

Fachbereich: Informatik und Elektrotechnik

Embedded Systems / Internet of Things AG

Projekt: Autonomes Fahrzeug

Leiter der Veranstaltung:

Prof. Dr. Ralf Patz
Prof. Dr.-Ing. Wolfram Acker

Projektteilnehmer:

Bosse Hauschildt (935105)
Tobias Paulikat (935204)

Plagiatsklausel:

Die Studierenden erklären mit der Abgabe ihres Berichts, dass Sie alle verwendeten Quellen im Bericht korrekt erwähnt haben und dass dieser Bericht vollständig ihre eigene Arbeit ist. Nicht zulässig ist die Weitergabe eigener Berichte oder die Nutzung fremder Berichte.

Inhalt

1. Stand der Technik	1
2. Zielsetzung	1
3. Mechanik	1
4. Platinendesign	2
4.1 Motorplatinen.....	2
4.2 UART-Platine	3
5. Mikrocontroller	4
6. Raspberry Pi	5
7. Ausblick	5

1. Stand der Technik

Im ausgehenden Entwicklungsstand war das Fahrzeug nicht fahrtüchtig. Die Signalkette von der grafischen Benutzeroberfläche bis zur Ansteuerung eines Motors über einen Mikrocontroller war funktionsfähig und dient als Grundlage für den weiteren Ausbau zum fahrtüchtigen Fahrzeug.

2. Zielsetzung

Ziel des Projektes ist eine fahrtüchtige Plattform, des unter 1. genannten Stand der Technik weiterzuentwickeln. Das Fahrzeug soll zum Abschluss des Projektes selbstständig einige Meter zurücklegen können. Die Fahrtrichtung und eine Drehung des Fahrzeugs sollen über das Display einstellbar und ausführbar sein.

3. Mechanik

Die grundlegende Mechanik des Fahrzeugs ist zu Beginn schon vorhanden gewesen. Es bestand aus einer PVC-Platte mit Aluprofilen an denen 4 Motoren per Lochband fixiert wurden. Das Problem dabei war, dass die Motoren sehr instabil und wacklig an der Grundplatte befestigt waren. Um die Stabilität zu gewährleisten, wurde zur Verbesserung mit Hilfe von Solidworks eine 3D Zeichnung angefertigt und in der Zentralwerkstatt der Fachhochschule Kiel in Auftrag gegeben.

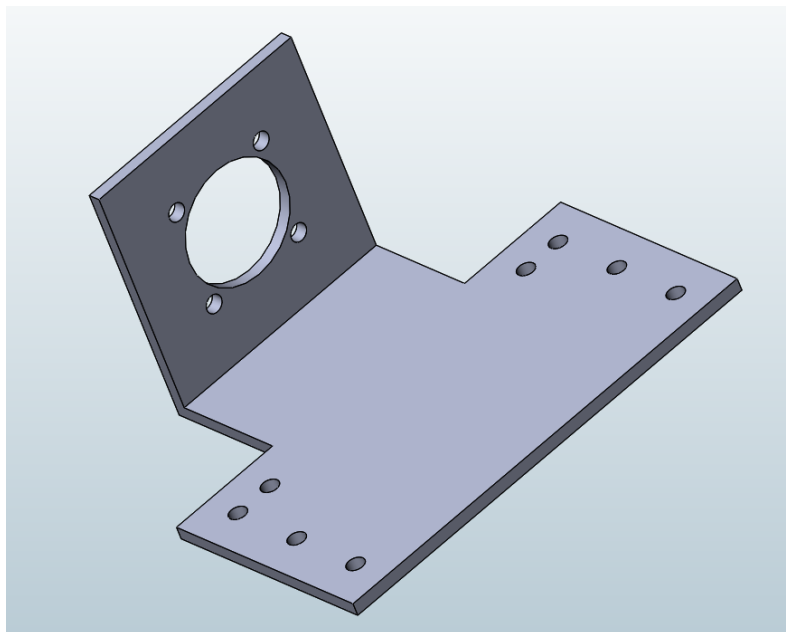


Abbildung 1: CAD Zeichnung der Motorhalterung

In der Zeichnung ist zu erkennen, dass ein Winkel entworfen wurde, der die Motoren mit den Profilen verbinden soll.

Dabei wird die Motorachse durch die 23mm Öffnung durchgeführt und mit 4 Schrauben fixiert. Das andere Ende des Winkels wird mit weiteren 6 Schrauben an den Ecken der Profile befestigt und dienen gleichzeitig als Eckverbindung für die Profile.

