



Bachelorstudiengang Informatik

Bachelorarbeit

BeadSort: Maschine zur Sortierung und Zusammenstellung von Steckperlen

vorgelegt von

Linus Barth

Betreuender Gutachter: Prof. Dr.-Ing. Oliver Theel

Zweiter Gutachter: Robert Schadek, M.Sc. Inform.

Oldenburg, den 16. Oktober 2017

Inhaltsverzeichnis

1. Einleitung	1
1.1. Hintergrund	1
1.2. Motivation	1
2. Aufgabenstellung und Zielsetzung	2
2.1. Anwendungsfälle	2
2.2. Anforderungen	3
3. Teilprobleme	5
3.1. Verwandte Lösungen	5
3.2. Ausrichtung	8
3.3. Ausgabeverfahren	13
3.4. Farberkennung	16
3.5. Vereinzelung	17
3.6. Sequenzialisierung	17
4. Entwurf	18
4.1. Einfüllbehälter	19
4.2. Sequenzialisierer	20
4.3. Ausrichter	21
4.4. Analysierer	22
4.5. Ausgeber	24
4.6. Stand	24
5. Test und Optimierung	26
5.1. Ansteuerung	26
5.2. Sequenzialisierer	26
5.3. Ausrichter	29
5.4. Vereinzelung	29
5.5. Analysierer	30
5.5.1. Existenzerkennung	30
5.5.2. Farberkennung	31
5.6. Ausgabeverfahren	37
6. Software	40
6.1. Konzept	40

6.2. Hardwareschicht	41
6.3. Programm	41
6.4. Kontrollschicht	42
6.5. Benutzerschicht	43
7. Bedienung	43
7.1. Startseite	43
7.2. Kalibrierung	44
7.3. Einstellungen	46
7.4. Farbmanagement	46
7.5. Programm	47
7.6. Ausführung	50
8. Evaluation und Fazit	51
8.1. Tests des Gesamtsystems	52
8.2. Fazit	60
A. Einrichtung	61

Abbildungsverzeichnis

1.	Erstellung von Puzzlesets	2
2.	farbliche Sortierung von Steckperlen	3
3.	Prototyp des Vibrationsausrichters	9
4.	Prototyp des Vibrationsausrichters	12
5.	Fixierte Ausrichtung von Steckperlen in einer Röhre	16
6.	Frontseite des BeadSort-3D-Modells	18
7.	Entwurf des Einfüllbehälters	19
9.	Entwurf des Sequenzialisierers	20
10.	Entwurf des Rotoraufsatzes	21
11.	Entwurf des Ausrichters	21
12.	Entwurf des Analysierers	22
13.	Entwurf des Ausgebers	24
14.	Entwurf des Stands	25
15.	Entwurf des Telleraufsatzes	28
16.	Entwurf des Analysierers - Version 2	32
17.	Beziehung der Farbwerte zur Abweichung	35
18.	Entwurf der Fixierung des Analysierers	36
19.	Entwurf des Ausgabeverfahrens - Version 2	38
20.	Abstrakte Übersicht der Software	40
21.	BeadSort-Software - Startseite	43
22.	BeadSort-Software - Powerdialog	44
23.	BeadSort-Software - Kalibrierung Teil 1	45
24.	BeadSort-Software - Kalibrierung Teil 2	45
25.	BeadSort-Software - Einstellungen	46
26.	BeadSort-Software - Farbmanagement	47
27.	BeadSort-Software - Programm	48
28.	BeadSort-Software - Programm - Unzugeordnete Steckperlen	48
29.	BeadSort-Software - Programm - Farbe	49
30.	BeadSort-Software - Programm - Puzzleset	49
31.	BeadSort-Software - Ausführung	50
32.	Zusammengesetzte BeadSort-Maschine	51
33.	Im Analysierer steckengebliebene Steckperle	52
34.	Ergebnis der Sortierung von Pastellfarben	53
35.	Beschädigte Steckperle im Vielfarbentest	56

36. Puzzleset-Erstellung	58
------------------------------------	----

1. Einleitung

1.1. Hintergrund

Um Kinder in ihrer Kreativität zu fördern, kann ihnen das Stecken (auch: Legen) von Bildern helfen. Dazu können z.B. Steckperlen und Steckplatten verwendet werden. Die Platten besitzen Noppen, auf die sich jeweils eine einzelne Steckperle stecken lässt. Die Steckperlen sind aber weiterhin abnehmbar und austauschbar. Erst durch Bügeln der gesetzten Steckperlen verschmelzen die Steckperlen zu einer festen Platte. Damit entsteht ein abgeschlossenes Bild, das sich unter anderem aufhängen lässt.

Die Steckperlen gibt es in verschiedenen Farben und Größen. Zusätzlich können sie Transparenz, Glitzer oder Farbmuster aufweisen. Es gibt drei verschiedene Größen: Mini (2,5 mm), Midi (5 mm) und Maxi (10 mm). Sie werden in großen Massen produziert und verkauft. So sind z.B. 13.000 Hama-Steckperlen für 8,99 € (zzgl. Versand) bei Amazon erhältlich.[6] Aufgrund der Massenproduktion und des niedrigen Preises besitzen die Steckperlen eine niedrige Qualität: die Steckperlen weisen sowohl in Farbe als auch in Größe und Form starke Abweichungen auf. [13, S. 23]

1.2. Motivation

Um ein bestimmtes Bild zu legen, müssen die benötigten Steckperlen erworben werden. Da die Steckperlen für gewöhnlich in Packungen von über 1000 Steckperlen erhältlich sind, ist es höchst unwahrscheinlich, die exakt passende Menge an Steckperlen erwerben zu können. Meist sind viele Steckperlen übrig. Die übrigen Steckperlen müssen dann entsorgt, verschenkt oder verstaut werden. Eine auf ein konkretes Bild zugeschnittene Zusammenstellung an Steckperlen wird im Folgenden als *Puzzleset* bezeichnet. Ein Puzzleset beinhaltet exakt die Steckperlen, die benötigt werden, um ein konkretes Bild zu legen. Offensichtlich bleiben beim Legen eines Bildes mit der Verwendung des passenden Puzzlesets keine Steckperlen übrig. Eine Maschine soll ein Puzzleset für ein jeweils vorgegebenes Bild erstellen können.

Es existieren Maschinen, die automatisiert Steckperlenbilder legen können. Im Rahmen einer studentischen Abschlussarbeit in der Abteilung Systemsoftware und verteilte Systeme wurde eine solche Maschine entwickelt und BeadBox genannt. [28] Die BeadBox benötigt farblich sortierte Steckperlen, um korrekt zu arbeiten. Zusätzlich müssen die Steckperlen stehend ausgerichtet sein. Es ist also erstens eine farbliche Vorsortierung und zweitens eine Vorausrichtung der Steckperlen notwendig. Von Hand ist die farbliche Vorsortierung und die Vorausrichtung sehr zeitaufwändig. Eine weitere Maschine soll daher die farbliche Vorsortierung und die Vorausrichtung übernehmen.

2. Aufgabenstellung und Zielsetzung

Ziel dieser Arbeit ist es, eine Maschine zur farblichen Sortierung und zur Ausrichtung von Steckperlen zu entwickeln. Im Folgenden wird diese Maschine als *BeadSort-Maschine* bezeichnet. Die BeadSort-Maschine soll Puzzlesets erstellen können. Zur Entwicklung der BeadSort-Maschine werden voraussichtlich Teile benötigt, die mit einem 3D-Drucker gedruckt werden können. Diese Teile werden mit der 3D-Modellierungssoftware *OpenSCAD* entworfen. Zum Drucken der entworfenen Teile wird ein Ultimaker 2 verwendet.

Ein Raspberry Pi soll als Steuerungseinheit der BeadSort-Maschine dienen. Zur Bedienung der BeadSort-Maschine wird ein Touch-Bildschirm verwendet, der an dem Raspberry Pi angeschlossen wird.

2.1. Anwendungsfälle

Die Maschine soll in folgenden zwei Anwendungsfällen eingesetzt werden können:

Der erste Anwendungsfall ist die Erstellung von Puzzlesets. In der Abbildung 1 ist der Anwendungsfall schematisch dargestellt. Zur Erstellung eines Puzzlesets muss ein Bild als Grundlage gewählt werden. Die Bilddatei des gewählten Bildes muss sich lokal auf dem Raspberry Pi oder auf einem am Raspberry Pi angeschlossenen USB-Datenträger befinden. Alle Bilddateien, die sich lokal oder auf einem angeschlossenen USB-Datenträger befinden, sollen als auswählbares Element auf dem Touch-Bildschirm dargestellt werden. Die Bilddatei des gewählten Bildes soll dann über den Touch-

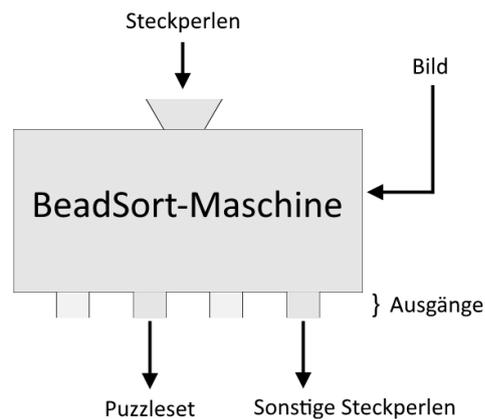


Abbildung 1: Erstellung von Puzzlesets

Bildschirm ausgewählt werden. Die BeadSort-Maschine soll dann für das ausgewählte Bild ein Puzzleset erstellen. Die Eingabeperlen, die das Puzzleset benötigt, werden über einen designierten Ausgang der BeadSort-Maschine ausgegeben. Ein anderer Ausgang ist für die Ausgabe nicht benötigter Steckperlen designiert und wird im Folgenden als *Sonstige-Ausgang* bezeichnet.

Auf dem angeschlossenen Bildschirm soll dargestellt werden, wie viele Steckperlen der

jeweiligen Farben benötigt werden und wie viele Steckperlen für die Vervollständigung des Puzzlesets gegenwärtig noch fehlen.

Der zweite Anwendungsfall ist die farbliche Sortierung von Steckperlen. In der Abbildung 2 ist der Anwendungsfall schematisch dargestellt. Die BeadSort-Maschine soll für die zu sortierenden Farben in folgender Weise konfiguriert werden: Die Konfiguration soll über den angeschlossenen Touch-Bildschirm erfolgen. Die zu sortierenden Farben werden dann jeweils einem Ausgang zugeordnet. Zudem wird ein Ausgang für den Sonstige-Ausgang freigehalten. Damit entspricht die maximale Anzahl an konfigurierbaren Farben der Anzahl an Ausgängen minus eins. Die BeadSort-Maschine soll dann die Eingabeperlen

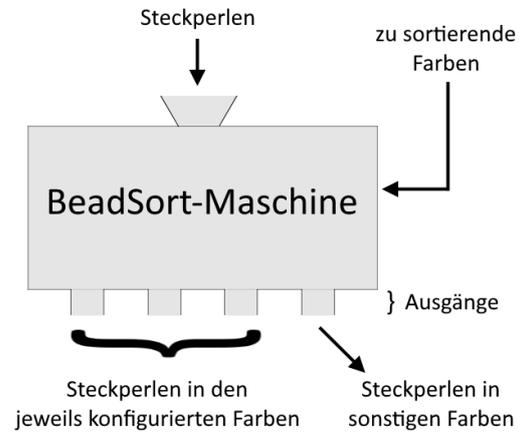


Abbildung 2: farbliche Sortierung von Steckperlen

verarbeiten. Die Steckperlen einer Farbe sollen an dem der Farbe zugeordneten Ausgang ausgegeben werden. Die Steckperlen, die keinem Ausgang zugeordnet werden können, werden an dem Sonstige-Ausgang ausgegeben.

Auf dem angeschlossenen Bildschirm soll dargestellt werden, wie viele Steckperlen der konfigurierten Farben jeweils sortiert wurden und wie viele Steckperlen an dem Sonstige-Ausgang ausgegeben wurden.

2.2. Anforderungen

Eingabe

Die BeadSort-Maschine soll über einen Eingangsbehälter verfügen, der eine Vielzahl an Steckperlen beinhalten kann. Die in den Eingangsbehälter eingegebenen Steckperlen werden im Folgenden als *Eingabeperlen* bezeichnet. Die Eingabeperlen müssen jeweils einfarbig sein. Die Eingabeperlen sollen keine Transparenz, keinen Glitzer und keine Muster besitzen. Die Eingabeperlen müssen in der Midi-Größe vorliegen. Die Ausrichtung der Eingabeperlen darf beliebig sein.

2. Aufgabenstellung und Zielsetzung

Ausgabe

Die Maschine soll mehrere Ausgänge besitzen, über denen die Eingabeperlen ausgegeben werden können. Die Wahl des Ausgangs hängt von der Farbe der jeweiligen Steckperle ab. Die Steckperlen sollen in einer festgelegten Ausrichtung ausgegeben werden, um die weitere Verarbeitung durch die BeadBox zu ermöglichen.

Die BeadSort-Maschine soll über den angebrachten Bildschirm Informationen über den Status des Sortierprozesses liefern. An späteren Stellen dieses Dokuments wird erläutert, welche genauen Informationen ausgegeben werden sollen. Der Anwender soll mithilfe der Bildschirmausgabe eine Abschätzung des Fortschritts der Puzzlesets machen können.

Verarbeitung

Es gibt keine Einschränkung der Farben, die die Eingabeperlen haben können. Die BeadSort-Maschine muss somit beliebige Farben erkennen können.

Die Eingabeperlen müssen von der BeadSort-Maschine einzeln betrachtet werden, daher ist eine Vereinzelnung der Steckperlen notwendig. Die BeadSort-Maschine muss also einen Vereinzelnungsmechanismus implementieren.

Da die Eingabeperlen nicht ausgerichtet sind, jedoch ausgerichtet ausgegeben werden sollen, muss die BeadSort-Maschine einen Ausrichtungsmechanismus implementieren. Bei der Sortierung von Steckperlen nach konfigurierten Farben wird den Ausgängen jeweils eine Farbe zugewiesen. Dabei scheint die Anzahl der Ausgänge die Sortiermächtigkeit einzuschränken. Mittels Rekursion lässt sich dieses Problem jedoch lösen. Beispielsweise könnte eine Maschine mit nur vier Ausgängen Steckperlen mit breiterer Farbpalette mit folgendem Schema korrekt sortieren. Es werden drei Farben gewählt und jeweils einem Ausgang zugewiesen. Alle Steckperlen die diesen Farben entsprechen, werden in den jeweiligen Ausgang weitergeleitet. Die sonstigen Steckperlen werden an dem Sonstige-Ausgang ausgegeben. Wenn keine Steckperlen mehr am Eingang vorhanden sind, dann wird der Vorgang beendet. Die Steckperlen aus dem Behälter des Sonstige-Ausgangs werden erneut in den Eingang gegeben. Die zuvor sortierten Steckperlen werden beiseite genommen. Nun werden drei andere Farben konfiguriert. Der Vorgang wiederholt sich, bis alle Steckperlen entsprechend sortiert sind. Im Folgenden wird diese Methode als *Methode zur Kompensierung fehlender Ausgänge* bezeichnet.

3. Teilprobleme

Die Entwicklung der BeadSort-Maschine wirft vier Teilprobleme auf. Diese werden im Folgenden näher erläutert.

Erkennung der jeweiligen Farbe der Steckperlen

Die eingegebenen Steckperlen sollen an dem jeweils korrekten Ausgang ausgegeben werden. Korrekt ist ein Ausgang für eine bestimmte Steckperle dann, wenn die für den Ausgang konfigurierte Farbe der Steckperlenfarbe entspricht. Die Maschine muss die jeweilige Farbe der Steckperlen somit erkennen können, um den korrekten Ausgang wählen zu können.

Vereinzelung der Steckperlen

Die Steckperlen werden der Maschine über einen Eingangsbehälter eingegeben. Es soll möglich sein, die Steckperlen einzuschütten. Für die Erkennung der Farbe müssen die Steckperlen jedoch einzeln betrachtet werden. Infolgedessen muss die Maschine eine Vereinzelung implementieren, die die Betrachtung einer einzelnen Steckperle ermöglicht.

Ausrichtung der Steckperlen

Die Steckperlen werden der Maschine in beliebiger Ausrichtung eingegeben. Die Ausgabe soll jedoch ausgerichtet erfolgen. Die Maschine muss die Steckperlen somit ausrichten können.

Ausgabe der Steckperlen am jeweilig korrekten Ausgang

Sobald die Ausgabe einer Steckperle ansteht, muss diese an den korrekten Ausgang befördert werden. Die Maschine muss somit eine Methode implementieren, die diese Beförderung vornimmt. Im folgenden wird diese Methode als *Ausgabeverfahren* bezeichnet.

Die aufgeführte Reihenfolge der Teilprobleme hat keinen Zusammenhang mit der Reihenfolge, in der die Maschine die Probleme lösen muss.

3.1. Verwandte Lösungen

Im Folgenden werden bereits existierende Sortiermaschinen untersucht. Dabei werden besonders die Lösungen der zuvor ermittelten Teilprobleme betrachtet. Ziel ist

3. Teilprobleme

es, die optimalste Zusammenstellung von Teilproblemlösungen zu finden. Als optimal gilt primär, dass die Anforderungen der BeadSort-Maschine erfüllt werden. Sekundäre Faktoren wie zum Beispiel Materialkosten werden an entsprechender Stelle abgewogen.

Es existieren viele Projekte im Internet, die sich mit der Entwicklung von Maschinen zur farblichen Sortierung befassen. Diese Projekte werden in vielen Fällen von einzelnen Personen durchgeführt. Dabei werden Videos von den Ergebnissen oftmals auf Videoportalen wie YouTube veröffentlicht. Die Maschinen sortieren Steckperlen oder Süßigkeiten wie Smarties und Skittles. Die Maschinen unterscheiden sich dabei teilweise stark in der Vorgehensweise und bringen verschiedene Vor- und Nachteile mit sich. Im Weiteren wird Bezug auf die gefundenen Sortiermaschinenprojekten genommen. Diese Projekte sind in Tabelle 1 vollständig vertreten. Die URLs, an denen Informationen zu den Sortiermaschinenprojekten gefunden wurden, sind in den jeweiligen Literaturverzeichniseinträgen zu finden

Bei der Recherche fiel auf, dass die meisten Steckperlen-Sortiermaschinenprojekte keine Ausrichtung der Steckperlen vornehmen. Daher wird in Tabelle 1 mit einem X angegeben, ob bekannt ist, dass eine Ausrichtung durchgeführt wird.

Entwickler	Sortiergut	Ausrichtung	Literatur
Bernhard Schwerdtner	Steckperlen		[27]
Geir Andersen	Steckperlen		[8]
Scanlime	Steckperlen	X	[25],[24],[23]
Sammy Linden	Steckperlen	X	[16],[17]
C. Mei, K. Liu, V. Auduong	Steckperlen		[11]
Gabriel Schneider	Steckperlen		[26]
MarVtec	Skittles		[18]
Rolf R Bakke	Skittles		[9]
Egen Riether	Skittles		[22]
Microcontrollers Projects	Skittles und M&M's		[20]

Tabelle 1: Recherchierte Sortiermaschinenprojekte

Es wurde lediglich ein Projekt gefunden, in dem das Ausrichten von Steckperlen vorgenommen wird. Eine Person mit dem Pseudonym *Scanlime* Scanlime veröffentlichte auf ihrem Blog im Februar 2011 eine Ausführung ihrer Ideen zu einer Steckperlen-Legemaschine [25]. Die Ausführung beinhaltet Konzepte zur farblichen Vorsortierung und Ausrichtung der Steckperlen. Im März 2011 folgte ein Beitrag auf *Thingiverse* [24], an dem ein Modell des derzeitigen Stands der Maschine im OpenSCAD-Format anhängt. Ein aktuellerer Beitrag konnte nicht gefunden werden. Der Beitrag enthält

ein Link zu einem Video, das die Funktionstüchtigkeit der Ausrichtung zeigt [23]. In dem Video ist nur der Teil der Maschine verwendet worden, der für die Ausrichtung zuständig ist. Dieser Teil besteht im Wesentlichen aus drei Komponenten:

- einem Trichter, durch den die Steckperlen nur ausgerichtet hindurch passen
- einem Motor, der den Trichter zum Vibrieren bringt
- einer Halterung, die den Trichter und den Motor mit einer Federung in Position hält

In dem 16-sekündigen Video ist zu Anfang der Trichter bis zum Rand gefüllt und der Motor ausgeschaltet. Dann wird der Motor eingeschaltet und die Steckperlen fangen an sich zu bewegen. Es ist zu sehen, dass die Steckperlen durch das Ausgangsloch des Trichters fallen, bis der Trichter leer ist. Da die Steckperlen nur ausgerichtet durch das Ausgangsloch passen, sind alle Steckperlen erfolgreich ausgerichtet worden.

Damit existiert ein Lösungsansatz für die Ausrichtung der Steckperlen. Bei der speziellen Suche nach Lösungen zur Ausrichtung von Steckperlen konnte zudem nur folgende weitere Quelle gefunden werden. Eine Person mit dem Benutzernamen *SammyLinden* teilte im Februar 2017 eine Erweiterung zu Scanlimes Ausrichtungslösung in einem Beitrag auf Thingiverse [16]. Dabei handelt es sich um einen Trichter, der in den vibrierenden Trichter von Scanlime eingesetzt wird. In dem Beitrag befindet sich ein Link zu einem Video, in dem eine erfolgreiche Ausrichtung gut ersichtlich ist [17]. Im Gegensatz zu dem Video von Scanlime sind die ausgerichteten Steckperlen erkennbar. Aus der Beschreibung wird jedoch nicht klar, warum er diesen Trichter entwickelt hat beziehungsweise was für Vorteile dieser bewirkt. Mittels Nachfrage im Kommentarbereich des Beitrags konnte dies jedoch erfragt werden. SammyLinden antwortete, dass das Loch im Originaldesign zu weit für die Steckperlen von Ikea war. Die Ikea-Steckperlen gingen seitlich in das Loch und blieben gelegentlich stecken.

