

Tastatur-Roboter

Mykhailo Zahorodniuk, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des faszinierenden Projektseminars LEGO Mindstorms im Bereich Elektrotechnik-Informationstechnik im Jahr 2024 an der renommierten Otto-von-Guericke-Universität Magdeburg wurde ein Roboter entwickelt und umgesetzt, der in der Lage ist, Texte über eine Tastatur einzugeben. Erfunden wurde nicht nur das richtige Konzept, sondern auch der Programmcode, der für die Effizienz sorgen soll.

Die Einführung in das Seminar begann mit der Erläuterung dieses Vorhabens, gefolgt von einer fundierten Einführung in die Grundlagen der Programmierung mit MATLAB. Dies bildete das unerlässliche Rüstzeug, um die Studierenden mit den essenziellen Kenntnissen und Fähigkeiten auszustatten, die für die Umsetzung ihres Roboterprojekts unabdingbar waren. Der Lernprozess erstreckte sich über verschiedene Phasen, die von der Theorie bis zur praktischen Umsetzung reichten.

Schlagwörter—LEGO Mindstorms, MATLAB, NXT-Farbsensor, Tastatur-Roboter, Textverarbeitung

I. EINLEITUNG

IM heutigen Streben nach ständiger Automatisierung und Optimierung des täglichen Lebens ist das Projectseminar ein Vorrreiter einer innovativen Lösung zur Verbesserung der Tastatureingabe. Mit dem unaufhaltsamen Fortschritt der Technologie manifestiert sich parallel dazu auch die rasant zunehmende Trägheit der Menschheit. Diese Entwicklung vollzieht sich in einem noch nie dagewesenen Tempo. Als Lösung zur Erleichterung der täglichen Routine wurde ein innovativer Tastaturroboter entwickelt, der in der Lage ist, eine vorgegebene Tastenkombination auf der Tastatur zu tippen. Nachdem er die exakte Tastenkombination programmiert hat, ist der Roboter in der Lage, in jedem Moment der Faulheit oder in Momenten der Ermüdung die erforderlichen Textzeilen mit Präzision zu reproduzieren.

Drei NXT-Motoren [1] und ein NXT-Farbsensor werden verwendet, um diesen Roboter zu bauen. Die Motoren ermöglichen die Bewegung in drei Achsen. Erster Motor wurde auf ein Fahrgestell mit vier Rädern gesetzt. Der zweite Motor bewegt den Push-Pull-Ausleger (siehe Abbildung 1) auf die Höhe der Tastatur. Und der dritte Motor ist für den Tastendruckmechanismus zuständig. Damit der Roboter in der Lage ist, die Tasten in der richtigen Reihenfolge zu drücken, während er sich bewegt, werden den richtigen Tasten auf der Tastatur verschiedene Farben zugeordnet, die vom Farbsensor des NXT auf der Tastatur erkannt werden.

Nach der Zuordnung der Farben zu den richtigen Tasten ist der Roboter in der Lage, die richtige Taste zu finden. Diese Kombination aus Technologie und Farbcodierung ermöglicht es dem Tastaturroboter, dem Benutzer zu helfen und in Momenten extremer Ermüdung immer zur Stelle zu sein. Dies steigert nicht

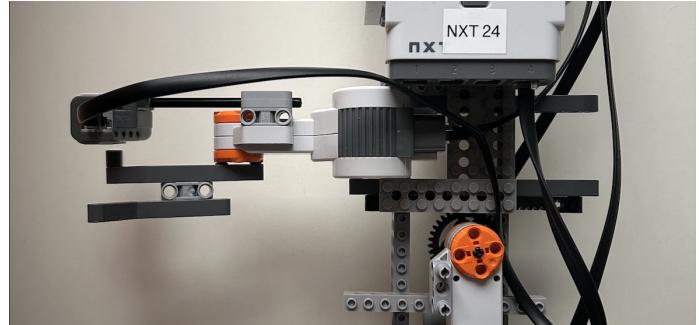


Abbildung 1: Push-Pull-Ausleger

nur die Effizienz, sondern löst auch die Probleme des modernen Lebens, indem die Technologie in den Alltag integriert wird.

