



# **Entwicklung einer mobilen Roboterplattform**

**Elektronische Produktentwicklung (MECH-B-4-AED-EPE1-ILV)**

**Gruppe Roboterplattform**

Bachelor Programm - Mechatronik

Drittes Semester

Lektor: Mathias Gfall

Gruppe: BA-MECH-23-AED

Autoren: Manuel Hofer, Alexander Oberhofer, Leo Maier, Fabian Augschöll

16. Dezember 2025

# Inhaltsverzeichnis

<b>1 Planung</b>	<b>1</b>
1.1 Anforderungen . . . . .	1
1.2 Ziel des Projekts . . . . .	1
1.3 Stückliste und Kosten . . . . .	1
1.4 Projektmeilensteine für die modulare Roboterplattform . . . . .	1
1.4.1 Semester 1: Entwicklung des fahrfähigen Prototyps . . . . .	1
1.4.2 Semester 2: Integration von Lokalisierung und Sensorfusion . . . . .	2
<b>2 Umsetzung</b>	<b>2</b>
2.1 Bauteilauswahl . . . . .	2
2.1.1 Mikrocontroller . . . . .	2
2.1.2 Batterie . . . . .	3
2.1.3 Motor . . . . .	3
2.1.4 Kabelloses laden . . . . .	3
2.1.5 Time of Flight . . . . .	3
2.1.6 Farbsensor . . . . .	3
2.1.7 Marvelmind . . . . .	3
2.2 Grundschaltungen . . . . .	3
2.2.1 Linearregler (LDO) – TLV767DRV . . . . .	3
2.2.2 Abwärts-/Aufwärtswandler – TPS63070RNMT . . . . .	4
2.2.3 Lademanagement-IC – MCP73831T-2ACI/OT . . . . .	4
2.2.4 Motortreiber – DRV8835DSSR . . . . .	5
2.2.5 Aufwärtswandler – TPS61088RHLT . . . . .	5
2.3 Gesamtschaltung . . . . .	5
2.4 Routing und Platine . . . . .	6
2.5 Mechanische Konstruktion . . . . .	7
2.6 Prototypen Kosten . . . . .	9
<b>3 2. Semester</b>	<b>10</b>
3.1 Stand nach dem ersten Semester . . . . .	10
3.2 Einführung . . . . .	10
3.3 Platinenbestückung . . . . .	10
3.4 Inbetriebnahme . . . . .	11
3.4.1 Jumper Konfiguration . . . . .	11
3.4.2 Erstes Einschalten und Spannungsniveau einstellen . . . . .	11
3.4.3 ESP Inbetriebnahme . . . . .	12
3.4.4 Motor Inbetriebnahme . . . . .	13
3.5 Mechanische Konstruktion . . . . .	13
3.6 Mängel . . . . .	13

<b>4 Auswertung</b>	<b>15</b>
4.1 Upgrades für Version 2 . . . . .	15
<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>V</b>
<b>Literatur</b>	<b>VI</b>

# 1 Planung

## 1.1 ANFORDERUNGEN

## 1.2 ZIEL DES PROJEKTS

Ziel des Projekts ist die Entwicklung einer autonomen Roboterplattform zur Veranschaulichung von Sensorfusion im Zusammenhang mit Positionsbestimmung. Die Plattform nutzt ein Indoor-Lokalisationsystem der Firma Marvelmind und kombiniert verschiedene Sensordaten wie Radodometrie und Gyroskopwerte. Eine zentrale Controllereinheit übernimmt die Verarbeitung der Daten, steuert die Antriebsmotoren und ermöglicht die Kommunikation mit einem Host-System. Die Plattform wird für den Akkubetrieb entwickelt, inklusive einer eigenen Platine mit Laderegelung, und ist modular erweiterbar, um zusätzliche Sensoren wie Näherungssensoren, LiDAR oder Kameras zu integrieren. Die Nutzung von Micro ROS erlaubt eine flexible Softwareentwicklung auf Basis eines STM32-Nucleo-Boards oder eines ESP-32. Damit entsteht eine vielseitige und erweiterbare Plattform zur praxisnahen Demonstration autonomer und redundanter Navigationslösungen im Indoor-Bereich.

## 1.3 STÜCKLISTE UND KOSTEN

Tabelle 1.1: Geschätzte Kosten des Prototyps

Bauteil	Einzelpreis [€]	Anzahl	Gesamtkosten [€]
Antrieb	30	4	120
Encoder	30	4	120
Mikrocontroller	90	1	90
Leiterplatte (PCB)	75	1	75
Rad	5	4	20
Sensor (Marvelmind)	650	1	650
Akku	50	1	50
Beliebige Sensoren	—	—	100
Zukaufteile	—	—	50
<b>Summe</b>			<b>1275</b>

## 1.4 PROJEKTMEILENSTEINE FÜR DIE MODULARE ROBOTERPLATTFORM

Das Projekt erstreckt sich über zwei Semester. Im ersten Semester liegt der Fokus auf der Entwicklung eines fahrfähigen Roboterprototyps mit modularer Erweiterbarkeit. Dies umfasst die Auswahl der Hardware und das Design der Platine. Im zweiten Semester steht die Erweiterung um Lokalisations- und Sensorfusionsmechanismen im Mittelpunkt. Hierbei sollen insbesondere das Marvelmind-Positionierungssystem sowie zusätzliche Sensoren eingebunden werden, um eine präzisere Navigation zu ermöglichen.

### 1.4.1 Semester 1: Entwicklung des fahrfähigen Prototyps

#### 1. Konzeptphase

- Literaturrecherche und Anforderungsdefinition
- Entwicklung eines modularen Hardwarekonzepts
- Grobkonzept für die Steuerung und Softwarearchitektur

#### 2. Hardwareauswahl und Komponentenbeschaffung

- Auswahl des Mikrocontrollers (z.B. STM32 Nucleo / Raspberry Pi / ESP32)

- Auswahl der Antriebskomponenten (Motoren, Motorsteuerung)
- Festlegung der modularen Erweiterungsschnittstellen
- Planung des Akkubetriebs inklusive Laderegelung
- Beschaffung der Komponenten

### 3. Platinendesign und Steuerung

- **Platinendesign:**

- Entwurf des Schaltplans für die Motorsteuerung und Sensoranbindung
- Layout der Leiterplatte und Fertigung
- Erste Tests der Steuerplatine

### 4. Gehäusedesign und Montage

- Entwicklung eines modularen Gehäuses (z.B. CAD-Modellierung, 3D-Druck)
- Integration der Steuerplatine, Motoren und Akkus in das Gehäuse

## 1.4.2 Semester 2: Integration von Lokalisierung und Sensorfusion

### 1. Grundlegende Funktionstests

- Test der Fahrfunktionen (Lenkung, Geschwindigkeit, Bremsverhalten)
- Evaluierung der modularen Erweiterbarkeit
- Optimierung der Softwaresteuerung

### 2. Integration des Marvelmind-Positionierungssystems

- Inbetriebnahme des Marvelmind-Systems
- Erste Tests zur Positionsbestimmung
- Verbindung mit dem Mikrocontroller und Integration in die Steuerung

### 3. Sensorfusion und Erweiterung

- Einbindung zusätzlicher Sensoren (Gyroskop, Radodometrie, Lidar, Kameras)
- Entwicklung und Implementierung von Sensorfusionsalgorithmen
- Verbesserung der Navigationsgenauigkeit durch optimierte Sensorkombination

### 4. Dokumentation und Abschlusspräsentation

- Detaillierte Dokumentation der Hard- und Software
- Erstellung von Bedienungs- und Aufbauanleitungen

# 2 Umsetzung

## 2.1 BAUTEILAUSWAHL

### 2.1.1 Mikrocontroller

Für das vorliegende Projekt wurde der ESP32-S3-WROOM-2-N32R16V ausgewählt, da er über eine USB-C-Schnittstelle verfügte, die ein schnelles und unkompliziertes Programmieren ermöglichte. Zudem bot der integrierte 2,4-GHz-WLAN-Chipsatz eine stabile Funkverbindung ohne zusätzlichen Hardwareaufwand. Der Mikrocontroller unterstützte nativ micro-ROS, was für die spätere Einbindung in ROS-basierte Systeme erforderlich war. Abschließend stellte die große Zahl frei verfügbarer GPIO-Pins sicher, dass sämtliche Sensoren direkt angeschlossen werden konnten.

