

Technische Universität Berlin
Fakultät IV Elektrotechnik und Informatik
Institut für Energie- und Automatisierungstechnik
Fachgebiet Elektronische Systeme der Medizintechnik
Leitung: Prof. Dr.-Ing. Dorothea Kolossa
Betreuer: Dipl.-Ing. Lars Paasche

Abschlussbericht

Projekt Elektronik BETH

Badminton Equipment for Training and Health

25. Februar 2024

Inhaltsverzeichnis

1 Einleitung	6
2 Gehäuse	8
3 Audio	11
3.1 Eingliederung in das Projekt	11
3.2 Funktionsweise	11
3.2.1 DF Player	11
3.2.2 Addierer und Rekonstruktionsfilter	12
3.2.3 Audio Verstärker	13
3.2.4 DC/DC Wandler	14
3.3 PCB	15
3.4 Programmierung	15
3.5 Reflektion der Arbeit	17
4 Energieversorgung	18
4.1 Eingliederung in das Projekt	18
4.2 Funktionsweise	19
4.3 Verwendete Baugruppen	20
4.3.1 Batteriezelle	20
4.3.2 Charge IC	20
4.3.3 Step-up DC-DC Converter	21
4.3.4 eFuse	21
4.3.5 Ein-Aus-Schalter Komponenten	22
4.3.6 Protection IC	23
4.4 Schaltplan und Leiterplatte	24
4.4.1 Schaltplan	24
4.4.2 PCB	25
4.5 Testen des Moduls	25
4.6 Reflektion der Arbeit	26
5 LED Beleuchtung	28
5.1 Eingliederung in das Projekt	28
5.2 Funktionsweise	29
5.3 Schaltplan	30
5.4 PCB	32

5.5	Programmierung	33
5.6	Reflexion der Arbeit	33
6	Sensorik	35
6.1	Eingliederung in das Projekt	35
6.2	Anforderungen	35
6.3	Funktionsweise	36
6.3.1	TOF Sensoren	36
6.3.2	Touch Sensor	37
6.4	Programmierung Interrupt	38
6.5	Schaltplan	39
6.6	Reflexion der Arbeit	41
7	Kommunikation	43
7.1	Eingliederung in das Projekt	43
7.2	Anforderungen	43
7.3	Implementierung	44
7.3.1	Raspberry Pi Programmierung	44
7.3.2	Graphical User Interface	46
7.3.3	ESP-32 Programmierung	48
7.4	Test der Kommunikation	52
8	Fazit	54

Abbildungsverzeichnis

1	BETH, bestehend aus der Basisstation (Raspberry Pi) und vier Satelliten	7
2	Mit einem CAD-Programm gezeichneten Bestandteile des Satellitengehäuses (ausser Status-LED-Fenster)	8
3	Unterteil des Satelliten mit eingesetzter Brücke (links) und Ansicht von oben auf den Satelliten mit Touch- und ToF-Sensoren (rechts)	9
4	offenes Gehäuse mit Hauptplatine des Satelliten	10
5	Satelliten im Spielmodus	10
6	DFPlayer	12
7	Filter und Addierer	13
8	LM386 - Audio Verstärker	14
9	PEME2-S5-S12-S	14
10	Hauptplatine	15
11	Black Box Modell des Energieversorgungsmoduls	19
12	Schematischer Aufbau des Energieversorgungsmoduls	20
13	Versuchsaufbau des Schalters mit ESP32	23
14	Schaltplan des Energiemoduls	24
15	Energiemodul	25
16	Schematischer Aufbau des LED-Moduls	30
17	WS2812B Pin-Konfiguration (aus Zedfy, o. D.)	31
18	Schaltplan der LED-Gruppe	32
19	Schematischer Aufbau der Verschaltung	36
20	TOF Sensor (aus AZ-Delivery, o. D.)	37
21	Touch Sensor	38
22	Plazierung der Sensoren im Satelliten	39
23	Vollständiger Schaltplan der Sensoren	40
24	TOF Sensor Schematic	41
25	Touch Sensor Schematic	41
26	Main Menu des GUIs	46
27	Trainings Menu	47
28	Protokoll	47
29	Ablaufdiagramm der Kommunikation	49

Acronyms

BMS	Battery Management System
TI	Texas Instruments
CCCV	Constant Current Constant Voltage
PCB	Printed Circuit Board

1 Einleitung

Die zunehmende Integration von Elektronik in unserem täglichen Leben eröffnet faszinierende Möglichkeiten für technologische Innovationen. Der Unikurses "Projekt Elektronik" hat nicht nur den klassisch akademischen Anspruch theoretisches Wissen zu vermitteln, sondern bietet auch eine einzigartige Gelegenheit, elektrotechnisches Fachwissen in praxisnahen Fällen anzuwenden und fordert die Anwendung ingenieurwissenschaftlicher Arbeitspraxisen und das Erweitern des Fähigkeitenhorizontes. In diesem Bericht steht ein Elektronikprojekt im Fokus, das verschiedene Aspekte der Elektrotechnik umfasst und dabei die Programmierung von Mikrocontrollern, die Analyse und Umsetzung der Energieversorgung kleinerer Geräte sowie die Implementierung von verschiedener Sensoren sowie Optischer- und Akustischerbausteine einschließt.

Das Ziel ist es ein System zu entwickeln, dass sowohl in sportlichen Felder sowie im Freizeitbereich Einsatz findet. Das hier entwickelte und vorgestellte BETH (Badminton Equipment for Training and Health) soll dem Spieler die Möglichkeit geben seine Reaktionsfähigkeit, Bewegungskoordination und Ausdauer zu verbessern, indem es eine dynamische und unterhaltsame Umgebung schafft.

Damit bietet das Projekt eine einzigartige (durchaus auch spielerische) Grundlage, theoretisches Wissen der Elektrotechnik in die Praxis umzusetzen. Durch die Programmierung von ESP32 Mikrocontrollern und eines Raspberry Pi's werden nicht nur grundlegende Kenntnisse in der Softwareentwicklung vertieft, sondern auch die Fähigkeit entwickelt, komplexe elektronische Systeme zu entwerfen und zu steuern. Die ESP32 Mikrocontroller stellen dabei das Herzstück sogenannter Satelliten dar. Diese Einheiten werden durch einen Akku mit Energie versorgt und beinhalten eine Sensoreinheit genauso wie optische und akustische Bestandteile.

Die Analyse des Energiekonsums und Umsetzung der Energieversorgung kleinerer Geräte sind ebenfalls Bestandteil des Projektes. Es soll eine effiziente, möglichst kostengünstige und anwenderfreundliche Lösung entwickelt werden.

Die gezielte Integration von Sensoren sowie Licht- und Tonbausteinen in das Projekt eröffnet die Möglichkeit, innovative und interaktive Systeme zu entwickeln. Diese Komponenten spielen eine zentrale Rolle für die Handhabung des Systems sowie den Anwendungsspaß über verschieden Einsatzgebiete hinweg.

Darüber hinaus ermöglicht die Vielfalt der in diesem Projekt behandelten Bereiche eine ganzheitliche Vertiefung der elektrotechnischen Kenntnisse und Arbeiten. Von der Softwareentwicklung über die Energieversorgung bis hin zur Integration von Sensoren und Aktoren werden verschiedene Themenbereiche abgedeckt. Dies fördert nicht nur ein breites Verständnis für elektrotechnische Zusammenhänge, sondern schult auch die

Fähigkeit, in interdisziplinären Teams zu arbeiten und komplexe Probleme zu lösen. Insgesamt bietet die Durchführung dieses Elektronikprojekts eine spannende Möglichkeit, elektrotechnisches Wissen praxisnah anzuwenden, ingenieurtechnische Fähigkeiten zu trainieren und verschiedene Bereiche der Elektrotechnik zu erkunden. In diesem Bericht werden die einzelnen Aspekte des Projekts ausführlich dargestellt und die erzielten Ergebnisse und technologischen Fortschritte präsentiert.



Abbildung 1: BETH, bestehend aus der Basisstation (Raspberry Pi) und vier Satelliten

2 Gehäuse

Das BETH-System besteht aus einer Basisstation und vier Satelliten (siehe Abb. 1). Für die Basisstation, die sich aus einem Raspberry Pi und einem Touchdisplay zusammensetzt, wurde ein käuflich erhältliches Gehäuse verwendet. Mit diesem Gehäuse kann die Basisstation so hingestellt werden, dass das Display durch einen Spieler oder eine Spielerin gut bedienbar und auch gut sichtbar ist.

Anders als die Basisstation musste für die Satelliten eine Gehäuse entworfen und hergestellt werden. Nach Abwägen der gegebenen Möglichkeiten wurde sich für ein additives Herstellungsverfahren entschieden. Das bedeutet, dass die Bestandteile des Satellitengehäuses mit einem CAD-Programm (*Autodesk Inventor*) gezeichnet (siehe Abb. 2) und anschließend mittels eines 3D-Druckers (*Ultimaker S5*) ausgedruckt wurden. Diese Herangehensweise ermöglichte eine flexible und hohe Gestaltungsfreiheit.

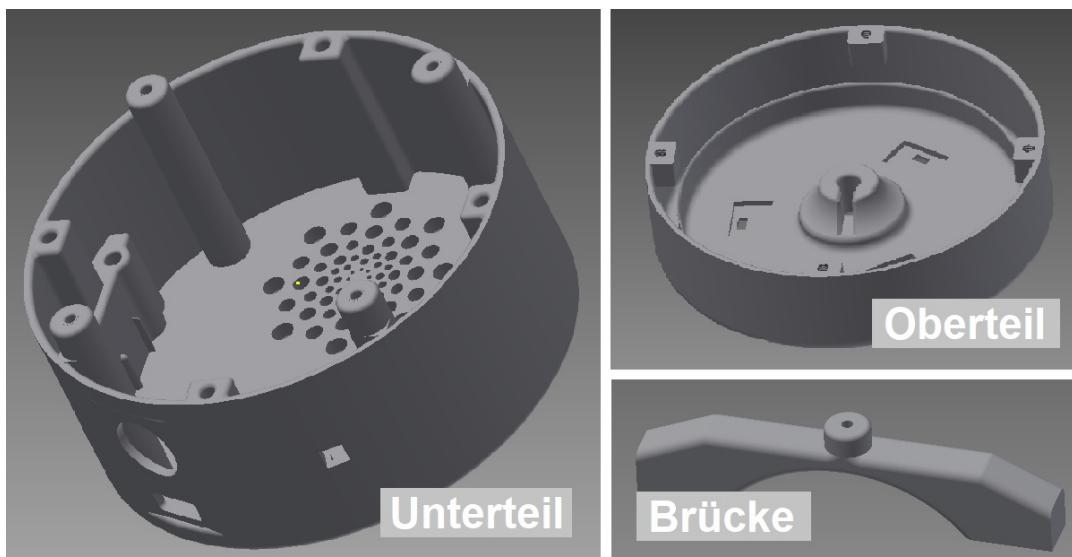


Abbildung 2: Mit einem CAD-Programm gezeichneten Bestandteile des Satellitengehäuses (ausser Status-LED-Fenster)

Nach einigen Probedrucken mit verschiedenen Filamenten, wurde für das Unterteil und die Brücke *Tough PLA* und für das Oberteil und das Status-LED-Fenster *CPE+* verwendet. Diese beiden Kunststoffe haben die Eigenschaften, dass sie formstabil und flexibel sind, so dass die Satelliten auch einen robusteren Spieleinsatz standhalten. Für die Verwendung von *CPE+* kam noch hinzu, dass dieses Material das harte Licht der LEDs streut, so dass ein schönes warmes Licht entsteht. Erwähnenswert ist auch, dass sich sowohl *Tough PLA* als auch *CPE+* im Nachgang gut bearbeiten ließen.

In dem Unterteil des Gehäuses wurden im Boden Öffnungen für einen Lautsprecher (sie-

he Kap. 3) und in die Seite des Gehäuses Öffnungen für einen Taster, eine USB-C-Buchse und eine Status-LED eingearbeitet. Weiterhin sind in dem Unterteil Auflageflächen für eine Platine mit Bauteilen für die Spannungsversorgung (siehe Kap. 4) und eine Platine mit LEDs (siehe Kap. 6) geschaffen worden. Um eine sichere Schraubverbindung zwischen Platinen und Kunststoffgehäuse zu erhalten, wurden M3-Schmelzgewinde verwendet (siehe Abb. 3).

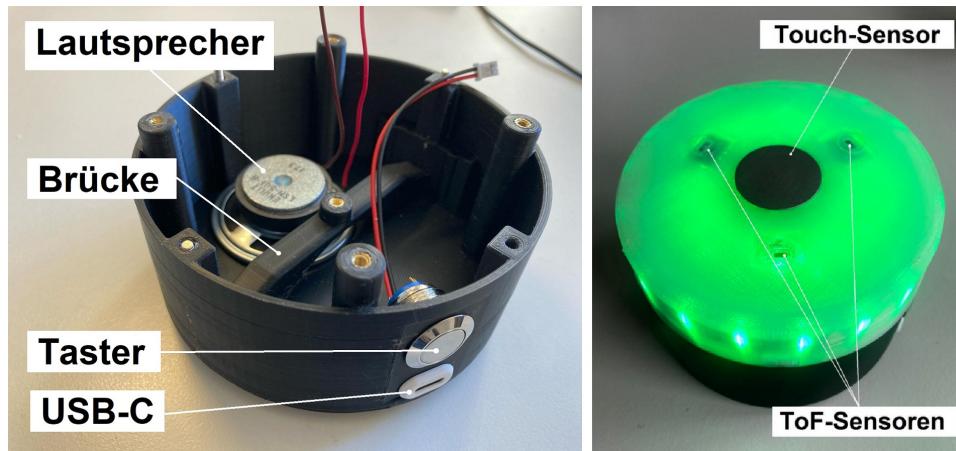


Abbildung 3: Unterteil des Satelliten mit eingesetzter Brücke (links) und Ansicht von oben auf den Satelliten mit Touch- und ToF-Sensoren (rechts)

Nach der Montage des Lautsprechers und des Akkuhalters auf dem Boden des Unterteils wurde die Brücke mittels einer Führungsnuß in das Gehäuse eingesetzt. Die Brücke wurde als Einzelteil angefertigt, damit man sie bei Bedarf zu einem späteren Zeitpunkt wieder heraus nehmen kann. Zwischen der unteren Platine und der LED-Platine befindet sich die Hauptplatine des Satelliten (siehe Abb. 4). Diese wird mit M3 x 17mm-Abstandshalter auf die untere Platine geschraubt.

In das Oberteil des Gehäuses sind ein LED-Streifen, ein Touchsensor und drei Abstands-sensoren verbaut (siehe Abb. 3). Die Platinen, auf denen sich die Abstandssensoren befinden, sind von unten in das Oberteil, mit einer kleinen Öffnung nach oben für den eigentlichen Sensor, eingeklebt. Die drei Abstandssensoren wurden mit einer runden Plexiglasscheibe abgedeckt. Die Plexiglasscheiben sind genau wie der Touchsensor planar in das Oberteil eingelassen, so dass eine glatte Oberfläche entsteht und keine Unebenheiten beim Berühren des Gehäuses stören. Unterhalb des Touchsensors auf der Innenseite des Oberteils befindet sich ein stumpfer Kegel. Dieser liegt auf der LED-Platine auf und ist über einen Abstandshalter mit der Brücke auf dem Boden des Unterteils verbunden. Durch diese Stützstruktur wird die mechanische Belastung beim Berühren des Touchsensors auf das gesamte Gehäuse verteilt.



