

# Emi.mem – Der Memoryroboter

Julian Fürtig, Elektro- und Informationstechnik  
Otto-von-Guericke-Universität Magdeburg

**Zusammenfassung**—Wie bereits seit 2013, fand auch in diesem Jahr 2020 wieder das LEGO Mindstorms Praktikum statt. In diesem sollten die teilnehmenden Studenten eigene Ideen entwickeln und diese mithilfe von LEGO-Mindstorms und MATLAB umsetzen. Ziel war es, nach zwei Wochen bei der Abschlusspräsentation einen funktionierenden Roboter vorzustellen. In diesem Paper wird der Memoryroboter behandelt, der auch ein Ergebnis des diesjährigen Praktikums ist. Ziel des Roboters war das Erkennen von zwölf Memorykarten und danach das richtige Zuordnen der Paare. Obwohl einige Probleme aufgetreten waren, war es möglich, am Ende der zwei Wochen einen funktionierenden Roboter zu präsentieren.

**Schlagwörter**—Emi.mem, LEGO, MATLAB, Memoryroboter, NXT, Otto-von-Guericke-Universität Magdeburg

## I. EINLEITUNG

DER Memoryroboter-Emi.mem sollte in der Lage sein, ein Memoryfeld von  $4 \times 3$  Feldern zu erfassen und die zusammengehörigen Paare finden. Das selbst gesteckte Ziel war es, einen Roboter zu entwerfen, der einen Nutzen hat. So war es mit Hilfe des Emi.mem möglich, ohne Kontakt zu anderen Menschen Memory zu spielen, oder einfach dem Roboter beim Spielen zuzuschauen. Somit stand das Ziel des Projekts fest. Um diese Idee umzusetzen, standen ein NXT-Baustein der als „Kopf“ funktioniert und auf dem der geschriebene Programmcode „läuft“ zur Verfügung. Weiterhin war es möglich, bis zu drei Motoren und vier Sensoren an den NXT-Baustein anzuschließen. Der Roboter sollte selbstständig die Felder anfahren, die Farben erkennen und diese dann speichern. Anschließend sollte er nacheinander die sechs Farben angeben und die dazugehörigen Paare zeigen. Es sollten zwei Spielmodi über ein GUI wählbar sein, ein Einzelspieler-Modus und dann einen für Memory üblichen Zwei-Spieler-Modus. Im Zwei-Spieler-Modus sollte es möglich sein, gegen den Roboter zu spielen.

## II. VORBETRACHTUNGEN

Zur Umsetzung des Roboters wurden zwei Farbsensoren, ein Tastsensor und drei Motoren benötigt. Als Grundlage diente ein Baukasten der LEGO-NXT-Serie und ein weiterer mit LEGO-Technik Teilen. Als erstes war es notwendig, sich mit der Funktion der Sensoren und der Motoren vertraut zu machen. Damit wird sich der nachfolgende Abschnitt beschäftigen.

### A. Der Memoryroboter

Der Memoryroboter orientiert sich zur Paarsuche an Farbmarkierungen außerhalb des Feldes, um die richtigen Positionen zu erreichen. Emi.mem beginnt damit, das gesamte Feld einzuscannen, um sämtliche Karten zu erfassen. Anschließend

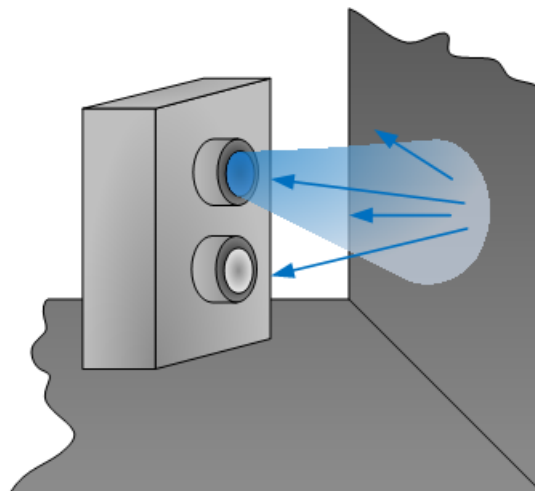


Abbildung 1. Prinzip des Farbsensors

fährt er die Paare einzeln an, um zu zeigen, wo sich diese befinden. Nachdem der Roboter wieder die Ausgangsposition erreicht hat, wird die Farbe via Sprache ausgegeben. Ein Zwei-Spieler-Modus ist ebenfalls vorhanden, um das Memory-Spiel nachzuempfinden.

