

# OpenMoverPlatform

## Projektstatus-Dokumentation

Marcel Kitzbichler

29. Januar 2026

## 1 Einleitung

Die OpenMoverPlatform ist ein mobiler Roboter, der selbstständig fahren oder ferngesteuert werden kann. Dieses Dokument beschreibt den aktuellen Stand der Mechanik, der Elektronik und der Software. Alle Designs und der Quellcode können im Projekt-Repository gefunden werden: <https://github.com/Marcel-Kitzbichler/OpenMoverPlatform>

## 2 Aufbau des Systems

### 2.1 Mechanik

Das Gehäuse und die mechanischen Teile wurden mit SolidWorks konstruiert. Das Design ist modular aufgebaut. Der Modulanschluss befindet sich oben auf der Plattform, dieser Anschluss ist sowohl für die Mechanische Befestigung als auch für die elektrische Verbindung des Moduls mit der Plattform zuständig.

#### 2.1.1 Der Antrieb

Der Antrieb wurde bisher in zwei Versionen entwickelt:

- **Version 1:** Ein erster Prototyp, um testen zu können ob ein komplett auf dem 3D-Drucker gefertigtes Getriebe überhaupt machbar ist.
- **Version 2 (Aktuell):** Diese Version ist fortgeschrittener. Sie besitzt ein Getriebe bestehend aus mehreren 3D-gedruckten Zahnrädern und einen Kettenantrieb. Durch die hohe Übersetzung kann sich der Roboter auch in schwierigen Gelände bewegen.

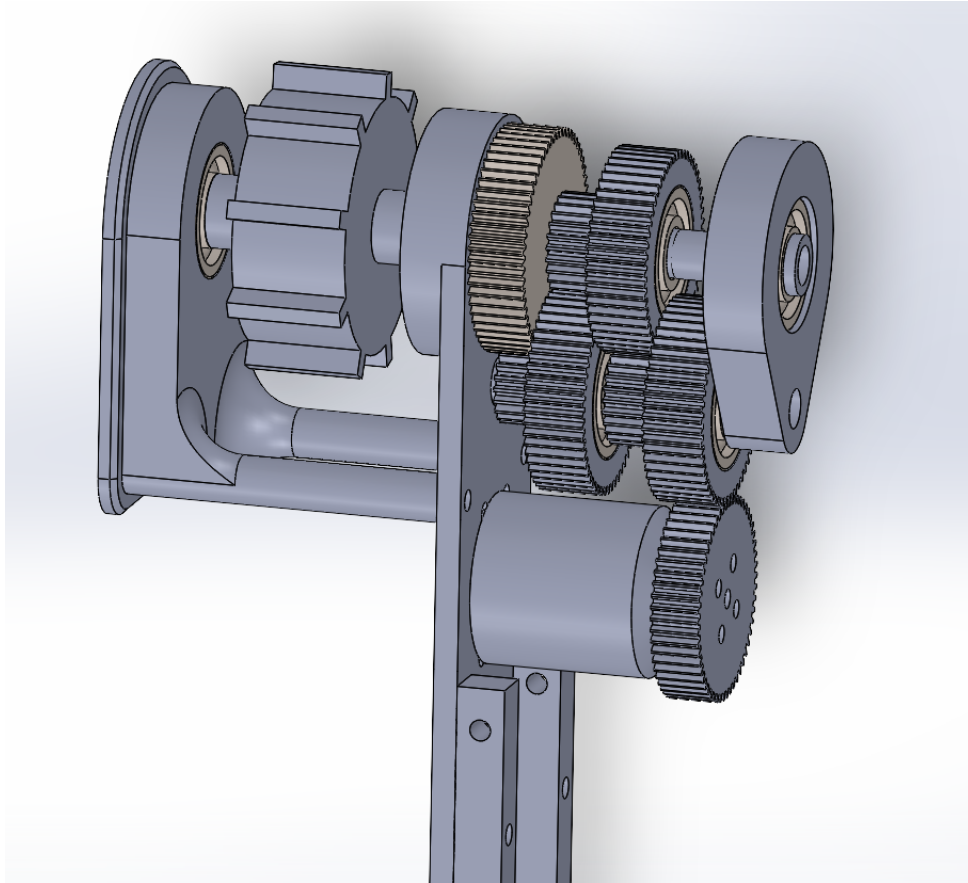


Abbildung 1: CAD-Modell des Antriebs V2

## 2.2 Elektronik

Das Herzstück der Elektronik ist eine eigene Platine (PCB) die in der Schule gefertigt wurde.

- **Steuerung:** Verwendet wird ein ESP32-S3 Mikrocontroller. Er ist leistungsstark und hat bereits ein USB und ein Bluetooth Interface zur Kommunikation mit jeglicher Peripherie integriert.
- **Sensoren:**
  - **Kompass:** Ein HMC5883L Sensor ist per I2C mit der Steuerung verbunden um beim Navigieren die Ausrichtung des Roboters feststellen zu können.
  - **GPS:** Damit der Roboter seine Position bestimmen kann ist ein BN-880 GPS Modul in die Obere Abdeckplatte integriert welche über eine Serielle Schnittstelle mit der Steuerung kommuniziert.
  - **Stromversorgung:** Die Batteriespannung wird durch einen Spannungsteiler dessen Ausgang direkt mit einem ADC Eingang des ESP32 verbunden ist überwacht.
  - **Modulanschluss:** Der kommende Modulanschluss kann auch im weitesten Sinne als Sensor gesehen werden, über diesen Anschluss registrieren sich in Zukunft Module bei der Steuerung und haben durch diesen Anschluss Zugriff

zur gleichen JSON API wie auch über die anderen Schnittstellen bereitgestellt wird.

