LEGO Mindstorms Roboter-Cardsdealer

Hlieb Khramov, Studiengang
Otto-von-Guericke-Universität Magdeburg

Ein Dokument, das das Projekt, dessen Grundidee und Funktionalität beschreibt sowie zentrale Entscheidungen während der Entwicklung des Roboters darlegt. Darüber hinaus enthält es eine Übersicht über die wesentlichen Herausforderungen bei der Konstruktion des Roboters und der Programmierung. Ergänzend sind die verwendeten Quellen und Programme aufgeführt, die im Rahmen der Projekterstellung zum Einsatz kamen.

I. Einleitung

DIESES Projekt ist ein Roboter zum Kartenspielen, der mit MATLAB-Programmen und dem LEGO Mindstorm-Konstruktor erstellt wurde. Die Idee dieses Roboters besteht darin, den Prozess des Kartenausteilens zu beschleunigen und außerdem Fehler beim Kartenausteilen zu minimieren, wenn beispielsweise einem der Spieler eine Karte mehr ausgeteilt wird. Das Ziel war es außerdem, einen Roboter mit veränderlichen Einstellungen zur Kartenverteilung zu entwickeln, der bei der Auswahl der Spielerzahl, der Kartenzahl und der Spielart selbst flexibler ist. Für dieses Projekt wurden MATLAB-Software, LEGO TECHNIK- und LEGO Mindstorm-Teile, während der Arbeit am Projekt gewonnenes Wissen sowie die Idee des Roboters selbst benötigt.

II. Vorbetrachtungen

Die Erstellung des Projekts wurde in drei Phasen unterteilt: Die erste Phase ist die Idee, die zweite Phase ist die Konstruktion, die dritte Phase ist die Programmierung.

A. Idee

Ursprünglich bestand die Idee darin, einen Roboter zur Steuerung eines Telefons zu bauen, der durch die Feeds sozialer Netzwerke scrollen und Fotos liken oder disliken sollte. Diese Idee wurde jedoch verworfen, da der Roboter nur über wenige Funktionen verfügte und Fotos nur anhand der Farbe erkennen konnte. Danach wollten wir zusammen einer anderen Gruppe und mache ein Projekt mit zwei schnell bewegende Roboter, wie bei einem Rennen. Auch diese Idee war schlecht, da die Maximalgeschwindigkeit aller Motoren gleich ist und daher auch die Geschwindigkeit der Roboter gleich sein wird. Am Ende entschied man sich, den eigenen Roboter zu bauen, der die Karten an die Spieler verteilt. Das Gute an diesem Roboter ist, dass er über zahlreiche Funktionen verfügt, wie etwa eine 270-Grad-Drehung, um allen Spielern Karten zu geben, eine Schaltfläche, um einem Spieler eine separate Karte zu geben, und auch die Möglichkeit, die Anzahl der Karten und Spieler zu ändern.

B. Konstruktion

Der Bau des Roboters brachte viele Herausforderungen mit sich, beispielsweise die Entwicklung eines Mechanismus zur Kartenausgabe. Der Mechanismus muss guten Kontakt mit der Karte haben, um nur eine Karte pro Motorumdrehung zu produzieren. Außerdem gab es ein Problem mit dem Gewicht des Turms: Er war zu schwer und konnte sich nicht um 270 Grad drehen. Auch die Kabel vom Motor zum Hauptbedienfeld des Roboters störten und mussten außen verlegt werden, da im Körper des Roboters nur wenig Platz für sie vorhanden war.

C. Programmierung

Beim Programmieren des Roboters stieß man auf nur ein Problem – das Fehlen einiger Plugins für den korrekten Betrieb des Roboters und seiner Sensoren mit dem MATLAB-Programm.

III. Konstruktion und Programmierung

Im Hauptteil werden das Konstruktion und die Realisierung erläutert.

Abbildung 1 zeigt den im Rahmen dieses Projekts hergestellten Roboter.

