



Bachelorarbeit

Hardware/Software Codesign

Julian-Benedikt Scholle
26. September 2014

Betreuer

Dr. Ing. Sebastian Zug
Dipl.-Inform Christoph Steup

Inhaltsverzeichnis

1	Einleitung	4
1.1	Carolo Cup	4
1.2	Das Auto	5
2	Anforderungsanalyse	6
2.1	Anforderungen laut Regelwerk	6
2.1.1	Fahrzeugantrieb und Energieversorgung	6
2.1.2	Fahrzeugantrieb und Energieversorgung	6
2.1.3	RC-Modus	6
2.1.4	Signalleuchten	7
2.2	Anforderungen durch Bewertungskriterien	7
2.3	Anforderungen durch Teammitglieder	7
2.4	Auswertung der Anforderungen	7
3	Konzept	9
3.1	Änderung der Anforderungen	9
4	Umsetzung	11
4.1	Motoransteuerung	11
4.1.1	Treiberbausteine	11
4.1.2	Mosfettreiber	12
4.2	Funk Empfänger	14
4.3	Beleuchtung	15
4.4	Distanzsensoren	16
4.4.1	Messprinzip	16
4.4.2	Probleme der GP2D Sensoren	16
4.4.3	Auswertung des Messsignales	17

4.5	Motor Drehzahlmessung	17
4.5.1	Hallsensor	18
4.6	Motorstrommessung am Shunt	18
4.6.1	Problem	19
4.6.2	Prinzip der Strommessung	19
4.6.3	Anforderungen	19
4.6.4	Bestimmung des Filtertyps	19
4.6.5	Die Filterschaltung	20
4.6.6	Dimensionierung des Verstärkers	20
4.6.7	Anforderungen an den Filter	21
4.6.8	Filterentwurf	21
4.7	Auslegung der Stromversorgung	27
4.7.1	Stromverbrauch der Komponenten	28
4.7.2	Auswahl des Reglers	29
4.8	Software	30
4.8.1	Software auf dem µController	30
4.8.2	Client Programm auf der Recheneinheit	30
4.8.3	Übertragungsprotokoll	30
5	Evaluierung	32
5.1	Strommessung	32
5.2	Stromverbrauch	33
5.3	Infrarotsensoren	33
5.4	Inertialsensor	33
5.5	Zeitverhalten der seriellen Verbindung	34
5.5.1	VoltageCurrent	34
5.5.2	Distance	34
5.5.3	Infrared	35
5.5.4	uCTime	35
5.5.5	Motor	35
5.5.6	IMU	36
5.5.7	Gesamt	36
6	Fazit	38
7	Ausblick	39

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 Autos veranschaulicht werden. Dabei werden auch die Probleme eines Projekts mit sich dynamisch ändernden Anforderungen gezeigt.

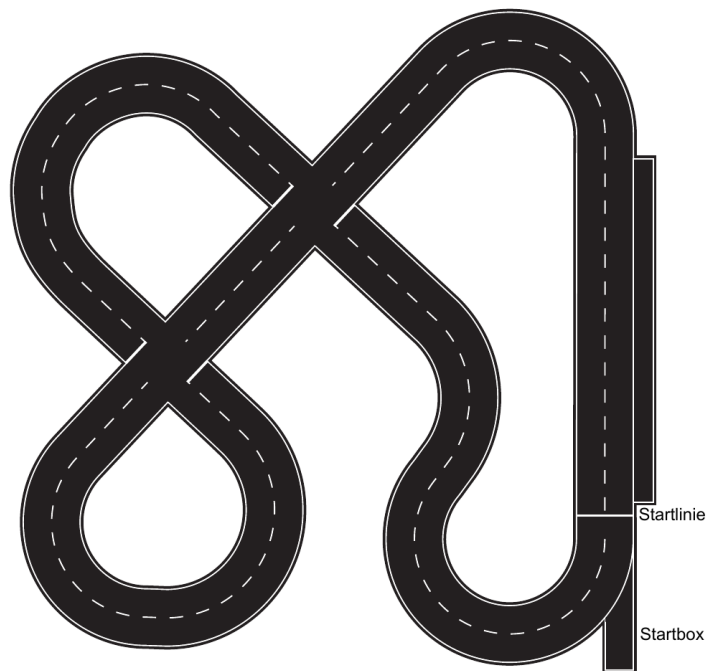
1.1 Carolo Cup

Der „Carolo-Cup“ ist ein jährlicher Hochschulwettbewerb der Technischen Universität Braunschweig, der Studenten die Möglichkeit, sich mit der Entwicklung und Umsetzung von autonomen Modellfahrzeugen auseinanderzusetzen [feFTBa]. 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 ihr 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.

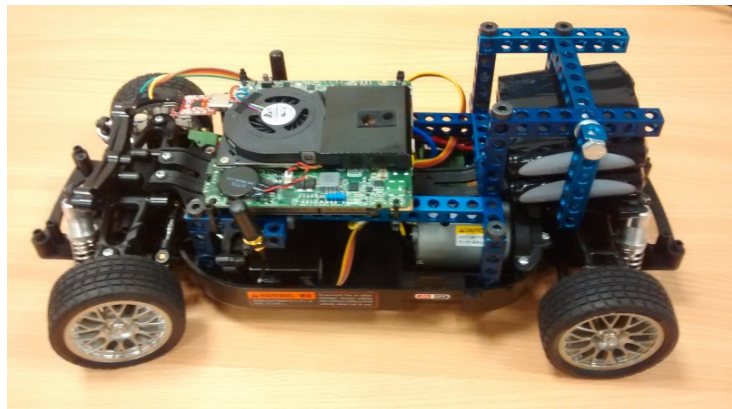
Die dynamischen Disziplinen bestehen dabei aus mehreren Szenarien, dem parallelen Einparken, einem einfachen Rundkurs sowie einem Rundkurs mit Hindernissen. Ein möglicher Rundkurs ist in Abbildung [1.1] zu sehen.

Abbildung 1.1: Möglicher
Rundkurs [feFTBb]



1.2 Das Auto

Abbildung 1.2: Möglicher
Rundkurs [feFTBb]



2.1 Anforderungen laut Regelwerk

Um am „Carolo-Cup“ Teilnehmen zu können ist ein Regelkonformes Fahrzeug nötig, darum wird nun ein Auszug aus den Anforderungen an das Auto kurz aufgelistet und ausgewertet. Alle Anforderungen können im Regelwerk des „Carolo-Cup“ nachgelesen werden [feFTBb]

2.1.1 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.

2.1.2 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.

2.1.3 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 Vorort genutzte Kamertechnik 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.

2.1.4 Signalleuchten

Durch die Anlehnung des Wettbewerbes an den realen Straßenverkehr muss das Auto über alle in echten Auto vorhandene Signalleuchten besitzen. Dazu gehören 3 rote Bremslichter am Heck des Autos sowie jeweils 2 gelbe Blinker Rechts und Links am Fahrzeug. Die Blinkfrequenz der Blinker muss 1-Hz betragen.

2.2 Anforderungen durch Bewertungskriterien

Weitere Anforderungen ergeben sich aus den Bewertungskriterien. Während der statischen Disziplinen muss das Team ihr Gesamtkonzept präsentieren. Schwerpunkte dabei sind, Hardware und Software Architektur sowie Energiebedarf und Herstellungskosten [feFTBb]. Daraus entstehen weitere Anforderungen: Energieeffizienz und Herstellungskosten.

In den dynamischen Disziplinen soll das Fahrzeug eine Strecke völlig autonom abfahren. Solch eine wie in Abbildung [1.1] aussehen. Das Fahrzeug darf seine Fahrspur dabei nie mit mehr als einem Rad verlassen. Um die Fahrbahn zu erkennen wird vom Team eine auf dem Auto montierte Kamera verwendet. Da laut Regelwerk keine Daten vom oder zum Auto übertragen werden dürfen, muss die Verarbeitung der Bilder auf dem Auto stattfinden. Daher ist auf dem Auto eine leistungsfähige Recheneinheit vonnöten. Hierzu wurden vom Lehrstuhl boards des Types "Pandaboard ES" zur Verfügung gestellt.

Innerhalb des parallelen Einparkens ist es nötig die Größe der Parklücke zu erkennen, bzw Abstände zu Objekten zu messen.

2.3 Anforderungen durch Teammitglieder

Inertialsensoren..

2.4 Auswertung der Anforderungen

Da ein elektrischer Antrieb vorgesehen ist, muss die nötige Elektronik zur Steuerung des Motors in die Platine integriert werden. Der Einfachheit halber wird die Lenkung von einem Servomotor übernommen. Damit das Auto die im RC-Modus nötigen Funksignale auswerten kann muss ein Empfänger integriert werden. Desweiteren muss das Auto mit den nötigen Signalleuchten ausgestattet sein, dazu gehören Blinker rechts und links jeweils vorne und hinten. Sowie 3 Bremsleuchten an der Rückseite und eine weitere Leuchte welche den RC-Modus anzeigt. Der Vollständigkeit halber kommt hier noch die Frontbeleuchtung hinzu

Zum bestimmen von Entfernungen zu anderen Autos oder Hindernissen werden Distanzsensoren benötigt

Da Energiebedarf eine Anforderung ist sollte bei der Auswahl der Komponenten auf Energieeffizienz geachtet werden.

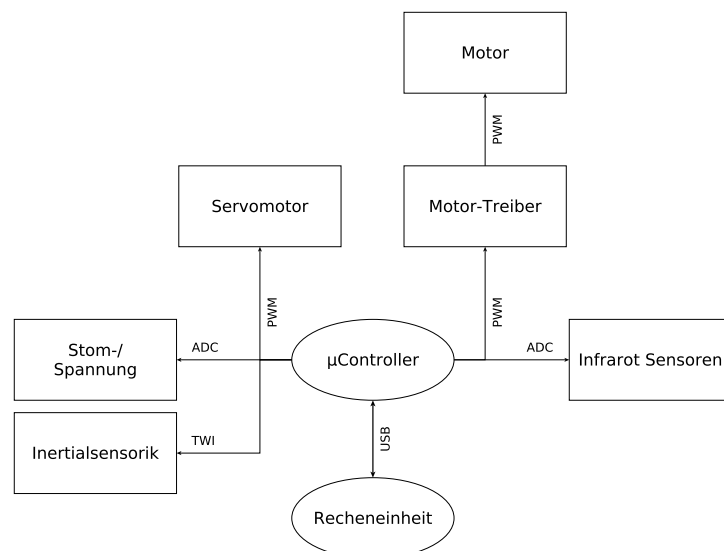
Zusammenfassend hier eine Liste mit den benötigten Komponenten.