Abbildung 4: offenes Gehäuse mit Hauptplatine des Satelliten

Nach diversen Anpassungen, ist es gelungen, dass die Gehäuse der Satelliten zum einen robust sind und zum eine kompakte Bauform haben. Durch die glatten Oberflächen besitzen die Satelliten eine angenehme Haptik und durch das Lichtspiel der LEDs bieten sie zusätzlich ein visuell ansprechendes Aussehen (siehe Abb.5).



Abbildung 5: Satelliten im Spielmodus

3 Audio

3.1 Eingliederung in das Projekt

Die Umsetzung einer effektiven Audiowiedergabe erfordert eine präzise Signalverarbeitung und Kontrolle. In diesem Kontext liegt der Fokus auf der Audiowiedergabe, bei der mehrere Schaltungselemente kombiniert wurden, um optimale Klangqualität und Steuerbarkeit zu gewährleisten. Ursprünglich war die Implementierung von Equalizern, Cross Fadern und Crossover-Elementen geplant. Später wird es entschieden, die nicht zu realisieren. Die Grundlage bildet ein Audioplayer-Modul, dem im Verlauf der Realisierung verschiedene Elemente hinzugefügt wurden.

Zu Beginn wurde das Stereosignal des Audioplayer-Moduls, bestehend aus den Kanälen links und rechts, mithilfe eines OPV-Addierers kombiniert, um eine Monowiedergabe auf einem einzelnen Lautsprecher zu ermöglichen. Anschließend wurde das kombinierte Signal einem Rekonstruktionsfilter unterzogen, um mögliche Störeinflüsse zu minimieren und eine saubere Klangreproduktion zu gewährleisten. Für eine zusätzliche Signalverstärkung wurde eine OPV-Schaltung integriert, um die Audiosignale weiter zu optimieren. Der Benutzer hat die Möglichkeit, die Lautstärke bequem mit zwei Tastern anzupassen. Abschließend erfolgte die Stromversorgung der OPV-Schaltung durch einen DCDC-Wandler mit einer Spannung von +12V. Diese durchdachte Abfolge von Schaltungselementen schafft ein robustes System für eine qualitativ hochwertige und benutzerfreundliche Audiowiedergabe.

Die folgenden Abschnitte werden detailliertere Einblicke in die einzelnen Komponenten und Schaltungselemente geben, um ein umfassendes Verständnis für die Struktur und Funktionsweise dieses Teil-Projekts zu vermitteln.

3.2 Funktionsweise

3.2.1 DF Player

Zuerst wurde ein DF Player bzw. ein Audioplayer-Modul (das von der Firma DFRobot hergestellt wird) verwendet. Es ist speziell für die Wiedergabe von Audiodateien mit einem einfachen Steuerungsprotokoll konzipiert. Der DF Player ist darauf ausgelegt, Audiodateien wie MP3, WAV und WMA von einer micro SD-Karte abzuspielen und kann über einfache Befehle gesteuert werden, die über eine serielle Schnittstelle (UART) von dem Mikrocontroller (hier ESP32S) gesendet werden. Es unterstützt Funktionen wie

das Abspielen, Pausieren, Stoppen, Vor- und Zurückspulen von Audiotracks. Diese Funktionen können über die Steuerschnittstelle gesteuert werden. Außerdem ist das Modul relativ klein und einfach zu integrieren, was es ideal für das Projekt war, bei denen eine Audiowiedergabe erforderlich ist.

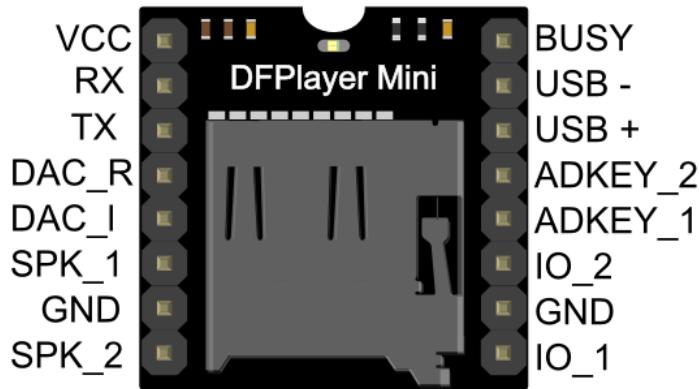


Abbildung 6: DFPlayer

Das DF Player verfügt über zwei Audiokanäle, die eine Stereowiedergabe ermöglichen (Audio links und rechts). Da jedoch in diesem speziellen Projekt nur ein Lautsprecher zur Verfügung stand, war es notwendig, die Signale zu addieren. Aus diesem Grund wurde die Addierer-Schaltung implementiert, die im folgenden Abschnitt näher erläutert wird.

3.2.2 Addierer und Rekonstruktionsfilter

Um die rechten und linken Stereosignale zu kombinieren, ist eine Schaltung mit einem Operationsverstärker (OPV) erforderlich, womit man die Stereo-Mitte-Signale bekommen kann. Für Audiosignale wird ein OPV mit geringem Rauschen bevorzugt. Ein geeigneter Operationsverstärker für unser Projekt ist der NE5532, der genau solche Eigenschaften aufweist. Da wir nur Unipolar Versorgung haben, müssen wir ein nicht invertierende Addierer nutzen, wo man die beiden Eingang- bzw. Audiosignale bekommt. und mit Hilfe von dem genutzten Potentiometer kann man den Verstärkungsrat bestimmen und um die Signale später je nach unserer weiteren Schaltungen zu kalibrieren haben wir einen Potentiometer dafür genutzt.

Beim digitalen Audio-Signalen während des Digitalisierungsprozesses braucht man einen Rekonstruktionstiefpassfilter, wo er die Eingangssignale "Glättet", da durch die An-

wendung eines Tiefpassfilters die Signalform des digitalen Signals geglättet wird. Das bedeutet, dass schnelle Änderungen oder Spitzen im Signal reduziert werden, um eine gleichmäßige und kontinuierliche Signale zu erhalten. Da die Audio-Signale hat maximale Frequenz von 20 kHz und der Eingang hat den Takt von 80 MHz würde es uns passen ein Tiefpass zweite Ordnung mit dem Grenzfrequenz von 10 kHz mit einem Low noise OPV (NE5532).

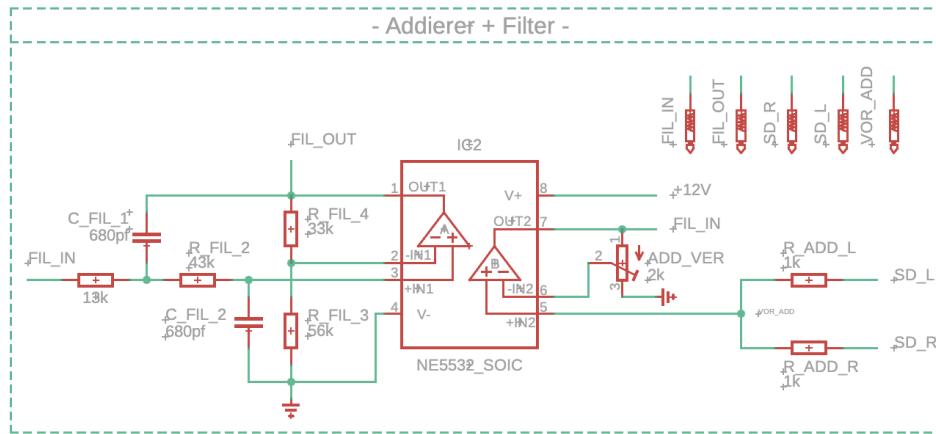


Abbildung 7: Filter und Addierer

3.2.3 Audio Verstärker

Der LM386 Audioverstärkerchip stellt eine häufig verwendete Lösung in der Welt der einfachen Audioschaltungen dar. Die Schaltung umfasst verschiedene Bauteile, die in ihrer Kombination eine effektive Verstärkung von Audiosignalen ermöglichen.

Die Verdrahtung der Schaltung beinhaltet den Anschluss des Audiosignals an Pin 3 des LM386, während Pins 4 und 6 mit der Masse des Gesamtsystems und die Spannungsversorgung verbunden werden.

Die pins 1 und 8 sind für die Einstellung von der maximalen Verstärkung die Signale stehen aber die würden nicht gebraucht, weil mit max gain von 20 kann man die Grenze von der Spannung schon erreichen und das passiert auf der Grund das wir schon große Eingangssignal haben.

Das verstärkte Audioausgangssignal wird am Pin 5 abgegriffen und mit einem Lautsprecher durch ein großen Elko, der die niedrige frequenzen Störsignale und die DC Anteile die Ausgangssignal herausfiltert, verbunden.

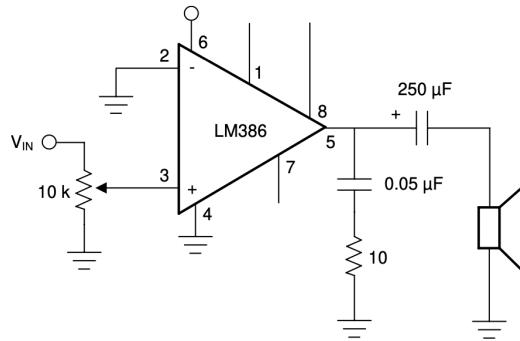


Abbildung 8: LM386 - Audio Verstärker

3.2.4 DC/DC Wandler

Für die Spannungsversorgung der OPVs wurde ein DC/DC-Wandler eingesetzt, um die Batteriespannung von $+5V$ auf $+12V$ umzuwandeln. Aufgrund des geringen Stromverbrauchs der OPVs, der nicht mehr als $100mA$ beträgt, wurde ein $5W$ Wandler ausgewählt. Der gewählte Wandler, PEME2-S5-S12-S, hat eine Leistung von $2W$ und kann einen Strom von $167mA$ liefern.

Um die Eingangs- und Ausgangswelligkeit weiter zu minimieren, wurde ein Filterkondensator an den Eingangs- und Ausgangsklemmen angeschlossen. Es ist wichtig sicherzustellen, dass die Kapazität des Kondensators geringer ist als die maximale kapazitive Last des Modells, um Startprobleme zu vermeiden. Eine zu große Kapazität könnte andernfalls Schwierigkeiten beim Starten verursachen.

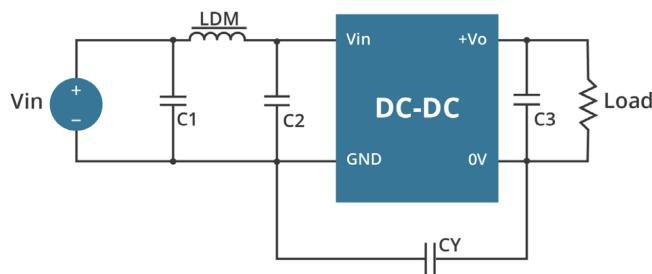


Abbildung 9: PEME2-S5-S12-S

Zusätzlich wurde für den Wandler eine Elektromagnetische Verträglichkeit (EMC) Schaltung implementiert. Dies ermöglicht es dem Wandler, in einer elektromagnetisch sensiblen Umgebung zu arbeiten, ohne Störungen zu verursachen oder selbst von anderen Geräten beeinträchtigt zu werden.

3.3 PCB

Um Störgeräusche zu minimieren, haben wir die Audiokomponenten so weit wie möglich von der Platine entfernt positioniert. Aus diesem Grund wurde der DC-DC-Wandler im oberen Bereich der Platine angebracht. Zudem haben wir eine Möglichkeit geschaffen, die UART-Verbindung zwischen dem Controller und dem DFPlayer mittels eines Schalters zu unterbrechen, um jederzeit ein Firmware-Update des Controllers durchführen zu können.



Abbildung 10: Hauptplatine

Darüber hinaus ist die Platine mit zwei Potentiometern ausgestattet, die es erlauben, den Gain der Addierer und die Verstärkung anzupassen. Für eine vereinfachte Überprüfung der Platine wurden für alle Signale Testpads integriert.

3.4 Programmierung

Für die Audiomodulierung des DF Players wurde die Bibliothek (DFRobotDFPlayerMini) verwendet. Um den DFPlayer Mini zu steuern, muss zunächst die zugehörige Bibliothek eingebunden werden:

```
#include <DFRobotDFPlayerMini.h>
```

Nachdem die Bibliothek eingebunden wurde, wird ein Objekt der DFPlayerMini-Klasse erstellt, um die Kommunikation mit dem Modul zu ermöglichen:

```
DFRobotDFPlayerMini myDFPlayer;
```

Das Erstellen des Objekts ermöglicht es uns, über einfache Befehle auf die Funktionalitäten des DFPlayer Mini zuzugreifen und die Audiomeldung zu steuern. Der DFPlayer Mini unterstützt eine Vielzahl von Funktionen, die es ermöglichen, MP3-Dateien von der microSD-Karte abzuspielen. Beispielsweise kann eine spezifische MP3-Datei durch Angabe ihrer Nummer auf der SD-Karte abgespielt werden:

```
myDFPlayer.play(1);           // Spielt die erste MP3-Datei ab
```

Weitere Funktionen erlauben es, die Lautstärke anzupassen, die Wiedergabe zu pausieren, fortzusetzen oder zwischen Titeln zu wechseln:

```
myDFPlayer.volume(10);       // Setzt die Lautstärke auf 10 (von 30)
myDFPlayer.pause();          // Pausiert die Wiedergabe
myDFPlayer.next();           // Springt zum nächsten Titel
myDFPlayer.start();          // Setzt die Wiedergabe fort
myDFPlayer.previous();       // Geht zum vorherigen Titel
```

Die Programmierung des DFPlayer Minis ermöglicht eine flexible und anpassbare akustische Gestaltung, um das Erlebnis unseres Projekts zu verbessern. Durch die Integration dieser Programmierfunktionalitäten können Soundeffekte und Musik in Echtzeit während des Betriebs gesteuert werden, um auf bestimmte Ereignisse oder Benutzeraktionen zu reagieren. Dies verbessert nicht nur die atmosphärische Wirkung des Systems, sondern schafft auch eine interaktiver und ansprechendere Nutzererfahrung.

3.5 Reflektion der Arbeit

Die Implementierung und Eingliederung der Audiokomponente in unser Projekt verlief erfolgreich und es konnte die nachstehenden Ziele erfüllen:

- Maximalpegel der Lautstärke liegt bei 90 dB
- Impedanz des Lautsprechers beträgt bis zu 4 Ohm
- Wechsel zwischen Liedern digital und analog mittels Taster möglich
- Unterstützt die Wiedergabe von Stereo-Mittelsignalen
- Signal dank Filterung frei von Rauschen
- Niedriger Energieverbrauch

Die einzige Herausforderungen bei der Hauptplatine ergaben sich daraus, dass die SMD-Komponenten eigenhändig gelötet wurden, was gelegentlich zu Lötfehlern oder ähnlichen Problemen führte. Das erforderte, dass die Komponenten wieder eingelötet werden mussten, was zusätzliche Zeit in Anspruch nahm.

