

Entwicklung eines Soundpuzzleroboters

Matthes Schaefer, Medizintechnik
Otto-von-Guericke-Universität Magdeburg

Abstract— Im Zuge des Projektseminar Elektrotechnik/Informationstechnik wurde ein Roboter auf Basis der LEGO Mindstorms Bausätze gebaut. Dabei wurde die Version EV3 genutzt, um mit Hilfe von Matlab einen funktionstüchtigen Roboter zu bauen, welcher in der Lage ist Farben zu scannen und abhängig von der Farbe bestimmte Abschnitte bekannter Melodien zu spielen. Durch die Aneinanderreihung verschiedener Farbkarten kann somit eine vollständige Melodie gespielt werden. Zum Wechsel der Farbkarten nutzt der Roboter eine Fahrzeugautomatik, sowie eine Start-Stoppautomatik. Das Ziel des Roboters ist es eine neuartige Form des Puzzelns zu ermöglichen, die nicht auf der Anordnung von Bildteilen, sondern auf der Anordnung von Melodieteilen basiert.

Schlagwörter: Matlab, Roboter, Farbsensor, Sound

I. EINLEITUNG

Die Integration von Lernrobotern, sowie vereinfachten Programmierumgebungen in Bildungsumgebungen hat in den letzten Jahren stark zugenommen. Die Verwendung von Roboterkits wie LEGO Mindstorms bietet eine praktische Möglichkeit, Studenten oder Schülern eine Einführung in die Welt der Robotik zu ermöglichen. In diesem Zusammenhang wurde im Rahmen des Projektseminars Elektrotechnik/Informationstechnik ein Roboter entwickelt, der auf Basis der LEGO Mindstorms EV3 Bausätze konstruiert wurde. Dabei war das Ziel, einen funktionsfähigen Roboter zu entwerfen, der in der Lage ist, Farben zu erkennen und abhängig von der erkannten Farbe bestimmte Abschnitte bekannter Melodien abzuspielen. Die genutzten Melodien, wurden aus bekannten Spielen wie Zelda, den Star Wars Filmen, sowie aus der Klassik gewählt. Die Umsetzung dieses Konzepts erfolgte durch die Programmierung des EV3 mit Matlab, welches durch die von der RWTH Aachen entwickelten Schnittstelle verschiedene Sensoren und Motoren ansteuern kann. Der Roboter soll eine innovative Herangehensweise an das Puzzle-Konzept bieten, indem er Melodieteile anstelle von Bildteilen verwendet. Dies ermöglicht es den Benutzer*innen, durch die Anordnung verschiedener Farbkarten eine vollständige Melodie zu erzeugen. Um den Wechsel zwischen den Farbkarten zu ermöglichen, ist der Roboter mit einer Fahrzeugautomatik sowie einer Start-Stoppautomatik ausgestattet, welche die zwei Motoren, die zum Fahren des Roboters nötig sind, ansteuern. Das Hauptziel dieses Projekts war es Student*innen Grundlagen der Robotik, der Programmierung, sowie Projektmanagementkompetenzen zu vermitteln, indem in Form von zweier Gruppen der komplette Prozess von Ideenfindung, über Umsetzung bis hin zur finalen Präsentation freie Hand gelassen wurde.