### 2.1.2 Batterie

Für unser Projekt nutzen wir einen 3,7V 6000mAh LiPo-Akku, der eine lange Laufzeit garantiert. Die 3C-Entladerate ist für unsere Anwendung, die unter 1A benötigt, mehr als ausreichend und bietet hohe Sicherheitsreserven. Zudem schützt eine integrierte Platine den Akku zuverlässig vor Überladung, Tiefentladung und Kurzschluss.

### 2.1.3 Motor

Wir haben diesen Mikro-Metallgetriebemotor gewählt, weil seine optimale Drehzahl von 310 U/min und seine kompakte Größe perfekt zu unserem kleinen Design passen. Er läuft effizient mit 6V und hat einen geringen Stromverbrauch, ideal für batteriebetriebene Projekte. Der integrierte Encoder liefert präzises Feedback zur Drehzahl (7250 Impulse/Umdrehung).

### 2.1.4 Kabelloses laden

Drahtloses Lademodul 5V/1A: Komfortable Erweiterung Wir haben uns für dieses drahtlose Lademodul entschieden, da es uns eine Ausgangsleistung von 5V und 1A liefert, was ideal zum Laden unserer Batterie ist. Es dient uns als praktische Erweiterung, um unser Gerät bequem in einer Ladestation aufladen zu können. Das Modul nutzt resonante magnetische Kopplung für eine stabile Energieübertragung und erreicht dabei eine beeindruckende Effizienz von 90%.

### 2.1.5 Time of Flight

Wir setzen den VL53L3CX Time-of-Flight (ToF) Distanzsensor ein, um Kollisionen zu verhindern. Er bietet einen zuverlässigen Erfassungsbereich von 25 bis 3000 mm und hat einen Blickwinkel von 25 Grad.

### 2.1.6 Farbsensor

Wir nutzen den Gravity TCS34725 RGB Farbsensor zur Linienverfolgung. Dieser I2C-Sensor erfasst Farben und liefert RGB-Werte. Dank hoher Empfindlichkeit, breitem Dynamikbereich und IR-Sperrfilter funktioniert er gut bei unterschiedlichen Lichtverhältnissen. Vier integrierte LEDs ermöglichen den Betrieb auch ohne externe Beleuchtung.

### 2.1.7 Marvelmind

Wir nutzen das Starter Set Super-MP-3D für die Simulation eines Rasenmähroboters. Seine hochauflösende Indoor-Positionierung mit  $\pm 2\text{cm}$  Genauigkeit ist entscheidend, um die präzise Navigation unseres simulierten Mähers nachzubilden. Basierend auf Super-Beacons, verfolgt es die genaue Position des Roboters, erstellt virtuelle Mähzonen (Geofencing) und überwacht diese. Diese Genauigkeit übertrifft herkömmliche UWB- und BLE-Systeme um ein Vielfaches und ist perfekt, um autonome Bewegung und Bahnen im Detail zu testen.

## 2.2 GRUNDSCHALTUNGEN

### 2.2.1 Linearregler (LDO) – TLV767DRV

Der **TLV767DRV** ist ein Low-Dropout-Linearregler, der aus einer höheren, eventuell schwankenden Eingangsspannung eine konstante Ausgangsspannung bereitstellt. Durch die geringe Dropout-Spannung kann der Regler auch bei niedrigen Spannungsdifferenzen effizient arbeiten. Typische Einsatzbereiche sind rauscharme Spannungsversorgungen für empfindliche analoge und digitale Schaltungen. Der Regler überzeugt durch eine gute Transientenantwort und hohe Ausgangsstabilität. Auf

der Platine wird er dazu verwendet, die USB-Spannungsversorgung für den Microcontroller auf 3,3 V herunterzusetzen.

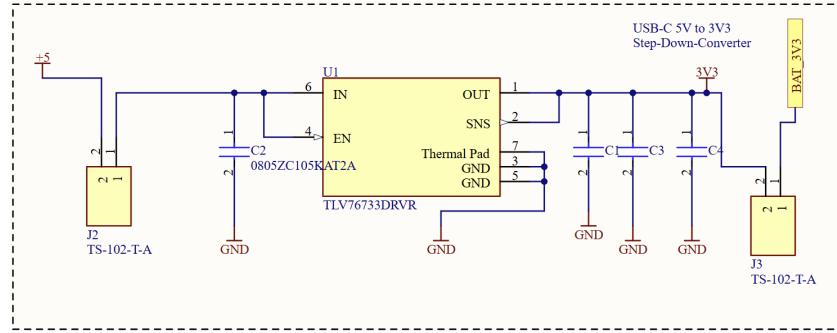


Abbildung 2.1: Linearregler LDO

### 2.2.2 Abwärts-/Aufwärtswandler – TPS63070RNMT

Der **TPS63070RNMT** ist ein Buck-Boost-Schaltregler, der sowohl höhere als auch niedrigere Ausgangsspannungen im Vergleich zur Eingangsspannung erzeugen kann. Diese Eigenschaft macht ihn ideal für Systeme mit variabler Versorgungsspannung, wie z. B. bei Batteriebetrieb. Er liefert bis zu 2 A Ausgangstrom, besitzt einen hohen Wirkungsgrad und unterstützt einen Energiesparmodus bei geringer Last. Integrierte Schutzfunktionen wie Übertemperatur- und Überspannungsschutz erhöhen die Betriebssicherheit. Auf der Platine wird er dazu verwendet, die Batteriespannung zuverlässig auf 3,3 V hinauf- bzw. herunterzuregeln.

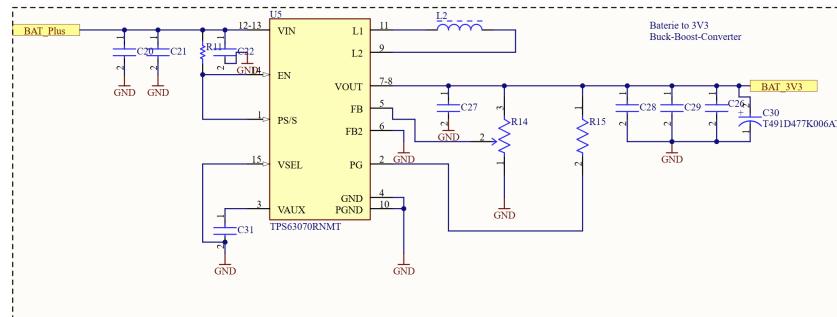


Abbildung 2.2: Battery Buck/Boost Converter

### 2.2.3 Lademanagement-IC – MCP73831T-2ACI/OT

Der **MCP73831T-2ACI/OT** ist ein kompakter Lademanagement-Controller für Einzelzellen-Lithium-Ionen- und Lithium-Polymer-Akkus. Er nutzt einen Konstantstrom-/Konstantspannungs-Algorithmus (CC/CV) und integriert Funktionen wie Strommessung und einstellbaren Ladestrom (bis 500 mA). Die Ladespannung ist auf 4,2 V geregelt. Weitere Merkmale sind thermische Schutzfunktionen, Verpolungsschutz und ein Statusausgang zur Ladezustandsanzeige.

