Masterarbeit

Integration von Car-to-X Kommunikation in die E/E-Architektur von Fahrzeugen

Christian Payerl, BSc

Institut für Elektrische Messtechnik und Messsignalverarbeitung
Technische Universitat Graz
Vorstand: Univ.-Prof. Dipl.-Ing. Dr. techn. Georg Brasseur

Begutachter: Univ.-Doz. Dipl.-Ing. Dr. techn. Daniel Watzenig

Graz, im Juli 2013

Kurzfassung

Car-to-X Kommunikation dient dem drahtlosen Informationsaustausch zwischen Fahrzeugen und Infrastruktur. Über die ausgetauschten Informationen kann der Fahrer frühzeitig auf Situationen reagieren, die sich außerhalb seines Sichtbereichs beziehungsweise außerhalb des Fahrzeugsensorbereichs befinden. Somit eröffnet die Car-to-X Kommunikation neue Möglichkeiten zur Verbesserung der Verkehrseffizienz und Verkehrssicherheit. Zahlreiche Publikationen befassen sich mit der Thematik aus Sicht der Funkübertragung, des Netzwerkaufbaus oder den notwendigen Sicherheitsmaßnahmen. Praktische Realisierungen beschränken sich bislang auf Versuchsaufbauten und Feldtests und sind nicht für den Serieneinsatz ausgelegt.

In dieser Masterarbeit werden alle derzeit am Markt verfügbaren prototypischen Car-to-X Plattformen bewertet. Die Plattform mit der größten Seriennähe wird erworben und die notwendigen Hardware- und Softwareschnittstellen zur Fahrzeugintegration evaluiert. Für die Untersuchung wird der Car-to-X Use-Case des elektronischen Bremslichts bei Notbremsung (EEBL) auf der Plattform implementiert, um die Schnittstellen in praxisbezogener Funktion zu testen. Die Evaluierung der Plattform erfolgt sowohl im Labor, als auch in einem Feldtest auf der Straße. Im Labor wird der Use-Case des EEBL umgesetzt, während beim Straßenfeldtest Reichweitenmessungen zwischen zwei Plattformen in Fahrzeugen durchgeführt werden.

Die Arbeit zeigt, dass das Linux-Betriebssystem und die dazugehörigen Softwareschnittstellen essenzielle Komponenten einer Car-to-X Plattform sind. Diese Tatsache impliziert, dass bisherige Linux-Softwarekomponenten, die ihren Ursprung im Bereich der Unterhaltungselektronik haben, Einzug in den Automotive-Sektor halten. Die früher strikt getrennten Welten der Unterhaltungselektronik und des Automotive-Bereichs verschmelzen dadurch zusehends.

Abstract

Car-to-x communication provides wireless exchange of information between vehicles and infrastructure. Due to the exchanged information, a driver is able to react on situations that are outside of his visual range or outside the vehicle sensor range. Hence, car-to-x communication enables new possibilities to increase traffic efficiency and road safety. Numerous publications deal with the issue from the perspective of wireless communication, network structure or safety countermeasures. So far, practical realizations are limited by experimental setups or field trials. These realizations are not designed for series-production.

This master thesis evaluates all car-to-x prototype platforms that are currently available. The platform, which is most suitable for future series-production is chosen for an implementation. For a car-to-x real world scenario the use-case of an electronic emergency braking light (EEBL) is implemented. On the basis of this use-case implementation, required hardware- and software-interfaces are evaluated for authentic vehicle integration. The evaluation takes place in the laboratory and on a road test setup. The use-case implementation of the EEBL is executed in the laboratory. Analysis of wireless communication coverage between two cars is carried out on the road.

This thesis shows that the Linux operating system and the corresponding interfaces are essential for car-to-x communication. Based on that fact, components that are originally developed for consumer-electronics are now applied to automotive systems. Thus, the previously strictly separated areas of consumer electronics and the automotive sector are merging progressively.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 01.Juli 2013 .....

(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, July 1st 2013 .....

(signature)

Danksagung

Ohne die Mithilfe und Unterstützung einiger wichtiger Personen, wäre es nicht möglich gewesen, diese Masterarbeit zu verfassen. Deshalb möchte ich mich hier bei diesen Personen besonders bedanken.