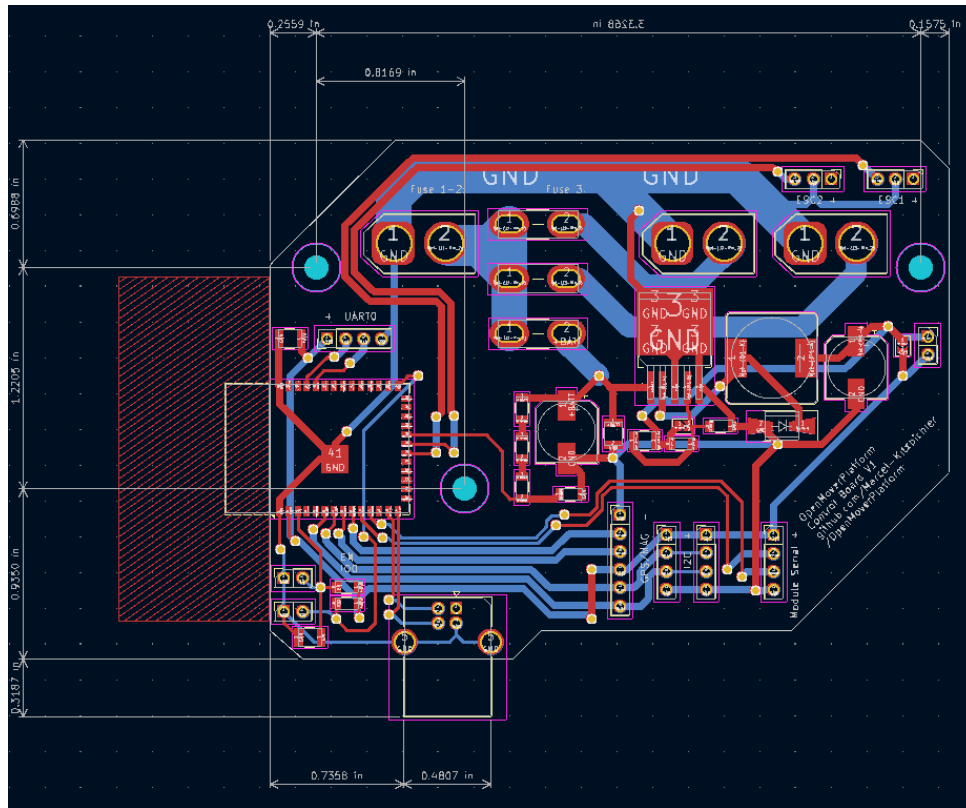


Abbildung 2: Design der Roboter-Platine

## 3 Software

Die Software besteht aus zwei Teilen: Dem Programm auf dem Roboter (Firmware) und dem Programm auf dem Computer zur Steuerung.

### 3.1 Firmware (Auf dem Roboter)

FreeRTOS bildet die Grundlage für die Firmware. Es gibt dauerhaft einen Task, der die Kommunikation mit externen Geräten bewerkstelligt. Wird ein Task gestartet, der Zugriff auf den Motor braucht, wird die PWM Schnittstelle für diesen Task reserviert und der Task handle für diesen Task Global gespeichert, um ihn im Notfall immer aus jeden anderen Task stoppen zu können. Ein Motor Task kann nur starten, wenn gerade kein anderer das Interface reserviert hat. Der Serial Manager Task ist dafür zuständig die JSON Apis bereitzustellen und bei Anfragen die benötigten Daten zu sammeln bzw. die angefragten Prozesse zu starten.

### 3.2 PlatformHelper

Um die Platform zu steuern und Daten auszutauschen gibt es sowohl eine Handy App als auch ein PC Programm, welches im Projekt Repo gefunden werden kann. Diese Anwendungen verbinden sich über die unten beschriebenen Schnittstellen mit der Steuerung.

## 4 Kommunikation

Die Plattform und jegliche Peripherie unterhalten sich über ein einfaches Textprotokoll (JSON). Es gibt verschiedene Befehle, die in der Zukunft noch erweitert und dann vollständig dokumentiert werden wird. Dieses JSON Interface kann auf gleiche Art und Weise über USB , eine Serielle Schnittstelle und über Bluetooth angesprochen werden. Aktuell sind diese Befehle implementiert:

- **Intent 1:** Gespeicherte Wegpunkte abrufen.
- **Intent 2:** Automatische Fahrt starten.
- **Intent 4:** Motoren direkt steuern.
- **Intent 5:** Neue Wegpunkte hochladen.
- **Intent 6:** Status-Informationen abfragen.

## 5 Aktueller Stand und Zukunft

**Was schon fertig ist:**

- Das Design für den neuen Kettenantrieb (V2).
- Der Entwurf der Platine.
- Die grundlegende Software zum Fahren und Steuern.
- Das PC-Programm zur Bedienung.

**Was noch fehlt:**

- Der finale Zusammenbau des neuen Antriebs.
- Neue Tests im Freien, um zu sehen, wie gut der Roboter im aktuellen Zustand alleine fährt.
- Erweiterung der Software und verbesserung der Stabilität