### B. Der Farbsensor

Der Farbsensor wird verwendet, um verschiedene Farben zu erkennen. Mit dem RGB-Sensor von LEGO ist es möglich, sechs verschiedene Farben zu erkennen und die entsprechende Farbe als Zeichenkette (String) in MATLAB ausgegeben zu lassen. Der Sensor kann durch Anstrahlen der zu erkennenden Oberfläche mit Licht in den Farben Rot, Grün und Blau anhand der Reflexionen die Farbe der Oberfläche ermitteln. Beispielsweise wird Weiß durch die gleich starke Reflexion der drei Farben ermittelt, während Schwarz durch geringe Reflexion der drei Farben bestimmt wird. Wenn nur eine der drei Farben zurückgeworfen wird hat die Karte die entsprechende Farbe. Gelb wird durch eine gleichstarke Reflexion von rot und grün ermittelt, blau wird dabei nicht zurückgeworfen. Die Funktionsweise des Farbsensors ist in Abbildung 1 erkennbar.

### C. Bedienung des Roboters

Der Roboter sollte mithilfe eines Tastsensors gestartet werden und nach dem Einlesen der Karten den „Spielvorgang“ starten. Der Programmablauf wurde am Ende mit einem GUI (Graphical User Interface) gestartet und nicht mit dem Tastsensor, da vorher noch der Spielmodus ausgewählt werden muss. Das eigentliche Spiel wurde aber weiterhin über den Tastsensor gestartet.

### III. HAUPTTEIL

#### A. Erste Schritte

Die erste Überlegung war, einen Roboter nach dem Prinzip eines Portalkrans zu bauen, der sich direkt über dem Memoryfeld befindet. Die Umsetzung dieser Idee war aber nicht möglich, da das senkrechte Herablassen des Farbsensors auf die Memorykarten ähnlich des Kranhakens mit LEGO schwierig umsetzbar war. Deshalb wurde auf die aktuelle Lösung zurückgegriffen, den Roboter neben dem Memoryfeld zu platzieren. Nachdem die Position des Roboters feststand, wurde mit ersten Versuchen begonnen, den Roboter neben dem Feld zu bewegen. Dazu wurde ein Grundrahmen aufgebaut, an dem vier Räder befestigt wurden. Zunächst wurde nur eine Achse mit einem außenliegenden Motor angetrieben. Dieses Chassis fuhr entlang der in Abbildung 3 sichtbaren Orientierungsmarken. Auf dieses Chassis wurde dann eine um 90° gedrehte Schienenkonstruktion aus Zahnschienen gesetzt, um die Armbasis entlang der „Feldspalten“ zu bewegen und so die Erreichbarkeit jedes Feldes zu gewährleisten. Auch die obere Basis wurde mithilfe eines eigenen Motors bewegt. Die erste Konstruktion sah vor, einen beweglichen Arm zu verwenden, um auch die hintere Feldreihe zu erreichen. Dies ist in Abbildung 5 dargestellt. Parallel zur mechanischen Umsetzung des Roboters wurde sich mit der Funktionsweise der Farbsensoren vertraut gemacht. Auch das Ansteuern der Motoren und das Abfragen des Tastsensors wurden ausprobiert.

#### B. Probleme

Die erste Version des Roboters war noch nicht perfekt und benötigte noch einiges an Optimierungen. Der Roboter war zum einen nicht sehr standsicher, zum anderen war durch den beweglichen Arm die Treffsicherheit des Farbsensors auf die Karten nicht immer gegeben. Des Weiteren gab es mehrere Probleme beim Anfahren der unteren Einheit. Eine weitere Herausforderung war, dass der obere Wagen bei ausgefahrenem Arm zum nach-vorn-Kippen neigte. Noch eine Schwierigkeit war, dass der Roboter die „Feldzeilen“ nicht genau genug anfuhr. Weitere Probleme waren, dass die Farbsensoren einen sehr geringen Abstand zur erfassenden Oberfläche und auch sehr große Farbflächen benötigten, damit die Messung zuverlässig funktionierte. Des Weiteren durfte das Umgebungslicht nicht zu hell sein, damit der Farbsensor die richtige Farbe erkannte.