4 Energieversorgung

Im Folgenden wird kurz die Rolle des Moduls im Projekt motiviert 4.1. Anschließend werden Funktionsweise 4.2 des Energiemoduls sowie die verwendeten Komponenten erklärt. Aus diesen resultiert dann der verwendete Schaltplan und das entwickelte Printed Circuit Board (PCB) Design (Sec: 4.4). Zuletzt wird noch auf die Erkenntnisse aus dem Anwendungsfall ?? und dem Arbeitsprozess 4.6 eingegangen.

4.1 Eingliederung in das Projekt

Bei der Entscheidung die einzelnen Bestandteile des B.E.T.H. Systems mit Energie zu versorgen, standen 3 Optionen zur Wahl. Zum einen die das Betreiben über das Anschließen an eine Steckdose, das Betreiben mit einer Batterie oder die Verwendung einer Akku basierten Versorgung. Die Entscheidung fiel auf die Variante mit einem Akku, da hierdurch ein flexibler Spielaufbau, einfache Handhabung sowie eine Ästhetik der Satelliten garantiert werden.

Das Modul der Energieversorgung stellt neben dem ESP32 µController einen der beiden Grundbausteine der Satelliten dar. Aus diesem Grund ist es sinnvoll zu verstehen, wie das Modul mit dem restlichen System interagiert.

Als Inspiration für das Platinendesign wurde die Vorlage Scott, o. D. des Youtubers The Great Scott verwendet. Dabei wurden einige Modifikationen, wie beim An- und Ausschalten durchgeführt, um das Platinendesign auf die Anforderungen des Projektes anzupassen. Das Modul hat drei verschiedene Eingänge. Dabei ist einer rein zum Laden des Lithium-Ionen Akkus und die beiden anderen zum An- bzw. Ausschalten des Moduls. Einer dieser Eingänge ist mit dem ESP Mikrocontroller, dem Herzstück und Steuerungseinheit des Satelliten, verbunden und der zweite mit einem Taster, der an der Wand des Gehäuses angebracht ist. Zu beachten ist, dass nur über den Taster der Satellit ein- und ausgeschaltet und über den ESP nur ausgeschaltet werden kann. Ferner hat das Energiemodul zwei Typen von Ausgängen. Neben dem offensichtlichen, den Ausgängen zur Energieversorgung auch eine weitere Verbindung zu dem Mikrocontroller, um den State of Charge der Akkuzelle zu überwachen. Abbildung 11 zeigt auch nochmal graphische Darstellung der Interaktion mit dem Energiemodul.

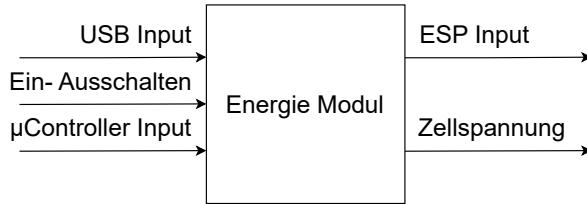


Abbildung 11: Black Box Modell des Energieversorgungsmoduls

4.2 Funktionsweise

Im Vorfeld zu dem Projekt wurden bestimmte Anforderungen an das Modul der Energieversorgung gestellt. Diese sind:

- Energieversorgung für 45 bis 60 Minuten ermöglichen.
- Aufladen des Akkus über einen USB-C Anschluss.
- Versorgungsspannung von 5 V DC bei einem Strom von maximal 1,5 A zur Verfügung stellen.
- Ein- und Ausschalten an dem Gehäuse ermöglichen.
- Abschalten des Systems durch Softwareeingabe erlauben (kein "deep sleep").
- Sicherung des Systems vor möglichen Gefahren wie Kurzschläüssen, Überspannungen, Tiefenentladung, etc.

Da es bereits für größere und komplexere Systemen, wie Smartphones oder sogar elektrischen Autos, integrierte Lösungen gibt, ließe sich auch diese Anforderungen mit einem einzigen state of the art Battery Management System (BMS) erreichen. Um allerdings ein besseres Verständnis für die Herausforderungen im Umgang mit Akkus zu bekommen sollen die Ziele mit Hilfe von mehreren ICs, die bestimmte Teilaufgaben erfüllen, erreicht werden. Hierfür wurde der in Abbildung 12 dargestellte Plan entwickelt. Über einen USB-C Anschluss wird ein Charge IC von Texas Instruments (TI) angesteuert, welcher die Batterie mittels des bekannten Constant Current Constant Voltage (CCCV) Ladeverfahren auflädt. Hinter der Batterie befindet sich eine Schaltung, die zum einen verhindert, dass die Batterie während des Ladens belastet wird und zum anderen das Ein- und Ausschalten des Systems ermöglicht. Ein Protection IC von Xysemi wird verwendet, um das System gegen Gefahren wie bspw. zu hohe Temperaturen abzusichern. Über einen weiteren Chip von TI wird die Ausgangsspannung der Batterie auf

5 V verstärkt. Bevor die Spannung an das System freigegeben wird, wird sie durch eine eFuse, ebenfalls von TI, weiter abgesichert wird.

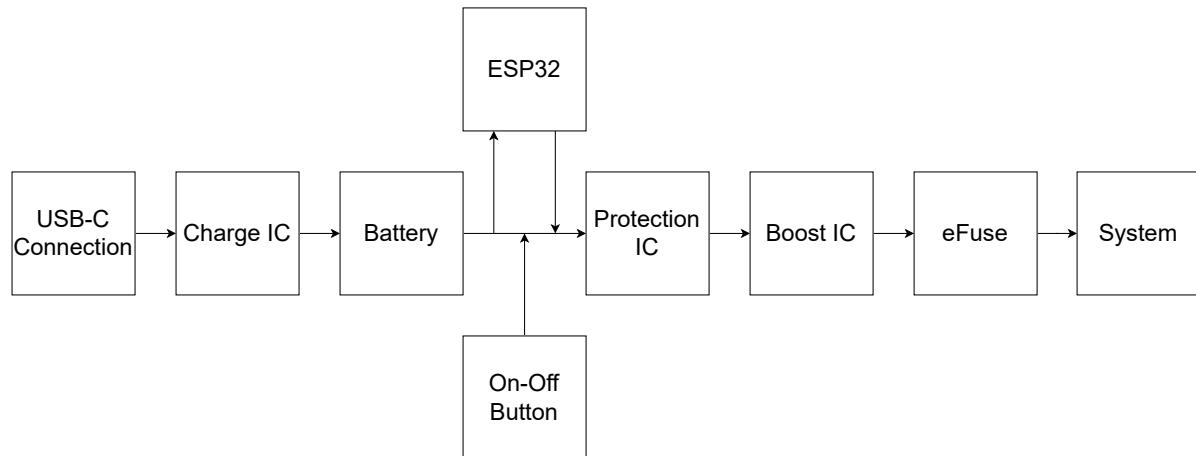


Abbildung 12: Schematischer Aufbau des Energieversorgungsmoduls

4.3 Verwendete Baugruppen

4.3.1 Batteriezelle

Um den verschiedenen Anforderungen an die Leistungsfähigkeit der Batteriezelle gerecht zu werden, wurde sich für die XTAR 18650 Industriezelle entschieden. Die Energiezelle zeichnet sich durch eine große Kapazität von 2600 mAh und einen großen Entladestrom von 4,5A aus. Als Ladeverfahren ist das CC-CV Ladeverfahren im Datenblatt Simon Gong, o. D. vorgesehen. Zusätzlich verfügt die XTAR 18650 Industriezelle über eine integrierte Schutzschaltung. Dennoch wird im Ladeschaltkreis eine eigene Schutzvorrichtung eingebaut. Die Nennspannung beträgt 3,6 V bis 3,7V und muss damit noch auf die 5V Betriebsspannung hochgeregelt werden. Bei einem Gewicht von 47g, einem Durchmesser von 18.5mm und einer Länge von 68.7mm, passen auch die nicht elektrischen Parameter zu den Anforderungen in diesem Projekt.

4.3.2 Charge IC

Zum Laden der Zelle wurde der BQ24092DGQR Texas Instruments, o. D.(a) von TI verwendet. Der BQ24092DGQR beinhaltet eine integrierte Schaltung, die für eine präzise und effiziente Ladesteuerung von 1S Lithium-Ionen-Batterien ausgelegt ist. Der Chip wurde aufgrund folgender Aspekte ausgewählt:

- Ladepräzision: Der Chip sorgt für eine präzise Steuerung des Ladevorgangs, um die Lebensdauer und Leistung des Lithium-Ionen-Akkus zu optimieren.
- Thermischer Schutz: Der integrierte Wärmeschutz schützt den Akku vor Überhitzung und erhöht die Sicherheit während des Ladevorgangs.
- Steuerung des Ladestroms: Der BQ24092DGQR ermöglicht eine präzise Steuerung des Ladestroms, was besonders wichtig ist, um den Akku effizient und sicher zu laden.
- Statusanzeigen: Der Chip kann Statusinformationen liefern, die anzeigen, ob der Ladevorgang abgeschlossen ist oder ob der Akku noch geladen wird.
- Automatische Abschaltung: Nach Abschluss des Ladevorgangs schaltet sich der BQ24092DGQR automatisch ab, um eine Überladung und die damit verbundenen Risiken zu vermeiden.

Aus dem entsprechenden Datenblatt Texas Instruments, o. D.(a) liegen sich neben diesen Funktionalitäten auch die notwendigen passiven Komponenten ermitteln und in den Schaltplan übertragen. Ferner ist in diesem Dokument ebenfalls ein Abschnitt den Layoutempfehlungen gewidmet, welche an späterer Stelle genutzt werden.

4.3.3 Step-up DC-DC Converter

Zum Hochsetzen der Spannung von 3,6 V auf 5 V wurde der TPS61085 Step-up DC-DC Converter für dieses Projekt ausgewählt. Aus dem Datenblatt Texas Instruments, o. D.(c) geht hervor dass der TPS61085 einen Eingangsspannungsbereich von 2,3 V bis 6 V und am Ausgang bis zu 18,5 V bei 3,6 A ausgeben kann. Weitere Features sind eine wählbare Schaltfrequenz 650 kHz/1,2 MHz, von der 1,2 MHz eingestellt wurden und ein Temperaturschutz vor Überhitzung. Laut Datenblatt Texas Instruments, o. D.(c) sind die häufigsten Anwendungen mobile Geräte, GPS Empfänger und Digitalkameras. Damit ist der TPS61085 zur Verwendung in den Satelliten des Projektes bestens geeignet.

4.3.4 eFuse

Um die Schaltung und den esp32 vor Kurzschlüssen, Überspannung, Spannungsspitzen und zu großen Einschaltströmen zu schützen wurde die eFuse TPS259621DDAT mit eingebaut. Als maximaler Strom wurde 1,8 A über einen Widerstand vom ILM (8) Pin eingestellt. Der Widerstand berechnet sich über die Formel $R_{ilm} = \frac{903}{I_{limit} - 0.0112}$ Texas Instruments, o. D.(b). Wenn der Ausgangsstrom das eingestellte Stromlimit überschreitet,

regelt der IC den Ausgangsstrom auf das Stromlimit herunter. Bei einem Kurzschlussstrom, welcher per Definition des Datenblattes Texas Instruments, o. D.(b) das 1.5-fache des Stromlimits beträgt, wird der Ausgangsstrom sehr schnell heruntergeregt. Das kann zu Spannungsschwankungen führen. Sollte es zur starken Entwicklung von Hitze kommen, führt der IC einen thermal-shutdown durch, sodass kein Strom mehr fließen kann.

4.3.5 Ein-Aus-Schalter Komponenten

Zum Ein- und Ausschalten des Gerätes wurde ein elektronischer Schalter gefordert, welcher sowohl durch einen Taster ein und ausgeschaltet, als auch durch den esp32 abgeschaltet werden kann. Dadurch kam ein mechanischer Kippschalter nicht in Frage. Um den Anforderungen gerecht zu werden, wurde ein CMOS J-K Flip Flop verwendet. Hier ist die Toggle Funktion, also wenn die J und K Pins bei gleichzeitig auf High sind Instruments, o. D., interessant. Deshalb werden sowohl der J als auch der K Pin auf die Batteriespannung gelegt. Um zu Schalten setzt nun der Taster oder ein Pin vom esp32 auf dem Clock eingang ein High. Damit wird unabhängig vom vorherigen Zustand der aktuelle Zustand geändert. Mit dem Flip Flop Ausgang Q wird dann ein pmos auf High oder Low gezogen, welcher das Gerät ein oder Ausschaltet.

Als pmos wurde der DMP3098L-7 gewählt Incorporated, o. D. Dieser kann einen Gleichstrom von bis zu 3.8A treiben und hat einen R_{on} von 70 mOhm. In diesem Schaltkreis werden insgesamt zwei pmos verwendet. Der erste soll dafür sorgen, dass das Gerät nicht gleichzeitig geladen und angeschaltet werden kann. Der zweite ist für das manuelle Ein- und Ausschalten zuständig.

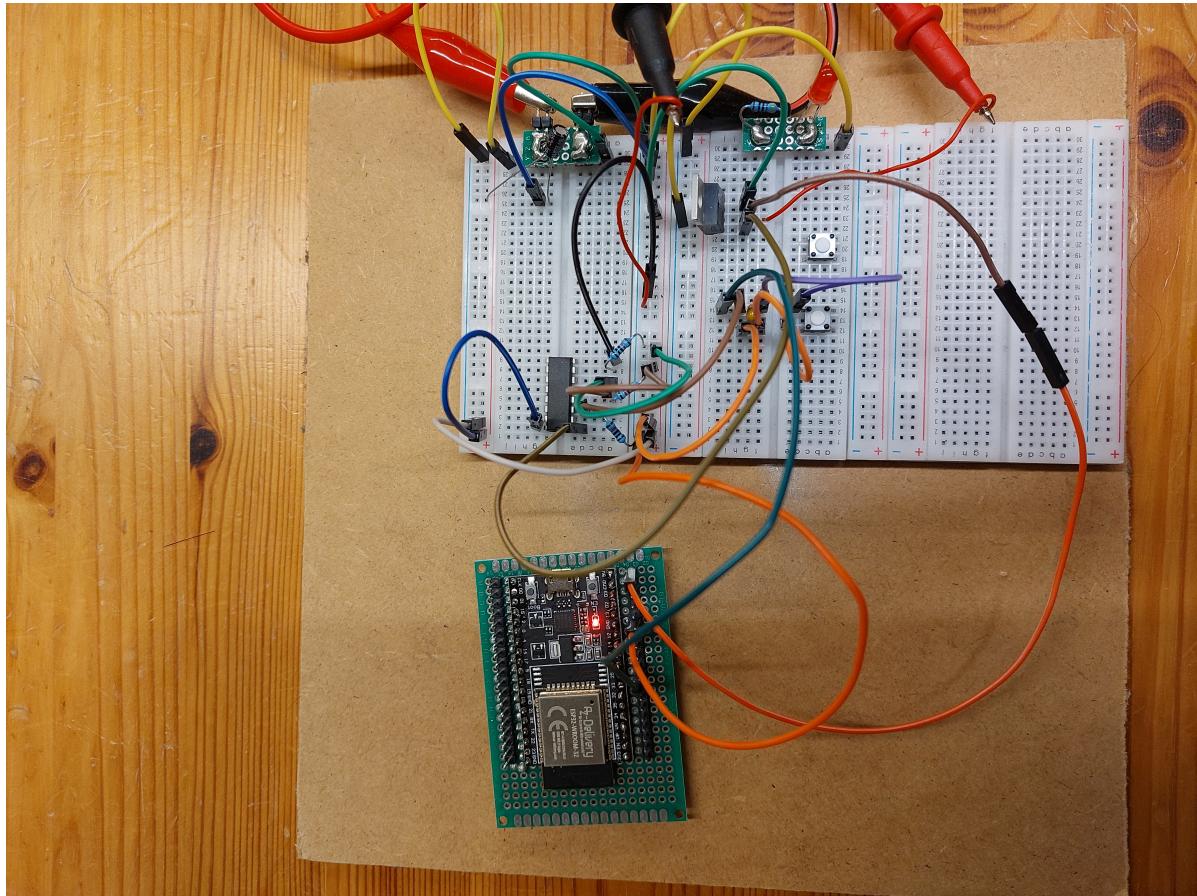


Abbildung 13: Versuchsaufbau des Schalters mit ESP32

Um die Funktionalität zu prüfen wurde die Schalter-Komponente auf dem Steckbrett aufgebaut, siehe 13. Als Ersatz für die Batterie wurde hier ein Labornetzteil mit einer eingestellten Spannung von 3,7 V verwendet. Beim Testen der Schaltung funktioniert alles wie erwartet.

