

Entwicklung und Test einer eingebetteten Elektronik für einen innovativen Schaltaktor

Malte Breitenbach, Johannes Faupel, Jonas Tautz, Johanna Vetter



**TECHNISCHE
UNIVERSITÄT
DARMSTADT**

Institut für Mechatronische Systeme im
Maschinenbau
Prof. Dr.-Ing. Stephan Rinderknecht

Zusammenfassung

Hier könnte Ihr Abstract stehen.



Inhaltsverzeichnis

1 Einleitung	5
1.1 Motivation	5
1.2 Ziele der Arbeit	5
1.3 Anforderungsliste	5
1.4 Aufbau der Arbeit	5
2 Aufbau des Prüfstands und Grundlagen	7
2.1 Grundlagen des Prüfstandes	7
2.1.1 Getriebe allgemein	7
2.1.2 Getriebe des Prüfstands	7
2.1.3 Aktor	8
2.1.4 Aktoransteuerung	9
2.1.5 Autobox	9
2.1.6 Positionsmessung und -regelung	9
2.2 Grundlagen Löten	10
2.3 Mikrocontroller	10
2.4 Eingebettete Systeme und Smart Actuators	11
2.5 Controller Area Network (CAN)	12
2.6 Pulsweitenmodulation	13
3 Vorgehensweise und Testverfahren	15
3.1 Entwicklungsmodell	15
3.1.1 Das V-Modell	15
3.1.2 Konkretes Vorgehen mit inkrementellem Entwicklungsansatz	17
4 Auswahl elektronischer Komponenten und Verschaltung	19
4.1 Anforderungen an die Komponenten	19
4.1.1 Passive Bauteile	19
4.1.2 Aktive Bauteile	19
4.2 Komponentenauswahl	20
4.2.1 Mikrocontroller	20
4.2.2 CAN-Transciever	22
4.2.3 Spannungsversorgung	23
4.3 Platinenanschluss	24
4.4 H-Brücke	24
4.5 Sensorik	25
5 Platinenentwurf	27
5.1 Elektronikgehäuse und Anforderungen an die Platinendimensionierung	27
5.2 Bauteilplatzierung	28
5.3 Dimensionierung der Leiterbahnen	29
5.3.1 Anwendung auf realer Platine	31
5.4 Routen der Platine	31
5.4.1 Design-Regeln und EMV	31
5.4.2 Leiterbahnverlegung des Entwicklungsboards	33
6 Software	35
7 Analyse und Performance	37

7.1 Beurteilung der Anforderungserfüllung	37
7.2 Kostenaufstellung	37

8 Fazit und Ausblick	39
-----------------------------	-----------

Abbildungsverzeichnis	41
------------------------------	-----------

1 Einleitung

1.1 Motivation

1.2 Ziele der Arbeit

Auch das Institut für Mechatronische Systeme (IMS) der TU Darmstadt nimmt sich dieser Aufgabe an und arbeitet im Rahmen des Projektes Speed4E, dass einen elektrifizierten Antriebsstrang mit Peak-Antriebsdrehzahlen von bis zu $50.000/min$ zum Ziel hat, an einem innovativem Schaltaktor. Im Verlauf vorheriger Arbeiten wurde bereits ein Tauchspulenaktor ausgelegt sowie eine Elektronik für ihn entwickelt. Das Ziel dieser Arbeit ist es nun, die bisherigen Funktionen auf einen Mikrocontroller zu implementieren sowie eine eingebettete Elektronik zu entwerfen, die den Aktor zu einem Smart Actuator transformiert. Des Weiteren sollen Sicherheits- und Überwachungsfunktionen für den Aktor entwickelt werden und Statusmeldungen per CAN gesendet werden.

1.3 Anforderungsliste

Um das übergeordnete Ziel weiter zu spezifizieren wurde zunächst eine Anforderungsliste erstellt. In dieser sind alle Forderungen an das Endprodukt gesammelt, sie dient damit als Basis und Referenz für die Produktentwicklung. Die Liste ist hierbei dynamisch, das heißt sie kann im Verlauf des Entwicklungsprozesses verändert oder ergänzt werden. Die formulierten Anforderungen werden schließlich noch nach Priorität kategorisiert und einer der vier folgenden Anforderungsarten [2013a]: Festforderungen (FF) sind unter allen Umständen einzuhalten. Eine Erfüllung ist für eine erfolgreiche Lösung notwendig. Bereichsforderungen (BF) geben einen Toleranzbereich an, innerhalb dessen sich der schlussendlich erreichte Wert befinden muss. Zielforderungen (ZF) geben an, welcher Wert (auch im Hinblick auf spätere Entwicklungen) angestrebt wird. Wünsche (W) sollten nach Möglichkeit erfüllt werden, sind aber keine Voraussetzung.

Relevanz	Anforderung	Erläuterung
FF	Benutzerfreundliche Kommunikation durch CAN Schnittstelle	Empfang von Befehlen, Senden von Statusmeldungen
FF	Nichtflüchtige Kalibrierung	Eine Kalibrierung ist nur einmalig und zur Rekalibrierung notwendig
BF	Schaltzeit	< 100 ms (Latenz zwischen Senden des Befehls und vollständig ausgeführtem Gangwechsel)
FF	Selbstständige Fehlererkennung	Überstrom, Temperatur, Eingangsspannung (OVP/UVP), Dekalibrierung
FF	Schnittstellen	CAN, 8-12VDC Versorgung (max XA), Programmerschnittstelle (für Updates & Bugfixes)
W	Wartbarkeit	Sicherung wechseln im eingebauten Zustand
BF	kompakte Baugröße	
BF	Effizienz (gemittelt über einen Schaltvorgang)	elektrischer Wirkungsgrad > 90 %
FF	Temperaturbeständigkeit	bis 105°C
BF	Aktorüberschwingen	Toleriert, solange kein unbeabsichtigter Gangwechsel
W	Schaltgabelkraft am Anschlag	möglichst gering
FF	Standby	Standbyleistungsaufnahme < 5W

Tabelle 1.1: Anforderungsliste

1.4 Aufbau der Arbeit

Nachdem in der oben stehenden Anforderungsliste die Ziele der vorliegenden Arbeit definiert wurden, soll nun das weitere Vorgehen zum Erreichen der Zielsetzung erläutert werden. Zunächst werden in Kapitel 2 der Stand des Prüfstands

vor Beginn des ADPs sowie wichtige Grundlagen als Basis für die weitere Bearbeitung und zum besseren Verständnis dargelegt. Kapitel 3 stellt das allgemeine methodische Vorgehen für das Entwerfen der eingebetteten Elektronik dar. In Kapitel 4 werden die verschiedenen Komponenten, die für die Elektronik benötigt werden, und ihre Funktionen beschrieben. Es wird erklärt, wie die einzelnen Komponenten miteinander interagieren und wie ihre Schaltung funktioniert. In Kapitel 5 soll schließlich das endgültige Platinendesign und inklusive schematischem Aufbau vorgestellt werden. Darauf folgt in Kapitel 6 die Vorstellung des Programms.

2 Aufbau des Prüfstands und Grundlagen

2.1 Grundlagen des Prüfstandes

In diesem Kapitel wird der verwendete Schaltaktorikprüfstand des IMS vorgestellt, an dem die Entwicklung des Smart Actuators stattgefunden hat. Beschrieben wird der Aufbau des Prüfstands, wie er vor den Änderungen im Rahmen dieses Projektes vorlag. Die Konstruktion des Prüfstandes erfolgte in vorangegangenen Arbeiten und wurde seitdem stetig weiterentwickelt. An ihm werden Schaltaktoriksysteme für Fahrzeugantriebe untersucht. Abbildung 2.1 zeigt die in dieser Arbeit verwendeten Subsysteme des Prüfstandes. Es erfolgt zunächst die Vorstellung des mechanischen Aufbaus, woraufhin der elektronische Aufbau anschließt.

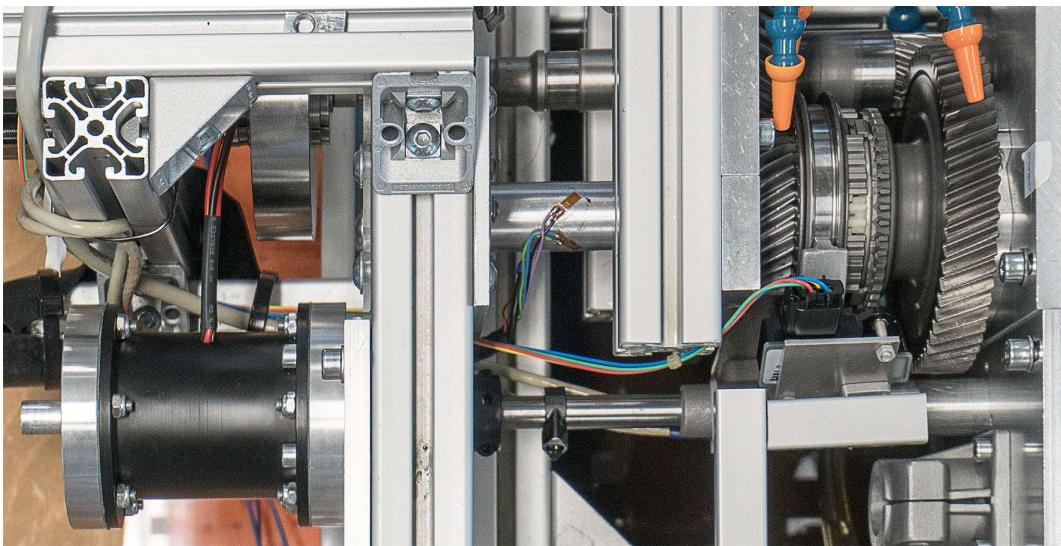


Abbildung 2.1: Prüfstand [adp]

2.1.1 Getriebe allgemein

Ein Getriebe ist im Automobil dafür zuständig, die Drehzahl des Motors in ein Drehmoment umzuwandeln, welches die Räder antreibt. Da Motoren nur einen kleinen Bereich von Motordrehzahlen abdecken werden mehrstufige Getriebe verwendet, die verschiedene Raddrehzahlen durch unterschiedliche Übersetzungsverhältnisse bereitstellen können. Das Einstellen des jeweiligen Ganges kann dabei per Hand (Handschaltgetriebe) oder automatisiert über einen Aktor (Schaltaktorik) erfolgen. Im Fahrzeuggetriebe, beispielhaft dargestellt in 2.2 ist die Eingangswelle, welche durch den Motor angetrieben wird, über eine Zahnradverbindung fest mit der Vorgelegewelle verbunden. Auf ihr sind noch weitere fest fixierte Zahnräder angebracht, deren Anzahl mit den verfügbaren Gängen übereinstimmt. Diese greifen jeweils in Losräder auf der Abtriebswelle. Um einen bestimmten Gang einzulegen muss nun das jeweilige Losrad für den Moment fest mit der Abtriebswelle verbunden werden, sodass nur diese Zahnverbindung ein Drehmoment überträgt. Dies geschieht über eine formschlüssige Verbindung mit einer Schaltmuffe, die über die durch den Aktor angetriebene Schaltgabel in Position gebracht wird.

2.1.2 Getriebe des Prüfstands

Der Prüfstand besitzt zwei Gänge, in die über eine Schaltgabel geschaltet werden kann. Eine Bewegung der Schaltgabel nach links legt Gang 1 über eine mechanische Synchronisierung ein, während mit Hilfe einer Bewegung nach rechts die Schaltung des Ganges 2 durch eine Klauenkupplung erfolgt. Somit können sowohl Schaltaktoriksysteme mit als auch ohne Synchronring untersucht werden. Die Bewegung der Schaltgabel wird durch einen Linearaktor ermöglicht, welcher



Abbildung 2.2: Fahrzeuggetriebe

von außen an das Item-Profil verschraubt ist. In diesem ist eine Tauchspule verbaut, die die benötigten Kräfte auf die Läuferstange aufbringt. Über eine starre Wellenkupplung sind Läuferstange und Schaltgabel miteinander verbunden, wodurch die Kräfte auf die Schaltgabel übertragen werden und Schaltvorgänge ermöglicht werden.

2.1.3 Aktor

Der momentan in dem Prüfstand verbaute Tauchspulenaktor wurde von Oliver Hahn im Rahmen seiner Bachelorthesis entwickelt und konstruiert. Er übernimmt im automatisierten Schaltvorgang die Aufgabe, die Schaltgabel über die Schaltstange translatorisch zu verschieben, welche ursprünglich mit einem Schalthebel per Hand ausgeführt wurde. Die Übertragung des Schaltbefehls an den Getriebeaktor erfolgt dabei durch ein elektrisches Signal (*shift by wire*). Sein Querschnitt ist in folgender Abbildung schematisch dargestellt.

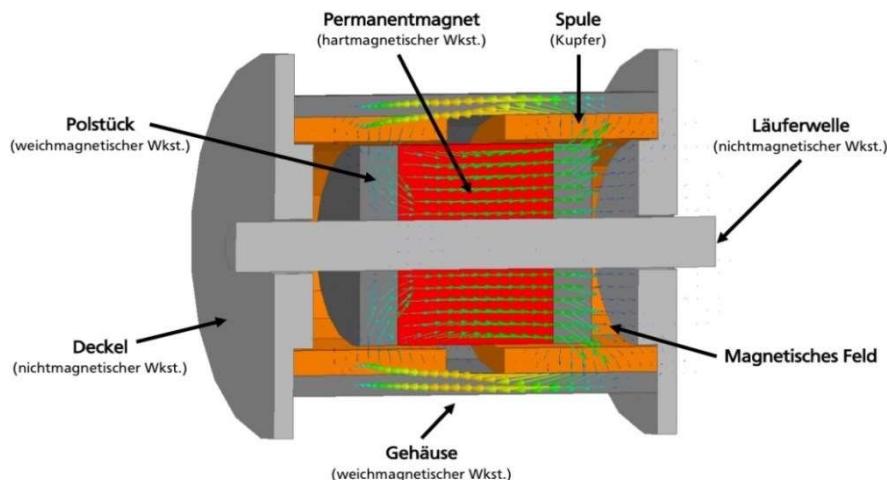


Abbildung 2.3: Querschnitt Tauchspulenaktor [Hahn2018]

Sein Aufbau ist zylindrisch und kann in den ortsfesten Stator und den beweglichen Läufer unterteilt werden. Der Stator des Aktors besteht aus zwei in Reihe geschalteten Kupferspulen, welche fest in dem Gehäuse aus Weicheisen liegen und nach oben und unten mit Deckeln aus Aluminium fixiert werden. Der Läufer besteht aus einer nichtmagnetischen Läuferwelle, auf der sich fünf Permanentmagneten aus Neodym-Eisen-Bor befinden, welche mit Hilfe von zwei Polstücken aus Weicheisen axial auf der Läuferstange montiert sind. Werden nun die Kupferspulen von Strom durchflossen, so wirkt eine vom Magnetfeld der Permanentmagneten induzierte Lorentzkraft orthogonal auf sie. Diese Kraft ist abhängig von der Stromstärke I , der magnetischen Flussdichte der Permanentmagneten B und der vom Magnetfeld durchsetzten Leiterlänge l :

$$F = I \cdot l \cdot B \quad (2.1)$$

Da die Spulen jedoch fest im Gehäuse verbaut sind, wirkt eine entgegengerichtete Kraft auf die Permanentmagneten, die auf der axial verschiebbaren Läuferwelle lagern. Diese Kraft bewirkt dann eine translatorische Bewegung der Welle und somit auch der Schaltgabel. Die Richtung der translatorischen Bewegung kann dabei über die Richtung des in den Spulen fließenden Stromes, der Betrag der Kraft über den Betrag des Stromes eingestellt werden. Anhand von Abbildung 2.4 ist zu erkennen, dass die Kraft-Weg-Kennlinien des Aktor innerhalb des Nutzungsbereichs von -10 Millimeter bis +10 Millimeter bei verschiedenen Stromstärken annähernd linear verlaufen.

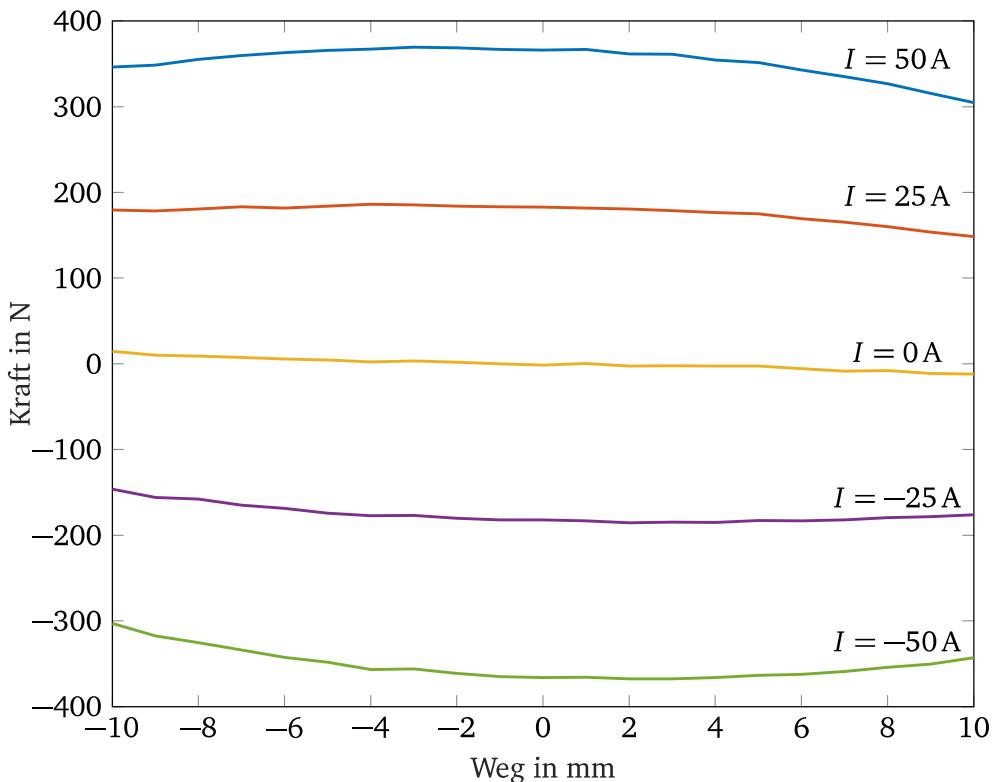


Abbildung 2.4: Kraft-Weg-Strom Kennlinien des Tauchspulenaktors [adp]

2.1.4 Aktoransteuerung

Zur Aktoransteuerung wird ein Arduino IBT2 Motortreiber verwendet, die Stromversorgung erfolgt über ein Manson SBC-2130 Battery Charger im Power Supply Mode. Der Motortreiber besteht aus zwei BTS7960 MOSFETs, die je nach Vorgabe des pulsdauermodulierten Signals (PWM-Frequenz) eine Spannung von bis zu 13,8V an den Aktor durchschalten.

