Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Patrick Kuncke

Erprobung von Echtzeit Ethernet basierten Automobil-Gateways in einem Prototypfahrzeug

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Patrick Kuncke

Erprobung von Echtzeit Ethernet basierten Automobil-Gateways in einem Prototypfahrzeug

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

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

Betreuender Prüfer: Prof. Dr. Franz Korf
Zweitgutachter: Prof. Dr. Wolfgang Fohl

Eingereicht am: 25. April 2016

Patrick Kuncke

Thema der Arbeit

Erprobung von Echtzeit Ethernet basierten Automobil-Gateways in einem Prototypfahrzeug

Stichworte

Echtzeit-Ethernet, Bussysteme, Kommunikationsgateway, verteilter CAN Bus, Fahrzeug-Netzwerke, CAN, time-triggered, TTEthernet

Kurzzusammenfassung

Echtzeit-Ethernet wird schon von vielen Automobil Herstellern selektiv eingesetzt und ist ein vielversprechender Kandidat, um die Kommunikation in zukünftigen Automobilen zu verbessern. An der Hochschule für Angewandte Wissenschaften Hamburg gibt es bereits ein Protoypfahrzeug mit eingebautem Real-Time Ethernet (RTE) Backbone Netzwerk. Diese Arbeit untersucht die Auswirkungen beim Einsatz von RTE Gateways im Prototypfahrzeug, welche für eine sanfte Migrationsstrategie CAN-Nachrichten via RTE Backbone senden. Die Entwicklung eines Konzeptes zum Senden von CAN Nachrichten via des RTE Backbone Netzwerks wird als erster Schritt durchgeführt, welches im Anschluss dann realisiert wird. Dazu werden die bestehenden RTE Gateways im Prototypfahrzeug neu konfiguriert und weiterentwickelt um CAN Nachrichten via RTE senden zu können. Anschließend folgen Messungen sowie eine Auswertung der Messdaten, um die Nachrichtenlaufzeit via RTE Backbone Netzwerk beurteilen zu können.

Title of the paper

Testing of real-time ethernet based automotive gateways in a prototype vehicle

Keywords

Real-time Ethernet, field bus, communication gateway, distributed CAN bus, In-vehicle network, can, time-triggered, TTEthernet

Abstract

Realtime Ethernet is widely used by automotive manufacturers for selected parts and is a promising candidate to improve the automotive communication networks in upcoming cars. At the University of Applied Sciences Hamburg already exists a car with integrated Realtime Ethernet backbone. This thesis investigates the impact of using Realtime Ethernet Gateways in the prototype vehicle which are sending CAN messages over the Realtime Ethernet backbone. The first step is to develop a concept to send CAN messages via the Realtime Ethernet backbone which will afterwards be realized. To achieve this the Realtime Ethernet gateways have to be reconfigured and further developed to send CAN messages via Realtime Ethernet. The final step will be to measure and evaluate the measured data to form an opinion over the transmit time via Realtime Ethernet.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

In aktuellen Automobilen sind mittlerweile, durch ihren hohen Anteil an elektronischen Komponenten, sehr komplexe, verteilte System entstanden. Mit jeder neuen Automobilgeneration wächst das System noch weiter und so haben wir bereits heutzutage ein System im Automobil, welches aus mehr als 70 Steuergeräten/ECU besteht. Diese Steuergeräte kommunizieren untereinander mit Hilfe von ca. 2500 Nachrichten, welche über ein zentrales Kommunikationsgateway geschickt werden (Marscholik und Subke (2007);Navet u. a. (2005)).

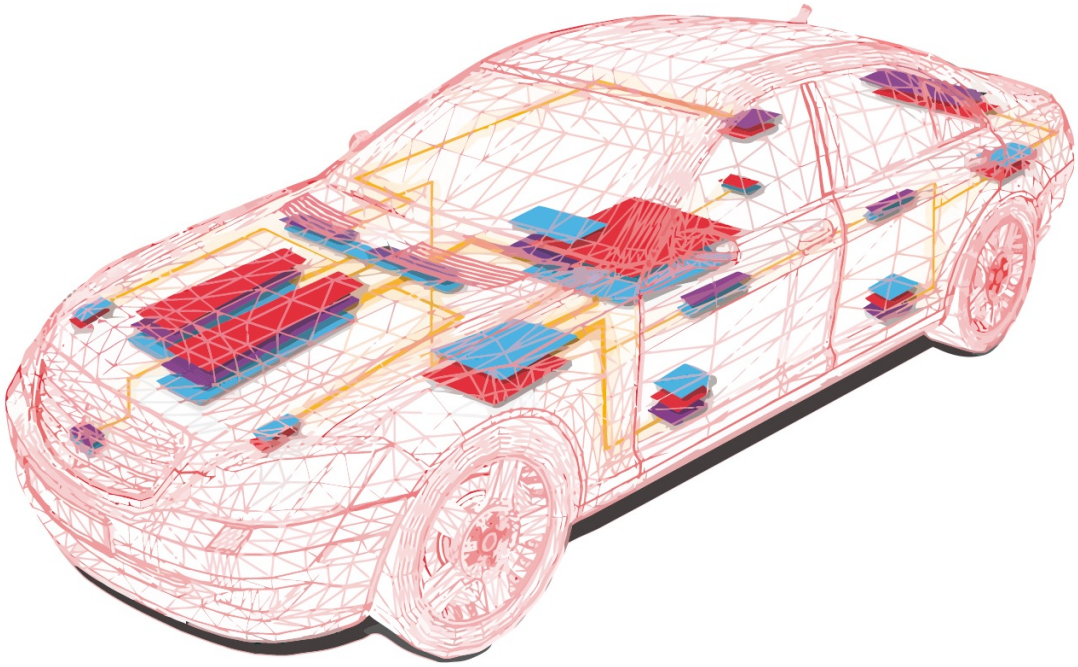

Abbildung 1.1: Symbolbild zur Veranschaulichung vernetzter elektronischer Systeme in einem aktuellen Automobil, Quelle: CoRE Arbeitsgruppe

In der Abbildung 1.1 wird in einem Symbolbild beispielhaft die Vernetzung elektronischer Komponenten in einem aktuellen Automobil dargestellt. Durch die stärkere Verbreitung von drive-by-wire, Vehicle-to-Vehicle (V2V) und Vehicle-to-Roadside (V2R) wird in den kommenden Jahren die Anzahl an Steuergeräten und Anforderungen an die elektronischen Systeme

weiterhin ansteigen. Da die elektrischen Systeme im Automobil immer komplexere Aufgabenstellungen zu bewältigen haben, wie z.B. das automatische Reagieren auf ein Verkehrshindernis, welches durch Kamera-, Radar- und Light Detection and Ranging (LIDAR)-Daten erkannt wurde, müssen die alten Netzwerktechnologien im Automobil ergänzt und zu neuen Netzwerktechnologien migriert werden. Die neuen Netzwerktechnologien müssen dabei die zukünftigen Anforderungen an Übertragungsbandbreite und deterministischer Echtzeitkommunikation erfüllen.

In der Automobilbranche wird Ethernet aktuell für ausgewählte Teilfunktionen in Fahrzeugen der Oberklasse eingesetzt. Anwendungsfälle sind dort unter anderem Multimedia- und Komfortanwendungen, die sehr bandbreitenintensiv sind. Im Jahr 2020, darauf deuten aktuelle Prognosen hin, soll bereits in 40% der neu produzierten Automobile zu mindestens anteilig Ethernet verwenden werden (ABI research (2014)). Andere Prognosen zeigen ebenfalls, dass voraussichtlich bis zum Jahr 2020 die Anzahl an Ethernet Ports in Neuwagen bei 300 Millionen Stück verteilt über alle Fahrzeugklassen liegen wird (Christiane Brünglinghaus). Dank dieser Entwicklung ist ein späterer Einsatz von RTE für sicherheitskritische Anwendungen mit hohem Bandbreitenbedarf auf der Basis von Echtzeitkommunikation sehr gut denkbar.

An der Hochschule für Angewandte Wissenschaften Hamburg (HAW-Hamburg) forscht die CoRE Arbeitsgruppe an einem switch-gestützten RTE, welches als Kommunikations-Backbone im Auto eingesetzt werden soll. Dabei wird geprüft, ob das Backbone technisch sinnvoll und flexibel genutzt werden kann. Die Verwendung von RTE als Nachfolger der aktuellen heterogenen Bussysteme im Automobil wird von der CoRE Arbeitsgruppe durch Simulationen und die Entwicklung eines technischen Prototyps vorangetrieben. Bei heterogenen Netzwerken werden durch steigende Anforderungen immer wieder neue Netzwerktechnologien benötigt, da die Anforderungen von den alten Netzwerktechnologien nicht mehr abgedeckt werden können. Diese heterogenen Netzwerke werden in der Planung und Entwicklung mit jeder neuen Netzwerktechnologie komplexer und kostenintensiver. Die Komplexität und die Kosten entstehen z.B. durch die aus organisatorischen Gründen geographisch dezentralen Entwicklungsprozesse der unterschiedlichen Dienstleister, die Entwicklung von Steuergeräten, welche auf den neuen Netzwerktechnologien basieren und die Integration der neuen Netzwerktechnologie in das bestehende Bussystem. Durch die Entwicklung eines homogenen Systems könnten die dezentralen Expertengruppen von heterogenen System enger zusammen geführt werden. Diese Veränderung kann natürlich nicht in einem einzigen Schritt erfolgen und deshalb muss bei der Migration darauf geachtet werden, dass sie mit Hilfe von Investitionsschutz durchgeführt wird.

Das bedeutet, dass die Hersteller weiterhin ihr bestehendes Wissen, Arbeitskräfte, Technik und Software weiterverwenden können. Um dies zu erreichen wird ein Gateway von der alten Technologie hin zur neuen Technologie benötigt.

Das Ziel dieser Arbeit ist die Unterstützung der Migration von Automobilbussystemen hin zu einem RTE-basierten Backbone-Netzwerk, durch die Überprüfung der Ersetzbarkeit des herkömmlichen zentralen Kommunikationsgateways. Hauptaugenmerk liegt hier auf der Verbindung von verschiedenen CAN-Bussystemen, welche über das RTE Backbone kommunizieren und nicht wie bisher über das zentrale Kommunikationsgateway. Um dies zu realisieren müssen alle Nachrichten zum zentralen VW Gateway durchgelassen werden, jedoch werden bestimmte Nachrichten, welche über das RTE Backbone gesendet werden sollen, beim weiterleiten vom zentralen ECU Gateway blockiert. Hierbei muss für Testzwecke sichergestellt werden, dass keine kritische CAN Nachrichten über das RTE Backbone geleitet werden.

Diese Arbeit ist so aufgebaut, dass in Kapitel 2 zuerst die Grundlagen von Bussystemen im Auto, RTE und dem RTE Gateway vorgestellt werden. Gefolgt von der Erläuterung der Anforderungen an das zu entwickelnde System in Kapitel 3. In Kapitel 4 wird das Konzept entwickelt und beschreibt die Vorgehensweisen, welche zur Erfüllung der Zielsetzung benötigt werden. Ebenfalls wird hier die zu tätigende Konfiguration an den vorhanden Gateways erläutert. Die Qualitätssicherung der entwickelten Komponenten erfolgt im Kapitel 5. Anschließend wird im Kapitel 6 das entwickelte System ausgewertet und die Zeitverhalten der einzelnen Systemkomponenten bzw. des Gesamtsystems evaluiert. In Kapitel 7 wird zum Schluss die Arbeit zusammen gefasst, ein Fazit gezogen und ein Ausblick auf künftige Arbeiten gegeben.

2 Grundlagen

In diesem Kapitel werden die Grundlagen erläutert, welche zum weiteren Verständnis der Arbeit benötigt werden. Hierzu gehören die grundlegenden Informationen über die aktuellen Bussysteme im Automobil und welchen Zweck sie haben.

Aktuell werden im Automobil verschiedene serielle Bussysteme eingesetzt. Jedes Bussystem hat dabei eine andere Aufgabe bzw. Anforderung und ist unterschiedlich stark verbreitet. Die Hauptunterschiede der verschiedenen Bussysteme liegen in

- Bandbreite
- Protokoll
- Topologie (Bus/Stern)
- Steuerung (Ereignis/Zeit)
- Kosten für die Anschaffung, Implementierung und Produktpflege

Zu den Bussystemen im Auto gehören Local Interconnect Network (LIN), CAN, Media Oriented Systems Transport (MOST) und FlexRay. In der Abbildung 2.1 wird zur Veranschaulichung der unterschiedliche Einsatz der Bussysteme im Automobil dargestellt. Dort ist zu erkennen, dass zur Kommunikation zwischen verteilten Bussystemen immer ein zentrales Gateway benötigt wird. Der LIN-Bus dient der Vernetzung von Sensoren und Aktoren und wird dort eingesetzt wo die Bandbreite und Vielseitigkeit von CAN nicht benötigt wird. Ein typische Anwendungsbeispiel für LIN wäre die Vernetzung innerhalb einer Tür oder eines Sitzes im Automobil. Der MOST-Bus ist für die Übertragung von Video-, Sprach- und Datensignalen gedacht und wird für Multimediaanwendungen im Automobil verwendet. Der FlexRay-Bus soll die erhöhten Anforderungen zukünftiger Vernetzung im Automobil erfüllen, welche nicht durch den CAN-Bus abgedeckt werden können. So hat der FlexRay-Bus eine höhere Datenübertragungsrate, Echtzeit-Fähigkeit, Ausfallsicherheit und wird für X-by-Wire-Systeme eingesetzt. Diese Arbeit beschäftigt sich jedoch nur mit dem CAN Bus, weil die anderen Bussysteme nicht im Fokus dieser Arbeit liegen.

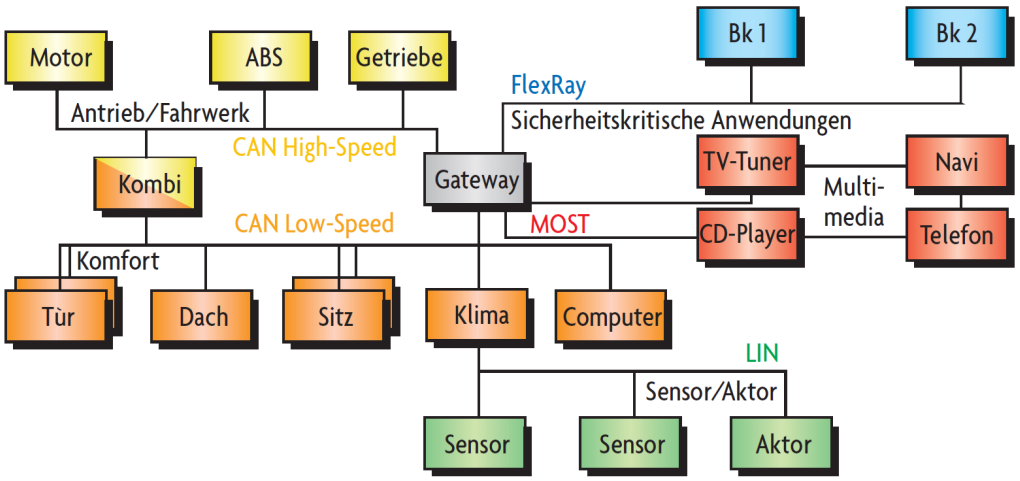

Abbildung 2.1: Vernetzung von aktuellen Fahrzeugen mit seriellen Bussystemen, Quelle: Vector Informatik GmbH (2014)

2.1 Controller Area Network (CAN)

Die Entwicklung für den CAN wurde 1983 von der Firma Bosch gestartet und gehört zu der Familie der Feldbusse. Im Jahr 1986 stellte Bosch das CAN beim Society of Automotive Engineers (SAE) Kongress vor, welches auch als die "Geburt vom CAN" bezeichnet wird. Ein Jahr später, im Jahr 1987 brachten die Firmen Intel und Philips die ersten CAN Controller Chips auf den Markt. Seitdem fangen immer mehr Firmen an CAN Chips zu entwickeln und zu verkaufen (Zhu (2010)). Das CAN wurde speziell dafür entwickelt Kabelbäume in Fahrzeugen zu reduzieren, das heißt Gewicht zu sparen und um die Steuergeräte im Automobil zu vernetzen. Es ist durch die International Organization for Standardization (ISO) international standardisiert (ISO 11898 1 (2003)) und definiert einen Multi-Master Betrieb mit differentiellen Pegel. Aus dem ISO 11898 1 (2003) entstanden die Standards ISO 11898 2 (2003) (Highspeed-CAN) und ISO 11898 3 (2006) (Lowspeed-CAN). Highspeed-CAN und Lowspeed-CAN sind nicht zueinander kompatibel, da sie sich nicht nur in der Datenrate, sondern auch in anderen Eigenschaften unterscheiden.

Das CAN Netzwerk ist aktuell das am meist benutzte Netzwerk im Automobil und die Anzahl an verkauften CAN Chips steigt weiter an. Laut Schätzungen gab es im Jahr 2014 weltweit ungefähr 1 Milliarde verkaufte Chips (CAN Newsletter Online (2014)). Das Zugriffsverfahren

Carrier Sense Multiple Access / Collision Resolution (CSMA/CR) wird im CAN Netzwerk als Kollisionsmanagement eingesetzt und erlaubt eine Übertragung von Frames mit der Nutzdatenlast von 0-8 Bytes bei einer Bandbreite von bis zu 1 MBit/s. Durch die 11 Bit (Base frame format/CAN 2.0A) bzw. die 29 Bit (Extended frame format/CAN 2.0B) lange Nachrichten-ID kann eine Adressierung, Nachrichtenpriorisierung sowie eine Busarbitrierung durchgeführt werden. Zur Arbitrierung muss jeder Sender die Busleitung überwachen während die Bits der Nachrichten-ID nacheinander dominant (0) bzw. rezessiv (1) auf die Busleitung geschrieben werden. Sobald eine Nachricht mit einer höheren Priorität (niedrige ID = hohe Priorität) gesendet wird, überschreiben die dominanten Bits einer niedriger priorisierten Nachricht und der Sender mit der niedrigeren Priorität merkt dies und stellt das Senden ein.

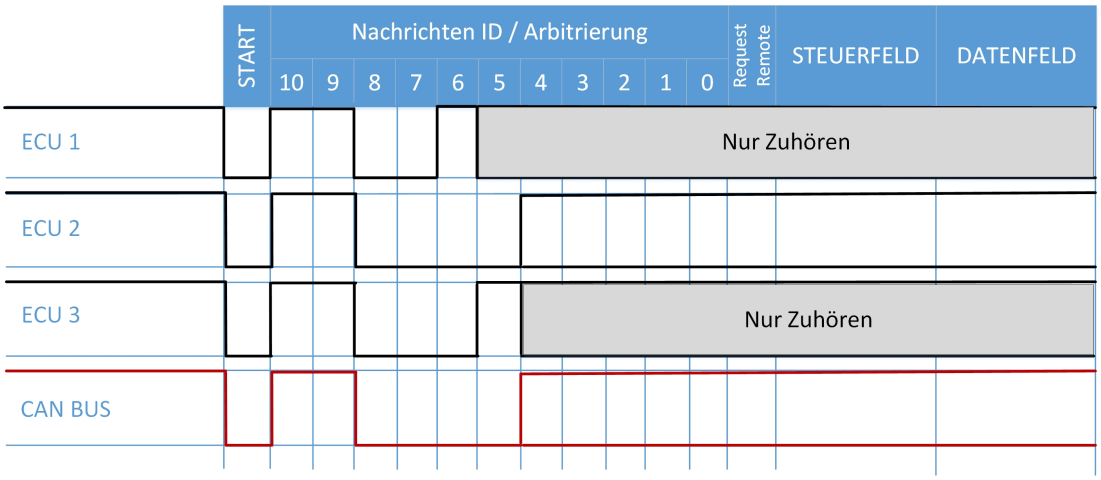

Abbildung 2.2: Veranschaulichung der CAN-Arbitrierung zwischen drei ECU, ECU zwei dominiert

Die Übertragung der Bits auf der Busleitung erfolgt mit Hilfe der Non-Return-to-Zero (NRZ) Kodierung. Das heißt jedem Bit-Wert (1/0) wird ein bestimmter Leitungszustand (high/low) zugeordnet. Hierbei kann es passieren, dass bei einer langen Sequenz von identischen Bits ebenso lange eine identische Spannung auf der Busleitung anliegt und sich der Pegel nicht ändert. Durch das Fehlen des Pegelwechsels fällt es dem CAN Controller schwer einzelne Bits bei einer Sequenz von identischen Bits zu erkennen. Aus diesem Grund gibt es im CAN-Protokoll das sogenannte Bit Stuffing. Beim Bit Stuffing wird nach 5 identischen Bits ein Pegelwechsel erzwungen. Nachteil dieser Lösung ist, dass CAN-Nachrichten mit gleicher Nutzlastlänge unterschiedlich groß sein können.

2.2 Real-Time Ethernet (RTE)

Das Netzwerkprotokoll, welches Allgemein als Ethernet bekannt ist und international als IEEE 802.3 (2012) standardisiert ist, bietet keine Möglichkeit zum Priorisieren von Nachrichten und unterstützt ausschließlich das Senden von asynchronen Nachrichten. Ein einzelner Sender könnte aus diesem Grund, unter Verwendung des Ethernet Protokolls, mit Hilfe einer bandbreitenintensiven Anwendung wie z.B. einem Video-Stream eine Infrastrukturkomponente wie z.B. einen Switch oder Router überlasten, sodass dieser Nachrichten von anderen Sender vernachlässigt. Die im Rahmen des IEEE 802.1Q (2011) Standards eingeführten Class of Service (CoS) Prioritäten sind aufgrund der geringen Auflösung der Prioritätsklassen (3 Bit) nicht dazu geeignet konkurrierendes Verhalten von Nachrichten auszuschließen. Bei hoher Systemkomplexität wäre eine Mehrfachverwendung der CoS durch unterschiedliche Nachrichten nicht auszuschließen, weshalb sich dieses nicht deterministische Verhalten nicht für sicherheits- und zeitkritische Anwendungen eignet.

