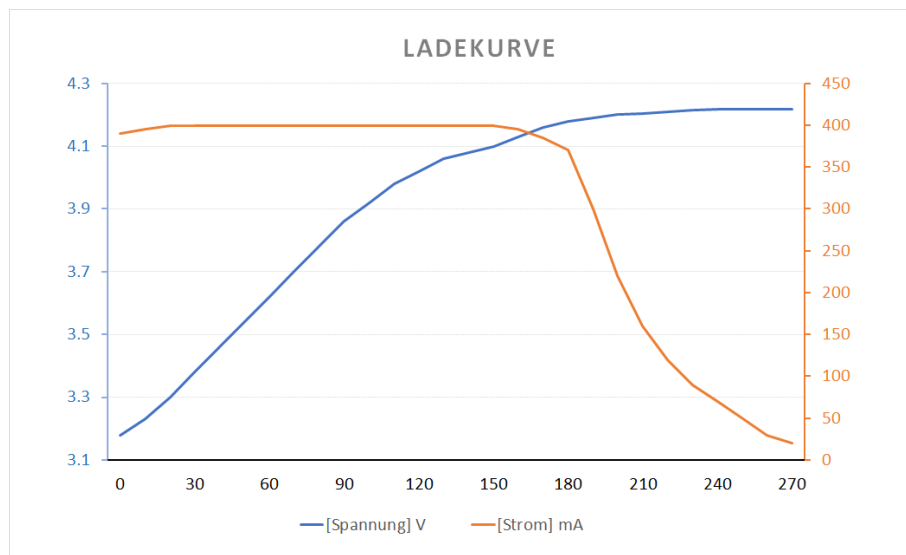


Abbildung 29: Ladekurve Lade-IC

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 30 die Abhängigkeit zwischen induziertem Strom und Spannung in Abhängigkeit zur Distanz z zwischen den Induktionsspulen.

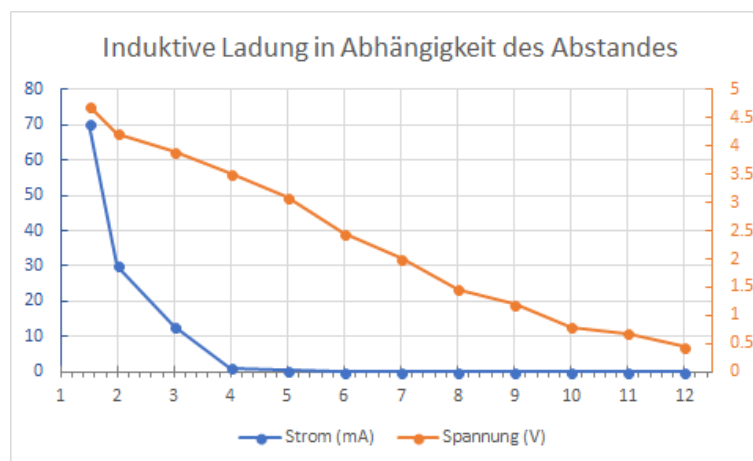


Abbildung 30: Induzierter Strom in Abhängigkeit der Distanz

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 Erkenntnisse 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 $70mA$, wobei eine Ladezeit von XX Stunden resultiert. Diese XX Stunden wurden während einem Laborversuch errechnet, wobei sich dieser in zwei Ladezyklen unterteilt. Beim ersten Ladezyklus 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 kann wie in nachfolgender Berechnung 5.2 linear hochgerechnet werden.

$$t_{charge_1} = \frac{1.1V}{\left(\frac{XXV}{XXMinuten}\right)} = XXMinuten = XXStunden \quad (5.2)$$

Für die Spannungsregelung sind somit XX Stunden notwendig. 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. Es ergab sich hierbei eine Zeit von:

$$t_{charge_2} = XXStunden \quad (5.3)$$

Der gesamte Ladezyklus ($t_{charge} = t_{charge_1} + t_{charge_2}$) benötigt somit XX Stunden um die Batterie vollständig aufzuladen.

5.2 Testkonzept Software

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

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 kleine Batterien umgesetzt, die im Bodenbereich des Dōjō angebracht werden. Das spezielle daran ist, dass sie induktiv geladen werden. Dadurch kann jeglicher „Kabelsalat“ zwischen dem Gerät und Ladestation vermieden werden. Ein kleine Einbuchtung am Boden des Dōjō ermöglicht das einfache aufstecken und laden. Aus diesem Grund wurde auf eine Implementierung einer USB-Connection verzichtet.

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 jeglicher 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 alle 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] Nordic Semiconductor, “nRF52832 Chip,” Website, unknown, online erhältlich unter http://www.nordicsemi.com/node_206/node_305/Product-Related-News/Nordic-nRF52832; abgerufen am 27. April 2018.
- [2] —, “nRF52 Development Kit,” Website, unknown, online erhältlich unter <https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52-DK>; abgerufen am 27. April 2018.
- [3] 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.
- [4] N. Peter, “Allgemeine Elektrotechnik,” 2009, unpublished thesis.
- [5] Wikipedia, “Knochenleitung,” Website, 2018, online erhältlich unter <https://de.wikipedia.org/wiki/Knochenleitung>; abgerufen am 26. April 2018.
- [6] 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.
- [7] —, “Dokumentation,” Website, Unknown, online erhältlich unter <http://infocenter.nordicsemi.com/index.jsp>; 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	Prototyp Ladestation Frontansicht	11
6	Prototyp Ladestation Draufsicht	12
7	Prototyp Ladestation Ansicht von Unten	12
8	Blockschaltbild Energiespeicherung	14
9	nRF52832 Microcontroller	14
10	nRF52 Development Kit	15
11	Knochenschallaktor [3]	16
12	Software:Layers	17
13	Software:Pollend	18
14	Statemachine-Diagramm	19
15	Statemachine: lookup	20
16	Statemachine: Special Beacon	21
17	Statemachine: Change Language	22
18	Statemachine: Wait	24
19	Statemachine: New Beacon	25
20	Statemachine: ADC Battery	25
21	Statemachine: Charge	26
22	PacketPayload Header	28
23	PacketPayload Payload	29
24	Software:Definierte MM	31
25	Software:MM Vergabe	31
26	Formatdefinition Lookup-File	33
27	PWM-Ausgang des NRF52	33
28	Ablauf PWM	34
29	Ladekurve Lade-IC	37
30	Induzierter Strom in Abhängigkeit der Distanz	37

A Anhang

A.1 Messresultate

bcsjdbcsjbvsjvbs