

Modulare Roboter mit RODOS – Beteigeuze NOVA

Nicolas Kessler

August 2020 Artikel

Abstract

Kamaro Engineering e. V. forscht seit mehr als zehn Jahren an Feldrobotern und nimmt mit den entwickelten Prototypen an verschiedenen Wettbewerben teil. Beispielsweise meistern die Roboter jährlich auf dem Field-Robot-Event neue Herausforderungen, die unterschiedlichste mechatronische Lösungen erfordern. Durch die von Event zu Event variierende Aufgabenstellung gewinnt eine modulare Bauweise mehr und mehr an Bedeutung, weil erprobte Komponenten wiederverwendet und aufgabenspezifisch mit neuen Komponenten kombiniert werden können. Dieser Artikel stellt das Projekt NOVA des Teams Elektrotechnik vor, mit dem angestrebt wird, modulare Roboter mit RODOS als Middleware zu konstruieren, deren Komponenten einfach in ROS integrierbar sind.

Modulare Roboter mit RODOS

Beteigeuze NOVA

Motivation und Hintergrund

Bisher setzte sich unser Gesamtsystem aus zwei Modulen zusammen: Einer mobilen Roboterplattform und einem aufgabenspezifischen Anbaugerät. Jedoch modularisiert NOVA einen Schritt weiter; die Roboterplattform selbst wird aus Modulen wie Batteriemanagementsystem, Inertialnavigationssystem, Kameras, Diagnose-Displays und Chassis aufgebaut werden. Das ist mit vielen Herausforderungen verbunden. Neben einer sinnvollen Aufteilung in Komponenten und ausführlicher Dokumentation dieser gehen mit der Modularisierung schwierige technische Herausforderungen einher: Die Kommunikation zwischen den Komponenten muss vollständig standardisiert sein — vom physikalischen Aufbau der Steckverbindungen bis zu der Repräsentation der Daten in der Software.

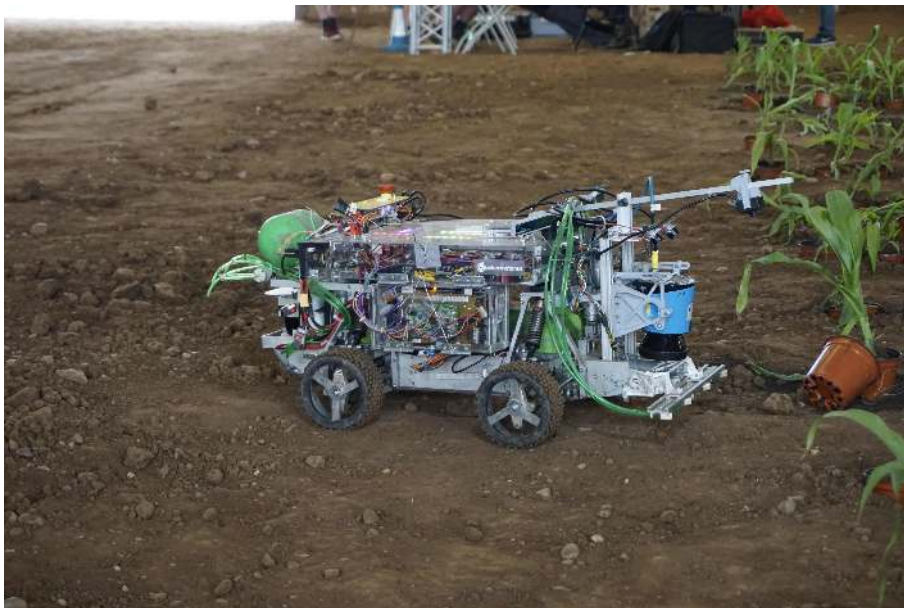


Abbildung 2.1: Beteigeuze [KE20a] mit Anbauten: Spritzeinrichtung vorne und Wassertank im Heck

