Tastaturroboter

Vladyslav Shkliarslyi, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

***Zusammenfassung*—Im Rahmen des Projektseminars Elektrotechnik/Informationstechnik (LEGO-Mindstorms) 2024 an der Otto-von-Guericke-Universität Magdeburg wurde den Studierenden zunächst das Ziel, einen Roboter zu erstellen, erläutert, dann die Grundlagen der Programmierung mit MATLAB vermittelt und schließlich die LEGO-NXT-Sets zur Verfügung gestellt.**

***Schlagwörter*—LEGO Mindsorm, MATLAB, NXT-Farbsensor, Tastatur-Roboter, Textverarbeitung**

I. Einleitung

Gleichzeitig mit dem Fortschritt der Technologie wächst auch die menschliche Faulheit, und zwar schneller als je zuvor. Um das Wachstum der Faulheit zu unterstützen, wurde ein Tastaturroboter entwickelt, der in der Lage ist, eine vorgegebene Tastenkombination auf der Tastatur einzugeben. Es kann oft vorkommen, dass man nach einem anstrengenden Tag überhaupt nicht in der Stimmung ist, auf eine Nachricht zu antworten, die aber trotz der Müdigkeit beantwortet werden muss. In diesem Moment der extremen Müdigkeit ist man immer so verzweifelt, dass der Rest des Tages den Bach runtergehen kann. Genau hier kommt der Tastaturroboter ins Spiel, der eine vorgegebene Tastenkombination eingeben würde. Für die Entwicklung des Roboters wurden drei NXT-Motoren und ein NXT-Farbsensor verwendet. Die Motoren ermöglichen die Bewegung entlang der drei Achsen. Damit der Roboter während der Fahrt die Tasten in der richtigen Reihenfolge drücken kann, gibt es verschiedene Farben auf der Tastatur, Jedem Buchstaben ist eine Farbe zugeordnet, die diesen Buchstaben auf der Tastatur repräsentiert.

Abbildung 1. NXT-Farbsensor

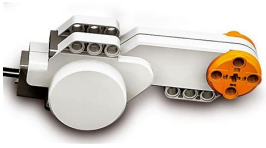

Abbildung 2. NXT-Motor

DOI: 10.24352/UB.OVGU-2024-015

II. Vorbetrachtungen

In diesem Abschnitt werden die Funktionen und Eigenschaften der für die Roboterentwicklung notwendigen Komponenten aus Sicht der Programmierung vorgestellt.

A. NXT-Motor

In der von MATLAB bereitgestellten Bibliothek, die zur Programmierung des Roboters verwendet wurde, stellt der NXT-Motor eine Struktur dar, deren Variablen die Eigenschaften des Notors repräsentieren. Bei der Programmierung des Tastaturroboters waren Eigenschaften wie `Tacholimit` und `Power` am relevantesten. Die Variable `Tacholimit` ist dafür verantwortlich, um wie viele Grad sich der Motor drehen kann. Der Wert dieser Variablen wird in Bogenmaß dargestellt, z. B. entspricht der Wert 360 einer vollen Umdrehung des Motors. Die Variable `Power` ist nicht nur für die Leistung verantwortlich, mit der der Motor arbeitet, sondern auch für die Richtung, in die sich der Motor dreht. Wenn Power ein negatives Vorzeichen hat, dann ist das die gleiche Leistung wie wenn Power ein positives Vorzeichen hat, aber sie dreht sich in die entgegengesetzte Richtung.

B. NXT-Farbsensor

Der NXT-Farbsensor kann insgesamt 13 Farben erkennen. Nachdem der Sensor eine Farbe erkannt hat, gibt er einen Rückgabewert in Form eines Arrays zurück. Dieses Array kann wiederum mit der von MATLAB bereitgestellten Funktion `strcmp()` mit einem String verglichen werden. Die Funktion gibt als Ergebnis des Vergleichs einen booleschen Wert zurück, wodurch sie für unendliche Schleifen geeignet ist, da man den Rückgabewert als Operation für den Ausgang der Schleife verwenden kann.

