

Farbsortiermaschine

Dokumentation

Autoren: *Niklas Kamm, HNBK*
Julian Krieger, HNBK
Pascal Gläß, HNBK

Betreuer: *Herr Hennig, HNBK*

Zeitraum: *11.09.2019 bis 01.03.2020*



Heinz-Nixdorf-Berufskolleg
für Elektrotechnik, Informations- und Telekommunikationstechnik
der Stadt Essen

Inhaltsverzeichnis

1. Einleitung	1
2. Projektbeschreibung	1
2.1. Organisatorische Vorgaben	1
2.2. Problemanalyse Projekt - Ziel	1
2.3. Beteiligte	1
2.4. Verlinkungen auf Websites	1
3. Projektplanung.....	2
3.1. Verwendete Systeme und Programme.....	2
3.1.1. Dokumentation und Planung.....	2
3.1.2. Video- & Bildbearbeitung 3D Modellierung	3
3.1.3. Programmierung	3
3.2. Soll - Planung	4
3.2.1. Konzeption.....	4
3.2.2. Aufbau	4
3.2.3. Software - Planung	5
3.2.4. Hardware - Planung	7
3.2.5. Schaltplan.....	8
3.3. Zeitplanung	10
4. Realisierung	11
4.1. Aufbau Hardware	11
4.2. Programmierung.....	11
4.2.1. Testcodes	11
4.2.2. Niklas Kamm.....	11
4.2.3. Julian Krieger	12
4.2.4. Pascal Gläß	12
5. Projektergebnisse	12
6. Projektbewertung	12
6.1. Julian Krieger	12
6.2. Pascal Gläß.....	13
6.3. Niklas Kamm	13
7. Anhang	14
7.1. Programmablaufplan (via PapDesigner)	14
7.2. Programmcode - Main (via Arduino)	14
7.3. Programmcode - Motor testen (via Arduino)	14
7.4. Programmcode - Sensoren testen (via Arduino)	14

Einbindungsverzeichnis

Video 1: Das Gestell erklärt (YouTube)	4
Video 2: Die Sortierscheibe erklärt (YouTube).....	5
Video 3: Die Sperrscheibe erklärt (YouTube)	5

Abbildungsverzeichnis

Abbildung 1: Programmablaufplan, grundlegend	6
Abbildung 2: Schaltplan.....	9
Abbildung 3: Gantt - Diagramm	10

1. Einleitung

Das Projekt und die Dokumentation wurden durch die Autoren eigenständig durchgeführt und dokumentiert. Betreut wurde es von Herrn Hennig und wurde im Schulunterricht am Heinz-Nixdorf Berufskolleg durchgeführt.

2. Projektbeschreibung

2.1. Organisatorische Vorgaben

Das Projekt wurde im Zeitraum vom 11.09.2019 bis 01.03.2020 durchgeführt. Je 2 Unterrichtsstunden finden in einem 2-Wochen-Rhythmus statt.

2.2. Problemanalyse | Projekt - Ziel

Das betreffende System (Bonbon - Abfüllanlage) verfügt derzeit nicht über eine Methode, neue, unsortierte Schokoladenkugeln zu sortieren. Diese müssen per Hand sortiert und in das richtige Fach gelegt werden. Das Projekt dient zur Konzipierung und Umsetzung eines Geräts, welches den Sortierprozess vereinfachen soll.

2.3. Beteiligte

Das Projekt wird gemeinsam von den drei Autoren Niklas Kamm, Julian Krieger und Pascal Gläß durchgeführt und dokumentiert. Im Rahmen des Schulunterrichts wird das Projekt von Herr Hennig im Fach „Rechner- und Systemtechnik“ betreut.

2.4. Verlinkungen auf Websites

In der Dokumentation folgen viele Verlinkungen auf folgende Websites:

- [YouTube](#) - Einbindungen von Videos
 - [GitHub](#) - Verwaltung des gesamten Projekts
 - rs.krieger-blog.de - Eine Website von Julian Krieger erstellt
- Viele zusätzliche Informationen, welche wir angeben, sind dort aufgelistet und verfügbar.

3. Projektplanung

3.1. Verwendete Systeme und Programme

Für das Projekt haben wir viele verschiedene Programme benutzt, jede Software nach Kategorie sortiert.

Wir haben hauptsächlich das Betriebssystem Windows 10 benutzt. Außerdem hat Julian mit Ubuntu 19.10 gearbeitet. Sämtliche Programme sind für Windows 10 verfügbar.

