



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# **Bachelorarbeit**

**Leonid Dubovyy**

**Konzeptionierung und Implementierung eines universellen und  
zuverlässigen Verfahrens zur Geschwindigkeitsermittlung an  
Gesamtfahrzeugen im AUDEx**

*Fakultät Technik und Informatik  
Department Fahrzeugtechnik und  
Flugzeugbau*

*Faculty of Engineering and Computer Science  
Department of Automotive and Aeronautical  
Engineering*

Leonid Dubovyy

**Konzeptionierung und Implementierung eines universellen und  
zuverlässigen Verfahrens zur Geschwindigkeitsermittlung an  
Gesamtfahrzeugen im AUDEx**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Fahrzeugbau  
am Department Fahrzeugtechnik und Flugzeugbau  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Engel  
Zweitgutachter: Prof. Dr. Netzel

Eingereicht am: 17. April 2025

**Leonid Dubovyy**

**Thema der Arbeit**

Konzeptionierung und Implementierung eines universellen und zuverlässigen Verfahrens zur Geschwindigkeitsermittlung an Gesamtfahrzeugen im AUDEx

**Stichworte**

Mikrocontroller, Geschwindigkeitsermittlung, Raddrehzahlmessung, Hall-Sensor, CAN-Bus

**Kurzzusammenfassung**

Diese Arbeit beschäftigt sich mit der Entwicklung eines zuverlässigen Systems zur Fahrzeuggeschwindigkeitsmessung im Rahmen des AUDEx-Projekts. Ziel war die Konzeption und Implementierung einer Lösung, die sowohl hohe Genauigkeit als auch Robustheit bietet. Dafür wurden verschiedene Messverfahren analysiert und Hall-Sensoren näher betrachtet. Das entwickelte System wurde erfolgreich in einem Modellfahrzeug getestet und zeigte eine hohe Messpräzision. Zukünftige Arbeiten könnten eine weitere Optimierung, eine Sensorfusion und eine Skalierung auf größere Fahrzeugmodelle umfassen.

**Leonid Dubovyy**

**Title of the paper**

Conceptualization and implementation of a universal and reliable method for speed determination in complete vehicles in AUDEx.

**Keywords**

Microcontroller, speed determination, wheel speed measurement, Hall sensor, CAN bus

**Abstract**

This thesis focuses on the development of a reliable system for vehicle speed measurement within the AUDEx project. The goal was to design and implement a solution that provides both high accuracy and robustness. To achieve this, various measurement methods were analyzed with particular attention to Hall sensors. The developed system was successfully tested in a model vehicle and demonstrated high measurement precision. Future work could involve further optimization, sensor fusion, and scaling to larger vehicle models.

# Danksagung

Im Rahmen meiner Bachelorarbeit möchte ich meiner Lebenspartnerin meine aufrichtige Dankbarkeit aussprechen. Ihre bedingungslose Unterstützung und Geduld haben einen wesentlichen Beitrag zum erfolgreichen Abschluss meiner Forschungsarbeit geleistet. Ihr Vertrauen hat mich motiviert und gestärkt.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Zielsetzung . . . . .	1
1.2. Motivation . . . . .	1
1.3. Bestandsaufnahme und Aufgabenstellung . . . . .	2
1.4. Aufbau der Arbeit . . . . .	2
<b>2. Grundlagen</b>	<b>4</b>
2.1. Prinzipien der Geschwindigkeitsmessung . . . . .	4
2.2. Messmethoden . . . . .	4
2.2.1. GPS . . . . .	4
2.2.2. Radar . . . . .	5
2.2.3. Induktive Sensoren . . . . .	5
2.2.4. Hall-Sensoren . . . . .	5
2.3. Anwendung von Mikrocontroller . . . . .	5
2.4. CAN-Bus-Kommunikation . . . . .	7
2.5. Übersicht über das AUDEx-Projekt . . . . .	9
<b>3. Analyse und Konzeptentwicklung</b>	<b>11</b>
3.1. Anforderungen an das AUDEx Messsystem . . . . .	11
3.2. Vergleich bestehender Technologien . . . . .	12
3.3. Auswahl des Messprinzips . . . . .	13
3.4. Konzept des Messsystems . . . . .	14
3.4.1. Bewertung bestehender Prototypen zur Geschwindigkeitsermittlung . . . . .	14
<b>4. Hardware-Entwicklung</b>	<b>17</b>
4.1. Asynchron Treiber . . . . .	17
4.2. Funktionsprinzip der induktiven Spule . . . . .	20
4.2.1. Verbesserung der Empfindlichkeit . . . . .	21
4.2.2. Induktive Spule Zusammenfassung . . . . .	24
4.3. Spulenkonstruktion und Berechnung . . . . .	25
4.3.1. Magnetischer Fluss und Induktion . . . . .	25
4.3.2. Induktivität der Spule . . . . .	26
4.3.3. Signalverarbeitung . . . . .	27
4.4. Funktionsprinzip des Hall-Sensors . . . . .	31
4.4.1. Vorteile von Hall-Sensoren . . . . .	32
4.4.2. Nachteile von Hall-Sensoren . . . . .	33

4.5.	Hall-Sensor als prallfreier Taster . . . . .	34
4.5.1.	Arbeitsweise als Taster . . . . .	36
4.6.	Finales Hardwaredesign . . . . .	38
<b>5.</b>	<b>Software-Entwicklung</b>	<b>39</b>
5.1.	Architektur und Anforderungen . . . . .	39
5.2.	Datenerfassung und Verarbeitung . . . . .	40
5.3.	Signalverarbeitung . . . . .	41
5.4.	Kommunikationsschnittstellen . . . . .	42
5.5.	Logischer Aufbau . . . . .	43
5.6.	Grenzen der Messgenauigkeit bei niedrigen Geschwindigkeiten . . . . .	45
<b>6.</b>	<b>Implementierung und Inbetriebnahme des Systems</b>	<b>47</b>
6.1.	Aufbau des Prototyps . . . . .	47
6.1.1.	Mechanischer Aufbau . . . . .	47
6.1.2.	Elektrischer Aufbau . . . . .	48
6.2.	Integration in das bestehende Fahrzeug . . . . .	51
6.2.1.	Montage im Fahrzeug . . . . .	55
<b>7.</b>	<b>Verifikation der Anforderungen</b>	<b>57</b>
<b>8.</b>	<b>Validierung</b>	<b>58</b>
8.1.	Testmethodik und Messergebnisse . . . . .	58
8.2.	Diskussion der Ergebnisse . . . . .	60
<b>9.</b>	<b>Zusammenfassung und Ausblick</b>	<b>64</b>
9.1.	Zusammenfassung der Ergebnisse . . . . .	64
9.2.	Ausblick auf zukünftige Arbeiten . . . . .	65
<b>A.</b>	<b>Anhang: Quellcode ECU-Modul (ESP32-S3 Mini)</b>	<b>67</b>
<b>B.</b>	<b>Anhang: ECU-Empfänger mit OLED-Display</b>	<b>71</b>
<b>C.</b>	<b>Anhang: Code für Anzeigemodul (CAN-Empfänger mit OLED)</b>	<b>72</b>
<b>D.</b>	<b>Anhang: CAN-Bus-Modul mit Stiftleisten-Verbindung)</b>	<b>75</b>

# Abbildungsverzeichnis

2.1.	FRC-EP170 von der Firma IXXAT [14]. . . . .	6
2.2.	Spannungspegel im Highspeed-CAN-Bus [17]. . . . .	7
2.3.	Frameaufbau einer CAN-Nachricht (in HEX: ID 111, Data 11 11) [17]. . . . .	8
2.4.	Darstellung der Byte-Positionen [17]. . . . .	9
3.1.	v.l.n.r.: ESP32 Super Mini, ESP32 WROOM, STM32 Blue Pill, RP2040-Zero. . . . .	13
4.1.	Drei Phasen Inverter in Simscape, MathWorks [19]. . . . .	18
4.2.	Typisches Aussehen von um 120 Grad verschobenen Sinuswellen (zwei Phasen). . . . .	19
4.3.	Spule mit drei Magneten. . . . .	25
4.4.	Lüfter mit drei Magneten (unten ist die Spule zu sehen). . . . .	26
4.5.	Digitale Welle mit 2,92V aus LM339L und analoge Sinuskurve mit 47,1V aus der Spule. . . . .	28
4.6.	Digitale Welle und analoge Sinuskurve. . . . .	29
4.7.	Experimentelle Point-to-Point Verlotung der Diode zwischen Spule und Eingang des LM339. . . . .	30
4.8.	Typisches Aussehen von DIP-14 Bauweise LM1815, LM339 und 74HC14. . . . .	31
4.9.	Hall-Effekt: 1. Platte, 2. Stromquelle, 3. Pos. Leitung, 4. Neg. Leitung, 5. Magnet, 6. Voltmeter, 7. Abgelenkte Pos. Atome, 8. Neg. Atome. . . . .	32
4.10.	Integrierte Schaltkreise eines TLE4905L von Fa. Infineon [13]. . . . .	33
4.11.	TLE4905L von Fa. Infineon [13]. . . . .	35
4.12.	Detektion von drei Magneten mit Hilfe des TLE4905L, abgelesen mit einem Oszilloskop. . . . .	37
5.1.	Vereinfachter logischer Aufbau von Code im ECU. . . . .	44
6.1.	Zwei verschiedene Montagestellen im Vergleich. . . . .	48
6.2.	Anschlusszeichnung der ECU. . . . .	49
6.3.	3D-gedruckte Teile. . . . .	50
6.4.	Anschlusszeichnung der ECU mit Transistor 2N2222. . . . .	51
6.5.	Zustände des Open-Collector-Ausgangs bei nicht erkanntem (links) und erkanntem Magneten (rechts). . . . .	52
6.6.	Fahrgestell von Absima 1:8 Team C GR8LE mit ECU und Hall-Sensor (unten rechts ist die Magnetplatzierung) [5]. . . . .	55
6.7.	Drehzahlen und Geschwindigkeit in Terminal von Visual-Studio-IDE . . . . .	56

8.1. Messung am Mitnehmer mit ECU und Drehzahlmesser der Fa. Voltcraft, die Werte stimmen überein. . . . .	62
8.2. Messung am Rad mit ECU und Drehzahlmesser der Fa. Voltcraft, die Werte stimmen überein. . . . .	63
B.1. ECU-Empfänger mit OLED-Display. . . . .	71
D.1. ECU-Modul mit Steckverbinder. . . . .	75
D.2. ECU-Modul mit Steckverbinder, Hall-Sensor-Anschluss. . . . .	76
D.3. CAD Baugruppe, ECU-Gehäuse. . . . .	76

# Tabellenverzeichnis

2.1.	Schreibweisen von Dezimalen, Hexadezimalen und binären Zahlen [17]. . . . .	9
3.1.	Vergleich bestehender Geschwindigkeitsmess-Technologien. . . . .	12
3.2.	Vergleich verschiedener Mikrocontroller. . . . .	12
4.1.	Relative Permeabilität ( $\mu_r$ ) für verschiedene Materialien. . . . .	22
4.2.	Theoretisch induzierte Spannungen für unterschiedliche Kerne bei 500 Windungen und unterschiedlichen Drehzahlen. . . . .	24
4.3.	Vergleich verschiedener Hall-Sensoren. . . . .	34
4.4.	Komponentenbeschreibung für die Geschwindigkeitsermittlung. . . . .	38
6.1.	Anzeigen zur Diagnose. . . . .	56
8.1.	Vergleich der Drehzahlerfassung an Welle und Rad. . . . .	59
8.2.	Radgeschwindigkeit in Abhängigkeit der Mitnehmerdrehzahl (Übersetzung 10:43, Raddurchmesser 10 cm). . . . .	60
8.3.	Vergleich der Drehzahlerfassung an Welle und Rad (angetrieben mit Akkuschauber). . . . .	60

# Abkürzungsverzeichnis

<b>AUDEx</b>	Automotive Development in 1:X
<b>GPS</b>	Global Positioning System
<b>ABS</b>	Antiblockiersystem
<b>Pkw</b>	Personenkraftwagen
<b>Lkw</b>	Lastkraftwagen
<b>PWM</b>	Pulsweitenmodulation
<b>GPIO</b>	General Purpose Input Output
<b>GND</b>	Ground
<b>MOSFET</b>	Metal-Oxide-Semiconductor Field-Effect Transistor
<b>RC</b>	Radio Controlled
<b>TWAI</b>	Two-Wire Automotive Interface
<b>CAN</b>	Controller Area Network
<b>ECU</b>	Electronic Control Unit
<b>ID</b>	Identifikationsnummer

# Symbolverzeichnis

$v$	Fahrzeuggeschwindigkeit in m/s
$\omega$	Winkelgeschwindigkeit in rad/s
$d$	Raddurchmesser in m
$f$	Frequenz der Signalflanken in Hz
$U$	Spannung in V
$R$	Widerstand in $\Omega$
$N$	Anzahl der Pulse
$T$	Abtastintervall in s
$A$	Fläche in m <sup>2</sup>
$\Phi$	Magnetischer Fluss in Wb
$\mu_r$	Relative Permeabilität (dimensionslos)
$L$	Induktivität in mH
$i$	Übersetzungsverhältnis

# 1. Einleitung

Die präzise Ermittlung der Fahrzeuggeschwindigkeit ist eine zentrale Herausforderung in der modernen Fahrzeugtechnik, insbesondere im Kontext von Regelungssystemen und Fahrerassistenzfunktionen. Im Rahmen des AUDEx-Projekts, das die Entwicklung und Erprobung innovativer Technologien an Modellfahrzeugen im Maßstab 1:8 und 1:5 ermöglicht, liegt der Fokus dieser Arbeit auf der Geschwindigkeitsermittlung.

Die Arbeit umfasst die Analyse und Auswahl geeigneter Messverfahren sowie die Entwicklung und Implementierung eines robusten Messsystems. Ziel ist es, ein System zu schaffen, das nicht nur genaue Ergebnisse liefert, sondern auch den Anforderungen an Modularität und Vielseitigkeit gerecht wird. Dabei werden sowohl hardware- als auch softwareseitige Aspekte berücksichtigt, um eine ganzheitliche Lösung zur Geschwindigkeitsmessung zu entwickeln.

## 1.1. Zielsetzung

Ziel dieser Arbeit ist die Entwicklung eines universellen und zuverlässigen Verfahrens zur Geschwindigkeitsermittlung an Gesamtfahrzeugen im Rahmen des AUDEx-Projekts. Dabei sollen verschiedene Messmethoden analysiert und ein Konzept erarbeitet werden, das sowohl präzise als auch robust ist.

Das entwickelte Verfahren soll auf unterschiedlichen Fahrzeugmodellen im AUDEx-Projekt (sowohl Pkw- als auch Lkw-Modelle) anwendbar sein. Um dieses Ziel zu erreichen, werden bestehende Technologien zur Geschwindigkeitsermittlung untersucht, bewertet und anschließend in einem geeigneten System implementiert. Ein besonderes Augenmerk liegt auf der Verifikation und Validierung der Funktionsfähigkeit des entwickelten Verfahrens anhand von Fahrversuchen. Abschließend sollen die gewonnenen Erkenntnisse dokumentiert und mögliche Optimierungspotenziale für zukünftige Entwicklungen aufgezeigt werden.

## 1.2. Motivation

Die genaue Bestimmung der Fahrzeuggeschwindigkeit ist ein entscheidender Faktor für zahlreiche Anwendungen in der Fahrzeugtechnik, insbesondere in den Bereichen Regelungstechn-

nik, Fahrdynamikregelung und Fahrerassistenzsysteme. Im AUDEx-Projekt wurde bisher ein Verfahren zur Geschwindigkeitsermittlung basierend auf der Erfassung der Raddrehzahlen implementiert, das jedoch in der Praxis Unzuverlässigkeiten und Ungenauigkeiten aufwies.

Um eine zuverlässige und skalierbare Lösung für verschiedene Fahrzeugmodelle zu ermöglichen, ist die Entwicklung eines neuen, robusten Messsystems notwendig. Diese Arbeit soll dazu beitragen, ein universelles Verfahren zu konzipieren, das verschiedene Sensortechnologien wie Hall-Sensoren, GPS oder induktive Sensoren integriert und dabei eine hohe Messpräzision gewährleistet. Durch die Entwicklung eines optimierten Systems können zukünftige Forschungen und Anwendungen im AUDEx-Projekt verbessert und erweitert werden.

### 1.3. Bestandsaufnahme und Aufgabenstellung

Für die Geschwindigkeitsmessung wurden bisher unterschiedliche Technologien untersucht, darunter die Integration von GPS-Modulen, Raddrehzahlsensoren, induktiven Sensoren und Hall-Sensoren. Das aktuell im AUDEx-Projekt implementierte Verfahren [4] basiert auf der Erfassung der Raddrehzahlen mit Hilfe von Hall-Sensoren, weist jedoch im Betrieb signifikante Unzuverlässigkeiten und Ungenauigkeiten auf.

Daher soll im Rahmen dieser Arbeit ein universelles und zuverlässiges Verfahren zur Geschwindigkeitsermittlung für verschiedene Fahrzeugmodelle im AUDEx-Projekt konzipiert und realisiert werden. Dabei sollen geeignete Hard- und Softwarelösungen entwickelt und implementiert werden. Durch die systematische Vorgehensweise soll ein robuster und anwendungsfähiger Lösungsansatz für die Geschwindigkeitsermittlung geschaffen werden, der sowohl für Forschungszwecke als auch für praxisnahe Lehrveranstaltungen nutzbar ist.

### 1.4. Aufbau der Arbeit

Die Struktur dieser Arbeit orientiert sich am V-Modell, einem etablierten Vorgehensmodell zur systematischen Entwicklung und Qualitätssicherung technischer Systeme. Es stellt sicher, dass jeder Entwurfsschritt durch eine entsprechende Verifikation oder Validierung auf der gegenüberliegenden Seite des Modells abgesichert wird. In diesem Sinne folgt die Gliederung der Arbeit den Phasen des V-Modells und bildet dessen Struktur methodisch ab:

- **Kapitel 2 – Grundlagen** (*Systemkontextanalyse / Domänenverständnis*): Dieses Kapitel vermittelt die theoretischen und technischen Grundlagen der Geschwindigkeitsmessung. Es werden verschiedene Messmethoden wie GPS, Radar, induktive Sensoren und Hall-Sensoren vorgestellt sowie das AUDEx-Projekt eingeordnet.

- **Kapitel 3 – Analyse und Konzeptentwicklung** (*Anforderungsanalyse und Systementwurf*): Die Anforderungen an das Geschwindigkeitsmesssystem werden erarbeitet und bestehende Technologien analysiert. Darauf aufbauend wird ein Konzept für ein verbessertes, zuverlässiges Verfahren entwickelt.
- **Kapitel 4 – Hardware-Entwicklung** (*Komponentendesign*): Hier werden die eingesetzten Hardwarekomponenten wie Spulen und Sensoren sowie ihre Auslegung und Verschaltung beschrieben.
- **Kapitel 5 – Software-Entwicklung** (*Moduldesign und Softwarearchitektur*): Die softwareseitige Erfassung und Verarbeitung der Sensordaten wird erläutert, einschließlich Signalverarbeitung und Kommunikationsschnittstellen.
- **Kapitel 6 – Implementierung und Inbetriebnahme des Systems** (*Implementierungsphase*): Die praktische Umsetzung des entwickelten Systems wird dokumentiert. Dazu gehören der Aufbau des Prototyps, die Inbetriebnahme sowie die mechanische und elektrische Integration ins Fahrzeugmodell.
- **Kapitel 7 – Verifikation der Anforderungen** (*Komponenten- und Integrationstest*): In dieser Phase wird die Einhaltung der definierten Anforderungen überprüft. Dazu zählen funktionale Prüfungen einzelner Module.
- **Kapitel 8 – Validierung der Ergebnisse** (*Systemtest / Abnahmetest*): Die Gesamtfunktionalität des Systems wird durch gezielte Tests und Fahrersatzmanöver überprüft und auf Praxistauglichkeit bewertet.
- **Kapitel 9 – Zusammenfassung und Ausblick** (*Projektabschluss*): Abschließend werden die wichtigsten Ergebnisse reflektiert und ein Ausblick auf mögliche Weiterentwicklungen im Kontext des AUDEx-Projekts gegeben.

## 2. Grundlagen

In diesem Kapitel werden Grundlagen zur Geschwindigkeitsmessung vorgestellt und näher erläutert.

### 2.1. Prinzipien der Geschwindigkeitsmessung

Die Messung der Geschwindigkeit eines Fahrzeugs ist eine zentrale Aufgabe in der Fahrzeugtechnik, insbesondere für die Regelung von Fahrassistenzsystemen und dynamischen Fahrzeugsteuerungen. Die Geschwindigkeit wird in der Regel über die Messung von physikalischen Größen wie Zeit, Weg oder Frequenzänderungen ermittelt. Diese Größen können direkt, beispielsweise durch Weg-Zeit-Messungen, oder indirekt, etwa über den Doppler-Effekt<sup>1</sup> oder induktive Sensoren, erfasst werden. Die Auswahl eines geeigneten Messprinzips hängt von den Anforderungen an Genauigkeit, Robustheit und Kosten ab.

Ein grundlegendes Prinzip ist die Bestimmung der Geschwindigkeit aus der zeitlichen Änderung der Position, die beispielsweise mit GPS-Daten oder durch Drehzahlsensoren erfasst werden kann. Zusätzlich werden in modernen Fahrzeugen vermehrt induktive oder Hallbasierte Sensoren zur Messung der Geschwindigkeit an Antriebswellen oder direkt an den Rädern eingesetzt.

### 2.2. Messmethoden

Für die Geschwindigkeitsmessung stehen unterschiedliche Technologien zur Verfügung, die je nach Einsatzzweck ausgewählt werden. Zu den gängigen Methoden gehören:

#### 2.2.1. GPS

Die satellitengestützte Geschwindigkeitsmessung mittels GPS basiert auf der Ermittlung der Fahrzeugposition in regelmäßigen Zeitintervallen. Aus der Differenz der Positionen und dem

---

<sup>1</sup>Der Doppler-Effekt beschreibt die Frequenzänderung einer Welle durch die Relativbewegung zwischen Sender und Empfänger.

Zeitabstand wird die Geschwindigkeit berechnet. Diese Methode ist besonders in offenen Umgebungen effizient, jedoch anfällig für Signalstörungen durch Gebäude oder dichte Bewaldung.

### 2.2.2. Radar

Radarsysteme senden elektromagnetische Wellen aus und analysieren deren Reflexion von der Fahrbahn. Die Geschwindigkeit wird aus der Frequenzverschiebung, die durch den Dopplereffekt entsteht, berechnet. Diese Methode ist schlupffrei und eignet sich hervorragend für dynamische Messungen, insbesondere bei hohen Geschwindigkeiten.

### 2.2.3. Induktive Sensoren

Induktive Sensoren werden häufig in ABS zur Drehzahlerfassung an den Rädern eingesetzt. Diese Sensoren arbeiten auf Basis elektromagnetischer Induktion: Ein rotierender Zahnkranz oder eine gezahnte Scheibe erzeugt in einer Spule ein wechselndes Magnetfeld, das proportional zur Drehzahl ist. Die daraus resultierende Frequenz wird zur Berechnung der Geschwindigkeit genutzt. Induktive Sensoren sind robust und wartungsfrei, jedoch können sie durch Verschmutzung oder mechanische Beschädigung beeinträchtigt werden.

### 2.2.4. Hall-Sensoren

Hall-Sensoren basieren auf dem Hall-Effekt, bei dem ein Magnetfeld eine Spannung in einem leitfähigen Material erzeugt. Diese Sensoren finden Anwendung in der Drehzahlerfassung an Rädern oder Antriebswellen. Sie messen entweder die Magnetfeldänderung eines rotierenden Magnets oder die Bewegung eines magnetischen Zahnrads. Hall-Sensoren sind weniger anfällig für Verschmutzungen als induktive Sensoren und bieten eine hohe Präzision, insbesondere bei geringen Geschwindigkeiten.

## 2.3. Anwendung von Mikrocontroller

Um Daten zu empfangen, zu verarbeiten und weiterzuleiten, wird ein Mikrocontroller benötigt. Ein Mikrocontroller besteht meist aus einem einzigen Chip, der einen vollständigen Computer enthält und mit seinen Ein- und Ausgängen ein funktionsfähiges System bildet. Zudem unterscheiden sich viele Mikrocontroller in den verfügbaren Schnittstellen sowie den dazugehörigen Onboard-Treibern.

## 2. Grundlagen

---

Viele Hersteller bieten fertige Onboard-Lösungen an, bei denen der Chip bereits auf einer Platine vormontiert ist, inklusive Ein- und Ausgängen sowie den dafür benötigten Widerständen.

Die gängigsten Mikrocontroller auf dem Markt sind der STM32, der ESP32 und der RP2040. Sie sind besonders attraktiv, da sie kein komplexes Betriebssystem benötigen und innerhalb weniger Sekunden einsatzbereit sind. Da lediglich ein Hauptprogramm in einer Schleife läuft, ist der Energieverbrauch äußerst gering.

