

Dreifarbiger LEGO[®]-Drucker

Finn Karstens, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Drucker sind aus dem modernen Leben kaum noch wegzudenken. Sie übertragen digitale Inhalte auf Papier und müssen dabei präzise und zuverlässig funktionieren. Naheliegend ist die Idee, eine ähnliche Maschine mithilfe der LEGO[®]-Mindstorms[®]-Plattform zu entwickeln. Als zusätzliche Herausforderung und Weiterentwicklung existierender LEGO[®]-Drucker wurde hier eine Farbwechselfunktion eingebaut, welche während des Druckes zwischen drei Farben wechseln kann. Die vorliegende Arbeit beschäftigt sich mit der Entwicklung eines solchen Prototypen. Sie dokumentiert wesentliche Design-Entscheidungen, stellt Programm-Abläufe dar und setzt sich mit der finalen Konstruktion auseinander. Dabei werden Herausforderungen wie Stabilität und Motoransteuerung weiter erläutert und mögliche Lösungen dargestellt.

Schlagwörter—Drucker, Farbwechsel, Motoransteuerung, Rastergrafik, Vektorgrafik

I. EINLEITUNG

DRUCKER gehören wohl mitunter zu den meist gehassten Maschinen in den Büros dieser Welt. Unzuverlässigkeit, Papierstaus und Probleme mit der Tinte sind häufige Unannehmlichkeiten [1]. Aber wie kompliziert ist es wirklich, beliebige digitale Inhalte von einem Computer auf ein Blatt Papier zu übertragen?

Das Ziel, in unter zwei Wochen ein Gerät aus Klemmbausteinen zu entwickeln, das vergleichbar mit kommerziellen Lösungen ist, wäre etwas hochgesteckt. Daher wurden einige Vereinfachungen festgelegt, die ebenfalls im Einklang mit den technischen Möglichkeiten der vorgeschriebenen LEGO[®]-Mindstorms[®]-Plattform stehen.

- 1) Sehr stark reduzierte Auflösung
- 2) Keine Mischung beliebiger Farben
- 3) Software-Umgebung auf MATLAB beschränkt
- 4) Keine Scan-Funktion

II. VORBETRACHTUNGEN

Die Idee, einen Drucker aus LEGO[®]-Steinen zu bauen, ist schon seit den Anfängen der LEGO[®]-Mindstorms[®]-Produktreihe vorhanden [2]. In bekannten Online-Portalen gibt es diverse Videos und andere Inhalte, in denen Leute ihre Ansätze und Umsetzungen präsentieren, z. B. [3], [4]. Hier war der Anspruch, einen Schritt weiter zu gehen, um so viel wie möglich in der gegebenen Zeit aus den Materialien herauszuholen.

A. Bereits existierende Umsetzungen

Im Internet existieren bereits viele Videos und Bauanleitungen zu LEGO[®]-Druckern. Dabei gibt es hauptsächlich zwei verschiedene Design-Konzepte.

1) *Roboter-Arm*: Diese Variante basiert auf einer Art Greifarm, der sich zu jedem beliebigen Punkt einer Ebene bewegen kann. Dies geschieht in einer 2D-Ebene mit einem zusätzlichen Motor für das Heben und Senken des Stifts. Der Ansatz ist besonders gut geeignet für das Zeichnen von runden Formen, da dies der natürlichen Bewegung der Motoren entspricht. Allerdings kommt es zu erheblichen Hebelwirkungen, falls das Bild, das gezeichnet werden soll, größer als ein bestimmter Bereich ist. Dies hat negative Auswirkungen auf die Präzision und das Endergebnis.

2) *X-Y-Drucker*: Dieser Ansatz entspricht einem handelsüblichen Drucker am ehesten. Hierbei wird ein Blatt durch einen Papiereinzug in y-Richtung hin und her bewegt und ein Druckkopf bringt die Farbe in x-Richtung am jeweiligen Punkt auf das Papier.

Letztendlich wurde als Grundlage der x-y-Drucker gewählt, da er einem kommerziellen Gerät am ehesten entspricht und sich verhältnismäßig einfach ansteuern lässt. Durch diese Wahl konnte der Fokus auf andere Probleme, wie Farbwechsler und Stabilität, gerichtet werden.