3.1.1. Dokumentation und Planung

Dokumentation

Für unsere Dokumentation haben wir [Microsoft Word](#) verwendet. Über unsere Schule verfügt jeder über eine Office 365 Lizenz.

Programmablaufplan

Wir haben unseren Programmablaufplan mit der kostenfreien Software "PapDesigner" in der Version 2.2.0.8.04 erstellt.

Entwickelt und zur Verfügung gestellt wurde die Software vom "Simon - Ohm Berufskolleg Köln".

Die Software steht über heise.de zum Download bereit.

Schaltplan

Für unsere Schaltpläne haben wir die kostenfreie Software ["Fritzing"](#) in der Version 0.9.3 verwendet.

Neben der Funktion Schaltpläne zu erstellen, bietet Fritzing außerdem die Möglichkeit eine Schaltung aus der Steckbrettansicht zu erstellen.

Dies ist ziemlich hilfreich um Schaltungen visuell darzustellen.

Gantt - Diagramm / Zeitplanung

Die ebenfalls kostenfreie Software ["ProjectLibre"](#) (Version 1.9.1) ermöglicht uns die Erstellung und Gantt - Diagrammen und sorgt daher für eine gute Zeitplanung in unserem Projekt.

Empfehlung: Wir haben viele Guides benutzt, um diese Software und all ihre Features genau zu verstehen.

Projektverwaltung

All unsere Dateien wie der Programmcode, die Ablaufpläne, die Dokumentation, etc. haben wir über ["GitHub"](#) verwaltet.

GitHub basiert auf dem Versionsverwaltungssystem "Git". Mit GitHub können wir sämtliche Dateien online speichern, sodass sie jederzeit zur Verfügung stehen.

Ein wichtiger Vorteil, den GitHub gegenüber Cloud Storages hat, ist die einfache Versionskontrolle aller Dateien, sowie das Fusionieren von mehreren Versionen.

Das bedeutet, dass mehrere Leute an einer Datei arbeiten können, ohne Verluste oder Konflikte zu befürchten.

Die Software ["GitHub Desktop"](#) sorgt für eine einfache Verwaltung auf lokaler Ebene.

3.1.2. Video- & Bildbearbeitung | 3D Modellierung

2D - Viewer

Von unserem Lehrer wurden uns die 2D - Dateien (.dxf) zur Verfügung gestellt. Um diese allerdings anzuschauen, brauchten wir eine Software, welche u.a. .dxf Dateien öffnen kann.

[Autodesk DWG TrueView 2018](#) ist eine ,für Schüler, Studenten und Lehrer kostenfreie, Software um 2D - Dateien anschauen zu können.

3D - Modellierung

Die 3D - Modellierung des Geräts wurde mit der kostenfreien Software "[Blender](#)" in der Version 2.80 erstellt.

Blender bietet umfassende Funktionen zur Erstellung von 3D - Modellen.

Videobearbeitung

Alle Videos, welche die Funktionen des Geräts beschreiben wurden mit "[Vegas Pro 16.0](#)" erstellt.

Diese professionelle Software ermöglicht das Erstellen, Bearbeiten und Rendern von sämtlichen Video- und Audioformaten.

Bildbearbeitung und Screenshots

Screenshots wurden entweder durch eine interne Funktion der jeweiligen Software gemacht, oder durch das kostenfreie Tool "[Lightshot](#)".

Bearbeitet wurden die Screenshots mit dem kostenfreien Grafikprogramm "[GIMP](#)" in der Version 2.10.12.

GIF Erstellung

Es gibt viele Online Tools, welche u.a. aus MP4 Dateien GIFs erstellen können. Leider wurde keines unseren Ansprüchen gerecht.

Daher haben wir uns für die Software "[Photoshop CC 2019](#)" entschieden. Mit ihr können wir aus MP4 Dateien ohne Beschränkungen (z.B. in der Framerate) GIFs erstellen.

3.1.3. Programmierung

Haupt - Entwicklungsumgebung (IDE)

Für die Programmierung, das Kompilieren und Übertragen zum Microcontroller haben wir die kostenfreie IDE "[Arduino IDE](#)" in der Version 1.8.10 verwendet.

Da die Software direkt von den Entwicklern der Arduino Plattform stammt, bietet sie viele Features, eingebaute Compiler und vieles mehr.