II. VORBETRACHTUNGEN

A. NXT-Motor

Im Rahmen der von MATLAB [2] vorgeschlagenen Bibliothek, die für die Programmierung des Roboters verwendet wurde, stellt der NXT-Motor [3] (siehe Abbildung 2) eine umfassende Struktur dar, deren Variablen die essenziellen Eigenschaften des Motors repräsentieren. Die präzise Steuerung des Motors erfolgt durch die gezielte Veränderung dieser Variablen. Bei der Entwicklung des Tastaturroboters waren insbesondere Eigenschaften wie das Tacholimit und die Power von entscheidender Bedeutung.

Die Tacholimit-Variable spielt eine zentrale Rolle, indem sie festlegt, um wie viele Grad sich der Motor drehen kann. Hierbei wird der Wert dieser Variable im Bogenmaß ausgedrückt, was beispielsweise die erfolgreiche Navigation der Tasten in der richtigen Reihenfolge ermöglicht. Dabei entspricht 360° Grad einer vollständigen Drehung des Motors.

Die Leistungsgröße ist nicht nur für die Leistung verantwortlich, mit der der Motor arbeitet, sondern beeinflusst auch die Drehrichtung des Motors. Wenn der obigen Variablen ein negativer Wert zugewiesen wird, kehrt sie die Bewegungsrichtung um. Im Gegenteil, ein positiver Wert ermöglicht eine Bewegung in die andere Richtung. Diese Logik bietet die Möglichkeit, die Bewegungsrichtung des Roboters präzise und gezielt anzupassen.

B. NXT-Farbsensor

Der NXT-Farbsensor [4] (siehe Abbildung 3) demonstriert seine Fähigkeit, insgesamt 13 verschiedene Farben zu erkennen (siehe Abbildung 1 im Anhang). Sobald der Sensor eine Farbe identifiziert hat, gibt er den Rückgabewert als Array zurück, das wiederum bequem mit einer Zeichenkette verglichen werden



Abbildung 2: NXT-Motor



Abbildung 3: NXT-Farbsensor

kann. Der Vergleich des Werts ergibt eine logische Antwort von 1 oder 0. Diese Werte können sowohl für den Vergleich als auch für das Verlassen des Zyklus verwendet werden. Dies stellt eine elegante Möglichkeit dar, den Sensor im Dauerbetrieb für eine effektive Überwachung der Farberkennung einzusetzen.

III. ENTWICKLUNGSPROZESS

A. Konstruktion

Um die Konstruktion zu realisieren, war es erforderlich, alle drei Motoren effizient zu nutzen und somit eine Bewegung in drei Achsen zu ermöglichen. Das Design umfasste die Entwicklung eines Teils auf dem Chassis, um entlang der Tastatur zu manövrieren. Dieser Aspekt gewährleistet horizontale Mobilität und ermöglicht es dem System, alle Bereiche der Tastatur zu erfassen und zu navigieren.

Für die vertikale Bewegung auf der Tastatur wurde ein Teil unter Verwendung des zweiten Motors entwickelt. Dieser Motor setzt einen Zahnradmechanismus in Gang, der seinerseits entlang der Führungsschienen bewegt wird und so eine präzise und



Abbildung 4: Drei-Achsen-Konstruktion

reibungslose Bewegung auf verschiedenen Ebenen der Tastatur ermöglicht. Dieser Ansatz zur vertikalen Bewegung ergänzt die horizontalen Fähigkeiten der Konstruktion und schafft so eine vollständige dreidimensionale Bewegungsfreiheit.

