

Der elektrische Morseapparat, kurz: El-Mo

Julian Sebastian Wenzel, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—In dem folgenden Dokument werden die Planung, die Konstruktion und der Bau, sowie die MATLAB-Programmierung eines neu gedachten Morsegerätes dargelegt. Im Folgenden werden zwei Geräte vorgestellt, da es zwei Konstruktionen unterschiedlicher Machart und Programmiererteile gibt. Grundsätzlich funktionieren beide Geräte derart, dass sie äußere Signale erhalten, auswerten und diese decodiert über den Bildschirm ausgeben.

Das erste der beiden Geräte wurde über zwei Taster angesteuert und verglich deren Werte. Das zweite, welches der tatsächliche El-Mo-Apparat –im Folgenden nur noch als „El-Mo“ bezeichnet ist, las einen Papierstreifen mit einer Strichcodierung ein.

Schlagwörter—Codierung, Elektrotechnik, LEGO Mindstorms, Morsen, Projektseminar

I. EINLEITUNG

Die heutzutage vorherrschende Meinung ist, dass in der Welt der Nachrichtenkommunikation der Morsecode, seiner extensiven Natur wegen, keinen Platz mehr hat. Doch sollte er deswegen ganz in Vergessenheit geraten? Immerhin ist er nicht digital derart abzufangen, wie es zum Beispiel E-Mail und SMS sind, kann also eine gewisse Art der digitalen Sicherheit bieten. Noch dazu kommt, dass der Morsecode keine ausgedehnten Apparaturen und Geräte benötigt, kann also in Situationen Verwendung finden, in denen zum Beispiel Platz eine große Rolle spielt.

Um also diese Art der Kommunikation am Leben zu erhalten, wird im Rahmen des Projektseminars ein Prototyp für eine neue Möglichkeit des Einlesenst des Morsecodes entwickelt. Diese wäre, dass die zu übermittelnde Botschaft nicht per Kontaktenschließung eines Stromkreises verschlüsselt wird, sondern — einem Barcode ähnlich — von einem Papierstreifen automatisch abgelesen werden kann. Offensichtlich ist dafür ein Zusammenspiel von Sensorik und Motoren von Nöten. Die Herausforderung liegt nun darin, die Präzision in dieser Zusammenarbeit zu gewährleisten.

II. VORBETRACHTUNGEN

A. Morsecode

Der erste Schritt für dieses Projekt war, die Funktionsweise des Morsens zu verstehen. Grundlegend für das Morsen ist, dass „zwei verschiedene Zustände (wie etwa Ton oder kein Ton) eindeutig und in der zeitlichen Länge variierbar dargestellt werden können. Dieses Übertragungsverfahren nennt man Morsetelegrafie.“ [1] Daraus folgt, dass mindestens zwei Werte zu beachten sind, sollte dieses Verfahren verwendet werden. Theoretisch könnten Zusatzsignale vereinbart werden, da es hier keine Beschränkungen gibt. Betrachtet man nun das Morsealphabet, so fällt auf, dass die einzelnen Buchstaben nach

ihrer Häufigkeit im englischen Sprachgebrauch codiert sind. Demzufolge sind „E“ und „T“ am einfachsten zu versenden, da sie jeweils nur aus einem Zeichen bestehen.

B. Erprobung der Umsetzungsmöglichkeiten

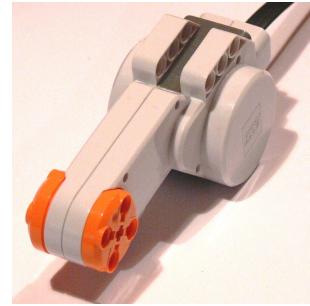
Der zweite Schritt dieses Projektes bestand darin, zu testen, ob sich die Idee erst einmal mit einer sensorlosen Konstruktion umsetzen ließe, und nur auf User-Input angewiesen wäre. Als hier die Problemstellen und deren Lösungen identifiziert waren, konnte in die nächste Phase übergegangen werden. Nach dem Bekanntmachen mit der Programmierumgebung Matlab und des LEGO-Mindstorms-Sets, wurde es klar, dass die Programmierung den größten Teil der Arbeit ausmachen würde. So wurden zuerst Tests zur richtigen Verwendung von Lichtsensoren (siehe Abbildung 1a), Tastern (siehe Abbildung 1b) und Motoren (siehe Abbildung 1c) durchgeführt, so dass die spätere Verwendung darauf fußen konnte.



