



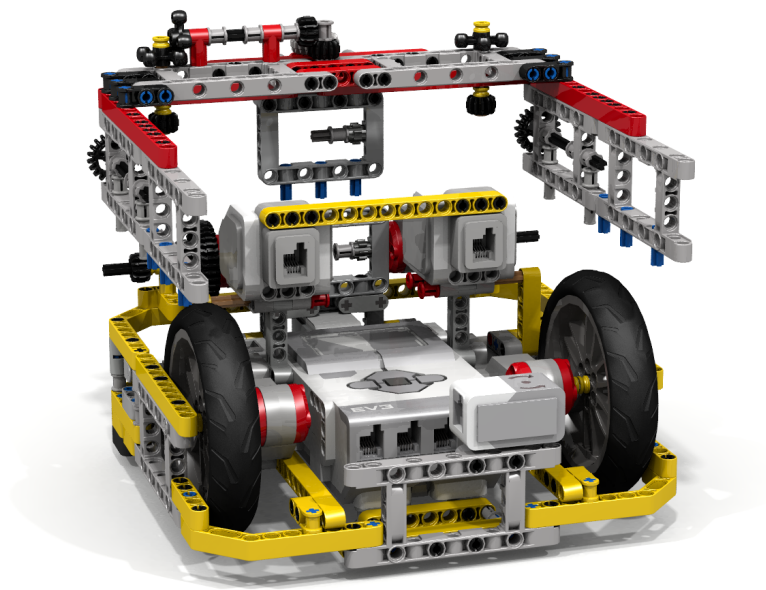
OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

EIT

FAKULTÄT FÜR
ELEKTROTECHNIK UND
INFORMATIONSTECHNIK

LEGO-Praktikum. entwickeln + programmieren + optimieren

Berichte der Studierenden zum Projektseminar
Elektrotechnik/Informationstechnik



„Flying Box Turtle Lego EV3 Robot“ von David Luders via Flickr (<https://flic.kr/p/AjQviw>)
veröffentlicht unter der Lizenz CC BY-SA (<https://creativecommons.org/licenses/by-sa/2.0/>)

Eine Schriftenreihe der Otto-von-Guericke-Universität Magdeburg, Fakultät für Elektrotechnik- und Informationstechnik, Institut für Medizintechnik sowie Institut für Elektrische Energiesysteme

Herausgeben von:

Mathias Magdowski, Thomas Schallschmidt, Thomas Gerlach und Enrico Pannicke

Band 5 vom Wintersemester 2021/2022

Inhaltsverzeichnis

Gruppe 1	1
1.1 Melody – Scan einer Musikspur auf Papier per LEGO-Roboter (Eric Koch)	1
1.2 Melody, der Musikbot (Maximilian Weitzel)	5
Gruppe 2	9
2.1 Bewässerungsroboter (Matt Müller)	9
2.2 Automatischer Bewässerungsroboter (Alexander Brian Schwerin)	13
Gruppe 3	17
3.1 Sally can Draw: Automatisierter Light-Painting-Roboter (Michelle Sophia Horn)	17
3.2 Sally Can Draw – Ein Automatisierter Lightpainting Roboter (Anna-Lena Thalhofer)	21
Gruppe 4	24
4.1 Stationärer Raumscanner mit Abstandssensoren (Steffen Bach)	24
Gruppe 5	28
5.1 Automatisierung einer Kaffeemaschine (Tim Holzki)	28
Gruppe 6	32
6.1 Farbsortierroboter (Badr Abdi)	32
Gruppe 7	35
7.2 Sprachgesteuerter Getränkeautomat mit Lego Mindstorms (Etienne Grundmann)	35
7.3 Sprachgesteuerter Getränkeautomat mit LEGO Mindstorms (Jannik Seidler)	39
Gruppe 9	43
9.1 Automatisierter Kartenmischer (Erik Borowski)	43
Gruppe 10	46
10.1 Farbsortierroboter (Hassan Al-Homada)	46
10.2 Farbsortierroboter (Hamza Alali)	49
10.3 Farbsortierroboter (Mohamad Jumaa Zamzam Kazak)	52

IMPRESSUM

Herausgeber: Mathias Magdowski, Thomas Schallschmidt, Thomas Gerlach und
Enrico Pannicke
Institut für Medizintechnik, Institut für Elektrische Energiesysteme
Fakultät für Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg
Postfach 4120, 39016 Magdeburg

DOI: 10.24352/UB.OVGU-2022-032

ISSN: 2629-6160

Redaktionsschluss: April 2022

Seminarzeitraum: 31. Januar – 14. Februar 2022

Bezug: Open Access, Digitale Hochschulbibliothek Sachsen-Anhalt
<http://edoc2.bibliothek.uni-halle.de/>

Dieses Werk ist unter einer Creative-Commons-Lizenz vom Typ Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International (CC BY-SA 4.0) zugänglich.

Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie <https://creativecommons.org/licenses/by-sa/4.0/deed.de> oder wenden Sie sich an Creative Commons, PO Box 1866, Mountain View, CA, 94042, USA.

1. Auflage, Magdeburg, Otto-von-Guericke-Universität, 2022

Erstellung des Sammelbandes mittels \LaTeX , `hyperref` und `pdfpages`

Melody – Scan einer Musikspur auf Papier per LEGO-Roboter

Eric Koch, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Das LEGO-Mindstorms-Projektseminar 2022 fordert nach einem Roboter, welcher anhand des NXT-Kontrollsteins gesteuert wird. Er soll aus LEGO-Bausteinen, Sensoren und Motoren sein, welche die Grundlage für verschiedenen Funktionen bieten. Dieser Roboter soll eine Musikspur, welche auf einem Blatt Papier dargestellt ist, auslesen. Diese Musikspur soll währenddessen interpretiert und die gelesenen Noten zeitgleich abgespielt werden. Im Folgenden werden die Konstruktion, Programmierung und die musiktheoretischen Hintergründe des Roboters, sowie eine neue Kodierung vorgestellt.

I. EINLEITUNG

ROBOTER sind in der heutigen Zeit essentiell. Ob nun einfache mechanische Arbeit auszuführen ist, oder schwierigere, situationsbedingte Aufgaben anstehen, sind sie als Ersatz für Menschen nicht wegzudenken. Zu den wichtigen Aufgaben gehören, unter anderen, das Auslesen und Interpretieren von Informationen in unterschiedlichster Form. So ist zum Beispiel das Scannen von Barcodes, wie etwa der Pfandautomat im Supermarkt, eine fast alltägliche Arbeit die ein Roboter zu leisten hat. Im LEGO-Mindstorms wurde ein ähnliches Problem von Melody gelöst. Melody ist ein LEGO-Roboter und hat die Aufgabe, eine gegebene Musikspur zu scannen und zu interpretieren. Dabei sollen möglichst viele, unterschiedliche Töne spielbar sein, sodass der Tonumfang unseres Roboters groß genug ist, um sinnvolle Lieder oder Melodien abspielen zu können. Außerdem soll die Musikspur zeitgleich vom NXT-Baustein ausgegeben werden, sodass die Melodie während des Scans zu hören ist.

II. VORBETRACHTUNGEN

Der NXT-Baustein lässt aufgrund der Anschlüsse vier Sensoren und drei Motoren zu. Die Motoren dienen ausschließlich der Bewegung, weshalb man zum Scannen lediglich die vier Sensoren benutzen kann. Außerdem ist der NXT-Baustein durch einen Befehl in der Lage, Töne anhand einer Frequenz für eine angegebene Zeit auszugeben. Diese Töne sind aber nur einzeln abspielbar. Das liegt daran, dass der NXT den vorherigen Ton abbricht, wenn ein neuer abgespielt werden soll.

A. Sensoren

Es gibt verschiedene Sensoren, welche sich über den NXT steuern lassen und welche verschiedene Werte zurückgeben. Für den Roboter, welcher lediglich eine Musikspur scannen soll, gibt es nur zwei sinnvolle Sensoren:

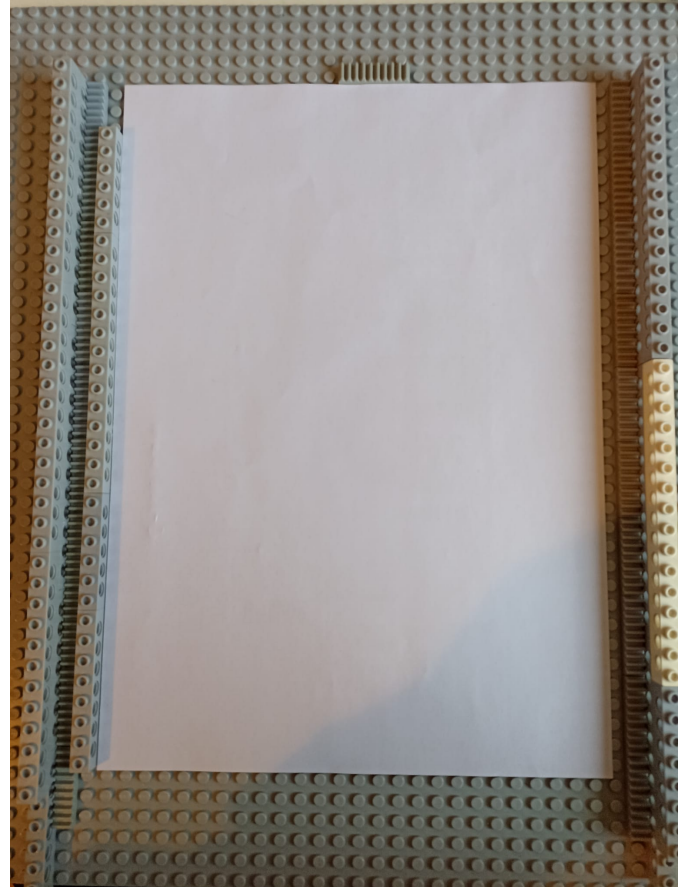


Abbildung 1. Schiene mit Zähnen für Zahnräder und Blatt

1) *Lichtsensor*: Der Lichtsensor scannt die Helligkeit einer Oberfläche, indem dieser die Oberfläche beleuchtet und das reflektierte Licht misst. Er gibt einen Wert zwischen 700 für helle und null für dunkle Oberflächen an. Damit lassen sich ideal schwarze und weiße¹ Oberflächen unterscheiden.

2) *Farbsensor*: Der Farbsensor scannt die Farbe einer Oberfläche und gibt einen RGB-Wert zurück, welcher möglichst die Farbe der Oberfläche darstellt. Damit lassen sich theoretisch mehr Informationen auslesen, da man statt Schwarz und Weiß auch zwischen verschiedenen Farben unterscheiden kann.

¹Natürlich gibt der Sensor auch die Möglichkeit her, verschiedene Grautöne zu unterscheiden.

B. Motoren und Bewegung

Da der Roboter die Musikspur zeitgleich abspielen soll, muss er sich nur gerade über die Musikspur bewegen. Er muss sich nicht drehen oder irgendwie lenken können. Es reicht also ein Motor und eine Antriebsachse. Da der Roboter gerade fahren muss, wird dieser an je einer Zahnstange am Rand mit Zahnrädern angetrieben, wie in Abbildung 1 zu sehen ist. Das Blatt mit der Musikspur liegt zwischen den Schienen.

C. Kodierung

Da nur ein Ton gleichzeitig abgespielt werden kann, kann man alle Sensoren für einen einzelnen Ton verwenden. So lassen sich aus vier Signalen, welche aus schwarz (für eine eins) oder weiß (für eine null) bestehen, 16 Signale darstellen. Mit einer Pause sind das also 15 Noten, die mit den Lichtsensoren einfach realisierbar sind.

D. Programmierung

Die Ansteuerung des NXT und der Sensoren/Motoren erfolgt über MATLAB mithilfe der RWTH Mindstorms NXT Toolbox [1]. So lassen sich die Werte der Sensoren auslesen und die Motoren ansteuern. Außerdem sind die Töne auf dem NXT über MATLAB abspielbar und die Frequenz dort ausrechenbar.

III. ROBOTER, PROGRAMMIERUNG UND KODIERUNG

A. Roboter

Der Roboter besteht aus einer Antriebsachse, welche ihn über das Blatt bewegen soll. An dieser Antriebsstange sind Zahnräder angebracht, welche auf der Zahnstange liegen. Je größer die Zahnung dieser Zahnräder, desto schneller bewegt sich der Roboter und desto instabiler ist die Achse. Da die Achse höher liegt und die Zahnräder am Rand sind, das Gewicht des Roboters aber in der Mitte liegt, wird die Achse gebogen. Außerdem wird die zusätzliche Höhe die Lesegenauigkeit senken, da die Sensoren weiter vom Blatt angeordnet und leicht angewinkelt sind. Wenn man eine zu kleine Zahnung wählt, liegen die Zahnräder eventuell nicht auf der Achse, da der Roboter zeitgleich auf den Sensoren gestützt ist. Zahnräder mit 24 Zähnen geben eine ideale Höhe, um die Sensoren so nah wie möglich am Boden zu haben.

Die Achse führt durch den Motor, welcher diese antreibt. Zusätzlich sind zwei (lockere) Räder mittig an der Achse angebracht, um diese zusätzlich zu stützen. Der Motor ist anhand zweier langer Stangen mit den vier Sensoren verbunden. Es gibt eine weitere Stange, welche den Motor mit dem NXT verbindet und an dem ein weiteres Stützelement für die Achse ist. Desweiteren sind je zwei Sensoren auf einer Seite mit einer kleineren Stange auch am NXT befestigt. An der Befestigung befinden sich je ein großes, loses Rad, um den NXT in der Luft zu halten.

Der NXT muss mit dem USB-Anschluss nach oben befestigt sein, da das Anschlussstück für den USB-Anschluss am längsten ist und nicht zwischen NXT und Boden passt. An der Oberseite mit dem USB-Anschluss ist auch der Motor angeschlossen. Die Sensoren führen über den NXT zum Boden, wo diese angeschlossen sind.

Eine Seitenansicht mit allen wesentlichen Elementen ist in Abbildung 2 zu sehen.

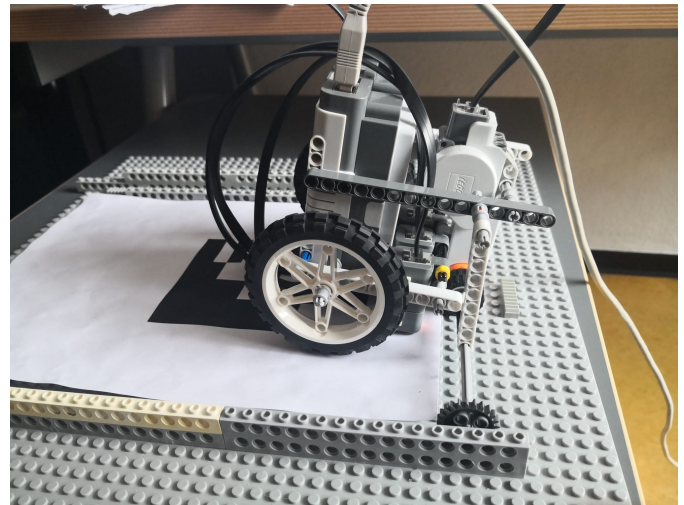


Abbildung 2. Seitenansicht vom Roboter

B. Programmierung

Der Roboter fährt langsam über das Blatt. Währenddessen misst er anhand der vier Sensoren die Helligkeit. Dabei vergleicht er die Sensorwerte jeweils mit einem bestimmten Schwellwert, welcher entscheidet ob der gelesene Wert nun als 'schwarz' oder 'weiß' interpretiert werden soll. Im Programm wird statt 'schwarz' eine Eins und statt 'weiß' eine Null geschrieben. Die vier Werte werden zu einer Dualzahl zusammengefasst und anschließend als Dezimalzahl interpretiert. So ergeben sich die Werte null (also 0000, vier mal weiß) bis 15 (1111, vier mal schwarz). Da nur weiß, also die Null, keinen Ton ergeben soll, wird in dem Fall das Abspielen ausgelassen. Andernfalls wird die jeweilige Zahl anhand des Grundtons und der Tonleiter, welche in das GUI (Abbildung 3) eingegeben werden, in die Frequenz konvertiert. Wie genau das abläuft, wird in Abschnitt III-D beschrieben. Der Ton wird abgespielt und der ganze Prozess 3500 mal wiederholt. Nach der 3500. Wiederholung ist der Roboter am Ende des Blattes. Danach wird der Motor gestoppt.

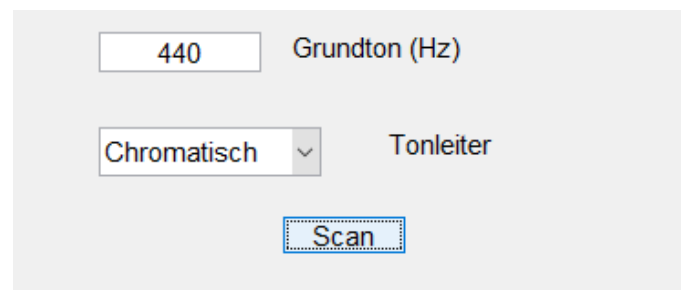


Abbildung 3. GUI zur Einstellung des Grundtons und der Tonleiter

C. Kodierung

Die Kodierung der Musik erfolgt auf DIN A4 Blättern. Der tiefste Ton entspricht dabei einer eins (also 0001). Je nach

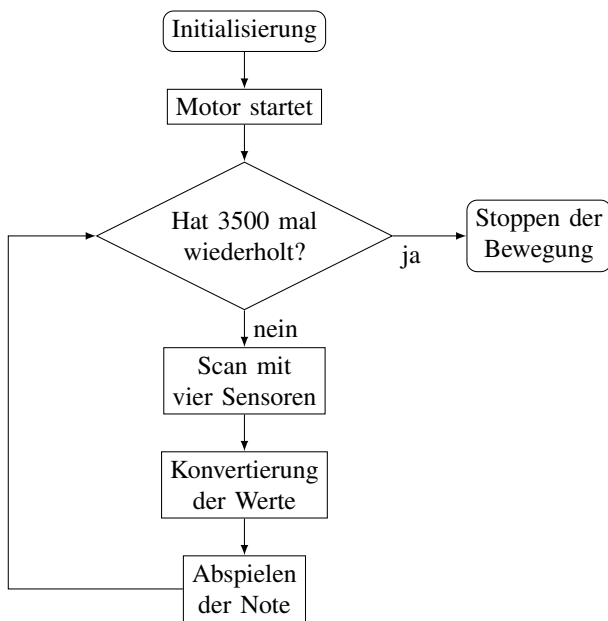


Abbildung 4. Programmablaufplan des Roboters

Tonleiter entspricht ein Ton darüber einer Zahl darüber, also in dem Fall die Zwei. So lassen sich 15 Noten kodieren, also in einer chromatischen Tonleiter etwa eine Oktave, in einer Dur/Moll-Tonleiter etwa zwei Oktaven. Auf dem Blatt sind die vier Bits in vier Spuren zu finden. Eine einzelne Spur ist dabei 23 mm breit, die Randspuren haben allerdings mehr Platz, da neben den Spuren freier Platz ist. Auf dem freien Platz lassen sich außerdem Grundton, Name und Tonleiter (bzw. Kodierung) vermerken. Die Spur rechts entspricht dem niedrigstwertigstem Bit und die linke Spur dem höchstwertigstem.

Die Längen lassen sich durch die Längen der Signale auf dem Blatt darstellen. Da der Roboter gleichmäßig fährt, entspricht die doppelte Signallänge auch der doppelten Notennlänge. Da die Note durch die vier Spuren entsteht, ist es wichtig, dass die Kanten an den Notengrenzen scharf sind, da sonst zwischen den Tönen ein neuer Ton entsteht. So kann bspw. zwischen Ton neun (1001) und Ton sechs (0110) der Ton 15 (1111) entstehen da sich die Töne überlappen, wenn nicht genau genug gezeichnet wurde. Es bietet sich an, ein Zeichenprogramm und einen Drucker zu benutzen, um die Kanten so scharf wie möglich darzustellen.

Ein weiteres Problem ergibt sich, wenn sich ein Ton wiederholt. Solange sich Töne unterscheiden, lassen sich neue Töne gut hören. Sobald aber zwei gleiche Töne nacheinander kommen, ist die Kodierung die gleiche, wie für einen längeren Ton. Aufgrund dessen muss eine kleine Pause zwischen den Tönen gelassen werden, damit sich der neue Ton hören lässt.

D. Musiktheorie

Die Frequenz verhält sich bei den Tönen logarithmisch, also sind Töne durch ein Frequenzverhältnis bestimmt. Eine Oktave zwischen zwei Tönen beschreibt ein Frequenzverhältnis von eins zu zwei. Dementsprechend sind zwei Oktaven

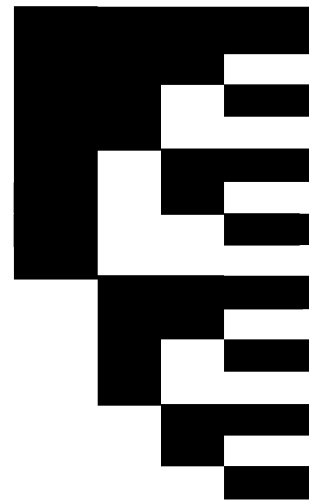


Abbildung 5. Zahlen von 1 bis 15 kodiert

durch ein Frequenzverhältnis von eins zu vier beschrieben. Eine Oktave besteht nach westlicher Musiktheorie aus zwölf Tönen (die sogenannte chromatische Tonleiter), wobei der 13. Ton wieder der erste Ton um eine Oktave erhöht ist.

Nach gleichstufiger Stimmung sind die Frequenzverhältnisse zwischen allen Tönen gleich groß. Wenn man den ersten Ton als „Grundton“ festsetzt, kann man die anderen Töne durch das Frequenzverhältnis beschreiben. Bspw. ist der erste Ton mit einem Verhältnis von eins zu eins oder der 13. (also eine Oktave) mit einem Verhältnis von eins zu zwei gekennzeichnet. Da alle Töne dazwischen dasselbe Frequenzverhältnis haben, kann das Frequenzverhältnis von einem einzelnen Tonschritt (In der Musik wird dieser ein Halbtonschritt genannt) ausgerechnet werden. So muss eine Oktave zwölf Halbtonschritte über dem Grundton sein, also das Frequenzverhältnis x zwölf mal beinhalten:

$$2 = x^{12} \Rightarrow x = \sqrt[12]{2}$$

Die Frequenz s des n -ten Tons kann also mit

$$s = g \cdot \sqrt[12]{2}^n$$

ausgerechnet werden, wenn g die Frequenz des Grundtons ist. Die Ton-Zahl der Kodierung kann also verwendet werden, um damit die Frequenzen in einer chromatischen Tonleiter auszurechnen.

Die Dur- und Moll-Tonleitern setzen sich anders zusammen. Sie bestehen größtenteils aus Ganztönen (welche sich aus zwei Halbtonschritten zusammensetzen).

1) *Die Dur-Tonleiter:* Ausgehend vom Grundton sind alle Tonschritte Ganztonschritte mit wenigen Ausnahmen: Zwischen dem dritten und vierten sowie zwischen dem siebten und achten Ton sind statt des Ganztonschritts ein Halbtonschritt zu finden. Der achte Ton entspricht dann einer Oktave, von da geht die Tonleiter von vorne los.

2) *Die Moll-Tonleiter:* Ähnlich wie die Dur-Tonleiter sind alle Tonschritte Ganztonschritte mit den Ausnahmen zwischen Ton zwei und drei sowie zwischen Ton fünf und sechs. Genauso wie bei der Dur-Tonleiter ist der achte Ton die Oktave, von welchem die Tonleiter wieder beginnt.

Die Implementation läuft relativ einfach: Um auf die Ganztonschritte zu kommen, muss der Wert mit zwei multipliziert werden (Also Ton eins wird zu Ton zwei, Ton zwei zu Ton vier usw.). So wird jeder Halbtonschritt zu einem Ganztonschritt. Da aber der Grundton der selbe bleiben soll, muss der Wert noch um eins gesenkt werden. An den entsprechenden Ausnahmestellen muss jeweils wieder der Ton (wenn er über einem bestimmten Wert ist) um eins gesenkt werden, um die Ganztöne wieder zu Halbtönen machen. Daraus ergeben sich dann die Exponenten für die Tonleitern Dur und Moll.

Da die Töne der Dur- und Moll-Tonleitern auch in der chromatischen Tonleiter vorhanden sind, dient die Implementation lediglich dazu, den Tonumfang erhöhen zu können. Während eine Oktave bei der chromatischen zwölf Töne enthält, enthält sie bei Dur und Moll nur sieben Töne. So ist der Tonumfang doppelt so groß wie bei der chromatischen Tonleiter. Weitere Informationen zur gleichstufigen Stimmung finden sich bei [2]

IV. PROBLEME UND GRENZEN

Im Endeffekt sind trotz aller Anpassung wesentliche Grenzen für den Roboter zu setzen: Die Lautstärke vom NXT ist natürlich begrenzt und lässt sich nicht einstellen. Die Entfernung muss also sehr kurz sein, um etwas verstehen zu können. Genauso darf die Distanz des Roboters zum Rechner nicht zu hoch sein, weil dieser schließlich mit dem Kabel angeschlossen ist. Der Roboter selbst zeigt auch viele Probleme: Zwar ist die Geschwindigkeit des Roboters einstellbar, aber keineswegs so genau, wie die Geschwindigkeit in der Musik. Auch die Kodierung stellt hier eine Grenze auf: je nach Wahl der Tonlänge ist man entweder gezwungen, sich auf Viertelnoten als kürzeste Note zu beschränken (Die Lesegenauigkeit der Sensoren schränkt die Kürze der Noten ein), oder aber man wählt die Noten generell größer. Das wiederum hat zur Folge, das weniger auf das Blatt passt. Die begrenzte Länge des Blattes ist auch ein Problem: Es lassen sich nur wenige Takte abspielen, auch weil nur eine einzelne Tonspur auf dem Blatt abgebildet ist.

Die nächste Komplikation ist bei den Tonläuten: Dadurch, dass nur ein Ton gleichzeitig abspielbar ist, lassen sich keine Akkorde spielen. Da der Roboter keinerlei Nebentöne erzeugt, klingt der Ton auch sehr blechern. Diese Eigenschaft macht den Ton nur beschränkt einsetzbar (Am nächsten wären vermutlich Bitpop oder Chiptune). Auch der Tonumfang ist, wie in III-D beschrieben, sehr begrenzt: Es lassen sich nur 15 verschiedene Töne abspielen. Dabei muss ein Ton der Grundton sein, und nach der Implementierung muss dieser Ton gleichzeitig der tiefste Ton sein (oder tiefer, aber dann müssen weitere Töne eingespart werden). Das ist natürlich bei einem Musikstück selten der Fall, weshalb auch hier einige Abstriche gemacht werden müssen. Der letzte Punkt ist, dass der NXT selbst auch eine interne Grenze für die Frequenz hat. Diese muss zwischen 200 und 14000 Hz liegen, nach oben ist das mehr als genug,

ein gewöhnlicher Flügel kommt bis ca. 4000 Hz, aber nach unten sind 200 Hz eine folgeschwere Grenze: der niedrigste Ton ist ein gis mit ca. 208 Hz, welcher am oberen Ende des Bass-Schlüssel liegt. Ein Klavier kommt zu ca. 28 Hz [3].

V. FAZIT UND AUSSICHT

Trotz seiner Grenzen ist der Roboter trotzdem dazu in der Lage, erkennbare, simple Melodien zu spielen, so sind Kinderlieder oder Videospelmusik gut zu erkennen und abspielbar. Außerdem ist die eigentliche Funktion, eine kodierte Musikspur zu erkennen und abzuspielen erfüllt. Der Roboter fährt über eine Spur, scannt diese Spur, dekodiert die kodierten Noten, rechnet die Frequenz aus und spielt diese mit dem NXT ab. Der Roboter selber ist natürlich verbesserbar und nicht perfekt, aber im Rahmen des LEGO-Seminars sind seine Leistungen genügend und er erfüllt alle Vorstellungen. In Abbildung 6 sieht man ein Beispiel anhand der Tetris-Melodie.

Verbesserungen sind natürlich möglich: Die Farbsensoren oder Grautöne sind zur Erweiterung des Tonumfangs verbesserbar. Die Fesslung an eine Tonleiter ist angenehmer durch Implementation machbar oder aber komplett aufhebbar durch das Eingeben der einzelnen Töne für jeden Tonwert.

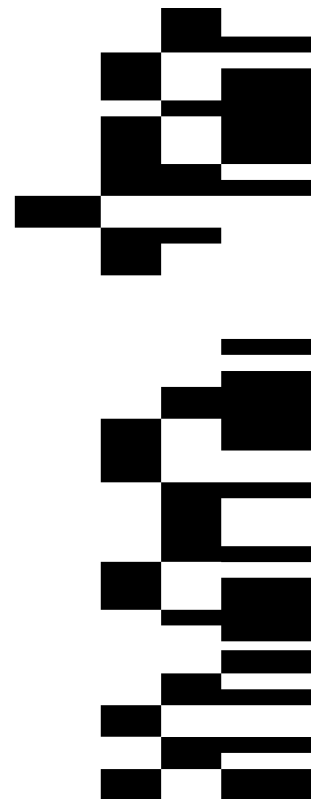


Abbildung 6. Korobeiniki kodiert in Moll, mit dem Grundton c'' (523 Hz)

LITERATURVERZEICHNIS

- [1] BEHRENS, Alexander: *RWTH - Mindstorms NXT Toolbox*. <https://www.mathworks.com/matlabcentral/fileexchange/18646-rwth-mindstorms-nxt-toolbox>. Version: 2022
- [2] SUITS, B. H.: *Scales: Just vs Equal Temperament*. <https://pages.mtu.edu/~suits/scales.html>
- [3] HAMMES, Ireen: *Wie viele Tasten hat ein Klavier?* <https://www.klavier-hammes.de/faq/klavier-tasten-anzahl>

Melody, der Musikbot

Maximilian Weitzel, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Das Projektseminar Elektro- und Informationstechnik wird jedes Jahr an der Otto-von-Guericke Universität Magdeburg durchgeführt. Im Rahmen des diesjährigen Projektseminars wurde ein Roboter erstellt. Dieser kann codierte Musik abspielen. Die Entwicklung und Konstruktion dieses Roboters erfolgten auf Basis von LEGO-Mindstorms-Sets sowie des LEGO-NXT-Steuerungscomputers. Die softwareseitige Implementierung wurde über MATLAB realisiert. In diesem Paper werden der Aufbau und die Funktionsweise des Roboters vorgestellt. Es wird weiterhin auf einige Herausforderungen während des Konstruktionsprozesses, sowie auf deren Lösungsansätze eingegangen. Außerdem werden einige theoretische Hintergründe betrachtet, welche im Entwicklungsprozess von Bedeutung waren.

Schlagwörter—LEGO-Mindstorms, MATLAB, Musik, NXT, Roboter, Töne

I. EINLEITUNG

TÖNE umgeben den Menschen in seinem alltäglichen Leben immer und überall. Nüchtern betrachtet ist Musik nicht viel mehr als eine Aneinanderreihung von Tönen. Dennoch ist Musik wahrscheinlich eines der schönsten Dinge, die ein Mensch mit seinem Ohr erleben kann. Aus diesem Grund beschäftigt sich die Menschheit bereits seit der Antike mit der Struktur von Tönen. Seitdem entstanden unzählige Systeme. Zum einen um Töne zu sortieren und zum anderen, um es leichter zu machen, die Töne zu gewünschter Musik zu kombinieren. In der Vergangenheit konnten bereits viele Fortschritte und Veränderungen in der Musiktheorie verzeichnet werden. Auf die theoretischen Zusammenhänge soll jedoch innerhalb dieses Papers nur am Rand eingegangen werden. Wesentlich entscheidender ist die Frage, wie man diese theoretischen Zusammenhänge nutzbar machen kann, um Musik zu erzeugen. Für den im nachfolgenden vorgestellten Roboter werden einige einzelne Aspekte der Musik- beziehungsweise Klangtheorie benötigt, zum einen die Art und Weise, wie ein Ton erzeugt wird, zum anderen, wie Töne überhaupt zu einer kohärenten Melodie kombiniert werden können. Damit verbunden ist die Fähigkeit, Musik auf einen Tonträger zu speichern und durch eine Vorrichtung wieder abspielbar zu machen.

Dies war das Kernziel bei der Entwicklung von Melody. Es existieren verschiedenste Methoden, um musikalische Klangerzeugnisse abzuspeichern und wieder abspielbar zu machen. Mittels des zur Verfügung gestellten NXT-Steins lassen sich Töne abspielen und über externe Sensoren Daten einlesen. Die Zielstellung lautete deshalb, die Möglichkeiten des NXT zur Klangerzeugung und zur Auswertung von Sensordaten nutzbar zu machen, um mittels eines Auswertungsalgorithmus eine im Voraus codierte Melodie abspielbar zu machen.

DOI: 10.24352/UB.OVGU-2022-017

Lizenz: CC BY-SA 4.0



Abbildung 1. Compact Disc (CD) als bekanntestes Beispiel eines optischen Datenspeichers [1]

II. VORBETRACHTUNGEN

Es werden zuerst einige Grundzüge aus der Musiktheorie vorgestellt, welche für die Entwicklung von Melody benötigt wurden. Anschließend wird kurz auf optische Datenträger, insbesondere am Beispiel einer Compact Disc (CD) eingegangen. Da diese für den Entwurf des Datenträgers wichtig waren. Danach wird mit dem eigenen Ansatz auf den eigentlichen Entwicklungsprozess übergeleitet.

A. Musiktheorie

Historisch haben sich im europäischen Raum bestimmte Tonintervalle entwickelt, bei denen die entsprechenden Frequenzen zu den Tönen in einem ganzzahligen Verhältnis zueinander stehen. Beispielhaft zu nennen sind hier die Oktave mit einem Verhältnis von 2:1, also einer Verdopplung der Frequenz pro Oktave. Nach dem gleichen Prinzip können zum Beispiel die Quinte (3:2) und die Quarte (4:3) festgelegt werden.

Um zu einem Grundton nun zum Beispiel eine Quinte zu bestimmen, muss der entsprechende Grundton mit $\frac{3}{2}$ multipliziert werden. Würde man einen Grundton mit $\frac{1}{2}$ multiplizieren, dann würde das einen Sprung um eine Oktave nach unten bedeuten. Die Zielstellung ist jetzt ein Verhältnis zwischen Quintensprüngen nach oben und Oktavensprüngen nach unten zu finden, um wieder zu einem Grundton zu gelangen. Dadurch soll bestimmt werden, in wie viele Teile ein Tonintervall aufgeteilt werden muss. Das bedeutet, wie viele Quinten- und Oktavensprünge zu einer Tonleiter zusammengefügt werden können.

Die mathematische Bestimmung erfolgt über Kettenbruchentwicklung, wird jedoch an dieser Stelle ausgelassen. Es sei auf die Quelle [2] verwiesen, wo detaillierte und umfangreiche Ergänzungen zu diesem Thema zu finden sind. Letztendlich ergibt sich eine Einteilung in 12 Oktavenschritte. Da in einer Tonleiter ein gleiches Frequenzverhältnis zwischen aufeinander folgenden Tönen gelten soll, ergibt sich ein konstantes

Verhältnis k zwischen zwei Tönen $\frac{T_1}{T_2} = 2^{\frac{1}{12}} = k$. Dieses Verhältnis wurde im Auswertungsalgorithmus verwendet [3].

B. Optische Datenspeicher/Compact Disc

Anfang der 1970er Jahre wurden erstmals Speicher verwendet, welche das Prinzip der optischen Abtastung nutzten, um große Mengen an Informationen auf einem Speichermedium abzurufen und zu speichern. Die optische Abtastung wird häufig durch einen Laser realisiert. Daher stammt auch der Name optischer Datenspeicher, da das Auslesen oder Schreiben der Daten nicht wie bei anderen Speichermedien beispielsweise durch einen magnetischen Schreib- beziehungsweise Lesekopf erfolgt [4].

Eines der wohl bekanntesten Beispiele für einen optischen Datenspeicher ist die 1982 eingeführte Compact Disc oder kurz CD. Die CD ist eine Kunststoffscheibe, auf welcher eine dünne Metallschicht sowie ein Schutzlack aufgebracht sind. In Abbildung 1 ist eine CD dargestellt. Gut zu erkennen sind hier die charakteristischen Reflexionserscheinungen, verursacht durch die Struktur einer CD. Sie besitzt einen Durchmesser von 8 cm bis 12 cm. Die gespeicherten Informationen auf einer CD befinden sich auf einer spiralförmig auf nahezu dem Großteil der Fläche einer CD angeordneten Struktur. Diese Struktur reflektiert aufgrund ihrer Oberflächenbeschaffenheit das Licht an verschiedenen Stellen unterschiedlich. Diese Schwankungen in der Helligkeit werden von einer Lesevorrichtung registriert und die dadurch codierten Informationen werden durch geeignete Protokolle decodiert [5].

C. Eigener Ansatz

Im zur Verfügung gestellten LEGO-Mindstorm-Set waren verschiedene Arten von Sensoren enthalten. Es standen folgende Sensoren zur Verfügung, Lichtsensoren zur Messung von Helligkeitsschwankungen, (RGB-)Farbsensoren zum Erkennen von Farbinformationen, Ultraschallsensoren zur Abstandsmessung sowie Tastsensoren, welche auf ein physisches Tastsignal reagieren. Es standen mehrere Motoren zur Verfügung, von welchem aber lediglich einer genutzt wurde. Am NXT-Stein selbst können vier Sensoren sowie drei Motoren angeschlossen werden. Der NXT-Stein besitzt die Möglichkeit zur Wiedergabe von Tönen, welche vorher in einer codierten Form eingelesen werden. Die Decodierung beziehungsweise Steuerung der Funktionen des NXT-Steins erfolgte im Rahmen dieses Praktikums über MATLAB. Als Schnittstelle diente hier eine öffentliche Toolbox der RWTH Aachen [6].

Durch die in den vorherigen Punkten erwähnten Aspekte der Musiktheorie sowie des bekannten Konzepts eines optischen Datenspeichers wurde die Zielstellung abgeleitet. Diese lautete, mittels der enthaltenen Lichtsensoren eine Vorrichtung zu konstruieren, welche codierte Lieder über Helligkeitsdifferenzen ausliest, anschließend per MATLAB decodiert und einen Ton ausgibt.

III. UMSETZUNG

In diesem Abschnitt werden zuerst der Aufbau der Lesevorrichtung und des Roboters allgemein erklärt. Anschließend wird

der Datenträger, welcher das codierte Lied enthält, in Aufbau und Funktionsweise vorgestellt. Es wird weiterhin betrachtet, wie der Roboter den Lesevorgang ausführt und daraus Töne abspielt.

