

Roboter zum Eierfärben

Wird keinen Osterhase ersetzen, aber das Eierfärben präziser und kreativer machen!

Nebrytov Maksym, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des innovativen LEGO-Mindstorms-Projekts im Studiengang Elektrotechnik und Informationstechnik im Jahr 2025 wurde ein autonomer Roboter entwickelt, der in der Lage ist, Ostereier automatisch und individuell zu bemalen. Die Idee zu diesem Projekt entstand aus der Überlegung, ein kreatives, gleichzeitig technisch herausforderndes und praxisnahes Vorhaben umzusetzen, das sowohl mechanische Präzision als auch intelligente Steuerung vereint. Der Gedanke, traditionelle Handarbeit durch Automatisierung kreativ zu interpretieren, führte schließlich zur Entwicklung eines Ostereier-Bemalroboters.

Neben der Konzeption des Roboters wurde ein präziser Programmcode implementiert, um eine effiziente und flexible Steuerung zu gewährleisten.

Die Einführung in das Seminar begann mit der Erläuterung dieses Vorhabens, gefolgt von einer fundierten Einführung in die Grundlagen der Programmierung mit MATLAB. Dies bildete das unerlässliche Rüstzeug, um die Studierenden mit den essenziellen Kenntnissen und Fähigkeiten auszustatten, die für die Umsetzung ihres Roboterprojekts unabdingbar waren. Der Lernprozessor streckte sich über verschiedene Phasen, die von der Theorie bis zur praktischen Umsetzung reichten.

Schlagwörter—LEGO-Mindstorms, MATLAB, Präsentation, Präzision, Roboter.

I. EINLEITUNG

Bei traditionellen Ostereierfarben sind Geduld und Präzision benötigt, um schöne Muster zu erzeugen. Die Qualität des Ergebnisses hängt von der ruhigen Hand und der gleichmäßigen Farbverteilung ab. Gleichzeitig kann es schwierig sein, gleichmäßige Muster zu erzeugen, insbesondere wenn mehrere Farben verwendet werden.

Mit dem Eierfarbe-Roboter lässt sich dieser Prozess automatisieren, wodurch eine gleichmäßige und kreative Gestaltung der Eier ermöglicht wird. Das System reduziert Farbfehler und erleichtert das Experimentieren mit verschiedenen Mustern.

Der Roboter ist mit einer stabilen Basis ausgestattet, die das Ei sicher hält und rotiert. Gleichzeitig bewegt sich der Pinselhalter entlang einer programmierten Bahn, um präzise Linien oder Muster aufzutragen. Ein integrierter Sensor kann dabei helfen, Farbaufträge zu kontrollieren und gegebenenfalls anzupassen. Dadurch wird ein gleichmäßiges und ästhetisches Ergebnis erzielt.

II. VORBETRACHTUNGEN

Einsatz von LEGO Mindstorms für die Robotik-Umsetzung. Nutzung von Motoren zur präzisen Steuerung der Farbauftragung. Integration verschiedener Muster durch gezielte Bewegungsabläufe. Herausforderungen in der exakten Fixierung der Eier.

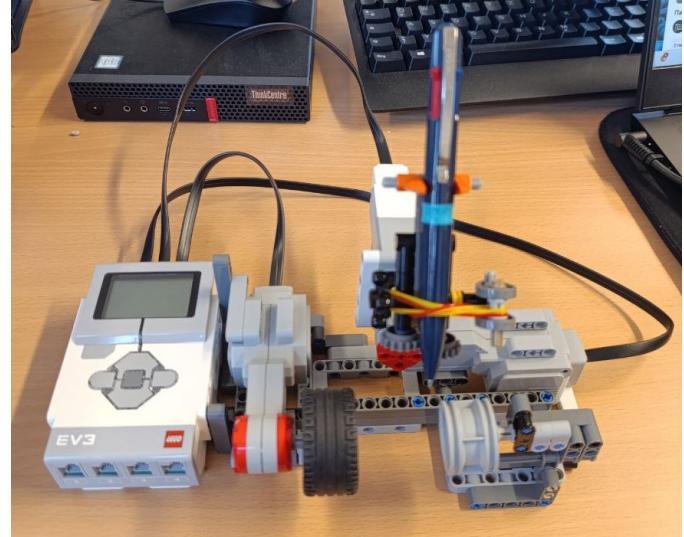


Abbildung 1. Roboter zum Eierfärben

A. Roboteraufgaben

Eine klare Definition der Aufgaben, die der Roboter ausführen soll, ist entscheidend, da Unklarheiten oder Unsicherheiten dazu führen können, dass eine vielversprechende Idee in endlosen Optimierungsversuchen stecken bleibt. Deshalb wurde ein solides konzeptionelles Fundament entwickelt, auf dem anschließend die Programmierung und der Aufbau mit LEGO-Elementen basieren.

