

Diplomarbeit

Erarbeitung einer praktischen Begleitaufgabe im Bereich Package inklusive Sensorik

eingereicht an der

Fakultät Kraftfahrzeugtechnik der
Westsächsischen Hochschule Zwickau
zur Erlangung des akademischen Grades eines

Diplomingenieurs (FH)

Vorgelegt von: cand. ing. **Köhler, Hans Henning** Geb. am 26.01.1997

Studiengang: Kraftfahrzeugtechnik
Studienschwerpunkt: Karosserieentwicklung und - konstruktion

Ausgegeben von: Prof. Dr.-Ing. Rick Voßwinkel
Erstbetreuer: Prof. Dr.-Ing. Rick Voßwinkel
Zweitbetreuer: Prof. Dr.-Ing. Lutz Nagel

Autorenreferat

Köhler, Hans Henning: Erarbeitung einer praktischen Begleitaufgabe im Bereich Package inklusive Sensorik. Westsächsische Hochschule Zwickau, Fakultät Kraftfahrzeugtechnik, Diplomarbeit, 2026

51 Seiten, 16 Bilder, 2 Tabellen, 5 Anlagen

Das Wahlmodul „Ergonomie und Package“ der Fakultät Kraftfahrzeugtechnik vermittelt seine Inhalte bislang überwiegend theoretisch. Die vorliegende Arbeit ergänzt es um eine praktische Begleitaufgabe: Studierende integrieren die Komponenten eines autonomen Modellfahrzeugs im Maßstab 1:10 in einen vorgegebenen Bauraum und bewerten ihre Lösung anhand integrierter Kennzahlen. Die didaktische Strukturierung orientiert sich am Drei-Phasen-Modell nach Wilhelm und Brovelli. In der Konstruktion kommt ein parametrisches CAD-Modell in CATIA 3DEXPERIENCE zum Einsatz. Ein Arduino-Mikrocontroller mit Time-of-Flight-Sensorik bildet die Hardwarebasis; die Programmierung kann KI-gestützt erfolgen. Aus der Arbeit gehen das Modulkonzept, das parametrische CAD-Modell mit integrierter Bewertungslogik, ein Hardware-Prototyp sowie ein KI-Leitfaden hervor. Die Erprobung mit Studierenden steht als nächster Entwicklungsschritt aus.

Selbstständigkeitserklärung



WHZ Westsächsische
Hochschule Zwickau
Hochschule für Mobilität

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit in allen Teilen selbstständig angefertigt und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt habe, und dass die Arbeit in gleicher oder ähnlicher Form in noch keiner anderen Prüfung vorgelegen hat. Mir ist bewusst, dass ich Autor/in der vorliegenden Arbeit bin und volle Verantwortung für den Text trage. ☒

Ich erkläre, dass ich wörtlich oder sinngemäß aus anderen Werken – dazu gehören auch Internetquellen – übernommene Inhalte als solche kenntlich gemacht und die entsprechenden Quellen angegeben habe. ☒

Mir ist bewusst, dass meine Arbeit auf Plagiate überprüft werden kann. Mir ist bekannt, dass es sich bei der Abgabe eines Plagiats um ein schweres akademisches Fehlverhalten handelt und dass Täuschungen nach der für mich gültigen Prüfungsordnung geahndet werden. ☒

Zusätzlich versichere ich, dass ich auf künstlicher Intelligenz (KI) basierende Werkzeuge nur in Absprache mit den Prüfern verwendet habe. Dabei stand meine eigene geistige Leistung im Vordergrund, und ich habe jederzeit den Prozess steuernd bearbeitet. ☒

Diese Werkzeuge habe ich im Quellenverzeichnis in der Rubrik „Übersicht verwendeter Hilfsmittel“ mit ihrem Produktnamen und einer Übersicht des im Rahmen dieser Prüfungs-/Studienarbeit genutzten Funktionsumfangs unter Angabe der Textstelle in der Arbeit vollständig aufgeführt. ☒

Ich versichere, dass ich keine KI-basierten Tools verwendet habe, deren Nutzung die Prüfer explizit schriftlich ausgeschlossen haben. Ich bin mir bewusst, dass die Verwendung von Texten oder anderen Inhalten und Produkten, die durch KI-basierte Tools generiert wurden, keine Garantie für deren Qualität darstellt. ☒

Ich verantworte die Übernahme jeglicher von mir verwendeter maschinell generierter Passagen vollumfänglich selbst und trage die Verantwortung für eventuell durch die KI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate. ☒

Zwickau, 27.05.2026

Ort, Datum

Unterschrift



Inhalt

1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
1.3 Aufbau der Arbeit	3
2 Theoretische Grundlagen	4
2.1 Package im Fahrzeugbau	4
2.2 Stand der Technik	7
2.3 Technische Grundlagen	11
2.4 Didaktische Grundlagen	17
3 Konzeption des Lehrmoduls	20
3.1 Anforderungsanalyse	20
3.2 Didaktischer Bezug	21
3.3 Aufgabenstellung für Studierende	22
3.4 Bewertungskriterien und KPIs	24
4 Aufbau und Erprobung des Modellfahrzeugs	29
4.1 Parametrische Konstruktion	29
4.2 Integrierte Bewertungslogik	36
4.3 Auswahl der Komponenten	38
5 Lehrmaterialien	41
5.1 CAD-Daten und Vorlagen	41
5.2 KI-Leitfaden und Anschlussdokumentation	42
5.3 Dokumentationsvorlagen	44
6 Diskussion	46
7 Zusammenfassung	50
Literaturverzeichnis	52
Übersicht verwendeter Hilfsmittel	54
Anlagen	I

I. Verzeichnis der Bilder

Bild 1: GPIO-Pinout Arduino Nano. Quelle: arduino.cc	11
Bild 2: ToF-Sensor VL53L0X. Quelle: hwlibre.com	15
Bild 3: Schaltplan des Aufbaus	16
Bild 4: Karosserieteil mit zu geringer Wandstärke	27
Bild 5: Fahrzeugkoordinatensystem nach DIN ISO 4130 (1979)	29
Bild 6: Gesamtansicht ohne Karosserie	30
Bild 7: Unterseite der Grundplatte mit Schwalbenschwanz-Aufnahmen	30
Bild 8: Grundplatte, Radstand 216 mm, Spurweite 131 mm	31
Bild 9: Grundplatte, Radstand 316 mm, Spurweite 231 mm	31
Bild 10: Sensorhalter mit PCB-Aufnahme	32
Bild 11: Draufsicht mit Messkegeln der ToF-Sensoren	33
Bild 12: Wannenhalter für Breadboard und Powerbank	34
Bild 13: Aufbau mit eingeblendetem Bewertungs-Set	35
Bild 14: Gesamtansicht mit Karosserie	35
Bild 15: Informationsfenster der zentralen KPI-Auswertung	38
Bild 16: Hardware-Prototyp des Modellfahrzeugs	51

II. Verzeichnis der Tabellen

Tabelle 1: Bewertungsstufen und Schwellwerte der KPIs	28
Tabelle 2: Technische Umsetzung der KPIs in der CAD-Datei	36

III. Verzeichnis der Anlagen

Anlage	Titel
1	Aufgabenstellung für Studierende
2	CAD-Zeichnung
3	Quellcode des Modells
4	Konstruktionstagebuch
5	Bewertungsbogen

IV. Kurzzeichenverzeichnis

Kurzzeichen	Einheit	Bedeutung
$V_{Bauteile}$	mm ³	Gesamtvolumen der verbauten Komponenten
$V_{Innenraum}$	mm ³	nutzbares Volumen innerhalb der Karosserie
c	m/s	Lichtgeschwindigkeit in Luft
d	m	Abstand zum Objekt
g	m/s ²	Erdbeschleunigung
t	s	Zeit zwischen Senden und Empfangen
η	-	Bauraumeffizienz

V. Abkürzungsverzeichnis

Abkürzung	Bedeutung
3DX	CATIA 3DEXPERIENCE
CAD	Computer-Aided Design
CAN	Controller Area Network
COG	Center of Gravity (Schwerpunkt)
DMU	Digital Mock-Up
EKL	Engineering Knowledge Language
FDM	Fused Deposition Modelling
GPIO	General Purpose Input/Output
IDE	Integrated Development Environment
I ² C	Inter-Integrated Circuit
KBE	Knowledge-Based Engineering
KEK	Karosserieentwicklung und -konstruktion
KI	Künstliche Intelligenz
KPI	Key Performance Indicators
LiDAR	Light Detection and Ranging
LIN	Local Interconnect Network
LLM	Large Language Model
OLED	Organic Light Emitting Diode
PBL	Problembasiertes Lernen
PCB	Printed Circuit Board
PLA	Polylactid (Polymilchsäure)
PWM	Pulsweitenmodulation

SCL	Serial Clock
SDA	Serial Data
ToF	Time-of-Flight

VI. Vorwort

Die vorliegende Arbeit wurde im Rahmen meines Diplomstudiums in der Kraftfahrzeugtechnik an der Westsächsischen Hochschule Zwickau angefertigt.

An dieser Stelle möchte ich allen Personen meinen Dank aussprechen, die zum Gelingen dieser Arbeit beigetragen haben.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Voßwinkel für die Betreuung der Arbeit, die fachlichen Anregungen sowie die konstruktiven Hinweise während der Bearbeitungsphase.

Herrn Prof. Dr.-Ing. Nagel danke ich für die Übernahme der Zweitbetreuung sowie für die wertvollen fachlichen Impulse. Sein Engagement in der Lehre hat meine Begeisterung für die Karosserieentwicklung geweckt und mich bei der Wahl meines Studienschwerpunkts bestärkt.

Mein Dank richtet sich außerdem an Frau Katharina Trömer vom HS3D-Center der Hochschule. Sie hat mich beim Wechsel von CATIA V5 auf CATIA 3DEXPERIENCE unterstützt, Materialien für die Elektronikkomponenten bereitgestellt und bei sämtlichen Fragen zur Software mit Rat und Tat zur Seite gestanden.

Abschließend möchte ich meiner Lebensgefährtin Jasmin sowie meiner Familie für ihre Geduld und ihr Verständnis während der gesamten Bearbeitungszeit meinen aufrichtigen Dank aussprechen.

1 Einleitung

1.1 Motivation

Im Wahlmodul „Ergonomie und Package“ werden Studierenden außerhalb des Studienschwerpunkts Karosserieentwicklung und -konstruktion die grundlegenden Prinzipien des Fahrzeug-Packages vermittelt. Bisher erfolgt diese Vermittlung jedoch hauptsächlich auf theoretischer Basis. Im Pflichtmodul existiert bereits ein praxisnahes Konzept, das es den Studierenden ermöglicht, theoretische Inhalte unmittelbar anzuwenden. Das didaktische Prinzip des objektorientierten Lernens, also die Festigung von Wissen durch die Anwendung auf konkrete praktische Aufgaben, hat sich in diesem Zusammenhang bewährt. Ziel dieser Diplomarbeit ist es, auch im Wahlmodul einen vergleichbaren Praxisbezug herzustellen, indem eine geeignete Begleitaufgabe entwickelt wird, die die Grundlagen des Package-Designs anhand eines überschaubaren, jedoch realistischen Projekts erfahrbar macht.

Die räumliche Unterbringung komplexerer Fahrzeugkomponenten unter Einhaltung funktionaler, ergonomischer und sicherheitstechnischer Anforderungen gehört zu den zentralen Herausforderungen der modernen Fahrzeugentwicklung. Ihre Bedeutung nimmt durch Elektrifizierung und Automatisierung weiter zu.

In der vorangegangenen Studienarbeit zum selben Thema (Köhler 2025) wurde die technische Machbarkeit eines solchen Lehrkonzepts untersucht. Als Ergebnis entstand ein autonomes Modellfahrgestell mit grundlegender Technik, das die typischen Package-Herausforderungen der Automobilindustrie in vereinfachter Form abbildet. Der Aufbau erfolgte mithilfe von Packagedaten, wobei der Fokus auf Passformen und der Druckbarkeit der Bauteile lag, die für das Fused Deposition Modelling (FDM)-Verfahren, als gängige Form des 3D-Drucks, ausgelegt wurden. Die Positionierung von Ultraschallsensoren, Mikrocontrollern, Servomotoren und Breadboards musste dabei in einem möglichst kompakten Bauraum erfolgen, was ähnliche Kompromisse erforderte wie im realen Fahrzeugbau. So benötigen beispielsweise die Sensoren freie Sichtfelder für zuverlässige Messungen, was der Integration von Radar- oder LiDAR-Systemen (Light Detection and Ranging) in Serienfahrzeugen entspricht.

Des Weiteren wurde in der Studienarbeit belegt, dass die Programmierung des Modells weitgehend an Large Language Models (LLMs) übergeben werden kann. Ein Vergleich zwischen Claude, ChatGPT und Gemini zeigte dabei signifikante Unterschiede im generierten Code: Während alle drei Modelle funktionsfähigen

Code lieferten, erwies sich Claudes konservativer Ansatz in komplexen Testszenarien als überlegen. Diese Erkenntnis unterstützt die Idee des Lehrmoduls, den Fokus auf die Konstruktion und Auslegung des Fahrzeugs zu setzen, anstatt umfangreiche Programmierkenntnisse vorauszusetzen.

In der Folge brachte die Studienarbeit als Ergebnis einen funktionsfähigen Prototypen hervor, gleichwohl stellt dies noch keine abschließende Lehrereinheit dar. Es fehlten eine ausformulierte Aufgabenstellung, Bewertungskriterien sowie aufbereitete Materialien für den Einsatz im Unterricht. Die Erkenntnisse dienen nun als Grundlage für das Verfeinern des Modells und der Technik sowie für die Entwicklung einer vollständigen Begleitaufgabe mit allen erforderlichen Zusatzmaterialien.

1.2 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung einer praxisorientierten Begleitaufgabe für das Wahlmodul „Ergonomie und Package“, die theoretische Inhalte praktisch anwendbar macht und Studierende vor realistische Probleme der Fahrzeugentwicklung stellt. Die Zielgruppe ist dabei Studierende außerhalb des Schwerpunkts Karosserieentwicklung und -konstruktion (KEK), denen ein grundlegendes Verständnis für Package-Zielkonflikte und deren Lösungsstrategien vermittelt werden soll. Daraus resultierend sollen die Studierenden in der Lage sein, vorgegebene Komponenten unter Berücksichtigung funktionaler und konstruktiver Anforderungen in einem begrenzten Bauraum anzuordnen, entwicklungstypische Zielkonflikte zu erkennen und dokumentierte Lösungsstrategien zu entwickeln.

Begleitet wird die Aufgabenstellung von Materialien, die den Einstieg erleichtern und eine Kompatibilität zwischen Modell und Bauteilen gewährleisten sollen. Hierzu zählen ein parametrisches Grundgestell als CAD-Vorlage, Packagedaten der zu verbauenden Komponenten sowie technische Dokumentationen zum Anschluss der Bauteile an den Mikrocontroller. Diese Dokumentationen umfassen Pin-Belegungen, Anschlussschemata und Hinweise zu Besonderheiten der einzelnen Komponenten, sodass Studierende diese Informationen als Grundlage für die KI-gestützte Code-Generierung nutzen können. Ein begleitender Leitfaden soll es den Studierenden ermöglichen, die Programmierung mithilfe von LLMs selbstständig durchzuführen, ohne dass umfangreiche Programmierkenntnisse vorausgesetzt werden. Auf diese Weise bleibt der Fokus auf der konstruktiven Auseinandersetzung mit dem Package.

Um den kollaborativen Charakter realer Entwicklungsprozesse abzubilden und den Austausch über unterschiedliche Lösungsansätze zu fördern, ist die Bearbeitung als Gruppenarbeit angedacht. Im Ergebnis soll eine schlüsselfertige Lehrereinheit entwickelt werden, die interessierte Dozenten in ihren Unterricht übernehmen können.

1.3 Aufbau der Arbeit

Die Arbeit gliedert sich in sieben Kapitel, deren Struktur dem Ablauf vom theoretischen Fundament bis zur praktischen Umsetzung folgt.

Ausgangspunkt bildet das zweite Kapitel mit den theoretischen Grundlagen in vier Abschnitten. Zunächst wird der Begriff des Fahrzeug-Packages definiert, seine wesentlichen Aspekte erläutert sowie der Einfluss auf Fahrverhalten und Sensorik beschrieben. Im Anschluss erfolgt eine Aufarbeitung des Stands der Technik parametrischer Gesamtfahrzeugmodelle und wissensbasierter CAD-Templates, um den vorliegenden Ansatz in den Forschungskontext einzuordnen. Die technischen Grundlagen des eingesetzten Mikrocontrollersystems, der Sensortechnik sowie des Schaltungsaufbaus werden im dritten Abschnitt behandelt, bevor der Letzte die didaktischen Grundlagen des problemorientierten Unterrichts und das strukturgebende Drei-Phasen-Modell darlegt.

Aufbauend auf diesen Grundlagen überführt das dritte Kapitel die gewonnenen Erkenntnisse in die Konzeption des Lehrmoduls. Nach einer Anforderungsanalyse wird der didaktische Bezug zu den zuvor beschriebenen Prinzipien hergestellt, bevor die konkrete Aufgabenstellung für Studierende sowie die Bewertungskriterien ausgearbeitet werden. Gegenstand des vierten Kapitels sind die parametrische Konstruktion des Modellfahrzeugs, die in die CAD-Arbeitsdatei integrierte Bewertungslogik sowie die Auswahl der verbauten Komponenten. Im fünften Kapitel werden die entwickelten Lehrmaterialien beschrieben, darunter CAD-Daten und Vorlagen, ein Leitfaden zur KI-gestützten Programmierung sowie Dokumentationsvorlagen für Studierende. Eine kritische Auseinandersetzung mit den Ergebnissen erfolgt im sechsten Kapitel, bevor das siebte Kapitel die Arbeit in einer Zusammenfassung abschließt.

2 Theoretische Grundlagen

2.1 Package im Fahrzeugbau

2.1.1 Definition und Bedeutung des Fahrzeug-Packages

Im Fahrzeugentwicklungsprozess bezeichnet der Begriff „Package“ mehr als nur die räumliche Anordnung von Komponenten. Nach Grabner und Nothhaft umfasst Package die Koordination und Harmonisierung der Anforderungen an Bauräume, Ergonomie und Gesamteigenschaften eines Fahrzeugs im Zusammenwirken mit allen beteiligten Fachstellen und erstreckt sich über den gesamten Produktlebenszyklus von der Idee bis zum Serienauslauf. Das Hauptaugenmerk liegt dabei auf der Verwaltung der gesamten Fahrzeuggeometrie (vgl. Grabner/Nothhaft 2006, S. 22).

Es wird verdeutlicht, dass Package nicht nur eine technische Disziplin ist, sondern auch eine koordinierende Funktion im gesamten Entwicklungsprozess übernimmt. Insbesondere im Spannungsfeld zwischen verschiedenen Anforderungen zeigt sich die Bedeutung eines Package-Konzepts: Eine gelungene Raumaufteilung ermöglicht nicht nur die effiziente Nutzung des verfügbaren Volumens, sondern auch die Minimierung von Interessenskonflikten zwischen den technischen Anforderungen.

Im Laufe der Zeit haben sich die Methoden des Packages in der Fahrzeugentwicklung grundlegend gewandelt. Während physische Tonmodelle lange Zeit das zentrale Werkzeug zur Überprüfung von Proportionen und Bauraumanordnungen darstellten, ermöglicht der digitale Wandel heute die vollständige Abbildung eines Fahrzeugs in virtueller Umgebung. Das sogenannte Digital-Mock-Up (DMU) hat das physische Modell in weiten Teilen der frühen Entwicklungsphase abgelöst und ermöglicht die parallele Bearbeitung durch verschiedene Fachbereiche, eine frühzeitige Kollisionsprüfung sowie die rechnergestützte Auswertung von Package-Kennzahlen, noch bevor ein erster physischer Prototyp entsteht. (vgl. Grabner/Nothhaft 2006, S. 3 f.)

2.1.2 Aspekte des Packages

Eine der wesentlichen Herausforderungen des Packages ist das Unterbringen zahlreicher Komponenten in einem begrenzten Bauraum. Bereits die Wahl einer Fahrzeugklasse legt Basisparameter wie Sitzposition, Kofferraumvolumen und die Gestaltung der Fahrgastzelle fest. Diese grundlegenden Anforderungen müssen mit zahlreichen Randbedingungen in Einklang gebracht werden: Fertigungsprozesse, gesetzliche Vorschriften und die äußeren Abmessungen des Fahrzeugs setzen enge Grenzen, die häufig im Konflikt mit der gewünschten Innenraumgröße stehen. Neben Komfort und Ergonomie muss die Gestaltung des Bauraums auch sicherheitsrelevante Vorgaben erfüllen. Hierzu zählen unter anderem der Fußgängerschutz, gesetzlich definierte Knautschzonen sowie die geschützte Unterbringung von Hochvoltbatterien in Elektrofahrzeugen. Zugleich ist es notwendig, den verfügbaren Bauraum für die Integration von Antrieb, Energiespeicher, Fahrwerk, Kühlsystemen und Elektronik effizient zu nutzen, wobei Funktionskonflikte sowie thermische oder elektromagnetische Beeinflussungen zwischen den Komponenten vermieden werden müssen.

Auch die Zugänglichkeit ist entscheidend für das Package-Design. Gut erreichbare Baugruppen erleichtern Produktion, Wartung und Reparaturen, reduzieren Montagezeiten und Servicekosten. Eine durchdachte Anordnung der Komponenten, die die Zugänglichkeit berücksichtigt, ist in frühen Entwicklungsphasen unerlässlich. Als anschauliches Beispiel für die Bedeutung des Packages kann die Entwicklung von Elektrofahrzeugen herangezogen werden. Durch den Wegfall des Verbrennungsmotors und des Getriebetunnels ergeben sich grundlegend neue Möglichkeiten für die Bauraumgestaltung. Viele Hersteller setzen daher auf Purpose-Design-Plattformen (vgl. Karle 2022, S. 8 f.), bei denen die Hochvoltbatterie flach im Fahrzeugboden integriert ist und die Elektromotoren direkt an den Achsen sitzen. Mithilfe dieser Architektur wird der Fahrzeugschwerpunkt deutlich gesenkt, gleichzeitig entstehen Freiräume im vorderen und hinteren Bereich des Fahrzeugs, die als zusätzlicher Stauraum genutzt werden können. Die konsequente Trennung zwischen der Antriebsplattform im Unterboden und dem darauf aufbauenden Fahrgastraum ermöglicht zudem eine modulare Fahrzeugentwicklung, bei der verschiedene Karosserievarianten auf derselben technischen Basis realisiert werden können.

2.1.3 Einfluss auf Fahrverhalten und Sensorik

Das Package beeinflusst sowohl das Fahrverhalten als auch die Leistungsfähigkeit der Sensorik in hohem Maß. Parameter wie Radstand, Spurweite, Schwerpunktlage und Achslastverteilung wirken sich direkt auf Dynamik, Komfort und Stabilität aus. Gleichzeitig bestimmt die Anordnung von Komponenten wie Motor, Batterie oder Fahrwerkselementen die strukturelle Steifigkeit, die Gewichtsverteilung und die Möglichkeiten zur Fahrwerksabstimmung. Eine ungünstige Schwerpunktlage kann beispielsweise zu erhöhter Wankneigung in Kurven führen, während eine unausgewogene Achslastverteilung die Fahrdynamik beeinträchtigt.

Für die Sensorik ist das Package ebenfalls entscheidend, da Einbauhöhe, Ausrichtung und freie Sichtfelder von Kamera-, LiDAR- oder Radarsystemen durch Karosserieform, Anbauteile und aerodynamische Anforderungen direkt beeinflusst werden. Auch Faktoren wie Verschmutzungsneigung, Temperaturbelastung und elektromagnetische Störeinflüsse hängen stark von der Integration in den Bauraum ab. Ein sorgfältig gestaltetes Package bildet die Basis für ein präzises Fahrverhalten und eine zuverlässige Umgebungswahrnehmung. Unangemessene Layouts hingegen erfordern einen späteren Einsatz von Software oder zusätzlicher Hardware zur Kompensation.

Die Package-Gestaltung beeinflusst somit nicht nur mechanische, sondern zunehmend auch elektronische und softwarebasierte Fahrzeugeigenschaften, sodass eine integrative Betrachtung aller Systeme bereits in frühen Entwicklungsphasen notwendig ist.

Für diese Zielkonflikte bietet die Integration von Kamerasystemen ein anschauliches Beispiel dafür, wie Package-Entscheidungen funktionale Anforderungen mit realen Randbedingungen in Einklang bringen müssen. Zum einen erfordert deren Funktionsweise ein möglichst freies Sichtfeld auf die unmittelbare Fahrzeugumgebung, zum anderen muss die Positionierung Umwelteinflüsse berücksichtigen. Eine Integration im Bereich der Scheinwerfer würde zwar eine günstige Perspektive auf den Nahbereich ermöglichen, jedoch sind diese Bereiche bei entsprechenden Witterungsbedingungen stark verschmutzungsgefährdet, was die Sensorfunktion beeinträchtigt. Aus diesem Grund werden Frontkameras überwiegend auf Höhe des Innenspiegels hinter der Windschutzscheibe verbaut, damit die Scheibenwischer ein freies Sichtfeld gewährleisten.

2.2 Stand der Technik

Die in dieser Arbeit verfolgte Methodik, ein parametrisches Modellfahrzeug mit integrierten Bewertungskriterien als Lehrmittel zu konstruieren, berührt zwei Forschungsfelder: die parametrische Fahrzeugmodellierung in frühen Entwicklungsphasen sowie die wissensbasierte Konstruktion mit eingebetteter Bewertungslogik. Folgender Abschnitt ordnet den gewählten Ansatz in den aktuellen Stand der Forschung ein und identifiziert die Forschungslücke, die diese Arbeit adressiert.

