



HOCHSCHULE
DER MEDIEN

Softwareprojekt

MI7 | Wintersemester 24/25

Projektdokumentation

DIY - Minidrohone

Martin Herdt, mh370@hdm-stuttgart.de
Fabio Bradner, fb109@hdm-stuttgart.de

<https://gitlab.mi.hdm-stuttgart.de/mh370/drohnenprojekt.git>

Inhaltsverzeichnis

1. Einleitung	1
1.1 Motivation: Warum ein Drohnenprojekt?.....	1
1.2 Projektziele.....	1
2. Entwicklungsfortschritt	1
3. Prototypenentwicklung	2
3.1 Prototyp - Verkabelung.....	2
3.2 Prototyp - Bluetooth.....	2
3.3 Prototyp - Test Aufbau.....	3
3.4. Prototyp - Aufbau (final).....	3
4. Vorgehensweise und Entwicklungszyklus	4
5. Testing und Fortschritt	4
5.1 Einzelsystemtests.....	4
5.2 Flugstabilitätstests.....	4
5.3 Stress- und Belastungstests.....	4
5.4 Unittest für MAC-Adressen Autorisierung.....	5
5.5 Integrationstest für Motorsteuerung.....	5
5.6 Funktionstest für den Failsafe-Mechanismus.....	5
6. Softwareentwicklung	5
7. SOLID Prinzipien und Architektur	6
8. Logging im Drohnenprojekt	6
9. Clean Code Prinzipien	7
10. Arbeiten mit GIT	8
11. Komponentenliste	8
12. Herausforderungen und Schwierigkeiten	9
13. Stellungnahme Gruppenmitglieder	10
14. Besonderheiten	11
15. Bereit zum Fliegen	11
16. Controller Tastenbelegung	11
17. Use-Case-Diagramm	12
18. Fazit	12

1. Einleitung

1.1 Motivation: Warum ein Drohnenprojekt?

Die Idee, eine eigene Minidrohne zu bauen, entstand aus der Kombination von Neugier und dem Wunsch, technische Fähigkeiten in verschiedenen Bereichen wie Mechanik, Elektronik und Programmierung praktisch anzuwenden. Da uns reine Softwareprojekte in letzter Zeit häufiger begegnet sind, wollten wir uns einer Hardware-Herausforderung widmen, die sowohl spannend als auch lehrreich ist. Die Entwicklung einer Minidrohne bietet viele interessante Aspekte: Von der Konstruktion des Rahmens über die Auswahl und Integration der Komponenten bis hin zur Entwicklung der Flugsteuerungssoftware.

1.2 Projektziele

Unser Ziel war es, eine Drohne zu bauen, die:

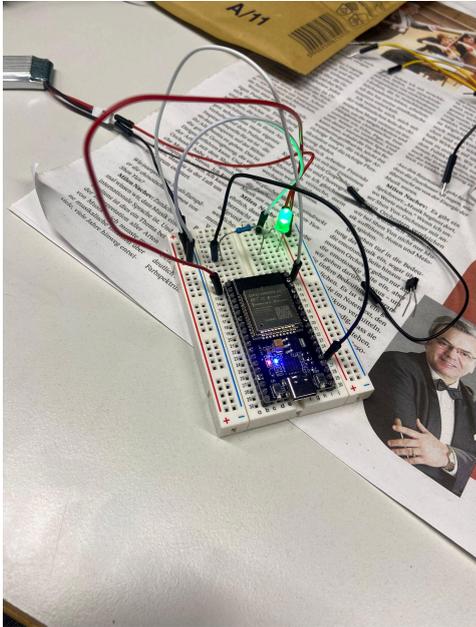
- Stabil abheben und landen kann,
- In der Luft präzise Manöver ausführen kann,
- Über einen Controller gesteuert werden kann.