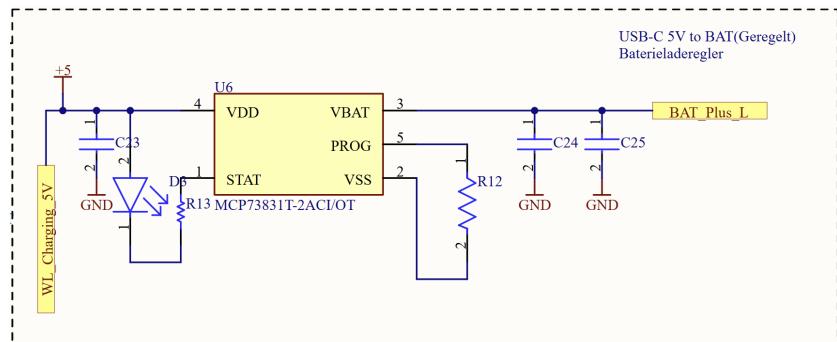


Abbildung 2.3: Lademanagement IC

### 2.2.4 Motortreiber – DRV8835DSSR

Der **DRV8835DSSR** ist ein dualer H-Brücken-Treiber zur Steuerung von zwei DC-Motoren oder einem Schrittmotor. Er unterstützt bidirektionalen Betrieb und kann Motoren mit bis zu 1,5 A (RMS) pro H-Brücke antreiben. Der Baustein verfügt über Schutzfunktionen gegen Überstrom, Kurzschluss, Unterspannung und Übertemperatur. Er kann im IN/IN- oder Phase/Enable-Modus betrieben werden und ist durch seine kompakte Bauform für mobile und platzkritische Anwendungen geeignet.

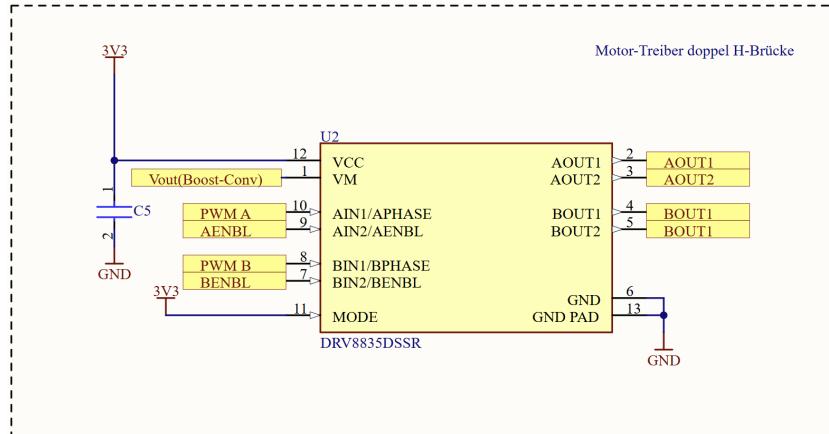


Abbildung 2.4: Motortreiber Doppel-H-Brücke

### 2.2.5 Aufwärtswandler – TPS61088RHLT

Der **TPS61088RHLT** ist ein synchroner Boost-Konverter, der eine höhere Ausgangs- als Eingangsspannung bereitstellt. Mit bis zu 10 A Schaltstrom bietet er hohe Leistungsreserven und verfügt über umfangreiche Schutzmechanismen. Der Baustein zeichnet sich durch hohe Effizienz und zuverlässigen Betrieb bei hohen Lasten aus. Auf der Platine wird er dazu genutzt, die Batteriespannung auf 6 V für den Motortreiber zu boosten.

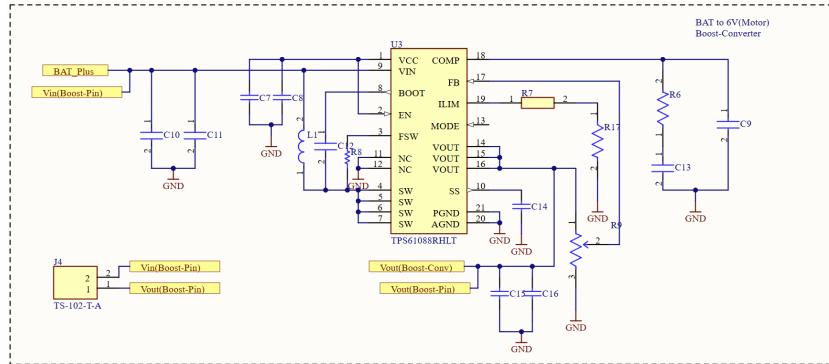


Abbildung 2.5: Boost Converter

## 2.3 GESAMTSCHALTUNG

Im Schaltplan erkennt man, dass die 5 V vom USB-C-Port zunächst sowohl zum Ladebaustein für die Batterie als auch in einen Step-Down-Regler laufen, der daraus 3,3 V erzeugt. Parallel dazu führt die wiederaufladbare Batterie nach dem Laden über einen eigenen Buck-Boost-Wandler ebenfalls eine 3,3-V-Spannung bereit. Diese beiden 3,3-V-Ausgänge sind direkt auf den gemeinsamen 3,3-V-Versorgungsbus gelegt – weil beide Regler genau auf dieselbe Spannung eingestellt sind, können sie einfach zusammengeführt werden: Die gerade höhere Quelle versorgt den Bus, die andere bleibt inaktiv.

Der USB-Teil des Designs übernimmt also zwei Aufgaben gleichzeitig: Einerseits speist er über den Step-Down-Regler das System sofort mit 3,3 V, andererseits lädt er im Hintergrund die Batterie

auf. Sobald die USB-Spannung wegfällt, übernimmt nahtlos der von der Batterie gespeiste Wandler die 3,3-V-Versorgung, ohne dass eine gesonderte Umschaltelektronik nötig wäre. Alle nachfolgenden Schaltungsteile – Sensoren, Steuerung und Anzeigen – greifen stets auf diesen einen 3,3-V-Bus zu und benötigen keine eigene Logik für die Quelle.

Die Datenleitungen von der USB-Schnittstelle werden über einen all in one ESD-Schutz IC geführt, um elektrostatischen Entladung und Überspannungsspitzen zu vermeiden, ohne die Signalqualität zu beeinträchtigen.

Für den ESP32-S3 wurden außerdem die Mindestanforderungen zur sicheren Inbetriebnahme umgesetzt: Ein dedizierter Reset-Taster setzt den Mikrocontroller zuverlässig zurück, während ein separater Boot-Taster das Starten im Programmier- oder Bootloader-Modus ermöglicht. So ist jederzeit sicher gestellt, dass der ESP32-S3 wie vorgesehen neu gestartet oder ins Firmware-Update versetzt werden kann.

Zur Anzeige des Systemzustands wurden zwei Status-LEDs realisiert:

- Betriebs-LED: Leuchtet dauerhaft, sobald das 3,3 V-Netz – entweder von USB oder vom Akku – aktiv ist, und signalisiert damit, dass das Gerät eingeschaltet und betriebsbereit ist.
- Charge-LED: Leuchtet während des Ladevorgangs der Batterie auf und erlischt, sobald der Ladebaustein das Ende des Ladevorgangs erkennt und der Akku vollgeladen ist.

