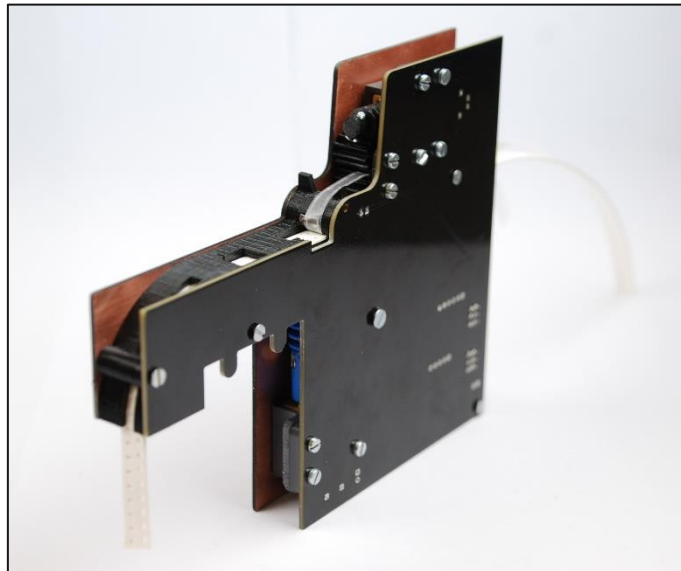


SMD Pick and Place Roboter Automatischer Feeder



Autor

Simon Huber
simon.huber@stud.hslu.ch

Dozent

Prof. Erich Styger
erich.styger@hslu.ch

Industriepartner

Institut Elektrotechnik und Informationstechnologie

Experte

Christian Jost
christian.jost@hslu.ch

Klassifikation | Einsicht

Hochschule Luzern – Technik & Architektur

Horw, 06. Juni 2018

Redlichkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche verwendeten Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Horw, 06. Juni 2018

Simon Huber

Abstract

This document deals with the results and solution finding process of the industrial project “SMD Pick and Place Robot”. This project was carried out as part of the PAIND+E1 module, at the Lucerne University of Applied Sciences & Arts with the industrial partner being the Institute for Electrical Engineering and Information Technology. With the aid of a self-made pick-and-place machine SMT components can be placed on small series of PCBs. This machine is based on OpenPnP and will be used in the electronics workshop at the university, as well as for demonstrations at conventions.

The aim of this project is to design and build an automated feeder for the PnP Machine. Multiple functionalities for the feeder have been defined, such as component transport from strips and rolls, cover tape peeling and communication with OpenPnP. The defined functionalities were split into mechanical, electrical and software tasks. Research was conducted on how to solve each task and the solutions combined to form a concept for the feeder. The concept included a feeder for the transportation of components, a master device for the communication between OpenPnP and the feeders, as well as a feederbank for the mounting of component reels.

This concept was used to build prototypes. For the feeder mechanical parts were machined from PCB or created using 3D printing. PCBs were designed for the hardware components and software was written to actuate the feeder in the desired manner. In multiple iterations changes were made to the design until a feeder fulfilling all functions was created. The feeder moves the tape with a sprocket, and peels off the cover tape with two meshed gears.

The feederbank was created as a simple mechanical holder for the rolls. The master device communicates over USB with the OpenPnP system, and relays commands via an UART Daisy-Chain to the feeders. The master software was not created as part of the project. Functional tests were run on all systems to confirm the functionality of the feeder and the master device.

The results of the prototype designing process as well as the results from the tests are evaluated and presented. The designs are shown and information given so that all components can be reproduced and more feeders built. Furthermore, optimization inputs are given so that the design may be improved.

Abbildungsverzeichnis

Abbildung 1 Gesamt-Systemübersicht	4
Abbildung 2 Übersicht Feederfunktionen	5
Abbildung 3 Systemübersicht Master	5
Abbildung 4 Ablaufdiagramm.....	6
Abbildung 5 Dimensionen eines SMD Streifens (Vishay, 2017)	8
Abbildung 6 N20 DC-Getriebemotor (TinySine, 2018)	9
Abbildung 7 Optischer Inkrementalgeber (Tycho, 2008)	11
Abbildung 8 Kommunikationsverbindungen.....	15
Abbildung 9 Rollen Dimensionen (KEMET, 2016).....	17
Abbildung 10 Darstellung Gesamtkonzept.....	18
Abbildung 11 PnP Maschine mit montierten Feeder und Master	19
Abbildung 12 Feeder für die Streifenbreiten (von hinten nach vorn) 24, 16, 12, 8 mm.....	19
Abbildung 13 offener Feeder mit eingezeichneten Teilfunktionen	20
Abbildung 14 Master	22
Abbildung 15 Feederbank mit zwei 8mm Streifenrollen	23
Abbildung 16 Verzahnung Sprocket mit Streifen	25
Abbildung 17 Darstellung Streifenloch.....	25
Abbildung 18 Sprocket Prototypen aus Acrylglas (oben) und 3D Druck (unten)	26
Abbildung 19 Programmfenster Tape Sprocket Creator (Gipson, 2011)	26
Abbildung 20 Sprocket Prototypen aus 1mm PCB Material	27
Abbildung 21 Verwendetes Design des Sprockets	27
Abbildung 22 3D Darstellung der Führung	28
Abbildung 23 Darstellung Streifen-Auflageflächen	29
Abbildung 24 Dimensionen eines SMD Streifens (Vishay, 2017)	29
Abbildung 25 Prototypen von Führungen aus 3D Druck.....	30
Abbildung 26 Prototypen der Wände aus PCB (v.l.n.r. Version 2, Version 1).....	31
Abbildung 27 Technische Skizzen Wall_B (links), Wall_F (rechts)	32
Abbildung 28 Darstellung des Sprocket-Getriebes	33
Abbildung 29 Hälften eines Schneckenrades	34
Abbildung 30 Schneckenrad Prototypen (v. l. Barrel-Cam, kurze Version, endgültige Version)	35
Abbildung 31 Darstellung des Peelers.....	35
Abbildung 32 Peeler-Mikroswitch in Führung.....	36
Abbildung 33 Motorhalter (v.l. Version 1, Version 2)	37
Abbildung 34 Montierter Sprocket-Antriebsmotor und Getriebe	37
Abbildung 35 Master Gehäuse	38
Abbildung 37 CAD der Feederbank	38
Abbildung 37 Aufliegen der Rolle im Feederbank.....	38
Abbildung 38 Funktionsplan Feeder.....	39
Abbildung 39 Elektroschema des Motorentreibers	40
Abbildung 41 a) Lichtschranke EE-SX1350 (Omron, 2017).....	41
Abbildung 41 b) Schaltungsbeispiel Lichtschranke (Omron, 2017).....	41
Abbildung 42 Elektroschema der Lichtschranke	42
Abbildung 43 Elektroschema der Switches und des LEDs	43
Abbildung 44 Verbindungstecker-Schema und Feeder Verbindungen.....	44
Abbildung 45 Elektroschema des Mikrokontrollers	45
Abbildung 46 Elektroschema zum Debug Interface	46
Abbildung 47 Lichtschranken mit PCB.....	47

Abbildung 48 Version 2 und 3 des Feeder-PCBs	48
Abbildung 49 Layout V5 mit Felddescription.....	49
Abbildung 50 Übersicht Layout Master-PCB	49
Abbildung 51 Bestücktes Master-PCB mit tinyK22	50
Abbildung 52 Softwareübersicht.....	51
Abbildung 53 Ablauf Schritterkennung	52
Abbildung 54 Ablauf Tasterauslesung	53
Abbildung 55 Zustandsautomat	54
Abbildung 56 Phase/Enable Mode der H-Brücke (Texas Instruments, 2016).....	56
Abbildung 57 Funktions-Header von MOT_Speed.....	56
Abbildung 58 Ablaufdiagramm Befehlsanalyse.....	58
Abbildung 59 Prototyp V0	59
Abbildung 60 Prototyp V1	60
Abbildung 61 Bestückter Prototyp V2.1	61
Abbildung 62 Schema Versuch 1	62
Abbildung 63 Signal der Lichtschranke mit 0,2mm Sprocket.....	62
Abbildung 64 Schema Versuch 2: Zusätzlicher Transistor.....	63
Abbildung 65 Schrittsignal mit Zusatz-Transistor.....	64
Abbildung 66 Signalverhalten Komparator an Fototransistor	64
Abbildung 67 Schema Versuch 3: Transistor & Komparator.....	65
Abbildung 68 Vergleich Transistor und Komparator	65
Abbildung 69 8 mm Schrittmessung	66
Abbildung 70 Peeler mit Hebel-Endschalter	67
Abbildung 71 Fehlfunktion Peeler	67
Abbildung 72 Hebel- und Knopftaster Prototypen Mikroswitch.....	68
Abbildung 73 Detektorschalter	68
Abbildung 74 Schema Strommessung.....	72
Abbildung 75 Zeitliche Stromaufnahme (leichte Last)	72
Abbildung 76 Zeitliche Stromaufnahme (grosse Last)	73
Abbildung 77 Risikomatrix.....	76

Tabellenverzeichnis

Tabelle 1 Kosteneinschätzung pro Feeder	3
Tabelle 2 Morphologischer Kasten.....	7
Tabelle 3 Evaluation Antrieb Transport.....	10
Tabelle 4 Evaluation Schritterkennung	11
Tabelle 5 Evaluation Abdeckungsentfernung.....	12
Tabelle 6 Vergleich Mikrokontroller.....	14
Tabelle 7 Eigenschaften Feeder	20
Tabelle 8 Eigenschaften Master	22
Tabelle 9 Eigenschaften Feederbank.....	23
Tabelle 10 Kennwerte DC-Getriebemotor.....	34
Tabelle 11 Messwerte Stromaufnahme	39
Tabelle 12 Pinbelegung Mikrokontroller.....	45
Tabelle 13 Zustände der Taster	52
Tabelle 14 Masterbefehle	57
Tabelle 15 Feeder Meldungen.....	57
Tabelle 16 Situations-Strommessung.....	71
Tabelle 17 Strommessung aller Feeder	71
Tabelle 18 Vergleich Feederbreite Ziel/Ist	78
Tabelle 19 Kosten für 50 Feeder	78
Tabelle 20 Status der definierten Ziele	79

Glossar

PnP	<i>Pick and Place</i> . Das Aufnehmen und Platzieren von Objekten, in diesem Zusammenhang mit einem Bestückungsautomaten für Elektronik-Komponenten
SMT	<i>Surface Mount Technology</i>
SMD	<i>Surface Mount Device</i>
Feeder	<i>Feeder</i> versorgen den Bestückungsautomaten mit Bauteilen.
Peeler	Der <i>Peeler</i> ist der Teil des Feeders, welche die Abdeckungsfolie vom Streifen trennt.
OpenPnP	Dies ist ein <i>Open Source SMT Pick and Place System</i> , welches Software anbietet und die Konfiguration mit Hardware ermöglicht, um einen Bestückungsautomaten in Betrieb zu nehmen.
Feederbank	Eine Halterung für Rollen von Streifen. Darauf können mehrere Rollen nebeneinander platziert werden.
Sprocket	<i>Kettenrad, Zahnkranz</i> . Ein Maschinenelement, welches zur Förderung einer Kette oder anderem perforiertem Material verwendet wird.
Drag	Englisch für <i>Ziehen/Schleppen</i> . Bei Feedern dieser Art wird der Streifentransport mit einem Zugmechanismus realisiert.
PCB	<i>Printed Circuit Board</i> ist die englische Bezeichnung für Leiterplatte.
UART	<i>Universal Asynchronous Receiver-Transmitter</i> ist eine serielle Schnittstelle.
USB	<i>Universal Serial Bus</i> ist ein serielles Bussystem.
PWM	<i>Pulsweitenmodulation</i> ist eine Modulationsart bei der die elektrische Spannung zwischen zwei Werten wechselt.
I2C	<i>Inter-Integrated Circuit</i> ist ein serieller Datenbus für die Kommunikation zwischen Komponenten.
SWD	<i>single wire device</i> ist ein Kommunikations-Bus System.
FSM	<i>finite state machine</i> ist die englische Bezeichnung für einen Zustandsautomaten.
RTOS	<i>real-time operating system</i>

Inhaltsverzeichnis

Abbildungsverzeichnis.....	iii
Tabellenverzeichnis	v
Glossar	vi
1 Einleitung.....	1
1.1 Ausgangslage	1
1.2 Aufgabenstellung PAIND	1
1.3 Anforderungen und Ziele	2
2 Systemübersicht	4
2.1 Gesamte Systemübersicht.....	4
2.2 Systemübersicht Feeder	5
2.3 Systemübersicht Master.....	5
2.4 Ablaufdiagramm	6
3 Lösungsfindung.....	7
3.1 Morphologischer Kasten	7
3.2 Recherchen.....	8
3.2.1 Transport des Streifens	8
3.2.2 Antrieb Transport	9
3.2.3 Schritterkennung	10
3.2.4 Abdeckung entfernen.....	11
3.2.5 Antrieb Abdeckungsentfernung	13
3.2.6 Softwarefunktionen.....	13
3.2.7 Mikrokontroller	13
3.2.8 Kommunikation	14
3.2.9 Master	16
3.2.10 Energieversorgung.....	16
3.2.11 Feederbank.....	17
4 Gesamtlösung.....	18
4.1 Gesamtkonzept	18
4.2 Feeder.....	19
4.3 Master	22
4.4 Feederbank.....	23
5 Entwicklung Lösungsvarianten	24
5.1 Mechanisches Konzept.....	24
5.1.1 Sprocket.....	24

5.1.2	Führung	27
5.1.3	Wand	31
5.1.4	Antrieb Sprocket.....	32
5.1.5	Peeler.....	35
5.1.6	Motorenhalter	37
5.1.7	Master Gehäuse	38
5.1.8	Feederbank.....	38
5.2	Elektrisches Konzept.....	39
5.2.1	Energieversorgung.....	39
5.2.2	Motorensteuerung	40
5.2.3	Lichtschränke.....	41
5.2.4	Switches & LED	43
5.2.5	Verbindungsaufbau Feeder	44
5.2.6	Mikrokontroller	45
5.2.7	Debug Interface	46
5.2.8	Feeder-PCB	46
5.2.9	Master-PCB.....	49
5.3	Software	50
5.3.1	Grundkonzept.....	51
5.3.2	Lichtschränke.....	51
5.3.3	Buttons	52
5.3.4	Application.....	53
5.3.5	Motoren.....	56
5.3.6	Kommunikation	57
6	Validierung	59
6.1	Prototyp V0.....	59
6.2	Prototyp V1.....	60
6.2.1	Peeler.....	60
6.2.2	Führung und Antrieb	60
6.2.3	Lichtschränke und Elektronik	60
6.3	Prototyp V2 und V2.1	61
6.4	Lichtschränke.....	62
6.4.1	Lichtschränken Signalmessung	62
6.4.2	Transistor und Komparator	63
6.5	Mikroswitch.....	67
6.5.1	Hebel-Endschalter	67

6.5.2	Knopf-Taster	68
6.5.3	Detektorschalter.....	68
6.6	Kommunikation	69
6.6.1	tinyK20 als Feeder	69
6.6.2	tinyK22 als Master	70
6.7	Energieversorgung.....	70
6.8	Benutzerfreundlichkeit.....	74
7	Diskussion.....	75
7.1	Projektmanagement.....	75
7.1.1	Projektplan	75
7.1.2	Risikomanagement	75
7.1.3	Arbeitsjournal	76
7.2	Auswertung Ziele.....	77
7.3	Optimierungen	79
7.3.1	Mechanischem Aufbau.....	79
7.3.2	Elektrischer Hardware	80
7.3.3	Software	80
7.4	Persöndlicher Rückblick.....	81
8	Bedienungsanleitung.....	83
9	Montageanleitung.....	83
10	Quellenverzeichnis	84
A	Anhang.....	86
A.1	Struktur Dokumentenanhang	86
A.2	Struktur CD.....	86
0_	Abgabedokument.....	86
1_	Projektmanagement.....	86
2_	Dokumente	86
3_	Datenblätter	86
4_	3D Daten.....	86
5_	CAD Dateien	87
6_	PCB Daten.....	87
7_	Sonstiges.....	87

1 Einleitung

1.1 Ausgangslage

Bestückungsautomaten, auch bekannt als pick-and-place machines (PnP) sind automatisierte Maschinen welche in der Fertigung von Leiterplatten dazu genutzt werden, Komponenten auf der Leiterplatte zu platzieren. Sie werden in der Industrie zur schnellen, genauen Bestückung von Leiterplatten genutzt. Durch die Miniaturisierung von Elektronik-Komponenten, insbesondere im Bereich von Surface Mount Technology (SMT), erlauben es solche Maschinen, wirtschaftlich Leiterplatten herzustellen.

Mit einem oder mehreren Bestückungsköpfen, welcher auf drei Achsen fahrbar ist und um die Z-Achse rotieren kann, werden die Bauteile auf der Platine platziert. Der Bestückungskopf holt das Bauteil von einer Rolle oder Tablett. Das Bauteil wird meist per Unterdruck angesaugt, danach mithilfe von einem optischen System die Winkel und Positionslage ermittelt, um danach an der korrekten Stelle auf der Leiterplatte platziert zu werden. Die Belieferung der Maschine mit Komponenten von Rollen wird mit sogenannten Feedern (Fütterungsautomaten) realisiert. Die Feeder führen die Rolle nach, entfernen die Abdeckung und legen das Bauteil offen für den Bestückungskopf dar. Nach dem Abholen des Bauteils wird der Streifen vom Feeder um ein Bauteil vorgefahren.

SMD Komponenten werden mehrheitlich in einem Streifen, welcher auf einer Rolle aufgewickelt ist, von den Herstellern geliefert. Die Bänder haben herstellerübergreifend einheitliche Grössen. Gängige Streifenbreiten sind 8,12, 16, 24, 32 und 44 mm. Die Streifen haben bis 24 mm auf einer Seite Löcher gestanzt, welche standardmässig in einem 4 mm Abstand sind. Breitere Streifen haben jeweils auf beiden Seiten eine Reihe von Löchern.

Mittels einer an der Hochschule gebauten PnP Maschine sollen Kleinserien von Platinen mit SMD Bauteilen bestückt werden. Das Ziel der Maschine ist der Einsatz in der Elektronik Werkstatt für studentische Arbeiten, zur Bestückung von Kleinserien und als Demonstrator an Führungen oder Messen. Diese Maschine basiert auf dem OpenPnP System. OpenPnP ist ein Open Source System welche Software und Hardware-Designs zur Verfügung stellt. Mit diesen Grundlagen kann eine eigene PnP Maschine gebaut und betrieben werden (OpenPnP, 2018).

1.2 Aufgabenstellung PAIND

In dieser Arbeit soll ein automatischer Feeder für SMD Bauteile erstellt werden. Der Feeder soll von einem Streifen oder Rolle neues Bestückungsmaterial zuführen. Dabei wird ein Streifen mit Bauteilen um jeweils ein Bauteil transportiert und gleichzeitig die Abdeckungsfolie entfernt. Hierzu soll eine Open Source Mechanik und Elektronik entwickelt werden, welche mit der an der HSLU entwickelter PnP Maschine zusammenarbeitet. Die komplette Aufgabenstellung ist im elektronischen Anhang im Ordner Projektmanagement abgelegt. Sie bietet die Grundlage, auf welche diese Arbeit realisiert wurde. Folgenden Funktionalitäten und Ziele sind aus der Aufgabenstellung gegeben.

- Kompakte Bauweise
- Kostengünstig
- Open-PnP kompatibel
- Mehrere Streifenbreiten
- Transport von Rollen und kurzen Streifen
- Nach Möglichkeit vorhandene Hardware verwenden
- Open Source

1.3 Anforderungen und Ziele

Aus der Aufgabenstellung und die damit verbunden Vorgaben wurden folgende Ziele festgelegt, welche mit der Realisierung des Feeders umgesetzt werden sollen.

Benutzerfreundlichkeit

Da die Maschine von Studenten sowie zur Demonstration genutzt wird soll der Feeder benutzerfreundlich und intuitiv genutzt werden können. Entsprechend sollten Einrichtung und Einstellung der Feeder mit möglichst wenig Aufwand erfolgen können.

Open-Source

Da die Mechanik und Elektronik auf Open-Source basiert werden, ist auch wünschenswert, dass das Konzept mit gängig erhältlichen Maschinen und Bauteilen erstellt wird. Dies beinhaltet beispielsweise Lasercutter und 3D Drucker. Als CAD Programm soll KiCAD zum Einsatz kommen.

Streifenbreiten

Das Konzept soll für Streifenbreiten von 8 bis 24mm kompatibel sein. Nach Möglichkeit soll das Konzept auch für weitere Streifenbreiten erweiterbar sein.

Bauteilgrösse

Im Rahmen der obengenannten Streifenbreiten sollen Komponenten mit einer Tiefe von maximal 12 mm transportiert werden. Als minimale Bauteilgrösse sollen mindestens Komponenten der Grösse 0805, und nach Abhängigkeit der mechanischen Toleranz auch Bauteile der Grösse 0603.

Grösse

Das Design des Feeders soll möglichst kompakt sein. Das wichtigste Mass ist die Breite des Feeders. Mit einem schmalen Feeder können mehrere Feeder nebeneinander an der PnP Maschine montiert werden, und somit auch eine grössere Anzahl an verschiedenen Komponenten der Maschine zugeführt werden.

Der zur Verfügung stehender Platz ist 340 mm breit. Der Feeder kann an dem Aluminium-Profil der PnP Maschine montiert werden, und in Richtung der Maschine ist ein Ausschnitt von 25mm vorgesehen, wo der leere Streifen unter der Maschine weggeführt werden kann.

Die Breite des Feeders wird als Streifenbreite + 15 mm festgelegt. Für 8 mm Streifen ergibt sich eine Gesamtbreite von 23 mm, für 12 mm Streifen 27mm, usw. Weitere Informationen zur Festlegung der Grösse und Dimensionierung des Feeders sind in Kapitel 3.2.1 zu finden.

Hardware

Wenn möglich sollen Mikrokontroller der Firma NXP verwendet werden, da Knowhow und Werkzeuge intern vorhanden sind. Eine Auswahl an kleinen Getriebemotoren und magnetische Encodern ist vorhanden.

Kosten

Die Materialkosten pro Feeder sollen unter 40 CHF liegen. Die Kosten für Arbeitszeit werden nicht berücksichtigt. Die Kosteneinschätzung stellte sich nach ersten Recherchen wie folgt zusammen.

Tabelle 1 Kosteneinschätzung pro Feeder

Position	Kosten
Elektronik Hardware	10.-
Motoren	12.-
PCB	10.-
weiteres Material (mech. Teile, 3D Druck)	8.-
Total	40.-

Funktionalitäten

Nebst Ziele, die den Aufbau des Feeders betreffen, sind auch die Funktionalitäten festzulegen. Aus der Aufgabenstellung ist definiert, dass der Feeder den Streifen jeweils um ein Bauteil transportieren soll. Zudem soll die Abdeckung entfernt werden und der Feeder mit der OpenPnP Software kompatibel sein.

Nebst der Ansteuerung via OpenPnP wurde als Ziel definiert, dass der Streifen im Feeder per Taster Vor- und Rückwärts gefahren werden kann. Somit kann der Streifen einfach im Feeder eingeführt sowie entfernt werden.

Es soll eine LED als visuelle Status-Meldung implementiert werden.

Die Feeder soll bis zur nächsten Abholbewegung des Bestückungskopfes ein neues Bauteil bereitstellen. Die von Prof. Styger genannte Zykluszeit des Bestückungskopfes ist, zum Abklärungszeitpunkt, bei ca. 5 Sekunden pro Bauteil. Entsprechend ist der benötigte Vorschub des Streifens relativ niedrig. Als Ziel wurde ein Geschwindigkeitsvorschub von 4mm pro Sekunde gewählt. Bei Bauteilen der Serie 0603 und 0805, welche Grösstenteils eingesetzt werden, ist 4 mm der Abstand zwischen den Bauteilen (Vishay, 2017). Somit wird für einen Bauteilschritt 1 Sekunde benötigt. Es ist eine Reserve eingeplant, für den Fall das die PnP Maschine schneller wird.

2 Systemübersicht

Dieses Kapitel bietet eine Übersicht über das gesamte System von der OpenPnP Software, der PnP Maschine und dem Feeder.

2.1 Gesamte Systemübersicht

Abbildung 1 zeigt das gesamte System auf. Die OpenPnP Software, welche auf einem PC läuft, kommuniziert über eine USB-Schnittstelle mit der PnP Maschine und einem Master-Gerät für die Feeder. Das Master-Gerät kommuniziert über ein eigens definiertes Protokoll mit den Feedern. Die Feeder werden extern mit 5 VDC gespeisen. Die PnP Maschine, durch ihre grösseren Aktoren, wird mit 24 VDC gespeisen. Die Streifen, welche von den Feedern an die Maschine transportiert werden, sind auf Rollen in einer Halterung montiert, welche nachfolgend Feederbank benannt wird.

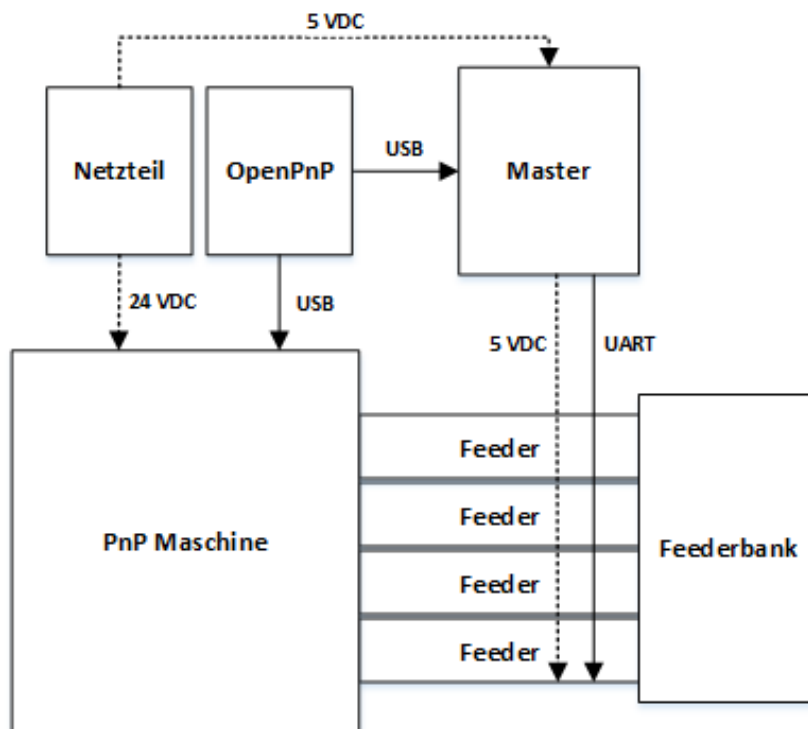


Abbildung 1 Gesamt-Systemübersicht

Im Rahmen dieses Projekts wurde ein Konzept erstellt, welches den Master, die Feeder und die Feederbank beinhaltet. Die Lösungsfindung sowie das erstellte Gesamtkonzept werden in den folgenden Kapiteln beschrieben.

2.2 Systemübersicht Feeder

Der Feeder hat fünf Teilfunktionalitäten die als Aufgaben umzusetzen sind. Diese umfassen:

- den Transport des Streifens
- die Schritterkennung (der Transport um ein Bauteil)
- die Abdeckung entfernen (Peeler)
- die Kommunikation (zwischen Feedern und OpenPnP)
- die Umsetzung obengenannter in der Software

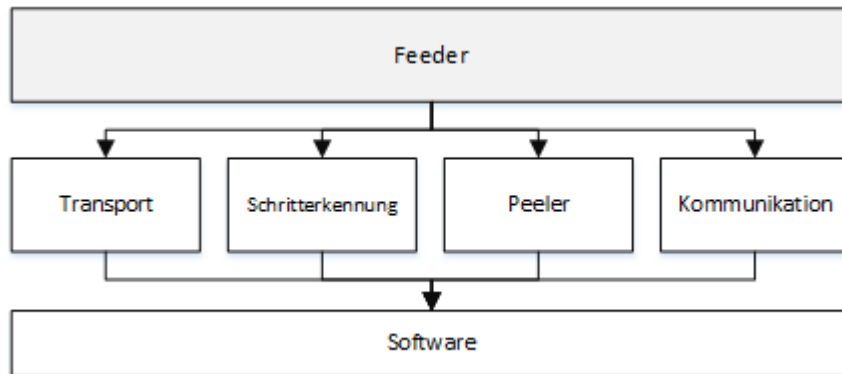


Abbildung 2 Übersicht Feederfunktionen

Abbildung 2 zeigt eine vereinfachte Systemübersicht des Feeders. Diese Teilfunktionalitäten beinhalten weitere Teilaufgaben. Eine entsprechende Recherche zur mechanischen und elektrischen Umsetzung sowie zu den Aktoren und Sensoren der Teilfunktionalitäten folgt in Kapitel 3.

2.3 Systemübersicht Master

Der Funktion des Masters ist es eine Schnittstelle zwischen OpenPnP und den Feedern zu bilden. Neben der Kommunikation wird auch die Energieversorgung über den Master geführt. Dies ist in Abbildung 3 dargestellt.

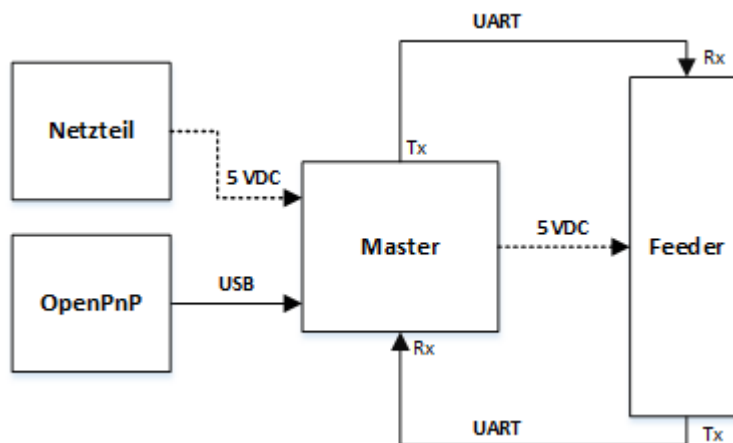


Abbildung 3 Systemübersicht Master

2.4 Ablaufdiagramm

Im Ablaufdiagramm in Abbildung 4 wird eine Übersicht der Funktionalität zwischen OpenPnP, Feeder und Master geliefert. Die einzelnen Teilschritte, respektive Komponenten werden in weiteren Kapiteln dieser Dokumentation beschrieben.

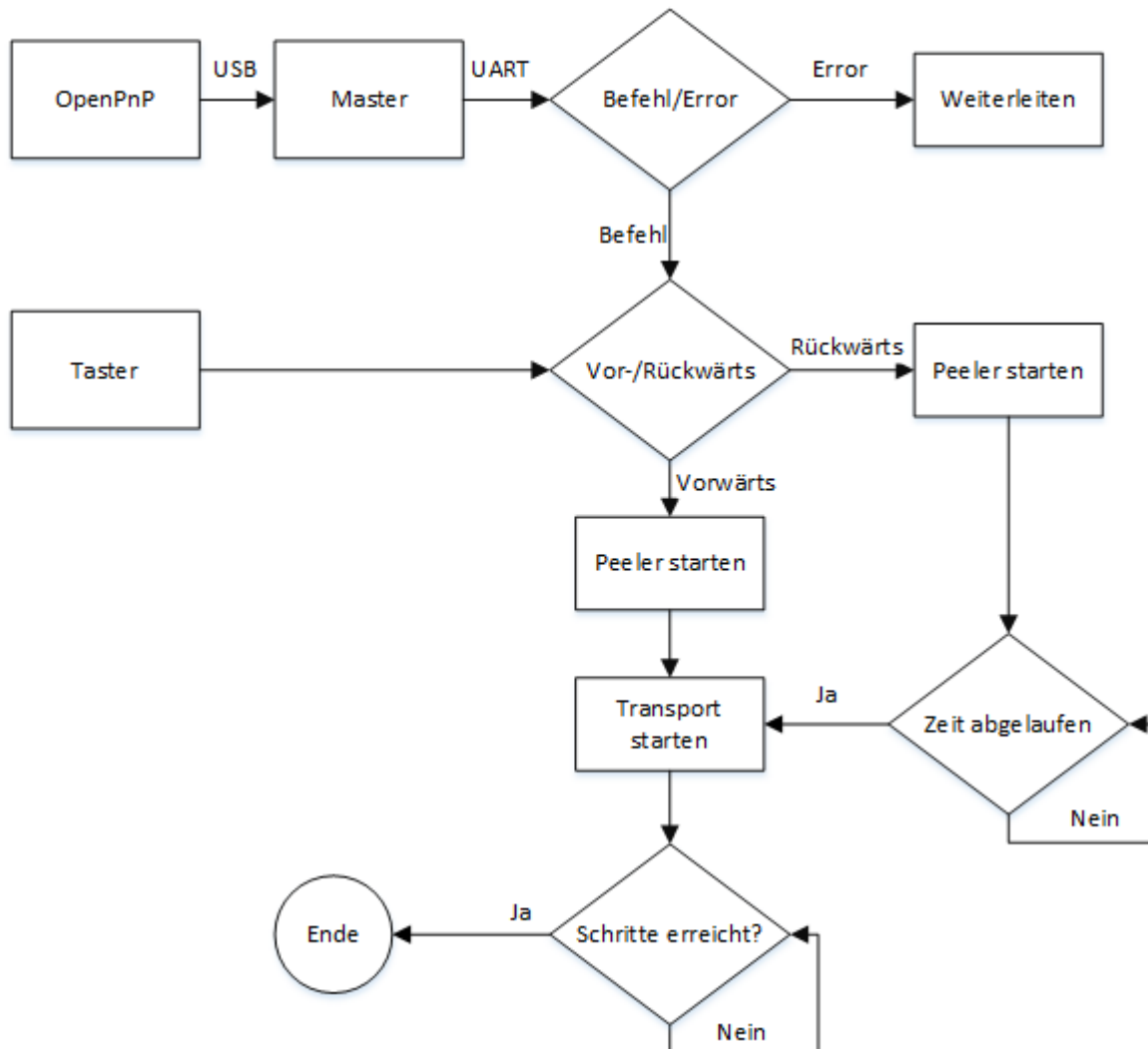


Abbildung 4 Ablaufdiagramm

3 Lösungsfindung

In diesem Kapitel werden die Teilfunktionalitäten des Systems in weitere Teilaufgaben unterteilt. Zu den einzelnen Positionen wird mit Unterstützung von Recherchen die gewählte Lösungsvarianten dargelegt. Der Entwicklungsprozess der gewählten Lösung wird in Kapitel 5 beschrieben.

Für die Entwicklung des Konzepts wurden industrielle sowie sogenannte Hobby-Lösungen für Feeder untersucht. Dabei wurde beachtet, wie die Teilfunktionen gelöst wurden und welche Vor- und Nachteile sich daraus ableiten liessen. Industrielle Feeder stechen insbesondere durch ihre hohe Geschwindigkeit heraus. Mit dieser hohen Geschwindigkeit entwickeln sich auch weitere Probleme, wie beispielsweise das Herausspicken von Komponenten aus dem Streifen, welche gelöst werden müssen. Hobby-Lösungen haben meist simple Ansätze, die bei niedriger Geschwindigkeit gut funktionieren und kostengünstig umzusetzen sind.

Für die Entwicklung des Konzepts wurde eine Kombination von Lösungsansätzen beider Welten einbezogen. Das in diesem Projekt erstellte Konzept hat eine grosse Ähnlichkeit wie ein Prototyp für OpenPnP von Hans Jørgen Grimstad (Grimstad, 2018). Seine Lösungsvariante stach durch ihre interessante Art heraus. Seine Lösung verwendete auch eine 3D gedruckte Führung und PCBs als Wände. Dennoch wurden alle mechanischen Komponenten dieses Konzepts selbst gezeichnet, getestet, verbessert und verwendet.