II. VORBETRACHTUNGEN

A. LEGO Mindstorms

LEGO Mindstorms ist eine Produktreihe von LEGO, die speziell für den Bau und die Programmierung von Robotern entwickelt wurde. Sie bietet eine Plattform für Bildungszwecke, sowie für die rudimentäre Entwicklung von Robotern. Die Kits enthalten modulare Bausteine, Sensoren und Motoren, die es Benutzern ermöglichen, Roboter nach ihren eigenen Vorstellungen zu konstruieren und zu programmieren. Dabei gibt es verschiedene Gerätegenerationen, welche verschiedene Sensoren nutzen. In diesem Projekt wurde die Version EV3 genutzt. Die Programmierung erfolgt typischerweise über eine grafische von LEGO selbst entwickelte Programmiersprache, die eine einfache und zugängliche Schnittstelle bieten, insbesondere für Anfänger*innen. Um den Funktionsumfang zu erweitern, wurde in diesem Projekt eine von der RWTH Aachen [1] entwickelte Schnittstelle genutzt, die es ermöglicht mit Hilfe von Matlab Code Sensoren, sowie Motoren des Roboters anzusteuern und auszulesen um somit komplexe Projekte, welche über die eigentlichen Fähigkeiten der grafischen Programmiersprache hinausgehen zu ermöglichen.

B. Auswahl der Melodien

Aufgrund der eingeschränkten Fähigkeiten des Lautsprechers, was sowohl die Verarbeitungsgeschwindigkeit des Mindstorm Prozessors angeht als auch die Fähigkeit Töne zu erzeugen, ist es essenziell Melodien zu wählen, die zum einen bekannt sind und somit für alle potentiellen Benutzer*innen wiedererkennbar sind. Des Weiteren kann der Brick des EV3 lediglich einen Ton gleichzeitig abspielen, wodurch komplexere Überlagerungen von Melodien nicht mit abspielbar wären. Damit ist man in der Auswahl der Melodien auf bekannte, simple Melodien begrenzt, die auch mit längeren Tönen wiedererkennbar sind. Dabei wurde sich für die Melodien von Star Wars, Zelda und Für Elise entschieden. Diese Songauswahl ist im Nachhinein erweiterbar.

C. Wiedergabe von Melodien

Um auf dem EV3 Tonausgaben zu ermöglichen, muss eine Anweisung an den Brick geschickt werden, welche eine Frequenz für einen bestimmten Ton enthält, die Länge des Tons, sowie die Lautstärke, in der der Ton ausgegeben werden soll. Für die Wiedergabe ganzer Melodien muss daher vorher eine Melodie in Frequenzen, sowie Tonlänge zerlegt werden, um

im folgenden Ton für Ton an den Brick gesendet zu werden. Dafür können Umwandlungstabellen genutzt werden, die korrespondierende Frequenzen zu den Tönen liefern.

D. Ungenauigkeiten von Sensoren

Die Sensoren, welche für diesen Roboter benötigt werden, bestehen aus zwei Tastsensoren, sowie zwei Farbsensoren. Die Tastsensoren sind darauf beschränkt auszugeben, ob sie im aktuellen Moment gedrückt werden, oder wie oft sie bereits gedrückt wurden. Zweiteres wurde im Programm genutzt, um mit einem der Taster die Melodien zu wechseln und mit Hilfe des anderen Tasters die Wiedergabe, sowie den Motor zu stoppen. In der Umsetzung der Projektidee waren vor allem die Ungenauigkeiten der Farbsensoren zu berücksichtigen, da diese häufig unzuverlässig in ihren Angaben waren. Besonders bei schlechter Beleuchtung, wurden Farben falsch erkannt. Um diesem Umstand vorzubeugen, wurden zwei Sensoren eingebaut, welche im Programm miteinander abgeglichen werden.

III. SOFTWAREENTWICKLUNG

Das Programm basiert auf einem Sich nach jeder Iteration wiederholendem Schema, welches im folgenden Ablaufdiagramm (Abb. 1) veranschaulicht wird.