An der HAW-Hamburg wurde in der CoRE Arbeitsgruppe bereits ein Protokoll für ein hartes Echtzeitverhalten in Kommunikationsnetzwerken entwickelt. Das RTE ist eine Erweiterung des Ethernet-Protokolls, welches ein deterministisches und vorhersagbares Verhalten aufweist. Dies ist durch das Kontrollieren des Sendeverhaltens auf Basis von einem Time Division Multiple Access (TDMA)-Schedules möglich. In einem TDMA Netzwerk verfügen alle Teilnehmer (Endgeräte/Infrastrukturkomponenten) über eine synchronisierte, globale Zeit und vordefinierte, kollisionsfreie Sendezeitpunkte. Durch diese Eigenschaften ist eine deterministische und zeitdeterministische Übertragung der Daten, ohne die Gefahr der ungeplanten Verzögerung durch die Überlastung dritter Netzteilnehmer, möglich. An der HAW-Hamburg wird ein RTE verwendet, welches auf dem TTEthernet der Firma TTTech basiert. Im Kontext dieser Arbeit wurde der erwähnte TTEthernet Stack verwendet und alle gewonnenen Ergebnisse basieren auf diesem Stack. Das TTEthernet erfordert als spezielle Infrastrukturkomponente einen Switch, welcher die Mechanismen zur Umsetzung der Übertragungsverfahren realisiert. Desweiteren werden Netzwerkteilnehmer benötigt, die einen TTEthernet Protokollstack mit einer proprietären API der Firma TTTech verwenden.

Im TTEthernet werden für die Priorisierung drei unterschiedliche Nachrichtenklassen definiert, welche zur Erlangung des Echtzeitverhaltens genutzt werden. Die höchste Priorität hat die Time Triggered (TT) Nachrichtenklasse, die dem TDMA-Ansatz folgt. Eine Priorität niedriger hat die bandbreitenlimitierende Rate Constraint (RC) Nachrichtenklasse, welche von der

niedrigsten Priorität gefolgt wird. Die Nachrichtenklasse mit der niedrigsten Priorität heißt Best Effort (BE) und entspricht dem Ethernet nach dem Standard IEEE 802.3 (2012). Die Ziel-Media Access Control (MAC)-Adresse wird in der TT-Nachrichtenklasse im Ethernet Header durch Felder für einen Critical Traffic (CT) Indikator und den Critical Traffic Identifier (CTID) ersetzt. Die Nachrichtenprioritäten werden mit Hilfe des CTID-Feldes faktisch mit einer Auflösung von 12 Bit realisiert. Die RC-Klasse basiert auf einem konfigurierbaren Bandwidth Allocation Gap (BAG) entsprechend des Avionics Full DupleX Switched Ethernet (AFDX)-Protokolls, welches eine Bezeichnung für den ARINC-Standard 664 ist. Nachrichten nach dem BAG Verfahren haben immer eine feste Bandbreite und werden in unserem Fall dann gesendet, wenn keine TT-Nachrichten gesendet werden müssen. Klassische Standard Ethernet Nachrichten, welche nicht deterministisch sind, werden mit Hilfe der BE-Nachrichtenklasse verschickt. In der Abbildung 2.3 wird Beispielhaft ein Sendezyklus im RTE-Netzwerk dargestellt, welcher das Zusammenspiel der einzelnen Nachrichtenklassen verdeutlicht.

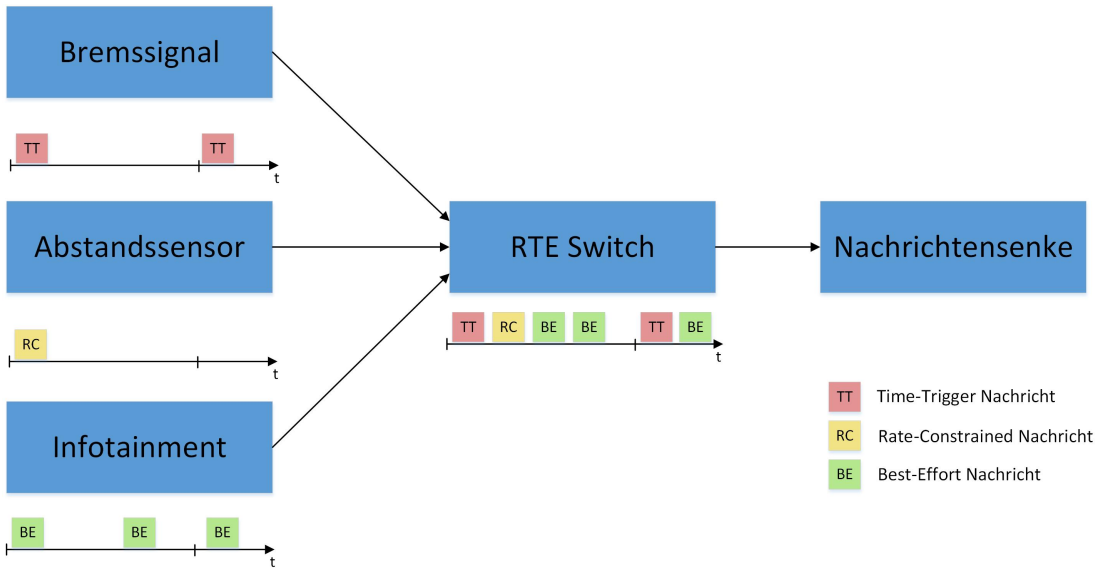

Abbildung 2.3: Beispielhafter Sendezyklus und Zusammenspiel der Nachrichtenklassen

Durch diese Implementierung müssen einige Punkte beachtet werden: Die Verwendung des TDMA-Ansatzes setzt eine globale, synchronisierte Zeitbasis voraus und da die im TT-Ethernet ersetzte Ziel-MAC-Adresse nicht mehr zur Wegfindung benutzt werden kann, muss dort ebenfalls eine andere Lösung benutzt werden. Die globale Zeitsynchronisation erfolgt durch den Austausch von Synchronisationsnachrichten, sogenannten Protocol Control Frames (PCF), welche über ein spezielles Feld im TT-Ethernet Header erkennbar sind. Die Hierarchie für

den Synchronisationsmechanismus stellt sich wie folgt da: Es gibt mindestens einen oder mehrere Synchronization Master (SM), keinen oder einen Compression Master (CM) und einen oder mehrere Synchronization Client (SC). Sollten mehrere SM aktiv sein, so senden alle SM ihre Zeit an den CM, welcher daraus eine globale Zeit bildet und diese an die SM und SC schickt. Diese stellen ihre Zeit dann auf die globale Zeit ein. Sollte nur ein SM im Netzwerk aktiv sein, wird nicht zwingend ein CM benötigt. Sollte doch ein CM vorhanden sein, so schickt der SM oder der CM die Zeit des einen SM direkt an die SC, welche ihre Zeit darauf einstellen. In diesem Fall kann auf die Zeitermittlung im CM verzichtet werden. Die Wegfindung erfolgt im RTE Netzwerk mit Hilfe der ersetzten Ziel-MAC-Adresse durch statisch vorkonfigurierte Nachrichtenrouten. Jede CTID wird dabei mit genau einem entsprechenden Virtual Link Identier (VLID) verknüpft. Zur Entwurfszeit wird jedem VLID eine Route von einem Sender zu einem oder mehreren Empfängern zugeordnet. Durch die CTID kann die Netzwerksturktur dann anhand von einem VLID eine Route zum Ziel herleiten.

2.3 Real-Time Ethernet Gateway

Das RTE Gateway ist im Rahmen der Master-Thesis von Jan Depke (Depke (2015)) im Jahr 2014 an der HAW-Hamburg entstanden. Sinn und Zweck des Gateways ist es, eine flexible Anbindung der bestehenden verschiedenen Bussysteme im Automobil an das RTE Backbone des Prototypfahrzeugs zu bekommen. Dies ist für die Migration hin zum RTE Backbone nötigt, damit Entwickler keinen harten Umschwung auf eine neue Technologie bei der Entwicklung machen müssen. Zu diesem Zweck ist das RTE Gateway entstanden, welches alte sowie neue Steuergeräte dezentral über das RTE Backbone verbindet und somit eine Kommunikation über das zentrale Gateway im Automobil ersetzt. Steuergeräte, welche wie bisher über im Automobil bestehende Bussysteme kommunizieren, sollen nicht merken, dass sich etwas an der Bustopologie verändert hat und funktionieren weiterhin. Im Rahmen der Master-Thesis von Jan Depke wurde die Verbindung der CAN-Busse an das RTE Backbone untersucht und getestet, jedoch wurde eine Kommunikation zwischen den CAN-Bussen nicht etabliert. Aktuell werden CAN Nachrichten mit beliebiger ID via RTE an einen Logging PC weitergeleitet. In Zukunft ist der Einsatz der RTE Gateways ebenfalls für andere Bussysteme möglich (MOST, Flexray).

Die Umwandlung von Busnachrichten - in dieser Arbeit werden nur CAN-Nachrichten behandelt - geschieht anhand von Regeln, welche im RTE Gateway konfiguriert werden müssen. Jedes mal wenn eine CAN Nachricht eintrifft prüft das RTE Gateway anhand der CAN ID, ob

es eine passende Regel für die Umwandlung hat. Sollte keine Regel vorhanden sein, so unternimmt das RTE Gateway nichts weiter. Trifft jedoch eine Regel zu, wird die CAN Nachricht für die Umwandlung vorbereitet und intern in eine Warteschlange für die weitere Verarbeitung hinzugefügt. Nachdem diese Umwandlung abgeschlossen ist, wird die CAN-Nachricht als RTE Nachricht in Richtung RTE Backbone geschickt.

Das RTE Gateway muss Nachrichten, welche es via RTE Backbone empfangen hat, entpacken und weiterverarbeiten können, damit die CAN Nachricht auf dem richtigen CAN-Bus ankommt. Dazu ist es notwendig, dass jede RTE Nachricht mit dem von Jan Depke entworfenen Transportprotokoll versehen wird, bevor sie über das RTE Backbone geschickt wird. In dem Transportprotokoll sind wichtige Strukturinformationen enthalten, welche zum Entpacken und Weiterverarbeiten benötigt werden.

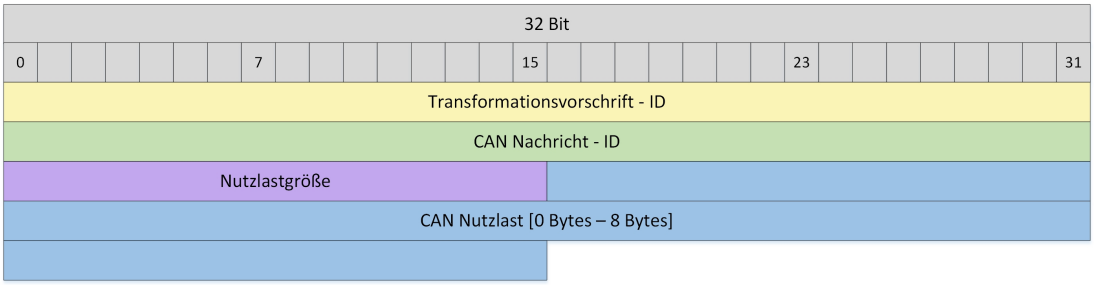

Abbildung 2.4: Darstellung des Aufbaus einer RTE Nachricht mit dem Transportprotokoll

In der Abbildung 2.4 ist der Aufbau einer RTE Nachricht beschrieben. Eine RTE Nachricht besteht nach diesem Transportprotokoll, zusätzlich zum RTE Header, aus Blöcken von 10 Bytes (keine Payload) bis 18 Bytes (inkl. größer Payload), welche jeweils eine CAN-Nachricht repräsentieren.

Zu dem Nachrichtenverkehr der CAN Nachrichten auf dem RTE Backbone kommt noch anderer Verkehr, der teilweise sehr erheblich und ungleichmäßig ist. Dies kann z.B. von der Rückfahrkamera sowie vom LIDAR System kommen. Damit es auf dem RTE Backbone nicht zu Engpässen kommt, hat Jan Depke im Rahmen seiner Master-Thesis (Depke (2015)) die Busauslastung optimiert. Da er keine Kontrolle über die bestehenden CAN-Anwendungen hat, konnte er auch nicht die Nachrichtenfrequenz steuern. Die einzige Möglichkeit ist die Steuerung der RTE Kommunikation, bei der die benötigte Bandbreite auf dem RTE Backbone reduziert wird in dem der Protokolloverhead reduziert wird.

Die minimale RTE-Payload beträgt 46 Bytes. Dementsprechend muss bei einem RTE Paket zusätzlich zu den 34 Bytes Header + 4 Bytes Prüfsumme noch eine Payload mit 46 Bytes folgen. Diese Payload wird allerdings nicht von einer CAN Nachricht ausgenutzt, da eine mit Transportprotokoll versehene CAN Nachricht wie in der Abbildung 2.4 zu sehen nur 10 bis 18 Bytes groß ist. Somit muss der restliche Platz der Payload via Padding gefüllt werden bis die 46 Bytes voll sind. Durch das Padding wird Bandbreite verbraucht ohne relevante Informationen zu übertragen. Bei einer RTE-Nachricht mit minimaler Payload beträgt der Protokolloverhead also 45 Prozent an der RTE Nachricht. Ziel ist es also eine größtmögliche RTE Nachricht mit 1500 Bytes Payload zu schicken. Hier wäre der Protokolloverhead nur noch bei 2,5 Prozent der RTE Nachricht. In der Abbildung 2.5 ist der Protokolloverhead für eine minimale und eine maximale RTE-Nachricht zu sehen. Die Lösung ist also mehrere CAN Nachrichten in einer RTE Nachricht zu übertragen. Hierbei entsteht natürlich eine Verzögerung, welche durch das Warten auf mehrere CAN Nachrichten entsteht. Die Verzögerung wird größer, je mehr Nachrichten gebündelt werden. Um eine kleinstmögliche Verzögerung zu erhalten, wird in dieser Arbeit wird das “bundling“ deaktiviert und nicht benutzt.

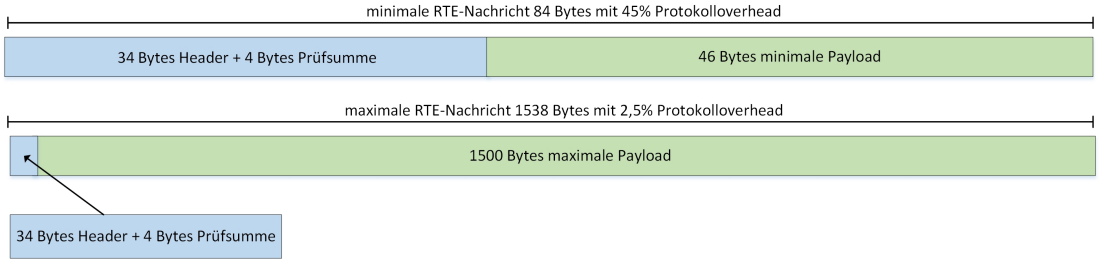

Abbildung 2.5: Darstellung des Overheads bei einer minimalen und maximalen RTE Nachricht

2.4 Prototypfahrzeug

Im Rahmen des RECBAR Forschungsprojektes, welches vom Bundesministerium für Bildung und Forschung (BMBF) unterstützt wird, hat die CoRE-Arbeitsgruppe an der HAW-Hamburg Zugriff auf ein Prototypfahrzeug (VW Golf 7) erhalten. Die CoRE-Arbeitsgruppe hat in Zusammenarbeit mit der IAV GmbH Berlin, dieses Fahrzeug für die Erprobung von RTE im Automobil mit einer RTE-Netzwerkinfrastruktur sowie RTE geeigneter Erprobungshardware ausgerüstet. Zu der Erprobungshardware gehören unter anderem Sensorik wie LIDAR-Scanner und Kameras, Embedded Systeme zur Anbindung der Sensorik an das RTE, datenverarbeitende

Computer und drei RTE-Gateways, welche zur Anbindung von Bus-Systemen im Automobil an das RTE dienen. Die RTE-Netzwerkinfrastruktur besteht aus drei miteinander verbundenen RTE-Switchen, an welche die Erprobungshardware für die RTE Kommunikation angeschlossen ist. In der Abbildung 2.6 sind zwei der drei RTE-Switche sowie die drei RTE-Gateways zu sehen.

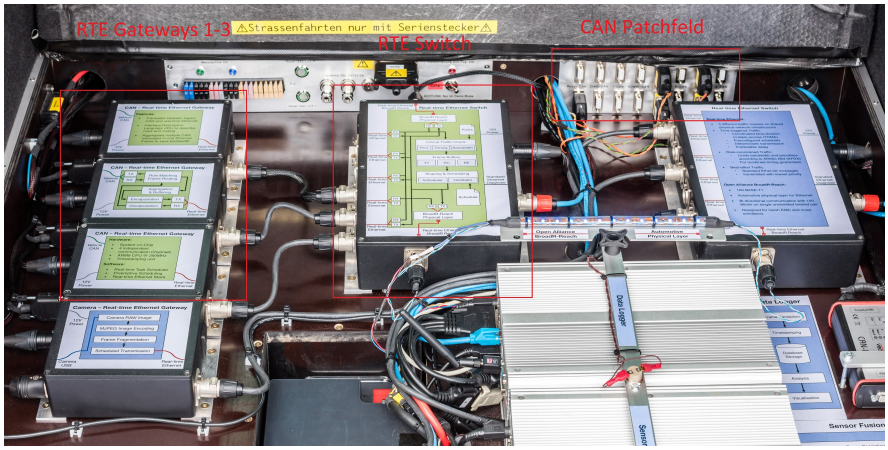

Abbildung 2.6: Kofferraum des RECBAR Prototypfahrzeug (VW Golf 7) Quelle: http://core.informatik.haw-hamburg.de/

3 Anforderungen

3.1 IST-Zustand

Bevor mit der Planung eines Konzeptes begonnen werden kann, muss erst einmal eine IST-Analyse des Systems im Prototypfahrzeug erstellt werden. Auf der Abbildung 3.1, ist der IST-Zustand grafisch dargestellt.

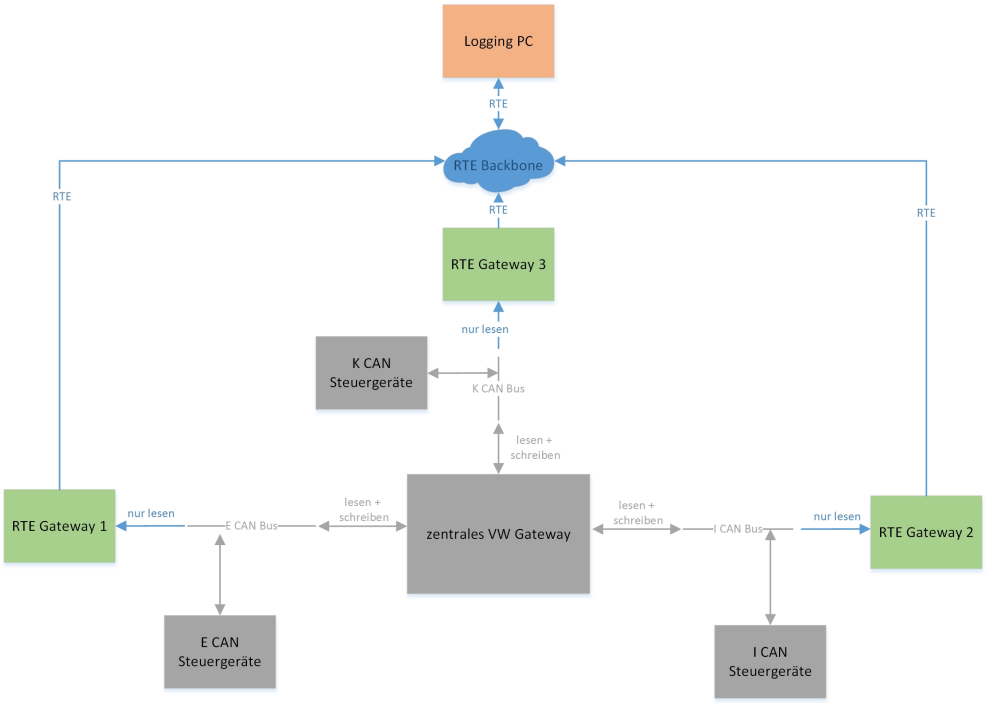

Abbildung 3.1: IST-Zustand im Prototypfahrzeug

Damit die CAN-Busse im Automobil miteinander kommunizieren können, sind alle CAN-Busse an ein zentrales Gateway angeschlossen. Dieses Gateway leitet CAN-Nachrichten an den richtigen CAN-Bus weiter sobald ein Steuergerät mit einem Steuergerät auf einem anderen CAN-Bus kommunizieren möchte. Im Prototypfahrzeug ist dies ein zentrales Gateway von

VW. In der Abbildung 3.1 sind das zentrale VW Gateway sowie die CAN-Busse und die CAN-Steuergeräte grau gekennzeichnet. Alle Komponenten, die in der Abbildung grau dargestellt sind, wurden vom Hersteller in das Fahrzeug eingebaut. Zusätzlich zu den vom Hersteller eingebauten Komponenten wurden von der CoRE-Arbeitsgruppe an der HAW-Hamburg in Zusammenarbeit mit der IAV GmbH Berlin drei RTE-Gateways im Prototypfahrzeug installiert. Diese RTE-Gateways dürfen keinen schreibenden Zugriff auf die CAN-Busse im Prototypfahrzeug haben, da dies sonst keine StVO Zulassung bekommen hätte. Aus diesem Grund wurden die CAN-Busse mit einem CAN-Kabel mit integrierter Diode an die RTE-Gateways angeschlossen. Diese Diode verhindert, dass aus Richtung der RTE-Gateways etwas auf den CAN-Bus geschrieben werden kann. Zusätzlich wurde das RTE Gateway so konfiguriert, dass es keine Nachrichten in Richtung des CAN-Bus sendet. In der Abbildung wurden Beispielhaft die drei CAN-Busse E-CAN, I-CAN und K-CAN dargestellt. In der aktuellen RTE Gateway Konfiguration könnte jedoch jeder beliebige CAN-Bus an die RTE-Gateways angeschlossen werden, da diese so konfiguriert sind, dass jede CAN-Nachricht, welche auf dem CAN-Bus gelesen wird, per RTE an den Logging PC geschickt wird. Im Kofferraum des Prototypfahrzeugs wurden sieben der im Automobil vorhanden CAN-Busse zum besseren Erreichen an ein Patchfeld angeschlossen. Die oben genannten CAN-Busse sind drei der sieben CAN-Busse, welche auf dem Patchfeld angeschlossen sind.