Als Unterstützung zum Vergleich der Lösungsvarianten wurde teilweise eine tabellarische Gewichtung vorgenommen. Die Eigenschaft der Lösungsvarianten wurden mithilfe einer Note von 1 bis 5 bewertet, wobei je höher desto besser gilt. Zusätzlich wurde jede Eigenschaft von 1-3 gewichtet, wobei 1 wünschenswert und 3 wichtig bedeutet. Bei jeder Lösungsvariante wurde danach die Summe der Eigenschaften (Note x Gewichtung) gebildet. Die Lösungsvariante mit der höchsten Summe wurde entsprechend für die Umsetzung gewählt.

3.1 Morphologischer Kasten

Als Übersicht der recherchierten Lösungsvarianten und zur Definition der gewählten Lösungskombination wurde ein Morphologischer Kasten erstellt. Die Grundlage für die Analyse diente eine ausgiebige Recherche der Teilaufgaben welche in Kapitel 3.2 aufgeführt ist. Die ausgewählte Lösungsvariante wurde in der Spalte von Variante 1 eingetragen.

Tabelle 2 Morphologischer Kasten

Teilaufgabe	Variante 1	Variante 2	Variante 3
Transport des Streifens	Sprocketfeeder	Dragfeeder	
Antrieb Transport	DC-Getriebemotor	Schrittmotor	Servomotor
Schritterkennung	Inkrementalgeber	Hallsensor	Lichtschanke am Streifen
Abdeckungs-entfernung	Zahnrad/Rollen	Messer/Umlenkung	Schwerkraft
Antrieb Abdeckungs-entfernung	DC-Getriebemotor	Schrittmotor	
Kommunikation	UART	I2C	
	Lösungsvariante		

3.2 Recherchen

3.2.1 Transport des Streifens

Streifen mit SMD Komponenten sind für den Transport des Streifens mit einer Reihe Löcher ausgestattet. Aus Abbildung 5 kann der Aufbau von einem Streifen interpretiert werden. Die Löcher sind einem definierten Abstand P_0 zueinander. Dieser Abstand kann aus dem Datenblatt verschiedener Hersteller entnommen werden und ist für gängige Streifen ab 0603 Komponentengrößen 4 mm.

Es gibt in der Industrie zwei Verfahren um mit diesen Löchern das Band zu Transportieren.

8, 12, 16 AND 24 mm EMBOSSED TAPE DIMENSIONS

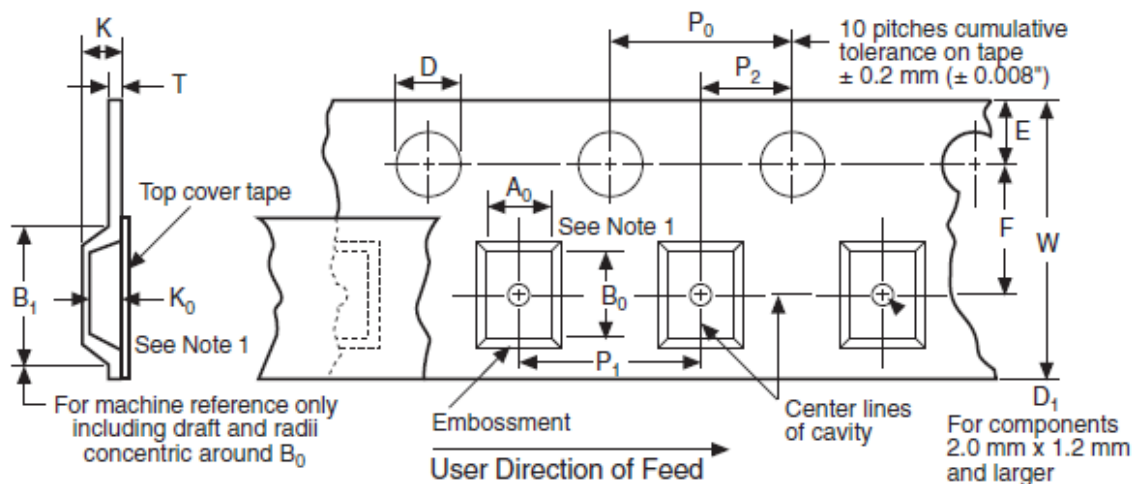


Abbildung 5 Dimensionen eines SMD Streifens (Vishay, 2017)

Dragfeeder

Der Dragfeeder zieht oder stösst das Band mithilfe eines Stiftes in einem Loch. Die Bewegungen können mit mechanischer Hebelwirkung oder mit einem eigenen elektrischen Antrieb durchgeführt werden. Der Dragfeeder kann mit einer einfachen Mechanik umgesetzt werden. Jedoch ist mit dieser nur eine Vorwärtsbewegung möglich. (Mortezix, 2017)

Sprocketfeeder

Sprocket, welches mit Kettenrad oder Zahnkranz aus dem Englischen übersetzt werden kann, behandelt den Streifen als Kette. Die rotierende Bewegung des Zahnkranzes fördert den Streifen wie eine Kette vorwärts, wobei die Zähne des Kranzes immer erneut in die Löcher des Streifens greifen. Der Sprocket wird durch einen elektrischen Motor angetrieben.

Evaluation

Obwohl der Dragfeeder mit einer einfachen Mechanik umgesetzt werden kann, hat der Sprocketfeeder mehrere Vorteile. Zum einen ist bei dem Dragfeeder der Weg durch den mechanischen Aufbau definiert. Jede Hubbewegung resultiert in einem gleichen Schritt, beispielsweise 4 mm. Um einen Weg von 12 mm zu erreichen muss der Feeder drei Mal betätigt werden. Der Sprocketfeeder kann einen variablen Weg ausführen. Mit erweiterter Umsetzung des Dragfeeders wäre eine Rückwärtsbewegung möglich, jedoch fokussiert dies Arbeit mehrheitlich auf die elektrotechnischen Aspekte und nicht auf das Gebiet des Maschinenbaus. Mit dem Sprocket-Antrieb kann der Streifen Vor- und Rückwärts

bewegt werden. Entsprechend wird der Transport des Streifens mithilfe eines elektrischen Antriebes und einem Sprocket realisiert werden.

3.2.2 Antrieb Transport

Der Transport des Streifens muss dieses angetrieben werden. Da im Kapitel 3.2.1 für die Umsetzung eines Sprocketfeeders entschieden wurde, muss der Antrieb eine uneingeschränkte Rotationsbewegung ausführen können. Da der Antrieb zudem positionsgenau anhalten muss und entsprechend auch tiefe Drehzahlen gewünscht sind, stehen zwei Typen elektrischer Antriebsmotoren zur Verfügung. Zudem gilt zu beachten, dass die Motoren eine möglichst kompakte Bauform aufweisen, damit der Feeder nicht zu breit wird.

DC-Getriebemotor

DC-Getriebemotoren bieten durch ihr Getriebe niedrige Drehzahlen an, kombiniert mit einer PWM Ansteuerung ist es möglich die Drehzahl noch weiter zu variieren. Mit dem Einsatz eines Drehgebers kann auch die Position des Motors eruiert werden. DC-Getriebemotoren sind zudem in kleiner Bauform zu einem tiefen Preis erhältlich. An der HSLU werden DC-Getriebemotoren der Bauform N20 für verschiedene Projekte eingesetzt. Ein solcher DC-Getriebemotor ist in Abbildung 6 dargestellt.



Abbildung 6 N20 DC-Getriebemotor
(TinySine, 2018)

Schrittmotor

Schrittmotoren bieten durch ihre niedrige Drehzahl und des Winkelschrittes gerade zwei wichtige Punkte für den Antrieb. Für den Feeder müsste ein Schrittmotor kompakter Bauweise verwendet werden. Aufbau in flacher Pancake-Form resultiert in Kosten grösser als für den Feeder als Ziel gesetzt wurde. Auch für kleinere Motoren der Grösse NEMA 8 sind die Kosten höher als bei vergleichbaren DC-Getriebemotoren (Pololu, 2018b). Die Gefahr des Schrittverlustes ist bei hoher Last auch gross und würde allenfalls eine zusätzliche Positionsüberwachung nötig machen, welche auch bei den DC-Getriebemotoren zum Einsatz kommt.

Servomotor

Für den Antrieb eines Feeders welche auf dem Drag-Prinzip beruht könnte ein Servomotor genutzt werden. Diese Lösungsvariante wird nicht weiter untersucht, da entschieden wurde ein Sprocketfeeder zu erstellen.

Evaluation

Tabelle 3 Evaluation Antrieb Transport

	DC-Getriebemotor	Schrittmotor	Gewichtung
Preis	5	2	3
Position	2	5	2
Drehzahl	3	4	2
Grösse	5	4	3
Spannung	4	4	1
Strom	4	2	2
Total	52	44	

Folglich wird für den Antrieb des Transports ein DC-Getriebemotor der Reihe N20 verwendet werden. Die genaue Auslegung des Motors wird im Kapitel 5.1.4 vorgestellt. Die Recherche zur Schritterkennung folgt im nächsten Kapitel.

3.2.3 Schritterkennung

Um den Streifen mit SMD Bauteilen jeweils um ein Bauteil zu verschieben, muss eine Positionserkennung des Streifens umgesetzt werden. Die Verwendung des DC-Motors als Antrieb des Bandes erfordert entsprechend eine Schritterkennung. Im Überblick können drei Varianten umgesetzt werden um die Position festzustellen.

- Positionserkennung am DC Motor (Drehgeber/Encoder)
- Positionserkennung anhand der Löcher im Streifen
- Positionserkennung anhand des Sprockets (Inkrementalgeber)

Um eine effiziente Regelung umzusetzen, sollte die Regelstrecke möglichst kurz sein um Fehler zu vermeiden. Die Positionierung des Bauteils kann mit einer relativen Positionierung erfolgen. Es muss nur gewährleistet werden, dass immer derselbe Weg zwischen zwei Bauteilen gefahren wird, und nicht die absolute Position des Streifens ermittelt werden.

Hallsensor

Die N20 DC-Motoren werden bereits in anderen Projekten an der HSLU eingesetzt, beispielsweise dem SUMO-Roboter aus dem Modul Infotronik. Bei diesem Fahrzeug werden Hallsensoren mit einem magnetischen Drehgeber verwendet um die Umdrehungen des Motors festzustellen. Eine solche Umsetzung wäre auch beim Feeder möglich. Die hohe Auflösung würde eine sehr genaue Positionierung des Bauteils erlauben. Nachteilig ist jedoch die lange Regelstrecke von der Antriebswelle des Motors bis zur Position des Streifens. Der DC-Motor besitzt ein Getriebe, welches ein weiteres Schneckengetriebe antreibt, an dem wiederum das Sprocket montiert ist welches den Streifen transportiert. Die mechanischen Toleranzen in diesen Übersetzungen könnten sich als Fehlerquelle in der Positionsregelung widerspiegeln.

Löcher-Erkennung am Streifen

Da der Streifen bereits mit definierten Löchern ausgestattet ist, könnten diese zur Positionserkennung verwendet werden. Als geeignete Lösung würde sich eine Locherkennung mit einer Lichtschranke anbieten. Mit einer zweiten Lichtschranke würde sich auch eine höhere Auflösung umsetzen lassen. Diese Lösung ist kostengünstig und einfach umzusetzen. Nachteilig an dieser Lösungsvariante ist, dass es Hersteller gibt welche ihre SMD-Bauteile in Streifen aus durchsichtigem Kunststoff liefern. Entsprechend würde die Lichtschranke in dieser Situation ihre Funktion nicht erfüllen können.

Inkrementalgeber

Eine weitere Variante der Positionsermittlung lässt sich als Inkrementalgeber umsetzen. Am Sprocket, welches den Streifen direkt antreibt, können direkt Schlitzze, respektive Bohrungen, gesetzt werden, und mithilfe einer Lichtschranke die Drehbewegung des Sprockets erkannt werden. Die Auflösung kann hier durch die Anzahl Bohrungen sowie einer zusätzlichen Lichtschranke erhöht werden. Mit dieser Variante ist die Schritterkennung unabhängig von dem Streifen, und die Umgebungsbedingungen bleiben konstant. In Abbildung 7 ist eine solche Umsetzung ersichtlich.

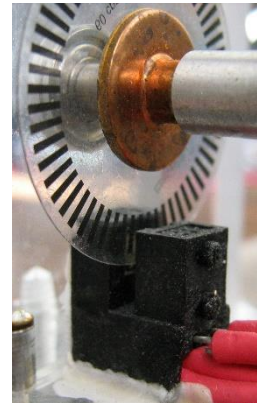


Abbildung 7 Optischer Inkrementalgeber (Tycho, 2008)

Evaluation

Tabelle 4 Evaluation Schritterkennung

	DC-Motor-Drehgeber	Locherkennung	Inkrementalgeber	Gewichtung
Preis	2	5	4	2
Auflösung	5	3	4	2
Genauigkeit	4	4	4	3
Fehlerquellen	4	3	5	3
Komplexität	2	4	4	2
Total	42	45	51	

Aus Tabelle 4 kann entnommen werden, dass für die Umsetzung der Schritterkennung ein optischer Inkrementalgeber umgesetzt werden soll. Die kurze Regelstrecke, die konstanten Umgebungsbedingungen und die definierbare Auflösung machen eine simple, aber effektive Umsetzung der Schritterkennung möglich. Zudem wird das Sprocket als Indikator verwendet.

3.2.4 Abdeckung entfernen

Als definierte Teilfunktion des Feeders muss auch die Abdeckfolie von dem Streifen entfernt werden. Diese Folie hält das Bauteil sicher im Streifen bis zur Entnahme durch die PnP Maschine. Für die Entfernung der Abdeckungsfolie bieten sich mehrere Lösungen unterschiedlicher Komplexität an. Für die Auswahl der Lösung waren Kriterien wie Zuverlässigkeit, einfache Bedienung und einfache Umsetzung relevant. Im Englischen wird im Zusammenhang mit den Feedern oftmals von dem Peeler gesprochen, welches vom Verb peel (abziehen, schälen) abgeleitet wird. In dieser Dokumentation wird auch dieser Begriff verwendet.

Messer

Diese rein mechanische Lösung hat keine beweglichen Teile. Mit einem messerartigen Metallstück wird die Folie von dem Streifen teils gelöst. Mit einer Führung wird die Folie umgeklappt und das SMD-Bauteil offengelegt. Nach der Abholstelle für den Bestückungskopf wird die Folie wieder zurückgeklappt und mit dem Streifen abgeführt. Eine solche Umsetzung erfordert, dass bei der Einführung eines Streifens in den Feeder, die Kombination von Messer, Streifen und Folie absolut korrekt zueinanderstehen. Abweichungen durch Streifendicke und/oder der Platzierung der aufgeklebten Folie auf dem Streifen erschweren den Betrieb dieser Lösung.

Schwerkraft

Als einfachste Lösung im Hobby-Bau von Feedern wird die Schwerkraft dazu genutzt die Folie abziehen. Die Folie wird zuerst von Hand vom Streifen gelöst und durch eine Umlenkung geführt. Danach wird ein Gewicht an der Folie angehängt. Mit einem Vorschub des Streifens wird auch ein neuer Abschnitt Folie freigelegt. Das Gewicht bewirkt durch die Umlenkung einen Zug auf die Folie aus. Dieser Zug löst die Folie von dem Streifen, bis ein Anschlag erreicht wird. Diese Variante verlangt, dass der Feeder eine gewisse Höhe über dem Boden/Tisch montiert ist, damit auch eine vernünftiger Strecke Folie abgezogen werden kann. Zudem muss der Benutzer das Gewicht regelmässig wieder oben an der Folie befestigen.

Zahn-/Gummirad

Mithilfe einer angetriebenen Lösung kann die Folie stets abgezogen werden, ohne dass eine Überwachung durch den Benutzer vonnöten wird. Die Folie kann, beim Einführung in den Feeder, zwischen zwei Zahn- oder Gummiräder eingeklemmt werden. Die Räder sind zusätzlich angetrieben. Bei einem Vorschub des Streifens durch den Transport-Antrieb wird wiederum Folie freigelegt, somit kann auch der Abziehmechanismus starten. Die Räder/Rollen drehen und ziehen die eingeklemmte Folie mit. Über einen definierten Weg oder durch eine Detektion der Folie kann der Antrieb wieder ausgeschaltet werden.

Evaluation

Tabelle 5 Evaluation Abdeckungsentfernung

	Messer	Schwerkraft	Radantrieb	Gewichtung
Zuverlässigkeit	3	3	4	3
Überwachung	4	2	5	2
Komplexität	3	4	3	2
Total	23	21	28	

Die Evaluation der Abdeckungsentfernung in Tabelle 5 gibt die Variante des Radantriebes als Lösung vor. Als einklemmende Mechanik werden vorzugsweise Zahnräder verwendet. Diese können kostengünstig als 3D-Druck Teile hergestellt werden, wohingegen Gummiräder eingekauft werden müssten. Die Variante mit dem Messer und dem Umlappen der Folie wurde zudem verworfen, weil in erster Recherche davon ausgegangen wurde, ein Feeder für alle Streifenbreiten zu erstellen. Somit wäre eine zuverlässige mechanische Umsetzung allenfalls sehr aufwendig geworden.

3.2.5 Antrieb Abdeckungsentfernung

Die Abdeckungsentfernung mit Zahnräder, welche die Folie durch einklemmen abziehen, benötigt einen Antrieb. Da eine unendliche Rotationsbewegung gefahren wird, kommen wie beim Antrieb des Transports dieselben elektrischen Antrieben in Frage. Zusätzlich wäre es möglich, den bereits vorhandenen Antrieb vom Transport auch für die Abdeckungsentfernung zu nutzen.

Das Abziehen der Folie muss auch nur einen definierten Weg pro Bauteilvorschub geschehen. Nach diesem Weg sollte der Antrieb ausgeschaltet werden. Da zwischen den Zahnrädern und Folie ein gewisser Schlupf als Risiko besteht, sollte der Weg nicht absolut definiert werden. Es soll eine Detektion der Folie eingesetzt werden. Aus Kosten- und Komplexitätsgründen wird auf einen einfachen Endschalter gesetzt. Sobald die Folie genug weit abgezogen wurde und diesen betätigt, wird auch der Antrieb ausgeschaltet. Entsprechend entfällt somit auch die Lösungsvariante, bei dem der Antrieb des Sprockets auch für den Antrieb dieser Zahnräder verwendet werden kann.

Evaluation

Die Evaluation kann an dieser Stelle aus Tabelle 3 entnommen wird. Gleich wie beim Antrieb des Transports wird ein N20 DC-Getriebemotor verwendet werden. Da zusätzlich ein Endschalter für die Positionserkennung verwendet wird, entfällt der Vorteil des Schrittmotors mit Winkelschritten komplett.

3.2.6 Softwarefunktionen

Für die Umsetzung der Teilfunktionalitäten wird auch Software benötigt. Nach einem Vorwärtsbefehl müssen die Antriebe eingeschaltet werden. Der Weg des Streifens muss über die Schritte erkannt werden, und entsprechend der Antrieb ausgeschaltet werden. Das Ansprechen des Endschalters führt dazu das auch der Abzieh-Antrieb ausgeschaltet wird. Um überhaupt den Einschaltbefehl zu erhalten muss eine Kommunikation mit OpenPnP vorhanden sein, respektive die manuellen Taster müssen eingelesen werden. Zudem sollte für den Benutzer in einer visuellen Form den Status des Feeders angezeigt werden. Zusammenfassend werden die Softwarefunktionen wie folgt definiert:

- Schritterkennung am Sprocket
- Detektion des Endschalters
- Ansteuerung beider Motoren (mit PWM und Richtung)
- Statusanzeige per LED
- Kommunikation mit OpenPnP

Solche vielfältigen Funktionen mit Hardware lassen sich mit einem Mikrocontroller umsetzen. Auf die definitive Softwareumsetzung wird in Kapitel 5.3 eingegangen.

3.2.7 Mikrocontroller

Die Wahl des Mikrocontrollers wurde durch die Aufgabenstellung vereinfacht. Als empfohlene Vorgabe sollten Mikrocontroller der Firma NXP verwendet werden. Nach Definition der Softwarefunktionalitäten, und im Zusammenhang mit dem Modul Infotronik, wurde mit Prof. Styger entschieden das Real-Time Betriebssystem FreeRTOS einzusetzen. Obwohl für die Funktionalitäten ein solches Betriebssystem nicht nötig wäre, bietet sie eine gute Grundlage um das System umzusetzen, das Potenzial von einfachen Einbindung von Erweiterungen, sowie einen grossen Lerneffekt für den Studenten.

Bereits im Rahmen von Infotronik wurde von Prof. Styger das Potential von der an der HSLU entwickelten tinyK20 und tinyK22 angesprochen, welche sich für die Verwendung in Industriearbeit

und Bachelorarbeit anbieten würden. Entsprechend wurden auch die ersten Versuchsaufbauten mit solchen Mikrocontrollerboards umgesetzt. Da der tinyK20, basierend auf dem Mikrocontroller MK20DX128VFT5 von NXP, alle benötigten Funktionen durchführen kann, und zudem kostengünstiger als der MK22FN512VLH12 vom tinyK22 ist, wird für den Feeder ein solcher Mikrocontroller eingesetzt.

Die folgende Tabelle stellt eine Übersicht beider Mikrocontroller dar (Mouser, 2018a und 2018b).

Tabelle 6 Vergleich Mikrocontroller

Attribut	MK20DX128VFT5	MK22FN512VLH12
Kern	ARM Cortex M4	ARM Cortex M4 (FPU)
Busweite	32	32/16
Maximale Taktfrequenz	50 MHz	120 MHz
Programmspeicher	128 kB	512 kB
RAM	16 kB	128 kB
Anzahl I/O	29	40
Stückpreis	5.80	7.76

3.2.8 Kommunikation

Schnittstelle

OpenPnP muss den Feedern einen Befehl senden können, damit diese einen Schritt machen. Das OpenPnP System kann über eine USB-COM Schnittstelle direkt mit einem Gerät mit kompatibler Software kommunizieren. An der PnP-Maschine ist ein solches Board integriert, in diesem Fall ein Smoothieboard (Smoothie, 2018). Dieses steuert die einzelnen Hardwarekomponenten (Motoren, Bestückungskopf, usw.) der Maschine gemäss den Befehlen von OpenPnP an. Eine weiterführende Kommunikation von dem Smoothieboard zu den Feedern wäre eine erste Lösungsvariante. Da das Smoothieboard gemäss Prof. Styger stark ausgelastet ist, kommt nur eine Kommunikation via I2C in Frage. Jedoch ratet Prof. Styger von der Umsetzung eines Slave-Devices auf einem Mikrocontroller ab. Dieses sei nur mit erheblichem Aufwand sicher umsetzbar.

Da somit die Schnittstelle des Smoothieboards entfällt, muss eine direkte Kommunikation mit OpenPnP umgesetzt werden. Da eine USB-Schnittstelle für jeden Feeder keinen Sinn macht, da bei 20 Feeder auch 20 USB-Steckplätze nötig wären, soll eine Master-Slave Struktur erstellt werden. Das Master-Device kommuniziert via der USB-Schnittstelle mit OpenPnP. In einem weiteren Schritt werden die Befehle vom Master interpretiert und an die Feeder weitergeleitet.

Nach dieser Definition der Master-Slave Topologie benötigt es eine Kommunikation zwischen diesen Geräten. Als einfache Variante bietet sich die Kommunikation via UART an. Die Einbindung dieser mit den ausgewählten MK20 Mikrocontroller ist einfach, da definierte Pins und Treibersoftware vorhanden sind. Von dem Softwareentwickler muss nur noch die Struktur der übertragenen Befehle definiert und implementiert werden.

Struktur

Folglich galt es noch die Struktur der UART Kommunikation zu definieren. Oftmals wird in Kommunikationen mit mehreren Teilnehmern eine Adressierung vorgenommen. Damit wird jedem Teilnehmer eine Adresse zugeteilt, womit er sich identifizieren kann. Befehle werden mit einem Adresse-Parameter ausgestattet, welche es ermöglicht den Teilnehmer direkt anzusprechen. Andere Teilnehmer reagieren nicht auf Befehle mit einer Adresse, welche nicht ihrer entspricht.

Die Adresse eines Feeders könnte beispielsweise mit einem DIP-Switch direkt am Feeder eingestellt werden. Somit müsste man beim Einrichten der Maschine überprüfen welcher Feeder welche Adresse hat, und dies auch so im Master oder OpenPnP hinterlegen. Bei einem Wechsel eines Feeders müsste das ganze Prozedur wiederholt werden. In Bezug auf Benutzerfreundlichkeit soll man möglichst wenig an den Feedern einstellen müssen. Entsprechend wurde entschieden eine andere Implementation zu verwenden.

Durch den Input eines HSLU-Mitarbeiters wurde die Anwendung eines Daisy-Chains (englisch für «Gänseblümchenkette») Aufbau empfohlen. Dieser Aufbau sieht vor, dass die Feeder in Serie zum nächsten stehen. Bei einem Aufbau mit UART heisst dies, dass jeweils der Sender (Tx) von einem Feeder mit dem Empfänger (Rx) vom nächsten Feeder verbunden ist. Folgende Darstellung soll den Aufbau verdeutlichen.

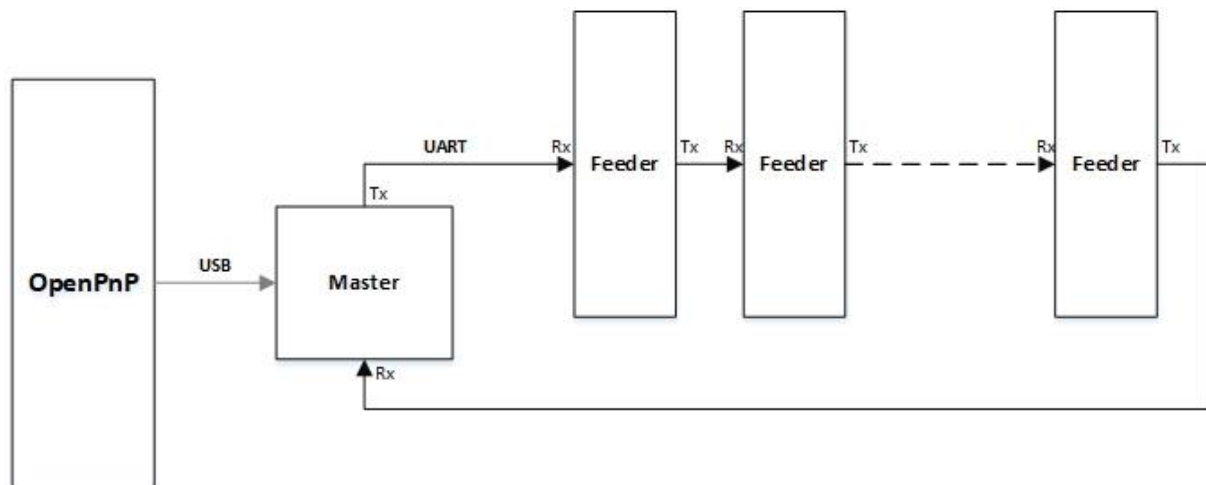


Abbildung 8 Kommunikationsverbindungen

Das OpenPnP kommuniziert über USB mit einem Master, welcher dann die Befehle per UART an den ersten Feeder weiterleitet. Falls die Adresse im Befehl nicht seiner entspricht, leitet der Feeder den Befehl an den nächsten Feeder weiter. Dies wird wiederholt bis ein Feeder seine Adresse erkennt und den Befehl ausführt. Eine Rückmeldung kann in selber Form weitergegeben werden bis es vom letzten Feeder wieder an den Master zurückgeleitet wird.

Adressierung

Die Implementation der Adressierung wird in diesem Fall mit einer Kombination von der Anzahl Feeder und dekrementieren implementiert. Jeder Feeder besitzt die Adresse 0 und reagiert nur auf jene Befehle mit dieser Adresse. Der Master, respektive das OpenPnP, kennt die Anzahl Feeder, und weiss auch in welcher Reihenfolge die Feeder an der PnP Maschine stehen. Dies muss vor dem Starten des Jobs in der OpenPnP Benutzeroberfläche eingerichtet werden. Um den dritten Feeder anzusprechen wird im Befehl die Position des Feeders als Adresse mitgegeben. Dieser Befehl wird vom Master an den ersten Feeder geleitet. Dieser untersucht die Adresse im Befehl, erkennt eine Zahl ungleich 0, und dekrementiert sie. Er packt den Befehl wieder und leitet es an den nächsten Feeder weiter. Dieser untersucht wiederum die Adresse, erkennt eine Zahl ungleich 0, und dekrementiert sie. Dies geht so lange weiter bis ein Feeder im Befehl die Adresse 0 auspackt. Dieser Feeder führt den Befehl aus und kann seinerseits eine Rückmeldung absenden.

Dieser Aufbau ermöglicht eine einfache Implementation von Befehlen und hält den Einrichtungsaufwand für die Feeder niedrig.

Auf die Feeder-Verbindungen wird im Kapitel 5.2.5 eingegangen. Die Kommunikation und die Implementierung der Befehle wird in Kapitel 5.3.6 verdeutlicht.

3.2.9 Master

Für die Kommunikationsverbindung zwischen OpenPnP und den Feedern wird ein Master eingesetzt, welcher als Schnittstelle zwischen USB und UART fungiert (Kapitel 3.2.8). Die Software für diesen Master wurde von Prof. Styger erstellt. Die Kommunikationsbefehle wurden in Absprache erstellt und in Feeder und Master implementiert. Sie folgen in Kapitel 5.3.6. Die Implementierung der Mastersoftware als Teil dieser Arbeit hätte den Rahmen gesprengt.

Von Prof. Styger wurde entschieden ein tinyK22 einzusetzen. Ein Vorteil dieses Boards ist es, dass es bereits eine USB Schnittstelle integriert hat. Für die Anbindung des Masters am Feeder-System wurde ein Gehäuse und PCB erstellt, welche in Kapitel 5.1.7 und 5.2.9 ausführlich beschrieben werden.

3.2.10 Energieversorgung

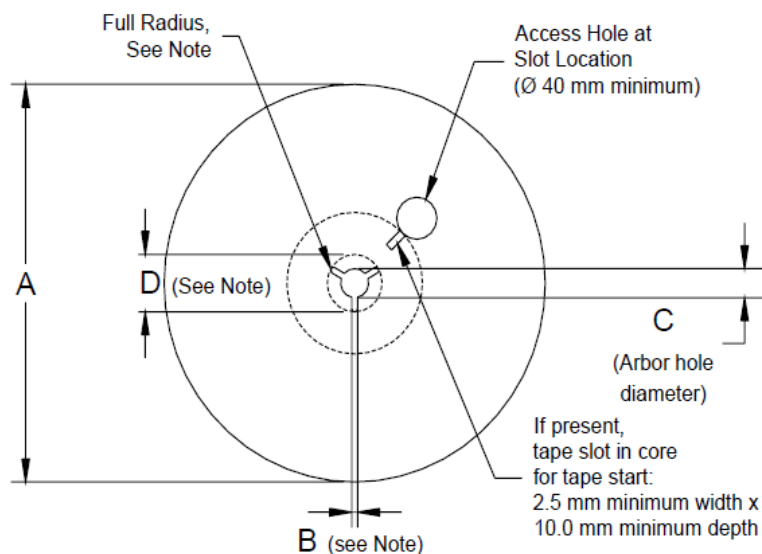
Die elektrischen Komponenten auf dem Feeder benötigen zwei verschiedene Spannungspotentiale. Die Logik mit dem Mikrokontroller benötigt 3.3 VDC. Die N20 DC-Motoren sollen laut Datenblatt mit 6 VDC betrieben werden. Aus Erfahrung im Infotronik laufen die Motoren jedoch auch problemlos mit einer Spannungsversorgung im Bereich von 5-8 VDC. Da eine Übertragung von 2 Potentialen in erhöhten Kabelbedarf widerspiegelt, und der Spannungsabfall bei kleinen Potentialen sich stärker bemerkbar macht, wird an jedem Feeder nur die höhere Spannungsversorgung zu Verfügung gestellt. Jeder Feeder soll einen eigenen Spannungsregler haben, welcher die 3.3 VDC für den Mikrokontroller und weiteren elektronischen Komponenten bereitstellt.

Da 5 VDC in der Welt der Elektro-Komponenten allgegenwärtig ist und sich entsprechende Netzteile kostengünstig finden lassen wird als höheres Spannungspotential 5 VDC verwendet. Die genaue Umsetzung der Energieversorgung sowie die Ermittlung der benötigten Energie findet sich in Kapitel 5.2.1.

3.2.11 Feederbank

Streifen von Komponenten werden von den Herstellern auf Rollen geliefert. Der Feederbank ist eine Halterung für diese Rollen. Der Streifen kann in den Feeder eingeführt werden, und wird beim Vorwärtsschritt von der Rolle abgewickelt. In Abbildung 9 ist die Dimensionierung einer Rolle ersichtlich. Die Rollen haben standardisierte Masse. Die Breite der Rolle hängt von der Breite des Streifens ab und ist ca. 6 mm breiter als der Streifen. Gängige Masse für die Durchmesser der Rollen sind:

- Aussendurchmesser (A): 178 / 330 mm
- Innendurchmesser (C): 13 mm



Note: Drive spokes optional; if used, dimensions B and D shall apply.

Abbildung 9 Rollen Dimensionen (KEMET, 2016)

An der HSLU sind die Komponenten grösstenteils auf Rollen mit einem Aussendurchmesser von 178 mm vorhanden. Der Feederbank wird für solche Rollen ausgelegt.

Die Rollen sollen auf einem Stab gesetzt werden, welche eine freie Rotation ermöglicht. Das Auswechseln der Rollen soll benutzerfreundlich sein. Es soll eine einfache, funktionserfüllende Halterung erstellt werden. In Kapitel 5.1.8 wird die Entwicklung der Feederbank beschrieben.

4 Gesamtlösung

Wie in Kapitel 2.2 aufgezeigt, kann das Konzept in verschiedene Funktionen unterteilt werden. Dabei ist neben den mechanischen und elektrischen Funktionen auch eine Einbindung dieser in der Software nötig. Das Konzept beinhaltet neben der Realisation des Feeders auch die Erstellung der Feederbank für die Rollen sowie den Master für die Kommunikation.

In diesem Kapitel werden zuerst die entwickelten Teilkonzepte dargestellt. In Kapitel 5 wird die Entwicklung des Konzepts aufgeführt. In Kapitel 6 werden weitere Prototypen und Versuche aufgezeigt, welche zur Validierung des erstellten Konzepts dienen.

4.1 Gesamtkonzept

In diesem Unterkapitel wird das Gesamtkonzept in Kürze vorgestellt. Damit soll dem Leser die nachfolgenden Entwicklungen zum Endkonzept verständlicher gemacht werden.

