

---

# **BACHELORARBEIT**

---

Herr  
**Chengkang Liu**

**PCB-Lötbeispiel mit  
Spielwert: Die Realisierung eines  
RGB-Anhängers**

Mittweida, 2024



**BACHELORARBEIT**

---

**PCB-Lötbeispiel mit Spielwert: Die Realisierung eines RGB-Anhängers**

Autor:  
**Chengkang Liu**

Studiengang:  
**Elektro- und Informationstechnik**

Seminargruppe:  
**EI19sA-BC**

Erstprüfer:  
**Prof. Dr.-Ing. René Pleul**

Zweitprüfer:  
**M.Sc. Markus Süß**

Einreichung:  
**Mittweida, 15. Februar 2024**

Verteidigung/Bewertung:  
**Mittweida, 2024**



### **Bibliografische Beschreibung:**

Chengkang Liu:

PCB-Lötbeispiel mit Spiel-wert: Die Realisierung eines RGB-Anhängers,  
85 Seiten, 34 Abbildungen, Mittweida, Hochschule Mittweida, Fakultät EIT,  
Bachelorarbeit, 2024

### **Referat:**

In dieser Arbeit wird das Design eines RGB-Anhängers auf Basis von WLAN für Intelligente mobile Geräte untersucht. Die Durchführung umfasste die Auswahl und Steuerung von RGB-LEDs, die Analyse und Anwendung von Mikrocontrollern, den Entwurf der Benutzeroberfläche, die Analyse der Stromversorgungsschaltungen, die Optimierung des Energieverbrauchs sowie das PCB-Design. Das Ergebnis ist ein als Lötpraxisprojekt für Kinder konzipierter RGB-Anhänger.

# Inhaltverzeichnis

<b>Inhaltverzeichnis</b> .....	I
<b>Abbildungsverzeichnis</b> .....	III
<b>Tabellenverzeichnis</b> .....	V
<b>Abkürzungsverzeichnis</b> .....	VI
1. Einleitung .....	1
1.1 Motivation .....	1
1.2 Zielsetzung .....	1
1.3 Kapitelübersicht .....	1
2. Festlegung von Spielidee und Lötpraxisarbeit .....	3
2.1 Suche nach der Spielidee der Lötpraxisarbeit .....	3
2.1.1 Marktangebote elektronischer Lötübungen .....	3
2.1.2 Aktuell interessante elektronische Spielideen für Kinder und Jugendliche .....	5
2.1.3 Bewertung und Auswahl der Spielidee .....	6
2.2 Manuelles elektrisches Löten .....	8
2.3 Anforderungen an das Anhängerdesign .....	9
2.4 Lösungskonzept .....	9
3. Auswahl der drahtlosen Kommunikationstechnologie für kurze Entfernungen .....	11
3.1 NFC .....	11
3.2 Bluetooth .....	13
3.3 WLAN und IoT .....	14
3.4 Auswahl der NahfeldkommunikationsLösung .....	16
4. RGB-LED Auswahl .....	17
4.1 Nicht-intelligent RGB-LED .....	18
4.1.1 Hardware .....	18
4.1.2 Kontrollweise .....	19
4.2 Intelligent RGB-LED .....	20
4.2.1 Hardware .....	21
4.2.2 Kontrollweise .....	24
4.3 Vergleich der Vor- und Nachteile zweier RGB-LEDs .....	28
5. ESP8266EX und CODE Entwurf .....	29
5.1 ESP8266EX .....	29

5.1.1 Innere Struktur .....	29
5.1.2 Der Energieverbrauch bei verschiedenen Arbeitsweisen .....	30
5.1.3 Arbeitsmodi .....	31
5.2 Code Entwurf.....	32
5.2.1 Webserver einrichten.....	32
5.2.2 HTML Entwurf und Benutzeroberfläche .....	33
5.2.3 Licht Setup.....	35
6. Optimierungen zur Reduzierung des Stromverbrauchs.....	38
6.1 WLAN-Modus auszuschalten .....	38
6.1.1 Modem-Schlafmodi.....	38
6.1.2 HF-Modul direkt auszuschalten.....	39
6.2 Farbe auszuwählen.....	42
6.3 Andere Möglichkeiten für Energiessparen .....	43
7. Batterieauswahl und Messung der Batterielebensdauer .....	44
7.1 Knopfbatterie .....	44
7.2AAA Batterie .....	47
8. Implementierung der Boost-Schaltung .....	50
8.1Boost-Schaltung Grundlage.....	50
8.2Wahl der Boost-Lösung.....	53
8.3Simulation.....	55
9. Schaltung und PCB Entwurf .....	57
9.1Schaltung .....	57
9.2 PCB Entwurf .....	58
10. Zusammenfassung.....	62
10.1Kostenabschätzung.....	62
10.2Ergebnis .....	62
Literatur.....	64
Anhang.....	65
Selbstständigkeitserklärung .....	73

# Abbildungsverzeichnis

Abbildung 1:Unterschiedliche Produkte.....	4
Abbildung 2 Interessante Spielideen .....	6
Abbildung 3:Blockdiagramm .....	10
Abbildung 4:Farb-Overlay.....	17
Abbildung 5:Pinbelegungsdiagramm 1[3].....	18
Abbildung 6:WP154A4SUREQBFZGW Parameter[3] .....	18
Abbildung 7:RBG Strom-Spannungs-Kennlinie[3] .....	19
Abbildung 8:Pinbelegungsdiagramm 2[4].....	21
Abbildung 9:Verbindungsweise[4].....	22
Abbildung 10:Datenübertragungsweise[4].....	23
Abbildung 11:Elektrische Parameter[4] .....	23
Abbildung 12:Daten Zusammensetzung[4] .....	24
Abbildung 13:Sequenzdiagramm und Datenübertragungszeit[4] .....	25
Abbildung 14:Interne Struktur von ESP8266EX[5].....	29
Abbildung 15:Stromverbrauch von HF-Modul[5].....	30
Abbildung 16:Flussdiagramm.....	42
Abbildung 17:Änderung des Innenwiderstands 1[6].....	45
Abbildung 18:Änderung des Innenwiderstands 2[7].....	46
Abbildung 19:Änderung der effektiven Kapazität[6].....	46
Abbildung 20:Entladekurve 1[8] .....	48
Abbildung 21:Entladekurve 2[8] .....	49
Abbildung 22:Boost konverter Struktur .....	50
Abbildung 23:Prozess des Schaltungsbetriebs[9].....	51



Abbildung 24:Rippelspannung[9].....	52
Abbildung 25:Synchroner Boost-Konverter .....	52
Abbildung 26:Tps613221 Interne Struktur[10].....	53
Abbildung 27:Benutzerdefinierte Daten[11] .....	54
Abbildung 28:Tps613221 Test Schaltung .....	56
Abbildung 29:Fehler .....	56
Abbildung 30:Schaltung .....	57
Abbildung 31:Muster .....	58
Abbildung 32:Bauteilgehäuse verbessern.....	59
Abbildung 33:Layout.....	60
Abbildung 34:Modelle in verschiedenen Farben.....	61

# Tabellenverzeichnis

Tabelle 1 Informationen zu verschiedenen Produkten .....	7
Tabelle 2 Minimale Betriebsspannung .....	24
Tabelle 3 Kostenabschätzung .....	62

## Abkürzungsverzeichnis

RGB-LED	Red-Green-Blue Light Emitting Diode
THT	Through-Hole Technology
SMD	Surface-Mounted Device
MCU	Microcontroller Unit
NFC	Near Field Communication
RFID	Radio-Frequency Identification
IRC	Infrared Communication
WLAN	Wireless Local Area Network
PAN	Personal Area Network
BLE	Bluetooth Low Energy
IoT	Internet of Things
GPIO	General-Purpose Input/Output
PWM	Pulse-Width Modulation
SPI	Serial Peripheral Interface
WiFi	Wireless Fidelity
HF	High Frequency
SRAM	Static Random-Access Memory
CPU	Central Processing Unit
MAC	Media Access Control
PMU	Power Management unit
RTC	Real-Time Clock

SDK	Software Development Kit
AP	Access Point
IDE	Integrated Development Environment
URL	Uniform Resource Locator
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
CSS	Cascading Style Sheets
HSV	Hue, Saturation, Value
EEPROM	Electrically Erasable Programmable Read-Only Memory
DCDC	Direct Current to Direct Current
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
DCR	Direct Current Resistance
CCM	Continuous Conduction Mode
DCM	Discontinuous Conduction Mode
ESR	Equivalent Series Resistance

# 1. Einleitung

## 1.1 Motivation

Mit der Entwicklung von Wissenschaft und Technologie steigt die Nachfrage nach Fortschritten in der elektronischen Technologie von Tag zu Tag. Gleichzeitig steigt auch die Nachfrage nach Talenten, die sich mit elektronischer Technologie auskennen. Bei der Förderung von Talenten gilt es, Stück für Stück anzufangen. Als Ausgangspunkt für das Erlernen elektronischer Technologie ist die Praxis des elektronischen Lötens zweifellos eine gute Wahl. Der Schwerpunkt dieser Arbeit liegt auf der Erstellung einer kleinen Lötarbeit für Kinder, die Interesse wecken kann, sodass junge Praktiker während des Lötprozesses gute Erfahrungen machen und gleichzeitig nach Abschluss des Lötvorgangs tatsächlich eine interessante und praktische elektronische Arbeit erhalten können.

## 1.2 Zielsetzung

Das Ziel der Bachelorarbeit ist es, ein ansprechendes Design für eine Lötpraxisarbeit für Kinder im Alter von 5-12 Jahren zu erarbeiten, die moderne technologische Elemente der Automatisierung und Grundlagen der Elektronik gleichzeitig enthält.

Um diese elektronische Lötpraxisarbeit attraktiv zu gestalten, sollte das Resultat einen hohen Spielwert haben und emotional ansprechend sein. Ein solches Bastel-Produkt soll gefunden und entwickelt werden.

Dabei sind die konkreten Anforderungen an ein solches Bastel-Produkt zu erarbeiten, eine konkrete Lösung zu entwerfen, auszuarbeiten und zu optimieren. Im Ergebnis soll ein entsprechender Bausatz zusammengestellt werden.

Alle für die Lötarbeit sowie für die Zusammenstellung des Bausatzes erforderlichen Dokumente sowie eine Kostenaufstellung sollen erarbeitet werden. Auf deren Grundlage soll es möglich sein, eine kindgerechte Anleitung für das Löten und die Benutzungsanleitung des Bastel-Produktes zu generieren.

## 1.3 Kapitelübersicht

Im Kapitel 2 nach der Einleitung stellt den technischen Stand des elektronischen Lötens und elektronischer Lötpraxisarbeiten vor, ermittelt die Anforderungen von RGB-Anhängern als elektronische Lötpraxisarbeiten und legt die Grundkonzepte ihrer Zusammensetzung fest.

Im Kapitel 3 wird die Wahl der drahtlosen Kommunikationstechnologie für kurze Entfernungen für diese Arbeit vorgestellt. Auf dieser Grundlage wurden die Auswahl der drahtlosen Kommunikationstechnologie und des Mikrocontrollers (MCU) festgelegt.

Im Kapitel 4 werden die Unterschiede zwischen intelligenten und nicht-intelligenten RGBLEDs in Bezug auf Hardware und Steuerungsmethoden beschrieben. Auf Basis der Analyse wurde die Auswahl für RGBLEDs getroffen.

Im Kapitel 5 wird die interne Struktur und Arbeitsweise des MCU erläutert und den Code-Entwurf für die Benutzeroberfläche und die Steuerung der RGBLEDs geschaffen.

Im Kapitel 6 werden die Optimierungsmethoden für den Energieverbrauch behandelt und die Arbeiten zur Optimierung des Energieverbrauchs abgeschlossen.

Im Kapitel 7 wird die Auswahl der Batterien und die Abschätzung der Batterielaufzeit eingeführt.

Im Kapitel 8 wird die Implementierung der Boost-Schaltung beschrieben, um die Batterielaufzeit zu verlängern.

Im Kapitel 9 wird die Schaltung vorgestellt, und das PCB-Entwurf, das für das Lötens in dieser Arbeit geeignet ist.

im Kapitel 10 wird mit einer Zusammenfassung der Kosten und einer Bewertung der Ergebnisse die Arbeit abgeschlossen.

## 2. Festlegung von Spielidee und Lötpraxisarbeit

### 2.1 Suche nach der Spielidee der Lötpraxisarbeit

In die Überlegungen für eine geeignete Spielidee werden verschiedene Aspekte einbezogen. Diese sind:

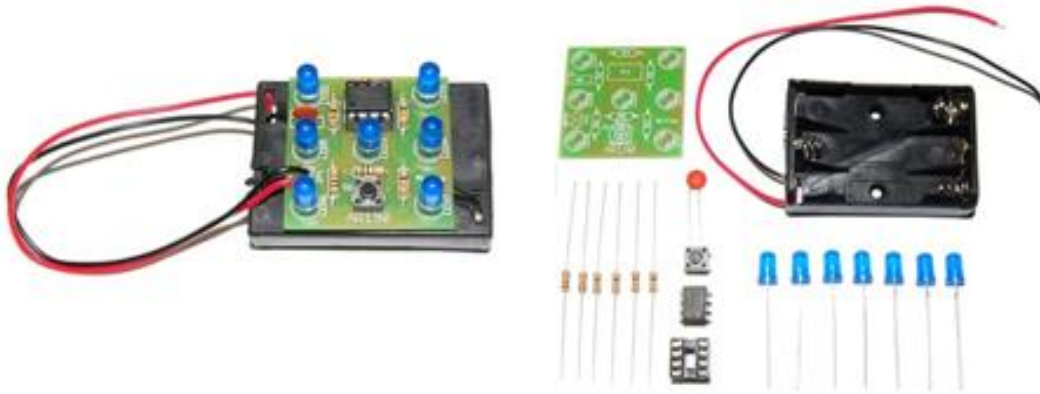
- Bestehende, am Markt verfügbare Angebote für elektronische Bausätze,
- Bekannte emotional ansprechende Spiele für Kinder und Jugendliche,
- Das aktuelle technische Umfeld von Kindern und Jugendlichen
- Mögliche Kosten des Bausatzes

#### 2.1.1 Marktangebote elektronischer Lötübungen

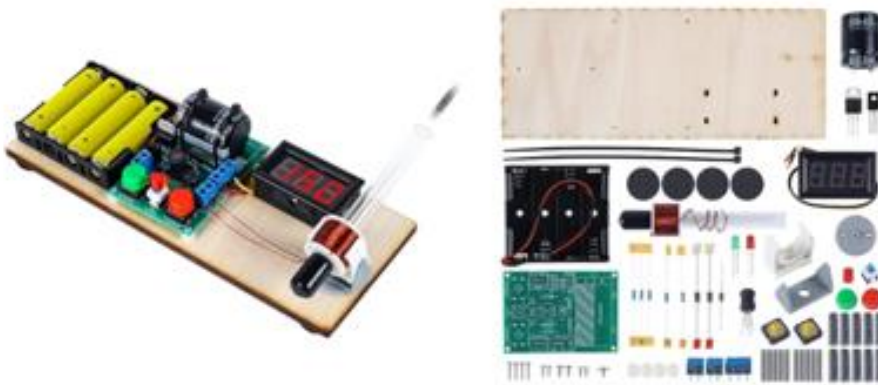
Derzeit gibt es viele Produkte für Lötübungen im Zusammenhang mit elektronischem Heimwerken, und auch ihre Themenkategorien sind sehr umfangreich. Folgende emotional ansprechende und attraktive Beispiele für Lötpraxisarbeiten stehen online als Bausätze zum Kauf zur Verfügung und dienen als Benchmark:

- a) Elektronischer Würfel,
- b) Elektromagnetische Mini-Wurfanlage
- c) Elektronische Uhren
- d) Ultraschall-Schwebe-Einrichtung

Beispielangebote für Bausätze sind in Abbildung 1 dargestellt.



(a) Elektronischer Würfel



(b) Elektromagnetische Mini-Wurfanlage



(c) Elektronische Uhren



(d) Ultraschall-Schwebe-Einrichtung

Abbildung 1: Unterschiedliche Produkte



- 1(a): Der elektronische Würfel verfügt über 7 LEDs und zeigt die Zufallszahl von 1 bis 6 an. Durch Drücken des Druckknopfes beginnen die LEDs unkontrolliert zu blinken. Durch das Blinken der LEDs wird das Würfeln simuliert. Das Modul ist klein und tragbar. Die Stromversorgung erfolgt über 3 AAA-Batterien, die am Batteriehalter befestigt sind.
- 1(b): Die elektromagnetischen Mini-Wurfanlage umwandelt elektrische Energie in magnetische Energie und dann kann sie einen Metallzylinder einspritzen. Die Schaltung wird mit 6 V Gleichstrom betrieben. Es nutzt die durch das elektromagnetische Feld erzeugte „Amperekraft“, um den Metallstab zu beschleunigen.
- 1(c): Die elektronische Uhr kann Uhrzeit, Datum (Jahr, Monat, Tag, Woche) und Temperatur anzeigen. Die Helligkeit kann so angepasst werden, dass sie sich automatisch anpasst, oder man kann die Helligkeit manuell anpassen. Mit Ausschaltspeicher und Alarmfunktion und enthält 4 einstellbare Musik. Die 4-stellige elektronische Digitaluhr kann auch die Temperatur genau messen.
- 1(d): Die ultraschall-Schwebe-Einrichtung arbeitet mit einer Frequenz von 40 kHz in der Luft und kann leichte Objekte mit einer Dichte von 2-3 mm erfassen, z.B. Schaumstoffbälle. Nachdem die Einrichtung eingeschaltet wurde, wird das Objekt am Knotenpunkt der stehenden Welle einer akustischen Kraft ausgesetzt, die die Schwerkraft überwindet und letztlich eine Levitation erreicht. Es ist aber zu beachten, dass eine 12V Gleichstromversorgung vom Benutzer für das Lötpraxisarbeiten bereitgestellt werden muss.

### **2.1.2 Aktuell interessante elektronische Spielideen für Kinder und Jugendliche**

Ein z.B. in China beliebtes Spielzeug ist „Elektro-Schlag-Spielzeug“. Es wird in Abbildung 2(a) dargestellt. Dieses Spielzeug mit Streichcharakter nutzt Finger als Leiter, wobei eine interne Boost-Schaltung die Spannung erhöht und einen sehr kurzen Stromschlag auslöst. Aufgrund der sehr kurzen Kontaktzeit und der begrenzten Leistung des Stromkreises ähnelt der Schock eher einem statischen Schock. Dieses Spielzeug kommt dem Wunsch einiger Menschen entgegen, ihren Freunden Streiche zu spielen, und wird wegen seiner Fähigkeit, zur sozialen Interaktion beizutragen, geschätzt.

Es wäre denkbar, dass eine solche Lötaufgabe für Kinder interessant ist, wenn Sicherheitsfragen berücksichtigt werden. Es wird zurzeit keine Lötpraxisarbeit für eine solches Spielzeug angeboten. Diese Spielidee wird deshalb als möglicher, zu entwickelnder Lötpraxisarbeit „e)“ geführt.



Abbildung 2 Interessante Spielideen

Ein für Kinder und Jugendliche emotional ansprechendes Element ist Licht in verschiedenen Farben. Es wird in Abbildung 2(b) dargestellt. Eine Überlegung ist deshalb, einen in Lichtfarbe und Blinken steuerbaren Anhänger auf Basis von RGB-LED zu entwickeln.

Durch die Programmierung von RGB-LED kann man versuchen, externe Tasten oder andere Methoden zu verwenden, um die Farbe und Helligkeit von RGB-LED zu steuern. Basierend auf der Kombination mehrerer RGB-LEDs können auch einige wunderschöne Farbmodi oder bestimmte Funktionen entworfen werden.

Eine Lötpraxisarbeit für einen solchen Anhänger gibt es ebenfalls nicht. Eine mögliche Entwicklung wird als Option „f“ in Abschnitt 2.1.3 geführt.

### 2.1.3 Bewertung und Auswahl der Spielidee

Die gefundenen Ideen und Benchmarks sind in Tabelle 1 zusammengefasst und bewertet. Unter ihnen ist Spielwert eine subjektive Bewertung basierend auf Verkaufszahl und Funktionalität, und Zeit Aufwand sowie Schwierigkeit sind Bewertungen basierend auf der Anzahl und der Lötbarkeit der gelöteten Komponenten.

Produkt	Spielwert (1-5)	Zeitaufwand (Stunde)	Schwierigkeitsgrad (1-5)	Preis (€)
a	3	0,5	2	12,2
b	4	2,5	4	31,99
c	2	2,5	4	13,99
d	3	1,5	3	19,99
e	5	möglich 2	möglich 3	Produkt 10,99
f	5	möglich 1	möglich 2	unsicher

Tabelle 1 Informationen zu verschiedenen Produkten

- a) Für einen elektronischen Würfel wird ein Spielwert von 3 vergeben. Es hat nur eine Funktion und hat geringe Verkaufszahlen. Daher würde der Spielwert nur bei 1 bis 2 liegen. Jedoch gibt es viele Spiele, bei denen Kinder diesen Würfel nutzen könnten. Es fehlt bei diesem Produkt andererseits der Bezug zum modernen technologischen Umfeld der Kinder, welches z.B. Tablets und Smartphones sind. Solche Geräte bieten bereits interessante Apps für das Würfeln. Deshalb wird der Spielwert nur auf 3 festgelegt.
- b) Für eine Elektromagnetische Mini-Wurfanlage wird ein Spielwert von 4 vergeben. Es kann kleine Metalle mit sehr hoher Geschwindigkeit abfeuern, was ziemlich wirkungsvoll ist. Allerdings gibt es zu viele Bauteile, was zu langen Löt- und Montagezeiten führt. Da die Spannung nach dem Boosten bis zu 200 V erreichen kann, müssen gleichzeitig vor dem Löten und während der Verwendung die Vorsichtsmaßnahmen in der Bedienungsanleitung strikt befolgt werden. Beispielsweise muss der Energiespeicherkondensator vor dem Löten entladen werden, Metallteile dürfen nicht entladen werden während des Gebrauchs usw. berührt werden. Darüber hinaus ist es aufgrund der großen Anzahl an Komponenten auch preislich am teuersten.
- c) Für Elektronische Uhren wird ein Spielwert von 2 vergeben. Dies ist ein sehr praktisches Elektronik-DIY-Projekt, das aber keinen großen Spaß macht. Für Kinder hat eine einfache elektronische Uhr keinen großen Reiz. Gleichzeitig besteht es aus zu vielen Teilen und ist nicht für Kinder ohne Lötterfahrung geeignet.
- d) Für eine Ultraschall-Schwebe-Einrichtung wird ein Spielwert von 5 vergeben. Dies ist ein großartiges Lernspielzeug für Kinder, die neugierig auf physikalische Phänomene sind. Da jedoch eine 12-V-DC-Stromversorgung im Voraus vorbereitet werden muss, ist dies nicht sehr benutzerfreundlich. Gleichzeitig ist der Preis relativ hoch.
- e) Für ein Elektro-Schlag-Spielzeug wird ein Spielwert von 5 vergeben. Da Spielzeuge, die Elektrizität nutzen, in der Regel spannender sind, ziehen sie jüngere Kinder mit ausgeprägter Neugier an und wecken so das Interesse am Löten.
- f) Für ein RGB-Anhänger wird ein Spielwert von 3 vergeben. Basierend auf den programmierbaren Eigenschaften von RGB-LED können Farbmodi und Funktionsmodi hinzugefügt werden, um den Spielwert des Anhängers zu verbessern. z.B. bei Verwendung von 6 LEDs kann es auch als elektronischer Würfel fungieren. Es kann so betrachtet werden, dass der Spielwert im Wesentlichen durch die Entwicklung von Farbe und Funktionalität auf Softwareebene bestimmt wird. Mit solchen Potenzialen kann sich der Spielwert auf 5 erhöhen.

Es ist nicht schwer festzustellen, dass aktuelle Lötpraxisarbeiten im Allgemeinen die Merkmale eines geringen Spielwerts oder aufwendig sind und einer relativ einfachen Funktionalität aufweisen. Wenn die Anzahl der Elemente zunimmt, erhöhen sich die Lötzeit, der Schwierigkeitsgrad und der Preis entsprechend. Daher kann diese Arbeit die Probleme

der derzeit auf dem Markt befindlichen Lötpraxisarbeiten als Referenz für die Gestaltung von Lötpraxisarbeiten nutzen.

In Bezug auf die beiden derzeit vorgeschlagenen Ideen, nämlich Elektro-Schlag-Spielzeug und RGB-LED Anhänger. Zwar Elektro-Schlag-Spielzeug eine interessante Idee ist, aber bei der Planung von Lötarbeiten müssen jedoch auch die Sicherheit des Lötens berücksichtigen. Beispielsweise ist es etwas gefährlich, Kinder Teile von Hochspannungsentladungskreisen löten zu lassen. Dieses Produkt eignet sich eher für die direkte Verwendung als Fertigprodukt als für das Selbstlöten. Im Vergleich dazu ist RGB-Anhänger eine idealere Idee. Es hat ein großes Spielpotenzial, hauptsächlich abhängig vom Entwicklungsstand von RGB-LED. Und indem die Anzahl der Komponenten so weit wie möglich reduziert wird, können Lötzeit, Schwierigkeit und Kosten geringer gehalten werden. Zusätzlich kann durch Einsatz z.B. eine Mikrocontrollers und einer Drahtloskommunikation mit Tablet oder Smartphone der Spielwert weiter erhöht werden. Darüber hinaus ist das Muster Von Anhänger auch flexibel einstellbar, z.B. einige bei Kindern beliebte Elemente wie Tiere oder Roboter. Daher ist RGB-Anhänger eine gute Wahl für die Entwicklung dieser Arbeit. Und als Schlussfolgerung wird einen solchen Anhänger mit Drahtloskommunikation und 6 LED als Thema ausgewählt.

## 2.2 Manuelles elektrisches Löten

Bei manuellem elektronischem Löten werden elektronische Bauteile mit Hilfe eines HandlötKolbens und Lot auf eine Leiterplatte gelötet. Der Lötprozess erfordert manchmal die Verwendung von Flussmitteln, um die Lötstellen zu reinigen, Oxide zu entfernen und das Fließen sowie Anhaften des Lots zu unterstützen. Da gleichzeitig die Temperatur des LötKolbens, die Erwärmungszeit der Kontakte an der Lötstelle und der Lotfluss kontrolliert werden müssen, entwickelt sich beim Löten eine präzise Hand-Augen-Koordination sowie detaillierte Bedienfähigkeiten.