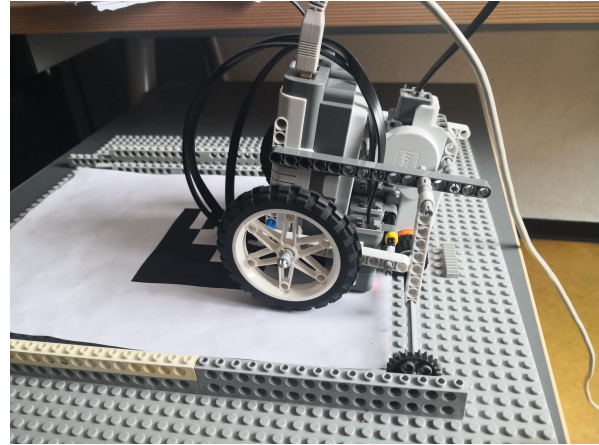


Abbildung 2. Roboter in Betriebsbereitschaft

A. Aufbau

Das zentrale Element bildet die Leseeinheit. Diese besteht aus dem NXT-Stein sowie vier Lichtsensoren. Es wurden vier Lichtsensoren verwendet, damit vier Spuren parallel eingelesen werden können. Das bedeutet, dass die Sensoren mit minimalem Abstand nebeneinander angeordnet sind. Jeder der Sensoren nimmt im jeweiligen Messbereich die Helligkeitsdifferenz auf. Bei der Konstruktion dieser Anordnung wurde abgewägt zwischen möglichen Interferenzen der einzelnen Sensoren untereinander und maximal möglicher Menge an aufzunehmenden Informationen. Der minimale Abstand der Sensoren, sowie die Verwendung der maximalen Anzahl an Sensoren sollte gewährleisten, dass eine größtmögliche Menge an Informationen auf dem Datenträger gespeichert und von der Leseeinheit ausgelesen werden können. Der Roboter in Betriebsbereitschaft ist unter Abbildung 2 dargestellt.

Die Leseeinheit ist senkrecht über dem Datenträger angeordnet. Die Sensoren sind auf den Datenträger gerichtet, um die einzelnen Spuren separat einlesen zu können. Während des Lesevorgangs wird die Leseeinheit langsam über den Datenträger bewegt. Die Geschwindigkeit der Leseeinheit muss mit der Anzahl an Werten koordiniert werden, die durch die Lichtsensoren aufgenommen werden. Für mögliche Anpassungen stellt das einen Nachteil dar, denn beide Werte müssen aufeinander mittels Versuch-und-Irrtum abgestimmt werden. In seiner aktuellen Konfiguration ist die Wahl dieser Werte ein Kompromiss aus maximal möglichen Messwerten und notwendiger Geschwindigkeit. Es ist allerdings auch möglich, andere Kombinationen zu wählen.

Die Bewegung der Leseeinheit wird über einen einzelnen Motor realisiert, der direkt an der Leseeinheit angebracht ist. Der Motor ist über eine Antriebsachse mit beidseitig am Rand platzierten Zahnschienen verbunden. Diese Zahnschienen ermöglichen eine Vorwärtsbewegung. Eine Herausforderung

bei der Realisierung der Bewegung stellte die Fahrstabilität der Leseinheit dar. Die Zahnräder der Leseinheit neigten dazu, die Spur der Zahnschienen zu verlassen oder sich schräg darüber zu bewegen, was im Extremfall zu einer Art von „Entgleisungseffekt“ beziehungsweise zu unsauber eingelesenen Sensorwerten führte. Eine Lösung hierfür bot die Konstruktion einer Führungsschiene. Auch hier wurde sich bewusst für eine simple Konstruktion entschieden, um mögliche neu entstehende Fehlerquellen ausschließen zu können. Die Führungsschiene besteht aus zu beiden Seiten angebrachten Wänden, welche geringfügig höher als die Zahnschienen selbst sind. Durch die geringe Höhe auf der Seite der Antriebsachse wird gewährleistet, dass sich die Leseinheit ungestört bewegen kann.

B. Datenträger

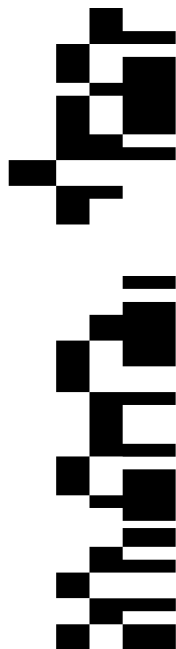


Abbildung 3. Beispiel des Layouts eines Datenträgers

Der Datenträger entspricht in seiner Größe einem Blatt nach DIN A4. Das ermöglicht maximal vier Spuren auf dem Datenträger einzuzeichnen. Eine Abbildung des Datenträgers findet sich in Abbildung 3. Diesen Spuren ist jeweils ein Bitwert vom niederwertigsten zum höchstwertigen Bit zugeordnet. Diese Zuordnung ist direkt über den MATLAB-Code implementiert, indem je einer Spur ein Array an Messwerten zugewiesen wird. Diese entsprechenden Arrays werden dann mit der jeweiligen Bit-Wertigkeit multipliziert, um eine Bit-Codierung zu ermöglichen. In Abbildung 4 ist der entsprechende Code zu diesem Verfahren abgebildet.

Um ein Lied zu codieren, werden zuerst alle Töne des gewünschten Lieds in binäre Form umgewandelt. Nach diesem Schritt wird ein entsprechender Datenträger in einem Zeichenprogramm erstellt und anschließend ausgedruckt. Dieses Vorgehen besitzt einige signifikante Vorteile gegenüber anderen

```
for i = 1:size
messwerte(1, i) = GetLight(SENSOR_1);
messwerte(2, i) = GetLight(SENSOR_2);
messwerte(3, i) = GetLight(SENSOR_3);
messwerte(4, i) = GetLight(SENSOR_4);
messwerte(:, i) = 1 - (messwerte(:, i) > scan);
noten(i) = messwerte(1, i) + messwerte(2, i)*2 + messwerte(3, i)*4 + messwerte(4, i)*8;
```

Abbildung 4. Code-Ausschnitt in welchem die Bit-Codierung umgesetzt wurde

Verfahren, wie zum Beispiel dem Einzeichnen per Hand. Zum einen kann dadurch schneller gearbeitet werden. Zum anderen sind bei sachgemäßer Verwendung Kanten schärfer und die gesamte Qualität des Datenträgers höher, was zu einer verbesserten Lesegenauigkeit und verringerten Störgeräuschen führt. Außerdem können mögliche Fehler beim Codieren und Erstellen schneller korrigiert werden.

Durch die Lichtsensoren werden Helligkeitsdifferenzen aufgenommen. Für den Datenträger wurde deshalb eine Schwarz-Weiß-Codierung gewählt. Diese ermöglicht, ein Lied in Bit-Codierung auf dem Datenträger darzustellen. Die beiden Zustände können dadurch simuliert werden. Nehmen die Sensoren während der Laufzeit einen Wert auf, gibt der NXT entsprechend einen Ton aus. Um eine Melodie zu erzeugen, werden jedoch neben durchgängigen Tönen auch Pausen benötigt. Diese Pausen sind durch weiße Striche direkt auf dem Datenträger markiert und werden von den Lichtsensoren entsprechend erkannt. Mögliche Softwarelösungen wurden aus Zeit- und Komplexitätsgründen nicht weiter verfolgt.

Auf dem Datenträger können, in seiner aktuellen Form, bis zu 15 Töne codiert und gelesen werden. Das resultiert aus der Größe eines DIN-A4 Blatts und der minimalen Strichbreite, welche gewählt werden musste, um ein korrektes Lesen der codierten Töne zu gewährleisten. Die Eigenschaften Blattgröße und Strichbreite, im Zusammenhang mit der Lesegenauigkeit der Sensoren, können in diesem Fall als limitierende Faktoren betrachtet werden.

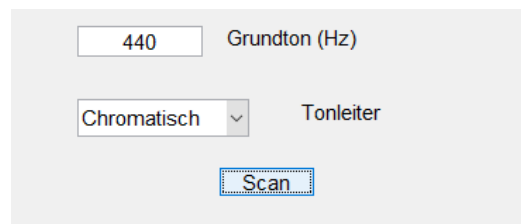


Abbildung 5. GUI zur Auswahl der Grundeinstellungen

C. Lesevorgang/Abspielen eines Tons

Bevor der Lesevorgang gestartet wird, müssen über ein Graphical User Interface (GUI) ein Grundton und eine Tonart eingestellt werden. Das Layout der GUI ist unter Abbildung 5 dargestellt. Für den Grundton kann eine beliebige Frequenz eingegeben werden. Die Tonarten Dur, Moll und Chromatisch werden durch ein Auswahlfenster gewählt. Nachdem diese Grundeinstellungen bestätigt wurden, wird der Lesevorgang gestartet.

Während des Lesevorgangs bewegt sich die Leseinheit langsam

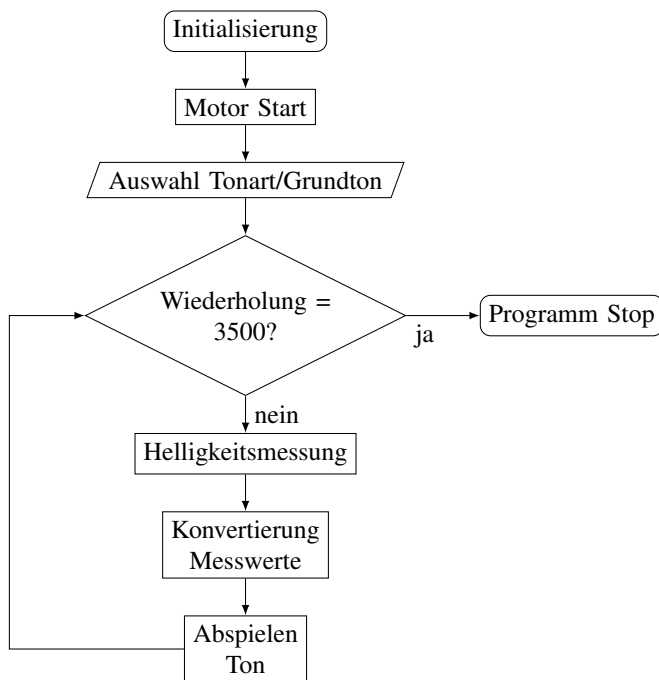


Abbildung 6. Programmablaufplan

über den Datenträger. Die vier nebeneinander angeordneten Sensoren nehmen jeweils in ihrer Spur die Helligkeitsdifferenz auf. Während der Laufzeit werden die gemessenen Werte in MATLAB in abspielbare Töne konvertiert und der Befehl zum Abspielen des entsprechenden Tons an den NXT geschickt. Die Messung wird so lange fortgeführt, bis die festgelegte Anzahl an Messwerten erreicht ist. Während der Erprobung wurde für die Anzahl der Messwerte pro Durchgang auf 3500 festgelegt. Der Programmablaufplan für den Lesevorgang befindet sich in Abbildung 6.

IV. ERGEBNISDISKUSSION

Die grundlegende Zielstellung lautete, einen Roboter zu konstruieren, welcher auf einem Datenträger codierte Töne auslesen kann und so eine Melodie abspielt. Diese Zielstellung konnte erfüllt werden. Es konnten verschiedene Lieder beziehungsweise Tonleitern codiert werden. Beispielhaft zu nennen sind hier „Twinkle twinkle little star“ oder „Korobeiniki“, besser bekannt als Titelmelodie zum Spiel „Tetris“. Es konnte eine vergleichsweise schnelle Methode zum Codieren und Auslesen von Musik entwickelt werden, mit der es möglich ist, einfache Melodien auf einem Datenträger zu speichern und wieder abzuspielen. Die Methode zum Einlesen der Musik konnte im Laufe der Projektarbeit verbessert werden. Durch den Einbau eines Filters und die Verbesserung des Erstellungsprozesses eines Datenträgers war es möglich, Störgeräusche zu minimieren und ein weitgehend fehlerfreies Lesen zu ermöglichen. Nach verschiedenen Versuchen konnte auch eine möglichst optimale Abstimmung zwischen Laufgeschwindigkeit des Motors und Menge an Sensordaten gefunden werden.

Es existieren Bereiche mit Optimierungspotenzial. Die Führung in ihrer aktuellen Form verhindert zwar, dass der Roboter völlig

die Spur verlässt. Jedoch kann es noch immer zu Abweichungen innerhalb der Führung kommen, was kleinere Lesefehler verursacht. Außerdem konnten der Platz, den die Führung einnimmt und der Platz, welcher für den Datenträger benötigt wird, nicht ideal aufeinander abgestimmt werden. Ebenso konnte der Erstellungsprozess eines Datenträgers signifikant verbessert werden. Dennoch gibt es auch hier Verbesserungspotenzial, denn der manuelle Erstellungsprozess per Zeichenprogramm stellt zwar eine Verbesserung dar, trotzdem sind mögliche Fehler durch unsachgemäße Handhabung nicht auszuschließen. Bedingt durch das Ende des Projektseminars wurden verschiedene Lösungsansätze nicht oder nur ansatzweise verfolgt werden. Verschiedene Bereiche mit Verbesserungspotenzial sind zum Beispiel, eine Schienenführung der Leseinheit, ein Filter für mögliche Störgeräusche oder eine Erhöhung der limitierten Speicherkapazität des Datenträgers.

V. ZUSAMMENFASSUNG UND FAZIT

Zusammenfassend lässt sich feststellen, dass die gruppeninterne Zielstellung eines Roboters, welcher codierte Melodien abspielbar machen kann, zufriedenstellend erreicht wurde. Es wurde ein Roboter konstruiert, welcher sich über einen Datenträger bewegen kann und welchem es möglich ist während der Laufzeit codierte Töne auszulesen und abzuspielen. Es konnten verschiedene Probleme bezüglich Hard- und Software gelöst werden. Insgesamt war es möglich, beliebig viele einfache Melodien zu codieren und abzuspielen.

Mögliche Verbesserungsansätze bezogen auf die im vorherigen Abschnitt genannten Bereiche stellen sich wie folgt dar. Durch die Verbesserung der Führung könnte der Roboter noch stabiler während des Fahrens werden, was eine verbesserte Tonqualität ermöglichen würde. Ebenfalls zur Verbesserung der Tonqualität beitragen könnte eine softwareseitige Verbesserung des Filters, welcher aktuell nur teilweise implementiert werden konnte. Auch die vergleichsweise geringe Größe der gesamten Vorrichtung und die damit verbundene Begrenzung der Abmessungen des Datenträgers stellen limitierende Faktoren dar. Man könnte hier zum Beispiel durch Vergrößerung des gesamten Aufbaus oder Implementierung von Farbsensoren verbesserte Ergebnisse erzielen.

LITERATURVERZEICHNIS

- [1] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Bild einer CD*. <https://upload.wikimedia.org/wikipedia/commons/c/ca/CD-ROM.png>. Version: Februar 2022. – abgerufen am : 21.02.2022
- [2] HARTFELDT, Christian ; EID, Wolfram ; HENNING, Herbert: *Mathematik in der Welt der Töne*. <http://www.math.uni-magdeburg.de/reports/2002/musik.pdf>. Version: Februar 2022. – abgerufen am : 20.02.2022
- [3] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Tonstruktur (mathematische Beschreibung)*. [https://de.wikipedia.org/wiki/Tonstruktur_\(mathematische_Beschreibung\)](https://de.wikipedia.org/wiki/Tonstruktur_(mathematische_Beschreibung)). Version: Februar 2022. – abgerufen am : 21.02.2022
- [4] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Optische Datenspeicher*. https://de.wikipedia.org/wiki/Optischer_Datenspeicher. Version: Februar 2022. – abgerufen am : 20.02.2022
- [5] PLANET WISSEN: *Die Compact Disc (CD)*. https://www.planet-wissen.de/kultur/musik/geschichte_der_tontraeger/pwiecompactdisc100.html. Version: Februar 2022. – abgerufen am : 21.02.2022
- [6] BEHRENS, Alexander: *RWTH-Mindstorms NXT Toolbox*. <https://www.mathworks.com/matlabcentral/fileexchange/18646-rwth-mindstorms-nxt-toolbox>. Version: Februar 2022. – abgerufen am : 25.02.2022

Bewässerungsroboter

Matt Müller, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Kurzfassung—Diese Arbeit befasst sich mit der Dokumentation der Entwicklung eines im Laufe des zweiwöchigen LEGO-Mindstorm-Projektseminars entstandenen Bewässerungsroboters. Als Konstruktionsgrundlage dienten zwei LEGO-Baukästen mit zahlreichen Sensoren und einem NXT-Stein, welcher mithilfe von MATLAB programmiert wurde. Es wird die mechanische Realisierung, Funktionsweise sowie der Entwicklungsprozess des abgeschlossenen Modells detailliert dargestellt. Zudem werden die Schwierigkeiten und Herausforderungen erläutert.

Schlagwörter— Bewässerung, LEGO-Mindstorm [1], MATLAB [2], NXT-Stein, Programmierung, Roboter

I. EINLEITUNG

JEDER ist diesem Problem sicherlich schon einmal im Leben begegnet, sei es durch tagelange Abwesenheit von der eigenen Wohnung aufgrund eines Urlaubs, durch fehlende Motivation oder durch schlichtes Vergessen, sich darum zu kümmern. Nach mehreren Tagen ohne Wasser sind die meisten Zimmerpflanzen vertrocknet. Daher war dieses Projektseminar Anlass dagegen Abhilfe zu schaffen und somit einige Pflanzen vor dem Vertrocknen zu retten. In vielen Bereichen des täglichen Lebens wird der Mensch ohnehin schon bei der Arbeit von Robotern unterstützt, wenn sie ihm nicht sogar komplett abgenommen wird. Sei es in der Fabrik bei Fließbandarbeiten, wie in der Automobilindustrie oder beim präzisen Laserschneiden von Metallplatten, oder sei es zu Hause, beim Staubsaugen oder Rasen mähen, die fleißigen Helfer sind nicht mehr wegzudenken. Also warum nicht auch beim Gießen der Zimmerpflanzen?

Auch in diesem Jahr fand das LEGO-Mindstorm-Projektseminar wieder an der Otto-von-Guericke-Universität in Magdeburg statt. Im Februar 2022 wurde den Studierenden der Elektromobilität und Elektrotechnik/Informationstechnik das Programmieren in MATLAB nähergebracht. Dazu wurde jeweils in Zweiergruppen an einem Legoroboter gearbeitet. Pandemiebedingt musste dieses Jahr jedoch alles auf Onlinepräsentationen umgestellt werden, was nicht nur Schlechtes mit sich brachte. Die Studierenden mussten zwar die Entwicklung des Roboters bei sich zu Hause vornehmen und hatte keine Austauschmöglichkeiten von Bauteilen untereinander, aber dafür war es möglich, die wichtigsten Präsentationen für die Nachwelt festzuhalten [3]. Im Rahmen dieser zwei Wochen, welche aus Durcharbeiten der Theorie in der ersten und Fertigstellen des Roboters in der zweiten Woche

bestanden, war es Ziel der Gruppe 2, um Alexander Schwerin und Matt Müller, einen solchen Roboter zu konstruieren, der den Menschen beim Gießen

der Zimmerpflanze unterstützt und somit im besten Fall ein längeres Leben der Pflanze ermöglicht. Als Grundlage des Bewässerungsroboters dienen der dank dem Toolkit der RWTH Aachen [4] programmierbare NXT-Stein, zwei LEGO-Motoren, ein Farb- sowie Soundsensor und zwei Tastsensoren.

II. VORBETRACHTUNGEN

Im Folgenden wird die bereits vorhandene Technik kurz vorgestellt.

A. Stand der Technik

Für den persönlichen Gebrauch im Garten reicht es meist, die Pflanzen ab und an per Rasensprenger, eventuell mithilfe einer Zeitschaltuhr oder direkt aus dem Schlauch per Hand zu bewässern. In modernen Haushalten findet man teilweise auch automatische Bewässerungsanlagen, die sich je nach Einstellung aus dem Boden ausfahren, den Rasen sowie die Pflanzen bewässern und sich nach getaner Arbeit wieder in Boden einfahren. Diese Variante hat den Vorteil, dass keine Wasserschläuche auf dem Boden liegen müssen. An manchen Orten, wie auf Friedhöfen oder auf denkmalgeschützten Grundstücken ist es jedoch nicht möglich, unterirdische Wasserleitungen zu verlegen. Um dennoch die Schönheit nicht durch herumliegende Schläuche zu stören, gibt es Bewässerungsroboter, wie z.B. Rainos [5]. Dieser geländetaugliche Roboter fährt auf dem Friedhof von einem Grab zum anderen (oder eben auf großen Gartengebieten, wie sie in Schlössern zu finden sind, von einer Pflanze zu anderen) und ist in der Lage, jedes Grab nach unterschiedlicher Einstellung zu bewässern. In Anbetracht des immer stärker werdenden Personalmangels in den Bereichen des Gartenbaus [6] und somit auch des Gärtnerberufs, ist ein solcher Bewässerungsroboter eine gute und auch kostengünstige Alternative, um sich weiterhin an der Schönheit der Pflanzen zu erfreuen. Da der Roboter einen großen Wassertank benötigt, um nicht nach jedem Gießvorgang zur Auffüllstation fahren zu müssen und außerdem eine Pumpe im Inneren benötigt, um genügend Wasserdruck aufbringen zu können, um das Wasser ausreichend weit spritzen zu können, ist die Konstruktion des Roboters sehr groß ausgefallen. Dies stellt in den genannten Anwendungsbereichen aber kein Problem dar.

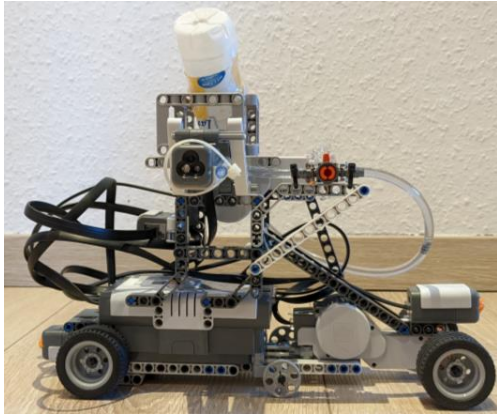


Abbildung 1: Aufbau

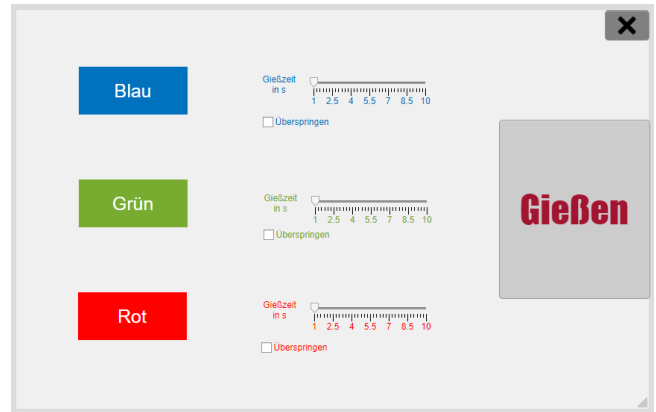


Abbildung 2: Graphical User Interface

B. Idee

Im Rahmen des Praktikums sollte ein solcher Bewässerungsroboter konstruiert sowie programmiert werden. Zur Verfügung standen zwei LEGO-Mindstorm-Baukästen, mit insgesamt einem NXT-Stein, drei Motoren, zwei Tastsensoren, einem Sound-, Farb-, Licht- und Ultraschallsensor sowie zahlreichen Legobausteinen. Diese brachten jedoch einige Einschränkungen mit sich, wenn es darum ging, einen Roboter mit so großem Ausmaß, wie die heutigen zu bauen. Zudem war es nur möglich, vier anstatt der oben genannten sechs Sensoren gleichzeitig an den NXT-Stein anzuschließen. Aufgrund dessen wurde der Anwendungsbereich auf das Fensterbrett umverlegt, da die Konstruktion dort kleiner gehalten werden muss und weniger Pflanzen zu bewässern sind. Nach einem Startsignal soll der Lego-Roboter von der einen zur anderen Seite fahren, wobei er sich die Positionen der sich auf dem Fensterbrett befindenden Pflanzen erkennen und merken kann. Anschließend sollen die erkannten Pflanzen angesteuert und nach Einstellung gegossen werden, denn es soll dem Benutzer mittels einer GUI möglich sein, die Bewässerungszeiten für jede einzelne Pflanze festzulegen. Um dies realisieren zu können, sind ein Antriebsmotor sowie ein Motor zum Öffnen des Ventils nötig. Der Wasserdruck entsteht allein durch die Gravitationskraft.

III. UMSETZUNG

In diesem Abschnitt wird der Aufbau sowie die Funktionsweise des Roboters erläutert

A. Konstruktion

Wie in Abbildung 1 zu sehen ist, wurde sich für ein längliches Design entschieden. Dies ist dem Anwendungsort des Roboters geschuldet, denn er wurde konstruiert, um auf dem Fensterbrett zu fahren. Zudem musste er hoch gebaut werden, damit auch Pflanzen in einem hohen Topf gegossen werden können. Ein weiterer Grund für den Hochbau des Roboters ist der Wassertank, welcher sich ganz oben auf dem Gerüst befindet. Dieser muss hoch genug über dem Schlauchende sitzen, um genügend Wasserdruck für das Gießen im Schlauch zu erzeugen. Um mehrere Pflanzen nacheinander bewässern zu können, wurde der Roboter mittels Rädern mobilisiert. Dies hatte jedoch

zur Folge, dass das ganze Konstrukt sehr instabil wurde, weshalb kleine Stützräder in der Mitte nötig waren. Angetrieben wird der Roboter durch einen Motor an der rechten Achse (Abbildung 1). Dieser ermöglicht es ihm, auf dem Fensterbrett von der einen zur anderen Seite hin und her zu fahren und so die Pflanzen zielgenau anzusteuern. Der zweite verbaute Motor sitzt etwas höher, direkt unter dem Wassertank und dient als Ventilöffner am Schlauch, wodurch der eigentliche Gießvorgang gesteuert wird. Von den zur Verfügung stehenden Sensoren, wurden zwei Tastsensoren sowie einen Farb- und Soundsensor verbaut. Die Tastsensoren befinden sich jeweils am linken und rechten Ende des Roboters (Abbildung 1). Sie dienen als eine Art Orientierungshilfe auf dem Fensterbrett und lassen ihn wissen, wenn eine Wand erreicht ist. Außerdem fährt der Roboter zwischen zwei Schienen, die links und rechts ausgehend von der Fahrtrichtung liegen, um ihn in der Bahn zu halten und einen Absturz vom Fensterbrett zu verhindern. Der Farbsensor befindet sich neben dem Schlauchende am Gießarm des Roboters. Dieser ist für die Erkennung der Pflanzen zuständig. Der Soundsensor liegt zentral auf der linken Seite des Gerüsts (Abbildung 1) und fungiert als Aktivator des Roboters (mehr zu den Sensoren in Abschnitt B.). Der NXT-Stein ist das Herzstück und sitzt unterhalb des Gießarms links neben dem Stützrad (Abbildung 1). Er ist Anlaufpunkt für alle eingehenden MATLAB-Befehle und ermöglicht so erst die Funktion des Roboters. Der NXT-Stein ist per Kabel mit allen Motoren sowie Sensoren verbunden.

B. Sensorik

Wie bereits erwähnt dient der Soundsensor als Aktivator des Programms sowie des Roboters. Der Soundsensor misst die Umgebungslautstärke in Dezibel beziehungsweise dBA. Wird ein bestimmter Wert überschritten, so wird das Programm, einschließlich des Roboters, gestartet. Als Grenze für das Loslegen des Bewässerungsroboters wurden 850 dBA festgelegt. Wird dieser Wert jedoch nur kurzzeitig überschritten, führt das nicht unweigerlich zu einem Start. Um ein willkürliches Losfahren des Roboters, beispielsweise ausgelöst durch das Knallen einer Tür, zu verhindern, wird in

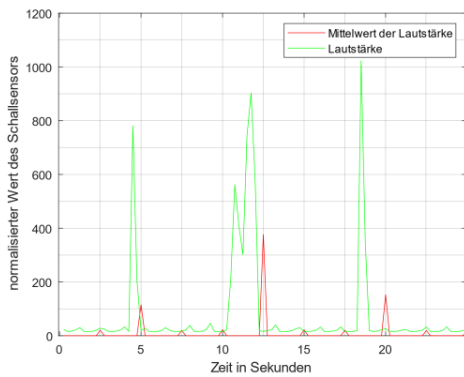


Abbildung 3: Diagramm für die Notwendigkeit der Mittelwertberechnung am Soundsensor

einem bestimmten Zeitintervall der Mittelwert von den des Soundsensors ermittelt Messwerten errechnet (Abbildung 3). Daher ist als Startsignal zum Beispiel ein Pfeifen von circa 5 Sekunden nötig. Unmittelbar nach dem Start des Roboters sind die Tastsensoren für den weiteren Programmablauf verantwortlich. Durch das Fahren gegen eine Wand wird entweder der eine oder der andere Tastsensor gedrückt, was von der Ausrichtung des Roboters abhängt. Das Drücken eines Tastsensors hat zur Folge, dass die Drehrichtung des Antriebsmotors geändert und so auch die Fahrtrichtung umgekehrt wird. Fährt der Roboter anfangs nach links und trifft dort auf eine Wand, so wird so der linke Tastsensor gedrückt. Nun wird die Drehrichtung des Antriebsmotors umgekehrt und der Roboter fährt nach rechts bis er auf eine Wand trifft und der rechte Tastsensor gedrückt wird. So ist es dem Roboter möglich, die zurückgelegte Strecke beziehungsweise die Länge des Fensterbretts zu ermitteln. Zuletzt kommt der Farbsensor zum Einsatz. Dieser wird benötigt, um die Pflanzen zu erkennen, die vorher mit farbigem Papier markiert wurden. Erkennt der Farbsensor eine der festgelegten Farben, speichert er die Position der Pflanze in Abhängigkeit von der zurückgelegten Strecke. So ist es dem Roboter möglich, die Pflanzen zielgenau anzusteuern.

C. Programm

Um das in MATLAB geschriebene Programm mit dem Roboter verknüpfen zu können, ist es nötig den NXT-Stein mittels Bluetooth oder USB-Kabel mit dem Computer zu verbinden. Sobald das Programm in MATLAB gestartet wurde, erscheint die GUI (Abbildung 2) auf dem Bildschirm des Computers. Hier soll nun festgelegt werden, ob oder wie lange welches Pflanzenprofil und somit welche Pflanze gegossen werden soll. Als Pflanzenprofile stehen zur Auswahl: blau, grün und rot. Diese stehen in direkter Verbindung mit den gleichfarbigen Papierstreifen, die an den Blumentöpfen angebracht sind. Diese Farbstreifen ermöglichen es dem Roboter die einzelnen Pflanzen unterscheiden zu können. So kann jede Pflanze nach ihren eigenen Bedürfnissen bewässert werden. Wurde der "Gießen"-Button nach Einstellungen zur Bewässerung der einzelnen Pflanzen betätigt, kann es für den Roboter losgehen. Dieser wartet nun auf das Startsignal, welches vom Soundsensor erfasst wird. Sobald er das Signal korrekt empfangen hat, fängt der Antriebsmotor an sich zu drehen und

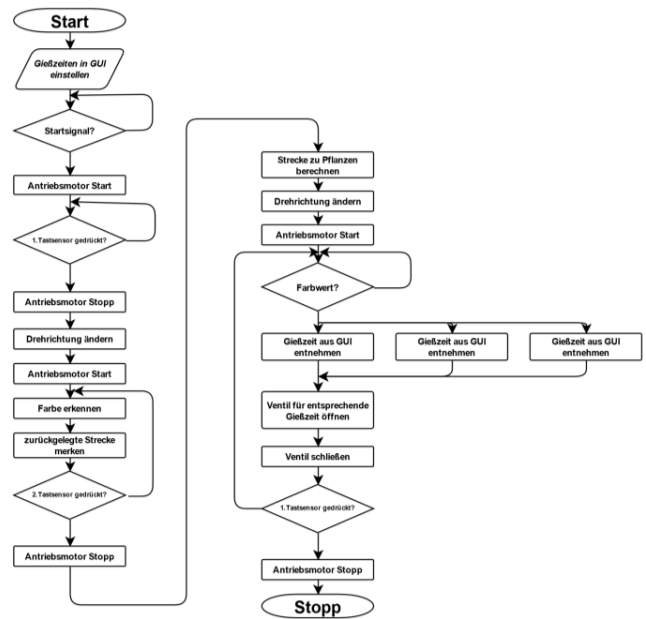


Abbildung 4: Programmablaufplan

der Roboter fährt zunächst in eine Richtung bis eine Wand erreicht wird. Durch das Auftreffen mit der Wand wird der dazugehörige Tastsensor gedrückt. Hierbei wird der Antriebsmotor zunächst gestoppt und der Roboter befindet sich in der Ausgangsposition. Dann wird die Drehrichtung des Antriebsmotors geändert und der Roboter fährt in die entgegengesetzte Richtung zur Wand auf der anderen Fensterbrettseite. Auf dem Weg dahin wird die Umgebung immer wieder vom Farbsensor gescannt. Erfasst dieser eine andere Farbe als Schwarz, welche erkannt wird, wenn kein farbiges Objekt in unmittelbarer Reichweite ist, wird die Position in einem Vektor in Abhängigkeit zur zurückgelegten Strecke gespeichert. Trifft der Roboter dann wieder auf eine Wand wird der zweite Tastsensor gedrückt und der Antriebsmotor wird gestoppt. Anschließend berechnet er die Abstände zu den erfassten Pflanzen. Nach Abschluss der Streckenberechnung ist der Roboter bereit, die Pflanzen zu gießen. Die Drehrichtung des Antriebsmotors wird erneut umgekehrt und die zuletzt erfasste Pflanze wird angesteuert. Sobald der Farbsensor blau, grün oder rot erkennt stoppt er. In Abhängigkeit der erkannten Farbe wird das entsprechende Farbprofil geladen und die Blume wird so lange gegossen, wie es anfangs in der GUI festgelegt wurde. Das Gießen erfolgt über Öffnen des Ventils mit Hilfe des zweiten Motors. Da der Wassertank höher liegt als das Schlauchende, fließt das Wasser aufgrund der Gravitationskraft bei offenem Ventil aus dem Tank in den Topf. Sobald die eingestellte Gießzeit erreicht wird, schließt der Motor das Ventil wieder und der Roboter fährt zur nächsten Pflanze. Dort wird der Gießprozess anhand des dazugehörigen Farbprofils wiederholt. Dies geschieht solange bis alle anfangs erfassten Pflanzen, nach der Einstellung in der GUI gegossen wurden. Nach erfolgreichem Gießen der letzten Pflanze kehrt der Roboter an seine

Ausgangsposition zurück. Damit ist der Gießprozess abgeschlossen. Im vorangegangenen Schema (Abbildung 4) ist der komplette Programmablauf vereinfacht dargestellt.

Studierenden das Programmieren in MATLAB näherzubringen.

IV. ERGEBNISDISKUSSION

Am Ende des Praktikums funktionierten der Bewässerungsroboter und auch die GUI wie erwartet, jedoch nur unabhängig voneinander. Aufgrund von Verknüpfungsproblemen, welche durch eine zu umfangreiche GUI auftraten, gab es den Anlass einer Überarbeitung beziehungsweise Vereinfachung der GUI. Außerhalb des zeitlichen Rahmens des Seminars wurde die vereinfachte GUI (Abbildung 2) entwickelt und mit dem Roboter verknüpft. Obwohl alles mit der neuen GUI einwandfrei funktioniert, gibt es dennoch einige kleine Mängel. Zum einen kommt es nach Betätigen des ersten Tastsensors zu einer unvorhersehbaren Ladepause, welche den Gießprozess teilweise stark verzögert. Zum anderen fasst der Wassertank, aufgrund der Instabilität des Konstrukts, nicht viel Wasser, was zu häufigem Nachfüllen führt. Optimierungsmöglichkeiten wie ein großer stationärer Wassertank wurden bedacht, waren jedoch mit den zur Verfügung stehenden Mitteln, vor allem zeitlich nicht umsetzbar. Hierbei hätte der Wassertank über einen langen Schlauch mit dem Roboter verknüpft werden und sich stets in einer höheren Lage befinden müssen, da der Wasserdruck für das Gießen allein durch die Gravitationskraft entsteht. Eine stabilere Konstruktion für einen größeren mitführbaren Wassertank wurde ebenso überdacht. Da dies jedoch unweigerlich zu einer Vergrößerung des benötigten Platzes führt und dies auf einem Fensterbrett nicht möglich ist, kam dies nicht in Frage. Zudem wäre eine Fernsteuerung, zum Aktivieren des Roboters von außerhalb des Hauses denkbar. Die größten Probleme wurden jedoch bereits während der Entwicklungsphase behoben. Dazu zählt ein lautes Piepen der Motoren, wenn diese in Bewegung waren. Folglich wurde der Soundsensor erneut ausgelöst und das Programm startete erneut von Beginn. Ein weiteres Problem waren zu schnelle Scanintervalle des Farbsensors, weswegen manche Pflanzen doppelt gegossen wurden.

V. ZUSAMMENFASSUNG UND FAZIT

Wenn man die Arbeitszeit nach den vorgesehenen zwei Wochen mitbetrachtet, ist die Entwicklung des Bewässerungsroboters insgesamt zu einem zufriedenstellenden Ergebnis gekommen. Er ist in der Lage, die vorhandenen Pflanzen zu erkennen und sie nach Einstellung des Nutzers fehlerfrei zu gießen, genau wozu er konstruiert wurde. Ein Großteil der Probleme wurde, wenn auch teilweise erst verspätet, bereits gelöst aber die übrig gebliebenen Mängel, wie der zu kleine Wassertank, lassen Verbesserungsmöglichkeiten offen. Diese sind allein mit LEGO nur schwer zu bewältigen aber gewiss nicht undenkbar und hätten mit mehr Zeit auf jeden Fall umgesetzt werden können. Ein täglicher Einsatz ist mit ausreichend Platz auf dem Fensterbrett für Pflanze und Roboter in jedem Fall vorstellbar.