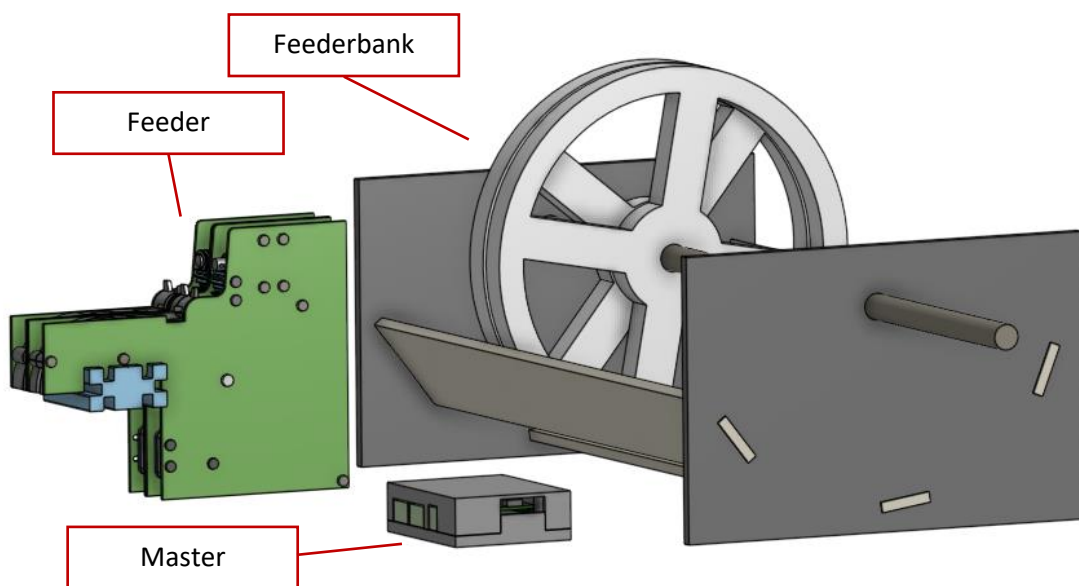


Abbildung 10 Darstellung Gesamtkonzept

Das Gesamtkonzept besteht aus dem Master, den Feedern und der Feederbank. Zusammen werden diese drei Objekte an die PnP Maschine von Prof. Styger angeschlossen. In Abbildung 10 sind diese drei Objekte in CAD zusammen dargestellt. In den nachfolgenden Unterkapiteln werden sie ausführlicher beschrieben. Die Feeder werden an der PnP Maschine auf einem Aluminiumprofil montiert. Dieses ist in der Darstellung als blaues Objekt erkennbar. Damit die Feeder ihre Position zueinander nicht verändern, werden sie mit 3D gedruckten Clips aneinander geklemmt. In Abbildung 11 ist ein Bild, worin die Feeder an die PnP Maschine montiert und mit dem Master verbunden sind.

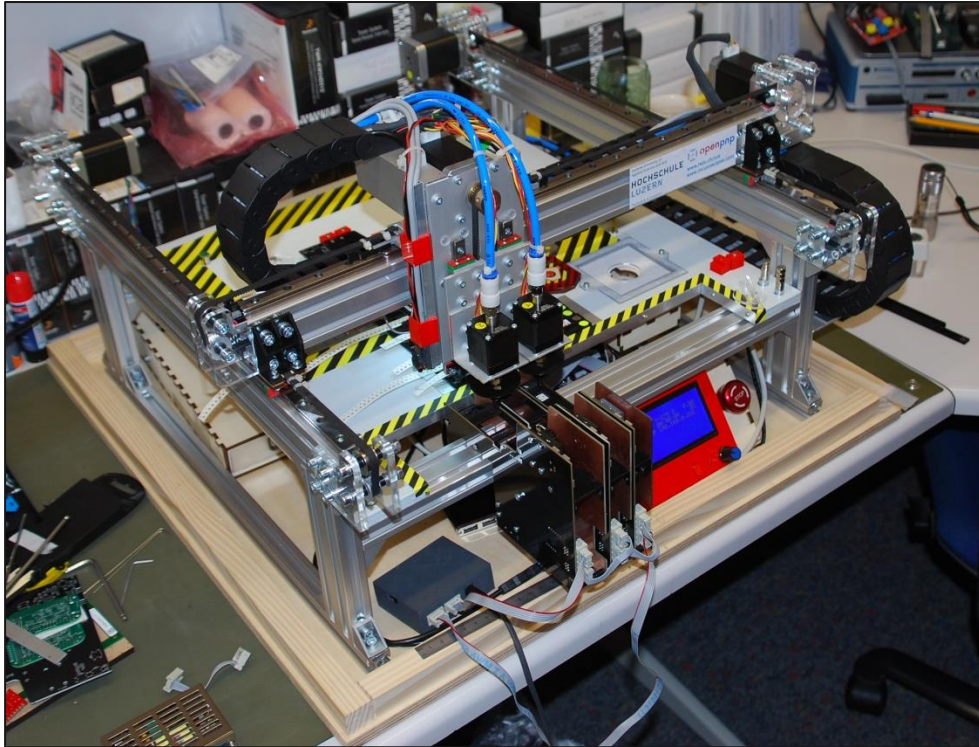


Abbildung 11 PnP Maschine mit montierten Feeder und Master

4.2 Feeder

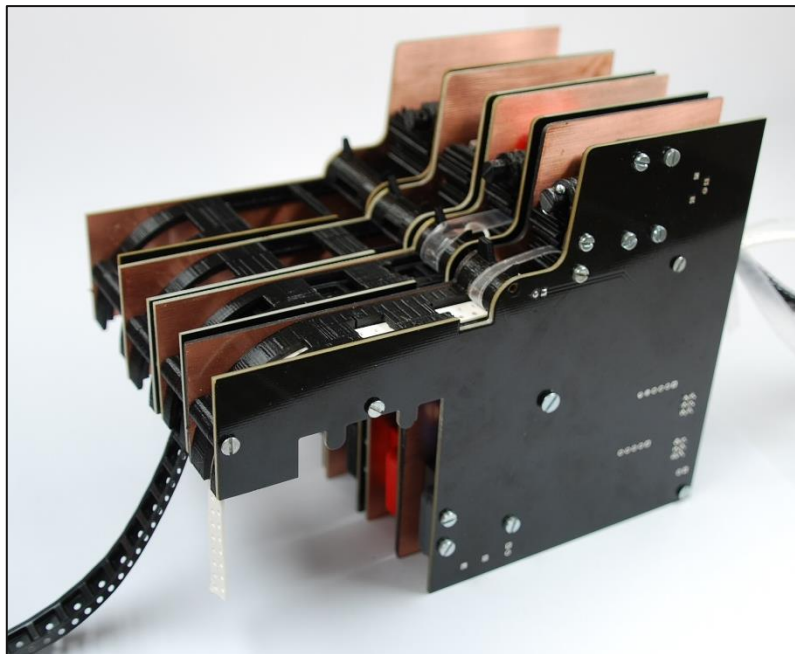


Abbildung 12 Feeder für die Streifenbreiten (von hinten nach vorn) 24, 16, 12, 8 mm

Im Rahmen dieses Projekts wurden Feeder für die Streifenbreiten von 8, 12, 16 und 24 mm erstellt. Diese sind in Abbildung 12 ersichtlich. Dabei wurde ein eigener mechanischer und elektrischer Aufbau erstellt, und diese funktionell mit Software verbunden. In Abbildung 13 werden die Teilfunktionen des Feeders aufgezeigt.

Zusammenfassend hat der Feeder folgende Eigenschaften:

Tabelle 7 Eigenschaften Feeder

Dimension	152 x 125 mm x (20.2 bis 31.7) mm
Energieversorgung	5 VDC (max. 600 mA)
Geschwindigkeit	8 mm/s
Kommunikation	UART Daisy-Chain
Mikrokontroller	MK20DX128VFT5 von NXP
Antriebsmotoren	N20 DC-Getriebemotoren 150:1
Schritterkennung	Lichtschränke EE-SX1350 Omron
Peelermechanismus, Sprocketantrieb, manuelle Ansteuerung, Status-LED	

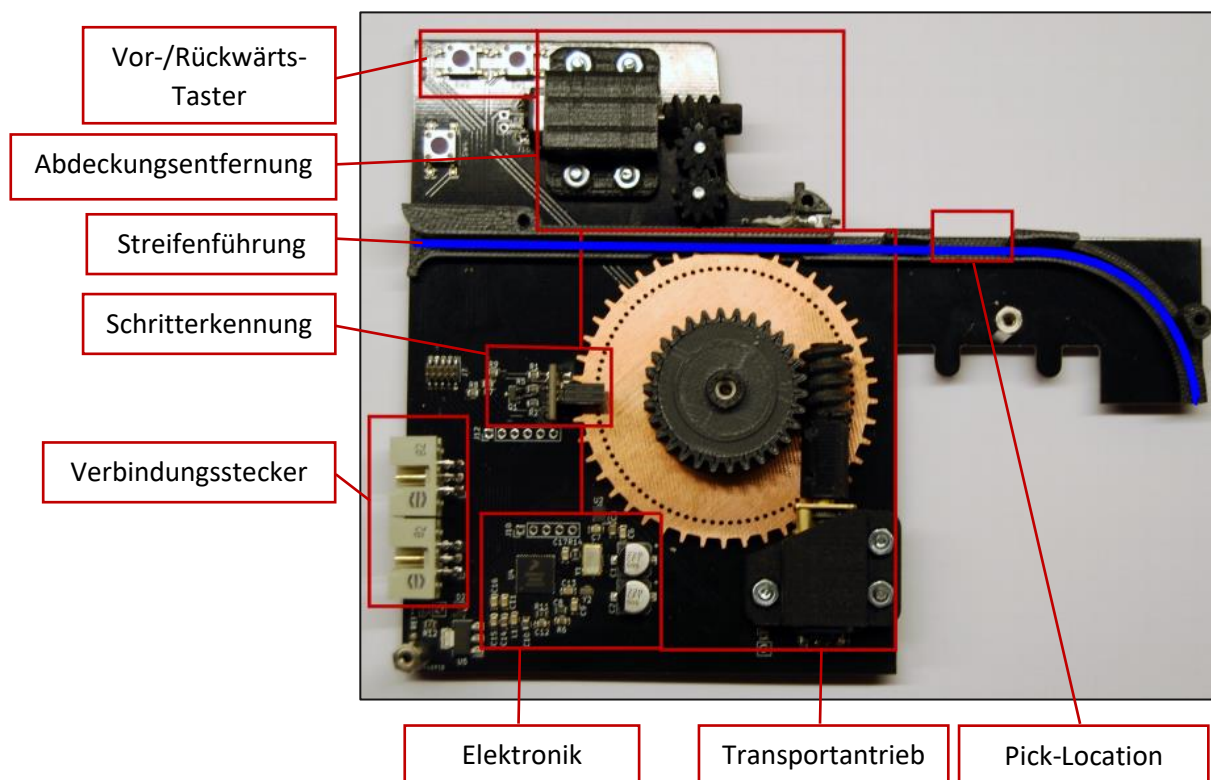


Abbildung 13 offener Feeder mit eingezeichneten Teilfunktionen

Vor-/Rückwärtstaster

Mit den Vor- und Rückwärtstaster kann manuell ein Schritt in die gewünschte Richtung ausgeführt werden. Diese Funktion ist insbesondere bei der Einrichtung eines Feeders vorteilhaft. Der Ablauf eines Schrittes in der Software wird in Kapitel 5.3 beschrieben.

Abdeckungsentfernung

Die Abdeckungsentfernung (Peeler) trennt die Abdeckungsfolie vom Streifen und ermöglicht es, dass die Bauteile aus dem Streifen entnommen werden können. Die Abdeckungsfolie wird zwischen den zwei Zahnrädern eingeklemmt und so abgezogen. Erreicht die Folie einen Mikroschalter, wird der Vorgang angehalten, bis wieder ein Schritt ausgeführt wird. Beim einem Rückwärtsschritt wird zuerst

die Folie zurückgeführt, und dann der Transport des gesamten Streifens ausgeführt. Die abgedeckte Folie wird frei hinten aus dem Feeder geführt. Die Abdeckungsentfernung wird mit einem DC-Getriebemotor angetrieben. Die gesamte Funktionalität wird in 5.1.5 beschrieben.

Streifenführung

Der Streifen wird in einer Führung, welche im 3D Druckverfahren hergestellt wurde, durch den Feeder gelenkt. Die Breite der Führung hängt von der Breite des Streifens ab. So gibt es für die Feeder von 8 bis 24 mm eine spezifische Führung. Alle haben dieselbe Grundform, unterscheiden sich jedoch in der Grösse der Pick-Location und der Breite. Die Entwicklung der Führung und die dazu gemachten Überlegungen werden in Kapitel 5.1.2 beschrieben.

Schritterkennung

Die Schritterkennung wird mithilfe einer Lichtschranke realisiert. Im Sprocket, welches den Streifen antreibt, sind Bohrungen welche von der Lichtschranke erkannt werden. In der Software des Mikrokontrollers werden die Schritte (Bohrungen) gezählt und entsprechend beim Erreichen des eingestellten Weges der Antrieb ausgeschaltet. Die Validierung der Schritterkennung durch Versuche wird in Kapitel 6.4 beschrieben.

Verbindungsstecker

Die Verbindungsstecker ermöglichen es die Feeder untereinander, respektive auch mit dem Master zu verbinden. Die Verbindung der Feeder ist in einem Ring aufgebaut. Dieser Daisy-Chain ermöglicht es mit einem einfachen Kommunikationsaufbau über UART ohne grösseren Einrichtungsaufwand die Feeder via Master mit OpenPnP anzusteuern.

Über dasselbe Verbindungskabel wird auch die Energieversorgung von 5 VDC der Feeder realisiert. Die Feeder-Verbindung wird ausgiebig in Kapitel 5.2.5 behandelt und ist in Abbildung 44 auf Seite 44 dargestellt.

Elektronik

Die Elektronik beinhaltet neben einem MK20DX128VFT Mikrokontroller von NXP auch eine H-Brücke DRV8835 zur Ansteuerung der Motoren. Die Spannungsversorgung der Logik mit 3.3 VDC wird auf jedem Feeder mit einem Spannungsregler vom Typ TC1262 realisiert. Die Versorgungsspannung der Feeder und der Motoren ist 5 VDC, welches über das Verbindungskabel geliefert wird.

Als Teil des elektrotechnischen Konzeptes wurde zudem eine Status-LED eingesetzt, welche es ermöglicht den Zustand des Feeders visuell darzustellen. Somit können Error-Meldungen oder Schritte des Transports einfach dargestellt werden.

Jeder Feeder kann über einen SWD-Header programmiert werden. Zudem sind Pads vorgesehen, mit welchen die Signale auf dem Feeder überprüft und gemessen werden können. So wird die allfällige Fehlersuche vereinfacht.

Die Elektronik-Konzepte samt Elektroschema werden in Kapitel 5.2 ausführlich beschrieben.

Transportantrieb

Der Transport des Streifens wird mithilfe eines DC-Getriebemotors und einem Sprocket aus PCB Material realisiert. Die Zähne des Sprockets greifen in die Löcher am Streifen und können den Streifen somit, wie eine Kette, bewegen. Das Schneckengetriebe wurde im 3D Druckverfahren hergestellt und

ermöglicht einen Vorschub von 8mm/s. Die Entwicklung des Transportantriebs wird in Kapitel 5.1.1 und 5.1.4 beschrieben.

Pick-Location

An der Pick-Location können die Bauteile vom Streifen entnommen werden. Der Bestückungskopf der Pick and Place Maschine wird an diese Stelle geführt, wo die Bauteile abgedeckt im Streifen liegen. Nach dem Abholen der Komponente kann ein neuer Schritt ausgeführt werden, und somit ein neues Bauteil bereitgestellt werden. Die Dimensionierung der Pick-Location ist für jede Streifenbreite spezifisch. Die breiteren Streifen können grössere Bauteile beinhalten, entsprechend wurde auch mehr Platz für die Entnahme bereitgestellt. Die Führung an der Pick-Location wurde so erstellt, dass die Abholhöhe sich über alle Streifenarten (Papier und Kunststoff) um nur 0.6 mm ändert. Dies vereinfacht die Einrichtung der Maschine. Ausführliche Informationen zur Pick-Location und Führung sind in Kapitel 5.1.2 ersichtlich.

4.3 Master

Der Master stellt die Schnittstelle zwischen dem OpenPnP-System und den Feedern dar. In der Systemübersicht in Kapitel 2 ist dargestellt, dass das OpenPnP über USB mit dem Master kommuniziert. Der Mikrokontroller im Master kommuniziert über UART mit den Feedern. Die Befehle und Rückmeldungen werden so zwischen allen Geräten übermittelt.

Der Master besteht aus einem tinyK22, einem Master-PCB und einem Gehäuse, welches die Elektronik schützt. Die Software auf dem tinyK22 wurde von Prof. Styger entwickelt. Es wurde ein PCB entwickelt, welches die Energieversorgung und Kommunikation ermöglicht.

Tabelle 8 Eigenschaften Master

Dimension	73.5 x 59.6 x 23 mm
Energieversorgung	5 VDC
Kommunikation Feeder	UART Daisy-Chain
Kommunikation OpenPnP	USB
Mikrokontroller	tinyK22

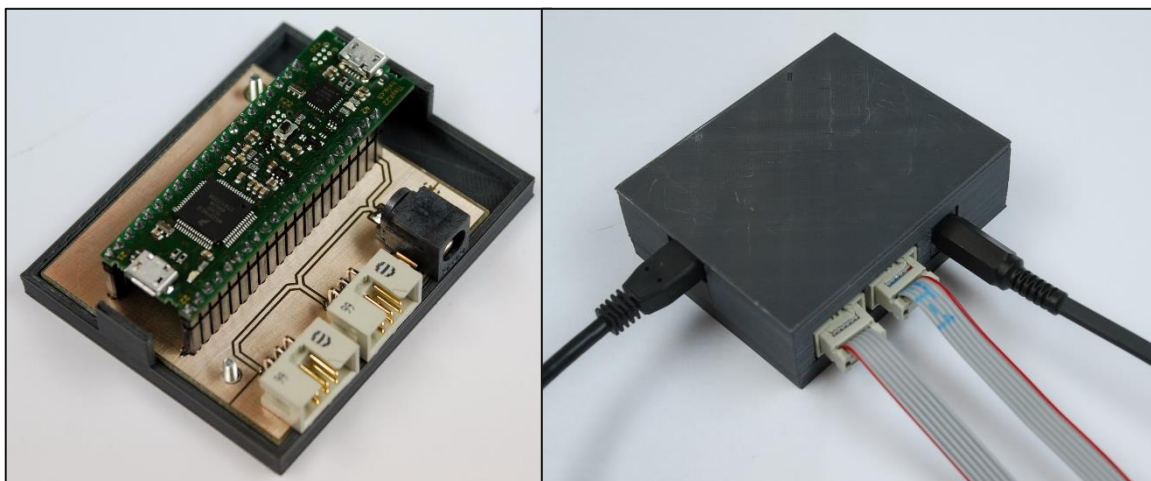


Abbildung 14 Master

In Abbildung 14 ist der Master ersichtlich. Die USB Verbindung geht direkt auf das tinyK22 (schwarzes Kabel links). Die Daisy-Chain UART Verbindung wurde über zwei Flachbandkabel (Rx/Tx) zu den

Feedern realisiert. Mit diesen Kabeln wird auch die Energieversorgung der Feeder umgesetzt. Das schwarze Kabel rechts ist die 5 VDC Spannungsversorgung vom Netzteil für Master und Feeder.

Das Gehäuse wurde im 3D Druckverfahren hergestellt, und kann mit zwei Schrauben zusammengesetzt werden.

Der Entwicklungsprozess vom Master-PCB wird in Kapitel 5.2.9 beschrieben, derjenige vom Gehäuse in Kapitel 5.1.7.

4.4 Feederbank

Der Feederbank ist ein einfaches Konstrukt aus Holz, auf welchem die Rollen montiert werden können. Der Feederbank ist in Abbildung 15 ersichtlich. Die Streifenrollen werden auf dem runden Holzstab gesetzt und werden durch die Vorwärtsschritte des Feeders abgerollt. Um die Rollen auszuwechseln kann der Stab seitlich aus der Feederbank herausgezogen werden. Die Rollen stehen dann auf den Wänden und dem Boden der Feederbank. Die gewünschte Rolle kann ausgetauscht werden und der Stab wieder hineingefahren werden. Durch die kurze Distanz von 1 mm zwischen Rollen und Wände verschieben die Rolle ihre Position kaum. Somit kann der Stab mühelos wieder durch alle Rollen gefahren werden. Der Feederbank ist für Rollendurchmesser von maximal 180 mm ausgelegt. Die Mehrheit an der HSLU eingesetzten Komponenten werden auf solchen Rollen geliefert. Die Umsetzung der Feederbank wird in Kapitel 5.1.8 beschrieben.

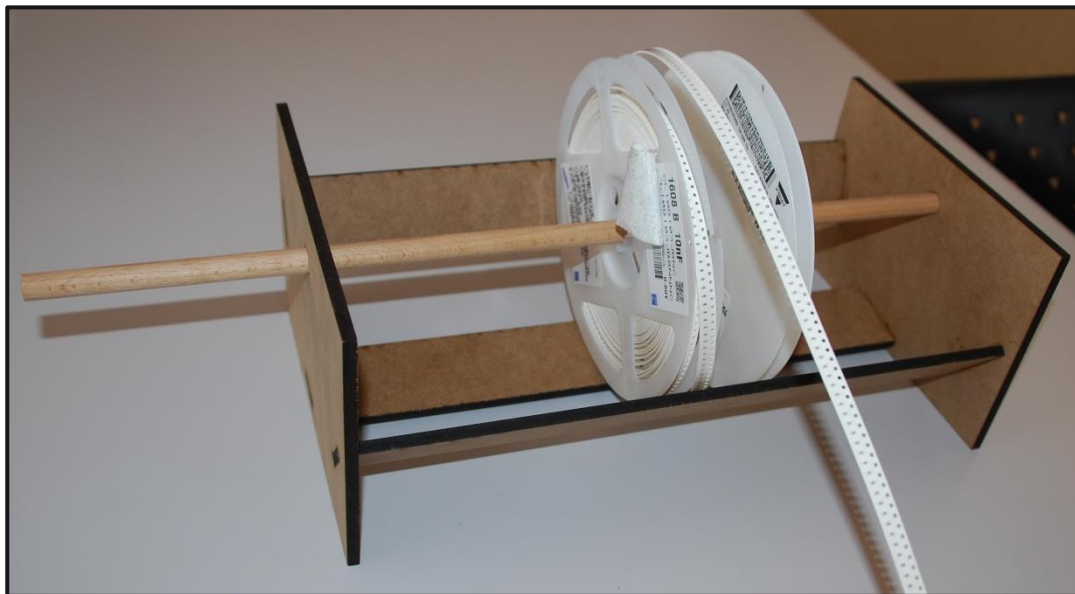


Abbildung 15 Feederbank mit zwei 8mm Streifenrollen

Tabelle 9 Eigenschaften Feederbank

Dimension	362 (500) x 204 x 135 mm
Anzahl 8mm Rollen	~25 Rollen

5 Entwicklung Lösungsvarianten

Wie in Kapitel 3 beschrieben, kann das Konzept in einzelne Teilfunktionen unterteilt werden. Die Entwicklung der Lösung um diese Teilfunktionen wird in diesem Kapitel aufgeführt. Dabei wurden die Teilaufgaben in drei Hauptgruppen unterteilt. Es folgen Unterkapitel zur mechanischen, elektrischen und Software Entwicklung.

Als mechanische Teilaufgaben gelten:

- Gehäuse des Feeders
- Führung des Bandes
- Sprocket
- Halter für Motoren
- Getriebe
- Halter für Rollen (Feederbank)
- Master-Gehäuse

Im Rahmen der elektrotechnischen Umsetzung muss ein PCB für den Feeder, sowie eines für den Master erstellt werden. Das Feeder-PCB muss folgende Funktionen erfüllen:

- Energieversorgung
- H-Brücke für Motorenantrieb
- Mikrokontroller
- Lichtschranke
- Kommunikation
- Taster
- Statusanzeige

Das Master-PCB muss zwei Funktionen umsetzen, nämlich:

- Schnittstelle OpenPnp – Feeder
- Energieversorgung der Feeder

5.1 Mechanisches Konzept

An der Hochschule für Technik und Architektur stehen Lasercutter, 3D Drucker und andere Maschinen zur Verfügung. Entsprechend wurde versucht ein Feeder zu erstellen, welcher mit diesen Mitteln hergestellt werden könnte.

In mehreren Iterationen wurde die Hardware erstellt, getestet und Anpassungen vorgenommen. In diesem Kapitel wird die Entwicklung der einzelnen Komponenten komplett aufgezeigt. Dabei wird zu jedem Komponenten die Überlegungen, die Entwürfe, Prototypen und endgültige Design vorgestellt. Das zusammengesetzte Konzept ist in Kapitel 4 beschrieben.

Für das Zeichnen der mechanischen Komponenten wurde das Online-Zeichnungsprogramm Onshape verwendet. Mit einem kostenlosen Konto können online 3D Zeichnungen erstellt werden, welche auch der Öffentlichkeit zugänglich sind. Zudem ist es, durch die Cloudbasierte Dienstleistung, möglich ohne weiteren Aufwand von verschiedenen Orten aus mit diesem Programm zu arbeiten.

5.1.1 Sprocket

Als eine der grössten Herausforderungen der mechanischen Teile wurde das Sprocket eingeschätzt. Obwohl es möglich ist ein Sprocket von industriellen Feedern als Ersatzteil zu bestellen, stand die Herstellung eines neuen Sprockets im Vordergrund. Der Grundaufbau des Sprockets ist einfach, denn die Zähne müssen nur im selben Abstand wie die Löcher im Streifen sein.

OpenPnP Design

Während der Recherche zum Antrieb (Kapitel 3.2.1) wurden bereits eine Sprocket-Design von einem Feeder für OpenPnP entdeckt (McKay, 2016). Dieses Design wurde abgeändert und aus Acrylglas gefräst. Der Prototyp ist in Abbildung 18 oben rechts ersichtlich. Dabei ist zu erkennen, dass die Zähne des Sprockets kurz und spitz sind. Es wurde erkannt, dass dies ein Nachteil ist, denn die Löcher des Streifens werden kaum von den Zähnen gepackt. Daraus entsteht die Gefahr, dass ein Schrittverlust passiert.

Eigenes Design

Nach weiterer Recherche wurde entschieden, das gesamte Sprocketdesign selbst zu erstellen. Der Durchmesser wird von dem Sprocket wird durch die Anzahl Zähnen und den Abstand zwischen diesen definiert. Der Lochabstand in den Streifen ist 4 mm. Der normierte Durchmesser der Löcher im Streifen

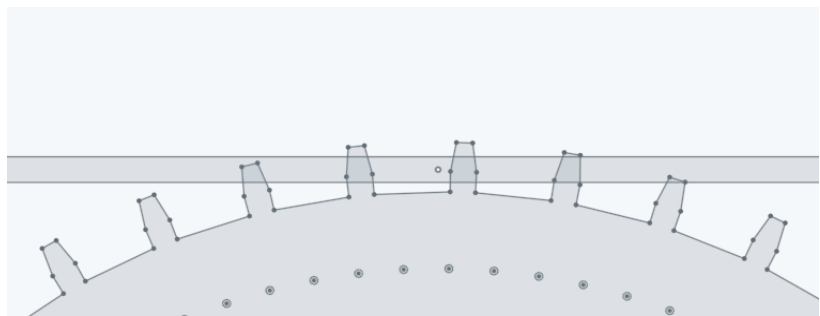
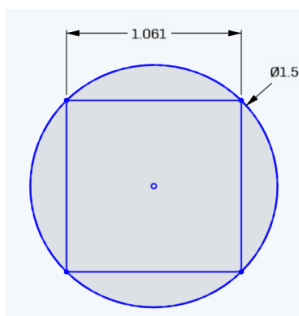


Abbildung 17 Darstellung Streifenloch Abbildung 16 Verzahnung Sprocket mit Streifen

ist 1.5 mm. Abbildung 17 zeigt auf, dass die Länge und Breite des Zahns maximal 1 mm sein darf, da der Zahn sonst nicht in das Loch hineinpasst. In mehreren Versuchen wurden verschiedene Durchmesser des Sprockets getestet. Um den Feeder kompakt zu halten, müsste auch das Sprocket einen kleinen Durchmesser haben. Da der Streifen tangential zum Sprocket läuft ist für eine saubere Führung wichtig, dass immer mehrere Zähne gleichzeitig mit dem Streifen in Berührung sind (Abbildung 16). Entsprechend resultiert dieses wieder in einem grösseren Durchmesser.

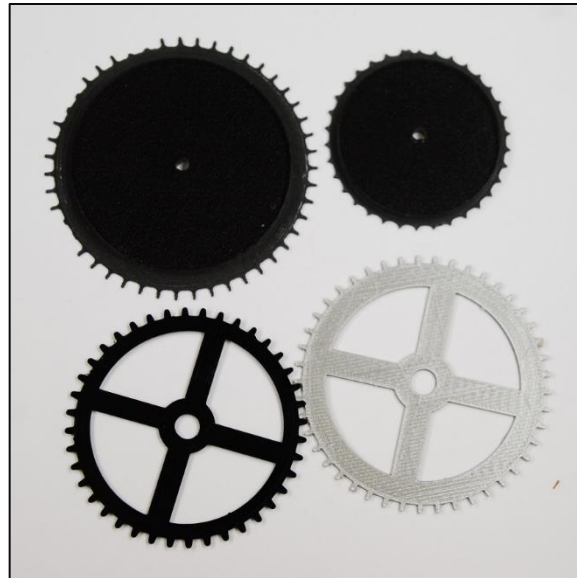


Abbildung 18 Sprocket Prototypen aus Acrylglas (oben) und 3D Druck (unten)

In Onshape wurden verschiedene Varianten von Durchmesser und Zahnbreite gezeichnet und hergestellt. Die ersten Varianten wurden versuchsweise aus Acrylglas gefräst oder im 3D-Druck Verfahren hergestellt. Abbildung 18 zeigt einige Prototypen von Sprockets. Diese Versuche liessen einige Fazits zu. Beim Herstellen der Sprockets aus Acrylglas wurden oftmals während dem Fräsvorgang einige Zähne abgebrochen. Dies ist auch in obiger Abbildung am Sprocket oben links erkennbar. Die Sprockets, welche 3D gedruckt wurden, waren in der Dicke von 1 mm nicht stabil genug um als Antrieb genutzt zu werden. In einer Projektsitzung wurde diskutiert PCB Material zu verwenden, da dieses schmal, stabil und sehr genau an der Hochschule selbst gefräst werden kann. Zudem gibt es PCB-Platten in der Breite von 1 mm. Die Beschreibung der Sprocket-Prototypen aus PCB folgt.

Tape Sprocket Creator

Das Zeichnen von Zahnradern ist komplex. Wobei der Zahnabstand noch relativ einfach zu zeichnen ist, muss der Zahn auch eine Form haben, welche ohne anzuecken in die Löcher des Streifens hinein und hinausgeht. Es war nach den ersten Prototypen klar, dass nur eine Vorlagenzeichnung oder externe Hilfe zu einem guten Resultat führen würden. Weitere Recherchen im Internet führten zu einer interessanten Lösung. Von einem gleichgesinnten auf der Suche nach einem Sprocket-Design wurde

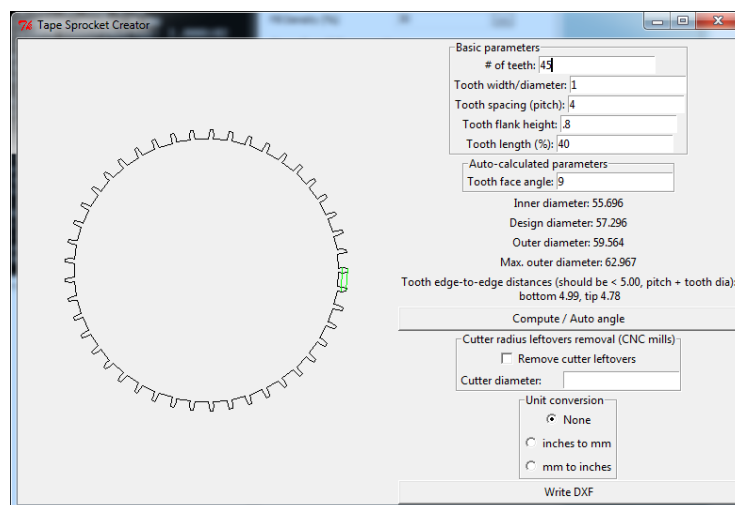


Abbildung 19 Programmfenster Tape Sprocket Creator (Gipson, 2011)

ein Python-Programm erstellt, mit dem Sprockets gezeichnet werden können. Mithilfe des Programms war es möglich weitere Prototypen zu erstellen welche den Streifen sauber führen konnten. Das Python-Skript ist im elektronischen Anhang im Ordner Sonstiges abgelegt (Gipson, 2011).



Abbildung 20 Sprocket Prototypen aus 1mm PCB Material

Abbildung 19 und Abbildung 20 zeigen das Programmfenster von Tape Sprocket Creator sowie mehrere Prototypen, welche aus PCB gefräst wurden. Die Designs für diese Prototypen stammten aus dem Tape Sprocket Creator.

Lösungsvariante

Im Kapitel 3.2.3 wurde entschieden, am Sprocket Bohrungen für die Schritterkennung zu platzieren. Der Durchmesser dieser Bohrungen beeinflusst die Lichtmenge, welche vom Fototransistor der Lichtschranke empfangen wird, und somit auch das erzeugte Signal. In Kapitel 6.4 wird auf dieses Thema eingegangen. Im eingesetzten Design des Sprockets sind 90 Bohrungen platziert, welche eine Schritterkennung im 2 mm Abstand erlaubt. Der Durchmesser der Bohrungen ist 0.7 mm. Das verwendete Sprocket ist in Abbildung 21 als Design und in vorheriger Abbildung 20 rechts als Prototyp dargestellt. Es hat einen Aussendurchmesser von 61 mm

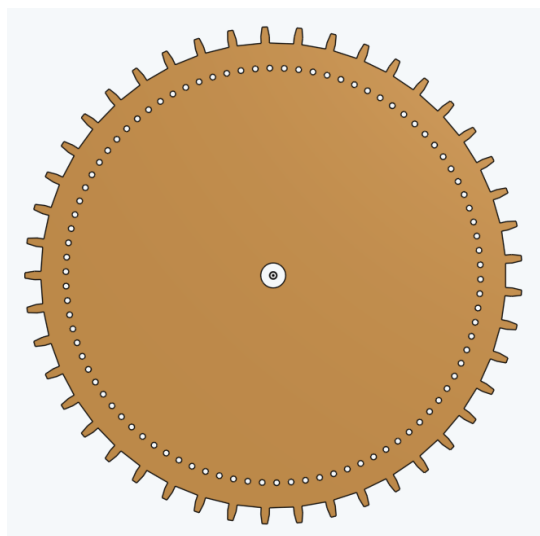


Abbildung 21 Verwendetes Design des Sprockets

5.1.2 Führung

In diesem Unterkapitel wird zuerst der funktionelle Aufbau der Führung beschrieben. In zweiten Teil werden verschiedene Prototypen kurz vorgestellt.

Beschreibung Aufbau

Für die Führung des Streifens wurden mehrere Prototypen im 3D Druckverfahren hergestellt. Die Grundform wurde bereits beim ersten Prototypen gesetzt, und in weiteren Phasen Anpassungen in Bezug auf die Montage und Führung gemacht. Mithilfe der 3D Darstellung in Abbildung 22 wird der Aufbau der Führung erläutert. Darauf folgend werden weitere Informationen gegeben, womit die Erstellung des hier vorliegenden Designs erklären. Die Führung besteht aus zwei Hälften. An dieser Stelle wird nur eine Seite erklärt, an welcher der Aufbau gut ersichtlich ist. Im Kapitel 4.2 gibt es eine Darstellung des kompletten Feeders.

