

Umblättermaschine

Andrii Vahanov, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Bericht geht es um die Entwicklung der Umblättermaschine unter Verwendung von LEGO Mindstorms im Rahmen des LEGO-Projektseminars 2026 an der Otto-von-Guericke-Universität Magdeburg. Der Roboter reagiert auf die Bewegung der Hand neben dem Ultraschallsensor, blättert das Buch mithilfe von zwei Motoren um und macht ein Foto. Das Programm wurde in MATLAB geschrieben. Im Beitrag werden der Aufbau und die Möglichkeiten der Maschine erklärt. Außerdem werden die Probleme, die während des Entwicklungsprozesses aufgetreten sind, sowie deren Lösungen erläutert.

Schlagwörter—Buch, LEGO Mindstorms, Maschine, MATLAB, Umblätterrung

I. EINLEITUNG

Die moderne Welt ist sehr digitalisiert, und es ist oft wichtig, Dokumente, Bücher oder Texte aus Heften zu scannen. Mit Papieren, deren Seiten getrennt sein können, ist das kein Problem, da der Scanner gut mit ihnen funktioniert. Wenn man jedoch mit Büchern arbeitet, muss man jede Seite selbst umblättern und fotografieren. Die Umblättermaschine erleichtert diesen Prozess. Der Roboter trennt eine Seite, blättert sie um und macht dann ein Foto. Deshalb läuft die Buchscannung viel schneller.

II. BESTANDTEILE DES ROBOTERS

In diesem Abschnitt werden wichtige Informationen bezüglich der genutzten Bauteile gegeben, um die Funktionsweise des Roboters verständlicher zu machen.

A. Entwicklungsplattform

In der Arbeit wird LEGO Mindstorms EV3 als Kontrolleinheit verwendet (siehe Abbildung 1). Der EV3 wird mit der MATLAB-Sprache programmiert, um das Programm zu vereinfachen und keine weiteren Apps zu benötigen.

B. Motoren

Für diesen Roboter benötigt man zwei große Motoren (siehe Abbildung 2). Der erste Motor mit dem Rad trennt die Seiten auseinander, und dann blättert der zweite Motor sie um. Dafür sollen beide Motoren miteinander synchronisiert sein.

C. Ultraschallsensor

Der Ultraschallsensor (siehe Abbildung 3) ist der Hauptbestandteil der Konstruktion. Der Sensor reagiert auf die Bewegung und lässt die Motoren drehen. Damit kann man kontrollieren, wie viele Seiten umgeblättert werden und wie groß die Pause dazwischen ist.



Abbildung 1. LEGO Mindstorms EV3



Abbildung 2. LEGO-Motor

III. AUFBAU UND PROGRAMMIERUNG

In diesem Abschnitt werden das Konzept und die Realisierung erläutert.

A. Aufbau

Das ganze Projekt besteht aus der EV3-Box, zwei Motoren, einem Ultraschallsensor, einer Taste und LEGO-Bausteinen.



Abbildung 3. LEGO-Ultraschallsensor

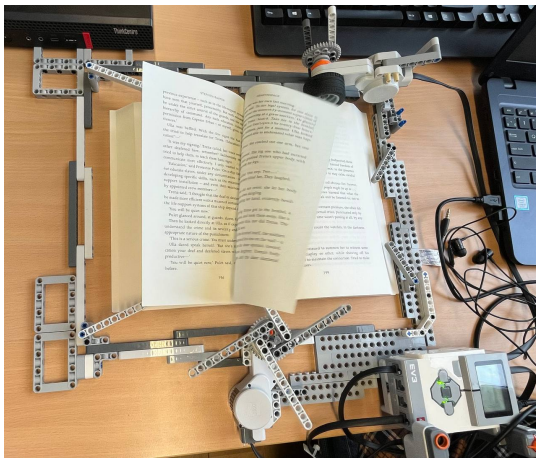


Abbildung 4. Umblättermaschine

Aus LEGO-Steinen wurden der Rahmen, der die Buchbewegung begrenzt, erstellt. An dem Rahmen sind beide Motoren und die EV3-Box befestigt (siehe Abbildung 4). Die EV3-Box ist auch mit dem Sensor und der Taste verbunden. Die Hauptidee war, dass der Ultraschallsensor reagiert, wenn der Abstand zum oberen Objekt kleiner als 20 cm ist. Falls diese Bedingung erfüllt ist, beginnt Motor A seine Bewegung. Er trennt eine Seite, damit Motor B sie umblättern kann. Die Motoren arbeiten nicht gleichzeitig, sondern nacheinander. Auch ist es unmöglich, bis zum Ende der ersten Umblätterung mit dem zweiten anzufangen.

Wichtig ist, dass der Roboter selbst keine Fotos machen kann. Dafür braucht man einen Buchscanner oder eine Kamera.

B. MATLAB-Programm

Das MATLAB-Programm ist einfach, trotzdem funktioniert es gut genug. Zuerst sollte man den EV3-Block, beide Motoren, den Sensor und die End-Taste im MATLAB-Code definieren. Dann werden die Parameter der Motoren sowie die Leistung und der Drehwinkel festgestellt. Wichtig ist, dass die Motoren nach dem Ende der Arbeit sofort die Bewegung unterbrechen sollen. Dafür kann man die Funktion `brake.mode = 'Brake'` nutzen.

Dann wurde der Hauptteil des Programms geschrieben. Den Programmablaufplan kann man in Abbildung 5 sehen.

Und der gesamte Code ist im Anhang in Listing 1 verfügbar.