Abschließend sollte noch erwähnt werden, dass das LEGO-Mindstorm-Projektseminar ein guter Weg ist, um den

LITERATURVERZEICHNIS

- [1] Wikipedia: LEGO Mindstorms NXT.
[Lego Mindstorms NXT – Wikipedia](#)
[Stand Februar 2022]
- [2] Mathworks: Matlab.
[MATLAB - MathWorks - MATLAB & Simulink](#)
[Stand Februar 2022]
- [3] Twitch: Mathias Magdowski.
[mathiasmagdowski - Twitch](#)
[Stand Februar 2022]
- [4] Mathworks: RWTH Aachen–Mindstorms NXT Toolbox.
[RWTH - Mindstorms NXT Toolbox - File Exchange - MATLAB Central \(mathworks.com\)](#)
[Stand Februar 2022]
- [5] Innok Robotics GmbH, „Rainos“.
[Rainos - Roboter zur automatischen Bewässerung von Gräbern](#)
[Stand Februar 2022]
- [6] Engelbert Kötter, „Fachkräfte werden immer knapper“, Dega. [Fachkräfte werden immer knapper - DEGA GARTENBAU - Fachmagazin \(dega-gartenbau.de\)](#)
[Stand Februar 2022]

Automatischer Bewässerungsroboter

Alexander Brian Schwerin, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—In diesem Artikel wird im Rahmen des Projektseminars Elektrotechnik/Informationstechnik 2022 die Konstruktion und die Programmierung eines automatischen Bewässerungsroboters beschrieben. Dieser Roboter ist in der Lage, Pflanzen anhand von Farbmarkierungen eine Position auf einer eindimensionalen Bewegungsachse zuzuordnen und entsprechend des vorher über eine GUI eingestellten Farbprofils zu gießen. Dabei werden außerdem einige übliche Bewässerungssysteme vorgestellt und auch einige Elemente von MATLAB und dem LEGO-Mindstorm-Set näher gebracht.

Schlagwörter—Bewässerung, Farbsensor, Geräuschaktivierung, LEGO-Mindstorm, MATLAB.

I. EINLEITUNG

Pflanzen sind wichtige Elemente des Alltags. Sie bilden nicht nur als Nahrungsgrundlage und Sauerstoffproduzenten die Grundlage für unser Überleben, sondern sind auch als ästhetische Elemente eine Bereicherung für jedermann mit einer positiven Auswirkung auf die mentale Gesundheit [6, 7]. Allerdings benötigen Pflanzen als Lebewesen bestimmte Grundvoraussetzungen um zu überleben. Diese Grundvoraussetzungen wie Wasser, Nährstoffe, die richtige Temperatur und genügend Licht müssen möglichst konstant und in der richtigen Menge beziehungsweise Intensität vorliegen, um ein gutes Wachstum zu gewährleisten. Das kann insbesondere für Privatpersonen im Alltag eine Herausforderung darstellen. Um dem entgegenzuwirken gibt es verschiedene Ansätze für automatische Bewässerungssysteme, die zumindest den Bewässerungsteil übernehmen. Allerdings sind alle üblichen Bewässerungssysteme stationäre Aufbauten. Die Idee dieses Bewässerungsroboters ist es einen anderen Ansatz zu verfolgen und ein mobiles Bewässerungssystem zu bauen, welches einzelne Schwächen der bisherigen Systeme überwinden soll.

II. VORBETRACHTUNGEN

A. Stand der Technik

Es gibt viele Ansätze, um den Prozess der Bewässerung zu automatisieren. Die Anwendungsbereiche gehen dabei von einzelnen Zimmerpflanzen bis zur großflächigen Bewässerung von landwirtschaftlich genutzten Ackerflächen. Um im Rahmen des Projekts zu bleiben, wird der Umfang aber auf den heimischen Garten und die Bewässerung einzelner Pflanzen im Innenbereich beschränkt.

1) *kommerzielle Lösungen*: Das kommerzielle Segment ist vorrangig im Bereich der Gartenbewässerung vertreten. Dort wird vor allem mit Wassersprinklern gearbeitet. Die grundlegende Steuerung wird über einen Bewässerungscomputer und elektrische Ventile umgesetzt. In den meisten Fällen werden die Gießzeiten über den Bewässerungscomputer festgelegt. Es gibt

aber auch Umsetzungen, welche über zusätzliche Sensoren wie zum Beispiel Bodenfeuchtigkeitssensoren die Gießzeiten anpassen. Entsprechende Beispiele für Bewässerungssysteme wären die Systeme von Gardena und Regenmeister, siehe [1]. Aber auch für den häuslichen Bereich gibt es einige Möglichkeiten. Einen Ansatz liefert zum Beispiel die Firma Parrot mit einem smarten Blumentopf, siehe [2]. Der Blumentopf nutzt dabei verschiedene Sensoren und pflanzenspezifische Daten, um die optimale Wasserversorgung der Pflanze und mehr sicherzustellen. Eine andere passive Möglichkeit ohne Elektronik stellen Ton- und Keramikwasserspender dar. Dabei handelt es sich um einen Kegel, bestehend aus Ton oder Keramik. Diesen steckt man in die Pflanzenerde und verbindet ihn je nach Modell mit einem Wasserreservoir. Über die feinen Poren des Kegels wird dann über eine längere Zeit das Wasser aus dem Reservoir abgegeben [2].

2) *selbst gebaute Lösungen*: Die Lösungen zum Nachbauen nutzen im Grunde dieselben Funktionsweisen wie auch das kommerzielle Segment. Die jeweiligen Umsetzungen sind allerdings in den meisten Fällen kreativer und fallen jedes Mal etwas anders aus. Es gibt zum Beispiel viele selbst gebaute Bewässerungssysteme auf der Basis von Mikrocontrollern. Diese sind praktisch der Ersatz für die Bewässerungscomputer und steuern Pumpen oder Ventile, siehe [3]. Andere Lösungen versuchen den Effekt von Keramikwasserspendern zu erzeugen. Dazu werden zum Beispiel wasserdurchlässige Seile mit einem Ende in ein Wassergefäß gelegt und mit dem anderen in die Pflanzenerde gesteckt. Dabei wird der Kapillareffekt im Seil genutzt, um langsam das Wasser aus dem Gefäß in die Erde zu befördern, siehe [4]. Eine weitere Möglichkeit, welche besonders in der 3D-Druck-Community vertreten ist, ist das Drucken eines Blumentopfes mit einem eingebauten Wasserreservoir. Dadurch lässt sich die Gießhäufigkeit senken [5].

B. LEGO-Roboter

Ein Punkt, der bei den bisher existierenden Lösungen auffällt, ist die geringe Flexibilität. Alle Systeme sind stationär und müssen manuell auf die jeweiligen Bedingungen eingestellt werden. Wenn zum Beispiel die Positionen einzelner Pflanzen getauscht werden oder einzelne Pflanzen ganz herausgenommen werden, gibt es in keinem der Systeme eine Möglichkeit, dies zu erkennen und automatisch darauf zu reagieren. Der LEGO-Bewässerungsroboter soll diese Lücke schließen und selbstständig anhand von Voreinstellungen und Farbmarkierungen Pflanzen erkennen und richtig versorgen.

C. Vorwissen

Um der nachfolgenden Vorstellung folgen zu können, ist ein wenig Vorwissen nötig. Bis auf das Ventil stammen alle

verwendeten Bauelemente inklusive der Elektronik aus dem LEGO-Mindstorm-Set. Das Set ist eine von LEGO entwickelte Zusammenstellung von Legoteilen, Sensoren, Motoren und dem NXT-Stein das einen einfachen Einstieg in die Robotik bieten soll. Dabei stellt der NXT als programmierbare Steuereinheit das Herzstück dar. Für die Programmierung wird beim Lego-Praktikum die besonders auf Matrizenrechnung spezialisierte Software MATrix LABoratory oder in Kurzform MATLAB verwendet. Für die Programmierung des NXT-Steins steht die Mindstorms-NXT-Toolbox zur Verfügung. Dabei handelt es sich um eine Bibliothek mit vorgefertigten Befehlen, welche von der RWTH Aachen entwickelt wurde. Für dieses Projekt wurden zwei Motoren, ein Farbsensor, ein Mikrofon und 2 Tastsensoren verwendet. Die Motoren lassen sich schrittweise drehen. Dabei ergeben 360 Schritte eine vollständige Umdrehung. Die GUI, also die grafische Nutzeroberfläche, welche zur Einstellung des Roboters genutzt wird, stammt aus dem Unterprogramm App Designer von MATLAB.

III. UMSETZUNG

A. Konstruktion

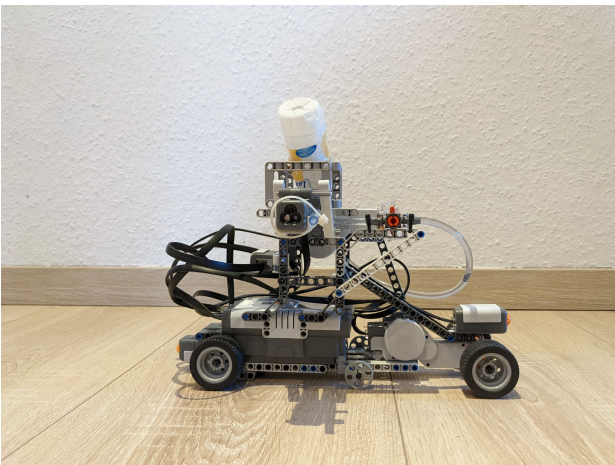


Abbildung 1. Seitenansicht des Roboters

Der Roboter lässt sich praktisch in zwei größere Bereiche aufteilen, einmal eine Unterkonstruktion, welche für die Bewegung zuständig ist und als eine Halterung für den NXT-Stein dient und den Arm, welcher die restlichen Sensoren und die Bewässerungsvorrichtung trägt. Der Roboter soll in der Lage sein, Pflanzen auch in räumlich begrenzten Gebieten wie zum Beispiel einer Fensterbank zu gießen. Deshalb ist die Unterkonstruktion länglich und schmal gestaltet. Um mehrere Pflanzen erreichen zu können ist die Unterkonstruktion mit Rädern und einem Antriebsmotor ausgestattet, wodurch es möglich ist, eindimensional zu fahren und somit eine ganze Fensterbank abzudecken. Da der Roboter aus LEGO besteht, müssen auch einige materialbedingte Probleme abgefangen werden. Zum einen stellt das relativ hohe Gewicht des Arms ein Problem für die längliche Konstruktion dar. Um die Unterkonstruktion zu entlasten, befindet sich deshalb ein Stützrad in der Mitte der Unterkonstruktion. Zum anderen

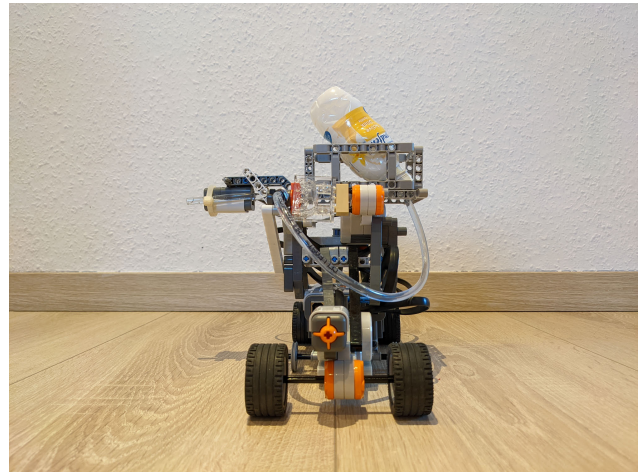


Abbildung 2. Frontansicht des Roboters

gestaltet es sich mit einer reinen LEGO-Konstruktion schwierig, eine wirklich gerade Linie abzufahren, da die hohe Belastung zu einer elastischen Verformung der Bauteile der Unterkonstruktion führt. Um dieses Problem zu lösen, fährt der Roboter in einer Führungsschiene, welche eine gerade Linienführung ermöglicht und den Roboter davor bewahrt, begrenzte Bereiche wie Fensterbänke ungewollt zu verlassen. Die Unterkonstruktion beinhaltet des Weiteren zwei Tastsensoren, welche an den Enden befestigt sind. Diese sind notwendig, um Wände und ähnliche Hindernisse erkennen zu können. Die Armkonstruktion trägt noch zwei weitere Sensoren. Es gibt ein Mikrofon, welches für die geräuschgesteuerte Aktivierung des Roboters zuständig ist und den Farbsensor, welcher zur Bestimmung der Farbmarkierungen an den Pflanzen genutzt wird. Der Farbsensor steht wie in Abbildung 2 zu sehen ist seitlich ab. Dadurch kann der Roboter Blumentöpfe mit einer Größe von maximal 14 cm erkennen. Mit entsprechend längeren Farbmarkierungen lassen sich auch Töpfe mit einer Mindesthöhe von 8 cm sicher erkennen. Um diese Töpfe gießen zu können, besitzt der Roboter eine Bewässerungsvorrichtung. Die Bewässerungsvorrichtung basiert auf dem Prinzip des Schweredruckes. Dadurch spart man sich die Füllstandskontrolle, welche bei einer Lösung mit einer Pumpe beispielsweise nötig gewesen wäre. Der Wasserbehälter sitzt in einer Halterung, welche etwas höher angebracht ist als die Austrittsöffnung am Farbsensor. Der Behälter ist über eine Schraubverbindung und einen Schlauch mit einem Ventil verbunden, welches über einen Motor gezielt angesteuert werden kann.

B. GUI

Um den Roboter auf verschiedene Pflanzen und ihre Bedürfnisse anpassen zu können, besitzt er einige Einstellungsmöglichkeiten. Diese sollten möglichst nutzerfreundlich und leicht erreichbar sein. Darum besitzt der Roboter eine GUI. Über diese GUI kann der Nutzer für 3 verschiedene Farbprofile die entsprechende Gießzeit einstellen, also bestimmen, wie lange oder ob das Ventil geöffnet werden soll und somit auch die Menge an Wasser festlegen. Dabei ist die Anzahl der Farbprofile entsprechend der benötigten Anzahl potenziell

erweiterbar und nur durch die Anzahl erkennbarer Farben des Farbsensors beschränkt. Um dies zu erreichen, gab es zwei Versionen der GUI. Die erste Version beinhaltete verschiedene bewegliche visuelle Elemente, war aber schlussendlich zu kompliziert und wurde nicht in der Regelzeit fertiggestellt. Der finale Roboter läuft deshalb auf einer weitaus simpleren und passenderen GUI, welche die nötigen Grundfunktionen bietet, siehe Abbildung 3.

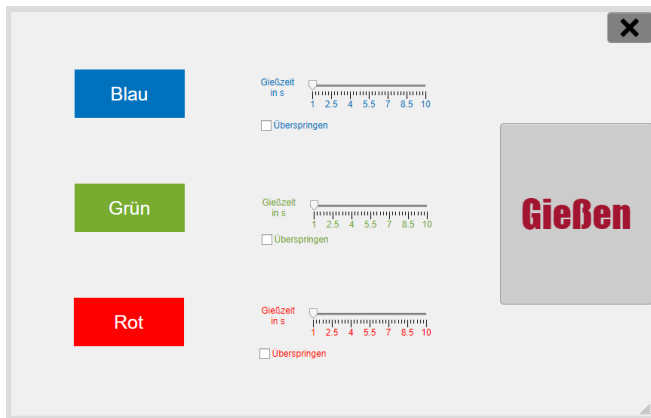


Abbildung 3. Grafische Nutzer-Oberfläche

C. Algorithmus

Der Algorithmus, der den Roboter zum Laufen bringt, besteht grundlegend aus fünf Hauptschleifen. Um diese besser nachvollziehen zu können, befindet sich im Anhang ein Programmablauf, siehe Abbildung 5. Bevor der Roboter seine Arbeit verrichten kann, müssen erst einmal die entsprechenden Werte über die GUI übermittelt werden. Sobald die Werte übertragen wurden, befindet sich der Roboter in Startposition und der erste Programmabschnitt läuft an. Über das Mikrofon wird fortan regelmäßig die Lautstärke gemessen. Aus diesen Werten wird dann ein Durchschnittswert gebildet, um ein versehentliches Auslösen zu verhindern. Der Effekt ist in Abbildung 4 gut erkennbar.

Auch laute Störgeräusche führen nicht zu einem starken Ausschlag im Durchschnitt erst wenn ein lang anhaltendes und lautes Geräusch wie zum Beispiel ein Pfeifen gemessen wird, übersteigt der Durchschnittswert den Sollwert und es kommt zu einer Aktivierung. Sobald das geschehen ist, beginnt der Roboter mithilfe des Tastsensors eine Begrenzung beziehungsweise Wand aufzuspüren. Dieser Schritt ist wichtig, da hierbei eine Kalibrierung stattfindet. Sobald der Roboter eine Begrenzung erreicht hat, kehrt sich die Fahrtrichtung um und der Farbsensor wird aktiviert. Die Bewegung ist ab diesem Schritt nicht mehr flüssig, sondern findet in kleinen Einzelschritten statt. Intern wird die Anzahl der Schritte in einer Variable festgehalten und genutzt, um die jeweiligen Strecken zu berechnen. Dabei wird die erste Begrenzung als Startpunkt angesehen. Bei einem Raddurchmesser von 45 mm und einer Schrittweite von 10 Grad hat der Roboter damit eine theoretische Genauigkeit von 3,9 mm. Mit jedem Schritt wird der Wert des Farbsensors überprüft. Wenn eine

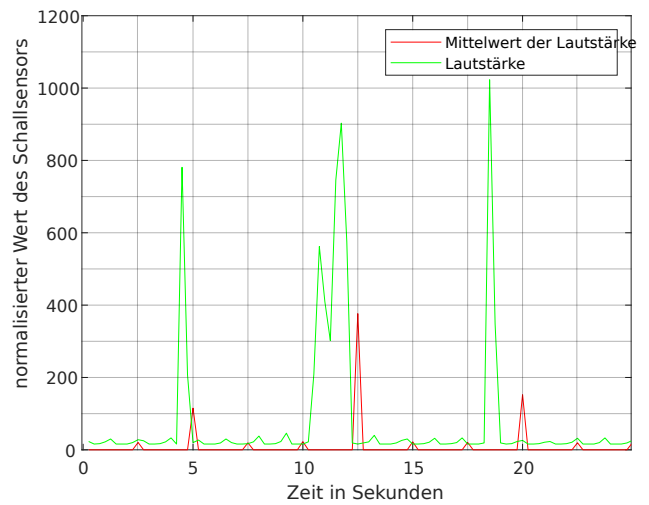


Abbildung 4. Mikrofonaufnahme über den NXT-Stein

Farbe erkannt wird, wird diese zusammen mit der jeweiligen Schrittzahl gespeichert. Um sicherzugehen, dass der Farbsensor eine Farbmarkierung nicht mehrfach misst, wird im Programm sichergestellt, dass die gespeicherten Schrittzahlen bzw. der Abstand der Markierungen einen voreingestellten Mindestabstand nicht unterschreitet. Erst wenn diese Bedingung erfüllt ist, wird der neue Wert gespeichert. Das Ganze wiederholt sich so lange, bis der Tastsensor das zweite Hindernis aufgespürt hat. Mithilfe der gespeicherten Schrittzahlen kann nun über die Gesamtschrittzahl die Strecke berechnet werden, die der Roboter abfahren muss, um zu den jeweiligen Pflanzen zu gelangen. Dabei ist es egal, wie viele Pflanzen es gibt und in welcher Reihenfolge diese auftauchen. Die Anzahl der Pflanzen ist nur durch die voreingestellte Obergrenze beschränkt, welche notwendig ist, um eine Neuskalierung der Speichervektoren innerhalb der Schleifen zu verhindern. Eine Neuskalierung ist nämlich ein ziemlich aufwendiger Prozess und würde zu einer Verzögerung des Programmablaufs führen. Das Problem betrifft die Speichervariablen für die Strecken der Pflanzen jedoch nicht, da diese erst bei der Berechnung in Abhängigkeit der gemessenen Pflanzenzahl erzeugt werden. Der Roboter fährt nun die Pflanzen in umgekehrter Reihenfolge, sprich von hinten nach vorne an. Dabei findet für jede Pflanze einzeln eine Fallunterscheidung mithilfe der gespeicherten Farbwerte statt. Je nach Farbe werden die entsprechenden Einstellungen aus der GUI genutzt. Wenn alle Pflanzen abgearbeitet wurden, sucht der Roboter wieder mithilfe des Tastsensors seine Ausgangsposition, sprich das erste Hindernis und beendet das Programm.

IV. ERGEBNISDISKUSSION

Der Bewässerungsroboter erfüllt die Erwartungen und weist alle geplanten Fähigkeiten auf. Der Roboter besitzt eine eigene GUI in der wichtige Einstellungen bezüglich des Gießverhaltens getroffen werden können, und ist nach der Initialisierung jederzeit durch akustische Signale aktivierbar. Des Weiteren ist er in der Lage, Pflanzen anhand von Farbmarkierungen zu

erkennen, auseinanderzuhalten und entsprechend den Einstellungen zu gießen. Dabei ist aufgrund einer Kalibrierung und einer Positionsbestimmung der Pflanzen eine flexible Reaktion auf Veränderungen bezüglich der Reihenfolge und Anzahl der Pflanzen möglich. Dennoch sind die Einsatzmöglichkeiten des Roboters relativ eingeschränkt. Die LEGO-Konstruktion ist trotz des länglichen Aufbaus zu breit für die meisten Fensterbänke. Demnach beschränkt sich das Einsatzgebiet eher auf Fußböden und Tische. Des Weiteren ist die Farberkennung über den Farbsensor eine heikle Sache. Die Reichweite des Sensors unterscheidet sich je nach Farbe und beträgt maximal einige Zentimeter. Außerdem muss im jeweiligen Code eine Art Filter eingebaut werden, um Mehrfachmessungen der selben Markierung zu vermeiden. Um die Farbmarkierungen zuverlässig erkennen zu können, benötigt der Roboter die bereits erwähnte Führungsschiene und auch die Markierungen müssen genau ausgerichtet sein. Des Weiteren ist die Auswahl an Töpfen stark eingeschränkt. Einerseits natürlich durch die Maximalhöhe von 14 cm, aber auch nach unten durch den unbeweglichen Arm auf etwa 8 cm. Da dieser fest verbaut ist müssen die Farbmarkierungen nämlich immer in der selben Höhe hängen und auch der Wasserstrahl ist nicht lenkbar. Dadurch lassen sich zu kleine Töpfe nicht immer zielgenau treffen. Außerdem ist die Wassermenge aufgrund des hohen Gewichts stark begrenzt.

V. ZUSAMMENFASSUNG UND FAZIT

Als Ergebnis des Projekts ist ein Roboter hervorgegangen, der nach den nötigen Voreinstellungen über eine GUI automatisch die Bewässerung mehrerer Pflanzen übernehmen kann und in der Lage ist Pflanzen über verschiedene Farbprofile zu unterscheiden. Um den Roboter besser an sein Aufgabengebiet anzupassen wäre es dennoch sinnvoll, auf technische Möglichkeiten außerhalb des LEGO-Segments zurückzugreifen. Somit könnte man den Roboter zum Beispiel über ein drahtloses Netzwerk ansteuern und weitere wichtige Daten wie Bodenfeuchtigkeits- und Nährstoffwerte einspeisen, um die Effizienz zu erhöhen. In diesem Zuge wäre auch ein Umstieg auf ein widerstandsfähigeres Material wie Aluminium für die Konstruktion sinnvoll, um die Traglast für einen größeren Wasserbehälter zu erhöhen und die Konstruktion schmäler zu gestalten, damit der Roboter auch auf dünnen Fensterbänken arbeiten kann.

LITERATURVERZEICHNIS

[1] SCHNEIDER, Jennifer: *Garten-Bewässerungssysteme im Test: Sieger der Stiftung Warentest.* https://praxistipps.chip.de/garten-bewaesserungssysteme-im-test-sieger-der-stiftung-warentest_102650, 2020. – [Online; accessed 23-Februar-2022-14:22 Uhr]
 [2] JOERG: *Das beste Bewässerungssystem für Zimmerpflanzen.* <https://schlauer-wohnen.com/das-beste-bewaesserungssystem-fuer-zimmerpflanzen/>, 2021. – [Online; accessed 23-Februar-2022-14:29 Uhr]
 [3] BK iLyas: *Arduino Plant Watering System - Complete guide.* <https://www.youtube.com/watch?v=JdvnfENodak>, 2020. – [Online; accessed 23-Februar-2022-14:48 Uhr]
 [4] SHOW, Rachael R.: *Self-Watering Plant Hack — Plant Doctor Hilton Carter.* https://www.youtube.com/watch?v=9E-i_RIPbJ8, 2020. – [Online; accessed 23-Februar-2022-15:01 Uhr]
 [5] PARALLELGOODS: *Self-Watering Planter (Small).* <https://www.thingiverse.com/thing:903411>, 2015. – [Online; accessed 23-Februar-2022-15:24 Uhr]

[6] KUHLMANN, Susanne: *Grünpflanzen für die Gesundheit.* <https://www.deutschlandfunk.de/gruenpflanzen-fuer-die-gesundheit-100.html>, 2009. – [Online; accessed 24-Februar-2022-18:10 Uhr]
 [7] PANZETER, Elena: *Pflanzen für die Seele.* <https://blog.uni-koblenz-landau.de/pflanzen-fuer-die-seele-zimmerpflanzen-psychologie/>, 2021. – [Online; accessed 24-Februar-2022-18:34 Uhr]

ANHANG

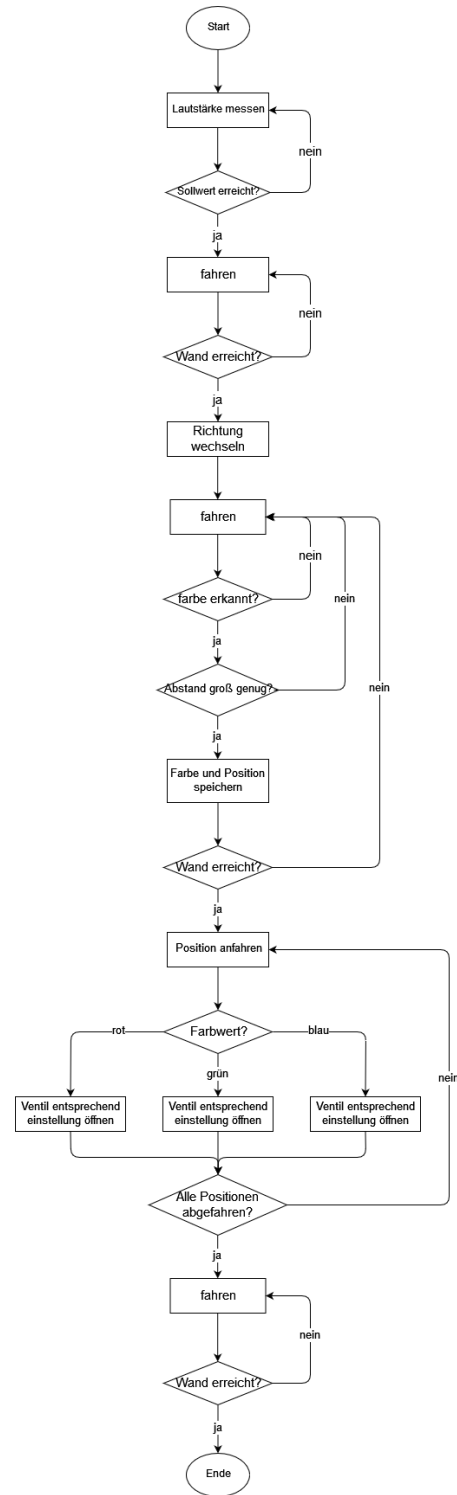


Abbildung 5. Ablaufplan des Algorithmus

Sally can Draw: Automatisierter Light-Painting-Roboter

Michelle Horn, Computervisualistik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des Projektseminars Elektrotechnik/ Informationstechnik 2022 wurde ein Light-Painting-Roboter entwickelt. Dieser ist in der Lage, Kanten in einem Bild zu finden und diese mit einer Lichtquelle, vor einem Hintergrund, nachzuzeichnen. Im Folgenden werden verschiedene Konstruktionsmöglichkeiten, sowie ihre Vor- und Nachteile beleuchtet. Außerdem werden aufgetretene Probleme und Lösungen erläutert. Das mit MATLAB implementierte Programm wird in den Schritten Bildverarbeitung und Kantenverfolgung erklärt.

Schlagwörter—Bildverarbeitung, LEGO-Mindstorms, Light-Painting, MATLAB, Roboter, Segmentierung

I. EINLEITUNG

LIGHT-PAINTING, oder auch Light-Writing, ist eine fotografische Technik, die mittels Langzeitbelichtung durchgeführt wird. Dabei wird im dunklen mit einer oder mehreren Lichtquellen gemalt oder geschrieben. Besonders schöne Effekte entstehen, wenn Light-Painting vor einem Hintergrund angewendet wird. Die Ergebnisse sind erst später auf dem Ergebnisbild sichtbar. Ist eine Linie gezeichnet kann sie nicht wieder entfernt werden, was bedeutet, dass ein Fotograf in der Regel viele Versuche benötigt, um ein Motiv wie gewünscht zu fotografieren. Um die Gesamtdauer eines Light-Painting-Bildes zu verkürzen, wurde *Sally can Draw*, der automatisierte Light-Painting-Roboter, konzipiert. So soll es durch eine exakte Linienführung möglich sein, das ideale Ergebnisbild in wenigen Versuchen aufzunehmen. Des Weiteren ist es so nicht notwendig, selbst zu zeichnen, was das Wechseln zwischen der Kamerabedienung und dem Zeichnen erspart. So ist es potenziell möglich, die Belichtungszeit zu reduzieren. Ab einer bestimmten Belichtungszeit kann das Bild bzw. die Umgebung auf dem Bild ansonsten nämlich zu hell werden. Der Roboter soll möglichst genau zeichnen. Dabei soll die Lichtquelle verdunkelt oder deaktiviert werden können, um keine fehlerhaften Linien zu erzeugen, wenn das Bild aus mehr als einem geschlossenen Kantenzug besteht. Außerdem wird das Zeichnen vor einer 3D-Szene als Hintergrund bevorzugt, weshalb der Light-Painting-Roboter in der Lage sein muss, im Raum zu zeichnen. Weiterhin werden Anforderungen an das Programm gestellt. Es muss in der Lage sein, eingelesene Bilder zu verarbeiten und darin Kantenzüge zu segmentieren. Dabei soll die Berechnungsdauer gering sein und es dürfen nicht zu viele Befehle in kurzer Zeit an den NXT-Stein übertragen werden.

DOI: 10.24352/UB.OVGU-2022-020

Lizenz: CC BY-SA 4.0

II. VORBETRACHTUNGEN

Es ergeben sich drei Konstruktionsansätze zur Umsetzung des Light-Painting-Roboters. Dabei ist die in Abschnitt II-C beschriebene Variante den anderen vorzuziehen, weil neben voller Funktionalität auch die Anforderung, vor einem Hintergrund zu zeichnen, erfüllt ist.

A. Umsetzung als Arm mittels inverser Kinematik

Der Ansatz, den Light-Painting-Roboter durch einen Arm mittels inverser Kinematik umzusetzen, wird zuerst verfolgt. Wie schematisch in Abbildung 1 zu sehen ist, hat der Arm zwei Freiheitsgrade und sein Endeffektor ist eine abdeckbare Kiste. Diese bietet Platz für eine AA-Batterie und eine LED. Hierbei werden drei Motoren verwendet, zwei zum Steuern des Armes und einer, um die Kiste mit einer Klappe abzudecken. Der NXT-Baustein befindet sich am Standsockel des Armes und bildet damit gleichzeitig ein Gegengewicht. Durch die limitierte Kabellänge ist der Arm sehr klein. Das beschränkt das Zeichnen auf einen kleinen Radius. Der erreichbare Detailgrad beim Zeichnen ist somit fraglich. Durch das Eigengewicht der Motoren kommt es zum Absacken der Arnteile. Außerdem gibt es Stabilitätsprobleme. Wegen diesen Problemen wird dieser Ansatz nicht weiter verfolgt, obwohl er den Vorteil bietet, vor einem Hintergrund zeichnen zu können.

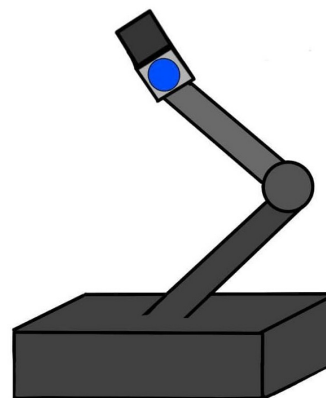


Abbildung 1. Skizzenhafter Entwurf des Light-Painting-Roboterarms

B. Umsetzung als Fahrzeug

Die Umsetzung als Fahrzeug löst die Stabilitätsprobleme, sowie die Limitierung durch die Kabellänge. Wie in Abbildung 2 zu sehen, ist der Prototyp frontgetrieben und verfügt

über die gleiche Klappvorrichtung wie in dem Abschnitt II-A beschriebenen Ansatz. Die zwei verbleibenden Motoren bilden die Vorderachse. Dabei ist ein Motor zum Vorwärts- und Rückwärtsfahren und der andere zum Lenken. Die Umsetzung als Fahrzeug erlaubt es lediglich auf dem Boden – also einem schwarzen Hintergrund zu zeichnen. Dafür ist es notwendig, die Kamera senkrecht über dem Boden zu platzieren, was zu Problemen führt. Zum einen ist ein Fernauslöser notwendig. Des Weiteren ist die Platzierung nicht mit einem Standardstativ möglich.



Abbildung 2. Prototyp des Fahrzeuges

C. Umsetzung als Gerüstkonstruktion

Die Gerüstkonstruktion besteht aus einem Rahmen an dem, wie in Abbildung 3 zu sehen ist, oben der NXT-Baustein sowie zwei Motoren angebracht sind. Der genaue technische Aufbau ist im Abschnitt III beschrieben. Als Leuchtmittel ist hierbei der in Abschnitt II-D erklärte Calliope mini verbaut. Die Umsetzung mit dem Gerüst vermeidet ebenfalls Stabilitätsprobleme und eine Limitierung durch die Kabellänge. Zudem ermöglicht sie das Zeichnen vor einem Hintergrund und bietet einen großen Zeichenradius. Das Fotografieren mit einem Standardstativ und ohne Fernauslöser ist gut umsetzbar.

D. Leuchtmittel

Eine einfache LED, die mit einer AA-Batterie betrieben wird, reicht als Leuchtmittel zum Zeichnen bereits aus. Sie ist nicht steuerbar und somit entweder an oder aus. Da eine Ansteuerung über den NXT nicht möglich ist, wird die LED vor dem Starten der Aufnahme manuell angeschaltet und dann mit einer motorisierten Klappe verdunkelt, um keine falschen Linien im Ergebnisbild zu erzeugen.

Der Calliope mini ist ein zu Bildungszwecken entwickelter Mikrocontroller [1]. Im Gegensatz zur einfachen LED kann die LED auf dem Calliope in verschiedenen Farben leuchten. Der Calliope mini ist einfach programmierbar, dazu sind verschiedene webbasierte Entwicklungsumgebungen nutzbar. Durch die Verwendung des Calliope mini kann auf die Nutzung der motorisierten Klappe verzichtet werden. Dazu werden in einem Testdurchlauf die Zeiten gemessen, in denen die Lichtquelle an und aus sein muss. Anschließend wird mittels dieser Zeiten ein Programm geschrieben, auf den Calliope mini geladen und parallel zur Kameraaufnahme gestartet.

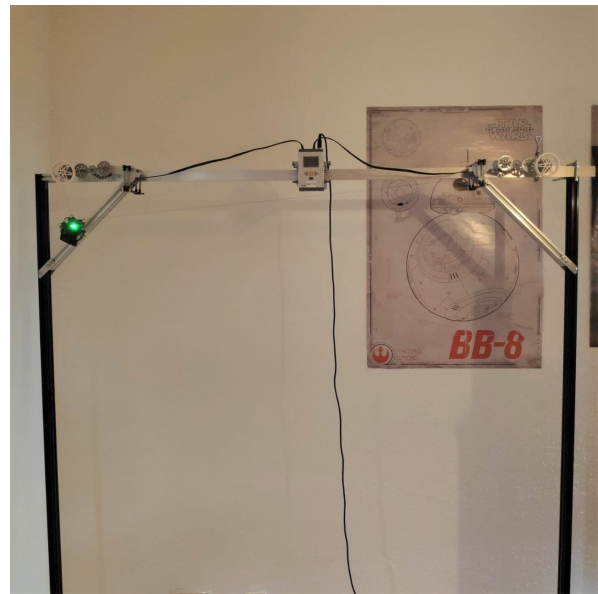


Abbildung 3. Gerüstkonstruktion mit montiertem Calliope mini

III. KONSTRUKTION

Die finale Umsetzung ist die Gerüstkonstruktion. Wie bereits im Abschnitt II-C erwähnt, besteht diese Konstruktion aus einem Metallrahmen. Auf der oberen Metallschiene sind der NXT-Baustein und zwei Motoren durch LEGO angebracht. Um den Zeichenradius zu maximieren sind die Motoren über eine Zahnradkonstruktion, wie in Abbildung 4 zu sehen, verlängert. Am Ende befindet sich jeweils eine Winde, auf die Fäden aufgewickelt ist. An diesen Fäden ist der Calliope mini befestigt. Hier sind alle leuchtenden Bestandteile außer der RGB-LED abgeklebt, um doppelt gemalte Linien im Ergebnisbild zu verhindern. Die Fäden werden durch die Motoren auf- und abgewickelt, um die Lichtquelle entlang der Kanten im Bild zu bewegen. Um ein Abspringen der Fäden von den Winden zu unterbinden, ist vor den Winden ein zusätzliches Rad angebracht.

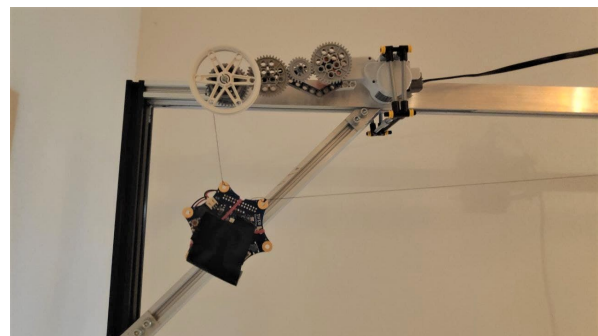


Abbildung 4. Verlängerung der Motoren und modifizierter Calliope mini

IV. SOFTWARE

Die softwareseitige Umsetzung erfolgt in zwei Schritten, der Bildverarbeitung und dem Algorithmus zum Bewegen

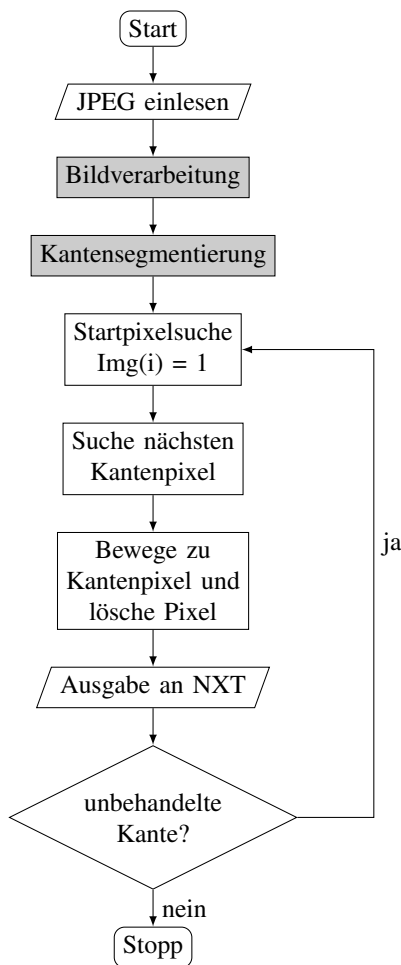


Abbildung 5. Programmablaufplan zur Erklärung des Programms

der Lichtquelle. Dazu gehört auch die Ansteuerung des NXT-Bausteins. Der Programmablaufplan in Abbildung 5 verdeutlicht die Umsetzung.