3.2 SOLL-Zustand

Das Ziel dieser Arbeit ist es, im Prototypfahrzeug die Kommunikation zwischen den CAN-Bussen, nicht mehr wie vom Hersteller gewünscht, über das VW Gateway laufen zu lassen. Stattdessen soll die Kommunikation über den bereits im Prototypfahrzeug verbaute RTE-Backbone geschehen. In der Abbildung 3.2 ist der SOLL-Zustand grafisch dargestellt.

Bisher wurde von einem Steuergerät eine CAN-Nachricht auf den CAN-Bus gesendet. Anschließend wurde vom VW Gateway geprüft, ob die Nachricht auf einen anderen CAN-Bus muss und gegebenenfalls dorthin gesendet. Auf dem Ziel CAN-Bus konnte das Ziel Steuergerät die CAN-Nachricht dann empfangen. Damit eine CAN-Nachricht stattdessen über den RTE-Backbone gesendet werden kann, muss diese CAN-Nachricht von einem RTE Gateway in eine Ethernet Nachricht umgewandelt werden. Anschließend muss die Nachricht von dem RTE-Gateway an das Ziel RTE-Gateway gesendet werden, um dort wieder in eine CAN-Nachricht umgewandelt zu werden. Gleichzeitig wird die CAN-Nachricht, während sie über den RTE-Backbone übertragen wird, ebenfalls an das VW Gateway gesendet. Das VW Gateway leitet diese Nachricht an den Ziel CAN-Bus weiter und wird dort von einem CAN-Filter blockiert.

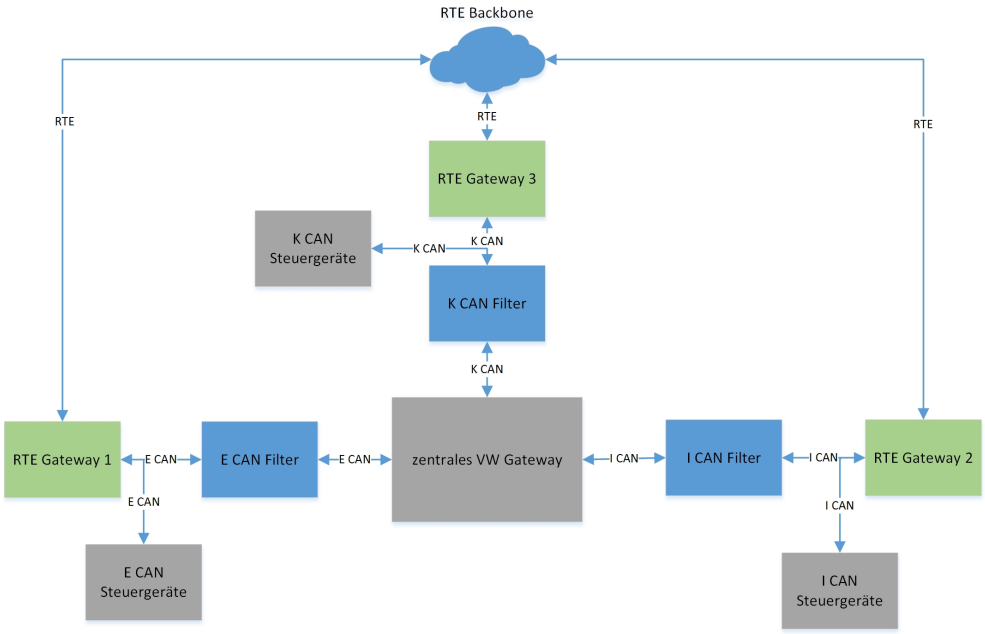

Abbildung 3.2: SOLL-Zustand des Kommunikationsnetzwerkes von CAN-Nachrichten via RTE-Backbone

Ebenfalls wird die CAN-Nachricht aus Richtung des RTE-Gateway blockiert sobald das Ziel CAN-Steuergerät die CAN-Nachricht erhalten hat, damit das VW Gateway die Nachricht nicht aus Richtung des Ziel CAN-Busses empfängt. In der Abbildung 3.3 ist die beschriebene Kommunikation genauer dargestellt und es wird beispielhaft der Weg einer Nachricht gezeigt. Die Funktionsweise dieser Kommunikation trifft jedoch auch auf alle anderen Nachrichten zu, welche über den RTE-Backbone geschickt werden.

3.3 Sicherheitsanforderungen

Ziel ist es im Prototypfahrzeug die CAN-Busse über den RTE-Backbone kommunizieren zu lassen. Für die ersten Tests zur Kommunikation über das RTE-Backbone sollten deshalb CAN-Nachrichten ausgewählt werden, welche für das Prototypfahrzeug nicht sicherheitskritisch sind. Das heißt es sollten keine Nachrichten sein, die z.B. für Bremsen oder Lenken zu ständig sind. Sollte bei der Übertragung der CAN-Nachrichten über den RTE-Backbone also ein Problem auftreten bzw. sollte die Nachricht nicht am Ziel ankommen, ist dies nicht sicherheitskritisch. Aus diesem Grund wurden selektiv CAN-Nachrichten auf dem E-CAN, I-CAN und K-CAN Bus ausgewählt.

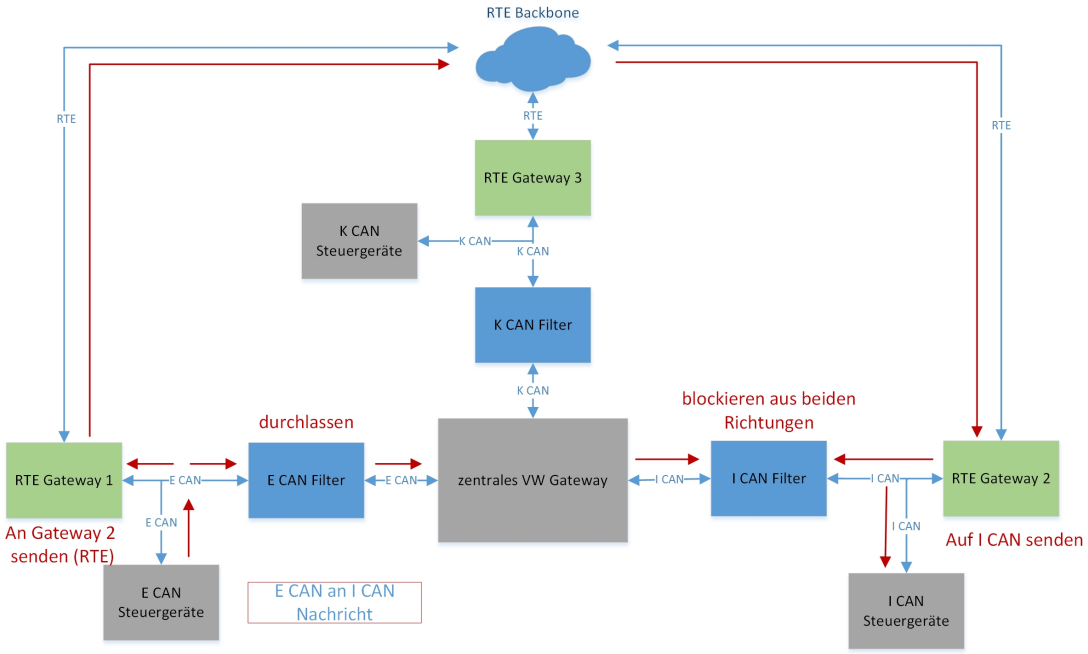

Abbildung 3.3: Beispiel Kommunikation über den RTE Backbone im SOLL-Zustand

Des Weiteren wurde als Sicherheitsanforderungen festgelegt, dass die Entwicklung des Projektes in mehreren Schritten getestet wird. Am Anfang der Entwicklung muss das Projekt im Labor getestet werden. Hier wird sichergestellt, dass die Systeme wie gewünscht funktionieren und keine unerwünschten Effekte auftreten. Unter anderem wird die Funktionalität der Teilsysteme sowie des Gesamtsystems und die Nachrichtenübermittlung geprüft. Als nächster Test der Entwicklung muss im Fahrzeuglabor der HAW-Hamburg sichergestellt werden, dass die Entwicklung im Prototypfahrzeug wie gewünscht funktioniert und kein Sicherheitsrisiko besteht. Hierbei wird das Prototypfahrzeug nicht bewegt um die Grundfunktionen zu testen. Als letzter Test in der Entwicklung wird anschließend auf dem Verkehrsübungsplatz getestet, ob die Entwicklung im fahrenden Zustand ebenfalls wie im Fahrzeuglabor funktioniert oder ob es Veränderungen gibt.

Eine weitere Anforderung für die gesamte Entwicklung ist, dass das VW Gateway alle CAN-Nachrichten bekommen muss, die über das RTE-Backbone verschickt werden. Das Prototypfahrzeug würde ansonsten Fehlermeldungen im Cockpit anzeigen, wenn das VW Gateway die CAN-Nachrichten nicht bekommen würde. Da das VW Gateway jedoch, wie im SOLL-Zustand beschrieben, sofort die Nachrichten an den Ziel CAN-Bus sendet, muss auf dem Ziel CAN-Bus dafür gesorgt werden das diese CAN-Nachricht aus Richtung des VW Gateways

blockiert wird. Die CAN-Nachricht soll nur via des RTE-Backbone bei den Steuergeräten ankommen. Aus diesem Grund muss ein weiteres Gerät, welches im weiteren Verlauf CAN-Filter genannt wird, entwickelt werden. So ist sichergestellt das es auf dem Ziel CAN-Bus nicht zu Konflikten zwischen der CAN-Nachricht vom VW Gateway und der CAN-Nachricht des RTE-Backbone kommen kann.

3.4 Zeitanforderungen

Die Bussysteme im Automobil sind teilweise als zeitkritische Systeme einzustufen. Während bei der Übertragung von Multimediainhalten keine schwerwiegenden Folgen bei Latenzausreißern auftreten, sind z.B. bei Systemen zur Motorsteuerung oder Ähnlichem die Anforderungen an das Echtzeitverhalten deutlich höher. Aus diesem Grund ist es wichtig, dass verschickte Signale so schnell wie möglich beim Ziel ankommen, damit es nicht zu einer sicherheitskritischen Situation wie z.B. einem Unfall kommt. Dazu wird im Automobil unter anderem das CAN-Netzwerk genutzt über welches Nachrichten im Mikrosekunden-Bereich ausgetauscht werden. Ziel ist es, dass es bei Echtzeit Nachrichten eine minimale Verzögerung gibt. Aus diesem Grund ist es das Ziel, dass für die Kommunikation über das entwickelte RTE-Backbone die kleinstmögliche Verzögerung der Nachricht entsteht. Da die Toleranzgrenzen der VW Steuergeräte im Prototypfahrzeug nicht bekannt sind, werden die ersten Tests ergeben, ob die Toleranzgrenzen groß genug für die Kommunikation über den RTE-Backbone sind. Anschließend sollte weiterhin versucht werden, die Verzögerung, welche durch den RTE-Backbone entsteht, zu minimieren.

4 Konzept und Implementierung

4.1 CAN Filter

4.1.1 Konfiguration der CAN Filter

Wie im SOLL-Zustand bereits beschrieben, muss der CAN-Filter bestimme Nachrichten auf dem Ziel CAN-Bus auf beiden CAN-Interfaces blockieren. Dazu ist es nötig, dass der CAN-Filter die Nachrichten empfängt und anschließend dahingehend analysiert, ob die Nachrichten auf dem zweiten CAN-Interface rausgeschickt werden sollen oder nicht. Die Analyse der Nachrichten sollte geschehen, in dem die CAN ID der empfangenen Nachricht mit einer Liste abgeglichen wird, in der IDs aufgelistet sind, die nicht weiter geleitet werden sollen. Das bedeutet, dass der CAN Filter bei allen Nachrichten, die er empfängt, prüfen muss ob sie blockiert werden sollen oder nicht. Dieses Blockieren geschieht, wie im SOLL-Zustand dargestellt, auf drei verschiedenen CAN-Bussen. Auf jedem CAN-Bus hat der CAN-Filter eine andere Blacklist an IDs, welche blockiert werden sollen, da die CAN IDs, welche übertragen werden, auf jedem CAN-Bus unterschiedlich sind. Die Blacklist der CAN-Filter beinhaltet alle CAN-Nachrichten IDs, welche über den RTE-Backbone an den jeweiligen Ziel CAN-Bus übertragen werden.

4.1.2 Softwarearchitektur CAN Filter

Die CAN-Filter haben zwei CAN Anschlüsse über die sie CAN-Nachrichten empfangen können. Ein Nachrichteneingang im CAN-Filter löst einen Interrupt Request (IRQ) aus, für den die CAN-Software eine Interrupt Service Routine (ISR) definiert haben muss. Dabei ist es egal, auf welchem CAN Interface eine Nachricht eintrifft, es wird für beide CAN Anschlüsse die gleiche ISR ausgeführt. In einem Echtzeitsystem ist es nicht ratsam, rechenintensive oder Rechenzeit schwankende Operationen innerhalb einer ISR zu platzieren. Aus diesem Grund wurde die ISR so gestaltet, dass es keine schwankende Operationen gibt und auch keine rechenintensiven Aufgaben durchgeführt werden. Sobald die ISR gestartet wird, wird die CAN-Nachricht aus dem IO Module in der ISR zwischen gespeichert. Anschließend wird die ID der CAN Nachricht mit einer vorher einprogrammierten Blacklist verglichen. Dies geschieht mit Hilfe des Binary Search Algorithmus, wodurch eine konstante Zeit zum durchsuchen der Liste benötigt wird.

Sollte die CAN ID nicht in der Blacklist vorhanden sein, wird die CAN-Nachricht auf dem CAN Interface gesendet, auf dem die Nachricht nicht empfangen wurde. So ist sichergestellt, dass die CAN Nachricht nicht auf dem selben CAN Anschluss rausgeschickt wird auf dem sie empfangen wurde. Sollte die CAN ID der CAN Nachricht jedoch in der Blacklist vorhanden sein, so wird die CAN Nachricht verworfen und die ISR beendet sich. In der Abbildung 4.1 ist der Ablauf innerhalb des CAN-Filters grafisch dargestellt.

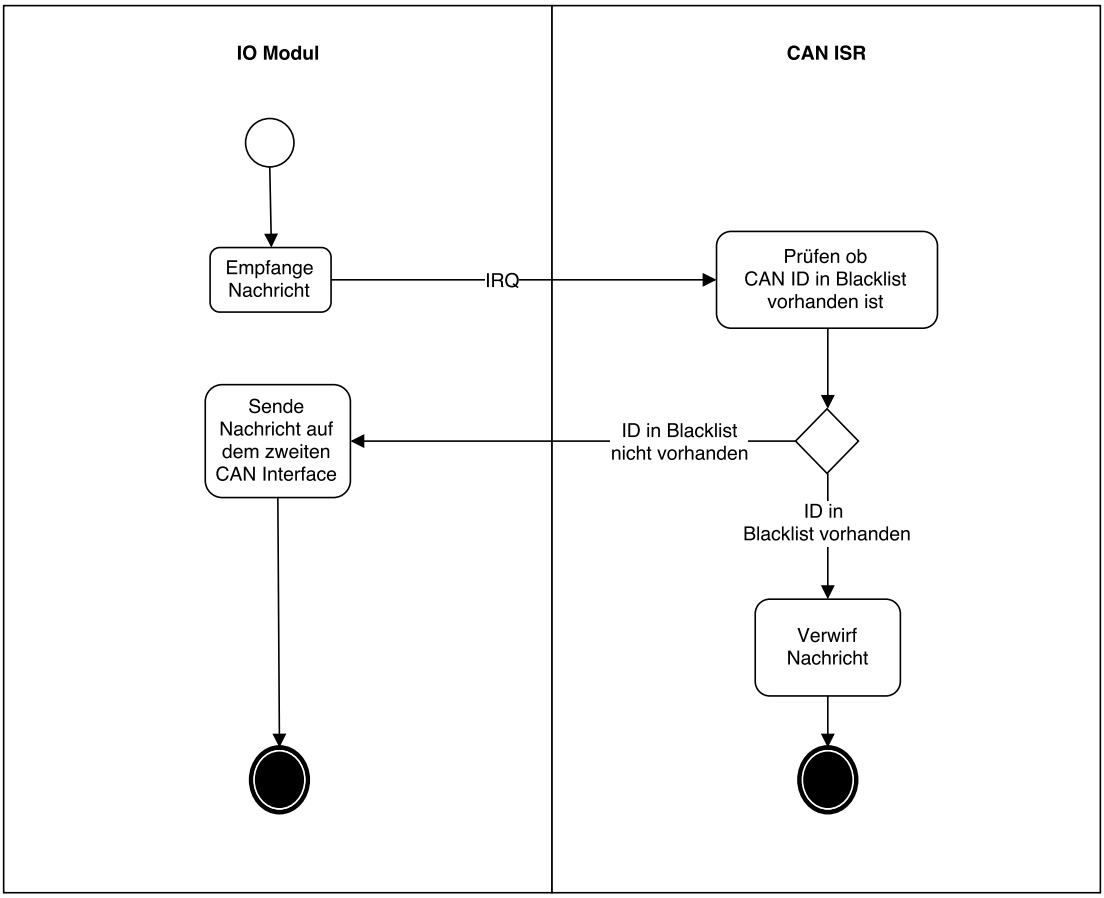

Abbildung 4.1: Aktivitätsdiagramm: Verhalten der im IO Modul implementierten ISR zur Behandlung eingehender CAN-Nachrichten

4.2 Real-Time Ethernet Gateway

Das im Prototypfahrzeug bereits eingebaute RTE Gateway hat eine Konfiguration, die noch nicht zum SOLL-Zustand passt. Aktuell werden CAN Nachrichten mit beliebiger CAN ID

vom RTE Gateway per Ethernet an einen Logging-PC gesendet. Dort werden die Nachrichten zu Analysezwecken in einer Datenbank gespeichert. Damit die CAN-Nachrichten nicht mehr nur an den Logging-PC geschickt werden, muss die Konfiguration des RTE Gateway angepasst werden. Für den Transport der CAN-Nachrichten via RTE-Backbone wurden für diese Arbeit nur bestimmte CAN IDs ausgewählt, welche nicht sicherheitskritisch für die Fahrt im Prototypfahrzeug sind. Als erstes muss das Gateway so konfiguriert werden, dass nicht mehr alle CAN-Nachrichten, welche das RTE Gateway bekommt, verarbeitet werden, sondern nur die für diese Arbeit ausgewählten CAN IDs. Zweitens muss die CAN-Nachricht via RTE-Backbone nicht nur beim Logging-PC ankommen, sondern ebenfalls beim jeweiligen Ziel RTE Gateway bzw. Ziel CAN-Bus. Hierzu wird im sendenden RTE Gateway durch Regeln festgelegt, an welches Ziel RTE Gateway die RTE-Nachricht gesendet wird. Im Prototypfahrzeug sind drei RTE Gateways installiert, das bedeutet, es gibt drei Ziel-Adressen. Das Ziel der RTE Frames wird mit der Hilfe der CTID festgelegt. Das sendende RTE Gateway trägt diese Ziel-Adresse in den RTE Frame ein und anschließend sendet der RTE Switch den RTE Frame an den richtigen Empfänger. Das Ziel RTE Gateway analysiert daraufhin der empfangene Frame und prüft, ob eine passende Regel zur Umwandlung in eine CAN-Nachricht existiert und wandelt diese gegebenenfalls um oder verwirft sie. Aktuell wurde das RTE Gateway so konfiguriert, dass CAN-Nachrichten 5ms lang gesammelt werden und dann alle Nachrichten in einem RTE-Frame verschickt werden. Dieses Sammeln von Nachrichten heißt “bundling“ und wurde von Jan Depke in seiner Master-Thesis (Depke (2015)) im Gateway aktiviert. Das Ziel vom Bundling ist es, die Bandbreite der RTE-Verbindung zu schonen, in dem ein geringerer Overhead bei den Ethernet Frames entsteht. Dies wirkt sich leider auf die Verzögerungszeit aus welche durch das Sammeln der Nachrichten wesentlich größer wird. Da in dieser Arbeit jedoch die CAN-Nachrichten nicht nur in einer Datenbank gespeichert werden sollen, sondern eine Kommunikation zwischen den CAN-Bussen darüber laufen soll, ist die Zeit von 5ms relativ groß. Für die Kommunikation zwischen den CAN-Bussen soll die Verzögerung auf dem RTE-Backbone so klein wie möglich gehalten werden. Aus diesem Grund wurde das Bundling auf dem RTE Gateway deaktiviert und das Senden via Scheduler eingesetzt. Auch das Senden via des Schedulers führte noch zu großen Verzögerungszeiten und so musste eine neue Idee zum Verarbeiten/Senden auf dem Gateway entwickelt werden. Dabei muss beachtet werden, dass der WorkerTask nicht unterbrochen werden darf. Bei ersten Tests stellte sich heraus, dass das komplette Gateway abstürzt sobald der WorkerTask unterbrochen wird. Bisher war dies kein Problem da der WorkerTask in einem festen Schedule aufgerufen wurde ohne das diese unterbrochen wurde. Aus diesem Grund wurde in der folgenden Architektur der WorkerTask als nicht unterbrechbare ISR realisiert. Da der WorkerTask relativ lange braucht ist

er leider nicht konform das eine ISR möglichst kurz ist und keine rechenintensiven Aufgaben ausführt. Dies ist leider ein notwendiges Übel was in Kauf genommen werden muss damit die Umwandlung der Nachrichten eine möglichst kurze Verzögerungszeit hat.