4.3.6 Protection IC

Um den Li-Ion Akku beim Betrieb zu schützen wurde der Battery-Protection IC XB8089D in den Schaltkreis mit eingebaut. Im Datenblatt Xysemi, o. D. sind folgende Eigenschaften aufgelistet:

- Geringer eigener Stromverbrauch
- Geringe äußere Beschaltung notwendig
- Schutz vor Überladung und Tiefenentladung der Li-Ion Zelle

- Schutz vor zu hohen Entladeströmen, damit auch einen Kurzschlusschutz
- Temperaturüberwachung

Damit ist der XB8089D gut dafür geeignet um für das Projekt zusätzliche Sicherheit zu garantieren. Besonders bei Li-Ion Zellen sollte besonders auf die Brand- und Explosionsgefahr hingewiesen werden. Der für diesen Schaltplan verwendete IC reduziert diese Gefahr jedoch enorm durch z.B. den Kurzschlusschutz.

4.4 Schaltplan und Leiterplatte

4.4.1 Schaltplan

Die Kombination der oben aufgeführten ICs mit den passiven Elementen aus den Datenblättern der Hersteller wurde ein entsprechender Schaltplan in Fusion 360 entworfen. Ferner wurden Testpunkte hinzugefügt, die eine reibungslose Inbetriebnahme erleichtern sollen. In Abbildung 14 ist der Schaltplan des Moduls dargestellt. Die einzelnen ICs wurden mit ihren notwendigen passiven Bauteilen zusammengefasst. Die Motivation hier von ist es ein übersichtliches Dokument zu kreieren, welches ein strukturiertes Arbeiten und eine erleichterte Fehlersuche ermöglicht.

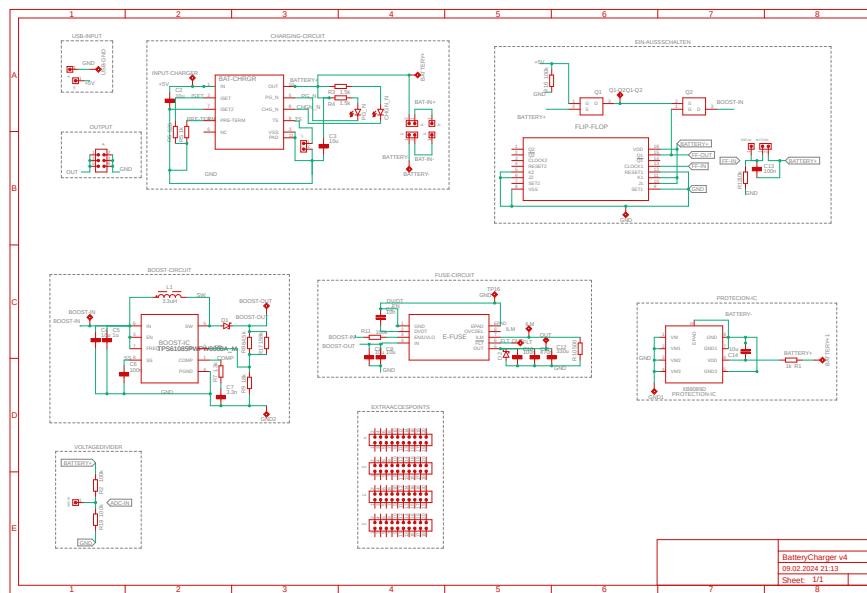


Abbildung 14: Schaltplan des Energiemoduls

4.4.2 PCB

Auf Basis des zuvor gezeigten Schaltplan wurde das PCB in Abbildung 15 generiert. Zu Beginn des Designprozesses wurden die ICs nach Herstellerangaben mit ihren jeweiligen passiven Elementen angeordnet. Anschließend wurden diese Gruppen ähnlich wie in der Schematik aus Abbildung 12 auf dem PCB platziert und Routing begonnen. Bei dem Routing Prozess wurde darauf geachtet großräumige Flächen an den Stellen mit hoher Energie zu verwenden. Zudem wurden an vielen Stellen Testpoints platziert, um später das Debuggen zu erleichtern. Zudem wurden an wichtigen Stellen Vias platziert, um Störkapazitäten zu reduzieren.

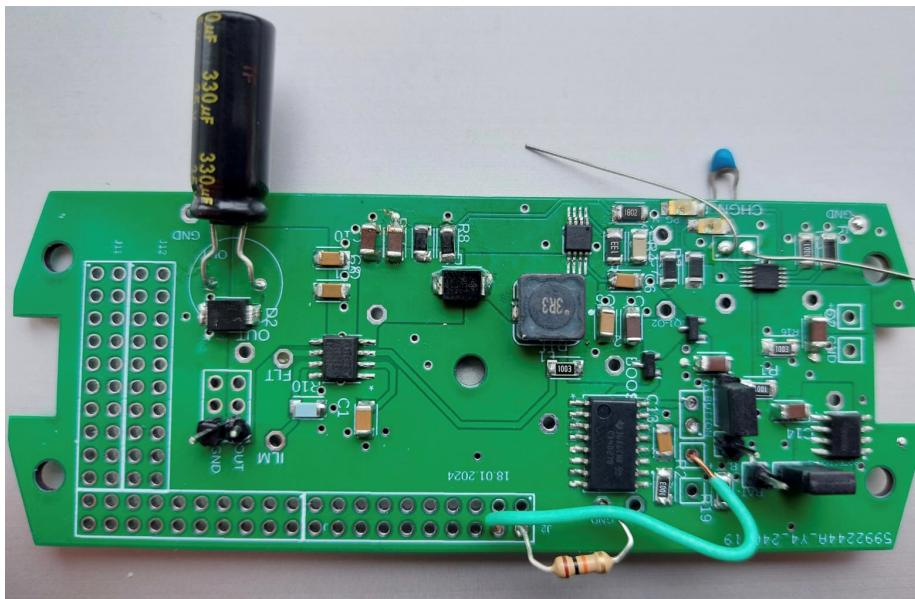


Abbildung 15: Energiemodul

4.5 Testen des Moduls

Das Testen der Platine für die Stromversorgung erfolgte in mehreren Phasen, da nicht alle Bauteile gleichzeitig zur Verfügung standen. Zu Beginn wurden die Funktionalität der Lade- und Boost-ICs getestet, da diese als erste zur Verfügung standen. Zum Testen wurden die entsprechenden ICs mit den zugehörigen passiven Bauteilen auf die Platinne gelötet. Hierbei lies sich feststellen, dass die Wärmebrücken in den Polygonen des PCB-Designs deaktiviert wurden. Dies geschah mit der Intention, dass die ICs besser verbunden werden und größere Ströme störungsfreier fließen könnten. Dies wirkte sich jedoch nachteilig auf den Lötprozess aus, da die Wärme aufgrund der großen thermischen Masse leicht abgeleitet wurde und das Zinn sich nicht so gut verteilen konnte

Zum Testen des Lade ICs wurde eine Spannung von 5 V aus einer externen Spannungsquelle angeschlossen und der Ladefortschritt mit einem Multimeter und einem Oszilloskop überwacht. Der Boost-IC konnte über einen Testpunkt gesteuert werden. Dazu wurden mit einer Spannungsquelle verschiedene Spannungen, die dem Ladezustand einer Batterie entsprechen, an den Eingang des ICs angeschlossen und der Ausgang beobachtet, welcher stets im Bereich von 4,8 V bis 5,1 V lag.

Nachdem die restlichen ICs, wie das JK-Flipflop und die Schutz-ICs, geliefert worden waren, konnte die Schaltung vollständig getestet werden. Im Schaltungsteil mit dem Flip-Flop, der für die Ein/Aus-Funktion wesentlich ist, wurde ein Fehler festgestellt. Beim Übertragen der Testschaltung aus Abbildung 13 in den Schaltplan wurde ein Pull-Down-Widerstand vergessen, was die Funktionalität zunächst einschränkte. Dank eines im Vorfeld erstellten Testplans konnte dieser Fehler leicht identifiziert werden. Auch die Behebung des Fehlers war einfach, da auf der Platine Leerstellen für Nachbesserungen vorgesehen waren.

Ein Lötfehler bei der eFuse, der einem Kurzschluss gleichkam, ermöglichte die direkte Überprüfung der Funktionalität. Sowohl die Last als auch die Batterie waren durch die Funktionen geschützt, und der Fehler konnte über den entsprechenden Pin erkannt werden. Auch das Beheben dieses Fehlers war dank etwas zusätzlichem Flussmittel unproblematisch.

Nach Abschließen der Tests lässt sich festhalten, dass eine Batterie mit der CCCV-Methode geladen werden kann. Während des Ladevorgangs wird eine Belastung des Systems durch einen PMOS erfolgreich verhindert. Wenn keine Ladespannung vorhanden anliegt, kann die Batterie nach Betätigung eines Schalters entladen werden, die Spannung wird auf 5 V erhöht, um die Satelliten zu betreiben. Diese sind während des Betriebes vor Kurzschlüssen oder anderen schädlichen Einwirkungen geschützt.

4.6 Reflektion der Arbeit

Die letzten Wochen und Monate waren äußerst lehrreich. Als das für die Stromversorgung zuständige Team haben wir es geschafft, alle Anforderungen an das System zu erfüllen:

- Die Stromversorgung für 45 bis 60 Minuten zu gewährleisten.
- Aufladen des Akkus über einen USB-C-Anschluss.
- Eine Versorgungsspannung von 5 Volt DC mit einem maximalen Strom von 1,5 Ampere bereitstellen.
- Ermöglichen des Ein- und Ausschalten am Gehäuse.

- Ermöglicht das Ausschalten des Systems per Software-Eingabe (kein "Deep Sleep").
- Schutz des Systems vor potenziellen Gefahren wie Kurzschlüssen, Überspannung, Tiefentladung usw.

Hierbei konnten wir unsere Fähigkeiten in Bezug auf Elektronikverständnis, Projektstrukturierung sowie der Ingenieursarbeit deutlich verbessern. Wir sind an der Herausforderungen des Arbeitsprozesses und des entwickeln eines Batterie Management Systems deutlich gewachsen.

Ein Aspekt, der im Vorfeld der Arbeit unterschätzt wurde, war das Layout der Leiterplatte. Die Schaltung ist relativ komplex und es war sehr zeitaufwändig das finale Design zu kreieren. Weiter war es herausfordernd den Überblick zu behalten und die entsprechenden Anforderungen/Empfehlungen aller Datenblätter zu berücksichtigen. Außerdem war es eine neue Erfahrung, die Schaltung gemäß den Layout-Empfehlungen beizubehalten und an die Gehäuseanforderungen anzupassen.

Eine Schwierigkeit, die sich im Nachhinein herausstellte, war, während des gesamten Arbeitsprozesses konzentriert zu bleiben. Es sind oft die kleinen Fehler, wie der vergessene Pull-Down-Widerstand, die sich in kurzen, unachtsamen Momenten einschleichen. Es ist jedoch erfreulich zu sehen, dass die erst kürzlich gelernte Praxis des Hinzufügens von Lochrastern auf leeren Stellen sehr nützlich sein kann, insbesondere beim Prototyping. Erwähnen sollte man auch die Tatsache, dass es keine Wärmebrücken gibt. Diese haben den Lötprozess zwar etwas erschwert, scheinen aber der Funktion der Schaltung zugute zu kommen, da trotz Strömen von bis zu 1,5 Ampere keine nennenswerte Wärmeentwicklung zu beobachten ist.

Wir können mit Stolz auf das Erreichte zurückblicken. Es ist sehr erfreulich, dass es uns gelungen ist, alle Herausforderungen projektübergreifend und in kleineren Gruppen zu meistern sodass am Ende alles so gut zusammengekommen ist.

5 LED Beleuchtung

5.1 Eingliederung in das Projekt

Die Integration der LED-Beleuchtung in das Projekt bietet nicht nur eine ästhetische Komponente, sondern erweitert auch die Funktionalität des Projektes. Die Wahl der RGB-LEDs WS2812B ermöglicht die flexible Ansteuerung jeder einzelnen LED, was eine vielfältige Gestaltung der Beleuchtung und die Umsetzung verschiedener Spielmodi ermöglicht. Darüber hinaus dienen die LEDs zur Anzeige von Statusinformationen wie dem Verbindungsstatus und dem Batteriezustand, wodurch eine benutzerfreundliche Interaktion gewährleistet wird.

Die LED-Beleuchtung wird durch eine speziell entworfene Platine realisiert, die die LEDs in einer kreisförmigen Anordnung aufnimmt. Dies ermöglicht eine gleichmäßige und ansprechende Ausleuchtung nach oben. Zusätzlich wird ein LED-Streifen verwendet, der eine 360°-Beleuchtung bietet und am Rand des Gehäuses angebracht ist. Dieser Ansatz sorgt für eine immersive visuelle Erfahrung während des Spielbetriebs.

Um den Status des Systems klar zu kommunizieren, wurde eine separate Platine für eine Status-LED entwickelt. Diese Platine wird auf der Seite des Gehäuses positioniert und zeigt durch verschiedene Farben und Blinkmuster den aktuellen Zustand des LEDmints-Systems an.

Die Implementierung der LED-Beleuchtung trägt nicht nur zur ästhetischen Aufwertung bei, sondern ermöglicht auch die Schaffung eines einzigartigen Spielerlebnisses durch die Integration von Lichteffekten in die Spielsituationen.

5.2 Funktionsweise

Die Platine für die LED-Beleuchtung wurde so gestaltet, dass sie neben den RGB-LEDs WS2812B auch alle notwendigen Komponenten enthält, um eine nahtlose Integration in das Gesamtsystem zu gewährleisten. Die Anordnung der LEDs in einem Ring ermöglicht die Erzeugung kreativer Lichteffekte, die das Spielerlebnis intensivieren.

Die WS2812B RGB-LEDs haben vier Pins: Versorgungsspannung, Masse, Eingangssignal und Ausgangssignal. Die Verbindung zwischen den LEDs erfolgt durch Verknüpfung des Ausgangssignal-Pins einer LED mit dem Eingangssignal-Pin der nächsten LED, und so weiter. Koppelkondensatoren von $0.1\mu\text{F}$ zwischen Versorgungsspannung und Masse jeder LED dienen dazu, einen stabilen Betrieb sicherzustellen.