Bisher setzten wir auf eine sogenannte „Masterplatine“, die verschiedenste Protokolle zur Sensorenanbindung implementiert. Über einem komplexen Kabelbaum wurden den Sensoren Schnittstellen über I2C und CAN bereitgestellt. Über UART und ein selbstentwickeltes Protokoll hatte die Masterplatine mit einem PC kommuniziert, auf dem eine ROS-Node [QCG⁺09] die Funktionalität der Hardware anderen ROS-Nodes zur Verfügung stellte. Während die Verka-

belung von Sensoren unaufwändig war, hat sich ihre Anbindung in die Software als diffizil herausgestellt. Zumal das Programm für die Masterplatine zuständig für die Anbindung aller Sensoren und Aktoren war und dabei verschiedene Randbedingungen an die Synchronisation einhalten musste. Bei der Entwicklung unseres vierten Roboters, Dschubba [KE20b], mussten wir einsehen, dass weder die Masterplatine noch der Kabelbaum wiederverwendbar sind. Die ROS-Nodes dagegen konnten mit nur wenigen Änderungen auf Dschubba portiert werden. Besonders die Komponenten für das autonome Fahren. Somit begannen viele Diskussionen und Experimente zu einem neuen Verkabelungskonzept.

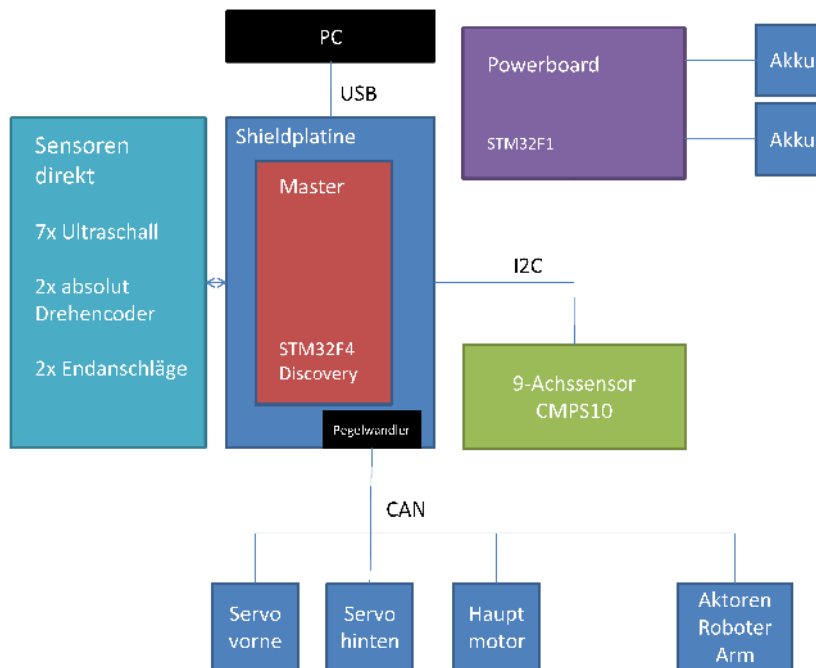


Abbildung 2.2: Übersicht des alten Kabelbaums

RODOS

NOVA sieht vor, modulare Roboter mit RODOS [MD09] [MR09] [Wik20] [RD19] zu entwickeln. RODOS ist eine echtzeitfähige, freie Middleware, die von der Universität Würzburg entwickelt wird. Die Software ist speziell für autonome Systeme gedacht, die hohen Anforderungen an die Zuverlässigkeit unterliegen. Darunter fallen neben Feldroboter auch Drohnen, Satelliten und U-Boote. RODOS ermöglicht die Entwicklung von Software für verteilte Systeme auf unkomplizierte Art. Dazu stellt es einen echtzeitfähigen Scheduler bereit, mit dem zuverlässig Taktraten eingehalten werden können. Ein Hardware Abstraction Layer (HAL) bietet einfache Möglichkeiten, Hardware, wie PWM, I2C, SPI, CAN und ADC anzusteuern. Die mitgelieferte Middleware ermöglicht eine publish-subscribe-basierte Kommunikation zwischen verschiedenen parallel laufenden Softwarekomponenten, in dem Falle auch Nodes genannt. Bei Publish-Subscribe horchen Nodes auf sogenannten Topics nach neuen Daten. Eine Node, die auf Daten wartet, ist ein Subscriber. Auf den Topics können andere Nodes jederzeit Daten senden, woraufhin Subscriber der Topic diese Daten empfangen.

Projekt NOVA im Überblick

Das neue Konzept sieht vor, dass jedes Modul RODOS für die Kommunikation benutzt und an einer Busplatine angeschlossen ist, die alle Module verbindet. Publish-subscribe-basierte Middlewares ermöglichen ein redundantes und robustes Design. Weiterhin bietet eine dezentrale Systemarchitektur für modulare Roboter mit RODOS die Möglichkeit, jederzeit auf Fehler im System reagieren zu können. Wie im Bild unten dargestellt kann das Team die Module für neue Roboter wiederverwenden, die Komponenten selber hängen nicht mehr von einer Masterplatine ab. Statt dessen sind sie selbstständige Module.