4.2.1 Softwarearchitektur Gateway

Zur Verarbeitung der eintreffenden Nachrichten wurde die Softwarearchitektur des RTE Gateways angepasst. Aktuell wird bei einem Nachrichteneingang im Gateway eine ISR ausgelöst. Trifft eine CAN-Nachricht ein, wird die CAN ISR gestartet. Die CAN ISR prüft dann in den Verarbeitungsvorschriften des RTE Gateway, ob die CAN-Nachricht überhaupt via RTE gesendet werden soll. Soll die CAN-Nachricht via RTE gesendet werden, wird diese einer Warteschlange des WorkerTask hinzugefügt. In der Abbildung 4.2 wurde der Ablauf der Aktivitäten beim Empfang einer CAN-Nachricht in einem Aktivitätsdiagramm dargestellt.

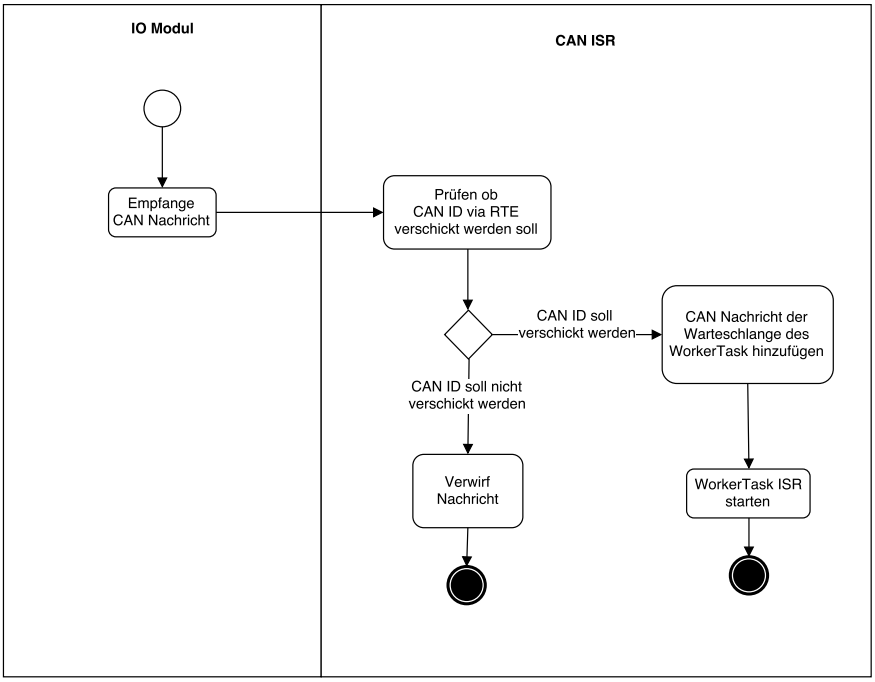

Abbildung 4.2: Aktivitätsdiagramm: Verhalten der im IO Modul implementierten ISR zur Behandlung eingehender CAN-Nachrichten

Das gleiche passiert ebenfalls bei RTE Nachrichten, die empfangen werden. Es wird eine RTE Nachricht empfangen und die RTE ISR gestartet. Die RTE ISR prüft zusätzlich in den Verarbeitungsvorschriften, ob die RTE Nachricht in eine CAN-Nachricht umgewandelt werden soll und fügt diese im Falle einer Umwandlung der Warteschlange des WorkerTask hinzu. Eine

Neuerung an dieser Stelle ist, dass die beiden ISR nach dem Hinzufügen der Nachrichten die WorkerTask ISR ebenfalls starten. In der Abbildung 4.3 ist der Ablauf der Aktivitäten beim Empfang einer RTE Nachricht in einem Aktivitätsdiagramm dargestellt.

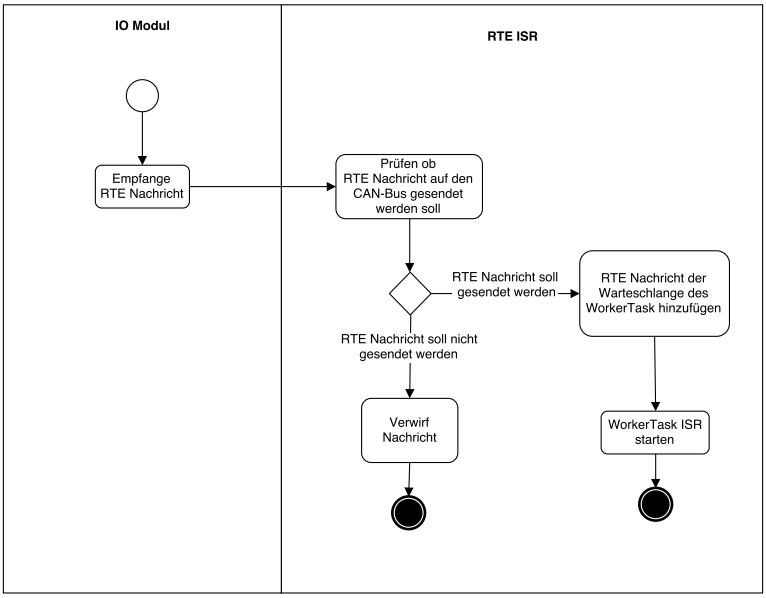

Abbildung 4.3: Aktivitätsdiagramm: Verhalten der im IO Modul implementierten ISR zur Behandlung eingehender RTE Nachrichten

Der WorkerTask blieb weitgehend unverändert. Bisher wurde der WorkerTask vom Scheduler zu einer festen Zeit aufgerufen. Neu ist, dass er jetzt direkt von der CAN Empfangs- oder RTE Empfangs-ISR gestartet wird. Hierdurch kann der WorkerTask sofort anfangen Nachrichten zu übersetzen und muss nicht erst auf den Scheduler warten. Der Ablauf in der WorkerTask ISR läuft wie in der Abbildung 4.4 dargestellt ab. Sobald die WorkerTask ISR gestartet wird, wird überprüft, ob die RTE und CAN Warteschlange leer ist. Sollten die Warteschlangen leer sein, gibt es für die WorkerTask ISR nichts zu tun und sie beendet sich. Sollte jedoch in einer der Warteschlangen eine Nachricht sein, wird zuerst geprüft, ob in der RTE Warteschlange eine Nachricht ist. Sollte dort eine Nachricht vorhanden sein, wird die Nachricht in eine CAN Nachricht umgewandelt und auf dem CAN Bus gesendet. Sobald die Nachricht gesendet wurde, prüft die Software, ob in der CAN Warteschlange eine Nachricht vorhanden ist. Dort würde das Programm ebenfalls landen, wenn in der vorherigen Überprüfung keine RTE Nachricht in der Warteschlange gewesen wäre. Sollte keine CAN-Nachricht vorhanden sein, würde die ISR wieder von vorne beginnen und überprüfen ob die Warteschlangen leer sind. Sollte allerdings eine CAN-Nachricht vorhanden sein, so wird sie in eine RTE Nachricht umgewandelt und

anschließend los geschickt. Sobald die RTE Nachricht verschickt ist, kehrt die ISR wieder in den Ausgangszustand zurück. Durch dieses Design ist gewährleistet, dass die WorkerTask ISR solange läuft, bis keine Frames mehr in den Warteschlangen sind.

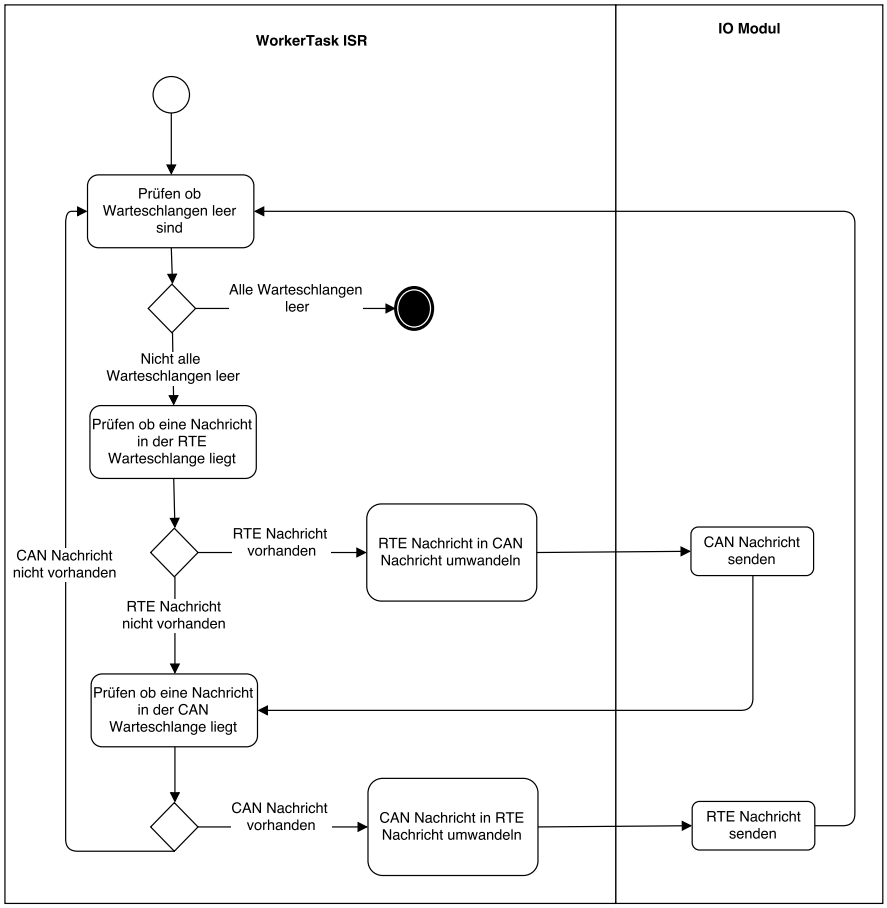

Abbildung 4.4: Aktivitätsdiagramm: Verhalten der WorkerTask ISR

5 Qualitätssicherung

5.1 Komponententest

Der Komponententest ist der erste geplante Test, den sich eine fertiggestellte Software-Komponenten zu unterziehen hat. Dabei ist der Komponententest insbesondere von Bedeutung, da er das Testobjekt in Isolation betrachtet und somit keine Wechselwirkungen mit anderen Komponenten auftreten können. Sollte also ein Fehler im Komponententest auftreten, so ist er eindeutig auf diese Komponente zurück zu führen wodurch die Suche nach der Fehlerursache erheblich vereinfacht wird. Das Ziel des Komponententests ist zu prüfen, ob die vorliegende Komponente die vorher festgelegten Anforderungen erfüllt.

5.1.1 CAN Filter

Die Anforderungen an den CAN-Filter sind in diesem Fall, dass bestimmte vorher festgelegte CAN Nachrichten nach der Analyse der CAN-ID vom CAN-Filter nicht weitergeleitet werden. Die für diese Arbeit wichtigen CAN-Busse sind der E-CAN, I-CAN und der K-CAN. In einem Komponententest müssen also alle möglichen Fälle abgedeckt werden in denen CAN-Nachrichten zwischen diesen CAN-Bussen ausgetauscht werden. Es gibt folgende Kommunikationsmöglichkeiten zwischen den CAN-Bussen.

Liste 5.1: Mögliche CAN-Kommunikation via RTE

- E-CAN sendet an I-CAN
- E-CAN sendet an K-CAN
- I-CAN sendet an E-CAN
- I-CAN sendet an K-CAN
- K-CAN sendet an E-CAN
- K-CAN sendet an I-CAN

Testfall	ID blockiert?	Test erfolgreich?
unterhalb der niedrigsten ID (nicht in der Blackliste enthalten)	nein	ja
niedrigste ID (in Blacklist enthalten)	ja	ja
höchste ID (in Blacklist enthalten)	ja	ja
oberhalb höchster ID (nicht in der Blackliste enthalten)	nein	ja

Tabelle 5.1: Test ob der CAN Filter die richtigen IDs blockiert

Für die drei verschiedenen CAN-Busse werden ebenfalls drei CAN-Filter benötigt. Auf jedem der drei genannten CAN-Busse wird ein CAN-Filter eingesetzt, der unterschiedliche CAN-Nachrichten blockieren soll. Dies bedeutet, dass jeder CAN-Filter eine andere Liste an CAN-IDs konfiguriert haben muss, die er blockieren soll. Jeder CAN-Filter wird dabei mit denselben Komponententests aber mit unterschiedlichen CAN-Nachrichten getestet. Getestet wurden hierbei pro CAN-Filter, ob alle CAN-IDs in der Liste blockiert werden, inklusive der niedrigsten und der höchsten CAN-ID. Ebenfalls wurde getestet, ob Nachrichten dessen CAN-ID nicht in der Liste vorhanden sind, weiter gesendet werden. Die Testfälle der Komponententests des CAN-Filters wurde in der Tabelle 5.1 dargestellt.

5.1.2 Gateway

Die Anforderungen an das RTE Gateway sind in diesem Fall, dass bestimmte vorher festgelegte CAN Nachrichten nach der Analyse der CAN-ID per RTE verschickt werden. Die für diese Arbeit wichtigen CAN-Busse bzw. welche Kommunikationsmöglichkeiten zwischen diesen CAN-Bussen besteht, wurde bereits in der Liste 5.1 dargestellt. Diese Liste gilt ebenfalls für die Kommunikation via RTE. Ähnlich wie bei den CAN-Filtern gibt es pro CAN-Bus jeweils ein RTE Gateway. In jedem RTE Gateway gibt es eine unterschiedliche Liste an CAN-IDs, welche per RTE an ein anderes RTE Gateway verschickt werden sollen. Jedes RTE Gateway wird dabei den gleichen Komponententests unterzogen, sie unterscheiden sich nur in den CAN-Nachrichten. Getestet wurde, ob alle in der Liste vorhanden CAN-Nachrichten via RTE verschickt wurden. Ebenfalls wurde getestet, ob CAN-Nachrichten, die nicht aufgelistet sind auch nicht von RTE Gateway via RTE versendet wurden. In der Tabelle 5.2 sind die Testfälle für die Komponententests der RTE Gateways aufgelistet. Überprüft wurden die Ergebnisse mit Hilfe der Software Wireshark, indem der Ethernet-Verkehr analysiert wurde.

Testfall	via RTE versendet?	Test erfolgreich?
nicht in der Liste enthaltene CAN-Nachrichten	nein	ja
in der Liste enthaltene CAN-Nachrichten	ja	ja

Tabelle 5.2: Test ob das GW nur die richtigen IDs über RTE sendet

5.2 Integrationstest

Nachdem in den Komponententests die Funktionalität der einzelnen Komponenten sichergestellt wurde, ist es wichtig zu testen wie sich die Systeme verhalten, wenn sie miteinander kommunizieren. Dabei muss für jede Kommunikationsmöglichkeit, welche in der Liste 5.1 aufgelistet wurde, sichergestellt werden, dass die Komponenten zusammen genauso funktionieren wie sie isoliert funktioniert haben. Wird z.b. eine Nachricht von dem E-CAN an den I-CAN geschickt, so muss diese auf dem E-CAN vom CAN-Filter an das VW Gateway durchgelassen werden und gleichzeitig von dem RTE Gateway per RTE an das RTE Gateway vom I-CAN gesendet werden. Auf dem I-CAN muss anschließend sichergestellt werden, dass der CAN-Filter die CAN-Nachricht, welche vom VW Gateway gesendet wird, blockiert. Ebenfalls muss sichergestellt werden, dass das RTE Gateway auf dem I-CAN die RTE Nachricht wieder in eine CAN-Nachricht umwandelt und sendet. Der CAN-Filter auf dem I-CAN blockiert die CAN-Nachricht, welche vom RTE Gateway kommt, in Richtung des VW Gateways, weil dieses die CAN-Nachricht nicht ein zweites Mal bekommen sollte. In der Liste 5.2 sind die Testfälle aufgelistet, welche pro CAN-Nachricht, die per RTE Gateway gesendet werden soll, geprüft werden muss. Erst nachdem diese Tests für jede Kommunikationsmöglichkeit erfolgreich waren, wurde mit dem Systemtest fortgefahren.

Liste 5.2: Testfälle für den Integrationstest

- lässt der CAN Filter auf dem Quellbus die Nachricht durch?
- sendet das RTE Gateway auf dem Quellbus die Nachricht per RTE?
- sendet das RTE Gateway auf dem Zielbus die Nachricht via CAN?
- blockiert der CAN-Filter auf dem Zielbus die Nachricht aus beiden Richtungen?

5.3 Systemtest

Beim Systemtest wird das System gegen die vollständigen Anforderungen getestet. Während die Komponententests und die Integrationstests im Labor aufgebaut und durchgeführt wurden,

werden für den Systemtest die Komponenten im Prototypfahrzeug eingebaut. Hierbei wird geprüft, ob die Komponenten im Prototypfahrzeug genauso funktionieren wie im Integrationstest. Die Komponenten müssen dafür so angeschlossen werden wie es im SOLL-Zustand in der Abbildung 3.2 dargestellt wurde. Der Test wird dabei als Blackbox-Test durchgeführt. Das heißt, es wird nur geprüft, ob die Ergebnisse den erwarteten Werten entsprechen. Dabei werden genauso wie beim Integrationstest alle Kommunikationsmöglichkeiten aus der Liste 5.1 getestet. Interessant ist hierbei, ob Fehlermeldungen im Cockpit des Prototypfahrzeugs angezeigt werden und ob die ausgewählten CAN Nachrichten von den Zielbus RTE Gateways gesendet werden. Wenn keine Fehlermeldungen mehr im Cockpit angezeigt werden und die CAN Nachrichten via RTE an ihrem richtigen Ziel ankommen ist die grundlegende Funktion des Systems gewährleistet.

6 Messungen und Evaluierung

In diesem Kapitel wird als erstes der Einsatz im Prototypfahrzeug beschrieben, auf welches die CoRE-Arbeitsgruppe Zugriff hat. Dieses Prototypfahrzeug spielt eine wichtige Rolle bei den Messungen, welche in diesem Kapitel durchgeführt werden sowie in der anschließenden Evaluation der Messergebnisse. Nachdem das Prototypfahrzeug näher vorgestellt wurde, wird detailliert darauf eingegangen, was für Messungen durchgeführt werden, wie die Messungen durchgeführt werden und warum. Die Messungen müssen dabei unter Laborbedingungen sowie unter realen Bedingungen geschehen. Zunächst werden im Labor Messungen bezüglich des Zeitverhaltens der CAN-Filter und RTE Gateways getätigt. Danach werden Messungen im Prototypfahrzeug durchgeführt. Die ersten Tests und Messungen im Prototypfahrzeug geschehen dabei im Fahrzeuglabor, während sich das Fahrzeug im Stand befindet. Im Anschluss daran werden Messungen gemacht während sich das Prototypfahrzeug in Bewegung befindet. Hierzu werden mit dem Prototypfahrzeug auf einem Verkehrsübungsplatz mehrere Runden gefahren.

Nach Beendigung der Messungen wird eine Evaluation der einzelnen Messungen durchgeführt. Am Anfang der Evaluation wird das Zeitverhalten des CAN Filters analysiert und bewertet. Daraufhin wird eine Evaluation des Zeitverhaltens des VW-Gateways im Stand sowie während der Fahrt durchgeführt. Anschließend wird ebenfalls das Zeitverhalten des RTE-Backbones im Stand sowie während der Fahrt bewertet. Im letzten Punkt des Kapitels wird das Zeitverhalten des VW-Gateways ins Verhältnis zum Zeitverhalten des RTE-Backbones gesetzt und bewerte.

6.1 Einsatz im Prototypfahrzeug

Der Einsatz im Prototypfahrzeug erfordert ein hohes Maß an Qualitätssicherung, damit die korrekte Funktion des Fahrzeugs gewährleistet werden kann. Hierzu muss geprüft werden, ob die Latenzen, welche bei der Übertragung über das RTE-Backbone entstehen, den erwarteten Werten entsprechen. Des Weiteren muss geprüft werden, ob auch wirklich alle Nachrichten, welche über das RTE-Backbone gesendet werden, auf dem Ziel CAN-Bus ankommen. Ein Verlust von CAN-Nachrichten könnte verheerend für die Funktion des Fahrzeugs sein. Der

CoRE-Arbeitsgruppe liegen nur wenig bis gar keine Informationen über die Funktionsweise der VW Steuergeräte und des VW Gateways vor. Die meisten Informationen werden vom Fahrzeughersteller unter Verschluss gehalten. Informationen über ausgewählte CAN-IDs, die zwischen bestimmten CAN-Bussen ausgetauscht werden, liegen vor. Aus diesen Gründen kann keine Aussage über die Testabdeckung im Prototypfahrzeug getätigt werden. Für den einfacheren Zugang zu den CAN-Bussen werden im Prototypfahrzeug, wie auf der rechten Seite in Abbildung 2.6 zu sehen, sieben der CAN-Busse an dem Patchfeld im Kofferraum angeschlossen. Es wurden für die Tests im Prototypfahrzeug der E-CAN, der I-CAN und der K-CAN selektiv ausgewählt. Auf diesen drei Bussen werden keine kritischen Nachrichten übertragen, sie eigenen sich deshalb für die Tests im Prototypfahrzeug.

6.2 Messinstrumente

Für die Messungen werden in der weiteren Arbeit die Software CANalyzer (Vector Informatik GmbH (a)) bzw. CANoe (Vector Informatik GmbH (b)) von Vector eingesetzt. Mit dieser Software ist es möglich mit Hilfe von zusätzlichen Hardware Interfaces Messdaten darzustellen, analysieren und sie zu speichern. In dieser Arbeit wurde das Hardware Interface VN5610 (Vector Informatik GmbH (c)) von Vector benutzt. Das VN5610 ermöglicht eine transparente Überwachung und Aufzeichnung von Ethernet-Datenströmen und CAN-Events mit minimaler Latenzzeit und hoher Zeitstempelauflösung. Es besitzt zwei Ethernet Ports, zwei CAN-Ports und besitzt eine Hardware-Zeitsynchronisation über alle Interface. Somit können die empfangenen Ethernet- und CAN-Frames zeitlich in Relation gesetzt werden, welches für die noch folgenden Messungen hilfreich ist.

