

# Entwicklung eines Low-Budget Drehgebers

Raphael Hafner

## Abstract

Um bei Winkelmessungen zu geringen Kosten nicht in der Anzahl der Umdrehung beschränkt zu sein, wird eine Neuentwicklung eines Low-Budget Drehgebers in diesem Projekt verfolgt. Die Anforderungen sind hierbei Herstellkosten im einstelligen Bereich, eine Auflösung von ca.  $0,1^\circ$  und der Drehgeber soll auf Arduino-Basis betrieben werden. Ziel dieser Arbeit ist es mit Hilfe von Simulationen die Grenzen des Systems festzulegen, mit diesen Erkenntnissen einen Prototyp zu erstellen und diesen anschließend mit einem Messgerät zu validieren.

In order not to be limited in the number of revolutions for angle measurements at low costs, a new development of a low-budget rotary encoder is being pursued in this project. The desired goals are manufacturing costs in the single-digit range, a resolution of approx.  $0.1^\circ$  and the rotary encoder should be operated on an Arduino basis. The aim of this work is to define the limits of the system with the help of simulations and to create a prototype with this knowledge and then to validate it with a measuring device.

## I. EINLEITUNG

Bisher werden zur absoluten Winkelmessung zum Großteil Potentiometer verwendet, welche allerdings einen entscheidenden Nachteil haben. Es handelt sich um die endliche Anzahl der mechanischen Umdrehungen. Herkömmliche Potentiometer kosten nur wenige Euro, bieten allerdings maximal eine gesamte Umdrehung (häufig nur  $300^\circ$ ). Es gibt Potentiometer, welche bis zu 10 Umdrehungen in eine Richtung gewährleisten, befinden sich aber preislich im 2-stelligen Bereich und müssen nach den möglichen 10 Umdrehungen wiederum in die entgegengesetzte Richtung gedreht werden.

Die Entwicklung eines neuen Low-Budget Drehgeber soll dies bezüglich Abhilfe schaffen. Folgende Ziele sollen hierfür erreicht werden;

Die Herstellkosten sollen sich auf einen Preis von ca. 2-3€ belaufen bei einer Auflösung von absoluten Winkeln von ca.  $0,1^\circ$ . Um ein breites Feld der Nutzbarkeit zu bedienen, soll der Drehgeber auf Arduino-Basis betrieben sein.

## II. VORRAUSSETZUNGEN

Um einen passenden Aufbau eines solchen Drehgeber zu entwickeln, müssen die drei grundlegende Komponenten eines Winkelencoders bestimmt werden.

Zuerst der Sender, d.h. jene Komponente, welche Energie in Form eines Signals in das System bringt. Um eine Winkelposition aus dem Signal zu erzeugen, wird ein Modulator benötigt, der das Signal des Senders so beeinflusst, dass zu jedem Zeitpunkt eine eindeutige Winkelstellung entsteht. Der Empfänger dient abschließend der Auswertung des modulierten Signals [1].

Diese drei Komponenten können mit Hilfe verschiedener sensorischer Prinzipien zusammenarbeiten, z.B.: optisch, potentiometrisch, magnetisch, etc [1].

Da das magnetische Prinzip für die gegebenen Anforderung die größte Erfolgchance (vor allem in Bezug auf Einhalten des Budgets und Machbarkeit der geforderten Auflösung) verspricht, wird der Drehgeber hiermit entwickelt.

## III. THEORETISCHER AUFBAU

Für ein magnetisches Abtastprinzip wird als Sender in jedem Fall ein Dauermagnet benötigt. Dieser kann entweder auf dessen magnetisches Feld  $H$  [A/m] oder magnetischen Flussdichte  $B$  [T] untersucht werden. Da die Sensoren, welche in Frage kommen hauptsächlich mit der magnetischen Flussdichte arbeiten, wird diese in Form von Simulationen näher betrachtet. Mit Hilfe der Simulationssoftware FEMM (FEMM 4.2) werden verschieden Magnetarten untersucht.

Die Simulationen in Bezug auf die gesetzten Ziele ergeben, dass ein zylindrischer, diametral-magnetisierter Dauermagnet am besten geeignet ist. Dieser kann auf das Ende einer drehbar-gelagerten Achse montiert und der Sensor radial dazu in einem feststehenden Gehäuse positioniert werden.

Die Modulation des Sendersignals erfolgt durch die Rotation des Magnets und somit der magnetischen Flussdichte, welche vom Sensor aufgenommen und verarbeitet wird.