2.1.5 Autobox

2.1.6 Positionsmeßung und -regelung

Die Position der Läuferstange wird über einen PLCD-25M Sensor gemessen, dessen berührungslose PLCD (permanent-magnetic linear contactless displacement) Technologie die magnetische Sättigung nutzt. Der weichmagnetische Kern

des Sensors ist über seine komplette Länge von einer Primärspule umgeben. Auf der Schaltgabel ist ein Permanentmagnet befestigt, welcher je nach Position zu einer lokalen Sättigung des weichmagnetischen Kerns führt. Über zwei Auswertungsspulen am Rand des Sensors kann nun die Position der Sättigungszone entlang der Sensorachse über eine induzierte Spannung bestimmt werden.

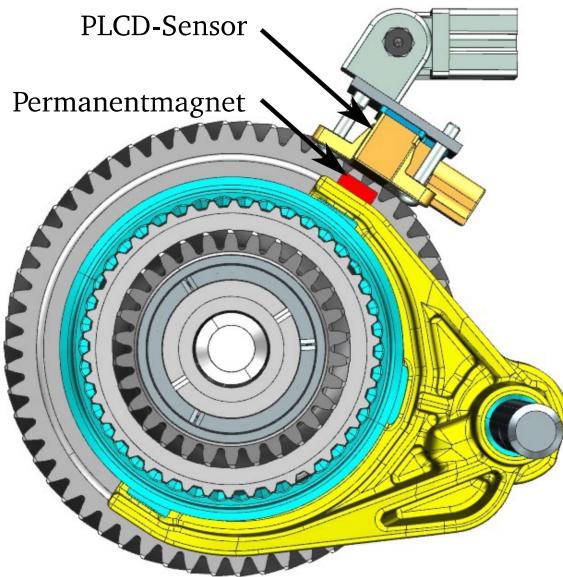


Abbildung 2.5: Einbauposition PLCD Sensor an der Schaltgabel [adp]

Der Sensor wird an die MicroAutoBox zwecks Spannungsversorgung sowie zur Durchführung der Kalibrierung angegeschlossen. Die Kalibrierung wurde in [messtechnik] entwickelt und erfolgt bisher bei jedem Start, damit es bei unbeabsichtigten Verstellungen des Sensors nicht zu verfälschten Messergebnissen kommt. Zu Beginn des Einfahrvorgangs der Schaltwalze wird dabei das Spannungssignal U_0 ausgelesen, welches der Ausgangslage von $x_0=0\text{mm}$ entspricht. Anschließend werden die Spannungssignale U_1 und U_2 an beiden Anschlägen gemessen. Die Wegdifferenz zwischen den beiden beträgt $x_{\text{ges}}=16,97\text{mm}$. Nun kann eine Geradengleichung zur Bestimmung der aktuellen Schaltgabelposition aufgestellt werden. U_a entspricht dabei immer dem aktuellen Ausgangssignal.

$$x = -\frac{-(U_a - U_0)}{U_1 - U_2} \cdot x_{\text{ges}} \quad (2.2)$$

Die bisherige Regelung der Schaltgabelposition wurde in einem vorangegangenem Advanced Design Project ausgelegt. Dazu wurden mithilfe des Ziegler-Nichols-Verfahren zunächst Parameter für einen PID-Regler ermittelt, welcher allerdings noch zu große Totzeiten und ein Überschwingen aufwies. In einer anschließenden iterativen Optimierung mit Störgrößenkompensation wurden die Regelparameter zu $KP = 2,6\%\text{V/mm}$, $KI = 20\%\text{V/s/mm}$ und $KD = 1,6\%\text{Vs/mm}$ bestimmt. Der Verlauf der Sprungantwort ist in nachstehender Abbildung (2.6) dargestellt.

2.2 Grundlagen Löten

Löten gehört zu den Fügeverfahren und bezeichnet das Verbinden zweier Metalle durch eine Metallegierung unter Einfluss von Wärme [NIeC {\\"o}ten]. Die Metallegierung wird dabei als Lot bezeichnet und hat eine geringere Schmelztemperatur (Liquidustemperatur) als die beiden zu verbindenden Metalle (Solidustemperatur). Durch das Löten entsteht eine feste, korrosionsbeständige sowie strom- und wärmeleitende Verbindung. Lötverfahren werden nach Arbeitstemperatur eingeteilt: es wird unterschieden in Weichlöten (bis 450°C), Hartlöten (450°C - 900°C) und Hochtemperaturlöten (über 900°C). Im Rahmen dieses Projektes wurde das Verfahren Weichlöten mit Lötkolben angewendet, um die elektrischen Bauteile auf der Platine anzubringen. Dazu konnten die vom Institut bereitgestellten Lötstationen und –materialien verwendet werden.

2.3 Mikrocontroller

Ein Mikrocontroller (MCU: *microcontroller unit*) ist ein hochintegrierter Halbleiterchip, der ein komplettes Mikrorechner-System enthält. Prozessoren, Speicher, Ein- und Ausgabegeräte sind somit auf einem kleinen Chip enthalten der zum Ziel hat, Steuerungs- und Kommunikationsaufgaben möglichst simpel mit wenig Bauelementen zu bearbeiten.

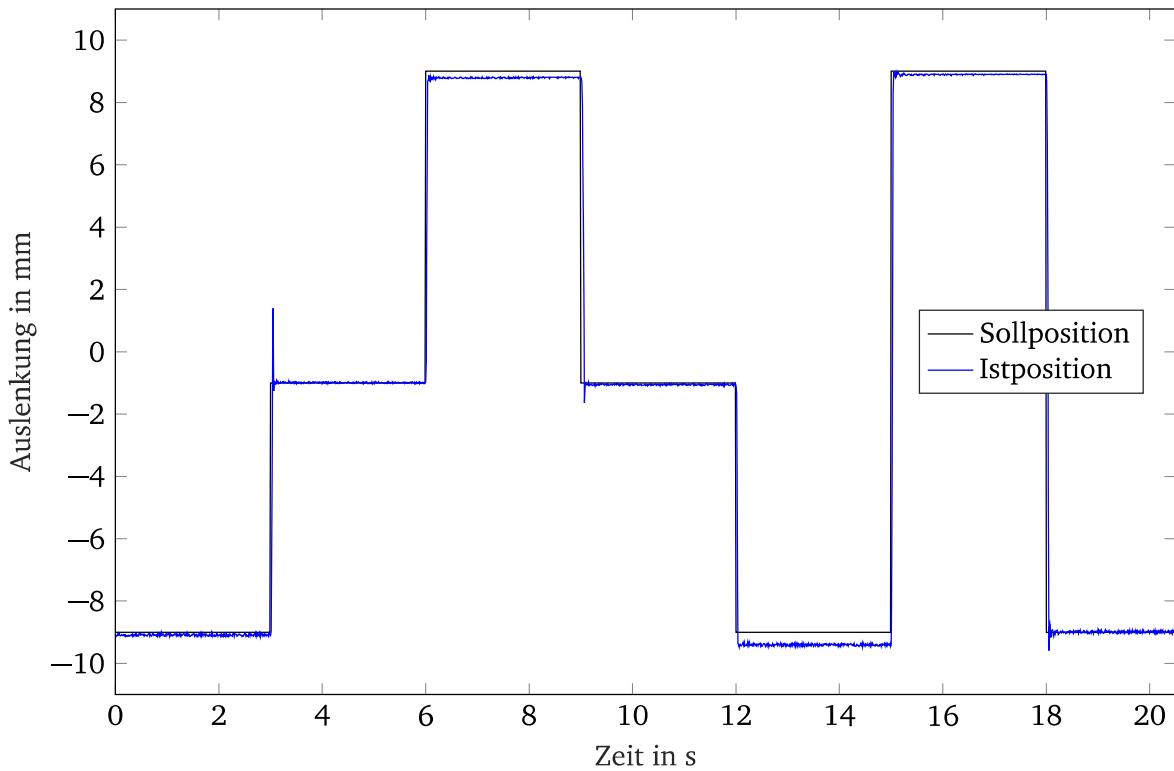


Abbildung 2.6: Sprungantwort Schaltgabelposition [adp]

In Abbildung 2.7 ist der typische Aufbau eines Mikrocontroller dargestellt. Die Schnittstellen zur Peripherie sind durch einen Betriebsspannungsanschluss, einen Takteingang, an den in der Regel ein Quarz angeschlossen wird, sowie die Portleitungen gegeben. Bei den Ports wird zwischen Eingangskanälen (**Input Port**), die digitale Signale lesen, und Ausgangskanälen (**Output Port**), die digitale Signale setzen beziehungsweise löschen, unterschieden. Des weiteren können I/O Pins digital oder analog sein, wobei analoge Signale mithilfe eines Analog/Digital Wandlers (AD-Wandler) zuerst in digital Signale umgewandelt werden müssen, damit der Mikrocontroller sie verarbeiten kann. Eine Spezialform von Eingängskanälen sind die Interrupt-Pins, die bei bestimmten Ereignissen Unterbrechungen des laufenden Programms verursachen, um temporär einen anderen Vorgang zu bearbeiten.

Intern sind die einzelnen Bausteine, die im folgenden kurz erläutert werden, über ein Bussystem verbunden. Der Prozessor (**CPU: central processing unit**) führt Berechnungen und logische Operationen durch. Der Taktgenerator gibt die Arbeitsfrequenz an, also wie schnell der CPU arbeiten soll. Der Arbeitsspeicher (**RAM: random access memory**) speichert temporär Daten, die aber spätestens nach dem Entfernen der Betriebsspannung gelöscht werden. Der Festspeicher (**ROM: read only memory**) behält seinen Speicherinhalt auch nach dem Entfernen der Betriebsspannung und enthält aufgrund dessen das Programm sowie Einstellungen und wichtige Daten. Im Vergleich zum RAM hat er eine langsame Schreibgeschwindigkeit. Der Timer hilft dabei, Anzahlen von Ereignissen zu zählen oder Zeitabstände zu messen, indem er Spannungswechsel an einem Eingangskanal zählt.

Die genaue Ausführung des Chips kann je nach Aufgabentyp variieren, sodass eine Vielzahl an verschiedenen Mikrocontrollern erhältlich ist. Diese unterscheiden sich meistens in der Größe des Speichers, in der Anzahl der Anschlüsse beziehungsweise Schnittstellen, in der Bitbreite, in den Taktraten sowie in der Bauform. Typischerweise werden Mikrocontroller in eingebetteten Systemen (*embedded systems*) verwendet, bei denen die Steuereinheit direkt im System selbst integriert ist. Übliche Anwendungen für Mikrocontroller sind Roboter, Handys, Temperaturregler oder Motorsteuerungen [**Brinkschulte**].

2.4 Eingebettete Systeme und Smart Actuators

Eingebettete Systeme werden definiert als Rechenmaschinen, die für den Anwender weitgehend unsichtbar in einem technischen System eingebettet sind und mit diesem in Wechselwirkung stehen [**Gessler2014**]. Der Rechner kann dabei für Regelungs-, Steuer- oder Überwachungsaufgaben und/ oder für Signal- und Datenverarbeitung zuständig sein. Der allgemeine Zweck ist es, die Stellglieder als Reaktion auf Eingangssignale, die von Sensoren oder manuell vorgegeben werden, zu steuern [**Broy2003**]. Eingebettete Systeme werden oft für speziell eine isolierte Aufgabe entwickelt und angepasst. Ein Aktor, beziehungsweise Aktuator (aus dem englischen: *actuator*) ist dafür da, die elektrischen Signale in eine

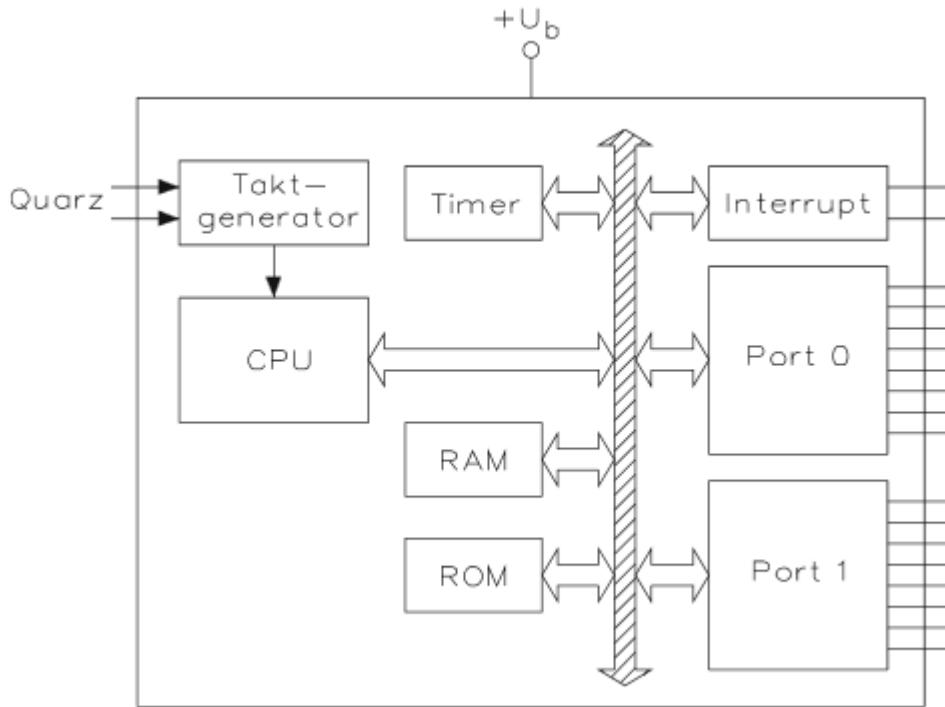


Abbildung 2.7: Blockschaltbild eines typischen Mikrocontrollers [Bernstein2015]

Bewegung oder Kraft umzusetzen und bildet somit das Stellglied. Ein intelligenter Aktor (*smart actuator*) ist definiert als das integrierte Stellglied inklusive aller Komponenten wie Motor, Steuerung, Sensorik und Kommunikationseinheit [**smartactuator**]. Das Konzept besteht darin, den Aktor um Informationsverarbeitungs- und Kommunikationsmöglichkeiten zu erweitern, der Aktor enthält dann die eingebettete Elektronik. Smart Actuators reduzieren beziehungsweise ersetzen die benötigte Interaktion mit einem Menschen oder externen Rechner und können somit effektiver und schneller arbeiten. Außerdem kann oft viel Bauraum eingespart werden, es gibt weniger Komponenten und deutlich weniger Verkabelung, womit der Einbau in ein Gesamtsystem leichter ist. Hervorzuheben sind weiterhin die verringerten Kosten, die Überwachung und Diagnose des eigenen Zustands sowie die verbesserte Präzision. Einbußen müssen Smart Actuators allerdings in Sachen Flexibilität, da sie oft für eine spezielle Aufgabe ausgelegt sind und ihr Verhalten stark vorgegeben ist [**smartaktor**].

2.5 Controller Area Network (CAN)

Um eine Kommunikation zwischen zwei Geräten (z.B. Sensoren, Aktoren und Steuergeräten) zu ermöglichen, muss ein System zur Datenübertragung vorhanden sein. Dabei hat sich bei vielen PKW- und NKW-Herstellern das Controller Area Network (CAN) durchgesetzt [Werner2014]. CAN ist ein von Bosch für den Automobilbau entwickeltes Bussystem, mit dem mehrere gleichberechtigte Teilnehmer verbunden werden und miteinander kommunizieren können [Woern2006]. Für ein CAN-Netzwerk werden zwei Leitungen CAN-High und CAN-Low benötigt, an denen alle zu verbindenden Teilnehmer parallel angeschlossen sind. Die beiden Leitungen werden an beiden Enden mit Wellenwiderständen von 120 Ohm verbunden. Der schematische Aufbau eines CAN-Buses mit zwei Teilnehmern ist in Abbildung 2.8 dargestellt.

