

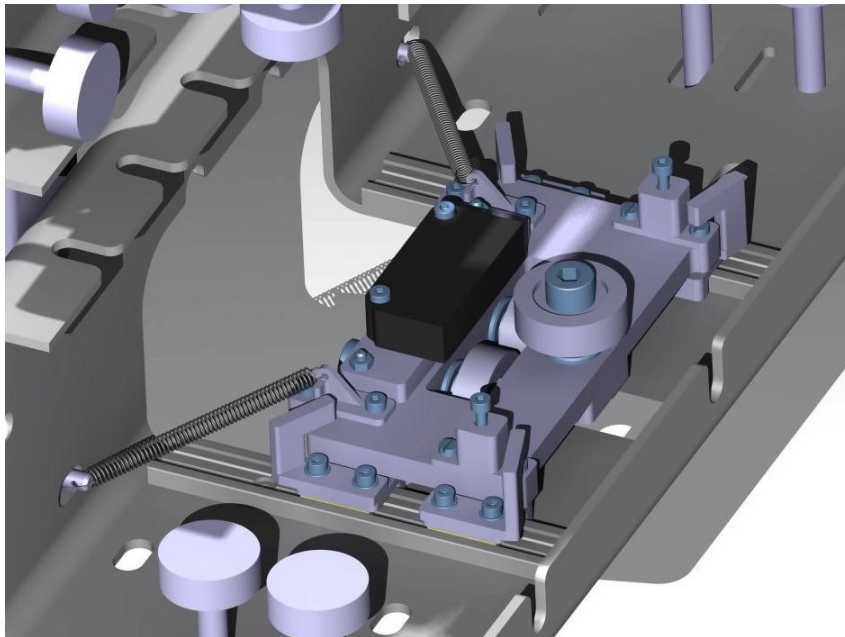
## Bachelorarbeit

# Design, Implementierung und Verifikation einer translatorischen sowie rotatorischen Positionserfassung von Komponenten einer robotischen Versuchsanlage mit magnetischen CANopen-Messgebern

RT-BA 2019/07

Autor:

Alexander Ferstl



Betreuer:

Dipl.-Ing. Martin Dziura  
 Lehrstuhl für Raumfahrttechnik  
 Technische Universität München

## Zusammenfassung

Die Zunahme der Anzahl der Schrottteile in den Erdborbits stellt eine große Herausforderung für die Raumfahrt dar. Das RACOON-Lab des Lehrstuhls für Raumfahrttechnik an der Technischen Universität München (TUM) dient der Simulation von Nahbereichsoperationen von Satelliten. Damit sollen neue Technologien entwickelt und bewertet werden, die Missionen zur Reduzierung des Weltraummülls ermöglichen.

Ziel dieser Arbeit ist die Implementation eines Positionserfassungssystems im Sonnen- und im Erdalbedosimulator des RACOON-Labs, um deren Positionierungen in der Anlage nachprüfbar und reproduzierbar zu gestalten. Dazu werden zwei translatorische sowie zwei rotatorische Hall-Sensoren genutzt. Die Datenausgabe soll in LabVIEW erfolgen. Konzeption, Design und Implementierung der dafür notwendigen Subsysteme findet auf Grundlage der sorgfältig erhobenen Anforderungen statt.

So wird aufgrund der komplizierten Schienengeometrie für die linearen Sensoren eine Vorrichtung zur Sensorpositionierung benötigt, welche diese im anforderungsgerechten Abstand zum absolutkodierte Magnetband hält. Dazu werden drei Varianten konstruiert, verglichen und schließlich eine ausgewählt. Diese ist größtenteils additiv fertigbar ist und wird nachfolgend in die Simulatoren integriert.

Da die Kodierer das Busprotokoll CANopen unterstützen, wird ein CAN-Busnetz implementiert und konfiguriert. Neben den Sensoren als Busteilnehmer wird eine Microcontroller-Schnittstellen-Lösung ausgewählt und an den Bus angeschlossen, um deren Kommunikation mit dem Steuerungs-PC zu ermöglichen. Die eingesetzten Komponenten sind ein Teensy USB Development Board und ein CAN-Transceiver, welche über ein Breadboard verbunden werden. Durch die Programmierung im Arduino-Framework erhält die Schnittstelle die gewünschte CANopen-Funktionalität. Dabei dient die Open-Source CAN-Bibliothek „FlexCAN“ als Grundlage.

Die Verifikation der Funktion des Positionserfassungssystems erfolgt über Testfahrten. Dabei werden Referenzpunkte auf der Schiene definiert und deren absolute Positionswerte dokumentiert. Am Ende dieser Arbeit ist eine Kommunikation mit den Sensoren über LabVIEW möglich.

Beim Testen des Drehencoders des Sonnensimulators fallen Ausgaben auf, welche von den realen Winkelpositionen abweichen. Die Behebung des Problems bedarf nachfolgender Maßnahmen auf Basis der dargelegten Ursachenanalyse.

## Abstract

The increase in the number of scrap parts in earth's orbits represents a major challenge for space travel. The RACOON-Lab of the Institute of Astronautics at the Technical University of Munich (TUM) is used for simulating close-range satellite operations. The aim is to develop and evaluate new technologies that will enable missions to reduce space debris.

The aim of this thesis is the implementation of a position determination system in the sun simulator unit and the earth's albedo simulator unit of the RACOON-Lab in order to make their positioning in the facility verifiable and reproducible. Two translatory and two rotary Hall sensors are used for this purpose. The data output is to be realized in LabVIEW. Conception, design and implementation of the necessary subsystems is based on the carefully defined requirements.

Due to the complicated rail geometry, a device for sensor positioning is required for the linear sensors, which holds them at the required distance from the absolutely coded magnetic tape. Three variants are designed, compared and finally one gets selected. It is mostly additive producible and subsequently integrated into the simulators.

Since the encoders support the CANopen serial bus protocol, a CAN bus network is implemented and configured. In addition to the sensors as bus participants, a microcontroller interface solution is selected and connected to the bus to enable their communication with the control PC. The components used are a Teensy USB Development Board and a CAN transceiver, which are connected via a breadboard. Programming in the Arduino framework gives the interface the desired CANopen functionality. The open source CAN library "FlexCAN" serves as the basis.

The verification of the functioning of the position determination system is done by test drives. Reference points on the rail are defined and their absolute position values are documented. At the end of this thesis, a communication with the sensors via LabVIEW is possible.

When testing the rotary encoder of the sun simulator, outputs are noticed which deviate from the real angular positions.

The elimination of the problem requires subsequent measures on the basis of the presented cause analysis.

## Inhaltsangabe

<b>1</b>	<b>EINLEITUNG</b>	<b>1</b>
1.1	Hintergrund und Motivation	1
1.2	Stand der Technik	2
1.2.1	RACOON-Labor	2
1.2.2	Grundlagen Weg- und Positionssensoren	4
1.2.3	CANopen-Systeme	7
1.3	Ziele dieser Arbeit	9
1.4	Abgrenzung	9
1.5	Vorgehensweise	10
<b>2</b>	<b>ANFORDERUNGSSYNTHESE</b>	<b>11</b>
2.1	CANopen-Spezifikationen	11
2.1.1	Überblick	11
2.1.2	Physikalische Schicht und Verbindungsschicht	11
2.1.3	Layer Setting Services (LSS)	11
2.1.4	Grundlegender Aufbau eines CANopen Geräts	12
2.1.5	Kommunikationsobjekte	12
2.1.6	Netzwerkmanagement	15
2.1.7	Profil für Encoder	16
2.2	Anforderungserhebung	16
<b>3</b>	<b>KONZEPTION UND DESIGN DES SYSTEMS</b>	<b>20</b>
3.1	Systemebene 0	20
3.1.1	Hardware-Systemarchitektur	20
3.1.2	CANopen-Netzwerk	22
3.2	Systemebene 1	23
3.2.1	Topologieanforderungen für die Magnetbandanbringung	23
3.2.2	Konzeption und Auswahl der Vorrichtung zur Positionierung der Wegerfassungssensoren	27
3.2.3	Auswahl und Aufbau des Schnittstellensystems	37
<b>4</b>	<b>IMPLEMENTIERUNG</b>	<b>40</b>
4.1	Montage des Magnetbands und der Sensorpositionierungsvorrichtung	40
4.2	Konfiguration des CANopen-Systems	41
4.2.1	Konfiguration der Encoder	41
4.2.2	Abschlusswiderstände	48



<b>4.3</b>	<b>Implementation des Schnittstellensystems</b>	<b>48</b>
4.3.1	Spannungsversorgung und Stromverbrauch	48
4.3.2	Zusammenfügen der Schnittstellen-Komponenten	49
4.3.3	Entwicklungsumgebung Atom	52
4.3.4	Wichtige Elemente der FlexCAN-Bibliothek	52
4.3.5	Programmierung des Mikrocontrollers	53
<b>5</b>	<b>ERGEBNISSE UND VERIFIKATION DES SYSTEMS</b>	<b>58</b>
<b>5.1</b>	<b>Ergebnisse</b>	<b>58</b>
5.1.1	Testfahrten - Lineare Encoder	58
5.1.2	Testfahrten - Rotatorische Encoder	60
<b>5.2</b>	<b>Verifikation</b>	<b>61</b>
5.2.1	Kostenaufstellung	62
5.2.2	Nicht erfüllte Anforderungen	62
5.2.3	Teilweise erfüllte Anforderungen	62
<b>6</b>	<b>DISKUSSION</b>	<b>63</b>
<b>6.1</b>	<b>Bewertung der IMAX-Positionierungsvorrichtung</b>	<b>63</b>
<b>6.2</b>	<b>Bewertung des Schnittstellensystems</b>	<b>63</b>
<b>6.3</b>	<b>Mögliche Ursachen für die differierenden Winkelmesswerte</b>	<b>64</b>
<b>7</b>	<b>ZUSAMMENFASSUNG UND AUSBLICK</b>	<b>65</b>
	<b>LITERATURVERZEICHNIS</b>	<b>66</b>
<b>A</b>	<b>ANHANG</b>	<b>70</b>
<b>A.1</b>	<b>LabVIEW-Blockdiagramm</b>	<b>70</b>

## Abbildungsverzeichnis

Abb. 1–1:	RACOON Funktionsprinzip [1, S. 4] .....	2
Abb. 1–2:	Sonnen und Erdalbedosimulator [1, S. 7] .....	3
Abb. 1–3:	Ursprünglicher Implementierungsstand der Encoder .....	4
Abb. 1–4:	Der Hall-Sensor und seine Wirkungsweise [3, S. 458] .....	6
Abb. 1–5:	Gegenüberstellung einer absoluten und einer inkrementellen zwei- spurigen Magnetbandkodierung .....	6
Abb. 1–6:	Einspurige Absolutkodierung des Magnetbands der IMAX-Encoder [7, S. 8] .....	7
Abb. 2–1:	Zustandsmaschine eines CANopen-Geräts .....	15
Abb. 3–1:	Hardware-Systemdiagramm .....	21
Abb. 3–2:	Auslegung des CANopen-Busnetzes .....	23
Abb. 3–3:	Aufbau Magnetband [6, S. 16] .....	24
Abb. 3–4:	Alle vier topologischen Optionen für die Magnetbandmontage .....	24
Abb. 3–5:	Funktionsfläche 1 .....	25
Abb. 3–6:	Funktionsfläche 2 .....	26
Abb. 3–7:	Funktionsflächen 3 und 4 .....	26
Abb. 3–8:	Stützende Säule für C-Schiene des geostationären Satelliten .....	27
Abb. 3–9:	Zwei fixe Sensorpositionierungen .....	29
Abb. 3–10:	Berechnungsskizze 1 .....	30
Abb. 3–11:	Berechnungsskizze 2 .....	31
Abb. 3–12:	Berechnungsskizze 3 .....	32
Abb. 3–13:	IMAX-Sensorpositionierungsvorrichtung - Variante 1 .....	33
Abb. 3–14:	IMAX-Sensorpositionierungsvorrichtung - Variante 2 .....	34
Abb. 3–15:	IMAX-Sensorpositionierungsvorrichtung - Variante 3 .....	35
Abb. 3–16:	Kräfte-Skizze - Variante 3 .....	36
Abb. 3–17:	Anschlussmöglichkeiten des CompactRIO 9014-Systems .....	37
Abb. 3–18:	Schnittstellen-Komponenten der Microcontroller-Lösung .....	38
Abb. 3–19:	RS-232-Transceiver .....	39
Abb. 4–1:	Montage der Sensorpositionierungsvorrichtung .....	41
Abb. 4–2:	Vollständig implementierte Sensorpositionierungsvorrichtung .....	41
Abb. 4–3:	CAN-USB-Schnittstelle mit Anschlüssen .....	42
Abb. 4–4:	Übergang vom Operation Mode in den Configuration Mode .....	43
Abb. 4–5:	Vorgehen beim Ändern der Knotennummer des IMAX-Sensors .....	44
Abb. 4–6:	Teensy 3.2 mit Pin-Belegung [36] .....	50
Abb. 4–7:	Schaltplan .....	51
Abb. 4–8:	Betriebsbereite Microcontroller-Schnittstelle .....	52
Abb. 4–9:	Programmablaufplan: setup()-Funktion .....	56
Abb. 4–10:	Programmablaufplan: loop()-Funktion .....	57
Abb. 5–1:	Messwerte der IMAX-Encoder bei zwei Testfahrten .....	58



Abb. 5–2: Position des ersten Referenzpunkts ..... 59

## Tabellenverzeichnis

Tab. 1–1:	Auswahl an CANopen- und CAN-Bibliotheken .....	9
Tab. 2–1:	Empfohlene Werte für CANopen-Netzwerke [24, S. 7, Tab. 2] .....	12
Tab. 2–2:	Zustände und Kommunikationsobjekte [25, S. 80, Tab. 37] .....	14
Tab. 2–3:	COB-IDs und Objektverzeichniseinträge der Kommunikationsobjekte [22, S. 106, Tab. 27] .....	14
Tab. 2–4:	Command Specifier bei NMT-Zustandsübergängen [22, S. 101] .....	16
Tab. 2–5:	Anforderungsliste .....	19
Tab. 3–1:	Ursprünglicher Zustand des CANopen-Netzwerks .....	22
Tab. 3–2:	Subsystem-Anforderungsliste .....	28
Tab. 3–3:	Morphologischer Kasten .....	36
Tab. 3–4:	Bewertung der IMAX-Positionierungsvorrichtungen .....	37
Tab. 4–1:	Pin-Belegung des Sub-D Anschlusses bei der Buskonfiguration .....	42
Tab. 4–2:	Knotennummern der Encoder vor und nach der Konfiguration .....	43
Tab. 4–3:	Objekt-Beschreibung des TPDO 1 [32] .....	45
Tab. 4–4:	SDO Leseanfrage und Bestätigungsnachricht .....	46
Tab. 4–5:	Command Code der Bestätigungsnachricht beim Expedited-Transfer [33] .....	46
Tab. 4–6:	SDO Schreibzugriff und Bestätigungsnachricht .....	46
Tab. 4–7:	Command Code für Schreibzugriff beim Expedited-Transfer [33] .....	47
Tab. 4–8:	Heartbeat-Nachricht und NMT-Status .....	47
Tab. 4–9:	Objekt-Beschreibung: Producer Heartbeat Time [32, S. 17] .....	48
Tab. 4–10:	SDO-Schreibzugriff auf Heartbeat Producer Time .....	48
Tab. 4–11:	Sichern aller Parameter im nicht flüchtigen Speicher [32, S. 16] .....	48
Tab. 4–12:	Spannungsversorgung und Stromverbrauch der Schnittstellenkomponenten .....	49
Tab. 4–13:	Funktionen des Can0-Objekts [14] .....	53
Tab. 4–14:	Beschreibung der CAN_message_t-Struktur [14] .....	53
Tab. 4–15:	Identifizier der Heartbeat-Nachrichten .....	54
Tab. 4–16:	Identifizier der Prozessdatenobjekte .....	54
Tab. 4–17:	Zusammenhang zwischen Nutzereingaben und Zustandsübergängen .....	55
Tab. 5–1:	Relevante Positionen und Absolutwerte .....	59
Tab. 5–2:	Positionen und Absolutwerte .....	60
Tab. 5–3:	Verifikation der Anforderungsliste .....	61
Tab. 5–4:	Kostenaufstellung .....	62

## Symbole und Formelzeichen

$\alpha$	deg	Hilfsgröße	$\gamma$	deg	Hilfsgröße
$\beta$	deg	Hilfsgröße	$I$	A	Elektrischer Strom
$B$	T	Magn. Flussdichte	$k_0$	m <sup>2</sup> /C	Hall-Koeffizient
$\delta$	deg	Hilfsgröße	$l_{IMAX}$	mm	IMAX-Länge
$d_1$	mm	Hilfsgröße	$l_{p,IMAX}$	mm	Projizierte IMAX-Länge
$d_2$	mm	Hilfsgröße	$R$	mm	Schienenradius
$d_3$	mm	Hilfsgröße	$s_{max}$	mm	Max. IMAX-Abstand
$F_{D,1}$	N	Druckfederkraft 1	$s_r$	mm	Rollenabstand
$F_{D,2}$	N	Druckfederkraft 2	$U_H$	V	Hall-Spannung
$F_{R,1}$	N	Rollenkraft 1	$x_{11}$	mm	IMAX-Abstand
$F_{R,2}$	N	Rollenkraft 2	$x_{12}$	mm	IMAX-Abstand
$F_Z$	N	Zugfederkraft	$x_2$	mm	IMAX-Abstand

## Akronyme

- ABS** Acrylnitril-Butadien-Styrol
- CAN\_H** CAN high
- CAN\_L** CAN low
- CC** Command Code
- CiA** CAN in Automation
- COB-ID** Communication Object Identifier
- CS** Command Specifier
- GNC** Guidance, Navigation and Control
- HSI** Human Spacecraft Interaction
- LED** Light Emitting Diode
- LOHET** Linear Output Hall Effect Transducer
- LRT** Institute of Astronautics
- LSB** Least Significant Byte
- LSS** Layer Setting Services
- MIT** Massachusetts Institute of Technology
- MSB** Most Significant Byte
- NMT** Networkmanagement
- PDO** Prozessdatenobjekt
- PLA** Polylactid
- RTR** Remote Transmission Request
- SDO** Servicedatenobjekt
- TPDO** Transmission Process Data Object
- TUM** Technische Universität München

# 1 Einleitung

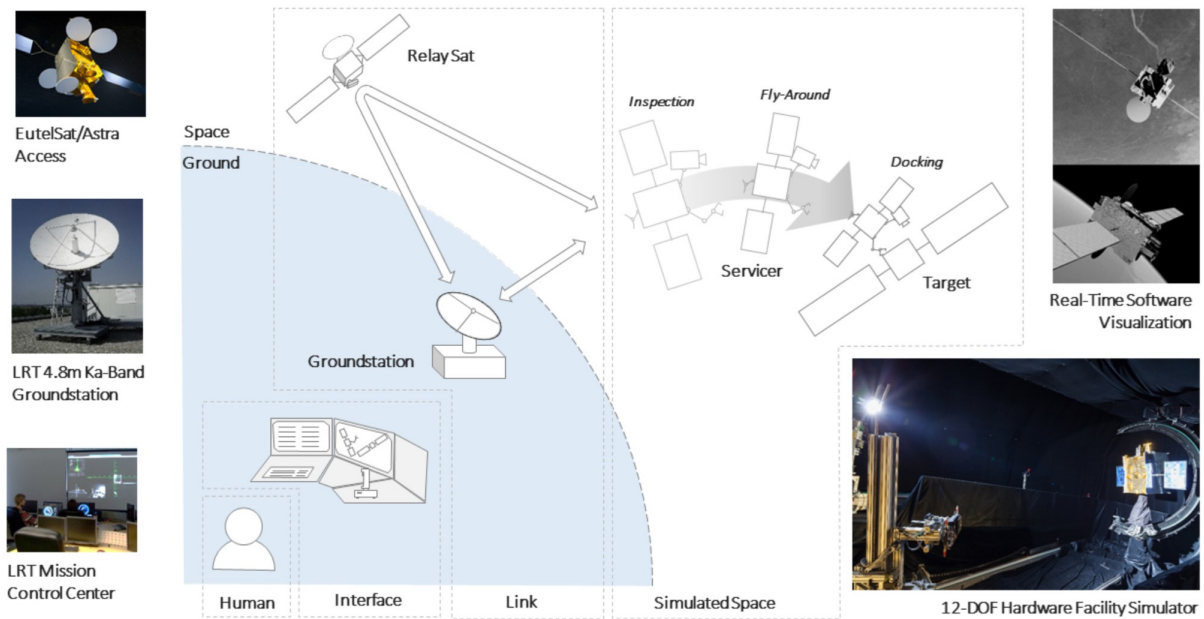
## 1.1 Hintergrund und Motivation

Weltraummüll stellt mittlerweile eine der größten Gefahren für Raumfahrtmissionen dar. Des öfteren sind aktive Satelliten Kollisionen mit Schrottteilen zum Opfer gefallen. Mit einer steigenden Anzahl an Satellitenmissionen gewinnt dieses Problem zunehmend an Bedeutung und Maßnahmen zur Reduzierung des Weltraumschrotts müssen entwickelt werden. Zwei Möglichkeiten den daraus resultierenden Gefahren entgegenzuwirken stellen die Entfernung von Müll aus aktiven Orbits und die Satelliten-Wartung im All dar. Um solche Missionen zu ermöglichen, sind jedoch weitaus verbesserte Systeme für Nahbereichsoperationen und die hochkritischen Rendezvous- und Andockmanöver erforderlich. Dieser daraus abgeleiteten Aufgabe der Entwicklung und Evaluierung neuer Technologien für Nahbereichsoperationen robotischer Weltraumsysteme hat sich der Lehrstuhl für Raumfahrttechnik an der Technischen Universität München (TUM) mit dem sogenannten RACOON-Projekt angenommen [1].

Themenschwerpunkte stellen dabei die Human Spacecraft Interaction (HSI) und die autonome Guidance, Navigation and Control (GNC) dar. Das Ziel von HSI ist den Menschen in die Kontrolle eines robotischen Raumfahrzeugs optimal einzubinden, während sich der zweite Bereich auf Verbesserungen in teil- und vollautonomen Endanflügen an unkooperative Ziele konzentriert [2].

Abbildung 1-1 zeigt den funktionalen Zusammenhang der einzelnen Simulationsbereiche. Dabei kann der Bedienende im lehrstuhleigenen Kontrollzentrum den sogenannten Chaser-Satelliten über die Ka-Band Bodenstation und einen Datenrelaissatelliten im simulierten Weltraum steuern. Es besteht die Möglichkeit den simulierten Weltraum durch Gebrauch der Echtzeit-Software-Visualisierung oder mithilfe der Hardware-Komponenten im RACOON-Laboratorium darzustellen [1].





**Abb. 1–1: RACOON Funktionsprinzip [1, S. 4]**

## 1.2 Stand der Technik

### 1.2.1 RACOON-Labor

Das RACOON-Laboratorium ist ein robotisches Hardware-in-the-Loop System mit zwölf Freiheitsgraden, welches zwei Satelliten und die Lichtverhältnisse durch Sonne und Erde in einem geostationären Orbit simuliert. Der Target-Satellit ähnelt dabei visuell und geometrisch einem um den Faktor vier verkleinerten generischen geostationären Satelliten, wohingegen der Chaser-Satellit lediglich durch eine Sensorplattform repräsentiert wird. Der Freiheitsgrad der Orbitposition wird durch die Position des Sonnen- und des Erdalbedosimulators festgelegt, welche dynamisch veränderliche Lichtverhältnisse während einer Mission simulieren. Dabei kommen Halogen-Metaldampf-Lampen mit einer sonnenähnlichen Spektralverteilung zum Einsatz. Um die Lichtverhältnisse beider Komponenten aufeinander abzustimmen, wurde beim Erdalbedosimulator zusätzlich ein optischer Filter verbaut [1, S. 5f.]. Abbildung 1-2 zeigt beide Simulatoren nebeneinander im Betrieb.

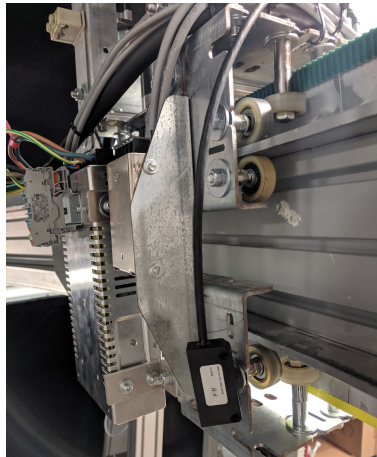


**Abb. 1–2: Sonnen und Erdalbedosimulator [1, S. 7]**

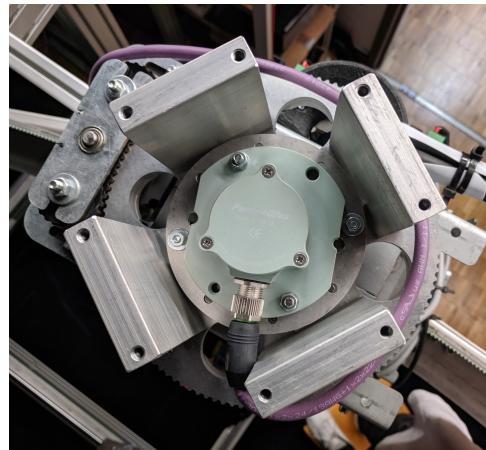
Beide Komponenten sind im Labor auf einer sich um die Satelliten aufgebauten Schiene über Rollen geführt, entlang der sie sich mithilfe des integrierten Schrittmotors bewegen können. Durch einen weiteren Schrittmotor kann jeder Simulator seinen Lampenaufbau um seine Hochachse drehen. Somit besitzen beide Komponenten zwei Freiheitsgrade. Um die Position der Simulatoren zu wechseln, wird die Anzahl der zu fahrenden Schritte über LabVIEW an die Motoren gesendet. Die Bewegungssteuerung ist über einen sich im Nachbarraum befindlichen PC realisiert. Als Schnittstelle zwischen diesem und der Anlage dient das CompactRIO 9014-System von National Instruments. Aufgrund der komplizierten Schienengeometrie und der daraus resultierenden schwierigen Messbarkeit von Fahrtstrecken entlang der Schiene ist eine genaue Umrechnung von Schritten in Meter nicht bestimmbar. Somit gibt es keine Gewissheit über die richtige Positionierung der Simulatoren im Betrieb und Schrittverluste im Positionierungsvorgang können nur schwer erkannt werden.

