

Fachbericht

PROJEKT 5 COCKTAILMASCHINE - TEAM SCHENK & AEBI
19. Januar 2020

Betreuer Dozent:	Prof. Dr. Schleuniger, Pascal
Team:	Schenk, Kim Aebi, Robin
Studiengang:	Elektro- und Informationstechnik
Semester:	Herbstsemester 2019

Abstract

In diesem Projekt wurde ein Konzept erstellt, um eine Cocktailmachine zu bauen. Dies reicht von der Analyse, was für Cocktailmaschinen es bereits gibt, über die Erstellung eines Grob- und eines Detailkonzeptes bis hin zur Evaluation der Komponenten. Der Aufbau wurde so gewählt, dass ein Glas mittels eines Linearantriebes auf einem Schlitten hin- und her gefahren wird und unter dem gewünschten Flüssigkeitsauslass stehen bleibt, wo es dann befüllt wird. Die Bedienung soll über ein Touch-Display geschehen. Die Verarbeitung der Daten wird ein Mikrocontroller übernehmen. Als mechanische Komponente wird pro Zutat eine Pumpe und ein Durchflusssensor verwendet sowie ein einzelner Motor, welcher den Linearantrieb mit dem Schlitten betreibt. Als Motor wurde ein bürstenloser Gleichstrommotor verwendet, da dieser ein sehr gutes Leistungs-/Gewicht-Verhältnis aufweist und in seiner Ansteuerung sehr interessant ist. Ziel des Projekt 5 war es, anhand des Konzeptes die einzelnen Teilsysteme aufzubauen und deren Funktion zu verifizieren und zu dokumentieren. Softwaremässig wurde die Basis für den Mikrocontroller geschrieben. Dies bedeutet, dass die Teilsysteme kontrollierbar sind und im Projekt 6 ausgebaut und zusammengeführt verwendet werden können. Die Software wurde komplett in C geschrieben und ausgiebig dokumentiert. Das Resultat zeigt, dass die Komponenten zusammenpassen und der Cocktailmachine im Projekt 6 nichts im Weg steht.

Inhaltsverzeichnis

1 Einleitung	1
2 Recherche	2
2.1 Aufbau bestehender Cocktailmaschine	2
2.1.1 Spirits	2
2.1.2 Der Cocktailmixer	4
2.1.3 CocktailAvenue	5
2.1.4 myRocktail	6
2.2 Entscheidungfindung des Aufbaus	8
3 Grobkonzept	10
3.1 Blockschaltbild	10
3.2 Motor	12
3.2.1 Aufbau	12
3.2.2 Funktionsprinzip	13
3.3 Resolver	14
3.3.1 Aufbau	15
3.3.2 Funktionsweise	15
3.4 TMC4671-EVAL Board	16
3.4.1 Aufbau	16
3.4.2 FOC	17
3.4.3 Park & Clarke	18
3.4.4 PID-Regler	19
3.5 Förderband	21
3.5.1 Anforderungen	21
3.5.2 Verwendeter Aufbau	21
3.6 Pumpen	22
3.6.1 Anforderungen	22
3.6.2 Mögliche Pumpen	22
3.6.3 Verwendete Pumpe	23
3.7 Durchflussmessung	24
3.7.1 Anforderungen	24
3.7.2 Verwendeter Durchflusssensor	24

3.8	Display	25
3.8.1	Anforderungen	25
3.8.2	In Frage kommende Displays	25
3.8.3	Verwendetes Display	26
3.9	Mikrocontroller	26
3.9.1	Anforderungen	26
3.9.2	Total benötigte Anschlüsse	27
3.9.3	In Frage kommende Mikrocontroller	27
3.9.4	Verwendeter Mikrocontroller	27
4	Detailkonzept	28
4.1	Speisungen	28
4.1.1	48V Speisung	28
4.1.2	12V Speisung	29
4.1.3	5V Speisung	31
4.1.4	3,3V Speisung	32
4.2	Resolver Interface	32
4.2.1	Problem	32
4.2.2	Schaltungsaufbau	32
4.2.3	Simulation	35
4.3	Pumpenansteuerung	39
4.4	Durchflussmessgeräte	39
4.5	Display	40
4.6	User Interface	41
4.6.1	Startanzeige	42
4.6.2	Zutatenanzeige	42
4.6.3	Listenanzeige	43
4.6.4	Zubereitungsabfrage	43
4.6.5	Zubereitungsbildschirm	44
4.6.6	Bereitanzeige	44
4.6.7	Menuanzeige	45
4.6.8	Bearbeitungsanzeige	45
4.6.9	Cocktaileinstellungsanzeige	46
4.6.10	Reinigungsanzeige1	46

4.6.11 Reinigungsanzeige2	47
4.6.12 Reinigungsanzeige3	47
4.6.13 Infoanzeige	48
4.6.14 Abbruchanzeige	48
4.6.15 Fehleranzeige	49
5 Hardware	50
5.1 Speisungen	50
5.1.1 12V Speisung	50
5.1.2 5V Speisung	51
5.2 Motorenansteuerung	52
5.2.1 Resolver	52
5.2.2 TMC4671	56
5.3 Pumpenansteuerung	57
5.4 Durchflussmessgeräte	58
5.5 Display	58
5.5.1 Test	58
5.5.2 Erwartung	58
5.5.3 Ergebnis	59
5.6 Mikrocontroller	60
5.6.1 Systemtest	60
5.6.2 Erwartung	60
5.6.3 Ergebnis	60
5.6.4 Programmablauf	63
5.6.5 Erwartung	63
5.6.6 Ergebnis	64
6 Software Mikrocontroller	65
6.1 Programmflussdiagramm main.c	65
6.2 Dokumentation	67
6.2.1 Einlesen der Headerfiles	67
6.2.2 Präprozessor-Anweisungen	67
6.2.3 Deklarationen	68
6.2.4 Initialisierungen	68
6.2.5 Mainroutine	69

6.3	libraries	69
6.3.1	Ring Buffer	70
6.3.2	SPI	70
6.3.3	UART	71
6.3.4	TMC4671	71
6.3.5	Nextion Display	72
6.4	Lizenzen Bibliotheken	72
6.4.1	SPI	72
6.4.2	Uart	72
6.4.3	Ring-Buffer	73
6.4.4	TMC4671	73
7	Fazit	74
7.1	Was wurde erreicht und was nicht?	74
7.1.1	Speisungen	74
7.1.2	Mikrokontroller	74
7.1.3	Motor	74
7.1.4	Pumpen	74
7.1.5	Durchflussmessgeräte	74
7.1.6	Display	75
7.2	Vergleich der gesetzten Ziele mit den erreichten Zielen	75
7.2.1	Pflichtzielerfüllung	76
7.2.2	Wunschzielerfüllung	77
8	Schlusswort	78
9	Ehrlichkeitserklärung	79
Literatur		80

1 Einleitung

Eine gelungene Party auf die Beine zu stellen verlangt einem einiges ab. Vor allem kostet es eine Menge Aufwand und Zeit. Dies gilt besonders, wenn es darum geht mit vielen Freunden zusammen zu feiern. Neben der gelungenen Musikauswahl und den Snacks darf eines auf gar keinen Fall fehlen, die Getränke. Um diese sicherzustellen, gibt es mehrere Möglichkeiten. Einerseits könnte jeder seine eigenen Getränke mitbringen, was jedoch bedeutet, dass es unter Umständen eine riesige Sauerei gibt oder viele Flaschen in der Gegend rumstehen. Andererseits könnte man als Gastgeber selber anbieten Cocktails zu mixen und so den Getränkenachschub zu gewährleisten. Da gibt es jedoch ein grosses Problem. Denn wären wir die Gastgeber, so würden wir nicht den ganzen Abend hinter der Bar stehen wollen, sondern lieber bedenkenlos mitfeiern. Damit genau dies möglich ist haben wir uns in diesem und dem nächsten Projekt (5&6) dazu entschieden eine automatisierte Cocktailmaschine zu entwerfen. Diese soll vollkommen autonom arbeiten und sollte problemlos von jeder beliebigen Person und in fast jedem Zustand bedient werden können.

In den folgenden Kapiteln ist dokumentiert, wie die Cocktailmaschine aussehen soll und aus welchen Teilsystemen diese bestehen wird. Ausserdem werden die einzelnen Teilsysteme genauer unter die Lupe genommen und in einem systemspezifischen Testverfahren evaluiert. Dieses Projekt bietet demnach die Basis des Projekt 6 und soll dieses so gut wie möglich vorbereiten.

2 Recherche

Der Rechercheteil soll eine Übersicht über die erfolgte Recherche, sowie die Entscheidungsfindung geben. Diese ist essentiell für das weitere Vorgehen und entscheidet somit über das Konzept. Ohne ausführliche Recherche und Entscheidungsfindung wäre eine strukturierte Lösungsfindung nicht möglich. Es werden dabei für jedes Teilsystem mehrere Möglichkeiten geprüft und analysiert.

2.1 Aufbau bestehender Cocktailmaschine

Um mit der eigentlichen Arbeit beginnen zu können ist es wichtig, dass man sich darüber erkundigt, was es bereits auf dem Markt gibt und wie diese Cocktailmaschinen funktionieren. Aus diesen Erkenntnissen kann dann entschieden werden, welche Funktionsart umgesetzt wird und wie der Aufbau aussehen soll.

2.1.1 Spirits

Aufbau

Spirits ist eine mobile Cocktailmaschine. Vom Aufbau her hat sie einen Edelstahlkorpus, in welchem die Elektronik, die Pumpen und die Ventile verbaut sind. Um den Korpus herum können die Getränkeflaschen in eine Getränkehälterung hineingestellt werden. Insgesamt haben 8 Flaschen Platz. Diese Kapazität ist jedoch erweiterbar, indem mehrere Maschinen miteinander gekoppelt werden. Ein spezieller Flaschendeckel, durch welchen ein Röhrchen führt, schließt die Flaschen. Das Röhrchen ist dazu da, die Flüssigkeit aus den Getränkeflaschen zu pumpen. Mit einem Silikonschlauch sind die Röhrchen mit dem Korpus verbunden. Auf der Frontseite ist ein Touch-Display verbaut, über welchem sich eine Abstellmöglichkeit für ein Glas befindet. Darüber befindet sich der Flüssigkeitsauslass. Auf der Rückseite befinden sich diverse Anschlüsse wie 2x USB, 1x Ethernet, 2x Chinch R+L, 1x Artnet¹ und die Stromversorgung. [1]

Bedienung

Die Bedienung kann neben dem Display auch über ein Smartphone, ein Tablett oder einen PC geschehen. Kommuniziert wird dabei über das eigene WLAN-Netz der Spirits-Cocktailmaschine, welche somit als Accesspoint dient. Sobald die Bestellung eingegangen ist, reicht es ein mit Eis gefülltes Glas unter den Flüssigkeitsauslass zu stellen und das Getränk zur Zubereitung freizugeben.[1]

¹Artnet=Kommunikationsprotokoll für intelligente Scheinwerfer, Moving Heads und Effektgeräte (Remote Device Management)

Technische Daten

Zeit für einen Cocktail:	: Aufgrund paralleler Getränkeförderung braucht der Spirits nur ca. 20s, um 3dl eines Cocktails zu erstellen. [1]
Anzahl verschiedene Cocktails:	: Abhängig von den Mischgetränken und benötigten Zutaten. Sind mehrere Cocktails erwünscht, können im Betrieb die Zutaten getauscht werden oder mehrere Spirits miteinander verknüpft werden. [1]
Stromversorgung:	: In Spirits ist ein Akku verbaut, so dass die Maschine auch ohne Steckdose verwendet kann. Zum Aufladen des Akkus und für den Dauerbetrieb benötigt sie jedoch eine Steckdose. [1]

Reinigung

Zur Reinigung können alle Schläuche in ein mit heissem Wasser gefüllten Gefäß hineingelegt werden. Weiter muss ein leeres Gefäß unter den Flüssigkeitsauslass gestellt werden. Über die App kann nun das Reinigungsprogramm aufgerufen werden. Nun befördern die Pumpen das heiße Wasser durch die Schläuche und reinigen alles. Nach dem Vorgang die Schläuche aus dem Wasser nehmen und nochmals laufen lassen, damit die Pumpen das restliche Wasser aus den Schläuchen pumpen. Bevor die Reinigung jedoch gestartet wird, kann die noch in den Schläuchen befindende Flüssigkeit in die Behälter zurück gepumpt werden.[1]

Sonstiges

Spirits kann sogar Musik abspielen. Es ist möglich, eigene Playlists zu erstellen und zu beeinflussen. Die Musik kann auf USB-Sticks, auf externen Festplatten oder auf dem internen Speicher abgelegt und aufgerufen werden. Mittels Voting können die Anwesenden den Musikverlauf steuern. Auch Lichteffekte sind mit Spirits möglich. So können DMX²-Geräte über das Artnet-Protokoll gesteuert werden. Diverse Lichtprogramme sind schon hinterlegt. Weitere Lichtszenen können angelegt und gespeichert werden. Von Spirits gibt es drei verschiedene Ausgaben, welche sich jedoch rein darin unterscheiden, dass sie unterschiedliche Getränkehälterungen haben.[1]



Abbildung 2.1: Spirits-Cocktailmaschine [1]

²DMX=Digitales MultipleX

2.1.2 Der Cocktailmixer

Aufbau

Der Cocktailmixer ist ein 1m breites und 76cm hohes Küchengerät auf Rollen. Im Unterschrank können bis zu 28 Zutaten gelagert werden. Das Gehäuse besteht aus Edelstahl und ist sehr aufwendig und professionell gestaltet, sodass es in einer Restaurantküche einen Platz finden könnte. Die Schläuche führen im Schrank zum Flüssigkeitsauslass und sind bei geschlossenen Fronttüren nicht sichtbar. Gemäss Hersteller können 14-40 Getränkeanschlüsse realisiert werden. Die Behälter haben standardmäßig ein Volumen von 5 Liter und können bei Bedarf auch auf 10 und 20 Liter erweitert werden. Über der Oberfläche befindet sich ein einziger Flüssigkeitsauslass. Die Maschine kann in der bestehenden Theke eingebaut werden. Gesteuert wird die Maschine von einer SPS-Steuerung, welche eine hohe Verlässlichkeit ohne Ausfällen gewährleisten soll.[2]

Bedienung

Die Bedienung geschieht mittels einem 7" Touch-Display, welches sich über dem Flüssigkeitsauslass befindet. Das selbe Display wird auch in der Schifffahrt gebraucht und ist deswegen besonders robust und Flüssigkeitsresistent.[2]

Technische Daten

Zeit für einen Cocktail:	: Für die Zubereitung eines Cocktails benötigt die Maschine lediglich 3-5 Sekunden. Somit sind bis zu 300 Cocktails in der Stunde möglich. [2]
Anzahl verschiedene Cocktails:	: Abhängig von der Anzahl Getränkeanschlüssen können mehr oder weniger Getränke zubereitet werden. Der Speicherplatz bietet jedoch Platz für 500 gespeicherte Cocktails. Die Cocktails können frei programmiert und gespeichert werden, genauso wie die Dosierungseinstellungen. Dies kann direkt an der Maschine vorgenommen werden und benötigt keiner Software. [2]
Stromversorgung:	: Der Cocktailmixer benötigt in jedem Fall einen Festanschluss an das Stromnetz. [2]

Reinigung

Über die Reinigung wurden keine Angaben gefunden.

Sonstiges

Die Erweiterung der Flüssigkeiten und Pumpen kann jeweils in Zweierschritten geschehen. In den für die Flüssigkeiten vorgesehenen Behältern ist eine Füllstanderkennung eingebaut. Der Support ist sehr ausführlich, es gibt Schulungen in der Software der Cocktailmaschine und die Maschine wird inklusive An- und Abfahrt aufgebaut. Das verwendete Material, worin die Getränke gelagert werden, sind lebensmittelzertifiziert.[2]



Abbildung 2.2: Der Cocktailmixer [2]

2.1.3 CocktailAvenue

Aufbau

Die Maschine CocktailAvenue besteht aus einer Rückwand, an der die Getränkeflaschen mit dem Flaschenhals nach unten montiert werden können. In der Rückwand ist ein Touchpad verbaut, über welches die Cocktails ausgewählt werden können. Das Glas wird gefüllt, indem es auf einem Schlitten unter die Flaschen fährt und dann die Getränke eingefüllt werden. Der Schlitten wird mittels einem Zahniemen hin und her gefahren. Das Auffüllen geschieht nicht mit Pumpen, sondern mit reiner Schwerkraft. Dosiert werden die Getränke mit Dosiergeräten, wie man sie aus den Bars kennt. Dabei wird eine Fixe Dosis (meist 2cl) hinaus gelassen. Die Dosiereinheit wird geöffnet, indem ein Hubzapfen aus der Halterung des Schlittens nach oben gefahren wird. Es gibt jedoch auch zwei Hahnen, aus denen Getränke hinausgelassen werden können. Vermutlich werden dort Pumpen eingesetzt.[3]

Bedienung

Die Bedienung geschieht über das Touchpad, welches in der Rückwand verbaut ist.[3]

Technische Daten

Zeit für einen Cocktail:	: Für die Zubereitung eines Cocktails braucht die Maschine gemäss einem Beispiel eines Youtube-Videos 25 Sekunden. [4]
Anzahl verschiedene Cocktails:	: Es wurden keine Angaben über die Anzahl verschiedene Cocktails gefunden.
Stromversorgung:	: Der Cocktailmixer benötigt in jedem Fall einen Festanschluss an das Stromnetz.[3]

Reinigung

Die Reinigung gestaltet sich bei der CocktailAvenue ziemlich einfach. Da es keine Pumpen und Schläuche gibt, entfällt ein sehr grosser Teil. Einzig die Dosiergeräte müssen zwischendurch demontiert und sauber gemacht werden. Auch der mechanische Aufbau der Schiene unterhalb der Getränke wurde so konzipiert, dass eine Reinigung sehr einfach ist und eventuelle Verunreinigungen den Betrieb nicht beeinflussen.[3]

Sonstiges

Die Antriebstechnik geschieht mit einer Zahnriemenachse. Sie besteht aus einem Aluprofil. Die Achse muss nicht geschmiert werden, was demnach bei Anwendungen mit hohen Hygieneanforderungen sehr vorteilhaft ist. Ebenso wird so ein wartungsfreier Betrieb gewährleistet. Der Antrieb wird mit einem Schrittmotor realisiert, was eine vibrationslose Bewegung garantiert. Damit die Kabel bei Bewegungen sauber geführt und nicht geknickt werden, wird eine Kabelkette verwendet. Zukünftig plant der Hersteller eine Erweiterung der modular aufgebauten Baureihe. Mittlerweile gibt es auch eine Version, bei der die Antriebsachse im Gehäuse verbaut ist. Die exakte Positionierung des Schlittens unter den Flaschen muss nicht sehr genau sein, da die Achse nach jeder Zubereitung wieder den Ausgangs- und Referenzpunkt anfährt. [3]



Abbildung 2.3: CocktailAvenue [3]

2.1.4 myRocktail

Aufbau

Bei my Rocktail handelt es sich um eine mietbare Cocktailmaschine, welche die Cocktails mittels eines Roboterarms zusammenmixt. Dabei kann er aus bis zu zwölf Zutaten den gewünschten Cocktail zubereiten. Die Maschine ist mobil aufgebaut und kann mittels Räder verschoben werden. Die Front der Maschine ist halbrund aufgebaut, wobei sich die Getränkeflaschen in dieser Rundung auf der Oberseite befinden. Diese sind mit dem Flaschenhals nach unten montiert, so dass das Getränk mittels automatischer Dosierkappe in das Glas abgefüllt werden kann. In der Mitte des Gerätes befindet sich zudem ein Getränkeauslass, welcher das gewünschte Süßgetränk oder Wasser hinzu mischt. [5]

Bedienung

Der Kunde kann mittels eines Touch-Displays das gewünschte Getränk auswählen und stellt dann sein Glas auf den dafür vorgesehenen Platz des Roboterarms. Danach fährt dieser die gewünschten Glaspositionen mit dem Glas an und befüllt dieses mittels den automatischen Dosierkappen mit der gewünschten Menge des Getränks. [5] [6]

Technische Daten

Zeit für einen Cocktail:	: Eine Zubereitungszeit wird nicht explizit angegeben. Allerdings kann gemäss eines Videos entnommen werden, dass die Zubereitungszeit ungefähr 80 Sekunden dauert. [6]
Anzahl verschiedene Cocktails:	: Es kann zwischen vier alkoholischen und vier nicht alkoholischen Getränken ausgewählt werden. Dabei besteht jedoch auf Kundenwunsch die Möglichkeit eigene Kreationen zu erschaffen.
Stromversorgung:	: Es konnten keine Informationen zur Stromversorgung ausfindig gemacht werden.

Reinigung

Die Endreinigung wird manuell durch das Servicepersonal durchgeführt. Ob dabei ein automatischer Reinigungsmodus existiert konnte nicht ausfindig gemacht werden. Allerdings wird auf dem Prospekt mit einer einfachen Reinigung geworben. [5]



Abbildung 2.4: myRocktail [5]

2.2 Entscheidungfindung des Aufbaus

Für die Entscheidungsfindung des Aufbaus wird eine Tabelle 2.1 erstellt, welche die Vor- und Nachteile der Maschinen aufzeigt.

Cocktailmaschine:	Vorteile:	Nachteile:
Spirits	<ul style="list-style-type: none"> – Schnelle Zubereitung (ca. 20s) – Mobiler Aufbau – Modularer Aufbau – Einfache Bedienung via App – Abspielen von Musik – Einfache Reinigung 	<ul style="list-style-type: none"> – Kein Showeffekt für Befüllung
Der Cocktailmixer	<ul style="list-style-type: none"> – Sehr schnelle Zubereitung (ca. 3-5s) – Mobiler Aufbau auf Rollen – Einfache Bedienung via Touchpad – Riesige Getränkeauswahl – Grosses Fassungsvolumen 	<ul style="list-style-type: none"> – Kein Showeffekt für Befüllung – Grosse, sperrige Maschine – Keine Appsteuerung – Grosser Reinigungsaufwand – Kein Showeffekt
Cocktail Avenue	<ul style="list-style-type: none"> – Schnelle Zubereitung (ca. 25s) – Einfache Bedienung via Touchpad – Einfacher Flaschenwechsel – Präzise Dosierung – Grosser Showeffekt durch Schlitten – Einfache Reinigung 	<ul style="list-style-type: none"> – Längere Zubereitungszeit durch Bewegung – Benötigt viel Platz – Stark begrenzte Getränkeauswahl
myRocktail	<ul style="list-style-type: none"> – Grosser Showeffekt durch Roboterarm – Einfache Bedienung via Touchpad – Einfacher Flaschenwechsel – Präzise Dosierung – Mobiler Aufbau auf Rollen – Einfache Reinigung 	<ul style="list-style-type: none"> – Längere Zubereitungszeit durch Bewegung (ca. 80s) – Benötigt viel Platz – Stark begrenzte Getränkeauswahl

Tabelle 2.1: Cocktailmaschinen Vergleich

Die Cocktailmachine soll einige Anforderungen erfüllen. Aus diesen resultiert dann die komplette Entscheidungsfindung bezüglich des Aufbaus. Dies ist zum einen der Showeffekt. Die Cocktailmachine soll ein Hingucker sein und dem Benutzer etwas für das Auge bieten. Daher soll das Glas zu den verschiedenen Befüllgetränken befördert werden, um diese zu befüllen. Außerdem soll es weniger als eine Minute dauern, bis ein Cocktail erstellt ist. Da der Roboterarm weit mehr als eine Minute benötigt um einen Cocktail zu erstellen, führt zur Entscheidung ein Förderband einzusetzen, wie es bei der Cocktail Avenue der Fall ist. Weiter soll dem Benutzer eine möglichst Benutzerfreundliche Bedienoberfläche geboten werden. Diese soll so intuitiv wie möglich gestaltet werden. Daher wird ein Touchscreen verbaut, wie es beim Cocktailmixer und der myRocktail der Fall ist. Von der Spirits soll die Flüssigkeitsbeförderung übernommen werden. Daher werden Pumpen, sowie Durchflussmessgeräte verbaut.

3 Grobkonzept

Das Grobkonzept soll eine Übersicht über den Aufbau der Maschine bieten. Dabei wird erläutert, aus welchen Hauptkomponenten die Cocktailmashine besteht und weshalb diese ausgewählt wurden. Ausserdem sollen deren technischen Daten aufgezeigt werden.

3.1 Blockschaltbild

Das Blockschaltbild soll die Funktionsweise der Cocktailmashine veranschaulichen. Dabei wird unterschieden zwischen der Steuerelektronik (MCS) und den externen Komponenten wie Netzteil, Touchscreen (HMS), Beleuchtung (ML), Pumpen, Durchflussmessgeräte, Endschalter und Motor. Aufgrund dieses Blockschaltbildes werden in den folgenden Kapitel die Hauptkomponenten ausgewählt.

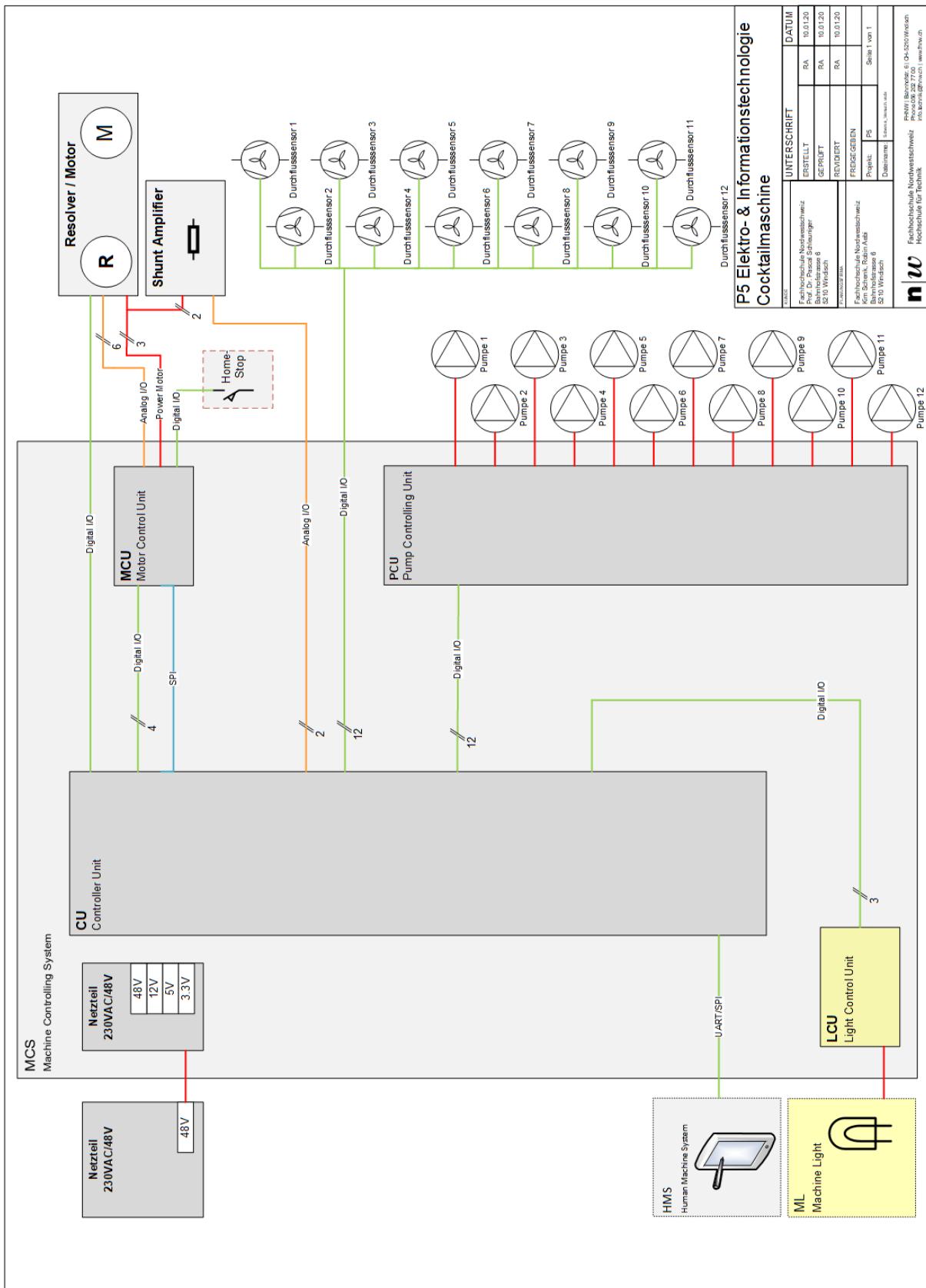


Abbildung 3.1: Blockschaltbild

3.2 Motor

Als Motor wird ein bürstenloser Gleichstrommotor (BLDC-Motor) verwendet. Diese Bauform wurde vom betreuenden Dozent vorgegeben. Ein BLDC-Motor zeichnet sich dadurch aus, dass er zu den Antrieben mit dem besten Verhältnis zwischen Leistung und Gewicht zählt. Außerdem verfügt er über ein relativ hohes Drehmoment. Die Anwendung bringt bis auf die Drehlager keinen Verschleiss mit sich. Er ist zuverlässiger, effizienter und zudem noch leiser als herkömmliche Motoren [7]. Im folgenden Kapitel soll gezeigt werden, wie solch ein Motor aufgebaut ist und nach welchem Funktionsprinzip er funktioniert.