Andere

Außerdem haben wir für manche Situationen die kostenfreie IDE "[Atom](#)" benutzt. Diese verfügt, dank ihrer Entwicklung von den Machern von GitHub, über eine sehr gute Kompatibilität zu GitHub Desktop, was sehr schnelle und kurze Code - Anpassungen ermöglicht. Sie eignet sich für Coding, enthält allerdings keinen Compiler.

3.2. Soll - Planung

3.2.1. Konzeption

Das Konzept des Geräts wurde von unserem Lehrer am Anfang der Projektphase vorgegeben. Außerdem wurde die Funktionsweise bereits geklärt und ist daher nicht änderbar. Lediglich der genaue Ablauf ist anpassbar, dieser muss ohne Vorlage geplant und programmiert werden.

3.2.2. Aufbau

Das Gerät lässt sich grob in drei Teile aufteilen:

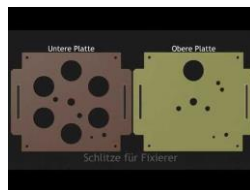
- Das Gestell, an welchem die Motoren und Sensoren montiert sind
- Die obere Scheibe, welche die Kugeln empfängt und verarbeitet
- Die untere Scheibe, welche die Kugeln zum passenden Loch abfüllt

Information:

Zu jedem der folgenden Teile gibt es Animationen, welche den Gesamtaufbau und die einzelnen Komponenten erklärt. Diese sind über YouTube als Video in Word eingebunden. Alternativ kann man die Animationen auch als MP4 oder GIF über „rs.krieger-blog.de“ abrufen.

Auf der Website findet man außerdem Informationsseiten zu den einzelnen Teilen, welche neben den Animationen (MP4 & GIF) weitere Informationen, wie z.B. die Abmaße oder Materialien enthält.

Das Gestell



Video 1: Das Gestell erklärt
(YouTube)

Das Gestell dient zur Halterung des gesamten Geräts. Es hält die beiden Scheiben, welche für das Sortieren verantwortlich sind, an der richtigen Stelle. Das Gestell besteht aus zwei Seitenwänden, welche das gesamte Konstrukt tragen. Zwei Platten, die in Schlitze in den Seitenwänden geschoben werden, sorgen für den Halt der Scheiben. Außerdem sind an ihnen auch die Farberkennung, die Lichtschranken sowie die Motoren zur Steuerung der Scheiben befestigt. Damit die Platten nicht ungewollt aus der Seitenwand fallen, fixieren wir diese, indem wir Schlitze in die Platten schneiden. Diese schauen hinter den Seitenwänden raus und können dort die Fixierer reinstecken, welche die Seitenwände und Platten zusammenhalten.

[Informationsseite auf rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

[MP4 über rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

[GIF über rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

Die Sortierscheibe



Video 2: Die Sortierscheibe erklärt
(YouTube)

Die Sortierscheibe dient zum Sortieren der Kugeln. Sie empfängt die Kugeln als erste und fährt sie dann zu einem Sensor, welcher die Farbe ermittelt. Nach der Ermittlung fährt die Scheibe zu dem passenden Auswurf, um die Kugel zu sortieren. Sie besteht grundlegend aus zwei Scheiben, welche durch Abstandhalter auseinandergehalten werden. Eine Röhre dient als Behälter für die Kugel.

[Informationsseite auf rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

[MP4 über rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

[GIF über rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

Die Sperrscheibe



Video 3: Die Sperrscheibe erklärt
(YouTube)

Die Sperrscheibe sorgt auch für die Sortierung. Sie bewegt sich zur Position der ermittelten Farbe, um einen Durchlass für die Kugel zu ermöglichen. Alternativ sorgt die obere Scheibe der Sperrscheibe dafür, dass die Kugel in der Sortierscheibe nicht durchfällt. Auch sie besteht grundlegend aus zwei Scheiben, welche durch Abstandhalter auseinandergehalten werden. Auch hier gibt es eine Röhre, welche als Durchlass für die Kugel genutzt wird. Dazu besitzt die Sperrscheibe noch eine Zentrierungsfunktion. Zwei kombinierte Bauteile sorgen dafür, dass die Sortier- und Sperrscheibe den gleichen Mittelpunkt haben. Das sorgt für perfekt übereinstimmende Löcher, durch die die Kugel fallen kann.

