

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

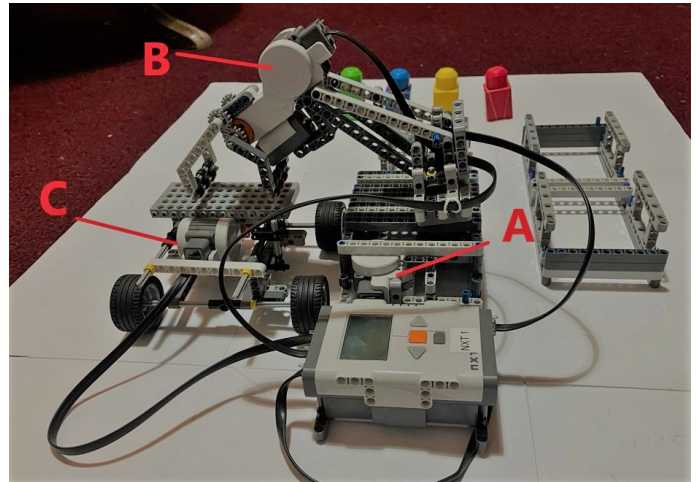


Abbildung 1. Motoren

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, einzeln können und selbstständig einsortieren können. Relativ schnell wurde die Basis des Roboters erstellt, wie in Abbildung 3 zu sehen.

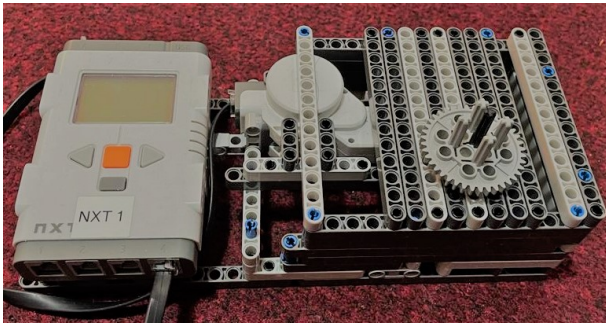


Abbildung 3. Basis des Roboters

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.

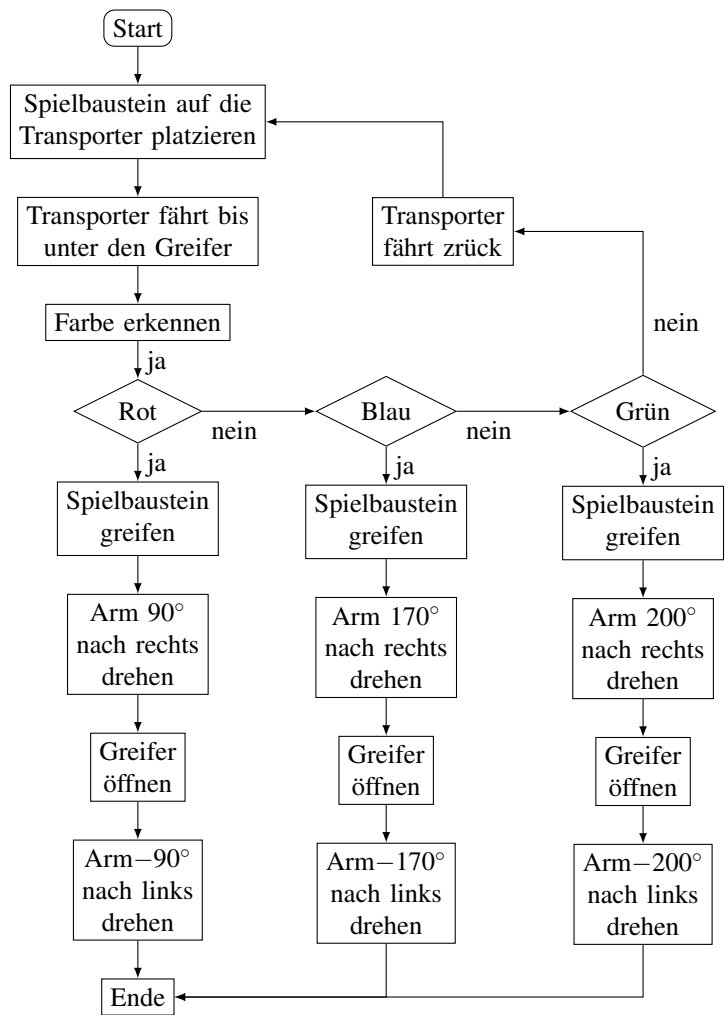


Abbildung 5. Programmablauf

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, dass 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, indem man

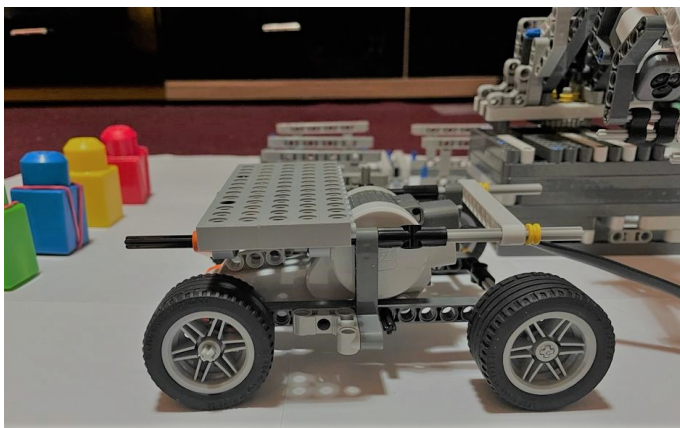


Abbildung 4. Transporter

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('RWTMindstormsNXT'))
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').
```