3.2.1 Aufbau

Ein bürstenloser Gleichstrommotor funktioniert im Grunde nicht nach dem Prinzip eines Gleichstrommotors. Vom Aufbau her ist er gebaut wie ein Wechselstrom-Synchronmotor, wobei die Erregung mit Permanentmagneten gewährleistet wird. Beim Synchronmotor am 400V-Netz geschieht die Magnetisierung der Spulen mit der sinusförmigen Netzfrequenz. Beim BLDC-Motor hingegen über die wechselnde Magnetisierung mit einem Gleichstrom. Es wird folglich eine Steuerlogik benötigt, welche diesen Gleichstrom auf die Spulen schaltet. In Abbildung 3.2 wird der Aufbau beider Motoren veranschaulicht. [8]

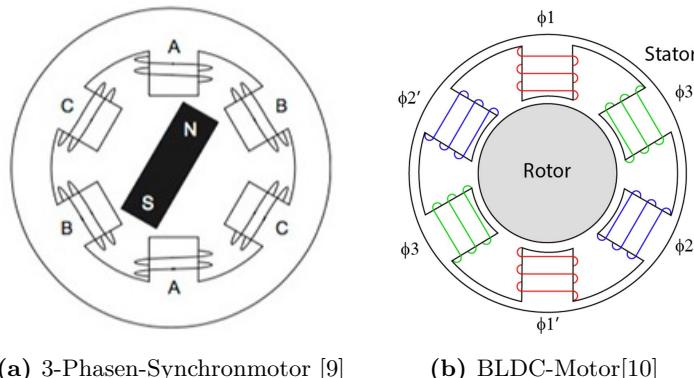


Abbildung 3.2: Vergleich des Aufbaus zwischen Synchron- und BLDC-Motor.

Hauptsächlich unterscheiden sich die BLDC-Motoren in Außen- und Innenläufer.

Aussenläufer

Aussenläufer haben den Rotor außerhalb des Stators (siehe Abb. 3.3). Der Rotor bei Außenläufern weist deshalb eine grösere Massenträgheit auf. Aus diesem Grund sind sie bevorzugt, wenn es darum geht zu erwartende Schwankungen in der Drehzahl oder im Drehmoment auszugleichen. [11] [12]

Innenläufer

Innenläufer haben den Rotor auf der Innenseite des Stators (siehe Abb. 3.2). Sie werden aufgrund der niedrigeren Trägheit für schnelle Richtungsänderungen bevorzugt.

3.2.2 Funktionsprinzip

In Abbildung 3.3 sind die Spulen in der Mitte statisch und bilden den Stator. Die Permanentmagnete sind am schwarzen Teil befestigt, welcher den Rotor darstellt. Durch die von einem Strom verursachte Magnetisierung der B-Spule ganz links auf der Abbildung, wird der Nord- und Südpol des Rotors im Gegenuhrzeigersinn angezogen. Die Kräftewirkung ist in der Abbildung mit einem grünen Pfeil dargestellt. Der Rotor beginnt sich zu drehen. Sobald der Rotor die Position erreicht, wie sie in der Mitte der Abbildung dargestellt wird, ändert sich die Magnetisierung von Spule B auf Spule C. Der Prozess der Anziehung und Drehung findet erneut statt und endet mit Erreichen der nächsten Position. Darauf folgt eine weitere Änderung der Magnetisierung von Spule C auf A. Wird dieser Prozess kontinuierlich wiederholt, dreht sich der Motor im Kreis.

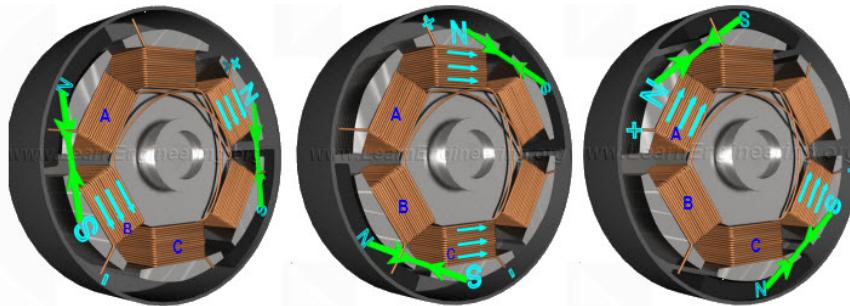


Abbildung 3.3: Kraftwirkung nur mit Anziehung (grün) [7]

Würde man den Motor blockartig betreiben, hätte man nur wenige Positionen, welche angesteuert werden können. Zusätzlich können Vibrationen auftreten. Um dies zu verhindern werden die Spulen mit einem PWM-Signal gespiesen. Dies hat zur Folge, dass der Strom durch die Spulen und somit auch die Magnetisierung regulierbar ist. So kann der Motor mit feinen Positionsänderungen angesteuert werden und er verhält sich ruhiger im Betrieb. Das PWM-Signal kann beispielsweise wie in Abbildung 3.4 dargestellt generiert werden.

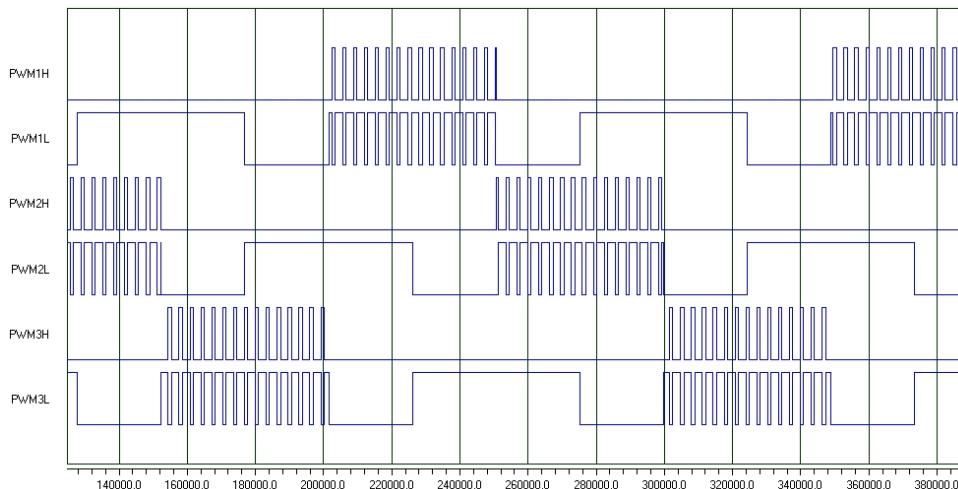


Abbildung 3.4: PWM-Signal der Kommuntierung. Das invers-PWM auf der Low-Side wird geschaltet, um die Body-Diode zu entlasten. [13]

Die PWM-förmige Kommutierungsspannung beeinflusst den Strom und somit die Magnetisierung der Spulen. Wird ein zunehmend breiteres PWM-Signal angelegt (größerer Duty-Cycle), steigt der Strom. Wird das Signal wieder schmäler (kleinerer Duty-Cycle), sinkt der Strom. Dies ist in Abbildung 3.5 ersichtlich.

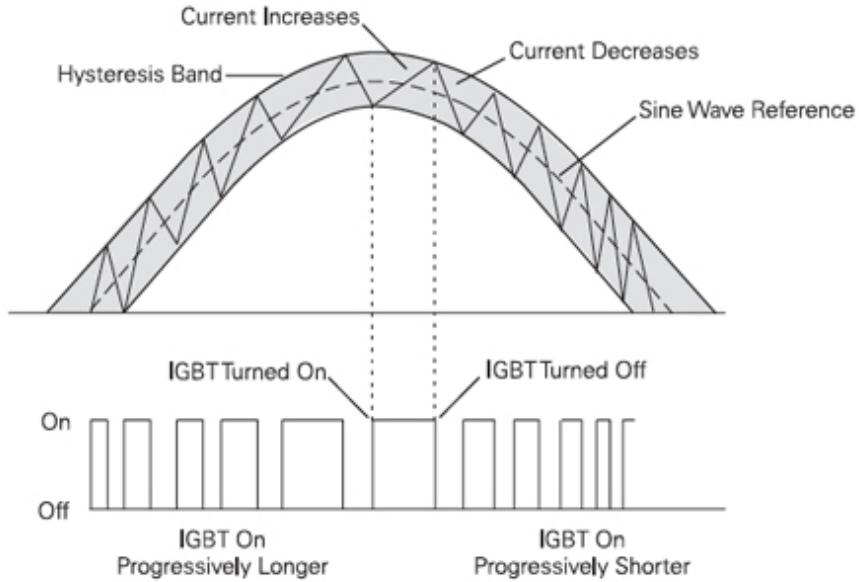


Abbildung 3.5: Kraftwirkung nur mit Anziehung (grün) [14]

Dieser Prozess kann verstärkt werden, indem das Prinzip der Abstossung genutzt wird. Dazu werden zusätzlich die Spulen gemäß Abbildung 3.6 magnetisiert. Je nach Anwendung kann es aber auch möglich sein, dass man einen Rotor still halten möchte. Dazu lässt man die Spulen permanent magnetisiert, ohne das Feld zu ändern. Die Aufgabe, dies zu steuern, übernimmt der von Trinamic entwickelte TMC4672. Dessen Funktion wird in Kapitel 3.4 erläutert.

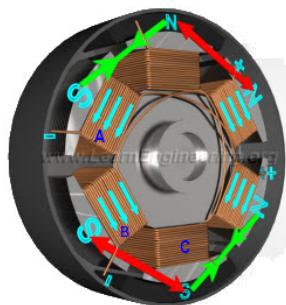


Abbildung 3.6: Kraftwirkung mit Anziehung (grün) und Abstossung (rot) [7]

3.3 Resolver

Damit die genaue Lage des Rotors festgestellt werden kann, benötigt es eine Drehpositionserfassung. Im Falle des verwendeten Motors wird ein Koordinatenwandler, auf Englisch auch Resolver genannt, verwendet. Dieser hat die Eigenschaft, eine kontinuierliche Winkellage des Rotors zu liefern und ist fest im AKM22h verbaut. Aus diesem Grund wird in diesem Kapitel die Funktion des Resolvers vertieft betrachtet.

3.3.1 Aufbau

Damit ein Resolver funktionieren kann, sind eine Erregerspule und zwei Sensorspulen nötig. Weiter muss mittels Induktion die Erregungsfrequenz auf die Erregerspule übertragen werden. Der schematische Aufbau ist in Abbildung 3.7a ersichtlich. Abbildung 3.7b zeigt, wie die Komponenten zu einander stehen und wie sie (bis auf die blaue Leiterschleife) im Resolver angeordnet sind.

Das Prinzip funktioniert folgendermassen. Es wird ein periodisches Signal, vorzugsweise Sinus, auf die statische gelbe Spule gegeben. Dieses wird mittels Induktion des drehbaren Transformators an die drehbare blaue Spule übertragen. Die blaue Spule erzeugt unabhängig ihrer Lage ein magnetisches Feld anhand des Erzeugersignals und bildet so die Erregerspule. Die grüne und pinke Sensorspulen werden je nach Lage der blauen Spule unterschiedlich stark erregt und erzeugen das benötigte Signal, welches dann von der Auswertelektronik verarbeitet wird.

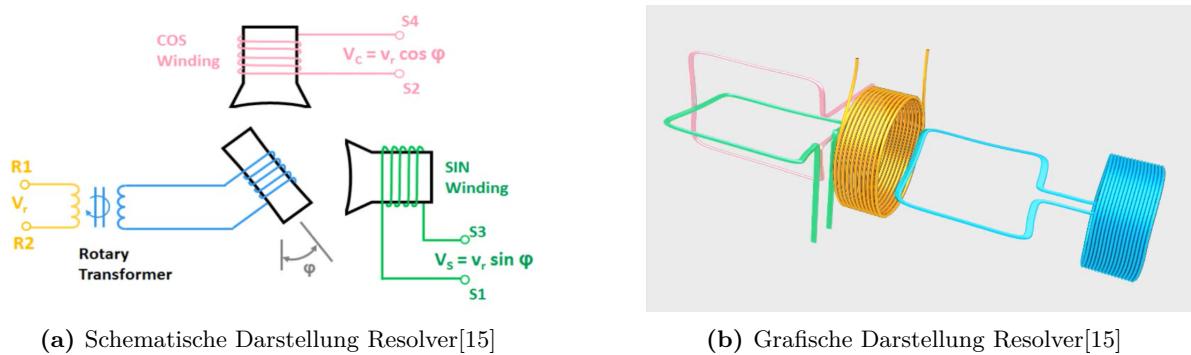


Abbildung 3.7: Schematische und grafische Darstellung eines Resolvers.

3.3.2 Funktionsweise

Steht der Rotor wie in Abbildung 3.8c bei 0° , so liegt die drehbare blaue Leiterschleife so, dass der von ihr ausgehende magnetische Fluss nur eine Wirkung auf die pinke Leiterschleife hat (siehe Abbildung 3.8a). Das Erregersignal wird folglich praktisch nur von der cos-Windung wahrgenommen, wie in Abbildung 3.8b ersichtlich ist.



Abbildung 3.8: Gegebenheiten bei Achsenstellung 0° .

Steht der Rotor wie in Abbildung 3.9c bei 90° , so liegt die drehbare blaue Leiterschleife so, dass der von ihr ausgehende magnetische Fluss nur eine Wirkung auf die grüne Leiterschleife hat (siehe Abbildung 3.9a). Das Erregersignal wird folglich praktisch nur von der sin-Windung wahrgenommen, wie in Abbildung 3.9b ersichtlich.

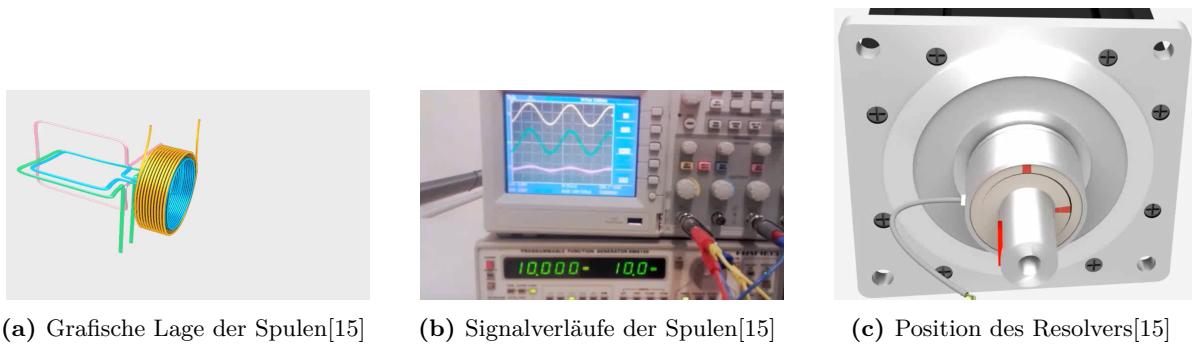


Abbildung 3.9: Gegebenheiten bei Achsenstellung 90°.

3.4 TMC4671-EVAL Board

Wie bereits angesprochen steuert der TMC4671 die Kommutierungsvorgänge für den BLDC. Doch er ist noch zu viel mehr in der Lage. Der TMC4671 ist ein voll integrierter Servo-Controller, der eine feldorientierte Steuerung für BLDC/PMSM, 2-Phasen-Schrittmotoren und DC-Motoren unterstützt. Ob integrierte ADCs, Lagesensor-Schnittstellen oder Positionsinterpolatoren, alle diese in der Hardware implementierten Steuerungsfunktionen bieten einen voll funktionsfähigen Servoregler für ein breites Spektrum an Servoanwendungen. [16]

3.4.1 Aufbau

Das komplette EVAL-Board, wie es in Abbildung 3.10 gezeigt wird, besteht aus einer Landungsbrücke (links), einem TMC4671 (mitte) und einer H-Brücke (rechts). Verbunden werden die drei Teile mittels zwei Eselsbrücken.

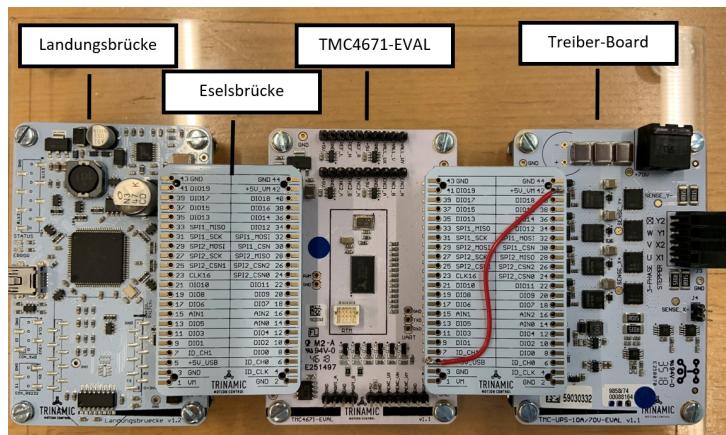


Abbildung 3.10: TMC4671-EVAL Board. unbekannt_foc-motorenansteuerung_2019

Die Landungsbrücke verfügt über einen USB-Anschluss, um vom Computer aus auf die Landungsbrücke zugreifen zu können. Sie bildet somit die Schnittstelle zwischen der TMCL-IDE und dem Motorentreiber, worüber die Konfiguration des Treibers stattfindet. Der Motorentreiber stellt anhand der Konfigurationen und des Feedbacks (Spulenströme und Encodersignale) die Steuersignale für den Gate-Treiber bereit. Der Gate-Treiber steuert die Spulenströme und magnetisiert so die Spulen des BLDC-Motors anhand der Steuersignale des Treibers. Dies wiederum verursacht eine Bewegung des Rotors.

3.4.2 FOC

FOC³ ist ein Stromregelverfahren für Elektromotoren. Es reguliert die Kraft und Lage des magnetischen Feldes unter Berücksichtigung der Rotorposition, sodass der Motor das geforderte Drehmoment als Soll-Drehmoment abgibt. FOC maximiert die Wirkleistung und minimiert die Leerlaufleistung. Dies wiederum ergibt eine Verminderung der Verlustleistung durch intelligente Regelung. [16]

Auf den Rotor eines BLDC-Motors wirken zwei Kraftkomponenten. Gemäss Abbildung 3.11 zieht eine Komponente radial in eine Richtung I_D und eine andere Komponente tangential in eine Richtung I_Q . Die zweite Komponente I_Q ist diejenige, welche ein Drehmoment auf den Rotor bringt. Der ideale Controller führt eine Regelung durch, welche einen rein drehmomenterzeugenden Strom I_Q erzeugt. [16]

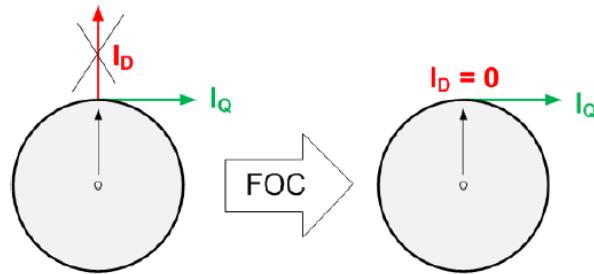


Abbildung 3.11: TMC4671-EVAL Board. [16]

Der Controller verwendet für das Stromregelverfahren die drei Phasenströme des Stators, welche zusammen als Vektor betrachtet werden (I_U, I_V, I_W). Unter Berücksichtigung der momentanen Ausrichtung des Rotors wird ein Spannungsvektor (U_U, U_V, U_W) berechnet, sodass nur ein drehmomentbildender Strom I_Q erzeugt wird (Transformation). [16]

Dazu sind einige statische Parameter nötig, wie z.B. die Polpaarzahl des Motors, Anzahl der Impulse pro Umdrehung des verwendeten Drehgebers. Aber auch einige dynamische Parameter wie z.B. die Phasenströme oder die Orientierung des Rotors. [16]

Von Bedeutung sind die Einstellungen der P und I Parameter zur Regelung der Phasenströme. Diese sind abhängig von den elektrischen Parametern des Motors wie z.B. der Widerstand, die Induktivität, die Gegen-EMF-Konstante oder die Versorgungsspannung. [16]

Abbildung 3.12 zeigt, an welcher Stelle im TMC4671 FOC (FOC23) aktiv ist. Wie zu erkennen ist, steht es mit fast allen Teilsystemen in Kontakt.

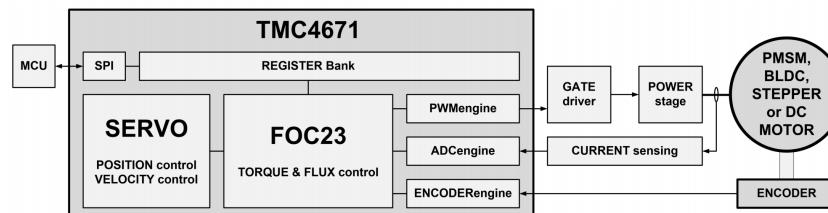


Abbildung 3.12: Standard-Anwendungs-Schaltung. [16]

³FOC = Field Oriented Control

3.4.3 Park & Clarke

Für das Stromregelverfahren werden die Funktionen Clarke, Park, iClarke und iPark verwendet. Im Folgenden wird gezeigt, wie sie im Zusammenhang mit dem drehmomentgebenden Strom, den Phasenströmen, den Phasenspannungen und den PI-Parameter stehen. In Abbildung 3.13 sind die Zusammenhänge grafisch dargestellt.

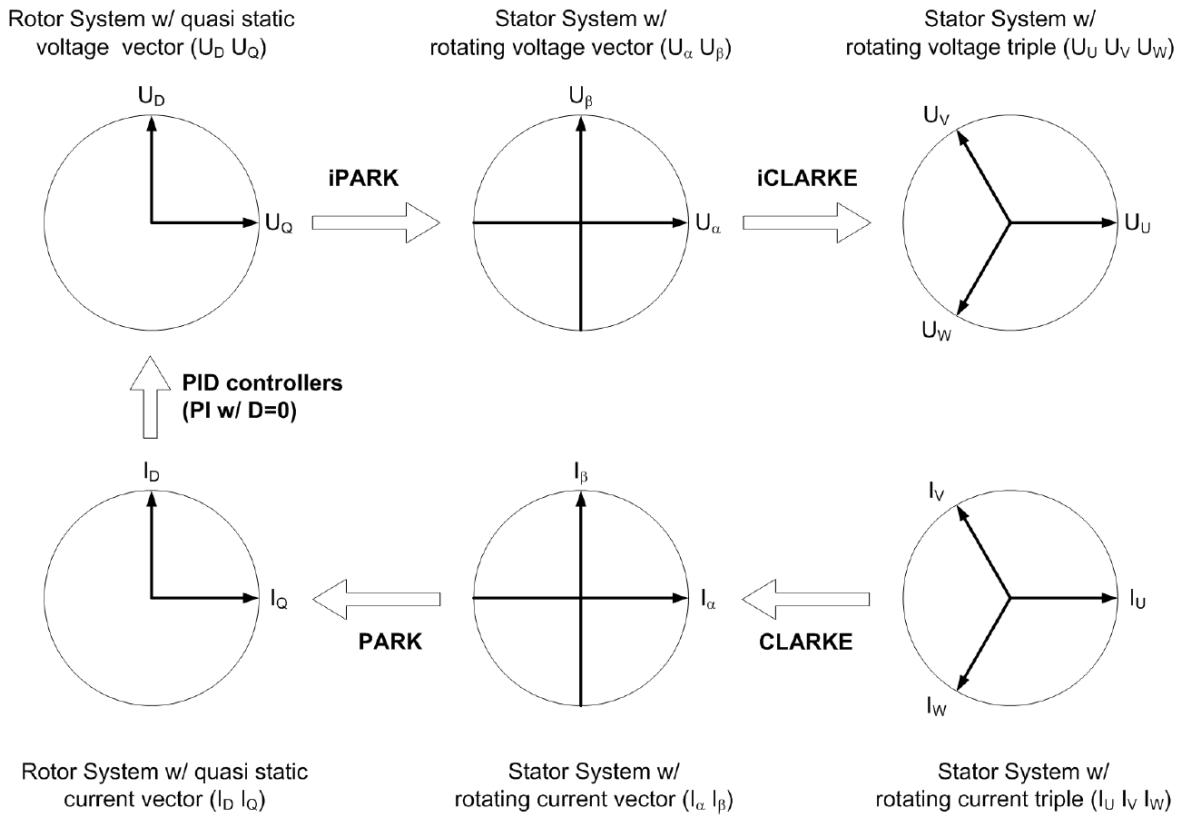


Abbildung 3.13: TMC4671-EVAL Board. [16]

Die Clarke-Transformation (CLARKE) bildet drei Motorphasenströme (I_U, I_V, I_W) auf ein zweidimensionales Koordinatensystem mit zwei Strömen (I_α, I_β) ab. Basierend auf dem tatsächlichen Rotorwinkel, der durch einen Encoder oder über eine sensorlose Elektronik ermittelt wird, bildet die Park-Transformation (PARK) diese beiden Ströme auf ein näherungsweise statisches Koordinatensystem mit zwei Strömen (I_Q, I_D) ab. Der aktuelle Strom I_D steht für den Flux und der aktuelle Strom I_Q für das Drehmoment. Der Flux zieht nur am Rotor, beeinflusst aber das Drehmoment nicht. Das Drehmoment wird durch den Strom I_Q beeinflusst. Zwei PI Regler (PID_Q, PID_D)⁴ ermitteln zwei Spannungen (U_Q, U_D), um die gewünschten Ströme für ein Soll-Drehmoment und -Flux zu errechnen. Die ermittelten Spannungen (U_Q, U_D) werden durch die inverse Park-Transformation (iPARK) gebildet. Die inverse Clarke-Transformation (iCLARKE) transformiert diese beiden Spannungen in drei Spannungen (U_U, U_V, U_W). [16]

Diese drei Spannungen sind der Eingang der PWM-Engine. Dies ist in Abbildung 3.12 und Abbildung 3.14 zu erkennen.

Für die Cocktailschnecke ist der Teil mit den PID-Reglern interessant, da wir an dieser Stelle das Verhalten des Motors regeln können. Insbesondere die Geschwindigkeit und Beschleunigung muss unter Kontrolle sein. Die Funktion der Regler wird deshalb genauer betrachtet.

⁴siehe Paragraph "PID-Regler"

3.4.4 PID-Regler

Im Folgenden wird erklärt, wie die PID-Regler im Treiber funktionieren. In Abbildung 3.14 ist zu erkennen, dass an jeder Stelle, an der Regler verfügbar sind, PID_n steht.

- PID_x = Regler für die Position x
- PID_v = Regler für die Geschwindigkeit v
- PID_Q = Regler für den tangentialen Strom I_Q
- PID_D = Regler für den radialen Strom I_D

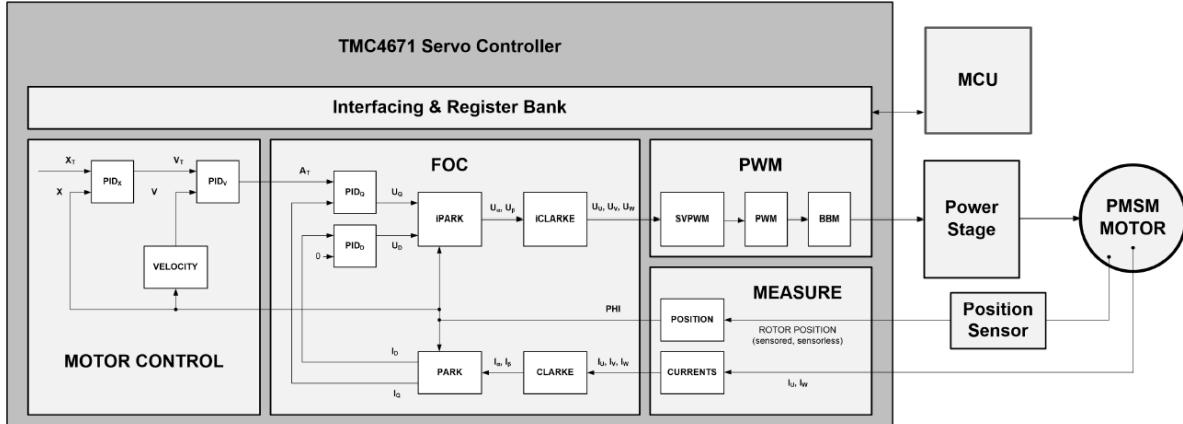


Abbildung 3.14: Hardware FOC Blockdiagramm.[16]

Ein DC-Motor wird vom Prinzip her wie in Abbildung 3.15 geregelt. Diese Abbildung zeigt einen vereinfachten Aufbau einer Antriebsregelung. Mittels den eingestellten PI-Parametern und dem eingestellten Betriebsmodus (Torque/Flux, Velocity, Position) werden beim TMC4671 die Abhängigkeiten der Regler geändert. Dies ist in Abbildung 3.16 ersichtlich. Im Drehmoment-Modus wird „TARGET_TORQUE“ als Ziel-Drehmoment verwendet. Im Geschwindigkeitsmodus wird das Zieldrehmoment vom Geschwindigkeits-PID-Regler vorgegeben. Im Drehmomentregler kann das maximale Drehmoment vorgegeben werden, welches auf den Motor gegeben wird. Eine detaillierte Übersicht der einzelnen Regler und deren Abhängigkeiten ist in Abbildung 3.17 zu finden.