Um die Positionierung der Simulatoren nachprüfbar und reproduzierbar zu gestalten, sollen zwei Messsysteme implementiert werden. Pro Himmelskörper kommt jeweils ein linearer und ein rotatorischer Hall-Encoder zum Einsatz. Alle vier Sensoren kommunizieren über das Bussystem „CANopen“ und sind bereits in der Anlage teilimplementiert. Abbildung 1-3 dokumentiert den vorgefundenen Einbaustand der Sensoren in den Simulatoren. Während die Drehencoder „SRH520CN“ von Penny & Giles bereits in Sonne und Erde fest installiert sind, fehlt bei den linearen Encodern von ELGO mit dem Namen „IMAX“ noch eine Vorrichtung zur Sensorpositionierung. Beide wurden bereits an die Stromversorgung angeschlossen und auf die gleiche Baudrate von 250 kbit/s

kalibriert. Auch eine CAN-Leitung je Himmelskörper ist bereits verlegt. Während an einem Ende jedes Kabels die Sensoren angeschlossen sind, enden die Leitungen auf der andere Seite ohne weiteren Anschluss.



(a) IMAX



(b) SRH520CN

**Abb. 1–3: Ursprünglicher Implementierungsstand der Encoder IMAX (a) und SRH520CN (b) im Sonnensimulator**

## 1.2.2 Grundlagen Weg- und Positionssensoren

In diesem Unterkapitel werden Grundlagen zur systematischen Unterteilung und zur Funktionsweise von Kodierern vermittelt, wobei verstärkt ein Augenmerk auf die in dieser Arbeit zum Einsatz kommenden Encoder gelegt wird.

Grundsätzlich wird zwischen einer Positionserfassung durch Schalter und durch Wegbeobachtung unterschieden. Bei der Positionserfassung durch Schalter kann das Erreichen einer bestimmten Position durch Näherungsschalter erkannt werden, wohingegen Wegbeobachtung durch Wegmesssysteme erfolgt [3, S. 457]. Letztere werden im Folgenden genauer erläutert.

### 1.2.2.1 Lineare und rotatorische Kodierer

Zwei Arten von Kodierern werden differenziert. Dabei handelt es sich um lineare und rotatorische Encoder. Erstere werden eingesetzt, um translatorische Bewegungen zu erfassen, während rotatorische Encoder Drehbewegungen erfassen können [4]. Beide wandeln eine mechanische Bewegung in ein elektrisches Ausgangssignal um.

### 1.2.2.2 Inkremental- und Absolutmesssysteme

Absolutmesssysteme sind Längen- und Winkelmesssysteme, welche die absolute Position von Translations- oder Rotationsbewegungen in numerischer Form ausgeben [3, S. 462]. Diese können sofort nach Inbetriebnahme einen absoluten Messwert ausgeben, wohingegen bei inkrementellen Systemen nach Einschalten des Kodierers der absolute Positionswert nicht bekannt ist. Eine Bewegung in eine Messrichtung setzt sich

bei inkrementellen Systemen aus einer Folge von positiven oder negativen Zählimpulsen zusammen. Absoluten Drehgebern ist es möglich absolute Winkelpositionen auszugeben, während inkrementelle zum Beispiel die Distanz, die Geschwindigkeit und die Position messen [4]. Drehgeber werden zusätzlich in Single-Turn und Multi-Turn Encoder unterteilt. Letztere können mehrere vollständige Umdrehungen unterscheiden. Während Singel-Turn Encoder für Messungen über kurze Distanzen genutzt werden, sind Multi-turn Encoder passender für längere Messstrecken und komplexere Positionierungsanforderungen [4]. Allgemein werden folgende Messprinzipien von Wegmesssystemen unterschieden:

- kapazitive Längenmessung
- induktive Längenmessung
- Längenmessung nach dem Wirbelstromprinzip
- Längenmessung durch Widerstandspotenziometer
- Dehnmessstreifen
- Hall-Sensoren
- akustische und optische Längensensoren

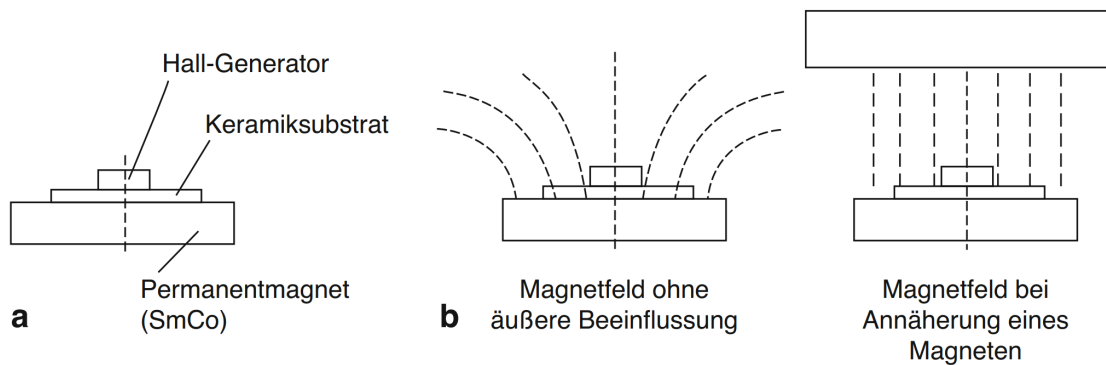
Für den Maschinen- und Anlagenbau sind vor allem die akustischen, optischen und Hall-Messverfahren von besonderer Bedeutung [3, S. 456f.].

### 1.2.2.3 Hall-Sensoren

Die Funktionsweise der Hall-Sensoren basiert auf dem Hall-Effekt. Dabei tritt eine Hall-Spannung  $U_H$  an einem vom Strom  $I$  durchflossenen Leiter auf, wenn sich dieser in einem Magnetfeld der Flussdichte  $B$  befindet.  $k_0$  ist der Hall-Koeffizient, der aus mechanischen Abmessungen und einer Materialkonstanten generiert wird [3, S. 457f.]. Formelmäßig gilt folgender Zusammenhang:

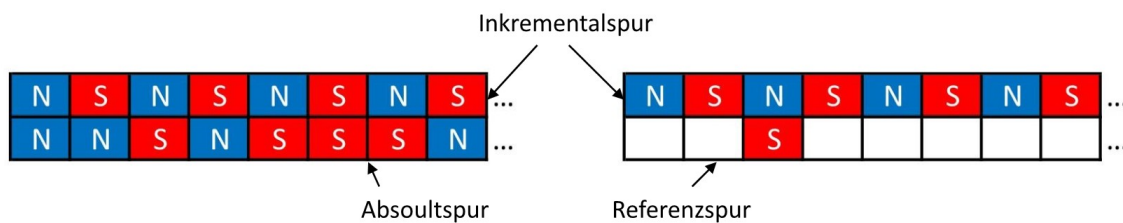
$$U_H = k_0 I B \quad (1-1)$$

Der Aufbau eines Hall-Sensors entspricht der Darstellung in Abbildung 1-4. Auf einem Permanentmagnet ist ein Keramiksubstrat angebracht. Darauf sitzt der sogenannte Hall-Generator. Bei Annäherung eines magnetischen Materials mit gegensätzlicher Polarität zu dem Permanentmagneten richten sich die divergierenden magnetischen Feldlinien parallel aus. Dies bewirkt eine Verstärkung des Feldes durch den Hall-Generator. Folglich steigt die Spannung  $U_H$ , die als Ausgangssignal über einen Linear Output Hall Effect Transducer (LOHET) linear zum Abstand des Werkstücks gemessen wird [3, S. 458]. Bei gleicher Polarität von Permanentmagnet und Magnetband divergieren die Feldlinien und die Spannung sinkt.



**Abb. 1–4: Der Hall-Sensor und seine Wirkungsweise. (a) Aufbau des Hall-Sensors. (b) Wirkungsweise [3, S. 458]**

Neben absoluten linearen Hall-Encodern existieren auch inkrementelle Hall-Sensoren, die Absolutwerte ausgeben können. Beide bewegen sich beim Messvorgang über ein zweispuriges Magnetband, bei welchem sich jede Spur aus einer Aneinanderreihung von Nord- und Südpolen zusammensetzt, die durch die unterschiedlichen Spannungen  $U_H$  unterschieden werden. Beide Systeme besitzen eine Inkrementalspur. Bei den absoluten Sensoren ist die zweite Spur eine Absolutspur, während bei den inkrementellen Kodierern eine Referenzspur zum Einsatz kommt. Dabei ist die Ausgabe von Absolutwerten beim inkrementellen System erst nach Absolvieren einer Referenzfahrt möglich [5]. Beispiele für die zwei unterschiedlichen Magnetbandkodierungen werden in Abbildung 1-5 gezeigt.



**Abb. 1–5: Gegenüberstellung einer absoluten (l) und einer inkrementellen (r) zweispurigen Magnetbandkodierung**

Bei Hall-Sensoren sind Schaltfrequenzen von mehr als 100 kHz bei einem Messabstand von bis zu 7 mm möglich. Da sie berührungslos arbeiten und somit keine mechanische Beanspruchung auftritt, können über 20 Mrd. Schaltspiele erreicht werden [3, S. 458].



#### 1.2.2.4 IMAX und SRH520CN

Das Unternehmen ELGO hat mit dem linearen Hall-Encoder „IMAX“ einen Sensor im Portfolio, dessen Magnetband, wie in Abbildung 1-6 dargestellt, lediglich eine Absolutspur besitzt. Es handelt sich hierbei um ein pseudo-absolutes System. Bei diesem ist, wie bei einem inkrementellen zweispurigen Magnetband, nach dem Einschalten eine einmalige Referenzfahrt notwendig, nach deren Absolvierung der Sensor korrekte Absolutwerte ausgibt. Die Länge der Strecke, die dieser dabei zurücklegen muss, beträgt 274 mm. Durch einen integrierten Istwertspeicher, der vor dem Ausschalten der Betriebsspannung den zuletzt gemessenen Absolutwert nicht flüchtig speichert, wird dem Nutzer die Notwendigkeit einer weiteren Referenzfahrt bei der nächsten Inbetriebnahme erspart. Der Encoder bietet eine Auflösung von bis zu 1 mm (Systemgenauigkeit in  $\mu\text{m}$  bei  $20^\circ\text{C}$ :  $\pm (1250 + 20 \cdot L)$ ,  $L$  = Messlänge in Meter) bei einer Wiederholgenauigkeit von  $\pm 1$  mm [6], [7].

Beim Drehgeber „SRH520CN“ handelt es sich um einen absoluten rotatorischen Single-Turn Hall-Encoder, der einen Winkel von  $0^\circ$  bis  $360^\circ$  mit einer Auflösung von  $0,022^\circ$  ausgeben kann. Der Sensor kann 16384 Positionen differenzieren. Eine Umrechnung des Absolutwerts in Grad findet durch die Multiplikation des gemessenen Positionswerts mit  $\frac{360}{16384}$  statt [8].

Beide Sensoren verfügen über eine CANopen-Schnittstelle gemäß dem Geräteprofil für Encoder - DS 406 [7], [8]. Der IMAX Sensor überträgt 6 Byte Echtzeitdaten. Während die ersten vier die Position des Sensors beschreiben, geben die nachfolgenden 2 Byte die Geschwindigkeit aus. Der Drehwinkelsensor überträgt lediglich seine Position in 4 Byte. Bei beiden wird die Nachricht nach dem Prinzip „Least Significant Byte (LSB) to Most Significant Byte (MSB)“ übertragen.



**Abb. 1–6: Einspurige Absolutkodierung des Magnetbands der IMAX-Encoder [7, S. 8]**

#### 1.2.3 CANopen-Systeme

Für eine Kommunikation mit CANopen-Sensoren wird eine Schnittstelle benötigt, welche sowohl mit dem Busnetzwerk als auch mit dem Computer, von welchem Nachrichten versandt und empfangen werden, verbunden ist. Dabei muss nicht zwingend eine kommerziell erhältliche CANopen-Schnittstelle gewählt werden. Neben fertigen CAN-Schnittstellen, welche mithilfe von entsprechender Software auch CANopen unterstützen, existieren auch diverse Open Source Bibliotheken, die CAN- oder CANopen-Funktionalität für Microcontroller bereitstellen. Im Folgenden werden beispielhaft verschiedene Systeme vorgestellt.

### 1.2.3.1 Vollständige, kommerziell erhältliche CANopen-Lösung

National Instruments bietet mit dem NI-9881 ein High-Speed-CANopen-Schnittstellenmodul für das im RACOON-Lab bereits genutzte CompactRIO-System an, mit dem CANopen-Anwendungen in LabVIEW entwickelt werden können. Allerdings ist diese Schnittstellen-Lösung mit einem Preis von 615 € im Vergleich zu den nachfolgenden die teuerste [9].

### 1.2.3.2 CAN-Schnittstellen mit zusätzlicher Software

Kickdrive Zero von der Flachmann und Heggelbacher BbR und der Fullmo GmbH ist ein kostenloses Softwareprodukt, welches in Kombination mit einer unterstützten CAN-Schnittstelle CANopen-Funktionalität bietet. Hierbei handelt es sich um ein CANopen Tool für Windows. Kickdrive Zero ist die kostenlose und eingeschränkte Version der Software „fullmo Kickdrive“, ist für eine CANopen-Kommunikation mit einsatzfähigen CANopen-Knoten jedoch ausreichend [10]. Unterstützte CAN-Schnittstellen sind ab circa 100 € erhältlich.

Auch das kostenlose Software-Framework „CanFestival“ erweitert CAN-Schnittstellen um eine gewünschte CANopen-Funktionalität. Es vereinfacht die Programmierung von CANopen-Knoten durch automatische C-Code Generierung auf Basis der Beschreibung des gewünschten Objektverzeichnisses. Das Software-Framework dient dabei der Konfiguration des Knotens. Im Betrieb gibt es keine Bindung an die Software. Wie bei Kickdrive werden verschiedene CAN-Schnittstellen unterstützt, darunter stellen AVR Microcontroller die günstigsten dar [11]. Der Einsatz des CanFestival-Frameworks erfordert im Vergleich zu den vorherigen Optionen allerdings einen höheren Implementierungsaufwand.

### 1.2.3.3 Microcontroller

Verschiedene Weiterentwicklungen des CanFestival-Codes sind auf der Open-Source-Plattform „GitHub“ zu finden. Dort gibt es zum Beispiel auch Projekte, die den CanFestival Code als Arduino-Bibliothek lauffähig machen oder Unterstützung für das Betriebssystem „Mbed“ hinzufügen [12], [13]. Des Weiteren existiert die Möglichkeit einen Microcontroller mithilfe von einer CAN-Bibliothek zu programmieren und die gewünschte CANopen-Funktionalität selbst hinzuzufügen. Mit Tabelle 1-2 folgt eine Auswahl an verschiedenen CANopen- und CAN-Bibliotheken.

Die FlexCAN-Bibliothek dient als Treiber für die CAN-Hardware eines Teensy USB Development Boards und wurde in der Arduino-Programmiersprache entwickelt. Die Version von Collin Kidder [14] ist eine Überarbeitung der ursprünglichen Bibliothek. Ein passendes Teensy USB Development Board [15] und ein Philips TJA 1050 High speed CAN-Transceiver [16] sind zusammen für unter 24 € erhältlich.



**Tab. 1–1: Auswahl an CANopen- und CAN-Bibliotheken**

<b>Funktionalität</b>	<b>Bibliothek</b>	<b>Beschreibung</b>
CANopen	CANopen for Python	CANopen-Python-Entwicklungsumgebung für verschiedene kommerziell erhältliche CAN-Schnittstellen [17]
	CANFestivino	Arduino-Bibliothek für CANopen [12]
	KaCanOpen	CANopen stack mit CANopen-zu-ROS Verbindung [18]
	CanFestival	Weiterentwicklung von CanFestival mit Verbesserungen für AVR Microcontroller [19]
	mbed-canfestival	Weiterentwicklung von CanFestival mit Mbed-Unterstützung [13]
CAN	MCP_CAN	CAN-Bibliothek für Arduino [20]
	avr-can-lib	Universelle CAN Bibliothek für AVRs [21]
	FlexCAN	CAN-Arduino-Bibliothek für Teesy 3.1, 3.2, 3.5 und 3.6 [14]

### 1.3 Ziele dieser Arbeit

Im Rahmen dieser Arbeit soll ein System zur Positionsbestimmung von zwei beweglichen Hardwarekomponenten des robotischen Versuchsaufbaus „RACOON-Lab“, welche Sonne und Erde simulieren, designt, implementiert und verifiziert werden. Bei den Messsystemen handelt es sich um zwei rotatorische und zwei translatorische CANopen-Messgeber, welche mittels eines Busnetzwerks und einer Schnittstelle mit dem Steuerungs-PC kommunizieren sollen. Neben der Auswahl und Implementierung der dafür passenden Schnittstelle muss auch eine Vorrichtung zur Positionierung des linearen Encoders entworfen und implementiert werden, die dessen anforderungsgerechten Abstand zum Magnetband im Betrieb gewährleistet. Die an den PC gesendeten Daten sollen eine nachprüfbar und reproduzierbare Positionierung der Simulatoren ermöglichen. Dazu sollen abschließend die Umrechnungsfaktoren von Motorschritten in Meter, beziehungsweise Grad berechnet werden und Referenzpunkte auf der Führungsschiene der Simulatoren definiert werden, die deren Positionierungen vereinfachen sollen.

### 1.4 Abgrenzung

Da die Encoder bereits vor Beginn der Arbeit gekauft wurden, ist deren Auswahl nicht Teil dieser Bachelorarbeit. Auch die mechanische Implementation der Drehgeber in die Anlage ist bereits abgeschlossen und wird deshalb nachfolgend nicht behandelt. Die

Implementation der Sensorkommunikation in das LabVIEW-Programm, mit welchem die Simulatoren angesteuert werden, konnte aufgrund von dessen Komplexität und dem damit einhergehenden zeitlichen Aufwand ebenfalls nicht erfolgen.

## 1.5 Vorgehensweise

Um ein Verständnis für die CANopen-spezifischen Anforderungen sicherzustellen, werden im zweiten Kapitel zunächst grundlegende Kenntnisse des Busprotokolls vermittelt. Die Anforderungen und Randbedingungen, welche die gewünschte Funktionsweise des Positionserfassungssystems beschreiben, werden im Anschluss geklärt und in einer Anforderungsliste festgehalten. Nachfolgend wird das Zusammenwirken der Hardware-Komponenten des Messsystems in einem Systemdiagramm veranschaulicht und der Implementierungsstand des Busnetzwerks dokumentiert. Es folgt die Ortswahl für die Montage des Magnetbands und die Konzeption der Subsysteme, welche für die Erfüllung der Anforderungen erforderlich sind. Dabei wird eine Vorrichtung zur Positionierung der IMAX-Sensoren in den Simulatoren und eine CAN-Schnittstellen-Lösung ausgewählt. Der Abschnitt der Implementierung erläutert die Konfiguration des Busnetzwerks und der Subsysteme. Zusätzlich wird der Einbau dieser in die Anlage und das vorhergehende Zusammenfügen der einzelnen Bestandteile beschrieben. Im sechsten Kapitel werden die Ergebnisse der durchgeführten Testfahrten dargestellt, um die Funktion des CANopen-Messsystems unter Einhaltung der Randbedingungen zu validieren. Abschließend erfolgt eine kritische Bewertung der eingesetzten Subsysteme.

## 2 Anforderungssynthese

### 2.1 CANopen-Spezifikationen

Zunächst soll in diesem Kapitel die grundlegende CANopen-Funktionalität erläutert werden, um anschließend Anforderungen daraus abzuleiten. Dabei wird sich sowohl strukturell als auch inhaltlich nah an dem von Holger Zeltwanger 2001 veröffentlichtem Buch mit dem Titel „CANopen“ [22] gehalten, welches die verschiedenen von der Organisation „CAN in Automation (CiA)“ veröffentlichten CANopen-Spezifikationen ausführlich beschreibt und zusammenfasst.

#### 2.1.1 Überblick

Bei CANopen handelt es sich um ein auf CAN basierendes Kommunikationsprotokoll, das die Applikations-, die Verbindungs- und die physikalische Schicht des OSI-Referenzmodells umfasst. Es beinhaltet Anwendungsschicht, Kommunikationsprofil, verschiedene Rahmenwerke, Applikationsprofile und standardisierte Geräteprofile [22, S. 11].

#### 2.1.2 Physikalische Schicht und Verbindungsschicht

OSI-Schicht 1 und 2 von CANopen bauen auf dem High-speed-Standard ISO 11989-2 und ISO 11989-1 auf. Die zugehörigen CANopen-Protokolle sind CiA 301 und CiA DR-303-1. CANopen-Nachrichten basieren auf dem 11-Bit-Identifizier Adressierungsschema [23]. Die Datenübertragung funktioniert über eine verdrehte Zweidrahtleitung. Dabei werden dominante und rezessive Bits durch die verschiedenen Spannungsdifferenzen zwischen den Drähten unterschieden. Letztere weisen eine Spannungsdifferenz von -500 mV bis +50 mV auf, während bei dominanten Bits eine Spannungsdifferenz von +1,5 V bis +3,0 V anliegt [22, S. 16]. An den beiden Leitungsenden weist ein lineares CANopen-Netz jeweils einen Abschlusswiderstand auf. Die realisierbare Buslänge ist stark von der festgelegten Datenrate abhängig, die zwischen 10 und 1000 kbit/s gewählt werden kann [22, S. 11ff.], [23]. Tabelle 2-1 stellt die in CiA DR-303-1 gezeigte CANopen-Empfehlung für die Netzauslegung bei weniger als 64 Knoten dar.

#### 2.1.3 Layer Setting Services (LSS)

Jeder Slave hat eine Knotennummer von 1 bis 127. Keine Nummer darf doppelt vergeben werden, um eine eindeutige Identifikation der Busteilnehmer zu gewährleisten. Es gibt genau einen Knoten, der die Funktion des NMT-Masters übernimmt und für Aufgaben des Netzwerkmanagements zuständig ist. Dieser hat immer die Knotennummer 0. Bei allen Knoten am Netz muss die Datenrate gleich gewählt sein, um eine funktionierende Kommunikation zu ermöglichen. Über die in CiA DSP-305 definierten Layer Setting Services sind Knoten-ID und Baudrate einstellbar. Diese Dienste sind jedoch nicht immer implementiert [22, S. 170ff.].

**Tab. 2–1: Empfohlene Werte für CANopen-Netzwerke [24, S. 7, Tab. 2]**

Buslänge in m	Buskabel <sup>1</sup>		Abschlusswiderstand in $\Omega$	Baudrate in kbit/s
	längenbezogener Widerstand in $m\Omega/m$	Querschnitt in $mm^2$		
0 - 40	70	0,25 - 0,34	124	1000 (bis 40 m)
40 - 300	< 60	0,34 - 0,6	150 - 300	$\geq$ 500 (bis 100 m)
300 - 600	< 40	0,5 - 0,6	150 - 300	< 100 (bis 500 m)
600 - 1000	< 26	0,75 - 0,8	150 - 300	< 50 (bis 1 km)

<sup>1</sup> bei einem Wellenwiderstand von 120  $\Omega$  und einer Signalverzögerung von 5 ns/m

#### 2.1.4 Grundlegender Aufbau eines CANopen Geräts

Jedes CANopen-Gerät setzt sich aus Kommunikationseinheit, Objektverzeichnis und Applikationseinheit zusammen. Ersteres legt die Zuteilung der CAN-Identifizier und die Nachrichten-Priorisierung fest. Die wichtigsten Elemente der Kommunikationseinheit sind die Prozessdatenobjekte, die Servicedatenobjekte und die Netzwerkmanagement-Objekte. Die Applikationseinheit beschreibt die Funktion des CANopen-Geräts im Bezug auf die Prozessumgebung. Die Verbindung zwischen Kommunikations- und Applikationseinheit stellt das Objektverzeichnis dar. Dieses ist eine strukturierte Anordnung von Daten, das die Konfiguration der Applikation und Kommunikation eines CANopen-Geräts ermöglicht. Die Einträge können durch den jeweiligen Index und Subindex gelesen oder geschrieben werden [22, Seite 40f.].

#### 2.1.5 Kommunikationsobjekte

PDO	Prozessdatenobjekte übertragen die Echtzeitdaten nach dem Producer-Consumer-Modell.
SDO	Servicedatenobjekte können lesend oder schreibend auf die Objektverzeichniseinträge zugreifen. Dies geschieht über das Client-Server-Modell.
NMT	Netzwerkmanagement-Objekte ermöglichen das Steuern der Statusmaschine durch die jeweiligen NMT-Befehle und das Überwachen der Busteilnehmer durch Node-Guarding- oder Heartbeat-Nachrichten.
Vordefinierte Objekte	Synchronisationsbefehl, Zeitstempel, Emergency-Nachricht [22, S. 44f.]

In welchem Zustand die jeweiligen Kommunikationsobjekte übertragen werden können,

wird in Tabelle 2-2 gezeigt .