Da Kinder in der Regel an Löterfahrung und flexiblen Steuerungsmöglichkeiten mangeln, kann der LötKolben beim Löten zu lange mit den Bauteilen in Berührung bleiben, was zu Schäden an den Bauteilen führen kann. Daher sollte man bei der Auswahl der Komponenten besonders darauf achten, robuste Komponenten auszuwählen, die nicht leicht beschädigt werden können. Gleichzeitig sollten die Vorsichtsmaßnahmen beim Löten im Voraus hingewiesen werden.

Der lötbare Bereich des manuellen Lötens beschränkt sich normalerweise auf durchkontaktierte Komponenten (THT) und einige oberflächenmontierte Komponenten (SMD). Unter diesen sind THT-Komponenten einfacher zu löten, während das manuelle Löten von SMD-Komponenten, insbesondere für extrem kleine Komponenten wie 0402 oder kleiner, ein höheres Maß an technischem Know-how erfordert. Wenn daher aufgrund von Designanforderungen einige SMD-Komponenten für Lötübungen ausgewählt werden, müssen mindestens größere Größen in Betracht gezogen werden, z. B. 0805 (2,0 mm x 1,25 mm) oder 1206 (3,2 mm x 1,6 mm). Wenn SMD-Komponenten tatsächlich verwendet werden, ist es auch möglich, Lötübungen für SMD-Komponenten hinzuzufügen, sofern

beim Anwender bereits ein entsprechendes Gerät (z.B. eine Heißluftpistole) verfügbar ist. Oder können SMD-Komponenten auch direkt als vorgelötete Teile verwendet werden.

### 2.3 Anforderungen an das Anhängerdesign

Basierend auf der obigen Diskussion müssen auch einige spezifische Anforderungen an den Anhänger ermittelt werden.

Die Hauptanforderungen an die Konstruktion sind niedrige Kosten, einfache Lötbarkeit, geringe Größe, ansprechendes Erscheinungsbild und ein gewisses Maß an Bedienfreundlichkeit, sodass auch Kinder ohne elektronische Kenntnisse die Lötarbeiten problemlos durchführen und steuern können. Aufgrund der oben genannten Anforderungen ist die Auswahl der Komponenten von entscheidender Bedeutung geworden, und auch die Art und Weise, wie mit Benutzern interagiert wird, ist zu einem Schwerpunkt geworden.

Aus Sicherheitsgründen ist sicherzustellen, dass keine gefährlichen Bauteile verwendet werden und der Lötende im Voraus über relevante Sicherheitsvorkehrungen informiert wird. Insbesondere während des Betriebs auf einer Rennstrecke dürfen keine Risiken entstehen.

Das Gewicht des Anhängers spielt eine entscheidende Rolle für das Benutzererlebnis. Eine Reduktion der Komponentenanzahl sowie der Einsatz leichterer Bauelemente sind effektive Maßnahmen zur Gewichtsminimierung.

Es wird empfohlen, vorzugsweise THT-Komponenten zu verwenden. Sollten SMD-Komponenten notwendig sein, ist eine vorherige Vorbereitung anzustreben. Zudem sollten im Layoutprozess die Größe und der Abstand der Pads so angepasst werden, dass das Löten erleichtert wird.

Bei der Auswahl von MCU und Batterien ist auf ein optimales Preis-Leistungs-Verhältnis zu achten, um teurere Komponenten zu vermeiden und die Kosten zu senken.

Neben der funktionalen Umsetzung sollte ebenfalls über die Integration von Produktmerkmalen nachgedacht werden, die das Erscheinungsbild aufwerten und die Bedienung intuitiv und einfach gestalten.

### 2.4 Lösungskonzept

Basierend auf der oben durchgeführten Analyse muss das Arbeitslogikdiagramm zusätzlich definiert werden. Im Allgemeinen ist mindestens ein Netzteil für die Stromversorgung, einige RGB-LEDs zur Anzeige verschiedener Farben und eine Mikrocontroller-Einheit (MCU) zur Steuerung der RGB-LEDs und zur Interaktion mit dem Benutzer erforderlich.

Die Steuerung von RGBLEDs erfordert in der Regel das Eingeben von Farbsignalen und Zeitsequenzsignalen, jedoch hängt die spezifische Steuerungsmethode hauptsächlich vom Modell der verwendeten RGBLEDs ab.

Und um die Benutzerfreundlichkeit zu erhöhen, kann die Steuerung der Anhänger über die in der Nähe des Benutzers befindlichen intelligenten Geräte (wie Smartphone, Tablets usw.) mittels drahtloser Kommunikation mit dem MCU realisiert werden. Auf dieser Basis ist eine noch idealere Vorstellung die Entwicklung eines Benutzerinterfaces, um die Steuerung intuitiver und benutzerfreundlicher zu gestalten.

Ausgehend von diesen Anforderungen kann ein grobes konzeptionelles Blockdiagramm, wie in Abbildung 3 dargestellt, erstellt werden.

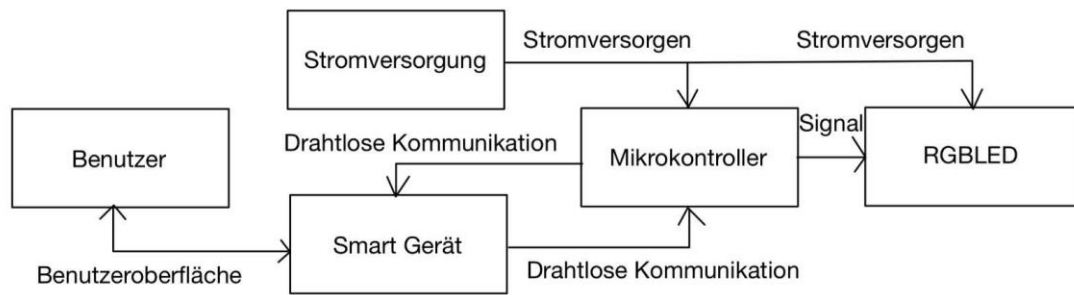


Abbildung 3:Blockdiagramm

### 3. Auswahl der drahtlosen

## Kommunikationstechnologie für kurze Entfernungen

Als Anhänger wird er zweifellos um den Hals getragen. Das bedeutet, dass die bequemste Art, es zu steuern, nicht darin besteht, nach Hause zu gehen, es an einen Computer anzuschließen, um seine RGB-Farben zu konfigurieren, und es dann herauszunehmen. Vielmehr lässt sich der Anhänger direkt vor Ort über physische Tasten am Anhänger selbst oder über intelligente Endgeräte wie Smartphones steuern. Mit anderen Worten, die Steuerung des Anhängers ist jederzeit möglich, solange er sich innerhalb einer gewissen Entfernung zum steuernden Gerät befindet. Dies impliziert, dass die Kommunikation mit dem RGB-Anhänger über eine drahtlose Technologie mit kurzer Reichweite stattfindet.

Die folgenden Arten von Nahbereichskommunikationstechnologien sind derzeit weit verbreitet:

- NFC (Near Field Communication)
- Bluetooth
- RFID (Radio Frequency Identification)
- IRC (Infrared Communication)
- WLAN (Wireless Local Area Network)

Unter diesen Technologien liegen RFID- und IRC-Technologien normalerweise nicht in dem Bereich, der von tragbaren Smart-Geräten wie Mobiltelefonen unterstützt wird. Daher werden in diesem Kapitel hauptsächlich NFC-, Bluetooth- und WLAN-Technologien vorgestellt und analysiert und schließlich die geeignete Technologie für die Entwicklung von RGB-Anhängern ausgewählt.

#### 3.1 NFC

Die NFC-Technologie (Near Field Communication) ist eine drahtlose Kommunikationstechnologie mit kurzer Reichweite, die die Interaktion zwischen elektronischen Geräten ermöglicht. Die Technologie, die üblicherweise in Mobiltelefonen und ähnlichen Smart-Geräten zum Einsatz kommt, kann Daten über eine Entfernung von nur

wenigen Zentimetern sicher übertragen. Hier sind einige wichtige Aspekte der NFC-Technologie:

- Funktionsprinzip: NFC basiert auf der Radiofrequenz-Identifikationstechnologie (RFID) und überträgt Daten zwischen Geräten durch elektromagnetische Induktionskopplung. Das bedeutet, dass ein Gerät (z. B. ein Smartphone) mit einem anderen Gerät (z. B. einem Zahlungsterminal) kommunizieren kann, ohne dass eine physische Verbindung erforderlich ist.
- Anwendungsszenarien: NFC ist weit verbreitet. Zu den häufigsten Anwendungen gehören kontaktloses Bezahlen (wie Apple Pay und Google Pay), elektronisches Ticketing (wie in öffentlichen Verkehrsmitteln), intelligente Türschlösser und vereinfachte Bluetooth-Kopplung. Es wird auch häufig zur Überprüfung der persönlichen Identität und zum Datenaustausch verwendet.
- Vergleich mit anderen Technologien: Im Vergleich zu anderen drahtlosen Technologien wie Bluetooth und Wi-Fi zeichnet sich NFC durch einen geringen Stromverbrauch, eine niedrige Übertragungsrate und eine Übertragung über kurze Entfernungen aus. Dadurch eignet sich NFC ideal für Anwendungen, die keine Datenübertragung über große Entfernungen oder Hochgeschwindigkeit erfordern oder nicht tolerieren können.

Bei RGB-Anhängern ist der größte Vorteil der NFC-Technologie der geringe Stromverbrauch. Ein Beispiel für einen kosteneffizienten NFC-Chip ist der RF430FRL15xH von Texas Instruments, der im passiven Modus keine externe Energiequelle benötigt, da er die benötigte Energie direkt aus dem Feld des NFC-Lesegeräts beziehen kann. Im Standby-Modus (Low-Power-Modus LPM3) weist der Chip einen äußerst niedrigen Energieverbrauch von nur 16  $\mu\text{A}$  auf. Im aktiven Modus der NFC-Kommunikation ist der Energieverbrauch ebenfalls gering und beträgt nur 140  $\mu\text{A}/\text{MHz}$  bei 1,5 V. [1]

Die Grenzen der NFC-Technologie sind jedoch ebenso offensichtlich:

- Entfernungsbegrenzung: Die Effektivität der NFC-Technologie beschränkt sich auf eine sehr geringe Reichweite, normalerweise nur einige Zentimeter. Das erfordert, dass der Benutzer sein Telefon oder ein anderes NFC-fähiges Gerät zum Kommunizieren sehr nahe an das NFC-Tag oder den NFC-Leser halten muss.
- Datenübertragungsrate: Die Übertragungsgeschwindigkeit von NFC ist im Vergleich zu anderen Technologien relativ gering. Dies mag für die Übertragung einfacher Befehle oder Identifikatoren ausreichend sein, ist jedoch für Anwendungen, die höhere Datenübertragungsraten benötigen, unter Umständen nicht geeignet.



- Beschränkung auf einzelne Aufgaben: NFC ist optimal für einzelne, einfache Interaktionen wie das Ändern von Einstellungen oder das Auslösen bestimmter Aktionen. Für Aufgaben, die kontinuierliche Kontrolle oder Echtzeit-Feedback benötigen, ist NFC weniger geeignet.

Die Nutzung eines RGB-Anhängers durch NFC ermöglicht Steuerungsvorgänge ausschließlich bei unmittelbarer Nähe des Smartphones zum NFC-Tag, was in vielen Anwendungsfällen als nicht praktikabel angesehen wird. Der NFC-Chip allein ist für umfassende Steuerungsaufgaben unzureichend und bedarf der Integration mit anderen Mikrocontroller-Einheiten (MCUs), um erweiterte Funktionen zu ermöglichen. NFC wird primär zur Verbesserung des Nutzererlebnisses durch berührungsbasierte Aktivierung eingesetzt und nicht als alleinige Steuerungslösung. Für eine vollständige Steuerung eines RGB-Anhängers ist daher die Kombination von NFC mit anderen Technologien wie Bluetooth oder WLAN erforderlich, was jedoch die Entwicklungsressourcen und -komplexität erhöht. Die direkte Anwendung alternativer Kommunikationstechnologien könnte unter Berücksichtigung der Kosten und des Aufwands eine effizientere Lösung darstellen.

### 3.2 Bluetooth

Bluetooth-Technologie ist ein drahtloser Kommunikationsstandard, der den Austausch von Daten über kurze Entfernungen sowohl zwischen stationären und mobilen Geräten als auch zwischen mobilen Geräten ermöglicht. Die Übertragung erfolgt über das 2,402 bis 2,48 GHz Funkfrequenzband, wobei ein Personal Area Network (PAN), bekannt als „Piconetz“, über eine drahtlose Kurzstreckenverbindung erstellt wird. Ein solches Netzwerk kann ein Master-Gerät und mehrere Slave-Geräte umfassen, die auf die Frequenzsprung-Spreizspektrum-Technologie setzen, um Interferenzen zu minimieren und die Sicherheit der Kommunikation zu gewährleisten.

Die Hauptvorteile von Bluetooth umfassen ein energieeffizientes Design und die Plug-and-Play-Fähigkeit, die keine komplexen Einrichtungsprozesse für die Vernetzung persönlicher Geräte erfordert. Bluetooth-Geräte sind zudem kompakt und leicht, was ihre Portabilität und Benutzerfreundlichkeit erhöht. Seit der ersten Einführung durch Ericsson im Jahr 1994 hat Bluetooth mehrere Entwicklungsstufen durchlaufen, wobei jede Version Verbesserungen in Bezug auf Übertragungsgeschwindigkeit, Sicherheit und Energiemanagement mit sich brachte. Aktuell hat sich Bluetooth zur Version 5.3 weiterentwickelt, die eine signifikante

Verbesserung in Geschwindigkeit, Reichweite, Broadcast-Kommunikationsfähigkeiten und der Verbindung zwischen IoT-Geräten bietet.

Bei der Weiterentwicklung der Bluetooth-Technologie stellt Bluetooth Low Energy (BLE) einen bedeutenden Fortschritt dar. Diese Technologie, eingeführt mit Bluetooth 4.0, ist für ihre energieeffiziente Datenübertragung bekannt und arbeitet im lizenzfreien 2,4-GHz-ISM-Band. Im Vergleich zum klassischen Bluetooth zeichnet sich BLE durch einen deutlich geringeren Energieverbrauch aus und ist insbesondere für Anwendungen konzipiert, bei denen eine lange Batterielebensdauer essentiell ist.[2]

Obwohl der geringe Energieverbrauch für RGB-Anhänger von großer Bedeutung ist, müssen auch andere Designaspekte berücksichtigt werden. Der ESP32, als ein kosteneffizienter und leistungsfähiger Chip, der BLE unterstützt, erfüllt die Anforderungen an niedrigen Energieverbrauch und kompakte Größe dieser Arbeit. Jedoch ist die Steuerung durch das Senden von Befehlen weniger benutzerfreundlich und intuitiv. Zudem stellt die ausschließliche Verfügbarkeit in SMD-Bauform eine Herausforderung für die Lötfreundlichkeit dar, insbesondere in Projekten, die eine einfache Handhabung priorisieren. Trotz seiner hohen Integration und der Verfügbarkeit zahlreicher GPIO-Ports, werden in diesem Projekt wahrscheinlich nur wenige davon genutzt, was auf eine mögliche Überdimensionierung für einfache LED-Farbsteuerungsprojekte hinweist.

### 3.3 WLAN und IoT

WLAN (Wireless Local Area Network) ist eine weit verbreitete Technologie für drahtlose Kommunikation, die es ermöglicht, über Funkfrequenzen ein Computernetzwerk ohne die Notwendigkeit physischer Kabelverbindungen zu etablieren. Im Wesentlichen ersetzt WLAN in einem Local Area Network (LAN) die kabelgebundenen Übertragungsmedien durch drahtlose Kanäle, hauptsächlich im 2,4-GHz- oder 5-GHz-Frequenzband, entsprechend der IEEE 802.11-Serie von Standards. Diese Standards definieren die technischen Spezifikationen für Hochgeschwindigkeits-WLANs, die für die meisten drahtlosen Netzwerkanwendungen im täglichen Leben verwendet werden.

Im Laufe seiner Entwicklung hat WLAN verschiedene technische Standards hervorgebracht, wobei Wi-Fi aufgrund seiner einfachen Implementierung, zuverlässigen Kommunikation, hohen Flexibilität und vergleichsweise niedrigen Kosten zum dominierenden Standard geworden ist. Wi-Fi ist allmählich zum Synonym für WLAN-Technologie im Allgemeinen

geworden, obwohl es streng genommen nur eine von mehreren Implementierungen der IEEE 802.11-Standards darstellt.

Das Internet der Dinge (IoT) ist eng mit WLAN-Technologie verbunden. Es vernetzt alltägliche Objekte und Geräte, sodass diese Daten sammeln, austauschen und kommunizieren können. Durch die Integration von IoT und WLAN profitieren IoT-Geräte von der drahtlosen Kommunikation, was eine einfache Verbindung zum Netzwerk und den Austausch von Daten ermöglicht. Diese Synergie gilt als ein Kernpunkt in der Entwicklung moderner Technologien.

Da viele IoT-Geräte batteriebetrieben sind und ein Batteriewechsel oft nicht einfach ist, sind energiesparende Verbindungstechnologien besonders wichtig. Hier kommt die WLAN-Technologie IEEE 802.11ah ins Spiel. Unter dem Namen „HaLow“ bekannt, wurde sie speziell für die Anforderungen des IoT entwickelt, mit einem Fokus auf niedrigen Energieverbrauch. IEEE 802.11ah verwendet das 900-MHz-Funkband, welches eine bessere Signalpenetration durch Hindernisse wie Wände bietet und somit eine umfassendere Abdeckung ermöglicht. HaLow optimiert den Energieverbrauch durch verschiedene Ansätze: Verbesserung des Schlafmodus, Anpassung der Datenübertragungsraten, Vereinfachung der Kommunikationsprotokolle, effizientes Bandbreitenmanagement und optimierte Netzwerkstruktur, was es zu einer idealen Lösung für IoT-Anwendungen macht. Da diese Technologie jedoch noch relativ neu ist, ist die Anzahl der Mikrocontroller-Einheiten (MCUs), die IEEE 802.11ah (Wi-Fi HaLow) unterstützen, derzeit noch begrenzt. Da Wi-Fi HaLow in Verbraucherprodukten noch keine weite Verbreitung gefunden hat, untersucht diese Arbeit die Möglichkeit, andere MCUs zu nutzen, die sowohl kostengünstig als auch energieeffizient sind.

Ein Beispiel hierfür ist der ESP8266EX, ein beliebter und preiswerter WLAN-Mikrocontroller-Chip, der von Espressif Systems entwickelt wurde. Seit seiner Markteinführung im Jahr 2014 hat der Chip aufgrund seiner hohen Leistungsfähigkeit bei gleichzeitig niedrigen Kosten schnell an Popularität im Bereich von IoT gewonnen.

ESP8266EX bietet drei Arbeitsmodi mit WLAN: den STA-Modus, den AP-Modus und den STA\_AP-Mischmodus. Im STA-Modus dient der ESP8266EX als Wi-Fi-Client, der Verbindungen zu bestehenden Wi-Fi-Netzen herstellen kann. Im AP-Modus funktioniert der ESP8266EX als WLAN-Zugangspunkt und erlaubt anderen WLAN-Geräten, sich mit ihm zu verbinden. Der STA\_AP-Mischmodus vereint die Fähigkeiten des STA- und AP-Modus, ermöglicht dem Chip, gleichzeitig als Client und Zugangspunkt zu agieren. Besonders im

AP-Modus kann der GPIO-Port des ESP8266EX direkt über eine Webseite gesteuert werden, indem man auf das Standard-Gateway zugreift, was eine intuitive Steuerungsmethode bietet. ESP8266EX unterstützt mehrere Schlafmodi, die unter verschiedenen Umständen einen niedrigen Stromverbrauch ermöglichen und somit zur Energieeinsparung beitragen. Darüber hinaus ist der ESP8266EX weitgehend modularisiert, wobei ESP01S das kleinste und leistungsstärkste WLAN-Modulmodell darstellt. Es handelt sich um eine verbesserte Version des vorherigen Modells ESP01 mit optimiertem Antennendesign und 1 MB FLASH, was die Entwicklung dieser Arbeit erheblich unterstützt. Mit lediglich 2 GPIO-Ports ist ESP01S besonders für kleinere Projekte wie RGB-Anhänger geeignet. Bezüglich der Lötbarkeit zeichnet sich ESP01S als ein Plug-in-WiFi-Modul aus, das sich durch THT darstellen lässt. Mit nur 8 Pins eignet es sich ausgezeichnet als Objekt für Lötübungen.

### 3.4 Auswahl der NahfeldkommunikationsLösung

Dieses Kapitel diskutiert und analysiert NFC, Bluetooth und WLAN - drei Kommunikationstechnologien - hinsichtlich ihres Übereinstimmungsgrades mit den Entwicklungsanforderungen dieser Arbeit.

Aus Kostengründen erscheint WLAN als die vorzuziehende Technologie gegenüber Bluetooth, da es eine kosteneffizientere Lösung bietet. In Bezug auf die Entwicklung von RGB-Anhängern erweist sich der Einsatz des ESP01S aufgrund seiner kompakten Größe, der effektiven Steuerungsinteraktionsmethode und der geringen Lötbarkeit als besonders geeignet.

Beim Einsatz von WLAN zur Steuerung müssen auch einige Punkte beachtet werden. In einigen speziellen Fällen, zum Beispiel wenn mehrere Anhänger gleichzeitig verwendet werden, kann die Steuerung nur dann erfolgen, wenn der Benutzer korrekt mit dem WLAN des zu steuernden Anhängers verbunden ist. Dies beeinträchtigt jedoch die Nutzung im Allgemeinen nicht.

Und Als Schlussfolgerung wurde entschieden, das ESP01S Modul mit dem ESP8266EX als MCU als Controller für die WLAN-Lösung zu verwenden.

## 4. RGB-LED Auswahl

RGB-LED ist eine vielseitige Lichtquelle, die ihren Namen von den drei unterschiedlichen Farben der darin enthaltenen Leuchtdioden (LEDs) – Rot, Grün und Blau – erhält. Diese LEDs sind in einem einzigen Gehäuse zusammengefasst, das meist rund oder rechteckig ist. Die Hauptkomponenten dieser LED-Leuchte bestehen aus unabhängigen LED-Elementen in den drei Farben, wobei jede eine Primärfarbe repräsentiert. Das Funktionsprinzip beruht auf der Aktivierung und Mischung dieser farbigen LED-Elemente. Strömt Strom durch ein rotes LED-Element, emittiert es rotes Licht; analog dazu leuchtet ein grünes LED-Element grün, wenn es aktiviert wird. Wie Abbildung 4(a) zeigt, bilden Rot, Grün und Blau die additiven Primärfarben in der Farbmischung, im Gegensatz zu den subtraktiven Primärfarben Magenta, Gelb und Cyan, wie in Abbildung 4(b) dargestellt. Die Überlagerung von rotem, grünem und blauem Licht in gleicher Intensität erzeugt weißes Licht. Ohne Überlagerung dieser Lichter bleibt das RGB-LED-Licht aus, was bedeutet, dass kein Licht emittiert wird. Dieses Prinzip wird als additives Farbsystem bezeichnet und findet breite Anwendung in verschiedenen Produkten mit aktiv leuchtenden Displays.

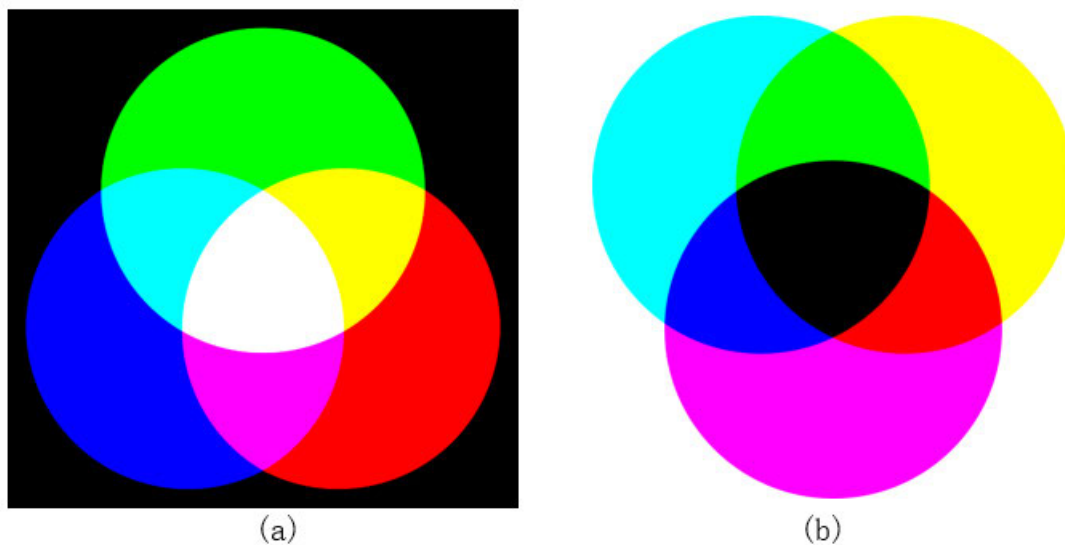


Abbildung 4: Farb-Overlay

Ziel dieses Kapitels ist es, mehrere RGB-LED-Leuchten zu kombinieren, um eine Vielzahl an steuerbaren Farbeffekten zu erzielen. Dazu betrachten wir zunächst die verschiedenen Steuerungsmethoden der derzeit auf dem Markt verfügbaren RGB-LED-Leuchten. Grundsätzlich lassen sich diese Leuchten in zwei Kategorien einteilen: intelligente RGB-LED-Leuchten und nicht-intelligente RGB-LED-Leuchten.

In diesem Kapitel erfolgt eine Analyse der Unterschiede zwischen diesen beiden Typen von RGB-LEDs, um eine fundierte Entscheidung für die Auswahl zu treffen. Die Analyse konzentriert sich darauf, die für die spezifischen Anforderungen am besten geeigneten Leuchten zu identifizieren, indem deren Funktionalitäten, Steuerungsoptionen sowie die damit verbundenen Vor- und Nachteile betrachtet werden.