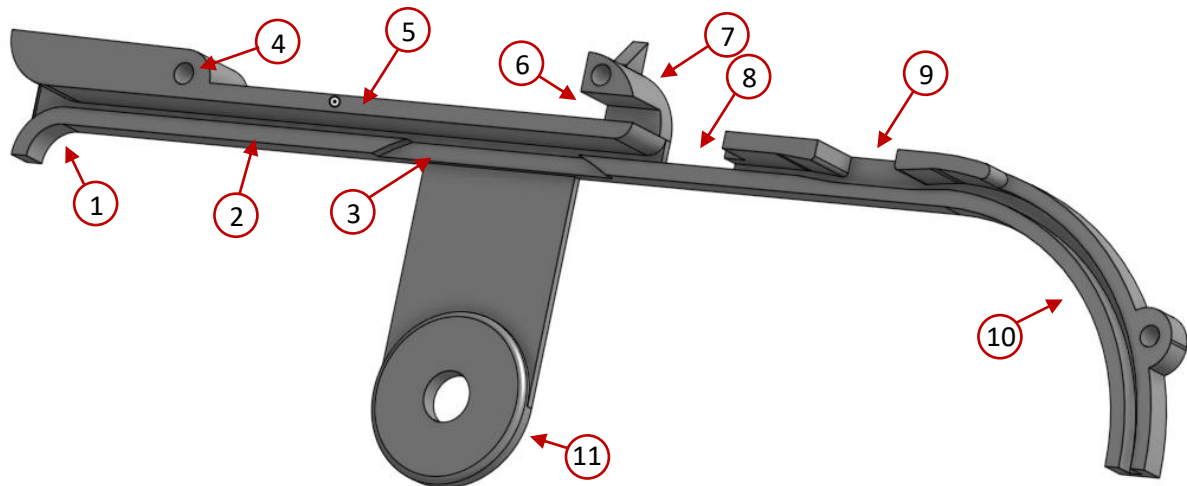


Abbildung 22 3D Darstellung der Führung

1. Verbreiterte Öffnung für die Einführung des Streifens
2. Untere Auflage für die Führung des Streifens
3. Ausschnitt für das Sprocket
4. Montageloch
5. Obere Abdeckung
6. Aussparung für den Mikroswitch vom Peeler
7. Führung der Abdeckungsfolie
8. Aussparung für die Abdeckungsfolie
9. Aussparung für die Entnahme des Bauteils
10. Umlenkung für den leeren Streifen aus dem Feeder
11. Abstandshalter für die Montage des Sprockets

Bei Punkt 1 wurde für die vereinfachte Einführung des Streifens die Öffnung verbreitert und abgerundet. Die Führung des Streifens geschieht auf der gesamten Länge zwischen Punkt 2 und 5, wobei die Distanz zwischen diesen 1.2 mm beträgt. Die Streifen aus Papier haben eine maximale Dicke von 1.1 mm, Streifen aus Kunststoff haben eine maximale Dicke von 0.6 mm (KEMET, 2016). Weitere Punkte der Entwicklung des Designs werden mit zusätzlicher Unterstützung von Abbildung 24 erklärt. Das komplette Datenblatt dieses Ausschnitts in der Abbildung ist im elektronischen Anhang im Ordner Datenblätter abgelegt.

Die untere Auflagefläche der Führung ist an der linken Hälfte der Führung 2.5 mm breit. Derjenige der rechten, je nach Führungsbreite, von 0.9 mm bis 2 mm breit. Die unterschiedliche Breiten lassen sich durch den eingezeichneten Punkt H sowie B1 in Abbildung 24 erklären. B1 wird durch die Komponentenbreite definiert. Dieses Mass hat bei unterschiedlichen Streifenbreiten und Komponenten einen Einfluss auf die mit H markierten Auflageflächen. Die mögliche Auflagefläche rechts wurden bei jeder Streifenbreite berechnet und ist in das Design eingeflossen. Die Auflagefläche links wurde konstant bei 2.5 mm gehalten, da diese Breite eine genügende Auflagefläche bereitstellt und nicht mit dem Mass B1 in Konflikt kommen kann. Abbildung 23 soll dieses Zusammenspiel von Streifen und den Punkten 2 und 5 verdeutlichen.

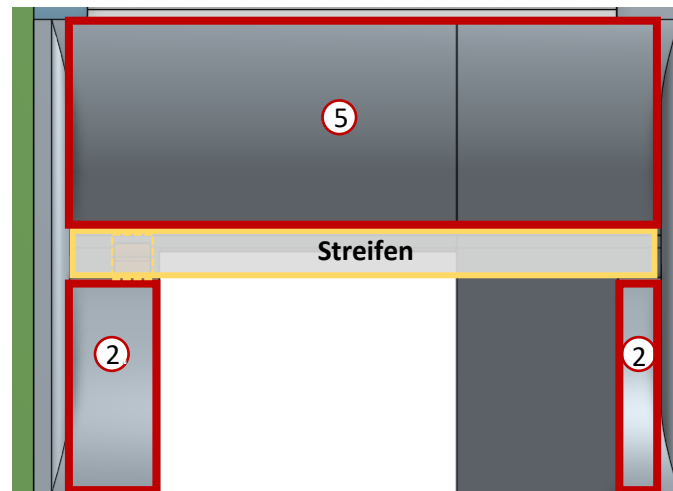


Abbildung 23 Darstellung Streifen-Auflageflächen

Bei Punkt 8 wird die Abdeckungsfolie vom Streifen herausgeführt. Die Folie wird über Punkt 7 zurück in den Peeler geführt. Der Peeler wird in Kapitel 5.1.5 ausführlich beschrieben. Bei Punkt 8 wurde vorgesehen, dass ein Deckel montiert werden könnte. Schliesst man den Deckel, kann man verhindern das Komponenten an dieser Stelle aus dem Streifen gespickt werden. Auf die Entwicklung des Deckels wurde verzichtet, da das Herausspicken bei Testläufen nicht beobachtet wurde. Bei der Umlenkung bei Punkt 7 ist zudem ein dreiecksförmiger Nocken vorhanden. An beiden Führungen ist dieser Nocken vorhanden. Mit diesem Nocken wird die Abdeckungsfolie mittig gehalten und somit auch gewährleistet, dass die Abdeckungsfolie über den Mikroswitch geführt wird. Weitere Informationen zu diesem Thema gibt es im Kapitel 5.1.5 und 6.5.

8, 12, 16 AND 24 mm EMBOSSED TAPE DIMENSIONS

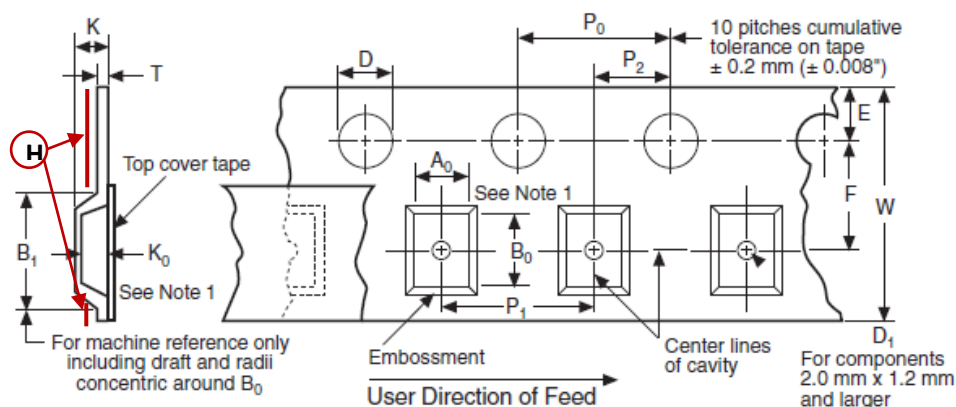


Abbildung 24 Dimensionen eines SMD Streifens (Vishay, 2017)

Bei Punkt 9 in Abbildung 22 wurde eine zusätzliche Umlenkung des Streifens vorgenommen. Dickere Streifen werden um bis zu 0,4 mm nach unten gelenkt. Dünnere Streifen werden an dieser Stelle nicht nach unten gedrückt und gehen wegen ihrer Stabilität waagrecht durch. Somit variiert die Abholhöhe des Bauteils bei verschiedenen Streifendicken nur um 0.6 mm. Um die Einrichtung der OpenPnP Maschine zu vereinfachen wird für alle Feederarten dieselbe Pick-Höhe eingestellt. Mit dieser Vereinheitlichung auf diesem automatisierten Feeder wird dieses Konzept beibehalten.

Der Biegeradius der Streifen ist normiert. Der maximale Biegeradius für Streifen bis 24 mm ist 30 mm (Vishay, 2017, S. 1). Entsprechend wurde dieses Mass in Punkt 10 eingehalten.

Der Abstand des Sprockets zur Wand wird mit Punkt 11 erreicht, und durch das Mass E definiert. Standardmässig ist E 1.75 mm.

Prototypen



Abbildung 25 Prototypen von Führungen aus 3D Druck

In Abbildung 25 sowie in Abbildung 59 von Kapitel 6.1 sind verschiedene Prototypen von Führungen ersichtlich. In der blauen Führung links dargestellt wurde die Möglichkeit getestet, die Führung und Wand als ein Teil zu drucken. Die Wandstärke von 1 mm war deutlich zu schwach um eine stabile Montage von Sprocket und Antrieb zu ermöglichen. Eine Erhöhung der Wandstärke würde deutlich mehr Druckmaterial, aber insbesondere auch mehr Zeit in Anspruch nehmen. Entsprechend wurde auf eine Weiterführung dieses Konzepts verzichtet. Der Prototyp rechts bereits die Grundform, welcher in der weiteren Entwicklung verwendet wurde. Die Grösse der Führung wurde durch die Seitenwände definiert, welche im nächsten Kapitel beschrieben werden.

5.1.3 Wand

Durch unterstützende Vorgaben von Prof. Styger und Recherchen von anderen für OpenPnP entwickelte Feeder (Grimstad, 2018), wurde entschieden PCB als Material für die Wände zu verwenden. PCB Material ist bei einer niedriger Dicke von 1.6 mm stabil, und bietet den entscheidenden Vorteil gleichzeitig auch als Leiterplatte für die Elektronik zu fungieren. Mit dieser Kombination kann die Breite des Feeders niedrig gehalten werden.



Abbildung 26 Prototypen der Wände aus PCB (v.l.n.r. Version 2, Version 1)

Die linke Wand (Wall_B) dient als Hauptmontage des mechanischen Aufbaus und als Leiterplatte der Elektronik. Die rechte Wand (Wall_F) schliesst den Aufbau wie ein Deckel und ermöglicht einen stabilen Aufbau. Die Grundform der Wand wurde durch den mechanischen Aufbau definiert. In mehreren Iterationen wurde die Grösse der Wände, die Platzierung der Montagelöcher und das Layout der Leiterplatte verändert. Die Wände des Feeders sind für alle Streifenbreiten dieselben. Abbildung 27 ist eine technische Skizze der Wände. Abbildung 26 enthält zwei Prototypen der linken Wand. Im umgesetzten Konzept wurde die Wand mit dem Layout mit Lötstopplack produziert, und diejenige ohne Layout als einseitiges PCB an der HSLU. Im Kapitel 4 gibt es eine Darstellung des kompletten Feeders, welcher dies verdeutlicht. Auf das Layout der Leiterplatte wird in Kapitel 3.2.8 eingegangen.

Aus Abbildung 27 sind einige Unterschiede zwischen den Wänden im endgültigen Design ersichtlich. In Wall_B (links) sind die Montagelöcher für die Motorhalterungen erkennbar. Zudem ist ein Ausschnitt an der Stelle vorhanden, wo die Abdeckung der Folie stattfindet. Dieser Ausschnitt erlaubt es gut an den Streifen heranzukommen, um beim Einrichten des Feeders die Folie vom Streifen zu trennen. In Wall_F (rechts) ist ein Ausschnitt links unten vorhanden. Auf Wall_B sind an dieser Stelle die Stecker als through-hole Komponenten montiert. Entsprechend hat es an der Stelle Lötstellen. Damit zwei Feeder geschlossen aneinanderlegen können, wurde der Ausschnitt auf der aufliegenden Wand geschaffen. Dieser Zusammenhang ist in Abbildung 44 ersichtlich.

Die Form der Wände wurde so gewählt, dass sie auf ein Aluminiumprofil 4020, aus welcher der Rahmen der PnP-Maschine gebaut wurde, montiert werden können. Bei beiden Wände wurde der vordere Teil (rechts) so gezeichnet, dass er direkt in die Schiene passt. Somit wird keine weitere Halterung benötigt um die Feeder an die PnP-Maschine zu montieren. Damit die Feeder satt aneinander liegen wurden 3D gedruckte Klemmen hergestellt. Mit diesen können die Feeder aneinander fix befestigt werden.

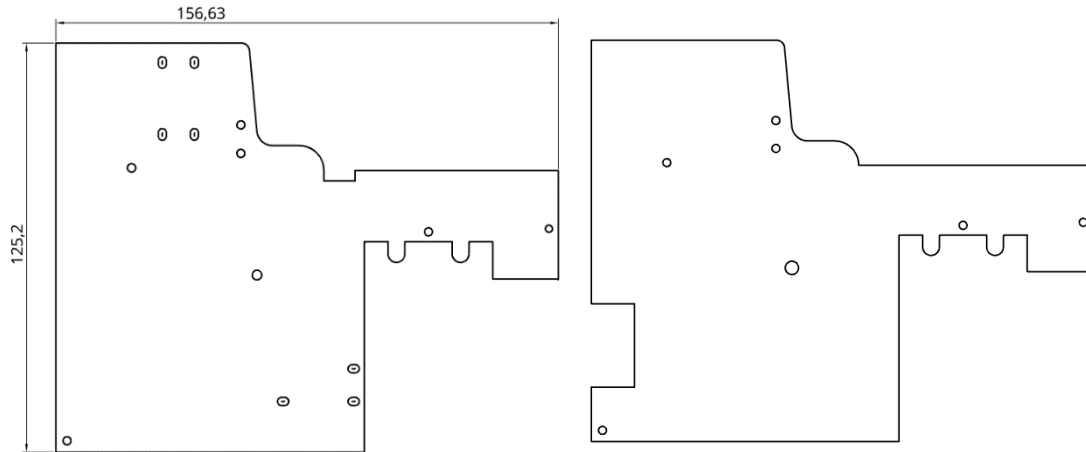


Abbildung 27 Technische Skizzen Wall_B (links), Wall_F (rechts)

5.1.4 Antrieb Sprocket

In Kapitel 1.3 wurde als Ziel für die Geschwindigkeit des Vorschubes 4 mm/s definiert. Der Sprocket besitzt einen Durchmesser von 61 mm. Folgend wird die minimale Drehzahl des Sprockets berechnet. Die maximale Drehzahl sollte jedoch nicht sehr hoch liegen, da durch das Beschleunigungs- und Bremsverhalten das Risiko grösser wird, dass die Bauteile aus dem Streifen springen könnten. Die Umdrehungsgeschwindigkeit nach der minimalen Drehzahl wird folgend berechnet.

$$U = d * \pi = 61 \text{ mm} * \pi = 191.63 \text{ mm}$$

$$n = \frac{v}{U} = \frac{4 \text{ mm}}{s * 191.63 \text{ mm}} = 0.02 \text{ s}^{-1} \approx 1.2 \text{ min}^{-1}$$

Die Drehzahl des Sprockets wird durch drei Stellen definiert. Als erstes lässt sich die Drehzahl eines DC-Motors durch eine Ansteuerung über PWM variabel einstellen. Als zweites ist an den für dieses Projekt verwendeten N20 DC-Motoren bereits ein Getriebe montiert. Dieses Getriebe gibt es mit verschiedenen Übersetzungen.

Die Montage der DC-Motoren im Feeder soll möglichst flach erfolgen, damit auch die Breite des Feeders niedrig bleibt. Die erstellten Motorenhalter werden im Kapitel 5.1.6 erklärt. Diese flache Montage hat zur Folge, dass die Übertragung von Motorenwelle zum Sprocket eine Richtungsänderung unterläuft. Dieses wird mit einem Schneckengetriebe erreicht, welche als dritter Punkt die Drehzahl beeinflusst.

Schneckengetriebe

Die Grösse der Zähne eines Zahnrads kann durch das Modul eines definiert werden. Die Berechnung des Moduls ist in folgender Formel dargelegt.

$$m = \frac{d}{z} \quad (m \dots \text{Modul}, d \dots \text{Durchmesser}, z \dots \text{Zähnezahl})$$

Bei den hier verwendeten Zahnrädern wurde das Modul 1 verwendet. Somit entspricht die Anzahl der Zähne dem Durchmesser. Diese Grösse des gewählten Moduls erlaubt es auch, die Zahnräder in stabiler Form im 3D-Druckverfahren herzustellen.

Um die gewünschte, niedrige Drehzahl zu erreichen sollte die Übersetzung des Schneckengetriebes hoch sein. Der Durchmesser des Zahnrades am Sprocket wurde jedoch auch durch mechanische Gegebenheiten definiert. In Abbildung 24 ist das Mass K ersichtlich, welches durch die Tiefe des Bauteils definiert wird. Recherchen haben ergeben, dass eine Bauteiltiefe von bis zu 12.5 mm möglich ist. Da das Zahnrad direkt unter dem Streifen montiert wird, wird der maximale Durchmesser des Zahnrades d_z durch dieses Mass beeinflusst. Somit folgt mit dem Durchmesser des Sprockets d_s :

$$d_z = d_s - K = 61 \text{ mm} - 2 * 12.5 \text{ mm} = 36 \text{ mm}$$

Es wurde entschieden ein Zahnrad mit dem Nutz-Durchmesser von 30 mm zu verwenden. Das Übertragungsverhältnis des Schneckengetriebes ist somit 30:1. In Abbildung 28 ist die CAD Darstellung des Schneckengetriebes ersichtlich. Für die Erstellung des CAD Aufbaus der Feeder wurden für die N20 Motoren Step-Files importiert. Diese wurden von Benjamin Galindo erstellt und sind auf GrabCAD erhältlich (Galindo, 2014).

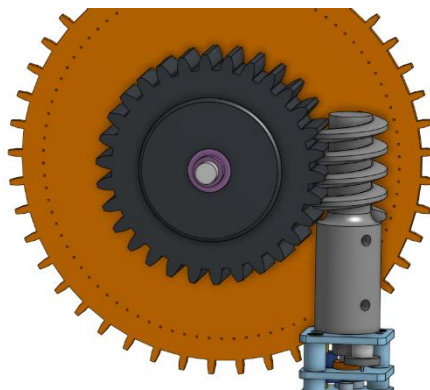


Abbildung 28 Darstellung des Sprocket-Getriebes

Drehzahl

Mit der Übersetzung des Schneckengetriebes kann die Drehzahl an der Antriebswelle des Getriebemotors berechnet werden.

$$n = 1.2 \text{ min}^{-1} * 30 = 36 \text{ min}^{-1}$$

Im Rahmen weiterer Projekte wie der Sumo-Roboter verwendet Prof. Styger bereits N20 Getriebemotoren. Für den Prototypenbau des Getriebes wurden solche Motoren verwendet, welche eine Übersetzung von 150:1 haben. Laut Datenblatt ist die Drehzahl bei maximaler Effizienz 170 min^{-1} . Die tatsächliche Drehzahl kann jedoch tiefer liegen. In Kapitel 6.1 wird bildlich ein erster Versuchsaufbau dargestellt. Als Fazit dieses Versuchsaufbaus konnte entschieden werden, dass die Motoren in dieser Form ein gutes Funktionsverhalten im Feeder aufweisen. Das Beschleunigungs- und Bremsverhalten bei dieser Geschwindigkeit birgt keine Gefahr, dass Komponenten aus dem Streifen springen.

Die Kennwerte des DC-Getriebemotors bei maximaler Effizienz sind laut Datenblatt wie folgt (Pololu, 2018a). Weitere Angaben zu den N20 DC-Getriebemotoren befinden sich im elektronischen Anhang im Ordner Datenblätter.

Tabelle 10 Kennwerte DC-Getriebemotor

Spannung	6 V
Strom	0.31
Übersetzungsverhältnis	150.58:1
Umdrehungsgeschwindigkeit	170 min ⁻¹

Bei dem endgültigen Konzept wurde die Stromaufnahme des Motors unter Belastung gemessen. Diese Resultate sind im Kapitel 6.7 präsentiert.

Design

Beim ersten Prototypen wurde ein Schneckenrad von Conrad gekauft (Conrad, 2018). Dabei konnte mit einem 3D-gedruckten Zahnrad der Antrieb des Sprockets getestet werden. Da die Form des eingekauften Schneckenrads nicht dem gewünschten entsprach und das 3D gedruckte Zahnrad genau gedruckt wurde, wurde entschieden beide Komponenten im 3D Druckverfahren herzustellen.

Das 3D Druckverfahren hat einen Nachteil, da überhängende Formen schlecht gedruckt werden können. Deshalb muss das Schneckenrad in zwei Hälften gedruckt werden, und nachträglich zu einem Stück zusammengeklebt werden. Die Hälften sind in Abbildung 29 ersichtlich.

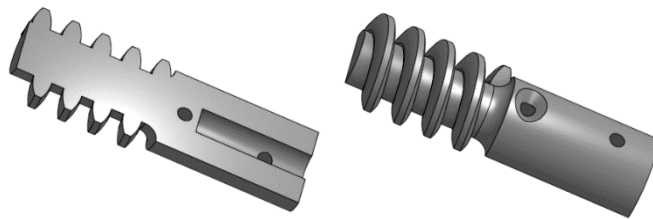


Abbildung 29 Hälften eines Schneckenrads

Da in der frühen Entwicklungsphase des Feeders unklar war, wie präzise der Antrieb angehalten werden könnte, wurden auch bei dem Entwickeln des Getriebes weitere Überlegungen gemacht. Bei Recherchen von Getriebearten wurde das Prinzip des Barrel-Cams gefunden (VertikalDesign, 2009), (Ravichandra, 2012). Dabei wird erfährt ein Teil des Schneckengetriebes keine Steigung. Somit wird bei der Rotationsbewegung des Schneckenrads das angetriebene Zahnrad angehalten. Die Überlegung folgte, dass damit ein Getriebe entwickelt werden könnte welches durch mechanische Auslegung immer eine Schrittweite bewirkte. Da jedoch das präzise Anhalten schwierig ist, könnte mit dem Barrel-Cam-Getriebe eine mechanische Unabhängigkeit eingebaut werden. Eine CAD Darstellung des entwickelten Barrel-Cam-Schneckenrads ist in Abbildung 30 ersichtlich. Das Barrel-Cam hätte jedoch zur Folge, dass die Übersetzung des Getriebes angepasst werden müsste, damit eine volle Umdrehung des Schneckenrads auch einen Schritt am Sprocket bewirkt. Ein Prototyp wurde erstellt und getestet. Es wurde jedoch erkannt, dass die mechanische und elektrische Kombination von Schneckengetriebe, grosse Übersetzungen und einer H-Brücke mit Bremsfunktion genügten um den Streifen wiederholt präzise anzuhaltten. Somit wurde entschieden das Getriebe in standardmässiger Form nach vorheriger Berechnung auszuführen.

Das eigen gezeichnete Schneckenrad wurde während der Entwicklung der Prototypen mehrmals verändert. Dabei wurden Änderungen an Grösse und Stabilität des Schneckenrads gemacht. Diese



Abbildung 30 Schneckenrad Prototypen (v. l. Barrel-Cam, kurze Version, endgültige Version)

Änderungen waren nötig, da die Position des Antriebsmotors verändert wurde und/oder Schwachstellen im Design erkannt wurden. Zwei weitere Designs sind auch in Abbildung 30 ersichtlich. Das im End-Konzept verwendete Schneckenrad hat eine Länge von 30 mm, welche durch den Aufbau des Feeders definiert wurde. Der Durchmesser beträgt 10.4 mm. Es sind zwei Bohrungen vorgesehen und in beiden wird ein Gewinde M2.5 geschnitten. Die erste Bohrung erlaubt es ein Gewindestift einzuschrauben, welcher auf die Motorwelle drückt. Somit wird eine schlupflose Kraftübertragung zwischen Motorwelle und Schneckenrad ermöglicht. Im dem zweiten Gewinde kann eine Schraube montiert werden, welche die zwei Hälften des 3D Drucks zusätzlich zusammenhält.

5.1.5 Peeler

Zahnräder

Der Peeler soll die Abdeckungsfolie von dem Streifen entfernen. In Kapitel 3.2.4 wurde entschieden, diese Folie zwischen zwei Zahnräder einzuklemmen und somit eine Zugbewegung ermöglichen. In Abbildung 31 wird der Peeler dargestellt. Als rote, gestrichelte Linie wird die Folie repräsentiert. Die Dimensionierung der Zahnräder wurde klein gehalten, um die Grösse des Feeders klein zu halten. Es wurde wiederum, wie bei Antriebsgetriebe in Kapitel 5.1.4, das Modul 1 verwendet. Die Zahnräder haben 10, respektive 11 Zähne. Die ungleiche Zahnung ermöglicht es, dass jeder Zahn gleich häufig mit den anderen kämmt. Es folgt somit eine gleichmässige Abnutzung. Alle mechanischen Komponenten wurden im 3D Druckverfahren hergestellt. Für die weiteren Prototypen wurden nur noch die Länge der Zahnräder der entsprechenden Feederbreite angepasst. Das Schneckenrad erfuhr noch eine weitere Entwicklung. Auch hier wurde eine Schraube vorgesehen, welche für die beiden Hälften des 3D Drucks zusammenhält.

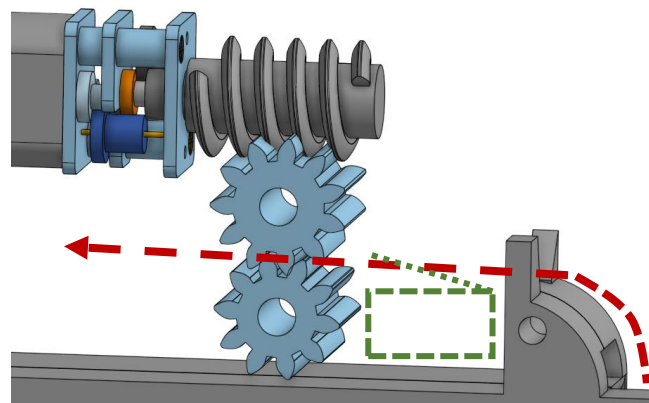


Abbildung 31 Darstellung des Peelers

Microswitch

Bei der Umlenkung der Folie wird in einer Aussparung der Führung ein Mikroswitch montiert. Wurde die Folie genügend weit abgezogen, liegt die Folie auf dem Mikroswitch auf und dieser wird betätigt. Dieses Signal wird verwendet um den Motor des Peelers auszuschalten. Die Umsetzung der Funktionalität in Software wird in Kapitel 5.3 erläutert.

Im ersten Prototyp wurde ein Mikroswitch mit Hebel verwendet. Er wurde an der Grün gekennzeichnete Stelle montiert. Die Annahme war, dass der Hebel genügend Kraft hätte, um die Folie vom Streifen zu lösen und entsprechend zu schalten. Im ersten Prototyp wurde ersichtlich, dass diese Funktionalität nicht gewährleistet werden könnte. Somit wurde ein alternativer Mikroswitch gesucht und eingesetzt. Der in Abbildung 32 ersichtlicher Mikroswitch wird in horizontaler Lage montiert und reagiert auch auf eine solche Bewegung. Für die Betätigung des Mikroswitches wird nur eine Kraft von 290 mN benötigt. Somit muss die Folie nicht sehr stark auf dem Mikroswitch aufliegen um diesen zu betätigen, und es wird weniger Zug auf den Streifen durch den Peeler verursacht. Die Abdeckungsentfernung wird als Gefahr angeschaut, denn durch diesen vertikalen Zug können durch ruckartige Bewegungen Komponenten aus dem Streifen geschüttelt werden. Durch das leichte Schalten des Mikroswitches wird die Folie möglichst wenig gespannt und ebenso dieses Risiko vermindert. Ein weiterer Prototypenaufbau hat die Funktionalität von diesem Aufbau des Peelers bestätigt. Eine ausführliche Darstellung der Prototypen um den Mikroswitch ist in Kapitel 6 zu finden.

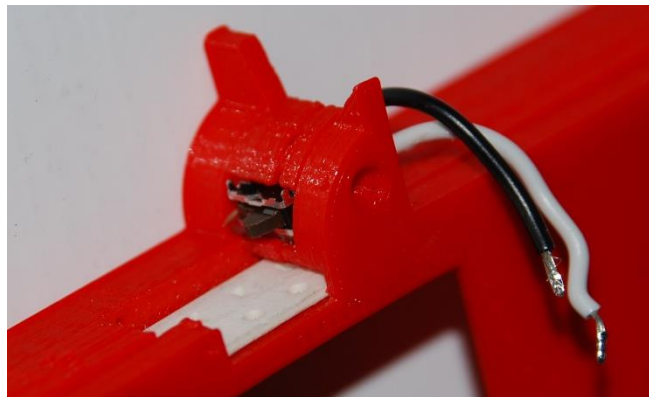


Abbildung 32 Peeler-Mikroswitch in Führung

5.1.6 Motorenhalter

Die Halter für die Motoren wurden im 3D Druckverfahren hergestellt. In ersten Prototypen wurde festgestellt, dass unter gewissen Umständen der Motor und Schneckenrad sich vom Antriebszahnrad wegdrücken konnten. Dieses Verhalten wird in Abbildung 34 dargestellt. Der Halter wurde vergrößert um den Motor mit Getriebe abzudecken. Zudem wurde die Wanddicke erhöht. Somit konnte eine bessere Stabilität erreicht werden. Um eine, bei allen Feedern, konstante Positionierung des Motors zu erreichen wurde im Halter zudem einen Anschlag für den Motor gegeben. Die Motorhalterversionen können in Abbildung 33 verglichen werden.

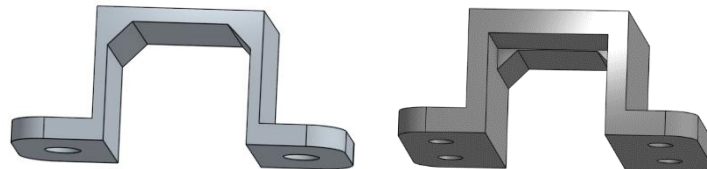


Abbildung 33 Motorhalter (v.l. Version 1, Version 2)

Da alle elektrischen Komponenten an der linken Wand montiert werden sollten, um einen einfachen Zusammenbau der Feeder zu ermöglichen, und um die in Abbildung 34 rot eingezeichnete Hebelwirkung zu vermindern, erhielt der Motorenhalter des Sprocket-Antriebmotors eine besondere Form. Das Schneckenrad wurde kurzgehalten und der Motor so möglichst nah am Zahnrad montiert. Um am Sprocket vorbei zu kommen wurde der Motorenhalter dementsprechend abgeschrägt.

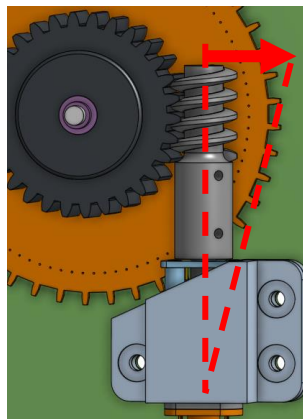


Abbildung 34 Montierter Sprocket-Antriebsmotor und Getriebe

5.1.7 Master Gehäuse

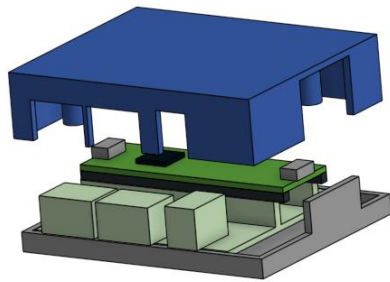


Abbildung 35 Master Gehäuse

Der Master besteht aus einem tinyK22 (Kapitel 3.2.9) und einem PCB, auf welchem Stecker montiert sind. Diese sind für die Verbindungen von Energieversorgung sowie der Kommunikation zwischen Feeder und Master. Das Gehäuse wurde erstellt um eine schützende Hülle um diese Elektronik zu haben.

Das Gehäuse besteht aus zwei Teilen welche zusammengeschraubt werden können. Dabei sind Aussparungen im Gehäuse für die verschiedenen Steckerverbindungen vorgesehen. Der Aufbau vom Gehäuse ist in der Montageanleitung ausführlich ersichtlich. Das PCB, welches für den Master entwickelt wurde, ist in Kapitel 5.2.9 beschrieben.

5.1.8 Feederbank

Der Feederbank, dessen Design in Abbildung 37 dargestellt ist, wurde simpel gehalten. An der PnP Maschine ist ein Ausschnitt für die Montage von Feedern vorgesehen. Dieser Ausschnitt ist ca. 350 mm lang. Die Länge der Feederbank wurde auf diesen Ausschnitt angepasst. Die Feederbank kann Rollen mit einem Aussendurchmesser von bis zu 180 mm aufnehmen. Der Feederbank kann circa 25 Rollen von 8 mm Streifen aufnehmen.

Um die Rollen auszuwechseln, kann der Stab seitlich aus der Führung entfernt werden. Die Rollen liegen dann auf den Wänden und dem Boden der Bank auf. Dies ist in Abbildung 37 dargestellt. Somit verschiebt sich die Position der Rollen um nur 1 mm und der Stab kann mühelos wieder eingefahren werden.

Die Teile der Feederbank wurden aus einer 6 mm MDF Platte gefräst. Der Stab ist aus Holz und hat einen Durchmesser von 12 mm. Ein Bild des Feederbanks ist in Abbildung 15 auf Seite 23 zu sehen.

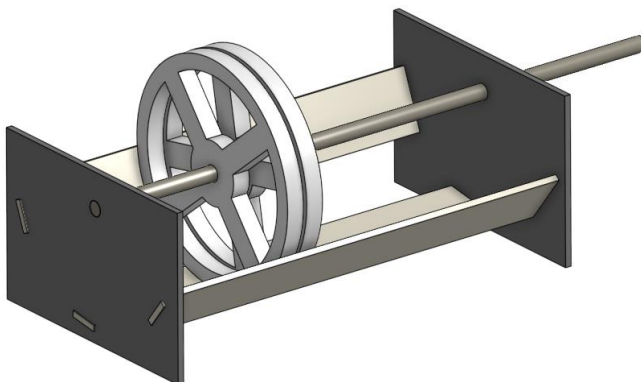


Abbildung 37 CAD der Feederbank

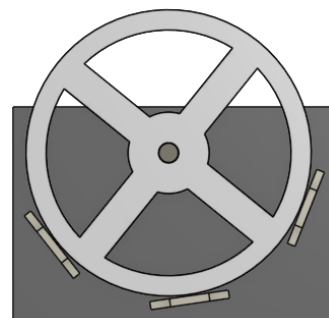


Abbildung 37 Aufliegen der Rolle im Feederbank

5.2 Elektrisches Konzept

In Absprache mit Prof. Styger wurden nach Möglichkeit Komponenten verwendet, welche bereits in Projekten an der HSLU verwendet wurden, so beispielsweise der Mikrokontroller im tinyK20 (Kapitel 3.2.7) oder DC-Getriebemotoren der Reihe N20 (Kapitel 3.2.2). In den folgenden Unterkapiteln wird auf die elektrische Umsetzung des Feeders eingegangen. Abbildung 38 beinhaltet eine Übersicht der Funktionen und Komponenten, welche im Feeder integriert sind.

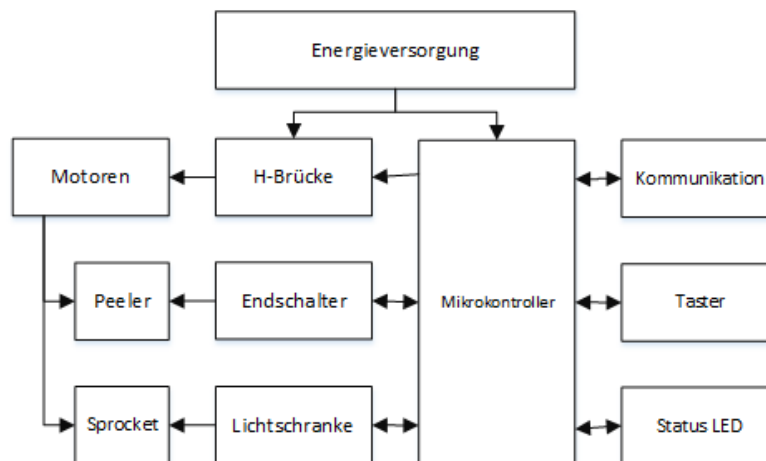


Abbildung 38 Funktionsplan Feeder

Das komplette Elektroschema zum Feeder ist im Anhang abgelegt. Das Elektroschema vom Master-PCB ist im elektronischen Anhang abgelegt.