#### C. Problemlösung

Um das Problem der Standsicherheit zu lösen, wurde der außen montierte Motor des Chassis nach innen verlegt und der NXT auf eine Seite über die Räder des Chassis verschoben, sodass der Schwerpunkt zwischen die Achsen rutschte und der Roboter nicht mehr zum Kippen neigte. Das Kippen des oberen Wagens wurde durch den Umbau von einem beweglichen Arm auf einen festen Arm unterbunden. Um dies zu realisieren, musste die Zahnschienenkonstruktion verlängert werden und der zuvor für das Ausklappen des Arms verantwortliche Motor wurde nun als Gegengewicht für den verlängerten Arm verwendet. Die Treffsicherheit des Farbsensors wurde durch den Wegfall der Armgelenke deutlich

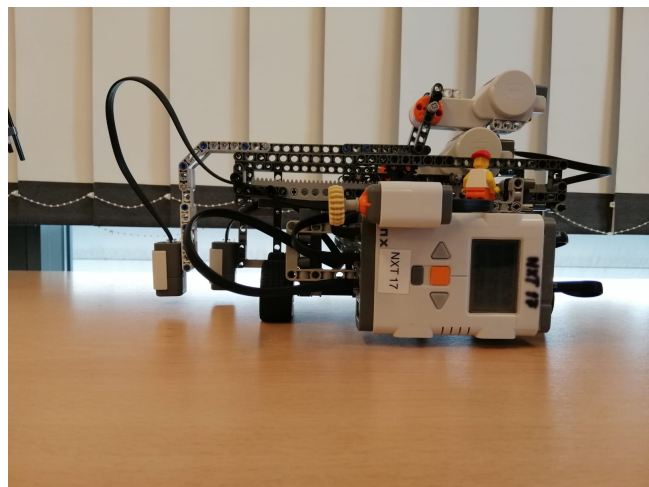


Abbildung 2. Rückansicht des Roboters, gut erkennbar sind die beiden Farbsensoren

verbessert. Ein Problem, welches die steifen Kabel verursachten, war, dass der obere Wagen „entgleiste“. Dieses Problem konnte durch das Anbringen von seitlichen Führungsschienen meist unterbunden werden. Ein Umkippen beim Anfahren konnte ebenfalls durch das Verschieben des Motors in das Chassis sowie durch das Antreiben aller vier Räder gelöst werden. Des Weiteren wurde die Kraft des Motors reduziert und durch den „Allradantrieb“ auch besser verteilt. Die Genauigkeit des Anfahrens der „Feldzeilen“ wurde durch das Anbringen eines zweiten Farbsensors realisiert. Dieser orientierte sich an den Markierungen neben dem Feld. Die rote Markierung legte den Startpunkt des Roboters fest, die schwarzen Markierungen jeweils eine Zeile. Dies stellt Abbildung 3 dar. Der geringe Abstand zur Oberfläche war ein Problem, da sich der Arm mit zunehmender Länge etwas nach unten bog und somit auf den Karten auflag, wodurch er kein Licht „einfangen“ und so nicht mehr messen konnte, weshalb die Höhe des Sensors sehr wichtig war. Das Ausrichten des zweiten Farbsensors zur Orientierung war auch ein Problem, da die Orientierungsmarken zu schmal waren, um durch den Sensor zuverlässig erkannt zu werden. In der finalen Version sind diese Marken noch breiter als in der Abbildung 3, um zu verhindern, dass der Roboter eine Markierung übersah und einfach weiterfuhr. Das hätte zur Folge, dass der gesamte Ablauf neu gestartet und der Roboter wieder manuell ausgerichtet werden müsste. Der Aufbau des Roboters ist in Abbildung 4 zu erkennen.

#### D. Programm

Nachdem sich das Team mit der Funktionsweise und Ansteuerung der Sensoren und Motoren vertraut gemacht hatte, wurde begonnen, die gewünschten Bewegungen in Quellcode umzusetzen. Als erstes wurde der Vorgang des Einlesens programmiert. Dazu wurden for-Schleifen verwendet, um alle Felder erfassen zu können. Damit der Roboter die einzelnen „Feldzeilen“ richtig anfuhr, wurden die Farbmarkierungen am Rand des Memoryfeldes mit eingebunden. Der Fahrmotor wurde solange bewegt, bis der zweite Farbsensor die schwarze Markierung erkannt hatte. Damit der Farbsensor beim erneuten