Es wurde zusätzlich noch die Software Wireshark (Wireshark Foundation) eingesetzt. Mit Hilfe von Wireshark ist eine detaillierte Analyse von Ethernet Frames möglich.

6.3 CAN Nachrichten Übertragungszeit

Da für die folgenden Messungen die Übertragungszeit einer CAN-Nachricht auf dem CAN Bus wichtig ist, muss diese genauer untersucht werden. Die Übertragungszeit in einem CAN-Netzwerk hängt hierbei von der Nachrichtenart (CAN 2.0A/B) bzw. der Nachrichtengröße sowie von der Geschwindigkeit des CAN-Busses ab. Ein CAN Bus, wie er im Prototypfahrzeug (VW Golf 7) verbaut ist, verfügt über eine Bruttobandbreite von 500 kbps und transportiert CAN 2.0A sowie CAN 2.0B Nachrichten. Um die minimale und maximale Nachrichtenübertragungszeit

ermitteln zu können, müssen die Längen der einzelnen CAN-Protokollfelder bekannt sein und der "Bit Stuffing" Mechanismus mit einbezogen werden. Wie im Abschnitt 2.1 beschrieben, verwendet das CAN-Protokoll das Bit Stuffing, um das Bit Timing weniger Präzise bestimmen zu müssen. Die Anzahl der eingefügten Stuffing Bits könnte für jede einzelne Nachricht bestimmt werden, in diesem Fall beschränke ich mich allerdings auf das Worst Case Szenario. Das Worst Case Szenario tritt dann ein, wenn bei den Bits der CAN-Nachrichten-ID sowie der maximal großen Nutzlast (8 Bytes) nach fünf identischen Bits eine Pegeländerung stattfindet und anschließend wieder fünf identische Bits folgen und sich der Vorgang bis zum Ende der Nachricht wiederholt. Hierbei muss beachtet werden, dass das eingefügte Stuffing Bit bereits zu den nächsten fünf identischen Bits gehört. Das Bit Stuffing wirkt sich dabei auf die Protokollfelder von "Start of Frame" inklusive dem "CRC" Feld aus. Bei einem CAN 2.0A Frame mit maximaler Payload wären das 19 Stuffing Bits und bei einem CAN 2.0B Frame wären es 25 Stuffing Bits. Anhand dieser Informationen kann die minimale und maximale Länge einer CAN-Nachricht berechnet werden (siehe Tabelle 6.1 und 6.2).

Feldname	**Länge in Bit**
Start of Frame	1
ID Message Identifier	11
Remote transmission bit	1
Identifier extension bit	1
reserved	1
Data length code (DLC)	4
Data	0-64
CRC	15
Bit Stuffing	0-19
CRC Delimiter	1
ACK-Slot	1
ACK-Delimiter	1
End of Frame	7
Inter frame space	3
Summe (min)	47
Summe (max)	130

Abbildung 6.1: Länge einer CAN2.0A Nachricht

Feldname	**Länge in Bit**
Start of frame	1
ID message identifier	11
Substitute remote request bit	1
Identifier extension bit	1
Extended identifier	18
Remote transmission bit	1
reserved	1
reserved	1
Data length code (DLC)	4
Data	0-64
CRC	15
Bit Stuffing	0-25
CRC Delimiter	1
ACK-Slot	1
ACK-Delimiter	1
End of Frame	7
Inter frame space	3
Summe (min)	67
Summe (max)	154

Abbildung 6.2: Länge einer CAN2.0B Nachricht

Mit Hilfe der Nachrichtengrößen kann nun die Übertragungszeit der Nachrichten berechnet werden. Setzt man in die Formel 6.1 die jeweiligen Nachrichtengrößen und die Busgeschwindigkeit ein, bekommt man für eine CAN2.0A Nachricht die minimale Übertragungszeit von 94 µs und für die maximale Übertragungszeit 260 µs. Bei einer CAN2.0B Nachricht beträgt die minimale Übertragungszeit 134 µs und die maximale Übertragungszeit 308 µs.

$$T = N/V$$

mit
Übertragungszeit T
Nachrichtengrösse N
Busgeschwindigkeit V

(6.1)

6.4 Messungen

Um die korrekte Funktionsweise der CAN-Filter und der RTE Gateways im Prototypfahrzeug nachzuweisen, werden im folgenden Abschnitt verschiedene Messungen vorgestellt. Mit diesen Messungen wird festgestellt, ob die CAN-Filter und die RTE Gateways im Prototypfahrzeugs im Stand sowie während der Fahrt wie geplant funktionieren und zusätzlich werden die Laufzeiten von Nachrichten im Netzwerk bestimmt. Dazu wurde in Labormessungen, Messungen im Stand sowie Messungen während der Fahrt unterschieden.

- Als erstes wurden der CAN-Filter sowie das RTE Gateway im Labor Messungen unterzogen. Mit diesen Messungen konnte eine Aussage darüber getroffen werden, wie groß die Laufzeiten einer Nachricht durch den CAN-Filter, sowie durch das RTE Gateway ist.
- Nach den Messungen im Labor wurden Messungen im Fahrzeuglabor der HAW-Hamburg durchgeführt. Dort wurden die Messungen durchgeführt, während das Prototypfahrzeug sich im Stand befand. Das heißt, mit dem Prototypfahrzeug wurde nicht gefahren, sondern es wurde nur die Zündung gestartet. Mit Hilfe dieser Messergebnisse konnte eine erste Aussage darüber getroffen werden, wie groß die Übertragungszeit einer Nachricht über das VW-Gateway ist. Zusätzlich wurde geprüft, ob die Laufzeiten einer Nachricht über das RTE-Backbone im Prototypfahrzeug den erwarteten Werten entsprach.
- Anschließend wurden Messungen auf dem Verkehrsübungsplatz durchgeführt, während sich das Prototypfahrzeug in Bewegung befand. Das Ziel der Messungen war es, eine

Aussage darüber treffen zu können, wie groß die Übertragungszeit einer Nachricht über das VW-Gateway ist, während sich das Prototypfahrzeug in Bewegung befindet.

Anschließend wurde analysiert, ob und wie sich diese von der Übertragungszeit im Stand des Prototypfahrzeuges unterscheidet. Ein weiteres Ziel der Messungen auf dem Verkehrsübungsplatz war es, eine Aussage darüber treffen zu können, wie groß die Übertragungszeit einer Nachricht über das RTE-Backbone während der Fahrt ist und ebenfalls zu prüfen, ob sich diese von den Werten im Stand des Prototypfahrzeuges unterscheidet.

6.4.1 Messungen im Labor

6.4.2 Messergebnisse des CAN Filters im Labor

Der Aufbau der Software des CAN-Filters ist sehr einfach. Wie in der Abbildung 4.1 dargestellt, gibt es nur eine ISR in der jegliche Aufgaben durchgeführt werden. Aus diesem Grund kann die Verzögerung, welche durch die CAN-Filter Software erzeugt wird, gemessen werden, indem die Laufzeit der ISR gemessen wird. Dazu wird am Anfang sowie am Ende der ISR mit Hilfe von GPIO Pins eine Signalschwankung erzeugt. Die Signalschwankung kann anschließend mit einem Oszilloskop gemessen werden. In den Abbildung 6.3 und 6.4 sind Screenshots aus dem Oszilloskop zu sehen. Die mit dem Oszilloskope gemessene minimale Laufzeit von $23.79\,\mu s$ ist in der Abbildung 6.3 dargestellt und in der Abbildung 6.4 ist die maximal gemessene Laufzeit der ISR von $31.59\,\mu s$ zu sehen. Die Laufzeit der ISR ist nicht abhängig von der Größe einer CAN Nachricht, da innerhalb der ISR nur die CAN ID der empfangen Nachricht abgefragt wird und anschließend in der Blacklist danach gesucht wird. Die ISR wird ebenfalls nicht vom Empfang der CAN-Nachricht bzw. vom Senden der Nachricht gebremst, da dies vom IO Modul übernommen wird. Die maximale Laufzeit von $31.59\,\mu s$ in Relation zur Frequenz der eintreffenden CAN Nachrichten ist sehr klein. Auf einem CAN Bus mit einer Übertagungsgeschwindigkeit von 500 kbps bräuchte eine minimale CAN Nachricht mit der Größe von 47 Bits ohne Bit Stuffing $94\,\mu s$. Selbst wenn auf beiden Interfaces zeitgleich Nachrichten eintreffen, hat der CAN Filter beide CAN Nachricht verarbeitet bis die nächste CAN Nachricht eintreffen würde.

Die Laufzeit einer Nachricht, welche über den CAN-Filter transportiert wird, wurde ebenfalls in einer Labormessung bestimmt. Dazu wurden bei der ersten Messung 3000 CAN-Nachrichten auf eines der zwei CAN-Interface des CAN-Filter geschickt und die Zeit jeweils vor sowie nach dem CAN-Filter gemessen. In dem in der Abbildung 6.5 dargestellten Messaufbau ist zu sehen, dass die CAN-Nachrichten nur auf einem der beiden CAN Anschlüsse des CAN-Filters eintreffen. Aus dieser Messung lässt sich die Verzögerung berechnen, welche vom CAN-Filter verursacht wird. In der Tabelle 6.1 wird das Ergebnis der Messung präsentiert. Aus der Messung

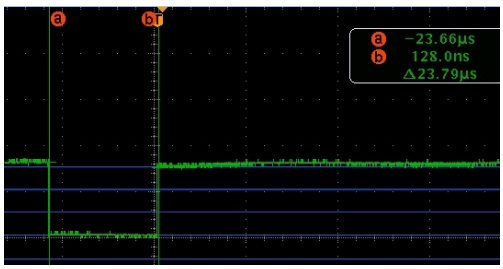

Abbildung 6.3: minimale Laufzeit der ISR

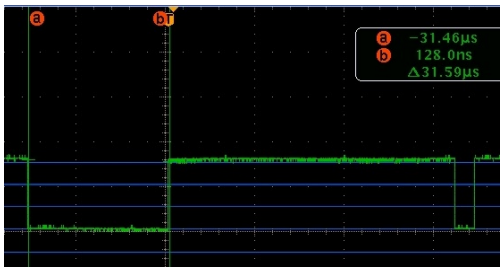

Abbildung 6.4: maximale Laufzeit der ISR

hat sich ergeben, dass eine Standard CAN-Nachrichten (CAN 2.0A) Laufzeit bei einer Payload von 8 Byte zwischen 245 µs-255 µs liegt. Bei einer erweiterten CAN-Nachricht (CAN 2.0B), welche ebenfalls mit einer Payload von 8 Byte, liegt die Laufzeit zwischen 286 µs-289 µs. Von diesen Zeiten muss anschließend die theoretische Übertragungszeit einer CAN-Nachricht abgezogen werden, damit die Verzögerungszeit des CAN-Filters übrig bleibt. Bei einer Standard CAN-Nachricht mit 8 Byte Payload, muss also von der Zeit 245 µs-255 µs die 222 µs (0 Stuff Bits) - 260 µs (19 Stuff Bits) abgezogen werden, welche die theoretische Übertragungszeit einer Standard CAN-Nachricht darstellt. Bei einer erweiterten CAN-Nachricht mit 8 Byte Payload muss von der Zeit 286 µs-289 µs die Übertragungszeit von 260 µs (0 Stuff Bits) - 296 µs (19 Stuff Bits) abgezogen werden. Im Idealfall hat der CAN-Filter also maximal eine Verzögerung von 23 µs-33 µs. In der Tabelle 6.1 ist ebenfalls ein erwarteter Wert eingetragen. Dieser Wert setzt sich aus der theoretischen Übertragungszeit einer CAN-Nachricht + 30 µs zusammen, welche die maximale Laufzeit der ISR im CAN-Filter darstellt. Die gemessenen Werte liegen unterhalb der erwarteten Werte. Da die Laufzeit der CAN Nachrichten nach Bearbeitung durch dem CAN-Filter nur wenige Mikrosekunden größer ist als ohne CAN-Filter, sollten keine zeitlichen Probleme mit dem VW-Gateway auftreten.

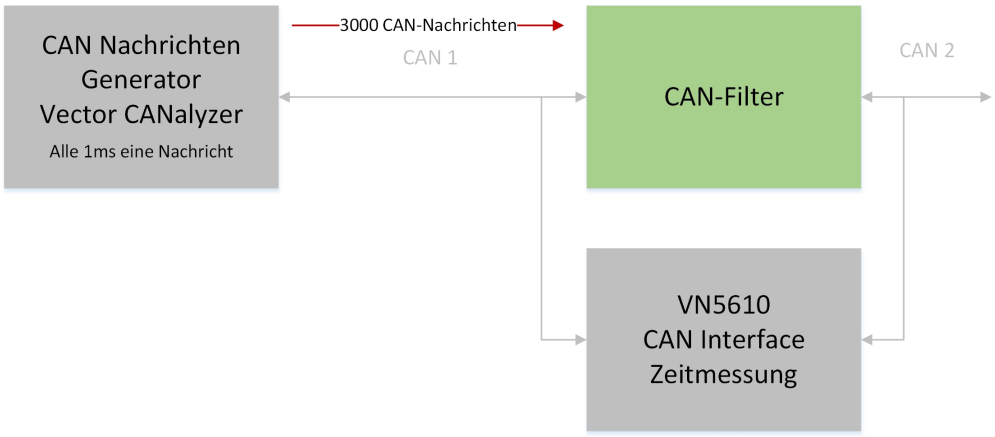

Abbildung 6.5: Messaufbau der CAN-Filter Messung mit 3000 CAN-Nachrichten auf einem CAN-Interface

CAN ID	Nachrichten-anzahl	minimale Latenz µs	maximale Latenz µs	durchschnittliche Latenz µs	erwarteter Wert µs
0x30c	369	247	250	248,40	252-290
0x6b2	370	244	247	245,49	252-290
0x700	370	250	253	251,49	252-290
0x17330510	369	287	290	288,66	288-338
0x17331910	369	286	289	287,67	288-338
0x17331A10	369	286	289	287,75	288-338
0x17331B10	369	285	288	286,62	288-338
0x17332110	369	286	289	287,64	288-338

Tabelle 6.1: Laufzeit der CAN-Nachrichten welche durch den CAN-Filter bearbeitet wurden. Nachrichten kommen nur auf einem CAN-Interface an.

Da die CAN-Filter jedoch zwei CAN-Interfaces haben, auf denen auch fast zeitgleich CAN-Nachrichten eintreffen können, muss diesbezüglich ebenfalls eine Messung durchgeführt werden. Dazu wurde ein etwas veränderter Messaufbau zur ersten CAN-Filter Messung benutzt. In der Abbildung 6.6 ist zu sehen, dass diesmal zwei CAN-Nachrichten Generatoren benutzt werden, um das fast zeitgleiche Eintreffen von CAN-Nachrichten auf dem CAN-Filter zu erzwingen. Die Uhren der CAN Generatoren sind nicht synchronisiert, weshalb der zweite CAN Generator die Zeit variiert wann er die CAN Nachricht los schickt. Hierzu wurde als zweiter CAN-Nachrichten Generator eine selbst entwickelte Software genutzt, die nacheinander zehn

mal alle 1000 µs eine Nachricht sendet, danach zehn mal alle 750 µs, dann zehn mal alle 500 µs und zum Schluss zehn mal alle 300 µs eine Nachricht sendet. Anschließend fängt der CAN-Nachrichten Generator wieder bei 1000 µs an. Zum einfachen Prüfen, ob alle Nachrichten angekommen sind, wird nach jeder gesendeten Nachricht die Payload um eins erhöht. Es wird jedes mal eine CAN-Nachricht mit der ID 12b und einer Payload von acht Byte gesendet. Im CAN-Nachrichten Generator im Vector CANalyzer hat sich nichts verändert. Dieser sendet alle 1000 µs eine CAN-Nachricht mit unterschiedlicher ID und einer Payload von 8 Byte. Dadurch, dass auf dem selbst entwickelten CAN-Nachrichten Generator die Zeit variiert, wann die Nachricht gesendet wird, wird irgendwann ein fast zeitgleiches Eintreffen von zwei CAN-Nachrichten auf dem CAN-Filter erzwungen. Da das Messgerät die Zeit über alle Interfaces synchronisiert, kann das fast zeitgleiche ankommen in den Messergebnissen nachgewiesen werden. In der Tabelle 6.2 ist zu sehen, dass die maximalen Werte der Übertragungszeit wesentlich höher sind sobald CAN-Nachrichten auf 2 Seiten des CAN-Filters eintreffen. Diese Werte sind jedoch nur Ausnahmen, da die durchschnittliche Übertragungszeit bei dieser Messung zwischen 318,08 und 380,09 liegt. Die minimale Latenz und die maximale Latenz in der Tabelle sind die Extremwerte, die in der Messung erreicht wurden. Dabei wird jedoch keine Aussage darüber getroffen wie oft diese Werte aufgetreten sind. Diese Werte sind jedoch nicht sehr oft erreicht worden, da die durchschnittliche Übertragungszeit ansonsten höher wäre. Die durchschnittliche Übertragungszeit ist im Gegensatz zum Empfang von Nachricht auf nur einem Interface höher, jedoch nur ca. 100 µs. Damit sind die Messungen bezüglich der CAN-Filter beendet und können im Prototypfahrzeug eingesetzt werden.

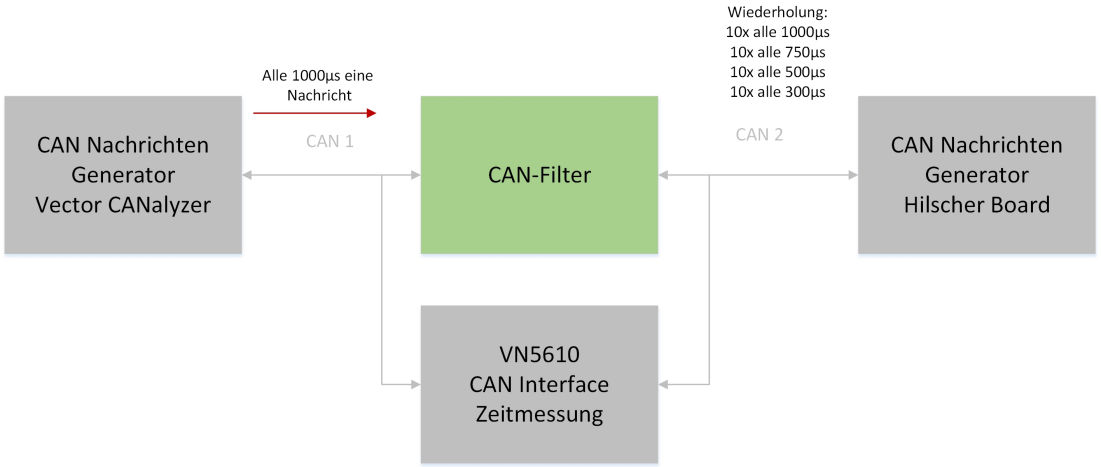

Abbildung 6.6: Messaufbau der CAN-Filter Messung mit CAN-Nachrichten auf beiden CAN-Interfacen

CAN ID	Nachrichten-anzahl	minimale Latenz µs	maximale Latenz µs	durchschnittliche Latenz µs	erwarteter Wert µs
0x12b	9814,5	265	535	318,08	252-290
0x30c	1592	247	1477	343,08	252-290
0x17330510	1592	287	939	356,43	288-338
0x17331910	1591	286	1235	372,41	288-338
0x17331b10	1591	285	1051	380,09	288-338

Tabelle 6.2: Laufzeit der CAN-Nachrichten welche durch den CAN-Filter bearbeitet wurden. Nachrichten kommen auf beiden CAN-Interfaces an.

6.4.3 Messergebnisse des RTE Backbones im Labor

Das RTE Gateway ist ein komplexeres System als der CAN-Filter. Das Design der RTE Gateway Software besteht aus mehreren ISR mit unterschiedlichen Prioritäten. Innerhalb der ISRs werden die Prozesse zur Umwandlung der Nachrichten gestartet. Außer diesen ISRs gibt es auf den Gateways nur den Synchronisationsprozess, welcher innerhalb des Schedulers abgearbeitet wird. In der Tabelle 6.3 wurden die fünf ISR, welche in einem RTE Gateway vorhanden sind, aufgelistet. Der Tabelle ist ebenfalls die Laufzeit sowie die Priorität der einzelnen ISR zu entnehmen. Die ISR “EthRecv0“ hat die höchste Priorität mit der Nummer eins. Sie wird beim Empfang eines Ethernet Frames von der Hardware gestartet und sorgt dafür, dass das Ethernet Frame von der Hardware in den Puffer der TT-Ethernet API kopiert wird. Anschließend wird von der TT-Ethernet API die ISR “CallbackTask“ (Priorität 9) gestartet, welche dafür zuständig ist die Ethernet Frames aus dem Buffer der TT-Ethernet API zu entfernen und fügt diese anschließend der Warteschlange der ISR “WorkerTask“ (Priorität 15) hinzu und startet diese. Wenn eine CAN-Nachricht empfangen wird, dann wird von der Hardware die “CANRecv“ (Priorität 10) ISR gestartet. Diese dient dazu, die CAN-Nachricht in die Warteschlange der “Worker“ ISR hinzuzufügen und anschließend wird diese ebenfalls gestartet. Die “WorkerTask"ISR wandelt CAN Nachrichten in Ethernet Nachrichten um und umgekehrt. Diese ISR läuft so lange bis die CAN und RTE Warteschlangen komplett leer sind. Im besten Fall bräuchte also eine CAN-Nachricht 45 µs-52 µs (CANRecv) + 258.2 µs-288.6 µs (WorkerTask CAN) über ein RTE Gateway bis die Nachricht per Ethernet verschickt wurde.