[Informationsseite auf rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

[MP4 über rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

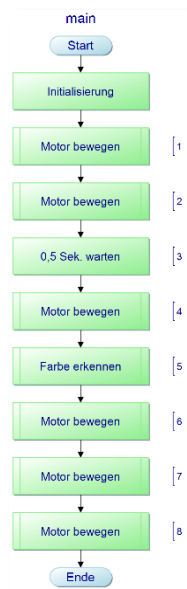
[GIF über rs.krieger-blog.de abrufen](https://rs.krieger-blog.de)

3.2.3. Software - Planung

Grundlegend

Der Programmcode ist für das Arduino Modell „Uno R3“ optimiert. Geschrieben und kompiliert wurde er mit der Arduino IDE in der Version 1.8.10.

Ablauf



Da hier nur eine grobe Übersicht über den Ablauf des Programms gegeben wird, wird nicht weiter auf die Initialisierung eingegangen. Diese finden Sie unter Punkt 4.2 ff.!

1. Der untere Motor bewegt die Sperrscheibe auf die Position „Gold“.
2. Der obere Motor bewegt die Sortierscheibe auf die Startposition.
3. Es wird 0,5 Sek. gewartet, um die Kugel in die Sortierscheibe fallen zu lassen
4. Der obere Motor bewegt die Sortierscheibe zur Farberkennung.
5. Die Farberkennung wird gestartet
6. Der untere Motor bewegt die Sperrscheibe zur erkannten Farbe.
7. Der obere Motor bewegt die Sortierscheibe zur erkannten Farbe. Dabei fällt die Kugel durch die Sperrscheibe in den richtigen Behälter.
8. Der untere Motor bewegt die Sperrscheibe zur Startposition.

Abbildung 1:
Programmablaufplan,
grundlegend

Aufteilung und Zuordnung

Nach einigen **gemeinsamen** Überlegungen haben wir bereits eine grobe Code-Struktur gehabt. Dies war für uns sehr wichtig, um die Kommunikation zwischen den Methoden besser zu planen. Ohne Wissen, welche Daten wir von den anderen Methoden empfangen oder welche wir übergeben müssen ist es schwer, funktionierenden Code zu programmieren. In dieser Struktur haben wir den Code in mehrere Abschnitte aufgeteilt:

1. Motorsteuerung, bewegt die Motoren zu verschiedenen Positionen
2. Farberkennung, erkennt die Farbe und gibt sie zurück
3. Setup + Loop, Pin - Initialisierung, Ausführung der Methoden

Nach der Aufteilung haben wir feststellen müssen, dass nicht jeder seine eigene Methode bekommen könnte. Dank unserer zuvor geplanten Struktur konnten wir aber dennoch eine, unserer Meinung nach gleichmäßige, Aufteilung und Zuordnung finden:

- Motorsteuerung (in zwei Teile aufgeteilt): Julian, Niklas
- Farberkennung: Pascal
- Setup + Loop: Gemeinsam (Kombinieren der einzelnen Abschnitte)

Sonstiges

Außerdem haben wir uns für **wichtige Regeln** entschieden, um das Zusammenspiel der einzelnen Abschnitte zu optimieren:

1. Der Code in 2 Versionen verfasst: Deutsche / Englische Kommentierung
2. Möglichst einfache, aber detaillierte Kommentierung
3. Korrektes Einrücken für strukturierten Code
4. Jede Methode bekommt einen eigenen Kommentar, welcher die Funktion beschreibt
5. Übergebene Parameter (wenn vorhanden) werden intern mit einem Unterstrich am Anfang markiert

6. So viel wie nötig, so wenig wie möglich! Das gilt für redundante Befehle sowie zusammenfassbare Variablen zu Arrays!

Weitere Details und Überlegungen zum Programmcode, sowie die Dokumentation jedes einzelnen finden Sie unter **Punkt 4.2 ff.**

3.2.4. Hardware - Planung

Da die Bauelemente, sowie das Design Teil der Vorgabe waren, gab es bei uns keine Entscheidungen in der Hardware zu treffen. Daher listen wir nur jedes Bauteil mit einer kurzen Beschreibung auf.

Nicht elektrotechnische Bauteile (Das Konstrukt)

Das gesamte Konstrukt wurde aus unbeschichteten mitteldichten Holzfaserplatten hergestellt. Das Design wurde von Herrn Hennig konzipiert und mithilfe eines Lasercutters ausgedruckt und uns zur Verfügung gestellt.

Der Arduino

Als Mikrocontroller - Board benutzen wir die Arduino Plattform. Da wir den Formfaktor, aber auch die gebrauchte Anzahl an Pins beachten müssen, benutzen wir ein Board des Modells „Arduino Uno R3“.