### 4.1 Nicht-intelligent RGB-LED

#### 4.1.1 Hardware

Die Funktionsprinzipien der derzeit auf dem Markt erhältlichen nicht-intelligenten RGB-LED-Leuchten sind nahezu gleich. Am Beispiel des WP154A4SUREQBFZGW von Kingbright sind die Pins in Abbildung 5 dargestellt.:

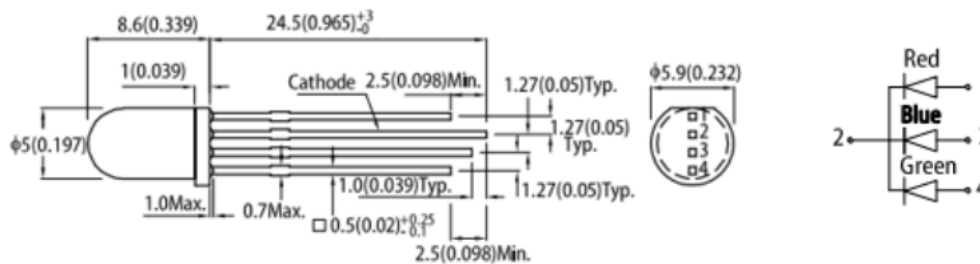


Abbildung 5: Pinbelegungsdiagramm 1[3]

Im Inneren des Gehäuses sind typischerweise drei separate Anodenstifte (positiv) und ein gemeinsamer Kathodenstift (negativ) angeordnet. Der Stromfluss erfolgt von einer der

**ELECTRICAL / OPTICAL CHARACTERISTICS at T<sub>A</sub>=25°C**

Parameter	Symbol	Emitting Color	Value		Unit
			Typ.	Max.	
Wavelength at Peak Emission I <sub>F</sub> = 20mA	λ <sub>peak</sub>	Hyper Red Blue Green	645 460 515	-	nm
Dominant Wavelength I <sub>F</sub> = 20mA	λ <sub>dom</sub> [1]	Hyper Red Blue Green	630 465 525	-	nm
Spectral Bandwidth at 50% Φ REL MAX I <sub>F</sub> = 20mA	Δλ	Hyper Red Blue Green	25 25 30	-	nm
Capacitance	C	Hyper Red Blue Green	45 100 45	-	pF
Forward Voltage I <sub>F</sub> = 20mA	V <sub>F</sub> [2]	Hyper Red Blue Green	1.9 3.3 3.3	2.5 4 4.1	V
Reverse Current (V <sub>R</sub> = 5V)	I <sub>R</sub>	Hyper Red Blue Green	-	10 50 50	μA
Temperature Coefficient of λ <sub>peak</sub> I <sub>F</sub> = 20mA, -10°C ≤ T ≤ 85°C	TC <sub>λpeak</sub>	Hyper Red Blue Green	0.14 0.04 0.05	-	nm/°C
Temperature Coefficient of λ <sub>dom</sub> I <sub>F</sub> = 20mA, -10°C ≤ T ≤ 85°C	TC <sub>λdom</sub>	Hyper Red Blue Green	0.05 0.03 0.03	-	nm/°C
Temperature Coefficient of V <sub>F</sub> I <sub>F</sub> = 20mA, -10°C ≤ T ≤ 85°C	TC <sub>V<sub>F</sub></sub>	Hyper Red Blue	-3 -3	-	mV/°C

Abbildung 6: WP154A4SUREQBFZGW Parameter[3]

Anoden zu der gemeinsamen Kathode, wodurch die jeweilige Farbe des Abschnitts aktiviert wird.

Die verschiedenen Parameter sind in Abbildung 6 dargestellt.

Im Datenblatt sind die Vorwärtsspannungsbereiche der drei Farbkanäle unterschiedlich, daher muss zur Ansteuerung vor jedem Kanal ein Vorwiderstand angeschlossen werden.

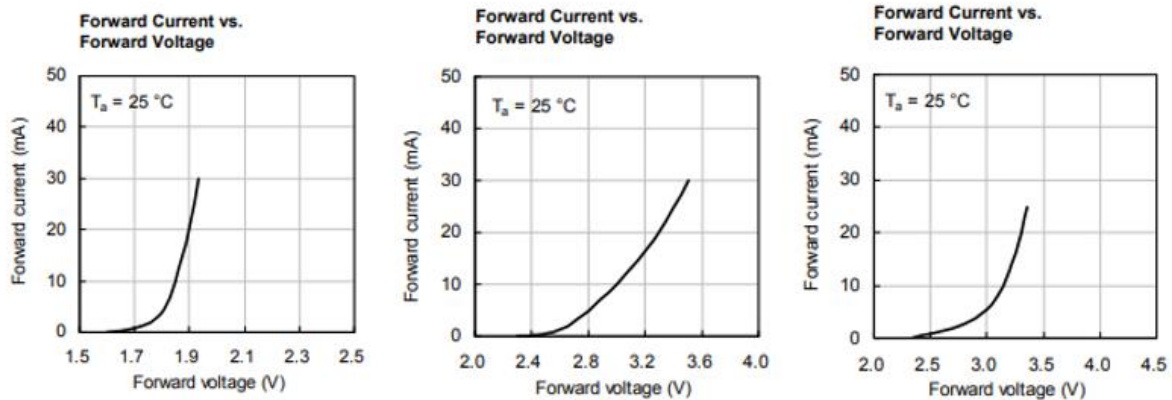


Abbildung 7: RGB Strom-Spannungs-Kennlinie[3]

Die Volt-Ampere-Kurve der drei RGB-Kanäle ist in Abbildung 7 dargestellt. Mit zunehmender Durchlassspannung steigt auch der Durchlassstrom jedes Farbkanals, wie in der Abbildung dargestellt. Im Allgemeinen beträgt der theoretische maximale Vorwärtsstrom eines einzelnen Kanals etwa 30 mA und der Gesamtstrom der drei Kanäle beträgt 90 mA. Wenn jedoch im tatsächlichen Betrieb die Durchlassspannung relativ niedrig ist, kann eine bessere Stromverbrauchsleistung erzielt werden (wenn beispielsweise die Spannung des roten Kanals 1,8 V und die Spannung des grünen und blauen Kanals 3,0 V beträgt).

#### 4.1.2 Kontrollweise

Bei nicht-intelligenten RGB-LED-Leuchten ist die Pulsweitenmodulation (PWM) die am häufigsten verwendete Steuerungsmethode. PWM ist eine Technik der elektronischen Steuerung, die den Ausgang einer Schaltung durch Anpassung des Tastverhältnisses des Signals modifiziert, also des Verhältnisses der Zeit, während der das Signal auf hohem Niveau ist, zum gesamten Zyklus. Bei der Steuerung von RGB-LEDs ermöglicht die PWM-Technik, durch Simulation einer sich kontinuierlich ändernden Spannung, die Helligkeit der LEDs zu variieren. Die zwei wesentlichen Parameter für die PWM-Steuerung bei RGB-LEDs sind die Frequenz (oder Periode) und das Tastverhältnis.

Periode und Frequenz: Die Periode eines PWM-Signals definiert die Zeitdauer einer vollständigen Wellenform, üblicherweise angegeben in Millisekunden (ms) oder

Mikrosekunden ( $\mu\text{s}$ ). Eine kürzere Periode führt zu einer häufigeren Wiederholung des PWM-Signals, wodurch sich bestimmt, wie schnell das Signal zyklisch wiederholt. Die Frequenz, gemessen in Hertz (Hz), gibt die Anzahl der vollständigen Wellenformen pro Sekunde an und beeinflusst die Reaktionsgeschwindigkeit des PWM-Signals. Höhere Frequenzen ermöglichen sanftere Übergänge in der Helligkeit. Die Beziehung zwischen Periode und Frequenz ist durch die Formel  $\text{Frequenz} = 1 / \text{Periode}$  gegeben.

**Tastverhältnis:** Das Tastverhältnis eines PWM-Signals beschreibt den Anteil der Zeit, in der das Signal in einem Zyklus auf hohem Niveau ist, im Verhältnis zur gesamten Zykluszeit, ausgedrückt in Prozent. Durch Anpassung des Tastverhältnisses lässt sich die Helligkeit der LEDs variieren. Bei einem Tastverhältnis von 50 % etwa wird die LED mit halber Helligkeit betrieben, was die Steuerung unterschiedlicher Helligkeitsstufen von RGB-LED-Leuchten ermöglicht. Mittels PWM wird die LED-Helligkeit durch Anpassen des Zeitanteils moduliert, in dem die LED innerhalb eines Zyklus aktiv leuchtet.

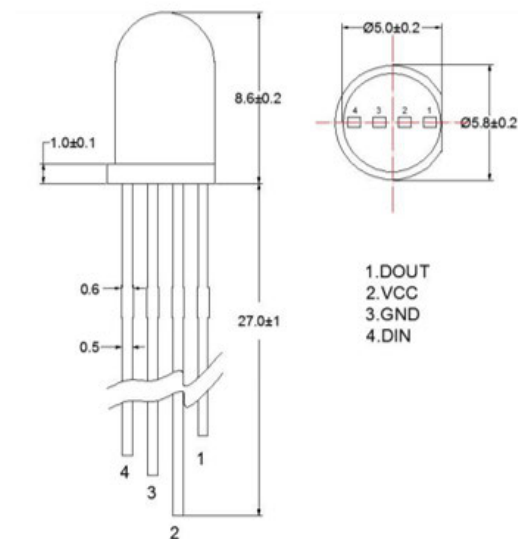
Die 8-Bit-PWM-Auflösung, die mit der Arduino IDE programmiert wird, repräsentiert das Tastverhältnis des PWM-Signals. Der maximale Wert in 8-Bit-Binärformat ist 255 (11111111), was 256 verschiedene Helligkeitsstufen ermöglicht. Um eine orangefarbene Anzeige zu erzielen, müssen die roten und grünen Kanäle gleichzeitig aktiviert werden. Das PWM-Tastverhältnis des roten Kanals wird auf 255 eingestellt, sodass er vollständig hell leuchtet, während das Tastverhältnis des grünen Kanals auf 128 eingestellt wird, was einer halben Helligkeit entspricht. Das Zusammenleuchten von Rot und Grün bei diesen Helligkeitsstufen erzeugt einen orangefarbenen Effekt. Für einen sanften Übergang von Orange zu Blau spielt die PWM-Frequenz eine wichtige Rolle. Beim Übergang zu Blau müssen die Tastverhältnisse der roten und grünen Pins auf 0 und des blauen Pins auf 255 eingestellt werden. Die Frequenz des PWM-Signals beeinflusst die Geschwindigkeit dieses Farbübergangs. Bei einer PWM-Frequenz von 1 kHz, mit einer Periode von 1 Millisekunde, wiederholt sich das Signal 1000 Male pro Sekunde. Unter der Annahme, dass das Tastverhältnis in jedem Zyklus um 1 reduziert wird, dauert der Übergang von Orange zu Blau 255 Millisekunden. Bei einer Frequenz von 100 Hz, mit einer Periode von 10 Millisekunden, dauert derselbe Übergang 2550 Millisekunden.

## 4.2 Intelligent RGB-LED



### 4.2.1 Hardware

Smart RGB LED, wie beispielsweise der WS2812D von WorldSemi, ist eine intelligente, extern gesteuerte LED-Lichtquelle, die sowohl einen Steuerschaltkreis als auch einen Lichtemissionsschaltkreis integriert. Dieses Modell verwendet ein vierpoliges Plug-in-Gehäuse, wobei Pin 1 für Dout, Pin 2 für VCC, Pin 3 für GND und Pin 4 für DIN vorgesehen ist. Das externe Steuersignal wird bei Pin 4 (DIN) eingegeben und über Pin 1 (Dout) ausgegeben. Die Arbeitsspannung dieser LED liegt im Bereich von 3,7 V bis 5,3 V. Das Pin-Diagramm wird in Abbildung 8 dargestellt:



#### PIN Function

NO.	Symbol	PIN	Function description
1	Dout	DATA OUT	Control data signal output
2	VCC	POWER SUPPLY	Power supply
3	GND	GROUND	Ground,data & power grounding
4	Din	DATA IN	Control data signal input

Abbildung 8:Pinbelegungsdiagramm 2[4]

Der WS2812D enthält eine intelligente digitale Schnittstellenschaltung, die Daten-Latch, Signalformung, Verstärkung und Treiberfunktionen integriert. Zusätzlich verfügt er über einen hochpräzisen internen Oszillator und einen Konstantstrom-Steuerabschnitt, um eine gleichbleibende Farbqualität der LED-Lichter zu gewährleisten. In der Praxis ermöglicht diese Konfiguration die sequenzielle Durchführung mehrerer Funktionen, um komplexe Lichteffekte zu erzielen.

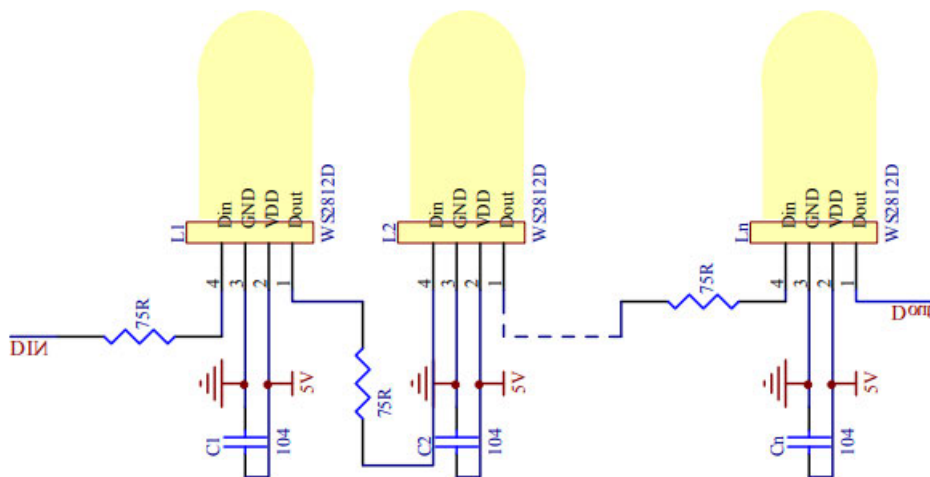
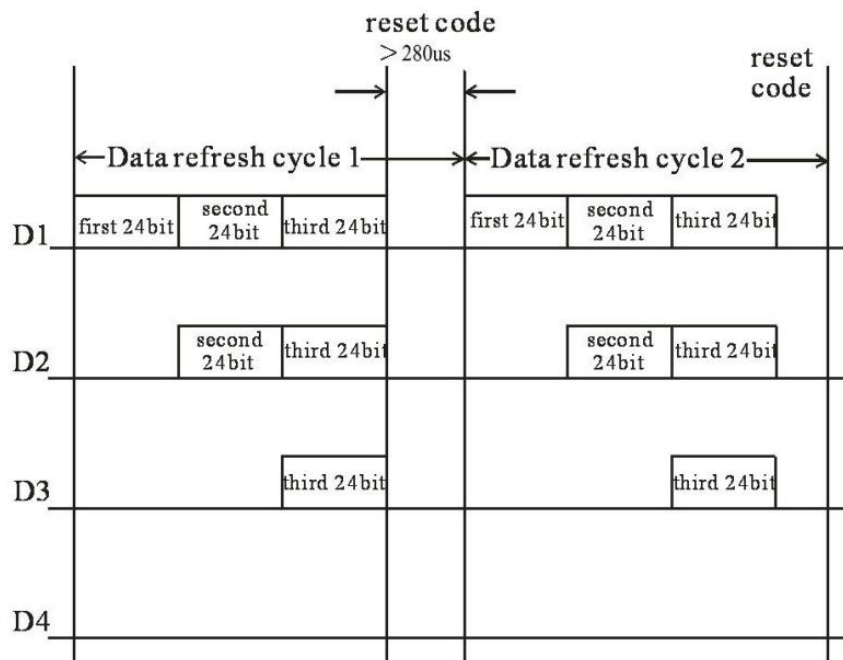


Abbildung 9:Verbindungsweise[4]

Data transmission method:



Note: The data of D1 is send by MCU, and D2, D3, D4 through pixel internal reshaping amplification to transmit.

Abbildung 10: Datenübertragungsweise[4]

Die Schaltungsverbindungsmethode und die Datenübertragungsmethode sind in Abbildung 9 und 10 dargestellt. Sobald die LED eingeschaltet wird, beginnt das DIN-Terminal, die von der Mikrocontroller-Einheit (MCU) gesendeten Daten zu akzeptieren. Die ersten 24-Bit-Daten werden von der ersten LED aufgenommen und in ihren internen Speicher übertragen, während die nachfolgenden Daten durch die interne Signalformungs- und Verstärkungsschaltung weitergeleitet und dann über den DO-Port an die nächste LED in der Kaskade ausgegeben werden. Mit jeder Durchleitung durch eine LED verringert sich die Datenmenge um 24 Bit. Um die Auswirkungen von Spannungsschwankungen der Stromversorgung zu minimieren, kann parallel zur LED ein 100-nF-Filterkondensator hinzugefügt werden. Für die Ansteuerung eines einzelnen WS2812 wird DIN mit dem PWM-Signal, das vom Controller erzeugt wird, verbunden, während Dout entweder mit dem DIN der nächsten LED oder, falls es die letzte LED in der Reihe ist, unbeschaltet bleibt.

**Electrical Characteristics**( $T_A=-20\sim+70^{\circ}\text{C}$ ,  $V_{CC}=5\text{V}$ ,  $V_{SS}=0\text{V}$ , unless otherwise specified)

Parameter	Symbol	Min	Tpy	Max	Unit	Conditions
RGB Current	$I_{out}$	32	36	40	mA	R+G+B
Input current	$I_I$	—	—	$\pm 1$	$\mu\text{A}$	$V_I=V_{CC}/V_{SS}$
High Voltage Input	$V_{IH}$	2.7V	—	$V_{CC}+0.7\text{V}$	V	$D_{IN}$
Low Voltage Input	$V_{IL}$	-0.3V	—	0.7V	V	$D_{IN}$

Abbildung 11: Elektrische Parameter[4]

Unter 5V-Betriebsbedingungen sind seine Parameter in Abbildung 11 dargestellt. Zu diesem Zeitpunkt beträgt der Betriebsstrom  $I_{out}$  eines einzelnen WS2812D 32 bis 40 mA, aber wenn die Versorgungsspannung abnimmt, nimmt auch  $I_{out}$  entsprechend ab.

In der tatsächlichen Entwicklung ist ein anderes WS2812D-Modell mit niedrigerer Betriebsspannung die bessere Wahl (dieses Modell ist jedoch schwieriger zu kaufen) und seine Betriebsspannung beträgt 3 V bis 5,5 V. Nach der eigentlichen Messung kann es nicht normal verwendet werden (das blaue Licht kann nicht leuchten), wenn die Spannung niedriger als 2,65 V ist. Gleichzeitig ist die minimale Antriebsspannung der drei RGB-Farben in Tabelle 2 aufgeführt.

	Rot	Grün	blau	Einheit
Minimale Betriebsspannung	1.95	2.4	2.65	V

Tabelle 2 Minimale Betriebsspannung

Nach der eigentlichen Messung beträgt der maximale Betriebsstrom bei einer Spannung von 3,3 V 10 mA (d. h. maximale Helligkeit unter Weißlichtbedingungen).

#### 4.2.2 Kontrollweise

Das Datenblatt gibt an, dass die Übertragungsgeschwindigkeit 800 Kbps beträgt, woraus sich eine PWM-Frequenz von 800 KHz ergibt. Dies entspricht einem Zyklus von  $1/800.000$  Sekunden, also 0,00000125 Sekunden oder 1,25 Mikrosekunden (us). Innerhalb dieses 1,25-us-Zyklus kann ein High- oder Low-Signal dargestellt werden, und mit 24 solcher Zyklen kann die Farbe einer LED dargestellt werden. Um die Farbdaten von dem Latch in die LED zu übertragen, muss nach dem Senden der Farbdaten einer LED ein Low-Signal von mehr als 280 Mikrosekunden gesendet werden.

##### Composition of 24bit Data

R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Note: Data transmit in order of RGB, high bit data at first.

Abbildung 12: Daten Zusammensetzung[4]

Die Farbe jeder LED muss durch eine Zahl einer bestimmten Größe dargestellt werden. Die Struktur dieser Zahl ist in Abbildung 12 dargestellt, wobei R7 bis R0 die rote Komponente darstellen, G7 bis G0 die grüne Komponente und B7 bis B0 repräsentiert die blaue Komponente. Jede Komponente belegt 8 Bit, sodass zur Darstellung einer Farbe insgesamt 24 Bit erforderlich sind (d. h. für die Programmierung ist mindestens eine 32-Bit-Zahl erforderlich).

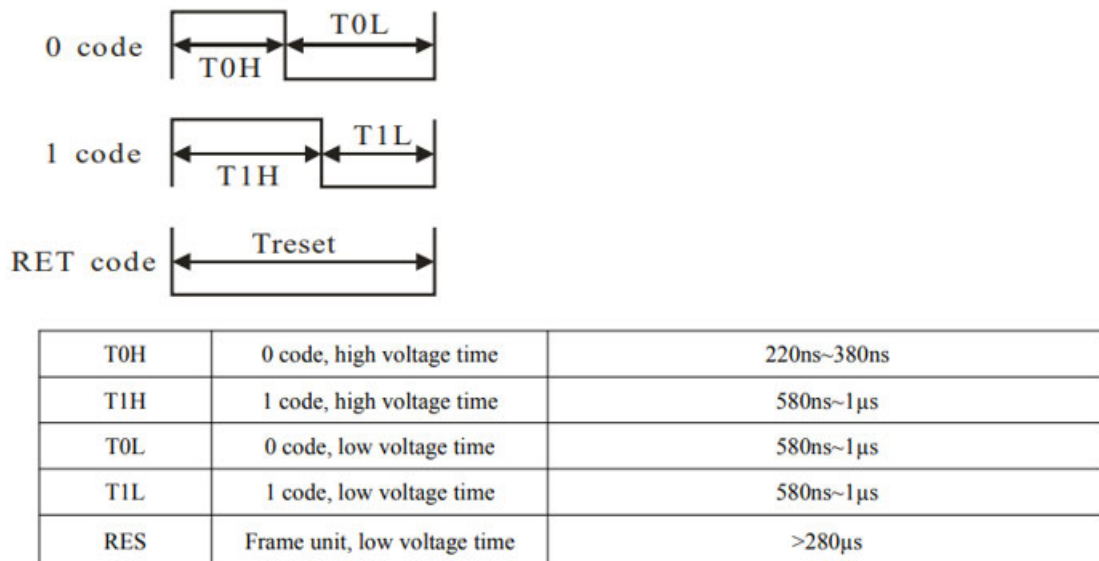


Abbildung 13:Sequenzdiagramm und Datenübertragungszeit[4]

Darunter sind das Zeitdiagramm und die Datenübertragungszeit in der Abbildung 13 dargestellt. In der tatsächlichen Verwendung kann es wie folgt vereinfacht werden:

- Logik-Low-Pegel: 0,3 µs High-Pegel + 0,6 µs Low-Pegel;
- Logik-High-Pegel: 0,6 µs High-Pegel + 0,6 µs Low-Pegel.

Durch Anpassen des E/A-Ports an die entsprechende Zeitdauer nach oben oder unten lassen sich unterschiedliche „0“- und „1“-Signale erzeugen.

Daher können beim Programmieren mit Arduino diese Konstanten vorab gemäß den Kommunikationsanforderungen von WS2812 konfiguriert werden:

```
// Bei der Programmierung mit der Arduino IDE können diese Konstanten entsprechend den
// Kommunikationsanforderungen des WS2812 vorab konfiguriert werden:
#define DATA_PIN // Definiert den Datenpin für WS2812
#define NUM_LEDS // Definiert die Anzahl der LEDs

// Definiert die Zeitparameter für die Datenübertragung des WS2812
#define T_HIGH 600 // Dauer des High-Signals (Mikrosekunden)
#define T_LOW_0 300 // Dauer des Low-Signals für die Darstellung von 0 (Mikrosekunden)
#define T_LOW_1 600 // Dauer des Low-Signals für die Darstellung von 1 (Mikrosekunden)

// Definiert ein Array der Größe NUM_LEDS, um die Farbdaten jeder LED zu speichern
uint32_t leds[NUM_LEDS];

// uint32_t ist ein Datentyp, der in der C (und C++) Standardbibliothek definiert ist und einen
// 32-Bit-Unsigned-Integer repräsentiert.
// Zum Beispiel, in der praktischen Anwendung:
// Rot wird als 0xFF0000 dargestellt (FF steht für die maximale Farbintensität, d.h. 255,
// wobei Grün und Blau 0 sind).
// Grün wird als 0x00FF00 dargestellt.
// Blau wird als 0x0000FF dargestellt.
```

// Die Modularisierung des Kommunikationsprozesses ist wie folgt:

// Der nächste Schritt ist der eigentliche Farbkonfigurationsprozess, beginnend mit zwei Funktionen für den Kommunikationsteil:

// Sendet Daten für High-Signal

```
void sendDataHigh() {
    digitalWrite(DATA_PIN, HIGH);
    delayMicroseconds(T_HIGH);
    digitalWrite(DATA_PIN, LOW);
    delayMicroseconds(T_LOW_1);
}
```

// Sendet Daten für Low-Signal

```
void sendDataLow() {
    digitalWrite(DATA_PIN, HIGH);
    delayMicroseconds(T_LOW_0);
    digitalWrite(DATA_PIN, LOW);
    delayMicroseconds(T_HIGH);
}
```

// Hierbei steuern die Funktionen sendDataHigh() und sendDataLow() die Übertragung von High- und Low-Signalen gemäß dem Kommunikationsprotokoll von WS2812.

// Die Funktionen digitalWrite() und delayMicroseconds() werden verwendet, um den Zustand des Datenpins und die Dauer zu kontrollieren.