Als aller erstes möchte ich mich bei meiner Familie bedanken, insbesondere bei meinen Eltern, die mich von Kindheit an in all meinen Bestrebungen unterstützt haben. Sie haben mich nicht nur in finanzieller Weise unterstützt, sondern sind mir immer in allen Lebenslagen beigestanden und haben mir Rückhalt gegeben.

Weiters möchte ich auch meinen Freunden für ihr Verständnis und ihre Unterstützung während der letzten Jahre danken.

Diese Masterarbeit ist in Zusammenarbeit des Virtuellen Fahrzeugs mit Magna Steyr Fahrzeugtechnik entstanden. Dabei möchte ich mich besonders bei meinen Betreuern am Virtuellen Fahrzeug, Herrn Dipl.-Ing. Joachim Hillebrand sowie Herrn Univ.-Doz. Dr. techn. Dipl.-Ing. Daniel Watzenig für die wertvolle Unterstützung während dieser Masterarbeit bedanken.

Seitens Magna Steyr Fahrzeugtechnik gebührt auch Herrn Dipl.-Ing. Kurt Tschabuschnig ein weiterer Dank für die gute Zusammenarbeit zwischen Forschung und Industrie.

Graz, im Juli 2013 Christian Payerl

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Car-to-X Kommunikation eröffnet neue Möglichkeiten zur Verbesserung der Verkehrseffizienz und Verkehrssicherheit. Die Basis von Car-to-X ist die drahtlose Kommunikation. Fahrzeuge tauschen untereinander (Car-to-Car) oder mit vorhandener Infrastruktur (Car-to-Infrastructure) Informationen über eine Drahtlosverbindung aus. Abbildung 1.1 zeigt die Car-to-X Kommunikation zwischen Fahrzeugen und Ampeln. Die übertragenen

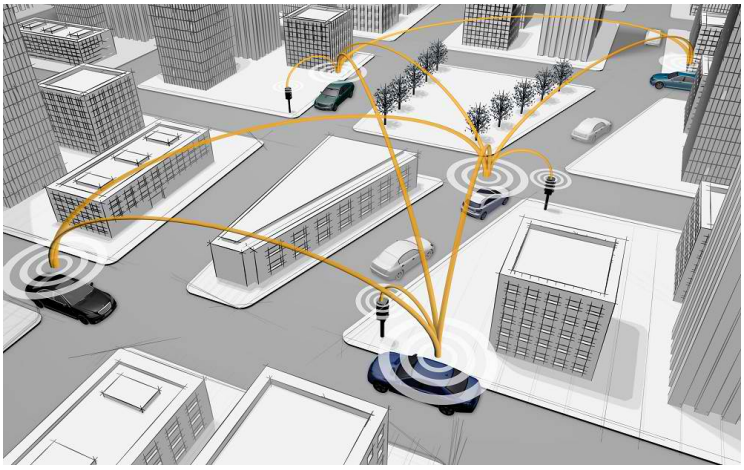

Abbildung 1.1: Car-to-X Kommunikation (Quelle: Daimler AG)

Informationen ermöglichen, über die Reichweite der Fahrzeugsensorik hinaus, Gefahrensituationen vorherzusehen und in weiterer Folge den Fahrer frühzeitig zu warnen. Über die Fahrzeugelektronik besteht die Möglichkeit unmittelbar in das Fahrverhalten einzugreifen, um Unfälle zu vermeiden. Die ausgetauschten Informationen können auch zur Beurteilung der Verkehrsumgebung herangezogen werden. In Zeiten des ständig zunehmenden Verkehrs und der damit verbundenen Steigerung des CO_2-Ausstoßes kann über eine frühzeitige Beurteilung der Verkehrsumgebung eine Verbesserung der Verkehrseffizienz durch Stauvermeidung und Verkehrsflussoptimierung erzielt werden.