Die Motoren

Da wir die Scheiben getrennt steuern wollen, brauchen wir 2 Motoren vom gleichen Modell. Dafür stehen uns 2x „28BYJ-48“ Motoren (M1, M2) zur Verfügung. Zur Verstärkung der Signale benutzen wir 2x „X113647“ Treiberplatinen (T1, T2).

Die Farberkennung

Die Farberkennung besteht aus einer RGB - LED (P3) sowie einer Erweiterungsplatine mit lichtempfindlichem Widerstand (B3).

Die Sensorplatine des Typs „Light Sensor 1.0“ wurde von „Seeed Studio“ hergestellt. Der Sensor benötigt einen Anschluss an 5V sowie GND. Ein Pin sorgt für die analoge Übertragung des Sensors an den Arduino.

Die Lichtschranken

Die Lichtschranken bestehen aus einer IR - LED (P1, P2) sowie einem lichtempfindlichen Widerstand (B1, B2), welcher in der Schaltung als Spannungsteiler dient.

3.2.5. Schaltplan

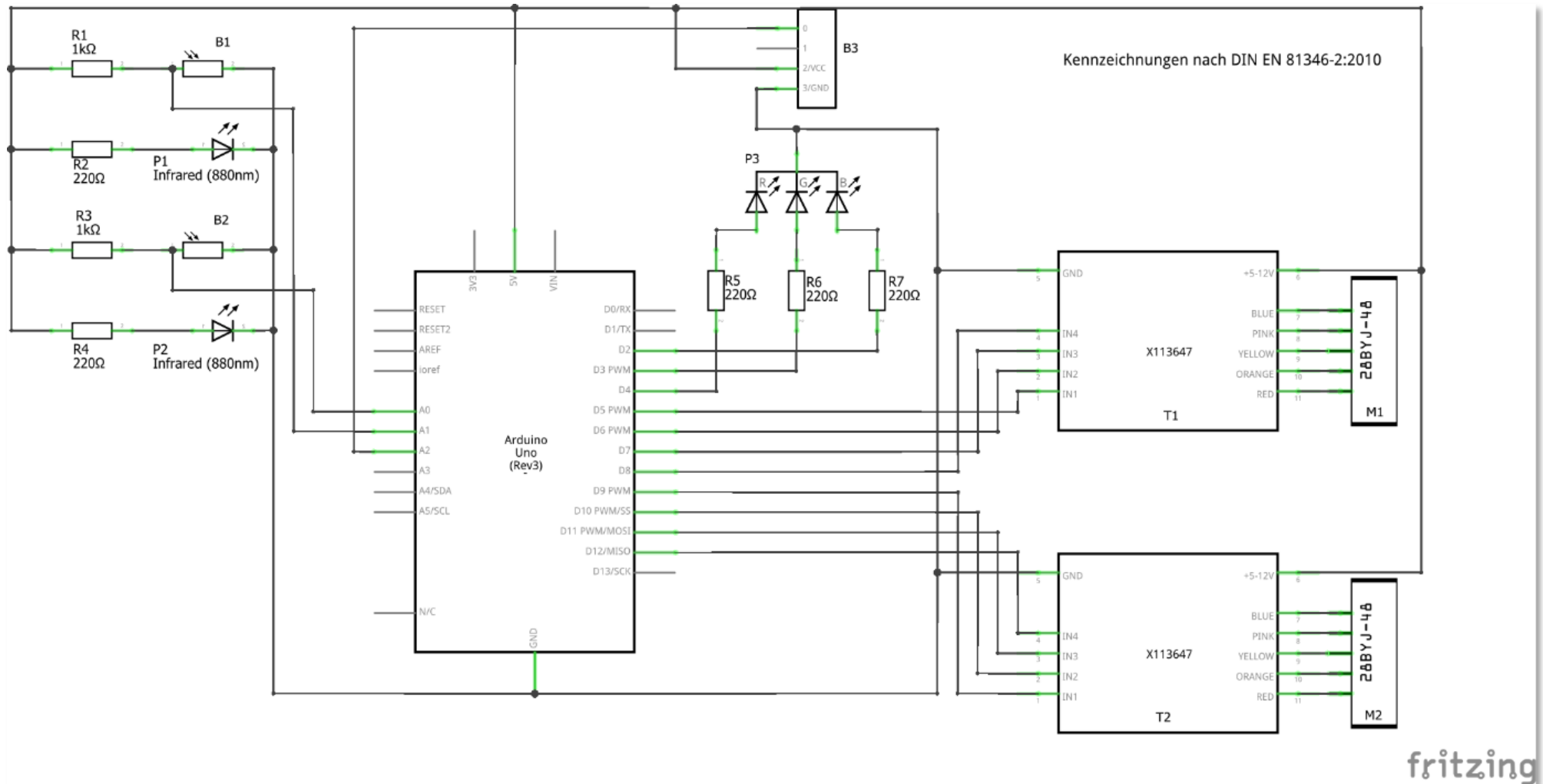
Nachdem alle Bauelemente geklärt sind, kam es nun zur Erstellung des Schaltplans. Hier war es besonders wichtig, die Sensoren an analoge Pins anzuschließen, um ein auswertbares Signal zu erhalten.