Als erstes muss durch den Nutzer mit Hilfe eines Tasters eine der drei Melodien (Star Wars, Für Elise, Zelda) gewählt werden. Da der Tastsensor auf zwei Modi begrenzt ist, einem Modus, in dem er nur während des drückens ein Signal ausgibt und einen Modus, in dem die absolute Anzahl des Drückens gezählt wird, musste man als erstes in den absoluten Zählmodus wechseln. Davon ausgehend wurde in dem Programm durch die Berechnung des Modulus

$n = \text{mod}(\text{tastercount}, 3)$ ein Wechsel zwischen drei Modi ermöglicht, welche jeweils einer Melodie entsprechen.

$$n = \begin{cases} 0, & \text{Starwars} \\ 1, & \text{Zelda} \\ 2, & \text{Für Elise} \end{cases}$$

Mit Hilfe von if Statements wurde jedem der drei möglichen Varianten von n eine Melodie zugewiesen, welche im Folgenden genutzt wird. Danach wurden die zwei verwendeten Farbsensoren ausgelesen und dreimal abgeglichen, falls sie in den drei Abgleichen nicht übereinstimmen, wurde der Wert eines der beiden Sensoren genutzt. Um darauf basierend einen Teil der Melodie abzuspielen, welche durch Switch Statements an die Farbausgabe der Farbsensoren gekoppelt sind. Eine typische Ausgabe wäre *blue*, welche dann z.B. *case blue* und den damit verbundenen Melodieausschnitt abspielt. Solange die Melodie gespielt wird, stoppt der Roboter, erst wenn eine Melodie beendet wird, wird die Fahrt fortgesetzt. Sobald der Farbsensor eine neue Farbe erkennt, wird der dazu korrespondierende Melodieabschnitt abgespielt und der Roboter stoppt. Das wiederholt sich bis man mit Hilfe eines zweiten Tasters, der auf dem gleichen Funktionsprinzip wie der Taster zum Melodiewechsel basiert, ein Stoppsignal ausgelöst wird und die Melodie beendet wird und der Motor in den Bremsmodus wechselt.

$$m = \begin{cases} 0, & \text{Roboter fährt, Melodie wird gespielt} \\ 1, & \text{Roboter bremst, Melodie stoppt.} \end{cases}$$

Damit das Programm im Nachhinein erweiterbar ist wurde es so aufgebaut, dass die einzelnen Melodien in eigene Funktionen platziert wurden. Dadurch ist es möglich im nachhinein weitere Melodien einzupflegen, solange sie die benötigte Formatierung für die einzelne Töne befolgen:

tone(Frequenz (in Hz), Lautstärke (in %), Länge (in ms))
Damit der Roboter erkennt, ob er weiterfahren kann, oder die Melodie noch zu Ende spielen muss wurde zudem eine Funktion implementiert, die Überprüft, ob er noch auf derselben Farbe steht, oder sich bereits auf einer neuen Farbe befindet und deshalb stoppt bis die Melodie zu Ende gespielt wurde.