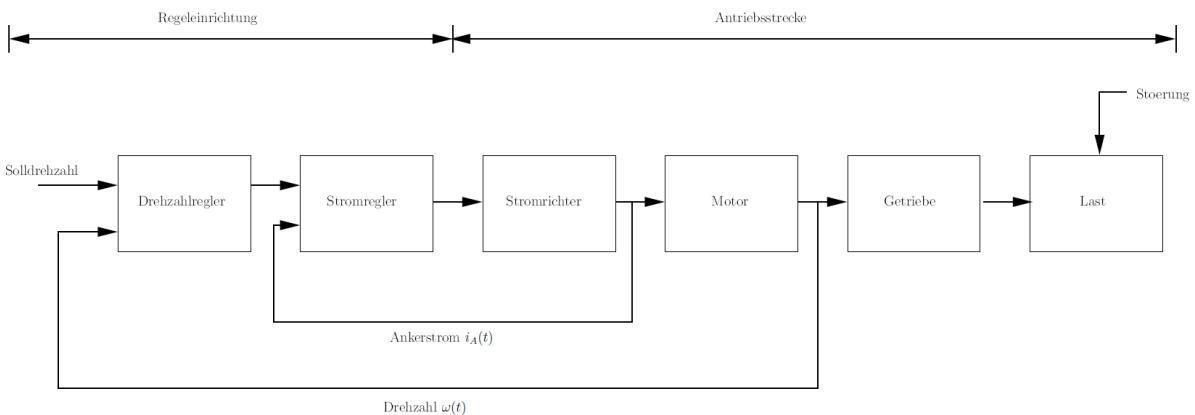


Abbildung 3.15: Der Prinzipielle Aufbau einer Antriebsregelung für einen DC-Motor. [17]

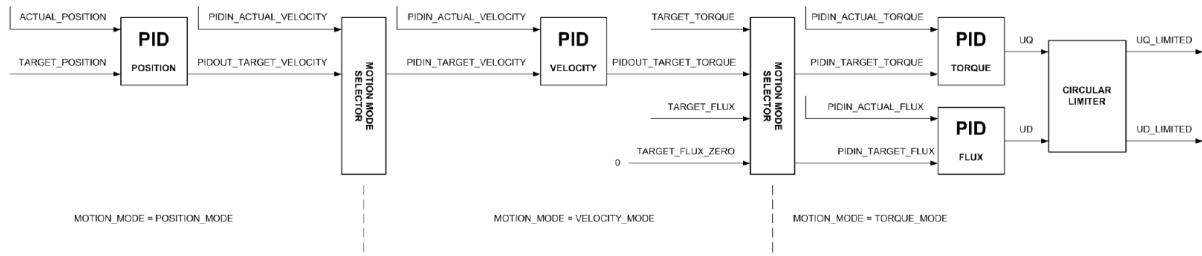


Abbildung 3.16: Abhängigkeiten der Regler.[16]

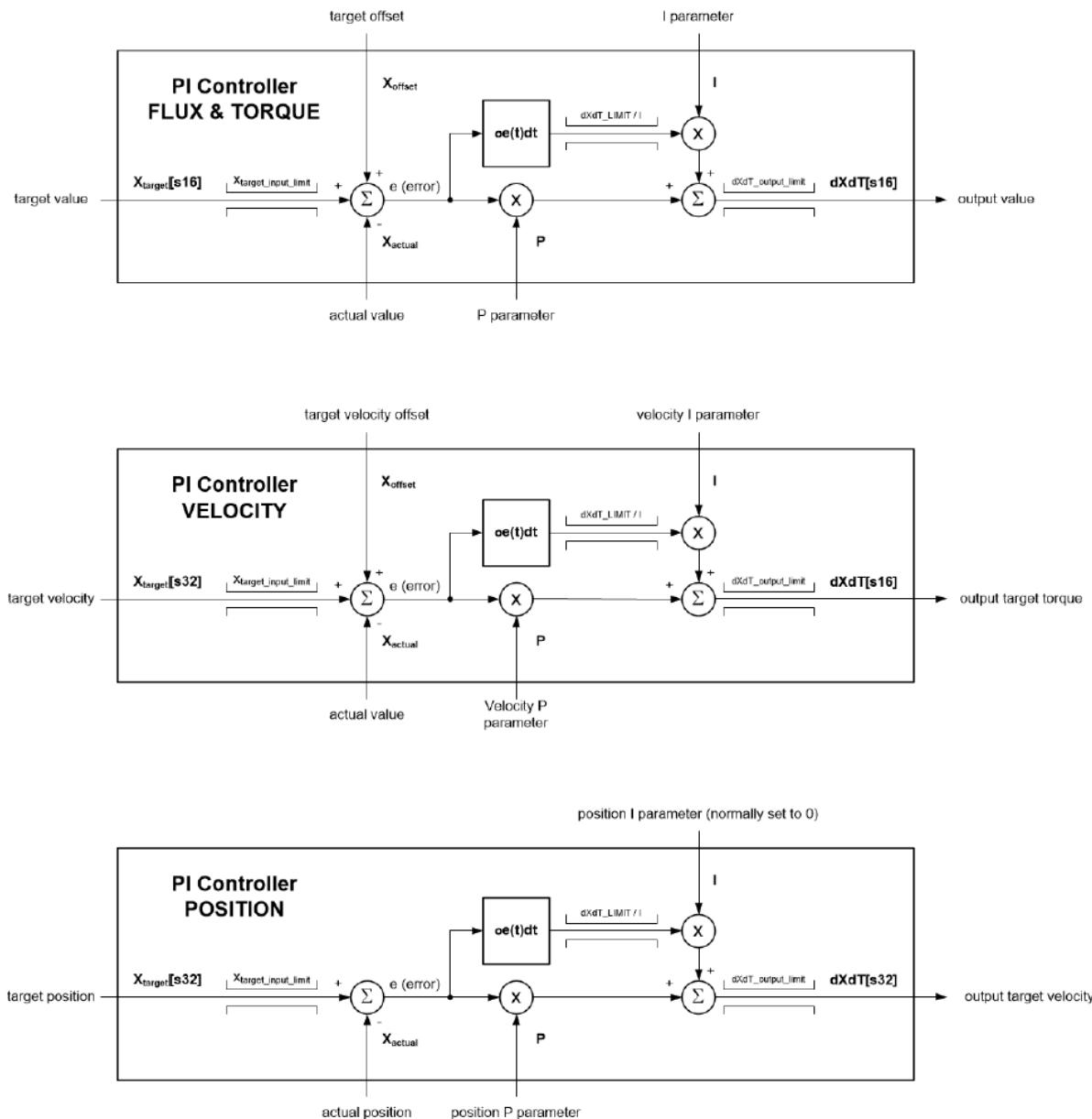


Abbildung 3.17: Die Reglerstufen im Detail. [16]

Abbildung 3.18 zeigt, wie die Regelung im Positionsmodus eine Ziel-Position vorgibt (rot) und wie die aktuelle, gemessene Position tatsächlich ist (blau). Zu erkennen ist, dass die blaue Linie die Zielposition ziemlich steil anfährt und nicht überschiesst.

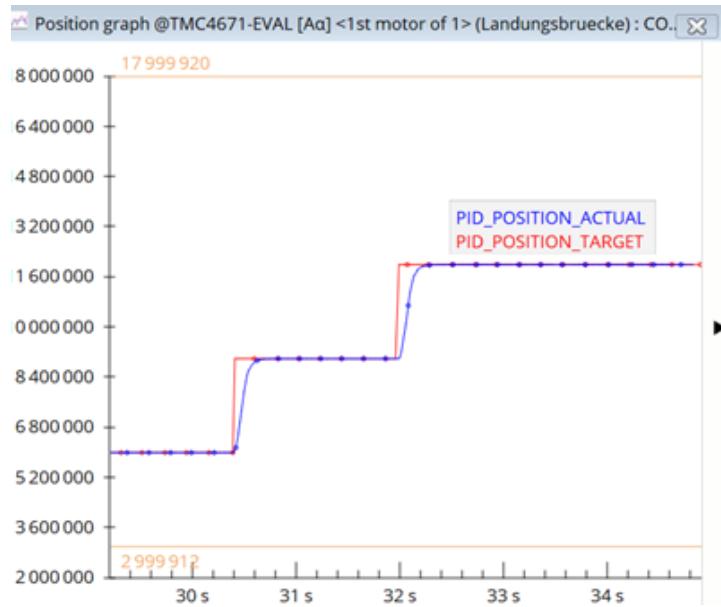


Abbildung 3.18: Geschwindigkeitsregler mit Zielgraph und aktueller Graph. [18]

3.5 Förderband

Da gemäss Kapitel 2.2 ein Förderband eingesetzt wird um das Glas zu bewegen, muss ein Motor eingesetzt werden, welcher das Glas mitsamt Inhalt transportieren kann. Ausserdem sollen die verschiedenen Anfahrtspositionen exakt angefahren werden können.

3.5.1 Anforderungen

Mechanischer Aufbau :	Länge	$80 \pm 10\text{cm}$ (Unter Annahme 10cm pro Flasche)
	Geschwindigkeit	180cm pro 30s = 6cm/sec
	Belastbarkeit	9.81N (1kg)
	Oberfläche	Rutschfest
	Führung	zwei Führungsstangen um den Schlitten zu bewegen.
	Schlitten	evt. 3D Druck

3.5.2 Verwendeter Aufbau

Der Aufbau der Cocktailmachine soll gemäss der Entscheidungsfindung des Aufbaus der Cocktailmachine 2.2 ähnlich aussehen, wie bei der CocktailAvenue 2.1.3. Allerdings sollen die Getränkebehälter nicht sichtbar hinter der Maschine platziert werden. Somit könnte man sagen, dass es eine Mischung aus der CocktailAvenue 2.1.3 und dem Cocktailmixer 2.1.2 sein wird. Ein erster Prototyp soll zuerst aus Holz angefertigt werden, bevor die Maschine dann aus Metall (vorzugsweise Chromstahl) aufgebaut wird.

3.6 Pumpen

Um die Flüssigkeit aus den Flaschen in das Glas abzufüllen, werden gemäss Kapitel 2.2 Pumpen eingesetzt. Es werden daher verschiedene Typen miteinander verglichen.

3.6.1 Anforderungen

Durchflussmenge	:	Aus Pflichtziel	min. 0.6l pro Minute
		Aus Wunschziel	min 1.2l pro Minute
Spannung	:	Spannungsversorgung	12-24V
Schlauchanschluss	:	Innendurchmesser	6-8mm
Sonstiges	:	Hygieneanforderungen	Lebensmittelpumpe

3.6.2 Mögliche Pumpen

Da es sich um Lebensmittel handelt welche befördert werden sollen, werden zwei verschiedene Pumpentypen verglichen. Einerseits ist dies die Schlauchpumpe, welche über ihren mechanischen Aufbau ein Vakuum erzeugt, und somit Flüssigkeit ansaugt. Der grosse Vorteil dieses Pumpentyps ist, dass diese durch den Aufbau in der Lage ist, die zu befördernde Menge zu dosieren. In Abbildung 3.19 kann der Aufbau dieses Pumpentyps betrachtet werden. [19]

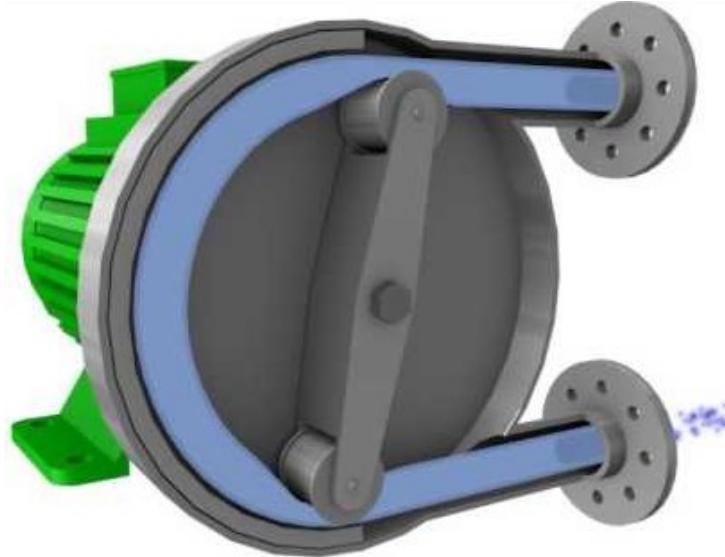


Abbildung 3.19: Aufbau einer Schlauchpumpe [20]

Bei jeder spezifizierten Umdrehung des Motors wird die im Schlauch befindliche Menge an Flüssigkeit zwischen zwei oder mehreren Rollen befördert. In Abbildung 3.19 ist dies nach jeder halben Umdrehung der Fall mit zwei Rollen. Werden mehrere Rollen verbaut, so kann diese Menge variiert werden. Mit der Regelung der Drehzahl des Motors kann somit die zu befördernde Flüssigkeitsmenge über die Zeit bestimmt werden. Eingesetzt werden diese Pumpen oft in der Medizinaltechnik, wo meist kleinere Mengen an Flüssigkeit befördert werden. Dabei kann mit einer Genauigkeit von weniger als 1% gerechnet werden. Werden grössere Flussraten benötigt, so steigt die Ungenauigkeit dieses Pumpentyps. So muss bei einer Flussrate von 800 ml/min mit einer Abweichung von $\pm 8\%$ gerechnet werden. Um diese Ungenauigkeit beheben zu können, müsste ein externes Durchflussmessgerät installiert werden. [19]

Beim zweiten Pumpentyp handelt es sich um einen weiteren Typ einer Vakuumpumpe, genauer gesagt eine Membranpumpe. Bei diesem Pumpentyp wird mittels Rotation über Gummimembranen ein Vakuum erzeugt, welches das zu befördernde Medium über ein Einwegventil einsaugt und über ein weiteres Einwegventil wieder abgibt. Eine solche Pumpe ist in Abbildung 3.20 zu sehen. [21]



Abbildung 3.20: Ansichtsbild einer Membranpumpe [22]

Der grosse Vorteil dieses Pumpentyps ist die hohe Lebensdauer sowie der geringe Preis. Auch bei diesem Pumpentyp kann mittels Drehzahl über die Zeit die zu befördernde Menge berechnet werden. Dies weist jedoch auch nicht die erforderliche Genauigkeit gemäss dem Pflichtenheft auf. Deshalb muss auch bei diesem Pumpentyp eine externe Durchflussmessung durchgeführt werden. [21]

Schlauchpumpe:	Membranpumpe:
<ul style="list-style-type: none"> - Spannung: 24V - Stromverbrauch: ca. 700mA - Durchfluss: bis 800ml/min - Messung des Durchflusses: extern - Preis der verglichenen Pumpe: 70.86\$ - Bestellseite: Aliexpress 	<ul style="list-style-type: none"> - Spannung: 12V - Stromverbrauch: <600mA - Durchfluss: 1000-1500ml/min - Messung des Durchflusses: extern - Preis der verglichenen Pumpe: 6.37\$ - Bestellseite: Aliexpress

Tabelle 3.1: Vergleich der zwei Pumpentypen [22][23]

3.6.3 Verwendete Pumpe

Beide Pumpen entsprechen mindestens den Wunschzielen gemäss dem Pflichtenheft. Allerdings ist die mechanische Abnutzung beider Pumpentypen verschieden. So wird bei der Schlauchpumpe der Schlauch relativ stark beansprucht, was zur Folge hat, dass dieser mit der Zeit ausgewechselt werden muss. Die Membranpumpe hingegen bietet da eine höhere Lebensdauer. Der grösste Unterschied bietet jedoch der Preis. Da die Schlauchpumpe mehr als 10 Mal so viel kostet, wie die Membranpumpe und davon 12stk. benötigt werden, wird die Membranpumpe eingesetzt. Ausserdem kann mit der Schlauchpumpe nur das Pflichtziel gemäss Pflichtenheft umgesetzt werden und mit der Membranpumpe auch das Wunschziel.

3.7 Durchflussmessung

Da gemäss Kapitel 3.6 Pumpen eingesetzt werden, welche nicht über eine integrierte Durchflussmessung verfügen, muss diese extern gemessen werden. Dies wird mittels Durchflusssensoren umgesetzt. Dazu werden verschiedene Möglichkeiten verglichen.

3.7.1 Anforderungen

Durchflussmenge	:	Aus Pflichtziel	min. 0.6l pro Minute
		Aus Wunschziel	min 1.2l pro Minute
Spannung	:	Spannungsversorgung	3.3V bis 5V
Schlauchanschluss	:	Innendurchmesser	6-8mm
Sonstiges	:	Hygieneanforderungen	Lebensmittelsensor

Es existieren viele verschiedene Varianten um eine Durchflussmessung durchführen zu können. Dazu gehören folgende Verfahren:

- akustische Verfahren
- gyroskopische Verfahren
- magnetisch-induktive Verfahren
- mechanisch-volumetrische Verfahren
- optische Verfahren
- thermische Verfahren
- Wirkdruck-/Stauverfahren

Viele dieser Verfahren werden nur bei grösseren Durchflussmengen eingesetzt. Wieder andere werden verwendet um den Durchfluss von verschiedenen Medien zu bemessen. Da es sich bei diesem Projekt um relativ geringe Durchflussmengen von weniger als 2l/min handelt eignet sich ein mechanisch-volumetrisches Verfahren am besten. [24]

3.7.2 Verwendeter Durchflusssensor

Beim verwendeten Durchflussmessgerät handelt es sich um einen sogenannten Flügelradzähler von Sea gemäss Abbildung 3.21. Dies ist ein hermetisch abgeschlossener Durchflussgeber, dessen Kernstück aus einem Flügelrad besteht. Bei Durchfluss wird dieses Flügelrad in Bewegung gesetzt, welches über einen Hallsensor elektronische Impulse erzeugt. Somit kann die durchfließende Menge elektronisch bestimmt werden. [25]

Technische Daten:
- Spannung: 3.3 - 5V
- Stromverbrauch: ca. 15mA
- Messbarere Durchflussmenge: 0.3-6l/min
- Preis: 2.09 CHF
- Bestellseite: Aliexpress

Tabelle 3.2: Technische Daten Sea Durchflussmessgerät [25]



Abbildung 3.21: Sea Flügelradzähler [25]

3.8 Display

Um die Maschine bedienen zu können, wird ein Display benötigt. Die Verwendung von Buttons soll verhindert werden. Aus diesem Grund wird ein Touch-Display implementiert.

3.8.1 Anforderungen

Kommunikation	:	UART oder SPI
Spannung	:	5V oder 3.3V
Bedienung	:	Touch-Panel
Grösse	:	± 10cm Diagonale
Sonstiges	:	Hintergrundbeleuchtung, eingebauter SD-Kartenslot vorteilhaft

3.8.2 In Frage kommende Displays

Folgende Displays kommen gemäss den Anforderungen für die Cocktailmaschine in Frage:

Displaynr.	Hersteller	Herstellernr.	Vertrieb	Kosten
1	Geekcreit®	TFT Touch Display	Banggood	CHF 12.78
2	4D Systems	4LCD-28320240-RTP	Mouser	CHF 18.76
3	4D Systems	4LCD-35480320-RTP	Mouser	CHF 23.77
4	Matrix Orbital	EVE2-43A-BLM-TPR	Mouser	CHF 48.04
5	Nextion	NX8048T070	Banggood	CHF 61.44

Display	1	2	3	4	5
Kommunikation	SPI	SPI	SPI	SPI	UART
Versorgungssp.	3.3V/5V	2.8V	2.8V	3.3V	5V
Strom	-	25mA	25mA	50mA	510mA
Vertikal	2.8"	2.8"	3.5"	4.3"	7"
Grösse (mm)	78 x 53	50 x 69.2	54.5 x 83	105.4 x 67.1	154.08 x 85.92
Pixel	240 x 320	240 x 320	320 x 480	480 x 272	800 x 480
Touch Panel	Resistive	Resistive	Resistive	Resistive	Resistive
Hintergrundbel.	LED	LED	LED	LED	LED
SD-Karte	Ja	Nein	Nein	Nein	Ja

Tabelle 3.3: Displays im Vergleich

3.8.3 Verwendetes Display

Um die Cocktailmaschine bedienen zu können, wird das Nextion-Display (5) gemäss Tabelle 3.3 eingesetzt. Das Nextion-Display ist zwar das teuerste Display in der Vergleichsserie, hat jedoch zwei grosse Vorteile. Zum Einen ist dies die grosse Entwicklungsumgebung und die vorhandene Community. Für die Displays von Nextion besteht eine Software, mit welcher relativ einfach GUI's (Graphical User Interface's) erstellt werden können. Dies erleichtert den Programmieraufwand ungemein. Des Weiteren ist die Bedienung auf einem 7-Zoll Display für diesen Zweck am benutzerfreundlichsten empfunden worden.

3.9 Mikrocontroller

Der Mikrocontroller ist dafür da, die verwendeten Bauteile gemäss Systemanforderungen anzusteuern und auszulesen. Er ist folglich dafür da, die Eingaben auf dem Display zu verarbeiten, die Getränke zu speichern und die zu bewegenden Peripherien zu steuern.

3.9.1 Anforderungen

Flüssigkeitsbeförderung	:	Pumpen	- 12x Digital Outputs
		Durchflussmessung	- 12x Digital Inputs
Motorentreiber	:	Kommunikation	- SPI
		Status	- 1x Digital Output
		EN_IN	- 1x Digital Input
Resolver	:	PWM 8kHz	- 1x Digital Output PWM
Display	:	Kommunikation	- UART
Computer USB	:	Kommunikation	- UART

3.9.2 Total benötigte Anschlüsse

Ins Gesamt sind folglich folgende Anschlüsse nötig:

- 14 Digital Outputs
- 13 Digital Inputs
- 1 PWM Pin
- 1 SPI
- 3 UART

3.9.3 In Frage kommende Mikrocontroller

Aufgrund einer ziemlich grossen Community und damit auch weit verbreiteter Anwendungsbereichen sowie Forumtauglichkeit wurden Mikrocontroller des Herstellers Atmel untersucht. Folgende Mikrocontroller erfüllen die Mindestanforderungen an das System:

- Atmega 640
- Atmega 1280
- Atmega 2560

In der folgenden Tabelle 3.4 sind die Eigenschaften der Mikrocontroller gegenübergestellt.

Mikrocontroller	ATMEGA640-16AU	ATMEGA1280-16AU	ATMEGA2560-16AU
Preis	CHF 7.93 (Digikey)	CHF 10.84 (Digikey)	12.09 (Digikey)
Kerngrösse	8-bit	8-bit	8-bit
Geschwindigkeit	16MHz	16MHz	16MHz
Konnektivität	EBI/EMI, I2C, SPI, UART	EBI/EMI, I2C, SPI, UART	EBI/EMI, I2C, SPI, UART
I/O	86	86	86
FLASH-Grösse	64KB	128KB	256KB
Spannung	2,7 bis 5,5V	2,7 bis 5,5V	4,5 bis 5,5V
Analogwandler	16x10b	16x10b	16x10b
Gehäuse	100-TQFP	100-TQFP	100-TQFP

Tabelle 3.4: Gegenüberstellung der in Frage kommenden Mikrocontroller

3.9.4 Verwendeter Mikrocontroller

In der Tabelle 3.4 ist ersichtlich, dass sich die Mikrocontroller nur in der Speichergrösse sowie Versorgungsspannung unterscheiden. Unter der Annahme, dass die Programmierung des Displays einen grossen Teil des Speichers einnimmt, wird ein Chip mit grossem FLASH-Speicher bevorzugt. Weiter besteht mit diesem Chip die Möglichkeit, mit einem Entwicklungsboard wie dem Arduino Mega, einen Prototyp des gesamten Systems zu simulieren. Deshalb wird für dieses Projekt der **ATMEGA2560-16AU** verwendet.

4 Detailkonzept

Im Detailkonzept werden die einzelnen Schaltungsteile aufgezeigt und erläutert. Dabei handelt es sich um die komplette Hardware mit den dazugehörigen Schemata. Dazu zählt die Speisung, der Mikrocontroller, die Motorenansteuerung, die Pumpenansteuerung sowie die Displayansteuerung.

4.1 Speisungen

Wie dem Grobkonzept in Kapitel 3.1 entnommen werden kann, werden für das System vier verschiedene Speisungen benötigt. Es handelt sich dabei um eine 48V-, eine 12V-, eine 5V- und eine 3.3V Spannungsquelle. Auf diese Speisungen und deren Realisierung wird in den folgenden Unterkapiteln eingegangen.

4.1.1 48V Speisung

Um die benötigten 48V des Motors sicher zu stellen, wird ein fertiges Netzteil extern eingekauft. Auch alle weiteren Speisungen werden aus den 48V generiert. Dabei muss jedoch abgeschätzt werden, wie viel Leistung dieses 48V Netzteil zur Verfügung stellen muss. Dazu wird in folgender Auflistung eine Leistungsabschätzung gemacht.

Bauteil	Beschreibung	Referenz	U	Max. I	
Motor	Stillstandstrom	[26, S.4]	48V	I_S	= 5.41A
	Nennstrom	[26, S.4]	48V	I_N	= 5.21A
TMC6200	Motorspannung	[27, S.36]	48V	$I_{VS}(24V)$	= 7mA
	Analog Versorgung	[27, S.36]	12V	$I_{VSA}(24V)$	= 8mA
	IC Versorgung	[27, S.36]	3.3V/5V	$I_{VCC}(5V)$	= 6mA
	Digital Versorgung	[27, S.36]	3.3V/5V	$I_{VIO}(5V)$	= 30uA
TMC4671	IC Versorgung	[16, S.140]	3.3V	I_{IO}	= k.A
	Analog Referenz	[16, S.141]	5V	U_{Ref}	= 5V
Resolver	TCA0372 Versorgung	[28, S.3]	5V	$I(25^\circ C)$	= 14mA
	TCA0372 Output		5V	I_{Spule}	= 90mA
	MC33202	[29, S.3]	5V	$I(25^\circ C)$	= 2.25mA
Mikrocontroller	IC Versorgung	[30, S.355]	5V	$I_{V_{max}}$	= 200mA
Level-Shifter	IC Versorgung	[31, S.3]	3.3V	$I_{V_{max}}$	= 150mA
ESP32	IC Versorgung	[32, S.21]	3.3V	$I_{V_{max}}$	= 68mA
Display	Versorgung	[33]	5V	$I_{V_{max}}$	= 510mA
Pumpen	Arbeitsspannung	[22]	12V	$I_{V_{max}}$	= 600mA
Durchflusssensor	Versorgung	[25]	5V	$I_{V_{max}}$	= 15mA

Die grossen Verbraucher, welche in das Gewicht fallen, sind in diesem Fall die Pumpen, das Display und der Förderbandmotor. Beim Förderbandmotor muss jedoch auch beachtet werden, dass dieser niemals unter Vollbelastung arbeiten wird. Auch bei den Pumpen wird jeweils immer nur eine arbeiten. Somit ergibt sich eine Leistung von ca. 16.44W für alle Komponenten ohne den Motor. Dies beinhaltet eine Pumpe, zwölf Durchflussmessgeräte, das Display und alle integrierten Schaltkreise. Da der Motor nie unter Vollast arbeiten wird, wird mit maximal 2A Arbeitsstrom gerechnet. Dies ergibt zusätzlich eine Leistung von 96W bei 48V.

Um auf der sicheren Seite zu sein wird daher ein Netzteil benötigt, welches bei 48V ca. 150W Leistung liefert. Dies entspricht bei 100% Wirkungsgrad der einzelnen Bauteile einem Ausgangstrom von 3,125A. Eingesetzt wird ein 48V/500W Netzteil von Aliexpress. Der Grund für diese Überdimensionierung ist, dass der Preis sich lediglich um 2CHF unterscheidet und man auf diese Weise in jedem Fall abgesichert ist. Dieses kann in Abbildung 4.1 begutachtet werden. [34].



Abbildung 4.1: Ansichtsbild des 48V Netzteils [34].

4.1.2 12V Speisung

Die 12V Speisung wird mittels Schaltspannungsregler realisiert. Es handelt sich hierbei um einen Regler von Monolithic Power Systems. Genauer gesagt um den MP24943DN-LF. Die Auswahl ist auf dieses Bauteil gefallen, da eine relativ hohe Eingangsspannung von 48V verarbeitet werden muss. Der MP24943DN-LF kann am Eingang mit Spannungen von 4.5-55V arbeiten und dabei eine Ausgangsspannung von 0.8-45V erzeugen. Dies bei einem maximalen Strom von bis zu 3A. Die Realisierung der 12V Speisung kann in Abbildung 4.2 betrachtet werden. [35]

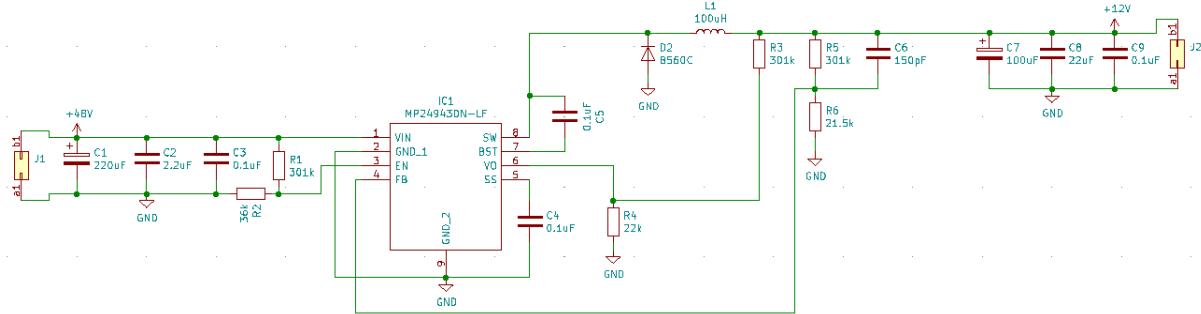


Abbildung 4.2: Schema der 12V Speisung

Bei den Kondensatoren C1-C3 handelt es sich um Filterkondensatoren, welche hochfrequente Störungen der Eingangsspannung von 48V glätten. Der Enable Eingang wird mit dem Spannungsteiler R1 & R2 auf aktiv gesetzt. [35]

Die gewünschte Ausgangsspannung wird mittels Spannungsteiler R5 & R6 eingestellt, welche auf den Feedback Eingang rückgekoppelt werden. Diese berechnet sich laut Datenblatt gemäss Formel 4.1. [35]

$$R6 = \frac{R5}{\frac{V_{out}}{0.8} - 1} \quad (4.1)$$

Bei einem Widerstandsverhältnis von $R_5=301\text{k}\Omega$ & $R_6=21.5\text{k}\Omega$ entspricht dies einer Ausgangsspannung von 12V.