Einige Hersteller haben das Defizit an fertigen Embedded-Plattformen<sup>2</sup> auf dem Markt erkannt und bieten eigene Lösungen für die Arbeit mit der Fahrzeugkommunikation, die Erstellung von Gateways sowie triggerbasierten Logdatenschreibern an. Einer dieser Hersteller ist IXXAT mit dem Modell FRC-EP170 (siehe Abbildung 2.1). Das FRC-EP170 arbeitet autark und verfügt über alle erforderlichen Schnittstellen und Interfaces für die Kommunikation zwischen Triggern/Sensoren, die Fahrzeugkommunikation und falls erforderlich die Anbindung an einen Server zur dezentralen Speicherung von Logdateien.



Abbildung 2.1.: FRC-EP170 von der Firma IXXAT [14].

---

<sup>2</sup>Embedded Platform: Spezielles Computersystem mit eigener Hardware und Software, integriert in größere Systeme, oft für Echtzeitanwendungen optimiert.

## 2.4. CAN-Bus-Kommunikation

<sup>3</sup>Der CAN-Bus ist nach wie vor ein zuverlässiges Netzwerkprotokoll für moderne Fahrzeuge, das eine schnelle Kommunikation zwischen verschiedenen Steuergeräten ermöglicht. Dieses Kommunikationssystem wurde in den 1980er-Jahren von Bosch speziell für den Einsatz im Fahrzeug entwickelt und bietet eine Vielzahl an Vorteilen. Sowohl Software als auch Hardware sind vergleichsweise einfach aufgebaut und eignen sich daher nicht nur für den Automobilbereich, sondern auch für industrielle Anwendungen, beispielsweise in Robotiksystemen oder der Gebäudeautomatisierung [24].

In dieser Arbeit kommt ausschließlich CAN-High-Speed mit einer Baudrate von 500 kbit/s zum Einsatz. Bei dieser Übertragungsrate sind Kabellängen bis zu 125 m möglich, wobei verdrehte Zweidrahtleitungen (CAN-High und CAN-Low) zur Störunterdrückung verwendet werden. Die physikalische Signalübertragung lässt sich mit einem Oszilloskop beobachten; die binären Daten sind dabei als Spannungsverläufe sichtbar. Die beiden Leitungen liefern eine maximale Differenzspannung von 2 V: Wird eine logische „1“ (rezessiv) gesendet, liegen beide Leitungen bei etwa 2,5 V – es entsteht keine Differenzspannung. Bei einer logischen „0“ (dominant) liegt CAN-High bei 3,5 V und CAN-Low bei 1,5 V. Das Signal ist spiegelsymmetrisch zur Mittelspannung von 2,5 V und wird als symmetrische Übertragung bezeichnet (s. Abbildung 2.2).

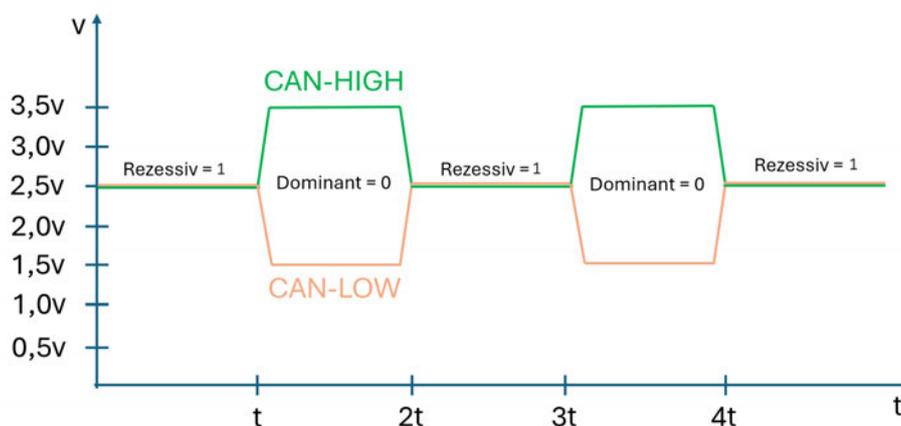


Abbildung 2.2.: Spannungspiegel im Highspeed-CAN-Bus [17].

<sup>3</sup>Teile dieses Kapitels wurden aus der eigenen Studienarbeit [17] übernommen und angepasst.

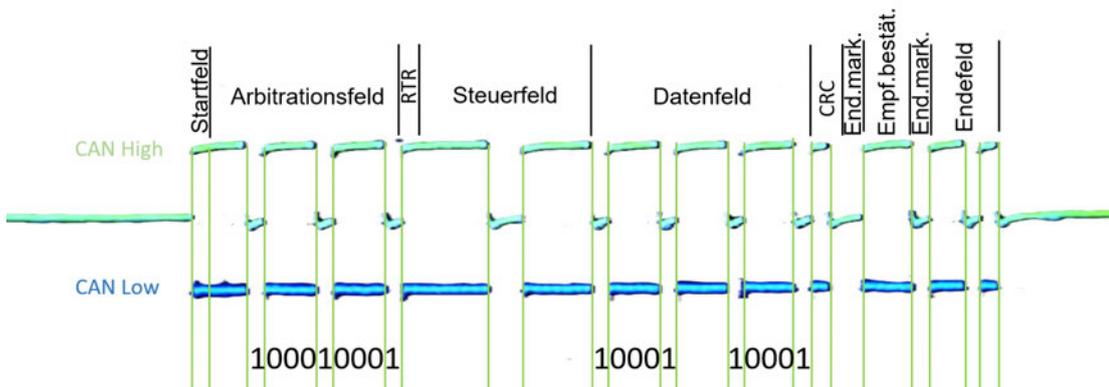


Abbildung 2.3.: Frameaufbau einer CAN-Nachricht (in HEX: ID 111, Data 11 11) [17].

Zur Vermeidung von Signalreflexionen und zur Reduktion von Störungen durch Umwelteinflüsse wird an jedem Ende der Busleitung ein Abschlusswiderstand von  $120\ \Omega$  zwischen CAN-High und CAN-Low angebracht.

Die auf der CAN-Leitung übertragenen Nachrichten werden als Frames bezeichnet und sind mit einem Oszilloskop lesbar. Um Störungen zwischen den Telegrammen zu vermeiden, sollte zwischen zwei Frames ein Mindestabstand von 2 ms eingehalten werden. Der Frame-Aufbau lässt sich als binäres Telegramm darstellen und besteht aus sieben nacheinander folgenden Feldern[18]:

- Startfeld (Startbit)
- Arbitrationsfeld (Identifier + RTR<sup>4</sup>-Bit)
- Steuerfeld (enthält den Datenlängencode)
- Datenfeld (0 bis 8 Bytes, max.  $2^8$  Werte pro Byte)
- CRC-8-Feld (Prüfsumme + Endemarkierung)
- Acknowledge-Feld (Bestätigungsbit + Endemarkierung)
- Endefeld (End-of-Frame)

Eine Standard-CAN-Nachricht verwendet einen 11-Bit-Identifier, entsprechend dem Format CAN 2.0A. Dies erlaubt  $2^{11} = 2048$  eindeutige Identifier. Das Datenfeld umfasst 0 bis 8 Bytes.

<sup>4</sup>RTR steht für „Remote Transmission Request“. Ein gesetztes RTR-Bit signalisiert eine Datenanforderung statt einer Datenübertragung. Bei RTR = 0 enthält der Frame Daten (Data Frame), bei RTR = 1 wird nur eine Anfrage gesendet (Remote Frame).

Jedes Byte kann Werte zwischen 0x00 und 0xFF annehmen, was 256 mögliche Werte pro Byte ergibt (s. Abbildung 2.4).

```

                                     ID      Byte 0  Byte 1  Byte 2  Byte 3  Byte 4  Byte 5  Byte 6
send_can_message("297", "A2EBF1E0F0F0FC")
send_can_message("0D7", "43FF")
send_can_message("36E", "48F8FEFF15")
send_can_message("21A", "010000")
```

Abbildung 2.4.: Darstellung der Byte-Positionen [17].

Die wichtigsten Informationen für Steuergeräte sind die ID- und Data-Frames auf Softwareebene, die in der Regel als hexadezimale Zahlen dargestellt werden (s. Tabelle 2.1). Ein anschaulicher Vergleich mit einer Straße verdeutlicht die Funktionsweise des CAN-Netzwerks: Die Straße repräsentiert den CAN-Bus, die Häuser entlang dieser Straße stehen für Steuergeräte, während die einzelnen Wohnungen in den Häusern Sensoren und Aktoren mit spezifischen Adressen den ID symbolisieren. Die übertragenen Daten entsprechen offenen Briefen, die theoretisch von jeder Wohnung gelesen werden können [9].

Zahlensystem	Beispiel	Schreibweise
Dezimal	255	255
Hexadezimal	255 (dezimal)	0xFF
Binär	255 (dezimal)	0b11111111

Tabelle 2.1.: Schreibweisen von Dezimalen, Hexadezimalen und binären Zahlen [17].

Ein großer Vorteil besteht darin, dass mehrere Steuergeräte, etwa eine ECU und ein Kombiinstrument, dieselbe ID auswerten können. So lassen sich beispielsweise Radgeschwindigkeiten von mehreren Geräten gleichzeitig verwenden, ohne dass redundante ID vergeben werden müssen. Eine genauere Analyse der Informations-Frames erfolgt in Kapitel 6.2.

## 2.5. Übersicht über das AUDEx-Projekt

Das AUDEx-Projekt der Hochschule für Angewandte Wissenschaften Hamburg ermöglicht die Entwicklung und Erprobung moderner Technologien in der Fahrzeugtechnik an Modellfahrzeugen im Maßstab 1:8 und 1:5. Ziel ist es, reale Fahrzeugsysteme und -funktionen auf ein kleineres Modell zu übertragen und praxisnah zu testen [7].

## *2. Grundlagen*

---

Das Projekt umfasst verschiedene Bereiche wie Fahrassistenzsysteme, elektrische Antriebstränge und Prüfstandsentwicklung. Das AUDEx-Projekt bietet eine Plattform, um innovative Lösungen zu testen und Erkenntnisse für reale Anwendungen zu gewinnen.

## 3. Analyse und Konzeptentwicklung

Die Analyse und Konzeptentwicklung bildet einen zentralen Bestandteil dieser Arbeit. Ziel ist es, die bestehenden Technologien zur Geschwindigkeitsermittlung zu evaluieren, Anforderungen an ein optimiertes Messsystem zu formulieren und darauf aufbauend eine geeignete Lösung zu entwickeln. Dabei wird besonderes Augenmerk auf die Anwendbarkeit im AUDEx-Projekt gelegt.

### 3.1. Anforderungen an das AUDEx Messsystem

Ein Geschwindigkeitsmesssystem muss vielfältigen Anforderungen genügen, um präzise, zuverlässige und robust einsetzbare Daten zu liefern. Die folgenden Anforderungen wurden basierend auf den Anwendungen im AUDEx-Projekt formuliert:

- **Genauigkeit:** Das System muss Messfehler minimieren und konsistente Ergebnisse liefern, auch unter variierenden Bedingungen.
- **Robustheit:** Das Messsystem muss unter rauen Umgebungsbedingungen wie Vibrationen funktionsfähig bleiben.
- **Energieeffizienz:** Insbesondere bei elektrischen Modellfahrzeugen muss der Stromverbrauch minimal sein, um die Batteriekapazität nicht unnötig zu belasten.
- **Gewichtsoptimierung:** Die Messkomponenten dürfen das Gesamtgewicht des Modellfahrzeugs nur minimal beeinflussen, da dies die Fahreigenschaften verändern könnte.
- **Kompaktheit:** Die Integration des Systems in das Modellfahrzeug erfordert eine kleine Bauweise, insbesondere für Radmontagen.
- **Flexibilität:** Das entwickelte Messsystem ist so ausgelegt, dass es sowohl in LKW- als auch in PKW-Modellen des AUDEx-Projekts eingesetzt werden kann.

### 3.2. Vergleich bestehender Technologien

Für die Geschwindigkeitsmessung stehen unterschiedliche Technologien zur Verfügung, die jeweils spezifische Vor- und Nachteile aufweisen. Tabelle 3.1 bietet einen kompakten Vergleich.

Tabelle 3.1.: Vergleich bestehender Geschwindigkeitsmess-Technologien.

Technologie	Vorteile	Nachteile
GPS	Schlupffrei, hohe Präzision	Signalstörungen in Gebäuden oder dichten Wäldern, kostenintensiv
Radar/Lidar	Schlupffrei, hohe Präzision	Hoher Energieverbrauch, teuer
Induktive Spule	Robust, wartungsfrei, leicht	Anfällig für Verschmutzungen, vergleichsweise niedrige Präzision
Hall-Sensor	Schlupffrei, präzise, geringer Stromverbrauch, kompakt	Empfindlich gegenüber Magnetfeldern und externen Störungen
Messung an Motorphasen	Keine zusätzliche Teilemontage nötig, Steuersignal wird abgelesen	Kalibrierungsaufwand, Erstellung von eigenen Motorsteuergerät, Schlupf vorhanden

Auf Grundlage der Anforderungen und der Vor- und Nachteile scheinen Hall-Sensoren und induktive Spulen die geeignetsten Kandidaten für den Einsatz im AUDEx-Projekt zu sein. Beide Technologien sind kompakt, energieeffizient und robust, was sie besonders für den Modellmaßstab geeignet macht.

Die Signale können zwar zunächst mit einem Oszilloskop abgelesen werden, müssen jedoch anschließend von einem Mikrocontroller verarbeitet werden. Dieser sollte möglichst kompakt, kostengünstig und mit einer eigenen CAN-Bus-Bibliothek ausgestattet sein. In Tabelle 3.2 ist bereits eine grobe Übersicht dargestellt.

Mikrocontroller	Kosten	Gewicht	Größe	WLAN/BT	CAN-Bib.
ESP32-C3 Super Mini [16]	4€	3g	18x24mm	Ja / Ja	Ja
ESP32 WROOM [6]	8€	7g	29x52mm	Ja / Ja	Ja
STM32 Blue Pill [1]	10€	7g	23x53mm	Nein / Nein	Ja
RP2040-Zero [21]	10€	3g	18x25mm	Nein / Nein	Nein
FRC EP170 [2]	3169€	940g	142x113mm	Nein / Nein	Ja

Tabelle 3.2.: Vergleich verschiedener Mikrocontroller.

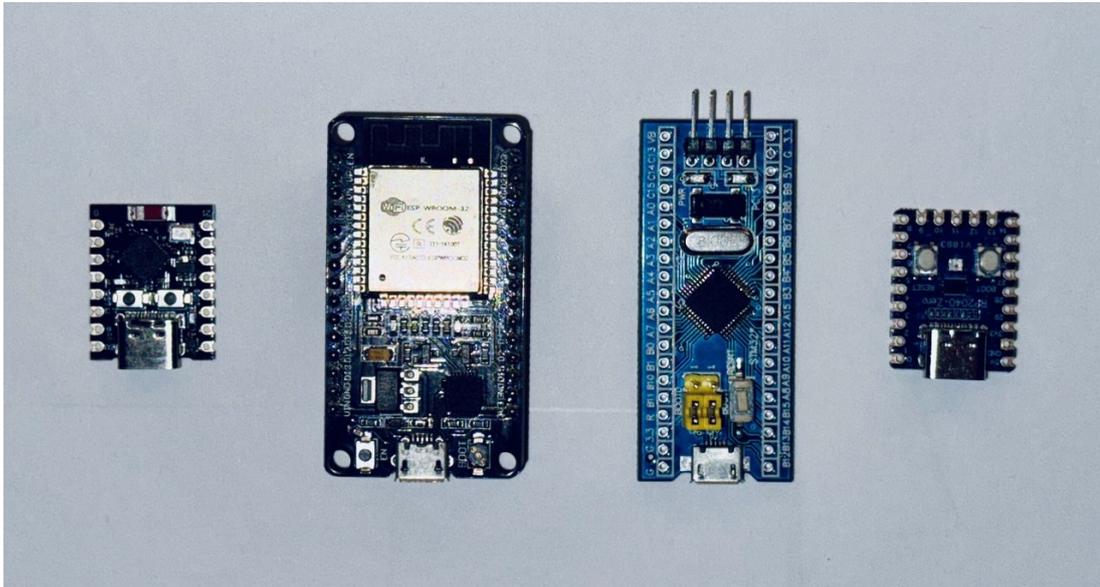


Abbildung 3.1.: v.l.n.r.: ESP32 Super Mini, ESP32 WROOM, STM32 Blue Pill, RP2040-Zero.

Da die AUDEx-Projekte üblicherweise auf der ESP32-Plattform basieren und dieser Mikrocontroller verschiedene Open-Source-CAN-Bus-Bibliotheken unterstützt, stellt insbesondere der ESP32 Super Mini (s. Abbildung 3.1) eine geeignete Wahl dar. Er überzeugt durch seine äußerst kompakten Abmessungen von lediglich 18x24mm, sein geringes Gewicht sowie den modernen USB-C-Anschluss, der eine zeitgemäße Schnittstelle zur Spannungsversorgung und Datenübertragung bietet. Darüber hinaus stellt der vergleichsweise niedrige Preis des ESP32 Super Mini einen weiteren Vorteil dar, der insbesondere bei kostenkritischen Anwendungen nicht vernachlässigt werden sollte.

### 3.3. Auswahl des Messprinzips

Nach der Analyse bestehender Technologien und der Anforderungen des AUDEx-Projekts wurde entschieden, Hall-Sensoren und induktive Spulen in der prototypischen Entwicklung einzusetzen. Ziel ist es, die Vor- und Nachteile dieser Technologien unter realen Einsatzbedingungen zu evaluieren. Die Entscheidung, ob die Sensoren an den Antriebswellen oder an den Rädern angebracht werden, bleibt vorerst offen. Diese Frage wird durch praktische Tests entschieden, bei denen Genauigkeit, Energieverbrauch, Gewicht und Montageaufwand eine Rolle spielen.

Die Vorteile dieser Vorgehensweise liegen in der Flexibilität und der Möglichkeit, die beste Lösung für das spezifische Fahrzeugdesign zu finden. Die Technologien bieten eine solide Grundlage für zukünftige Weiterentwicklungen.

## 3.4. Konzept des Messsystems

Das Messsystem wird modular aufgebaut, bestehend aus folgenden Komponenten:

### 1. *Sensorik:*

- Hall-Sensoren oder induktive Spulen zur Drehzahlerfassung.
- Temporäre Montage an Rädern und Antriebswellen, um die optimale Positionierung zu evaluieren.

### 2. *Datenverarbeitung:*

- Ein Microcontroller erfasst und verarbeitet die Signale der Sensoren in Echtzeit.

### 3. *Kommunikation:*

- Integration über den CAN-Bus ermöglicht die einfache Anbindung an das Fahrzeugsteuerungssystem.

### 4. *Software:*

- Eine flexible Architektur ermöglicht die parallele Verarbeitung mehrerer Sensoren oder Sensortypen.

Dieses Konzept ermöglicht eine detaillierte Vergleichsanalyse und bildet die Grundlage für die endgültige Auswahl der besten Messmethode.

### 3.4.1. Bewertung bestehender Prototypen zur Geschwindigkeitsermittlung

Das derzeit verwendete Verfahren zur Geschwindigkeitsermittlung weist gewisse Messungenauigkeiten auf und erweist sich insgesamt als wenig zuverlässig.

Ein erster auffälliger Punkt ist der Einsatz eines Hall-Sensors vom Typ TLE4935L. Dieser Sensor ist bipolar und benötigt eine gezielte Ausrichtung der Magnete mit abwechselndem Nord- und Südpol, um sicher zwischen HIGH- und LOW-Zustand zu schalten. Grundsätzlich stellt diese Funktionsweise keinen Fehler dar, jedoch reagieren derartige Sensoren empfindlich auf externe magnetische Störungen. Dies kann dazu führen, dass einzelne Magnete vom Sensor nicht erkannt bzw. übersprungen werden, was die Messgenauigkeit erheblich beeinträchtigt.

Selbst bei korrekt ausgerichteten Magneten zeigt sich ein weiteres Problem: Die verwendeten runden Magnete minderer Qualität verfügen oft über keine saubere Feldverteilung. Je nach Position relativ zum Sensor können an den Seitenflächen sowohl Nord- als auch Südpol gleichzeitig wirken. Dies kann selbst bei gleichmäßiger Rotation zu instabilem Schaltverhalten führen.

Ein zusätzlicher Schwachpunkt liegt im Abstand zwischen Sensor und Magneten. In der bisherigen Ausführung [4] ist dieser Abstand zu groß gewählt, was die Signalqualität weiter verschlechtert. Da es sich um ein Fahrzeugmodell im Maßstab 1:8 handelt, können bereits geringe Abweichungen zu spürbaren Messfehlern führen.

Weiterhin befindet sich in unmittelbarer Nähe zu den Magneten eine Metallschraube, die das lokale Magnetfeld beeinflusst und damit zusätzliche Messfehler verursachen kann [3].

Ein weiterer Aspekt, der im Verlauf dieser Arbeit experimentell untersucht werden soll, ist ein möglicher Spannungseinbruch oder ein kurzzeitiger Kurzschluss während des Schaltvorgangs des Hall-Sensors. Der Sensor wird über die 5 V-Versorgung des ESP32 betrieben. Da es sich um einen Open-Collector-Ausgang<sup>1</sup> handelt, wird der Ausgang beim Schalten gegen GND<sup>2</sup> gezogen. Ohne einen geeigneten Pull-up-Widerstand kann dies zu Spannungseinbrüchen führen, die sich negativ auf die Spannungsversorgung des Mikrocontrollers auswirken könnten.

Außerdem wird in [4] ein Spannungsteiler aus zwei Widerständen verwendet. Bei der Nachbildung der Schaltung fiel jedoch auf, dass die abgebildeten Widerstandswerte nicht im Standard-E12-Sortiment erhältlich sind. Diese müssten daher aufwändig durch eine Kombination aus Parallel- und Reihenschaltung realisiert werden.

Konkret handelte es sich um die Werte 1.15 k $\Omega$  und 3.09 k $\Omega$ , die dazu dienen sollten, den GPIO-Eingang<sup>3</sup> des ESP32 vor Überspannung zu schützen, da der Motorregler bzw. Motortreiber eine Versorgungsspannung von 5 V liefert. Die resultierende Spannung am GPIO berechnet sich folgendermaßen:

$$U_{\text{GPIO}} = \frac{R_2}{R_1 + R_2} \cdot V_{\text{in}} = \frac{3.09 \text{ k}\Omega}{1.15 \text{ k}\Omega + 3.09 \text{ k}\Omega} \cdot 5 \text{ V} = \frac{3.09}{4.24} \cdot 5 \text{ V} \approx 0.728 \cdot 5 \text{ V} \approx 3.64 \text{ V}$$

Dieser Wert überschreitet die für den ESP32 spezifizierte maximale Eingangsspannung von 3.3 V, was auf Dauer zu einer Beschädigung des Mikrocontrollers führen kann.

---

<sup>1</sup>Ein Open-Collector-Ausgang ist ein Ausgangstyp, bei dem der Transistor im Ausgang offen bleibt und nur nach Masse (GND) schalten kann. Ein externer Pull-up-Widerstand ist notwendig, um ein definiertes High-Potenzial zu erzeugen.

<sup>2</sup>GND bezeichnet den elektrischen Massebezug bzw. 0-V-Potential in Schaltungen.

<sup>3</sup>GPIO bezeichnet universell verwendbare Ein- oder Ausgangspins eines Mikrocontrollers, die digital und teils analog geschaltet oder gelesen werden können.

### *3. Analyse und Konzeptentwicklung*

---

In der Arbeit [4] wird ein Spannungsregler vom Typ L4940V5 verwendet. Es ist zu prüfen, ob dieser Regler bei einer Eingangsspannung von 6 V tatsächlich eine stabile Ausgangsspannung von 5 V zur Versorgung des ESP32 liefern kann. Die 6 V stammen aus dem Motorregler bzw. -treiber. Bei plötzlicher Beschleunigung steigen die Stromstärken deutlich an, was zu einem temporären Spannungsabfall führen kann.

Alle hier gewonnenen Erkenntnisse werden in dieser Arbeit berücksichtigt und fließen in die Konzeption eines verbesserten Prototyps ein.

## 4. Hardware-Entwicklung

In diesem Kapitel wird die Geschwindigkeitsmessung mithilfe von Hall-Sensoren und induktiven Sensoren detailliert untersucht. Es wird außerdem die Möglichkeit betrachtet, einen eigenen 3-Phasen-Treiber zu entwickeln, um die Motordrehzahl direkt auszulesen. Beide Sensortechnologien überzeugen durch ihre kompakte Bauweise und ihren geringen Energieverbrauch, wodurch sie sich besonders gut für den Einsatz in Modellfahrzeugen eignen.

