



Projektbeschreibung inverses Pendel

Bachelor: Mechatronik

4. Semester

Lehrveranstaltungsleiter: Matthias Gfall

Gruppe: BA-MECH-23

Verfasser: Stefan Hörtnagl, Lennart Götz, Maximilian Eckstein

30. Juni 2025

Einleitung

Im Rahmen der Lehrveranstaltung „Elektronische Produktentwicklung 1“ wurde mit der hardwareseitigen Entwicklung eines mobilen inversen Pendels begonnen. Das Ziel des Projekts besteht darin, einen zweirädrigen Roboter mit Dualantrieb zu entwickeln und zu bauen, der einen Pendelkörper in instabiler Gleichgewichtslage balanciert. [1]

Zentrales Thema dieses Semester war die Hardwarekonzeptionierung und -entwicklung. Ein Mikrocontroller erfasst die Sensordaten, verarbeitet sie und steuert entsprechend die Motoren, um das System zu stabilisieren.[1]

Basierend auf einer Literaturrecherche zur Regelung inverser Pendel wurde ein Grobkonzept erstellt, das die Auswahl geeigneter Komponenten (Sensoren, Aktoren und Mikrocontroller) sowie die Konstruktion, das Schaltungsdesign und die PCB-Entwicklung umfasst. Aufgrund der Komplexität des Themas sind mindestens zwei Hardware-Iterationen vorgesehen. Im ersten Teil der Lehrveranstaltung wurde zunächst die digitale Entwicklung umgesetzt.[1]

Nach Abschluss kann der entwickelte Aufbau als Demonstrator oder als Plattform für Laborversuche im Bereich der Regelungstechnik eingesetzt werden.[1]

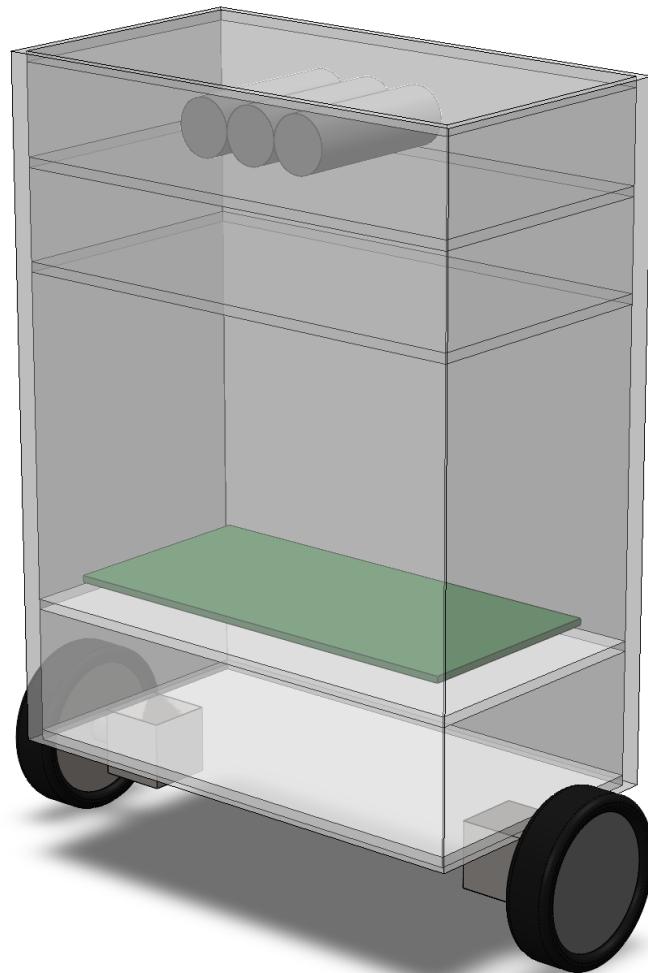


Abbildung 0.1: Entwurf inverses Pendel

Inhaltsverzeichnis

1 Projektzusammenfassung	1
2 Konzeptionierung	1
2.1 Energieversorgung	1
2.2 Antrieb	1
2.3 Kommunikationsschnittstellen	2
2.4 Sensorik	2
3 Schaltungsentwicklung	3
3.1 STM32 und Lagesensor	3
3.2 UART Kommunikation	5
3.3 Wifi-Schnittstelle	8
3.3.1 Allgemeine Spezifikationen	8
3.4 Schaltungsdetails des WiFi-Moduls (ESP8266)	9
3.4.1 Stromversorgung	10
3.4.2 Stromversorgung Antenne	10
3.4.3 Antenne	11
3.4.4 Flash	12
3.4.5 Oszillator	12
3.4.6 Backup Lösung	13
3.4.7 Kleinteile - Status LED	13
3.4.8 Kleinteile Allgemein	14
3.4.9 Breakout	15
3.5 Spannungsversorgung	16
3.6 Pin-Zuordnung und externe Schnittstellen	18
4 PCB Design	20
4.1 Stecker und Schnittstellen	20
4.2 STM32	20
4.3 UART Umschaltung	21
4.4 PWR Batterie	22
4.5 PWR USB und Power Mux	22
4.6 USB-UART Wandler	23
4.7 Leiterplattendesign für ein Standalone ESP8266EX-Modul	23
4.8 Backup Lösung	24
4.9 Bootloader Aktivierung	25

5 Mechanischer Aufbau	26
5.1 Rahmenstruktur	26
5.2 Ebene 0	26
5.3 Ebene 1	27
5.4 Ebene 2	27
5.5 Ebene 3	27
5.6 Ebene 4	27
5.7 Abmessungen	27
6 Kostenkalkulation	28
7 Pinbelegung und Funktionszuordnung	29
Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
Literaturverzeichnis	VII

1 Projektzusammenfassung

Gegenstand des Projekts ist der Bau eines Fahrgestells für den Einsatz in der Lehre. Das Gestell soll als inverses Pendel ausgeführt werden und sich – nach Implementierung eines geeigneten Regelalgoritmus – selbstständig in der oberen Ruhelage stabilisieren. Daraus ergeben sich die in Kapitel 2 beschriebenen Anforderungen. [1]

Obwohl das System in weiten Teilen digital konzipiert und in Form von Schaltplänen, PCB-Layouts und 3D-Modellen ausgearbeitet wurde, konnte aufgrund der hohen Komplexität und des begrenzten zeitlichen Rahmens bislang keine physische Umsetzung erfolgen. Die Realisierung des Prototyps ist jedoch für eine spätere Projektphase vorgesehen.[1]

2 Konzeptionierung

2.1 ENERGIEVERSORGUNG

Um einen sicheren Betrieb, schnelle Ladezeiten und den geforderten einfachen Akkuwechsel zu ermöglichen, kommt ein Lithium-Ionen Akku von Makita zum Einsatz. Dieser bietet eine hohe Energiedichte und beinhaltet bereits ein Battery-Management System welches Tiefenentladungen verhindert. Außerdem sind im Gestell zwei verschiedene Steckplätze für den Akku vorgesehen. Dies ermöglicht eine Variation des Schwerpunkts des Gesamtsystems.

Da die Spannung des Akkus unter Belastung signifikant absinken kann, wird ein DC-DC Wandler zur Entkopplung verwendet. Dadurch bleibt die Spannung am Motortreiber auch unter Last stabil. Die Regelung der Motorleistung wird dadurch ebenfalls vereinfacht.

2.2 ANTRIEB

Motorauslegung:

Für die Motorauswahl wurde zunächst ein maximaler Neigungswinkel definiert, bei dem das Pendel noch stabilisiert werden soll. Mithilfe dieses Neigungswinkels wurde ein maximales Kippmoment berechnet, das als Auslegungskriterium für das erforderliche Motordrehmoment herangezogen wurde.

Zur Berechnung des Kippmoments in MATLAB wurde der mechanische Aufbau des Pendels einschließlich seiner Dimensionen definiert. Sämtliche Bauteile wurden virtuell innerhalb des Systems positioniert. Mithilfe des Schwerpunktsatzes konnte der Gesamtschwerpunkt des Pendels approximiert werden. In Kombination mit dem vorgegebenen Neigungswinkel wurde daraus das maximale Kippmoment inklusive Sicherheitsreserve ermittelt.

Von Seiten des MCI wurde empfohlen, einen Gleichstrommotor mit Getriebe zu verwenden. Nachdem das erforderliche Drehmoment sowie die Betriebsspannung (12V) festgelegt worden waren, konnte ein geeigneter Motor ausgewählt werden. Neben dem Drehmoment war dabei auch eine ausreichend hohe Drehzahl entscheidend, um schnelle Regelbewegungen zu ermöglichen und das Pendel zuverlässig in der oberen Ruhelage auszuregeln.

Motor Treiber:

Um das Drehmoment der gewählten Motoren über den STM32 regeln zu können, wird eine geeignete Leistungselektronik benötigt. Zum Einsatz kommt eine H-Brücke, die ein PWM-Signal zur Ansteuerung der Motoren erzeugt. Durch Variation des Duty Cycles wird der Effektivwert der Motorspannung angepasst. Infolgedessen stellt sich ein mittlerer Motorstrom ein, der das resultierende Drehmoment bestimmt.

Es bestehen folgende Anforderungen an die H-Brücke:

Betriebsspannung: 12V

maximale Strombelastung: 7A

optional: Unterstützung von 2 Motoren

optional: Unterstützung von 3.3V Signalen des STM32

Anhand dieser Merkmale, wurde ein Dual-Channel Motor Treiber mit folgenden technischen Daten ausgewählt:

Betriebsspannung	6.5V - 37V
I_{max}	12A
I_{peak}	70A
Steuerspannung	3V - 5V

Tabelle 2.1: Technische Daten

2.3 KOMMUNIKATIONSSCHNITTSTELLEN

Eine Kernanforderung an das System ist die Unterstützung der MATLAB-Codegenerierung. Aus Simulink soll C-Code generiert werden, der anschließend über eine USB-Schnittstelle auf den Controller übertragen wird.

Zusätzlich sollen Telemetriedaten während des Betriebs drahtlos übertragen werden können. Dazu wurde ein ESP8266 Chip integriert, der Daten vom Controller empfängt und über WiFi versenden kann. Der ESP8266 soll - ebenso wie der Mikrocontroller - über USB (via UART) programmierbar sein.

Da der ESP8266 nur über eine einzige UART-Schnittstelle verfügt, wurde ein dualityer Signal-Multiplexer eingesetzt, um sowohl die Programmierung als auch die Kommunikation mit dem STM32 zu ermöglichen.

Aus diesen Anforderungen entwickelten wir folgende Topologie für die UART-Kommunikation:

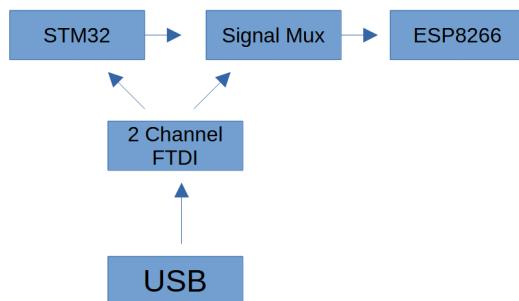


Abbildung 2.1: UART Topologie

2.4 SENSORIK

Um den Pendelkörper in der oberen Ruhelage ausregeln zu können, muss dessen Neigung erfasst werden. Ergänzend liefert die Messung der Beschleunigung wertvolle Informationen über die Systemdynamik.

Zur Erfassung von Neigung und Beschleunigung wurde der Inertialsensor LSM6DSOX ausgewählt. Dieser kombiniert einen 3-Achsen-Beschleunigungssensor mit einem 3-Achsen-Gyroskop und ermöglicht so eine präzise Erfassung der Bewegungs- und Lageinformationen bei gleichzeitig geringem Energieverbrauch.

3 Schaltungsentwicklung

3.1 STM32 UND LAGESENSOR

Um die Regelung des inversen Pendels zu implementieren, wird ein Mikrokontroller verwendet. Da der Steuerungs-Code automatisch aus MATLAB generiert werden soll, wurde hier auf entsprechende Kompatibilität geachtet. Um ausreichend Rechenleistung für zukünftige Erweiterungen (z.B. LiDAR) bereitzustellen, fiel die Wahl auf einen STM32H7A3. Zur Reduzierung der Komplexität wurde ein Gehäuse mit 64 Pins verwendet, um ein möglichst minimalistisches Leiterplattenlayout zu ermöglichen.

Um den stabilen Betrieb des STM32 zu ermöglichen, haben wir uns am Design-Guide von STMicroelectronics speziell für diese Chipserie orientiert. An geeigneter Stelle wurden Keramikkondensatoren platziert um die Spannungsversorgung zu stabilisieren. Außerdem wird einen externer Quarz mit 24Mhz verwendet, da das resultierende Clocksignal damit frequenzstabil ist.