B. Zusätzliche Herausforderungen

Nahezu alle der bereits existierenden LEGO[®]-Drucker haben eine Gemeinsamkeit: Sie können lediglich eine Farbe darstellen. Da nur 3 Motoren für die Funktionalität eines einfachen Druckers benötigt werden, steht bei der Verwendung eines LEGO[®] Mindstorms[®] EV3s mit 4 Motor-Anschlüssen noch ein weiterer Motor für die Umsetzung eines Farbwechsel-Mechanismus zur Verfügung. Dieser Mechanismus muss also mit einem Motor auskommen, aber auch robust und mit möglichst geringen Toleranzen funktionieren, da sonst die Positionierung des Stiftes unzuverlässig ist. Gerade bei der Verwendung von Klemmbausteinen sind Toleranzen und Spiel im Mechanismus ein besonders häufiges Problem, das spezielle Bautechniken zur Stabilisierung erfordert [5].

Ein weiterer Faktor, der die Entwicklung beeinflusst, ist die Verwendung von Vektor- oder Rastergrafiken. Die Umsetzung des Druckers ist hier bewusst allgemein gehalten, sodass sowohl das Drucken von Vektor- als auch Rastergrafiken möglich sein soll.

III. UMSETZUNG

Der Prozess, einen Drucker zu bauen und zu programmieren, wurde in mehrere Unterschritte eingeteilt. Einerseits wurde beim Bau des Druckers zwischen Basis und Druckkopf unterschieden. Die Ergebnisse der parallelen Arbeit wurden dann später zusammengeführt. Andererseits wurde erst mit der Programmierung begonnen, als bereits ein erster Prototyp

fertiggestellt war. Dadurch konnte dieser dann validiert und im Nachhinein noch etwas weiter angepasst werden.

A. Basis

Die Basis bildet das Grundgerüst des Druckers. Sie muss stabil sein und darf nicht schwingen, damit der Druckkopf auf den dafür vorgesehenen Schienen fahren kann, ohne dass es dabei zur ungewollten Verschiebung der Position der Stifte kommt. Außerdem verfügt sie über einen Motor, der für den Papiereinzug und die Bewegung entlang der y-Achse zuständig ist. Wie in Abbildung 1 zu sehen, ist ein zusätzlicher Ultraschall-Sensor vorhanden, um zu erkennen, ob ein Blatt erfolgreich eingezogen wurde. Dadurch kann das Papier in die richtige Position zum Drucken gebracht werden und es wird verhindert, dass der Drucker sich selbst bedruckt.

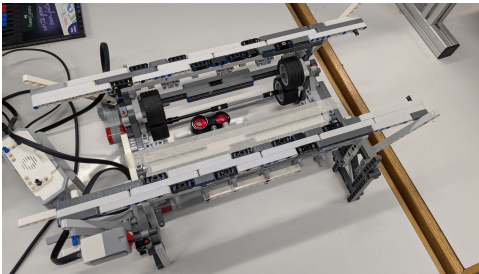


Abbildung 1. Konstruktion der Basis des Druckers

Ein zusätzlicher Drucksensor ist für die Erkennung der Nullposition der x-Achse zuständig und wird für die Kalibrierung benutzt. Diese erfolgt wie in Abbildung 2 dargestellt. Dabei fährt der Drucker von einer beliebigen Startposition aus immer in die Nullposition und dann die maximal mögliche x-Strecke und wieder zurück zur Nullposition. Sollte es dabei zu einem Problem kommen, wird dies noch vor Beginn des eigentlichen Druckes erkannt und kann behoben werden.

Durch die Kalibrierung wird sichergestellt, dass der Drucker nachfolgend immer in gleicher Ausgangslage auf der x-Achse ist. Dies funktioniert trotz des Spiels der Klemmbausteine verhältnismäßig zuverlässig.

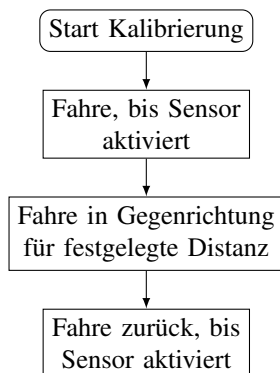


Abbildung 2. Programmablaufplan zur Kalibrierung einer Achse

B. Druckkopf

Das zentrale Element des Druckkopfes ist der Farbwechsler. Dieser wird in Abschnitt III-C detailliert betrachtet. Dazu

kommt ein Hebe-Senk-Mechanismus, der die Höhe aller Stifte steuert. Somit wird sichergestellt, wann der aktive Stift aufdrückt oder nicht. Ein weiterer Motor steuert die Position des Druckkopfes entlang der x-Achse.