Die Status-LED auf der Seite des Gehäuses kommuniziert über verschiedene Farben und Blinkmuster wichtige Informationen, wie den Verbindungsstatus mit anderen Satelliten oder den Batteriezustand. Durch klare und intuitive visuelle Rückmeldung trägt sie zur Benutzerfreundlichkeit des Systems bei.

Die Programmierung des Mikrocontrollers (ESP32) ermöglicht die Steuerung der LED-Beleuchtung und die Anpassung der Lichteffekte entsprechend den Anforderungen des Spiels. Mithilfe der Neopixel-Bibliothek von Adafruit Industries können individuelle LEDs im Ring präzise gesteuert werden. Zum Beispiel kann die Farbe einer bestimmten LED durch die Angabe von RGB-Werten festgelegt werden.

Die Flexibilität der Programmierung erlaubt es, verschiedene Lichteffekte zu erzeugen, von schlichten Farbwechseln bis hin zu komplexen Animationen. Durch die geschickte Kombination von Elektronik und visuellen Effekten entsteht ein innovatives und unterhaltsames LEDminton-System. Während des Spielbetriebs kann die Beleuchtung in Echtzeit gesteuert werden, um das Spielerlebnis zu personalisieren und auf Ereignisse im Spiel zu reagieren. Dies trägt nicht nur zur ästhetischen Aufwertung bei, sondern ermöglicht auch die Schaffung eines einzigartigen Spielerlebnisses durch die Integration von Lichteffekten in die Spielsituationen.

5.3 Schaltplan

Aufgrund des begrenzten Platzes im Gehäuse wurde entschieden, 16 LEDs auf die Hauptplatine unseres Moduls zu setzen. Diese Hauptplatine ist mit dem Mikrocontroller verbunden, um das Eingangssignal, die Spannungsversorgung und die Masse zu erhalten, und leitet das Signal an die Status-LED und den LED-Streifen weiter.

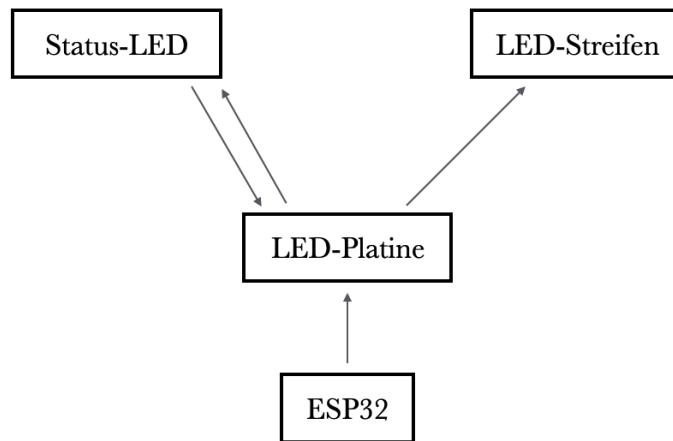
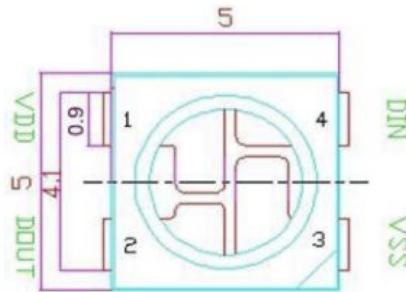


Abbildung 16: Schematischer Aufbau des LED-Moduls

Der Schaltplan der LED-Platine wurde in Fusion 360 entworfen. Die verwendeten LEDs WS2812B haben vier Pins (siehe Abbildung 17): Versorgungsspannung, Masse, Eingangssignal und Ausgangssignal. Um das Signal von einer WS2812B-LED zur nächsten zu übertragen, wird der Ausgangssignal-Pin der ersten LED mit dem Eingangssignal-Pin der nächsten LED verbunden, und so weiter. Koppelkondensatoren von $0.1\mu\text{F}$ zwischen der Versorgungsspannung und der Masse aller LEDs dienen dazu, einen stabilen Betrieb sicherzustellen.

Pin-Konfiguration



Pin-Funktion

Nr.	Symbol	Funktionsbeschreibung
1	VDD	LED Stromversorgung
2	DOUT	Ausgang Steuerungsdatensignal
3	VSS	Masse
4	DIN	Eingang Steuerungsdatensignal

Abbildung 17: WS2812B Pin-Konfiguration (aus Zedfy, o. D.)

Die LED-Platine hat drei Steckverbinder (siehe Abbildung 18), zwei 1x3 (JP2 und JP3) und einen 1x4 (JP1). JP2 ist mit der Mikrocontroller-Platine verbunden und nimmt das Eingangssignal mit Pin 1, die Versorgungsspannung mit Pin 2 und die Masse mit Pin 3 auf. JP1 ist mit der Status-LED verbunden. Das Ausgangssignal dieser einzelnen LED wird an Pin 4 von JP1 und das Eingangssignal wird an Pin 1 von JP1 angeschlossen. Pin 4 von JP1 ist mit Pin 1 von JP3 kurzgeschlossen und ist mit dem Eingang des LED-Streifens zu verbinden. Bei den drei Steckern ist Pin 2 mit der Versorgungsspannung und Pin 3 mit der Masse zu verbinden.

Um die LEDs vor Unregelmäßigkeiten in der Versorgungsspannung zu schützen und um sicherzustellen, dass der Strom nicht zu hoch wird, wurde schließlich ein Filter mit einem größeren Kondensator von $680\mu\text{F}$ und einem kleinen Widerstand von 6.8Ω auf der Platine angebracht.

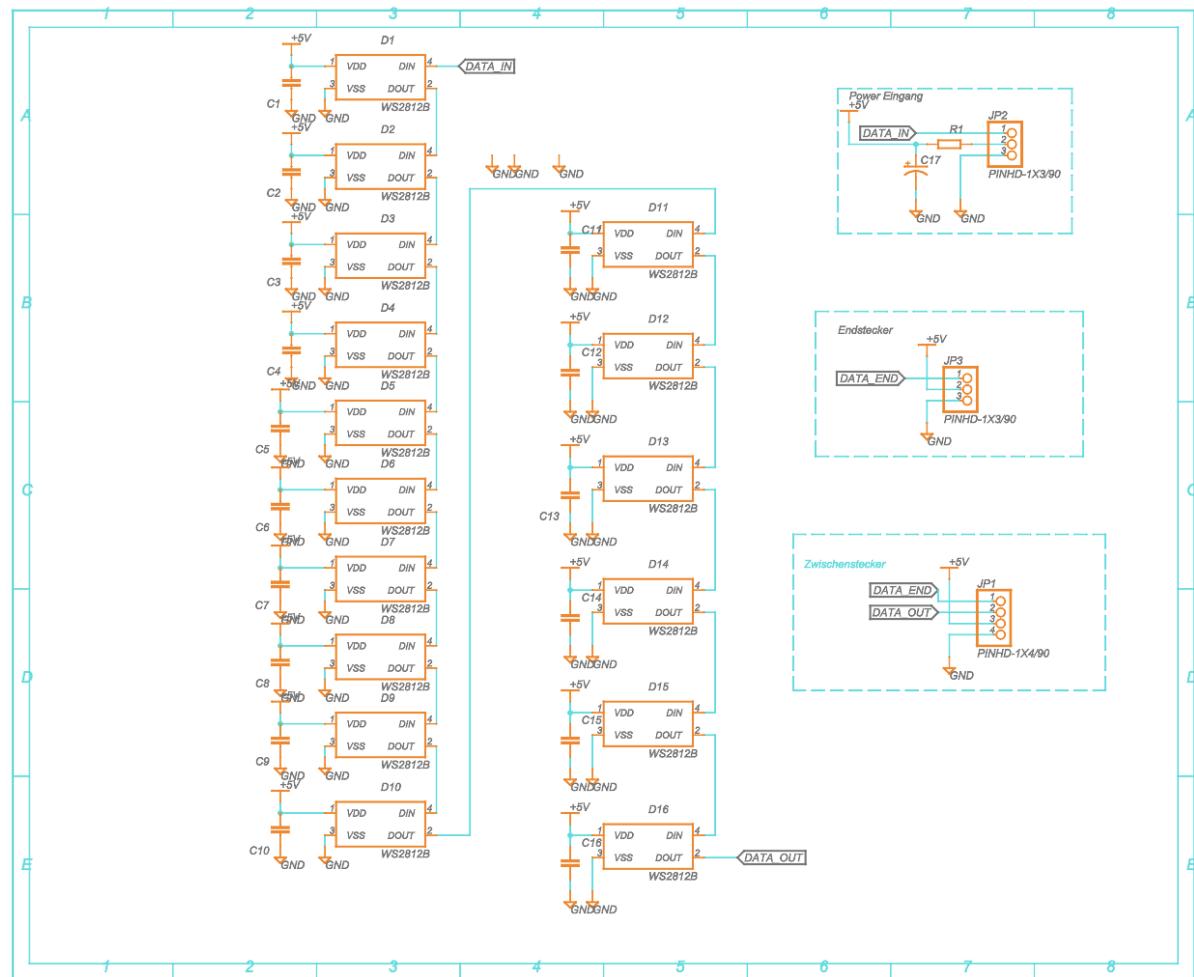


Abbildung 18: Schaltplan der LED-Gruppe

5.4 PCB

Ausgehend vom Schaltplan wurde die PCB mit Hilfe der Software Fusion 360 erstellt. Zunächst wurden die Abmessungen der Platine nach Absprache mit dem Gehäuse Verantwortlichen festgelegt. Die Größe der Platine beträgt 78x78 mm, sie hat in jeder Ecke und in der Mitte Bohrungen mit einem Durchmesser von 3.3 mm, die zur Positionierung der Platine im Gehäuse dienen.

Die LEDs wurden zusammen mit ihren jeweiligen Koppelkondensatoren in einem Kreis angeordnet, und auf der anderen Seite der Leiterplatte wurden der große Kondensator und der Widerstand platziert.

5.5 Programmierung

Für die Steuerung der LEDs haben wir uns für die Neopixel-Bibliothek von Adafruit Industries entschieden. Um die Bibliothek zu verwenden, muss sie zuerst eingebunden werden:

```
#include <Adafruit_NeoPixel.h>
```

Nach der Einbindung wird ein Objekt der Neopixel-Klasse erstellt:

```
Adafruit_NeoPixel ws2812b(NUM_PIX, PIN, NEO_GRB + NEO_KHZ800);
```

Hierbei steht "NUM_PIX" für die Anzahl der Neopixel im LED-Ring, und "PIN" repräsentiert den Pin, an dem der LED-Ring angeschlossen ist. Diese Initialisierung ermöglicht es, die Neopixel-LEDs im Projekt anzusteuern.

Die Neopixel-Bibliothek bietet eine Vielzahl von Funktionen für die individuelle Kontrolle jeder LED im Ring. Zum Beispiel kann die Farbe einer bestimmten LED durch Angabe von RGB-Werten festgelegt werden:

```
ws2812b.setPixelColor(LED_INDEX, RED, GREEN, BLUE);
```

Hierbei steht "LED_INDEX" für die Position der LED im Ring, und "RED", "GREEN" und "BLUE" repräsentieren die Intensität der jeweiligen Farbkomponente.

Um die gewünschten Lichteffekte zu erzeugen, können verschiedene Funktionen wie das Ändern der Helligkeit, das Durchlaufen von Farbverläufen und das Blinken einzelner LEDs verwendet werden. Die Programmierung der Neopixel-LEDs erlaubt somit eine flexible und anpassbare visuelle Gestaltung, um das LEDmintons-Erlebnis zu verbessern. Die Integration dieser Programmierfunktionalitäten ermöglicht es, die Beleuchtungseffekte in Echtzeit während des Spielbetriebs zu steuern und auf bestimmte Ereignisse oder Benutzeraktionen zu reagieren. Dadurch wird nicht nur die visuelle Attraktivität des Systems verbessert, sondern auch eine interaktivere und unterhaltsamere Spielerfahrung geschaffen.

5.6 Reflexion der Arbeit

Die Entwicklung und Integration der LED-Beleuchtung in das LEDmintons-Projekt war äußerst erfolgreich. Unser Team hat die folgenden Ziele erreicht:

- Erfolgreiche Umsetzung visuell ansprechender Beleuchtungseffekte für eine verbesserte Spielerfahrung.
- Integration von RGB-LEDs (WS2812B), um unterschiedliche Farben und individuelle Adressierung jeder LED zu ermöglichen.
- Erstellung einer eigenen Platine für die LED-Anordnung, um eine 360°-Beleuchtung zu realisieren.
- Implementierung von Status-LEDs für eine klare Anzeige von Verbindungsstatus und Batteriezustand.

Die Programmierung der LEDs erfolgte reibungslos unter Verwendung der Neopixel-Bibliothek. Durch die individuelle Ansteuerung jeder LED konnten wir vielfältige Lichteffekte realisieren, die nicht nur die ästhetische Qualität des Projekts verbessern, sondern auch zur Spielerinteraktion beitragen.

Die Herausforderungen lagen vor allem in der physischen Integration der LED-Platine in das Gehäuse und der Ausrichtung, um die gewünschten Beleuchtungseffekte zu erzielen. Auch die Verkabelung und Koordination zwischen den verschiedenen Teams waren entscheidende Aspekte für den Erfolg des LED-Bereichs.

Die Erfahrung hat gezeigt, dass die sorgfältige Abstimmung zwischen Elektronik, Design und Programmierung für ein optimales Ergebnis von entscheidender Bedeutung ist. Der Prozess hat nicht nur unsere technischen Fähigkeiten gestärkt, sondern auch die Bedeutung einer interdisziplinären Zusammenarbeit verdeutlicht.

6 Sensorik

Im folgendem Kapitel wird auf die im Projekt verwendete Sensorik genauer eingegangen. Dazu wird zunächst die Eingliederung ins Projekt beschrieben, danach folgen die Anforderungen, die die Sensoren erfüllen mussten, deren Funktionsweise, die Programmierung und der Schaltplan.

6.1 Eingliederung in das Projekt

Sensoren sind dazu da Informationen aus der Umgebung aufzunehmen und diese dann weiterzuverarbeiten. Sensoren, oder Messaufnehmer, sind also in der Lage nichtelektrische Messgröße in elektrische Primärsignale umzuwandeln, die dann mit weiteren Elektroniken weiterverarbeitet werden kann. Die erste Herausforderung für das Projekt war es einen geeigneten Sensor zu finden, der schnelle und genauere Ergebnisse liefert. Der Sensor sollte in der Lage sein, den Badminton-Schläger auf einem kurzen oder langen Abstand detektieren zu können, schnellen und langsame Schläge aufnehmen zu können und programmierbar zu sein.

6.2 Anforderungen

Da BETH vor allem zum Badmintontraining angedacht ist und die individuellen Reaktionszeiten ausgegeben werden sollen, wurde bei der Auswahl der geeigneten Sensoren insbesondere auf die Reaktionsgeschwindigkeit geachtet. Dazu würden die Satelliten auf dem Badmintonfeld verteilt aufgestellt werden und sobald ein Satellit aufleuchten, soll mit dem Badmintonschläger über den Satelliten in der Luft geschlagen werden. Diese Bewegung über dem Satelliten soll schnellstmöglich registriert und weiterverarbeitet werden. Zu diesem Zweck wurde ein Sensor benötigt, der Entfernungen zu einem Objekt zuverlässig messen kann. In der engeren Auswahl standen ein Ultraschallsensor und ein Time of Flight (kurz TOF) Sensor wobei die Entscheidung nach dem Testen beider Sensoren auf den TOF fiel, welcher im Folgendem genauer beschrieben wird.

