# BLE HID Hardware-Erweiterungsmodul für Drohnenfernbedienungen

**Studienarbeit**

des Studiengangs IT-Automotive

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

**Fabian Kuffer**

14. November 2022

| | |
|---|---|
| **Bearbeitungszeitraum** | 4. Oktober 2022 - 8. Juni 2023 |
| **Matrikelnummer, Kurs** | 2044882, TINF-20ITA |
| **Betreuer** | Prof. Dr. Karl Friedrich Gebhardt |

# Erklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema: *BLE HID Hardware-Erweiterungsmodul für Drohnenfernbedienungen* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Stuttgart, 14. November 2022

---

Fabian Kuffer

## Kurzfassung

Kurzfassung

## Abstract

Abstract

# Inhaltsverzeichnis

# Abkürzungsverzeichnis

| | |
|---|---|
| **ATT** | Attribute Protocol |
| **BBR** | Bluetooth Basic Rate |
| **BLE** | Bluetooth Low Energy |
| **CID** | Kanalidentifizierer |
| **GAP** | Generic Access Profile |
| **GATT** | Generic Attribute Pribute Profile |
| **HCI** | Host Controller Interface |
| **HID** | Human Interface Device |
| **HOGP** | HID over GATT Profile |
| **ISM** | Industrial, Scientific and Medical |
| **L2CAP** | Logical Link Control and Adaption Protocol |
| **LL** | Link Layer |
| **PHY** | Physical Layer |
| **PPM** | Puls-Positions-Modulation |
| **SDP** | Service Discovery Protocol |
| **SIG** | Special Interest Group |
| **SMP** | Security Manager Protocol |
| **UUID** | universal unique identifier |

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Quellcodeverzeichnis

# 1 Einleitung

Zwei arten von Drohnen. Freestyle/Renndrohen ... und Consumerdrohnen welche viele Sensoren haben und einfach zu fliegen sind. Renndrohnen sind im Acro-Modus komplex zu fliegen, da dort viel gesteuert werden muss. Es wird Training benötigt. Entweder im Freien oder in Simulatoren, um weniger zu zerstören.

Neben dem eigentlichen Multicopterfliegen stellt für Renn- und Freestyle-Multicopterpiloten das Training einen wichtigen Bestandteil dar. Dieses kann in zwei Varianten durchgeführt werden. Der Multicopterpilot trainiert entweder am Flugplatz. Hier können aber durch Abstürze hohe Reparaturkosten und lange Reparaturzeiten entstehen. Oder der Multicoperpilot trainiert im Simulator am Rechner. Damit die gewohnte Fernbedienung ebenfalls am Rechner verwendet werden kann, bieten einige Hersteller die Möglichkeit an, die Fernbedienung als USB-HID-Joystick zu verwenden. Durch die immer leistungsfähiger werdenden Smartphones und Tablets wäre es wünschenswert, auf mobilen Geräten Simulatoren für das Training zu verwenden. Das Problem hierbei ist jedoch, dass die Verbindung der Multicopterfernbedienung mit dem mobilen Gerät über USB nur eingeschränkt beziehungsweise unmöglich ist. Beseitigt werden kann dieses Problem bei einigen Fernbedienungen mit Modulschächten, mit Hilfe derer die Tasten- und Joysticksignale über andere Funkstandards übertragen werden können. Ziel der Arbeit ist es, ein Hardware-Erweiterungsmodul für Multicopterfernbedienungen zu entwickeln. Vorausgesetzt wird im Rahmen dieser Arbeit, dass die Fernbedienungen einen Modulschacht aufweisen und die Firmware OpenTX beziehungsweise eine Abspaltung davon verfügbar ist. Das Erweiterungsmodul soll sich dabei durch BLE als HID-Gerät an Endgeräten authentifizieren, wodurch die Multicopterfernbedienung als kabelloser Joystick an Endgeräten verwendet werden kann. Weitere zusätzliche Optionen – sofern zeitlich machbar – sind zum einen, ein kleines LED-Display einzubauen, womit die Bedienung des Moduls erleichtert werden kann. Zum anderen eine GUI zu entwickeln, um den Updateprozess für das BLE-HID-Modul zu vereinfachen. Die GUI kann dafür mit dem Framework Electron für eine systemunabhängige Verwendung entwickelt werden.