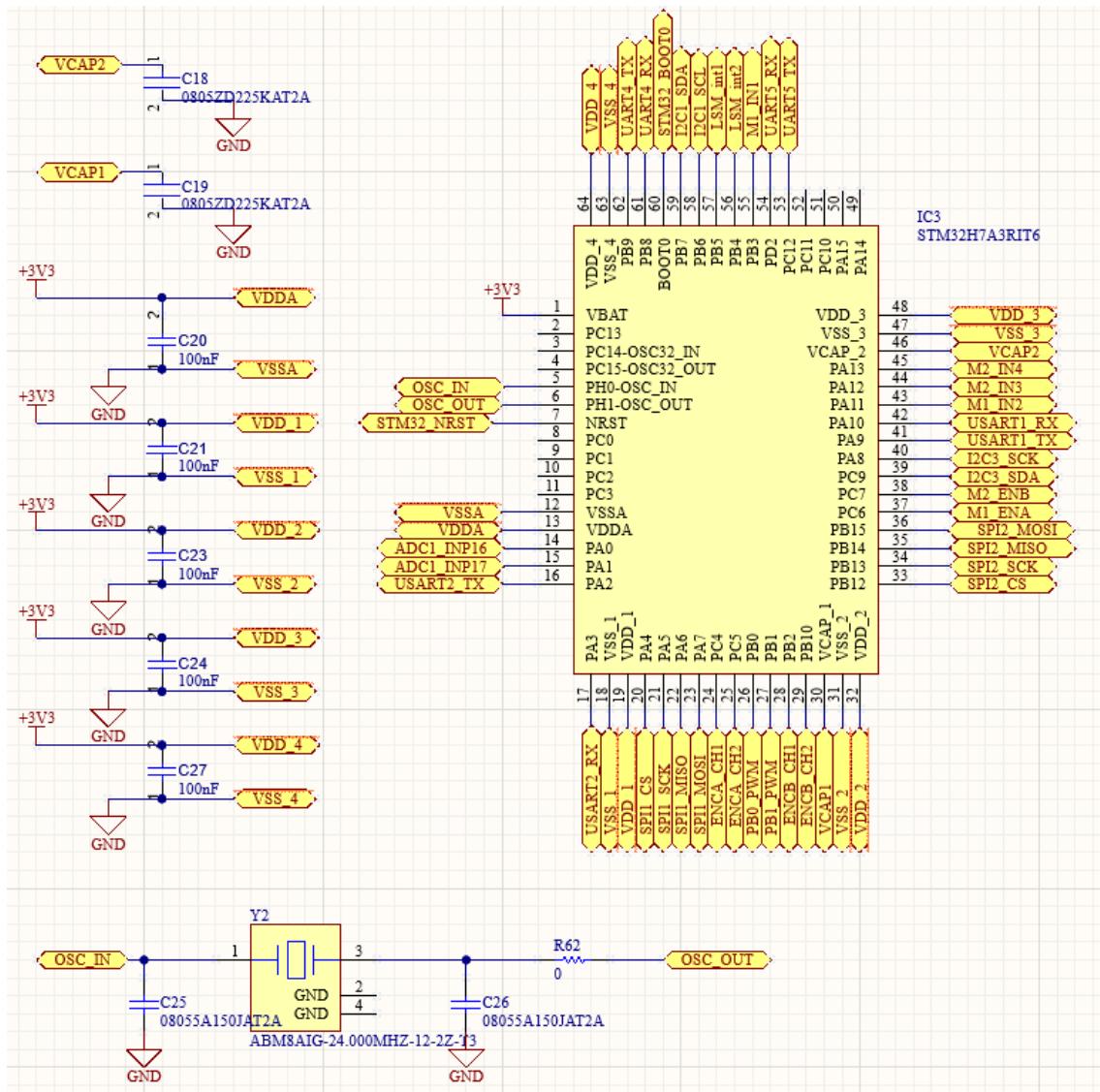


Abbildung 3.1: STM32 mit Quarz und Kondensatoren

Zur Erfassung der Neigung des Pendels sowie gegebenenfalls der Beschleunigung wurde ein geeigneter Sensor integriert. Die ursprünglich in Betracht gezogene und weit verbreitete MPU9250 wurde verworfen, da der Hersteller die Produktion eingestellt hat. Stattdessen kommt ein LSM6DSOX von STMicroelectronics zum Einsatz. Dieser Chip umfasst ein 3-Achsen-Gyroskop sowie Beschleunigungsmessung in 3 Achsen. Die Sensordaten werden über eine SPI Verbindung an den STM32 übermittelt. Da die Schnittstelle bidirectional ist, kann der Sensor auch über den STM32 konfiguriert werden. Beim Schaltungsdesign wurde sich erneut an der Dokumentation von STMicroelectronics orientiert. Zur Glättung der Versorgungsspannung wurden Keramikkondensatoren in unmittelbarer Nähe zum Sensor platziert.

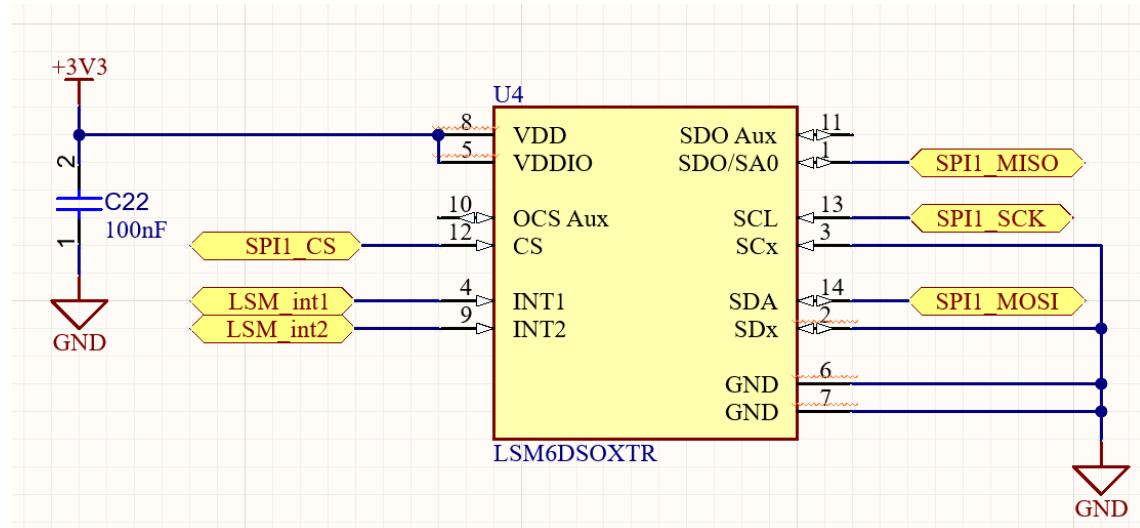


Abbildung 3.2: LSM6DSOX

3.2 UART KOMMUNIKATION

Basierend auf unserer UART-Topologie wurde die Umsetzung der UART-Kommunikation in drei Teile gegliedert: FTDI-Chip (USB-UART), UART-Signalumschaltung und Bootloader Aktivierung

FTDI-Chip:

Mithilfe des FTDI-Chips können sowohl der STM32 als auch der ESP8266 programmiert werden. Der Chip verfügt über zwei unabhängige UART-Kanäle, die vom Computer jeweils als separate virtuelle COM-Ports erkannt und genutzt werden können. Dadurch ist es gezielt möglich, den gewünschten Controller für die Programmierung auszuwählen.

Prinzipiell fungiert der FTDI-Chip als Signalumsetzer: Er wandelt das eingangsseitige USB-Signal vom PC in ein TTL/UART-Signal um, das von den Controllern verarbeitet werden kann.

Bei der Ausarbeitung der Beschaltung des FTDI-Chips wurde neben dem Design Guide des Herstellers auch ein bestehendes Breakout-Modul von FTDI herangezogen, das auf demselben Chip basiert.

Der EEPROM-Speicher ist nicht essenziell für den Betrieb, kann jedoch zur Ablage hilfreicher Informationen genutzt werden – beispielsweise eines benutzerdefinierten Gerätenamens oder der Konfiguration des COM-Ports. Dies erleichtert die Handhabung und Integration des Projekts insbesondere bei mehrfacher Verwendung an verschiedenen Computern.

Der FTDI ist so beschaltet, dass er nur aktiv wird, sobald ein Gerät am USB-C-Port angeschlossen ist.

Darüber hinaus verfügt der FTDI über einen externen Schwingkreis, um ein stabiles Taktsignal bereitzustellen, das für eine zuverlässige USB-Kommunikation gemäß Spezifikation erforderlich ist.

Kondensatoren und Ferritperlen dienen der Entkopplung sowie der Filterung hochfrequenter Störungen und verbessern dadurch die Signalqualität sowie die EMV-Eigenschaften der Schaltung.

Außerdem wurden vier Status-LEDs zur Anzeige der RX- und TX-Aktivität von Kanal 1 und 2 implementiert.

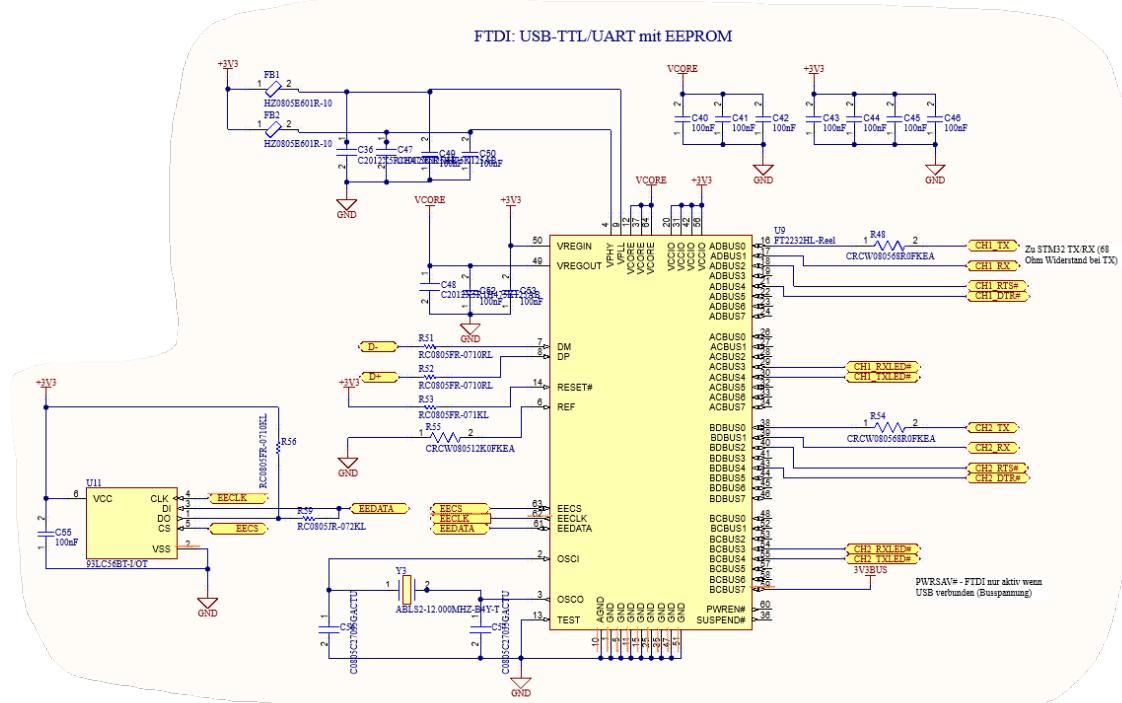


Abbildung 3.3: FTDI-Chip mit Peripherie

UART-Signalumschaltung:

Zur flexiblen Umschaltung der UART-Verbindung des ESP8266 zwischen STM32 und FTDI wurde ein analoger Multiplexer (TS5A23157) implementiert. Dieser ermöglicht, abhängig von den Steuerleitungen IN1 und IN2, die gezielte Verbindung der TX- und RX-Leitungen entweder zum Mikrocontroller (STM32) oder zum USB-UART-Konverter (FTDI CH2). Die Steuerung kann automatisch über die invertierte RTS-Leitung des FTDI erfolgen oder manuell per Schalter.

Schalter 1 legt fest, ob die Umschaltung automatisch (über die RTS-Leitung des FTDI) oder manuell erfolgt. Schalter 2 bestimmt – im manuellen Modus – welcher UART-Kanal mit dem ESP8266 verbunden wird.

Da der ESP8266 im Normalfall nur selten programmiert werden muss, ist die Standardverbindung über die NC-Kontakte (Normally Closed) mit dem STM32 vorgesehen.

Da jeweils einer der beiden UART-Pfade (STM32-ESP oder FTDI-ESP) inaktiv ist, wurden beide RX-Leitungen mit Pull-Up-Widerständen auf High gezogen, um unerwünschtes Floating zu vermeiden. Zusätzlich sind die UART-Leitungen zum ESP8266 mit 68-Ohm-Widerständen abgesichert, um Pegelkonflikte bei gleichzeitigem Anschluss zu vermeiden.

Für den Fall, dass die Programmierung des ESP8266 über die eigens entwickelte USB-UART-Schaltung nicht funktioniert, ermöglichen Jumper-Verbindungen den Anschluss eines externen Programmieradapters.