- Elektronik zur Motoransteuerung

- Servomotor zur Lenkung
- Funk Empfänger für den RC-Modus
- Beleuchtung
- Distanzsensoren
- Interalsensorik

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 Kommunikation mit der darüber liegenden Recheneinheit. Herzstück der Platine ist dabei ein Atmel AT90CAN128 μ Controller an welchen über verschiedene Protokolle 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 Akkuspannung und dem Motorstrom. Eine Übersicht über die Sensorik bzw. Aktorik und ihre Anbindung ist in Abbildung [3.1] zu sehen.

Abbildung 3.1: Konzept



3.1 Änderung der Anforderungen

Nach der erfolgreichen Teilnahme am “Carolo-Cup Junior” im Februar 2014, begann die Weiterentwicklung des Konzepts, während der Ent-

wicklung kamen einige Flaschenhalse zum vorschein. Sodas in der Projektphase die Hardwareplattform geändert werden musste. Die Rechenleistung der Pandaboards stellte sich als unzureichend heraus und es wurden mehr Distanzsensoren gewünscht. Die Pandaboards wurden nach der Absprache mit dem Team durch ein Intel NUC vom Typ D34010WYB ersetzt. Laut Datenblatt [Cor] besitzt der NUC einen Weitbereichseingang zur Spannungsversorgung, dieser ist für 12-24 Volt zugelassen. Sodas sich die ursprüngliche Wahl der Akkus als Vorteil herausstellt. Die Spannung von 14,4 Volt der 4 Zellen LiPo Akkus passt genau in diesen Bereich, daher kann der NUC direkt an die Akkus angeschlossen werden.

4.1 Motoransteuerung

4.1.1 Treiberbausteine

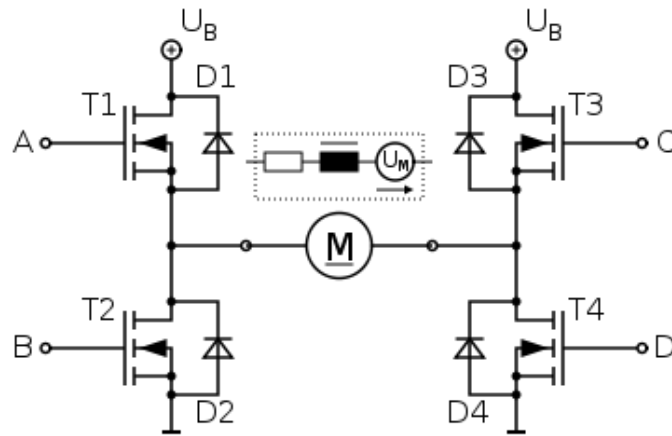
Da die gewählten Akkus eine Spannung von 14,4V aufweisen, kann der original Motortreib leider nicht verwendet werden. Denn dieser benötigt eine Spannung von 7,4V. Da der AVR Mikrokontroller mit 5V betrieben wird, wird ein Motortreiber benötigt der mit den 5V Pegeln arbeiten kann. In vielen Mikrocontroller Projekten und in unserem ersten Prototyp wird der L298 DUAL FULL-BRIDGE DRIVER verwendet. Dieser ist leider auch bei der Benutzung beider Kanäle auf 4 Ampere begrenzt [STM14], was beim Prototy zu einer Permanenten überlastung des Treibers führt. Leider sind keine vollintegrierten Motortreiber mit der benötigten Belastbarkeit verfügbar. Um die benötigte Belastbarkeit zu erreichen wird der zur Ansteuerung benötigte Vierquadrantensteller aus diskreten Mosfets aufgebaut.

Vierquadrantensteller

Definition nach Wikipedia [Wik]:

“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 4.1:
Vierquadrantensteller



Auf Grund der hohen Belastbarkeit werden meist 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 Mosfettreiber, was das Schaltungsdesign enorm vereinfacht.

4.1.2 Mosfettreiber

Verfügbarkeit

Mosfettreiber 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.

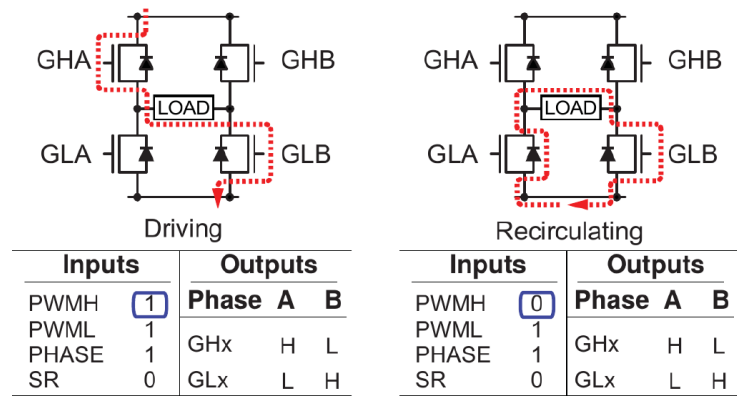
Eine Tabelle auf Mikrocontroller.net [mik14] zeigt eine Auswahl an verfügbaren Mosfettreibern. Dort sind zwei “Full Bridge Driver” aufgeführt, welche für dieses Projekt passend sind. Allerdings fällt die Entscheidung auf einen anderen Treiber, dem Allegro A3941.

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.

Der Treiber lässt sich in verschiedenen Modi betreiben:

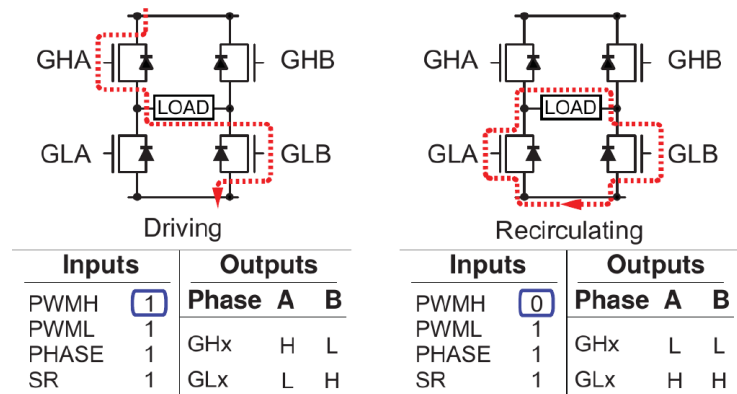
Abbildung 4.2: Slow decay, diode recirculation, high-side PWM



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.

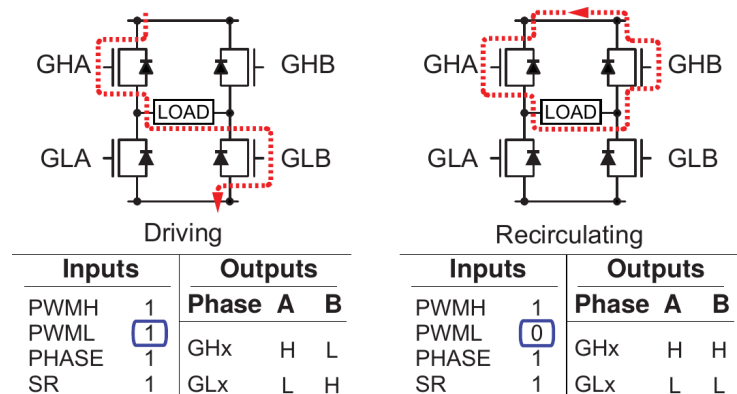
Abbildung 4.3: Slow decay, SR active, high-side PWM



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 Mosfest stärker gebremst als in der Voherigen Konfiguration. Dabei ist darauf zu achten, dass beinahe die gesamte induzierte Spannung über den beiden Mosfets (GLA/GLB) abfällt. Was zu einer starken Hitzeentwicklung führen kann.

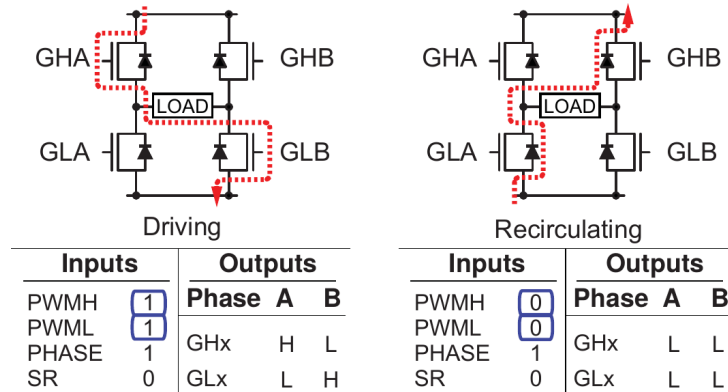
Abbildung 4.4: Slow decay, SR active, low-side PWM



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

Diese Konfiguration entspricht im Grunde den beiden vorherigen, Nur das dass 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)

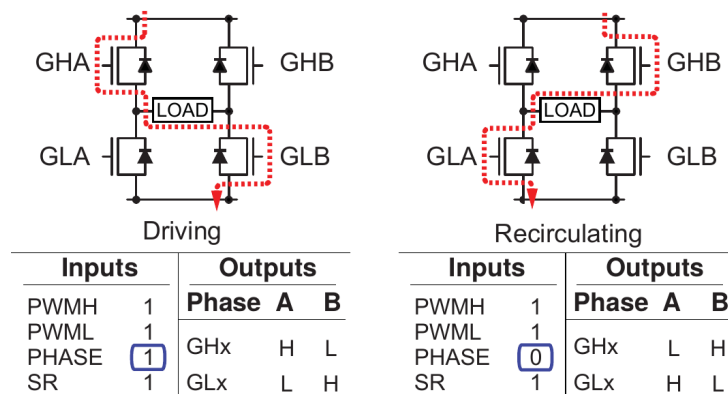
Abbildung 4.5: Fast decay, diode recirculation



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 das 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 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



Diese Konfiguration zeigt den Einfluss des PHASE-Pins. Liegt am PHASE-Pin 1 ein 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.