Abbildung 3 zeigt den fertigen Druckkopf.

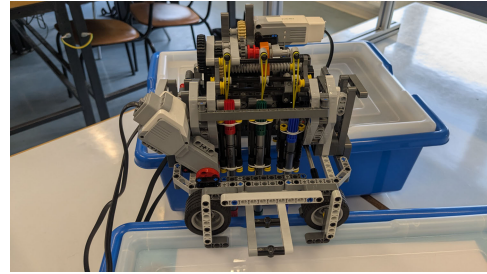


Abbildung 3. Konstruktion des Druckkopfes des Druckers

C. Farbwechsler

Das Merkmal, das den hier entwickelten Drucker von bereits existierenden Konstruktionen unterscheidet, ist der Farbwechsler. Dieser Mechanismus sorgt dafür, dass ein Stift von dreien als primärer Stift ausgewählt werden kann.

Dabei wird ein kleiner Schlitten durch eine Schneckenschraube an die jeweilige Position des Stiftes bewegt. Dies sorgt dafür, dass der Stift an der jeweiligen Stelle des Schlittens nach vorne und gleichzeitig nach unten gedrückt wird. An den Positionen, an denen der Schlitten nicht ist, klappen die Stifte automatisch wieder nach oben und hinten.

Damit der Drucker immer weiß, wo sich der Schlitten befindet, durchläuft auch der Farbwechsler zu Beginn des Druckes eine Kalibrierungsroutine. Diese verläuft analog zu Abbildung 2 und der in Abschnitt III-A beschriebenen Kalibrierung der x-Achse. Durch das Ausführen dieses Ablaufs wird sichergestellt, dass der Mechanismus funktionsfähig ist und sich in Nullposition befindet.

Die Wahl der Stifte ist eine weitere wichtige Entscheidung. Hier wurde sich für Filzstifte entschieden, da dort die Linienqualität weniger stark vom exakten Winkel abhängt, in dem der Stift aufkommt. Darüber hinaus ist die Spitze auch etwas flexibler, was Toleranzen, die durch die LEGO®-Konstruktion entstehen, wieder ausgleicht.

D. Codeaufbau in MATLAB

Den Kern der Software des Druckers bildet die EV3 Toolbox für MATLAB der RWTH Aachen. Diese erlaubt es, dem EV3 Befehle über eine aktive USB-Verbindung zu senden und dadurch Motoren und Sensoren anzusprechen. Der nötige Code zur Ansteuerung wurde in Funktionen geschachtelt und diese wiederum zu einer eigenen Klasse verbunden. Dies hat den Vorteil, dass sinnhafte Code-Abschnitte einfach wiederverwendet werden können und die konkrete Steuerung des Druckers an einem anderen, übersichtlicheren Ort stattfinden kann. Zu diesem Zweck wurde der Live-Editor von MATLAB verwendet. Dort können simple Code-Segmente erstellt werden, die dann Funktionen der Drucker-Klasse aufrufen. So ist es z. B. möglich, eine Sektion zu erstellen, die den Drucker initialisiert und eine

Linie druckt. Ein weiterer Vorteil dieser Struktur liegt in der Erstellung von Drucker-Skripten. Dies findet bei kommerziellen Druckern ebenfalls in Form von PostScript Anwendung [6]. Durch die Erstellung solcher Skripte gibt es einen einfachen Weg, die primitiven Funktionen, die durch die Drucker-Klasse bereit gestellt werden, zu komplexeren Abläufen zu vereinen. Ein Beispiel ist das Drucken eines Quadrats. Dafür werden lediglich Funktionen zum Heben und Senken des Druckkopfes, Farbauswahl und das Fahren entlang einer geradlinigen Bahn benötigt. Diese werden durch die Drucker-Klasse bereitgestellt.

E. Drucken einer Rastergrafik

Das Drucken von Rastergrafiken ist mit diesem Drucker im Vergleich zu Vektorgrafiken einfacher, da beide Achsen getrennt voneinander an die richtige Position fahren können. Dies vereinfacht die Motorsteuerung.