2. Entwicklungsfortschritt

Die Entwicklung unserer Drohne verlief in mehreren Schritten. Hier eine Übersicht über den Fortschritt:

1. Konzeptphase: Definition der Anforderungen an die Drohne (Größe, Gewicht, Flugzeit, Steuerung).
2. Komponentenauswahl: Auswahl der benötigten Hardware-Komponenten (siehe Komponentenliste unten).
3. 3D-Druck: Konstruktion des Drohnengehäuses mit CAD-Software und Herstellung mittels 3D-Druck.
4. Montage: Zusammenbau der Drohne, inklusive Motorhalterungen, Verkabelung und Controller.
5. Softwareentwicklung: Programmierung des Flight Controllers und der Steuerlogik
6. Tests und Optimierungen: Flugtests, Fehleranalyse und Optimierung von Hardware und Software.

3. Prototypenentwicklung



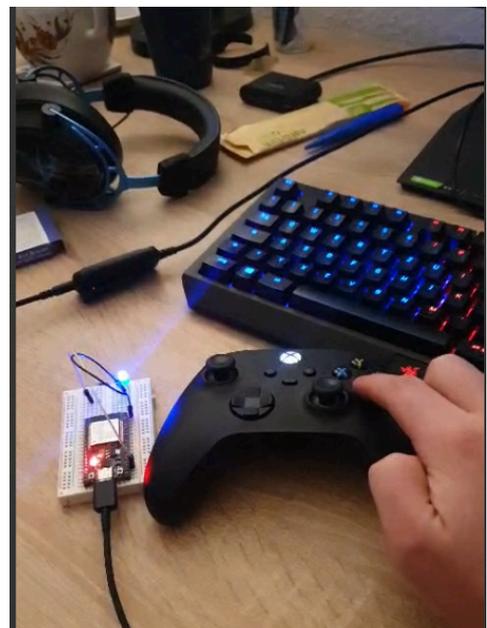
1. Prototyp Versuch

3.1 Prototyp - Verkabelung

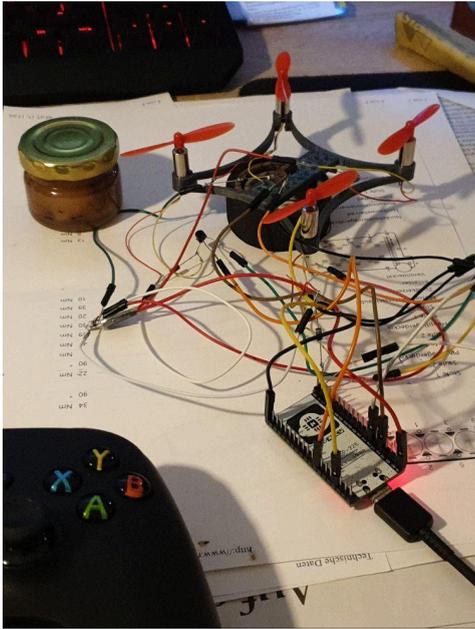
Für den ersten Prototyp wurde zunächst eine provisorische Testverkabelung erstellt, um die grundlegenden Funktionen der Drohne zu überprüfen. Dabei stand insbesondere die Funktionsfähigkeit der elektronischen Komponenten im Fokus. Zur ersten Überprüfung wurde eine einfache LED-Schaltung genutzt, die durch Blinken signalisierte, dass der Mikrocontroller korrekt arbeitet und Befehle verarbeitet. Nachdem diese grundlegenden Tests erfolgreich abgeschlossen wurden, konnte die Verkabelung weiter optimiert und auf eine stabilere Lösung umgestellt werden.

3.2 Prototyp - Bluetooth

Für die Implementierung der Bluetooth-Verbindung wurde zunächst die Kommunikation zwischen dem Mikrocontroller und dem Controller eingerichtet. Anschließend wurde die Verbindung getestet, indem der Controller Signale an den Mikrocontroller sendete. Zur Überprüfung der erfolgreichen Datenübertragung wurde eine LED als Indikator genutzt, die per Knopfdruck auf dem Controller zum Blinken gebracht wurde. Dieser einfache Test bestätigte, dass die Steuerbefehle korrekt empfangen und verarbeitet wurden, wodurch die Grundlage für die weitere Integration der Steuerung geschaffen wurde.



