



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Näher dran.



Hochschule Karlsruhe – Technik und Wirtschaft
Fakultät für Maschinenbau und Mechatronik

Weiterentwicklung eines autonom fahrenden Demonstrators für Fahrerassistenzsysteme

Thesis
zur Erlangung des Grades
Master of Science (M. Sc.)

Christof Kary
geb. am 05.04.1993
in Rastatt
Matrikel-Nr.: 58516

Betreuer der Hochschule Karlsruhe
Herr Prof. Dr.-Ing. Ferdinand Olawsky

Betreuer am Arbeitsplatz
Herr Dipl.-Ing. (FH) Arthur Kessler

Flacht, 1. April 2018 bis 30. September 2018

Aufgabenbeschreibung

Eidesstattliche Erklärung

Der Verfasser erklärt, dass er die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt hat. Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Flacht, den 9. August 2018

Christof Kary

Vorwort

Danksagung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Kurzfassung

Titel der Arbeit in deutscher Sprache

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Abstract

Titel der Arbeit in englischer Sprache

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Abkürzungsverzeichnis

Kurzform	Bedeutung
μ C	Mikrocontroller
ACK	Acknowledgement
Bus	Binary Utility System
CAN	Controller Area Network
CAPL	Communication Access Programming Language
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
ECU	Electronic Control Unit
ID	Identifier
ISO	International Organization for Standardization
KWP	Keyword Protokoll
LIN	Local Interconnect Network
MOST	Media Oriented Systems Transport
OBD	On-Board-Diagnose
OSI	Open System Interconnection
SG	Steuergerät
UDS	Unified Diagnostic Services
UML	Unified Modeling Language

Nomenklatur

Zeichen	Einheit	Dimension
ϑ	C	Temperatur
ϱ	kg/m ³	Dichte
p	N/m ²	Druck

Inhaltsverzeichnis

Kurzfassung	I
Abstract	III
Abkürzungsverzeichnis	V
Nomenklatur	VII
1. Einleitung	1
1.1. Motivation	1
1.2. Zielsetzung der Arbeit	1
2. Grundlagen	3
2.1. Autonomes Fahren	3
2.1.1. Überblick Fahrerassistenzsysteme	3
2.1.2. Autonomiestufen	3
2.2. Bussysteme	4
2.2.1. Kommunikationsmodell	5
2.2.2. Controller Area Network CAN	6
2.2.3. CAN-Protokoll: Physical Layer	7
2.2.4. CAN-Protokoll: Data Link Layer	9
2.3. Hilfsmittel	12
2.3.1. Vector CANoe	12
2.3.2. Vector CANcaseXL	12
3. Ausgangssituation	15
3.1. Hardware	15
3.2. Software	15
3.3. Sensorik	16
4. Diagnosesystem	17
4.1. Konzept der Fehlerdiagnose	17
4.1.1. Anforderungen an die Fehlerdiagnose	19
4.2. Aufbau einer Datenkommunikation auf CAN-Bus	21

Inhaltsverzeichnis

4.3. Implementierung der Diagnosefunktion	21
4.4. Ergebnisbetrachtung	21
4.4.1. Test und Validierung	21
4.4.2. Mehrwert der Diagnosefunktion	21
5. Dynamische Längs- und Querregelung	23
5.1. Umsetzung des Fahralgorithmus	23
5.2. Kritische Analyse der implementierten Algorithmen	23
5.3. Optimierung der Regelung	23
5.4. Implementierung der optimierten Spurregelung	23
5.5. Ergebnisbetrachtung	23
5.5.1. Test und Validierung	23
5.5.2. Mehrwert der optimierten Spurregelung	23
6. Zusammenfassung und Ausblick	25
6.1. Zusammenfassung	25
6.2. Ausblick	26
Tabellenverzeichnis	29
Abbildungsverzeichnis	31
Literaturverzeichnis	34
A. Anhang	35
A.1. Anhang 1	35
A.2. Anhang 2	35

1. Einleitung

1.1. Motivation

Bereits seit einigen Jahren zeichnet sich durch den Einzug moderner Fahrerassistenzsysteme ein Umbruch im individuellen und gesellschaftlichen Umgang mit dem Automobil ab. Als treibende Aspekte für den technologischen Umbruch in der Automobilentwicklung sind die steigende Verkehrssicherheit, der erhöhte Fahrkomfort und eine energieoptimale Fahrzeugführung zu nennen. Alleine im Jahr 2017 kamen 3.177 Menschen bei Verkehrsunfällen auf deutschen Straßen ums Leben [Quelle]. FAS bieten Potential um Unfälle zu senken.

Autonomes Fahren könnte einen Umbruch im individuellen und gesellschaftlichen Umgang mit dem Automobil nach sich ziehen und damit auch Einfluss nehmen auf Verkehr, Mobilität oder Raumstrukturen.

1.2. Zielsetzung der Arbeit

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a

1. Einleitung

nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2. Grundlagen

Die vorliegende Schrift ist grundsätzlich als Forschungs- und Entwicklungsarbeit des automatisierten Fahrens in der Automobilbranche anzusiedeln. Es werden zunächst einige Schlüsseltechnologien der Fahrerassistenzsysteme vorgestellt und Ihre Notwendigkeit für den Entwicklungsfortschritt hin zum vollautomatisierten Fahren begründet.

2.1. Autonomes Fahren

2.1.1. Überblick Fahrerassistenzsysteme

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2.1.2. Autonomiestufen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum.