Zur Erstellung des Schaltplans haben wir das kostenfreie Programm „Fritzing“ in der Version 0.9.3 benutzt.

Dort mussten wir einige Bauteile hinzufügen, da sie nicht standardmäßig in der Bauteilliste enthalten waren. Weitere Erklärungen, sowie den Download der Dateien finden Sie [hier](#).

Außerdem werden in der Schaltplanansicht standardmäßig die amerikanischen Symbole angezeigt, welche nicht der deutschen Norm entsprechen. Daher hat Julian mit dem Programm „Inkscape“ neue Vektorgrafiken erstellt, um den Normen zu entsprechen. Diese sind [hier](#) einseh- und herunterladbar.

Abbildung 2: Schaltplan



3.3. Zeitplanung

Unsere Zeitplanung fand am Anfang des Projektzeitraums statt. Wir haben ein Gantt - Diagramm mit der kostenfreien Software „ProjectLibre“ in der Version 1.9.1 erstellt. Neben der Auflistung unserer Prozesse haben wir auch eine Einteilung der Aufgaben vorgenommen. Diese ist über die Ressourcenplanung in „ProjectLibre“ eingepflegt.

Die folgenden Screenshots zeigen das Diagramm im Soll - Ist Vergleich:

Die blauen Balken zeigen die Zeit an, welche wir am Anfang eingeplant haben (Soll). Die schwarzen Balken in den Blauen zeigen, wie viel Zeit wir tatsächlich benötigt haben (Ist).

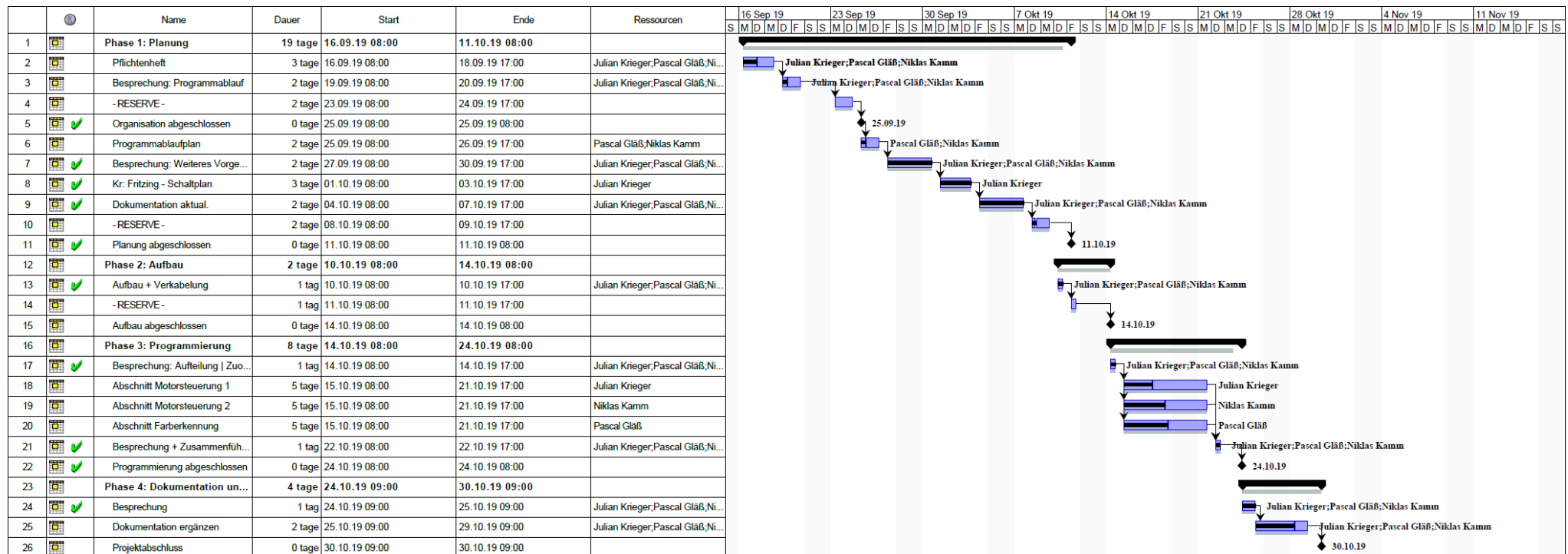
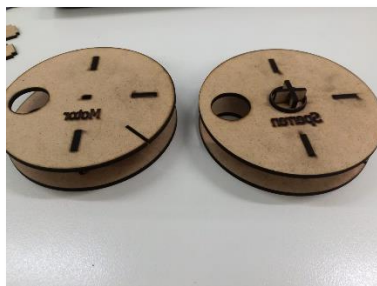
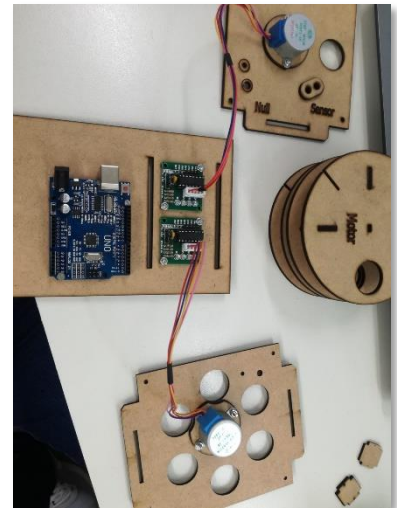
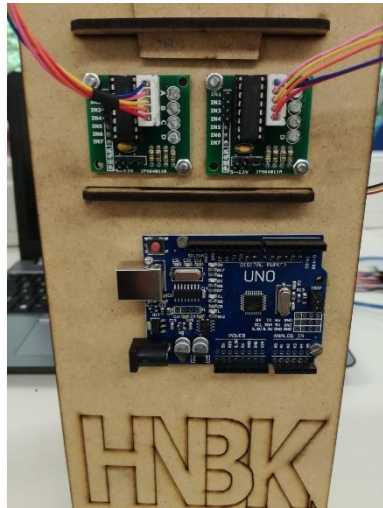