**Tab. 2–2: Zustände und Kommunikationsobjekte [25, S. 80, Tab. 37]**

	Initialisation	Pre-Operational	Operational	Stopped
PDO			x	
SDO		x	x	
Synchronisation Object		x	x	
Time Stamp Object		x	x	
Emergency Object		x	x	
Boot-Up Object	x			
NMT Objects		x	x	x

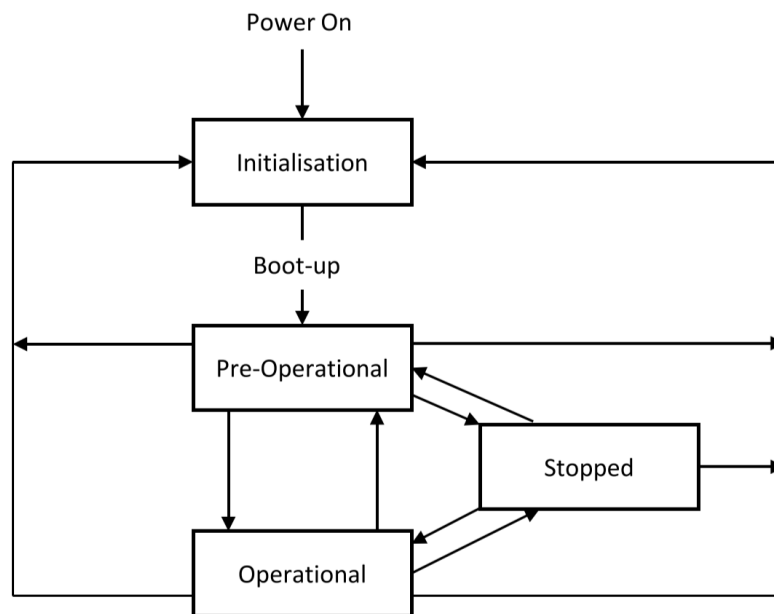
Die verschiedenen Nachrichten werden anhand des Communication Object Identifiers (COB-ID) unterschieden, der sich aus einem Funktionscode (4 Bit) und der Knoten-ID (7 Bit) zusammensetzt [22, S. 105]. Tabelle 2-3 weist den wichtigsten Kommunikationsobjekten ihren Identifier und den zugehörigen Eintrag im Objektverzeichnis zu.

**Tab. 2–3: COB-IDs und Objektverzeichniseinträge der Kommunikationsobjekte [22, S. 106, Tab. 27]**

Objekt	Funktionscode	COB-ID-Bereich	Index im Objektverzeichnis
Broadcast-Nachrichten			
NMT Objects	0000	0	-
Sync	0001	80h	1005h, 1006h, 1007h
Time Stamp	0010	100h	1012h, 1013h
Punkt-zu-Punkt-Nachrichten			
Emergency	0001	81h - FFh	1014h, 1015h
PDO1 (tx)	0011	181h - 1FFh	1800h
PDO1 (rx)	0100	201h - 27Fh	1400h
PDO2 (tx)	0101	281h - 2FFh	1801h
PDO2 (rx)	0110	301h - 37Fh	1401h
SDO (tx)	1011	581h - 5FFh	1200h
SDO (rx)	1100	601h - 67Fh	1200h
NMT Error Control	1110	701h - 77Fh	1016h, 1017h

### 2.1.6 Netzwerkmanagement

Übergänge zwischen den vier Zuständen „Initialisation“, „Pre-Operational“, „Operational“ und „Stopped“ werden entweder durch den NMT-Master oder durch interne Ereignisse ausgelöst. Bei Inbetriebnahme des Systems wird in der Initialisierungsphase gestartet. Wenn diese abgeschlossen ist, springen die Knoten nach automatischem Aussenden einer Boot-up-Nachricht in den Pre-Operational-Modus. Der NMT-Master kann daraufhin einen Befehl senden, um die Busteilnehmer in den Operational-Zustand zu setzen [22, S. 97ff.]. Abbildung 2-1 zeigt die Zustandsmaschine und die verschiedenen Zustandsübergänge.



**Abb. 2–1: Zustandsmaschine eines CANopen-Geräts**

Beim Zurücksetzen in den Initialisierungszustand wird zwischen „Reset\_Node“ und Reset „Reset\_Communication“ unterschieden. Während Letzteres nur die Parameter des Kommunikationsprofils zurücksetzt (1000h bis 1FFFh)<sup>2</sup>, stellt „Reset\_Node“ zusätzlich die Parameter des herstellerspezifischen Bereichs (2000h bis 5FFFh) und des Geräteprofils (6000h bis 9FFFh) zurück [22, S. 101].

Um zwischen den einzelnen Zuständen zu wechseln, sendet der Master eine Nachricht mit zwei Datenbytes. Das erste ist der Command Specifier (CS) und spezifiziert den Zustandsübergang, während das zweite die Knotennummer des adressierten Busteilnehmers angibt. Wenn ein Zustandswechsel aller Teilnehmer angestrebt wird, wird das zweite Byte gleich Null gewählt [22, S. 102]. Der Zusammenhang zwischen Zustandsübergang und Command Specifier ist Tabelle 2-4 zu entnehmen.

<sup>2</sup>Bereich im Objektverzeichnis



**Tab. 2–4: Command Specifier bei NMT-Zustandsübergängen [22, S. 101]**

CS	NMT-Befehl
1h	Start_Remote_Node
2h	Stop_Remote_Node
80h	Enter_Pre_Operational_state
81h	Reset_Node
82h	Reset_Communication

### 2.1.7 Profil für Encoder

In den zu implementierenden Sensoren kommt das Profil für Encoder und Nockenschaltwerke (CiA DSP-406) zum Einsatz. Grundsätzlich werden die Encoder-Klassen C1, C2 und C3 unterschieden. C1 umfasst dabei die Basis-Funktionalität, welche von allen Encodern unterstützt wird. In dieser Klasse ist die Ausgabe von Warn- und Alarmmeldungen nicht möglich. Die Implementierung der Heartbeat-Funktionalität sowie der LSS-Dienste ist optional [26]. Da es sich bei dem IMAX-Sensor und dem SRH520CN-Drehgeber um C1-Encoder handelt, werden die anderen Klassen nicht weiter spezifiziert. Während Baudrate und Node-ID beim linearen Sensor mithilfe der LSS-Dienste veränderbar sind, bietet der rotatorische Encoder diese Einstellmöglichkeiten nicht. Heartbeat-Nachrichten können von beiden versendet werden.

Zusätzlich werden die vier verschiedenen Betriebsarten Polled Mode, Cyclic Mode, COS Mode und Sync Mode unterschieden [22, S. 155]. Diese können bei der Programmierung eines Encoders implementiert werden.

- Polled Mode Die Datenübergabe funktioniert über Remote-Frames.
- Cyclic Mode Die Daten werden ereignisgesteuert, über einen ablaufenden, im Encoder integrierten Timer übertragen.
- COS-Mode Auch der Change-of-State-Modus funktioniert ereignisgesteuert. Positionsdaten werden nur übertragen, wenn sich diese verändert haben.
- Sync-Mode Nach Empfang eines Sync-Telegramms, welches vom NMT-Master ausgesandt wird, gibt der Encoder den absoluten Positionswert zurück [22, S. 155].

Während der Drehencoder SRH520CN sowohl im Cyclic-Mode als auch im Sync-Mode betrieben werden kann, ist beim linearen Sensor IMAX nur der Cyclic-Mode implementiert.

## 2.2 Anforderungserhebung

Um das Positionserfassungssystem optimal auszulegen, ist es zunächst wichtig alle Top-Level-Anforderungen zu erfassen. Dazu werden im Folgenden die Anforderungen, welche die gewünschte Funktionalität des Positionserfassungssystems beschrei-

ben, sowie die einzuhaltenden Randbedingungen ausführlich und begründet erläutert. Abschließend werden diese tabellarisch in einer Anforderungsliste zusammengefasst. Daraus leiten sich Konzeption und Auslegung des Systemaufbaus und die Anforderungen an die zu entwerfenden Subsysteme ab. Die Einhaltung der in diesem Kapitel ausgearbeiteten Anforderungen wird in Kapitel 5 überprüft.

Aufgrund des beschränkten Budgets für Arbeiten am RACOON Lab muss die Implementierung des Sensorsystems in einem festgelegten Kostenrahmen erfolgen. Die Implementierung des Positionserfassungssystems darf den Maximalbetrag von 400 € nicht überschreiten.

Die Grundfunktion des Positionserfassungssystems ist die absoluten Winkelpositionen von Sonnen- und Erdalbedosimulator und deren Absolutpositionen entlang der Führungsschiene zu erfassen und an den Steuerungs-PC zu übermitteln. Dies soll unter Einhaltung von nachfolgenden Randbedingungen erfolgen.

Um eine fehlerfreie Funktion der Encoder zu gewährleisten, muss die Montage und Positionierung der Sensoren den Herstelleranforderungen entsprechen und darf die darin genannten Toleranzen nicht überschreiten. Die Spezifizierung dieser Anforderung findet im Unterkapitel 3.2.2 zur Konzeption der Positionierungsvorrichtung für die IMAX-Sensoren statt.

Gegenstand einer nachfolgenden Arbeit soll das Implementieren einer Regelung zum Verfahren der Simulatoren sein. Die Sensordaten sollen dabei als Ist-Werte dienen, um nach Abgleich mit den gewünschten Positionen Fahrwege in Motorschritte umzurechnen. Deshalb soll bei jedem alle 10 ms stattfindenden Durchlauf der LabVIEW-Schleife ein neuer Sensorwert vorliegen, dessen Messzeitpunkt nicht länger als 10 ms zurückliegt. Somit kann eine Aktualität der Messdaten gewährleistet werden. Die Datenausgabe in LabVIEW ist hierbei Voraussetzung und wird deshalb als Anforderung hinzugefügt.

Damit die Sensoren PDOs übertragen, muss der NMT-Master die Knoten nach Durchlaufen des Initialisierungsprozesses in den Operational-Zustand setzen. Da der Steuerungs-PC zusammen mit einer passenden Schnittstelle die Funktion des Masters übernimmt, muss es möglich sein die Echtzeitdatenübertragung durch eine Eingabe in LabVIEW zu starten.

Falls während des Betriebs auffällt, dass die Sensoren nicht wie gewünscht arbeiten, soll zu jeder Zeit die Option bestehen das CANopen-System zurückzusetzen. Dies soll separat, ohne Beeinflussung eines anderen Bestandteils des Simulationssystems, erfolgen. Somit kann das Beheben des Problems durch das Zurücksetzen der Parameter des Kommunikationsprofils, des herstellereigenen Bereichs und des Geräteprofils der Encoder versucht werden.

Da es sich beim IMAX-Encoder um einen pseudo-absoluten Sensor handelt, muss sich dieser bei Inbetriebnahme und nach dem Überfahren der Enden des Magnetbands mit einer Referenzfahrt über eine Strecke von 274 mm kalibrieren, bevor er korrekte Absolutwerte ausgibt. Um die Nutzung von falschen Positionsdaten zu verhindern, sollen diese von den korrekten eindeutig unterscheidbar sein.



Die nachfolgende Tabelle stellt die herausgearbeiteten Anforderungen an das CANopen-System thematisch geordnet dar. Dabei wird zwischen Umsetzung, Funktion und Systemdesign unterschieden.

**Tab. 2–5: Anforderungsliste**

<b>Top-Level-Anforderungen</b>		
<b>ID</b>	<b>Anforderung</b>	<b>Kategorie</b>
<b>T.1</b>	Die Kosten für die Implementierung des Positionserfassungssystems dürfen 400 € nicht überschreiten	Umsetzung
<b>T.2</b>	Das Positionserfassungssystem soll dem Steuerungs-PC die absoluten Winkelpositionen des Sonnen- sowie des Erdalbedosimulators und die absoluten Positionen entlang der Führungsschiene übermitteln	Funktion
<b>T.3</b>	Die Montage und Positionierung der Sensoren muss den Herstelleranforderungen genügen	Design
<b>T.4</b>	Das Positionserfassungssystem muss die Randbedingungen der Anlage berücksichtigen und darf deren Funktion nicht einschränken	Design
<b>T.5</b>	Der Messzeitpunkt der ausgegebenen Positionsdaten soll bei der Aktualisierung der Steuerungssoftware nicht länger als 10ms zurückliegen	Design
<b>T.6</b>	Die Prozessdatenobjekte der Sensoren sollen am Steuerungs-PC in LabVIEW ausgegeben werden	Design
<b>T.7</b>	Für den Übergang der Sensoren vom Pre-Operational- in den Operational-Modus muss der entsprechende NMT-Befehl vom Steuerungs-PC an die Sensoren gesendet werden können	Design
<b>T.8</b>	Das CANopen-System muss getrennt vom Rest des Simulationssystems jederzeit in den Initialisierungszustand zurückgesetzt werden können	Design
<b>T.9</b>	Die absoluten Positionswerte der IMAX-Sensoren sollen von den während der Referenzfahrt ausgegebenen Werten unterschieden werden können	Design

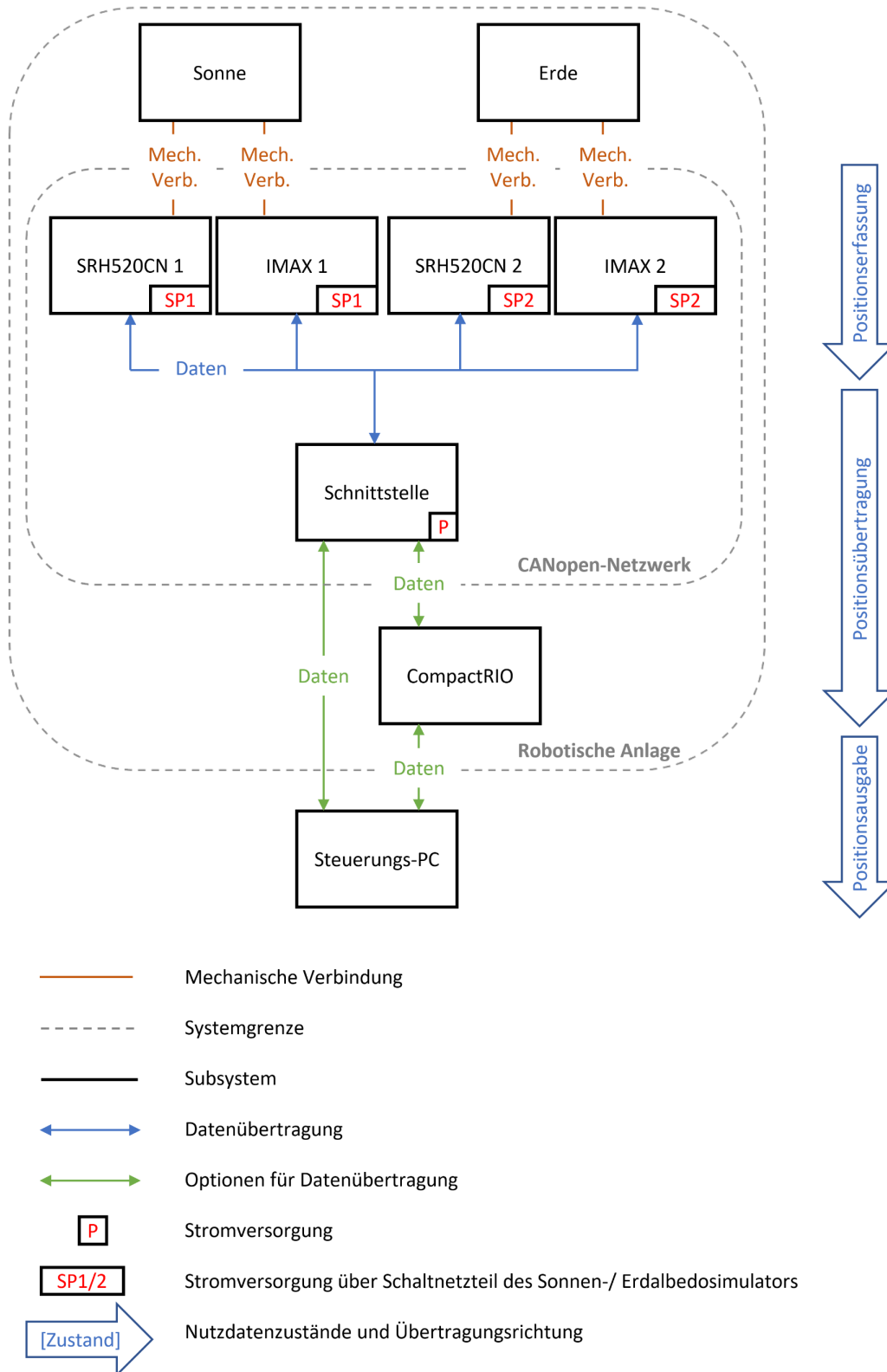
## 3 Konzeption und Design des Systems

### 3.1 Systemebene 0

Dieses Unterkapitel vermittelt einen Überblick über den grundlegenden Aufbau des Positionserfassungssystems und die Einordnung in die robotische Anlage, bevor der strukturelle Aufbau des CANopen-Netzwerks betrachtet wird.

#### 3.1.1 Hardware-Systemarchitektur

Abbildung 3-1 zeigt das Zusammenwirken der einzelnen Hardware-Subsystem-Komponenten. Die Encoder müssen mit der Sonne beziehungsweise der Erde mechanisch verbunden sein, um deren Position erfassen zu können und die Grundvoraussetzung für die Erfüllung von **T.2** zu schaffen. Die Kommunikation der Sensoren mit dem Steuerungs-PC wird über eine an das Busnetzwerk angeschlossene Schnittstelle realisiert. Dabei kann die Verbindung von PC und Schnittstelle entweder direkt oder indirekt über das CompactRIO-System erfolgen und muss der in **T.5** festgelegten, zeitlichen Anforderung gerecht werden. Die Stromversorgung wird im Systemdiagramm nur für die Komponenten des CANopen-Netzwerks angegeben, welche im Rahmen dieser Arbeit implementiert werden. Diese findet bei den Sensoren über das an den jeweiligen Simulator angebrachte Schaltnetzteil statt, während die Stromversorgung der Schnittstelle über den PC, das Schaltnetzteil im Schaltkasten der Anlage oder das CompactRIO erfolgen kann. Da sich der Steuerungs-PC in dem an die robotische Anlage angrenzenden Raum befindet, ist dieser im Diagramm außerhalb des Systems der robotischen Anlage dargestellt.



**Abb. 3–1: Hardware-Systemdiagramm**

### 3.1.2 CANopen-Netzwerk

Der Ist-Zustand des Busnetzwerks wird im Folgenden betrachtet. Unter Zuhilfenahme von Tabelle 2-1 wird die Eignung von bereits implementierten Komponenten überprüft und die Struktur des Busnetzes konzeptioniert.

Da die Drehencoder bereits vom Hersteller auf eine nicht änderbare Baudrate von 250 kbit/s voreingestellt wurden, ist die Datenübertragungsrate nicht beeinflussbar. An die Stichleitungen der Sensoren ist je Simulator eine Zweidrahtleitung mit Schirmung angeschlossen. Vom Simulator ausgehend laufen diese Kabel mit einem Querschnitt von  $0,5 \text{ mm}^2$  in jeweils eine Kabeltrommel. Die CAN-Leitungen, welche die Kabeltrommel verlassen, weisen einen Querschnitt von  $1 \text{ mm}^2$  auf und enden offen. Alle verwendeten Leitungen sind mit einem Wellenwiderstand von  $120 \Omega$  und einer Signalverzögerung von  $5 \text{ ns/m}$  angegeben und entsprechen somit den empfohlenen Kabelparametern aus Tabelle 2-1. Auf Sonnen- und Erdseite sind bereits zwei  $120 \Omega$  Abschlusswiderstände an den Leitungsenden verbaut. Während die IMAX-Encoder über einen internen Abschlusswiderstand von  $120 \Omega$  verfügen, besitzen die Drehencoder keinen zusätzlichen Widerstand zwischen CAN high (CAN\_H) und CAN low (CAN\_L).

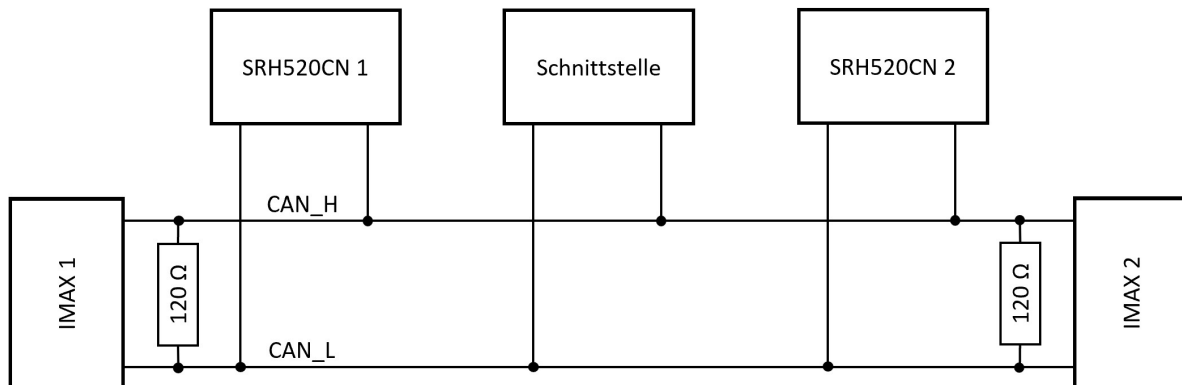
Da das Busnetzwerk eine Kommunikation mit allen vier Sensoren ermöglichen soll, werden die offenen Enden beider CAN-Leitungen vor der Betrachtung des Ist-Zustands gedanklich verbunden. Somit liegt ein linearer CAN-Bus mit einer Länge von 60 m vor. In Tabelle 3-1 findet eine Auflistung der Busparameter im vorgefundenen Zustand statt.

**Tab. 3–1: Ursprünglicher Zustand des CANopen-Netzwerks**

Parameter	Wert
Baudrate	250 kbit/s
Netzlänge	60 m
Leitungsquerschnitt	$0,5 \text{ mm}^2$ ; $1 \text{ mm}^2$
Interner Abschlusswiderstand IMAX	$120 \Omega$
Interner Abschlusswiderstand SRH520CN	-
Verbaute Abschlusswiderstände	$2 \times 120 \Omega$

Mit Blick auf Tabelle 2-1 ist zu sehen, dass die Busnetzlänge für eine Baudrate von 250 kbit/s im Bereich der Empfehlung liegt. Wegen der internen Abschlusswiderstände in den IMAX-Sensoren ist die zusätzlich hinzugefügte Buserminierung überflüssig und muss entfernt werden. Die linearen Encoder werden somit die Enden des CAN-Busses bilden. Folglich liegt die Buserminierung etwas unter der Empfehlung von  $150 \Omega$ . Die noch auszuwählende Schnittstelle sollte also keinen internen Abschlusswiderstand auf CAN-Seite aufweisen, da dieser zu einer weiteren Senkung des gesamten Buswiderstands führen würde. In Abbildung 3-2 wird der strukturelle Aufbau des CANopen-Netzwerks verdeutlicht. Dabei werden die internen Abschlusswiderstände der Komponenten zur Veranschaulichung außerhalb ihrer Systemgrenze dargestellt.





**Abb. 3–2: Auslegung des CANopen-Busnetzes. Die IMAX-Sensoren an den Leitungsenden sorgen für die Buserminierung**

Durch Testen des Busnetzwerks soll bestimmt werden, ob die Verbindungen der verschiedenen Draht-Querschnitte der CAN-Leitungen Probleme verursachen. Falls keine funktionierende Kommunikation zustande kommt, müssen die entsprechenden Kabel ausgetauscht werden.

## 3.2 Systemebene 1

Da die IMAX-Sensoren im Gegensatz zu den Drehencodern noch nicht fest in den Simulatoren verbaut wurden, muss im Umfang dieser Arbeit für das Wegerfassungssystem der Einbauort für dessen Magnetband gewählt und eine anforderungsgerechte Vorrichtung für die Sensorpositionierung konstruiert werden.

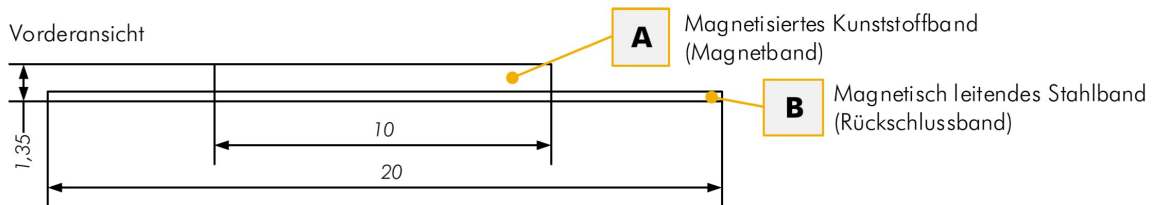
### 3.2.1 Topologieanforderungen für die Magnetbandanbringung

Bei der Auswahl des Ortes für die Magnetbandanbringung müssen die Randbedingungen der Anlage, wie in **T.5** gefordert, berücksichtigt werden. Es wird vom Hersteller empfohlen das Magnetband in einer Nut oder an einer Kante anliegend zu montieren. Außerdem soll es auf einer glatten Oberfläche aufgeklebt werden, da sonst die Messgenauigkeit abnimmt [7, S. 15]. Bei dem kodierten Magnetband handelt es sich um das AB20-80-1-R-D-17-BK100 der Firma ELGO, dessen Aufbau und Maße in Abbildung 3-3 dargestellt sind. Dabei ist das für die Anbringung kritische Maß die Breite des Bandes, welche 20 mm misst.