2. Grundlagen

Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2.2. Bussysteme

Die rasante Zunahme an elektronischen Systemen und *Steuergeräten* SG (englisch *Electronic Control Unit* ECU) in den letzten Jahrzehnten machten einen geregelten Datenaustausch in der Fahrzeugtechnik unerlässlich. Stetig steigende Anforderungen an Fahrtsicherheit, Motorsteuerung und Komfortsysteme erforderten zwingend einen sicheren und schnellen Informationsfluss zwischen den kommunizierenden Steuergeräten, sodass ein verteiltes und vernetztes Gesamtsystem entstand. Alle für eine jeweilige Funktion benötigten Daten konnten systemweit zur Verfügung gestellt werden. Aus dem zunehmenden Elektrifizierungsgrad ergaben sich heutige moderne Elektronikarchitekturen im Kfz in Form von seriellen Bussystemen. Das Akronym Bus stammt [1] nach von *Binary Utility System*, was auf ein drahtgebundenes Übertragungsmedium mit Anschluss an alle Systemkomponenten hinweist. Es werden komplexe Datenmengen über einzelne Leitungen bitweise übertragen, wodurch sich eine Vielzahl an Vorteilen ergibt: Der Verkabelungsaufwand sämtlicher elektrischer Leitungen wird minimiert, wodurch sich die Kosten, das Gewicht und die Fehleranfälligkeit reduzieren. Zudem wird eine Mehrfachnutzung von Informationen möglich, was die Anzahl der verbauten Sensoren senkt. Eine Diagnosefunktion wird umsetzbar und das Gesamtsystem ist flexibel für Änderungen und Erweiterungen. Die Kommunikation innerhalb eines Gesamtsystems, welches aus mehreren miteinander verknüpften Bussystemen bestehen kann, wird als *On-Board-Kommunikation* bezeichnet. Um eine übergeordnete Datenkommunikation einzelner Vernetzungsbereiche und Netzwerke zu erhalten, müssen die einzelnen Bussysteme mit unterschiedlichen Protokollen physikalisch und logisch miteinander verbunden werden. Diese Funktion wird von einem *Gateway* übernommen. Ein Gateway stellt sämtliche Daten netzwerkübergreifend zur Verfügung. Dabei kann die Funktion entweder in bereits vorhandene Steuergeräte integriert werden oder es kommen eigene zentrale oder dezentrale Gateway-Steuergeräte zum Einsatz. Bei einer *Off-Board-Kommunikation* stellt ein Gateway die Verbindung zwischen dem geschlossenen Gesamtnetz im Fahrzeug zu einem externen Gerät her [2].

Heute standardisierte und gängige Datenkommunikationssysteme sind in Tabelle 2.1 aufgeführt [2]. Auf die zur Anwendung wichtigste Form der Buskommunikation, dem *CAN-Bus*, wird an späterer Stelle näher eingegangen.

Neben dem heutzutage am häufigsten eingesetzten Bussystem (vgl. Unterabschnitt 2.2.2)

haben sich aufgrund der speziellen Anwendungsfälle und der Eigenentwicklung unterschiedlichster Hersteller, vor allem aber zur Kostenreduzierung, weitere Systeme etabliert. Die kostengünstige Variante zur seriellen Datenübertragung *Local Interconnect Network* LIN weist eine vergleichsweise geringe Datenrate auf und wird daher mittlerweile lediglich in der Komfortelektronik als Kommunikationsschnittstelle zwischen Sensorik und Aktorik verbaut. Da als physikalisches Übertragungsmedium nur eine Eindrahtleitung zum Einsatz kommt, ist das Netzwerk relativ störanfällig, was eine Verwendung in sicherheitsrelevanten Bereichen ausschließt. Eine deutlich höhere Ausfallsicherheit, aber zugleich signifikant teurere Datenübertragung liefert der sog. *FlexRay*. Aufgrund der hohen Datenraten von bis zu 10 Mbit/s bietet dieses deterministische Feldbussystem ein hohes Potential für zeit- und sicherheitskritische Anwendungsfälle [3]. Für Multimediaanwendungen im Automobilbereich hat sich der *Media Oriented Systems Transport* MOST-Bus etabliert, der als Übertragungsmedium Lichtwellenleiter verwendet und damit sehr hohe Bitraten von bis zu 150 Mbit/s ermöglicht. Ein MOST-Netzwerk ist in der Regel als Ringtopologie aufgebaut und liefert daher eine lediglich geringe Ausfallsicherheit [4].

Tabelle 2.1.: Klassifikation serieller Bussysteme [2]

Bussystem	Typische Anwendung	Maximale Datenrate	Übertragungsmedium	Sicherheitsanforderung
LIN	Komfort, Karosserie	20 kbit/s	Eindrahtleitung	gering
CAN (Low Speed)	Komfort, Karosserie	125 kbit/s	Verdrillte Zweidrahtleitung	hoch
CAN (High Speed)	Antrieb, Fahrwerk, Diagnose	1 Mbit/s	Verdrillte Zweidrahtleitung	hoch
FlexRay	Fahrwerk, X by Wire	10 Mbit/s	Verdrillte Zweidrahtleitung	sehr hoch
MOST	Infotainment	150 Mbit/s	Lichtwellenleiter	gering

2.2.1. Kommunikationsmodell

Um einen reibungslosen und nachvollziehbaren Datenaustausch zu gewährleisten, mussten mit dem Einzug der Bussysteme in der Automobilentwicklung auch einheitliche,

2. Grundlagen

herstellerübergreifende Kommunikationsstrukturen eingeführt werden. 1983 wurde der gesamte Datentransfer in einem Datennetz von der *International Organization for Standardization* ISO in sieben einzelne *Layer* (Schichten) unterteilt und die komplexe Kommunikationshierarchie beschrieben. Durch das in *DIN EN ISO/IEC 7498-1* [5] festgehaltene *Open System Interconnection* OSI-Schichtenmodell kann eine standardisierte und herstellerübergreifende Kommunikation im gesamten Busnetzwerk erzielt werden. Das OSI-Schichtenmodell wird in Tabelle 2.2 beschrieben. Für die Automobilindustrie und für Kfz-Anwendungen sind die grau hinterlegten Schichten noch nicht relevant. Wichtig sind vor allem die beiden untersten Layer *Physical* und *Data Link*. Diese Schichten werden an späterer Stelle genauer beschrieben.