Dafür wird die gewünschte Grafik in MATLAB als Matrix geladen und richtig orientiert. Danach fährt der Drucker jeden Pixel, sprich jede Position der Matrix, welche auf das Blatt übertragen wird, ab und platziert dort einen Farbpunkt, wo der Farbwert des Pixels es vorsieht. Dies wird detailliert durch den Ablaufplan 4 beschrieben.

Anzumerken ist, dass die Initialisierung auch die Kalibrierungen beinhaltet, die in Abbildung 2 dargestellt und in den Abschnitten III-A und III-C näher erläutert werden. Ein wohldefinierter Ausgangszustand wird dadurch zu Beginn eines Druckes sichergestellt.

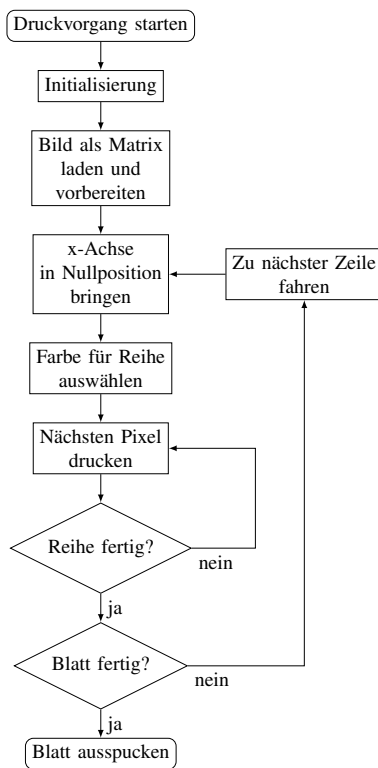


Abbildung 4. Programmablaufplan zum Drucken einer Rastergrafik

F. Drucken primitiver Formen

Neben dem Verarbeiten einer Rastergrafik ermöglicht die Verwendung eines Stiftes auch das Zeichnen von Elementen

einer Vektorgrafik. Die primitivste Form neben dem Punkt ist die Linie. Anstatt einen vollwertigen Parser für Vektorgrafiken in MATLAB zu schreiben, wurde zunächst nur die Implementierung dieser Linienfunktion in Betracht gezogen.

Für die genaue Kontrolle der Motoren ist eine erweiterte Motoransteuerung nötig. Diese wird genauer in Abschnitt III-G behandelt. In der Zeit des Seminars konnte keine Funktion zum Zeichnen von Geraden mit beliebigen Winkeln mehr fertiggestellt werden. Die aktuelle Implementierung umfasst das Zeichnen von 0°-, 45°- und 90°-Linien. Damit können bereits Quadrate, Rechtecke und weitere primitive Formen mit 45°- und 90°-Winkeln gezeichnet werden. Außerdem wäre eine Darstellung von Zahlen denkbar, die einer Sieben-Segment-Anzeige gleicht. Jedoch konnte dies auch aus zeitlichen Gründen nicht mehr fertiggestellt werden.

G. Motoransteuerung

Bei der Motorsteuerung müssen einige Eigenheiten der verwendeten LEGO®-Mindstorms®-Plattform beachtet werden, um optimale Ergebnisse zu erzielen.

1) *Brake-Mode*: Der Brake-Mode definiert das Verhalten der Motoren, nachdem sie gestoppt wurden. Im Falle des Druckers ist es essenziell, dass die Motoren nach dem Stoppen ihre Position halten. Das ist zwar mit erhöhtem Energieaufwand verbunden, sorgt aber dafür, dass z.B. die Position des Druckkopfes nicht verfälscht wird. Deshalb muss der Brake-Mode auf „brake“ gestellt werden.

2) *Synchrone Ansteuerung*: Um eine gleichmäßige und vorhersehbare Bewegung der Motoren zu erhalten, ist es nicht ausreichend, beide Motoren für x- und y-Achse nacheinander mit jeweils einem Befehl zu starten. Durch die Latenz zwischen zwei Befehlen drehen beide Motoren nicht zur exakt selben Zeit und bewegen sich daher nicht so wie gewünscht. Deshalb müssen beide Motoren mittels Motorsteuerung synchronisiert werden. Die Firmware des EV3s stellt dafür eine Funktion zur Verfügung, die zwei Motoren mit unterschiedlicher Geschwindigkeit synchron ansteuern kann [7].