## 1.1 Motivation

In den letzten Jahren ist die Leistungsfähigkeit von Tablets gestiegen. Jedoch ist es schwer mittels USB eine Verbindung aufzubauen. Dafür muss auf ein Funkstandard ausgewichen werden –> Bluetooth. Es wird dann genau BLE verwendet, da dieses unter Apple ohne Einschränkungen verwendet werden kann. Dadurch findet eine Ausweitung für Simulatoren auf mobile Geräte statt, da es zurzeit die Kommunikation mit Tablets schwer ist. –> Schreiben, dass es mittels einem Modul an Controllern gelöst werden soll.

## 1.2 Stand der Technik

Gibt im Umfeld nur wenig bis keine BLE HID Geräte zum Verbinden mit Smartphone. Eine Möglichkeit via USB und via Betaflight-Flightcontroller.

Schreiben, was es für andere Module statt ESP gibt. Der Modulschacht wird zurzeit nur für andere Übertragungsstandards für Drohnen verwendet.

Bilder bis jetzt erstellen

Alles bis technische Grundlagen einmal schreiben

# 2 Aufgabenstellung

## 2.1 Softwareentwicklung

## 2.2 Platinendesign

## 2.3 Gehäuseerstellung

# 3 Technische Grundlagen

## 3.1 Human Interface Device (HID)

### 3.1.1 Allgemein

### 3.1.2 Report Deskriptor

## 3.2 Bluetooth

### 3.2.1 Allgemein

Bluetooth ist ein Kurzstreckenkommunikationssystem, bei welchen die Hauptmerkmale auf Robustheit, einen geringen Stromverbrauch und geringe Kosten gelegt wurde. Bluetooth wir in zwei Kategorien aufgeteilt. Die erste Kategorie ist Bluetooth Basic Rate (BBR). Die zweite Kategorie ist BLE. Beide Kategorien beinhalten dabei Mechanismen, um Bluetooth-Geräte zu entdecken, einen Verbindungsaufbau durchzuführen sowie eine Verbindung herzustellen. Das Augenmerk bei BLE Produkten liegt dabei auf einen niedrigen Stromverbrauch, welche durch eine geringere Datenrate und eine geringere Einschaltdauer während den Datenaustausch als bei BBR realisiert wird. Die Übertragungsrate bei BLE in der physikalischen Schicht beträgt 2 MB/s. Zu beachten ist, dass ein Bluetooth-Controller entweder nur BLE, BBR oder beide Bluetooth-Kategorien unterstützen kann. [1, S. 187]

Die Übertragungsfrequenz von BLE ist im lizenzfreien 2.4 GHz Industrial, Scientific and Medical (ISM)-Band von 2402 MHz bis 2480 MHz [2, S. 4], [1, S. 190]. Das Frequenzband ist in 40 physikalische Kanäle mit jeweils einer Bandbreite von 2 MHz aufgeteilt, wie in Abbildung 1 zu sehen ist [1, S. 190]. Drei dieser 40 physikalischen Kanäle sind für das sogenannte Advertising vorhanden [1, S. 190], welches für die Geräteentdeckung, den Verbindungsaufbau und für das Broadcasting von Nachrichten vorhanden ist [2, S. 4]. Die restlichen Kanäle sind für eine allgemeine Datenübertragung vorhanden [1, S. 190]. Zusätzlich zu der Aufteilung des Frequenzbandes in Kanäle werden Kanäle in Zeiteinheiten aufgeteilt, welche Events genannt werden [1, S. 190]. Daten werden in Paketen innerhalb eines Events übertragen. Zusätzlich wird bei der Übertragung von Daten Frequenzhopping betrieben, welches zu Beginn jedes Events stattfindet [1, S. 190f.].

Bild anpassen und schreiben, abgewandelt von ...

Die Kompatibilität zwischen Bluetooth-Geräten wird durch sogenannte Profile sichergestellt. Profile beschreiben dafür Funktionen und Eigenschaften von jeder Schicht im Bluetoothsystem [1, S. 277]. Ebenso werden die benötigten Nachrichten und Prozeduren für die verwendeten Profile durch die Bluetooth Special Interest Group (SIG) spezifiziert [1, S. 1241].