III. Hauptteil

A. Aufbau

Der Roboter bewegt sich auf vier Rädern, die von einem Motor angetrieben werden, der sich unter der Haube in Abbildung 3 befindet. Der Roboter kann nur in zwei Richtungen fahren, vorwärts und rückwärts, aber das ist auch genügend. Auf den Rädern befindet sich ein Gestell, auf dem eine NXT-Steuerung und eine weitere Konstruktion angebracht sind. Diese weitere Konstruktion enthält zwei Motoren, von denen einer dazu dient, den Druckteil, der auf der Abbildung 5 zu sehen ist, über die Tastatur zu schieben, und der zweite dazu dient, den Druckteil in die Tastatur hinein zu bewegen.

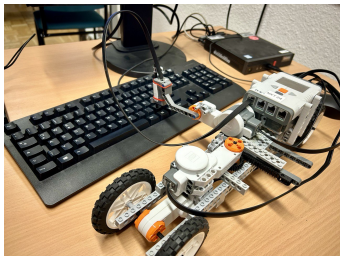

Abbildung 3. Tastaturroboter

Abbildung 4. Sicht von oben

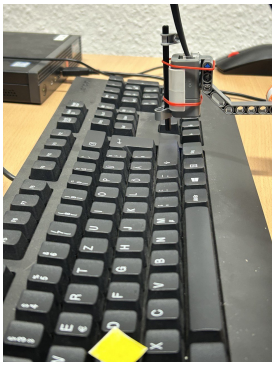

Abbildung 5. der drückende Teil

IV. Funktionsweise

In diesem Teil werden die drei genutzten Motoren des Verständnis halber solcherweise unterteilt:

Motor 1 - bewegt den Roboter entlang der Tastatur
Motor 2 - schiebt den drückenden Teil über die Tastatur
Motor 3 - drückt den drückenden Teil in die Tastatur hinein

A. Fahrt

Vor dem Start schiebt der Roboter den vorderen Teil, an dem der Farbsensor befestigt ist, durch den Antrieb von Motor 2 über die Tastatur um eine vordefinierte Länge aus. Mit dem ausgefahrenen Farbsensor fährt der Roboter entlang der Tastatur. Der Farbsensor analysiert in jedem Moment die Farbe, die sich vor ihm befindet. Wenn die Farbe erkannt wird, hält der Roboter an. Wenn die richtige Farbe gedrückt wurde, fährt der Roboter zur nächsten Farbe. Wenn nicht, wird die ursprüngliche Länge, um die der Farbsensor herausgezogen wird, etwas verkürzt und beim dritten Versuch vergrößert. Es ist zu beachten, dass die Tastenkombination vor der Fahrt festgelegt werden muss.

Listing 1. Definition der Struktur

```
typedef struct {
    char y;// Abstand zur Y-Achse
    int nummer;// Reihenfolgenummer
}Buchstabe;
typedef struct {
    Buchstabe p;
    Buchstabe o;
    Buchstabe w;
    Buchstabe e;
    Buchstabe r;
    Buchstabe o;
    Buchstabe f;
    Buchstabe f;
}Word;
```

V. Software

Wie bereits erwähnt, muss die Tastenkombination im Voraus definiert werden. Dazu wird eine Struktur verwendet, die mehrere Strukturen enthält. Da MATLAB eine seltsame Art hat, Strukturen zu definieren, wird in Listing 1 ein Beispiel mit der Definition derselben Struktur, jedoch in der Sprache C, gezeigt. Für den Code wurde ein rekursiver Ansatz gewählt. In der Main-Funktion werden zunächst drei Motoren und ein Farbsensor sowie die Struktur, die das Wort repräsentiert, initialisiert. Danach wird die rekursive Funktion `typing` aufgerufen, die in Listing 2 zu sehen ist. Die Funktion nimmt 6 Werte als Parameter an, nämlich 3 Motoren, eine Struktur, die aktuelle Farbe und den Rückgabewert, den eine andere Funktion zurückgibt. Die aktuelle Farbe ist eine Zahl. Durch diese Zahl versteht der Roboter, welche Farbe er suchen soll. Die Anzahl der Farben ist vorher festgelegt, so dass sie als Steuerung für den Ausgang der Rekursion dient. Der letzte Parameter `Z` ist der Rückgabewert, der angibt, ob die gewünschte Taste gedrückt wurde. Wenn dieser Wert nicht gleich 1 ist, bedeutet dies, dass die gewünschte Taste nicht gedrückt wurde, und der Roboter muss das Tacholimit des zweiten Motors entweder vergrößern oder verkleinern. Die Funktion `some`, die auch in Abbildung 8 zu sehen ist, übernimmt die Vorbereitung der Motoren und die Bestimmung der Farbe, die der Roboter suchen soll. Am Ende der Vorbereitung übergibt die Funktion `some` die Kontrolle an eine andere Funktion, in der eine Endlosschleife abläuft, während der der Farbsensor arbeitet. Wenn der Farbsensor erfolgreich die gewünschte Farbe findet, wird die Schleife beendet, der dritte Motor angetrieben und die Taste gedrückt. Nachdem die Taste gedrückt wurde, dreht sich der dritte Motor bis zum Ausgangszustand. Danach kehrt die Kontrolle an die Stelle zurück, an der die letzte Funktion in der Funktion