Um einer Überspannung vorbeugen zu können, wird am Eingang VO ein Spannungsteiler implementiert. Diese wird am VO-Eingang mit einer Referenzspannung von 0.9V verglichen. Übersteigt die Spannung an VO die Referenzspannung von 0.9V, so wird der Regler ausgeschaltet, bis die Spannung wieder unter 0.9V fällt. Als maximale Ausgangsspannung wurde hierbei eine Spannung von 13V gewählt. Diese Wahl wurde getroffen, da die 12V ausschliesslich für die Ansteuerung der Pumpen verwendet wird und diese eine Spannung von 13V verkraften können ohne Schaden zu nehmen. Der Spannungsteiler wird gemäss Datenblatt mit der Formel 4.2 berechnet.

$$R4 = \frac{R3}{\frac{V_{ovp}}{V_{ovref}} - 1} \quad (4.2)$$

Bei einem Widerstandsverhältnis von $R_3=301\text{k}\Omega$ & $R_4=22\text{k}\Omega$ entspricht dies einer Überspannungsschutzwelle von 13.21V. [35]

Der Rippel des Spulenstroms lässt sich gemäss Formel 4.3 berechnen. Dieser sollte gemäss Datenblatt ca. 30% des maximalen Ausgangsstroms von 3A betragen. [35]

$$L1 = \frac{Vout * (Vin - Vout)}{Vin * \Delta IL * fosc} \quad (4.3)$$

Der interne Oszillator läuft dabei bei einer Frequenz von 100kHz. Bei der ausgewählten Spule von $100\mu\text{H}$ erhalten wir ein ΔI_L von 0.9A. Außerdem wird im Datenblatt darauf hingewiesen, dass die gewählte Spule auf mindestens 125% des maximalen Ausgangsstroms von 3A ausgelegt werden soll. Auch der Gleichstromwiderstand der Spule sollte $\leq 200\text{m}\Omega$ sein. [35]

Mit den Kondensatoren C7, C8 & C9 wird die Ausgangsspannung zum Abschluss noch geglättet. Bei den Eingangskondensatoren, sowie den Ausgangskondensatoren sollte es sich um low ESR Typen handeln. [35]

4.1.3 5V Speisung

Bei der 5V Speisung wurde erneut auf den MP24943DN-LF von Monolithic Power Systems gesetzt. Die Beschaltung ist dabei die gleiche, wie bei der 12V Speisung. Der Unterschied ist jedoch, dass für die Ausgangsspannung und für die Überspannungsschwelle andere Widerstandsverhältnisse gewählt werden müssen. Auch die Spule muss anders ausgelegt werden. In Abbildung 4.3 ist das Schema der 5V Speisung zu sehen. [35]

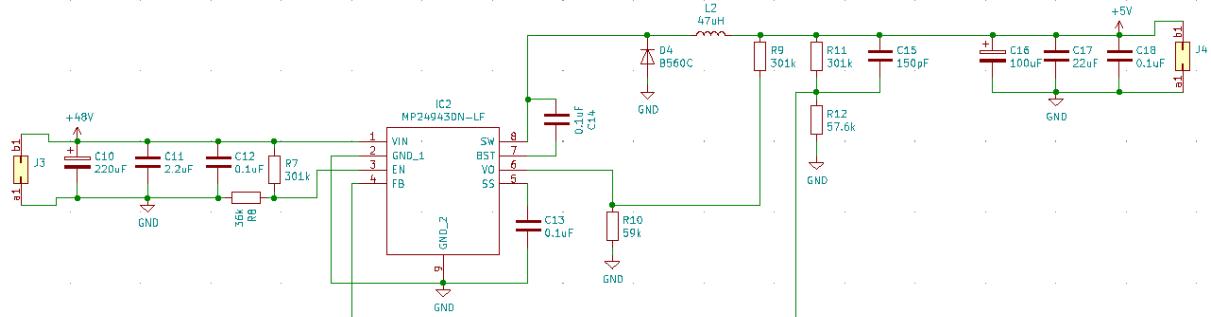


Abbildung 4.3: Schema der 5V Speisung

Das Widerstandsverhältnis von R11 & R12, welches die Ausgangsspannung definiert, wurde gemäss Formel 4.1 berechnet. Somit ergeben sich für $R11=301\text{k}\Omega$ und für $R12=57.6\text{k}\Omega$, Was einer Ausgangsspannung von 4.98V entspricht. [35]

Beim Überspannungsschutz muss darauf geachtet werden, dass der Mikrokontroller AtMega2560-16AU nur in einem Spannungsbereich von 4.5V-5.5V betrieben werden darf. Die maximal verträgliche Eingangsspannung liegt laut Datenblatt bei 6V. Somit muss der Überspannungsschutz so gestaltet werden, dass die Schwelle von 6V nicht überschritten werden kann. Um dies erreichen zu können, wurde für $R9=301\text{k}\Omega$ und $R10=53\text{k}\Omega$ gewählt. Gemäss Formel 4.2 erhält man so eine Überspannungsschutzwelle von 6V. [35]

Der interne Oszillator läuft wiederum bei einer Frequenz von 100kHz. Bei der ausgewählten Spule von $47\mu\text{H}$ erhalten wir ein ΔI_L von 0.953A. Auch hier gilt gemäss Datenblatt, dass die gewählte Spule auf mindestens 125% des maximalen Ausgangsstroms von 3A ausgelegt werden soll. Auch der Gleichstromwiderstand der Spule sollte $\leq 200\text{m}\Omega$ sein. [35]

Mit den Kondensatoren C16, C17 & C18 wird die Ausgangsspannung zum Abschluss noch geglättet. Bei den Eingangskondensatoren, sowie den Ausgangskondensatoren sollte es sich um low ESR Typen handeln. [35]

4.1.4 3,3V Speisung

Um die Treiber der Motorenansteuerung ansteuern zu können, wird zusätzlich eine 3,3V Speisung verbaut. Diese muss gemäss der Leistungsabschätzung in Kapitel 4.1.1 einen maximalen Strom von ca. 250mA liefern können. Dazu wird ein Linearregler eingesetzt, welcher von der 5V Speisung aus betrieben wird. Es handelt sich um den LF33CDT-TRY von STMicroelectronics. Dieser hat eine fixe Ausgangsspannung von 3,3V, bei einem maximalen Strom von 1A. [36]

4.2 Resolver Interface

Wie in Kapitel 3.3 beschrieben wurde, benötigt der Resolver ein Sinus-Signal, welches als Referenzsignal für die Sin- und Cos-Spule dient. Ebenso muss das zurückkehrende Signal gefiltert und verstärkt werden. Dazu wird auf ein Resolver-Interface der Firma NXP zurückgegriffen. Dieses hat drei Verstärker-Schaltkreise. Der erste ist der Sinus erzeugende Schaltkreis, die beiden anderen sind Filter- und Verstärkungsschaltkreis. Jeweils einer für das Sinus- und Cosinus-Signal.

4.2.1 Problem

Das zu lösende Problem besteht in der Transformation eines Rechtecksignals, welches vom Mikrocontroller gegeben wird, in ein Sinussignal, welches für den Resolver gebraucht wird. Der Verstärkerschaltkreis für die rückkehrenden Signale muss gewährleisten, dass das Signal in einem Bereich von 1..4V liegt. Dafür benötigt das Signal einen Offset von 2.5V.

4.2.2 Schaltungsaufbau

Der Opamp IC700 A transformiert das Rechtecksignal mittels Integrator in ein Dreiecksignal. Der Opamp IC700 B integriert das Dreiecksignal und erzeugt ein sinusähnliches Signal. Die Widerstandsverhältnisse R1/R3 und R2/R4 beeinflussen die Linearität des Integrators. Je höher dieses Verhältnis ist, desto besser wird das Sinussignal am Ausgang. Ist das Verhältnis jedoch zu hoch, wird die Schaltung störungsanfälliger. Um ein Signal mit hoher Qualität zu erreichen, sollte der Duty-Cycle des Eingangssignals genau 50% betragen.[37]

Als Vorlage dient das Resolver-Interface der Firma NXP. Die Schaltung ist in Abbildung 4.4 dargestellt.

Opamp1:

Das Dreiecksignal soll innert einer halben Periodendauer um drei Volt steigen. Das heisst, es ergibt sich gemäss Formel 4.4 eine Slew-Rate von $0.048\text{V}/\mu\text{s}$. Der auf 5.6nF dimensionierte Kondensator wird so gemäss Formel 4.5 einen maximalen Strom von $269\mu\text{A}$ leiten. [38]

$$\text{Slew - Rate}_1(\text{SR}_1) = \frac{\Delta U_1}{T/2} = \frac{3\text{V}}{62.5\mu\text{s}} = 0.048 \frac{\text{V}}{\mu\text{s}} \quad (4.4)$$

$$I_{C_2} = C_2 \cdot \text{Slew - Rate}_1 = 5.6 \cdot 10^{-9} \cdot 0.048 \frac{\text{V}}{10^{-6}\text{s}} = 269\mu\text{A} \quad (4.5)$$

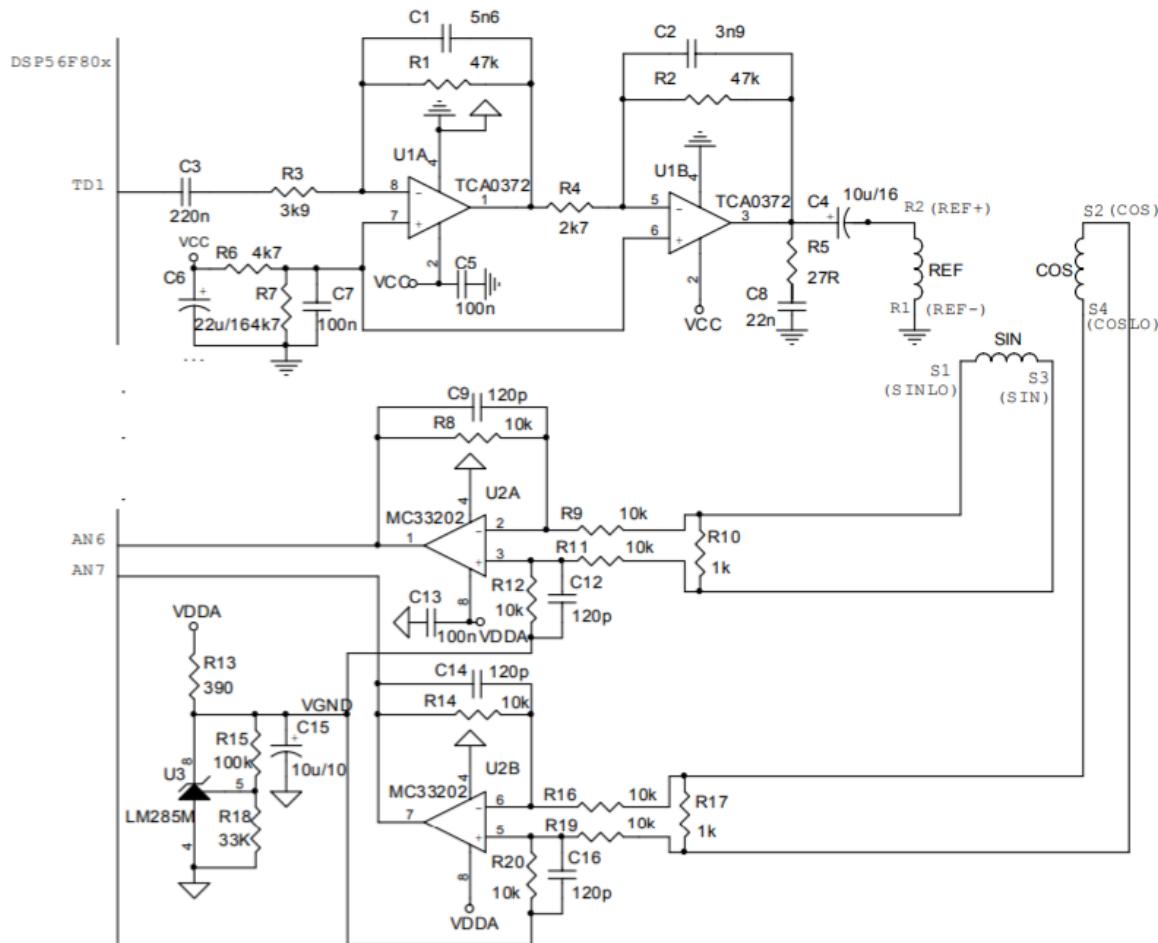


Abbildung 4.4: Schema Resolver der Firma NXP.[37]

So kann gemäss Formel 4.6 der Widerstand R3 dimensioniert werden. Dessen Wert beträgt ungefähr 9300Ω . [38]

$$R3 = \frac{\pm 2.5V}{I_{C1}} = \frac{\pm 2.5V}{269\mu A} = 9.3k\Omega \quad (4.6)$$

Mit dem Widerstand R1 und der Kapazität C2 kann die obere Grenzfrequenz bestimmt werden. Diese Größen wurde aus dem Datenblatt des NXP-Interfaces übernommen und ergaben zusammen gemäss Formel 4.7 eine Frequenz von:

$$\omega_{g1} = \frac{1}{R2 \cdot C2} = 3799 rad^{-1} \quad (4.7)$$

Opamp2:

Der zweite Opamp integriert das Dreiecksignal zu einem sinusförmigen Signal. Da das Ausgangssignal für die Spule 5V betragen soll und diese Änderung auch in 62.5µs geschehen soll,

benötigen wir eine Slew-Rate von $0.08V/\mu s$, wie mit Formel 4.8 berechnet. Der daraus maximal resultierende Strom ergibt mit der in Formel 4.9 $624 \mu A$.

$$Anstiegsrate_2(SR_2) = \frac{\Delta U_2}{T/2} = \frac{5V}{62.5\mu s} = 0.08 \frac{V}{\mu s} \quad (4.8)$$

In Formel 4.9 ist der Faktor 2 zu beachten. Dieser wird verwendet, da der Eingangsstrom dreieckförmig ist. Grundlage ist, dass die Spannung über dem Kondensator das Integral des Stromes über die Zeit ist. So ergibt sich beim Integrieren eine Fläche mit einer Spannungsamplitude von 3V mal eine Zeit von $62.6\mu s$ geteilt durch 2.

$$I_{C3} = C_3 \cdot Slew - Rate_2 \cdot 2 = 3.9 \cdot 10^{-9} \cdot 0.080 \frac{V}{10^{-6}s} \cdot 2 = 624 \mu A \quad (4.9)$$

Aus dem Maximalstrom kann der Widerstand R4 dimensioniert werden. Er beträgt ziemlich genau 2400Ω .

$$R4 = \frac{\pm 1.5V}{I_{C3}} = \frac{\pm 1.5V}{624\mu A} = 2.4k\Omega \quad (4.10)$$

Mit dem Widerstand R2 und der Kapazität C3 kann die obere Grenzfrequenz bestimmt werden. Diese Größen wurden aus dem Datenblatt des NXP-Interfaces übernommen und ergaben zusammen gemäss Formel 4.11 eine Frequenz von:

$$\omega_{g2} = \frac{1}{R2 \cdot C3} = 5456 rad^{-1} \quad (4.11)$$

Opamp3

Das zurückkommende Signal muss auch gefiltert und verstärkt werden, sodass es in einem Spannungsbereich zwischen 1V und 4V liegt. Das Übersetzungsverhältnis des Resolvers beträgt 2:1, weswegen eine Spannung von $\pm \Delta U$ von $\pm 1.25V$ erwartet wird. Der Differenzverstärker hat somit den Verstärkungsfaktor gemäss Formel 4.12 von 1.2.

$$V = 3V/2.5VV = \frac{U_A}{U_E} = \frac{3V}{2.5V} = 1.2 \quad (4.12)$$

Daraus kann der Widerstand R12 berechnet werden. Dieser Beträgt gemäss Formel 4.13 $12k\Omega$.

$$R12 = V \cdot R9 = 1.2 \cdot 10k\Omega = 12k\Omega \quad (4.13)$$

Die Kondensatorgrösse wurde aus dem Datenblatt von NXP übernommen. Damit wird jedoch die Grenzfrequenz festgelegt. Diese ergibt mit R12 und C6 gemäss Formel 4.14 644444 rad^{-1}

$$\omega_{g2} = \frac{1}{R12 \cdot C6} = 644.44 krad^{-1} \quad (4.14)$$

4.2.3 Simulation

Die berechneten Werte wurden in der Simulation eingesetzt. Da die Opamps und der Kondensator C1 die Schaltung beeinflussen, wurden kleine Anpassungen an den Werten von R3 und R4 gemacht, um die gewünschten Effekte zu erlangen. Die Simulationsergebnisse werden im Folgenden aufgelistet:

- C1 = 220nF
- C2 = 5.6nF
- C3 = 3.9nF
- R1 = 9kΩ
- R2 = 47kΩ
- R3 = 47kΩ
- R4 = 2.5kΩ

Abbildung 4.5a zeigt die Opamp-Schaltung für das Referenzsignal. Darin enthalten sind die beiden Verstärkerschaltungen in Serie geschaltet. Beide haben einen Offset von 2.5V. Der Kondensator C1 bildet einen Hochpass, wodurch sich die Spannungen in der gesamten Schaltung ein wenig anheben und es eine Weile dauert, bis die Schaltung eingeschwungen ist. Die passiven Elemente R5 und C4 verhindern ein ungewolltes Schwingen des Ausgangs, wenn induktive Lasten angetrieben werden. Abbildung 4.5b zeigt die Differenzverstärkerschaltung auf Empfängerseite.

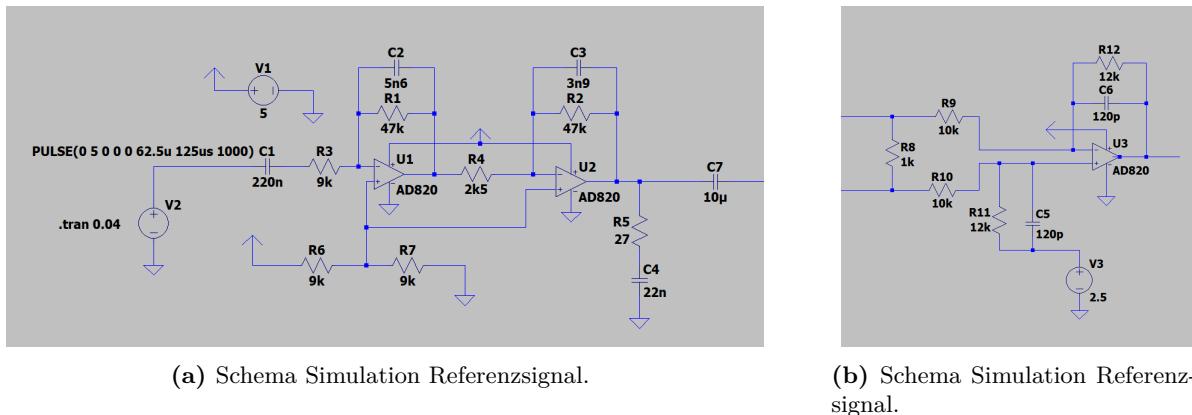


Abbildung 4.5: Schema Simulation Referenzsignal Resolver.

In Abbildung 4.6 ist der Einschwingvorgang zu erkennen. Das gesamte System ist nach etwa 20 ms eingeschwungen. Das blaue Signal ist die Spannung zwischen dm Kondensator C1 und R1, das grüne Signal ist das Dreiecksignal am Ausgang des ersten Opamps und das rote Signal ist das Sinus-Signal am Ausgang des zweiten Opamps.

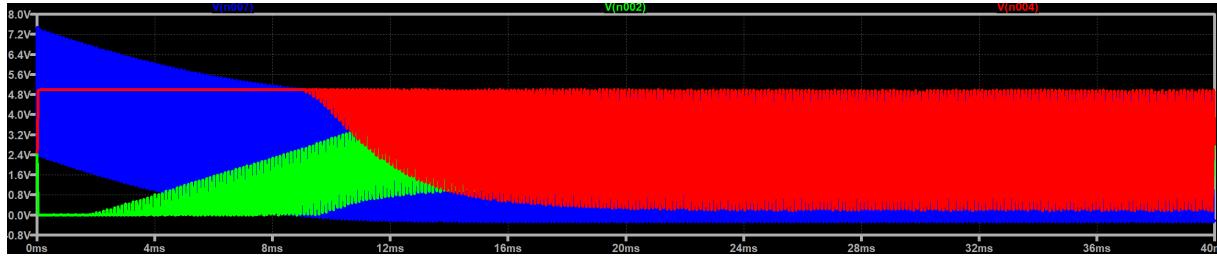


Abbildung 4.6: Einschwingvorgang

In den Abbildungen 4.7 und 4.8 ist die Spannung und der Strom am und durch den Punkt zwischen C1 und R3 zu sehen. Der Strom durch C1 und R1 ist folglich der selbe. Die Spannung an diesem Punkt ist das Eingangssignal für die Integratorschaltung am Opamp1.

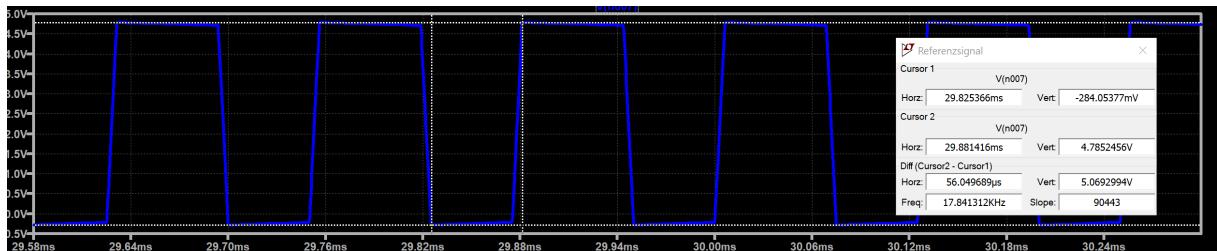


Abbildung 4.7: Eingangs PWM-Signal zwischen C1 und R1. Zu erkennen: Durch den Strom durch R3 fällt eine Spannung über dem Widerstand ab. Der Spannungsabfall lässt sich im Rest der Schaltung nicht gross bemerken.

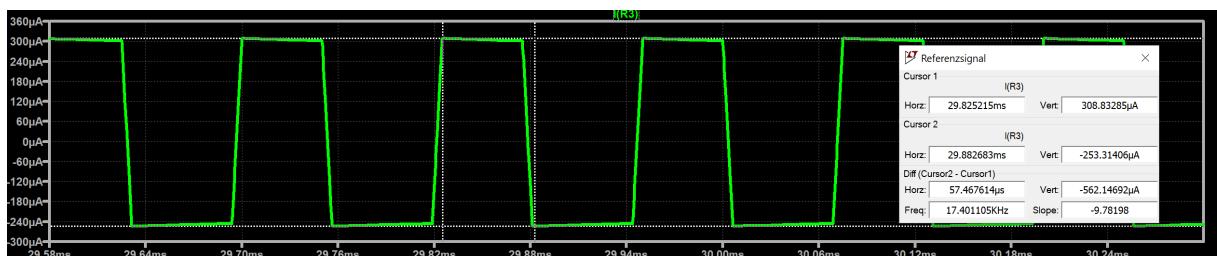


Abbildung 4.8: Eingangs-Strom durch R3. Zu erkennen: Der halbe Strom I_{P-P} beträgt $281\mu A$ und ist somit leicht höher als in Formel 4.5 berechnet.

In den Abbildungen 4.9 und 4.10 ist die Spannung am Ausgang vom Opamp1 und der Strom durch R4 abgebildet. Die Ausgangsspannung vom Opamp1 ist das Eingangssignal des Opamp2.

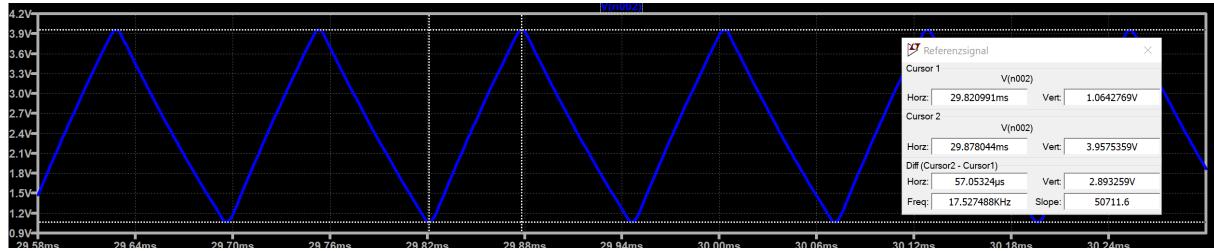
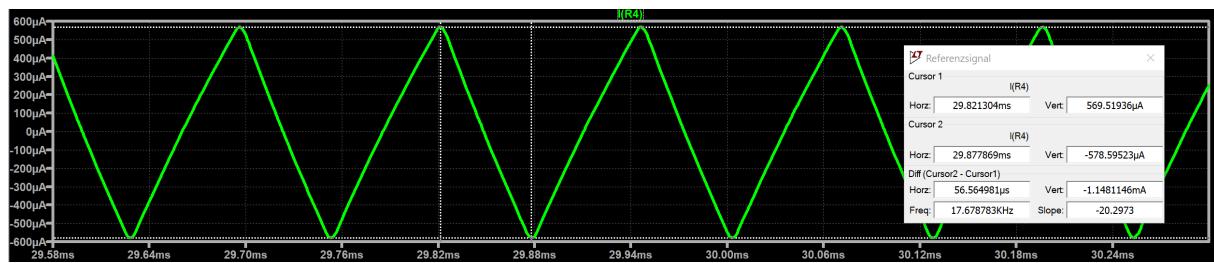
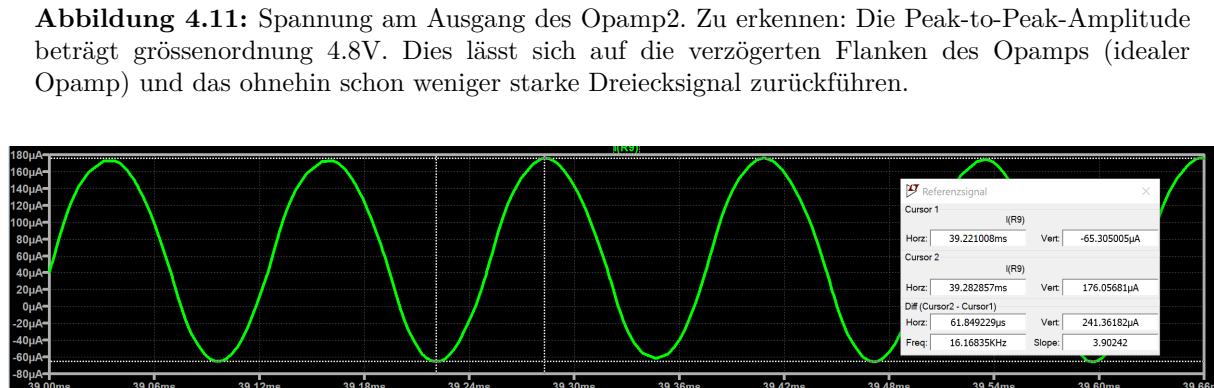
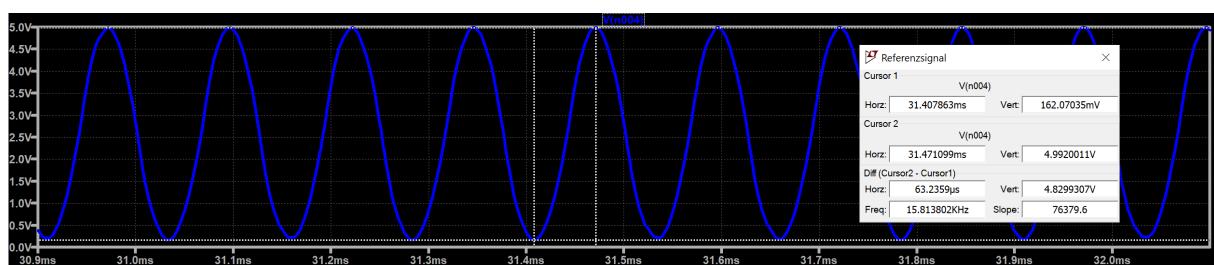


Abbildung 4.9: Spannung am Ausgang des Opamp1. Zu erkennen: Die Peak-to-Peak-Amplitude beträgt größenordnung 2.9V. Dies lässt sich auf die verzögerten Flanken des Opamps zurückführen, da mit einem idealen Opamp gerechnet wurde.



In den Abbildungen 4.11 und 4.12 ist die Spannung am Ausgang vom Opamp1 und der Strom durch R4 abgebildet. Die Spannung am Ausgang vom Opamp1 ist das Eingangssignal des Opamp2.



In den Abbildungen 4.13 und 4.14 ist die Spannung am Ausgang vom Opamp3 und der Strom durch R9 abgebildet. Die Spannung am Ausgang vom Opamp3 ist das Eingangssignal des TMC4671. Die Ausgangsspannung für den Controller ist in Abbildung 4.15 dargestellt.

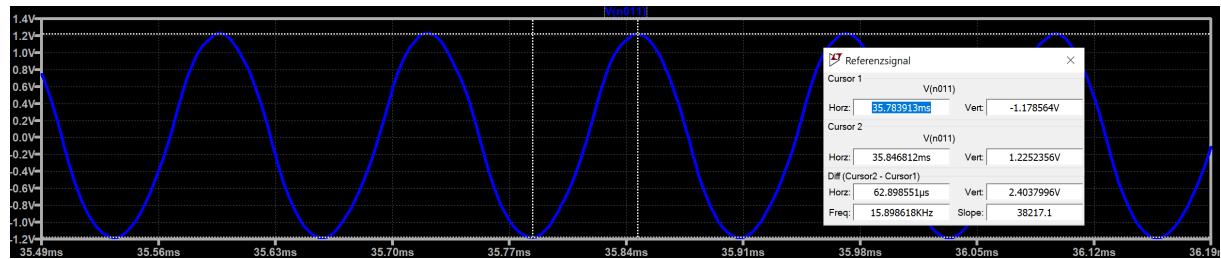


Abbildung 4.13: Eingangsspannung für Verstärkerschaltung Mikrocontrollersignal.