Sollten mehrere Teilnehmer gleichzeitig versuchen eine Nachricht zu senden und somit ein gleichzeitiger Buszugriff vorliegt, kommt es zu einer Kollision. Um diese Aufzulösen wird ein CSMA/CR-Verfahren (*Carrier Sense Multiple Access/Collision Resolution*) verwendet, bei dem die höher gewichtete (dominante) Nachricht die niedriger gewichtete (reressive) Nachricht überschreibt, sodass der Übertragungsversuch beendet wird. Die Einteilung der Relevanz einer Nachricht erfolgt auf Basis der sogenannten Identifier(ID)-Bits, wobei niedrige ID-Bits eine höhere Relevanz als höhere ID-Bits haben. Dadurch wird eine Hierarchie der Nachrichten erzeugt, wodurch die Nachricht mit der niedrigsten ID immer gesendet werden darf. Dementsprechend werden wichtigen Nachrichten niedrige IDs zugewiesen [Lawrenz2010]. Die Geschwindigkeit der Datenübertragung zwischen den Teilnehmer ist dabei von der Länge der Leitungen abhängig und beträgt maximal 1Mbit/s. Nach [Werner2014] kann die Bitrate überschlagsmäßig mit

$$\text{Buslänge} \leq 40\ldots50\text{m} \cdot \frac{1\text{Mbit/s}}{\text{Bitrate}} \quad (2.3)$$

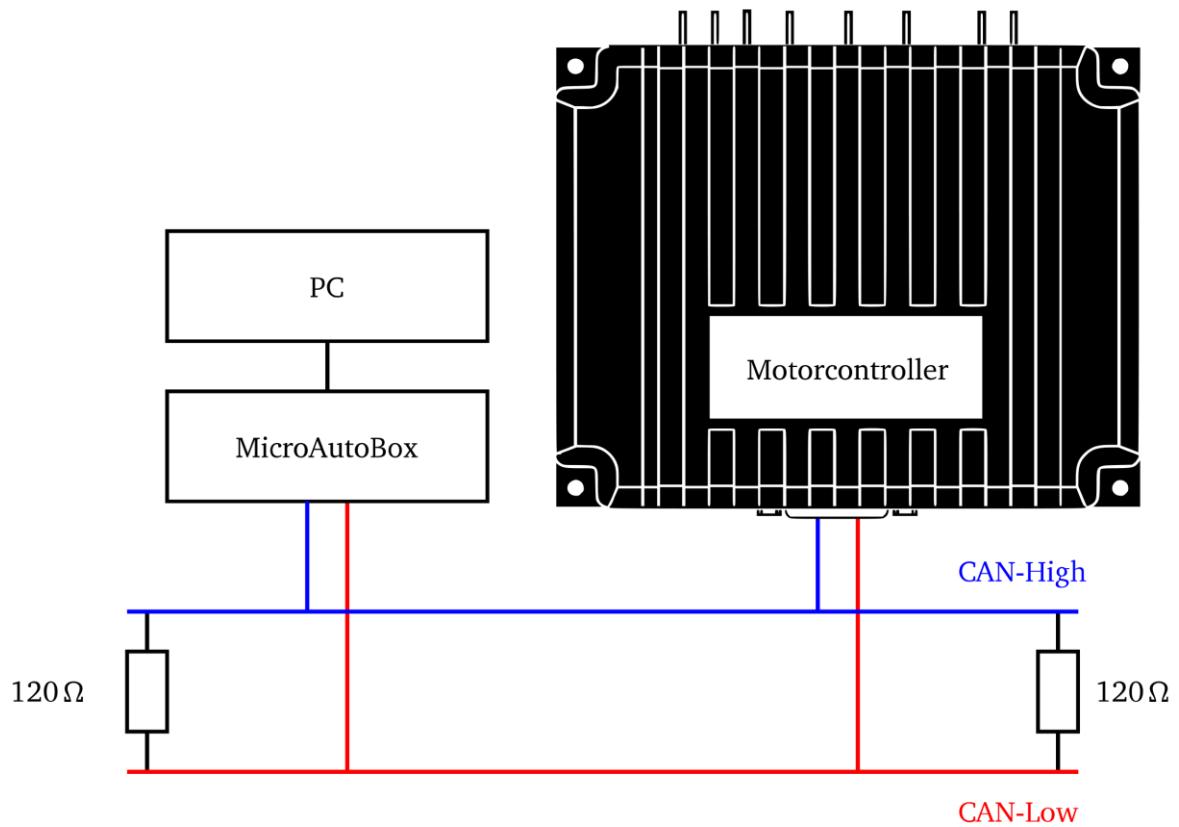


Abbildung 2.8: Aufbau CAN-Bus [manual]

berechnet werden.

Im Folgenden wird speziell auf die in dieser Arbeit eingerichtete CAN-Kommunikation und den Aufbau der CAN-Nachrichten eingegangen: Es wird ein CAN-Bus verwendet, mit dem lediglich zwei Teilnehmer miteinander verbunden werden. Diese sind zum einen die MicroAutoBox und zum anderen der Mikrocontroller des Smart Actuators. Die Länge der Leitungen zwischen den beiden ist wesentlich kürzer als 3m, wodurch nach Formel (2.3) die maximale Bitrate von 1Mbit/s genutzt wird.

2.6 Pulsweitenmodulation

Pulsweitenmodulation (PWM) bietet die Möglichkeit, ein analoges Signal anhand einer digitalen Quelle zu bilden, indem das Tastverhältnis eines Rechteckimpulses bei gleichbleibender Frequenz moduliert wird. Dabei wird der Effekt ausgenutzt, dass sich ein digitales Signal, welches schnell genug und in einem gewissen Tastverhältnis seinen Zustand wechselt, sich verhält wie ein analoges Signal mit konstanter Spannung. So ist es mithilfe von PWM zum Beispiel möglich mit einem Mikrocontroller, der nur einen gewissen Spannungsbetrag liefern kann, verschiedenen hohen Spannungssignale am Endgerät zu erzeugen um es zu betreiben. Es muss allerdings darauf geachtet werden, dass die Pulse im Endeffekt nicht mehr als einzelne Pulse wahrgenommen werden können, das heißt die Frequenz der Pulse muss höher sein als die Abtastrate des Empfängers.

Das Tastverhältnis, oder der Tastgrad, p beschreibt dabei das Verhältnis zwischen der Einschaltzeit t_{ein} und der Periodendauer T :

$$p = \frac{t_{\text{ein}}}{T} = \frac{t_{\text{ein}}}{t_{\text{ein}} + t_{\text{aus}}} \quad (2.4)$$

Das Tastverhältnis kann logischerweise nur Werte zwischen 0 und 1 annehmen.

Der zeitliche Mittelwert der Spannung U_m ergibt sich dann bei einer Betriebsspannung V_b zu:

$$U_m = V_b \cdot p \quad (2.5)$$

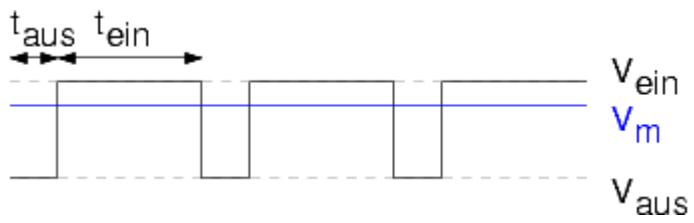


Abbildung 2.9: beispielhaftes PWM Signal

In Abbildung 2.9 ist ein pulsweitenmoduliertes Signal mit dem entsprechendem zeitlichen Mittelwert aufgetragen. Das Tastverhältnis entspricht hier $p = 0,75$, das heißt 75% der Betriebsspannung werden an den angeschlossenen Verbraucher weitergegeben.

3 Vorgehensweise und Testverfahren

Das folgende Kapitel thematisiert wie während des Projekts vorgegangen wird, welche Entwicklungsmodelle zum Einsatz kommen und welche Testkonzepte angewendet werden.

3.1 Entwicklungsmodell

Sowohl für die Entwicklung eines Softwaresystems als auch für die Entwicklung von Elektronik haben sich Modelle etabliert, die eine Systematik in den Entwicklungsprozess bringen. Dadurch soll eine hohe Qualität des Entwicklungsprodukts sichergestellt werden. Unter Qualität, insbesondere bei Softwaresystemen, werden nach [BasSof] bzw. ISO-Norm 25010 [ISO_25010] Eigenschaften verstanden wie Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz und Änderbarkeit.

Für die vorliegende Aufgabenstellung ist ein Softwaresystem zur

- Verarbeitung der Sensorik,
- Anwendung des Regelgesetzes,
- Ansteuerung der Aktorik und
- Bereitstellung der Smart-Funktionalitäten

notwendig. Gleichermassen muss jedoch eine Elektronik entwickelt werden, auf welcher das Softwaresystem ausgeführt wird und durch welche die Schnittstelle zu notwendigen Komponenten hergestellt wird. Um beide Systeme parallel zu entwickeln wird das V-Modell gewählt, das in Abbildung 3.1 dargestellt ist.

3.1.1 Das V-Modell

Das V-Modell ist ein Vorgehensmodell, dass in verschiedenen Ausführungen existiert. Das hier vorgestellte und angewendete Modell entspricht dem Modell nach [Boehm 79]. Als weitere Quelle wird [BasSof] herangezogen und im Folgenden referenziert. Der linke Zweig stellt dabei die konstruktive Entwicklung dar, während jede Aktivität eine dazu korrespondierende Testaktivität im rechten Zweig besitzt.

Konstruktive Entwicklung (linker Zweig)

Zunächst wird demnach eine Anforderungsliste im Zuge der *Anforderungsdefinition* erstellt, die die Entwicklungsaufgabe genauer spezifiziert und später eine Überprüfungsgrundlage bietet inwiefern das fertige Gesamtsystem des Wunschsystems entspricht. Daran anschließend wird ein *funktionaler Systementwurf* durchgeführt. Hierbei werden die Anforderungen in Funktionen überführt, die das Gesamtsystem erfüllen muss um den Anforderungen gerecht zu werden. Darauf folgend findet der *technische Systementwurf* statt. Dafür wird das System in unabhängige Teilsysteme unterteilt und Schnittstellen zur Umwelt ermittelt. Daran logisch anknüpfend wird die *Komponentenspezifikation* durchgeführt. Dafür wird für jede Komponente (elementares Teilsystem) die Aufgabe, das gewünschte Verhalten und Schnittstellen zu anderen Teilsystemen definiert.

Den letzten reinen Entwicklungsschritt bildet die *Komponentenentwicklung*. Dabei werden schließlich die einzelnen Komponenten nach der zuvor erarbeiteten Spezifikation entwickelt. Eine Komponente stellt dabei ein Teilsystem dar, das eine feste Funktion erfüllt, und in der Regel aus einem Softwareteil und einem Hardwareteil besteht. Beide Teile werden parallel in genauer Abstimmung entwickelt, da die gewünschte Funktion nur durch beide Systeme im Zusammenspiel erfüllt werden kann. Das Softwaresystem besteht dabei in der Regel aus einem Treiber, der die komplexe Hardware-Schnittstelle für das Restsystem versteckt und komfortable Schnittstellen bereitstellt. Ein Beispiel dafür stellt der Treiber für den Tauchspulenaktuator dar, der eine Pulsweite in Prozent und ein Aktivierungssignal entgegen nimmt. Intern werden dann daraus Steuersignale für die Halbbrücken generiert und über die Elektronikschaltung dem Aktor zugeführt. Neben Elektronikschaltungen mit zugehörigem Treiber existieren auch *reine Softwarekomponenten*, die unabhängig der Elektronik entwickelt werden wie bspw. eine Verhaltenslogik für ein- und ausgehende CAN-Nachrichten. Ebenso existieren *reine*

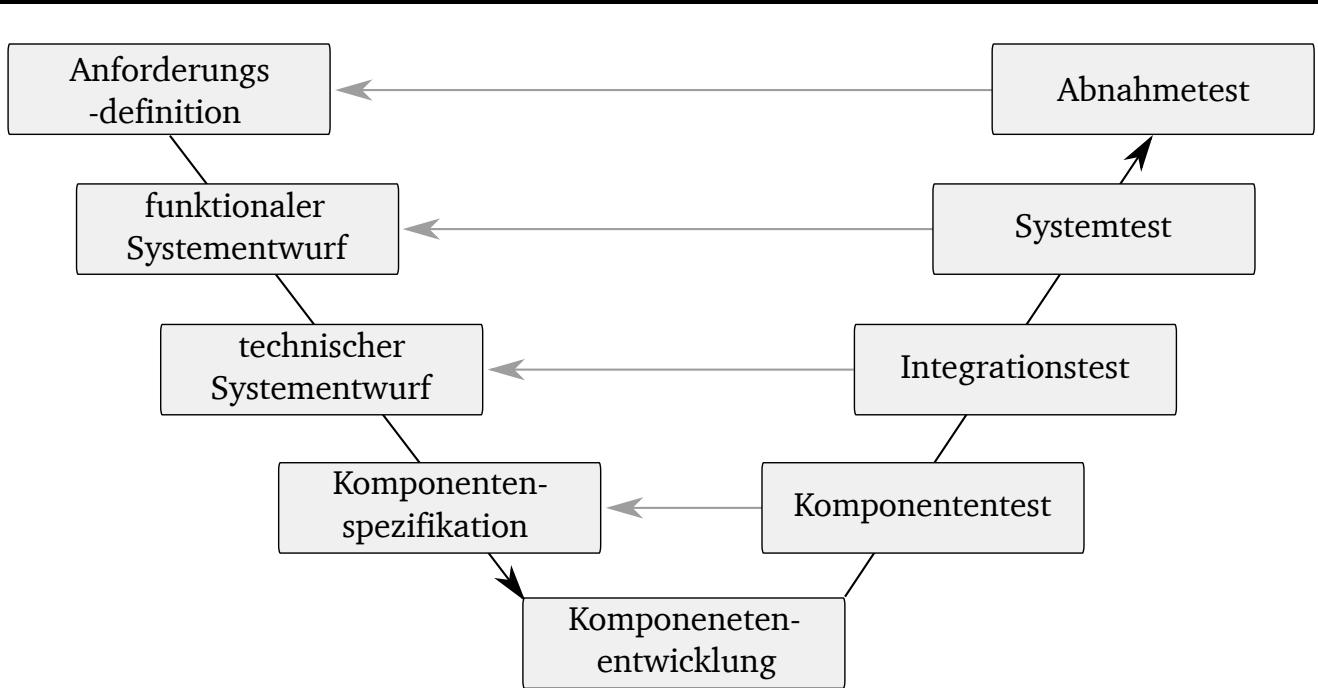


Abbildung 3.1: V-Modell nach [Boehm 79]

Elektronikkomponenten, die keinen direkten Bezug zur Software aufweisen und daher unabhängig zur Software entwickelt werden können.

Testaktivitäten (rechter Zweig)

Alle folgenden Informationen sowie detaillierte Ausführungen bezüglich des Testverfahrens können [BasSof] entnommen werden.

Nachdem die Entwicklung aller Komponenten abgeschlossen ist, müssen die einzelnen Komponenten auf die geforderte Funktionalität überprüft werden. Dazu wird jede Komponente so isoliert wie möglich vom Restsystem getestet, ggf. unter Einsatz sogenannter Testtreiber und Platzhalter, um die zu testenden Komponenten mit Testdaten oder Testfällen auszuführen. Durch die Isolation wird sichergestellt, dass die einzelne Komponente als solche funktioniert und Fehlerzustände leichter eingegrenzt und lokalisiert werden können. Jeder Testfall enthält dabei feste Vorbedingungen, Eingabewerte und ein gefordertes Sollverhalten bzw. geforderte Ausgabewerte. Die Auswahl der Testfälle wird durch die Art der Komponenten unterschieden. *Treiber mit zugehöriger Hardwarekomponente* bieten Schnittstellen, die beliebige Zahlenwerte und Kombinationen entgegen nehmen. Daraus resultiert eine quasi unendliche Menge an möglichen Testfällen, die nicht praktikabel getestet werden können. Vollständiges Testen ist daher nicht möglich. Aus diesem Grund werden die Testfälle aus einem Klassifikationsbaum mit anschließender Grenzwertanalyse gewonnen. Dabei wird jeweils mindestens ein Testwert aus jeder Gruppe ausgewählt. Die Gruppierung wird so vorgenommen, dass von jeder Gruppe aus Testwerten gleiches Verhalten angenommen wird. Als Beispiel kann wieder die Ansteuerung des Tauchspulenaktors herangezogen werden. Es wird davon ausgegangen, dass alle Werte zwischen -100 und 0 eine Bewegung in eine feste Richtung bewirken. Dagegen bewirken Werte von 0 bis 100 eine Bewegung in die Gegenrichtung. Werte außerhalb des Zahlenbereichs werden auf entsprechend -100 oder +100 abgerundet. Durch eine Grenzwertanalyse werden diese Gruppen (auch Äquivalenzklassen genannt) noch in Untergruppen unterteilt, in denen die oft kritischen Grenzwerte (-100, 0 und 100) noch eigene Gruppen bilden. Eine Sonderstellung besitzt die Null, weil hier keine Reaktion erwartet wird. Durch kombinatorische Paarbildungen werden darüber hinaus auch beliebige Kombinationen aus abhängigen Eingangswerten, bspw. Aktivierungssignal (enable) und Pulsbreite (PWM [%]), berücksichtigt. Um die Testfallgenerierung durch Klassifikationsbäume zu verdeutlichen, ist in Abbildung 3.2 der Klassifikationsbaum grafisch dargestellt. Die untere Zeile aus konkreten Werten stellt beispielhafte Testwerte aus der darüberliegenden Äquivalenzklasse dar. Darunter ist über die Gitterstruktur ange deutet wie aus der paarweisen Kombination der einzelnen Testwerte Testfälle bestimmt werden. *Testfall 1* entspricht dem Aufrufen des Treibers mit einer PWM von -100 % und einem aktivierte *enable*-Signal. Es wird daher erwartet, dass durch die H-Brücke der Motor voll in "negative" Bewegungsrichtung bestromt bzw. beschleunigt wird. Ob die Elektronik als solche den Erwartungen bzw. der Spezifikation entspricht, kann durch Strom-/Spannungs- sowie Temperaturmessungen verifiziert werden.