A. Bildverarbeitung

Bei der softwareseitigen Umsetzung wird zunächst ein Bild eingelesen und in ein Grauwertbild umgewandelt. Dieses Bild wird an die Funktion `highpass.m` übergeben. Diese führt auf dem übergebenen Bild eine Filterung mittels des Laplacian-of-Gaussian-Filters durch. Dieser entspricht einer Rauschreduktion mit anschließender Hochpassfilterung. Dabei ist es gegebenenfalls notwendig, den Parameter σ manuell anzupassen, sollte das Ergebnis nicht ausreichend für die Weiterverarbeitung sein. σ bestimmt dabei die Breite der Kanten. Je kleiner σ ist desto breiter sind die Kanten. Ein größer gewähltes σ führt zu schmaleren oder keinen Kanten. Der für die Beispielbilder gewählte Wert liegt zwischen 0,3 und 0,6. Das resultierende Kantenbild ist in Abbildung 6 (mitte) zu sehen. σ liegt hier bei 0,4.

Dieses Kantenbild wird an die Funktion `GetEdges.m` übergeben. Hier werden die Kanten mit dem Canny-Edge-Operator segmentiert. Dabei entsteht ein Binärbild. Eins steht für Orte an denen eine Kante ist, sonst steht dort eine Null.

Danach werden zu kleine Kanten entfernt. Das ist notwendig, da meist kleine fälschlicherweise erkannte Kanten in der Nähe von echten Kanten auftreten. Dabei werden Kantenzüge gelöscht, die aus weniger Pixeln als ein Schwellenwert bestehen. Dieser Schwellenwert wird manuell angepasst und ist stark vom jeweiligen Bild und auch vom gewählten σ abhängig. Für die getesteten Beispielbilder liegt dieser Wert zwischen 50 und 500. Das Ergebnis dieses Schrittes ist in Abbildung 6 (unten) zu sehen. Die Schwelle zum Löschen kleiner Kanten ist hier auf 400 gesetzt.

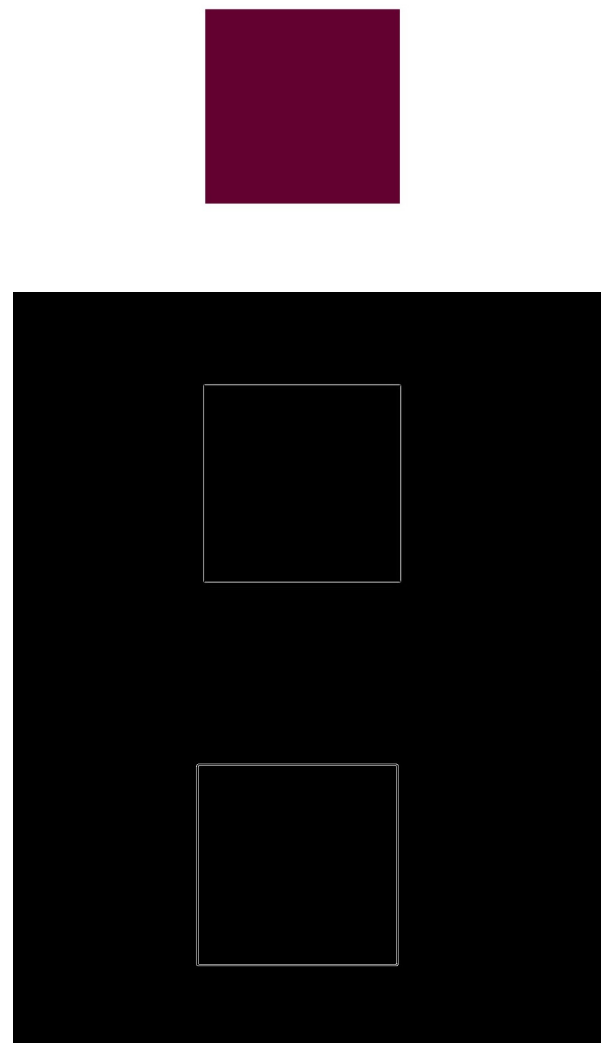


Abbildung 6. Originalbild (oben), Kantenbild als Resultat der Filterung (mitte), segmentierte Kanten (unten)

B. Kantenverfolgung

Als Ausgangsposition für den Calliope mini ist die obere linke Ecke des Rahmens definiert. Auf dem Binärbild wird das Startpixel gesucht, indem das Bild, ausgehend von der linken oberen Ecke, traversiert wird. Das Startpixel ist hierbei das erste Einspixel. Von diesem Pixel werden die umliegenden Pixel in

der 8-Nachbarschaft betrachtet. Ist eines dieser Pixel ebenfalls Teil der Kante wird das aktuelle Pixel gelöscht und das gefundene Pixel wird betrachtet. Wird kein nächstes Kantenpixel gefunden, wird in einem größeren Umkreis erneut gesucht. So werden kleine Löcher im Kantenzug überbrückt. Dieser Radius wird schrittweise größer, um bei einer abgeschlossenen Kante einen neuen Startpixel der nächsten Kante zu finden. Wird bei diesem Prozess keine Kante gefunden, wurde das gesamte Bild gezeichnet und der Algorithmus terminiert. Über die Position der Pixel im Bild wird der Abstand zu den Motoren berechnet. Daraus folgt die benötigte Fadenlänge. Hierzu werden die Motoren pro Pixel um 3° bewegt.

V. ERGEBNISDISKUSSION

In Abbildung 7 ist das aus dem Originalbild in Abbildung 6 (oben) resultierende Ergebnisbild zu sehen. Die Belichtungszeit dieses Fotos liegt bei 18 Sekunden, es wurde mit Blende 29 und ISO-100 fotografiert. Es ist also möglich, komplexere Bilder zeichnen zu lassen, ohne dass die Bildqualität durch Umgebungslicht beeinflusst wird. Dabei fällt auf, dass die Linienführung nicht optimal ist. Horizontale Linien biegen sich leicht nach unten und senkrechte Linien verlaufen nur teilweise gerade. Nachdem die Linie größtenteils gerade verläuft macht sie einen Knick um die Position zu korrigieren. Bei geschlossenen Kantenzügen erreicht die Linie ihren Anfangspunkt erneut genau.

Zum Problem der ungenauen Linienführung ist bisher keine Lösung bekannt. Lösbar sind jedoch einige Konstruktionsprobleme. Aufgrund der verschiedenen Ansätze, um den Light-Painting-Roboter zu konstruieren, die in Abschnitt II aufgeführt sind, ist die entstandene Konstruktion als Prototyp zu betrachten. Trotzdem löst dieser Ansatz viele Probleme aus den vorherigen Entwicklungsstufen. Die Stabilität ist weitestgehend gewährleistet. Es gibt minimale Bewegungen der gesamten Konstruktion bei Bewegung der Motoren. Diese hängen mit der Inkompatibilität zwischen dem Gerüst und LEGO zusammen. Die Limitierung des Zeichenradius durch die begrenzte Kabellänge wird durch die im Abschnitt III erläuterte Erweiterung behoben. Zudem ist das Zeichnen vor einem Hintergrund mit optimaler Perspektive möglich. Daraus folgt auch eine einfache Handhabung beim Fotografieren mittels eines standardmäßigen Stativs.

VI. ZUSAMMENFASSUNG UND FAZIT

Der als beweglicher Arm, Fahrzeug und Gerüstkonstruktion konzipierte Light-Painting-Roboter wird als Gerüstkonstruktion umgesetzt. Dabei werden sowohl die technische als auch die softwareseitige Umsetzung veranschaulicht. Außerdem werden aufgetretene Probleme, ihre Lösungen und die Ergebnisse dargestellt. Der Prototyp ist in der Lage einfache Kantenzüge nachzuzeichnen, bietet aber Potential zur Verbesserung.

Der Light-Painting-Roboter kann durch ein mobiles und stabileres Gerüst ergänzt werden. Ein ansprechenderes Design kann erarbeitet werden. Außerdem können auf der Softwareseite einige Verbesserungen bzw. Erweiterungen vorgenommen werden. Zum einen sollte das Problem der Linienführung gelöst werden. Zudem kann der Algorithmus

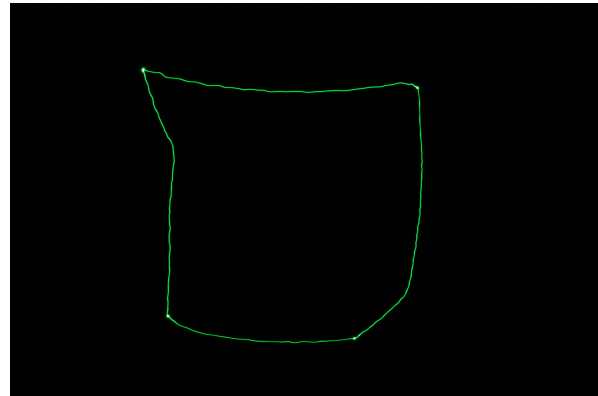


Abbildung 7. Ergebnisbild

um eine Tiefensuche ergänzt werden, um die Reihenfolge, in der die Kanten abgefahren werden, zu beeinflussen. Die Programmierung des Calliope mini kann über eine Bluetooth Verbindung durchgeführt werden, ohne den Calliope mini bei Fertigstellung eines neuen Programms mit dem Computer verbinden zu müssen. Dabei kann über eine Einbindung der Programmierung in MATLAB nachgedacht werden. Die Einstellung der Parameter in der Bildverarbeitung kann beispielsweise über eine GUI nutzerfreundlicher gestaltet werden.

Zusammengefasst bietet der Light-Painting-Roboter viel Potential zur Verbesserung und Erweiterung.

LITERATURVERZEICHNIS

- [1] CALLIOPE GGMBH: *Calliope mini, technische Daten*. <https://calliope.cc/calliope-mini/technische-daten>. Version: Februar 2022

Sally Can Draw – Ein Automatisierter Lightpainting Roboter

Anna-Lena Thalhofer, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des zweiwöchigen Projektseminars Elektrotechnik/Informationstechnik soll mit Hilfe des LEGO-Mindstorm-Sets ein Roboter konstruiert und mit MATLAB programmiert werden. Entwickelt wurde dabei der Lightpainting Roboter mit dem Maskottchen Sally.

Schlagwörter—Bilder, LEGO Mindstorm, Lightpainting, Roboter, Sally

I. EINLEITUNG

LIGHTPAINTING [1] ist für viele Menschen ein Hobby, welches die Bestandteile malen und fotografieren kombiniert. Durch Langzeitbelichtung [2] kann man im Dunkeln mit Leuchtstäben oder auch Wunderkerzen Formen kreieren. Dabei kann man auch die unterschiedlichsten Farben benutzen. Je nach Motiv und der Anzahl Farben im Bild, kann dies längere Zeit und mehrere Versuche benötigen. Für eine einzelne Person dauert das noch um einiges länger, da man immer wieder zwischen Kamera und Hintergrund des Bildes pendeln muss. Innerhalb des LEGO-Mindstorms-Projekts wurden mehrere Konzepte erdacht, wie man den Prozess des Lightpaintings einfacher und schneller gestalten kann.

II. VORBETRACHTUNGEN

Wie bereits erwähnt, wurden mehrere Konzepte ausgedacht und in Betracht gezogen. Dabei hatte jedes Konzept seine eigenen Vor- und Nachteile. Es wurde sowohl die technische als auch die software bezogene Seite betrachtet.

A. Fahrzeug

Das erste Konzept ist ein Fahrzeug. Dieses soll auf dem Boden fahren der ein möglichst dunkler bzw. schwarzer Untergrund sein sollte. Wie man schon erkennt, ist das einer der Nachteile des Fahrzeugs, da im Bild die ganze Szenerie wichtig ist. Durch den schwarzen Untergrund geht der Effekt des Lightpaintings verloren. Das Fahrzeug muss klein und kompakt sein, damit es nicht zu Lenkschwierigkeiten kommt und das Endergebnis somit keine Fehler aufweist. Auf dem Fahrzeug befindet sich eine kleine Box, in der sich die LED befindet. Diese Box soll durch eine motorbetriebene Klappe geöffnet und geschlossen werden. Dadurch sollte kontrolliert werden, wann auf dem Bild ein Lichtstreifen gesehen werden soll und wann nicht

B. Beweglicher Arm

Ein weiteres Konzept ist ein beweglicher Arm, der zwei Motoren als Gelenke hat. Ein dritter Motor soll wie beim Fahrzeug eine Klappe zum Abdecken der LED sein. Im Gegensatz zu dem Fahrzeug kann man den Arm vor einer Szenerie aufstellen und hat damit einen Hintergrund im Bild. Allerdings muss auch der Arm relativ klein sein, da es eine Begrenzung aufgrund der Kabellänge zwischen den Motoren und dem NXT-Stein gibt. Ein weiteres Problem hierbei ist, dass man sowohl hinreichend stabil als auch mit möglichst wenig Bauteilen bauen muss. Durch zu viel Gewicht auf dem Arm kann die Kontrolle durch eine höhere Motorgeschwindigkeit verloren gehen.

C. Metallgerüst

Das letzte Konzept enthält am wenigsten LEGO-Elemente. An einem Metallgerüst soll in jeder Ecke jeweils ein Motor angebracht werden. An den Motoren ist ein Rad angebracht, welches einen Faden aufwickeln soll. An diesem Faden hängt ein Calliope mini, den man separat programmieren kann. In diesem Fall dient der Calliope mini als LED. Dadurch kann man auch mehrere Farben benutzen, wodurch man in der Theorie sehr viel kreativer sein kann. Genauso wie beim Arm kann man dieses Gerüst vor einen Hintergrund stellen. Der eigentliche Effekt vom Lightpainting geht also auch hier nicht verloren.

D. Software

Grundsätzlich ist die Programmierung bei jedem Konzept gleich. Als erstes wird ein Bild hochgeladen, welches der Roboter später malen soll. In diesem Bild werden die Kanten herausgefiltert. Der komplexe Teil, der auch bei jedem Konzept anders ist, ist zu programmieren, wie der Roboter diese Pfade später langfahren soll.

III. KONSTRUKTION UND PROGRAMMIERUNG

Als erstes wurde der bewegliche Arm gebaut (siehe Abbildung 1). Währenddessen wurde schon mehrfach umgebaut, verkleinert und versucht, möglichst viel Gewicht einzusparen. Es wurde aber recht schnell festgestellt, dass die LEGO-Steine zu viel Gewicht haben um den Arm klein und beweglich zu halten. Daher gab es nie einen richtigen Prototypen.

Als nächstes wurde das Fahrzeug (siehe Abbildung 2) gebaut. Es wurde versucht, die Vorderräder mit einem Motor zu betreiben und diese Achse mit einem weiteren Motor zu verbinden, der die Lenkung übernimmt. Auf der festen

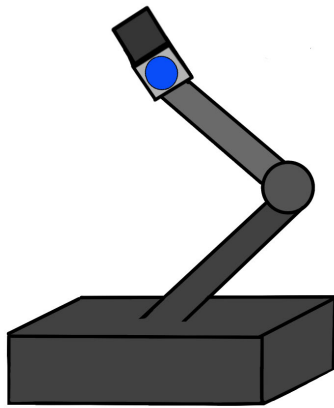


Abbildung 1. Zeichnung des beweglichen Armes

Achse sollte die Box mit der Klappe befestigt werden. Die Schwierigkeit bei dieser Konstruktion war eine Verbindung zu dem NXT-Stein zu schaffen, ohne dass etwas am Boden schleift oder nicht kompakt genug ist. Nach einer Testfahrt wurde beschlossen, nicht länger an diesem Prototyp zu arbeiten, sondern sich mit dem letzten Konzept zu befassen.



Abbildung 2. Prototyp des Fahrzeuges

Das letzte Konzept (siehe Abbildung 3) besteht nur noch aus einem Metallgerüst, dem NXT-Stein, zwei Motoren, den Verbindungsteilen aus LEGO und einem Calliope mini (siehe Abbildung 4) als Lichtquelle. Dies ist sowohl die einfachste als auch die stabilste Konstruktion. Die Motoren sollen so angesteuert werden, dass sie den Faden so auf- und abwickeln je nachdem in welche Richtung der Calliope mini sich bewegen soll.

Der Calliope mini muss durch ein externes Programm angesteuert werden. Dadurch kann MATLAB nicht mit dem Calliope mini verwendet werden. Deshalb muss man einen Timer programmieren, damit die LED nach x Sekunden an- und

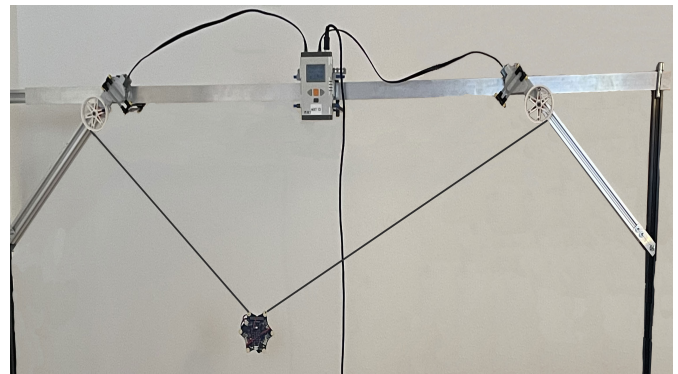


Abbildung 3. Prototyp des Metallgerüsts

ausgeht oder sogar die Farbe wechselt. Eine andere Möglichkeit ist den Calliope mini über Bluetooth anzusteuern. Dies ist aber auch kein optimaler Fall, da die Übertragung auf den Calliope mini eine Weile dauert. Vor den eigentlichen Aufnahmen muss getestet werden, welche Helligkeit der LED am geeignetsten für das Endergebnis ist. In den meisten Fällen ist eine dunklere Helligkeit geeignet, da ansonsten das komplette Bild zu hell werden könnte.

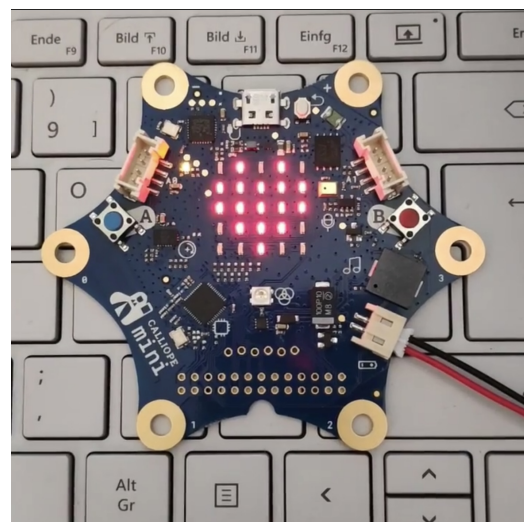


Abbildung 4. Calliope mini

Die Software wurde wie folgt programmiert: Als erstes liest man eine JPEG-Datei ein. Diese wird dann von dem Programm bearbeitet und die Kanten werden herausgefiltert mithilfe des Gaussian Filters [3]. Hat man nun das Kantenbild, sucht das Programm einen Startpixel. Daraufhin wird das Programm den nächstgelegenen Kantenpixel suchen. Zu diesem wird sich dann bewegt und der vorherigen Pixel gelöscht. Dies wird zum NXT-Stein gesendet. Das Programm macht solange weiter, bis keine Pixel mehr vorhanden sind. Diesen Prozess kann man ebenfalls in dem Programmablaufplan in Abbildung 5 nachvollziehen.

IV. ERGEBNISDISKUSSION

Die letzte Konstruktion war die einzige, mit der ein Ergebnisbild (siehe Abbildung 6) gemacht wurde. Anhand des

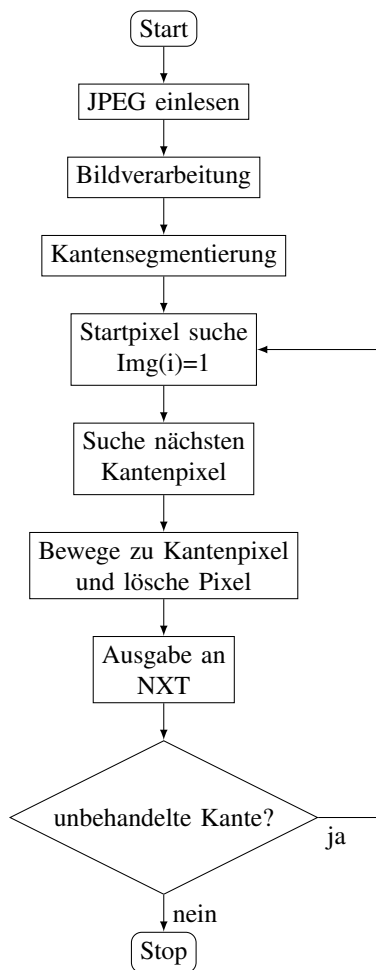


Abbildung 5. Programmablaufplan

Bildes kann man gut erkennen, was mit dem Prototyp schon gut geht und woran man noch arbeiten muss. Unter anderem sieht man, dass die horizontale Bewegung besser ist als die vertikale Bewegung. Dadurch, dass die Fäden nur ein dünner Nähgarn war, hatten diese auch nicht die Kraft gegen die Schwerkraft anzukommen und den Calliope mini dort zu halten wo er sein soll. Das kann man an der unteren Kurve erkennen. Nichtsdestotrotz kann man im Endergebnis erkennen was es werden sollte. Das Gerüst war zwar im Vergleich zu den anderen Konstruktionen am stabilsten, aber dennoch war es sehr wackelig. Wenn die Motoren etwas schneller sind um die Belichtungszeit gering zu halten, fängt das Gerüst leicht an zu wackeln, was man natürlich vermeiden möchte um ein gutes Bild zu erhalten.

V. ZUSAMMENFASSUNG UND FAZIT

Innerhalb der zwei Wochen wurden das Fahrzeug, der bewegliche Arm und das Metallgerüst gebaut und getestet. Durch etwaige Probleme wurde das Metallgerüst weiterverfolgt und ausgebaut. Durch die verschiedenen Konzepte fehlte am Ende die Zeit um das Gerüst zu verbessern, daher kann man dieses nur als einen Prototyp bezeichnen. Dennoch macht es für einen Prototypen bereits eine gute Arbeit.

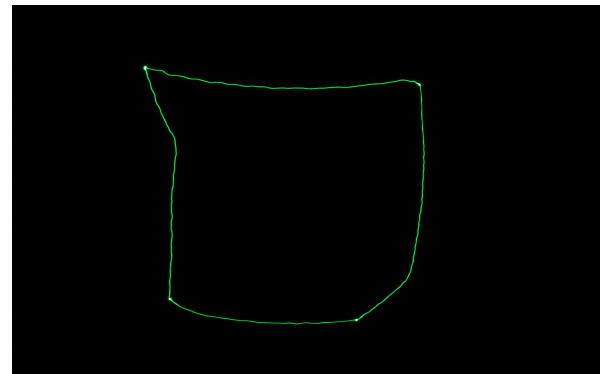


Abbildung 6. Ergebnisbild

In Zukunft, wenn man andere Materialien als das LEGO-Mindstorm-Set zur Verfügung hat, kann man die ganze Konstruktion stabiler und die einzelnen Komponenten einfacher verbauen. Weil das Metallgerüst nicht LEGO kompatibel ist, ist alles sehr leicht verrutscht. Ein weiteres Problem, welches in der Zukunft durch die Abschaffung der LEGO Bauteile einfach gelöst werden kann, ist die limitierte Kabellänge zwischen den Motoren und dem NXT-Stein. Abschließend kann man behaupten, dass die Idee, das Lightpainting einfacher zu gestalten und zu automatisieren, funktioniert hat, aber trotzdem definitiv noch verbessert werden kann.

LITERATURVERZEICHNIS

- [1] WIKIPEDIA: *Light Painting*. https://de.wikipedia.org/wiki/Light_Painting. Version: August 2021
- [2] WIKIPEDIA: *Langzeitbelichtung*. <https://de.wikipedia.org/wiki/Langzeitbelichtung>. Version: November 2021
- [3] MATHWORKS: *Gaussian Filter*

Stationärer Raumscanner mit Abstandssensoren

Steffen Bach, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des LEGO-Mindstorms-Praktikums wurde das Scannen von Räumen und die Darstellung der dazugehörigen Grundrisse erforscht. Dazu wurde ein Roboter gebaut, der mit Hilfe von Ultraschall- und Lasersensoren eine 360° Abstandsmessung im Raum durchführt und die Messwerte durch Verwenden der Eulerschen Formel in einem Koordinatensystem darstellt. Dieses Funktionsprinzip hat viele Anwendungsmöglichkeiten und kann in einfachster Form beispielsweise einem Staubsaugerroboter bei der Orientierung im Raum helfen.

Schlagwörter—Grundriss, kabellos, Lasersensor, LEGO-Mindstorms, Raumscanner, Ultraschallsensor

I. EINLEITUNG

FÜR den Menschen ist die Fortbewegung im Raum und das Ausweichen vor Hindernissen, Wänden oder ähnlichem keine anspruchsvolle Fähigkeit, obwohl hierfür sehr komplexe Mechanismen benötigt werden. Die gesamte Komplexität offenbart sich erst, wenn man einem Roboter beibringen will, sich autonom fortzubewegen. Das primäre Problem ist es, dem Roboter ein Gefühl für den Raum und dessen Grundriss zu vermitteln. Ein Mensch erfasst den Grundriss eines Raumes durch seinen Sehsinn und seinen Tastsinn. In der Dunkelheit verlässt sich der Mensch auf den Tastsinn und tastet sich die Wand entlang, bis bei ihm ein grober Grundriss des Raumes entsteht. Das Gleiche kann auch ein Roboter, der mit Tastsensoren ausgestattet ist und die Wand entlangfährt. Bei ausreichendem Licht orientiert sich der Mensch mit seinem Sehsinn und kann den Grundriss eines Raumes von einer Position aus erkennen. Ein Roboter benötigt für diese Art der Grundrisserfassung einen Sensor, der den Abstand zu Objekten und Wänden messen kann. Dafür können Abstandssensoren mit Ultraschallwellen, elektromagnetischen Wellen oder Laserstrahlen genutzt werden. Die beiden letzten Messsensoren werden bei Radaren (Radiowave Detection And Ranging) und Lidaren (Light Detection And Ranging) eingesetzt. In diesem Projekt werden sowohl Ultraschall- als auch Lasersensoren zur Grundrisserfassung verwendet. Diese Technologie kann beispielsweise einem Staubsaugerroboter bei der Orientierung im Raum helfen oder im Bereich der Virtual Reality den virtuellen Raum auf den tatsächlichen Grundriss begrenzen und so Zusammenstöße mit Wänden vermeiden.

II. VORBETRACHTUNGEN

Solche Technologien zur Grundrisserfassung existieren bereits, aber haben auch ihre Nachteile.

DOI: 10.24352/UB.OVGU-2022-022

Lizenz: CC BY-SA 4.0

A. Beispiel Staubsaugerroboter

Laut einem Staubsaugerroboter-Ratgeber [1] nutzen Staubsaugerroboter aus verschiedenen Preissegmenten sehr unterschiedliche Methoden zum Abfahren des gesamten Grundrisses. Die billigsten Modelle belassen es bei einer zufällig generierten Fahrtstrecke, die bestimmte Flächen mehrmals abfährt und bei verwinkelten Räumen in Eckbereichen endlos umherkreist. Dies ist zwar sehr ineffizient und zeitintensiv, aber dafür spart man sich weitere Sensoren und teure intelligente Software. Modelle im mittleren Preissegment fahren an der Wand entlang, speichern sich die Begrenzung und fahren dann innerhalb dieses Umrisses. Dies vermeidet ein Verirren des Roboters, aber die Hauptfläche des Raumes wird trotzdem in zufälligen Mustern abgefahren. Vereinzelt Modelle des höchsten Preissegments wie ein Modell der Marke Roborock [2] nutzen ein Lidar-System zur Erstellung des Grundrisses und zusätzlich intelligente Software, die die effizienteste Fahrtstrecke ermittelt. Klare Vorteile sind die Effizienz und die Zeitersparnis, aber wie bereits erwähnt, ist diese Technologie sehr kostenintensiv. Eine mögliche Lösung ist die Grundrisserfassung mit einem externen Raumscanner, der die Daten an die Software des Staubsaugerroboters weiterleitet und somit das Lidar-System im Roboter erspart.

B. Funktionsprinzip dieses Raumscanners

Der Raumscanner dieses Praktikums ist ein im Raum zentral stationierter Abstandssensor, der um seine eigene Achse gedreht wird. In Abbildung 1 sieht man, wie die Messwerte durch Rotation nacheinander aufgenommen werden. Der Abstand zu den Objekten wird aus der Hin- und Rückflugzeit der ausgesendeten Ultraschallwellen berechnet. Der verbaute Ultraschallsensor der Marke LEGO-Mindstorms misst laut dem Datenblatt im Bereich von 0 cm bis 255 cm mit einer Messgenauigkeit von ± 3 cm. Daraus ergibt sich für den Raum eine Limitation, da die Wände nicht weiter als 255 cm vom Raumscanner entfernt sein dürfen.

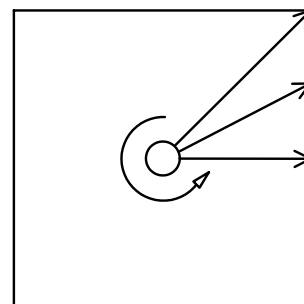


Abbildung 1. Messvorgang mit zentral stationiertem Scanner

III. ENTWICKLUNGSPROZESS

Der Entwicklungsprozess gliedert sich in mehrere Abschnitte, die hier sinnvoll abgetrennt und chronologisch dargestellt werden.

A. Roboteraufbau

Der Roboter wurde mit einem Ultraschallsensor ausgestattet, der auf einer Drehplattform stationiert ist. Ein LEGO-Motor dreht diesen Sensor 360° um seine Achse. Diese Motoren haben eine Drehgenauigkeit von $\pm 1^\circ$ und durch eine eingebaute Zahnradübersetzung von 7:1 wurde diese Abweichung auf ein Siebentel, also $\pm 0,15^\circ$ reduziert. Durch die Biegsamkeit der einzelnen Bauteile ist aber von einer geringeren Genauigkeit auszugehen.

B. User Interface

Zur Steuerung des Roboters und dem Aufnehmen der Messdaten wurde mit dem in MATLAB eingebauten App-Designer ein User Interface entwickelt, welches folgende Aufgaben übernimmt:

- Initialisierung und Deinitialisierung des NXT-Steins
- Rotation des Ultraschallsensors mit Gradeingabe
- Scannen der Umgebung mit eingegebener Anzahl an Messpunkten
- Verarbeitung und Ausgabe der Messpunkte im Koordinatensystem

C. Aufnahme der Messdaten

Das in der User Interface eingebaute Programm erstellt aus der vom Nutzer eingegebenen Anzahl an Messpunkten zwei Vektoren dieser Länge. Die zwei Vektoren werden benötigt, da die Messpunkte in der Eulerschen Formel mit Winkel und dazugehörigem Betrag aufgenommen werden. Im ersten Vektor werden die Winkel im Bereich von 0 bis $2 \cdot \pi$ mit gleichem Abstand eingetragen. Anschließend dreht der Motor den Ultraschallsensor um 360° gegen den Uhrzeigersinn und stoppt bei jedem Winkel, der im Winkelvektor eingetragen wurde. Während der Stopps misst der Ultraschallsensor den Abstand zur nächstliegenden Wand und das Programm speichert den Messwert im Betragsvektor zum entsprechenden Winkel.

D. Darstellung im Koordinatensystem

Zur Darstellung der Messdaten im Koordinatensystem wird der Betragsvektor und der Winkelvektor elementweise in die Eulersche Formel gebracht und geplottet. Die Verzerrung der Darstellung lässt sich durch eine einheitliche Achseneinteilung beider Achsen vermeiden. Die Ausgabe der ersten Messung kann man in Abbildung 2 sehen.

E. Auswertung der Ultraschallmessung

Wie die Abbildung 2 vermuten lässt, war der Grundriss des Raumes während der Messung ein gleichseitiges Dreieck. Aus der Abbildung 2 geht allerdings nicht hervor, dass die Wände geradlinig und die Ecken lückenlos waren. Somit ergeben sich bei der Ultraschallmessung zwei signifikante

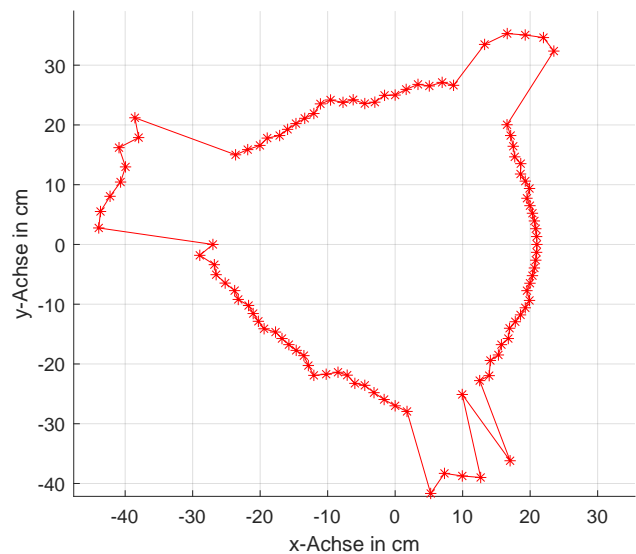


Abbildung 2. Messung mit Ultraschallsensor, gleichseitiges Dreieck als Grundriss

Probleme. Das erste Problem ist das seltsame Verhalten der Messwerte in Ecken, welches wir durch den sehr flachen Einfallswinkel der Ultraschallwellen erklären. Außerdem könnte auch die Interferenz von reflektierten Ultraschallwellen ein Grund für dieses Verhalten sein. Das zweite Problem ist die starke Wandkrümmung. Zum leichteren Verständnis der Ursache dient die Abbildung 3. Dort kann man nachvollziehen, dass der Ultraschallsensor durch die breite Schallkeule bei schräg stehenden Wänden eine kürzere Entfernung misst, als beabsichtigt. Konkret in Abbildung 3 ist der Betrag des schwarzen Pfeils gesucht und als Messwert wird der Betrag des roten Pfeils zurückgegeben. Diese Verkürzung von Abständen führt zu einer Krümmung, die mit flacherem Einfallswinkel verstärkt wird. Zur Behebung der Probleme wurde der Ultraschallsensor durch einen Lasersensor ersetzt.

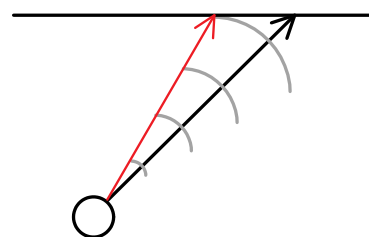


Abbildung 3. Messungenauigkeit aufgrund von breiter Schallkeule, Wandkrümmung als Folge

F. Problemlösung durch Lasersensor

Ein Lasersensor funktioniert auf einem ähnlichen Prinzip wie ein Ultraschallsensor. Statt Ultraschallwellen werden aber Photonen ausgesendet und aus deren Hin- und Rückflugzeit der Abstand zu Objekten berechnet. Der verwendete Lasersensor der Marke CQRobot [3] misst im Bereich von 4 cm bis 400 cm mit einer Genauigkeit von $\pm 5\%$. Der wesentliche

Vorteil gegenüber dem Ultraschallsensor ist eine deutlich kleinere Messkeule, wodurch das Problem der Wandkrümmung gelöst werden sollte. Allerdings ergibt sich ein neues Problem aus der Verwendung des Lasersensors, denn die Messwerte lassen sich nicht so einfach in MATLAB importieren. Als Schnittstelle dient hier ein Mikrocontroller (Modell: ESP8266), der mithilfe eines Arduino Programms die Messwerte des Lasersensors abrufen und an MATLAB weiterleitet. Der bereits in Abbildung 2 vermessene Grundriss wurde mit dem Lasersensor erneut vermessen. Die Ausgabe kann man in Abbildung 4 sehen. Es lässt sich eine klare Verbesserung erkennen, da die Wände sehr geradlinig abgebildet werden und die Messpunkte in den Ecken nicht ausbrechen. Das Problem mit dem sehr flachen Einfallswinkel in den Ecken lässt sich, aber selbst mit dem Lasersensor nicht vollkommen lösen und verbleibt als Abrundung.

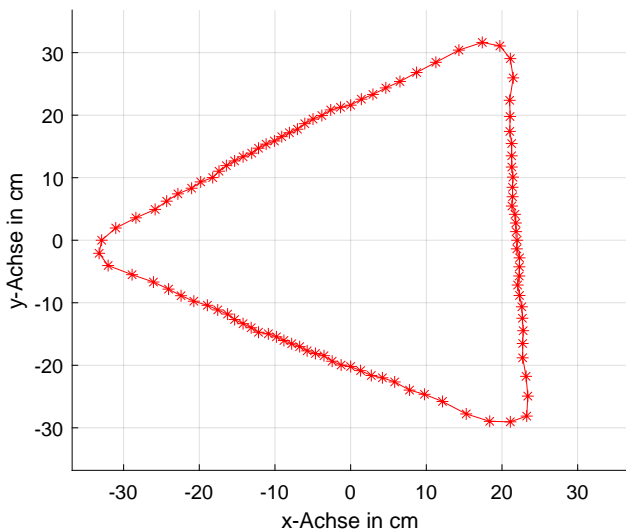


Abbildung 4. Messung mit Lasersensor, gleichseitiges Dreieck als Grundriss

G. Ausbaumöglichkeiten

Bis hierhin wurde die Vermessung eines Raumes nur durch eine Messstation betrachtet. Natürlich kann die Vermessung mit weiteren Messstationen ausgebaut werden. In Abbildung 5 sieht man eine beispielhafte Vermessung mit zwei Messstationen. Die erste Messstation stellt ihre Messpunkte wie gewohnt im Koordinatensystem dar und alle Messpunkte der zweiten Messstation werden um die Entfernung zur ersten Messstation verschoben. Die in Abbildung 5 gelb dargestellten Messpunkte liegen außerhalb des Messbereichs und werden daher gelöscht. Die in Rot dargestellten Messpunkte entsprechen in der Realität nur einem Punkt und werden daher zu einem mittig liegenden Punkt zusammengefasst. Die ursprünglichen Punkte werden gelöscht. Dies führt zu einer höheren Messgenauigkeit im Schnittbereich beider Messstationen und zum Ausweiten des maximal messbaren Raumes. Am besten lässt sich eine solche Vermessung durch einen mobilen Roboter realisieren, der von einer Position zur nächsten Position selbstständig fährt und den gefahrenen Weg zu all seinen Messpunkten hinzurechnet.