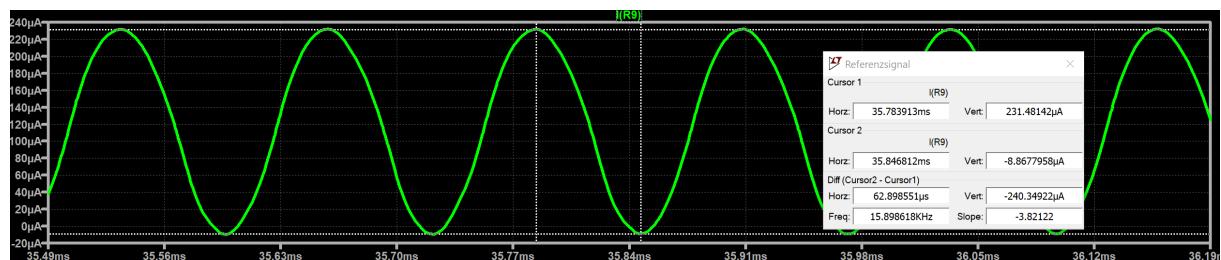


Abbildung 4.14: Strom durch Widerstand R9.

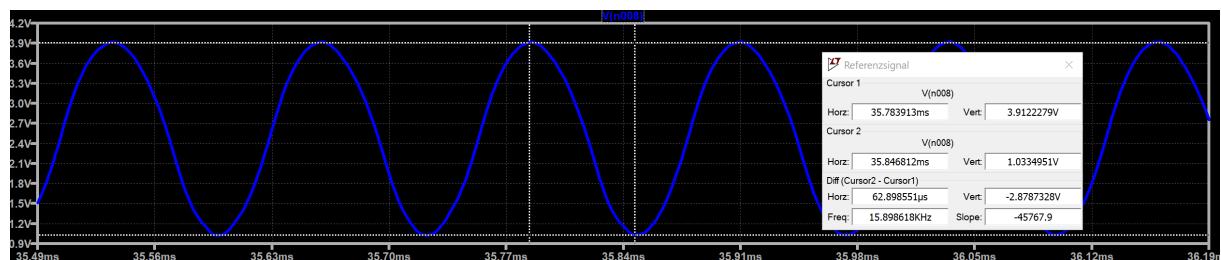


Abbildung 4.15: Ausgangsspannung für TMC4671. Zu erkennen: Das Signal liegt zwischen 1V und 4V, wie es der Controller braucht.

4.3 Pumpenansteuerung

Die 12 Pumpen werden direkt über den Mikrokontroller via Digitalausgänge angesteuert. Da jedoch der Mikrokontroller nicht genügend Strom liefern kann, wird ein Logik-MOSFET eingesetzt. Bei dem MOSFET handelt es sich um einen IRLR8726 von Infineon. Mit diesem wurde schon in früheren Projekten gearbeitet, bei welchen sich dieser als zuverlässig herausstellte. Dieser MOSFET zeichnet sich durch einen geringen R_{DSon} von $5.8\text{m}\Omega$, eine kleine Gatekapazität von 15nC , einer V_{DS} Breakdownvoltage von 30V und einer hohen Strombelastbarkeit von maximal 86A aus. Die Ansteuerung ist relativ simpel aufgebaut und kann in Abbildung 4.16 begutachtet werden.

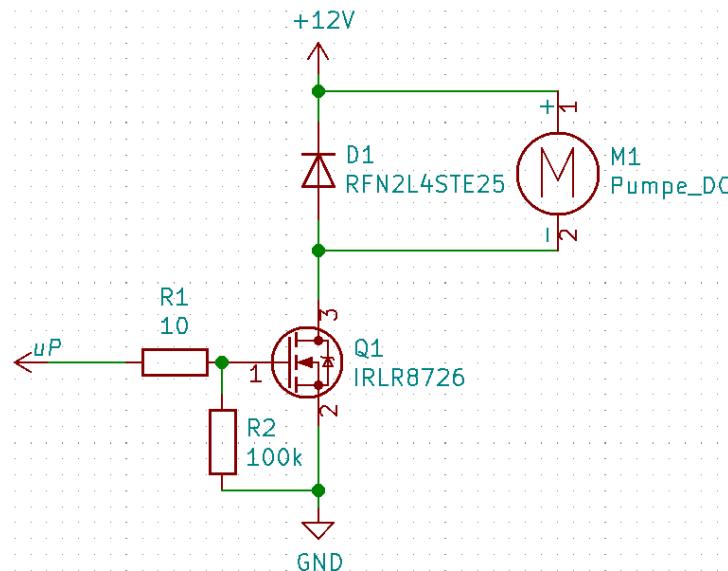


Abbildung 4.16: Schema der Pumpenansteuerung

Um den Gate-Eingang zu schonen, wird ein 10Ω Widerstand vgeschaltet. Parallel zur Pumpe ist eine Freilaufdiode verbaut, welche den Drain-Eingang der MOSFETS beim Ausschalten der Pumpe vor Spannungsspitzen schützt.

4.4 Durchflussmessgeräte

Um die Durchflussmessgeräte ansteuern zu können, wird keine zusätzliche Elektronik benötigt. Es wird lediglich eine 5V Speisung sowie ein Analog- oder Digitaleingang eines Mikrocontrollers benötigt. Dabei wird jedoch ein Digitaleingang bevorzugt, da keine unterschiedlichen Spannungen eingelesen werden müssen, sondern lediglich High- und Lowpegel. Das Durchflussmessgerät sendet somit bei laufender Umdrehungszahl ein gepulstes Signal, welches mit steigendem Durchfluss die Frequenz erhöht. Dies geschieht bei gleichbleibendem Duty-Cycle von 50%.

4.5 Display

Beim Display handelt es sich um das in Kapitel 3.8 evaluierte 7"Display von Nextion. Genauer gesagt um den Typ NX8048T070. Dieses wird über eine UART-Schnittstelle angesteuert und wird mit 5V gespiesen. Das verwendete Display ist in Abbildung 4.17 zu sehen.



Abbildung 4.17: Ansichtsbild des verwendeten Display's. [39]

Um mit diesem Displaytyp arbeiten zu können, ist es wichtig, dass man sich einen Überblick über die Eigenschaften verschafft. Die wichtigsten Eigenschaften des Display's können in Tabelle 4.1 eingesehen werden.

Speisespannung:	5V
Arbeitsstrom:	510mA
Sleep-Mode Strom:	15mA
Arbeitstemperatur:	-20°bis 70°
Flash Speicher:	16MB
RAM Speicher:	3584BYTE
Farben:	65536 Farben
Auflösung:	800*480
Touch Typ:	resistiv
Hintergrundbeleuchtung:	LED

Tabelle 4.1: Eigenschaften des Nextion Display's [33].

Die zwei grössten Vorteile dieses Displaytyps sind die Entwicklungsumgebung, sowie die grosse Community. Nextion bietet von sich aus einen GUI-Editor (Graphical User Interface Editor) an, welcher gratis zur Verfügung gestellt wird. Mit Hilfe dieses Tools können spielerisch leicht grafische Oberflächen geschaffen werden mit verschiedensten Komponenten wie Tasten, Textfelder, Bilder, Slider und viele mehr. Die Oberfläche des Editors kann in Abbildung 4.18 begutachtet werden.



Abbildung 4.18: Entwicklungsumgebung von Nextion (Nextion Editor).

Die Software ist so einfach wie möglich gehalten. Im roten Feld 1 können Komponenten wie Slider, Textfelder, Tasten ect. per Klick in die erstellte Displayoberfläche im orangen Feld 2, im Ansichtsfenster eingefügt werden. In diesem Feld ist zu sehen, was schlussendlich auf dem Display ersichtlich ist. Es können natürlich mehrere Displayseiten erstellt werden. Diese können im gelben Feld 3 hinzugefügt, gelöscht oder umbenannt werden. Per Klick auf die gewünschte Seite wird diese im Ansichtsfenster 2 angezeigt und kann dann bearbeitet werden. Im grünen Feld 4 befinden sich die Eigenschaften der Elemente der erstellten Displayseite. Wird ein Element auf der Displayseite angeklickt, so erscheinen dort dessen Eigenschaften. Diese können dann beliebig angepasst werden. Dazu gehören zum Beispiel die Elementgrösse oder die Platzierungsposition. Im blauen Feld 5 kann den einzelnen Elementen ein Reaktionsbefehl gegeben werden. Dabei kann ausgewählt werden, ob das Element auf Drücken, Loslassen oder beides Reagieren soll. Außerdem können dort Befehle eingegeben werden. Also zum Beispiel bei einem Taster: "page 2". Dies bedeutet so viel wie: "Falls das Element gedrückt wird, dann zeige Displayseite 2 an". Natürlich können so auch Befehle über die UART-Schnittstelle weiter gegeben und empfangen werden. Das violette Feld 6 bietet eine Outputübersicht an, welche zu Simulationszwecken verwendet wird. Im letzten grauen Feld 7 können Vorlagen abgespeichert werden. Dazu gehören Bilder, Textstyle Vorlagen, Audio- und Videodateien.

4.6 User Interface

Um die Cocktailmaschine schlussendlich sauber bedienen zu können, wird das in Kapitel 5.5 beschriebene Display als Benutzerschnittstelle fungieren. Weitere Schnittstellen werden erstmal nicht verwendet. Damit der Benutzer jedoch die Maschine möglichst intuitiv und einfach bedienen kann, muss zuerst eine geeignete Menustruktur erstellt werden. Im Projekt 5 wird noch nicht die gesamte Struktur auf dem Display programmiert. Trotzdem wird die gesamte Struktur erarbeitet, damit klar ist, welche Schnittstellen benötigt werden. Somit kann ein Testprogramm erstellt werden, welches alle später benötigten Schnittstellen und Funktionen testet. Das komplette Programm wird in Projekt 6 erstellt. Die erarbeitete Menustruktur sieht wie folgt aus.

4.6.1 Startanzeige

Auf der Startanzeige 4.19 kann mit den Pfeiltasten ein gewünschter Cocktail ausgewählt werden. Dabei wird immer ein Bild des Cocktails mit dem dazugehörigen Namen angezeigt. Wird auf den Button «Zutaten» gedrückt erscheint die Zutatenanzeige 4.20. Wird auf den Button «Menu» gedrückt so gelangt man in die Menuanzeige 4.25. Wird auf den Cocktail gedrückt, so gelangt man in die Zubereitungsabfrage 4.22. Mit dem Button «Ohne Alkohol» oder «Mit Alkohol» kann zwischen alkoholischen und nicht alkoholischen Cocktails ausgewählt werden. Wird der Button «Liste» betätigt, so wird die Listenanzeige 4.21 der alkoholischen oder nicht alkoholischen Cocktails angezeigt.



Abbildung 4.19: Ansichtsbild Startanzeige

4.6.2 Zutatenanzeige

In der Zutatenanzeige 4.20 wird dem Kunden eine Übersicht über die beinhaltenden Zutaten des ausgewählten Cocktails gegeben. Um aus der Zutatenanzeige zu gelangen, kann mit «OK» bestätigt werden.



Abbildung 4.20: Ansichtsbild Zutatenanzeige

4.6.3 Listenanzeige

In der Listenanzeige 4.21 wird dem Kunden eine Liste der verfügbaren Cocktails aufgezeigt, wobei dieser mit den Pfeiltasten durchblättern und den gewünschten Cocktail auswählen kann. Wird auf einen Cocktail gedrückt, so gelangt man wieder in die Startanzeige 4.19 mit dem ausgewählten Cocktail.



Abbildung 4.21: Ansichtsbild Listenanzeige

4.6.4 Zubereitungsabfrage

In der Zubereitungsabfrage 4.22 wird der Kunde gefragt, ob er einen 3dl oder 5dl Cocktail zubereiten will. Wird «Ja 3dl» betätigt, so wird ein 3dl Getränk zubereitet. Wird «Ja 5dl» betätigt, so wird ein 5dl Getränk zubereitet und man gelangt auf den Zubereitungsbildschirm 4.23. Mit «Abbrechen» gelangt man zurück zur Startanzeige 4.19.



Abbildung 4.22: Ansichtsbild Zubereitungsabfrage

4.6.5 Zubereitungsbildschirm

Auf dem Zubereitungsbildschirm 4.23 wird der Kunde darüber informiert, dass der Cocktail zubereitet wird. Gleichzeitig wird dem Kunden ein Fact zum gewählten Cocktail gezeigt, um die Wartezeit ein wenig zu versüßen. Mit dem Button «Abbrechen» kann der laufende Prozess abgebrochen werden und es erscheint die Abbruchanzeige 4.32. Ist der Cocktail fertig, so wird dem Kunden die Bereitanzeige 4.24 angezeigt.



Abbildung 4.23: Ansichtsbild Zubereitungsbildschirm

4.6.6 Bereitanzeige

Die Bereitanzeige 4.24 zeigt dem Kunden, dass der Cocktail fertig ist und dieser nun entnommen werden kann. Dieser Bildschirm erscheint für 5 Sekunden. Nach dieser Zeit erscheint automatisch wieder die Startanzeige 4.19.



Abbildung 4.24: Ansichtsbild Bereitanzeige

4.6.7 Menuanzeige

Wird in der Menuanzeige 4.25 «Cocktail bearbeiten» ausgewählt, so kommt man in die Bearbeitungsanzeige 4.26. Wird «Maschine Reinigen» gedrückt, so gelangt man in die Reinigungsanzeige 4.28. Wird «Maschineninfo» betätigt, so gelangt man in die Infoanzeige 4.31. Mit «Zurück» gelangt man wieder auf den Startanzeige 4.19.



Abbildung 4.25: Ansichtsbild Menuanzeige

4.6.8 Bearbeitungsanzeige

In der Bearbeitungsanzeige 4.26 kann der zu bearbeitende Cocktail ausgewählt werden. Wird einer ausgewählt, so gelangt man in die dazugehörige Cocktaileinstellungsanzeige 4.27. Mit den Pfeiltasten kann zwischen den verschiedenen Cocktails navigiert werden. Wird «Zurück» betätigt, so gelangt man wieder zur Menuanzeige 4.25.



Abbildung 4.26: Ansichtsbild Bearbeitungsanzeige

4.6.9 Cocktaileinstellungsanzeige

In der Cocktaileinstellungsanzeige 4.27 kann mittels Schieberegler jede Zutat in der Menge angepasst werden. Es kann jedoch nie 100% überschritten werden. Wird «Standardeinstellung» gedrückt, so wird der Cocktail auf die jeweilige Standardeinstellung eingestellt. Betätigt man «OK», so gelangt man wieder in die Bearbeitungsanzeige 4.26.

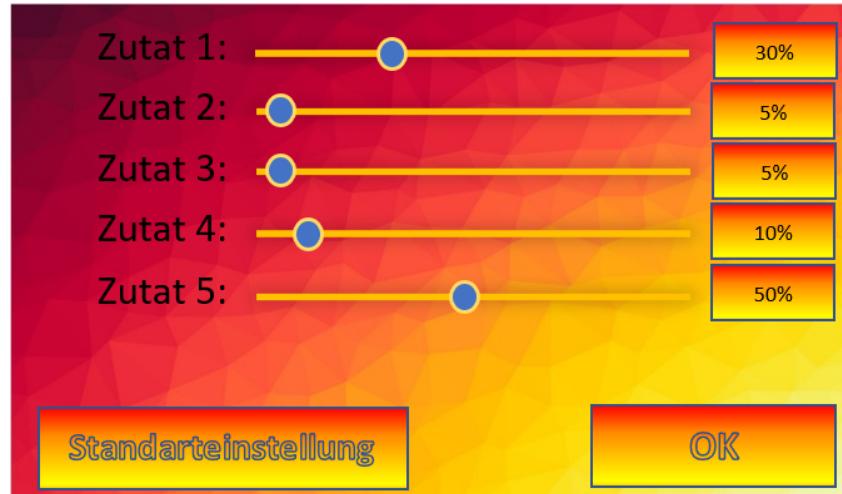


Abbildung 4.27: Ansichtsbild Cocktaileinstellungsanzeige

4.6.10 Reinigungsanzeige1

In der Reinigungsanzeige1 4.28 wird man Stück für Stück durch den Reinigungsmodus geführt. In der Reinigungsanzeige1 wird man aufgefordert den Reinigungsbehälter mit warmem Wasser zu befüllen und alle Schläuche darin ein zu setzen. Wird «Weiter» gedrückt, so gelangt man in die Reinigungsanzeige2 4.29. Wird «Abbrechen» gedrückt, so gelangt man zurück in die Menuanzeige 4.25.

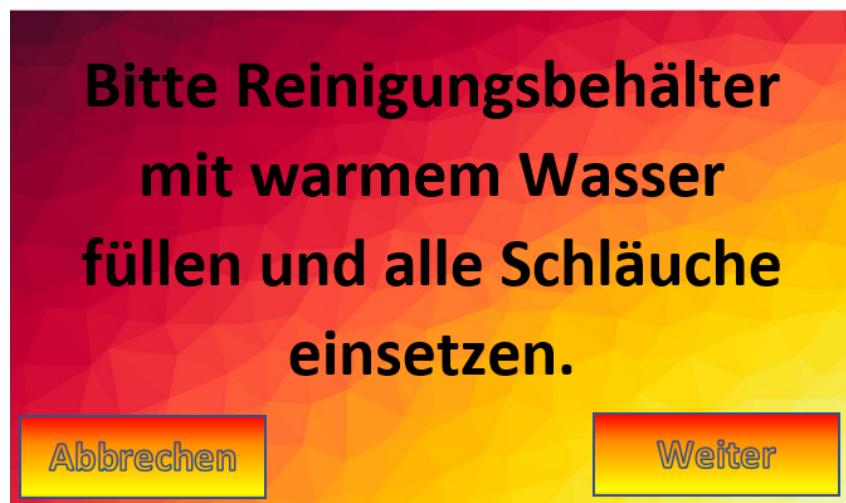


Abbildung 4.28: Ansichtsbild Reinigungsanzeige1

4.6.11 Reinigungsanzeige2

In der Reinigungsanzeige2 4.29 wird gefragt, ob der Reinigungsvorgang gestartet werden soll. Mit «Start» wird der Vorgang gestartet und man gelangt in die Reinigungsanzeige3 4.30. Wird «Abbrechen» gedrückt, so gelangt man in die Menuanzeige 4.25 zurück.



Abbildung 4.29: Ansichtsbild Reinigungsanzeige2

4.6.12 Reinigungsanzeige3

In der Reinigungsanzeige3 4.30 wird der Kunde darüber informiert, dass die Reinigung nun durchgeführt wird. Die Reinigung kann zu diesem Zeitpunkt nicht abgebrochen werden, da sonst Flüssigkeit in den Schläuchen verbleiben könnte, welche den nächsten Cocktail ungenießbar macht. Ist die Reinigung beendet, so erscheint die Startanzeige 4.19.



Abbildung 4.30: Ansichtsbild Reinigungsanzeige3

4.6.13 Infoanzeige

In der Infoanzeige 4.31 werden dem Kunden Informationen zur Cocktailsmaschine gegeben. Mit «Zurück» gelangt man erneut in die Menuanzeige 4.25.



Abbildung 4.31: Ansichtsbild Infoanzeige

4.6.14 Abbruchanzeige

Wird ein Vorbereitungsvorgang abgebrochen, so erscheint die Abbruchanzeige 4.32, welcher den Kunden darüber informiert, dass der Vorgang abgebrochen wurde. Sobald der Abbruchvorgang beendet ist, erscheint wieder die Startanzeige 4.19.



Abbildung 4.32: Ansichtsbild Abbruchanzeige

4.6.15 Fehleranzeige

Wird über eine gewisse Zeit trotz eingeschalteter Pumpen kein Durchfluss festgestellt, so erscheint die Fehleranzeige 4.33 und die Maschine stoppt. Der Benutzer wird dabei aufgefordert die Flüssigkeitsbehälter zu kontrollieren und diese gegebenenfalls wieder aufzufüllen. Ist dies erledigt, so kann mit «Erledigt» bestätigt werden und der Cocktail wird an der gestoppten Stelle fortgesetzt mit dem vorherigen Zubereitungsbildschirm 4.23.



Abbildung 4.33: Ansichtsbild Fehleranzeige

5 Hardware

Im folgenden Kapitel wird die verwendete Hardware aufgebaut und verifiziert. Dabei werden die einzelnen Schaltungsteile Ausgemessen und/oder getestet.

5.1 Speisungen

Es wurden drei verschiedene Speisungen aufgebaut, welche auf ihre Funktion geprüft wurden. Dies ist die 12V Speisung, gemäss Kapitel 4.1.2, welche zur Ansteuerung der Pumpen benötigt wird. Des Weiteren eine 5V Speisung gemäss Kapitel 4.1.3, welche für den Mikrokontroller und das Display benötigt wird und eine 3,3V Speisung für die Motorentreiber.

5.1.1 12V Speisung

Die 12V Speisung wurde unter einer Belastung von ca. 1A und bei einer Eingangsspannung von 48V ausgemessen. Um dies zu erreichen ist eine Last von 12Ω angeschlossen worden. In Abbildung 5.1 kann das Ergebnis begutachtet werden.

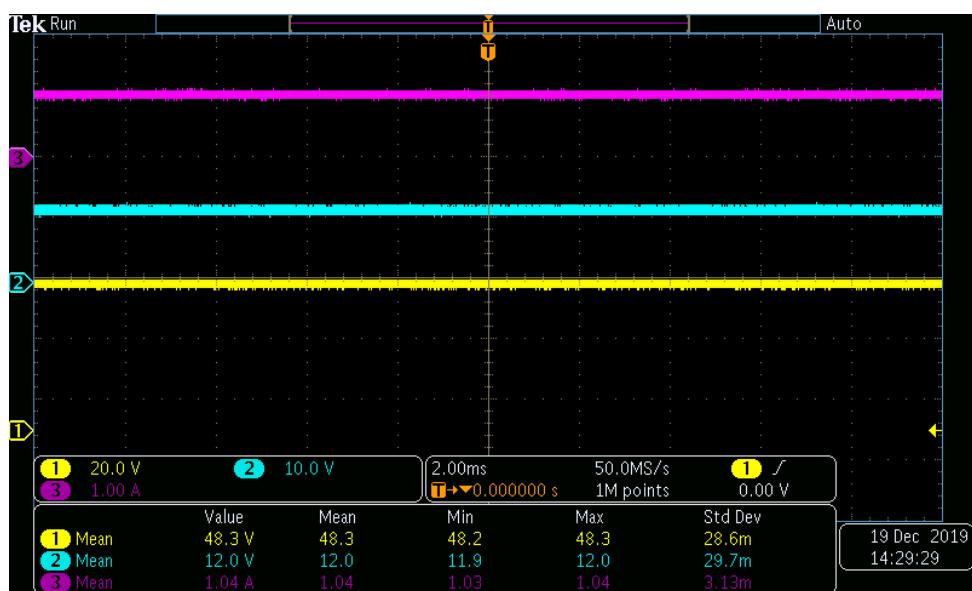


Abbildung 5.1: Messung der Ausgangsspannung bei einem Laststrom von 1A

Folgende Spannungen und Ströme sind in Abbildung 5.1 zu sehen:

- 1 Eingangsspannung (gelb)
- 2 Ausgangsspannung (blau)
- 3 Ausgangsstrom (violett)

Wie man erkennen kann, wird die Ausgangsspannung sauber auf 12V geregt. Diese bleibt auch unter grösserer Belastung stabil. Die Testkondition wurde bewusst auf 1A Ausgangsstrom gesetzt, da die Pumpen diesen Strom niemals erreichen können. Somit ist sichergestellt, dass die Funktion der Speisung voll umfänglich gewährleistet ist.

Der Schaltregler taktet die Spule gemäss Datenblatt mit einer Frequenz von ca. 100kHz. Um dies auszumessen, wurde die getaktete Spannung an V_{sw} ausgemessen. Dies ist in Abbildung 5.2 zu sehen.

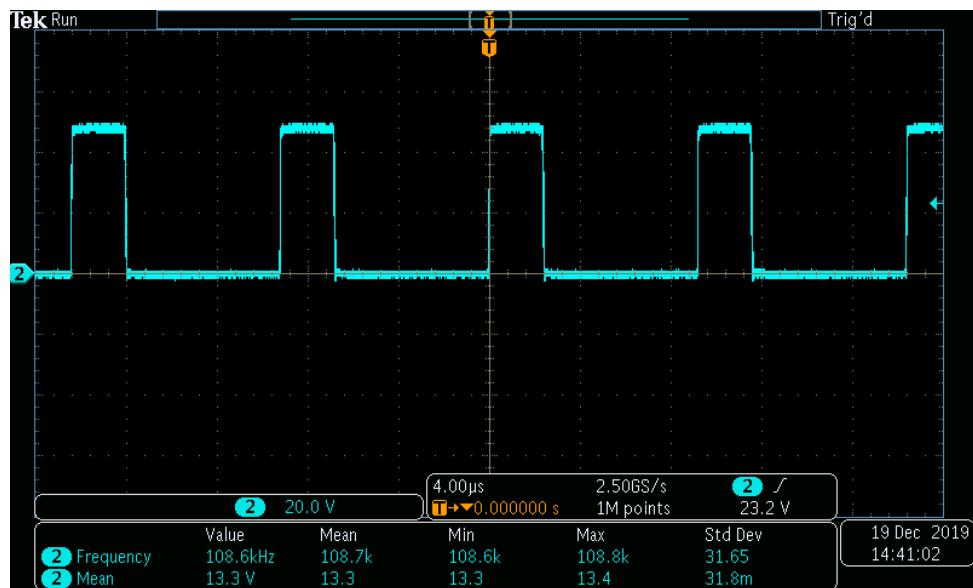


Abbildung 5.2: Messung der Schaltfrequenz des 12V Reglers

Die Taktfrequenz liegt wie erwartet bei ca. 100kHz. Bei der Messung ergab sich eine Schaltfrequenz von 108.6kHz.

5.1.2 5V Speisung

Bei der 5V Speisung handelt es sich gemäss Kapitel 4.1.3 fast exakt um den selben Aufbau, wie bei der 12V Speisung. Diese wird jedoch bei weitem nicht so stark belastet wie die 12V Speisung. Trotzdem wurde auch diese Speisung bei einem Laststrom von ca. 1A ausgemessen. In Abbildung 5.3 kann die Ausgangsspannung begutachtet werden.

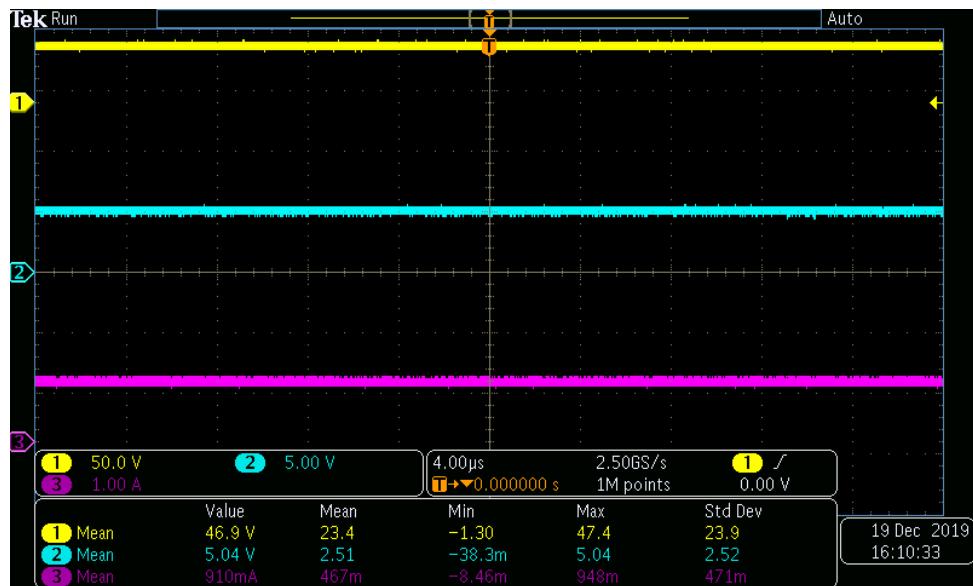


Abbildung 5.3: Messung der Ausgangsspannung bei einem Laststrom von 1A

Folgende Spannungen und Ströme sind in Abbildung 5.3 zu sehen:

- 1 Eingangsspannung (gelb)
- 2 Ausgangsspannung (blau)
- 3 Ausgangstrom (violett)

Auch bei der 5V Speisung wird die Ausgangsspannung erwartungsgemäß und zuverlässig auf 5V geregelt. Um die Schaltfrequenz des Reglers zu überprüfen, wurde auch hier wieder V_{sw} ausgemessen. Dies ist in Abbildung 5.4 zu sehen.

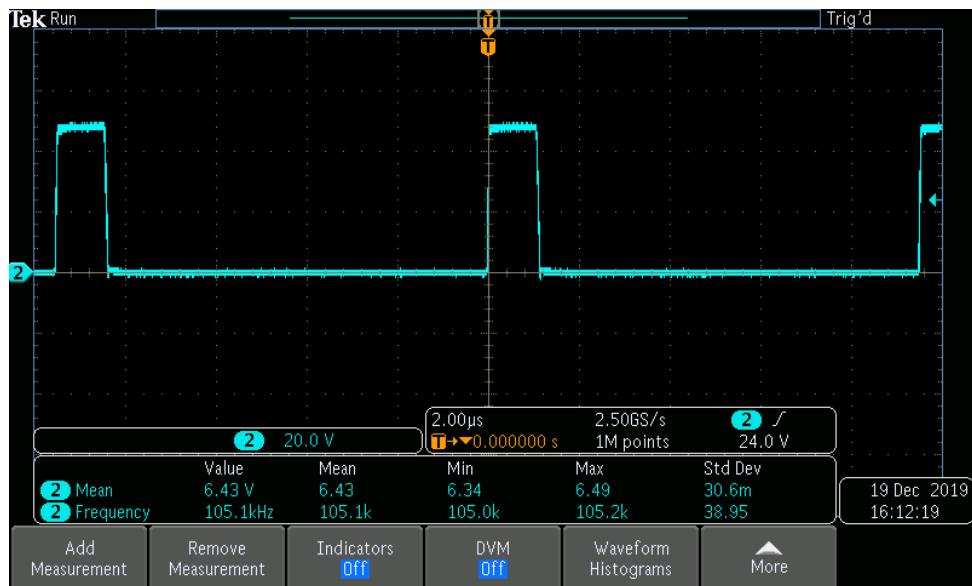


Abbildung 5.4: Messung der Schaltfrequenz des 5V Reglers

Auch hier wurde die Erwartung von ca. 100kHz erfüllt. Die Messung ergab eine Schaltfrequenz von 105.1kHz.