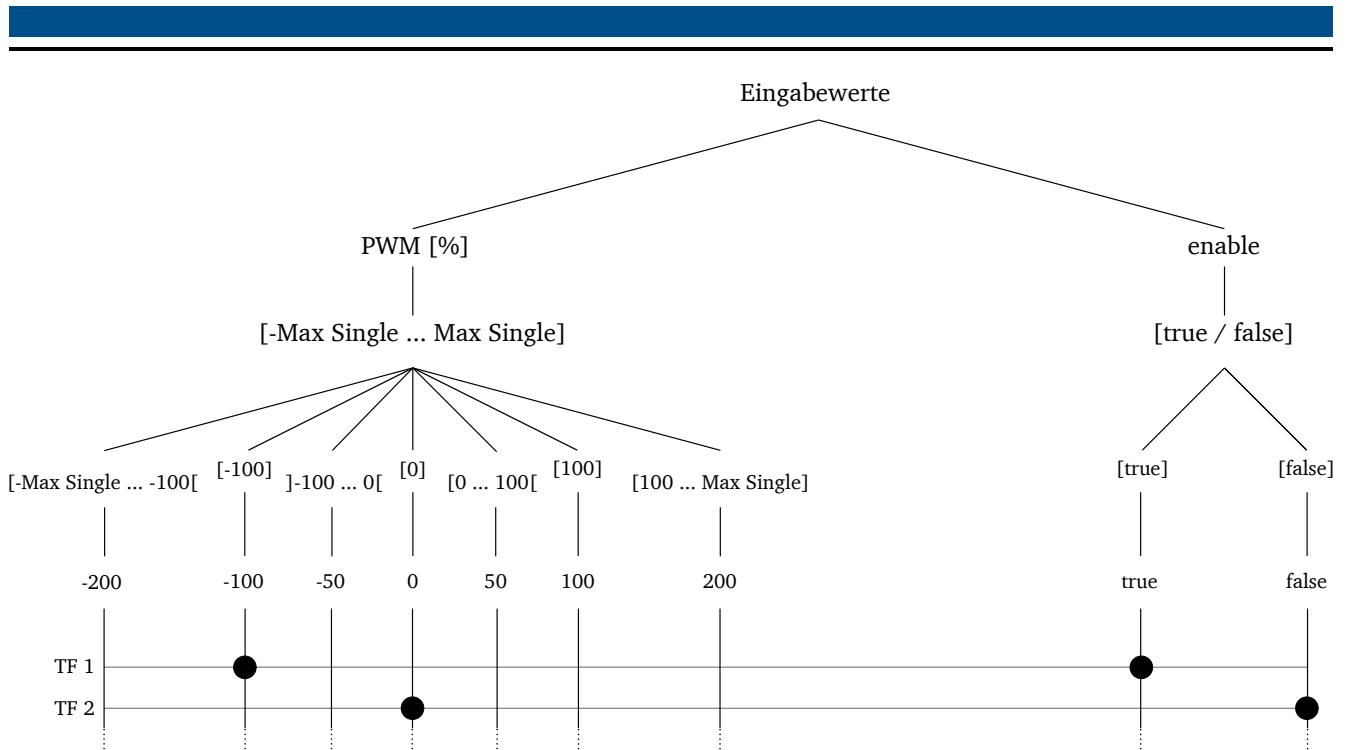


Abbildung 3.2: Klassifikationsbaum am Beispiel der H-Brücke

Für *reine Softwarekomponenten* hingegen werden Testfälle auf eine geeigneter Weise generiert. Bei Softwarekomponenten auf Basis eines Zustandsautomaten werden Testfälle aus einer zustandsbasierten Testfallgenerierung gewonnen. Dabei wird als Testziel vorausgesetzt, dass alle Entscheidungen (Knoten) eines Zustandsautomaten mindestens einmal ausgeführt werden, was als Entscheidungsüberdeckungskriterium bezeichnet wird. Dieses Kriterium schließt auch automatisch ein, dass alle Zustände (Knoten) besucht worden sind. Wie viele Testfälle getestet werden müssen hängt dabei stark von der Größe und Komplexität des Zustandsautomaten ab.

Nach dem Test der einzelnen Komponenten wird eine Integration vollzogen. Unter Integration versteht man dabei das Zusammenführen der Komponenten zu einem Gesamtsystem. Es stehen viele Varianten zur Verfügung, dazu zählen *Bottom Up*-, *Top Down*- und *Big Bang*-Integration. Letztere Variante sieht vor, dass alle Komponenten in einem Zug ins Gesamtsystem integriert werden. Auftretende Fehler können jedoch in diesem Fall nur schwierig zugeordnet werden. Günstiger ist daher die *Bottom Up*-Integration, wobei nach und nach Komponenten hinzugefügt werden und übergeordnete Komponenten, die mit mehreren Einzelkomponenten Schnittstellen haben, später integriert werden. Da im vorangehenden *Komponententest* bereits die Komponenten getestet sind, ist im anschließenden *Integrationstest* mit Fehlern zu rechnen, die durch Schnittstellen zwischen den Komponenten hervorgerufen werden. Es kann davon ausgegangen werden, dass die Komponenten intern funktionieren und ausschließlich das Zusammenspiel der Komponenten fehlerhaft ist.

Zuletzt entsteht somit ein Gesamtsystem, das darauf getestet werden muss, ob die im *funktionalen Systementwurf* festgelegten Funktionen erfüllt werden. Um dies zu testen werden *Use-Case*-Szenarios erstellt, die eine übliche Nutzung simulieren. Dabei wird von mehreren Akteuren ausgegangen, die mit dem System in der realen Anwendung agieren werden. Der Schaltaktor wird bspw. über ein übergeordnetes Steuergerät (*electrical Control Unit*) angesteuert. Die Schnittstelle wird durch CAN realisiert, sodass ein entsprechendes Steuergerät über CAN Signale/Befehle einen Schaltvorgang anfordern kann oder eine Fehlermeldung auslesen kann. Einen typischen *Use-Case* würde bspw. ein Schaltbefehl vom neutralen in den zweiten Gang darstellen. Dieser kann erfolgreich ablaufen oder durch eine mechanische Blockade verhindert werden. Abschließend erfolgt der Abnahmetest, der in der Regel durch den Auftraggeber durchgeführt wird. Hier wird explizit verglichen, ob die versprochenen Leistungen im Pflichtenheft auch erreicht werden.

Diese und weitere Informationen für den interessierten Leser finden sich in [BasSof].

3.1.2 Konkretes Vorgehen mit inkrementellem Entwicklungsansatz

Neben dem V-Modell fließen auch inkrementelle Ansätze in die Entwicklung ein. So wird das V-Modell mehrmals durchlaufen, während nach jeder Iteration ein ausführbares und testbares Entwicklungsprodukt vorliegt. Diese Zwischenprodukte entsprechen der Anforderungsliste unter Umständen nur teilweise, sodass ggf. mehrere Iterationen bis zu einem zufriedenstellenden Produkt notwendig sind. Für das vorliegende Projekt wird in der ersten Iteration ein Prototyp angestrebt, der bereits die Grundfunktionalitäten bereitstellt. Dies umfasst die Möglichkeit einen Schaltvorgang

Iteration	Schaltzeit	Nichtflüchtige Kalibrierung	kompakte Baugröße	Effizienz	Fehlererkennung	...
1	✓	✓	✗	✗	✗	...
2	✓	✓	✗	✗	✓	...
3	✓	✓	✗	✓	✓	...
...						

Tabelle 3.1: Beispielhafter Ausschnitt aus Systemtest-Ergebnissen über mehrere Iterationen

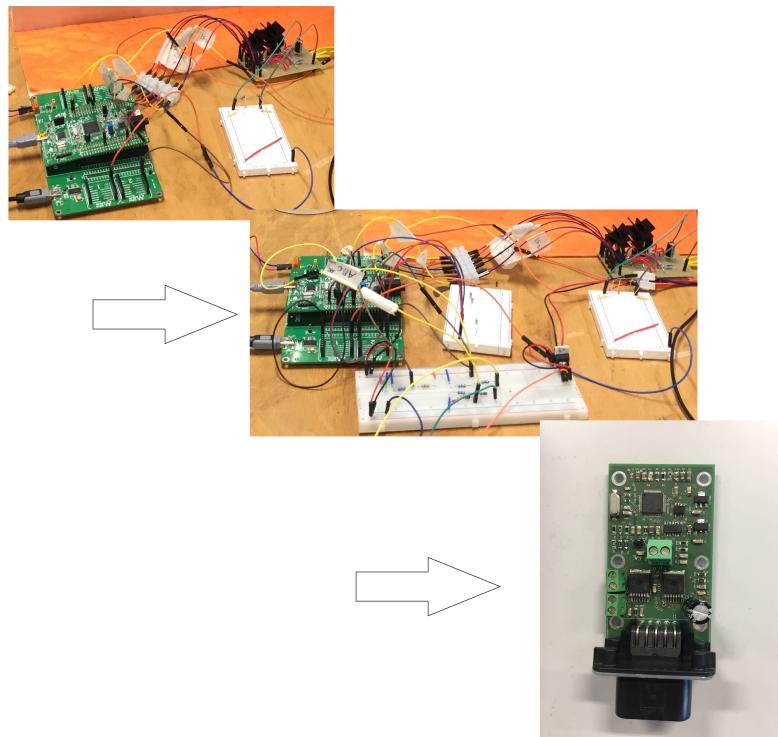


Abbildung 3.3: Bildliche Darstellung des Prototypings (Iterationsprozesses)

nach Anforderungsspezifikation durchzuführen. Dafür muss unter Anderem die Aktorik ansteuerbar sein und die Sensorknik notwendige und hinreichend genaue Regelgrößen liefern. Anforderungen an Größe, Effizienz oder Integrierbarkeit sind zunächst nur sekundär oder überhaupt nicht verfolgt worden. Durch das Ergebnis werden Erkenntnisse gewonnen, die in der zweiten Iteration berücksichtigt werden. Bei folgenden Iteration wird versucht über geeignete Maßnahmen (bspw. durch eine angepasste Komponentenspezifikation) bisherige Ergebnisse zu verbessern und weitere Anforderungen zu erfüllen, wie die Fähigkeit Fehlerzustände (Überstrom, etc.) zu erkennen. Einen wesentlichen Iterationsschritt stellt der Übergang von Steckbrett- auf Platinenbasis dar. Durch diesen Übergang treten weitere Anforderungen wie Baugröße und Integrierbarkeit in den Vordergrund. Nach dem letzten Iterationsschritt sollten alle Anforderungen nachweislich erfüllt sein. In Abbildung 3.3 sind die Iterationen bildlich dargestellt, während in Tabelle 3.1 ein Beispielhafter Ausschnitt aus dem Ergebnis des jeweiligen *Systemtests* pro Iteration dargestellt sind.

4 Auswahl elektronischer Komponenten und Verschaltung

4.1 Anforderungen an die Komponenten

Die benötigten elektronischen Komponenten werden üblicherweise in „Passive Bauelemente“ und „Aktive Bauelemente“ untergliedert. Unter passiven Bauelementen versteht man Komponenten, die das Eingangssignal ohne Verstärkung übertragen. Aktive Bauelemente benötigen meist eine Hilfsquelle und können somit ein Eingangssignal verstärken [haendschke]. Für den Einsatz elektronischer Komponenten im Automobilbereich hat sich für passive Bauteile die Zertifizierung nach AEC-Q200 und für aktive Bauteile nach AEC-Q100 als Qualitätsstandard etabliert. Anhand dieser Zertifizierungen wird eine nachweisliche Belastbarkeit der Bauelemente nach dem jeweiligen Standard definiert. Für die Anforderungen an die Platine werden die passiven Bauelemente nach AEC-Q200 und die aktiven Bauelemente nach AEC-Q100 mindestens in Grade 2 eingestuft, um den Temperaturanforderung bis 105 °C gerecht zu werden [aecq].

4.1.1 Passive Bauteile

Unter die benötigten passiven Bauteile fallen lediglich Kondensatoren und Widerstände, welche nach dem AEC-Q200 Standard ausgewählt werden. Anforderung ist danach eine hohe thermische Belastbarkeit von mindestens 105 °C. Um eine geringe Bauteilgröße für die Platine zu bekommen sollten sowohl Widerstände als auch Kondensatoren als SMD (*surface-mounted device*) gekauft werden. Für Entkopplungskondensatoren werden nach [ldo] Keramikkondensatoren mit XR7 präferiert. Als Baugröße bei manueller Bestückung bieten sich die Standardgrößen 1206 oder 0805 (Angabe in $\frac{1}{100}$ Zoll), welche noch per Hand lötbar sind. Die Widerstände für die Sensorik müssen mit sehr hoher Präzision gewählt werden um Messfehler aufgrund falscher Widerstandswerte zu vermeiden. Im Anwendungsfall bieten sich daher Dünnschichtwiderstände an.

4.1.2 Aktive Bauteile

Die aktiven Bauelemente der Platine müssen dafür ausgelegt werden, um im späteren alle Anforderungen aus der Anforderungsliste gerecht zu werden. Demnach müssen sie in ihrer logischen Verschaltung einerseits in der Lage sein den Tauchspulenaktor anzusteuern und auszuregeln, andererseits aber auch die Bedingungen über Leistungsaufnahme etc. erfüllen. Kern der Platine ist ein Mikrocontroller, welcher schnell genug sein muss um Gangwechsel in weniger als 0,1s durchzuführen. Weiterhin muss der Mikrocontroller die CAN-Kommunikation mit der MicroAutobox unterstützen. Da der Mikrocontroller nicht über die 13.8 V der Autobatterie gespeist werden kann, wird ein Spannungsregler benötigt, welcher auf die Betriebsspannung des Mikrocontrollers regelt. Um aus den physikalischen Bussignalen CAN-HIGH und CAN-LOW für den Mikrocontroller verwertbare Signale zu erhalten wird ein CAN-Transceiver benötigt. Über ihn ist eine Kommunikation des Mikrocontrollers mit der CAN-Peripherie erst möglich. Zum Regeln des Tauchspulenaktors ist ein Motortreiber nötig, welcher vom Mikrocontroller angesteuert wird um die Spannung an den Tauchspulenaktor durchzuschalten. Eine Ansteuerung des Motortreibers mittels PWM-Signal sollte möglich sein. Der Motortreiber muss dabei für zukünftige Aktorkonfigurationen für Ströme bis zu 55A ausgelegt sein. Des Weiteren wird eine Spannungsebene von 5 V benötigt um die Lagesensorik des Aktors zu unterstützen, wofür ein zweiter Spannungsregler eingeplant werden muss.

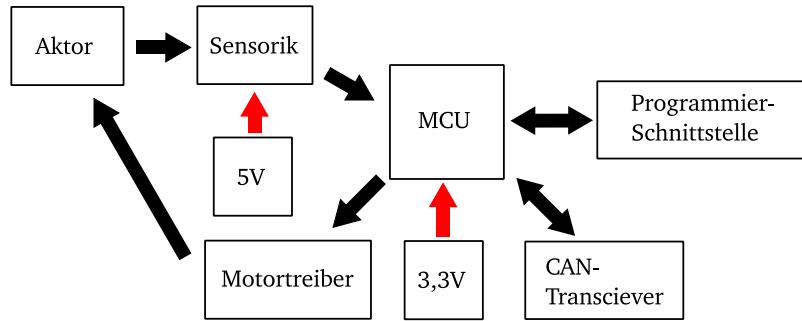


Abbildung 4.1: Aktive Bauelemente

4.2 Komponentenauswahl

Anhand der Anforderungen an die Platinenelemente sind die endgültigen Komponenten der Platine zu wählen. Es müssen Mikrocontroller, Spannungsregler, CAN-Transciever und Motortreiber ausgewählt werden. Weiterhin wird eine Temperaturbeständigkeit bis 105 °C gefordert.

4.2.1 Mikrocontroller

Die Recheneinheit zur Regelung des Tauchspulenaktors ist ein STM32F405RGT7. Dieser ist mit verschiedenen Pin-Anzahlen erhältlich und auf der Platine aus Platzgründen in der Ausführung LQFP64 mit 64 Pins verbaut. Der Mikrocontroller zeichnet sich durch seine hohe CPU-Geschwindigkeit von 168 MHz und seine große Programm-Speichergröße von 1 MB aus. Die Produktbezeichnung „T7“ steht dabei für die zulässige Betriebstemperatur von -40...105 °C. Weiterhin besitzt dieser Mikrocontroller die geforderten CAN-Schnittstellen, anhand derer mit der MicroAutobox kommuniziert werden soll. In Abbildung 4.2 ist die Minimalbeschaltung des Mikrocontrollers zu sehen. Diese ist dem Datenblatt des STM32 entnommen[stm32]. Die Pins mit den Bezeichnungen VDD...VDD_4 sind die Versorgungspins des Mikrocontrollers. An diesen Pins liegt die Versorgungsspannung an, welche nach dem Datenblatt zwischen 1,8 und 3,6 V liegt. Diese ist aufgrund der Versorgungsspannung anderer Bauteile zu VDD = 3.3 V gewählt. Die VDD Pins werden durch die Keramikkondensatoren C7-C11 entkoppelt, welche nach dem Datenblatt dimensioniert sind und möglichst nah an den dem jeweiligen VDD-Pin plaziert werden sollen. Die Entkopplung wird benötigt um Welligkeiten der Spannungsversorgung zu filtern und parasitäre Induktivitäten zu entkoppeln[decoupling]. Der Pin VBAT wird unter Anderem für die Versorgung der Echtzeituhr (RTC), der Backup-Register und des Backup-SRAMs genutzt und wird, wenn keine zweite Spannungsversorgung neben der Hauptversorgung vorhanden ist, ebenfalls an VDD angeschlossen [stm32]. VDDA ist der Pin für die Spannungsversorgung des Analog-Digital-Converters (ADC) und wird ebenfalls mit der Versorgungsspannung von VDD angeschlossen. Die Kapazitäten C5 und C6 sind nach den Herstellerangaben dimensioniert und sorgen ebenfalls für eine Entkopplung der Spannungsversorgung. Die Induktivität L1 bietet die Möglichkeit, je nach Welligkeit der Versorgungsspannung, diese über einen LC-Filter zu filtern um eine bessere Auflösung des ADCs zu gewährleisten. Wenn dieser Filter nicht benötigt wird, sollte L1 durch einen 0 Ω Widerstand ersetzt werden. VCAP_1 und VCAP_2 sind die Ausgänge des internen Voltage Regulators des STM32 und C5 bzw. C6 werden zur Glättung der intern geregelten Spannung genutzt[stm32]. Die Pins VSS, VSS_2, VSSA sind die GND Anschlüsse für den Mikrocontroller und den internen ADC.