Tabelle 2.2.: Zusammenfassung des OSI-Schichtenmodells aufgeteilt in Layer, Schicht und Funktionen [5]

	Layer	Schicht	Funktion
7	Application	Anwendung	Zugriff auf das Kommunikationssystem, Entkopplung Anwendung von Kommunikation
6	Presentation	Darstellung	Semantik, Datenkompression, Verschlüsselung, Übersetzer verschiedener Datenformate
5	Session	Sitzungssteuerung	Unterhalten längerer Sitzungen, Definition von Synchronisationspunkten
4	Transport	Datentransport	Verbindungsauf- und -abbau, Segmentierung, Sequenzierung, Assemblierung
3	Network	Vermittlung	Routing, Adressvergabe, Teilnehmererkennung und -überwachung
2	Data Link	Datensicherung	Botschaftsaufbau, Buszugriff, Flusskontrolle Fehlersicherung
1	Physical	Bitübertragung	Physikalische Busanbindung, Stecker, Übertragungsmedium, Leitungscodierung

2.2.2. Controller Area Network CAN

Das Bussystem *Controller Area Network* CAN wurde erstmals in den 1980er Jahren von der *Robert Bosch GmbH* präsentiert und gilt seit 1994 als offener Industriestandard. Mit der ISO-Norm *ISO 11898* wurde die CAN-Spezifikation international vereinheitlicht. Heute stellt CAN durch seine hohe Datenübertragungsrate und der geringen

Fehleranfälligkeit die am weitesten verbreitete Kommunikationsspezifikation in der Automobilindustrie dar, kommt jedoch auch häufig in industriellen Anwendungen zum Einsatz. Aufgrund der daraus resultierenden hohen Stückzahlen an CAN-Controllern ergibt sich ein stetig sinkender Stückpreis für die zugehörigen Steuergeräte, was als weitere Stärke dieses Bussystems zu zählen ist. Die Steuergeräte, oder auch *Bus-Knoten* bezeichnet, sind in einem CAN-Netzwerk üblicherweise in Form einer Linientopologie nach der *Multi-Master*-Architektur miteinander verbunden. Jeder Knoten ist berechtigt, den Datentransfer auf dem Bus ereignisgesteuert anzustoßen [6, 7].

2.2.3. CAN-Protokoll: Physical Layer

Die unterste Schicht im OSI-Modell beschreibt die physikalische Busankopplung. Das Übertragungsmedium des CAN-Busses wird in den häufigsten Fällen als verdrehte Zweidrahtleitung, als sog. *Twisted-Pair-Leitung*, ausgeführt, wodurch sich die magnetischen Felder der beiden Leitungen weitestgehend gegenseitig neutralisieren. Eine hohe Datenrate und Busauslastung können zu Reflexionen im Bussystem führen. Um diesen unerwünschten Effekt zu minimieren, müssen die Enden der Busleitung mit einem Abschlusswiderstand versehen werden. Neben der physikalischen Busleitung kommen hardwareseitig weitere Bauteile wie der *Mikrocontroller* (μC), der CAN-Controller und der CAN-Transceiver zum Einsatz. Der Mikrocontroller verarbeitet die Kommunikationsdienste der höheren OSI-Schichten in der Kommunikationssoftware. Die grundlegenden Funktionen sind hingegen in den restlichen Bauteilen implementiert. Der CAN-Controller wickelt das Protokoll ab, während der Transceiver die physikalische Verbindung zum Übertragungsmedium herstellt. Der prinzipielle Aufbau eines CAN-Netzwerks wird in Abbildung 2.1 verdeutlicht.

Die physikalische Signalübertragung in einem CAN-Netzwerk basiert auf der Übertragung von Spannungsdifferenzen zwischen der *CAN-High*-Leitung (CANH) und der *CAN-Low*-Leitung (CANL). Wie in Tabelle 2-1 dargestellt, unterscheidet die ISO-Norm zwischen dem *Low-Speed-CAN* (Class B) und dem *High-Speed-CAN* (Class C). Der Low-Speed-CAN zeichnet sich durch seine auf 125 kbit/s begrenzte Datenrate aus. Dadurch findet er häufig in Komfortsystemen wie Klimasteuergeräten Anwendung. Die Signale werden über nominelle Potentiale auf dem Bus übertragen. Beim High-Speed-CAN hingegen werden differentielle Potentiale verwendet. Dieser besitzt eine maximale Datenrate von bis zu 1 Mbit/s und eignet sich daher für zeitkritische Anwendungen wie Antriebs- und Fahrdynamikregelung. Die unterschiedlichen Signalpegel werden in Abbildung 2.2 erläutert. Aufgrund der Busankopplung ermöglicht der Low-Speed-CAN zusätzliche Mechanismen zur Fehlererkennung. Bei Ausfall einer Leitung bleibt er betriebsfähig und gilt daher gegenüber dem High-Speed-CAN als fehlertoleranter [7, 8].

2. Grundlagen

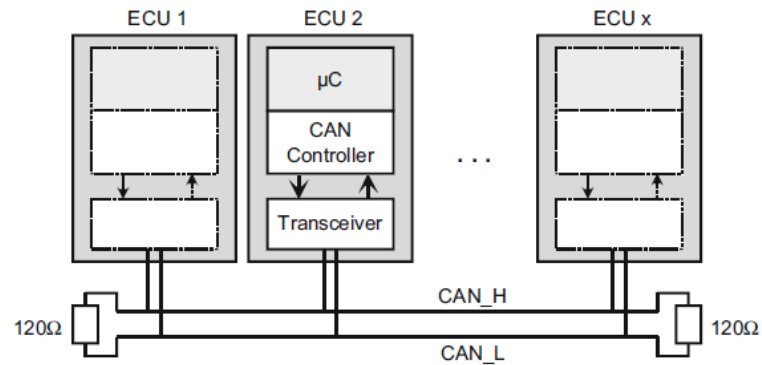


Abbildung 2.1.: CAN-Netzwerk: Ein einzelner CAN-Knoten besteht aus einem Mikrocontroller, einem CAN-Controller und einem CAN-Transceiver. Der Abschlusswiderstand unterdrückt Busreflexionen [7].

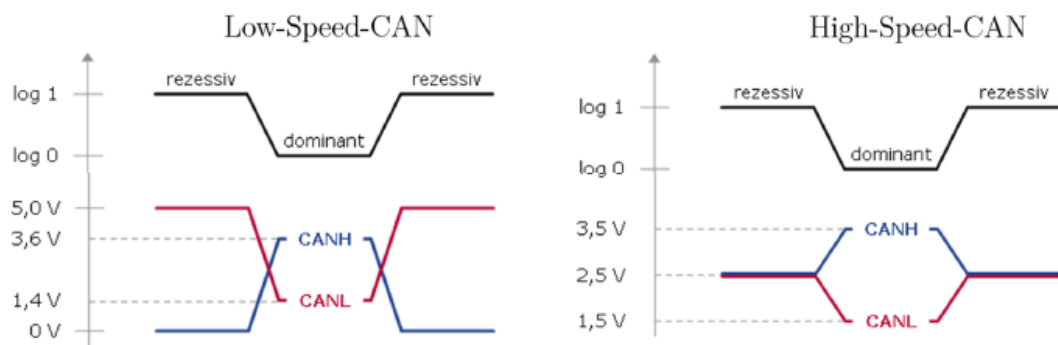


Abbildung 2.2.: Signalpegel Low-Speed-CAN (links) und High-Speed-CAN (rechts) [8].