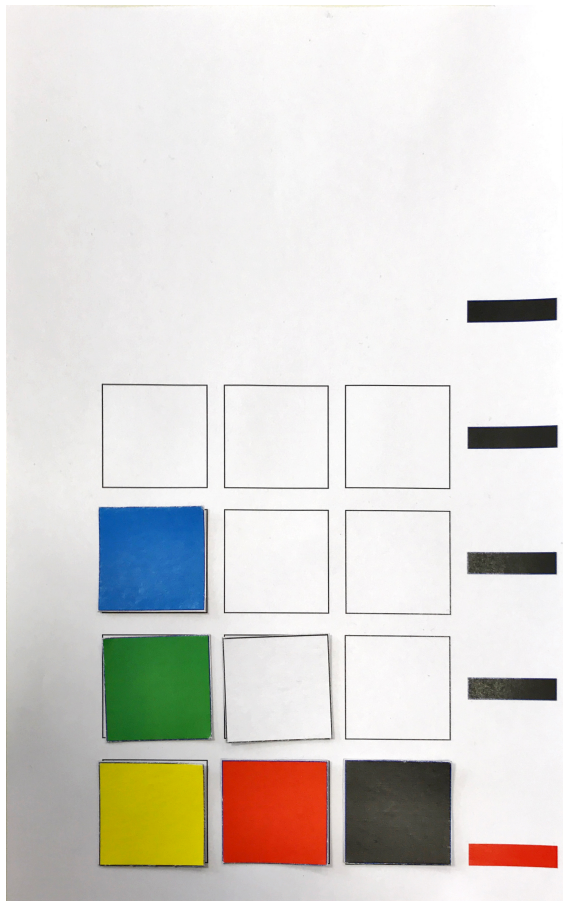


Abbildung 3. Memoryfeld mit Orientierungsmarken

Losfahren nicht nochmal die schwarze Markierung erkannte, wurde eine kurze Pause in das Programm mittels Pause-Befehl eingebaut. Erst nach wenigen Sekunden wurde das Programm fortgesetzt und so das doppelte Erfassen der Markierungen verhindert. Das Einlesen der Karten ist in Abbildung 4 zu sehen. Um einen Startpunkt für das Spiel zu erhalten, wurde die rote Markierung angebracht. Diese Markierung wurde erst nach dem Erfassen des gesamten Feldes benötigt, um das Programm zu vereinfachen, da das Zählen von schwarzen Markierungen einen höheren Programmieraufwand zur Folge gehabt hätte. Das Anfahren der Felder in die andere Richtung auf der zweiten Ebene wurde dann über Laufzeiten des Motors realisiert. Der Motor lief abhängig von dem anzufahrenden Feld eine längere Zeit oder eben eine kürzere. Das Einlesen der Felder erfolgt über ein Array (Feld) welches dem Memoryfeld nachempfunden war.

Nach dem Beenden des Einlesevorgangs begann der Roboter damit, die Felder sequentiell miteinander zu vergleichen. Die erste Memorykarte, in Abbildung 3 die schwarze Karte, wurde mit den restlichen Karten verglichen. Sobald der Roboter die zugehörige Karte gefunden hatte, startete er den zweiten Teil des Programms. In diesem Teil wurde zuerst jedem Feld zugeordnet, wie sich der Roboter bewegen muss. Dies wurde mittels Koordinaten des Arrays realisiert. Jeder Arraykoordinate wurde ein eindeutiger Weg zugeordnet, sodass es möglich war, jedes Feld unabhängig anzufahren. Bei der nächsten Farbe wurde

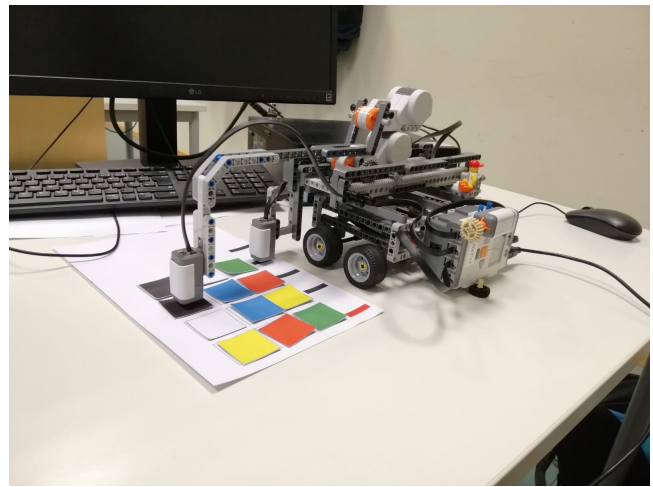


Abbildung 4. Roboter in Aktion

das erste Feld übersprungen und mit dem zweiten Feld weiter gemacht. In der Abbildung 3 ist das die rote Memorykarte. Jetzt verglich er wieder diese Karte mit den anderen und wenn er die dazu passende Karte gefunden hatte, wurden wieder die beiden Felder angefahren und zum Schluss die gefundene Farbe sprachlich ausgegeben. Die Sprachausgabe wurde realisiert, indem zuvor die Farben eingesprochen und anschließend über den im NXT verbauten Lautsprecher ausgegeben wurden.