ISR Bezeichnung	Priorität	Laufzeit µs
EthRecv0	1	5
CallbackTask	9	30
CANRecv	10	45-52
Worker ISR CAN	15	258,2 - 288,6
Worker ISR Eth	15	180-190

Tabelle 6.3: Auflistung der ISR in einem RTE Gateway inklusive der Laufzeit und Priorität

Die internen Laufzeiten im RTE Gateway in Tabelle 6.3 wurden bestimmt, indem am Anfang und am Ende der ISR der Strom auf einem GPIO PIN gewechselt wurde. Diese Veränderung kann dann mit einem Oszilloskop gemessen werden. Die Zeiten der ISRs im Gateway sind wichtig, weil das Gateway nur aus diesen ISR besteht, welche die Übertragungszeit beeinflussen. Nachdem ich die internen Laufzeiten im RTE Gateway bestimmt habe, wurden erste Messungen zur Laufzeit einer Nachricht über das RTE Gateway durchgeführt. Dazu wurden für die erste RTE Gateway Messung, CAN-Nachrichten in Richtung dieses RTE Gateways geschickt. Ziel dabei war es, heraus zu finden wie groß die Laufzeit einer CAN-Nachricht ist, bis sie auf dem Ethernet versendet wurde. Zu diesem Zweck wurden vor dem RTE Gateway auf dem CAN-Bus sowie nach dem RTE Gateway, auf dem Ethernet die Zeit gemessen. In der Abbildung 6.7 wurde der Messaufbau zum besseren Verständnis grafisch dargestellt. In dem Test wurde eine CAN-Nachricht mit der CAN ID 0x6b2 wiederholt auf dem CAN Bus gesendet. Diese CAN-Nachricht wird im Gateway durch die ISR “CANRecv“ sowie die ISR “Worker ISR CAN“ verzögert. Die erwartete reine Gateway-Verzögerung für diese CAN-Nachricht lässt sich also aus den Laufzeiten dieser beiden ISRs bestimmen.

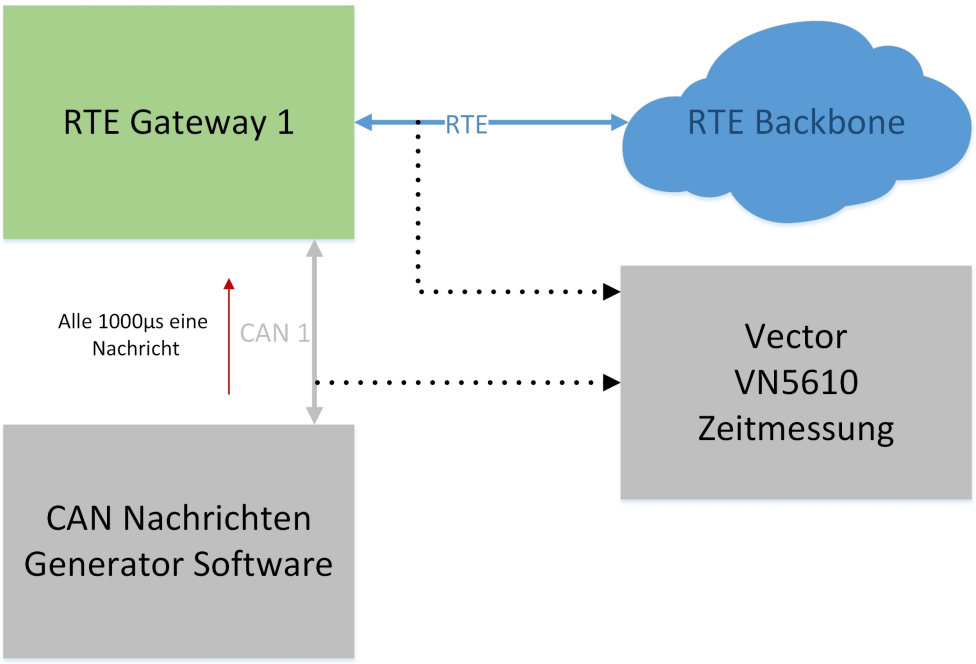

Abbildung 6.7: Messaufbau im Labor zur Erfassung der Umwandlungszeit

Das Messergebnis ist in Tabelle 6.4 zu sehen. Die durchschnittliche Übertragungszeit von 398.014 µs liegt damit also höher als der erwartete Wert von 303.2 µs-340.6 µs, welcher aus den addierten Laufzeiten der ISRs besteht. Der erwartete Wert der CAN Umwandlungszeit im Gateway ergibt sich aus der Formel 6.2. Dieser Test wurde nur mit einer CAN-Nachricht mit der ID 0x6b2 durchgeführt, da die sich die Laufzeiten mit anderen CAN IDs nicht großartig verändern würden.

$$T = TR + TW$$

mit
erwartete CAN Umwandlungszeit T
CANRecv ISR Zeit TR
Worker ISR CAN Zeit TW

(6.2)

CAN ID	Nachrichten-anzahl	minimale Latenz µs	maximale Latenz µs	durchschnittliche Latenz µs	erwarteter Wert µs
0x6b2	2219	362	706	398,014	303,2-340,6

Tabelle 6.4: Laufzeit der CAN-Nachrichten, welche durch das RTE Gateway bearbeitet wurden. Vom CAN-Bus zum Ethernet.

Da es auf dem Gateway aber auch dazu kommen kann, dass zeitnah eine CAN-Nachricht sowie eine Ethernet Nachricht eintrifft, muss getestet werden, ob sich die Latenzen in dem Fall erhöhen. Aus diesem Grund wurden im Labor zwei CAN-Nachrichten Generatoren benutzt, um das fast zeitgleiche Ankommen von Nachrichten auf dem Gateway zu simulieren. Wie bei den Messungen des CAN-Filters schon erklärt, kann mit Hilfe des Messgerätes in den Ergebnissen anhand der Zeit kontrolliert werden, dass Nachrichten zeitgleich eintreffen. Das Messgerät synchronisiert die Zeit über alle Messeingänge. In der Abbildung 6.8 ist der Messaufbau für die Messung der Übertragungszeit, wenn auf beiden Interfaces Nachrichten empfangen werden, dargestellt. Auf dem CAN-Bus des zu messenden Gateways eins, wurden alle 1000 µs eine CAN-Nachricht mit der ID 0x6b2 von der Vector CANalyzer Software geschickt (CAN-Nachrichten Generator 1). Damit auf dem Ethernet Interface der Empfang von Nachrichten simulierten werden konnte, müssen auf dem Ethernet Nachricht gesendet werden. Aus praktischen Gründen wurde ein zweites RTE Gateway genutzt, welches übersetzte CAN-Nachrichten auf dem Ethernet verschickt. Die CAN-Nachrichten wurden dabei von einer selbst entworfenen Software generiert und verschickt, welche bereits bei der Ermittlung der Nachrichtenlaufzeit bei einem CAN-Filter benutzt wurde. Diese Software variiert die Zeit beim verschicken der CAN-Nachricht mit der ID 0x3a1, zwischen 300 µs und 1000 µs und provoziert so ein fast zeitgleiches eintreffen von CAN- und Ethernet-Nachrichten auf dem Gateway eins. Dies kann in den Messergebnissen anhand der Ankunftszeit nachvollzogen werden. Es wurde für die CAN-Nachricht mit der ID 0x6b2 gemessen, wie lange die Laufzeit vom CAN-Bus zum Ethernet ist und für das Ethernet Frame, welches die CAN-Nachricht mit der ID 0x3a1 beinhaltet, wurde bestimmt, wie groß die Laufzeit vom Ethernet zum CAN-Bus ist. Die gemessenen Zeiten wurden dabei durch das Messgerät Vector VN5610 auf alle Mess-Eingänge synchronisiert.

In der Tabelle 6.5 sind die Ergebnisse der Messung dargestellt. Es ist klar zu erkennen, dass die Laufzeiten deutlich ansteigen, sobald auf beiden Interfaces Nachrichten empfangen werden. Dies passiert, da für die “WorkerTask“ ISR keine genaue Vorhersage getroffen werden kann,

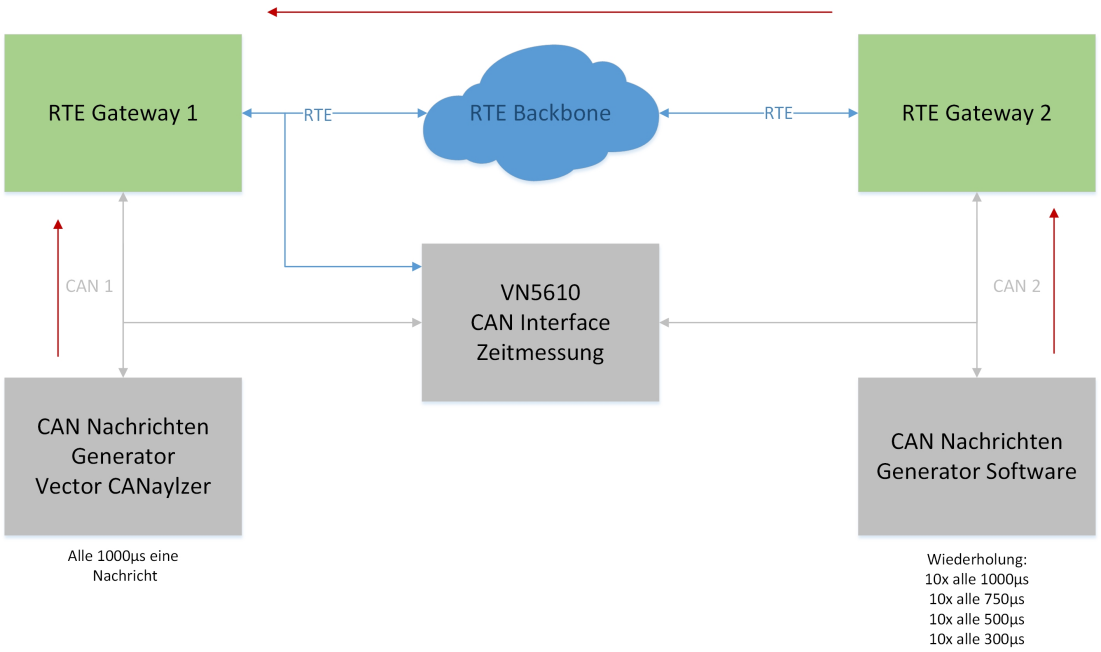

Abbildung 6.8: Messaufbau der RTE Gateway Messung mit Nachrichten auf beiden Interfacen

wie lange eine Nachricht braucht, wenn mehrere Nachrichten in den Warteschlangen sind. Die "WorkerTask" ISR arbeitet nach dem FIFO Prinzip und läuft solange bis keine Nachrichten mehr in den Warteschlangen vorhanden sind. Je mehr Nachrichten in der Warteschlange sind, desto länger muss die Nachricht auf ihre Umwandlung warten. Die Warteschlange füllt sich, sobald die Nachrichten schneller eintreffen als das Gateway die Nachrichten umwandeln kann. Da die CAN Umwandlungszeit, welche mit Hilfe der Formel 6.2 berechnet wurde, 303.2 µs-340.6 µs beträgt und CAN Nachrichten maximal 300 µs für die Übertragung auf dem CAN Bus benötigen, kann es vorkommen, dass CAN Nachrichten eintreffen während das RTE Gateway noch eine Nachricht umwandelt. Hinzu kommt noch das Ethernet Interface, auf dem ebenfalls Nachrichten empfangen werden. Es ist also nicht zu vermeiden, dass die Warteschlangen im Gateway sich füllen. Aus diesem Grund stimmen die Werte auch nicht mit den erwarteten Werten überein.

CAN ID	Nachrichten-anzahl	minimale Latenz µs	maximale Latenz µs	durchschnittliche Latenz µs	erwarteter Wert µs
0x6b2	897	358	10003	1140,29	303,2-340,60
0x3a1	716	439	1383	679,75	215-225

Tabelle 6.5: Laufzeit der CAN-Nachrichten welche durch das RTE Gateway bearbeitet wurden. Nachrichten kommen auf beiden CAN-Interfaces an.

6.4.4 Messungen im Prototypfahrzeug

Nachdem die Messungen im Labor abgeschlossen waren, wurden die ersten Messungen am Prototypfahrzeug im Stand vorgenommen. Damit die Laufzeiten für die Kommunikation von CAN-Nachrichten über den RTE Backbone ins Verhältnis zur Kommunikation über das VW-Gateway gesetzt werden können, muss als erstes eine Messung der Laufzeiten von CAN-Nachrichten über das VW-Gateway durchgeführt werden. Anschließend muss eine Messung der Laufzeiten von CAN-Nachrichen über den RTE Backbone durchgeführt werden. Hierzu muss eine Messung der Laufzeiten von CAN-Nachrichten durchgeführt werden, welche zwischen den drei Bussen E-CAN, I-CAN und K-CAN ausgetauscht werden. Die CAN-Busse E-CAN, I-CAN und K-CAN wurden explizit ausgewählt, da die Nachrichten, welche über diese CAN-Busse transportiert werden, keine sicherheitskritischen Auswirkungen haben. Am Patchfeld, welches im Kofferraum des Prototypfahrzeugs vorhanden ist, sind die drei CAN-Busse verfügbar und das Messgerät VN5610 von Vector kann dort für Messungen direkt angeschlossen werden. Da das Messgerät nur zwei CAN-Messspitzen hat, müssen für jeden Messaufbau drei Messungen durchgeführt werden, welche die Kommunikation zwischen den drei Bussen aufzeichnet. Die Messaufbauten unterteilen sich dabei in vier verschiedene Messsituationen. Messungen des VW-Gateways im Stand, Messung des VW-Gateways während der Fahrt, Messung des RTE-Backbone im Stand und die Messung des RTE-Backbones während der Fahrt. Es müssen also insgesamt im Prototypfahrzeug zwölf (vier mal drei) Messungen durchgeführt werden.

Hierzu wird für jede Messungen das Messgerät VN5610 von Vector mit zwei CAN Bussen verbunden und die Messung in der Vector CANoe Software gestartet. Ab diesem Zeitpunkt speichert die Software alle Nachrichten, die auf den angeschlossenen CAN-Bussen übertragen werden. Im Anschluss wurden in Excel die Laufzeiten, welche die CAN-Nachrichten für den Transport zwischen den CAN-Bussen benötigen, ausgerechnet. Dies kann nur mit CAN-

Nachrichten geschehen, die zwischen den angeschlossenen CAN-Bussen ausgetauscht werden.

Für die Messungen der Laufzeiten von CAN-Nachrichten über das VW-Gateway im Stand sowie während der Fahrt wird der gleiche Messaufbau verwendet. Die Abbildung 6.9 stellt den Messaufbau für die Messung der Kommunikation über das VW-Gateway dar. Für diese Messungen muss keine Veränderung im Prototypfahrzeug geschehen. Das Messgerät wird, wie oben beschrieben, direkt mit den am Patchfeld vorhanden Anschlüsse der CAN-Busse verbunden und die Messung anschließend gestartet.

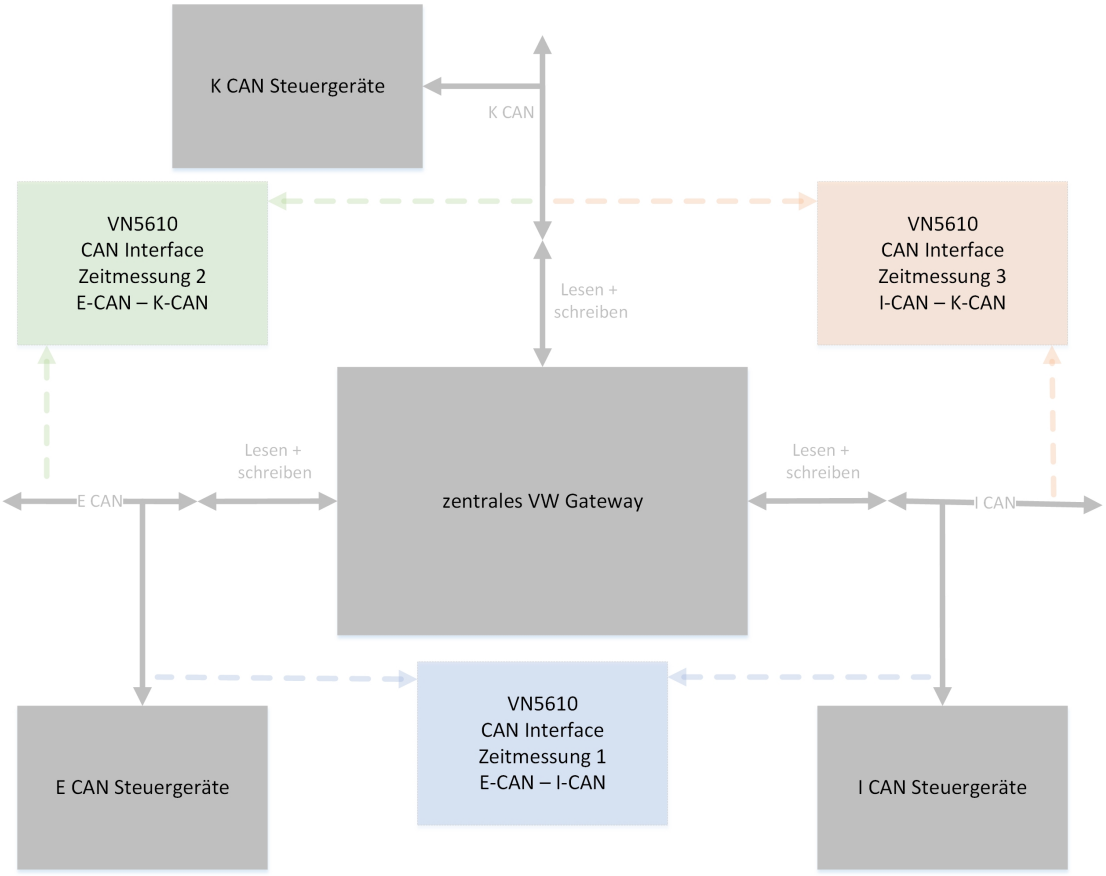

Abbildung 6.9: Messaufbau für die Messung der Übertragungszeit via VW-Gateway

Nachdem die Messungen zur Ermittlung der Laufzeit über das VW-Gateway durchgeführt wurden, wird eine Messung zur Bestimmung der Laufzeit von CAN-Nachrichten über den RTE-Backbone durchgeführt. In der Abbildung 6.10 wird der Messaufbau für die Messung der Laufzeiten über den RTE-Backbone dargestellt. Als erstes werden die CAN-Filter im Pro-

totypfahrzeug installiert, die in der Abbildung grün markiert sind. Danach werden die RTE Gateways per CAN-Kabel, welche in der Abbildung ebenfalls grün markiert sind, mit dem Patchfeld verbunden. Anschließend wird das Messgerät VN5610 ebenfalls mit den CAN-Bussen verbunden, welche mit den RTE Gateways verbunden sind.

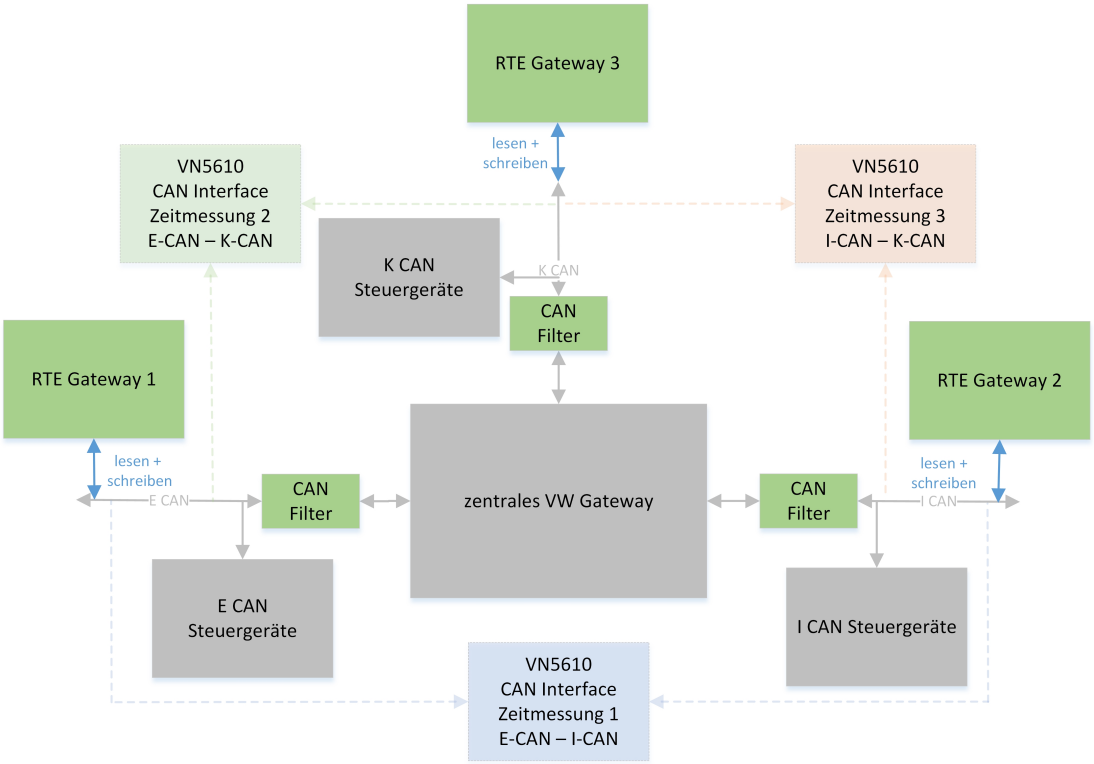

Abbildung 6.10: Messaufbau für die Messung der Übertragungszeit via RTE Backbone

Die Messungen im Prototypfahrzeug während des Standes wurden im Fahrzeuglabor des Department für Fahrzeugtechnik und Flugzeugbau der HAW-Hamburg durchgeführt. Dazu wurden erst drei Messungen mit dem VW-Gateway durchgeführt und danach drei Messungen mit dem RTE-Backbone. Für die Messungen während das Prototypfahrzeug sich in Bewegung befindet, dürfen keine Fehlermeldungen im Cockpit des Prototypfahrzeugs erscheinen. Ansonsten wäre es ein zu großes Risiko mit dem Prototypfahrzeug zu fahren. Des Weiteren muss für die Messungen während der Fahrt ein Privatgelände genutzt werden. Das Prototypfahrzeug hat nämlich keine gültige StVO Zulassung während die Kommunikation der CAN-Busse über den RTE-Backbone abläuft. Ebenfalls dürfen keine dritten Verkehrsteilnehmer durch das Prototypfahrzeug gefährdet werden. Sobald die Veränderungen wieder rückgängig gemacht worden sind, d.h. die Kommunikation geschieht wieder über das VW-Gateway, gilt die StVO

Zulassung wieder. Aus diesen Gründen wurde der Verkehrsübungsplatz der Verkehrswacht Hamburg ausgewählt. Dort gibt es einen abgetrennten Sonderübungsplatz, auf den während der Durchführung der Messungen keine anderen Fahrzeuge Zutritt haben und somit ungestört getestet werden kann. Auf der Abbildung 6.11 ist der Verkehrsübungsplatz dargestellt und der Sonderübungsplatz sichtbar. Die Strecke, welche während der Messung gefahren wurde, wurde in Rot auf dem Sonderübungsplatz eingezeichnet.