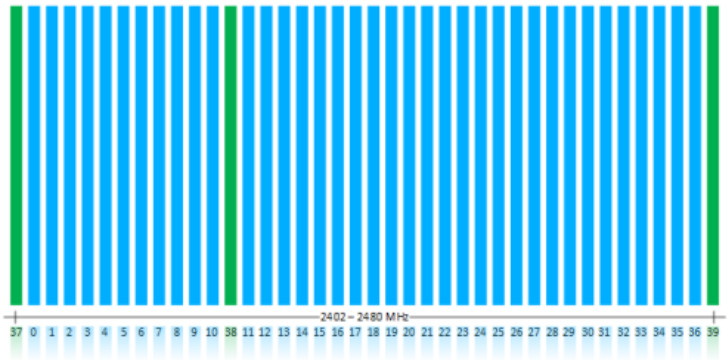


Abbildung 1: Frequenzband mit Kanälen von BLE [2, S. 4]

Bluetooth-Geräten werden unterschiedliche Rollen zugewiesen. Dafür gibt es die Rollen Observer, Broadcaster, Central und Peripheral. Ein Gerät in der Rolle Broadcaster verschickt Advertising-Pakete und ein Gerät welches nur Advertising-Pakete empfangen kann hat die Observer Rolle. So kann eine einseitige Kommunikation zwischen Geräten mittels Advertising-Paketen erfolgen. Eine andere Art der Kommunikation ist mittels einer Verbindung bei dem das Initiatorgerät eine Verbindungsanfrage eines Broadcastergeräts annimmt. Daraufhin bekommt das Initiatorgerät die Rolle Central und das Gerät welches ursprünglich in der Rolle Broadcaster war, die Rolle Peripherial. Anzumerken ist, dass ein Gerät zu jeder Zeit mehrere Rollen unterstützen kann, welche jedoch alle der Bluetooth-Controller unterstützen muss. [1, S. 190f., S. 278, S. 1246ff.]

### 3.2.2 Benötigte Komponenten eines BLE-Geräts

Ein BLE-Gerät benötigt einen Mindestumfang an Funktionen damit es laut Bluetooth SIG BLE kompatibel ist. In Abbildung sind die benötigten Funktionen und deren Zusammenspiel durch ein Schichtenmodell dargestellt. Die Funktionen können dabei in einen Hostteil und einen Controllerteil aufgeteilt werden. Im Hostteil befinden sich die Funktionen Logical Link Control and Adaption Protocol (L2CAP), Generic Access Profile (GAP), Attribute Protocol (ATT), Generic Attribute Pribute Profile (GATT), Service Discovery Protocol (SDP) und Security Manager Protocol (SMP). Im Controllerteil befinden sich die Funktionen Physical Layer (PHY) und Link Layer (LL). Die Kommunikation zwischen den Hostteil und dem Controllerteil finden mittels des Host Controller Interface (HCI) statt [1, S. 1735]. [1, S. 193]

Referenz hinzufügen

Bild erstellen

In den nachfolgenden Unterkapiteln werden die wichtigsten Informationen jeder benötigten Funktion von BLE beschrieben.

#### Physical Layer (PHY)

Die physikalische Schicht in BLE ist zum Verschicken und erhalten von Paketen über eines der physikalischen Funkkanäle verantwortlich. [1, S. 209]

### Link Layer (LL)

Die Verbindungsschicht im BLE-System besteht aus mehreren Komponenten. Eine Komponente ist für die Erstellung, Modifizierung und das Freigeben von logischen Verbindungen zuständig. Eine weitere Komponente ist für das Kodieren und Dekodieren von Bluetooth Paketen zuständig. Auch gibt es eine Komponente welche für die Datenflusskontrolle, die Datenbestätigung und für die Wiederübertragung von Paketen zuständig ist. Die letzten Komponenten in der Verbindungsschicht ist für den Zugriff auf das Radiomedium zuständig. Dafür gibt es einen Scheduler, welcher Zeitschlitze des physikalischen Mediums an die höherliegenden Dienste verteilt. [1, S. 207f.]

### Host Controller Interface (HCI)

Das Host Controller Interface stellt die Möglichkeit bereit, dass der Hostteil die Funktionen des Controllerteil erreichen kann. Die Übertragung des HCI kann dabei wahlweise mittels USB, UART oder anderen Bussystemen stattfinden. [1, S. 1735f.]

### Logical Link Control and Adaption Protocol (L2CAP)

Das Logical Link Control and Adaption Protocol ist die Schicht im BLE-Stack, welches eine kanalbasierte Abstraktion zu den Applikationen und Diensten der höheren Schichten bereitgestellt. Diese Schicht kümmert sich zusätzlich, um die Segmentierung, den Zusammenbau, das Multi- und Demultiplexing von Daten auf einer beziehungsweise mehreren logischen Verbindungen. [1, S. 195, S. 1013]

Logical Link Control and Adaption Protocol baut dabei auf dem Konzept von logischen Kanälen auf, wobei jeder Endpunkt eines logischen Kanals einen eindeutigen Kanalidentifizierer (CID) hat [1, S. 1021]. Die logischen Kanäle werden über logische Verbindungen der LL-Schicht übertragen [1, S. 1013].