Diese Sensoren gewährleisten eine zuverlässige und schlupffreie Ermittlung der Geschwindigkeit, die sowohl bei niedrigen als auch bei hohen Geschwindigkeiten präzise Ergebnisse liefert. Die Entwicklung eines eigenen Motorsteuergeräts könnte die Notwendigkeit solcher Sensoren teilweise überflüssig machen, da die Motordrehzahl vorgegeben wird und diese unter Berücksichtigung des Raddurchmessers direkt zur Berechnung der Geschwindigkeit genutzt werden kann.

Das Sensorkonzept basiert auf der parallelen Verwendung von Hall-Sensoren und selbstgewickelten induktiven Spulen. Diese Technologien bieten eine präzise Drehzahlerfassung bei minimalem Stromverbrauch und ermöglichen eine flexible Anpassung an die physikalischen Anforderungen eines Modellfahrzeugs. Ziel ist es, die Vorteile beider Methoden zu nutzen und ihre Leistung in verschiedenen Montageszenarien zu vergleichen. Um das komplette Hardware-Redesign für den jeweiligen Sensor zu erstellen, werden jedoch weitere Bauteile benötigt; diese werden im kommenden Kapitel beschrieben.

### 4.1. Asynchron Treiber

Ein Treiber für einen bürstenlosen Gleichstrommotor kann als eine Anordnung aus sechs Schaltern betrachtet werden (s. Abbildung 4.1). Durch die kontinuierliche Änderung des Drehfelds im Stator (hier das Außengehäuse) wird der Rotor angeregt und bewegt sich in Richtung der aktivierten Spule. Zwar könnten die sechs Schalter theoretisch durch mechanische Relais ersetzt werden, jedoch sind Relais nicht in der Lage, den hohen Drehzahlen standzuhalten. Daher sind Hochleistungs-MOSFETs<sup>1</sup> die deutlich bessere Wahl. Für die Steuerung werden

---

<sup>1</sup>Als MOSFET bezeichnet man einen spannungsgesteuerten Leistungstransistor, der häufig in der Ansteuerung von Motoren oder Schaltvorgängen eingesetzt wird.

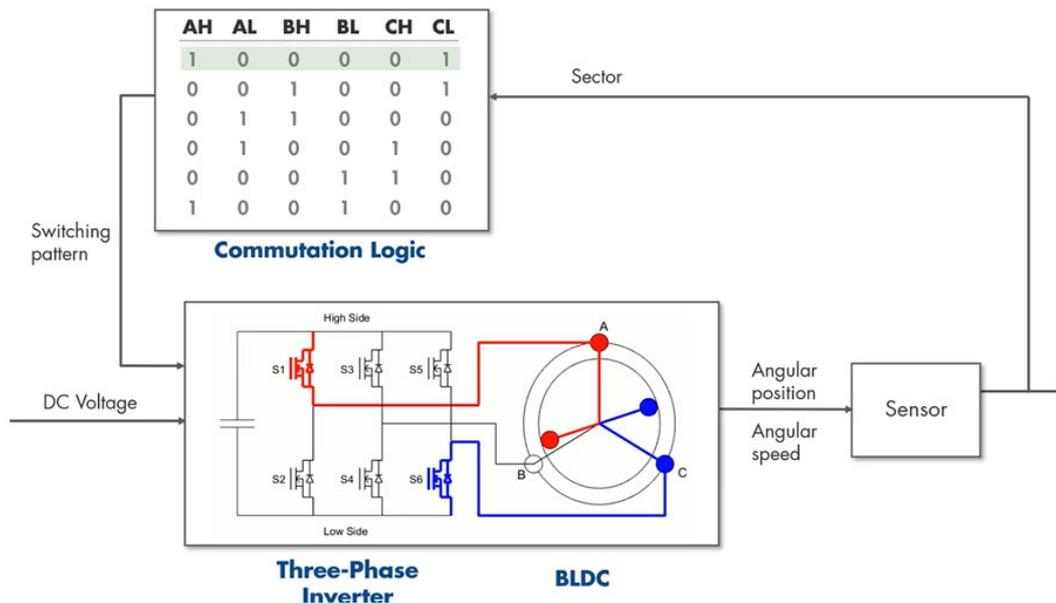


Abbildung 4.1.: Drei Phasen Inverter in Simscape, MathWorks [19].

sogenannte H-Brücken benötigt, jeweils eine pro Phase. Jede H-Brücke besteht aus einem n-MOSFET für die Low-Seite, der die Verbindung zwischen dem Minuspol und der Motorspule herstellt. Und einem p-MOSFET<sup>2</sup> für die High-Seite, der den Pluspol mit der Motorspule verbindet. Die n-MOSFETs (Logic-Level-MOSFETs)<sup>3</sup> können direkt mit den GPIO-Pins des ESP32 bei 3,3 V angesteuert werden. Die p-MOSFETs hingegen benötigen eine höhere Schaltspannung von etwa 5 V. Diese Spannung kann aus der Stromversorgung des ESP32 abgegriffen werden, allerdings sind dafür zusätzliche Transistoren notwendig, die als Schaltverstärker fungieren.

Ohne eine geeignete Signalsteuerung könnte ein quadratisches Schaltsignal entstehen, das den Motor ineffizient antreibt, zu Überhitzung führt und im schlimmsten Fall Wicklungskurzschlüsse verursacht. Glücklicherweise bietet der ESP32 die Möglichkeit, PWM-Signale zu erzeugen. Diese PWM-Signale simulieren durch wechselnde schnelle und langsame Impulse eine Sinuswelle, die für die Motorsteuerung notwendig ist. Die Phasen dieser Sinuswellen werden jeweils um 120 Grad verschoben, wie in der Abbildung 4.2 dargestellt.

Da das Oszilloskop nur zwei Anschlüsse hat, sind im Bild lediglich zwei der drei Phasen sichtbar. Um die Sinuswelle deutlich sichtbar zu machen, wird ein Filter aus einem Kondensator

<sup>2</sup>Ein p-MOSFET (positiver Typ) leitet, wenn die Gate-Spannung gegenüber Source negativ ist. Er wird häufig auf der High-Side verwendet, z. B. zur Lastabschaltung.

<sup>3</sup>Logic-Level-n-MOSFETs sind spezielle Transistoren, die bereits bei niedrigen Gate-Spannungen von etwa 3,3V bis 5V vollständig durchschalten. Dadurch eignen sie sich besonders gut für den direkten Anschluss an Mikrocontroller-Ausgängen.

und einem Widerstand verwendet. Das Signal hat eine Spannung von 3,3 V, was für die Ansteuerung der MOSFET-Schaltung ausreichend ist.

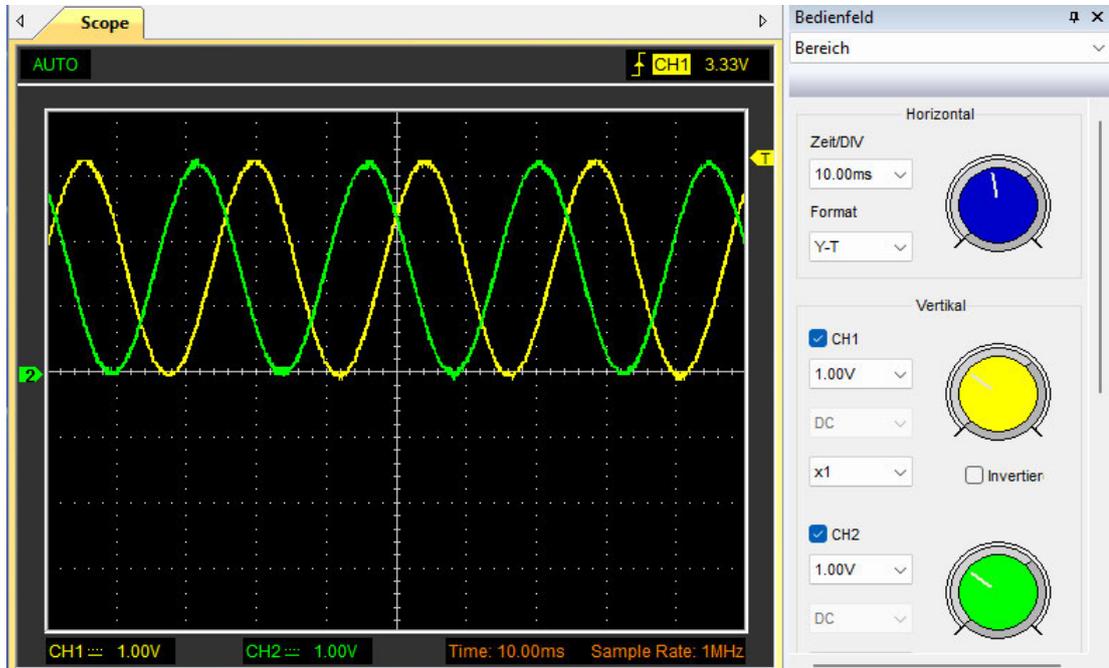


Abbildung 4.2.: Typisches Aussehen von um 120 Grad verschobenen Sinuswellen (zwei Phasen).

Um die Geschwindigkeit des Fahrzeugs aus den Informationen des Inverters/Treibers zu bestimmen, genügt es zwei aufeinanderfolgende logische Signale als eine Umdrehung des Motors zu interpretieren. In Abbildung 4.1 ist im Fenster *Commutation Logic* ein Beispiel solcher logischen Signale zu finden.

Um die Fahrzeuggeschwindigkeit aus der Motordrehzahl zu berechnen, müssen die Übersetzungsverhältnisse des Getriebes, des Differentials sowie der Raddurchmesser berücksichtigt werden. Zusätzlich wird zur Umrechnung in km/h ein Faktor von 3,6 verwendet. Die Geschwindigkeit ergibt sich aus folgender Formel:

$$v = \omega_{\text{Motor}} \cdot \pi \cdot d_{\text{Rad}} \cdot \frac{1}{i_{\text{Getriebe}}} \cdot \frac{1}{i_{\text{Differential}}} \cdot 3.6$$

Dabei ist:

- $v$ : Fahrzeuggeschwindigkeit in km/h
- $\omega_{\text{Motor}}$ : Motordrehzahl in 1/s (Umdrehungen pro Sekunde)

- $i_{\text{Getriebe}}$ : Getriebeübersetzungsverhältnis (z. B. 14:42 entspricht 3.0 [5])
- $i_{\text{Differential}}$ : Differentialübersetzungsverhältnis (z. B. 10:43 entspricht 4.3 [5])
- $d_{\text{Rad}}$ : Raddurchmesser in Metern

Beispielrechnung mit folgenden Werten:

- $d_{\text{Rad}} = 0.10 \text{ m}$ ,
- $\omega_{\text{Motor}} = 10/\text{s}$ ,
- $i_{\text{Getriebe}} = 3$ ,
- $i_{\text{Differential}} = 4.3$

Die Fahrzeuggeschwindigkeit ergibt sich zu:

$$\begin{aligned}v &= \omega_{\text{Motor}} \cdot \pi \cdot d_{\text{Rad}} \cdot \frac{1}{i_{\text{Getriebe}}} \cdot \frac{1}{i_{\text{Differential}}} \cdot 3.6 \\ &= 10/\text{s} \cdot \pi \cdot 0.10 \text{ m} \cdot \frac{1}{3} \cdot \frac{1}{4.3} \cdot 3.6 \\ &\approx 0.8767 \text{ km/h}\end{aligned}$$

Im Falle, dass zwei aufeinanderfolgende logische Einsen eines beliebigen Schalters bzw. MOSFETs aus der Tabelle *Commutation Logic* (s. Abbildung 4.1) zehnmal pro Sekunde auftreten, lässt sich daraus schließen, dass sich das Fahrzeug mit einer Geschwindigkeit von etwa 0.88 km/h bewegt.

Leider weist so ein Verfahren eine Abweichung in Drehzahl des Motors zu Drehzahl des Magnetfeldes im Stator auf, dieser heißt Schlupf und ist ein Grund für den Namen des Asynchronmotors. Je nach Drehzahl unterscheidet sich der Schlupf. Um diesen in Berechnungen zu berücksichtigen, müsste dieser unter verschiedenen Lasten für verschiedenste Motoren experimentell ermittelt werden.

## 4.2. Funktionsprinzip der induktiven Spule

Generell entsteht in jeder Leitung, die durch Strom durchflossen wird, ein Magnetfeld. Die Richtung kann mit der rechten Handregel bestimmt werden.

Eine induktive Spule erzeugt eine Spannung  $U_{\text{ind}}$ , wenn ein ferromagnetisches Zahnrad vorbeiläuft und den magnetischen Fluss  $\Phi$  ändert. Dies wird durch das Faraday'sche Induktionsgesetz beschrieben:

$$U_{\text{ind}} = -N \cdot \frac{d\Phi}{dt}$$

wobei:

- $N$ : Anzahl der Windungen,
- $\Phi$ : Magnetischer Fluss.

Der magnetische Fluss  $\Phi$  wird durch die Beziehung

$$\Phi = B \cdot A$$

bestimmt, wobei:

- $B$ : Magnetische Flussdichte, gemessen in Tesla (T),
- $A$ : Querschnittsfläche der Spule, gemessen in Quadratmetern (m<sup>2</sup>).

Der ferromagnetische Kern der Spule konzentriert das Magnetfeld und verbessert die Induktion, was die Sensorempfindlichkeit erhöht.

#### 4.2.1. Verbesserung der Empfindlichkeit

Der Effekt der Spannungserzeugung kann weiter erhöht werden, indem anstelle eines Stahlzahnrad eine rotierende Scheibe mit eingebetteten Dauermagneten verwendet wird. Dies bietet folgende Vorteile:

*Konstante und stärkere Magnetfelder:* Dauermagnete erzeugen ein stabileres und stärkeres Magnetfeld als die temporäre Magnetisierung eines Stahlzahnrad.

*Erhöhung der Flussänderungsrate:* Die Änderung des magnetischen Flusses ( $\frac{d\Phi}{dt}$ ) ist bei einem System mit Magneten größer, da das Magnetfeld bei der Bewegung der Magnete schneller wechselt.

Dieses Prinzip ähnelt stark einem Generator mit Dauermagneten. Der Unterschied besteht darin, dass der magnetische Fluss  $\Phi$  durch rotierende Dauermagnete erzeugt wird, die an einer Achse befestigt sind. Die rotierenden Magnete induzieren eine Wechselspannung in den stationären Spulen des Generators.

Werden stärkere Magnete, höhere Frequenzen oder mehr Windungen angewendet, wächst die induzierte Spannung entsprechend. Ein erheblicher Multiplikator in diesem Fall ist der Spulenkern. Der Kern kann:

- **völlig fehlen (Luftkern):** Der Magnetische Flussdichte bleibt unverändert ( $B = B_{\text{magnetisch}}$ ).
- **aus Stahl bestehen:** Hier wird der magnetische Fluss verstärkt, abhängig von der relativen Permeabilität ( $\mu_r$ ).
- **aus ferromagnetischem Stahl bestehen:** Die Permeabilität ist noch höher und verstärkt  $B$  entsprechend.

**Relative Permeabilitätswerte:** Die  $\mu_r$ -Werte für verschiedene Kernmaterialien sind in Tabelle 4.1 dargestellt:

Material	Relative Permeabilität ( $\mu_r$ )
Luft (ohne Kern)	1
Normaler Stahl	1000
Ferromagnetischer Stahl	2000

Tabelle 4.1.: Relative Permeabilität ( $\mu_r$ ) für verschiedene Materialien.

Die induzierte Spannung  $U_{\text{ind}}$  wird entsprechend der Permeabilität des Kerns berechnet:

$$B_{\text{eff}} = B \cdot \mu_r$$

$$U_{\text{ind}} = -N \cdot B_{\text{eff}} \cdot A \cdot f_{\text{eff}}$$

wobei:

- $B_{\text{eff}}$ : Magnetische Flussdichte im Kern,
- $N$ : Anzahl der Windungen,
- $A$ : Querschnittsfläche der Spule,
- $f_{\text{eff}}$ : Effektive Frequenz (abhängig von Drehzahl und Anzahl der Magneten).

Dies bringt nicht ausschließlich Vorteile mit sich, denn eine induzierte Spannung von über 3.3 V kann die GPIO-Pins des ESP32 beschädigen.

Beispiel für eine theoretisch induzierte Spulenspannung:

Für die Berechnung der induzierten Spannung (s. Tabelle 4.2) werden die folgenden Werte und Formeln verwendet:

- Anzahl der Spulenwindungen:  $N = 500$
- Drahtdurchmesser:  $d = 0,2 \text{ mm} = 0,2 \times 10^{-3} \text{ m}$
- Länge des Ferromagneten:  $l = 2 \text{ cm} = 0,02 \text{ m}$
- Durchmesser des Ferromagneten:  $r = 2,5 \text{ mm} = 0,0025 \text{ m}$
- Magnetische Flussdichte von Neodym-Magnet:  $B = 1,2 \text{ T}$
- Anzahl der Magnete pro Umdrehung: 3
- Drehzahlen ( $f$ ): 2, 5, 10, 20, 40, 80, 160 U/s

Nach Formeln:

- Fläche des Ferromagneten:

$$A = \pi \cdot r^2$$

- Änderung des magnetischen Flusses:

$$\frac{d\Phi}{dt} = B \cdot A \cdot f_{\text{eff}}$$

wobei die effektive Frequenz:

$$f_{\text{eff}} = f \cdot \text{Magneten pro Umdrehung}$$

- Induzierte Spannung:

$$U_{\text{ind}} = -N \cdot \frac{d\Phi}{dt} = -N \cdot B \cdot A \cdot f_{\text{eff}}$$

Beispielrechnung für  $f = 2 \text{ U/s}$ , ohne Kern:

- Fläche des Ferromagneten:

$$A = \pi \cdot (0.0025)^2 = 1.9635 \times 10^{-5} \text{ m}^2$$

- Effektive Frequenz:

$$f_{\text{eff}} = 2 \cdot 3 = 6 \text{ Hz}$$

- Magnetischer Flusswechsel pro Sekunde:

$$\frac{d\Phi}{dt} = 1.2 \text{ T} \cdot 1.9635 \times 10^{-5} \text{ m}^2 \cdot 6 \text{ Hz} = 1.415 \times 10^{-4} \text{ Wb/s}$$

- Induzierte Spannung:

$$U_{\text{ind}} = -500 \cdot 1.415 \times 10^{-4} \text{ Wb/s} = -0.07075 \text{ V}$$

Drehzahl (U/s)	Ohne Kern (V)	Mit Stahlkern (V, $\mu_r = 1000$ )	Mit ferromagnetischem Kern (V, $\mu_r = 2000$ )
2	-0,07075	-70,75	-141,5
5	-0,176875	-176,875	-353,75
10	-0,35375	-353,75	-707,5
20	-0,7075	-707,5	-1415
40	-1,415	-1415	-2830
80	-2,83	-2830	-5660
160	-5,66	-5660	-11320

Tabelle 4.2.: Theoretisch induzierte Spannungen für unterschiedliche Kerne bei 500 Windungen und unterschiedlichen Drehzahlen.

#### 4.2.2. Induktive Spule Zusammenfassung

Die Verwendung von rotierenden Dauermagneten anstelle eines Stahlzahnrad in Kombination mit induktiven Spulen erhöht die Effizienz und verbessert die Ausgangsspannung der Sensoren erheblich. Allerdings führt diese Erkenntnis zu zwei Herausforderungen: Niedrige Drehzahlen werden vom GPIO-Eingang des ESP32 nicht erkannt, während hohe Drehzahlen das ESP32 durch Überspannung beschädigen können. In Abbildung 4.3 sind deutlich drei Ausschläge zu erkennen, die durch drei Magnete verursacht werden, die auf einem Lüfterrad montiert

sind. Da der Lüfter sieben Blätter besitzt, aber nur mit drei Magneten bestückt ist, entsteht ein Abstand zwischen den einzelnen Ausschlägen (s. Abbildung 4.4).

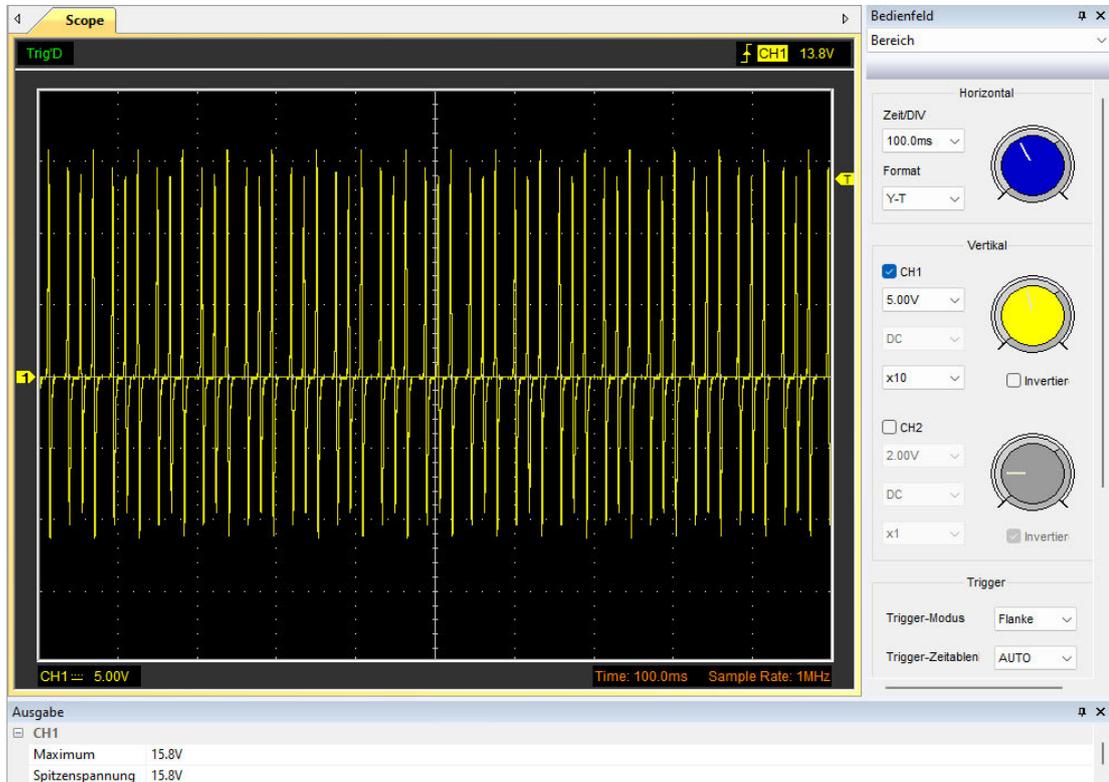


Abbildung 4.3.: Spule mit drei Magneten.

Das Signal, das mit einem Oszilloskop<sup>4</sup> gemessen wurde, ist analog und daher für Weiterverarbeitungszwecke nicht optimal. In diesem Fall werden bei einer Drehgeschwindigkeit von 18 Umdrehungen pro Sekunde 15,8 V induziert. Diese Spannung ist jedoch zu hoch und würde den GPIO-Eingang des ESP32 beschädigen, wenn keine Schutzmaßnahmen ergriffen werden.

## 4.3. Spulenkonstruktion und Berechnung

### 4.3.1. Magnetischer Fluss und Induktion

Der magnetische Fluss  $\Phi$  in einer Spule wird berechnet als:

$$\Phi = B \cdot A$$

<sup>4</sup>Das verwendete Oszilloskop ist ein Hantek 6022BE.

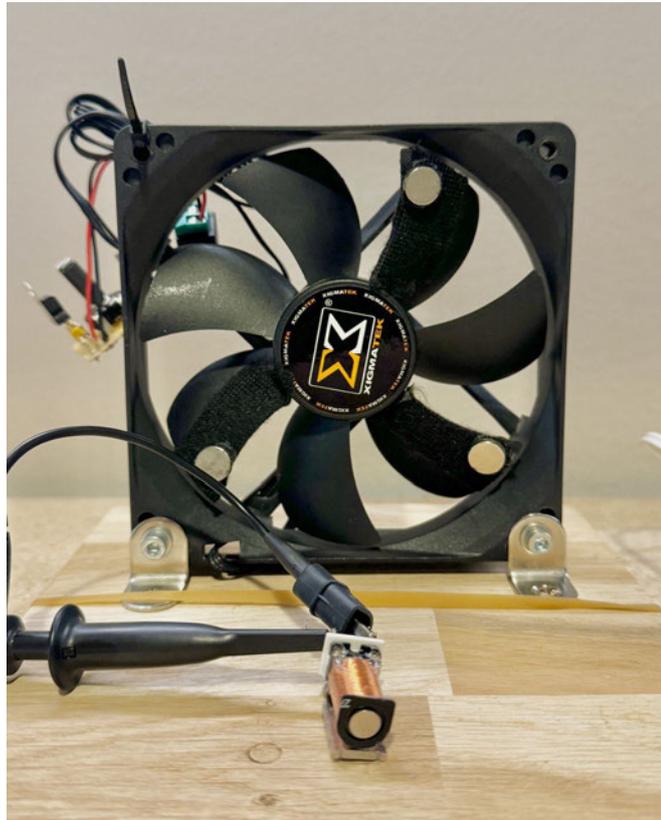


Abbildung 4.4.: Lüfter mit drei Magneten (unten ist die Spule zu sehen).

wobei: -  $B = \mu_0 \cdot \mu_r \cdot H$ , die magnetische Flussdichte, -  $A$ : Querschnitt des Kerns, -  $\mu_0 = 4\pi \cdot 10^{-7} \text{ H/m}$ , die magnetische Feldkonstante, -  $\mu_r$ : Relative Permeabilität des Kerns, -  $H$ : Magnetische Feldstärke.

