

Oszilloskop-Plotter

Ahmad Alhamid, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung— Das Projektseminar Elektrotechnik und Informationstechnik wird jedes Jahr an der Otto-von-Guericke-Universität durchgeführt. In diesem Jahr wurde ein Oszilloskop-Plotter als Kunstprojekt angefertigt. Dieses Projekt hat keinen technischen Nutzen und löst kein Problem. Es ist ein Kunstprojekt und diese Kunst nutzt das Wissen über Elektrizität, in dem eine Person wie eine Etch a sketch zeichnen kann, aber mit Hilfe von Spannungsänderung in Potentialmeter

Schlagwörter Potentiometer, MATLAB, Oszilloskop, Gleichstrommaschine, LEGO Mindstorms, Spannung.

I. EINLEITUNG

Das LEGO Mindstorms-Oszilloskop-Plotter-Projekt ist eine Demonstration der Verwendung von LEGO Mindstorms Robotik, um das klassische Etch a Sketch-Spielzeug nachzubauen. Das Projekt beinhaltet die Verwendung von zwei Potentiometern, die mit einem Oszilloskop verbunden sind, wobei jedes LEGO Motor die Bewegung eines Potentiometer steuert. Die mit Zahnrädern ausgestatteten Motoren bewegen die Potentiometer auf eine Weise, die dem Malmechanismus eines Etch-A-Sketch ähnelt.

II. VORBETRACHTUNGEN

Für das Projekt werden LEGO-Mindstorms-Komponenten wie Motoren, Getriebe, LEGO-Teile und ein programmierbarer Steuerungscomputer verwendet. Mit Potentiometern wird die Position eines Stiftes Rotation auf der x- und der anderer y-Achse gesteuert, um die Bewegung eines Etch a Sketchs zu imitieren. Während sich die Motoren drehen, passen sie die Positionen der Potentiometer an und ermöglichen so das Zeichnen von Formen und Mustern auf dem Oszilloskop, indem sie die Spannungsänderung im Potenzialmesser messen. Das Projekt kombiniert Elemente der Robotik, Elektronik und Programmierung, um die nostalgische Erfahrung der Verwendung eines Etch-A-Sketchs in einem modernen Kontext nachzubilden.

Die Integration eines Oszilloskops ermöglicht es, die elektrischen Signale in Echtzeit zu visualisieren und das Verhalten des Systems zu analysieren. Dies hilft, ein tieferes Verständnis der grundlegenden physikalischen Prinzipien zu erlangen und ermöglicht die Weiterentwicklung der technischen Fähigkeiten.

A. Motivation

Die Motivation war, dass ein kleines Projekt bereits

gemacht wurde und in den sozialen Medien von Dr. Mathias Magdowski hochgeladen wurde und die Potentiometer manuell mit der Hand gedreht wurden und die Idee war, dies mit einigen NXT-Komponenten und Programmcodes zu automatisieren. Dies wird helfen, das Projekt zu nutzen, um viel stabilere und komplexere Zeichnungen zu zeichnen.

B. LEGO NXT

LEGO Mindstorms NXT stellt eine vielseitige Robotikplattform dar, konzipiert und entwickelt von der renommierten LEGO-Gruppe. Das NXT-System besteht aus einer Reihe von Komponenten, die es den Benutzern ermöglichen, komplexe robotische Systeme zu erstellen. Mit einem Fokus auf technische Präzision bietet das LEGO eine visuelle Programmierumgebung, die es den Nutzern ermöglicht, maßgeschneiderte Programme zu erstellen. Dies geschieht durch das Verbinden von Blöcken, die eine Vielzahl von Aktionen und Steuerungsmechanismen repräsentieren [1]. Diese Blöcke ermöglichen es die Funktionalität des Roboters in einem intuitive Format zu konfigurieren und anzupassen.