Daraufhin stellten sich folgende Fragen:

- Bleiben Steckperlen immer noch ab und zu stecken?
- Wieviele Steckperlen richtet die Lösung ungefähr pro Minute aus?
- Wurde bereits versucht, einen Einfüllbehälter über den Ausrichter zu bauen, sodass eine größere Menge an Steckperlen eingegeben werden kann? Und hat das Gewicht einer großen Menge Perle dann einen schlechten Einfluss auf die Ausrichtungsfähigkeit?

3. Teilprobleme

Außerdem fiel nach näherer Betrachtung des Videos auf, dass SammyLinden einen anderen Motor als Scanlime verwendet und diesen näher an den Trichter positioniert hat. Das wirft die Frage auf, welche Gründe dies hatte. Diese und die oben aufgelisteten Fragen wurden ebenfalls im Kommentarbereich von SammyLindens Beitrag gestellt. Jedoch kam darauf in den nachfolgenden Wochen keine Antwort. Es musste ohne diese Informationen weitergearbeitet werden.

Scanlimes Lösung zur Ausrichtung von Steckperlen wird im Folgenden als *Vibrationsausrichter* bezeichnet. SammyLindens Erweiterung des Vibrationsausrichters ist eine Anpassung zur Verwendung von Ikea-Steckperlen. Die BeadSort-Maschine soll aber hauptsächlich mit Hama-Steckperlen verwendet werden. Daher wird im Folgendem nicht näher auf diese Erweiterung eingegangen.

Scanlime beschreibt ihr Projekt auf Thingiverse als "very much a work in progress"[16], also noch in Arbeit. Die 3D-Modelldateien aus dem Thingiverse-Beitrag bestätigen dies, da das Modell unvollständig scheint. Seit dem Beitrag sind keine weiteren Fortschritte des Projektes zu finden. Es lässt sich somit keine vollständige Sortiermaschine finden, die das farbliche Sortieren und Ausrichten von Steckperlen vornimmt. Das macht die Entwicklung der BeadSort-Maschine notwendig.

Es lässt sich keine Maschine finden, die Puzzlesets erstellen kann. Die BeadSort-Maschine ist somit vermutlich einzigartig in sowohl der Ausrichtung von Steckperlen als auch in der Erstellung von Puzzlesets.

Da keine weiteren Lösungen zur Ausrichtung von Steckperlen gefunden werden können, besteht wenig Information über mögliche Methoden zur Ausrichtung. Die Lösungen zu den Teilproblemen müssen zueinander kompatibel sein. Die Lösung eines Teilproblems könnte somit die möglichen Lösungen anderer Teilprobleme einschränken. Daher wird das Teilproblem der Ausrichtung zuerst gelöst, um weitere Einschränkungen zu vermeiden.

3.2. Ausrichtung

Eine eigene Entwicklung eines Ausrichtungsmechanismus wird aus folgenden Gründen vorerst nicht vorgenommen:

- Der Zeitaufwand zur Ermittlung von Möglichkeiten zur Ausrichtung von Steckperlen ist nicht abschätzbar
- Das Prototyping für jede gefundene Möglichkeit kann einen hohen Zeitaufwand und hohe Materialkosten beanspruchen

- Die Bearbeitungszeit dieser Arbeit ist begrenzt
- Der Vibrationsausrichter ist bereits getestet
- Die 3D-Modell-Dateien des Vibrationsausrichters sind öffentlich verfügbar und ermöglichen einen schnellen Nachbau

Die BeadSort-Maschine wird somit den Vibrationsausrichter zur Ausrichtung von Steckperlen verwenden. Es ist jedoch nicht bekannt, ob eine große Menge Steckperlen auf dem Vibrationsausrichter die Ausrichtungsfähigkeit einschränkt. Daher wird ein Prototyp gebaut, mit dem dies getestet wird. Zusätzlich ist nicht bekannt, ob der Vibrationsausrichter zuverlässig genug ist. Es sollten keine Steckperlen stecken bleiben. Es kann also sein, dass eine Änderung oder Erweiterung notwendig ist, um die Zuverlässigkeit zu erhöhen.

Da das 3D-Modell des Vibrationsausrichters im OpenSCAD-Format vorhanden ist, kann das Modell ohne weiteres mit der OpenSCAD-Software bearbeitet werden. Die Teile des Vibrationsausrichters, die Scanlime zum Zusammenbau mit weiteren Komponenten vorgesehen hat, sind für einen Prototyp nicht notwendig und werden daher entfernt. Die Motorhalterung ist auf einen speziellen Motor zugeschnitten. Da keine Angabe zu den genauen Motordaten gegeben ist, kann dieser nicht erworben werden. Die Motorhalterung muss somit abgeändert werden. Dazu wird zuerst ein kleiner DC-Motor ausgewählt, der eingesetzt werden soll. Die Abmessungen dieses Motors werden verwendet, um eine passende Motorhalterung in das Modell zu modellieren. Das Ergebnis ist in Abbildung 3 zu sehen:



Abbildung 3: Prototyp des Vibrationsausrichters

Der erste Test des Prototypen wird wie folgt vorgenommen:

3. Teilprobleme

- Der Trichter wird mit Steckperlen befüllt. Für diesen Test werden 28 Steckperlen als Eingabe in den Trichter festgelegt. Der Trichter ist mit 28 Steckperlen ungefähr bis zum oberen Rand gefüllt
- Der Vibrationsmotor wird eingeschaltet
- Sobald alle Steckperlen durch sind oder erkennbar ist, dass sich eine Steckperle verhakt hat, wird der Vibrationsmotor ausgeschaltet und der Test beendet

Es werden fünf Durchläufe des Tests durchgeführt. Im ersten Durchlauf des Tests fallen die Steckperlen in 10,5 Sekunden durch den Trichter. Da alle Steckperlen korrekt ausgerichtet wurden, ist der Durchlauf als erfolgreich anzusehen. In den vier weiteren Durchläufen bleibt eine Steckperle quer im Ausgangsloch des Trichters stecken und verstopft diesen somit. Die Abweichungen in Größe und Form einer Steckperle können unter Umständen einen Einfluss auf die Wahrscheinlichkeit besitzen, mit der diese Steckperle stecken bleibt. Es waren jedoch keine Besonderheiten an den steckengebliebenen Steckperlen erkennbar, die für die Verstopfung verantwortlich sein könnten.

Das Ergebnis des Tests ist zu unzuverlässig. Es müsste sehr häufig manuell in die BeadSort-Maschine eingegriffen werden, um die Verstopfung aufzulösen. Mögliche Lösungen zur Verbesserung könnten wie folgt aussehen:

1. Ein Mechanismus könnte die Verstopfung auflösen. Die verstopfte Steckperle könnte zum Beispiel von unten aus dem Ausgangsloch herausgedrückt werden, sodass diese sich wieder im Trichter befindet
2. Eine stärkere Vibration könnte womöglich die verstopften Steckperlen aus dem Ausgangsloch herausrütteln oder sogar eine Verstopfung verhindern
3. Eine Änderung der Lochgröße oder der Trichterform könnte das Verstopfen verhindern
4. Eine Verringerung der Steckperlen im Trichter. Die Idee ist, dass die Steckperlen in geringerer Anzahl eine größere Bewegungsfreiheit besitzen und dadurch möglicherweise seltener oder garnicht mehr stecken bleiben

Im folgenden werden die Lösungen jeweils näher betrachtet um abschließend die geeignetste Lösung zu ermitteln:

Lösung 1 stellt eine Problemauflösungsstrategie dar. Sie verhindert nicht das Verstopfen von Steckperlen, sondern löst es wieder auf. Dies bringt folgende Probleme mit

sich. In der zum Auflösen der Verstopfung benötigten Zeit werden keine Steckperlen ausgerichtet. Somit verringert sich die Effizienz des Vibrationsausrichters. Zudem muss entweder erkannt werden, ob eine Verstopfung vorliegt oder periodisch eine Entstopfung durchgeführt werden. Zur Erkennung einer Verstopfung wird zusätzliche Hardware benötigt und die Komplexität der Software gesteigert. Zusätzlich wird zur Entwicklung der Verstopfungserkennung Arbeitszeit benötigt. Wird die Entstopfung periodisch ausgeführt, so muss diese auch im nicht verstopften Fall funktionieren. Da während der Entstopfung keine Steckperlen ausgerichtet werden können, bedeutet das je nach Periodizität eine geringere Effizienz der Vibrationsausrichtung. Ein Vorteil der Lösung ist, dass bei idealer Implementierung die Verstopfung immer selbst aufgehoben wird und nie ein manuelles Eingreifen nötig wird.

Lösung 2 könnte sowohl eine Problemlösungs- als auch eine Problemvermeidungsstrategie darstellen. Es ist unbekannt, ob eine stärkere Vibration einen positiven Einfluss auf die Auflösung oder Vermeidung von Verstopfungen hat. Es könnte sogar zu mehr Verstopfungen kommen. Zudem wäre ein stärkerer Motor notwendig. Das bedeutet höhere Kosten und wahrscheinlich eine notwendige Überarbeitung der Motorhalterung. Es ist außerdem denkbar, dass die Steckperlen bei geringer Befüllung so stark in Bewegung gesetzt werden, dass sie seltener bis gar nicht mehr in das Ausgangsloch fallen. Die von der Vibration entstehenden Geräusche würden verstärkt werden und eine Verwendung der BeadSort-Maschine unangenehmer machen. Diese Lösung kann jedoch mit relativ geringem Aufwand angewandt werden.

Lösung 3 könnte das Problem der Verstopfung vermeiden. SammyLinden hat mit seiner Erweiterung zur Änderung der Lochgröße von guten Erfolgen berichtet. Zu beachten ist dabei jedoch, dass das Originaldesign für die verwendeten Hama-Steckperlen gedacht war und SammyLinden eine Anpassung für Ikea-Steckperlen vorgenommen hat. Außerdem hat SammyLinden nicht beantwortet, ob er die Verstopfungen ganz vermeiden konnte. 3D-Drucker besitzen je nach verwendetem 3D-Drucker und Material unterschiedliche Genauigkeiten. [1] Das führt zu einer abweichenden Lochgröße je nach Drucker und Material. Zudem schrumpft der Druck beim Abkühlen je nach Material unterschiedlich stark. [1] Somit ist eine Anpassung der Lochgröße auf den verwendeten 3D-Drucker und das verwendete Material ohnehin empfehlenswert. Aufgrund der parametrisierten Erstellung von 3D-Modellen mit OpenSCAD lässt sich die Ausgangslochgröße mit geringem Aufwand verändern. Lediglich die Druckzeit und die Druckkosten sind nachteilig.

Es ist leicht zu testen, ob Lösung 4 einen Vorteil bringt. Der oben durchgeführte Test muss lediglich mit einer geringeren Anzahl an Steckperlen durchgeführt wer-

3. Teilprobleme

den. Jedoch muss bei Erfolg eine Methode entwickelt werden, die die Füllmenge an Steckperlen nur bis zu einer ausreichend geringen Menge begrenzt. Dies würde einen Zeitaufwand und unter Umständen weitere Materialkosten bedeuten.

Aufgrund der Ergebnisse von SammyLinden und den Erkenntnissen über Abweichungen des 3D-Druckes aufgrund von 3D-Drucker und Material wird zunächst Lösung 3 angewendet.

Die Größe des Ausgangsloch wird zunächst von 7 Millimeter Durchmesser auf 6.6 Millimeter Durchmesser reduziert. Der entstehende Druck des zweiten Prototypen weist eine deutlich höhere Qualität auf. Der Durchmesser des Ausgangsloches beträgt beim Druck nur noch ca. 6 Millimeter. Auf Abbildung 4 ist der zweite Prototyp zu sehen.



Abbildung 4: Prototyp des Vibrationsausrichters

Der Test für den ersten Prototypen wird für den zweiten Prototypen dreimal wiederholt. In allen Durchläufen bleibt keine der 28 eingegebenen Steckperlen stecken.

Es besteht die Annahme, dass eine größere Menge Steckperlen die Ausrichtung verschlechtern würde. Das höhere Gewicht könnte die notwendige Vibration der Steckperlen im Trichter unterdrücken. Es wird ein Behälter konstruiert, der diese Annahme prüfen soll. Dieser Behälter besitzt an seinem Boden ein Loch mit dem Durchmesser des Ausrichtungstrichters. Der Behälter wird so über dem Ausrichter positioniert, dass sich der Ausrichtungstrichter direkt unter dem Loch am Boden des Behälters befindet. Dann werden ungefähr 3000 Steckperlen in den Behälter eingegeben. Ein Teil davon fällt wie gewünscht in den Ausrichtungstrichter. Dann wird der Vibrationsmotor gestartet.

Es verklemmte sich direkt nach Start des Vibrationsmotors eine Steckperle. Auch

nach Wiederholungen des Tests verklemmte sich entweder sofort oder nach wenigen Steckperlen erneut eine Steckperle.

Der Vibrationsausrichter kann somit nur kleine Mengen oder gar einzelne Steckperlen ausrichten. Es muss also eine Methode gefunden werden, die den Füllstand des Ausrichtungstrichters reguliert.

Die Eingabe der Steckperlen über ein Rohr könnte das Problem lösen. Die Idee ist, dass nur Steckperlen aus dem Rohr in den Trichter fallen, wenn ausreichend Platz im Trichter ist. Diese Vermutung wird wie folgt getestet. Ein Rohr mit einem Innendurchmesser von 8mm wird verwendet. In diesem Rohr können sich keine zwei Steckperlen nebeneinander befinden, da der Innendurchmesser kleiner als der Durchmesser von zwei Steckperlen ist. Das Rohr ist ungefähr 40cm lang und wird so über dem Ausrichtungstrichter fixiert, dass es 1cm in den Trichter hineinragt. Das Rohr wird bis oben hin mit Steckperlen befüllt. Dabei fallen die ersten Steckperlen in den Ausrichtungstrichter, bis der Füllstand so hoch ist, dass keine Steckperlen mehr aus dem Rohr fallen können. Dann wird der Vibrationsmotor gestartet. Dieser Test wird dreimal durchgeführt. Nach jeweils 25, 20 und 24 Sekunden sind alle Steckperlen erfolgreich ausgerichtet worden. Keine Steckperlen sind während des Tests aus dem Ausrichtungstrichter übergelaufen.

Damit reguliert ein Rohr den Füllstand des Ausrichtungstrichters. Es muss somit jedoch eine Methode entwickelt werden, die die eingegebenen Steckperlen in ein Rohr positioniert. Dazu mehr in Sektion `refsec:sequenzialisierung`. Der Ausrichter ist an dieser Stelle fertiggestellt.

Die Ausrichtung sollte aus folgendem Grund vor dem Ausgabeverfahren geschehen. Die Konstruktion des Ausrichters müsste inklusive Hardware für jeden Ausgang jeweils repliziert werden. Wenn vor dem Ausgabeverfahren ausgerichtet wird, dann wird diese Redundanz vermieden. Spätere Optimierungen (zum Beispiel eine geänderte Ausgangslochgröße) müssen zudem nur auf eine Instanz angewandt werden. Es muss aber beachtet werden, dass das Ausgabeverfahren die Steckperlen in ausgerichtet belässt. Diese Einschränkung wird akzeptiert.

3.3. Ausgabeverfahren

Die Ausgabeverfahren von folgenden Entwicklern weisen zueinander eine hohe Ähnlichkeit in der Funktionsweise auf:

- Egen Riether [22]
- Gabriel Schneider [26]

3. Teilprobleme

- C. Mei, K. Liu, V. Auduong [11]
- Bernhard Schwerdtner [27]

Die Ähnlichkeit besteht in dem beweglichen Rohr, das das auszugebende Teil durchläuft. Ein Aktor (für gewöhnlich ein Servo) kann die Position des Rohrausgangs auf festgelegte Positionen verändern. Eine solche festgelegte Position stellt einen Ausgang der Sortiermaschine dar. Im Weiteren wird dieses Verfahren als *Rohrverfahren* bezeichnet. Abgesehen vom Rohrverfahren existieren zwei weitere Ausgabeverfahren, die jeweils in einer der recherchierten Sortiermaschinen verwendet werden.

Zum einen handelt es sich um ein *Binärbaumverfahren* in der Sortiermaschine von Microcontrollers Projects [20]. Bei diesem Verfahren ist die Wurzel des Binärbaums der Eingang der auszugebenden Teile. Zudem ist jedes Blatt des Binärbaums jeweils ein Ausgang der Sortiermaschine. Mittels Aktoren wird ein Durchgang von der Wurzel bis zum gewünschten Blatt geöffnet. Alle anderen Wege werden blockiert. Das auszugebende Teil kann dann nur in den gewünschten Ausgang fallen.

Zum anderen handelt es sich um ein *Fließbandverfahren* von Rolf R Bakke [9]. Bei diesem Verfahren werden die auszugebenden Teile auf einem Fließband fortbewegt. An festgelegten Positionen können Aktoren das auszugebende Teil vom Fließband herunterbewegen. Diese festgelegten Positionen stellen jeweils einen Ausgang der Maschine dar.

Die Aktoranzahl von sowohl dem Binärbaumverfahren wie auch dem Fließbandverfahren skaliert mit der Anzahl an gewünschten Ausgängen. Beim Rohrverfahren hingegen ist die Aktoranzahl unabhängig von der Anzahl an gewünschten Ausgängen. Um vier Ausgänge zu realisieren benötigen die drei Verfahren folgende Anzahl an Ausgängen:

- Das Binärbaumverfahren benötigt bei vier Ausgängen zwei Aktoren, da für jede Ebene ein Aktor benötigt wird.
- Das Fließbandverfahren benötigt bei vier Ausgängen bereits vier Aktoren. Ein Aktor bewegt das Fließband fort und drei Aktoren bewegen das auszugebende Teil an jeweils der gewünschten Position vom Fließband herunter.
- Das Rohrverfahren benötigt unabhängig von der Anzahl an Ausgängen nur einen Aktor. Dieser bewegt das Rohr zur Position des gewünschten Ausgangs.

Die Komplexität wächst mit der Anzahl an Aktoren, da jeder Aktor separat angesteuert werden muss. Die Komplexität des Rohrverfahrens ist bei einer Ausgangszahl von vier oder mehr somit geringer als bei dem Binärbaumverfahren oder dem

Fließbandverfahren. Dies ist unter Umständen der Grund dafür, dass die meisten recherchierten Sortiermaschinen das Rohrverfahrens verwenden.

Aus folgendem Grund wird eine hohe Anzahl an Ausgängen bevorzugt: Wenn die Anzahl an Ausgängen geringer als die Anzahl an Farben in der zu sortierenden Steckperlenmenge ist, dann muss die Methode zur Kompensierung fehlender Ausgänge angewandt werden. Dabei bestimmt die Anzahl an Ausgängen, wie häufig eine Iteration der Methode für eine gegebene Steckperlenmenge notwendig ist. Zudem bedeutet jede Iteration einen erheblichen zusätzlichen Zeitaufwand. Eine hohe Anzahl an Ausgängen verringert somit für gewöhnlich die Zeit, die das vollständige Sortieren von Steckperlen benötigt.

Bernhard Schwerdtner zeigt mit seiner Sortiermaschine [27], dass das Rohrverfahren mit zwölf Ausgängen gut umsetzbar ist. Diese Ausgangszahl ist relativ hoch. Das ist daran zu erkennen, dass seine Maschine im Vergleich zu den anderen recherchierten Sortiermaschinenprojekten am meisten Ausgänge besitzt. Die BeadSort-Maschine soll mindestens dieselbe Anzahl an Ausgänge besitzen. Damit soll die Wahrscheinlichkeit gemindert werden, die Methode zur Kompensierung fehlender Ausgänge durchführen zu müssen.

Das Rohrverfahren wird bevorzugt, da die Komplexität auch bei hoher Anzahl an Ausgängen gering bleibt. Es wurde bereits festgelegt, dass das Ausgangsverfahren die Ausrichtung der Steckperlen beibehalten muss. Daher wird im Folgenden geprüft, ob die Beibehaltung der Ausrichtung im Rohrverfahren möglich ist. Die Idee ist, den Innendurchmesser des Rohrs so schmal zu entwerfen, dass die Steckperle ihre Ausrichtung aufgrund von Platzmangel nicht ändern kann. Eine Steckperle benötigt zur Änderung ihrer Ausrichtung in einem Rohr ihre Diagonale als Innendurchmesser des Rohrs. Abbildung 5 veranschaulicht dies. Auf der Abbildung sind Querschnitte einer Röhre zu sehen. Die Röhren besitzen einen kleineren Innendurchmesser als die Diagonale einer Steckperle. In der linken Röhre dreht sich eine Steckperle mit der maximal möglichen Rotation, da sie an die Innenwand der Röhre anstößt. Mithilfe der rechten Röhre wird gezeigt, dass eine Steckperle sich nicht in der gezeigten unmöglichen Rotation befinden kann, da der Platz in der Röhre nicht ausreicht.

3. Teilprobleme

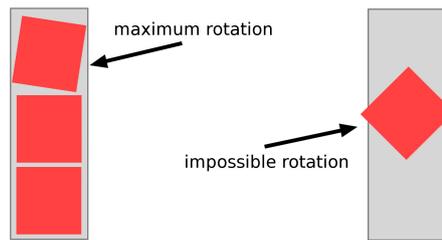


Abbildung 5: Fixierte Ausrichtung von Steckperlen in einer Röhre

Demnach kann das Rohrverfahren die Ausrichtung beibehalten und wird als Ausgabeverfahren der BeadSort-Maschine gewählt.

3.4. Farberkennung

Es ist nicht in allen recherchierten Sortiermaschinen erkenntlich, wie die Farberkennung realisiert wurde. Die Maschinen, bei denen diese Informationen verfügbar sind, verwenden ohne Ausnahme jeweils ein Farbsensormodul. Es wird daher ebenso ein Farbsensormodul verwendet. Die recherchierten Sortiermaschinen verwenden teilweise verschiedene Farbsensormodule. Die Tabelle 2 listet die verwendeten Farbsensormodule der jeweiligen Sortiermaschinen.