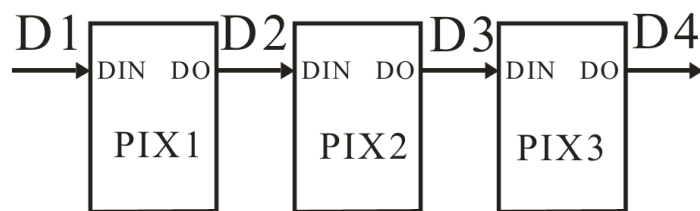
4.2 Funk Empfänger

Da die für die Funkkommunikation das 2,4-GHz Frequenzband nicht genutzt werden darf können naheliegende Techniken wie W-Lan oder Bluetooth nicht genutzt werden.

4.3 Beleuchtung

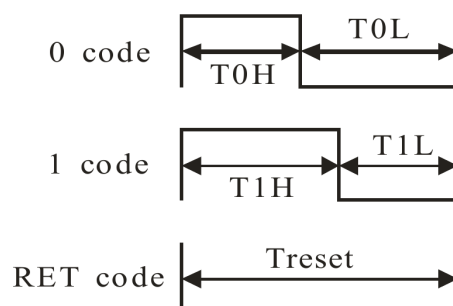
Die einfachste Möglichkeit eine Beleuchtung am Auto zu realisieren sind Leuchtdioden, diese sind in allen erdenklichen Farben zu bekommen. Weiter Vorteile sind das Leuchtdioden sehr Energieeffizient sind, außerdem sind sie zugünstigen Preisen zu bekommen. LEDs stellen keine großen Anforderungen an die 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 Ledstrips mit Chips von "Worldsemi" sind hierbei die prominentesten Vertreter. Zu erwähnen wähen 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 duche eine einzige Siganleitung mit fixem Takt angesteuert, so dass hier eine Separate Taktleitung entfällt. Der Nachteil an dieser Methode ist, dass die Timings genau eingehalten werden müssen um die Daten korrekt zu übertragen. Wie In Abbildung 4.7 werden die LEDs kaskadiert. Die Daten werden dann durch die LEDs geschoben.

Abbildung 4.7: Kaskadierung der LEDs



Um die beleuchtung am Auto zu realisieren werden LEDs mit WS2812 genutzt. Die LEDs können an einem beliebigen Eingang des AVR-Microcontrollers angeschlossen werden. Da die Timings exakt eingehalten werden müssen, ist die Software zur ansteuern in Assembler geschrieben. Jede LED wird mit einem 24Bit Datenwort angesprochen, dieses enthält die Helligkeitsstufen für jede der drei Grundfarben. Die Daten werden dabei 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 4.8: Codierung des LED Signals



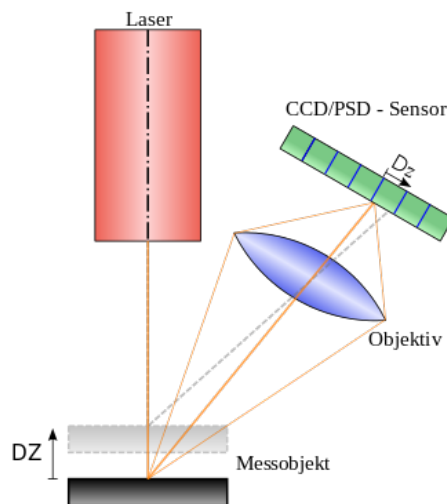
Abschnitt	Beschreibung	Dauer	Abweichung
T0H	0 Code, high Zeit	$0.35\mu s$	$\pm 150ns$
T1H	1 Code, high Zeit	$0.7\mu s$	$\pm 150ns$
T0L	0 Code, low Zeit	$0.8\mu s$	$\pm 150ns$
T1L	0 Code, low Zeit	$0.6\mu s$	$\pm 150ns$
RES	Reset Code, low Zeit	über $50\mu s$	

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 optische Abstandsmessung durch Triangulation projiziert ein Projektor einen Lichtpunkt auf das Messobjekt [4.12]. Ein optischer Sensor misst dann den Winkel des vom Messobjekt reflektierten Lichtes. Durch Triangulation kann dann durch den fest definierten Abstand des optischen Sensors von der Lichtquelle die Entfernung zum Objekt berechnet werden. [Hug07]. In Abbildung 4.12 ist diese Prinzipveranschaulicht

Abbildung 4.9: Prinzip der Lasertriangulation



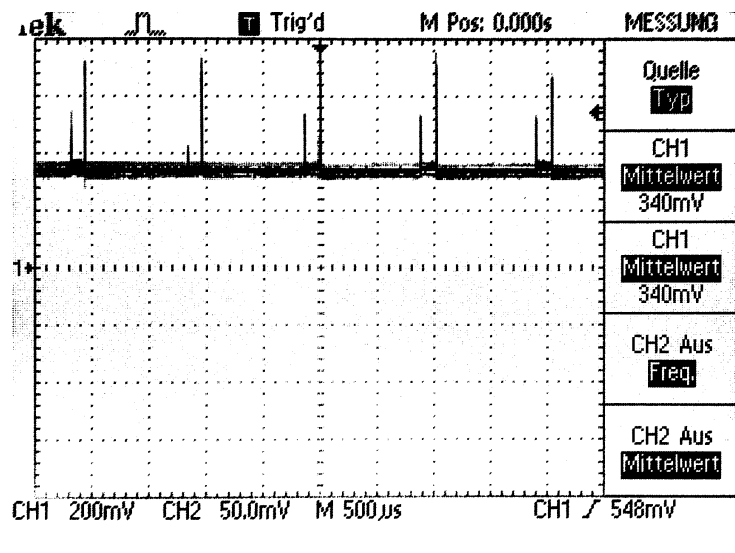
Vorteile des Messprinzips: Da es sich um eine rein trigonometrische Messung handelt, kann sie zur kontinuierlichen Messung von Beweglichen Objekten verwendet werden. Außerdem besitzen Sensoren nach diesem Prinzip einen kleinen Messfleck.

Nachteile: Die Messung ist stark von der Oberfläche des der Messobjektes abhängig, spiegelnde Oberflächen stellen ein großes Problem dar. In Staubigen oder Nebeligen Umgebungen wird das Lichtmöglicherweise zu früh reflektiert.

4.4.2 Probleme der GP2D Sensoren

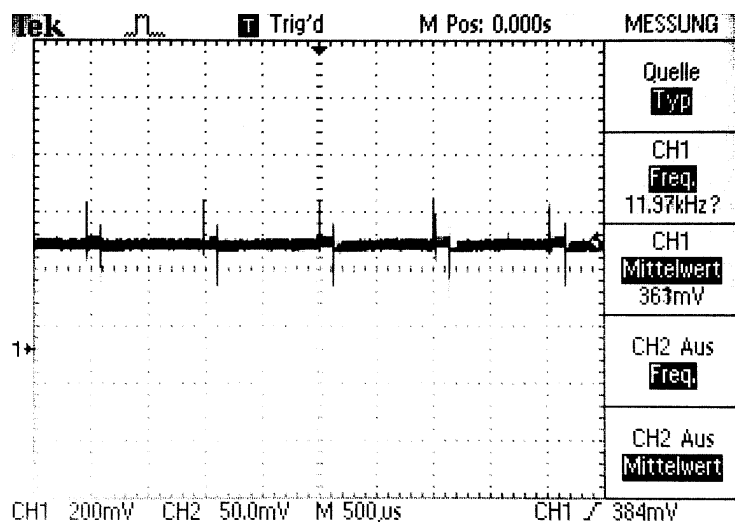
Die Sensoren verfügen über einen analogen Ausgang. Bei analogen Signalen ist generell mit Störungen zu rechnen. Die GP2D Sensoren scheinen hohe Anforderungen an die Energieversorgung zu stellen. Hier ist eine Entstörung mittels Kondensator von nöten da im Messsignal sonst große Spikes entstehen, wie in Abbildung 4.10 zu sehen.

Abbildung 4.10: Ausgangssignal
GP2D120



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

Abbildung 4.11: Ausgangssignal
GP2D120 entstört



4.4.3 Auswertung des Messsignales

Das Messsignal vom Sensor wird über einen ADC-Eingang des AVR Microcontrollers ausgelesen.

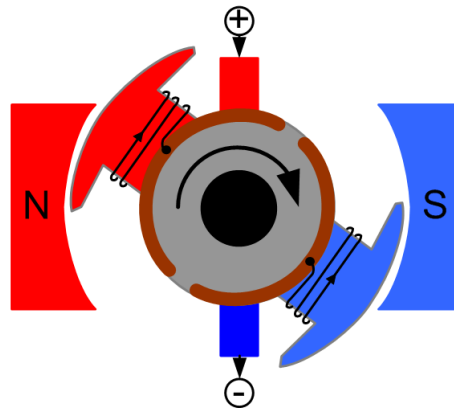
4.5 Motor Drehzahlmessung

Da 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 Leistung am Motor verändern, deshalb ist es nötig diese zu Regeln. Dafür ist jedoch eine Rückführung der Geschwindigkeit des Autos nötig. Eine Aufintegration der beschleunigungs Daten der Interalsensorik führt auf dauer leider zu erheblichen Abweichungen und ist daher für eine Regelung nicht geeignet. Leider ist auch eine Odometrie an den Rädern des Autos aus mechanischen Gründen schwer zu realisieren.

4.5.1 Hallsensor

Durch die feste Übersetzung des Getriebes bietet die Messung der Motordrehzahl eine gute Näherung für die aktuelle Geschwindigkeit. Eine Möglichkeit die Motordrehzahl zu messen ist es das Magnetfeld des Motorankers auszuwerten. Dazu sehen wir uns den Aufbau eines Gleichstrommotors an.

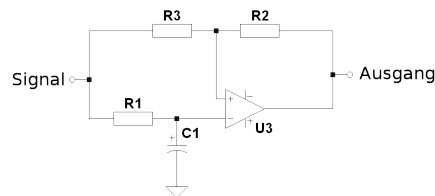
Abbildung 4.12: Aufbau eines Gleichstrommotors



Ein Gleichstrommotor besteht aus zwei grundsätzlichen Teilen, einem unbeweglichen Teil, dem 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, deren Polung jede halbe Umdrehung kommutiert wird. Durch die Kommutierung ändert sich die Polung der Elektromagneten. Das sich so ändernde Magnetfeld kann mit einem Hallsensor ausgewertet werden. Das so entstehende Signal ähnelt dabei über der Zeit einer Sinusschwingung. Mit Hilfe eines Schmitt-Triggers kann daraus ein Drehzahlsignal generiert werden.