Die Winkel haben eine Materialstärke von 2mm, um der Hebelkraft ausreichend entgegenzuwirken.

Die Winkel wurden so designt, dass diese universell für jede Seite einsetzbar sind.

4. Platinendesign

Die vorherige Motorplatine wies zu Beginn noch einige Mängel auf. Ebenfalls fehlte bisher eine Platine zur Kommunikation mit Hilfe von UART zwischen dem eingesetzten Raspberry Pi und den Mikrocontrollern, welche per CAN Bus angesteuert werden.

Die Platinen wurden im Anschluss nach einer Prototypentestphase, extern bei einem Leiterplattenhersteller gefertigt.

4.1 Motorplatinen

Die existierende Motorplatine hatte ein Problem mit der Spannungsversorgung des Mikrocontrollers, welche dazu führte, dass dieser nicht ordnungsgemäß starten konnte. Als vorrübergehende Lösung wurde der verwendete Spannungsregler entfernt und provisorisch eine Spannungsversorgung über die Programmierschnittstelle hergestellt.

Eine Fehlersuche ergab ein Problem mit den Glättungsdröseln der analogen Spannungsversorgung des Mikrocontrollers. Da das Design jedoch den analogen Teil des Controllers nicht nutzt, konnten diese Dröseln durch einfache Verbindungen ersetzt werden.

Daraufhin konnte das Platinendesign der Motorsteuerung überarbeitet werden, bei der im gleichen Zuge noch weitere Optimierungsmaßnahmen durchgeführt wurden.

Darunter fiel unter anderem:

- Reduzierung der kapazitiven Last (Verringerung der Pufferkondensatoren)
- Verbessertes Leiterbahndesign
- Leistungsreduzieren durch SMD H-Brücken
- Leistungsfähigeren Spannungsregler
- Schraubterminals zur Spannungsversorgung und zum Anschluss der Motoren
- Auswahlschalter zur Festlegung der Motorposition

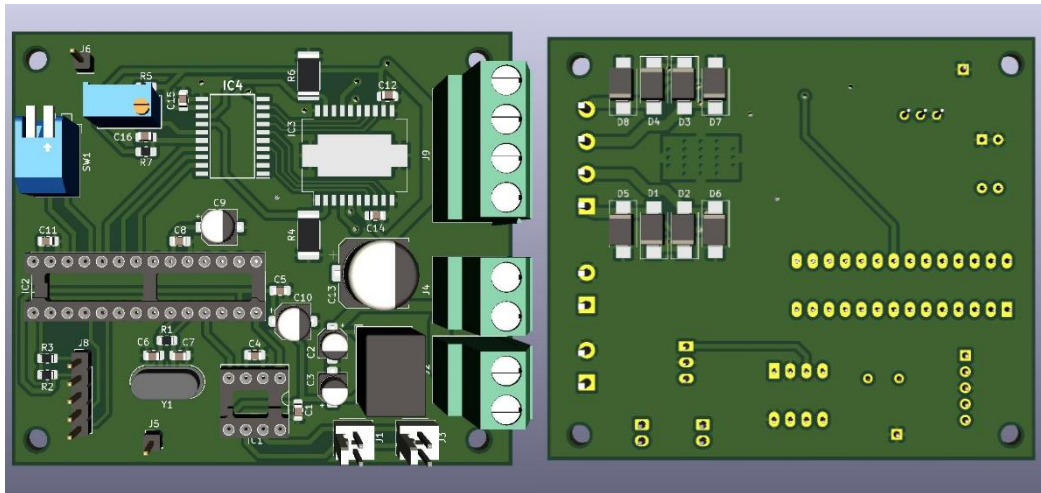


Abbildung 2: Vorder- und Rückseite Motorplatinen

4.2 UART-Platine

Die nichtexistierende UART Platine musste von Beginn an entworfen werden. Es bestand nur eine vorläufige Steckschaltung, die als Grundlage und Kontrolle der Funktion diente.

Benötigte Funktionen der Platine:

- UART-Kommunikation zwischen Raspberry Pi und Mikrocontroller
- CAN-Bus Kommunikation zu den Motorplatinen
- Spannungsversorgung von 6V auf 5V für den Raspberry Pi
- Als HAT aufsteckbar auf den Raspberry Pi

Da der vorhandene Akku eine Nennspannung von 6 Volt besitzt und diese im laufenden Betrieb stark schwankt und bis 4,5V reduziert sein kann, muss gewährleistet sein, dass der Raspberry Pi dauerhaft mit 5V versorgt wird. Hierfür wurde ein spezieller DC/DC Wandler herausgesucht und mit auf der Platine untergebracht.