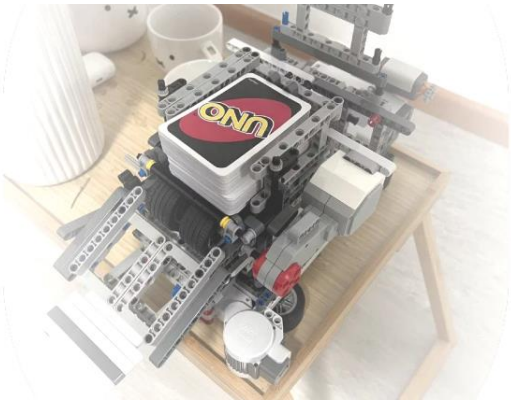

Abbildung 1: Roboter

A. Konstruktion

Die Abbildung 2, 3 und 4, zeigen den Mechanismus zur Ausgabe der Roboterkarten von verschiedenen Seiten.

DOI: 10.24352/UB.OVGU-2025-024

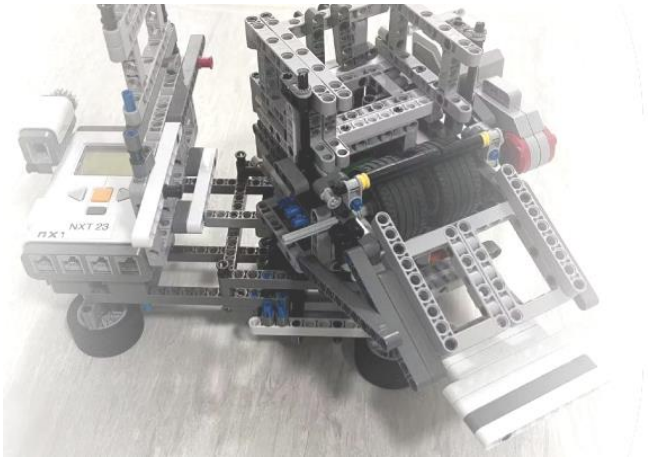

Abbildung 2: Blick auf den Mechanismus aus einem Winkel

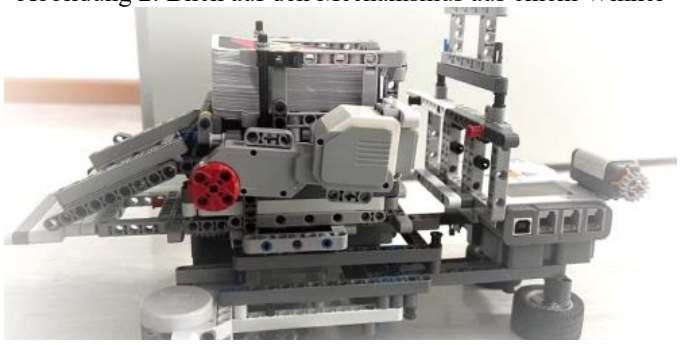

Abbildung 3: Blick auf den Mechanismus von der Seite

Abbildung 4: Blick auf den Mechanismus von oben

Um diesen Mechanismus zu implementieren, waren zwei Motoren erforderlich: einer zum Drehen des Turms, der zweite für den Kartenausgabemechanismus. Im Turm selbst haben wir einen Mechanismus zur Kartenausgabe gebaut, wofür wir ein Paar Lego-Räder verwendet haben. Sie waren ideal, um festen Kontakt mit der Karte herzustellen. Zusätzlich wurde für die Karten eine Hülle angefertigt, damit diese nicht herausfallen oder sich bewegen können, da hierdurch auch die Kartenausgabe beeinträchtigt wird.

B. Programmierung

Abbildung 5 zeigt ein vereinfachtes Flussdiagramm des Robotercodes, das den Start, die Prüfung und die Ausführung der Hauptfunktionen des Roboters beschreibt.