Entwickler	Farbsensormodul
Rolf R Bakke	ams TCS3200
C. Mei, K. Liu, V. Auduong	ams TCS34725
Bernhard Schwerdtner	ams TCS230
Geir Andersen	ams TCS230

Tabelle 2: Verwendete Farbsensormodule

Farbsensormodule eignen sich für die Erkennung einfarbiger Flächen. Alle in Tabelle 2 aufgeführten Farbsensormodule würden den Anforderungen genügen. Bei einer Recherche der aufgeführten Farbsensormodule fielen jedoch vor allem für den ams TCS34725 [7] folgende Rezensionen auf:

TCS34725 has integrated IR filter. Sold by Adafruit. This makes it perfect if you work arround Infrared. Now, if i only knew the human body emits infrared i wouldve bought this one. (arduino5 in Arduino Forum [14])

Out of these three our favorite is the Adafruit TCS34725 due its superior examples, tutorials and simple wiring.

(AlfredH1 auf Instructables [15])

Preislich unterscheiden sich die aufgeführten Farbsensormodule nicht erheblich. Aus diesen Gründen wird das Farbsensormodul TCS34725 für die Erkennung von Steckperlenfarben ausgewählt.

Es besteht die Annahme, dass die Ausrichtung einer Steckperle die Erkennung der Farbe beeinflusst. Die beliebige Ausrichtung bedeutet für den Farbsensor, dass die farbige Flächengröße aus der Perspektive des Sensors variiert. Zwei Sensorwerte derselben Steckperle in unterschiedlichen Ausrichtungen könnten somit erhebliche Abweichungen aufweisen. Die Farberkennung einer Steckperle sollte somit nach dessen Ausrichtung geschehen, um eine zuverlässigere Erkennung zu gewährleisten. Weiterhin muss die Steckperle zur individuellen Betrachtung vereinzelt sein. Eine Farberkennung während der Vereinzlung ist daher erforderlich. Die recherchierten Sortiermaschinen bestätigen dies durch Umsetzung der Farberkennung während der Vereinzlung. Die Ausrichtung der Steckperlen muss schlussfolgernd nicht nur vor der Farberkennung, sondern auch vor der Vereinzlung geschehen.

3.5. Vereinzlung

Ein Vereinzlungsverfahren muss die vorher ausgerichteten Steckperlen vereinzeln. Es wird ein Schieber verwendet, der eine einzelne Steckperle vom Ausgang des Ausrichters aufnehmen kann. Der Schieber kann eine aufgenommene Steckperle dann zum Eingang des Ausgabeverfahrens bewegen. Eine ähnliche Vereinzlungsstrategie wurde von MarVtec, Egen Riether, Sammy Linden und Henning Ziegler [28, S. 14 ff.] bereits realisiert und lieferte akzeptable Ergebnisse. Die Farberkennung kann direkt nach Aufnahme einer Steckperle im Schieber erfolgen.

3.6. Sequenzialisierung

Die Steckperlen sollen bereits vor der Vereinzlung, der Farberkennung und dem Ausgabeverfahren ausgerichtet werden. Damit ist das Ausrichtungsverfahren auf die Menge der Steckperlen anzuwenden. Das Ausrichtungsverfahren benötigt die Steckperlen jedoch in einer Röhre übereinander platziert. Daher wird im Folgenden ein Verfahren ermittelt, das die Steckperlen aus dem Eingangsbehälter in eine Röhre überträgt. Diese Übertragung wird als *Sequenzialisierung* bezeichnet, da die Steckperlen sequenziell angeordnet werden.

Die Sortiermaschine von Egen Riether sequenzialisiert das Sortiergut mittels eines Rotors. Die Rotorflügel bewegen das Sortiergut so, dass es ohne Steckenbleiben in eine Röhre fällt. Dieses Verfahren wird übernommen.

4. Entwurf

Die Teilprobleme sind an dieser Stelle gelöst. In der späteren Testphase werden die Lösungen gegebenenfalls überarbeitet. Mit OpenSCAD wird ein 3D-Modell entworfen, das die Lösungen implementiert. Das Modell wird modular aufgebaut, um eine einfache Anpassung beziehungsweise Erweiterung der Maschine zu ermöglichen. Eine Steckvorrichtung soll dabei gewährleisten, dass die Module ausreichend präzise aufeinandergesteckt werden können. Die Abbildung 6 zeigt eine frontale Gesamtansicht des 3D-Modells. Die Module befinden sich in der Reihenfolge, in der sie in der BeadSort-Maschine zusammengesetzt werden sollen.

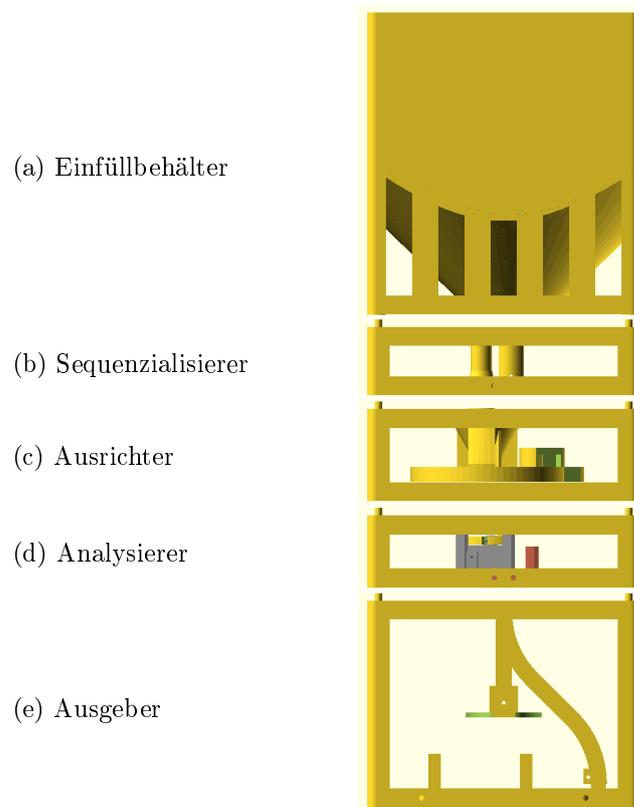


Abbildung 6: Frontseite des BeadSort-3D-Modells

Eine Einschränkung des 3D-Druckens wurde bei der Modellierung berücksichtigt. Es handelt sich dabei um die Einschränkung, dass keine Überhänge gedruckt werden können. Daher sind Teile des Modells so konzipiert, dass sie mehrteilig gedruckt und

daraufhin zusammengesetzt werden. Alle 3D-Modelle befinden sich im OpenSCAD-Format (.scad) auf der dieser Arbeit beiliegenden DVD.

Im Folgenden wird der Aufbau der Module nacheinander erläutert. Dabei wird mittels eckiger Klammern auf die entsprechende Markierungen in einem Bild verwiesen. Markierungen bestehen jeweils aus einem weißen Buchstaben in der Mitte eines rot ausgefüllten Kreises. Der Ausdruck *Abb.1 [A]* verweist zum Beispiel auf eine Markierung in Abbildung 1, die aus einem weißen A in einem rot ausgefüllten Kreis besteht.

4.1. Einfüllbehälter

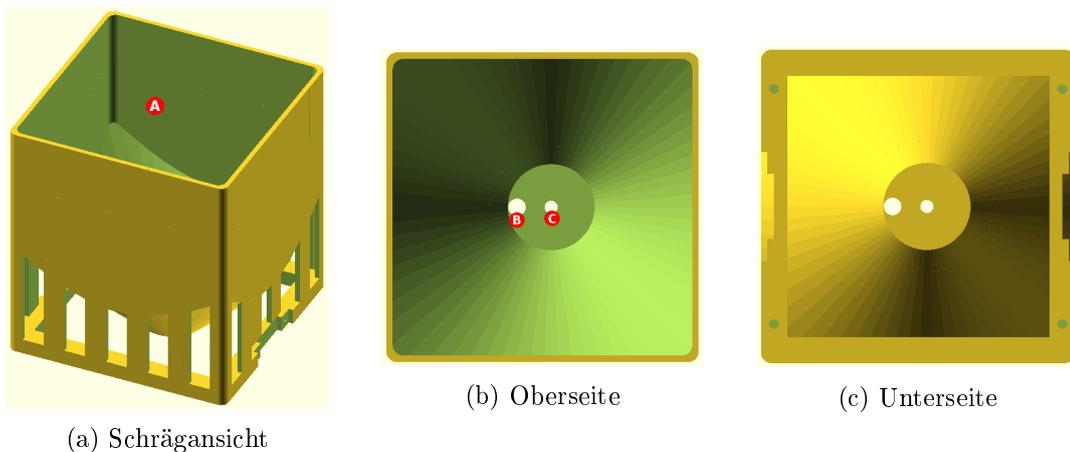


Abbildung 7: Entwurf des Einfüllbehälters

Das Modul des Einfüllbehälters besitzt einen Behälter mit Platz für ungefähr 14.000 Steckperlen (Abb.7 [A]). Typische Mengen käuflicher Steckperlensets sind 1.000-13.000 Steckperlen. So passt ein gekauftes Steckperlenset für gewöhnlich vollständig in den Behälter.

Am Boden des Behälters ist ein Loch (Abb.7 [B]), durch das jeweils eine Steckperle unausgerichtet passt. Die Größe ist so gewählt, dass keine zwei Steckperlen nebeneinander hindurch passen.

Der Sequenzialisierer besitzt einen Rotor, der über die Modulhöhe hinaus geht (siehe 4.2 Sequenzialisierer). Der Rotor ragt somit in das Modul des Einfüllbehälters hinein. Dazu ist ein Loch (Abb.7 [C]) zentral im Boden des Behälters platziert. Dadurch schaut der Rotor in den Behälter und es kann ein Aufsatz darauf platziert werden (siehe 4.2 Rotoraufsatz). Dieser Aufsatz kann dann von dem Aktor im Sequenzialisierer rotiert werden.

4. Entwurf

4.2. Sequenzialisierer

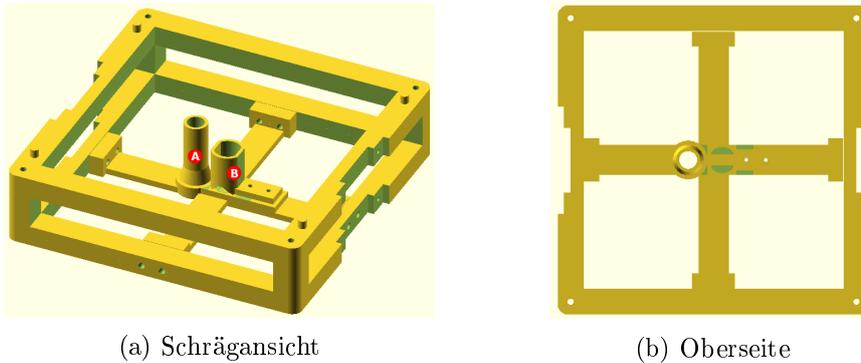


Abbildung 9: Entwurf des Sequenzialisierers

Das Modul des Sequenzialisierers besteht aus zwei Teilen. Der erste Teil ist eine Röhre, die Steckperlen weiterführt, die in das Steckperlenloch im Einfüllbehälter (Abb.7 [B]) gefallen sind. Der zweite Teil ist eine Motorhalterung, die einen Motor so fasst, dass der Rotor des Motors nach oben gerichtet ist. Wie bereits in Sektion 4.1 erwähnt, wird der Rotor des Motors über das Modul hinausragen. Auf das herausragende Stück wird dann ein Aufsatz gesteckt, der in Sektion 4.2 beschrieben wird. Als Motor wird ein Getriebemotor verwendet. Getriebemotoren weisen je nach Getriebe mehr Kraft bei geringerer Drehzahl auf. Eine große Steckperlenmenge im Einfüllbehälter erfordert möglicherweise hohe Kraft. Die geringe Drehzahl wird erfordert, damit Steckperlen eine Chance haben, durch das Steckperlenloch zu fallen. Bei zu hoher Drehzahl würde der Rotoraufsatz die Steckperlen möglicherweise über das Loch herüber bewegen. Ein Getriebemotor entspricht somit den Anforderungen. Es wird eine Variante des Micro N20 Getriebemotor verwendet. [5] Diese Variante besitzt laut Spezifikation eine Nennspannung von 6V, eine Drehzahl von 60 UpM und ein Drehmoment von 0.0147 Nm.

Rotoraufsatz

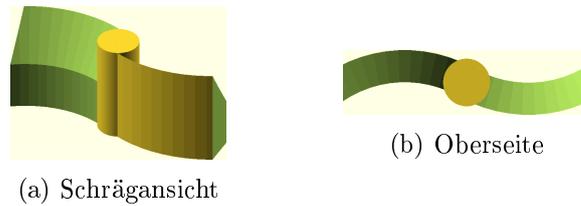


Abbildung 10: Entwurf des Rotoraufsatzes

Der Rotoraufsatz befindet sich zentral im Einfüllbehälter und bewegt die Steckperlen im Inneren des Behälters. Die Steckperlen sollen durch das Steckperlenloch im Einfüllbehälter (Abb.7 [B]) fallen. Dabei entstehende Verklümmungen am Eingang des Steckperlenloches sollen durch die Rotation des Aufsatzes aufgelöst werden.

4.3. Ausrichter

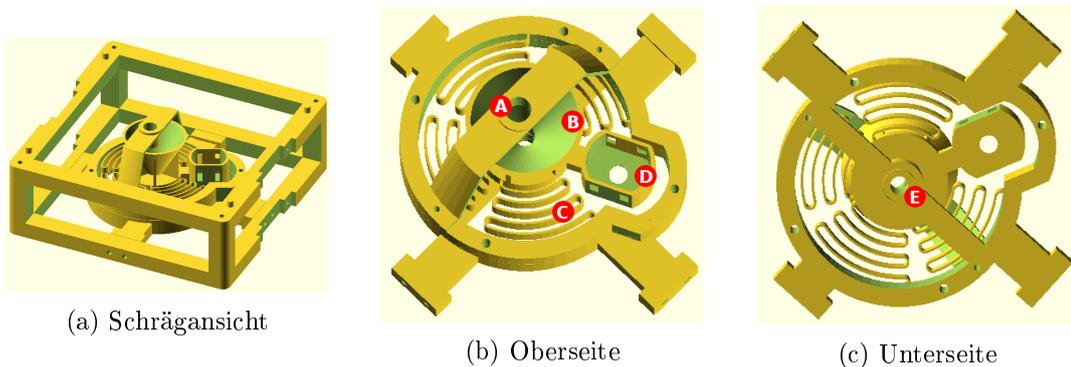
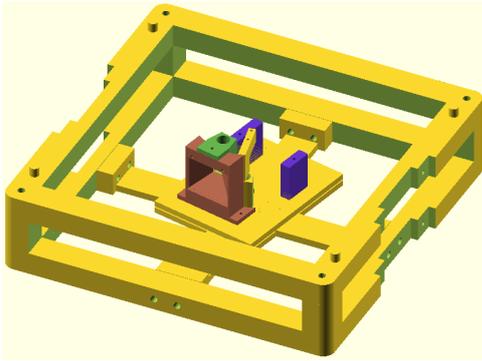


Abbildung 11: Entwurf des Ausrichters

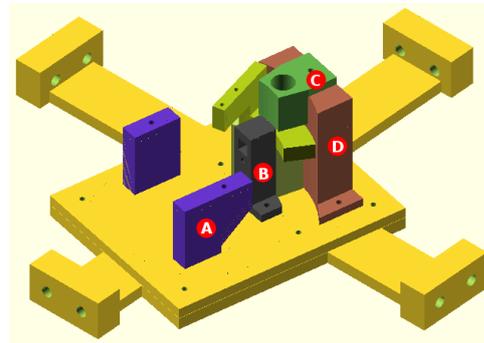
Das Modul des Ausrichters besitzt einen gefederten Innenteil. Der Innenteil besteht aus einem Trichter (Abb.11 [B]) und einer Motorhalterung (Abb.11 [D]). Der Innenteil kann aufgrund der Federung (Abb.11 [C]) in Vibration gesetzt werden, obwohl der Rest des Moduls still steht. Die Steckperlen können über die kurze Röhre (Abb.11 [A]) unausgerichtet in den Trichter eingegeben werden. Die Steckperlen können aufgrund des Innendurchmessers nur ausgerichtet durch die kurze Röhre am Ausgang (Abb.11 [E]) fallen. Als Basis des Entwurfs diente das Modell von Scanlime auf Thingiverse [24].

4. Entwurf

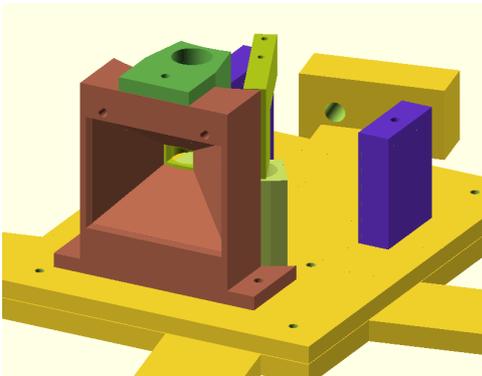
4.4. Analysierer



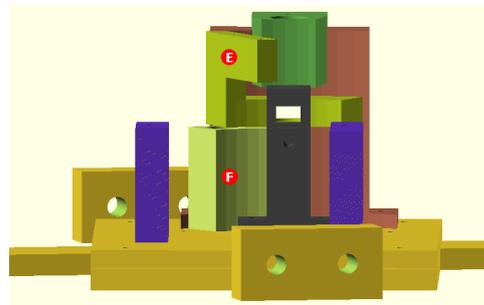
(a) Schrägansicht



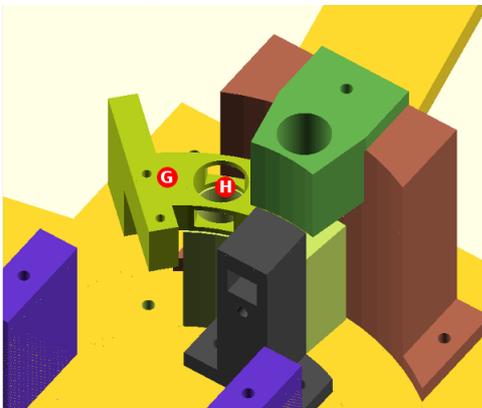
(b) Schrägansicht ohne Container 1



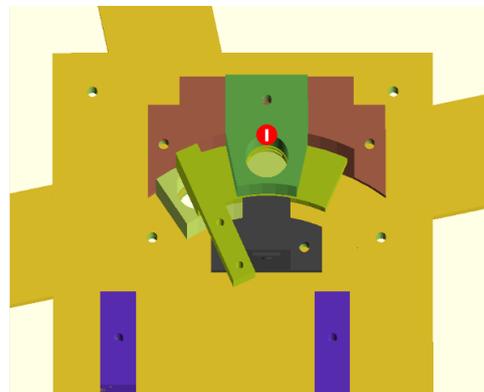
(c) Schrägansicht ohne Container 2



(d) Betrachtungsloch



(e) Servoarm am Ausgang



(f) Oberseite

Abbildung 12: Entwurf des Analysierers

Das Modul des Analysierers hat drei Aufgaben. Die erste Aufgabe ist das Vereinzeln der Steckperlen. Dazu wird ein Servo in die Servohalterung (Abb.12 [A]) einge-

setzt. Der Servo kann den Vereinzelungsarm (Abb.12 [E]) entlang dem Ausgangsbogen (Abb.12 [F]) bewegen. Eine Vereinzelung findet dann wie folgt statt. Der Vereinzelungsarm wird so positioniert, dass sich das Steckperlenloch im Arm (Abb.12 [H]) unter dem Steckperlenloch von der Eingangsröhre (Abb.12 [I]) befindet. Diese Position ist in Abbildung 12f gut erkennbar. Durch die Eingangsröhre kann dann eine Steckperle in den Vereinzelungsarm fallen. Der Servo kann den Arm dann so positionieren, dass sich das Steckperlenloch des Arms über dem Steckperlenloch der Ausgangsröhre befindet. Diese Position ist in der Abbildung 12e gut erkennbar.

Die zweite Aufgabe ist die Erkennung der Farbe einer Steckperle. Dazu existiert ein Fenster im Vereinzelungsarm, durch das eine vereinzelte Steckperle betrachtet werden kann. An der Seite des Fensters befindet sich eine Sensorhalterung (Abb.12 [D]), die ebenfalls ein Fenster auf derselben Höhe hat. Der Farbsensor, der an die Sensorhalterung angebracht wird, kann somit die vereinzelte Steckperle durch die Fenster hindurch betrachten.

Die dritte Aufgabe ist die Erkennung, ob eine Steckperle in den Vereinzelungsarm gefallen ist. Im Weiteren wird diese Erkennung als *Existenzerkennung* bezeichnet. Wenn diese Erkennung fehlen würde, ließen sich die Steckperlen immer noch erfolgreich sortieren. Die Farberkennung und die darauffolgende Vereinzelung könnten periodisch ausgeführt werden. In dem Fall, dass eine Steckperle in den Vereinzelungsarm gefallen ist, würde diese korrekt sortiert werden. Wenn dies nicht der Fall ist, dann würde die Farberkennung ein undefiniertes Ergebnis liefern. Das undefinierte Ergebnis führt zu der Wahl eines undefinierten Ausgang. Da die Vereinzelung jedoch keine Steckperle vereinzelt und damit auch keine Steckperle in das Ausgabeverfahren weiterführt, wird auch keine Steckperle falsch sortiert. Eine fehlende Erkennung des Vorliegens einer Steckperle führt aber dazu, dass keine korrekte Angabe über die Menge der gescannten Steckperlen einer bestimmten Farbe gemacht werden kann. Die korrekte Angabe der Mengen ist aber notwendig, um Puzzlesets erstellen zu können. Es muss bekannt sein, wieviele Steckperlen einer Farbe bereits für das Puzzle gefunden wurden, um zu wissen, wieviele Steckperlen der jeweiligen Farben noch fehlen. Aus diesem Grund wurde die Erkennung des Vorliegens einer Steckperle implementiert. Für diese Erkennung besitzt der Vereinzelungsarm ein zweites Fenster auf der gegenüberliegenden Seite des ersten Fensters. Durch dieses Fenster leuchtet eine LED, wenn geprüft werden soll, ob eine Steckperle vorliegt. Diese LED und dessen Licht wird im Weiteren jeweils als *Erkennungs-LED* und *Erkennungslicht* bezeichnet. Der Farbsensor kann anhand des Lichteinflusses erkennen, ob das LED-Licht direkt auf den Sensor strahlt. Wenn sich eine Steckperle im Vereinzelungsarm befindet, dann

4. Entwurf

wird der Großteil des LED-Lichtes blockiert und der Sensor erkennt, dass sich eine Steckperle im Vereinzelungsarm befindet.