Voraussetzung hierfür ist eine kabellose Ansteuerung des Roboters, die in unserem Fall durch einen Raspberry Pi Zero WH [4] realisiert wurde. Auf dem Raspberry Pi läuft ein Server, der über das WLAN mit der Computersoftware VirtualHere kommuniziert. Im Prinzip überträgt diese Schnittstelle nur Daten von einem USB-Port zu einem anderen USB-Port über das WLAN.

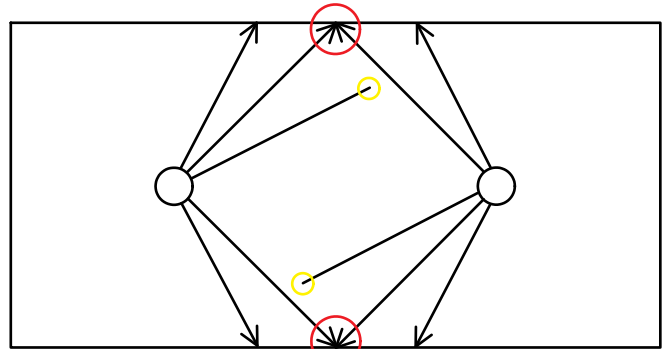


Abbildung 5. Messvorgang mit zwei Messstationen

IV. ERGEBNISDISKUSSION

Das Endergebnis dieses Projektes, welches in Abbildung 6 zu sehen ist, ist ein stationärer Raumscanner, der mit Hilfe eines Lasersensors einen Raum vermessen und dessen Grundriss in einem Koordinatensystem darstellen kann. Die Messung ist durch den Lasersensor ziemlich genau und sogar bei glatten durchsichtigen Objekten wie Fenstern funktionsfähig. Nur das Messverhalten mit Spiegeln wurde nicht überprüft. Der Raumscanner wird über eine Powerbank mit Strom versorgt und lässt sich komplett kabellos vom Computer über eine MATLAB-App steuern.

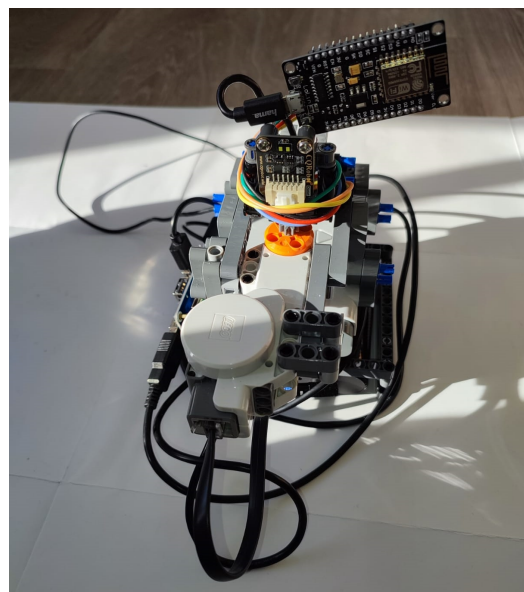


Abbildung 6. Raumscanner mit Lasersensor und kabelloser Ansteuerung

V. ZUSAMMENFASSUNG UND FAZIT

Das Ziel des Projektes war es, einen Roboter zu bauen, der einen Raum einscannen und den entsprechenden Grundriss darstellen kann. Die Umsetzung dieses Roboters ist für eine Messstation gelungen. Beim Scannen von Räumen ohne Möblierung ließ sich nur das Problem mit der Abrundung spitzwinkliger Ecken erkennen. Das Scannen von möblierten Räumen wurde nicht ausreichend getestet. Außerdem wurden einige Grundlagen für zukünftige Projekte mit mehreren Messstationen erarbeitet. Dazu zählen eine kabellose Ansteuerung des Roboters und ein Prinzip zur Punktüberlagerung. Es fehlt ein fahrbarer Untersatz mit Messung der gefahrenen Strecke und die autonome Auswahl einer angemessenen Position für eine weitere Messstation. Eventuell lässt sich auch die Dauer der Scans minimieren, sodass man sich bewegende Objekte in Echtzeit im Koordinatensystem darstellen kann.

LITERATURVERZEICHNIS

- [1] FRANK BRALL: *Orientierung und Fahrstrategien*. <https://www.aku-und-roboter-staubsauger.de/ratgeber/orientierung-im-raum/>. Version: 2015
- [2] ROBOROCK: *Roborock S6-Reihe*. <https://www.aku-und-roboter-staubsauger.de/ratgeber/orientierung-im-raum/>. Version: 2020
- [3] CQROBOT: *Time-of-Flight (ToF) Long Distance Ranging Sensor*. https://www.cqrobot.com/index.php?route=product/product&product_id=1110. Version: 2019
- [4] BERRYBASE: *Raspberry Pi Zero WH*. <https://www.berrybase.de/raspberry-pi/raspberry-pi-computer/boards/raspberry-pi-zero-wh>. Version: 2018

Automatisierung einer Kaffeemaschine

Tim Holzki, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des LEGO-Mindstorms-Praktikums 2022 wurde den Studierenden die Möglichkeit gegeben, unter Verwendung von zur Verfügung gestellten LEGO-Mindstorms Bausätzen eigene Problemstellungen zu finden und mittels dieser solche zu lösen. Hierbei wurden grundlegende Fähigkeiten der effizienten Projektbearbeitung, Programmierung und praktischen Konstruktion vermittelt und aufgebaut. Als Programmierumgebung wurde MATLAB verwendet, welche über eine Toolbox mit der Hauptrecheneinheit, dem NXT-Stein, kommuniziert und die Ansteuerung von Sensorik und Motorik erlaubt. In dem hier behandelten Projekt wurde sich dazu entschlossen die Funktion einer Kaffeemaschine insoweit zu automatisieren, als dass mittels des Baukastens die benötigten Ressourcen – Wasser und Kaffeebohnen – auf ausreichende Verfügbarkeit überprüft werden und der Brühvorgang automatisiert gestartet wird. Im Verlauf, des über zwei Wochen laufenden Projektes, wurden zudem von jedem Team eigene Social-Media Präsenzen gepflegt, um über die reine Entwicklungsarbeit hinaus auch die Öffentlichkeitskommunikation der eigenen Arbeit zeitgemäß zu leisten.

Schlagwörter—Elektrotechnik, Klemmbausteine, Konstruktion, Maschinenbau, Programmierung, Projektarbeit

I. EINLEITUNG

DAS Automatisieren von Haushaltsgeräten, wie Spül- und Waschmaschinen, Mikrowellen u. ä. über das IoT ist innerhalb des letzten Jahrzehnts immer mehr zu einem fast schon erwartbaren Standardfeature an eben solchen Geräten im höheren Preissegment geworden. Es versteht sich jedoch für die meisten Studierenden, dass diese Geräte als finanzielle Fehlentscheidungen zu werten sind und somit andere günstigere technische Lösungen für die örtliche Trennung von Haushaltsgeräten und der eigenen Person gefunden werden müssen. Infolgedessen wurde sich durch die Bereitstellung der bereits erwähnten LEGO-Sets die Aufgabe gestellt die eigene Kaffeemaschine ferngesteuert zu automatisieren. Hierfür war es im ersten Schritt erforderlich, die ansonsten händisch zu bedienenden Elemente, wie das Bedienfeld mittels einer motor-gesteuerten Konstruktion zu manipulieren und ohne weiteres Eingreifen ansteuern zu können. Des Weiteren war es nötig, die Hauptressourcen, welche für einen Brühvorgang notwendig sind, also Kaffeebohnen und Wasser über unterschiedlichen Sensoren zu überwachen um eine Beschädigung der Maschine oder einen Abbruch des Brühens zu vermeiden.

II. VORBETRACHTUNGEN

Das Problem der Füllstandsüberwachung von festen und flüssigem Material ist in vielen Bereichen der Industrie gegeben. Hieraus folgen etablierte gut nachbildbare Lösungen für die Sensorik der Überwachung von Wasser und Bohnen. Die Ansteuerung von eigentlich für den Menschen bestimmten

Bedienflächen mittels Roboter kann in der Qualitätskontrolle zur Überprüfung der Qualität von Schaltflächen o. Ä. gefunden und teilweise übernommen werden. Das Zusammenspiel dieser Komponenten wird über die gestellte MATLAB Umgebung realisiert.

A. Bedienfeld

Die Qualitätsüberprüfung von Schaltern oder anderen Schnittstellen, die vom Menschen bedient werden wird in der Industrie meist mittels, eigens für diesen Zweck gebauter, Teststände überprüft, so auch in [1]. Hier werden Schalter aus dem Cockpit eines Kfz in relativer kurzer Zeit auf die Anzahl ihrer maximalen Schaltvorgänge überprüft, daher ist es auch hier erforderlich Maschinen zu entwerfen, die eine eigentlich nur für den Menschen konzipierte Schnittstelle zuverlässig bedienen können. In diesem Projekt soll das Bedienfeld ebenfalls als eine rein für den Menschen entworfene Schnittstelle von einer Maschine bedient werden.

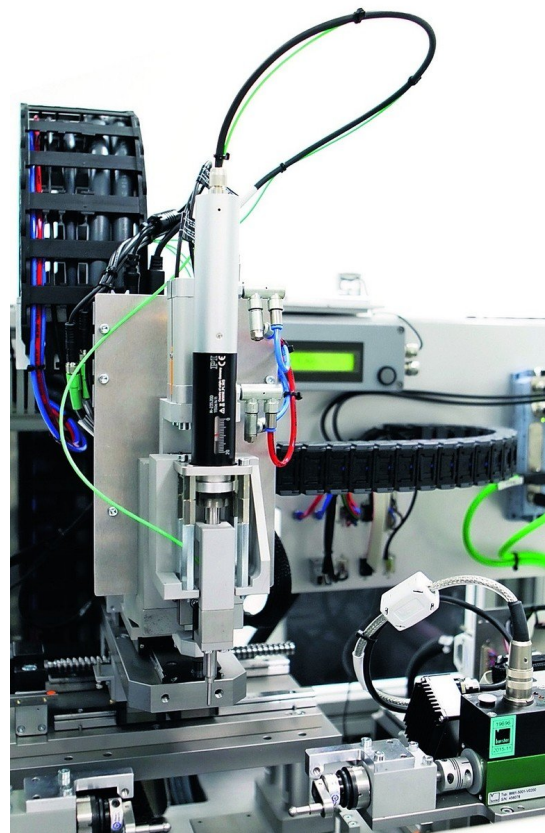


Abbildung 1. Teststand für Belastungstests von Kfz-Schaltern; Bild entnommen aus [1]

B. Wasservorrat

Die Messung eines Füllstandes wird in der Industrie oft mittels Ultraschallsensoren durchgeführt, so auch in [2]. Hier werden zudem die Vorteile bei der Füllstandsmessung von Flüssigkeiten mittels Ultraschall aufgezeigt. So muss der Sensor nicht in Kontakt mit der zu messenden Substanz kommen und unterliegt keiner Beeinflussung durch die Eigenschaften des zu messenden Stoffes, wie bei einer gewichtsbasierten Messung durch Temperaturunterschiede hervorgerufene Dichteänderungen, der Fall wäre. Der mitgelieferte Ultraschallsensor in dem NXT-Set ist also prädestiniert, hier zum Einsatz zu kommen.

C. Bohnenvorrat

Das Erkennen von einzelnen Elementen inmitten großer Mengen ist im großtechnischen Format meist Aufgabe von Sortiermaschinen, als Beispiel soll hierfür [3] dienen. Diese Geräte erfassen in einer Unzahl von Körnern, Früchten o. Ä. Werte wie Feuchtigkeitsgehalt, Geometrie, Mängel und verschiedenste stoffliche Eigenschaften des Sortierguts über Laser und Kameras. Wenn auch die im bereitgestellten Baukasten vorhandenen RGB-Sensoren weitaus nicht so detaillierte Analysen zulassen, so arbeiten sie doch nach dem selben grundsätzlichen Prinzip. Aus rein optisch aufgenommenen Messdaten, welche nicht intrusiv ermittelt wurden, sind Schlüsse auf das zu beobachtende Objekt zu ziehen. Hiermit kann also in seiner einfachsten Abwandlung auch schlicht das Vorhandensein eines Objektes, hier den Kaffeebohnen, festgestellt werden.



Abbildung 2. Sortiermaschine ausgestattet mit Laser und Kameras; Bild entnommen aus [3]

D. Softwaretechnische Umsetzung

Die einzelnen, bereits betrachteten Komponenten müssen, um ein funktionierendes Gesamtsystem zu ergeben, über eine Schnittstelle miteinander verbunden werden. Hierfür wird in MATLAB die RWTH-Toolbox [4] benutzt, die die Kommunikation über USB-Kabel oder Bluetooth zwischen NXT-Stein und Computer erlaubt. Zwar existiert von LEGO eine mitgelieferte, benutzerfreundlichere Programmierumgebung, jedoch erlaubt diese nicht, die sehr viel detaillierte Steuerung und Auswertung der NXT-Schnittstellen, wie die hier benutzte. Über MATLAB wird ebenfalls die graphische Benutzeroberfläche (GUI) erstellt, die die direkte Ansteuerung und Auswertung der Sensoren und Motoren zugänglicher macht.

III. HARDWARE-ENTWICKLUNG

Der grundlegende Programmablauf, gezeigt in Abbildung 3, legt das Vorgehen für die geschriebene Software fest. Nachdem das Programm gestartet wurde, gibt den Nutzenden die gewünschte Anzahl an Tassen und Stärke seines Kaffees an. Wird nun der Start des Brühvorganges initialisiert, werden die Sensoren für Bohnen und Wasser ausgelesen. Sollte hier ein Mangel festgestellt werden, gibt das GUI dies an den Bedienenden zurück und wird in den Ausgangszustand versetzt. Der Brühvorgang, also die Ansteuerung der Motoren findet erst statt wenn sowohl genug Wasser, als auch Bohnen vorhanden sind. Je nachdem wie viele Tassen und welche Stärke am Anfang ausgewählt wurden, wird daraufhin eine Sequenz von Motorenbefehlen für eben diese bestimmte Kombination an die Bedienfeldsteuerung weitergeleitet und somit die Maschine gestartet.

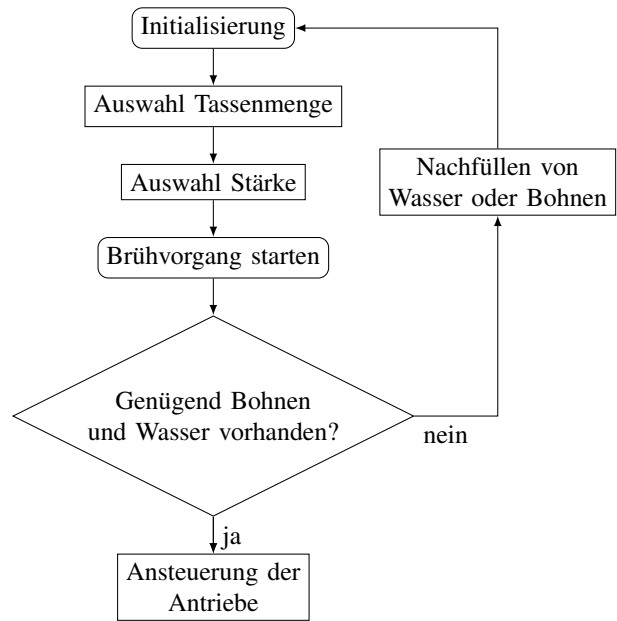


Abbildung 3. Grundlegender Programmablaufplan zur Kaffeemaschinenautomatisierung

A. Sensoraufbau Bohnenvorrat

Um die Bohnen wie in Abbildung 5 zu erkennen, wurde sich nicht auf eine direkte Messung des RGB-Farbwertes der Bohnen verlassen. Das Problem in dieser Herangehensweise liegt darin, dass verschiedene Sorten von Kaffeebohnen unterschiedliche Farbtöne besitzen und diese bei schlechter Ausleuchtung nicht mehr mit dem Sensor erkannt werden können. Hieraus folgte der Ansatz einer Helligkeitsbestimmung im Bohnenvorrat. Die generell dunkle Farbe der Bohnen kann hierbei durch ein untergelegtes weißes Blatt weiter kontrastiert werden (siehe Abbildung 5), um eine sicherere Messung zu garantieren. In MATLAB kann nun ein ausgewählter Helligkeits-Grenzwert bestimmt werden, ab dem bestätigt wird, dass ausreichend Bohnen verfügbar sind.



Abbildung 4. Bohnenvorrat mit angebautem RGB-Sensor

B. Sensoraufbau Wasservorrat

Der Ultraschallsensor wurde über dem Wasserbehälter angebracht und misst die zeitliche Verzögerung mit der ein ausgesendetes Ultraschallsignal wieder bei ihm empfangen werden kann. Hieraus lässt sich nun die Entfernung zur messenden Oberfläche, also dem Wasser errechnen. Nachdem der Sensor auf seine endgültige Position ausgerichtet war, wurde der vom Sensor ausgegebene Entfernungswert für den Füllstand korrespondierend zu der kochbaren Anzahl Tassen notiert und im MATLAB Code für die Auswahlfunktion abgespeichert. Somit wird überprüft, ob der Wasserstand den nötigen Füllstand für eine bestimmte Anzahl Tassen unterschreitet oder nicht, und je nachdem ausgehend von Abbildung 3 fortgefahren.



Abbildung 5. Wasservorrat mit angebautem Ultraschall-Sensor

C. Aufbau Bedienfeldsteuerung

Um das Bedienfeld (siehe Abbildung 7) mit Hilfe der Motoren anzusteuern wurde eine Konstruktion (siehe Abbildung 6) gewählt, die sich zweier drehender Hebel, einer davon in Zahnradform, bedient. Diese werden über eine Untersetzung von den Motoren angetrieben und bewegen dabei abgerundete Fortsätze über das Bedienfeld, was zu einem Eindringen der Taste und somit der Auswahl einer Funktion führt. Die Untersetzung ergab sich zum einen aus den geringen Dimensionen, welche der Aufbau besitzen musste um ein Einschleiben der Kaffeekanne weiter zu ermöglichen, was die direkte Anordnung der Motoren nebeneinander nicht praktikabel machte und sie somit ein ineinander versetzter Bau erforderlich wurde. Zum Anderen stellte sich heraus, dass die Winkelpositionierung der Motoren nicht ausreichend genau war, um die Tasten wiederholt präzise zu treffen und somit die Untersetzung den Fehler auf eine längere rotatorische Strecke verteilte. Das obere Zahnrad wählt hier sowohl die Tassen als auch die Starttaste aus, während der untere Hebel die Stärke des Kaffees bestimmt.

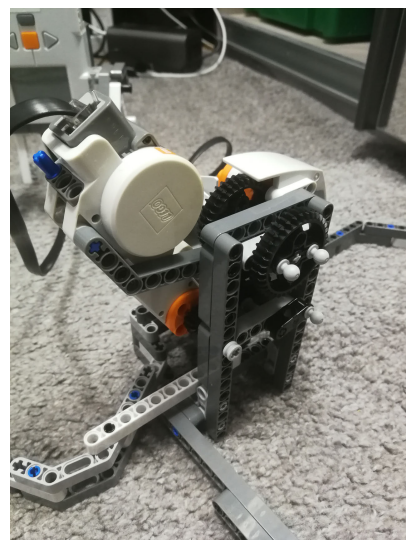


Abbildung 6. Mechanik zur Manipulation des Bedienfeldes



Abbildung 7. Zu manipulierendes Bedienfeld

D. GUI

Das in MATLAB gebaute GUI (siehe Abbildung 8) beinhaltet die Funktionen der Auswahl von Tassenanzahl als auch Stärke, darüber hinaus werden im rechten Teil Rückmeldungen zur Verfügbarkeit von Wasser und Bohnen gegeben.

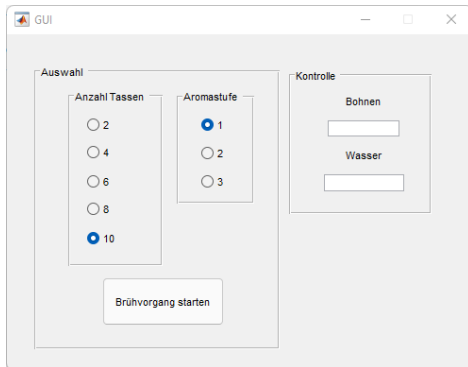


Abbildung 8. GUI mit verschiedenen Auswahl- und Stuselementen

IV. ERGEBNISDISKUSSION

Die Maschine funktionierte mit Anbauten (siehe Abbildung 9) einwandfrei. Die Sensoren lieferten ausreichend genaue und reproduzierbare Ergebnisse. Ebenso zeigte sich die Motorsteuerung unter Zuhilfenahme der Untersetzung als sehr genau. Probleme können jedoch in der Bequemlichkeit des Nachfüllens gefunden werden. So ist es nicht möglich Wasser mit dem aufgesetzten Ultraschallsensor nachzufüllen, dieser muss hierfür abgenommen werden. Außerdem ist die Beschaffenheit der Oberfläche auf der die Maschine steht zu beachten, da die Befestigungsarme, welche die Bedienfeldsteuerung an das Gehäuse drücken, eine raue Unterlage benötigen, um hierfür ausreichend Kraft aufzubringen.



Abbildung 9. Zusammengebaute automatisierte Kaffeemaschine

V. ZUSAMMENFASSUNG UND FAZIT

Wenn auch die Umsetzung der Fernsteuerung aufgrund veralteter Bluetooth-Standards nicht umgesetzt werden konnte, so ist die Automatisierung soweit als erfolgreich zu sehen, als dass die Maschine eigenständig ihre Bereitschaft zum Kochen einer gewünschten Menge Kaffee erkennt und den Vorgang vorhersehbar durchführt. Jedoch ist die Gesamtkonstruktion durch die verwendete RWTH-Toolbox auf eine USB-Verbindung mit einem PC angewiesen. Es wäre also zu überlegen die Programmierung in der LEGO internen Umgebung zu wiederholen und somit den NXT-Stein als auch den gesamten Aufbau unabhängig von einem ausführenden Computer nutzen zu können. Weitere Aufgaben für die Zukunft wären darüber hinaus die Befestigung des Ultraschallsensors zu überarbeiten, um ein nachträgliches Einfüllen, ohne Abnehmen des Sensors, zu ermöglichen. Die Sicherung der Bedienfeldsteuerung ist ebenfalls im jetzigen Zustand noch nicht ausgereift genug für eine alltägliche Nutzung. Es wäre auch hier vorstellbar die momentan verwendeten Befestigungsarme durch andere Maßnahmen zu ersetzen, die einen sichereren und einen von den Umgebungsbeschaffenheiten unabhängigeren Halt garantieren würden. Hier würden sich auch Komponenten, welche nicht im LEGO-Katalog enthalten sind, anbieten, wie z. B. Klebepads, Kabelbinder oder auch den Befestigungsarmen ähnliche Konstruktionen aus Metall die weniger Spiel und Flexibilität zulassen würden.

LITERATURVERZEICHNIS

- [1] TATJE, Joachim: *Kfz-Schalter im Härtetest*. <https://quality-engineering-industrie.de/branchen/automobil/kfz-schalter-im-haertetest/#slider-intro-2>. Version: 2017
- [2] ENDRESS+HAUSER: *Ultraschall Füllstandsmessung*. <https://www.de.endress.com/de/messgeraete-fuer-die-prozesstechnik/fuellstandssensor/ultraschall-fuellstandsmessung>
- [3] TOMRA: *Kaffesortiermaschinen von TOMRA*. <https://www.tomra.com/de-de/sorting/food/your-produce/grains-and-seeds/coffee>
- [4] ATORF, L. ; SONDERMANN, B. ; STADTMANN, T. ; ROSSMANN, J.: *RWTH - Mindstorms EV3 Toolbox*. <https://git.rwth-aachen.de/mindstorms/ev3-toolbox-matlab>. Version: 2018

Farbsortierroboter

Badr Abdi, ETIT

Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—In diesem Jahr findet wieder das LEGO-Mindstorms-Projektseminar der Otto-von-Guericke Universität Magdeburg statt. Wie seit 2013 bauen, programmieren und realisieren Studierende unterschiedliche Roboter aus LEGO auf, die am Ende dokumentiert und vor einer Jury vorgestellt werden. Zu den Robotern des Jahres gehört der Farbsortierroboter, mit dem sich in diesem Paper befasst wird. Der Farbsortierroboter ist in der Lage, die kleine Kugeln nach Farbe zu sortieren und in die entsprechenden Behälter aufzuräumen. Der Farbsortierroboter warf einige Konstruktions- und Programmierprobleme auf. Diese Probleme und ihre Lösungen werden später in diesem Artikel behandelt.

Schlagwörter—Farbsortierer, LEGO, NXT 2.0, OVGU, Projektseminar, Roboter

I. EINLEITUNG

HEUZUTAGE können Roboter verschiedene und mehrere Tätigkeiten bzw. Aufgaben gleichzeitig erledigen und sich selbständig bewegen. Mit der Entwicklung der Industrie ersetzen Roboter die Arbeit der Menschen und werden dadurch immer wichtiger und essenzieller. Roboter können das Leben vereinfachen, die Produktivität erhöhen und sind in der Lage auch häusliches Arbeiten zu übernehmen, wie zum Beispiel Staubsaugroboter, die komplett eigenständig die Reinigung von Böden aller Arten übernehmen. Auch Aktivitäten im Sportbereich können mit moderner Technik verbessert werden, wie der Videoassistent im Fußball oder das Hawk-Eye im Tennis (Abb. 1). Sie zeichnen das Spiel auf und können die genaue Ballposition bestimmen, wenn Entscheidungen angezweifelt werden.

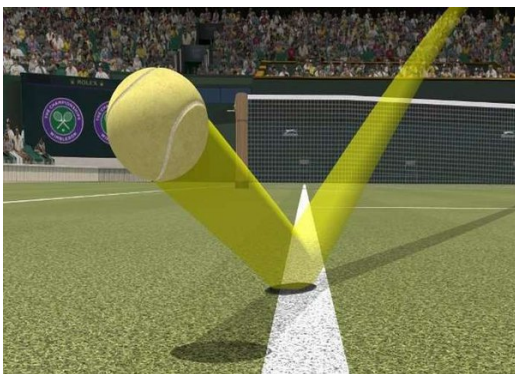


Abbildung 1. Hawk-Eye aus [1]

Der wichtigste Faktor der Roboter allgemein ist die Reduzierung bzw. Verringerung der menschlichen Arbeitskraft, und hier kommt der Farbsortierroboter ins Spiel. Das Hauptziel eines Farbsortierers ist das Sortieren bzw. Aufräumen nach

DOI: 10.24352/UB.OVGU-2022-025

Lizenz: CC BY-SA 4.0

Farbe. Der hier vorgestellte Farbsortierroboter kann Legokugeln sortieren und entsprechend der Farbe zählen, was an einem Bildschirm angezeigt wird. Ein praktisches Beispiel für einen Farbsortierer wäre das Aufräumen von Bowlingbällen nach der Farbe, um Chaos zu minimieren.

II. VORBETRACHTUNGEN

Der Farbsortierroboter besteht hauptsächlich aus einem Farbsensor und Motoren, damit er sortieren kann.

A. Idee und Prinzip

Die Idee des Projekts ist ein Roboter, der Legokugeln nach ihrer Farbe sortiert. Der Roboter soll zunächst eine Kugel von den anderen separieren, dann die Farbe über einen Sensor bestimmen und anschließend in den entsprechenden Behälter befördern. Der Prozess muss unendlich oft wiederholt werden, bis der Benutzer das Programm beendet. Das Prinzip ist sehr einfach und wird von großen Unternehmen verwendet, wie zum Beispiel bei Getreidereinigern, vor allem Bohnenreinigern (Abb. 2). Diese Maschinen lassen die unerwünschten farbige Bohnen nicht durch und entsorgt sie in andere Behälter. Beim Farbsortierroboter sind die Idee und das Prinzip das Gleiche. Bei Getreidereinigern kommen noch andere Faktoren hinzu, wie zum Beispiel die Genauigkeit oder Temperatur.

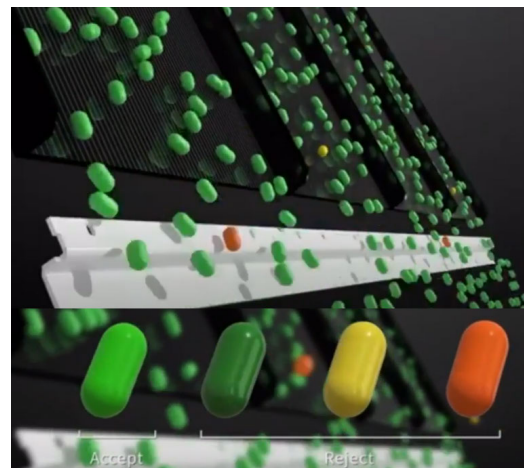


Abbildung 2. Getreidereiniger aus [2]

B. Einordnung der Kugeln

Die Kugeln sind auf einer statischen Rampe zufällig aufgereiht und rollen mittels Schwerkraft von selbst nach unten. Eine mechanische Barriere auf der Rampe separiert eine einzelne Kugel vom Rest und eine Barriere am Ende stoppt

die Kugel vom Herunterfallen. Ein motorisiertes Fahrzeug mit mehreren Behältern unter der Rampe dient zum Einsortieren und Auffangen. Die erste Idee war ein statischer Behälter mit einer rotierenden Rampe. Aber stellt es sich heraus, dass dadurch nur maximal 3 Behälter erreicht werden konnten.

III. KONSTRUKTION UND REALISIERUNG DES FARBSORTIERERROBOTERS

A. Aufbau

Für die mechanische Konstruktion des Roboters stehen zwei großen Legokästen zur Verfügung. Diese enthalten mehrere LEGO-Mindstorms-Sets inklusive eines NXT-Steins, drei Motoren und verschiedenen Sensoren. Um die Konstruktion des Farbsortierroboters zu verwirklichen, werden drei Motoren, ein Farbsensor, das NXT-Gerät und verschiedene Lego-Bausteine gebraucht. Der Farbsortierroboter besitzt zwei Teile. Der statische Teil ist mit dem NXT-Gerät verbunden und kann maximal 12 Kugeln gleichzeitig tragen. Die Kugeln können auch während des Programmablaufs eingesetzt werden. Wie in Abbildung 3 zu sehen ist, ist der bewegliche Teil ein Fahrzeug mit mehreren Behältern, das die Kugeln auffängt und auf drei Farben (Rot, Blau und Grün) aufteilt.



Abbildung 3. Aufbau des Farbsortierroboters [3]

B. Umsetzung

Der Farbsortierroboter besteht hauptsächlich und mechanisch aus drei Motoren und einem Farbsensor. Die mechanischen Bauteile können durch eine Schnittstelle angesteuert werden.

1) *Farbsensor*: Der RGB-Sensor von LEGO wird zur Erkennung der Farben verwendet. Der Sensor illuminiert das abzutastende Objekt in den Farben Rot, Grün und Blau. Der Sensor erfasst das reflektierte Licht für eine genaue Farberkennung. Je mehr das Objekt eine Farbe (Rot, Grün oder Blau) reflektiert, desto sicher wird sie erkannt. Der Sensor kann auch andere Farben wie Weiß (Rot, Grün und Blau stark reflektiert), Schwarz (keine Farbe reflektiert) oder Gelb (nur

Rot und Grün reflektiert) erkennen.

Im MATLAB muss erst der Farbsensor eingerichtet werden. Daran ist nur der Modus FULL interessant, da alle 6 Farben (Rot, Grün, Blau, Gelb, Weiß und Schwarz) erkannt und als Zeichenkette ausgegeben werden.

2) *Motoren*: Motoren sind sehr wichtige Bauteile des Farbsortierroboters, weil sie vielfältig einsetzbar sind. Der erste Motor, auch Startmotor genannt, bewegt die Barriere zur Separation der Kugeln durch Umwandlung der mechanischen Rotation in eine Linearbewegung (siehe Abbildung 4, links). Dieser Motor spielt eine essenzielle Rolle, da er nur eine Kugel bis zum nächsten Schritt passieren lässt.

Der zweite Motor, auch Barrieremotor genannt (siehe Abbildung 4, Mitte), hat die Aufgabe, die Kugel nach der Farberkennung zu blockieren, damit der entsprechende Behälter unter die Rampe bewegt werden kann.

Der dritte Motor, auch Transportmotor genannt (siehe Abbildung 4, rechts), soll die Behälter unter der Rampe bewegen und die Kugel auffangen.

Im MATLAB müssen die Motoren zunächst auch eingerichtet werden. Im NXT-Gerät gibt es Anschlüsse, die mit Buchstaben gekennzeichnet sind. Damit kann man die Motoren in MATLAB richtig definieren. In diesem Fall ist der Startmotor als `MotorA`, Barrieremotor als `MotorB` und Transportmotor als `MotorC` definiert. Die Motoren müssen so eingestellt werden, dass die erwünschte Kraft und Drehung konfiguriert sind. Dafür stehen die Befehle `MotorX.Power` und `MotorX.TachoLimit` zur Verfügung.



Abbildung 4. Die drei für den Farbsortierroboter verwendeten Motoren (links: Startmotor, mittig: Barrieremotor, recht: Transportmotor) [3]

C. GUI

Das System soll, nachdem alles funktioniert, mit einer GUI (graphical user interface) gesteuert werden (siehe Abbildung 5). Das System kann sowohl den Farbsortierroboter starten als auch den Vorgang stoppen. Außerdem werden die Kugeln nach Farbe gezählt und ihre Anzahl auf dem Bildschirm ausgegeben bzw. angezeigt. Die Gesamtanzahl aller Kugeln wird auch ausgegeben. Eine Möglichkeit zum Zurücksetzen der Anzahlwerte ist realisiert. In der GUI sind alle Tasten, von Start bis Exit, Drucktasten. Wenn man Start drückt, befindet sich das Programm in einer unendlichen Schleife, die nur mit Stop gestoppt werden kann. Das Zurücksetzen der ausgegebenen Anzahlwerte während der Programmablaufs ist leider nicht möglich, sondern nur nachdem das System gestoppt wurde.

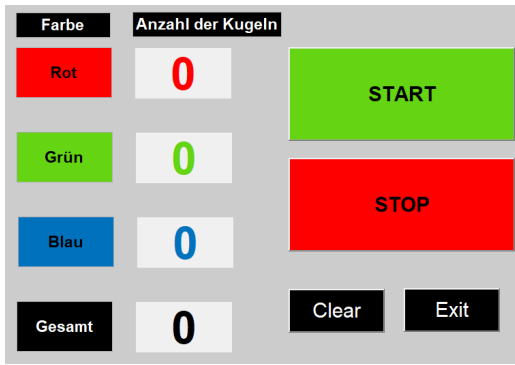


Abbildung 5. GUI des Farbsortierroboters [3]

D. Programmlaufplan

Ein beispielhafter Programmablaufplan zur Erklärung des Verfahrens ist in Abbildung 6 dargestellt.

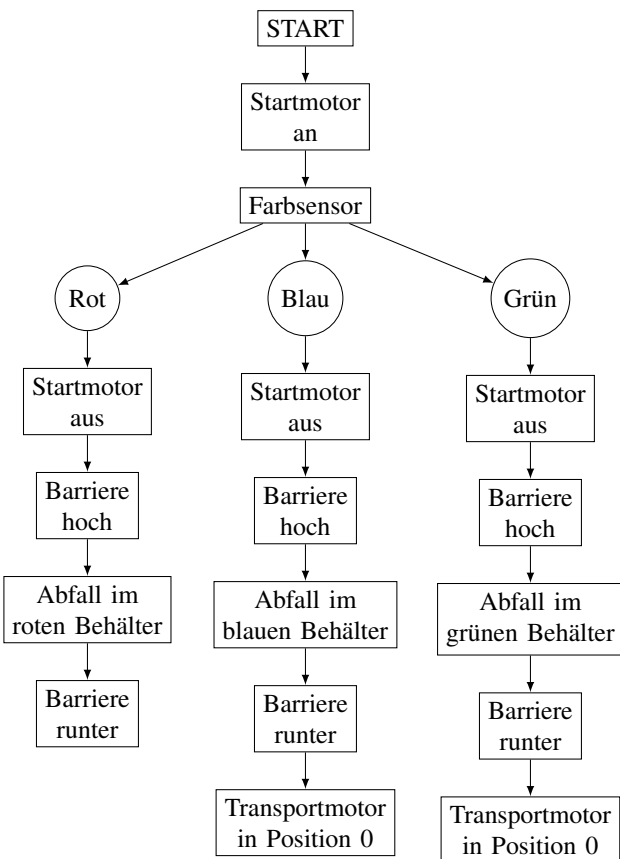


Abbildung 6. Programmablaufplan des Farbsortierroboters

E. Funktionsweise des Programms

Nach dem Drücken von Start schaltet der Startmotor ein. Der Startmotor ist mechanisch so aufgebaut, dass er nur eine Kugel durch lässt. Während die Kugel rollt, erkennt der Farbsensor die Farbe der Kugel und gibt Befehle aus. Für alle Farberkennung wird der Startmotor auf jeden Fall ausgeschaltet. Die Kugel befindet sich dann am Ende der Rampe, weil die Barriere ihren

Weg blockiert. Abhängig von der Kugelfarbe unterscheiden sich die nächsten Befehle:

Wenn der Farbsensor „Grün“ ausgibt, drehen sich die zwei Räder des Transportmotors für nur eine Rotation, was 360° entspricht. Nach der Bewegung des Transportmotors wird die Barriere gehoben und die Kugel fällt direkt in den grünen Behälter. Während des Prozesses zählt gleichzeitig das Programm, dass eine grüne Kugel detektiert wurde und der Anzahlwert der grünen Kugeln wird auf dem GUI-Bildschirm im grünen Feld angezeigt. Die Barriere geht wieder runter und der Transportmotor fährt zurück in Position 0, was -360° bedeutet.

Das gleiche Verfahren wird bei der Farbe „Blau“ durchgeführt. Der Unterschied liegt beim Fahrtabstand des Transportmotors. Während die Räder des Transportmotors sich bei „Grün“ um 360° drehen, bewegen sich die gleichen Räder bei „Blau“ nur um eine halbe Rotation, was 180° entspricht.