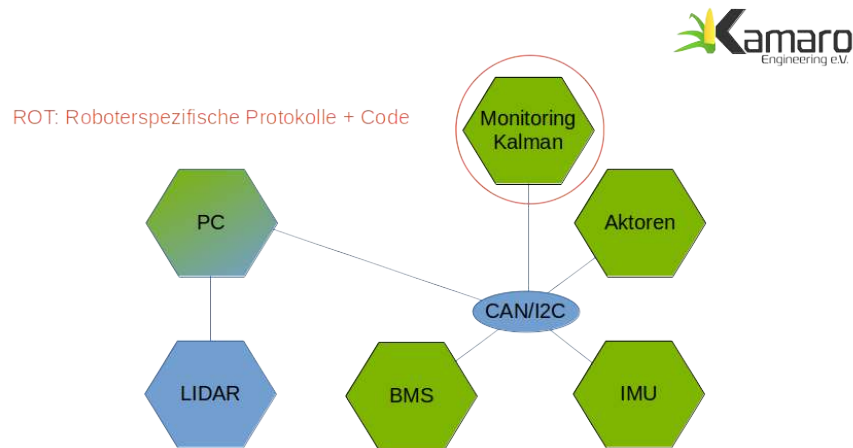


Abbildung 2.3: Übersicht des neuen Kabelbaums

Es bietet sich hier an, nicht nur die Middleware, sondern auch die Steckverbindungen zu standardisieren. Dabei hat sich Kamaro für spritzwasserfeste 7W2 Steckverbinder entschieden, die CONEC [Gmb20] uns freundlicherweise gesponsert hat. Ein Entwurf für die Busplatine stellt das Bild unten dar. Über den Stecker laufen zwei 24-Volt-Spannungsversorgungen und ein CAN-Bus. Das CAN-Protokoll eignet sich für unsere Zwecke bestens. Designed für die Kommunikation in Autos, entsprechen die erlaubten Leiterbahnlängen und Datenraten unseren Anforderungen. Weiterhin wird der CAN-Bus dezentral arbitriert. Somit benötigt man keinen Master. Eine Spannungsversorgung ist für Motoren gedacht und liefert bis zu 20 Ampere bei 24 Volt, die zweite ist für Elektronik wie Sensoren und Displays und liefert bis zu 5 Ampere bei 24 Volt. Die Spannung für die Motoren kommt direkt aus der Batterie. Die Stromversorgung für die Logik stellen Spannungsregler bereit, die die Spannung stabilisieren. Nach der Fertigstellung einer kleinen Testplattform setzen wir NOVA auf Beteigeuze um.

Anbindung der Hardware an ROS

Wie gesagt, erforderte bisher die Anbindung der Sensoren in ROS viel Arbeit, welche das neue Konzept erleichtert. Im alten Verkabelungskonzept setzte dies ein selbstgeschriebenes Protokoll um. Diese Lösung ist nur schwer zu warten und macht die Integration weiterer Sensoren aufwendig. RODOS setzt wie ROS auf eine publish-subscribe basierte Kommunikation und