Als Sensor (Empfänger) kommen eine Reihe verschiedener Magnetsensoren in Betracht (AMR-, GMR-, Hall-Sensor). Aufgrund der hohen möglichen Auflösung und der geringsten Kosten, werden für den neuen Drehgeber Hall-Sensoren eingesetzt [1,3].

Hall-Sensoren arbeiten mit dem Hall-Effekt. Dieser tritt auf, wenn sich ein stromdurchflossener Leiter in einem Magnetfeld befindet. Auf Grund der dadurch induzierten Lorentzkraft entsteht eine Spannung senkrecht zur fließenden Stromrichtung im Leiter und dem Magnetfeld. Diese sogenannte Hall-Spannung kann gemessen werden und liegt je nach Betriebsart des Sensors bei 0-3,3V bzw. 0-5V. Der Betrag dieser Hall-Spannung ist direkt an die Ausrichtung des Magnetfeldes gekoppelt. Um einen vollständigen Polwechsel aufzeichnen zu können, muss der benötigte Sensor bipolar aufgebaut sein, das heißt der Sensor orientiert sich nicht nur an einen sondern an beiden Polen. Daraus ergibt sich, dass bei einer Hall-Spannung von 0V der Nordpol direkt auf den Sensor gerichtet (z.B.:  $B = -100\text{mT}$ ) ist und bei 5V der Südpol direkt auf den Sensor zeigt ( $B = +100\text{mT}$ ) [3].

Durch Simulationen verschiedener erhältlicher Dauermagneten und dem Abgleich verschiedener erhältlicher Hall-Sensoren kommt folgende Kombination zum Einsatz.

Zum einen ein diametral-magnetisierter NdFeB-Dauermagnet N38 (der Firma ChengYang) mit einem Durchmesser von 5mm und einer Tiefe von 2mm. Zum anderen ein linearer bipolarer Hall-Sensor des Herstellers Texas Instruments (DRV5055A3QLPG), welcher mit 3,3V oder 5V betrieben werden kann und eine Magnetflussdichte B von -85mT bis +85mT als Extremwerte auswerten kann [4].

Die Simulation des NdFeB-Magneten ergab ein Abstand von 6,65mm zum Mittelpunkt des Sensors, bei welchem die Extremwerte von ca.  $B = \pm 85\text{mT}$  auftreten. Somit sollte der Abstand des definierten Messpunkts des Sensors zum Mittelpunkt des Magneten, so exakt wie möglich, diesen Abstand haben. In Abbildung 1 ist die Simulation hierfür zu sehen. Der Mittelpunkt des Magneten liegt bei P(0/2,5).

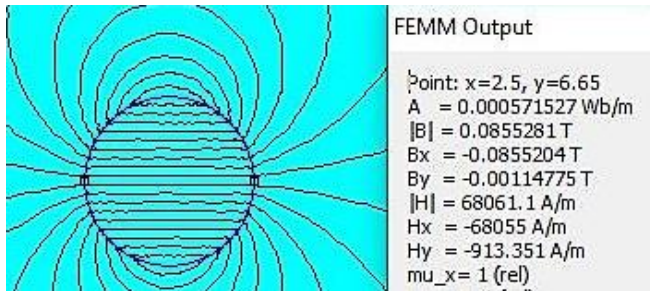


Abbildung 1: Simulation des NdFeB-Magneten mit FEMM 4.2

Zur Auswertung der Signale soll wie bereits erwähnt ein Arduino verwendet werden. Da das Arduino Uno das am weitesten verbreitetsten Modell ist, wird in diesem Projekt ein Board auf dieser Basis verwendet (Elegoo Uno R3).

Der Hall-Sensor wird mit einem analogen Eingang des Arduinos verbunden und übergibt die anliegende Hall-Spannung. Im Arduino wird das Signal (0-5V) über einen A/D-Wandler in Werte von 0-1023 ( $2^{10}$  Werte  $\pm 10$  Bit) konvertiert [2,5]. Somit ergibt sich eine maximal erreichbare Auflösung von  $\frac{180^\circ (\text{Nordpol-Südpol})}{1024 \text{ Werte}} = 0,176^\circ$ .

Zum Vergleich; für herkömmliche Potentiometer stehen Arduino-seitig ebenfalls 10 Bit zur Verfügung, allerdings bezogen auf (maximal)  $360^\circ$ , daraus folgt eine halb so große Auflösung von  $0,352^\circ$ .

