

Sortierbobby – Der Sortierroboter

Saad Al-Hamid, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Seit 2013 findet jährlich das Projektseminar der Fakultät Elektrotechnik/Informationstechnik statt, welches die Aufgabe stellt, den Studierenden einen Roboter mithilfe der Bausteine bauen zu lassen und entsprechend auch zu simulieren. Die Idee war, im Projektseminar einen Sortierroboter zu entwickeln. In dem nachfolgenden Artikel wird veranschaulicht, wie sich der Sortierroboter mithilfe der Lego Bausteine, den dazugehörigen Geräten und MATLAB realisieren lässt. Im Zuge des Projekts ist ein Roboter entwickelt worden, der Bälle entsprechend ihrer Farbe sortiert.

Schlagwörter—Farberkennung, LEGO® MINDSTORMS®, MATLAB, NXT-Gerät, Sortierroboter

I. MOTIVATION

In der heutigen Welt werden viele Arbeiten durch von Menschen erfundene Roboter erledigt, welches es unmöglich macht, sie wegzudenken. Die Roboter ermöglichen vieles, was die Menschen nicht schaffen können. Viele Menschen fürchten sich vor Robotern, doch wenn man sich mit den Robotern und ihrer Programmierung beschäftigt, so lassen sich mehr Vorteile als Nachteile herausfiltern. Die Roboter ermöglichen eine erhöhte Gesamtproduktivität, sind flexibler und agiler, ermöglichen einen geringeren Kostenfaktor, haben eine hohe Präzision und werden nie müde.

Ein Typ von Robotern, welcher von Menschen eingesetzt wird, ist der Sortierroboter. Der Sortierroboter hat, wie der Name schon sagt, die Aufgabe, Objekte an ihre richtige Stelle zu bringen. Der Sortierroboter wird heute überall auf der Welt eingesetzt. Ein Beispiel, wo Sortierroboter eingesetzt werden, sind Kliniken. In den Kliniken werden sie eingesetzt, um Essenswagen oder Medikamentenwagen zu transportieren und zu sortieren.

Deshalb war das Ziel im Projektseminar, einen Sortierroboter zu entwickeln mit den Namen „Sortierbobby“. Die Aufgabe von „Sortierbobby“ ist die Sortierung beliebiger Objekte.

II. VORBETRACHTUNGEN

Die Idee bestand darin, dass ein Sortierroboter gebaut wird. Um dies zu verwirklichen, wurden drei Motoren, einen Farbsensor, das NXT-Gerät und verschiedene LEGO®-Bausteine gebraucht.

A. Motor

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, den Greifarm nach unten und oben zu bewegen. Der dritte Motor (C) hat die Aufgabe, die jeweiligen Objekte zu greifen. Die Motoren sind in Abbildung 1 dargestellt.