Abbildung 4.13: Schmitt-Trigger Schaltung

Der Hallsensor ist erfolgreich durch ein Belüftungsloch im Motor platziert worden. 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 möglich den Schmitt-Trigger so auszulagern, dass er in beide Drehrichtungen zuverlässig funktioniert. Alternativ zu diesem Vorgehen gibt es andere Lösungen. Eine speziell für diesen Motor angefertigte Achsverlängerung wird mit einer Inkrementalgeberscheibe befestigt. Diese wird durch einen Sharp GP1A30R Sensor ausgewertet und liefert ein Drehzahlsignal an den externen Takteingang des Timer 1 vom AVR-Microcontroller.

4.6 Motorstrommessung am Shunt

4.6.1 Problem

An einem mit PWM angesteuertem DC-Motor soll eine Strommessung mit Hilfe eines Shuntwiderstandes durchgeführt werden. Aufgrund der PWM Ansteuerung muss der DC-Anteil aus dem Signal herausgefiltert werden!

4.6.2 Prinzip der Strommessung

Die Messspannung wird über einen Shuntwiderstand zur Masse gemessen! Aufgrund nicht vorhandener Datenblätter des Motors wird von einem experimentell Ermittelten maximalen Strom des Motors ausgegangen. Dieser beträgt bei einer Betriebsspannung von 20V ca. 20A. Da einen Shunt mit einer maximalen Belastbarkeit von 2 Watt eingesetzt wird, darf der maximale Spannungsabfall am Shunt 100mV nicht überschreiten. Nach dem Ohmschen Gesetz ergibt sich dadurch ein Widerstand von $0,005\Omega$ für den Shunt. 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.

Die über den Shuntwiderstand gemessene Spannung soll über den ADC Eingang des Mikrocontrollers eingelesen werden. Vorher jedoch muss das Signal gefiltert werden, da der Strom durch die Ansteuerung mittels der Pulsweitenmodulation nicht konstant ist!

4.6.3 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 unseres 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 $\frac{5}{2^{10}} = 4,88mV$. 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, so das ein Kompromiss gefunden werden muss.

4.6.4 Bestimmung des Filtertyps

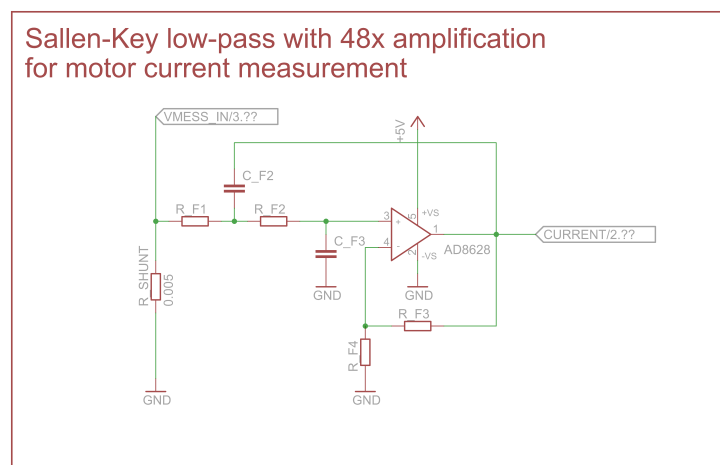
Aufgrund des sehr niedrigen Spannungspegels am Shunt, ist es nötig das Signal zu verstärken. Da zum verstärken des Signals aktive elektronische Elemente notwendig sind, z.B. ein Operationsverstärker, wird an dieser Stelle gleich ein aktiver Filter verwendet. Dieser gibt uns die Möglichkeit des Messsignal zu verstärken und gleichzeitig zu Filtern. Da wir als unser Signal im optimalen Fall eine Gleichspannung darstellt müssen wir die Hochfrequenten Anteile unseres Signales herausfiltern, dies geschieht mit Hilfe eines Tiefpasses. Es gibt im Grunde 2 übliche aktive Tiefpässe, den Tiefpass mit Mehrfachgegenkopplung und den Sallen-Key Filter.

Ersterer verwendet einen invertierenden Verstärker, dieser invertiert das Messsignal. Da der µController 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 Nachteil nicht hat. So dass ab dieser Stelle ein Sallen-Key Tiefpass entworfen wird.

4.6.5 Die Filterschaltung

wie im vorherigen Abschnitt diskutiert wird hier ein Sallen-Key Tiefpass entworfen. Zum Entwurf der Schaltung wurde Eagle genutzt.

Abbildung 4.14: Sallen-Key Tiefpass mit Shunt



4.6.6 Dimensionierung des Verstärkers

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

Für einen Nichtinvertierenden Verstärker ergibt sich dann:

$$v = 1 + \frac{R_{F3}}{R_{F4}}$$

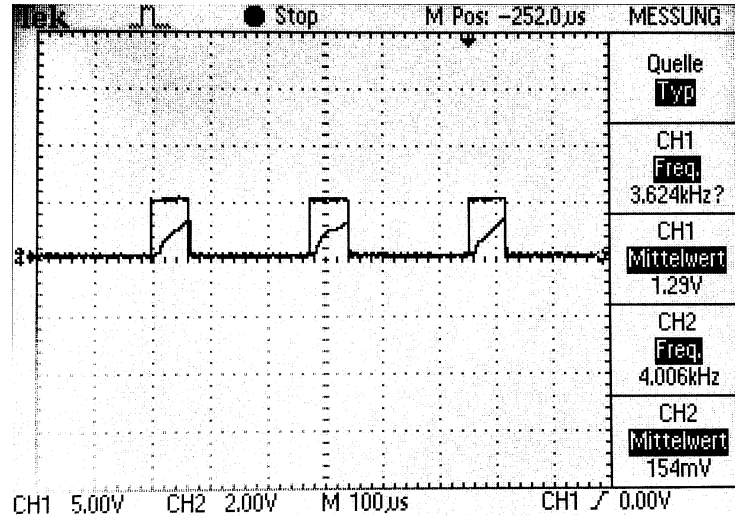
$$50 = 1 + \frac{R_{F3}}{R_{F4}}$$

$$49 \cdot R_{F4} = R_{F3}$$

Wobei $R_{F4} = 47k\Omega$ und $R_{F3} = 1k\Omega$ gewählt werden, was eine Verstärkung von 48 ergibt.

4.6.7 Anforderungen an den Filter

Abbildung 4.15: Spannung am Shunt + PWM



Da dem Messsignal wie in Abbildung 4.15 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 Ursprungssignals 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 Ω_s wird hier die PWM Frequenz angesetzt. Für $H(\omega = 2\pi f_{PWM})$ gilt also:

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

Da das Projekt möglichst kostensparend durchgeführt werden soll, also auch Bauteilsparend, 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.

4.6.8 Filterentwurf

Bestimmung des Filtertyps

Des Filtertyp muss in zweierlei Hinblick bestimmt werden. Einmal im Hinblick auf die Schaltung und seinem Frequenzgang. Im groben gibt es 2 mögliche aktive Tiefpassfilterschaltungen, den Sallen-Key Tiefpass mit nicht invertierendem OPV und dem aktiven Tiefpass mit Mehrfachgegenkopplung (invertierender OPV). Der aktive Tiefpass mit Mehrfachgegenkopplung benötigt allerdings negative Spannungsniveaus die auf der

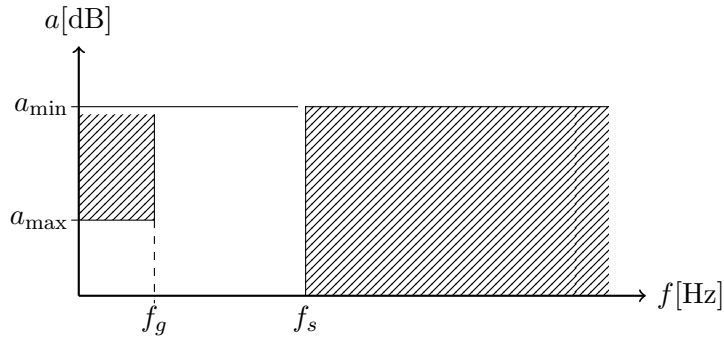
Treiberplatine nicht zur Verfügung stehen, deshalb wird an dieser Stelle nur der Sallen-Key Tiefpass betrachtet. Was den Frequenzgang angeht, gibt es viele Filtercharakteristiken, eine Auswahl an häufig verwendeten Charakteristiken wird hier verglichen.

Der *Butterworth*-Filter besitzt einen maximal flachen Verlauf des Frequenzganges im Durchlassbereich und eine monoton verlaufende Dämpfung im Sperrbereich. Leider hat der Butterworth-Filter nur eine geringe Flankensteilheit im Sperrbereich (20dB/Dekade pro Ordnung). Ein Butterworth-Filter 1. Ordnung entspricht einem 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 Endsignal einer Gleichspannung entsprechen sollte, ist der Butterworth-Filter die bessere Wahl.

Abbildung 4.16: Tiefpass
Toleranzfeld



Für unsere Schaltung wird ein Sallen-Key Tiefpass 2. Ordnung entworfen. Die PWM-Frequenz f_{PWM} beträgt 3,9kHz. Die Sperrfrequenz entspricht der PWM-Frequenz, also der Frequenz unserer Grundschwingung. Ω entspricht der mit der Grenzfrequenz normierten Frequenz $\Omega = \frac{f}{f_g}$. Nach (4.2) ergibt sich für Abbildung 4.16 $f_s = f_{PWM} = 3,9kHz$, $a_{min} = 55,9dB$ und a_{max} wird auf 3dB festgelegt.

Butterworth

Bestimmung der Grenzfrequenz

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

Die Filterordnung nach Butterworth wird nach (4.3) bestimmt. Umgestellt nach Ω_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 Ω_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.