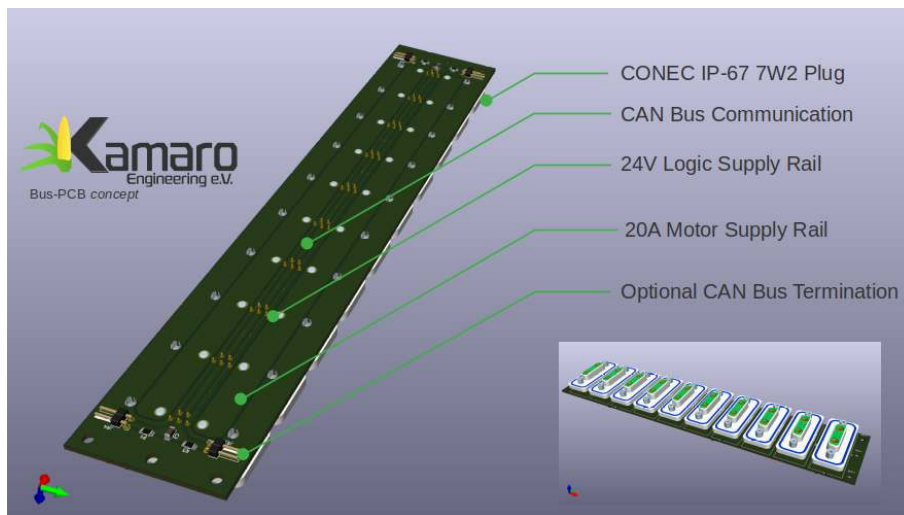


Abbildung 2.4: Busplatine – Konzeptdarstellung

ein von der Universität Würzburg entwickelter Übersetzer ist in der Lage, RODOS- in ROS-Nachrichten und umgekehrt zu übersetzen. Somit ist für die Anbindung eines Sensors kein Boilerplate-Code mehr notwendig. Nach der Fertigstellung einer RODOS-Node, die den Sensor ausliest und die Messwerte published, ist nur noch ein Eintrag im RODOS-ROS-Übersetzer notwendig, damit ROS-Nodes die Daten nutzen können.

Erste Tests

Bevor wir die alte Elektronik von Beteigeuze auseinandernehmen wollten, musste sichergestellt werden, dass die neue Software die Motoren der Chassis ansteuern kann. Die verwendeten Motortreiber von Dunkermotoren, sowie die Absolutencoder nutzen CANOpen für die Kommunikation. Damit bietet die Ansteuerung mehr Möglichkeiten, ist jedoch auch schwieriger als zum Beispiel über PWM. Glücklicherweise ist es uns gelungen, den alten Code dazu aus der Masterplatine wiederzuverwenden und in die neue Software einzubauen. Der Testlauf erster RODOS-Nodes, die über Publish-Subscribe kommunizieren und die Motoren ansteuern, ist im Bild unten zu sehen.

Mittlerweile sind die ersten Sensoren schon über RODOS auslesbar. Darunter fallen der populäre BNO055, ADCs, GPS (über NMEA) und optische Sensoren, über die wir die Geschwindigkeit der Roboter messen wollen. Wie bereits erwähnt, ist die Synchronisation der Messdaten keine einfache Aufgabe, aber hier hilft RODOS. RODOS stellt Puffer und FIFO-Warteschlangen bereit, sodass Programmierer*innen die Umsetzung verschiedener Lösungsansätze leicht fällt. Das Video unten zeigt, wie der BNO055 auf dem STM32F407 Discovery-Board verwendet wird, um einen Kompass zu bauen.

Seit der Entwicklung erster Nodes für Aktoren und Sensoren ist ein kleiner Versuchsroboter entstanden. Das Bild unten zeigt die Platine mit einem Raspberry-Pi und Motoren. Weiterhin wird eine neuere Version des Einstiegsroboters über eine kleine Busplatine verfügen. Damit können wir neue Komponenten am Schreibtisch entwickeln und später ohne Änderung an den Robotern für die Wettbewerbe anschließen.

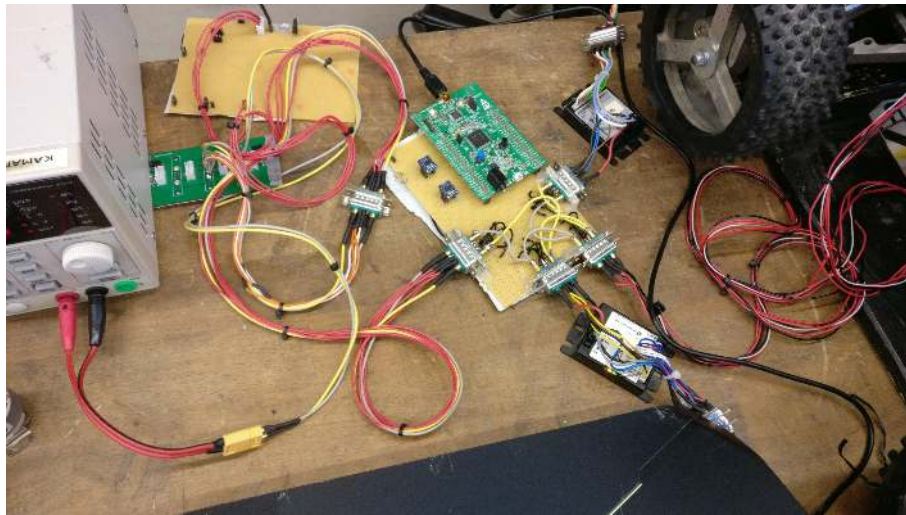


Abbildung 2.5: Der Roboter fährt. Wenn die Elektronik nebenan liegt.