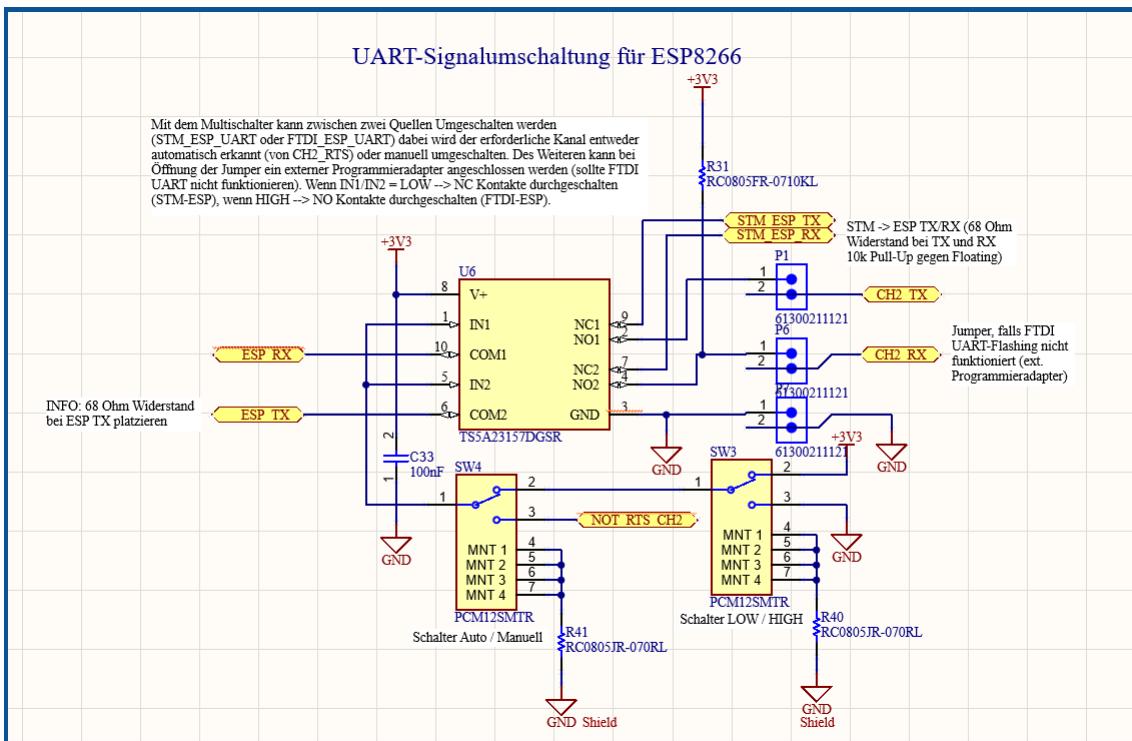


Abbildung 3.4: UART-Signalumschaltung

Bootloader Aktivierung:

Zur Aktivierung des Bootloaders für STM32 und ESP8266 wurde eine Schaltung entwickelt, die sowohl automatisch über die RTS- und DTR-Leitungen des FTDI-Chips als auch manuell über Taster ausgelöst werden kann. Die Signale RTS und DTR sind als active-LOW ausgeführt und werden über einen Inverterbaustein (SN74LVC14AD) sowie nachgeschaltete Open-Drain-MOSFETs entkoppelt, um Kurzschlüsse beim gleichzeitigen manuellen Betätigen der Taster zu vermeiden und die FTDI-Ausgänge zu schützen.

Da die Transistorschaltungen invertierend wirken, wird durch die zusätzliche Invertierung im IC das ursprüngliche logische Verhalten wiederhergestellt. Die Umschaltung erfolgt entweder automatisch über Softwaresteuerung durch den FTDI oder alternativ durch manuelle Betätigung der vier Taster.

Für den STM32 wird der Bootloader aktiviert, indem BOOT0 auf High und anschließend der Reset-Pin (NRST) kurzzeitig auf Low gezogen wird. Beim ESP8266 erfolgt die Aktivierung durch ein kurzes Herunterziehen des EN-Pins (Reset) bei gleichzeitigem Low-Pegel auf GPIO0. Beide Controller starten so im Flashmodus und können über die jeweilige UART-Schnittstelle programmiert werden. Über Jumper lassen sich die FTDI-Signale gezielt trennen, um beispielsweise einen externen Programmieradapter zu verwenden oder die automatische Umschaltung zu deaktivieren.

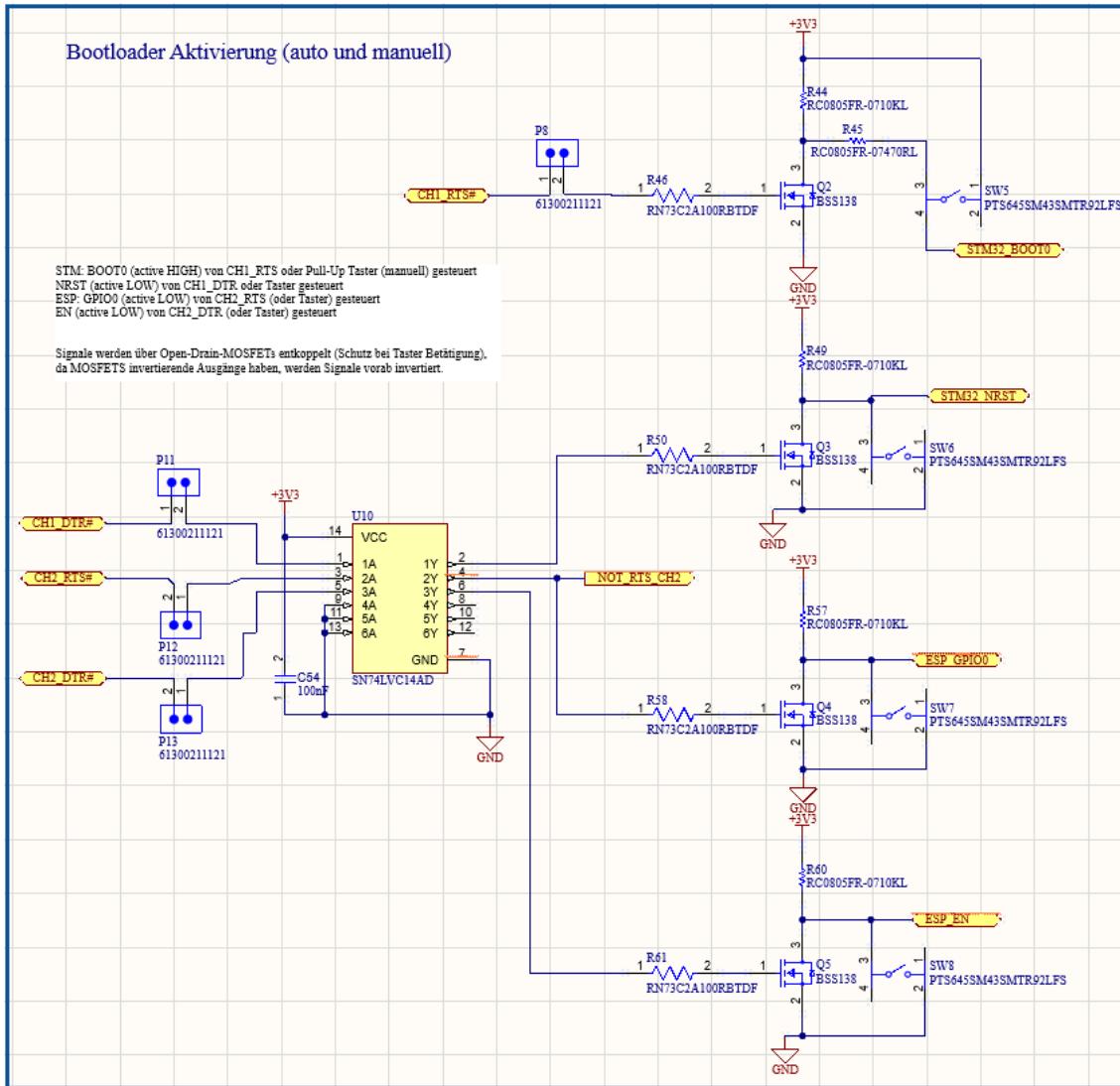


Abbildung 3.5: Bootloader-Schaltung

3.3 WIFI-SCHNITTSTELLE

Für die drahtlose Kommunikation des Mikrocontrollers wurde der ESP8266EX[2] WiFi-Chip gewählt. Er erfüllt alle Projektanforderungen und besitzt eine kompakte Bauweise. Aufgrund seiner weiten Verbreitung in vielen Anwendungen ist er eine solide Basis. Der ESP8266EX bietet dabei eine Vielzahl von Funktionen:

- IEEE 802.11 b/g/n-Standard
- 32-Bit RISC-Prozessor mit bis zu 160 MHz Taktfrequenz
- diverse digitale Schnittstellen wie GPIO, SPI, I²C, UART und PWM.

Er kann im Deep-Sleep-Modus mit $20\ \mu\text{A}$ im Ruhemodus gehalten werden und eignet sich dadurch besonders für batteriebetriebene und energieeffiziente IoT-Anwendungen.

Ein entscheidender Vorteil ist die Unterstützung durch eine große Community sowie die umfangreiche Dokumentation und Softwareentwicklungsumgebung (SDK), die eine schnelle und flexible Softwareentwicklung ermöglicht. Die WLAN-Kommunikation wird hauptsächlich dazu verwendet, Echtzeitdaten der Sensoren und des Systemzustands zu senden und auf einem entsprechenden Webserver Interface wiederzugeben. Aufgrund der selbst designten PCB-Antenne wurde eine vollumfängliche Backup-Lösung vorgesehen.

3.3.1 Allgemeine Spezifikationen

- Prozessor: Tensilica L106, 32-bit RISC, bis 160 MHz
- Speicher: Externe SPI-Flash bis 16 MB, < 50 kB SRAM nutzbar
- Wi-Fi: 802.11 b/g/n (2.4 GHz)
- Datenrate: Bis 72.2 Mbps (802.11n)
- Schnittstellen: GPIO, UART, SPI, I²C, I²S, PWM, ADC , IR
- Betriebsspannung: 2.5 V – 3.6 V
- Stromverbrauch:
 - Aktiv: ca. 80 mA
 - Light-Sleep: 0.9 mA
 - Deep-Sleep: $20\ \mu\text{A}$

Die herstellerspezifischen Hardware-Design-Rules wurden bei der Entwicklung des Schaltplans und des PCB-Designs berücksichtigt und eingehalten.

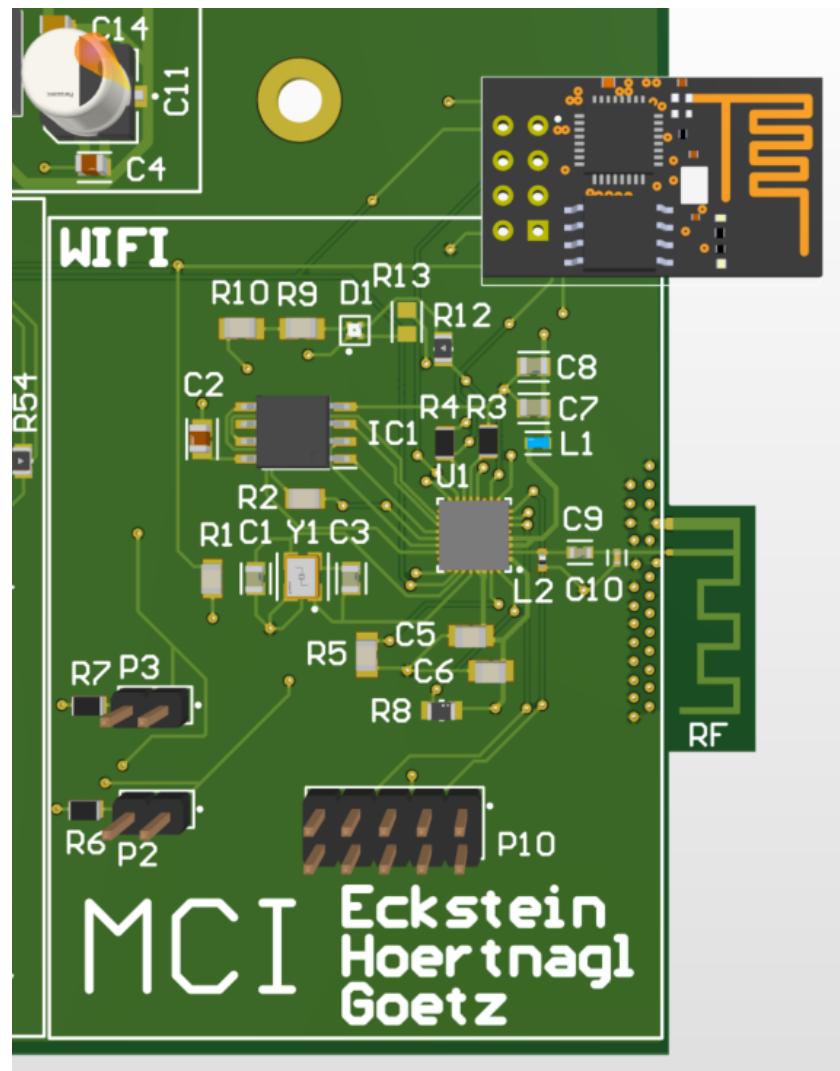


Abbildung 3.6: PCB Abschnitt WiFi

3.4 SCHALTUNGSDETAILS DES WIFI-MODULS (ESP8266)

Die Peripherie wurde hierbei in die folgenden Sektionen am Schaltplan unterteilt:

- Spannungsversorgung
- Antenne und Stromversorung Antenne
- Flash
- Oszillator
- Backup Modul
- Kleinteile

3.4.1 Stromversorgung

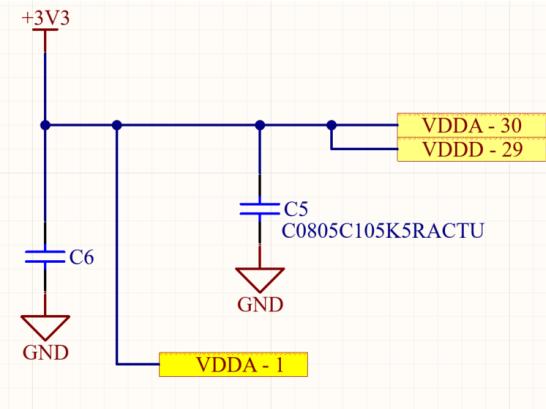


Abbildung 3.7: Stromversorgung ESP8266

Die Stromversorgung wurde gemäß den Design-Richtlinien[3] umgesetzt. Dazu gehören zwei $1\mu F$ -Kondensatoren, die zur Filterung hochfrequenter Störungen beitragen. Die Versorgung über VDDA30, 29 und 1 dient der Spannungsbereitstellung für die analogen Komponenten des ESP, die besonders empfindlich gegenüber Spannungsschwankungen sind. Darach wird eine störungsfreie Verwendung garantiert, ohne dass das Risiko von Paketverlusten oder unerwarteten Resets besteht.