Draufsicht

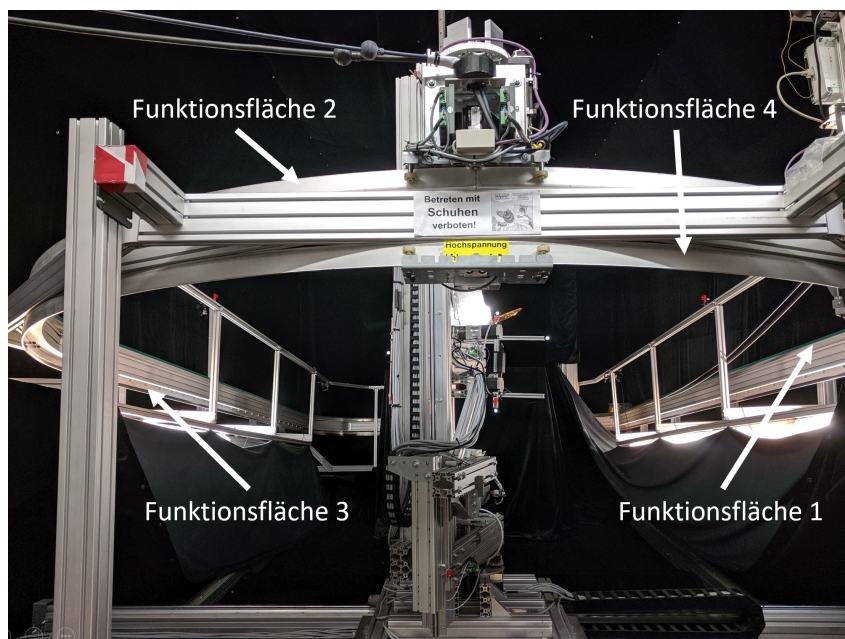
<SN XX/000000001/00000	MB20-80-20-1-R-D-1X-BK100	ELGO
------------------------	---------------------------	------

Vorderansicht



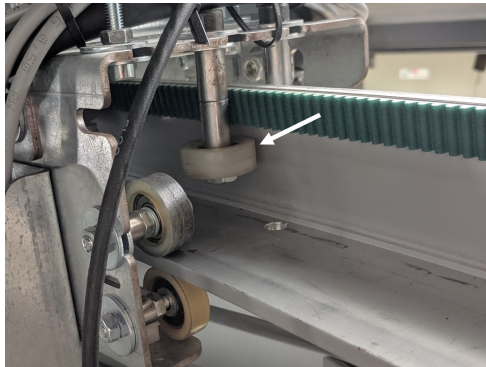
**Abb. 3–3: Aufbau Magnetband [6, S. 16]**

Anhand der folgenden Bilder werden die vier zur Verfügung stehenden Funktionsflächen der Rollenführung vorgestellt und hinsichtlich ihrer Eignung für die Magnetbandmontage bewertet. Abbildung 3-4 ordnet diese den verschiedenen Bereichen auf der Schiene zu. Da die Aufnahme nach der Magnetbandmontage angefertigt wurde, ist das Band auf Funktionsfläche 3 sichtbar.

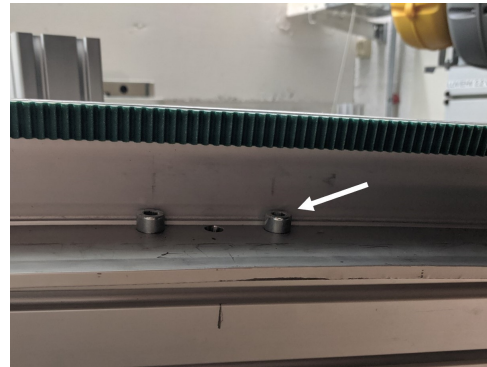


**Abb. 3–4: Alle vier topologischen Optionen für die Magnetbandmontage**

Die Möglichkeit das Magnetband unterhalb der in (a) markierten Rolle zu positionieren, zeigt Abbildung 3-5. Diese kann aufgrund des Zahnriemens kaum nach oben verschoben werden. Da der Abstand zwischen der Rolle und den in (b) gezeigten Schraubenköpfen die Breite des Bands unterschreitet, ist eine Anbringung auf Funktionsfläche 1 nicht möglich.



(a) Rolle



(b) Schraubenköpfe

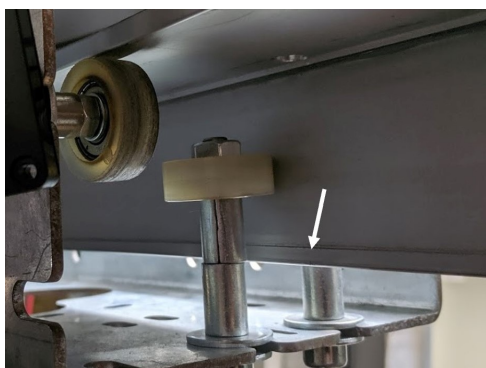
**Abb. 3–5: Funktionsfläche 1**

Funktionsfläche 2 befindet sich auf der Außenseite der Führungsschiene und ist in Abbildung 3-6 dargestellt. Die obere Rolle liegt dem Kraftangriffspunkt des Schrittmotors genau gegenüber. Ihre Position darf nicht verändert werden. Die beiden unteren Rollen liegen auf gleicher Höhe zu den Rollen auf der Innenseite, um sich gegenseitig abzustützen. Somit sollen bei Versatz der Rollen auf der Außenseite auch die innen laufenden Rollen um den selben Weg verlängert beziehungsweise gekürzt werden. Da die Rollen auf der Innenseite kaum nach oben oder nach unten verschoben werden können, gilt dies ebenso für die Rollen auf der Außenseite. Selbst bei maximalem Versatz ist der vertikale Abstand zwischen den in Abbildung 3-9 markierten Rollen zu gering für die Magnetbandmontage. Unterhalb der Rollen tritt bei den ungeraden Schienenteilen ein Knick in der Führungsschiene, einhergehend mit einer starken Unebenheit auf der Funktionsfläche, auf. Da auch die glatte Fläche unterhalb der Rollen zu wenig Platz für die Anbringung des Bands bietet, scheidet Funktionsfläche 2 ebenfalls als Option aus.

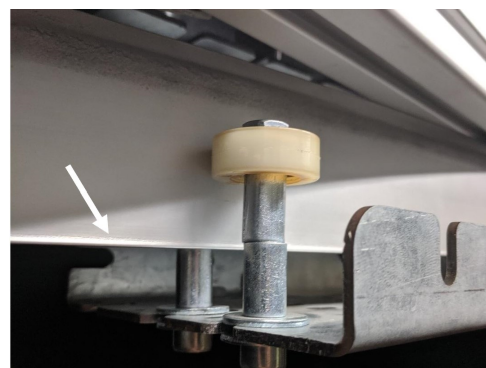


**Abb. 3–6: Funktionsfläche 2**

Abbildung 3-7 zeigt die unteren beiden Funktionsflächen. Sowohl Innenseite (a), als auch Außenseite (b) bieten über sowie unter den Rollen die Möglichkeit das Magnetband zu montieren. Bei beiden Optionen ist ein Versatz der Rollen erforderlich. Durch die Montage an der Unterkante kann eine gerade und somit spannungsarme Montage des Magnetbands leicht erreicht werden. Zusätzlich bietet das Blech, auf welchem die unteren Rollen montiert sind, durch die vielen bereits vorhandenen Langlöcher den Vorteil, eine Vorrichtung zur Sensorpositionierung mit geringem Aufwand anbringen zu können. Die kleine Unebenheit, welche innen und außen nahe der Unterkante sichtbar ist, kann durch eine geeignete Auswahl eines doppelseitigen Klebebands für die Magnetbandmontage ausgeglichen werden. Aus diesen Gründen soll das Magnetband an Funktionsfläche 3 oder 4 angebracht werden. Die Unterkante soll als Montageort bevorzugt werden.



(a) Funktionsfläche 3

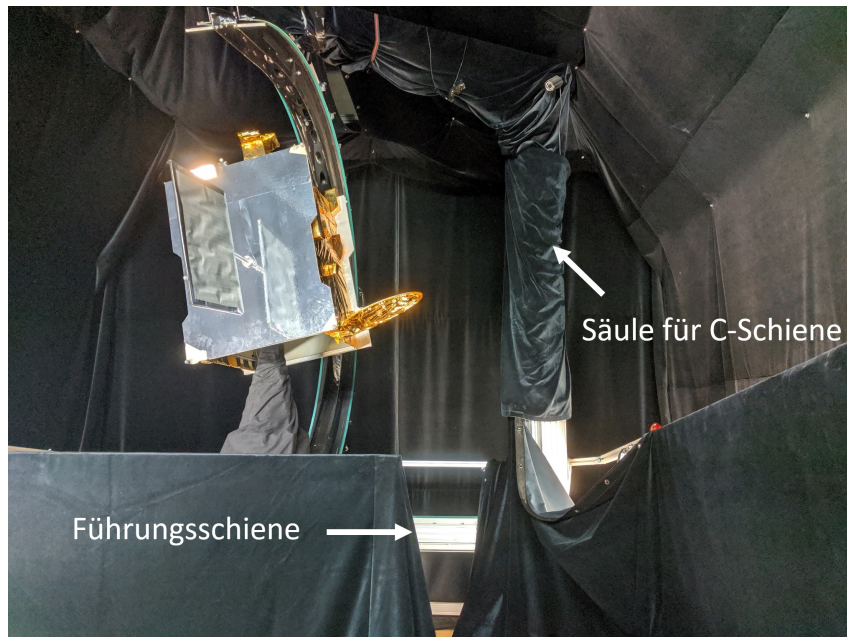


(b) Funktionsfläche 4

**Abb. 3–7: Funktionsflächen 3 und 4. (a) zeigt die Innenseite, während (b) die Außenseite des unteren Bereichs der Führungsschiene darstellt.**



Da sich die IMAX-Sensoren nach Überfahren der Magnetband-Enden neu kalibrieren müssen, um absolut messend zu operieren, soll sich dieser Übergang an einer möglichst unkritischen, selten frequentierten Stelle auf der Schiene befinden. Eine Positionierung eines Simulators hinter der Säule, welche die C-Schiene stützt, verhindert eine realistische Beleuchtung der Satelliten aufgrund der Verdeckung der Simulator-Lampe und ist für die Simulation deshalb unbrauchbar. Anfang und Ende des Magnetbands sollen deshalb hinter dieser in Abbildung 3-8 markierten Säule liegen.



**Abb. 3–8: Stützende Säule für C-Schiene des geostationären Satelliten**

### **3.2.2 Konzeption und Auswahl der Vorrichtung zur Positionierung der Wegfassungssensoren**

#### **3.2.2.1 Anforderungen an die Sensorpositionierung**

Für die Konzeption der Vorrichtung zur IMAX-Sensorpositionierung werden in Tabelle 3-2 Top-Level-Anforderungen spezifiziert und Subsystem-Anforderungen aus diesen abgeleitet. Alle sind dem Design zuzuordnen. In der rechten Spalte werden die Anforderungen referenziert, in welchen die jeweiligen Subsystem-Anforderungen ihren Ursprung haben.

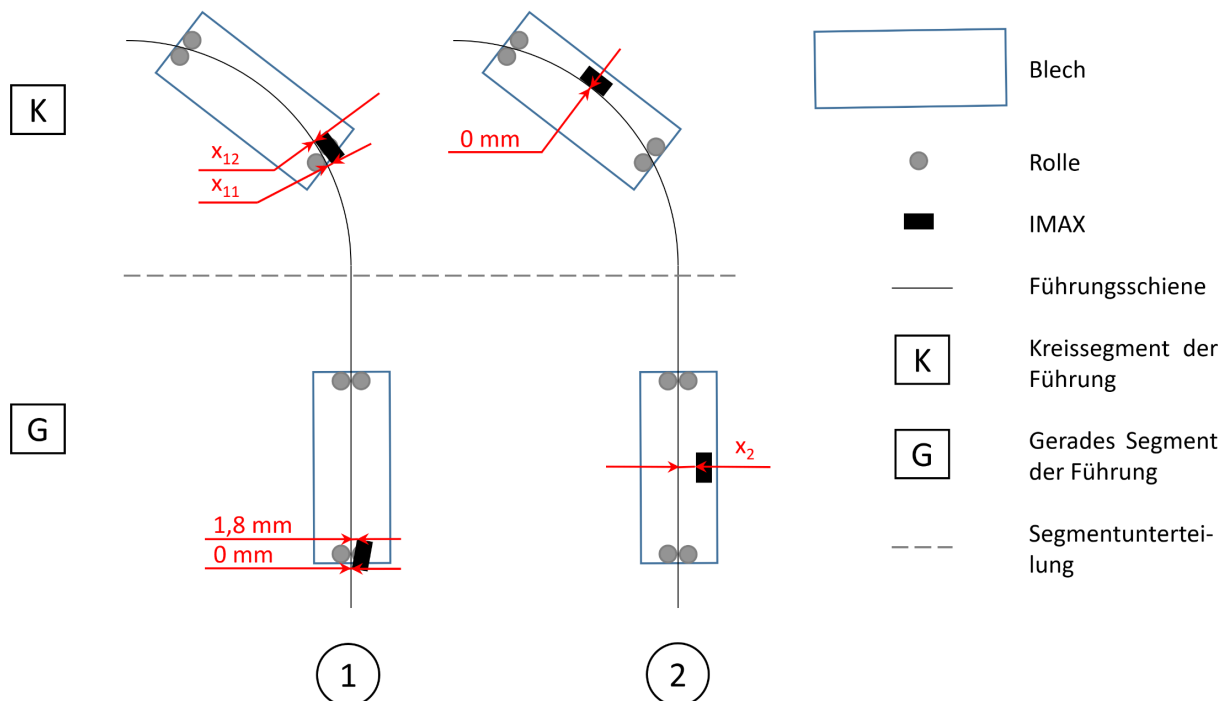
Tab. 3–2: Subsystem-Anforderungsliste

Anforderungen an die Sensorpositionierung der linearen Encoder		
ID	Anforderung	Ursprung
<b>S.1</b>	Der Leseabstand des Sensors darf an der am weitesten vom Magnetband entfernten Stelle maximal 1,8 mm betragen	<b>T.3</b>
<b>S.2</b>	Der Seitenversatz des Sensors zum Magnetband darf den Betrag von 1 mm nicht überschreiten	<b>T.3</b>
<b>S.3</b>	Der Gierwinkel des Sensors zum Magnetband darf 0,5° nicht überschreiten	<b>T.3</b>
<b>S.4</b>	Das System zur Sensorpositionierung soll mit den lehrstuhleigenen Werkzeugen fertigbar sein	<b>T.1</b>
<b>S.5</b>	Das Sensorpositionierungssystem muss Bewegungen des Sonnen- und Erdalbedosimulators von 2,5 mm in positiver sowie negativer Richtung entlang der Hochachse der Führungsschiene ausgleichen können	<b>T.3, T.4, S.2</b>
<b>S.6</b>	Die Vorrichtung zur Sensorpositionierung soll Beweglichkeit des Sensors relativ zum Simulator auf einer zum Magnetband senkrechten Ebene schaffen, während sie den entlang der Schiene gemessenen Abstand des Encoders zum Simulator konstant hält	<b>T.3, T.4, S.1</b>

**S.1** bis **S.3** wurden dabei der Betriebsanleitung des IMAX-Sensors entnommen [7, S. 18]. Bei **S.5** und **S.6** bedarf es weiterer Erläuterung:

**S.5** leitet sich aus **S.2** ab. Die vertikale Position des Sonnen- und des Erdalbedosimulators ist mit acht Führungsrollen überbestimmt. Durch Vibrationen im Betrieb kommt es zu einer Lockerung der Schrauben, welche den Kraftschluss zwischen Simulator und Rollen herstellen. Dies ermöglicht den Simulatoren kleine Bewegungen relativ zur Hochachse der Schiene auszuführen, welche die Seitenversatz-Toleranz der IMAX-Sensoren überschreiten. Da mit steigender Entfernung zur Kabelrolle deren auf den Simulator wirkende Zugkraft wächst, kann es je nach Stellung zu dieser neben dem Absinken auch zum Steigen des Simulators kommen. Auch Übergänge zwischen Schienenstücken, welche einen vertikalen Versatz aufweisen, müssen berücksichtigt werden. Die Minimalanforderung an den vertikalen Ausgleich der Positionierungsvorrichtung wurde mit einem Wert von 5 mm festgelegt (**S.5**). Dieser sollte ausreichend für den funktionierenden Einsatz der Vorrichtung über einen großen Zeitraum sein, bevor die Simulatorrollen nachjustiert werden müssen.

Die Erfüllung der Anforderung **S.6** wird durch **S.1** verlangt. Die Führungsschiene hat sowohl gerade als auch gebogene Abschnitte. Durch den Abstand der auf dem Blech montierten Rollen, tritt bei fixer Sensorpositionierung das in Abbildung 3-9 schematisch dargestellte Problem auf. Der größte Abstand des linearen Encoders zum Magnetband überschreitet bei fixer Sensorpositionierung entweder auf gerader oder gekrümmter Strecke die in **S.1** geforderte Toleranz. Somit muss das IMAX-Positionierungssystem die in **S.6** geforderte Beweglichkeit bereitstellen.



**Abb. 3–9: Zwei fixe Sensorpositionierungen und deren Abstände zur Schienenführung in geradem und gekrümmtem Führungsschiensegment**

Im Folgenden wird nachgewiesen, dass ein starrer Sensoreinbau des IMAX-Encoders bei gleichzeitiger Erfüllung von **S.1** nicht möglich ist. Bei den in Abbildung 3-9 dargestellten Kreissegmenten soll es sich um den kleinsten Radius handeln, der in der Anlage vorkommt, da der Vergleich von gerader Strecke zu diesem den größten Abstandsunterschied zwischen Sensor und Führungsschiene aufzeigt. Diese Abstandsdifferenz fällt bei der in Abbildung 3-9 unter 1 gezeigten Position über einer Rolle auf der Schienenaußenseite am geringsten aus. Der minimale Abstand senkrecht zum Magnetband soll  $0\text{ mm}$  betragen, während der maximale  $1,8\text{ mm}$  misst.  $x_{11}$  und  $x_{12}$  müssen einen Wert zwischen  $0\text{ mm}$  und  $1,8\text{ mm}$  annehmen, damit diese fixe Sensorpositionierung theoretisch realisierbar ist. Der Beitrag des Magnetbands und des doppelseitigen Klebebands zum Radius wird hierbei vernachlässigt, da diese nur einen sehr geringen Anteil am Radius haben.

**Gegeben:**

1-K:  $R = 800 \text{ mm}$ ;

1-G: Min. Abstand:  $0 \text{ mm}$ ; Max. Abstand:  $s_{max} = 1,8 \text{ mm}$ ;

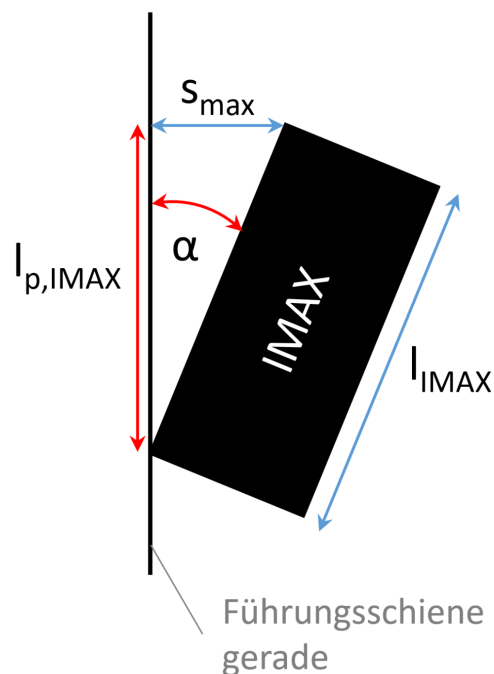
Abstand der Rollen zueinander:  $s_R = 303 \text{ mm}$ ;

Länge des IMAX Sensors:  $l_{IMAX} = 50 \text{ mm}$ ;

**Zu überprüfen:**

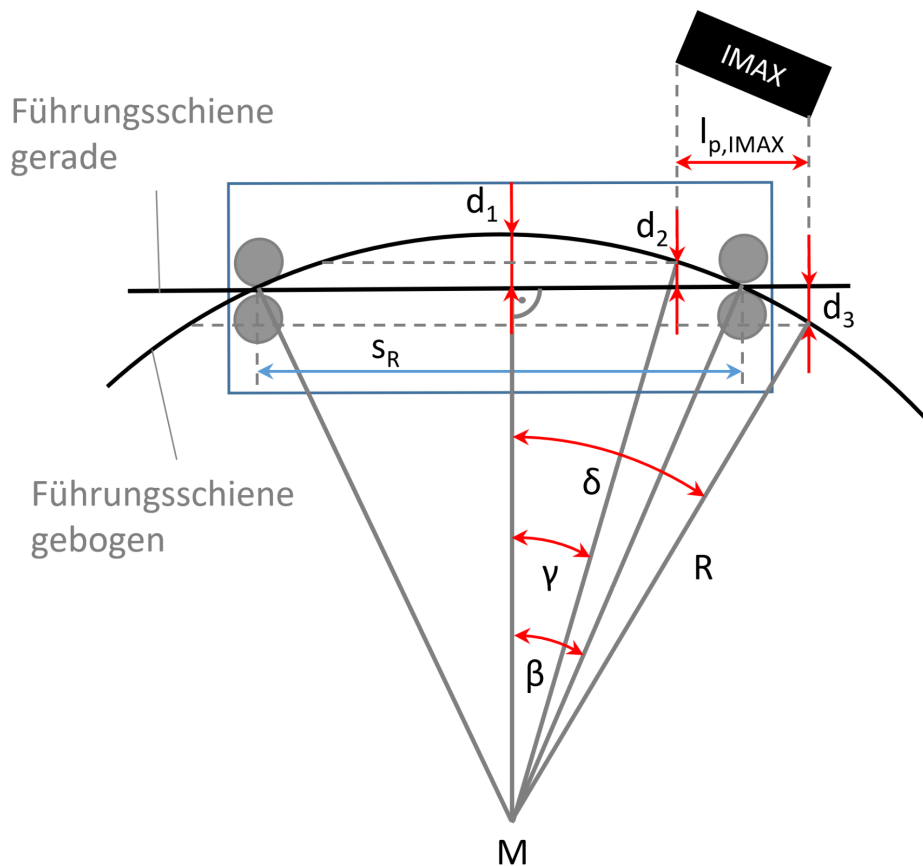
1-K:  $0 \text{ mm} \leq x_{11} \leq 1,8 \text{ mm}$ ;  $0 \leq x_{12} \leq 1,8 \text{ mm}$ ;

Zur Überprüfung der Bedingungen für  $x_{11}$  und  $x_{12}$  werden mehrere Hilfsgrößen definiert und in Abbildung 3-10 und 3-11 dargestellt. In allen nachfolgenden Berechnungsskizzen werden die zu Beginn der Berechnung gegebenen Größen blau und die zu berechnenden Größen rot gekennzeichnet.



**Abb. 3–10: Berechnungsskizze 1**





**Abb. 3–11: Berechnungsskizze 2**

Anfangs muss die projizierte Länge des IMAX-Sensors  $l_{p,IMAX}$  aufgrund seiner Schräglage zum geraden Schienensegment berechnet werden. Die Abstandsänderung des Sensors zur Führungsschiene, welche durch die Änderung der Abrollstelle der unter dem Sensor liegenden Rolle auftritt, wird hierbei vernachlässigt.

$$\alpha = \arcsin\left(\frac{s_{max}}{l_{max}}\right) \quad (3-1)$$

$$l_{p,IMAX} = \frac{s_{max}}{\tan(\alpha)} \quad (3-2)$$

Somit ist  $l_{p,IMAX}$  bekannt. Um  $d_2$  und  $d_3$  berechnen zu können, wird nun  $d_1$  bestimmt. Dazu wird der Winkel  $\beta$  benötigt.

$$\beta = \arcsin\left(\frac{\frac{s_R}{2}}{R}\right) \quad (3-3)$$

$$d_1 = R(1 - \cos(\beta)) \quad (3-4)$$

Der Abstand  $d_1$  beträgt  $14,48\text{ mm}$ . Folgend werden die Winkel  $\gamma$  und  $\delta$  berechnet. Dabei entspricht die Strecke von der jeweiligen projizierten Bauteilkante des Sensors zur Mitte des Blechs  $\frac{s_R}{2} - \frac{l_{p,IMAX}}{2}$  für  $\gamma$  und  $\frac{s_R}{2} + \frac{l_{p,IMAX}}{2}$  für  $\delta$ .

$$\gamma = \arcsin\left(\frac{\frac{s_R}{2} - \frac{l_{p,IMAX}}{2}}{R}\right) \quad (3-5)$$

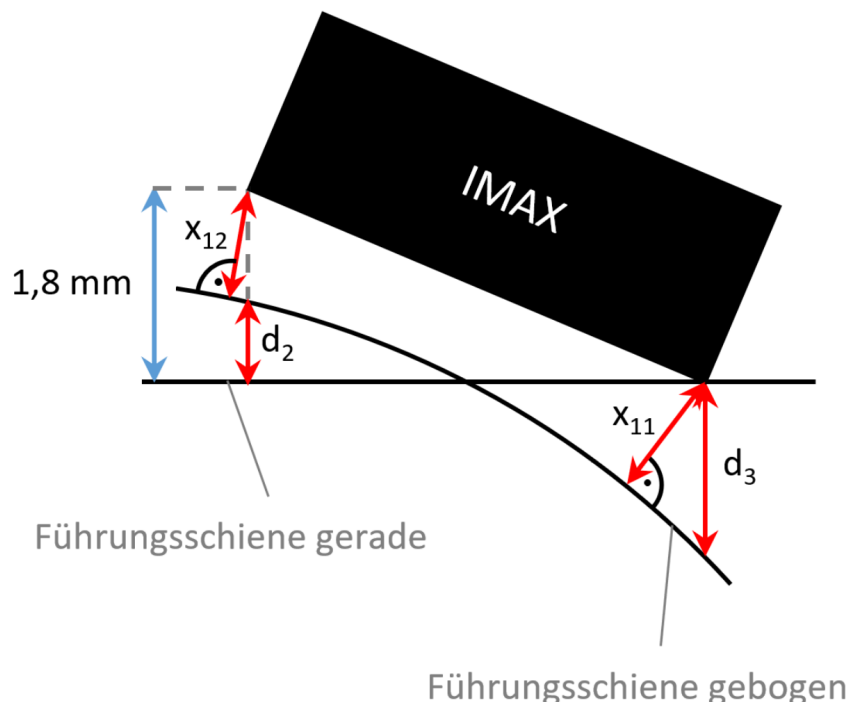
$$\delta = \arcsin\left(\frac{\frac{s_R}{2} + \frac{l_{p,IMAX}}{2}}{R}\right) \quad (3-6)$$

Schließlich sind alle Hilfsgrößen für die Berechnung von  $d_2$  und  $d_3$  gegeben. Somit können diese im Folgenden bestimmt werden.

$$d_2 = d_1 - R(1 - \cos(\gamma)) \quad (3-7)$$

$$d_3 = R(1 - \cos(\delta)) - d_1 \quad (3-8)$$

Mit den gegebenen Werten erhält man  $d_1 = 4,41\text{ mm}$ ,  $d_2 = 4,41\text{ mm}$  und  $d_3 = 5,23\text{ mm}$ . In Abbildung 3-12 wird zur Überprüfung der Bedingungen für  $x_{11}$  und  $x_{12}$  der Zusammenhang zu den Hilfsgrößen  $d_2$  und  $d_3$  dargestellt.