Wie im Datenblatt des Sensors beschrieben ist, gibt ein Sensor lediglich über  $180^\circ$  eindeutige Digitalwerte aus. Somit ist es nicht möglich anhand eines einzelnen Werts die absolute Winkelposition zu bestimmen. (Bsp.: Bei Sensor A besitzt ein Winkel von  $60^\circ$  denselben Digitalwert wie bei einem von  $300^\circ$ ). Auf Grund dessen ist es notwendig mehreren Sensoren einzusetzen.

Um potenzielle Messfehler zu umgehen, werden drei Sensoren verwendet und in einem Winkel von  $120^\circ$  zueinander um den Magneten positioniert. Dieser Aufbau bietet den Vorteil, dass in den Extremstellungen, das heißt wenn ein Pol mittig zwischen zwei Sensoren steht (beide Sensoren liefern dieselbe Flussdichte), dann ist der gegenüberliegende Pol des Magneten genau auf den dritten Sensor ausgerichtet (maximale od. minimale Flussdichte).

In Abbildung 2 ist ein Schnitt des Aufbaus des Prototyps des Drehgebers zusehen. Die Achse, auf welcher stirnseitig der Magnet montiert ist, ist drehbar über ein Kugellager im feststehenden Gehäuse gelagert, welches die drei Sensoren zum Magneten positioniert. Sowohl Achse und Gehäuse werden mit einem 3D-Drucker erstellt.

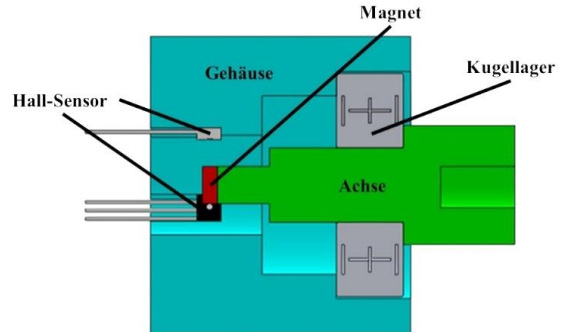


Abbildung 2: Schnittdarstellung des Drehgeberaufbau

#### IV. VERARBEITUNG DER MESSWERTE

Die erwarteten Messwerte der drei Sensoren werden nun in Abhängigkeit zueinander betrachtet, um eine Winkelstellung ausgeben zu können.

Ein Sensor, welcher direkt auf den Südpol des Magneten gerichtet ist sendet die maximale Hall-Spannung von 5V an das Arduino und wird in den maximalen digitalen Zahlenwert 1023 konvertiert. Bei einer Umkehrung des Magnetfelds, das heißt einer  $180^\circ$  Drehung des Magneten fällt die Hall-Spannung auf ihr Minimum von 0V und dem zugehörigen digital Wert von 0. Somit ergeben sich für eine komplette Umdrehung des Magneten 2046 Digitalwerte ( $2 \cdot 1024 = 2048$ , abzüglich der doppelten Werte für  $180^\circ$  und  $360^\circ = 2046$ ). Die Auflösung beträgt demnach über eine komplette Umdrehung:  $\frac{360^\circ}{2046} = 0,176^\circ$ .

Da die Sensoren um  $120^\circ$  zueinander versetzt sind, ergibt sich für ein Bereich von  $0^\circ$ - $360^\circ$  folgendes Diagramm (Abb.2).

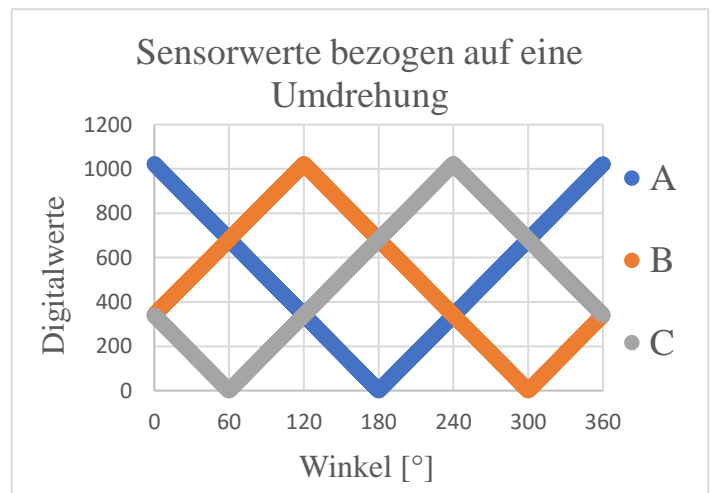


Abbildung 3: Sensorwerte bezogen auf eine Umdrehung

Wie in Abbildung 3 zu sehen ist, bestehen sechs Winkelbereiche, in welchen sich die 3 Sensor-Digitalwerte eindeutig zueinander verhalten. Im ersten Bereich von 0-60° ist das Signal  $A > B$  &  $B > C$ , dies ändert sich im