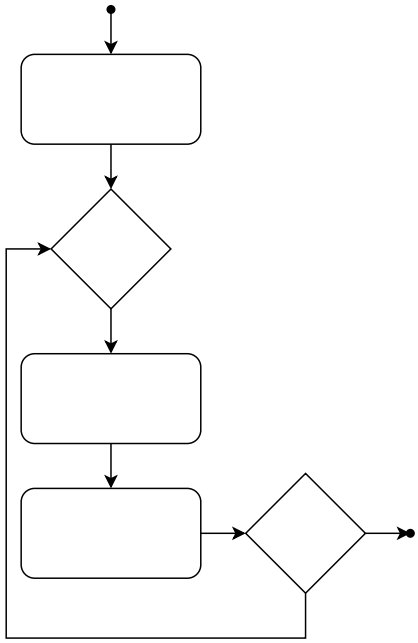

Abbildung 5: Flussdiagramm

Ein Beispiel unseres Hauptcodes für den Kartenausgabemechanismus, erstellt mit der For-Schleife.

```
for j = 1:a
  for k = 1
    motorA.Power = 40;
    if c == 1
      motorA.TachoLimit = 450;
    elseif c == 2
      motorA.TachoLimit = 350 * c;
    elseif c == 3
      motorA.TachoLimit = 300 * c;
    else
      motorA.TachoLimit = 265 * c;
    end

    motorA.SendToNXT();
    motorA.WaitFor(c);
    motorA.Stop('off');

    motorA.Power = -50;
    cotorA.TachoLimit = 500;
    motorA.SendToNXT();
    motorA.WaitFor(1);
    motorA.Stop('off');
  end

  motorB.Power = -15;
  motorB.TachoLimit = round(angle_per_player);
  motorB.SendToNXT();
  motorB.WaitFor(1);
  motorB.Stop('off');
end
```

Außerdem ist hier der Code für den Sensor zur Kartenausgabe nach Zyklusende auf dem Knopf vorhanden.

```
while true
 sensorState = GetSwitch(SENSOR_4);
if sensorState == 1
   press_time = tic; % Запоминаем момент нажатия

   % Ждём, пока кнопка удерживается
   while GetSwitch(SENSOR_4) == 1
      elapsed_time = toc(press_time);
      if elapsed_time > 1
         break;
      end
   end

   if elapsed_time > 0.3

      disp('Double!');
      motorA.Power = 90;
      motorA.TachoLimit = 310 * 2;
      motorA.SendToNXT();
      motorA.WaitFor(1.5);
      motorA.Stop('off');

      motorA.Power = -100;
      motorA.TachoLimit = 200;
      motorA.SendToNXT();
      motorA.WaitFor(1);
      motorA.Stop('off');
   else
      % Если кнопка удерживалась менее 1 секунды – выдаём 1 карту
      disp('One card!');
      motorA.Power = 100;
      motorA.TachoLimit = 360;
      motorA.SendToNXT();
      motorA.WaitFor(1);
      motorA.Stop('off');

      motorA.Power = -100;
      motorA.TachoLimit = 200;
      motorA.SendToNXT();
      motorA.WaitFor(1);
      motorA.Stop('off');
   end

   % Ждём, пока пользователь отпустит кнопку
   while GetSwitch(SENSOR_4) == 1
   end
end
```

IV. Ergebnisdiskussion

Das Endergebnis dieses Projekts ist die einwandfreie Bedienung aller oben beschriebenen Funktionen des Roboters sowie die hochwertige Montage von Mechanismen für die weitere Nutzung des Roboters. Während der Arbeit an dem Projekt sind uns viele Fehler und Probleme begegnet, wie beispielsweise der schwere Roboterturm, der Mechanismus zur Kartenausgabe, der Platzmangel für das USB-Kabel usw. Auch bei der Installation von Plugins auf dem Computer und beim Arbeiten mit Batterien gab es Probleme und man musste sparsam mit ihnen umgehen, da der Roboter viel Energie verbrauchte und die Batterien schnell leer waren.

V. Zusammenfassung und Fazi

Durch die Konstruktion und Programmierung des Roboters konnten Kenntnisse im Umgang mit LEGO-Komponenten, im Entwurf komplexer mechanischer Strukturen sowie in der Anwendung von MATLAB zur Programmierung vertieft werden. Darüber hinaus wurde der Entwicklungsprozess durch fortlaufendes Testen und Optimieren der Mechanismen unterstützt.

Zur weiteren Verbesserung des Roboters bietet es sich an, eine Mobilitätsfunktion durch den Einbau eines Fahrwerks zu integrieren, um die Einsatzmöglichkeiten zu erweitern. So könnte der Roboter beispielsweise auf einem Tisch navigieren und dabei eigenständig Spielkarten austeilen, etwa bei einem Pokerspiel. Auch eine Kartenmischfunktion ließe sich umsetzen, was jedoch den Einsatz von zwei zusätzlichen Motoren erfordern würde.

Literaturverzeichnis

[1] Meets MINDSTORMS: How to Control LEGO NXT Robots Using MATLAB for Educational Purposes
[2] Required Equipment - MATLAB LEGO Mindstorms NXT 2.0 Instruction Manual
[3] Projektseminar Elektrotechnik/Informationstechnik (LEGO Mindstorms)
[4] Video mit dem Titel LEGO Card Shuffler and Dealer