2. Prototyp Versuch



3.3 Prototyp - Test Aufbau

Für den Prototyp wurde die Verkabelung zwischen dem Mikrocontroller und den Motoren durchgeführt, um eine funktionierende Ansteuerung der Antriebseinheiten zu gewährleisten. Anschließend erfolgte ein erster Test der Motorsteuerung, bei dem die Drehzahlregelung und Reaktionsfähigkeit überprüft wurden. Nach erfolgreicher Prüfung wurden die Motoren fest am Gehäuse angebracht, sodass eine stabile und vibrationsarme Befestigung sichergestellt wurde.

3. Prototyp Versuch

3.4. Prototyp - Aufbau (final)

Der erste vollständige Prototyp wurde realisiert, indem die zuvor getestete Verkabelung durch eine stabilere und dauerhafte Lösung ersetzt wurde. Dafür wurden die Leiterbahnen auf einer Lochrasterplatte verlötet, um eine zuverlässige elektrische Verbindung zwischen dem Mikrocontroller, den Motoren und den weiteren Komponenten sicherzustellen. Diese optimierte Schaltung erhöhte die Stabilität und reduzierte mögliche Kontaktprobleme, wodurch die Drohne für weitere Tests und die Integration der Steuerungssoftware vorbereitet wurde.



4. Prototyp Versuch

4. Vorgehensweise und Entwicklungszyklus

Unsere Herangehensweise und der Entwicklungszyklus waren iterativ aufgebaut und bestanden aus folgenden Schritten:

1. **Recherche:** Sammeln von Informationen über bestehende Drohnenprojekte und Grundlagen der Flugphysik.
2. **Planung:** Erstellung eines Konzepts und eines Zeitplans, um die verschiedenen Entwicklungsphasen zu organisieren. Definition der Ziele und Erstellung eines groben Designs.
3. **Prototyping:** Bau eines ersten Prototyps mit einfachen Materialien, um grundlegende Funktionen zu testen. Anschließend Aufbau eines funktionalen Prototyps.
4. **Testen:** Durchführung von Tests zur Überprüfung der Hardware- und Softwarefunktionalität.
5. **Optimierung:** Behebung von Problemen und Verbesserung der Performance basierend auf Testergebnissen.
6. **Finalisierung:** Integration aller Komponenten und Vorbereitung für den Einsatz.

5. Testing und Fortschritt

Nach der Montage der Drohne haben wir sie in mehreren Stufen getestet:

5.1 Einzelsystemtests

- Test der Motoren und Propeller.
- Überprüfung der Stromversorgung und Steuerungskomponenten.

5.2 Flugstabilitätstests

- Stabilität beim Schweben.
- Reaktionsfähigkeit auf Steuerbefehle.

5.3 Stress- und Belastungstests

- Flüge unter maximaler Last.
- Testen der Komponenten bei längerer Betriebsdauer

5.4 Unittest für MAC-Adressen Autorisierung

Die Funktion `isAuthorizedController` überprüft, ob der sich zu verbinden versuchende Controller eine autorisierte MAC-Adresse besitzt. Diese Funktion kann geprüft werden, wenn der Microcontroller mit einem Rechner verbunden ist und "t" eingegeben wird. Es wird eine nicht autorisierte Test-Adresse überprüft und eine autorisierte Test-Adresse. Das Ergebnis wird über den seriellen Monitor ausgegeben. Danach wird in den normalen Betrieb zurückgekehrt.

5.5 Integrationstest für Motorsteuerung

Die Motorsteuerung wird getestet, indem man über einen verbundenen Rechner "m" eingibt. Daraufhin werden die Motoren für zwei Sekunden angesteuert. Danach wird auch hier in den normalen Betrieb zurückgekehrt.