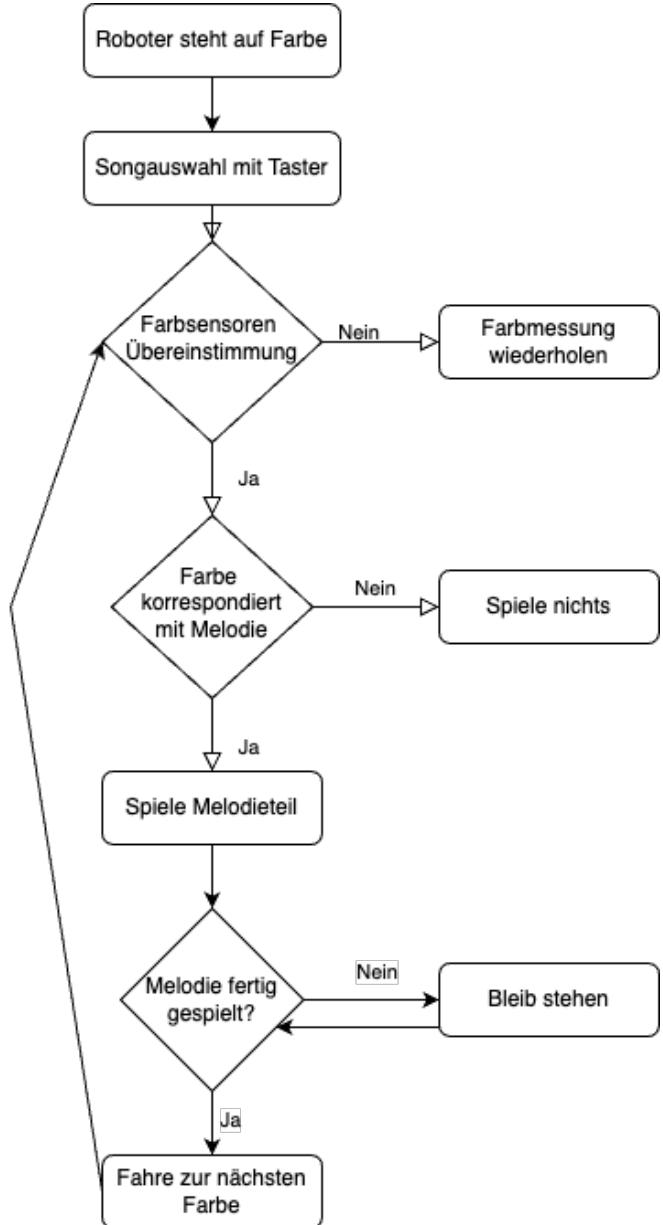


Abbildung 1 Ablaufdiagramm

IV. GRUNDKONSTRUKTION

Neben der softwareseitigen Umsetzung war es essentiell ein Gerüst zu bauen, dass sowohl zwei Sensoren als auch zwei Motoren halten kann. Nachdem in der ersten Variante des Roboters eine Differentialsteuerung gescheitert ist, weil die verbauten Teile zu Ungenauigkeiten in der Steuerung führten, wodurch der Roboter nicht in der Lage war geradeaus zu fahren und zufällig nach recht oder links abgewichen ist. Daher wurden die Ketten durch vier Räder ersetzt, was es ermöglichte, geradeaus zu fahren. Neben den essenziellen Bestandteilen war auch das Design ein großer Interessenspunkt in der Entwicklung, da ein ikonisches Design zu einem hohen Wiedererkennungswert führt und dadurch auch für ein marktreifes Produkt entscheidend ist.