B. Funktionsweise der Lego Komponenten

Bevor mit der Konstruktion des Eierfarbe-Roboters begonnen wurde, stand zunächst die Untersuchung der Funktionsweise der EV3-Motoren (siehe Abbildung 1) im Fokus.

Besonderes Augenmerk galt zudem der Handhabung des Pinsels und der möglichen Integration von Sensoren zur Farbkontrolle, da diese eine entscheidende Rolle für die Qualität der Bemalung spielen.

Durch diese Voruntersuchungen konnten die Möglichkeiten des LEGO-Systems für die weitere Umsetzung besser eingeschätzt werden. Gleichzeitig ermöglichte dieses Wissen ein tieferes Verständnis der Softwaresteuerung, was sich später als hilfreich erwies, um einen stabilen und gut strukturierten Quellcode zu entwickeln.

C. Technische Umsetzung und Softwarekomponenten

Trotz zahlreicher Designkonzepte wurde das Projekt anfangs durch die Grenzen von MATLAB und die strukturellen



Abbildung 2. EV3-Controller



Abbildung 4. Kleiner Motor



Abbildung 3. Großer Motor

Herausforderungen beim Zusammenbau der LEGO-Elemente eingeschränkt. Diese Faktoren begrenzen auch die Funktionalität des Roboters. Zusätzlich stellte die Implementierung in MATLAB das Team vor unerwartete Schwierigkeiten, da bestimmte Funktionen Anpassungen im Quellcode erforderlich machten, was zu mehreren Überarbeitungen führte.

III. KONSTRUKTION

A. Design

Der Eierfarbe-Roboter basiert auf dem EV3-Controller (siehe Abbildung 2), der als zentrale Steuereinheit fungiert und die verschiedenen Bewegungsabläufe koordiniert. Damit die Farbmuster präzise und gleichmäßig auf das Ei aufgetragen werden, kommen drei Motoren zum Einsatz, die spezifische Aufgaben übernehmen:

Motor A: Verantwortlich für die Drehung des Eis [2]. Durch eine gleichmäßige Rotation wird sichergestellt, dass die Farbe lückenlos aufgetragen wird und keine ungleichmäßigen Farbverläufe entstehen. Die Geschwindigkeit und Richtung der Drehung können je nach gewünschtem Muster variiieren (siehe Abbildung 3).

Motor C: Regelt die horizontale Bewegung des Pinsels [2]. Durch diese kontrollierte Seitwärtsbewegung können verschiedene Muster auf das Ei Oberfläche aufgetragen werden. (siehe Abbildung 3).

Motor B: Steuert die vertikale Bewegung des Pinsels. Dies ermöglicht eine kontrollierte Farbauftragung in unterschiedlichen Höhen des Eis. Abhängig vom gewählten Design kann der Pinsel sanft oder mit stärkerem Druck aufgesetzt werden, um verschiedene Intensitäten der Farbauftragung zu erzielen (siehe Abbildung 4).

B. Herausforderungen und Lösungen

Zu Beginn wurde der Roboter mit einer rotierenden Halterung für das Ei entworfen, die eine gleichmäßige Rotation ermöglichen und eine präzise Farbauftragung gewährleisten sollte (siehe Abbildung 1). Allerdings traten während der ersten Tests mehrere Herausforderungen auf.

Ein zentrales Problem war die Stabilität des Eis. In der ursprünglichen Konstruktion verrutschte das Ei während der Drehbewegung, was zu ungleichmäßigen Farbmustern führte. Durch eine verbesserte Fixierung konnte dieses Problem behoben werden.

Ein weiteres Hindernis war die Präzision der Motoren. Erste Tests zeigten, dass die Bewegungen nicht genau genug waren, wodurch die Muster unsauber wurden. Durch Feinjustierungen der Steuerparameter in MATLAB konnte die Genauigkeit optimiert und eine gleichmäßige Farbauftragung erreicht werden.