5.6 Funktionstest für den Failsafe-Mechanismus

Der Failsafe-Mechanismus stoppt die Motoren automatisch, wenn eine bestimmte Zeit seit der letzten Betätigung des Controllers vergangen ist. Dieser Mechanismus wird über einen Funktionstest durch die Eingabe von "f" gestartet. Die Motoren stoppen. Eine Ausgabe über den Erfolg des Tests erfolgt ebenfalls über den seriellen Motor. Wie gewohnt wird danach in den normalen Betrieb übergegangen.

6. Softwareentwicklung

Der Flight Controller wurde mit der Programmiersprache C++ auf dem ESP32-Mikrocontroller entwickelt. Hier sind die wichtigsten Softwarekomponenten:

1. PID-Regler: Zur Stabilisierung der Drohne in der Luft.
2. Sensorintegration: Verwendung eines Gyroskops und Beschleunigungsmessers zur Lagedetektion.
3. Datenkommunikation: Anbindung eines Xbox-Controllers über Bluetooth zur Steuerung.
4. Failsafe-Mechanismen: Automatisches Landen bei Signalverlust oder geringer Batterieladung.

7. SOLID Prinzipien und Architektur

- **Single Responsibility Principle:**
Jede Komponente erfüllt eine klar definierte Aufgabe, darunter die Flugsteuerung, Sensorverarbeitung, Motoransteuerung und Datenkommunikation.
- **Open-Closed Principle:**
Die Software ist modular aufgebaut und ermöglicht Erweiterungen wie neue Sensoren oder Flugmodi ohne grundlegende Code-Änderungen.
- **Liskov Substitution Principle:**
Motor- und Sensormodule können ausgetauscht werden, ohne Anpassungen am zentralen Flight Controller vornehmen zu müssen.
- **Interface Segregation Principle:**
Klare Trennung der Schnittstellen für Sensoren, Motorsteuerung und Controller-Eingaben sorgt für eine schlanke und effiziente Codebasis.

Architektur des Codes:

Die Drohne ist modular aufgebaut, wobei der Flight Controller die Steuerlogik übernimmt und Sensordaten in Echtzeit verarbeitet. Die Kommunikationsschnittstelle stellt die Verbindung zum Controller her, während ein Failsafe-Mechanismus kritische Situationen erkennt und eine sichere Landung einleitet. Diese Struktur gewährleistet eine einfache Wartung und Erweiterbarkeit.

8. Logging im Drohnenprojekt

Im Drohnenprojekt setzen wir gezieltes Logging ein, um wichtige Systemereignisse, Fehlermeldungen und Diagnosedaten zu protokollieren. Dabei werden insbesondere Steuerbefehle, Sensordaten und Motoraktivitäten überwacht, um die Systemstabilität sicherzustellen. Alle relevanten Informationen werden über den seriellen Monitor ausgegeben, wodurch eine direkte Analyse während der Testphasen möglich ist. Zusätzlich haben wir eine Logging-Struktur implementiert, die Fehlerzustände wie Signalverlust, unerwartete Sensordaten oder Motorstörungen erkennt und dokumentiert. Dadurch können potenzielle Probleme schneller identifiziert und behoben werden. Insbesondere wird die Verbindung zwischen Controller und Drohne kontinuierlich überwacht, um Verzögerungen oder Abbrüche sofort zu erkennen und entsprechend zu reagieren.

9. Clean Code Prinzipien

In unserem Drohnenprojekt wurden **Clean Code Prinzipien** angewendet, um eine strukturierte und wartbare Codebasis zu gewährleisten. Dazu gehören eine **klare Namensgebung**, eine **modulare Struktur** mit kurzen, fokussierten Funktionen sowie **sinnvolle Kommentare** für bessere Verständlichkeit. Ein **Failsafe-Mechanismus** sorgt für die automatische Motorabschaltung, während **Logging und die Vermeidung von Magic Numbers** die Wartung und Fehleranalyse erleichtern.