#### 4.3.2. Induktivität der Spule

Die Induktivität  $L$  wird berechnet mit:

$$L = \frac{\mu_0 \cdot \mu_r \cdot N^2 \cdot A}{l}$$

wobei: -  $N$ : Anzahl der Windungen, -  $l$ : Länge des magnetischen Pfads, -  $A$ : Querschnitt des Kerns.

Beispielberechnung:

Gegeben:

- $\mu_r = 2000$  (relative Permeabilität, Eisenkern),
- $N = 200$  (Windungszahl),
- $A = 0.5 \text{ cm}^2$  (Querschnittsfläche),
- $l = 5 \text{ cm}$  (Länge des Eisenkerns)

Berechnung der Induktivität  $L$  nach folgender Formel:

$$L = \frac{\mu_0 \cdot \mu_r \cdot N^2 \cdot A}{l}$$

Einsetzen der Werte:

$$L = \frac{4\pi \cdot 10^{-7} \cdot 2000 \cdot 200^2 \cdot 0.5 \times 10^{-4} \text{ m}^2}{0.05 \text{ m}}$$

$$L \approx 10.05 \text{ mH}$$

#### 4.3.3. Signalverarbeitung

Wie in Kapitel 4.2.1 gezeigt wurde, dass das induzierte Signal analog ist, wird dieses Problem mit Hilfe eines Schmitt-Triggers gelöst. Am besten eignen sich hierfür Bauteile wie der 74HC14, der aus einem analogen Signal ein digitales erzeugen kann. Da jedoch die Spannung des Signals den Arbeitsbereich des 74HC14 übersteigt, wird stattdessen ein ähnliches Bauteil verwendet: der LM339L.

Dabei handelt es sich um einen Vierfach-Komparator, der Spannungen miteinander vergleichen und ein digitales Ausgangssignal erzeugen kann. Die Funktionsweise ist wie folgt: Der Komparator vergleicht die Spannung des Eingangssignals (z. B. das analoge Signal der Spule) mit einer Referenzspannung. Liegt die Eingangsspannung über der Referenzspannung, schaltet der Ausgang auf High; liegt sie darunter, schaltet dieser auf Low. Dadurch wird aus dem sinusförmigen analogen Signal eine rechteckige digitale Welle.

Um das Signal zu stabilisieren und kleine Schwankungen im analogen Signal zu vermeiden, wird eine Hysterese<sup>5</sup> eingeführt. Dies geschieht durch eine geeignete Rückkopplung am

---

<sup>5</sup>Hysterese bezeichnet einen Bereich zwischen zwei Schaltschwellen, innerhalb dessen keine Änderung des Ausgangszustands erfolgt. Dadurch werden unerwünschte Schaltvorgänge durch kleine Signaländerungen vermieden.

LM339L. Der Ausgang des LM339L ist ein Open-Collector-Ausgang, was bedeutet, dass ein Pull-Up-Widerstand erforderlich ist, um ein korrektes High-Signal zu erzeugen.

Die Umwandlung des analogen Signals der Spule in ein digitales Rechtecksignal ermöglicht eine zuverlässige Weiterverarbeitung, insbesondere durch Mikrocontroller wie den ESP32. In Abbildung 4.5 wird der Vergleich zwischen der induzierten Spannung der Spule (in Grün) und dem Ausgangssignal des LM339L (in Gelb) dargestellt. Es sind deutliche Unterschiede zu erkennen: Die induzierte Spannung erreicht Spitzenwerte von bis zu 47,1 V (analoge Sinuskurve), während die Spannung am Ausgang des LM339L stabil bei 3,14 V liegt (digitale Rechteckwelle).

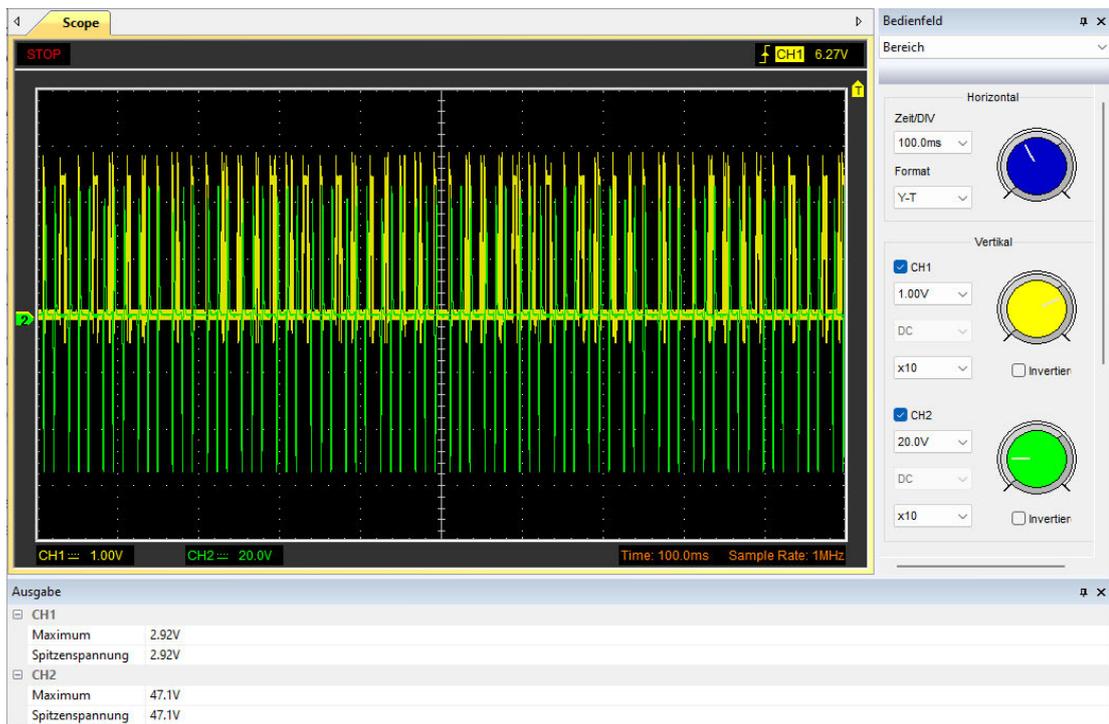


Abbildung 4.5.: Digitale Welle mit 2,92V aus LM339L und analoge Sinuskurve mit 47,1V aus der Spule.

In Abbildung 4.6 ist ein deutlicher Unterschied zwischen dem digitalen und dem analogen Signal zu erkennen. Zur Veranschaulichung wurde das Spannungsniveau im Oszilloskop entsprechend angepasst.

In Kapitel 4.2.1 wurde die Abhängigkeit der induzierten Spannung von der Drehzahl aufgezeigt. Bei geringeren Drehzahlen wird nur eine geringe Spannung induziert, was bedeutet, dass der Komparator diese niedrigeren Spannungen nicht wahrnehmen kann und dadurch unplausible Signale an den ESP32 weiterleitet.



Abbildung 4.6.: Digitale Welle und analoge Sinuskurve.

Da der Komparator lediglich die Spannung der Spule mit der Referenzspannung vergleicht, die aus der Versorgungsspannung des Komparators abgeleitet wird, muss die Schaltung entsprechend angepasst werden. Zu diesem Zweck wird eine Diode zwischen der Spule und dem Komparator eingesetzt, um die negative Halbwelle des Signals vorab abzuschneiden (s. Abbildung 4.7). Ein negativer Aspekt hierzu wäre, dass eine Diode vom Typ 1N4001 0,7 V verbraucht, was die Schaltung für niedrigere Drehzahlen noch unbrauchbarer macht.

Die Herausforderung mit zu niedrigen Spannungen bleibt, diese kann man jedoch mit dem LM1815 beseitigen. Dieser hat einen sehr großen Arbeitsbereich von 100 Millivolt bis zu 120 Volt und kann mit 2-12 V betrieben werden. Ein weiterer Vorteil ist, dass er einen Open-Collector-Ausgang hat, der die Spannung nach unten zieht, sodass kein Spannungsteiler benötigt wird. Er kann den GPIO-Eingang von 3,3 V auf 0 V ziehen. Der LM1815 ist ein Zero-Crossing-Detektor<sup>6</sup>, der speziell für die Verarbeitung von Signalen aus variablen Reluktanzsensoren<sup>7</sup> entwickelt

<sup>6</sup>Nulldurchgangsdetektor (ZCD) erkennt die Nulldurchgänge des Wechselspannungssignals einer Spule und gibt bei jedem Nulldurchgang ein digitales Signal aus. [11]

<sup>7</sup>Die auf dem Prinzip der variablen Reluktanz basierenden Drehzahlsensoren bestehen aus einem Eisenkern, einer induktiven Spule und einem Dauermagneten. Ein eisenhaltiges Polrad passiert die Sensorfläche und verändert so die Stärke des Magnetfelds, was dazu führt, dass in der Spule AC-Spannung erzeugt wird. [23]

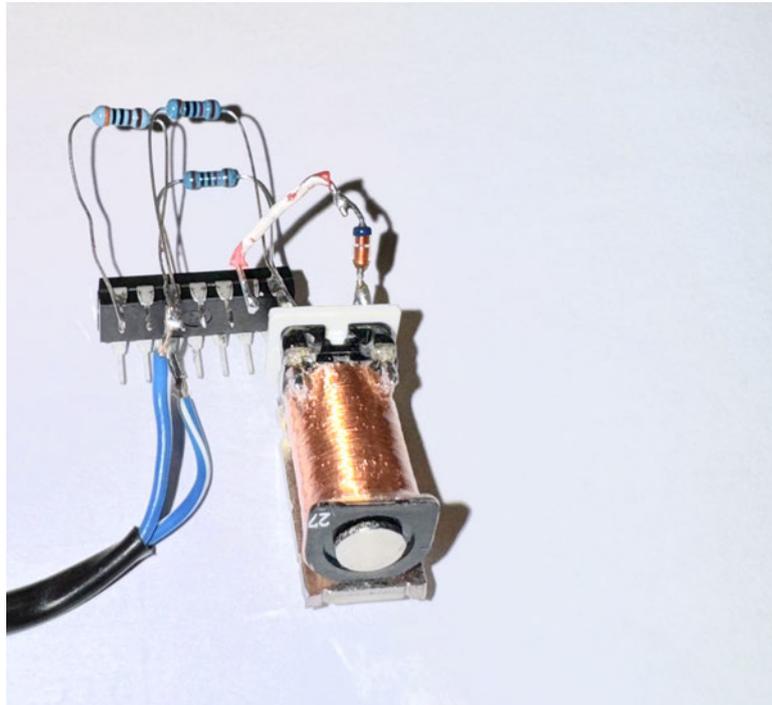


Abbildung 4.7.: Experimentelle Point-to-Point Verlötung der Diode zwischen Spule und Eingang des LM339.

wurde. Er erkennt die Null-Durchgänge des Wechselnungssignals, das vom Sensor erzeugt wird, und wandelt diese in ein digitales Ausgangssignal um. Dieses Signal reflektiert die Frequenz des Eingangssignals und kann direkt von einem Mikrocontroller wie dem ESP32 über einen GPIO-Pin gelesen werden. Da der LM1815 einen Open-Collector-Ausgang besitzt, wird der Ausgang mit einem Pull-up-Widerstand an 3,3 V (der Betriebsspannung des ESP32) angeschlossen. Dadurch ist das Ausgangssignal 3,3 V-kompatibel und sicher für den Mikrocontroller. Wenn der LM1815 einen Null-Durchgang erkennt, zieht er den Ausgang auf 0 V herunter. Dies ermöglicht eine zuverlässige und sichere Erfassung der Drehzahlssignale über den gesamten Spannungsbereich des Sensors, ohne dass zusätzliche Spannungsteiler oder Pegelanpassungen erforderlich sind. Einer der größten Nachteile des LM1815 ist sein Preis und seine Größe. Genauso wie der 74HC14 und der LM339 hat er in dem Fall eine Bauform von DIP-14<sup>8</sup> (s. Abbildung 4.8). Dies kann in Anwendungen mit begrenztem Platzangebot oder strengen Budgetvorgaben nachteilig sein.

---

<sup>8</sup>Dual in-line package oder auch zweireihiges Gehäuse - die 14 Pins werden in zwei Reihen aufgeteilt, Reihenabstand von 7,62 mm, Pinabstand von 2,54 mm.

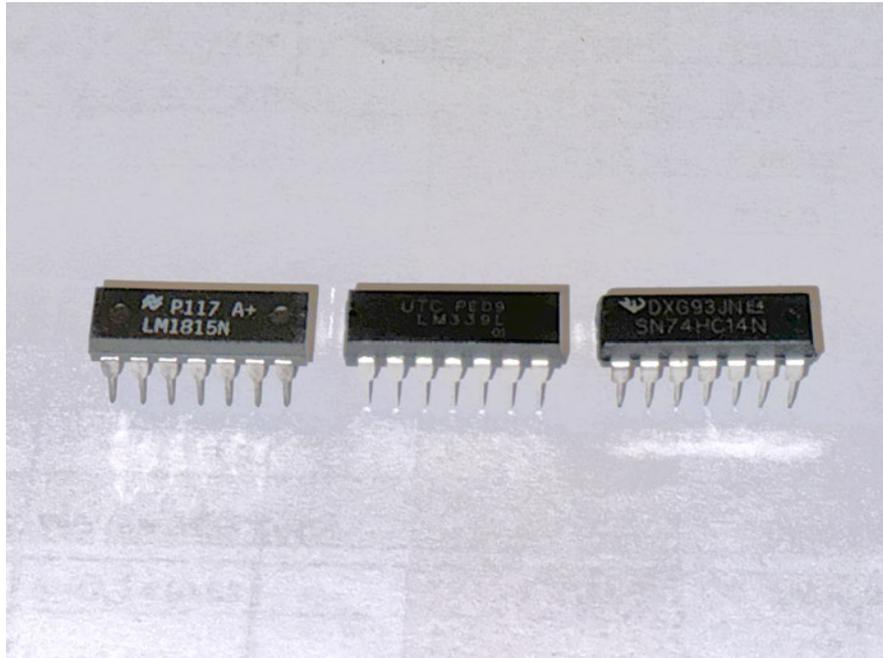


Abbildung 4.8.: Typisches Aussehen von DIP-14 Bauweise LM1815, LM339 und 74HC14.

#### 4.4. Funktionsprinzip des Hall-Sensors

Hall-Sensoren basieren auf dem Hall-Effekt, einem physikalischen Prinzip, das 1879 von Edwin Hall entdeckt wurde. Wenn ein elektrischer Strom durch ein leitfähiges Material fließt und gleichzeitig ein magnetisches Feld senkrecht zu diesem Strom angelegt wird, entsteht eine Spannung quer zur Stromrichtung [12]. Diese sogenannte Hall-Spannung ist proportional zur Stärke des Magnetfelds und erlaubt es, das Magnetfeld zu messen (s. Abbildung 4.9).

Die wesentlichen Bestandteile eines Hall-Sensors sind:

- **Hall-Platte:** Das Element, in dem der Hall-Effekt auftritt.
- **Signalaufbereitung:** Elektronische Schaltungen zur Verstärkung und Verarbeitung des Signals.
- **Ausgang:** Zur Weitergabe des Signals in analoger oder digitaler Form.

Hall-Sensoren können sowohl zur Messung von magnetischen Feldstärken als auch zur Positions-, Geschwindigkeits- oder Strommessung eingesetzt werden. Je nach Anwendung gibt es verschiedene Typen von Hall-Sensoren, wie lineare Hall-Sensoren, Schaltsensoren oder 3D-Hall-Sensoren.

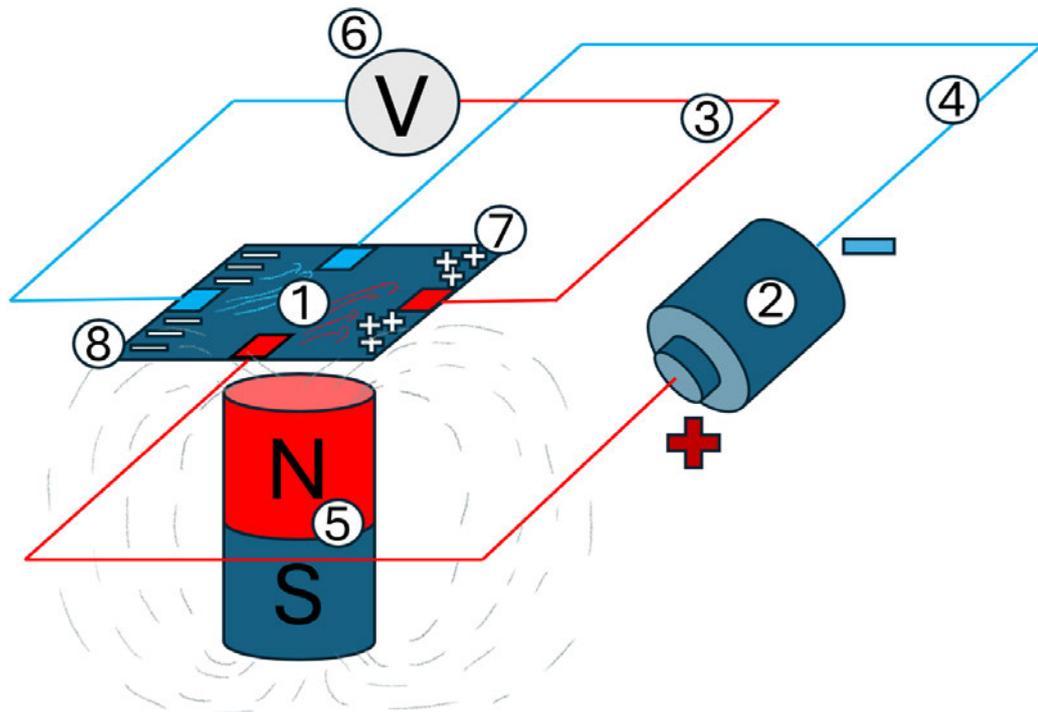


Abbildung 4.9.: Hall-Effekt: 1. Platte, 2. Stromquelle, 3. Pos. Leitung, 4. Neg. Leitung, 5. Magnet, 6. Voltmeter, 7. Abgelenkte Pos. Atome, 8. Neg. Atome.

#### 4.4.1. Vorteile von Hall-Sensoren

- **Kontaktlose Messung:** Kein mechanischer Verschleiß und hohe Zuverlässigkeit, da keine beweglichen Teile verwendet werden.
- **Vielseitigkeit:** Einsetzbar für die Messung von Position, Drehzahl, Strom und Magnetfeldstärke, wodurch zahlreiche Anwendungen abgedeckt werden können.
- **Kompakte Bauweise:** Einfach in elektrische und elektronische Systeme integrierbar, auch bei begrenztem Platzangebot.
- **Robustheit:** Hohe Widerstandsfähigkeit gegenüber Verschmutzung, Feuchtigkeit, Temperaturschwankungen und anderen Umwelteinflüssen.
- **Digitaler Ausgang:** Moderne Hall-Sensoren liefern oft einen digitalen Ausgang in Form eines präzisen Rechtecksignals, was die Weiterverarbeitung vereinfacht.

- **Integrierte Schaltkreise:** Integrierte Schaltkreise in Hall-Sensoren enthalten Operationsverstärker, Schalttransistoren und Schmitt-Trigger, was komplexe externe Schaltungen und Berechnungen überflüssig macht und Energie spart (s. Abbildung 4.10).

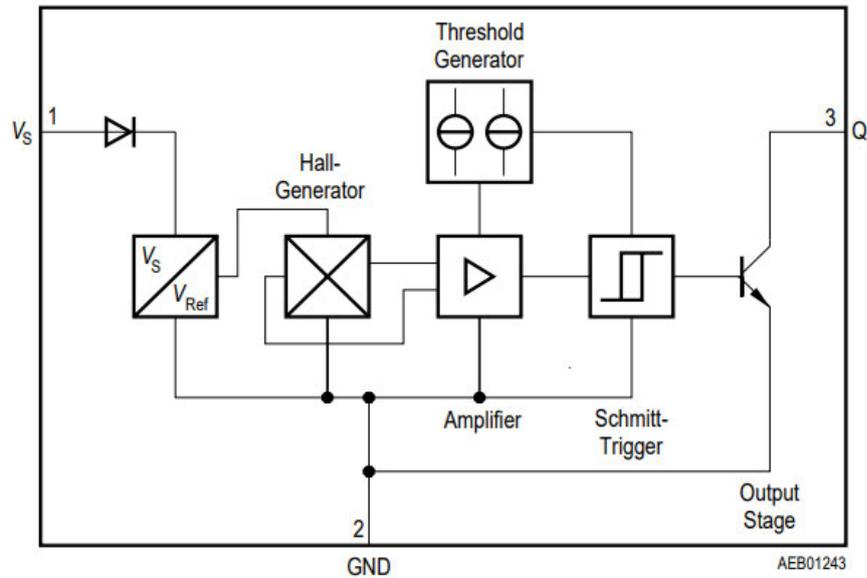


Abbildung 4.10.: Integrierte Schaltkreise eines TLE4905L von Fa. Infineon [13].

#### 4.4.2. Nachteile von Hall-Sensoren

- **Empfindlichkeit gegenüber Störungen:** Externe magnetische Felder können Messungen beeinflussen.
- **Begrenzte Temperaturbereiche:** Manche Sensoren funktionieren nur in einem eingeschränkten Temperaturfenster.
- **Kosten:** Hochpräzise Sensoren oder Spezialtypen sind teuer.
- **Signalrauschen:** Besonders bei linearen Hall-Sensoren verfälschen externe Einflüsse das Ausgangssignal.

Tabelle 4.3 stellt den Vergleich der Hall-Sensoren dar.

Typ	Vorteile	Nachteile
Linear	Hohe Empfindlichkeit, kostengünstig, einfach zu integrieren	Begrenzter Temperaturbereich, relativ langsame Reaktionszeit
Digital (Schalter)	Geringer Stromverbrauch, zuverlässig, robust	Nur Schaltbetrieb, keine analoge Ausgangsoption
3D-Hall-Sensor	Misst Magnetfelder in drei Achsen, hohe Präzision	Höhere Kosten, komplexere Integration
Linear	Weitreichender Temperaturbereich, hohe Empfindlichkeit	Etwas teurer als Standard-Hall-Sensoren
Stromsensor mit Hall	Geeignet für AC- und DC-Strommessungen, einfach in Schaltungen einzubinden	Begrenzte Messbereiche, anfällig für äußere Magnetfelder

Tabelle 4.3.: Vergleich verschiedener Hall-Sensoren.

#### 4.5. Hall-Sensor als prallfreier Taster

Hall-Sensoren messen die Änderung des Magnetfelds  $B$ , das durch rotierende Zahnräder, Magnete oder andere ferromagnetische Materialien erzeugt wird. Die Hall-Spannung  $U_H$  ist proportional zur Magnetfeldstärke  $B$  und wird durch folgende Gleichung beschrieben:

$$U_H = k_H \cdot B$$

wobei  $k_H$  die Hall-Konstante des verwendeten Materials ist. Diese Eigenschaft macht Hall-Sensoren zu präzisen und zuverlässigen Werkzeugen für verschiedene Anwendungen, insbesondere in der Positions- und Geschwindigkeitsmessung.

Für den Zweck der Geschwindigkeitsmessung wird häufig ein gängiger Hall-Sensor aus der Automobilindustrie, der TLE4905L, verwendet. Dieser Sensor zeichnet sich durch folgende Eigenschaften aus:

- **Unipolarer Betrieb:** Der Sensor reagiert nur auf Magnetfelder einer bestimmten Polarität, was die Signalverarbeitung vereinfacht.
- **Weiter Versorgungsspannungsbereich:** Funktioniert zuverlässig in einem Spannungsbereich von 3.8 V bis 24 V, wodurch er in zahlreichen Anwendungen einsetzbar ist.
- **Kompatibilität mit Mikrocontrollern:** Der Sensor kann problemlos mit einem Mikrocontroller wie dem ESP32 betrieben werden, da dieser ihn mit 5 V versorgen kann.

#### 4. Hardware-Entwicklung

---

Durch seine robuste und zuverlässige Bauweise eignet sich der TLE4905L (s. Abbildung 4.11) hervorragend für den Einsatz im Maschinenbau und Automotivbereich, da keine mechanischen Teile vorhanden sind, die Verschleiß unterliegen oder durch Kontaktprellen gestört werden.