1.1 Beschreibung

Drahtlose Kommunikation im Fahrzeug ist durch Nutzung von Komfort- und Infotainment-Anwendungen basierend auf Bluetooth, Global Positioning System (GPS) oder Universal Telecommunication System (UMTS) bereits Stand der Technik. Bislang wird Drahtlos-Kommunikation in Fahrzeugen jedoch ausschließlich für nicht-sicherheitskritische Anwendungen verwendet. Mit Einführung der Car-to-X Kommunikation können fehlerhafte oder manipulierte Daten der drahtlosen Kommunikation erstmals Einfluss auf die Sicherheit des Fahrers nehmen. Car-to-X Systeme müssen daher deutlich komplexeren Anforderungen genügen, als die bisher eingesetzten drahtlosen Systeme im Fahrzeug. Aus diesem Grund arbeitet in Europa das European Telecommunications Standards Institute (ETSI) an der Festlegung von Standards für eine europaweite Einführung von Car-to-X. Die Integration von Car-to-X in die Fahrzeugarchitektur stellt daher auch Automobilhersteller vor neue Herausforderungen. Diese Arbeit beleuchtet den aktuellen technischen Stand von Car-to-X Systemen und gibt einen Aufschluss darüber, wie Car-to-X Kommunikation in die bestehende E/E-Architektur von Fahrzeugen integriert wird und welche Auswirkungen die Integration auf das Gesamtsystem hat. Zur Integration wird der Car-to-X Use-Case des elektronischen Bremslichts bei Notbremsung (EEBL) umgesetzt. Die Evaluierung der benötigten Hardware- und Softwareschnittstellen erfolgt im Zuge eines Laboraufbaus, wobei zwei Car-to-X Plattformen miteinander über den spezifizierten Drahtlos-Standard kommunizieren. Um die Reichweite einer seriennahen Car-to-X Plattform einschätzen zu können, wird ein Straßenfeldtest durchgeführt. Dafür werden zwei Fahrzeuge mit der Car-to-X Plattform ausgestattet und die Reichweite anhand der Paket Error Rate gemessen.

1.2 Gliederung

Diese Masterarbeit ist folgender Maßen gegliedert: In **Kapitel 2** werden die Grundlagen basierend auf Literaturrecherchen zur Car-to-X Thematik behandelt. Dabei wird zu Beginn die Systemarchitektur anhand des Feldversuchs sim^{TD} näher erklärt, sowie auf die aktuellen Standards und auf die Sicherheitsthematik eingegangen. **Kapitel 3** beschäftigt sich mit der Wahl einer geeigneten Car-to-X Plattform und bringt die Hardware- und Softwarearchitektur der gewählten Plattform näher. Die Integration der gewählten Plattform in die E/E-Architektur eines Fahrzeugs, wird in **Kapitel 4** behandelt. Zusätzlich wird im Detail auf die verwendeten Hardware- und Softwareschnittstellen eingegangen. Das **Kapitel 5** beschreibt den durchgeführten Feldtest zur Ermittlung der Reichweite der Drahtloskommunikation und legt die Ergebnisse des Feldtests dar. **Kapitel 6** beinhaltet eine Zusammenfassung und die Gesamtergebnisse dieser Arbeit.

Kapitel 2

Grundlagen

Dieses Kapitel beschreibt technische Grundlagen der Car-to-X Kommunikation auf Basis von Literaturrecherchen. In Abschnitt 2.1 wird die grundlegende Systemarchitektur eines Car-to-X Netzwerks (Vehicular Ad-Hoc Network (VANET)s) anhand des Feldversuchs sim^{TD} erläutert. Abschnitt 2.2 beschreibt bereits spezifizierte Standards der Car-to-X Kommunikation und zeigt regionale Unterschiede bezüglich der Standardisierungen in den USA und Europa auf. Abschnitt 2.3 behandelt das Thema Sicherheit und Datenschutz innerhalb des VANETs.