C. Kamera

Beim Aufbau kam der CZUR Shine Scanner [1] (siehe Abbildung 7) zusammen mit der zugehörigen Software [2] zum Einsatz. Der Grund dafür ist die gute Bildqualität und die Möglichkeit, Fotos nach dem Umblättern automatisch zu machen.

IV. ERGEBNISDISKUSSION

Bevor das Projekt seine Ziele erreichte, hatte unser Team viele Probleme getroffen. Es ist möglich, alle Herausforderungen in zwei große Teile zu trennen: Das sind Konstruktionsprobleme und Programmiermissverständnisse.

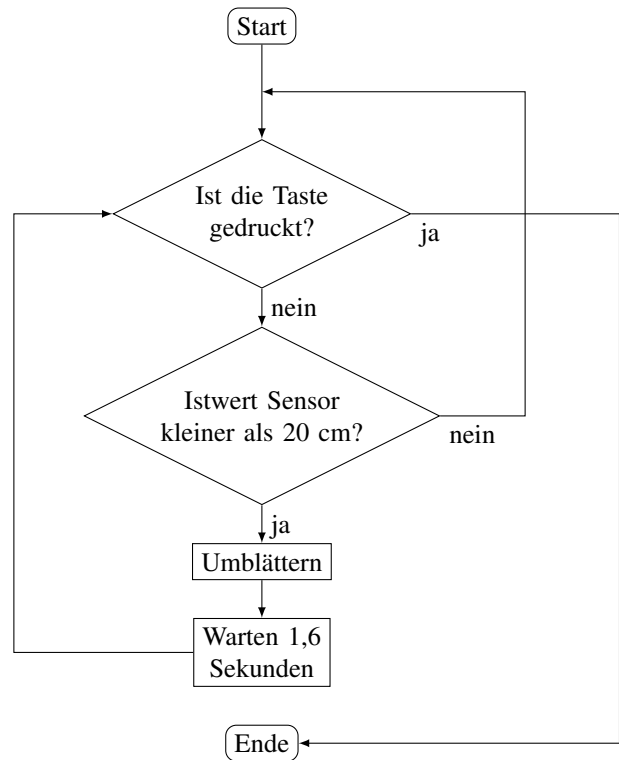


Abbildung 5. Programmablaufplan

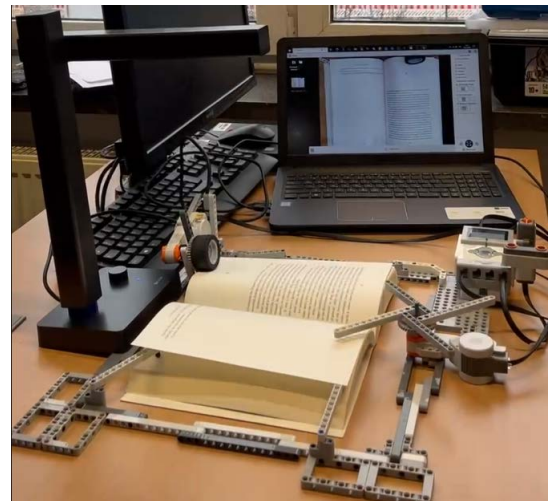


Abbildung 6. Fertiger Roboter

A. Konstruktionsherausforderungen

Es gab zwei große Dinge, wegen derer die Maschine nicht funktioniert hat. Die erste war die instabile Konstruktion. Die Rahmen konnten die Buchbewegung nicht begrenzen. Das wurde durch eine totale Änderung der Maschine gelöst.

Das zweite Problem ist schwerer zu lösen. Es liegt an verschiedenen Buchformaten und -größen. Die Lösung ist eine einfache Rahmenkonstruktion, die man für jedes Buchformat schnell verändern kann.



Abbildung 7. CZUR-Kamera

B. Probleme mit dem Programm

Diese Herausforderungen waren viel einfacher, aber gefährlicher für Bücher, weil die Motoren sie sehr leicht beschädigen könnten. Die Probleme waren die Synchronisation der Motoren sowie die richtige Leistung und der richtige Winkel. Um sichere Parameter zu finden, sollte das Team viele verschiedene Zahlen testen.

V. ZUSAMMENFASSUNG UND FAZIT

Insgesamt hat das Projekt seine erklärten Ziele erreicht (siehe Abbildung 6). Selbstverständlich kann man den Roboter noch verbessern und weiterentwickeln. Beispielsweise wird die Maschine mit allen Buchgrößen effizient arbeiten, oder werden die Rahmen das Buch automatisch begrenzen.

VI. ANHANG

A. Programmcode der Umblättermaschine

Listing 1. Programmcode

```

b = EV3();
b.connect('usb');
ma = b.motorA;
ma.setProperties('Power', 20, ...
... 'LimitValue', 180);
ma.brakeMode = 'Brake';
mb = b.motorB;
mb.brakeMode = 'Brake';
s1 = b.sensor1;
s4 = b.sensor4;
while s1.value != 1
    if s4.value < 20
        ma.start();
        waitFor(ma);
        ma.stop();
        mb.setProperties('Power', ...
... -20, 'LimitValue', 180);
        mb.start();
        waitFor(mb);
        mb.stop();
        pause(1.6);
    end
end

```

LITERATURVERZEICHNIS

- [1] CZUR TECH CO: *CZUR Shine Scanner*. <https://www.czur.com/product/shineultra>. Version: März 2026
- [2] CZUR TECH CO: *CZUR Software*. <https://www.czur.com/support>. Version: März 2026