Zudem stellte sich heraus, dass die Optimierung der Muster eine genauere Synchronisation der Motoren erforderte. Manche Designs konnten anfangs nicht präzise umgesetzt werden. Durch eine Reihe von Testphasen und Anpassungen der Bewegungsabläufe gelang es, eine verbesserte Abstimmung der Motoren zu erreichen und die gewünschten Muster exakt zu reproduzieren (siehe Abbildung 5).

Durch diese Anpassungen wurde die Stabilität der Konstruktion verbessert, unerwünschte Vibrationen reduziert und die Gesamtleistung des Roboters erheblich optimiert.

C. Coding in MATLAB

Bei der Programmierung des Roboters wurde beschlossen, verschiedene Färbemodi zu implementieren, um eine präzise und vielseitige Gestaltung der Eier zu ermöglichen. Die Steuerung erfolgt über MATLAB [1], wobei die Befehle über ein Kabel an den EV3-Controller gesendet werden. Ein Versuch, Bluetooth zu verwenden, scheiterte, da keine stabile Verbindung hergestellt werden konnte. Daher blieb die kabelgebundene Lösung die einzige praktikable Option.

Der Quelltext ist modular aufgebaut und besteht aus einzelnen Befehlsblöcken, die die Bewegungen der drei Motoren steuern. Jeder Abschnitt definiert die Geschwindigkeit, Drehrichtung und Dauer der Motorbewegungen. Dabei wurden verschiedene Färbemodi programmiert:

Gleichmäßiges Färben: Das Ei dreht sich kontinuierlich, während der Pinsel gleichmäßig Farbe aufträgt. Streifenmuster: Der Pinsel wird in regelmäßigen Abständen gesenkt und angehoben, um ein gestreiftes Muster zu erzeugen.

Die Steuerung erfolgt durch eine for-Schleife, die die kontinuierliche Bewegung der Motoren sicherstellt. Zusätzlich wurde eine if-Bedingung implementiert, um den Farbauftrag zu kontrollieren. Falls das Ei nicht korrekt rotiert oder der Pinsel eine unerwartete Position erreicht, stoppt der Roboter automatisch, um eine fehlerhafte Bemalung zu vermeiden. Die Funktionsweise des Programmes (siehe Abbildung 6).