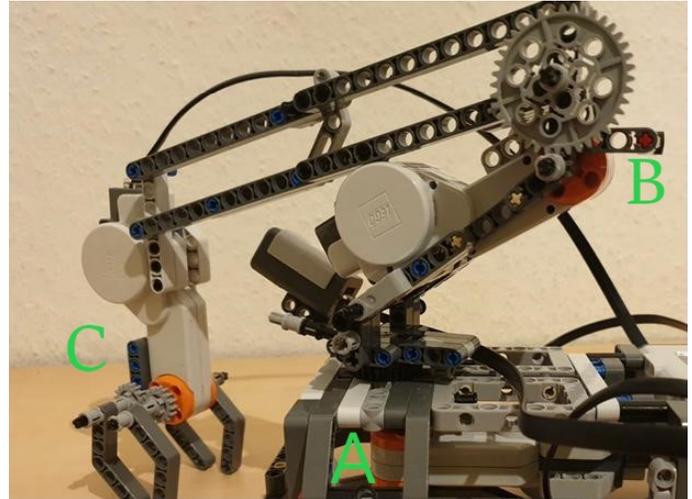


Abbildung 1. Position der Motoren A, B und C

B. Farbsensor

Der Farbsensor erkennt die Farbe des jeweiligen Objekts. Die Objekte waren drei unterschiedlich gefärbte Bälle. Der erste Ball ist rot. Der Farbsensor soll diese Farbe erkennen und dann soll der Ball an die richtige Stelle gebracht werden. Der zweite Ball ist blau. Nach dem Erkennen der blauen Kugel vom Sensor soll sich der Greifarm schließen und dann 45° nach rechts bewegen. Danach bewegt sich der Arm nach unten und schlägt den Ball mit dem geschlossenen Greifarm seitlich an. Der Roboter weigert sich sozusagen, den Ball aufzunehmen. Der dritte Ball mit grüner Farbe soll nach dem Erkennen mithilfe des Farbsensors ebenfalls an die richtige Stelle gebracht werden. Der Farbsensor ist in Abbildung 2 zu sehen.

C. NXT-Gerät und LEGO®-Bausteine

Die LEGO®-Bausteine waren der Schlüssel zu allem. Ohne die Bausteine wäre es nicht möglich gewesen, den Roboter zu bauen. Dennoch waren die meisten Probleme wegen der Bausteine. Zum Beispiel hat sich der Roboter nicht bewegt. Obwohl man dachte, das alles richtig war und schon mit dem Programmieren begonnen wurde, funktionierte er nicht richtig. Es wurde von vorne wieder angefangen und schließlich nach vielen gescheiterten Versuchen gelang es, den Roboter problemlos zu bewegen.

Das NXT-Gerät agierte als Übersetzer zwischen dem programmierten Skript in MATLAB und dem Roboter.



Abbildung 2. Farbsensor

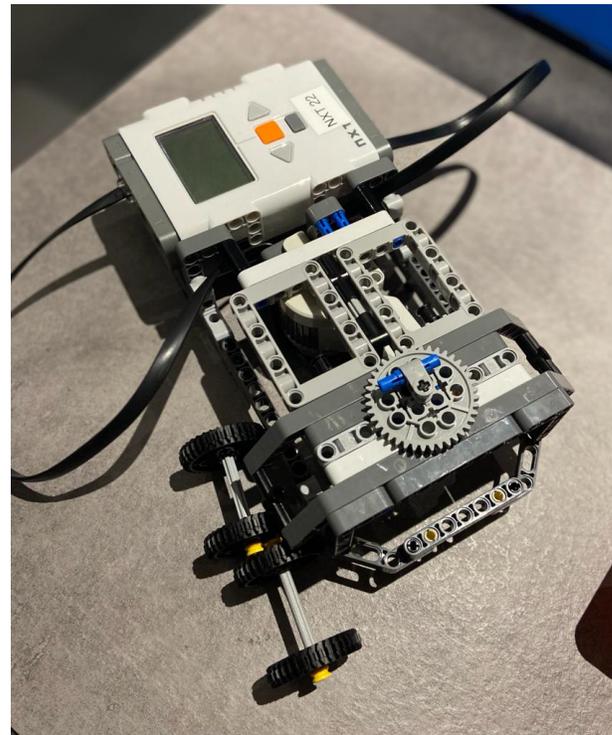


Abbildung 3. Basis des Roboters

III. HAUPTTEIL

A. Konstruktion

Der Sortierroboter wurde aus 3 Teilen aufgebaut. Der unterste Teil des Roboters war die Befestigung am Boden, damit sich der Greifarm ohne Probleme bewegen kann, ohne dabei zur Seite oder nach vorne und hinten zu kippen. Die Basis des Roboters ist in Abbildung 3 zu sehen. In der Basis war Motor A eingebaut (siehe Abbildung 1).

Der zweite Teil des Roboters war der Arm. Er sollte so konstruiert werden, dass er den Greifer nach unten und oben bewegen kann. Er sollte die passende Länge haben, damit er beim Anheben der Bälle nicht gegen den Rest des Konstrukts schlägt. Die Bewegung des Arms wurde durch Motor B realisiert (siehe Abbildung 1). Den Arm selbst zeigt Abbildung 4.