2.2.1 Parametrische Gesamtfahrzeugmodelle

In der industriellen Fahrzeugentwicklung zeichnet sich eine zunehmende Verwendung zentralisierter 3D-CAD-Mastermodelle ab, die als gemeinsame Datenbasis für die beteiligten Entwicklungsdisziplinen dienen. Stadler et al. beschreiben ein solches Konzeptwerkzeug, das aus drei verknüpften Hauptmodulen besteht: einem Geometriemodul mit parametrisierten Templates für verschiedene Fahrzeugtypen, einem Funktionsmodul für Gewichts- und Fahrdynamikberechnungen sowie einem zentralen Datenrepository zur Verwaltung aller Fahrzeugparameter (Stadler et al. 2013, S. 163 f.). Die Gesamtfahrzeuggeometrie wird dabei über wenige zentrale Parameter gesteuert, sodass Änderungen an einer Stelle automatisch auf alle verknüpften Komponenten wirken. Bidirektionale Schnittstellen zwischen Datenbank und 3D-Modell ermöglichen es, Parameteränderungen sowohl aus der CAD-Umgebung als auch aus externen Werkzeugen vorzunehmen (Stadler et al. 2013, S. 164).

Ein zentraler Aspekt dieser Mastermodelle ist die Integration verschiedener Validierungswerkzeuge in ein gemeinsames Modell. Beschrieben wird der Packageprozess von Stadler et al. als eine übergreifende geometrische Produktvalidierung, die zugleich als geometrisch-technische Schnittstelle zwischen den beteiligten Arbeitsbereichen dient (Stadler et al. 2013, S. 165). Dabei werden Anforderungen aus Ergonomie, Gesetzgebung, Styling und technischer Funktionalität in einem Modell zusammengeführt und iterativ optimiert. Das Packaging verläuft als iterativer Workflow, der von einem groben Konzeptlayout ausgeht und schrittweise zu einem umfassenden Gesamtfahrzeuglayout verfeinert wird (Stadler et al. 2013, S. 167 f.).

Für die Geometrievalidierung spielen DMU-Prozesse eine wesentliche Rolle: Komponentenpositionierung, Kollisions- und Abstandsanalysen sowie Schnittfunktionen ermöglichen eine frühzeitige Überprüfung des Fahrzeugkonzepts

im virtuellen Raum (Stadler et al. 2013, S. 167). Dieser Ansatz reduziert den Bedarf für physische Prototypen und erreicht bereits in frühen Entwicklungsphasen einen höheren Reifegrad des Produkts.

2.2.2 Wissensbasierte CAD-Templates

Vajna et al. beschreiben wissensbasierte Konstruktion als die Anwendung computergestützter Automatisierungs- und Unterstützungssysteme im Konstruktionsprozess, deren Ziel es ist, die Zugänglichkeit vorhandenen Konstruktionswissens zu jedem Zeitpunkt sicherzustellen und Routinetätigkeiten zu beschleunigen (Vajna et al. 2018, S. 424 f.).

Potenziale und Grenzen solcher CAD-Automatisierungen anhand konkreter Anwendungsfälle in der Karosserieentwicklung wurden von Kreis et al. untersucht. Die Autoren beschreiben verschiedene Stufen der CAD-Automatisierung mit zunehmender Komplexität: Auf der Grundstufe steht das parametrisch-assoziative Design, bei dem die Geometrie über Parameter und Relationen gesteuert wird. Aufbauend darauf folgt die Stufe des wissensbasierten CAD, bei der zusätzlich Formeln, Regeln und automatisierte Prüfoperationen in das Modell integriert werden. Höhere Stufen umfassen interaktive CAD-Anwendungen mit eingebetteten Softwaremodulen sowie übergeordnete CAD-Frameworks (Kreis et al. 2021, S. 852 f.).

Als Wissensträger im CAD-Modell identifizieren Kreis et al. unter anderem Parameterstrukturen, Relationen und Formeln, Template-Modelle, benutzerdefinierte Features sowie DMU-Funktionalitäten wie kinematische Simulationen und Kollisionsprüfungen (Kreis et al. 2021, S. 853). Diese Wissensträger ermöglichen, das Konstruktionswissen dauerhaft im Modell zu verankern und es bei jeder Parameteränderung automatisch anzuwenden.

Von besonderer Relevanz für die vorliegende Arbeit ist das Konzept der funktionalen CAD-Templates, das Kreis et al. detailliert beschreiben. Ein solches Template besteht aus einem Input-Bereich für Geometriedaten und Parameter, einem operativen Modul für die automatisierte Geometrieerstellung und -prüfung, Assistenzmodulen für die Benutzerführung sowie einem Output-Bereich, der Ergebnisdaten wie Geometrie, Simulationsergebnisse und Berichte bereitstellt (Kreis et al. 2021, S. 854 f.). Solche Workflows mit funktionalen Templates folgen einem vierstufigen Ablauf: Bereitstellung der Eingabedaten, Ausführung der implementierten Algorithmen, automatische Geometriegenerierung mit integrierter Kontrolle sowie Erstellung und Bereitstellung der Ausgabedaten.

Die Autoren illustrieren die praktische Anwendung anhand eines Steinschlag-Simulationstools, das direkt in die CAD-Umgebung eingebettet ist. Physikalische Berechnungen und Prüfoperationen sind im Template als Formeln und Regeln implementiert, sodass die Ergebnisse als farbcodierte Bewertungszonen auf der Karosseriefläche dargestellt werden können (Kreis et al. 2021, S. 860 f.). Dieses Beispiel zeigt, dass auch komplexe Bewertungen direkt im CAD-Modell realisiert werden können, ohne auf externe Berechnungstools zurückgreifen zu müssen.

Gleichzeitig weisen Kreis et al. auf Herausforderungen hin, die mit zunehmender Automatisierung einhergehen: Mit steigender Komplexität der eingebetteten Logik nimmt der Wartungsaufwand zu, während die Nachvollziehbarkeit der im Modell hinterlegten Regeln für unbeteiligte Nutzer abnimmt. Die Wahl der Automatisierungsstufe muss daher stets im Verhältnis zum erwarteten Nutzen betrachtet werden (Kreis et al. 2021, S. 857 f.).

2.2.3 Einordnung der vorliegenden Arbeit

Der in dieser Diplomarbeit verfolgte Ansatz lässt sich als vereinfachte Anwendung der beschriebenen KBE-Methodik (Knowledge-Based Engineering) interpretieren. Die Integration von Bewertungskriterien wie Schwerpunktlage, Sichtfeldanalyse und Bauraumeffizienz direkt in die CAD-Arbeitsdatei folgt dem Prinzip funktionaler Templates, bei denen die Bewertungslogik unmittelbar im Modell verankert ist.

Konkret lassen sich mehrere Parallelen zwischen dem industriellen Vorgehen und dem hier entwickelten Lehrmodul identifizieren: Die zentrale Parametrik von Radstand und Spurweite folgt dem Prinzip der parametrisch-assoziativen Geometriesteuerung, wie es Stadler et al. für industrielle Konzeptmodelle mit bidirektionaler Datenschnittstelle beschreiben (Stadler et al. 2013, S. 163 f.). Eine automatische Aktualisierung der Bewertungskörper bei Parameteränderungen entspricht dem Prinzip assoziativer Modellverknüpfungen, das Kreis et al. als Grundlage wissensbasierter Konstruktion identifizieren (Kreis et al. 2021, S. 852). Mithilfe der Möglichkeit zur Selbstprüfung während des Konstruktionsprozesses wird sich am iterativen Charakter des Packaging-Workflows orientiert, der von groben Konzeptlayouts schrittweise zu validierten Gesamtfahrzeuglayouts führt (Stadler et al. 2013, S. 167 f.). Die Organisation der Bewertungselemente in einem klar gekennzeichneten Geometrical Set adressiert die von Kreis et al. beschriebene Herausforderung der Nachvollziehbarkeit für unbeteiligte Nutzer (Kreis et al. 2021, S. 858 f.).

Während die in der Literatur beschriebenen industriellen Templates auf komplexen Berechnungsroutinen, umfangreichen Regelwerken und spezialisierten Softwaremodulen basieren, nutzt das Lehrmodul bewusst einfachere Mechanismen, wie parametrische Hilfskörper für die Schwerpunktbewertung, Volumenkörper für die Sichtfeldanalyse und eine Kollisionsprüfung als geometrisches Bewertungswerkzeug. Diese Reduktion ist didaktisch motiviert, da die Studierenden die Bewertungsmechanismen nachvollziehen und die Ergebnisse eigenständig interpretieren sollen, ohne auf spezialisierte KBE-Module zurückgreifen zu müssen. Im Sinne der von Kreis et al. beschriebenen Stufenlogik bewegt sich das Lehrmodul damit auf der Stufe des wissensbasierten CAD, oberhalb der rein parametrischen Modellierung, jedoch unterhalb interaktiver CAD-Anwendungen mit eigener Programmlogik (Kreis et al. 2021, S. 852 f.).

Die vorgestellten Arbeiten beschreiben parametrische Mastermodelle und wissensbasierte CAD-Templates, die in der industriellen Fahrzeugentwicklung zur automatisierten Validierung von Package-Konzepten eingesetzt werden. Hierbei richten sich die entwickelten Werkzeuge ausschließlich an Konstruktionsingenieure in der professionellen Entwicklung. Mit der Übertragung dieser Methodik auf den Bereich der Hochschullehre, bei der die Komplexität gezielt reduziert wird, um Studierenden ohne erweiterte CAD-Kenntnisse den Zugang zu den Prinzipien des Packageprozesses zu ermöglichen, wurde in der gesichteten Literatur bisher nicht behandelt. Ebenso fehlen Ansätze, die parametrische Fahrzeugmodelle mit integrierten Bewertungskriterien als didaktisches Werkzeug nutzen, um konstruktive Zielkonflikte im Package nachvollziehbar zu machen.

In der vorliegenden Arbeit wird diese Lücke adressiert, indem die grundlegenden Prinzipien parametrischer Packagemodelle mit eingebetteten Bewertungskriterien in ein Lehrmodul überführt werden. Dabei werden die in der Industrie etablierten Mechanismen wie zentrale Parametrik, assoziative Modellverknüpfungen und integrierte Prüfoperationen auf ein Komplexitätsniveau reduziert, das für Studierende im Wahlmodul zugänglich ist. Hieraus abgeleitete Aufgabenstellung und deren didaktische Einbettung werden in den folgenden Kapiteln beschrieben.

2.3 Technische Grundlagen

2.3.1 Arduino Nano

Die Arduino-Plattform ist ein vielseitiges System, das aus Open-Source-Hardware- und -Softwarekomponenten besteht. Sie richtet sich an Personen mit und ohne Vorkenntnisse in der Mikrokontrollertechnik und bietet einen niedrighschwelligen Einstieg in die Welt der Elektronik und Programmierung. Im Gegensatz zu einem klassischen Computer, der als Allzweckgerät mit eigenem Betriebssystem arbeitet, beschränkt sich der Arduino als Mikrocontroller auf die gezielte Steuerung von Bauteilen und die Verarbeitung vorgegebener Aufgaben. Er empfängt Eingangssignale von Sensoren, verarbeitet sie gemäß den im Programm vorgegebenen Anweisungen und steuert daraufhin die Aktoren an.

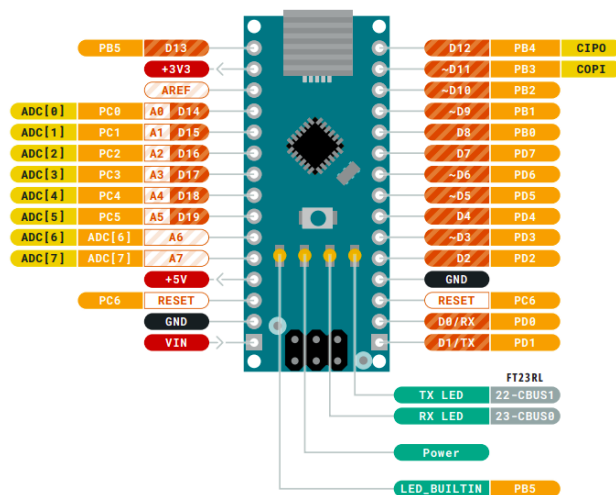


Bild 1: GPIO-Pinout Arduino Nano. Quelle: arduino.cc

Die Bauweise der Arduino-Modelle zeichnet sich durch die Offenlegung sämtlicher Anschlussmöglichkeiten aus. Über die GPIO-Pins (General Purpose Input/Output) des Arduino Nano (Bild 1) können externe Komponenten mittels Steck- und Lötverbindungen verbunden werden. Das ermöglicht den flexiblen Aufbau und die Anpassung verschiedener Schaltkreise. Diese Eigenschaft vereinfacht die Prototypentwicklung und Fehlerbehebung erheblich, weshalb die Plattform für den Einsatz in der Lehre geeignet ist.

In Arduino-Projekten kommen häufig Sensoren zum Einsatz, die physikalische Größen wie Distanz, Feuchtigkeit oder Temperatur erfassen und in elektrische Signale umwandeln. Diese Signale können analog oder digital sein, wobei an jedem Arduino die entsprechenden GPIO-Pins für beide Varianten vorhanden sind.

Aktoren hingegen sind die Systemkomponenten, die auf die Steuersignale des Arduino reagieren und daraufhin Funktionen ausführen. Im Kontext des Modellfahrzeugs sind dies die Servomotoren, die Bewegungen ausführen, sowie ein OLED-Display (Organic Light Emitting Diode), das als Statusanzeige dienen kann. Die Programmierung des Mikrocontrollers erfolgt mit der Arduino IDE (Integrated Development Environment), einer plattformübergreifenden Entwicklungsumgebung. Es wird eine vereinfachte Programmiersprache genutzt, die an C und C++ angelehnt ist. Ein Arduino-Programm, auch „Sketch“ genannt, gliedert sich grundsätzlich in drei Bestandteile: Im optionalen Deklarationsbereich werden Bibliotheken eingebunden und globale Variablen definiert.

Die Funktion `setup()` wird beim Start des Mikrocontrollers einmalig ausgeführt und dient der Initialisierung von Pin-Modi und Peripheriegeräten.

Darauf folgt `loop()`, die die Hauptprogrammlogik enthält und nach `setup()` kontinuierlich wiederholt wird, bis die Stromversorgung unterbrochen wird oder eine Abschlussbedingung erfüllt ist.

Arduino IDE stellt eine Vielzahl integrierter und externer Bibliotheken zur Verfügung, die die Umsetzung typischer Aufgaben erheblich erleichtern. So existieren spezialisierte Bibliotheken, unter anderem zur Steuerung von Servomotoren, zum Auslesen verschiedener Sensoren oder zur Kommunikation mit Displays. In diesen Bibliotheken sind vordefinierte Funktionen enthalten, wodurch sich auch komplexe Prozesse effizient und mit geringem Programmieraufwand realisieren lassen.

Die Ansteuerung der im Modellfahrzeug eingesetzten Komponenten erfordert ein grundlegendes Verständnis der Signalverarbeitung zwischen Mikrocontroller und Peripherie. Zentral ist dabei die Pulsweitenmodulation (PWM), ein Verfahren zur Steuerung von Aktoren mittels digitaler Signale.

Bei der PWM wird ein digitales Signal mit konstanter Frequenz erzeugt, dessen Pulsbreite jedoch variiert werden kann. Das Verhältnis zwischen Einschaltzeit und Periodendauer wird als Tastgrad bezeichnet und bestimmt die an den Aktor übertragene effektive Leistung. Bei den verwendeten Servomotoren vom Typ FS90R interpretiert die interne Elektronik die Pulslänge als Geschwindigkeits- und Richtungsvorgabe: Eine Pulslänge von 1,5 Millisekunden entspricht dem Stillstand, während Werte darüber und darunter Rotation in entgegengesetzte Richtungen mit entsprechender Geschwindigkeit bewirken. Der Arduino stellt mehrere PWM-fähige Pins zur Verfügung, die über die Servo-Bibliothek angesteuert werden können,

sodass die Studierenden keine zusätzliche Programmierung zur Erzeugung der Signale vornehmen müssen.

Je nach Sensorart erfolgt die Signalverarbeitung. Die im Modell verwendeten Time-of-Flight-Sensoren (ToF) kommunizieren über den „I²C-Bus“ (Inter-Integrated Circuit) mit dem Mikrocontroller, einem seriellen Protokoll, das die Übertragung von Daten über lediglich zwei Leitungen (SDA „Serial Data“ für Daten, SCL „Serial Clock“ für das Taktsignal) ermöglicht. Diese Art der Kommunikation erlaubt den gleichzeitigen Betrieb mehrerer Sensoren auf einem einzigen Bus, wobei jeder Sensor eine eindeutige Adresse besitzt. Entsprechende Bibliotheken abstrahieren die Komplexität des Protokolls, sodass Messwerte über einfache Funktionsaufrufe abgerufen werden können.

Die Spannungsversorgung des Systems basiert auf der 5-V-Logik des Arduino Nano. Sowohl die Servomotoren als auch die Sensoren werden mit dieser Spannung betrieben, wodurch zusätzliche Spannungswandler überflüssig werden und die Komplexität der Schaltung reduziert wird. Bei der Auswahl der Energiequelle muss die Stromaufnahme der Komponenten berücksichtigt werden: Während der Arduino selbst nur wenige Milliampere benötigt, benötigen die vier Servomotoren unter Last mehrere hundert Milliampere.

2.3.2 Sensortechnik

ToF-Sensoren wie der VL53L0X von STMicroelectronics bestimmen Abstände durch die Zeitmessung eines ausgesandten Lichtimpulses. Ein kurzer Infrarotimpuls mit einer Wellenlänge von 940 Nanometern wird ausgesendet, von einem Objekt reflektiert und vom Sensor wieder empfangen. Aus der gemessenen Laufzeit lässt sich der Abstand zum Objekt wie in Formel (1) berechnen:

$$d = \frac{t \cdot c}{2} \quad (1)$$

Dabei bezeichnet d den Abstand zum Objekt, t die Laufzeit des Lichtimpulses zwischen Aussenden und Empfangen sowie c die Lichtgeschwindigkeit mit etwa 3×10^8 m/s. Die Zeit für einen Abstand von einem Meter beträgt etwa 6,7 ns, weshalb die Elektronik des Sensors diese präzise Zeitmessung übernimmt.

Mit dem VL53L0X lassen sich Entfernungen von 30 bis 2000 mm erfassen, wobei die in der Praxis zuverlässige Reichweite je nach Oberflächenbeschaffenheit und Umgebungslicht bei etwa 1200 mm liegt.

Unter idealen Bedingungen beträgt die Auflösung 1 Millimeter bei einer Genauigkeit von ± 3 %. Der Messkegel des Sensors umfasst etwa 25° , was eine großflächige

Erfassung des relevanten Nahbereichs ermöglicht. Der eingesetzte Infrarotlaser ist für das menschliche Auge unsichtbar und entspricht der Laserklasse 1, wodurch keine Gefahr für das menschliche Auge besteht.

Ein wesentlicher Vorteil des VL53L0X liegt in seiner kompakten Bauform: Mit Abmessungen von etwa $12 \times 25 \times 2$ Millimetern lässt er sich platzsparend in den begrenzten Modellbauraum integrieren (siehe S. 15 Bild 2), beispielsweise hinter Scheinwerferattrappen oder in schmalen Frontspoilern. Durch Kommunikation über den I²C-Bus lassen sich alle vier Sensoren am Arduino mit nur zwei gemeinsamen Leitungen verbinden. Diese Buskommunikation entspricht dem Prinzip moderner Fahrzeugnetzwerke, bei denen Systeme wie CAN (Controller Area Network) oder LIN (Local Interconnect Network) den Verkabelungsaufwand durch gemeinsam genutzte Datenleitungen minimieren. Zudem arbeiten ToF-Sensoren unabhängig von der Oberflächenfarbe oder -struktur des Zielobjekts, was eine zuverlässige Hinderniserkennung auch bei dunklen oder matten Oberflächen ermöglicht.

Werksseitig besitzen alle VL53L0X-Sensoren dieselbe I²C-Adresse, weshalb ein paralleler Betrieb am gemeinsamen Bus zunächst ausgeschlossen ist. Über den XSHUT-Eingang lässt sich jedoch jeder Sensor einzeln in einen Reset-Zustand versetzen, sodass eine sequenzielle Adressänderung möglich ist. Anwendungsschrift AN4846 von STMicroelectronics beschreibt das hierzu notwendige Vorgehen: Zunächst werden alle Sensoren über XSHUT zurückgesetzt, anschließend wird einer der Sensoren aktiviert und dessen Standardadresse softwareseitig auf einen abweichenden Wert umgestellt. Dieser Vorgang wird für jeden weiteren Sensor wiederholt, sodass nach Abschluss der Initialisierung alle vier Module über eindeutige Adressen (im vorliegenden Aufbau 0x30 bis 0x33) am gemeinsamen Bus erreichbar sind (vgl. STMicroelectronics 2016b). Da die geänderten Adressen nicht persistent im Sensor gespeichert werden, muss die Initialisierungsroutine bei jedem Systemstart erneut durchlaufen werden.

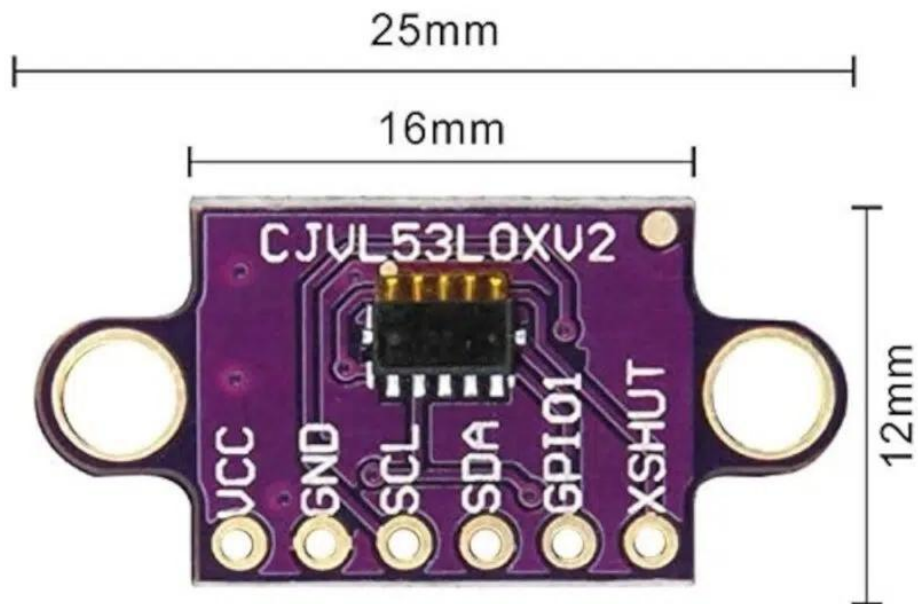


Bild 2: ToF-Sensor VL53L0X. Quelle: hwlibre.com

2.3.3 Schaltungsaufbau und Bedienung

Die Energieversorgung des Modellfahrzeugs basiert auf den Prinzipien moderner Elektrofahrzeuge und verbindet funktionale Anforderungen mit didaktischen Überlegungen. Als Energiespeicher dient eine kompakte Powerbank, die im Fahrzeug integriert ist und über ein kurzes USB-Kabel mit dem Arduino-Mikrocontroller verbunden ist. Bei der internen Anordnung müssen die Studierenden eine typische Package-Aufgabe lösen: Es ist darauf zu achten, dass der Energiespeicher so positioniert wird, dass der Schwerpunkt günstig liegt und gleichzeitig ausreichend Raum für die übrigen Komponenten bleibt. Bei der Auswahl der Powerbank ist es wichtig, dass sie ausreichend Strom liefert, um einen stabilen Betrieb zu gewährleisten.

Für das Laden der Powerbank sowie die Programmierung des Mikrocontrollers wird das Modell geöffnet, sodass beide Komponenten direkt zugänglich sind. Eine Trennung zwischen externem Ladeanschluss und internem Programmierzugang wurde konzeptionell erwogen, jedoch im Rahmen des vorliegenden Prototyps zugunsten einer einfacheren Konstruktion zurückgestellt. Die Integration eines solchen Ports könnte in einer Weiterentwicklung des Modells umgesetzt werden und stellt eine zusätzliche Package-Aufgabe für die Studierenden dar, da Zugänglichkeit und gestalterische Integration gegeneinander abgewogen werden müssten. Über einen Startknopf wird der Fahrbetrieb initiiert, der an einer gut erreichbaren Stelle der Karosserie positioniert werden sollte. Nach dem Betätigen startet das Fahrzeug seinen autonomen Betrieb mit einem frei festzulegenden Zeitlimit für den Durchlauf.

Diese zeitliche Begrenzung dient sowohl der Schonung der Komponenten als auch der praktischen Durchführung im Unterricht, da mehrere Gruppen nacheinander ihre Modelle testen können. Auch die Positionierung des Startknopfs stellt eine Package-Entscheidung dar, bei der Zugänglichkeit und Schutz vor versehentlicher Betätigung gegeneinander abgewogen werden müssen. Die vollständigen Verbindungen aller elektronischen Komponenten sind in Bild 3 dargestellt. Die vier ToF-Sensoren und das Display teilen sich den I²C-Bus an den Pins A4 (SDA) und A5 (SCL), wobei die einzelnen Sensoren zusätzlich über vier digitale XSHUT-Leitungen voneinander getrennt initialisiert werden. Über weitere vier Digitalpins werden die Servomotoren per PWM angesteuert. Der Taster ist gegen GND geschaltet und nutzt den internen Pull-up-Widerstand des Mikrocontrollers. Einer einheitlichen Konvention folgend, sind die Farben der Leitungen wie folgt vergeben: Rot kennzeichnet die Spannungsversorgung, Schwarz die Masse, Blau die Datenleitung, Orange die Taktleitung des I²C-Busses, Grün die XSHUT-Kabel, Gelb die PWM-Verbindungen der Servos und Violett die Tasterleitung.