// Der nächste Schritt ist das Senden der RGB-Daten an WS2812:

```
void sendRGBData(uint32_t *data, int numLEDs) {
    for (int i = 0; i < numLEDs; i++) {
        uint32_t color = data[i];

        // Sendet 24-Bit-Farbdaten
        for (int j = 23; j >= 0; j--) {
            if (color & (1UL << j)) {
                sendDataHigh();
            } else {
                sendDataLow();
            }
        }
    }
}
```

// Hierbei durchläuft die Schleife for (int j = 23; j >= 0; j--) jedes Bit der Variable color, von der höchsten (23.) bis zur niedrigsten (0.) Stelle.

// 1UL << j erzeugt eine Zahl, bei der nur das j-te Bit gesetzt ist.

// Das bitweise UND (&) prüft, ob das entsprechende Bit in color gesetzt ist.

```
// Abschließend wird als Beispiel das Einstellen eines Regenbogeneffekts gezeigt:
// Definition der Regenbogenfarben
uint32_t rainbowColors[] = {0xFF0000, 0xFFFF00, 0x00FF00, 0x00FFFF, 0x0000FF};
int numRainbowColors = 5;

// LED-Farbdaten
uint32_t leds[NUM_LEDS];

void setup() {
  // Konfiguriert DATA_PIN als Ausgang
  pinMode(DATA_PIN, OUTPUT);
  // Initialisiert LEDs auf aus
  for (int i = 0; i < NUM_LEDS; i++) {
    leds[i] = 0x000000; // Schaltet alle LEDs aus
  }
}

void loop() {
  for (int i = 0; i < numRainbowColors; i++) {
    for (int j = 0; j < NUM_LEDS; j++) {
      leds[j] = rainbowColors[(i + j) % numRainbowColors];

      // Ruft die Funktion sendRGBData() auf, um die Farbdaten an WS2812 zu senden
      sendRGBData(leds, NUM_LEDS);

      // Verzögerung für den Farbflusseffekt
      delay(100);
    }
  }
}
```

Im Entwicklungsumfeld der Arduino IDE sind `setup()` und `loop()` zwei grundlegende und zentrale Funktionen, die das grundlegende Gerüst eines Arduino-Programms bilden. Die `setup()`-Funktion wird einmal zu Beginn des Programms ausgeführt und dient zur Initialisierung. Die `loop()`-Funktion wird nach `setup()` kontinuierlich wiederholt ausgeführt. `pinMode()` ist eine eingebaute Funktion der Arduino-Sprache, die verwendet wird, um digitale Pins als Eingang (INPUT), Ausgang (OUTPUT) oder in einem anderen speziellen Modus (wie INPUT\_PULLUP) zu konfigurieren. Hier werden die Pins als Ausgang konfiguriert, um hohe oder niedrige Pegel an die externe Schaltung zu liefern. Die äußere `for`-Schleife durchläuft jede Farbe im Regenbogenfarben-Array. Die innere `for`-Schleife durchläuft jedes LED und setzt es auf die aktuelle Regenbogenfarbe. Durch die Verwendung von  $(i + j) \% \text{numRainbowColors}$  in der Modulo-Operation wird sichergestellt, dass der berechnete Index auch dann innerhalb des gültigen Bereichs des Arrays „kreist“, wenn der Wert von  $i + j$  die Länge des Arrays übersteigt. Da die `loop()`-Funktion kontinuierlich

wiederholt wird, beginnt  $i$  nach dem Ende der Schleife wieder bei 0, und die innere  $j$ -Schleife beginnt ebenfalls von vorne.

Diese Methode bietet volle Kontrolle über die Hardware und ermöglicht es Entwicklern, feine Anpassungen für spezifische Anwendungen vorzunehmen. Das Schreiben und Debuggen von Code auf niedriger Ebene ist jedoch komplex und direkte Steuerung von Timing und Datentransfer ist fehleranfälliger, was bedeutet, dass mehr Zeit für das Debugging benötigt wird. Insbesondere unter den Entwicklungsanforderungen dieser Arbeit, wenn Dutzende verschiedener Farbmodi programmiert werden sollen, würde dies erhebliche Zeitkosten verursachen. Glücklicherweise gibt es viele Drittanbieterbibliotheken, die den Prozess der Konfiguration von WS2812-Farbmodi erheblich vereinfachen können. Daher wird in Abschnitt 4.2.3 diese Methode vorgestellt und verwendet, um die Farbmodi von WS2812 zu konfigurieren.

### 4.3 Vergleich der Vor- und Nachteile zweier RGB-LEDs

Für die Entwicklungsanforderungen niedriger Kosten und kleiner Größe ist die Verwendung von nicht-intelligenten RGB LEDs keine ideale Wahl. Obwohl nicht-intelligente RGB LEDs möglicherweise einen geringfügig niedrigeren Energieverbrauch haben, ist dies in der Entwicklung kein signifikanter Vorteil.

Im Gegenteil, da die Steuerung einer einzelnen nicht-intelligenten RGB LED 3 GPIO-Ports erfordert, wird eine Bussteuerungsmethode sehr notwendig, wenn das Design viele RGB LEDs benötigt. Wenn man die Steuerung über das SPI-Protokoll in Betracht zieht, muss ein PWM-Erweiterungschip, wie der PCA9685, hinzugefügt werden. Darüber hinaus führt das Hinzufügen eines zusätzlichen MCUs zu einer Erhöhung der Entwicklungskosten und zu einer komplexeren Verdrahtung, was für Projekte, die niedrige Kosten und ein vereinfachtes Design anstreben, nicht vorteilhaft ist.

Daher haben intelligente RGB LEDs in solchen Projekten mit kleinen, kostengünstigen Anhängern deutliche Vorteile.



## 5. ESP8266EX und CODE Entwurf

Dieses Kapitel führt eine Analyse der Funktionskomponenten des verwendeten Mikrocontrollers durch. Ziel ist es, die Funktionserfordernisse zu erfüllen und den Energieverbrauch zu reduzieren. Auf dieser Grundlage wird die Code-Entwicklung durchgeführt.

### 5.1 ESP8266EX

#### 5.1.1 Innere Struktur

ESP8266EX ist ein hochintegrierter Wi-Fi-Chip, der häufig in IoT-Anwendungen eingesetzt wird. Der von Espressif Systems entwickelte Chip ist wegen seiner Kosteneffizienz und Energieeffizienz beliebt. Seine interne Zusammensetzung ist in Abbildung 14 dargestellt.

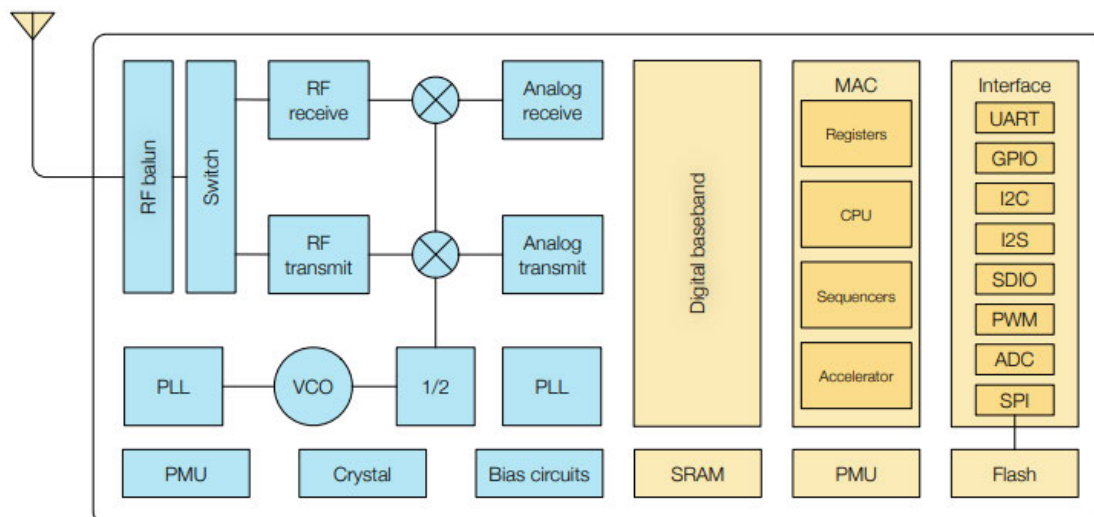


Abbildung 14:Interne Struktur von ESP8266EX[5]

ESP8266EX kann einfach als aus den folgenden Modulen bestehend betrachtet werden:

- Wi-Fi-Modul: einschließlich HF-Frontend, Schalter, HF-Empfang/Sende, PLL, VCO, Frequenzteiler (1/2), Quarzoszillator (Crystal) und Vorspannungsschaltungen (Bias-Schaltungen). Dieses Modul verwaltet die physikalischen Aspekte drahtloser Signale, einschließlich der Erzeugung und des Empfangs von HF-Signalen.
- Digitales Basisband-Verarbeitungsmodul: Enthält eine digitale Basisband-Verarbeitungseinheit und den zugehörigen SRAM. Dieser Teil übernimmt die Modulation und Demodulation digitaler Signale sowie andere digitale Signalverarbeitungsfunktionen.
- Verarbeitungs- und Steuermodule: einschließlich CPU, MAC, Register, Sequenzer (Sequencer) und Beschleuniger (Accelerator). Dieses Modul ist für die Ausführung von Programmcode, die Datenverarbeitung, die Verwaltung der Netzwerkkommunikation und andere Steuerfunktionen verantwortlich.

- Energieverwaltungsmodul: einschließlich PMU. Dieser Teil ist für die Stromverteilung, die Batterieladesteuerung und das Stromverbrauchsmanagement verantwortlich.
- Speichermodul: inklusive Flash-Speicher. Wird zum Speichern von Firmware, Programmcode und anderen Langzeitdaten verwendet.
- Schnittstellenmodul: einschließlich GPIO, PWM, SPI usw. bei Interface. Dieses Modul stellt Kommunikationsschnittstellen zu externen Geräten und anderen elektronischen Komponenten bereit.

Der ESP8266EX verfügt über einen integrierten 32-Bit-RISC-Prozessor Tensilica L106 mit extrem geringem Stromverbrauch. Die Standard-CPU-Taktfrequenz beträgt 80 MHz. Wenn Leistung erforderlich ist, kann sie über SDK und andere Konfigurationsmethoden auf 160 MHz erhöht werden.

Nachdem die interne Struktur des ESP8266EX bestimmt wurde, kann durch das möglichst Abschalten überflüssiger Module unter der Voraussetzung der Erfüllung der Funktionsanforderungen das Energiesparen erreicht werden.

### 5.1.2 Der Energieverbrauch bei verschiedenen Arbeitsweisen

Bezüglich des verwendeten WLAN-Standards unterstützt ESP8266EX das Protokoll IEEE 802.11 b/g/n und kann im 2,4-GHz-Frequenzband arbeiten. In Bezug auf die HF-Leistung bietet es eine Ausgangsleistung von maximal +20 dBm bei guter Empfangsempfindlichkeit, wodurch das Modul in verschiedenen Umgebungen eine stabile drahtlose Verbindung aufrechterhalten kann. Bezüglich des Stromverbrauchsteils, der in dieser Arbeit am meisten im Fokus steht, ist der Stromverbrauch für jeden Parameter des ESP8266EX in Abbildung 15 dargestellt.

Parameters	Min	Typical	Max	Unit
TX802.11 b, CCK 11 Mbps, P <sub>OUT</sub> = +17 dBm	-	170	-	mA
TX802.11 g, OFDM 54Mbps, P <sub>OUT</sub> = +15 dBm	-	140	-	mA
TX802.11 n, MCS7, P <sub>OUT</sub> = +13 dBm	-	120	-	mA
Rx802.11 b, 1024 bytes packet length, -80 dBm	-	50	-	mA
Rx802.11 g, 1024 bytes packet length, -70 dBm	-	56	-	mA
Rx802.11 n, 1024 bytes packet length, -65 dBm	-	56	-	mA

Abbildung 15: Stromverbrauch von HF-Modul[5]

In dieser Tabelle ist der typische Stromverbrauch des WLAN-Chips ESP8266EX unter verschiedenen drahtlosen Kommunikationsstandards und unter verschiedenen Bedingungen aufgeführt. Der typische Stromverbrauch des Chips variiert in verschiedenen Betriebsmodi erheblich. Im Allgemeinen arbeitet der ESP8266EX im TX-Modus, wenn er Daten an andere Geräte oder Router im Netzwerk übertragen muss. Wenn der ESP8266EX auf Daten vom Netzwerk wartet oder diese empfängt, arbeitet er im RX-Modus. Von dieser Tabelle aus ist ersichtlich, dass die Einstellung des Betriebsmodus auf RX-Modus, wodurch der ESP8266EX zu einem Datenempfänger wird, eine gute Möglichkeit zur Energieeinsparung darstellt. Gleichzeitig verfügt der ESP8266EX über ein integriertes Low-Power-

Management mehrerer Schlafmodi, das durch das Herunterfahren verschiedener Module entsprechend unterschiedlicher Arbeitsanforderungen erreicht wird.

Die drei integrierten spezifischen Schlafmodi sind wie folgt:

- Modem-Schlafmodi: CPU ist betriebsbereit. WLAN und Radio sind ausgeschaltet. Der durchschnittliche Gesamtstrom beträgt etwa 20 mA.
- Leichtschlafmodi: Die CPU und alle Peripheriegeräte werden angehalten. Jedes Weckereignis (MAC, Host, RTC-Timer oder externer Interrupt) weckt den Chip. Der durchschnittliche Gesamtstrom beträgt etwa 2 mA.
- Tiefschlafmodi: Nur die RTC befindet sich im Betriebszustand und andere Teile des Chips sind ausgeschaltet. Der durchschnittliche Gesamtstrom kann deutlich weniger als 1 mA betragen. Der gemessene Stromwert beträgt 20µA bei 2,5V.

Am bemerkenswertesten ist der Modem-Sleep-Modus. In diesem Modus wird das WLAN-Modul nicht vollständig ausgeschaltet, sondern zeitweise eingeschaltet. Er wird für Anwendungen verwendet, bei denen die CPU ständig arbeiten muss, z. B. bei PWM- oder I2S-Anwendungen. Wenn bei der Aufrechterhaltung einer Wi-Fi-Verbindung keine Datenübertragung stattfindet, kann der Wi-Fi-Modem-Schaltkreis ausgeschaltet werden, um gemäß dem 802.11-Standard Strom zu sparen. Aufgrund des übermäßigen Stromverbrauchs des WLAN-Moduls sollten Funktionen wie der Modem-Schlafmodus, der das WLAN-Modul ausschalten kann, bei Designüberlegungen Vorrang haben.

### 5.1.3 Arbeitsmodi

Je nachdem, wie WLAN genutzt wird, verfügt der ESP8266EX über drei voreingestellte Arbeitsmodi: STA-Modus (Station), SoftAP-Modus (Soft Access Point) und STA+SoftAP-Hybridmodus.

- STA-Modus: In diesem Modus fungiert der ESP8266EX als WLAN-Client. Er kann eine Verbindung zu vorhandenen Wi-Fi-Netzwerken herstellen und mit diesen interagieren. Dies ist der am häufigsten verwendete Modus zum Verbinden Ihres Geräts mit dem Internet oder einem lokalen Netzwerk. Wenn innerhalb eines bestimmten Zeitraums keine Datenübertragung erfolgt, wechselt der ESP8266EX im STA-Modus automatisch in den Modem-Schlafmodus.
- SoftAP-Modus: In diesem Modus fungiert der ESP8266EX selbst als WLAN-Hotspot. Andere Wi-Fi-Geräte können sich mit diesem vom ESP8266EX erstellten Netzwerk verbinden. Dieser Modus wird häufig für die direkte Kommunikation zwischen Geräten verwendet, beispielsweise in Umgebungen, in denen kein Wi-Fi-Netzwerk vorhanden ist.
- STA+SoftAP-Hybridmodus: Der ESP8266EX kann auch gleichzeitig im Stations- und SoftAP-Modus ausgeführt werden. Dies bedeutet, dass er sowohl eine Verbindung zu einem Wi-Fi-Netzwerk herstellen als auch ein eigenes Wi-Fi-Netzwerk erstellen kann, mit dem andere Geräte eine Verbindung herstellen können.

Derzeit scheint der STA-Modus der ideale Arbeitsmodus für den RGB-Anhänger zu sein, da der RGB-Anhänger lediglich Daten akzeptieren muss und der ESP8266EX danach auch automatisch in den Schlafmodus wechseln kann, um Strom zu sparen. Die Nutzung des STA-Modus bedeutet jedoch, dass eine Netzwerkverbindung vor der Verwendung des RGB-Anhängers hergestellt werden muss. Ein weiteres Problem ist die Notwendigkeit, das

Netzwerk vorab zu konfigurieren, mit dem eine Verbindung hergestellt werden soll. Dies schränkt die tatsächliche Kontrolle des Benutzers über die Konfiguration durch den Code ein und beeinträchtigt die Intuitivität.

Im SoftAP-Modus kann der ESP8266EX jedoch direkt an das LAN senden, und andere Wi-Fi-Geräte können sich mit diesem Netzwerk verbinden und es dann direkt steuern, indem sie auf das Standard-Gateway des Netzwerks zugreifen. Anschließend kann eine einfache Webseite auf der Adresse des Standard-Gateways entworfen werden, über die dann Daten an den ESP8266EX übertragen werden. Diese Steuerungsmethode ist intuitiv und einfach und kann die Steuerungsanforderungen dieser Arbeit erfüllen.

## 5.2 Code Entwurf

Basierend auf der obigen Diskussion wird eine Steuerungsmethode basierend auf dem SoftAP-Arbeitsmodus zusammengefasst. Als Nächstes werden in dieser Arbeit die grundlegenden Komponenten des Codes auf dieser Basis diskutiert. Da die Arduino IDE die direkte Entwicklung für ESP8266 unterstützt und im Vergleich zum offiziellen SDK den Zugriff auf eine Vielzahl von Open-Source-Bibliotheken direkt in der Arduino IDE ermöglicht, wird diese Entwicklungsumgebung bevorzugt. Darüber hinaus ist das Brennen der Firmware einfacher und schneller, was die Präferenz für diese Umgebung weiter verstärkt.

### 5.2.1 Webserver einrichten

Für einen neuen ESP8266EX-Mikrocontroller lautet die Standard-Gateway-Adresse 192.168.4.1. Nachdem das Smartgerät mit dem WLAN verbunden ist, kann der Benutzer über dieses Gateway drahtlos mit der MCU kommunizieren. Zuvor muss der Webserver konfiguriert werden.

- Einrichtung des Web-Servers: Es muss ein ESP8266WebServer-Objekt erstellt und der zu überwachende Port angegeben werden (normalerweise 80). Anschließend müssen verschiedene Routen und Verarbeitungsfunktionen festgelegt werden, die definieren, wie der Server auf bestimmte HTTP-Anfragen reagiert.
- Definition der Anfrageverarbeiter: Für jede Route (URL-Pfad) auf dem Server muss eine Funktion definiert werden, um Anfragen zu verarbeiten und Antworten zu senden. Hierbei handelt es sich oft um die Verwendung von HTML-Seiten.
- Beginn der Anfrageüberwachung: Mit `server.begin()` wird der Server gestartet, um anschließend Anfragen von Clients zu überwachen.
- Hauptschleife: In der `loop()` -Funktion wird `server.handleClient()` kontinuierlich aufgerufen, um eingehende HTTP-Anfragen zu verarbeiten.

Um die Konfiguration des Webserver abzuschließen, müssen folgende Bibliotheken eingebunden werden:

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
```

Für diese beiden Bibliotheken lautet der Code für diesen Teil wie folgt:

```
// Wird verwendet, um eine Instanz der ESP8266WebServer-Klasse zu erstellen, die Port 80
überwacht – den Standardport für HTTP.
```

```
ESP8266WebServer server(80);

void setup() {
  // SoftAP-Modus Konfiguration
  WiFi.softAP(ssid, password);

  // Definiert eine Route. Bei Anforderung des Root-Pfades ("/") wird die Funktion
  handleRoot aufgerufen.
  server.on("/", handleRoot);

  // Startet den Server, um auf eingehende Anforderungen am Port zu lauschen.
  server.begin();
}

void loop() {
  // Kontinuierliche Überprüfung und Bearbeitung eingehender HTTP-Anforderungen.
  server.handleClient();
}

// Reagiert auf Anfragen an den Root-Pfad. Sendet eine HTTP-Antwort mit dem Statuscode
200, Inhaltstyp "text/html" und einer Willkommensnachricht.
void handleRoot() {
  server.send(200, "text/html", "Welcome to ESP8266 Web Server");
}
```

### 5.2.2 HTML Entwurf und Benutzeroberfläche

Als nächster Schritt ist die Entwicklung einer Benutzeroberfläche erforderlich, also des Designs des HTML-Inhalts, durch den Benutzer direkt mit dem ESP8266EX interagieren können. Trotz des breiten Gestaltungsspielraums für das Layout der Seite – beispielsweise durch Anpassung von CSS-Stilen – liegt der Schwerpunkt dieses Abschnitts auf der Implementierung bestimmter grundlegender und essenzieller Elemente. Nach Fertigstellung können diese Elemente als String direkt in den Programmcode integriert und nach Bedarf abgerufen werden. Die erforderlichen Schritte umfassen:

- HTML-Erstellung: Es wird ein HTML-String entwickelt, der grundlegende HTML-Elemente wie Überschriften, Formulareingaben und Schaltflächen umfasst. Dies beinhaltet auch verschiedene Formularelemente für Benutzereingaben, wie Textfelder, Dropdown-Menüs und Schieberegler. Jede Benutzerinteraktion, wie das Klicken auf eine Schaltfläche, erfordert einen dedizierten Anforderungspfad.
- HTML-Einbettung: Der HTML-Code wird als Zeichenkette im Arduino-Code gespeichert, wobei der Syntax `R"( ... )"` verwendet wird. Dies ermöglicht die Einbettung von mehrzeiligem HTML-Code auf eine Weise, die mehrere Zeilen und Sonderzeichen ohne die Notwendigkeit von Escape-Zeichen unterstützt. Diese Methode erleichtert das Schreiben von Strings mit komplexem Format, wie beispielsweise HTML-Code.

- Verarbeitung von GET- und POST-Anfragen: Im Servercode des ESP8266EX werden spezifische Routen und Funktionen implementiert, um GET- und POST-Anfragen zu bearbeiten. Dies ist entscheidend für die Verarbeitung von Benutzerbefehlen.
- Antwort des HTML-Dokuments: Die `server.send()`-Methode wird in der Verarbeitungsfunktion verwendet, um das HTML-Dokument als HTTP-Antwort zu senden.
- Analyse der Anforderungsdaten: In der Verarbeitungsfunktion für POST-Anfragen werden basierend auf den vom Benutzer übermittelten Daten entsprechende Aktionen ausgeführt.

Der endgültige Code sollte die folgenden Teile enthalten:

```
// Definiert den HTML-Inhalt für die Startseite
const char index_html[] = R"rawliteral(
<!DOCTYPE HTML>
<html>
<head>
  <title>ESP8266 Steuerung</title>
</head>
<body>
  <h1>ESP8266 Steuerungsinterface</h1>
  <form action="/action">
    <input type="text" name="input" placeholder="Befehl eingeben">
    <input type="submit" value="Absenden">
  </form>
</body>
</html>
)rawliteral";

// Verarbeitet Anfragen an den Wurzelpfad und gibt das Steuerungsinterface zurück
void handleRoot() {
  server.send(200, "text/html", index_html);
}

// Verarbeitet die Aktion, empfängt Eingaben und gibt Ergebnisse zurück
void handleAction() {
  String input = server.arg("input"); // Eingabewert abrufen
  // Führt Aktionen basierend auf dem Eingabewert aus

  // Gibt das Ergebnis der Aktion zurück
  server.send(200, "text/html", "Empfangen: " + input + "<br><a href='\"/\"'>Zurück</a>");
}

// Richtet den Server und die Routen ein
void setup() {
```

```
// Definiert Routen und zugehörige Verarbeitungsfunktionen
server.on("/", handleRoot);
server.on("/action", handleAction);

// Startet den Server
server.begin();
}

// Hauptloop, bearbeitet Client-Anfragen
void loop() {
  server.handleClient();
}
```

### 5.2.3 Licht Setup

Um die Ausdruckskraft von Lichteffekten zu verbessern, können für Einstellungen die Open-Source-Funktionsbibliothek FastLED genutzt werden. FastLED ist ein Open-Source-Projekt auf GitHub, das eine breite Palette von LED-Hardware unterstützt, darunter WS2811, WS2812, NeoPixel und APA102. Diese Bibliothek bietet umfangreiche Farbsteuerungsfunktionen, einschließlich der RGB- und HSV-Farbbräume.

Die in diesem Entwurf verwendeten Befehle lauten wie folgt:

```
FastLED.addLeds<LED_TYPE, DATA_PIN, COLOR_ORDER>(LEDS, NUM_LEDS);
// Initialisiert den LED-Streifen. LED_TYPE bezeichnet den Typ der LEDs, z.B. WS2812;
DATA_PIN ist der Datenpin für die LEDs; COLOR_ORDER gibt die Farbreihenfolge an
(z.B. GRB); LEDS ist das Array, das den Zustand der LEDs speichert; NUM_LEDS ist die
Anzahl der LEDs.
```

```
FastLED.setBrightness(brightness);
// Stellt die globale Helligkeit ein. Die Helligkeit ist ein Byte groß.
```

```
leds[i] = CRGB::Color;
// Bestimmt die Farbe der i-ten LED. Die Farbe kann jede in der FastLED-Bibliothek
voreingestellte Farbe sein, z.B. macht leds[i] = CRGB::Red die Farbe Rot.
```

```
fill_solid(leds, NUM_LEDS, CRGB::Color);
// Füllt den gesamten LED-Streifen mit einer einzigen Farbe.
```

```
fill_rainbow(leds, NUM_LEDS, uint8_t initialHue, uint8_t deltaHue);
// Weist jeder LED im Streifen eine Farbe zu, um einen kontinuierlichen Regenbogeneffekt
zu erzeugen.
```

uint8\_t initialHue: Startfarbton im HSV-Farbmodell, repräsentiert die Farbart (z.B. Rot, Blau). Die Größe dieses Werts ist ein Byte.

uint8\_t deltaHue: Veränderung des Farbtons zwischen den LEDs. Kleinere Werte führen zu sanfteren Übergängen, während größere Werte deutlichere Farbsprünge erzeugen.