Der letzte Teil des Roboters war der Greifer. Er ist dafür zuständig, die Objekte zu greifen und anzuheben. Die Objekte, die er anhebt, sollen während der Bewegung des Arms nicht herausfallen. Motor C gewährleistet das Anheben und Greifen der Objekte. Den Greifer zeigt Abbildung 5.

B. Programmablauf

Der Roboter kann sich nicht von selbst bewegen. Dem Farbsensor mussten die farbigen Bälle hingehalten werden. Wenn dieser die Ballfarbe erkannt hat, so wurden die Bälle unter dem Greifer hingestellt.



Abbildung 4. Arm des Roboters

Nach dem Erkennen des roten Balls bewegt sich der Arm (Motor B) nach unten, um dann mit dem Greifer (Motor C) den Ball aufzunehmen. Danach bewegt der Arm den Ball nach oben. Danach bewegt sich der Greifarm 90° nach links (realisiert durch Motor A). Eine bereitgestellte Tasse dient dazu, den Ball aufzufangen, nachdem der Arm sich etwas nach unten bewegt hat und der Greifer öffnet, damit der Ball in die Tasse fällt.

Beim blauen Ball schließt sich zuerst der Greifer, bevor der Arm 45° nach rechts bewegt wird. Danach bewegt sich der Arm nach unten, um dann mit Schwung den blauen Ball wegzuschlagen.

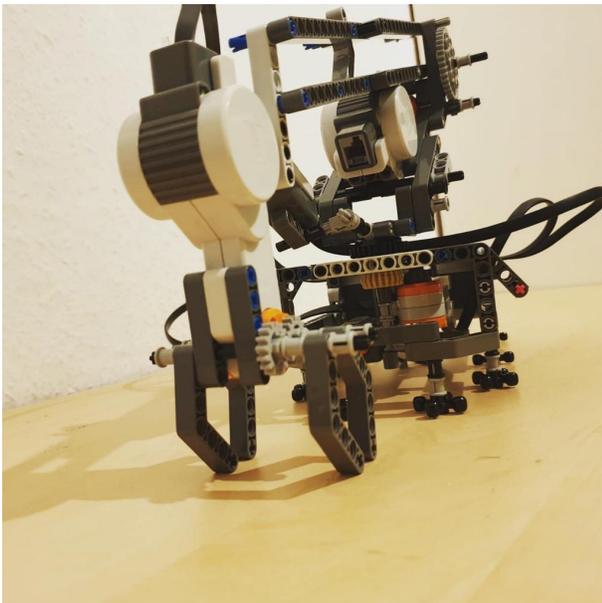


Abbildung 5. Greifer des Roboters

Wenn der Farbsensor den grünen Ball erkennt, so soll der gleiche Ablauf wie beim roten Ball ausgeführt werden; mit dem Unterschied, dass die Tasse 90° rechts vom Roboter abgestellt wird.

Ein beispielhafter Programmablaufplan zur Erklärung einer Prozedur oder einer Routine ist in Abbildung 6 dargestellt. Im Anhang befindet sich der Programmiercode in Matlab, womit sich die definierten Bewegungen vom Roboter realisieren lässt.

IV. ERGEBNISDISKUSSION

Das anfangs gesetzte Ziel, einen Roboter zu entwickeln, welcher in der Lage ist, verschiedenfarbige Objekte zu erkennen und entsprechend zu sortieren, wurde erfolgreich umgesetzt. Das Projekt wurde innerhalb der vorgegebenen Zeit umgesetzt, auch ohne große Vorkenntnisse zu diesem Thema. Eine Verbesserungsmöglichkeit wäre, dass man nicht die Bälle vor den Farbsensor halten muss, sondern dass der Farbsensor selber die farbigen Bälle analysiert. Das würde gehen, wenn mehr LEGO®-Bauteile und auch ein weiterer Farbsensor vorhanden wären. Aufgrund der pandemiebedingten Einschränkungen musste der Roboter zuhause gebaut werden, was die Lage erschwerte. Auch der Kontakt zu anderen Studierenden war eingeschränkt. Der Kontakt wäre wichtig gewesen, um sich gegenseitig helfen zu können, falls etwas nicht klappt oder wenn Bauteile fehlen.