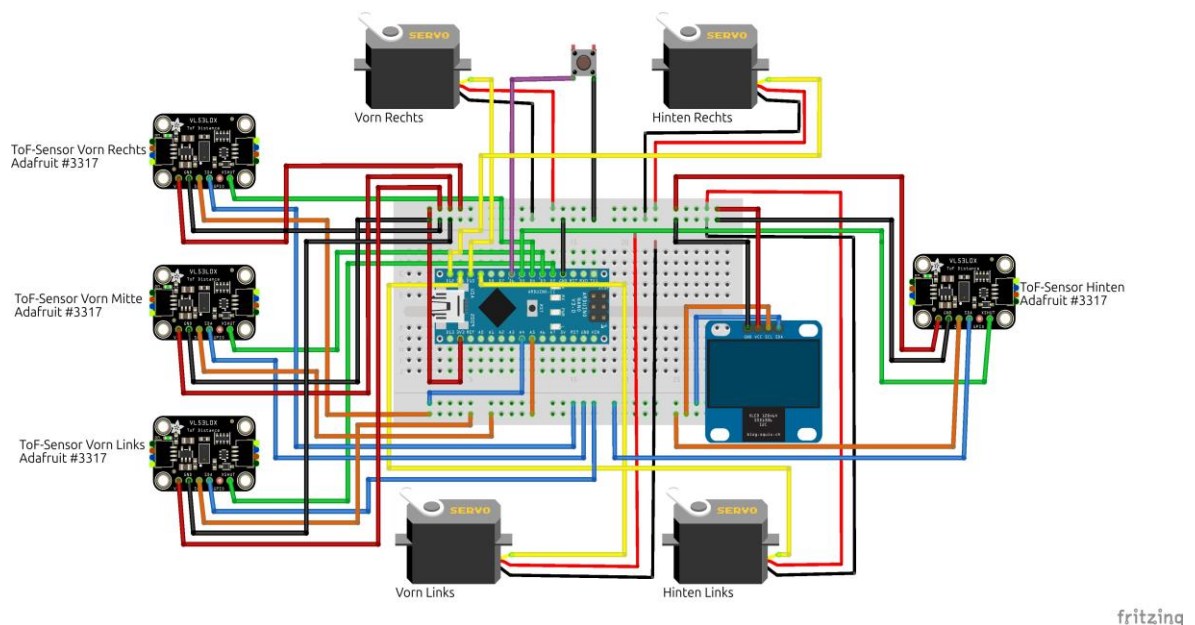


Bild 3: Schaltplan des Aufbaus

2.4 Didaktische Grundlagen

Die Vermittlung technischer Inhalte in der Hochschullehre steht vor einer grundlegenden Herausforderung: Theoretisches Wissen allein führt selten zu nachhaltigem Verständnis, während rein praktische Übungen ohne konzeptionelle Einbettung oberflächlich bleiben. Ein vielversprechender Ansatz zur Überwindung dieser Diskrepanz ist das problembasierte Lernen, bei dem Studierende nicht mit fertigen Lösungen konfrontiert werden, sondern sich Wissen durch die aktive Auseinandersetzung mit authentischen Problemstellungen erschließen. Das folgende Kapitel legt die bildungstheoretischen und didaktischen Grundlagen dar, auf denen das vorliegende Lehrmodul aufbaut.

2.4.1 Problemorientierter Unterricht

Klafki liefert mit seinem Ansatz zum problemorientierten Unterricht eine Orientierungsgrundlage für den didaktischen Ansatz des Lehrmoduls. Dessen bildungstheoretisches Konzept fordert, dass Schule und Unterricht auf sogenannte epochaltypische Schlüsselprobleme konzentriert werden (Klafki 2007, S. 56 ff.). Mit diesem Begriff bezeichnet Klafki die zentralen Herausforderungen einer historischen Epoche, an deren Bewältigung sich Bildung orientieren sollte. Anstatt abstrakte Lerninhalte zu vermitteln, soll Unterricht Lernende befähigen, die drängenden Probleme ihrer Zeit zu verstehen und an deren Lösung mitzuwirken. Beispiele solcher Schlüsselprobleme sind in der Gegenwart etwa die Digitalisierung, der Klimawandel und die nachhaltige Mobilität.

Voraussetzung für einen gelingenden problemorientierten Unterricht ist, dass die Aufgabe aus einem realen Anwendungsfeld stammt und nicht für Lehrzwecke konstruiert wurde. Eine künstliche Problemstellung würde lediglich vorangegangene Lerninhalte nachträglich rechtfertigen, anstatt selbst Ausgangspunkt der inhaltlichen Auseinandersetzung zu sein. Erst durch die Authentizität der Aufgabe wird das Problem zum eigentlichen Kern des Unterrichts.

Übertragen auf das vorliegende Lehrmodul bedeutet dies, dass die Herausforderung der Bauraumintegration eines autonomen Fahrzeugs nicht künstlich für den Unterricht geschaffen wurde, sondern einer realen Problemstellung der Automobilentwicklung entspricht, sodass Studierende eine authentische Aufgabe bearbeiten und der Wissenstransfer zwischen Theorie und Praxis gefördert wird.

Im Sinne dieser Definition stellt die Integration von Sensorsystemen, Steuergerät und Energiespeicher in begrenzte Bauräume ein epochaltypisches

Schlüsselproblem der modernen Fahrzeugentwicklung dar, das durch die zunehmende Elektrifizierung und Automatisierung weiter an Bedeutung gewinnt. Diese Problemstellung verbindet mehrere Fachkompetenzen miteinander, sodass ein fächerverbindender Unterrichtsansatz besonders geeignet erscheint.

Klafkis Konzept der kategorialen Bildung verbindet zwei traditionell gegensätzliche Bildungstheorien zu einem integrativen Ansatz (vgl. Klafki 2007, S. 96). Die materiale Bildungstheorie betont die Aneignung von Wissen und Inhalten, während die formale Bildungstheorie die Entwicklung von Fähigkeiten und Denkweisen in den Vordergrund stellt. Er überwindet diesen Gegensatz mit dem Konzept der doppelseitigen Erschließung: Das Allgemeine erschließt sich durch das Exemplarische, und das Exemplarische durch das Allgemeine. Für das vorliegende Lehrmodul bedeutet dies, dass das konkrete Package-Problem des Modellfahrzeugs nicht als Selbstzweck behandelt wird, sondern als exemplarischer Zugang zu den allgemeinen Prinzipien des Fahrzeug-Packages.

Im modernen Kompetenzkonzept werden kategoriale Bildungsprozesse durch den Erwerb domänenspezifischer Kompetenzen beschrieben, wobei Klieme et al. Kompetenzen als kontextspezifische Leistungsdispositionen definieren, die sich auf bestimmte Domänen beziehen (Klieme et al. 2003, S. 72). Das vorliegende Lehrmodul fördert sowohl kategoriale als auch interkategoriale Kompetenzen, wobei die Verknüpfung beider Kompetenzarten durch die problemorientierte Struktur begünstigt wird. Somit ist der bildungstheoretische Rahmen abgesteckt; für die konkrete Gestaltung des Lernprozesses bedarf es jedoch einer operativen Struktur, die im folgenden Abschnitt dargestellt wird.

2.4.2 Drei-Phasen-Modell

Klafkis Konzept des problemorientierten Unterrichts liefert eine bildungstheoretische Begründung dafür, dass Lernprozesse an authentischen Problemstellungen ausgerichtet werden sollten. Es beschreibt jedoch keine konkrete Ablaufstruktur, die den Lernprozess von der Problemkonfrontation bis zur Lösungsintegration durchläuft. Diese strukturgebende Funktion übernimmt das problembasierte Lernen (PBL), das in der hochschuldidaktischen Forschung als international verbreitetes Lehrkonzept gilt und sich nach Marx und Götze insbesondere für die Ingenieurwissenschaften eignet (vgl. Marx/Götze 2018, S. 189). Ausgangspunkt ist stets eine authentische, offen formulierte Problemstellung, die als Lernimpuls fungiert, kognitiv und emotional aktivierend wirkt und einen Bezug zur künftigen Berufspraxis herstellt (vgl. Marx/Götze 2018, S. 194).

Für die strukturelle Umsetzung des Lernprozesses wird der Drei-Phasen-Ansatz nach Wilhelm und Brovelli (2009) herangezogen. Dieser wurde an der Pädagogischen Hochschule Zentralschweiz in Luzern als Weiterentwicklung des verbreiteten Siebensprungmodells entwickelt, da dessen kleingliedrige Struktur die Lernenden zum sequenziellen Abarbeiten von Teilschritten führte, statt eine kritische Gesamtsicht auf das Problem zu fördern (vgl. Wilhelm/Brovelli 2009, S. 199). Der Drei-Phasen-Ansatz gliedert den Lernprozess stattdessen in drei zeitlich und inhaltlich klar voneinander abgrenzbare Phasen, die intern jedoch iterativ durchlaufen werden.

In der Analysephase wird das Problem zunächst identifiziert und aus verschiedenen Perspektiven betrachtet, um daraus konkrete Lernfragen abzuleiten. Rückschlaufen sind dabei möglich, da sich authentische Problemstellungen selten linear erschließen lassen (vgl. Wilhelm/Brovelli 2009, S. 199f.). Die Verstehensphase dient der Wissensaneignung durch Experimente, Literaturstudium oder gezielte Instruktion. Unterschiedliche Vorwissensstände und Informationsquellen führen zu verschiedenen Sichtweisen, die innerhalb der Gruppe abgeglichen werden müssen. In der Synthesephase werden die erarbeiteten Kenntnisse zu zwei bis drei realistischen Erklärungsszenarien zusammengeführt und gegeneinander abgewogen. Hierdurch wird das anfängliche Problemfeld neu betrachtet und die Übertragbarkeit der gewonnenen Erkenntnisse auf andere Fälle herausgearbeitet (vgl. Wilhelm/Brovelli 2009, S. 200).

Wilhelm und Brovelli verstehen PBL nicht als eine der vielen didaktischen Methoden, sondern als ein Grundelement des Lernens bei komplexen Themen (vgl. Wilhelm/Brovelli 2009, S. 200). Die konkrete Umsetzung dieses Ansatzes in der vorliegenden Arbeit wird in Abschnitt 3.2.2 dargestellt.

3 Konzeption des Lehrmoduls

3.1 Anforderungsanalyse

Ausgangspunkt für die Konzeption des Lehrmoduls ist eine Analyse der Anforderungen, die sich aus dem hochschuldidaktischen Kontext (vgl. Abschnitt 2.4), der Zielgruppe sowie den verfügbaren Ressourcen ergeben. Es können weder vertiefte CAD-Kenntnisse noch Programmiererfahrung vorausgesetzt werden; die Aufgabenstellung muss so gestaltet sein, dass der Einstieg niedrigschwellig möglich ist, ohne den inhaltlichen Anspruch zu mindern.

Studierende sollen elektronische und mechanische Komponenten in begrenztem Raum nach funktionalen, ergonomischen und fertigungstechnischen Anforderungen anordnen, eine passende Karosserie entwerfen, Konstruktionsentscheidungen umsetzen, Package-Konflikte erkennen und begründet dokumentieren können. Dieses Ziel lässt sich in drei Teilkompetenzen gliedern: die methodische Kompetenz im Umgang mit parametrischen CAD-Werkzeugen und der Freiformflächenmodellierung, die fachliche Kompetenz zur Beurteilung von Package-Kennzahlen sowie die kommunikative Kompetenz zur Darstellung und Verteidigung des eigenen Lösungsansatzes.

Aus organisatorischer Sicht ist das Modul auf die Bearbeitung in Kleingruppen ausgelegt, wobei der Zeitrahmen auf den Umfang einer semesterbegleitend zu bearbeitenden Aufgabe ausgerichtet ist. Als technische Infrastruktur wird empfohlen, den Studierenden eine parametrische CAD-Umgebung, FDM-fähige 3D-Drucker sowie das Arduino-Mikrocontrollersystem bereitzustellen. Das vorliegende Lehrmodul wurde für die Verwendung mit CATIA 3DEXPERIENCE (im Folgenden: 3DX) entwickelt; die zugrunde liegenden Prinzipien parametrischer Modellierung und integrierter Bewertungsfunktionen sind grundsätzlich auf vergleichbare CAD-Systeme übertragbar, wenngleich die konkrete Bewertungslogik dort je nach Funktionsumfang neu aufgebaut werden müsste. Sämtliche bereitgestellten Materialien sind so gestaltet, dass sie innerhalb der genannten Infrastruktur ohne zusätzliche Lizenzen oder Werkzeuge nutzbar bleiben.

Daraus ergibt sich für das Lehrmodul die Anforderung, eine vollständige und in sich geschlossene Lernumgebung bereitzustellen, die neben der Aufgabenstellung auch die notwendigen Hilfsmittel zur eigenständigen Bearbeitung umfasst. Hierzu zählen ein parametrisches CAD-Grundmodell mit integrierten Bewertungsfunktionen, Packagedaten der zu verbauenden Komponenten, ein Leitfaden zur KI-gestützten

Programmierung des Mikrocontrollers sowie Dokumentationsvorlagen, die die iterative Bearbeitung begleiten. Auf diese Weise kann der Fokus der Studierenden auf die konstruktive Auseinandersetzung mit dem Package gerichtet werden, während die programmiertechnische Umsetzung durch geeignete Hilfsmittel unterstützt wird.

3.2 Didaktischer Bezug

Das entwickelte Lehrmodul setzt die in Abschnitt 2.4 dargestellten didaktischen Prinzipien in einer konkreten Lernumgebung um, wobei die Konstruktion eines autonom fahrenden Modellfahrzeugs als didaktisches Mittel zur Vermittlung von Kompetenzen im Bereich Package dient.

3.2.1 Problemorientierung und Authentizität

Im Fokus steht die zentrale Aufgabenstellung, die die Integration von Sensoren, Steuereinheit und Energiespeicher in einen begrenzten Bauraum erfordert, bei gleichzeitiger Optimierung der Schwerpunktlage und der Sensorabdeckung. Damit greift das Modul eine authentische Problemstellung der Fahrzeugentwicklung auf, die im Sinne des in Abschnitt 2.4.1 dargestellten Ansatzes nach Klafki den Unterricht konstituiert. Auftretende Zielkonflikte zwischen konkurrierenden Anforderungen, etwa zwischen optimaler Schwerpunktlage und guter Zugänglichkeit der Komponenten, entsprechen den typischen Abwägungen in der industriellen Package-Entwicklung.

3.2.2 Umsetzung des Drei-Phasen-Modells

Strukturell folgt das Lehrmodul dem in Abschnitt 2.4.2 beschriebenen Drei-Phasen-Ansatz nach Wilhelm und Brovelli (2009). Den Auftakt bildet die Analysephase, in der die Studierenden mit der Aufgabenstellung konfrontiert werden und die Vorgaben zu den verbauten Komponenten sowie den Maßstab 1:10 erhalten, während Radstand und Spurweite als freie Gestaltungsparameter festgelegt werden. Die scheinbare Einfachheit der Aufgabe weicht dabei schnell der Erkenntnis, dass typische Zielkonflikte wie die Konkurrenz zwischen günstiger Schwerpunktlage und freien Sensorsichtfeldern keine triviale Lösung zulassen. Aus dieser Auseinandersetzung leiten die Studierenden eigene Lernfragen ab, etwa zur Sensorpositionierung oder zur Masseverteilung.

In der Verstehensphase erarbeiten die Gruppen Teillösungen in den Bereichen CAD-Konstruktion, Bauraumplanung und Mikrocontrollerprogrammierung. Mithilfe

der parametrischen Modellierung in 3DX werden nicht nur Werkzeugkompetenzen vermittelt, sondern auch die Abhängigkeiten zwischen Konstruktionsparametern und Fahrzeugeigenschaften unmittelbar erfahrbar gemacht. Der KI-gestützte Programmierleitfaden (siehe Kapitel 5.2) setzt den Fokus auf die konstruktive Auseinandersetzung mit dem Package, sodass keine vertieften Programmierkenntnisse vorausgesetzt werden.

In der Synthesephase werden die Teillösungen zu einem funktionsfähigen Gesamtsystem integriert. Im praktischen Test konfrontiert das Fahrverhalten die Studierenden unmittelbar mit den Auswirkungen ihrer Konstruktionsentscheidungen: Ein ungünstig platzierter Schwerpunkt führt zu Instabilität in Kurven, verdeckte Sensoren verhindern die Hinderniserkennung. Diese Erfahrbarkeit entspricht im Sinne von Wilhelm und Brovelli der geforderten Exemplarität, wodurch sich die am Modell erarbeiteten Package-Kenntnisse auf reale Fahrzeugentwicklungsprozesse übertragen lassen.

3.2.3 Kompetenzerwerb

Sowohl fachspezifische als auch übergreifende Kompetenzen werden im Lehrmodul angesprochen. Auf fachlicher Ebene erwerben die Studierenden Kenntnisse in der parametrischen CAD-Modellierung, der Package-Analyse sowie der Bewertung von Konstruktionslösungen anhand definierter Kriterien, wobei die Arbeit mit dem Konstruktionsskelett und den daraus abgeleiteten Parametern ein Verständnis für die methodische Vorgehensweise in der Fahrzeugentwicklung vermittelt. Darüber hinaus fördert das Modul überfachliche Kompetenzen wie systematisches Problemlösen, die Fähigkeit zur Abwägung konkurrierender Anforderungen sowie die Dokumentation und Präsentation von Entwicklungsergebnissen. Die Gruppenarbeit schult zudem die Kommunikations- und Teamfähigkeit.

3.3 Aufgabenstellung für Studierende

Gegenstand der Begleitaufgabe ist die Entwicklung eines autonom fahrenden Modellfahrzeugs im Maßstab 1:10 auf Basis eines vorgegebenen parametrischen Grundgestells. Ausgehend von den festgelegten Fahrzeugparametern Radstand und Spurweite sind die bereitgestellten Komponenten, bestehend aus Mikrocontroller, vier Abstandssensoren, vier Servomotoren, einem Energiespeicher, einem Breadboard und einem Startknopf, so im verfügbaren Bauraum anzuordnen, dass ein funktionsfähiges Gesamtsystem entsteht. Für die

Inbetriebnahme und Diagnose steht ergänzend ein Display zur Verfügung, das die Sensorwerte in Echtzeit anzeigt und die Fehlersuche erleichtert, jedoch nicht zwingend in die finale Karosserie integriert werden muss. Jede Komponente ist mit einem individuellen Halter und Befestigungselementen zu versehen, die in die vorgesehenen Aufnahmen der Grundplatte eingepasst werden. Auf dieser Grundlage ist eine Karosserie zu entwerfen, die die konstruktiven Entscheidungen der Gruppe in konstruktiver und funktionaler Hinsicht umfasst.

Die Aufgabe gliedert sich in drei aufeinander aufbauende Arbeitsschritte. Im ersten Schritt analysieren die Studierenden den verfügbaren Bauraum und entwickeln ein Konzept zur Komponentenanzordnung, wobei typischen Package-Zielkonflikten zwischen günstiger Schwerpunktlage, freien Sensorsichtfeldern und ausreichender Zugänglichkeit für Wartung und Programmierung erkannt und begründet abgewogen werden müssen. Im zweiten Schritt wird das erarbeitete Konzept in der CAD-Umgebung 3DX umgesetzt. Hierfür steht das parametrische Grundgestell mit integrierten Bewertungsfunktionen als Vorlage zur Verfügung, sodass die Studierenden ihre Lösung anhand der definierten Key Performance Indicators (KPIs) jederzeit selbstständig prüfen und iterativ verbessern können. Zuletzt wird das fertige Modell gedruckt, aufgebaut und auf einem Parcours erprobt. Mithilfe eines begleitenden KI-Leitfadens erfolgt die Programmierung des Mikrocontrollers, sodass keine vertieften Programmierkenntnisse vorausgesetzt werden.

Die Bearbeitung erfolgt in Gruppen von zwei bis drei Personen und ist auf einen Zeitraum von zehn Wochen ausgelegt, wobei die Aufgabe bewusst zu Beginn des Semesters parallel zur theoretischen Lehrveranstaltung angesetzt werden sollte. Auf diese Weise können Inhalte aus der Vorlesung unmittelbar in die praktische Arbeit einfließen, während den Studierenden gegen Ende des Semesters ausreichend Zeit für weitere Prüfungsleistungen bleibt. Abzugeben sind das vollständige CAD-Modell, eine Dokumentation des Entwicklungsprozesses einschließlich der getroffenen Package-Entscheidungen sowie eine abschließende Kurzpräsentation der Lösung. Im folgenden Abschnitt werden die Bewertungskriterien erläutert, die für diese Projektaufgabe von besonderer Relevanz sind.

3.4 Bewertungskriterien und KPIs

Die Bewertung der studentischen Arbeiten erfolgt anhand quantifizierbarer Kennzahlen, sogenannter KPIs, die eine objektive Beurteilung der Package-Lösung ermöglichen. Zentrales Ziel des Bewertungskonzepts ist die Möglichkeit zur Selbstprüfung während der Konstruktionsphase, sodass Studierende iterativ Verbesserungen vornehmen können, bevor die finale Abgabe erfolgt. Hierfür werden die Bewertungskriterien direkt in die CAD-Arbeitsdatei integriert, deren technische Umsetzung in Abschnitt 4.2 beschrieben wird.

Schwerpunktlage

Die Lage des Fahrzeugschwerpunkts beeinflusst das Fahrverhalten in mehrfacher Hinsicht. In Längsrichtung bestimmt die Schwerpunktlage die Achslastverteilung, also das Verhältnis der auf Vorder- und Hinterachse wirkenden Gewichtskräfte. Eine ausgeglichene Achslastverteilung verbessert die Fahrstabilität und reduziert die Neigung zum Über- oder Untersteuern. Für die Bewertung werden drei Stufen definiert: Eine Hinterachslast zwischen 48 und 52 % gilt als optimal, zwischen 45 und 55 % als gut und zwischen 40 und 60 % als akzeptabel. In Querrichtung wird die Abweichung des Schwerpunkts von der Fahrzeuglängsachse bewertet, da eine ungleichmäßige Seitenverteilung das Kurvenverhalten beeinträchtigt. Bezugsgröße ist hierbei die Spurweite, sodass die Toleranzgrenzen automatisch mit der gewählten Fahrzeuggeometrie skalieren. Durch ihren vertikalen Wert beeinflusst die Schwerpunkthöhe sowohl die Wankneigung bei Kurvenfahrten als auch die Kippstabilität. Als Bewertungsreferenz dient die Oberkante der Reifen: Ein Schwerpunkt auf oder unterhalb dieser Höhe gilt als ideal.

Die Bewertung erfolgt auf Basis der Schwerpunktkoordinaten, die Studierende nach der Ermittlung mittels Measure Inertia in die Bewertungssteuerung eintragen.

Sichtfeldanalyse der Sensoren

Von besonderer Relevanz für die Funktionsfähigkeit des autonomen Fahrzeugs ist die ungehinderte Erfassung der Umgebung durch die Sensoren. Jeder der vier Messkegel wird als Volumenkörper direkt im CAD-Modell des jeweiligen Sensors hinterlegt und bewegt sich automatisch mit dessen Position. Mittels einer vollautomatischen Kollisionsprüfung zwischen den Messkegeln und den übrigen Fahrzeugkomponenten wird geprüft, ob Karosserieteile das Sichtfeld eines Sensors einschränken. Jede erkannte Kollision signalisiert eine potenzielle Beeinträchtigung der Hinderniserkennung. Ziel ist eine kollisionsfreie Positionierung aller Sensoren,

wobei die drei Frontsensoren gemeinsam eine lückenlose Abdeckung des vorderen Erfassungsbereichs gewährleisten.

Sensorabdeckung

Neben der ungehinderten Ausrichtung der einzelnen Sensoren ist die gemeinsame Flächenabdeckung der Frontsensoren ein eigenständiges Bewertungskriterium. Dabei werden zwei voneinander unabhängige Größen bewertet. Der Gesamtabdeckungswinkel beschreibt den Winkelbereich, den die Frontsensoren gemeinsam erfassen. Er wird automatisch aus der Geometrie der Messkegel berechnet und passt sich jeder Sensorausrichtung an. Eine Überlappungsquote beschreibt ergänzend, zu welchem Anteil sich die Messkegel der Sensoren überschneiden. Während ein gewisser Überlappungsgrad eine redundante Abdeckung und damit eine robustere Hinderniserkennung gewährleistet, deutet eine übermäßige Überschneidung auf eine suboptimale Ausnutzung des verfügbaren Sichtfelds hin, wohingegen eine Anordnung ohne Überlappung auf Lücken im Erfassungsbereich hinweist.

Bauraumeffizienz

Die Bauraumeffizienz η beschreibt das Verhältnis zwischen dem für Komponenten genutzten Volumen und dem insgesamt verfügbaren Gesamtvolumen. Hohe Effizienz deutet auf eine kompakte Anordnung hin, während niedrige Werte auf ungenutzten Bauraum hindeuten. Berechnet wird er nach Formel (2):

$$\eta = V_{Bauteile} / V_{Innenraum} \quad (2)$$

Dabei bezeichnet $V_{Innenraum}$ das nutzbare Volumen des Karosserieinnenraums und $V_{Bauteile}$ das Gesamtvolumen aller Komponenten innerhalb der Baugruppe. Für die Bewertung sind drei Stufen vorgesehen: Ein Nutzungsgrad von mindestens 35 % gilt als effizient, Werte zwischen 15 und 35 % als ausreichend und darunter liegende Werte als ineffizient. Als mögliche Erweiterung in einer Weiterentwicklung wird anstelle des Gesamtinnenraumvolumens ein separates Greenhouse-Volumen als Bezugsgröße eingesetzt, um die Bauraumnutzung auf den eigentlichen Fahrgastraum zu beschränken.

