



---

## Bachelorarbeit

# Hardware und Software Design für ein autonomes Modellfahrzeug

Julian-Benedikt Scholle  
6. Oktober 2014

### **Gutachter**

Prof. Dr. Jörg Kaiser  
Dipl.-Inform Christoph Steup

Otto-von-Guericke-Universität Magdeburg  
Fakultät für Informatik  
Universitätsplatz 2  
39106 Magdeburg

---

---

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>3</b>
1.1 Carolo Cup . . . . .	3
1.2 Aufbau der Arbeit . . . . .	4
<b>2 Anforderungsanalyse</b>	<b>5</b>
2.1 Ausgangssituation . . . . .	5
2.2 Anforderungen laut Regelwerk . . . . .	6
2.2.1 Fahrzeugantrieb und Energieversorgung . . . . .	6
2.2.2 Fahrzeugantrieb und Energieversorgung . . . . .	6
2.2.3 RC-Modus . . . . .	6
2.2.4 Signalleuchten . . . . .	6
2.3 Auswertung des Regelwerks . . . . .	6
2.4 Andere Anforderungen . . . . .	7
2.4.1 Anforderungen durch Bewertungskriterien . . . . .	7
2.4.2 externe Anforderungen . . . . .	7
2.5 Zusammenfassung . . . . .	7
2.6 Änderung der Anforderungen . . . . .	8
<b>3 Konzept</b>	<b>9</b>
3.1 Grundlegender Aufbau . . . . .	9
3.2 Der Mikrocontroller . . . . .	10
3.3 Motortreiber . . . . .	10
3.4 Spannungsversorgung . . . . .	10
3.5 Anschluss an das Pandaboard . . . . .	11
<b>4 Umsetzung</b>	<b>12</b>
4.1 Motoransteuerung . . . . .	12

4.1.1	Auswahl der Mosfets . . . . .	13
4.1.2	Mosfettetreiber . . . . .	13
4.2	Motorstrommessung am Shunt . . . . .	17
4.2.1	Problem . . . . .	18
4.2.2	Anforderungen . . . . .	19
4.2.3	Bestimmung des Filtertyps . . . . .	19
4.2.4	Die Filterschaltung . . . . .	20
4.2.5	Dimensionierung des Verstärkers . . . . .	20
4.2.6	Anforderungen an den Filter . . . . .	21
4.2.7	Filterentwurf . . . . .	21
4.3	Beleuchtung . . . . .	28
4.4	Distanzsensoren . . . . .	29
4.4.1	Messprinzip . . . . .	29
4.4.2	Probleme der GP2D Sensoren . . . . .	30
4.4.3	Auswertung des Messsignales . . . . .	31
4.5	Odometrie . . . . .	31
4.5.1	Hall-Sensor . . . . .	31
4.5.2	Gabellichtschranke . . . . .	32
4.6	Auslegung der Stromversorgung . . . . .	33
4.6.1	Stromverbrauch der Komponenten . . . . .	33
4.6.2	Auswahl des Reglers . . . . .	35
4.7	Das Layout . . . . .	36
4.8	Software . . . . .	37
4.8.1	Software auf dem Mikrocontroller . . . . .	37
4.8.2	Client Programm auf der Recheneinheit . . . . .	37
4.8.3	Übertragungsprotokoll . . . . .	38
<b>5</b>	<b>Evaluierung</b>	<b>41</b>
5.1	Strommessung . . . . .	41
5.2	Spannungsversorgung . . . . .	43
5.3	Inertialsensor . . . . .	45
5.4	Infrarotsensoren . . . . .	48
5.5	Odometrie . . . . .	49
5.6	Stromverbrauch . . . . .	51
<b>6</b>	<b>Ausblick und Fazit</b>	<b>53</b>

# 1

## Einleitung

Für den Hochschulwettbewerb „Carolo-Cup“ der Technischen Universität Braunschweig soll ein autonom fahrendes Fahrzeug im Maßstab von 1:10 entwickelt werden. Im Rahmen der Arbeit wird der Entwicklungsprozess der Motortreiberplatine des Fahrzeugs veranschaulicht. Dabei werden auch die Probleme eines Projekts mit sich dynamisch ändernden Anforderungen gezeigt.

---

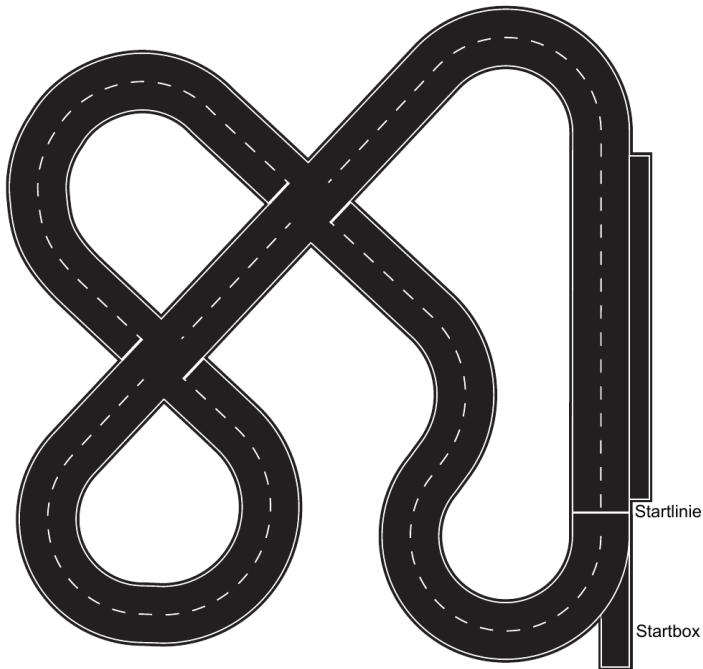
### 1.1 Carolo Cup

Der „Carolo-Cup“ ist ein jährlicher Hochschulwettbewerb der Technischen Universität Braunschweig. Dieser bietet Studenten die Möglichkeit, sich mit der Entwicklung und Umsetzung von autonomen Modellfahrzeugen auseinander zu setzen [Leha]. Ziel des Wettbewerbes ist es ein möglichst kostengünstiges und energieeffizientes Modellfahrzeug im Maßstab 1:10 zu entwickeln. Das Fahrzeug muss dabei möglichst schnell und fehlerfrei bestimmte Aufgaben bewältigen. Die Aufgaben werden dabei in statische und dynamische Disziplinen unterteilt.

In den statischen Disziplinen muss das Team sein Fahrzeugkonzept vor einer Jury, bestehend aus Experten aus Industrie und Forschung, verteidigen. Dabei wird auf die Hardware- und Softwarearchitektur sowie Energiebedarf und Herstellungskosten eingegangen. Des Weiteren müssen die Lösungskonzepte zur Bewältigung der dynamischen Disziplinen vorgestellt werden.

Die dynamischen Disziplinen bestehen aus mehreren Szenarien, dem parallelen Einparken, einem einfachen Rundkurs sowie einem Rundkurs mit Hindernissen. Ein möglicher Rundkurs ist in Abbildung[1.1] zu sehen.

*Abbildung 1.1: Möglicher Rundkurs [Lehb]*



---

## 1.2 Aufbau der Arbeit

Die Arbeit gliedert sich in 5 Teile. In der Anforderungsanalyse wird die Ausgangssituation der Arbeit beschrieben und analysiert, welche Funktionen benötigt werden. Im Kapitel 3, dem Konzept, wird der Aufbau der Platine beschrieben und es werden grundlegende Komponenten festgelegt. Das vierte Kapitel beschreibt die Umsetzung der in der Anforderungsanalyse geforderten Funktionen. Kapitel 5 beschäftigt sich mit der Evaluierung einiger als wichtig erachteten Komponenten und bewertet den Stromverbrauch. Im letzten Kapitel wird das Ergebnis der Arbeit bewertet und mögliche Verbesserungen vorgeschlagen.

# 2

# Anforderungsanalyse

## 2.1 Ausgangssituation

Die Ausgangsposition der Arbeit stellt sich folgendermaßen dar. Das Fahrzeug basiert auf einem Fahrgestell vom Typ "Tamiya TT-01-Type E", welches bereits über einen Motor und einem Motortreiber verfügt. Auf Grund der Herkunft der Komponenten, dem Modellbau, ist keine Dokumentation der elektrischen Komponenten verfügbar. Aus diesem Grund kann der originale Motortreiber nicht verwendet werden. Da das Fahrzeug über bereits vorhandene 4-Zellen-Lithium-Polymer-Akkus (14,4V Nennspannung) versorgt werden soll, um die Anschaffung teurer Akkus zu vermeiden, hätte dieser so oder so nicht verwendet werden können. Für das Fahrzeug wurde bereits eine Treiberplatine in Form eines Prototyps gefertigt. Diese verfügt über ein Crumb123 Mikrocontroller Modul (At90can128), welches einen Motortreiber vom Typ "L298 DUAL FULL-BRIDGE DRIVER" ansteuert. Des Weiteren verfügt der Prototyp über einen 3 A Schaltregler von Texas Instruments, welcher Mikrocontroller und zwei Pandaboard mit Energie versorgt. Der Schaltregler befindet sich durch den Betrieb zweier Pandaboard jedoch bereits an seiner Leistungsgrenze, da ursprünglich nur der Betrieb eines Pandaboard vorgesehen war. Der verwendete Motortreiber wurde genutzt, da er durch die Verwendung in anderen Projekten bereits vorhanden war, jedoch ist dieser nur für Ströme von bis zu 4 A ausgelegt [STM14]. In einem Versuch wurden für den im Fahrzeug vorhandenen Motor Ströme von bis zu 20 A gemessen. Durch die Unterdimensionierung des Treibers überhitzt dieser bereits nach wenigen Minuten Fahrbetrieb. Da der vorhandene Motor jedoch weiter verwendet werden soll, muss hier ein leistungsfähiger Ersatz entwickelt werden. Des Weiteren fehlen dem Prototyp wichtige Anschlüsse für weitere Sensoren und Beleuchtung, welche nun integriert werden sollen.

Bei der Entwicklung der neuen Treiberplatine sollen, um Kosten zu sparen, möglichst viele vorhandene Komponenten verwendet werden. Dazu gehören Mikrocontroller vom Typ "Atmel At90can128", welche in großer Stückzahl vorhanden sind. Benötigte Sensoren in Form eines Inertialsensors (Sparkfun SEN-10724) und Sharp GP2D Sensoren werden ebenfalls

vom Lehrstuhl zur Verfügung gestellt. Des Weiteren sind Sortimente von SMD Widerständen und Kondensatoren in der Größe 805 vorhanden, sodass diese Bauform bei der Entwicklung der Platine bevorzugt wird.

---

## **2.2 Anforderungen laut Regelwerk**

Um am „Carolo-Cup“ teilnehmen zu können ist ein regelkonformes Fahrzeug nötig, weshalb nun ein Auszug aus den Anforderungen an das Fahrzeug kurz aufgelistet und ausgewertet wird. Alle Anforderungen können im Regelwerk des „Carolo-Cup“ nachgelesen werden [Lehb]

### **2.2.1 Fahrzeugantrieb und Energieversorgung**

Laut Regelwerk sind alle Teams zur Verwendung eines elektrischen Antriebs verpflichtet. Die Anzahl der angetriebenen Räder ist nicht vorgeschrieben. Des Weiteren muss das Auto durch Akkus mit Strom versorgt werden. Die Übertragung von Daten ist während der Dauer der Disziplinen nicht gestattet

### **2.2.2 Fahrzeugantrieb und Energieversorgung**

Es ist eine Zweiradlenkung der Vorderachse vorzusehen. Die übrige Gestaltung des Fahrwerks bleibt den Teams überlassen. Als technische Ausprägung ist ausschließlich die Achsschenkellenkung zugelassen.

### **2.2.3 RC-Modus**

In Notsituationen muss es möglich sein das Fahrzeug mit Hilfe einer Funkfernbedienung anzuhalten und manuell zu steuern. Eine solche Notsituation tritt ein, wenn das Auto seine Aufgabe aufgrund eines Fahrfehlers oder anderem Fehlverhalten nicht mehr autonom fortführen kann. Der RC-Modus muss per Fernbedienung eingeschaltet und ausgeschaltet werden, bei Aktivierung des RC-Modus muss das Fahrzeug unverzüglich angehalten werden. Während des Wettbewerbs darf die Geschwindigkeit des Autos  $0,3 \frac{m}{s}$  nicht überschreiten. Da das 2,4-GHz Band bereits durch die Vor Ort genutzte Kameratechnik belegt ist, können diese Frequenzen nicht für den RC-Modus genutzt werden. Der RC-Modus muss durch eine blaue Leuchte an der höchsten Stelle des Fahrzeuges angezeigt werden, welche mit einer Frequenz von 1-Hz blinkt.

### **2.2.4 Signalleuchten**

Durch die Anlehnung des Wettbewerbes an den realen Straßenverkehr muss das Auto über alle in echten Autos vorhandene Signalleuchten besitzen. Dazu gehören drei rote Bremslichter am Heck des Autos, sowie jeweils zwei gelbe Blinker rechts und links am Fahrzeug. Die Blinkfrequenz der Blinker muss 1-Hz betragen.

---

## **2.3 Auswertung des Regelwerks**

Der vorhandene Prototyp entspricht bereit den Anforderungen bezüglich des elektrischen Antriebs, jedoch mit den bereits beschriebenen Un-

zulänglichkeiten. Um diese Unzulänglichkeiten zu korrigieren, muss ein leistungsfähiger Motortreiber integriert werden. Die Lenkung des Autos kann von einem einfachen Servomotor vorgenommen werden. Ein solcher wurde bereits in Form eines IQ-620CMG des Herstellers Hype angeschafft. Damit das Auto die im RC-Modus nötigen Funksignale auswerten kann, muss ein Empfänger integriert werden. Dieser wird jedoch über USB zur Verfügung gestellt und ist deswegen nicht Teil dieser Arbeit. Des Weiteren muss das Auto mit den nötigen Signalleuchten ausgestattet sein, dazu gehören Blinker rechts und links jeweils vorne und hinten, sowie drei Bremsleuchten an der Rückseite und eine weitere Leuchte welche den RC-Modus anzeigt. Der Vollständigkeit halber kommt hier noch die Frontbeleuchtung hinzu.

---

## 2.4 Andere Anforderungen

### 2.4.1 Anforderungen durch Bewertungskriterien

Abseits des Regelwerkes ergeben sich weitere Anforderungen durch die Bewertungskriterien des Wettbewerbes. Dabei handelt es sich um nicht-funktionale Anforderungen. So muss das Team während der statischen Disziplinen ihr Gesamtkonzept präsentieren. Schwerpunkte dabei sind, Hardware- und Software-Architektur sowie Energiebedarf und Herstellungskosten [Lehb]. Daraus entstehen weitere Anforderungen: Energieeffizienz und Herstellungskosten.

### 2.4.2 externe Anforderungen

Folgende Anforderungen werden durch das Team an die Platine gestellt. Die Integration einer Inertialsensorik (Sparkfun SEN-10724) zur Inertialnavigation, Anschlüsse Infrarotsensoren vom Typ Sharp GP2D120 und GP2D120, die Integration einer Odometrie zur Geschwindigkeitsmessung, sowie eine 5V Stromversorgung zum Betreiben eines Pandaboard. Die Platine soll ein ROS-Interface auf einem Pandaboard zur Verfügung stellen, mit welchem die Aktorik gesteuert und die Sensorwerte ausgelesen werden können. Das Auto soll über ein vorhandenes Netzteil betrieben werden können (ca. 20V).

Des Weiteren muss die Platine in eine dafür vorgesehene Lücke des Fahrzeugs passen. Diese Lücke hat Maße von 13 cm mal 5.4 cm.

---

## 2.5 Zusammenfassung

Um den Anforderungen zu genügen, muss ein Motortreiber mit einer Belastbarkeit von mindestens 20 Ampere integriert werden. Ein Mikrocontroller zum Steuern der Aktorik inklusive des Motors, sowie zum Auslesen der Sensorsignale muss zur Verfügung gestellt werden. Anschlüsse für zwei Sharp Sensoren vom Typ "GP2D120" oder "GP2D12" sowie weitere Anschlüsse für den Inertialsensor, Beleuchtung, Servomotor werden ebenfalls benötigt. Es muss eine Odometrie zur Messung der Geschwindigkeit integriert werden. Es muss eine effiziente Energieversorgung aller

Bauteile zur Verfügung gestellt werden, welche auch zwei Pandabboards stabil betreiben kann. Die maximale Eingangsspannung muss mindestens 20 Volt betragen, um das Auto auch über das Netzteil betreiben zu können. Da das Fahrzeug energieeffizient arbeiten soll, ist eine Messung des Motorstromverbrauches essenziell, um das Fahrverhalten bezüglich der Effizienz bewerten zu können. Zu guter Letzt muss die Platine über eine Schnittstelle mit einem Pandaboard kommunizieren und seine Daten dort über ein ROS-Interface zur Verfügung zu stellen.

---

## 2.6 Änderung der Anforderungen

Nach der erfolgreichen Teilnahme am “Carolo-Cup Junior” im Februar 2014, begann die Weiterentwicklung des Konzepts. Während der Entwicklung kamen jedoch einige Flaschenhälse zum Vorschein, sodass in der Projektphase die Hardwareplattform geändert werden musste. Die Rechenleistung der Pandabboards stellte sich als unzureichend heraus und es wurden mehr Distanzsensoren gewünscht. Die Pandabboards wurden nach der Absprache mit dem Team durch ein Intel NUC vom Typ D34010WYB ersetzt. Laut Datenblatt [Int] besitzt der NUC einen Weitbereichseingang zur Spannungsversorgung, dieser ist für 12-24 Volt zugelassen. Sodas der NUC direkt an den 14.4 V der Akkus und den 20 V des Netzteils betrieben werden kann. Um Verwirrungen zu vermeiden wird im Folgenden legendlich von Recheneinheit gesprochen, wenn es nicht relevant ist ob es sich um NUC oder Pandaboard handelt.

# 3

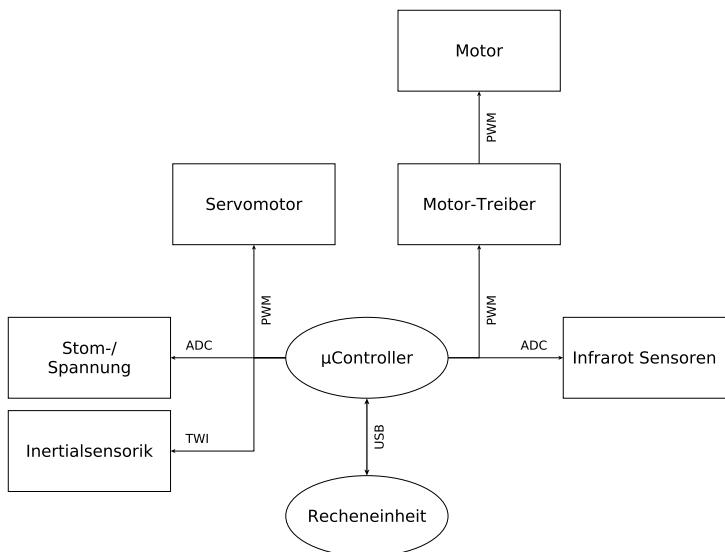
## Konzept

 Konzeptkapitel sinnvoll?

### 3.1 Grundlegender Aufbau

Die Treiberplatine ist der zentrale Punkt für das Einsammeln aller Messwerte und die Ansteuerung der Aktorik. Dabei übernimmt sie sowohl die Energieversorgung der Komponenten als auch die Kommunikation mit der darüber liegenden Recheneinheit. Herzstück der Platine ist dabei ein Mikrocontroller, an welchen über diverse Schnittstellen die Aktorik bzw. Sensorik angeschlossen ist. Die Platine selber kommuniziert über USB mit der Recheneinheit und stellt dieser eine Schnittstelle zum Auslesen der Messwerte und Einstellen der Stellgrößen für die Aktorik zur Verfügung. Weitere Aufgaben der Platine sind die Überwachung von Zuständen wie z.B. der Akkusspannung und dem Motorstrom. Eine Übersicht über die Sensorik bzw. Aktorik und ihre Anbindung ist in Abbildung [3.1] zu sehen.

Abbildung 3.1: Konzept



Die fertig entwickelte Platine soll als Prototyp gefertigt werden, deshalb

wird darauf geachtet, dass alle verwendeten Bauteile mit Hand bestückt werden können. Die Größe aller SMD Bausteine wird aufgrund des vorhandenen Sortiments auf 0805 beschränkt.

### 3.2 Der Mikrocontroller

Als Mikrocontroller wird ein “Atmel At90can12” [Atm] verwendet, da dieser vor Ort vorhanden ist und die nötigen Funktionen wie PWM-Kanäle und einen Analog-/Digital-Wandler mitbringt und über die nötige Leistung und Speicherkapazität verfügt (128KB). Der Mikrocontroller kann durch ein einzelnes 5V-Netz mit Strom versorgt werden.