darauffolgenden Bereich von 60°-120° zu  $A < B$  &  $A > C$ . Jede dieser Abhängigkeitsbeziehungen kommt einmal vor vgl. Tabelle 1.

Winkelbereich [°]	Abhängigkeitsbeziehung
0-60	$A > B$ & $B > C$
60-120	$A < B$ & $A > C$
12-180	$A < C$ & $B > C$
180-240	$A < B$ & $B < C$
240-300	$A > B$ & $A < C$
300-360	$A > C$ & $B < C$

*Tabelle 1: Winkelbereiche mit zugehörigen Abhängigkeitsbeziehungen*

Die Sensorwerte werden ausgelesen und mit Hilfe der Abhängigkeitsbeziehungen kann ein eindeutiger Winkelbereich zugeordnet werden.

Eine Möglichkeit die absoluten Winkel zu bestimmen ist, dass der Graph des Sensors A mit zwei linearen Funktionen beschrieben wird und je nachdem welche Abhängigkeitsbeziehung der 3 Sensorwerte eintritt, der Wert des Sensors A in einer der beiden Funktionen verrechnet wird. Die fallende Flanke, das heißt der Bereich von 0°-180°, kann mit der Funktion  $A1 = \frac{-1023}{180} * x + 1023$  beschrieben werden und die steigende Flanke (180°-360°) mit  $A2 = \frac{1023}{180} * x - 1023$ .

Da der Winkel die x-Komponente dieser Funktionen ist, muss nach x aufgelöst werden.

$$x = (A1 - 1023) * \frac{-180}{1023} \text{ bzw. } x = (A2 + 1023) * \frac{180}{1023}$$

Eine weitere Möglichkeit zur Winkelbestimmung, ist das Hinterlegen der möglichen Werte der drei Sensoren in einer 4-spaltigen Tabelle. In der ersten Spalte steht der Winkel und in den weiteren die Digitalwerte der Sensoren. Jeder Winkel wird hierbei durch eine einzigartige Kombination der drei Digitalwerte dargestellt. Zum Beispiel gilt für den Winkel 180° = 0 (A), 682 (B), 682 (C).

Diese Möglichkeit benötigt neben einer höheren Speicherkapazität (zusätzliches Speichermedium wie SD-Karte etc. notwendig) auch eine erhöhte Rechenleistung, damit die Wertepaarungen und schnellstmöglich einem Winkel zugeordnet werden kann [2]. Auf Grund dessen wird vorerst die erste beschriebene Möglichkeit umgesetzt.

## V. PRAKTISCHER AUFBAU

Der praktische Aufbau des Drehgebers bezieht sich auf den ersten Prototyp, um darzustellen, ob dieser Aufbau die gestellten Anforderungen erreichen kann.

### *Elektronik:*

Wie im Datenblatt des Sensor-Herstellers beschrieben, werden die drei Pins entsprechend mit dem Arduino verbunden. Dies sind zum einen die analogen Ausgänge, welcher für die Sensoren A, B, C mit den Analog In-Ports A0, A1, A2 des Arduinos verbunden werden. Die Vcc-Pins und die GND-Pins der Sensoren werden mit Hilfe eines Steckbretts parallel verbunden, das heißt alle Vcc-Pins in einer isolierten Reihe und alle GND-Pins in einer anderen. Die Vcc-Reihe wird dem 5V Anschluss des Arduinos verbunden und die GND-Reihe mit GND des Arduinos [2,5].

### *Mechanischer Aufbau:*

Der Mechanische Aufbau entspricht dem theoretischen Aufbau aus Abbildung 2. Der Außendurchmesser des Gehäuses beträgt 40mm und eine Länge von 35mm. Das Gehäuse ist hohl und beidseitig geöffnet. An einem Ende können die drei Sensoren im 120°-Winkel zueinander über Nuten in das Gehäuse gesteckt werden. Am anderen Ende wird die Achse mit aufgedrehtem Rillenkugellager (9x30x10) eingeführt. Die Achse ist so am Kugellager geschultert, dass der stirnseitig aufgeklebte Magnet exakt auf Höhe der Messpunkte der Sensoren sitzt, wenn die Achse im Gehäuse montiert ist.