Kippstabilität

Als ergänzender fahrdynamischer Kennwert wird der Rollover-Index (Static Stability Factor) berechnet, der die Kippstabilität des Fahrzeugs gegen seitliches Umkippen unter Querbeschleunigung beschreibt. Der Index entspricht dem dimensionslosen Verhältnis der halben Spurweite zur Schwerpunkthöhe über der Fahrbahn und gibt damit die maximale Querbeschleunigung in g an, die das Fahrzeug ohne Kippgefahr

aufnehmen kann. Ein hoher Indexwert steht für einen tief liegenden und breit abgestützten Schwerpunkt, der eine geringe Kipptendenz gegenüber seitlichen Kräften bedeutet. Da der Rollover-Index ausschließlich auf den bereits für die Schwerpunktlage ermittelten Koordinaten sowie dem Masterparameter Spurweite beruht, wird er ohne zusätzliche Messungen aus den vorhandenen Modelldaten abgeleitet. Bewertet wird er in vier Stufen: Werte größer als 1,5 gelten als sehr stabil, Werte zwischen 1,2 und 1,5 als stabil, Werte zwischen 1,0 und 1,2 als grenzwertig und Werte unter 1,0 als erhöhte Kippgefahr.

Fertigungsgerechte Gestaltung

Sämtliche Bauteile des Modellfahrzeugs werden im FDM-Verfahren gefertigt. Als Druckmaterial kommt Polylactid (PLA) zum Einsatz, das sich aufgrund seiner niedrigen Drucktemperatur und guten Maßhaltigkeit als Standardmaterial für das FDM-Verfahren etabliert hat. Die fertigungsgerechte Gestaltung ist ein eigenständiges Bewertungskriterium, das jedoch, anders als die übrigen KPIs, nicht automatisiert ausgewertet werden kann, sondern eine manuelle Prüfung durch die Studierenden erfordert. Vor der Übergabe an den 3D-Drucker sind alle selbstkonstruierten Bauteile mit der Funktion „Measure Thickness“ in 3DX auf ausreichende Wandstärken zu überprüfen. Verrundungen erfordern besondere Aufmerksamkeit, da sie die verbleibende Wandstärke entlang des Krümmungsverlaufs lokal deutlich unter die Mindestwandstärke senken können, ohne dass dies in der Modellansicht unmittelbar erkennbar ist. Unterschreiten Bereiche der Geometrie die für das FDM-Verfahren empfohlene Mindestwandstärke von 1 mm, sind diese konstruktiv anzupassen, um Druckfehler zu vermeiden. Bild 4 auf Seite 27 verdeutlicht exemplarisch, welche Folgen eine unzureichende Wandstärke in der Praxis haben kann.

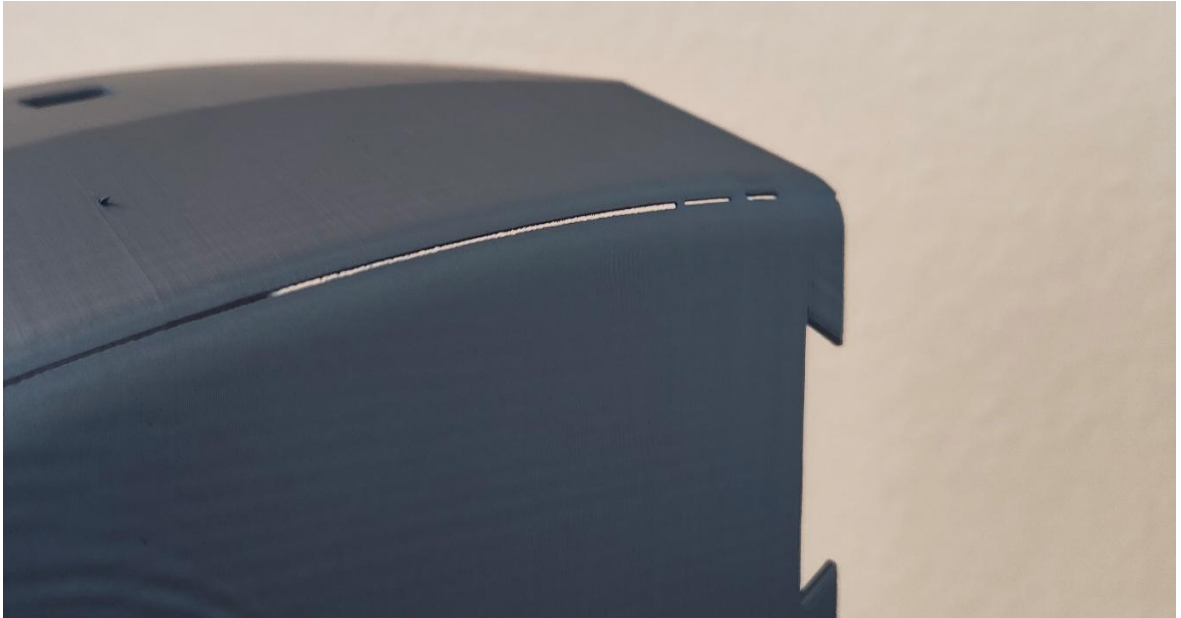


Bild 4: Karosserieteil mit zu geringer Wandstärke

Tabelle 1 (S. 28) fasst die Bewertungsstufen aller KPIs zusammen. Die Schwellenwerte wurden in Abstimmung mit dem Zweitbetreuer festgelegt (Nagel 2026, persönliche Mitteilung).

Tabelle 1: Bewertungsstufen und Schwellwerte der KPIs

Kriterium	Bewertungsstufe	Bereich
Achslastverteilung (Hinterachsanteil)	Optimal	48–52 %
	Gut	45–55 %
	Akzeptabel	40–60 %
Querlage des Schwerpunkts	Optimal	≤ 2 % der Spurweite
	Gut	≤ 5 % der Spurweite
	Akzeptabel	≤ 10 % der Spurweite
	Kritisch	> 10 % der Spurweite
Schwerpunkthöhe	Ideal	$\leq$ Oberkante Reifen
Rollover-Index	Sehr stabil	$> 1,5$
	Stabil	1,2–1,5
	Grenzwertig	1,0–1,2
	Erhöhte Kippgefahr	$< 1,0$
Sichtfeldanalyse	Bestanden	Keine Kollision Messkegel $\leftrightarrow$ Karosserie
	Nicht bestanden	Kollision vorhanden
Bauraumeffizienz η	Effizient	≥ 35 %
	Ausreichend	15–35 %
	Ineffizient	< 15 %
Mindestwandstärke (FDM)	Druckbar	$\geq 1,0$ mm

4 Aufbau und Erprobung des Modellfahrzeugs

4.1 Parametrische Konstruktion

Ausgangspunkt der Konstruktion ist ein zentrales Skelettpart, in dem alle maßgeblichen Fahrzeugparameter als benannte Variablen hinterlegt sind. Beide primären Treiber sind Radstand und Spurweite, aus denen sämtliche abhängigen Maße mittels Formeln abgeleitet werden. Hierunter fallen die Länge und Breite der Grundplatte sowie die Positionen der Servohalter und der Räder. Die Basisgeometrie orientiert sich am Fahrzeugkoordinatensystem, das auch in der industriellen Fahrzeugkonstruktion verwendet wird. Auf der Vorderachse in der Fahrzeugmitte liegt der Ursprung, die X-Achse zeigt nach hinten, die Y-Achse nach rechts und die Z-Achse nach oben (vgl. DIN ISO 4130 1979, Bild 5). Alle nachgelagerten Bauteile verweisen auf dieses Koordinatensystem, sodass eine Änderung der Masterparameter die gesamte Baugruppe automatisch aktualisiert.

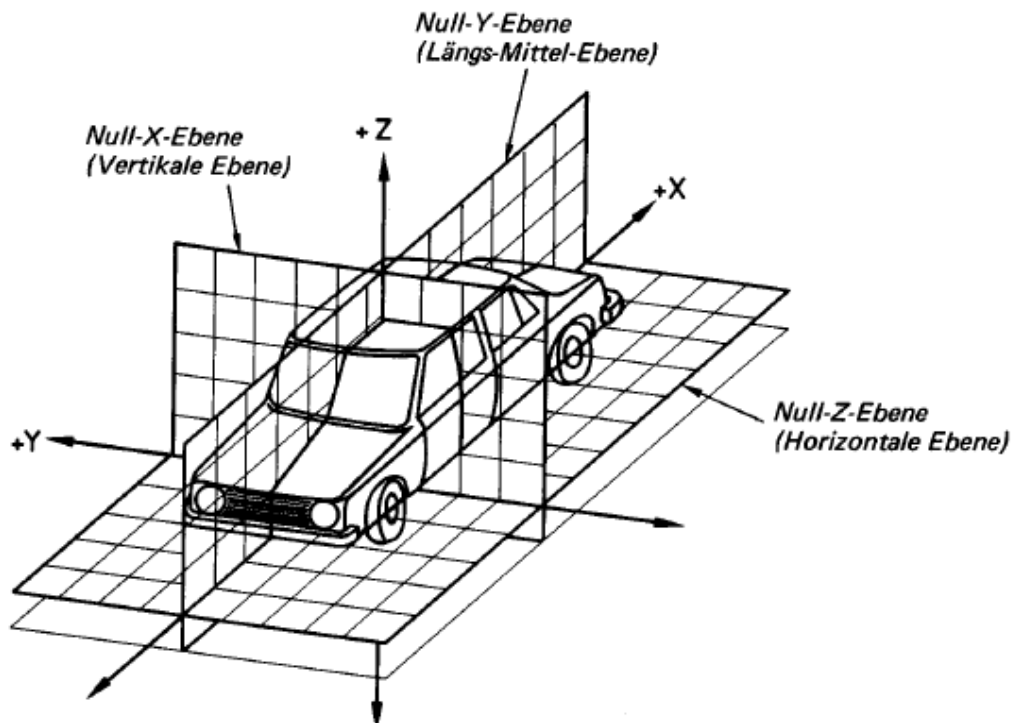


Bild 5: Fahrzeugkoordinatensystem nach DIN ISO 4130 (1979)

Bild 6 zeigt den fertigen Aufbau ohne Karosserie als Übersicht; die folgenden Abschnitte beschreiben die einzelnen konstruktiven Bestandteile im Detail.

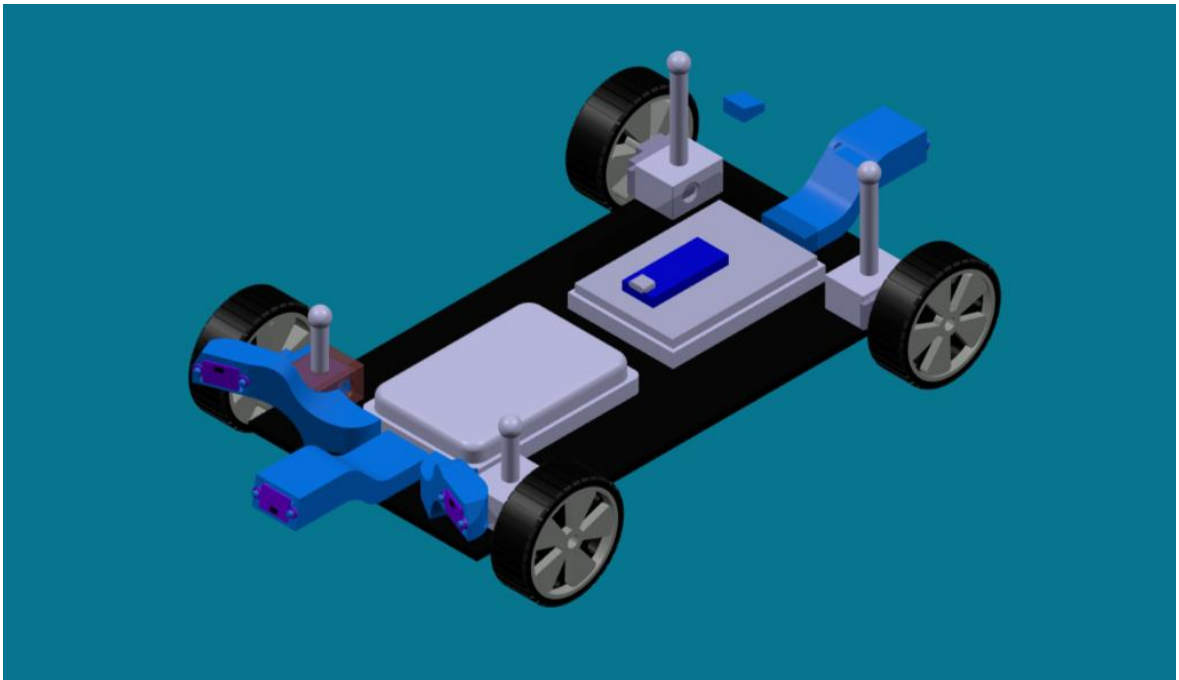


Bild 6: Gesamtansicht ohne Karosserie

Die Grundplatte ist als vierteiliges System ausgeführt, wobei die Segmente im FDM-Verfahren gedruckt und anschließend zusammengefügt werden. Fertigungsbedingt ist diese Aufteilung notwendig, da die Gesamtabmessungen die Druckfläche handelsüblicher FDM-Drucker überschreiten würden. Mithilfe von Schwalbenschwanz-Einsätzen werden die vier Segmente an der Unterseite miteinander verbunden (siehe Bild 7). Mittels einer entsprechenden Gestaltung der Bauteilgeometrie wird der Anteil an notwendigen Stützstrukturen auf ein Minimum reduziert, was den Nachbearbeitungsaufwand begrenzt.

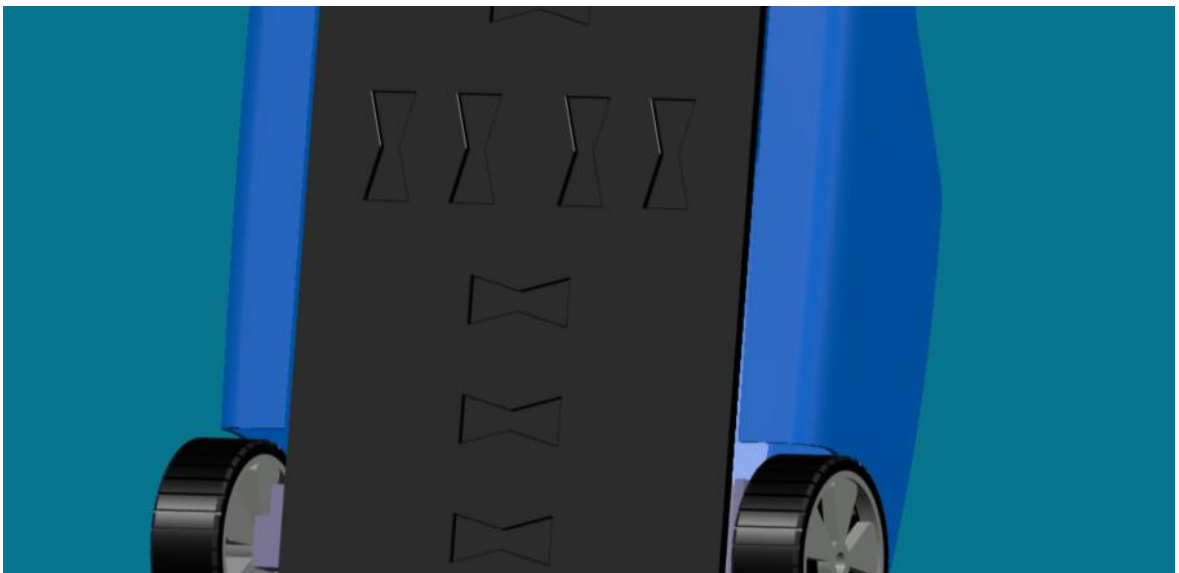


Bild 7: Unterseite der Grundplatte mit Schwalbenschwanz-Aufnahmen

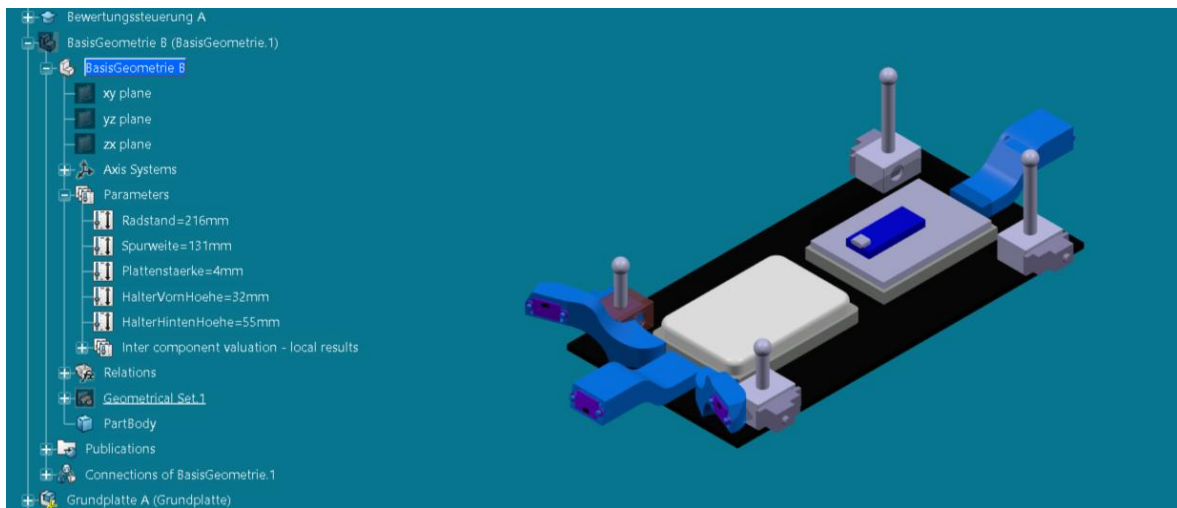


Bild 8: Grundplatte, Radstand 216 mm, Spurweite 131 mm

Bilder 8 und 9 zeigen die Grundplatte in zwei unterschiedlichen Parameterkonfigurationen. Unterschiede zwischen den beiden Varianten zeigen sich ausschließlich in den Werten der Masterparameter Radstand und Spurweite. Die abhängige Geometrie skaliert automatisch gemäß den hinterlegten Formeln.

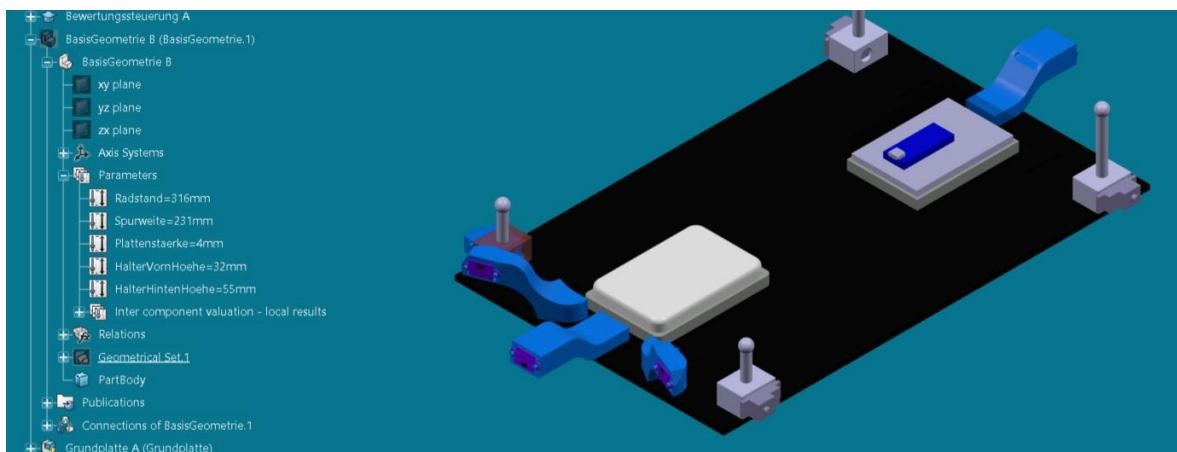


Bild 9: Grundplatte, Radstand 316 mm, Spurweite 231 mm

Für die Befestigung der restlichen Bauteile an der Grundplatte wurde ein zweistufiges Montagesystem entwickelt. Rechteckige Positionierungstaschen dienen zunächst der lagerichtigen Ausrichtung der Halter beim Klebeprozess. Die rechteckige Form wurde gewählt, da sie im Gegensatz zu kreisförmigen Aufnahmen nicht nur die Position, sondern auch die Winkelerorientierung des Halters festlegt. Mit einer definierten Toleranz von 0,2 mm ermöglichen die Taschen ein sicheres Einsetzen der Halter, ohne unerwünschtes Spiel in der Endposition zu verursachen. Für die unterschiedlich ausgerichteten Sensorhalter werden die Taschen über ein rotiertes lokales Achssystem im jeweiligen Part erzeugt, wobei der Ausrichtungswinkel als Parameter hinterlegt werden kann.

Auf der Grundplatte werden die Positionen der Montagetaschen vollständig mithilfe von Formeln bestimmt und skalieren automatisch entsprechend den Masterparametern. Die Frontsensoren sind frei im Raum positioniert und werden jeweils mit ihren Haltern an die Grundplatte befestigt. Breadboard und Powerbank sind in Längsrichtung ausgerichtet und in Querrichtung symmetrisch zur Fahrzeuglängsachse angeordnet, um eine ausgewogene Massenverteilung zu gewährleisten.

Da 3DX-Bauteilmassen ausschließlich aus Volumen und Materialdichte berechnet werden und keine direkte Masseneingabe möglich ist, wurden für die Elektronikbauteile Pseudo-Materialien angelegt. Hierbei wird die Dichte eines Materials so gewählt, dass das Produkt aus dem modellierten Bauteilvolumen und der Pseudodichte der realen Bauteilmasse entspricht. Auf diese Weise fließen die tatsächlichen Massen aller Komponenten korrekt in die automatische Schwerpunktberechnung der Baugruppe ein. Ergänzend wurde ein Hilfsmaterial mit nahezu vernachlässigbarer Dichte angelegt und allen Referenzgeometrien, Hilfsbodies und Bewertungskörpern im Geometrical Set „BEWERTUNG_NICHT_AENDERN“ zugewiesen. Auf diese Weise tragen die Bewertungsgeometrien rechnerisch nicht zur Gesamtmasse bei und verfälschen die Schwerpunktberechnung nicht.

Jeder der vier Abstandssensoren wird von einer Halterung aufgenommen, die das Sensor-PCB über dessen Montagelöcher im benötigten Winkel zur Grundplatte positioniert. Ein Ausschnitt in der Halterung führt die Kabel geordnet zum Breadboard weiter. Der Neigungswinkel der Halterung zur Grundplatte wurde iterativ angepasst, um die optimale Kombination aus Nahbereichsabdeckung und Überschneidung der Messkegel zu erreichen. Die finale konstruktive Ausführung der Halterungen wird nach Abschluss der Erprobungsphase festgelegt.

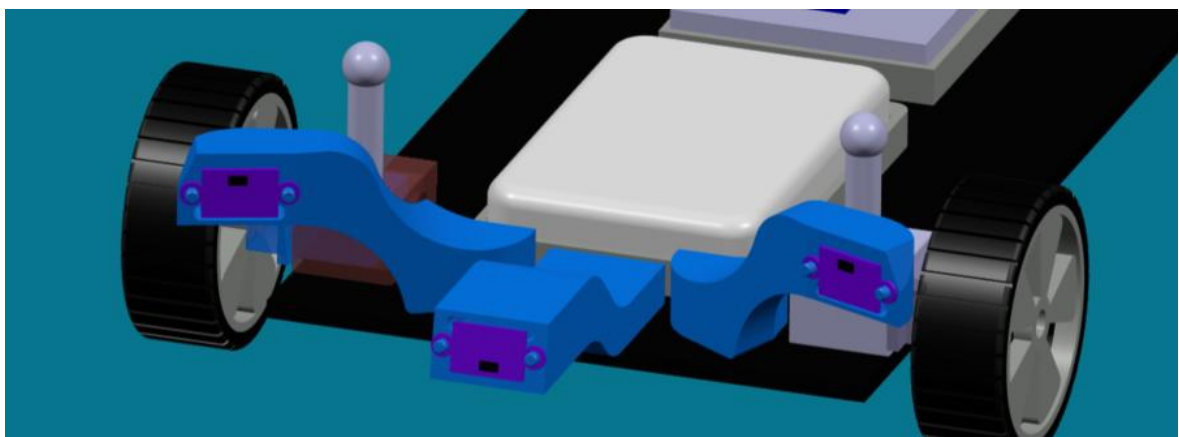


Bild 10: Sensorhalter mit PCB-Aufnahme

Bei der Anordnung der Frontsensoren wurde darauf geachtet, dass die beiden äußeren Halter zur Längsachse des Fahrzeugs ausgerichtet sind und der mittlere Sensor parallel dazu verläuft (siehe S. 32 Bild 10). Mit dieser Konfiguration überschneiden sich die Messkegel der Frontsensoren im relevanten Nahbereich, was eine lückenlose Abdeckung der vorderen Fahrzeugumgebung ermöglicht. Ein vierter Sensor ist heckseitig verbaut und sichert die Rückwärtsfahrt bei Ausweichmanövern ab.

Die Messkegel sind im CAD-Modell als eigene Geometrien innerhalb des jeweiligen Sensor-Parts modelliert und bewegen sich automatisch mit der Sensorposition, sodass die Sichtfeldanalyse mittels Kollisionsprüfung jederzeit ohne manuelle Aktualisierung zur Verfügung steht. Wie in Bild 11 verdeutlicht, überlappen sich die Messkegel der drei Frontsensoren im relevanten Nahbereich.