2.1 Feldversuch sim^{TD}

Seit einigen Jahren gibt es internationale Bestrebungen nach praktischen Umsetzungen im Bereich Car-to-X in Form von Feldversuchen. In Europa werden aktuell durch das europäische Projekt DRIVE C2X [1] Feldversuche in Deutschland, Finnland, Frankreich, Italien, den Niederlanden und Schweden durchgeführt. Im DRIVE C2X Projektzeitraum von 2011 bis 2013 werden Car-to-X Anwendungen über Wireless Local Area Network (WLAN) und UMTS getestet [2]. Abbildung 2.2 zeigt die Übersicht der einzelnen Feldversuche des DRIVE C2X Projekts. Zu den weltweit größten Feldversuchen zählt „Sichere Intelligente Mobilität Testfeld Deutschland“ (sim^{TD}), der sich im Vergleich zu den bis dato durchgeführten Feldversuchen dadurch unterscheidet, dass die Wireless-Kommunikation nicht unter einigen wenigen Versuchsfahrzeugen, sondern in einem großen Rahmen stattfindet. Durch die größere Anzahl an Fahrzeugen liefert der Feldversuch repräsentative Ergebnisse im Hinblick auf den späteren realen Einsatz. Aus diesem Grund wird im Folgenden auf die Rahmenbedingungen des Feldversuchs eingegangen.

Abbildung 2.1: Feldversuche im Rahmen des DRIVE C2X Projekts (Quelle [2])

2.1.1 Rahmenbedingungen

Im Projekt „Sichere Intelligente Mobilität Testfeld Deutschland“ (sim^{TD}) arbeiten 17 Partner aus Industrie und Forschungsinstituten zusammen, um die Funktionalität von Car-to-X unter realen Umgebungsbedingungen zu testen. Im Zuge des Projekts werden bereits entwickelte Konzepte aus vorigen Projekten wie PRE-DRIVE C2X [3] evaluiert, sowie Ansätze zur Verbesserung erarbeitet. Die Leitung des Projekts liegt beim Automobilhersteller Daimler [4]. Der Feldversuch mit 120 Fahrzeugen, welche mit On Board Unit (OBU)s ausgestattet waren, wurde von Juli bis Dezember 2012 durchgeführt. Die 170 km lange Teststrecke war mit 100 Road Side Unit (RSU)s ausgestattet und umfasste 96 km Autobahn, 53 km Landstraße und 24 km innerstädtische Straßen. Der Autobahnabschnitt umfasste auch das „Frankfurter Kreuz“, welches mit 310 000 Fahrzeugen täglich eine der meistbefahrenen Autobahnkreuzungen Europas darstellt [1]. Auf der Teststrecke werden Car-to-X Anwendungsfälle für Verkehrsfluss, Warnungen, Fahrerassistenz und Zusatzdienste umgesetzt. Im Folgenden wird auf Informationen eingegangen, die zum Zeitpunkt der Masterarbeit verfügbar waren. Die vollständige Evaluierung seitens sim^{TD} wurde noch nicht publiziert.

2.1.2 Systemarchitektur

In Abbildung 2.2 ist die Systemarchitektur in sim^{TD} dargestellt. Die Systemarchitektur kann in die Komponenten OBU, RSU und ICS (ITS Central Station) eingeteilt werden. In den folgenden Unterpunkten werden die einzelnen Komponenten erläutert, um ein Verständnis der Systemarchitektur zu erlangen.

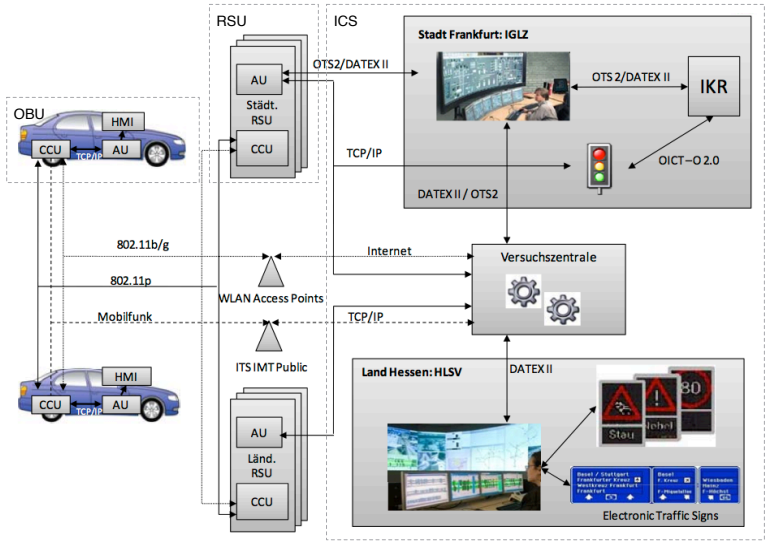

Abbildung 2.2: Gesamtsystemarchitektur in sim^{TD} (Quelle [5])