```
CHSV(hue, saturation, value);
// Setzt die Farbe unter Verwendung des HSV-Farbmodells fest, wobei 'hue' für den Farbton,
'saturation' für die Sättigung und 'value' für die Helligkeit steht. Jeder dieser Werte ist ein
Byte groß.
```

```
FastLED.show();
// Aktualisiert die LEDs, um die vorgenommenen Änderungen anzuzeigen.
```

Als Beispiel kann ein Design einer gelben, fließenden Wasserlampe genommen werden:

```
void thinkEffect() {
  for (int i = 1; i < NUM_LEDS - 1; i++) {
    leds[i] = CRGB::Yellow; // setzt LED auf Gelb
    FastLED.show();
    delay(1500); // Dauer
    leds[i] = CRGB::Black; // löscht LED-Farbe
    FastLED.show();
  }
  leds[NUM_LEDS - 1] = CRGB::Yellow; // setzt LED auf Gelb
  FastLED.show();
  delay(1500); // Dauer
  leds[NUM_LEDS - 1] = CRGB::Black; // löscht LED-Farbe
  FastLED.show();
}
```

Beim Entwurf einer Fließwasserlampe kann durch Einsatz der Anweisungen `CRGB::Color` und `FastLED.show()` innerhalb einer `for`-Schleife die integrierte Farbe für jede LED gemäß ihrer Reihenfolge festgelegt werden. Anschließend wird mit `CRGB::Black` die Farbe der jeweils vorherigen LED gelöscht, indem sie auf Schwarz gesetzt wird. Um die Farbe einer LED über einen bestimmten Zeitraum zu erhalten, wird `delay(x)` verwendet, wobei `x` die Dauer in Millisekunden angibt. Dabei ist zu berücksichtigen, dass die Indizierung im Array mit 0 beginnt, was bedeutet, dass beispielsweise das sechste LED im Array als das fünfte angesprochen wird.

Das Design eines Abschnitts mit bunten Lauflichtern wird wie folgt realisiert:

```
void rainbowThoughtEffect() {
  static byte hue = 0;
  for (int i = 1; i < NUM_LEDS - 1; i++) {
    leds[i] = CHSV(hue, 255, 255);
    FastLED.show();
    delay(50);
    leds[i] = CRGB::Black;
  }
}
```



```
FastLED.show();
hue += 21;
}
leds[NUM_LEDS - 1] = CHSV(hue, 255, 255);
FastLED.show();
delay(200);
leds[NUM_LEDS - 1] = CRGB::Black;
FastLED.show();
}
```

Zur Einstellung der Farben wird das HSV-Farbmodell eingesetzt. Mit der kontinuierlichen Zunahme des Farbtons bei jedem Durchlauf variieren die resultierenden Farben erheblich. Durch Tests wurde festgestellt, dass eine Erhöhung des Farbtons um 21 bei jedem Schritt eine deutliche und effektive Veränderung bietet. Ein größerer Inkrementwert ermöglicht eine noch ausgeprägtere Darstellung der Farbübergänge, wodurch die Farbveränderungen markanter werden.

Ein weiteres Beispiel für die Anwendungsmöglichkeiten von FastLED ist das Design eines regenbogenähnlichen Fließeffekts, das durch die Unterstützung der Bibliothek erheblich vereinfacht wurde:

```
void rainbowFlowEffect() {
  static byte hue = 0;
  fill_rainbow(leds, NUM_LEDS, hue, 7);
  FastLED.show();
  hue += 2;
  delay(5);
}
```

In diesem Beispiel wird ausschließlich die Funktion `fill_rainbow()` verwendet. Durch die kontinuierliche Anpassung des Ausgangsfarbtons lässt sich ein umfassender Regenbogeneffekt realisieren.

## 6. Optimierungen zur Reduzierung des Stromverbrauchs

Nach der vollständigen Bestätigung der Funktionsweise des Anhängers richtet sich der Fokus nun auf die Optimierung dieser Abläufe. Aktuelle Tests zeigen, dass das ESP8266EX-Modul im WiFi-Modus (AP-Modus und STA-Modus) einen Gesamtstromverbrauch von 70mA hat. Unter Berücksichtigung der Nichtnutzung des TX-Modus und der Leistungsdaten des ESP8266EX aus Abbildung 14 lässt sich ableiten, dass der ESP8266EX hauptsächlich im RX-Modus arbeitet, mit einem Stromverbrauch des HF-Moduls zwischen 50 und 56mA. Bei einer Betriebsspannung von 3V verbraucht eine einzelne LED unter maximaler Helligkeit und weißem Licht bis zu 10mA (und normalerweise ist der Stromverbrauch weniger als 5mA), was den ESP8266EX zum Hauptenergieverbraucher im WiFi-Modus macht. Aber eine Analyse des Energieverbrauchs verschiedener LED-Farben könnte ebenfalls zur Vermeidung des Einsatzes energieintensiver Farben beitragen.

Derzeit ist das HF-Modul durchgehend aktiv, obwohl es nur für die Farbkonfiguration zur Kommunikation mit dem Anhänger benötigt wird. Eine Überarbeitung des Gesamtarbeitsablaufs steht daher an. In der neuen Konfiguration sollte geprüft werden, wie das HF-Modul oder andere Module bei Nichtgebrauch deaktiviert werden können. Dieser Ansatz kann mit einer detaillierten Analyse der Funktionsweise des ESP8266EX und des WiFi-Modus in Kapitel 5 beginnen.

### 6.1 WLAN-Modus auszuschalten

Im Abschnitt 5.1.2 werden die drei offiziellen Schlafmodi des ESP8266EX vorgestellt:

- Modem-Schlafmodus (die CPU bleibt aktiv, während Wi-Fi und Funk teilweise deaktiviert sind)
- Leichter Schlafmodus (CPU und alle Peripheriegeräte sind inaktiv, und jedes Aufwacheignis aktiviert den Chip wieder)
- Tiefschlafmodus (nur die RTC bleibt aktiv, während die anderen Teile des Chips inaktiv sind)

Da die RGB-LED die CPU zur Berechnung der aktuellen Farbe benötigt und keine anderen externen Geräte angeschlossen sind, ist das HF-Modul das einzige Modul, das abgeschaltet werden kann. Dies entspricht genau dem Modem-Schlafmodus unter diesen drei Schlafmodi.

#### 6.1.1 Modem-Schlafmodi

Es ist zu beachten, dass im Modem-Schlafmodus das HF-Modul nicht vollständig abgeschaltet, sondern zeitweise aktiviert wird. Dieser Modus wird für Anwendungen genutzt, in denen die CPU kontinuierlich arbeiten muss, wie z.B. bei PWM-Anwendung. Falls keine Datenübertragung während der Aufrechterhaltung einer Wi-Fi-Verbindung stattfindet, kann der Wi-Fi-Modemschaltkreis gemäß dem 802.11-Standard abgeschaltet werden, um Energie zu sparen.

Es gibt allerdings eine Einschränkung bei der Nutzung des Modem-Schlafmodus. Im SoftAP-Modus agiert der ESP8266 als WLAN-Zugangspunkt, was eine kontinuierliche

Signalübertragung erforderlich macht, damit andere Geräte eine Verbindung herstellen können. Daher muss das WLAN-Modul aktiviert bleiben, was dem Prinzip des Modem-Schlafmodus entgegensteht. Folglich ist der Modem-Schlafmodus üblicherweise mit dem Stationsmodus (STA-Modus) verbunden und dient dazu, den Energieverbrauch zu reduzieren, wenn das Gerät mit einem Wi-Fi-Netzwerk verbunden ist.

Obwohl der Modem-Schlafmodus nicht direkt im SoftAP-Modus angewendet werden kann, ist die Messung des tatsächlichen Betriebsstroms in diesem Modus dennoch von Referenzwert. Die Tests wurden im STA-Modus durchgeführt, wobei der Stromverbrauch nach zwei Durchläufen unregelmäßig von 15 mA auf 40 mA anstieg, was grundsätzlich mit dem im Datenblatt beschriebenen Betriebsmechanismus des Modem-Schlafmodus übereinstimmt.

### 6.1.2 HF-Modul direkt auszuschalten

In der Definition der ESP8266WiFiType.h-Bibliothek gibt es eine Möglichkeit, WLAN direkt auszuschalten.

```
typedef enum WiFiMode
{   WIFI_OFF = 0, WIFI_STA = 1, WIFI_AP = 2, WIFI_AP_STA = 3
} WiFiMode_t;
```

Nachdem die Methode zum Abschalten entdeckt wurde, wurde versucht, das WiFi-Modul mit `WiFi.mode(WIFI_OFF)` direkt zu deaktivieren.

Es zeigt sich jedoch, dass nach der Deaktivierung des AP-Modus das durch den AP-Modus erstellte Netzwerk korrekt abgeschaltet wird (es verschwindet aus der WLAN-Auswahlliste des Mobiltelefons), der tatsächliche Stromverbrauch sich jedoch nicht verändert. Dies könnte an der begrenzten Hardwarekontrolle liegen, die zur Verfügung steht, wenn das offizielle SDK nicht genutzt wird - die Arduino IDE ermöglicht keinen tieferegehenden Zugriff auf die Hardware.

Da die bisherige Methode nicht zur Energieeinsparung beiträgt und der aktuell verwendete Code nicht für die Programmierung mit dem SDK geeignet ist, muss eine alternative Lösung zur Deaktivierung des HF-Moduls gefunden werden.

Bei der Nutzung der WLAN-Modus des ESP8266EX müssen diese explizit im Code konfiguriert werden. Interessanterweise bleibt das HF-Modul ausgeschaltet, bis ein WLAN-Modus aktiv konfiguriert wird. Messungen zufolge liegt der Stromverbrauch vor dieser Konfiguration bei etwa 15 mA. Dies deutet darauf hin, dass es möglich sein sollte, den Zustand des ESP8266EX auf den Zeitpunkt vor der Konfiguration des WLAN-Modus zurückzuführen, um sicherzustellen, dass das HF-Modul nicht aktiviert wird.

Für diesen Prozess ist es erforderlich, einen nichtflüchtigen Speichertyp zu nutzen, der Daten auch nach einem Stromausfall bewahrt. Daher bietet sich die Nutzung des EEPROMs als Lösung an. Obwohl der ESP8266EX keinen physischen EEPROM-Chip besitzt, ermöglicht es sein interner Flash-Speicher, einen Bereich zur Simulation eines EEPROMs zuzuweisen. Diese Simulation erfolgt tatsächlich im internen Flash-Speicher des ESP8266, indem ein spezifischer Speicherbereich für EEPROM-Funktionen reserviert wird. Die Implementierung wird durch die Integration der speziellen Bibliothek `EEPROM.h` ermöglicht.

Diese Anpassung zielt darauf ab, die technischen Informationen präzise und zugänglich zu machen, und unterstreicht die Fähigkeit des ESP8266, Funktionen eines EEPROMs ohne physischen Chip zu simulieren.

Unter anderem ist die Verwendung von EEPROM auf ESP8266EX wie folgt:

- Initialisierung: Mit `EEPROM.begin(size)` wird das EEPROM initialisiert, wobei `size` die Anzahl der Bytes angibt, die genutzt werden sollen.
- Daten lesen: Ein Byte von einer bestimmten Adresse wird mit `EEPROM.read(address)` gelesen.
- Daten schreiben: Ein Byte wird an eine spezifische Adresse mit `EEPROM.write(address, value)` geschrieben.
- Änderungen speichern: Änderungen im EEPROM werden mit `EEPROM.commit()` gespeichert.
- Ende der Nutzung: Der Betrieb des EEPROMs wird mit `EEPROM.end()` beendet und Ressourcen werden freigegeben.

Dies ist eine praktikable Methode und der Hauptprozess ist wie folgt:

- Empfangen des Farbmodusbefehls im WLAN-Modus.
- Schreiben der Farbmodusnummer in das EEPROM.
- Neustarten.
- Lesen der Farbmodusnummer im EEPROM und Ausführen des Farbmodus, ohne das Hochfrequenzmodul einzuschalten.

Der Betriebsmodus nach dem Ausschalten des HF-Moduls kann als Energiesparmodus bezeichnet werden. Um diesen Modus in einen frei konfigurierbaren Modus zu setzen, kann dies durch das Erkennen externer Tastensignale über den GPIO-Port erreicht werden. Nachdem ein logischer Bewertungsteil und wichtige Anti-Shake-verarbeitung hinzugefügt wurden, sieht der spezifische Implementierungscode wie folgt aus:

```
#define buttonPin 0 // GPIO-Pin für die externe Tastenverbindung festlegen
bool apModeActive = false; // Variable zum Aufzeichnen des SoftAP-Modusstatus
bool buttonPressed = false; // Variable, die den Status externer Tasten aufzeichnet

void setup() {
  EEPROM.begin(3); // EEPROM initialisieren, Größe nach Bedarf anpassen
  pinMode(buttonPin, INPUT_PULLUP); // Internen Pull-up-Widerstand aktivieren

  effect = EEPROM.read(0); // Erste Variable, z. B. Farbeffekt, lesen
  color = EEPROM.read(1); // Zweite Variable, z. B. eine feste Farbe, lesen
  brightness = EEPROM.read(2); // Dritte Variable, z. B. Helligkeit, lesen
  // Weitere Initialisierung
}

void loop() {
  int buttonState = digitalRead(buttonPin); // Externen Tastenstatus lesen

  if (buttonState == LOW && !buttonPressed) {
```

```
buttonPressed = true;
delay(50); // Einfache Anti-Shake-Verzögerung

if (!apModeActive) {
  WiFi.mode(WIFI_AP); // In den AP-Modus wechseln
  WiFi.softAP(ssid, password);
  apModeActive = true;
  delay(1000); // Zeit für den Moduswechsel lassen
} else {
  WiFi.mode(WIFI_OFF); // WLAN ausschalten
  apModeActive = false;
  EEPROM.write(0, static_cast<uint8_t>(effect));
  EEPROM.write(1, static_cast<uint8_t>(color));
  EEPROM.write(2, static_cast<uint8_t>(brightness));
  EEPROM.commit();
  ESP.restart(); // ESP8266 neu starten
}
} else if (buttonState == HIGH && buttonPressed) {
  buttonPressed = false;
  delay(50); // Einfache Anti-Shake-Verzögerung
}