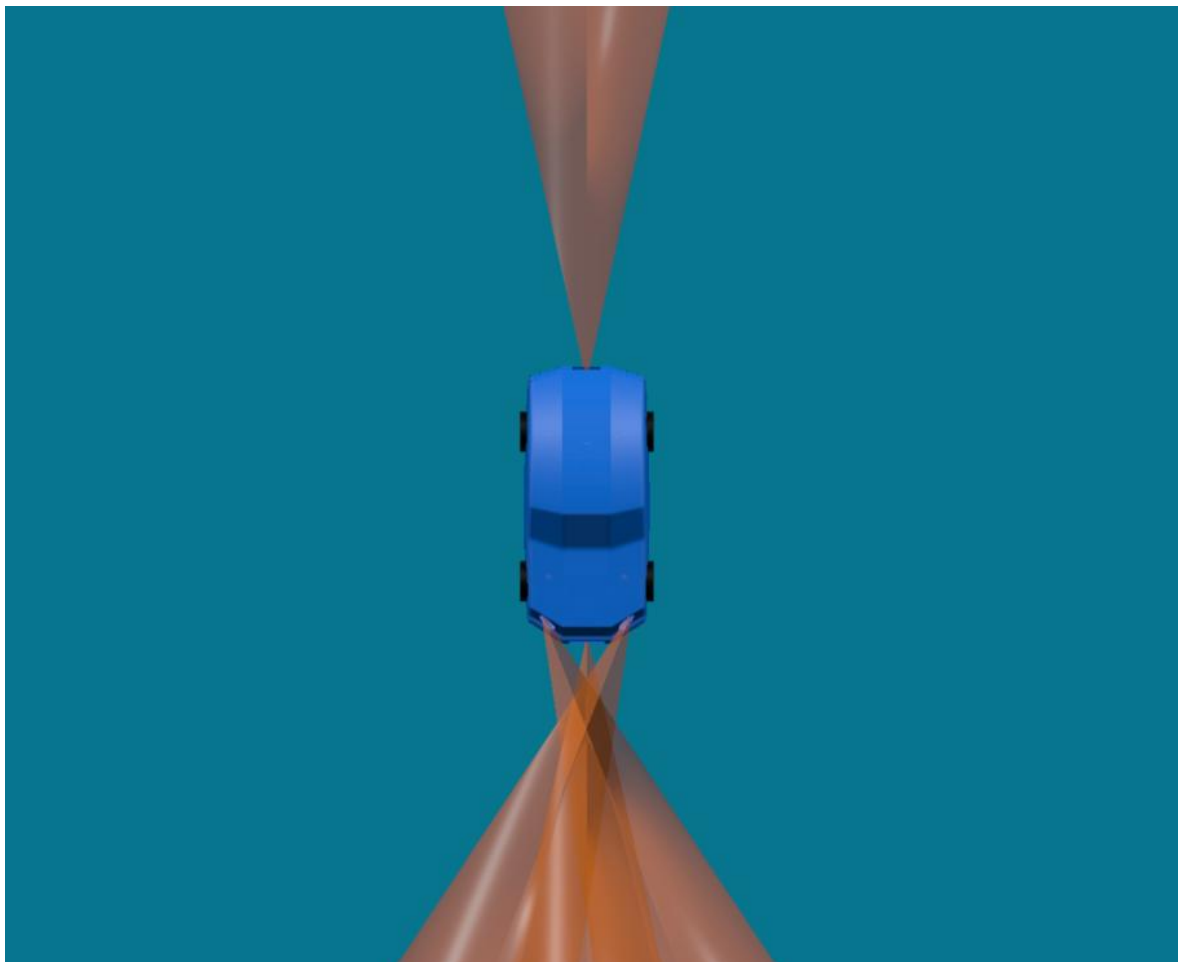


Bild 11: Draufsicht mit Messkegeln der ToF-Sensoren

Breadboard und Powerbank werden jeweils in einer wannenförmigen Halterung positioniert. Eine umlaufende Lippe greift über den Bauteilrand und fixiert die Komponente durch Eigengewicht in der Wanne, ohne dass zusätzliche Befestigungsmittel erforderlich sind. Somit ist eine werkzeuglose Entnahme möglich, was den Zugang für Wartung und Programmierung erleichtert. Die Wannengeometrie wird direkt aus den im Modell hinterlegten Bauteilmodellen abgeleitet, sodass ein Austausch gegen eine Komponente mit abweichenden Maßen durch Modellanpassung möglich ist. In Bild 12 ist ihre symmetrische Anordnung im Modell dargestellt.

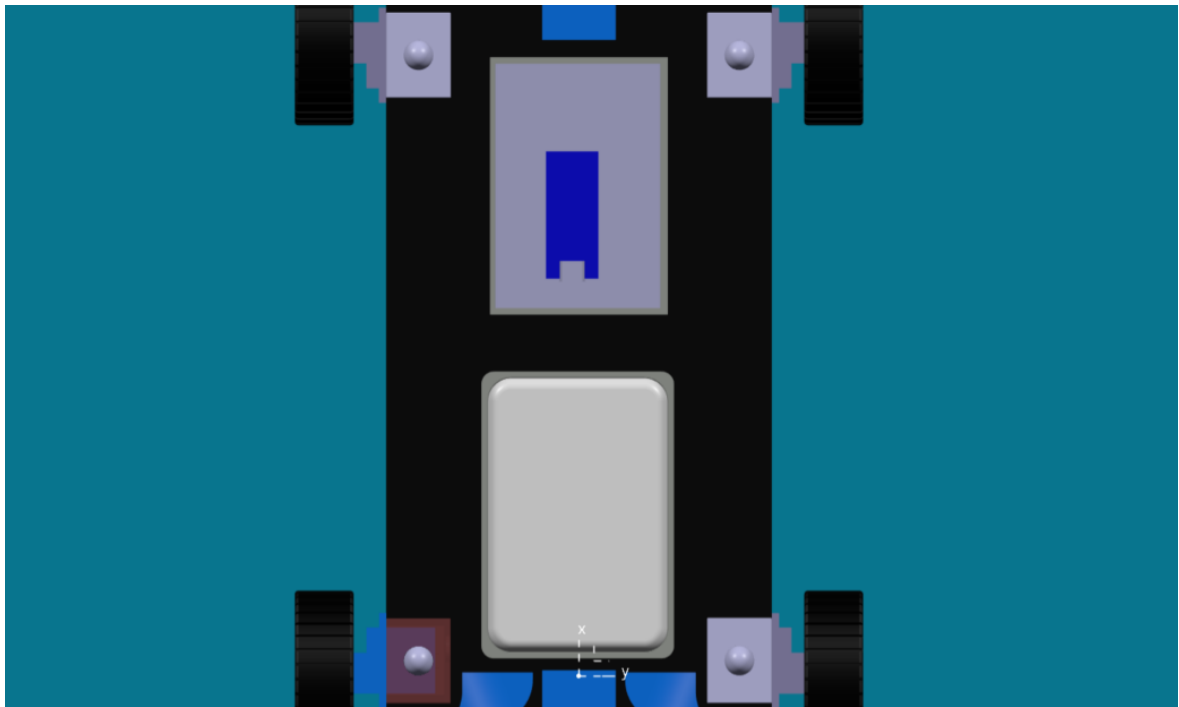


Bild 12: Wannenhalter für Breadboard und Powerbank

Die Bewertungselemente sind in einem separaten Geometrical Set mit der Bezeichnung „BEWERTUNG_NICHT_AENDERN“ organisiert, das initial ausgeblendet ist und bei Bedarf zur Selbstprüfung eingeblendet werden kann. Bild 13 zeigt den Aufbau mit eingeblendetem Bewertungs-Set, Bild 14 ergänzt diese Ansicht um die Karosserie. Aufbau und Funktionsweise der integrierten Bewertungslogik werden in Abschnitt 4.2 beschrieben.

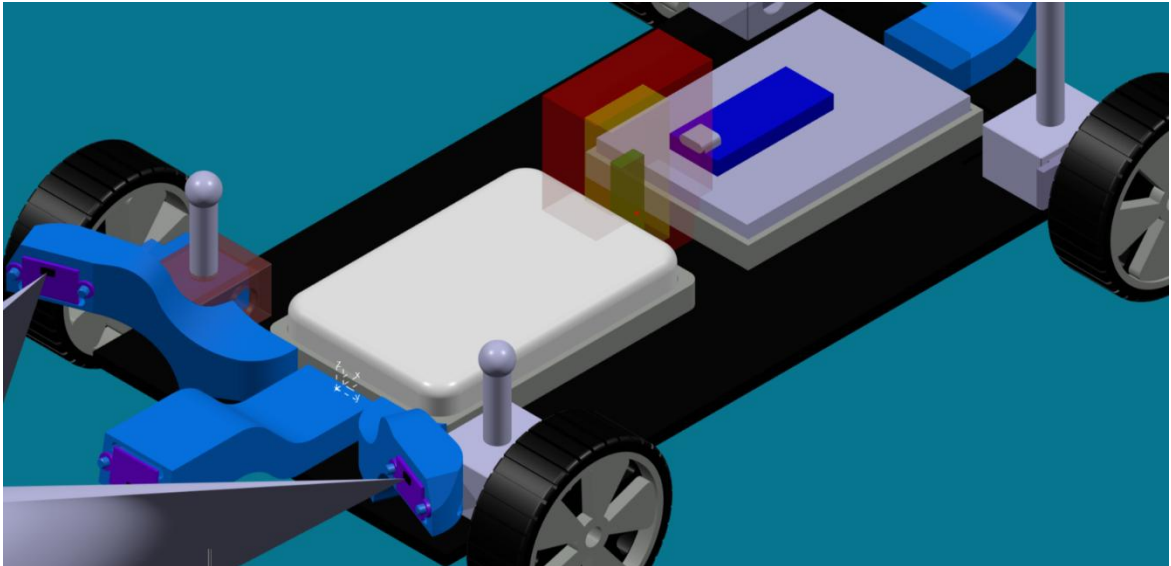


Bild 13: Aufbau mit eingeblendetem Bewertungs-Set

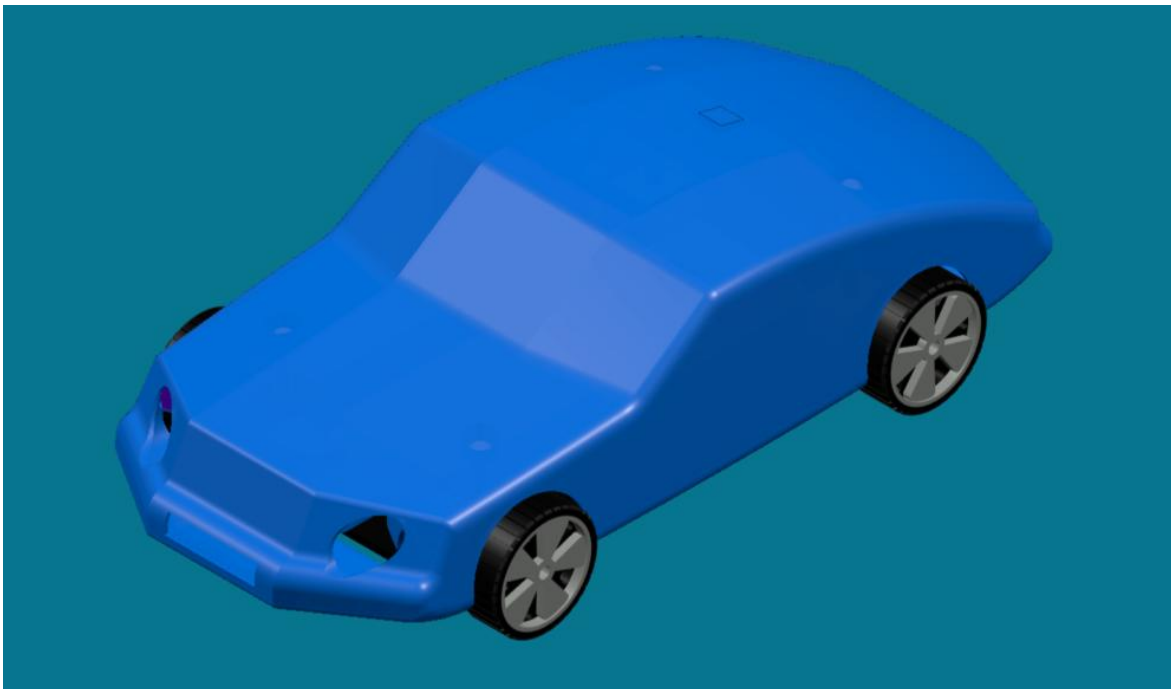


Bild 14: Gesamtansicht mit Karosserie

4.2 Integrierte Bewertungslogik

Die in Abschnitt 3.4 beschriebenen Bewertungskriterien werden mithilfe parametrischer Geometrieelemente und Regeln in der Engineering Knowledge Language (EKL) direkt in die CAD-Arbeitsdatei integriert. Eine Übersicht über die technische Umsetzung aller KPIs zeigt Tabelle 2. Die folgenden Abschnitte erläutern ausgewählte Designentscheidungen, die über die rein mechanische Umsetzung hinausgehen. Da alle relevanten Eingangsgrößen aus der Modellgeometrie abgeleitet werden, aktualisiert sich die Bewertungslogik bei jedem Assembly-Update automatisch.

Tabelle 2: Technische Umsetzung der KPIs in der CAD-Datei

KPI	Eingangsgrößen	EKL-Mechanismus	Output-Parameter
Achslastverteilung	COG_X, Radstand	Formel + Regel mit If-Bedingung	String (Stufe) + parametrische Quader-Bodies (optimal/gut/akzeptabel)
Querlage Schwerpunkt	COG_Y	Toleranzvergleich	String (Stufe)
Schwerpunkthöhe	COG_Z, Reifendurchmesser	Summenformel + Vergleich mit Reifenoberkante	String (Stufe)
Rollover-Index	COG_Z, Spurweite	Formel: $(\text{Spurweite} / 2) / \text{Schwerpunkthöhe}$	Numerischer Wert + Stufe
Sichtfeldanalyse	Messkegelvolumina, Karosserie-Referenzkörper	distance()-Funktion	String („KOLLISION“ / „KEINE KOLLISION“)
Gesamtabdeckungswinkel	Extrempunkte an Kegelhändern, Koordinatenursprung	Winkelberechnung zwischen Punkten	String + numerischer Wert in Grad
Überlappungsquote	Einzelne Kegelvolumina, smartVolume-Vereinigung	Differenzformel: $\sum V_{\text{einzel}} - V_{\text{vereinigt}}$	Numerischer Wert in Prozent

KPI	Eingangsgrößen	EKL-Mechanismus	Output-Parameter
Bauraumeffizienz	V_Bauteile (Σ Komponentenvolumina), V_Innenraum (Hälftenreferenz $\times 2$)	Verhältnisformel $\eta = V_Bauteile / V_Innenraum$	Numerischer Wert + Stufe
Gesamtauswertung	alle obigen Outputs	zentrale Regel mit Schalter-Parameter	Informationsfenster, gegliedert nach Kategorien

Schwerpunktbewertung und Kippstabilität

Die Schwerpunktkoordinaten ergeben sich aus der Massenanalyse des Gesamtmodells und werden von den Studierenden händisch in die Eingabeparameter COG_X, COG_Y und COG_Z (Center of Gravity) übernommen, aus denen die Bewertung berechnet wird. Zur räumlichen Veranschaulichung sind im Geometrical Set drei parametrische Quaderkörper hinterlegt, deren Abmessungen über Formeln mit Radstand und Spurweite verknüpft sind und so die Zonen automatisch nachskalieren: optimal, gut und akzeptabel. Der Rollover-Index ergibt sich ohne zusätzliche Eingaben aus denselben Koordinaten in Kombination mit der Spurweite.

Sichtfeldanalyse

Eine Kollisionsprüfung zwischen Sensor-Messkegeln und Karosserie nutzt die EKL-Funktion distance(), die den kleinsten Abstand zwischen dem Messkegel-Volumen und dem Karosserie-Referenzkörper ermittelt; ist der Wert null, wird der Bewertungsparameter auf „KOLLISION“ gesetzt. Gegenüber einer booleschen Überschneidungsoperation hat dieser Ansatz den Vorteil, auch im kollisionsfreien Fall einen definierten Rückgabewert zu liefern, während eine Schnittoperation hier mangels Schnittkörper einen Fehlerzustand auslösen würde.

Sensorabdeckung

Gesamtabdeckungswinkel und Überlappungsquote leiten sich unmittelbar aus der Messkegelgeometrie ab (vgl. Tabelle 2) und aktualisieren sich daher ohne zusätzliche Eingaben bei jeder Änderung der Sensorausrichtung. Entscheidend ist dabei die smartVolume-Operation zur Bestimmung des Vereinigungsvolumens, da sie das tatsächlich überdeckte Volumen ohne Doppelzählung der Überschneidungsbereiche liefert.

Bauraumnutzung

Das Innenraumvolumen wird über die Funktion Measure Inertia auf einem Referenzvolumen, das aus der Innenfläche der Karosserie abgeleitet wird, bestimmt. Im Modell liegt lediglich eine Hälftenreferenz der Karosserie vor, daher wird der gemessene Wert mit dem Faktor 2 auf das Gesamtvolumen hochgerechnet.

Gesamtauswertung

Eine zentrale EKL-Regel führt die Ergebnisse aller KPIs zusammen und wird über einen booleschen Schalterparameter aktiviert. Bei Aktivierung öffnet sich ein Informationsfenster mit den Bewertungsergebnissen, gegliedert in die Kategorien Schwerpunktlage, Sensorik und Bauraumnutzung (siehe Bild 15). Auf diese Weise steht eine kompakte Übersicht über die Konstruktionsgüte zur Verfügung, ohne die einzelnen Parameterwerte ablesen zu müssen.

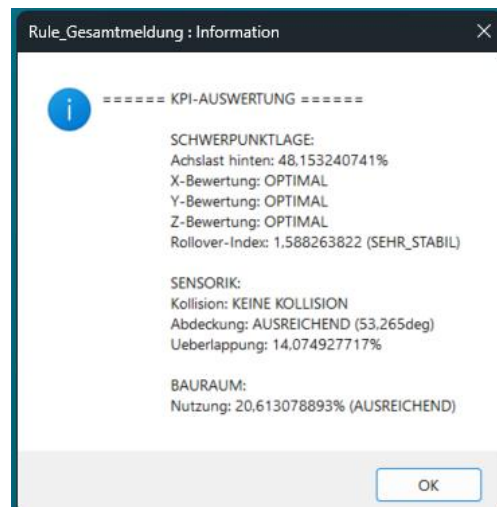


Bild 15: Informationsfenster der zentralen KPI-Auswertung

4.3 Auswahl der Komponenten

Die Auswahl der verbauten Komponenten folgt drei übergeordneten Kriterien: dem Potenzial, realistische Package-Zielkonflikte zu erzeugen; der Handhabbarkeit für Studierende ohne vertiefte Elektronikkenntnisse sowie der Verfügbarkeit und den Anschaffungskosten. Aufgabe aller ausgewählten Komponenten besteht darin, zum einen realistische Integrationsprobleme zu schaffen und zum anderen mit geringem Einarbeitungsaufwand in Betrieb genommen werden zu können.

Als Mikrocontroller kommt der Arduino Nano zum Einsatz. Seine kompakte Bauform von 45 × 18 mm, die breite Verfügbarkeit sowie die umfangreiche Bibliotheksunterstützung machen ihn zur geeigneten Plattform für den Einsatz an

Hochschulen. Gegenüber leistungsfähigeren Alternativen wie dem ESP32 wurde der Arduino Nano bewusst gewählt, da die reduzierte Komplexität den Fokus auf die konstruktive Aufgabenstellung lenkt und die KI-gestützte Codegenerierung mit dieser Plattform zuverlässig funktioniert.

Für die Abstands- und Hinderniserkennung werden vier VL53L0X-Sensoren eingesetzt. In der vorangegangenen Studienarbeit (Köhler 2025) kamen noch Ultraschallsensoren zum Einsatz; der Wechsel auf ToF-Sensoren erfolgte auf Basis der in Abschnitt 2.3.2 beschriebenen Eigenschaften: Die kompakte Bauform erleichtert die Integration in den begrenzten Modellbauraum innerhalb der Karosserie, die I²C-Schnittstelle reduziert den Verkabelungsaufwand erheblich, und der 25-Grad-Messkegel ermöglicht eine gezielte Sichtfeldgestaltung, die im CAD-Modell direkt bewertet werden kann. Die Anzahl von vier Sensoren ist dabei fest vorgegeben, um die automatische Auswertung der Sichtfeldanalyse ohne manuelle Anpassung der Bewertungslogik zu gewährleisten.

Als Antriebskomponenten werden vier Servomotoren des Typs FS90R eingesetzt, die auf der Grundplatte in seitlicher Ausrichtung montiert sind und als Nabenmotoren jeweils ein Rad direkt antreiben. Dabei handelt es sich um Servomotoren mit kontinuierlicher Rotation, deren Drehzahl und Richtung über das PWM-Signal des Mikrocontrollers gesteuert werden. Diese Motorenart wurde gewählt, weil sie eine direkte Radmontage ohne zusätzliche Getriebe erlaubt und somit den konstruktiven Aufwand für die Begleitaufgabe auf ein sinnvolles Maß reduziert.

Ergänzt wird der Hardwareaufbau durch drei weitere Komponenten. Ein OLED-Display mit einer Auflösung von 128 × 64 Pixeln dient während der Inbetriebnahme zur Echtzeitanzeige der Sensorwerte und erleichtert die Fehlersuche; da es ebenfalls über den I²C-Bus angebunden ist, verursacht es keinen zusätzlichen Verkabelungsaufwand und muss nicht zwingend in die finale Karosserie integriert werden. Ein Breadboard übernimmt die Verdrahtung sämtlicher Komponenten und stellt die gemeinsame Spannungs- und Masseverteilung bereit, wodurch eine werkzeuglose Anpassung der Schaltung möglich ist. Ein Taster fungiert als Startknopf für den Fahrbetrieb und ist gegen Masse geschaltet; seine Positionierung an der Karosserie ist eine eigenständige Package-Entscheidung, bei der Zugänglichkeit und Schutz vor versehentlicher Betätigung gegeneinander abzuwägen sind.

Für Lehrende, die das Modul eigenständig durchführen, sind bei der Komponentenauswahl einige praktische Gesichtspunkte zu beachten. Hinsichtlich der Energieversorgung sollte die gewählte Powerbank eine ausreichende Stromabgabe von mindestens 2 A gewährleisten, da die vier Servomotoren unter Last gemeinsam einen Strom von mehreren hundert Milliampere benötigen und eine unterdimensionierte Stromquelle zu einem instabilen Betrieb führen kann. Darüber hinaus empfiehlt sich bei der Auswahl aller verbauten Komponenten eine möglichst einfache äußere Geometrie, da die Studierenden für jede Komponente individuelle Halter konstruieren. Bauteile mit komplexen Freiformflächen oder unregelmäßigen Außenkonturen erhöhen den konstruktiven Aufwand unverhältnismäßig und lenken den Fokus von der eigentlichen Package-Aufgabe ab. Sollten einzelne Komponenten aufgrund ihrer Geometrie nicht ohne Weiteres von Studierenden eingefasst werden können, wäre es denkbar, entsprechende Halter vorzufertigen und als Teil der Vorlage bereitzustellen. Dies steht jedoch im Widerspruch zur Grundidee der Aufgabe, bei der die eigenständige konstruktive Auseinandersetzung mit jeder Komponente einen wesentlichen Lernanteil darstellt, und sollte daher auf Ausnahmefälle beschränkt bleiben.

5 Lehrmaterialien

5.1 CAD-Daten und Vorlagen

Den Studierenden wird ein vorbereitetes Datenpaket zur Verfügung gestellt, welches die zentralen Ausgangsbestandteile der konstruktiven Bearbeitung umfasst. Dieses Paket umfasst eine schriftliche Aufgabenstellung, eine CAD-Grunddatei in 3DX und eine bebilderte Anleitung zur Verwendung des Modells.

Die schriftliche Aufgabenstellung umfasst die zu erbringenden Konstruktionsleistungen und unterscheidet zwischen den vorgegebenen Komponenten und den frei zu gestaltenden Bestandteilen wie den Sensor- und Wannenhaltern oder der Karosserie (vollständige Aufgabenstellung siehe Anlage 1). Sie nennt zudem die Bewertungsgrundlage sowie die Modalitäten der Abgabe.

Die CAD-Grunddatei stellt den vorbereiteten Konstruktionsrahmen in 3DX bereit. Sie enthält das Skelett-Part mit den Masterparametern Radstand und Spurweite, die Grundplatte mit den vorpositionierten Motorhaltern, die Sensor-Parts mit den eingebetteten Messkegel-Bodies sowie das Set „BEWERTUNG_NICHT_AENDERN“. Die Materialien für die Elektronikkomponenten sind bereits in der Bibliothek hinterlegt und können den jeweiligen Komponenten zugewiesen werden, sodass die Schwerpunktberechnung von Beginn an valide Ergebnisse liefert. Studierende können die Konstruktionsaufgabe damit unmittelbar beginnen, ohne dass eigene Vorarbeiten an den Referenzelementen erforderlich sind (die zugehörige CAD-Darstellung befindet sich in Anlage 2).

Die bebilderte Anleitung führt durch die wesentlichen Arbeitsschritte beim Umgang mit der Grunddatei. Sie zeigt, wie eigene Bauteile mit den Referenzen des Bewertungssets verknüpft werden, wie die Bewertungselemente zur Selbstprüfung eingeblendet und wieder ausgeblendet werden und welche Bereiche der Datei nicht verändert werden dürfen. Ergänzend enthält diese Handlungsempfehlungen für typische Konstruktionssituationen, etwa zur Positionierung der Sensorhalter oder zur Anpassung der Karosserieaußenhaut.

5.2 KI-Leitfaden und Anschlussdokumentation

Die Programmierung des Mikrocontrollers stellt für Studierende ohne Vorerfahrung in der Softwareentwicklung eine potenzielle Hürde dar, die vom eigentlichen Lernziel des Lehrmoduls ablenken kann. In der vorangegangenen Studienarbeit wurde jedoch belegt, dass LLMs funktionsfähigen Code für den verwendeten Hardwareaufbau generieren können, sodass der Fokus der Studierenden auf die konstruktive Auseinandersetzung mit dem Package gerichtet ist (vgl. Köhler 2025). Der vorliegende Leitfaden beschreibt, wie diese Werkzeuge effektiv eingesetzt werden können, ohne dass tiefgreifende Programmierkenntnisse erforderlich sind. Er ist bewusst modellunabhängig formuliert und lässt sich mit gängigen KI-Assistenten wie Claude, ChatGPT oder vergleichbaren Systemen verwenden.