Die zweite Anwendung sieht vor, dass die Satelliten nah aneinander, zum Beispiel auf einer Tischplatte, plaziert werden. Sobald ein Sensor aufleuchtet, soll durch die Detektion einer Berührung das Signal weitergegeben werden, dass der nächste Satellit aufleuchten kann. Da der in diesem Projekt verwendete ESP32 einen eingebauten Touchsensor besitzt, wurde an dieser Stelle auf einen externen Sensor verzichtet.

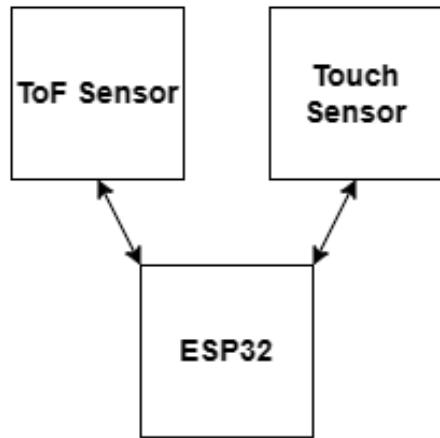


Abbildung 19: Schematischer Aufbau der Verschaltung

6.3 Funktionsweise

In diesem Abschnitt soll die Funktionsweise der im Projekt verwendeten Sensoren genauer erläutert werden.

6.3.1 TOF Sensoren

Die Entfernungsmessung wird vom Time of Flight Sensor (TOF) übernommen. Der Sensor wurde wegen seiner Zuverlässigkeit in der Detektion, schnellerer Reaktion und seiner kleineren Größe ausgewählt. Der TOF ist in der Lage, einen Abstand von 40mm bis 4m mit einer Messgenauigkeit von $\pm 5\%$ zuverlässig zu messen. Die Messung basiert darauf, dass eine Quelle Licht in Richtung des Zielobjektes sendet, dieses dann an einem Objekt reflektiert wird und zum Sensor zurückkehrt. Die dafür benötigte Zeit wird gemessen und daraus eine Distanz berechnet. Dies bedeutet, dass der Laser blitzschnelle Ergebnisse ermöglicht. Der Sensor hat 6 Pins: einen Versorgungspin, einen GND-Pin, SCL und SDA Pins für das I2C-Kommunikationsprotokoll, und einen Xshut Pin, welcher dafür sorgt, dass der Sensor nicht den ganzen Zeit angeschaltet ist, und einen GPIO Pin (Global Purpose Input Output), der dem User die Entscheidung überlässt, wozu er verwendet wird. In unserem Fall ermöglicht dieser Pin die Verwendung des INTERRUPTS. Die Versorgungsspannung beträgt 5V mit einer maximalen Stromversorgung 20mA.

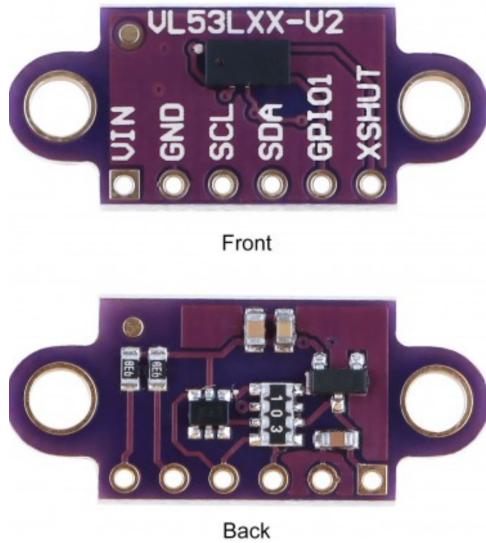


Abbildung 20: TOF Sensor (aus AZ-Delivery, o. D.)

Der Sensor wird von der Basis angeschaltet, die einen Befehl via Wifi an den Esp32s schickt. Soweit sich der Badmintonschläger oberhalb des Laser befindet, misst der Sensor die entsprechende Entfernung und schaltet sich aus, bis der nächste Befehl geschickt wird. Um die Sensorik zu verbessern werden in diesem Projekt drei TOF Sensor genutzt, die softwaremäßig verODERt werden.

6.3.2 Touch Sensor

Der Touchsensor wird mithilfe des Touchpins des ESP32 aufgebaut. Der ESP32s besitzt mehr als 9 Touchpins. Diese Touchpins können Veränderungen in allem, was eine elektrische Ladung besitzt, wie zB. die menschliche Haut, erkennen. So können Veränderungen der elektrischen Ladung, die durch eine Berührung des Touchpins ausgelöst wurden, detektiert werden. Der Pin wird durch ein Kabel, welches auf den Deckel des Satelliten zu einer Metallplatte führt, erweitert. Sobald jemand die Metallplatte berührt, wird ein elektrische Signal zum ESP32 geschickt. Wichtig dabei ist nicht zu vergessen, dass, je häufiger die Platte berührt wird, sich ebenfalls die Gefahr einer "Überladung" erhöht. Die Schutzvorrichtung dagegen, bestehend aus einer Supressordiode, wird detaillierter im Schaltplan dargestellt.

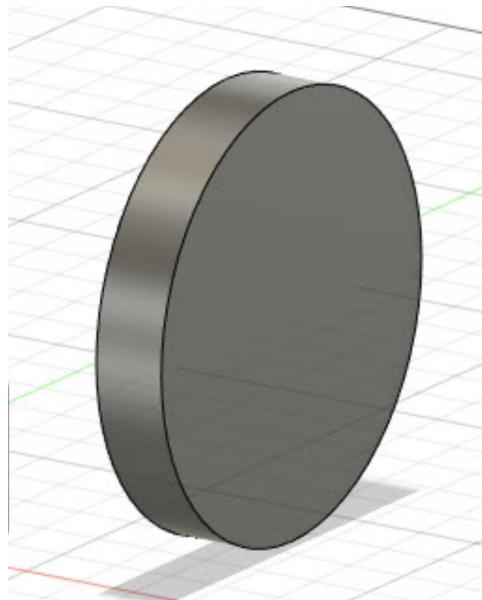


Abbildung 21: Touch Sensor

6.4 Programmierung Interrupt

Um einen reibungslosen Programmablauf und eine möglichst zuverlässige Detektion von Bewegungen durch die TOF Sensoren zu ermöglichen, wurden bei der Programmierung sogenannte Interrupts verwendet. Diese sind notwendig, damit die Leistung der TOF Sensoren verbessert wird und zu jedem Zeitpunkt eine Veränderung der Umwelt wahrgenommen werden kann, welche dann unverzüglich an den ESP32 weitergeleitet wird. Normalerweise wird eine ständige Überprüfung des Status im "loop" durchgeführt, damit der nächsten Befehl ausgeführt wird. Dies verlangsamt jedoch den Prozess. Dies lässt sich in einem komplexen Programm nicht leicht integrieren, besonders wenn diese Überprüfung ein delay verlangt was den ganzen Mikroprozessor anhält.

Um solche Problemen zu verhindern werden Interrupts verwendet, die stattdessen auf eine Unterbrechung warten während dazwischen die restlichen Befehle ausgeführt werden. Somit reagiert der Sensor nicht nur schneller, sondern lässt sich leicht mit dem Rest integrieren. Wie vorher erwähnt, wird für diese Funktion der GPIO Pin vom TOF genutzt. Jeder TOF Sensor hat seine eigene Interruptleitung, um die Verwendung einer externen Logikbauteil bzw. ODER zur verwenden. Alle drei TOFs werden über die Software verordert.

6.5 Schaltplan

Bei der Erstellung der Schaltpläne und Platinen wurde berücksichtigt, dass der Platz innerhalb der Satelliten begrenzt ist und, dass jede Teilgruppe unterschiedlich viele Bauteile zu platzieren hat. Da die Sensoren möglichst nah am Deckel angebracht werden sollten (Abb. 22) wurde nur wenig Platz auf der Platine benötigt. Aus diesem Grund wurde entschieden, dass eine eigene Platine für die Sensorgruppe nicht notwendig sei und die benötigten Bauteile auf die Platine der Audiogruppe hinzugefügt werden können.

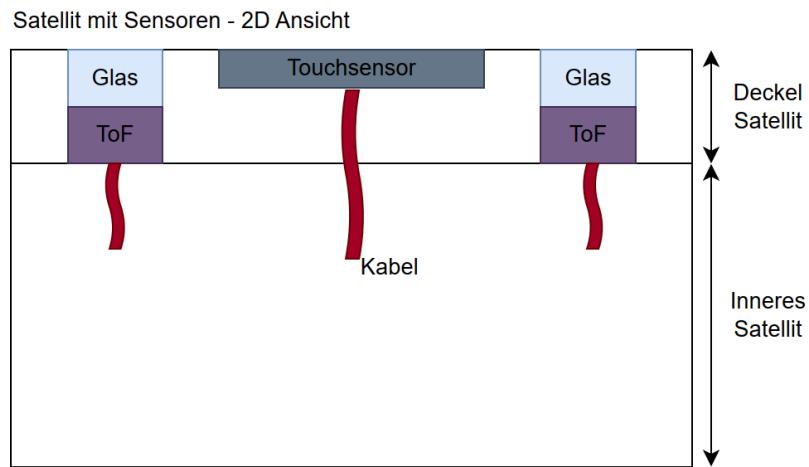
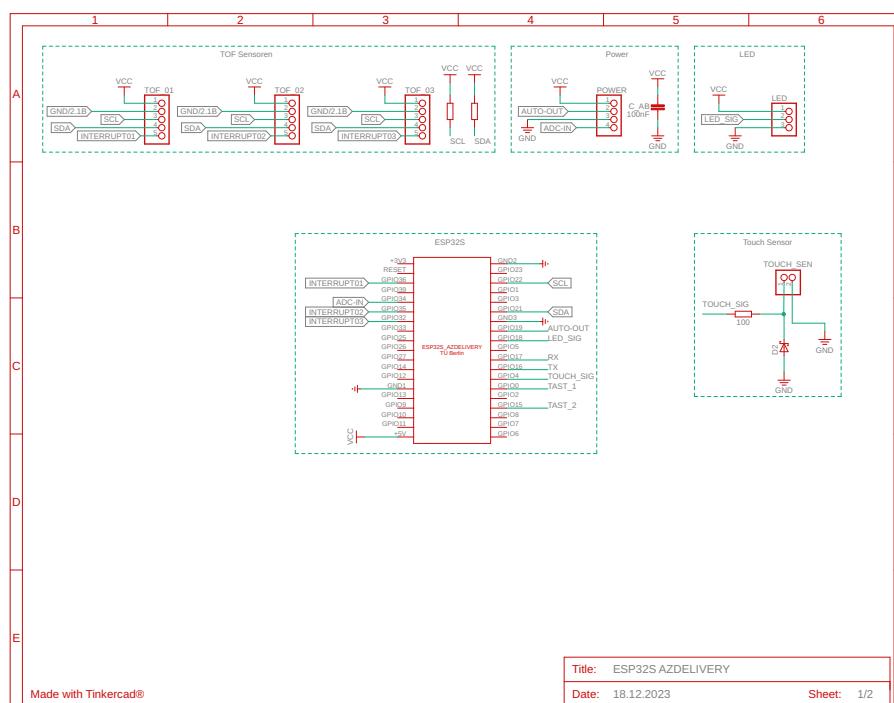


Abbildung 22: Plazierung der Sensoren im Satelliten

Im Folgenden kann man den Schaltplan aus Fusion300, in dem die Platinen designt wurden, erkennen.



25.02.24 17:53 /Users/brendon/Downloads/lalala.pdf (Sheet: 1/2)

Abbildung 23: Vollständiger Schaltplan der Sensoren

Zunächst wird die Verschaltung der TOF Sensoren in Abb. 24 vergrößert dargestellt. Da in diesem Projekt nur fünf der sechs Pins des TOF verwendet wurden, wird ebenfalls

nur ein Stecker mit 5 Steckplätzen auf der Platine verbaut. Zwischen den SDA und SCL Pins und der Versorgungsspannung wurden Widerstände eingebaut.

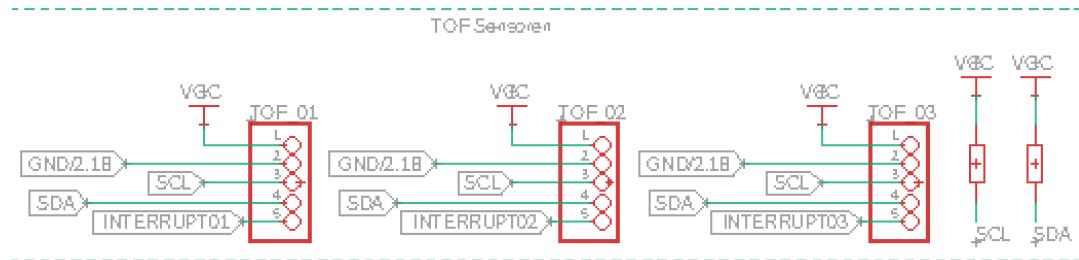


Abbildung 24: TOF Sensor Schematic

In der folgenden Grafik wird die Verschaltung des Touchsensors dargestellt. Zum Schutz vor Überladungen und den damit verbundenen, negativen Auswirkungen auf den ESP32 wird eine Supressordiode und ein Widerstand verwendet.

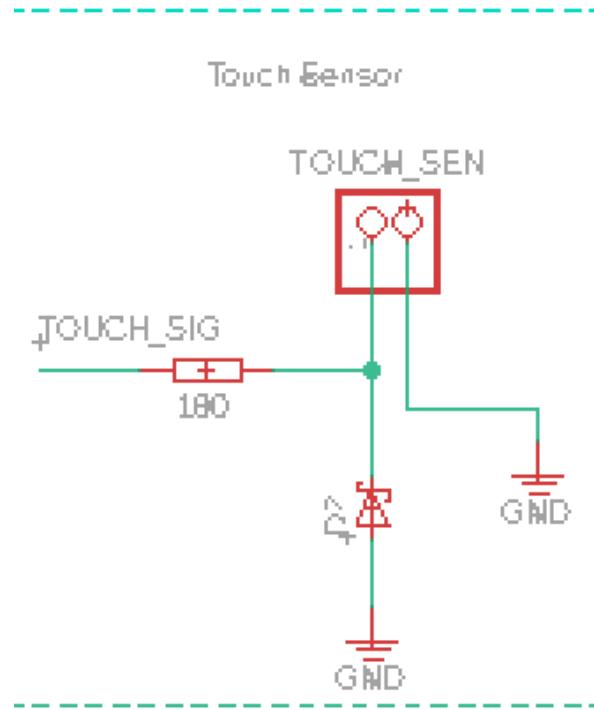


Abbildung 25: Touch Sensor Schematic

6.6 Reflexion der Arbeit

Bei der Implementierung der Sensorik gab es einige Hürden, die jedoch gemeistert werden konnten. Es wurde folgendes erreicht:

- Die Time of Flight Sensoren reagieren zuverlässig
- Der Touchsensor ist funktionstüchtig
- Die Sensoren sind im Spielgeschehen integriert

Als Ausblick für weitere Verbesserungen der Sensorik könnte versucht werden alle drei Time of Flight Sensoren gleichzeitig verwenden zu können.