(a) Lichtsensor



(b) Tastsensor



(c) Servomotor

Abbildung 1. Verwendete Sensoren und Motoren im Verlauf des Projekts von LEGO Mindstorms

III. UMSETZUNG

A. Prototypentwicklung

Es wurde also zuerst ein durch Taster angesteuerter Morseapparat gebaut. In dessen erster Entwicklungsphase wurde versucht, die Eingabesignale, derer es zwei gab, durch nur

einen einzigen Taster zu erreichen. Die Unterscheidung der langen und kurzen Signale sollte durch eine Stoppuhrfunktion erreicht werden, sodass einzig die Länge des Signals ausschlaggebend sein sollte. Da dies scheiterte, wurde der Ansatz der Vergleichsmessung verfolgt; ein weiterer Taster musste betätigt werden, damit die Signale aufgenommen werden konnten. Mit dieser Methodik wurden erfolgreich die ersten Buchstaben übermittelt.

B. Programmierung

Um die Werte der Taster abzuspeichern, wurde ein Array aus Einsen erstellt. Kamen nun Signale über die Taster ein, unterschieden sich diese im Programm in ihren Werten. Bei langen Signalen wurde im Array eine Eins addiert, bei kurzen änderte sich nichts, und die Trennungen zwischen diesen beiden wurde mit einer 0 dargestellt. Der Ablauf einer Programmschleife, wird in der Abbildung 2 verdeutlicht:

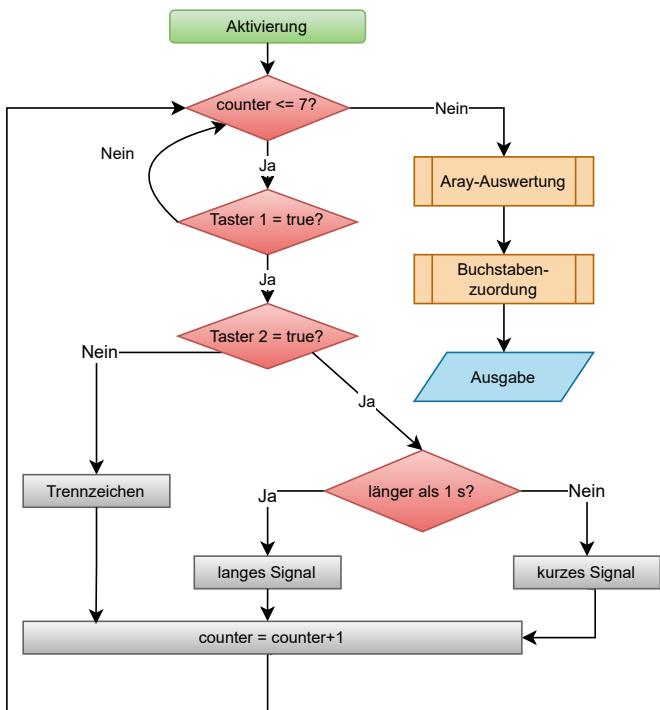


Abbildung 2. Flussdiagramm für den Prototypen von El-Mo

Wie zu sehen ist, wird in Abbildung 2 auch die Entschlüsselung der Array-Werte dargestellt. Diese erfolgte über eine for-Schleife, in der eine Variable rekursiv mit Werten über eine Summation befüllt wurde. Im folgenden wird die dafür zuständige Gleichung erläutert, dabei wird die Variable mit dem Wert als „W“ bezeichnet und der Array-Wert bei der i-ten Iteration „A(i)“:

$$W = W + A(i) \cdot 10^{(7-i)} \quad (1)$$

Um es etwas zu verdeutlichen, hier ein Beispiel: Hat man den Buchstaben „N“ gemorst, steht in dem Array die Zahlenfolge 2,0,1,0,0,0,0. Durch die Anweisung wird „W“ zuerst auf $2 \cdot 10^6$ gesetzt. In dem dritten Durchlauf wird dann dazu noch $1 \cdot 10^4$

addiert, so dass das Ergebnis 2010000 wäre. Diese Zahl wäre der Stellvertreter für den Buchstaben „N“. Danach wird in einer switch-Anweisung dieser Zahl der korrekte Buchstabe zugeordnet und auf der MATLAB-Oberfläche ausgegeben.