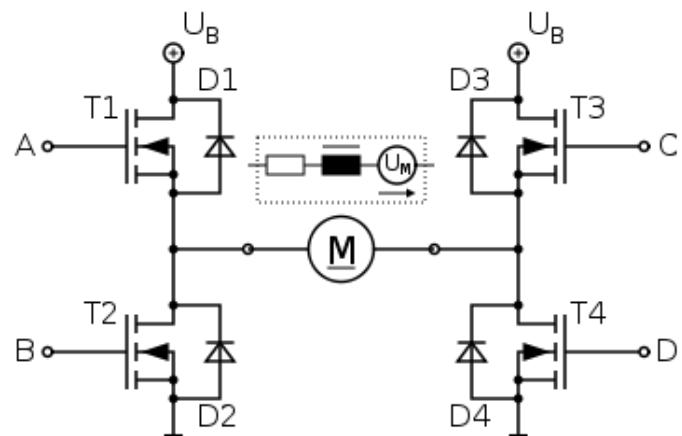
### 3.3 Motortreiber

Da der Motortreiber eine hohe Belastbarkeit von mindestens 20 Ampere aufweisen soll und vollintegrierte Motortreiber in dieser Leistungsklasse praktisch nicht zu bekommen sind, wird der Motortreiber als Vierquadrantensteller mit diskreten Mosfets ausgeführt.

#### Vierquadrantensteller:

*“Ein Vierquadrantensteller besteht aus einer elektronischen H-Brückenschaltung aus vier Halbleiterschaltern, meist aus Transistoren, welche eine Gleichspannung in eine Wechselspannung variabler Frequenz und variabler Pulsbreite umwandeln kann. Vierquadrantensteller in der Energietechnik können auch Wechselspannungen unterschiedlicher Frequenzen in beiden Richtungen ineinander umwandeln.” [?]*

Abbildung 3.2:  
Vierquadrantensteller [Wikc]



### 3.4 Spannungsversorgung

Da der Mikrocontroller und alle vorhandenen Sensoren eine Betriebsspannung von 5 Volt aufweisen, wird auf ein anderes Spannungsnetz verzichtet. Da die Platine somit nur mit einer einzigen Spannungsversorgung ausgestattet ist, werden alle weiteren Bauteile so ausgewählt, dass sie an einem 5V-Netz betrieben werden können.

---

### 3.5 Anschluss an das Pandaboard

Da die verwendeten Pandaboard sind bereits über GPIO-Anschlüsse verfügen, wird auf eine USB Anbindung verzichtet, stattdessen wird die Platine direkt an die SPI-Anschlüsse eines der Pandaboard angeschlossen. Da die Verbindung mechanisch stabil sein muss und trotzdem einfach zu entfernen sein soll, wird hier eine RJ45-Buchse verwendet. Diese stellt insgesamt 8 Leitungen zur Verfügung. Um eine größere Flexibilität zu erreichen wird durch diese zusätzlich die UART geleitet. Die einzelnen Pins sind wie folgt belegt:

Tabelle 3.1: Belegung der RJ45 Buchse

Pin	Funktion [V]
1	RX
2	MISO
3	TX
4	MOSI
5	SCK
6	5 Volt
7	Reset (AVR)
8	GND

#### Hinweis:

*Da die Pandaboard jedoch im Laufe der Projektphase durch einen Intel NUC ersetzt wurden, welcher nicht über einen externen SPI-BUS verfügt, musste die Verbindung der Platine anders realisiert werden. Mit Hilfe der ebenfalls aufgeführten UART konnte mit einem UART-USB Adapter eine Verbindung zum Intel NUC hergestellt werden. Da die Software erst im späteren Projektverlauf geschrieben wurde, nutzt sie die Anbindung über den UART-USB Adapter.*

# 4

## Umsetzung

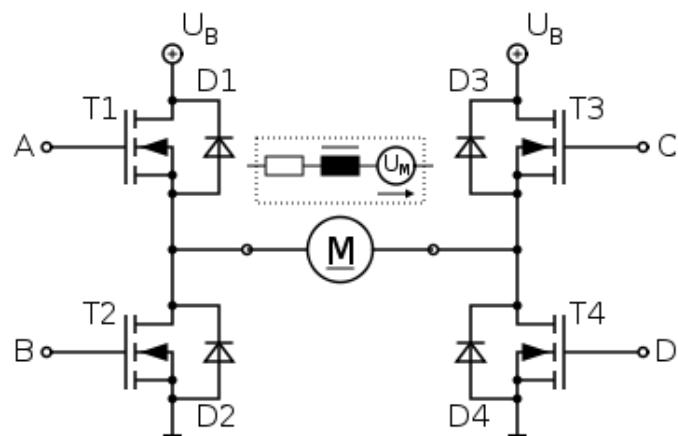
Text verbessern....oder  
entfernen

In diesem Kapitel werden die technischen Hintergründe der einzelnen Komponenten erläutert und ihre Umsetzung beschrieben. ....

### 4.1 Motoransteuerung

Wie bereits im Konzept beschrieben, wird der Motorteiber als diskreter Vierquadrantensteller ausgeführt. Auf Grund der hohen Belastbarkeit und leistungslosen Ansteuerung werden meist p-Kanal-Mosfets als Halbleiterschalter genutzt. Um die beiden oberen Mosfets (T1/T3) durchzuschalten ist auf Grund des fehlenden Massepotentials eine Gatespannung oberhalb der Betriebsspannung nötig. Diese wird meist mittels Bootstrapping zur Verfügung gestellt. Da das simultane Durchschalten der übereinander liegenden Mosfets zu einem Kurzschluss führen würde, muss dies durch eine Schutzschaltung verhindert werden. Um all diese Funktionen zur Verfügung zu stellen gibt es bereits fertige Mosfettetreiber, welche das Schaltungsdesign enorm vereinfachen.

Abbildung 4.1:  
Vierquadrantensteller [Wikc]



minimale Max Betriebss-  
spannung im Anforderungs-  
analyse/Konzept festlegen

#### 4.1.1 Auswahl der Mosfets

Die Auswahl der Mosfets wird durch den geforderten Strom von 20A und der maximalen Betriebsspannung von 20V eingegrenzt. Weitere Kriterien wie die Gate-Source-Spannung, welche zum Durchschalten der Mosfets benötigt wird, sind auf Grund der Tatsache, dass ein Mosfettetreiber verwendet werden soll nicht von Belang, da dieser die benötigten Spannungen generiert. Für die Anwendung in einem Vierquadrantensteller ist die wahrscheinlich wichtigste Größe der Drain-Source-Widerstand im Einschaltzustand  $R_{DS(ON)}$ . Ein kleiner  $R_{DS(ON)}$  ist von Vorteil und führt zu einer kleinen Verlustleistung an den Mosfets. Mosfets mit einem hervorragenden  $R_{DS(ON)}$  von nur 12 mΩ sind die FDD6690A von Fairchild Semiconductor [Sem]. Weitere Kenngrößen dieser sind eine maximale Drain-Source-Spannung von 30V sowie eine Belastbarkeit von 46A, eine gute Kühlung vorausgesetzt. Aufgrund der kleinen Gate Ladung von 13nC eignen sie sich gut für hohe Schaltfrequenzen. Aufgrund ihres niedrigen  $R_{DS(ON)}$  werden sie für die Platine genutzt.

#### 4.1.2 Mosfettetreiber

##### Verfügbarkeit

Mosfettetreiber gibt es in vielen Ausführungen, unter anderem als "Single Channel High Side Driver", "Half Bridge Driver", "Full Bridge Driver" und "3 Phase Driver". Da für den verbauten DC-Motor eine Vollbrücke nötig ist, um den Motor in alle Richtungen zu betreiben, werden an dieser Stelle ausschließlich "Full Bridge Driver" untersucht.

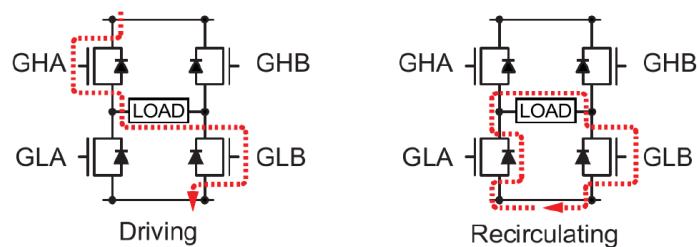
Eine Tabelle auf Mikrocontroller.net [mik14] zeigt eine Auswahl an verfügbaren Mosfettreibern. Dort sind zwei "Full Bridge Driver" aufgeführt, welche für dieses Projekt passend sind. Allerdings fällt die Entscheidung auf einen anderen Treiber, dem Allegro A3941 [All].

##### Allegro A3941

Der Allegro A3941 ist für Betriebsspannungen von 5,5V bis 50V geeignet und liegt damit in der Spezifikation des Projekts. Des Weiteren verfügt der Motor über einen integrierten 5V-Regulator und kann somit ohne Spannungsregulator am Akku betrieben werden. Über zwei Ausgänge der Treibers können diverse Fehler ausgelesen werden. Auch sind alle gängigen Schutzschaltungen wie z.B. ein Kurzschlusschutz enthalten. Daher ist er hervorragend für das Projekt geeignet.

Der Treiber lässt sich in verschiedenen Modi betreiben:

Abbildung 4.2: Slow decay, diode recirculation, high-side PWM [All]



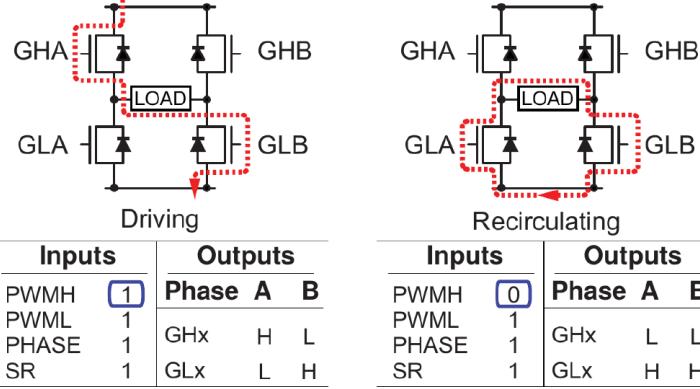
Inputs		Outputs	
		Phase A	Phase B
PWMH	1	GHx	H L
PWML	1	Ghx	H L
PHASE	1	GLx	L H
SR	0		

Inputs		Outputs	
		Phase A	Phase B
PWMH	0	GHx	H L
PWML	1	Ghx	H L
PHASE	1	GLx	L H
SR	0		

Konfiguration: PWML=1, PHASE=1, SR=0 und PWM an PWMH (high-side PWM)

Bei aktivierten PWMH fließt der Strom durch den GHA-Mosfets über den Motor und dann über den GLB-Mosfet. In diesem Modus wird der Motor angetrieben. Wenn PWML deaktiviert ist, zirkuliert der vom Motor induziert Strom durch GLB und durch die interne Diode von GLA, der Motor wird dadurch gebremst.

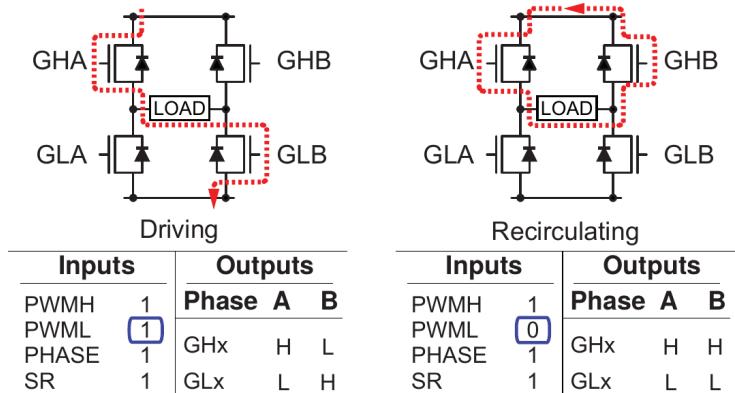
Abbildung 4.3: Slow decay, SR active, high-side PWM [All]



Konfiguration: PWML=1, PHASE=1, SR=1 und PWM an PWMH (high-side PWM)

Bei aktivierten PWMH fließt der Strom durch den GHA-Mosfets über den Motor und dann über den GLB-Mosfet. In diesem Modus wird der Motor angetrieben. Wenn PWML deaktiviert ist, zirkuliert der vom Motor induziert Strom durch GLB und durch GLA. Der Motor wird durch den niedrigeren Innenwiderstand des Mosfets stärker gebremst als in der vorherigen Konfiguration. Dabei ist darauf zu achten, dass beinahe die gesamte vom Motor induzierte Spannung über den beiden Mosfets (GLA/GLB) abfällt, was zu einer starken Hitzeentwicklung führen kann.

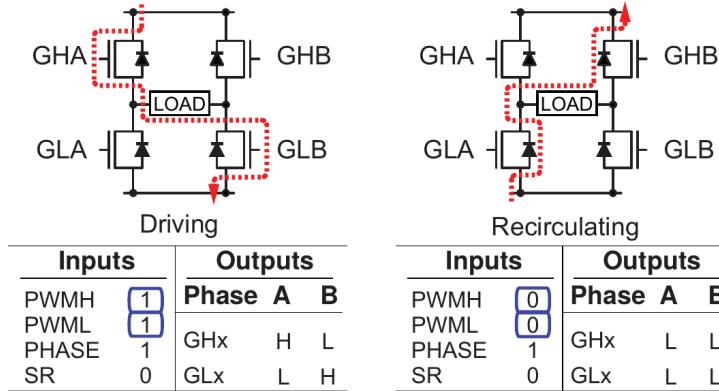
Abbildung 4.4: Slow decay, SR active, low-side PWM [All]



Konfiguration: PWMH=1, PHASE=1, SR=1 und PWM an PWML (low-side PWM)

Diese Konfiguration entspricht im Grunde den beiden vorherigen, nur dass das PWM-Signal an den unteren Mosfets anliegt. Der SR-Pin entscheidet wieder darüber ob im "Bremsmodus" die internen Dioden genutzt werden (SR=0) oder nicht (SR=1).

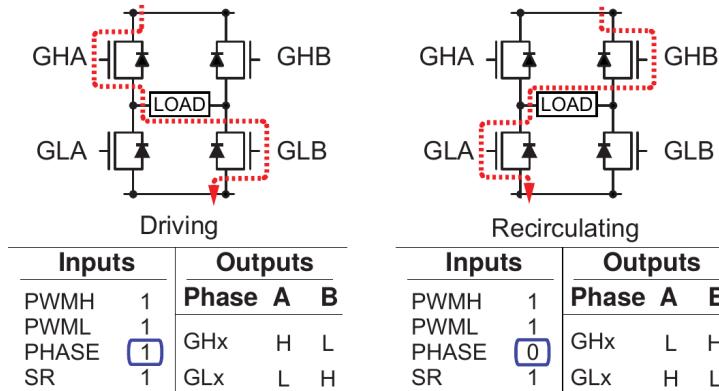
Abbildung 4.5: Fast decay, diode recirculation [All]



Konfiguration: PWMH=1, PWML=1, PHASE=1, SR=1

In dieser Konfiguration werden die oberen und unteren Mosfets gleich geschaltet. Im "Bremsmodus" führt das dazu, dass der induzierte Motorstrom nicht über die Mosfets zirkulieren kann. Der Strom fließt stattdessen zurück in die Spannungsquelle, was abhängig von der Spannungsquelle zu Schäden führen kann. Wird die Schaltung jedoch an einem geeigneten Akku betrieben, ist es so möglich die Energie zu nutzen und damit den Akku zu laden.

Abbildung 4.6: Fast decay, SR active, full four-quadrant control [All]



Diese Konfiguration zeigt den Einfluss des PHASE-Pins. Liegt am PHASE-Pin 1 an, fließt der Strom von links nach rechts. Liegt 0 an, fließt er von rechts nach links. Mithilfe des PHASE-Pins wird also die Polung des Motors festgelegt.

### Schaltplan

In Folgenden wird geklärt wie sich die Beschaltung des Allegro A3941 bestimmen lässt. Die Schaltung in Abb. 4.7 entspricht dabei den Vorgaben im Datenblatt.

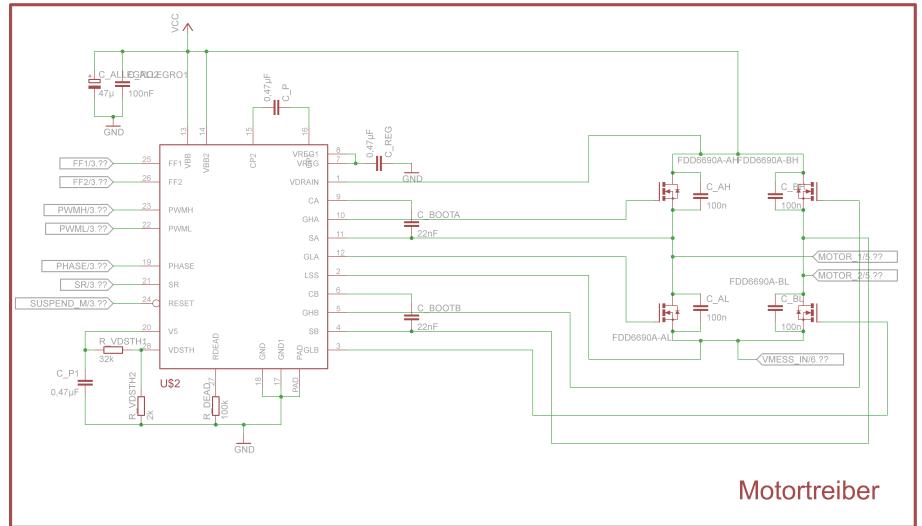


Abbildung 4.7: Schaltplan Allegro  
A3941

$R_{DEAD}$  legt die Totzeit zwischen den Mosfetumschaltungen fest. Wird sie zu niedrig gewählt, können in jeder PWM-Periode Kurzschlüsse auftreten. Wird sie zu hoch gewählt, beschränkt man das Tastverhältnis unnötig, was die Motorleistung reduziert. Da wir "Low Gate Charge" Mosfets nutzen und die Einschalt- bzw. Ausschaltzeit der Mosfets hauptsächlich von dieser Größe abhängt, ist diese Zeit sehr klein. Eine genaue Bestimmung dieser Zeit ist sehr aufwändig, deswegen wird hier ein mittlerer Wert von  $100\text{k}\Omega$  gewählt, dieser soll sich laut Datenblatt zwischen 3 und  $240\text{k}\Omega$  befinden.

Als Nächstes werden die Größen der Bootstrapkondensatoren  $C_{BOOTA}$  und  $C_{BOOTB}$  bestimmt. Diese sind für eine ordnungsgemäße Funktion der Schaltung richtig zu wählen. Zu kleine Kondensatoren verhindern ein Durchschalten der oberen Mosfets, was zu einer nicht funktionsfähigen Schaltung führt. Zu große Werte hingegen verringern das maximale Tastverhältnis des PWM Signals, da die Aufladung der Kondensatoren Zeit in Anspruch nimmt. Laut Datenblatt hat sich folgende Faustformel gut bewährt:

$$C_{BOOT} = \frac{Q_{GATE} \cdot 20}{V_{BOOT}}$$

$V_{BOOT}$  entspricht dabei in etwa  $V_{REG}$ , welche laut Datenblatt [All] 13V beträgt. Die Gateladung  $Q_{GATE}$  der FDD6690A Mosfets beträgt typischerweise 13 nC [Sem]. Damit ergibt sich für  $C_{BOOT}=20\text{nF}$ . Es werden demnach die nächst größeren 22nF als Bootstrapkondensatoren genutzt.

Die Spannung  $V_{DSTH}$  am VDSTH-Eingang soll der Spannung entsprechen die maximal über jeden Mosfet abfallen darf, bevor der Treiber einen Kurzschuss detektiert und abschaltet. Da der maximale Strom durch unseren Motor 20 Ampere beträgt, lässt sich diese Spannung, einfach berechnen. Der Widerstand der Mosfest im Einschaltzustand beträgt in unserem Fall ( $U_{GS} = 13\text{ V}$ ) etwas weniger als  $12\text{ m}\Omega$  bei einem Strom von 20 A ergibt sich damit eine Spannung von  $12\text{ m}\Omega \cdot 20\text{ A} = 0.24\text{ V}$ . Da wir den Motor über sein gesamtes Leistungsspektrum nutzen wollen, muss diese Spannung etwas darüber gewählt werden.

Da der Strom in VDSTH nur minimal ist (ca  $10\ \mu\text{A}$ ), kann diese Spannung über einen einfachen Spannungsteiler zur Verfügung gestellt werden. Ein Spannungsteiler mit Widerständen zu  $2\ \text{k}\Omega$  und  $32\ \text{k}\Omega$  erzeugt uns zusammen mit den 5V aus dem V5 Ausgang eine Spannung von  $\frac{2\ \text{k}\Omega}{32\ \text{k}\Omega+2\ \text{k}\Omega} \cdot 5\ \text{V} = 0.24\ \text{V}$ . Der Spannungsteiler wird durch  $R_{VDSTH1}$  und  $R_{VDSTH2}$  gebildet.

Die Größe des  $C_{REG}$  Kondensators hängt direkt mit der Größe der Bootstrapkondensatoren zusammen. Er soll ca 20 mal größer als  $C_{BOOTA}$  bzw  $C_{BOOTB}$  gewählt werden, sodass hier ein  $470\text{nF}$  Kondensator gewählt wird.

Das Datenblatt empfiehlt  $100\ \text{nF}$  entkoppelungs Kondensatoren zwischen Drain und Source eines jeden Mosfets. Diese wurden in Form von  $C_{AH}, C_{AL}, C_{BH}$  und  $C_{BL}$  verbaut. Alle restlichen Größen werden direkt dem Datenblatt entnommen.