Ein weiteres Problem war der Farbsensor. Er funktionierte nur bei bestimmten Lichtverhältnissen. Das hatte zur Folge, dass die Bälle so nah wie möglich an den Farbsensor gehalten werden mussten, damit eine optimale Aufnahme erzielt werden konnte. Wenn der Ball nicht vom Farbsensor erkannt wurde, konnte das Programm nicht ausgeführt werden.

V. ZUSAMMENFASSUNG UND FAZIT

Abschließend kann man sagen, dass die Entwicklung des Sortierroboters zufriedenstellend abgelaufen ist. Im Projekt gab

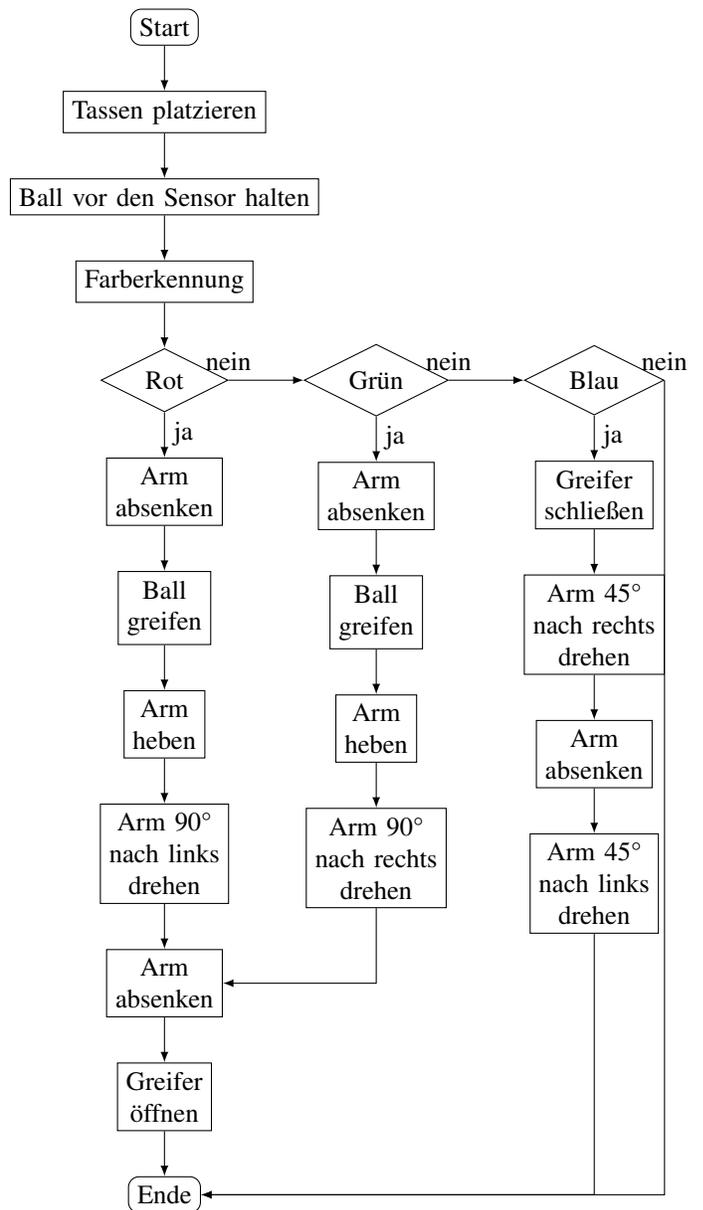


Abbildung 6. Programmablauf

es Herausforderungen, die wir bewältigen mussten. Das Projekt erfordert Grundkenntnisse in MATLAB und die Studierenden sollten eigene Ideen integrieren können.