5.2 Motorenansteuerung

Im Folgenden wird die Motorenansteuerung verifiziert. Die Ansteuerung gliedert sich in zwei Teile, einerseits das Feedback, andererseits der Motorentreiber. Deshalb wird nur auf diese Bereiche eingegangen.

5.2.1 Resolver

Der Ausgang des PWM-Pins beträgt 8kHz, wie in den Abbildungen 5.5a und 5.5b ersichtlich.

Das nachgebaute Resolver-Interface ergab folgende Messungen:

Abbildung 5.6 zeigt das PWM-Signal hinter dem Kondensator. Das hier anliegende Signal wird über dem Opamp integriert.

Das Integrierte Signal ist in Abbildung 5.7 zu erkennen. Die Amplitude beträgt 3.12V, dies liegt am Maximalwert der Spannungsspitze, ansonsten entspricht die Messung der Erwartung aufgrund der Simulation.

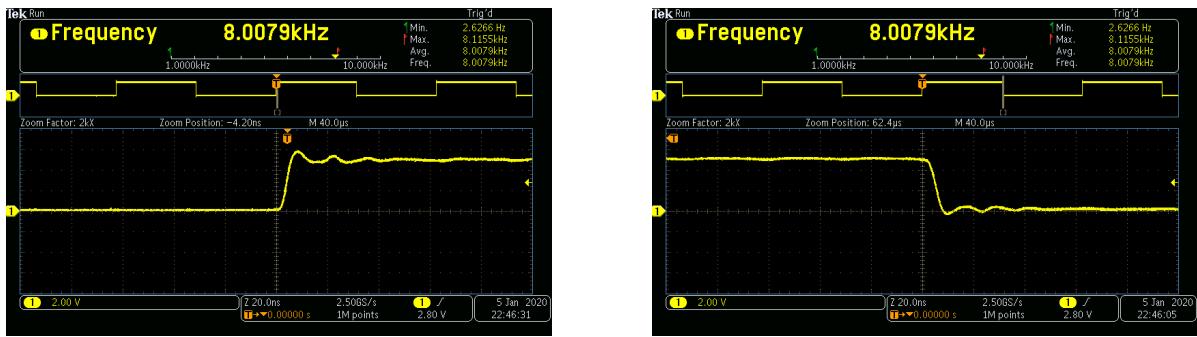


Abbildung 5.5: Schematische und grafische Darstellung eines Resolvers.

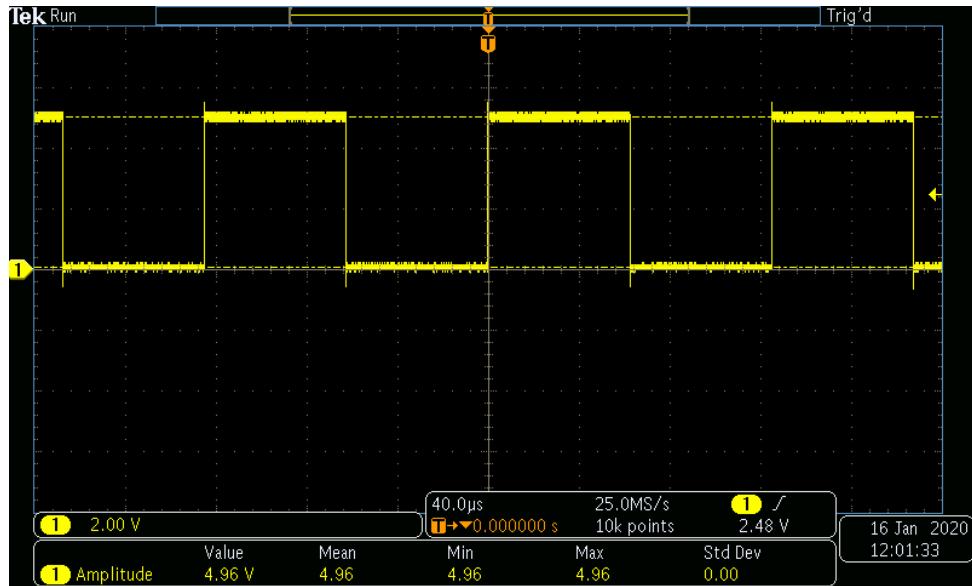


Abbildung 5.6: Messung zwischen Opamp U1 und Widerstand R4.

Abbildung 5.8 zeigt das angenäherte Sinussignal, welches auf die Erregerspule geführt wird. Das Signal hat mit $3.6V_{PP}$, im Gegensatz zur Simulation mit $4.8V_{PP}$, eine zu kleine Amplitude. Es ist zu erkennen, dass das Signal oben und unten abgeschnitten ist. Die Gründe dafür konnten nicht herausgefunden werden, können bei Bedarf im Projekt 6 herausgefunden werden. Da die Rede war, für das Projekt 6 einen Motor mit ABN-Encoder zu verwenden, könnte auf diesen Schritt verzichtet werden. Ein Einfluss könnten die unterschiedlichen Widerstände machen, welche den Spannungsteiler am positiven Anschluss des Opamps bilden.

Abbildung 5.9 zeigt das einkommende Signal der Sinusspule am Resolver. Die Signalform hat sich durch die Induktivitäten im Resolver verbessert. Es hat eine Amplitude von $1.18V_{PP}$, was im Gegensatz zur simulierten Amplitude von $2.4V_{PP}$. Somit ist die Übertragungsamplitude nicht 2:1, sondern 3.6:1.12. Möglicherweise liegt dies am nicht sauberen Ausgangssignal und einer kleinen Verschiebung der Messspule zur Erregerspule.

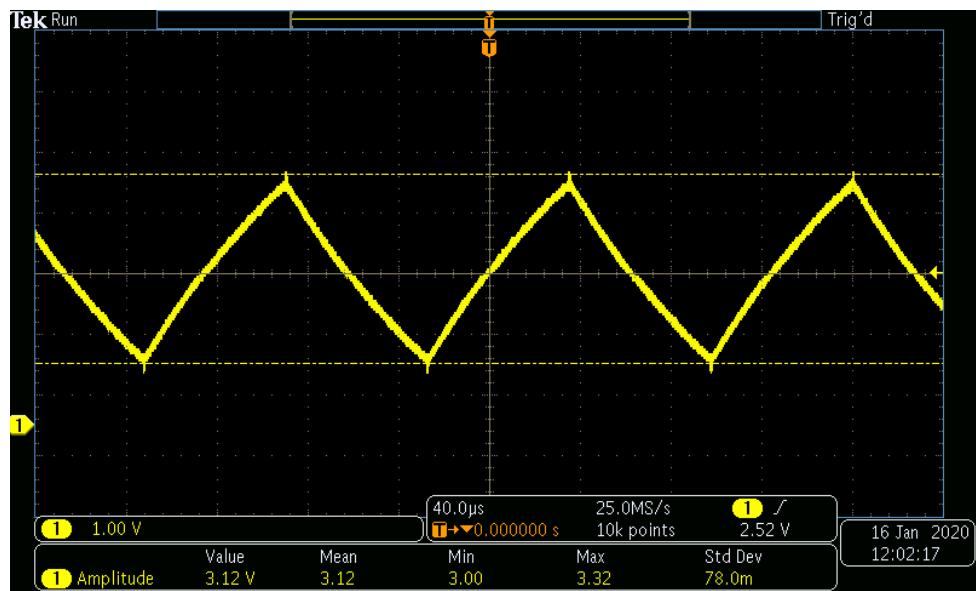


Abbildung 5.7: Dreiecksignal nach dem ersten Opamp.

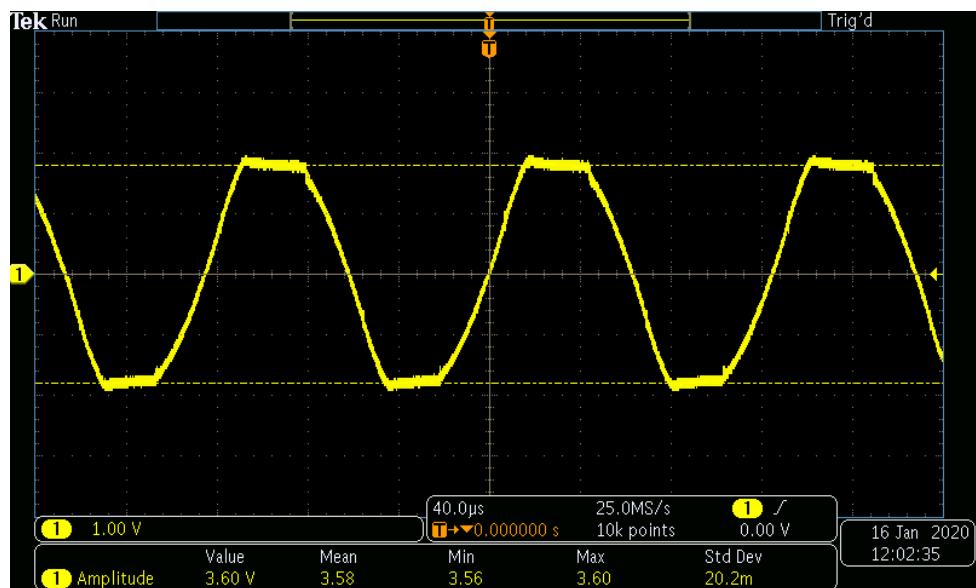


Abbildung 5.8: Angenähertes Sinussignal nach dem zweiten Opamp.

Abbildung 5.10 zeigt das aufbereitete Signal für den Motorentreiber. Es hat eine Signalamplitude von $1.4V_{PP}$ und einen Offset von $2.5V$, was einen Spannungsbereich von $1.8V_{PP}$ bis $3.2V_{PP}$ ergibt. Dies liegt im Bereich des Treibers und kann theoretisch skaliert werden für die Verarbeitung der Signale. Die Schaltung ist somit ausreichend für den Resolver, kann jedoch noch verbessert werden.

Abbildung 5.11 zeigt das Signal der Cosinus Messpule. Es hat eingangsseitig eine Amplitude von $1.18V_{PP}$ und ist auch hier tiefer als die errechneten $2.4V_{PP}$.

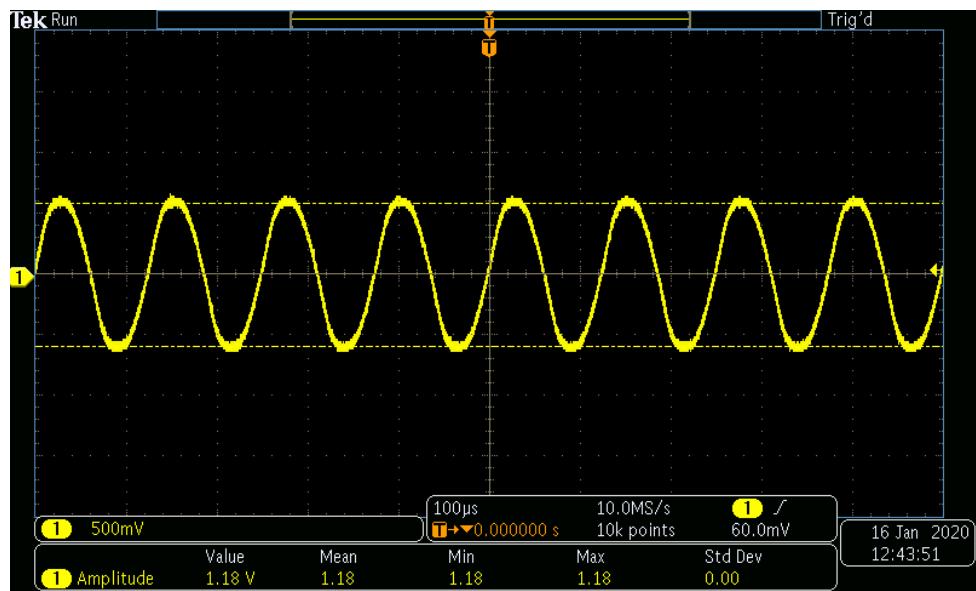


Abbildung 5.9: Rückkehrendes Signal aus Resolver.

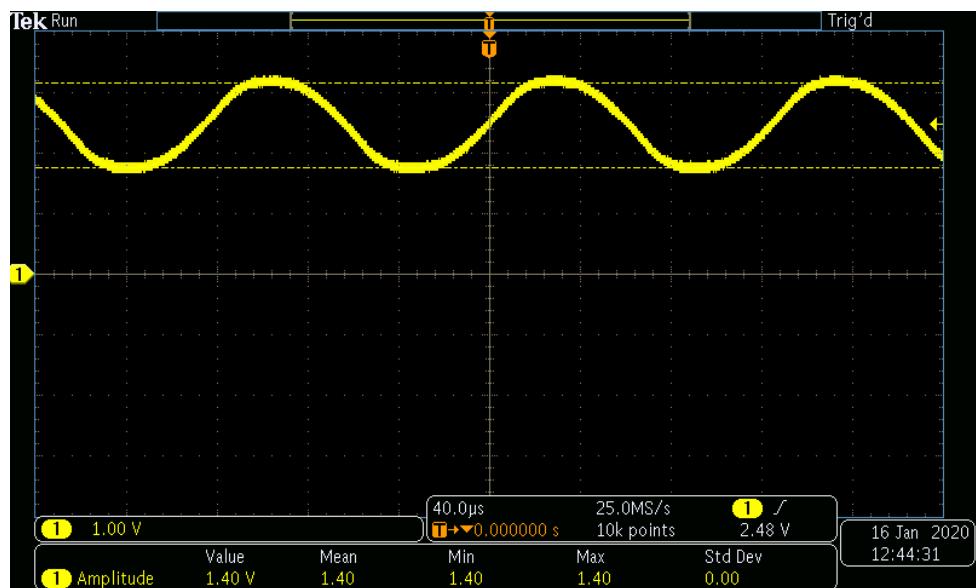


Abbildung 5.10: Messung am Sinus-Eingang, aufbereitet für Mikrocontroller.

Abbildung 5.12 zeigt das aufbereitete Signal für den Motorentreiber. Es hat eine Signalamplitude von $1.4V_{PP}$ und einen Offset von $2.5V$, was einen Spannungsbereich von $1.8V_{PP}$ bis $3.2V_{PP}$ ergibt. Dies liegt im Bereich des Treibers und kann theoretisch skaliert werden für die Verarbeitung der Signale. Die Schaltung ist somit ausreichend für den Resolver, kann jedoch noch verbessert werden.

Eine Implementierung des eingebauten Resolvers hat leider nicht geklappt, da die Eingangssignale nicht richtig verarbeitet werden konnten. So konnte der Motor nur im Torque- und Flux-Modus betrieben werden, was sehr zum Nachteil der Cocktaillmaschine wäre. Darum wurde entschieden, mit einem ABN-Encoder zu arbeiten, um uns trotzdem mit dem Chip und seinen Möglichkeiten vertraut zu machen.

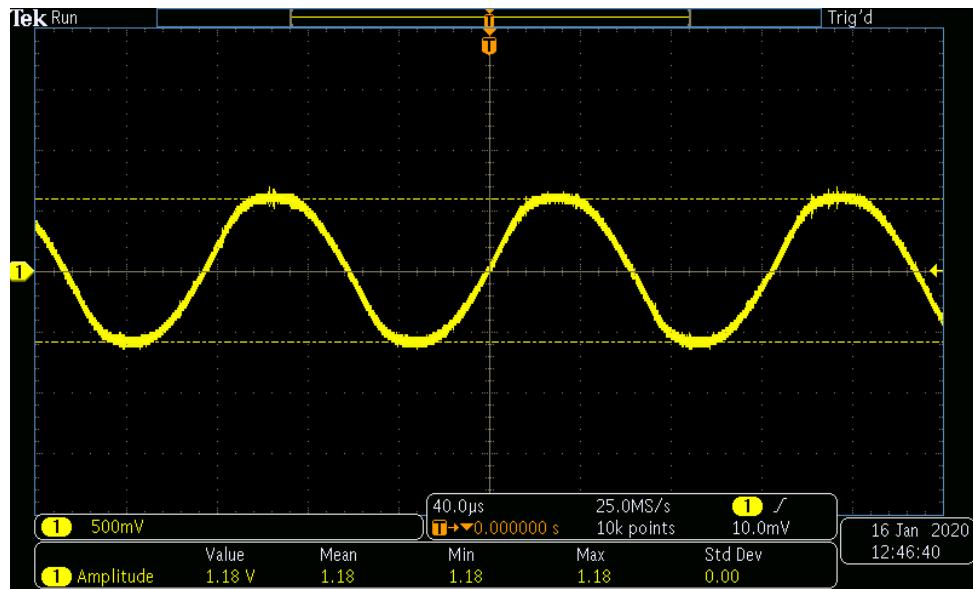


Abbildung 5.11: Messung am Cosinus-Eingang, aufbereitet für Mikrocontroller.

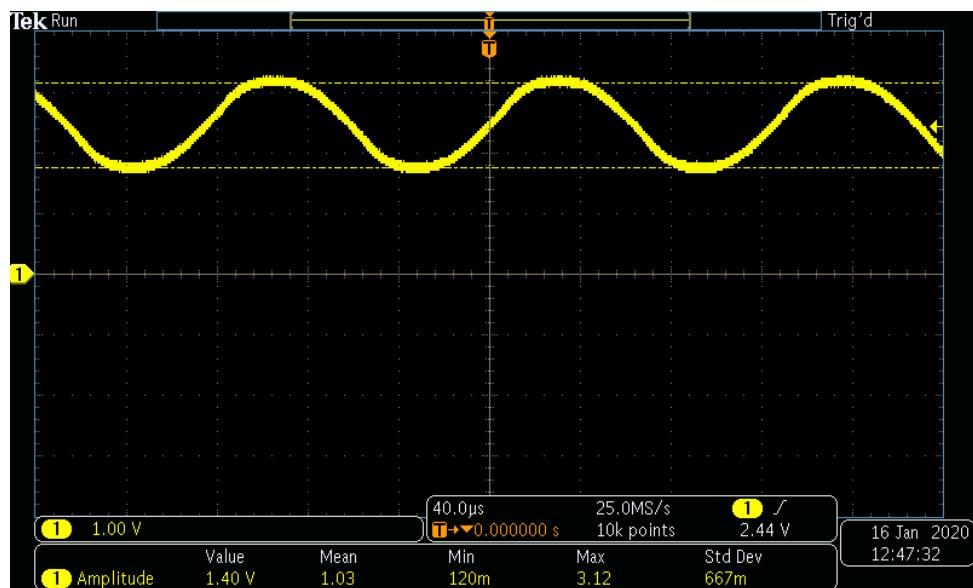


Abbildung 5.12: Messung am Sinus-Eingang, aufbereitet für Mikrocontroller.

5.2.2 TMC4671

Für die Ansteuerung wurde die Software *TMCL-IDE 3.0* verwendet. Anfangs wurde der Motor in den Openloop versetzt. Danach wurden die Analogeingänge für die Strommessung skaliert, gefolgt von der Initialisierung des ABN-Encoders. Mit schrittweisem Erhöhen der PI-Parametern war es möglich, den Motor in verschiedenen Modi zu steuern.

Der Torque-/Flux-Modus konnte mit Einstellen deren PI-Parametern gesteuert werden, bei Belastung drehte der Motor langsamer, das Drehmoment blieb das selbe.

Mit setzen der PI-Paramter des Velocity-Modus, war der Treiber in der Lage, bei ändernder Belastung den Rotor stets auf einer konstanten Drehzahl zu halten, indem er das Drehmoment erhöhte.

Bei Aktivierung des Position-Modus und setzen dessen PI-Parametern, konnte der Treiber den Rotor je nach Wert um einen gewissen Winkel drehen.

Für den Bewegungsablauf ist es möglich, die Limits der Parameter zu setzen, um das Bewegungsverhalten zu kontrollieren. Im Falle der Cocktailmaschine dürfen die Bewegungen nicht zu abrupt sein, aber dürfen bei längeren Strecken auch zügiger vorangehen.

Aufgrund technischer Schwierigkeiten mit dem EVAL-Board, konnten leider keine Bilder der Software während des Betriebs gemacht werden.

5.3 Pumpenansteuerung

Die Pumpenansteuerung gemäss Kapitel 4.3 besteht lediglich aus einem Logic-FET, einer Diode und zwei Widerständen. Die Anforderungen an die Schaltung sind für den Verwendungszweck relativ gering. In Abbildung 5.13 kann der Schaltmoment begutachtet werden.

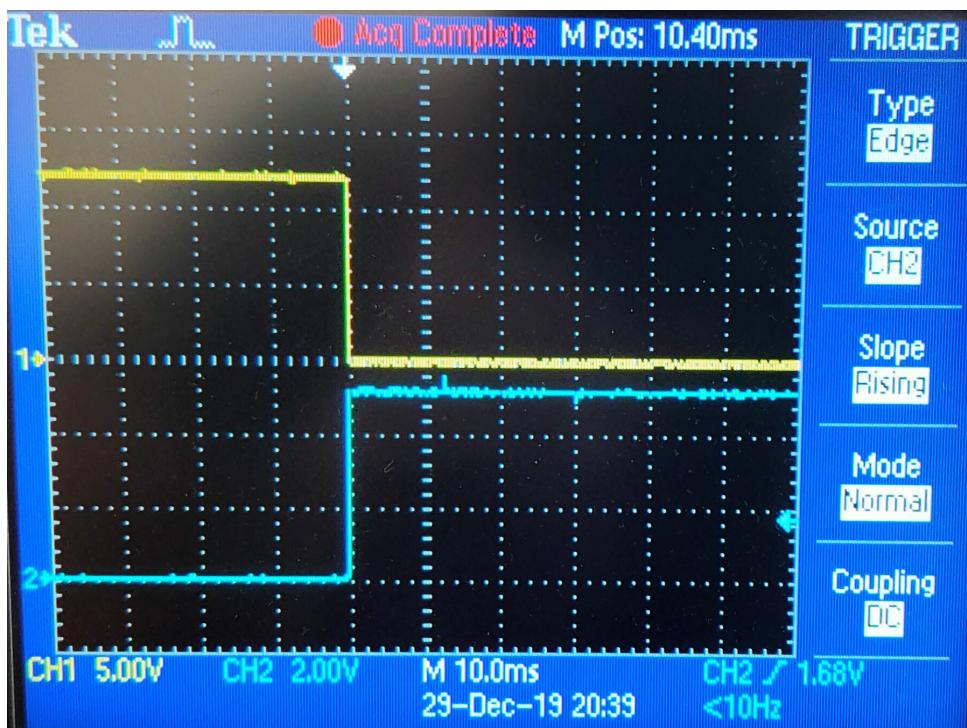


Abbildung 5.13: Messung der Schalteigenschaft der Pumpenansteuerung

Folgende Spannungen und Ströme sind in Abbildung 5.13 zu sehen:

- 1 Speisespannung am Drain
- 2 Gatespannung

Es wird Wert darauf gelegt, dass der Logic-FET bei erfolgter Ansteuerung komplett durchsteuert. Die Anstiegszeit der Schaltflanke ist für diesen Zweck irrelevant. Daher wird diese auch nicht weiter detailliert verfolgt.

5.4 Durchflussmessgeräte

Um zu sehen wie exakt mit den Durchflussmessgeräten gearbeitet werden kann, wurden einige Versuchsreihen durchgeführt. Dabei ist in einem ersten Schritt ein Programm in C erstellt worden, welches eine Pumpe über die in Kapitel 4.3 erwähnte Pumpenansteuerung ein- und ausschalten kann. Danach wurde ein Digitaleingang implementiert, welcher die erzeugten Pulse des Durchflussmessgerätes bei erfolgendem Durchfluss einlesen kann. Somit ist man nun in der Lage, durch das Zählen der Pulse, den Durchfluss zu bestimmen.

Da vom Händler bei Aliexpress kein Datenblatt abgelegt wurde, wurde in einer ersten Phase ermittelt, wie viele Pulse bei einer gewünschten Menge von 250ml eingelesen werden. Dabei ergab sich ein Wert von 981 Pulsen bei 250ml. Danach wurde diese Füllmenge 10 Mal abgefüllt und die Ergebnisse miteinander verglichen. Es wurde dabei eine Toleranz von weniger als 0.5% festgestellt. In einer zweiten Messreihe wurden die 981 Pulse auf 3l hochgerechnet und dieser Wert ausgelesen. Dabei sollte beim Erreichen von 3924 Pulsen 3l Flüssigkeit abgefüllt sein. Bei diesem Test wurde eine Genauigkeit von 2.7% festgestellt.

All diese Versuche wurden vorgängig durchgeführt, bevor die eigentliche Evaluation starten sollte, um zu sehen wie gut alles arbeitet und ob die Toleranzen eingehalten werden können. Leider ist durch ein Missgeschick das vorhandene Durchflussmessgerät zerstört worden, weshalb keine ausführliche Evaluation mit den dazugehörigen Messwerten durchgeführt werden konnte. Somit kann nur gesagt werden, dass die Messgeräte die erforderlichen Toleranzen einhalten, aber diese in Projekt 6 nochmals auf deren Genauigkeit geprüft werden müssen, sobald die neuen Messgeräte angekommen sind.

5.5 Display

Im Folgenden wird das Display geprüft. Da der Mikrocontroller den Programmfluss steuert, wird die einzige Aufgabe des Displays sein, dem Mikrocontroller zu melden, wenn ein Button gedrückt wurde. Es geht folglich darum, die UART-Schnittstelle zu prüfen.

5.5.1 Test

In diesem Kapitel wird die Funktion des Displays verifiziert. Die Funktion wurde getestet, indem in der Nextion Software *Nextion Editor* ein String über die UART-Schnittstelle ausgegeben und vom Oszilloskop abgebildet wurde. Der auszugebende String kann in der Software definiert und per Mikro-SD-Karte in den Programmspeicher des Displays geladen werden. Er wird versendet, sobald auf den "Menu"-Knopf gedrückt wird, welcher in Abbildung 5.14 zu erkennen ist.

5.5.2 Erwartung

Der zu sendende String wurde auf "0xFF 0xB1 0xFF 0xFF 0xFF" festgelegt und soll die ID des "Menu"-Buttons wiederspiegeln, welcher sich auf Seite 0 befindet und Button 1 heißt. Das erste Zeichen (0xFF) gibt zu erkennen, auf welcher Seite wir uns befinden (eigentlich 0, doch es gab Probleme 0x00 zu interpretieren, weswegen entschieden wurde, anstelle von einem 0x00 ein 0xFF zu versenden). Die letzten drei Bytes (0xFF 0xFF 0xFF) geben das Ende der Übertragung zu erkennen. Wird der Button gedrückt, so wird erwartet, dass das Display die Seiten-ID gefolgt von der Button-ID und des Kommunikationsschlusses sendet.



Abbildung 5.14: Nextion Software. Zu beachten: prinrh-Funktion 0xFF 0xB1 0xFF 0xFF 0xFF.

5.5.3 Ergebnis

Die Abbildung 5.14 zeigt, in welcher Folge die Daten an den Bus übergeben werden. Abbildung 5.15 zeigt eine Abbildung des Oszilloskops, welches die Datenübertragung festhält.

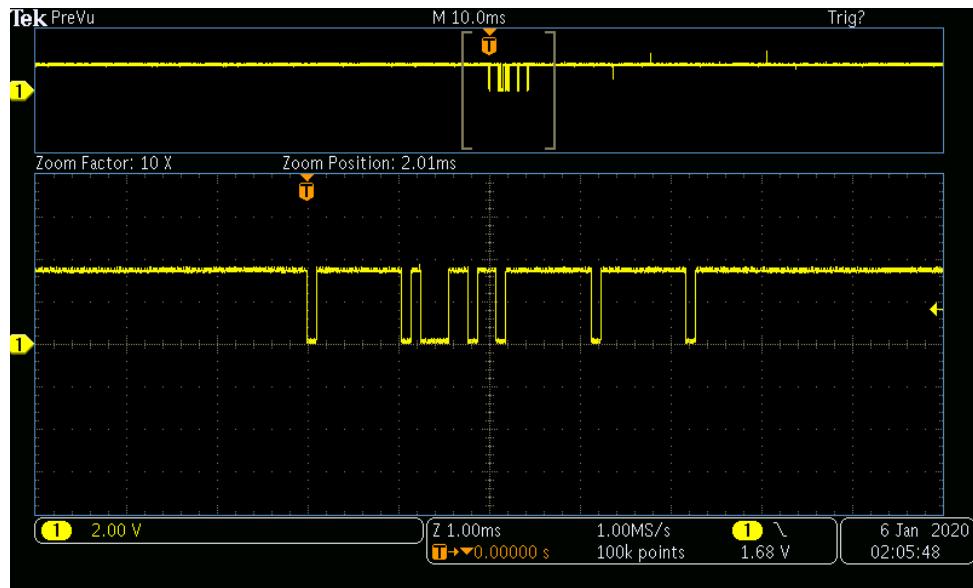


Abbildung 5.15: Messung gesendeter Daten des Displays. Zu beachten: 0xFF 0xB1 0xFF 0xFF 0xFF.

Wie zu erkennen ist, stimmt die Datenfolge, welche vom Display an den Bus weitergegeben wird, die Funktion ist somit bestätigt.

Wichtig ist die Reihenfolge der Bits, wie zu erkennen ist, findet die Übertragung mit dem LSB-first statt, entgegen der SPI-Schnittstelle, welche mit dem MSB-first kommuniziert.

5.6 Mikrocontroller

Der Test des Mikrocontrollers soll eine reine Funktionskontrolle mit Überprüfung der Datenübertragung sein, welche das Zusammenspiel zwischen Display, Mikrocontroller, Pumpen und Motorentreiber testet.