Einstiegsprompt

Der folgende Prompt hat sich als geeigneter Ausgangspunkt erwiesen und kann direkt in einen KI-Assistenten eingegeben werden. Die Pinbelegung entspricht dem getesteten Hardwareaufbau und muss bei Übernahme nicht angepasst werden:

„Hallo, ich habe ein Breadboard, einen Arduino Nano, ein OLED-Display 128x64, vier VL53L0X Time-of-Flight-Sensoren sowie vier FS90R-Servos. Ich möchte, dass daraus ein kleines Fahrzeug wird, das Sensoren zur Hinderniserkennung verwendet. Kannst du mir bitte dabei helfen?

Verwende bitte ausschließlich die Pololu VL53L0X-Bibliothek.

Angeschlossen habe ich alles bereits so, dass es funktionieren muss:

XSHUT Sensor vorne links: Pin D2

XSHUT Sensor vorne Mitte: Pin D3

XSHUT Sensor vorne rechts: Pin D4

XSHUT Sensor hinten: Pin D5

I²C-Adresse vorne links: 0x30

I²C-Adresse vorne Mitte: 0x31

I²C-Adresse vorne rechts: 0x32

I²C-Adresse hinten: 0x33

I²C-Adresse Display: 0x3C

Servo vorne links: Pin D7

Servo vorne rechts: Pin D8

Servo hinten links: Pin D9

Servo hinten rechts: Pin D10

Taster (Startknopf): Pin D6, gegen GND geschaltet mit internem Pull-up. Das Fahrzeug soll erst nach Tasterdruck losfahren.

SDA: A4

SCL: A5

5 V und Masse sind an allen Geräten über die Außenbahnen des Breadboards angeschlossen. Der Stoppwert aller Servos ist 90."

Bibliothekshinweis

Auf dem Arduino Nano ist der Arbeitsspeicher begrenzt. Die Adafruit VL53L0X-Bibliothek überschreitet dieses Limit bei der Verwendung aller vier Sensoren. Es muss daher, wie im Prompt bereits angegeben, explizit die Pololu VL53L0X-Bibliothek angefordert werden. Falls ein LLM dennoch Adafruit-Code vorschlägt, kann der Hinweis als separate Nachricht wiederholt werden: *„Bitte verwende ausschließlich die Pololu VL53L0X-Bibliothek, nicht die Adafruit-Variante.“*

Iteratives Vorgehen

Selten ist der erstgenerierte Code sofort einsatzbereit. Empfehlenswert ist ein schrittweises Vorgehen: Zunächst wird ein Grundgerüst generiert, das nur die Sensorauslesung und die serielle Ausgabe der Messwerte umfasst, jedoch ohne Fahrlogik. Sobald die Sensorwerte plausibel erscheinen, wird im nächsten Schritt die Fahrlogik ergänzt. Wenn das Fahrzeug sich nicht wie gewünscht verhält, sollte die Situation möglichst konkret beschrieben werden:

„Das Fahrzeug dreht sich nach links, obwohl das Hindernis rechts ist. Der linke Frontsensor ist an der Adresse 0x30, der mittlere an 0x31 und der rechte an 0x32. Wie muss die Ausweichlogik angepasst werden?“

Vage Beschreibungen wie „es funktioniert nicht“ führen selten zu hilfreichen Antworten. Je konkreter das beobachtete Verhalten beschrieben wird, desto gezielter kann das LLM helfen.

Was immer selbst geprüft werden muss

LLMs können plausibel klingenden, aber fehlerhaften Code erzeugen. Folgende Punkte sollten vor dem Aufspielen des Sketches immer manuell geprüft werden: Die Pinnummern im Code müssen mit der tatsächlichen Verkabelung übereinstimmen, da ein vertauschter Pin einen Servo dauerhaft drehen oder stoppen lassen kann. Die Servogeschwindigkeiten sollten vor dem ersten Test auf niedrige Werte nahe 90 eingestellt werden, um unkontrollierte Bewegungen zu vermeiden. Beim ersten Test empfiehlt es sich, das Fahrzeug anzuheben, damit die Räder frei drehen und kein Schaden entsteht, falls die Logik noch nicht fehlerfrei ist.

Anschlussdokumentation

Neben dem KI-Leitfaden ist eine vollständige Dokumentation des Hardwareaufbaus und des finalen Sketches für die Wartung und Weiterentwicklung des Modells unerlässlich. Studierende sollten am Ende ihrer Bearbeitung festhalten, welche Anpassungen gegenüber dem Einstiegsprompt vorgenommen wurden und welche Änderungen am generierten Code erfolgten. Diese Dokumentation erleichtert sowohl die Fehlersuche als auch die Übergabe an nachfolgende Gruppen oder Semester. Code-Anpassungen lassen sich im Konstruktionstagebuch (siehe Abschnitt 5.3) dokumentieren. Als Referenz ist der finale Sketch des Modellfahrzeugs in Anlage 3 dokumentiert.

5.3 Dokumentationsvorlagen

Begleitend zur konstruktiven Bearbeitung führen die Studierenden ein Konstruktionstagebuch. Dieses dokumentiert den Konstruktionsverlauf sowie Anpassungen am Hardwareaufbau und am Programmcode in digitaler Form und ermöglicht es den Dozierenden, den Gedankengang der Bearbeitenden nachzuvollziehen. Die Dokumentation dient gleichzeitig als Bewertungsgrundlage, da sie die Iterationsschritte zwischen den einzelnen KPI-Auswertungen sichtbar macht.

Im Konstruktionstagebuch werden die getroffenen Konstruktionsentscheidungen, die zugrunde liegenden Überlegungen sowie auftretende Schwierigkeiten und deren Lösungen festgehalten. Die Studierenden sind frei in der Wahl des Umfangs und der Detailtiefe der einzelnen Einträge. Der wesentliche Anspruch besteht darin, den Konstruktionsprozess für Dritte nachvollziehbar zu dokumentieren. Eine fortlaufende Pflege des Tagebuchs während der Bearbeitungsphase ist der rückblickenden Erstellung am Ende vorzuziehen, da die unmittelbare Dokumentation die Auseinandersetzung mit dem eigenen Vorgehen fördert. Eine Vorlage stellt Anlage 4 bereit.

Eine einfache Tabelle erfasst die Entwicklung der Bewertungskennzahlen im Verlauf des Konstruktionsprozesses. Mit jeder Konstruktionsiteration werden die zugehörigen Kennzahlen erfasst, sodass sich die Auswirkungen einzelner Entscheidungen unmittelbar auf die Gesamtbewertung ablesen lassen. Die Tabelle umfasst alle iterativ veränderlichen KPIs aus Abschnitt 3.4: Schwerpunktlage, Rollover-Index, Sichtfeldanalyse, Gesamtabdeckungswinkel, Überlappungsquote sowie Bauraumeffizienz. Die fertigungsgerechte Gestaltung wird erst unmittelbar

vor dem Druck geprüft und ist daher nicht Teil der Iterationstabelle. Ein freies Bemerkungsfeld ermöglicht zusätzlich, jede Iteration mit einem kurzen Hinweis zu versehen.

Ergänzend stellt Anlage 5 einen Bewertungsbogen bereit, der Dozierenden eine strukturierte Notenfindung anhand der definierten KPIs ermöglicht und den Studierenden Transparenz über die Bewertungssystematik verschafft.

Die fertige Dokumentation ist Bestandteil der Abgabe und wird gemeinsam mit der CAD-Datei im Anhang der studentischen Arbeit eingereicht.

6 Diskussion

Die Zielsetzung dieser Arbeit bestand in der Entwicklung einer praxisorientierten Begleitaufgabe für das Wahlmodul „Ergonomie und Package“, die ohne vertiefte CAD- oder Programmierkenntnisse zugänglich ist und reale Package-Konflikte erfahrbar macht. Das entwickelte Modul setzt diesen Anspruch durch eine mehrstufige Vorbereitung der Lernumgebung um. Das parametrische CAD-Grundmodell legt die Geometrie des Fahrzeugs bereits in Teilen fest, sodass die Studierenden ihre konstruktive Arbeit auf die Positionierung der Sensoren, der Powerbank und des Breadboards sowie auf die Konstruktion der zugehörigen Halterungen konzentrieren können. Den gestalterischen Abschluss bildet die Entwicklung einer Karosserie.

Die Kenntnisse für die Karosseriegestaltung im Bereich der Freiformflächenmodellierung sind umfangreicher als zunächst angenommen. Für die Umsetzung des Moduls an der Westsächsischen Hochschule Zwickau wäre daher eine zusätzliche Lehrveranstaltung zum Generative Shape Design in 3DX sinnvoll, die von den zuständigen Mitarbeitenden geleitet werden könnte. Alternativ können die Studierenden öffentlich zugängliche Onlineanleitungen nutzen.

Von den drei in Abschnitt 3.1 definierten Teilkompetenzen werden die methodische und die fachliche Kompetenz durch das Modul am stärksten gefördert. Die methodische Kompetenz ergibt sich aus dem unmittelbaren Umgang mit der parametrischen CAD-Umgebung und dem Mikrocontrollersystem, die fachliche aus der Auseinandersetzung mit den Package-Kennzahlen und deren Bewertung. Die kommunikative Kompetenz wird hingegen nur indirekt gefördert und ergibt sich vor allem aus der Gruppenarbeit sowie der abschließenden Kurzpräsentation, ohne dass hierfür eine eigene methodische Schulung vorgesehen ist.

Das Drei-Phasen-Modell nach Wilhelm und Brovelli (2009) hat sich auch im Verlauf der eigenen Bearbeitung als tragfähig erwiesen. Zunächst wurde der Umsetzungsumfang festgelegt, anschließend wurden mögliche Lösungswege recherchiert und schließlich zu einem funktionsfähigen Prototypen zusammengeführt. Die wiederholte Rückkehr in vorangegangene Phasen erwies sich dabei nicht als Bruch, sondern als notwendige Anpassung an den zunehmenden Erkenntnisstand.

Auch die Forderung nach problemorientiertem Unterricht im Sinne Klafkis lässt sich im Modul einlösen, da die Bauraumintegration eine reale Herausforderung der Fahrzeugentwicklung darstellt und lediglich im Maßstab reduziert sowie durch die

Bewertungslogik in eine prüfbare Form überführt wurde. Die verwendeten KPIs zur Schwerpunktlage, Bauraumeffizienz und Nahbereichsabdeckung sind aus realen Bewertungsmustern der Package-Entwicklung abgeleitet und für Studierende ohne Vorerfahrung vereinfacht; die wesentlichen Zielkonflikte des Originals bleiben dabei abgebildet.

Konzeptionell orientiert sich das entwickelte Modul an den industriellen parametrischen Packagemodellen, wie sie bei Kreis et al. (2021) und Stadler et al. (2013) beschrieben werden. Während diese Arbeiten den vollumfänglichen Einsatz in der Fahrzeugentwicklung beschreiben, überführt das vorliegende Modul die Prinzipien in ein vereinfachtes, didaktisch zugängliches Format. Die zentralen Elemente wie die Modellierung mit Masterparametern, integrierte Bewertungsfunktionen und assoziative Verknüpfung von Geometrie und Kennzahlen werden dabei beibehalten, jedoch auf ein Komplexitätsniveau reduziert, das sich im Rahmen einer semesterlangen Begleitaufgabe bearbeiten lässt.

Die Eignung des problembasierten Lernens für die Ingenieurwissenschaften, wie sie Marx und Götze (2018) beschreiben, entspricht den Erfahrungen aus dem eigenen Studienverlauf. Praxisorientierte Begleitaufgaben haben sich gegenüber rein theoretisch ausgerichteten Modulen als wirksamer bei der langfristigen Wissensverankerung erwiesen, wenngleich diese Beobachtung nicht systematisch erhoben wurde.

Über die unmittelbare Anwendung im Wahlmodul „Ergonomie und Package“ hinaus bietet die Grundstruktur des Moduls weitere Einsatzmöglichkeiten. Die Kombination aus vorbereitetem parametrischem CAD-Grundmodell und integrierter Bewertungsfunktion ist nicht spezifisch für die Package-Thematik, sondern lässt sich auf andere Bereiche der Fahrzeugkonstruktion übertragen. Module mit gesamtfahrzeugorientiertem Charakter, etwa aus der Fahrwerks- oder Antriebsstrangkonstruktion, sind naheliegende Anwendungsfelder. Das zugrunde liegende Prinzip der geführten Abstraktion realer Aufgaben unter Einbeziehung KI-gestützter Hilfsmittel ist nicht auf die Ingenieurwissenschaften beschränkt. Obwohl diese didaktische Grundidee nicht originär in der vorliegenden Arbeit entwickelt wurde, ergibt sich aus der konkreten Umsetzung ein übertragbares Konzept für vergleichbare Lehrformate.

Eine selbstkritische Anmerkung betrifft den parametrischen Bezug der Spurweite im eigenen Arbeitsmodell. Definitionsgemäß bezeichnet die Spurweite den Abstand

zwischen den Radmittelpunkten einer Achse. In der ursprünglichen Konstruktion wurden die zugehörigen Bezugsebenen jedoch auf den Rand der Grundplatte und nicht auf die Radmittelpunkte gelegt, sodass der gleichnamige Parameter im Modell streng genommen die Grundplattenbreite und nicht die tatsächliche Spurweite des Fahrzeugs steuert. In der für die Ausgabe an Studierende vorgesehenen Datei wurde dieser Bezug auf die Radmitten korrigiert. Im eigenen Modell ließ sich die nachträgliche Korrektur ohne Verlust der bereits ausgearbeiteten Halterungsgeometrie nicht umsetzen, da sich sämtliche parametrisch abhängigen Positionierungstaschen durch die Anpassung verschoben hätten.

Eine zentrale Einschränkung des entwickelten Moduls besteht in der Plattformbindung an 3DX. Die Übertragung der bereitgestellten CAD-Daten in ein anderes Programm ist mit erheblichem Aufwand verbunden. Die Erstellung einer geometrisch gleichwertigen Datei ist zwar grundsätzlich möglich, die Bewertungslogik muss jedoch entsprechend den Funktionsumfängen des jeweiligen Systems neu aufgebaut werden. Je nach Zielprogramm kann dies zu einer abweichenden Anzahl an KPIs oder zu anderen Bewertungen führen. Hinsichtlich der Berechnungsmöglichkeiten und der geometrischen Verknüpfungen bestehen darüber hinaus Grenzen, die vom Funktionsumfang von 3DX selbst vorgegeben sind.

Ergänzend ist festzuhalten, dass die Bearbeitung in 3DX im Verlauf der Arbeit mehrfach durch fehlerhafte Speichervorgänge erschwert wurde. Einzelne Konstruktionsstände waren nach dem erneuten Öffnen nicht mehr reproduzierbar, sodass Teilbereiche des Modells rekonstruiert werden mussten. Während sich dieser Aufwand im Rahmen der vorliegenden Arbeit durch die parametrische Modellstruktur in Grenzen halten ließ, kann ein vergleichbares Verhalten im Lehrbetrieb zu erheblichen Verzögerungen führen, weshalb regelmäßige Sicherungskopien außerhalb der Plattform empfehlenswert sind.

Eine weitere Einschränkung ergibt sich aus der erforderlichen technischen Infrastruktur. Während die Anschaffung der elektronischen Komponenten und der Zugang zur CAD-Umgebung an der Hochschule ohne Weiteres realisierbar sind, setzt die praktische Erprobung der konstruierten Bauteile die Verfügbarkeit eines 3D-Druckers während der Bearbeitungsphase voraus. Sofern die Studierenden keinen eigenen Zugang zu einem solchen Gerät haben, ist eine Bereitstellung durch die Hochschule erforderlich. Sinnvoll wäre eine begleitende Kurzconsultation vor jedem Druckauftrag, um vermeidbare Fehldrucke zu reduzieren.

Über diese strukturellen Einschränkungen hinaus ist das Modul bisher rein konzeptionell entwickelt; eine Erprobung mit Studierenden steht noch aus. Welche der getroffenen Annahmen sich in einer solchen Erprobung bewähren, lässt sich aus der bisherigen Bearbeitung nicht abschließend beurteilen. Insbesondere die angesetzte Bearbeitungszeit von zehn Wochen sowie die damit verknüpfte zeitliche Staffelung der einzelnen Arbeitsschritte beruhen auf einer Abschätzung des erforderlichen Aufwands und müssten in realen Bedingungen überprüft werden.

Schließlich klammert das Modul bewusst eine Reihe von Aspekten der realen Package-Entwicklung aus. Tiefgreifende sicherheitstechnische Prüfungen, crashrelevante Bewertungen und fertigungsbezogene Freigabeprozesse würden den Rahmen einer Wahlmodul-Begleitaufgabe überschreiten und sind nicht Bestandteil der Aufgabenstellung. Diese Reduktion ist eine notwendige Voraussetzung für die didaktische Zugänglichkeit, schränkt jedoch an einigen Stellen den unmittelbaren Praxisbezug ein.

7 Zusammenfassung

Die vorliegende Arbeit erweitert das Wahlmodul „Ergonomie und Package“ der Fakultät Kraftfahrzeugtechnik um eine praxisorientierte Begleitaufgabe. Die Bauraumintegration in einem autonomen Modellfahrzeug stellt die zentrale Aufgabenstellung dar. Theoretisch vermittelte Inhalte werden so in die praktische Anwendung überführt. Die Programmierung des Mikrocontrollers kann KI-gestützt erfolgen. Der konzeptionelle Umfang hat sich im Verlauf der Bearbeitung über das anfänglich erwartete Maß hinaus erweitert. Insbesondere die Komplexität der Karosseriegestaltung und der Umfang der integrierten Bewertungslogik haben Anpassungen am ursprünglichen Konzept erforderlich gemacht.

Die bildungstheoretische Grundlage des Moduls ist der problemorientierte Unterricht nach Klafki. Die didaktische Struktur orientiert sich am Drei-Phasen-Modell nach Wilhelm und Brovelli.

Aus der Arbeit gehen das ausgearbeitete Modulkonzept, das parametrische CAD-Modell mit integrierter Bewertungslogik, ein fahrender Hardware-Prototyp sowie ergänzende Materialien wie der KI-Leitfaden hervor. Dieser enthält Empfehlungen zur Formulierung von Prompts sowie zur Bewertung des generierten Quellcodes. Den größten Beitrag leistet das CAD-Modell. Es nutzt die Berechnungsmöglichkeiten von 3DX für geometrische Auswertungen, die im regulären Studienverlauf bislang nicht vertieft behandelt werden.

Die formelbasierte Verknüpfung von Geometrie und Bewertung erwies sich als zentraler Vorteil des Modells. Da die Bewertungslogik ihre Eingangsgrößen aus der jeweils aktuellen Modellgeometrie entnimmt, aktualisiert sie sich bei jeder Konstruktionsänderung mit jedem Assembly-Update. Die Studierenden können dadurch iterativ Verbesserungen vornehmen, bevor die finale Abgabe erfolgt. Eine weitere Erkenntnis betrifft die Modellierung der Bauteilmassen. Da 3DX Massen ausschließlich aus Volumen und Materialdichte berechnet, wurden für die Elektronikkomponenten Pseudo-Materialien definiert. Damit fließen die korrekten Bauteilmassen in die Schwerpunktberechnung ein, ohne die Geometrie modifizieren zu müssen.

Im Umgang mit der KI-gestützten Codegenerierung zeigte sich, dass eine eindeutig eingegrenzte Aufgabenstellung entscheidend für die Verlässlichkeit der erzeugten Quellcodes ist. Während sich für klar umschriebene Anwendungsbereiche reproduzierbare Ergebnisse erzielen lassen, führen offene Aufgabenstellungen zu Code, der iterativ verbessert werden muss.

Den nächsten Schritt bildet die Erprobung des Moduls mit Studierenden. Nur unter realen Bedingungen lässt sich prüfen, ob die getroffenen Annahmen belastbar sind. Das gilt insbesondere für die angesetzten Bearbeitungszeiträume. Daraus ergeben sich auch fundierte Empfehlungen zur zeitlichen Strukturierung der einzelnen Phasen.

Darüber hinaus bietet das Konzept weitere Ansatzpunkte für eine inhaltliche Erweiterung. Ein Innenraum lässt sich derzeit nicht integrieren, da das verfügbare Volumen nahezu vollständig von der Elektronik belegt ist. Eine Anpassung der Gesamtgeometrie würde diesen Aspekt ermöglichen. Eine kamerabasierte Erfassung der Fahrerperspektive mit einem ESP32-CAM-Modul ist technisch möglich, wurde jedoch zugunsten eines konsistenten Hardwarekonzepts zurückgestellt. Ein externer Lade- und Programmierport, der es ermöglicht, die Karosserie zur Inbetriebnahme nicht öffnen zu müssen, ist konstruktiv ergänzbar und bildet eine eigenständige Package-Teilaufgabe.

Die Arbeit schafft eine Grundlage, die theoretische Lehre und konstruktive Praxis im Wahlmodul verbindet; ein aufgebauter Hardware-Prototyp (Bild 16) liegt bereits vor, seine Wirksamkeit ist in einer anschließenden Erprobung mit Studierenden zu prüfen.



Bild 16: Hardware-Prototyp des Modellfahrzeugs

Literaturverzeichnis

Arduino (2026): Arduino Nano Dokumentation. Online verfügbar unter <https://docs.arduino.cc/hardware/nano/>, abgerufen am 06.02.2026.

DIN ISO 4130 (1979): Straßenfahrzeuge – 3-dimensionales Bezugssystem und primäre Bezugspunkte – Definitionen. Berlin: Beuth Verlag.

FEETECH (2016): FS90R Micro Continuous Rotation Servo Datenblatt. Online verfügbar unter <https://www.pololu.com/product/2820>, abgerufen am 06.02.2026.

Grabner, J.; Nothhaft, R. (2006): Konstruieren von Pkw-Karosserien. Grundlagen, Elemente und Baugruppen, Vorschriftenübersicht, Beispiele mit CATIA V4 und V5. 3. Auflage. Berlin: Springer.

Karle, A. (2022): Elektromobilität – Grundlagen und Praxis. 6., aktualisierte und erweiterte Auflage. München: Carl Hanser Verlag.

Klafki, W. (2007): Neue Studien zur Bildungstheorie und Didaktik. Zeitgemäße Allgemeinbildung und kritisch-konstruktive Didaktik. 6. Auflage. Weinheim/Basel: Beltz.

Klieme, E.; Avenarius, H.; Blum, W.; Döbrich, P.; Gruber, H.; Prenzel, M.; Reiss, K.; Riquarts, K.; Rost, J.; Tenorth, H.-E.; Vollmer, H. J. (2003): Zur Entwicklung nationaler Bildungsstandards. Eine Expertise. Bonn: Bundesministerium für Bildung und Forschung.

Köhler, H. H. (2025): Interaktives Lehrmodul zur Konstruktion auf der Packagegrundlage eines autonomen Modellfahrzeuges. Studienarbeit. Westsächsische Hochschule Zwickau, Fakultät Kraftfahrzeugtechnik.

Kreis, A.; Hirz, M.; Rossbacher, P. (2021): CAD-Automation in Automotive Development – Potentials, Limits and Challenges. In: Computer-Aided Design & Applications, 18(4), S. 849–863.

Marx, S.; Götze, D. (2018): Problembasiertes Lernen in der Hochschullehre. In: Dombrowski, U.; Marx, S. (Hrsg.): Klimaling – Planung klimagerechter Fabriken. Berlin: Springer, S. 189–216.

Nagel, L. (2023): Fahrzeugpackage. Vorlesungsunterlagen. Westsächsische Hochschule Zwickau, Fakultät Kraftfahrzeugtechnik.

STMicroelectronics (2016a): VL53L0X - World smallest Time-of-Flight ranging and gesture detection sensor. Datenblatt. Online verfügbar unter <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>, abgerufen am 04.02.2026.

STMicroelectronics (2016b): AN4846 – Using multiple VL53L0X in a single design. Application Note. Online verfügbar unter https://www.st.com/resource/en/application_note/an4846-using-multiple-vl53l0x-in-a-single-design-stmicroelectronics.pdf, abgerufen am 06.02.2026.

Stadler, S.; Hirz, M.; Thum, K.; Rossbacher, P. (2013): Conceptual Full-Vehicle Development Supported by Integrated Computer-Aided Design Methods. In: Computer-Aided Design & Applications, 10(1), S. 159–172.

Vajna, S.; Weber, C.; Zeman, K.; Hehenberger, P.; Gerhard, D.; Wartzack, S. (2018): CAx für Ingenieure. Eine praxisbezogene Einführung. 3. Auflage. Berlin: Springer Vieweg.

Wilhelm, M.; Brovelli, D. (2009): Problembasiertes Lernen (PBL) in der Lehrpersonenbildung: Der Drei-Phasen-Ansatz der Naturwissenschaften. In: Beiträge zur Lehrerbildung, 27(2), S. 195–203.

Übersicht verwendeter Hilfsmittel

Gemäß der Selbstständigkeitserklärung werden nachfolgend alle auf künstlicher Intelligenz (KI) basierenden Werkzeuge aufgeführt, die im Rahmen dieser Arbeit verwendet wurden. Die Nutzung erfolgte in Absprache mit den Prüfern.

Produktname	Genutzter Funktionsumfang
Claude (Anthropic), verschiedene Modelle (Opus, Sonnet 4.5, 4.6, 4.7)	Code-Generierung für Arduino-Steuerung; iterative Verbesserung des generierten Codes und Lektorat der gesamten Arbeit

Anmerkung: Die generierten Codes wurden kritisch geprüft, manuell angepasst und empirisch validiert. Die Verantwortung für alle Inhalte liegt vollständig beim Autor.

Anlagen

Anlage 1: Aufgabenstellung für Studierende

Begleitaufgabe Package mit Sensorik

Anleitung zur Nutzung der CAD-Grunddatei in CATIA 3DEXPERIENCE

1 Übersicht der Grunddatei

Die bereitgestellte CAD-Grunddatei enthält alle vorbereiteten Elemente, die für die Bearbeitung der Package-Aufgabe erforderlich sind. Die folgende Tabelle gibt einen Überblick über die enthaltenen Bestandteile und deren Funktionen.