Zur flexiblen Anbindung externer Module und zur Konfiguration der Hardware stehen mehrere programmierbare und festverdrahtete Pins zur Verfügung: Ein frei programmierbarer Taster ermöglicht individuelle Eingaben oder das Auslösen vordefinierter Funktionen. Zwei separate I<sup>2</sup>C-Leitungen wurden herausgeführt, um bei Bedarf Redundanz oder unabhängige Sensor- und Aktormodule parallel betreiben zu können. Die Motorausgänge sind direkt auf Pins gelötet, ebenso wie die UART-Schnittstelle für serielle Kommunikation und zwei weitere frei belegbare GPIO-Pins.

Für optionale Hardware-Modi wurden mehrere kleine Jumper („Brücken“) integriert: Sie erlauben das Umgehen des Motortreiber-Boosters, das gezielte Trennen oder Verbinden der 3,3-V-LDO-Eingangs- und Ausgangsseite sowie weitere Anpassungen. Schließlich existieren spezielle Pins für einen externen Ein-/Ausschalter, der das gesamte Batterie-Versorgungsnetz komfortabel schaltet.

## 2.4 ROUTING UND PLATINE

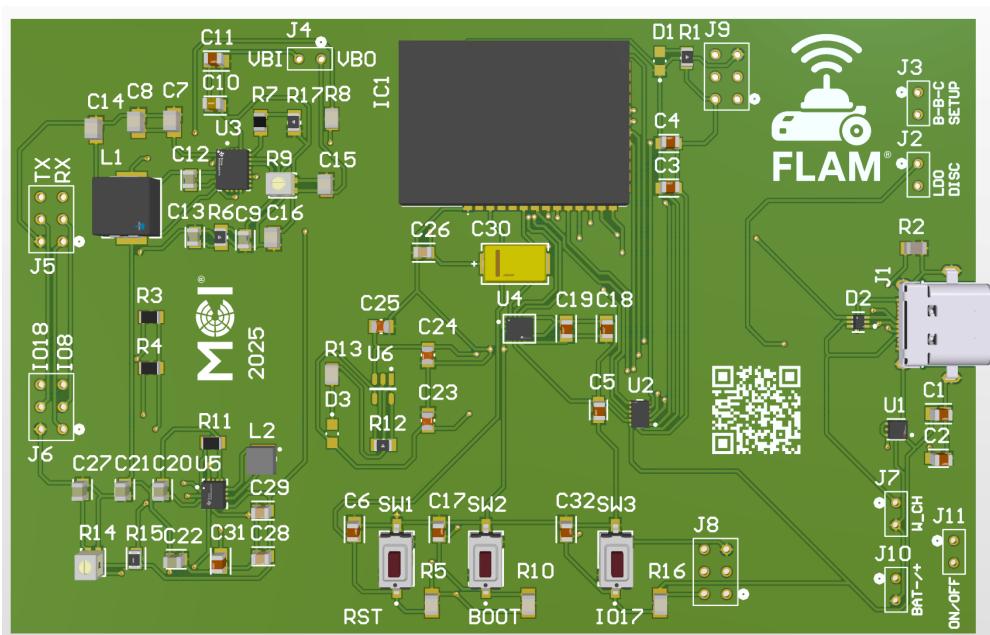


Abbildung 2.6: Oberansicht der bestückten Leiterplatte

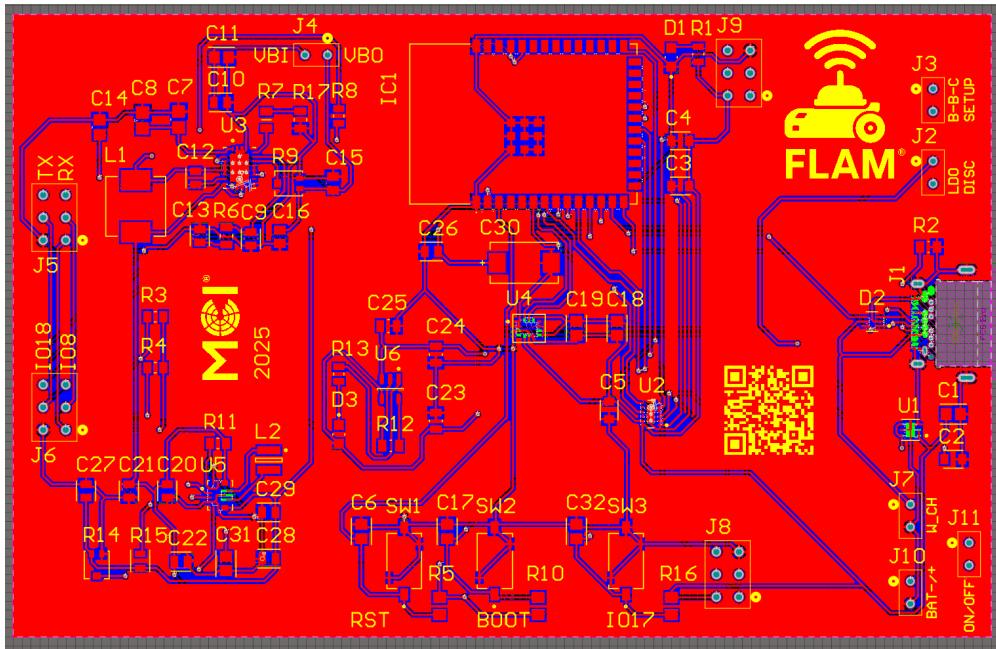


Abbildung 2.7: Leiterplattenlayout ohne Bestückung zur Darstellung der Leiterbahnen

Die Leiterplatte wurde zweilagig ausgeführt, mit kompakter Bauteilplatzierung und funktional strukturierter Anordnung. Kondensatoren wurden gezielt nahe an ICs platziert, wo dies für die Stabilität kritisch war.

Für den USB-C-Anschluss, der durch die Platine führt, war ein Ausschnitt notwendig, um mechanisch genug Platz zu schaffen. Sensoranschlüsse befinden sich bewusst weiter vorne auf der Platine, während die Motoranschlüsse rückseitig in der Nähe der Motorenposition liegen.

Ein QR-Code auf der Platine verlinkt direkt zur Projekt-Repository. Außerdem wurde ein individuell gestaltetes Logo mit den Initialen der Projektmitarbeiter integriert, um dem Board eine persönliche Note zu geben.

Abbildung 2.6 zeigt die fertige Platine mit bestückten Bauteilen von oben. In Abbildung 2.7 ist das Leiterplattenlayout ohne Bestückung dargestellt, wodurch die Leiterbahnen und das Routing klar erkennbar sind.

## 2.5 MECHANISCHE KONSTRUKTION

Der Reifen ( 2.8) besteht aus 3D-gedrucktem TPU A70, einem besonders weichen thermoplastischen Polyurethan, das speziell für maximale Oberflächenhaftung auf glatten und matten Glasoberflächen ausgewählt wurde. Die innere Struktur des Reifens orientiert sich an existierenden luftlosen Reifenmodellen, wie sie bereits in der Automobilindustrie Anwendung finden.

Die Felgen ( 2.9) sind so konzipiert, dass sie exakt auf das D-Profil der Motorachse passen. Zur besseren Sicherung der Felge auf der Achse ist im hinteren Teil ein Schmelzgewindeguss eingesetzt, durch den die Felge mittels einer Standardschraube oder einer Madenschraube fixiert wird. Der Zusammenbau von Felge und Reifen erfolgt durch einfaches Aufstecken beider Komponenten.