2.2.4. CAN-Protokoll: Data Link Layer

Der *Data Link Layer* beschreibt das Zugriffsverfahren und den strukturellen Aufbau eines CAN-Frames. Unter einem Frame versteht man den gesamten Datenrahmen, der in einer einzelnen Botschaft über den CAN-Bus übermittelt wird. Zwischen den Begriffen Frame, Botschaft und Nachricht wird nachfolgend keine Unterscheidung getroffen. Typischerweise gibt es mehrere Arten von CAN-Frames, die in verschiedene Kategorien unterteilt werden:

- *Data Frame* zur Übertragung von Nutzdaten.
- *Remote Frame* zur Anforderung von Nutzdaten (also Data Frames) von beliebigen CAN-Knoten. Setzt sich bis auf das fehlende Data Field wie ein Data Frame zusammen.
- *Error Frame* zur Signalisierung entdeckter Fehler während des Kommunikationsbetriebs. Mit dem Übertragen eines Error Frames geht der Abbruch der laufenden Botschaftsübertragung einher.

Das Botschaftenformat eines Data Frames ist in Abbildung 2.3 dargestellt [8]. Die einzelnen Komponenten und deren Aufgaben werden nachstehend in der zugehörigen Tabelle 2.3 näher erklärt.

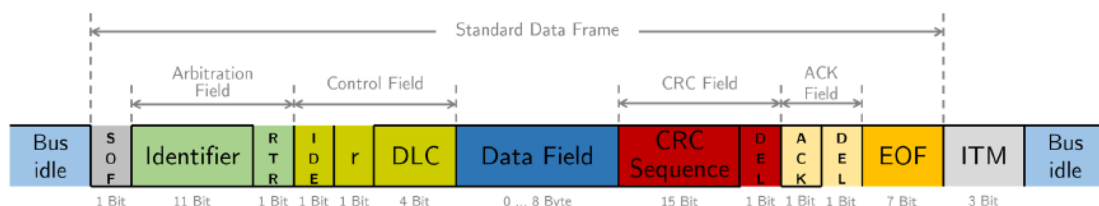


Abbildung 2.3.: Aufbau des Standard CAN Data-Frames [8].

Während der gesamten Laufzeit der Netzwerkverbindung senden die Busknoten ihre Botschaften nach dem *Broadcast*-Verfahren unaufgefordert und ereignisgesteuert auf den Bus. Jeder Empfänger einer Botschaft entscheidet, ob der Inhalt relevant ist und er damit die Nachricht auswertet oder ignoriert. Diese *Akzeptanzfilterung* findet auf Basis der inhaltsbasierten Adressierung eines Frames statt. Hierzu kennzeichnet der *Identifier* ID den spezifischen Inhalt einer Botschaft. Der Identifier ist in einem herkömmlichen CAN-Protokoll 11 Bit lang. Somit lassen sich bis zu $2^{11} = 2.048$ Botschaften eindeutig unterscheiden. Aufgrund der stetig wachsenden Komplexität moderner Busarchitekturen besteht zudem die Möglichkeit, einen Identifier im Extended-Format aufzubauen. Der ID umfasst dann 29 Bit, wodurch $2^{29} = 536.870.912$ unterschiedliche Botschaften differenziert werden können.

2. Grundlagen

Tabelle 2.3.: Funktionen der einzelnen Felder im Data-Frame [6, 7].

Feld	Name	Länge	Funktion
kein Feld	SOF	1 Bit	Der dominante <i>Start of Frame</i> signalisiert den Start eines Frames. Durch den Wechsel von einem rezessiven (Buspegel im Ruhezustand) auf ein dominantes Bit findet eine netzwerkweite Synchronisation statt.
Arbitration Field	Identifier	11 Bits	Der Identifier dient der bitweisen Arbitrierung und beschreibt die logische Adressierung der Botschaft.
	RTR	1 Bit	Der <i>Remote Transmission Request</i> kennzeichnet den Frametyp. Im Fall des Data Frames wird das Bit dominant übertragen.
Control Field	IDE	1 Bit	Die <i>Identifier Extension</i> legt fest, ob der Identifier im Standard-Format (11 Bit) oder im Extended-Format (29 Bit) vorliegt.
	r	1 Bit	Dieses Bit ist für künftige Erweiterungen reserviert.
	DLC	4 Bits	Der <i>Data Length Code</i> übermittelt die Anzahl der nachfolgenden Nutzbytes.
Data Field	Data Field	0...64 Bits	Im Datenfeld werden bis zu acht Bytes an Nutzdaten übertragen. Es beinhaltet die zu übermittelnden Signalwerte und Nutzinformationen.
CRC-Feld	CRC	15 Bits	Der <i>Cyclic Redundancy Check</i> bildet eine Prüfsumme der Nutzdaten und dient der Fehlererkennung bei der Botschaftsübertragung.
	DEL	1 Bit	Auf das CRC-Feld folgt ein rezessives <i>Delimiter</i> -Bit.
ACK-Feld	ACK	1 Bit	Der Sender einer Botschaft setzt das <i>Acknowledgment</i> -Bit auf rezessiv, der Empfänger quittiert das korrekte Ergebnis des CRC-Feldes mit einem dominanten Bit.
	DEL	1 Bit	Das ACK-Feld wird ebenfalls durch ein <i>Delimiter</i> -Bit begrenzt.
kein Feld	EOF	7 Bit	Der <i>End of Frame</i> markiert das Ende einer Botschaft mit sieben rezessiven Bits.