Der dritte Motor im System erfüllt eine wichtige Funktion – er agiert als Druckmechanismus. Dieser Motor ist für die Umsetzung des Texteingabeprozesses verantwortlich und bietet präzisen und steuerbaren Druck beim Betätigen der Taste. Dies gewährleistet eine hohe Eingabetreue und zusätzliche Flexibilität bei der Anpassung der Systemparameter an die spezifischen Anforderungen des Benutzers.

Somit gewährleistet die Integration der drei Motoren in die Konstruktion umfassende Mobilität in drei Dimensionen (siehe Abbildung 4). Dieses sorgfältig durchdachte Design ermöglicht eine effiziente Interaktion der Motoren, um maximale Funktionalität und Optimierung der Texteingabe über die Tastatur zu erreichen.

B. Programmierung

Aufgrund der besonderen Art der Strukturdefinition in MATLAB ist es wichtig, die später zu setzende Tastenkombination im Voraus zu definieren (siehe Abbildung 2 im Anhang). Für die Implementierung des Codes wurde ein rekursiver Ansatz gewählt. Die Hauptfunktion beginnt mit der Initialisierung der drei Motoren, des Farbsensors und einer Struktur, die die Art des zu schreibenden Wortes widerspiegelt. Dann wird eine rekursive Funktion [5] aufgerufen (siehe Abbildung 3 im Anhang). Diese Funktion benötigt sechs Parameter, darunter

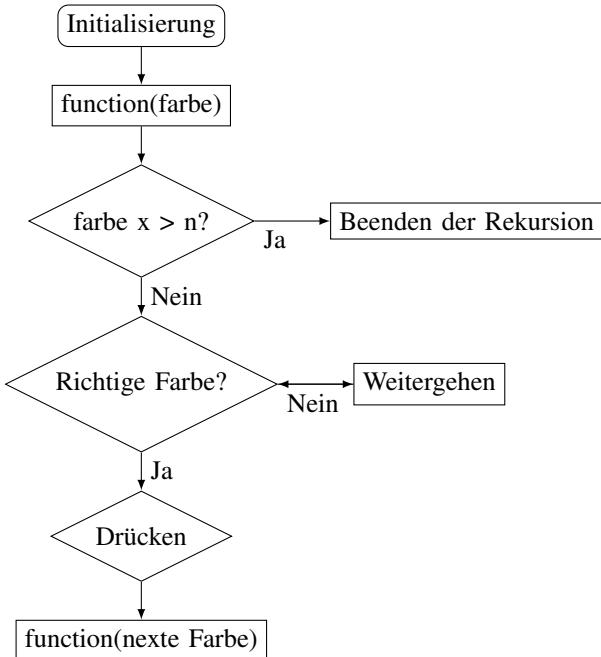


Abbildung 5: Schema des Code-Algorithmus

die drei Mechanismen, die Struktur, die aktuelle Farbe und einen Rückgabewert, der von einer nachfolgenden Funktion geliefert wird.

Die aktuelle Farbe wird durch eine Zahl dargestellt, sodass der Roboter feststellen kann, nach welcher Farbe er suchen muss. Die Anzahl der Farben ist vordefiniert und dient als Voraussetzung für das Verlassen der Rekursion. Der letzte Parameter, der Rückgabewert, gibt Auskunft darüber, ob die Zieltaste gedrückt wurde. Ist dieser Wert ungleich 1, bedeutet dies, dass die Zieltaste nicht gedrückt wurde. In einem solchen Fall muss der Roboter die Geschwindigkeit des zweiten Motors entweder erhöhen oder verringern.