Bestandteil	Inhalt / Funktion
BasisGeometrie	Skelett-Part mit den Masterparametern Radstand und Spurweite. Alle abhängigen Geometrien leiten sich aus diesen beiden Werten ab.
Grundplatte	Vierteiliges Plattensystem mit vorpositionierten Motorhaltern, Servomotoren und Positionierungstaschen. Skaliert automatisch mit den Masterparametern.
Sensor-Parts (4x)	VL53L0X-Sensormodelle mit eingebetteten Messkegel-Bodies. Die Kegel bewegen sich automatisch mit der Sensorposition mit.
BEWERTUNG_NICHT_AENDERN	Geometrical Set mit allen KPI-Bewertungselementen. Initial ausgeblendet. Darf nicht verändert werden.
Bewertungssteuerung	Enthält die Eingangsparameter COG_X/Y/Z sowie die Schalter zur Aktivierung der KPI.
Materialbibliothek	Pseudo-Materialien für alle Elektronikkomponenten mit realen Masseneigenschaften.
Karosserie_A	Vorbereiteter Platzhalter-Part (20 × 20 × 20 mm Würfel). Bauen Sie diesen zur eigenen Karosserie aus. Enthält das Geometrical Set „GeoSet_Bauraumauswertung“ für die Bauraumnutzungsberechnung. Den Part-Namen dürfen Sie nicht verändern.
Breadboard und Powerbank	Bereits in der Grunddatei vorpositioniert. Weisen Sie das Pseudo-Material zu und konstruieren Sie die Wannenhalter.

2 Erste Schritte

2.1 Datei öffnen und Masterparameter einstellen

Öffnen Sie die Grunddatei in CATIA 3DEXPERIENCE. Navigieren Sie im Strukturbaum zum Part BasisGeometrie und passen Sie die beiden Masterparameter Radstand und Spurweite an die gewünschten Maße an.

1. Doppelklick auf BasisGeometrie → Part aktivieren
2. Parameter Radstand und Spurweite durch Doppelklick editieren und Zielwerte eintragen
3. Aktualisierung über das Update-Symbol auslösen
4. Prüfen, ob Grundplatte und Halter korrekt skaliert wurden

Hinweis: Alle abhängigen Geometrien wie Grundplatte, Positionierungstaschen und Bewertungszonen aktualisieren sich automatisch. Eine manuelle Anpassung einzelner Elemente ist nicht erforderlich.

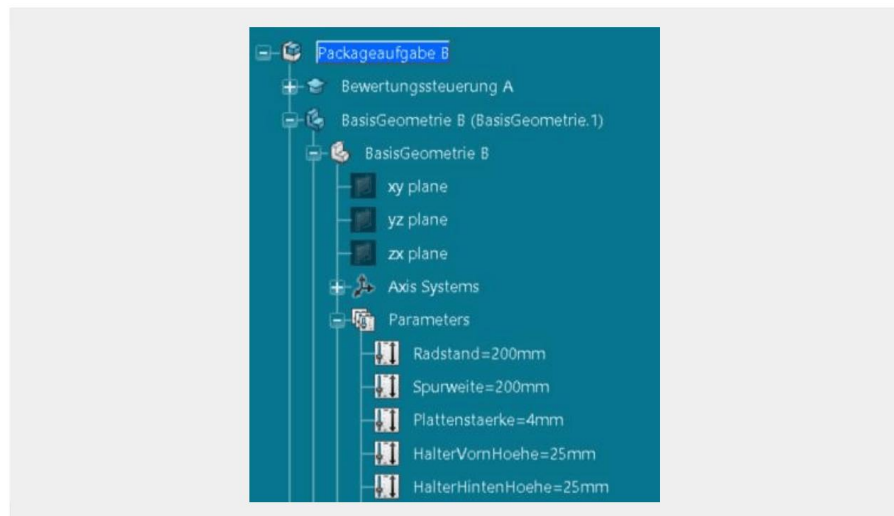


Abb. A-1: Masterparameter Radstand und Spurweite in der BasisGeometrie

2.2 Materialien zuweisen

Für eine korrekte Schwerpunktberechnung muss jeder Komponente das zugehörige Material aus dem Materialkatalog zugewiesen werden. Die Materialien sind bereits auf die tatsächlichen Bauteilmassen kalibriert.

5. Bauteil im Strukturbaum markieren
6. Rechtsklick → Eigenschaften → Reiter Material
7. In der Bibliothek das passende Material auswählen und zuweisen

Bauteil	Pseudo-Material
Arduino Nano	<i>Arduino Nano</i>
VL53L0X Sensor (×4)	<i>VL53L0X</i>
FS90R Servo (×4)	<i>FS90R Servo</i>
Powerbank	<i>Powerbank</i>
Breadboard	<i>Breadboard</i>
Alle gedruckten Bauteile (Grundplatte, Halter, Karosserie ...)	<i>PLA</i>

Hinweis: Weisen Sie auch allen selbst konstruierten Bauteilen das PLA-Material zu, sobald sie in die Assembly eingefügt sind. Ohne PLA-Zuweisung wird deren Masse bei der Schwerpunktberechnung nicht korrekt berücksichtigt.

3 Eigene Bauteile konstruieren

3.1 Sensorhalter

Für jeden der vier VL53L0X-Sensoren ist eine individuelle Halterung zu konstruieren, die den Sensor über dessen Montagelöcher (Stifte mit einem Radius von 1,4 mm, im Lochabstand von 20 mm, jeweils 3 mm über der Oberfläche herausstehend) im gewünschten Winkel zur Grundplatte positioniert. Die Halterung wird mithilfe der Positionierungstaschen an der Grundplatte befestigt.

8. Neues Part anlegen und über Engineering Connections mit der Grundplatte verknüpfen
9. Orientierung des Sensors (Neigungswinkel, Drehung) festlegen
10. Halterungskörper modellieren Mindeststärke 1 mm
11. Kabelführungsaussparung (16*3,25 mm) einarbeiten
12. Sensor-Part in die Assembly einbauen und auf Halterung referenzieren

Hinweis: Die Sensorhalter bestimmen maßgeblich den Abdeckungswinkel und die Überlappungsquote. Nutzen Sie das Bewertungs-Set (siehe Abschnitt 5) zur iterativen Optimierung.

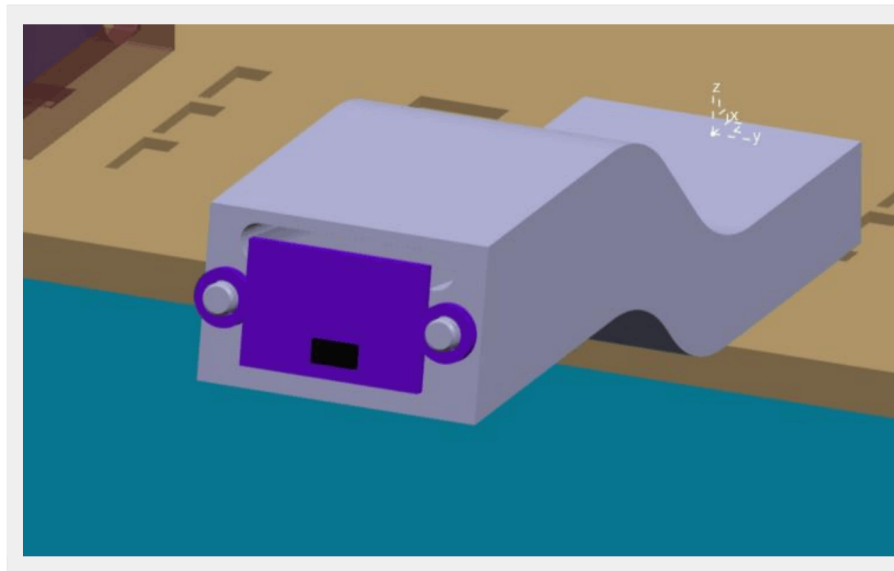


Abb. A-2: Beispiel Sensorhalter mit Sensor-PCB und Messkegel

3.2 Wannenhalter für Breadboard und Powerbank

Breadboard und Powerbank werden in wannenförmigen Halterungen aufgenommen. Die Wannengeometrie lässt sich direkt aus den Bauteilabmessungen ableiten. Eine umlaufende Lippe sichert das Bauteil formschlüssig.

13. Abmessungen des Bauteils aus dem bereitgestellten CAD-Modell oder Datenblatt entnehmen
14. Wanne mit 0,1–0,2 mm Spiel je Seite modellieren
15. Umlaufende Lippe ~1,5 mm über Bauteilrand vorsehen

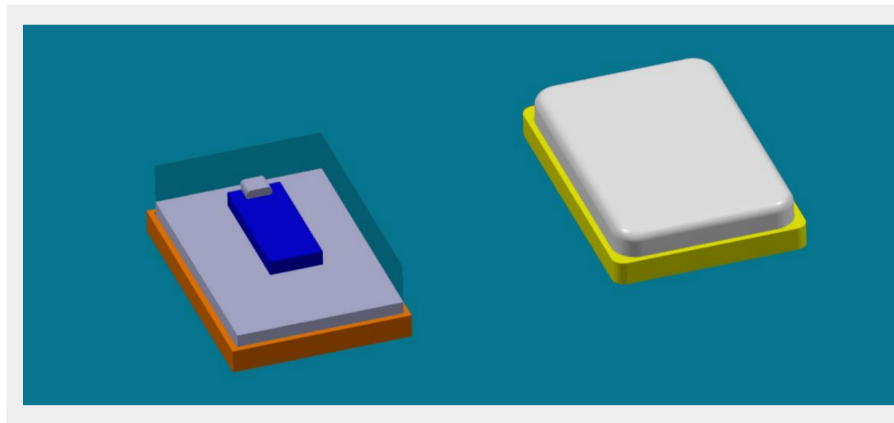


Abb. A-3: Wannenhalter für Breadboard (links) und Powerbank (rechts)

3.3 Karosserie

In der Grunddatei ist der Part „Karosserie_A“ mit einem einfachen $20 \times 20 \times 20$ mm Würfel als Platzhalter vorbereitet. Ersetzen Sie diesen im Laufe der Aufgabe durch Ihre eigene Karosseriegeometrie. Den Part-Namen dürfen Sie nicht verändern, da alle KPI-Referenzen darauf verweisen. Gestalten Sie die Karosserie als Freiformfläche, die alle Komponenten des Fahrzeugs überdeckt, ohne deren Funktion zu beeinträchtigen. Die Außenhaut muss die Sensor-Messkegel kollisionsfrei passieren lassen.

16. Packageraum durch Einblenden des Bewertungs-Sets prüfen
17. Außenhaut als Flächenmodell mit Generative Shape Design erstellen
18. Mit ThickSurface in Solid wandeln, Wandstärke mindestens 1 mm
19. Ausschnitte für Sensor-Öffnungen als Pockets einarbeiten
20. Aufnahmen für Karosserie auf den Haltern konstruieren (Pad+Remove)
21. Kollisionsprüfung mit KPI-Auswertung durchführen (Abschnitt 5)

Hinweis: Verrundungen können die Wandstärke lokal deutlich verringern. Prüfen Sie vor dem Druck alle Verrundungsbereiche mit Measure Thickness auf die Einhaltung der Mindestwandstärke von 1 mm.

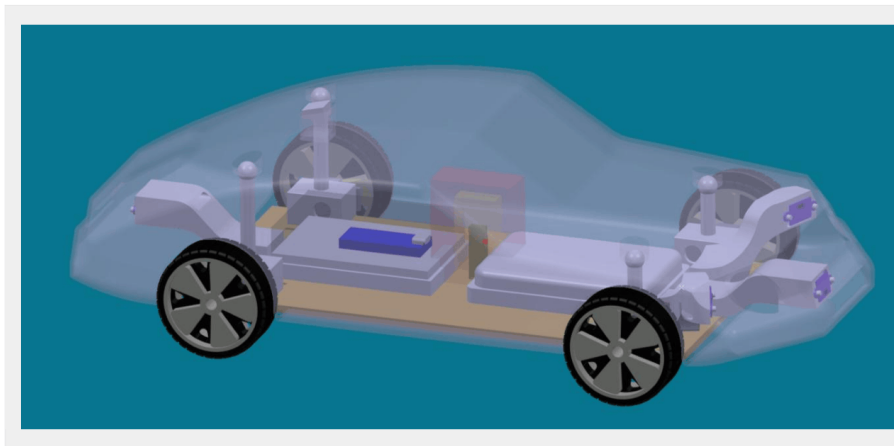


Abb. A-4: Beispielkarosserie mit eingeblendetem Bewertungs-Set

3.4 Bauraumauswertung vorbereiten

Für die automatische Berechnung der Bauraumnutzung ist im Part „Karosserie_A“ das Geometrical Set „Geometrical Set.Bewertung“ vorbereitet. Sobald Ihre Karosserie als Solid vorliegt, ziehen Sie das Innenraumvolumen in dieses Set, damit die KPI-Formel korrekte Werte liefert.

- Karosserie-Innenfläche als geschlossenes Volumen anlegen (Insert → Surface-Based Features → Close Surface, Innenflächen der Schale auswählen)
- Im Strukturbaum Karosserie_A per Doppelklick aktivieren → darin „Geometrical Set.Bewertung“ per Doppelklick als aktives Set setzen
- Das geschlossene Innenraumvolumen als Referenz des „Trim.1“ auswählen
- Assembly-Update auslösen → Das System trimmt das Volumen automatisch an der Fahrzeugmitte, dickt es um 1 mm auf und berechnet V_Bauraum_Halb
- KPI-Auswertung aktivieren (Abschnitt 5) → Bauraumnutzung wird automatisch ausgegeben

Hinweis: Solange Ihre Karosserie noch nicht eingebaut ist, zeigt die Bauraumnutzung einen sehr kleinen Wert (Platzhalter-Würfel). Das ist korrekt und aktualisiert sich automatisch, sobald Sie Ihr Innenraumvolumen in das Geometrical Set referenziert haben.

3.5 Optionale Anpassung: Felgendesign

Sie können das Design der Felgen nach Belieben verändern. Diese Anpassung ist vollständig optional und fließt nicht in die KPI-Bewertung ein. Masse und Schwerpunktlage der Felgen werden weiterhin korrekt über das zugewiesene PLA-Material berücksichtigt, solange Sie das Material nicht ändern.

4 Schwerpunktberechnung und Dateneingabe

Die KPI-Auswertung der Schwerpunktlage erfordert die manuelle Übertragung der Schwerpunktkoordinaten aus der 3DEXPERIENCE-Massenanalyse in die Bewertungssteuerung.

4.1 Massenanalyse durchführen

22. Im Strukturbaum das Gesamt-Assembly markieren
23. Measure Inertia aufrufen
24. Werte Gx, Gy, Gz aus dem Ergebnisfenster ablesen

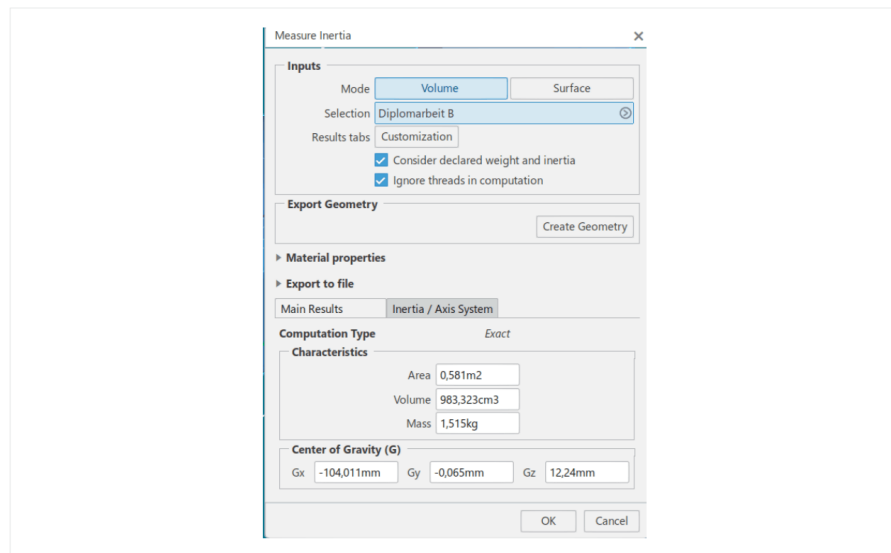


Abb. A-5: Massenanalyse mit Schwerpunktkoordinaten

4.2 Werte in Bewertungssteuerung eintragen

25. Im Strukturbaum Bewertungssteuerung → Parameters öffnen
26. COG_X: abgelesenen Gx-Wert eintragen
27. COG_Y: abgelesenen Gy-Wert eintragen
28. COG_Z: abgelesenen Gz-Wert eintragen

Hinweis: Das Vorzeichen der Koordinaten aus der Massenanalyse direkt übernehmen — mit einer Ausnahme: Die X-Achse der Basisgeometrie ist gegenüber der Measure-Inertia-Ausgabe invertiert. Die Eingabeformel für COG_X berücksichtigt dies automatisch, sodass der abgelesene Gx-Wert ohne manuelle Vorzeichenkorrektur eingetragen werden kann.

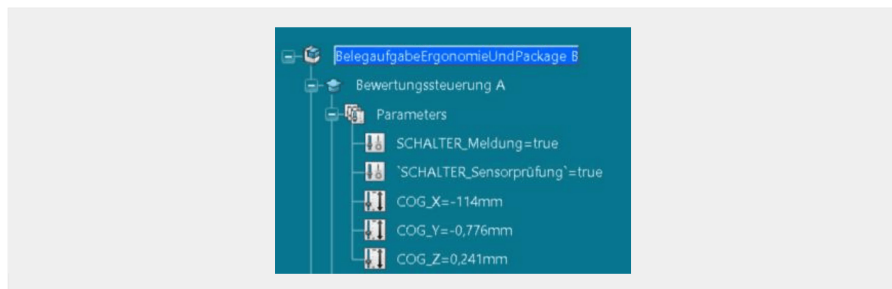


Abb. A-6: Eingabeparameter COG_X/Y/Z in der Bewertungssteuerung

5 KPI-Auswertung aktivieren

5.1 Bewertungs-Set einblenden

Das Geometrical Set BEWERTUNG_NICHT_AENDERN ist standardmäßig ausgeblendet und kann zur Selbstprüfung jederzeit eingeblendet werden.

1. Im Strukturbaum BEWERTUNG_NICHT_AENDERN markieren
2. Rechtsklick → Hide / Show
3. Die Schwerpunktzonen und Sensor-Messkegel werden in der 3D-Ansicht sichtbar
4. Zum Ausblenden: Rechtsklick → Hide / Show

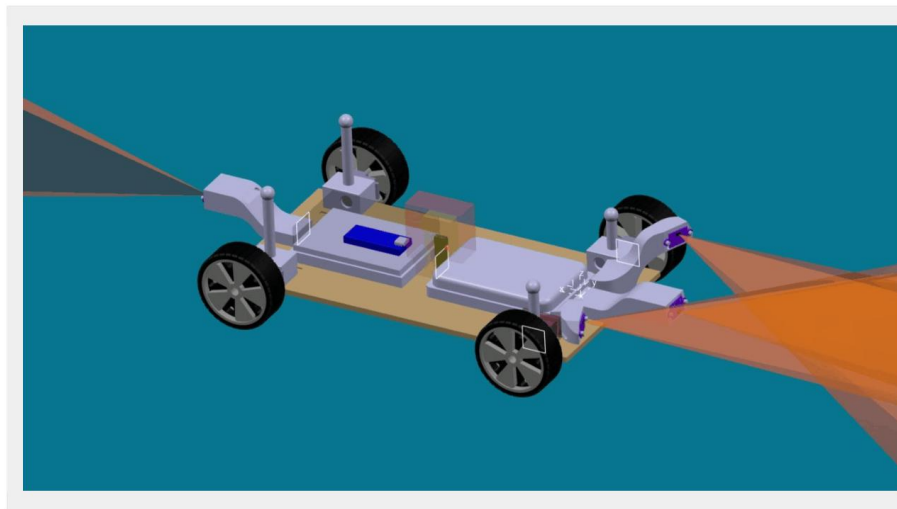


Abb. A-7: Eingeblendetes Bewertungs-Set mit Schwerpunktzonen und Messkegeln

5.2 Gesamtauswertung anzeigen

5. In Bewertungssteuerung → Parameters den Parameter SCHALTER_Meldung auf true setzen
6. Das KPI-Auswertungsfenster erscheint nach dem nächsten Assembly-Update
7. Ergebnisse ablesen und Konstruktion ggf. anpassen
8. SCHALTER_Meldung wieder auf false setzen, um die Meldung zu deaktivieren

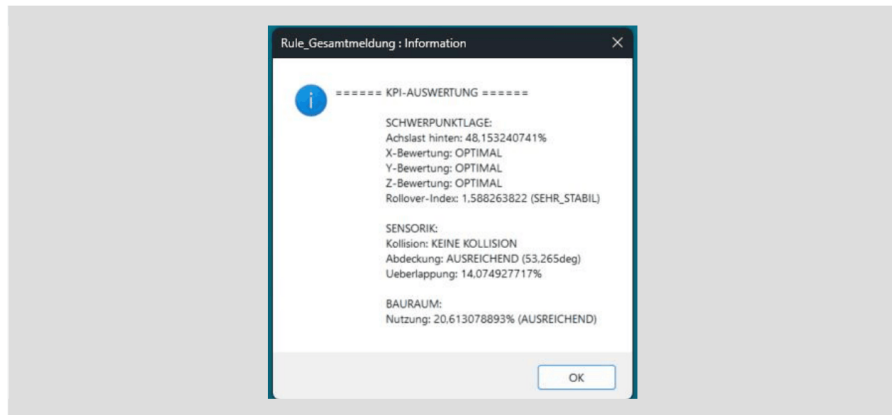


Abb. A-8: KPI-Auswertungsfenster mit allen Bewertungsergebnissen

5.3 Interpretation der Ergebnisse

KPI	Bewertung	Zielwert
Achslast hinten	OPTIMAL / GUT / AKZEPTABEL	48–52% optimal, 45–55% gut, 40–60% akzeptabel
Schwerpunkt X	OPTIMAL / GUT / AKZEPTABEL	Innerhalb der parametrischen Zonen
Schwerpunkt Y	OPTIMAL / GUT / AKZEPTABEL	Symmetrisch zur Längsachse
Schwerpunkt Z	OPTIMAL / GUT / AKZEPTABEL	Unterhalb Reifenoberkante = optimal
Rollover-Index	SEHR_STABIL / STABIL / GRENZWERTIG / KRITISCH	> 1,5 sehr stabil, > 1,2 stabil
Kollision	KEINE KOLLISION / KOLLISION	Alle Messkegel kollisionsfrei
Abdeckung	AUSREICHEND / GERING	Abdeckungswinkel der Frontsensoren
Überlappung	Prozentwert	Redundante Sensorabdeckung
Bauraumnutzung	EFFIZIENT / AUSREICHEND / INEFFIZIENT	≥ 35% effizient, 15–35% ausreichend

6 Fertigungsprüfung vor dem Druck

Bevor Sie Bauteile zum 3D-Druck übergeben, prüfen Sie alle selbst konstruierten Teile auf FDM-Druckbarkeit.

- 37. Bauteil aktivieren (Doppelklick im Strukturbaum)
- 38. Measure → Measure Thickness aufrufen
- 39. Kritische Bereiche konstruktiv anpassen

Hinweis: Besondere Aufmerksamkeit gilt Verrundungen: Der Krümmungsverlauf kann die Wandstärke lokal deutlich reduzieren, ohne dass dies in der normalen Modellansicht erkennbar ist.