Der Zweispieler-Modus begann damit, dass der Roboter erst das Memoryfeld einlas. In dieser Zeit konnte auch der menschliche Spieler sich das Memoryfeld merken. Dieser durfte dann beginnen und so lange weiter spielen, bis er einen Fehler beging. Erst dann war der Roboter an der Reihe. Dieser fand jetzt mit einer Wahrscheinlichkeit von 60% ein passendes Paar. Dies war notwendig, da es sonst unmöglich wäre, gegen den Roboter zu gewinnen, da dieser eigentlich nicht „vergessen“ kann. Die Umsetzung erfolgte mittels Zufallsgenerator, welcher mit einer voreingestellten Wahrscheinlichkeit von 60% zu einem Erfolg des Roboters führte, mit einer Wahrscheinlichkeit von 40% zu einem Misserfolg und damit zur Ausgabe einer Fehlermeldung. Die Eingabe des realen Spielers erfolgte im GUI, dort war das Memoryfeld dargestellt. Der Spieler musste dort seine gefundenen Felder markieren, damit der Roboter nicht versuchte, diese Felder zu suchen.

#### IV. ERGEBNISDISKUSSION

Abschließend, war das Projekt Memoryroboter ein Erfolg. Die schwerwiegendsten Probleme konnten behoben werden und auch der Zweispieler-Modus konnte zum Schluss noch erfolgreich getestet werden. Eine Einzelspieler-Runde dauerte knapp 6 Minuten, welche auf die konstruktionsbedingt langsamen Bewegungen zurückzuführen sind. Ein ungelöstes Problem blieb, dass das Ausrichten des Roboters am Feld nicht immer exakt genug war, sodass er sich nicht parallel zum Feld bewegte, wodurch gelegentlich ein Neustart des Programms erforderlich war. Um dies zu verhindern, war es manchmal notwendig, die Position des Memoryfeldes zu korrigieren, da sonst der Vorgang unnötig lange dauerte. Das Umdrehen der Karten zu realisieren war nicht möglich, da dazu noch mindestens

zwei weitere Motoren nötig gewesen wären. Die Anzahl der Motoren war aber durch den NXT-Baustein auf maximal drei beschränkt.

#### V. ZUSAMMENFASSUNG UND FAZIT

Der Memoryroboter-Emi.mem war in der Lage alleine, aber auch gegen einen anderen Spieler zu spielen. Im Ein-Spieler-Modus benötigte der Roboter etwa 6 Minuten für ein Spiel, während im Zwei-Spieler-Modus, abhängig von der Eingabegeschwindigkeit des menschlichen Mitspielers, deutlich mehr Zeit erforderlich war. Zukünftig wäre es möglich, mit Hilfe eines zweiten NXT das Umdrehen der Karten zu ermöglichen, sodass es dem realen Memoryspiel näher käme. Des Weiteren wäre es möglich, den Farbsensor durch ein Webcam zu ersetzen, und so nicht nur Farben, sondern auch Motive zu erkennen, und die Paare zu finden.

#### ANHANG

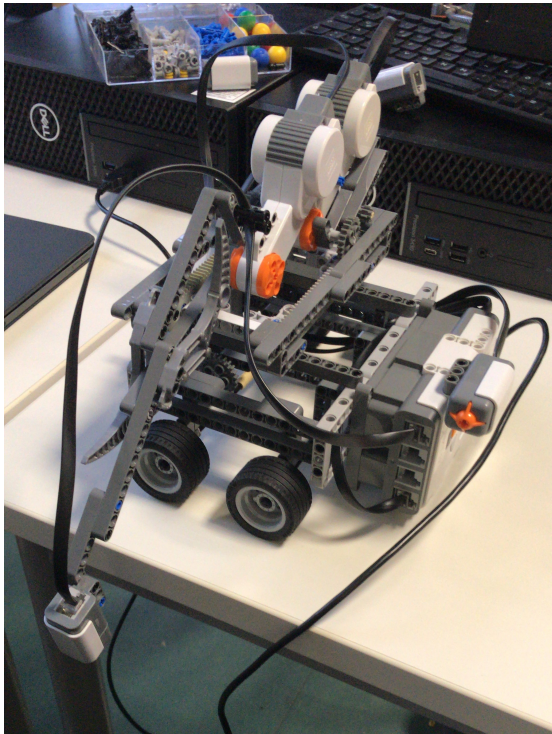


Abbildung 5. Erster Prototyp mit beweglichem Arm und kurzem oberem Aufbau

#### LITERATURVERZEICHNIS

Abbildung 1 - Felix Bangemann: Vortrag Sensoren.  
<https://elearning.ovgu.de/mod/resource/view.php?id=44971>.  
Februar 2020