Da der verwendete AVR Mikrocontroller nur über begrenzte PWM fähige Anschlüsse verfügt, wird lediglich der PWM-Kanal für die oberen Mosfets (PWMH) an einen PWM Ausgang des AVR angeschlossen. Durch diese Entscheidung werden Betriebsmodi, welche ein Gleichschalten der Mosfets erfordern, erschwert und können nur durch ein Software PWM realisiert werden. Der Vorteil es Betriebsmodus “Fast decay, diode recirculation” ist, dass es möglich ist Bremsenergie des Fahrzeugs zurück in den Akku zu speisen. Da die Akkus jedoch mit einer integrierten Ladeelektronik ausgestattet sind, liegt die Vermutung nahe, dass ein Laden des Akkus so nicht möglich ist. Deshalb wird der Betriebsmodus im Rahmen dieser Arbeit nicht untersucht.

Alle weiteren Anschlüsse des Allegros werden direkt an Pins von PORTC am Mikrocontroller angeschlossen. Da die Fehlerausgänge als Open-Collector ausgeführt sind, werden die internen Pullup-Widerstände des Mikrocontrollers genutzt.

 [Verweis auf Bild von uC](#)

## 4.2 Motorstrommessung am Shunt

Da innerhalb des Wettbewerbes Energieeffizienz wichtig ist, ist es notwendig den aktuellen Stromverbrauch zu kennen. Es wird davon ausgegangen, dass der Stromverbrauch der Recheneinheit und Sensoren konstant ist, bzw.. sich mit der fertigen Software über die Zeit nicht wesentlich ändert. Da der Motor einen großen zu erwartenden Verbrauch hat, ist es wichtig seinen Verbrauch zu bestimmen. Mithilfe dieser Informationen ist es möglich verschiedene Fahrregler Auslegungen im Aspekt des Verbrauches zu vergleichen. Um den Verbrauch zu bestimmen, ist es notwendig die anliegende Spannung und den Strom durch den Motor zu messen. Die H-Brücke wird dabei als Teil des Motors betrachtet, daher entspricht Spannung über den Motor der Akkuspannung. Diese lässt sich einfach über einen Spannungsteiler an den ADC des Mikrocontrollers anschließen. Als Werte für den Spannungsteiler werden  $1.8\ \text{k}\Omega$  und  $10\ \text{k}\Omega$  gewählt.

## voltage divider for battery voltage measurement

Abbildung 4.8: Spannungsteiler  
zur Akkuspannungsmessung

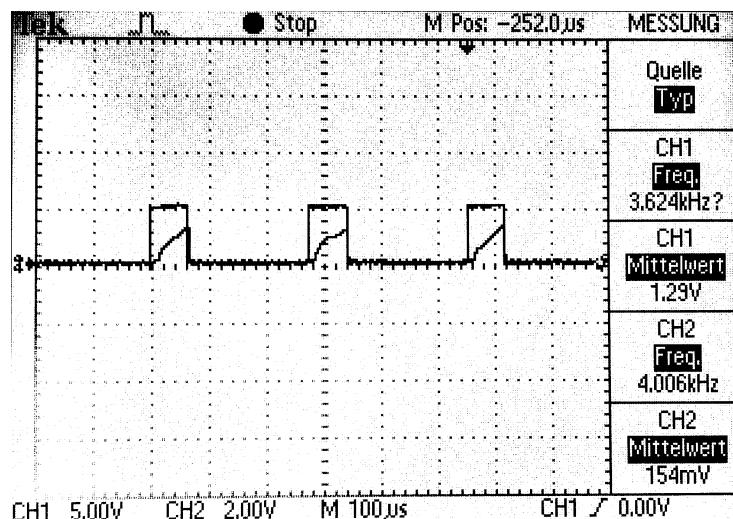


Da der Innenwiderstand des ADC sehr groß ist ( $100\text{ m}\Omega$ ), kann der Spannungsteiler als unbelastet betrachtet werden. Damit ergibt sich das Teilungsverhältnis zu  $\frac{R_{STL}}{R_{STH}+R_{STL}} = 0,153$ . Die maximal messbare Akkuspannung beträgt dadurch  $\frac{5\text{ V}}{0,153} = 32.7\text{ V}$ . Die Messung des Stroms gestaltet sich wesentlich aufwändiger und wird im Folgenden erläutert.

### 4.2.1 Problem

Die Messung des Motorstroms ist mit einem Problem behaftet, da der Motor über eine Pulsweitenmodulation angesteuert wird. Der Verlauf des Stroms ist pulsierend, abhängig von der Frequenz und dem Tastgrad der Pulsweitenmodulation. In Abb. 4.9 zu erkennen ist der Verlauf des PWM Signals (obere Kurve) und des Stroms, welcher nach einer  $1 - e^t$  Funktion ansteigt. Ohne weitere Filterung des Signals wäre eine extrem hohe Abtastfrequenz des ADC nötig, um den genauen Stromfluss zu messen. Da für eine Verbrauchsmessung der Mittelwert des Stroms interessant ist, kann dieser mit Hilfe eines Tiefpasses erzeugt werden.

Abbildung 4.9: Spannung am  
Shunt + PWM



Eine kostengünstige Variante den Strom zu messen ist den Spannungsabfall über einen Shuntwiderstand zu bestimmen. Die Größe des Shunt-

widerstandes sollte nur so klein wie nötig gewählt werden. Ein zu kleiner Shuntwiderstand benötigt einen starken Messverstärker, um das Signal auswerten zu können. Die Qualität und Genauigkeit dieses Signals nimmt allerdings mit steigender Verstärkung ab.

Nach kurzer Recherche wurde die Größe des Shuntwiderstands auf R2512 (ca 6.35 mm x 3.00 mm) begrenzt. Messshunts in dieser Größe gibt es problemlos bis zu einer Belastbarkeit von 2 Watt. Bei einem maximalen Motorstrom von 20A ergibt sich somit ein maximaler Spannungsabfall von 100 mV. Nach dem Ohmschen Gesetz ergibt sich so eine Shunt-Größe von  $\frac{0.1\text{V}}{20\text{A}} = 0.005 \Omega$ . Shuntwiderstände in der Größe sind problemlos zu bekommen. Da es sich hier um eine Worst Case Rechnung handelt, wird der zusätzliche Widerstand des Shuntwiderstandes und der damit verringerte Strom bewusst ignoriert.

Der Shunt wird direkt unter der H-Brücke des Motortreibers gegen Masse geschaltet. So wird eine Messung gegen Masse durchgeführt. Die so über den Shuntwiderstand gemessene Spannung könnte dann über den ADC Eingang des Mikrocontrollers eingelesen werden. Vorher jedoch muss das Signal gefiltert und verstärkt werden.

#### 4.2.2 Anforderungen

Die maximale Auflösung des Mikrocontrollers soll ausgenutzt werden. Der ADC des Mikrocontrollers arbeitet mit einer Auflösung von 10 Bit und einer Referenzspannung von 5V. Um die Auflösung des ADC auszunutzen muss das Signal, aufgrund des geringen Spannungsabfalls, verstärkt werden.

Als Anforderung ergibt sich außerdem, dass der maximale Ripple des Endsignals kleiner ist als der Quantisierungsfehler des ADC. So ist es möglich auf eine zusätzliche digitale Filterung weitgehend zu verzichten. Die kleinst mögliche zu erfassende Spannung des ADC beträgt  $\frac{5}{2^{10}} = 4,88\text{mV}$ . Diesen Wert sollte der Ripple des Endsignales nicht überschreiten. Aus einem möglichst kleinem Ripple resultiert eine möglichst hohe Filterordnung bzw. eine niedrige Grenzfrequenz. Allerdings soll  $U_{DC}$  einer Änderung des Mittelwertes, also einer Änderung des Tastverhältnisses, möglichst schnell folgen. Diese Anforderung widerspricht der Vorherigen, sodass ein Kompromiss gefunden werden muss.

#### 4.2.3 Bestimmung des Filtertyps

Da zum Verstärken des Signals aktive elektronische Elemente notwendig sind, z.B. ein Operationsverstärker, wird an dieser Stelle gleich ein aktiver Filter verwendet. Dieser gibt uns die Möglichkeit das Messsignal zu verstärken und gleichzeitig zu filtern. Da das Signal im optimalen Fall eine Gleichspannung darstellt, müssen die hochfrequenten Anteile des Signales herausgefiltert werden. Dies geschieht mit Hilfe eines Tiefpasses. Es gibt im Grunde zwei übliche aktive Tiefpässe, den Tiefpass mit Mehrfachgegenkopplung und den Sallen-Key Filter. Ersterer verwendet einen invertierenden Verstärker, dieser invertiert das Messsignal. Da der Mikrocontroller jedoch nur mit positiven Spannungen umgehen kann, müsste man hier mit einer negativen Referenzspannung arbeiten, was den Schaltungsaufwand unnötig vergrößern würde. Der Sallen-Key Tiefpass benutzt einen nicht invertierenden Verstärker, welcher diesen Nach-

teil nicht hat. Sodass ab dieser Stelle ein Sallen-Key Tiefpass entworfen wird.

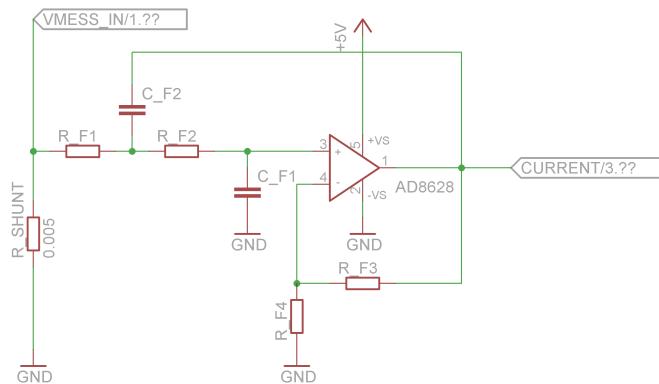
**Quellen (verweise auf übliche Filtertypen)**

#### 4.2.4 Die Filterschaltung

Wie im vorherigen Abschnitt diskutiert, wird hier ein Sallen-Key Tiefpass entworfen. Zum Entwurf der Schaltung wurde Eagle genutzt. Genutzt wird ein AD8628 Rail-to-Rail Operationsverstärker von Analog Devices, da dieser direkt an einer einzelnen 5V Spannung betrieben werden kann und über hervorragende Eigenschaften verfügt [Dev]. Die fertige Schaltung kann in Abb. 4.10 betrachtet werden. Um Konflikte im Schaltplan zu vermeiden, wurden den Variablen ein "F" vorangestellt.

Sallen-Key low-pass with 27x amplification  
for motor current measurement

Abbildung 4.10: Sallen-Key  
Tiefpass mit Shunt



#### 4.2.5 Dimensionierung des Verstärkers

In bisherigen Rechnungen wurde ein maximaler Spannungsabfall von 100mV am Shunt errechnet. Da der Messbereich des ADC voll ausgenutzt werden soll, ist es nötig das Messsignal zu verstärken. Hierzu wird ein nicht-invertierender Verstärker genutzt. Da der Messbereich des ADC bis 5V reicht, wird hier eine 50-fache Verstärkung angestrebt.

Die Beschaltung, welche den Verstärkungsfaktor des Sallen-Key Tiefpass festlegt, entspricht der eines nicht-invertierenden Verstärkers:

$$v = 1 + \frac{R_3}{R_4}$$

$$50 = 1 + \frac{R_3}{R_4}$$

$$49 \cdot R_4 = R_3$$

Wobei  $R_{F4} = 47\text{k}\Omega$  und  $R_{F3} = 1\text{k}\Omega$  gewählt werden, was eine Verstärkung von 48 ergibt. Die Größen sollten nicht zu klein gewählt werden, da der Operationsverstärkerausgang sonst zu stark belastet wird. Ein Gesamt-widerstand von 5 kΩ resultiert stets in einem Stromverbrauch von unter 1 mA und ist akzeptabel.

#### 4.2.6 Anforderungen an den Filter

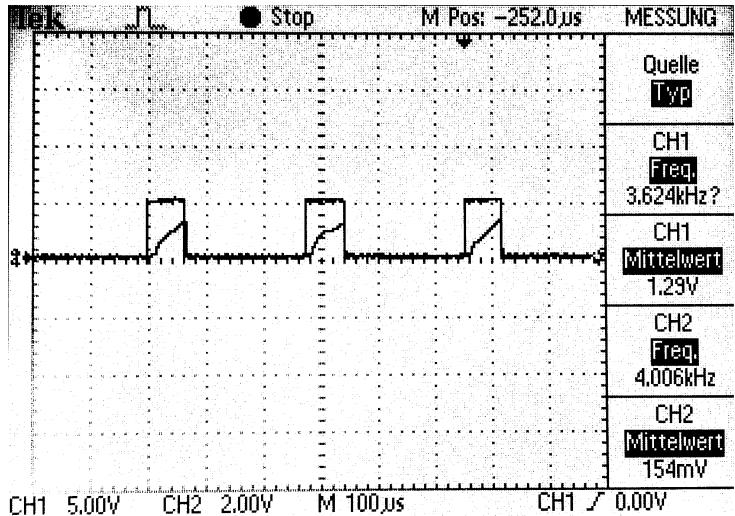


Abbildung 4.11: Spannung am Shunt + PWM

Da dem Messsignal wie in Abbildung 4.11 zu erkennen, die PWM Frequenz zu Grunde liegt, wird sich bei der Dimensionierung des Filters einer Idee nach [Alt08] bedient, nach der die maximale Amplitude des Ripple der Grundschwingung bei einem Tastverhältnis von 0.5 entspricht. Die Amplitude der Grundschwingung ergibt sich aus dem ersten Koeffizienten der Fourierreihe einer Rechteckschwingung.

$$A_1 = K \cdot \frac{1}{\pi} [\sin(\pi p) - \sin(2\pi(1 - \frac{p}{2}))] \quad (4.1)$$

Wobei  $p$  dem Tastverhältnis und  $K$  der maximale Amplitude des Ursprungssignal entspricht [Alt08].  $K$  entspricht den errechneten 100mV multipliziert mit dem Verstärkungsfaktor 48, also 4.8V. Das Tastverhältnis  $p$  wird zu 0,5 angenommen. Mit (4.1) ergibt sich für die Amplitude der Grundschwingung  $A_1 = K \cdot \frac{2}{\pi} = 3,056V$ .  $A_1$  soll auf  $< 4,88mV$  gedämpft werden. Als Sperrfrequenz  $\Omega_s$  wird hier die PWM Frequenz angesetzt. Für  $H(\omega = 2\pi f_{PWM})$  gilt also:

$$H(\omega = 2\pi f_{PWM}) \leq \frac{4,88mV}{3,056V} \hat{=} 20 \cdot \log\left(\frac{4,88mV}{3,056V}\right) = -55,9dB \quad (4.2)$$

Um die Komplexität der Schaltung gering zu halten, wird im Folgenden von den üblichen Konventionen zur Dimensionierung von Filtern abgewichen. Statt eine fixe Grenzfrequenz festzulegen und die benötigte Filterordnung zu bestimmen, wird die Filterordnung vorgegeben und die Grenzfrequenz variiert.

#### 4.2.7 Filterentwurf

##### Bestimmung der Filtercharakteristik

Die Charakteristik eines Filters entscheidet über seinen Frequenzgang. Es gibt viele dieser Filtercharakteristiken, eine Auswahl an häufig verwendeten Charakteristiken wird hier verglichen.

Der *Butterworth*-Filter besitzt einen maximal flachen Verlauf des Frequenzganges im Durchlassbereich und eine monoton verlaufende Dämpfung.

fung im Sperrbereich. Leider hat der Butterworth-Filter nur eine geringe Flankensteilheit im Sperrbereich (20dB/Dekade pro Ordnung). Ein Butterworth-Filter 1. Ordnung entspricht einen einfachen RC-Filter.

Der *Tschebyscheff*-Filter hat eine höhere Flankensteilheit als der Butterworth-Filter, allerdings entsteht beim Tschebyscheff-Filter Welligkeit im Durchlassbereich, welche mit höherer Ordnung zunimmt. Durch die Welligkeit im Durchlassbereich würde ein zusätzlicher Ripple im Signal entstehen, weshalb der Tschebyscheff-Filter nicht für den geforderten Filter geeignet ist.

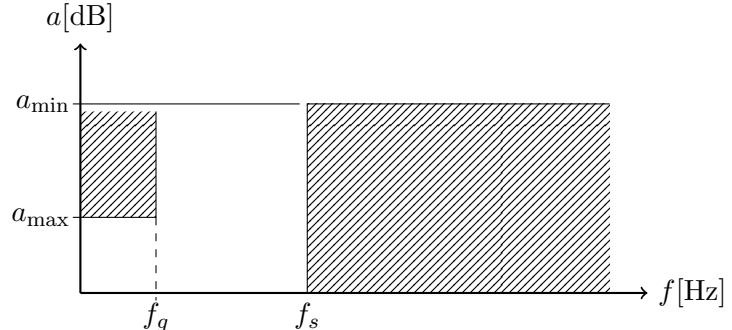
Der *Bessel-Filter* hat den Vorteil einer konstanten Gruppenlaufzeit, hat dafür aber eine noch geringere Flankensteilheit als der Butterworth-Filter. Da eine konstante Gruppenlaufzeit für den geforderten Filter nicht von Vorteil ist, da das Endsignals einer Gleichspannung entsprechen sollte, ist der Butterworth-Filter die bessere Wahl.

### Bestimmung der Grenzfrequenz

Zur Bestimmung der Grenzfrequenz des Filters ist es nötig die PWM-Frequenz der Motoransteuerung zu kennen. Diese kann jedoch nur im Betrieb optimal bestimmt werden, da sie auch von subjektiven Kriterien wie dem Motorgeräusch abhängt. Um den Filter trotzdem auslegen zu können wird sie auf 3,9kHz festgelegt. Diese Frequenz lässt sich im AVR durch den Phase Correct Modus mit einem Prescaler von 8 erreichen.

$$f_{PWM} = \frac{f_{CPU}}{\text{Prescaler} \cdot 2 \cdot 256} = \frac{16\text{MHz}}{8 \cdot 2 \cdot 256} = 3906,25\text{Hz}$$

Abbildung 4.12: Tiefpass Toleranzfeld



Für die Schaltung wird nun ein Sallen Key Tiefpass 2. Ordnung nach Butterworth entworfen. Die PWM-Frequenz  $f_{PWM}$  beträgt 3,9kHz.

Die Sperrfrequenz entspricht der PWM Frequenz, also der Frequenz der Grundschwingung.  $\Omega$  entspricht der mit der Grenzfrequenz normierten Frequenz  $\Omega = \frac{f}{f_g}$ . Nach (4.2) ergibt sich für Abbildung 4.12  $f_s = f_{PWM} = 3,9\text{kHz}$ ,  $a_{min} = 55,9\text{dB}$  und  $a_{max}$  wird auf einen üblichen Wert von 3dB festgelegt.

$$n \geq \frac{\log \sqrt{\frac{e^{2a_{min}} - 1}{e^{2a_{max}} - 1}}}{\log \Omega_s} \quad (4.3)$$

Die Filterordnung nach Butterworth wird nach (4.3) bestimmt [Omab]. Umgestellt nach  $\Omega_s$  ergibt sich:

$$\Omega_s \leq \left( \frac{e^{2a_{min}} - 1}{e^{2a_{max}} - 1} \right)^{\frac{1}{2n}} \quad (4.4)$$

Für die Berechnung der Sperrfrequenz  $\Omega_s$  müssen  $a_{min}$  und  $a_{max}$  in Neper umgerechnet werden. Wobei:

$$1\text{dB} = \frac{\ln 10}{20}\text{Np} = 0,115129255\text{Np}$$

Damit ergibt sich für  $a_{min} = 55,9\text{dB} \cdot \frac{\ln 10}{20} = 6,45\text{Np}$  und für  $a_{max} = 3\text{dB} \cdot \frac{\ln 10}{20} = 0,345\text{Np}$ . Die Filterordnung wird auf 2 festgelegt.

$$\Omega_s \leq \left( \frac{e^{2 \cdot 6,45\text{Np}} - 1}{e^{2 \cdot 0,345\text{Np}} - 1} \right)^{\frac{1}{2}} = 35,8 \quad (4.5)$$

Die Grenzfrequenz  $f_g$  ergibt sich jetzt aus:

$$\frac{f_g}{\Omega_s} \leq \frac{3,9\text{kHz}}{35,8} = 108,9\text{Hz} \quad (4.6)$$

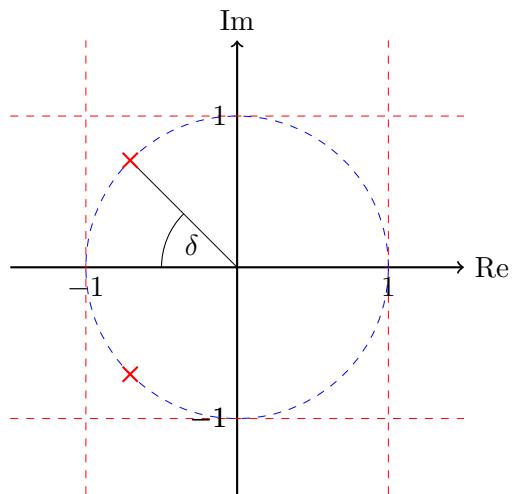
### Filterentwurf

Im vorherigen Abschnitt wurde berechnet, dass die Grenzfrequenz des Filters kleiner als 108,9Hz sein muss. Im Folgenden wird nun ein Sallen-Key Filter 2. Ordnung mit einer Grenzfrequenz von 100Hz entworfen. Die genaue Wahl der Grenzfrequenz ist hier nicht relevant da die realen Bauteile nicht in allen Größen verfügbar sind und daher am Schluss variiert werden müssen, wodurch sich die Grenzfrequenz des Filters leicht ändert.