5.6.1 Systemtest

Der Test sieht wie folgt aus:

1. Bei Druck auf den Button des Displays:
 - Button-ID wird vom Display an den Mikrocontroller gesendet.
 - Mikrocontroller verarbeitet die ID.
2. Bei der ID (0xFF 0xB1 0xFF 0xFF 0xFF) löst der Mikrocontroller folgende Testfunktionen aus:
 - Mikrocontroller initialisiert den TMC4671.
 - Mikrocontroller ändert Text und Bild auf Display.
 - Mikrocontroller toggelt eine LED (Simuliert die Ansteuerung eines FET)

5.6.2 Erwartung

Wird der Menu-Button 0xB1 auf der Seite 0xFF gedrückt, so wird erwartet, dass der Mikrocontroller die Funktionen in folgender Reihenfolge abarbeitet.

1. Statusübermittlung
 - UART_0-Schnittstelle muss angesprochen werden.
 - Vor jedem Ausführen einer Funktion teilt der Mikrocontroller dem Computer mit, an welchem Punkt der Software er sich gerade befindet.
2. Initialisierung TMC
 - SPI-Schnittstelle muss angesprochen werden.
 - Motor dreht sich noch nicht.
3. Text und Bild ändern
 - UART_1-Schnittstelle muss angesprochen werden.
 - Gin-Tonic-Text muss mit Lemon Soda ersetzt werden.
 - Bild von Gin-Tonic muss mit anderem Bild ersetzt werden.
4. LED-Toggeln (Simuliert die Ansteuerung eines FET)
 - Die Hardware des Mikrocontrollers muss angesprochen werden
 - LED toggelt bei jeder Beführung des Buttons.

5.6.3 Ergebnis

Folgende Ergebnisse wurden beim Test erzielt:

1. Statusübermittlung

Wie in Abbildung 5.16 zu sehen ist, funktioniert die Übertragung einwandfrei. Der Mikrocontroller läuft durch die Software und zeigt, an welcher Stelle er ist. Die Funktion ist somit bestätigt.

```

Beginn Tests: \n
\n\n
\n\n
Initialisiere TMC: \n
\n\n
Setze Text Display: \n
t0.txt="Lemon Soda"\n
\n\n
Setze Bild: \n
pic 250,80,7\n
\n\n
Toggle LED: \n
\n\n

```

Abbildung 5.16: Ausgabe der abzuhandelnden Schritte, wie sie gemäss Button 0xB1 der Seite 0xFF programmiert wurden, an die USB-Schnittstelle.

2. Initialisierung TMC

In Abbildung 5.17 ist zu erkennen, dass die Übertragung korrekt ausgegeben wird. Da die gesamte Übertragung zu gross ist, wurde das Ende dokumentiert, erkennbar in der Abbildung 5.18. Die Initialisierung konnte so nicht dokumentiert werden aufgrund der Probleme mit dem EVAL-Board. Die Initialisierung gilt deshalb noch als teilverifiziert. Die komplette Implementierung des Chips wird Teil des Projekt 6 sein.

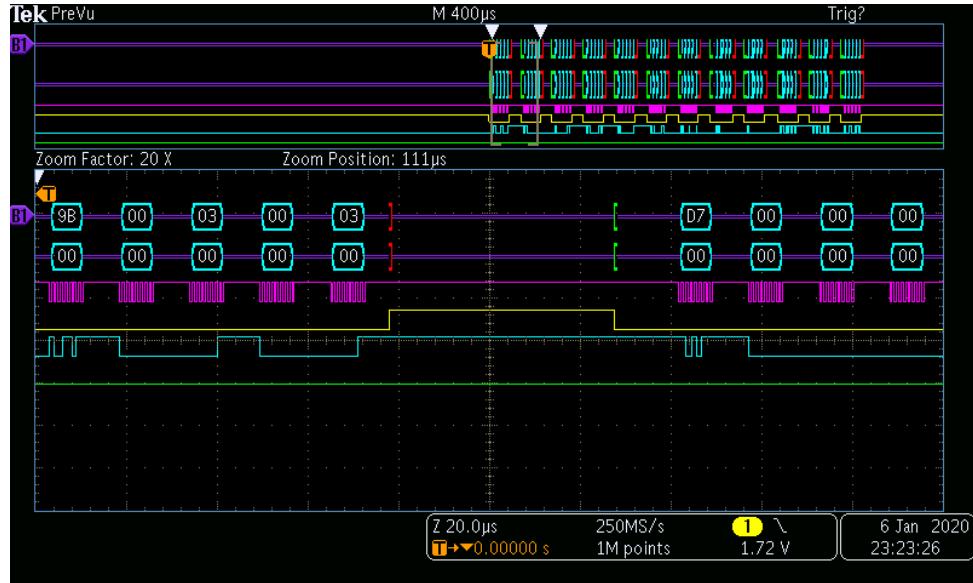


Abbildung 5.17: Beginn der Ausgabe der SPI-Schnittstelle, wie sie gemäss TMC4671_init() programmiert wurde.

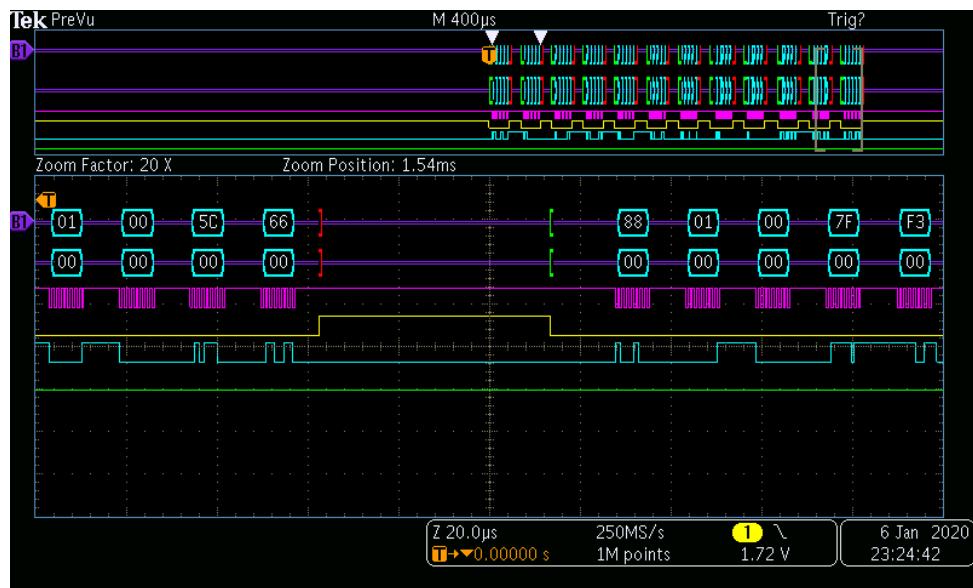


Abbildung 5.18: Ende der Ausgabe der SPI-Schnittstelle, wie sie gemäss TMC4671_init() programmiert wurde.

3. Text und Bild ändern

Abbildung 5.19a zeigt den ersten Bildschirm, wenn es eingeschaltet wird. Nachdem der Button “Menue“ gedrückt wurde, wird der Text von Gin Tonic zu Lemon Soda geändert und das Bild mit dem Drink wird gewechselt. Die Funktion gilt als bestätigt.



(a) Seite 0xFF bevor der Button gedrückt wird.

(b) Seite 0xFF nachdem der Button gedrückt wird.

Abbildung 5.19: Bildschirmanzeigen vor und nach dem Drücken des Buttons.

4. LED toggeln

Wie in Abbildung 5.20a ersichtlich, geht die eingebaute LED des Arduino-Mega-Boards an. Sobald erneut auf den Button gedrückt wird, geht diese wieder aus wie in Abbildung 5.20b erkennbar ist. Die Ansteuerung eines FET während des Programmflusses ist somit verifiziert.

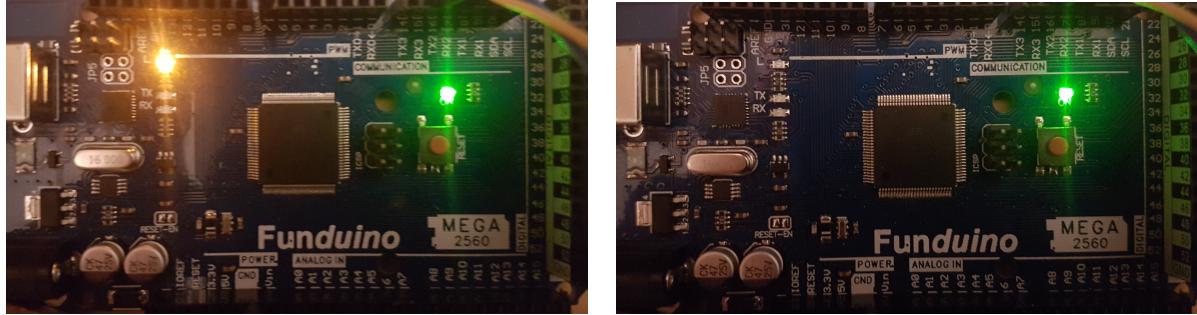


Abbildung 5.20

5.6.4 Programmablauf

Der Test sieht wie folgt aus:

1. Bei Druck auf den Button des Displays
 - Button-ID wird vom Display an den Mikrocontroller gesendet
2. Mikrocontroller verarbeitet die ID und löst bei **0x07 0x01 0xFF 0xFF 0xFF** folgende Funktionen aus:
 - Unterhaltungstext schreiben.
 - Seite wechseln “Getränk fertig”
 - Seite wechseln “Cocktailliste”

5.6.5 Erwartung

Wird der Menu-Button 0x01 auf der Seite 0x07 gedrückt, wird erwartet, dass der Mikrocontroller die Funktionen in folgender Reihenfolge abhandelt.

1. Jeweilige Seite ändern (alle Seiten)
 - UART_1-Schnittstelle muss angesprochen werden.
 - Timer auf Seite “Cocktail wird zubereitet“ betreibt eine Progressbar.
 - Jeweilige Seite wird aufgerufen.

5.6.6 Ergebnis

1. Gewünschte Seiten aufrufen

Die Seiten werden wie im Programmfluss vorgegeben aufgerufen. Auch die Progressbar läuft wie erwartet von 0% nach 100%. Nach Ablauf des Programms sind weitere Eingaben möglich, die Software hängt sich während dem Betrieb also nicht auf. Die Erwartungen sind folglich verifiziert.

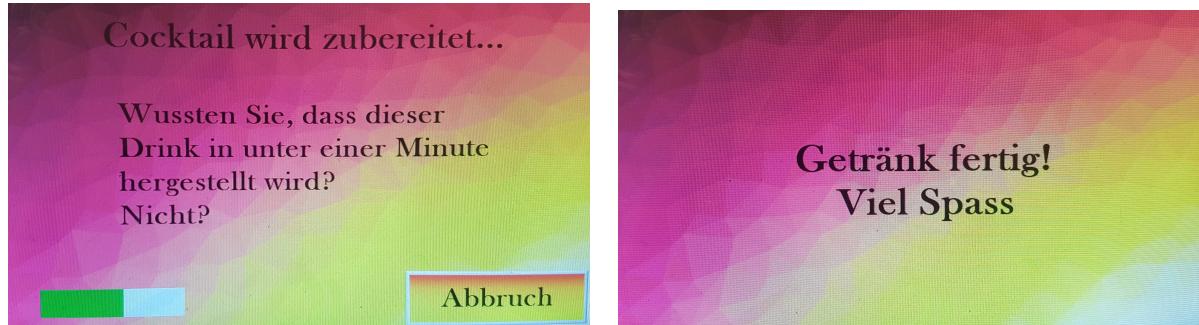


Abbildung 5.21: Anzeigen des Displays



Abbildung 5.22: Angezeigtes Display, um Programm weiter bedienen zu können.

6 Software Mikrocontroller

Die Software steuert die Zubereitung der Drinks und stellt die Schnittstellen bereit, welche nach aussen kommunizieren. Die Software für den Mikrocontroller wurde vollständig in C geschrieben und wird im Folgenden erklärt.

6.1 Programmflussdiagramm main.c

Abbildung 6.2 zeigt den Hauptprogrammfluss vom Start des Programms über die Initialisierung des µC bis in den Mainloop.

Der Aufbau der Software gliedert sich klassisch nach folgendem Prinzip. Die aufgelisteten Punkte werden in Kapitel 6.2 beschrieben.

Modulrumpf
Einlesen der Headerfiles.
Präprozessor-Anweisungen.
Deklarationen.

Mainloop
Initialisierungen.
Mainroutine.

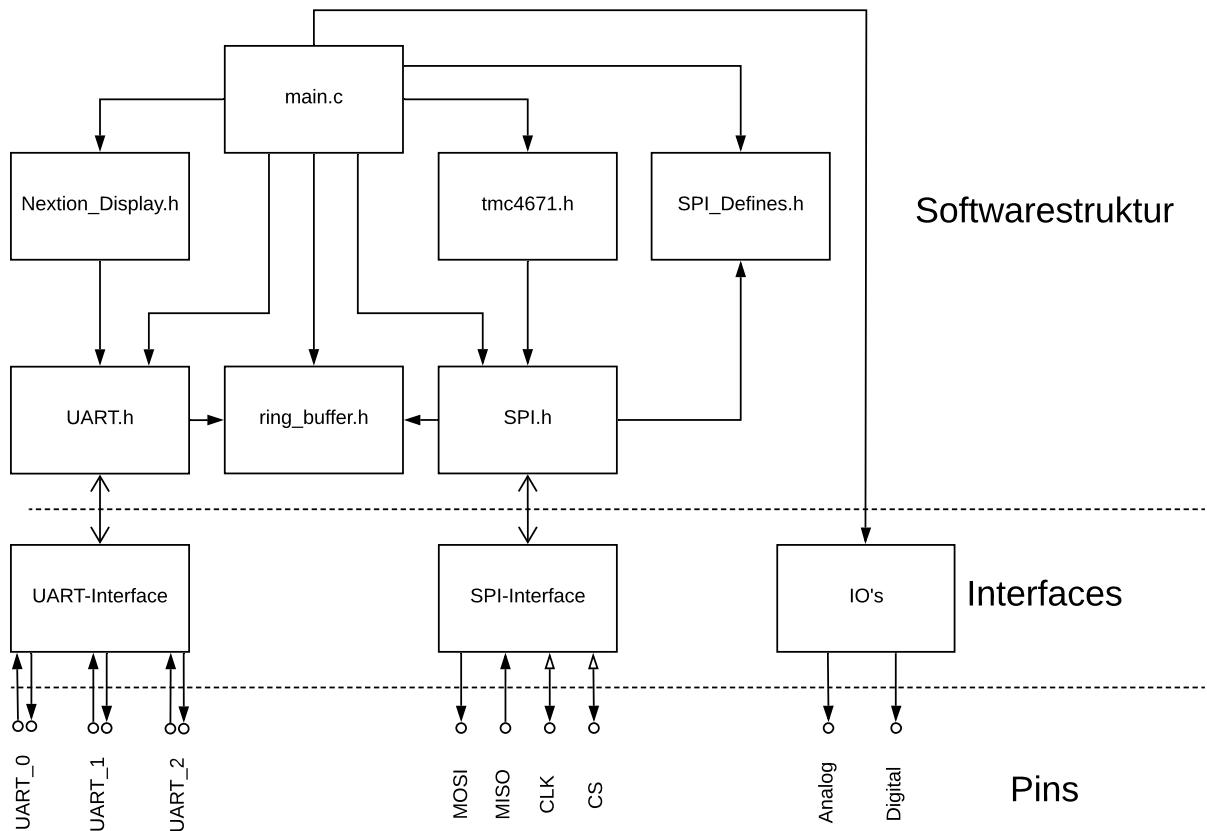


Abbildung 6.1: Softwarestruktur Mikrocontroller.

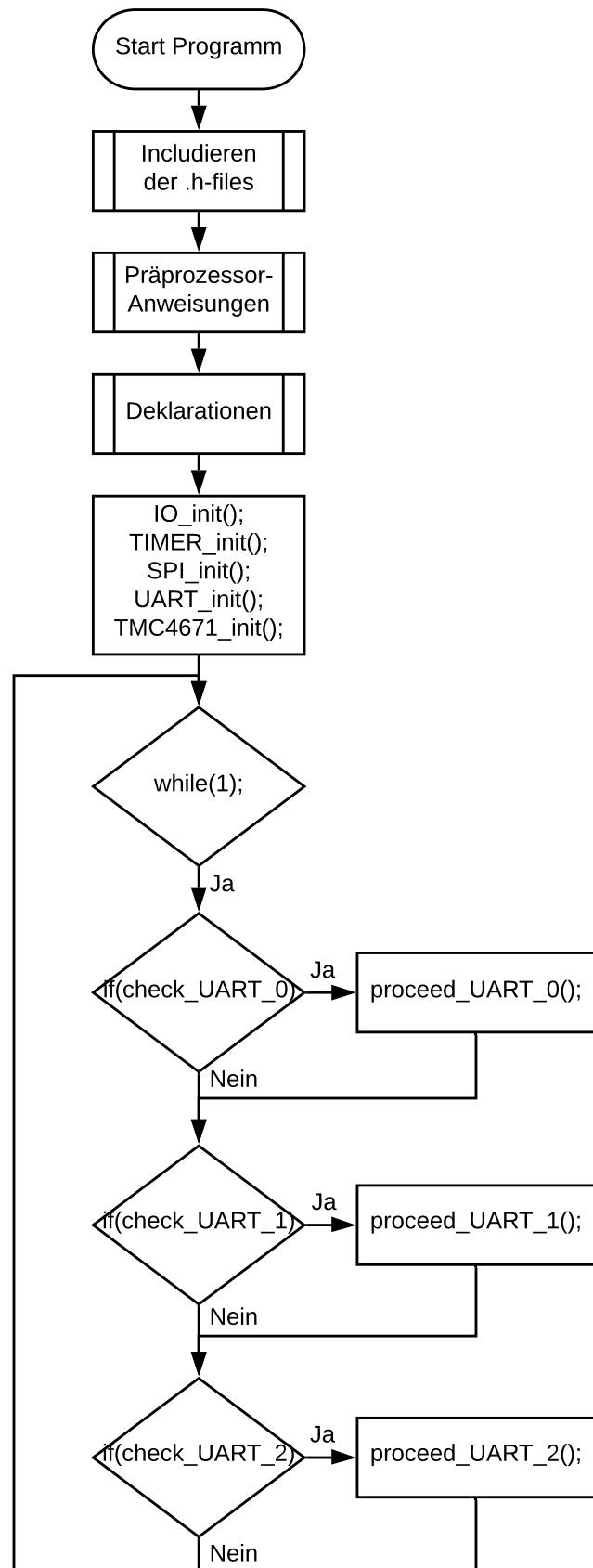


Abbildung 6.2: Programmfluss main.c.

6.2 Dokumentation

Im Foldenden werden die einzelnen Teile der Software im Detail dokumentiert. Sie gliedert sich nach den in Kapitel 6 beschriebenen Aufbau der Software.

6.2.1 Einlesen der Headerfiles

Als erstes kommen die Standardheaderfiles. Sie umfassen ein Deklarations-Set, welches für einen Basisbetrieb ausgelegt ist.

#include <avr/io.h>	Includiert AVR-spezifische IO-Deklarationen für Registernamen etc.
#include <avr/interrupt.h>	Includiert Interrupt Deklarationen.
#include <util/setbaud.h>	Includiert das Headerfile, welches helfende Makros enthält für die Kalkulation der Baudrate. [40]
#include <string.h>	Includiert die Definitionsdatei, in welcher die beiden Gruppen von Funktionen für Zeichenketten deklariert wurden. [41]
Nach den Standardheaders kommen die projektbezogenen Headers. Diese wurden erstellt, um die Funktionen innerhalb der Software lesbarer zu gestalten und einen logischen Softwareaufbau zu erlangen. So können die verschiedenen Teilsysteme voneinander abgekoppelt und separat implementiert werden.	
#include "...ring_buffer.h"	Includiert die Ringbuffer-Header. Diese wird im SPI- und UART-Header verwendet, da sie unter Anderem ein Typedef <i>ring_buffer_t</i> enthält.
#include ".../UART.h" #include ".../SPI.h"	Includiert die Kommunikations-Headers. Eine davon ist der UART-Header, die Andere ist der SPI-Header. In beiden werden zwei Buffer initialisiert, jeweils einen für die zu sendenden Daten und einer für die zu empfangenen Daten.
#include ".../TMC4671.h"	Includiert die Befehls-Bibliothek des TMC4671. Diese enthält einige Funktionen aus der SPI_TMC-library. Sie ermöglicht eine kompakte Befehlseteilung, da je nach Ansteuerung des Motors mehrere Register des Treibers beschrieben werden müssen. Folglich ist es sinnvoll, diese in einer Funktion zu vereinen, was in diesem Header geschieht.
#include ".../Nextion.h"	Includiert die Befehls-Bibliothek des Nextion_Display. Diese enthält einige Funktionen, um das Display anzusteuern. Sie ermöglicht eine bisher einigermassen kompakte Befehlseteilung, da der Wrapper noch nicht fertiggestellt ist. Einige eigens geschriebene Funktionen sind jedoch schon tauglich für die Software.

6.2.2 Präprozessor-Anweisungen

Die Präprozessor-Anweisungen werden noch vor dem eigentlichen Kompilieren umgesetzt. Dabei handelt es sich um Makros (#define) und Headerdateien (#include).

In-/Outputs	Hier werden die Pindefinitionen gemacht, wie sie während des Betriebs vorhanden sind. Dazu gehören die Pins für: SPI (MOSI, MISO, CLK, CS), UART (TX0, RX0, TX1, RX1), Pumpen 12x und Durchflussmessungen 12x.
Pinnamen und Masken	An dieser Stelle werden die verschiedenen Masken erstellt. Dies ermöglicht eine einfache Deklaration von ganzen Registern, wenn es darum geht, die In- und Outputs eines Systems zu definieren oder einzelne Zustände zu ändern.
Makros	Makros werden nur gerade zwei im Präprozessor deklariert. Diese sind dazu da, die benötigten SPI-Slaves auszuwählen. So wird in der einen Funktion der entsprechende Pin auf GND gezogen und mit der zweiten wieder auf 5V gebracht.

6.2.3 Deklarationen

An dieser Stelle werden die für den Programmfluss nötigen Variablen, Funktionen und Buffer vordeklariert. Dabei geht es vor allem um Variablen zur Verarbeitung der Kommunikationsdaten.

Variablen	Hier werden die Variablen deklariert, welche für den Programmfluss benötigt werden. Diese umfassen bisher Variablen für SPI- und UART-Kommunikation sowie bufferspezifische Variablen des Typs <i>ring_buffer_t</i> .
Funktionen	In diesem Abschnitt werden Funktionen vordeklariert, welche im main.c vorkommen. Diese umfassen Funktionen für die Deklaration der IO-Ports, für die Funktion des Buffers, für das Polling ob eine komplette Datenübertragung stattgefunden hat (komplett bedeutet in diesem Fall, dass ein Ende gemäss Kommunikationsprotokoll erreicht wurde), für die Verarbeitung der Daten im Falle einer kompletten Übertragung und für ein Blinklicht, um den Status des Mikrocontrollers zu erkennen.

6.2.4 Initialisierungen

Im Mainloop wird, noch vor der while(1)-Schleife, als erstes die Funktion für die Initialisierung der IO-Ports aufgerufen, dann die Initialisierung der Kommunikationschnittstellen und zuletzt die Initialisierung des Motorentreibers. Diese werden im Folgenden erklärt:

IO-Ports	Initialisiert die In-/Outputs, basierend auf den Definitionen im Standardheader <avr/io.h> und in den Präprozessor-Anweisungen. Die Initialisierung an sich wird in der Funktion void IO_Init(void) aufgerufen.
SPI	Initialisiert die SPI-Schnittstelle. Massgebend ist das SPI Control Register (SPCR). Dieses wurde nach SPI-Mode 3, MSB first und 1MHz konfiguriert und entspricht so den Anforderungen des Motorentreibers, welcher über SPI angesteuert wird. Die Funktion zur Initialisierung der SPI-Schnittstelle heisst SPI_init().

UART	Initialisiert die UART-Schnittstelle. Massgebend sind die Register UCSRnB und UCSRnC bzw. UCSRnB und UCSRnC. Die Variable n im Registername steht für den UART-Port n, da der Atmega2560 mehrere UART-Schnittstellen hat. Da drei UART-Ports benötigt werden, wird die Reihenfolge der Schnittstellen foldendermassen festgelegt: UART-Port 0 = USB, UART-Port 1 = Nextion-Display. Für eine Erweiterung mit einem WLAN-Modul wird UART- Port 2 verwendet. Die Register UBRRnH und UBRRnL definieren die Baudrate der Kommunikation. Mit der Bibliothek <util/setbaud.h> kann der Wert mit UBRRH_VALUE und UBRLL_VALUE angegeben werden. Die beiden Werte werden mit einem Makros berechnet.
TMC4671	Initialisiert den TMC4671. Im Gegensatz zu den vorherigen Initialisierungen geht es bei dieser Initialisierung nicht darum, etwas im Mikrocontroller zu initialisieren, sondern der Motorentreiber wird auf die Treiberschaltung des AKM22h und den AKM22h an sich initialisiert. Dazu werden die entsprechenden Register beschrieben, welche durch Verwendung der Trinamic-Software <i>TMCL-IDE 3.0</i> ermittelt wurden.
Ring Buffer	Für jede Schnittstelle wird sende- und empfangsseitig mit einem Ring Buffer gearbeitet. Diese müssen auch jeweils initialisiert werden, was mittels einer Funktion der Bibliothek Ring_Buffer.h geschieht. Die gemeinte Funktion heisst RB_init(ring_buffer_t);. Der Buffer verhindert einen Datenstau aufgrund zu langsamer Datenverarbeitung im Mikrocontroller. Die Daten werden im Buffer zwischengespeichert und können verarbeitet werden, sobald der Mikrocontroller dazu kommt.
Interrupts	Soweit möglich wurde bei jeder Schnittstelle mit Interrupts gearbeitet. In Kombination mit dem Buffer wird der Hauptprogrammfluss nur wenig gestört und erlaubt trotzdem eine zuverlässige Verarbeitung der einkommenden Daten. Die globalen Interrupts müssen jeweils mit sei(); aktiviert werden.

6.2.5 Mainroutine

In der Mainroutine wird bisher nur geprüft, ob die Buffer Daten enthalten. Sobald bei einkommenden Daten das Ende einer Übertragung signalisiert erkannt wird, werden diese in einer weiteren Funktion ausgewertet.

6.3 libraries

Damit das lauffähige Programm eine Struktur aufweist und der Code übersichtlich bleibt, werden die verschiedenen Aufgabebereiche in Libraries geschrieben. Diese werden zu Beginn des Programmes eingefügt, um die darin enthaltenen Funktionen verwenden zu können.

Zu den Libraries gehören die SPI-, SPI-TMC, UART-, Ring_Buffer- und die TMC4671- Library. Diese fünf libraries werden im Folgenden kurz erklärt.

6.3.1 Ring Buffer

In der Ring-Buffer Library befinden sich Funktionen, mit denen der Ring Buffer beschrieben und ausgelesen werden kann. Außerdem können verschiedene Informationen zum Datenbestand innerhalb des Buffers abgefragt werden.

1. void RB_init(ring_buffer_t *rb);
2. unsigned char RB_free(ring_buffer_t *rb);
3. unsigned char RB_length(ring_buffer_t *rb);
4. unsigned char RB_readByte(ring_buffer_t *rb);
5. unsigned char RB_writeByte(ring_buffer_t *rb, unsigned char data);
6. unsigned char RB_read(ring_buffer_t *rb, unsigned char *data, unsigned char data1);
7. unsigned char RB_write(ring_buffer_t *rb, unsigned char *data, unsigned char data1);

Die Funktionen haben folgende Aufgaben bzw. führen folgende Aufgaben aus:

1. Initialisiert einen neuen Buffer.
2. Gibt die Anzahl freier Plätze im Buffer zurück.
3. Gibt die Anzahl belegter Plätze im Buffer zurück.
4. Liest und löscht ein Byte aus dem Buffer.
5. Schreibt ein Byte in den Buffer.
6. Liest mehrere Bytes aus dem Buffer aus.
7. Schreibt mehrere Bytes in den Buffer rein.

6.3.2 SPI

Wie beschrieben wird die SPI-Library verwendet, um Daten über das Serial Peripheral Interface zu versenden. Dazu wird die SPI-Library verwendet, welche folgende Funktionen beinhaltet:

1. void SPI_init(void);
2. SPI_Transmit_IT(unsigned char * data, unsigned char nbytes);
3. ISR(SPI_STC_vect);

Die Funktionen haben folgende Aufgaben bzw. führen folgende Aufgaben aus:

1. Initialisiert die Hardware sowie Ring-Buffer für die Empfangs- siwue Sendeseite.
2. Füllt den Buffer mit zu übertragenden Daten und schreibt die Daten aus dem Buffer wieder in das Datenregister der SPI-Schnittstelle. Dies geschieht für ein Byte, die noch im Buffer vorhandenen Bytes werden für die Übertragung über die Interrupt-Routine gehandelt, welche ausgelöst wird sobald das erste Byte an den Slave gesendet wurde.
3. Interrupt-Routine: Falls Daten im Schreib-Buffer des SPI sind, sollen diese Byte-weise in das SPI-Datenregister geschrieben werden. Sind keine Daten vorhanden, soll nichts gemacht werden.

6.3.3 UART

Auch die UART-Library wurde schon erklärt. So ist die Schnittstelle dafür da, Daten über die Universelle Asynchrone Receiver-/Transmitter schnittstelle zu senden. Es ist eine gängige Schnittstelle, welche über ein USB-Kabel realisiert werden kann. Sie ermöglicht also eine Verbindung zu einem Computer.