4.5. Ausgeber

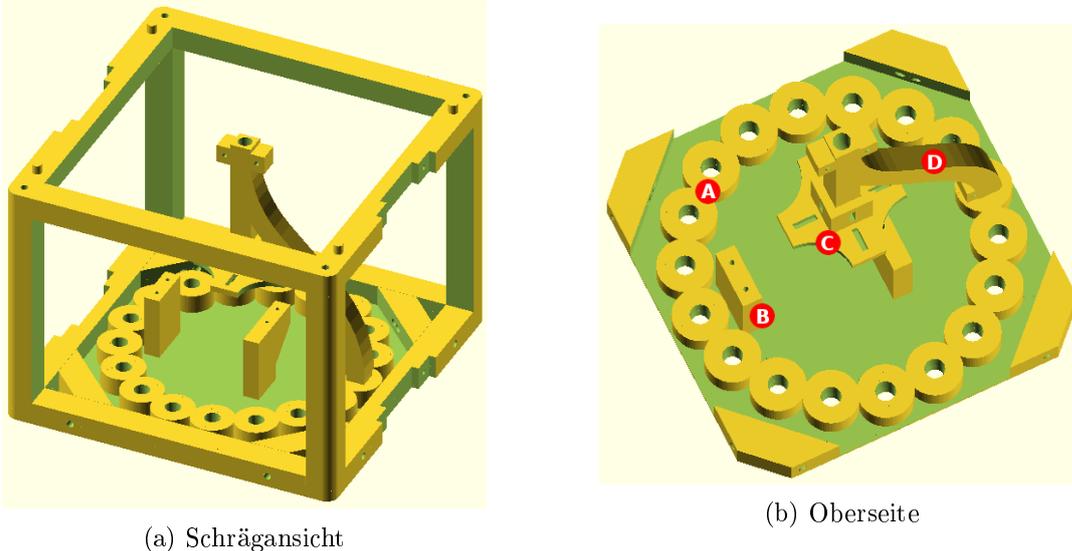


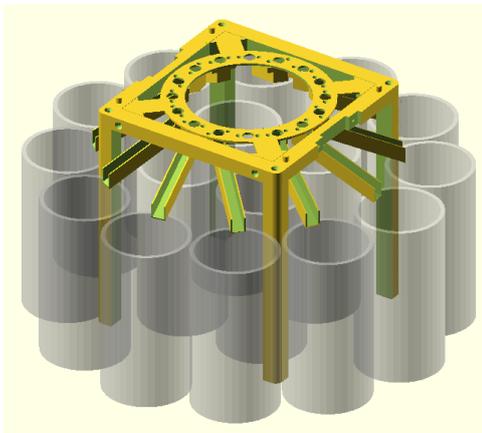
Abbildung 13: Entwurf des Ausgebers

Das Modul des Ausgebers besitzt eine Röhre (Abb.13 [D]), dessen Ende über eines der Ausgangslöcher (Abb.13 [A]) bewegt wird. Die Bewegung wird mittels eines Servos mit der Bezeichnung TowerPro MG995 erfolgen. Dieser kann in der Halterung (Abb.13 [B]) befestigt werden. Dann kann die Röhre mithilfe des Adapters (Abb.13 [C]) an dem Rotor des Servos befestigt werden.

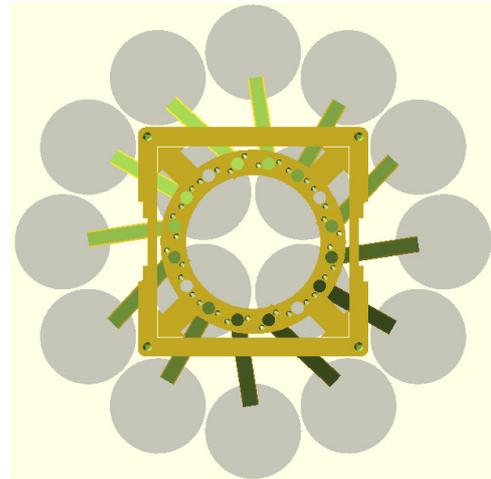
4.6. Stand

Es wurde ein Stand entworfen, auf den die BeadSort-Maschine aufgesetzt werden kann. In Abbildung 14 ist der Entwurf des Stands abgebildet. Die grau durchsichtigen Zylinder stellen jeweils ein Ausgangsbehälter dar. In den Ausgangsbehältern sollen die sortierten Steckperlen ausgegeben werden. Die Verwendung von einzelnen Behältern ermöglicht ein einfaches Entnehmen und Austauschen der sortierten Steckperlen. Dieser Stand besitzt Rutschen, die die Steckperlen von den Ausgängen der BeadSort-Maschine zu jeweiligen Ausgangsbehältern führen sollen. Im folgenden werden diese Rutschen als *Ausgangsrutschen* bezeichnet. Die Ausgangsbehälter werden auch als *Ausgangsgläser* bezeichnet, da Gläser als Ausgangsbehälter genutzt werden.

Es wird eine Holzboden konstruiert, in die der Stand und die Ausgangsbehälter eingesetzt werden können. So lässt sich die BeadSort-Maschine trotz der vielen einzelnen Behälter leichter transportieren.



(a) Schrägansicht



(b) Oberseite

Abbildung 14: Entwurf des Stands

5. Test und Optimierung

5.1. Ansteuerung

Der Getriebemotor im Sequenzialisierer und der Gleichstrommotor im Ausrichter werden über einen Motortreiber betrieben. Der verwendete Motortreiber ist ein Modul, auf dem ein L298N-Chip verbaut ist [4]. Dieses Modul kann zwei Motoren bidirektional ansteuern. Für die bidirektionale Ansteuerung von zwei Motoren bekommt das Modul neben der Versorgungsspannung vier PWM-Signale: für jede Richtung jedes Motors je ein Signal. Da der Gleichstrommotor im Ausrichter lediglich zur Erzeugung von Vibration dient, muss sich der Motor nur in eine Richtung drehen können. Daher genügt ein PWM-Signal für die Ansteuerung des Ausrichters. Es ist nicht auszuschließen, dass der Sequenzialisierer beide Drehrichtungen des Motors benötigt. Daher werden vorsorglich zwei PWM-Signale für die Ansteuerung des Sequenzialisierers verwendet. Insgesamt werden somit drei PWM-Signale benötigt.

Zur Ansteuerung der Hardware wird ein Raspberry Pi 3 Model B dienen. Dieser verfügt über ausreichend PWM-Schnittstellen, um die Servos, die LEDs und den Motortreiber anzusteuern. Außerdem verfügt er über eine I2C-Schnittstelle, die erforderlich ist, um das Farbsensormodul zu verwenden.

Testansteuerung

Die Hardware und die BeadSort-Module werden schon vor der Entwicklung der Bediensoftware getestet. Entsprechende Erkenntnisse könnten unter Umständen Anpassungen an der BeadSort-Maschine erfordern, die einen erheblichen Einfluss auf die Bediensoftware haben. Das Risiko, die Bediensoftware nachträglich umschreiben zu müssen, soll durch vorherige Tests gemindert werden. Ein Arduino Mikrocontroller-Board wird verwendet, um diese Tests auszuführen. Die Arduino Mikrocontroller-Boards sind sehr einsteigerfreundlich und lassen sich mit geringem Aufwand programmieren. Es wird der Arduino Nano gewählt. Dieser verfügt über ausreichend Schnittstellen und ist günstig zu erwerben.

5.2. Sequenzialisierer

Für den Test des Sequenzialisierers wird das Modul des Einfüllbehälters mit dem des Sequenzialisierers zusammengesteckt. Dann wird der Rotoraufsatz auf den im Einfüllbehälter befindlichen Rotor gesteckt. Es werden ungefähr 100 Steckperlen in den Einfüllbehälter eingegeben und daraufhin der Getriebemotor gestartet.

Das Ergebnis ist wie folgt:

Nach wenigen Steckperlen verklemmte sich eine Steckperle zwischen einem Flügel des Rotoraufsatzes und der Steckperlenloch im Einfüllbehälter. Der Motor musste abgeschaltet und die Verklemmung per Hand aufgehoben werden. Dann wurde der Motor wieder eingeschaltet, doch die Verklemmung trat erneut nach wenigen Steckperlen auf. Dieses Problem wiederholte sich auch bei weiteren Versuchen.

Der Sequenzialisierer ist somit zu unzuverlässig und benötigt ein viel zu häufiges manuelles Eingreifen. Aus diesem Grund wird der Rotoraufsatz so verändert, dass keine Verklemmungen mehr direkt am Steckperlenloch auftreten können. Dazu werden die 3D-gedruckten Flügel des Rotoraufsatz gegen eine Bürste getauscht. Die Bürste soll durch die Nachgiebigkeit der Borsten Verklemmungen vermeiden.

Um die Rotoraufsätze im folgenden zu unterscheiden, wird der vorherige Rotoraufsatz als *Flügelauflaufsatz* und der neue Rotoraufsatz als *Bürstenaufsatz* bezeichnet.

Der Test wird mit dem Bürstenaufsatz wie zuvor durchgeführt. Es werden wieder ungefähr 100 Steckperlen verwendet. Alle Steckperlen fallen nacheinander durch das Steckperlenloch. Sie wurden somit vollständig sequenzialisiert. Es werden zwei Wiederholungen des Tests gemacht und auch diese werden fehlerfrei abgeschlossen.

Der Test wird nun mit ungefähr 1000 Steckperlen ausgeführt. Dabei fällt auf, dass der Getriebemotor wesentlich schwerer läuft. Die Steckperlen werden jedoch noch vollständig sequenzialisiert, ohne dass ein Eingriff notwendig wird. Es besteht der Verdacht, dass der Getriebemotor die Last einer größeren Steckperlenmenge nicht mehr bewegen kann. Daher wird nun eine Menge von ungefähr 3000 Steckperlen in den Einfüllbehälter gegeben und der Test ein erneutes Mal ausgeführt. Während des Tests stockt der Getriebemotor häufig und kommt schließlich zum Stillstand. Der Test wird abgebrochen.

Die maximal mögliche Eingabemenge von Steckperlen müsste auf ungefähr 1000 festgelegt werden, um eine erfolgreiche Sequenzialisierung zu ermöglichen. Dies stellt jedoch eine große Einschränkung dar. Daher wird nach einer Lösung gesucht, die das Problem des Motorstillstands löst. Ein Lösungsansatz wäre die Verwendung eines stärkeren Motors. Es müsste jedoch unter Umständen auch ein stärkerer Motortreiber verwendet werden. Sowohl der Motor als auch der Motortreiber würden weitere Kosten verursachen. Weiterhin müsste die Motorhalterung überarbeitet werden. Ein zu starker Motor könnte zudem die Steckperlen verformen.

Ein weiterer Lösungsansatz wäre die Entwicklung eines Rotoraufsatzes, der den Motor weniger belastet. Der Wechsel eines Rotoraufsatzes ist unerheblicher als der Wechsel zu einem anderen Motor. Zudem entstehen lediglich die Kosten des 3D-Druckes des

5. Test und Optimierung

neuen Rotoraufsatzes.

Der Lösungsansatz des besseren Rotoraufsatzes ist dem des stärkeren Motors vorzuziehen. Jedoch ist nicht abzuschätzen, wie lange die Entwicklung eines solchen Rotoraufsatzes benötigt. Die Sortiermaschine von Gabriel Schneider besitzt jedoch einen Sequenzialisierer, dessen Konzept den Motor womöglich relativ gering belastet. Daher wird vorerst dieser Lösungsansatz verfolgt und der folgende Entwurf entwickelt:

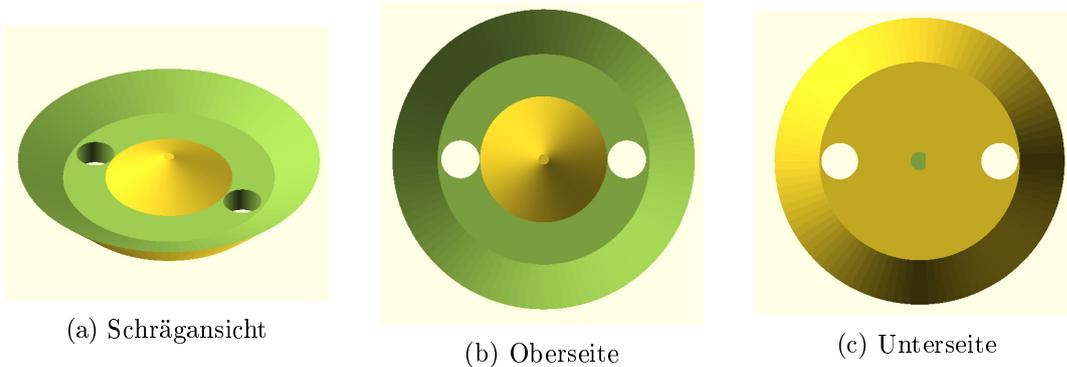


Abbildung 15: Entwurf des Telleraufsatzes

Der Aufsatz wird aufgrund der Form im Folgenden als *Telleraufsatz* bezeichnet. Der Aufsatz besitzt ein Loch, in das eine einzelne Steckperle reinfallen kann. Mit der Rotation des Aufsatzes befindet sich das Loch kurzzeitig über dem des Einfüllbehälters. Zu diesem Zeitpunkt kann die Steckperle in das Steckperlenloch des Einfüllbehälters fallen. Die Rotation des Aufsatzes soll bewirken, dass eine neue Steckperle in das Loch des Aufsatzes fallen kann.

Der neue Rotoraufsatz wird in den Einfüllbehälter eingesetzt. 100 Steckperlen werden eingegeben. Dann wird der Test wie zuvor durchgeführt. Nach einigen sequenzialisierten Steckperlen tritt eine Verklemmung auf. Eine Steckperle klemmt zwischen dem Loch des Rotoraufsatzes und dem Steckperlenloch des Einfüllbehälters. Die Verklemmung muss von Hand aufgelöst werden. Danach wird der Test fortgeführt. Nach einigen sequenzialisierten Steckperlen tritt erneut eine Verklemmung an derselben Stelle auf. Dies wiederholt sich alle 6 bis 12 Steckperlen.

Die Verklemmungen treten mit dem Telleraufsatz seltener als mit dem Flügelaufsatz auf. Daher ist neben einer Verklemmungsvermeidung auch eine Verklemmungsauflösung zu erwägen. Eine Verklemmung kann mit einer Rotation des Aufsatzes in die entgegengesetzte Richtung gelöst werden. Dies kann mittels entsprechender Ansteuerung an den Motortreiber ohne weiteres umgesetzt werden. Die Entwicklung einer Verklemmungserkennung könnte sich jedoch als sehr aufwändig erweisen. Es wird

eine periodische Verklemmungsauflösung bevorzugt. Dazu wird während der Sequenzialisierung alle fünf Sekunden eine halbe Sekunde in die entgegengesetzte Richtung rotiert, um mögliche Verklemmungen aufzulösen. Dies verringert die Effizienz, stellt aber eine einfach zu implementierende Lösung dar.

Eine Wiederholung des Tests mit der Verklemmungsauflösung sequenzialisiert alle 100 Steckperlen erfolgreich. Es ist zu sehen, wie die auftretenden Verklemmungen nach kurzer Zeit korrekt aufgelöst werden. Zwei weitere Wiederholungen des Tests werden durchgeführt und verlaufen ebenfalls erfolgreich.

Der Test wird zwei weitere Mal mit jeweils ungefähr 1000 und 10000 eingegebenen Steckperlen durchgeführt. In beiden Tests kann sich der Motor problemlos drehen. Aus Zeitgründen wurden die Tests nach jeweils ungefähr 100 sequenzialisierten Steckperlen abgebrochen.

Die Sequenzialisierung läuft nun zuverlässig.

5.3. Ausrichter

Für den Test des Ausrichters werden die Module des Einfüllbehälters und des Sequenzialisierers mit dem des Ausrichters zusammengesteckt. So kann zusätzlich zum Ausrichten zugleich die Kompatibilität des Ausrichters mit dem Sequenzialisierer getestet werden. Es werden ungefähr 100 Steckperlen in den Einfüllbehälter gegeben. Dann wird der Vibrationsmotor und schließlich der Getriebemotor gestartet. Alle Steckperlen werden erfolgreich sequenzialisiert und ausgerichtet.

5.4. Vereinzeln

Für den Test wird lediglich das Modul des Analysierers verwendet. Der Arduino wird wie folgt programmiert. Der Vereinzelnarm positioniert sich zu Beginn so, dass sich das Loch im Vereinzelnarm direkt unter dem Loch der Eingangsröhre befindet. Nach vier Sekunden wird der Vereinzelnarm so positioniert, dass sich das Loch im Vereinzelnarm direkt über dem Loch der Ausgangsröhre befindet. Nach einer weiteren Sekunde beginnt das Programm von vorne.

Der Test wird folgendermaßen durchgeführt. Das zuvor beschriebene Programm wird zu Beginn gestartet. Dann wird die Eingangsröhre von Hand mit Steckperlen befüllt. Sobald der Vereinzelnarm eine Steckperle entnimmt, wird sofort eine weitere Steckperle nachgefüllt.

Der Test zeigt, dass Steckperlen häufig nicht in das Loch des Vereinzelnarmes fallen, weil dieser nicht ausreichend genau positioniert wurde. Das gleiche Problem tritt

5. Test und Optimierung

an der Ausgangsröhre auf. Die Steckperle wird entweder zurück zur Eingangsröhre bewegt oder verhakt sich zwischen der Ausgangsröhre und dem Vereinzlungsarm. Mittels einer Hin- und Herbewegung des Vereinzlungsarmes soll die Positionsabweichung des Vereinzlungsarmes eine gewisse Toleranz erhalten. Eine Steckperle soll damit sowohl zuverlässiger aus der Eingangsröhre in das Loch des Armes, als auch aus dem Loch des Armes in die Ausgangsröhre fallen. Diese Hin- und Herbewegung wird im Folgenden als *Wiggle* (zu Deutsch: wackeln) bezeichnet.

Nach der Implementation des Wiggles wird der Test wiederholt. Es wurden 20 Steckperlen in die Eingangsröhre gegeben. Alle Steckperlen wurden erfolgreich separiert. Die Separierung läuft nun zuverlässig.

5.5. Analysierer

Für die folgenden Tests wird lediglich das Modul des Analysierers verwendet.

5.5.1. Existenzerkennung

Zunächst wird getestet, ob das Eintreffen einer Steckperle zuverlässig erkannt werden kann. Die Existenz-LED wird eingeschaltet und der Vereinzlungsarm wird so positioniert, dass sich das Steckperlenloch des Vereinzlungsarms unter dem der Eingangsröhre befindet. In dieser Position kann das Existenzlicht durch die Fenster des Vereinzlungsarms auf den Sensor leuchten. Die Sensor-LED, die für die Beleuchtung der Steckperle zur Erkennung der Farbe zuständig ist, wird ausgeschaltet. Der vom Sensor ermittelte Lichtwert wird notiert. Dieser Wert beträgt 124. Dann wird eine Steckperle in den Vereinzlungsarm gesetzt und der Lichtwert erneut ermittelt. Der Wert beträgt nun 1. Eine Steckperle blockiert das Existenzlicht daher wie gefordert. Der Mittelwert der beiden Werte beträgt gerundet 63 und wird im Folgenden als Schwellenwert gewählt. Wenn der Sensor einen Wert niedriger als den Schwellenwert erkennt, dann befindet sich vermutlich eine Steckperle im Vereinzlungsarm. Wenn der Sensor einen Wert gleich oder größer als den Schwellenwert erkennt, dann befindet sich vermutlich keine Steckperle im Arm.

Das Programm zur Vereinzlung, das für den Arduino geschrieben wurde, wird um die Existenzerkennung erweitert. Sobald eine Steckperle erkannt wird, soll diese einzeln getrennt werden. Das erleichtert das Testverfahren, da kein Entfernen der Steckperlen von Hand nötig ist. Beim Start des Programms wird der Wiggle solange wiederholt, bis eine Steckperle eintrifft. Dazu wird nach jedem Wiggle der aktuelle Lichtwert ermittelt. Ein Wiggle benötigt 0,5 Sekunden. Sobald der Lichtwert unter den Schwellen-

wert fällt, ist wahrscheinlich eine Steckperle eingetroffen und es wird die Vereinzelung durchgeführt.

Der Test wurde nach der Programmierung des Arduinos durchgeführt. Das Programm wurde gestartet und es wurden 50 Steckperlen nacheinander eingegeben. Alle Steckperlen wurden erfolgreich erkannt und vereinzelt. Die Existenzerkennung läuft damit zuverlässig.

Um die Kompatibilität der Module zu testen, wird das Modul des Analysierers mit den Modulen des Ausrichters, des Sequenzialisierers und des Einfüllbehälters zusammengesteckt. Dann wurden 100 Steckperlen in den Einfüllbehälter eingegeben. Die Sequenzialisierung, Ausrichtung und Vereinzelung wurden dann gestartet. Vereinzelt wird eine Steckperle, sobald sie von der Existenzerkennung erkannt wird. Alle eingegebenen Steckperlen konnten erfolgreich verarbeitet werden.