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

Die Grenzfrequenz f_g ergibt sich jetzt aus:

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

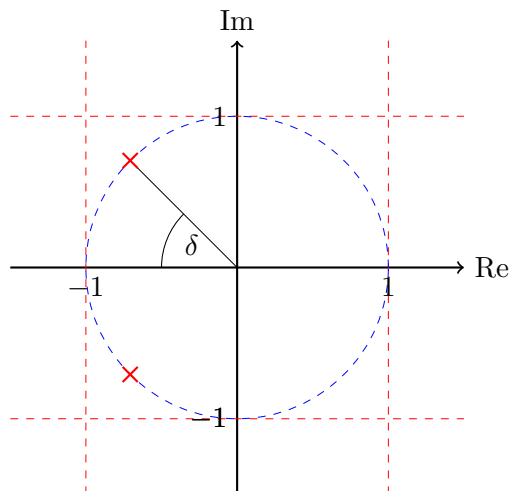
Filterentwurf

Im vorherigen Abschnitt wurde berechnet, dass die Grenzfrequenz der 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 Abbildung [4.17]

Abbildung 4.17:
Polstellen-Nullstellendiagramm,
Butterworth 2. Ordnung



Charakteristisch für den Butterworthfilter ist, dass sich die Polstellen auf einer Kreisbahn befinden. Auf die Grenzfrequenz normiert hat dieser beim Butterworthfilter 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 Ω_P und Polgüte Q_P einfach ablesen lassen. Die Polfrequenz Ω_P ist der Betrag der normierten Polstelle, welcher beim Butterworth-Filter immer eins ist. Die Polgüte

ist abhängig von δ und ergibt sich zu: $Q_P = \frac{1}{2 \cos \delta}$. Für unseren Butterworthfilter ergeben sich also $Q_P = 0,707$ und $\Omega_P = 1$

Betrachten wir die Übertragungsfunktion eines Sallen-key Tiefpasses 2. Ordnung:

$$A(P) = \frac{A_0}{1 + \omega_g(R_2C_2 + R_1C_2 + R_1C_2(1 - A_0))P + \omega_g^2 R_1R_2C_1C_2P^2}$$

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 zweiter 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

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

Dabei sind immernur die positiven, reellen Lösungen zu verwenden. Schließlich gibt es in der Realität keinen negativen Widerstand, leider.

Bestimmung der Bauteilwerte

$Q_P = 0,707$ und $\Omega_P = 1$, $A_0 = 48$, $\omega_g = 100\text{Hz}$ 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 aufgebaut werden kann.

In den Berechnungen viel 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 3.90µF und wird ebenfalls auf 82nF gesetzt.

$$R_1 = \frac{1}{2 \cdot 100Hz \cdot 0,707} \cdot \frac{82nF \pm \sqrt{82nF^2 - 4 \cdot 0.707^2 \cdot 82nF(82nF + 82nF(1 - 48))}}{82nF(82nF - 82nF(1 - 48))}$$

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

$$R_2 = \frac{1}{2 \cdot 100Hz \cdot 0,707} \cdot \frac{82nF \pm \sqrt{82nF^2 - 4 \cdot 0.707^2 \cdot 82nF(82nF + 82nF(1 - 48))}}{82nF^2}$$

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

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

$$C_1 = 82nF$$

$$C_2 = 82nF$$

$$R_1 = 2579\Omega$$

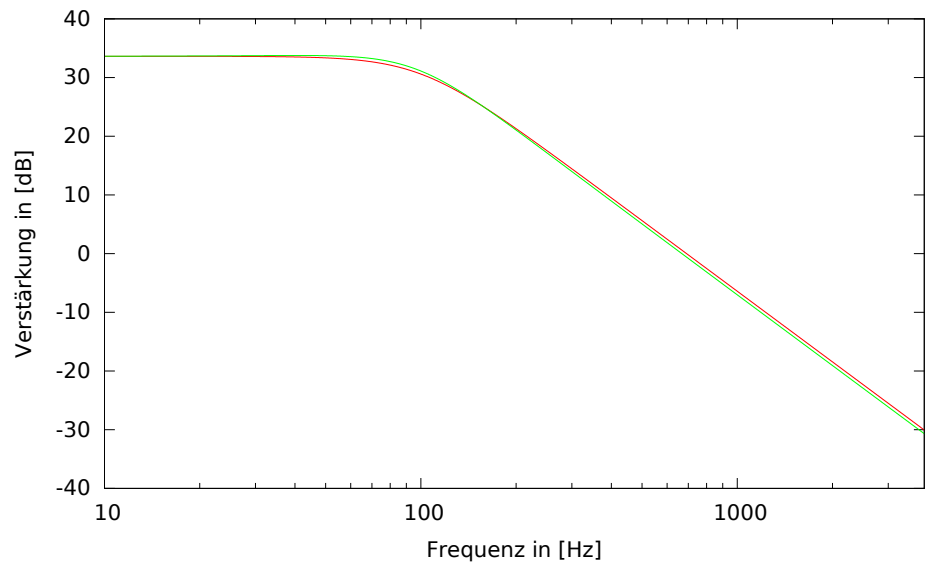
$$R_2 = 146079\Omega$$

In der Folgenden Abbildung ist das Ergebniss 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 unseres Eingangssignals, wird das bereits verstärkte Signal also um 63,7 dB gedämpft. Gefordert waren hier 55,9dB, so das der Filter den geforderten wert übersteigt, wass an der niedrigeren Grenzfrequenz von 100Hz statt 108,9Hz liegt.

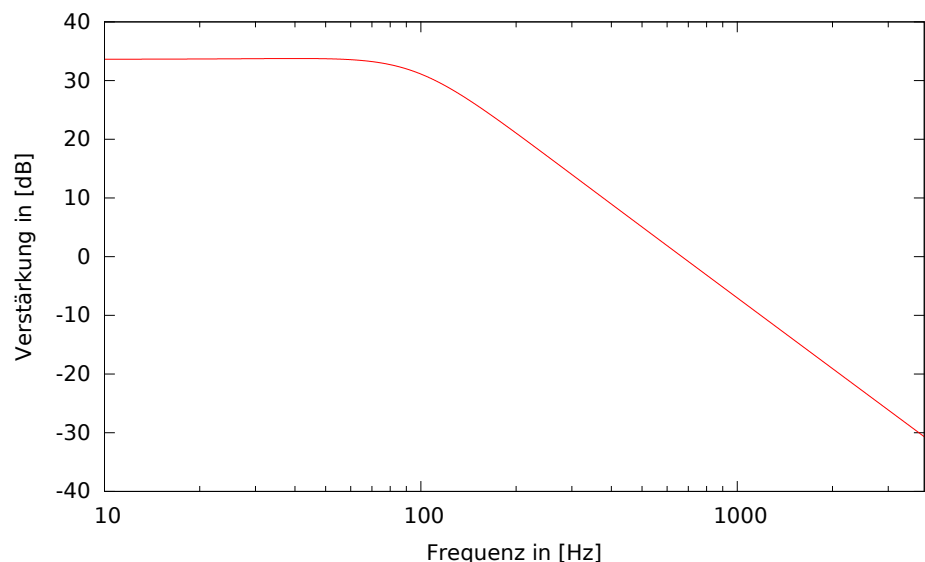
Abbildung 4.18: Frequenzgang
des berechneten Filters



Leider kann ein solcher Filter nur mit erheblichen Aufwänden gebaut werden, da es keine fertigen Widerstände in den Größen 2579Ω und 146079Ω 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 Realbauteilwerten 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 immernoch bei exakt 33,6dB. Bei 3,9 kHz im Diagramm gut zu erkennen wird trotz der höheren eine höhere Dämpfung als vorher erreicht. Diese liegt bei 33,7dB, daran kann man erkennen dass es sich nicht mehr um einen idealen Butterworthfilter handelt.

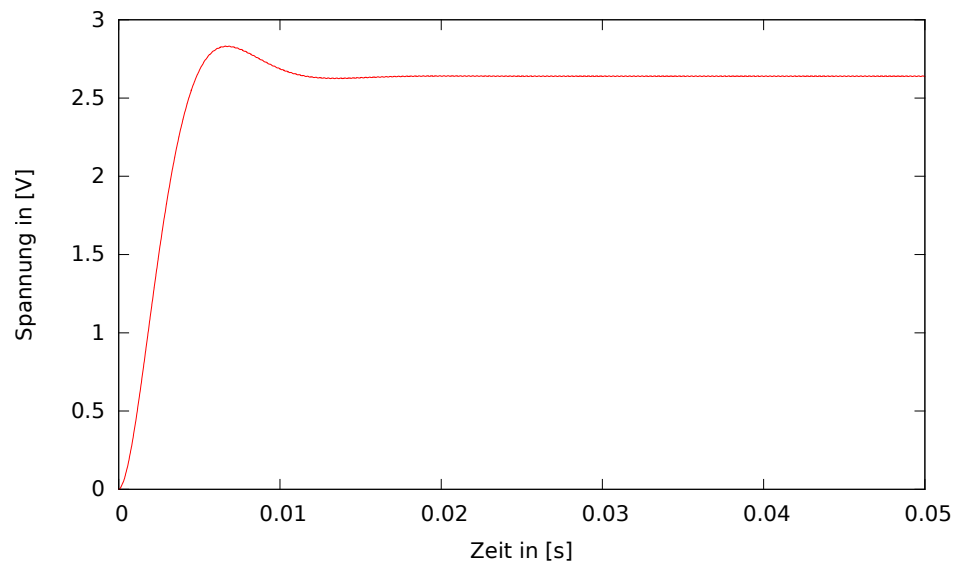
Abbildung 4.19: Frequenzgang
des berechneten Filters mit
finalen Bauteilwerten



Im der folgenden Abbildung [4.20] ist die Antwort des Filters auf ein Rechtecksignal mit 3,9kHz, einem Tastverhältnis von 0,5 und einer Amplitude von 50mV. Das Überspringen im Bereich von 7ms ist charakteristisch für den Butterworthfilter und wirkt sich negativ auf die Messung

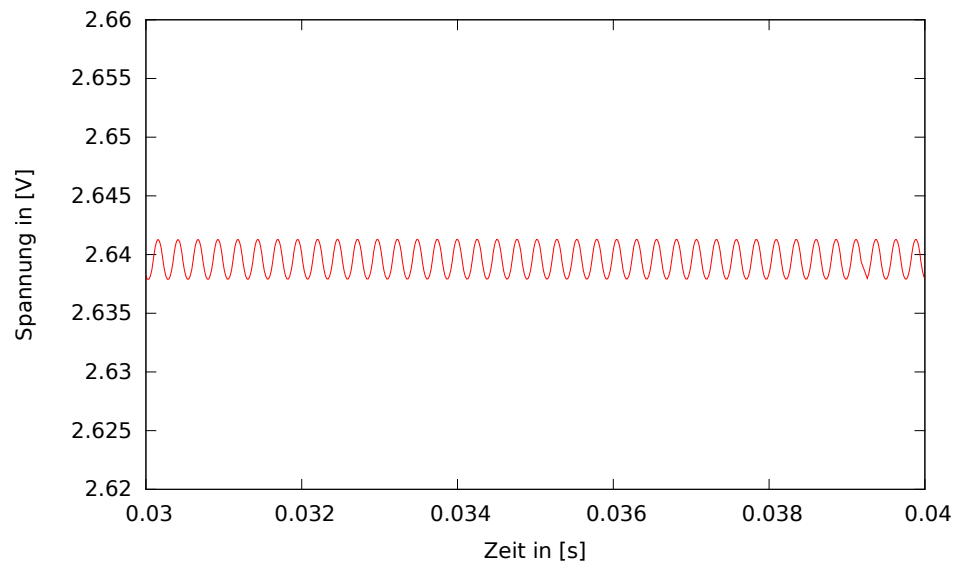
des Stromes aus. Allerdings werden solch große Sprünge in der Praxis nicht auftreten werden, da der Strom durch die große Induktivität des Motors nur langsam ansteigt.