C. Mechanische Aufgaben

Der eigentliche El-Mo, hatte zwei große mechanische Hürden zu meistern. Zum einen erfolgte die Signaleingabe nun nicht mehr über Taster, sondern über Papierstreifen, welche mit einer Art Barcode beschrieben waren. Doch damit El-Mo mit diesen arbeiten konnte, mussten die einzulesenden Streifen sehr genau gezeichnet sein. Zum anderen hat der Automatismus zum Einziehen des Streifens vorgeschriebene Eckpunkte einzuhalten. Die Breite der Segmente des Codes auf dem Papier, wurde in Zentimeterschritten gemessen. Darum musste auch El-Mo in diesen Schritten arbeiten können. Durch eine begrenzte Auswahl an Radgrößen musste also die Programmierung dafür sorgen, dass der Motor dieses Rad im richtigen Takt um die richtige Winkelzahl bewegt. Durch Messungen und anschließende Rechnungen wurde es irgendwann klar, dass der Motor sich um 24° zu drehen hat, damit El-Mo seine Aufgabe zufriedenstellend lösen konnte.

Erschwernd zu der Präzisions-Problematik beim Einlesen des Strichcodes kam, dass der Papierstreifen sicher und ohne zu verrutschen eingezogen werden konnte. Um dies zu gewährleisten, wurde die Gummierung des Rades mit verbaut. Daraus entstand jedoch ein sehr hoher Reibungswiderstand zwischen Gummi und Unterlage. Infolge dessen hatte der Motor immense Schwierigkeiten damit, die nötige Leistung aufzubringen, den Streifen um die richtige Distanz zu bewegen. Dieses Problem wurde mit einer Zahnradübersetzung im Verhältnis von 1:6 gelöst. So konnte die Motorsleistung um das Sechsfaache gesteigert werden, womit der Reibungswiderstand ausgeglichen werden konnte.

D. Codierungs Normierung

Damit El-Mo mit der Strichcodierung arbeiten konnte, mussten bestimmte Normen erstellt werden. Zum einen galt es, festzulegen, dass der Beginn jedes Strichcodes mit einem weißen, einen Zentimeter breiten Segments beginnt. Das spielt bei dem Einziehen des Streifens eine wichtige Rolle, da dieser Prozess mit der Änderung der gemessenen Helligkeit arbeitet (s. Abschnitt E: „El-Mo’s Programm“). Zum anderen musste klar festgelegt werden, wie breit die Segmentierung der Streifen sein durfte. Da dies willkürlich gewählt werden konnte, wurde beschlossen, dass die Länge der Signale mit Zentimetereinheiten arbeiten soll. Einen Überblick zu den Normierungen findet sich in Tabelle 1 „Übersicht zur Codierung der Papierstreifen“. Wie diese in der Praxis umgesetzt wurden, ist in Abbildung 3 zu sehen.

*Tabelle 1
Übersicht zur Codierung der Papierstreifen*

Signalart	Farbe	Länge in cm	Array-Wert
langes Signal	Schwarz	2	2
kurzes Signal	Schwarz	1	1
Trennzeichen	Weiß	1	0

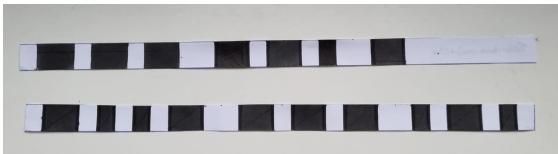


Abbildung 3. Beispiel eines für El-Mo lesbaren Strichcodes
oben: S, G, T
unten: X, O, R

E. Grundsätze der Funktionsweise von El-Mo

Viele Lösungsansätze, die bei der Bearbeitung des Prototypen entwickelt wurden, konnten tatsächlich in die Programmierung von El-Mo übertragen werden. Einzig die Aufgabe des automatischen Einziehens ist gänzlich neu gewesen. Nach dem Positionieren des Streifens wird der Motor aktiviert und das Papier wird Richtung Sensor geschoben. In einer Kontrollsleife wird geprüft, ob der Sensor etwas Helles detektiert. Ist dies der Fall, so wird der Motor kurz gestoppt. Anschließend wird der Streifen so platziert, dass der Sensor über der 5 cm-Marke, vom Beginn des Streifens gemessen, schwebt.

