Robotik trifft Logik: Sudoku Solver

Affaan Sameer Shaikh, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

***Zusammenfassung*—Das Ziel des Projekts bestand in der Konstruktion eines Roboters, der in der Lage ist, ein Sudoku-Rätsel auf einem Blatt Papier zu identifizieren und zu lösen. Das zu lösende Sudoku-Rätsel wird zunächst gescannt, um eine Identifikation des gegebenen Sudoku-Rätsels zu ermöglichen. Im Anschluss werden die Zahlen an MATLAB gesendet, wo das Sudoku mit Hilfe von Kodierung gelöst wird. Schließlich wird die Lösung, die von MATLAB geschickt wurde, vom Roboter geschrieben. Das Projekt wurde in drei Phasen unterteilt: Scannen des Sudoku, Lösen in MATLAB und Schreiben der Lösung. Der Lichtsensor erwies sich als zu schwach, um das Sudoku präzise zu scannen. Das Projekt scheiterte bereits in Phase 1, sodass eine Weiterführung nicht möglich war.**

***Schlagwörter*—Schlagwörter—LEGO Mindstorms, MATLAB, Sudoku, Lichtsensor, Roboter, Projektseminar**

I. Einleitung

DER Sudoku-Solver Roboter beinhaltet eine Vielzahl von Komponenten. Das klassische japanische Rätsel wurde mit großem Eifer gelöst. Allerdings wurden die Schwierigkeiten, die auftreten würden, sowie die zahlreichen Stunden, die für die Programmierung, Fehlerbehebung und ähnliches erforderlich wären, nicht vorhergesehen. Der Roboter verfügt über keinen spezifischen Anwendungsbereich, dennoch stellt das Projekt eine interessante Herausforderung dar. Für das Lösen eines Sudoku-Rätsels wurden etwa zehn Minuten benötigt, während für das Lösen zwanzig Minuten benötigt wurden. Die erste Herausforderung bestand darin, den Roboter so zu bauen, dass sich der Lichtsensor frei um das Sudoku herum bewegen kann. Des Weiteren musste sichergestellt werden, dass sich der Stift komfortabel um das Sudoku herum bewegen lässt. Daher wurde beschlossen, den Stift und das Sudoku in der gleichen Halterung zu befestigen, sodass sie sich gemeinsam bewegen können (Abbildung 1). Der Lichtsensor und der Stift wurden daher zusammen mit Rädern wie bei einem Auto befestigt, sodass sich das System mit einem einzigen Motor vorwärts und rückwärts bewegen lässt. Diese Vorgehensweise wird im weiteren Verlauf detaillierter erläutert.

II. Vorbetrachtungen

Das Projekt bestand in der Konstruktion eines Roboters, der in der Lage ist, Sudokus zu lösen. Daher war es erforderlich, sich mit der Lösungsmethode von Sudokus vertraut zu machen, was mit MATLAB in Angriff genommen wurde.

A. Konzept des Sudokus und Grundlagen

Ein klassisches Sudoku weist ein 9×9-Gitter auf, welches sich leicht in neun 3×3-Gitter unterteilen lässt. Das Hauptziel eines Sudokus besteht darin, das Gitter mit den Zahlen 1 bis 9 so auszufüllen, dass jede Ziffer nur einmal in jeder Zeile und nur einmal in jeder Spalte erscheint. Unter der Voraussetzung, dass der Zeilenindex mit i und der Spaltenindex mit j und $k = 3$ bezeichnet werden, lässt sich folgende Gleichung aufstellen:

$$\forall 1 \leq i \leq k^2, \forall 1 \leq x, j \leq k^2 : S(i,j) = 0 \vee x \neq j \quad (1)$$
$$\rightarrow S(i,j) \neq S(i,x)$$

$$\forall 1 \leq j \leq k^2, \forall 1 \leq x, i \leq k^2 : S(i,j) = 0 \vee x \neq i \quad (2)$$
$$\rightarrow S(i,j) \neq S(x,j)$$

$$\forall 0 \leq i, j \leq k-1, \forall 1 \leq a, b, c, d \leq k : \quad (3)$$
$$S(i \cdot k + a, j \cdot k + b) = 0 \vee (a \neq c \vee b \neq d)$$
$$\rightarrow S(i \cdot k + a, j \cdot k + b) \neq S(i \cdot k + c, j \cdot k + d)$$