Abbildung 3: Gantt - Diagramm

4. Realisierung

4.1. Aufbau Hardware

Zum Start haben wir sämtliche elektronischen Bauteile auf ihre Funktionstüchtigkeit getestet. Mehr zu der Funktionsweise der verwendeten Programme finden Sie unter 4.2.1.



4.2. Programmierung

4.2.1. Testcodes

Vor dem Aufbau haben wir uns dafür entschieden, kurze Programmcodes zu schreiben, welche sämtliche Bauteile auf ihre Funktion überprüfen soll. Wir haben zwei simple Codes geschrieben, einen, um die Funktionstüchtigkeit der Motoren zu testen und einen, um sowohl die Lichtschranken als auch die Farberkennung zu testen. Letzteren haben wir außerdem benutzt, um die entsprechenden Sensorwerte zu messen. Beide Codes sind im Anhang unter 7.3 und 7.4 aufgelistet.

4.2.2. Niklas Kamm

Mein Programmier-Part war die Motorsteuerung. Das Problem hier ist, dass es schwieriger ist nur einen Teil zu programmieren als alle Module zusammen. Hinzu kommt, dass wir die Bauteile anfangs nicht zuhause hatten und daher nicht direkt testen konnten, ob alles

funktioniert. Dennoch hat die Umsetzung gut funktioniert, ich bin eigentlich auf keine Probleme gestoßen.

4.2.3. Julian Krieger

Mein Ziel war es Niklas' Part so einzubetten, dass wir im ,loop' lediglich eine Methode für sämtliche Steuerungen benötigen. Das bedeutet, es muss der richtige Motor und die richtige Lichtschranke für den gewollten Schritt angesteuert werden. Umgesetzt habe ich das, indem meine Methode mit 2 Parametern aus dem loop übergeben wird:

Erstens, den benötigten Motor, zweitens, die gewünschte Position. Die Positionen sind in Schritten (unseres Schrittmotors) gewählt.

Neben der Ansteuerung der Motoren war es außerdem wichtig, die richtige Lichtschranke anzusteuern, um Informationen über die Position der Scheibe zu bekommen. Dies habe ich gelöst, in dem eine Variable benutzt wird, in der, abhängig vom gewählten Motor, die passende Lichtschranke gespeichert wird. Beim Abfragen der Lichtschranke wird dann auf diese Variable zugegriffen.