Der Hall-Sensor erzeugt ein klares, digitales Ausgangssignal ohne Prellen, da er intern über eine integrierte Signalaufbereitung verfügt. Diese verhindert das typische Prellen mechanischer Schalter und sorgt dafür, dass Signale eindeutig und zuverlässig erkannt werden. Ein weiterer Vorteil dieser Entprellung ist, dass Mehrfachauslösungen bzw. mehrfaches Erfassen desselben Ereignisses vermieden werden – was insbesondere bei Zählimpulsen oder Rotationsmessungen wichtig ist. Dies führt zu einer längeren Lebensdauer und einer höheren Präzision im Betrieb. Die Gehäuse bietet auch die Möglichkeit, den Sensor mit Harz zu begießen und ein eigenes Gehäuse mit einer Montageöse zu bauen.

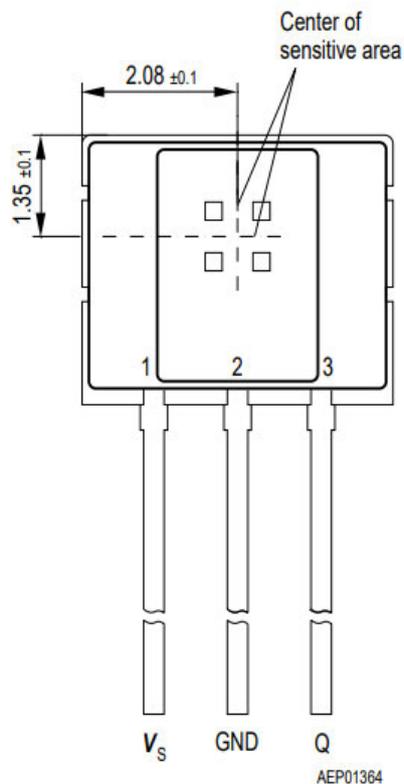


Abbildung 4.11.: TLE4905L von Fa. Infineon [13].

### 4.5.1. Arbeitsweise als Taster

Ein Hall-Sensor kann als berührungsloser Schalter eingesetzt werden, indem er das Vorhandensein oder die Änderung eines Magnetfeldes erkennt. Der Sensor reagiert auf einen Magneten, der in seiner Nähe bewegt wird und gibt ein digitales Signal aus. Diese Funktion ermöglicht eine entprellte Schalttechnik, da keine mechanischen Kontakte verwendet werden, die altern oder verschleifen können.

In Abbildung 4.12 sind drei Magnete dargestellt, die nacheinander von einem TLE4905L detektiert werden. Aufgrund der internen Schaltung des Hall-Sensors (z. B. mit einem integrierten Schmitt-Trigger) erzeugt dieser ein sauberes Rechtecksignal. Das Signal auf dem Oszilloskop zeigt keine Störungen oder Rauschen, was die hohe Qualität der Signalverarbeitung demonstriert.

Der TLE4905L ist ein digitaler Hall-Effekt-Sensor mit Open-Collector-Ausgang. Das bedeutet, dass er mit einer Versorgungsspannung von 5 V betrieben wird und einen Ausgang besitzt, der entweder auf HIGH (ca. 5 V) geht, wenn kein Magnetfeld erkannt wird, oder auf LOW (0 V), wenn ein Magnet detektiert wird.

Die Abstände zwischen den Signalen entsprechen der Anzahl der Magnete, die auf einem Lüfterrad mit sieben Blättern angebracht sind. Ein längerer Abstand zwischen zwei Signalen entsteht, weil sich dort keine Magnete befinden. Dies verdeutlicht, wie präzise der Sensor die Magnetfelder detektiert.

Die Kästchen auf der Oszilloskop-Oberfläche sind jeweils 100 ms breit, sodass zehn Kästchen einer Dauer von 1 s entsprechen. Bei einer Aufzeichnung von 32 Ausschlägen durch drei Magnete ergibt sich eine Frequenz von 11.7 Hz, was 11.7/s entspricht. Zur besseren Verdeutlichung wurden die Abstände farblich hervorgehoben: Rot und Grün stehen jeweils für den Abstand eines Lüfterblatts, während Lila den Abstand von zwei Lüfterblättern repräsentiert (s. Abbildung 4.12).

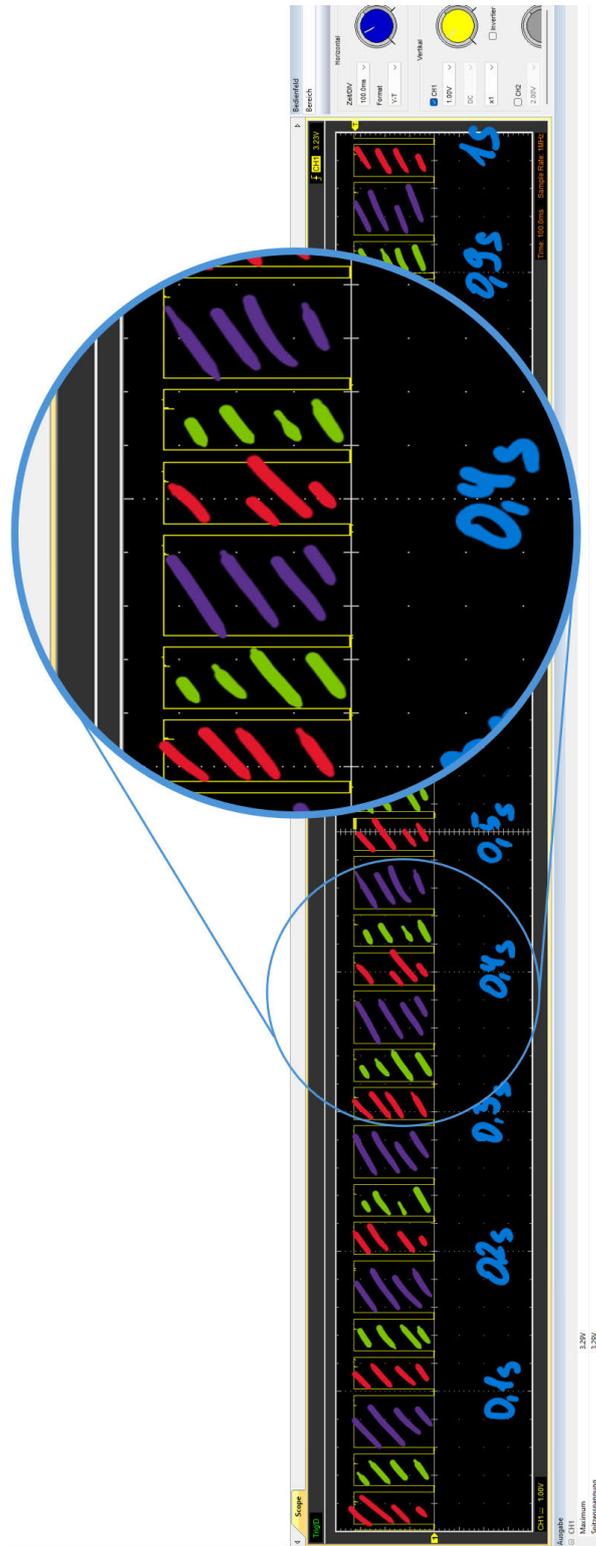


Abbildung 4.12.: Detektion von drei Magneten mit Hilfe des TLE4905L, abgelesen mit einem Oszilloskop.

## 4.6. Finales Hardwaredesign

Da ein induktiver Sensor zusätzliche Bauteile benötigt und deutlich schwerer ist als ein Hall-Sensor, wurde entschieden, sich auf Hall-Sensoren zu beschränken. Außerdem bietet der Markt keine kompakten induktiven Sensoren an, die in die Problemstellung des RC-Fahrzeugsystems integriert werden könnten. Alternativlösungen müssten aufwendig selbst entwickelt werden, was mit einem erheblichen zusätzlichen Aufwand verbunden wäre.

Um erste, einfache Tests durchzuführen, die Funktion des Aufbaus zu prüfen, sowie vorläufig die Drehzahl und die Geschwindigkeit am Rad zu erfassen, wird folgendes Konzept umgesetzt: Auf einem Breadboard werden ein ESP32 mit CAN-Bus Modul und ein OLED-Display angebracht. Der Hall-Sensor wird auf ein weiteres ESP32 Super Mini angeschlossen und mit einem CAN-Bus Modul versehen. In Kapitel 7 ist ein einfaches Verfahren mit Ständerbohrmaschine beschrieben, die variablen Drehzahlen hat und im Bohrfutter mit Magneten bestückt ist. Die von den Hall-Sensoren erfassten Signale werden anschließend im ESP32 Super Mini verarbeitet, per CAN-Bus weitergeleitet, an weitere ESP32 WROOM gesendet und auf dem OLED-Display als Drehzahl dargestellt. Folgende Bauteile werden benötigt (s. Tabelle 4.4).

Bauteil	Bezeichnung	Mögliche Herausforderung
Mikrocontroller (Sender)	ESP32 Super Mini	Stabile Stromversorgung
Mikrocontroller (Empfänger)	ESP32 WROOM	Stabile Stromversorgung
CAN-Bus Modul	unklar	Versorgung mit 3,5V
Display	OLED SSD 1306	Pull-Down Widerstände
Hall-Sensor	TLE4905E	Fester Sitz
Trigger	Magnete auf Bohrmaschinenfutter	Fester Sitz
Leitungen	Kabel	Fester Sitz

Tabelle 4.4.: Komponentenbeschreibung für die Geschwindigkeitsermittlung.

## 5. Software-Entwicklung

In diesem Kapitel wird das Konzept der Software beschrieben. Der Fokus liegt auf der Berechnung der Drehzahl sowie der Implementierung und Übertragung der Software auf den Mikrocontroller ESP32 Super Mini. Als Entwicklungsumgebung wurde die Arduino-IDE verwendet. Sie zeichnet sich durch eine einfache Bedienbarkeit aus, erfordert keine zusätzlichen Anpassungen für den ESP32 Super Mini und ermöglicht eine schnelle Kompilierung des Programmcodes. Die Programmierung erfolgt in einer auf C/C++ basierenden Sprache. Alle dafür benötigten Bibliotheken konnten direkt über die Suchfunktion der Arduino-IDE gefunden werden. Eine manuelle Installation zusätzlicher Komponenten war nicht erforderlich.

### 5.1. Architektur und Anforderungen

Die Softwarearchitektur des Systems wurde modular aufgebaut, um eine klare Trennung der Aufgabenbereiche zu gewährleisten. Ziel ist eine effiziente, robuste und erweiterbare Verarbeitung der Sensorsignale sowie deren Weitergabe an das ECU-System im Rahmen des AUDEx-Projekts. Die Implementierung erfolgt auf einem ESP32 Super Mini, der aufgrund seiner Rechenleistung, integrierten Kommunikationsschnittstellen und geringen Baugröße für das Projekt gut geeignet ist.

Das Softwaremodul erfüllt folgende funktionale Anforderungen:

- Echtzeiterfassung digitaler Eingangssignale (z.,B. Hall-Sensor),
- Berechnung der Drehfrequenz (Hz) aus Flankenählung pro Zeiteinheit,
- Aufbereitung der Messwerte für CAN-Übertragung,
- Bereitstellung der Parameter zur Kalibrierung und Konfiguration (z.,B. Impulszahl pro Umdrehung).

Die nichtfunktionalen Anforderungen beinhalten die Echtzeittauglichkeit (feste Latenzgrenzen für Interruptbehandlung und Nachrichtenversand), die Robustheit gegenüber Signalstörungen sowie eine ressourcenschonende Nutzung von RAM und Rechenzeit.

Darüber hinaus soll die Software strukturiert und portabel gestaltet sein, sodass sie für andere Hardwareplattformen oder Sensorvarianten angepasst werden kann.

### 5.2. Datenerfassung und Verarbeitung

Die Erfassung der Signale erfolgt über digitale GPIO-Eingänge des ESP32. Ein Interrupt<sup>1</sup> wird auf die fallende Flanke des Hall-Sensors gelegt. Bei jeder erkannten Flanke wird ein globaler Zähler inkrementiert. Parallel wird in einem periodischen Zeitintervall (z. B. alle 500 ms) ein Software-Timer ausgelöst, der den aktuellen Zählerstand aufnimmt, die Drehfrequenz berechnet und den Zähler zurücksetzt.

Die Frequenz wird berechnet aus:

$$f = \frac{N}{T}$$

mit:

- $N$ : Anzahl der Impulse im Intervall,
- $T$ : Abtastzeit in Sekunden (z. B. 0.5 s).

Die berechnete Frequenz wird intern für die CAN-Kommunikation aufbereitet und optional zu Debug-Zwecken über UART ausgegeben.

Ein exemplarischer Codeausschnitt zur Initialisierung, Datenerfassung und Berechnung der Drehzahl ist im Folgenden dargestellt:

```
1 // Globale Variablen
2 volatile uint32_t pulseCount = 0; // Flanken zaehlen
3 unsigned long previousMillis = 0; // Zeitstempel
4 const unsigned long interval = 500; // Messintervall in
   Millisekunden
5
6 // ISR fuer Hall-Sensor
7 void IRAM_ATTR handleHallSensor() {
8   pulseCount++; // jede fallende Flanke = ein Puls
9 }
10
11 void setup() {
12   pinMode(HALL_SENSOR_PIN, INPUT_PULLUP);
```

---

<sup>1</sup>Ein Interrupt ist eine hardware- oder softwaregesteuerte Unterbrechung des Programmablaufs, die eine sofortige Reaktion auf externe Ereignisse wie Signalfanken ermöglicht.

```
13 attachInterrupt(digitalPinToInterrupt(HALL_SENSOR_PIN),
14               handleHallSensor, FALLING);
15 }
16 void loop() {
17     unsigned long currentMillis = millis();
18     if (currentMillis - previousMillis >= interval) {
19         previousMillis = currentMillis;
20         float revolutions = pulseCount / 2.0; // 2 Flanken = 1 Umdrehung
21         float ups = revolutions / (interval / 1000.0); // Umdrehungen
                pro Sekunde
22         pulseCount = 0;
23         // Weitere Verarbeitung oder Uebertragung
24     }
25 }
```

Der Interrupt wird mit der Funktion `attachInterrupt()` initialisiert. Die Zählung der Pulse erfolgt flankengesteuert, wodurch eine hohe Genauigkeit auch bei niedrigen Drehzahlen gewährleistet ist. Die Umrechnung von Impulsen auf Umdrehungen basiert auf der Annahme, dass zwei Flanken (eine pro Magnet) einer vollständigen Radumdrehung entsprechen.

Optional kann ein gleitender Mittelwert über mehrere Intervalle implementiert werden, um die Messwerte zu glätten und zufällige Schwankungen durch magnetisches Rauschen oder mechanisches Spiel zu reduzieren.

### 5.3. Signalverarbeitung

Die erfasste Frequenz entspricht der Drehzahl des Mitnehmers. Diese wird direkt über die CAN-Schnittstelle übertragen, ohne weitere Umrechnungen auf die Raddrehzahl oder Fahrzeuggeschwindigkeit.

Die Berechnung der Raddrehzahl sowie der resultierenden Geschwindigkeit erfolgt später im Steuerungsmodul des Fahrzeugs, welches spezifische Parameter wie die Übersetzung (z.B. 43:10) und den Raddurchmesser (z.B. 0.1 m) berücksichtigt.

Zur Umrechnung in der Zielapplikation kann folgendes Verfahren verwendet werden:

$$f_{\text{rad}} = \frac{10}{43} \cdot f_{\text{mitnehmer}}$$

$$v = f_{\text{rad}} \cdot \pi \cdot d$$

mit:

- $f_{\text{rad}}$ : Raddrehzahl in Hz,
- $f_{\text{mitnehmer}}$ : gemessene Mitnehmerdrehzahl in Hz,
- $d$ : Raddurchmesser in Metern,
- $v$ : Fahrzeuggeschwindigkeit in m/s.

Die Umrechnung in km/h erfolgt durch Multiplikation mit dem Faktor 3,6. Die Entscheidung, diese Berechnungen nicht direkt auf dem ESP32 durchzuführen, erlaubt eine flexiblere Anwendung für verschiedene Fahrzeugtypen mit abweichenden Übersetzungen und Radkonfigurationen.

### 5.4. Kommunikationsschnittstellen

Die Übertragung der erfassten Mitnehmerdrehzahl erfolgt über die CAN-Schnittstelle mithilfe eines Transceivermoduls, das direkt mit dem ESP32 Super Mini verbunden ist. Die Kommunikation wird mit dem TWAI-Treiber realisiert. Die Konfiguration des CAN-Moduls erfolgt mit einer Baudrate von 500 kbit/s. Der folgende Auszug zeigt die Initialisierung:

```
1 // Konfiguration der TWAI-Schnittstelle am ECU-Modul (Sender)
2 twai_general_config_t g_config =
3     TWAI_GENERAL_CONFIG_DEFAULT(CAN_TX_PIN, CAN_RX_PIN, TWAI_MODE_NORMAL);
4
5 twai_timing_config_t t_config =
6     TWAI_TIMING_CONFIG_500KBITS(); // 500 kbit/s CAN-Bus Geschwindigkeit
7
8 // Treiberinstallation und Start
9 twai_driver_install(&g_config, &t_config, &f_config);
10 twai_start();
```

Die CAN-Nachricht verwendet das Format:

[ID: 0x111, DATA: 4 Byte]

und überträgt die aktuelle Mitnehmerdrehzahl. Der Frequenzwert (in Hz) wird mit 100 multipliziert, um zwei Nachkommastellen abzubilden, und anschließend als 32-Bit-Wert im Big-Endian-Format übertragen.

Beispiel: Eine gemessene Drehzahl von 10,00 Hz wird zu

$$10,00 \cdot 100 = 1000 = 0x000003E8$$

Der dezimale Wert 1000 wird in vier Bytes aufgeteilt:

00 00 03 E8

wobei das höchstwertige Byte zuerst übertragen wird.

Der relevante Ausschnitt aus dem Programmcode lautet:

```
1 // Mitnehmerdrehzahl (ups) wird mit 100 multipliziert, um zwei
  Nachkommastellen zu erhalten
2 uint32_t ups_int = static_cast<uint32_t>(ups * 100);
3
4 // Aufteilung in 4 Bytes (Big-Endian-Reihenfolge)
5 message.data[0] = (ups_int >> 24) & 0xFF; // hoechstwertiges Byte
6 message.data[1] = (ups_int >> 16) & 0xFF;
7 message.data[2] = (ups_int >> 8) & 0xFF;
8 message.data[3] = ups_int & 0xFF; // niederwertigstes Byte
```

Die CAN-Nachricht wird in einem festen Intervall (500 ms) gesendet, synchron zur Erfassung und Berechnung der Drehzahl. Dabei wird die erfolgreiche Übertragung über Status-LEDs angezeigt:

```
1 if (twai_transmit(&message, pdMS_TO_TICKS(1000)) == ESP_OK) {
2   digitalWrite(GREEN_LED_PIN, HIGH); // gruene LED: erfolgreich
3   digitalWrite(REDF_LED_PIN, LOW);
4 } else {
5   digitalWrite(GREEN_LED_PIN, LOW);
6   digitalWrite(REDF_LED_PIN, HIGH); // rote LED: Fehler beim Senden
7 }
```

Zur Fehlerdiagnose wird die Status-LED-Anzeige genutzt: Eine grüne LED signalisiert erfolgreiche Übertragung, eine rote LED weist auf einen Sendefehler hin. Optional können Debug-Werte über die serielle UART-Schnittstelle ausgegeben werden.

Die übertragene Mitnehmerdrehzahl wird fahrzeugseitig verwendet, um in der Steuerung die aktuelle Geschwindigkeit zu berechnen. Dies ermöglicht eine flexible Anpassung an unterschiedliche Rad- und Übersetzungskonfigurationen im AUDEX-Projekt.

### 5.5. Logischer Aufbau

Der logische Aufbau des Systems basiert auf einem Mikrocontroller, der die zentrale Verarbeitungseinheit für die Drehzahlerfassung und die CAN-Kommunikation darstellt. Die we-

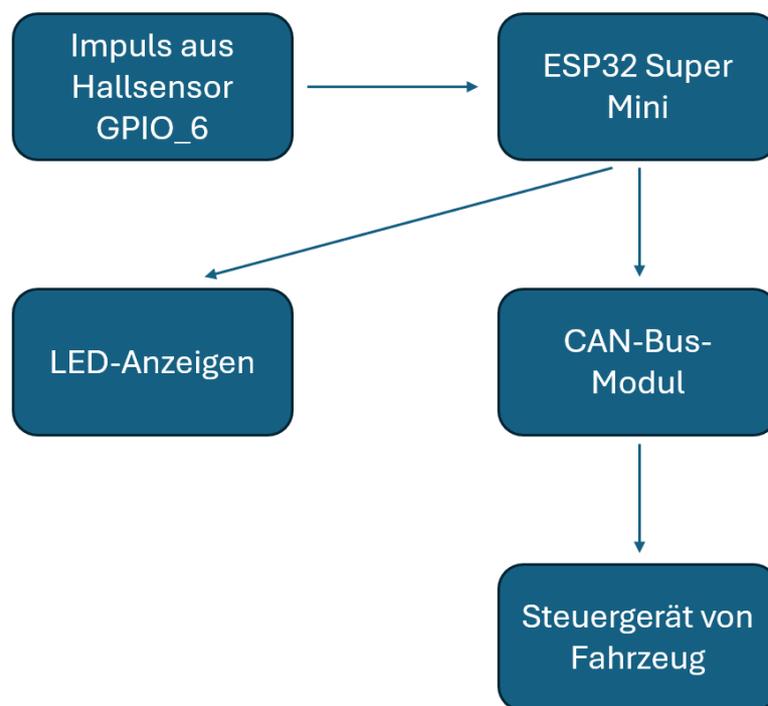


Abbildung 5.1.: Vereinfachter logischer Aufbau von Code im ECU.

sentlichen Funktionseinheiten des Systems sind über definierte GPIO-Pins mit der Hardware verbunden.

Der Hall-Sensor ist an einem digitalen Eingang (GPIO 6) angeschlossen und liefert ein Puls-Signal, sobald ein vorbeilaufender Magnet detektiert wird. Das Signal des Sensors wird ausgewertet. Die Impulsanzahl wird in einer globalen Zählervariable zwischengespeichert.

Die Kommunikation erfolgt über den Transceiver, der über zwei dedizierte Pins (GPIO 20/21) an den CAN-Bus angebunden ist.

Begleitend zur Kommunikation wird eine einfache Statusanzeige über LEDs umgesetzt: Eine grüne LED (GPIO 7) signalisiert erfolgreiche Übertragung, während eine rote LED (GPIO 9) im Fehlerfall aktiviert wird. Zusätzlich dient eine blaue LED (GPIO 1) als optischer Indikator, ob aktuell ein Magnet vom Hall-Sensor erkannt wird (s. Abbildung 5.1).

Der logische Ablauf folgt somit einem zyklischen Schema:

1. Zählen der Impulse (asynchron),
2. Zeitgesteuerte Auswertung der Pulse in der Hauptschleife,
3. Berechnung der Umdrehungsfrequenz,

4. Aufbereitung der Daten und Übertragung über CAN,
5. Rückmeldung über Status-LEDs.

Das System ist modular aufgebaut und kann auf andere Fahrzeugplattformen übertragen werden, indem nur der Sensoraufbau oder die Berechnungslogik auf Empfängerseite angepasst wird. Die gesamte Drehzahlverarbeitung bleibt lokal auf dem Steuergerät. Somit ist der Aufbau sowohl hardwareseitig robust als auch softwareseitig erweiterbar.

*[Siehe Anhang A für Details zum Code in ECU.]*

### 5.6. Grenzen der Messgenauigkeit bei niedrigen Geschwindigkeiten

Bei der gewählten Konfiguration mit zwei Magneten am Mitnehmer und einer Flankenerkennung pro Magnet entstehen pro Umdrehung zwei Impulse. Der aktuelle Aufbau sieht eine Erfassung der Flanken im 500 ms-Intervall vor. Dies entspricht einer Abtastrate von 2 Hz.

Zur Einschätzung der Messgenauigkeit bei langsamer Fahrt wird exemplarisch eine Fahrzeuggeschwindigkeit von 0.1 m/s betrachtet. Der Umfang des Rads mit einem Durchmesser von 0.1 m beträgt:

$$U = \pi \cdot d = \pi \cdot 0.1 \text{ m} \approx 0.314 \text{ m}$$

Die Raddrehzahl ergibt sich zu:

$$f_{\text{rad}} = \frac{v}{U} = \frac{0.1 \text{ m/s}}{0.314 \text{ m}} \approx 0.318 \text{ Hz}$$

Das Übersetzungsverhältnis zwischen Rad und Mitnehmer beträgt:

$$\frac{43}{10} = 4.3$$

Damit ergibt sich die Mitnehmerdrehzahl zu:

$$f_{\text{mitnehmer}} = f_{\text{rad}} \cdot \frac{43}{10} = 0.318 \text{ Hz} \cdot 4.3 \approx 1.37 \text{ Hz}$$

Da pro Umdrehung zwei Impulse erkannt werden, ergibt sich eine Impulsfrequenz von:

$$f_{\text{pulse}} = f_{\text{mitnehmer}} \cdot 2 = 1.37 \text{ Hz} \cdot 2 \approx 2.74 \text{ Hz}$$

Innerhalb eines 500 ms-Messintervalls ergibt das:

$$N = f_{\text{pulse}} \cdot T = 2.74 \text{ Hz} \cdot 0.5 \text{ s} \approx 1.37 \text{ Impulse}$$

Diese geringe Anzahl an Pulsen bedeutet, dass bereits die Abweichung von nur einem Impuls zu einer deutlich verfälschten Berechnung der Frequenz führen kann. Entsprechend ist die Drehzahlerfassung bei sehr niedrigen Geschwindigkeiten in diesem Aufbau als ungenau zu bewerten. Besonders im Bereich unter 0.1 m/s, beispielsweise bei 0.05 m/s, kann keine zuverlässige Messung mehr gewährleistet werden.