//CPU-Aufgaben im SoftAP-Modus ausführen
if (apModeActive) {
...
}
// CPU-Aufgaben im Energiesparmodus ausführen
if (!apModeActive) {
...
}
}
```

Durch den internen Pull-up-Widerstand erhöht sich der Pegel von buttonPin und verringert sich beim Drücken des externen Knopfes. Im logischen Bewertungsteil erfolgt ein Moduswechsel nur, wenn die Bedingungen buttonState == LOW und !buttonPressed gleichzeitig erfüllt sind, um mehrfache Erkennungen desselben Druckvorgangs zu vermeiden. Die Verwendung von static\_cast<uint8\_t> stellt sicher, dass die Variablen Effekt, Farbe und Helligkeit als 8-Bit-Werte in das EEPROM geschrieben werden, entsprechend der Anforderung von EEPROM.write(), die ein 8-Bit-Datenbyte erfordert. Dies verhindert Datenverlust oder Fehler beim Schreiben, falls der ursprüngliche Variablentyp mehr als 8 Bit umfasst (z.B. int).

Durch diese Methode wurde dem ESP8266EX erfolgreich ein Energiesparmodus hinzugefügt, der es ermöglicht, den letzten Farbmodus auch nach einem Neustart zu bewahren. Interessanterweise betrug der gemessene Betriebsstrom des ESP8266EX konstant

etwa 15,6 mA, was den Angaben im Datenblatt des ESP8266EX entspricht. Diese Methode erweist sich hinsichtlich der Steuerungsanforderungen von RGB-Anhängern als effektiver im Vergleich zum offiziell voreingestellten Modem-Schlafmodus.

Das endgültige vollständige Flussdiagramm ist wie Abbildung 16 dargestellt.

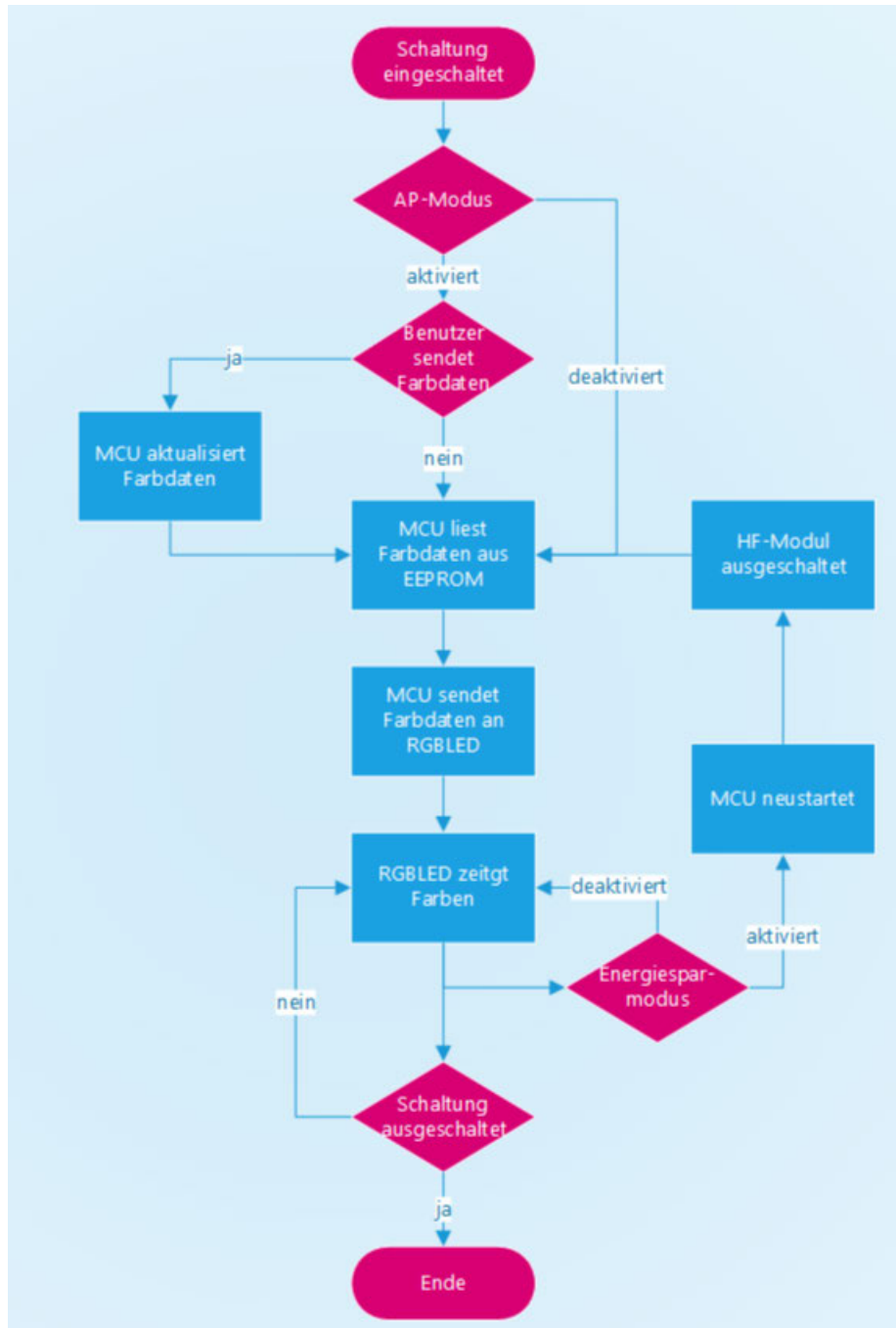


Abbildung 16: Flussdiagramm

## 6.2 Farbe auszuwählen

Bei den Energiesparmaßnahmen für WS2812D-LEDs spielt die Farbauswahl auch eine wichtige Rolle. Da die endgültige Farbe einer LED durch das Mischen der drei Primärfarben Rot, Grün und Blau entsteht, führt die Wahl bestimmter Farben zu einem geringeren

Energieverbrauch. Als Beispiel dient die Farbe Weiß mit den Farbdaten 0xFFFFFF, die durch die maximale Intensität aller drei Farbkomponenten charakterisiert ist und daher den höchsten Stromverbrauch aufweist.

Für die langfristige Anzeige einer Farbe empfiehlt es sich, Farben mit niedrigeren Intensitätswerten der Farbkomponenten zu wählen oder Farben, die nur aus zwei oder einer der RGB-Farben zusammengesetzt sind. Praktische Messungen mit einer Gruppe von sechs LEDs bei einer Spannung von 3V haben gezeigt, dass der Stromverbrauch bei maximaler Intensität einer Einzelfarbe etwa 23mA beträgt. Im Vergleich dazu verbraucht eine Mischfarbe aus zwei Farben, wie Gelb (Farbdaten 0xFFFF00), etwa 42.7mA. Diese Beobachtungen bestätigen, dass der Stromverbrauch weitgehend dem Prinzip der einfachen Addition folgt.

Durch die bewusste Auswahl der Farben für die Langzeitanzeige kann somit nicht nur Energie gespart, sondern auch eine breite Farbpalette beibehalten werden.

### 6.3 Andere Möglichkeiten für Energiessparen

Derzeit gibt es zwei weitere Möglichkeiten, Energie zu sparen: die Reduzierung der Sendeleistung des HF-Moduls und die Reduzierung der Kernfrequenz des ESP8266EX.

Unter anderem kann die Reduzierung der Sendeleistung des HF-Moduls nur durch die Steuerung der zugrunde liegenden Hardware über das offizielle SDK erreicht werden. Es ist jedoch zu beachten, dass ESP8266EX, da derzeit der RX-Modus verwendet wird, nur für den Empfang von Signalen verantwortlich ist, sodass eine Reduzierung der Sendeleistung theoretisch nicht möglich ist.

Die Reduzierung der Kernfrequenz muss über andere inoffizielle SDKs erreicht werden, da die Reduzierung der Kernfrequenz zu starker Unsicherheit führen kann, wenn die MCU läuft, und der Strom bei laufender MCU nur etwa 15 mA beträgt, selbst wenn diese Methode erfolgreich ist. Dies führt jedoch nicht zu einer großen Änderung des Gesamtenergieverbrauchs, daher wird angenommen, dass diese Methode vorerst den Stromverbrauch nicht effektiv senkt.

## 7. Batterieauswahl und Messung der Batterielebensdauer

Bei den in Anhängern verwendeten Batterien sind Kostengünstigkeit, Miniaturisierung, geringes Gewicht und lange Batterielebensdauer die Hauptfaktoren. Allerdings sind die Parameter verschiedener Batterien (z. B. Innenwiderstand, Abschaltspannung) unterschiedlich. Aus Kostengründen werden in diesem Kapitel daher Knopfzellen und Alkali-AAA-Batterien zum Vergleich ausgewählt. Gleichzeitig basiert dies auf der tatsächlichen Leistung von Der RGB-Anhänger, Anforderungen besprechen, wie man eine Batterie auswählt.

### 7.1 Knopfzelle

Bei Miniaturisierung und geringem Gewicht als Hauptanforderungen kann die Verwendung von Knopfzellen für die Stromversorgung in Betracht gezogen werden. Als Beispiel dient die CR2032, eine Mangandioxid-Lithiumzelle, die von Größe und Form einer Münze ähnelt. Je nach Marke wiegt der Akku etwa 3 g, bietet eine Kapazität von ungefähr 220 mAh und eine Nennspannung von 3 V. Obwohl die Kapazität nicht besonders groß ist, ist die Nennspannung einer Knopfzelle ausreichend, um MCUs und RGB-LEDs zu betreiben. Ihre geringe Größe und das leichte Gewicht machen sie zur ersten Wahl für kleine elektronische Produkte.

Dies scheint eine ideale Option zu sein, es gibt jedoch Schwierigkeiten bei der Verwendung von Knopfzellen zur Stromversorgung. Am Beispiel des CR2032 von muRata beträgt sein Standardentladestrom 0,2 mA, der maximale Laststrom des Anhängers erreicht jedoch 100 mA, was zu bedeuten scheint, dass er für dieses Design nicht geeignet ist. Der Standard-Entladestrom ist jedoch immerhin 0,2 mA. Ob sich eine Erhöhung des Entladestroms auf die Nutzung auswirkt, ist noch fraglich. Aus diesem Grund muss in diesem Abschnitt bestätigt werden, ob Knopfzellen für RGB-Anhänger verwendet werden können.

Der Innenwiderstand der Zelle bestimmt ihre Zellenspannung unter tatsächlicher Last. Ein hoher Innenwiderstand führt dazu, dass die Zellenspannung erheblich abfällt, was dazu führen kann, dass der PCB-Anhänger nicht richtig funktioniert. Daher muss die Änderung



des Batteriewiderstands bestimmt werden. Als beispiel kann Abbildung 17 genommen werden.

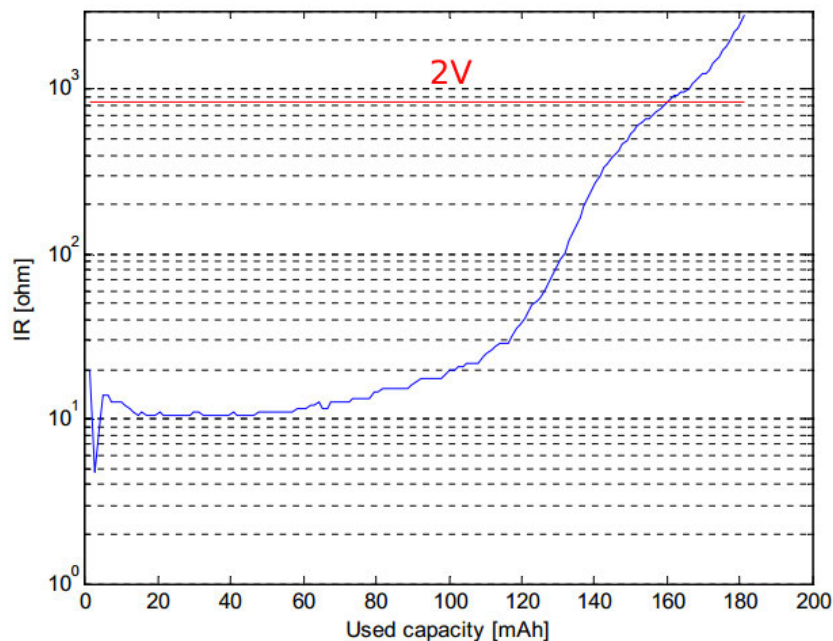


Abbildung 17: Änderung des Innenwiderstands 1[6]

Bei Verwendung einer neuen Knopfatterie beträgt ihr anfänglicher Innenwiderstand  $R_I$  etwa  $10 \Omega$ , und wenn der Anhänger im WiFi-Modus arbeitet, beträgt ihr Laststrom  $I_L=100$  mA, selbst wenn die Anfangsspannung  $V_A=3,3$  V (die Spannung einer neuen Batterie) beträgt ist normalerweise höher als die Nennspannung), die tatsächliche Spannungsabgabe der Batterie zu diesem Zeitpunkt

$$V_O = V_A - I_L \cdot R_I = 2.3V$$

$V_O=2,3V < 2,6V$ . Es ist ersichtlich, dass der zu große Innenwiderstand der Knopfatterie direkt zu einem erheblichen Abfall der Batteriespannung führt und die ordnungsgemäße Funktion des Stromkreises beeinträchtigt.

Aber ist es möglich, die Spannung durch eine Boost-schaltung auf 3,3 V zu halten? Dies ist tatsächlich jedoch auch problematisch, da der Innenwiderstand auch die Wärme erhöht. Wenn die Kapazität abnimmt und die Spannung während der Batterieentladung abnimmt, erhöht sich der Innenwiderstand der Batterie und die Effizienz der Boost-Schaltung nimmt ab (was zu einem Anstieg des Ausgangsstroms des Batteriepol führt), was dazu führt Die Leistung der Knopfatterie erhöht sich mit zunehmender Nutzung. Die Wärmeentwicklung nimmt mit der Zeit immer weiter zu.

$$V_B \cdot I_B \cdot \eta = V_L \cdot I_L$$

$$\Delta T = \frac{Q}{mc} = \frac{I_B^2 R_I t}{mc}$$

$\Delta T$  ist die sich ändernde Temperatur,  $Q$  ist die Wärme,  $m$  ist die Masse der Knopfatterie und  $c$  ist die spezifische Wärmekapazität der Knopfatterie.

Wenn die Masse unverändert bleibt und sich die spezifische Wärmekapazität geringfügig ändert, nimmt der Innenwiderstand  $R_I$  der Batterie zu, wenn die Kapazität abnimmt und die Spannung  $V_B$  während der Batterieentladung abnimmt, und der Wirkungsgrad  $\eta$  der Boost- und Buck-Schaltung nimmt ab (resultierend). Im Batterieanschluss steigt der Ausgangsstrom  $I_B$ , was dazu führt, dass die Knopfatterie mit der Zeit die Wärmeerzeugung  $Q$  beschleunigt, was zu einem Anstieg von  $\Delta T$  führt.

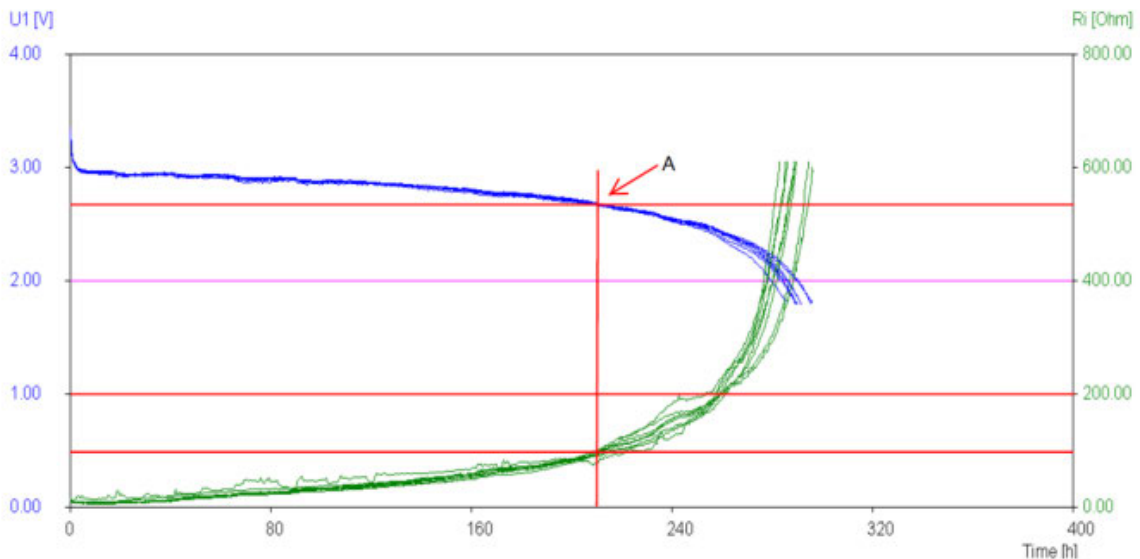


Abbildung 18:Änderung des Innenwiderstands 2[7]

Unterstützt durch Daten aus einem anderen Experiment, gemäß der Abbildung 18, ist die rote Linie, an der sich A befindet, wenn die Spannung der Knopfatterie 2,6 V beträgt. Zu diesem Zeitpunkt hat ihr Widerstand 100 Ω erreicht, und dieser Zeitraum liegt noch innerhalb der normalen Lebensdauer der Knopfatterie. Wenn zu diesem Zeitpunkt der WLAN-Modi aktiviert ist, wird die durch den Innenwiderstand der Batterie erzeugte Wärmeleistung unerträglich. Mit der Zeit steigt die Temperatur weiter an. Zu diesem Zeitpunkt wird die Stromversorgung durch die Knopfatterie nicht mehr stabil noch sicher.

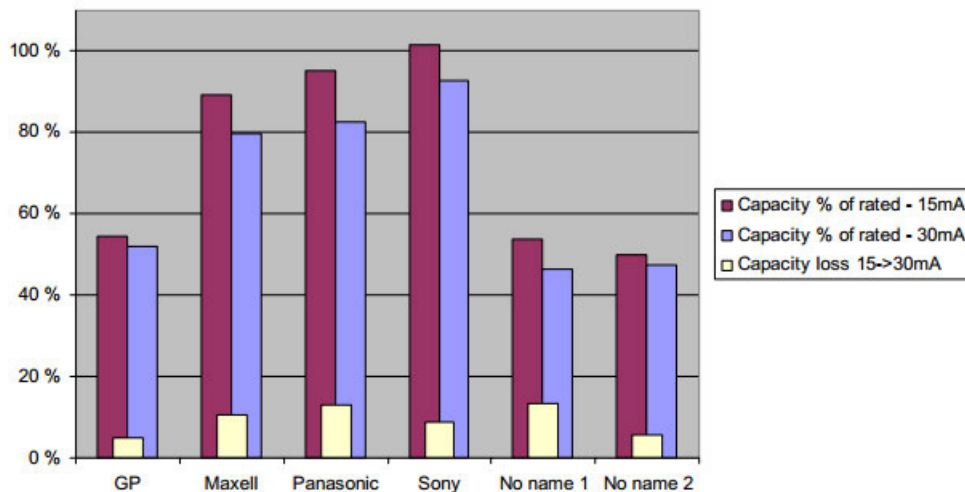


Abbildung 19:Änderung der effektiven Kapazität[6]

Gleichzeitig führt ein hoher Laststrom auch direkt zu einer Verringerung der effektiven Kapazität der Batterie. Die Abbildung 19 zeigt ein Diagramm des effektiven Kapazitätsverhältnisses der Batterie beim Laden von Spitzenströmen von 15 mA und 30 mA. Die Abbildung vergleicht Knopfzellen verschiedener Marken. Mit steigendem Strom nimmt die effektive Kapazität deutlich ab. Bei Lasten von 100 mA und mehr wird die Abnahme der effektiven Kapazität zu einem wesentlichen Einflussfaktor.

Im Allgemeinen eignen sich Knopfzellen besser für die Übertragung kurzzeitiger Stromspitzen als für die kontinuierliche Übertragung hoher Lastströme. Daher sind Knopfzellen aus den oben genannten Gründen für den Einsatz in dieser Bauform nicht geeignet.

## 7.2 AAA Batterie

AAA-Batterien sind weit verbreitete kleine Trockenzellenbatterien. sind kleiner als AA-Batterien und werden häufig in tragbaren elektronischen Geräten verwendet. Seine Standardgröße beträgt normalerweise 10,5 mm Durchmesser und 44,5 mm Länge. Nicht wiederaufladbare Modelle haben eine Nennspannung von 1,5 Volt. Die Kapazität von alkalischen AAA-Batterien liegt normalerweise zwischen 1000 mAh und 1200 mAh, was fast dem Fünffachen der Kapazität von Knopfzellen entspricht. Das Gewicht beträgt etwa 11 g, was etwa dem 3,7-fachen einer Knopfzelle entspricht.

Unter den AAA-Batterien sind R03-Batterien und LR03-Batterien am häufigsten. R03-Batterien sind Zink-Kohle-Batterien. Dieser Batterietyp kostet weniger, hat aber eine geringere Kapazität und eine höhere Selbstentladungsrate als LR03-Batterien. R03-Batterien werden typischerweise für Geräte mit geringem Stromverbrauch wie Fernbedienungen oder Wanduhren verwendet. LR03-Batterien sind alkalische (Alkaline-)Batterien. Im Vergleich zu Kohlenstoff-Zink-Batterien haben Alkalibatterien eine höhere Energiedichte und eine längere Lebensdauer. halten auch höheren Strombelastungen besser stand und eignen sich daher für den Einsatz in stromhungrigeren Geräten wie Spielzeug und Taschenlampen.

Wenn man bedenkt, dass LR03-Alkalibatterien in Bezug auf Kapazität und Haltbarkeit im Allgemeinen besser sind als R03-Kohle-Zink-Batterien. Daher wird in diesem Abschnitt LR03 zur Diskussion ausgewählt.

Da die Spannung von AAA-Batterien nur 1,5 V beträgt, sind zum Erreichen der minimalen Betriebsspannung von 2,6 V zwei AAA-Batterien erforderlich.

Denn die Spannung der AAA-Batterie beträgt nur 1,5V, um ohne Verwendung einer Boost-Schaltung die minimale Betriebsspannung des LED-Blaulichts von 2,65V zu erreichen und gleichzeitig die maximale Spannung des ESP8266EX von 3,6 nicht zu überschreiten V müssen zwei AAA-Batterien verwendet werden, die zur Stromversorgung in Reihe geschaltet werden.

Da die grundlegenden Parameter wie Innenwiderstand und Kapazität je nach Batteriemarke stark variieren, kann der theoretische Batterielebensdauertest nur als Schätzung zur Referenz herangezogen werden.

Am Beispiel der MN2400-Batterie der Duracell Company ist ihre Entladekurve in Abbildung 20 und Abbildung 21 dargestellt

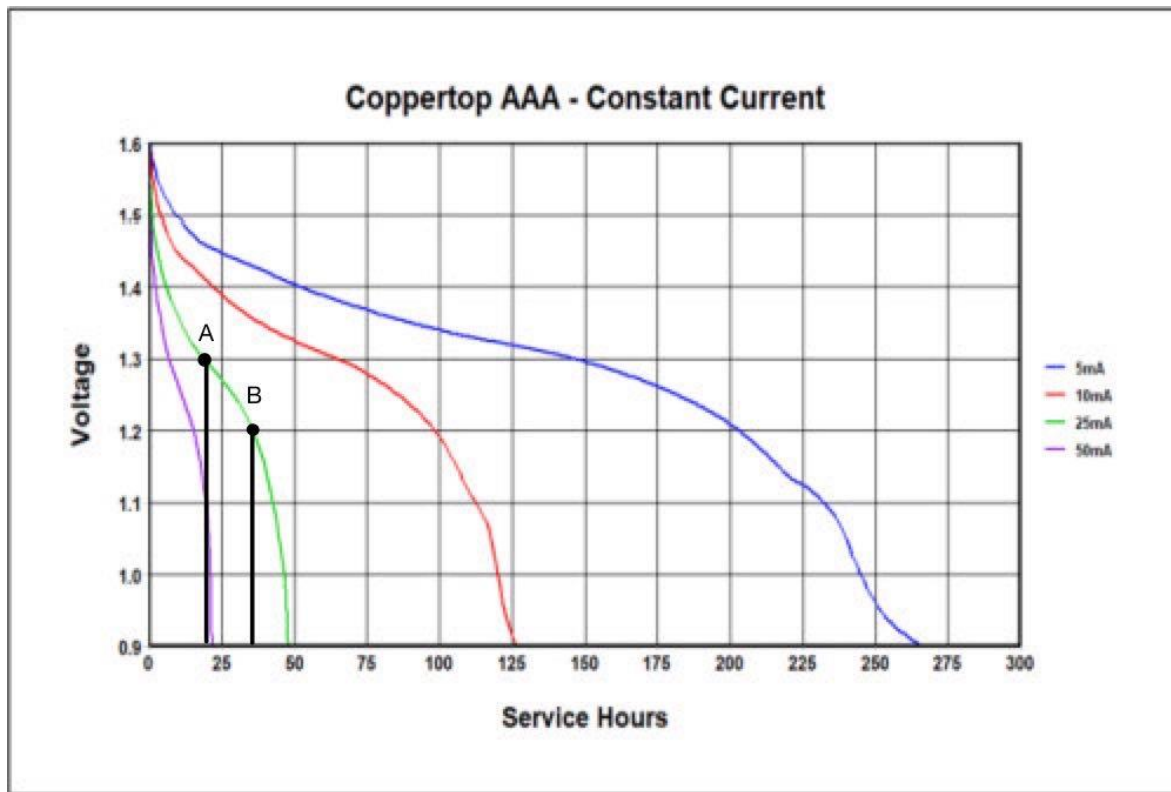


Abbildung 20:Entladekurve 1[8]

Der Mindeststrom des ESP8266EX im Energiesparmodus beträgt derzeit 15 mA. Der Gesamtstrom der Schaltung setzt sich aus diesem Wert und dem Stromverbrauch der LED zusammen. Unter den Farbmodus, den derzeit entworfen wird, hat der Farbmodus, der am wenigsten Strom verbraucht, einen Strom von etwa 5,5 mA. Daher kann die 25-mA-Kurve in Abbildung 18 als Beispiel genommen werden, um die theoretische maximale Arbeitszeit grob abzuschätzen.

Da die minimale Betriebsspannung des blauen Teils der LED 2,65 V beträgt, entspricht dies einer Einzelbatteriespannung von 1,325 V. Nimmt man also Punkt A in der Abbildung als Referenz, beträgt die maximale normale Arbeitszeit etwa 17,5 Stunden.

Die minimale Antriebsspannung des grünen Teils der LED beträgt 2,4 V, und die minimale Antriebsspannung des HF-Moduls beträgt 2,35 V, beträgt die maximale Betriebszeit des HF-Moduls unter Verwendung von Punkt B in der Abbildung etwa 37,5 Stunden.

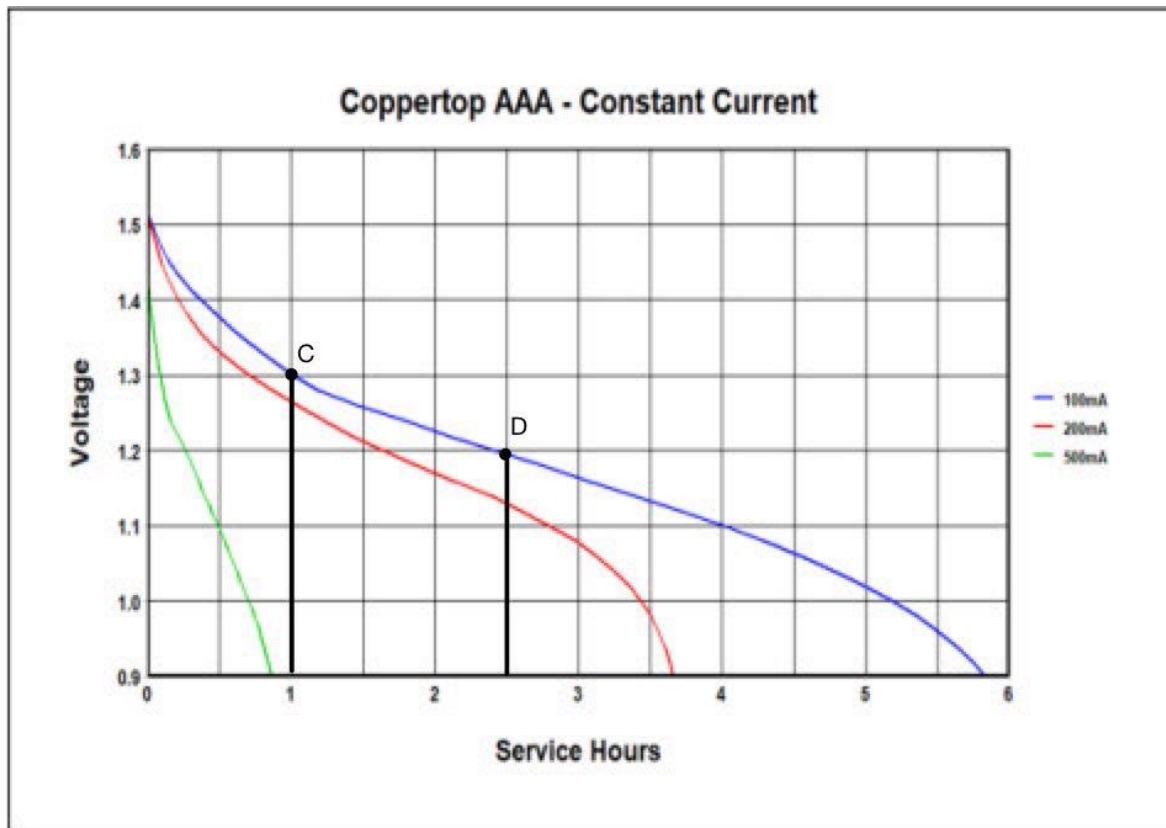


Abbildung 21:Entladekurve 2[8]

Wenn die Farbe über ESP8266EX eingestellt werden muss, d.h. wenn das HF-Modul eingeschaltet wird, steigt der Gesamtstrom um 50 bis 55 mA. Zu diesem Zeitpunkt erreicht der Gesamtstrom von ESP8266EX 70 mA. Auf dieser Basis muss der Gesamtstrom zum LED-Strom addiert werden. Daher kann man die 100-mA-Kurve in Abbildung 19 als Referenz verwenden.

Bei einer Spannung von 1,3 V, also Punkt C als Referenz, beträgt die maximale Zeit, in der die LED normal arbeiten kann, wenn das HF-Modul kontinuierlich eingeschaltet ist, etwa 1 Stunde. Nimmt man die Spannung bei 1,2 V, also Punkt D als Referenz, beträgt die maximale Zeit, die das HF-Modul normal arbeiten kann, wenn das HF-Modul kontinuierlich eingeschaltet ist, etwa 2,5 Stunden.

Ob das HF-Modul des ESP8266EX aktiviert ist, spielt eine entscheidende Rolle für die Betriebsdauer. Die durchgeführte Gegenüberstellung unterstreicht zudem die Wichtigkeit der Optimierungsmaßnahmen, die in Kapitel 5 diskutiert werden.

## 8. Implementierung der Boost-Schaltung

Derzeit liegt der Spannungsbereich von zwei verwendeten Alkali-AAA-Batterien zwischen 1,8 V und 3,3 V. Wenn die Batteriespannung unter 2,65 V fällt, kann die LED nicht richtig funktionieren, was tatsächlich zu einer Verkürzung der Batterielebensdauer führt. Dieses Problem kann durch Hinzufügen einer Boost-Schaltung gelöst werden.

### 8.1 Boost-Schaltung Grundlage

Beim Entwurf elektronischer Schaltkreise ist der DC-DC-Aufwärtswandler eine häufig verwendete Methode zur Spannungserhöhung. Seine Vorteile sind geringes Gewicht, niedrige Kosten, einfacher Aufbau und sehr hoher Wirkungsgrad. Diese Eigenschaften werden auch in diesem Entwurf verwendet. erforderlich. Die Topologie eines Standard-Aufwärtswandlers ist in Abbildung 22 dargestellt.

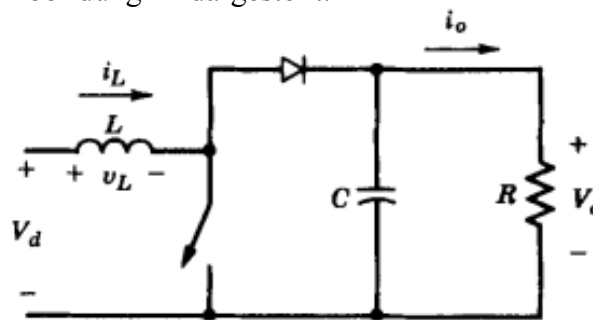


Abbildung 22: Boost konverter Struktur

Sein Funktionsprinzip ist hauptsächlich in zwei Stufen unterteilt. Der Schaltkreis, wenn der Schalter (in praktischen Anwendungen normalerweise ein MOSFET) geschlossen ist, ist in Abbildung 23(a) dargestellt. Während der Tonne-Zeit, wenn der Schalter geschlossen ist, beträgt die Spannung  $V_L$  an der Induktivität  $L$   $V_d$  (Versorgungsspannung). Zu diesem Zeitpunkt beginnt die Induktivität  $L$ , Energie zu speichern, und der Strom  $i_L$  steigt linear mit der Zeit an. Da die Diode zu diesem Zeitpunkt in Sperrrichtung vorgespannt ist, ist gleichzeitig der Lastanschluss (Kondensator  $C$  und Widerstand  $R$ ) vom Eingangskreis isoliert.

Der Schaltkreis bei ausgeschaltetem Schalter ist in Abbildung 23(b) dargestellt. Wenn der Schalter während der toff-Zeit ausgeschaltet wird, kann sich der Strom der Induktivität  $L$  nicht plötzlich ändern und die Induktivität  $L$  versucht, den Strom konstant zu halten. Die Polarität der Spannung an beiden Enden kehrt sich um und die Diode wird in Vorwärtsrichtung vorgespannt. Der Induktor  $L$  gibt Energie ab und überträgt die Energie an die Last und den Kondensator  $C$ . Zu diesem Zeitpunkt wird der Kondensator  $C$  geladen und die Spannung  $V_o = V_d + V_L$  erzielt einen Boost-Effekt. Wenn der Schalter wieder ausgeschaltet wird, wird der Kondensator  $C$  entladen, wodurch ein gleichmäßiger Gleichstromausgang an die Last  $R$  bereitgestellt wird.

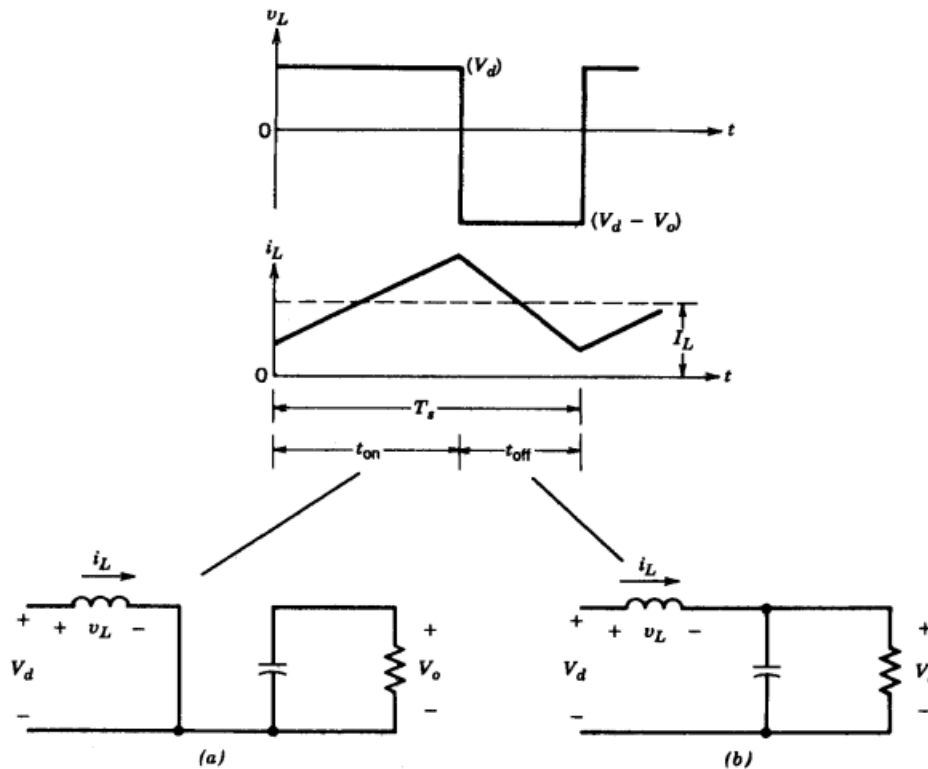


Abbildung 23: Prozess des Schaltungsbetriebs[9]

In diesem Zyklus kann durch Steuerung der Öffnungs- und Schließzeit des Schalters (Tastverhältnis  $D$ ) die Ausgangsspannung gesteuert werden. Darunter ist die Beziehung zwischen der Eingangsspannung  $V_d$  und der Ausgangsspannung  $V_o$ .

$$\frac{V_o}{V_d} = \frac{T_s}{t_{off}} = \frac{1}{1 - D}$$

Angenommen, es handelt sich um eine verlustfreie Schaltung, also  $P_d = P_o$

$$V_d I_d = V_o I_o$$

Und

$$\frac{I_o}{I_d} = (1 - D)$$

In dieser Schaltung bestimmen die Größe der Induktivität, des Kondensators und der Last das Ansprechverhalten und die Stabilität der Schaltung. Die Induktivität bestimmt die Änderungsrate des Stroms, die Kapazität bestimmt die Gleichmäßigkeit und Energiespeicherkapazität der Spannung und die Größe der Last beeinflusst die Ausgangsspannung.

Ein weiterer erwähnenswerter Faktor ist die Welligkeit der Spannung, da für einige Komponenten im eigentlichen Schaltkreis möglicherweise Anforderungen an die Welligkeit der Versorgungsspannung bestehen. Unter der Annahme, dass im kontinuierlichen Modi die

Welligkeitsstromkomponente des Diodenstroms  $I_D$  vollständig durch den Kondensator fließt und ihr Durchschnittswert durch den Lastwiderstand fließt, ist der schattierte Bereich in Abbildung 24 die Ladung  $\Delta Q$ . Die Rippelspannung  $\Delta V_o$  beträgt zu diesem Zeitpunkt

$$\Delta V_o = \frac{\Delta Q}{C} = \frac{I_o DT_s}{C} = \frac{V_o DT_s}{R C}$$

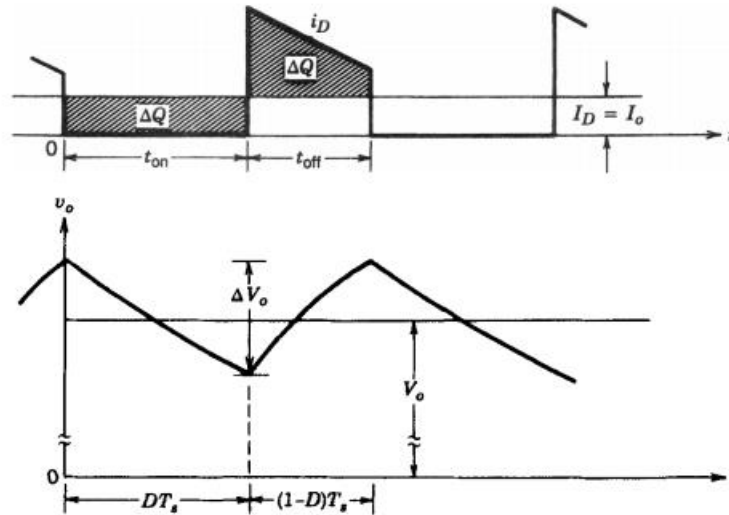


Abbildung 24: Rippelspannung[9]

Auf der Basis des Standard-Aufwärtswandlers verbessert der Synchron-Aufwärtswandler die Effizienz der Aufwärtswandlung weiter. Es handelt sich um einen derzeit auf dem Markt verbreiteten Aufwärtswandler. Sein Aufbau und sein Prinzip ähneln dem Standard-Aufwärtswandler. Der besondere Unterschied zwischen beiden besteht darin, dass der synchrone Aufwärtswandler MOSFETs anstelle von Dioden verwendet. Seine Topologie ist in Abbildung 25 dargestellt.

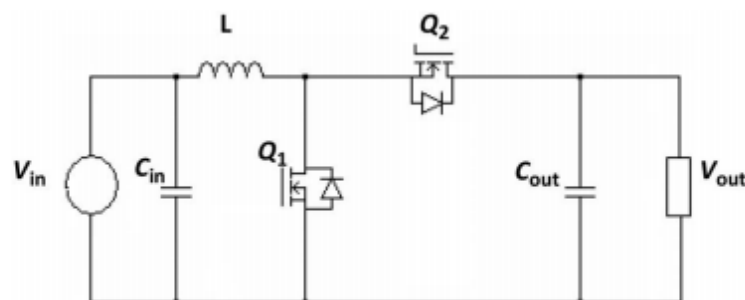


Abbildung 25: Synchroner Boost-Konverter

Der wesentliche Vorteil des synchronen Aufwärtswandlers liegt darin, dass der Energieverlust, der durch den Spannungsabfall der Diode in einem herkömmlichen Aufwärtswandler entsteht, durch den Einsatz eines zweiten Transistors (Q2) anstelle der Diode reduziert wird. Dies führt zu einer verbesserten Umwandlungseffizienz.

Zudem setzen synchrone Aufwärtswandler auf die Verwendung von PWM (Pulsweitenmodulation). Eine präzise Steuerung der Gating-Signale ist zwingend notwendig, um das gleichzeitige Einschalten beider Schalter zu verhindern.



## 8.2 Wahl der Boost-Lösung

Angeichts der Notwendigkeit, weitere MCUs für die Steuerung des Gate-Signals hinzuzufügen, und der begrenzten Fläche der Leiterplatte, stellt die eigenständige Entwicklung einer DC-DC-Aufwärtswandlungsschaltung keine optimale Lösung dar. Stattdessen wird die Verwendung von handelsüblichen Stromversorgungs-Chips empfohlen, um das Design des Boost-Schaltkreises effizient zu gestalten.

Für die LED-Modellreihe wurde ein Spannungsbereich von 3V-5,5V ausgewählt, der teilweise mit dem Spannungsbereich des ESP8266EX (2,6V-3,6V) übereinstimmt, wodurch beide Komponenten mit derselben Spannungsquelle betrieben werden können. Da die typische Betriebsspannung des ESP8266EX bei 3,3V liegt und auf dem Markt viele Aufwärtswandler mit einer festen Ausgangsspannung von 3,3V verfügbar sind, wird diese Spannung als Betriebsspannung genutzt.

Nach einer sorgfältigen Bewertung von Parametern wie maximalem Laststrom, Schwankungsbereich der Eingangsspannung, Effizienz der Aufwärtswandlung und Anzahl der Komponenten wurde der Texas Instruments TPS613221 als geeigneter Boost-Wandler für diesen Schaltungsentwurf ausgewählt.

Tps613221 ist ein synchroner Aufwärtswandler, dessen interne Struktur in Abbildung 26 dargestellt ist.

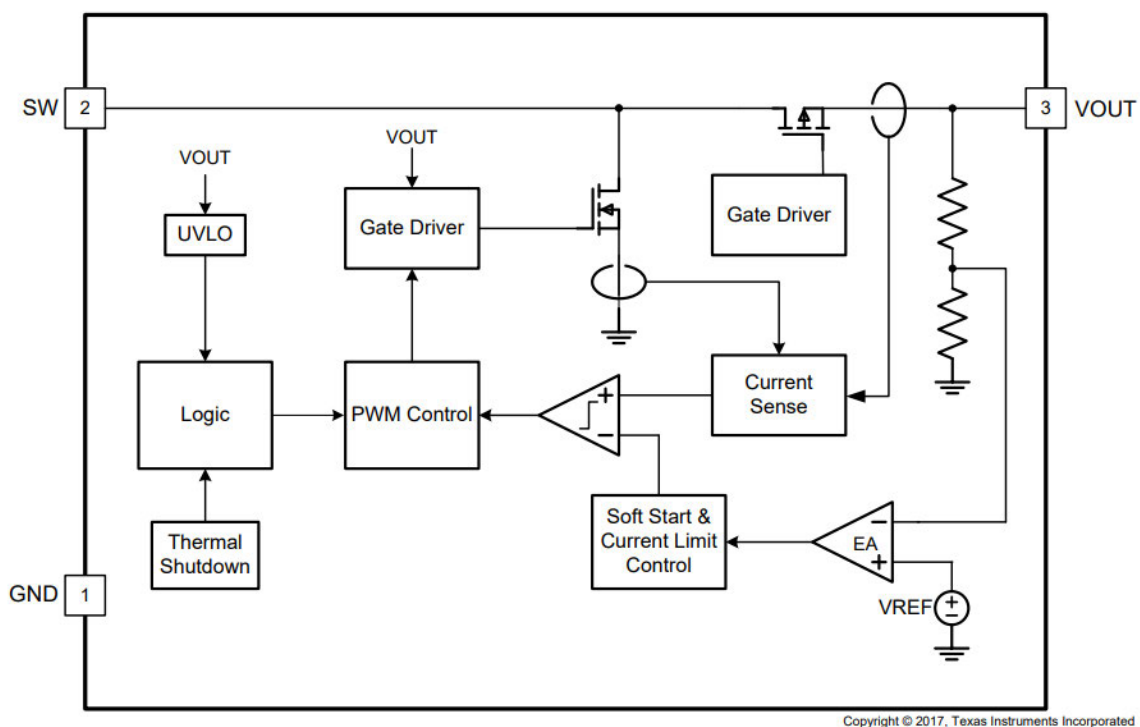


Abbildung 26: Tps613221 Interne Struktur[10]

Im Vergleich zur Grundtopologie in Abbildung 23 sind im tps613221 mehrere Rückkopplungsregelkreise und Schutzkreise hinzugefügt.

Die Softstart- und Strombegrenzungssteuerung wird mit der Stromerkennungsschaltung „Current Sense“ kombiniert. Dieses Modul ist dafür verantwortlich, die Ausgangsspannung während des Startvorgangs schrittweise zu erhöhen, um einen übermäßigen Strom beim Start zu verhindern. Es steuert auch die Strombegrenzung, um Überstromzustände zu verhindern.

Die Logik entscheidet anhand verschiedener Sensoren und Rückmeldungssignale, wie die Schaltknoten gesteuert werden.

EA (Error Amplifier) wird mit der Rückkopplungsspannung (VREF) verglichen, um das PWM-Signal anzupassen und die Ausgangsspannung stabil zu halten.

UVLO (Under Voltage Lock Out) verhindert den Betrieb des Wandlers bei zu niedriger Eingangsspannung und schützt den Schaltkreis vor dem Betrieb bei ungeeigneten Spannungen.

Die thermische Abschaltung wird verwendet, um Geräte bei Überhitzung abzuschalten, um Schäden zu vermeiden.

Die Induktivität ist die wichtigste Komponente im Design eines Leistungsreglers, da sie den stationären Betrieb, das Übergangsverhalten und die Schleifenstabilität beeinflusst. Der Induktor hat drei wichtige Parameter: Induktivitätswert, Sättigungsstrom und Gleichstromwiderstand (DCR). [10]

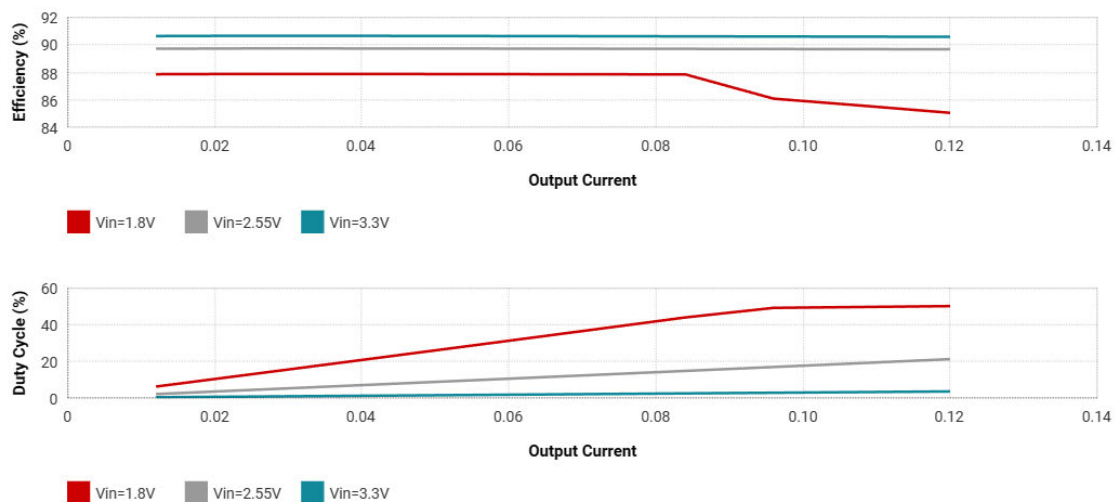


Abbildung 27: Benutzerdefinierte Daten[11]

Im Datenblatt lässt sich erkennen, dass der Ripple-Strom  $I_{LH}$  der Induktivität stabil bei 200 mA liegt. Wie in Abbildung 27 dargestellt, ergibt sich aus kundenspezifischen Daten, dass die Boost-Konversionseffizienz  $\eta$  von tps613221 bei der minimalen Batteriespannung von 1,8 V ungefähr 85 % beträgt, das Tastverhältnis bei 0,5 liegt und die Schaltfrequenz  $f_{SW}$  1,09 MHz beträgt. Zu diesem Zeitpunkt kann die Induktivität zu diesem Zeitpunkt gemäß der im Datenblatt angegebenen Berechnungsformel wie folgt berechnet werden.

$$L = \frac{V_{in} \times (V_{out} - V_{in} \times \eta)}{f_{SW} \times I_{LH} \times V_{out}} = 4.43 \mu H$$

Gleichzeitig muss der Sättigungsstrom des Induktors höher sein als der berechnete Spitzenstrom des Induktors, da sonst der übermäßige Spitzenstrom im Induktor das Gerät beschädigt und die Zuverlässigkeit des Systems verringert. Weil

$$\frac{V_{out} \times I_{out}}{V_{in} \times \eta} > \frac{I_{LH}}{2}$$

Daher arbeitet der tps613221 im CCM-Modus. Unter der Annahme, dass der Ausgangsstrom  $I_{out}$  den Maximalwert 0,12 A hat, beträgt der Spitzenstrom der Induktivität zu diesem Zeitpunkt

$$I_{L,peak} = \frac{V_{out} \times I_{out}}{V_{in} \times \eta} + \frac{I_{LH}}{2} = 359mA$$

Um einen Spielraum zu lassen, kann der minimale Sättigungsstrom auf 400 mA geregelt und die Induktivität anhand dieses Parameters ausgewählt werden.

Gleichzeitig verfügt der Induktor im eigentlichen Stromkreis über einen Gleichstromwiderstand (DCR), der zu Spannungsabfall und Leistungsverlust führt. Bei der Auswahl des Induktors sollte darauf geachtet werden, Induktorkomponenten mit einem möglichst kleinen DCR zu verwenden.

Der Ausgangskondensator  $C_{out}$  ist auch eine wichtige Komponente im Hochsetzsteller, die dazu dient, die Welligkeit der Ausgangsspannung auf ein akzeptables Maß zu begrenzen. Da im Datenblatt des ESP8266EX keine eindeutige Beschreibung der Spannungswelligkeit zu finden ist, geht die Berechnung von einem Grenzwert von 1 % der Versorgungsspannung aus. Nach der Formel im vorherigen Abschnitt.

$$\Delta V_o = \frac{\Delta Q}{C} = \frac{I_o DT_s}{C}$$

Und

$$C_{out} = \frac{I_o DT_s}{\Delta V_o} = 1.67\mu F$$

Andererseits haben Kondensatoren auch einen ESR-Reihenwiderstand, der tatsächlich zu einer Erhöhung der Spannungswelligkeit führt. Daher sollte man bei der tatsächlichen Auswahl eines Kondensators auch auf die Auswirkungen des ESR achten.

### 8.3 Simulation

Um eine Designverifizierung und Leistungstests von tps613221 durchzuführen, das offizielle Spice-Modell wurde heruntergeladen und die LTSpice-Software zur Schaltungssimulation wurde verwendet. Das Modell von tps613221 kann direkt über sein Spice-Modell generiert werden. Sein Simulationsschaltplan ist in Abbildung 28 dargestellt.

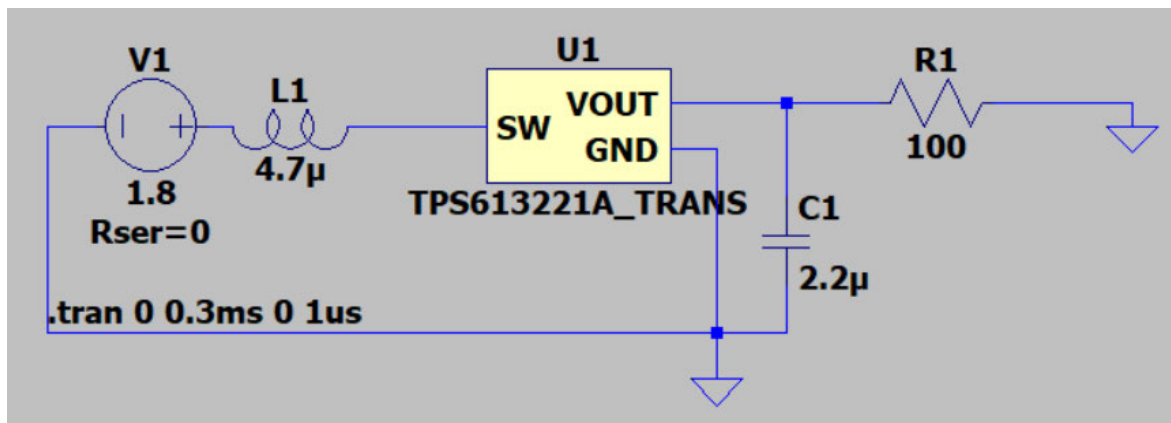


Abbildung 28:Tps613221 Test Schaltung

Da die Schaltfrequenz normalerweise bei etwa 1 MHz liegt, kann die Schrittzeit auf 1 μs eingestellt werden. Beim Ausführen der Simulation trat jedoch der in Abbildung 29 gezeigte Fehler auf.

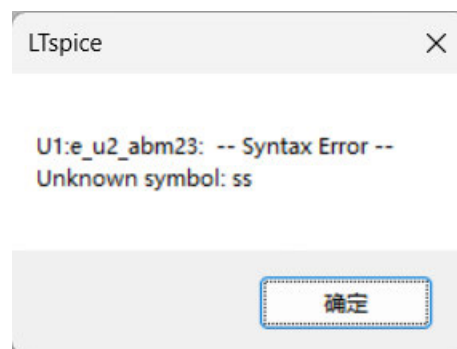


Abbildung 29:Fehler

Dies kann durch einen Syntaxfehler innerhalb des Spice-Modells oder der offiziellen Verschlüsselung des Modells verursacht werden. Nachdem keine Lösung für das Problem gefunden wurde und aufgrund der Tatsache, dass die Testzyklen für die praktische Schaltungstests zu lang sind, wird vorerst kein TPS613221-Boost-Schaltkreis in die tatsächliche Schaltung integriert. Gleichzeitig führt der Einsatz von Boost-Schaltungen zu höheren Kosten und Schwierigkeiten beim Löten und ist nicht vorteilhaft für das Layout der PCB (bei der Gestaltung muss darauf geachtet werden, dass das Erscheinungsbild nicht beeinträchtigt wird). Daher ist das Vorhandensein einer Boost-Schaltung nicht notwendig, solange die Akkulaufzeit noch akzeptabel ist.

## 9. Schaltung und PCB Entwurf

Bei einer Lötübungsarbeit bestimmt das Design der Leiterplatten- und Layoutteile das Aussehen und die Lötmethodologie der Arbeit. In diesem Kapitel werden die Zusammensetzung, das Erscheinungsbild und das Layoutdesign der Schaltung erläutert und die Bauteilgehäuse einiger Komponenten angepasst, um die Schwierigkeit beim Löteten zu verringern.

### 9.1 Schaltung

Nach der Bestimmung der in der Schaltung verwendeten Komponenten ist die endgültige verwendete Schaltung in Abbildung 30 dargestellt (BAT1 ist Batteriehalter).

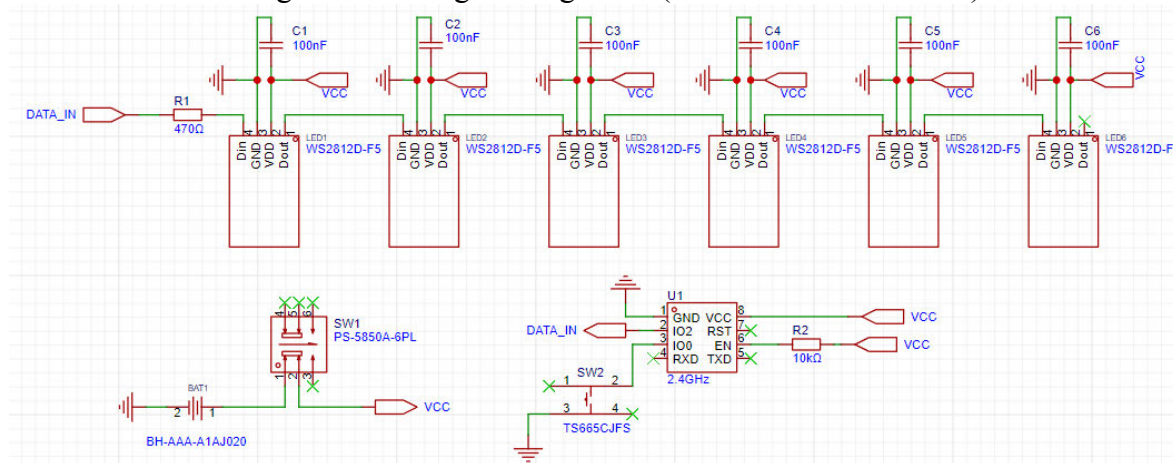


Abbildung 30: Schaltung

Im Stromversorgungsteil wird kein Aufwärtswandler verwendet, was den Schaltungsaufbau vereinfacht. Er besteht aus drei Teilen: LED-Lichtgruppenschaltung, Batterieschaltung und Mikrocontroller-Schaltung.

Im Abschnitt der LED-Lichtgruppenschaltung wird der Widerstand R1 eingesetzt, um den Strom bei DATA\_IN zu begrenzen, was dazu beiträgt, Signalreflexionen zu reduzieren und die I/O-Pins des Mikrocontrollers vor Beschädigungen zu schützen. Dieser Widerstand spielt auch eine schützende Rolle und hilft dabei, die Signalintegrität zu bewahren, insbesondere wenn DATA\_IN über größere Entfernungen übertragen wird oder bei höheren Datenübertragungsraten.

Im Experiment wurde der Widerstand R1 entfernt, und es stellte sich heraus, dass dies keine signifikanten Auswirkungen auf die Funktionsweise des WS2812D hatte. Daher könnte R1 aus Kostengründen in diesem Design weggelassen werden. Die Kondensatoren C1 bis C6 dienen als Entkopplungskondensatoren, um die Stromversorgung zu glätten und Störungen sowie Spitzen auf der Stromleitung zu absorbieren, die durch das Schalten der LEDs verursacht werden. Da die Stromversorgung des Batteriekreises in der aktuellen Schaltung sehr stabil ist, könnten diese Entkopplungskondensatoren ebenfalls entfernt werden. Aus Gründen der Verbesserung der Schaltungsrobustheit werden jedoch der Widerstand R1 und die Entkopplungskondensatoren C1 bis C6 vorläufig beibehalten.

Im Batterieschaltkreis wird die Batterie über einen selbstsperrenden Schalter (SW1) mit VCC verbunden, um den Mikrocontroller und die LEDs mit Strom zu versorgen.

Im Teil des Mikrocontroller-Schaltkreises steuert der ESP8266EX das Ein- und Ausschalten der WLAN-Funktion, indem er die Spannung am GPIO0-Port erkennt und Datensignale über den GPIO2-Port an den WS2812D sendet. Der Widerstand R2 dient als Pull-up-Widerstand, um den EN-Pin auf High-Niveau zu halten. Da der EN-Pin des Chips keine externe Steuersignale benötigt, sondern lediglich zur Aufrechterhaltung des aktiven Zustands des Chips dient, ist es praktikabler, den EN-Pin direkt mit VCC zu verbinden, wodurch R2 in der tatsächlichen Anwendung entfernt werden kann.

Die Anzahl der LEDs verdient ebenfalls Beachtung. Da aktuell Batterien für die direkte Stromversorgung verwendet werden und der maximale Stromverbrauch einer einzelnen LED nur 10 mA beträgt, kann der Stromkreis mehr LEDs steuern. Wird jedoch ein Aufwärtswandler für die Stromversorgung genutzt und die Batteriespannung ist niedrig, kann es zu erhöhter Wärmeentwicklung kommen. Dies liegt an der sinkenden Effizienz des Aufwärtswandlers und dem steigenden Innenwiderstand der Batterie. Daher wird abgeraten, zu viele LEDs zu nutzen, wenn ein Aufwärtswandler eingesetzt wird. In der aktuellen Schaltung werden nur 6 LEDs verwendet. Dies beruht auf der Absicht, die Batterielebensdauer zu maximieren und der begrenzten Leiterplattenfläche des Anhängers. Zugleich ermöglicht die Nutzung von genau 6 LEDs die Implementierung einer elektronischen Würfelfunktion durch Programmierung.

## 9.2 PCB Entwurf

Nach der Festlegung der Schaltung sind ein ansprechendes Erscheinungsbild und ein vernünftiges Layout von großer Bedeutung. Obwohl die Freiheit im Design des Erscheinungsbildes recht groß ist, muss zumindest sichergestellt werden, dass das Layout genügend Raum bietet. Als Referenz für das Design wurde ein originelles Muster eines Roboterkopfes verwendet, wie in Abbildung 31 dargestellt. Im Vergleich zu einigen unregelmäßigen Mustern hat dieses Muster eine größere Gesamtfläche für das Layout, und die Darstellung des Roboters passt gut zu dem Eindruck, der durch Elemente wie elektronisches Löten, Fernsteuerung und RGB-LEDs vermittelt wird.

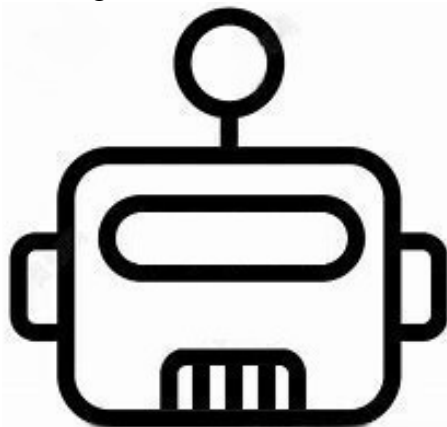


Abbildung 31: Muster

Bevor mit dem Layout fortgefahren wurde, waren einige Anpassungen am Gehäuse dieses Bauteils zu erfolgen. Das offizielle Gehäuse des WS2812D weist das Problem auf, dass der Abstand zwischen den Pads zu gering ist, was das Löten erschwert.

Um sicherzustellen, dass sich der WS2812D in der richtigen Position im Anhänger befindet und die Größe des Anhängers angemessen ist, wurden die Pad-Abstände nicht angepasst. Stattdessen wurde die Form der Pads auf längliche Kreise eingestellt, und sie wurden gegeneinander versetzt, um das Problem zu vermeiden, dass die Löt pads beim Löten leicht miteinander verbunden werden können. Gleichzeitig müssen gemäß dem aktuellen Modell des WS2812D die Positionen der Pins beim Löten markiert werden (Pin 3 dieses Modells hat die längste Länge, andere Modelle werden anders sein). Das Bauteilgehäuse vor und nach der Änderung ist in Abbildung 32 dargestellt.

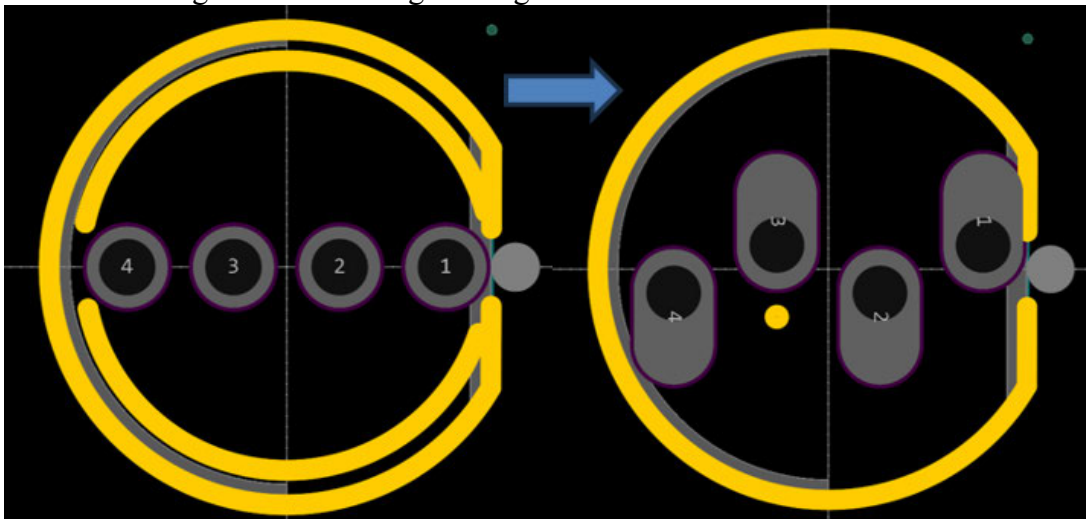


Abbildung 32: Bauteilgehäuse verbessern

Um gleichzeitig die Integrität des Frontmusters der Leiterplatte sicherzustellen, sind alle Widerstände und Kondensatoren in dieser Schaltung SMD-Komponenten (diese Komponenten können vorgelötet werden). Und die MCU, der Batteriehalter, die Schalter und die LEDs sind ebenfalls vorhanden. Alle Komponenten befinden sich an bestimmten Positionen und der Layoutteil befindet sich im Grunde alles auf der Rückseite. Gleichzeitig muss als Anhänger auch darauf geachtet werden, dass das Gewicht auf beiden Seiten ausgeglichen ist. Das endgültige Layout ist in Abbildung 33 dargestellt.

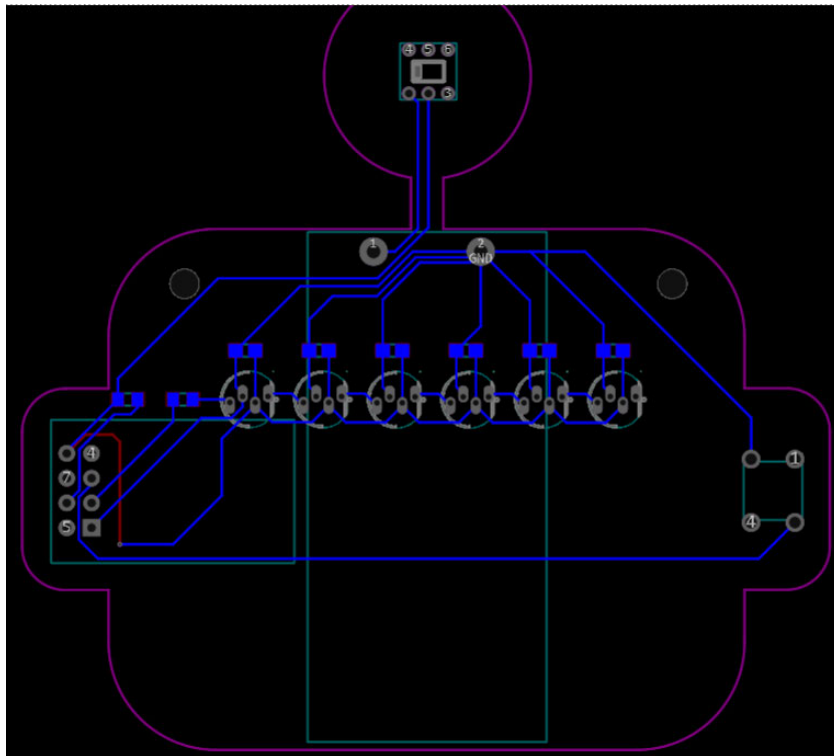


Abbildung 33:Layout

Darüber hinaus sollte man beim Verlegen auf die Drahtbreite achten. Im Allgemeinen muss die Drahtbreite von Stromkabeln und Erdungskabeln breiter sein. Nach der Berechnung kann die minimale Drahtbreite bestätigt werden[12]. Da der durch die Drähte fließende Strom nicht groß ist, wird die Drahtbreite schließlich auf 0,254 mm eingestellt. Auf beiden Seiten der Platine sind Bohrlöcher mit einem Radius von 1,5 mm vorgesehen, damit das Trageband des Anhängers hindurchgeführt werden kann. Da sich das schwerste Batterieteil in der Mitte des Anhängers befindet, liegt der Schwerpunkt des Anhängers in der Mitte, wodurch der Anhänger beim Tragen ausreichend ausbalanciert ist. Nach Fertigstellung des Layouts kann man der Siebdruckebene auch einige Notizen hinzufügen, um Benutzer an die Verwendung zu erinnern.

Vor der Herstellung der Leiterplatte kann der Anhänger auch durch verschiedene Leiterplattenfarben variiert werden, wie in Abbildung 34 dargestellt.



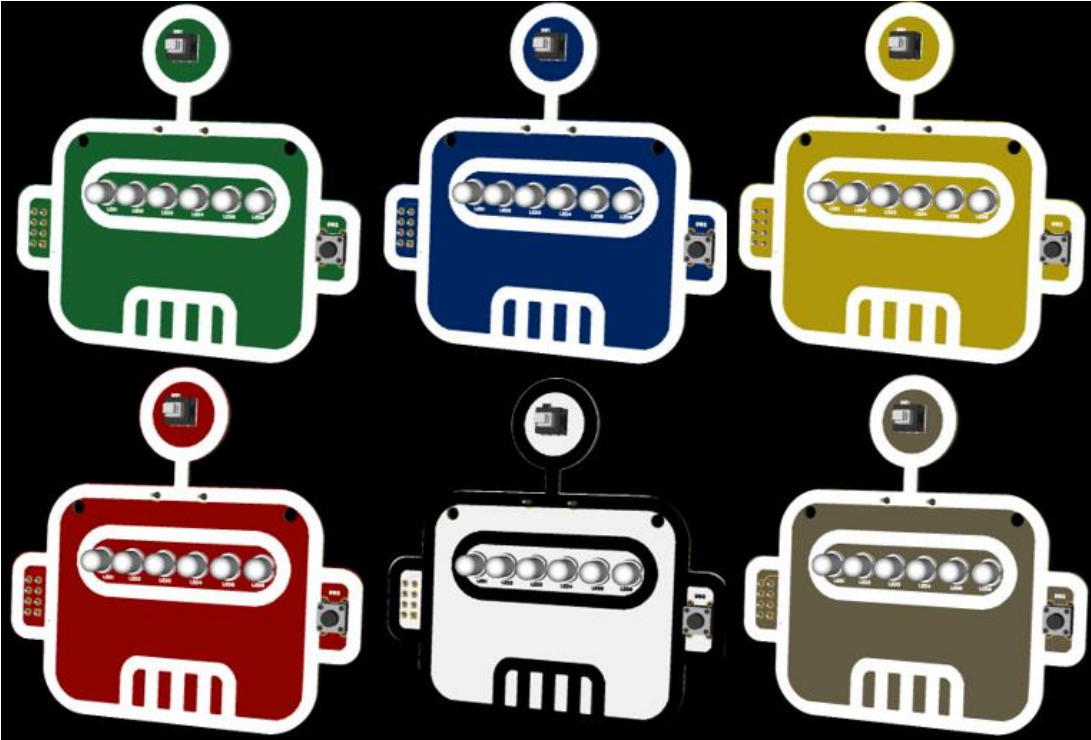


Abbildung 34:Modelle in verschiedenen Farben

## 10. Zusammenfassung

### 10.1 Kostenabschätzung

Nachdem die Schaltung und die Leiterplatte festgelegt sind, werden auch die Gesamtkosten eines solchen Lötübungsprojekts ermittelt. Die in der Schaltung verwendeten Komponenten und deren einzelnen Kosten sind in Tabelle 3 aufgeführt (basierend auf dem Preis beim Kauf von 100 Exemplaren).

Designator	Menge	Herstellerteil	Hersteller	Wert	Preis(€)
BAT1	1	BH-AAA-A1AJ020	MYOUNG	–	0.2102
C1-C6	6	CC0805KRX7R9BB104	YAGEO	100nF	0.0042
LED1-LED6	6	WS2812D	Guangze	–	0.0242
R1	1	0805W8F4700T5E	UNI-ROYAL	470Ω	0.0017
R2	1	0805W8F1002T5E	UNI-ROYAL	10kΩ	0.0013
SW1	1	PS-5850A-6PL	G-Switch	–	0.0902
SW2	1	TS665CJFS	SHOU HAN	–	0.0470
U1	1	ESP-01S	Ai-Thinker	2.4GHz	1.4824

Tabelle 3 Kostenabschätzung

Alle Komponenten außer LEDs können unter [13] erworben werden. Und die LEDs können unter [14] erworben werden. Die Gesamtkosten für die fertigen Komponenten betragen ca. 2,00 Euro. Zusammen mit den Leiterplattenkosten belaufen sich die Gesamtkosten für eine einzelne RGB- Anhänger auf ca. 2,38 Euro (ohne Versandkosten). Basierend auf dem Preis jedes Produkts in Tabelle 1 wird der RGB-Anhänger sehr wettbewerbsfähig sein.

### 10.2 Ergebnis

Abschließend kann als Ergebnisbewertung anhand der in Kapitel 1 und Kapitel 2 definierten Designanforderungen erfolgen.

Da während eines zehnstündigen Tests keine offensichtliche Überhitzung der Komponenten festgestellt wurde und die Betriebsspannung sowie der Stromverbrauch des Schaltkreises niedrig sind, ist er bei normaler Nutzung sicher.

Bezüglich des Gewichts des Anhängers ergab die Messung ein Gewicht von 45g, wobei die Batterie 28g wiegt. Eine weitere Gewichtsreduktion könnte durch die Verwendung eines Boost-Konverters erreicht werden, um eine einzelne AAA-Batterie auf 3.3V zu erhöhen. Allerdings würde die Effizienz des Boost-Konverters bei niedriger Batteriespannung sinken und der Innenwiderstand der Batterie steigen, was zu einer deutlicheren Erwärmung und damit zu einer erhöhten Gefahr führen könnte.

Da der Schaltkreis einfach und die Anzahl der verwendeten Komponenten gering ist und die SMD-Komponenten vorab gelötet wurden, verringert sich die Lötzeit. Außerdem wurde das Gehäuse des schwer zu lötenen Bauteils WS2812D geändert, was das Löten erleichtert.

In Bezug auf Energieeffizienz hat diese Arbeit eine andere als die offizielle Methode verwendet, um das HF-Modul zu schalten, wodurch der Stromverbrauch des ESP8266EX von 70mA auf 15mA reduziert wurde. Zudem wurden weitere Methoden zur Energieeinsparung analysiert, um den Energieverbrauch so weit wie möglich zu senken und die Nutzungsdauer des Anhängers zu verlängern.

In Bezug auf den Spielwert kann durch den Einsatz von RGB-LEDs, die  $2^{24}$  Farben pro LED anzeigen können, und die Kombination mehrerer LEDs eine Vielzahl von Farbwechseln realisiert werden, was ein hohes Spielspotenzial bietet. Zusätzlich hat diese Arbeit verschiedene Farbmodi und Steuerungen implementiert, deren Code im Anhang zu finden ist.

In Bezug auf die Benutzerfreundlichkeit der Steuerung kann der Nutzer nur eine Webseite benutzen, nachdem er sich mit dem Netzwerk verbunden hat, ohne eine App herunterzuladen. Dies ist eine sehr intuitive und einfache Steuerungsmethode, die sogar Kinder leicht bedienen können.

Zu Beginn des Designs war das Gesamtkostenziel unter 5 Euro festgelegt, aber tatsächlich wird die Gesamtkosten auf nur 2,38 Euro reduziert. Dies ist ein sehr gutes Ergebnis, das hauptsächlich auf die Analyse in Kapitel 2 über die MCU-Auswahl zurückzuführen ist. Obwohl Bluetooth dieselbe Steuerungsfunktionalität bieten könnte, würde allein ein Bluetooth-Chip mehr als 5 Euro kosten. Dies zeigt, dass die Auswahl der richtigen Steuerungsmethode je nach Entwicklungsanforderungen notwendig ist.

Auch das Aussehen des Anhängers ist ein wichtiger Attraktionspunkt. Im Designprozess hat diese Arbeit ein Muster verwendet, das zum Thema "Lötübung" passt, und versucht, die Integrität des Musters bei der Layoutgestaltung so weit wie möglich zu bewahren.

Abschließend könnte innerhalb des Kostenrahmens auch ein geeignetes Gehäuse für den Anhänger entworfen werden, um den Tragekomfort und die ästhetische Wirkung zu verbessern.

## Literatur

- [1] RF430FRL15xH NFC ISO 15693 Sensor Transponder.  
URL: <https://www.ti.com/product/RF430FRL152H>
- [2] Wikipedia: Bluetooth, Dezember 2023.  
URL: <https://en.wikipedia.org/wiki/Bluetooth>
- [3] WP154A4SUREQBFZGW T-1 3/4 (5mm) Full Color LED Lamp. URL:  
<https://www.kingbrightusa.com/images/catalog/SPEC/WP154A4SUREQBFZGW.pdf>
- [4] WS2812D-F5-1261 Intelligent controlled integrated light source.  
URL: <https://file.elecfans.com/web2/M00/49/F1/poYBAGKjK4aAOuxhAAT5Z-gqaRw441.pdf>
- [5] ESP8266EX Datasheet.  
URL: [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)
- [6] White Paper SWRA349.  
URL: <https://www.dmcinfo.com/Portals/0/Blog%20Files/Coin%20cells%20and%20peak%20current%20draw.pdf>
- [7] Designtipps für den knopf-batteriebetriebenen CC2640.  
URL: <https://www.ti.com.cn/cn/lit/an/zhca744/zhca744.pdf>
- [8] MN2400 Size: AAA (LR03) Alkaline-Manganese Dioxide Battery.  
URL: <https://www.duracell.com/wp-content/uploads/2020/02/MN24US1119.pdf>
- [9] Mohan, Ned; Undeland, Tore M.; Robbins, William P. (2003) Power Electronics - Converters, Applications, and Design. 3. Auflage. Hoboken, NJ: John Wiley & Sons, Inc.
- [10] TPS61322 6.5- $\mu$ A Quiescent current, 1.8-A switch current boost converter.  
URL: [https://www.ti.com/lit/ds/sym-link/tps61322.pdf?ts=1707204536596&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS61322](https://www.ti.com/lit/ds/sym-link/tps61322.pdf?ts=1707204536596&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS61322)
- [11] URL: [Power Designer \(ti.com\)](https://www.ti.com/power-designer)
- [12] URL: <https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-pcb-trace-width>
- [13] URL: [www.lcsc.com](https://www.lcsc.com)
- [14] URL: <https://item.taobao.com/item.htm?u=f2tcnc5a50fa&id=723529883280&spm=a1z09.2.0.0.19fc2e8dT36ZYd>

# Anhang

## Quelle Code

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <FastLED.h>
#include <EEPROM.h>

// AP-Einstellungen
const char *ssid = "RGBRobot-Setup";
const char *password = "";

// Webserver-Port
ESP8266WebServer server(80);

// FastLED-Einstellungen
#define NUM_LEDS 6
#define buttonPin 0 // GPIO-Pin für externen Taster einstellen
#define DATA_PIN 2 // GPIO-Pin für Data-Pin einstellen
CRGB leds[NUM_LEDS];
bool apModeActive = false;
bool buttonPressed = false;
int colorIndex;
int brightness;

// Voreingestellte Farben
CRGB presetColors[] = {
    CRGB::Red,
    CRGB::Green,
    CRGB::Blue,
    CRGB::Yellow,
    CRGB::Purple,
    CRGB::White
};

//HTML und CSS Entwurf
const char welcomePage[] = R"rawliteral(
<!DOCTYPE HTML>
<html>
<head>
<title>RGB ROBOT HEAD</title>
<style>
  body {
    font-famil y: sans-serif;
    color: #444;
    background-color: #ffecb3;