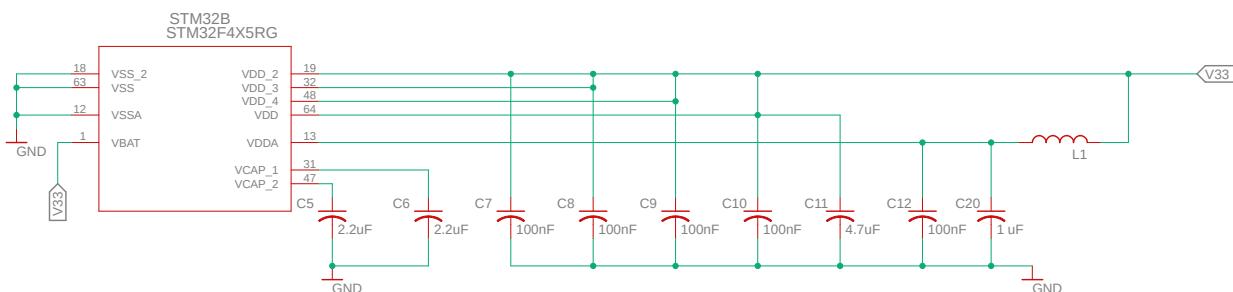


Abbildung 4.2: Minimalbeschaltung des STM32F405RGT7

In Abbildung 4.3 sind die Pin-Belegungen des Mikrocontrollers dargestellt, welche durch Tabelle 4.1 genauer beschrieben sind. Unter Sensorik fallen die Ausgänge für Lagesensorik, Temperatursensorik und Messung der Versorgungsspannung. Die UART Pins können zum debuggen auf der endgültigen Platine genutzt werden. Zur CAN-Kommunikation werden die Pins CAN1-TX(PA11) und CAN1-RX(PA12) benötigt, womit der Mikrocontroller über einen CAN-Transciever Nachrichten an die MicroAutobox senden und empfangen kann. Zum Programmieren des Mikrocontrollers wird eine Serial Wire Debug (kurz: SWD) Schnittstelle aufgebaut, welche sich mittels externem ST-Link/V2 mit dem Computer verbinden lässt. Dazu werden die jeweiligen Pins über den Platinenstecker nach außen geführt. Weiterhin müssen Pins zum Beschalten der H-Brücke und zum Auslesen der Strommessungen belegt werden. Die Pins OSC_IN und OSC_OUT sind zum Anschließen eines externen Quarzes, welcher als Taktgeber für den Mikrocontroller genutzt wird. Der Pin NRST wird für den Fall eines notwendigen Resets herausgeführt.

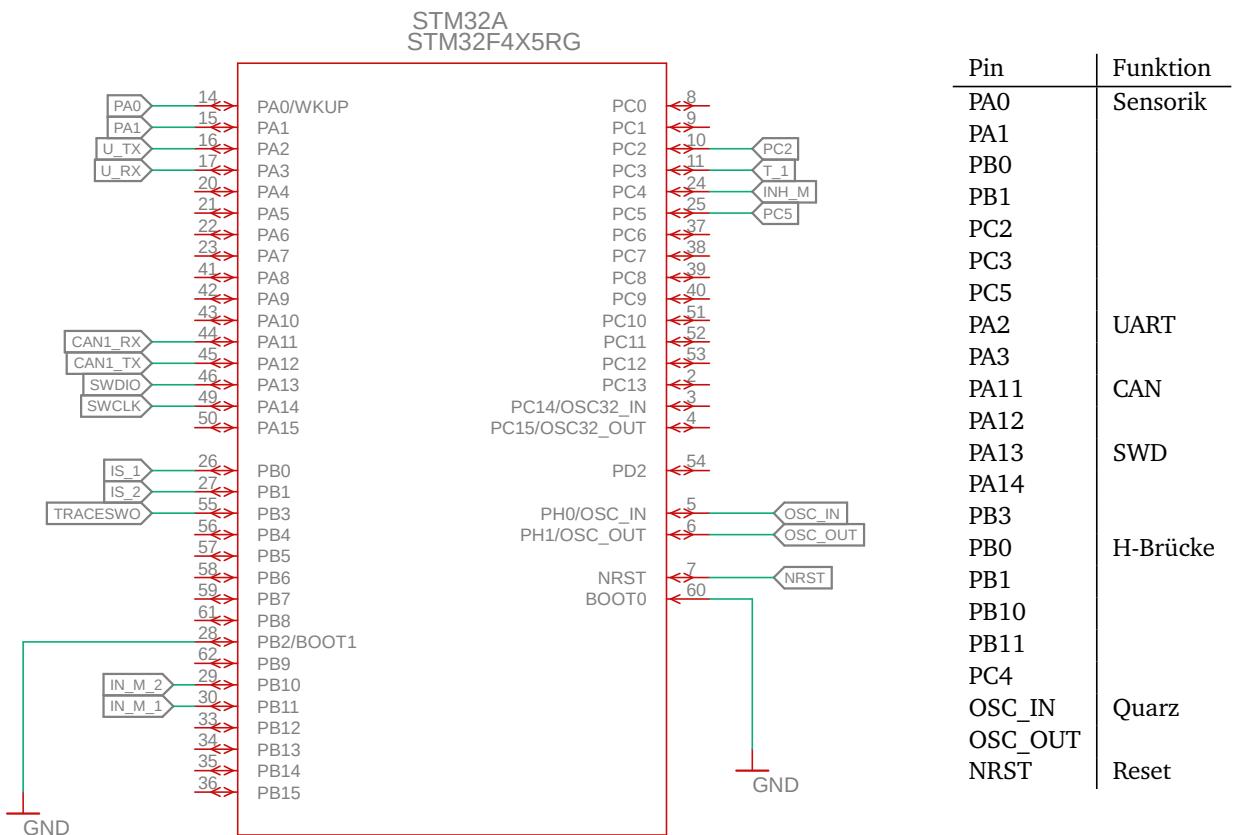


Abbildung 4.3: Pin-Belegung des Mikrocontrollers

Tabelle 4.1: Pin-outs

Wichtig bei der Beschaltung ist außerdem die Belegung von BOOT0 und BOOT1, welche den Speicherbereich definieren aus dem der Mikrocontroller beim Startvorgang sein Programm lädt. Nach Tabelle 4.2 muss für den Start im Main Flash memory BOOT0 auf GND-Niveau gesetzt werden, während die Belegung von BOOT1 undefiniert ist und somit mit GND belegt werden kann.

BOOT1	BOOT0	Boot mode	Aliasing
x	0	Main Flash memory	Main Flash memory is selected as the boot space
0	1	System memory	System memory is selected as the boot space
1	1	Embedded SRAM	Embedded SRAM is selected as the boot space

Tabelle 4.2: Boot modes des STM32F405RGT7 nach [stmref]

Um möglichst akurate Taktraten zu haben und somit Schaltvorgänge und Sensorikdaten genau und reproduzierbar zu machen, wird wie vom Hersteller des Mikrocontrollers empfohlen ein externer Taktgeber benutzt [stmref]. Auf der Platine wird hierbei ein ABLS-8.000MHZ-K4T von Abracon genutzt, welcher eine Nennfrequenz von 8MHz hat. Abbildung 4.4 zeigt die letztendliche Verschaltung auf der Platine. Die Lastkapazitäten CQ1 und CQ2 von 22pF sind dabei nach dem

Datenblatt des Herstellers gewählt. Die Widerstände R19 und R20 sind einerseits verbaut, um den externen Quarz vom Mikrocontroller trennen zu können, andererseits wird R20 benutzt, um den Stromfluss des Quarzes zu beschränken [stmquarz]. Nach [stmquarz] lässt sich eine Abschätzung von R20 über

$$R_{20} = \frac{1}{2\pi f C_{01}}$$

gewinnen. Damit ergibt sich ein Richtwert von etwa 900Ω bei $CQ1 = 22\text{pF}$. Ein zu niedriger Widerstand erhöht die Verlustleistung über den Quarz, während ein höherer Widerstand zum Stillstand der Oszillation führen kann. Auf der Platine ist ein Widerstand von $1\text{k}\Omega$ gewählt.

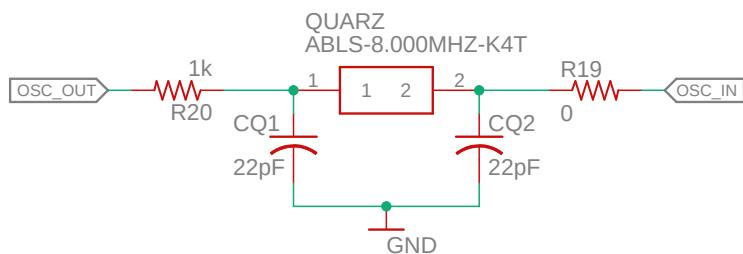


Abbildung 4.4: Anschluss externer Quarz

4.2.2 CAN-Transciever

Zur Kommunikation des Mikrocontrollers mit der MicroAutobox ist eine Signalübertragung über CAN vorgesehen. Da der Mikrocontroller die physikalischen Bussignale nicht verarbeiten kann, wird die Kommunikation des Mikrocontrollers mit dem Bussystem über einen CAN-Transciever gestaltet. Dieser sorgt dafür, dass den Mikrocontroller nur für ihn lesbare Daten erreichen und übersetzt gleichzeitig die vom Mikrocontroller ins Bussystem gesendete Nachrichten. Die Pins des Mikrocontrollers für diese Aufgaben sind CAN1-RX zum Erhalten von Informationen und CAN1-TX zum Senden von CAN-Nachrichten. Auf Busebene gibt es die Signale CAN-HIGH und CAN-LOW, welche in Kapitel **BLA** beschrieben wurden. In Abbildung 4.5 ist die Verschaltung des CAN-Transcievers SNHVD230QD von Texas Instruments zu sehen. An Pin VCC wird die Spannungsversorgung von 3.3 V angeschlossen. VREF ist ein Ausgangspin mit halber VCC-Spannung beispielsweise zum Entwerfen einer Split-Termination. Pin D ist der Anschlusspin für die gesendeten Nachrichten des Mikrocontrollers über CAN1-TX und Pin R sendet die übersetzten CAN-Nachrichten des Bussystems an CAN1-RX.

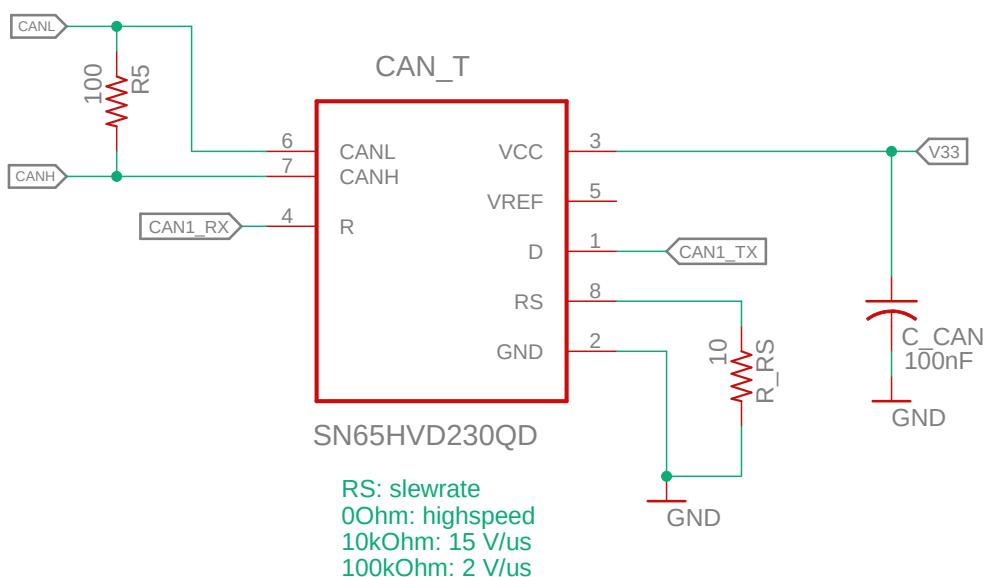


Abbildung 4.5: CAN-Transceiver Verschaltung

Über den Pin RS lassen sich durch den Widerstand R_RS verschiedene *slew rates* einstellen und damit, wie schnell der Ausgangstransistor bei der Übermittlung von Nachrichten durchschaltet [cantrans]. Für einen Widerstandswert von 0Ω ist die Geschwindigkeit maximal, während sie bei $10\text{ k}\Omega$ etwa $15 \frac{\text{V}}{\mu\text{s}}$ beträgt. Zwischen CAN-HIGH und CAN-LOW wird nach dem Datenblatt ein 120Ω geschaltet. Bei der Wahl von R5 und R_RS wurde sich an der bestehenden Konfiguration des STM-Discovery-Shields orientiert, mit der die CAN-Kommunikation aufgebaut wurde.

4.2.3 Spannungsversorgung

Die komplette Elektronik wird auch weiterhin mit dem bisher verwendeten Manson SBC-2130 Battery Charger versorgt. Dieser stellt eine konstante Spannung von 13,8 Volt. Da die verschiedenen Komponenten jedoch Versorgerspannungen von 3,3 Volt und 5 Volt benötigen, muss die Schaltung durch einen Spannungsregler erweitert werden. Zusätzlich werden dadurch Schwankungen in der Eingangsspannung geglättet.

Low Dropout Spannungsregler

Ein LDO ist ein Festspannungsregler, der eine festgelegte und somit invariable Ausgangsspannung liefert, die sich auch dann nicht ändert, wenn die Eingangsspannung schwankt. Die Schaltung eines LDO-Reglers aus einer Referenzspannungsquelle, einem Differenzverstärker und einem Stellglied in Form eines Leistungstransistors. Die hier verwendeten Ausführungen sind P-Kanal-MOSFET-basierte Regler. Das Blockschaltbild eines solchen LDOs ist in folgender Abbildung schematisch dargestellt.

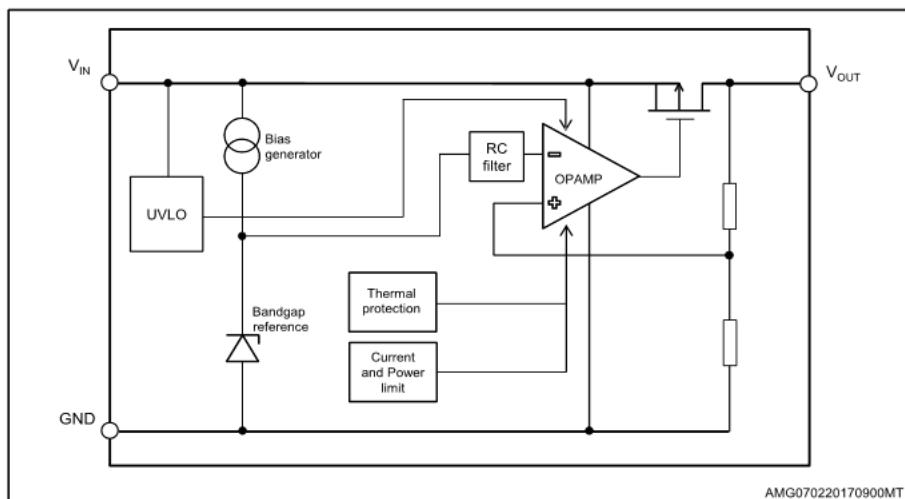


Abbildung 4.6: Blockschaltbild LDO

Der Differenzverstärker vergleicht die Ausgangsspannung mit einer stabilen Referenzquelle aus einer Zenerdiode, sodass diese gemessene Spannungsabweichung über das Stellglied ausgeregelt werden kann. Ist die Ausgangsspannung zu niedrig, so wird der Transistor stärker angesteuert bis die geforderte Ausgangsspannung erreicht wird, im umgekehrten Fall wird der Strom über den Transistor reduziert. Der Transistor wird in dieser Schaltung also quasi wie ein veränderlicher Widerstand verwendet, an dem die überflüssige Spannungsdifferenz abfällt und in Wärme umgewandelt wird. Die verwendeten LDOs besitzen außerdem eine Strombegrenzungsschaltung und eine Schutzschaltung, die die Betriebstemperatur überwacht und das Bauteil vor thermischer Überlastung schützt, sowie eine Unterspannungsabschaltung.