3.4.2 Stromversorgung Antenne

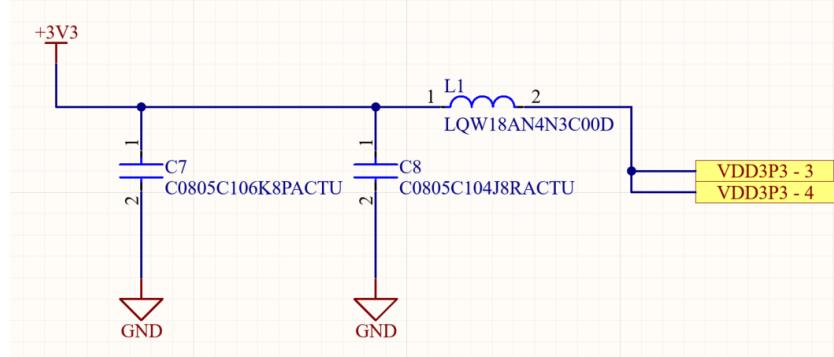


Abbildung 3.8: Stromversorgung Antenne

Diese Schaltung ist mit dem internen Power-Amplifier und dem Low-Noise-Amplifier verbunden und sorgt für die Versorgung des Antennenbereichs des Chips. Dabei fungiert der $10\mu\text{F}$ Kondensator (C7) als Energiespeicher, da während des Sendens ein hoher Stromverbrauch entstehen kann. Der Kondensator C8 dient zur Filterung von hochfrequenten Rückkopplungen. Die Schaltung wurde nahe den verwendeten Pins im PCB-Layout vorgesehen. Dadurch werden typische Probleme wie Rückkopplungen, Rauschen im RF-Signal oder Störungen im Betrieb vermieden.

3.4.3 Antenne

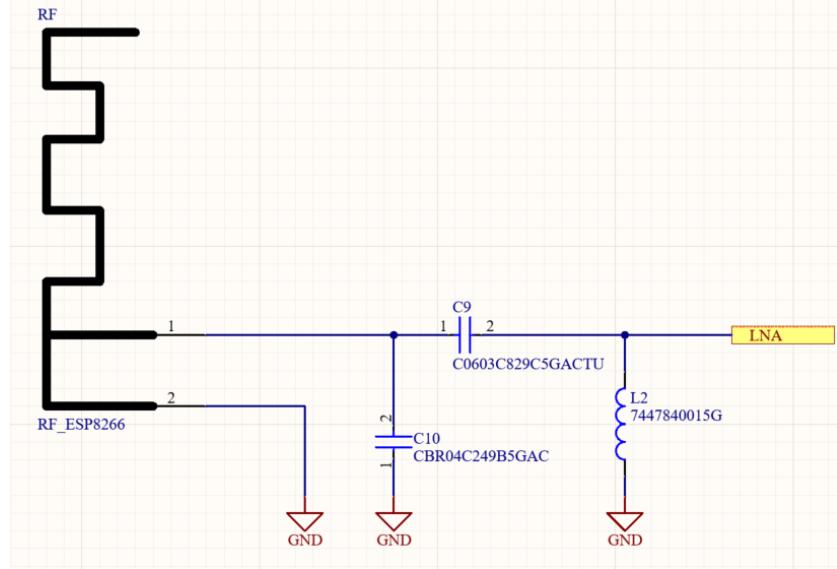


Abbildung 3.9: Antenne

Der RF-Ausgang des ESP8266EX hat eine charakteristische Impedanz von $39 + j6 \Omega$. Um eine verlustarme Leistungsübertragung zur Antenne sicherzustellen, ist eine Anpassung an 50Ω notwendig.

Zur Impedanzanpassung des RF-Ausgangs an die 50Ω Antenne wurde ein π -Matching-Netzwerk ausgelegt. Ziel war es, die komplexe Ausgangsimpedanz des LNA-Pins ($Z_s = 39 + j6 \Omega$) auf die Antennenimpedanz von $Z_L = 50 \Omega$ abzustimmen. Als Betriebsfrequenz wurde 2.45 GHz gewählt (Mittenfrequenz im 2.4 GHz-WLAN-Band).

Für das Matching wurde folgende Bauteilkombination verwendet:

- $C9 = 8,2 \text{ pF}$
- $L2 = 1,5 \text{ nH}$
- $C10 = 2,4 \text{ pF}$

Die Impedanzanpassung erfolgte mithilfe eines Matching-Rechners. Als Antenne wurde eine direkt im PCB integrierte Variante umgesetzt, die über definierte Leiterbahnstrukturen realisiert ist. Die Antennenfläche wurde gemäß den offiziellen Hardware Design Guidelines von Espressif konsequent von metallisierten Flächen und Signalleitungen freigehalten und durch GND-Vias abgeschirmt [3]. Dadurch wird eine hohe Sendeleistung sowie eine gute Empfangsempfindlichkeit im 2.4 GHz-Band sichergestellt.

3.4.4 Flash

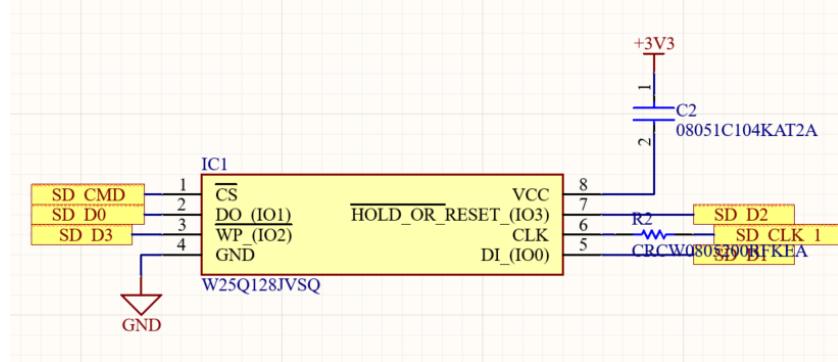


Abbildung 3.10: Flash Beschaltung ESP8266

Der ESP besitzt keinen internen Flash-Speicher, weshalb ein externer Flash-Speicher vorgesehen wurde. Die Kommunikation zwischen ESP und Flash erfolgt hierbei über die SPI-Schnittstelle. C2 (100nF) ist ein Bypass-Kondensator, wie in den Guidelines[3] empfohlen, um Spannungsschwankungen zu dämpfen. R2 (200Ω) in Serie mit der Takteleitung (CLK): Dies reduziert die Treiberstromspitzen und verhindert EMV-Probleme. Es ist darauf zu achten den Flash im richtigen Modus zu konfigurieren, damit die Firmware korrekt gelesen werden kann.

Der gewählte Flash bietet hierbei folgende Spezifikationen[4]:

- 128 MBit = 16 MB Speicher
- Standard, Dual und Quad SPI
- 2.7 - 3.6 V Betriebsspannung

3.4.5 Oszillator

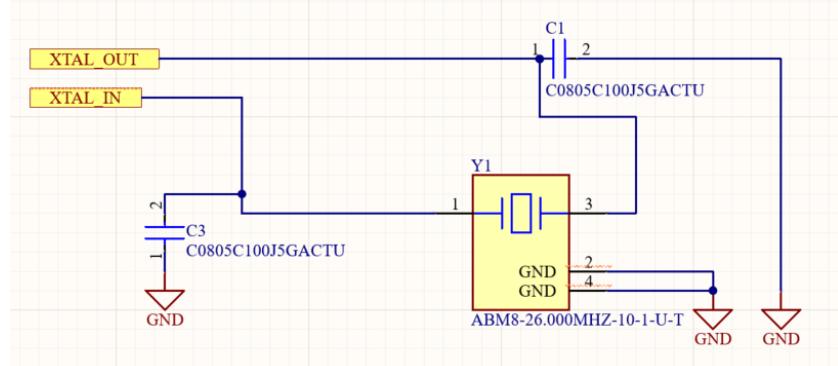


Abbildung 3.11: ESP Oszillator

Der ESP wird mit einer Frequenz von 26 MHz betrieben. Die Taktquelle hierfür ist der Oszillator[5]. Dieser ist notwendig, um die interne PLL des Chips zu takten und somit die Betriebsfrequenz des Prozessors und der WLAN-Funktionalität sicherzustellen. Bei der Verwendung des ESP Flash Download Tools muss die gewählte Frequenz korrekt eingestellt werden.

Der ESP8266EX unterstützt Kristallfrequenzen von 24 MHz, 26 MHz oder 40 MHz. Die Frequenzgenauigkeit des Quarzes liegt bei ± 10 ppm, was den Anforderungen für zuverlässige WLAN-Kommunikation entspricht.

Der Quarz ist mit den Pins XTAL_IN und XTAL_OUT des Controllers verbunden. Für eine korrekte Funktion werden zwei Kondensatoren (C1 und C3 mit 10 pF) gegen Masse geschaltet, die den Eingang und Ausgang des Oszillators stabilisieren.

3.4.6 Backup Lösung

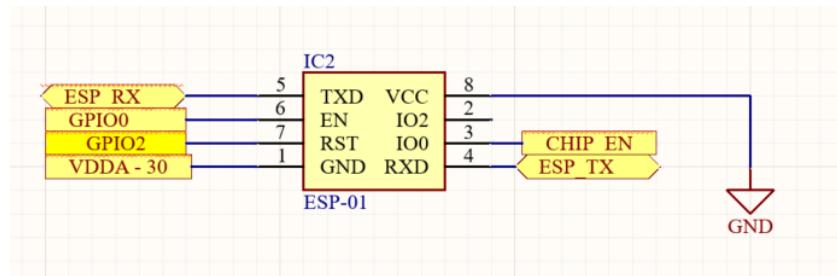


Abbildung 3.12: WiFi Backup

Die selbst entwickelte PCB-Antenne sowie die kritischen Anforderungen an die Kommunikation wurden durch eine Backup-Option berücksichtigt[6], die den Wechsel auf ein bewährtes, externes Modulsystem ermöglicht. Dafür wurden auf der Leiterplatte zusätzliche Footprints für alternative Komponenten vorgesehen, die mit dem bestehenden System kompatibel sind. Bei Bedarf kann der eingelötete ESP8266 entfernt und durch die Backup-Lösung ersetzt werden.

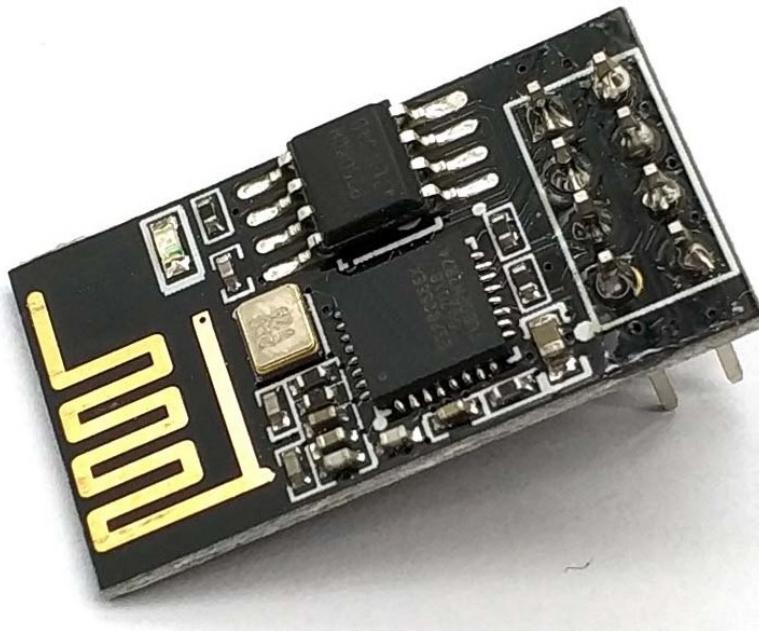


Abbildung 3.13: Backup - ESP01[6]

3.4.7 Kleinteile - Status LED

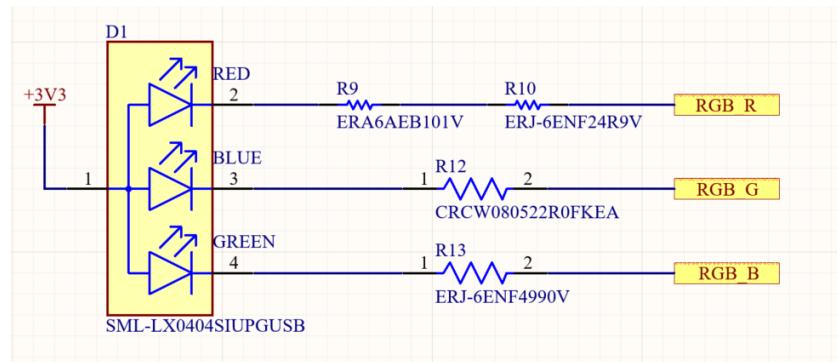


Abbildung 3.14: LED Schaltplan

Zur Visualisierung des aktuellen Systemzustands wurde eine mehrfarbige Status-LED integriert. Diese wird über frei verfügbare, PWM-fähige GPIOs des ESP8266 angesteuert und ermöglicht die softwareseitige Erzeugung beliebiger Farben durch additive Farbmischung. Beispielsweise könnte bei Fehlfunktionen ein rotes Licht, beim Senden von Daten ein blaues Signal angezeigt werden. Die zugehörigen Vorwiderstände wurden wie folgt berechnet:

$$R = \frac{V_{\text{Versorgung}} - V_{\text{LED}}}{I_{\text{LED}}}$$

Dabei ist:

- $V_{\text{Versorgung}}$: Betriebsspannung (z. B. 3,3 V)
- V_{LED} : Vorwart-Voltage der LED
- I_{LED} : gewünschter Strom (10 mA)

Dabei spiegeln die Werte der Tabelle das Datenblatt der LED wider.