Wenn der Farbsensor „Rot“ ausgibt, muss das Programm weniger Befehle ausführen, weil der rote Behälter schon direkt unter der Rampe positioniert ist, Was dazu führt, dass der Transportmotor gar keine Befehle bekommt.

IV. ERGEBNISDISKUSSION

Das Endergebnis sieht positiv aus. Der Farbsortierroboter hat einwandfrei funktioniert. Natürlich traten einige Probleme während der Konstruktion und der Programmierung auf, die im Folgenden erwähnt werden.

Beim ersten Versuch der Separation der Kugeln trat das erste Problem auf. Die Geschwindigkeit des Startmotors war zu groß, sodass die Kugeln nacheinander von der Rampe flogen. Danach erkannte der Farbsensor aufgrund der Raumhelligkeit die Farbe nicht genau genug. Zur Lösung wurde eine Abdeckung darauf aufgebaut, um die Raumhelligkeit zu verringern. Ein Prozess für unbekannte Farben wie zum Beispiel Gelb, Weiß oder Schwarz sind im Programm nicht enthalten, weshalb diese Farben nicht sortiert werden. Ein anderes Problem tauchte beim Drücken der Stop-Taste in der GUI auf. Durch die while-Schleife wird immer wieder das Programm durchlaufen. Wenn die Stopp-Drucktaste betätigt wird, stoppt das System nur für der nächste Kugel, aber nicht für die detektierte Kugel.

V. ZUSAMMENFASSUNG UND FAZIT

Der Farbsortierroboter funktioniert am Ende gut. Wenn das Projektseminar länger als zwei Wochen gedauert hätte, wäre es möglich gewesen, komplexere Maschinen zu bauen, auch bei gleicher Anzahl zur Verfügung stehender Legosteine. Der Farbsortierroboter kann in anderen Faktoren verbessert werden und in größeren Dimensionen entwickelt werden.

ANHANG

Link zum Quellcode in [3]

LITERATURVERZEICHNIS

[1] Hawk-Eye. <https://qph.fs.quoracdn.net/main-qimg-c24e3874896bd4d570921ca291654267-lq>, . - 17.02.2022
 [2] Bildschirmaufnahme der Getreideeinermaschine. <https://www.youtube.com/watch?v=GlnqPxglbxA>, . - 19.02.2022
 [3] Quellcode und Bilder. <https://cloud.ovgu.de/s/wzGKjP8pbLnaxZk>, . - 03.03.2022

Sprachgesteuerter Getränkeautomat mit Lego Mindstorms

Etienne Grundmann, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Abstract—Im Rahmen des Projektseminars Elektrotechnik/Informationstechnik sollte ein Automat entwickelt werden, dessen Konstruktion und Programmierung auf Elementen von LEGO Mindstorms basiert. Hierbei handelt es sich um einen Getränkeautomaten mit erweitertem Funktionsumfang. Dieses Projekt beschäftigt sich mit der Analyse von aktuellen Herausforderungen und zeigt entsprechende Möglichkeiten zur Verbesserung. Die dabei entwickelten Ansätze werden im weiteren Verlauf mechanisch umgesetzt und die benötigten Programmabläufe mit der Software MATLAB realisiert. Zudem werden die entstandenen Ergebnisse erläutert und ein möglicher Ausblick gegeben. So wurde am Ende des Projekts ein Automat konzipiert, welcher auf Sprachbefehle reagiert und unterschiedliche Getränke mithilfe von Sensordaten und Motoren abfüllen kann.

Schlagwörter—Getränkeautomat, LEGO-Mindstorms, NXT, EV3, Sprachsteuerung, MATLAB.

I. EINLEITUNG

Automaten begegnen uns in der heutigen Zeit täglich im Alltag. Sei es der Kaffeeautomat im Haushalt oder ein öffentlich zugänglicher Fahrkartenautomat. Diese Automaten besitzen spezielle Funktionen, die für den entsprechenden Anwendungsfall ausgelegt sind. So definiert man einen Automaten als eine Maschine, die vorbestimmte Abläufe selbsttätig („automatisch“) ausführt. Hierbei spielt der Begriff Automatik eine wichtige Rolle. Er steht für eine Vorrichtung, die einen Vorgang steuert und regelt [1]. In Hinblick auf Getränkeautomaten weist aber genau diese Automatik oft klare Grenzen auf. In vielen Fällen bieten diese Automaten die Möglichkeit, eine Auswahl über einen Tastendruck zu treffen, um daraufhin für eine vorgegebene Zeit ein Magnetventil zu öffnen, welches dafür sorgt, dass das gewünschte Getränk aus der Maschine ausgegeben wird.

Hierbei kann man unter anderem feststellen, dass der Prozess mithilfe von weiteren Sensoren und Eingabemethoden noch verbessert werden kann. Außerdem stellt sich die Frage, wie man vorgeht, wenn man keine Taste betätigen kann oder möchte. Das Ziel des Projekts ist dabei, diesen Vorgang zu optimieren und vor allem weiter zu automatisieren, um die Bedienung des Geräts für den Anwender zu erleichtern und eine größere Flexibilität bieten zu können. Hierbei entstand die Idee für einen intelligenten Getränkeautomaten, der sich auch über eine Sprachsteuerung bedienen lässt.

II. VORBETRACHTUNGEN

Bevor ein Automat mit einer Vielzahl von automatisierten Funktionen realisiert werden kann, sind bestimmte Vorbetrachtungen notwendig, welche nachfolgend erläutert werden.

A. Konventionelle Getränkeautomaten

Um die Funktionsweise eines intelligenten Getränkeautomaten zu erläutern, sollen zunächst die gegenwärtigen Einschränkungen von konventionellen Automaten dargestellt werden. Dabei ist die Zuverlässigkeit ein wichtiges Thema. Getränkeautomaten besitzen einen festgelegten Programmablauf, jedoch mangelt es in einigen Fällen an entsprechenden Sensoren, um mögliche Fehler erkennen und vermeiden zu können. So kann es beispielsweise vorkommen, dass ein Getränkeautomat damit beginnt, ein Getränk auszugeben, obwohl keine Becher mehr vorhanden sind und der Anwender darüber nicht informiert wird. Zudem ist die Funktionalität dieser Automaten oft nur ausreichend. Es fehlen jedoch erweiterte Funktionen, wie das genaue Bestimmen der erforderlichen Füllmenge oder die Erkennung von Objekten. Außerdem sind Getränkeautomaten in vielen Fällen nur mit einer bestimmten Form und Größe von Behältern kompatibel, weshalb beim Befüllen die Menge nicht angepasst werden muss. Wird beispielsweise eine besonders kleine Tasse platziert, könnte diese überlaufen. Schließlich ist die Interaktion mit dem Automaten über Tasten und andere Bedienelemente nur eingeschränkt barrierefrei. Die Möglichkeit, Sprachbefehle nutzen zu können, kann die Barrierefreiheit dabei verbessern.

B. Idee

Die Idee hinter diesem Projekt besteht darin, einen intelligenten Getränkeautomaten herzustellen. Dabei wird der Automat als „intelligent“ bezeichnet, da er mithilfe von Sensoren Messwerte erhebt, auf dessen Grundlage mögliche Probleme von konventionellen Getränkeautomaten gelöst werden können. Hierbei ist die Eingabe von Befehlen über eine Sprachsteuerung ein wichtiger Ansatz. Zudem sollte der Automat eine gewisse Auswahl von unterschiedlichen Getränken bieten. Dabei ist das Ziel, mindestens drei verschiedene Produkte bereitstellen zu können. Die Konstruktion sollte so ausgelegt sein, dass sie mit unterschiedlichen Behältnissen kompatibel ist,

beziehungsweise sich im Idealfall daran anpassen kann. Helligkeitssensoren könnten dafür eingesetzt werden, Objekte, welche vor dem Automaten platziert werden, zu erkennen. Ob und welche Flüssigkeit sich in einem Glas oder Becher befindet, könnte über einen Farbsensor bestimmt und anhand dieser Information eine Vorauswahl getroffen werden. Zudem sollte auch die Temperatur der eingefüllten Flüssigkeit gemessen werden können. Abschließend sollte der Füllstand überwacht werden, um die Füllmenge anzupassen und zu verhindern, dass zu viel oder zu wenig Flüssigkeit abgegeben wird. In diesem Fall kann ein Ultraschallsensor eingesetzt werden.

C. Umsetzung

Zur Realisierung dieser Idee wurden folgende Vorüberlegungen getroffen. Die akustischen Signale für eine Sprachsteuerung werden über ein Mikrofon aufgenommen und verarbeitet. Sobald ein Getränk aus einer Auswahl von möglichen Artikeln genannt wird, muss der Automat bereit sein, die entsprechende Flüssigkeit auszugeben. Wenn zum Beispiel ein Glas vor den Automaten gestellt wird, sieht der Nutzer anhand von LEDs, dass seine Eingabe erkannt wurde. Im nächsten Schritt wird der Automat den Füllstand messen, den Inhalt bestimmen und mit dem Befüllen beginnen. Hierbei ist zu beachten, wie der Transport der Flüssigkeit in das Glas umgesetzt werden kann. Dabei bieten sich im Wesentlichen drei Möglichkeiten an. Die erste Möglichkeit besteht darin, mithilfe der Schwerkraft eine Flüssigkeit aus einem Vorratsbehälter über einen Schlauch zu transportieren und den Durchfluss über Ventile zu regulieren. Eine weitere Methode wäre, anstatt der Ventile einen Mechanismus zu verwenden, welcher den Schlauch zusammendrückt. Die letzte Möglichkeit besteht darin, dass Pumpen den Vorgang aktiv unterstützen und über Relais zugeschaltet werden können. Parallel dazu wird die Temperatur am Gerät angezeigt. Nachfolgend wird das Ende des Vorgangs mithilfe der LEDs signalisiert und der Getränkeautomat wird in seinen Ausgangszustand zurückgesetzt. Man kann das Glas nach diesem Arbeitsschritt wieder entnehmen. Zusätzlich wird ein Abbruch des Programms über eine Taste ermöglicht, um den Vorgang abschließend oder im Falle eines Fehlers, zu beenden.

III. REALISIERUNG

Die Vorbetrachtungen dienen als Grundlage für die nachfolgende Realisierung des Projekts, wobei neben dem Aufbau und der Sprachsteuerung auch die Themen Flüssigkeiten und Sensorik bedeutsam sind.

A. Aufbau

Der Automat wurde aus mehreren Segmenten zusammengesetzt. Als Grundlage dient ein Grundgerüst, welches dafür ausgelegt ist, die Last von allen Einzelementen tragen und diese daran befestigen zu können. Hierbei ist der größte Anteil am Gesamtgewicht auf die Vorratsbehälter (Flaschen) zurückzuführen. An der Oberseite ist ein sog. Sensorpaket angebracht. Wie in Abbildung 1 sichtbar wird, besteht das Sensorpaket aus drei LEGO-Sensoren; einem Tastsensor, einem Ultraschallsensor sowie einem Farbsensor. Zusätzlich befindet sich vor den Sensoren ein Schlauch. Die Sensoren können durch



Abbildung 1: Sensorpaket mit Farbsensor sowie Ultraschall- und Tastsensor

einen LEGO-Motor auf- und abwärts bewegt werden. An der Vorderseite sind zwei LEDs in die horizontalen Träger integriert und werden über Batterien mit einer Gesamtspannung von 9 V betrieben. Ein weiterer Motor betätigt hierbei einen Schalter, um die LEDs ein- und auszuschalten. Wie in Abbildung 2 dargestellt, befinden sich auf dem Grundgerüst zwei Halterungen, um Flaschen mit einem Volumen von 0,5 l mit dem Automaten zu verbinden. Es ist ein Temperatursensor in die Halterung integriert, um die Temperatur der Flüssigkeiten zu bestimmen und die Messwerte an Programmierbausteine zu übermitteln. Ein weiterer Motor wird verwendet, um die gemessene Temperatur auf einer Skala von warm (Rot) bis kalt (Blau) anzuzeigen. Um die Flüssigkeit abzufüllen, werden Schläuche mit einem Innendurchmesser von 4 mm verwendet und mit einer Y-Verbindung verbunden. Hierbei regulieren zwei Motoren den Durchfluss. Zur Steuerung des Automaten werden, aufgrund der Vielzahl von Sensoren und Motoren, zwei LEGO-Programmierbausteine verwendet. Dabei kommen ein NXT und ein EV3 zum Einsatz.



Abbildung 2: Aufbau des Getränkeautomaten mit Becher

B. Sprachsteuerung

Die Auswahl des Getränks erfolgt durch den Nutzer über einen Sprachbefehl. Die Spracherkennung wird dabei auf einem Computer ausgeführt, welcher parallel dazu auch die Kommunikation zu den Programmierbausteinen übernimmt. Die sog. Speech-to-text-Transkription, wodurch gesprochene Worte in einen schriftlichen Text umgewandelt werden, wird in diesem

Fall mithilfe der kostenfreien Webseite „webcaptioner.com“ realisiert. Dort lassen sich zum einen die Transkription starten und zum anderen auch einzelne Wörter als HTTP-Befehl übermitteln [2]. Dazu wird ein sog. Webhook verwendet, welcher im Sekundentakt Befehle an eine zuvor definierte URL sendet. Dabei wird auch ein Port angegeben, auf dem das Gerät entsprechende Daten empfangen kann, beispielsweise Port 4000 oder 4001. Die URL ist in diesem Fall der Computer selbst, d.h. man verwendet „http://localhost:4001/“ oder alternativ „http://127.0.0.1:4001/“. Damit man die Daten verarbeiten kann, müssen die HTTP-Befehle von MATLAB ausgewertet werden, da die Programmierung des Automaten in MATLAB erfolgt. Die Umsetzung ermöglicht die Funktion „Web Server“, welche von MathWorks angeboten wird und sich in MATLAB integrieren lässt.

„This function WEBSERVER, is a HTTP webserver for HTML code and images and also allows execution of Matlab code through the web.“ [3]

In diesem Fall startet die Funktion einen Webserver auf dem zuvor definierten Port 4001 und gibt die eingehenden Befehle in der Konsole aus. Im weiteren Verlauf wird das Skript, wie im Anhang dargestellt, so angepasst, dass lediglich bestimmte Schlagwörter verarbeitet werden, da das Ziel verfolgt wird, zwischen drei verschiedenen Getränken zu wählen, wobei alle Eingaben, welche nicht den vorgeschriebenen möglichen Eingaben entsprechen, verworfen werden. Abbildung 3 zeigt den Programmablaufplan dieser Spracherkennung. Es ist erkennbar, dass nachdem der Server gestartet wurde und eine Spracheingabe erfolgt ist, überprüft wird, ob die Worte „Cola“, „Fanta“ oder „Spezi“ enthalten sind. Sollte dieser Fall eintreten, so wird die entsprechende MATLAB-Funktion ausgeführt. Nach der Ausführung lässt sich das Programm über einen Taster abbrechen, anderenfalls wird der Vorgang wiederholt.

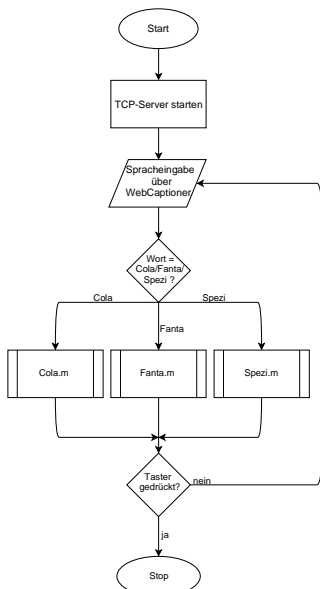


Abbildung 3: Programmablaufplan der Spracherkennung

C. Flüssigkeiten

Neben dem Transport der Flüssigkeiten besteht die Herausforderung darin, eine geeignete Methode zu finden, um den

Durchfluss zu unterbrechen. In diesem Projekt werden die Schläuche an einer Stelle eingedrückt, sodass sie vollständig geschlossen werden. Dabei wird die einfache Verformbarkeit des Schlauchs als Vorteil genutzt. Die Motoren üben den notwendigen Druck aus, indem sie bis zu einem festgelegten Punkt rotieren. Nachfolgend werden die beiden Flüssigkeiten über Schläuche zusammengeführt. Da sich die Ausgabe an einem beweglichen Sensorkopf befindet, existiert eine Schlauchreserve von einigen Zentimetern in der Konstruktion, sodass sowohl das Minimum als auch Maximum des Sensorpakets erreicht werden kann, wodurch sowohl kleine als auch große Becher verwendet werden können.

D. Sensorik

Um das Programm zu realisieren, werden fünf Sensoren benötigt. Wie in Abbildung 4 dargestellt, muss zunächst die Helligkeit bestimmt werden, um zum Beispiel ein Glas zuverlässig zu erkennen. Diese Messung erfolgt über die Bestimmung der Beleuchtungsstärke mit dem Helligkeitssensor. Wird ein Schwellwert von 210 lx unterschritten, fährt der Automat im Programmablauf fort. Ein Tastsensor wird dafür benötigt, ein Signal zu übertragen, falls der Sensorkopf den Rand eines Behälters berührt, sodass dieser nicht weiter abwärts bewegt wird. Nachdem die Temperaturmessung mit Temperatursensor erfolgt ist, bestimmt der Ultraschallsensor den Abstand zwischen Sensorkopf und Wasseroberfläche. Bei Unterschreitung des Werts von 5 cm schließen die Ventile, damit die maximale Füllhöhe eingehalten wird. Ein weiterer Tastsensor kann zum Abbruch betätigt werden.

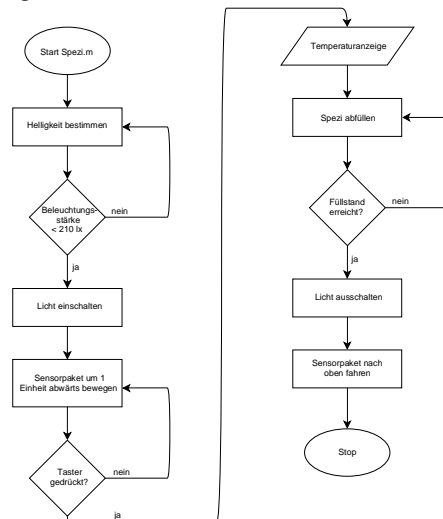


Abbildung 4: Programmablaufplan eines MATLAB-Skripts für ein spezifisches Getränk

IV. ERGEBNISDISKUSSION

Am Ende des Projekts ist ein Getränkeautomat entstanden, welcher einen Sprachbefehl akzeptiert und die Möglichkeit bietet, zwischen drei verschiedenen Getränken zu wählen. Dabei kann aus zwei Flaschen ein neues Getränk gemischt werden. Zudem ist der Automat in der Lage, die Füllmenge zu bestimmen und Gläser oder Becher beliebiger Größe präzise auf ein eingestelltes Maximum zu befüllen. Dieser Prozess kann auch mehrfach mit unterschiedlicher Getränkewahl wiederholt

werden. Außerdem bietet der Automat eine Temperaturanzeige und signalisiert zusätzlich den aktuellen Zustand visuell. Zwar wurde die Funktion, welche eine Flüssigkeit, die sich noch im Becher befindet, eindeutig identifiziert, in diesem Projekt nicht mehr umgesetzt, jedoch stand die Kosten-Nutzen-Bilanz aufgrund der Komplexität dieser Funktion ohnehin in einem negativen Verhältnis.

Bei der Realisierung der Idee sind auch einige Probleme aufgetreten. So gab es zu Beginn des Projekts ein Problem mit der Kommunikation zwischen Computer und Programmierbausteinen mittels Bluetooth. Diese Verbindung war stets instabil und sorgte für Fehler bei der Ausführung von Code, weshalb die Programmierbausteine kabelgebunden verwendet wurden.

Im weiteren Verlauf zeigte sich, dass ein Farbsensor zur Erkennung von farbigen Flüssigkeiten ungeeignet ist. Ein Farbsensor erkennt Objekte durch die Verwendung von mehrfarbigem Licht. So werden drei LEDs als Sendelichtquelle für die Wellenlängen Rot, Grün und Blau verwendet. Schließlich wird ein Fotodetektor verwendet, um darauf die reflektierte Strahlung des Objekts nacheinander abzubilden [4]. Da dieser Sensor die Flüssigkeiten mit den LEDs direkt durchdringt, kann er in diesem Fall nicht die entsprechenden Messwerte liefern und somit nicht eingesetzt werden. Alternativ wäre die Verwendung von Helligkeitssensoren möglich, da die Getränke unterschiedlich lichtdurchlässig sind. Auch wäre die Verwendung von Sensoren mit Infrarot-LEDs denkbar, da deren Wellenlängen bei 780 Nanometer beginnen und von vielen Flüssigkeiten absorbiert werden [5].

Bei Verwendung der Sprachsteuerung trat zunehmend das Problem auf, dass einzelne Befehle gar nicht, unvollständig oder nur mit sehr großer Latenz vom Webserver erfasst wurden. Diese Problematik kann unter Umständen auf die Auslastung und Leistungsfähigkeit des verwendeten Computers zurückzuführen sein, da bei diesem Projekt mehrere Prozesse parallel benötigt wurden.

Zur besseren Umsetzung des Abfüllvorgangs wären entsprechende Ventile hilfreich, welche einen Innendurchmesser von acht bis zehn Millimeter besitzen. Die bei diesem Projekt verwendete Methode ist teilweise anfällig für Fehler, da die Motoren nicht immer exakt die festgelegte Anzahl an Umdrehungen umsetzen, die Ausgangsposition nicht erreichen und somit die Leitung in manchen Fällen noch geringfügig geöffnet oder geschlossen ist. Die Ventile sollten dabei möglichst einfach zu betätigen und aus Kunststoff oder einem ähnlich leichten Material gefertigt sein, um das Gesamtgewicht nicht zusätzlich zu erhöhen. Dafür könnten diese speziell angefertigt werden, um den Anforderungen zu entsprechen. Beispielsweise ließen sich Kunststoffteile in einem 3D-Drucker herstellen. Sollte es nicht möglich sein, Ventile zu verwenden, so wäre die Konstruktion von Getrieben möglich, sodass mehr Kraft auf einen Punkt konzentriert werden kann und die Toleranz bei ungenauer Anzahl an Umdrehungen größer wäre.

V. ZUSAMMENFASSUNG UND FAZIT

Die Motivation hinter diesem Projekt war, einen Automaten zu entwickeln, der die Kritikpunkte an konventionellen Automaten, welche zuvor genannt wurden, umgeht und die Idee von

einem intelligenten Getränkeautomaten möglichst gut umsetzt. Dieses Ziel konnte mit dem Endprodukt sowohl theoretisch als auch praktisch erreicht werden. Als Vorlage dafür dienten bereits existierende Automaten. Das Ergebnis bietet auch einige Verbesserungsmöglichkeiten, wie zum Beispiel die Verwendung von Bluetooth-Adaptoren, die eine höhere Kompatibilität in Verbindung mit anderen Geräten bieten oder die Auslagerung der Spracherkennung auf ein anderes Gerät, dank der Kommunikation über HTTP.

Als Erweiterung des Konzepts für Getränkeautomaten wäre, neben einer reinen Eingabe von Sprachbefehlen, eine akustische Rückmeldung denkbar, sodass der Nutzer über seine Auswahl informiert wird und einen möglichen Fehler gleich korrigieren kann. Außerdem wäre die Messung der Füllstände in den Vorratsbehältern, in diesem Fall Flaschen, hilfreich. Hier könnte bei offenen Behältern ein Ultraschallsensor den genauen Füllstand angeben und den Nutzer darüber informieren, wie viele Getränke, basierend auf der Bechergroße, noch abgefüllt werden können. Die akustischen Rückmeldungen können dabei mittels Text-To-Speech-Transkription erzeugt und ausgegeben werden. Dabei wird ein Text eingegeben und automatisch vorgelesen.

Natürlich müssten für einen realen Betrieb des Automaten auch noch optische Anpassungen vorgenommen werden, jedoch konnte die gewünschte Funktionsweise im Wesentlichen realisiert werden.

ANHANG

```
[...]
if strcmp(request.Wort, 'Fanta')
    disp('Fanta ausgewählt')
    Fanta;
    disp('ausgeführt')
    pause(1)
else if strcmp(request.Wort, 'Cola')
[...]
```

Abbildung 5: Kurzer Ausschnitt des Quelltextes zur Erkennung eines möglichen Befehls

LITERATURVERZEICHNIS

- [1] WIKIPEDIA, THE FREE ENCYCLOPEDIA, 2021. *Automat* [online]. Wikipedia, 15.07.2021 [Zugriff am: 18.02.2022]. Verfügbar unter: <https://de.wikipedia.org/wiki/Automat>
- [2] WEB CAPTIONER, 2022. *Web Captioner* [online]. Web Captioner [Zugriff am: 18.02.2022]. Verfügbar unter: URL: <https://webcaptioner.com>
- [3] KROON, Dirk-Jan, 2022. *Web Server* [online]. MATLAB Central File Exchange, 18.10.2011 [Zugriff am: 18.02.2022]. Verfügbar unter: <https://www.mathworks.com/matlabcentral/fileexchange/29027-web-server>
- [4] ELEKTROPRAKTIKER, 2005. Farbsensoren – Konkurrenz für unser Auge. In: *Elektropraktiker* [online]. 08.2005 [Zugriff am: 18.02.2022]. Verfügbar unter: <https://www.elektropraktiker.de/nc/fachartikel/farbsensoren-konkurrenz-fuer-unser-auge/>
- [5] FRAUENDORF, Judith, 2002. *Entwicklung und Anwendung von Fernerkundungsmethoden zur Ableitung von Wasserqualitätsparametern verschiedener Restseen des Braunkohlentagebaus in Mitteldeutschland* [Dissertation]. Martin-Luther-Universität Halle-Wittenberg. Halle (Saale): Universitäts- und Landesbibliothek Sachsen-Anhalt. Verfügbar unter: <http://dnb.info/967158389/34>

Sprachgesteuerter Getränkeautomat mit LEGO Mindstorms

Jannik Seidler, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Abstract—Im Rahmen des Studienfachs Elektrotechnik/ Informationstechnik soll ein Konzept geschaffen werden, dessen Konstruktion und Programmierung auf Elementen von LEGO Mindstorms basiert. Dieses Projekt beschäftigt sich mit einem Getränkeautomat, der mehr Funktionen als ein herkömmlicher Automat besitzt. Die dabei entwickelten Ansätze werden im weiteren Verlauf mechanisch umgesetzt und die benötigten Programmabläufe mit der Software MATLAB realisiert. Das finale Produkt ist ein Automat, der mithilfe von Sprachbefehlen und Sensoren unterschiedliche Getränke in unterschiedliche Behälter abfüllen kann.

Schlagwörter—LEGO-Praktikum, NXT-Programmierung, Getränkeautomat, Spracherkennung.

I. EINLEITUNG

Automaten sind Teil des alltäglichen Lebens. Sie sind so integriert, das man manchmal nicht merkt, wie viel Arbeit uns dadurch abgenommen wird.

Man definiert einen Automaten als eine Maschine, die vorbestimmte Abläufe selbsttätig („automatisch“) ausführt. Hierbei spielt der Begriff Automatik eine wichtige Rolle, denn er steht für eine Vorrichtung, die einen Vorgang selbstständig steuern und regeln kann [1]. In Hinblick auf Getränkeautomaten weist aber genau diese Automatik oft klare Grenzen auf. So findet die Getränkewahl über Tastendruck statt, wonach ein Ventil öffnet, oder eine Pumpe für eine bestimmte Zeit läuft. Dabei sind diese Vorgänge aber nur auf standardisierte Becher vorgesehen, die komplett leer sind, also nicht berücksichtigen, wenn eine Person einen halbvollen Becher nachfüllen möchte.

Hierbei lässt sich erkennen, dass noch Verbesserungspotenzial besteht, welches mithilfe von weiteren Sensoren und Eingabemethoden erreicht werden kann. Außerdem stellt sich die Frage, wie man vorgeht, wenn man keine Taste betätigen kann, möchte oder soll z.B. aufgrund der Covid-19-Pandemie. Das Ziel des Projekts ist dabei, diesen Vorgang zu optimieren, weiter zu automatisieren und für mehr Personen zugänglich zu machen. Hierbei entsteht die Idee für einen intelligenten Getränkeautomaten, der sich über eine Sprachsteuerung bedienen lässt.

II. VORBETRACHTUNGEN

Bevor ein Automat mit einer Vielzahl von automatisierten Funktionen realisiert werden kann, sind Voraussetzungen notwendig, welche nachfolgend erläutert werden.

A. Konventionelle Getränkeautomaten

Um die Idee eines intelligenten Getränkeautomaten verständlicher zu machen, sollen zunächst die gegenwärtigen Einschränkungen erläutert werden. Getränkeautomaten besitzen einen festgelegten Programmablauf, jedoch mangelt es in einigen Fällen an hilfreichen Sensoren, um Fehler erkennen und vermeiden zu können. Es kann passieren, dass ein Getränkeautomat beginnt, ein Getränk auszugeben, obwohl keine Becher mehr vorhanden sind und der Anwender darüber nicht informiert wird, oder dass der Becher erst nach dem Abfüllen des Getränkes in die Ablage fällt. Zudem ist die Funktionalität dieser Automaten oft nur ausreichend. Eine Erweiterung zu den vorhandenen Funktionen wäre zum Beispiel das Bestimmen der erforderlichen Füllmenge anhand eines nicht-standardisierten Mehrwegbechers oder die Erkennung von Störobjekten, um unnötiges Ausschütten des Getränks zu vermeiden.

Aufgrund der Covid-19-Pandemie ist das häufige Benutzen der Auswahl Tasten fragwürdig, da diese zu Risiken werden. Diese mögliche Gefahr lässt sich umgehen, indem die Getränkeauswahl per Spracheingabe stattfindet.

B. Idee

Die Idee hinter diesem Projekt besteht darin, einen intelligenten Getränkeautomaten herzustellen. Dabei wird der Automat als „intelligent“ bezeichnet, da mithilfe von Sensoren Messwerte erhoben und ausgewertet werden, mit dessen Hilfe mögliche Fehler von konventioneller Getränkeautomaten gelöst werden können. Hierbei ist die Eingabe von Befehlen über eine Sprachsteuerung ein wichtiger Ansatz. Er soll eine gewisse Auswahl unterschiedlicher Getränke bieten. Dabei ist das Ziel, drei verschiedene Produkte bereitstellen zu können. Die Konstruktion soll so konzipiert werden, dass sie mit unterschiedlichen Getränkebehältern kompatibel ist, beziehungsweise sich im Idealfall daran anpassen kann. Helligkeitssensoren werden zur Objekterkennung verwendet, welche vor dem Automaten platziert werden, um unnötiges Ausschütten bei Nichterkennung zu vermeiden. Ob und welche Flüssigkeit sich in einem Glas oder Becher befindet, könnte über einen Farbsensor bestimmt und anhand dieser Information eine Voraus-

wahl getroffen werden, falls der Nutzer dasselbe Getränk nochmal haben möchte. Zudem soll die Temperatur der eingefüllten Flüssigkeit gemessen werden können, um dem Nutzer zu signalisieren, ob sein Getränk warm oder kalt ist. Zuletzt soll die Füllmenge im Becher überwacht werden, damit die richtige Menge vom Getränk abgegeben wird. Zu diesem Zweck wird ein Ultraschallsensor eingesetzt.

C. Umsetzung

Zur Realisierung dieser Idee wurden folgende Überlegungen getroffen. Die Signale für die Sprachsteuerung werden über ein Mikrophon aufgenommen und verarbeitet. Sobald eine Eingabe mit dem bekannten Repertoire übereinstimmt, muss der Automat die entsprechende Flüssigkeit ausgeben. Wenn zum Beispiel ein Glas vor den Automaten gestellt wird, erkennt der Nutzer anhand von LEDs, dass seine Eingabe, sein Getränkewunsch, erkannt wurde. Im nächsten Schritt wird der Automat den Füllstand fortlaufend messen, den Inhalt bestimmen und mit dem Befüllen beginnen. Um das Glas im Anschluss daran befüllen zu können, bieten sich drei Lösungsalternativen an. Die erste Möglichkeit besteht darin, mithilfe der Schwerkraft eine Flüssigkeit aus einem Vorratsbehälter über einen Schlauch zu transportieren und die Zulaufmenge per Ventil zu regeln. Eine weitere Methode wäre, anstatt der Ventile einen Mechanismus zu verwenden, welcher den Schlauch zusammendrückt. Die dritte Lösung besteht darin, dass das Getränk mithilfe von Pumpen gefördert wird, welche per Relais bedient werden können.

Ebenso wird die Temperatur des Getränks für den Nutzer sichtbar angezeigt. Zuletzt wird per LEDs das Ende des Befüllens signalisiert und der Automat in den Ausgangszustand zurückgesetzt. Man kann das Glas nach diesem Arbeitsschritt wieder entnehmen.

III. REALISIERUNG

Die Vorbetrachtungen dienen als Grundlage für die nachfolgende Realisierung des Projekts, wobei neben dem Aufbau und der Sprachsteuerung auch die Themen „Flüssigkeiten“ und „Sensorik“ bedeutsam sind.

A. Aufbau

Der Automat wird aus mehreren Segmenten zusammengesetzt. Als Grundlage dient ein Grundgerüst, welches dafür ausgelegt ist, die Last aller Einzelelemente tragen und diese daran befestigen zu können. Hierbei ist der größte Anteil am Gesamtgewicht auf die Vorratsbehälter (Flaschen) zurückzuführen. An der Oberseite befindet sich ein sog. Sensorkpaket. Wie in Abbildung 1 sichtbar wird, besteht das Sensorkpaket aus drei LEGO-Sensoren: einem Tastsensor, einem Ultraschallsensor und einem Farbsensor. Zusätzlich befindet sich vor den Sensoren ein Schlauch, welcher die Flüssigkeit ins Glas leitet. Die Sensoren können durch einen LEGO-Motor auf- und abwärts bewegt werden, bis entweder die maximal ausfahrbare Länge erreicht ist, oder der Tastsensor auf den Becherrand drückt.

An der Vorderseite des Automaten sind zwei LEDs integriert und werden über einen 9 V Batterieblock betrieben. Ein weiterer Motor betätigt den Schalter, um die LEDs ein- und auszu-



Abbildung 1: Sensorkopf mit Farbsensor sowie Ultraschall- und Tastsensor

schalten. Wie in Abbildung 2 dargestellt, befinden sich auf dem Grundgerüst zwei Halterungen für ½ Liter Flaschen. Es ist ein Temperatursensor in einer Halterung verbaut, um die Temperatur der Flüssigkeit zu bestimmen und die Messwerte zu übermitteln. Ein weiterer Motor wird verwendet, um die gemessene Temperatur auf einer Skala von warm (Rot) bis kalt (Blau) anzuzeigen. Um die Flüssigkeit abzufüllen, werden Silikonschläuche mit einem Innendurchmesser von 4 mm verwendet und mit einer Y-Verbindung verbunden. Der Durchfluss wird von zwei Motoren mittels Abklemmen des Schlauches reguliert. Zur Steuerung des Automaten werden, aufgrund der Vielzahl von Sensoren und Motoren, zwei LEGO-Programmierbausteine verwendet. Dabei kommen ein NXT und ein EV3 zum Einsatz.



Abbildung 2: Aufbau des Getränkeautomaten mit Becher

B. Sprachsteuerung

Das Getränk wird per Spracheingabe des Nutzers gewählt, die auf einem Computer verarbeitet wird. Dieses Programm nimmt dann die Kommunikation zu den Programmierbausteinen auf.

Die sogenannte „Speech-to-text Transkription“, bei der gesprochene Worte in schriftlichen Text umgewandelt werden, wird mithilfe der kostenfreien Webseite „webcaptioner.com“ realisiert. Mit dieser Internetseite lassen sich Wörter per HTML-Code über einen Webhook an eine definierte URL senden [2].

Die Daten werden an einen Port gesendet, beispielsweise Port 4000 oder 4001 und die URL ist der Computer selbst (Beispiel: „http://localhost:4001/“ oder „http://127.0.0.1:4001/“). Die Auswertung der HTTP-Befehle erfolgt in MATLAB mit der „Webserver“-Funktion. Dieses wird von MathWorks zum Download angeboten.

„This function WEBSERVER, is a HTTP webserver for HTML code and images and also allows execution of Matlab code through the web.” [3]

Im Beispiel des Getränkeautomaten werden nur die ausgewählten Getränke in der Konsole angezeigt. Dafür wird das Skript so angepasst, dass nur die Nennung dieser ausgewählten Getränkeprodukte Beachtung findet. Andere Spracheingaben werden zwar erkannt, jedoch nicht ausgewertet. Abbildung 3 zeigt den Programmablaufplan dieser Spracherkennung. Es wird deutlich, dass, nachdem der Server gestartet wurde und eine Spracheingabe erfolgte, überprüft wird, ob die Worte „Cola“, „Fanta“ oder „Spezi“ enthalten sind. Sollte dieser Fall eintreten, so wird die entsprechende MATLAB-Funktion umgesetzt. Nach der Ausführung lässt sich das Programm über einen Taster abbrechen, anderenfalls wird der Vorgang wiederholt.

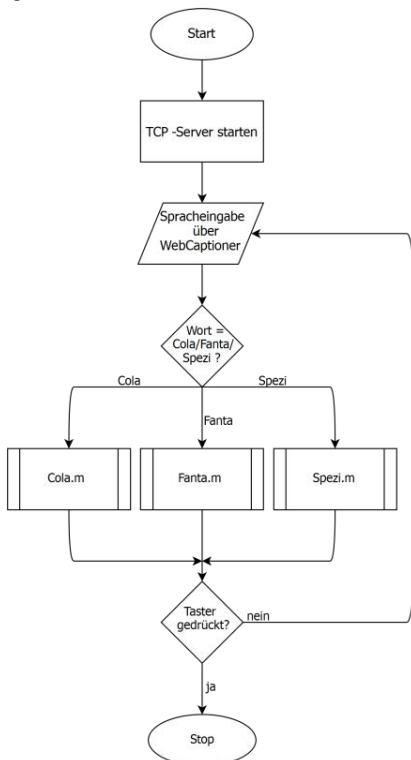


Abbildung 3: Programmablaufplan der Spracherkennung

C. Flüssigkeiten

Um den Durchfluss zu unterbrechen werden in diesem Projekt die Schläuche an einer Stelle eingedrückt, sodass sie vollständig geschlossen werden. Dabei wird die einfache Verformbarkeit des Schlauchs als Vorteil genutzt. Die Motoren üben den notwendigen Druck aus, indem sie bis zu einem festgelegten Punkt rotieren. Da sich die Flüssigkeitsausgabe an einem beweglichen Sensorkopf befindet, existiert eine Schlauchreserve von einigen Zentimetern in der Konstruktion, sodass

sowohl das Minimum als auch das Maximum des Sensorpakets erreicht werden kann. Dadurch können sowohl kleine als auch große Becher verwendet werden.