Verschaltung auf Platine

Grund für die Wahl dieser Art von Spannungsreglern ist ihre kompakte Bauform, ihr günstiger Einkaufspreis und das geringe Rauschen im Vergleich zu Schaltreglern da keine Schaltvorgänge auftreten. Auf der Platine werden LDOs vom Typ LDL1117 von STMicroelectronics verwendet. Für die 3.3 V Versorgung ist dies der LDL1117S33R und für die 5 V der LDL1117S50R LDO. Nach dem Datenblatt [ldo] ist die Eingangskapazität zu $1\mu\text{F}$ und die Ausgangskapazität zu $4.7\mu\text{F}$ zu wählen. Diese werden aus Stabilitätsgründen und zur Entkopplung verwendet. Empfohlen wird weiterhin Keramikkondensatoren zu verwenden, welche X5R oder X7R Dielektrika aufweisen. In Abbildung 4.7 ist der Schaltplan der Spannungsversorgung für die Platine zu sehen. An VIN wird die Batteriespannung angelegt, welche mit $1\mu\text{F}$ gegen

GND entkoppelt wird. An VOUT liegen die jeweiligen 3.3 V bzw. 5 V an, welche jeweils über $4.7 \mu\text{F}$ gegen GND geschaltet sind. Zur allgemeinen Spannungsglättung wird eine große Kapazität zwischen der Batteriespannung und GND geschaltet, welche starke Schwankungen beim Durchschalten der H-Brücke verhindern soll.

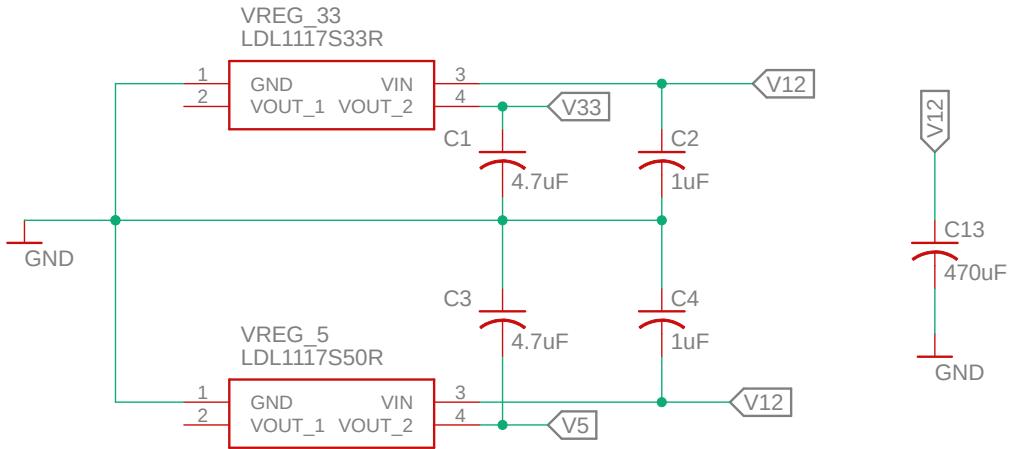


Abbildung 4.7: Schaltplan Spannungsversorgung

4.3 Platinenanschluss

Die Platine wird mit einem 1-776267-1 von TE Connectivity verbunden um notwendige Signalleitungen nach außen zu führen und zusätzlich Spannungsversorgung und Lagesensorik mit der Platine verbinden zu können. In Abbildung 4.8 ist die Pin-Belegung des Steckers zu sehen. Um den Mikrocontroller über SWD programmieren zu können, muss der Stecker die notwendigen Signalleitungen SWDIO, SWCLK, TRACESWO, V33 und GND nach außen führen. Über diese Pins können über einen ST-Link/V2 Programme von einem Computer auf den Mikrocontroller übertragen werden. Weiterhin soll der Mikrocontroller über CAN-Signale mit der MicroAutobox kommunizieren können. Dazu werden die Schnittstellen CANH(CAN-HIGH) und CANL(CAN-LOW) über den Stecker nach außen geführt. Die Steckerpins V5, L_1, L_2 und GND werden für die Lagesensorik benötigt. Pin 4 und Pin 5 sind beide mit der Batterie verbunden und bilden die Spannungsversorgung der gesamten Platine. Aufgrund der potentiell hohen Ströme wird die Versorgung über zwei Pins zugeführt. Über den NRST-Pin lässt sich der Mikrocontroller zurücksetzen.

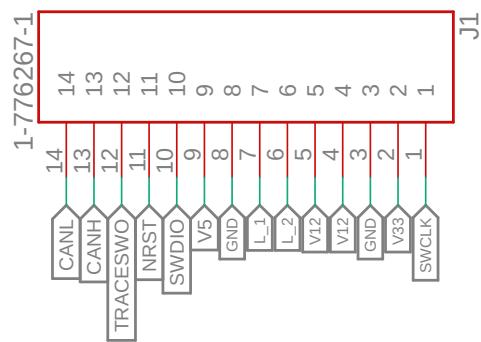


Abbildung 4.8: Anschlusspins Stecker

4.4 H-Brücke

Jonas fragen.

4.5 Sensorik

Malte fragen.



5 Platinenentwurf

5.1 Elektronikgehäuse und Anforderungen an die Platinendimensionierung

Die Platine wird in ein Gehäuse eingebettet, welches parallel zu und in Absprache mit dieser Arbeit entwickelt wurde. Die Anforderung, den Aktor durch die entwickelte Elektronik zu einem Smart Actuator zu transformieren wird erfüllt, in dem die Platine durch das Gehäuse am Aktorgehäuse verschraubt wird. Abbildung 5.1 zeigt schematisch die Lage der Platine (grün) im Gehäuse. In grau ist der Verbindungsstecker zu sehen, der abgedichtet im Gehäuse liegt.

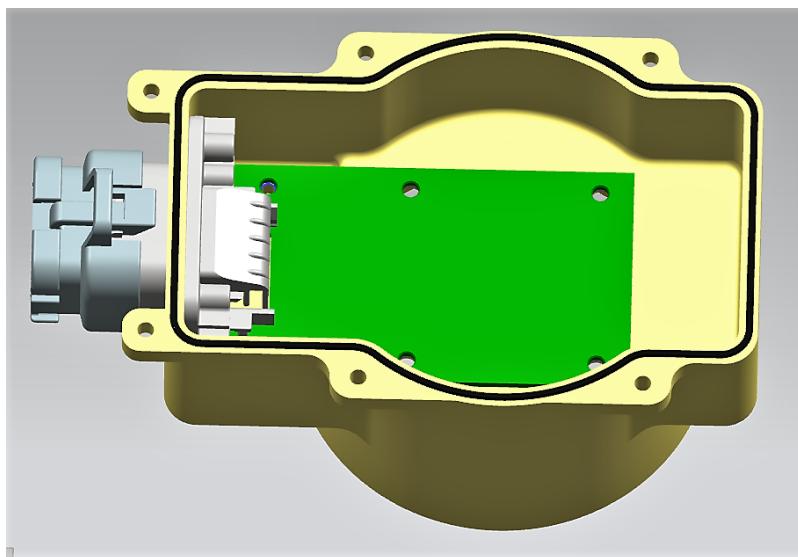


Abbildung 5.1: Derzeitiger Entwurf des Elektronikgehäuses

Durch die Größe des Aktors ergeben sich Höchstmaße an das Platinengehäuse und die Platine selbst. Um ein Einbauen zu garantieren, dürfen die Außenmaße der Platine $88.8 \text{ mm} \cdot 50 \text{ mm}$ nicht überschreiten. In diesem Fall war der Platz ausreichend, um alle nötigen Schaltungen und Funktionen auf der Platine zu verwirklichen. Als Ausweichlösung, falls die ebenen Maße nicht ausgereicht hätten war angedacht, die Elektronik auf zwei Platinen aufzuteilen (z.B. Logik- und Leistungsteil), und sie in zwei Ebenen parallel übereinander anzutragen. Dies ist weiterhin denkbar, falls die Elektronik erweitert werden soll. Gelöste Teile auf der Platine, welche einen großen Bauraum nach oben benötigen sind vor allem der THT Kondensator mit $1000 \mu\text{F}$ sowie die Wire-to-Board Reihenklemmen. Diese sind bereits auf der linken Seite (Steckerseite) der Platine angeordnet, sodass über der rechten Seite relativ unproblematisch eine zweite Platinenebene eingeführt werden könnte. Da das Gehäuse aufgrund von kostengünstiger und einfacher Fertigung aus Plastik geplant und hergestellt wurde, muss eine Lösung gefunden werden um die Platine mit GND zu verbinden, zu kühlen und elektromagnetisch abzuschirmen. Es bietet sich an, eine Aluminiumplatte, die den Maßen der Platine entspricht, zwischen Platine und Gehäuse anzubringen. Dabei wird auf Aluminium zurückgegriffen, da es sich um ein sehr leichtes und zudem kostengünstiges Metall handelt, welches außerdem eine hohe Wärmeleitfähigkeit von $204 \frac{\text{W}}{\text{mK}}$ besitzt [WIEC\bigrth]. Durch das Loch in der Unterseite des Gehäuses, welches in Abbildung 5.2 zu erkennen ist, liegt die Aluminiumplatte, in dieser Darstellung grün eingefärbt, am Aktorgehäuse an. Das hat den Vorteil, dass die Platine Wärme an die Aluminiumplatte über Konduktion abgeben kann, welche wiederum ihre Wärme an das Aktorgehäuse aus Weicheisen abgeben soll. Diese Wärmeübertragung soll durch Wärmeleitpaste beziehungsweise -kleber unterstützt werden. Diese Lösung erspart weitere externe Kühlkörper. Diese Methode hat die Einschränkung, dass nur die aluminiumfreie Seite der Platine bestückt werden darf, ansonsten müssen die betroffenen Bereiche in der Aluminiumplatte ausgespart werden. Da die Platine mit THT-Bauteilen teils auch auf der Unterseite verlötet werden musste, wurden die nötigen Aussparungen durch Bohrlöcher realisiert.

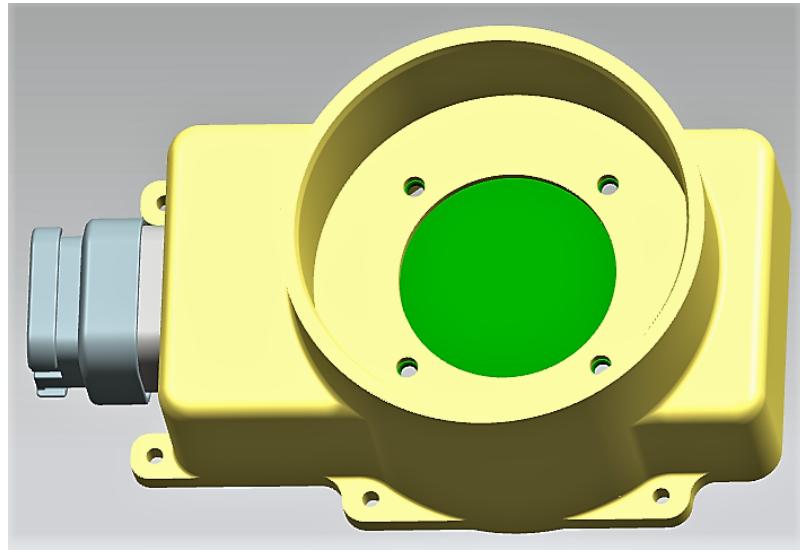


Abbildung 5.2: Rückseite des Elektronikgehäuses zur Darstellung der Aktoranbindung

Die vier Schrauben, mit denen Platine, Aluminiumplatte und Platinengehäuse am Aktorgehäuse verschraubt werden, sorgen zusätzlich für einen optimalen Wärmeübergang von Platine bis zum Aktor. Diese vier Schraubverbindungen plus zwei weitere Schraubverbindungen, welche Platine, Aluminiumplatte und Platinengehäuse verbinden, sorgen außerdem für eine großflächige GND-Verbindung über die Aluminiumplatte. Die Auflageflächen der Schrauben sind jeweils auf GND-Potential gesetzt, sodass Schrauben und Platte auch dieses Potential annehmen. Eine letzte Sache, die es bei Platinengehäusen zu beachten gibt, ist die notwendige Abschirmung gegen Eintreten sowie Abstrahlung elektromagnetischer Strahlung. Das Kunststoffgehäuse ist dabei weniger geeignet als die Aluminiumplatte, die aufgrund der elektrischen Leitfähigkeit eine gute Basisschirmung besitzt [bopla]. Für Kunststoffgehäuse wird im Allgemeinen eine Beschichtung aus Aluminium oder Kupfer empfohlen [Gwinner2006].

5.2 Bauteilplatzierung

Beim Layout-Entwurf müssen mehrere Sachen berücksichtigt werden: Die Platinenmaße und die Platzierung der GND-Schrauben müssen eingehalten werden und die Lokalität der Bauteile sollte gut durchdacht sein. Beispielsweise sollten die Halbbrücken nahe an der Spannungsversorgung platziert werden, um lange leistungsführende Leitungen zu verhindern. Weiterhin müssen die Leitungen der sensiblen Sensorik gegen elektromagnetische Rückwirkung geschützt werden. Beim engültigen Platinenentwurf, welcher in Abbildung 5.3 zu sehen ist, ist die Lokalität der Leistungs-Bauteile, also der Halbbrücken mit den Bezeichnungen HB_1 und HB_2, klar von der Platzierung der Logikebene getrennt [emcdes]. Die Halbbrücken befinden sich auf der rechten Hälfte der Platine, nahe des Steckers, sodass lange Leistungsleitungen vermieden werden können. Der Klemmblock (*terminal block*) als Anschlussklemme für den Tauchspulenaktor ist ebenfalls nahe der Halbbrückenausgänge platziert, sodass auch die Ausgangsleitungen geringe Leitungslängen aufweisen. In der Grafik ist dieser Block mit der Bezeichnung TB_1 gekennzeichnet. Der Kondensator zum Stabilisieren der Batteriespannung ist im Bild als C13 gekennzeichnet und ebenfalls nahe der Spannungsversorgung platziert. Aus Platzgründen mussten auch die Klemmblöcke für Temperatur- und externe Strommessung (J2 und J3) auf der rechten Platinenhälfte platziert werden. Da sich die leistungsführenden Leitungen jedoch auf die Mitte der Platine konzentrieren stellt dies kein Problem dar. In der linken Platinenhälfte befindet sich der Logikteil, welche den Mikrocontroller (STM32), den CAN-Transceiver (CAN_T), den externen Taktgeber (QUARZ) und den Leitungsverstärker (CD74HCT) umfasst. Neben diesen Bauteilen befindet sich ebenfalls die Spannungsregler für 3.3 V und 5 V, sodass bei der Spannungsversorgung für die ICs keine Probleme durch elektromagnetische Rückwirkungen auftreten und die Versorgungsleitungen zu den Bauteilen kurz gehalten werden kann. Bei der Platzierung der Bauteile spielt ebenfalls die Ausrichtung des Mikrocontrollers eine wichtige Rolle, sodass die benötigten Pins des Mikrocontrollers möglichst geringe Entferungen zum jeweiligen Bauteil aufweisen sodass die Signalleitungen sich selten kreuzen und von geringer Länge sein können [emcdes]. So befinden sich beispielsweise die Ausgangspins CAN1_RX und CAN1_TX des Mikrocontrollers in Abbildung 5.3 auf der oberen Seite des STM32 und die Leitungen zum CAN-Transceiver (CAN_T) können deshalb leicht verlegt werden. Die Leitungen für die H-Brücke sind auf der rechten Seite des Mikrocontrollers platziert, sodass sich diese leicht zum Leitungstreiber (CD74HCT) legen lassen. Die Pins für den externen Quarz befinden sich an der unteren Seite des STM32 und somit nahe des Schwingquarzes. Gerade bei diesem Bauteil ist die Entfernung zum Mikrocontroller möglichst gering zu halten und eventuelle Leitungskapazitäten zu verhindern [stmquarz]. Ebenfalls sollte der Taktgeber möglichst weit von den Halbbrücken entfernt platziert sein um

Störungen durch Hochfrequenzsignalen vorzubeugen [**stmquarz**]. Die Entkopplungskondensatoren der ICs sollten möglichst nahe an den Versorgungseingängen platziert werden, sodass beispielsweise die Entkopplungskondensatoren C10 und C11 nahe des VDD Pins des Mikrocontrollers (STM32) liegen [**emcdes**].