### Generic Access Profile (GAP)

Das Generic Access Profile beschreibt die Basisfunktionalitäten welche ein BLE-Gerät benötigt [1, S. 207]. Dabei werden auch alle, in diesem Kapitel vorgestellten Schichten als Mindestanforderung aufgelistet und die alle benötigten Fähigkeiten die eine BLE-Rolle benötigt [1, S. 277f., S. 1241].

Weitere wichtige Eigenschaften die in GAP definiert sind, ist zum einen die Bluetooth-Geräteadresse. Diese Geräteadresse wird verwendet, um ein Bluetooth-Gerät eindeutig zu identifizieren. Eine weitere Eigenschaft, welche in GAP definiert wird, ist der Gerätename. Dieser Name ist eine benutzerfreundliche Zeichenfolge der an entfernten Geräten angezeigt wird. Der Gerätename kann bis zu 248 Byte lang sein und sollte in UTF-8 kodiert sein. Es muss davon ausgegangen werden, dass ein Gerät nur die ersten 40 Zeichen verwenden kann. [1, S. 1251ff.]

Damit eine Verfolgung von Geräteadressen minimiert werden kann, gibt es in BLE zwei Arten von Geräteadressen. Zum einen eine sich verändernde öffentliche Adresse, welche an allen BLE-Geräte verschickt wird. Zum anderen gibt es sich nicht verändernde private Adressen, welche von Geräten ausgerechnet werden kann, welche schon einmal eine Verbindung mit dem Gerät aufgebaut haben. Damit können Geräte, welche schon einmal mit einem anderen Gerät verbunden waren, überprüfen, ob es sich um ein bereits bekanntes Gerät handelt. [2, S. 18]

Auch wird in GAP beschrieben, wie der Bluetooth-Pin für eine Authentifizierung zweier Geräte im Verbindungsmodus aufgebaut sein muss. Diese Pin ist sechs Zeichen lang und besteht aus Ziffern. [1, S. 1253]

Zu guter Letzt, beschreibt GAP noch die verschieden Sicherheitsmodi, welche durch die verschiedenen BLE-Rollen implementiert sein müssen [1, S. 1337].

### Service Discovery Protocol (SDP)

SDP stellt die Möglichkeit bereit, die verfügbaren Dienste und die zugehörigen Merkmale eines Bluetooth-Geräts für entfernte Geräte sichtbar zu machen [1, S. 1173]. Dabei pflegt das Gerät, welches SDP bereitstellt, eine Liste aller Dienste und Merkmale des Geräts [1, S. 1177].

### Security Manager Protocol (SMP)

SMP definiert Methoden zum Verbindungsaufbau und zum Schlüsselaustausch zwischen Bluetooth-Geräten [1, S. 1554]. Die gerätespezifischen Schlüssel, werden für die Identifizierung von Geräten und für den verschlüsselten Datenaustausch zwischen Geräten verwendet [1, S. 1556], [2, S. 18].

Der Verbindungsaufbau und der dazugehörige Schlüsselaustausch für die Identifizierung der Geräte erfolgt in 3 Phasen. Die erste Phase ist die Anfrage für einen Verbindungsaufbau. Die zweite Phase, nach einer erfolgreichen Anfrage, ist die Generierung eines Schlüssels mit einer kurzen oder langen Lebenszeit. Die letzte Phase ist die Bereitstellung der generierten Schlüssel an die Gegenstelle. [1, S. 1556]

Zu beachten ist, dass es verschiedene Möglichkeiten gibt einen Verbindungsaufbau herzustellen, der abhängig von den Sicherheitsansprüchen der Anwendung definiert werden kann [2, S. 18].

### Attribute Protocol (ATT)

ATT ist ein Teilnehmer-zu-Teilnehmer Protokoll zwischen zwei Geräten [1, S. 206]. ATT definiert dabei zwei Rollen, den Client und den Server [1, S. 1410]. ATT erlaubt es Geräten – Clients – kleine Werte – sogenannte Attribute [1, S. 279] – zu lesen, zu schreiben und zu entdecken, welche sich auf dem Gerät mit der Rolle Server befinden [1, S. 1409]. Ein Gerät kann simultan in der Rolle Server und Client sein [1, S. 279].