### Finaler Entwurf

Betrachten wir das Polstellen-Nullstellendiagramm eines Butterworth Filters 2. Ordnung, wie in [Abb. 4.13]

Abbildung 4.13:  
Polstellen-Nullstellendiagramm,  
Butterworth 2. Ordnung



Charakteristisch für den Butterworth-Filter ist, dass sich die Polstellen auf einer Kreisbahn befinden. Auf die Grenzfrequenz normiert hat dieser

beim Butterworth-Filter den Radius eins. Bei einem Butterworth 2. Ordnung befinden sie sich genau bei  $\delta = 45^\circ$ . Das Interessante am Polstellen-Nullstellendiagramm ist, dass sich Polfrequenz  $\Omega_P$  und Polgüte  $Q_P$  einfach ablesen lassen. Die Polfrequenz  $\Omega_P$  ist der Betrag der normierten Polstelle, welcher beim Butterworth-Filter immer eins ist. Die Polgüte ist abhängig von  $\delta$  und ergibt sich zu:  $Q_P = \frac{1}{2\cos\delta}$ . Für den Butterworth-Filter ergeben sich also  $Q_P = 0,707$  und  $\Omega_P = 1$

Betrachten wir die Übertragungsfunktion eines Sallen-Key Tiefpasses 2. Ordnung [Kru00, S. 101]:

$$A(P) = \frac{A_0}{1 + \omega_g(R_2C_1 + R_1C_1 + R_1C_2(1 - A_0))P + \omega_g^2R_1R_2C_1C_2P^2}$$

mit

$$A_0 = 1 + \frac{R_6}{R_5}$$

Die Bauteilwerte erhält man durch einen Koeffizientenvergleich mit der entnormierten Übertragungsfunktion ( $P = \frac{s}{\omega}$ ) eines Tiefpasses 2. Ordnung:

$$A(P) = \frac{A_0}{1 + \frac{1}{\omega_g\Omega_P Q_P}s + \frac{1}{\omega_g^2\Omega_P^2}s^2}$$

Die Auflösung des Vergleiches ist mit vielen mathematischen Umformungen verbunden, deswegen wird hier auf eine externe Quelle verwiesen [Kru00, S. 102]. Nach dem Koeffizientenvergleich ergibt sich

$$\begin{aligned} C_1 &< \frac{C_2 \cdot (1 + 4Q_P^2(A_0 - 1))}{4Q_P^2} \\ R_1 &= \frac{1}{2\omega_g\Omega_P Q_P} \cdot \frac{C_2 \pm \sqrt{C_2^2 - 4Q_P^2C_2(C_1 + C_2(1 - A_0))}}{C_2(C_1 - C_2(1 - A_0))} \\ R_2 &= \frac{1}{2\omega_g\Omega_P Q_P} \cdot \frac{C_2 \pm \sqrt{C_2^2 - 4Q_P^2C_2(C_1 + C_2(1 - A_0))}}{C_1C_2} \\ Q_p &= \frac{\sqrt{R_1R_2C_1C_2}}{C_1(R_1 + R_2) + R_1C_2(1 - A_0)} \\ \Omega_p &= \frac{1}{\omega_g\sqrt{R_1R_2C_1C_2}} \end{aligned}$$

Dabei sind immer nur die positiven, reellen Lösungen zu verwenden.

### **Bestimmung der Bauteilwerte**

Um die Übersicht zu wahren wird die Berechnung der Bauteilwerte hier nur auszugsweise aufgeführt. Zur Erinnerung, die gegebenen Werte sind  $Q_P = 0,707$ ,  $\Omega_P = 1$ ,  $A_0 = 48$  und  $\omega_g = 2 \cdot \pi \cdot 100Hz$ .  $A_0$  ist die Gleichspannungsverstärkung, sie beschreibt den gewünschten Verstärkungsfaktor der bereits in einem vorherigen Abschnitt mit

48 bestimmt wurde. Die Berechnungen wurden mit Hilfe eines Python-Scriptes ausgeführt, dabei wurden verschiedene Konfigurationen durchgerechnet. Hauptsächlich wurde dabei darauf geachtet, dass sich der Filter mit den vor Ort vorhandenen SMD-Bauteilen aufbauen lässt.

In den Berechnungen fiel auf, dass bei steigender Größe der Kondensatoren die Größe der Widerstände sinkt. Da Widerstände auch in großen Größen vorhanden waren, wurde für den frei wählbaren  $C_2$  ein kleiner Wert von 82nF gewählt.

$$C_1 < \frac{C_2 \cdot (1 + 4 \cdot 0.707_P^2(48 - 1))}{4 \cdot 0.707^2}$$

$$C_1 < 3.90\mu F$$

$C_1$  soll nur kleiner sein als 3.90 $\mu F$  und wird ebenfalls auf 82nF gesetzt.

$$B_1 = \frac{82nF \pm \sqrt{82nF^2 - 4 \cdot 0.707^2 \cdot 82nF(82nF + 82nF(1 - 48))}}{82nF(82nF - 82nF(1 - 48))}$$

$$R_1 = \frac{1}{2 \cdot 100Hz \cdot 0,707} \cdot B_1$$

$$R_1 = [-3176\Omega, 2579\Omega]$$

$$B_2 = \frac{82nF \pm \sqrt{82nF^2 - 4 \cdot 0.707^2 \cdot 82nF(82nF + 82nF(1 - 48))}}{82nF^2}$$

$$R_2 = \frac{1}{2 \cdot 100Hz \cdot 0,707} \cdot B_2$$

$$R_2 = [146079\Omega, -118626\Omega]$$

Da nur positive Werte genutzt werden, ergeben sich die Bauteilwerte nun zu:

$$C_1 = 82nF$$

$$C_2 = 82nF$$

$$R_1 = 2579\Omega$$

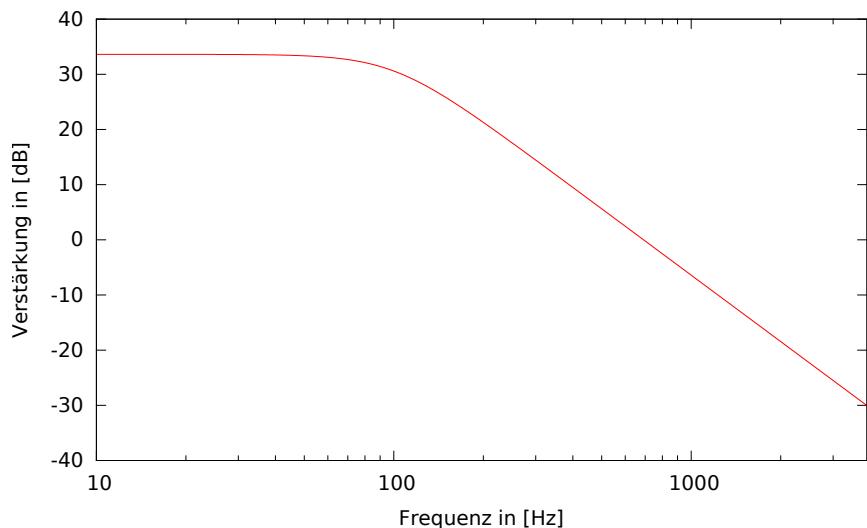
$$R_2 = 146079\Omega$$

In der folgenden Abbildung ist das Ergebnis der Simulation zu sehen. An der Abbildung leider nicht gut zu erkennen, liegt der -3dB Punkt genau bei 100Hz. Die Frequenzachse des Diagrammes geht genau bis 3,9kHz. Es ist eine Verstärkung von 48 des Ursprungssignals gewünscht. Diese Verstärkung wird mit 33,6 dB bei 10Hz, erreicht.

$$20 \cdot \log 48 = -33,6dB$$

Bei 3,9kHz erreicht der Filter eine Dämpfung von -30,1dB. Zusammen mit der Verstärkung von 33,6dB des Eingangssignals wird das bereits verstärkte Signal also um 63,7 dB gedämpft. Gefordert waren hier 55,9dB, sodass der Filter den geforderten Wert übersteigt, was an der niedrigeren Grenzfrequenz von 100Hz statt 108,9Hz liegt.

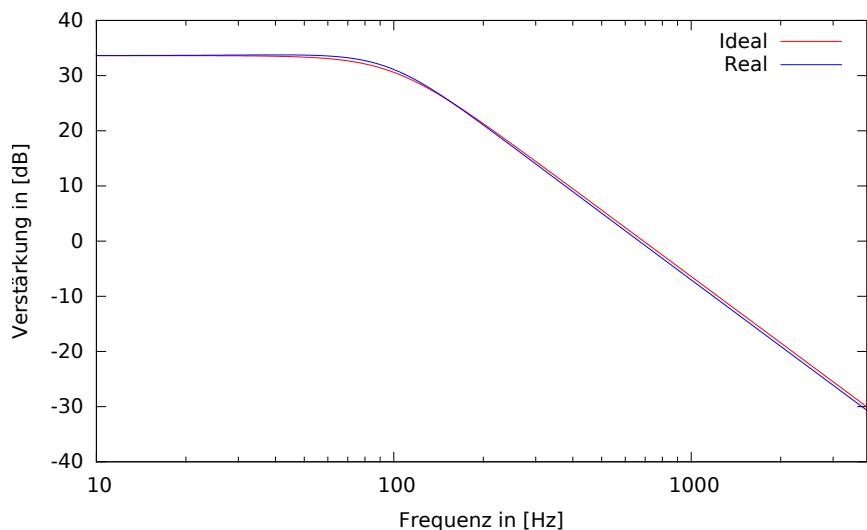
Abbildung 4.14: Frequenzgang des berechneten Filters



Leider kann ein solcher Filter nur mit erheblichen Aufwendungen gebaut werden, da es keine fertigen Widerstände in den Größen  $2579\Omega$  und  $146079\Omega$  gibt. Da jedoch alle Widerstände der E12 Reihe vor Ort vorhanden sind, werden die realen Werte wie folgt gewählt:  $R_1 = 2,7k\Omega$   $R_2 = 150k\Omega$ , da sie den nächsten Größen in der E12 Reihe entsprechen.

In der folgenden Abbildung ist die Simulation des Filters mit den realen Bauteilwerten(blau) im Vergleich zum idealen Filter(rot) zu sehen. Die Grenzfrequenz des Filters (-3dB) liegt diesmal mit 104Hz etwas über den ursprünglichen 100Hz. Da wir die Werte von  $R_5$  und  $R_6$  nicht verändert haben, liegt die Verstärkung bei 10Hz immer noch bei exakt 33,6dB. Bei 3,9 kHz, im Diagramm gut zu erkennen wird trotz der höheren Grenzfrequenz eine höhere Dämpfung als vorher erreicht. Diese liegt bei 33,7dB. Daran kann man erkennen, dass es sich nicht mehr um einen idealen Butterworth-Filter handelt.

Abbildung 4.15: Frequenzgang des berechneten Filters mit finalen Bauteilwerten

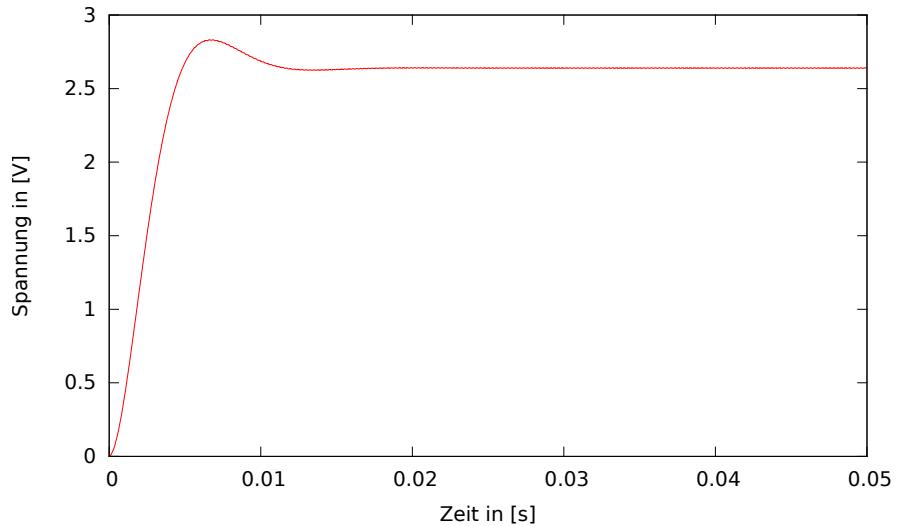


Überall Kommas oder Punkte als Dezimaltrenner

In der folgenden Abbildung [4.16] ist die Antwort des Filters auf ein Rechtecksignal mit 3,9kHz, einem Tastverhältnis von 0,5 und einer Amplitude von 50mV zu sehen. Das Überschwingen im Bereich von 7ms ist

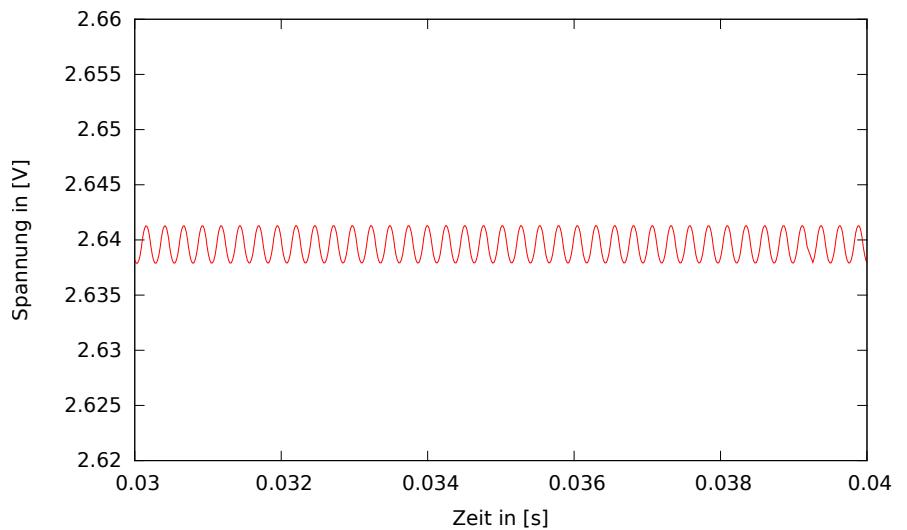
charakteristisch für den Butterworth-Filter und wirkt sich negativ auf die Messung des Stroms aus. Allerdings werden solch große Sprünge in der Praxis nicht auftreten, da der Strom durch die große Induktivität des Motors nur langsam ansteigt.

Abbildung 4.16: Sprungantwort des Filters



Die in Abbildung [4.17] gut zu erkennende Restwelligkeit (Ripple) beträgt 3,36mV und liegt damit deutlich unter den geforderten 4,88mV. Als Eingangssignal dient hier ein Rechtecksignal mit 3,9kHz und einem Tastverhältnis von 0,5, die Amplitude liegt bei 50mV. Die Tatsache, dass das Signal 240mV über den rechnerischen 2,40V ( $0.5V \cdot 48$ ) liegt, röhrt daher, dass LT-Spice die Steig- und Fallzeiten in den low-Bereich des Rechtecksignals legt, wodurch der Mittelwert des Signals bei 2,64V liegt.

Abbildung 4.17: Restwelligkeit des Filters



Da der berechnete Filter den Anforderungen bestens genügt, wird dieser in die Schaltung übernommen.

Zeilenumbruch/Seitenwechsel  
prüfen

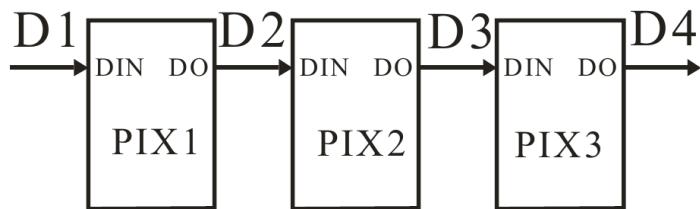
### 4.3 Beleuchtung

Die einfachste Möglichkeit eine Beleuchtung am Auto zu realisieren sind Leuchtdioden, welche in allen erdenklichen Farben zu bekommen sind. Weiter Vorteile sind ihre Energieeffizienz und günstigen Preise. LEDs stellen zudem keine großen Anforderungen an die Energieversorgung.

Das Hauptproblem bei der Integration von vielen LEDs ist die Verkabelung. Eine große Erleichterung bei der Integration sind LED-Streifen. Diese LED-Streifen gibt es in vielen Ausführungen. Für diese Arbeit interessant sind allerdings nur jene Vertreter, welche die Ansteuerung jeder einzelnen LED zulassen. Die LED-Streifen mit Controllern von "Worldsemi" sind hierbei die prominentesten Vertreter. Zu erwähnen wären hierbei die Modelle WS2801, WS2811 und WS2812. Der WS2812 ist dabei nahezu identisch mit dem WS2811 mit dem Unterschied, dass der WS2812 bereits in eine RGB-LED integriert ist. Im Unterschied zum WS2801 werden die beiden durch eine einzige Signalleitung mit fixem Takt angesteuert, sodass hier eine separate Takteleitung entfällt.

Der Nachteil an dieser Methode ist, dass das Timing genau eingehalten werden muss um die Daten korrekt zu übertragen. Wie in Abbildung 4.18 werden die LEDs kaskadiert. Die Daten werden dann durch die LEDs geschoben.

Abbildung 4.18: Kaskadierung der LEDs [Wor]



Um die Beleuchtung am Auto zu realisieren, werden LEDs mit WS2812 genutzt. Da die LEDs an einem beliebigen Eingang des AVR-Mikrocontrollers angeschlossen werden können, wird der LED-Streifen an Pin PA5 angeschlossen. Da das Timing exakt eingehalten werden muss, ist die Software zur Ansteuerung in Assembler geschrieben.

Die LEDs werden wie folgt angesteuert:

Jede LED wird mit einem 24Bit Datenwort angesprochen, welches die Helligkeitsstufen für jede der drei Grundfarben enthält. Die Reihenfolge der Daten ist dabei grün, rot und dann blau. Das höchstwertige Bit wird zuerst übertragen. Die Daten werden ohne Pause gesendet, bis alle LEDs im Strang die nötigen Daten erhalten haben. Nach jeder Übertragung muss eine Pause von mindestens 50µs eingehalten werden, damit die LEDs die Daten übernehmen. Die einzelnen Bits der Übertragung sind dabei folgendermaßen codiert:

Abbildung 4.19: Codierung des LED Signals [Wor]

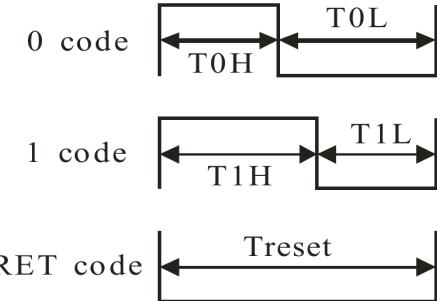


Tabelle 4.1: Signallängen

Die genauen Signallängen können Tabelle 4.1 entnommen werden.

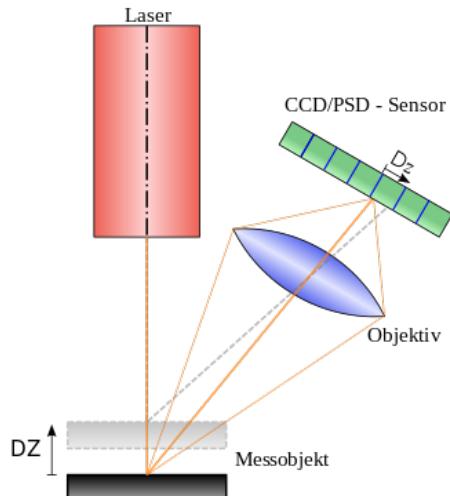
Abschnitt	Beschreibung	Dauer	Abweichung
T0H	0 Code, high Zeit	$0.35\mu s$	$\pm 150\text{ns}$
T1H	1 Code, high Zeit	$0.7\mu s$	$\pm 150\text{ns}$
T0L	0 Code, low Zeit	$0.8\mu s$	$\pm 150\text{ns}$
T1L	0 Code, low Zeit	$0.6\mu s$	$\pm 150\text{ ns}$
RES	Reset Code, low Zeit	über $50\mu s$	

## 4.4 Distanzsensoren

### 4.4.1 Messprinzip

Die ausgewählten Sensoren der Sharp GP2D Reihe basieren auf einer optischen Abstandsmessung, genauer der optischen Abstandsmessung durch Triangulation. Bei der optischen Abstandsmessung durch Triangulation projiziert ein Projektor einen Lichtpunkt auf das Messobjekt [4.20]. Ein optischer Sensor misst dann den Winkel des vom Messobjekt reflektierten Lichts. Durch Triangulation kann dann durch den fest definierten Abstand des optischen Sensors von der Lichtquelle die Entfernung zum Objekt berechnet werden. [Hug07]. In Abb. 4.20 ist dieses Prinzip veranschaulicht.