### *Codierung:*

Für das Auslesen analoger Werte enthält die Arduino-Bibliothek die Funktion `analogRead()`, welche in diesem Fall eine Spannung von 0-5V in Werte von 0-1023 kompiliert [7]. Diese Werte werden wie zuvor beschrieben auf ihre gegenseitige Abhängigkeit untersucht und anschließend der Wert von Sensor A in die entsprechende Funktion eingesetzt. Die Ausgabe des Winkels erfolgt über den seriellen Monitor.

Der Code kann folgendermaßen aussehen:

```
int Sensor1 = A0; //Pin A0 wird gelesen
int Sensor2 = A1; //Pin A1 wird gelesen
int Sensor3 = A2; //Pin A2 wird gelesen
int A; int B; int C; float phi;
void setup() {Serial.begin(9600);}
void loop() {
  A = analogRead(Sensor1);
  B = analogRead(Sensor2);
  C = analogRead(Sensor3);
  Serial.print("Der aktuelle Winkel beträgt:");
  if (A>=B && B>=C) //0-60°
  { phi = ((float)(A-1023.0)*(-180.0/1023.0));
    Serial.print(phi);}
  else if (A<B && A>=C) //60.2-120°
  { phi = ((float)(A-1023.0)*(-180.0/1023.0));
```

```

Serial.print(phi);}
else if (A<C && B>=C) //120.2-180°
{ phi = ((float)(A-1023.0)*(-180.0/1023.0));
Serial.print(phi);}
else //180.2-360°
{ phi = ((float)(A+1023.0)*(180.0/1023.0));
Serial.print(phi);}
Serial.println("°.");}

```

VI. AUSWERTUNG DER ERGEBNISSE

Die ersten Ergebnisse, das heißt die errechneten Winkel ausgegeben im seriellen Monitor, beginnen mit ca. 70° und steigen beim Drehen kontinuierlich bis ca. 120°. Darauf folgt ein Sprung auf 240° und ein wiederum kontinuierliches Verhalten bis 290°, worauf in Sprung auf den Anfang von 70° folgt.

Um herauszufinden, weshalb diese Sprünge auftreten werden die einzelnen digitalen Werte der Sensoren betrachtet. Hierfür werden die Ergebnisse der Funktion analogRead() für die drei Sensoren nebeneinander stehend im seriellen Monitor ausgegeben. Die minimalen und maximalen Werte über eine ganze Umdrehung sind in Tabelle 2 für die jeweiligen Sensoren angegeben.

	Min	Max
A	352	622
B	354	650
C	361	620

Tabelle 2: min. und max. Werte der Sensoren des Prototyps

Es ist zu erkennen, dass keiner der Sensoren die erwarteten Minima und Maxima von 0 bzw. 1023 ausgeben. Dies hängt mit einem zu großen Abstand von Sensor zu Magnet zusammen, demnach befindet sich der Sensor im Prototyp nicht im optimalen Arbeitspunkt. Dies konnte durch Verringerung des Abstands per Hand bewiesen werden. Der Arbeitsbereich von ±85mT (Herstellerangaben) ist demnach nicht getroffen.

Da davon auszugehen ist, dass die Angaben im Datenblatt des Herstellers stimmen, muss davon ausgegangen werden, dass die magnetischen Eigenschaften des Magnets nicht mit denen in der Simulation berechneten übereinstimmen und somit nur ein kleiner Abstand zugelassen werden darf.

Darüber hinaus sollten die minimalen und maximalen Werte der drei Sensoren gleich groß sein, aber auch hier liegen einige Differenzen vor, was wiederum auf die Genauigkeit der Positionierung der Sensoren zurückzuführen ist.