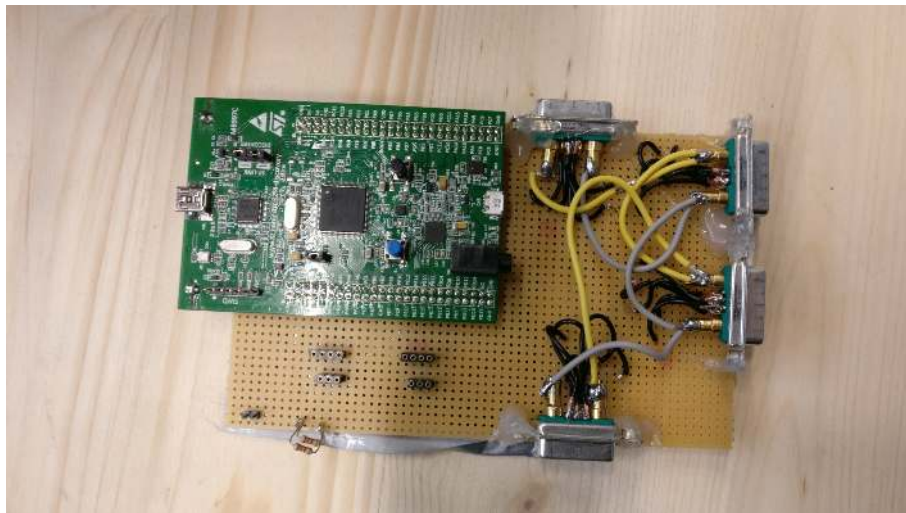


Abbildung 2.6: Erste Prototypplatine zum Ansteuern der Motoren

Vorteile von NOVA in der Teamarbeit

NOVA baut auf RODOS auf und somit ist auch hier eine Lernkurve vorhanden. Da aber die Middleware auf jeder Platine die gleiche ist, über eine schlanke API verfügt und das Gitlab-Repo umfassende Lernunterlagen bereitstellt, fiel uns die Einarbeitung leicht. Zusätzlich gibt es intern Tutorien für die Programmierung Kamaro-spezifischer RODOS-Nodes. Eine große Herausforderung besteht darin, die Hardware in der Software exakt anzusteuern. Bei Sensoren müssen Register über I2C richtig gesetzt werden, und Aktoren können eine Regelung erfordern. Hier hat NOVA die Arbeit für das Team Elektrotechnik schon bedeutend erleichtert: Die unterschiedlichen Komponenten entwickeln momentan kleine Gruppen zeitgleich — Jede fokussiert sich dabei auf ihre Komponente. Die zwei Platinen in den unteren Bildern wurden gleichzeitig mit einer dritten Version der Busplatine erstellt. NOVA vereinfacht den Entwicklungsprozess deutlich: Sobald Quellcode und Hardware eines Moduls mit einer Steckbrettschaltung im Test funktionieren, erfolgt die Realisierung als PCB. Diese wird über die Busplatine an den Roboter angeschlossen und ist sofort funktionsbereit.