Abbildung 4.20: Prinzip der Lasertriangulation [Wikb]



Vorteile des Messprinzips: Da es sich um eine rein trigonometrische Messung handelt, kann sie zur kontinuierlichen Messung von beweglichen

Objekten verwendet werden. Außerdem besitzen Sensoren nach diesem Prinzip einen kleinen Messfleck.

Nachteile: Die Messung ist stark von der Oberfläche des Messobjektes abhängig, spiegelnde Oberflächen stellen ein großes Problem dar. In staubigen oder nebligen Umgebungen wird das Licht möglicherweise zu stark gestreut, sodass eine korrekte Messung nicht möglich ist.

#### 4.4.2 Probleme der GP2D Sensoren

Die Sensoren verfügen über einen analogen Ausgang. Bei analogen Signalen ist generell mit Störungen zu rechnen. Die GP2D Sensoren scheinen hohe Anforderungen an die Energieversorgung zu stellen. Hier ist eine Entstörung mittels Kondensator vorausgesetzt, da im Messsignal sonst große Spikes entstehen, wie in Abbildung 4.21 zu sehen.

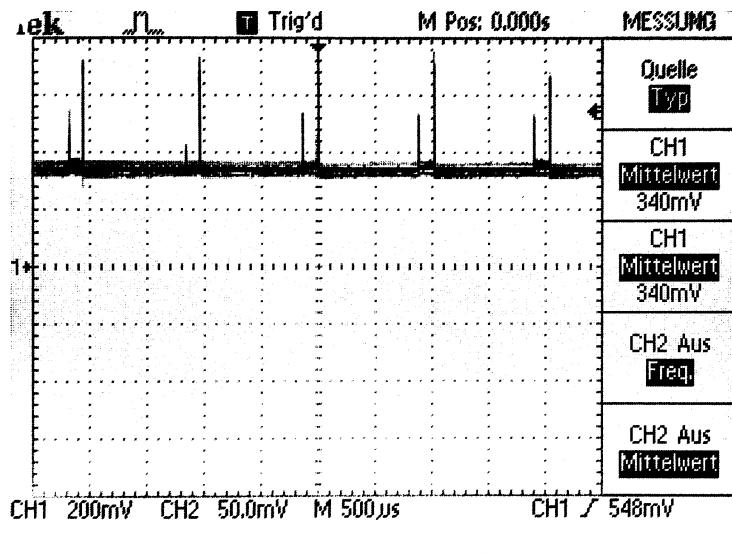


Abbildung 4.21: Ausgangssignal GP2D120

Nach der Entstörung mit einem 82 nF Kondensator direkt am Sensor zwischen VCC und GND sind die Störungen bereits stark vermindernt.

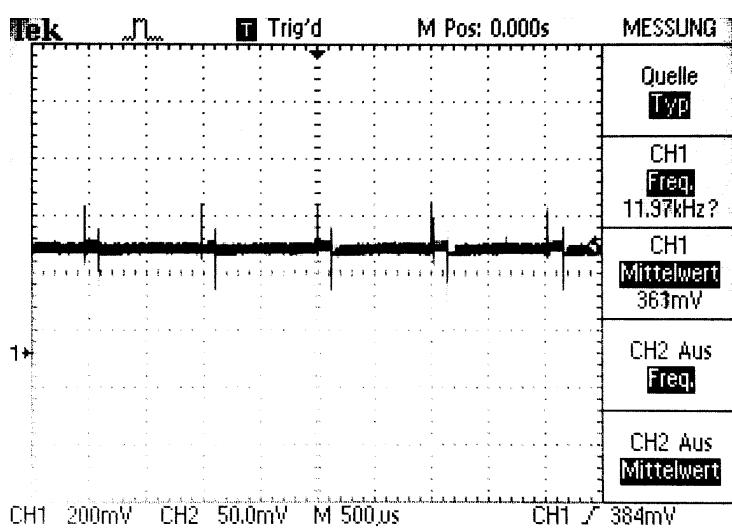


Abbildung 4.22: Ausgangssignal GP2D120 entstört

#### 4.4.3 Auswertung des Messsignals

Das Messsignal vom Sensor wird über einen ADC-Eingang des AVR Mikrocontrollers ausgelesen. Angeschlossen werden die Sensoren über die PH-Connector Reihe des Herstellers JST, welche auch an den Sensoren verbaut sind.

---

## 4.5 Odometrie

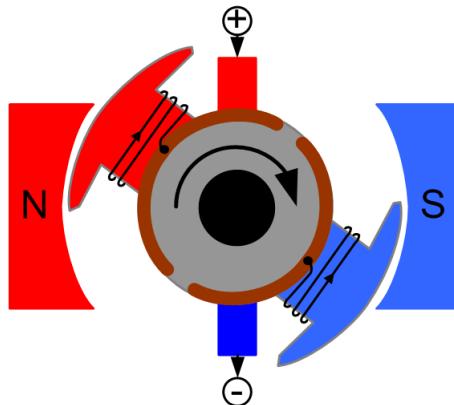
Das Auto benötigt in unterschiedlichen Situationen unterschiedlich viel Leistung, um seine Geschwindigkeit zu halten. Besonders in Kurven ist durch die erhöhte Reibung mehr Motorleistung nötig. Über den Motor treiber lässt sich jedoch nur die mittlere Spannung am Motor verändern, deshalb ist es nötig diese zu regeln. Dafür ist jedoch eine Rückführung der Geschwindigkeit des Autos nötig. Eine Aufintegrierung der Beschleunigungsdaten der Inertialsensorik führt auf Dauer zu erheblichen Abweichungen und ist daher für eine Regelung nicht geeignet. Auch eine Odometrie an den Rädern des Autos ist aus mechanischen Gründen schwer zu realisieren.

Durch die feste Übersetzung des Getriebes bietet sich die Messung der Motordrehzahl an. Damit lässt sich eine gute Näherung für die aktuelle Geschwindigkeit erreichen.

### 4.5.1 Hall-Sensor

Eine Möglichkeit die Motordrehzahl zu messen ist es das Magnetfeld des Motorankers auszuwerten. Dazu sehen wir uns den Aufbau eines Gleichstrommotors an.

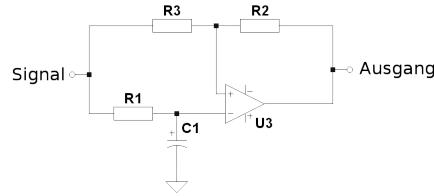
Abbildung 4.23: Aufbau eines Gleichstrommotors [Wika]



Ein Gleichstrommotor besteht aus zwei grundsätzlichen Teilen, einem unbeweglichen Teil, den Stator und einem beweglichen Teil, dem Anker. Der Stator besteht aus sich gegenüberliegenden Permanentmagneten, welche zwei entgegengesetzt gepolte Magnetfelder erzeugen. Der Anker besteht aus Elektromagneten dessen Polung jede halbe Umdrehung kommutiert wird. Durch die Kommutierung ändert sich die Polung der Elektromagneten. Das sich so ändernde Magnetfeld kann mit einem Hall-Sensor ausgewertet werden. Das entstehende Signal ähnelt dabei über der Zeit einer Sinusschwingung. Mit Hilfe eines Schmitt-Trigger kann daraus ein Drehzahlsignal generiert werden.

Abbildung 4.24: Schmitt-Trigger Schaltung

Der Hallsensor ist erfolgreich durch ein Belüftungsloch im Motor platziert wurden. Als Referenzspannung für den Schmitt-Trigger wird der Mittelwert des Sensorsignals, erzeugt durch einen Tiefpass, genutzt.



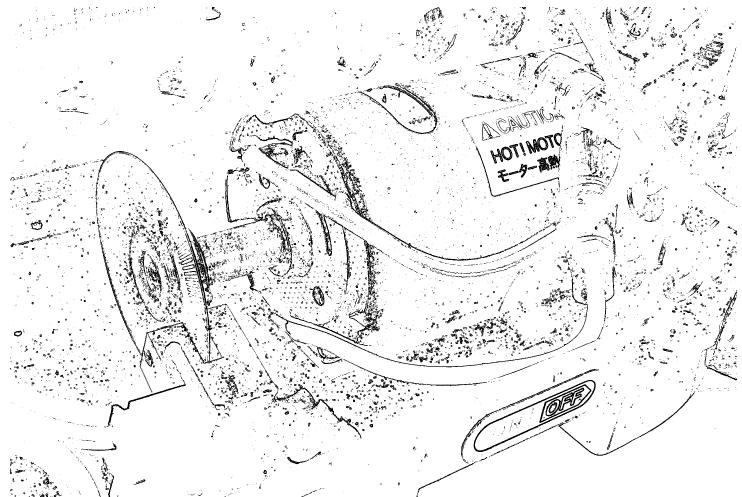
Leider führt dieses Vorgehen nicht zum Erfolg, da die Magnetfeldstärke stark von der Drehrichtung des Motors abhängt. Es war nicht möglich den Schmitt-Trigger so auszulegen, dass er in beide Drehrichtungen zuverlässig funktioniert.

Im Schaltplan befindet sich eine andere Schaltung, diese wurde aus Zeitnot kurz vor der Bestellung der Platine eingearbeitet und funktioniert ebenfalls nicht!

#### 4.5.2 Gabellichtschanke

Alternativ zum Vorgehen mit einem Hall-Sensor wurde eine andere Lösung implementiert. An eine speziell für diesen Motor angefertigte Achsverlängerung wurde eine Inkrementalgeber scheibe befestigt, zu sehen in Abb. 4.25. Diese wird durch einen Sharp GP1A30R Sensor ausgewertet, welcher ein Drehzahlsignal an den externen Takteingang des Timer 1 vom AVR Mikrocontroller liefert.

Abbildung 4.25: Motor mit Inkrementalgeber



Die Scheibe ist dabei mit 120 Strichmarkierungen versehen. Eine Motordrehung entspricht also 120 Impulsen, im folgenden Motorticks genannt. Über das Übersetzungsverhältnis des Getriebes und dem Radumfang kann dann der zurückgelegte Weg berechnet werden. Das Übersetzungsverhältnis des Fahrzeugs ist abhängig von drei Komponenten, dem Stirnrad, dem Motorritzel sowie einer festen internen Übersetzung. Das verwendete Stirnrad hat 61 Zähne, das Motorritzel 19. Damit ergibt sich ein Verhältnis von 61:19 zusammen mit der internen Übersetzung von 2.6 haben wir ein Übersetzungsverhältnis von 8,35. Der Radumfang beträgt ca. 21 cm.

Quelle raussuchen für interne Übersetzung

$$\frac{120 * 8,35}{0,21m} = \frac{ticks}{m} = 4760 \quad (4.7)$$

Mit dieser Größe lassen sich die Motorticks einfach in Meter umrechnen.

## 4.6 Auslegung der Stromversorgung

Um das Layout der Platine möglichst simpel zu halten und damit kostengünstig zu bleiben, wurden alle Komponenten so ausgewählt, dass diese über eine einzige 5V Spannungsquelle mit Energie versorgt werden können. Es ist wichtig den Stromverbrauch aller Komponenten abzuschätzen, um die Spannungsversorgung sinnvoll dimensionieren zu können. Eine zu schwache Spannungsquelle kann zu Instabilitäten führen, während eine überdimensionierte Geld verschenkt.

### 4.6.1 Stromverbrauch der Komponenten

In diesem Abschnitt soll eine Abschätzung des Stromverbrauchs vorgenommen werden. Dabei wird keinen Wert auf hohe Genauigkeit gelegt, es soll nur eine ungefähre Größenordnung für den Stromverbrauch gefunden werden.

#### Servomotor

Der Stromverbrauch des Servomotors ist schwer zu ermitteln. Da es sich um einen Modellbauservomotor handelt sind im Datenblatt hierzu keine Daten aufzufinden. Da ein Messaufbau für die Abschätzung des Stromverbrauches zu aufwändig ist, werden hier Messwerte eines ähnlichen Servos aus einem Artikel [Mor] herangezogen. Laut diesem hat eine Servomotor keinen konstanten Stromverbrauch. Der Stromfluss wird immer wieder unterbrochen, sodass es zu einem intervallartigen Stromfluss kommt. Nur wenn der Servomotor dauerhaft belastet wird, kommt es zu einem konstanten Stromfluss. Im Artikel werden mehrere Servomotoren vermessen, der Motor der dem verwendet am nächsten kommt ist der "Graupner 4421" mit folgenden Daten:

**Servo Modell [rob] in Anforderungen mit aufnehmen**

Technische Daten "Graupner 4421" [Gra]:

- Stellzeit(60°): 0,11s
- Stellmoment: 88Ncm

Technische Daten des verwendeten Servomotors [rob]:

- Stellzeit(60°): 0,13s (4,8V) / 0,16s (6,0V)
- Stellmoment: 92Ncm (4,8V) / 78Ncm (6,0V)

Dieser hat laut des Artikels eine maximale Stromaufnahme von 1,2A. Um noch Luft nach oben zu haben wird hier ein Verbrauch von 1,8A angenommen.

## **Pandaboard ES**

Leider gibt es vom Hersteller des Pandaboard keine konkreten Angaben zum Stromverbrauch. Der Hersteller empfiehlt jedoch ein Netzteil mit 4A [Omaa], wobei auch ein Betrieb an USB mit Hilfe eines Y-Kabels möglich ist. Die USB-2.0 Spezifikation [USB] sieht eine maximale Stromabgabe von 500mA vor.

Der Stromverbrauch des normalen Pandaboard (ohne ES) beträgt ca. 800mA [Bar]. Nähere Angaben zum Stromverbrauch des normalen Pandaboard (ohne ES) finden sich in [Bar]. Der Verbrauch des Pandaboard ES dürfte durch den schnelleren Prozessor minimal darüber liegen. Durch Anschluss von USB-Geräten an das Board kann der Stromverbrauch jedoch noch steigen. Die USB-Spezifikation [USB] sieht pro Port eine maximale Stromentnahme von 500mA vor. Da das Pandaboard über 2 USB-Ports verfügt liegt der maximale zusätzliche Verbrauch bei 1A, sodass hier ein Gesamtverbrauch von 2A veranschlagt wird.

**Hinweis:** Das Pandaboard ES wurde im Laufe des Projekts durch einen Intel NUC ersetzt, welcher jedoch nicht über die 5V Schiene versorgt wird. Daher entfällt der Verbrauch des Boards in den Messungen der Evaluierung.

## **LED Beleuchtung**

Auch wenn LEDs den Ruf haben besonders energieeffizient zu sein, ist der Stromverbrauch bei einer größeren Anzahl nicht zu unterschätzen. Da wir RGB-LEDs einsetzen besteht ein LED-Modul aus 3 LEDs in den Grundfarben rot, blau und grün. Laut Datenblatt [Wor] haben die LEDs eine Stromaufnahme von 20mA, also 60mA pro Modul. Um regelwerkkonform zu sein, werden folgende Beleuchtungen benötigt: Blinker rechts und links jeweils vorne und hinten. Sowie eine Leuchte, welche den RC-Modus anzeigt. Zusätzlich zu den im Regelwerk vorgeschriebenen Beleuchtungen werden noch je zwei Frontscheinwerfer und drei Rück-/Bremslichter integriert, so dass sich eine Anzahl von 10 LED-Modulen ergibt. Der maximale durch die LEDs verursachte Strom liegt somit bei 600mA.

## **Mikrocontroller**

Der maximale Stromverbrauch des AVR Mikrocontrollers liegt laut Datenblatt [Atm] bei 200mA, wenn IO-Pins belastet werden. Der Mikrocontroller selber braucht jedoch bei 5V Betriebsspannung und 16MHz nur 29mA. Da die IO-Pins des Controllers nur wenig belastet werden, wird hier nur ein Verbrauch von 100mA veranschlagt.

## **Sharp Sensoren**

Die Sharp GP2D120 Distanzsensoren verbrauchen laut Datenblatt [Sha] 50mA, da zwei dieser Sensoren verbaut werden ergeben sich 100mA.

## **Sonstige Peripherie**

Da der Stromverbrauch der restlichen Komponenten minimal ist, werden hier pauschal 100mA veranschlagt.

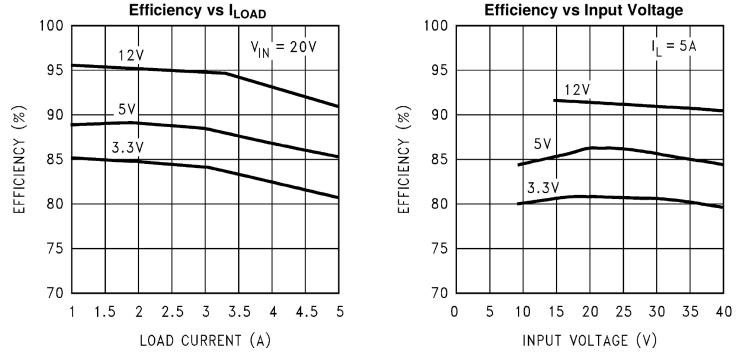
#### 4.6.2 Auswahl des Reglers

Der Gesamtstromverbrauch der Komponenten beträgt 4800mA. Ein Linearregler ist hier nicht mehr praktikabel, da dieser bei einer Akkusspannung von 14,4V und 4800mA über 45 Watt Leistung in Wärme umwandeln würde.

$$P_{\text{linear}} = (14.4 \text{ V} - 5 \text{ V}) \cdot 4.8 \text{ A} = 45.120 \text{ W}$$

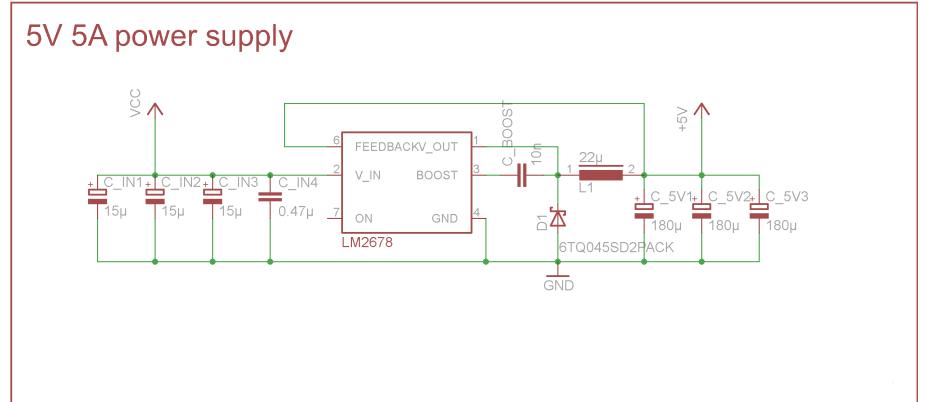
Schaltregler in Form von Abwärtswandlern haben hingegen einen sehr hohen Wirkungsgrad. Eine gute Wahl ist die Schaltregler-Reihe von Texas Instruments, da diese einfach zu bekommen sind, und nur wenig Außenbeschaltung benötigen. Ein Exemplar das unseren Anforderungen entspricht ist der LM2678 von Texas Instruments, dieser kann dauerhaft einen Strom von 5A liefern und sein Wirkungsgrad liegt selbst bei Maximallast bei über 80%. Eine Übersicht dazu findet sich in Abbildung [4.26].

Abbildung 4.26: Regulator  
Wirkungsgrad [Ins]



Ausgehend von ca. 24 Watt Leistungsaufnahme ( $4.8A * 5V$ ) und einem minimalen Wirkungsgrad von 80% ergibt sich damit eine überschauliche Verlustleistung von 6 Watt.

Abbildung 4.27: Schaltplan des  
5V Reglers

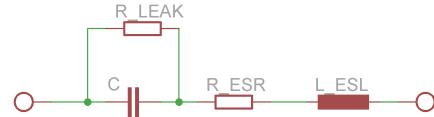


Die Beschaltung erfolgt dabei nach den Empfehlungen des Datenblattes. Die verwendeten  $180 \mu\text{F}$  Kondensatoren sollen laut Datenblatt Low-ESR Kondensatoren sein. Die verbauten Kondensatoren stammen aus Nichicon's L8 Serie und haben einen Serienwiderstand von nur  $12 \text{ m}\Omega$ .

*Abbildung 4.28: Ersatzschaltbild eines Kondensators*

### Low-ESR Kondensatoren:

Low-ESR Kondensatoren zeichnen sich durch einen niedrigen Serienwiderstand ( $R_{ESR}$ ) aus. Dieser liegt in Reihe(Serie) zum idealen Kondensator (Abb. 4.28).