Um die Platine als HAT zu realisieren, konnte über die GPIO Stiftheiste des Raspberry Pi eine dazu passende Buchsenleiste verwendet werden. Diese ermöglicht es die Platine auf den Raspberry Pi aufzustecken. Über die genannte Steckverbindung, konnte somit auch eine UART Kommunikation und die Spannungsversorgung umgesetzt werden.

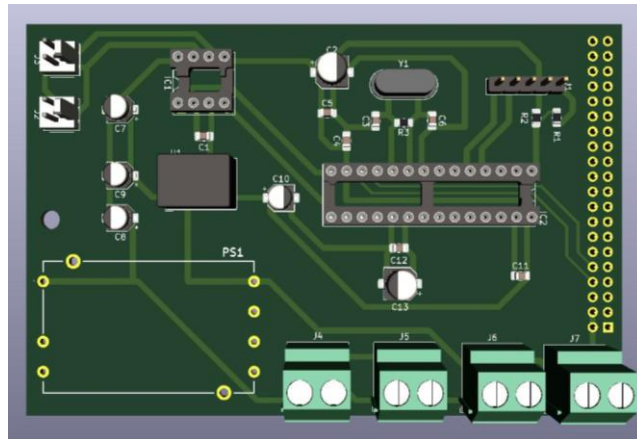


Abbildung 3: UART Platine

5. Mikrocontroller

Da der bisherige Programmcode so aufgebaut war, dass im Programmcode selbst die Motorposition festgelegt werden muss und somit vor jeder Programmierung der Mikrocontroller kontrolliert werden muss, an welcher Position der Mikrocontroller sitzt, wurde diese Funktion durch die bereits erwähnten Auswahlschalter ersetzt. Die durch die Auswahlschalter belegten Eingänge, werden zum Programmstart einmalig abgefragt und nach dem anliegenden Pegel zugeordnet. Hierfür wurden die Eingänge mit einem internen Pull Up Widerstand versehen.

Das bisherige Anfahrverhalten der Motoren war nicht linear. Um ein besseres Verhalten zu implementieren, wurde eine konstante Anstiegszeit festgelegt und über Summenbildung eine konstante Beschleunigung auf integriert.

Im Allgemeinen wurde anschließend der Programmcode weiter aufgeräumt und einige Testmethoden entfernt.

Der bereits vorhandene UART Programmcode wurde des Weiteren in ein eigenständiges Projekt überführt. Da dieser bereits Vollfunktionsfähig war, mussten hier keine Anpassungen durchgeführt werden.

6. Raspberry Pi

Der Raspberry Pi soll weiterhin als zentrale Steuereinheit verwendet werden.

Dazu wurde die in QT Entworfenene Benutzeroberfläche erweitert und die vorhandenen Buttons mit Funktionen belegt.

Das bisherige Programm konnte nur eine bestimmte fest einprogrammierte Schrittzahl und Fahrtrichtung vorgeben. Durch die Erweiterung des Codes kann nun die Fahrtrichtung und die Schrittzahl angepasst werden.

Es gibt derzeit 4 Fahrtrichtungen, welche über ein Dropdown Menü ausgewählt werden:

- Vorwärts
- Rückwärts
- Linksdrehung
- Rechtsdrehung

Die Schrittzahl kann über einen Slider von 0 bis 20000 Schritte eingestellt werden.

Mit Hilfe des Start Buttons kann die eingestellte Fahrt begonnen werden.

7. Ausblick

Um das Fahrzeug weiter zu verbessern, sollte die Verkabelung und die Montage der einzelnen Platinen auf dem Chassis umgesetzt werden. Die grafische Nutzeroberfläche des Raspberry Pi ist derzeit sehr einfach gestaltet und müsste für eine gute Anwendbarkeit überarbeitet werden.

Die Fahrzeugbewegung ist derzeit auf einzelne Fahrtabschnitte beschränkt. Um diese komplexer gestalten zu können, müsste eine hintereinander Kettung von Fahrtabschnitten umgesetzt werden.

Aktuell ist kein differentiellles Fahren möglich. Das Fahrzeug kann derzeit eine Drehung nur im Stand ausführen und ist dadurch beschränkt manövrierfähig.