**Abb. 3-12: Berechnungsskizze 3**

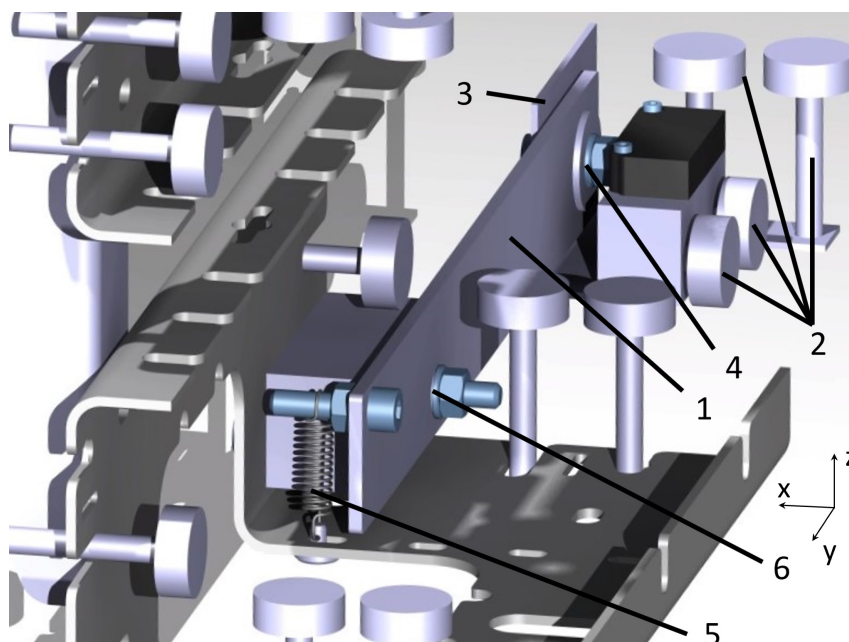
Da der berechnete Wert für  $d_2$  größer als  $1,8\text{ mm}$  ist, nimmt  $x_{12}$  einen negativen Wert an. Somit ist eine der zu überprüfenden Bedingungen nicht erfüllt und folglich ist keine fixe Sensorpositionierung möglich.

### 3.2.2.2 Auswahl der Vorrichtung

Die im Folgenden vorgestellten Vorrichtungen zur Sensorpositionierung wurden alle in CATIA V5 mit dem Ziel konstruiert, die Anforderungen **S.1** bis **S.6** zu erfüllen. Zunächst wird deren Funktionsweise beschrieben. Der nachfolgende Vergleich ist Grundlage für die Auswahl des Systems, welches im nächsten Kapitel implementiert werden soll.

Variante 1:

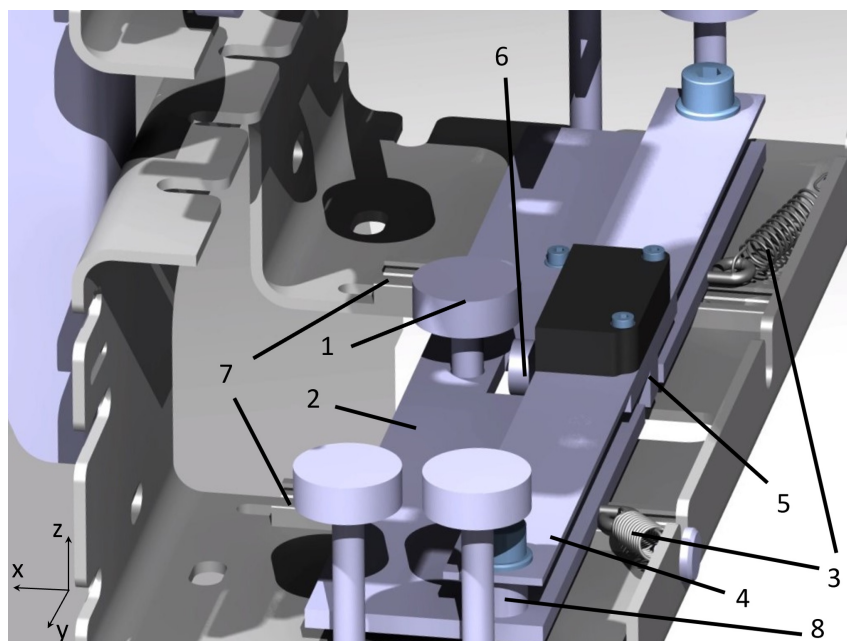
Abbildung 3-13 zeigt die erste Möglichkeit für eine anforderungsgerechte Befestigung der IMAX-Sensoren am Simulator. In dieser Variante wird der Sensor neben dem jeweiligen Simulator über einen elastischen Plastikstreifen aus Acrylnitril-Butadien-Styrol (ABS) (1) und vier Rollen (2) in Position gehalten. Das Magnetband befindet sich in diesem Fall anliegend an der Unterkante von Funktionsfläche 3. Das ABS-Teil kann sich in Kurven elastisch biegen und erfüllt somit **S.6**. Das Winkelblech (3), an welchem der Sensor angebracht ist, ist über eine Hülse (4) mit dem Plastikstreifen verbunden. Diese einfache Gleitlagerung ermöglicht den beiden Bauteilen sich relativ zueinander zu verdrehen. Durch die Zugfeder (5) links neben der zweiten Gleitlagerung (6) werden die Rollen unter dem IMAX-Sensor gegen die Unterkante der Schienenführung gedrückt. Somit bleiben der Gierwinkel und der Seitenversatz zwischen Sensor und Magnetband unbeeinflusst von Bewegungen der Simulatoren in vertikaler Richtung mit der geforderten Toleranz (**S.5**). Durch die zwei rechts neben dem Sensor abrollenden Führungsrollen und dem über dessen gesamte Länge fast gleichbleibenden Abstand des ABS-Teils zur Funktionsfläche 3, kann die Längsneigung und die Seitenneigung innerhalb der Toleranzen gehalten werden.



**Abb. 3–13: IMAX-Sensorpositionierungsvorrichtung - Variante 1**

### Variante 2:

Die zweite Variante, den Sensor in anforderungsgerechtem Abstand zum Magnetband zu halten, ist in Abbildung 3-14 dargestellt. Das Band ist hierbei anliegend an der Unterkante von Funktionsfläche 4 angebracht. Da sich der IMAX-Sensor in der Mitte des Blechs befindet, muss das System die Distanz  $x_2$  ausgleichen können, welche in Abbildung 3-9 unter 2 dargestellt ist.  $x_2$  gleicht dabei der Hilfsgröße  $d_1$ . Das Zurücklegen dieser Strecke von  $14,48\text{ mm}$  wird durch eine Schienenkonstruktion auf dem Blech ermöglicht. Die Rolle (1), welche auf Funktionsfläche 3 läuft und fest mit der Aluminiumplatte (2) verschraubt ist, sorgt für den anforderungsgerechten Abstand zwischen Encoder und Magnetband. Der Kontakt dieser Rolle mit der Schiene wird durch die zwei Zugfedern (3) garantiert. Der vertikale Bewegungsausgleich von 5 mm findet durch den elastischen Plastikstreifen (4) und die darunter angebrachten Druckfedern statt. Diese werden durch die Radaufhängung (5) der kleinen Rolle (6) in Position gehalten. Während ein Ende jeder Druckfeder auf der Aluminiumplatte aufliegt, drückt das andere gegen den Plastikstreifen. Das bewirkt das Anpressen der kleinen Rolle gegen die Unterkante der Schienenführung. Die Platte, auf welcher die Federn stehen, ist auf den zwei Gleitelementen der Schienenführung (7) gelagert. Zwei Abstandshülsen (8) halten die Enden des Plastikstreifens in einem definierten Abstand zur dieser.

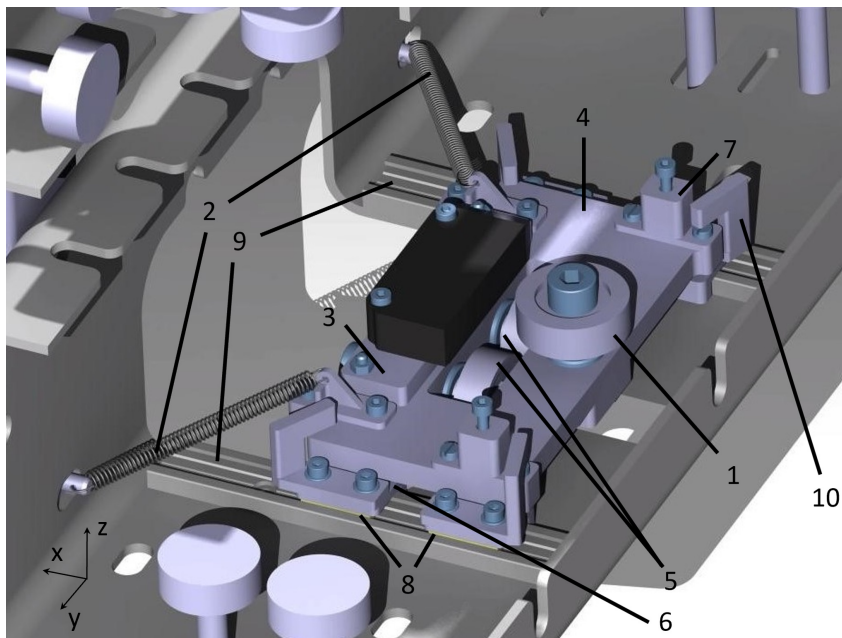


**Abb. 3–14: IMAX-Sensorpositionierungsvorrichtung - Variante 2**

### Variante 3:

Die in Abbildung 3-15 dargestellte dritte Variante ist wie die zweite ebenfalls mittig auf dem Blech positioniert und muss deshalb auch  $x_2$  ausgleichen können. Das Magnetband liegt an der Unterkante von Funktionsfläche 3 an. Auch hier findet die Abstandskontrolle senkrecht zur Schiene durch die dem Magnetband gegenüberliegende Rolle (1) und zwei Zugfedern (2) statt. Der Sensor ist auf einem Abstandsklotz (3) montiert und mit der darunter liegenden Platte (4) über zwei Langlöcher verschraubt. Die Ent-

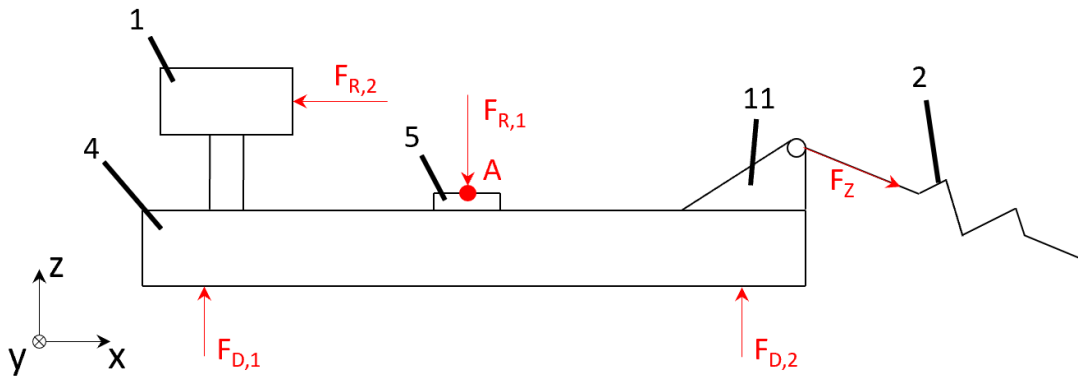
fernung zwischen Encoder und Magnetband ist folglich einstellbar. Die beiden unter den Sensor liegenden Rollen (5) sind mit der Platte verschraubt und werden an die Unterkante der Führungsschiene angepresst. Die Höhenregelung funktioniert über die vier Druckfedern, auf welchen die Platte gelagert ist. Jede einzelne wird über eine kurze Zylinderführung (6) in Position gehalten. Die beiden Federn, welche auf der dem Sensor gegenüberliegenden Seite liegen, sind durch die entsprechenden Aufbauten (7) einstellbar, um Korrekturen und Nachjustierungen vornehmen zu können. Die Zylinderführungen sind auf Gleitelementen (8) montiert, von welchen jeweils zwei in einer auf dem Blech montierten Schiene (9) geführt werden. Durch die L-förmigen Aufbauten (10) über den Gleitelementen wird ein vertikaler Maximalabstand von Platte zu Blech definiert, um das Austreten der Zylinderführungen aus der Platte zu verhindern. Somit kann bei Überschreiten des in **S.5** geforderten Höhenausgleichs die Seitenversatz-Toleranz des Encoders genutzt werden.



**Abb. 3–15: IMAX-Sensorpositionierungsvorrichtung - Variante 3**

Aufgrund der Systemkomplexität dieser Vorrichtung werden zusätzlich die auf diese wirkenden Kräfte schematisch in seitlicher Ansicht in Abbildung 3-14 dargestellt. Dabei werden beide Zugfederkräfte, jeweils zwei Druckfederkräfte und die auf die unteren Führungsrollen wirkenden Kräfte zu der Resultierenden  $F_{R,1}$  zusammengefasst. Auch die beiden Rollen selbst werden wegen der konstant gleichmäßigen Kraftverteilung vereinfacht als eine dargestellt. Der Kraftangriffspunkt der Zugfedern liegt auf Seite der Vorrichtung durch die entsprechenden Bauteile (11) 8 mm über der Platte. Dies bewirkt ein Momentengleichgewicht des auf den Druckfedern gelagerten Aufbaus um den in Abbildung 3-14 gezeigten Rollenauflegepunkt A in der Schienenkurve mit dem kleinsten Radius unter Erfüllung der Bedingung  $F_{D,1} = F_{D,2}$ . Dort ist die Zugfederkraft am größten. Die dem Magnetband zuzuwendende Seite des IMAX-Sensors steht hierbei parallel zum Magnetband. In allen anderen Bereichen der Führungsschiene stehen

die z-Komponenten der Zugfederkräfte zu den Kräften in x-Richtung in einem größeren Verhältnis. In diesen Abschnitten verhindert die Selbsthemmung der senkrechten Rolle weitgehend eine Verkippung der Platte. Im Gegensatz zu den vorherigen Varianten kann diese, bis auf die verwendeten Normteile, komplett additiv gefertigt werden.



**Abb. 3–16: Kräfte-Skizze - Variante 3**

Tabelle 3-3 vergleicht aufwands- und kostenbestimmende Parameter der verschiedenen Varianten. Nachfolgend werden deren Vor- und Nachteile in Tabelle 3-4 zusammengefasst aufgelistet.

**Tab. 3–3: Morphologischer Kasten**

Parameter	Variante 1	Variante 2	Variante 3
Gesamtanzahl der Teile	46	46	70
Selbst herzustellende Teile	7	6	18
Fertigungsverfahren	Sägen, Bohren, Gewinde schneiden, 3D-Druck	Sägen, Bohren, Gewinde schneiden, 3D-Druck	3D-Druck
Zukaufteile	1xZugfeder	2xSchiene, 2xGleitelement, 2xDruckfeder, 2xZugfeder	2xSchiene, 4xGleitelement, 4xDruckfeder, 2xZugfeder
Kosten durch Zukaufteile	2 €	16 €	22 €



**Tab. 3–4: Bewertung der IMAX-Positionierungsvorrichtungen**

	Pro	Kontra
<b>Var. 1</b>	+ Wenig Teile + Gute Zugänglichkeit + Leichte Montierbarkeit	- Mehrere Arbeitsschritte notwendig - Sehr groß
<b>Var. 2</b>	+ Wenig Teile + Relativ kompakt	- Mehrere Arbeitsschritte notwendig - Schlechte Montierbarkeit - Standardrolle passt nicht
<b>Var. 3</b>	+ Komplett additiv fertigbar + Sehr kompakt + Ersatzteilherstellung aufwandsarm	- Viele Teile - Schlechte Montierbarkeit - Hohe Systemkomplexität

Durch die Vielzahl an Teilen weist die dritte Variante die höchste Systemkomplexität auf. Dennoch fällt nach Abwägung der Vor- und Nachteile die Wahl auf die dritte Variante zur Sensorpositionierung. Der ausschlaggebende Grund hierfür ist die additive Fertigbarkeit der Vorrichtung. Diese ermöglicht eine aufwandsarme und kostengünstige Produktion der Einzelteile. Bei Variante 1 und 2 sind zusätzliche Bearbeitungsschritte nötig. Darüber hinaus weist die Vorrichtung auch eine kompaktere Bauform als die anderen zwei Varianten auf und nimmt somit weniger Bauraum im Simulator in Anspruch. Die Mehrkosten fallen relativ gering aus und werden aufgrund der Vorteile in Kauf genommen.

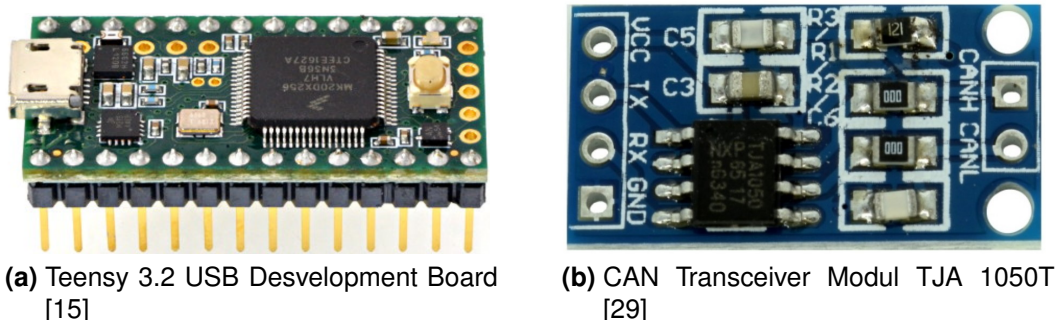
### 3.2.3 Auswahl und Aufbau des Schnittstellensystems

Abbildung 3-14 gibt einen Überblick über die unbenutzten seriellen Anschlüsse des CompactRIO-Systems. Neben einem USB-Port, der nur mit Speichermedien kommunizieren kann, besitzt es einen RS-232-Port und einen freien Slot im Chassis [27].

**Abb. 3–17: Anschlussmöglichkeiten des CompactRIO 9014-Systems**

Den geringsten Implementierungsaufwand würde das in der Einleitung bereits vorgestellte CANopen-Modul von National Instruments für das CompactRIO-System erfordern. Da der Preis jedoch die in **T.2** gesetzte Grenze sprengt, wird diese Möglichkeit ausgeschlossen. Durch Anforderung **T.7**, welche eine Datenausgabe in LabVIEW verlangt, ist auch der Gebrauch von kommerziellen CAN-Schnittstellen mit herstellereigener Software hinfällig. So ist es auch der CAN-Schnittstelle „CPC-USB/ARM7-GTI“, welches dem Lehrstuhl für dieses Projekt vom Unternehmen EMS Dr. Thomas Wünsche zur Verfügung gestellt wurde, nicht möglich **T.7** zu erfüllen. Da es sich bei den vom Nutzer der Anlage zu sendenden Daten nur um NMT-Nachrichten handelt (**T.8**), kann auf eine volle CANopen-Implementation der Schnittstelle verzichtet werden. Es reicht aus, wenn diese die CAN-Funktionalität beherrscht und die NMT-Funktionalität über zusätzliche Programmierung hinzugefügt wird. Zusätzlich ist die Microcontroller-Lösung am günstigsten. Somit bietet diese die geeignetste Schnittstellenlösung.

Wegen positiven Berichten zu gut dokumentierten CAN-Bibliotheken bei Nutzung des Teensy USB Development Boards, fällt die Wahl auf den Teensy 3.2 [15], welcher 21 € kostet. Zusätzlich zu dem 32 Bit Arduino-kompatiblen Microcontroller wird ein High-speed CAN-Transceiver benötigt, der eine Kommunikation zwischen dem Teensy und dem CANopen-Netz ermöglicht. Dieser wandelt die Signalpegel des Teensys in das differentielle CAN-Format um. Dessen Auswahl richtet sich nach der CAN-Bibliothek, da unterschiedliche Bibliotheken verschiedene Transceiver unterstützen. Die FlexCAN-Bibliothek von Collin Kidder weist eine gute Dokumentation auf und wurde deshalb für die Implementation ausgewählt. Dabei wird der Transceiver „TJA 1050T“ von Philips [16] genutzt, dessen Einsatzfähigkeit in Verbindung mit der Bibliothek erfolgreich getestet wurde. Dieser kostet circa 3 €. Die Verbindung der beiden Hardware-Komponenten soll über ein Adafruit Perma-Proto Half-sized Breadboard [28] stattfinden, welches preislich bei etwa 5 € liegt. Der Transceiver benötigt eine externe Spannungsversorgung von 5 V, welche auch für den Teensy genutzt werden kann. Beide Komponenten sind in Abbildung 3-15 dargestellt.



(a) Teensy 3.2 USB Development Board [15]

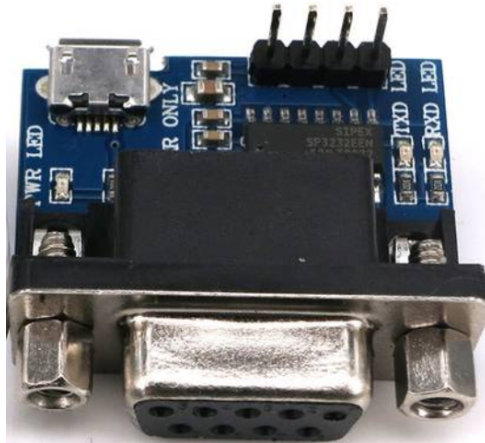
(b) CAN Transceiver Modul TJA 1050T [29]

**Abb. 3–18: Schnittstellen-Komponenten der Microcontroller-Lösung**

Da das CompactRIO-9014 sich gut zugänglich im Schaltschrank neben dem Schaltnetzteil befindet und eine direkte und schnelle LabVIEW-Ausgabe ermöglicht, wird dessen RS-232 Port für die Kommunikation mit der Schnittstelle gewählt, welcher mit 115,2kbit/s eine ausreichende Baudrate besitzt. Da dieser allerdings andere Span-



nungspegel erwartet als das Board ausgibt, ist für eine funktionierende Datenübertragung zusätzlich ein 4 € teurer RS-232-Signalwandler nötig [30]. Dieser ist in Abbildung 3-16 dargestellt.



**Abb. 3–19: RS-232-Transceiver**

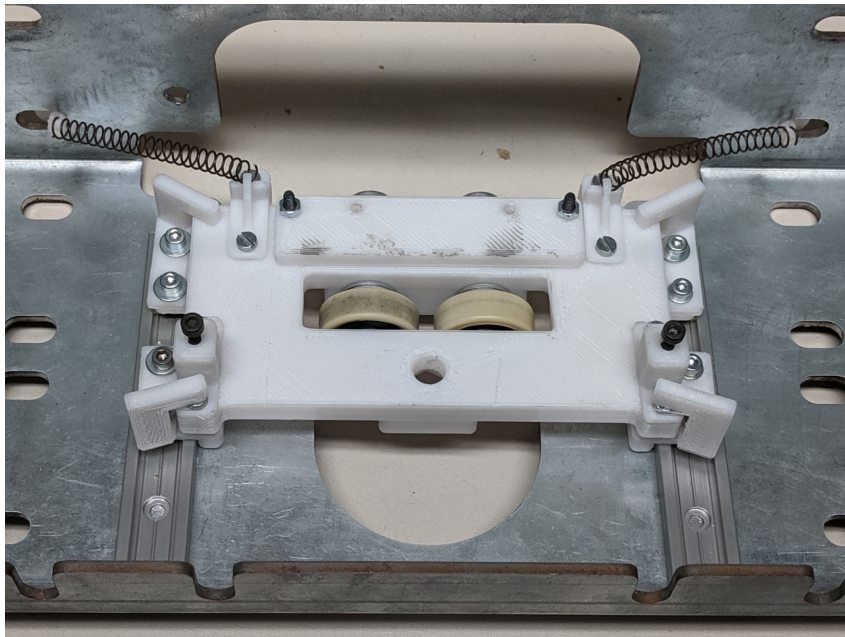
Obwohl die Konfiguration des CANopen-Busses auch über dieses Setup durch entsprechende Programmierung des Teensys erledigt werden kann, bietet das EMS Interface dafür das effizientere Mittel. Das mitgelieferte Programm des Herstellers ermöglicht in Verbindung mit der CAN-Schnittstelle einfaches Versenden von Nachrichten und eine History-Funktion, mit deren Hilfe versandte und empfangene Nachrichten überprüft werden können. Somit wird das CANopen-System über die EMS Schnittstelle konfiguriert, während im operationellen Betrieb die Microcontrollerlösung zum Einsatz kommen soll.