Der Sortierroboter wird schon in vielen Bereichen eingesetzt, da er die Arbeit erleichtert und schneller die Aufgaben erledigt. Beispielsweise kann der Sortierroboter zukünftig zum Sortieren des Mülls verwendet werden, was nicht nur den Menschen einen Vorteil bringt, sondern auch der Natur.

LITERATURVERZEICHNIS

[1] A. Herz, "Die Vor- und Nachteile der Automation durch Roboter," <https://wandelbots.com/die-vor-und-nachteile-der-automation-durch-roboter/>

ANHANG

Die Demonstration des Roboters wurde aufgenommen und in YouTube eingestellt:
<https://www.youtube.com/watch?v=Jbd8RcFAnXA>

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

```

handle = COM_OpenNXTEx('Bluetooth', '001653193FC4', 'bluetooth.ini')
port = SENSOR_3;
OpenNXT2Color(port, 'FULL', handle);
pause(3)
color = GetNXT2Color(port, handle);
display('go');
pause(5)
if strcmp(color, 'GREEN')
    motorA = NXTMotor('A', 'Power', 20, 'TachoLimit', 350)
    motorA.SendToNXT(handle);
    pause(3)
    motorB = NXTMotor('B', 'Power', -20, 'TachoLimit', 20)
    motorB.SendToNXT(handle);
    pause(3)
    motorA = NXTMotor('A', 'Power', -20, 'TachoLimit', 350)
    motorA.SendToNXT(handle);
    pause(3)
    motorC = NXTMotor('C', 'Power', 20, 'TachoLimit', 300)
    motorC.SendToNXT(handle);
    pause(3)
    motorB = NXTMotor('B', 'Power', 20, 'TachoLimit', 20)
    motorB.SendToNXT(handle);
    pause(3)
    motorC = NXTMotor('C', 'Power', -20, 'TachoLimit', 300)
    motorC.SendToNXT(handle);
elseif strcmp(color, 'RED')
    motorA = NXTMotor('A', 'Power', 20, 'TachoLimit', 350)
    motorA.SendToNXT(handle);
    pause(3)
    motorB = NXTMotor('B', 'Power', -20, 'TachoLimit', 20)
    motorB.SendToNXT(handle);
    pause(3)
    motorA = NXTMotor('A', 'Power', -20, 'TachoLimit', 350)
    motorA.SendToNXT(handle);
    pause(3)
    motorC = NXTMotor('C', 'Power', -20, 'TachoLimit', 200)
    motorC.SendToNXT(handle);
    pause(2)
    motorB = NXTMotor('B', 'Power', 20, 'TachoLimit', 20)
    motorB.SendToNXT(handle);
    pause(3)
    motorC = NXTMotor('C', 'Power', 20, 'TachoLimit', 200)
    motorC.SendToNXT(handle);
else
    motorB = NXTMotor('B', 'Power', -20, 'TachoLimit', 20)
    motorB.SendToNXT(handle);
    pause(2)
    motorC = NXTMotor('C', 'Power', 20, 'TachoLimit', 200)
    motorC.SendToNXT(handle);
    pause(2)
    motorA = NXTMotor('A', 'Power', 20, 'TachoLimit', 330)
    motorA.SendToNXT(handle);
    pause(3)
    motorC = NXTMotor('C', 'Power', -100, 'TachoLimit', 350)
    motorC.SendToNXT(handle);
    pause(2)
    motorC = NXTMotor('C', 'Power', 20, 'TachoLimit', 150)
    motorC.SendToNXT(handle);
    pause(2)
    motorA = NXTMotor('A', 'Power', -20, 'TachoLimit', 330)
    motorA.SendToNXT(handle);
    pause(2)
    motorB = NXTMotor('B', 'Power', 20, 'TachoLimit', 20)
    motorB.SendToNXT(handle);
    pause(2)
end
CloseSensor(port, handle);}

```