OBU (On Board Unit)

Die On Board Unit ist die Schnittstelle zwischen Fahrer, Fahrzeug und Umgebung. Auf der OBU findet nicht nur Wireless- und Buskommunikation statt, die Hardwareplattform stellt auch die Softwarefunktionalität der Car-to-X Anwendungen zur Verfügung. Im Projekt sim^{TD} ist die OBU nicht als eine integrierte Plattform implementiert. Die Funktionalität der OBU wird auf zwei separaten Plattformen, einem Router und einem Host untergebracht. Abbildung 2.3 zeigt einen Überblick über Zuordnung der OBU-Funktionalitäten zu Router und Host. Das Herzstück des Routers ist ein 400 MHz Mikrocontroller mit einem Linux Betriebssystem. Der Router ermöglicht die Wireless-Kommunikation über unterschiedliche Standards wie IEEE 802.11p, IEEE 802.11b/g und UMTS. Über IEEE 802.11p erfolgt die Kommunikation der Fahrzeuge untereinander wie auch die Kommunikation zwischen Fahrzeug und RSU. Die Kommunikation über IEEE 802.11b/g, welche für herkömmliches WLAN im Bereich der Unterhaltungselektronik spezifiziert ist, dient zum Performancevergleich zu IEEE 802.11p. Für den Datentransfer von sicherheitsrelevanten Nachrichten über große Distanzen wird UMTS verwendet. Die GPS Lokalisierung des Routers unterstützt Differential GPS und Dead-Reckoning [6]. Damit der Router die Controller Area Network (CAN)-Daten der unterschiedlichen Fahrzeuge in sim^{TD} verarbeiten kann, wird ein intelligenter CAN-Treiber mit einer standardisierten Programmierschnittstelle verwendet. Dies ermöglicht die Nutzung standardisierter Application Programming Interface (API)s [7]. Der Router kommuniziert mit dem Host über Ethernet, wobei Softwarefunktionalität wie Navigation und Car-to-X Anwendungen

in Form von Java-Anwendungen auf dem Host implementiert ist. Als Host wird ein Windows-PC verwendet.

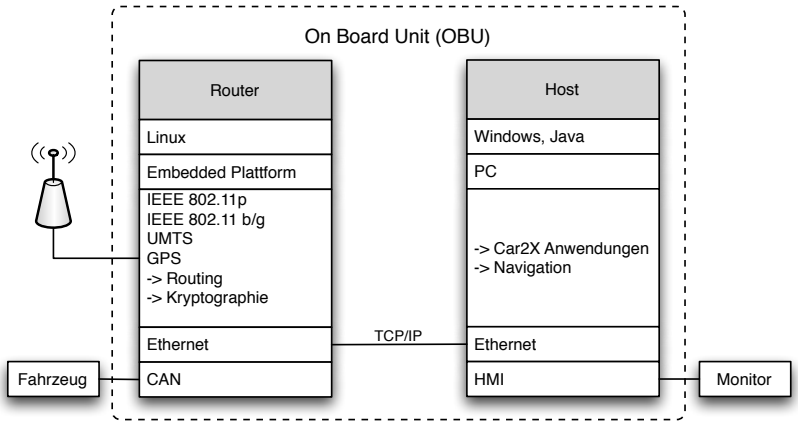

Abbildung 2.3: Router und Host bilden die On Board Unit in sim^{TD}

RSU (Road Side Unit)