Die Messungen auf dem Verkehrsübungsplatz wurden äquivalent zu den Messungen im Stand durchgeführt. Erst wurden drei Messungen mit dem VW-Gateway durchgeführt, danach drei Messungen mit dem RTE-Backbone. Der Messaufbau ist dabei äquivalent zu dem Messaufbau aus dem Fahrzeuglabor, der in Abbildung 6.9 und 6.10 dargestellt wird.

Abbildung 6.11: Verkehrsübungsplatz der Verkehrswacht Hamburg. Quelle: www.verkehrswacht-hamburg.de

Während der Messungen im Fahrzeuglabor sowie auf dem Verkehrsübungsplatz wurde folgendes Vorgehen festgelegt.

1. Auto aus
2. Vergewissern, dass keine CAN-Nachrichten mehr verschickt werden (Vector CANoe Software)
3. Starten der Messung

4. Auto starten
5. Messung im Fahrzeuglabor für 50 Sekunden laufen lassen / eine Runde auf dem abgetrennten Bereich des Verkehrsübungsplatz fahren (ca. 50 Sekunden)
6. Messung anhalten
7. Auto ausschalten

Diese zwölf Messungen werden nach der Präsentation der Messergebnisse analysiert. Anschließend werden die Laufzeiten des RTE-Backbones ins Verhältnis zu den Laufzeiten des VW-Gateways gesetzt.

6.4.5 Messergebnisse des VW-Gateways im Prototypfahrzeug

Wie bereits beschrieben wurden im Prototypfahrzeug Messungen im Stand sowie während der Fahrt durchgeführt. Alle Messungen wurden wie im Messaufbau in der Abbildung 6.9 dargestellt durchgeführt. Die Ergebnisse der Messungen wurden in Excel aufgearbeitet. In den Tabellen 6.6, 6.7 und 6.8 sind die gemessenen Werte aufgelistet, welche während der Messungen im Stand und während der Fahrt des Prototypsfahrzeug aufgenommen wurden. Den Tabellen wurde eine Spalte für die Übertragungszeit der CAN Nachricht ohne des VW-Gateways hinzugefügt. Dieser setzt sich bei der Messung des VW-Gateways aus der theoretischen Übertragungszeit zusammen, welche eine CAN-Nachricht auf dem CAN-Bus benötigt. Da noch kein Wert bekannt für die Laufzeit des VW-Gateways ist, wurde das VW-Gateway bei der CAN Übertragungszeit nicht in die Berechnung mit einbezogen und somit ist der "transmit time" Wert hier immer geringer als der tatsächlich gemessene Wert.

Messung	**CAN ID**	**Payload**	**Nachrichten-anzahl**	**min** µs	**max** µs	**avg** µs	**transmit time** µs
Stand	0x30c	8	344	281	569	321,16	222
Fahren	0x30c	8	859	277	602	318,26	222
Stand	0x17330510	3/4/8	20	222	423	285,40	178-258
Fahren	0x17330510	3/4/8	56	222	561	291,57	178-258
Stand	0x17331910	3/4/5/8	24	233	640	337,58	178-258
Fahren	0x17331910	3/4/5/8	69	232	867	335,38	178-258
Stand	0x17331b10	3/4/8	20	230	419	287,65	178-258
Fahren	0x17331b10	3/4/8	61	231	606	303,77	178-258

Tabelle 6.6: Vergleich der Kommunikation zwischen dem E-CAN und dem I-CAN via VW-Gateway im Stand/Fahren

Messung	**CAN ID**	**Payload**	**Nachrichten-anzahl**	**min** µs	**max** µs	**avg** µs	**transmit time** µs
Stand	0x30c	8	343	271	572	288,78	222
Fahren	0x30c	8	783	269	701	302,34	222
Stand	0x324	8	343	276	550	304,27	222
Fahren	0x324	8	783	273	791	323,09	222
Stand	0x395	8	206	274	436	282,23	222
Fahren	0x395	8	469	267	792	291,55	222
Stand	0x397	8	206	274	562	307,75	222
Fahren	0x397	8	470	271	579	297,45	222
Stand	0x6b4	8	103	268	1260	518,90	222
Fahren	0x6b4	8	243,5	268	725	319,81	222
Stand	0x6b7	8	20	272	348	282,20	222
Fahren	0x6b7	8	49	270	445	284,67	222
Stand	0x17331910	3/4/5/8	23	222	330	264,00	178-258
Fahren	0x17331910	3/4/5/8	53	222	467	260,06	178-258
Stand	0x17331b10	3/4/8	20	221	713	295,60	178-258
Fahren	0x17331b10	3/4/8	47	221	520	287,62	178-258

Tabelle 6.7: Vergleich der Kommunikation zwischen dem E-CAN und dem K-CAN via VW-Gateway im Stand/Fahren

Messung	**CAN ID**	**Payload**	**Nachrichten-anzahl**	**min** µs	**max** µs	**avg** µs	**transmit time** µs
Stand	0x30b	8	414	296	618	322,17	222
Fahren	0x30b	8	977	289	679	330,93	222
Stand	0x481	8	41	286	847	379,59	222
Fahren	0x481	8	98	285	681	352,65	222
Stand	0x484	8	207	270	744	291,38	222

Fahren	0x484	8	488	271	683	309,67	222
Stand	0x485	8	207	265	759	316,53	222
Fahren	0x485	8	488	263	902	334,70	222
Stand	0x551	8	104	307	389	341,26	222
Fahren	0x551	8	244	270	600	306,70	222
Stand	0x5f0	8	103	296	585	323,26	222
Fahren	0x5f0	8	264	287	778	316,29	222
Stand	0x643	8	21	291	853	520,48	222
Fahren	0x643	8	49	289	512	334,59	222
Stand	0x650	8	23	282	878	408,43	222
Fahren	0x650	8	53	279	566	341,02	222
Stand	0x65a	8	20	284	547	355,3	222
Fahren	0x65a	8	54	280	867	409,06	222
Stand	0x668	8	103	286	838	339,37	222
Fahren	0x668	8	244	280	617	310,24	222
Stand	0x6b4	8	104	288	332	294,21	222
Fahren	0x6b4	8	244	281	547	309,63	222
Stand	0x6b7	8	21	291	601	343,81	222
Fahren	0x6b7	8	49	286	551	333,78	222
Stand	0x6b8	8	21	295	889	399,29	222
Fahren	0x6b8	8	49	295	1056	412,61	222
Stand	0x16a9540a	8	20	346	523	385,55	258
Fahren	0x16a9540a	8	49	343	895	379,55	258
Stand	0x17330110	3/4/5/6/7/8	22	235	404	294,18	178-258
Fahren	0x17330110	3/4/5/6/7/8	56	233	563	282,68	178-258
Stand	0x17330810	3/4/5/8	42	243	372	310	178-258
Fahren	0x17330810	3/4/5/8	106	235	604	310,98	178-258
Stand	0x17330910	3/4/5/8	22	224	540	275,95	178-258
Fahren	0x17330910	3/4/5/8	56	224	545	266	178-258
Stand	0x17330c10	3/5/8	23	244	534	304,39	178-258
Fahren	0x17330c10	3/5/8	56	244	426	292,13	178-258
Stand	0x17330d10	3/4/5/8	24	224	356	272,88	178-258

Fahren	0x17330d10	3/4/5/8	54	223	336	266,13	178-258
Stand	0x17330e10	3/8	21	231	393	280,76	178-258
Fahren	0x17330e10	3/8	49	230	639	276,78	178-258
Stand	0x17330f10	4/5/6/7/8	30	228	551	311,93	194-258
Fahren	0x17330f10	4/5/6/7/8	140	250	587	341,24	194-258
Stand	0x17331110	3/4/5/8	49	244	513	306,39	178-258
Fahren	0x17331110	3/4/5/8	116	243	1078	372,56	178-258
Stand	0x17331210	3/5/8	21	237	423	322,43	178-258
Fahren	0x17331210	3/5/8	49	255	1146	449,29	178-258
Stand	0x17331310	3/4/5/8	22	273	440	361,09	178-258
Fahren	0x17331310	3/4/5/8	53	227	602	315,64	178-258
Stand	0x17332810	3/4/5/6/7/8	24	248	471	328,63	178-258
Fahren	0x17332810	3/4/5/6/7/8	66	246	528	347,03	178-258
Stand	0x17332811	6/8	3	307	813	490,33	226-258
Fahren	0x17332811	6/8	9	310	491	363,56	226-258
Stand	0x17333110	2/3/4/5/6/7/8	22	211	510	325,23	162-258
Fahren	0x17333110	2/3/4/5/6/7/8	60	210	496	287,98	162-258
Stand	0x17333111	2/4/5/8	2	382	445	413,5	162-258
Fahren	0x17333111	2/4/5/8	11	248	329	300,73	162-258
Stand	0x17333210	2/3/4/5/8	27	239	436	307,74	162-258
Fahren	0x17333210	2/3/4/5/8	69,5	223	525	303,30	162-258
Stand	0x17333211	2/3/5/8	1	273	273	273	162-258
Fahren	0x17333211	2/3/5/8	48	223	858	348,06	162-258
Stand	0x17333410	3/5/8	25	240	977	431,8	178-258
Fahren	0x17333410	3/5/8	58	237	603	370	178-258

Tabelle 6.8: Vergleich der Kommunikation zwischen dem I-CAN und dem K-CAN via VW-Gateway im Stand/Fahren

6.4.6 Messergebnisse des RTE Backbones im Prototypfahrzeug

Um eine Aussage über das Zeitverhalten des RTE-Backbones zu treffen, wurden wie bereits erläutert sechs Messungen durchgeführt. Die Ergebnisse der Messungen wurden in Excel aufgearbeitet. In den Tabellen 6.9, 6.10 und 6.11 sind die Werte aufgelistet, welche im Stand

sowie während der Fahrt mit dem Prototypfahrzeug gemessen wurden. Den Tabellen wurde die Spalte “expected“ hinzugefügt, welche die erwartete Verzögerungszeit via des RTE-Backbones darstellt. Die Zeit wurde mit Hilfe der Formel 6.3 berechnet und den Zeiten der ISRs im Gateway, welche in Tabelle 6.3 dargestellt wurden.

$$T = TCR + TCW + TRR + TRW \tag{6.3}$$

erwartete RTE Verzögerungszeit T
CANRecv ISR Zeit TCR
Worker ISR CAN Zeit TCW
RTE Recv ISR Zeit TRR
Worker ISR RTE Zeit TRW

Messung	**CAN ID**	**Payload**	**Nachrichten-anzahl**	**min** µs	**max** µs	**avg** µs	**expected** µs
Stand	0x30c	8	845	830	4324	1022,66	755,2-802,6
Fahren	0x30c	8	995	819	9535	933,82	755,2-802,6
Stand	0x17330510	3/4/8	51	785	11052	1320,63	711,2-838,6
Fahren	0x17330510	3/4/8	64	772	11222	994,45	711,2-838,6
Stand	0x17331910	3/4/5/8	57	773	1182	857,91	711,2-838,6
Fahren	0x17331910	3/4/5/8	78	776	14837	1354,69	711,2-838,6
Stand	0x17331b10	3/4/8	50,5	765	10454	1379,98	711,2-838,6
Fahren	0x17331b10	3/4/8	70	771	5941	1153,81	711,2-838,6

Tabelle 6.9: Vergleich der Kommunikation zwischen dem E-CAN und dem I-CAN via RTE-Backbone im Stand/Fahren

Messung	**CAN ID**	**Payload**	**Nachrichten-anzahl**	**min** µs	**max** µs	**avg** µs	**transmit time** µs
Stand	0x30c	8	852	840	13091	984,03	755,2-802,6
Fahren	0x30c	8	910	837	15926	1831,03	755,2-802,6
Stand	0x324	8	851	820	32966	1438,92	755,2-802,6
Fahren	0x324	8	910	844	16075	1857,87	755,2-802,6
Stand	0x395	8	511	840	3468	936,83	755,2-802,6

Fahren	0x395	8	544,5	838	3266	941,60	755,2-802,6
Stand	0x397	8	511	838	1527	907,71	755,2-802,6
Fahren	0x397	8	545	840	4862	1080,63	755,2-802,6
Stand	0x6b4	8	255	903	6203	1143,29	755,2-802,6
Fahren	0x6b4	8	287,5	903	1542	1015,66	755,2-802,6
Stand	0x6b7	8	52	913	1656	1029,73	755,2-802,6
Fahren	0x6b7	8	58	912	2449	1328,41	755,2-802,6
Stand	0x17331910	3/4/5/8	58	790	16651	1115,79	711,2-838,6
Fahren	0x17331910	3/4/5/8	74	790	1793	966,82	711,2-838,6
Stand	0x17331b10	3/4/8	51	755	3039	1030,88	711,2-838,6
Fahren	0x17331b10	3/4/8	64	774	1379	971,69	711,2-838,6

Tabelle 6.10: Vergleich der Kommunikation zwischen dem E-CAN und dem K-CAN via RTE-Backbone im Stand/Fahren

Messung	**CAN ID**	**Payload**	**Nachrichten-anzahl**	**min** µs	**max** µs	**avg** µs	**transmit time** µs
Stand	0x30b	8	1008,5	938	1890	1018,65	755,2-802,6
Fahren	0x30b	8	1172,5	934	51037	1353,30	755,2-802,6
Stand	0x481	8	94	956	100826	5352,22	755,2-802,6
Fahren	0x481	8	103,5	954	29763	2458,61	755,2-802,6
Stand	0x484	8	490,5	952	9090	1202,11	755,2-802,6
Fahren	0x484	8	577,5	947	99634	1606,65	755,2-802,6
Stand	0x485	8	495	499	100825	2855,89	755,2-802,6
Fahren	0x485	8	575	272	100883	5088,93	755,2-802,6
Stand	0x551	8	253	947	31287	2535,80	755,2-802,6
Fahren	0x551	8	296	925	2458	1011,38	755,2-802,6
Stand	0x5f0	8	253	916	2208	1150,01	755,2-802,6
Fahren	0x5f0	8	357	921	4913	1209,81	755,2-802,6
Stand	0x643	8	50	1041	1885	1249,16	755,2-802,6
Fahren	0x643	8	56	952	25011	2486,43	755,2-802,6
Stand	0x650	8	50	960	98880	3805,24	755,2-802,6
Fahren	0x650	8	63	970	7294	1944,11	755,2-802,6

Stand	0x65a	8	50	1262	2218	1575,68	755,2-802,6
Fahren	0x65a	8	65	937	8620	1696,20	755,2-802,6
Stand	0x668	8	253	961	1482	1019,24	755,2-802,6
Fahren	0x668	8	295,5	935	9397	1516,33	755,2-802,6
Stand	0x6b4	8	252	917	7236	1359,65	755,2-802,6
Fahren	0x6b4	8	292,5	918	17315	1190,80	755,2-802,6
Stand	0x6b7	8	50	943	19119	1437,06	755,2-802,6
Fahren	0x6b7	8	59	934	17588	1419,02	755,2-802,6
Stand	0x6b8	8	50	1040	19115	1692,68	755,2-802,6
Fahren	0x6b8	8	59	1171	17819	1939,95	755,2-802,6
Stand	0x16a9540a	8	50	1129	2149	1421,12	791,2-838,6
Fahren	0x16a9540a	8	59	990	2158	1314,51	791,2-838,6
Stand	0x17330110	3/4/5/6/7/8	56	890	29757	3739,09	711,2-838,6
Fahren	0x17330110	3/4/5/6/7/8	84	907	48363	2638,95	711,2-838,6
Stand	0x17330810	3/4/5/8	105	905	1374	1037,63	711,2-838,6
Fahren	0x17330810	3/4/5/8	136	891	34045	1467,82	755,2-802,6
Stand	0x17330910	3/4/5/8	55	910	1320	989,69	711,2-838,6
Fahren	0x17330910	3/4/5/8	73	891	34704	1878,18	711,2-838,6
Stand	0x17330c10	3/5/8	59	892	1352	972,53	711,2-838,6
Fahren	0x17330c10	3/5/8	73	881	24383	1808,52	711,2-838,6
Stand	0x17330d10	3/4/5/8	54,5	897	1584	997,65	711,2-838,6
Fahren	0x17330d10	3/4/5/8	71	899	1422	993,41	711,2-838,6
Stand	0x17330e10	3/8	50	903	1290	996,18	711,2-838,6
Fahren	0x17330e10	3/8	65	888	4283	2018,71	711,2-838,6
Stand	0x17330f10	4/5/6/7/8	81	903	10825	1323,43	727,2-838,6
Fahren	0x17330f10	4/5/6/7/8	192	904	24357	1265,85	727,2-838,6
Stand	0x17331110	3/4/5/8	122	909	1699	991,98	711,2-838,6
Fahren	0x17331110	3/4/5/8	167	904	24185	1280,80	711,2-838,6
Stand	0x17331210	3/5/8	50	886	21017	1818,68	711,2-838,6
Fahren	0x17331210	3/5/8	69,5	896	2342	1089,19	711,2-838,6
Stand	0x17331310	3/4/5/8	53	901	1264	1007,66	711,2-838,6
Fahren	0x17331310	3/4/5/8	77,5	1001	2381	1333,91	711,2-838,6

Stand	0x17332810	3/4/5/6/7/8	73	903	83951	12610,04	711,2-838,6
Fahren	0x17332810	3/4/5/6/7/8	85,5	900	27219	1426,83	711,2-838,6
Stand	0x17332811	6/8	5	991	1123	1029,50	759,2-838,6
Fahren	0x17332811	6/8	42	929	21848	2637,49	759,2-838,6
Stand	0x17333110	2/3/4/5/6/7/8	50,5	893	1333	989,06	695,2-838,6
Fahren	0x17333110	2/3/4/5/6/7/8	64	896	74602	4653,21	695,2-838,6
Stand	0x17333111	2/4/5/8	5,5	936	1125	1004,40	695,2-838,6
Fahren	0x17333111	2/4/5/8	127	902	33226	2195,52	695,2-838,6
Stand	0x17333210	2/3/4/5/8	61,5	892	52704	2405,05	695,2-838,6
Fahren	0x17333210	2/3/4/5/8	83	891	132684	3420,20	695,2-838,6
Stand	0x17333211	2/3/5/8	3	937	965	951,33	695,2-838,6
Fahren	0x17333211	2/3/5/8	141,5	902	9213	1121,93	695,2-838,6
Stand	0x17333410	3/5/8	60	904	5574	2683,52	711,2-838,6
Fahren	0x17333410	3/5/8	72,5	901	2386	1034,99	711,2-838,6

Tabelle 6.11: Vergleich der Kommunikation zwischen dem I-CAN und dem K-CAN via RTE-Backbone im Stand/Fahren

6.5 Evaluierung

6.5.1 Bewertung des Zeitverhalten des CAN Filter

Das Zeitverhalten des CAN-Filters spielt im Prototypfahrzeug eine große Rolle, wenn die Kommunikation der CAN-Busse über das RTE-Backbone geschieht. Wie im Kapitel 6.4.4 bereits erläutert, produziert das Prototypfahrzeug Fehlermeldungen sobald CAN-Nachrichten bei dem VW-Gateway nicht rechtzeitig bzw. gar nicht ankommen. Aus diesem Grund ist es wichtig, dass die CAN-Nachrichten, welche über das RTE-Backbone transportiert werden, ebenfalls beim VW-Gateway ankommen. Damit keine Fehlermeldungen erzeugt werden, wurde das Zeitverhalten des CAN-Filters untersucht.

Aus den Messungen im Abschnitt 6.4.2 geht hervor, dass die Laufzeit der ISR des CAN-Filters höchstens 30 µs dauert. Im Vergleich zu der Übertragungszeit einer CAN-Nachricht auf dem CAN-Bus, welche im Worst Case bis zu 308 µs dauern kann (siehe 6.3), ist die Laufzeit der ISR im CAN-Filter sehr gering. Die Messungen im Labor zur Laufzeit einer CAN-Nachricht über den CAN-Filter bestätigen dies ebenfalls. So benötigt eine CAN-Nachricht über den CAN-

Filter unter Volllast durchschnittlich maximal 380.09 µs. Diese Laufzeit ist somit nur wenige Mikrosekunden größer als die maximale Übertragungszeit einer CAN-Nachricht auf dem CAN-Bus. Durch den entwickelten CAN-Filter entstehen aus diesem Grund keine Probleme und somit muss die Laufzeit einer CAN-Nachricht über den CAN-Filter für die weiteren Messungen nicht weiter beachtet werden.