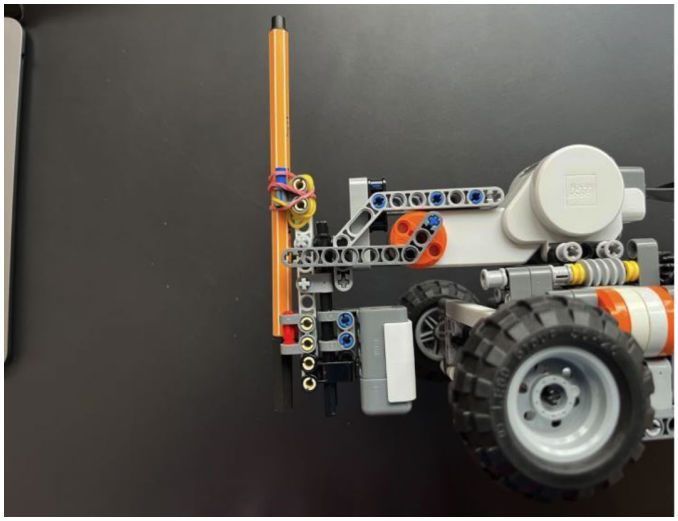

Abbildung 1. Stift und Lichtsensor zusammen

III. Realisierung

A. Aufbau

Der Aufbau besteht aus einem Lichtsensor und einem Motor. Die Bewegung von zwei Rädern ist in der Regel relativ einfach, jedoch gestaltete sich die Befestigung derart, dass sich vier Räder gleichzeitig mit einer stabilen Struktur bewegen konnten. Dies führte zur Idee, einen Roboter mit einem Motor zu bauen. Die Anbringung eines Stiftes und eines Sensors an der Vorderseite ermöglichte die Bewegung, das Scannen und das Schreiben. Des Weiteren wurde die Entscheidung getroffen, einen Motor einzusetzen, der den Stift dreht, sowie

DOI: 10.24352/UB.OVGU-2024-006

einen weiteren Motor, der den Stift hoch und runter bewegt. Dadurch sollte eine einfache Kodierung der beiden Motoren gewährleistet werden. Die zugrunde liegende Idee schien vielversprechend, allerdings konnte sie in der Praxis nicht umgesetzt werden. Der resultierende Roboter (Abbildung 2) wies eine hohe Instabilität und Ineffizienz auf.

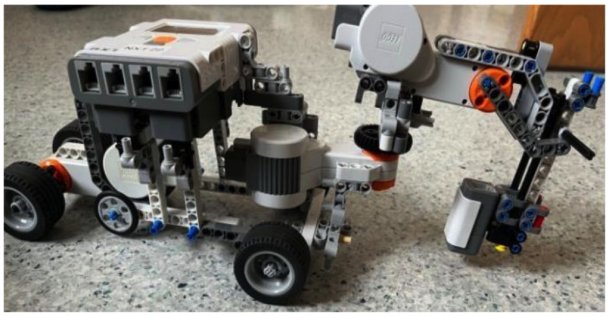

Abbildung 2. Alte Aufbau des Sudoku Solver

In der vorliegenden Version wurde ein größeres Zahnrad verwendet, das über mehr Befestigungen verfügte. Dadurch konnte eines der zuvor genannten Probleme, nämlich die Instabilität, die durch das Gewicht des Lichtsensors verursacht wurde, gelöst werden (Abbildung 3). Mithilfe dieses Zahnrads war es dem Roboter nun möglich, den Stift zu bewegen und auch perfekt zu schreiben.