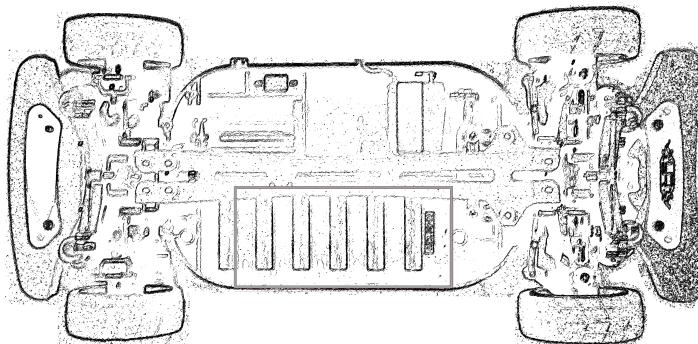
Dieser verursacht Verluste innerhalb des Kondensators, was bei Belastung zur Erwärmung des Kondensators führt und seine Lebensdauer verringert. Je kleiner der ESR desto niedriger die Verluste im Kondensator. Weiterhin kann ein Kondensator mit kleinem ESR schneller ge- und entladen werden, als ein herkömmlicher Kondensator. Durch diese Eigenschaften ist ein ESR Kondensator hervorragend zur Störunterdrückung geeignet.

---

## 4.7 Das Layout

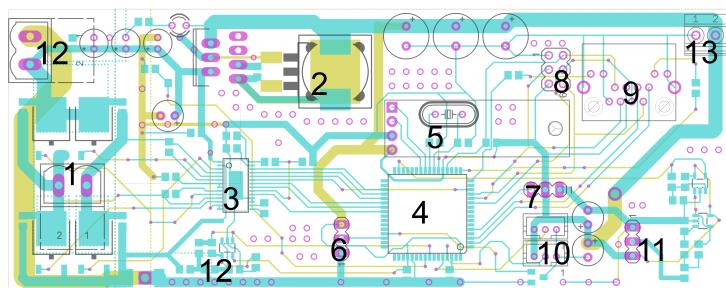
In Abbildung 4.30 ist das fertige Platinenlayout zu sehen. Bei der Entwicklung des Layouts wurde darauf geachtet, dass die Platine in die vorgesehene Lücke des Fahrzeugs passt.

*Abbildung 4.29: Fahrzeug Platinen-Position*



*Abbildung 4.30: Platinenlayout*

Die Lücke hat Maße von 13.0 cm mal 5.4 cm. Das Platinenlayout hat eine Größe von 13.0 cm mal 5.05 cm und passt somit in die Lücke ohne dabei zu viel Spielraum zu haben, was in einem festen Sitz der Platine resultiert.



Auf dem Layout sind folgende Komponenten zu sehen:

- 1 H-Brücke mit Motoranschluss

- 2 5 A Schaltregler
- 3 Allegro Mosfettetreiber
- 4 Atmel Mikrocontroller
- 5 Sparkfun Inertialsensor ( Anschluss und Position )
- 6 Anschluss für den LED-Strang
- 7 Anschluss für den Servomotor
- 8 ISP Anschluss zum Programmieren des Mikrocontrollers
- 9 RJ45 Buchse zum Anschluss der Pandaboard (mit SPI und UART)
- 10 Anschlüsse für die Sharp GP2D Sensoren
- 11 Anschluss und Filterschaltung für die Odometrie (Hallsensor)
- 12 Shunt mit Filterschaltung zur Strommessung
- 13 Stromanschluss für die Pandaboard

## 4.8 Software

Die Software besteht im Grunde aus zwei Teilen, zum einem der Firmware auf dem Mikrocontroller zum anderem aus der Software auf der Recheneinheit, welche die Daten vom Mikrocontroller ausliest und über ROS published. In den folgenden Abschnitten werden erst die beiden Softwareteile erläutert, dann wird das Übertragungsprotokoll veranschaulicht.

### 4.8.1 Software auf dem Mikrocontroller

Die Software auf dem Mikrocontroller ist vollständig in C++ geschrieben. Eine vollständige Dokumentation der Software ist als Doxygen Dokument verfügbar. Die Software fungiert auf dem Controller als Service und wartet legendlich auf eine Anfrage von der seriellen Schnittstelle, welche sie bearbeitet und bei Bedarf beantwortet.

### 4.8.2 Client Programm auf der Recheneinheit

Das Client Programm, im folgenden SerialNode genannt, wurde zuerst in Python implementiert. Durch die Verwendung der pyserial Bibliothek zum Ansprechen der seriellen Schnittstelle wurde jedoch eine enorm hohe CPU-Last verursacht. Da sich das Problem kurzfristig nicht lösen ließ, die Rechenleistung für andere Aufgaben benötigt wird und auch Energieeffizienz ein wichtiges Kriterium ist, wurde das Programm erneut in C++ implementiert. Unter Verwendung der Systemaufrufe von Poll konnte das Abfragen der seriellen Schnittstelle auf Systemebene ausgeführt werden, was die Effizienz stark verbesserte. Während die Python Implementierung einen CPU-Kern zu 100% auslastete liegt die C++ Implementierung im unteren einstelligen Prozentbereich. Eine vollständige Dokumentation der Software ist ebenfalls als Doxygen Dokument verfügbar.

Das Programm stellt nach seinem Start folgende ROS-Topics zur Verfügung:

Ros-Topic	Ros-Datentyp
/sensors/current	std_msgs/Float32
Der aktuelle Motorstrom in Ampere	
/sensors imu / data raw	ottocar_msgs/simpleImu
Die Daten werden im ROS-Standartformat für Inertialsensoren zur Verfügung gestellt	
/sensors/IR1	std_msgs/Float32
Der aktuelle Spannungswert des Sensors in Volt	
/sensors/IR2	std_msgs/Float32
Der aktuelle Spannungswert des Sensors in Volt	
/sensors/motor_revolutions	std_msgs/UInt32
Die Anzahl der vergangenen Motorumdrehungen seit dem Start des Mikrocontrollers	
/sensors/motor_state	std_msgs/UInt8
Der aktuelle Zustand des Motortreibers	
/uC_time	std_msgs/UInt32
Die vergangene Zeit auf dem Mikrocontrollers seit dessen Start in Millisekunden	
/sensors/voltage	std_msgs/Float32
Die aktuelle Akkuspannung in Volt	

Tabelle 4.2: ROS-Publisher

Das Programm hört auf folgende ROS-Topics:

Tabelle 4.3: Ros-Subscriber

Ros-Topic	Ros-Datentyp
/actuators/speed_cmd	std_msgs/Int8
Die neue Motorgeschwindigkeit von -128 bis +127	
/actuators/angle_cmd	std_msgs/Int8
Der neue Servowinkel von -128 bis +127	
/actuators/motor_reset	std_msgs/Bool
Initiiere einen Reset im Motortreiber	

#### 4.8.3 Übertragungsprotokoll

Da die Übertragung der Daten via ROS-Serial im ersten Prototypen zu vielen Problemen geführt hat, wurde ein neues Übertragungsprotokoll entwickelt. Dabei wurde auf Fehlertoleranz und niedrigen Ressourcenverbrauch geachtet. Der Datendurchsatz muss hier ausreichend sein, um alle Daten stabil mit 100Hz übertragen zu können. Der grundlegende Ablauf der Datenübertragung ist in Abb. 4.31 und 4.32 zu sehen.

Während eines Lesevorganges (Abb. 4.31) wartet der Mikrocontroller auf ein fest definiertes Startsignal von der Recheneinheit. Nachdem das Startsignal empfangen wurde, wird auf eine weitere Preamble gewartet. Dies ist notwendig, um bei Asynchronitäten die Wahrscheinlichkeit eines zufälligen Startsignals zu verringern. Wurde die Preamble erfolgreich empfangen, erwartet der Mikrocontroller eine gültige Topic ID. Wurde eine falsche Preamble empfangen, wartet der Mikrocontroller erneut auf ein gültiges Startsignal. Abhängig von der Topic ID verfährt

der Mikrocontroller dann im Programm fort und beantwortet die Anfrage entsprechend. Für manche Topics ist eine Bestätigung (Acknowledge) nötig. Bekommt der Mikrocontroller keine Bestätigung sendet er die Daten erneut. Eine Bestätigung wird nur gesendet, wenn die vom Client berechnete Checksumme mit der mitgesendeten identisch ist. Dies ist besonders bei den Daten der Inertialsensorik von Nöten, da hier defekte Daten zu einem dauerhaften Fehler führen. Die Anzahl dieser Retransmits ist begrenzt und wird vom Client festgelegt. Wurde ein Acknowledge empfangen oder für den Topic ist keine Bestätigung nötig, wartet der Controller erneut auf ein Startsignal.

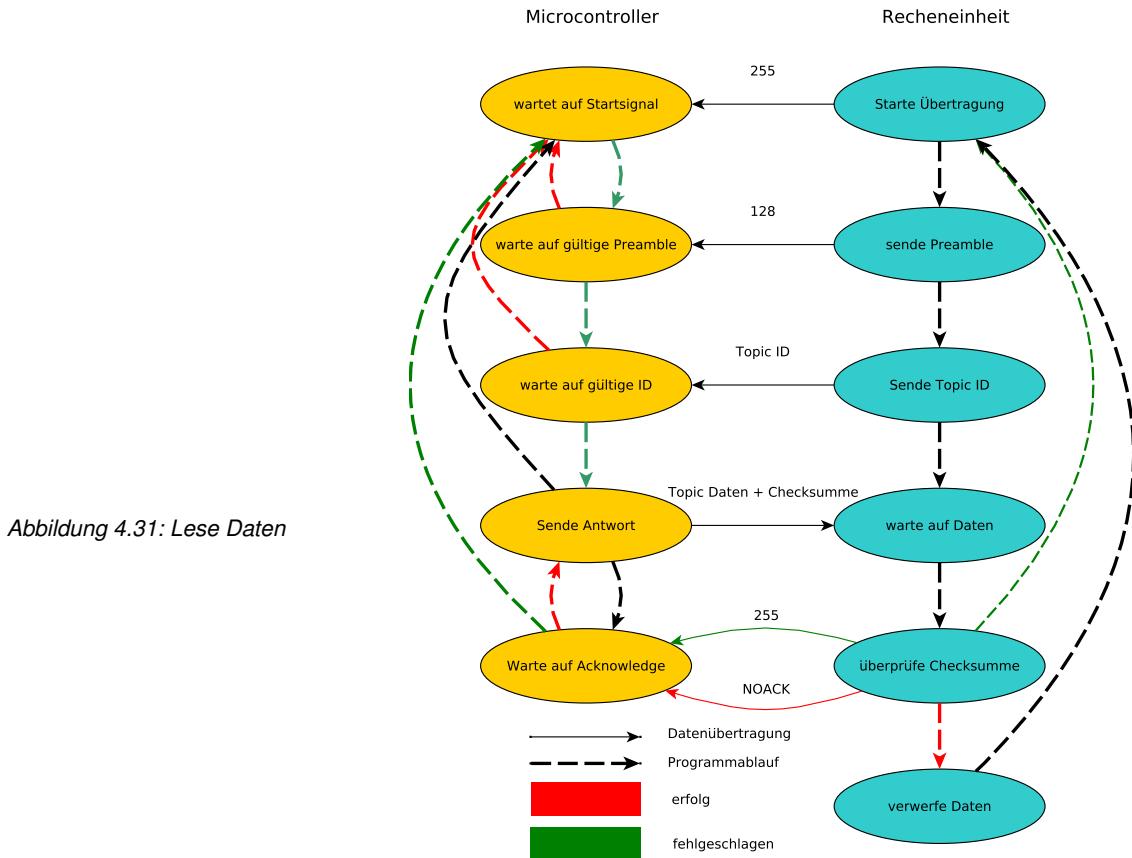
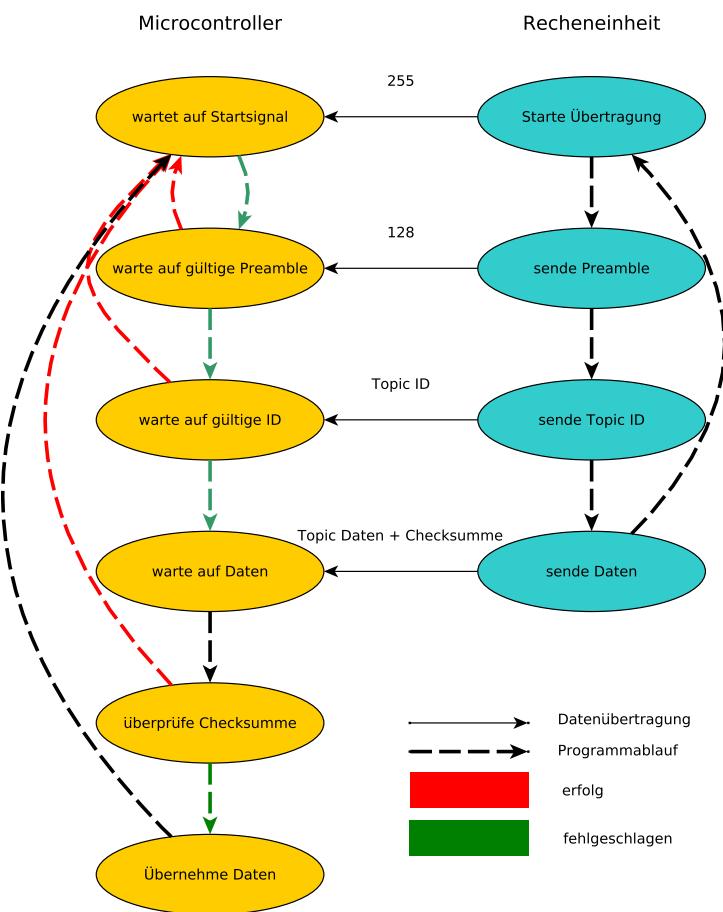


Abbildung 4.31: Lese Daten

Der Ablauf eines Schreibvorganges (Abb. 4.32) auf dem Mikrocontroller ist ähnlich. Der Mikrocontroller wartet auf das Startsignal und eine Preamble sowie eine Topic ID. Nach dem Empfang der Topic ID erwartet er die nötigen Daten, samt Checksumme. Diese ist nötig, um die Daten zu verifizieren. Bei der Ansteuerung des Motors kann ein Bitfehler zu fatalen Folgen führen. Beispielsweise kann die Recheneinheit oder der Mikrocontroller abstürzen, wenn die Motorleistung plötzlich stark erhöht wird. Daher werden die Daten nur übernommen, wenn die Checksumme erfolgreich verifiziert wurde. Sollte die Checksumme nicht verifiziert werden, springt der Mikrocontroller erneut in den Startzustand und wartet auf ein Startsignal, ohne die Daten zu übernehmen. Der Client bekommt davon nichts mit, wenige einzelne Fehler können durch die hohe interne Datenrate von 100 Hz ignoriert werden. Häufige oder dauerhafte Fehlübertragungen können durch die Reaktion des Autos leicht identifiziert werden. Häufige oder dauerhafte Fehlübertragungen würden durch das fortlaufende Lesen und Schreiben auf dem Mikrocontroller so oder so zu Fehlern in beiden Richtungen führen.

Abbildung 4.32: Schreibe Daten



# 5

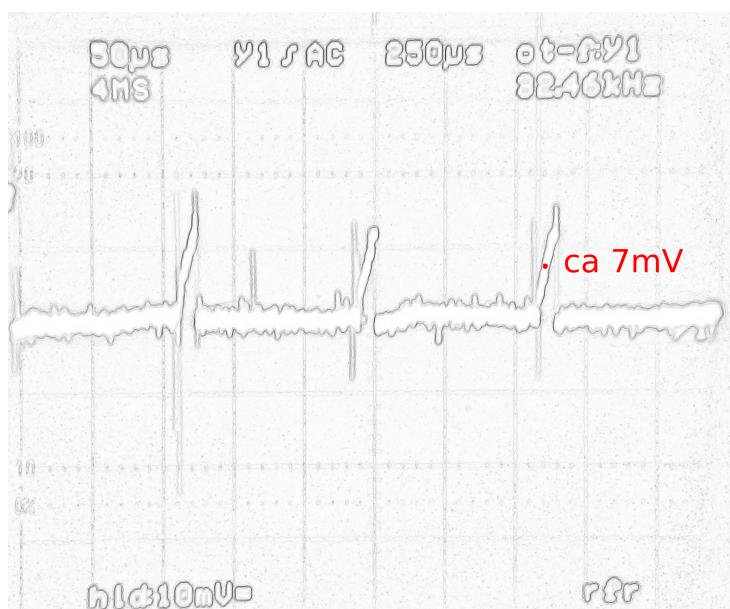
## Evaluierung

### 5.1 Strommessung

Zur Qualitätsbewertung der Strommessung wurde das Fahrzeug aufgebockt. Das Fahrzeug befand sich während aller Messungen im Akkubetrieb, bei ca. 16,8V Akkuspannung. Der Motor wurde, soweit nicht anders beschrieben, mit einem PWM-Tastgrad von 30:256 angesteuert. Zur Beurteilung der Messergebnisse wurden Messungen an diversen Punkten vorgenommen und mit der Ausgabe des Mikrocontrollers verglichen.

Zuerst wurde dazu die Spannung direkt am Shuntwiderstand gemessen. Dabei kann der periodische Anstieg der Spannung am Shunt beobachtet werden, zu sehen in Abb. 5.1. Die Spannung am Shunt ist proportional zum Strom durch den Motor.

Abbildung 5.1: Spannung am Shunt



Die mittlere Spannung aus Abb. 5.1 lässt sich abschätzen, indem die Spannung in der Mitte eines Impulses abgelesen und mit dem Tastverhältnis multipliziert wird.  $7\text{ mV} \cdot \frac{30}{256} = 0.82\text{ mV}$  Mit Hilfe dieses Mittelwertes können wir grob abschätzen, ob der nachfolgende Filter seine

Funktion erfolgreich erfüllt.

Als Nächstes wird die Spannung nach der Filterschaltung direkt am Operationsverstärker gemessen. Es fällt auf, dass das Signal trotz Tiefpass noch Störungen in Form des Motorstromes enthält. Dies ist eine Auswirkung der gestörten Betriebsspannung, welche später untersucht wird.

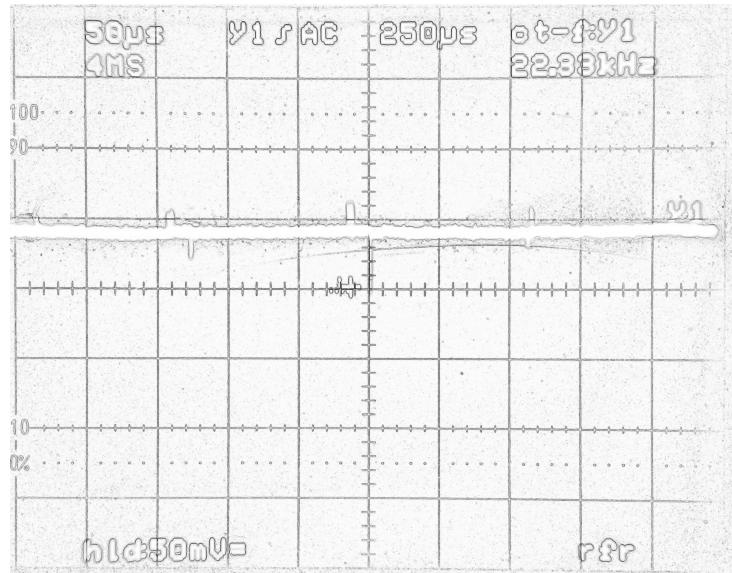


Abbildung 5.2: Spannung nach dem Filter

Wie in Abb. 5.2 zu sehen wurde die Eingangsspannung erfolgreich verstärkt und hat nun etwa eine Spannung von 40 mV. Dividiert durch den Verstärkungsfaktor 48 ergibt sich eine Spannung von 0.83 mV. Welche sehr nahe an dem vorher abgeschätzten Wert von 0.82 mV liegt. Der Strom lässt sich nun mit Hilfe des Ohmschen Gesetzes  $U = R \cdot I$  errechnen.

$$I = \frac{0.83 \text{ mV}}{0.005 \Omega} = 166.7 \text{ mA}$$

Der vom Mikrocontroller ausgegebene Strom beträgt im Mittel 156.1 mA.

Zur weiteren Bewertung der Messung wurden jeweils ca. 1000 Samples an Daten zu unterschiedlichen PWM-Tastgraden aufgezeichnet. Dabei soll untersucht werden, ob die Qualität der Messwerte mit zunehmender Geschwindigkeit stabil bleibt. Die Ergebnisse der Messungen sind in Tabelle 5.1 zu sehen.

Tabelle 5.1: Motorstrom im Leerlauf

Tastgrad	Erwartungswert [A]	Standardabweichung [A]
10	0.0044	0.0212
20	0.0730	0.0214
30	0.1600	0.0215
40	0.2629	0.0214
50	0.3653	0.0185
60	0.4720	0.0231

Die Standardabweichung der Messwerte scheint stabil zu bleiben. Da der Strom sich in diesen Messungen in einem sehr niedrigen Bereich befindet, wurden die Messungen erneut unter Last durchgeführt, der Motor befand sich währenddessen im Kurzschlussbetrieb.