D. Sensorik

Um das Programm zu realisieren werden fünf Sensoren benötigt. Wie in Abbildung 4 dargestellt, wird der Helligkeitssensor mit dem Schwellenwert von 210 lm genutzt, um das Vorhandensein eines Bechers zu erkennen. Wird der Schwellenwert unterschritten, wird das Programm fortgesetzt und das Licht eingeschaltet. Der Tastsensor sendet ein Signal falls er den Rand des Behälters berührt, damit das Programm fortfahren kann. Nach der Temperaturmessung mit dem Temperatursensor bestimmt der Ultraschallsensor den Abstand von der Getränkeoberfläche. Bei Unterschreitung des Werts von 5 cm schließen die Motoren, damit die maximale Füllhöhe eingehalten wird. Ein weiterer Tastsensor kann zum Abbruch betätigt werden.

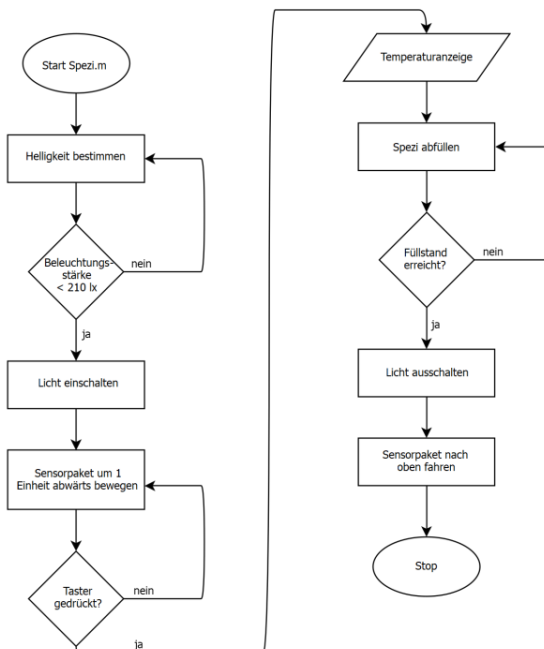


Abbildung 4: Programmablaufplan eines MATLAB-Skripts für ein spezifisches Getränk

IV. ERGEBNISDISKUSSION

Es wurde ein Getränkeautomat geschaffen, der anhand eines Sprachbefehls zwischen drei Getränken unterscheiden kann. Dabei kann aus zwei Flaschen ein neues Getränk gemischt werden. Der Automat ist in der Lage, die Füllmenge zu bestimmen und Behälter beliebiger Größe auf ein eingestelltes Maximum zu befüllen. Dieser Prozess kann mit unterschiedlicher Getränkewahl wiederholt werden. Zudem bietet der Automat eine Temperaturanzeige und signalisiert zusätzlich den aktuellen Zustand visuell. Die Funktion, eine Flüssigkeit zu identifizieren, wurde zwar getestet, jedoch als unzuverlässig und ineffizient eingestuft. Deshalb wurde darauf verzichtet.

Bei der Realisierung der Idee sind auch einige Probleme aufgetreten. Es wurde geplant, per Bluetooth mit den Programmierbausteinen zu kommunizieren. Dies funktionierte jedoch

aufgrund veralteter Bluetooth-Module nicht. Aus diesem Grund wurde per Kabel kommuniziert.

Im weiteren Verlauf zeigte sich, dass ein Farbsensor zur Erkennung farbiger Flüssigkeiten ungeeignet ist. Dieser erkennt Objekte durch die Verwendung von mehrfarbigem Licht. So werden drei LEDs als Sendelichtquelle für die Wellenlängen Rot, Grün und Blau verwendet. Schließlich wird ein Fotodetektor verwendet, um darauf die reflektierte Strahlung des Objekts nacheinander abzubilden [4]. Da dieser Sensor die Flüssigkeiten mit den LEDs direkt durchdringt, kann er in diesem Fall nicht die entsprechenden Messwerte liefern und somit nicht eingesetzt werden. Alternativ wäre die Verwendung von Helligkeitssensoren möglich, da die Getränke unterschiedlich lichtdurchlässig sind. Auch wäre die Verwendung von Sensoren mit Infrarot-LEDs denkbar, da die typische Wellenlänge von 780 Nanometer von vielen Flüssigkeiten absorbiert wird [5].

Bei der Sprachsteuerung zeigte sich phasenweise das Problem, dass Befehle gar nicht, unvollständig oder nur mit sehr großer Latenz erfasst wurden. Dies kann unter Umständen auf die Auslastung und mangelnde Leistungsfähigkeit des verwendeten Computers zurückzuführen sein, da bei diesem Projekt mehrere Prozesse parallel benötigt wurden. Zudem lag dieses Problem vielleicht auch an der genutzten Spracherkennungssoftware.

Zur Verbesserung des Abfüllvorgangs wären Ventile hilfreich, welche einen Innendurchmesser von acht bis zehn Millimeter besitzen, da sonst der Füllvorgang einen sehr langen Zeitraum in Anspruch nimmt. Die bei diesem Projekt verwendete Methode mit dem Abklemmen des Schlauches war teilweise fehleranfällig. Die Motoren wiesen nicht die notwendige Leistung auf und konnten so nicht reproduzierbar die festgelegte Anzahl an Umdrehungen ausführen. Die Ausgangsposition wurde nicht erreicht und somit konnte die Leitung in manchen Fällen nicht korrekt geöffnet oder geschlossen werden. Dieses Problem könnte mit einem Getriebe behoben und gleichzeitig die Präzision erhöht werden.

Bei der Verwendung von Ventilen sollte auf Leichtgängigkeit geachtet werden, damit die Motoren nicht zu stark belastet werden. Diese könnte man per 3D-Druck selber herstellen, um Gewicht zu sparen.

V. ZUSAMMENFASSUNG UND FAZIT

Die Motivation zu diesem Projekt bestand darin einen Automaten zu schaffen, der die zuvor genannten Probleme konventioneller Automaten umgeht und die Idee von einem intelligenten Getränkeautomaten möglichst gut umsetzt. Dies wurde mit dem Endprodukt sowohl auf theoretischer als auch auf praktischer Basis erreicht und umgesetzt. Die Vorlagen zur Verbesserung stammten von bereits existierenden Automaten. Obwohl das Ergebnis zufriedenstellend ist, bieten sich einige Verbesserungsmöglichkeiten an. In diesem Zusammenhang ist die Verwendung von Bluetooth-Adaptoren zu nennen, die eine höhere Kompatibilität in Verbindung mit anderen Geräten bieten. Weiterhin ist die Auslagerung der Spracherkennung auf ein anderes Gerät denkbar, dank der Kommunikation über HTTP. Zudem könnte man als letzten Schritt den Automaten

unabhängig von einem separaten Rechenmodul machen und dieses in den Automaten einbauen.

Eine Erweiterung des Konzepts für Getränkeautomaten wäre es, neben der reinen Eingabe von Sprachbefehlen zusätzlich eine akustische Rückmeldung zu installieren. Diese würde die Getränkeauswahl bestätigen, sodass der Nutzer über seine Auswahl informiert wird und einen möglichen Fehler gleich korrigieren kann. Die akustischen Rückmeldungen können dabei mittels Text-To-Speech-Transkription erzeugt und ausgegeben werden. Dabei wird ein Text eingegeben und automatisch vorgelesen.

Eine weitere Verbesserung wäre es, die Füllstände in den Vorratsbehältern messen zu können. Das beträfe in diesem Fall den Stand in den Flaschen. Hier könnte bei einem offenen Behälter ein Ultraschallsensor den Füllstand angeben und den Nutzer darüber informieren, wie viele Getränke basierend auf der Bechergröße noch abgefüllt werden könnten. Zudem bestünde so die Möglichkeit bei einem leeren Behälter eine Nachricht versendet werden, dass der Behälter aufgefüllt werden soll.

Für einen funktionstüchtigen, betriebsbereiten Automaten müssten auch noch optische Anpassungen wie eine Außenverkleidung vorgenommen werden, jedoch konnte die gewünschte Funktionsweise im Wesentlichen realisiert werden.

ANHANG

```
[...]
if strcmp(request.Wort, 'Fanta')
    disp('Fanta ausgewählt')
    Fanta;
    disp('ausgeführt')
    pause(1)
else if strcmp(request.Wort, 'Cola')
[...]
```

Abbildung 5: Kurzer Ausschnitt des Quelltextes zur Erkennung eines möglichen Befehls

LITERATURVERZEICHNIS

- [1] WIKIPEDIA, THE FREE ENCYCLOPEDIA, 2021. *Automat* [online]. Wikipedia, 15.07.2021 [Zugriff am: 18.02.2022]. Verfügbar unter: <https://de.wikipedia.org/wiki/Automat>
- [2] WEB CAPTIONER, 2022. *Web Captioner* [online]. Web Captioner [Zugriff am: 18.02.2022]. Verfügbar unter: URL: <https://webcaptioner.com>
- [3] KROON, Dirk-Jan, 2022. *Web Server* [online]. MATLAB Central File Exchange, 18.10.2011 [Zugriff am: 18.02.2022]. Verfügbar unter: <https://www.mathworks.com/matlabcentral/fileexchange/29027-web-server>
- [4] ELEKTROPRAKTIKER, 2005. Farbsensoren – Konkurrenz für unser Auge. In: *Elektropraktiker* [online]. 08.2005 [Zugriff am: 18.02.2022]. Verfügbar unter: <https://www.elektropraktiker.de/nc/fachartikel/farbsensoren-konkurrenz-fuer-unser-auge/>
- [5] FRAUENDORF, Judith, 2002. *Entwicklung und Anwendung von Fernerkundungsmethoden zur Ableitung von Wasserqualitätsparametern verschiedener Restseen des Braunkohlentagebaus in Mitteldeutschland* [Dissertation]. Martin-Luther-Universität Halle-Wittenberg. Halle (Saale): Universitäts- und Landesbibliothek Sachsen-Anhalt. Verfügbar unter: <http://d-nb.info/967158389/34>

Automatisierter Kartenmischer

Erik Borowski, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Zuge des LEGO-Praktikums 2022 an der Otto-von-Guericke-Universität Magdeburg sollten auch in diesem Jahr die Studierenden eine automatisierte Maschine, die eine sinnvolle Funktion ausführt, mithilfe des LEGO-Mindstorms-NXT Set und MATLAB programmieren und entwickeln. Für dieses Projekt sollte eine Maschine, die auf Grundlage der gegebenen technischen Voraussetzungen eine unmittelbar nützliche Funktion im Alltag bietet entworfen werden. Die Wahl fiel auf einen automatisierten Kartenmischer, der zwei separate Stapel an Karten auf unterschiedliche Weise durchmischen kann. Im Folgenden wird die Entstehung des Projektes von der ursprünglichen Idee und Konzeption über die tatsächliche Entwicklung bis hin zum endgültigen Ergebnis erläutert.

Schlagwörter—Automatisierung, Kartenmischer, Zufallsalgorithmus, Lego Mindstorms NXT, MATLAB

I. EINLEITUNG

Die Automatisierung des Alltags ist weit fortgeschritten und zieht sich vom öffentlichen Sektor bis in kleinste private Anwendungsgebiete hindurch. Wichtige, anstrengende oder auch einfach lästige Aufgaben sollen bestenfalls nicht mehr durch Menschenhand, sondern durch intelligenzvolle, automatisierte und vor allem zuverlässige Maschinen vollführt werden. Dies beginnt bei großen Themengebieten wie beispielsweise der Medizin, im Verkehrsbereich oder der Logistik von großen Wirtschaftsunternehmen und endet im eigenen Heim bei einfachen Haushaltsgeräten, die von einem Großteil der Menschheit benutzt als selbstverständlich angesehen werden. Ein automatischer Kartenmischer, der den in Abbildung 1 dargestellten Vorgang ausführt, ist in diesem Fall ein gutes Beispiel für die fortschreitende Automatisierung noch so kleiner und individueller Aufgaben. Die Anforderungen an den Kartenmischer, der in diesem Projekt entwickelt werden sollte, lagen darin, die Karten eines beliebigen Kartenstapels zuverlässig zu durchmischen und dies auf unterschiedliche Weisen realisieren zu können.

II. VORBETRACHTUNGEN

Als Vorbild und Orientierung dienten bereits existierende Kartenmischer, die zumeist durch einen einfachen Mechanismus die Karten durchmischen. Ausgehend davon, sollte dieser herkömmliche Mechanismus durch weitere Möglichkeiten erweitert werden, um die Karten auf andere Weise zu mischen.

A. Allgemeine Funktionsweise eines Kartenmischers

Der einfachste Weg des automatisierten Mischens von Karten schien darin, die Karten wie beim herkömmlichen Mischen von Hand in zwei Stapel zu unterteilen und gleichzeitig mit selben Tempo zu durchmischen. Dafür müssen die Karten der



Abbildung 1. Vorgang des Kartenmischens von Hand [1]

beiden Stapel mit gleichem Tempo und zur selben Zeit in ein gemeinsames Fach transportiert werden, in dem dann der neue durchmischte Stapel liegt.

B. Erweiterung der Funktionsweise

1) *Manuelle Auswahl*: Mithilfe der manuellen Auswahl sollte der Anwender mithilfe von Tastern selbst bestimmen, ob Karten von Stapel 1 oder Stapel 2 Karten ausgewählt werden und beliebig zwischen den Stapeln wechseln können, um die Karten nach Belieben zu durchmischen.

2) *Zufallsalgorithmus*: Mithilfe des Zufallsalgorithmus sollte nacheinander immer einer der Stapel angesteuert werden, von dem dann eine Karte in das gemeinsame Fach transportiert werden sollte. Eine Funktion, die zufällig entweder eine 1 (für Stapel 1) oder eine 2 (für Stapel 2) ausgibt, dann eine Karte vom jeweiligen Stapel auswählt und sich dann wiederholt bis alle Karten im gemeinsamen Fach liegen, sollte das Verhalten realisieren.

C. Hard- und Software zur Umsetzung

1) *LEGO-Mindstorms-NXT*: Für die Hardware und mechanische Funktionsweise stellte das LEGO-Mindstorms-NXT-Set die Grundlage dar. Neben den einfachen Bausteinen zur Konstruktion standen verschiedene Sensoren wie z. B. Farb-, Licht- oder Ultraschallsensoren und Motoren zur Verfügung, die über den NXT-Stein angesteuert werden (s. Abb. 2). Für den Kartenmischer wurden Tastsensoren verwendet, die dem Anwender ermöglichen bestimmte Funktionen zu starten und zu stoppen. Außerdem wurden zwei Motoren verwendet, um die Karten zu bewegen.



Abbildung 2. NXT-Stein, Motoren und Sensoren [2]

2) *MATLAB und RWTH Aachen Toolbox*: Die Programmierung des NXT-Bausteins erfolgte mithilfe der numerischen Rechen- und Programmierungssoftware MATLAB von MathWorks und der von der RWTH Aachen entwickelten Toolbox [3], die es ermöglicht, den NXT-Stein innerhalb der Matlabumgebung zu programmieren und darüber die Motoren und Sensoren anzusprechen.

III. KONZEPT UND REALISIERUNG

A. Mechanische Konstruktion

Der prinzipielle Aufbau des Kartenmischers (s. Abb. 3) besteht aus zwei Fächern für die beiden Kartenstapel die gegenüber voneinander liegen. Zwischen diesen Fächern befindet sich etwas tiefer das Fach für den gemeinsamen, gemischten Stapel. Dieses Fach lässt sich nachdem der Mischvorgang beendet ist, mit dem neuen Stapel hinausnehmen. Die Motoren sind genauso wie drei Tastsensoren mit dem NXT-Stein verbunden.

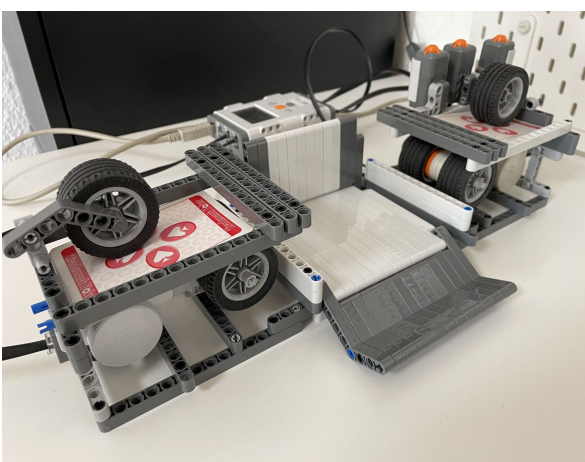


Abbildung 3. Ansicht des gesamten Kartenmischer

Unter den beiden Fächern für die ursprünglichen Kartenstapel befindet sich jeweils ein Motor, an denen sich Räder befinden, und über den Fächern ein Rad als Gewicht, welches auf den Kartenstapel drückt (s. Abb. 4).

Die Räder, die vom Motor in Rotation gebracht werden, liegen direkt an der untersten Karte des Stapels an und sorgen dafür, dass diese in das gemeinsame Fach in die Mitte bewegt werden. Das Gewicht welches von oben auf den Kartenstapel wirkt, sorgt dafür, dass die Karten enger aneinander gepresst werden. Dadurch verbessert sich die Haftung der Räder unter dem Kartenstapel und der Mischvorgang kann besser und vor allem selektiver stattfinden. Liegen die Karten nämlich zu locker auf den Rädern auf, werden sehr große Teile des Stapels und teilweise sogar fast der ganze Stapel auf einmal in das mittlere Fach bewegt. Dementsprechend würde dann nur ein sehr mangelhafter Mischvorgang entstehen.

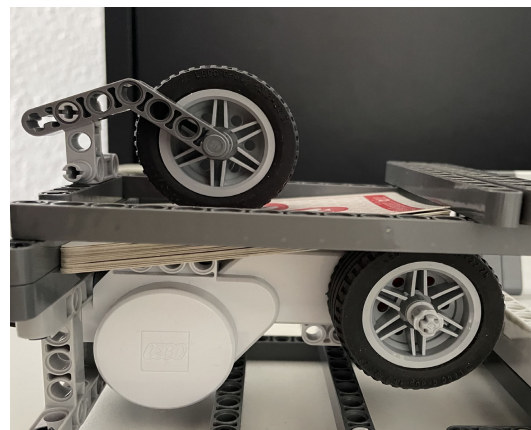


Abbildung 4. Seitliche Detailansicht eines Kartenfachs

B. Programmierung der Mischmechanismen

1) *Gleichzeitiges Aktivieren der Motoren*: Wie bereits erwähnt, soll die Maschine mit drei unterschiedlichen Mechanismen die Karten durchmischen können. In der ersten Variante sorgt das Betätigen eines Tasters, welches einer logischen 1 entspricht, dafür, dass beide Motoren mit selbem Tempo aktiviert werden. Durch das Betätigen eines zweiten Tasters können die Motoren zum Stillstand gebracht werden, wenn die Karten durchmischt.

2) *Wechselndes Aktivieren der Motoren*: In der zweiten Variante können die beiden Motoren manuell angesteuert werden. Dadurch kann der Anwender beliebig selbst bestimmen, von welchem Stapel die Karten gezogen werden. Er kann steuern, wie lange der jeweilige Motor aktiviert ist und damit auch die Menge an Karten die vom Stapel gezogen werden sollen. Realisiert wird das durch zwei Taster die, solange sie aktiviert sind, die Motoren aktivieren. Wird der Taster losgelassen, stoppt auch der jeweilige Motor.

3) *Auswahl durch Zufallsalgorithmus*: In der dritten Variante soll ein Zufallsalgorithmus dafür sorgen, dass jeweils zufällig ein Stapel ausgewählt wird, von dem eine Karte in das gemeinsame Fach befördert wird. Dafür wurde mithilfe von MATLAB eine Schleife entwickelt, in der sich wiederholend durch eine Zufallsfunktion die Zahl 1 oder 2 ausgegeben wird. Je nachdem wird dann der Motor unter Stapel 1 oder Stapel 2 aktiviert, um eine Karte davon zu ziehen. Der Vorgang läuft nach Betätigen

des Tasters automatisiert ab. Sind alle Karten im gemeinsamen Fach, kann durch einen weiteren Taster das Programm gestoppt werden. Im Folgenden ist der Programmablaufplan grafisch dargestellt:

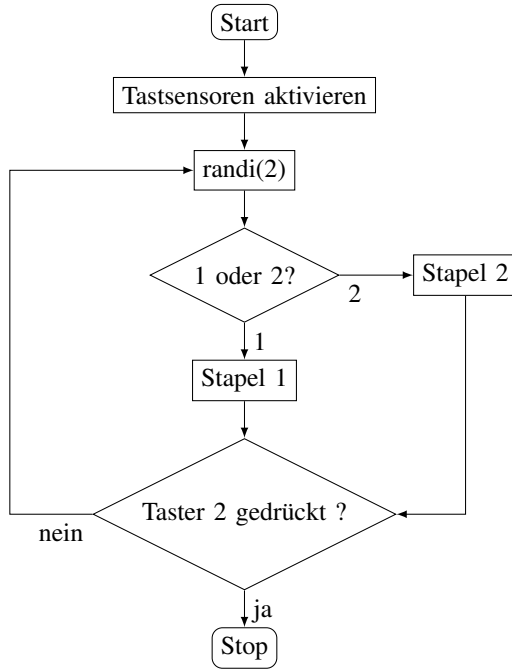


Abbildung 5. Programmablaufplan beim Mischen durch Zufallsgenerator

Dabei stellt die Operation `randi(2)` die Funktion dar, mit der MATLAB für jeden Schleifendurchgang zufällig eine 1 oder 2 ausgibt.

IV. ERGEBNISDISKUSSION

Der fertige Kartenmischer konnte sowohl die Karten in das gemeinsame Fach bewegen, als auch die drei verschiedenen Programme zum Durchmischen der Karten ausführen [4]. Allgemein lagen die Probleme eher in der mechanischen Konstruktion begründet, da die Programmierung des Vorgangs vergleichsweise simpel war und die Funktionen auch wie gewünscht ausführen lassen konnte.

Die erste Variante, in der die Motoren gleichzeitig aktiviert wurden, funktionierte gut und zuverlässig, wie in der Vorbetrachtung angenommen. Das Ergebnis war ein gut durchmischter Stapel im davor vorgesehenen Fach und der Roboter führte dies zuverlässig bei mehreren Durchgängen hintereinander durch.

Die zweite Variante, in der die Stapel manuell ausgewählt werden konnten, funktionierte ebenfalls gut.

Die Funktion in der dritten Variante per Zufallsgenerator funktionierte prinzipiell auch gut. Das Problem lag nur darin, dass mechanisch nicht gut genug die Karten angesteuert werden konnten. Das eigentliche Ziel war es immer nur eine Karte pro Schleifendurchgang vom Stapel in das gemeinsame Fach zu ziehen. Teilweise funktionierte das auch gut, aber oft kam es auch dazu, dass mehrere Karten herausgezogen wurden oder auch nur teilweise herausgezogen werden konnten. Ursache dafür ist, dass es nicht möglich war, alleine durch

die mechanische Konstruktion immer nur gezielt eine Karte auszuwählen. Der Einsatz von zusätzlichen Sensoren um das gewünschte Verhalten zu erreichen, war im Rahmen des LEGO-Mindstorms-NXT-Sets nicht möglich.

V. ZUSAMMENFASSUNG UND FAZIT

Abschließend lässt sich sagen, dass die ursprüngliche Idee des Kartenmischer realisiert werden konnte und auch die verschiedenen Mechanismen zum Durchmischen prinzipiell gut funktionierten. Die bereits erwähnten Probleme waren hauptsächlich durch die mechanischen Gegebenheiten bedingt. Das größte Verbesserungspotential liegt darin, die Karten einzeln auswählen zu können. Dies könnte z.B. mit besseren Sensoren, als sie das LEGO-Mindstorms-NXT-Set, bietet, realisiert werden. Erweiterungsmöglichkeiten der Idee gibt es beim Ausgeben der Karten, sodass diese aus dem gemeinsamen Fach auch automatisiert ausgegeben werden können. Im Idealfall können die Karten dabei auch einzeln ausgewählt werden. Darüber hinaus könnte der Kartenmischer durch einen kompakteren Aufbau mobil gemacht werden, sodass er sich z. B. auf einem Tisch bewegen und die Karten direkt vor den Spielern ausgeben kann.

LITERATUR- UND QUELLENVERZEICHNIS

- [1] Pixahive.: <https://pixahive.com/photo/cards-shuffling/>
- [2] Wikimedia Commons.: https://commons.wikimedia.org/wiki/File:Lego_Mindstorms_NXT_2.0_-_Stein_und_Sensoren.png
- [3] Mathworks: RWTH Aachen Toolbox. <https://de.mathworks.com/matlabcentral/fileexchange/18646-rwth-mindstorms-nxt-toolbox>
- [4] YouTube, Mathias Magdowski, Playlist LEGO-Praktikum 2022, Kartenmischroboter aus dem Lego-Praktikum 2022 an der OVGU Magdeburg mischt Spielkarten.: <https://www.youtube.com/watch?v=6MJdsQAHln0>

Farbsortierroboter

Hassan Al-Homada, Elektromobilität
 Otto-von-Guericke-Universität Magdeburg

Zusammenfassung— - Jedes Jahr im Wintersemester wird das LEGO-Mindstorm- Projektseminar von dem Otto von Guericke Universität Magdeburg durchgeführt. Die Studierende, die an dem Seminar teilnehmen, werden in Gruppen verteilt, jede Gruppe muss ihre eigene Projektidee haben, welche ausgearbeitet, realisiert und am Ende vor einer Jury und den anderen Studierende vorgestellt werden. Eine Gruppe davon hat sich entschlossen, einen Sortierroboter aufzubauen. Die Aufgabe war LEGO-Bausteine zusammensetzen und anschließend durch einen Mechanismus nach Farben einzusortieren.

Schlagwörter— Farbsortierroboter, LEGO, NXT, Otto-von-Guericke Universität

I. EINLEITUNG

IN der Literatur und anderen Medien wird der Roboter vor allem als „Maschinenmensch“ thematisiert beziehungsweise als autonomes Maschinenwesen, das dem Menschen als Helfer oder aber als Bedrohung gegenübersteht. In den folgenden Jahren sind Maschinen unverzichtbar geworden und nicht mehr aus der Wirtschaft und dem privaten Haushalt wegzudenken. Roboter bieten zahlreiche Vorteile an. Sie erledigen die Arbeit schneller als die Menschen, sind billiger anzuschaffen und verursachen keine Fehler. Ein Typ von Robotern, welcher von Menschen eingesetzt wird, ist der Sortierroboter. Die haben die Aufgabe, Objekte an ihre richtige Stelle zubringen. Der Sortierroboter wird heute überall auf der Welt eingesetzt z.B. in Müllentsorgungsprojekten, da werden sie verwendet, um die verschiedene Abfälle artmäßig zu sortieren. Deshalb war das Ziel im Projektseminar, einen Sortierroboter zu entwickeln [1].

II. VORBETRACHTUNGEN

Drei Motoren, einen Farbsensor, das NXT-Gerät und verschiedene LEGO-Bausteine. dies sind die Hauptteile, aus denen ein Sortierroboter besteht.

A. Motoren

Es wurde überlegt, drei Motoren in den Roboter einzubauen. Der erste Motor (A) ist dafür zuständig, dass sich der Greifarm horizontal, also nach links und rechts bewegen kann. Der zweite Motor (B) hat die Aufgabe, die jeweiligen Objekte zu greifen. Der dritte (C) Motor ist für den Transporter eingebaut. Die Motoren sind in Abbildung 1 dargestellt.

B. Farbsensor

Ein Farbsensor ist ein fotoelektrischer Sensor, der anhand eines Senders einen Lichtstrahl ausgibt und dann anhand eines Empfängers das Licht auswertet, das vom Erkennungsobjekt zurück reflektiert wird. Ein Farbsensor kann die empfangene

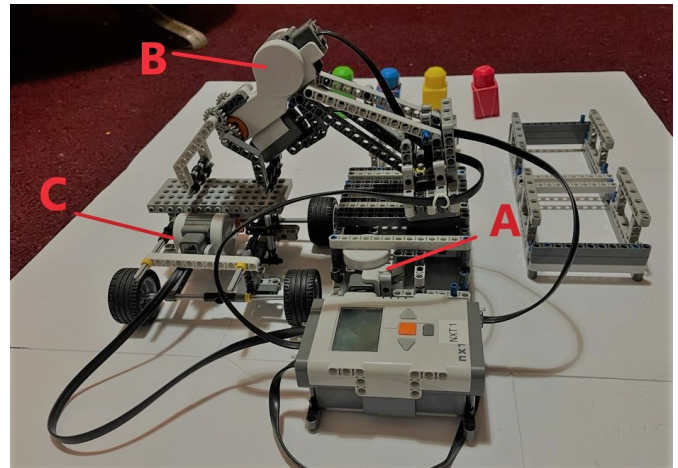


Abbildung 1. Motoren

Lichtintensität/Farbwerte für rot, grün und blau auswerten, was die Bestimmung der Farbe des Messobjekts ermöglicht. Man sieht den Farbsensor in Abbildung 2 .



Abbildung 2. Farbsensor

C. NXT-Gerät und LEGO-Baustein

Der NXT ist ein Steuerungscomputer der Produktserie LEGO-Mindstorm. Er besitzt Anschlüsse für mehrere Sensoren und Aktoren, sowie USB- und Bluetooth- Schnittstellen. Der Computer kann über spezielle Programmiersprachen wie MATLAB am PC programmiert werden. Die LEGO-Bausteine spielen auch eine große Rolle. Ohne die Bausteine wäre es nicht möglich gewesen, den Roboter zuzubauen. Dennoch waren die meisten Probleme wegen der Bausteine. Es dauert ziemlich viel Zeit bis man zur Endform des Roboters kam.

III. AUFBAU UND FUNKTIONSWEISE

Die Idee war es, einen Roboter beziehungsweise eine automatisierte Vorrichtung zu entwerfen, welche Legosteine nach ihrer Farbe sortiert. Dieser Roboter sollte Würfel, die

auf einen Transporter gelegt bzw. befördert wurden, vereinzeln können und selbstständig einsortieren können. Relativ schnell wurde die Basis des Roboters erstellt, wie in Abbildung 3 zu sehen.

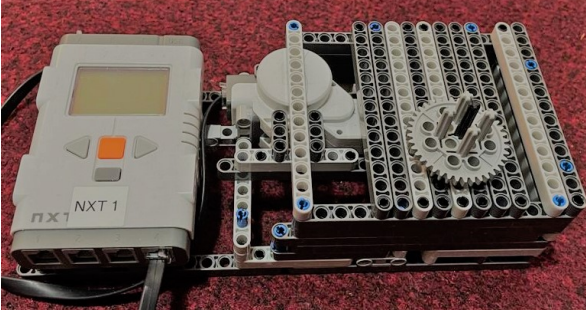


Abbildung 3. Basis des Roboters

A. Programmablauf

Ein idealer Programmablaufplan zur Erklärung eines Prozesses oder einer Routine ist in Abbildung 5 dargestellt. Zur Übertragung des Programmes muss der Roboter an der USB-Schnittstelle mit dem PC verbunden werden. Die Kabel sind alle angeschlossen, und das NXT-Gerät ist schon an. Das Programm wird gestartet. Per Hand legt man die farbigen Würfel auf den Transporter, danach fährt dieser nach vorne und parkt genau vor dem Greifer. Der Transporter ist in Abbildung 4 dargestellt. Ein Farbsensor, der direkt hinter dem Greifer eingebaut ist, entscheidet welche Farbe der Würfel hat. Ist die erkannte Farbe Rot, dann greift der Roboter den Würfel mit dem Greifer, danach bewegt sich der Arm des Roboters um 90° nach links und lässt den Würfel in den richtigen Korb fallen. Am Ende kehrt der Arm an den Nullpunkt (bzw. 0°) zurück. Inzwischen fährt der Transporter wieder nach hinten, wo man ihn manuell beladen kann. Falls aber die erkannte Farbe Blau ist, wiederholt sich dann der ganze Ablauf, aber diesmal bewegt sich der Arm um 170° nach links und wirft den Würfel in den richtigen Korb. Sollte die Farbe des Würfels aber Grün sein, dann dreht sich der Arm um 200° nach links.

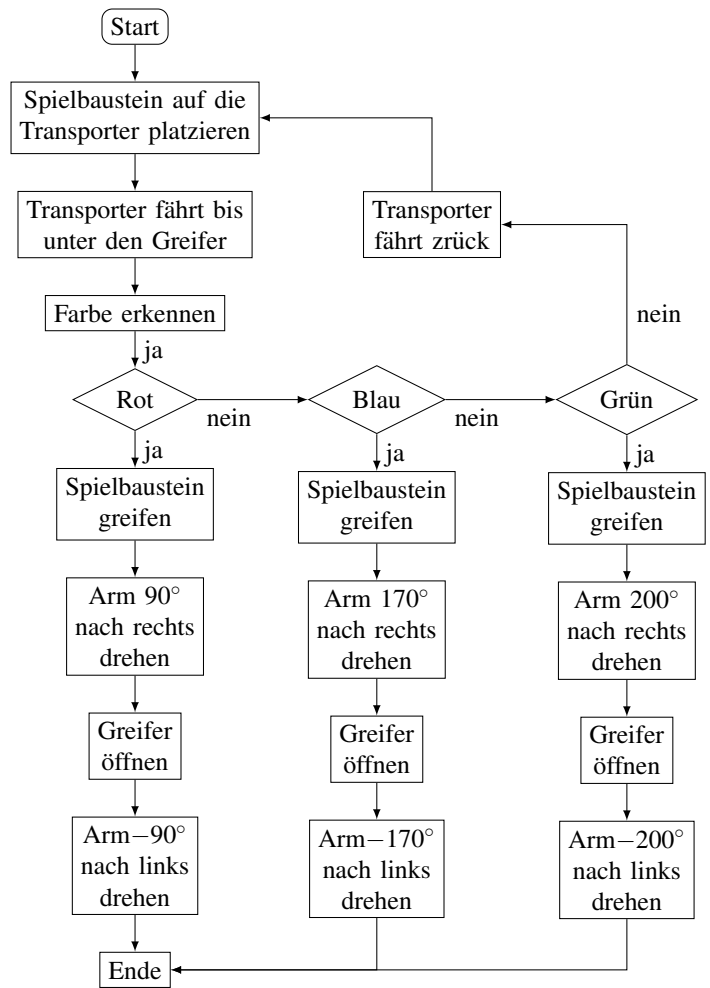


Abbildung 5. Programmablauf

B. Probleme

Die erste Version des Roboters war noch nicht perfekt und benötigte noch einiges an Optimierungen. Es wurde im oberen Bereich des Roboters ein Motor mit einem Arm aufgebaut, damit sich der Arm nach oben und nach unten bewegt. Es hat nicht funktioniert, der Motor hat sich auch kaum bewegt, weil sich der Arm wegen seines Gewichts nach vorne neigt. Um dieses Problem zu lösen, wurde der Motor entfernt und durch einen Transporter ersetzt. Der Transporter bringt die Würfel vor dem Greifer, sodass sich der Roboter nur nach rechts und nach links bewegen muss. Der Transporter ist in Abbildung 4 dargestellt. Ein weiteres Problem entstand, als der Code geschrieben wurde, denn beim ersten Versuch gab der NXT die ganze Zeit einen Ton von sich. Dies war ein Zeichen dafür, dazu viele Befehle auf einmal an den NXT gesendet wurden und er dadurch überlastet war.

Ein drittes Problem ist aufgetaucht, das die Ungenauigkeit des Farbsensors betrifft. Es kann sein, dass beim Scannen der Würfel, die Farben falsch gelesen werden oder nur schwarz erkennen. Das hat zur Folge, dass entweder die Würfel an die falsche Stelle gebracht wird, oder gar keine Bewegung stattfindet. Man kann dieses Problem vermeiden, in dem man

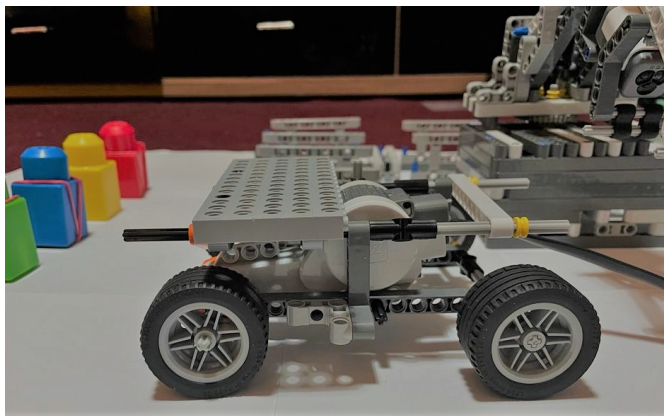


Abbildung 4. Transporter

der Farbsensor so nah wie möglich von den Würfeln einbaut.

IV. ERGEBNISDISKUSSION

Nach zwei Wochen des Projektseminars wurde ein funktionierender Farbsortierroboter gebaut, der LEGO-Bausteine nach ihrer Farbe sortiert. Das Ziel dieses Praktikums wurde erreicht. Der Roboter kann die farbigen Würfel erkennen danach sie greifen und an den richtigen Platz bringen. Eine Verbesserungsmöglichkeit wäre, dass man nicht nur den Farbsensor verwendet, sondern auch den Ultraschallsensor und den Tastsensor, Beispielsweise kann man den Ultraschallsensor auf den Transporter einbauen, damit er richtig kennt, ob der Transporter beladen ist oder nicht.

V. ZUSAMMENFASSUNG UND FAZIT

Während des Projektseminars LEGO-Mindstorms wurde erfolgreich ein Roboter entwickelt. Trotz einiger Probleme und der nicht ganz vollständigen Erreichung des ursprünglichen Ziels, kann das abgeschlossene Projekt auf einige alltägliche und einfache Aufgaben übertragen werden. Das Projekt hat dem Autor in dem Umfang viel Freude bereitet und dabei noch eine neue Programmiersprache gelehrt. Als letzter Satz soll aber festgehalten werden, dass der Roboter das zuverlässig tut, wofür er erdacht wurde und somit als ein voller Erfolg in die Geschichte des LEGO-Mindstorm Projektseminars der OVGU eingeht.

LITERATURVERZEICHNIS

- [1] Jörg Franke. *Handbuch Mensch-Roboter-Kollaboration*. Carl Hanser Verlag GmbH Co KG, 2019.

