Farbsortiermaschine

Dokumentation

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

**Betreuer:** *Herr Hennig, HNBK*

**Zeitraum:** *11.09.2019 bis 01.03.2020*

Inhaltsverzeichnis

[1. Einleitung 1](#_Toc21014189)

[2. Projektbeschreibung 1](#_Toc21014190)

[2.1. Organisatorische Vorgaben 1](#_Toc21014191)

[2.2. Problemanalyse | Projekt – Ziel 1](#_Toc21014192)

[2.3. Beteiligte 1](#_Toc21014193)

[3. Projektplanung 2](#_Toc21014194)

[3.1. Verwendete Systeme und Programme 2](#_Toc21014195)

[3.1.1. Projektverwaltung 2](#_Toc21014196)

[3.1.2. Verwendete Software 2](#_Toc21014197)

[3.2. Soll – Planung 2](#_Toc21014198)

[3.2.1. Konzeption 2](#_Toc21014199)

[3.2.2. Aufbau 2](#_Toc21014200)

[3.2.3. Software – Planung 2](#_Toc21014201)

[3.2.4. Hardware – Planung 4](#_Toc21014202)

[3.2.5. Schaltplan 5](#_Toc21014203)

[3.3. Zeitplanung 6](#_Toc21014204)

[4. Realisierung 7](#_Toc21014205)

[4.1. Aufbau Hardware 7](#_Toc21014206)

[4.2. Programmierung 7](#_Toc21014207)

[4.2.1. Testcodes 7](#_Toc21014208)

[4.2.2. Niklas Kamm 7](#_Toc21014209)

[4.2.3. Julian Krieger 7](#_Toc21014210)

[4.2.4. Pascal Gläß 7](#_Toc21014211)

[5. Projektergebnisse 7](#_Toc21014212)

[6. Projektbewertung 7](#_Toc21014213)

[7. Anhang 7](#_Toc21014214)

[7.1. Ablaufdiagramm (via PapDesigner) 8](#_Toc21014215)

[7.2. Schaltplan (via Fritzing) 8](#_Toc21014216)

[7.3. Gantt – Diagramm (via ProjectLibre) 8](#_Toc21014217)

[7.4. Programmcode (via Arduino) 8](#_Toc21014218)

Einbindungsverzeichnis

[Video 1: Das Gestell erklärt (YouTube) 2](#_Toc20955645)

[Video 2: Die Sortierscheibe erklärt (YouTube) 3](#_Toc20955646)

[Video 3: Die Sperrscheibe erklärt (YouTube) 3](#_Toc20955647)

Abbildungsverzeichnis

[Abbildung 1: Programmablaufplan, grundlegend 4](file:///C:\Users\Julian\Documents\GitHub\color_sorting_machine\documentation\main_documentation.docx#_Toc20955752)

[Abbildung 2: Schaltplan 6](#_Toc20955753)

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. Die Projektarbeit inklusive Dokumentation ist geistiges Eigentum nach §2 UrhG und darf lediglich durch Rechteinhaber (Autoren) und berechtige Dritte (Angestellte Lehrer des Heinz – Nixdorf Berufskollegs) nach §§ 15, 16, 17,18 & 19 UrhG verwertet werden.

1. Projektbeschreibung
   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.

* 1. 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.

* 1. 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.

1. Projektplanung
   1. Verwendete Systeme und Programme
      1. Projektverwaltung

All unsere Dateien wie der Programmcode, die Ablaufpläne, die Dokumentation, etc. haben wir über ["GitHub"](https://github.com/kriegersBlog/color_sorting_machine) 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.

* + 1. Verwendete Software

Eine Übersicht über sämtliche benutzte Software finden Sie [hier](https://rs.krieger-blog.de/Software.html). Jedes Programm ist mind. für das Betriebssystem Windows 10 verfügbar. Genutzt haben wir neben Windows 10 auch Ubuntu 19.03.

* 1. Soll – Planung
     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.

* + 1. Aufbau

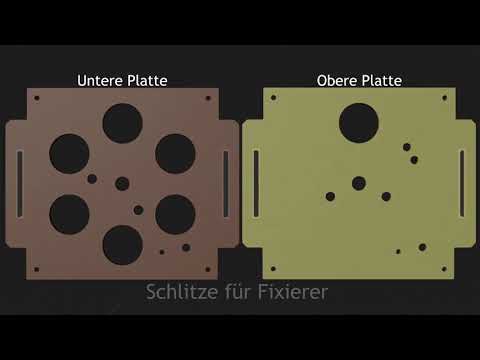
Das Gerät lässt sich grob in drei Teile aufteilen:

* 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   
Julian Kriegers Website *„*[rs.krieger-blog.de](https://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

[](https://www.youtube.com/watch?v=jNPcI48agFU)

Video : 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/Gestell.html)

[MP4 über *rs.krieger-blog.de* abrufen](https://rs.krieger-blog.de/Gestell_erklaert.mp4)

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

#### Die Sortierscheibe

[](https://www.youtube.com/watch?v=aMCSTyTmY-k)

Video : 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 Behältnis für die Kugel.