Die Funktion bereitet die Motoren vor und bestimmt die Farbe, die der Roboter für die Suche verwenden soll. Sobald die Vorbereitung abgeschlossen ist, übergibt die Funktion die Kontrolle an eine andere Funktion, die eine Endlosschleife startet (siehe Abbildung 5), in der der Farbsensor aktiv ist. Wenn der Farbsensor erfolgreich die gewünschte Farbe erkennt, wird die Schleife unterbrochen und der dritte Motor tritt in Aktion und drückt die Taste. Wenn die Taste gedrückt wird, kehrt der dritte Motor in seine Ausgangsposition zurück. Die Steuerung kehrt dann zu dem Punkt zurück, an dem diese letzte Aktion in der Funktion eingeleitet wurde. Alle Elemente kehren in ihren ursprünglichen Zustand zurück, und wenn die Funktion erfolgreich abgeschlossen wurde, übergibt sie den Befehl zum Weitergehen an die Funktion, die den Text eingibt. Innerhalb derselben Funktion wird die Funktion erneut aufgerufen, allerdings mit einer anderen Farbe als der zuvor gefundenen. Dieser rekursive Aufruf wird so lange wiederholt, bis die aktuelle Farbe mit der zuvor angegebenen Bedingung für das Beenden der Funktion übereinstimmt.

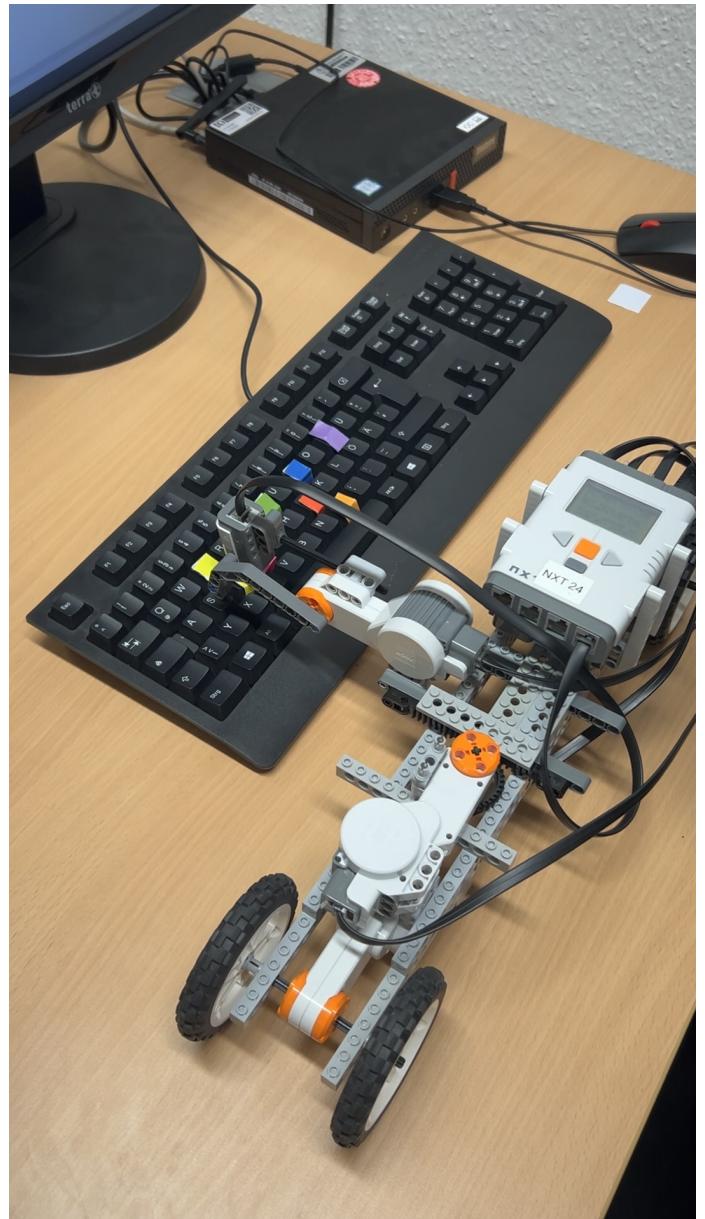


Abbildung 6: Tastatur-Roboter

IV. ERGEBNISDISKUSSION

Das Ergebnis ist ein Roboter, der Tasten mit ausreichender Geschwindigkeit und Präzision drücken kann. Während des Betriebs verfügt der Roboter über eine bemerkenswerte Fähigkeit zur Selbstkorrektur, was bedeutet, dass die Wahrscheinlichkeit, dass er die richtige Taste nicht erreicht, minimal ist.