Durch den ereignisgesteuerten Kommunikationsaufbau kann es häufig vorkommen, dass mehrere Busteilnehmer eine Botschaftsübertragung zum gleichen Zeitpunkt beginnen möchten. Da eine Übertragung aber nicht unterbrochen werden kann und oftmals essentielle sicherheitsrelevante Nachrichten unmittelbar mit vergleichsweise marginalen Botschaften versendet werden, ist es von hoher Bedeutung, sämtliche Botschaften mit Prioritäten zu versehen. Es wird das *Carrier Sense Multiple Access with Collision Avoidance* CSMA/CA-Verfahren angewendet. Durch eine *bitweise Arbitrierung* wird dafür gesorgt, dass bei einer Kollision auf dem Bus die Botschaft mit der höchsten Priorität weiterhin übertragen wird. Die höchst priorisierte Botschaft weist immer den niedrigsten nominalen ID auf. Durch den Vergleich der dominanten und rezessiven Buspegel ist jeder Knoten darüber informiert, welche Botschaft gerade auf dem Bus übertragen wird. Stellt ein Knoten fest, dass der Pegel auf dem Bus nicht seiner gesendeten Botschaft entspricht, stellt dieser umgehend den Sendevorgang ein. Die Echtzeitfähigkeit des Gesamtsystems wird somit nicht gefährdet und Botschaften mit hoher Priorität bleiben deterministisch.

Neben einer korrekten Zugriffsfolge sind besonders die Datensicherung und eine zuverlässige Fehlererkennung von großem Stellenwert. Hierzu kommen Mechanismen wie der *Cyclic Redundancy Check* CRC, *Acknowledgement* ACK oder *Bitstuffing* zum Einsatz. In der CRC-Sequenz wird aus dem Frame ein Polynom gebildet, welches durch Division mit einem Modulo-Operator eine Prüfsumme ergibt. Der Empfänger berechnet bei Erhalt einer Botschaft ebenfalls eine CRC-Prüfsumme. Stimmen die erhaltenen und berechneten Sequenzen überein, war der strukturelle Aufbau der Botschaftsübertragung korrekt. Mindestens ein Empfänger bestätigt den positiven Erhalt, indem er das ACK-Bit auf einen dominanten Pegel setzt. Liegt ein ACK-Fehler vor, wurde also entweder vom Sender ein Fehler verursacht oder es befinden sich keine weiteren Teilnehmer am Bus.

Die Bitstuffing-Methodik dient der zeitlichen Nachsynchronisation während der Übertragungslaufzeit. Die Synchronisation der Busteilnehmer erfolgt eigentlich über den Flankenwechsel zwischen unterschiedlichen Bits. Liegen jedoch mehrere homogene Bits hintereinander vor, kann diese Synchronisation nicht stattfinden. Um eine korrekte Übertragung zu gewährleisten, fügt der Sender bis zum Ende der CRC-Sequenz nach fünf homogenen Bits immer ein komplementäres Bit ein. Der Empfänger entfernt seinerseits diese Stuff-Bits, bevor das empfangene Datenpaket ausgewertet wird. Wird von einem Empfänger eine Folge von mehr als fünf homogenen Bits erkannt, liegt ein Bitstuffing-Fehler vor. Bei der Übertragung eines Data Frames mit 11 Bit Identifier kann der gesamte Botschaftsrahmen also um 29 Bit¹ auf bis zu 132 Bit anwachsen.

Wurde einer der genannten Fehler erkannt, wird ein Fehlersignal gesetzt, welches aus sechs dominanten Bits besteht und damit bewusst gegen die Bitstuffing-Regel verstößt.

¹im Worst Case bei acht Datenbytes

2. Grundlagen

In diesem Fall wird ein aktives *Error Flag* in einem Error Frame von dem Knoten versendet, welcher den Sendefehler zuerst detektiert hat. Die beteiligten CAN-Knoten stellen den Sendevorgang umgehend ein. Wird ein Fehler eines Knoten mehrfach detektiert, kann diesem das Senden verweigert werden. Zur Sicherstellung der systemweiten Datenkonsistenz und zur Reduzierung der Busauslastung kann ein fehlerhafter Knoten vollständig von der Buskommunikation ausgeschlossen werden [6, 7, 8].

2.3. Hilfsmittel

In diesem Abschnitt werden die wichtigsten Hilfsmittel vorgestellt, die während der Erstellung der vorliegenden Arbeit und den nachfolgend beschriebenen Abschnitten genutzt wurden.

2.3.1. Vector CANoe

CANoe ist ein vielseitiges Software-Werkzeug der Firma *Vector Informatik GmbH*. Es bedient Anwendungsgebiete wie Analyse, Diagnose, Simulation, Stimulation und Test während dem gesamten Entwicklungsprozess von Steuergeräten und Netzwerken. *CANoe* unterstützt dabei sämtliche gängigen Bussysteme sowie die normierten Transport- und Diagnoseprotokolle. Neben der Überwachung des realen Busverkehrs bietet die Software die Möglichkeit, einfache Teilsysteme bis hin zu komplexen Restbussystemen zu simulieren. Dabei kann die Buskommunikation jederzeit textuell dargestellt und die Signale visuell veranschaulicht werden [9].

Das Softwaretool liefert zusätzlich die integrierte und firmeneigene Programmiersprache *Communication Access Programming Language* CAPL. Die auf *C* basierte Sprache wurde speziell für die Anforderungen zur Entwicklung von Bussystemen angepasst. CAPL bietet als eventorientierte Skriptsprache die Möglichkeit, bequem auf Signale zuzugreifen und die Signalwerte zu verändern. Es steht eine Vielzahl an vordefinierter Funktionen zur Verfügung. Ebenso können eigene grafische Benutzeroberflächen erstellt werden, wodurch komplexe Simulationsumgebungen erleichtert werden [10].

2.3.2. Vector CANcaseXL

Das *CANcaseXL* der Firma *Vector Informatik GmbH* ist ein USB-Interface, das eine physikalische Ankopplung eines Computers mit der Software *CANoe* an ein reales Bussystem erlaubt. Mithilfe dieser Ankopplung ist es möglich, sowohl Busbotschaften zu generieren und diese auf den Bus zu senden, als auch reale Botschaften von angekoppelten Systemen zu auslesen und auszuwerten. Über die beiden D-Sub Anschlüsse ist eine

Verbindung zu CAN- oder LIN-Netzwerken realisierbar. Zwar wurde das CANcaseXL inzwischen von neueren Modulen abgelöst und bietet lediglich einen eingeschränkten Funktionsumfang gegenüber der weiteren Produktpalette der *Vector Informatik GmbH*, jedoch eignet es sich aufgrund der vergleichsweise kompakten Bauform besonders für die in dieser Arbeit beschriebenen Anwendung. Zudem ist der integrierte Funktionsumfang für den Einsatz als CAN-Interface völlig ausreichend [11].