LED-Farbe	V_{LED}	I_{LED}
Rot	195 V bis 2,15 V	10 mA
Grün	2,75 V bis 2,95 V	10 mA
Blau/Weiß	2,7 V bis 2,85 V	10 mA

Berechnung der Widerstände:

$$R_{\text{rot}} = \frac{3.3 \text{ V} - 2.05 \text{ V}}{0.01 \text{ A}} = 125 \Omega$$

$$R_{\text{grün}} = \frac{3.3 \text{ V} - 2.85 \text{ V}}{0.01 \text{ A}} = 45 \Omega$$

$$R_{\text{blau/weiß}} = \frac{3.3 \text{ V} - 2.8 \text{ V}}{0.01 \text{ A}} = 50 \Omega$$

3.4.8 Kleinteile Allgemein

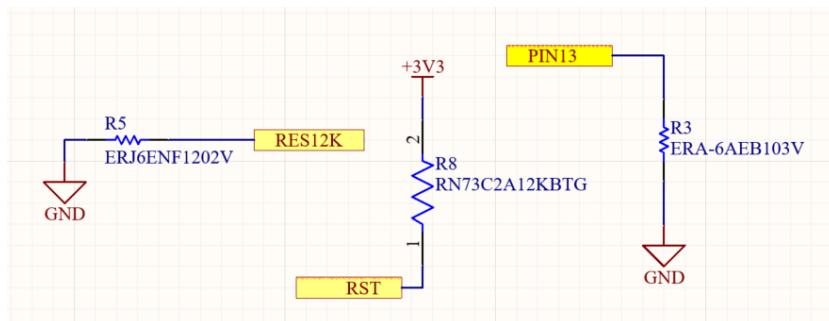


Abbildung 3.15: Pegel wichtiger Pins ESP8266

Der Widerstand R5 wurde zur internen Stromquellenkalibrierung in Verbindung mit dem RES12K-Pin auf Ground geschaltet. Er gehört zum Standarddesign des ESP8266 und ist für einen stabilen Betrieb Pflicht.

Der RST-Pin ist ein Active-Low-Pin und muss für den Betrieb auf High geschaltet werden. Dies ist mit einem 12kΩ Pull-Up-Widerstand realisiert.

Zusätzlich ist ein 10kΩ Pull-Down-Widerstand (R3) an einem Strapping-Pin vorgesehen. Dieser stellt sicher, dass der ESP8266 im Flash-Boot-Modus startet. Das ist für das Programmieren über UART erforderlich. Somit weiß der ESP, was er beim Start machen soll.

3.4.9 Breakout

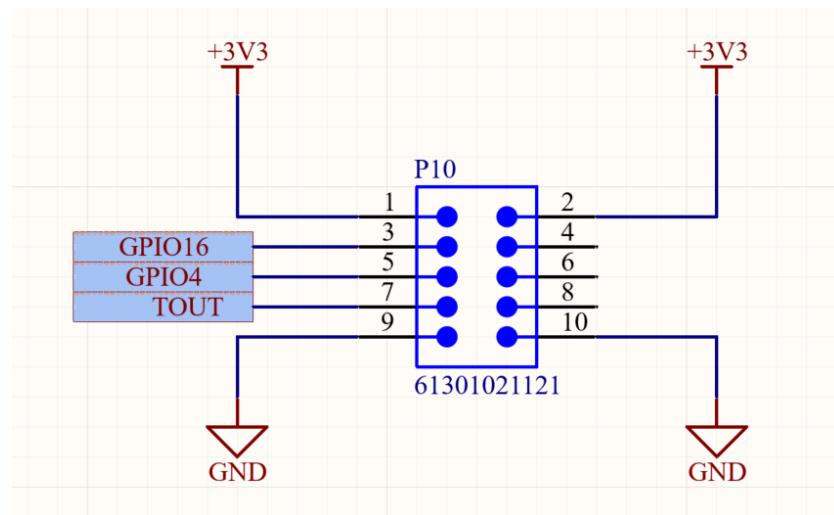


Abbildung 3.16: Breakout Pins ESP8266

Über die herausgeführten TOUT-Pins kann die Spannungsversorgung für die Inbetriebnahme oder für spätere Anwendungen überwacht werden. GPIO 4 und 16 wurden ebenso für die Möglichkeit einer späteren Anbindung über die Pinleiste herausgeführt.

3.5 SPANNUNGSVERSORGUNG

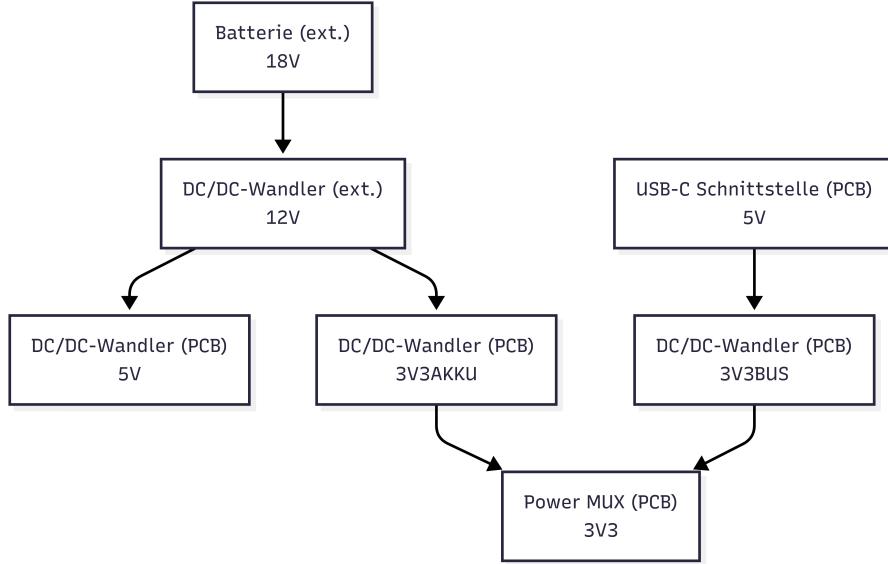


Abbildung 3.17: Energieversorgungsschema

Die Energieversorgung des Systems basiert auf zwei unabhängigen Eingangsquellen: einer externen 18V-Batterie sowie der USB-C-Schnittstelle mit 5V. Die Batteriespannung wird extern auf 12V reduziert und auf der Leiterplatte mittels DC/DC-Wandlern in die benötigten Spannungen 5V und 3.3V (3V3AKKU) gewandelt. Parallel dazu wird die USB-Spannung über einen weiteren Wandler auf 3.3V (3V3BUS) transformiert. Die Stromversorgung ist somit sowohl im mobilen Akkubetrieb als auch im stationären USB-Betrieb möglich.

Der USB-Versorgungszweig dient primär Debugging- und Entwicklungszwecken. Die Platine kann vollständig betrieben und programmiert werden, ohne dass eine Batterie angeschlossen ist. Dies erleichtert Firmware-Updates, Tests und Fehlersuche erheblich, insbesondere im stationären Laborbetrieb.

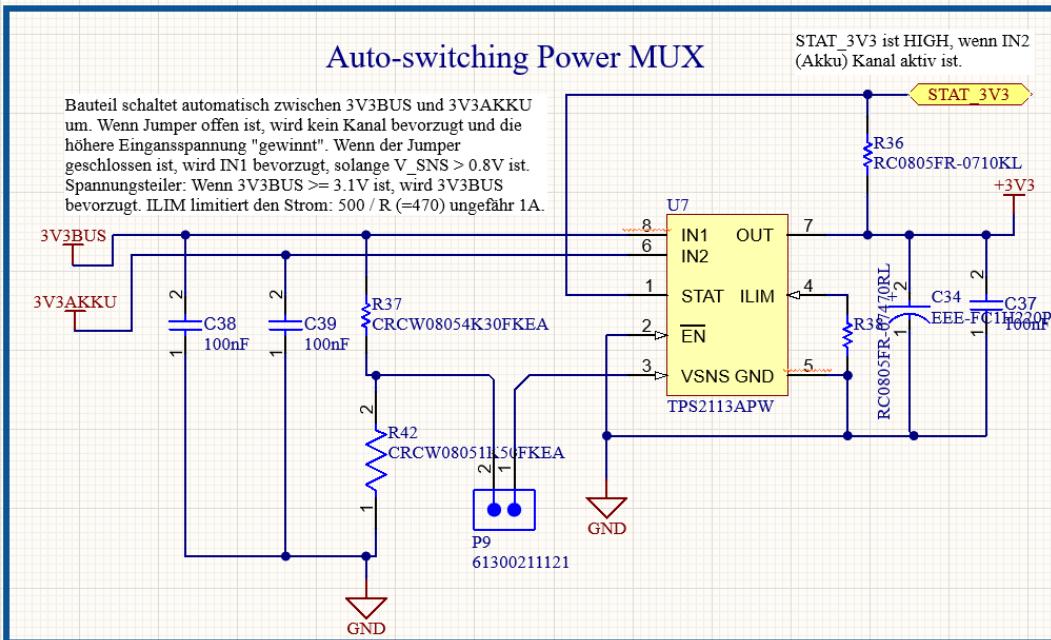


Abbildung 3.18: automatische Spannungsumschaltung

Ein zentraler Bestandteil ist der Power-Multiplexer (TPS2113A), der automatisch zwischen den beiden 3.3V-Zweigen umschaltet. Dabei kann entweder die Quelle mit höherer Spannung priorisiert werden, oder – sofern der Jumper gesetzt ist – der USB-Zweig (IN1) bevorzugt werden. Die Auswahl erfolgt intern über einen Spannungsteiler (VSNS), der bei einer USB-Spannung $\geq 3.1V$ die Umschaltung steuert. Über einen externen Widerstand ($R_{38} = 470\Omega$) lässt sich zudem der maximale Ausgangsstrom auf ca. 1A begrenzen. Am Ausgang sorgen Elkos und Keramik-Kondensatoren für die notwendige Stabilität der Versorgungsspannung.

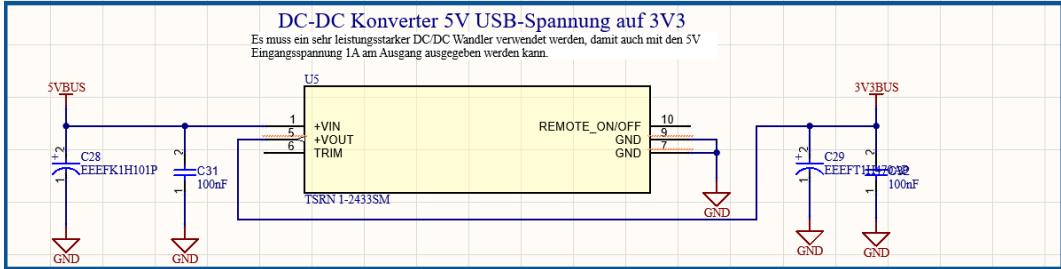


Abbildung 3.19: USB Spannungsversorgung

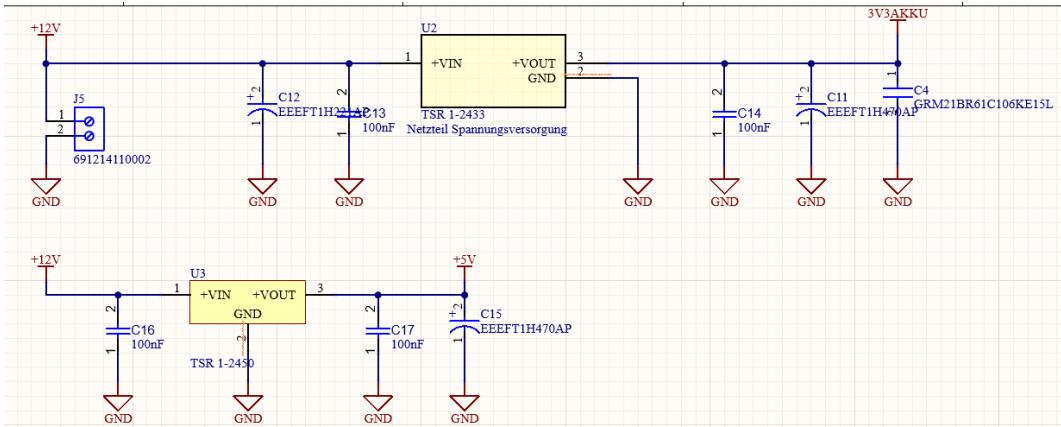


Abbildung 3.20: Spannungsversorgung durch Batterie

Für alle Spannungswandlungen kommen hocheffiziente DC/DC-Wandler der TSR-Serie zum Einsatz. Neben dem Wandler für die 3V3BUS-Spannung wird auch die Akkusspannung (12V) auf 5V und 3.3V gewandelt. Insbesondere für den 3.3V-Kreis ist eine ausreichende Stromreserve erforderlich, da der WLAN-Chip (ESP8266) hohe Spitzenströme zieht. Ein linearer Spannungsregler wäre hier ineffizient gewesen und hätte unnötig Wärme erzeugt. Alle Wandler sind ein- und ausgangsseitig mit Elektrolyt- und Keramikkondensatoren gepuffert – Elkos liefern hohe Kapazität zur Energiespeicherung, KerKos reagieren schnell auf Lastwechsel und filtern hochfrequente Störungen.