5.5.2. Farberkennung

Der Farbsensor stellt die ermittelte Farbe als Werte im RGB-Raum bereit. Jeder Wert (also Rotwert, Grünwert und Blauwert) wird als 16-Bit-Wert an den Mikrocontroller übertragen. Der Farbsensor wird zunächst mit einer roten Steckperle getestet. Es wird erwartet, dass der Sensor einen hohen Rotwert zurückgibt, während die Grün- und Blauwerte an 0 grenzen. Im Folgenden werden Farbwerte nach folgendem Schema angegeben: Rotwert/Grünwert/Blauwert. Die rote Steckperle wird in den Vereinzelungsarm platziert. Dann wird der RGB-Farbwert mithilfe des Sensors ermittelt. Der Wert beträgt 74/67/44. Dieser Wert weicht stark von dem erwarteten Farbwert ab. Vorallem der Grünwert ist ähnlich hoch wie der Rotwert. Auch der Blauwert ist näher am Rotwert als erwartet. Es wird vermutet, dass die Entfernung des Sensors von der Steckperle zu groß ist und der Sensor viel Licht auffängt, das nicht von der Steckperle zurückgeworfen wird. Um diese Vermutung zu überprüfen wird der Sensor aus dem Modul entnommen und auf einen näheren Abstand zu der Steckperle gebracht. Bei einem Abstand von 3mm vom Sensor zur Steckperle wird 392/114/117 als Farbwert zurückgeliefert. Dieser Wert liegt wesentlich näher an den Erwartungen. Der Rotwert ist nun mehr als dreimal so hoch wie jeweils der Blau- und Grünwert. Das Umgebungslicht (z.B. Zimmerlampen) wurde während der Farberkennung minimal gehalten und wird somit als Grund für die erhöhten Grün- und Blauwerte ausgeschlossen. Die Blau- und Grünwerte könnten bedeuten, dass die Farbe der Steckperle jeweils aus etwas Grün und Blau besteht. Dieses Ergebnis wird vorerst als ausreichend angesehen und eine neue Version des Analysierers wird entwickelt, um den Sensorabstand zu korrigieren. Die Abbildung 16 zeigt die neue Version.

5. Test und Optimierung

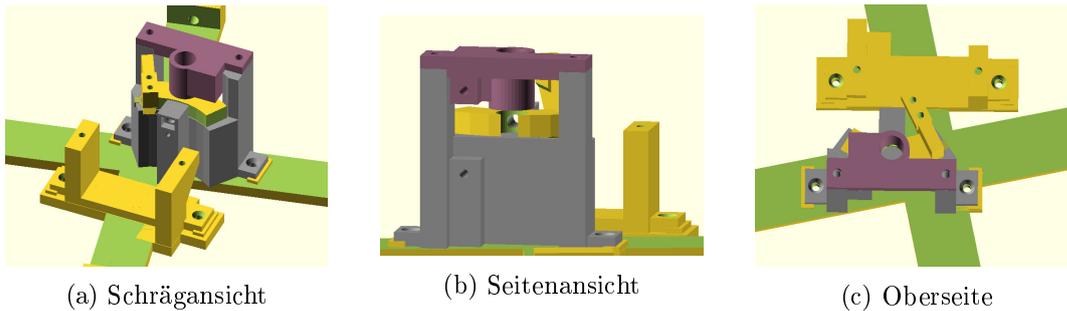


Abbildung 16: Entwurf des Analysierers - Version 2

Das Programm auf dem Arduino wird um eine Farberkennung erweitert. Dazu wird nach der Existenzerkennung und vor der Vereinzelung die Sensor-LED eingeschaltet und der Farbwert mittels Sensor gelesen. Die Sensor-LED wird dann wieder ausgeschaltet und die Vereinzelung wird durchgeführt. Es soll getestet werden, ob der Sensor identische Werte liefert, wenn die Farbe identisch ist. Dieselbe graue Steckperle wird dreimal in die Eingangsröhre des neuen Analysierers gegeben. Folgende Tabelle zeigt die jeweils gelesenen Farbwerte:

Nummer	Farbwert
1	1001/1624/922
2	352/521/276
3	1203/1859/1082

Die maximale Abweichung der Farbwerte betragen 649, 1103 und 646 in jeweils dem Rot-, Grün- und Blauwert. Die Abweichungen sind damit sehr hoch. Jedoch scheint es so, als sei das Verhältnis der Rot- Grün- und Blauwerte ähnlich. Um dies zu überprüfen, werden die jeweiligen Verhältnisse berechnet:

Nummer	R/G/B-Verhältnis
1	28,22%/45,79%/25,99%
2	30,64%/45,34%/24,02%
3	29,03%/44,86%/26,11%

Die maximale Abweichung der Verhältnisse betragen 2,42%, 0,93% und 2,09% für jeweils das Rot-, Grün- und Blauverhältnis. Die Farbverhältnisse ähneln sich somit. Das bedeutet, dass sich die Farbwerte hauptsächlich durch ihre Intensität unterscheiden. Es besteht die Vermutung, dass der Sensor das von der Steckperle reflektierte Licht unterschiedlich stark aufgenommen hat. Bei näherer Betrachtung des Vereinzelungsarms und der einfallenden Steckperlen wird deutlich, dass die Steckperlen viel Bewegungsfreiheit im Loch des Vereinzelungsarms besitzen. Dadurch befinden sich die Steckperlen beim Einwurf in geringfügig unterschiedlichen Positionen während der

Farberkennung. Dies führt unter Umständen zu den gezeigten Intensitätsabweichungen. Die Bewegungsfreiheit ist jedoch notwendig, damit die Steckperlen zuverlässig in den Vereinzelungsarm fallen. Zur Korrigierung der Positionsabweichung wird daher eine Methode entwickelt, die die Steckperle an dieselbe Position im Vereinzelungsarm bewegt. Diese Methode wird im Folgenden *Positionskorrektur* genannt. Um die Korrektur vorzunehmen, wird der Vereinzelungsarm lediglich vor der Farberkennung hin- und herbewegt. Der Bewegungsradius ist dabei etwas höher als der Bewegungsfreiraum, sodass die Steckperle in jedem Fall mitbewegt wird. Die graue Steckperle aus dem vorherigen Test wird verwendet um zu überprüfen, ob die Positionskorrektur eine Verbesserung darstellt. Das Programm auf dem Arduino wird zuvor erweitert, um die Positionskorrektur vorzunehmen. Dann wird die graue Steckperle fünfmal in die Eingangsröhre gegeben und der Farbwert mittels Sensor gelesen.

Nummer	Farbwert
1	1032/1649/943
2	1012/1602/914
3	1121/1772/1022
4	1123/1779/1025
5	929/1518/870

Die maximalen Abweichungen der Farbwerte betragen nun 194, 261 und 155 in jeweils dem Rot-, Grün- und Blauwert. Damit sind die Farbwerte deutlich konsistenter geworden. Die Abweichungen betragen jedoch noch bis zu 20,9% des Farbwertes.

Ein höherer Sensorabstand könnte bewirken, dass mögliche Abweichungen ein geringeres Ausmaß besitzen. Eine Abweichung von 0,5mm führt bei einem Sensorabstand von 3mm beispielsweise zu einer prozentualen Abweichung von ca. 16,7%. Dieselbe Abweichung führt bei einem Sensorabstand von 5mm hingegen zu einer prozentualen Abweichung von nur ca. 10%. Ein Test soll diese Vermutung überprüfen. Der Sensor wird auf den neuen Abstand gebracht und der vorherige Test wird wiederholt.

Nummer	Farbwert
1	1171/1233/811
2	1180/1264/833
3	1172/1242/824
4	1186/1269/836
5	1240/1321/879

Die maximalen Abweichungen der Farbwerte betragen nun 69, 88 und 68 in jeweils dem Rot-, Grün- und Blauwert. Die Abweichungen betragen damit nur noch bis zu 8,4% des Farbwertes. Diese Ergebnisse werden vorerst als ausreichend bewertet.

Um Farben erkennen zu können, müssen sie vorher eingespeichert worden sein. Die

5. Test und Optimierung

Entwicklung eines Arduinoprogramms zur Einspeicherung von Farben wird als zeitaufwändig eingeschätzt. Daher wird an dieser Stelle die eigentliche Software für den Raspberry Pi programmiert, die eine solche Einspeicherung von Farben enthält. Damit soll der Aufwand nicht zweimal aufgebracht werden müssen. Die für den Arduino programmierten Softwareteile werden in die Software für den Raspberry Pi übertragen. In Sektion 6 wird näher auf die Software eingegangen.

Nun werden einige Steckperlenfarben eingespeichert. Dazu wird eine Steckperle der jeweiligen Farbe eingegeben und die Einspeicherung auf dem Raspberry Pi gestartet. Damit liegen nun Farbwerte vor, mit denen eingegebene Steckperlen abgeglichen werden können. Um die zuvor behandelten Abweichungen zu berücksichtigen, muss die Abgleichung eine gewisse Toleranz besitzen. Der Toleranzwert muss ausreichend groß sein, um eine Steckperlenfarbe zuverlässig zu erkennen. Gleichzeitig muss der Wert jedoch auch möglichst gering sein, um Fehlerkennungen zu vermeiden.

Um einen guten Toleranzwert zu finden, wird wie folgt vorgegangen. Es werden die drei Steckperlenfarben schwarz, weiß und grau gewählt. Der Farbwert der entsprechenden Steckperlen wird jeweils fünfmal eingelesen. Aus den Farbwerten wird jeweils der Durchschnittsfarbwert der Extrema und die maximale Abweichung gebildet. In der Tabelle 3 sind die Ergebnisse zu sehen. Die zuvor bereits eingelesenen Werte einer grauen Steckperle wurden wiederverwendet.

Farbe	Farbwerte	Durchschnittsfarbwerte der Extrema	Max. Abw.
schwarz	276/268/172 294/295/185 280/276/174 262/273/172 254/265/166	274/280/175	40/30/19
weiß	3577/3635/2400 3624/3710/2445 3639/3699/2433 3780/3890/2551 3741/3818/2508	3678/3762/2475	203/255/151
grau	1171/1233/811 1180/1264/833 1172/1242/824 1186/1269/836 1240/1321/879	1206/1277/845	69/88/68

Tabelle 3: Durchschnittsfarbwerte und maximale Abweichungen

Dann wird die Beziehung der Komponenten der Durchschnittsfarbwerte zu den jeweiligen Komponenten der maximalen Abweichung betrachtet. Dazu wurde folgendes Streudiagramm inklusive Regressionsgerade erstellt.

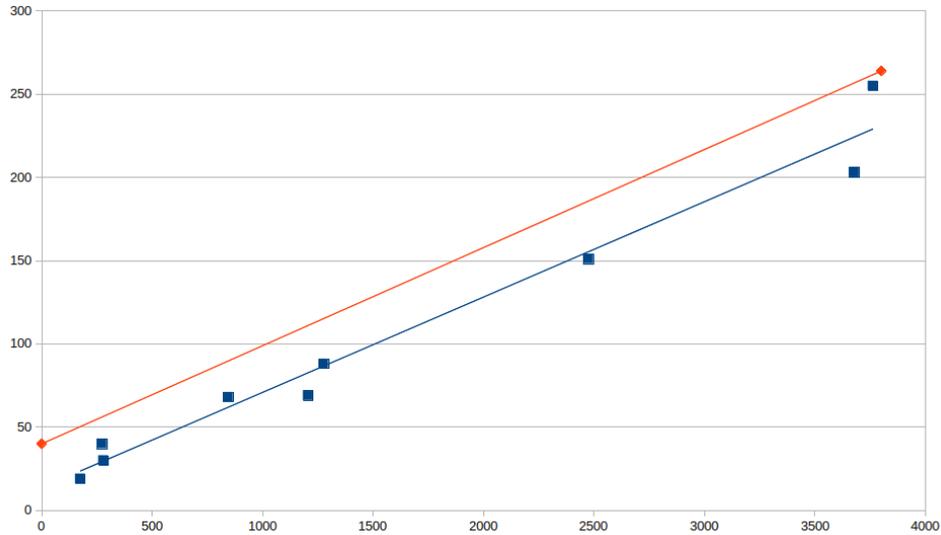


Abbildung 17: Beziehung der Farbwerte zur Abweichung

Die rote Gerade wurde von Hand hinzugefügt und wird im Folgenden als Toleranzgerade bezeichnet. Die Steigung der Toleranzgeraden ähnelt der Regressionsgeraden. Die Toleranzgerade ist jedoch so weit an der Y-Achse nach oben verschoben, dass sich alle eingezeichneten Punkte unter ihr befinden. Das bedeutet, dass alle Abweichungen von der Toleranzgeraden abgedeckt sind. Die Funktion der Toleranzgeraden entspricht somit der maximal erlaubten Abweichung abhängig vom Farbwert.

Somit gilt: $MaximalErlaubteAbweichung = 0,08 * Farbwert + 40$

Mit der Formel kann die Software bereits eine eingespeicherte Steckperlenfarbe erkennen. Folgende zuvor gemachte Beobachtung könnte jedoch noch die Fehlerkennungen eingrenzen. Die Rot-, Grün- und Blauverhältnisse weisen bei gleichen Farben sehr hohe Ähnlichkeit auf. Die jeweiligen Verhältnisse können von der Software leicht berechnet und zusätzlich verglichen werden. Eine gewisse Toleranz ist jedoch auch hier notwendig. Die zuvor berechneten Abweichungen der RGB-Verhältnisse der grauen Steckperle werden als Maßstab verwendet. Eine Toleranz von 2,5% wird damit gewählt.

Die Überprüfung der Abweichung sowie des RGB-Verhältnisses wird in die BeadSort-Software implementiert. Daraufhin werden jeweils fünf Steckperlen der Farben hellblau, rot, grün, schwarz, weiß und grau nacheinander in den Analysierer gegeben. Alle

5. Test und Optimierung

Steckperlen werden korrekt erkannt. Die Farberkennung funktioniert somit zuverlässig.

Es wird ein Test zur Prüfung der Kompatibilität zu den anderen Modulen durchgeführt. Dazu wird das Modul des Einfüllbehälters, des Sequenzialisierers, des Ausrichters und des Analysierers zusammengesteckt. Dann werden 1000 gemischtfarbige Steckperlen in den Einfüllbehälter eingegeben. Die Sequenzialisierung, der Ausrichter und die Vereinzelung bei Existenzerkennung werden gestartet. Keine Steckperle wird jedoch von der Existenzerkennung erkannt. Beim Auseinanderstecken der Module ergibt sich, dass die Steckperlen an der Schnittstelle des Ausrichters und des Analysierers steckenbleiben. Die Steckperlen bleiben auf dem Rand der Eingangsröhre des Analysierers stehen, anstatt hineinzurutschen. Beim Abnehmen des Ausrichter-Moduls fallen die Steckperlen entweder in den Analysierer oder aus der Maschine heraus. Auch bei wiederholten Versuchen des Tests kommt es immer sofort zu diesem Fehler.

Bei näherer Betrachtung des Analysierers ist zu sehen, dass der Ausgangsbogen etwas geneigt ist. Die Eingangsröhre könnte daher versetzt sein, da sie auf dem Ausgangsbogen befestigt ist. Es wird eine Erweiterung entworfen, die die Eingangsröhre in der korrekten Position fixiert. Diese Erweiterung fixiert vorsichtshalber auch die Servohalterung, da diese womöglich auch leicht geneigt ist. Die Abbildung 18 zeigt den Analysierer mit Fokus auf die Erweiterung. Die Erweiterung wird als Fixierung bezeichnet und in hellbau in der Abbildung dargestellt.

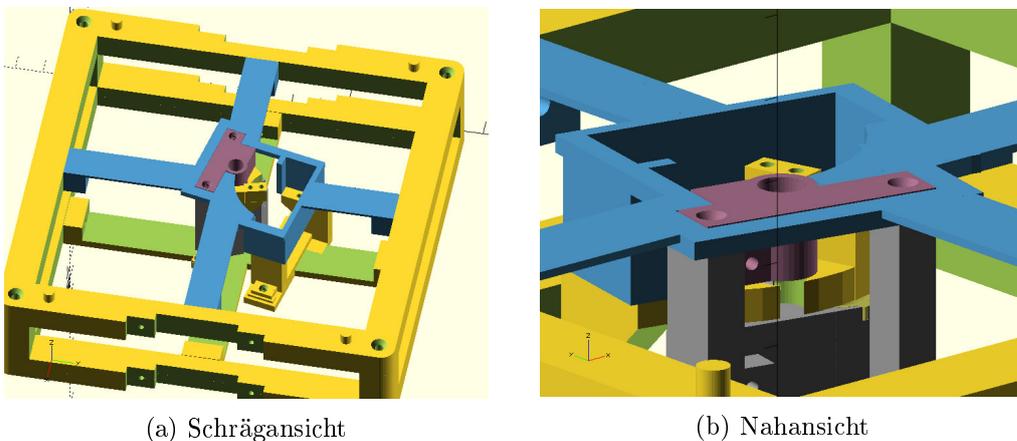


Abbildung 18: Entwurf der Fixierung des Analysierers

Nach Einbau der Fixierung wird der Test wiederholt. Die Steckperlen bleiben nun nicht mehr an der Schnittstelle stecken. Ungefähr 100 Steckperlen konnten erfolgreich

vereinzelt werden. Danach wurde der Test aus Zeitgründen beendet.

5.6. Ausgabeverfahren

Für den Test des Ausgabeverfahrens wurde nur das Modul des Ausgebers verwendet. Der Servo wurde manuell angesteuert, sodass sich die Röhre über die Ausgangslöcher bewegt. Dabei fiel auf, dass der TowerPro MG995 Servo lediglich einen Drehwinkel von etwas über 180 Grad besitzt. Das Datenblatt wurde daraufhin überprüft. Dieses spezifiziert sogar nur einen maximalen Drehwinkel von 120 Grad. [10] Um alle Ausgangslöcher erreichen zu können, musste der Servo demnach ersetzt werden. Die Recherche nach einem Ersatzservo zeigt, dass ein Drehwinkel von 360 Grad bei Servos nicht üblich ist. Sogenannte Segelwindenservos besitzen jedoch häufig sogar einen Drehwinkel von mehreren Umdrehungen und werden für gewöhnlich im Schiffsmo-
dellbau verwendet. Der Segelwindenservo mit der Bezeichnung GWS S125 1T 2BB wurde gewählt, da dieser einer der wenigen noch erhältlichen Segelwindenservos zu sein scheint, der einen Drehwinkel von genau 360 Grad besitzt. Das *1T* in der Modellbezeichnung des Servos steht für *1 Turn* (zu Deutsch: eine Umdrehung) und entspricht damit 360 Grad. [12]

Außerdem fiel auf, dass die Röhre etwas versetzt über einige der Ausgangslöcher stand. Zugleich fiel während der Bewegung der Röhre auf, dass das Eingangsloch der Röhre nicht zentriert war. Eine Justierung der Röhre mittels der Befestigungsschrauben war notwendig. Jedoch konnte die Röhre auch nach mehrmaliger Nachjustierung nicht korrekt justiert werden. Es wurde sich für eine Überarbeitung des Moduls entschieden, um die nötige Feinjustierung zu vermeiden. In Abbildung 19 ist die neue Version zu sehen. Es wurde eine Fixierung des Eingangsloches der Röhre hinzugefügt, die mithilfe eines Kugellagers das Eingangsloch zentral hält. Außerdem wurde die Servohalterung an den jetzt verwendeten Segelwindenservo angepasst. Die Ausgangslöcher sind nun größer, um weniger anfällig für geringe Abweichungen des Ausgangsloches der Röhre zu sein.

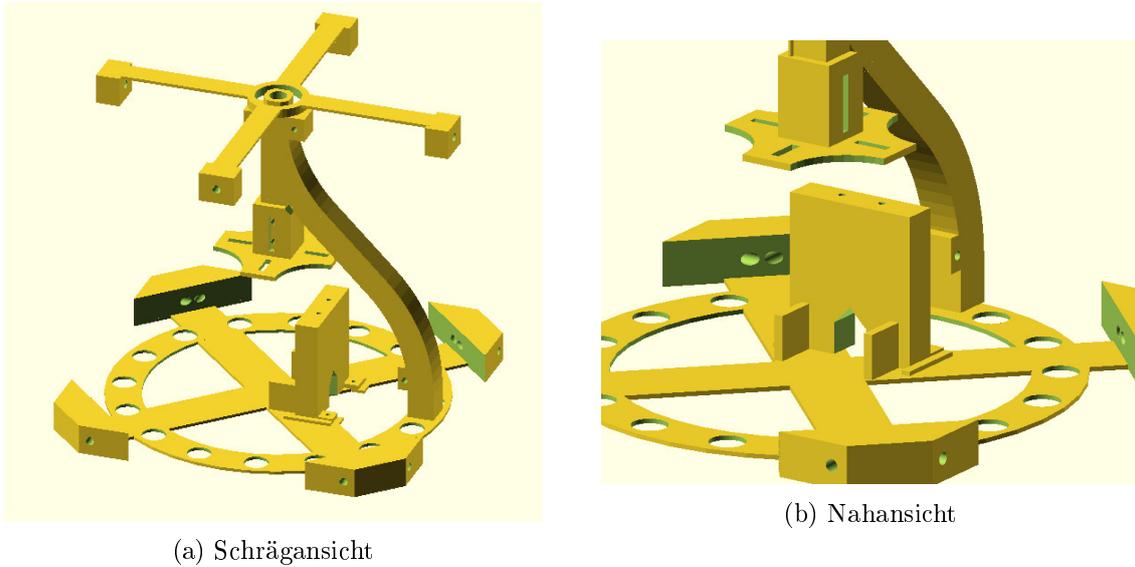


Abbildung 19: Entwurf des Ausgabeverfahrens - Version 2

Der neue Servo kann die Röhre nun über alle Ausgangslöcher bewegen. Das Ausgangsloch der Röhre ist nun auch nicht mehr versetzt über den Ausgangslöchern des Moduls. Das Eingangsloch der Röhre ist nun ebenfalls in jeder Servoposition optimal zentriert.

Die Röhre wird nacheinander über die Ausgangslöcher bewegt. Bei Erreichen eines Ausgangsloches wird der Ansteuerungswert notiert. Mit den Ansteuerungswerten der jeweiligen Ausgangslöcher soll die BeadSort-Software die Röhre zu einem beliebigen Ausgang bewegen können. Zum Test werden die Ausgänge jeweils mehrmals angefahren. Dabei fällt auf, dass sich die Röhre nicht zuverlässig an einen Ausgang positionieren lässt. Die Röhre fährt die Ausgänge nur korrekt an, wenn sich der Servo in dieselbe Richtung dreht, in der er sich drehte, als die Ansteuerungswerte notiert wurden. Muss sich der Servo in die entgegengesetzte Richtung drehen, dann fährt der Servo etwas zu weit über das Ausgangsloch. Dann würde eine eingegebene Steckperle nicht durch das Ausgangsloch fallen und am Ende der Röhre stehen bleiben. Es scheint, als würde das Momentum der Drehbewegung die Röhre etwas zu weit bewegen. Es wird vermutet, dass der Servo diese Ungenauigkeit nicht korrigiert, da sich die angesteuerte Position möglicherweise im sogenannten *Deadband* befindet.