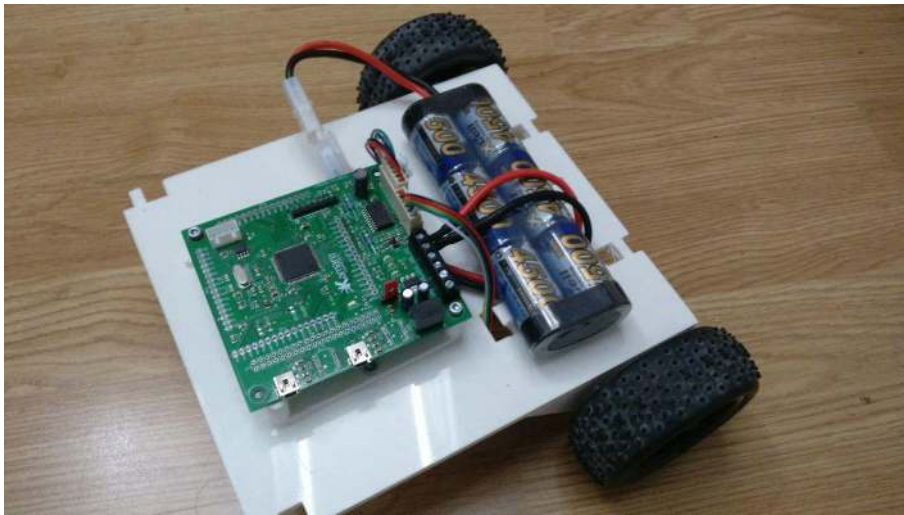


Abbildung 2.7: Miniroboter – Prototyp

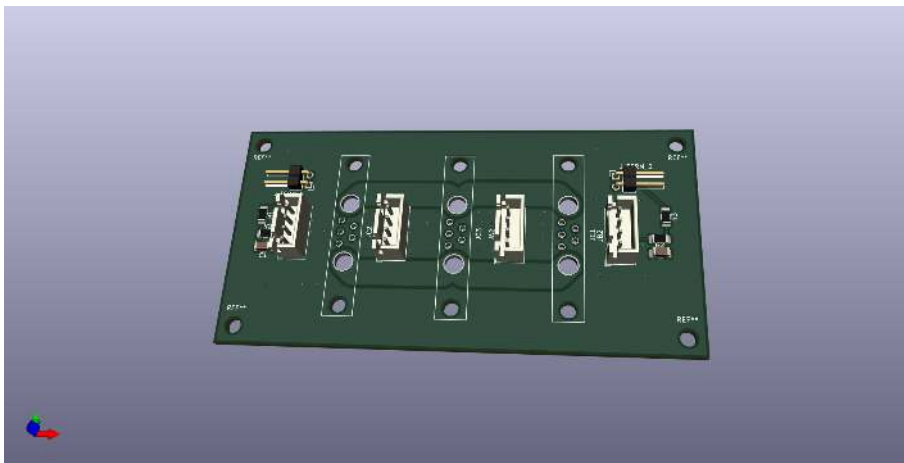


Abbildung 2.8: Die Mini-Version der Busplatine

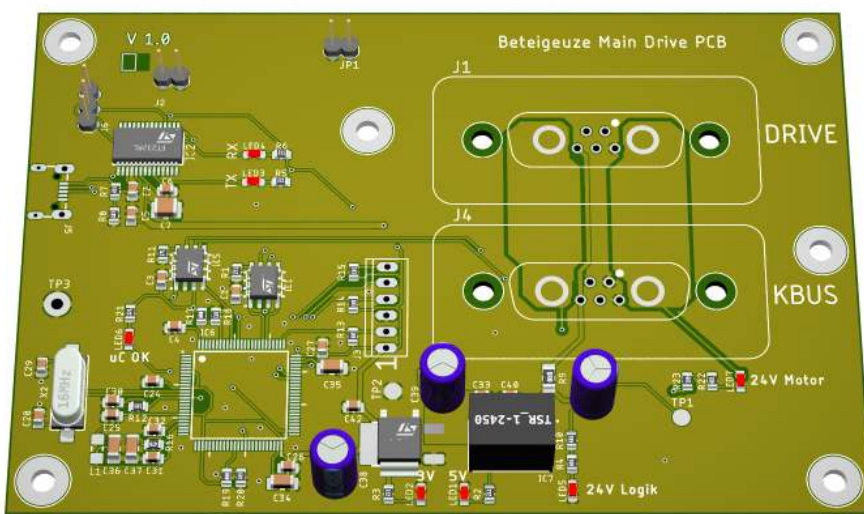


Abbildung 2.9: Platine zur Ansteuerung des Hauptmotors in der Mitte von Beteigeuze