Listing 2. Funktion typing

```
function [] = typing(Letters,
             current_color,
             motor1,
             motor2,
             motor3,z)
    if current_color > x
        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
end
```

`some` aufgerufen wurde. Dort kehrt alles in den ursprünglichen Zustand zurück und die Funktion `some` übergibt bei Erfolg 1 an die Funktion `typing`. Dann wird in der gleichen Funktion `typing` die gleiche `typing` aufgerufen, die jetzt aber eine andere Farbe als die gesuchte Farbe hat. So wird die Funktion `typing` rekursiv aufgerufen, bis die aktuelle Farbe größer wird als die Steuerung für den Ausgang der Funktion.

VI. Ergebnisdiskussion

Das Ergebnis ist ein funktionsfähiger Tastaturroboter, der in der Lage ist, eine vorgegebene Kombination von Buchstaben mit hoher Genauigkeit einzugeben. Es gibt natürlich noch viele weitere Möglichkeiten, die Konstruktion und Funktionalität des Roboters zu verbessern, aber diese wurden aufgrund der begrenzten Zeit nicht berücksichtigt.

VII. Zusammenfassung und Fazit

Das Ziel des Projekts war nicht nur, einen Roboter mit einer bestimmten Funktionalität zu entwickeln, sondern auch praktische Erfahrungen zu sammeln, die Arbeitsumgebung auszuprobieren, die Herangehensweise an nicht triviale Probleme zu entwickeln, in einem Team zu arbeiten, zu lernen, wichtige Entscheidungen selbst zu treffen und viele andere praktische Kenntnisse zu erwerben.

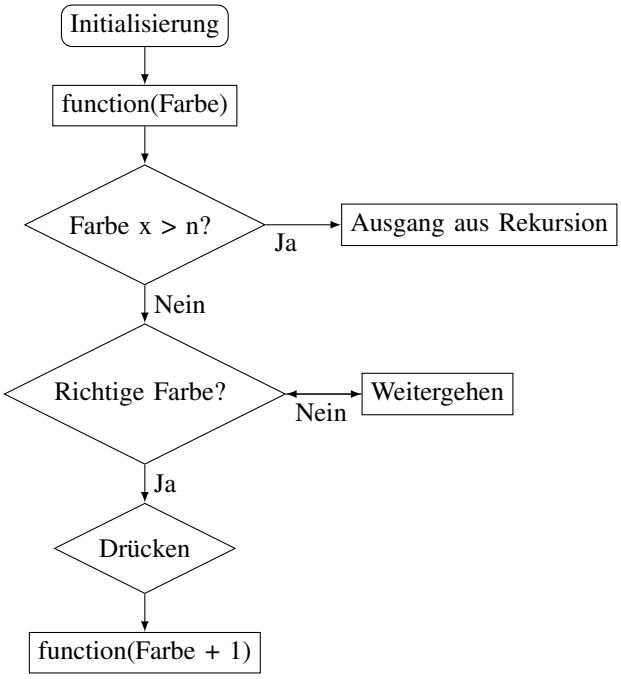

Abbildung 6. Schema des Code-Algorithmus

Literaturverzeichnis

[1]*MATLAB: Dokumentation*.
https://de.mathworks.com/help/matlab/ref/function.html
Version:2023

[2]*Brickwiki: NXT Components*.
https://brickwiki.org/wiki/NXT
Version:2013

[3] *MINDSTORMSNXT: User manual Lego*.
https://www.manualslib.com/manual/726722/Lego-Mindstorms-Nxt.html#manual