Da die Position immer sehr ähnlich überschritten wird, lässt sich folgende Lösungsstrategie anwenden. Bei der Ansteuerung einer Position wird nicht die gewünschte Position an den Servo übermittelt, sondern eine Position kurz davor. Der Servo würde dadurch seinen Motor frühzeitiger anhalten. Das Momentum soll dann die restliche

Strecke zur gewünschten Position erbringen. Die Strategie wird in die Software implementiert und die Differenz zur eigentlichen Position schrittweise erhöht, bis aus beiden Drehrichtungen dieselbe Position angefahren wird. Die so ermittelte Differenz beträgt 4. Mit dieser Differenz steuert der Servo aus beiden Drehrichtungen nahezu dieselbe Position an. Ein höherer Wert führt zu einer Unterschreitung und ein niedrigerer Wert zu einer Überschreitung der Position.

Nun können die Ausgangslöcher aus beiden Drehrichtungen ziemlich genau angefahren werden. Ein Test soll zeigen, ob die Genauigkeit ausreichend ist. Dazu wird jedes Ausgangsloch aus beiden Drehrichtungen angefahren. An jedem Ausgangsloch werden jeweils drei Steckperlen eingegeben. Während des Tests blieben zwei Steckperlen am Ausgangsloch stecken und mussten per Hand befreit werden. Das Steckenbleiben erinnert an ein ähnliches Problem beim Analysierer. Die Genauigkeit des Servos im Analysierermodule war ebenfalls unzureichend. Die Wiggle-Lösung, die das Problem im Analysierer beseitigte, wird daher auf den Servo im Ausgabeverfahren übertragen. Der Test wird wiederholt. Alle Steckperlen konnten nun erfolgreich ausgegeben werden. Damit funktioniert das Ausgabeverfahren nun zuverlässig.

6. Software

Es wurde eine Software entwickelt, die auf dem Raspberry Pi beim Start der BeadSort-Maschine ausgeführt werden soll. Diese Software steuert den Ablauf des Sortiervorgangs. Dazu nutzt sie die Hardwareschnittstellen des Raspberry Pi, um die Aktoren der Maschine anzusteuern und die Sensoren auszulesen. Die Software beinhaltet zudem eine Benutzeroberfläche, die in der Sektion 7 näher erläutert wird. Im Folgenden wird die Architektur der Software beschrieben.

6.1. Konzept

Grundsätzlich besteht die Software aus folgenden vier Teilen:

- Hardwareschicht
- Programm
- Kontrollschicht
- Benutzerschicht

Die folgende Abbildung soll die Beziehung der vier Teile zueinander in abstrakter Weise skizzieren.

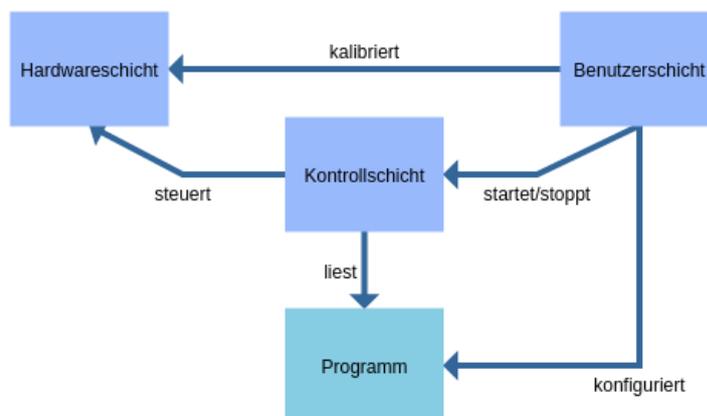


Abbildung 20: Abstrakte Übersicht der Software

Im Weiteren wird jedes der gelisteten Softwareteile nacheinander erläutert.

6.2. Hardwareschicht

Die Hardwareschicht stellt den anderen Schichten die Hardware in abstrakter Form zur Verfügung. Sie verwendet dazu die *pigpio*-Bibliothek. [2] Diese Bibliothek ermöglicht den einfachen Zugriff auf die GPIO-Pins des Raspberry Pi. Sie stellt Methoden zur Verfügung, die unter anderem das Erstellen eines PWM-Signals sowie die Kommunikation mit I2C vereinfachen. So kann die Hardwareschicht die gesamte Hardware ansteuern. Die Servos, die LEDs und der Motortreiber werden mit PWM-Signalen angesteuert. Mit dem Farbsensormodul wird über I2C kommuniziert. So müssen dem Modul spezielle Befehle gesendet werden, um diesen zum Beispiel zu starten oder zu konfigurieren. Die speziellen Befehle ließen sich aus dem Datenblatt des Farbsensormoduls [7] und einem Python-Projekt [3] als Anwendungsbeispiel ablesen.

6.3. Programm

Das Programm besteht aus jeweils einer Konfiguration für jeden verfügbaren Ausgang. Diese Konfiguration beschreibt, welche Steckperlen aus dem jeweiligen Ausgang ausgegeben werden sollen. Die möglichen Konfiguration lassen sich in folgende vier Typen unterteilen:

Ausgeschlossen

Ausgänge die so konfiguriert sind werden bei der Auswahl eines passenden Ausganges ausgeschlossen. Das bedeutet, dass die BeadSort-Maschine aus diesem Ausgang keine Steckperlen ausgibt.

Farbe

Ein Ausgang kann mit einer Steckperlenfarbe konfiguriert sein. Die BeadSort-Maschine gibt dann alle Steckperlen dieser Farbe an diesem Ausgang aus, sofern kein Puzzleset diese Steckperlenfarbe noch benötigt. Wenn mehrere Ausgänge auf dieselbe Farbe konfiguriert worden sind, so wird der Ausgang gewählt, der auch in der Benutzeroberfläche zuvor kommt. Dies gilt auch, wenn eine Steckperle durch die Toleranz der Farbabweichung mehreren Ausgängen zugewiesen werden könnte.

Puzzleset

Die BeadSort-Maschine gibt an einem so konfigurierten Ausgang die Steckperlen aus, die notwendig sind, um das Puzzleset zu vervollständigen. Ein solcher Ausgang hat dabei eine höhere Priorität, als ein Ausgang, der mit einer Farbe konfiguriert wurde.

Ein Puzzleset wird anhand von drei Parametern bestimmt. Der erste Parameter ist das Eingabebild. Dieses kann in der Benutzeroberfläche ausgewählt werden. Der zweite Parameter ist eine Auswahl von Steckperlenfarben. Das Eingabebild wird auf diese Farben reduziert beziehungsweise angepasst, sodass das Eingabebild aus den vom Benutzer gegebenen Steckperlen gelegt werden kann. Der dritte Parameter ist die gewünschte Breite des aus dem Puzzleset resultierenden Steckperlenbildes. Die Höhe wird so aus der Breite bestimmt, dass das Seitenverhältnis des Steckperlenbildes dem des Eingabebildes möglichst nahe ist. Die Bildverarbeitungen werden mithilfe der *OpenCV*-Bibliothek [19] realisiert.

Unzugeordnet

Mit solch einer Konfiguration werden aus einem Ausgang Steckperlen ausgegeben, die keinem anderen Ausgang zugewiesen werden können. Solche Steckperlen werden im Folgenden als *unzugeordnete Steckperlen* bezeichnet. Es kann von Vorteil sein, einen solchen Ausgang festzulegen. Bei Erkennung einer Steckperlenfarbe, die keinem Ausgang zugewiesen werden kann, wird die BeadSort-Maschine angehalten. Dies kann mit dieser Konfiguration verhindert werden. Wenn mehrere Ausgänge diese Konfiguration besitzen, dann werden die unzugeordneten Steckperlen gleichmäßig auf diese Ausgänge aufgeteilt.

6.4. Kontrollschicht

Die Kontrollschicht steuert den Ablauf des Sortierprozesses. Sie kann von der Benutzerschicht aus gestartet oder gestoppt werden. Mithilfe der Hardwareschicht steuert sie die Hardware so, dass die Steckperlen korrekt sequenzialisiert, ausgerichtet, vereinzelt, analysiert und ausgegeben werden. Die Kontrollschicht nutzt das konfigurierte Programm, um die Steckperlen den korrekten Ausgang zuzuweisen.

6.5. Benutzerschicht

Die Benutzerschicht beinhaltet die Benutzeroberfläche, die auf dem Touchscreen des Raspberry Pi angezeigt wird. Diese wird in der Sektion 7 näher erläutert. Neben der Oberfläche beinhaltet diese Schicht zudem die Oberflächenlogik. Mit dieser kann die Oberfläche mit den anderen Schichten interagieren. So kann sie die Hardwareschicht kalibrieren, die Kontrollschicht starten und stoppen und das Programm konfigurieren.

7. Bedienung

Die Bedienung der BeadSort-Maschine erfolgt über den Raspberry Pi beziehungsweise den daran angeschlossenen Touch-Bildschirm. Sobald die Stromzufuhr der BeadSort-Maschine hergestellt wurde, fährt der Raspberry Pi hoch und führt die BeadSort-Software aus. In der Bedienoberfläche enthaltene Bezeichnungen und Texte sind in englischer Sprache verfasst.

7.1. Startseite

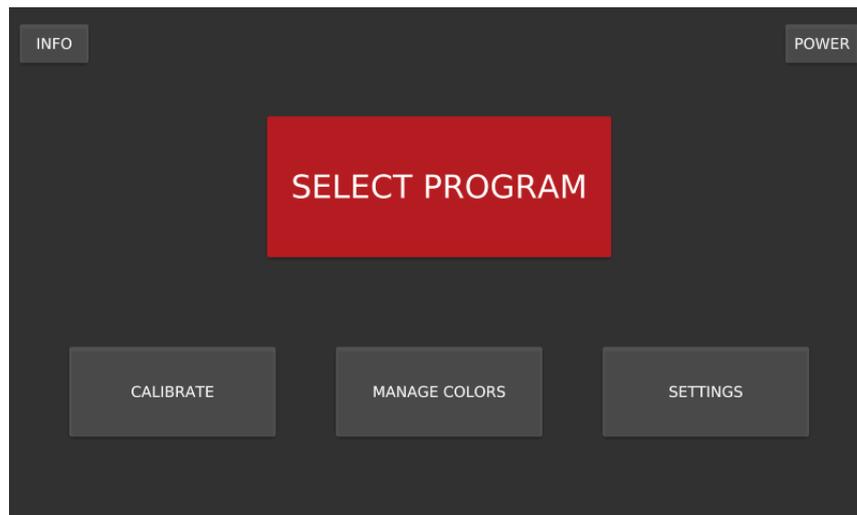


Abbildung 21: BeadSort-Software - Startseite

Beim Start der BeadSort-Software erscheint die Startseite (siehe Abbildung 21). Der *Power*-Knopf öffnet einen Dialog, in dem gewählt werden kann, ob der Raspberry Pi heruntergefahren oder die BeadSort-Software lediglich geschlossen werden soll (siehe Abbildung 22).

7. Bedienung

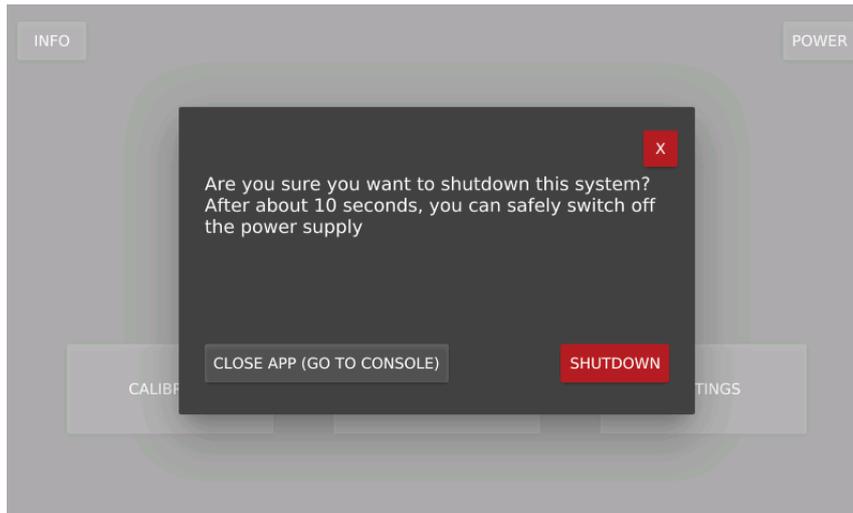


Abbildung 22: BeadSort-Software - Powerdialog

Das Schließen der BeadSort-Software führt dazu, dass die Konsole des Raspbian-systems erscheint. An dieser Stelle muss eine Tastatur an den Raspberry Pi angeschlossen werden, um eine weitere Bedienung zu ermöglichen. Zusätzlich ist eine SSH-Verbindung zum Raspberry Pi bei aufgebauter Netzwerkverbindung möglich. Der Zugang zur Konsole ist für Wartungsarbeiten vorgesehen und daher für gewöhnlich nicht notwendig.

Der *Shutdown*-Knopf fährt den Raspberry Pi herunter. Der Raspberry Pi besitzt jedoch keine Technologie, um seine Stromzufuhr zu unterbrechen. Daher muss die Stromzufuhr nach dem Herunterfahren von Hand unterbrochen werden. Ein abgeschlossenes Herunterfahren kann anhand einer auf dem Raspberry Pi befindlichen LED erkannt werden. Da diese LED durch das umschließende Gehäuse nicht gut ersichtlich ist, muss eine Zeit von zehn Sekunden gewartet werden, bis die Stromzufuhr unterbrochen wird.

7.2. Kalibrierung

Das Drücken des *Calibrate*-Knopfes auf der Startseite öffnet die Kalibrierungsseite (siehe Abbildung 23 und 24).

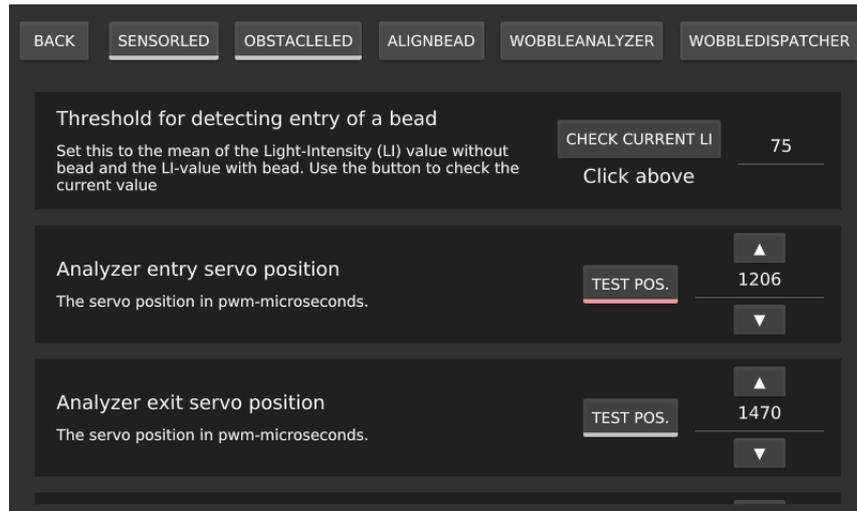


Abbildung 23: BeadSort-Software - Kalibrierung Teil 1

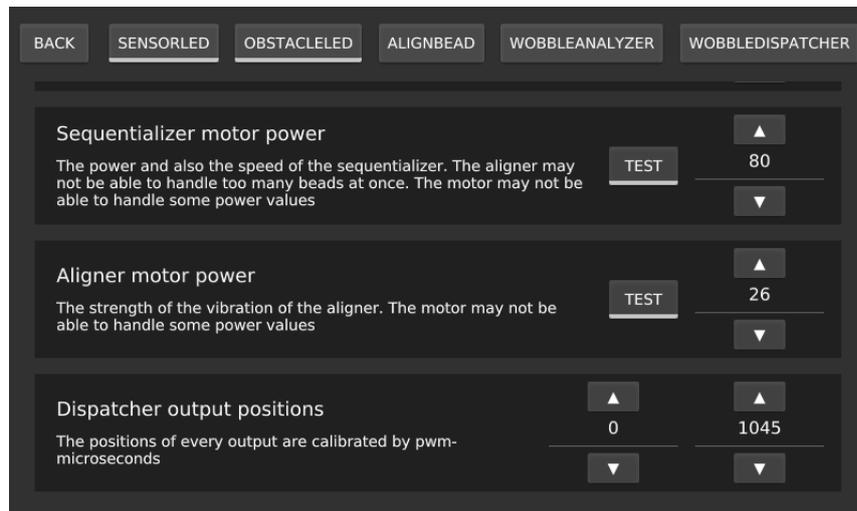


Abbildung 24: BeadSort-Software - Kalibrierung Teil 2

Auf dieser Seite befindet sich am oberen Bildschirmrand eine Reihe von Knöpfen. Der *SensorLED*- und der *ObstacleLED*-Knopf schalten jeweils den Zustand der SensorLED und der Existenz-LED. Der *AlignBead*-Knopf führt die Positionskorrektur aus. Der *WiggleAnalyzer*- und der *WiggleDispatcher*-Knopf führt jeweils den Wiggle des Analysierers und des Ausgabeverfahrens aus.

Die dunkel hinterlegten Felder beinhalten Eingabemöglichkeiten zur Kalibrierung. So kann der Schwellenwert der Existenzerkennung, die Spannung der Motoren (als

7. Bedienung

PWM-Wert von 0-255) und die Positionswerte für die Servos kalibriert werden.

7.3. Einstellungen

Das Drücken des *Settings*-Knopfes auf der Startseite öffnet die Einstellungsseite (siehe Abbildung 25).

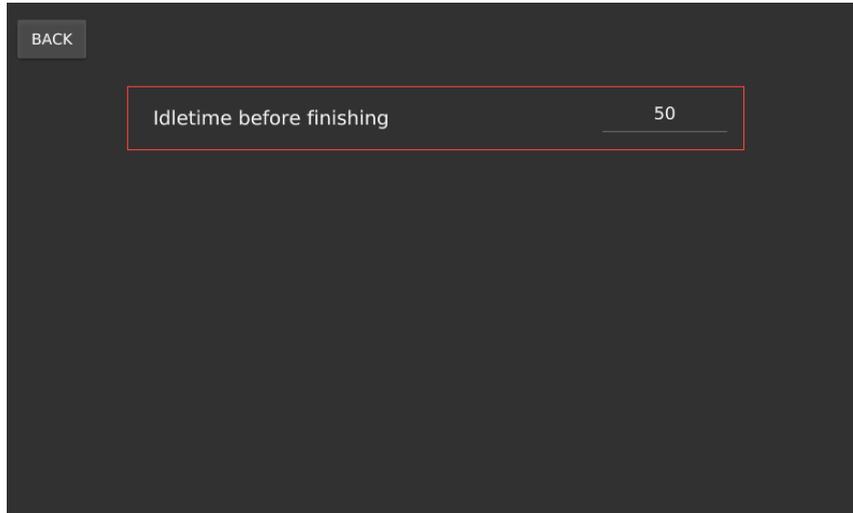


Abbildung 25: BeadSort-Software - Einstellungen

Die Einstellungsseite beinhaltet lediglich eine änderbare Einstellung. Die BeadSort-Maschine kann nicht erkennen, ob sich noch Steckperlen im Eingangsbehälter befinden. Daher wird eine maximale Wartezeit festgelegt. In dieser Zeit muss die Existenzerkennung eine Steckperle erkennen, andernfalls wird der Sortiervorgang beendet. Nach jeder Ausgabe einer Steckperle beginnt die Wartezeit von vorne. Die Wartezeit wird in Sekunden angegeben.

7.4. Farbmanagement

Das Drücken des *Manage Colors*-Knopfes auf der Startseite öffnet das Farbmanagement (siehe Abbildung 26). Auf dieser Seite können erkennbare Steckperlenfarben hinzugefügt, geändert und gelöscht werden. Die Steckperlenfarben werden in *Sets* eingeordnet. Diese Sets können ebenfalls hinzugefügt, geändert und gelöscht werden. In Abbildung 26 sind die Sets Hama, Test und First zu sehen. Das Hama-Set ist ausgewählt, daher werden in der mittleren Spalte die im Hama-Set enthaltenen Steckperlenfarben angezeigt. Die Steckperlenfarbe mit der Bezeichnung 028 ist ausgewählt.

Auf der rechten Seite befinden sich die Eigenschaften der gewählten Steckperlenfarbe. In den Textfeldern befindet sich die Farbbezeichnung und der Farbcode. Darunter wird die Steckperlenfarbe in einem Farbfeld visuell dargestellt. Wenn auf das Farbfeld gedrückt wird, dann öffnet sich ein Farbauswahldialog. Mit diesem Dialog kann eine beliebige Farbe gewählt werden. Die letzte Eigenschaft ist der Farbwert, den der Farbsensor für eine Steckperle dieser Steckperlenfarbe zurückgibt. Der darunter befindliche *Start Colorsensing*-Knopf startet einen Vorgang zum Einlesen einer Steckperlenfarbe. Wenn der Vorgang gestartet wurde, dann liest der Analysierer den Farbwert der als nächstes eintreffenden Steckperle ein. Der gelesene Farbwert überschreibt dann den Farbwert der aktuell ausgewählten Steckperlenfarbe im Farbmanagement.