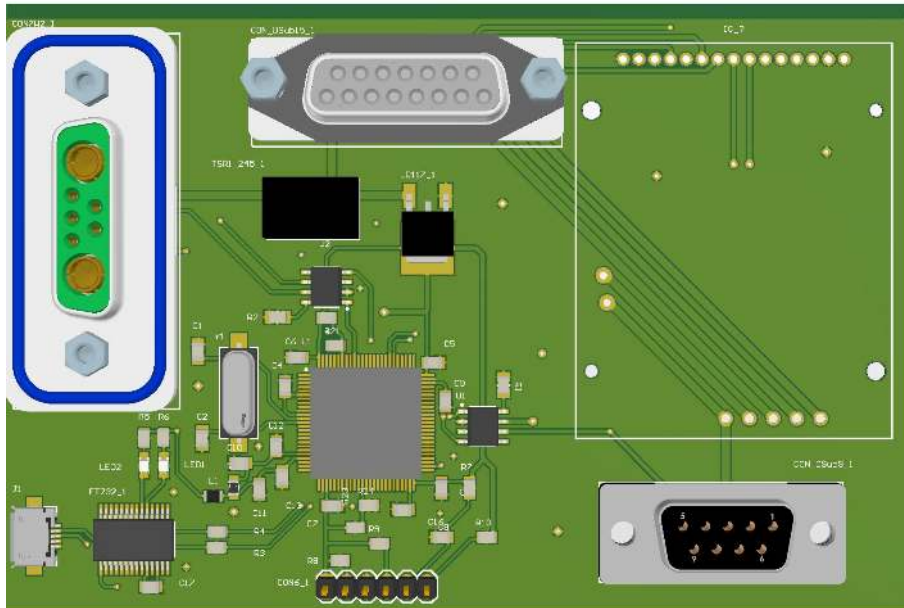


Abbildung 2.10: Platine zur Ansteuerung der Lenk-Servos von Beteigeuze

Zukünftige Entwicklungen

Kamaro Engineering e. V. arbeitet an vielen neuen Hardwaremodulen für modulare Roboter mit RODOS. Ziel ist, dass diese sich an der Busplatine zusammenstecken lassen, so wie Entwickler in ROS einzelne Nodes über Launchfiles kombinieren. Aber auch RODOS-Nodes sollen wiederverwendbar sein. Allen voran die Nodes, die Sensoren auslesen, da diese in der Programmierung sehr aufwendig sind. Wie bei den ROS-Nodes, die Sensoren über UART auslesen, soll ein Paketmanager RODOS-Nodes und Bibliotheken bereitstellen.

Eine weitere Entwicklung ist die Portierung von RODOS auf den deutlich günstigeren STM32F103. Aus alten RODOS-Versionen konnten wir den Code für Timer, UART und Kernel wieder integrieren.

Nun steht der Umbau von Beteigeuze kurz vor der Fertigstellung und wir sind auf die ersten Testfahrten gespannt! Sie auch? Dann folgen Sie unserem Blog für die neusten Updates zu Hardwarekomponenten, innovativen Algorithmen und der Zukunft der Feldrobotik!

Literaturverzeichnis

- [Gmb20] GMBH, CONEC ELEKTRONISCHE BAUELEMENTE: *CONEC Homepage*. <https://conec.com/de/>, 2020. [Online; accessed 20-August-2020].
- [KE20a] KAMARO-ENGINEERING: *Beteigeuze Robot*. <https://kamaro-engineering.de/roboter-2/beteigeuze/>, 2020. [Online; accessed 20-August-2020].
- [KE20b] KAMARO-ENGINEERING: *Dschubba Robot*. <https://kamaro-engineering.de/roboter-2/dschubba/>, 2020. [Online; accessed 20-August-2020].
- [MD09] MONTENEGRO, SERGIO und FRANK DANNEMANN: *Rodos-real time kernel design for dependability*. ESASP, 669:66, 2009.
- [MR09] MONTENEGRO, SERGIO und JOHN RICHARDSON: *RODOS operating system for Network Centric Core Avionics*. In: *Proceedings of Conference on Advances in Satellite and Space Communications, Colmar, France*, 2009.
- [QCG⁺09] QUIGLEY, MORGAN, KEN CONLEY, BRIAN GERKEY, JOSH FAUST, TULLY FOOTE, JEREMY LEIBS, ROB WHEELER und ANDREW Y NG: *ROS: an open-source Robot Operating System*. In: *ICRA workshop on open source software*, Band 3, Seite 5. Kobe, Japan, 2009.
- [RD19] RODOS-DEVELOPERS: *RODOS Version 200 Gitlab Repository*. <https://gitlab.com/rodos/rodos>, 2019. [Online; accessed 20-August-2020].
- [Wik20] WIKIPEDIA, AUTHORS: *RODOS Wikipedia Article*. [https://de.wikipedia.org/wiki/Rodos_\(Betriebssystem\)](https://de.wikipedia.org/wiki/Rodos_(Betriebssystem)), 2020. [Online; accessed 20-August-2020].