4.2.4. Pascal Gläß

Bei dem Programmieren der Farbsortiermaschine, genauer gesagt, bei der Farberkennung, hatte ich so meine Schwierigkeiten. Zuerst wusste ich nicht genau wie ich an die ganze Sache heran gehen soll bzw. wie ich das ganze realisieren soll. Jedoch wurde dieses Problem durch den Programmablaufplan behoben.

Mit dem Teil des Programmablaufplans, welcher sich mit die Farberkennung befasste, hatte ich also einen strikten Plan wie ich vorzugehen habe. Der Rest wurde dann einfacher. Lediglich ein weiteres Problem hatte ich an der Stelle im Code, welcher sich mit der Funktion beschäftigt, die Farbe zu erkennen. Doch mit ein paar Hilfestellungen von Google und verstrichener Zeit durch eigene Überlegungen, konnte ich den ganzen Code fertigstellen.

5. Projektergebnisse

6. Projektbewertung

Unter diesem Punkt geben alle Projektmitglieder ihre persönlichen Meinungen, Erfahrungen, etc. getrennt ab.

6.1. Julian Krieger

Das Projekt hat mir sehr viel Spaß gemacht. Es war eine ideale Gelegenheit, den Ablauf eines Projekts in der IT-Branche kennenzulernen.

Mit den Jungs waren sämtliche Aufgaben, welche wir zusammen durchgeführt haben, schnell und vor allem gut erledigt. In unseren Besprechungen haben wir schnell Strategien zur Herangehensweise entwickelt und gute Lösungen gefunden. Bei den geteilten Aufgaben konnte man sicher auf die Kollegen verlassen, alle Aufgaben waren noch vor der geplanten Zeit fertiggestellt. Ich denke, dass jeder mit sehr viel Ehrgeiz an die Sache herangegangen ist und die Ergebnisse dies bestätigen können.

Meine mir selbst zusätzlich auferlegten Aufgaben, wie z.B. die Erstellung des 3D - Modells oder die Programmierung einer Website für das Projekt, haben mir sehr viel Spaß gemacht.

Ich habe das Projekt als Chance gesehen, über den Unterricht hinweg weitere Kenntnisse in anderen Bereichen zu erlangen.

6.2. Pascal Gläß

Mir persönlich hat das Projekt ‚Farbsortiermaschine‘ sehr gut gefallen. Das selbständig geplante Arbeiten, welches wir in der Gruppe organisieren sollten, hat mir besonders gut gefallen. Ich bin der Meinung, dass sich dadurch jeder in das Projekt einbringen konnte und man somit auch bewusst Schwächen, wie bei mir das Programmieren ausbessern kann. Ebenfalls bin ich der Meinung, dass das Projekt wie wir es kennenlernen durften zum Arbeiten anregt. Ich hatte das Gefühl, dass wir immer weiter machen wollten, auch wenn es nicht in den Zeitplan gepasst hätte. Erweiterungen würden mir hierbei nicht einfallen. Jedoch würde ich zwei Verbesserungsvorschläge einbringen:

- Schrittmotoren gegen Servomotoren austauschen. Das sorgt für eine einfachere Positionserkennung und Ausrichtung.
- Das Projekt wäre bei uns schon früher fertig gewesen, wenn wir die Bauteile früher zu Verfügung gehabt hätten.

6.3. Niklas Kamm

Das Projekt ist eine interessante Erweiterung der Bonbon-Abfüllanlage. Mit den verschiedenen Elementen des Projekts (Gantt - Diagramm, Schaltplan, usw.) lassen sich die Aufgaben auch gut auf das Team verteilen. Schade finde ich es, dass der RS Unterricht nur alle 2 Wochen stattfindet und wir daher nur begrenzt Zeit hatten, mit dem Lehrer persönlich Rücksprache zu halten.

7. Anhang

Der Anhang enthält alle bereits oben thematisierten Abbildungen bzw. Dateien im Detail.

Alle Bilder basieren auf Dateien, die mit der Dokumentation verfügbar sind. Diese sind auch als Datei einsehbar.

- 7.1. Programmablaufplan (via PapDesigner)
- 7.2. Programmcode - Main (via Arduino)
- 7.3. Programmcode - Motor testen (via Arduino)
- 7.4. Programmcode - Sensoren testen (via Arduino)