ANHANG

Für weitere Fotos während des Vorbereitungsprozesses <https://www.instagram.com/jumaazamzam/>
Die Demonstration des Roboters wurde aufgenommen und in YouTube eingestellt:
<https://www.youtube.com/watch?v=xd-P95R91HA>

Im folgenden wird der Programmcode gezeigt, mit welchem der Roboter in Bewegung gesetzt wurde:

```
clear all
close all
addpath(genpath('RWTHMindstormsNXT'))
handle=COM_OpenNXT();
COM_SetDefaultNXT(handle);
MotorA = NXTMotor(MOTOR_A);
MotorB = NXTMotor(MOTOR_B);
MotorC = NXTMotor(MOTOR_C);
port = SENSOR_4; %sensor
OpenNXT2Color(port,"FULL",handle);
display("los geht ");
pause(2)
while 1
motorC = NXTMotor(MOTOR_C,"Power",-30,"TachoLimit",500)
%Wagen fährt und haltet unter dem Greifer
motorC.SendToNXT(handle);
motorC.WaitFor()
color = GetNXT2Color(port,handle); %farabe erkennen
pause(3)
if strcmp(color,"GREEN")
motorB = NXTMotor(MOTOR_B,"Power",-10,"TachoLimit",30)
%Greifer schließt
motorB.SendToNXT(handle);
motorA = NXTMotor(MOTOR_A,"Power",-10,"TachoLimit",350)
%Drehen um 200 grad
motorA.SendToNXT(handle);
motorA.WaitFor()
motorB = NXTMotor(MOTOR_B,"Power",10,"TachoLimit",30)
```

```
%Greifer öffnet
motorB.SendToNXT(handle);
motorA = NXTMotor(MOTOR_A,"Power",10,"TachoLimit",350)
%Drehen um -200 grad
motorA.SendToNXT(handle);
motorA.WaitFor()
motorC = NXTMotor(MOTOR_C,"Power",30,"TachoLimit",500)
motorC.WaitFor()
pause(3)
elseif strcmp(color,"BLUE")
motorB = NXTMotor(MOTOR_B,"Power",-10,"TachoLimit",30)
%Greifer schließt
motorB.SendToNXT(handle);
motorA = NXTMotor(MOTOR_A,"Power",-10,"TachoLimit",300)
%Drehen um 170 grad
motorA.SendToNXT(handle);
motorA.WaitFor()
motorB = NXTMotor(MOTOR_B,"Power",10,"TachoLimit",30)
%Greifer öffnet
motorB.SendToNXT(handle);
motorA = NXTMotor(MOTOR_A,"Power",10,"TachoLimit",300)
%Drehen um -170 grad
motorA.SendToNXT(handle);
motorA.WaitFor()
motorC = NXTMotor(MOTOR_C,"Power",30,"TachoLimit",500)
motorC.SendToNXT(handle);
motorC.WaitFor()
pause(3)
elseif strcmp(color,"RED")
motorB = NXTMotor(MOTOR_B,"Power",-10,"TachoLimit",30)
%Greifer schließt
motorB.SendToNXT(handle);
motorA = NXTMotor(MOTOR_A,"Power",-10,"TachoLimit",180)
%Drehen um 90 grad
motorA.SendToNXT(handle);
motorA.WaitFor()
motorB = NXTMotor(MOTOR_B,"Power",10,"TachoLimit",30)
%Greifer öffnet
motorB.SendToNXT(handle);
motorA = NXTMotor(MOTOR_A,"Power",10,"TachoLimit",170)
%Drehen um -90 grad
motorA.SendToNXT(handle);
motorA.WaitFor()
motorC = NXTMotor(MOTOR_C,"Power",30,"TachoLimit",500)
motorC.SendToNXT(handle);
motorC.WaitFor()
pause(3)
else
motorC = NXTMotor(MOTOR_C,"Power",30,"TachoLimit",500)
motorC.SendToNXT(handle);
motorC.WaitFor()
pause(4)
end
end
CloseSensor(port,handle);
COM_CloseNXT('all').
```

Farbsortierroboter

Hamza Alali, Elektrotechnik und Informationstechnik
 Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Die Otto-von-Guericke-Universität präsentiert eine jährliche Veranstaltung für Studierende der Elektro- und Informationstechnik, um einen Roboter zu entwickeln und die dabei auftretenden Probleme zu lösen. Dies soll das Wissen der Studierende über die Bedeutung der Robotik in Gegenwart und Zukunft erweitern. Die Idee war, im Projektseminar einen Sortierroboter zu entwickeln. Der folgende Artikel zeigt, wie ein Sortierroboter mit LEGO-Steinen, zugehöriger Ausrüstung und MATLAB implementiert wird. Im Laufe des Projekts wurde ein Roboter entwickelt, der die Spielbausteine ihrer Farbe nachsortieren kann.

Schlagwörter—Farbsortierroboter, LEGO, Motoren, NXT Gerät, Sensor

I. EINLEITUNG

ROBOTER spielen in der heutigen Welt eine große wichtige Rolle, Sie werden fast überall genutzt. Man kann es sich einfach nicht vorstellen, auf sie zu verzichten. Das liegt daran, dass Roboter viele Vorteile haben. Der wichtigste Vorteil von Robotern am Arbeitsplatz ist der Kostenfaktor. Roboter sind viel billiger als menschliche Arbeitskräfte und die Kosten sinken zunehmend. Sie können überall und unter allen Umgebungsbedingungen arbeiten, sei es im Weltraum, unter Wasser, bei extremer Hitze oder Wind. Das Beste an Robotern ist, dass sie nie müde werden und buchstäblich rund um die Uhr an bestimmten Aufgaben fehlerfrei arbeiten können. Das menschliche Gehirn wird dagegen schläfrig, wenn es kontinuierlich arbeitet [1].

Es gibt viele Typen von Robotern, die verschiedene Aufgaben erledigen. Ein Typ von diesen sind, Sortierroboter, die Objekte nach bestimmten Kriterien sortiert. Sie werden überall eingesetzt. Ein Beispiel dafür ist der Sortierroboter mit einer Maschine, die gleichzeitig bis zu sechs verschiedene Sorten von Wertstoffen aus dem Materialstrom aussortieren kann. Erkennt werden Flaschen, durchsichtig farbig, Polypropylen, gemischte Kunststoff-Verbindungen, Aluminium, Papier, Kartonagen, Störstoffe, schwarze Kunststoffe, usw. Die Sortiertiefe ist damit unschlagbar hoch und effizient [2].

Aus diesen Gründen war das Ziel im Projektseminar einen Sortierroboter zu entwerfen, der beliebige Objekte nach Farbe sortieren kann

II. AUFBAU UND FUNKTIONSWEISE

In diesem Abschnitt wird es schrittweise erklärt und diskutiert, wie der Roboter aufgebaut sowie programmiert wird und wie er funktioniert.

A. Aufbau

Der Roboter besteht grundsätzlich aus 3 Motoren, einem Farbsensor, NXT-Gerät und LEGO-Bausteinen.

DOI: 10.24352/UB.OVGU-2022-030

Lizenz: CC BY-SA 4.0

1) **Motoren:** Die drei Motoren haben folgende Aufgaben, Motor (1) wird in die Basis des Roboters eingebaut, siehe Abbildung 1, ist dafür zuständig, den Geifer nach links und rechts zu drehen.

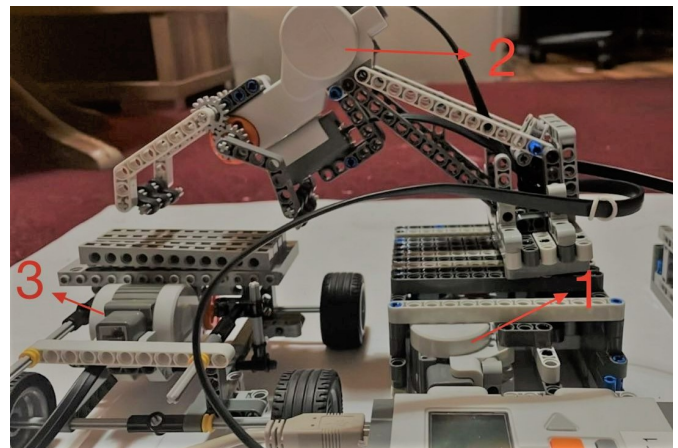


Abbildung 1. Motoren

Motor (2) ist für das Greifen von Spielbausteinen. Der Greifer ist durch einen festen Arm mit der Basis verbunden und die Spitze des Greifers ist aus Gummi aufgebaut, siehe Abbildung 2. Das ermöglicht den Spielbaustein richtig zu greifen, so dass dieser während der Bewegung nicht raus fällt.

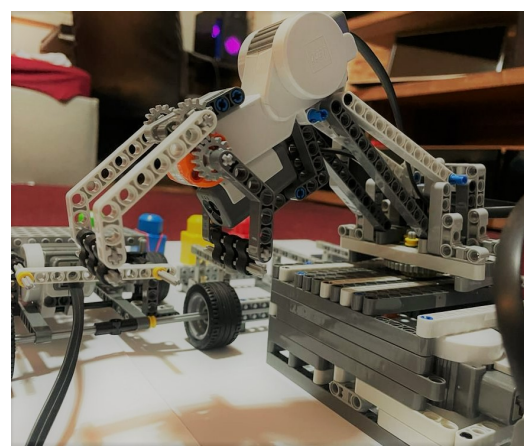


Abbildung 2. Greifer

Motor (3) ist in Transportbasis eingebaut, siehe Abbildung 3. Das ermöglicht die Transportbasis nach vorne und hinten zu bewegen. Damit ist es möglich, die Spielbausteine unter den Greifer zu bringen.

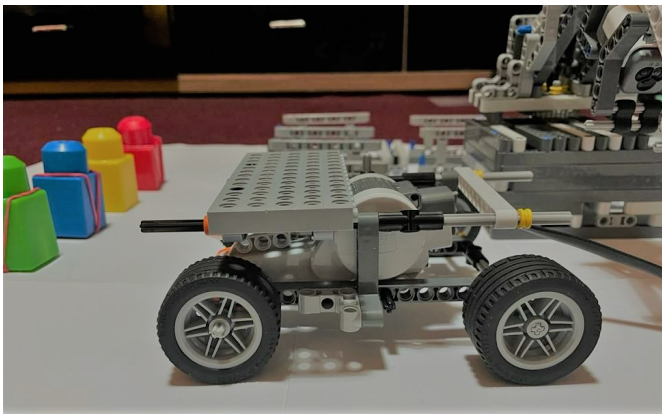


Abbildung 3. Transportbasis

2) *Farbsensor*: Der Farbsensor hat die Aufgabe, die Farbe des jeweiligen Objekts zu erkennen. Die Objekte waren drei Spielbausteinen mit unterschiedlicher Farbe (grün, blau, rot). Der Farbsensor soll die Farbe erkennen und nach dem Erkennen soll der Stein an die richtige Stelle gebracht werden. Der Farbsensor wurde unter dem Greifer eingebaut, siehe Abbildung 4 und Abbildung 2. Das hat den Vorteil, die Farbe richtig zu erkennen.

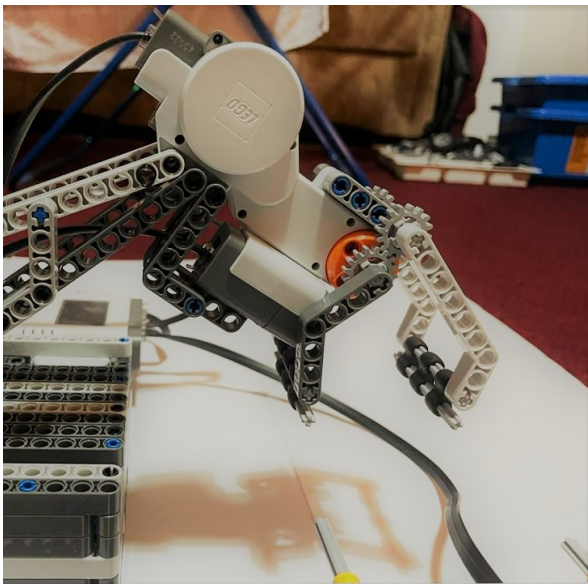


Abbildung 4. Position des Farbsensors

3) *NXT-Gerät und LEGO-Bausteine*: Die LEGO-Bausteine sind die wichtigsten Sache am Roboter, da ohne die Bausteine unmöglich wäre, den Roboter zu bauen. Das NXT-Gerät dient dazu, die Befehle vom Computer zu empfangen und dann an die Motoren weiterzuleiten. Es agierte als Übersetzer zwischen dem programmierten Script in MATLAB und dem Roboter.

B. Konstruktion

Der Sortierroboter wurde aus 4 Teilen aufgebaut, siehe Abbildung 6. Der unterste Teil (A) ist die Basis des Roboters.

Sie ist am Boden befestigt und wurde so konstruiert, damit sie stabil und stark genug ist, um der Arm-Teil (B) ohne Probleme bewegen zu können, ohne dabei zur Seite oder nach vorne und hinten zu kippen. In die Basis wurde der Motor (1) eingebaut, siehe Abbildung 1. Der zweite Teil (B) des Roboters ist der Arm. Er sollte so konstruiert werden, dass er fest ist. Am Ende des Armes wurden der Motor (2) und der Greifer eingebaut. Er sollte die passende Länge haben, damit er beim Greifen den Spielbaustein nicht gegen den Rest des Konstrukts schlägt. Die Bewegung des Arms wurde durch Motor (1), der in die Basis eingebaut ist, realisiert. Der dritte Teil (c) des Roboters ist die Transportbasis. Sie ist dafür zuständig, die Objekte unter den Greifer zu bringen. Für die Bewegung der Transportbasis wurde Motor (3) genutzt. Der letzte Teil (D) sind die Kästen, in welche die Spielbausteine sortiert wurden. Nach dem Erkennen der Farbe soll sich der Greifer schließen und die Spielbausteine in den jeweils richtigen Kasten werfen.

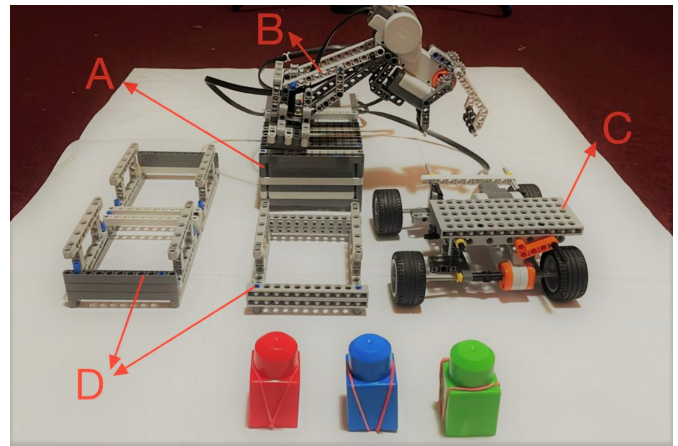


Abbildung 5. Gesamte Form des Roboters

C. Programmablauf

Das Programm wird gestartet, in dem auf Run im MATLAB-Programm geklickt wird. Dann platziert man den Spielbaustein auf die Transportbasis. Nach 4 Sekunden fährt die Transportbasis bis unter den Greifer im nächsten Schritt wird der Farbsensor die Farbe erkennen. Der Roboter ist so programmiert, dass er nur drei Farben erkennt (grün, blau, rot). Falls ein Spielbaustein mit einer anderen Farbe auf die Transportbasis hingestellt wird, fährt die Transportbasis immer wieder zurück und wiederholt diesen Vorgang, bis es ein Spielbaustein von diesen drei Farben ist. Wenn es die Farbe Rot ist, wird der Greifer schließen und den Arm nach recht um 90° drehen, wo die Kiste ist. Dann öffnet der Greifer und der Arm kehrt an den Nullpunkt zurück. Falls die erkannte Farbe grün ist, schließt der Greifer und dreht den Arm in diesem Fall um 200° nach rechts, öffnet den Greifer, damit der Spielbaustein in die richtige Kiste geworfen wird. Dann dreht sich der Arm nach links um -200° . Bei der Farbe Blau ist der gleiche Ablauf wie beim roten und grünen Spielbaustein auszuführen mit dem Unterschied, dass der Arm um 170° nach rechts vom dem Nullpunkt gedreht wird. Ein beispielhafter Programmablaufplan zur Erklärung einer Prozedur oder einer Routine ist in Abbildung 7 dargestellt.

III. PROBLEME

Bei der Entwicklung des Roboters sind viele Probleme aufgetaucht, zum Beispiel mechanische Probleme. Es wurde zuerst versucht, einen vertikaler Arm einzubauen, der sich nach unten und nach oben anhand eines Motors bewegt, siehe Abbildung 6. Das hat nicht geklappt, da der Motor nicht stark genug war, um das Gewicht vom Greifer zu tragen. Deswegen wurde die Idee entwickelt, den Arm fest einzubauen und eine Transportbasis zu nutzen, die den Spielbaustein unter den Greifer bringt. Zum anderen ist die Ungenauigkeit des Farbsensors ein Problem. Es kann sein, dass beim Scannen der Spielbausteine die Farben falsch gelesen werden. Man kann dieses Problem vermeiden, indem man den Farbsensor in eine gute Position, so nah wie möglich zum Spielbaustein einbaut, siehe Abbildung 4. Eine weitere Herausforderung ergab sich mit dem Programmablauf. Es wurde so programmiert, dass die Transportbasis immer zurück fährt, wenn die Farbe nicht richtig gelesen (grün, blau, rot) wird.

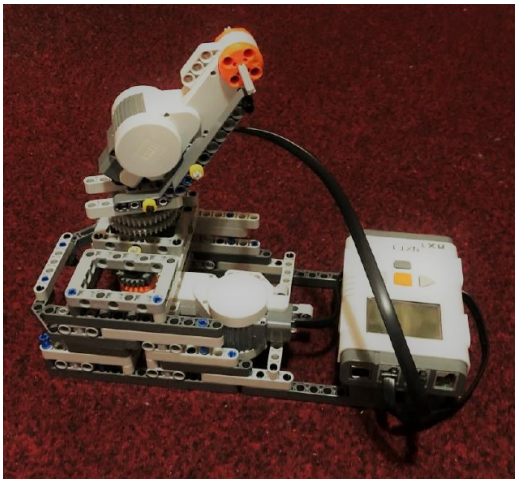


Abbildung 6. Erster Prototyp

IV. ERGEBNISDISKUSSION

Das Ziel war, einen Roboter zu entwickeln, welcher in der Lage ist, verschiedenfarbige Objekte zu erkennen und entsprechend zu sortieren. Das wurde erfolgreich umgesetzt. Der Roboter kann die farbigen Spielbausteine erkennen, greifen und an die richtige Stelle bringen. Das wurde innerhalb der vorgegebenen Zeit umgesetzt. Eine Verbesserungsmöglichkeit wäre noch mehrere Sensoren zu dem Roboter hinzufügen, zum Beispiel einen Ultraschallsensor für die Transportbasis, damit diese in der Lage ist, sich von selbst an den richtigen Platz unter den Greifer zu stellen.

V. ZUSAMMENFASSUNG UND FAZIT

Am Ende des Projektseminars wurde erfolgreich ein Roboter entwickelt, der seine Aufgaben verwirklicht hat. Trotz einiger Probleme, wurde das Ziel erfolgreich abgeschlossen.

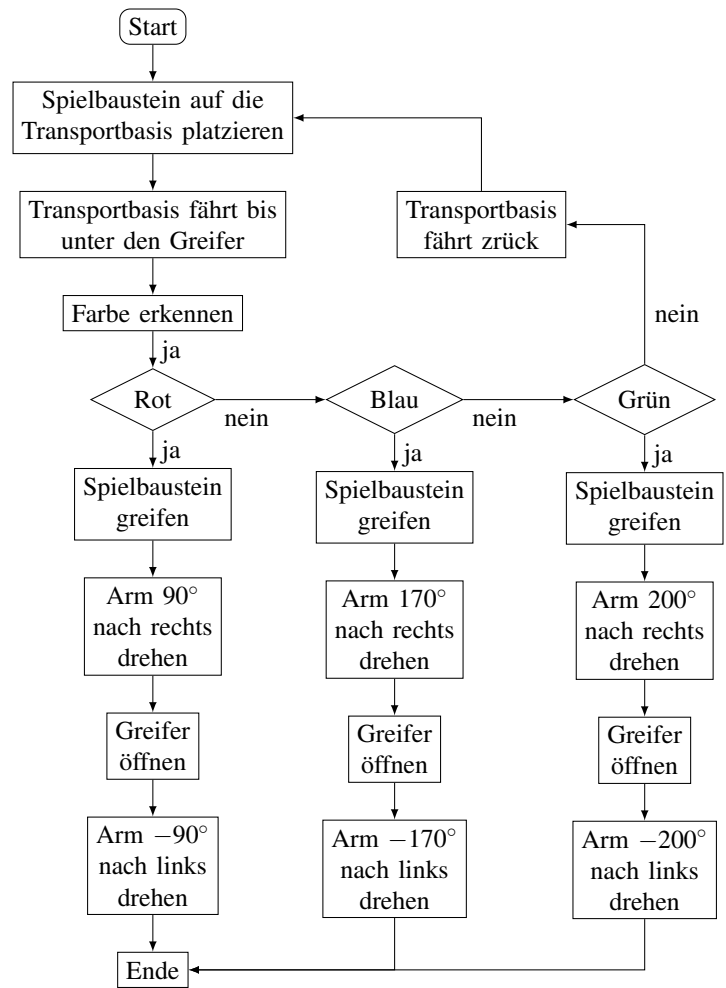


Abbildung 7. Programmablaufplan

Das Projekt erfordert Grundkenntnisse in MATLAB und die Studierenden sollten eigene Ideen integrieren können. Solche Sortierroboter werden schon in vielen Bereichen eingesetzt, wie z.B. schon in der Einleitung erwähnt wurde, für das Sortieren verschiedenen Objekte. Das kann für die Müllsortierung verwendet werden, was bei Müllentsorgung helfen kann und einen großer Vorteil zum wohle für den Menschen als auch für die Natur bringt.

LITERATURVERZEICHNIS

- [1] Tim Cole. *Erfolgsfaktor Künstliche Intelligenz: KI in der Unternehmenspraxis: Potenziale erkennen-Entscheidungen treffen*. Carl Hanser Verlag GmbH Co KG, 2020.
- [2] Jörg Franke. *Handbuch Mensch-Roboter-Kollaboration*. Carl Hanser Verlag GmbH Co KG, 2019.

ANHANG

Weitere Fotos über den Vorbereitungsprozess finden sie unter folgendem Link:
<https://www.instagram.com/jumaazamzam/>
 Die Demonstration des Roboters wurde aufgenommen und in YouTube eingestellt:
<https://www.youtube.com/watch?v=xd-P95R91HA>

Farbsortierroboter

Mohamad Jumaa Zamzam Kazak, Elektrotechnik und Informationstechnik
 Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Seit 2013 präsentiert die Universität eine jährliche Veranstaltung für Studenten der Elektro- und Informationstechnik, um einen Roboter zu bauen, der Problem löst. Dies soll das Wissen der Studenten über die Bedeutung der Robotik in Gegenwart und Zukunft erweitern. Der Roboter besteht aus LEGO-Steinen, drei Motoren und einem Farbsensor, die alle über das NXT-Gerät verbunden sind. Dieses Gerät wird von MATLAB programmiert.

Schlagwörter—Farbsortierroboter, LEGO, Motoren, NXT Gerät, Sensor

I. EINLEITUNG

ZAHNERSTAZ kommt aus dem 3D-Drucker. Malerinnen und Lackierer nutzen digitale Anwendungen, um Farben zu mischen. Kaufleute spezialisieren sich auf E-Commerce. Landwirte überwachen ihre Felder mit Hilfe digital gesteuerter Drohnen. Überall findet die Digitalisierung Eingang in die Arbeitswelt. Sie betrifft nicht mehr nur klassische IT-Unternehmen, sondern Unternehmen quer durch sämtliche Branchen und Sektoren und macht auch vor dem Mittelstand nicht Halt. Selbst kleine Unternehmen verzahnen ihre Produktion mit modernster Informations- und Kommunikationstechnik und setzen künstliche Intelligenz für lernfähige Maschinen und Roboter ein. Das zeigt: Kaum ein Ausbildungsberuf kommt heute noch ohne aktuelles Wissen über Digitalisierung aus [1]. Wie alle Arbeitsbereiche benötigen auch Maler und Lagerarbeiter einen Roboter, der Farben erkennt und an der für sie geeigneten Stelle ablegt.

II. VORBETRACHTUNGEN

Im nächsten Abschnitt wird kurz der aktuelle Stand des Farbsortierroboters vorgestellt und ein Einblick in die Teile des Roboters aus LEGO-Steine gegeben.

A. Farbsensor:

Dieser Teil des Roboters ist die Grundlage der Projektidee. Es ist so konzipiert, dass es drei Farben erkennt: Blau, Rot und Grün. Er erkennt die Farbe des Bausteins. Dann greift der Roboter den Baustein und legt ihn an der vorgesehenen Stelle ab.

B. Motoren:

Das Design dieses Roboters benötigte drei Motoren, um die Arbeit des Roboters abzuschließen. Seine Arbeit gliedert sich wie folgt:

Motor A dreht den Arm um etwa 340 Grad.

Motor B ist der Motor, durch den der Roboter den Baustein greift, nachdem er seine Farbe erkannt hat. Aber wenn der

DOI: 10.24352/UB.OVGU-2022-031

Lizenz: CC BY-SA 4.0

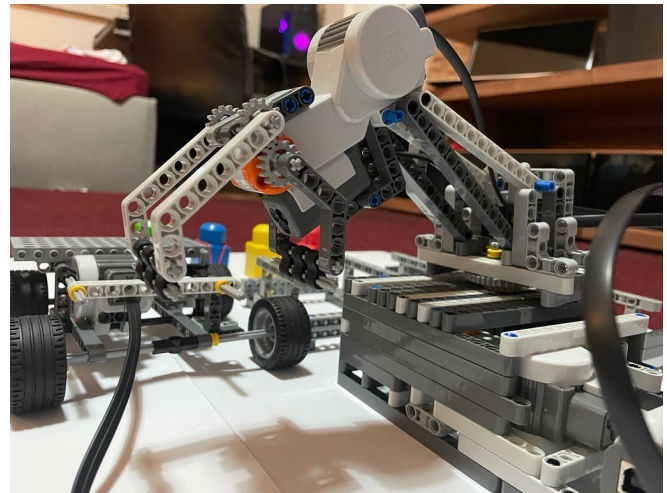


Abbildung 1. Position des Farbsensors

Baustein nicht unter den drei Farben ist, dann hält er den Baustein nicht.

Motor C ist der Motor, mit dem der Transportwagen den Baustein zum Farbsensor bewegt, am Hebel vollständig zum Stehen kommt und nach vier Sekunden an seinen Platz zurückkehrt.

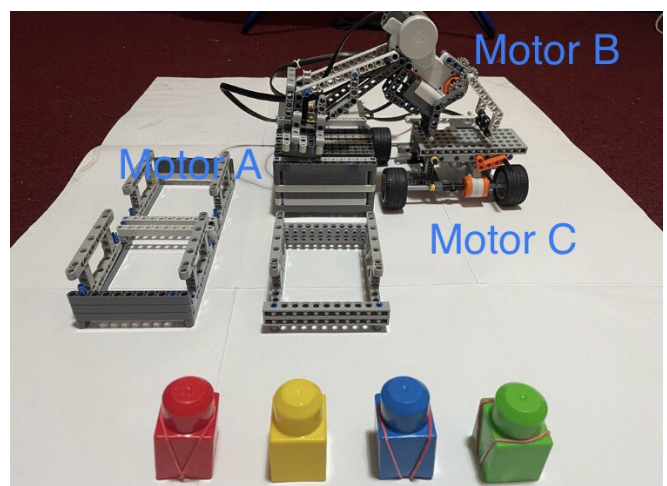


Abbildung 2. Motoren A, B und C

C. NXT-Gerät:

Der NXT ist das Robotergehirn, mit dem der Roboter programmiert und alle Motoren daran angeschlossen werden konnten.

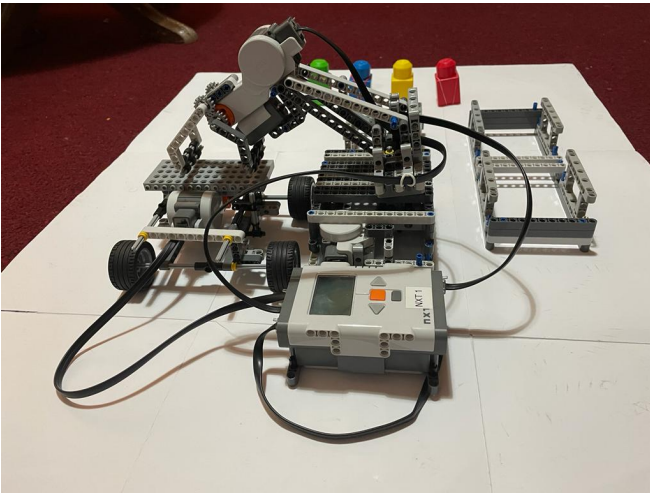


Abbildung 3. NXT-Gerät

D. LEGO-Steine:

Die LEGO-Steine waren das größte Problem des Roboters, da der Prototyp mehrfach modifiziert wurde. Zum Beispiel muss sich der Arm nach unten bewegen, um den Stein aufzunehmen, und dann den Arm nach oben bewegen.

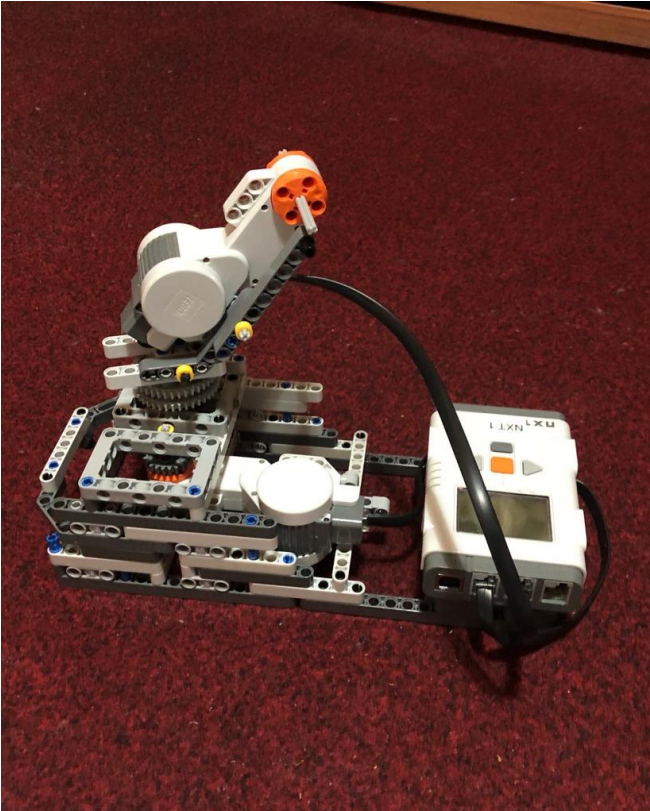


Abbildung 4. Prototyp 1

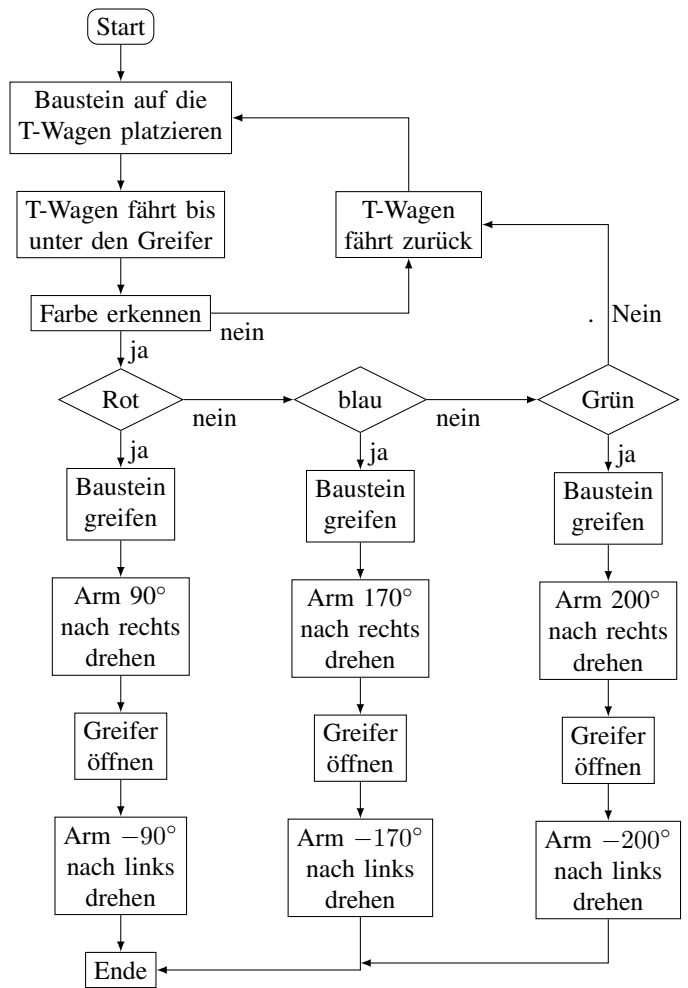


Abbildung 5. Programmablauf

III. AUFBAU UND PROGRAMMIERUNG:

A. Teile des Roboters und Probleme jedes Teils:

Der Sortierroboter besteht aus vier Hauptteilen, nämlich der Basis mit dem Motor A, dem Roboterarm mit dem Motor B, dem Transportwagen mit dem Motor C und dem Farbsensor (siehe Abbildung 2).

1) *Basis mit dem Motor A:* Die Basis wurde von Anfang an stabil konstruiert, um sicherzustellen, dass der Roboter während der Drehung nicht wackelt. Wie Abbildung 2 zeigt, ist ein Motor A in der Basis versteckt. Beim Prototyp 1 (siehe Abbildung 4) wackelte es etwas, aber dieses Problem wurde gelöst, indem die Geschwindigkeit des Arms während der Drehung reduziert wurde.

2) *Roboterarm mit dem Motor B:* Der Roboterarm ist auf einer bestimmten Höhe befestigt, um den Durchgang des Transportwagens unter dem Greifer zu erleichtern (siehe Abbildung 2). Durch den Motor B kann der Roboter den Baustein fangen und loslassen.

Aufgrund mechanischer Probleme wurde der Motor im Prototyp 1 durch LEGO-Steine ersetzt und der Transportwagen hergestellt (siehe Abbildung 4 und Abbildung 1).

3) *Transportwagen mit dem Motor C*: Der Bau des Transportwagens war der größte Verdienst für die Lösung vieler Probleme, indem er den Baustein trägt und direkt vor den Greifer und den Farbsensor legt. Dann kehrt der Transportwagen nach vier Sekunden wieder an seine Stelle zurück.

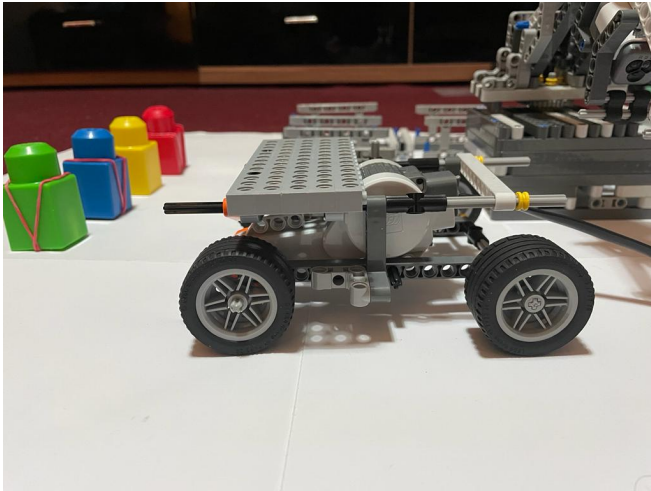


Abbildung 6. Transportwagen

4) *Farbsensor*: Der Farbsensor wurde zweimal versetzt. Das erste Mal war er auf der Basis verbaut, dort erkannte aber nur Schwarz. Daher wurde er ausgetauscht und an dem Greifer befestigt (siehe Abbildung 1).

B. Programmablauf

Die Programmierung wurde schrittweise durchgeführt, so dass die Motoren A, B und C und am Ende der Farbsensor programmiert wurden. Im Folgenden wird die Vorgehensweise des Roboters erläutert (siehe Abbildung 5).

Das Programm wird gestartet, in dem auf Run in MATLAB geklickt wird. Per Hand bringt man die Bausteine auf dem Transportwagen, dann fährt dieser vor den Farbsensor, sodass der Roboter die Farben gut erkennt. Zwischen 3 Farben (Rot, Blau und Grün) muss unterschieden werden. Ist die erkannte Farbe Rot, greift der Roboter den Baustein mit dem Greifer. Dann bewegt der Arm sich dann um 90° nach rechts und lässt den Baustein in den angegebenen Platz fallen. Am Ende kehrt der Arm an den Nullpunkt zurück. Falls aber die erkannte Farbe Blau ist, wiederholt sich dann der ganze Ablauf, aber diesmal bewegt sich der Arm um 170° nach rechts und lässt den Baustein in den angegebenen Platz fallen. Sollte die Farbe des Bausteins Grün sein, dann dreht sich der Arm um 200° nach rechts, und lässt den Baustein in den angegebenen Platz fallen.

IV. ERGEBNISDISKUSSION

Der Roboter kann jetzt tun, was geplant war. Aber es hätte weiterentwickelt werden können, wenn es mehr Motoren gegeben hätte. Insgesamt weist die finale Version kein ungelöstes Problem auf.

Durch diese Erfahrung wurde die Angst genommen, einen Roboter zu entwerfen, was ansonsten ein Problem war. Es war

gut, MATLAB zu lernen und praktisch auf einen Robot er anzuwenden.

V. ZUSAMMENFASSUNG UND FAZIT

Am Ende kann man sagen, dass der Roboter seine Arbeit gut macht. Er könnte auch zu einem mobilen Roboter ausgebaut werden, indem der Basis ein Motor hinzugefügt und die Basis erweitert wird. Ebenso könnte das Programm so entwickelt werden, dass es automatisch ausgeführt wird, wenn man eine bestimmte Farbe zum Suchen auswählen. Der Greifer kann auch so entwickelt werden, dass er zum Tragen von Gegenständen verschiedener Größen und Formen verwendet werden kann.

LITERATURVERZEICHNIS

- [1] PETER ALTMAIER: *ARBEIT DER ZUKUNFT*. <https://www.zeitbild.de/arbeiterzukunft/>. Version: Mai 2019

ANHANG

Der Vorbereitungsprozess während des Baus des Roboters ist auf folgender Seite dokumentiert:

<https://www.instagram.com/jumaaazamzam/>

Das Ergebnis ist in folgendem Video dargestellt:

<https://www.youtube.com/watch?v=xd-P95R91HA>