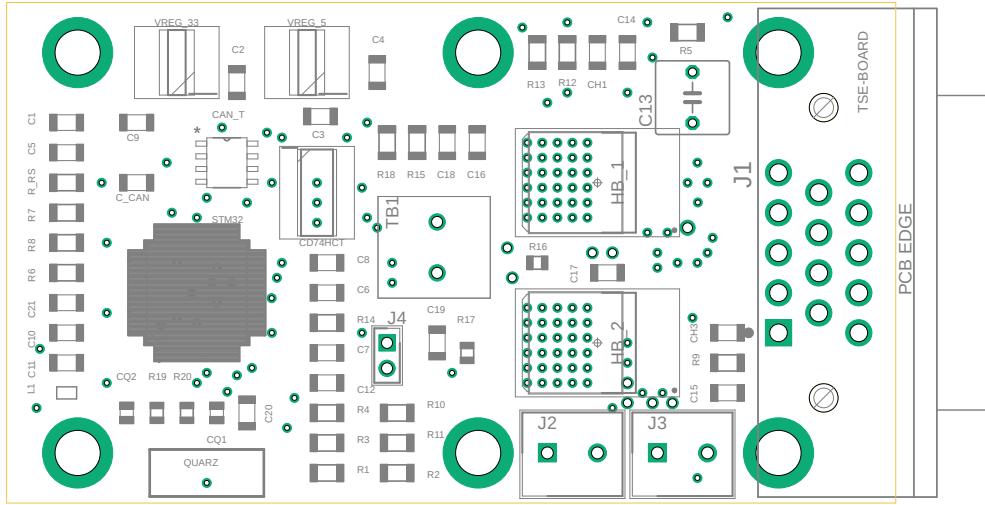


Abbildung 5.3: Platzierungsgrafik der Bauteile

5.3 Dimensionierung der Leiterbahnen

Um die Leiterbahnbreite für die Halbbrücken berechnen zu können, müssen die besonderen Gegebenheiten eines Schaltvorgangs beachtet werden. Es handelt sich um Kurzzeitbelastungen im zeitlichen Rahmen von maximal 100 ms mit einer Stromstärke von maximal 60 A. Aufgrund der Kurzzeitbelastung können die Leitungsquerschnitte deutlich geringer gewählt werden, als die Strombelastbarkeit nach der Norm IPC 2221 es für Leiterbahnen unter Dauerlast zulässt. In einem stromdurchflossenen elektrischen Leiter entsteht Verlustleistung gemäß

$$P = I^2 R$$

da Energie aufgrund des Leitungswiderstandes

$$R = \rho \frac{l}{A} (1 + \alpha_r (T(t) - T_r))$$

dissipiert. Dieser Leitungswiderstand hängt sowohl von den materialabhängigen Konstanten ρ (spezifischer Widerstand) und α_r (Temperaturkoeffizient zur Referenztemperatur T_r), als auch von der Leiterlänge l , dem Leiterquerschnitt A und der Temperaturänderung $T(t)$ zur Referenztemperatur T_r ab. Aufgrund der Erwärmung der Leiterbahn durch die Verlustleistung ist die Temperaturänderung zeitabhängig. Nach dem Stromwärmegesetz gilt für die Ableitung der Wärmeenergie nach der Zeit

$$\frac{\partial Q_W}{\partial t} = P = I^2 \rho \frac{l}{A} (1 + \alpha_r (T(t) - T_r)). \quad (5.1)$$

Die Änderung der Wärmeenergie führt zu einer Änderung der Temperatur welche sich mittels Wärmekapazität C beschreiben lässt

$$Q_W = (T(t) - T_r) C_W, \quad (5.2)$$

wobei sich die Wärmekapazität aus dem Produkt der materialabhängigen spezifischen Wärmekapazität c und der Masse m des Körpers zusammensetzt

$$C_W = cm. \quad (5.3)$$

Daraus folgt für die zeitliche Ableitung von Gleichung 5.2 die Differentialgleichung

$$\frac{\partial Q_W}{\partial t} = \frac{\partial ((T(t) - T_r)C_W)}{\partial t} = \frac{\partial T(t)}{\partial t} C_W. \quad (5.4)$$

Durch Einsetzen von Gleichung 5.1 in Gleichung 5.4 und der Notationsvereinfachung

$$\xi = I^2 \rho \frac{l}{A} (\alpha_r T_r - 1)$$

ergibt sich die Differentialgleichung erster Ordnung

$$T(t) I^2 \rho \frac{l}{A} \alpha_r - \frac{\partial T(t)}{\partial t} C = \xi. \quad (5.5)$$

Die Differentialgleichung lässt sich mittels Exponentialansatz und der Anfangsbedingung $T(0) = T_r$ lösen, sodass sich eine Funktion der Temperatur in Abhängigkeit der Zeit ergibt

$$T(t) = \left(T_r - \frac{\xi}{I^2 \rho \frac{l}{A} \alpha_r} \right) e^{\frac{I^2 \rho \frac{l}{A} \alpha_r}{C_W} t} + \frac{\xi}{I^2 \rho \frac{l}{A} \alpha_r}$$

woraus durch Einsetzen von ξ , C_W und $m = \rho l A$ folgt:

$$T(t) = \left(T_r - \frac{\alpha_r T_r - 1}{\alpha_r} \right) e^{\frac{I^2 \rho \alpha_r}{A^2 \rho c} t} + \frac{\alpha_r T_r - 1}{\alpha_r} \quad (5.6)$$

Für den Anwendungsfall der Platinenauslegung wird jedoch der minimale Leiterquerschnitt bei maximal zulässiger Temperaturänderung benötigt, sodass die Gleichung umgestellt nach dem Leiterquerschnitt A

$$A = I \sqrt{\frac{\rho \alpha_r t}{c \varrho \ln\left(\frac{T(t) - \frac{\alpha_r T_r - 1}{\alpha_r}}{T_r - \frac{\alpha_r T_r - 1}{\alpha_r}}\right)}} \quad (5.7)$$

lautet. Mit Gleichung 5.7 lässt sich für eine bestimmte Belastungsdauer t_i und der maximal zulässigen Endtemperatur der Platine $T(t_i)$ für einen beliebigen Stromfluss I berechnen. Die Herleitung dieser Formel basiert auf der Annahme, dass die Leiterbahn keinen thermischen Austausch mit ihrer Umwelt erfährt und sich somit theoretisch ohne Begrenzung erhitzt kann. Nach dem Energieerhaltungssatz wird die Leiterbahn durch ihren Umgebungsraum und ihre Umgebungstemperatur begrenzt. Unter der Annahme, dass die Umgebungstemperatur durch Nutzung des Aktorgehäuses als Kühlkörper deutlich geringer ist als die maximal zulässige Leiterplattentemperatur ist die Leitungstemperaturabschätzung durch den Wärmeaustausch geringer als in Gleichung 5.6 berechnet. Der Leitungsquerschnitt nach Gleichung 5.7 ist somit überdimensioniert und kann gut als minimale Abschätzung genutzt werden.

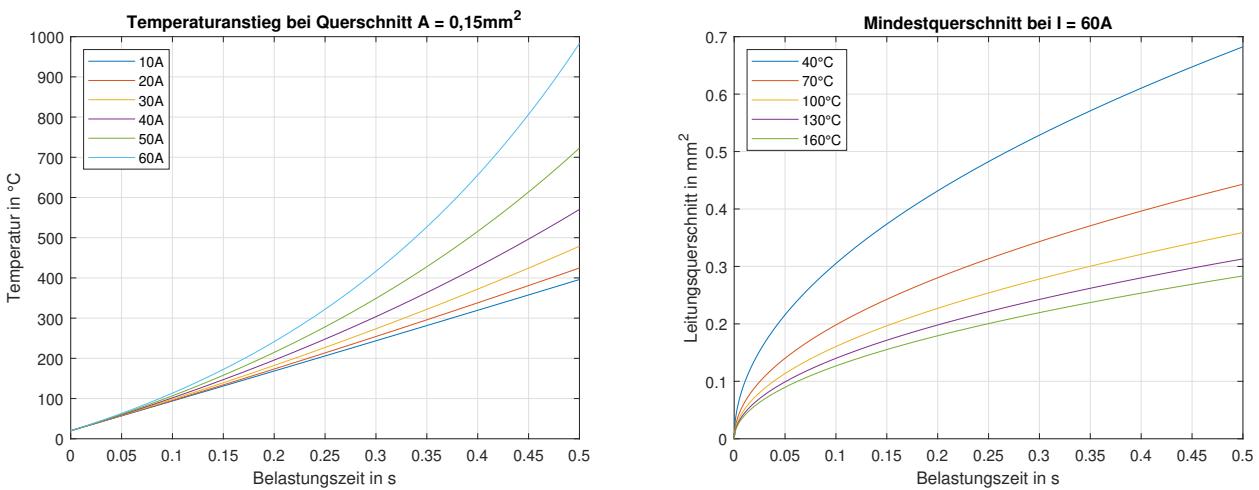


Abbildung 5.4: Dimensionierungsauswirkungen Temperatur und Leitungsquerschnitt

In Abbildung 5.4 ist in der linken Abbildung der Temperaturanstieg über die Belastungsdauer bei verschiedenem Stromfluss und konstantem Querschnitt zu sehen. Es lässt sich erkennen, dass die Temperatur stärker über die Zeit ansteigt, je höher der Stromfluss über die Leiterbahn ist. Weiterhin ist der Steigungsverlauf exponentiell. In der rechten Abbildung ist der minimale Leitungsquerschnitt über die Belastungszeit bei konstantem Strom und verschiedenen zulässigen Maximaltemperaturen zu sehen. Der Verlauf des Leitungsquerschnittsanstiegs ist logarithmisch und bei niedrigeren zulässigen Leitungsmaximaltemperaturen steigt der benötigte minimale Leitungsquerschnitt.

5.3.1 Anwendung auf realer Platine

Die Kenndaten für die minimale Dimensionierung und die Kennwerte des Platinenmaterials sind in Tabelle 5.1 beschrieben. Als Maximaltemperatur wurde dabei der Wert 80 °C gewählt. Der FR4-Kern der Platine ist nach Angaben des Herstellers für eine Temperatur bis zu 135 °C ausgelegt. Die lineare Temperaturabhängigkeit des Kupfers gilt annähernd im Bereich von -50...150 °C [Stiny2015]. Weiterhin ist die Anforderung an die Bauelemente, dass sie einen Temperaturbereich von mindestens -40...105 °C abdecken. Um sich innerhalb dieser Bereiche zu bewegen und bei Temperaturausschlägen einen Sicherheitsfaktor zu berücksichtigen, sind die Leiterbahnen für die Maximaltemperatur von 80 °C auszulegen. Nach der Tabelle ist für eine Kupferschichtdicke von 70 µm eine Leiterbahnbreite von mindestens 2.61 mm zu wählen. Beim Platinendesign ist diese aus Sicherheitsgründen zu 3 mm gewählt, was einem maximalem Stromfluss von etwa 69.1 A entspricht. Die Dimensionierung betrifft die Spannungsversorgungsleiterbahnen von der Batterie zu den Halbbrücken und die Leiterbahnen von den Halbbrücken zum Aktorklemmblock. Die restlichen Leiterbahnen auf der Platine sind für Dauerlast nach IPC 2221 auszulegen. Im Anhang findet sich ein Tabellenauszug nach IPC 2221 für die Strombelastbarkeit und eine Tabelle der Leistungsdaten der Platinenelemente, nach der die Leiterbahnen dimensioniert sind.

Kenngröße	Zeichen	Wert	Einheit
Temperaturkoeffizient Cu (20°C)	α_{20}	$3,93 \cdot 10^{-3}$	K^{-1}
Spez. Widerstand Cu	ρ	$1,721 \cdot 10^{-2}$	$\Omega \cdot mm^2/m$
Spez. Wärmekapazität Cu	c	385	J/K
Dichte Cu	ϱ	8,92	kg/m^3
Referenztemperatur	T_{20}	20	°C
Maximalstrom	I_{max}	60	A
Belastungsdauer	t_i	0,1	s
Maximaltemperatur	$T(t_i)$	80	°C
Kupferschicht Dicke	d	$70 \cdot 10^{-6}$	m
resultierender Querschnitt	A	0,1824	mm^2
resultierende Leiterbahnbreite	b	2,6059	mm

Tabelle 5.1: Kenndaten Platine mit Kupferschicht

5.4 Routen der Platine

Bei der letztendlichen Verlegung (*routing*) der Leiterbahnen gibt es einige Dinge zu beachten, die einerseits Einfluss auf die Elektromagnetische Verträglichkeit (EMV), andererseits auch auf die Komplexität des Routingvorgangs haben.

5.4.1 Design-Regeln und EMV

Das Verlegen langer Leitungen sollte nach Möglichkeit vermieden werden um potentielle Störgrößen wie elektromagnetische Induktion und kapazitives Leitungsverhalten zu verringern [haendschke]. Wie bereits in Abschnitt sec:bau erwähnt, sollten deshalb eine gewisse Lokalität der Bauteile eingehalten werden. Um die Leiterbahnen verlegen zu können sollten die Leiterplattenebenen (*Layer*) abwechselnd verschiedene präferierte Verlegerichtungen aufweisen, sodass die Signale leicht an jeden Punkt der Platine verlegt werden können. In Abbildung 5.5 ist schematisch der Aufbau einer zwei Layer Platine dargestellt. Im Falle einer gleichen Verlegerichtung kann es auftreten, dass sich die Signalleitungen auf keinem der vorhandenen Layer kreuzen können und somit unnötig lange Leitungen verlegt werden müssen [haendschke]. Signale können zwischen den Platinen-Layern über Durchkontaktierungen (*vias*) wechseln. Durchkontaktierungen sollte man jedoch so wenig wie möglich nutzen, da diese induktives Leitungsverhalten bewirken. In der Abbildung ist diese metallische Durchkontaktierung zwischen Layer 1 und Layer 2 exemplarisch dargestellt. Eine Signalleitung in Layer 1 kann, wenn sie mit der Metallfläche der Via verbunden ist, in Layer zwei weiter verlegt werden.

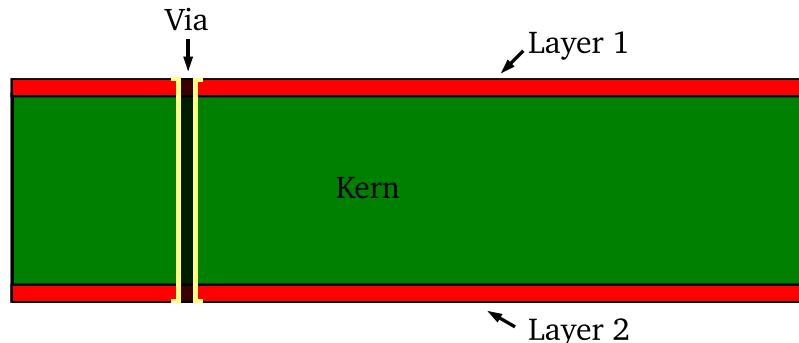


Abbildung 5.5: Darstellung einer zwei Layer Platine inklusive Via

Prinzipiell können Platinen mit deutlich mehr Layer-Ebenen entworfen werden, zwei Layer Platinen werden jedoch bei industriellen Endprodukten aufgrund ihrer geringen Kosten präferiert [emcdes]. Im endgültigen Platinendesign wurde sich deshalb auch für eine 2 Layer Platine entschieden. Beim Verlegen der Leiterbahnen sollte weiterhin darauf geachtet werden, dass möglichst keine Winkel über 45 Grad auftreten, da diese unerwünschte Reflexionen hervorrufen können [emcdes]. Bei wichtigen Signalleitungen sollte darauf geachtet werden, dass der Abstand zu anderen Leitungen größtmöglich ist, um eine Rückwirkung der Leiter aufeinander zu minimieren. Zur Verbesserung der Schirmung der Signalleitungen gegeneinander wird häufig eine GND-Leitung zwischen den Signalleitungen verlegt, welche die Rückwirkung verringert [Franz2012]. Wenn diese Grundregeln der Leiterbahnanplanung bekannt ist, gibt es nach [emcdes] drei Schritte, die zu einer fertigen Platine führen:

- Komponentenauswahl und Platzierung (siehe Kapitel 4 und Abschnitt 5.2)
 - Entwurf des Versorgung- und Entkopplungskonzepts
 - Verlegen der Signalleitungen

Nachdem in Abschnitt 5.2 die Bauteilplatzierung erläutert wurde, ist in Abbildung 5.6 der erste Entwurf der Logikebene der Platine mit Versorgungskonzept und verschalteten Abblockkondensatoren zu sehen. Die Versorgungsleitungen werden dabei vor den Signalleitungen verlegt, da diese je nach Leistungsführung in ihrer Breite angepasst werden müssen. Die Logikebene beschränkt sich in dem Entwurf auf den STM32 und einen CAN-Transceiver mit Pin-Ausgängen zur Ansteuerung der H-Brücke. Im dritten Schritt werden dann die Signalleitungen verlegt. Die Abbildung ist nur exemplarisch erstellt und wurde nie in Auftrag gegeben. Sie soll lediglich eine Nachvollziehbarkeit der Herangehensweise bewirken.

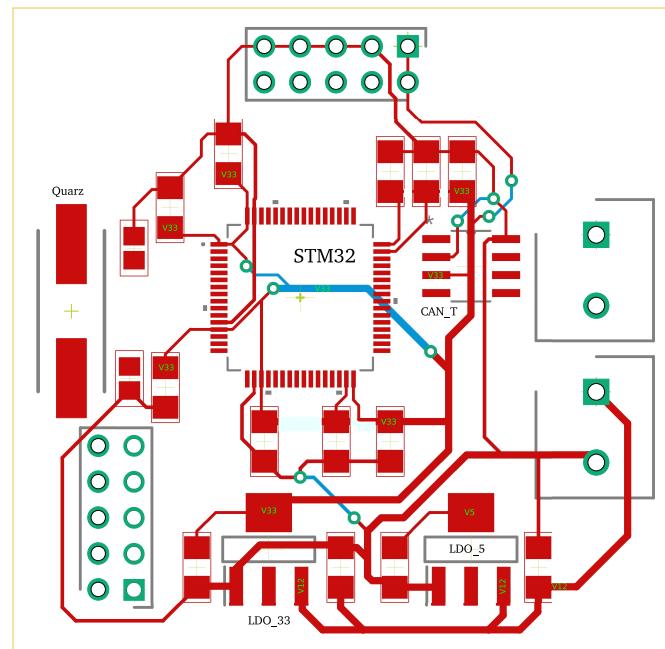


Abbildung 5.6: Logikboard zur Endplatine inklusive Versorgungsleitungen

Üblicherweise werden zur Verbesserung der EMV nach dem Verlegen der Leiterbahnen die freien Platinenflächen zur Abschirmung auf GND Niveau gesetzt, welche als Teilmassenflächen bezeichnet werden. In der einschlägigen Literatur wird darauf hingewiesen, dass diese mit Vorsicht behandelt werden müssen, da Resonanz- und Antenneneffekte auftreten können. Um diesen Effekten vorzubeugen müssen die Masseflächen so gut wie möglich miteinander verbunden werden, sodass diese bei richtigem Anschluss die Schirmungsfunktion übernehmen [Franz2012]. Bei einem Layout mit 4 oder mehr Layern wird empfohlen vollständige GND-Layer und Versorgungslayer einzuführen, sodass eine optimale Abschirmung erfolgen kann [Franz2012]. Ein weiterer Effekt der im Bezug auf GND-Flächen betrachtet werden sollte ist das so genannte GND-Bouncing. Dieser Effekt tritt auf, wenn man keine gute Verbindung zwischen den GND-Flächen hat und somit unterschiedliche GND-Potentiale entstehen. Verstärkt wird dieser Effekt von Switching-ICs, also aktiven Bauteilen mit Schaltcharakteristiken, welche teilweise beim gleichzeitigen Umschalten von 1 nach 0 einen Zeitpunkt aufweisen, zu dem VCC und GND kurzgeschlossen sind und sich somit das GND Potential kurz anhebt [gndbnc]. Aufgrund dieses Effektes sollte die GND-Fläche so gut wie möglich verbunden sein um das GND Potential gleichmäßig zu halten. Problematisch ist dabei auch nicht zwingend wenn es kein geschlossenes GND-Layer gibt, solange die GND-Verbindungen der Ebenen sich ausreichend überlappen und über Durchkontaktierungen gut verbunden sind [Franz2012]. Zur Abschirmung der aktiven Bauelemente gegen Welligkeiten der Spannungsversorgung werden wie in Kapitel 4 erwähnt Abblockkondensatoren verwendet. Dabei sollten möglichst SMD Bauteile genutzt werden um entstehende parasitäre Induktivitäten zu verringern [emcdes]. Als Abschirmung gegen von außen autretende elektromagnetische Felder sei die Schirmwirkung von Kühlkörpern oder Gehäuseteilen aus Metall hervorgehoben [Franz2012].