Der gesamte Quelltext ist darauf ausgelegt, eine präzise Steuerung der Bewegungen zu gewährleisten und gleichzeitig genügend Flexibilität für zukünftige Erweiterungen zu bieten. Die modulare Struktur ermöglicht eine einfache Anpassung der Muster und die Integration neuer Funktionen.

```

motorEgg = motorB;
motorBrush = motorA;
motorMove = motorC;

disp('Wählen Sie einen Malmodus:');
disp('1 - Einheitliche Farbgebung');
disp('2 - Gestreifte Färbung');

mode = input('Geben Sie 1, 2 ein: ');

if mode == 1
    disp('Gleichmäßiges
Färben beginnt...');

    motorBrush.setProperties
    ('power',20,'limitValue', 145);
    start(motorBrush);
    pause(1);

    for i = 1:50
        powerValue = 57;
        motorEgg.setProperties
        ('power',-25);
        start(motorEgg);
        motorMove.brakeMode='brake';
        motorMove.setProperties
        ('power',30,'limitValue',
        powerValue);
        start(motorMove);
        motorMove.waitFor;
        motorMove.setProperties
        ('power',-30,'limitValue', powerValue);
        start(motorMove);
        motorMove.waitFor;
        stop(motorEgg);
    end

    motorBrush.setProperties
    ('power',-20);
    start(motorBrush);
    stop(motorBrush);

    stop(motorMove);
    elseif mode == 2

        for i = 1:13
            motorEgg.setProperties
            ('power',-30);
            start(motorEgg);
            motorBrush.setProperties
            ('power',10,'limitValue', 140);
            start(motorBrush);
            pause(2.5);
            stop(motorBrush);
            motorBrush.setProperties
            ('power',-10);
            start(motorBrush);
            pause(0.5);
            motorMove.setProperties
            ('power',10,'limitValue', 4);
            start(motorMove);
            pause(0.5);
            stop(motorEgg);
        end

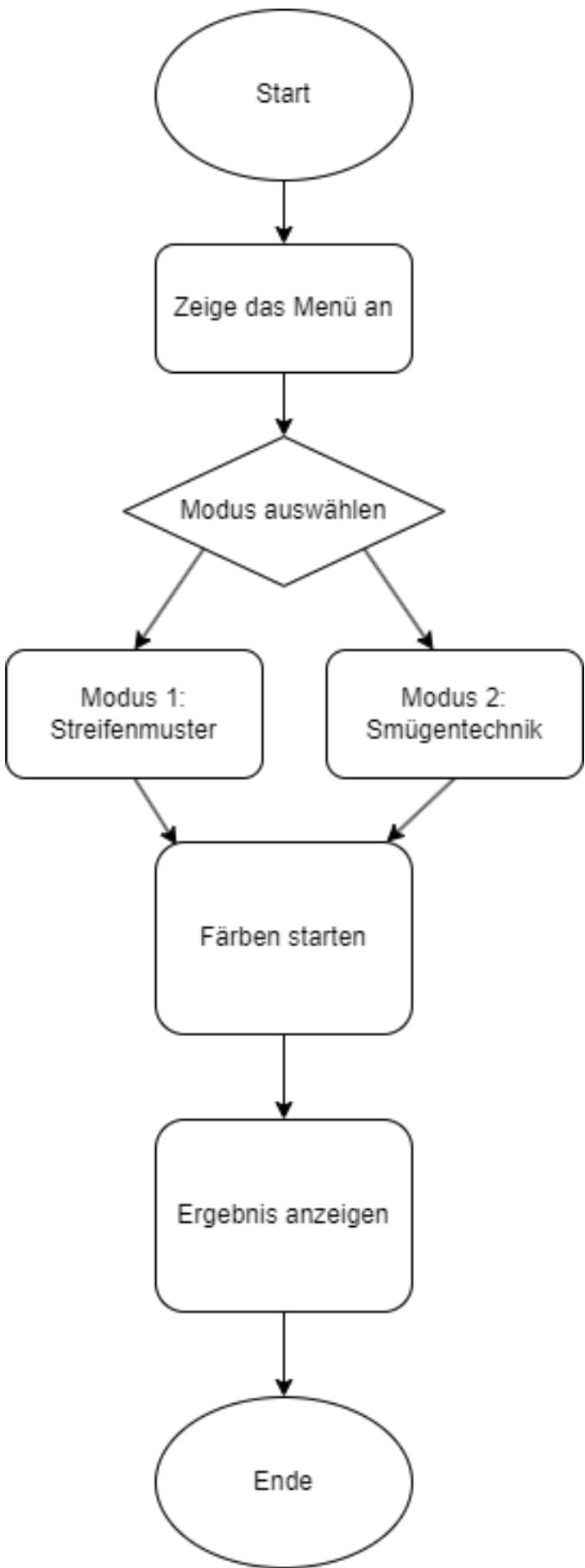
        stop(motorBrush);
        stop(motorMove);
        stop(motorEgg);
        disp('Falsche Wahl!');
    end

    disp('Programm ist abgeschlossen.');

```



Abbildung 5. Bemalte Eier



IV. ERGEBNISDISKUSSION

Nach mehreren Tests mit unterschiedlichen Mustern und Farben wurde bestätigt, dass der Roboter die gestellten Anforderungen erfüllte. Die einzige Einschränkung bestand in der Verbindung zum Laptop über ein Kabel, das für die Steuerung über MATLAB notwendig war. Trotzdem funktionierte die Motorsteuerung zuverlässig, und der Roboter konnte verschiedene Maltechniken präzise ausführen.

Insgesamt wurde der Roboter als stabil und effizient bewertet, wobei kleinere Verbesserungen in der Software und Mechanik das Potenzial für eine noch präzisere Farbgestaltung bieten (siehe Abbildung 5).

V. ZUSAMMENFASSUNG UND FAZIT

Das Projekt erreichte nicht nur das Ziel, einen funktionierenden Roboter zu konstruieren, sondern ermöglichte auch wertvolle Einblicke in die Kombination von Mechanik, Programmierung und Design. Die präzise Steuerung ermöglicht eine gleichmäßige Farbauftragung, und zukünftige Erweiterungen könnten die Effizienz und Funktionalität weiter verbessern.

LITERATURVERZEICHNIS

- [1] MATHWORKS: *MATLAB*. Verfügbar unter: <https://de.mathworks.com/products/matlab.html> Version: 2024
- [2] MINDSTORMS NXT: *NXT Motor*. Verfügbar unter: <https://mindstormsnxt.blogspot.com/2006/08/closer-look-at-nxt-motors.html> Version: 2006

Abbildung 6. Programmablaufplan