5.2.1 Energieversorgung

In Kapitel 3.2.10 wurde entschieden, dass jeder Feeder von extern mit 5 VDC gespeisen wird. Die 5 VDC wird zum einen über einen Spannungsregler in 3.3 VDC umgewandelt. Dieses ist der Logikpegel, der zur Speisung des Mikrokontrollers und weiteren elektronischen Komponenten verwendet wird. Die 5 VDC werden zudem direkt als Energieversorgung für die DC-Getriebemotoren verwendet.

Auf jedem Feeder wird für die Energieversorgung der Logik ein Spannungsregler auf 3.3 VDC integriert. Diese Energieversorgung wurde direkt von dem tinyK20 übernommen und als gut befunden.

Im Kapitel 6.7 befinden sich Messungen zur Stromaufnahme des Feeders in verschiedenen Situationen. Die Spannungsversorgung war bei allen Messungen 5 VDC und die Raumtemperatur betrug ca. 23 °C. Folgende Tabelle präsentiert die Werte der Messungen. Dazu gilt zu beachten, dass diese durchschnittliche Werte sind und die Aufnahme sich je nach Belastung der Motoren variieren kann. Diese Belastung wird durch die Art des Streifens und etwaige Verdrehungen, sowie die Drehrichtung des Motors, beeinflusst.

Tabelle 11 Messwerte Stromaufnahme

Situation	Stromaufnahme
Gesamte Logik (MC, Lichtschranke, Status-LED, H-Brücke)	55 mA
Logik & Beide Motoren (leichte Last)	110 - 130 mA
Logik & Beide Motoren (starke Last)	Bis 420 mA
Anlaufstrom	600 mA

Mit dem Bestückungskopf der PnP Maschine können jeweils zwei Komponenten von den Feedern gleichzeitig geholt werden. Zudem wurde definiert, dass bis zum nächsten Pick des Kopfes der Vorschub des Streifens abgeschlossen werden muss. Somit kann angenommen werden, dass jeweils höchstens zwei Feeder gleichzeitig laufen. Entsprechend muss die Energieversorgung für die Anzahl x Feeder wie folgt ausgelegt werden.

$$(x - 2) * I_{Logik} = (x - 2) * 55 \text{ mA} = I_{GesLogik}$$

$$2 * I_{sLast} = 2 * 600 \text{ mA} = I_{GesLast}$$

Somit folgt für die gesamte Stromaufnahme (Anlaufstrom).

$$I_{GesLogik} + I_{GesLast} = (x - 2) * 55 \text{ mA} + 1200 \text{ mA}$$

Für 20 Feeder bedeutet dies eine kurzzeitige Stromaufnahme von 2.19 A. An der PnP Maschine von Prof. Styger ist bereits eine 5 VDC Netzteil vorhanden, welches diesen Bedarf decken kann. Für den durchschnittlichen Betrieb kann jedoch angenommen werden, dass die Stromaufnahme deutlich tiefer liegt. Denn der Anlaufstrom wird nur wenige Millisekunden aufgenommen, bis die Motoren drehen. Bei einer gemittelten Last mit 250 mA und obiger Formel mit 20 Feedern errechnet sich ein Strom von 1.5A.

$$I_{GesLogik} + I_{mLast} = (x - 2) * 55 \text{ mA} + (2 * 250 \text{ mA}) = 1.49 \text{ A}$$

Da die Feeder jedoch auch nicht konstant in Betrieb sind, sinkt die benötigte Energie noch weiter. Eine solche Messung wurde nicht gemacht, da bis zur Abschluss dieser Arbeit die Kombination von OpenPnP Maschine mit den Feedern noch nicht ausgiebig getestet wurde.

5.2.2 Motorensteuerung

Die Motoren werden mit einer H-Brücke vom Typ DRV8835 der Firma Texas Instruments angesteuert. Dieser Motorentreiber kann zwei DC-Motoren einzeln ansteuern und ein gesamter Strom von 1.5 A schalten. Der Treiber kann die Motoren anhalten, sowie vorwärts und rückwärts drehen lassen. Diese Funktionalitäten sind für den Feeder notwendig. Mit jeweils zwei Pins pro Motor kann vom Mikrokontroller die Drehrichtung sowie die Geschwindigkeit des Motors per PWM gesteuert werden.

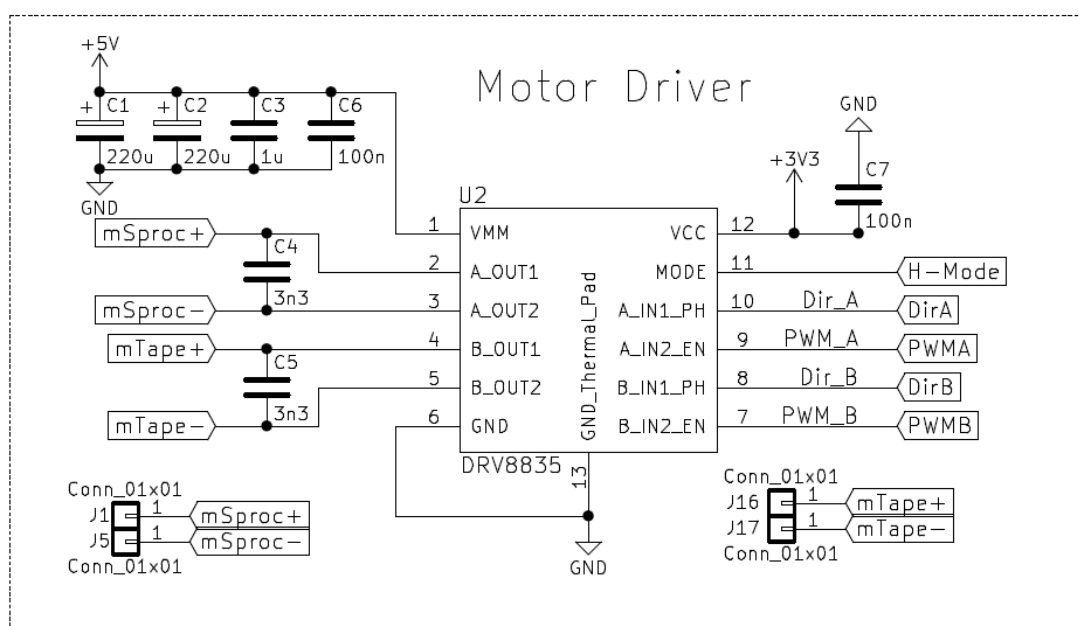


Abbildung 39 Elektroschema des Motorentreibers

Der DRV8835 wird auch im SUMO Roboter verwendet. Die Auslegung des Motorentreibers wurde vom SUMO-Roboter als gut befunden und als Gesamtes übernommen.

Die Kondensatoren C1 und C2 sind als Puffer der Energieversorgung vorgesehen. Beim Anlaufen der Motoren liefern sie einen Teil des kurzzeitigen Anlaufstromes und entlasten somit die Energieversorgung. Die Kondensatoren C3, C6 und C7 vermindern Rauschen an den Spannungseingängen des Treibers. C4 und C5 vermindern Rauschen, welches durch die DC-Getriebemotoren verursacht wird.

5.2.3 Lichtschranke

Die Lichtschranke wird in Kombination mit dem Sprocket verwendet, um den Transport des Streifens zu regeln. Die Bohrungen am Sprocket haben einen Abstand von 3 mm zum Rand, somit wird das Risiko von fremden Lichteinfall verhindert. Die verwendete Lichtschranke, eine EE-SX1350 von Omron (Abbildung 41 a), hat auf einer Seite eine Infrarotdiode, und auf der anderen Seite einen Fototransistor, welcher auf das Infrarotlicht reagiert. Je mehr Infrarotlicht einfällt, desto mehr Strom fließt durch den



Abbildung 41 a) Lichtschranke EE-SX1350 (Omron, 2017)

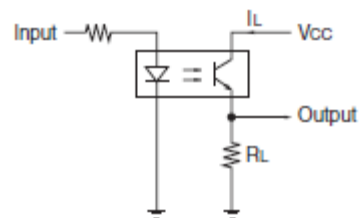


Abbildung 41 b) Schaltungsbeispiel Lichtschranke (Omron, 2017)

Transistor. Dieser Strom, kombiniert mit einem Widerstand, kann verwendet werden um einen Spannungspegel zu erzeugen, welcher als Signal verwendet werden kann (Abbildung 41 b).

Diese Lichtschranke wurde aus Gründen des mechanischen Designs verwendet. Durch den Abstand der Bohrungen zum Rand des Sprockets musste die Lichtschranke eine gewisse Tiefe haben. Zudem dürfte die Lichtschranke selbst nicht allzu breit sein, damit sie nicht die Breite des Feeders negativ beeinflussen würde. Die Lichtschranke wird direkt auf der PCB-Wand aufgelötet und durch das Lötzinn fix befestigt. Das Bein der Lichtschranke mit dem Fototransistor wird an der Sprocket-nahen Seite montiert, um Störungen durch Fremdlicht und Reflektionen zu vermeiden.

Im endgültigen Sprocketdesign wurden Bohrungen mit einem Durchmesser von 0.7 mm eingelassen. Der gemessene Strom durch Lichteinfall durch eine solche Bohrung war durchschnittlich 660 μA . Als R_L , respektive R5 im Elektroschema (Abbildung 42) wurde ein Widerstand von 4.7 k Ω verwendet. Dies resultierte in eine Spannung von 3.1 V. Im Versuchsaufbau, welcher in Kapitel 6.4 aufgezeigt wird, wurde ersichtlich, dass das Schaltverhalten der Lichtschranke keine steilen Flanken aufweist. Es wurde ein weiterer Transistor mit dem Signal der Lichtschranke geschaltet, um diese steilen Flanken zu erhalten. Zudem wird so auch gewährleistet, dass bei niedrigerem Lichteinfall, welches in einem tieferen Spannungspegel resultiert, auch ein Schaltsignal vorliegt. Der eingesetzte Transistor BC847 schaltet bei ca. 0.7 V, und kann einen Hoch-Pegel von 3.3 V bereitstellen. Der Mikrokontroller erkennt einen Hoch-Pegel erst ab 2.3 V (bei einer Spannungsversorgung von 3.3V). So wird gewährleistet, dass am Mikrokontroller ein gutes Signal vorliegt.

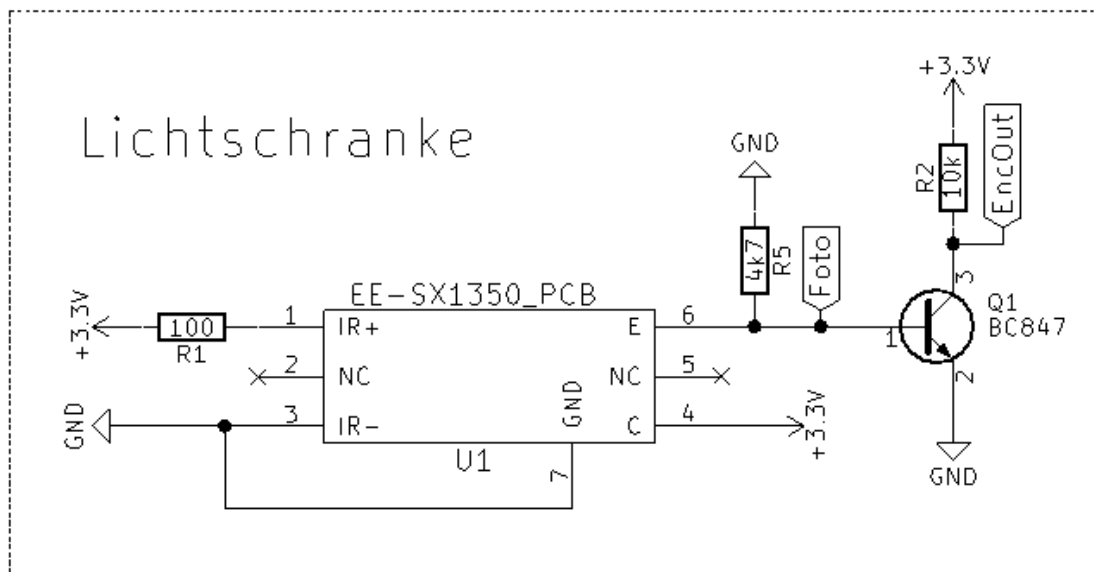


Abbildung 42 Elektroschema der Lichtschranke

Das Infrarot LED hat laut Datenblatt eine typische Vorwärtsspannung V_F von 1.2V und kann mit einem Strom I_F von 20 mA bis maximal 30 mA betrieben werden. Daraus folgt für den Vorwiderstand R1:

$$\frac{V_{CC} - V_F}{I_F} = \frac{3.3 \text{ V} - 1.2 \text{ V}}{20 \text{ mA}} = 105 \Omega$$

Es wird ein Widerstand der Reihe E24 von 100 Ω als Vorwiderstand des LEDs verwendet.

5.2.4 Switches & LED

In Kapitel 1.3 wurde definiert, dass der Streifen im Feeder per manueller Ansteuerung vor- und rückwärts gefahren werden kann. Dies wird mit zwei Taster (SW1 und SW2), welche vom Mikrokontroller überwacht werden, umgesetzt. Nebst der manuellen Ansteuerung des Feeders wurde auch eine Statusanzeige als Ziel definiert. Diese wird mit dem LED D1 erreicht.

Das LED D1 hat laut Datenblatt eine Vorwärtsspannung V_F von 2.0V und kann mit einem Strom I_F von 25 mA betrieben werden. Dieselbe Rechnung wie bei dem Infrarot LED (Kapitel 5.2.3) oben führt hier zu einem Vorwiderstand der Reihe E24 von 47 Ω , welches in einem Vorwärtsstrom von 23.4 mA resultiert.

Wie die Vor- und Rückwärtstaster wird auch der Mikroschitch (SW3) des Peelers vom Mikrokontroller überwacht. Der Mikroschitch wird durch Kontakt mit der Abdeckungsfolie betätigt. Eine ausführliche Erklärung der Funktionalität befindet sich in Kapitel 5.1.5.

Abbildung 43 zeigt den Ausschnitt des Elektroschemas zu diesen Komponenten auf. Auf eine Entprellung mit Kondensatoren wurde verzichtet, da diese in der Software (Kapitel 5.3.3) umgesetzt wird.

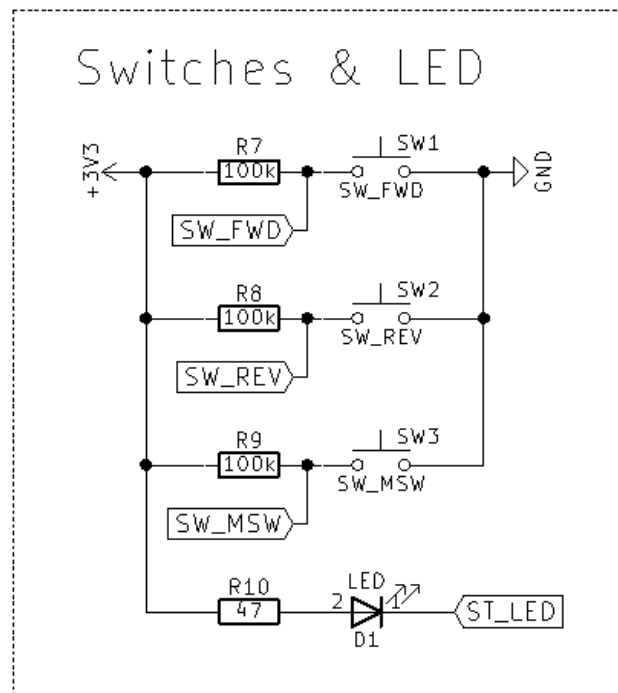


Abbildung 43 Elektroschema der Switches und des LEDs

5.2.5 Verbindungsaufbau Feeder

In Kapitel 3.2.8 wurde entschieden die Kommunikation mittels einem Daisy-Chain über UART zu lösen. Somit müssen die Feeder miteinander verbunden werden. Um die Energieversorgung und die

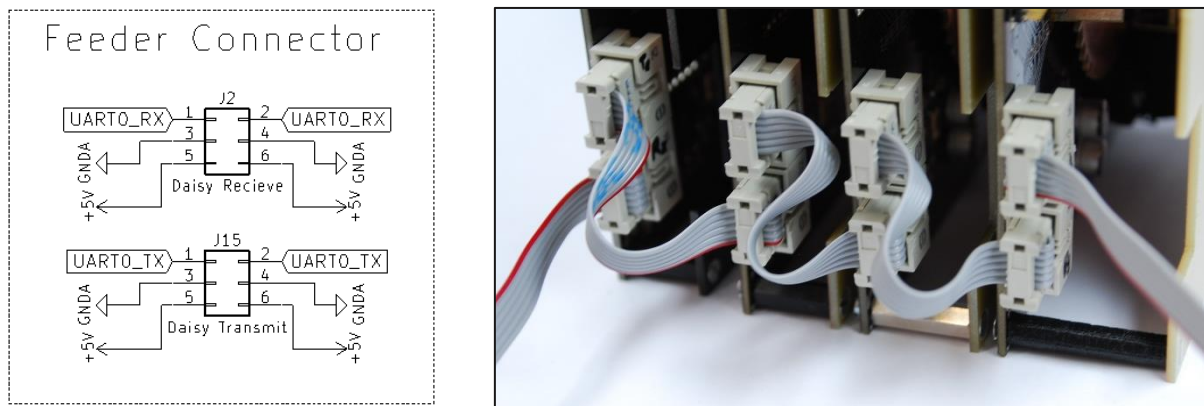


Abbildung 44 Verbindungstecker-Schema und Feeder Verbindungen

Kommunikation Feeder zu ermöglichen, sowie um den Verdrahtungsaufwand klein zu halten, werden die Feeder mit einem Flachbandkabel verbunden. Da die UART Datenleitungen wegen des Daisy-Chains nicht durchgehend verbunden werden dürfen, wird jeweils nur eine Verbindung von einem Feeder zum nächsten realisiert. Über diese Steckverbindung wird nebst der Datenleitung auch die Versorgungsspannung von 5 VDC von einem Feeder zum nächsten übergeben. Die schematischen und mechanischen Verbindungen sind Abbildung 44 ersichtlich. Die Kabel, welche in Abbildung 44 rechts und links aus dem Bild führen, werden mit dem Master verbunden.

Die Verbindungsstecker stammen aus der DIN 41651 Reihe. Diese Stecker sind robust, halten auch ohne einzurasten gut und können nicht verdreht eingesteckt werden. Somit sind diese Stecker auch vorteilhaft im Sinne der Benutzerfreundlichkeit.

5.2.6 Mikrokontroller

In Kapitel 3.2.7 wird entschieden, dass als Mikrokontroller zur Steuerung des Feeders ein MK20DX128VFT mit ARM Cortex M4 zum Einsatz kommt. Dieser Mikrokontroller ist derselbe welcher auf dem tinyK20 verwendet wird. Da die Grundschialtung um den Mikrokontroller sich bei dem tinyK20 bewährt hat, wird dieselbe komplett für den Feeder übernommen (Abbildung 45). Zu diesem gehören die Speisungsanbindung, welche links oben erkennbar ist, sowie die Quarze Y1 und Y2.

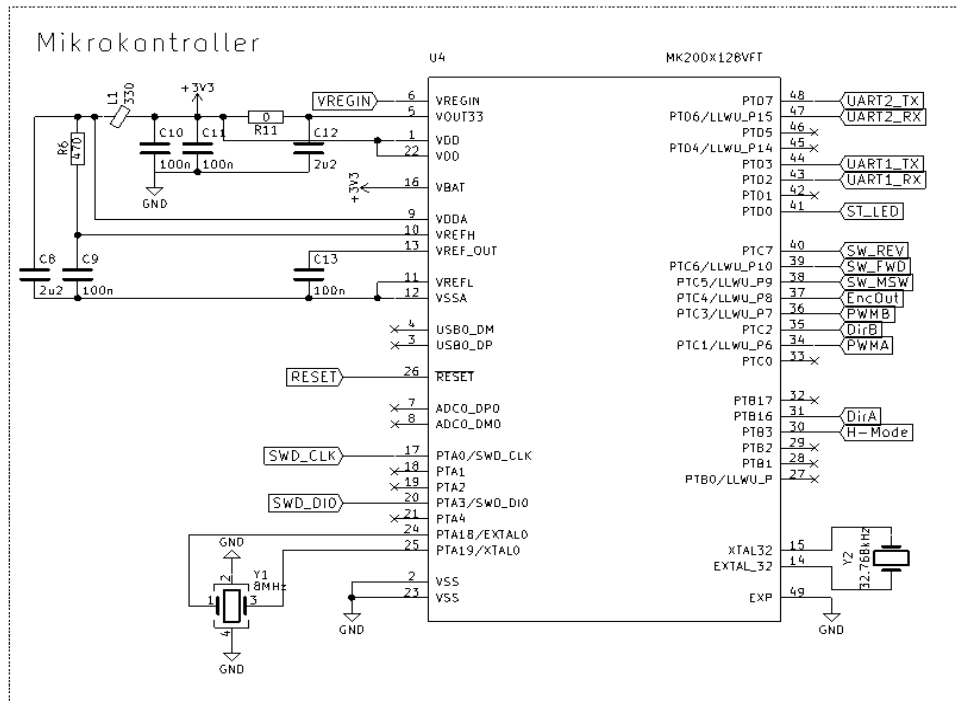


Abbildung 45 Elektroschema des Mikrokontrollers

Aus dem Schemaausschnitt sind auch die verwendeten Pins des Mikrokontrollers, welche zur Ansteuerung der H-Brücke und dem Einlesen der Taster und Lichtschranke verwendet werden, ersichtlich. Folgende Tabelle beschreibt die Pinbelegungen, welche für die Feeder-Funktionalitäten relevant sind.

Tabelle 12 Pinbelegung Mikrokontroller

Nr.	Pin MC	Bezeichnung	Beschreibung
30	PTB3	H-Mode	Modus der H-Brücke
31	PTB16	DirA	Richtungswahl Sprocketmotor
34	PTC1	PWMA	PWM-Signal Sprocketmotor
35	PTC2	DirB	Richtungswahl Peelermotor
36	PTC3	PWMB	PWM-Signal Peelermotor
37	PTC4	EncOut	Schrittsignal der Lichtschranke
38	PTC5	SW_MSW	Mikroswitch
39	PTC6	SW_FWD	Vorwärts-Taster
40	PTC7	SW_REV	Rückwärts-Taster
41	PTD0	ST_LED	Status-LED
43	PTD2	UART2_RX	UART Recieve für Debug Interface
44	PTD3	UART2_TX	UART Transmit für Debug Interface
47	PTD6	UART0_RX	UART Recieve für Feeder Kommunikation
48	PTD7	UART0_TX	UART Transmit für Feeder Kommunikation

5.2.7 Debug Interface

Das Debug Interface als Single-Wire-Device (SWD) Target wurde vom tinyK20 übernommen. Über diesem Anschluss lässt sich der Mikrokontroller des Feeders programmieren und debuggen. Zudem ist auf diesen Stecker auch eine UART Schnittstelle geführt. Somit kann während dem Debuggen auch über UART Informationen übertragen werden. In Abbildung 46 ist zudem der Reset-Taster des Mikrokontrollers ersichtlich.

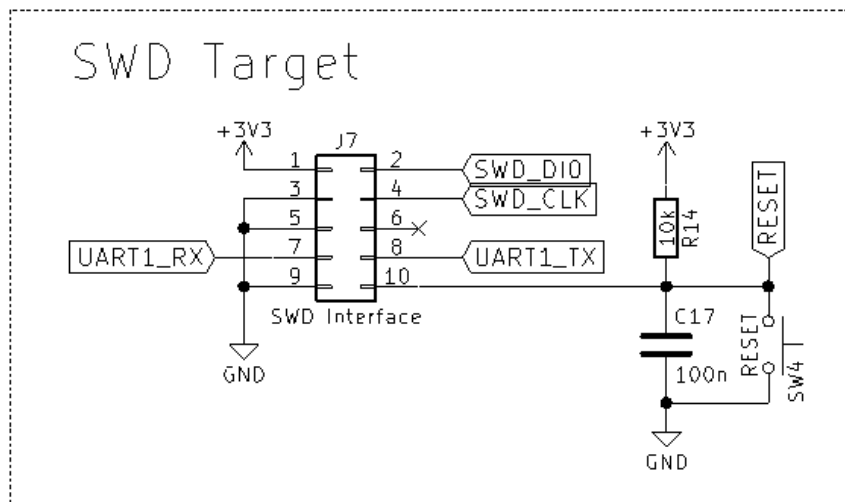


Abbildung 46 Elektroschema zum Debug Interface

5.2.8 Feeder-PCB

Das Feeder-PCB wurde in mehreren Iterationen entwickelt. In diesem Unterkapitel werden die entwickelten Versionen vorgestellt. Das Layout der bei Projektanfrage vorliegenden Version wird auch beschrieben. Die KiCAD-Projekte der Versionen sind im elektronischen Anhang im Ordner KiCAD abgelegt.

Version 1

Version 1 hatte noch keine elektrische Funktion. Dieses PCB wurde erstellt, um die mechanische Montage an der PnP-Maschine zu testen, sowie die Montage der mechanischen Komponenten wie die Führung und die Motoren an der Wand. Dabei wurde erkannt, dass die Halterung am Aluminiumprofil gut funktionierte. Jedoch war das PCB zu klein, um neben den mechanischen Komponenten auch die elektrischen Komponenten wie den Mikrokontroller zu tragen. Somit wurde die Version 2 verlängert. Version 1 ist in Abbildung 26 auf Seite 31 abgebildet.

Version 2

Version 2 war das erste PCB, welches auch ein Layout für die Elektronik besass. Das PCB wurde gegenüber Version 1 vergrößert, um Platz für alle Komponenten zu schaffen. Dies führte auch dazu, dass die Führungen aus 3D Druck verlängert werden mussten. Beim Bestücken von Version 2 wurde erkannt, dass der Footprint des Mikrokontrollers zu klein war. Entsprechend wurde dies in Version 2.1 angepasst. Version 2 ist in Abbildung 48 im linken Bild zu sehen.

Version 2.1

Mit angepasstem Footprint des Mikrokontrollers wurde dieses PCB bestückt. In Kapitel 6.2 wird auf die Erkenntnisse des Prototyps mit diesem Layout eingegangen. Zusammenfassend beschrieben musste nach dem Test dieses PCBs der Footprint der Lichtschranke verändert werden.

Version 3

In Version 3 wurde der Footprint der Lichtschranke angepasst. Zudem wurden Fiducials platziert, damit das PCB auch an der PnP Maschine bestückt werden könnte. Diese Version wurde als definitive Version für den Feederbau bei PCBWay bestellt. Somit wurden diese Änderungen nicht vorgängig an der HSLU getestet.

Für die Montage der Lichtschranke wird ein zusätzliches kleines PCB verwendet. Dieses wird verwendet, um die Anschlüsse der stehenden Lichtschranke auf das Feeder-PCB zu leiten. Zudem bietet es einen Schutz der Lichtschranke, welches selbst leicht bricht. Dieses kleine PCB wurde in Version 3 an das Feeder-PCB angebracht, damit beide PCBs als ein komplettes PCB bestellt werden konnten. Das Lichtschrankens-PCB kann vom Feeder-PCB getrennt werden. Die Lichtschranke wird zuerst getrennt bestückt, gebacken und erst danach auf dem Feeder-PCB aufgelötet. Zwei frischgebackene Lichtschranken mit PCB sind in Abbildung 47 abgebildet.



Abbildung 47 Lichtschranken mit PCB

Version 3 ist im rechten Bild in Abbildung 48 ersichtlich. Das angesetzte Lichtschrankens-PCB ist rechts an Version 3 erkennbar. Dieses Stück ist in Version 2 nicht vorhanden. Das Lichtschrankens-PCB ist im Elektroschema als Encoder Mounting Board beschriftet.

In Version 3 wurden beim Bestücken und Testen mehrere kleine Fehler gefunden. Mehrere gedruckte Referenzen (Silkscreens) waren bei den falschen Pads platziert. Da jedoch für das Bestücken das KiCAD zu Hilfe genommen wurde, konnten diese Fehler erkannt und somit Folgefehler vermieden werden.

Getauscht werden mussten die Referenzen von L1 und C10, sowie R6 und C8. Diese Komponenten liegen gleich nebeneinander.

Beim Testen wurde erkannt, dass bei der Lichtschranke eine Leiterbahn eine ungewollte Verbindung auf den GND-Plane verursachte. Beim Wechseln des Lichtschrankens-Footprint von Version 2.1 zu Version 3 wurde diese Bahn übersehen. Diese Verbindung kann jedoch mit dem Skalpell getrennt werden und das PCB einsatzfähig gemacht werden.

Version 4

In Version 4 wurden die Fehler aus Version 3 korrigiert. Beim Bestellvorgang bei PCBWay wurden vom Hersteller mehrere Punkte in Bezug auf die Anbindung des Lichtschrankens-PCBs gemeldet. Diese Meldungen führten auch zu Anpassungen in diesem Bereich, welche in Version 4 umgesetzt wurden. Es wurden Änderungen an selbst erstellten Footprints für die Lichtschranke gemacht. Version 4 wurde nicht produziert.

Version 5

In Version 5 wurden der Footprint der Fiducials gemäss wünschen von Prof. Styger angepasst. Zudem wurde der Umriss von der veränderten CAD-Dateien des Feeders übernommen. Diese betraf Änderungen an Bohrungsdurchmessern. Version 5 ist die aktuellste Version welche bei Projektabgabe bestand. In Version 5 wurde auch die Seitenwand ohne Layout überarbeitet.

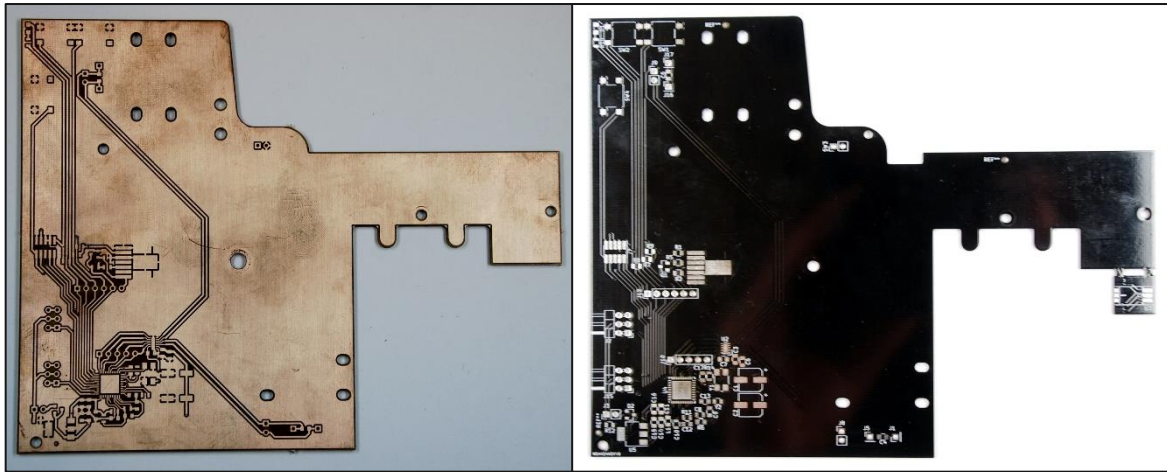


Abbildung 48 Version 2 und 3 des Feeder-PCBs

Beschreibung Layout

Das Layout wurde zwischen Version 3 und 5 kaum verändert, es wurden nur Fehlerkorrekturen vorgenommen. In Abbildung 49 sind die verschiedenen Felder des PCBs beschrieben. Dabei wurden die in Zusammenhang stehenden Komponenten gruppiert und die verschiedenen Funktionen voneinander getrennt. Beispielsweise wurde die H-Brücke samt Leiterbahnen zu den Motoren von der Kommunikation und Lichtschranke getrennt. So soll ein Übersprechen von Leiterbahnen verhindert werden und Störungen minimiert werden. Die mit blau umrandeten Header werden im Normalfall nicht bestückt, bieten aber bei allfälliger Fehlersuche eine Möglichkeit Signale auszulesen.

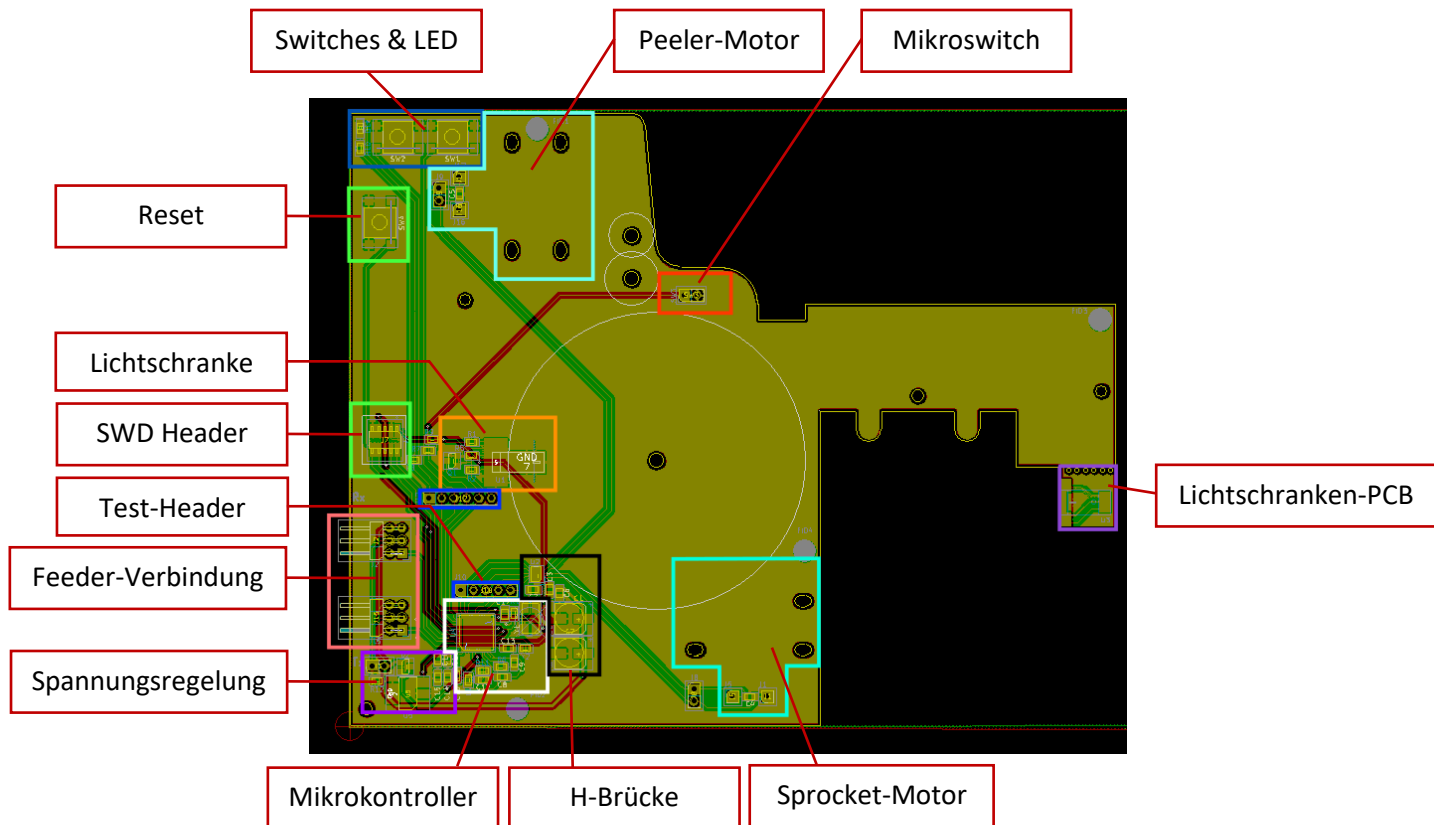


Abbildung 49 Layout V5 mit Feldbeschreibung

5.2.9 Master-PCB

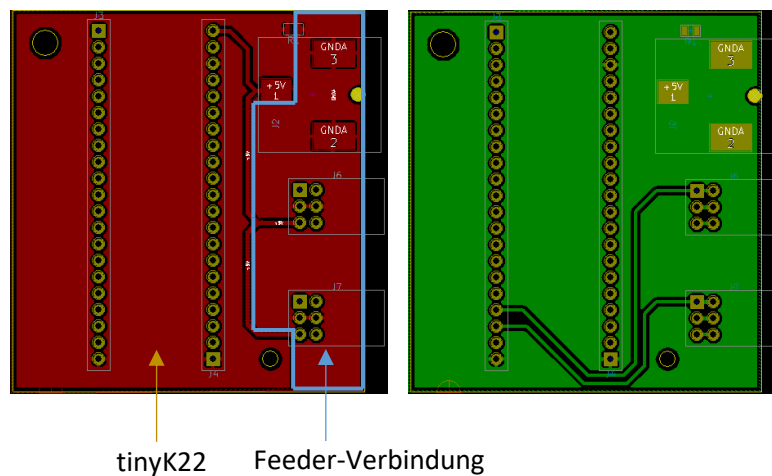


Abbildung 50 Übersicht Layout Master-PCB

Das Master-PCB wurde erstellt um eine saubere Lösung für die Verbindungen zwischen Master und Feeder zu ermöglichen. Da nebst der Kommunikation auch die Energieversorgung der Feeder über dasselbe Kabel verläuft, wurde die Haupt-Energieversorgung auch über das Master-PCB geführt. Mit dieser kann zudem gleichzeitig das tinyK22 gespeist werden, welches eine zusätzliche Versorgung erspart.