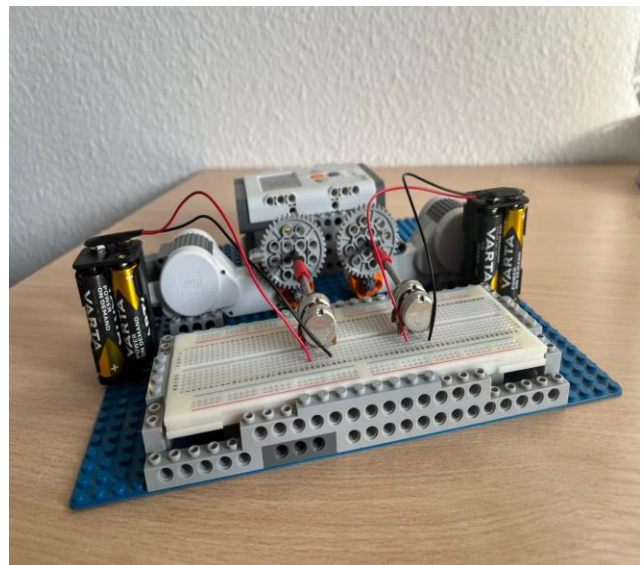


Abbildung 1 Aufbau des Oszilloskop-Plotters

Abbildung 1 zeigt, wie der Hardware-Teil des Projekts aufgebaut und miteinander verbunden ist.

III. KONSTRUKTION UND REALISIERUNG

Um dieses Projekt zu realisieren, werden einige mechanische Teile benötigt, die aus der LEGO-Mindstorms-Box stammten. Die verwendeten Hardwarekomponenten umfassten verschiedene Zahnräder, zwei Motoren und andere Teile zur Verbindung aller Komponenten. Zusätzlich werden auch einige externe Komponenten wie ein Breadboard, Batterien und ein Potentiometer bezogen. Außerdem wird ein Oszilloskop um die Zeichnung darzustellen benutzt.

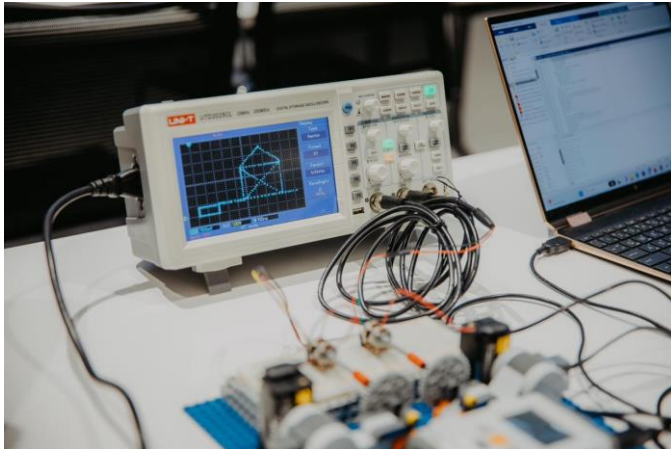


Abbildung 2 Vom Oszilloskop-Plotter gezeichnetes "Haus des Nikolaus"

A. Zahnräder

Ein 8-Zahn-Zahnrad für jeden Motor und ein 40-Zahn-Zahnrad für jeden Potentiometer wurden verwendet. Dies reduzierte die Rotation. Dadurch verringert sich das Übersetzungsverhältnis, da das Potentiometer nicht eine komplette Drehung von 360 Grad hat. Das Oszilloskop wurde verwendet, um die Spannungsänderung in jedem Potentiometer zu messen.

B. Motoren

Die NXT-Motoren aus dem LEGO-Mindstorms-Set wird verwendet. Diese Motoren sind speziell für LEGO-Roboter mit 9 Volt-Gleichstrommaschine entwickelt und bieten eine gute Leistung und Präzision [2].

C. Oszilloskop

Das Oszilloskop wurde verwendet, um die Spannungsänderung in jedem Potentiometer zu messen. ein Potentiometer ist an Kanal 1 und das andere ist an Kanal 2 angeschlossen.

Die Einstellungen des Oszilloskops wurden so konfiguriert, dass Kanal 1 auf der x-Achse und Kanal 2 auf der y-Achse dargestellt werden.