Tabelle 5.2: Motorstrom unter Last

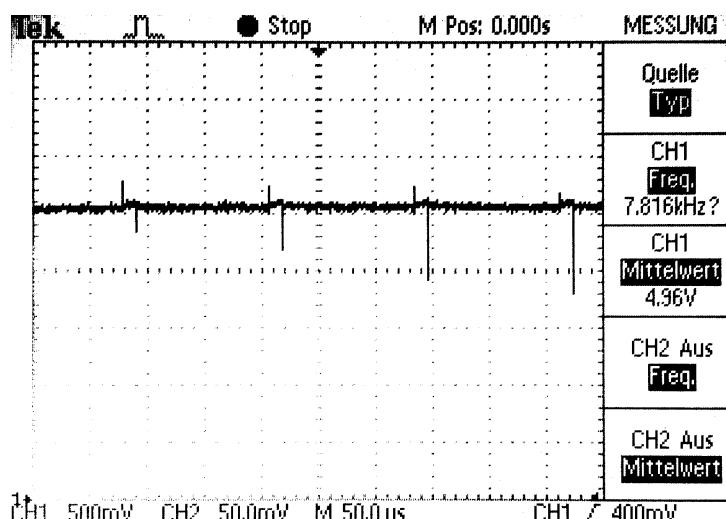
Tastgrad	Erwartungswert [A]	Standardabweichung [A]
10	0.0454	0.0183
20	0.4635	0.0243
30	1.1694	0.0338
40	1.9822	0.0563
50	3.8737	0.0477
60	4.7023	0.0706

Der Tabelle 5.2 kann leicht entnommen werden, dass die Ströme unter Last stark ansteigen. Die Standardabweichung steigt im Verhältnis zum Strom nur leicht. Sodass die Streuung der Messung mit steigendem Strom zwar zunimmt, die relative Abweichung der Werte jedoch nicht zunimmt.

## 5.2 Spannungsversorgung

Während der Untersuchung der Strommessung sind Störungen im Ausgabesignal der Filterschaltung festgestellt worden. Um die Ursache der Störung zu finden, wurde die Spannungsversorgung des Fahrzeugs untersucht.

Abbildung 5.3: Störungen im 5V Netz



Betrachtet man den Verlauf der Spannung im 5V-Netz lässt sich leicht erkennen, dass das Signal von einem dem Strom sehr ähnlichen Signal überlagert wird. Eine solche Störung kann nur von der Betriebsspannung kommen, da der Motor nicht an das 5V-Netz angeschlossen ist. Scheinbar ist der 5V-Schaltregler nicht in der Lage diese Störungen vollständig auszugleichen. Daher wird nun die Betriebsspannung untersucht, um das Ausmaß der Störungen beurteilen zu können. Die Werte wurden in einer 1:10 Teilung gemessen, daher sind alle Spannungen mit dem Faktor 10 zu multiplizieren.

Im Akkubetrieb (Abb. 5.4) kann gut beobachtet werden, dass die Spannung um mehr als 100 mV schwankt. Die Frequenz der Spannungseinbrüche entspricht in etwa der PWM Frequenz von 7812.5 Hz. Der Motor wurde währenddessen mit einem Tastgrad von 30:256 angesteuert und wurde nicht belastet.

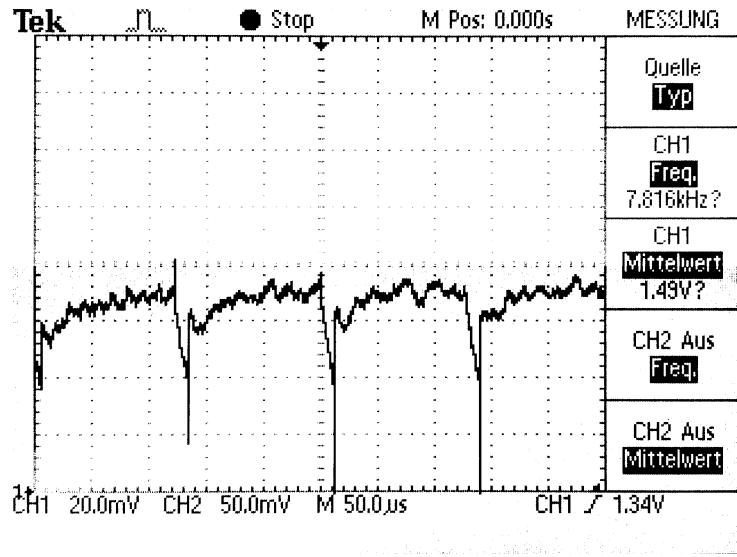


Abbildung 5.4: Störungen der Betriebsspannung mit Akku

Vergleicht man die Messungen im Akkubetrieb (Abb. 5.6) mit den Messungen im Netzbetrieb (Abb. 5.5), kann man erkennen das das Netzteil versucht die Schwankungen auszugleichen.

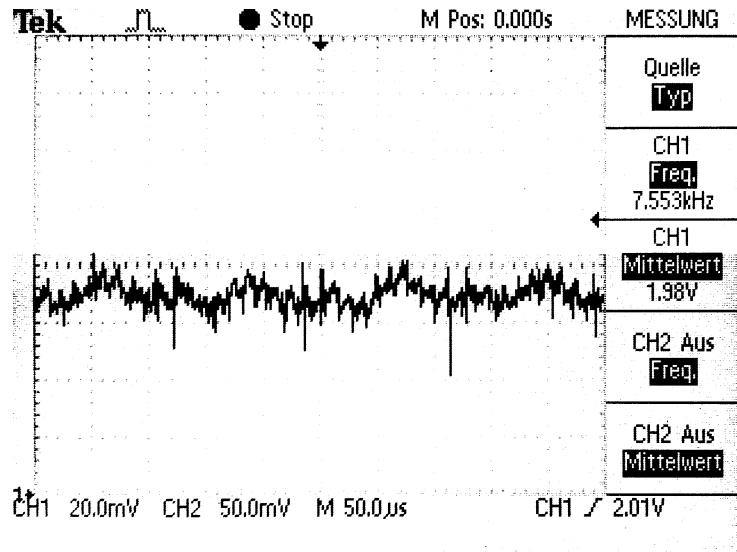


Abbildung 5.5: Störungen der Betriebsspannung mit Netzteil

Unter Last bei einem Tastgrad von 50:256 beträgt die Amplitude der Schwankungen im Akkubetrieb über 3 V. Die Messung wurde ohne 1:10 Teilung durchgeführt. Erstaunlicherweise war trotz der starken Schwankungen noch ein stabiler Betrieb des Fahrzeugs möglich. Dies wurde jedoch nur ca. 30 Sekunden getestet, da die hohe Belastung zu einer starken Wärmeentwicklung des Motors führte.

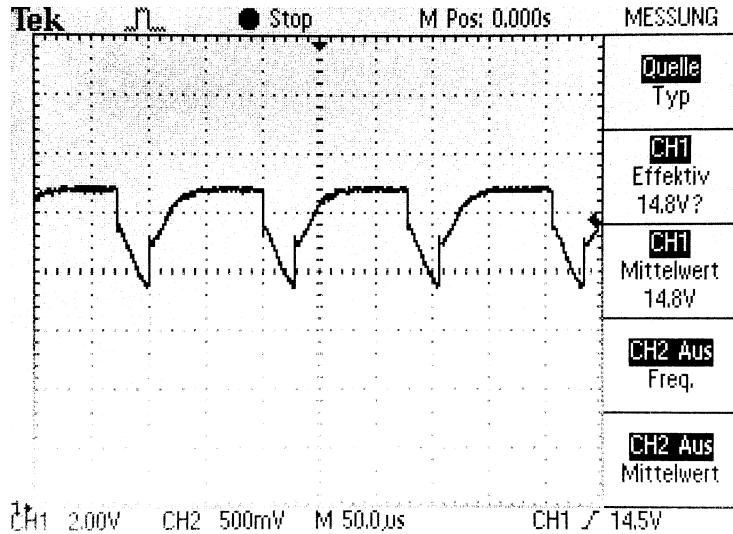


Abbildung 5.6: Störungen der Betriebsspannung mit Akku unter Last

Betrachtet man das 5V-Netz unter diesen Bedingungen (Abb. 5.7) zeigt sich die beeindruckende Leistung des 5V Schaltreglers. Das 5V-Netz scheint trotz der immensen Störung stabiler zu sein als bei niedriger Motorlast.

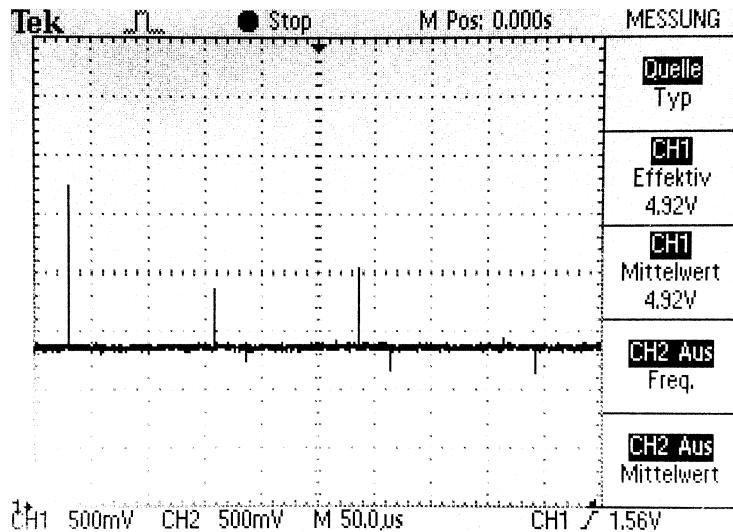


Abbildung 5.7: Störungen des 5V Netzes unter Last

Da das 5V Netz auch unter Last stabil bleibt, sind auch bei höheren Motorlasten keine größeren Störungen in den Sensormesswerten zu erwarten.

### 5.3 Inertialsensor

Da die Auswertung der Inertialsensorik ist nicht Teil dieser Arbeit. Um jedoch die Abhängigkeit der Inertialsensorik von der Motorlast zu untersuchen, wurden einige Messwerte aufgenommen.

Die Messwerte wurden unter Last bei dem angegebenen Tastgrad aufgenommen. Um die Anzahl der Messwerte gering zu halten wurde lediglich die x-Achse der Sensoren untersucht.

*Tabelle 5.3: Accelerometer*

Tastgrad	Erwartungswert [ $\text{m s}^{-2}$ ]	Standardabweichung [ $\text{m s}^{-2}$ ]
20	-0.02034	0.00078
30	-0.02047	0.00225
40	-0.02035	0.00097
50	-0.02034	0.00074
60	-0.02028	0.00189

*Tabelle 5.4: Gyroskop*

Tastgrad	Erwartungswert [ $\text{rad s}^{-1}$ ]	Standardabweichung [ $\text{rad s}^{-1}$ ]
20	0.72961	0.27632
30	0.77174	0.30825
40	0.78556	0.27121
50	0.77792	0.27066
60	0.87919	0.23369

*Tabelle 5.5: Magnetometer*

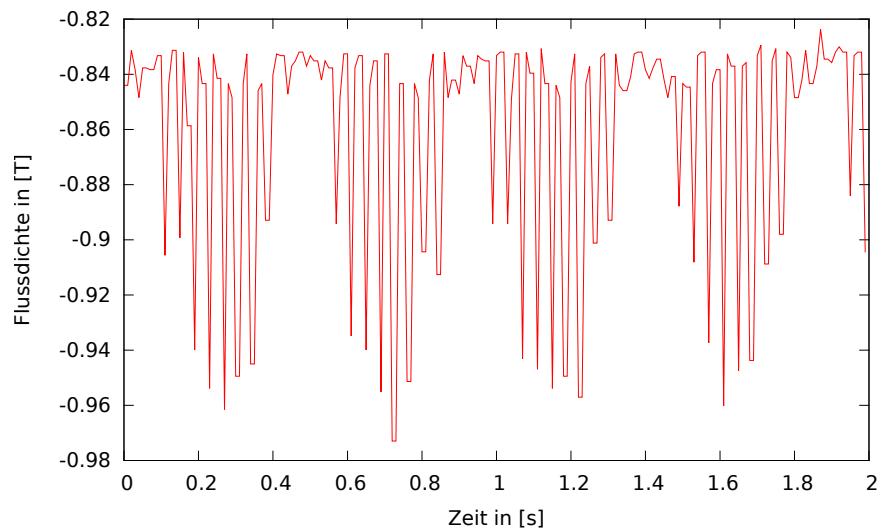
Sowohl beim Accelerometer als auch beim Gyroskop lässt sich anhand der Daten keine Abhängigkeit erkennen.

Tastgrad	Erwartungswert [T]	Standardabweichung [T]
20	0.06974	0.00436
30	0.07602	0.00715
40	0.07336	0.01154
50	0.06504	0.01366
60	0.06564	0.01894

Beim Magnetometer lässt sich jedoch erkennen, dass die Streuung der Daten mit zunehmender Motorlast ebenfalls zunimmt. Da der Motor jedoch ein Magnetfeld erzeugt, welches mit zunehmender Belastung zunimmt, war ein Einfluss auf die Messwerte zu erwarten. Diese Störungen sollten bei der Verwendung des Magnetometers beachtet werden.

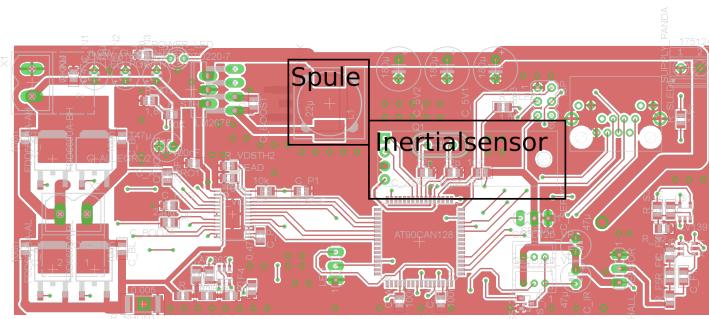
Weiter Störungen beim Magnetometer entstehen, wenn der Servomotor belastet wird (Abb. 5.8). Durch die Art und Weise wie der Servomotor arbeitet, treten starke Schwankungen im Stromverbrauch des Servomotors auf. Bei der Untersuchung des 5V-Netzes konnten allerdings keine Störungen in der Spannungsversorgung festgestellt werden.

Abbildung 5.8: Störungen des Magnetometers



Nach ausführlicher Untersuchung konnte der Fehler jedoch im Platinenlayout ausgemacht werden.

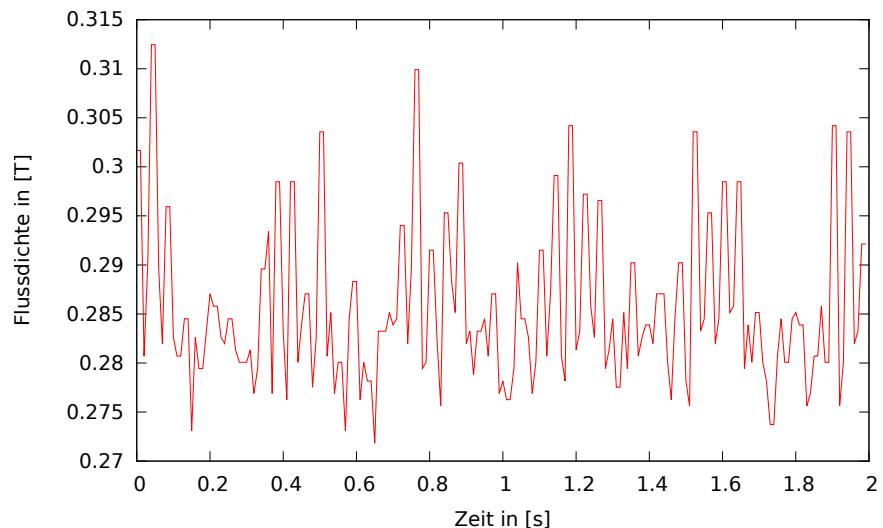
Abbildung 5.9: Platinenlayout mit IMU und Spule



Der Inertialsensor wurde bei der Entwicklung des Layouts zu nahe an die Filterspule des Schaltreglers gelegt. Da das Magnetfeld einer Spule proportional zum durchfließenden Strom ist, treten die Störungen besonders bei starken Lastwechseln im 5V-Netz auf. Die konstante Belastung belegt das Magnetometer nur mit einem Offset.

Durch die Verschiebung des Inertialsensors, mit Hilfe einer Lochrasterplatine, konnten die Störungen jedoch stark verringert werden (Abb. 5.10).

Abbildung 5.10: Störungen des Magnetometers nach der Korrektur



Standardabweichung mit  $t$   
geschrieben??

## 5.4 Infrarotsensoren

Tabelle 5.6: Infrarotsensoren

Im Kapitel Umsetzung wurde bereits über die hohen Anforderungen der GP2D Sensoren berichtet. Da bei starker Motorlast kleine Störungen im 5V-Netz auftreten, wird nun untersucht, wie groß der Einfluss der Störungen auf die Infrarotsensoren ist. Die Werte wurden dabei wieder in unterschiedlichen Lastszenarien aufgezeichnet.

Tastgrad	Erwartungswert [V]	Standardabweichung [V]
0	1.535	0.0243
30	1.533	0.0248
60	1.570	0.1185

Bis zu einem Tastgrad von 30:256 ist keine signifikante Verschlechterung der Messwerte erkennbar, bei höheren Tastgraden ist jedoch mit einer signifikanten Verschlechterung der Messwerte zu rechnen. Abbildung 5.11 und 5.12 zeigen den Verlauf des Signals.

Abbildung 5.11: Störungen des Sensorsignals bei einem Tastgrad von 30:256

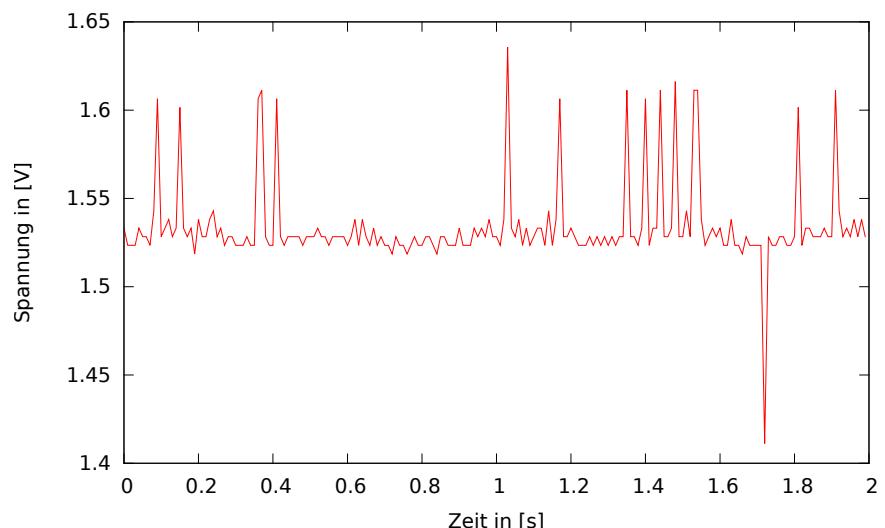
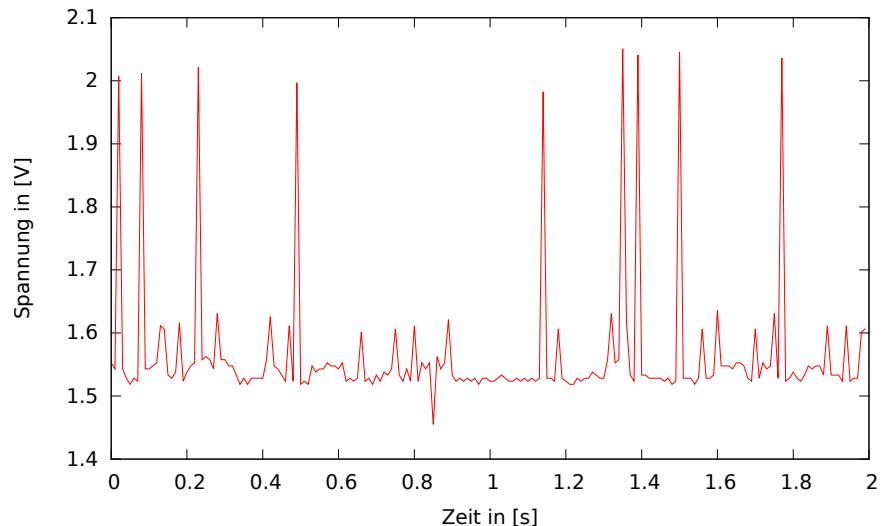


Abbildung 5.12: Störungen des Sensorsignals bei einem Tastgrad von 60:256



Trotz der Störungen bei hohen Tastgraden befinden sich die Störungen im akzeptablen Bereich, da die Distanzsensoren ausschließlich während des Einparkvorganges genutzt werden. Dieser findet jedoch nur bei niedrigen Geschwindigkeiten statt, sodass dort keine hohen Tastgrade vonnöten sind.

## 5.5 Odometrie

Dringend überarbeiten

Zur Auswertung der Odometrie wird das errechnete Verhältnis von Ticks zu Meter von 4760 validiert. Dazu wurden 10 Messungen durchgeführt, in denen das Fahrzeug eine Strecke von 10m möglichst gerade zurücklegen musste. Dabei wurden Start- sowie Endwert der vergangenden Motorticks aufgenommen und die Differenz berechnet.