6.5.2 Bewertung des Zeitverhalten des VW-Gateways

Wie im Abschnitt 6.3 dargestellt, benötigt eine CAN Nachricht im Worse Case 308 µs für die Übertragung. Im Stand beträgt die durchschnittliche Übertragungszeit zwischen den CAN-Bussen via des VW-Gateways, wie in den Tabellen 6.6, 6.7 und 6.8 dargestellt, zwischen 264 µs-520.48 µs. Während des Fahrens hingegen, beträgt die durchschnittliche Übertragungszeit zwischen den CAN-Bussen via des VW-Gateways, zwischen 260.06 µs-449.29 µs. Für die durchschnittliche Übertragungszeit macht es also keinen Unterschied, ob das Prototypfahrzeug steht oder ob es fährt, die Übertragungszeiten sind sehr nah beieinander. Da dies ein Blackbox Test ist, kann keine genaue Aussage zur Testüberdeckung getroffen werden. Aufgrund der Analyse der Daten während der Fahrt wird davon ausgegangen, dass die typischen Fälle abgedeckt sind. In den Tabellen sind die Prioritäten während der Arbitrierung nicht zu erkennen. Die Tabellen wurden nach den CAN IDs sortiert, jedoch variiert die durchschnittliche Übertragungszeit sehr stark und es ist keine Reihenfolge der Prioritäten an den Übertragungszeiten zu erkennen. Im Verhältnis zu den Übertragungszeiten einer CAN-Nachricht auf dem CAN-Bus mit 500 kbps, wo die Übertragungszeit zwischen 94 µs-308 µs dauert, sind die Verzögerungen durch das VW-Gateway im Stand und Fahren, welche zwischen 260 µs-520 µs liegen, nur wenige Mikrosekunden höher als ohne VW-Gateway.

In der folgenden Rechnung wird die Latenzzeit des VW-Gateways bestimmt. Als Beispiel wurde die CAN-Nachricht mit der ID 30C ausgewählt. Diese ist eine Standard CAN-Nachricht, welche immer eine 8 Byte Payload hat. Die einzige Varianz, die noch bleibt und nicht beeinflusst werden kann, ist die Anzahl der Stuff Bits, welche zwischen 0 und 19 Bit liegen kann. Mit Hilfe der Informationen aus dem Abschnitt 6.3 kann die theoretische Übertragungszeit von 222-260 µs der CAN-Nachricht 30C errechnet werden. Damit die Latenz des VW-Gateways bestimmt werden kann, wurde mit Hilfe der Formel 6.4 und der errechnete Übertragungszeit von 222-260 µs die Latenz des VW-Gateways berechnet. Für die CAN-Nachricht 30C liegt die VW-Gateway Latenz zwischen 58.26 µs und 96.26 µs. Die Varianz lässt sich durch die unterschiedliche Anzahl an Stuff Bits erklären.

$$G = L - T$$

mit
VW Gateway Latenz G
gemessene Übertragungszeit L
CAN Bus Übertragungszeit T

(6.4)

6.5.3 Bewertung des Zeitverhalten des RTE-Backbones

Wie in der Bewertung der Laufzeit vom VW-Gateway erläutert, dauert die Übertragung einer CAN-Nachricht maximal 300 µs. In den Tabellen 6.9, 6.10 und 6.11 ist zu erkennen, dass die Laufzeit einer Nachricht über das RTE-Backbone während das Fahrzeug steht, im Durchschnitt zwischen 857 µs-5352 µs liegt. Während der Fahrt ist zu erkennen, dass die Übertragungszeit einer Nachricht zwischen 933 µs-5141 µs dauert. Aus diesen Zeiten ist erkennbar, dass es keinen Unterschied macht, ob das Auto steht oder am fahren ist, da die Übertragungszeiten sich im selben Rahmen von 800 µs bis 5400 µs befinden.

Die Übertragungszeit aus den Tabellen kommen durch die verschiedenen Verzögerungen auf dem RTE-Backbone zustande. Damit eine Nachricht über das RTE-Backbone ankommt, muss sie durch das erste RTE Gateway, dann über den RTE Switch und als letztes durch das zweite RTE Gateway. Wie im Abschnitt 6.4.3 bereits dargestellt hat das erste RTE Gateway eine Verzögerung, bestehend aus der CAN ISR mit 45 µs bis 52 µs, welche die Nachricht empfängt und den CAN Worker mit 258.2 µs bis 288.6 µs, der die Nachricht in eine RTE Nachricht umwandelt und versendet. Der RTE Switch hat eine Verzögerung von 20 µs. Die Verzögerung im zweiten RTE Gateway kommt durch die RTE ISR zustande, die 30 µs braucht und durch den RTE Worker mit 190 µs, welcher die Nachricht in eine CAN-Nachricht umwandelt und diese auf den CAN-Bus raus schickt. Mit Hilfe der Formel 6.3 lässt sich also die RTE Verzögerungszeit berechnen, welche im besten Fall 533.2 µs bzw. 580.6 µs beträgt. Dazu kommt noch die Übertragungszeit einer CAN-Nachricht auf dem Bus von 222 µs-260 µs, die im Abschnitt 6.3 berechnet wurde und bis jetzt in der Rechnung noch nicht beachtet worden ist. Das bedeutet, dass die theoretische Übertragungszeit auf dem RTE-Backbone im Best Case 755.2 µs bzw. 840.6 µs liegt. Diese theoretischen Zeiten kommen den gemessenen Zeiten im besten Fall von 857 µs bzw. 933 µs sehr nah. In der Abbildung 6.12 sind die Verzögerungen, welche auf dem RTE-Backbone entstehen können, grafisch dargestellt.

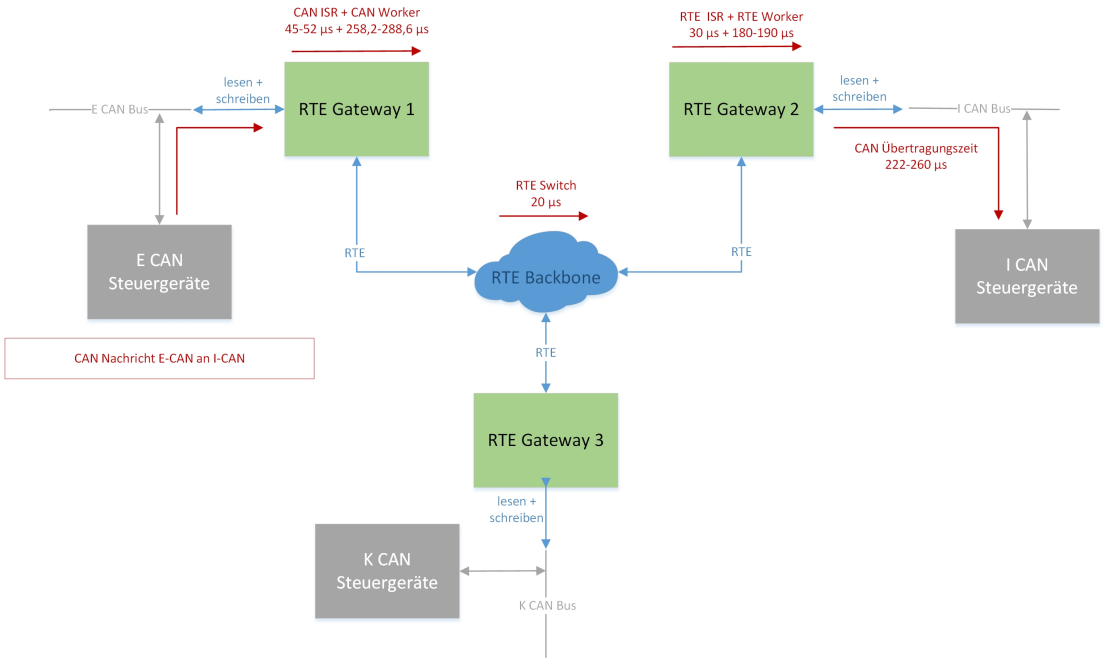

Abbildung 6.12: Darstellung der Verzögerungen, welche beim Übertragen einer Nachricht im RTE Netzwerk entstehen können

Während die Übertragungszeit einer Nachricht über das VW-Gateway bei 260 µs-520 µs liegt, ist die Übertragungszeit über das RTE-Backbone mit minimal 857 µs und maximal 5352 µs deutlich größer. Dies sind nur die extrem Werte, welche bei den Messungen aufgetreten sind. In den Tabellen mit den Messergebnissen ist zu erkennen, dass es sich hierbei nur um Ausreißer handelt. So ist der durchschnittliche Wert der Übertragungszeit wesentlich geringer als die Ausreißer. Dazu kommt, dass die angegebene Übertragungszeit auf dem RTE-Backbone sehr stark zwischen den angegeben Werten variiert. Durch die Arbitrierung sowie die 4 verschieden priorisierten ISR in den RTE Gateways können ungeplante Verzögerungen auftreten. Die Gateways sind so konstruiert, dass die CAN/RTE ISR, welche für den Empfang der Nachricht zuständig sind, die höchste Priorität haben. Beide Empfangen die Nachrichten, fügen sie einer der Warteschlangen hinzu und starten den Worker Task, welcher pro Durchgang eine Nachricht umwandelt. Der Worker Task läuft so lange, bis keine Nachrichten mehr in der CAN/RTE Warteschlange sind. Je mehr Nachrichten in den Warteschlangen sind, desto länger muss die Nachricht auf ihre Umwandlung warten. Die Warteschlangen füllen sich, sobald die Nachrichten schneller eintreffen als das Gateway die Nachrichten umwandeln kann. Dieses Design hat sich als am effizientesten herausgestellt, leider können jedoch Verzögerungen

auftreten, je nachdem wie viele Nachrichten im RTE Gateway behandelt werden müssen. Diese Verzögerungen machen sich im RTE Gateway durch die hohen Latenzen von bis zu 5400 µs bemerkbar. Da jedoch keine Fehlermeldungen im Cockpit des Prototypfahrzeuges erscheinen, sind die Toleranzgrenzen in den CAN-Steuergeräten vom Hersteller anscheinend so groß gewählt, dass selbst bei 5400 µs keine Probleme auftreten und keine Fehlermeldung generiert wird.

7 Zusammenfassung, Fazit und Ausblick

7.1 Zusammenfassung der Arbeit und Ergebnisse

Die Zielsetzung dieser Arbeit ist es, die Migration von Automobilbussystemen hin zu einem RTE-basierten Backbone-Netzwerk durch die Überprüfung der Ersetzbarkeit des herkömmlichen zentralen Kommunikationsgateways voran zu treiben. Zu diesem Zweck werden in dieser Arbeit die bereits vorhanden RTE-Gateways mit einer neuen Konfiguration versehen und zusätzliche CAN-Filter eingebaut. Nachdem die Kommunikation via RTE-Backbone sichergestellt ist, wird die Nachrichten Laufzeit untersucht. Dazu werden verschiedene Messungen der Kommunikation via des VW Gateways und der Kommunikation via des RTE-Backbones durchgeführt und anschließend evaluiert. Auf diese Weise kann eine Aussage darüber getroffen werden wie gut die Kommunikation via des RTE-Backbones, mit dezentralen Gateways, im Verhältnis zu der vom Automobil Hersteller implementierten Kommunikationslösung, mit zentralem Gateway, ist.

Die Arbeit beginnt mit der Vorstellungen des bereits im Automobil etablierten CAN-Busses und dessen Eigenschaften und Funktionsweise. Im Anschluss daran wird das RTE präsentiert und das Potential von RTE als automobiles Kommunikationswerkzeug erläutert sowie das bereits vorhanden RTE Gateway vorgestellt. Fortgeführt wird die Arbeit mit der Untersuchung des aktuellen Zustandes im Prototypfahrzeuges, auf den die Vorstellung des SOLL-Zustandes folgt und die Anforderungen diesbezüglich erstellt werden.

Im Kapitel 4 wird das Konzept sowie die Implementierung der Software für die CAN-Filter und die RTE Gateways erläutert. Hierzu wird als erstes die Anforderung an die CAN-Filter und die RTE Gateways dargestellt sowie anschließend die Funktionsweise der daraus entstandenen Software erläutert.

Im Kapitel 5 wird eine Qualitätssicherung der entwickelten Komponenten durchgeführt. Begonnen wird mit den Komponententests zur Sicherstellung der Funktionsweise der der einzelnen

isolierten Komponenten. Im Anschluss daran wird ein Integrationstest durchgeführt, der die Zusammenarbeit der einzelnen Komponenten testet. Im letzten Schritt der Qualitätssicherung wird ein Systemtest durchgeführt, welcher die Funktionsweise der entwickelten Komponenten im Prototypfahrzeug sicherstellt.

Die Messungen und die anschließende Evaluation erfolgen im Kapitel 6. Nach der Vorstellung weiterer Informationen zum Prototypfahrzeug wird die Übertragungszeit von CAN-Nachrichten auf dem CAN-Bus untersucht. Diese Informationen zur Übertragungszeit auf dem CAN-Bus sind wichtig, um im weiteren Verlauf Aussagen zur Übertragungszeit via RTE-Backbone formulieren zu können. Anschließend werden die entwickelten Komponenten im Labor genauer analysiert und die Messergebnisse präsentiert, um Informationen über das interne Zeitverhalten zu erhalten. Im Anschluss daran werden Messungen im Prototypfahrzeug durchgeführt. Hierbei werden sowohl Messungen zur Untersuchung des Zeitverhalten des VW Gateways sowie Messungen zur Untersuchung des Zeitverhalten des RTE-Backbones, während des Standes und der Fahrt des Prototypfahrzeuges, getätigt. Nachdem die Messungen durchgeführt und die Ergebnisse präsentiert werden, folgt eine Evaluierung der Messergebnisse.

7.2 Fazit

RTE ist ein guter Kandidat für ein zukünftiges, dezentrales automobiles Backbone-Netzwerk. RTE hat einen zeitlichen Determinismus im Bereich der harten Echtzeitanforderungen, bietet eine hohe Bandbreite und die Austauschbarkeit der Technologie auf der physikalischen Ebene ist gegeben. Somit kann RTE auch zukünftige wachsende Anforderungen der Automobilindustrie erfüllen. Für die Akzeptanz von RTE im Automobilbereich ist es wichtig, etablierte Bustechnologien an das RTE Netzwerk anbinden zu können. Die wirtschaftliche Belastung kann dadurch niedrig gehalten werden, dass die wesentlichen Teile des Wissens, der Arbeitskräfte und der Hard- und Software beibehalten werden können.

CAN-Steuergeräte werden in absehbarer Zeit nicht durch native RTE-Steuergeräte ersetzt, da CAN-Steuergeräte preiswert, zuverlässig und weit verbreitet sind. Damit eine sanfte Migrationsstrategie mit Weiterverwendung existierender CAN-Steuergeräte möglich ist, präsentiert diese Arbeit die ersten Versuche der Kommunikation von CAN-Steuergeräten via des RTE-Backbones. Die ersten Versuche einer Kommunikation via RTE-Backbone werden durch Zeitmessungen der neu konfigurierten bzw. neu entwickelten Komponenten und durch Messungen der Nachrichtenlaufzeiten via VW Gateway sowie via des RTE-Backbones vervollständigt.

Die im Rahmen dieser Arbeit entwickelte Kommunikation von CAN-Steuergeräten via RTE-Backbone verwendet eine angepasste Konfiguration der von Jan Depke im Rahmen seiner Master-Thesis (Depke (2015)) entwickelten RTE Gateways und wurde zu Testzwecken im Prototypfahrzeug der Core Arbeitsgruppe eingesetzt. Während der Qualitätssicherung stellte sich heraus, dass die Möglichkeit besteht CAN-Steuergeräte via RTE-Backbone kommunizieren zu lassen. Die Untersuchung und Auswertung der Kommunikation via RTE zeigt, dass die Kommunikation höhere Nachrichtenlaufzeiten als eine rein auf dem CAN-Bus bestehende Kommunikation hat, jedoch entstehen keine Fehlermeldungen im Prototypfahrzeug und die Nachrichten kommen an. Ebenfalls zeigt die Auswertung, dass die RTE Gateways bei hoher Frequenz an Nachrichten überlastet werden können und das führt zu größeren Nachrichtenlaufzeiten inklusive einiger zeitlicher Ausreißer. Dies hängt mit den Umwandlungszeiten der Nachrichten auf den RTE Gateways zusammen und das Problem könnte durch eventuelle zukünftige Optimierungen der Umwandlungszeiten behoben werden.

7.3 Ausblick auf zukünftige Arbeiten

Die auf dieser Arbeit basierenden zukünftigen Arbeiten können zwei unterschiedliche Themengebiete haben: die Optimierung des Laufzeitverhaltens der RTE Gateways und die Ausdehnung des praktischen Einsatzes der bestehenden CAN Kommunikation via RTE auf weitere CAN-Nachrichten.

Die Optimierung der Laufzeiten im RTE Gateway kann z.B. darin bestehen, die Laufzeiten der Umwandlungsprozesse von CAN nach RTE und umgekehrt zu beschleunigen. Aktuell sind die Umwandlungszeiten immer noch sehr hoch im Vergleich zur reinen Übertragung auf dem CAN-Bus. Wenn diese Zeiten optimiert werden können, würde der Einsatz der RTE Technologie noch interessanter werden als sie ohnehin schon ist. Somit würde die RTE Technologie vom Zeitverhalten noch dichter an die Übertragungszeit auf den CAN-Bussen heran kommen.

Die Ausdehnung des praktischen Einsatzes der bestehenden CAN Kommunikation via RTE kann z.B. die Ausweitung der via RTE übertragenen CAN-Nachrichten geschehen. Hierbei muss darauf geachtet werden, dass diese Ausweitung nur Stück für Stück geschieht und anschließend sichergestellt werden muss, dass diese CAN Nachrichten korrekt übertragen werden.

Abkürzungsverzeichnis

AFDX Avionics Full DupleX Switched Ethernet

BAG Bandwidth Allocation Gap

BE Best Effort

CAN Controller Area Network

CM Compression Master

CoS Class of Service

CSMA/CR Carrier Sense Multiple Access / Collision Resolution

CT Critical Traffic

CTID Critical Traffic Identifier

ECU Electronic Control Unit

HAW-Hamburg Hochschule für Angewandte Wissenschaften Hamburg

IRQ Interrupt Request

ISO International Organization for Standardization

ISR Interrupt Service Routine

LIDAR Light Detection and Ranging

LIN Local Interconnect Network

MAC Media Access Control

MOST Media Oriented Systems Transport

NRZ Non-Return-to-Zero

PCF Protocol Control Frames

RC Rate Constraint

RTE Realtime Ethernet

SC Synchronization Client

SM Synchronization Master

TDMA Time Division Multiple Access

TT Time Triggered

VLID Virtual Link Identier

VW Volkswagen

V2R Vehicle-to-Roadside

V2V Vehicle-to-Vehicle

Literaturverzeichnis

[ABI research 2014] ABI RESEARCH: *Ethernet In-vehicle Networking to Feature in 40Shipping Globally by 2020.* https://www.abiresearch.com/press/ethernet-in-vehicle-networking-to-feature-in-40-of/. 2014. – [Online; Zugriff 20-August-2015]

[CAN Newsletter Online 2014] CAN NEWSLETTER ONLINE: *Growing car sales increase CAN business.* http://www.can-newsletter.org/engineering/engineering-miscellaneous/nr_growing-car-sales-increase-the-can-business_140107/. 2014. – [Online; Zugriff 04-September-2015]

[Christiane Brünglinghaus] CHRISTIANE BRÜNGLINGHAUS: *Am Ethernet im Auto fÃ¼hrt kein Weg vorbei.* http://www.springerprofessional.de/am-ethernet-im-auto-fuehrt-kein-weg-vorbei/4979586.html. – [Online; Zugriff 19-August-2015]

[CoRE Arbeitsgruppe] CORE ARBEITSGRUPPE: *Hochschule fuer Angewandte Wissenschaften Hamburg.* http://core.informatik.haw-hamburg.de/. – [Online; Zugriff 19-August-2015]

[Depke 2015] DEPKE, Jan: *Inkrementelle Konsolidierung automobiler Bussysteme auf Basis eines Realtime Ethernet Backbones*, Hochschule fuer Angewandte Wissenschaften Hamburg, Masterarbeit, 2015

[IEEE 802.1Q 2011] IEEE 802.1Q: *IEEE Std 802.1Q - IEEE Standard for Ethernet.* 2011

[IEEE 802.3 2012] IEEE 802.3: *IEEE Std 802.3 - IEEE Standard for Ethernet.* 2012

[ISO 11898 1 2003] ISO 11898 1: *Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling.* 2003

[ISO 11898 2 2003] ISO 11898 2: *Road vehicles – Controller area network (CAN) – Part 2: High-speed medium access unit.* 2003

[ISO 11898 3 2006] ISO 11898 3: *Road vehicles – Controller area network (CAN) – Part 3: Low-speed, fault-tolerant, medium-dependent interface.* 2006

[Marscholik und Subke 2007] Marscholik, Christoph ; Subke, Peter: *Datenkommunikation im Automobil: Grundlagen, Bussysteme, Protokolle und Anwendungen.* Heidelberg : Hüthig, 2007 (Hüthig Praxis). – ISBN 3-7785-2969-2

[Navet u. a. 2005] Navet, Nicolas ; Song, Yeqiong ; Simonot-Lion, Françoise ; Wilwert, Cédric: Trends in Automotive Communication Systems. In: *Proceedings of the IEEE* 93 (2005), Juni, Nr. 6, S. 1204–1223. – ISSN 0018-9219

[Vector Informatik GmbH a] Vector Informatik GmbH: *CANalyzer Homepage.* http://vector.com/vi_canalyzer_de.html. – [Online; Zugriff 19-April-2015]

[Vector Informatik GmbH b] Vector Informatik GmbH: *CANoe Homepage.* http://vector.com/vi_canoe_de.html. – [Online; Zugriff 19-April-2015]

[Vector Informatik GmbH c] Vector Informatik GmbH: *VN5610 Handbuch.* http://vector.com/portal/medien/cmc/manuals/VN56xx_Manual_DE.pdf. – [Online; Zugriff 19-April-2015]

[Vector Informatik GmbH 2014] Vector Informatik GmbH: *Kompendium ausgewÃ¤hlter Fachartikel zur Elektronik-Entwicklung in verteilten Systemen.* http://vector.com/portal/medien/cmc/marketing_items/web/91110.pdf. 2014. – [Online; Zugriff 15-April-2016]

[Wireshark Foundation] Wireshark Foundation: *Wireshark Homepage.* https://www.wireshark.org/. – [Online; Zugriff 19-April-2015]

[Zhu 2010] Zhu, Yu: *CAN and FPGA Communication Engineering: Implementation of a CAN Bus based Measurement System on an FPGA Development Kit.* Hamburg : Diplomica Verlag GmbH, 2010. – ISBN 978-3-8366-9925-9

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 25. April 2016 Patrick Kuncke