5.4.2 Leiterbahnverlegung des Entwicklungsboards

Die Verlegung der Leiterbahnen auf dem endgültigen Entwicklungsboard geschieht nach den zuvor definierten Designregeln, wobei aufgrund des geringen Platzes häufig Kompromisse eingegangen werden müssen. Das Entwicklungsboard weist 2 Layer auf, welche der Anordnung aus Abbildung 5.5 entsprechen und im Folgenden über die Bezeichnung Top-Layer (obere Leiterebene) und Bottom-Layer (untere Leiterebene) gekennzeichnet werden. Grundanforderungen an die Platine ist die Positionierung der GND-Löcher, durch welche die Platine mittels Schrauben über das Aktorgehäuse mit GND verbunden ist. In Abbildung 5.7 und Abbildung 5.8 ist das fertige Top-und Bottom-Layer der Platine zu sehen. Problematisch für das Einhalten der Designregeln ist hauptsächlich die Platzierung der Halbbrücken und der mittleren Schraubenlöcher, welche den Platz für die Signalleitungen von dem Mikrocontroller in Richtung Platinenstecker stark begrenzen. Die präferierte Verlegerichtung auf dem Top-Layer ist vertikal gewählt um die jeweiligen ICs über kurze Wege ohne große Leitungsverluste erreichen zu können. Dabei liegen CAN-Transceiver und Leistungsverstärker oberhalb und der externe Schwingquarz unterhalb des Mikrocontrollers, was optimal für die präferierte Leitungsrichtung ist. Problematisch ist die Leitungsverlegung in Richtung der Halbbrücken und des Steckers, da der Spalt zwischen Schraubloch und Halbbrücke in der oberen Platinenhälfte durch die Versorgungsspannung belegt ist.

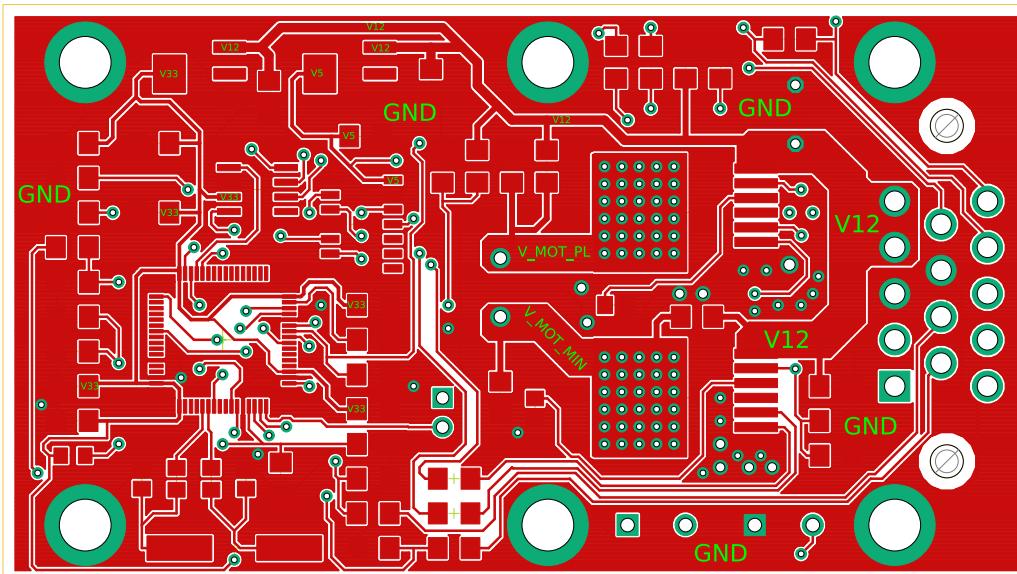


Abbildung 5.7: Darstellung des Top-Layers

Die präferierte Verlegerichtung des Bottom-Layers ist horizontal ausgerichtet, sodass diese Leiterebene hauptsächlich zum Erreichen der Halbbrücken und des Steckers genutzt wird. Bei Betrachtung von Abbildung 5.8 fällt jedoch auf, dass der Platz für die Leitungen sehr begrenzt ist und somit ein Teil der horizontalen Leiterbahnen auf das Top-Layer verlegt werden musste. Eine weitere wichtige Aufgabe beim Verlegen der Leiterbahnen ist die Reduktion elektromagnetischer Rückwirkung, wobei sich an die Methoden aus Unterabschnitt 5.4.1 gehalten wird. Potentielle Gefahrenquelle der Rückstrahlung sind die Halbbrücken, da diese beim Schalten mit PWM-Signal Stromimpulse in hohen Frequenzen durchschalten können. Wichtige Signalleitungen wie die der Sensorik sind also möglichst weit von den Flächen mit der Bezeichnung V_MOT_PL und V_MOT_MIN (Halbbrücken Ausgänge) zu verlegen. Aufgrund des Platzmangels lassen sich die Signalleitungen der Sensorik nicht vollständig aus dem Einflussgebiet der Halbbrücken entfernen, sind jedoch von diesen so weit wie möglich entfernt verlegt. Bei genauerer Betrachtung der Layer fällt auf, dass jedoch trotzdem Signalleitungen nahe des Einflussgebietes der Halbbrücken verlaufen. Diese Leiterbahnen sind hauptsächlich die Leiterbahnen der SWD-Verbindung, also der Programmierschnittstelle der Platine. Daher stellt dies kein Problem dar, da die Platine nicht im laufenden Aktorbetrieb programmiert wird und somit die Lokalität im Einflussgebiet der Halbbrücken kein Hindernis darstellt. Die Leitungen zur Ansteuerung der Halbbrücken HB1_IN und HB2_IN sollten ebenfalls nicht im direkten Einflussgebiet liegen, damit die Halbbrücken vom Mikrocontroller noch problemlos angesteuert werden können. Als Konzept zur Abschirmung elektromagnetischer Rückwirkung wurde bereits der Einsatz von Abblockkondensatoren erwähnt, der eine wesentliche Rolle spielt. Neben diesen sind die freien Boardflächen auf beiden Layern auf GND-Potential gesetzt um damit ebenfalls Schirmungseffekte erzielen zu können. Diese sind zwar auf beiden Layern nicht optimal zusammenhängend, es wurde aber darauf geachtet über Durchkontaktierungen eine stabile gemeinsame GND-Fläche zu erhalten, welche gegen GND-Bouncing geschützt ist. Die GND-Flächen werden ebenfalls über die Aluminiumplatte und den Aktor auf ein gleichmäßiges Potential gezogen.

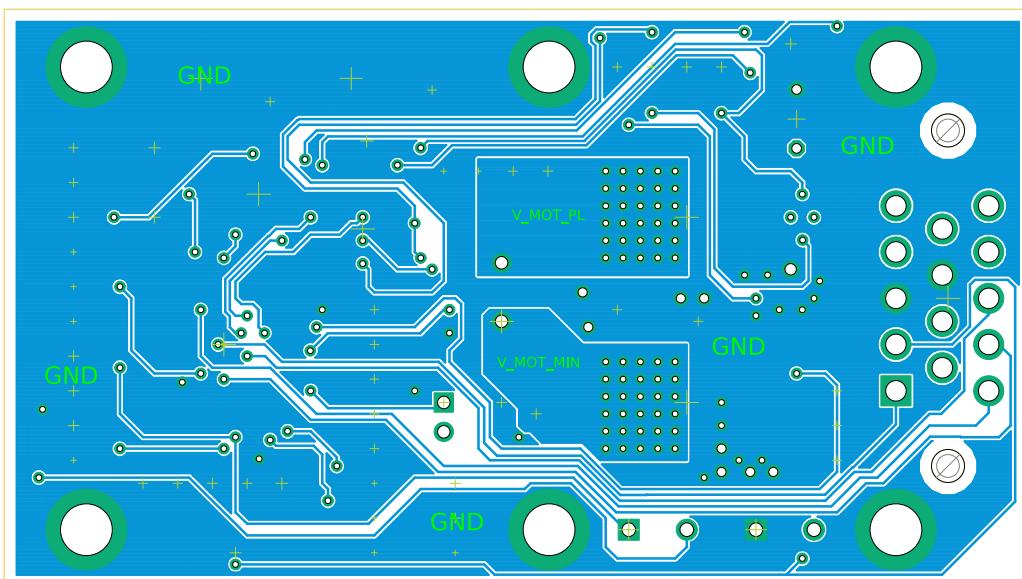


Abbildung 5.8: Darstellung des Bottom-Layers

Neben den EMV-Maßnahmen sollte auch die Wärmeentwicklung der Bauteile berücksichtigt werden. Aufgrund des hohen Stromflusses stellen die Halbbrücken besondere Anforderungen an die Kühlung. Insbesondere in Transistorschaltvorgängen innerhalb der Halbbrücke während denen ein Teil der Versorgungsspannung an den Transistoren abfällt und somit nach $P = U \cdot I$ erhöhte Wärmeenergie entsteht. Um die Halbbrücken unter ihrer kritischen Temperatur halten zu können, muss die entstehende Wärme abgeführt werden. Dies kann beispielsweise durch Durchkontaktierungen (*Thermische Vias*) unter den kritischen Bauteilen erreicht werden. Über die Thermischen Vias vergrößert sich die zusammenhängende Metallfläche auf der sich die Wärme ausbreiten kann in das Bottom-Layer, und kann somit gut über den anliegenden Aluminium-Körper abgeführt werden.

6 Software



7 Analyse und Performance

7.1 Beurteilung der Anforderungserfüllung

7.2 Kostenaufstellung

Die Kosten für die fertige Platine inklusive aller Bauteile belaufen sich bei einer einzigen Platine auf 80 Euro. Bei einer Stückzahl von 100 kann der Preis bereits auf 33,5 Euro gesenkt werden, während die Bauteilkosten bei einer Fertigung von 1000 Platinen nochmal auf knapp unter 25 Euro sinken. Diesen großen Preisunterschied verursacht vor allem die unbestückte Platine selbst, die bei Bestellung von einer einzigen 38,5 Euro kostet und bei einer Bestellung von 1000 Stück nur noch 0,82 Euro. Tabelle 7.1 zeigt die verwendeten Bauteile und deren Anzahl sowie den kumulierte Preis pro Bauteilart (Anzahl des Bauteils multipliziert mit dem Einzelpreis) für jeweils eine Fertigung von einer Platine, von 100 Platinen und von 1000 Platinen. Die Preise stammen dabei von den Anbietern, bei denen die Komponenten jeweils eingekauft wurden.

Anzahl	Bauteil	Bauteilpreis	Bauteilpreis 100+	Bauteilpreis 1000+
1	Platine	38.5 Euro	2.39 Euro	0.82 Euro
1	Mikrocontroller	10.66 Euro	7.79 Euro	6.58 Euro
2	Halbbrücken	6.56 Euro	5.96 Euro	5.04 Euro
1	Leitungstreiber	0.617 Euro	0.348 Euro	0.252 Euro
1	CAN Transceiver	2.82 Euro	2.28 Euro	1.51 Euro
1	Voltage Regulator 3,3V	0.337 Euro	0.162 Euro	0.103 Euro
1	Voltage Regulator 5V	0.337 Euro	0.162 Euro	0.103 Euro
1	Klemmblock	1.16 Euro	1.07 Euro	0.912 Euro
2	Klemmblock	0.742 Euro	0.618 Euro	0.526 Euro
1	Steckverbinder	0.0442 Euro	0,0442 Euro	0,0387 Euro
1	AMPSEAL Automotive Steckverbinder	7.09 Euro	6.22 Euro	5.23 Euro
1	Quarz	0.605 Euro	0.355 Euro	0.384 Euro
21	Widerstände	4.88 Euro	3.046 Euro	1.6754 Euro
21	Kondensatoren	5.907 Euro	3.0648 Euro	1.501 Euro
	Gesamtpreis	80,26 Euro	33,51 Euro	24,68 Euro

Tabelle 7.1: Preisliste

Die gesamte Preisaufstellung für die Stückzahlen eins, 100 und 1000 inklusive der einzelnen Widerstände und Kondensatoren, sowie die Händlerlinks zu allen Bauteilen befindet sich im Anhang. In nachfolgenden Abbildungen wird die Verteilung der Kosten auf die verschiedenen Bauteilgruppen dargestellt. Die Bauteile wurden unterteilt in die Platine, die passiven Bauteile (Kondensatoren, Widerstände und Schwingquarz), die integrierten Halbleiterchips (Mikrocontroller, Spannungsregler, CAN Transceiver, Leitungstreiber, Halbbrücken) sowie die Stecker, die die Schnittstellen nach außen darstellen.

Es ist zu erkennen, dass bei geringen Stückzahlen die Platine ungefähr die Hälfte der Kosten ausmacht, während sie bei hohen Stückzahlen fast gar nicht mehr ins Gewicht fällt. Bei hohen Stückzahlen sind der größte Kostenfaktor die ICs.

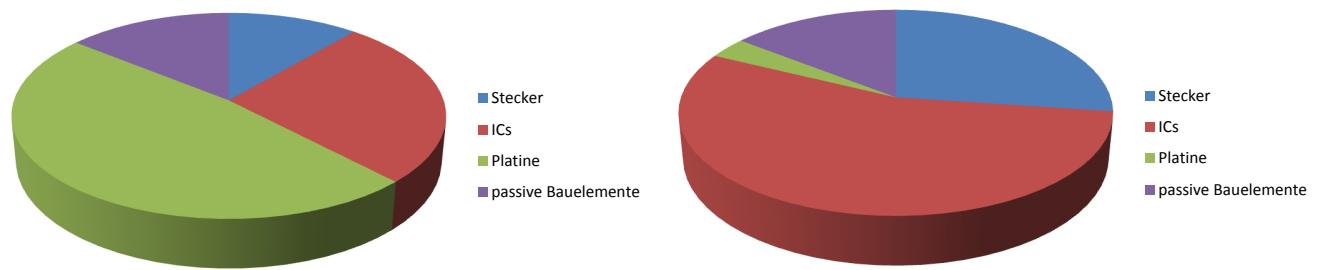


Abbildung 7.1: Aufteilung der Kosten für die Stückzahlen 1 (links) und 1000 (rechts)

8 Fazit und Ausblick



Abbildungsverzeichnis

2.1 Prüfstand [adp]	7
2.2 Fahrzeuggetriebe	8
2.3 Querschnitt Tauchspulenaktor [Hahn2018]	8
2.4 Kraft-Weg-Strom Kennlinien des Tauchspulenaktors [adp]	9
2.5 Einbauposition PLCD Sensor an der Schaltgabel [adp]	10
2.6 Sprungantwort Schaltgabelposition [adp]	11
2.7 Blockschaltbild eines typischen Mikrocontrollers [Bernstein2015]	12
2.8 Aufbau CAN-Bus [manual]	13
2.9 beispielhaftes PWM Signal	14
3.1 V-Modell nach [Boehm 79]	16
3.2 Klassifikationsbaum am Beispiel der H-Brücke	17
3.3 Bildliche Darstellung des Prototypings (Iterationsprozesses)	18
4.1 Aktive Bauelemente	20
4.2 Minimalbeschaltung des STM32F405RGT7	20
4.3 Pin-Belegung des Mikrocontrollers	21
4.4 Anschluss externer Quarz	22
4.5 CAN-Transceiver Verschaltung	22
4.6 Blockschaltbild LDO	23
4.7 Schaltplan Spannungsversorgung	24
4.8 Anschlusspins Stecker	24
5.1 Derzeitiger Entwurf des Elektronikgehäuses	27
5.2 Rückseite des Elektronikgehäuses zur Darstellung der Aktoranbindung	28
5.3 Platzierungsgrafik der Bauteile	29
5.4 Dimensionierungsauswirkungen Temperatur und Leitungsquerschnitt	30
5.5 Darstellung einer zwei Layer Platine inklusive Via	32
5.6 Logikboard zur Endplatine inklusive Versorgungsleitungen	32
5.7 Darstellung des Top-Layers	33
5.8 Darstellung des Bottom-Layers	34
7.1 Aufteilung der Kosten für die Stückzahlen 1 (links) und 1000 (rechts)	38