Clean Code Guidelines:

❖ Klare Namensgebung

- Englische Bezeichnungen
- Variablen in LowerCamelCase
- Konstanten in Caps
- Klassen in UpperCamelCase

❖ Strukturierte und modulare Funktionen

- Jede Funktion hat eine klar definierte Aufgabe
- Funktionen bleiben kurz und fokussiert

❖ Sinnvolle Kommentare

- Wichtige Abschnitte sind verständlich beschrieben
- Kommentare in Deutsch für bessere Lesbarkeit

❖ Gute Code-Formatierung

- Einheitliche Einrückung und Struktur
- Logische Gruppierung für bessere Übersicht

❖ Failsafe-Mechanismus

- Automatische Motorabschaltung bei fehlendem Signal
- Erhöhung der Sicherheit

❖ Vermeidung von Magic Numbers

- Nutzung vordefinierter Konstanten
- Erhöht die Wartbarkeit des Codes

❖ Logging für Debugging

- Serielle Ausgaben zur Überwachung der Steuerung

10. Arbeiten mit GIT

Im Rahmen unseres Projekts haben wir die Git-Flow-Strategie angewendet, indem wir verschiedene Aufgabenteile in separaten Feature-Branches bearbeitet haben. Jeder Task wurde in einem eigenen Branch entwickelt, um parallel an verschiedenen Teilen des Projekts arbeiten zu können, ohne uns gegenseitig zu beeinträchtigen.

Sobald ein Task abgeschlossen war und alle Tests erfolgreich durchgeführt wurden, haben wir den entsprechenden Branch in die Entwicklungsumgebung integriert. Dafür haben wir einen Pull-Request erstellt und nach einer Überprüfung durch andere Teammitglieder den Branch in die Entwicklungsumgebung gemerged.

Auf diese Weise konnten wir sicherstellen, dass alle Änderungen vor der Integration in die Hauptentwicklungslinie gründlich überprüft wurden und die Stabilität des Codes gewährleistet war. Nachdem alle Aufgaben erfolgreich abgeschlossen und in die Entwicklungsumgebung integriert wurden, haben wir den finalen Entwicklungsstand in die main-Branch gemerged.

Diese Branch wird als Ausgangspunkt für zukünftige Entwicklungen und Veröffentlichungen verwendet.

11. Komponentenliste

Hier sind die wichtigsten Komponenten, die wir für unsere Drohne verwendet haben:

- ESP32: Mikrocontroller für die Steuerung.
- 4x bürstenlose Elektromotoren: Für den Antrieb.
- Propeller: Passend zu den Motoren.
- Gyroskop (MPU6050): Für die Stabilisierung.
- LiPo-Akku: Stromversorgung der Drohne.
- Drohnengehäuse: Selbst konstruiert und 3D-gedruckt.
- 4x IRLZ44N Mosfets zur Spannungsversorgung
- Xbox-Controller: Zur Steuerung der Drohne.
- Freilaufdioden: Zum Schutz vor Rückströmen in der Schaltung.

12. Herausforderungen und Schwierigkeiten

Während der Entwicklung unserer Minidrohne standen wir vor verschiedenen Herausforderungen, die wir mit Geduld und Einsatz bewältigen mussten. Die Kombination aus mechanischen, elektronischen und softwaretechnischen Aspekten stellte hohe Anforderungen an unser technisches Verständnis und unsere Problemlösungsfähigkeiten. Einige dieser Herausforderungen umfassten:

- **Elektronische Komplexität:** Die Entwicklung einer stabilen Schaltung erforderte eine intensive Einarbeitung in Schaltpläne, Stromkreise und deren praktische Umsetzung. Besondere Herausforderungen lagen in der Auswahl und Integration geeigneter Bauteile für eine zuverlässige Stromversorgung.
- **Komponentenwahl:** Einige Bauteile erwiesen sich als ungeeignet, sodass wir Anpassungen vornehmen und neue Komponenten beschaffen mussten, was zu Verzögerungen und zusätzlichem Aufwand führte.
- **Feinmotorische Arbeiten:** Das präzise Lötten filigraner Verbindungen war anspruchsvoll und erforderte höchste Konzentration, um Fehler wie Kurzschlüsse oder kalte Lötstellen zu vermeiden.
- **Organisatorische Schwierigkeiten:** Da das Team verkleinert wurde, mussten Aufgaben effizienter verteilt und klare Kommunikationswege geschaffen werden, um den Fortschritt sicherzustellen.
- **Anpassungsfähigkeit an Änderungen:** Mehrfache Anpassungen an Steuerungslogik und Hardware erforderten zusätzliche Tests und verbessertes Zeitmanagement.

Was gut lief: Trotz der Herausforderungen hielten wir uns an unsere Termine und Deadlines. Die Zusammenarbeit war effektiv, mit klaren Aufgabenverteilungen und gegenseitiger Unterstützung. Unsere strukturierte Vorgehensweise half uns, Probleme schnell zu lösen und das Projekt erfolgreich umzusetzen.

13. Stellungnahme Gruppenmitglieder

Martin:

Die größten Herausforderungen in unserem Drohnenprojekt lagen in der Teamorganisation, der Zusammenarbeit und der technischen Umsetzung. Besonders die komplexe Schaltung und Steuerung erforderten viel Einarbeitung und präzises Arbeiten. Dabei wurde deutlich, wie wichtig klare Kommunikation und eine strukturierte Aufgabenverteilung sind.

Während des Projekts konnte ich meine technischen und organisatorischen Fähigkeiten weiterentwickeln. Rückblickend würde ich als ersten Schritt mehr Zeit in eine umfassendere Recherche investieren, um alle Aspekte besser zu verstehen und das Projekt gezielter zu planen. Dadurch hätten wir mögliche Herausforderungen früher erkannt und effizienter bewältigen können.

Zusätzlich würde ich noch mehr Zeit in die Software investieren, um die Steuerung der Drohne weiter zu optimieren.

Fabio:

Die wichtigste Erkenntnis des Projekts liegt für mich in der Planung. Besonders der Einkauf der richtigen Hardware-Bauteile als auch der Zusammenbau der Drohne hat sehr viel Zeit und doppelten Aufwand in Anspruch genommen.

Mit dem aus dem Projekt dazugewonnenen Wissen würden mir diese Vorgänge in Zukunft leichter fallen, trotzdem würde ich mehr Zeit dafür einplanen. Zudem habe ich erneut gemerkt, dass es wichtig ist, die Muss-Anforderungen auf das Mindeste zu reduzieren, um direkt zu Beginn Prioritäten zu setzen. Positiv aus dem Projekt nehme ich zudem die neuen Erkenntnisse zu elektronischen Bauteilen und Schaltkreisen mit.

Wichtig zu erwähnen finde ich, dass die Gespräche auf der MediaNight sehr hilfreich waren und zu neuen Ideen angeregt haben. Zusammengefasst war das Projekt durch die Hardware-Komponente sehr abwechslungsreich und ich konnte mir einige neue Fähigkeiten aneignen.