Diese Funktion benötigt einen zusätzlichen Parameter, `turnRatio`. Die Implementierung dieser Ansteuerung funktioniert dabei folgendermaßen: Der Hauptmotor bekommt Geschwindigkeit, Anzahl an Umdrehungen, den Nebomotor und `turnRatio` als Parameter. So lange sich der Hauptmotor dreht, wird sich der Nebomotor ebenfalls drehen und nimmt dabei die Geschwindigkeit `turnRatio` an. Zusätzlich kann durch Vorzeichen und Addition/Subtraktion von 100 die Richtung beider Motoren gesteuert werden.

Bei einem symmetrischen Aufbau der Motoren ist diese Variante gut geeignet, um eine einfache Lenkung zu realisieren. Da im Fall des Druckers der Aufbau aus Platzgründen nicht symmetrisch ist und eine Zahnrad-Übersetzung vorhanden ist, erschwert dies allerdings die Motoransteuerung. Drehen sich beide Motoren gleich schnell und viel, entsteht im Normalfall eine 45°-Linie. Um dieses Ergebnis hier zu erreichen, muss die entsprechende `turnRatio` mit gegebener Formel (2) errechnet werden. Das benötigte Übersetzungsverhältnis (1) wurde mit einem Online-Tool berechnet [8].

$$r_x = 0,6 \quad (1)$$

$$t = (1 - r_x) \cdot 100 \quad (2)$$

Zur Umrechnung von der Distanz in mm in Tacho-Einheiten, mit denen die EV3 Toolbox arbeitet, wurde im MATLAB-Stil eine Umrechnungs-Matrix verwendet. Der Vektor mit Maßangaben in mm wird dabei durch eine Matrix-Multiplikation mit der Umrechnungsmatrix A (4) in Grad-Motorumdrehung umgerechnet. Die Konstante (3a) gibt den Durchmesser des verwendeten LEGO®-Reifens an [9]. Die experimentell bestimmte Kompression der Gummi-Reifen durch Eigengewicht wird durch (3b) und (3c) berücksichtigt. Der resultierende Vektor enthält die nötigen Motorumdrehungen in Grad, mit denen die angegebene Strecke trotz Berücksichtigung der speziellen Motoransteuerung zurückgelegt werden kann.

$$d = 43,2 \text{ mm} \quad (3a)$$

$$c_x = 0,955 \text{ mm} \quad (3b)$$

$$c_y = 0,0 \text{ mm} \quad (3c)$$

$$A = \begin{pmatrix} \frac{360^\circ}{\pi \cdot (d - c_x)} \cdot \frac{1}{r_x} & 0 \\ 0 & \frac{360^\circ}{\pi \cdot (d - c_y)} \end{pmatrix} \quad (4)$$

Um grobe Ungenauigkeiten und stärkere Schwingungen zu vermeiden, wurde nach verschiedenen Tests die Leistung der Motoren vorerst auf 35% begrenzt.

IV. ERGEBNISDISKUSSION

Trotz der verhältnismäßig kurzen Entwicklungszeit hat der Drucker brauchbare Ergebnisse produziert, die zusätzlich durch Zeitraster-Aufnahmen festgehalten werden konnten. Die Ursachen für die momentanen Limitierungen sind Zeitmangel und die LEGO®-Mindstorms®-Plattform selbst. Mit etwas mehr Zeit und Budget würden sich Probleme wie Kabellänge und Stabilität bzw. Toleranzen der Konstruktion sicherlich noch weiter optimieren lassen. Ein weiteres Problem ist die Fixierung des Blattes. Durch das momentan lose Aufliegen entsteht ein kleiner Luftspalt zwischen Blatt und Basis, der beim Aufdrücken des Stifts für eine leichte Vorwärtsbewegung sorgt. Dies könnte durch eine Weiterentwicklung der Basis verbessert werden.

Außerdem könnte eine Verbesserung des Farbwechslers nicht nur die Zuverlässigkeit beim Wechseln der Farbe erhöhen, sondern auch die Höhe der Stifte fixieren. Dadurch könnte sichergestellt werden, dass alle Farben gleich stark aufdrücken, was momentan nicht der Fall ist. Verbesserungen des Codes könnten zudem das Zeichnen von Geraden mit beliebigem Winkel sowie von runden Formen ermöglichen. Auch könnte die Druckzeit noch durch einige Optimierungen bei der Bewegung verringert werden. Ein weiterer Drucksensor am anderen Ende der Schiene könnte die Maßgenauigkeit während des Druckes noch erhöhen. Da die Gesamtkonstruktion jedoch eine gewisse Größe erreicht hat, war die verfügbare Kabellänge nicht ausreichend.