```

```
    margin: 40px;
    text-align: center;
}
h1 {
    color: #333;
}
.effect-category {
    background: #ffffcf2;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    margin-top: 20px;
}
input[type=range] {
    width: 100%;
}
input[type=submit] {
    background-color: #ff9a8c;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}
input[type=submit]:hover {
    background-color: #ff766a;
}
a {
    display: block;
    margin: 10px 0;
    text-decoration: none;
    color: #007BFF;
}
</style>
</head>
<body>
    <h1>RGB ROBOT HEAD!</h1>

    <!-- Kategorie 1: LightingMode -->
    <div class="effect-category">
        <h2>LightingMode</h2>
        <form action="/lightingmode" method="get">
            Brightness: <input type="range" name="brightness" min="0"
max="128"><br>
            Color Preset:
            <select name="colorpreset">
                <option value="0">Red</option>
                <option value="1">Green</option>
                <option value="2">Blue</option>
```

```

        <option value="3">Yellow</option>
        <option value="4">Purple</option>
        <option value="5">White</option>
</select>
    <br>
    <input type="submit" value="LightingMode">
</form>
</div>

<!-- Kategorie 2: Warning Effect、Thinking Effect、Blue Stars Effect -->
<div class="effect-category">
    <h2>Robot Emotion</h2>
    <a href="/warning">Warning</a><br>
    <a href="/thinking">Thinking</a><br>
    <a href="/bluestars">Blue Stars</a><br>
</div>

<!-- Kategorie 3: Rainbow Flow Effect 和 Rainbow Thought Effect -->
<div class="effect-category">
    <h2>Rainbow Time</h2>
    <a href="/rainbowflow">Rainbow Flow</a><br>
    <a href="/rainbowthought">Rainbow Thought</a><br>
</div>

</body>
</html>
)rawliteral";

enum Effect { None, Warning, Thinking, RainbowFlow, BlueStars, Rain-
bowThought } currentEffect;

void handleRoot() {
    server.send_P(200, "text/html", welcomePage);
}

void handleLightingModeEffect() {
    currentEffect = None;
    FastLED.clear(); // LED-Farbe löschen
    FastLED.show();
    delay(1000);

    // Überprüfen, ob die Parameter brightness und colorpreset richtig
    vorhanden sind
    if (server.hasArg("brightness") && server.hasArg("colorpreset")) {
        int brightnessValue = server.arg("brightness").toInt();
        brightness = brightnessValue;

        String colorPresetStr = server.arg("colorpreset");
        colorIndex = colorPresetStr.toInt();
    }
}