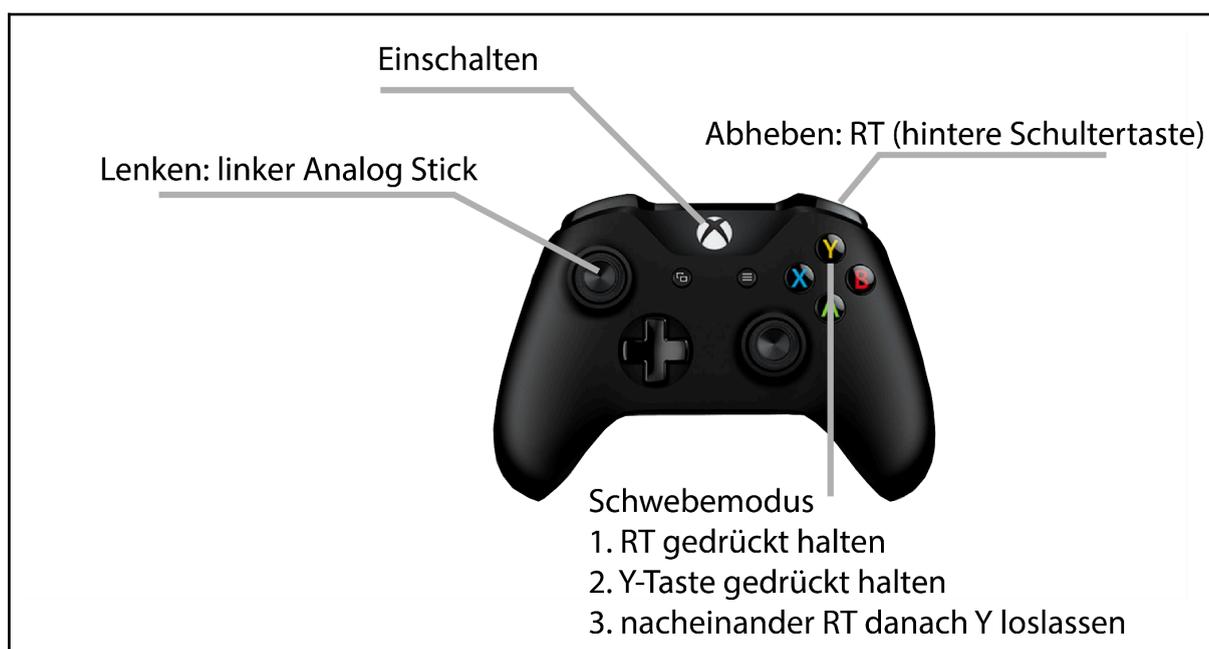
14. Besonderheiten

Die Drohne zeichnet sich durch ihre hohe Anpassungsfähigkeit aus. Sie kann mit nahezu allen Bluetooth-Controllern gesteuert werden, was eine flexible Nutzung ermöglicht. Zudem sind die Motoren modular aufgebaut und lassen sich bei Bedarf leicht austauschen. Das Gehäuse wurde mittels 3D-Druck gefertigt, wodurch es individuell anpassbar und leicht ersetzbar ist. Darüber hinaus wurde der Code so entwickelt, dass er einfach modifiziert werden kann, um verschiedene Steuerungs- und Flugeigenschaften nach Bedarf zu optimieren.