Um mit den definierten Maßen weiterzuarbeiten, muss die Software angepasst werden. Die Arduino-Bibliothek bietet

diesbezüglich die map()-Funktion, mit welcher es möglich ist, gegebene Wertebereich auf einen beliebigen Wertebereiche abzubilden [8]. Beispielsweise kann der Wertebereich des Sensors A (352 – 622) auf einen Wertebereich von 8 Bit, also 256 Werte abgebildet werden. Dieser Bereich wurde gewählt, da bei allen Sensoren die Wertebereiche mindestens 256 Werte beinhalten.

```
A = map(A, 352, 622, 0, 255);
```

Da jeder Wert außer 0 und 255 innerhalb einer Umdrehung doppelt vorkommt ergibt sich ein gesamter Wertebereich von 510 Werten (vgl. 9 Bit = 512 Werte). Die Auflösung liegt bei  $\frac{360}{510} = 0,706^\circ$ .

Die Auflösung wird zwar hierbei erheblich verschlechtert, allerdings muss davon ausgegangen werden, dass selbst wenn die Sensoren optimal zueinander und zum Magneten positioniert sind, immer noch Abweichungen zu den theoretischen Werten auftreten können. Die map()-Funktion wird auch in diesem Fall das Ergebnis verbessern.

Es ist zudem nicht nötig alle drei Sensorwerte mit der map()-Funktion zu bearbeiten. Am zuverlässigsten ist es, die analogen Werte auszulesen und dann nach der Bestimmung des Winkelbereichs durch die Abhängigkeitsbeziehungen (if A>B && B>C...) jeweils den Wert von A anzupassen (A = map(A, 352, 622, 0, 255);) und mit diesem weiter zurechnen.

Der Wertebereich liegt nun nur noch zwischen 0 und 255, weshalb die auf gestellten Funktionen dementsprechend abgeändert werden müssen.

$$x = (A1 - 255) * \frac{-180}{255} \text{ bzw. } x = (A2 + 255) * \frac{180}{255}$$

Das Ergebnis ist ein kontinuierliche Winkelwiedergabe über die gesamte Umdrehung, ohne Sprünge oder Unregelmäßigkeiten. Die Winkel werden in den erwarteten Schritten von ca. 0,71° ausgegeben.

VII. VALIDIERUNG DER ERGEBNISSE

Um die Richtigkeit der Ergebnisse zu überprüfen, wird ein herkömmliches Potentiometer integriert. Hierfür befindet sich eine Bohrung in der Achse, in welche das Potentiometer gesteckt wird. Ein Potentiometer besitzt 3 Pins; GND, Vcc, und einen analogen Ausgang. Die Verkabelung erfolgt analog zu den eingesetzten Hall-Sensoren. Der analoge Ausgang des Potentiometers ist hierbei mit dem analogen Eingang A4 des Arduinos zu verbinden. Laut Datenblatt des Potentiometers wird eine Drehung von 300° gewährleistet [6].

Durch das Deklarieren eines Integer Poti, eines Integer Potiwert und einer float-Variable Potiwinkel und ergänzen folgender Zeilen Code (innerhalb der Schleife);

```

Potiwert = analogRead(Poti); //Auslesen des Potis (Werte von 0-1023)
Potiwinkel=((float)Potiwert/1023.0)*300.0;
//Umrechnung in einen Winkel
Serial.print("Poti:"); Serial.print(Potiwinkel);

```



kann der Winkel welcher mit dem neuen Drehgeber gemessen wird, mit dem vom Potentiometer gemessenen Winkel verglichen werden.

Das Ergebnis ist eine parallele Winkelveränderung der beiden Messgeräte, das heißt, wenn das eingesteckte Poti um ca. 15° gedreht wird, liegt die Differenz der Winkelwerte des Drehgebers bei ebenfalls ca. 15°. Somit wurde bewiesen, dass mit diesem Aufbau exakte Winkel gemessen werden können. Allerdings sind die Zuverlässigkeit und Reproduzierbarkeit dieses Systems ungenügend, was vor allem auf die ausbaufähige Positionierung der Sensoren zurückzuführen ist.

## VIII. DISKUSION & AUSBLICK

Ziel dieser Arbeit ist war es einen Low-Budget Drehgeber zu entwickeln. Es ist gelungen einen Winkel zu messen, auch wenn bezüglich der Auflösung noch nicht die gewünschte Performance erreicht werden konnte. Demnach sind technisch ein Großteil der gewünschten Ziele verfolgt worden und nun gilt es zusammenzustellen, was ein Drehgeber gekostet hat.

Der Einkaufspreis eines Sensors liegt bei einer Stückzahl von 40stk bei 1,35€. Bei drei benötigten Sensoren errechnet sich ein Wert von 4,05€.