Von hier an wird der Papierstreifen um die Länge von einem Zentimeter eingezogen, die Helligkeitsstufe des Segments geprüft und mit der Helligkeitsstufe des zuvor eingelesenen Segments verglichen. Sind diese identisch (Schwarz-Schwarz), so wird ein langes Signal detektiert und eine „2“ in das Array eingetragen. Sind sie nicht identisch – entweder Schwarz-Weiß oder Weiß-Schwarz – so wird im ersten Fall ein kurzes Signal eingelesen und eine „1“ in das Array eingetragen und im zweiten Fall ein Trennzeichen, bei dem eine „0“ in das Array geschrieben wird.

Zum Schluss wird der Streifen erneut positioniert, damit der nächste Vergleich beginnen kann. Ist die maximale Anzahl von Zeichen erreicht, wird die Prüfschleife beendet. Das Auslesen des Arrays und die Buchstabenzuordnung konnte eins zu eins aus dem Prototypen übernommen werden.

IV. ERGEBNISDISKUSSION

Das Ergebnis von zwei Wochen intensiver Arbeit lässt sich wie folgt zusammenfassen: Der Apparat erkennt problemlos die Segmente des Codes und gibt den entschlüsselten Buchstaben korrekt aus. Eine Momentaufnahme dieses Prozesses ist in Abbildung 4 zu sehen.

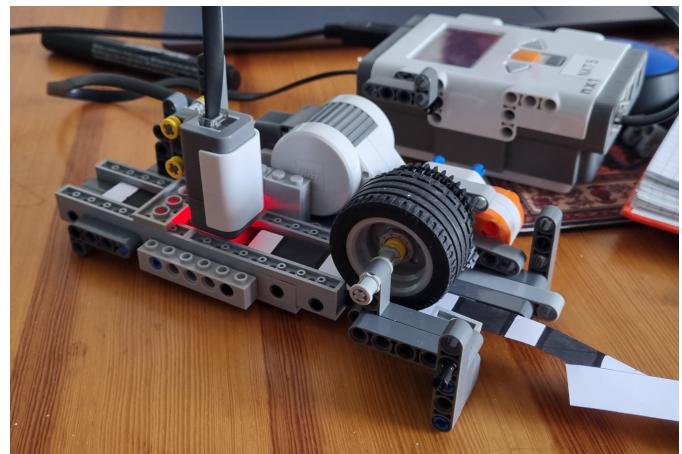


Abbildung 4. El-Mo mit Lichtsensor, während ein Streifen eingelesen wird. Hinter dem Rad ist die Zahnradübersetzung zu erkennen.

Die größten unerwarteten Herausforderungen traten weniger im Programmierteil als in mechanischen Ungenauigkeiten zutage. Bis kurz vor der Fertigstellung zog der Motor immer etwas zu viel des Papiers ein, was nach einigen Segmenten zum falschen Einlesen des Codes führte. Letztendlich bestand die Lösung darin, zu erkennen, dass die Fehlerkorrektur über das Anpassen der Gradzahl, um die sich der Motor zu drehen hatte, erfolgen konnte, sowie in der Erkenntnis, dass eine Unterstützung mit einer Zahnradübersetzung nötig war.

V. ZUSAMMENFASSUNG UND FAZIT

Abschließend ist zu sagen, dass unter den gegebenen Bedingungen die erbrachte Leistung optimal war, wobei auch über Potential zur Verbesserung nachgedacht wurde. Sollte El-Mo zu einem späteren Zeitpunkt weiterentwickelt werden, so wäre es überlegenswert, zu versuchen, die Ausgabe der Buchstaben nicht über MATLAB erfolgen zu lassen. Vielleicht bietet sich hier eine elegantere Lösung, möglicherweise sogar das Koppeln mehrerer NXT, die über Bluetooth mit einander verbunden sind.

VI. LITERATURVERZEICHNIS

[1] Morsecode, Wikipedia, 2024.02.14, 13:18 Uhr
<https://de.wikipedia.org/wiki/Morsecode>

VII. ANHANG

Für ein besseres Verständnis der Unterschiede zwischen dem Code des Prototypen und El-Mo lässt sich folgendes sagen: Vom Prinzip her ähneln sich die beiden, da es in beiden gleichende Abschnitte für die Signalarten gibt, mit dem Unterschied, wie diese eingelesen werden. El-Mo jedoch arbeitet auch mit einem Motor und hat sehr viel mehr Vergleichsschleifen, die zu einem viel größeren Code-Gebilde führen, als es beim Prototypen der Fall ist. Dies ist für den visuell Ausgelegten möglicherweise besser mit Hilfe der Abbildung 5 nachzuvollziehen

Da es wahrscheinlich seltsam erscheinen mag, warum ein Flussdiagramm vom Prototypen aber nicht von El-Mo im Dokument auftaucht, so ist dieses als Abbildung 5 hier zu finden.