Die Leistung des Roboters zeigt sich besonders in seiner Fähigkeit, Text effizient einzugeben. Die präzise Erkennung der Tastenpositionen durch spezielle Sensoren und die ständige Überprüfung der Position des Roboters durch einen Farbsensor gewährleisten eine außergewöhnliche Genauigkeit und Zuverlässigkeit während des gesamten Prozesses.

Der Roboter analysiert nicht nur die Oberfläche der Tastatur mithilfe spezieller Sensoren, sondern prüft auch während der Aufgabe kontinuierlich seine Position mit dem Farbsensor. Dieser Mechanismus ermöglicht es dem Roboter, seine Po-

sition automatisch anzupassen, was Genauigkeit und relative Zuverlässigkeit bei der Texteingabe gewährleistet.

Die beiden Hauptfunktionen – die genaue Erkennung der Tastenpositionen und die automatische Überprüfung der Roboterposition – arbeiten zusammen, um eine automatische Texteingabe zu ermöglichen.

V. FAZIT

Das Projekt erreichte nicht nur das Ziel, einen funktionsfähigen Roboter zu schaffen, sondern sammelte auch wertvolle praktische Erfahrungen. Zusammenarbeit, kreatives Denken und die Umsetzung von Ideen waren ebenfalls wesentliche Bestandteile dieses Projekts. Das Ergebnis ist ein ausgezeichnet realisierter Roboter, der seine gesteckten Ziele erfolgreich erfüllen kann. Dieses Projekt eröffnet nicht nur vielversprechende Perspektiven für die Weiterentwicklung der Idee, sondern verdeutlicht auch das erhebliche Potenzial für künftige Verbesserungen.

ANHANG

```
% The color index values roughly
% correspond to the following
% table (when using modes 0 and 1):
%
% 0 = black
% 1 = violet
% 2 = purple
% 3 = blue
% 4 = green
% 5 = lime
% 6 = yellow
% 7 = orange
% 8 = red
% 9 = crimson
% 10 = magenta
% 11 to 16 = pastels
% 17 = white
```

Abbildung 1: Farben, die der Sensor erkennt

```
Enter.x = 1;
Enter.y = 1;
WW.x = 315;
WW.y = 2;
NN.x = 175;
NN.y = 0;
global Letters;
Letters.enter = Enter;
Letters.W = WW;
Letters.N = NN;
```

Abbildung 2: Vereinfachtes Beispiel für eine Struktur

```
if current_color > 9
    return
end
retval = some(Letters, current_color,
    motor1, motor2, motor3, z);
if retval ~= 1
    typing(Letters, current_color, motor1
        , motor2, motor3, 0)
else
    typing(Letters, current_color + 1,
        motor1, motor2, motor3, 1)
end
```

Abbildung 3: Rekursive Funktion

LITERATUR

- [1] BRICKWIKI: *NXT Components*. <https://brickwiki.org/wiki/NXT>. Version: 2013
- [2] MATHWORKS: *MATLAB*. <https://de.mathworks.com/products/matlab.html>. Version: 2024
- [3] MINDSTORMSNXT: *NXT Motor*. <https://mindstormsnxxt.blogspot.com/2006/08/closer-look-at-nxt-motors.html>. Version: 2006
- [4] LEGOPEDIA: *NXT Farbsensor 9694*. https://lego.fandom.com/de/wiki/NXT_Farbsensor_9694. Version: 2012
- [5] MATHWORKS: *MATLAB Syntax*. <https://de.mathworks.com/help/matlab/ref/function.html>. Version: 2023