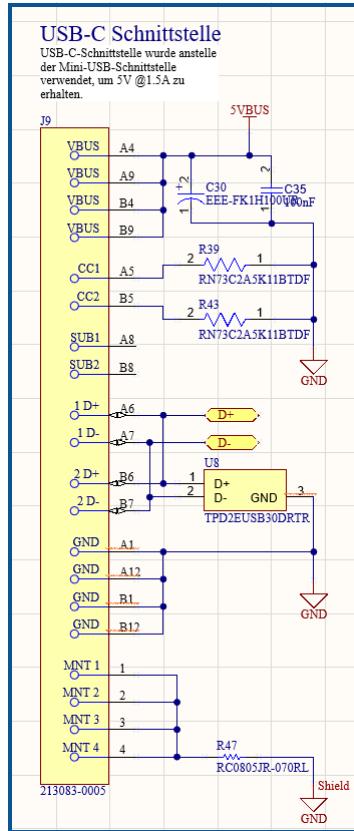


Abbildung 3.21: USB-C Schnittstelle

Die USB-C-Schnittstelle wurde anstelle eines Micro-USB-Anschlusses gewählt, da über sie sicher 5V bei bis zu 1.5A bezogen werden können. Die korrekte Stromstärkevereinbarung erfolgt über passende Pull-Down-Widerstände ($5.11\text{k}\Omega$) an den CC-Pins. Zusätzlich schützt eine ESD-Diode die empfindlichen Datenleitungen ($D+/D-$) vor Überspannung oder elektrostatischer Entladung.

3.6 PIN-ZUORDNUNG UND EXTERNE SCHNITTSTELLEN

Um die Position besser regeln zu können, wird außerdem die Rotation der Motoren über Encoder erfasst. Jeder Motor hat dazu zwei Channels, sodass auch die Drehrichtung erfasst werden kann. Allerdings liefern die Encoder ein Signal mit einer Amplitude von 5V, während die Eingänge am STM32 mit einem Spannungsniveau von 3.3V arbeiten. Um das Spannungsniveau anzupassen, wurde ein Spannungsteiler konzipiert. Die Frequenz des Signals ist gering genug, sodass die parasitären Eigenschaften dieser Schaltung nicht signifikant sind.

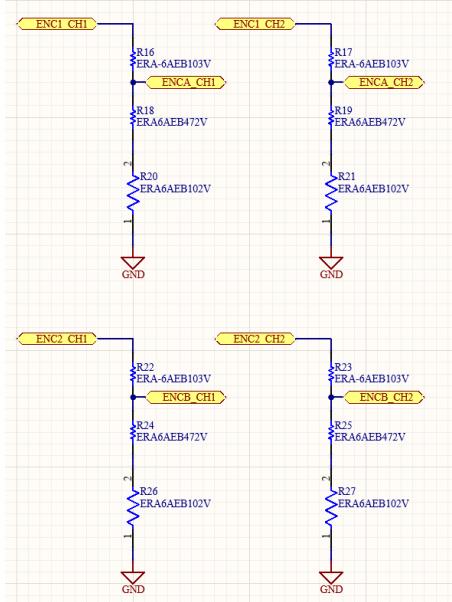


Abbildung 3.22: Encoder Spannungsteiler

Die Spannung am Eingang des STM32 ergibt sich bei dieser Schaltung:

$$U_{stm} = 5V \cdot \frac{R_2+R_3}{R_1+R_2+R_3} = 5V \cdot \frac{4.7k\Omega+1k\Omega}{10k\Omega+4.7k\Omega+1k\Omega} = 3.2V$$

Um in Zukunft eine Erweiterung mit Sensoren oder weiteren Aktoren zu ermöglichen, wurden zahlreiche Schnittstellen über Pins nach außen geführt. Dabei wurde darauf geachtet, dass von jedem relevanten Protokoll entsprechende Pins ausgeführt sind. Falls diese Schnittstellen nicht gebraucht werden, können die Pins auch als GPIOs genutzt werden.

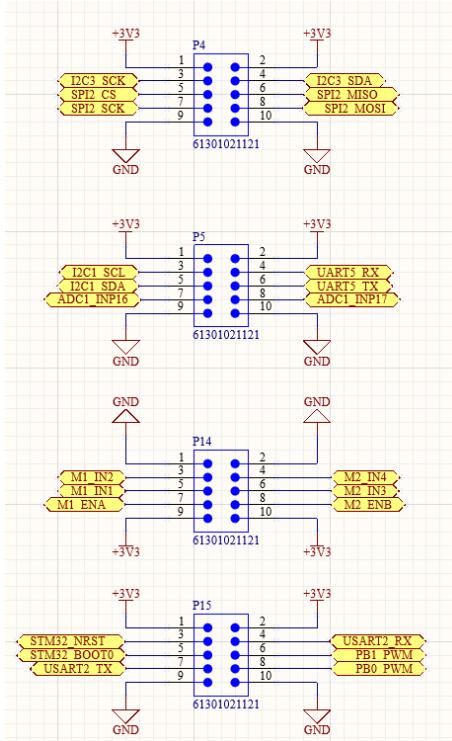


Abbildung 3.23: Pinout für Erweiterungen

4 PCB Design

Zur besseren Übersichtlichkeit und zur Vereinfachung kommender Arbeitsschritte, haben wir uns für eine klare Trennung von verschiedenen Bereichen auf der Platine entschieden. Nach der systematischen Zuordnung aller Komponenten entschieden wir uns für einen Vier-Layer-Aufbau, da bei mehreren Bauteilen eine hohe Anzahl an Signalkontakten genutzt wird.

Die beiden inneren Layer wurden jeweils als durchgehende Polygone für die Betriebsspannung (3.3V) und Masse (GND) vorgesehen. Dadurch kann die Spannungsversorgung über einfache Vias an beliebigen Stellen angebunden werden, was sowohl die Platzierung als auch das Routing deutlich erleichtert.

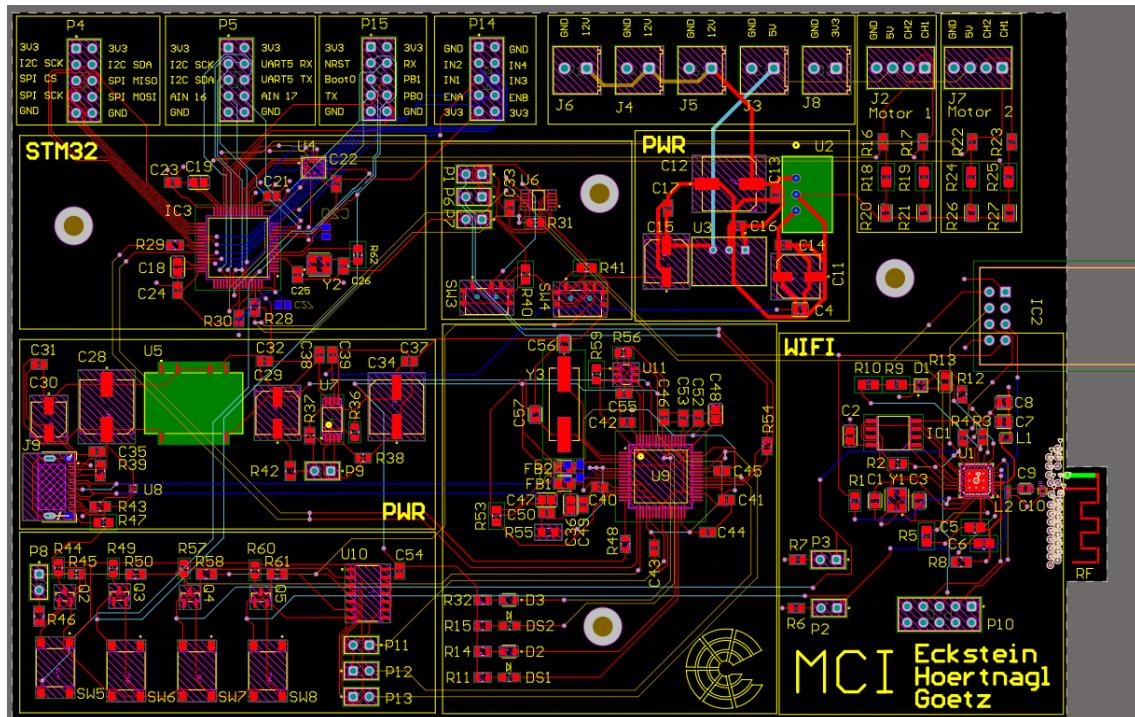


Abbildung 4.1: PCB Design

4.1 STECKER UND SCHNITTSTELLEN

- P4, P5, P15: Reserve für zukünftige Erweiterungen
 - P14: Signale für die H-Brücke, zur Ansteuerung der Motoren
 - J3-6, J8: Schraubklemmen für externe 12V, sowie Anschluss der Peripherie
 - J2, J7: Encoder Signale der Motoren

4.2 STM32

Beim Layout des PCB um den STM32 wurden zunächst alle Komponenten platziert. Dabei wurde insbesondere darauf geachtet, dass die Keramikkondensatoren nah an den Anschlüssen für die Versorgungsspannung liegen. Andernfalls würde der Widerstand und die Induktivität der Leiterbahnen den Effekt der Kondensatoren zu sehr mindern. Dies gilt auch für den externen Oszillator, der ebenfalls nah an den Anschlüssen platziert wurde. Aus Platzgründen wurden einige der Kondensatoren auf der Rückseite der Platine platziert. das ermöglicht ein sauberes Routing und die Kondensatoren sind trotzdem nah genug an den Versorgungspins.

Anschließend konnten die Pins zu den Steckern P4, P5, P14, P15 ausgeführt werden. Dabei wurde je ein Layer pro Stecker verwendet, um das Tracing sauberer zu gestalten.

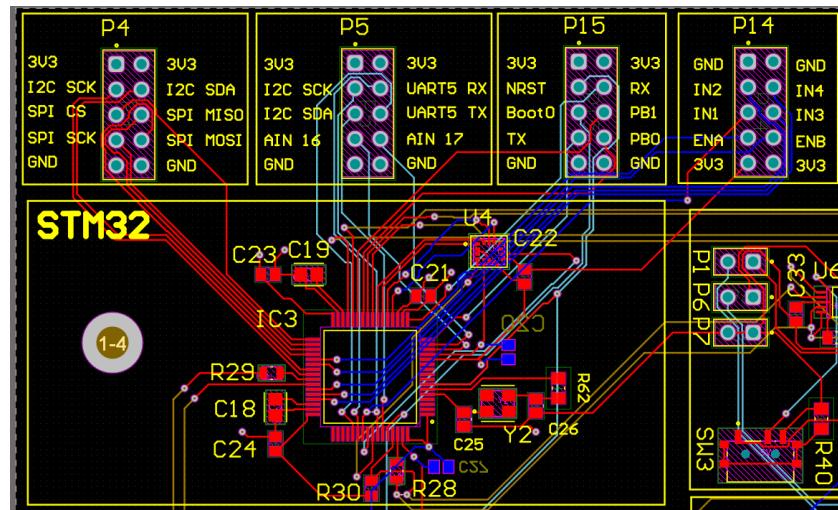


Abbildung 4.2: STM32 PCB Layout

4.3 UART UMSCHALTUNG

Bei der Umsetzung der UART-Umschaltung mittels analogem Multischalter wurde besonders auf eine kompakte und zentralisierte Platzierung geachtet. Zudem war es wichtig, die zugehörigen Schalter gut zugänglich und in direkter Nähe zueinander anzuordnen, um eine intuitive Bedienung zu ermöglichen. Wie bei allen Versorgungspins wurde auch hier darauf geachtet, den Keramikkondensator möglichst nah am Versorgungseingang des MUX zu platzieren, um eine stabile Spannungsversorgung sicherzustellen.

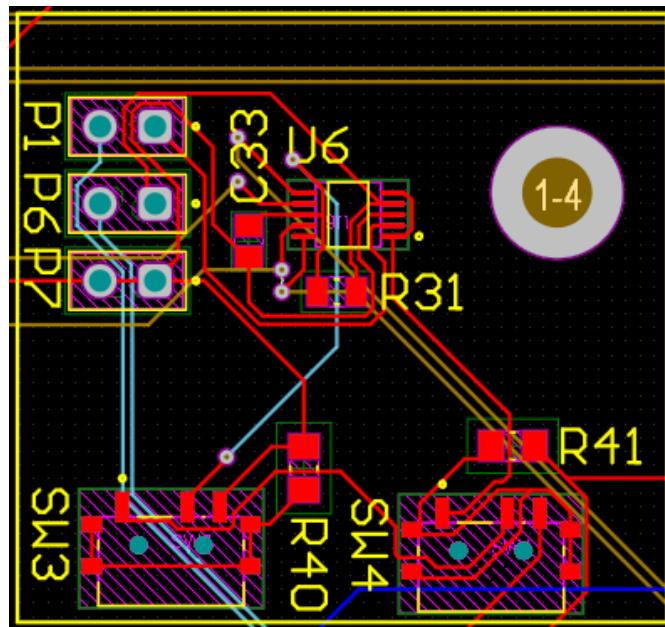


Abbildung 4.3: PCB UART Umschaltung

4.4 PWR BATTERIE

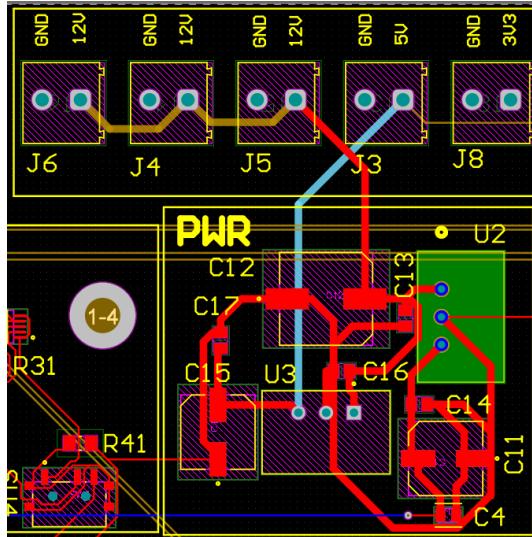


Abbildung 4.4: PCB Batterie PWR

Die Spannungswandler und Kondensatoren wurden nah an den Schraubklemmen platziert, um die Leiterbahnen mit hoher Strombelastung kürzer zu halten. Außerdem wurde die Breite der Leiterbahnen erhöht, um die Stromtragfähigkeit zu gewährleisten.