3. Ausgangssituation

3.1. Hardware

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.2. Software

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3. Ausgangssituation

3.3. Sensorik

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4. Diagnosesystem

Dieser Kapitel soll detailliert die zentrale Aufgabe der Arbeit beschreiben. Es solle Verfahren zu Diagnose des EVObots entwickelt und erstellt werden, um im Fahrbetrieb die internen Zustands- und Sensordaten echtzeitnah darzustellen und somit die Applikation des Fahralgorithmus zu erleichtern. Zunächst wird die Bedeutung der Diagnose eines komplexen Fahrzeugsystems mit vernetzten Systemen aufgezeigt und ein Konzept zur Fehlerdiagnose am Modellfahrzeug aufgebaut. Nachfolgend wird das Vorgehen und der Aufbau eines Diagnosesystems ausführlich beschrieben. Abschließend wird die Diagnosefunktion validiert und die gewonnen Ergebnisse betrachtet.

4.1. Konzept der Fehlerdiagnose

Aktuelle Fahrzeugsystemen sollten selbstredend stets fehlerfrei und ohne Probleme agieren und die vom Fahrer gewünschten Funktionen vollständig umsetzen. Trotzdem kann es unter gewissen Umständen und äußeren Voraussetzung zu einem Fehlverhalten kommen. Ein Fehler kann im Betrieb eines mechatronischen Systemverbund sowohl mechanisch, elektronisch, als auch softwareseitig auftreten. Durch den stetigen Anstieg des Softwareanteils und der sicherheitskritischen Prozesse im automotiven Bereich steigt die Gefahr eines Softwarefehlers oder einer Software-Anomalie. Ein Fehler wird dabei allgemein nach *EN ISO 9000:2005* [12] als „Nichterfüllung einer Anforderung“ oder nach *DIN 55350* [13] genauer als „eine unzulässige Abweichung eines Merkmals von einer vorgegebenen Forderung“ beschrieben.

Ein Softwarefehler lässt sich allgemeingültig entweder durch physikalische beziehungsweise chemische Effekte oder durch menschliche Fehlermechanismen wie Denkfehler, Verständnisfehler, Interpretationsfehler oder simple Tippfehler beschreiben. Dabei kann man diese Beschreibung eines Fehlers grundsätzlich in zwei Kategorien einordnen: Man spricht von einem physikalischen Fehler, wenn einzelne Komponenten oder Teilsysteme ausfallen, oder von einem funktionalen Fehler, wenn ein System ausführbar ist, seine Funktion aber nicht korrekt umgesetzt wird [14]. Ein funktionaler Fehler ist offenbar in den meisten Fällen das Ergebnis von Design-Mängeln und demnach auf eine menschliche Ursache zurückzuführen. Eine Fehlhandlung in der Umsetzung einer Softwareanwendung

4. Diagnosesystem

führt in aller Regel zu einem Fehlerzustand bei der Programmausführung. Dieser Fehlerzustand kann unter Umständen auch unerkannt bleiben. Tritt der Fehlerzustand jedoch aus Programmsicht nach Außen auf, spricht man von der Fehlerwirkung, die für den Anwender wahrnehmbar ist. Diese Fehlerwirkung kann sich je nach Schwere des Fehlers als ein abweichender Rückgabewert einer Berechnung oder bis hin zum Totalausfall des Systems äußern [15].

Im Sinne der Zuverlässigkeit gilt es also, das Auftreten eines Fehlerzustandes in einem sicherheitskritischen System zu vermeiden oder besser einen sicheren Zustand im Fehlerfall einzunehmen. Ein übergeordnetes Beispiel eines *Fail-Safe*-Zustandes kann aus der Technologie des hochautomatisierten Fahrens gegeben werden: Fällt das Kamerasystem zur Fahrspurerkennung unerwartet aus, soll ein automatisiert fahrendes Fahrzeug nicht unkontrolliert weiterfahren, sondern die Fahrgeschwindigkeit reduzieren und auf Basis der verfügbaren Streckendaten am rechten Fahrbahnrand zum Stehen kommen, um einen potentiellen Schaden zu minimieren.

Die Komplexität der vernetzten Funktionen und Systeme erfordert eine umfassende Kommunikation der Softwarekomponenten und ebenso ausgereifte Methoden zur Diagnose möglicher Fehler. Kommt es zu einem Fehlerfall oder Systemausfall, ist die Ursache in einem komplexen und verteilten Systemnetzwerk meist schwer auszumachen. Aufgrund dessen wurden bereits früh in der Entwicklung elektronischer Datenkommunikation Diagnosesysteme als Analysewerkzeug eingeführt. Mit einem aus Hard- und Software bestehenden Diagnosesystem kann die Datenbuskommunikation aufgezeichnet und diagnoserelevante Informationen über den Zustand der Teilkomponenten zu einem externen Testgerät übertragen und ausgewertet werden. Ein Diagnosesystem gilt damit als umfangreiches Werkzeug zur schnellen Fehlererkennung und Fehleranalyse. Während dem Entwicklungsprozess lässt sich ein Diagnosesystem auch nutzen, um über die Diagnosekommunikation die Steuergeräte-Applikation durchzuführen. Dabei ist es nicht nur nötig die reine Datenübertragung einheitlich zu standardisieren, sondern ebenfalls die Applikationsschicht der Protokolle (Vgl. Tabelle 2.2) zu normieren. Dies bietet die Möglichkeit, den Aufwand und die Pflege der Diagnoseschnittstellen und den Diagnose-testern zu begrenzen.

Um eine einheitliche Diagnosekommunikation zu schaffen, wurden seit den 1990er Jahren diverse *Diagnoseprotokolle* entwickelt, die zunächst teils proprietär und inkompatibel zu einander umgesetzt waren, inzwischen jedoch meist herstellerübergreifend genormt sind. Als heute gängige Diagnoseprotokolle sind das *Keyword 2000 Protokoll* KWP 2000, die *Unified Diagnostic Services* UDS oder die *On-Board-Diagnose* OBD zu nennen. Für weiterführende Informationen der genannten Standards sei auf [7] und [16] verwiesen.

4.1.1. Anforderungen an die Fehlerdiagnose

Um ein Diagnosesystem für das Demonstratorfahrzeug mit angemessenen Mitteln und Werkzeugen umzusetzen, wird zunächst der gewünschte Funktionsumfang beschrieben und darauf aufbauend sämtliche Anforderungen an das System definiert. Dazu ist in Abbildung 4.1 ein Use-Case-Diagramm in der grafischen Modellierungssprache UML für den konkreten Anwendungsfall der Diagnosefunktion dargestellt. Hieraus lassen sich leicht die Spezifikationen und geforderte Funktionalitäten an die spätere Diagnose ableiten.

