

# 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

**R**OBOTER 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 ansteuern 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) *Lichtsensoren*: 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 weiße<sup>1</sup> Oberflächen unterscheiden.

2) *Farbsensoren*: 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.

<sup>1</sup>Natürlich 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.

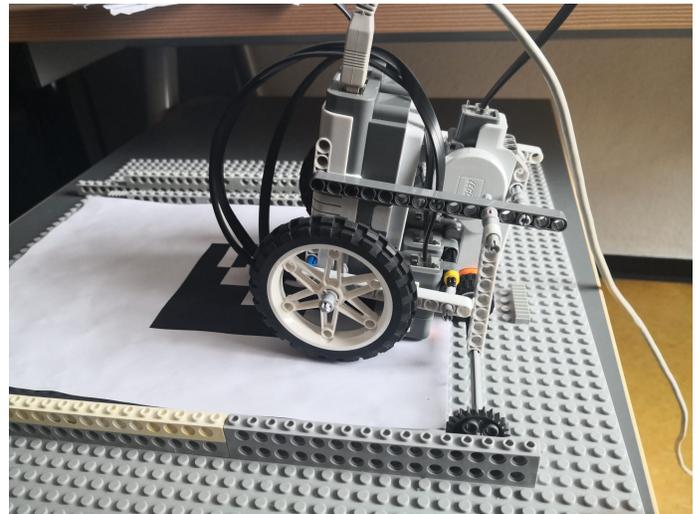


Abbildung 2. Seitenansicht vom Roboter

**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.

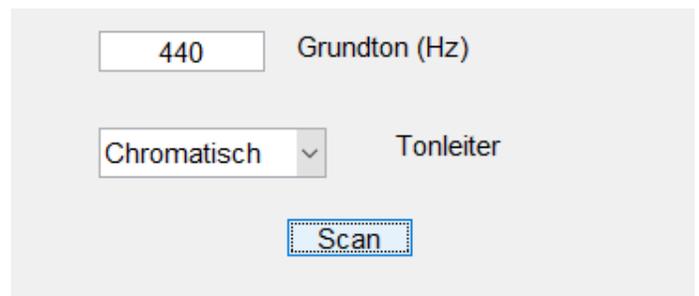


Abbildung 3. GUI zur Einstellung des Grundtons und der Tonleiter

**C. Kodierung**

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

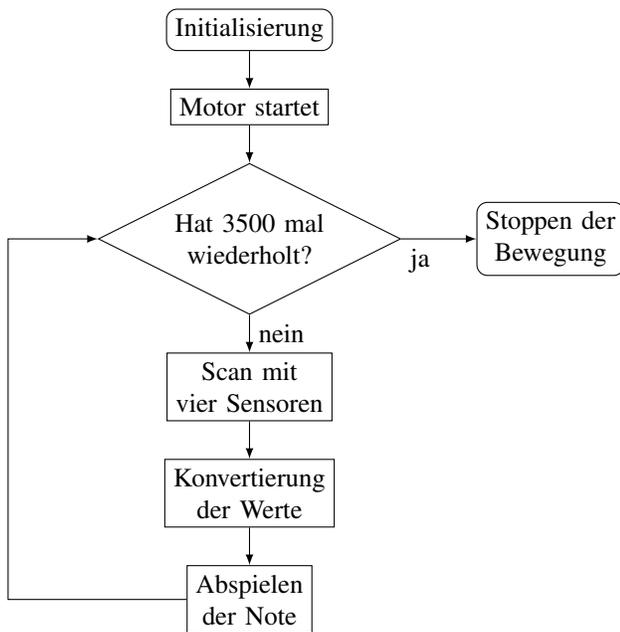


Abbildung 4. Programmablaufplan des Roboters

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 niedrigwertigstem 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 anfangen 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 Halbtonen 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 Halbtonen 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, dass 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 Tonalitäten: Dadurch, dass nur ein Ton gleichzeitig abspielbar ist, lassen sich keine Akkorde spielen. Da der Roboter keinerlei Nebentöne erzeugt, (welche die Musikinstrumente im wesentlichen unterscheiden), 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 folgenschwere Grenze: der niedrigste Ton ist ein Gis mit ca. 208 Hz, welcher am oberen Ende des Bass-Schlüssels 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.

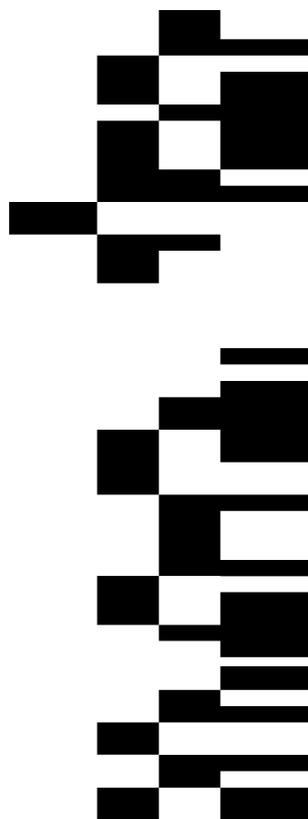


Abbildung 6. Korobeiniki kodiert in Moll, mit dem Grundton c'' (523 Hz)

#### 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>