Das PCB wurde in zwei Grundflächen aufgeteilt. Der Groundplane der Schnittstellen gegen aussen und das tinyK22 als eigenständiges Mikrocontrollerboard wurden voneinander getrennt. Die Groundplanes werden über einen $0\ \Omega$ Widerstand voneinander getrennt. Somit werden kann ein Anteil des Rauschens, welches von den Verbindungsleitungen aufgenommen wird, vom tinyK22 ferngehalten werden. Das Layout ist in Abbildung 50 dargestellt. In Abbildung 51 ist ein bestücktes PCB mit montiertem tinyK22 zu sehen. Dieses ist bereits im Boden vom Mastergehäuse montiert.

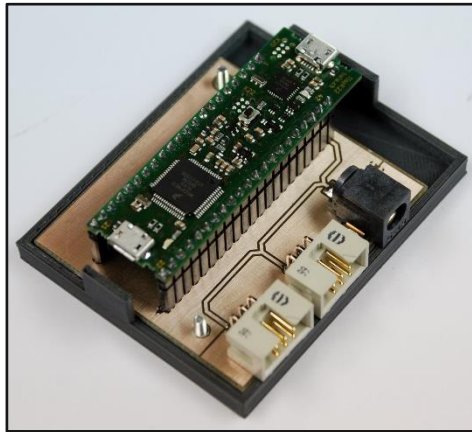


Abbildung 51 Bestücktes Master-PCB mit tinyK22

5.3 Software

Dieses Kapitel befasst sich mit dem Aufbau der Software sowie den Überlegungen, welche zu den einzelnen Teilfunktionen gemacht wurden. Das Kapitel beinhaltet eine Systemübersicht, welches das Grundkonzept darstellt. In weiteren Unterkapiteln werden die Software-Implementation von der Lichtschranke, der Taster, den Motoren, dem Haupt-Zustandsautomaten sowie der Kommunikation beschrieben. Die Software wurde in Kinetis Design Studio geschrieben, mit Einbindung des Software-Tools Processor Expert.

In Kapitel 3.2.6 wurden die Softwarefunktionalitäten definiert. Abbildung 38 Abbildung 38 Funktionsplan Feeder auf Seite 39 beinhaltet eine Übersicht der Zusammenhänge von Softwarefunktionen und elektrischen Komponenten.

5.3.1 Grundkonzept

Für die Umsetzung der Funktionalitäten wird das Echtzeitbetriebssystem FreeRTOS eingesetzt. Im Modul Infotronic wurden Erfahrungen mit FreeRTOS gemacht und, mit Empfehlung von Prof. Styger, auch den Einsatz in diesem Projekt festgelegt. Die Verwendung von einem RTOS in diesem Projekt bot den Vorteil, dass man die Teilfunktionen einzeln und unabhängig implementieren konnte.

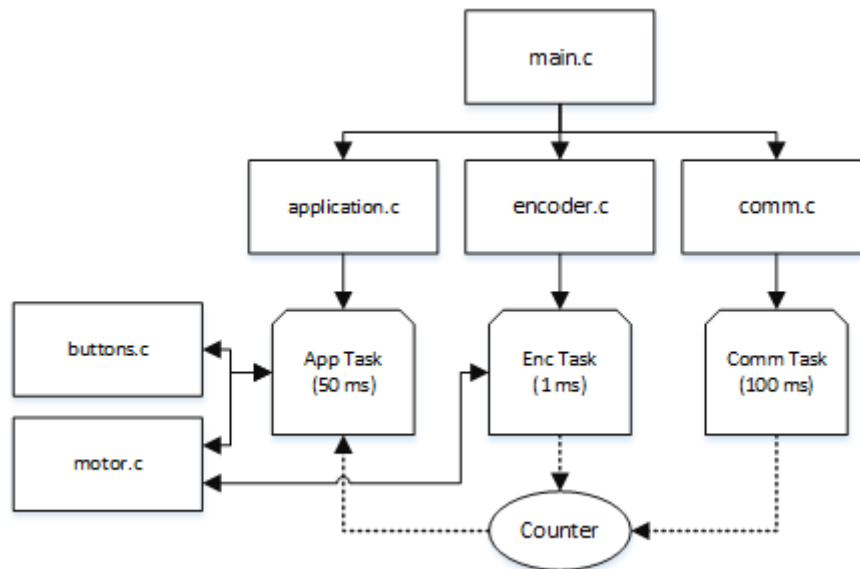


Abbildung 52 Softwareübersicht

Abbildung 52 zeigt die grundlegenden Zusammenhänge der Software dar. In der Software wurden drei Tasks erstellt. Der «Enc» Task ist für die Lichtschranke zuständig. Der «Comm» Task ist für die Kommunikation verantwortlich, und der dritte Task, «App», kombiniert alle Informationen und beinhaltet die eigentliche Funktionalität vom Feeder, der Streifentransport. Die Schrittbefehle werden über die Variable «Counter», welche folgend weiter beschrieben wird, übergeben. In folgenden Unterkapiteln wird zu allen Softwareteilen ausführlichere Informationen gegeben.

5.3.2 Lichtschranke

Der Zustand der Lichtschranke, respektive der Erkennung eines Schrittes erfolgt durch Sampling. Nach dem Nyquist-Shannon-Abtasttheorem sollte ein Signal mit der doppelten Frequenz des Signals abgetastet werden. Aus den Versuchen in Kapitel 6.4 ist bekannt, dass die Periodendauer des Signals ca. 140 ms lang ist, wobei der High-Pegel durchschnittlich 40 ms lang ist. Durch Verunreinigungen kann sich die Zeitdauer verkürzen. Wird davon ausgegangen, dass der High-Pegel 20 ms dauert, müsste mit einer Frequenz von 100 Hz abgetastet werden.

$$f_{\text{sample}} = 2 * f_{\text{max}} = 2 * 50 \text{ Hz} = 100 \text{ Hz}$$

Für eine konstante, wiederholende Positionierung ist jedoch wichtiger, dass der Schritt möglichst schnell erkannt wird. Somit wird eine höhere Abtastrate verwendet. Hierbei wird die Task-Scheduling Funktionalität vom FreeRTOS verwendet, welches es ermöglicht, eine Taskfrequenz von 1 kHz einzusetzen. Mit dieser Abtastrate wird das Signal der Lichtschranke im 1 ms Takt abgetastet. Somit wird ein Wechsel des Signales sehr schnell erkannt. Die Abtastrate könnte mit Verwendung eines eigenen Timers, respektive Interrupts, erhöht werden. Dies wird jedoch würde nicht zwingend zu einer Verbesserung der Positionierung führen, da die mechanischen Toleranzen und das Bremsverhalten des DC-Getriebemotors auch einen gewissen Einfluss auf das System haben.

Der Lichtschranken-Task wird im 1 ms Takt aufgerufen. In dieser Zeit wird das eingelesene Signal mit dem vorherigen Signal verglichen. Wird eine Veränderung festgestellt, wird ein Flag gesetzt. Um externe Fehleinflüsse zu vermindern, wird nun weitere drei Taskaufrufe festgehalten ob das Signal gleichbleibt. Erst zu diesem Zeitpunkt wird dann ein Schritt gezählt. Das Flussdiagramm in Abbildung 53 in Kombination mit der Signaldarstellung macht dieses Verhalten deutlich.

Wird erkannt, dass der komplette Bauteilweg erreicht wurde, wird in diesem Task direkt auch der Antriebsmotor des Sprockets ausgeschaltet. Über einem weiteren Flag wird der Wechsel des Zustandes an den anderen Tasks übermittelt.

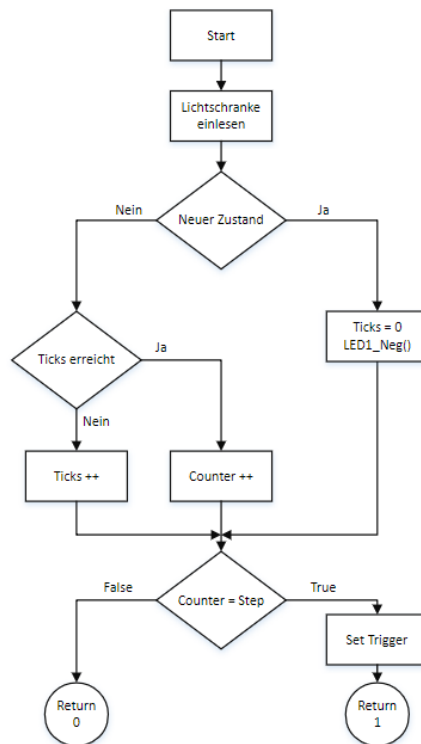


Abbildung 53 Ablauf Schritterkennung

5.3.3 Buttons

Die Buttons Source Datei beinhaltet mehrere Funktionen, welche zur Auswertung der Taster verwendet werden. Für die Taster wurden drei Zustände implementiert, wobei nur der Rückwärtstaster eine Implementierung von BUT_LPRESS besitzt.

Tabelle 13 Zustände der Taster

Zustand	Beschreibung
BUT_IDLE	Unbetätigt
BUT_PRESS	Betätigt (kurz)
BUT_LPRESS	Betätigt (lange)

Die Taster werden über einen Switch Case abgefragt, wobei jeder Taster als eigener Case definiert ist. Der Wert des Tasters wird ausgelesen. Folgendes Flussdiagramm verdeutlicht die Abarbeitung der Tasterauslesung. Bei einem PRESS-State des Vorwärtstasters und einem LPRESS des Rückwärtstasters die numerische Variable Counter verändert. Diese wird zur Ausführung eines Schrittbefehls verwendet. Bei einem kurzen Tasterdruck (PRESS) des Rückwärtstasters wird der Feeder angehalten. Dies ist in der Darstellung des Zustandsautomaten in Abbildung 55 erkennbar.

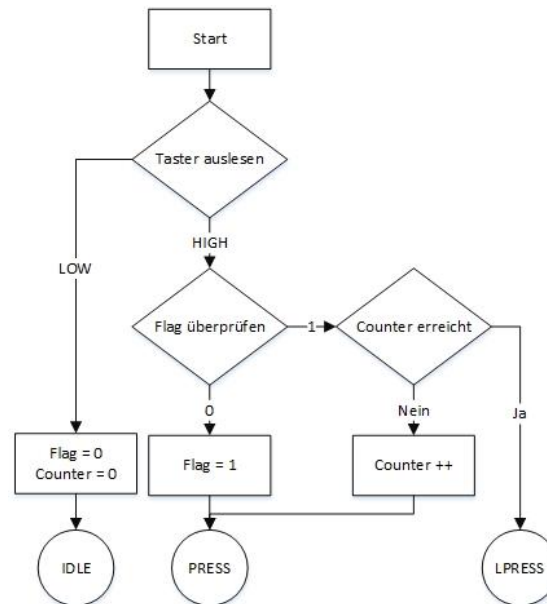


Abbildung 54 Ablauf Tasterauslesung

5.3.4 Application

In der AppTask der Software wird die Grundfunktionalität abgearbeitet. Dieser besteht aus zwei Zustandsautomaten, welche miteinander agieren, aber aus Gründen von Funktionsaufteilung und Verständlichkeit in zwei getrennte Systeme aufgeteilt wurden. Des Weiteren wird in der App Task auch die Zustände der Taster abgefragt.

Teil des Application Source Files ist auch der Counter, folgend APP_Counter genannt. Diese numerische Variable wird dazu verwendet, um Vor- und Rückwärtsschritte zu ermöglichen. Diese Variable wird über eine Betätigung der Taster, respektive bei einem Befehl über die Kommunikation verändert. Soll der Feeder einen Schritt vorwärts machen, wird APP_Counter inkrementiert, bei einem Rückwärtsschritt dekrementiert. Im Zustandsautomaten wird APP_Counter ausgelesen, und entsprechend dem Wert von APP_Counter einen Zustandswechsel eingeleitet. Vor dem Wechsel wird jedoch der Wert von APP_Counter in Richtung 0 inkrementiert/dekrementiert. So werden die Schrittbefehle vom Feeder abgearbeitet.

In Abbildung 55 wird der Zustandsautomat (Finite State Machine, folgend FSM) aufgezeigt. Darauffolgend sind die einzelnen Zustände kurz beschrieben. Diese Darstellung beinhaltet beide Zustandsautomaten. Sprocket FSM beinhaltet die Grundfunktionalität des Streifentransports, ändert zudem auch teilweise die Zustände im Peeler FSM. Dieser Teil ist für den Peeler-Antrieb zuständig. Da der Peeler unter anderen Ein- und Ausschaltbedingungen arbeitet wurden seine Funktionen in den zweiten Zustandsautomaten verschoben. Die übergreifenden Verbindungen sind mit einer gestrichelten Linie von links nach rechts dargestellt. Der Zustand Error wird mit einem Timeout erreicht, welcher beginnt sobald IDLE verlassen wird, also ein Schritt gefahren werden muss. Wird IDLE nach der definierten Zeit nicht erreicht, wird der Error Zustand aktiv. Folgend sind die Zustände

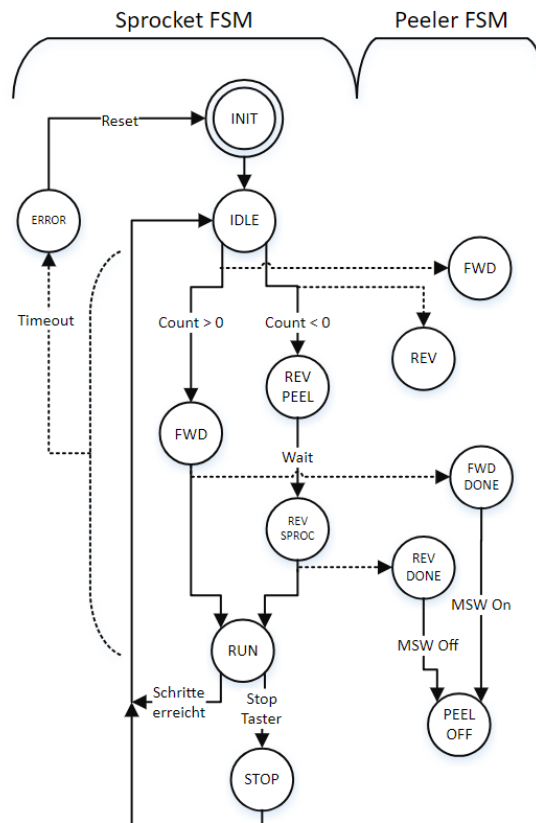


Abbildung 55 Zustandsautomat

ausführlicher beschrieben.

Sprocket FSM

INIT

Dieser Zustand stellt die Initialisierung aller Zustände nochmals sicher. Dabei werden die Motoren ausgeschaltet und die Lichtschranken-Counter zurückgesetzt. An dieser Stelle blinkt die Status-LED des Feeders 2-mal, um somit den einsatzbereiten Betrieb des Feeders darzustellen.

IDLE

Hier wird bei jeder Iteration der Lichtschranken-Counter zurückgesetzt. Somit wird verhindert, dass Fehlzählungen, welche durch äussere Einflüsse entstanden sind, einen Fehlschritt verursachen können. An dieser Stelle wird die Variable APP_Counter ausgelesen, welche bei einem Wert ungleich 0 zu einem Zustandswechsel führt.

FWD

Der Sprocketmotor wird in Vorwärts-Drehrichtung eingeschalten. Der Zustand des Peelers wird folgend auf FWD_DONE gewechselt, und derjenige des Feeder FSMs auf RUN.

REV_PEEL

Der Zustand des Peelers wird auf PEEL_REV gesetzt. Der Peeler hat eine kleinere Drehzahl als der Streifenantrieb (Siehe Kapitel 5.1.4 und 5.1.5). Bei einem Vorwärtsbetrieb kann der Peeler problemlos nachlaufen und die Abdeckungsfolie abziehen, auch wenn der Schritt vom Streifenantrieb abgeschlossen ist. Beim Rückwärtsbetrieb muss zuerst der Peeler eine gewisse Menge an Folie zurückfördern, damit beim Rückwärtsschieben des Streifens keine Klemmwirkung zustande kommt. In diesem Zustand wird nun eine Zeit gewartet, dass der Peeler rückwärtslaufen kann. Nach Ablauf der Zeit wird der Zustand zu REV_SPROC gewechselt.

REV_SPROC

An dieser Stelle wird nun der Sprocketmotor in Rückwärtsdrehrichtung eingeschalten. Es folgt ein Zustandswechsel zu RUN und beim Peeler auf REV_DONE.

RUN

In diesem Zustand wird der Flag der Lichtschranke abgefragt, welcher angibt, ob der Weg gefahren wurde. Ist dieser auf 1, wird der Zustand auf FSM_IDLE zurückgestellt. Der Sprocket-Antriebsmotor wurde an dieser Stelle bereits durch den Lichtschrankentask ausgeschalten. Der Peelerzustand wird auf PEEL_FWD_DONE gewechselt.

Es wird nebst der Lichtschranke auch der Zustand des Rückwärtstasters abgefragt. Wird dieser hier kurz betätigt, wird ein Wechsel zu FSM_STOP und PEEL_OFF gemacht.

STOP

Dieser Task wird verwendet um den Sprocket-Antriebsmotor auszuschalten. Dieser Zustand wird nur aufgerufen, wenn der Rückwärtstaster kurz betätigt wird. So kann der Benutzer bei laufendem Betrieb den Feeder stoppen.

ERROR

In diesem Zustand wird das Status-LED geblinkt. Damit wird ein Fehlverhalten des Feeders dem Benutzer visuell übermittelt.

Peeler FSM

OFF

Der Peelermotor wird ausgeschalten.

FWD

Der Peelermotor wird, falls der Mikroschitch nicht betätigt ist, im Vorwärtsbetrieb eingeschalten.

REV

Der Peelermotor wird im Rückwärtsbetrieb eingeschalten.

FWD DONE

Ist der Mikroschitch betätigt, wird in den Zustand zu PEEL_OFF gewechselt.

REV DONE

Ist der Mikroschwitch nicht betätigt, wird in den Zustand PEEL_OFF gewechselt.

5.3.5 Motoren

In Kapitel 5.2.2 wird die H-Brücke sowie die Motorenansteuerung beschrieben. Für die Ansteuerung der Motoren wird die H-Brücke in Phase/Enable Mode betrieben. Der Datenblattauszug in Abbildung 56 zeigt die Funktionalität der H-Brücke in diesem Modus.

MODE	xENABLE	xPHASE	xOUT1	xOUT2	FUNCTION (DC MOTOR)
1	0	X	L	L	Brake
1	1	1	L	H	Reverse
1	1	0	H	L	Forward

Abbildung 56 Phase/Enable Mode der H-Brücke (Texas Instruments, 2016)

Mit dem Pin H-Mode am Mikrokontroller wird die H-Brücke in diesen Modus geschaltet. Für die jeweiligen Motoren sind die Pins PWMA/PWMB und DirA/DirB vorhanden, welche als xEnable und xPhase interpretiert werden können.

xPhase entspricht der Drehrichtung (DirA/B). Mit einem logischen 1 wird der Rückwärtsbetrieb eingeschaltet, bei logisch 0 der Vorwärtsbetrieb.

Das PWMA/B wird an xEnable angelegt. Bei einem logischem 1 im PWM Signal kann der Motor in der mit xPhase eingestellten Richtung drehen.

In den motor.h und motor.c Dateien werden diese Funktionalitäten umgesetzt. Es werden zwei Motoren definiert, MOT_SPROC und MOT_TAPE, welche als Referenz vom Sprocket-Antrieb und Peeler-Antrieb fungieren. Dazu wurden die beiden Drehrichtungen MOT_FWD und MOT_REV definiert. Zudem können die Drehrichtungen der Motoren im Headerfile invertiert werden. Dies kann je nach Montage der Motoren nötig sein.

Beim Einschalten des Feeders wird MOT_Init() aufgerufen, welches den H-Brücken Modus setzt und die Motoren ausschaltet.

Mit dem Befehl MOT_Speed kann die Drehrichtung und Geschwindigkeit eines Motors eingestellt werden. Als Parameter erhält dieser Befehl den Motor, die Geschwindigkeit in Prozent (0-100) und die Drehrichtung (MOT_FWD/MOT_REV). Mit der Geschwindigkeitsangabe 0 werden die Motoren ausgeschaltet.

```
uint8_t MOT_Speed(MOT_Device motor, uint8_t speedPercent, MOT_Direction dir){[]
```

Abbildung 57 Funktions-Header von MOT_Speed

Durch die Implementationen in diesem Befehl werden die Eingänge der H-Brücke entsprechend gesetzt und die Motoren drehen oder halten an. Als Rückgabe der Funktion über den uint8_t könnte eine Fehlerrückmeldung erfolgen. Diese ist in der momentanen Implementation der Software jedoch nicht realisiert.

5.3.6 Kommunikation

Die Verbindung der Feeder wird in Kapitel 5.2.5 beschrieben. Aus diesem folgte, dass jeder Feeder Befehle an die Adresse 0 ausführt. Ist die Adresse im Befehl ungleich 0, wird die Adresse dekrementiert und an den nächsten Feeder weitergeleitet. Der Aufbau der Befehle wird in diesem Kapitel noch ausführlich beschrieben.

Die Kommunikation der Feeder läuft über eine UART Schnittstelle. In der Software des Feeders wird eine Shell-Komponente verwendet, welche mit weiteren referenzierten Komponenten den hardwarenahen Code erstellt, damit die Kommunikation realisiert werden kann. Zudem wurden Code-Teile des Sumo-Roboters aus dem Modul Infotronik übernommen und angepasst. Somit konnte für die Implementation der Kommunikation eine einfach verständliche Umgebung geschaffen werden.

Die Processor Expert Komponenten reagieren über Interrupts auf ein- und ausgehende Befehle, welche in einem Buffer gespeichert werden. Der Kommunikationstask ruft den Befehl COMM_ReadAndParse auf, welcher den Eingangsbuffer auf neue Befehle überprüft. Werden Strings gefunden, werden diese an die Funktion COMM_ParseCommand übergeben. Diese Funktion iteriert durch die Teile des Befehls. Wird ein Befehl erkannt, werden die relevanten Informationen wie Schrittzahl entnommen und entsprechende Reaktionen eingeleitet. Die Befehle von Master und Feeder werden in folgenden Tabelle aufgezeigt.

Tabelle 14 Masterbefehle

Kopf	Adresse	Befehl	Parameter	Beschreibung
CMD	-1	CNT		Anzahl Feeder zählen
CMD	xx	FWD	<Schrittzahl>	Vorwärts um x Schritte
CMD	xx	REV	<Schrittzahl>	Rückwärts um x Schritte

Tabelle 15 Feeder Meldungen

Kopf	Adresse	Befehl	Parameter	Beschreibung
CMD	-1	ERR	<msg>	Fehlerrückmeldung
CMD	-1	FWD	ok	Vorwärts-Schrittbefehl erhalten
CMD	-1	REV	ok	Rückwärts-Schrittbefehl erhalten
CMD	xx	CNT		Anzahl Feeder zählen (Adresse um 1 dekrementiert)

Wird ein Befehl während der Übertragung so verändert, dass er von den Feedern nicht verstanden wird, wird ein ERR Meldung generiert. Der Parameter des ERR Befehls ist in dem Fall der Inhalt des erhaltenen Befehls.

Beim Erhalt einen FWD oder REV Befehls wird die Anzahl Schritte aus dem Befehl entnommen. Der APP_Counter wird um diese Anzahl erhöht und ein FWD/REV ok gesendet. Beim nächsten Aufruf von dem Applications-Task (Kapitel 5.3.4) wird der APP_Counter überprüft und entsprechend gehandelt. Abbildung 58 beinhaltet die ein Ablaufdiagramm welches die Analyse der empfangenen Befehle in der Software darstellt.

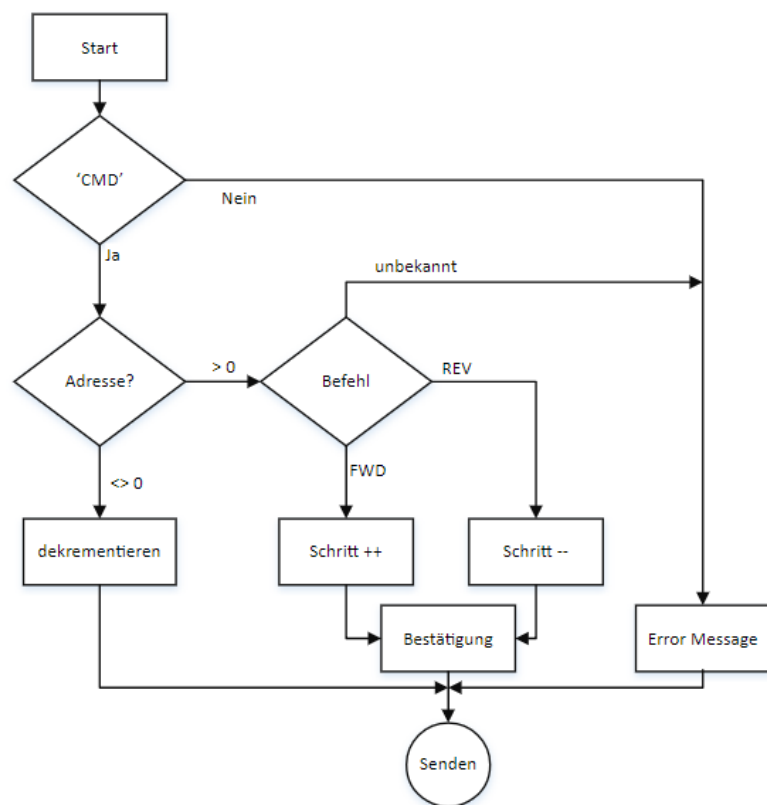


Abbildung 58 Ablaufdiagramm Befehlsanalyse

6 Validierung

In diesem Kapitel werden bestimmte Entwicklungen von Prototypen sowie Versuchsaufbauten beschrieben. Diese Informationen dienen zur Verdeutlichung und Validierung von dem Entwicklungsprozess sowie als Ergänzung weiterer Kapitel dieser Dokumentation.

6.1 Prototyp V0

Der erste Prototyp wurde hergestellt um die Grundideen des Feederkonzepts zu überprüfen. Dazu gehörten:

- Führung des Streifens aus 3D Druck
- Peeler mit 3D Druck Zahnräder
- Schneckengetriebe Streifenantrieb aus 3D Druck
- Sprocket aus PCB
- Schritterkennung mit Lichtschranke und Sprocket

Dieser Prototyp ist in Abbildung 59 erkennbar. Alle Teilfunktionen wurden einzeln hergestellt und in solcher Art, dass sie unabhängig voneinander platziert werden konnten. Alle obengenannten Ideen konnten in ihrer Funktion bestätigt werden. Dieser Prototyp wurde auch im Lichtschrankentest von Kapitel 6.4.1 verwendet. Die Höhe für den Streifen in der Führung war deutlich zu gross ausgelegt (3 mm), und wurde somit in folgenden Prototypen auf 1.2mm reduziert.

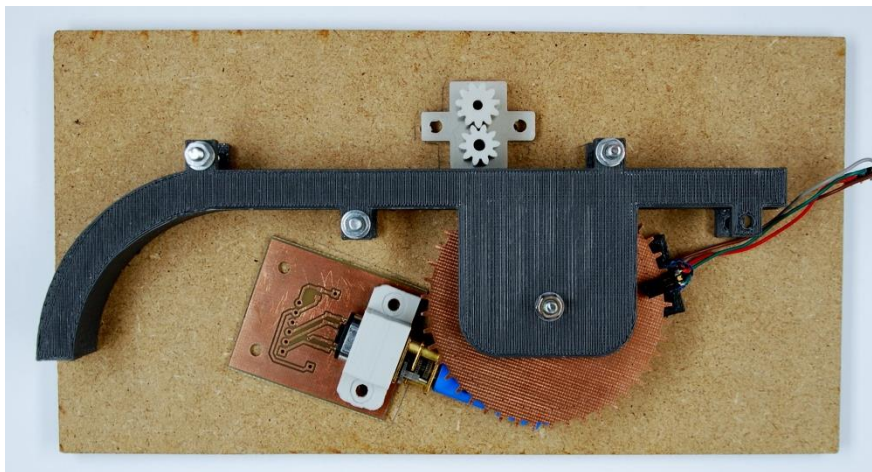


Abbildung 59 Prototyp V0

6.2 Prototyp V1

Der Prototyp V1 war der erste Prototyp, bei welchem PCB-Material für die Seitenwände verwendet wurde. Die Wände hatten jedoch kein eigenes elektrisches Layout. Die elektrischen Funktionalitäten wurden mit einem tinyK22, Test-PCBs und einem Steckbrett realisiert. Die KiCAD Projekte zum H-Brücken PCB sowie für den Encoder sind im elektronischen Anhang im Ordner KiCAD Projects abgelegt.

Mit diesem Prototyp konnte die Kombination von allen mechanischen Komponenten sowie die Softwarefunktionalitäten getestet werden. Dies führte nebst einigen kleinen Änderungen im Design zu drei Grösseren. Folgend werden in Kürze die Erkenntnisse zu den einzelnen Punkten erläutert.

6.2.1 Peeler

In diesem Prototyp wurde der Mikroschwitch in Hebelform getestet. Die ausführlichen Erkenntnisse sind in Kapitel 6.5.1 beschrieben. Zusammenfassend wurde erkannt, dass die Hebelfunktion nicht wie gewünscht funktioniert, und es wurde eine alternative Lösung entwickelt. Diese ist im Kapitel 6.5 beschrieben.

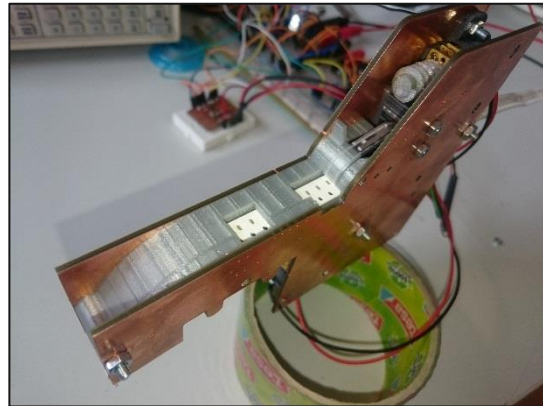
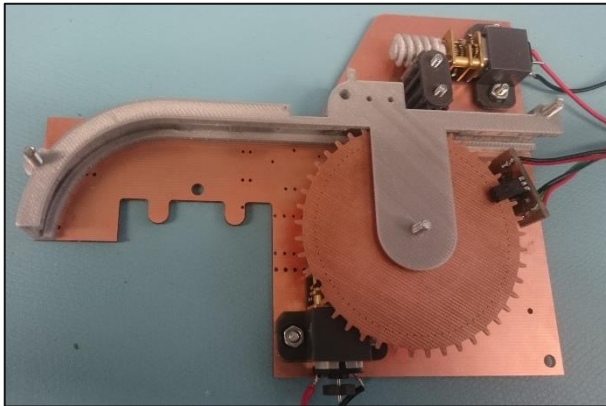


Abbildung 60 Prototyp V1

6.2.2 Führung und Antrieb

In diesem Prototyp wurde die Führung des Streifens gegenüber Version 0 verkleinert. Bei dieser Version wurde erkannt, dass wenn ein Papierstreifen nach der Pick-Location umgelenkt wird, eine deutlich höhere Reibung entsteht. Um den Streifen durch die Umlenkung zu stossen musste der Sprocket-Antrieb mehr Kraft aufwenden. Blockierte der Streifen, war es möglich, dass das Schneckenrad sich vom Zahnrad wegdrückte. Dieses Verhalten wird in Kapitel 5.1.4 beschrieben. Entsprechend musste ein Motorenhalter entwickelt werden, welches dieses Verhalten verhinderte. Zudem wurde die Führung des Streifens bei der Umlenkung vergrössert, damit der Streifen weniger klemmen konnte.

6.2.3 Lichtschranke und Elektronik

In diesem Prototyp wurde die Funktionalität der Lichtschranke weiter getestet (Kapitel 6.4.2). Zudem wurde erkannt, dass die Montage und Positionierung von diesem vereinfacht werden würde, wenn er direkt auf die PCB-Wand aufgelötet werden konnte. Damit wurde entschieden, die gesamte Elektronik und der Grossteil der mechanischen Komponenten auf derselben PCB-Wand zu montieren. Damit dies erreicht werden konnte musste die PCB-Wand vergrössert werden.

6.3 Prototyp V2 und V2.1

Der Prototyp V2 war der erste Prototyp, auf dem das Layout direkt auf die PCB-Wand gemacht wurde. In einer ersten Iteration wurde erkannt, dass der falsche Mikrokontroller-Footprint verwendet wurde. Die mechanische Auslegung der Wand konnte hingegen als korrekt angenommen werden. Die Version 2.1 hatte ein funktionsfähiges Layout, welches bestückt wurde. Mit diesem Prototyp wurde die elektrischen Funktionalitäten ohne den Test-PCBs getestet. Dabei wurde erkannt, dass die Montagerichtung der Lichtschranke viel Fremdlicht auf den Fototransistor lenkte. Somit wurde die Schritterkennung verschlechtert. Es wurde entschieden, die Lichtschranke zu drehen, damit der Fototransistor direkt unter dem Sprocket liegt, und somit kaum Fremdlicht eindringen könnte.