Abbildung 3. Zahnrad

Um die Stabilität der NXT-Box zu gewährleisten, wurden erstens größere Räder verwendet und zweitens die gesamte Struktur direkt an der NXT-Box befestigt. Auf diese Weise konnte auch das zweite Problem gelöst werden (Abbildung 4). Als Ergebnis wurde ein Roboter erzielt, der sich vorwärts und rückwärts bewegte und dem Stift ermöglichte, sich auf und ab zu bewegen, was das Scannen und Schreiben sehr einfach machte.

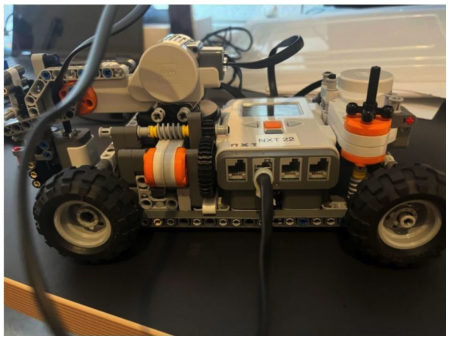

Abbildung 4. Neuer Aufbau des Sudoku Solver

B. *Software*

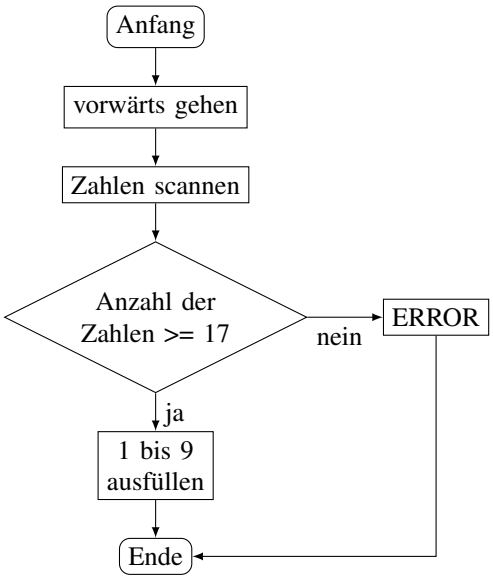

Abbildung 5. Ablaufplan

Die erste Herausforderung bestand also darin, die Zahlen mithilfe eines Lichtsensors zu scannen. Dies stellt eine grobe Erklärung für das dar, was versucht wurde zu erreichen (Abbildung 5).

Abbildung 6. Zahlen Scannen

Die Lösung des Sudokus wird durch einen Algorithmus ermittelt, der nach dem bewährten Backtracking-Prinzip [3] funktioniert. Die Lösung wird durch ein gezieltes Ausprobieren ermittelt, wobei mit der ersten freien Zelle begonnen wird. Es wird nach einer gültigen Ziffer gesucht. Sofern eine gültige Ziffer identifiziert wird, wird der Prozess in der nächsten freien Zelle wiederholt. Andernfalls wird ein Schritt zurückgesetzt und nach einer neuen, gültigen Ziffer gesucht. Die Vorteile dieses Algorithmus liegen in seiner einfachen Rekursivität sowie der Garantie, dass er eine Lösung findet, sofern eine vorhanden ist. Eine weitere Optimierung des Algorithmus kann durch die Berücksichtigung der Anzahl der möglichen Lösungskandidaten in der Abarbeitungsreihenfolge anstelle einer sequenziellen Bearbeitung der freien Zellen erzielt werden. Es wird empfohlen, nach jedem Schritt zu prüfen, welche Zelle die meisten Lösungskandidaten aufweist und diese als nächstes zu bearbeiten. Im Anschluss erfolgt die Bearbeitung der freien Zelle mit der geringsten Anzahl an Lösungskandidaten.

Diese Methode führt in der Regel zu einer erheblichen Reduzierung der Laufzeit. Des Weiteren wird dem Benutzer der aktuelle Stand angezeigt.