Ein Magnet bei einer Stückzahl von 50stk kostet ca. 0,2€.

Das Kugellager ist nicht neu eingekauft worden, sondern aus dem Hochschulbestand, vergleichbare Modelle kosten neu ca. 1€.

Da die restlichen Komponenten entweder mit dem 3D-Drucker erstellt worden sind oder wie das Arduino nicht mit einkalkuliert werden setzt sich eine Gesamtsumme von ca. 5,5€ zusammen.

Letztendlich ist demnach auch der Preis für einen Drehgeber höher als gewünscht. Allerdings ist mit diesem Drehgeber ein Messbereich von beliebig vielen Umdrehungen, bei einer annehmbaren Auflösung, möglich. Bei einer Auslegung für eine höhere Stückzahl wird sich der Herstellpreis zudem deutlich verringern, da sowohl Sensoren wie auch die Magneten bei größeren Bestellungen kostengünstiger werden.

Um mehr Performance zu erlangen, sollten folgende Aspekte verbessert werden;

Der Abstand von Sensoren zu Magnet muss neu definiert werden. Da die Simulationsergebnisse nicht der Realität entsprechen muss händisch der Abstand verringert werden, um eine einen höheren Flussdichtenbereich aufnehmen zu können. Aus Tests ohne Messmittel ging hervor, dass der Abstand mindestens halbiert werden muss. Bei bekanntem Abstand wäre die optimale Lösung hierfür eine im Gehäuse positionierbare Platine, welche präzise mit den Sensoren bestückt werden kann.

Zur weiteren Verbesserung der Auflösung kann statt einem Arduino-Uno das Modell Due eingesetzt werden. Dieses betreibt die Hall-Sensoren mit 0V-3,3V und wandelt die analogen Werte in 2048 Digitalwerte. Demnach ist hiermit eine maximale Auflösung von ca. 12 Bit ( $\frac{360}{4096} = 0,088^\circ$ ) möglich [7].

Um den Drehgeber noch vielseitiger zu gestalten, könnte auch ein Sensor dazu verwendet werden Umdrehungen zu Zählen und eventuell auch bezogen auf die Zeit für eine Umdrehung die aktuelle Geschwindigkeit wiedergeben.

## IX. FAZIT

In diesem Projekt wurde versucht mit möglichst geringem Kostenaufwand einen absolut Drehgeber mit hoher Auflösung zu entwickeln. In der Theorie ist mit dem gewählten Aufbau eine Auflösung von bis zu 11 Bit möglich, was allerdings praktisch nicht erreicht werden konnte. Die momentan mögliche Auflösung beträgt ca. 9 Bit bei Herstellkosten von insgesamt unter 6€.

Zusammenfassend zeigt dieses Projekt, dass es möglich ist einen Drehgeber zu geringen Kosten zu produzieren, welcher die herkömmlichen Messgeräte wie Potentiometer in diesem Segment, vor allem durch seine endlose Lagerung, ersetzen kann.

## X. QUELLEN

- [1] Stefan Balsler, „Encoder und Motor-Feedback-Systeme“, Springer Vieweg, 2016
- [2] Thomas Brühlmann, „Sensoren im Einsatz mit Arduino“, mitp Verlag, 2017
- [3] A. Grosz, M. J. Haji-Sheikh, S. C. Mukhopadhyay, „High sensitivity magnetometers“, Springer, 2017
- [4] [https://www.ti.com/lit/ds/symlink/drv5055.pdf?ts=1630398678093&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDRV5055%253FkeyMatc h%253DDRV5055%2526tisearch%253Dsearch-everything%2526usecase%253DGPN](https://www.ti.com/lit/ds/symlink/drv5055.pdf?ts=1630398678093&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDRV5055%253FkeyMatc h%253DDRV5055%2526tisearch%253Dsearch-everything%2526usecase%253DGPN) (30.08.2021)
- [5] <https://makeabilitylab.github.io/physcomp/sensors/hall-effect.html> (30.08.2021)
- [6] <https://asset.conrad.com/media10/add/160267/c1/-/en/000445518DS01/datenblatt-445518-piher-pc16sh-10ip06472a2020mta-dreh-potentiometer-mono-02-w-47-k-1-st.pdf> (25.08.2021)
- [7] <https://www.arduino.cc/reference/de/language/functions/analog-io/analogread/> (24.07.2021)
- [8] <https://www.arduino.cc/reference/de/language/functions/math/map/> (30.08.2021)