D. Programmierung

Um das Projekt zu programmieren, wurde MATLAB mit der RWTH-Toolbox verwendet. Die RWTH Toolbox stellt verschiedene Befehle zur Verfügung [3]. Mit diesen werden die Sensoren und Motoren in Betrieb genommen. Das Programm wird in MATLAB geschrieben und an den programmierbaren Computer NXT gesendet. Zuerst wurde das

Projekt gebaut und alle Komponenten wurden fixiert. Wie in der Abbildung 2 dargestellt, dann wurde das Programm geschrieben. Im Programm wurde zunächst ein Code geschrieben, um den Potentiometer in den Anfangszustand zu setzen, da der Potentiometer keine freie Rotation hat. Der Code wurde als Anweisung an jeden Motor geschrieben, sich um einen bestimmten Winkel zu drehen, bis der gemessene Startpunkt erreicht ist. Dann wurde eine weitere Funktion geschrieben, um beide Potentiometer in einen bestimmten Punkt, genannt Startzustand im Oszilloskop, zu bewegen. Dann wurden beide Funktionen dem Hauptcode hinzugefügt, der den Code für die zu zeichnende Figur enthält. Die zu zeichnende Figur wird als Grafikfunktion in MATLAB geschrieben, wobei ein Motor den Befehl erhält, sich zu drehen, um einen bestimmten Winkel zu zeichnen, und sich zusammen dreht, um diagonale Linien zu zeichnen.

IV. PROBLEME UND ERGEBNISDISKUSSION

Während der zwei Wochen der Arbeit an diesem Projekt traten viele Probleme auf, aber das schwierigste Problem bestand darin, die Motoren so zu programmieren, dass sie diagonale Linien zeichnen konnten. Beide Motoren mussten gleichzeitig und mit gleicher Geschwindigkeit rotieren, was sehr komplex war, um sie richtig einzustellen und zu konfigurieren, um perfekte diagonale Linien zu erhalten. und der Grund dafür war, dass die Motoren mit einem externen Bauteil an den Potentiometern befestigt waren und es eine kleine Toleranz gab, so dass der Motor Millisekunden vor dem Potentiometer läuft. Nach vielen Versuchen und harter Arbeit wurde das Problem jedoch gelöst, und es gelang, perfekte Diagonalen zu erzeugen. Ursprünglich sollte das Projekt dazu dienen, ein Nikolaushaus zu zeichnen, aber am Ende gelang es, viele verschiedene Figuren und Kreise zu zeichnen. Wie in der Abbildung 3 und 4 dargestellt, was mit der Hand manuell sehr schwierig wäre.