Die Road Side Unit ist das Bindeglied zwischen Fahrzeug und Verkehrszentrale. Im Projekt sim^{TD} sind 100 RSUs im Einsatz, 80 davon sind mit dem Hessischen Landesamt für Straßen- und Verkehrswesen (HLSV) verbunden und 20 mit der Integrierten Gesamtverkehrsleitzentrale der Stadt Frankfurt am Main (IGLZ) [8]. Netzwerk-, Routing- und Transportschicht basieren auf der OBU-Systemarchitektur. Um Nachrichten zwischen Fahrzeugen und Verkehrszentralen zu übermitteln, bedarf es einer Erweiterung der Architektur mit Algorithmen für Verwaltung und Verteilung der Nachrichten. Die Kommunikation erfolgt in Richtung Fahrzeuge über IEEE 802.11p, in Richtung der Verkehrszentrale über TCP/IP. Das Transportmedium auf der die TCP/IP-Kommunikation erfolgt, ist bei stationären RSUs Glasfaser beziehungsweise xDSL, bei mobilen RSUs wird zur Weiterleitung UMTS verwendet [9].

ICS (ITS Central Station)

Die sim^{TD}-Systemarchitektur besteht im Bereich der Verkehrszentralen aus der Integrierten Gesamtverkehrsleitzentrale der Stadt Frankfurt am Main (IGLZ), dem Hessischen Landesamt für Straßen- und Verkehrswesen (HLSV) und der sim^{TD}-Versuchszentrale. Die Aufgaben der einzelnen Zentralen setzen sich folgendermaßen zusammen: Das IGLZ überwacht den innerstädtischen Verkehr in Frankfurt am Main und ist in der Lage durch Wechselverkehrszeichen und Ampeln den urbanen Verkehrsfluss zu steuern. Analog dazu ist das HLSV für Autobahnnetz und Freilandstraßen in Hessen verantwortlich. Die Versuchszentrale überwacht das gesamte System und zeichnet Daten für spätere Analysen auf [8].

Zur Beurteilung der Gesamtverkehrssituation ermittelt die jeweilige Verkehrszentrale einmal pro Minute den aktuellen Verkehrszustand im jeweiligen Zuständigkeitsbereich (IGLZ städtisch, HLSV ländlich). Die Ergebnisse werden in einer Datenbank gespeichert und den RSUs übermittelt. Eine Beurteilung der Gesamtverkehrssituation wird durch Zusammenführen der Daten zwischen HLSV und IGLZ ermöglicht, indem IGLZ und HLSV gegenseitig Daten austauschen. Bei Bedarf kann die OBU diese Daten anfordern und dem Fahrer beispielsweise einen Überblick über die aktuelle Verkehrssituation auf seiner Route liefern und gegebenenfalls Alternativrouten vorschlagen [8].

2.1.3 Datensicherheit

In die Systemarchitektur von sim^{TD} wurde zum Zweck der Datensicherheit eine Public Key Infrastruktur integriert. In der Versuchszentrale stellt die Zertifizierungsstelle (CA) als Teil dieser Public Key Infrastruktur einen Public Key zur Verfügung. Über diesen Public Key wird von der OBU eines Fahrzeugs eine Signatur errechnet. Die OBU signiert die eigenen Nachrichten, wodurch in den Verkehrszentralen die Authentizität eingehender Nachrichten überprüft wird [8]. In sim^{TD} ist die CCU als Teil der OBU jedoch nicht für Verschlüsselungsoperationen optimiert. Aus diesem Grund wurde eine 512-Bit RSA Verschlüsselung verwendet anstatt des ECDSA-256, welcher nach IEEE 1609.2 [10] vorgeschlagen wird. Im Vergleich zum ECDSA-256 kann die 512-Bit RSA Signaturverifikation 27 mal schneller durchgeführt werden. Dennoch besteht bei großem Verkehrsaufkommen die Möglichkeit, dass die Signaturen nicht schnell genug verifiziert werden können. In diesem Fall wird auf einen Fallback-Algorithmus zurückgegriffen [11].