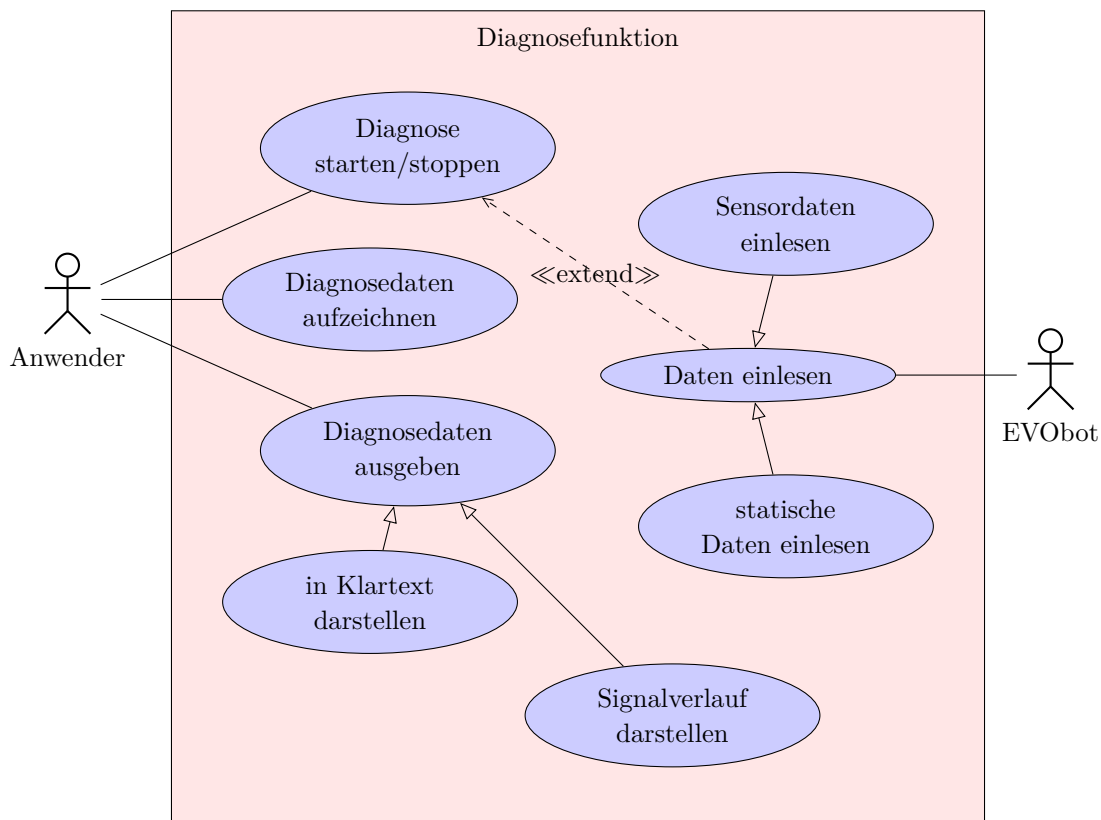


Abbildung 4.1.: Use-Case-Diagramm einer Diagnosefunktion für den Fahrdemonstrator

Das Gesamtsystem besteht aus Fahrzeug, Diagnosewerkzeug und Anwender. Aus dem Use-Case-Diagramm ergeben sich die funktionalen Anforderungen wie folgt:

- Das Diagnosesystem muss die Diagnosedaten innerhalb der Programmlaufzeit an ein Ausgabegerät übermitteln.
- Das Diagnosesystem muss die Diagnosedaten auf einem externen Ausgabegerät darstellen.

4. Diagnosesystem

- Das Diagnosesystem muss die Daten sowohl in Klartext ausgeben, als auch die einzelnen Signalverläufe grafisch darstellen.
- Die Diagnosefunktion muss vom Anwender gestartet und gestoppt werden können.
- Das Diagnosesystem darf im inaktiven Zustand keine Daten vom Fahrzeug übertragen, um die Systemauslastung gering zu halten.

Zudem lassen sich weiterhin folgende, nichtfunktionale Anforderungen stellen:

- Die Umsetzung der Datenkommunikation soll dem Vorbild eines realen und aktuellen Kraftfahrzeuges entsprechen.
- Die Datenkommunikation darf kabelgebunden oder kabellos stattfinden.
- Das Diagnosesystem muss ein echtzeitnahes Daten-Monitoring ermöglichen.
- Das Diagnosesystem soll eine übersichtliche Visualisierung für den Anwender bieten.
- Die Einbindung in das bestehende System soll ohne grundlegende Modifikationen in den implementierten Algorithmen möglich sein.
- Die Datenkonsistenz und Datensicherung bei der externen Kommunikation muss gewährleistet sein.
- Die Diagnosefunktion muss für spätere Ergänzungen erweiterbar sein.
- Die Programmierschnittstelle muss offen und modular sein.

4.2. Aufbau einer Datenkommunikation auf CAN-Bus

Echtzeitnah, Livebeurteilung und Aufzeichnung möglich, visuell gut umsetzbar, Aufwand überschaubar, Einbindung in bestehendes System nur durch Erweiterung, vergleichbare Umsetzung wie im Kfz, Datenkonsistenz und Zuverlässigkeit (Sicherheitsaspekte von CAN), kabellos oder kabelgebunden (keine langen Fahrwege), leichte Erweiterung

→ Anforderungen definieren → Möglichkeiten zur Umsetzung (großer Vorteil CANoe vorhanden und kann Inhaus genutzt werden) →

Off-Board-Kommunikation als Diagnose in ISO Schicht 7 geregelt

Generierung von Diagnosedaten ist von Beginn der Entwicklung an essentiell, um spätere Komplexität zu beherrschen.

Fehler sind üblich in komplexen Systemen. Sollten vermieden werden, kommen aber unter bestimmten Bedingungen und Voraussetzungen vor. Abhängig von Implementierung, Fehlervermeidung usw. Fale-Safe-Zustand. Fehlerzustand, Fehlerwirkung Wie äußert sich Fehler oder Fehlfunktion? → unplausibler Signalwert, falscher Wert oder out of range

Abweichung zum SOLL ist nicht unbedingt programmseitiger Fehler, sondern kann aus Logikfunktion des Programmierers folgen.