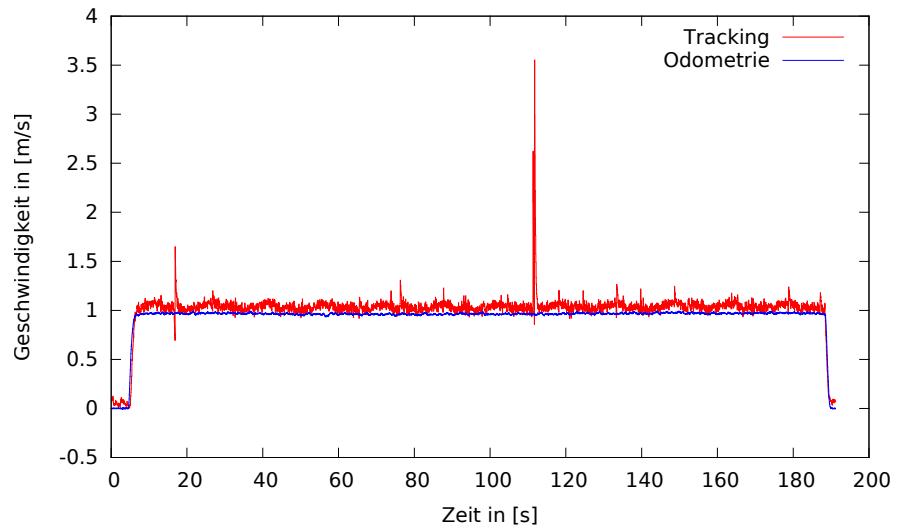
*Tabelle 5.7: Magnetometer*

Startwert	Endwert	Differenz	pro Meter
-1050	48839	49889	4988,9
919	50796	49877	4987,7
-3692	46145	49837	4983,7
78153	127970	49817	4981,7
73414	123348	49934	4993,4
118248	168199	49951	4995,1
195752	245667	49915	4991,5
255970	305846	49876	4987,6
349832	399721	49889	4988,9
418516	468432	49916	4991,6

Der Mittelwert über alle Messungen beträgt 4989,01 und weicht somit nach oben vom errechneten Wert ab. Diese Abweichung lässt sich durch den Schlupf erklären. Der Wert wurde ebenfalls dadurch vergrößert, dass das Fahrzeug, während der Messungen, nicht exakt gerade gefahren ist.

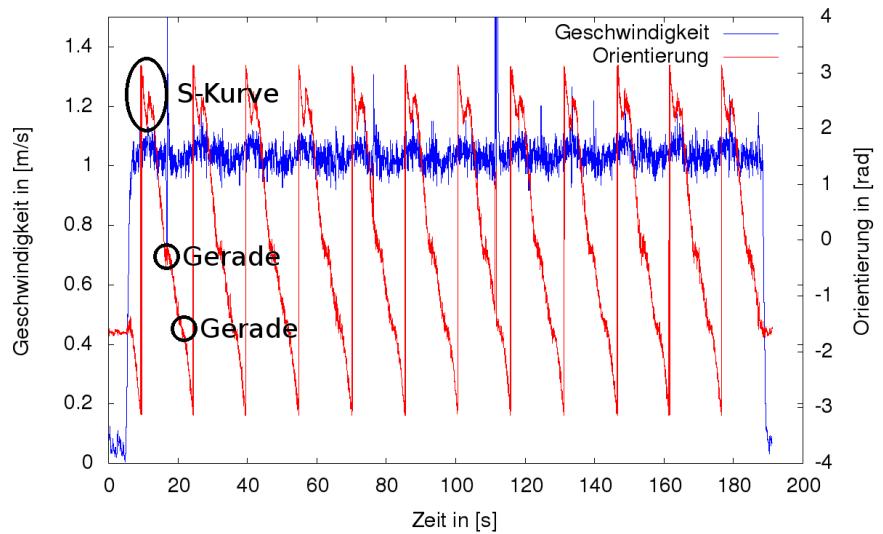
Im Folgenden wurde die vom Auto berechnete Geschwindigkeit mit einem Kamera-basierenden Trackingsystem verglichen. Das Fahrzeug fuhr dabei einen Rundkurs (siehe Abb. 5.15), wobei ein Geschwindigkeitsregler das Fahrzeug anhand der Odometriedaten auf  $1 \text{ m s}^{-1}$  hielt.

Abbildung 5.13: Vergleich Odometrie mit Kameratracking



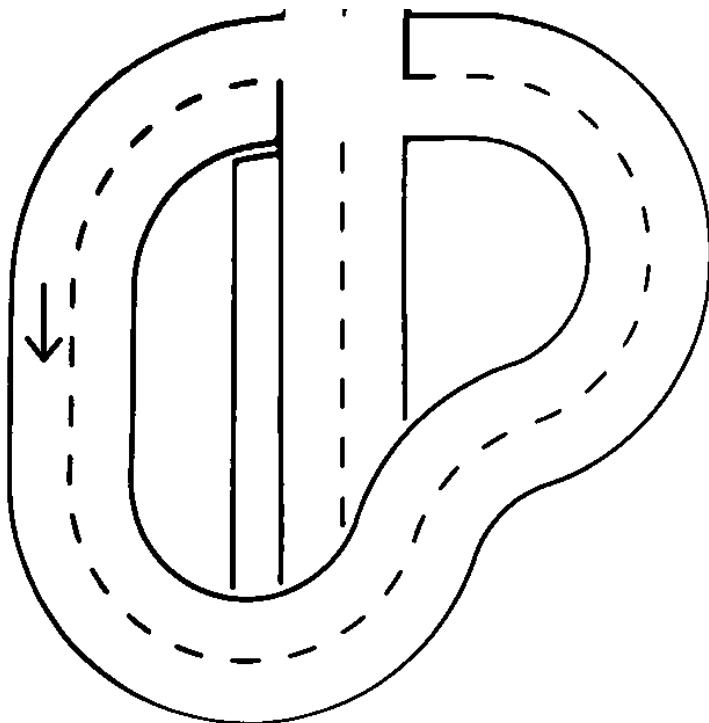
Auffällig ist, dass die Geschwindigkeit vom Trackingsystem mit einer sinusartigen Schwingung überlagert ist. Eine Erklärung zeigt ein Vergleich der Geschwindigkeit mit der Orientierung des Fahrzeugs (Abb. 5.14).

Abbildung 5.14: Orientierung und Geschwindigkeit



Während der Datenaufzeichnung fuhr das Fahrzeug den OttoCar-Rundkurs ab, es startete an der Position des Pfeils in dessen Richtung.

Abbildung 5.15:  
OttoCar-Rundkurs



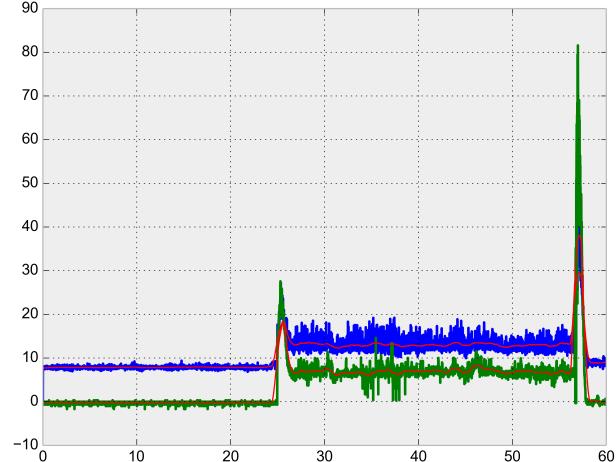
Sobald das Auto beschleunigt hat fängt die Orientierung an sich zu ändern. Direkt nach dem 1. Sprung der Orientierung (von  $-Pi$  zu  $Pi$ ) beginnt sie ihre Orientierung zu ändern. Währenddessen durchfährt das Fahrzeug die S-Kurve des Rundkurses. Der Kurvenradius der S-Kurve ist sehr gering, daher fährt das Fahrzeug beinahe gerade. An dieser Stelle hat die Geschwindigkeit des Trackings ihr Maximum, das Fahrzeug fährt also in Kurven bei gleicher Drehzahl langsamer zu fahren, was sich durch den erhöhten Schlupf in den Kurven erklären lässt. Die Abweichungen der Geschwindigkeit in den Kurven ist jedoch sehr gering, sodass diese nur wenig ins Gewicht fallen. Der Wert der Geschwindigkeit ist mit Hilfe des Trackings nicht verifizierbar, da eine Kalibrierung dessen auf Grund des Entwicklungsstandes nicht erfolgt ist. Das Tracking ist somit als Referenzsystem nicht geeignet.

## 5.6 Stromverbrauch

Der Stromverbrauch des Fahrzeugs ist ein wichtiges Kriterium in den statischen Disziplinen. Um den Stromverbrauch im laufenden Betrieb messen zu können, wird hier ein Versuchsaufbau verwendet welcher der Messung des Motorstromes ähnelt. Die Schaltung besteht dabei aus einem Shuntwiderstand, einer aktiven Filterschaltung und einem Arduino, welcher die Daten zum NUC weiterleitet. Der Vorteil dieser Methode ist, dass die Daten unter realen Bedingungen in Echtzeit aufgezeichnet werden können. Abb. 5.16 zeigt den Verlauf des Stromes während folgendem Szenario: Bis ca. 25s steht das Auto, sämtliche Software ist dabei auf dem Fahrzeug aktiv. Ab 25s beschleunigt das Fahrzeug auf  $1.3 \text{ m s}^{-1}$  und verharrt dort bis ca. Sekunde 57, in welcher es gegen eine Wand fährt. Der grüne Graph stellt dabei den Strom durch den Motor dar, während der blaue Graph den Gesamtverbrauch des Fahrzeugs darstellt. Die roten Linien stellen einen gleitenden Mittelwert aus den letzten 200 Messwerten dar. Gut zu erkennen ist, dass der Stromverbrauch des Fahrzeugs

im Stand unter 10 Watt liegt. Der Mittelwert des Verbrauches im Stand beträgt 7,9 Watt, während das Fahrzeug während der Fahrt knapp 13 Watt an Leistung aufnimmt. Nur während das Fahrzeug beschleunigt benötigt es für die Dauer des Beschleunigungsvorganges mehr Leistung. Fährt das Fahrzeug gegen ein Hindernis, sodass die Räder blockieren, befindet sich der Motor im Kurzschlussbetrieb, dabei reduziert sich sein Widerstand auf den Ohmschen Widerstand des Motors, was zu einem hohen Stromfluss durch den Motor führt. Dauerhaft kann das durch Überhitzung zur Zerstörung des Motors oder der Treiberplatine führen.

*Abbildung 5.16: Stromverbrauch des Fahrzeuges*



Die starken Schwankungen der Messwerte werden durch die vor den Akku geschaltete Strommessung verursacht. Der Shuntwiderstand in dieser Schaltung hat eine Größe von  $0.2 \Omega$ , was den Innenwiderstand der Akkus stark erhöht und dadurch die Spannungsschwankungen der Betriebsspannung verstärkt.

**Strommessung bei anderen Motormodi**

---

# 6

## Ausblick und Fazit

Im Rahmen dieser Arbeit wurde eine Treiberplatine für das OttoCar geschaffen, ausgehend von einem Anfangsbestand von Komponenten sollte die Platine, diversen Anforderungen genügen. Die Anforderungen umfassten dabei die Ansteuerung des Motors und des Servomotors, welche erfolgreich umgesetzt wurden, und zuverlässig ihren Dienst verrichten. Auch die Sharp GP2D Sensoren konnten erfolgreich in das System integriert werden.....Evaluierung Sensorqualität Die benötigte Beleuchtung wurde durch einen LED-Streifen mit WS2812 Controllern umgesetzt, welche durch ihre Bauart eine einfache Integration in das Auto ermöglichen. Da diese als RGB-LEDs ausgeführt sind Übertreffen sie die Anforderungen sogar, da sie nahezu jede mögliche Farbe Annnehmen können. Innerhalb der Software können diese einzeln Angesteuert werden und so zu Fehlerdiagnose genutzt werden. Bei der Integration des Sparkfun Inertialsensors traten Probleme mit dem Magnetometer auf, welche im Nachhinein nur provisorisch abgemildert werden konnten. Trotz diese Probleme kann der Inertialsensor erfolgreich zur Inertialnavigation eingesetzt werden

Die Auslegung der 5V Stromversorgung kann ebenfalls als Erfolg gewertet werden, der Regler schafft es trotz enormer Schwankungen der Betriebsspannung die Spannung, bis auf minimale Störungen stabil zu halten, so dass es zu keiner Störung der Sensormesswerte kommt. Die Odometrie konnte die Anforderungen leider nicht vollständig erfüllen, da nur eine Wegmessung nach vorne möglich ist. Die Qualität der Geschwindigkeitsangabe ist dafür auch bei Mittleren Geschwindigkeiten bis  $2 \text{ m s}^{-1}$  auch in Kurven gut brauchbar. Durch die Fusion, der Inertialsensorik und Odometrie konnte ein Inertialnavigationssystem entwickelt werden, welche eine gute Aussage über die Postion des Autos machen kann.

Als großer Erfolg, kann die Energieeffizienz des Systems gewertet werden! In zukünftigen entwicklungen sollte die Leistungsellektronik jedoch von den Sensorkmgetrennt werden, um Störungen zu vermeiden

---

# Abbildungsverzeichnis

1.1	Möglicher Rundkurs [Lehb]	4
3.1	Konzept	9
3.2	Vierquadrantensteller [Wikc]	10
4.1	Vierquadrantensteller [Wikc]	12
4.2	Slow decay, diode recirculation, high-side PWM [All]	13
4.3	Slow decay, SR active, high-side PWM [All]	14
4.4	Slow decay, SR active, low-side PWM [All]	14
4.5	Fast decay, diode recirculation [All]	15
4.6	Fast decay, SR active, full four-quadrant control [All]	15
4.7	Schaltplan Allegro A3941	16
4.8	Spannungsteiler zur Akkuspannungsmessung	18
4.9	Spannung am Shunt + PWM	18
4.10	Sallen-Key Tiefpass mit Shunt	20
4.11	Spannung am Shunt + PWM	21
4.12	Tiefpass Toleranzfeld	22
4.13	Polstellen-Nullstellendiagramm, Butterworth 2. Ordnung	23
4.14	Frequenzgang des berechneten Filters	26
4.15	Frequenzgang des berechneten Filters mit finalen Bauteilwerten	26
4.16	Sprungantwort des Filters	27
4.17	Restwelligkeit des Filters	27
4.18	Kaskadierung der LEDs [Wor]	28
4.19	Codierung des LED Signals [Wor]	29
4.20	Prinzip der Lasertriangulation [Wikb]	29
4.21	Ausgangssignal GP2D120	30
4.22	Ausgangssignal GP2D120 entstört	30

4.23	Aufbau eines Gleichstrommotors [Wika]	31
4.24	Schmitt-Trigger Schaltung	32
4.25	Motor mit Inkrementalgeber	32
4.26	Regulator Wirkungsgrad [Ins]	35
4.27	Schaltplan des 5V Reglers	35
4.28	Ersatzschaltbild eines Kondensators	36
4.29	Fahrzeug Platinen-Position	36
4.30	Platinenlayout	36
4.31	Lese Daten	39
4.32	Schreibe Daten	40
5.1	Spannung am Shunt	41
5.2	Spannung nach dem Filter	42
5.3	Störungen im 5V Netz	43
5.4	Störungen der Betriebsspannung mit Akku	44
5.5	Störungen der Betriebsspannung mit Netzteil	44
5.6	Störungen der Betriebsspannung mit Akku unter Last	45
5.7	Störungen des 5V Netzes unter Last	45
5.8	Störungen des Magnetometers	47
5.9	Platinenlayout mit IMU und Spule	47
5.10	Störungen des Magnetometers nach der Korrektur	47
5.11	Störungen des Sensorsignals bei einem Tastgrad von 30:256	48
5.12	Störungen des Sensorsignals bei einem Tastgrad von 60:256	48
5.13	Vergleich Odometrie mit Kameratracking	50
5.14	Orientierung und Geschwindigkeit	50
5.15	OttoCar-Rundkurs	51
5.16	Stromverbrauch des Fahrzeuges	52

---

# Literaturverzeichnis

- [All] Allegro. Datenblatt automotive full bridge mosfet driver. <http://www.allegromicro.com/~/media/Files/Datasheets/A3941-Datasheet.ashx>. Online; accessed 20-Mai-2014.
- [Alt08] David M. Alter. Using pwm output as a digital-to-analog converter on a tms320f280x digital signal controller. Technical report, Texas Instruments, 2008.
- [Atm] Atmel. Datenblatt at90can. <http://www.atmel.com/Images/doc7679.pdf>. Online; accessed 20-Mai-2014.
- [Bar] Omar R Barron. Panda test data. [http://omappedia.org/wiki/Panda\\_Test\\_Data](http://omappedia.org/wiki/Panda_Test_Data). Online; accessed 30-Juni-2014.
- [Dev] Analog Devices. Datenblatt ad8628. <http://www.atmel.com/Images/doc7679.pdf>. Online; accessed 20-Mai-2014.
- [Gra] Graupner 4421. [http://www.graupner.de/fileadmin/downloadcenter/servolist/e/20080804095209\\_servoliste\\_screen.pdf](http://www.graupner.de/fileadmin/downloadcenter/servolist/e/20080804095209_servoliste_screen.pdf). Online; accessed 20-September-2014.
- [Hug07] Manfred Hugenschmidt. *Lasermesstechnik - Diagnostik der Kurzzeitphysik ; 12 Tabellen*. Springer-Verlag, Berlin Heidelberg New York, 2007.
- [Ins] Texas Instruments. Datenblatt lm2678 simple switcher. <http://www.ti.com/lit/ds/symlink/lm2678.pdf>. Online; accessed 20-Mai-2014.
- [Int] Datenblatt intel nuc board d34010wyb. [http://downloadmirror.intel.com/23090/eng/D54250WYB\\_D34010WYB\\_TechProdSpec06.pdf](http://downloadmirror.intel.com/23090/eng/D54250WYB_D34010WYB_TechProdSpec06.pdf). Online; accessed 20-September-2014.
- [Kru00] Gerhard Krucker. Elektronische signalverarbeitung. 2000.
- [Leha] Carolo-cup. <https://wiki.ifr.ing.tu-bs.de/carolocup/carolo-cup>. Online; accessed 20-September-2014.

- [Lehb] Carolo-cup regelwerk 2015. <https://wiki.ifr.ing.tu-bs.de/carolocup/system/files/Hauptwettbewerb2015.pdf>. Online; accessed 20-September-2014.
- [mik14] mikrocontroller.net. Mosfet-treiber. <http://www.mikrocontroller.net/articles/MOSFET-%C3%9Cbersicht#MOSFET-Treiber>, Juni 2014. Online; accessed 17-Juni-2014.
- [Mor] Kurt Moraw. Empfänger- stromversorgung. <http://www.flyheli.de/rxversorgung.htm>. Online; accessed 30-Juni-2014.
- [Omaa] Pandaboard faq. [http://omappedia.org/wiki/PandaBoard\\_FAQ#What\\_are\\_the\\_specs\\_of\\_the\\_Power\\_supply\\_I\\_should\\_use\\_with\\_a\\_PandaBoard.3F](http://omappedia.org/wiki/PandaBoard_FAQ#What_are_the_specs_of_the_Power_supply_I_should_use_with_a_PandaBoard.3F). Online; accessed 30-Juni-2014.
- [Omab] Prof. Dr.-Ing. Abbas Omar. Vorlesungsscript analoge filter.
- [rob] Iq-620cmg coreless. <https://www.hype-rc.de/deu/shop/product/080-620cmg/servo-iq-620cmg-coreless.html>. Online; accessed 20-September-2014.
- [Sem] Fairchild Semiconductor. Datenblatt fdd6690a. <https://www.fairchildsemi.com/datasheets/FD/FDD6690A.pdf>. Online; accessed 20-Mai-2014.
- [Sha] Sharp. Datenblatt gp2d120. [http://www.sharpsma.com/webfm\\_send/1205](http://www.sharpsma.com/webfm_send/1205). Online; accessed 20-Mai-2014.
- [STM14] STMicroelectronics. L298 dual full-bridge driver. [https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf), Juni 2014. Online; accessed 17-Juni-2014.
- [USB] Usb 2.0 specification. [http://www.usb.org/developers/docs/usb20\\_docs/#usb20adopters](http://www.usb.org/developers/docs/usb20_docs/#usb20adopters). Online; accessed 20-September-2014.
- [Wika] 2 pole dc motor pm. [http://de.wikipedia.org/wiki/Datei:Motor\\_DC\\_2Pole.png](http://de.wikipedia.org/wiki/Datei:Motor_DC_2Pole.png). Online; accessed 30-September-2014.
- [Wikb] Laserprofilometer. [http://de.wikipedia.org/wiki/Datei:Laserprofilometer\\_DE.svg](http://de.wikipedia.org/wiki/Datei:Laserprofilometer_DE.svg). Online; accessed 30-September-2014.
- [Wikc] Vierquadrantensteller. <http://de.wikipedia.org/wiki/Vierquadrantensteller>. Online; accessed 20-September-2014.
- [Wor] Worldsemi. Datenblatt ws2812. <http://www.adafruit.com/datasheets/WS2812.pdf>. Online; accessed 20-Mai-2014.

---

## Eidesstattliche Erklärung

Ich erkläre, dass ich meine Bachelor-Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe und dass ich alle Stellen, die ich wörtlich oder sinngemäß aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe. Die Arbeit hat bisher in gleicher oder ähnlicher Form oder auszugsweise noch keiner Prüfungsbehörde vorgelegen.

Ich versichere, dass die eingereichte schriftliche Fassung der auf dem beigefügten Medium gespeicherten Fassung entspricht.

Magdeburg, den 6. Oktober 2014

---

Julian-B. Scholle