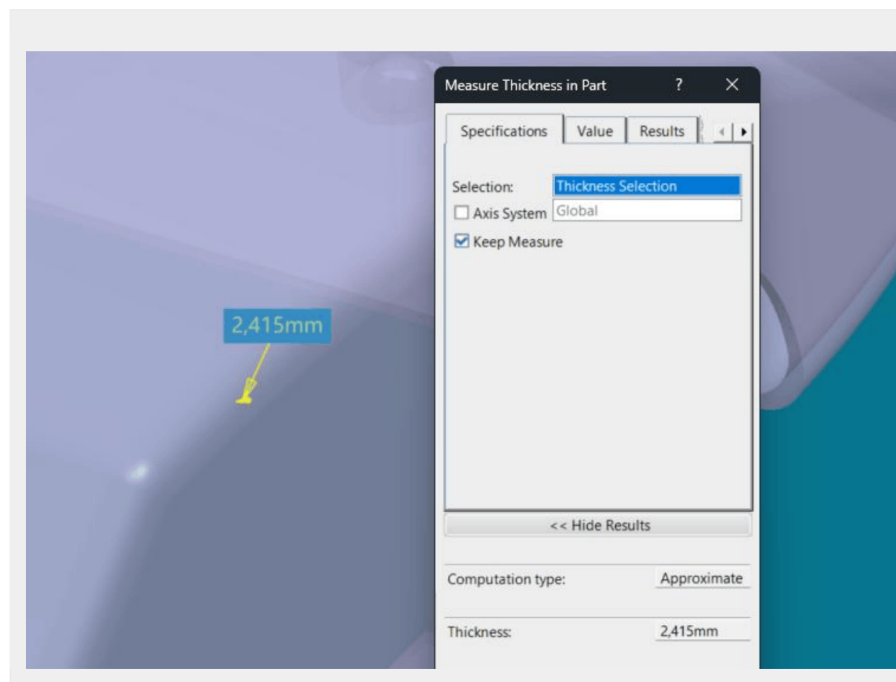


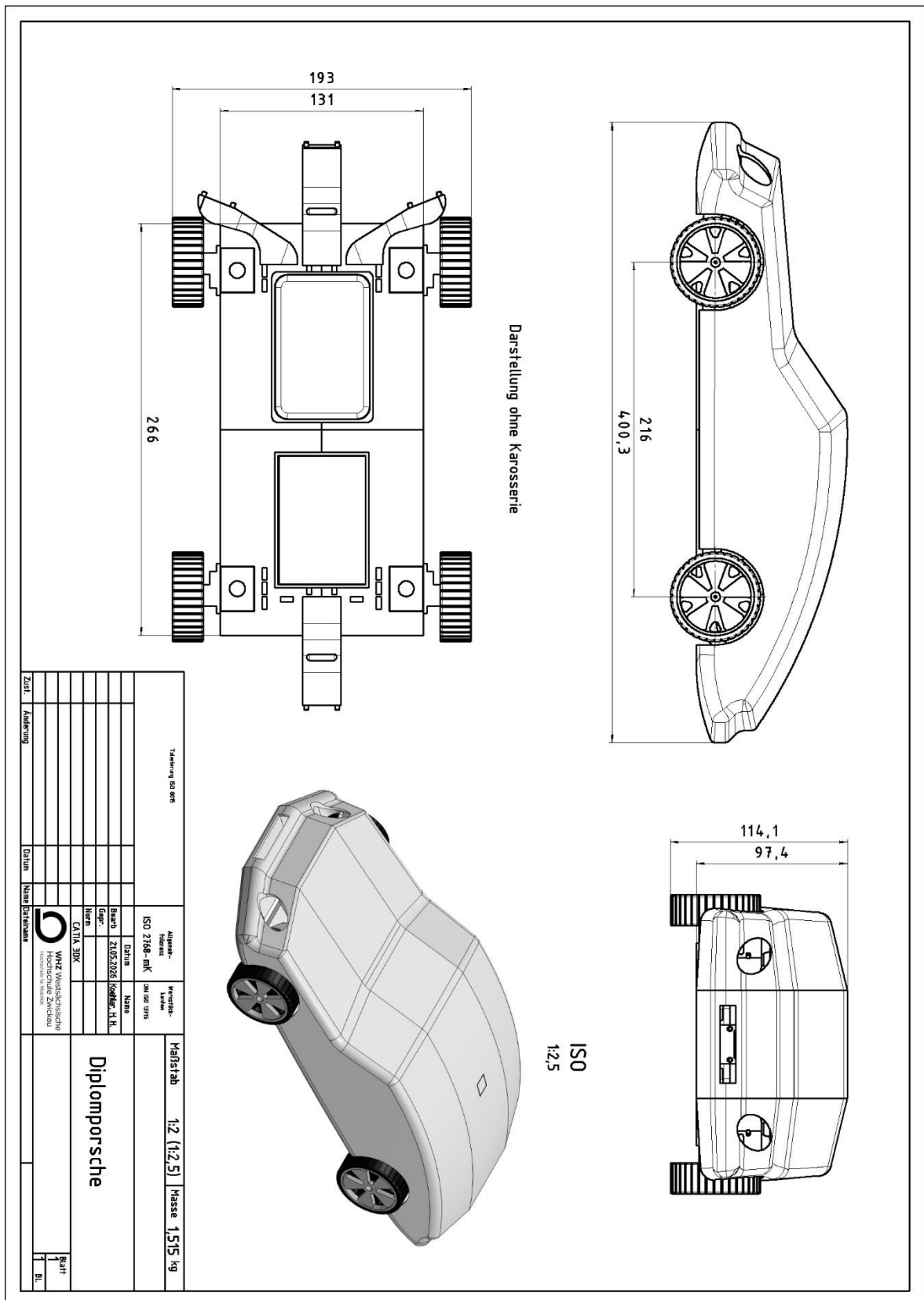
Abb. A-9: Measure Thickness in kritischen Bereichen

7 Geschützte Bereiche der Grunddatei

Die folgenden Elemente der Grunddatei dürfen nicht verändert werden, da sie die Grundlage für die automatische KPI-Auswertung bilden.

Element	Warum geschützt?
BEWERTUNG_NICHT_AENDERN	Enthält die gesamte KPI-Bewertungslogik, die Zonen-Quader sowie die Formeln. Änderungen verfälschen die Auswertung.
Messkegel-Bodies in Sensor-Parts	Die Kegel bilden die Grundlage für die Kollisions- und Überlappungsprüfung.
Motorhalter auf Grundplatte	Vorpositioniert und formelbasiert verknüpft. Position darf nicht manuell verschoben werden.
Pseudo-Materialien	Festgelegte Dichten für korrekte Massenzuweisung. Werte nicht ändern.

Anlage 2: CAD-Zeichnung



Anlage 3: Quellcode des Modells

```
#include <Wire.h>
#include <VL53L0X.h>
#include <Servo.h>
#include <avr/wdt.h>
#include <EEPROM.h>

// =====
// Pin-Konfiguration
// =====
#define XSHUT_FL 2 // Frontsensor links
#define XSHUT_FM 3 // Frontsensor mitte
#define XSHUT_FR 4 // Frontsensor rechts
#define XSHUT_H 5 // Hecksensor
#define TASTER 6 // Start-/Stopp-Taster (Pull-Up intern)
#define PIN_VL 9 // Servo vorne links
#define PIN_VR 10 // Servo vorne rechts
#define PIN_HL 11 // Servo hinten links
#define PIN_HR 12 // Servo hinten rechts

// =====
// Parameter
// =====
// Kalibrierungs-Offsets der Sensoren (mm), empirisch ermittelt
#define OFFSET_FL 21
#define OFFSET_FM 0
#define OFFSET_FR 60
#define OFFSET_H 47

// Reaktionsschwellen fuer die Fahrlogik
#define SCHWELLE_FRONT 250 // Abstand mm fuer Frontalhindernis
#define SCHWELLE_SEITE 100 // Abstand mm fuer seitliche Korrektur

// Servosteuerung (FS90R: 0...180 um Mitte 90)
#define VORWAERTS 45
#define RUECKWAERTS 135

// Sicherheits-Timeout fuer einen Fahrlauf
#define TIMEOUT_MS 30000

// Recovery-Parameter
#define MAX_FAIL 10 // 9999-Reads in Folge bis Sensor-Reinit
#define LOG_INTERVAL_MS 500 // Logging-Intervall waehrend Fahrt
#define LOG_MAX_ENTRIES 90 // 90 * 11 Byte = 990 Byte, passt in 1KB EEPROM

// =====
// Globale Objekte
// =====
VL53L0X sensor_FL, sensor_FM, sensor_FR, sensor_H;
Servo vl, vr, hl, hr;

// =====
// Zustandsvariablen
// =====
bool aktivModus = false;
unsigned long startZeit = 0;
bool letzterTasterZustand = HIGH;
bool tasterGedrueckt = false;
uint8_t fail_count[4] = {0, 0, 0, 0};
unsigned long letztes_log = 0;
uint16_t log_anzahl = 0;

// =====
XV
```

```

// Status-Codes (fuer EEPROM-Logging)
// =====
#define STATUS_STOP      0
#define STATUS_VOR       1
#define STATUS_ZURUECK   2
#define STATUS_LINKS     3
#define STATUS_RECHTS    4
#define STATUS_FMTOT     5

// =====
// Servo-Steuerung
// =====

// Alle vier Servos auf ihre Pins legen
void servoAttach() {
    vl.attach(PIN_VL);
    vr.attach(PIN_VR);
    hl.attach(PIN_HL);
    hr.attach(PIN_HR);
}

// Alle Servos vom PWM-Signal trennen (eliminiert Brummen im Stillstand)
void alleStop() {
    vl.detach();
    vr.detach();
    hl.detach();
    hr.detach();
}

void fahreVorwaerts() {
    servoAttach();
    vl.write(VORWAERTS);  vr.write(RUECKWAERTS);
    hl.write(VORWAERTS);  hr.write(RUECKWAERTS);
}

void fahreRueckwaerts() {
    servoAttach();
    vl.write(RUECKWAERTS); vr.write(VORWAERTS);
    hl.write(RUECKWAERTS); hr.write(VORWAERTS);
}

void dreheLinks() {
    servoAttach();
    vl.write(RUECKWAERTS); vr.write(RUECKWAERTS);
    hl.write(RUECKWAERTS); hr.write(RUECKWAERTS);
}

void dreheRechts() {
    servoAttach();
    vl.write(VORWAERTS); vr.write(VORWAERTS);
    hl.write(VORWAERTS); hr.write(VORWAERTS);
}

// =====
// Sensorik
// =====

// Liefert kalibrierten Abstand in mm.
// Rueckgabe 9999 = ausserhalb Messbereich oder Timeout
uint16_t lese(VL53L0X &s, int16_t offset) {
    uint16_t val = s.readRangeContinuousMillimeters();
    if (val > 2000) return 9999;
    int16_t korrigiert = (int16_t)val - offset;
    if (korrigiert < 0) return 0;
    return (uint16_t)korrigiert;
}

```

```

// =====
// Auto-Recovery
// =====

// Einzelnen Sensor neu initialisieren (z.B. nach Dauerausfall)
void reinitSensor(uint8_t idx) {
    const uint8_t xshut_pins[] = {XSHUT_FL, XSHUT_FM, XSHUT_FR, XSHUT_H};
    const uint8_t adressen[] = {0x30, 0x31, 0x32, 0x33};
    VL53L0X* const sensoren[] = {&sensor_FL, &sensor_FM, &sensor_FR, &sensor_H};
    const char* const namen[] = {"FL", "FM", "FR", "H"};

    Serial.print(F("Re-init "));
    Serial.print(namen[idx]);
    Serial.print(F(": "));

    digitalWrite(xshut_pins[idx], LOW);
    delay(10);
    digitalWrite(xshut_pins[idx], HIGH);
    delay(10);

    if (sensoren[idx]->init()) {
        sensoren[idx]->setAddress(adressen[idx]);
        sensoren[idx]->setTimeout(500);
        sensoren[idx]->startContinuous();
        fail_count[idx] = 0;
        Serial.println(F("OK"));
    } else {
        Serial.println(F("FEHLER"));
    }
}

// Komplette XSHUT-Sequenz neu durchlaufen
// Wird ausgelöst wenn alle vier Sensoren gleichzeitig ausfallen
// (typischerweise nach Busblockade)
void reinitAlleSensoren() {
    Serial.println(F("=== KOMPLETT-REINIT alle Sensoren ==="));

    digitalWrite(XSHUT_FL, LOW);
    digitalWrite(XSHUT_FM, LOW);
    digitalWrite(XSHUT_FR, LOW);
    digitalWrite(XSHUT_H, LOW);
    delay(50);

    digitalWrite(XSHUT_FL, HIGH);
    delay(10);
    sensor_FL.init();
    sensor_FL.setAddress(0x30);
    sensor_FL.setTimeout(500);

    digitalWrite(XSHUT_FM, HIGH);
    delay(10);
    sensor_FM.init();
    sensor_FM.setAddress(0x31);
    sensor_FM.setTimeout(500);

    digitalWrite(XSHUT_FR, HIGH);
    delay(10);
    sensor_FR.init();
    sensor_FR.setAddress(0x32);
    sensor_FR.setTimeout(500);

    digitalWrite(XSHUT_H, HIGH);
    delay(10);
    sensor_H.init();
    sensor_H.setAddress(0x33);
    sensor_H.setTimeout(500);

    sensor_FL.startContinuous();
}

```

```

    sensor_FM.startContinuous();
    sensor_FR.startContinuous();
    sensor_H.startContinuous();

    for (uint8_t i = 0; i < 4; i++) {
        fail_count[i] = 0;
    }
}

// =====
// EEPROM-Logging
// =====

// Schreibt einen Sample-Eintrag (11 Byte) ins EEPROM
void logSchreibe(uint16_t fl, uint16_t fm, uint16_t fr, uint16_t h,
                uint8_t statuscode) {
    if (log_anzahl >= LOG_MAX_ENTRIES) return;

    uint16_t addr    = 2 + log_anzahl * 11;
    uint16_t time_ds = (uint16_t)((millis() - startZeit) / 100);

    EEPROM.put(addr, time_ds); addr += 2;
    EEPROM.put(addr, fl);      addr += 2;
    EEPROM.put(addr, fm);      addr += 2;
    EEPROM.put(addr, fr);      addr += 2;
    EEPROM.put(addr, h);        addr += 2;
    EEPROM.write(addr, statuscode);

    log_anzahl++;
    EEPROM.put(0, log_anzahl);
}

// Gibt den letzten gespeicherten Run als Tab-separierte Tabelle ueber
// Serial aus. Wird beim Hochfahren aufgerufen.
void logAusgabe() {
    uint16_t count;
    EEPROM.get(0, count);
    if (count == 0 || count > LOG_MAX_ENTRIES) {
        Serial.println(F("=== Keine Log-Daten ==="));
        return;
    }
    Serial.print(F("=== Letzter Run: "));
    Serial.print(count);
    Serial.println(F(" Samples ==="));
    Serial.println(F("zeit_s\tfL\tfM\tfR\tH\tstatus"));

    const char* const statusname[] = {
        "STOP", "VOR", "ZURUECK", "LINKS", "RECHTS", "FMTOT"
    };

    for (uint16_t i = 0; i < count; i++) {
        uint16_t addr = 2 + i * 11;
        uint16_t time_ds, fl, fm, fr, h;
        uint8_t st;

        EEPROM.get(addr, time_ds); addr += 2;
        EEPROM.get(addr, fl);      addr += 2;
        EEPROM.get(addr, fm);      addr += 2;
        EEPROM.get(addr, fr);      addr += 2;
        EEPROM.get(addr, h);        addr += 2;
        st = EEPROM.read(addr);

        Serial.print(time_ds / 10);
        Serial.print('.');
        Serial.print(time_ds % 10);
        Serial.print('\t');
        Serial.print(fl); Serial.print('\t');
        Serial.print(fm); Serial.print('\t');

```

```

        Serial.print(fr); Serial.print('\t');
        Serial.print(h); Serial.print('\t');
        Serial.println(st < 6 ? statusname[st] : "?");
    }
    Serial.println(F("== Ende Log =="));
}

// =====
// Hilfsfunktionen
// =====

void stoppMelden(const char *grund) {
    alleStop();
    Serial.print(F("STOPP: "));
    Serial.println(grund);
    Serial.println(F("Bereit, Knopf zum Starten"));
}

// Erkennt fallende Flanke am Taster, Entprellung 50 ms
void tasterUpdate() {
    bool aktuell = digitalRead(TASTER);
    tasterGedrueckt = false;
    if (aktuell == LOW && letzterTasterZustand == HIGH) {
        delay(50);
        if (digitalRead(TASTER) == LOW) {
            tasterGedrueckt = true;
        }
    }
    letzterTasterZustand = aktuell;
}

// =====
// Setup
// =====
void setup() {
    wdt_disable();
    Serial.begin(9600);
    delay(500);

    // Daten vom letzten Run rausgeben bevor das aktuelle Setup
    // sie ueberschreiben koennte
    logAusgabe();

    Wire.begin();
    delay(100);
    pinMode(TASTER, INPUT_PULLUP);

    // XSHUT aller Sensoren auf LOW = alle Sensoren in Shutdown
    pinMode(XSHUT_FL, OUTPUT); digitalWrite(XSHUT_FL, LOW);
    pinMode(XSHUT_FM, OUTPUT); digitalWrite(XSHUT_FM, LOW);
    pinMode(XSHUT_FR, OUTPUT); digitalWrite(XSHUT_FR, LOW);
    pinMode(XSHUT_H, OUTPUT); digitalWrite(XSHUT_H, LOW);
    delay(10);

    // Jeden Sensor einzeln hochfahren und auf individuelle Adresse setzen
    digitalWrite(XSHUT_FL, HIGH);
    delay(10);
    sensor_FL.init();
    sensor_FL.setAddress(0x30);
    sensor_FL.setTimeout(500);

    digitalWrite(XSHUT_FM, HIGH);
    delay(10);
    sensor_FM.init();
    sensor_FM.setAddress(0x31);
    sensor_FM.setTimeout(500);
}

```

```

digitalWrite(XSHUT_FR, HIGH);
delay(10);
sensor_FR.init();
sensor_FR.setAddress(0x32);
sensor_FR.setTimeout(500);

digitalWrite(XSHUT_H, HIGH);
delay(10);
sensor_H.init();
sensor_H.setAddress(0x33);
sensor_H.setTimeout(500);

sensor_FL.startContinuous();
sensor_FM.startContinuous();
sensor_FR.startContinuous();
sensor_H.startContinuous();

Serial.println(F("setup ende, Knopf zum Starten"));
wdt_enable(WDTO_8S);
}

// =====
// Hauptschleife
// =====
void loop() {
    wdt_reset();
    tasterUpdate();

    // Wartezustand: auf Taster reagieren um Fahrlauf zu starten
    if (!aktivModus) {
        if (tasterGedrueckt) {
            aktivModus = true;
            startZeit = millis();
            log_anzahl = 0;
            EEPROM.put(0, log_anzahl);
            letztes_log = millis() - LOG_INTERVAL_MS;
            Serial.println(F("START"));
        }
        return;
    }

    // Aktiv-Modus: Abbruchbedingungen pruefen
    unsigned long vergangen = millis() - startZeit;
    if (vergangen >= TIMEOUT_MS) {
        aktivModus = false;
        stoppMelden("Timeout");
        return;
    }
    if (tasterGedrueckt) {
        aktivModus = false;
        stoppMelden("Manuell gestoppt");
        return;
    }

    // Sensoren lesen
    uint16_t fl = lese(sensor_FL, OFFSET_FL);
    uint16_t fm = lese(sensor_FM, OFFSET_FM);
    uint16_t fr = lese(sensor_FR, OFFSET_FR);
    uint16_t h = lese(sensor_H, OFFSET_H);

    // Fail-Counter pflegen, Auto-Recovery wenn noetig
    uint16_t werte[4] = {fl, fm, fr, h};
    uint8_t alle_tot = 0;
    for (uint8_t i = 0; i < 4; i++) {
        if (werte[i] == 9999) {
            fail_count[i]++;
            alle_tot++;
        } else {

```

XX

```

        fail_count[i] = 0;
    }
}
if (alle_tot == 4 && fail_count[0] >= 3) {
    reinitAlleSensoren();
    return;
}
for (uint8_t i = 0; i < 4; i++) {
    if (fail_count[i] >= MAX_FAIL) {
        reinitSensor(i);
    }
}

// Fahrentscheidung
const char *status      = "";
uint8_t      statuscode = STATUS_STOP;

if (fm == 9999) {
    // Frontsensor mitte ist tot - sicherheitshalber stoppen
    status = "FM TOT";
    statuscode = STATUS_FMTOT;
    alleStop();
}
else if (fm < SCHWELLE_FRONT) {
    // Frontalhindernis - Stopp, Rueckwaerts, dann zur freien Seite drehen
    status = "STOP";
    statuscode = STATUS_STOP;
    alleStop();
    delay(300);
    fahreRueckwaerts();
    delay(400);
    if (fl > fr) {
        dreheLinks();
        status = "< LINKS";
        statuscode = STATUS_LINKS;
    } else {
        dreheRechts();
        status = "RECHTS >";
        statuscode = STATUS_RECHTS;
    }
    delay(400);
}
else if (fl != 9999 && fl < SCHWELLE_SEITE) {
    // Hindernis links - leicht nach rechts korrigieren
    dreheRechts();
    status = "RECHTS >";
    statuscode = STATUS_RECHTS;
    delay(200);
}
else if (fr != 9999 && fr < SCHWELLE_SEITE) {
    // Hindernis rechts - leicht nach links korrigieren
    dreheLinks();
    status = "< LINKS";
    statuscode = STATUS_LINKS;
    delay(200);
}
else {
    // Freie Fahrt
    fahreVorwaerts();
    status = ">>>";
    statuscode = STATUS_VOR;
}

// Sample loggen wenn das Intervall erreicht ist
if (millis() - letztes_log >= LOG_INTERVAL_MS) {
    logSchreibe(fl, fm, fr, h, statuscode);
    letztes_log = millis();
}

```

```
// Live-Monitoring waehrend Kabel angesteckt ist
Serial.print(F("FL:")); Serial.print(fl);
Serial.print(F(" FM:")); Serial.print(fm);
Serial.print(F(" FR:")); Serial.print(fr);
Serial.print(F(" H:")); Serial.print(h);
Serial.print(F(" | ")); Serial.println(status);

    delay(100);
}
```

Anlage 4: Konstruktionstagebuch

Konstruktionstagebuch

Begleitaufgabe Package und Sensorik – Wahlmodul Ergonomie und Package

Fakultät Kraftfahrzeugtechnik, Westsächsische Hochschule Zwickau

Allgemeine Angaben

Gruppe / Gruppen-Nr.	
Mitglieder	
Fahrzeugparameter	Radstand: _____ mm Spurweite: _____ mm

Hinweise zur Führung des Tagebuchs

Das Konstruktionstagebuch dokumentiert den eigenen Entwicklungsprozess und bildet einen Teil der Bewertungsgrundlage. Es ist während der Bearbeitung fortlaufend zu führen, nicht erst rückblickend zu erstellen.

Iterationen werden in der Iterationstabelle festgehalten, sobald sich ein oder mehrere KPIs geändert haben. Konstruktionsentscheidungen, die nicht in den KPIs sichtbar werden (Wahl von Halterformen, Verbindungstechniken, Karosseriegestaltung usw.), gehören in den Abschnitt „Konstruktionsentscheidungen“. Die auftretenden Probleme und ihre Lösungen werden im letzten Abschnitt festgehalten.

Bei mehr als acht Iterationen, drei Entscheidungen oder drei Problemfällen sind die Tabellen entsprechend zu erweitern.

Iterationsverlauf

Mit jeder Konstruktionsiteration werden die zugehörigen KPI-Werte aus der CAD-Datei übernommen. Achslast = Hinterachsanteil; RI = Rollover-Index; Sens. ok = Anzahl der Sensoren ohne Kollision (max. 4); η = Bauraumeffizienz.

Nr.	Datum	Achslast [%]	COG_Z [mm]	RI	Sens. ok	η [%]	Bemerkung
1							
2							
3							
4							
5							
6							
7							
8							

Konstruktionsentscheidungen

Festgehalten werden konstruktive Entscheidungen mit ihrer Begründung sowie verworfenen Alternativen, sodass der Lösungsweg für Dritte nachvollziehbar bleibt.

Nr. / Datum	1 /
Komponente / Bereich	
Entscheidung	
Begründung	
Verworfen Alternativen	

Nr. / Datum	2 /
Komponente / Bereich	
Entscheidung	
Begründung	
Verworfen Alternativen	

Nr. / Datum	3 /
Komponente / Bereich	
Entscheidung	
Begründung	
Verworfen Alternativen	

Schwierigkeiten und Lösungen

Festgehalten werden Probleme, die im Verlauf der Bearbeitung aufgetreten sind, sowie der Weg zu ihrer Lösung.

Nr. / Datum	1 /
Beobachtung / Problem	
Ursache (vermutet/ermittelt)	
Lösung	

Nr. / Datum	2 /
Beobachtung / Problem	
Ursache (vermutet/ermittelt)	
Lösung	

Nr. / Datum	3 /
Beobachtung / Problem	
Ursache (vermutet/ermittelt)	
Lösung	

Anlage 5: Bewertungsbogen

Bewertungsbogen

Begleitaufgabe Package und Sensorik – Wahlmodul Ergonomie und Package
Fakultät Kraftfahrzeugtechnik, Westsächsische Hochschule Zwickau

Allgemeine Angaben

Gruppe / Gruppen-Nr.	
Mitglieder	
Abgabedatum	
Datum der Bewertung	

Quantitative Bewertung der KPIs

Die Messwerte werden aus der CAD-Datei abgelesen. Die zugehörige Bewertungsstufe wird durch Ankreuzen markiert. Punkvergabe pro KPI:

10 = optimal/sehr stabil/effizient

5 = gut/stabil/ausreichend

1 = akzeptabel/grenzwertig

0 = ungenügend/ineffizient/Kollision/nicht druckbar

KPI	Messwert	Bewertungsstufe	Pkt.
Achslastverteilung (Hinterachsanteil)	_____ %	☐ optimal (48–52 %) ☐ gut (45–55 %) ☐ akzeptabel (40–60 %) ☐ ungenügend	
Querlage Schwerpunkt (COG_Y)	_____ mm	☐ innerhalb Toleranz ☐ außerhalb Toleranz	
Schwerpunkthöhe (COG_Z)	_____ mm	☐ ≤ Oberkante Reifen (ideal) ☐ > Oberkante Reifen	
Rollover-Index (SSF)	_____	☐ Sehr stabil (> 1,5) ☐ stabil (1,2–1,5) ☐ grenzwertig (1,0–1,2) ☐ erhöhte Kippgefahr (< 1,0)	
Sichtfeldanalyse (4 Sensoren)	____/4 ok	☐ Alle Sensoren bestanden ☐ Kollisionen vorhanden (Anzahl: _____)	

KPI	Messwert	Bewertungsstufe	Pkt.
Gesamtabdeckungswinkel (Frontsensoren)	_____ °	☐ vollständige Abdeckung ☐ Lücken im Erfassungsbereich	
Überlappungsquote (Frontsensoren)	_____ %	☐ ausreichend redundant ☐ zu hohe Überschneidung ☐ keine Überlappung	
Bauraumeffizienz η	_____ %	☐ effizient ($\geq 35\%$) ☐ ausreichend ($15\text{--}35\%$) ☐ ineffizient ($< 15\%$)	
Mindestwandstärke (FDM)	min _____ mm	☐ druckbar ($\geq 1,0\text{ mm}$) ☐ nicht druckbar (Bereiche $< 1,0\text{ mm}$)	

Qualitative Bewertung

Noten von 1 bis 5 gemäß der Hochschulnotenskala.

Bewertungsbereich	Bemerkung	Note 1–5
Konstruktionstagebuch (Vollständigkeit, Nachvollziehbarkeit, Iterationsdichte)		
Funktion des Modells (autonome Hinderniserkennung, mechanische Ausführung)		
Präsentation (Argumentation, Verständlichkeit, Diskussion der Zielkonflikte)		
Konstruktive Qualität (Fertigungsgerechtigkeit, Detaillierung, Wahl der Halter)		

Gesamtergebnis

Gesamtnote	
Bemerkungen	