Abbildung 4.20: Sprungantwort des Filters



Die in Abbildung [4.21] gut zu 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 rührt daher dass LT-Spice die steig und fall Zeiten in den low-Bereich des Rechteck signals legt, wodurch der Mittelwert des Signals bei 2,64V liegt.

Abbildung 4.21: Restwelligkeit des Filters



4.7 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 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.

4.7.1 Stromverbrauch der Komponenten

In diesem Abschnitt soll eine Abschätzung des Stromverbrauches 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 leider 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 ein Servomotor keinen konstanten Stromverbrauch. Der Stromfluss wird immer wieder unterbrochen, so dass 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 unsern am nächsten kommt ist der "Graupner 4421" mit folgenden Daten:

Technische Daten "Graupner 4421" [Gmb]:

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

Technische Daten des verwendeten Servomotors [rMGCK]:

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

Der Stromverbrauch des normalen Pandaboards (ohne ES) beträgt ca. 800mA [Bar]. Nähere Angaben zu Stromverbrauch des normalen Pandaboards (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 [For] sieht pro Port eine maximale Stromentnahme von 500mA vor. Da das Pandaboard über 2 USB ports verfügt liegt der maximale zusätzliche Verbrauch bei 1A Sodas hier ein gesamt Verbrauch von 2A veranschlagt wird.

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 Regelwerk konform 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 2 Frontscheinwerfer und Rückleuchten integriert. So dass sich eine Anzahl von 9 LED-Modulen ergibt. Der maximale durch die LEDs verursachte Verbrauch liegt somit bei 540mA.

Bremslichter???

Microkontroller

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 Distanz Sensoren liegt laut Datenblatt [Sha] bei 50mA, da 2 dieser Sensoren verbaut werden ergeben sich 100mA.

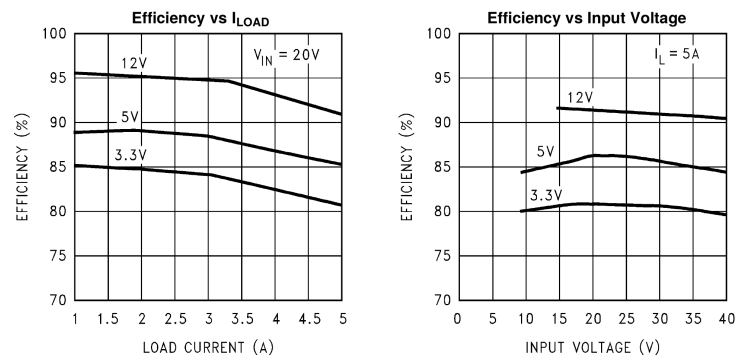
Sonstige Peripherie

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

4.7.2 Auswahl des Reglers

Der Gesamtstromverbrauch der Komponenten beträgt 4.740mA. Ein Linearregler hier nicht mehr praktikabel, da dieser bei einer Akkuspannung von 14,4V und 3.740mA fast 45 Watt Leistung in Wärme umwandeln würde. Eine gute Wahl für diesen Einsatzzweck 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 [4.22].

Abbildung 4.22: Regulator
Wirkungsgrad



Ausgehend von ca. 24 Watt Leistungsaufnahme ($4,8A * 5V$) und einem minimalen Wirkungsgrad von 80% ergibt sich damit eine überschauliche Verlustleistung von 6 Watt.

4.8 Software

Die Software besteht im Grunde aus zwei Teilen, zum einem der Firmware auf dem μ Controller zum anderen aus der Software auf dem NUC, welche die Daten vom μ Controller ausliest und über ROS publiziert. In den Folgenden Abschnitten werden Erst die Beiden Softwareteile erläutert, dann wird das Übertragungsprotokoll veranschaulicht.

4.8.1 Software auf dem μ Controller

Die Software auf dem μ Controller ist vollständig in C++ geschrieben. Eine vollständige Dokumentation der Software ist als Doxygen Dokument verfügbar.

4.8.2 Client Programm auf der Recheneinheit

Das Client Programm im folgenden SerialNode genannt Im erten Schritt wurde hier ein Python Programm genutzt, welches jedoch einen Nachteil mit sich bringt. Da das ansprechen der seriellen Schnittstelle unter pyserial sehr hohe CPU-Last mit sich bringt. Da die so verschwendete 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 verbessert. Während die Python Implementierung einen CPU-Kern zu 100% auslastete liegt die C++ Implementierung im unteren einstelligen Bereich.

4.8.3 Übertragungsprotokoll

Da die Übertragung der Daten via ROS-Serial im ersten Prototypen zu vielen Problemen geführt hat, wurde ien neues Übertragungsprotokoll entwickelt. Dabei wurde auf Fehlertolleranz 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 den Abbildungen [4.23] und [4.24] zu sehen.

Abbildung 4.23: Lese Daten

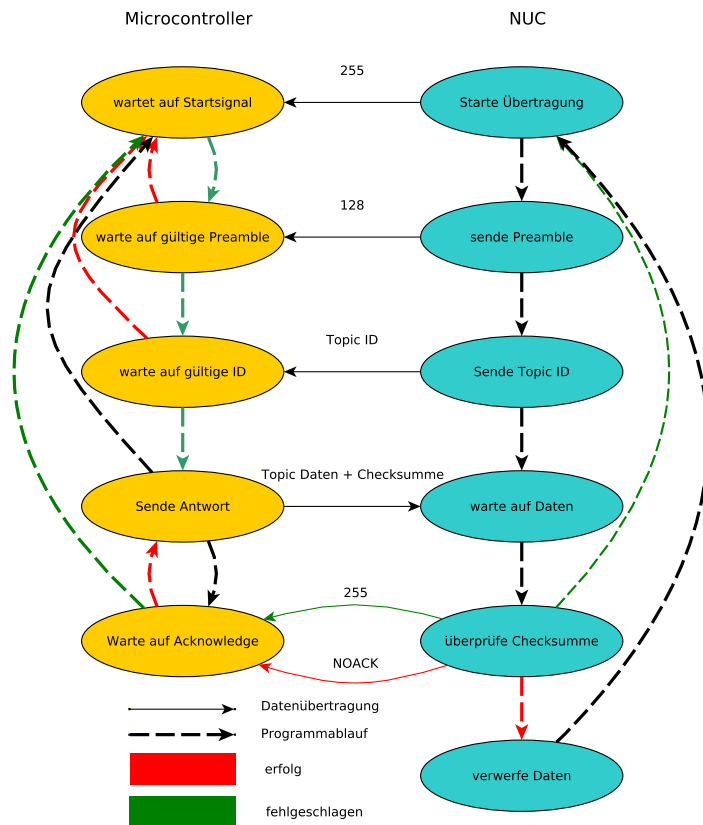
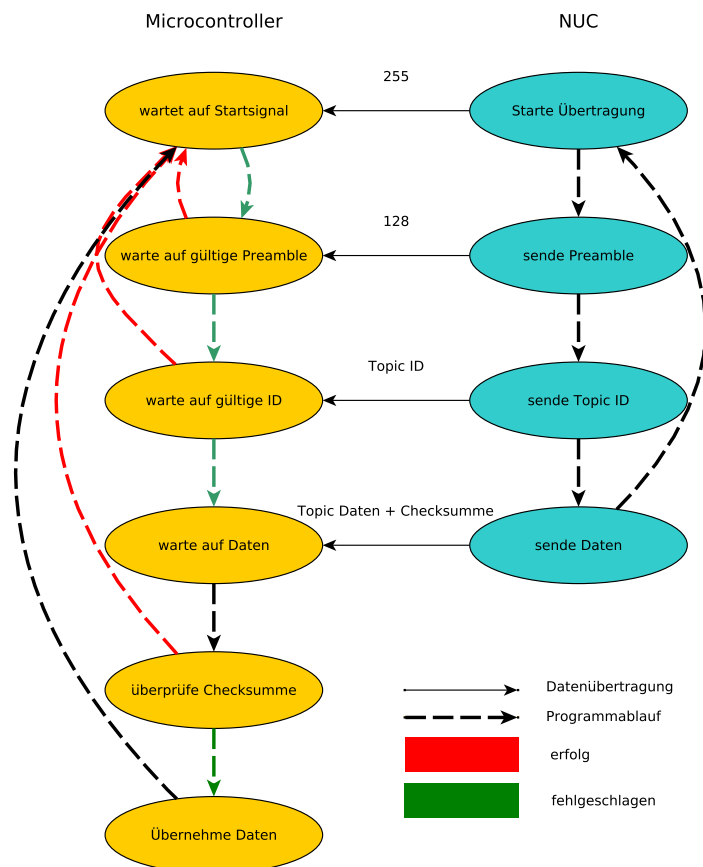