4.2. Aufbau einer Datenkommunikation auf CAN-Bus

4.3. Implementierung der Diagnosefunktion

4.4. Ergebnisbetrachtung

4.4.1. Test und Validierung

4.4.2. Mehrwert der Diagnosefunktion

5. Dynamische Längs- und Querregelung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.1. Umsetzung des Fahralgorithmus

5.2. Kritische Analyse der implementierten Algorithmen

5.3. Optimierung der Regelung

5.4. Implementierung der optimierten Spurregelung

5.5. Ergebnisbetrachtung

5.5.1. Test und Validierung

5.5.2. Mehrwert der optimierten Spurregelung

6. Zusammenfassung und Ausblick

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

6.1. Zusammenfassung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6. Zusammenfassung und Ausblick

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

6.2. Ausblick

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante.

Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Tabellenverzeichnis

2.1. Klassifikation serieller Bussysteme [2]	5
2.2. Zusammenfassung des OSI-Schichtenmodells aufgeteilt in Layer, Schicht und Funktionen [5]	6
2.3. Funktionen der einzelnen Felder im Data-Frame [6, 7].	10

Abbildungsverzeichnis

2.1. CAN-Netzwerk: Ein einzelner CAN-Knoten besteht aus einem Mikrocontroller, einem CAN-Controller und einem CAN-Transceiver. Der Abschlusswiderstand unterdrückt Busreflexionen [7].	8
2.2. Signalpegel Low-Speed-CAN (links) und High-Speed-CAN (rechts) [8]. . .	8
2.3. Aufbau des Standard CAN Data-Frames [8].	9
4.1. Use-Case-Diagramm einer Diagnosefunktion für den Fahrdemonstrator . .	19

Literatur

- [1] Niels Klußmann. *Lexikon der Kommunikations- und Informationstechnik. Telekommunikation, Datenkommunikation, Multimedia, Internet*. 2., erw. und aktualisierte Aufl. Heidelberg: Hüthig, 2000.
- [2] Vector Informatik GmbH. *Einführung in die Seriellen Bussysteme im Kfz*. Copyright: 2018 Vector Informatik. URL: https://elearning.vector.com/vl_sbs_introduction_de.html (geprüft am 08.06.2018).
- [3] Vector Informatik GmbH. *Einführung in CAN*. Copyright: 2018 Vector Informatik. URL: https://elearning.vector.com/vl_can_introduction_de.html (geprüft am 08.06.2018).
- [4] Daniel Schüller. “Bussysteme im Automobil. Ausarbeitung zum Seminarvortrag”. Koblenz: Universität Koblenz-Landau, 2005-01-20. (Geprüft am 08.06.2018).
- [5] International Organization for Standardization, Hrsg. *ISO/IEC 7498-1: Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*. ISO/IEC. Genf, CH: ISO/IEC Copyright Office, 15. Nov. 1994. URL: [http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip) (geprüft am 08.06.2018).
- [6] Henning Wallentowitz und Konrad Reif, Hrsg. *Handbuch Kraftfahrzeugelektronik. Grundlagen - Komponenten - Systeme - Anwendungen*. ger. 2., verb. und aktualisierte Aufl. ATZ/MTZ-Fachbuch. Wiesbaden: Vieweg + Teubner, 2011. 724 S.
- [7] Werner Zimmermann und Ralf Schmidgall. *Bussysteme in der Fahrzeugtechnik. Protokolle, Standards und Softwarearchitektur*. 5., aktual. und erw. Aufl. ATZ / MTZ-Fachbuch. Wiesbaden: Springer Vieweg, 2014. 507 S. URL: <http://dx.doi.org/10.1007/978-3-658-02419-2>.
- [8] Vector Informatik GmbH. *Einführung in FlexRay*. Copyright: 2018 Vector Informatik. URL: https://elearning.vector.com/vl_flexray_introduction_de.html (geprüft am 08.06.2018).

Literatur

- [9] Vector Informatik GmbH. *CANoe. Produktinformation*. 2018. URL: https://vector.com/portal/medien/cmc/info/CANoe_ProductInformation_DE.pdf (geprüft am 08.06.2018).
- [10] Vector Informatik GmbH. *CAPL Documentation. Vector KnowledgeBase*. Vector Informatik GmbH. 7.08.2017. URL: <https://kb.vector.com/entry/48/> (geprüft am 08.06.2018).
- [11] Vector Informatik GmbH. *Handbuch CANcaseXL*. 2015. URL: https://vector.com/portal/medien/cmc/manuals/CANcaseXL_Manual_DE.pdf (geprüft am 08.06.2018).
- [12] DIN Deutsches Institut für Normung e.V., Hrsg. *Qualitätsmanagementsysteme*. DIN EN ISO. Version 2005-12. Berlin: DIN Deutsches Institut für Normung e.V., 1. Nov. 2015. URL: <http://perinorm-fr.redi-bw.de/volltexte/CD21DE07/2325650/2325650.pdf?> (geprüft am 13.07.2018).
- [13] DIN Deutsches Institut für Normung e.V., Hrsg. *Begriffe zum Qualitätsmanagement*. DIN. Version 11. Berlin: DIN Deutsches Institut für Normung e.V., 1. Mai 2008. URL: <http://perinorm-fr.redi-bw.de/volltexte/CD21DE03/1415055/1415055.pdf?> (geprüft am 13.07.2018).
- [14] Josef Börcsök. *Elektronische Sicherheitssysteme. Hardwarekonzepte, Modelle und Berechnung*. 2., überarbeitete Auflage. Praxis. Heidelberg: Hüthig GmbH & Co. KG, 2007. 608 S. URL: http://deposit.d-nb.de/cgi-bin/dokserv?id=2995265&prov=M&dok_var=1&dok_ext=htm.
- [15] ISTQB AISBL, German Testing Board e.V. *ISTQB GTB Standardglossar der Testbegriffe. Version vom 21.05.2017*. 2017. URL: <http://glossar.german-testing-board.info/> (geprüft am 20.07.2018).
- [16] Florian Schäffer. *OBD - Fahrzeugdiagnose in der Praxis*. ger. Elektronik. Haar bei München: Franzis, 2012. 240 S.

A. Anhang

A.1. Anhang 1

A.2. Anhang 2