V. THEORETISCHE WEITERENTWICKLUNG

Mit mehr Zeit und Ressourcen könnte man zunächst die in Abschnitt IV angesprochenen Verbesserungen durchführen. Wie in Abschnitt III-F bereits erwähnt, arbeiten kommerzielle Drucker hauptsächlich mit PostScript [6]. Dies und ein Vektorformat (z. B. SVG) könnten unterstützt werden und somit eine deutlich bessere Anbindung an bereits etablierte Technologien ermöglichen. Das Zeichnen von runden Formen ist zwar prinzipiell möglich, setzt aber die Steuerung der Beschleunigung der Motoren voraus. Dies ist nicht in der Firmware für den EV3 vorgesehen, könnte sich aber möglicherweise durch schrittweises Ändern der Geschwindigkeit simulieren lassen. Zusätzlich könnte neben einer fortgeschrittenen Motorsteuerung auch noch eine fortgeschrittenere Bildverarbeitung implementiert werden. Diese wäre dann in der Lage, Rastergrafiken runter zu skalieren und Verarbeitung von Pixelwerten zu ermöglichen, die irgendwo zwischen den Farben der vorhandenen Stiften liegen.

VI. ZUSAMMENFASSUNG UND FAZIT

Trotz all der offenen Verbesserungsmöglichkeiten wurde die vorhandene Zeit sehr effektiv genutzt und es konnte ein brauchbares Ergebnis erzielt werden. Bei der Konstruktion und Programmierung einer Drucker-ähnlichen Maschine müssen viele Faktoren berücksichtigt und ein nicht unerheblicher Anteil der Zeit für das Testen eingeplant werden. Durch leichte Hardware-Anpassungen und einige Nachbesserungen am Code könnte aus der finalen Konstruktion noch viel mehr herausgeholt werden, als in der zur Verfügung stehenden Zeit möglich war. Es zeigt sich, dass es nicht unbedingt schwer ist, einfache Inhalte vom Computer auf ein Blatt zu übertragen. Der Teufel liegt jedoch im Detail und je brauchbarer das Ergebnis sein soll, desto mehr Aufwand fließt in die Optimierung von Mechanismen und des Codes.

LITERATURVERZEICHNIS

- [1] DAVID LEE: Drucker: Ist der Ruf erst ruiniert, wirbt es sich ganz ungeniert. <https://www.galaxus.de/de/page/drucker-ist-der-ruf-erst-ruiniert-wirbt-es-sich-ganz-ungeniert-30891>. – Abgerufen: 26.02.2024
- [2] SPILLERREC: *Lego RCX Plotter*. YouTube. <https://www.youtube.com/watch?v=IHMDctb1Qo>. – Abgerufen: 26.02.2024
- [3] JK BRICKWORKS: *LEGO Printer*. YouTube. <https://www.youtube.com/watch?v=dHmgALgFRGM>. – Abgerufen: 26.02.2024
- [4] LAURENS VALK: *LEGO MINDSTORMS NXT 2.0 Plotter / Printer*. YouTube. <https://www.youtube.com/watch?v=tNp62EfaJ60>. – Abgerufen: 26.02.2024
- [5] UNBEKANNT: *Tolerances, accuracies and their sensible limits*. LEGOISM.INFO. <https://legoism.blogspot.com/2017/01/tolerances-accuracies-and-their.html>. – Abgerufen: 26.02.2024
- [6] WIKIPEDIA: *PostScript*. <https://de.wikipedia.org/wiki/PostScript>. – Abgerufen: 26.02.2024
- [7] MINDBOARDS: *LEGO MINDSTORMS EV3 source code*. GitHub. https://github.com/mindboards/ev3sources/blob/78ebaf5b6f8fe31cc17aa5dce0f8e4916a4fc072/lms2012/c_output/source/c_output.c#L71. – Abgerufen: 26.02.2024
- [8] UNBEKANNT: *LEGO Gear Ratio Calculator*. <https://gears.sariel.pl/>. – Abgerufen: 26.02.2024
- [9] UNBEKANNT: *LEGO Wheels Chart*. <http://www.wheels.sariel.pl/>. – Abgerufen: 26.02.2024