Ein Attribut besteht jeweils aus drei Eigenschaften. Die erste Eigenschaft ist der Attribut-Typ, welcher durch eine universal unique identifier (UUID) definiert wird und in SDP definiert sind. Die zweite Eigenschaft ist der Attribut-Handle. Der Attribut-Handle ist ein einzigartiger

Identifikator für ein Attribut auf einem Gerät mit der Server-Rolle. Dadurch das Handle ist das Attribut eindeutig auf dem Gerät definiert. Die letzte Eigenschaft eines Attributs sind die Berechtigungen, welche durch eine höhere Schicht definiert werden muss. [1, S. 1410ff.]

Attribut-Handles haben eine Länge von 16 Bit und können durch weitere spezielle Attribute gruppiert werden [1, S. 1412f.]. Die Entdeckung aller vorhandenen Attribute eines Servers durch einen Client erfolgt durch eine höhere Schicht des BLE-Stacks [1, S. 1410].

Die hinterlegten Werte eines Attributs bestehen aus einem Oktett-Array mit einer fixen oder variablen Länge [1, S. 1413].

### Generic Attribute Pribute Profile (GATT)

GATT baut auf ATT auf und stellt ein Framework für die Daten, welche in ATT gespeichert werden, bereit. GATT stellt wie ATT zwei Rollen – den Server und den Client – bereit. Ebenso definiert GATT das Format der Daten, welche auf dem GATT-Server gespeichert werden dürfen, in sogenannten Profilen. Attribute werden hierfür in Profile, Dienste und Merkmale untergliedert, wie in Abbildung zu sehen ist. Ein Applikationsprofil besteht aus einen oder mehreren Diensten, um bestimmte, definierte Use-Cases abzudecken und definiert darüber hinaus die benötigten Dienste, Merkmale und Attribute [1, S. 207]. Ein Dienst enthält eine Ansammlung von Merkmalen und kann andere Dienste inkludieren. Ein Merkmal enthält ein Wert, sowie eine Menge von Deskriptoren. Durch diesen Aufbau ist es einen Client möglich die Daten eines bestimmten Profils auszulesen ohne davor den Aufbau der Attribute des Servers kennen zu müssen. [1, S. 280, S. 1480]

Referenz hinzufügen

Anzumerken ist, dass jedes Attribut, welches in ATT vorhanden ist, entweder in einer Dienstdeklarierung oder in einer Dienstdefinition enthalten sein soll. [1, S. 1483]

Bild hinzufügen

Das GATT-Profil soll von anderen Profilen als Grundstruktur verwendet werden, damit eine reibungslose Kommunikation zwischen einen Client und Server sichergestellt werden kann, wie in Abbildung zu sehen ist. [1, S. 1470]

Referenz hinzufügen

Bild hinzufügen

Ein Dienst stellt unter GATT eine Ansammlung von Daten dar, um ein bestimmtes Verhalten durch das vorhandene Gerät darzustellen. Ein Dienst kann zur Vereinfachung der Verhaltensdarstellung weitere Dienste inkludieren. Dienste können in zwei Gruppen eingeteilt werden. Zunächst einmal in die primären Dienste. Primäre Dienste bieten alleinstehende Funktionalitäten an. Im Gegensatz dazu gibt es sekundäre Dienste, welche optionale Funktionalitäten enthalten und von mindestens einen primären Dienst inkludiert werden müssen. [1, S. 281]

Die Definition eines Dienstes umfasst die inkludierten Dienste sowie die benötigten und optionalen Merkmale [1, S. 1481].

Der Start eines Dienstes in der Liste der ATT-Attribute wird durch ein spezielles Attribut festgelegt, mit dem Attribut-Typ *primärer Dienst* oder *sekundärer Dienst*. Das Ende eines Dienstes wird durch eine Folgedeklaration eines neuen Dienstes festgelegt. [1, S. 1483]

Merkmale sind Werte eines Dienstes welche aus mehreren ATT-Attributen besteht. Ein Merkmal besteht aus drei Komponenten. Der Deklaration, den Eigenschaften des Merkmals und dem dazugehörigen Wert. Zusätzlich können noch Deskriptoren in einem Merkmal enthalten sein, um die Berechtigungen des Merkmals zu setzen. [1, S. 281]

Der Start eines Merkmals in der Liste der ATT-Attribute wird durch ein spezielles Attribut festgelegt, welche den Attribut-Typ *Merkmal* enthält. Das Ende eines Merkmals stellt eine neue Merkmaldeklaration oder eine neue Dienstdeklaration dar. [1, S. 1484ff.]