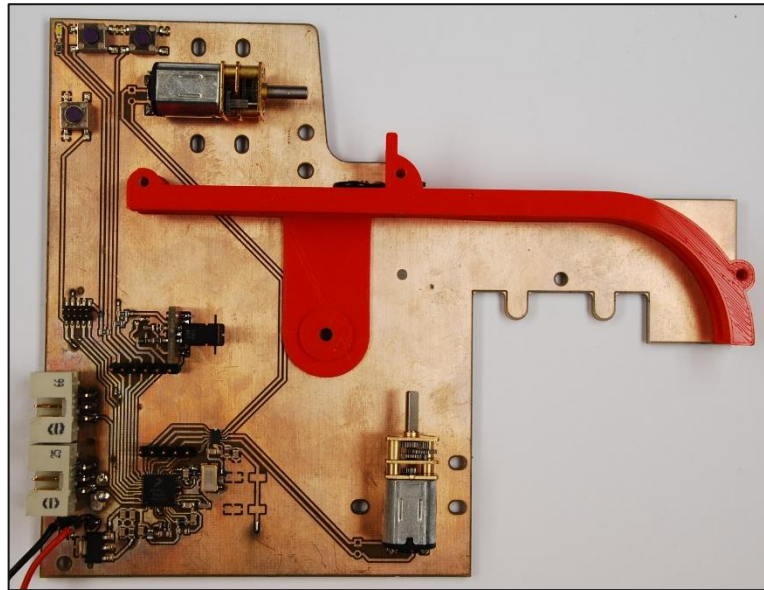


Abbildung 61 Bestückter Prototyp V2.1

Bei dieser Änderung des Footprints der Lichtschranke entstand ein Fehler, welcher erst in der bei PCBWay bestellten Version 3 bemerkt wurde. Eine Verbindung zwischen zwei GND-Pads wurde nicht gelöscht. Bei dem neuen Footprint verursachte dies eine Verbindung zwischen dem Ausgang des Fototransistors und GND. Entsprechend konnte das Ausgangssignal nicht genutzt werden. Dieser Fehler konnte jedoch mit dem Auftrennen dieser Leiterbahn mit einem Skalpell gelöst werden.

An diesem Prototyp wurde auch der Detektor-Schalter als Mikroschwitch getestet, welches in Kapitel 6.5.3 beschrieben wird. In diesem Zusammenhang wurden mehrere Meter 8 mm Streifen durch den Feeder geführt. Dabei wurde bei einem Versuch nach 2.3 Meter Streifen 1 Schrittfehler beobachtet. Die Präzision der Schritte wurde nur von Auge überwacht, jedoch konnte das Fazit gezogen werden, dass die 0805 Komponenten bis zum Schrittverlust alle von einem Bestückungskopf hätten aufgenommen werden können. Die 2.3 Meter resultieren bei einem Bauteilabstand von 4 mm in 575 Bauteilen. In weiteren Versuchen rund um die Kommunikation wurden Schritte Vor- und Rückwärts getestet, wobei äussert selten ein Schrittverlust festgestellt wurde. Diese Versuche wurden nicht in Zahlen festgehalten.

6.4 Lichtschranke

Die Validierung der Funktionalität der Lichtschranke war ein wichtiger Arbeitsschritt dieses Projekts. Die Lichtschranke ist eine Schnittstelle zwischen mechanischem Aufbau und elektrischen Komponenten, welche eine wichtige Rolle in der Software spielen.

6.4.1 Lichtschranken Signalmessung

Messung

Im ersten Prototyp wurde das Signal der Lichtschranke mit einem Sprocket mit Bohrungen von 0.2mm getestet. Das verwendete Oszilloskop war ein Hewlett-Packard 54645D, das Multimeter ein Fluke 87.

Zuerst wurde das Sprocket von Hand gedreht, um den durch Lichteinfall verursachten Strom am Transistorausgang der Lichtschranke zu messen. Der schematische Schaltungsaufbau ist in Abbildung 62 dargestellt. Durch diese Messung konnten Ströme im Bereich von 0.4 bis 0.7 mA gemessen werden. Somit wurde der Lastwiderstand des Fototransistors auf 4.7 kΩ festgelegt. Bei dem maximal gemessenen Strom von 0.7 mA würde eine Spannung von 3.3 VDC am Mikrokontroller-Pin anliegen.

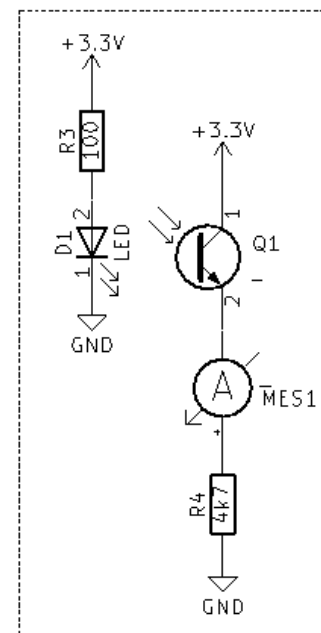


Abbildung 62 Schema Versuch 1

Im nächsten Schritt wurde das Sprocket über das Schneckengetriebe und DC-Getriebemotor angetrieben. Bei einer Ansteuerung des Motors mit 5 VDC konnten periodische Impulse von der Lichtschranke aufgezeichnet werden. Dies ist in Abbildung 63 ersichtlich. Die Einstellung des



Abbildung 63 Signal der Lichtschranke mit 0,2mm Sprocket

Oszilloskops war auf der Y-Achse 1 V/div und auf der X-Achse 200 ms/div. Die Impulse haben eine durchschnittliche Höhe von 2.1 VDC. Die Impulse haben eine durchschnittliche Zeitdauer von ca. 40 ms und eine Periode von 160 ms. Dabei gilt zu beachten, dass das Schneckengetriebe eine Übersetzung von 40:1 hatte. In weiteren Prototypen wurde ein Getriebe mit 30:1 eingesetzt. Dazu mehr in Kapitel 5.1.4. In der Abbildung 63 ist zudem ersichtlich, dass an einer Stelle kein deutlicher Impuls erzeugt wurde. Dies war auf Verunreinigungen in der Bohrung des Sprockets zurückzuführen.

Fazit

Aus diesem Messaufbau konnten mehrere Schlüsse gezogen werden.

- Die Kombination von Lichtschranke und PCB-Sprocket mit Bohrungen funktioniert und kann für das Feederkonzept verwendet werden.
- Verunreinigungen in den Bohrungen können das Signal verfälschen und einen Schrittverlust verursachen.
- Die Flanken des Signals steigen mit dem Lichteinfall an, respektive auch wieder ab. Es wird ein Dreieck-Förmiges Signal erzeugt.
- Die Zeitdauer der Impulse (40 ms) erlaubt es, dass der Mikrokontroller das Signal abtasten kann und somit keine Interrupts mit Flanken-Detektion eingesetzt werden müssen.
- Die Impulshöhe ist nicht konstant, und die Flankensteigung ist auch abhängig von dem Lichteinfall. Solche Toleranzen könnten zu einer Verminderung der Signalqualität führen.
- Die Positionierung der Lichtschranke gegenüber der Sprocketbohrungen beeinflusst die Menge des einfallende Lichtes stark. Grössere Bohrungsdurchmesser sollen hier Toleranzen der Montagesposition ermöglichen.

6.4.2 Transistor und Komparator

Aus der Messung des Schrittsignals der Lichtschranke in Kapitel 6.4 ging hervor, dass die Impulshöhe sowie die Flanken der Impulse unterschiedlich waren. Damit die Abtastung des Signals durch den Mikrokontroller möglichst fehlerfrei und genau geschehen kann, muss das Signal verbessert werden. Somit wurde entschieden, den Einsatz eines weiteren Transistors sowie eines Komparators zu überprüfen. Das verwendete Oszilloskop war das MSO6052A von Agilent Technologies.

Messaufbau Transistor

In einem ersten Messaufbau wurde ein NPN Transistor (BC337) eingesetzt, welcher von dem Fototransistor der Lichtschranken geschaltet wurde. Der BC337 schaltet dann den Logikpegel als Signal, welches vom Mikrokontroller eingelesen werden würde. An dieser Stelle wurde ein Oszilloskop angeschlossen, und das Signal überprüft. Das Schema des Schaltungsaufbaus ist in Abbildung 64 dargestellt, das resultierende Signal in Abbildung 65.

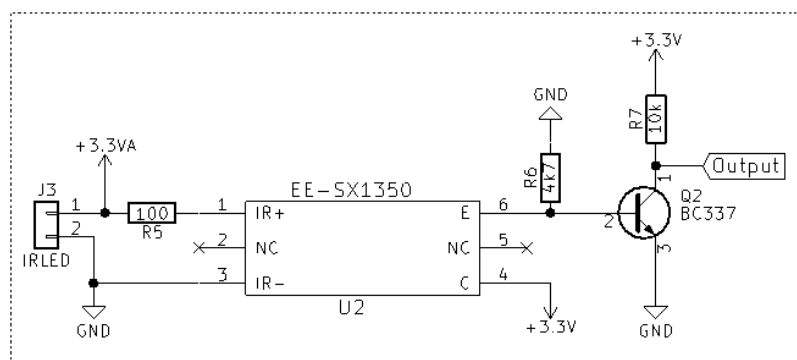


Abbildung 64 Schema Versuch 2: Zusätzlicher Transistor

Aus der Aufzeichnung (Abbildung 65) können die Flanken des Signals von dem Fototransistor (grün) gut mit demjenigen des zusätzlichen BC337 Transistors (orange) verglichen werden. Bei einer Schwelle von 0.65 VDC schaltet der BC337 durch. Von GND bis zum Hoch-Pegel von 3.3 VDC steigt das Signal in 1 ms an. Im Vergleich zum Signal des ersten Versuches (Abbildung 63), wird mithilfe des BC337 ein rechteck-förmiges Signal erzeugt. Das Rauschen welches in den Signalen ersichtlich ist, ist auf den offenen Messaufbau zurückzuführen.

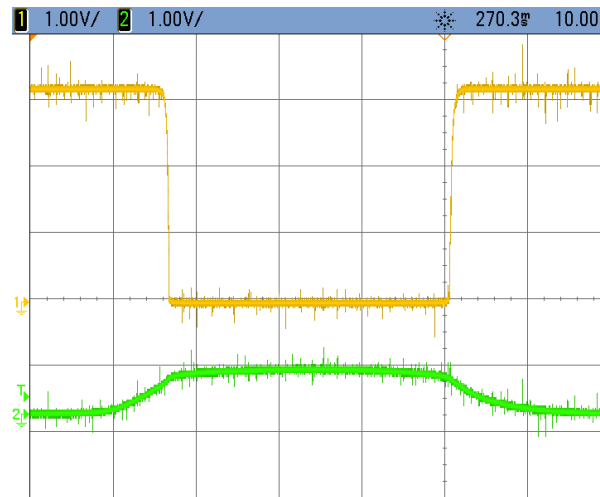


Abbildung 65 Schrittsignal mit Zusatz-Transistor

Messaufbau Komparator

Der obige Messaufbau wurde in einem weiteren Schritt mit einem Komparator erweitert. Komparatoren haben die Eigenschaft, sehr steile Flanken zu erzeugen. In einem ersten Aufbau wurde der Komparator direkt an den Ausgang des Fototransistors angeschlossen. Jedoch wurde festgestellt, dass, wie in Abbildung 66 dargestellt, bei abfallender Flanke des Fototransistors der Komparator prellt. Somit wäre keine Verbesserung des Signals möglich. Die Umsetzung einer Hysterese würde ein solches Verhalten unterdrücken. Jedoch kann keine Hysterese zwischen Fototransistor und Komparator direkt einfach umgesetzt werden. Die zusätzlichen Widerstände der Hysterese beeinflussen den Lastwiderstand am Fototransistor und der erzeugte Spannungspegel wird verändert. Somit müsste zwischen Fototransistor und Komparator eine trennende Komponente wie ein Transistor eingesetzt werden. Es folgte ein neuer Messaufbau.

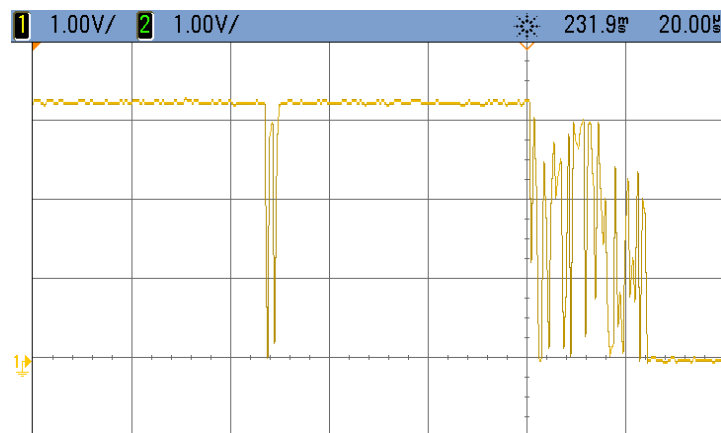


Abbildung 66 Signalverhalten Komparator an Fototransistor

Messaufbau Komparator & Transistor

Die Schaltung wurde nach Abbildung 67 aufgebaut und das Signal des Transistors und des Komparators verglichen. Das Signal wird in Abbildung 68 dargestellt. Bei einer Auflösung von 200 $\mu\text{s}/\text{div}$ ist deutlich zu erkennen, dass der Komparator (grün) schneller schaltet als der BC337 Transistor (orange).

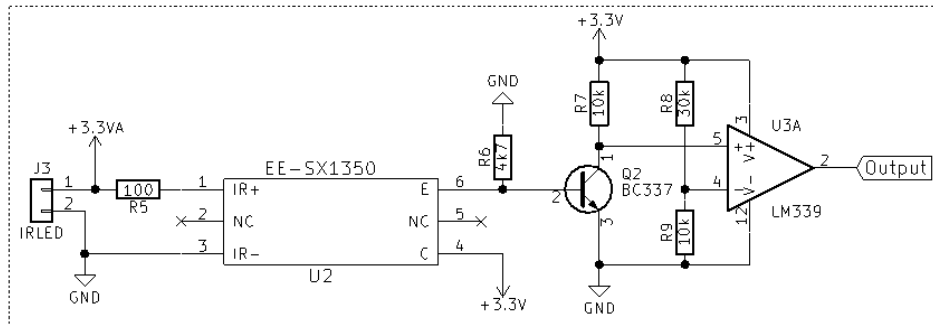


Abbildung 67 Schema Versuch 3: Transistor & Komparator

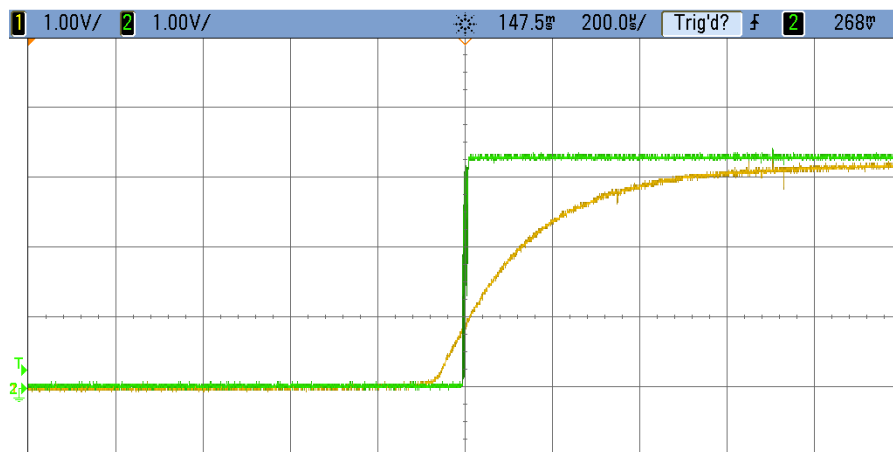


Abbildung 68 Vergleich Transistor und Komparator

Signalmessung

Abbildung 69 zeigt ein Signal über einen Weg von 8 mm, mit dem Einsatz von einem zusätzlichen Transistor. Das Signal hat klare Flanken, und die Perioden des Signales sind deutlich zu erkennen. Der Zustand des Signals kann in dieser Form gut von einem Mikrokontroller abgetastet werden. Das Rauschen des Signals kann dem Messaufbau zugeschrieben werden. In dieser Messung wurde ein Schneckengetriebe mit einer Übersetzung von 30:1 verwendet, zudem haben die Bohrungen im Sprocket einen Durchmesser von 0.6 mm. Dies widerspiegelt sich in dem zeitlichen Verhalten des Signales. Die Periodendauer ist ca. 140 ms, wobei der Lichteinfall ca. 40 ms dauert.

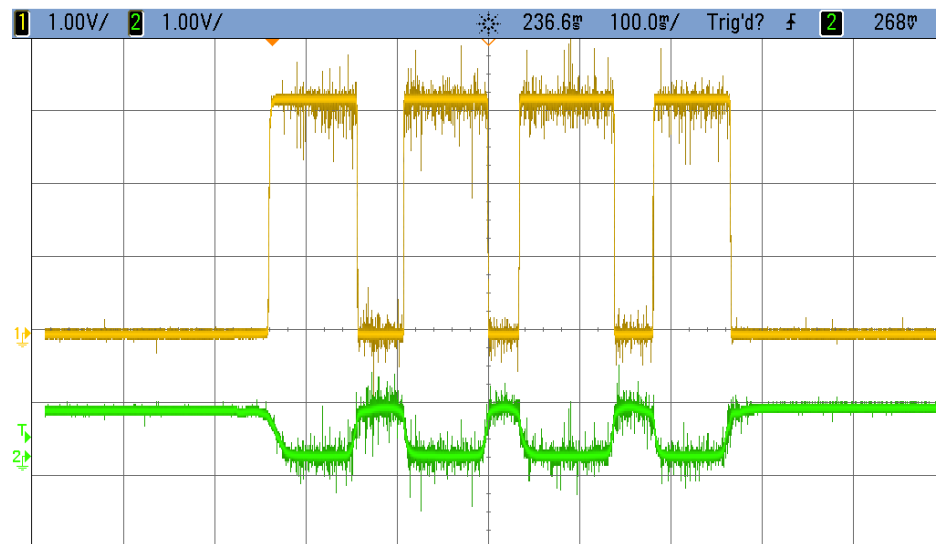


Abbildung 69 8 mm Schrittmessung

Fazit

Durch die obigen drei Messaufbauten wurde der Nutzen eines zusätzlichen Transistors, respektive auch Komparators ersichtlich. Das Erzeugen steiler Flanken und dabei auch den Logikpegel sauber zu schalten ermöglichen es das Signal mit dem Mikrokontroller angemessen fehlerfrei abzutasten.

Zum Zeitpunkt dieser Messdurchführung war bereits bekannt, dass für die Umsetzung der Software ein FreeRTOS eingesetzt werden würde. Der Task, welcher die Lichtschranke abtastet, läuft mit einer Frequenz von 1 kHz. Somit würde eine Abtastung im 1 ms Takt geschehen. Mit dem Signal in Abbildung 68 wird deutlich, dass der Komparator überflüssig wird. Der BC337 Transistor benötigte ca. 300 μ s um über eine Schwelle von 2.3 VDC zu gelangen. Ab diesem Spannungspegel erkennt der Mikrokontroller einen Hoch-Pegel (NXP, 2012, S. 11). Dies resultiert somit im Vergleich zur Abtastrate in einem vernachlässigbaren Fehler.

6.5 Mikroschwitch

In diesem Kapitel werden die getesteten Varianten des Mikroswitches vorgestellt.

6.5.1 Hebel-Endschalter

In der Darstellung des Peelers in Abbildung 31 wurde mit grün gestrichelter Linie ein Mikroswitch mit Hebel-Arm eingezeichnet. Im ersten Prototyp wurde dieser verwendet, um das Anliegen der Folie anzugeben. In Abbildung 72 unten ist die gedruckte Führung mit montiertem Mikroswitch erkennbar. In den Testläufen mit diesem Prototyp wurde ersichtlich, dass dieser Aufbau nicht wunschgemäss funktionierte.

Bei einem Vorwärtsschritt wurde die Folie nur nach vorne gezogen, und folglich blieb der Mikroswitch betätigt. In der Software konnte so nicht unterschieden werden ob der Peeler starten musste oder nicht. Abbildung 70 verdeutlicht das Verhalten der Abdeckungsfolie im Prototypen. Die rote gestrichelte Linie stellt die Abdeckungsfolie im angezogenem Zustand dar. Sie wird über den Mikroswitch (dunkelgrau) geführt und zwischen den Peeler-Zahnradern (hellblau) eingeklemmt. Die blaue gestrichelte Linie stellt die Folie nach einem vorwärtsschritt dar. Über dem Mikroswitch behält sie dieselbe Position, wie rot dargestellt. Entsprechend ist nicht bekannt, dass die Folie nachgezogen werden müsste. Mit diesem Aufbau wurde erkannt, dass die Position, und daraus auch die Art, des Mikroswitches geändert werden musste. Das Fehlverhalten des Peeler ist auch im Bild von Abbildung 71 erkennbar. Die Folie wurde mit der Führung bis zur Pick-Location gezogen, da der Mikroswitch stets betätigt war.



Abbildung 71 Fehlfunktion Peeler

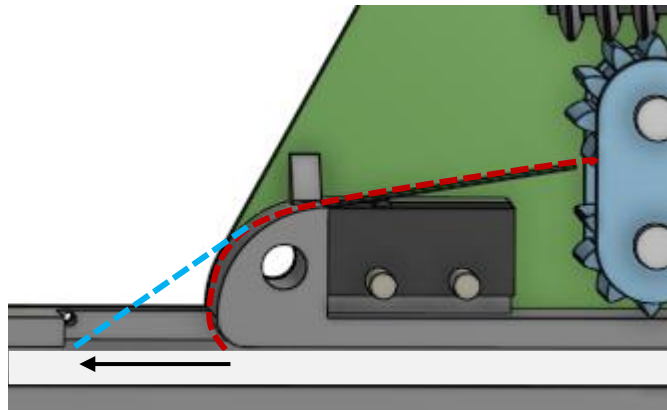


Abbildung 70 Peeler mit Hebel-Endschalter

6.5.2 Knopf-Taster

Mithilfe des ersten Testaufbaus wurde entschieden, den Mikroswitch neu zu platzieren. Er sollte in dem Bereich montiert werden, wo eine Änderung festgestellt werden konnte, folglich dort wo die Folie einen Weg macht. Um die Funktionalität zu testen wurde kurzerhand ein Knopftaster an der Stelle montiert. Dies ist in folgender Abbildung in der oberen Führung erkennbar.

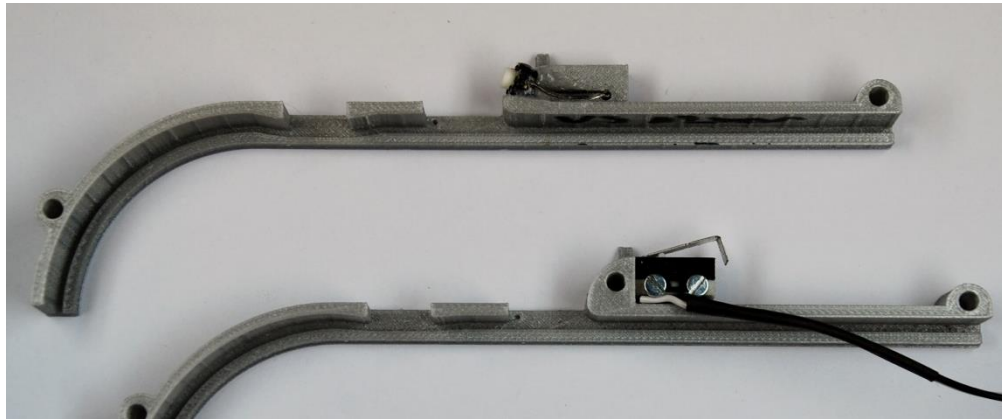


Abbildung 72 Hebel- und Knopftaster Prototypen Mikroswitch

Dieser Versuchsaufbau zeigte, dass eine Montage an der Stelle die gewünschte Funktionalität mit sich bringen konnte. Der verwendete Taster wurde jedoch nicht zuverlässig von der Folie betätigt. Die benötigte Druckkraft des Tasters konnte von der dünnen Folie nicht in jedem Fall ausgeübt werden. Zudem war es möglich, dass die Folie seitlich am Taster vorbeiging.

6.5.3 Detektorschalter

Da die Grundfunktionalität mit dem Knopf-Taster bestätigt wurde, jedoch die Kraft nicht ausreichte, wurde ein neuer Taster gesucht. Die Suche führte zu kleinen Endschaltern, welche nur eine sehr geringe Kraft zur Betätigung benötigen. Diese Detektor Taster werden beispielsweise in CD-Laufwerken eingesetzt, um zu erkennen ob er aus oder eingefahren ist. Abbildung 73 zeigt eine Führung mit dem eingesetzten Detektorschalter ESE-22MH57 von Panasonic. Die Peelerfunktion wurde mit diesem Schalter wieder überprüft und bestätigt. Ein leichter Druck der angezogenen Abdeckungsfolie reicht aus um den Schalter zu betätigen. Um zu gewährleisten, dass die Folie stets über dem Mikroswitch geführt wird, wurden auch die Nocken der Folienführung vergrößert.

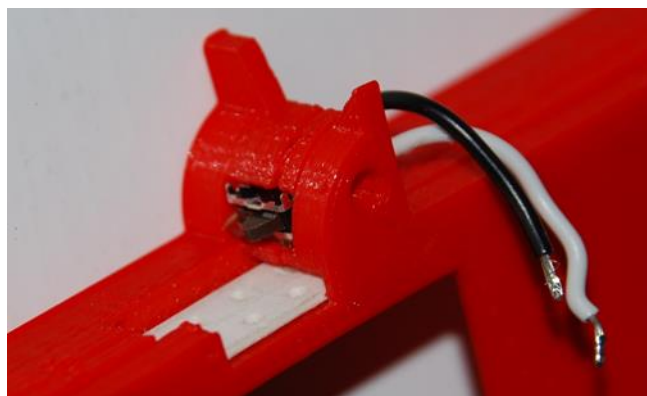


Abbildung 73 Detektorschalter

6.6 Kommunikation

Die Kommunikation ist ein relevanter Teil dieses Projekts und die Funktionalität musste nach Erstellung des Grundkonzepts überprüft und bestätigt werden.

6.6.1 tinyK20 als Feeder

In einem ersten Versuchsaufbau wurde die Daisy-Chain Kommunikation mit vier tinyK20 getestet. Von einem Terminal Programm auf einem Computer wurden die definierten Befehle (Kapitel 5.3.6) an ein tinyK20 über eine UART Schnittstelle gesendet, welches die Befehle an das nächste tinyK20 weiterleitete. Diese Übergabe erfolgte durch die vier tinyK20 bis die Befehle wiederum am Computer eingelesen wurde. Der Computer stellte in diesem Aufbau das Master-Device dar.

Beispielsweise wurde ein FWD-Befehl vom Computer (Master) gesendet.

Master:	CMD 1 FWD 2	→ Feeder 2 soll zwei Schritte vorwärts machen
Feeder 1:	CMD 0 FWD 2	→ Dekrementieren der Adresse
Feeder 2:	CMD -1 FWD ok	→ erhält Befehl mit Adresse 0, Statusrückmeldung i.O.
Feeder 3:	CMD -2 FWD ok	→ Dekrementieren der Adresse
Feeder 4:	CMD -2 FWD ok	→ Dekrementieren der Adresse

Am Computer wurde über das Terminalprogramm die Meldung von Feeder 4 angezeigt.

Fazit

Dieser Aufbau ermöglichte es die Funktion des Daisy-Chains und deren Software-Implementation zu bestätigen. Dieser Aufbau wurde wiederholt verwendet um Änderungen an den Kommunikationsbefehlen zu testen. Während alle Versuchen dieser Art wurden keine Störungen über die UART Kommunikation festgestellt.

6.6.2 tinyK22 als Master

Nach dem erfolgreichen Testen des Daisy-Chains mit den vier tinyK20 wurde von Prof. Styger ein Master für die Feeder erstellt. Diese Software läuft auf einem tinyK22, welche die Kommunikationsschnittstelle zwischen OpenPnP und den Feedern bildet.

In einem Versuchsaufbau wurde eine Kommunikation zwischen einem Terminalprogramm auf einem Computer, dem tinyK22 und einem tinyK20 aufgebaut. Auf dem tinyK22 wurde der Mastercode geladen, auf dem tinyK20 der Feedercode.

Der Master erhält vom OpenPnP derselbe Befehl, welcher für die Feeder verwendet wird, jedoch ist diesem voran noch die Präambel «feeder send» angefügt. Entsprechend sehen die Meldungen bei einem Vorwärtsbefehl wie folgt aus.

OpenPnP:	feeder send CMD 1 FWD 2	→ Feeder 2 soll zwei Schritte vorwärts machen
Master:	CMD 1 FWD 2	→ Feeder 2 soll zwei Schritte vorwärts machen
Feeder 1:	CMD 0 FWD 2	→ Dekrementieren der Adresse
Feeder 2:	CMD -1 FWD ok	→ erhält Befehl mit Adresse 0, Statusrückmeldung i.O.
Feeder 3:	CMD -2 FWD ok	→ Dekrementieren der Adresse
Feeder 4:	CMD -2 FWD ok	→ Dekrementieren der Adresse

Der Master auf dem tinyK22 erhält die Meldung von Feeder 4 und leitet die Bestätigung an das OpenPnP weiter.

6.7 Energieversorgung

Die Stromaufnahme wurde bei mehreren Feedern untersucht. Die Annahme besteht, dass durch den selben Aufbau sich die Stromaufnahme von Feeder zu Feeder vernachlässigbar unterscheidet. Dieses Verhalten wurde durch die Messung bestätigt.

Messung 1

Für die Strommessung wurde ein Fluke 183 Multimeter verwendet. Dieses wurde in Serie mit der Spannungsversorgung zum Feeder verbunden. Die Versorgungsspannung wurde durch ein Labornetzteil HM8040-3 von Hameg geliefert. Diese lag bei 5.02 VDC.

Folgende Werte sind durchschnittliche Werte, welche beobachtet wurde. Dabei gilt zu beachten, dass durch die Anlaufströme der Motoren kurzzeitig höhere Ströme auftreten können. Zudem sind die Sprocket-Motoren durch die Software-Implementation impulsartig angesteuert. Dies bewirkt, dass der gemessene Strom über mehrere Schrittbefehle keinen konstanten Charakter hat. Des Weiteren können Unterschiede zwischen dem Vorwärts- und Rückwärtsbetrieb festgestellt werden. Dies weitestgehend Aufgrund des mechanischen Aufbaus des Feeders.

Tabelle 16 Situations-Strommessung

Situation	Stromaufnahme [mA]
Logik (Mikrokontroller, Lichtschranke, H-Brücke)	42.5
Gesamte Logik (mC, Lichtschranke, Status-LED, H-Brücke)	55
Logik & Streifenantrieb (vorwärts)	115
Logik & Peeler-Antrieb	78
Logik & Beide Motoren (leichte Last)	160
Logik & Beide Motoren (grosse Last)	410

Als leichte Last kann der normale Betrieb des Feeders angenommen werden, wobei der Streifen keine Klemmwirkung durch eine Verdrehung oder Ähnlichem erfährt. Der Streifen geht 'leicht' durch die Führung. Bei grosse Last wurde das Gegenteil simuliert. Der Streifen wurde von Hand festgehalten und somit eine Vorwärtsbewegung verhindert. Somit wurde der Antriebsmotor blockiert und der Strom vom Kippmoment konnte gemessen werden. Der Peelermotor wurde nicht blockiert. Im Falle, dass die Abdeckungsfolie nicht weiter eingezogen werden kann, wurde beobachtet, dass diese einfach reisst oder dass sie aufgrund vom Schlupf der Zahnräder gespannt bleibt.

Im Rahmen dieser Arbeit wurden vier Feeder erstellt. Je ein Feeder für die in der Zielsetzung definierten Streifenbreiten von 8, 12, 16 und 24 mm. In einer weiteren Messung wurden die Feeder miteinander verglichen. Auch hier werden durchschnittliche Werte der Logik und der Motoren bei leichter Last präsentiert.

Tabelle 17 Strommessung aller Feeder

Feeder	Stromaufnahme Logik [mA]	Stromaufnahme mit Motoren [mA]
8 mm	42.6	125
12 mm	43.1	110
16 mm	43.8	129
24 mm	42.6	116

Fazit

Der elektrische Aufbau ist bei allen Feedern gleich, deswegen wurde keine grossen Schwankungen erwartet. Die Messungen bestätigten diese Annahme.

Messung 2

In dieser Messung wurde die zeitliche Stromaufnahme des 8 mm Feeders untersucht. Hier ging es hauptsächlich darum den Anlaufstrom der Motoren aufzuzeichnen. In der ersten Messung konnten nämlich diese impulsartigen Ströme nicht zuverlässig festgehalten werden.

Für diese Messung wurde ein Shunt-Widerstand (R1) von 0.2 Ohm in die Spannungsversorgungsleitung eingebunden. Mit einem Oszilloskop wurde die Spannung über diesen Widerstand gemessen und konnte somit zurückgerechnet werden. Dieser Messaufbau verursacht einen Spannungsfehler in der Messung. Die Eingangsspannung am Feeder sinkt um den Spannungsabfall über den Shunt. Der Spannungsabfall bleibt durch den kleinen Widerstand des Shunt-Widerstandes jedoch verhältnismässig klein, und verursacht am Feeder keinen Nennenswerten betriebsunterschiede.

Für die Spannungsversorgung wurde ein Labornetzteil HM8040-3 von Hameg verwendet. Die Spannung lag bei 4.985 Volt. Das verwendete Oszilloskop war ein MSO6052A von Agilent Technologies.

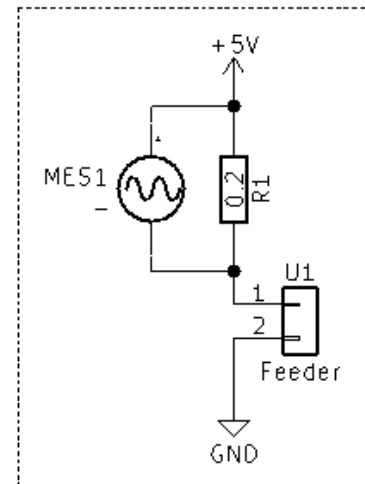


Abbildung 74 Schema Strommessung

Die Auflösung lag bei 20 mV/div, entsprechend ist auch Rauschen auf den Messaufnahmen erkennbar. Dieses wurde durch das Netzteil sowie dem allgemeinen Messaufbau verursacht.

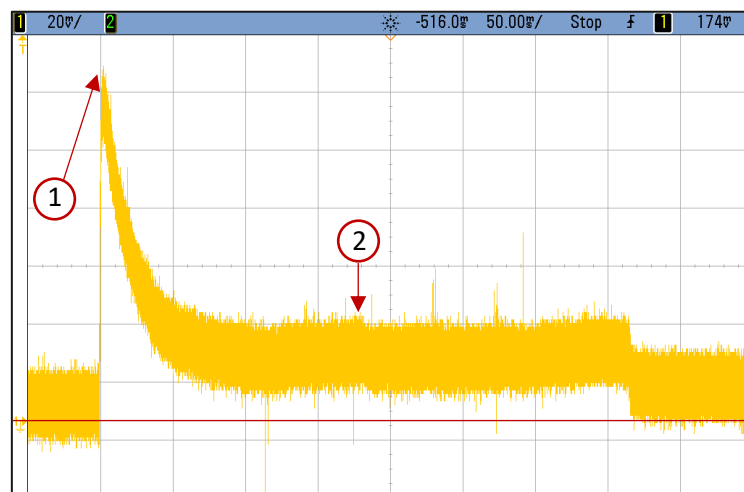


Abbildung 75 Zeitliche Stromaufnahme (leichte Last)

In Abbildung 75 ist die Stromaufnahme mit leichter Last ersichtlich. Die rot eingezeichnete Linie unten stellt 0 VDC dar. Beide Motoren laufen gleichzeitig an und erzeugen einen Spannungsabfall über den Shunt-Widerstand von 120 mV. Dies ist bei Punkt 1 ersichtlich. Mithilfe des ohmschen Gesetzes kann dies in einen Strom umgerechnet werden.

$$I = \frac{U}{R} = \frac{0.12 \text{ V}}{0.2 \Omega} = 0.6 \text{ A}$$

Nachfolgend sinkt der Strom, da die Motoren drehen. Bei Punkt zwei wurde der Strom wiederum entnommen, da dies einen mittleren Wert darstellt. Der Spannungsabfall über R1 ist ca. 25 mV.

$$I = \frac{U}{R} = \frac{0.025 \text{ V}}{0.2 \Omega} = 0.125 \text{ A}$$

Dieser errechnete Strom entspricht den Beobachtungen aus der ersten Strommessung.