Eine effektive und hardwareseitig einfache Möglichkeit zur Verbesserung der Impulsauflösung besteht darin, nicht nur eine Flanke (z. B. die fallende), sondern beide Signalflanken (steigend und fallend) auszuwerten. Dies kann durch die Verwendung des Interrupt-Modus CHANGE anstelle von FALLING im Programmcode erfolgen. Dadurch wird bei jeder Änderung des Eingangssignals ein Interrupt ausgelöst:

```
1 // Erfassung beider Flanken
2 attachInterrupt(digitalPinToInterrupt(HALL_SENSOR_PIN),
  handleHallSensor, CHANGE);
```

Da pro Magnet dann zwei Flanken gezählt werden, verdoppelt sich die Anzahl der erfassten Impulse pro Umdrehung. Bei zwei Magneten entstehen somit insgesamt vier Impulse pro Umdrehung. Dies verbessert die Genauigkeit der Messung erheblich, insbesondere bei niedrigen Geschwindigkeiten, ohne dass zusätzliche Sensorik erforderlich ist.

Alternativ wären weitere Optimierungen denkbar, etwa:

- Erhöhung der Anzahl der Magnete am Mitnehmer,
- Verlängerung des Messintervalls,
- Softwareseitige Glättung über gleitende Mittelwertbildung.

Diese Erweiterungen und Optimierungen sind jedoch nicht Gegenstand der vorliegenden Arbeit, könnten aber im Rahmen einer zukünftigen Weiterentwicklung umgesetzt werden.

## 6. Implementierung und Inbetriebnahme des Systems

In diesem Abschnitt wird das eigentliche Produkt zur Messung der Geschwindigkeit vorgestellt. Die praktische Lösung wird erarbeitet und auf Chassis des Fahrzeugs montiert. Außerdem wird eine adressenbasierte Kommunikation auf CAN-Bus gezeigt, was eine einfache Anbindung an ein beliebiges Fahrzeug in AUDEx mit Hilfe von ECUs bietet [17].

### 6.1. Aufbau des Prototyps

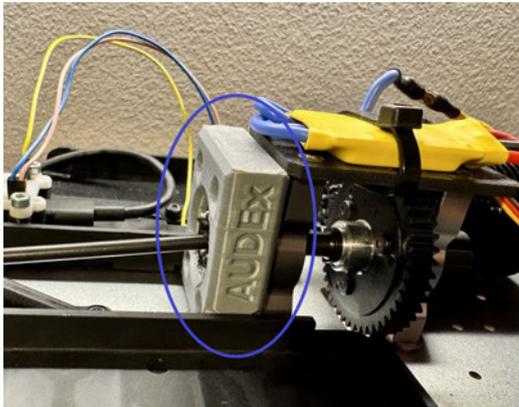
Um einen ersten Prototypen aufzubauen, werden Hall-Sensoren vom Typ TLE4905L verwendet. Einer dieser Sensoren wird direkt auf der Antriebswelle nach dem Getriebe platziert, ein weiterer Sensor wird probeweise an der Radnabe platziert [4].

#### 6.1.1. Mechanischer Aufbau

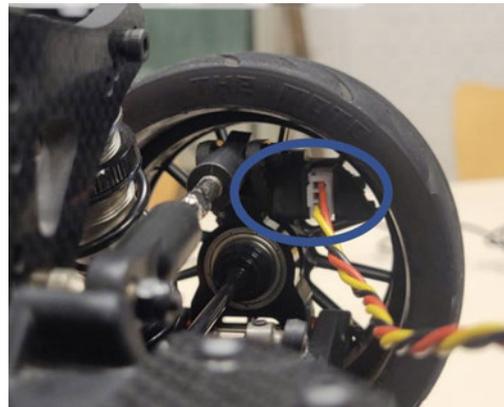
Die Sensoren werden zunächst flexibel montiert, um theoretisches Vorwissen durch praxisnahe Experimente zu validieren und zu festigen. Dabei werden verschiedene Montageplätze (s. Abbildung 6.1) für die Sensoren untersucht, die in Frage kommen:

- *An den Radnaben:* Ermöglicht die direkte Erfassung der Raddrehzahl. Es werden mehrere Sensoren benötigt, um in Kurven die Geschwindigkeitsdifferenzen plausibel zu ermitteln. Höheres Gewicht durch längere Kabel und Anzahl der Sensoren. Hohes Radgewicht durch Magnete führt zu hohen Trägheitsmomenten. Die Programmierung gestaltet sich aufwändig.
- *An der Antriebswelle:* Überzeugt durch einfache Montage der Magneten sowie relativ kleines Gewicht, bei dem Umbau des Fahrzeugs von Front auf Heck oder Allradantrieb, muss Sensor nicht verschoben und das Programm nicht umgeschrieben werden.

Für die Erfassung der Drehzahl der Antriebswelle werden zwei Neodym-Magnete mit dem Südpol nach außen auf dem Mitnehmer befestigt. Der Mitnehmer verfügt praktischerweise über zwei flach gefräste Stellen, die sich sehr gut für die Montage eignen. Die Magnete haben eine Größe von 10 mm × 3 mm und eine Haftkraft von 17.7 N (bzw. 1.8 kg [22]).



(a) Montagestelle an dem Mitnehmer



(b) Montagestelle an der Radfelge [15]

Abbildung 6.1.: Zwei verschiedene Montagestellen im Vergleich.

Ein 3D-gedrucktes Gehäuse dient als Halterung für den Hall-Sensor. Das Gehäuse wird passgenau konstruiert und kann bei Bedarf mit Kabelbindern befestigt werden; hierfür sind vier entsprechende Öffnungen vorgesehen.

Ein weiterer Vorteil ergibt sich durch die Montage auf dem Mitnehmer mit einer Übersetzung von 10:43 [5]. Durch dieses Übersetzungsverhältnis dreht sich der Mitnehmer bei einer Umdrehung des Rades etwa viermal. Dadurch kann auf die Anbringung von sechs Magneten direkt am Rad verzichtet werden und bei einer Erfassung an allen vier Rädern sogar auf insgesamt 24 zusätzliche Magnete.

### 6.1.2. Elektrischer Aufbau

Wie in Kapitel 4.6 beschrieben, wird ein Hall-Sensor von ESP32 Super Mini abgelesen. Die ausgewerteten Daten werden per CAN-Bus weiter versendet. Somit bildet das CAN-Bus Modul mit ESP32 ein vollständiges ECU, das lediglich eine zuverlässige Stromquelle benötigt. Um Fehlerquellen zu minimieren, werden Verbindungen zwischen dem CAN-Bus Modul und ESP32 verlötet, dazu kommen zwei LEDs, die den Status der CAN-Verbindung anzeigen. Ein DC-DC Wandler, oder ein Spannungsteiler L4940V5 kann das Problem mit der Stromversorgung lösen. Dieser ist zwar nicht optimal, befindet sich jedoch schon am Chassis [4]. Bei dem späteren Validieren wird daher eine Powerbank angewendet.

Eine der Schwierigkeiten beim Erstellen einer ECU besteht in der Tatsache, dass die gängigen CAN-Bus Module auf dem Markt für eine 5V-Betriebsspannung ausgelegt sind. Dies kann jedoch die GPIO-Ein- und Ausgänge des ESP32 beschädigen, da diese nur 3,3 V vertragen.

Daher wird ein spezieller CAN-Bus-Transceiver benötigt, der auch mit 3,3 V Pegeln kommunizieren kann. Ein geeigneter Kandidat dafür ist der VP230. Dieser ist als fertiges, kompakt aufgebautes Modul mit der Bezeichnung SN65HVD230 erhältlich.

Der SN65HVD230 ist kompatibel mit den gängigen CAN-Bus-Bibliotheken für den ESP32 und benötigt keine zusätzlichen Treiber. Die Modulplatine ist kompakt und eignet sich gut für den direkten Einsatz in ECU.

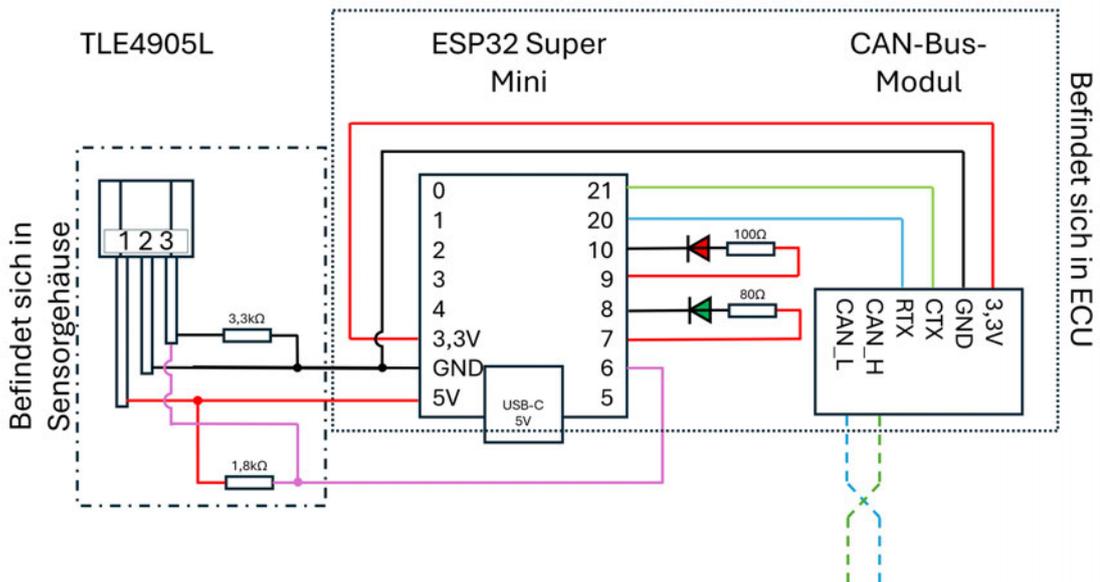


Abbildung 6.2.: Anschlusszeichnung der ECU.

Wie in Abbildung 6.2 zu sehen ist, benötigt der TLE4905L einen Spannungsteiler. Da die minimale Betriebsspannung des TLE4905L bei 3,8 V liegt, wird er über die 5 V-Bordspannung des ESP32 versorgt. Dies würde jedoch dazu führen, dass GPIO\_6 beschädigt werden könnte, da die ESP32-GPIOs nur 3,3 V tolerant sind. Die Widerstände mit 1,8 kΩ und 3,3 kΩ teilen die Ausgangsspannung des Sensors auf etwa 3,2 V, wodurch GPIO\_6 zuverlässig geschützt wird.

$$U_{\text{GPIO}} = U_{\text{in}} \cdot \frac{R_2}{R_1 + R_2} = 5 \text{ V} \cdot \frac{3,3 \text{ k}\Omega}{1,8 \text{ k}\Omega + 3,3 \text{ k}\Omega} \approx 3,2 \text{ V}$$

Das ESP32 Super Mini sowie das CAN-Bus-Modul werden in einem eigenen Gehäuse untergebracht (s. Abbildung 6.3). Dies ermöglicht einen modularen Austausch im Falle von Beschädigungen und bietet ein flexibles Packaging, es handelt sich hierbei um eine eigenständige ECU.

[Siehe Anhang D für Details zur CAN-Bus-Modul mit Stiftleisten-Verbindung]

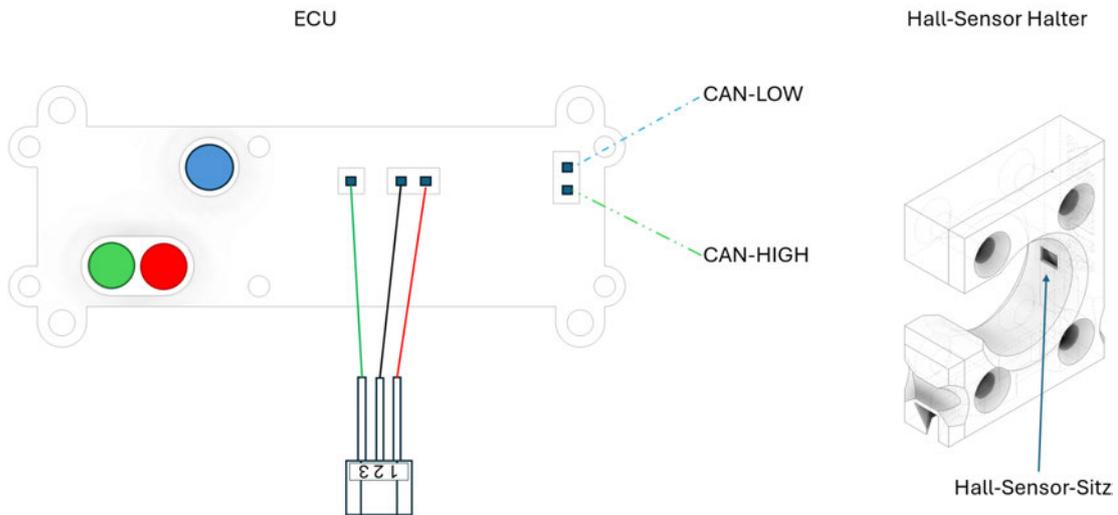


Abbildung 6.3.: 3D-gedruckte Teile.

Zwei LEDs dienen der Diagnose von Kommunikationsfehlern. Sie werden physisch über je einen Vorwiderstand direkt mit den entsprechenden GPIO-Ein- bzw. -Ausgängen verbunden. Um unnötig komplexe Verbindungen zu vermeiden, werden GPIO\_10 und GPIO\_8 softwareseitig als Masse (Low-Pegel) konfiguriert.

Die Stromversorgung erfolgt vorerst über den USB-C-Anschluss mit 5 V.

Der Sensor TLE4905L befindet sich zusammen mit den Spannungsteiler-Widerständen in einem separaten Gehäuse.

Eine weitere Lösung für das Problem mit zu hohen Pegeln am GPIO ist die Verwendung von Transistoren wie dem 2N2222 [8] (s. Abbildung 6.4) oder eines Pegelwandlers wie dem TXS0108E<sup>1</sup> zur Pegelsenkung. Mit Hilfe eines Transistors kann eine korrekte Ausgangsspannung von 3.3 V realisiert werden.

Die Anwendung eines Transistors stellt eine sichere Lösung dar, da dieser die 5 V vollständig von GPIO\_6 trennt. Da die Verschaltung nicht selbsterklärend ist, wurde zur besseren Nachvollziehbarkeit eine interaktive Schaltung in [10] erstellt.

Die Schaltung erfordert ein gewisses Verständnis von Open-Collector-Ausgängen und NPN-Transistoren. Abbildung 6.5 veranschaulicht den Aufbau. Zudem kann die interaktive Version direkt im Browser unter folgendem Link aufgerufen werden:

<https://tinyurl.com/2c2xtnsy>

<sup>1</sup>Der TXS0108E ist ein bidirektionaler Logikpegelwandler, der Spannungen zwischen 1,2 V und 5 V automatisch anpassen kann. Ideal zur Verbindung von 3,3 V- und 5 V-Systemen.

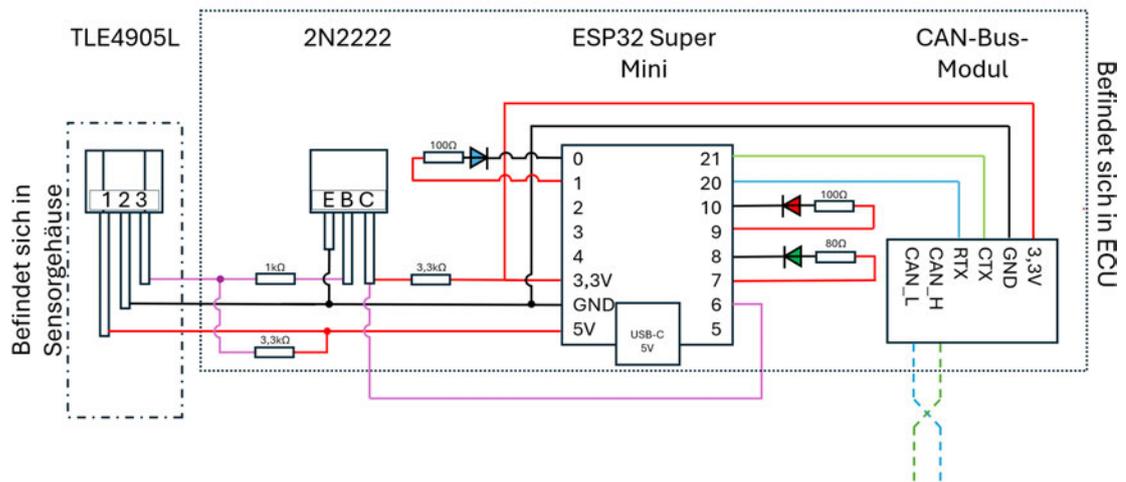


Abbildung 6.4.: Anschlusszeichnung der ECU mit Transistor 2N2222.

## 6.2. Integration in das bestehende Fahrzeug

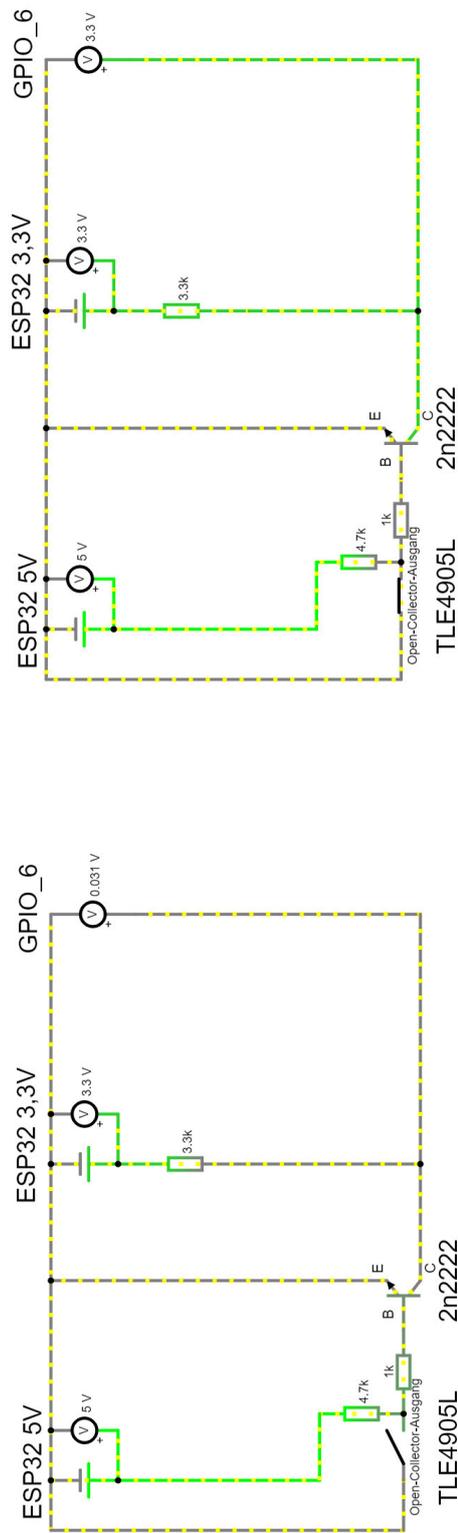
Um die adressbasierte CAN-Bus-Kommunikation zwischen dem entwickelten ECU und der Fahrzeugsteuerung zu realisieren, muss ein geeignetes Kommunikationsprotokoll definiert werden. Eine einfache Möglichkeit besteht darin, eine beliebige CAN-ID zu vergeben, beispielsweise 0x111, und die vom ESP32 Super Mini berechnete Geschwindigkeit in km/h in die zur Verfügung stehenden acht Datenbytes eines CAN-Frames einzubetten.

Ein Beispiel für ein solches CAN-Frame wäre:

0x111 00 00 80 FF 35 15 00 00

Die Aufschlüsselung der Bytes erfolgt wie folgt:

- **Byte 0:** 00 – Reserve
- **Byte 1:** 00 – Reserve
- **Byte 2:** 80 – ganzzahliger Teil der Geschwindigkeit (80 km/h)
- **Byte 3:** FF – Trennbyte (z. B. als Dezimalpunkt-Markierung)
- **Byte 4:** 35 – erste Nachkommastelle (0,35 km/h)
- **Byte 5:** 15 – weitere Nachkommastellen (0,0015 km/h)



(b) Magnet erkannt – Open-Collector-Ausgang zieht nach Masse, GPIO\_6 wird auf 3.3 V gezogen [10].

(a) Magnet nicht erkannt – Open-Collector-Ausgang ist hochohmig, GPIO\_6 liegt bei ca. 0 V [10].

Abbildung 6.5.: Zustände des Open-Collector-Ausgangs bei nicht erkanntem (links) und erkanntem Magneten (rechts).

- **Byte 6:** 00 – Platzhalter für zukünftige Erweiterungen (z. B. Zähler)
- **Byte 7:** 00 – weiterer Platzhalter (z. B. Prüfsumme)

Die Werte werden im Fahrzeug durch den ESP32 WROOM empfangen, zusammengesetzt und anschließend an die Benutzeroberfläche weitergeleitet. Daraus ergibt sich eine Gesamtgeschwindigkeit von:

$$80 \text{ km/h} + 0.35 \text{ km/h} + 0.0015 \text{ km/h} = 80.3515 \text{ km/h}$$

Ein weiteres Beispiel ist das Senden einer niedrigen Geschwindigkeit von 0.3614 km/h, wie sie etwa beim Einparken vorkommen kann. Diese würde im CAN-Frame wie folgt kodiert:

0x111 00 00 00 FF 36 14 00 00

Um eine klare Struktur in der CAN-Kommunikation zu gewährleisten und gleichzeitig die Anzahl benötigter Datenbytes gering zu halten, empfiehlt es sich, die Geschwindigkeitsinformation auf zwei Bytes zu beschränken. Zu diesem Zweck wurde ein bewährtes Verfahren aus einem BMW-Fahrzeug übernommen [17].

Die Geschwindigkeit – beispielsweise 40.31 km/h – wird dabei in zwei Bytes aufgeteilt. Diese entsprechen einem sogenannten Haupt- und einem Sekundärbyte. Die Nachricht wird über die CAN-ID 0x1A1 übertragen.

Das Prinzip beruht auf folgender Codierung:

- Die Geschwindigkeit wird zunächst mit 100 multipliziert:

$$40.31 \times 100 = 4031$$

- Dieser Wert wird anschließend durch 256 (entspricht von 0x00 bis 0xFF) geteilt:

$$\frac{4031}{256} = 15.74609$$

- Der ganzzahlige Anteil (15) wird als Hauptbyte verwendet.
- Der Nachkommaanteil (0.74609) wird mit 256 multipliziert, um den Sekundärbyte zu erhalten:

$$0.74609 \times 256 \approx 191$$

Somit ergibt sich die CAN-Nachricht in hexadezimaler Darstellung zu:

0x1A1 0F BF

Auf Seiten des Steuergeräts (ECU) erfolgt die Interpretation der empfangenen Werte durch Rückrechnung. Die beiden Bytes werden zunächst in Dezimalwerte umgewandelt:

- 0F = 15
- BF = 191

Die Geschwindigkeitsberechnung erfolgt dann gemäß folgender Formel:

$$v = \frac{191 + (15 \times 256)}{100} = \frac{191 + 3840}{100} = 40.31 \text{ km/h}$$

Diese Methode ermöglicht eine kompakte und dennoch präzise Darstellung von Geschwindigkeitswerten im CAN-Netzwerk.

Im Laufe der Arbeit an der Implementierung wurde erkannt, dass es nicht sinnvoll ist, die Geschwindigkeit direkt über das CAN-Bus-System zu übertragen. Der Grund dafür ist, dass das ECU-Modul in verschiedenen Fahrzeugen im Rahmen des AUDEx-Projekts eingesetzt wird. Dies bedeutet, dass die Räder unterschiedliche Durchmesser aufweisen, was eine eindeutige Geschwindigkeitsangabe erschwert.

Stattdessen wurde entschieden, die Drehzahl in Hertz über CAN an das Steuerungsmodul zu senden. Die Umrechnung in die tatsächliche Fahrzeuggeschwindigkeit erfolgt dann fahrzeugspezifisch – basierend auf dem Raddurchmesser und der empfangenen Drehzahl.