## 4 Implementierung

### 4.1 Montage des Magnetbands und der Sensorpositionierungsvorrichtung

Das Magnetband soll an die Unterkante von Funktionsfläche 3 angeklebt werden. Für den Kraftschluss zwischen dem Stahlband und der Aluminium-Profilschiene sorgt ein Hochleistungsklebeband der Firma 3M. Dabei handelt es sich um das VHB Klebeband 4952, welches leistungsstarke und dauerhafte Haltbarkeit verspricht und für Metallverbindungen besonders gut geeignet ist. Zusätzlich beseitigt es durch seine Dicke von 11 mm die in Kapitel 3.2.1 gezeigte Unebenheit und sorgt somit für eine glatte Oberfläche. Vor Anbringen des Magnetbands wird die Oberfläche der Aluminium-Führung und die Rückseite des Magnetbands mit Isopropanol gereinigt, um eine optimale Haftung des Klebebands zu gewährleisten. Außerdem wird bei beiden Simulatoren der Metallkorb, auf dem die Vorrichtung zur Sensorpositionierung angebracht wird, abmontiert. Im Folgenden wird das Magnetband montiert. Mittig hinter der tragenden Säule der C-Schiene werden Anfang und Ende des Magnetbands angebracht. Da der Klebstoff des VHB-Bands druckempfindlich ist, wird es beim Anbringen fest mit der Hand angepresst. Um das Abschälen des Magnetbands an den Enden zu verhindern, werden diese mit Klebeband an die Schiene gepresst. Die dafür vom Hersteller vorgesehenen Endkappen können nicht verwendet werden, da diese beim Überfahren den Sensor beschädigen würden.

Die selbst herzustellenden Teile der Positionierungsvorrichtung werden aus Polylactid (PLA) gedruckt. Dabei muss festgestellt werden, dass die M3 Gewinde, welche der Befestigung des Sensors dienen, aufgrund von mangelnder Genauigkeit des Druckers in zwei zusätzlichen Arbeitsschritten nachgebohrt und nachgeschnitten werden müssen, um ihren Zweck zu erfüllen. Abbildung 4-1 zeigt die Vorrichtung in der Montage. Wegen der schlechten Erreichbarkeit der Einzelteile im eingebauten Zustand des Metallkorbs, wird die Montage der Vorrichtung auf dem Metallkorb vor dessen Wiedereinbau in den Simulator vorgenommen. Außerdem werden die auf den Funktionsflächen 3 und 4 laufenden Rollenkonstruktionen, welche den Korb auf der Aluminiumschiene führen, durch eine Hülse um 8 mm verlängert, sodass diese in sicherem Abstand über dem Magnetband laufen. Die vollständig implementierte Vorrichtung wird in Abbildung 4-2 dargestellt.



**Abb. 4–1: Montage der Sensorpositionierungsvorrichtung**



**(a) Funktionsfläche 4**



**(b) Funktionsfläche 3**

**Abb. 4–2: Vollständig implementierte Sensorpositionierungsvorrichtung**

## 4.2 Konfiguration des CANopen-Systems

### 4.2.1 Konfiguration der Encoder

#### 4.2.1.1 Änderung der Knotennummer mithilfe der Layer Setting Services

Alle Sensoren sind bereits auf 250 kbit/s voreingestellt. Bevor die CAN-Kabel der beiden Simulatoren verbunden werden können, muss sichergestellt werden, dass alle vier Encoder unterschiedliche Knotennummern aufweisen. Dazu wird das CAN-USB-Gerät von EMS über einen 9-poligen Sub-D Adapter nacheinander an die offenen Kabelenden angeschlossen. Dies wird in Abbildung 4-3 veranschaulicht. Die Pin-Belegung ist wie in CiA 303-1 empfohlen [24, Seite 9] und wird in Tabelle 4-1 dargestellt. Der

Ausgang des Geräts wird per USB mit dem PC verbunden, auf welchem die Bus-Monitoring Software des Herstellers installiert wurde. Darin muss ein Channel mit einer Baudrate von 250 kbit/s geöffnet werden, um mit den Sensoren kommunizieren zu können.



**Abb. 4–3: CAN-USB-Schnittstelle mit Anschlüssen**

**Tab. 4–1: Pin-Belegung des Sub-D Anschlusses bei der Buskonfiguration**

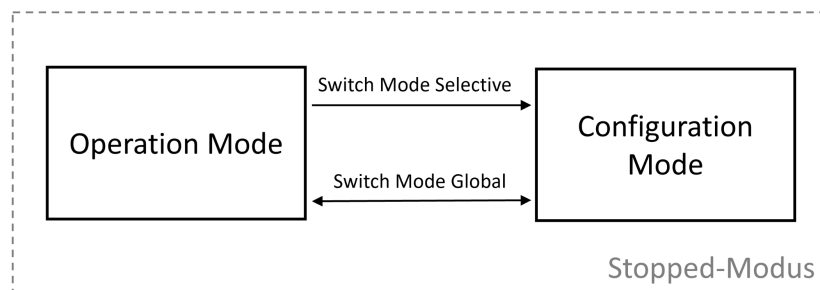
Pin-Nummer	Leitung
2	CAN_L
5	Schirmung
7	CAN_H

Folgend werden alle CANopen-Slaves durch die entsprechende im Unterkapitel 2.2.6 beschriebene NMT-Nachricht vom Pre-Operational- in den Operational-Zustand geschaltet. Da keine Probleme auftreten und alle Nachrichten der Encoder in der CAN Suite ausgegeben werden, können die unterschiedlichen Leitungsquerschnitte des CAN-Netzwerks als unkritisch für die Kommunikation bewertet werden. Die voreingestellten Knotennummern sind Tabelle 4-2 zu entnehmen.

**Tab. 4–2: Knotennummern der Encoder vor und nach (in Klammern) der Konfiguration**

	Sonnensimulator	Erdalbedosimulator
<b>IMAX</b>	4 (3)	4
<b>SRH520CN</b>	32	34

Beide IMAX-Sensoren besitzen die gleiche Knotennummer. Folgend muss diese bei einem der beiden Encoder geändert werden, um alle Identifier am Busnetzwerk eindeutig zuordnen zu können. Somit soll die Knotennummer des IMAX-Encoders des Sonnensimulators mithilfe der LSS-Dienste von 4 auf 3 geändert werden. Dazu wird die Schnittstelle an das Busnetzwerk der Sonne angeschlossen, um die Busteilnehmer zunächst in den Stopped-Zustand zu schalten. Nun wird vom Operation-Modus in den Configuration-Modus übergegangen. Das kann, wie in Abbildung 4-4 gezeigt, über zwei Befehle erfolgen. Switch-Mode-Selektive setzt einzelne Knoten in den Konfigurationsmodus, während Switch-Mode-Global alle Busteilnehmer anspricht. Bei ersterem wird die gesamte LSS-Adresse benötigt. Diese besteht aus Hersteller-ID, Produkt-Code, Revisions-Nummer und Seriennummer oder entspricht somit dem Identity-Objekt, welches unter dem Index 0023h im Objektverzeichnis zu finden ist [22, S. 171], [22, S. 83]. Das Zurücksetzen in den Operation-Modus funktioniert nur über den Switch-Mode-Global.

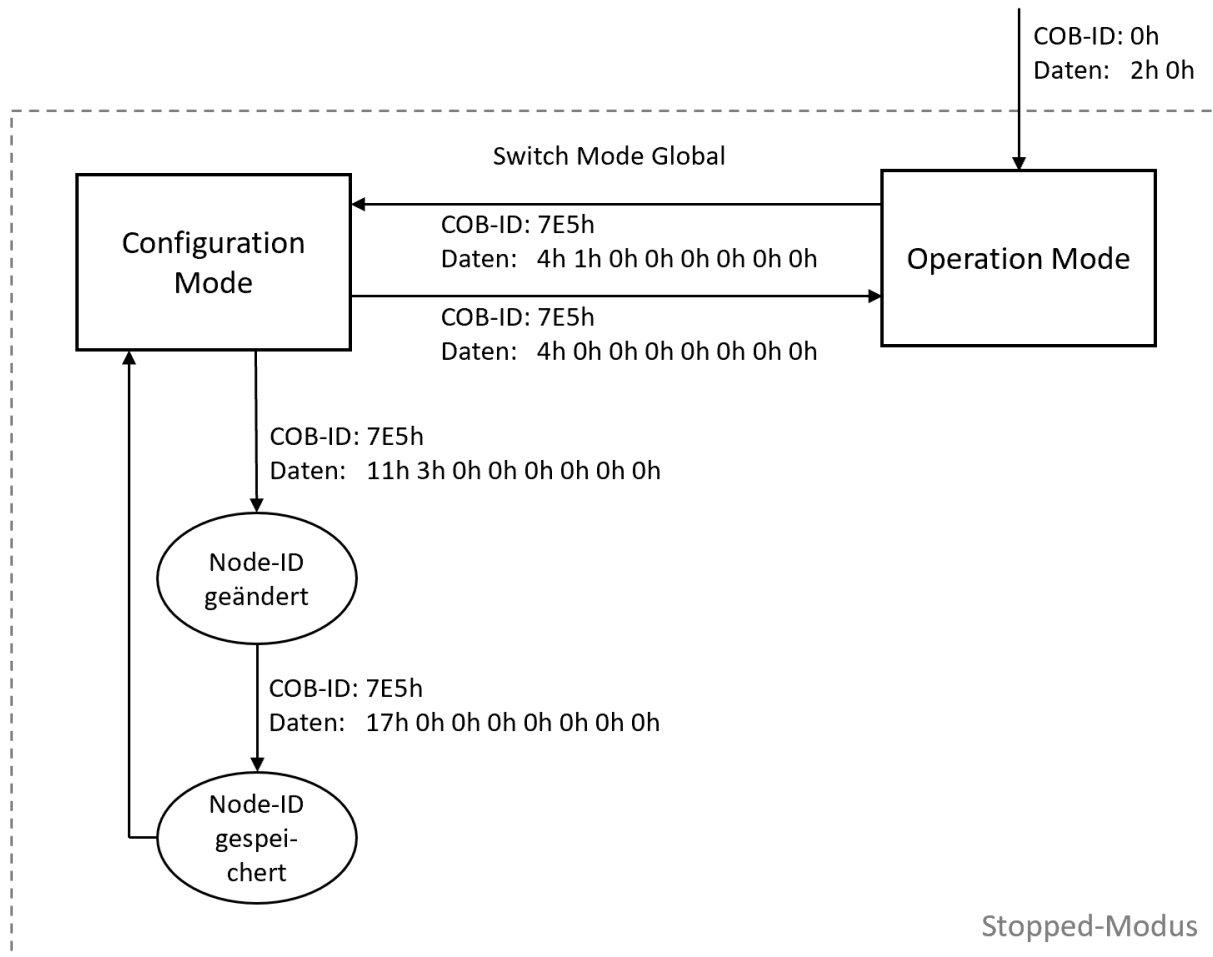


**Abb. 4–4: Übergang vom Operation Mode in den Configuration Mode**

Da im Drehencoder die LSS-Dienste nicht implementiert sind und somit Knotennummer und Baudrate nicht verändert werden können, wird der Switch-Mode-Global verwendet. Das erste übertragene Byte steht für Switch-Mode-Global, während das zweite mit dem Wert 0 in den Operation-Modus und mit 1 in den Configuration-Modus wechselt. Die restlichen 6 Byte sind von CiA reserviert und somit gleich Null zu setzen. Im nächsten Schritt kann die Knotennummer des IMAX-Sensors verändert werden. Das erste Byte spezifiziert dabei wieder die Aufgabe der Nachricht, während im zweiten Byte die neue Knotennummer einzutragen ist. Diese soll den Wert 3 annehmen. Die restlichen Byte sind ebenfalls reserviert. Daraufhin antwortet der Knoten mit einer Nachricht, welche die Änderung bestätigt. Die erfolgreiche Speicherung der Konfiguration wird ebenfalls vom LSS-Slave rückgemeldet. Um die Konfiguration abzuschließen, wird zurück in den Operation-Modus gewechselt. Bei dem gesamten Vorgehen wird



sich direkt an das CANopen Layer Setting Services Protokoll gehalten [31]. Abbildung 4-5 zeigt die einzelnen Schritte und die gesendeten Nachrichten. Die einzelnen Byte werden dabei durch ein Leerzeichen getrennt. Das h kennzeichnet die Zahlen als hexadezimal.



**Abb. 4–5: Vorgehen beim Ändern der Knotennummer des IMAX-Sensors**

#### 4.2.1.2 Betriebsart der Sensoren

Wie bereits in 2.1.7 beschrieben, kann der IMAX-Encoder nur im Cyclic-Modus betrieben werden. Somit wird aufgrund der Übersichtlichkeit für alle Sensoren die asynchrone Datenübertragung gewählt, auf welche diese bereits voreingestellt sind. Während der lineare Encoder alle 5 ms ein PDO sendet, überträgt der Drehwinkelsensor seine Positionsdaten alle 10 ms. Das Intervall der rotatorischen Sensoren ist somit zu groß, um die zeitliche Anforderung **T.5** einzuhalten. Wenn dieser seine Daten kurz vor der Aktualisierung der LabVIEW-Schleife sendet, diese jedoch aufgrund der Zeitverzögerung bei der Datenübertragung nach dieser Aktualisierung eintreffen, werden die Positionsdaten erst beim erneuten Durchlaufen der Schleife registriert. Das hat zur Folge, dass

der Messzeitpunkt der Daten bereits über 10 ms zurückliegt. Somit sollen die rotatorischen Encoder ebenfalls auf eine asynchrone Übertragung von 5 ms eingestellt werden. Dazu muss das Transmission Process Data Object (TPDO) 1 konfiguriert werden. Dieses steht für das erste von zwei aktiven TPDOs, welches im Objektverzeichnis unter dem Index 1800h konfiguriert werden kann. Tabelle 4-3 zeigt den Aufbau dieses Objekts.

**Tab. 4–3: Objekt-Beschreibung des TPDO 1 [32]**

Index	Subindex	Name	Datentyp	Zugriff	Voreingestellter Wert
1800h	00h	Höchster unterstützter Subindex	U8	RO	5
	01h	COB-ID	U32	RW	180h + Knoten-ID
	02h	Übertragungstyp	U8	RW	FEh
	05h	Event Timer	U16	RW	000Ah

Beim Übertragungstyp kann zwischen synchron und asynchron gewählt werden. FEh steht für die asynchrone Übertragung, während 01h-FDh den synchronen Modus kennzeichnet [32, S. 18]. Im synchronen Betrieb legt der gewählte Wert fest, nach dem wievielten Empfang des Sync-Telegramms die Positionsdaten gesendet werden. Da dieser Subindex den gewünschten Wert beinhaltet, ist dessen Änderung nicht erforderlich. Über den in Subindex 05h vorzufindenden Event Timer lässt sich das zyklische Intervall festlegen, nach dessen Ablauf der Knoten seine Nutzdaten sendet. Der Event Timer ist ein 1 Byte großer Wert. Um diesen zu ändern, kommt folgend der Expedited-SDO-Transfer zum Einsatz, welcher auf Objektverzeichniseinträge mit bis zu 4 Byte Inhalt zugreifen kann. Es wird zwischen Lese- und Schreibzugriff unterschieden. Eine SDO-Nachricht hat dabei immer eine Länge von 8 Byte und das Remote-Transmission-Request-Bit ist gleich 0 zu setzen. Der Expedited-SDO-Transfer umfasst zwei Nachrichten, da auf die Nachricht, welche Lese- oder Schreibzugriff fordert, eine Bestätigungsnachricht folgt. Der Identifier eines Requests hat den Wert 600h + Knoten-ID, während die Confirmation-Nachricht einen Identifier von 580h + Knoten-ID aufweist [22, S. 60ff.]. Tabelle 4-4 zeigt den Aufbau der Anforderung des Lesezugriffs auf einen Objektverzeichniseintrag, der zugehörigen Bestätigungsnachricht und darunter die gesendete und die empfangene Nachricht, wodurch der Event Timer des TPDO 1 ausgelesen wird. Der Index und die nach dem Subindex folgenden Nutzdaten werden dabei nach dem „LSB to MSB“-Prinzip übertragen.

**Tab. 4–4: SDO Leseanfrage und Bestätigungsnachricht**

Identifizier	B0	B1	B2	B3	B4	B5	B6	B7
600h + Knoten-ID	CC	Index		Subindex	00h	00h	00h	00h
580h + Knoten-ID	CC	Index		Subindex	Data			
600h + Knoten-ID	40h	00h	18h	05h	00h	00h	00h	00h
580h + Knoten-ID	4Bh	00h	18h	05h	0Ah	00h	00h	00h

Der Command Code (CC) nimmt für einen Lesezugriff den Wert 40h an. Bei der Bestätigungsnachricht gibt dieser neben der Funktion Auskunft über die Anzahl an übertragenen Byte und den Transfertype. Die Funktion ist hier die Antwort auf die Leseanfrage und der Transfertype ist Expedited. Die Zusammenhänge, die in diesem Fall für Command Code und Byteanzahl gelten, werden in Tabelle 4-5 beschrieben.

**Tab. 4–5: Command Code der Bestätigungsnachricht beim Expedited-Transfer [33]**

Command Code	Anzahl der gesendeten Byte
43h	4
47h	3
4Bh	2
4Fh	1

Somit überträgt die Bestätigungsnachricht aus Tabelle 4-4 2 Byte, die sich mit der Objektbeschreibung in Tabelle 4-3 decken. Der Schreibzugriff des Expedited-SDO-Transfers unterscheidet sich vom Lesezugriff im Command Code und den letzten 4 Byte. Tabelle 4-6 zeigt den Aufbau der Nachrichten.

**Tab. 4–6: SDO Schreibzugriff und Bestätigungsnachricht**

Identifizier	B0	B1	B2	B3	B4	B5	B6	B7
580h + Knoten-ID	CC	Index		Subindex	00h	00h	00h	00h
600h + Knoten-ID	2Bh	00h	18h	05h	05h	00h	00h	00h
580h + Knoten-ID	60h	00h	18h	05h	00h	00h	00h	00h

Der Command Code der Schreibanfrage spezifiziert ebenfalls Funktion, Transfertype und die Anzahl der übertragenen Byte. Hierbei ist die Funktion der Schreibzugriff und die Übertragungsart ist ebenfalls Expedited. Der entsprechende Command Code wird abhängig von der zu übertragenden Datenmenge aus Tabelle 4-7 gewählt.



**Tab. 4–7: Command Code für Schreibzugriff beim Expedited-Transfer [33]**

Command Code	Anzahl der gesendeten Byte
23h	4
27h	3
2Bh	2
2Fh	1

Um den Event Timer zu ändern, müssen 2 Byte versandt werden, damit sich die Größe des Eintrags im Objektverzeichnis mit der Anzahl der versandten Daten deckt. Der Command Code der Nachricht ist somit 2Bh. Bei der Bestätigungsnachricht des Schreib-Transfers beträgt dieser 60h. Um die Einstellung zu überprüfen wird der Subindex erneut ausgelesen. Im Fall der Drehencoder wird daraufhin wieder der Eintrag Ah ausgegeben. Aus weiteren Konfigurationsversuchen wird ersichtlich, dass die Event Timer nur auf ein Vielfaches von 10 ms einstellbar sind. Somit stellt 10 ms das kleinste mögliche asynchrone Übertragungsintervall für die rotatorischen Sensoren dar.

#### 4.2.1.3 Überwachung der Busteilnehmer

Die Drehwinkelsensoren legen im Betrieb alle 100 ms eine Heartbeat-Nachricht auf den Bus. Diese umfasst 1 Byte an Daten mit dem der NMT-Status wie in Tabelle 4-8 gekennzeichnet wird, in welchem sich der jeweilige Encoder befindet. Zusätzlich kann bei Ausbleiben dieser Nachricht auch eine Fehlfunktion eines Sensors erkannt werden.

**Tab. 4–8: Heartbeat-Nachricht und NMT-Status [32, S. 10]**

NMT-Status	Daten
Initialisation	0h
Stopped	4h
Operational	5h
Pre-Operational	7Fh

Um die Heartbeat-Nachrichten zum Überwachen der Busteilnehmer nutzen zu können, müssen diese bei den IMAX-Sensoren zunächst aktiviert werden. Das Intervall, in dem diese gesendet werden sollen, wird gleich dem der Drehencoder gewählt. Tabelle 4-9 zeigt die Objektbeschreibung des dafür zu konfigurierenden Objekts der IMAX-Sensoren.

**Tab. 4–9: Objekt-Beschreibung: Producer Heartbeat Time [32, S. 17]**

Index	Subindex	Name	Datentyp	Zugriff	Voreingestellter Wert
1017h	00h	Producer Heartbeat Time	U16	RW	0000h

Da das Objekt eine Größe von 2 Byte hat, wird auch hier der Expedited-SDO-Transfer verwendet. Beide IMAX-Encoder werden mit der in Tabelle 4-10 beschriebenen Nachricht auf das gewünschte Übertragungsintervall der Heartbeat-Nachrichten konfiguriert.

**Tab. 4–10: SDO-Schreibzugriff auf Heartbeat Producer Time**

Identifizier	B0	B1	B2	B3	B4	B5	B6	B7
600h + Knoten-ID	2Bh	17h	10h	00h	64h	00h	00h	00h

Damit die Einstellungen auch nach einem Neustart der Encoder noch vorhanden sind, müssen die Parameter im nicht flüchtigen Speicher gesichert werden. Dies wird durch die in Tabelle 4-11 beschriebene Nachricht erreicht, welche alle Parameter speichert.

**Tab. 4–11: Sichern aller Parameter im nicht flüchtigen Speicher [32, S. 16]**

Identifizier	B0	B1	B2	B3	B4	B5	B6	B7
600h + Knoten-ID	2Bh	10h	10h	01h	73h	61h	76h	65h

#### 4.2.2 Abschlusswiderstände

Da das Busnetzwerk eine Kommunikation mit allen vier Sensoren ermöglichen soll, werden die offenen Kabelenden der beiden CAN-Kabel verbunden. Wie bereits in Kapitel 3 beschrieben, sind die in den IMAX-Encodern integrierten  $120\ \Omega$  Abschlusswiderstände ausreichend zur Terminierung des CAN-Busses, weshalb die beiden zusätzlichen Widerstände entfernt werden. Da der CAN-Transceiver ebenfalls einen  $120\ \Omega$  Widerstand besitzt, wird auch dieser entfernt, bevor er an den Bus angeschlossen wird.

### 4.3 Implementation des Schnittstellensystems

#### 4.3.1 Spannungsversorgung und Stromverbrauch

Zunächst soll die Spannungs- und Stromversorgung von dem Teensy, dem CAN-Transceiver und dem RS-232-Transceiver geklärt werden. Während der RS-232-Anschluss hierfür keine Möglichkeit bietet, stellt der USB-Anschluss des CompactRIO-9014 eine 5 V Spannungsversorgung und einen Strom von 500 mA bereit [27, S. 13]. Tabelle 4-12 zeigt die benötigten Eingangsspannungen und den maximalen Stromverbrauch der Schnittstellen-Komponenten. Da die Eingangsspannung von 5 V für alle Schnittstellenbestandteile genutzt werden kann und deren gesamter maximal benötigter Strom mit

326 mA unter dem vom USB-Anschluss bereitgestellten Strom liegt, eignet sich dieser für das Betreiben der einzelnen Teile.

**Tab. 4–12: Spannungsversorgung und Stromverbrauch der Schnittstellenkomponenten**

Schnittstellenkomponente	Eingangsspannung	maximaler Strom
Teensy 3.2	3,3 V bis 5 V	250mA [34]
CAN-Transceiver	5 V	75 mA [16]
RS-232-Transceiver	3 V bis 5,5 V	1 mA [35]
	<b>Summe</b>	326 mA

#### 4.3.2 Zusammenfügen der Schnittstellen-Komponenten

Abbildung 4-6 zeigt die Pin-Belegung des Teensy 3.2 Boards. Der Teensy, der CAN-Transceiver und die Kabeldrähte werden mit dem Breadboard verlötet. Zusätzlich wird eine Light Emitting Diode (LED) integriert, welche die Stromversorgung der Schnittstelle nachprüfbar macht. Dieser muss ein 220  $\Omega$  Widerstand vorgeschaltet werden. Die Pins 3 und 4 sorgen für die Kommunikation des Teensys mit dem CAN-Transceiver, welcher direkt an eine Stichleitung des Busnetzwerks angeschlossen ist. Die serielle Datenübertragung findet über die Pins 0 und 1 statt.