Die Library an sich beinhaltet folgende Funktionen:

1. `UART_init(void);`
2. `UART_Transmit_IT_n();`
3. `ISR(USARTn_UDRE_vect);`
4. `ISR(USARTn_RX_vect);`

Die Funktionen haben folgende Aufgaben bzw. führen folgende Aufgaben aus:

1. Initialisiert die UART-Schnittstellen.
2. Übermittelt Daten über die gewählte UART-Schnittstelle n.
3. Sobald das UART-Interface bereit ist, neue Daten zu senden, wird dieses Interrupt ausgelöst. Solange sich Daten im Sende-Buffer befinden, werden diese nach und nach über die UART-Schnittstelle versendet.
4. Sobald Daten über die UART-Schnittstelle empfangen wurden, wird diese Interruptroutine ausgelöst. Darin werden die empfangenen Daten in den Buffer zwischengespeichert.

6.3.4 TMC4671

In der Library für den TMC4671 sind Funktionen enthalten, mit denen der Treiber über SPI initialisiert und angesteuert werden kann. Die wichtigsten Funktionen darin sind diejenigen, welche gewährleisten, dass der Treiber die Daten in der richtigen Reihenfolge empfängt. Sie sind im Folgenden aufgelistet. So kann einer Funktion eine Adresse mit zugehörigem Wert mitgegeben werden. Innerhalb der Funktion werden die Daten der richtige Reihenfolge nach in einem Array abgelegt und über die Sendefunktion der SPI-Library über die gleichnamige Schnittstelle versendet.

Es befinden sich nebst diesen Sendefunktionen noch mehr Funktionen darin, mit denen die Parameter des TMC4671 eingestellt werden können. Die Funktionen enthalten in der Regel die gewünschte Adresse des Zielparameters und eine der Sendefunktionen, womit der Zielparameter übermittelt und in den Treiber geladen wird. Von diesen Funktionen sind auch einige aufgelistet, welche für die Cocktailmachine von Bedeutung sein können.

Um zu verstehen, wie die wichtigen Funktionen aufgebaut sind, muss geklärt werden, nach welcher Datenfolge der Motorentreiber arbeitet. Im Datenblatt vom Trinamic TMC4671 wurden folgende Informationen gefunden:

- 40-bit Datagrammlänge (1 ReadWrite-Bit, 7 Adress-Bits + 32 Daten-Bits).
- Sofortige SPI-Leseantwort (Registerlesezugriff über ein einziges Datagramm).
- SPI-Clock-Frequenz bis 1MHz (8MHz in zukünftigen Versionen).

In den aufgelisteten Funktionen befindet sich jeweils der Parameter „`unsigned int motor`“. Desse Funktion ist unbekannt, es wird jedoch vermutet, dass diese Einstellung für einen Treiber bestimmt ist, welcher mehrere Motoren ansteuern kann. Um die Funktionalität der Funktionen zu gewährleisten, wurde dieser Parameter jeweils mit „`#define MOTOR0 0`“ deklariert und geladen. Die wichtigen Funktionen sind im folgenden aufgelistet und können als „SPI-Wrapper“

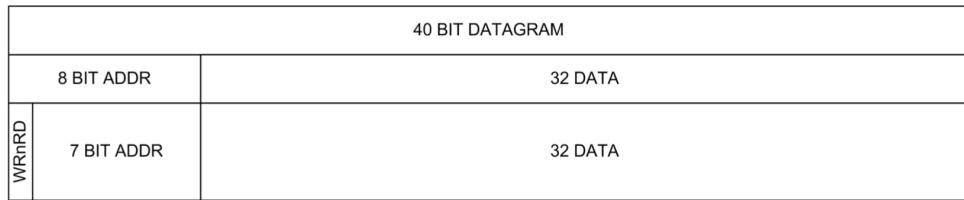


Abbildung 6.3: SPI-Datenstruktur. [16]

betrachtet werden, welcher die Verbindung zwischen der SPI-Library und der TMC4671-Library bildet: [42]

1. void tmc40bit_writeInt(unsigned int motor, unsigned char address, unsigned long value);
2. int tmc40bit.ReadInt(unsigned int motor, unsigned char address);

Im Falle der Cocktailmachine docken wir uns hier mit der Funktion “tmc40bit_(read/write)Int“ der TMC4671-Library und der Funktion “SPI_Transmit_IT“ an die SPI-Library des Mikrocontrollers. Nun können die restlichen Funktionen der TMC4671-Library verwendet werden, ohne im Programmfluss die SPI-Library verwenden zu müssen. Ausnahme: Chip-Select des Treibers auf enable/disable setzen.

1. Schreibt eine 8-Bit Adresse mit einem folgenden 24-Bit Wert.
2. Schreibt eine 8-Bit Adresse und liest einen 24-Bit Wert zurück.

6.3.5 Nextion Display

Die offizielle Library für das Nextion-Display wurde nicht verwendet, da sie in C++ geschrieben ist. Auch eine veröffentlichte Version auf GitHub, welche in C geschrieben worden war, stellte sich als zu kompliziert heraus. Deswegen wurde bisher versucht, eine vereinfachte Library zu schreiben. Sie basiert auf dem gleichen Prinzip wie die SPI-Library: Es wird ein Wrapper geschrieben, welcher sich nach dem Kommunikationsprinzip des Displays richtet. Bei Bedarf kann so eine Funktion geschrieben werden, welche einen Datenaustausch zwischen Mikrocontroller und Display auslöst. Leider ist dies beim Nextion-Display aufwendiger aufgrund unterschiedlich langen Übertragungsarrays. Deshalb wird diese eine Aufgabe für das Projekt sechs sein.

6.4 Lizenzen Bibliotheken

6.4.1 SPI

Die SPI-Library wurde selbst geschrieben. Lediglich die Methodik mit dem RingBuffer wurde aus der UART-Library auf das SPI angewendet. Somit könnte man sagen, dass die Idee von dort stammt. [43]

6.4.2 Uart

Die UART-library enthält keine spezifisch definierte Lizenz. Die Bibliothek wurde zusammen mit der RingBuffer-Library gefunden auf Youtube. In der Videobeschreibung gibt es einen Link, mit dem man die Bibliothek runterladen kann. Der Autor heiss Jr Sf. [43]

6.4.3 Ring-Buffer

Die RingBuffer-library enthält keine spezifisch definierte Lizenz. Die Bibliothek wurde zusammen mit der UART-Library gefunden auf Youtube. In der Videobeschreibung gibt es einen Link, mit dem man die Bibliothek runterladen kann. Der Autor heiss Jr Sf. [43]

6.4.4 TMC4671

Die Library stammt von Trinamic. Folgende Lizenz wurde erwähnt:

MIT License

Copyright (c) 2019 Trinamic Motion Control GmbH & Co. KG

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "SSoftware"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[44]

7 Fazit

Um ein Fazit ziehen zu können, wird zurückgeschaut, was alles erreicht wurde und was nicht. Danach wird dies mit den Pflicht- und Wunschzielen verglichen.

7.1 Was wurde erreicht und was nicht?

Um aufzuzeigen, was alles erreicht wurde, werden nun alle Blöcke einzeln betrachtet und analysiert.

7.1.1 Speisungen

Es sind vier verschiedene Speisungen aufgebaut worden, welche allesamt erfolgreich auf ihre Funktion getestet wurden. Dazu gehört das 48V Netzgerät, welches als fertiges Netzteil eingekauft wurde. Danach folgt die 12V Speisung für die Pumpen, sowie die 5V Speisung für die integrierten Schaltkreise, das Display und die Durchflusssensoren. Auch der 3,3V Linearregler für die Treiber wurde erfolgreich getestet.

7.1.2 Mikrocontroller

Alle Testprogramme sind erfolgreich in C erstellt worden und mittels ATmega2560 Evaluation Board geprüft. Dazu gehört die Kommunikation mit dem Display über die UART-Schnittstelle, das Ein- und Ausschalten der Pumpen, die Messung eines Durchflusses mittels Durchflussmessgerät und die Kommunikation über SPI mit dem Motor.

7.1.3 Motor

Der Motor kann ohne Feedback im Openloop betrieben werden. Das Feedback vom Resolver des Motors konnte nicht richtig dargestellt werden, weshalb ein ABN-Encoder verwendet wurde. Mit dem ABN-Encoder war es dann möglich, den Motor im Geschwindigkeits- und Positionsmodus anzusteuern. Da während dem Testen Komplikationen auftraten, konnte jedoch nicht weitergemacht werden.

7.1.4 Pumpen

Die Pumpen arbeiten zuverlässig mit der 12V-Speisung und dem Mikrocontroller zusammen. Bei Ansteuerung durch den Mikrocontroller werden diese über die Pumpenansteuerung sauber ein- und ausgeschaltet.

7.1.5 Durchflussmessgeräte

Mit Hilfe der Durchflussmessgeräte kann zuverlässig die beförderte Menge an Flüssigkeit bestimmt werden. Diese ist auf unter 3,3% genau. Dies konnte jedoch nur in einem kürzeren Testlauf getestet werden, da das zu testende Durchflussmessgerät durch einen Handhabungsfehler zerstört wurde und kein Ersatz auf die Schnelle aufgetrieben werden konnte. Die Erfüllung dieses Ziels wird daher im Projekt 6 noch einmal sauber dokumentiert.

7.1.6 Display

Das Display kann erfolgreich betrieben werden. Es ist gelungen, eine grafische Oberfläche zu schaffen, welche auch wie gewünscht auf Befehle reagiert. Außerdem kann erfolgreich mit dem Mikrocontroller kommuniziert werden. Es ist möglich Befehle an den Mikrocontroller zu senden und auch solche zu erhalten. Außerdem können Textinhalte vom Mikrocontroller auf das Display übertragen werden. Dies funktioniert auch zuverlässig anders rum.

7.2 Vergleich der gesetzten Ziele mit den erreichten Zielen

In diesem Abschnitt wird aufgezeigt, welche Ziele erreicht wurden und welche Ziele nicht. Außerdem wird erläutert, welche Ziele weshalb nicht erreicht wurden. Dies wird einmal für die Pflichtziele, sowie für die Wunschziele durchgeführt.

7.2.1 Pflichtzielerfüllung

Nummer	Pflichtziele	Anforderungen												
1	Recherche	Die Recherche muss die Beschreibung drei verschiedener Cocktailsmaschinen enthalten. Damit eine Entscheidung für den Aufbau gefällt werden kann, müssen diese verglichen werden.												
2	Konzept	<p>Das Konzept muss sich komplett auf die Recherche abstützen und im Grunde die Fragen beinhalten, welche sich mit den Projektzielen auseinandersetzen. Diese sind:</p> <ul style="list-style-type: none"> • Wie sieht der mechanische Aufbau der Maschine aus? • Welche Pumpen werden für die Flüssigkeitsbeförderung verwendet? • Wie wird die Menge der durchfliessenden Flüssigkeit gemessen? • Welcher Umfang umfasst die Benutzeroberfläche? • Welcher Microkontroller ist weshalb für die Anwendung geeignet? • Wie ist es möglich, die Maschine zu reinigen? • Wie kann erreicht werden, dass die Gläser nicht überlaufen? 												
3	Fördertechnik	<p>Die Ansteuerung der Fördertechnik muss so geschehen, dass der Inhalt beim Fahren mit dem Schlitten nicht überläuft. Ein Brushless DC-Motor oder Steppermotor ist erwünscht. Die Dimensionen des Förderbands soll folgende Kriterien erfüllen:</p> <table> <tbody> <tr> <td>• Länge:</td> <td>$90 \pm 10\text{cm}$ (Unter Annahme 10cm pro Flasche)</td> </tr> <tr> <td>• Geschwindigkeit:</td> <td>min. 6cm/sek</td> </tr> <tr> <td>• Belastbarkeit:</td> <td>9.81N (1kg auf Schlitten)</td> </tr> <tr> <td>• Oberfläche:</td> <td>Rutschfest</td> </tr> <tr> <td>• Führungen:</td> <td>2 Führungsstangen mit Gewindestange um den Schlitten zu bewegen</td> </tr> <tr> <td>• Schlitten:</td> <td>3D-Druck</td> </tr> </tbody> </table>	• Länge:	$90 \pm 10\text{cm}$ (Unter Annahme 10cm pro Flasche)	• Geschwindigkeit:	min. 6cm/sek	• Belastbarkeit:	9.81N (1kg auf Schlitten)	• Oberfläche:	Rutschfest	• Führungen:	2 Führungsstangen mit Gewindestange um den Schlitten zu bewegen	• Schlitten:	3D-Druck
• Länge:	$90 \pm 10\text{cm}$ (Unter Annahme 10cm pro Flasche)													
• Geschwindigkeit:	min. 6cm/sek													
• Belastbarkeit:	9.81N (1kg auf Schlitten)													
• Oberfläche:	Rutschfest													
• Führungen:	2 Führungsstangen mit Gewindestange um den Schlitten zu bewegen													
• Schlitten:	3D-Druck													
4	Pumpen	<p>Für die Flüssigkeitsbeförderung sollen Pumpen verwendet werden. Die Dauer deren Ansteuerung regelt die Menge der durchfliessenden Flüssigkeit auf eine Genauigkeit von 10ml bei einem Inhalt von 3dl. Die Regelung darf demnach eine Toleranz von 3.3% aufweisen.</p> <p>Weiter sollen die Getränke von einer Menge von 3dl in unter einer Minute fertiggestellt werden. Daraus folgt eine Mindestdurchflussrate von 0.6l pro Minute</p>												
5	Microkontroller	Der Microkontroller muss alle Komponenten ansteuern können, damit auf Multiplexer oder Schieberegister verzichtet werden kann. Dazu gehören die Pumpen sowie die Flüssigkeitsmessung. Zudem soll er alle benötigten Schnittstellen (SPI, UART) unterstützen, damit eine Kommunikation mit allen Komponenten stattfinden kann. Dies umfasst den Treiber des DC-Motors (SPI) und das Display (SPI) und zu einem späteren Zeitpunkt den Bluetooth- oder WiFi-Chip (UART).												
6	Display	Das Display soll über SPI angesteuert werden. Der Benutzer soll mittels Touch-Eingabe das Gerät bedienen können und sämtliche Eingaben ermöglichen. Dies umfasst das Auslösen der Getränkezubereitung, den Reinigungsmodus und speichern von Getränke.												
7	Software	Die Software für den Mikrocontroller soll in C geschrieben sein.												

7.2.2 Wunschzielerfüllung

Nummer	Wunschziele	Anforderungen
1	Reinigung	Das System soll einen Selbstreinigungsmodus haben, der jedoch nur unter Aufsicht des Benutzers geschehen kann. Die Aufsicht verhindert unkontrolliertes Reinigen.
2	Durchflussmessung	Die Menge der durchfliessenden Flüssigkeit muss auf 1ml genau sein bei einem Inhalt von 3dl. Dies entspricht einer Toleranz von 0.33%. Weiter soll ein Getränk mit einer Menge von 3dl in unter einer halben Minute fertiggestellt sein. Dies entspricht unter Berücksichtigung der Bewegung zwischen den Getränken einer Mindestdurchflussrate von 1.2l pro Minute.
4	Messstation	Das System soll den Füllstand im Glas erkennen, um ein Überlaufen zu verhindern.
5	Software	Die Software soll nach dem MVC-Prinzip funktionieren.

Fördertechnik:

Für das Förderband wurde ein BLDC-Motor (Brushless-DC-Motor) ausgewählt. Auch der Aufbau des Förderbandes wurde festgelegt und in Kapitel 3.5 spezifiziert. Allerdings wurde entschieden, dieses erst im Projekt 6 gemäss den Anforderungen aufzubauen. Somit wird auch der Motor in Kombination mit dem Förderband erst im Projekt 6 getestet, wenn dieser komplett implementiert ist und ein Feedback erhalten werden kann. Ein sanftes Anfahren des Motors ist jedoch schon im Projekt 5 erreicht worden. Dieses sanfte Anfahren wird durch einen PI-Controller ermöglicht.

Display:

Das gesetzte Ziel für das Display wurde insofern nicht erreicht, dass die Ansteuerung nicht über SPI realisiert wurde, sondern über UART gemäss Kapitel 5.5. Dies wurde jedoch mit dem Fachdozenten besprochen. Alle anderen Punkte wurden erreicht.

Reinigung:

Ein Reinigungsmodus wurde gemäss Kapitel 4.6 erarbeitet. Da jedoch die Cocktailmachine erst in Projekt 6 aufgebaut wird, wurde auch kein fertiger Reinigungsmodus implementiert.

Durchflussmessung:

Mit dem getesteten Durchflussmessgerät konnte eine Skalierungsgenauigkeit von 2.7% und eine Wiederholungsgenauigkeit von 0.5% in einem Prüfzyklus festgestellt werden. Allerdings konnte nicht genauer getestet werden aus den in Kapitel 5.4 genannten Gründen. Somit konnte dieses Wunschziel nicht erreicht werden.

Messstation:

Es wurde keine Füllstandserkennung implementiert. Es wurde jedoch durch eine Abfrage dem Benutzer überlassen, das jeweils richtige Glas zu platzieren. Durch die Durchflussmessgeräte wird sichergestellt, dass nur die bestätigte Flüssigkeitsmenge abgefüllt wird.

Software:

Die Software wurde nicht nach dem MVC-Prinzip erstellt.

8 Schlusswort

Während des ganzen Projekt 5 wurde intensiv an der Vorbereitung der Bachelor-Thesis (Projekt 6) gearbeitet. Dabei sollte sichergestellt werden, dass alle grundlegenden Funktionen gewährleistet sind. Das wichtigste war es, eine ausführliche Recherche zu starten, damit ein klarer roter Faden erstellt werden konnte. Dies ermöglichte es uns auch ein sauberes Konzept zu erstellen. Aus dem Konzept konnte im Anschluss erfolgreich auf die einzelnen Teilsysteme eingegangen werden. Dazu gehörte es, diese in einem Testaufbau aufzubauen und zu Validieren. Am Ende des Projekts 5 ist es jetzt möglich die einzelnen Teilsysteme fertigzustellen und zusammen zu führen.

Im Projekt 6 soll nun aus den erfolgten Ergebnissen des Projekt 5 eine funktionierende Cockpitmaschine aufgebaut werden. Somit werden die in Projekt 5 validierten Teilsysteme zusammengeführt und als Gesamtsystem getestet. Ein grosser Teil wird dabei die Entwicklung der Software einnehmen.

9 Ehrlichkeitserklärung

Mit der Unterschrift bestätigt der Unterzeichnende Projektleiter, dass die vorliegende Projektdokumentation selbstständig im Team und ohne Verwendung anderer, als der angegebenen Hilfsmittel verfasst wurde, sämtliche verwendeten Quellen erwähnt und die gängigen Zitierregeln eingehalten wurden. Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden.

Unterschrift:

Ort, Datum:

Literatur

- [1] S. Koths, *Spirits - die mobile Cocktailmaschine - das Ding des Jahres*. Adresse: <https://www.the-cocktailmachine.com/> (besucht am 28. Sep. 2019).
- [2] H. BG Innovation, *Die Cocktailmaschine – Die Cocktailmaschine – in 3 Sekunden bis zum perfekten Cocktail*. Adresse: <http://www.der-cocktailmixer.de/> (besucht am 28. Sep. 2019).
- [3] G. igus, *Automatisiertes Cocktail-Mixgerät*. Adresse: <https://www.igus.ch/info/linear-guides-cocktail-mixing-device?L=de> (besucht am 28. Sep. 2019).
- [4] A. Cocktail, (1) *COCKTAIL AVENUE / Die Cocktail-Maschine für Ihre private Veranstaltung - YouTube*, Jan. 2017. Adresse: <https://www.youtube.com/watch?v=FxKb5PHiULA> (besucht am 28. Sep. 2019).
- [5] myRocktail.de, *Home - myRocktail ® - der Cocktail-Roboter für Ihren Event*. Adresse: <https://www.myrocktail.de/index.php> (besucht am 18. Okt. 2019).
- [6] C. A. Würfel, *myRocktail in Aktion*, Youtube, März 2017. Adresse: <https://www.youtube.com/watch?v=qXK0wAEwK-U> (besucht am 12. Jan. 2020).
- [7] u. imajey Consulting Engineers Pvt Ltd, *Brushless DC Motor, how it is work?* Adresse: <https://www.learnengineering.org/brushless-dc-motor.html> (besucht am 24. Okt. 2019).
- [8] *Bürstenloser Gleichstrommotor*, de, Page Version ID: 191719702, Aug. 2019. Adresse: https://de.wikipedia.org/w/index.php?title=B%C3%BCrstenloser_Gleichstrommotor&oldid=191719702 (besucht am 26. Okt. 2019).
- [9] *The Difference Between BLDC and Synchronous AC Motors*. Adresse: <https://www.motioncontroltips.com/faq-whats-the-difference-between-bldc-and-synchronous-ac-motors/> (besucht am 26. Okt. 2019).
- [10] *Choosing Between Brush and Brushless DC Motors*, en-US, Juni 2018. Adresse: <https://www.alliedmotion.com/choosing-between-brush-and-brushless-dc-motors/> (besucht am 26. Okt. 2019).
- [11] D. Bruner, *Bürstenloser Motor Aufbau, Polzahl, Nutenzahl, Durchmesser, Länge*, Mai 2016. Adresse: https://www.technik-consulting.eu/Weiterbildung/B%C3%BCrstenloser_Motor.html (besucht am 24. Okt. 2019).
- [12] H. Hembach, „Systematischer Vergleich von BLDC-Motorkonzepten mit Anwendung auf nass laufende Wasserpumpen kleiner Leistung“, Dissertation, Universität der Bundeswehr München, Fakultät für Elektrotechnik und Informationstechnik, Neubiberg, 2007. Adresse: <https://d-nb.info/1062916581/34>.
- [13] Marc, *BLDC mit einem Hardware PWM - Mikrocontroller.net*, Aug. 2015. Adresse: <https://www.mikrocontroller.net/topic/374734> (besucht am 10. Jan. 2020).
- [14] Lehane, *Wie sehen die Kommutierungswellen für einen bürstenlosen Motor aus?*, Nov. 2012. Adresse: <https://de.switch-case.com/50776487> (besucht am 15. Jan. 2020).
- [15] *Wie funktioniert ein Resolver - Technische Animation*. Adresse: <https://www.youtube.com/watch?v=7PKJ52b1Qvs> (besucht am 26. Okt. 2019).
- [16] Trinamic, *Datasheet Trinamic TMC4671*, 2018. Adresse: http://www.farnell.com/datasheets/2644677.pdf?_ga=2.105217152.2018910714.1572115367-1375218258.1566549587.
- [17] J. Prof. Dr.-Ing. Raisch, *Stromregelung eines Gleichstrommotors*. Adresse: http://www.control.tu-berlin.de/images/9/9a/PR%5C_Grundlagen%5C_1a.pdf.
- [18] U. von Herr Schleuniger, *FOC-Motorenansteuerung mit TMC4671*, Sep. 2019.
- [19] W. Acky69, *Schlauchpumpe*, de, Page Version ID: 183006981, Nov. 2018. Adresse: <https://de.wikipedia.org/w/index.php?title=Schlauchpumpe&oldid=183006981> (besucht am 18. Okt. 2019).

- [20] W. creative commons, *Techniker Schule Butzbach/Verfahrenstechnik/Fördertechnik /Schlauchpumpe – Projektwiki - ein Wiki mit Schülern für Schüler*. Jan. 2019. Adresse: https://projektwiki.zum.de/wiki/Techniker_Schule_Butzbach/Verfahrenstechnik/F%C3%B6rdertechnik/Schlauchpumpe (besucht am 18. Okt. 2019).
- [21] W. Jens über die Felder, *Membranpumpe*, de, Page Version ID: 191608487, Aug. 2019. Adresse: <https://de.wikipedia.org/w/index.php?title=Membranpumpe&oldid=191608487> (besucht am 18. Okt. 2019).
- [22] A. Store, *US \$6.37 15% OFF/AIYIMA 12 V DC mini wasserpumpe Trinkwasser brunnen elektrische pumpen Korrosion widerstand für tee maschine Wasserfilter-in Pumpen aus Heimwerkerbedarf bei AliExpress - 11.11_Doppel-11Tag der Singles*. Adresse: <https://de.aliexpress.com/item/32836659386.html?spm=a2g0s.9042311.0.0.27424c4d9xQTPG> (besucht am 18. Okt. 2019).
- [23] L. Dongguan Honlite Industrial Co., *800 ml/min, 2 rollen, Honlite 24Vd Schlauchpumpe mit Austauschbare Pumpe Kopf und PharMed BPT Schlauch Rohr-in Pumpen aus Heimwerkerbedarf bei AliExpress - 11.11_Doppel-11Tag der Singles*. Adresse: https://de.aliexpress.com/item/32502501883.html?spm=a2g0o.productlist.0.0.35d75512hisw0G&algo_pvid=d4544cf4-f814-4c59-ab06-6aaee917255a&algo_expid=d4544cf4-f814-4c59-ab06-6aaee917255a-1&btsid=ccf08b6c-bc8d-4f15-8c23-4498db800467&ws_ab_test=searchweb0_0,searchweb201602_5,searchweb201603_52 (besucht am 18. Okt. 2019).
- [24] W. Alfaomega, *Durchflussmesser*, de, Page Version ID: 192493176, Sep. 2019. Adresse: <https://de.wikipedia.org/w/index.php?title=Durchflussmesser&oldid=192493176> (besucht am 18. Okt. 2019).
- [25] F. +. T. Store, *US \$1.97 28% OFF/C18 Neue Heiße 1pc Wasser Kaffee Flow Sensor Schalter Meter Durchflussmesser Zähler 0,3 6L/min-in Durchfluss-Sensoren aus Werkzeug bei AliExpress - 11.11_Doppel-11Tag der Singles*. Adresse: https://de.aliexpress.com/item/32905517966.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp= (besucht am 21. Okt. 2019).
- [26] M. C. S. Sigmatec, *Servomotoren Serie AKM*, März 2018. Adresse: https://www.sigmatek-automation.com/fileadmin/user_upload/downloads/AKM-Technische-Daten-pd.pdf (besucht am 11. Jan. 2020).
- [27] Trinamic, *TMC6200_Datasheet*, Feb. 2013. Adresse: <https://www.elinfor.com/pdf/TRINAMIC/TMC6200-TRINAMIC.pdf> (besucht am 26. Nov. 2019).
- [28] O. Semiconductor, *Operational Amplifiers, Dual Power, 1.0 A Output Current*, Jan. 2020. Adresse: <https://www.onsemi.com/pub/Collateral/TCA0372-D.PDF> (besucht am 11. Jan. 2020).
- [29] —, *Low Voltage, Rail-to-Rail Operational Amplifiers*, Juli 2018. Adresse: <https://www.onsemi.com/pub/Collateral/MC33201-D.PDF> (besucht am 11. Jan. 2020).
- [30] Atmel, *8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash*, Feb. 2014. Adresse: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf (besucht am 11. Jan. 2020).
- [31] N. Semiconductors, „74HC4050 Hex non-inverting HIGH-to-LOW level shifter“, en, Jg. 2016, S. 14, Feb. 2016. Adresse: <https://www.nxp.com/docs/en/data-sheet/74HC4050.pdf> (besucht am 11. Jan. 2020).
- [32] E. Systems, *ESP32 Series*, Jan. 2020. Adresse: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (besucht am 11. Jan. 2020).
- [33] Patrick, *NX8048T070*, en-US. Adresse: <https://nextion.tech/datasheets/nx8048t070/> (besucht am 11. Jan. 2020).

- [34] Aliexpress, *US \$1.94 35% OFF/DC12V 13,8 V 15V 18V 24V 27V 28V 30V 32V 36V 42V 48V 60V 360W 600W 1000W Schaltnetzteil Quelle Transformator AC DC SMPS-in Schaltnetzteil aus Heimwerkerbedarf bei AliExpress*. Adresse: http://de.aliexpress.com/item/32905696401.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp= (besucht am 11. Jan. 2020).
- [35] Mouser, *MP24943DN-LF Monolithic Power Systems (MPS) / Mouser*, de-ch. Adresse: <https://www.mouser.ch/ProductDetail/946-MP24943DNLF> (besucht am 11. Jan. 2020).
- [36] „Very low drop voltage regulator with inhibit function“, en, S. 54,
- [37] M. Mienkina, P. Pekarek und F. Dobes, „56F80x Resolver Driver and Hardware Interface“, en, S. 28, Adresse: <https://www.nxp.com/docs/en/application-note/AN1942.pdf>.
- [38] y. User: (Gast), *Berechnung eines Integrators - Mikrocontroller.net*, Sep. 2009. Adresse: <https://www.mikrocontroller.net/topic/151574> (besucht am 18. Jan. 2020).
- [39] Made-In-China, *Nextion Nx8048t070 7.0 Bildschirm-Baugruppe des Zoll-HMI intelligente intelligente Usart Uart Seriender noten-TFT LCD für Himbeere-PU Arduino foto auf de.Made-in-China.com*. Adresse: https://de.made-in-china.com/co_tekegift/image_Nextion-Nx8048t070-7-0-Inch-HMI-Intelligent-Smart-Usart-Uart-Serial-Touch-TFT-LCD-Screen-Module-for-Raspberry-Pi-Arduino_enieginywwNstdqvnzAum.html (besucht am 11. Jan. 2020).
- [40] P. Klingelbiel, *C Standard-Bibliothek: string.h*, Nov. 1999. Adresse: <http://www2.hs-fulda.de/~klingebiel/c-stdlib/string.htm> (besucht am 18. Dez. 2019).
- [41] doxygen, *avr-libc: <util/setbaud.h>: Helper macros for baud rate calculations*, Feb. 2016. Adresse: https://www.nongnu.org/avr-libc/user-manual/group__util__setbaud.html (besucht am 18. Dez. 2019).
- [42] *TMC-EvalSystem/TMC4671_eval.c at master · trinamic/TMC-EvalSystem*. Adresse: https://github.com/trinamic/TMC-EvalSystem/blob/master/boards/TMC4671_eval.c (besucht am 27. Okt. 2019).
- [43] J. S. User: *ring buffer uart - part1*. Adresse: <https://www.youtube.com/watch?v=zaHU1tzAZE0> (besucht am 14. Jan. 2020).
- [44] *trinamic/TMC-EvalSystem*, en. Adresse: <https://github.com/trinamic/TMC-EvalSystem> (besucht am 18. Jan. 2020).