Es wurde sich somit für die Ableitung aus einfacheren Varianten der CAN-Nachricht Übertragung entschieden.

Bei einer Drehzahl von 10/s ergibt sich folgende Darstellung auf dem CAN-Bus:

- CAN-ID: 0x111
- Daten: 00 00 03 E8

Die Drehzahl wird mit 100 multipliziert, sodass 10/s als 1000 übertragen wird. Dieser Wert wird in 4 Bytes als Hexadezimalzahl kodiert (00 00 03 E8) und unter der ID 0x111 versendet.

Die empfangene Drehzahl kann anschließend mithilfe einer beliebigen Formel sowie unter Berücksichtigung der jeweiligen Übersetzung und des Raddurchmessers in eine Geschwindigkeit umgerechnet werden.

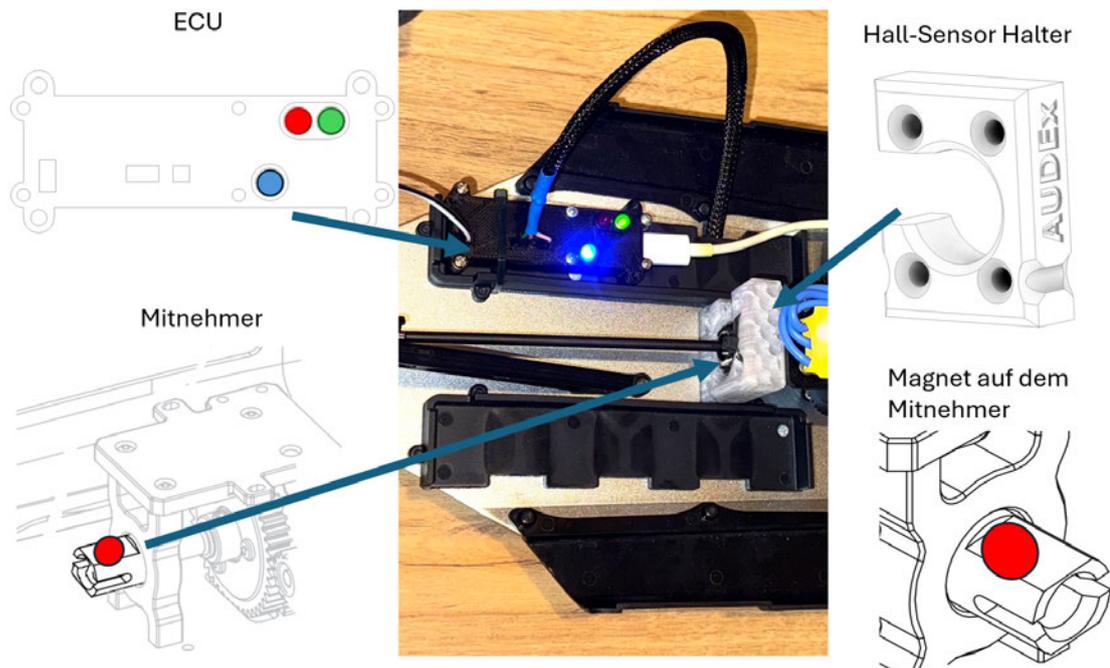


Abbildung 6.6.: Fahrgestell von Absima 1:8 Team C GR8LE mit ECU und Hall-Sensor (unten rechts ist die Magnetplatzierung) [5].

### 6.2.1. Montage im Fahrzeug

Die im Rahmen dieser Arbeit entwickelte ECU, die ESP32 Super Mini, CAN-Bus-Modul sowie der Transistorschaltung wurde in ein separates, kompaktes Gehäuse integriert, um eine modulare und servicefreundliche Anbringung im Fahrzeug zu ermöglichen. Diese Konstruktion erlaubt einen schnellen Austausch im Fall von Beschädigungen und erleichtert den Einbau in verschiedene Fahrzeugplattformen innerhalb des AUDEX-Projekts (s. Abbildung 6.6).

Die Leitungsverlegung wurde so gestaltet, dass keine beweglichen Teile beeinträchtigt werden. Der Hall-Sensor wurde mit speziell angefertigten 3D-gedruckten Halter in unmittelbarer Nähe zur Mitnehmer montiert.

Die Stromversorgung erfolgt über einen USB-C-Port mit 5 V Versorgungsspannung, was auch den Betrieb der integrierten LEDs zur Diagnose (s. Tabelle 6.1) der CAN-Kommunikation gewährleistet.

Die flexible Bauweise der ECU erlaubt es, das System bei Bedarf unkompliziert an anderen Modellfahrzeugen des Projekts zu adaptieren. Zusätzliche Schutzmaßnahmen wie hitzebeständige Ummantelungen oder vibrationsdämpfende Elemente können je nach Einsatzszenario ergänzt werden.

Blaue LED	Magnet detektiert
Rote LED	CAN-Bus Leitung unterbrochen
Grüne LED	CAN-Bus Leitung identifiziert

Tabelle 6.1.: Anzeigen zur Diagnose.

The image shows a Visual Studio Code editor window with a C++ project. The code in the editor defines a function `recieveCanDubovyy()` that receives CAN bus messages and calculates motor revolutions per second (RPM) and speed in km/h. The terminal window below the code displays the following output:

```

Motordrehzahl:3.00 U/s  Raddrehzahl:0.70 U/s  Geschwindigkeit:0.79 km/h
Motordrehzahl:2.00 U/s  Raddrehzahl:0.47 U/s  Geschwindigkeit:0.53 km/h
Motordrehzahl:3.00 U/s  Raddrehzahl:0.70 U/s  Geschwindigkeit:0.79 km/h
Motordrehzahl:0.00 U/s  Raddrehzahl:0.00 U/s  Geschwindigkeit:0.00 km/h
Motordrehzahl:3.00 U/s  Raddrehzahl:0.70 U/s  Geschwindigkeit:0.79 km/h
Motordrehzahl:4.00 U/s  Raddrehzahl:0.93 U/s  Geschwindigkeit:1.05 km/h
Motordrehzahl:5.00 U/s  Raddrehzahl:1.16 U/s  Geschwindigkeit:1.32 km/h
Motordrehzahl:6.00 U/s  Raddrehzahl:1.40 U/s  Geschwindigkeit:1.58 km/h
    
```

Abbildung 6.7.: Drehzahlen und Geschwindigkeit in Terminal von Visual-Studio-IDE

In Abbildung 6.7 ist die Benutzeroberfläche der Visual-Studio-IDE dargestellt. Das entwickelte ECU-Modul wurde an ein AUDEx-Fahrzeug angeschlossen. Die erfasste Drehzahl wird über den CAN-Bus an ein ESP32-Modul übertragen, welches als Fahrzeugsteuerung dient. Von dort aus erfolgt eine Weiterleitung der Daten über die serielle Schnittstelle an das Terminal der IDE.

Die Anzeige der Werte im Serial Monitor dient sowohl der Fehlerdiagnose (Debugging) als auch dem Nachweis der funktionierenden Kommunikationsverbindung. Durch diese Architektur kann das entwickelte ECU-System flexibel in verschiedenen AUDEx-Fahrzeugen eingesetzt werden.

## 7. Verifikation der Anforderungen

Bevor mit der Validierung begonnen und Tests unter realen Bedingungen durchgeführt werden können, ist zunächst eine Verifikation der Anforderungen erforderlich.

Dabei wird überprüft, ob das entwickelte System mit den zuvor definierten Anforderungen übereinstimmt und die angestrebte Funktionalität erfüllt. Die Verifikation stellt sicher, dass alle Komponenten korrekt umgesetzt wurden und das System innerhalb des vorgesehenen Rahmens zuverlässig arbeitet.

Die eigentlichen Genauigkeitsnachweise und Referenzmessungen erfolgen im Rahmen der Validierung in Kapitel 8.

Der Fokus der Verifikation liegt auf der technischen Umsetzung und der korrekten Funktion des Systems. Im Abgleich mit den Anforderungen aus Kapitel 3.1 stehen dabei insbesondere folgende Punkte im Mittelpunkt:

- **Genauigkeit:** Das System liefert konsistente Messergebnisse auch unter variierenden Bedingungen und bei mehrfacher Messung.
- **Robustheit:** Die Komponenten befinden sich in einem stabilen, 3D-gedruckten Gehäuse, das für den mobilen Einsatz geeignet ist.
- **Energieeffizienz:** Das System kann mit einer Powerbank betrieben werden; sowohl der verwendete ESP32 Super Mini als auch der zugehörige Programmcode sind auf geringen Energieverbrauch ausgelegt.
- **Gewichtsoptimierung:** Die Anzahl der verwendeten Magnete wurde bewusst minimal gehalten, um unnötiges Zusatzgewicht, insbesondere an den Rädern, zu vermeiden.
- **Kompaktheit:** Die Bauweise des Systems ist sehr kompakt und ermöglicht eine flexible Integration sowie eine optimierte Gewichtsverteilung im Modellfahrzeug.
- **Flexibilität:** Das System ist sowohl für PKW- als auch für LKW-Modelle des AUDEx-Projekts einsetzbar; lediglich die Positionierung des Hall-Sensors muss an die jeweilige Geometrie angepasst werden.

## 8. Validierung

Zur Validierung des entwickelten Systems wird überprüft, inwiefern die implementierte Lösung zur Geschwindigkeitsermittlung im praktischen Einsatz die definierten Anforderungen erfüllt und in realistischen Anwendungsfällen zuverlässige Messergebnisse liefert. Ziel dieser Untersuchungen ist es, die Messergebnisse des Hall-Sensors und der entwickelten ECU mit den Werten externer Messwerkzeuge zu vergleichen. Die Ergebnisse müssen dabei übereinstimmen, um die Funktionsfähigkeit und Genauigkeit des Systems zu bestätigen.

### 8.1. Testmethodik und Messergebnisse

Zur Ermittlung der Messgenauigkeit werden im Folgenden verschiedene Methoden vorgestellt. Zum Einsatz kommt ein geeichtes Drehzahlmessgerät DT-30LK der Fa. Voltcraft mit unterschiedlichen Messmethoden, wie in Abbildung 8.1 und 8.2 vorgestellt. Ziel ist es, die Genauigkeit, Robustheit und Zuverlässigkeit des Gesamtsystems zu bewerten. Die angewandte Testmethodik umfasst (s. Tabelle 8.1):

- Drehzahlermittlung am Mitnehmer mit der selbstgebauten ECU unter realen Bedingungen auf dem Fahrgestell (Welle ECU).
- Messung am Mitnehmer mit dem Drehzahlmesser auf dem Fahrgestell (Welle Ref.).
- Drehzahlermittlung am Rad mit der selbstgebauten ECU unter realen Bedingungen auf dem Fahrgestell (Rad ECU).
- Messung am Rad mit dem Drehzahlmesser auf dem Fahrgestell (Rad Ref.).
- Vergleich und Bewertung der Messergebnisse beider Systeme (Abw. Welle).
- Identifikation möglicher Messabweichungen sowie Fehlersuche und ggf. Fehlerbehebung (Abw. Rad).

<b>Welle (Ref.)</b> Hz	<b>Welle (ECU)</b> Hz	<b>Rad (Ref.)</b> Hz	<b>Rad (ECU)</b> Hz	<b>Abw. Welle</b> %	<b>Abw. Rad</b> %
20	20,2	4,6	4,8	4,5	4,3
40	39,9	9,3	9,4	0,3	1,1
60	60,2	13,9	14,0	0,3	0,7
80	80,1	18,6	18,6	0,1	0,0
100	100,0	23,3	23,4	0,0	0,4
120	120,0	28,0	28,0	0,0	0,0
140	140,2	32,6	32,7	0,1	0,3
160	159,8	37,2	37,3	0,1	0,3
183	184,4	42,6	42,7	0,8	0,2

Tabelle 8.1.: Vergleich der Drehzahlerfassung an Welle und Rad.

Die minimale Drehzahl am Mitnehmer, die durch den Motortreiber in Kombination mit dem Getriebe ermöglicht wird, beträgt 20 Hz (entspricht 20 Umdrehungen pro Sekunde). Die maximale Drehzahl liegt bei 183 Hz.

Obwohl das Differential eine Übersetzung von 43:10 aufweist, ist es in der Praxis schwierig, die Drehzahl des Mitnehmers mit dem Motortreiber so genau einzustellen, dass sie sich durch den Faktor 4,3 exakt teilen lässt. Aus diesem Grund wurde die Geschwindigkeit mithilfe eines Test-Displays abgelesen, das zur Simulation der Fahrzeug-ECU diente.

*[Siehe Anhang B für Details zur ECU Simulator der Fahrzeugsteuerung]*

Die entstandenen Abweichungen wurden nach dieser Formel bestimmt:

$$\text{Abweichung (\%)} = \left| \frac{\text{Messwert} - \text{Referenzwert}}{\text{Referenzwert}} \right| \times 100$$

Im späteren Fahrzeugeinsatz kann der Anwender eine eigene Applikation mit dem passenden Raddurchmesser erstellen. Das Fahrgestell RG8LG-LA verwendet Räder mit einem Durchmesser von 10 cm. Dabei kann sich der Anwender am Code des Simulators der Fahrzeugsteuerung orientieren (s. Tabelle 8.2).

*[Siehe Anhang C für Details zur Code in ECU Simulator der Fahrzeugsteuerung]*

Analog zur Berechnung in Kapitel 4.1 ergeben sich bei den gegebenen Drehzahlen die folgenden Fahrzeuggeschwindigkeiten:

Um das ECU auch bei Drehzahlen unterhalb von 20/s zu testen und die Überlegungen aus Kapitel 5.6 nachzuvollziehen, wurde das Getriebe des Fahrgestells nicht mit dem eingebauten Motor betrieben, sondern durch einen Akkuschauber angetrieben, der eine sehr langsame

Mitnehmer [Hz]	Radgeschwindigkeit [km/h]
20	5,26
40	10,51
60	15,77
80	21,02
100	26,32
120	31,57
140	36,83
160	42,08
183	48,13

Tabelle 8.2.: Radgeschwindigkeit in Abhängigkeit der Mitnehmerdrehzahl (Übersetzung 10:43, Raddurchmesser 10 cm).

und kontrollierte Drehbewegung erlaubt. Der Schrauber wurde im Bereich von etwa 1/s bis 5/s eingesetzt, um zu überprüfen, ob das entwickelte Steuergerät auch bei sehr niedrigen Fahrgeschwindigkeiten zuverlässig funktioniert. Diese Untersuchungen sind insbesondere im Hinblick auf Anwendungsfälle wie automatisiertes Einparken von Bedeutung (s. Tabelle 8.3).

Welle (Ref.) Hz	Welle (ECU) Hz	Rad (Ref.) Hz	Rad (ECU) Hz	Abw. Welle %	Abw. Rad %
1	N/A	N/A	N/A	N/A	N/A
2	2.0	0.50	0.50	0.0	0.0
3	3.1	0.70	0.75	3.3	7.1
4	3.9	0.93	0.97	2.5	4.3
5	5.0	1.16	1.10	0.0	5.2

Tabelle 8.3.: Vergleich der Drehzahlerfassung an Welle und Rad (angetrieben mit Akkuschauber).

## 8.2. Diskussion der Ergebnisse

Die Messergebnisse bestätigen, dass das entwickelte System die Anforderungen an Genauigkeit und Robustheit erfüllt.

Die maximale Abweichung von 4.5 %, die bei 20 Hz am Mitnehmer gemessen wurde, entstand höchstwahrscheinlich nicht durch fehlerhafte Programmierung oder einen fehlerhaften Aufbau. Vielmehr liegt die Ursache darin, dass der verwendete Motortreiber bei so niedrigen Drehzahlen instabil arbeitet. Dieser ist ursprünglich für höhere Drehzahlen ausgelegt und stammt aus dem Drohnenbereich.

Die mit Hilfe eines Akkuschraubers als Antrieb erzeugten Messwerte zeigen, dass auch niedrigere Drehzahlen zuverlässig detektiert werden können. Die untere Grenze liegt jedoch bei mindestens  $2/s$  an der Mitnehmerwelle, was bei einem Raddurchmesser von 10 cm einer linearen Geschwindigkeit von etwa  $0.146 \text{ m/s}$  entspricht. Die bei 5 Hz an der Mitnehmerwelle gemessene Abweichung von 5.2 % ist auf die Betätigung des Auslösers am Akkuschauber zurückzuführen, der ab einer bestimmten Drehzahl unkontrolliert beschleunigt. Folgende Punkte wurden beobachtet:

### 1. Stärken

- Der Hall-Sensor zeigte eine herausragende Leistung bei stabilen und schnellen Geschwindigkeiten.

### 2. Schwächen

- Die Hall-Sensoren sind anfällig für magnetische Störungen in der Umgebung.

### 3. Verbesserungspotenziale

- Eine bessere Abschirmung der Hall-Sensoren könnte externe Störungen minimieren.
- Ein Hall-Sensor und der zugehörige Magnet (z. B. ein Ringmagnet) könnten direkt an der Motorwelle angebracht werden. Dies erhöht die Messpräzision insbesondere bei niedrigen Drehzahlen bzw. geringen Geschwindigkeiten, beispielsweise bei  $0.1 \text{ m/s}$  (entspricht etwa  $0.36 \text{ km/h}$ ).

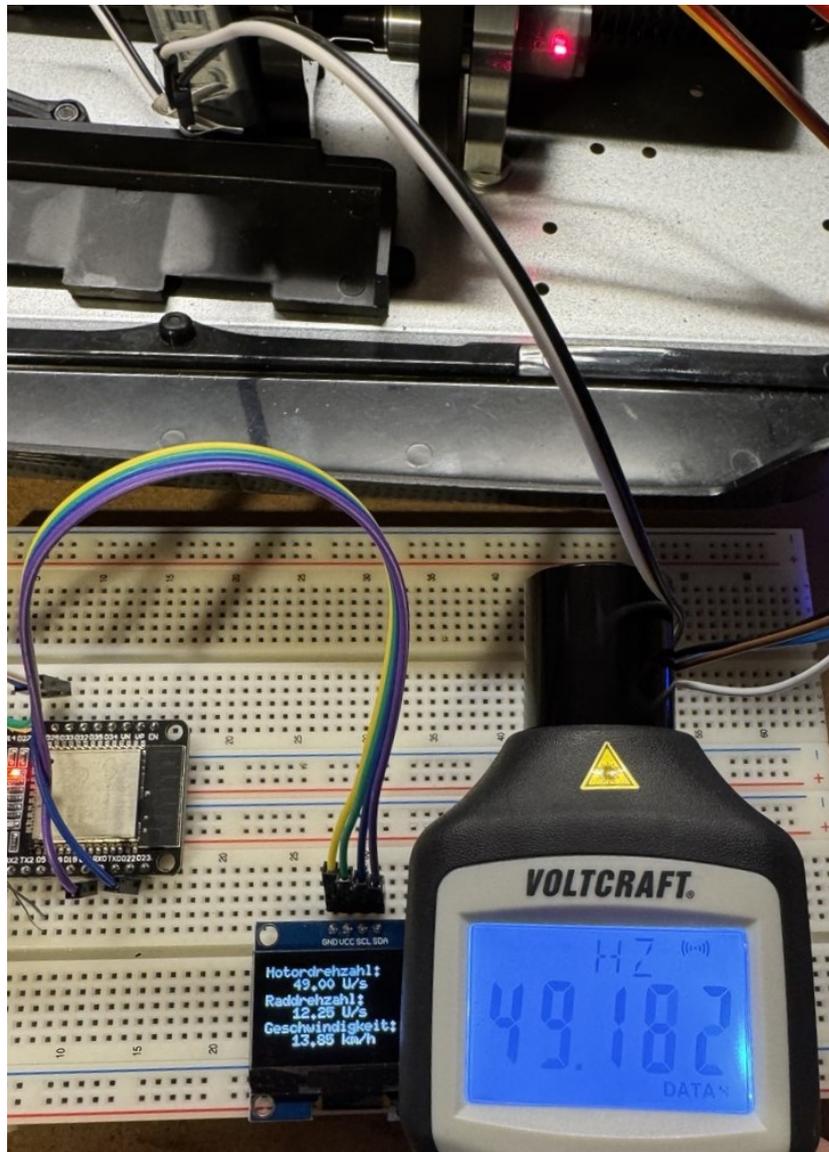


Abbildung 8.1.: Messung am Mitnehmer mit ECU und Drehzahlmesser der Fa. Voltcraft, die Werte stimmen überein.

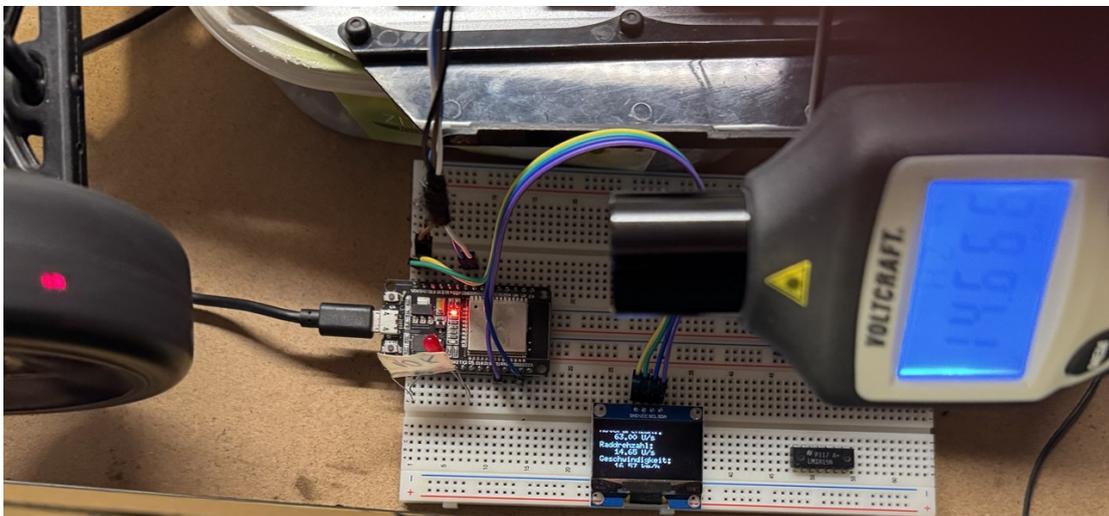


Abbildung 8.2.: Messung am Rad mit ECU und Drehzahlmesser der Fa. Voltcraft, die Werte stimmen überein.

# 9. Zusammenfassung und Ausblick

## 9.1. Zusammenfassung der Ergebnisse

Die vorliegende Arbeit beschäftigte sich mit der Entwicklung, Implementierung und Validierung eines Geschwindigkeitsmesssystems für Modellfahrzeuge. Ziel war es, eine zuverlässige, kompakte und energieeffiziente Lösung für das AUDEx-Projekt bereitzustellen. Die zentralen Ergebnisse können wie folgt zusammengefasst werden:

- **Entwicklung und Implementierung:**

- Es wurde ein modulares Sensorsystem realisiert, bestehend aus Hall-Sensor zur Erfassung der Rotationsfrequenz.
- Ein Mikrocontroller (ESP32-S3 Mini) verarbeitet die Sensordaten, berechnet die Umdrehungsfrequenz und überträgt diese über eine CAN-Schnittstelle.
- Das System wurde um Statusanzeigen (LEDs) erweitert.

- **Verifikation der Anforderungen:**

- Die Messwerte wurden im Labor und unter realen Bedingungen mit einem geeichten Referenzgerät verglichen.
- Die maximale Abweichung betrug lediglich 4.5 % bei sehr niedrigen Frequenzen.
- Die Drehzahlmessung am Mitnehmer und Rad verlief innerhalb der erwarteten Toleranzen, siehe Kapitel 7.

- **Validierung und Einsatzgrenzen:**

- Der Einsatz bei langsamen Fahrmanövern (z. B. 0.1 m/s) zeigte die prinzipielle Tauglichkeit.
- Die untere Anwendungsgrenze liegt bei etwa 2 Hz an der Mitnehmerwelle, entsprechend ca. 0.15 m/s.
- Die Genauigkeit kann durch softwareseitige Optimierungen (z. B. Nutzung beider Flanken, gleitender Mittelwert) weiter verbessert werden.

- **Vergleich mit anderen Systemen:**

- Im Vergleich zu früheren Messsystemen bot das entwickelte System Vorteile in Bezug auf Reaktionszeit und Flexibilität.
- Besonders bei Indoor-Anwendungen, Tunneldurchfahrten oder in signalarmen Umgebungen stellt es eine robuste Alternative dar.

- **Stärken und Schwächen:**

- **Stärken:** Geringer Stromverbrauch, einfache Integration, schlupffreie Erfassung.
- **Schwächen:** Empfindlichkeit gegenüber magnetischen Störfeldern, eingeschränkte Auflösung bei sehr geringen Geschwindigkeiten.

Zusammenfassend erfüllt das entwickelte Geschwindigkeitsmesssystem die im AUDEX-Projekt formulierten Anforderungen und bietet eine stabile Grundlage für zukünftige Erweiterungen und die Integration in komplexere Fahrzeugsysteme.