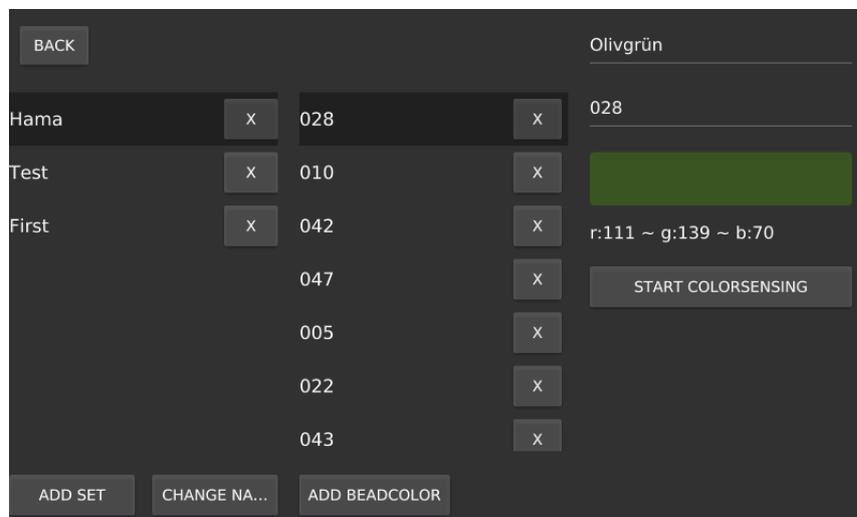


Abbildung 26: BeadSort-Software - Farbmanagement

7.5. Programm

Das Drücken des *Select Program*-Knopfes auf der Startseite öffnet die Programmseite (siehe Abbildung 27). Auf dieser Seite kann das Programm konfiguriert werden, das während des Sortiervorgangs berücksichtigt wird. Das Programm beinhaltet für jeden Ausgang jeweils eine Konfiguration.

7. Bedienung

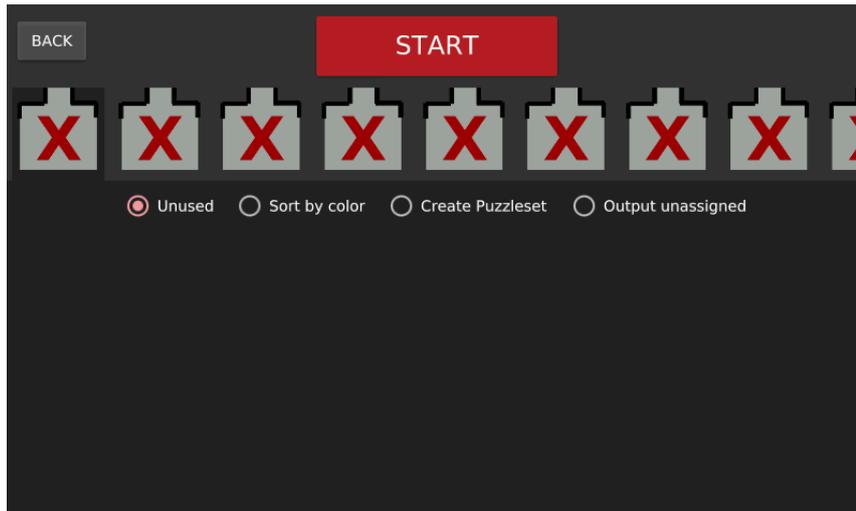


Abbildung 27: BeadSort-Software - Programm

Die Konfiguration lässt sich mit den Auswahlknöpfen unterhalb der Ausgangsauswahl wählen. Die Ausgänge sind standardmäßig als ausgeschlossen konfiguriert.

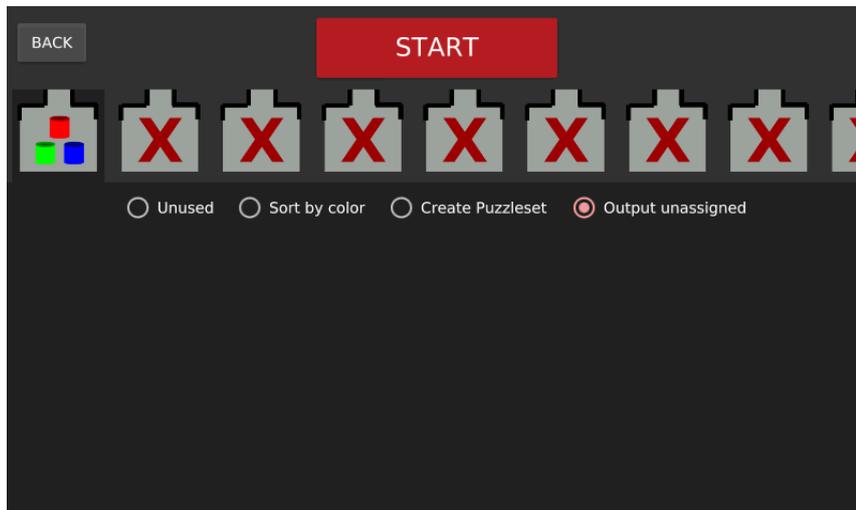


Abbildung 28: BeadSort-Software - Programm - Unzugeordnete Steckperlen

Ausgänge können wie in Abbildung 28 so konfiguriert werden, dass an dem jeweiligen Ausgang unzugeordnete Steckperlen ausgegeben werden.

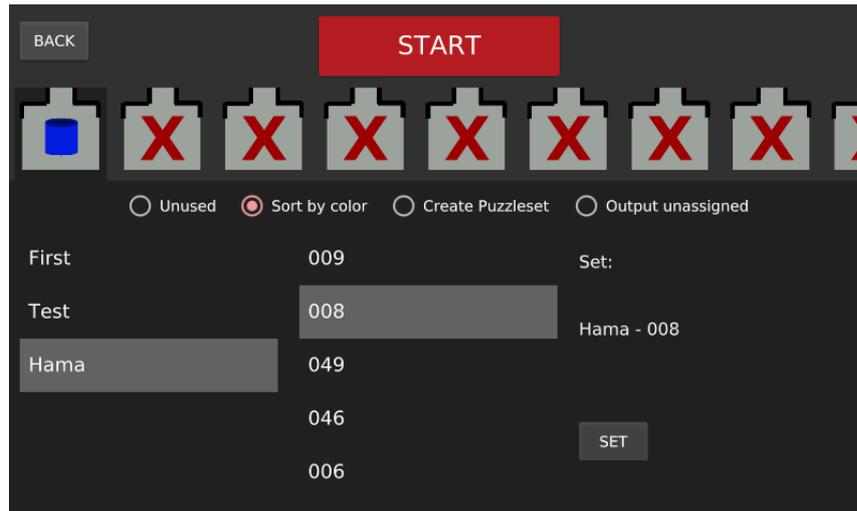


Abbildung 29: BeadSort-Software - Programm - Farbe

Wenn ein Ausgang nach Farbe konfiguriert wird, dann erscheint eine Auswahlmöglichkeit (siehe Abbildung 29). Diese Auswahlmöglichkeit beinhaltet jede in dem Farbmanagement eingespeicherte Steckperlenfarbe.

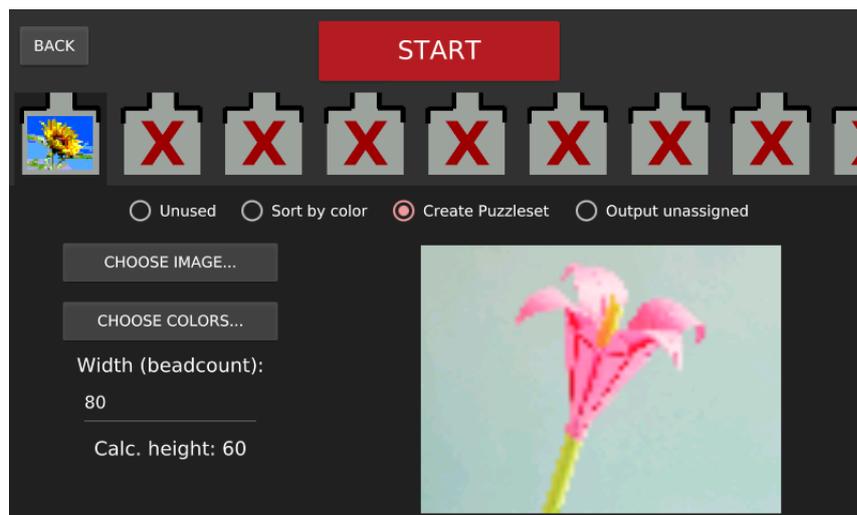


Abbildung 30: BeadSort-Software - Programm - Puzzleset

Wenn ein Ausgang als Puzzleset-Ausgang konfiguriert wird, dann erscheint eine Oberfläche zur Erstellung des Puzzlesets (siehe Abbildung 29). Dann muss mithilfe des *Choose Image...*-Knopfes ein Bild gewählt werden, welches als Grundlage für das Puzzleset dient. Dann werden mithilfe des *Choose Colors...*-Knopfes die gewünsch-

7. Bedienung

ten Steckperlenfarben gewählt, die das Puzzleset beinhalten soll. Eine Vorschau des Steckperlenbildes wird mit den gewählten Farben generiert. Solange die Vorschau mit dem Finger festgehalten wird ist das gewählte Bild in den Originalfarben zu sehen. Die gewünschte Breite des entstehenden Steckperlenbildes kann über ein Eingabefeld angegeben werden.

7.6. Ausführung

Nach Fertigstellung des Programms kann mit dem *Start*-Knopf auf der Programmseite der Sortiervorgang gestartet werden. Dann wird die Ausführungsseite (siehe Abbildung 31) angezeigt.



Abbildung 31: BeadSort-Software - Ausführung

In der Mitte der Seite wird der aktuelle Status des Sortiervorganges ausgegeben. Mit dem *Pause*-Knopf kann der Vorgang pausiert werden. Nur im pausierten Zustand kann der Vorgang mit dem *Abort*-Knopf abgebrochen werden. Nach Abbruch des Vorgangs wird die Programmseite erneut angezeigt.

Die jeweils gewählte Konfiguration für die Ausgänge wird am unteren Bildschirmrand angezeigt. Unter den Ausgängen wird jeweils die Anzahl der bereits an dem jeweiligen Ausgang ausgegebenen Steckperlen angezeigt.

8. Evaluation und Fazit



Abbildung 32: Zusammengesetzte BeadSort-Maschine

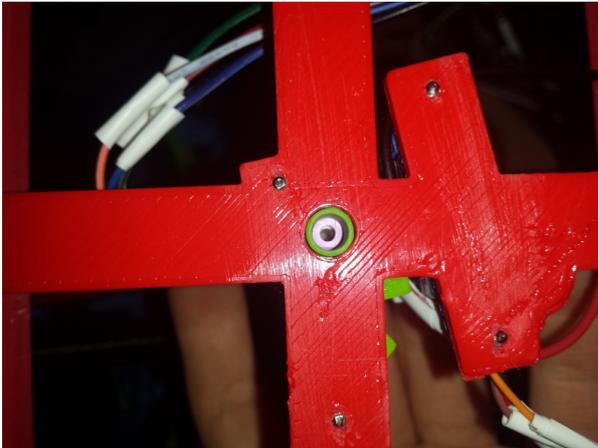
An dieser Stelle ist die BeadSort-Maschine einsatzbereit. In Abbildung 32 ist die zusammengesetzte BeadSort-Maschine zu sehen. In dieser Sektion wird die Maschine getestet und das Testergebnis analysiert. Mit der Analyse soll die Geschwindigkeit und Fehlerhäufigkeit ermittelt werden. Dabei ist grundsätzlich zwischen fatalen und nicht fatalen Fehlern zu unterscheiden. Ein Fehler ist fatal, wenn die Maschine den Sortierprozess nicht fortführen kann. Andernfalls ist ein Fehler nicht fatal. Ein fataler Fehler ist zum Beispiel das Steckenbleiben einer Steckperle in der Maschine. Wenn eine Steckperle in der Maschine stecken bleibt, dann stauen sich nachfolgende Steckperlen an dieser steckengebliebenen Steckperle auf und können nicht mehr sortiert werden. Ein nicht fataler Fehler ist zum Beispiel das Ausgeben einer Steckperle in einen falschen Ausgang.

8.1. Tests des Gesamtsystems

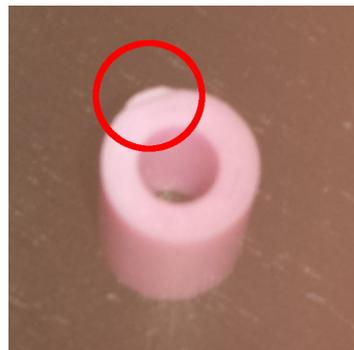
Pastellfarbentest

Es werden insgesamt drei Tests durchgeführt. Im ersten Test soll die BeadSort-Maschine Steckperlen nach sechs verschiedenen Farben sortieren. Dazu wird eine Pastellfarben-Steckperlenmischung von Hama verwendet. Diese beinhaltet 13000 Steckperlen in sechs verschiedenen Farben. Die Steckperlenfarben werden in das Farbmanagement der BeadSort-Software eingetragen. Die Steckperlenmischung wird in den Eingangsbehälter gegeben. Sechs Ausgänge werden so konfiguriert, dass jede Steckperlenfarbe jeweils einem Ausgang zugewiesen ist. Ein Ausgang wird für unzugeordnete Steckperlen konfiguriert. Jede in diesen Ausgang ausgegebene Steckperle muss fehlerhaft sortiert worden sein, da die Mischung nur die sechs zugewiesenen Farben enthält. Der Sortierprozess wird gestartet.

Nach 3 Stunden und 56 Minuten wird der Test beendet. Ein fataler Fehler ist aufgetreten. Eine Steckperle ist in der Ausgangsröhre des Analysierers stecken geblieben (siehe Abbildung 33a). Bei näherer Betrachtung der Steckperle fällt auf, dass die Steckperle beschädigt ist. In Abbildung 33b ist die Beschädigung rot eingekreist zu sehen. Diese Beschädigung ist wahrscheinlich für das Steckenbleiben verantwortlich. Steckperlen könnten in verschiedenster Weise beschädigt sein. Eine Vermeidung des Steckenbleibens beschädigter Steckperlen scheint daher nur schwer möglich. Es wird vorausgesetzt, dass die eingegebenen Steckperlen unbeschädigt sein müssen, um eine Sortierung zu ermöglichen.



(a) Steckenbleiben in der Ausgangsröhre



(b) Beschädigung der Steckperle

Abbildung 33: Im Analysierer steckengebliebene Steckperle

Es wurde deutlich, dass eine Sortierung der gesamten 13000 Steckperlen zu zeitaufwändig ist. Daher wurde der Test nach dem Entfernen der steckengebliebenen Steckperle nicht fortgesetzt. Das bis dahin erlangte Ergebnis wurde als ausreichend angesehen. Im Folgenden wird dieses Ergebnis dargestellt.



Abbildung 34: Ergebnis der Sortierung von Pastellfarben

In Abbildung 34 sind die sortierten Steckperlen in den jeweiligen Ausgangsgläsern zu sehen. Während des Sortierprozesses sind einige Steckperlen aus der Maschine gefallen. Die meisten davon liegen neben den Ausgangsgläsern auf der Standplatte. Einige sind in die unbenutzten Ausgangsgläser gefallen. Insgesamt konnten 30 Steckperlen gefunden werden, die aus der Maschine gefallen sind. Im Folgenden wird angenommen, dass es sich dabei um alle herausgefallenen Steckperlen handelt. Aufgrund der kleinen Größe ist es jedoch möglich, dass nicht alle herausgefallenen Steckperlen gefunden wurden. Es konnte nur schwer erkannt werden, an welcher Stelle die Steckperlen aus der Maschine fallen, da das Herausfallen relativ selten geschieht. Zusätzlich sind die Steckperlen aufgrund ihrer Größe schwer zu erkennen während sie fallen. Es wird vermutet, dass die Steckperlen beim Aufprall auf die Ausgangsrutschen gelegentlich so zurückprallen, dass sie die Rutsche verlassen. Daher werden die Rutschen mit einer Art Deckel versehen, die den Anfang der Rutschen so bedeckt, dass keine Steckperlen herausfallen können.

Laut der BeadSort-Software wurde die folgende Anzahl Steckperlen in die jeweiligen

8. Evaluation und Fazit

Ausgänge sortiert:

Ausgang	Anzahl Steckperlen	Falsch sortierte Steckperlen
Rot	105	1 (Gelb)
Pink	242	1 (Grün)
Blau	186	0
Gelb	299	0
Grün	208	0
Lila	329	12 (Pink), 1 (Gelb), 1 (Blau)
Unzugeordn. Steckp.	327	327
Gesamt	1696	343

Tabelle 4: Ergebnisübersicht des Sortierens nach Pastellfarben

In Tabelle 4 ist neben der Anzahl der im Ausgang befindlichen Steckperlen auch die Anzahl der jeweils falsch sortierten Steckperlen angegeben. Zusätzlich ist die Gesamtzahl der sortierten Steckperlen sowie die Gesamtzahl der falsch sortierten Steckperlen angegeben. Die korrekt sortierten Steckperlen ergeben sich nun, wenn die falsch sortierten Steckperlen (343) und die aus der Maschine gefallenen Steckperlen (30) von der Gesamtzahl sortierter Steckperlen (1696) abgezogen wird. Die Anzahl der korrekt sortierten Steckperlen beträgt demnach 1323. Das bedeutet eine Fehlerrate von 22% für nicht fatale Fehler.

Die nicht fatalen Fehler lassen sich in *False Positives*, *False Negatives* und anderweitigen Fehler unterscheiden. False Positives sind hierbei Steckperlen, die fälschlich in einen Ausgang gefallen sind, der für nicht zugeordnete Steckperlen konfiguriert wurde. So gab es in diesem Test 327 False Positives, da jede Steckperle in dem Ausgang für nicht zugeordnete Steckperlen fälschlich darin liegt. False Negatives sind hingegen Steckperlen, die fälschlich in einen Ausgang gefallen sind, der mit einer Steckperlenfarbe oder einem Puzzleset konfiguriert wurde. So gab es in diesem Test 16 False Negatives. Anderweitige Fehler sind hierbei die Steckperlen, die sich außerhalb der Maschine befinden. Demnach gibt es 30 anderweitige Fehler.

Die False Positives entstehen wahrscheinlich durch eine Abweichung des vom Sensor gelesenen Farbwertes. Dafür können kleinste Positionsabweichungen verantwortlich sein. Diese wurden bereits in Sektion 5.5.2 durch die Positionskorrektur minimiert. Eine mögliche Lösung wäre die Erhöhung der Toleranzwerte bei der Erkennung von Farben. Die Erhöhung könnte jedoch mit sich führen, dass die Anzahl an False Negatives steigt, da die BeadSort-Software ähnliche Farben eher als dieselbe Farbe ansehen könnte. Später in dieser Sektion wird ein Test mit teilweise ähnlicheren Farben durchgeführt. Falls die Anzahl an False Negatives gering bleibt, wird daraufhin eine

Erhöhung der Toleranzwerte in Betracht gezogen.

Es wird vermutet, dass einige der False Negatives dieses Tests mit dem Herausfallen der Steckperlen aus der Maschine zusammenhängen. Eine gelbe Steckperle ist beispielsweise in den Ausgangsbehälter der roten Steckperlen gefallen. Der farbliche Unterschied zwischen gelb und rot ist sehr hoch. Es ist unwahrscheinlich, dass der Farbsensor einen derartig groben Fehler verursacht. Es ist wahrscheinlicher, dass die herausgefallenen Steckperlen zufällig in die Ausgangsgläser zugewiesener Farben gefallen sind. Eine Ausnahme zu dieser Vermutung könnten die zwölf pinken Steckperlen sein, die als False Negatives in den Ausgang der lilanen Steckperlen landeten. Die farbliche Differenz der lilanen Steckperlen zu den pinken Steckperlen ist gering genug, um einen Fehler durch den Farbsensor zu rechtfertigen. Die Erweiterung an den Ausgangsrutschen könnte das Herausfallen behoben haben und damit die Ursache der False Negatives einschränken. Der nächste Test wird daher durchgeführt, bevor eine Lösung zu den False Negatives gesucht wird.

Die BeadSort-Maschine benötigte für den Test 3 Stunden und 56 Minuten. Im Durchschnitt konnte sie somit ungefähr 7,19 Steckperlen pro Minute sortieren. Korrekt sortiert wurden im Durchschnitt ungefähr 5,61 Steckperlen pro Minute. Die Maschine würde hochgerechnet ungefähr 38 Stunden und 39 Minuten benötigen, um eine Steckperlenmischung mit 13000 Steckperlen vollständig und korrekt zu sortieren. Dies scheint ein recht hoher Wert, jedoch gibt es keine zeitlichen Anforderungen, die die BeadSort-Maschine erfüllen muss.

Vielfarbentest

In diesem Test sollen alle Ausgänge der BeadSort-Maschine verwendet werden. Es wird eine Steckperlenmischung von Hama mit der Nummer 67 verwendet. Diese enthält 13.000 Steckperlen in 22 verschiedenen Farben. 15 der in der Mischung enthaltenen Farben werden in die BeadSort-Maschine eingetragen. Die Ausgänge werden jeweils mit einer Farbe konfiguriert. Die Steckperlen werden in den Einfüllbehälter gegeben. Dann wird der Sortierprozess gestartet. Nach sechs Stunden wird der Test beendet. Während des Tests sind insgesamt drei fatale Fehler aufgetreten. Während eines fatalen Fehlers wurde die gemessene Zeit angehalten. Die fatalen Fehler waren wie folgt:

1. Eine Steckperle blieb in der Ausgangsröhre des Ausrichters stecken. An der Steckperle konnte keine Beschädigung festgestellt werden. Wiederholte erneute Einwürfe der Steckperle in den Ausrichter konnten den Fehler nicht reproduzie-

8. Evaluation und Fazit

ren. Somit ließ sich die Ursache nicht ermitteln. Der Test wurde fortgesetzt.

2. Eine Steckperle blieb in der Ausgangsröhre des Analysierers stecken. Die steckengebliebene Steckperle war beschädigt. In Abbildung 35 ist diese Steckperle zu sehen. Die Beschädigung ist rot eingekreist. Dieser Fehler ähnelt dem fatalen Fehler im Pastellfarbentest. Es wird erneut angenommen, dass die Beschädigung für das Steckenbleiben verantwortlich war. Der Test wird fortgesetzt.
3. Der Vibrationsmotor ging während des Tests kaputt. Eine Ursache konnte nicht ermittelt werden. Unter Umständen ist dies auf eine mindere Qualität des Motors zurückzuführen. Der Motor wird vorerst lediglich ersetzt. Dann wurde der Test fortgesetzt. Bei einem erneuten Ausfall sollte ein Motor höherer Qualität eingesetzt werden.