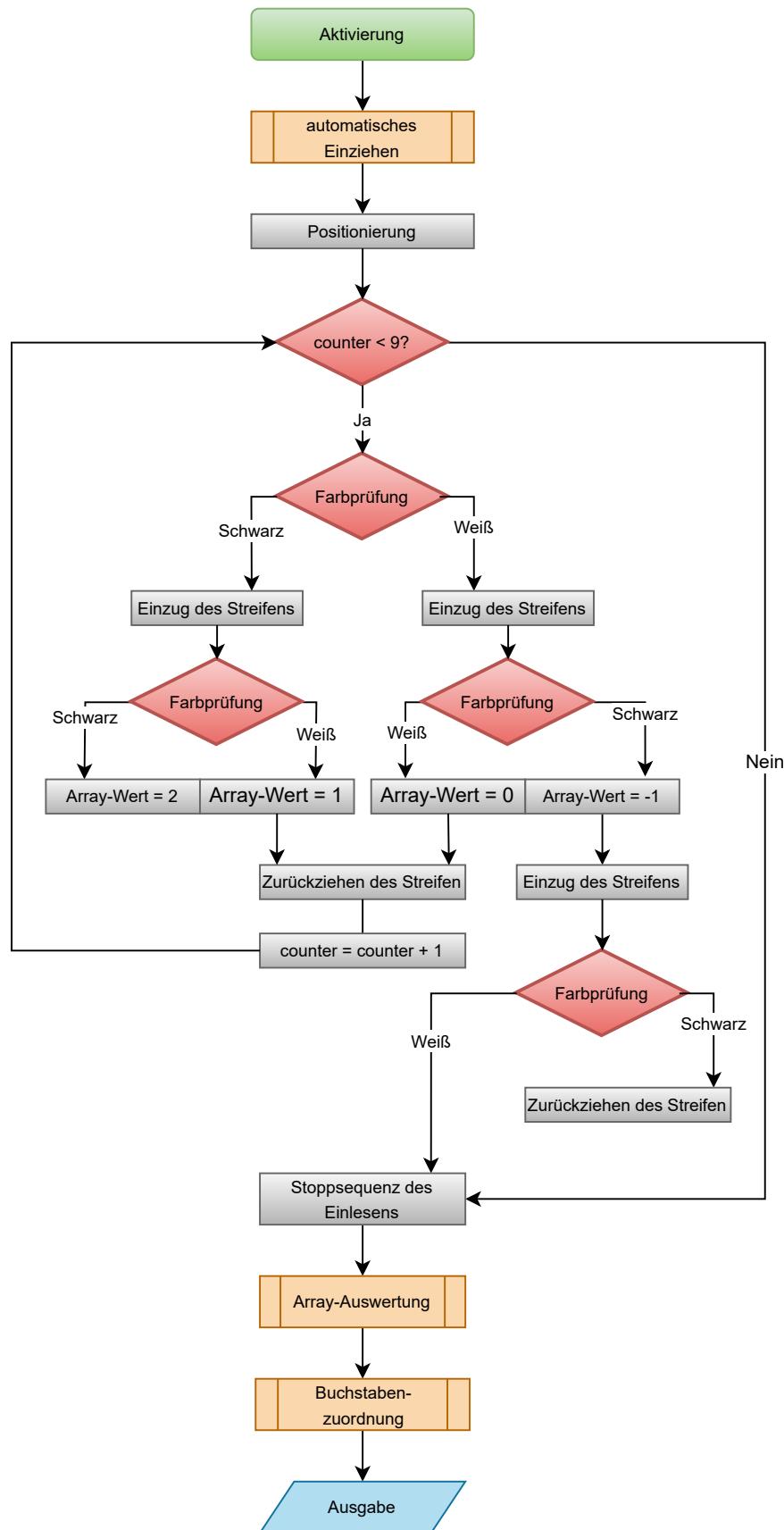


Abbildung 5. Flussdiagramm des Codes von El-Mo