## 9.2. Ausblick auf zukünftige Arbeiten

Aufbauend auf den Ergebnissen dieser Arbeit ergeben sich mehrere sinnvolle Ansätze für zukünftige Weiterentwicklungen und Optimierungen des entwickelten Systems. Diese lassen sich in folgende Kategorien unterteilen:

1. **Optimierung der Hardware:**

- Entwicklung einer verbesserten elektromagnetischen Abschirmung für Hall-Sensoren zur Reduktion externer Störeinflüsse.
- Umrüstung des bisherigen Systems mit zwei Magneten auf eine Mehrmagnet-Anordnung auf der Welle. In Kombination mit einer angepassten Software würde dies die Messauflösung und Präzision deutlich erhöhen.
- Integration eines zusätzlichen Hall-Sensors zur Erkennung der Drehrichtung der Welle.
- Implementierung einer aktiven Spannungsüberwachung zur frühzeitigen Erkennung von Versorgungseinbrüchen oder Fehlfunktionen.

2. **Erweiterung der Software:**

- Entwicklung und Integration von Algorithmen zur Sensordatenfusion, beispielsweise zur Kombination von Hall-Sensor- und GPS-Daten, um eine robustere und genauere Geschwindigkeitsermittlung zu ermöglichen.

- Erweiterung um Diagnosetools zur Detektion von Sensorausfällen oder Plausibilitätsabweichungen, insbesondere bei der Verwendung mehrerer Hall-Sensoren.
- Verwendung des Interrupt-Modus CHANGE anstelle von FALLING im Programmcode verdoppelt die Genauigkeit, in Kapitel 5.6 beschrieben.
- Ausgabe von quadratischen Mittelwerten.

**3. Langzeittest und Umweltrobustheit:**

- Durchführung von Langzeittests unter realen Einsatzbedingungen zur Bewertung von Zuverlässigkeit und Langzeitstabilität.
- Erweiterte Tests unter Extrembedingungen wie hohen Temperaturen, Feuchtigkeit oder mechanischer Vibration, um die Robustheit des Systems abzusichern.

**4. Integration weiterer Technologien:**

- Kombination des entwickelten Systems mit GPS-Modulen zur globalen Positions- und Geschwindigkeitsbestimmung, wodurch neue Einsatzbereiche erschlossen werden könnten (s. Kapitel 2.2.1).

**5. Übertragung auf andere Anwendungen:**

- Skalierung des Systems auf größere Maßstäbe (z. B. 1:5 oder 1:1), um einen Einsatz auch in realen Fahrzeugen zu ermöglichen.
- Nutzung des modularen Aufbaus zur Adaption auf verschiedene Fahrzeugplattformen und andere domänenspezifische Anwendungen.

Zusammenfassend bietet diese Arbeit eine solide Grundlage für die Weiterentwicklung intelligenter Geschwindigkeitsmesssysteme. Die erzielten Ergebnisse liefern nicht nur einen direkten Mehrwert für das AUDEX-Projekt, sondern eröffnen auch Perspektiven für den Einsatz in vergleichbaren Forschungs- und Entwicklungsprojekten innerhalb der Fahrzeugtechnik.

## A. Anhang: Quellcode ECU-Modul (ESP32-S3 Mini)

```
1
2 // TWAI-Treiber fuer CAN-Kommunikation (ESP32)
3 #include <driver/twai.h>
4
5 // --- Pinbelegung fuer ESP32-S3 Mini ---
6 #define CAN_TX_PIN GPIO_NUM_21
7 #define CAN_RX_PIN GPIO_NUM_20
8 #define HALL_SENSOR_PIN GPIO_NUM_6
9
10 #define GREEN_LED_PIN GPIO_NUM_7
11 #define GREEN_LED_GND_PIN GPIO_NUM_8
12 #define RED_LED_PIN GPIO_NUM_9
13 #define RED_LED_GND_PIN GPIO_NUM_10
14
15 #define LED_PIN GPIO_NUM_1 // Blaue LED (Anzeige bei keinem Magnet)
16 #define LED_GND_PIN GPIO_NUM_0
17
18 // --- Globale Variablen fuer Drehzahlerfassung ---
19 volatile uint32_t pulseCount = 0; // Flanken zaehlen
20 unsigned long previousMillis = 0; // Zeitstempel fuer Intervall
21 const unsigned long interval = 500; // Messintervall: 500ms
22
23 // --- Interrupt-Service-Routine: wird bei jeder fallenden Flanke aufgerufen
24 // ---
25 void IRAM_ATTR handleHallSensor() {
26     pulseCount++; // eine Flanke entspricht einem Puls
27 }
28
29 void setup() {
30     Serial.begin(115200); // Serielle Ausgabe zur Analyse
31
32     // --- Pin-Konfiguration ---
33     pinMode(HALL_SENSOR_PIN, INPUT_PULLUP); // Eingang mit Pull-Up-Widerstand
```

```
33 pinMode(GREEN_LED_PIN, OUTPUT);
34 pinMode(GREEN_LED_GND_PIN, OUTPUT);
35 pinMode(RED_LED_PIN, OUTPUT);
36 pinMode(RED_LED_GND_PIN, OUTPUT);
37 pinMode(LED_PIN, OUTPUT);
38 pinMode(LED_GND_PIN, OUTPUT);
39
40 // GND-Pins der LEDs fest auf LOW (gemeinsame Kathode)
41 digitalWrite(GREEN_LED_GND_PIN, LOW);
42 digitalWrite(RED_LED_GND_PIN, LOW);
43 digitalWrite(LED_GND_PIN, LOW);
44
45 // LEDs initial ausschalten
46 digitalWrite(GREEN_LED_PIN, LOW);
47 digitalWrite(RED_LED_PIN, LOW);
48 digitalWrite(LED_PIN, LOW);
49
50 // --- Interrupt aktivieren (auf fallende Flanke) ---
51 attachInterrupt(digitalPinToInterrupt(HALL_SENSOR_PIN), handleHallSensor,
52               FALLING);
53
54 // --- CAN-Konfiguration ---
55 twai_general_config_t g_config = TWAI_GENERAL_CONFIG_DEFAULT(CAN_TX_PIN,
56                   CAN_RX_PIN, TWAI_MODE_NORMAL);
57 twai_timing_config_t t_config = TWAI_TIMING_CONFIG_500KBITS(); //
58                               Bitrate: 500 kbit/s
59 twai_filter_config_t f_config = TWAI_FILTER_CONFIG_ACCEPT_ALL(); //
60                               keine Filter
61
62 // --- CAN-Treiber installieren ---
63 if (twai_driver_install(&g_config, &t_config, &f_config) == ESP_OK) {
64     Serial.println("CAN-Treiber erfolgreich installiert.");
65 } else {
66     Serial.println("Fehler beim Installieren des CAN-Treibers.");
67     while (1); // Endlosschleife bei Fehler
68 }
69
70 // --- CAN starten ---
71 if (twai_start() == ESP_OK) {
72     Serial.println("CAN-Treiber gestartet.");
73 } else {
74     Serial.println("Fehler beim Starten des CAN-Treibers.");
75     digitalWrite(GREEN_LED_PIN, HIGH);

```

```
72     digitalWrite(RED_LED_PIN, HIGH);
73     while (1); // Endlosschleife bei Fehler
74 }
75
76 previousMillis = millis(); // Startzeit speichern
77 }
78
79 void loop() {
80     unsigned long currentMillis = millis();
81
82     // --- Blaue LED: leuchtet, wenn kein Magnet erkannt wird ---
83     if (digitalRead(HALL_SENSOR_PIN) == HIGH) {
84         digitalWrite(LED_PIN, HIGH); // Magnet nicht erkannt LED an
85     } else {
86         digitalWrite(LED_PIN, LOW); // Magnet erkannt LED aus
87     }
88
89     // --- Auswertung alle 500ms ---
90     if (currentMillis - previousMillis >= interval) {
91         previousMillis = currentMillis;
92
93         float revolutions = pulseCount / 2.0; // Zwei Flanken entsprechen einer
          Umdrehung
94         float ups = revolutions / (interval / 1000.0); // Umdrehungen pro
          Sekunde
95
96         // --- CAN-Nachricht vorbereiten ---
97         twai_message_t message;
98         message.identifier = 0x111;
99         message.extd = 0;
100        message.rtr = 0;
101        message.data_length_code = 4;
102
103        // Ups * 100 fuer 2 Nachkommastellen, als 32-bit Wert im Big-Endian-
          Format
104        uint32_t ups_int = static_cast<uint32_t>(ups * 100);
105        message.data[0] = (ups_int >> 24) & 0xFF;
106        message.data[1] = (ups_int >> 16) & 0xFF;
107        message.data[2] = (ups_int >> 8) & 0xFF;
108        message.data[3] = ups_int & 0xFF;
109
110        // --- CAN-Nachricht senden ---
111        if (twai_transmit(&message, pdMS_TO_TICKS(1000)) == ESP_OK) {
```

## A. Anhang: Quellcode ECU-Modul (ESP32-S3 Mini)

---

```
112     Serial.print("Umdrehungen pro Sekunde: ");
113     Serial.println(ups);
114     digitalWrite(GREEN_LED_PIN, HIGH); // Erfolg grüne LED
115     digitalWrite(RED_LED_PIN, LOW);
116   } else {
117     Serial.println("Fehler beim Senden der Nachricht.");
118     digitalWrite(GREEN_LED_PIN, LOW);
119     digitalWrite(RED_LED_PIN, HIGH); // Fehler rote LED
120   }
121
122   pulseCount = 0; // Zaehler zuruecksetzen fuer naechste Messung
123 }
124 }
```

## B. Anhang: ECU-Empfänger mit OLED-Display

In diesem Anhang wird der ECU-Empfänger näher erläutert (siehe Abbildung B.1). Der Aufbau basiert auf einem frei wählbaren ESP32-Modul, einem SH1106-basierten OLED-Display sowie einem SN65HVD230-CAN-Transceiver.

Die empfangenen CAN-Nachrichten mit der ID 0x111 werden interpretiert, dekodiert und auf dem Display in Echtzeit visualisiert. Dabei werden unter anderem die Mitnehmerdrehzahl, die berechnete Raddrehzahl sowie die resultierende Fahrzeuggeschwindigkeit angezeigt.

Der zugehörige Quellcode befindet sich im Anhang C.

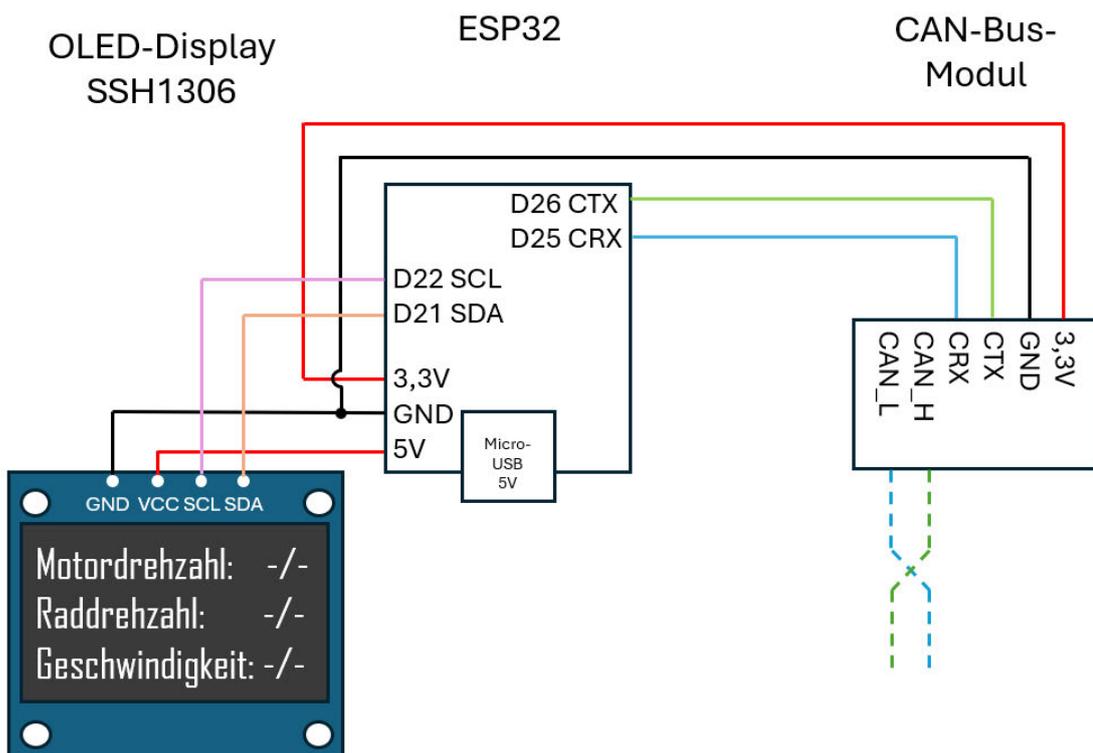


Abbildung B.1.: ECU-Empfänger mit OLED-Display.

## C. Anhang: Code für Anzeigemodul (CAN-Empfänger mit OLED)

```
1
2 // TWAI-Treiber fuer CAN-Kommunikation (ESP32)
3 #include <driver/twai.h>
4 #include <Wire.h>
5 #include <U8g2lib.h> // Bibliothek fuer OLED-Display
6
7 // Pinbelegung fuer CAN
8 #define CAN_TX_PIN GPIO_NUM_26
9 #define CAN_RX_PIN GPIO_NUM_25
10
11 // Display-Konfiguration fuer SH1106 (I2C)
12 U8G2_SH1106_128X64_NONAME_F_HW_I2C display(U8G2_R0, U8X8_PIN_NONE,
13     GPIO_NUM_22, GPIO_NUM_21);
14
15 // Fahrzeugparameter
16 #define WHEEL_DIAMETER_CM 10.0
17 #define GEAR_RATIO 4.3
18 #define PI 3.14159265358979323846
19
20 void setup() {
21     Serial.begin(115200);
22     Serial.println("Initialisiere OLED-Display...");
23
24     display.begin();
25     display.clearDisplay();
26     display.setFont(u8g2_font_6x10_tr);
27     display.drawStr(0, 10, "CAN-OLED-Display");
28     display.sendBuffer();
29     delay(1000);
30
31     twai_general_config_t g_config = TWAI_GENERAL_CONFIG_DEFAULT(CAN_TX_PIN,
32         CAN_RX_PIN, TWAI_MODE_NORMAL);
33     twai_timing_config_t t_config = TWAI_TIMING_CONFIG_500KBITS();
```

```
32 twai_filter_config_t f_config = {
33     .acceptance_code = 0x111 << 21,
34     .acceptance_mask = ~(0x7FF << 21),
35     .single_filter = true
36 };
37
38 if (twai_driver_install(&g_config, &t_config, &f_config) == ESP_OK) {
39     Serial.println("CAN-Treiber erfolgreich installiert.");
40 } else {
41     Serial.println("Fehler beim Installieren des CAN-Treibers.");
42     display.clearBuffer();
43     display.drawStr(0, 10, "Fehler: CAN-Treiber!");
44     display.sendBuffer();
45     while (1);
46 }
47
48 if (twai_start() == ESP_OK) {
49     Serial.println("CAN-Treiber gestartet.");
50 } else {
51     Serial.println("Fehler beim Starten des CAN-Treibers.");
52     display.clearBuffer();
53     display.drawStr(0, 10, "Fehler: Start CAN!");
54     display.sendBuffer();
55     while (1);
56 }
57 }
58
59 void loop() {
60     twai_message_t message;
61
62     if (twai_receive(&message, pdMS_TO_TICKS(1000)) == ESP_OK) {
63         if (message.identifier == 0x111 && message.data_length_code == 4) {
64             Serial.println("Nachricht mit ID 0x111 empfangen.");
65
66             uint32_t motor_revolutions = (message.data[0] << 24) |
67                                         (message.data[1] << 16) |
68                                         (message.data[2] << 8) |
69                                         message.data[3];
70
71             float motor_rps = motor_revolutions / 100.0;
72             float wheel_rps = motor_rps / GEAR_RATIO;
73             float circumference_cm = PI * WHEEL_DIAMETER_CM;
74             float speed_cms = wheel_rps * circumference_cm;
```

```
75     float speed_kmh = (speed_cms / 100000.0) * 3600.0;
76
77     Serial.print("Motordrehzahl: ");
78     Serial.print(motor_rps);
79     Serial.println(" U/s");
80     Serial.print("Raddrehzahl: ");
81     Serial.print(wheel_rps);
82     Serial.println(" U/s");
83     Serial.print("Geschwindigkeit: ");
84     Serial.print(speed_kmh);
85     Serial.println(" km/h");
86
87     display.clearBuffer();
88     display.setCursor(0, 10);
89     display.print("Motordrehzahl:");
90     display.setCursor(20, 19);
91     display.printf("%.2f U/s", motor_rps);
92
93     display.setCursor(0, 30);
94     display.print("Raddrehzahl:");
95     display.setCursor(20, 39);
96     display.printf("%.2f U/s", wheel_rps);
97
98     display.setCursor(0, 49);
99     display.print("Geschwindigkeit:");
100    display.setCursor(20, 58);
101    display.printf("%.2f km/h", speed_kmh);
102
103    display.sendBuffer();
104    } else {
105        Serial.println("Nachricht empfangen, aber falsche ID oder Datenlaenge.
106            ");
107        display.clearBuffer();
108        display.drawStr(0, 10, "Fehler: Nachricht!");
109        display.sendBuffer();
110    }
111    } else {
112        Serial.println("Keine Nachricht empfangen.");
113        display.clearBuffer();
114        display.drawStr(0, 10, "Fehler: Kein Empfang!");
115        display.sendBuffer();
116    }
}
```

## D. Anhang: CAN-Bus-Modul mit Stiftleisten-Verbindung)

In diesem Anhang wird das CAN-Bus-ECU-Modul, das als Sender fungiert, anhand von Abbildungen D.1, D.2 und D.3 näher vorgestellt. Die zugehörigen CAD-Modelle wurden so angepasst, dass kompakte Steckverbinder mit Stiftleisten [20] verwendet werden können. Diese Steckverbindungen verfügen über Rasternasen und sind besonders gut für Umgebungen mit hoher mechanischer Belastung und Vibration geeignet.



Abbildung D.1.: ECU-Modul mit Steckverbinder.

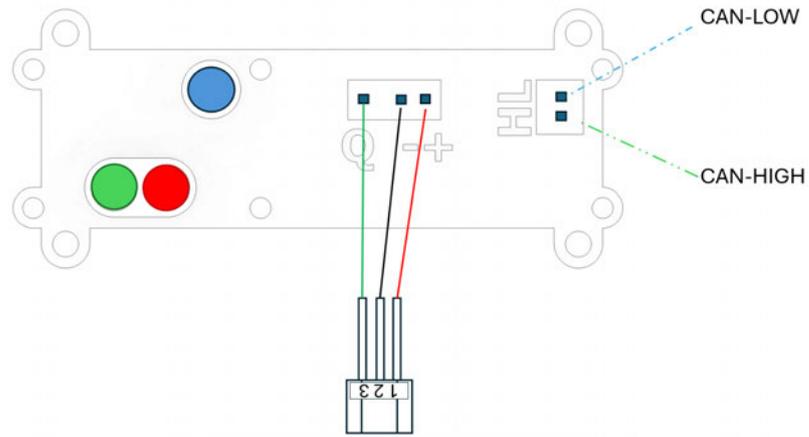


Abbildung D.2.: ECU-Modul mit Steckverbinder, Hall-Sensor-Anschluss.

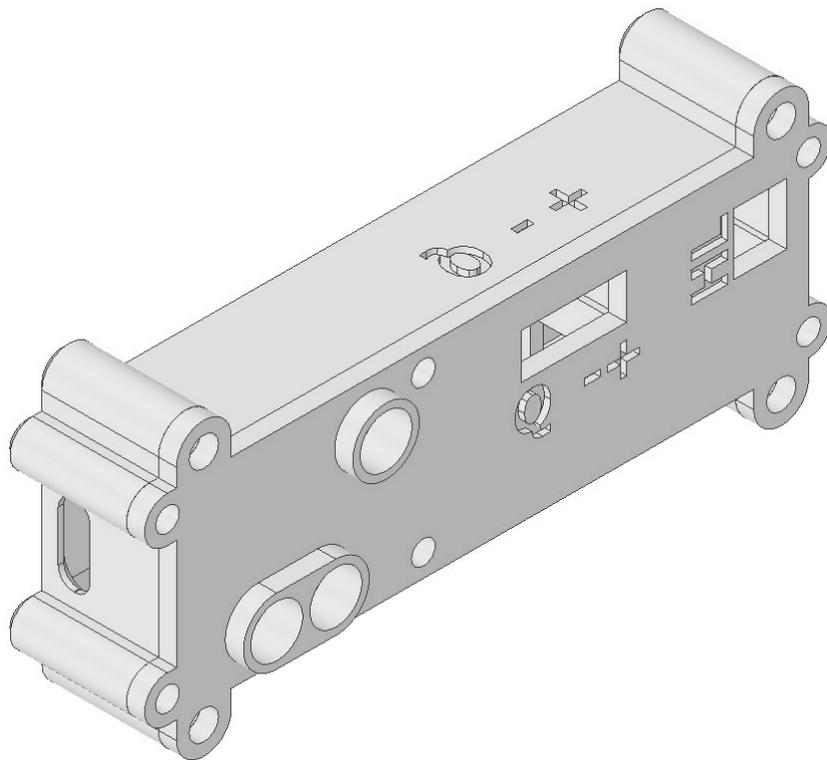


Abbildung D.3.: CAD Baugruppe, ECU-Gehäuse.

# Literaturverzeichnis

- [1] az-delivery.de. Link zur STM32, 2025. Letzter Zugriff: [07.02.2025].
- [2] balticdiag.com. Link zur FRC-EP170, 2025. Letzter Zugriff: [18.03.2025].
- [3] Benedikt Kleinheit. Bewertung und Lösungsfindung zur Problematik der Raddrehzahlerfassung im Projekt AUDEx (Fallstudie), 2022. Letzter Zugriff: [14.02.2025].
- [4] Yannick Bünn and Eric Franz. Konzeptionierung, Aufbau und Inbetriebnahme eines 1:X-FPV RC-Fahrzeugmodells (BA), 2020. Letzter Zugriff: [18.03.2025].
- [5] ABSIMA + Team C. Link zur Conrad, 2018. Letzter Zugriff: [18.03.25].
- [6] christians-shop.de. Esp32\_wroom. Link zur ESP32\_WROOM, 2025. Letzter Zugriff: [07.02.2025].
- [7] Christoph Olbrich. Link zur ipg-automotive, 2024. Letzter Zugriff: [28.03.2025].
- [8] DIOTEC. Link zur DIOTEC, 2025. Letzter Zugriff: [28.03.2025].
- [9] emotive. Link zur emotive, 2002. Letzter Zugriff: [18.03.25].
- [10] falstad. Link zur Falstad, 2025. Letzter Zugriff: [28.03.2025].
- [11] farnell.com. Link zur Farnell, 2025. Letzter Zugriff: [16.02.2025].
- [12] E. H. Hall. On a new action of the magnet on electric currents. *American Journal of Mathematics*, 2(3):287–292, 1879. Link – Letzter Zugriff: [30.03.2025].
- [13] infineon.com. Link zur Infineon, 2025. Letzter Zugriff: [14.02.2025].
- [14] IXXAT. Link zur FRC-EP170, 2024. Letzter Zugriff: [18.03.2025].
- [15] Jan Philipp Laskowski und Patrick Tews. Fehleranalyse des bestehenden Sensorkonzeptes für die Abnahme der Raddrehzahlen an AUDEx-Fahrezug (Projektarbeit), 2023. Letzter Zugriff: [14.02.2025].

- [16] jf-elektronik.de. Link zur ESP32-C3, 2025. Letzter Zugriff: [07.02.2025].
- [17] Leonid Dubovyy. Simulation und Fehlerdiagnose in der CAN-Bus-Umgebung: Integration von ECUs und OBDII-Schnittstelle (Studienarbeit), 2024. Letzter Zugriff: [14.02.2025].
- [18] Me-Systeme. Link zur Me-Systeme, 2022. Letzter Zugriff: [18.03.25].
- [19] MathWorks Melda Ulusoy. Link zur Three-Phase Inverter, 2025. Letzter Zugriff: [18.03.2025].
- [20] reichelt.de. Link zur Reichelt, 2025. Letzter Zugriff: [02.04.25].
- [21] roboter-bausatz.de. Link zur RP2040, 2025. Letzter Zugriff: [07.02.2025].
- [22] supermagnete.de. Link zur SuperMagnete, 2025. Letzter Zugriff: [02.04.25].
- [23] TE.com. Link zur TE, 2025. Letzter Zugriff: [16.02.2025].
- [24] VECTOR. Link zur Vector, 2014. Letzter Zugriff: [18.03.25].

*Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.*

A solid black rectangular box used to redact the signature of the author.

Hamburg, 17. April 2025 Leonid Dubovyy