7 Kommunikation

In diesem Abschnitt, der sich mit dem Thema "Kommunikation" befasst, werden die technische Implementierung und Realisierung der Kommunikation zwischen den Satelliten (ESP32) und der Basisstation (Raspberry Pi) erläutert.

7.1 Eingliederung in das Projekt

Sämtliche Benutzerinteraktionen, wie beispielsweise die Auswahl eines Spiels, die Anzeige von Reaktionszeiten und der Status der Satelliten, erfolgen über die Basisstation. Daher spielt die Kommunikation zwischen der Basisstation und den Satelliten eine entscheidende Rolle. Alle Anforderungen im Bereich der Kommunikation werden im nachfolgenden Abschnitt spezifiziert.

Bei der Gestaltung des Projekts wurden zwei Kommunikationsmethoden in Betracht gezogen, nämlich Bluetooth und drahtlose Verbindung (WLAN). Nach sorgfältiger Überlegung wurde entschieden, dass WLAN die bessere Option ist und alle Anforderungen erfüllt. Auf diese Aspekte wird im nächsten Abschnitt eingegangen.

Für das Kommunikationsprotokoll standen ESP-NOW und das Transmission Control Protocol (TCP) zur Auswahl. ESP-NOW ist ein von Espressif definiertes drahtloses Kommunikationsprotokoll, das die direkte, schnelle und energiesparende Steuerung intelligenter Geräte ermöglicht, ohne dass ein Router erforderlich ist. Die Entscheidung für die TCP-Kommunikation wurde getroffen, da eine direkte Kommunikation zwischen den Satelliten nicht erforderlich war und die Implementierung von TCP einen geringeren Anwendungsaufwand erforderte.

7.2 Anforderungen

Die formulierten Anforderungen sind wie folgt:

- Die Basisstation soll in der Lage sein, gleichzeitig mit allen Satelliten verbunden zu werden.
- Die Verbindung zwischen den Satelliten und der Basisstation soll stabil sein.
- Beide Seiten sollen in der Lage sein, Unterbrechungen der Verbindung schnellstmöglich zu erkennen.
- Es soll kein oder minimaler Paketverlust in der Verbindung auftreten.

- Die Satelliten müssen in der Lage sein, sich unmittelbar nach dem Einschalten oder bei einer unterbrochenen Verbindung sofort wieder zu verbinden.
- Die Verbindungen zwischen der Basisstation und den Satelliten sollen unabhängig voneinander sein.
- Verbindungsverzögerungen sind zu minimieren.
- Die Kommunikation soll innerhalb eines maximalen Abstands von 10 Metern funktionieren.

Die Entscheidung für WLAN anstelle von Bluetooth in diesem Projekt basiert auf mehreren Überlegungen, darunter die Unterstützung einer größeren Anzahl gleichzeitiger Verbindungen, die erhöhte Stabilität über längere Entfernung sowie die höhere Bandbreite für eine schnellere und qualitativ hochwertigere Datenübertragung. Abhängig von den spezifischen Anforderungen des Projekts und den verfügbaren Ressourcen könnte die Implementierung von WLAN möglicherweise weniger Entwicklungsaufwand erfordern, insbesondere wenn bereits Erfahrung mit dieser Technologie vorhanden ist. Insgesamt bietet WLAN somit eine robustere Grundlage für die Kommunikationsanforderungen dieses Projekts im Vergleich zu Bluetooth.

7.3 Implementierung

Im Rahmen der Implementierung wird das Raspberry Pi OS mit Desktop-Umgebung auf die SD-Karte überschrieben. Networkmanager, ein Tool für die Netzwerkkonfiguration, wird in Raspberry Pi genutzt, um WLAN zu konfigurieren. Der ESP-32 wird in der Arduino-Umgebung programmiert. Sämtliche Nachrichten werden im JSON-Format formatiert und übertragen.

7.3.1 Raspberry Pi Programmierung

Ein kabelloser Hotspot wird über die Networkmanager GUI erstellt, wobei WPA2-PSK als Verschlüsselungsmethode ausgewählt wird. Die Kommunikation und die Benutzeroberfläche (GUI) für Interaktionen werden mithilfe von Python (Version 3.12) implementiert. Eine Trennung zwischen Kommunikation und GUI erfolgt durch den Multi-Prozess-Ansatz, wodurch beide Teile parallel laufen können, ohne sich zu blockieren. Es werden gemeinsame Variablen für jeden Satelliten erstellt, auf die beide Prozesse zugreifen können. Der TCP-Server mit einem Port 8888 wird mithilfe der eingebauten Bibliothek `socket` mit der `TCP_KEEPALIVE`-Konfiguration realisiert. Vor der Inbetriebnahme des TCP-Servers wird der Port 8888 in der eingebauten Firewall von Raspberry

Pi mittels `iptables`-Programm für den Zugriff freigegeben. Außerdem werden die Kommunikationsaufgaben auf mehrere Threads verteilt.

Diese Aufgaben sind wie folgt:

- Ankommende Verbindungen akzeptieren
- Bestehende Verbindungen überwachen
- Ankommende Nachrichten einlesen
- Ausgehende Nachrichten von GUI abholen und senden

Der Überwachungsthread läuft immer und aktualisiert die gemeinsamen Variablen, so dass die Änderungen der Verbindung sofort in der GUI angezeigt werden können. Die Threads für die letzten beiden Aufgaben werden automatisch beendet, wenn die Verbindung unterbrochen wird.

7.3.2 Graphical User Interface

Für die Erstellung der GUI wurde beschlossen, das Framework Qt5 mit der Python-Bibliothek PyQt5 zu verwenden.

Das Design der Benutzeroberfläche ist in zwei Teile gegliedert: eine obere Bar, in der der Status der einzelnen Satelliten in Echtzeit angezeigt wird, sowie eine Taste zum Beenden des Programms. Unterhalb der oberen Bar befindet sich das Hauptfenster, das je nach den Eingaben unterschiedliche Komponenten anzeigt.

Die Hauptseiten des Hauptfensters werden im Folgenden beschrieben:

Main Menu: Der Startpunkt des Programms. Hier kann man den Titel lesen und auswählen, was der nächste Schritt ist. Drei Tasten bieten Optionen, um zu verschiedenen Seiten zu navigieren: Start und Protokoll.

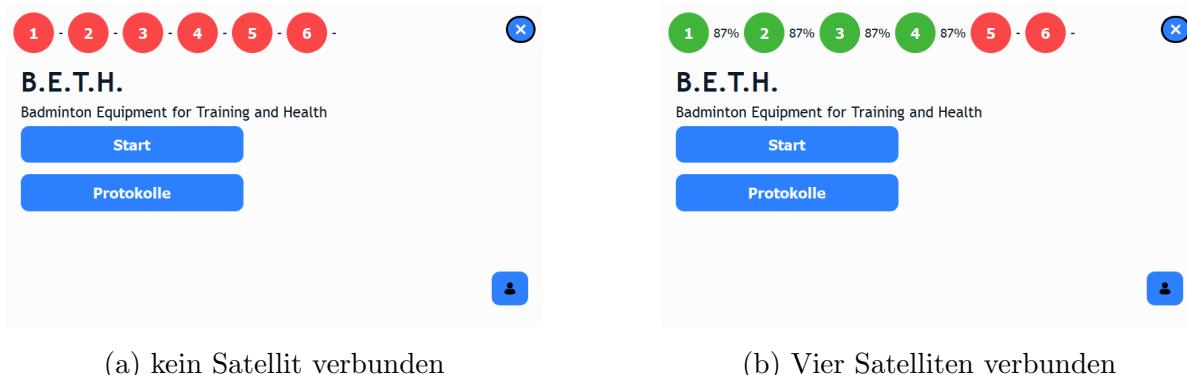


Abbildung 26: Main Menu des GUIs

Start: Hier werden verschiedene Trainingsroutinen zusammen mit einer Erklärung angezeigt, so dass sie ausgewählt und gestartet werden können.

Zum aktuellen Zeitpunkt ist ein Trainingsmodus verfügbar, das Reaktionstraining. Hier muss man so schnell wie möglich auf die sich zufällig aktivierenden Satelliten reagieren. Am Ende erhält der/die Benutzer/in ein Ergebnis, das Informationen über seine Leistung anzeigt.

Protokoll: Die Informationen, die während vergangener Trainings gesammelt wurden, können hier visualisiert und analysiert werden. Es ist geplant, mehr aufschlussreiche Daten über Diagramme und einfache Analysen bereitzustellen.



(a) Start Menu

(b) Anzahl von Runden

Abbildung 27: Trainings Menu



Abbildung 28: Protokoll

Eine kurze Reflexion: Gleichzeitigkeit und Parallelität spielen bei der Entwicklung von Benutzeroberflächen eine entscheidende Rolle, insbesondere wenn sie von externen Ereignissen abhängen, die sich ihrer direkten Kontrolle entziehen. Durch die Integration von Parallelität können UIs effektiv mehrere Aufgaben gleichzeitig bewältigen und so Reaktionsfähigkeit und reibungslose Benutzerinteraktionen gewährleisten. In unserem speziellen Fall, in dem unsere Benutzeroberfläche auf die Kommunikation mit den Satelliten angewiesen war, wurde die Verwendung von Multithreading mit thread-sicheren Mechanismen aufgrund des unvorhersehbaren Ereignis-Timings unabdingbar, um Robustheit zu gewährleisten. Mit Hilfe von Multithreading konnte die GUI die Kommunikation mit den Satelliten gleichzeitig abwickeln und dabei das unterschiedliche Tempo der ein- und ausgehenden Datenströme berücksichtigen. Thread-sichere Mechanismen gewährleisteten die Datenintegrität und verhinderten Konflikte, wodurch die Zuverlässigkeit und Belastbarkeit der grafischen Benutzeroberfläche bei der Verarbeitung von Echtzeit-Satellitenereignissen verbessert wurde.

7.3.3 ESP-32 Programmierung

Der Code der Satelliten wird in C++ mit Hilfe von Espressifs `Wifi`-Bibliothek und `ArduinoJson` implementiert. Für die Programmierung der ESP32s wird PlatformIO eingesetzt. PlatformIO bietet nützliche Funktionen an, wie die verbesserte Organisation von Header- und Quelldateien sowie die automatische Herunterladung aller Bibliotheken und erweiterte Debugging-Tools. Die Zugangsdaten für das WLAN sind im ESP-32-Code hartkodiert. Die IP-Adresse der Basisstation wird durch die Funktion `Wi-Fi.gatewayIP()` ermittelt, und daraufhin wird eine TCP-Verbindung mittels des `WiFiClient`-Objekts hergestellt. Die Nachrichten werden mithilfe von Methoden in `ArduinoJson` (`serializeJson`, `deserializeJson`) serialisiert bzw. deserialisiert. Nach erfolgreicher Herstellung der TCP-Verbindung sendet der ESP-32 eine Nachricht, welche die Satellite-ID und den Batteriestand enthält. Diese Informationen werden anschließend in der grafischen Benutzeroberfläche im Raspberry Pi dargestellt. Der Programmablauf für die Kommunikation wird in folgendem Ablaufdiagramm 29 veranschaulicht.

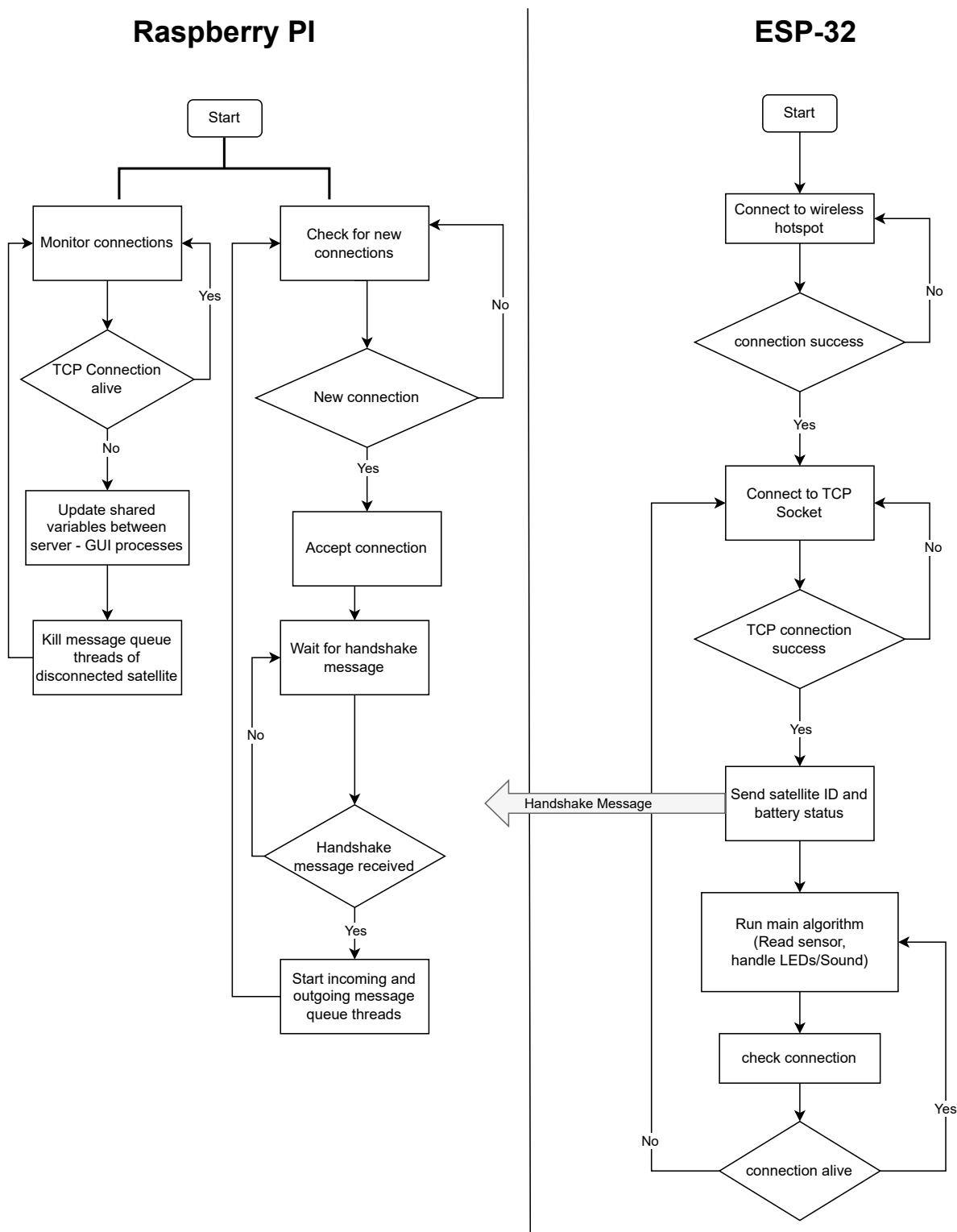


Abbildung 29: Ablaufdiagramm der Kommunikation

Die Codes aller Hardwarekomponenten werden in objektorientierten Klassen organisiert, um eine bessere Strukturierung des Quellcodes zu ermöglichen. Insgesamt werden sechs Klassen erstellt: Audio, Power, Sensor (für den ToF-Sensor), Touchsensor, LEDs und Communication. In der `setup()`-Funktion des ESP32 werden alle diese Teile als globale Variablen initialisiert, um sie in der Spiellogik zu verwenden. Zusätzlich wird ein Debug-Modus entwickelt, um alle diese Komponenten gleichzeitig zu debuggen.