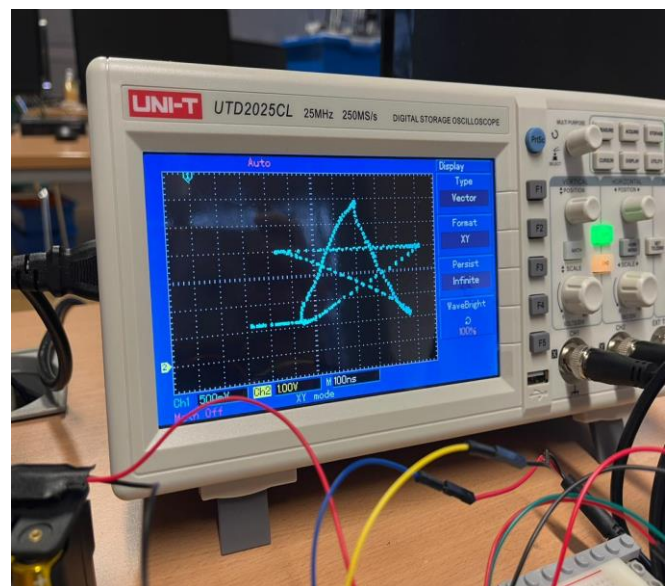


Abbildung 3 Zeichnung eines Sterns

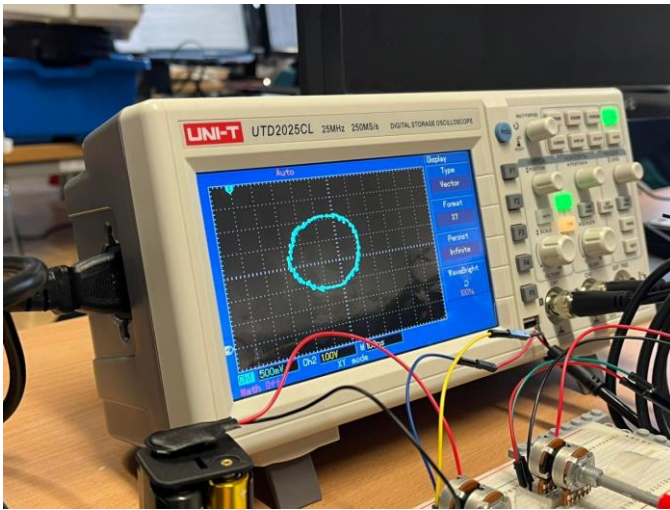


Abbildung 4 Zeichnung eines Kreis

V. ZUSAMMENFASSUNG

Zusammenfassend kann festgestellt werden, dass die Ziel des Projekts zufriedenstellend erreicht wurde. Hätte das Projektseminar länger als zwei Wochen gedauert wäre es möglich eine größere Version dieses Projekts gebaut und komplexere Zeichnungen kodiert hätte. Abschließend kann man sagen, dass man in diesem Projekt viel über MATLAB Programmierung und Spannungen lernt. Außerdem ist zu erwähnen, dass das LEGO-Mindstorm-Projektseminar eine gute Möglichkeit ist, Studierende mit der Programmierung in MATLAB vertraut zu machen.

LITERATURVERZEICHNIS

- [1] LEGO® 9V Technic Motors compared characteristics.
<https://www.philohome.com/motors/motorcomp.htm>
- [2] WIKIPEDIA, Steckplatine (<https://de.wikipedia.org/wiki/Steckplatine>)
Version: Januar 2024
- [3] RWTH -Mindstorms NXT Toolbox
(<https://www.mathworks.com/matlabcentral/fileexchange/18646-rwth-mindstorms-nxt-toolbox>) Version: Februar 2023

ANHANG

So wurde der in Abbildung 2 dargestellte Prozess in MATLAB programmiert:

```
clear all
handle=COM_OpenNXT();
COM_SetDefaultNXT(handle);
% program

OBJA=NXTMotor(MOTOR_A);
OBJB=NXTMotor(MOTOR_B);

% Motoren auf Anfang
disp('Motor A auf Anfang:');
anfangswertA=motor_an_anfang(OBJA,-10);
endwertA=1500+anfangswertA;
disp('Motor B auf Anfang:');
anfangswertB=motor_an_anfang(OBJB,10);
```

```
endwertB=anfangswertB-1460;
```

```
%load('endwerte.mat')
```

```
% Endwerte etwas nach "innen" verschieben
```

```
anfangformA=anfangswertA+900;
```

```
anfangformB=anfangswertB-900;
```

```
endformA=endwertA-50;
```

```
endformB=endwertB+50;
```

```
% % Haus des Nikolaus
```

```
form_x=[0,3,3,5,7,7,3,7,3,10];
```

```
form_y=[0,0,3,4.5,3,0,3,3,0,0];
```

```
% x-Werte für Motor B umrechnen
```

```
m=(endformB-anfangformB)/12;
```

```
n=anfangformB;
```

```
motorB_pos=m*form_x+n;
```

```
% y-Werte für Motor A umrechnen
```

```
m=(endformA-anfangformA)/6;
```

```
n=anfangformA;
```

```
motorA_pos=m*form_y+n;
```

```
figure(2)
```

```
plot(motorB_pos,motorA_pos)
```

```
% Motorpower
```

```
power=60;
```

```
OBJA.SpeedRegulation=1;
```

```
OBJB.SpeedRegulation=1;
```

```
OB-
```

```
JA=motor_an_position(OBJA,anfangformA,power);
```

```
OBJA.SendToNXT;OBJA.WaitFor;
```

```
OBJB=motor_an_position(OBJB,anfangformB,power
```

```
);OBJB.SendToNXT;OBJB.WaitFor;
```

```
% Schleife über alle Punkte
```

```
for n=1:length(form_x)
```

```
    tacho_limitA=motorA_pos(n)-
```

```
OBJA.ReadFromNXT.Position;
```

```
    tacho_limitB=motorB_pos(n)-
```

```
OBJB.ReadFromNXT.Position;
```

```
    % Power berechnen und auf 100 begrenzen
```

```
powerA=max([5,min([100,round(156.5902/149.523
```

```
6*abs(tacho_limitA/tacho_limitB)*power))]);
```

```
    pow-
```

```
erB=max([5,min([100,round(149.5236/156.5902*a
```

```
bs(tacho_limitB/tacho_limitA)*power))]);
```

```
    disp(['Power A: ',num2str(powerA)])
```

```
    disp(['Power B: ',num2str(powerB)])
```

```
    if powerA>powerB
```

```
        powerA=power;
```

```
    else
```

```
        powerB=power;
```

```
    end
```