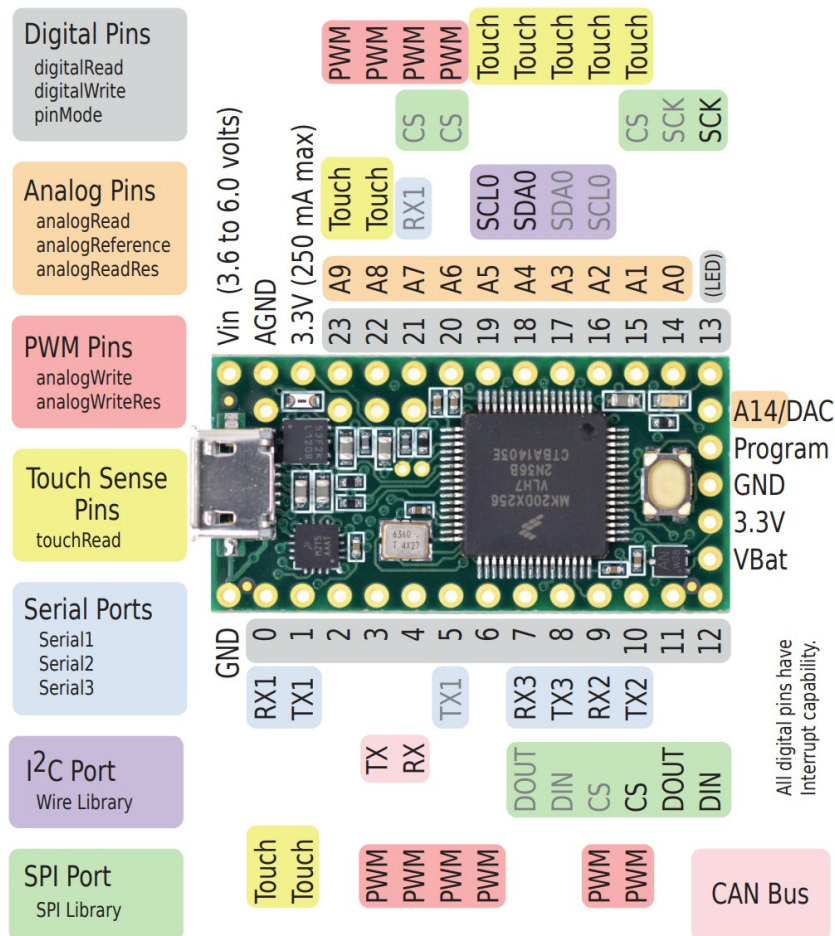


Abb. 4–6: Teensy 3.2 mit Pin-Belegung [36]

Durch eine in die Steckplatine eingelötete Pinheadreihe und ein entsprechendes Kabel wird die Stromversorgung und die Kommunikation der Breadboard-Komponenten mit dem RS-232-Transceiver erreicht. Dieser ist über seinen Micro-USB-Anschluss und ein entsprechendes Kabel an dem USB-Port und direkt an dem RS-232-Anschluss des CompactRIO angeschlossen. Während die Verbindungen der einzelnen Schnittstellenteile dem Schaltplan in Abbildung 4-7 zu entnehmen sind, zeigt Abbildung 4-8 die vollständig implementierte Schnittstelle in der Anlage.

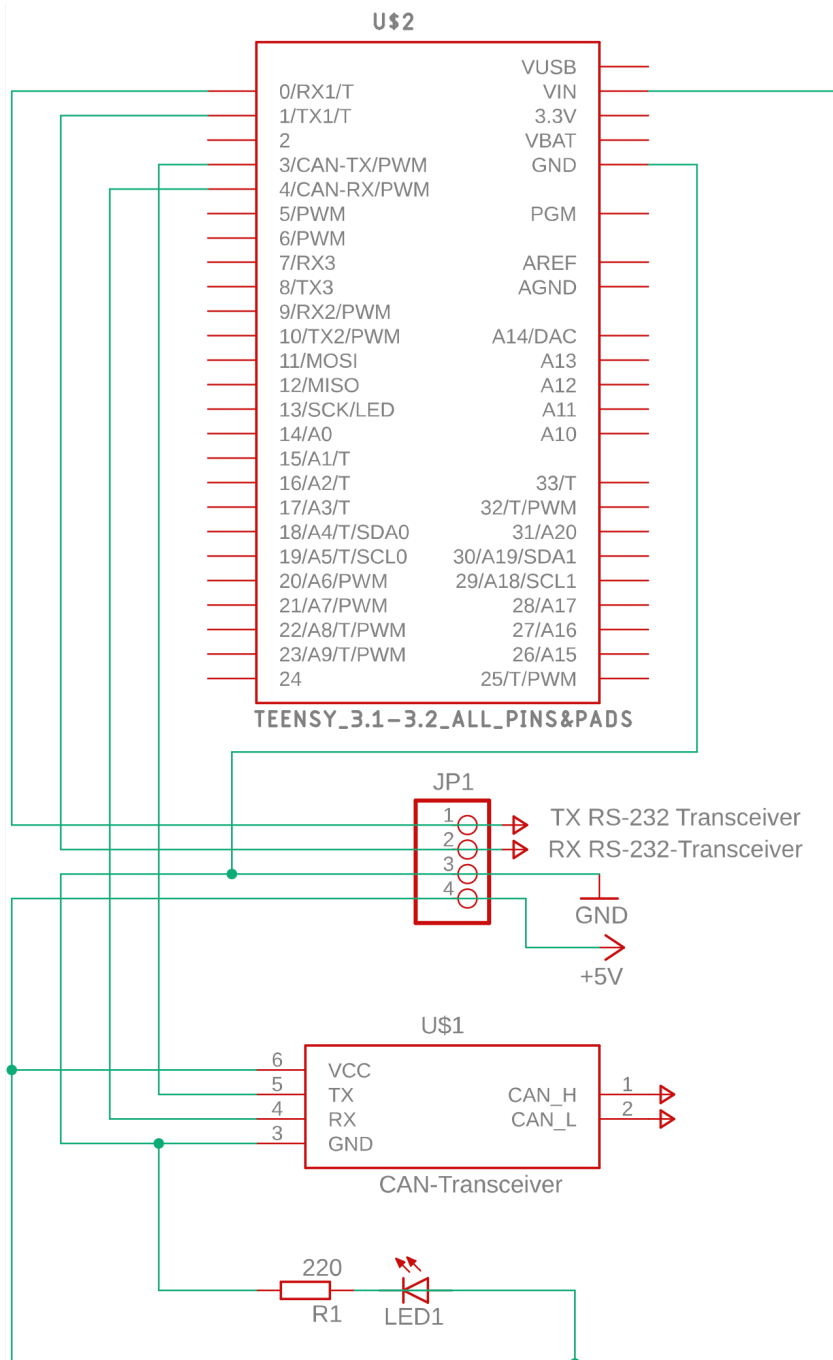
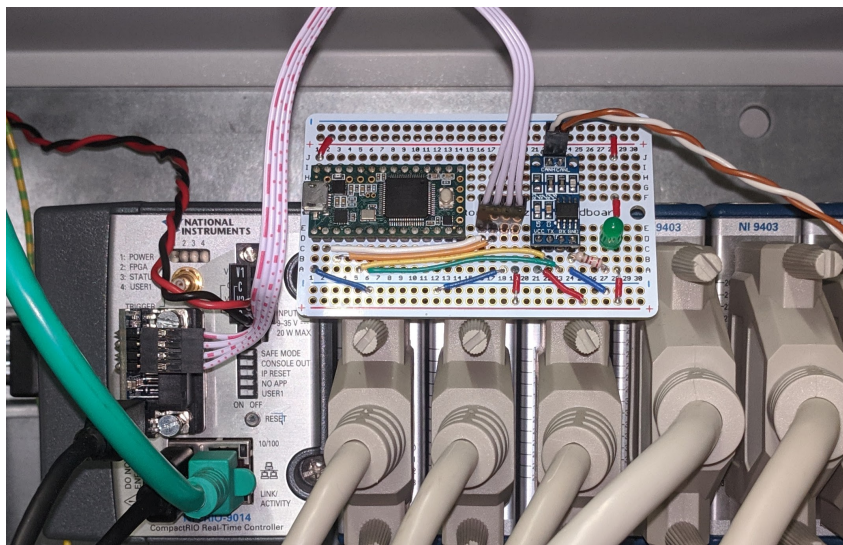


Abb. 4-7: Schaltplan



**Abb. 4–8: Betriebsbereite Microcontroller-Schnittstelle**

### 4.3.3 Entwicklungsumgebung Atom

Bei Atom handelt es sich um einen Open-Source-Texteditor, der in dieser Arbeit unter Windows genutzt wird. Dieser bietet einige Vorteile. Da Atom vom Projekt-Hosting-Dienst GitHub entwickelt wird, lässt es sich leicht mit Projekten des Onlinedienstes arbeiten. Die Syntaxhervorhebung sorgt für eine gute Übersichtlichkeit beim Programmieren [37]. Zusätzlich wird durch den Paketmanager die integrierte Entwicklungsumgebung PlatformIO genutzt. Dieses Open-Source-Paket bietet Unterstützung für verschiedene Microcontroller und Frameworks, so auch für Teensy-Boards und das Arduino-Framework. Dabei erlaubt der Library Manager eine simple Bibliothek-Einbindung in eigene Projekte. Auch intelligente C/ C++ Code-Vervollständigung, ein integriertes Terminal, Debugger und ein Serial Port Monitor sind neben anderen Funktionen in PlatformIO enthalten [38]. Bevor mit der Programmierung begonnen werden kann, wird das Teensy 3.1/ 3.2 Paket installiert und die FlexCAN Bibliothek von Collin Kidder in das Projekt eingebunden. Das nachfolgende Programm wird im dafür angelegten Projekt mit dem Namen „RACOON CAN Encoders“ in Gitlab hinterlegt [39].

### 4.3.4 Wichtige Elemente der FlexCAN-Bibliothek

Die Bibliothek enthält bereits vordefinierte CAN-Bus-Objekte. Beim Teensy 3.2 kommt das Objekt „Can0“ zum Einsatz. Tabelle 4-13 erklärt die zugehörigen, im folgenden Programm genutzten Funktionen.

**Tab. 4–13: Funktionen des Can0-Objekts [14]**

Funktion	Beschreibung
Can0.begin()	Startet die CAN-Kommunikation
Can0.end()	Beendet die CAN-Kommunikation
Can0.write(Nachricht)	Sendet eine CAN-Nachricht; gibt 0 zurück wenn kein Puffer zum Senden verfügbar ist
Can0.read(Nachricht)	Speichert empfangene Nachricht unter „message“ ab; gibt 1 zurück wenn eine CAN-Nachricht verfügbar ist; gibt 0 zurück wenn keine CAN-Nachricht verfügbar ist
Can0.available()	Gibt die Anzahl an Nachrichten im Empfangspuffer zurück

Die CAN\_message\_t-Struktur beschreibt den Aufbau einer CAN-Nachricht. Werte für Identifier und Datenbyte werden dabei standardmäßig in dezimal eingetragen und ausgegeben. Die einzelnen Bestandteile werden in Tabelle 4-14 erläutert.

**Tab. 4–14: Beschreibung der CAN\_message\_t-Struktur [14]**

Parameter	Beschreibung	Relevante erlaubte Werte
id	11- oder 29-Bit-ID	11-Bit: 0 bis 4095
flags.extended	Unterscheidet Standard- und Extended-IDs	0 für 11-Bit-IDs
flags.remote	Spezifiziert, ob ein Remote Transmission Request (RTR) vorliegt	0 für Nachricht ohne RTR
flags.overrun	Puffer-Überlauf	0 für kein Überlauf erkannt; 1 für Überlauf erkannt
len	Anzahl der Datenbyte	0 bis 8
timestamp	Hardware-basierter Zeitstempel	Gibt den Zeitpunkt aus, zu welchem die Nachricht empfangen wurde
buf	Datenbyte der Nachricht	Alle Werte die sich im Umfang der 0-8 Byte darstellen lassen

#### 4.3.5 Programmierung des Mikrocontrollers

Das Arduino-Programm weist zwei Teile auf. Im ersten wird eine setup()-Funktion durchlaufen. Dieser Abschnitt der Programmierung wird nach Einschalten des Mikrocontrollers nur einmal durchlaufen. Daraufhin wird in die loop()-Funktion gewechselt. Dieser Teil der Programmierung wird in einer Endlosschleife ausgeführt [40].

Die setup()-Funktion wird genutzt, um die serielle Kommunikation des Teensy mit dem Steuerungs-PC und die Kommunikation mit dem CAN-Bus zu starten. Die Baudrate für die CAN-Kommunikation wird auf 250 kbit/s eingestellt, während die serielle



Datenübertragung aufgrund der Limitierung des RS-232 Ports des CompactRIO mit 115,2 kbit/s gestartet wird. Mithilfe der Identifier der Heartbeat-Nachrichten wird überprüft, ob sich alle Encoder melden und somit ihre Einsatzbereitschaft signalisieren. Wenn dies der Fall ist, wird in den Operational-Zustand geschaltet, die Systemzeit über die millis()-Funktion gespeichert und die loop()-Funktion gestartet. Wenn sich einer oder mehrere Sensoren während des Setup-Prozesses nicht melden, wird eine Nachricht ausgegeben, welche die nicht sendenden Encoder spezifiziert. Die Identifier der Heartbeat-Nachrichten erhält man durch Addition von 700h und der Knotennummer des jeweiligen Sensors. Die abgefragten Dezimalwerte sind Tabelle 4-15 zu entnehmen.

**Tab. 4–15: Identifier der Heartbeat-Nachrichten**

Simulator	Sensor	Identifier in hexadezimal	Identifier in dezimal
Sonne	IMAX	703h	1795
	SRH520CN	720h	1824
Erde	IMAX	704h	1796
	SRH520CN	722h	1826

Die loop()-Funktion beginnt mit einer while-Schleife, welche durchlaufen wird, solange neue CAN-Nachrichten im Teensy-Empfangspuffer liegen. Diese ordnet den empfangenen Nachrichten anhand der Identifier die zugehörigen Encoder zu und speichert die Nachrichten ab. Trifft ein neues Prozessdatenobjekt eines Encoders ein, wird dessen veraltete Nutzdaten-Nachricht überschrieben. Die PDO-Identifier berechnen sich aus 180h + Knotennummer. Tabelle 4-16 ordnet den Sensoren ihre PDO-Identifier zu. Da alle CANopen-Knoten im normalen Betrieb im selben Zustand operieren und die Heartbeat-Nachrichten somit den selben Inhalt aufweisen, überschreibt jede neue vom Teensy gelesene Heartbeat-Nachricht die letzte gespeicherte.

**Tab. 4–16: Identifier der Prozessdatenobjekte**

Simulator	Sensor	Identifier in hexadezimal	Identifier in dezimal
Sonne	IMAX	183h	387
	SRH520CN	1A0h	416
Erde	IMAX	184h	388
	SRH520CN	1A2h	418

Wenn sich keine CAN-Nachricht im Empfangs-Puffer des Teensy befindet, wird die aktuelle Systemzeit mit der gespeicherten verglichen. Wenn die Differenz größer als 1 s ist, wird die gespeicherte Systemzeit mit der aktuellen überschrieben. Zusätzlich werden die Positionen und Geschwindigkeiten der IMAX-Sensoren, die Positionen der Drehwinkelsensoren und der Zustand des CANopen-Systems ausgegeben. Das Ausgabeintervall von 1 s wurde gewählt, um dem Nutzer des Messsystems das Auslesen



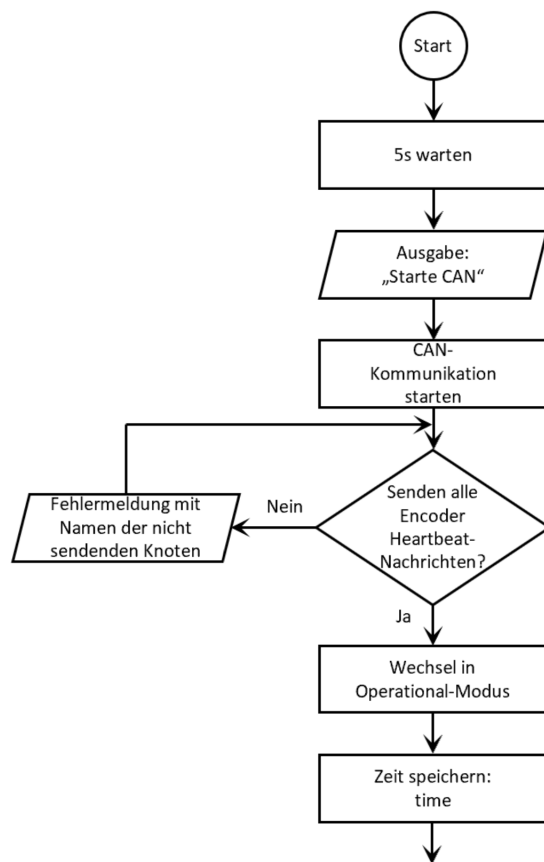
der Positionsdaten zu ermöglichen.

Im letzten Abschnitt der loop()-Funktion wird überprüft, ob eine Eingabe des Nutzers der Anlage vorliegt. Dieser kann über das Senden der Werte 1 bis 5 die CANopen-Statusmaschine kontrollieren. Somit sind die Anforderungen **T.7** und **T.8** erfüllt. Der Zusammenhang zwischen Zustand, in welchen durch Senden des entsprechenden NMT-Befehls geschaltet wird und dem Eingabewert zeigt Tabelle 4-17. Differiert dieser von den aufgeführten Werten, wird die nutzlose Eingabe rückgemeldet.

**Tab. 4–17: Zusammenhang zwischen Nutzereingaben und Zustandsübergängen**

Eingabe	Zustand
1	Operational
2	Stopped
3	Pre-Operational
4	Knoten zurücksetzen
5	Kommunikation zurücksetzen
sonst	Ausgabe: „Falsche Eingabe!“

Abbildung 4-9 und 4-10 zeigen den nach DIN 66001 entworfenen Programmablaufplan. Dabei werden setup()-Funktion und loop()-Funktion separat dargestellt. Das LabVIEW-Blockdiagramm, welches zur Ausgabe der Sensorwerte erstellt wurde, weist den Namen „SerialMonitor.vi“ auf und ist im Anhang zu finden.



**Abb. 4–9: Programmablaufplan: setup()-Funktion**

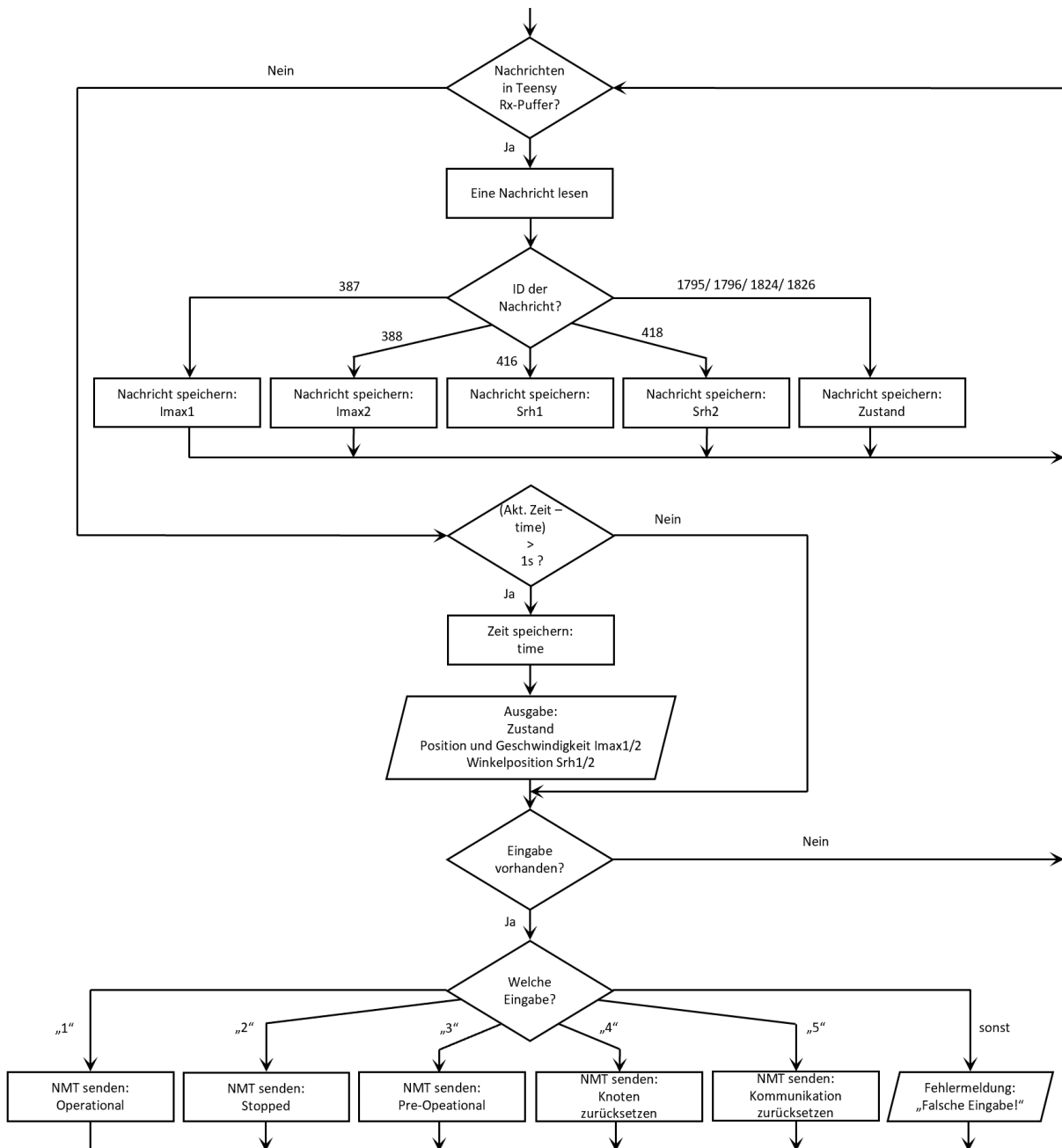


Abb. 4–10: Programmablaufplan: loop()-Funktion

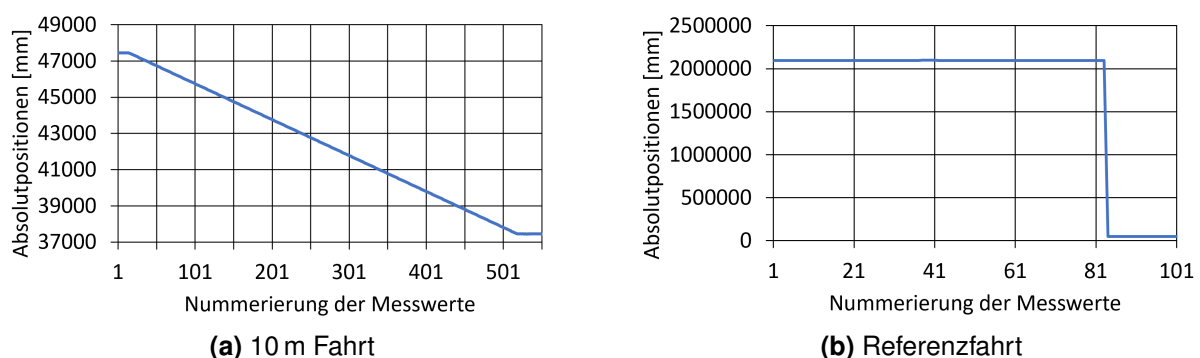
## 5 Ergebnisse und Verifikation des Systems

### 5.1 Ergebnisse

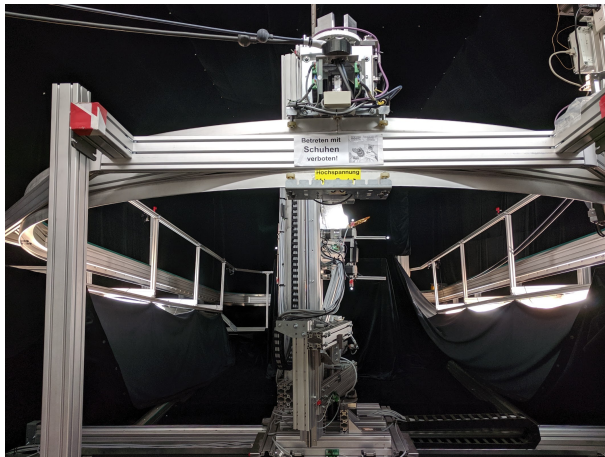
Um die Funktion der Encoder zu verifizieren, wurden mehrere Testfahrten durchgeführt. Die Simulatoren werden dabei über die Eingabe einer Anzahl an Motorschritten verfahren. Zusätzlich werden betriebsrelevante Positionen auf der Führungsschiene angefahren und deren Absolutwerte dokumentiert.

#### 5.1.1 Testfahrten - Lineare Encoder

Der geschätzte Umrechnungsfaktor von Motorschritten in Meter beträgt 160000 für Fahrten entlang der Schiene. Durch eine erste Testfahrt des Sonnensimulators mit 1600000 Motorschritten wird ein genauere Umrechnungsfaktor von Meter in Schritte berechnet. Der Sensor misst 10068 mm. Der Umrechnungsfaktor wird somit auf 158919 Schritte pro Meter aktualisiert. Die zweite Fahrt soll den Umrechnungsfaktor validieren. Somit wird der Simulator 1589190 Schritte verfahren. Da die IMAX-Messwerte einen Weg von 10002 mm anzeigen, ist der Umrechnungsfaktor relativ akkurat. Abbildung 5-1 (a) zeigt die zugehörigen gemessenen Absolutpositionen. Zusätzlich werden zwei Referenzpositionen auf der Symmetrieebene der Schiene festgelegt, von welchen eine vor Betriebsbeginn angefahren werden soll, um einen definierten Startpunkt zu schaffen. Dies dient einer genaueren Abschätzung über die zu fahrenden Strecken und dem schnelleren Anfahren der Simulatoren von Positionen, welche reproduziert werden sollen. Abbildung 5-2 zeigt die Position des ersten Referenzpunktes. Der vom Sensor des Sonnensimulators gemessene Absolutwert beträgt dort 37616 mm, während der IMAX-Encoder der Erde an dieser Stelle den Wert 37625 mm überträgt. Die zweite Position befindet sich zur ersten im Abstand von der Hälfte des Schienenumfangs entlang der Schiene. Während der dort gemessene Positionswert der Sonne 48093 mm beträgt, steht die Überprüfung des Werts des Erdalbedosimulators an dieser Position noch aus, da dieser aufgrund von geometrischen Schwierigkeiten im Testzeitraum nicht an diese Referenzstelle gefahren werden konnte.