Die H-Brücke um die Motoren zu treiben, wird über den Anschluss J4 mit der Versorgungsspannung verbunden. Somit ist an dieser Stelle mit höheren Strömen zu rechnen.

4.5 PWR USB UND POWER MUX

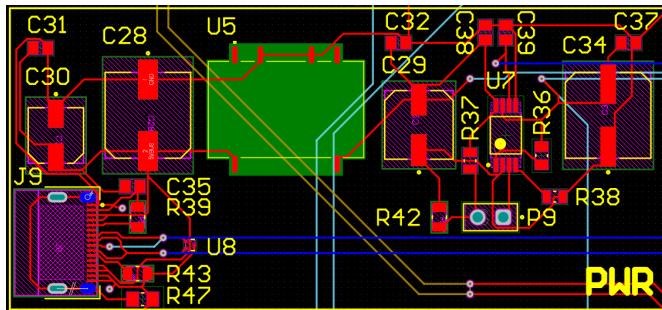


Abbildung 4.5: USB Schnittstelle und Spannungsumschaltung

Das gezeigte PCB-Layout zeigt den Bereich der USB-Versorgung sowie die automatische Umschaltung zwischen den beiden 3.3V-Versorgungszweigen mittels Power-MUX (TPS2113A). Die USB-C-Schnittstelle (links unten) liefert bis zu 1.5A und wurde aufgrund ihrer symmetrischen Bauform gewählt, wodurch das Einsticken unabhängig von der Orientierung möglich ist. Die Spannungsregulation auf 3.3V erfolgt über einen leistungsstarken DC/DC-Wandler (U5). Zur Stabilisierung der Spannungsversorgung wurden jeweils Elektrolyt- und Keramikkondensatoren direkt an den Ein- und Ausgängen platziert. Der MUX (U7) befindet sich zentral, der Konfigurationsjumper (P9) erlaubt die Priorisierung des Versorgungspfads. Die Masseflächen sind großzüig ausgeführt und verbessern die EMV-Eigenschaften sowie die Stromtragfähigkeit.

4.6 USB-UART WANDLER

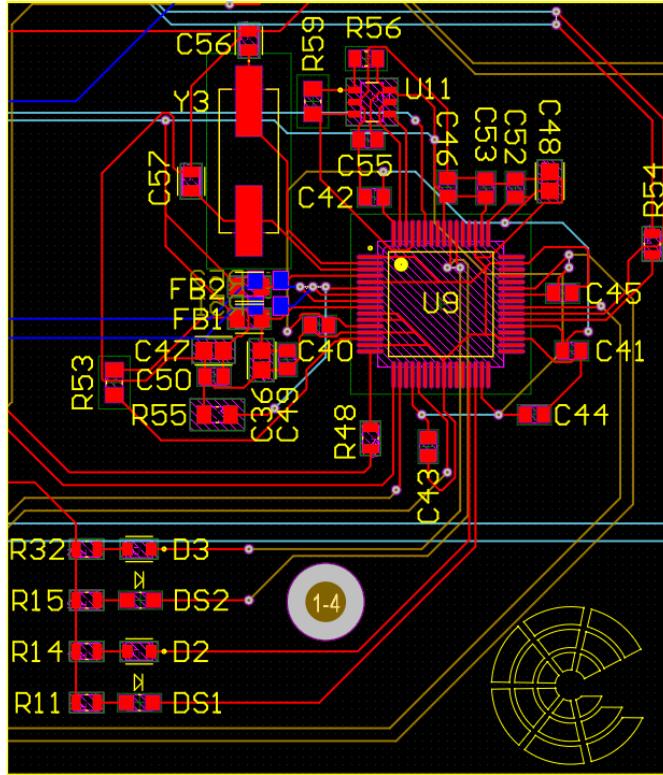


Abbildung 4.6: PCB FTDI Schaltung

Die Abbildung zeigt den zentralen Bereich der Kommunikationsschnittstelle, rund um den FTDI-Chip (U9), der als USB-zu-Dual-UART-Brücke fungiert und somit eine Schlüsselrolle bei der seriellen Kommunikation zwischen PC, STM32 und ESP8266 übernimmt. Der Baustein wurde bewusst möglichst zentral auf der Platine positioniert, um kurze und übersichtliche Leitungswege zu gewährleisten.

Besonderes Augenmerk wurde auf die korrekte Platzierung der Entkopplungskondensatoren gelegt. Zahlreiche Keramikkondensatoren wurden gemäß Designguideline so nahe wie möglich an den jeweiligen Versorgungspins platziert, um eine stabile Spannungsversorgung bei schnellen Lastwechseln sicherzustellen. Aus Platzgründen mussten zwei Widerstände auf die Rückseite der Platine verlagert werden.

Auch der Schwingquarz (Y3) wurde möglichst nahe an den entsprechenden Pins des FTDI-Chips positioniert, um ein sauberes Taktsignal ohne Störungen zu gewährleisten. Die UART-Signalleitungen sind mit Status-LEDs (DS1, DS2, D2, D3) ausgestattet, die die Aktivität der RX- und TX-Leitungen visuell darstellen und somit beim Debugging unterstützen.

4.7 LEITERPLATTENDESIGN FÜR EIN STANDALONE ESP8266EX-MODUL

Für den Entwurf eines eigenständigen ESP8266EX-Moduls gelten spezielle Designvorgaben bezüglich der Stromversorgung, des Quarzoszillators, der HF-Leitungsführung und der Masseanbindung. Die Versorgungsspannung beträgt 3.3V, die Stromversorgungsleitungen besitzen eine Mindestbreite von 15 mil.

Die Eingangs- und Ausgangskondensatoren am Oszillator sind seitlich neben dem Chip positioniert und nicht direkt in der Leitung. Zudem wurden keine hochfrequenten digitalen Signale unter dem Quarz geführt und es wurden keine störenden Bauteile wie Schaltregler oder Spulen in unmittelbarer Nähe verbaut.

Die Antennenanbindung wurde möglichst kurz ausgeführt und mit vielen Masse-Vias abgeschirmt. Es wurden auch keine Biegungen in den Antennenleitungen umgesetzt. Hochfrequente Signale wie SDIO_CLK oder UART wurden nicht in der Nähe der Antenne platziert. Die freistehende PCB-Antenne wurde außerhalb des regulären Boards angebracht, ohne Masse- und Versorgungsebene. Zudem ist die Antenne möglichst weit von anderen Störquellen entfernt.

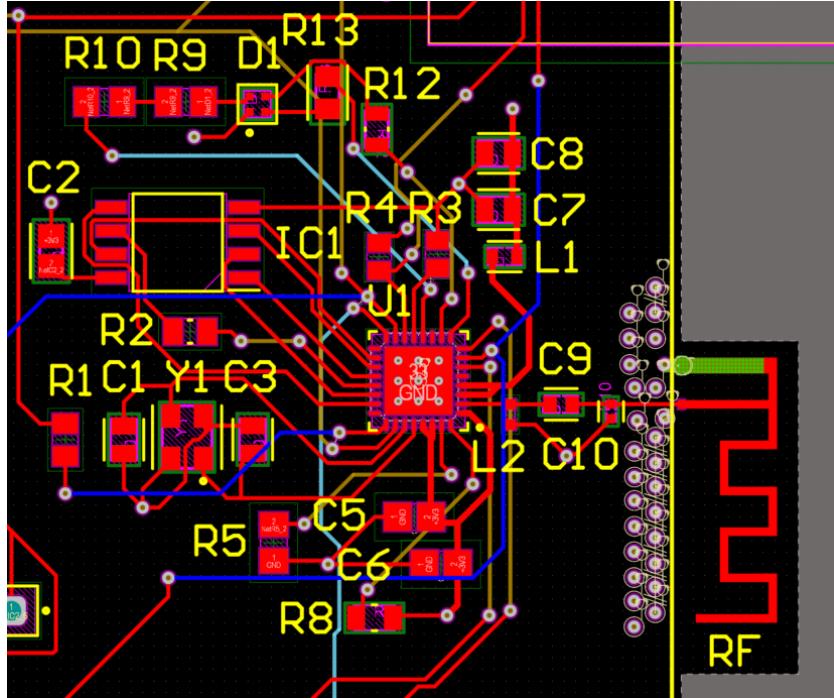


Abbildung 4.7: PCB-Layout WiFi

4.8 BACKUP LÖSUNG

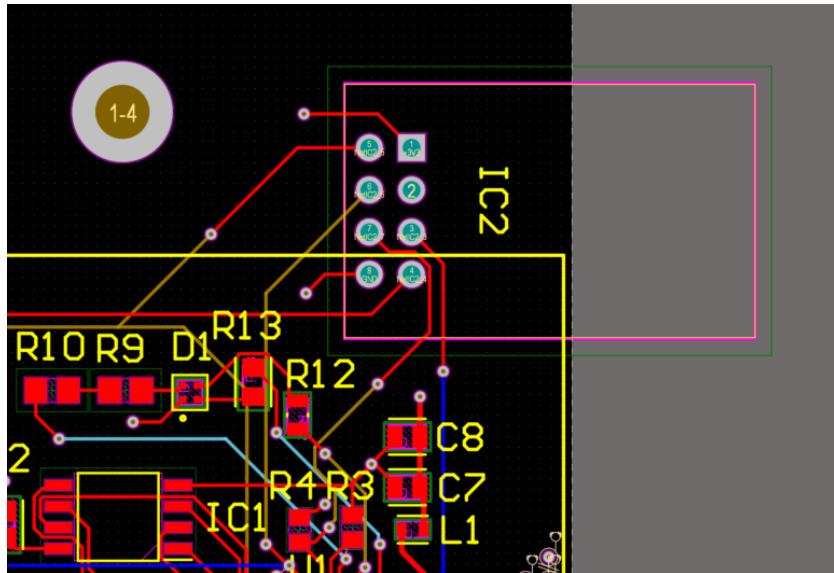


Abbildung 4.8: Backup WiFi-Kommunikation PCB

Als vollständige Backup-Lösung wurde, wie beschrieben, der ESP01 verwendet. Dieser kann mit den bestehenden Pins in den dafür vorgesehenen Footprint eingelötet werden. Dabei wurde die gleiche Anbindung wie bei der regulären Variante an die UART-, Reset- und Enable-Pins vorgesehen.

4.9 BOOTLOADER AKTIVIERUNG

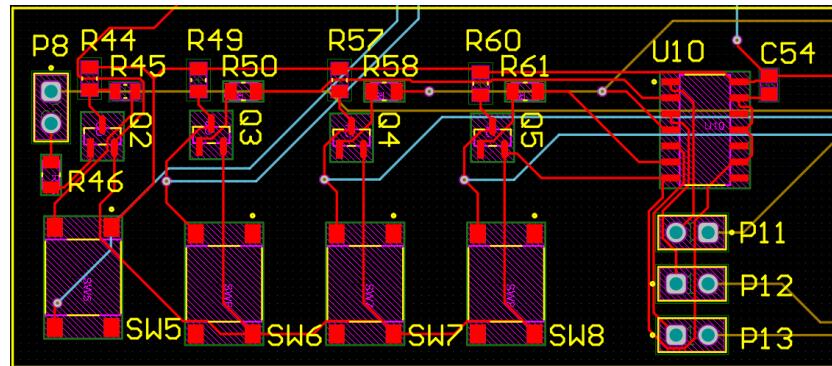


Abbildung 4.9: Bootloaderschaltung PCB

Die Bootloader-Aktivierungsschaltung wurde aus Gründen der Übersichtlichkeit möglichst symmetrisch und kompakt aufgebaut. Alle zugehörigen Bedienelemente – jeweils ein Reset- und Boot-Taster für STM32 und ESP8266 – wurden bewusst nebeneinander an einem Ort platziert, um eine intuitive Bedienung zu ermöglichen. Aufgrund von Platzmangel wurde im finalen Layout von der ursprünglich vertikalen auf eine horizontale Anordnung gewechselt.

Die Transistorschaltungen (Q2–Q5) dienen der entkoppelten Signalweiterleitung zwischen FTDI-Chip und Ziel-MCUs, während der Inverter (U10) für die korrekte Logiksteuerung benötigt wird. Die Jumper erlauben bei Bedarf die Deaktivierung der automatischen Bootlogik oder den Anschluss eines externen Programmieradapters. Eine nachträgliche manuelle Beschriftung der Taster ist vorgesehen, da dies im Layoutprozess versäumt wurde.

5 Mechanischer Aufbau

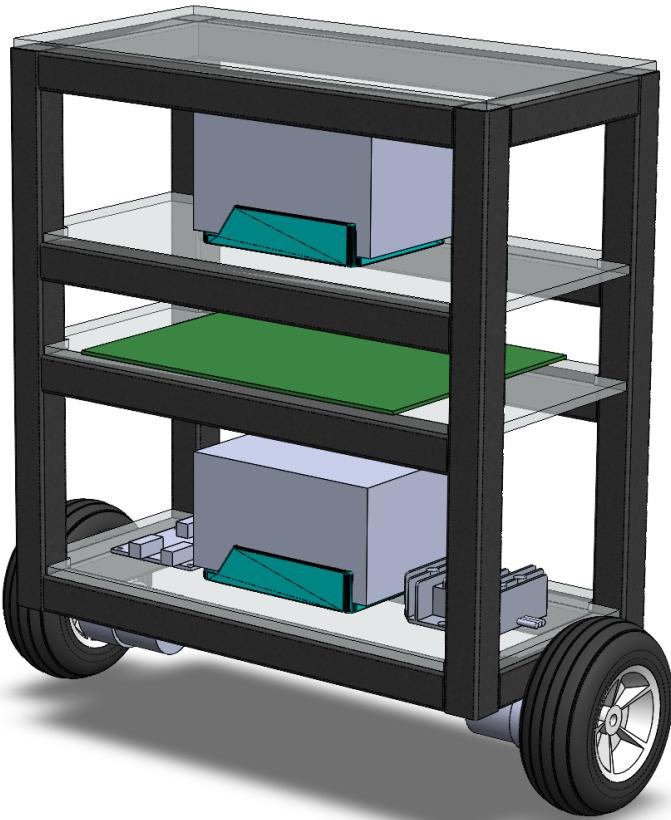


Abbildung 5.1: Konstruktion

Die Konstruktion besteht aus einem mehrstufigen Rahmengehäuse, in dem die elektronischen Komponenten und der mechanische Antrieb verbaut sind. Die Böden bilden hierbei 5 mm dicke Plexiglasplatten.