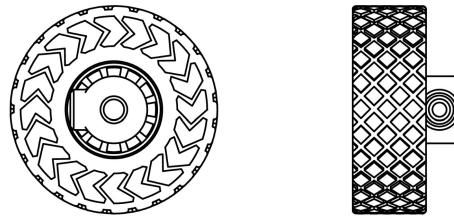


Abbildung 2.8: Reifen mit Felge in Seiten- und Vorderansicht



Abbildung 2.9: Bessere Ansicht der Felge

Die Baseplate ( 2.10) der Plattform ist vollständig aus PLA im 3D-Druckverfahren gefertigt. An der Unterseite befindet sich eine Vertiefung zur Aufnahme der Wireless-Charging-Spule. Auf der Oberseite sind zwei Führungselemente vorgesehen, in die die Platine passgenau eingelegt werden kann. Seitlich sind jeweils zwei Seitenplatten angebracht, die Ausschnitte und Befestigungslöcher zur Montage der Motoren enthalten. Hinter diesen befindet sich der Einbauplatz für die Batterie. An der Vorderseite ist ein Kugelrad aus Kunststoff montiert, das Kratzer auf empfindlichen Oberflächen minimiert. Die Form der Baseplate ist als abgerundetes Rechteck gestaltet, um zu verhindern, dass das Fahrzeug in den Ecken seines Operationsraumes stecken bleibt.

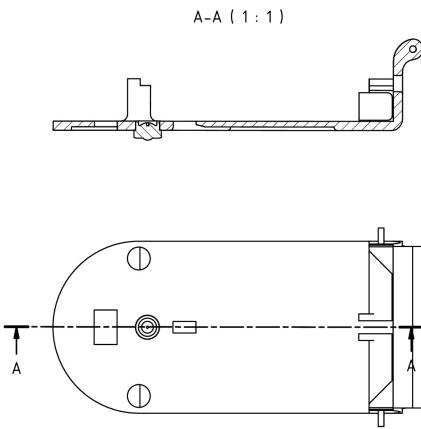


Abbildung 2.10: Seiten- und Oberansicht der Basisplatte

Die aufklappbare Abdeckung ist am unteren Ende leicht nach außen gebogen, was ein bündiges Schließen der Konstruktion ermöglicht. An der Oberseite befinden sich vier Montagelöcher zur Befestigung des Marvelmind-Ortungssystems, sodass die Ultraschallsignale aus allen Richtungen ungehindert empfangen und gesendet werden können. An der Vorderseite der Abdeckung ist ein Ausschnitt für den ToF-Sensor integriert.

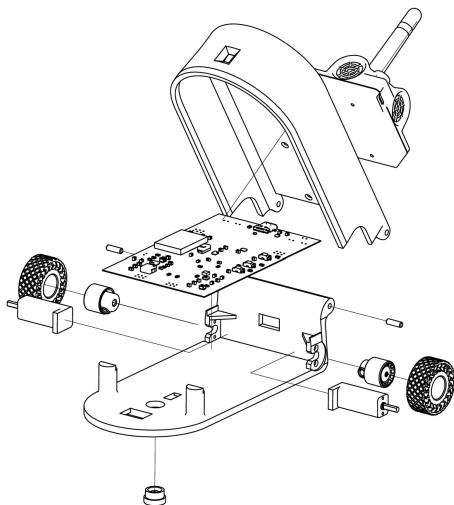


Abbildung 2.11: Explosionsdarstellung der gesamten Konstruktion

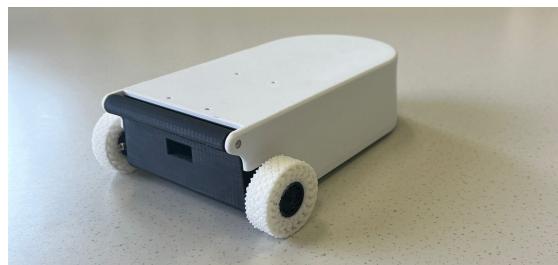


Abbildung 2.12: Prototyp der fertigen Konstruktion mit Platzhaltern für Motor und Kuglerad

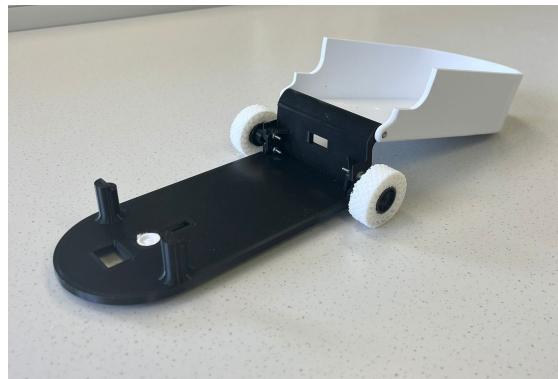


Abbildung 2.13: Prototyp der fertigen Konstruktion mit Platzhaltern für Motor und Kuglerad

## 2.6 PROTOTYPEN KOSTEN

Die in Tabelle 2.1 aufgeführten Positionen basieren auf aktuellen Marktwerten und verstehen sich als Nettopreise. Unter Berücksichtigung der Umsatzsteuer belaufen sich die Gesamtkosten auf 1292,87€. Da die Nettosumme von 1077,39€ den veranschlagten Kostenrahmen unterschreitet, konnte das vorgegebene Budget von ca. 1200€(netto) eingehalten werden. Die verbleibenden Mittel stehen somit zur Verfügung, um im kommenden Semester die Realisierung der Version V2 voranzutreiben.

Tabelle 2.1: Kostenaufstellung für die Mobile Robotics Education Cell (KST: Zentrum PRA)

Bauteil	Lieferant	Bestellnr.	Preis in €
Kugelrolle	RS-Components	262 618	13.92
E-Bauteile	Mouser	262 616	105.54
Marvelmind	Marvelmind	266 040	807.60
E-Bauteile	Robotshop Inc.	262 615	75.12
PCB+Stencil	Aisler	262 613	56.85
Motor Driver	Reichelt	266 856	18.36
		<b>Netto</b>	<b>1077,39</b>
		USt. (20 %)	215.48
		<b>Brutto</b>	<b>1292,87</b>

## 3 2. Semester

### 3.1 STAND NACH DEM ERSTEN SEMESTER

Im ersten Semester des Projekts lag der Fokus auf der Entwicklung des fahrfähigen Prototyps. Zunächst wurden die Anforderungen und Rahmenbedingungen definiert, um ein modulares Hardwarekonzept zu entwickeln. Daraufhin erfolgte die detaillierte Auswahl aller elektronischen Bauteile (wie ESP32-S3 Mikrocontroller, Batterie, Motoren) und die Erstellung einer Stückliste. Die Komponenten wurden bestellt, und parallel dazu das Platinendesign abgeschlossen, inklusive Schaltplan für Laderegelung und Motorsteuerung. Da die bestellten Zukaufteile nicht mehr rechtzeitig vor den Sommerferien ankamen und somit ein sofortiger Aufbau unmöglich war, verlagerte sich der Fokus auf die Mechanik: Die gesamte Gruppe entwarf und modellierte das modulare Gehäuse und Chassis der Plattform in CAD, einschließlich der Baseplate, der Abdeckung sowie speziell auf den Motor abgestimmter Räder und Felgen. Dies führte zu einem vollständigen, montierbaren Prototypen am Ende des Semesters, der nun bereit für die Integration der Sensorik ist.