Abbildung 4.24: Schreibe Daten



5.1 Stormmessung

Abbildung 5.1: Spannung am Shunt

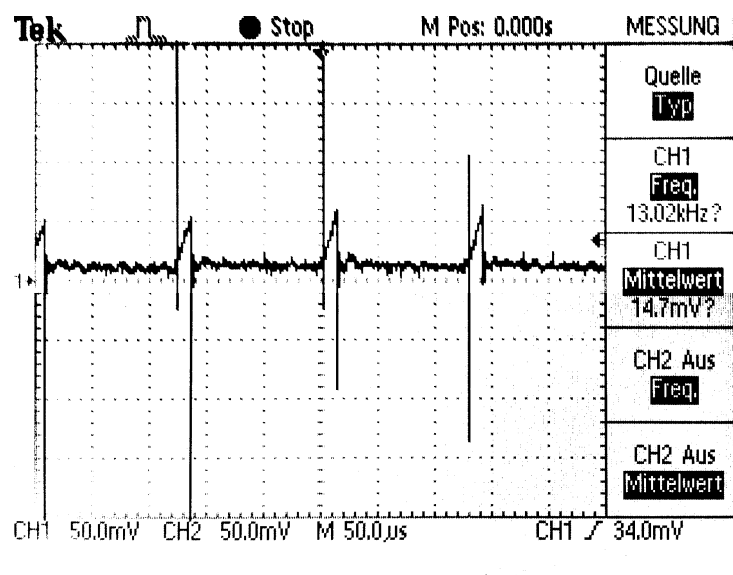
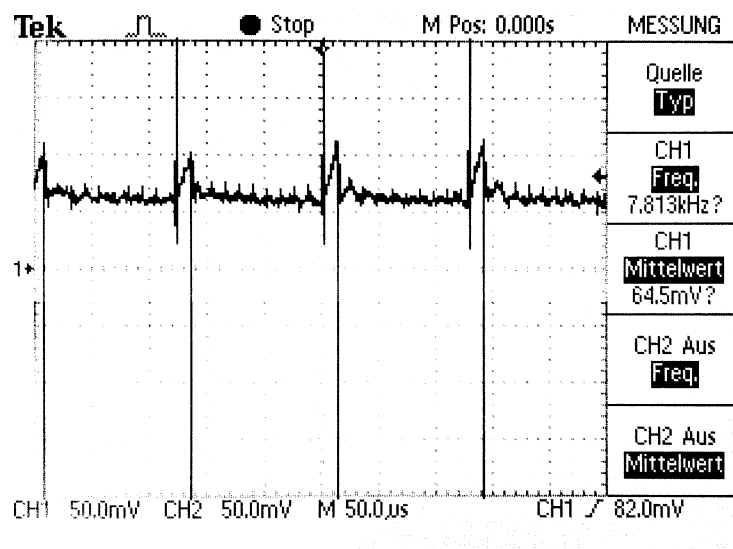


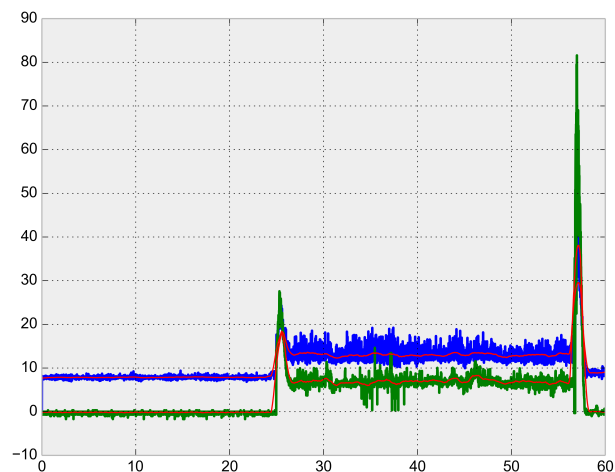
Abbildung 5.2: Spannung nach dem Filter



5.2 Stromverbrauch

Der Stromverbrauch des Autos 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. 5.3 zeigt den Verlauf des Stromes während folgendem Szenario: Bis ca 25s steht das Auto, sämtliche Software ist dabei auf dem Auto aktiv. Ab 25s beschleunigt das Auto auf $1,3 \frac{m}{s}$ und verährt 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 Autos darstellt. Die roten Linien stellen einen gleitenden Mittelwert aus den letzten 200 Messwerten dar. Gut zu erkennen ist, dass der Stromverbrauch des Autos im Stand unter 10 Watt liegt. Der Mittelwert des Verbrauches im Stand beträgt 7,9 Watt, während das Auto in der Fahrt knapp 13 Watt an Leistung aufnimmt. Nur während des Beschleunigungsvorganges mehr Leistung. Fährt das Auto 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 5.3: Salle-Key Tiefpass mit Shunt



5.3 Infrarotsensoren

5.4 Inertialsensor

5.5 Zeitverhalten der seriellen Verbindung

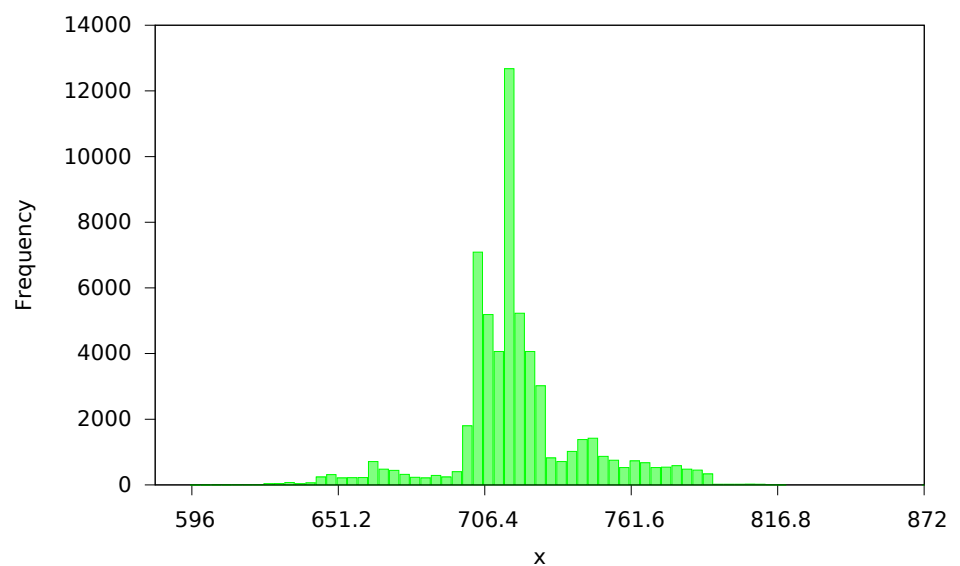
Um eine Aussage über das Alter eines Messwertes zu machen.

[Atm] adc Wandlung=13Takte Prescaler=128 bei 16MHz=125000kHz
=104uS pro Messung

Bytedauer = 11Bit (1Start+8Daten+2Stopp) 500000Baud -> 22us Byt-
dauer -> Preamble=5Bytes (4x255+ID) = 110us