### 3.2.3 Sollanforderungen durch Apple

### 3.2.4 HID over GATT Profile (HOGP)

#### HID-Dienst

#### Report Merkmal

#### Report Map Merkmal

#### HID Information Merkmal

#### HID Control Point Merkmal

#### Zusätzliche Bedingungen durch das HID over GATT Profile

#### Batteriedienst

#### Zusätzliche Bedingungen durch das HID over GATT Profile

#### Geräteinformationsdienst

#### Zusätzliche Bedingungen durch das HID over GATT Profile

#### Scan Parameters Profil

### 3.2.5 Bluetooth-Stacks

#### Bluedroid

#### NimBLE

## 3.3 Übertragungsprotokolle am Fernbedienungsmodulschacht

### 3.3.1 Puls-Positions-Modulation (PPM)

### 3.3.2 CRSF

### 3.3.3 SBUS

### 3.3.4 MULTI

## 3.4 Mikrocontroller ESP

mehrere Kerne; Pins; Kommunikationsmöglichkeiten; CE z Zertifizierung (da Antenne schon vorhanden muss nicht erneut zertifiziert werden)

## 3.5 FreeRTOS

sheduling und Kommunikation zwischen Tasks, interrupts und priorisierung

## 3.6 BITMAP Schriftarten

## 3.7 libevdev

# 4 Umsetzung

Zu Beginn soll auf Basis eines ESPRESSIF ESP32-WROOM-32-Entwicklerboards die Kommunikation zum Endgerät als BLE-HID-Gerät entwickelt werden. Als nächster Schritt wird die Kommunikation mit der Multicopterfernbedienung über den vorhandenen Modulschacht implementiert. Sobald beide Kommunikationsschnittstellen einzeln funktionsfähig sind, sollen diese im darauffolgenden Schritt in einem Gesamtsystem zusammengeführt werden. Als letzter Schritt soll die gesamte benötigte Hardware auf eine Platine gebaut werden und für das Modul ein 3D-gedrucktes Gehäuse hergestellt werden. Das ESP32-WROOM-32-Modul soll für die BLE-Kommunikation verwendet werden, da es ein kostengünstiges und nach CE zertifiziertes Modul ist. Die Authentifizierung an den Endgeräten soll als HID erfolgen, da dadurch keine zusätzlichen Treiber entwickelt werden müssen und ebenso die Zertifizierung durch Endgerätehersteller entfällt wie beispielsweise bei Apple.

Schreiben warum hid profil, da dafür kein Treiber geschrieben werden muss. Bluetooth-Profile und benötigte HID-struktur. Datenaustausch esp und fernbedienung. Tasks am ESP erklären wie die priorisiert sind und interrupts. Display erklären und wie dort geschrieben werden kann. PCB-Design erklären. (Erklären für was die zonen sind und was beachtet werden musste, batterie auslesen, esd schutz, usb zu serial, schutzschaltung strom, Spannungsregulierung Datenleitungen) Case-design erklären und was dort beachtet wurde.

# 5 Validierung und Gegenüberstellung

## 5.1 Validierung des Funktionsumfangs

Schauen ob das Gerät unter Linux, Android, Windows, iOS funktioniert. Schauen ob die Kommunikation mit dem Kontroller funktioniert.

## 5.2 Validierung der Leistung

? Keine Ahnung was ich da gemeint habe.

## 5.3 Gegenüberstellung BLE-Modul und USB-Verbindung

Test zunächst mit servo probiert, um nicht an Platine direkt arbeiten zu müssen. Jedoch ist der Delay nicht in einen glaubwürdigen bereich, da zu lang. Eine Studie gefunden, bei dem optokoppler verwendet wurden, und verschiedene Geräte getestet wurde als vergleichswert verwendbar. Dadurch neuer Versuchsaufbau mit optokoppler. Schauen ob es in einen bereich mit den restlichen ist und wie viel schlechter es wurde. Vielleicht gaußverteilung darstellen und werte dafür raussrechnen. Schreiben dass x mal getestet wurde.

# 6 Rekapitulation und Ausblick

# Literatur

[1] *Bluetooth Core Specification*, Revision v5.3, Bluetooth SIG, 2021.

[2] *UG103.14: Bluetooth LE Fundamentals*, Revision 0.7, SILICON LABS.

# Anhang

A. Assignment

B. List of CD Contents

C. CD

# B. List of CD Contents