### 3.2 EINFÜHRUNG

Mit Beginn des zweiten Semesters und der Ankunft der meisten bestellten Komponenten ist die Entwicklungsphase abgeschlossen, und der Fokus verlagert sich auf die physische Realisierung der Plattform. Der erste und kritischste Schritt ist die Bestückung der entworfenen Leiterplatte. Nach der Montage gilt es, die Hardware zum Leben zu erwecken: Dies beinhaltet die vorsichtige Inbetriebnahme der Versorgungsspannungen und Motortreiber. Die unmittelbar folgenden Aufgaben sind intensives Troubleshooting und Debugging sowohl auf Hardware- als auch auf Firmware-Ebene, um sicherzustellen, dass die grundlegenden Funktionen wie stabile Stromversorgung, Motoransteuerung und Mikrocontroller-Kommunikation fehlerfrei funktionieren. Nur auf dieser stabilen Basis kann im Anschluss mit dem Hauptziel des zweiten Semesters, der Integration des Marvelmind-Lokalisierungssystems und der Sensorfusionsalgorithmen, begonnen werden.

### 3.3 PLATINENBESTÜCKUNG

Nach dem Abschluss des Designs und der Ankunft der gefertigten Leiterplatte (PCB) und aller Bauteile wurde mit der physischen Bestückung begonnen. Dieser kritische Fertigungsschritt umfasste die folgenden Phasen:

- 1. Vorbereitung und Kontrolle:** Die Platine wurde zunächst optisch begutachtet und auf mögliche Mängel kontrolliert. Parallel dazu erfolgte die Überprüfung der elektronischen Bauteile auf Vollständigkeit und korrekte Spezifikation.

2. **Lötastenauftrag:** Eine Bestückungsschablone wurde präzise auf der Platine fixiert. Mittels eines Rakels wurde die Lötpaste möglichst gleichmäßig über die Schablone aufgetragen, um eine optimale und reproduzierbare Verteilung des Lötzinns auf allen Pads zu gewährleisten.
3. **Bauteilplatzierung:** Die verschiedenen SMD-Bauteile wurden anschließend präzise mit einem Vakuum-Pick-and-Place-Werkzeug auf die mit Lotpaste versehenen Pads platziert und ausgerichtet.
4. **Reflow-Löten:** Abschließend wurde die gesamte Platine für etwa fünf Minuten bei einer Temperatur von ca. 200°C in einem Reflow-Ofen (Backofen) erhitzt. Dieser Prozess ließ die Lötpaste schmelzen und sorgte für eine dauerhafte und zuverlässige elektrische und mechanische Verbindung der Bauteile mit der Platine.

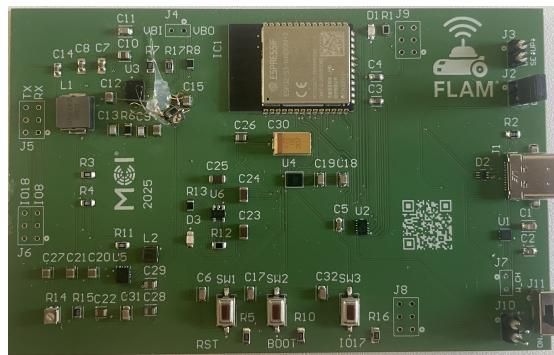


Abbildung 3.1: Bestückte Platine

## 3.4 INBETRIEBNAHME

Um die Platine sicher in Betrieb zunehmen ohne Komponenten zu gefährden wurde zu Beginn die Spannungsversorgung immer mit Strombegrenzung genutzt. Sollte ein Fehler vorliegen, konnte somit gewährleistet werden, dass die Spannung einbricht und die Spannungsversorgung nur mehr den begrenzten Strom speist und somit im Idealfall und bei richtig gesetzter Begrenzung keine Komponenten zu schaden kommen.

### 3.4.1 Jumper Konfiguration

### 3.4.2 Erstes Einschalten und Spannungs niveau einstellen

Für das erste Einschalten wurden alle Steckbrücken entfernt, welche die verschiedenen Teilschaltungen verbinden. Dabei wurde eine Konstantspannungsquelle mit Strombegrenzung genutzt um die Platine über die Batterieverbindung (J10) zu versorgen. Durch Kurzschließen des Ein-Aus-Schalters (J11) können somit die Spannungsregler bzw. die Spannungswandler, welche die Batteriespannung verarbeiten gesondert getestet werden.

Zu Beginn wurde der Buck-Boost Converter (U5) eingestellt. Dieser regelt die Batteriespannung zuverlässig auf 3.3V, unabhängig davon ob diese unter- oder oberhalb der 3.3V liegen. Zum Einstellen wurde das Potentiometer (R14) justiert bis am Steckverbinder 3 Pin 1 (J3-Pin1) 3.3V anliegen.

Danach wurde der Boost Converter (U3) in Betrieb genommen. Dafür wurde das Potentiometer (R9) so eingestellt, dass am Steckverbinder 4 Pin 1 (J4-Pin1) konstant 6V anliegen. Dabei fiel auf, dass das Spannungsniveau nicht konstant gehalten werden konnte. Nach weitere Rechere und mehrerer Versuche wurde der Chip U3 als defekt deklariert und ausgetauscht.

Mit diesen Spannungsniveaus eingestellt ist die Batterieseitige Spannungsversorgung überprüft und bereit durch eine Batterie ersetzt zu werden.

Des weiteren wurde die Versorgung über USB überprüft. Dazu wurde die Spannungsversorgung am Batterieanschluss (J10) entfernt und dafür der Jumper 2 (J2) eingesetzt. Über ein modifiziertes USB-C Kabel konnte die Platine Strombegrenzt mit 5V über den USB-C Anschluss (J1) gespeist werden.

Tabelle 3.1: Beschreibung der Jumper und Verbinder

Jumper/Verbinder	Erweiterte Beschreibung
J1	USB-C Anschluss (Buchse); dient der Datenübertragung und/oder der Stromversorgung des Roboters.
J2	Jumper, um 5V (meist von USB) mit dem Low-Dropout-Regler (LDO) zu verbinden. Dieser LDO erzeugt daraus die 3.3V Systemspannung, wenn das Gerät über USB versorgt wird.
J3	Jumper, um die 3.3V-Stromversorgungsschienen (Netzwerke) von USB und Batteriebetrieb miteinander zu verbinden oder zu trennen (z.B. zur Auswahl der Stromquelle).
J4	Jumper zur Überbrückung (Bypass) des Buck-Boost-Wandlers (Bauteil U3). Dieser Wandler regelt die Batteriespannung (z.B. von einer LiPo-Zelle) auf 6V hoch/runter.
J5	Jumper oder Header zur Konfiguration des Time-of-Flight (TOF) Abstandssensors; wählt vermutlich die Kommunikationsschnittstelle (z.B. zwischen I2C und UART).
J6	Anschluss-Header für einen Farbsensor (via I2C-Bus) sowie zwei zusätzliche, frei programmierbare GPIO-Pins (General Purpose Input/Output).
J7	Anschluss-Header für ein optionales Wireless-Charging-Modul (zum kabellosen Laden der Batterie).
J8	Anschluss-Header für Motor A, einschließlich der Signalleitungen für den zugehörigen Encoder (Drehgeber zur Positionserfassung).
J9	Anschluss-Header für Motor B, einschließlich der Signalleitungen für den zugehörigen Encoder (Drehgeber zur Positionserfassung).
J10	Hauptanschluss (z.B. JST-Stecker) für die Stromversorgung durch die Batterie oder einen Akkupack.
J11	Anschluss-Header (Stiftleiste) für einen (externen) physischen Ein-/Ausschalter zur Unterbrechung der Hauptstromversorgung.

Am Steckverbinder 3 (J3) wurden dann die passenden 3.3 V gemessen. Zudem leuchtete die Power-LED. Somit konnte verifiziert werden, dass die Versorgung über USB-C einwandfrei funktioniert.