5.1 RAHMENSTRUKTUR

Die Rahmenstruktur besteht aus ITEM-Profilen mit den Abmessungen 20 x 20 mm. Diese werden mit systemspezifischen Verbindungsschrauben und Nutensteinen miteinander verbunden. Die Struktur bietet die Möglichkeit, offen auf das System zuzugreifen und die Position der Akkus zu wechseln, um den Schwerpunkt der Anwendung anzupassen.

5.2 EBENE 0

Die grundlegende Mechanik, bestehend aus Motoren und Rädern, wird auf der Unterseite der Konstruktion mithilfe von 3D-gedruckten Halterungen umgesetzt. Die Reifen haben einen Durchmesser von 10 cm und werden an den Motorenachsen befestigt.

5.3 EBENE 1

Zur Ansteuerung der Gleichstrommotoren kommt ein kompakter Motortreiber auf einer kleinen, separaten Platine zum Einsatz. Dieser ist auf der untersten Ebene des Aufbaus untergebracht und übernimmt die Leistungsansteuerung der Antriebsmotoren. Ebenso wie eine der beiden möglichen Positionen, an denen der Akku der Anwendung platziert werden kann. Der DC/DC-Wandler vervollständigt die Ebene.

5.4 EBENE 2

Diese Ebene ist vollständig für das PCB vorgesehen. Hierbei werden Kabel durch die Plexiglasplatten geführt und nach der Vormontage mithilfe von Kabeldurchführungen verbaut.

5.5 EBENE 3

Auf dieser Ebene befindet sich die zweite, erhöhte Position des Akkus der Anwendung. Ein erhöhter Schwerpunkt erleichtert die Regelung und sorgt somit für ein gutes Ergebnis.

5.6 EBENE 4

Diese Ebene ist vollständig für spätere Erweiterungen vorgesehen, zu denen unter anderem Kameras und weitere Sensoren gehören.

5.7 ABMESSUNGEN

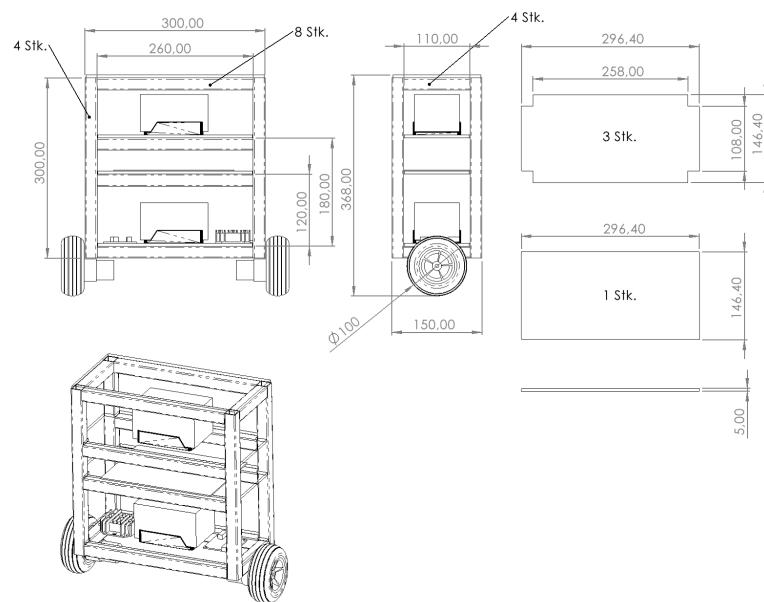


Abbildung 5.2: Abmessungen Gehäuse

Die Außenmaße des Rahmens betragen dabei 30 cm in der Länge, 150 cm in der Tiefe und 30 cm in der Höhe. Damit ist er kompakt und leicht zu transportieren.

6 Kostenkalkulation

Im Rahmen der Projektzwischenphase wurde eine detaillierte Kostenabschätzung der aktuell geplanten Bauteile und Komponenten durchgeführt. Die in Tabelle 6.1 aufgeführten Positionen basieren auf einer finalen Auswahl aller relevanten elektrischen und mechanischen Komponenten, wie sie in der endgültigen Umsetzung vorgesehen sind. Obwohl die Bestellung zum aktuellen Zeitpunkt noch nicht erfolgt ist, wird mit keiner wesentlichen Änderung der Bauteilliste mehr gerechnet. Kosten für mechanische Rahmenprofile und Plexiglasscheiben wurden nicht berücksichtigt, da diese über hausinterne Ressourcen zur Verfügung gestellt werden.

Pos.	Bauteil	Beschreibung	ungefähre Kosten (€)
01	PCB	4-Layer PCB	85,00
02	PCB-Komponenten	alle Chips und Bauteile, die auf die Platine gelötet werden	100,00
03	Räder	Rad 100mm mit Kugellager (2x)	30,00
04	Motor	12V DC-Getriebemotor mit Encoder 90:1, 38kgcm (2x)	65,00
05	Motordriver	Dual-Channel DC Motor Driver 12A PWM	35,00
06	Akku	Makita Akku BL1840B 18V/4Ah Li-Ion	80,00
07	DC/DC-Wandler	Bauer Electronics DC-DC 18V–36V zu 12V, 10A, 120W	20,00
08	ESP01	ESP8266 ESP-01S WiFi Modul	5,00
09	Akkuadapter	QUPERR Adapter mit Sicherung und Schalter (2x)	20,00
Gesamtkosten			440,00

Tabelle 6.1: Stückliste mit Komponenten und ungefähren Einzelkosten

Das ursprünglich vorgesehene Budget von 400€ wird nach aktuellem Stand mit den geplanten Gesamtkosten von 440€ geringfügig überschritten. Diese Überschreitung ist vor allem auf folgende Punkte zurückzuführen:

- Der Einsatz einer 4-Layer-Leiterplatte war aufgrund der hohen Pinanzahl und des komplexen Routings notwendig und verursacht höhere Fertigungskosten als eine herkömmliche 2-Layer-Platine.
- Die gewählten DC-Getriebemotoren mit hohem Drehmoment sind kostenintensiver, erfüllen jedoch die mechanischen Anforderungen des Systems zuverlässig.
- Der vorgesehene Makita-Akku mit integriertem Batteriemanagementsystem bietet hohe Sicherheit und Komfort, bringt jedoch auch erhöhte Anschaffungskosten mit sich.
- Einige spezifische Leiterplattenkomponenten wie der STM32-Mikrocontroller sind aufgrund ihrer Leistungsfähigkeit vergleichsweise teuer. Gleiches gilt für den DC/DC-Wandler TSRN1-2433SM, der aufgrund seiner Fähigkeit, auch bei nur 5V Eingangsspannung ausreichend Strom auf 3.3V bereitzustellen, für die Spannungsversorgung des Systems essenziell ist.

In Summe erscheinen die prognostizierten Ausgaben angesichts der gewählten Architektur und Zielanforderungen nachvollziehbar und sinnvoll.

7 Pinbelegung und Funktionszuordnung

Pin	Funktion	Beschreibung	Bemerkung
PA0	ADC1_INP16	reserve pin	
PA1	ADC1_INP16	reserve pin	
PA2	USART2_TX	reserve pin	
PA3	USART2_RX	reserve pin	
PA4	SPI1_CS	lagesensor	
PA5	SPI1_SCK	lagesensor	
PA6	SPI1_MISO	lagesensor	
PA7	SPI1_MOSI	lagesensor	
PA8	I2C3_SCK	Reserve Pin	
PA9	USART1_RX	for flashing	
PA10	USART1_TX	for flashing	
PA11	M1_IN2	in Motor 1	
PA12	M2_IN3	in Motor 2	
PA13	M2_IN4	in Motor 2	
PA14			
PA15			
PB0	PWM	Reserve pin	tim3 ch3
PB1	PWM	Reserve pin	tim3 ch4
PB2	ENC2_CH1	Encoder 2	
PB3	M1_IN1	in Motor 1	
PB4	lsm_int2	Interrupt 2 sensor	
PB5	lsm_int1	Interrupt 1 sensor	
PB6	I2C1_SCL	reserve pin	
PB7	I2C1_SDA	reserve pin	
PB8	UART4_RX	for Esp	
PB9	UART4_TX	for Esp	
PB10	ENC2_CH2	Encoder 2	
PB11			
PB12	SPI2_CS	reserve pin	
PB13	SPI2_SCK	reserve pin	
PB14	SPI2_MISO	reserve pin	
PB15	SPI2_MOSI	reserve pin	
PC0			
PC1			
PC2			
PC3			
PC4	ENC1_CH1	Encoder 1	
PC5	ENC1_CH2	Encoder 1	
PC6	M1_ENA	PWM Motor 1	tim3 ch1
PC7	M2_ENB	PWM Motor 2	tim3 ch2
PC8			
PC9	I2C3_SDA	reserve pin	
PC10			
PC11			
PC12	UART5_TX	reserve pin	
PC13			
PC14			
PC15			
PD2	UART5_RX	reserve pin	

Tabelle 7.1: Pinbelegung und Funktionen

Abbildungsverzeichnis

0.1	Entwurf inverses Pendel	II
2.1	UART Topologie	2
3.1	STM32 mit Quarz und Kondensatoren	3
3.2	LSM6DSOX	4
3.3	FTDI-Chip mit Peripherie	5
3.4	UART-Signalumschaltung	6
3.5	Bootloader-Schaltung	7
3.6	PCB Abschnitt WiFi	9
3.7	Stromversorgung ESP8266	10
3.8	Stromversorgung Antenne	10
3.9	Antenne	11
3.10	Flash Beschaltung ESP8266	12
3.11	ESP Oszillator	12
3.12	WiFi Backup	13
3.13	Backup - ESP01[6]	13
3.14	LED Schaltplan	13
3.15	Pegel wichtiger Pins ESP8266	14
3.16	Breakout Pins ESP8266	15
3.17	Energieversorgungsschema	16
3.18	automatische Spannungsumschaltung	16
3.19	USB Spannungsversorgung	17
3.20	Spannungsversorgung durch Batterie	17
3.21	USB-C Schnittstelle	18
3.22	Encoder Spannungsteiler	19
3.23	Pinout für Erweiterungen	19
4.1	PCB Design	20
4.2	STM32 PCB Layout	21
4.3	PCB UART Umschaltung	21
4.4	PCB Batterie PWR	22
4.5	USB Schnittstelle und Spannungsumschaltung	22
4.6	PCB FTDI Schaltung	23
4.7	PCB-Layout WiFi	24
4.8	Backup WiFi-Kommunikation PCB	24
4.9	Bootloaderschaltung PCB	25
5.1	Konstruktion	26
5.2	Abmessungen Gehäuse	27

Tabellenverzeichnis

2.1	Technische Daten	2
6.1	Stückliste mit Komponenten und ungefähren Einzelkosten	28
7.1	Pinbelegung und Funktionen	29

Literaturverzeichnis

- [1] Matthias Gfall, "Hardwareseitige entwicklung eines mobilen inversen pendels. produktentwicklung 1," Aufgabenstellung, MCI, Innsbruck, 14.02.2025.
- [2] Espressif Systems, "Esp8266 datasheet," Online, 2025. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- [3] ——, "Esp8266 hardware-deisgn-guide: Hardware design guidelines version 2.8," Online, 2025. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp8266_hardware_design_guidelines_en.pdf
- [4] Mouser Electronics, Inc., "W25q128jveiq tr: Nor flash spiflash, 3v, 128m-bit, 4kb uniform sector," Online, 2025. [Online]. Available: https://eu.mouser.com/ProductDetail/Winbond/W25Q128JVEIQ-TR?qs=qSfuJ%252Bfl%2Fd7SkvmcTerX5A%3D%3D&msclkid=d5edd80ef1201ff1669fb06f63e1a632&utm_source=bing&utm_medium=cpc&utm_campaign=DSAs+Other+Suppliers+Austria+English&utm_term=%2FProductDetail%2F&utm_content=Other+Suppliers+DSAs%3A+PDP
- [5] ——, "Abm8-26.000mhz-10-1-u-t: Oszillator," Online, 2025. [Online]. Available: https://eu.mouser.com/ProductDetail/ABRACON/ABM8-26.000MHZ-10-1-U-T?qs=76dmnhCH%2FMP1JRP6EvWbSw%3D%3D&msclkid=535646f8f811166bd1e32e996b76dd5d&utm_source=bing&utm_medium=cpc&utm_campaign=DSAs%20Other%20Suppliers%20Austria%20English&utm_term=%2FProductDetail%2F&utm_content=Other%20Suppliers%20DSAs%3A%20PDP
- [6] reichelt elektronik GmbH, "Entwicklerboards - esp8266 wifi-modul: Debo esp8266," Online, 2025. [Online]. Available: https://www.reichelt.de/de/de/shop/produkt/entwicklerboards_-_esp8266-wifi-modul-192142