Zuallererst wird die LEDs-Klasse initialisiert, um bestimmte Ereignisse beim Start anzuzeigen, wie zum Beispiel das Aufleuchten der LEDs in Violett beim Starten des Sensors und die farbcodierte Anzeige des Batteriestands (Grün für einen Batteriestand über 50%, Gelb für einen Batteriestand zwischen 50% und 15%, Rot für einen Batteriestand unter 15%).

Für die Audio-Klasse wird eine UART-Kommunikation zwischen den DFPlayer-Modul und ESP32 mit den entsprechenden Pins des ESP32 gestartet und dann an diese Klasse übergeben. Zudem wird die `play()`-Funktion implementiert, um den gewünschten Song von der SD-Karte abzuspielen. Dabei wird jedoch zuerst der BUSY-Pin des DFPlayer-Moduls überprüft, bevor ein Song abgespielt wird.

Anschließend werden die Sensor- und TouchSensor-Klasse initialisiert. Die Sensor-Klasse erfordert ein `Adafruit_VL53L0X`-Objekt, einen Interrupt-Pin, einen XShut-Pin und eine neue I2C-Adresse. Die Sensor-Klasse ändert intern mit Hilfe des XShut-Pins die vordefinierte I2C-Adresse auf die neu ausgewählte Adresse, um mehrere ToF-Sensoren gleichzeitig zu betreiben. Der TouchSensor liest lediglich einen Pin des ESP32 mit Hilfe der `analogRead`-Funktion aus.

Für die Stromversorgung wird die Power-Klasse initialisiert, die zwei Pins für das Einlesen des Batteriestands und das Ausschalten des Satelliten benötigt.

Zuletzt wird die Communication-Klasse initialisiert und nach zwei Sekunden wird jeder Satellit automatisch versuchen, sich mit der Basisstation zu verbinden, sofern alle oben genannten Module erfolgreich gestartet wurden. Beim Verbinden mit dem WLAN-Netzwerk des Raspberry Pi und dem TCP-Port blinken alle LEDs blau bzw. violett.

Nach erfolgreicher Kommunikation sendet jeder Satellit eine Handshake-Nachricht wie oben beschrieben und wartet auf weitere Befehle von der Basisstation. Außerdem wird

jede Minute der aktuelle Batteriestand periodisch an die Basisstation gesendet.

Während des Spiels erhält der Satellit zunächst den Befehl `set_game`, um sich auf das Spiel vorzubereiten. Wenn der Benutzer den Startbutton in der GUI drückt, erhalten alle Satelliten den nächsten Befehl `start_countdown`, um den Countdown-Song abzuspielen. Danach erhält der ausgewählte Satellit den Befehl `wait_swing`, um die Messung für einen Schlag zu starten. Wenn innerhalb eines Zeitraums von 10 Sekunden ein Schlag erfolgt, übermittelt der Satellit die relative Dauerzeit mithilfe eines anderen Befehls `swing-success` an die Basisstation. Falls der Spieler innerhalb von 10 Sekunden keinen Schlag ausführt, sendet der Satellit einen Befehl `swing_failure` für einen Fehlschlag.

Alle oben genannten asynchronen Ereignisse werden mithilfe von `ReactESP` implementiert, das es ermöglicht, verzögerte und periodische Codeblöcke/Funktionen auszuführen. Eine ausführliche Liste aller Befehle wird in Tabelle 1 dargestellt.

Befehl & Richtung		Beschreibung	Argumente
Rasp. PI \Rightarrow ESP32 (Anfrage)	ESP32 \Rightarrow Rasp. PI (Antwort)		
-	handshake	ESP32 sendet nach erfolgreiche TCP-Verbindung.	<code>int id</code> <code>int battery_status</code>
connected	-	Rasp. PI sendet nach erfolgreiche TCP-Verbindung.	-
set_game	game_set	Rasp. PI sendet, nachdem ein Spiel ausgewählt wird.	<code>string game_name</code>
start_countdown	countdown_started	Rasp. PI sendet, nachdem der Startbutton gedrückt wird.	-
wait.swing	swing_success oder swing_failure	Rasp. PI sendet zu ausgewählte Satellit zur Messung.	<code>int maximum_wait_time</code> <code>int relative_reaction_time</code>
-	satellite_status	ESP32 sendet periodisch für den Verbindungs- und Batteriestatus.	<code>int battery_status</code>
set_led	led_set	Rasp. PI sendet, um LEDs zu steuern.	<code>string color (R;G;B)</code>

Tabelle 1: Liste aller Befehle zw. Raspberry PI und ESP32.

7.4 Test der Kommunikation

An erster Stelle wird die Kommunikation unabhängig von anderen Komponenten (LED, Sensorik, Audio) getestet. Die Ergebnisse zeigen, dass die parallelen Verbindungen zwischen der Basisstation und den vier Satelliten reibungslos funktionieren, wobei dieser Testfall über drei Stunden ohne Unterbrechungen durchgeführt wird. Ein weiterer Testfall beinhaltet die Überprüfung der Erkennung von Verbindungsunterbrechungen bei einem Stromausfall, wobei beide Seiten die Unterbrechung erfolgreich erkennen können, jedoch nur, wenn der TCP-Kanal untätig war. Wenn der TCP-Kanal beschrieben wird,

kann die schreibende Seite die Unterbrechung nicht erkennen.

Zu einem späteren Zeitpunkt werden die Kommunikation und die Spiellogik mit allen Komponenten nach einer umfangreichen Debug-Sitzung getestet. Jeder Satellit funktioniert einwandfrei, und sie werden zur Benutzung bereitgestellt.

8 Fazit

Abschließend lässt sich festhalten, dass die erfolgreiche Implementierung und Realisierung der Kommunikation zwischen den Satelliten und der Basisstation ein entscheidender Schritt in unserem Elektronikprojekt war. Die Wahl des Kommunikationsweges mittels WLAN anstelle von Bluetooth erwies sich als sorgfältig durchdacht, da sie eine größere Anzahl gleichzeitiger Verbindungen unterstützt, eine erhöhte Stabilität über längere Entfernungn bietet und eine höhere Bandbreite für eine schnellere Datenübertragung ermöglicht.

Die Integration von Sensoren auf der Satellitenseite ist der Schlüssel zur Erfassung und präzisen Verarbeitung von Daten. Dies ermöglicht nicht nur die Anpassung an unterschiedliche Umgebungen, sondern eröffnet auch neue Anwendungsbereiche durch eine zielgerichtete Interaktion.

Die Entscheidung für die LED-Technologie in unserem Projekt geht über die bloße visuelle Darstellung von Informationen hinaus. Unsere bewusste Wahl von LEDs erweitert die Interaktionsmöglichkeiten und bietet nicht nur eine klare Anzeige von Statusinformationen, sondern auch einen ästhetischen Mehrwert. Dies unterstreicht unsere Vision, nicht nur effiziente, sondern auch ansprechende Lösungen zu schaffen.

Ein weiterer entscheidender Aspekt ist unsere durchdachte Stromversorgung, die das Rückgrat unserer Lösung bildet. Eine zuverlässige Energiezufuhr gewährleistet nicht nur die Kontinuität der Kommunikation, sondern sichert auch den reibungslosen Betrieb aller Komponenten. Unser Fokus auf nachhaltige und zuverlässige Stromversorgung macht unser Projekt nicht nur technisch überzeugend, sondern auch umweltbewusst.

Zusammenfassend repräsentiert unser Elektronikprojekt eine durchdachte Lösung, die verschiedene Aspekte der Elektronik verbessert und dabei einen Mehrwert für Anwender und Umwelt schafft.

Literatur

- (o. D.). URL: [https://github.com/mairas/ReactESP](https://de.aliexpress.com/item/1005004359552599.html?pdp_npi=2&40dis%21EUR%216%2C96%2E2%82%AC%214%2C52%2E2%82%AC%21%21%21%21%21%40211b812f16835616338094631ea130%2112000029049574542%21bt&_t=pvid%3Aea42e089-ecdf-4909-b022-b4b1cb938e41&afTraceInfo=1005004359552599_pc_pcBridgePPC_xxxxxx_1683561634&spm=a2g0o.pclist.product.mainProduct&gatewayAdapt=glo2deu#nav-specification.</p><p>Alaoui, Chakib (Feb. 2011). „Testing and simulation of solid state heating and cooling“. In: <i>International Journal of Engineering Science and Technology</i> 3.</p><p><i>An asynchronous programming library for the ESP32 and other microcontrollers using the Arduino framework.</i> (o. D.). 20.02.2024. URL: <a href=).
- AP63205 Datenblatt* (2019). URL: <https://www.diodes.com/assets/Datasheets/AP63200-AP63201-AP63203-AP63205.pdf>.
- Belovski, Ivaylo u. a. (2017). „Mathematical Model of Thermoelectric Peltier Module“. In: *Journal of Communication and Computer* 14, S. 60–64. DOI: [10.17265/1548-7709/2017.02.002](https://doi.org/10.17265/1548-7709/2017.02.002).
- Bux, Céline (Aug. 2020). *Erbseenkugelfisch (Zwergkugelfisch)*. URL: <https://www.zooplus.de/magazin/aquaristik/fischarten-portraits/zwergkugelfisch>.
- CONVERTER, DC-DC (o. D.). *PEME2-S / DC-DC CONVERTER*. 23.12.2023. URL: https://www.mouser.de/datasheet/2/670/peme2_s-2821195.pdf.
- CUI Devices (o. D.). *Peltier Application Note*. 30.08.2023. URL: <https://www.cuidevices.com/catalog/resource/peltier-app-note.pdf>.
- AZ-Delivery, Shop (o. D.). *AZ-Delivery*. 30.01.2024. URL: <https://www.az-delivery.de/products/vl53l0x-time-of-flight-tof-laser-abstandssensor>.
- DFPlayer (o. D.). *DFPlayer*. 23.12.2023. URL: https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299.
- ESP32-CAM Datenblatt und Anleitung* (2020). SBC-ESP32-CAM. Simac Electronics GmbH. URL: <https://joy-it.net/de/products/SBC-ESP32-Cam>.
- Falkenbach, Oliver (2016). *Bleitellurid*. 30.08.2023. URL: http://geb.uni-giessen.de/geb/volltexte/2016/12109/pdf/FalkenbachOliver_2016_05_17.pdf.
- ILI9341 Datenblatt* (2007). ILI9341. ILI TECHNOLOGY CORP. URL: <https://www.ilitek.com.tw/>.
- Incorporated, Diodes (o. D.). *DMP3098L P-Channel MOSFET*. 14.12.2023. URL: <https://www.diodes.com/assets/Datasheets/ds31447.pdf>.

- Instruments, Texas (Juli 2019). *Driving a Peltier Element (TEC): Efficiency and Aging.* Application Report SLUA979A. Revised January 2020. Texas Instruments. URL: https://www.ti.com/lit/an/slua979a/slua979a.pdf?ts=1693843692633&ref_url=https%253A%252F%252Fwww.google.com%252F.
- (o. D.). *cd4027B CMOS Dual J-K Flip Flop.* 14.12.2023. URL: https://www.ti.com/lit/ds/symlink/cd4027b.pdf?ts=1702549634133&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252Fde-de%252FCD4027B.
- Keyestudio DS18b20 Temperature Detector Sensor* (o. D.). Stand: 09.08.2023. URL: <https://ecksteinimg.de/Datasheet/Keyestudio/KS0316/KS0316.pdf>.
- Keyestudio TDS Sensor* (o. D.). Stand: 30.08.2023. URL: <https://ecksteinimg.de/Datasheet/Keyestudio/KS0429/KS0429.pdf>.
- Lawal, Olarewaju Mubashiru und Zhibing Chang (2021). „Development of an effective TE cooler box for food storage“. In: *Case Studies in Thermal Engineering* 28, S. 101564. ISSN: 2214-157X. DOI: <https://doi.org/10.1016/j.csite.2021.101564>. URL: <https://www.sciencedirect.com/science/article/pii/S2214157X21007279>.
- Model-View-Presenter Design Pattern* (2023). URL: <https://support.touchgfx.com/docs/development/ui-development/software-architecture/model-view-presenter-design-pattern>.
- Multiple DS18B20 sensors using UART* (Juni 2023). URL: <https://controllerstech.com/multiple-ds18b20-sensors-using-uart/>.
- QuickCool, Shop (o. D.). *Peltierelement.* 30.08.2023. URL: <https://quickcool-shop.de/k106/Bibliothek-Peltierelement>.
- Saini, Suryakant (o. D.). URL: <https://drive.google.com/drive/folders/1ZfONWXZUQo7OrEwPo1F1T>
- Santos, Rui (o. D.). *Touch Sensor.* 2019. URL: <https://randomnerdtutorials.com/esp32-touch-pins-arduino-ide/>.
- Scott, Great (o. D.). *DIY LIPO Supercharger Kit.* 01.02.2024. URL: https://cdn.sparkfun.com/assets/0/0/f/c/7/Manual_DIY_LiPo_Supercharger_Kit.pdf.
- Simon Gong Levi Li, Andy Yan (o. D.). *XTAR 18650-2600 Industriezelle Datasheet.* 26.11.2023. URL: https://cdn-reichelt.de/documents/datenblatt/D500/MSDS_18650_2600.pdf.
- TC2117-3.3 Datenblatt* (2010). URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/21665D.pdf>.
- Texas Instruments (o. D.[a]). *BQ24092DGQR.* 23.11.2023. URL: https://www.ti.com/lit/ds/symlink/bq24092.pdf?ts=1700771122094&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252Fde-de%252FBQ24092.

- 252Fwww.ti.com%252Fproduct%252Fde-de%252FBQ24092%252Fpart-details%252FBQ24092DGQR.
- Texas Instruments (o. D.[b]). *TPS2596*. 23.11.2023. URL: https://www.ti.com/lit/ds/symlink/tps2596.pdf?ts=1700733610424&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS2596%252Fpart-details%252FTPS259630DDAR.
- (o. D.[c]). *TPS61085*. 23.11.2023. URL: https://www.ti.com/lit/ds/symlink/tps61085.pdf?ts=1700750241486&ref_url=https%253A%252F%252Fwww.google.com%252F.
- TouchGFX AL Development Introduction* (2023). URL: <https://support.touchgfx.com/docs/development/touchgfx-hal-development/touchgfx-al-development-introduction>.
- UWE electronic (o. D.). *Thermoelectric Modules*. 30.08.2023. URL: <https://www.uweelectronic.de/en/temperaturmanagement-2/peltierelemente/funktionsweise.html>.
- What is TouchGFX?* (2023). URL: <https://support.touchgfx.com/docs/introduction/what-is-touchgfx>.
- XH Connector Datasheet* (o. D.). URL: <https://www.jst-mfg.com/product/pdf/eng/eXH.pdf>.
- Xysemi (o. D.). *XB8089D*. 23.11.2023. URL: https://datasheet.lcsc.com/lcsc/1811151452_XySemi-XB8089D_C79928.pdf.
- Your Gateway to Embedded Software Development Excellence* (o. D.). 20.02.2024. URL: <https://platformio.org/>.
- Zedfy, GmbH (o. D.). *WS2812B*. 2023. URL: <https://www.zedfy.com/led-blog/wie-funktioniert-eine-ws2812b-led>.