### 3.4.3 ESP Inbetriebnahme

Für die Inbetriebnahme des ESP-32 wurde sowohl die Batterie als auch die USB-Schnittstelle verwendet. Dazu wurden die Jumper (J2, J3, J11) verbunden. Über die USB-C Schnittstelle wurde die Platine mit dem PC verbunden und konnte über die Arduino IDE erkannt werden. Es wurde ein Minimalbeispiel umgestzt um die korrekte Funktionsweise zu überprüfen. Um einen Esp32 zu flashen muss folgender Ablauf beachtet werden:

- **boot betätigen:** Drücken und Halten der BOOT-Taste. Diese Taste ist mit **GPIO0** verbunden und zieht diesen Pin auf **LOW**. Dies signalisiert dem Chip, dass er in den Download-Modus wechseln soll.
- **reset betätigen:** Drücken der RESET-Taste. Dies startet den Neustart-Prozess des ESP32, während der Zustand von **GPIO0** (**LOW**) aufrechterhalten wird.
- **reset lösen:** Loslassen der RESET-Taste. Der Chip startet nun und liest den Zustand der Boot-Pins. Da **GPIO0 = LOW** ist, wird der interne **ROM-Bootloader** aktiviert und in den Flash-Modus gewechselt.
- **boot lösen:** Loslassen der BOOT-Taste. Der Chip befindet sich bereits im Flash-Modus und wartet über UART auf Daten. Der Zustand von **GPIO0** spielt nun keine Rolle mehr.
- **upload:** Das Host-Programm beginnt mit der seriellen Übertragung und dem Schreiben der neuen Firmware in den Flash-Speicher des ESP32.

Das Minimalbeispiel umfasste den Test der seriellen Kommunikation über die PC-Verbindung mit einer Baudrate von 115200, den Test der I/O Schnittstelle und den Test der Motorencoder. Zusätzlich wurde auch der implementierte IMU-Sensor getestet, dabei konnten verifizierbare Werte über I2C ausgelesen werden.

### 3.4.4 Motor Inbetriebnahme

In Folge der Mikrocontroller-Inbetriebnahme wurde der Motortreiber (U2) in Betrieb genommen und getestet. Dafür wurde ein weiteres Minimalbeispiel geschrieben, dieses umfasste rein die Motoransteuerung mit konstanter Geschwindigkeit und eine serielle Bestätigung der Programmausführung. Während dieses Tests wurde die Spannungsversorgung (Boost-Converter U3) des Motortreibers übermäßig heiß und brannte durch. Infolgedessen konnte die Motoransteuerung nicht weiter getestet werden.

## 3.5 MECHANISCHE KONSTRUKTION

Das Gehäuse wurde großzügig dimensioniert, um nachträgliche Modifikationen und Wartungsarbeiten zu erleichtern. Während Länge und Breite primär durch die Platinenmaße bestimmt werden, orientiert sich die Bauhöhe an der optimalen Reichweite des Farbsensors. Hier musste das gesamte Gehäuse nach unten versetzt werden (stufe für Motoren), da aufgrund eines TPU engpasses die Reifen nichtmehr angepasst werden konnten.

Die Bohrungen zur Verbindung von Chassis und Karosserie wurden unterdimensioniert angelegt, um durch nachträgliches Aufbohren eine bessere Passgenauigkeit zu erzielen.

Wichtige Hinweise zur Montage: Da im Laborbestand keine M1,6-Schrauben verfügbar waren, wurde eine spezifische Halterung konstruiert, um die Motoren zu fixieren. Hier ist darauf zu achten, die Befestigungsschrauben der Motoren zu isolieren, um Kurzschlüsse auf der Platine zu vermeiden.

Folgende Optimierungen wurden in der aktuellen CAD-Datei bereits umgesetzt:

- Vergrößerung der Farbsensor-Aussparung für eine beidseitige Montageoption.
- Verschiebung der Aussparung für den USB-C-Port zur besseren Erreichbarkeit.

## 3.6 MÄNGEL

### Motortreiber U2

Bei der Inbetriebnahme des Motortreibers (U2) traten folgende Probleme auf:

- **Fehlerhafte Pin-Zuordnung:** Im Schaltplan waren die Pins PWM\_A und PWM\_B vertauscht beschriftet. Um dieses Problem zu beheben, wurde die Verschaltung mittels Fädeldrähten korrigiert. Dieser Fehler wurde als nicht kritisch eingestuft, da die Eingänge über Pulldown-Widerstände verfügen, wodurch undefinierte Zustände und damit verbundene Hardwareschäden verhindert wurden.
- **Bauteilausfall:** Im weiteren Verlauf trat ein Kurzschluss am Motortreiber auf. Als Ursache wurde die extrem kompakte Bauform des Bauteils identifiziert. Da es sich um den Chip mit den kleinsten Abmessungen und dem geringsten Pin-Abstand auf der Platine handelte, war der Lötprozess erschwert, was vermutlich zu einer fehlerhaften Verbindung führte. Die daraus resultierende übermäßige Stromaufnahme überlastete den vorgesetzten Boost-Converter (U3), welcher daraufhin thermisch zerstört wurde.

**Boost-Converter U3**

Nach einer detaillierten Fehleranalyse und Rücksprache mit Fachpersonal konnte die Ursache für die Überhitzung des Boost-Converters auf ein grundlegendes Layout-Problem zurückgeführt werden.

Unter dem Chip wurde lediglich eine isolierte Massefläche platziert, welche keine direkte Anbindung an das große Massenpolygon aufwies. Diese Fläche dient jedoch primär der Wärmeverteilung (*heat distribution*). Für eine ausreichende Wärmeabfuhr wäre eine Anbindung an eine möglichst große Kupferfläche zwingend erforderlich gewesen.

# 4 Auswertung

Nachfolgende Tabelle zeigt die Projektziele und deren Umsetzungsstand bei Projektende.

Kategorie	Soll	Ist	Offen	Status
<b>Mechanik</b>	Entwicklung eines modularen Gehäuses und fahrfähigen Chassis.	Gehäuse, Baseplate und Reifen (TPU) fertig konstruiert und montiert.	abgeschlossen	✓
<b>Mikrocontroller &amp; Logik</b>	Inbetriebnahme ESP32, Kommunikation und Basis-Software.	ESP32 geflasht, USB-Kommunikation und I2C funktionieren.	UART Schnittstelle mit Marvelmind testen	✓
<b>Spannungsversorgung (Low Power)</b>	Stabile 3,3V Versorgung für Logik und Sensoren.	3,3V Versorgung via USB und LDO stabil, LEDs leuchten.	eventuell Sperrdioden (z.B. Schottky Dioden) implementieren	✓
<b>Batteriekreis</b>	Stabile 3,3V Versorgung für die komplette Platine aus der Batterie.	Stabile 3,3 V über Buck-Boost-Converter. Batterie-Ladestruktur funktioniert.	eventuelle Einbindung von kabellosem Laden (Lieferung nicht angekommen)	✓
<b>Antrieb &amp; Leistung</b>	Ansteuerung der Motoren über 6 V Boost-Converter.	Boost-Converter (U3) durch thermische Überlast defekt, Motor dreht nicht.	Neudesign Motortreibereinheit	✗