Abbildung 35: Beschädigte Steckperle im Vielfarbenentest

Keine Steckperle ist während des Tests aus der Maschine gefallen. Die Erweiterung der Ausgangsrutschen hat dieses Problem wahrscheinlich behoben. Außerdem ist in keinem Ausgang, der nach einer Farbe konfiguriert wurde, eine stark unterschiedlich farbige Steckperle vorhanden. Die Vermutung, dass die herausgefallenen Steckperlen im Pastellfarbentest in die Ausgangsgläser gefallen sind, wird damit bestätigt. Jedoch sind viele Steckperlen ähnlicher Farbe in demselben Ausgang ausgegeben worden. Das liegt wahrscheinlich an der Toleranz der Farberkennung.

Ausgang	Anzahl Steckperlen
Türkis	191
Violettrot	147
Beige	431
Grau	217
Olivgrün	154
Grün	75
Rot	7
Gelb	22
Hellrot	115
Hellgrün	127
Lila	76
Braun	161
Rotbraun	191
ic Hellbraun	120
Hautfarbe	2
Unzugeordn. Steckp.	554
Gesamt	2590

Tabelle 5: Ergebnisübersicht des Sortierens nach Pastellfarben

In der Tabelle 5 sind die Anzahlen der sortierten Steckperlen laut BeadSort-Software zu sehen. Es fallen vor allem zwei Reihen auf. Es wurden laut der BeadSort-Maschine lediglich 7 rote und 2 hautfarbene Steckperlen sortiert. Bei Betrachtung der Ausgangsgläser ist jedoch zu sehen, dass sich viele rote Steckperlen in den Ausgangsgläsern der violettroten und der rotbraunen Steckperlen befinden. Die Ausgangsgläser der beige und der hellroten Steckperlen beinhalten viele hautfarbene Steckperlen. Die Toleranz der Farberkennung verursacht diese Fehler wahrscheinlich. So sieht die BeadSort-Maschine die roten Steckperlen mal als violettrot und mal als rotbraun an. Die hautfarbenen Steckperlen werden mal als beige und mal als hellrot angesehen. Dies verursacht die False Negatives. Eine Verringerung der Toleranz würde dieses Problem unter Umständen lösen. Jedoch würde die Anzahl der False Positives dadurch steigen.

Die verwendete Steckperlenmischung enthält mehr Farben als Ausgänge verwendet wurden beziehungsweise verfügbar waren. Dadurch kann die Anzahl unzugeordneter Steckperlen (laut BeadSort-Maschine) nicht als False Positives angesehen werden. Die Prüfung aller unzugeordneten Steckperlen von Hand bedeutet einen hohen Aufwand und wird daher nicht durchgeführt. Jedoch ergibt eine Betrachtung des Ausgangsglases der unzugeordneten Steckperlen, dass zumindestens einige False Positives vorhanden sind. Von einer Verringerung der Toleranz wird daher abgesehen. Dieser Test

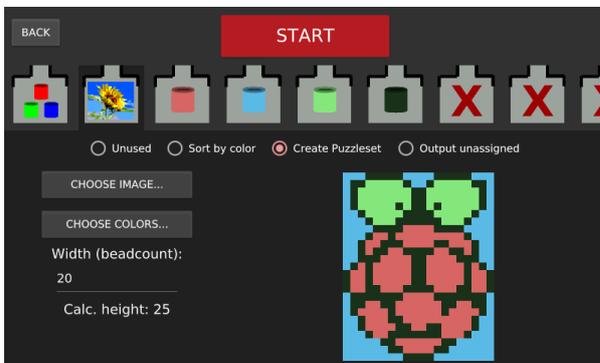
8. Evaluation und Fazit

ermöglicht keine leichte Ermittlung der Fehlerraten von False Negatives und False Positives. Daher wird von einer Ermittlung in diesem Test abgesehen.

Der Test dauerte sechs Stunden und es wurden 2590 Steckperlen sortiert. Die BeadSort-Maschine hat somit pro Minute 7,19 Steckperlen sortiert. Dieser Wert trifft exakt mit der Geschwindigkeit aus dem Pastellfarbentest überein. Die Anzahl korrekt sortierter Steckperlen pro Minute kann aufgrund der unbekanntem Fehlergröße nicht ermittelt werden.

Puzzlesettest

In diesem Test wird ein Puzzleset erstellt. Das Puzzleset soll aus dem Raspberry Pi Logo erstellt werden. In Abbildung 36a ist eine Vorschau des an der BeadSort-Maschine konfigurierten Puzzlesets zu sehen. Das Puzzleset soll aus den vorher sortierten Farben Pastellrot, Pastellgrün, Pastellblau und Olivgrün erstellt werden. Der erste Ausgang wird für unzugeordnete Steckperlen konfiguriert. Der zweite Ausgang soll das Puzzleset ausgeben. Der dritte bis sechste Ausgang wird jeweils den im Puzzleset verwendeten Farben zugeordnet. Nur diese vier Farben werden in den Einfüllbehälter eingegeben. Es werden jeweils ungefähr 200 Steckperlen eingegeben, sodass das Puzzleset fertiggestellt werden kann. Dann wird der Test gestartet. Kurz nachdem das Puzzleset vollständig ist, wird der Test beendet. Die Maschine gab nach der Fertigstellung des Puzzlesets keine Steckperlen mehr an dem entsprechenden Ausgang aus.



(a) Vorschau des erstellten Puzzlesets



(b) Gelegtes Bild aus dem Puzzleset

Abbildung 36: Puzzleset-Erstellung

In Abbildung 36b ist das aus dem Puzzleset gelegte Bild zu sehen. Das Puzzleset ist entgegen der Anzeige der BeadSort-Maschine nicht vollständig. Fünf Steckperlen

fehlen. Es scheint, als wären keine Steckperlen aus der Maschine gefallen. Es liegen weder Steckperlen auf dem Stand noch sind Steckperlen offensichtlich falsch einsortiert worden. Daher kann nicht ermittelt werden, warum fünf Steckperlen fehlen.

Es gab in diesem Test keine False Positives. Die gewählten Farben waren wahrscheinlich ausreichend unterschiedlich. Es gab jedoch 140 False Negatives. Zusätzlich sind fünf Steckperlen des Puzzlesets anscheinend falsch einsortiert worden. Das entspricht einer Fehlerrate von 19,52% für nicht fatale Fehler. Dieses Ergebnis ist zufriedenstellend. Ein fataler Fehler ist jedoch wiederholt aufgetreten. Der Getriebemotor des Sequenzialisierers blieb trotz geringen Füllstandes des Einfüllbehälters mehrmals stehen. Einige Male startete er von selbst wieder. Viermal musste jedoch nachgeholfen werden. Dazu wurde der Rotoraufsatz lediglich etwas von Hand gedreht. Der Getriebemotor muss wahrscheinlich ausgetauscht werden, um zukünftig zuverlässige Ergebnisse zu erzielen.

Der Test dauerte eine Stunde und 51 Minuten an. Es wurden insgesamt 743 Steckperlen sortiert. Die BeadSort-Maschine sortierte somit 6,69 Steckperlen pro Minute. Dieser Wert ist etwas geringer als die Werte der vorherigen Tests. Der Grund ist wahrscheinlich der Getriebemotor, der gelegentlich pausiert und den Sortierprozess dadurch verlangsamt. Es wurden 598 Steckperlen korrekt sortiert. Das entspricht einer Geschwindigkeit von 5,39 Steckperlen pro Minute.

An dieser Stelle wird eine Fehlerrate der fatalen Fehler ermittelt. Dabei wird das Anhalten des Getriebemotors als lediglich ein fataler Fehler angesehen. Der Motor hätte zu dem Zeitpunkt ausgetauscht werden müssen. Es war jedoch kein Ersatz vorhanden und eine Neuanschaffung war zu zeitaufwändig. So gab es insgesamt fünf fatale Fehler. Die Maschine sortierte insgesamt 5029 Steckperlen. Das entspricht einem fatalen Fehler je 1005,8 sortierter Steckperlen. Ein unbeaufsichtigtes Sortieren ist damit nur eingeschränkt möglich. Es wird empfohlen, die Maschine in Hörreichweite zu platzieren. Die Geräusche der BeadSort-Maschine deuten fatale Fehler oft an. Das Geräusch eines steckenbleibenden Servos ist beispielsweise leicht erkennbar und deutet auf eine steckengebliebene Steckperle hin.

8.2. **Fazit**

Die BeadSort-Maschine kann Steckperlen nach Farbe sortieren und nimmt dem Benutzer mühsame Arbeit ab. Die Maschine sortiert unterschiedliche Steckperlenfarben sehr gut. Bei ähnlichen Farben kommt es schnell vor, dass sich Steckperlen in denselben Ausgang ausgegeben werden, obwohl sie jeweils einem anderen Ausgang zugeordnet sind. Möglicherweise lässt sich die Genauigkeit der BeadSort-Maschine mit weiteren Methoden verbessern. Die Erstellung von Puzzlesets funktioniert jedoch gut und macht die BeadSort-Maschine vermutlich einzigartig.

Die Fehlerrate von fatalen Fehler schränkt die BeadSort-Maschine etwas ein. Unbeaufsichtigtes Betreiben der Maschine sollte vermieden werden. Die Hardware könnte im schlimmsten Fall beschädigt werden, wenn ein fataler Fehler nicht umgehend beseitigt wird. Der Einbau von besserer Hardware (speziell: Vibrationsmotor und Getriebemotor) würde jedoch die Fehlerrate fataler Fehler mindern. Ein anderer bekannter Auslöser fataler Fehler sind die beschädigten Steckperlen. Möglicherweise sind hochwertigere Steckperlen erhältlich, die keine Beschädigungen besitzen. Es kann jedoch sein, dass eine Anpassung der Steckperlenröhren der Maschine bei Wechsel der Steckperlenmarke notwendig wird. Dies ist jedoch aufgrund der parametrisierten 3D-Modelle ohne großen Aufwand möglich. Die BeadSort-Maschine ist aufgrund ihrer Modularität wartungsfreundlich und gut erweiterbar. So könnte eine Erweiterung entwickelt werden, mit der die BeadSort-Maschine die sortierten Steckperlen an die BeadBox weiterleitet. Die BeadSort-Maschine wurde nur bei normalem Zimmerlicht getestet. Es wird vermutet, dass Anpassungen notwendig sind um Störungen durch größeren Lichteinfluss wie Sonnenlicht zu kompensieren.

Die BeadSort-Software erfüllt die Anforderungen dieser Arbeit. Die Bedienung mittels Touchscreen funktioniert gut, ist aber etwas träge. Die Trägheit lässt sich womöglich auf die relativ geringe Leistung des Raspberry Pi zurückführen. Aufgrund von wenig Erfahrung mit vorallem dem Qt-Framework kam es häufiger zu Problemen bei der Softwareentwicklung. So traten sporadisch scheinende Abstürze auf, die größtenteils behoben werden konnten. Die Softwareentwicklung mit Qt auf einem Raspberry Pi wird jedoch gut unterstützt und die Einrichtung verlief mit nur wenigen Problemen. Zusammenfassend wird die Entwicklung der BeadSort-Maschine als Erfolg angesehen. Sie entspricht den gestellten Anforderungen und ist alltagstauglich. Die Unterstützung transparenter Steckperlen könnte sich dazu unter Umständen leicht nachrüsten lassen. Die Transparenz einer Steckperle könnte über den von der Existenz-LED durchscheinenden Licht ermittelt werden.

A. Einrichtung

Für die Entwicklung der Software wurde das *Qt*-Framework [21] verwendet. Dieses Framework bietet eine umfangreiche Sammlung an Bibliotheken, die die Anwendungsentwicklung erleichtern sollen. Das *Qt*-Framework ist sehr verbreitet und bekannt für seine gute Unterstützung bei der Entwicklung grafischer Oberflächen. Als Entwicklungsumgebung wurde der von *Qt* entwickelte *Qt Creator* verwendet. Der *Qt Creator* besitzt neben höchster Kompatibilität mit dem *Qt*-Framework eine gute Unterstützung für Kreuzkompilierung. Kreuzkompilierung ist die Kompilierung von Software für eine Architektur von einer anderen Architektur aus. Damit lässt sich Software zum Beispiel auf einem Desktop-Computer, der für gewöhnlich eine *x86-Architektur* besitzt, entwickeln und kreuzkompilieren. Der *Raspberry Pi*, der eine *ARM-Architektur* besitzt, kann die krosskompilierte Software dann ausführen. Der Kompilierungsvorgang würde auf dem *Raspberry Pi* aufgrund der relativ geringen Rechenleistung für gewöhnlich erheblich mehr Zeit in Anspruch nehmen. Daher wird die Kreuzkompilierung bevorzugt. Im Folgenden werden die notwendigen Vorbereitungen der Kreuzkompilierung erläutert.

Es wurde eine Kreuzkompilierungsumgebung aufgesetzt, bei der *Qt 5.9.1* eingesetzt wird. Das Tutorial von Amir Mansouri zur Kreuzkompilierung mit *Qt* und einem *Raspberry Pi 3* wurde als Hilfestellung verwendet. Als Betriebssystem des *Raspberry Pi* wurde *Raspbian* gewählt. Zum Zeitpunkt des Aufsetzens der Umgebung war *Raspbian Jessie* die aktuellste Version. Aus diesem Grund wurde diese Version gewählt. Mittlerweile existiert eine neue Version namens *Raspbian Stretch*. Jedoch wird *Raspbian Jessie* weiterhin genutzt, da keine notwendigen Vorteile bekannt sind und das Upgrade auf *Raspbian Stretch* Zeit benötigt, die in anderen Bereichen dieser Arbeit besser investiert ist.

Die Prozedur zum Aufsetzen der Kreuzkompilierungsumgebung lässt sich in folgende Schritte kurzfassen.

- Installieren von *Ubuntu*, dem *Qt Creator* und des *Qt* Quellcodes
- Herunterladen der Kreuzkompilierungswerkzeuge und des *Raspbian*-Abbildes
- Vorbereiten des *Raspbian*-Abbildes
- Konfigurieren, Kreuzkompilieren und Installieren von *Qt* in das *Raspbian*-Abbild
- Aufspielen des modifizierten *Raspbian*-Abbildes auf den *Raspberry Pi*
- Einrichten der Kreuzkompilierung im *Qt Creator*

A. Einrichtung

Damit ist die Kreuzkompilierungsumgebung erfolgreich aufgesetzt. Die BeadSort-Software benötigt jedoch zwei weitere Bibliotheken abgesehen von den Qt-Bibliotheken des Qt-Frameworks. Die erste Bibliothek ist die pigpio-Bibliothek. Diese ist jedoch im Raspbian vorinstalliert und kann ohne weiteres im Qt Creator-Projekt der BeadSort-Software eingebunden werden. Die zweite benötigte Bibliothek ist die OpenCV-Bibliothek. Da diese nicht vorinstalliert ist, muss sie für den Raspberry Pi kreuzkompiliert werden. Dazu wird der Quellcode der Bibliothek in der Version 3.2 heruntergeladen und mit folgendem CMake-Befehl kreuzkompiliert:

```
1 cmake -DCMAKE_BUILD_TYPE=Release -DBUILD_ZLIB=ON -DCMAKE_CXX_COMPILER
   =/home/linus/crosscompile-tools/tools/arm-bcm2708/gcc-linaro-arm-
   linux-gnueabihf-raspbian-x64/bin/arm-linux-gnueabihf-g++ -
   DCMAKE_AR=/home/linus/crosscompile-tools/tools/arm-bcm2708/gcc-
   linaro-arm-linux-gnueabihf-raspbian-x64/bin/arm-linux-gnueabihf-
   ar -DCMAKE_C_COMPILER=/home/linus/crosscompile-tools/tools/arm-
   bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/bin/arm-linux-
   gnueabihf-gcc -DWITH_TBB=ON -DBUILD_TBB=ON -DENABLE_NEON=ON -
   DCMAKE_TOOLCHAIN_FILE=../platforms/linux/arm-gnueabi.toolchain.
   cmake ..
```

Listing 1: Befehl zur Kompilierung von OpenCV

Nach der Kompilierung wird die Bibliothek in das Raspbian-Abbild installiert. Daraufhin kann die OpenCV-Bibliothek im Qt Creator-Projekt der BeadSort-Software eingebunden werden. Schließlich ist die Umgebung bereit, die BeadSort-Software zu kompilieren und auf dem Raspberry Pi auszuführen.

Literatur

- [1] 3Faktur. *Genauigkeiten bzw. Toleranzen im 3D-Druck*. Zugriff: 06.10.2017. Sep. 2016. URL: <https://3faktor.com/genauigkeiten-bzw-toleranzen-im-3d-druck/>.
- [2] abyz.co.uk. *The pigpio library*. Zugriff: 06.10.2017. URL: <http://abyz.co.uk/rpi/pigpio/>.
- [3] Adafruit. *Adafruit Python TCS34725*. Zugriff: 06.10.2017. URL: https://github.com/adafruit/Adafruit_Python_TCS34725.
- [4] AliExpress. *L298N Motor Driver Board Module*. Zugriff: 06.10.2017. URL: <https://www.aliexpress.com/item/Special-promotions-1pcs-lot-L298N-motor-driver-board-module-L298-for-arduino-stepper-motor-smart-car/1872427142.html>.
- [5] AliExpress. *Micro N20 DC Gear Motor 6v 60RPM*. Zugriff: 06.10.2017. URL: <https://de.aliexpress.com/item/12MM-6V-60RPM-Powerful-High-Torque-DC-Gear-Box-Motor/32277597895.html>.
- [6] Amazon. *Hama 211-00 - Bügelperlen Dose mit circa 13.000 Perlen, 10 Farben gemischt*. Zugriff: 06.10.2017. URL: <https://www.amazon.de/dp/B00BP5Y1PE/>.
- [7] ams. *TCS3472 Datenblatt*. Zugriff: 06.10.2017. URL: <http://ams.com/eng/content/download/319364/1117183/287875>.
- [8] Geir Andersen. *Perler beads sorter*. Zugriff: 06.10.2017. März 2010. URL: https://www.youtube.com/watch?v=YRt-_E1jE8k.
- [9] Rolf R Bakke. *My Nonstop / Skittles sorting machine*. Zugriff: 06.10.2017. Nov. 2014. URL: <https://www.youtube.com/watch?v=tSEHDBSynVo>.
- [10] Electronicos Caldas. *TowerPro MG995 Datenblatt*. Zugriff: 06.10.2017. URL: http://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf.
- [11] Vivian Auduong Carolyn Mei Kerlina Liu. *Perler Bead Sorter Touchpoint*. Zugriff: 06.10.2017. Apr. 2017. URL: <https://www.youtube.com/watch?v=65022ECh40>.
- [12] GWS. *Sail Winch S125 Series*. Zugriff: 06.10.2017. URL: <http://www.gwsus.com/english/product/servo/s125.htm>.
- [13] Marius Hacker. "Erweiterung eines automatischen Systems zur Erstellung von Steckperlenbildern um Mechanismen zur Fehlertoleranz und Qualitätskontrolle". Masterarbeit. C.v.O. Universität Oldenburg, 2016.

- [14] arduino5 (in Arduino Forum). *The Great Color Sensor Thread*. Zugriff: 06.10.2017. URL: <https://forum.arduino.cc/index.php?topic=291882.0>.
- [15] Instructables. *Everything you need to know about colour sensors*. Zugriff: 06.10.2017. URL: <http://www.instructables.com/id/Everything-you-need-to-know-about-colour-sensors/>.
- [16] Sammy Linden. *Bead-funnel*. Zugriff: 06.10.2017. Feb. 2017. URL: <http://www.thingiverse.com/thing:2081971>.
- [17] Sammy Linden. *Hama / perler bead sorting machine*. Zugriff: 06.10.2017. Feb. 2017. URL: <https://www.youtube.com/watch?v=YHqOfRUBrX0>.
- [18] MarVtec. *Arduino Projekt Sortiermaschine*. Zugriff: 06.10.2017. Sep. 2016. URL: <https://www.youtube.com/watch?v=UNjU1Hwrt2U>.
- [19] OpenCV. *OpenCV Startseite*. Zugriff: 06.10.2017. URL: <http://opencv.org/>.
- [20] Microcontrollers Projects. *Sorting Machine - Skittles and M&M's 8 Color Sorting*. Zugriff: 06.10.2017. Aug. 2015. URL: <https://www.youtube.com/watch?v=Q6NqGH1LBY>.
- [21] Qt. *Qt Startseite*. Zugriff: 06.10.2017. URL: <https://www.qt.io/>.
- [22] Egen Riether. *Skittles Sorting Machine 3*. Zugriff: 06.10.2017. Apr. 2013. URL: <https://www.youtube.com/watch?v=CZ08XF634w4>.
- [23] Scanlime. *3D Printed Vibrating Perler Bead Hopper*. Zugriff: 06.10.2017. Feb. 2011. URL: <https://www.flickr.com/photos/micahdowty/5407791891/in/set-72157625540465089/>.
- [24] Scanlime. *Perler Bead Robot*. Zugriff: 06.10.2017. März 2011. URL: <http://www.thingiverse.com/thing:7332>.
- [25] Scanlime. *Perler Bead Robot Ideas*. Zugriff: 06.10.2017. Feb. 2011. URL: <http://scanlime.org/2011/02/perler-bead-robot-ideas/>.
- [26] Gabriel Schneider. *Steckperlen-Sortierer*. Zugriff: 06.10.2017. März 2017. URL: <https://www.youtube.com/watch?v=sg5qM0B347g>.
- [27] Bernhard Schwerdtner. *Sortiermaschine für Bügelperlen*. Zugriff: 06.10.2017. Jan. 2017. URL: <https://www.youtube.com/watch?v=Ew2EFa5V5R8>.
- [28] Henning Ziegler. "BeadBox: Entwicklung eines informatischen Systems zur automatisierten Erstellung von Bügelperlenbildern". Bachelorarbeit. C.v.O. Universität Oldenburg, 2013.

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

Oldenburg, den 16. Oktober 2017

Linus Barth