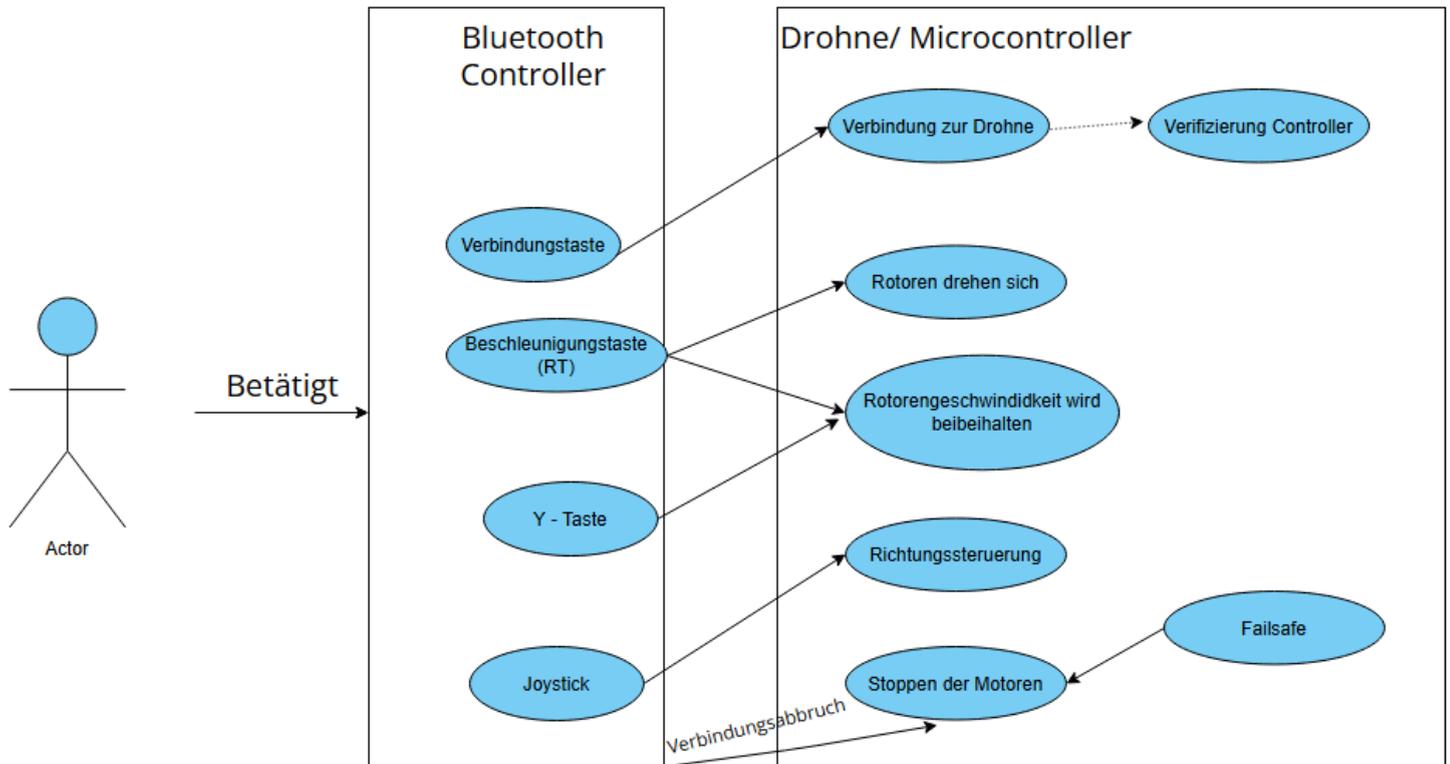
15. Bereit zum Fliegen

1. Drohne leicht mit Finger nach unten drücken
2. Akku an die dafür vorgesehene Steckverbindung anbringen und fixieren
3. Ein Motor springt kurz auf Vollast beim Akku anstecken, deshalb **festhalten!!!**
4. Nun den Controller einschalten und verbinden
5. Bei anderen Controller bitte vorher beim ESP 3.Sek die RST Taste gedrückt halten und dann den Controller in den Bluetooth Pairing Mode versetzen
6. Bereit zum Abflug

16. Controller Tastenbelegung



17. Use-Case-Diagramm



18. Fazit

Das DIY-Drohnenprojekt war eine herausfordernde, aber lehrreiche Erfahrung. Durch die Kombination von Mechanik, Elektronik und Softwareentwicklung konnten wir unser Wissen erweitern. Trotz anfänglicher Schwierigkeiten mit der Komponentenwahl und der Schaltungsentwicklung haben wir eine funktionstüchtige Minidrohne realisiert.

Besonders positiv waren unsere **effiziente Zusammenarbeit**, die **klare Aufgabenverteilung** und die **Einhaltung von Deadlines**, wodurch wir das Projekt strukturiert umsetzen konnten. Gespräche auf der **MediaNight** lieferten zudem wertvolle Anregungen.

Neben den technischen Fortschritten hat uns das Projekt viel **Freude bereitet**. Das gemeinsame Tüfteln, das Beheben von Problemen und die ersten erfolgreichen Flugtests waren besonders motivierend. Rückblickend zeigt das Projekt, wie wichtig Planung, Tests und Anpassungsfähigkeit sind. Die gewonnenen Erkenntnisse werden uns in zukünftigen Projekten helfen.