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

[MP4 über *rs.krieger-blog.de* abrufen](https://rs.krieger-blog.de/Sortierscheibe_erklaert.mp4)

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

#### Die Sperrscheibe

[](https://www.youtube.com/watch?v=wi2LKV1NE64)

Video : 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 besitz die Sperrscheibe noch einen 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/Sperrscheibe.html)

[MP4 über *rs.krieger-blog.de* abrufen](https://rs.krieger-blog.de/Sperrscheibe_erklaert.mp4)

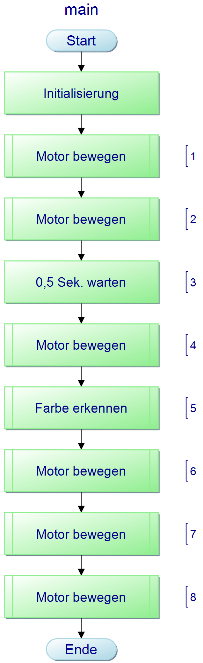
[GIF über *rs.krieger-blog.de* abrufen](https://rs.krieger-blog.de/Sperrscheibe_erklaert.gif)

* + 1. 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 : 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 wird auf Englisch verfasst und kommentiert
2. Jede Zeile wird zumindest kurz mit einem Kommentar zusammengefasst
3. Korrektes Einrücken ist wichtig für strukturierten Code!
4. Jede Methode bekommt einen eigenen Kommentar, welcher die Funktion beschreibt
5. Übergebene Parameter (wenn vorhanden) werden intern mit einem „\_“ am Anfang benannt
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**.

* + 1. 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.

* + 1. 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. Eine kurze Übersicht findet man [hier](https://krieger-blog.de/2019/09/15/fritzing-bauelemente-aus-dem-unterricht/).

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](https://krieger-blog.de/2019/09/15/normgerechte-fritzing-symbole/) einseh- und herunterladbar.

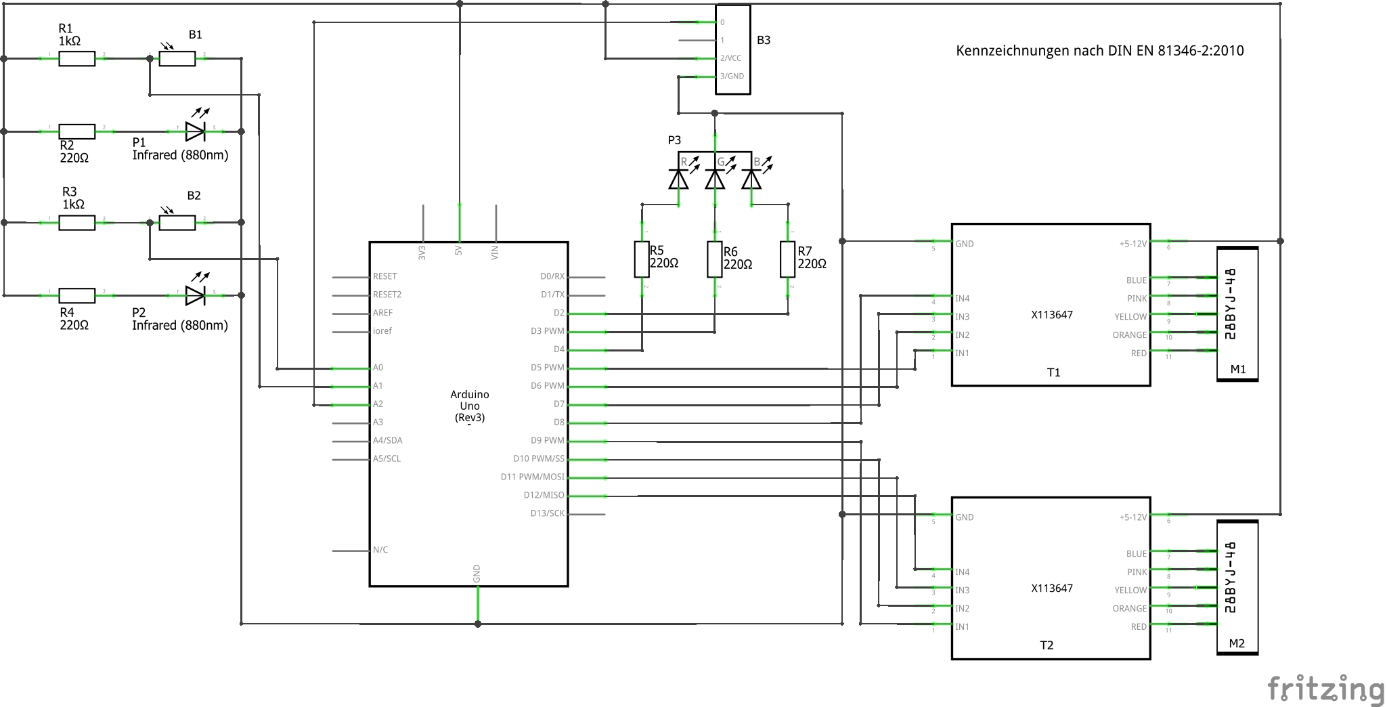


Abbildung : Schaltplan

* 1. 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.

* Screenshot GANTT Diagramm – Kurze Erklärung + Verwunderung warum so schnell
* Möglichst kurzhalten

1. Realisierung
   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**.

* Fotos von elektrischem Test
* Mit Bildern dokumentieren / Foto von Pins bzw. Anschlüssen
* Kurze Erklärungen zu Bildern
  1. Programmierung
     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.

* + 1. Niklas Kamm
* Erklärung was die eigene Aufgabe ist
* PAP Screen vom eigenen Teil
* Kommunikation
  + 1. Julian Krieger
    2. Pascal Gläß

1. Projektergebnisse

* Anschauliche Darstellung der Ergebnisse

1. Projektbewertung

* Fazit, ggf. neue Fragestellungen
* Jeder seine eigene Aussage / Meinung -> Aufteilung in 3 Bereiche
* Kritische Eigenreflexion
* Mögliche zukünftige Erweiterungen
* Verbesserungsvorschläge

1. 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.

* 1. Ablaufdiagramm (via PapDesigner)
  2. Schaltplan (via Fritzing)
  3. Gantt – Diagramm (via ProjectLibre)
  4. Programmcode (via Arduino)