C. Probleme

Der Sensor war nicht in der Lage, die Zahlen exakt zu scannen, beispielsweise konnte er den Unterschied zwischen 1 und 7 nicht erkennen, da sie sich sehr ähneln. Zudem wurde nicht immer erkannt, dass eine Zahl überhaupt existierte. Der Code wurde so gestaltet, dass mindestens 17 Zahlen erkannt werden müssen, um mit der Lösung des Sudokus beginnen zu können. Die Existenz dieser 17+ Zahlen wurde vom Sensor nicht einmal erkannt, geschweige denn, dass diese Zahlen identifiziert und dann das Sudoku gelöst wurden. Es wurde in helleren und dunkleren Umgebungen ausprobiert, um zu sehen, ob der Sensor auf diese Weise besser abschnitt (Abbildung 7). Selbst das Handy wurde direkt neben dem Sensor beleuchtet, in der Hoffnung, dass es funktionieren würde, jedoch war leider die Zeit vorbei. Im Rahmen der durchgeführten Untersuchungen wurde ein Code entwickelt, der ein Sudoku lösen kann, sowie ein Roboter, der Zahlen schreiben kann. Aus diesem Grund wurde beschlossen, den Roboter als Abschlussprojekt vorzustellen, der auf Befehl Zahlen schreibt.

Abbildung 7. Versuch mit Licht aus Handy

IV. Ergebnisdiskussion

Nachdem die Vorschläge und Genehmigungen der Professoren eingeholt worden waren, wurden die implementierten Funktionen des Roboters geändert. Der Roboter musste in die Lage versetzt werden, die Zahlen 1 bis 9 ohne Probleme auf ein leeres Blatt Papier zu schreiben, und es wurde auch versucht, "Danke" zu schreiben.

V. Zusammenfassung und Fazit

Der Code wurde hauptsächlich in 4 Teile aufgeteilt: Vorwärts, Rücken, Links, Rechts, basierend auf der Bewegung des Stifts. Zum Beispiel würde zweimal abwärts/aufwärts eine gerade Linie ergeben, die die Nummer eins ergibt. Ähnlich würde rechts, rücken, links, rechts, rücken, links die Zahl drei ergeben.

Anhang

```
COM_SetDefaultNXT(handle);
Downie = NXTMotor('A','Power',100);
Downie.TachoLimit = 250;
Downie.SendToNXT();
Downie.WaitFor();
Uppie = NXTMotor('A','Power',-100);
Uppie.TachoLimit = 250;
Uppie.SendToNXT();
Uppie.WaitFor();
Leftie = NXTMotor('C','Power',-100);
Leftie.TachoLimit = 80;
Leftie.SendToNXT();
Leftie.WaitFor();
Rightie = NXTMotor('C','Power',100);
Rightie.TachoLimit = 80;
Rightie.SendToNXT();
Rightie.WaitFor();%Steuercode
Downp;
Down;
Down;
Right;
Right;
Up;
Up;
Uppie = NXTMotor('A','Power',-100);
Uppie.TachoLimit = 30;
Uppie.SendToNXT();
Uppie.WaitFor();
Left;
Left;
Leftieyo = NXTMotor('C','Power',100);
Leftieyo.TachoLimit = 10;
Leftieyo.SendToNXT();
Leftieyo.WaitFor();
Upp;
Down;
Downp;
Right;
Right;
Upp;%Beispiel von Acht
```

Der folgende Code zeigt, wie die Zahl Acht geschrieben wird, und die Funktionen darin repräsentieren die Bewegungen der Kabine und des Auslegers. Die Notwendigkeit, eine Anweisung zweimal zu wiederholen, resultiert aus Reibung zwischen Stift und Papier, und durch mehrere Versuche wurde festgestellt, wo dies erforderlich ist.

Literaturverzeichnis

[1] Adriano Parracciani:
One-Motor Car Lego NXT, https://www.pinterest.de/pin/537054324290642123/, Version:Februar 2024

[2] Hans Andersson:
Sudoku Solver, https://tiltedtwister.com/sudokudownload.html, Version:August 2009

[3] Eckart Sußenburger:
Lösungs und Generierungsalgorithmen für Sudoku. Trier: Fachhochschule Trier, Fachbereich Informatik, Bachelor Abschlussarbeit 17.04.2007