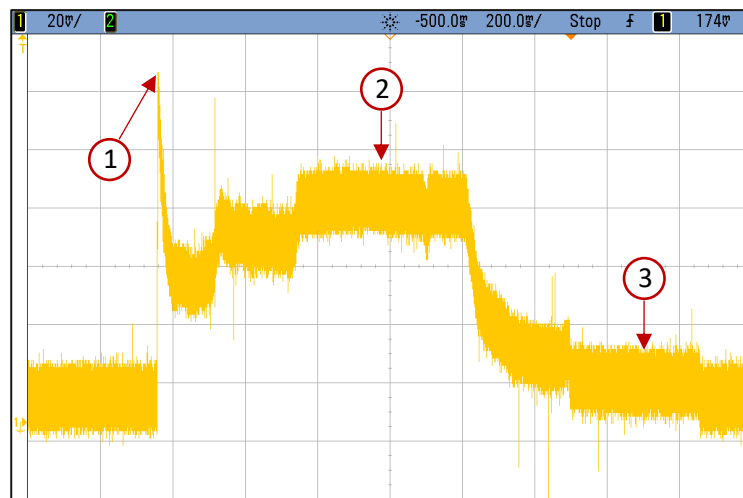


Abbildung 76 Zeitliche Stromaufnahme (grosse Last)

In Abbildung 76 ist die Stromaufnahme bei simulierter grosse Last ersichtlich. Der Streifen wurde von Hand festgehalten. Der Anlaufstrom ist wiederum bei Punkt 1 ersichtlich, und entspricht dem von der leichten Last. In dem zeitlichen Verlauf ist ersichtlich wie der Strom durch das Blockieren des Sprocket-Antriebs anstieg. Bei Punkt zwei ist der Spannungsabfall 80 mV. Dies ergibt nach obiger Formel einen Strom von 400 mA, was wiederum den Werten aus der ersten Strommessung entspricht. Bei Punkt drei läuft nur noch der Peelmotor. Der Sprocket-Antrieb wurde kurz vor dieser Stelle nach dem Erreichen der Schritte ausgeschaltet. Die Stromaufnahme vom Betrieb mit nur dem Peelmotor ist umgerechnet 90 mA (bei einem Spannungsabfall von 18 mV).

Fazit

Der Anlaufstrom konnte mit dieser Methodik gut aufgezeichnet und verdeutlicht werden. Die errechneten Stromwerten aus den aufgezeichneten Spannungsdaten entsprachen den Werten aus der ersten Strommessung. Somit konnten diese bestätigt werden. Der Anlaufstrom ist etwas höher als bei blockiertem Sprocket-Antrieb, dies ist darauf zurückzuführen, dass der Peelmotor an dieser Stelle auch aus dem Stillstand anlaufen muss. Dieser Wert ist relevant für die Auslegung der Energieversorgung, da die Versorgung diesen Strom kurzzeitig liefern können muss. Aus dem Datenblatt der Motoren (Pololu, 2018a) ist der Strom bei Blockade des Motors mit maximal 1.6 A angegeben. Die gemessenen Werten liegen deutlich unter diesem. Zum einen sind die Werte aus dem Datenblatt errechnet, zum anderen ist die Versorgungsspannung in dieser Messung tiefer als die vorgegeben 6 V. Dies beeinflusst das Drehmoment des Motors, und entsprechend auch den aufgenommenen Strom.

6.8 Benutzerfreundlichkeit

Im Rahmen einer inoffiziellen Umfrage bei Mitstudenten im Bereich Elektrotechnik an der HSLU wurde die Funktionalität eines Feeders besprochen. Thema dabei war die Benutzerfreundlichkeit des Feeders. Nach einer kurzen Erklärung konnten die Mitstudierende den Feeder manuell betätigen, ein Streifen in den Feeder führen und diesen einrichten. Die Kommunikation mit OpenPnP wurde beschrieben, da keine Einrichtung von dem System vorhanden war.

Positive Rückmeldungen der Mitstudierenden betrafen:

- Die einfache Bedienung über den beiden Tastern.
- Die Status-LED welche die Schritterkennung übermittelte.
- Die einfache Montage auf dem Aluminiumprofil
- Der Verbindungen mit Flachbandkabel

Negative Bewertungen wurden zu diesen Punkten gegeben:

- Die benötigte Fingerfertigkeit um die Abdeckungsfolie einzufädeln.
- Die Geräusche des Feeders während eines Schrittes
- Die Platzierung der Verbindungsstecker unterhalb des Streifens

Zusammenfassend kann bestätigt werden, dass der einfach zu bedienen ist. Die negativen Rückmeldungen werden aufgenommen und allenfalls in Optimierungen des Feeders einfließen.

7 Diskussion

7.1 Projektmanagement

7.1.1 Projektplan

Der detaillierte Projektplan ist im Anhang abgelegt. In diesem Projektplan wurden am Anfang des Projekts die auszuführenden Tasks definiert und diesen Zeitslots zugewiesen. Dabei wurden 5 Meilensteine in regelmässigen Abschnitte festgelegt, bei denen definierte Arbeitspakete abgeschlossen werden sollten.

- **Meilenstein 1:** Technologierecherchen abgeschlossen und Funktionsweise definiert.
- **Meilenstein 2:** Grundfunktionen im Einzelnen getestet, erste mechanische Prototypen und Test-PCB für Hardware
- **Meilenstein 3:** Mechanik und Elektronik kombiniert getestet, PCB Layout für Bestellung abgeschlossen. Software Grundfunktionen umgesetzt, Kommunikation in Testaufbau bestätigt
- **Meilenstein 4:** Bestelltes PCB bestückt und getestet, Mechanik fertig und in End-Testphase, Feederbank gebaut. Master gebaut und getestet.
- **Meilenstein 5:** Projektabschluss. Feederkonzept umgesetzt: mechanischer Aufbau für alle Streifenbreiten erstellt, Software- & Kommunikationsfunktionen getestet und bestätigt.

Da die Entwicklung des Feeders etliche Teilaufgaben beinhaltete, welche zudem stark voneinander abhängig waren, war es schwierig den Zeitplan konkret einzuhalten. Zudem wurden laufend Änderungen an dem mechanischen Aufbau, insbesondere der Führung, vorgenommen, welche etliche Zeit in Anspruch nahm. Nichtsdestotrotz konnten die Meilensteine erreicht werden und das Konzept bis zum Projektabschluss funktionsfähig umgesetzt werden.

7.1.2 Risikomanagement

Die Risikoanalyse wurde am Anfang des Projekts erstellt und laufend erweitert. Die Risikoanalyse ist im elektronischen Anhang im Ordner Projektmanagement abgelegt. Bei der Risikoanalyse wurde versucht, mögliche Risiken zu erkennen und geeignete Massnahmen zu treffen, um diesen entgegenzuwirken. Die Risiken sind nach Wahrscheinlichkeit und Schadenshöhe von 10% bis 100% eingestuft. Zudem wurde Massnahmen definiert, welche die Wahrscheinlichkeit, respektive Schadenshöhe minimieren sollten. Die Risikomatrix wird in Abbildung 77 präsentiert, und darauffolgend einige Risiken beschrieben.

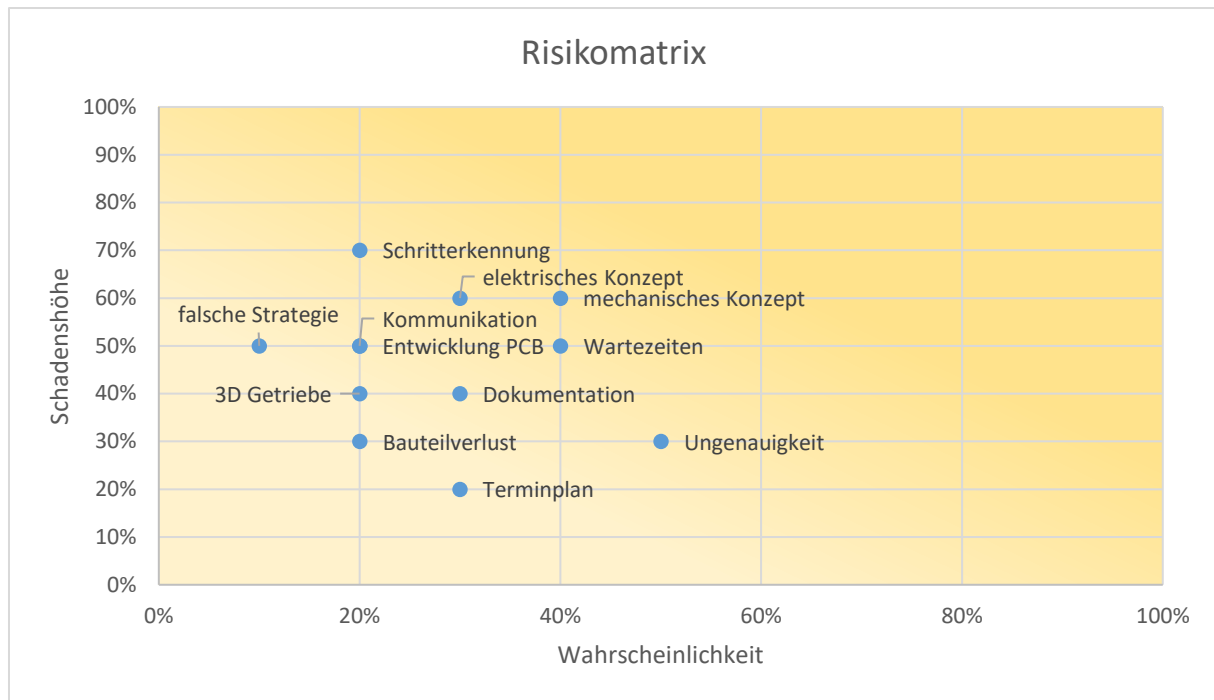


Abbildung 77 Risikomatrix

Dokumentation

Das Risiko Dokumentation bezieht sich darauf, dass die Dokumentation nicht vernachlässigt wird. Als Gegenmassnahme soll ein wöchentliches Arbeitsjournal, mit Terminerinnerungen im Kalender, geführt werden. Mithilfe von diesem können wichtige Ideen, Details und Arbeitsschritte festgehalten werden.

Schritterkennung

Bei diesem Risiko geht es darum, dass die Schritte mit der gewählten Lösung von Sprocket und Lichtschranke nicht erkannt werden, und es entsprechend einen Schrittverlust gibt. Entsprechend wurde am Sprocket der Bohrungsdurchmesser relativ gross gewählt, und ausführliche Tests durchgeführt.

Ungenauigkeit

Dieses Risiko betrifft der mechanische Aufbau des Feeders. Die mechanischen Komponenten wie die verschiedenen Getrieben an Motoren und aus 3D Druck, sowie der selbst erstellte Sprocket haben in der Fertigung Toleranzen. Diese können sich auf die Schrittweite auswirken und somit eine unregelmässige Positionierung des Bauteils zur Folge haben. Als Gegenmassnahme soll eine möglichst hohe Druckqualität verwendet werden, und während der Montage mögliche Abstände zwischen Zahnradern eliminiert werden.

7.1.3 Arbeitsjournal

Während dem gesamten Entwicklungsprozess wurde ein Arbeitsjournal gehalten. Darin wurde Arbeitsschritte festgehalten, sowie Stichwortartig wichtige Erkenntnisse aus Recherchen, Versuchen und Sitzungen. Am Anfang des Projekts wurden darin auch Überlegungen zu Teilfunktionen und Aufgaben festgehalten. Das Arbeitsjournal ist im elektronischen Anhang im Ordner Projektmanagement abgelegt.

7.2 Auswertung Ziele

In Kapitel 1.3 wurden Ziele definiert, welche es mit der Umsetzung dieses Projekts zu erfüllen galt. In diesem Kapitel wird ausgewertet, ob die Ziele erreicht wurden.

Benutzerfreundlichkeit

Die Benutzerfreundlichkeit wurde im Rahmen von Kapitel 6.8 überprüft. Die Rückmeldungen von Mitstudierenden zur Einrichtung und Bedienung des Feeders fielen grösstenteils positiv aus. Jedoch können keine Angaben zur Einrichtung der Feeder im OpenPnP-System gemacht werden. Da die Einbindung der Feeder in OpenPnP von Prof. Styger durchgeführt wird, wird dieser Punkt in der Zielauswertung nicht beachtet. Entsprechend kann bestätigt werden, dass das Ziel der Benutzerfreundlichkeit erreicht wurde.

Open-Source

Für die Elektronik und Mechanik wurden KiCAD, respektive OnShape verwendet. Beide gibt es als kostenlose Programme, welche von jedermann verwendet werden können. Das CAD-Projekt ist über OnShape öffentlich zugänglich. Die einzelnen Komponenten werden zudem auf GitHub zusammen mit den PCB-Dateien, Software und Dokumentation der Öffentlichkeit zugänglich gemacht. Dies wird von Prof. Styger gemacht.

Alle Komponenten sind entweder selbst herstellbar oder bei einem Lieferanten erhältlich. Ein Grossteil der mechanischen Komponenten kann im 3D Druckverfahren hergestellt werden und hält somit auch die Kosten eines Feeders niedrig. Zu allen selbst erstellten Komponenten sind Daten vorhanden, welche die Weiterentwicklung und Produktion des Feeders ermöglichen.

Streifenbreiten

Es wurden Feeder für 8, 12, 16 und 24 mm Streifenbreiten entwickelt. Für die Erweiterung auf breitere Streifen müssen nur Teile des mechanischen Aufbaus angepasst werden. Die PCB-Wände und Elektronik können übernommen werden. Dieses Ziel gilt als erfüllt.

Bauteilgrösse

Durch den mechanischen Aufbau können Streifen von 8 bis 24 mm transportiert werden, mit einem minimalen Bauteilschritt von 2mm. Der mechanische Aufbau hat gewisse Toleranzen, die von Feeder zu Feeder unterschiedlich sein können. Diese Abweichungen entstehen hauptsächlich durch die Verwendung von einem 3D gedruckten Schneckengetriebe. Wegen diesem entsteht Spiel, welches sich im Bauteilschritt widerspiegelt.

In Testläufen konnten gute Erfahrungen gemacht werden, welche bestätigen, dass ca. 500 0805 Bauteile ohne Schrittverlust entnommen werden können (Kapitel 6.3). Zu 0603 Bauteile können keine definitiven Angaben gemacht werden. Sollte der Feeder eine geringe Schritttoleranz haben, sind solche Werte auch bei solchen kleineren Komponenten möglich.

Da der Feeder im Rahmen dieser Arbeit nicht direkt an der PnP Maschine getestet wurde, können keine abschliessende Daten zu diesem Punkt gemacht werden. Dennoch werden die gemachten Testläufe positiv bewertet und dieses Ziel als erfüllt bewertet.

Grösse

Die Feeder sollten möglichst schmal sein, damit möglichst viele Feeder nebeneinander an der PnP Maschine montiert werden können. Das Ziel der Feedergrösse wurde als Streifenbreite + 15 mm definiert. Aus Tabelle 18 ist zu entnehmen, dass bei allen Feedern die angestrebte Breite nicht überschritten wurde. Somit wurde das Ziel eingehalten.

Tabelle 18 Vergleich Feederbreite Ziel/Ist

Feeder	Ziel	Ist
8 mm	23 mm	20.2 mm
12 mm	27 mm	20.2 mm
16 mm	31 mm	23.2 mm
24 mm	39 mm	31.7 mm

Hardware

Mit dem Einsatz des MK20DX128VFT5 von NXP und von kleinen DC-Getriebemotoren, welche auch bei anderen Projekten der HSLU eingesetzt werden, konnte dieses Ziel erreicht werden.

Kosten

Die ausführliche Kostenzusammenstellung ist im elektronischen Anhang im Ordner 1_Projektmanagement abgelegt. Dabei wurden die Kosten für die Herstellung von 50 Feeder berechnet. Da die Feeder für verschiedene Streifenbreiten unterschiedliche Abstandsbolzen benötigen wurde ein gemittelter Preis verwendet. Der Unterschied bezüglich verwendetem PLA Druckmaterial wurde ignoriert, da dies circa 2 Gramm ausmacht. Es wurde der gemittelte Wert verwendet. Die Versandkosten der Lieferanten wurden nicht miteinbezogen.

Tabelle 19 Kosten für 50 Feeder

	Kosten für 50 Feeder	Kosten pro Feeder
Total	1877.01	37.54

In den Zielen lag die Kosteneinschätzung bei 40 CHF pro Feeder. Die Kostenzusammenstellung eines Feeders liegt sogar leicht unter dem angestrebten Preis. Somit wurde dieses Ziel erfüllt.

Die Kosten für einen Master wurden auch berechnet. Die Gesamtkosten für einen einzelnen Master belaufen sich auf 6.20 CHF.

Funktionalitäten

In den Zielen wurden auch die Funktionalitäten aus der Aufgabenstellung definiert. Der Status in Bezug auf Zielerreichung wird in folgender Tabelle dargestellt. In dieser Tabelle ist zudem auch der Status vorhergenannten Ziele eingetragen.

Tabelle 20 Status der definierten Ziele

Funktion	Status
Bauteiltransport (Schrittweise)	Erreicht
Abdeckung entfernen	Erreicht
OpenPnP kompatibel	Erreicht (Masterkommunikation)
Manuelle Ansteuerung	Erreicht
Status-LED	Erreicht
Geschwindigkeit (4 mm/s)	Erreicht (8mm/s)
Benutzerfreundlichkeit	Erreicht
Open-Source	Erreicht
Streifenbreiten (8, 12, 16, 24 mm)	Erreicht
Bauteilgrösse (0805/0603)	0805 erreicht, 0603 möglich aber unbestätigt
Grösse	Erreicht
Hardware (NXP, Getriebemotor)	Erreicht
Kosten (40 CHF)	Erreicht (37.50)

7.3 Optimierungen

In diesem Kapitel werden Optimierungen beschrieben, welche in einer Weiterentwicklung von diesem Feeder umgesetzt werden könnten.

An dieser Stelle wird jedoch auch vermerkt, dass eine grössere Änderung des Designs notwendig wäre, um den Feeder mit dem Vision-System der PnP Maschine kompatibel zu machen. Diese Anpassung wäre durchaus machbar, wird jedoch in folgenden Optimierungsvorschlägen nicht berücksichtigt.

7.3.1 Mechanischem Aufbau

Breite

Die Breite der Feeder wird durch die Breite des Streifens, respektive der Führung definiert. Die 3D gedruckten Führungen könnten um ca. 1 mm, je nach Führung, weniger breit erstellt werden. Entsprechend müssten jedoch die Abstandshalter angepasst werden. Bei dem Feeder für 24 mm hat der Abstandshalter bereits eine Länge welche die Herstellung aus 3D Druck nötig macht. Es könnte somit nebst der Anpassung der Führung auch der Abstandshalter bei allen Feedern angepasst werden. Die Herstellung dieser Halter im 3D Druckverfahren würde zudem die Kosten der Feeder etwas senken.

Zusammenbau

Im momentanen Design führen die Schrauben zu einer Verbreiterung des Feeders. Die M2.5 Schrauben werden in der rechten Seitenwand versenkt. Diejenige in der linken Seitenwand nicht, da dies im Layout nicht berücksichtigt wurde. Insbesondere der Kopf der M3 Schraube, welche für den als Welle genutzten Distanzhalter verwendet wird, benötigt Platz. Diese Schraube kann leider nicht komplett versenkt werden, da die PCB-Wand nicht genug dick ist. Eine alternative Montagemöglichkeit der Sprocket-Welle, und die Anpassung des Layouts für die Versenkung der Schrauben, würde es ermöglichen den Feeder um 2 mm schmaler zu machen.

Pick-Location

Die Distanz zwischen Pick-Location und der Aussparung für den Peeler könnte verringert werden. Respektive könnten diese Stellen kombiniert werden. Im jetzigen Design war ursprünglich ein Deckel an der Peeler-Stelle eingeplant, welches jedoch nicht umgesetzt wurde. Mit der Kombinierung dieser Stellen könnte zum einen der Feeder verkürzt werden. Zudem würden weniger Komponenten aufgedeckt werden, welches zu weniger Verlust bei Streifenwechsel führen würde.

Schneckengetriebe

Die Schneckengetriebe wurde im 3D Druckverfahren hergestellt. Damit sie verwendet werden können müssen sie mechanisch nachbearbeitet werden. Dies benötigt Zeit. Zudem sind die mechanischen Toleranzen der Teile von dem jeweiligen 3D Drucker abhängig. Um Zeit zu ersparen und die Toleranzen zu minimieren sollte das Design so angepasst werden, damit eingekaufte Schneckengetriebe aus Kunststoff oder Metall eingesetzt werden können.

7.3.2 Elektrischer Hardware

Minimalisierung

Die Elektronik um den Mikrokontroller wurde von dem tinyK20 übernommen. Dabei wurden gewisse Komponenten übernommen, welche nicht absolut notwendig sind. Die Elektronik könnte überarbeitet werden und somit Komponenten eingespart werden. Dies würde zu einer Verringerung des Bestückungsaufwandes sowie den Kosten führen.

Layout

Das Layout könnte kompakter ausgelegt werden, damit die Grösse des Feeders verringert werden könnte. Zudem könnten die Verbindungsstecker oberhalb vom Streifen gesetzt werden, damit das Umstecken erleichtert werden könnte.

Fehler der eingesetzten Feeder-PCB Version 3 wurden bis zur Version 5 korrigiert. Diese Version ist im elektronischen Anhang dieser Dokumentation beigelegt.

Taster

Die verwendeten Vorwärts- und Rückwärtstaster könnten ausgetauscht werden. Benutzerfreundlicher wären abgewinkelte Taster, welche von oben betätigt werden können.

7.3.3 Software

Nachlaufzeit Peeler

Da der Detektor-Switch beim Peeler bereits bei leichtem Druck geschaltet wird, wird die Abdeckungsfolie nicht komplett angezogen. In Kapitel 5.1.5 wurde dies als positiver Punkt beschrieben. Mechanisches Spiel lässt den Streifen nicht immer an derselben Stelle anhalten. Der Peeler könnte den Streifen noch weiter nach vorne ziehen, bis auch das Spiel zwischen den Übergängen überwunden wäre. Somit könnte allenfalls die Genauigkeit des Feeders erhöht werden.

Error-Meldungen

In der Softwareversion (Stand Projektabgabe 06.06.2018) wird nur ein Timeout verwendet, um ein allfälliges Fehlverhalten des Feeders festzustellen. Diese Funktionalität könnte erweitert werden. In Zusammenhang mit elektronischen Erweiterungen könnte eine redundante Schritterkennung im Sinne eines Quadraturencoders erstellt werden.

Code-Verbesserungen

Allgemein wurde beim Entwickeln der Software Funktion vor Effizienz gestellt. Dies aufgrund des grossen Zeitaufwandes für die Entwicklung von Mechanik und Elektronik. In einem weiteren Schritt könnte die Software überarbeitet werden um dieser eine schönere und verständlichere Form zu geben.

7.4 Persönlicher Rückblick

Rückblickend kann ich sagen, dass die Realisation dieses Projekts sehr interessant und lehrreich war, aber auch zeitintensiv. Der Schwerpunkt des Projekts lag bei der Entwicklung des Feeders. Nebst der Umsetzung der Mechanik, Elektronik und der Software musste zudem Rücksicht darauf genommen werden, wie der Feeder an die PnP Maschine angebunden wird und mit OpenPnP kommuniziert, und wie der Feeder von Benutzern gehandhabt wird.

Die Arbeit war sehr komplex und eine grosse Herausforderung für mich, aber eine die ich sehr gerne umsetzte. Die Möglichkeit ein (Teil-)System von der Skizze zum funktionierenden Prototypen zu entwickeln begeisterte mich. Die Freiheit in der Entwicklung der Lösungswahl, mit hervorragender Unterstützung von dem betreuenden Dozenten Prof. Styger, erlaubten es mir wichtige Erfahrungen in der Projektumsetzung zu machen. Ein Projekt dieser Grössenordnung alleine Umzusetzen war eine Herausforderung. In früheren Projekten wie PREN war ich Teil eines Teams, wo die Arbeiten aufgetrennt wurden. Das PAIND Projekt hatte den Vorteil, dass man selbst immer den Überblick über den Stand des Projekts und die auszuführenden Arbeiten hatte. So wusste man auch wie man in Bezug auf die Projektplanung stand.

Das Projektmanagement mit Planung, Arbeitsjournal und Risikomatrix unterstützte mich bei der Entwicklung des Feeders. Mit der Planung mussten früh alle Arbeitspakete definiert werden. Somit wurde bereits ein Überblick über alle auszuführenden Arbeiten geschaffen. Mit diesem wurde mir zu Anfang an klar, welchen Aufwand ich betreiben musste, und konnte mich entsprechend einrichten. Das Arbeitsjournal wurde zwar nicht immer Wöchentlich nachgeführt, war aber eine grosse Unterstützung während dem Erstellen der Dokumentation, da viele Zusatzinformationen darin eingetragen waren.

Ich konnte Erfahrungen im Gebiet des CAD Zeichnens und dem 3D Druck machen, wo ich vorher noch keine hatte. Die Begeisterung für den 3D Druck hatte im Rückblick auch Nachteile. Die Getriebe würden ruhiger laufen, wenn sie eingekauft worden wären. Zudem hätte einiges an Zeit in der Entwicklung von diesen erspart werden können. In dem Sinne ist dies ein wichtiger Punkt den ich mitnehme. Die Modelle die ich in CAD erstellte unterstützten dieses Projekt wesentlich. Es konnte ständig überprüft werden wie und ob die verschiedenen Komponenten einander beeinflussten. So konnten Probleme bereits vor der Produktion eruiert und gelöst werden. Die Verwendung solcher CAD Modelle ist meines Erachtens wichtig.

Mit diesem Projekt konnte ich zudem weitere Erfahrungen in KiCAD machen. Ich war es mir gewohnt in Altium zu arbeiten. Dieses Projekt hat mir die Vorteile der Einfachheit von KiCAD nähergebracht und mir ein Tool gegeben, welches ich auch nach dem Studio Lizenzlos verwenden kann.

Die Umsetzung der Software mit Kinetis Design Studio war eine lehrreiche Erfahrung im Bereich des C-Programmierens, welches ich auch privat anwenden können wird. Der Einsatz von FreeRTOS brachte mir die Vorteile von diesem näher und vereinfachte die Umsetzung der Softwarefunktionalitäten im Feeder.

Die Kommunikationslösung mit einem Daisy-Chain zeigte auch, wie gut einfache, aber unkonventionelle Lösungen bei einer Problemstellung Abhilfe schaffen können.

Da ich den gesamten Feeder gezeichnet, gelayoutet, programmiert, montiert und getestet habe, und ein funktionierenden Prototypen entwickelt habe bin ich doch auch Stolz auf das Erreichte. Ich habe deutlich mehr Zeit in dieses Projekt investiert als durch das Modul PAIND vorgeschrieben wird. Die Schätzung der Arbeitszeit liegt über 300 Arbeitsstunden. Die Erstellung des Feeders in dieser Form war nur möglich, da gewisse Teile der Elektronik und Software von anderen Projekten wie das tinyK20 und dem SUMO-Roboter übernommen werden konnten.

In diesem Projekt lernte ich auch, dass man gewisse Funktionen einfach ausgiebig testen muss, bevor klar ist ob sie funktionieren. Ich arbeitete mehrmals mit Annahmen, die sich nicht bestätigten. Zudem darf und sollen Komponenten eingekauft und getestet werden, damit eine optimale Technologierecherche durchgeführt werden kann.

Der entwickelte Feeder ist immer noch ein Prototyp. Er hat Optimierungsbedarf und Potenzial. Aber die Funktion dieses Konzepts konnte bewiesen werden, und er kann mit der PnP Maschine von Prof. Styger verwendet werden. Es muss deutlich zu viel Zeit investiert werden, um einen zusammenzubauen, jedoch ist er kostengünstig und einzigartig. Ich konnte meine Kreativität in diesem Projekt umsetzen und der Feeder, der ich nun erstellt habe, bestätigt mir, dass sich die vielen investierten Stunden gelohnt haben.

8 Bedienungsanleitung

Die Bedienungsanleitung ist im Anhang vom Dokument sowie im elektronischen Anhang abgelegt.

In der Bedienungsanleitung werden die Schritte beschrieben, welche nötig sind um den Feeder an der PnP Maschine mechanisch einzurichten. Dabei wird auch auf den Handbetrieb eingegangen, der nötig ist damit ein Streifen in den Feeder geführt werden kann. Die Einrichtung des Feeders in OpenPnP wird in dieser Anleitung nicht beschrieben.

9 Montageanleitung

Die Montageanleitung ist im Anhang vom Dokument sowie im elektronischen Anhang abgelegt.

In der Montageanleitung ist jeweils eine Anleitung für den Zusammenbau des Feeders, des Masters sowie der Feederbank. Die Arbeitsschritte werden einzeln beschrieben. Zudem wird in einem zusätzlichen Kapitel die Arbeitsschritte zur Nachbearbeitung der 3D Druckteile beschrieben, damit der Zusammenbau möglich ist.

10 Quellenverzeichnis

Conrad. (2018). *Werklehrmittel Kunststoff-Schnecke Modelcraft*. Retrieved 02.06.2018, from <https://www.conrad.ch/de/werklehrmittel-kunststoff-schnecke-modelcraft-o-x-l-11-mm-x-31-mm-modul-typ-10-bohrungs-o-2-mm-239966.html>

Galindo, B. (2014, October) *150:1 Micromotor + gearbox (N20)*. Retrieved 21.03.2018, from <https://grabcad.com/library/150-1-micromotor-gearbox-n20-1>

Gipson, T. (2011, February). *Tape Sprocket Creator*. Retrieved 27.02.2018, from <http://tim.cexx.org/?p=798>

Grimstad, H.J. (2018, February). *Feeding Frenzy*. Retrieved 20.02.2018, from <https://blog.exploratory.engineering/post/feeder/>

KEMET. (2016, March). *Tape & Reel Packaging Information*. Retrieved 21.05.2018, from http://www.kemet.com/Lists/FileStore/C0G_Pulse_Qualification.pdf

McKay, J. (2016). *Pick-And-Place_OSH_feeder*. Retrieved 24.02.2018, from https://github.com/Joel-Mckay/Pick-And-Place_OSH_feeder

Mortezix. (2017, January) *Pick and place 3d printable feeder*. Retrieved 24.02.2018, from <https://github.com/Mortezix/pnpfeeder>

Mouser. (2018a). *MK20DX128VFT5*. Retrieved 27.05.2018, from <https://www.mouser.ch/ProductDetail/NXP-Freescale/MK20DX128VFT5?qs=%2fha2pyFaduj24fYdwuH5brDSz7l%2fEundzriCKWWCwT%252bfxtik21hOMA%3d%3d>

Mouser. (2018b). *MK22FN512VLH12*. Retrieved 27.05.2018, from <https://www.mouser.ch/ProductDetail/NXP-Freescale/MK22FN512VLH12?qs=sGAepiMZZMtzpSA5GSDwa4%252bsMgzeBaoc3vEAt7LZ5os%3d>

NXP. (2012, May). *K20 Sub-Family*. Retrieved 17.04.2018, from <https://www.mouser.ch/datasheet/2/302/K20P48M50SF0-1126149.pdf>

Omron. (2017, April). *Photomicrosensor EE-SX1350*. Retrieved 03.04.2018, from https://www.mouser.ch/datasheet/2/307/en-ee_sx1350-1076544.pdf

OpenPnP. (2018) *OpenPnP*. Retrieved 24.02.2017, from <http://openpnp.org/>

Pololu. (2018a). *Micro Metal Gearmotors*. Retrieved 03.04.2018, from <https://www.pololu.com/file/0J1487/pololu-micro-metal-garmotors.pdf>

Pololu. (2018b). *Produktenvergleich Schrittmotoren*. Retrieved 27.02.2018, from <https://www.pololu.com/search/compare/87>

Ravichandra, J.P. (2012, November). *Drum cam (Barrel cam) and follower*. Retrieved 8.03.2018, from <https://www.youtube.com/watch?v=9ZUUYZ1a4e4>

Smoothie. (2018). *Smoothieboards*. Retrieved 11.05.2018, from <http://smoothieware.org/smoothieboard>

Texas Instruments. (2016, August). *DRV8835 Dual Low-Voltage H-Bridge IC*. Retrieved 06.03.2018, from <http://www.ti.com/lit/ds/symlink/drv8835.pdf>

TinySine. (2018). *Micro Metal Gearmotor*. Retrieved 24.02.2018, from https://www.tinyosshop.com/index.php?route=product/product&path=64_82_172&product_id=222

Tycho. (2008, October). *Inkrementalgeber mit Gabellichtschränke*. Retrieved 11.05.2018, from https://upload.wikimedia.org/wikipedia/commons/9/91/Inkrementalgeber_mit_gabellichtschränke.JPG

VertikalDesign. (2009, September). *Mechanical Principles (1930) by Ralph Steiner*. Retrieved 8.03.2018, from <https://www.youtube.com/watch?v=mkQ2pXkYjRM>

Vishay. (2017, August). *Tape and Reel Specifications*. Retrieved 26.02.2018, from <https://www.vishay.com/docs/60034/tapereel.pdf>

A Anhang

A.1 Struktur Dokumentenanhang

- Aufgabenstellung
- Projektplan
- Elektroschema Feeder
- Bedienungsanleitung
- Montageanleitung

A.2 Struktur CD

0_Abgabedokument

- Abgabedokument

1_Projektmanagement

- Aufgabenstellung
- Risikomanagement
- Projektplan
- Arbeitsjournal
- Kostenzusammenstellung

2_Dokumente

- Betriebsanleitung
- Montageanleitung
- Elektroschema Feeder
- Elektroschema Master

3_Datenblätter

- Vishay Tape Size
- Kemet Tape Size
- Polulo N20 Gearmotors
- TI H-Bridge DRV8835
- Omron Lichtschranke EE-SX1350
- NXP Mikrokontroller MK20DX128VFT5
- Status LED Orange
- Microchip Spannungsregler TC1262
- Detector Switch
- Distanzbolzen 15mm

4_3D Daten

- Gemeinsame Dateien
- 8 mm Dateien
- 12 mm Dateien
- 16 mm Dateien
- 24 mm Dateien
- Mastergehäuse Dateien
- Druckeinstellungen

5_CAD Dateien

- Onshape Projektlink
- Assemblies
 - 8 mm
 - 12 mm
 - 16 mm
 - 24 mm
- DXF
 - PCB_Sketch_F
 - PCB_Sketch_B
 - Sprocket
 - Feederbank_Wall
 - Feederbank_Board

6_PCB Daten

- Altium (Sprocket Projekt)
- CAM Files
- KiCAD

7_Sonstiges

- Tape Sprocket Creator Python Script