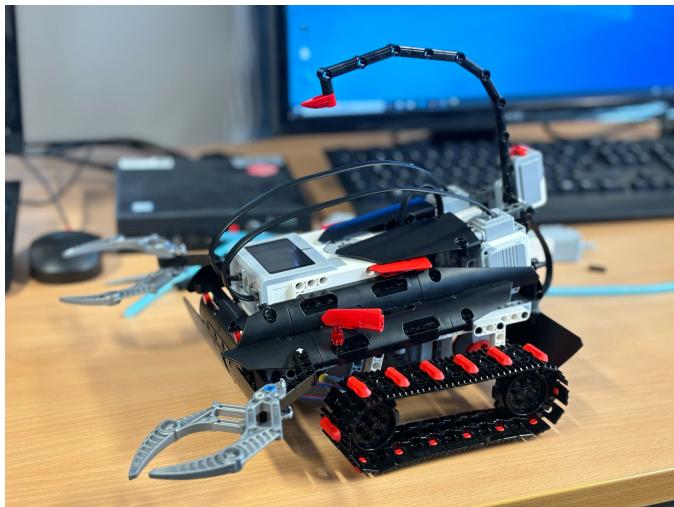


Abbildung 2 Erste Iteration

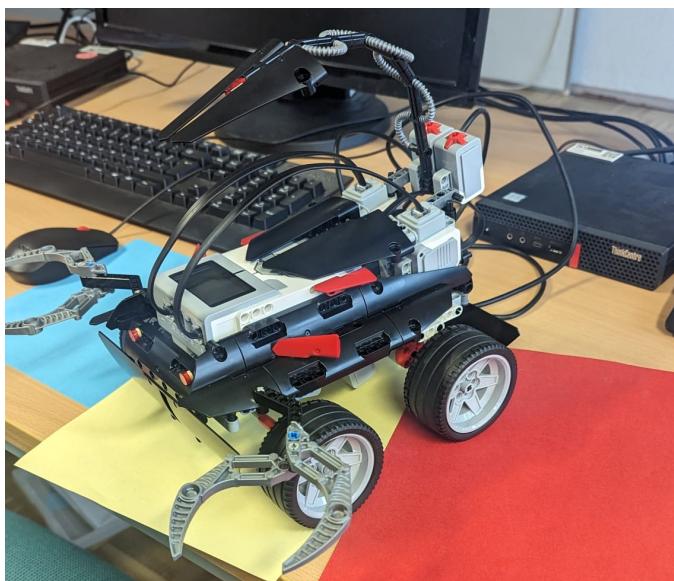


Abbildung 3 Finale Variante

V. PERSPEKTIVISCHE ERWEITERUNGEN

Aktuell sind drei Melodien eingepflegt, das kann unbegrenzt erweitert werden, unter der Bedingung, dass man für jede weitere Funktionen implementiert, die demselben Schema, wie die bereits implementierten Melodien entsprechen. Der begrenzende Faktor wäre dabei die Steuerung zwischen den Melodien, da diese durch die eingeschränkte Nutzbarkeit der Taster, wie bereits in den Vorüberlegungen aufgeführt, nur für eine geringe Anzahl an Melodien komfortabel zu nutzen ist, da man mit jedem Wechsel durch verschiedene Melodien wechseln muss und keine spezifische Melodie heraussuchen kann. Eine weitere Möglichkeit zur Erweiterung wäre der Anbau eines Lautsprechers, der zum einen eine bessere Tonqualität hat, als der in dem Steuerungsmodul vorhandene und auch komplexere Tonabfolgen mit geringerer Latenz abspielen kann. Das würde ein weiteres Feld an Melodien ermöglichen. Zur weiteren Implementierung von Melodien sollte zudem eine Zusatzfunktion im Programm gegeben sein, die es ermöglicht aus fertigen Songs umgewandelte, abspielbare Melodien zu fertigen. Das wäre eine sehr aufwendige Erweiterung, die für eine größere Melodieauswahl jedoch unumgänglich wäre. Durch einen Umbau der Steuerung wäre es in Zukunft auch möglich andere Fahrmodi einzubauen, die beispielsweise eine Änderung der Fahrtrichtung in Korrespondenz mit der ausgewählten Farbe ermöglichen.

VI. ZUSAMMENFASSUNG UND FAZIT

Die Nutzung des LEGO Mindstorms bietet einen guten Einstieg in die Roboterentwicklung. Durch die große Anzahl an kompatiblen Klemmbausteinen sind der Umsetzung verschiedenster Projekte kaum Grenzen gesetzt. Die größten Limitationen stellen die Sensoren dar, da diese zum einen nicht genau genug sind und zum anderen nur stark begrenzt in ihren Ausgabemodi sind. Daher sind komplexere Projektideen eingeschränkt, jedoch durch die Nutzung weiterer Mikrocontroller, Sensoren oder ähnlichem möglich. Die Einbindung von Matlab funktioniert gut, wenn auch häufig mit Konnektivitätsproblemen, wodurch in der Entwicklung viel Zeit für Troubleshooting eingeplant werden muss. Die größten Herausforderungen in der Entwicklung dieses Roboters, war zum einen die zuverlässige Steuerung, zum anderen der Umgang mit den Sensoren. Von der Projektidee her war das Projekt ideal um einen Einstieg in die Programmierung zu bieten, da verschiedenste grundlegende Programmierkonzepte implementiert werden mussten.

LITERATURVERZEICHNIS

[1] <https://www.lfb.rwth-aachen.de/de/academics/lectures/mindstorms/>
veröffentlicht von der RWTH Aachen (aufgerufen am 22.02.2024)