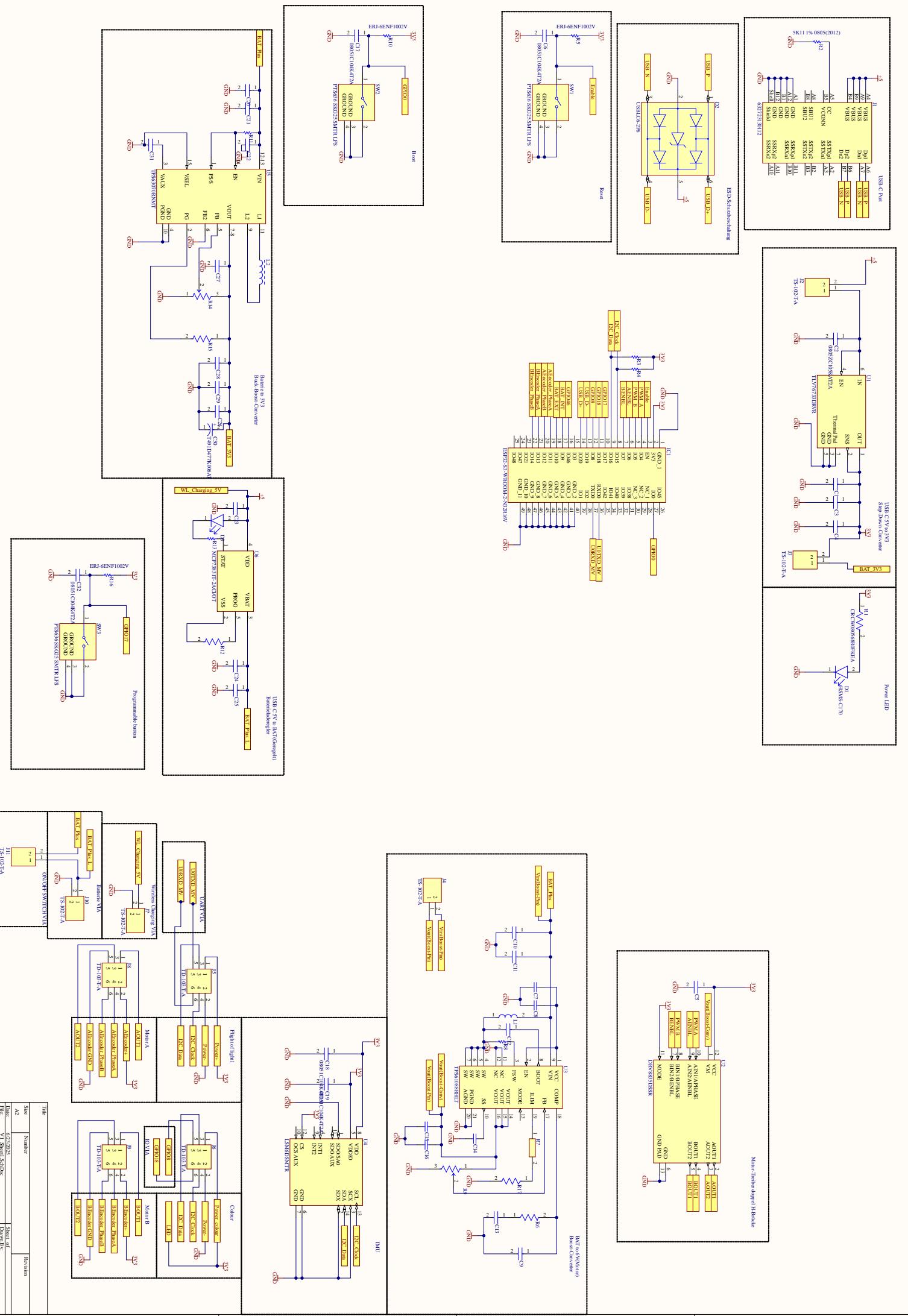
Tabelle 4.1: Soll-Ist-Vergleich der Projektziele

## 4.1 UPGRADES FÜR VERSION 2

Nach der Behebung der identifizierten Mängel und der Analyse der ersten Iteration lassen sich folgende Verbesserungen für Version 2 ableiten:

- **Anpassung der Motorsteuerung:** Um initiale technische Probleme zu umgehen und den Fahrbetrieb zuverlässig sicherzustellen, wird ein L9110S-Motortreiber in Form eines Entwicklerboards (Development Board) beschaffen und implementiert. Dies gewährleistet die grundlegende Mobilität des Systems auch vor der finalen Integration.
- **Miniaturisierung des Systems:** In Version 1 wurde sowohl auf der Platine als auch in der mechanischen Konstruktion bewusst zusätzlicher Raum für Analysezwecke vorgesehen. Das PCB-Layout und das Gehäuse bieten daher erhebliches Potenzial zur Komprimierung, um die angestrebte Zielgröße (analog einer Zigarettenenschachtel) zu erreichen.
- **Integration einer Status-LED:** Zur Vereinfachung des Debugging-Prozesses und für Funktionstests ist die Integration einer LED auf der Platine vorgesehen, welche direkt über den ESP32 angesteuert werden kann.
- **Optimierung der Reifenhafung:** Derzeit kommt für die Bereifung das im Labor verfügbare TPU mit einer Härte von 95 Shore zum Einsatz. Um die Traktion auf glatten Oberflächen (z. B. Glas) zu erhöhen, wird die Beschaffung und der Test eines weicheren TPU-Materials empfohlen.

- **Induktives Laden:** Obwohl das induktive Laden bereits in Version 1 konzeptionell für Platine und Gehäuse vorgesehen war, konnte die Hardware aufgrund von Lieferengpässen nicht implementiert werden. Die Nachrüstung ist für Version 2 geplant.
- **Batteriemanagement:** Der Ladezustand des Akkumulators wird bislang lediglich geschätzt oder durch externe Messgeräte ermittelt. Für Version 2 ist eine integrierte Überwachung des Batteriestatus vorzusehen.
- **Display-Integration:** Die Platine der ersten Version verfügt bereits über herausgeföhrte I<sup>2</sup>C-Schnittstellen. In der nächsten Iteration soll hierüber ein Display (Monitor) angeschlossen werden, um relevante Systemdaten wie Geschwindigkeit, Batteriestatus und Betriebsmodi direkt anzuzeigen.
- **Mechanische Fixierung der Sensoren:** Komponenten wie Farbsensor, ToF-Sensor und Batterie wurden in der ersten Version lediglich verklebt. Für Version 2 sind spezifische Halterungen innerhalb des Gehäuses zu konstruieren.
- **Implementierung variabler Fahrmodi:** Zusätzlich besteht die Möglichkeit, softwareseitig unterschiedliche Fahrmodi zu implementieren. Relevante Telemetriedaten sollen über das integrierte WLAN des ESP-Moduls auf einem Endgerät (z. B. Smartphone) ausgelesen werden können, um die Parameter für die Sensorfusion dynamisch anzupassen.



# Abbildungsverzeichnis

2.1	Linearregler LDO	4
2.2	Battery Buck/Boost Converter	4
2.3	Lademanagement IC	4
2.4	Motortreiber Doppel-H-Brücke	5
2.5	Boost Converter	5
2.6	Oberansicht der bestückten Leiterplatte	6
2.7	Leiterplattenlayout ohne Bestückung zur Darstellung der Leiterbahnen	7
2.8	Reifen mit Felge in Seiten- und Vorderansicht	8
2.9	Bessere Ansicht der Felge	8
2.10	Seiten- und Oberansicht der Basisplatte	8
2.11	Explosionsdarstellung der gesamten Konstruktion	9
2.12	Prototyp der fertigen Konstruktion mit Platzhaltern für Motor und Kuglerad	9
2.13	Prototyp der fertigen Konstruktion mit Platzhaltern für Motor und Kuglerad	9
3.1	Bestückte Platine	11

# Tabellenverzeichnis

1.1	Geschätzte Kosten des Prototyps . . . . .	1
2.1	Kostenaufstellung für die Mobile Robotics Education Cell (KST: Zentrum PRA) . . .	10
3.1	Beschreibung der Jumper und Verbinder . . . . .	12
4.1	Soll-Ist-Vergleich der Projektziele . . . . .	15

# Literatur