```

```
    if (colorIndex >= 0 && colorIndex < sizeof(presetColors) / sizeof(presetColors[0])) {
        FastLED.setBrightness(brightness);
        fill_solid(leds, NUM_LEDS, presetColors[colorIndex]);
        FastLED.show();
        server.send(200, "text/html", welcomePage);
    } else {
        server.send(400, "text/plain", "Invalid input");
    }
} else {
    server.send(400, "text/plain", "Missing parameters");
}
}
```

```
void warningEffect() {
    for (int i = 0; i < NUM_LEDS; i++) {
        leds[i] = CRGB::Red; // LED auf Rot setzen
    }
    FastLED.show();
    delay(500); // Blinkintervall
    FastLED.clear();
    FastLED.show();
    delay(500);
}
```

```
void thinkingEffect() {
    for (int i = 1; i < NUM_LEDS - 1; i++) {
        leds[i] = CRGB::Yellow;
        FastLED.show();
        delay(100); // Dauer
        leds[i] = CRGB::Black; // LED-Farbe löschen
        FastLED.show();
    }
    leds[NUM_LEDS - 1] = CRGB::Yellow;
    FastLED.show();
    delay(1500); // Dauer
    leds[NUM_LEDS - 1] = CRGB::Black;
    FastLED.show();
```

```
    for (int i = NUM_LEDS - 2; i >= 1; i--) {
        leds[i] = CRGB::Yellow;
        FastLED.show();
        delay(100);
        leds[i] = CRGB::Black;
        FastLED.show();
    }
    leds[0] = CRGB::Yellow;
    FastLED.show();
```



```
    delay(1500);
    leds[0] = CRGB::Black;
    FastLED.show();
}

void rainbowFlowEffect() {
    static byte hue = 0;
    fill_rainbow(leds, NUM_LEDS, hue, 7);
    FastLED.show();
    hue += 2;
    delay(5);
}

void blueStarsEffect() {
    for (int i = 0; i < NUM_LEDS; i++) {
        leds[i] = CHSV(170, 200, random8(3) * 50);
    }
    FastLED.show();
    delay(350);
}

void rainbowThoughtEffect() {
    static byte hue = 0;
    for (int i = 1; i < NUM_LEDS - 1; i++) {
        leds[i] = CHSV(hue, 255, 255);
        FastLED.show();
        delay(50);
        leds[i] = CRGB::Black;
        FastLED.show();
        hue += 21;
    }
    leds[NUM_LEDS - 1] = CHSV(hue, 255, 255);
    FastLED.show();
    delay(200);
    leds[NUM_LEDS - 1] = CRGB::Black;
    FastLED.show();

    for (int i = NUM_LEDS - 2; i >= 1; i--) {
        leds[i] = CHSV(hue, 255, 255);
        FastLED.show();
        delay(50);
        leds[i] = CRGB::Black;
        FastLED.show();
        hue += 21;
    }
    leds[0] = CHSV(hue, 255, 255);
    FastLED.show();
    delay(200);
    leds[0] = CRGB::Black;
    FastLED.show();
}
```

```
}

void handleWarningEffect() {
    currentEffect = Warning;
    server.send(200, "text/html", welcomePage);
}

void handleThinkingEffect() {
    currentEffect = Thinking;
    server.send(200, "text/html", welcomePage);
}

void handleRainbowFlowEffect() {
    currentEffect = RainbowFlow;
    server.send(200, "text/html", welcomePage);
}

void handleBlueStarsEffect() {
    currentEffect = BlueStars;
    server.send(200, "text/html", welcomePage);
}

void handleRainbowThoughtEffect() {
    currentEffect = RainbowThought;
    server.send(200, "text/html", welcomePage);
}

void setup() {
    EEPROM.begin(3); // EEPROM initialisieren, nur 3 Byte verwenden
    Serial.begin(115200);
    pinMode(buttonPin, INPUT_PULLUP); // Internen Pull-up-Widerstand verwenden
    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);

    thinkingEffect();
    currentEffect = static_cast<Effect>(EEPROM.read(0)); // Annahme, dass currentEffect in Adresse 0 gespeichert ist
    colorIndex = EEPROM.read(1); // Farbindex lesen
    brightness = EEPROM.read(2); // Helligkeit lesen

    server.on("/", handleRoot);
    server.on("/lightingmode", handleLightingModeEffect);
    server.on("/warning", handleWarningEffect);
    server.on("/thinking", handleThinkingEffect);
    server.on("/rainbowflow", handleRainbowFlowEffect);
    server.on("/bluestars", handleBlueStarsEffect);
    server.on("/rainbowthought", handleRainbowThoughtEffect);
}
```

```
    server.begin();
}

void loop() {
    // Externen Tasterstatus lesen
    int buttonState = digitalRead(buttonPin);

    if (buttonState == LOW && !buttonPressed) { // Taste wurde gedrückt und
war vorher nicht gedrückt
        Serial.println("Button pressed"); // Serielle Ausgabe: Taste gedrückt
        buttonPressed = true;
        delay(50); // Einfache Anti-Shake-Verzögerung

        // AP-Modus ein- und ausschalten
        if (!apModeActive) { // AP-Modus aktivieren, wenn er nicht aktiv ist
            thinkingEffect();
            Serial.println("Turning AP mode ON"); // Serielle Ausgabe: AP-Modus
einschalten
            WiFi.mode(WIFI_AP); // In den AP-Modus wechseln
            WiFi.softAP(ssid, password);
            apModeActive = true;
            delay(1000); // Serielle Ausgabe: AP-Modus ausschalten
        } else { // Taste wurde nicht gedrückt
            Serial.println("Turning AP mode OFF"); // Serielle Ausgabe: Taste
losgelassen
            apModeActive = false;
            delay(1000); // Mehrfachreaktionen in kurzer Zeit vermeiden
            EEPROM.write(0, static_cast<uint8_t>(currentEffect));
            if(currentEffect == None){
                EEPROM.write(1, static_cast<uint8_t>(colorIndex));
                EEPROM.write(2, static_cast<uint8_t>(brightness));
            }
            EEPROM.commit();
            ESP.restart();
        }
    } else if (buttonState == HIGH && buttonPressed) { // Taste nicht gedrückt
        Serial.println("Button released"); // Taste wird losgelassen
        buttonPressed = false;
        delay(50); // Einfache Anti-Shake-Verzögerung
    }

    // Die CPU führt Aufgaben im Energiesparmodus aus
    if (apModeActive) {
        server.handleClient();
    }

    switch (currentEffect) {
        case Warning:
            warningEffect();
    }
}
```

```
        break;
    case Thinking:
        thinkingEffect();
        break;
    case RainbowFlow:
        rainbowFlowEffect();
        break;
    case BlueStars:
        blueStarsEffect();
        break;
    case RainbowThought:
        rainbowThoughtEffect();
        break;
    default:
        break;
}
}

if (!lapModeActive) {
    switch (currentEffect) {
    case None:
        FastLED.setBrightness(brightness);
        fill_solid(leds, NUM_LEDS, presetColors[colorIndex]);
        FastLED.show();
        delay(5000); // um Mehrfachantworten in kurzer Zeit zu vermeiden
        break;
    case Warning:
        warningEffect();
        break;
    case Thinking:
        thinkingEffect();
        break;
    case RainbowFlow:
        rainbowFlowEffect();
        break;
    case BlueStars:
        blueStarsEffect();
        break;
    case RainbowThought:
        rainbowThoughtEffect();
        break;
    default:
        break;
    }
}
}
```

## Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Chemnitz, 15.02.2024