**Abb. 5–1: Messwerte der IMAX-Encoder bei zwei Testfahrten**



(a) Position relativ zur Anlage



(b) Ausrichtung an Kante

**Abb. 5–2: Position des ersten Referenzpunkts**

Um festzustellen, welche Absolutwerte über die gesamte Messstrecke ausgegeben werden können, soll das Magnetband von Ende zu Ende abgefahren werden. Da sich das Kabel des Sonnensimulators als dafür nicht lang genug herausstellt und der Erdbedoesimulator aus dem oben genannten Grund diese Stellen ebenfalls nicht erreichen kann, kann diese Messung nicht stattfinden. Aus dem Versuch mit der Sonne die Enden anzufahren, wurde der maximale Absolutwert dieses Simulators auf der Schiene bekannt. Dieser beträgt 49000 mm. Da sich das Limit des Sonnensimulators in unter 1 m Abstand zum Magnetbandende befindet, nehmen die absoluten Positionswerte der IMAX-Sensoren nach Absolvieren der Referenzfahrt einen kleineren Wert als 50000 mm an. Bei der Durchführung der Kalibrierungsfahrt werden Positionswerte nahe 2000000 mm ausgegeben. Abbildung 5-2 (b) zeigt die gemessenen Absolutwerte während und kurz nach einer Referenzfahrt. Da diese in großem Abstand zueinander liegen, können die auf dem Magnetband kodierte Absolutwerte von den während der Referenzfahrt ausgegebenen Sensorwerten unterschieden werden (**T.9**). Tabelle 5-1 stellt relevante Positionen für den Betrieb der Anlage mit ihren zugehörigen Absolutwerten dar.

**Tab. 5–1: Relevante Positionen und Absolutwerte**

Beschreibung	Simulator	Absolutwert
Referenzpunkt 1	Sonne	37616 mm
	Erde	37625 mm
Referenzpunkt 2	Sonne	48093 mm
	Erde	-
Maximaler Positionswert	Sonne	49000 mm

### 5.1.2 Testfahrten - Rotatorische Encoder

Die Drehwinkelsensoren werden mit einem an dem drehbaren Simulatorenbau angebrachten Laserpointer getestet. Dazu wird an der angestrahlten, 1,5 m entfernten Wand der Startpunkt gekennzeichnet. Nach dem Drehen des Simulators wird dieser Punkt wieder angefahren, um die Exaktheit und Reproduzierbarkeit der Sensorausgaben zu überprüfen. Tabelle 5-2 dokumentiert die Testfahrten. Die Start- und Endpunkte beziehen sich dabei auf die jeweiligen Sensormesswerte. Der in der Bemerkungsspalte angegebene Umrechnungsfaktor von Motorschritten in Grad bezieht sich auf die Sensormesswerte.

**Tab. 5–2: Positionen und Absolutwerte**

Simulator	Motorschritte	Startpunkt	Endpunkt	Bemerkung
Sonne	-10000	296,7°	183,5°	88,339 Schritte/Grad
	-25000	296,7°	15,0°	88,746 Schritte/Grad
	-26345	296,7°	0,0°	88,793 Schritte/Grad
	-31965	296,7°	296,7°	88,792 Schritte/Grad 360° Sensormessung
	-32020	296,7°	296,0°	88,772 Schritte/Grad 360,0° Lasermessung
	-16010	296,7°	116,4°	88,796 Schritte/Grad 180,3° Sensormessung
Erde	-10000	114,8°	1,3°	88,106 Schritte/Grad
	-10080	114,8°	0,0°	87,805 Schritte/Grad
	+5930	114,8°	181,7°	88,640 Schritte/Grad
	+22985	0,0°	259,5°	88,574 Schritte/Grad
	+26985	0,0°	304,5°	88,621 Schritte/Grad
	+31965	0,0°	0,0°	88,792 Schritte/Grad 360,0° Sensormessung 360,0° Lasermessung

Beide Sensoren zeigen bei einer Anzahl von 31965 Schritten eine Umdrehung von 360,0° an (Umrechnungsfaktor: 88,792 Schritte/Grad). Beim Erdalbedosimulator stimmen die Lasermessung und die Ausgabe des rotatorischen Encoders überein. Der Sensor der Sonne weist hingegen differierende Positionswerte bezüglich der Sensormessung und der Beobachtung mittels Laserpointer auf. Eine vollständige Umdrehung des Simulators wird erst nach 32020 Schritten erreicht (echter Umrechnungsfaktor: 88,944 Schritte in Grad). Dabei gibt der Sensor eine Drehung von 360,7° aus. Somit beträgt die Differenz zwischen Sensorausgabe und Lasermessung 0,7°. Beim Fahren von 16010 Motorschritten, also einer Drehung um 180°, kommt es zu einer Abweichung von 0,3°.

## 5.2 Verifikation

Zur Verifikation der in Kapitel 2 erhobenen Anforderungen wird Tabelle 5-2 genutzt. Wenn die Anforderung erfüllt wurde, ist sie mit grün gekennzeichnet. Rot steht für nicht erfüllte und gelb für teilweise erfüllte Anforderungen. Zusätzlich wird in Klammern das Kapitel angegeben, das den Beleg für die Beurteilung liefert. Wenn quantifizierte Ergebnisse vorliegen, werden diese angegeben.

**Tab. 5–3: Verifikation der Anforderungsliste**

Top-Level-Anforderungen		
ID	Anforderung	Beurteilung
T.1	Die Kosten für die Implementierung des Positionserfassungssystems dürfen 400 € nicht überschreiten	177 € (5.2.1)
T.2	Das Positionserfassungssystem soll dem Steuerungs-PC die absoluten Winkelpositionen des Sonnen- sowie des Erdalbedosimulators und die absoluten Positionen entlang der Führungsschiene übermitteln	(5.1)
T.3	Die Montage und Positionierung der Sensoren muss den Herstelleranforderungen genügen	(3.2.2)
T.4	Das Positionserfassungssystem muss die Randbedingungen der Anlage berücksichtigen und darf deren Funktion nicht einschränken	(3.2.2)
T.5	Der Messzeitpunkt der ausgegebenen Positionsdaten soll bei der Aktualisierung der Steuerungssoftware nicht länger als 10ms zurückliegen	(4.2.1)
T.6	Die Prozessdatenobjekte der Sensoren sollen am Steuerungs-PC in LabVIEW ausgegeben werden	(4.3, A.2)
T.7	Für den Übergang der Sensoren vom Pre-Operational- in den Operational-Modus muss der entsprechende NMT-Befehl vom Steuerungs-PC an die Sensoren gesendet werden können	(4.3.3)
T.8	Das CANopen-System muss getrennt vom Rest des Simulationssystems jederzeit in den Initialisierungszustand zurückgesetzt werden können	(4.3.3)
T.9	Die absoluten Positionswerte der IMAX-Sensoren sollen von den während der Referenzfahrt ausgegebenen Werten unterschieden werden können	(5.1.1)



### 5.2.1 Kostenaufstellung

Mit Tabelle 5-2 findet eine Kostenaufstellung statt, um die Erfüllung von **T.1** zu beurteilen. Zur Verifikation der ersten Anforderung werden die entstandenen Ausgaben zusammengerechnet.

**Tab. 5–4: Kostenaufstellung**

Subsystem	Teile	Preis
Positionierungsvorrichtung	Schienen und Gleitelemente	24 €
	Federn	20 €
	Klebeband	100 €
Schnittstelle	Teensy 3.2	21 €
	CAN-Transceiver	3 €
	Breadboard	5 €
	RS-232-Transceiver	4 €
	<b>Summe</b>	<b>177 €</b>

Mit einer Gesamtsumme von 177 € liegen die Kosten unter dem zur Verfügung gestellten Budget von 400 €. **T.1** ist somit erfüllt.

### 5.2.2 Nicht erfüllte Anforderungen

Bei der Implementierung der Drehencoder wurde festgestellt, dass deren kleinste asynchrone Übertragungsrate der TPDOs bei 10 ms liegt. Folglich kann **T.5** für die rotatorischen Sensoren nicht erfüllt werden.

### 5.2.3 Teilweise erfüllte Anforderungen

Da in Kapitel 5.1 festgestellt wurde, dass die Ausgabe des Drehencoders sich nicht mit den tatsächlichen Winkelposition des Simulators deckt, ist diese Anforderung für diesen Sensor nicht erfüllt. Mögliche Ursachen für die abweichenden Positionswerte werden in Kapitel 6 diskutiert.

## 6 Diskussion

### 6.1 Bewertung der IMAX-Positionierungsvorrichtung

Das in Kapitel 3.1.2 ausgewählte System erfüllt die Anforderungen an die Sensorpositionierung. Beim Design des Systems wurde allerdings fälschlicherweise angenommen, alle selbst herzustellenden Teile in einem Arbeitsschritt additiv fertigen zu können. Da der verwendete 3D-Drucker jedoch die M3-Gewinde nicht genau genug drucken kann, wurden die zugehörigen Löcher nachgebohrt und die Gewinde mit einem Windeisen eingeschnitten. Da das je Vorrichtung nur ein Bauteil mit je zwei M3-Gewinden betrifft, fällt der dadurch entstehende zusätzliche Aufwand relativ gering aus.

Die Montage der Vorrichtung gestaltet sich als sehr aufwändig. Zum Zusammenbauen der Vorrichtung muss der Metallkorb, auf welchem diese angebracht wird, aus dem jeweiligen Simulator ausgebaut werden, um alle Schrauben und Teile der Vorrichtung befestigen zu können. Vor allem das Rückanbringen des Metallkorbs an den Simulator ist schwierig und benötigt eine zweite Person. Folglich ist neben der Montage auch eine Reparatur des Systems zeitaufwändig.

Durch die sechs Federn und die vielen Einzelteile hat das System zusätzlich eine hohe Systemkomplexität. Um zu prüfen, ob und wie oft die Vorrichtung vor Simulationsbetrieb nachjustiert werden muss und ob Ermüdungserscheinungen auftreten, kann erst mit zunehmender Einsatzzeit beurteilt werden.

### 6.2 Bewertung des Schnittstellensystems

Kapitel 5 zeigt, dass auch das in dieser Arbeit ausgearbeitete Schnittstellen-System seine Aufgaben erfüllt. Im Vergleich zu einem ready-to-use CANopen-Interface ist die Implementierung eines Microcontroller-Systems mit CAN-Bibliothek für eine Kommunikation mit CANopen-Encodern mit zusätzlichem Arbeitsaufwand verbunden. Neben dem Zusammenfügen der einzelnen Komponenten gestaltet sich vor allem der Programmieraufwand für die Implementierung in die Anlage aufwändiger. Zusätzlich ist eine Konfiguration der Busteilnehmer über die Microcontroller-Programmierung aufgrund der fehlenden graphischen Benutzeroberfläche und der Arduino-Programmstruktur unintuitiv und arbeitsintensiv, weshalb in dieser Arbeit hierfür eine industrielle CAN-USB-Schnittstelle zum Einsatz kam.

Große Vorteile ergeben sich durch die Programmierbarkeit und die nicht vorhandene Kopplung der Hardware an eine bestimmte Software. Somit wird eine Variabilität der Ausgaben ermöglicht. Dieses System kann mit kleinen Programmänderungen verschiedenste Anforderungen erfüllen und in unterschiedlichen Einsatzszenarien verwendet werden. Die Programmierung im Arduino-Framework ist dabei in Verbindung mit der eingesetzten FlexCAN-Bibliothek einfach und somit auch für Programmier-Laien mit wenig Einarbeitungsaufwand verständlich.

Schließlich muss auch der Preis berücksichtigt werden. Das implementierte System unterbietet diesbezüglich die kommerziell erhältlichen, fertigen Systeme bei weitem.

Folglich bietet das eingesetzte Schnittstellen-System eine geeignete Alternative zu kommerziellen Angeboten, mit der Variabilität im Einsatz gewonnen und Kosten gespart werden.

### **6.3 Mögliche Ursachen für die differierenden Winkelmesswerte**

Im Folgenden werden mögliche Ursachen für die Abweichung der Positionswertausgaben des Drehencoders der Sonne zu den Positionsbeobachtungen des Simulators mittels Laserpointer beleuchtet.

Eine mangelhafte Befestigung des Laserpointers könnte dabei ein möglicher Grund für die differierenden Positionsmessungen sein, da dieser während den Messungen lediglich mit dem rotierenden Aufbau des Sonnensimulators verklemmt war. Dass die Lockerung des Laserpointers den Fehler verursacht, ist allerdings unwahrscheinlich, da die Abweichung der Ausgaben reproduzierbar ist und sich linear mit der Anzahl der Motorschritte zu ändern scheint.

Da die Encoderausgaben keine Sprünge aufweisen und es sich bei dem Sensor um ein absolut messendes System handelt, ist auch eine Fehlfunktion der Sensoren eher unwahrscheinlich.

Beide Drehencoder benötigen die exakt gleiche Anzahl an Motorschritten um eine volle Umdrehung anzuzeigen. Somit ist ein Mangel an der mechanischen Verbindung zwischen dem Schrittmotor, dem Simulator und dem rotatorischen Sensor die wahrscheinlichste Ursache für die differierenden Positionsbeobachtungen.

## 7 Zusammenfassung und Ausblick

Im Rahmen dieser Bachelorarbeit erfolgte die Konzeption und Implementierung eines CANopen-Sensorsystems in Sonnen- und Erdalbedosimulator der robotischen Anlage „RACOON-Lab“. Bei den Sensoren handelt es sich um zwei rotatorische und zwei lineare Encoder.

Grundlage für die Konzeption des Systems stellten die Top-Level-Anforderungen dar. Somit wurden die Randbedingungen geklärt, die für ein funktionierendes Busnetz nötig sind und das Zusammenwirken der einzelnen Hardwarekomponenten des Sensor-Systems besprochen. Die Ortswahl für die Magnetbandmontage fiel auf die innere der unteren beiden Führungsflächen der Schiene. Auf Grundlage der Subsystemanforderungen für die benötigte Positionierungsvorrichtung für den IMAX-Sensor wurden drei unterschiedliche Varianten in CATIA V5 designt. Diese mussten den Anforderungen an die Toleranzen des Sensors und den Randbedingungen der Anlage gerecht werden. Aufgrund der Kompaktheit und der additiven Fertigbarkeit wurde die dritte vorgestellte Vorrichtung ausgewählt. Die gewünschte Schnittstellenfunktionalität zwischen dem Steuerungs-PC und dem CAN-Bus sollte ein preiswertes Microcontroller-System bereitstellen. Dabei sollte die nötige CANopen-Funktionalität durch Programmierung im Arduino-Framework auf Basis der frei verfügbaren FlexCAN-Bibliothek hinzugefügt werden.

Die Implementierung beschäftigt sich mit dem Einbau und der Konfiguration des Systems. Da beide IMAX-Sensoren vom Hersteller auf die gleichen Knotennummern eingestellt waren, wurde diese beim Sensor des Sonnensimulators über die CANopen Layer Setting Services geändert, um einen Betrieb aller Sensoren am selben Netzwerk zu ermöglichen. Zusätzlich wurde bei den IMAX-Sensoren die Heartbeat-Funktion zur Geräteüberwachung aktiviert. Die Datenübertragung der Microcontroller-Schnittstelle an den PC und deren Stromversorgung wurde über den Anschluss an das sich in der Anlage befindliche CompactRIO-System gelöst.

Das implementierte Microcontroller-Programm gibt neben der Position der Sensoren mithilfe der Heartbeat-Nachrichten auch Auskunft über den Zustand der CANopen-Statusmaschine. Die Datenausgabe findet dabei in LabVIEW statt. Der Bediener der Anlage hat außerdem die Möglichkeit alle NMT-Befehle im Broadcast zu übertragen. Das Programm wurde bereits überarbeitet. Sowohl die ursprüngliche als auch die überarbeitete Version sind im Abgabeordner hinterlegt.

Um die Funktion der Komponenten des implementierten Messsystems zu validieren wurden Testfahrten durchgeführt. Die Funktion der Vorrichtung zur Sensorpositionierung und der Schnittstelle wurden dabei erfolgreich verifiziert. Durch die Messdaten des linearen Sensors wurde ein genauer Umrechnungsfaktor von Motorschritten in Meter berechnet. Des Weiteren wurden zwei Referenzpunkte auf der Führungsschiene der Anlage festgelegt, die das Verfahren der Simulatoren erleichtern sollen. Die an diesen Stellen gemessenen Absolutwerte der IMAX-Sensoren wurden dokumentiert. Dabei steht noch die Messung des Absolutwerts der Erde am Referenzpunkt 2 aus.

Der Drehencoder des Sonnensimulators fällt bei den Testfahrten mit differierenden Messwerten bezüglich der Positionsüberprüfung mittels Laserpointer auf. Die Ursache dafür konnte im Umfang dieser Arbeit nicht geklärt werden. Da eine Konfiguration der Drehwinkelsensoren auf eine anforderungsgerechte Positionsdatenübertragung nicht möglich ist, kann auch die zeitliche Anforderung an die Datenübertragung (**T.5**) nicht erfüllt werden.

Anschließend an diese Arbeit sollen die in Kapitel 7.3 aufgestellten Theorien zur Ursache der differierenden Positionswerte nachgeprüft werden. Zusätzlich sollen die absoluten Positionswerte des IMAX-Sensors am zweiten Referenzpunkt und an den Magnetbandenden gemessen und dokumentiert werden. Um das Schnittstellensystem robuster zu machen, ist außerdem geplant ein passendes Gehäuse für die Microcontroller-Schnittstelle additiv zu fertigen. Verbesserungspotenzial hat auch die Einarbeitung in LabVIEW. Die Schleife zum Ausgeben der Daten ist noch vollkommen entkoppelt vom Programm, mit welchem die Simulatoren verfahren werden. Während das Auslesen der absoluten Positionswerte aktuell vom Bediener der Software durchgeführt werden muss, könnte das Programm in Zukunft die Positionsdaten in einem Regelkreis zum Verfahren der Anlage nutzen. Somit könnten Positionen durch Eingeben des absoluten Positionswerts exakt angefahren werden. Das würde den Bediener der Anlage entlasten und ein genaueres Positionieren der Simulatoren ermöglichen. Dazu muss die Ausgabe des jetzigen Programms jedoch stark verändert werden.

## Literaturverzeichnis

- [1] J. Harder, M. Dziura, S. Haberl, "Future Technologies for Operating Robots in Space," Paper, Lehrstuhl für Raumfahrttechnik IAC-17,D1,6,7,x38585.
- [2] "RACOON Lab." URL <http://www.lrt.mw.tum.de/index.php?id=54> (Überprüfungsdatum 12.11.2019).
- [3] E. Hering, R. Martin, J. Gutekunst, and J. Kempkes, Eds., *Elektrotechnik und Elektronik für Maschinenbauer*, 4th ed., ser. VDI-Buch. Berlin: Springer Vieweg, 2018. URL <http://dx.doi.org/10.1007/978-3-662-57580-2>
- [4] M. Anderson, "Difference between absolute and incremental Encoders." URL <https://realpars.com/absolute-vs-incremental-encoder/> (Überprüfungsdatum 12.11.2019).
- [5] "Absolute vs. inkrementelle magnetkodierte Weg- und Winkelmesssysteme." URL [https://www.balluff.com/fileadmin/user\\_upload/PDF/basics/011\\_BML-Absolute-vs-Incremental\\_V1\\_DE.pdf](https://www.balluff.com/fileadmin/user_upload/PDF/basics/011_BML-Absolute-vs-Incremental_V1_DE.pdf) (Überprüfungsdatum 10.11.2019).
- [6] "IMAX." URL [https://www.elgo.de/fileadmin/user\\_upload/pdf/flyer/sensors/IMAX-00-FL-D.pdf](https://www.elgo.de/fileadmin/user_upload/pdf/flyer/sensors/IMAX-00-FL-D.pdf) (Überprüfungsdatum 12.11.2019).
- [7] "Betriebsanleitung: SERIE IMAX." URL [https://www.elgo.de/fileadmin/user\\_upload/pdf/manual/sensors/IMAX-00-MA-D.pdf](https://www.elgo.de/fileadmin/user_upload/pdf/manual/sensors/IMAX-00-MA-D.pdf) (Überprüfungsdatum 12.11.2019).
- [8] "CAN OUTPUT CONTACTLESS ROTARY POSITIONS SENSORS." URL [https://www.cw-industrialgroup.com/getattachment/39072fa1-e50e-4d70-9729-c3d561194127/srh520cn\\_brochure](https://www.cw-industrialgroup.com/getattachment/39072fa1-e50e-4d70-9729-c3d561194127/srh520cn_brochure) (Überprüfungsdatum 12.11.2019).
- [9] "NI-9881: CANopen-Schnittstellenmodul der C-Serie." URL <https://www.ni.com/de-de/support/model.ni-9881.html> (Überprüfungsdatum 12.11.2019).
- [10] "Kickdrive Zero & fullmo Kickdrive - CANopen tools for Windows." URL <https://kickdrive.de/en/> (Überprüfungsdatum 12.11.2019).
- [11] "CANFestival - Overview." URL <https://canfestival.org/doc> (Überprüfungsdatum 12.11.2019).
- [12] "CANFestivino." URL <https://github.com/jgeisler0303/CANFestivino> (Überprüfungsdatum 12.11.2019).
- [13] "mbed-canfestival." URL <https://github.com/Jian-Zhan/mbed-canfestival> (Überprüfungsdatum 12.11.2019).
- [14] "Arduino library for CAN on Teensy 3.1, 3.2, 3.5, and 3.6." URL [https://github.com/collin80/FlexCAN\\_Library](https://github.com/collin80/FlexCAN_Library) (Überprüfungsdatum 12.11.2019).
- [15] "Teensy USB Development Board: Teensy 3.2." URL <https://www.pjrc.com/store/teensy32.html> (Überprüfungsdatum 12.11.2019).

- [16] “TJA1050: High speed CAN transceiver.” URL <https://www.nxp.com/docs/en/data-sheet/TJA1050.pdf> (Überprüfungsdatum 12.11.2019).
- [17] “CANopen for Python.” URL <https://github.com/christiansandberg/canopen> (Überprüfungsdatum 12.11.2019).
- [18] “KaCanOpen.” URL <https://github.com/KITmedical/kacanopen> (Überprüfungsdatum 12.11.2019).
- [19] “CanFestival.” URL <https://github.com/ljessendk/CanFestival> (Überprüfungsdatum 10.11.2019).
- [20] “MCP\_CAN Library for Arduino.” URL [https://github.com/coryjfowler/MCP\\_CAN\\_lib](https://github.com/coryjfowler/MCP_CAN_lib) (Überprüfungsdatum 12.11.2019).
- [21] “Universelle CAN Bibliothek (avr-can-lib).” URL <https://github.com/dergraaf/avr-can-lib> (Überprüfungsdatum 10.11.2019).
- [22] H. Zeltwanger, Ed., *CANopen*. Berlin: VDE-Verl., 2001.
- [23] “CANopen -The standardized embedded network.” URL <https://www.can-cia.org/canopen> (Überprüfungsdatum 12.11.2019).
- [24] *CiA 303: Part 1: Cabling and connector pin assignment*, CAN in Automation, 18.09.2017, Version 1.9.0.
- [25] *CiA 301: CANopen application layer and communication profile*, CAN in Automation, 21.02.2011, Version 4.2.0.
- [26] “CiA 406: Profile for Encoders.” URL <https://www.can-cia.org/can-knowledge/canopen/cia406/> (Überprüfungsdatum 11.11.2019).
- [27] “NI cRIO-9012/9014: USER MANUAL AND SPECIFICATIONS.” URL <http://www.sal.wisc.edu/PFIS/docs/rss-nir/archive/public/Product%20Manuals/ni/ni-9012-14-manual.pdf> (Überprüfungsdatum 12.11.2019).
- [28] “Adafruit Perma-Proto Half-sized Breadboard PCB.” URL <https://www.adafruit.com/product/571> (Überprüfungsdatum 12.11.2019).
- [29] “TJA1050 CAN Controller Module.” URL <https://www.optimusdigital.ro/en/others/738-modul-controller-can-tja1050.html> (Überprüfungsdatum 12.11.2019).
- [30] “RS232 Serial Port to TTL Digital Converter Module SIP-PEX SP3232EEN 5V 3.3V.” URL <https://www.conrad.de/de/p/rs232-serial-port-to-ttl-digital-converter-module-sipex-sp3232een-5v-3-3v-802236383.html> (Überprüfungsdatum 12.11.2019).
- [31] *CiA 305: Layer Setting Services*, CAN in Automation, 31.05.2000, Version 1.0.
- [32] “SRH520CN Issue 6: Object 1800h: TPDO 1 Communication Parameter: CANopen Technical Information.” URL [https://www.cw-industrialgroup.com/getattachment/7a17a486-5609-40aa-ad92-b88a47d04d98/pgs-canopen\\_technical\\_data](https://www.cw-industrialgroup.com/getattachment/7a17a486-5609-40aa-ad92-b88a47d04d98/pgs-canopen_technical_data) (Überprüfungsdatum 12.11.2019).



- [33] “SDO – Service Data Objects – CanOpen.” URL <http://www.byte-me.org.uk/canopenparent/canopen/sdo-service-data-objects-canopen/> (Überprüfungsdatum 12.11.2019).
- [34] “Teensy 3.2.” URL <http://www.hobbytronics.co.uk/teensy-v32> (Überprüfungsdatum 11.11.2019).
- [35] “SP3222E/3232E - True +3.0V to +5.5V RS-232 Transceivers.” URL <https://pdf1.alldatasheet.com/datasheet-pdf/view/157042/SIPEX/SP3232EEN.html> (Überprüfungsdatum 12.11.2019).
- [36] “Teensy 3.2 Pins.” URL <https://www.pjrc.com/teensy/pinout.html> (Überprüfungsdatum 12.11.2019).
- [37] “Atom - A hackable text editor for the 21st Century.” URL <https://atom.io/> (Überprüfungsdatum 10.11.2019).
- [38] “PlatformIO IDE for Atom.” URL <https://atom.io/packages/platformio-ide> (Überprüfungsdatum 12.11.2019).
- [39] “RACOON CAN Encoders.” URL <https://gitlab.lrz.de/racoon-projects/can-encoders> (Überprüfungsdatum 11.11.2019).
- [40] “Arduino-Programmierhandbuch: Struktur.” URL <http://www.ov-meschede.de/wp-content/uploads/2018/05/Arduino-Programmierhandbuch.pdf> (Überprüfungsdatum 10.11.2019).

# A Anhang

## A.1 LabVIEW-Blockdiagramm