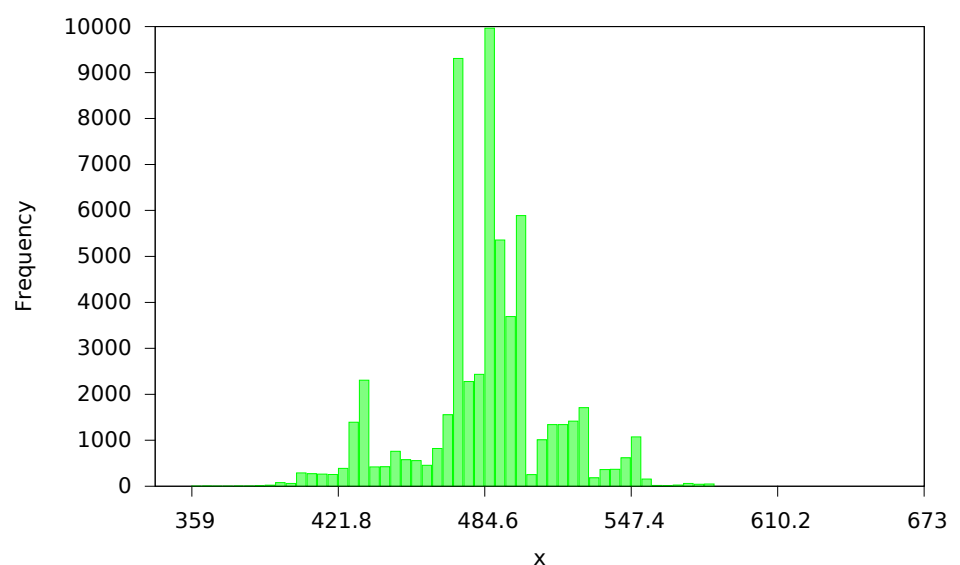
5.5.1 VoltageCurrent

25.0319430503 718.591761425



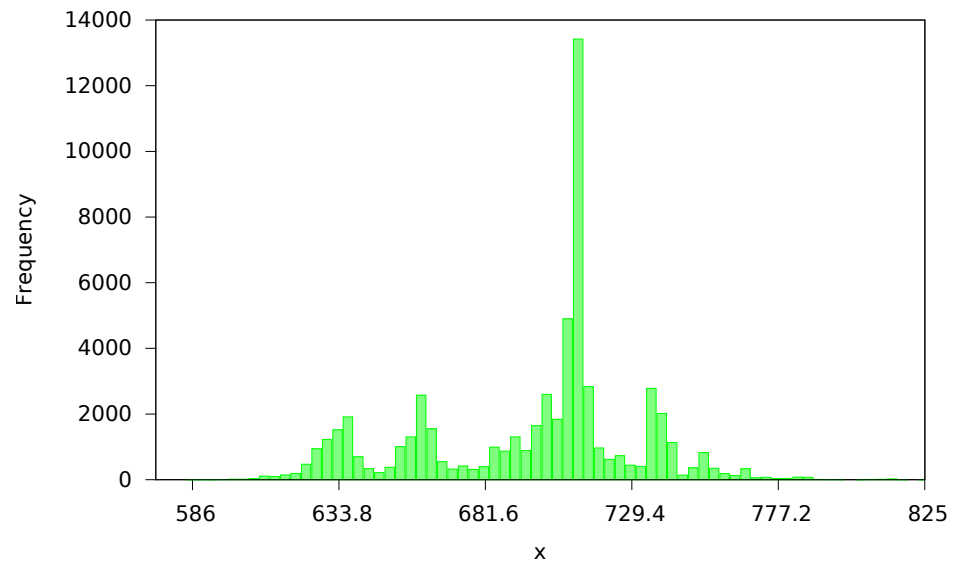
5.5.2 Distance

28.1616619788 484.734158193



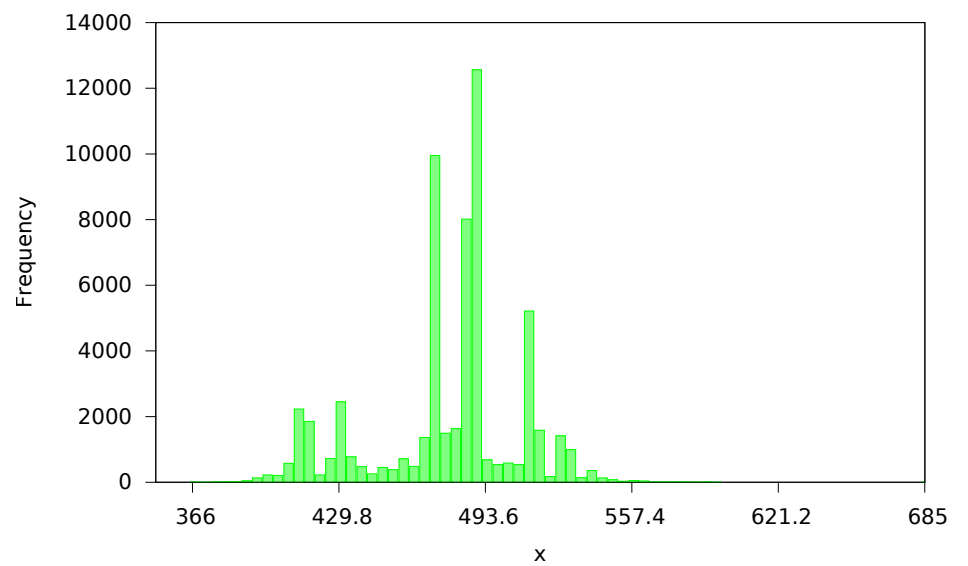
5.5.3 Infrared

35.0709303606 696.91743058



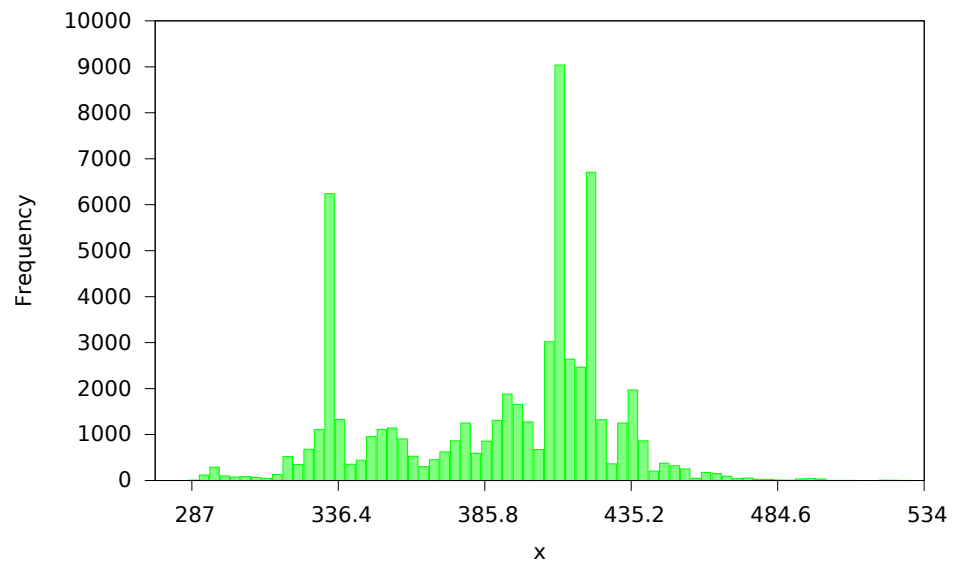
5.5.4 uCTime

31.2060346589 477.761007865



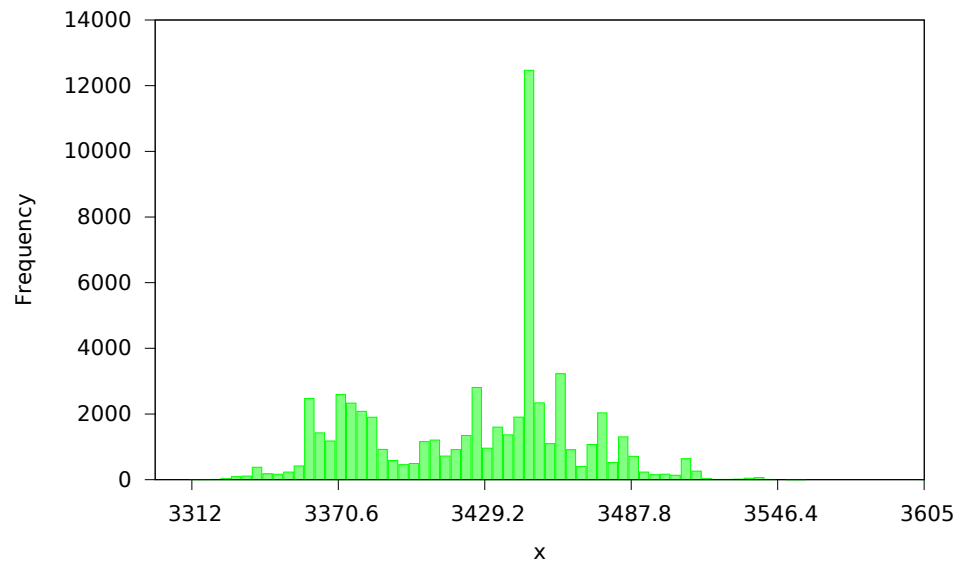
5.5.5 Motor

38.0093527069 391.111706657



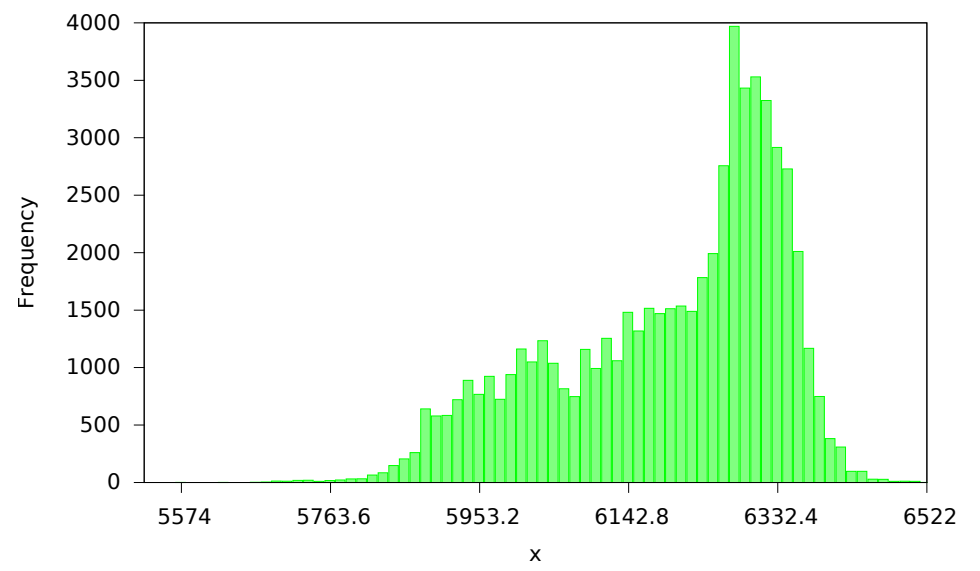
5.5.6 IMU

40.8272894054 3425.96398337



5.5.7 Gesamt

142.365993405 6195.08004809



6

Fazit



Ausblick

Abbildungsverzeichnis

1.1	Möglicher Rundkurs [feFTBb]	5
1.2	Möglicher Rundkurs [feFTBb]	5
3.1	Konzept	9
4.1	Vierquadrantensteller	12
4.2	Slow decay, diode recirculation, high-side PWM	13
4.3	Slow decay, SR active, high-side PWM	13
4.4	Slow decay, SR active, low-side PWM	13
4.5	Fast decay, diode recirculation	14
4.6	Fast decay, SR active, full four-quadrant control	14
4.7	Kaskadierung der LEDs	15
4.8	Codierung des LED Signals	15
4.9	Prinzip der Lasertriangulation	16
4.10	Ausgangssignal GP2D120	17
4.11	Ausgangssignal GP2D120 entstört	17
4.12	Aufbau eines Gleichstrommotors	18
4.13	Schmitt-Trigger Schaltung	18
4.14	Salle-Key Tiefpass mit Shunt	20
4.15	Spannung am Shunt + PWM	21
4.16	Tiefpass Toleranzfeld	22
4.17	Polstellen-Nullstellendiagramm, Butterworth 2. Ordnung .	23
4.18	Frequenzgang des berechneten Filters	26
4.19	Frequenzgang des berechneten Filters mit finalen Bauteil- werten	26
4.20	Sprungantwort des Filters	27
4.21	Restwelligkeit des Filters	27
4.22	Regulator Wirkungsgrad	30
4.23	Lese Daten	31

4.24	Schreibe Daten	31
5.1	Spannung am Shunt	32
5.2	Spannung nach dem Filter	32
5.3	Salle-Key Tiefpass mit Shunt	33

Literaturverzeichnis

- [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. Online; accessed 30-Juni-2014.
- [Cor] Intel Corporation. Intel nuc board d34010wyb. http://downloadmirror.intel.com/23090/eng/D54250WYB_D34010WYB_TechProdSpec06.pdf. Online; accessed 20-September-2014.
- [feFTBa] Lehrstuhl fuer elektronische Fahrzeugsystem TU Braunschweig. Carolo-cup. <https://wiki.ifr.ing.tu-bs.de/carolocup/carolo-cup>. Online; accessed 20-September-2014.
- [feFTBb] Lehrstuhl fuer elektronische Fahrzeugsystem TU Braunschweig. Carolo-cup regelwerk 2015. <https://wiki.ifr.ing.tu-bs.de/carolocup/system/files/Hauptwettbewerb2015.pdf>. Online; accessed 20-September-2014.
- [For] USB Implementers Forum. Usb 2.0 specification. http://www.usb.org/developers/docs/usb20_docs/#usb20adopters. Online; accessed 20-September-2014.
- [Gmb] Graupner/SJ GmbH. Graupner 4421. http://www.graupner.de/fileadmin/downloadcenter/servoliste/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.

- [Kru00] Gerhard Krucker. Elektronische signalverarbeitung. 2000.
- [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.
- [Oma] Omappedia. Pandaboard faq. 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.
- [rMGCK] robbe Modellsport GmbH Co. KG. Iq-620cmg coreless. <https://www.hype-rc.de/deu/shop/product/080-620cmg/servo-iq-620cmg-coreless.html>. Online; accessed 20-September-2014.
- [Sha] Sharp. Datenblatt gp2d120. 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, Juni 2014. Online; accessed 17-Juni-2014.
- [Wik] Wikipedia. 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 26. September 2014

Julian-B. Scholle