2.1.4 Zusammenfassung

Der sim^{TD}-Feldversuch zählt zu den bis dato größten durchgeführten Feldtests. Bei der Umsetzung des Feldversuchs wurden möglichst reale Rahmenbedingungen geschaffen. Auch die sim^{TD}-Systemarchitektur könnte in realen Szenarien Anwendung finden. Anstatt der verwendeten OBU, bestehend aus Windows PC und Router, werden jedoch in Zukunft Embedded-Plattformen als OBUs im Einsatz sein. Im Hinblick auf Datensicherheit wurde aus Performance-Gründen ein schwächerer Algorithmus zur Signierung implementiert. Abschnitt 2.3 behandelt dazu das Thema Sicherheitsarchitektur und Kryptographie im Detail und gibt einen Überblick der definierten Standards nach ETSI und IEEE. Zum Zeitpunkt des Verfassens dieser Arbeit liegen noch keine offiziellen Ergebnisse zum sim^{TD} Feldversuch vor.

2.2 Standardisierungen und regionale Unterschiede

Die Entwicklung im Bereich Car-to-X wird vor allem in den USA und Europa forciert. Dabei verfolgen die einzelnen Länder unterschiedliche Ziele, was zu regionalen Unterschieden in den Standardisierungen führt und einer internationalen Harmonisierung entgegenwirkt [12]. Wie die Standards in den einzelnen Ländern definiert sind und in welchen Punkten sie sich unterscheiden, ist in folgenden Abschnitten 2.2.1 (USA), und 2.2.2 (Europa) angeführt.

2.2.1 USA

In den USA liegt das Hauptziel von Car-to-X in Verkehrssicherheit und Verkehrseffizienz, welche auf infrastrukturbasierter Kommunikation aufbauen [13]. Die Standardisierung wird vom Institute of Electrical and Electronics Engineers (IEEE) durchgeführt, wobei als Sammelbezeichnung für den US spezifischen Protokollstack das Akronym Wireless Access for the Vehicular Environment (WAVE) verwendet wird. Abbildung 2.4 zeigt den Aufbau des WAVE Protokollstacks. Die Farbgebung dient der Zuordnung zu den Schichten des ISO/OSI-Modells. Die Bitübertragungs- und Sicherungsschicht des ISO/OSI Modells werden im WAVE Protokollstack durch den IEEE 802.11p Standard [14] repräsentiert. In diesem Abschnitt wird überblicksartig auf die einzelnen WAVE-Standards eingegangen. Der IEEE 802.11p Standard stellt die Basis der Car-to-X Kommunikation dar und ist ebenso die Grundlage des europäschen TC-ITS G5 Standards (Abschnitt 2.2.2), daher wird im Folgenden auf diesen detaillierter eingegangen.

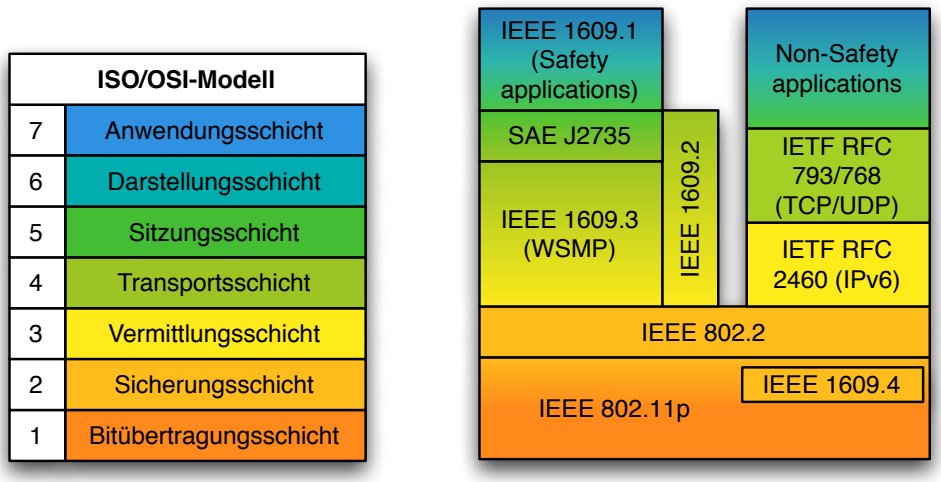

Abbildung 2.4: Aufbau des WAVE Protokollstacks im Vergleich zum ISO/OSI-Modell

2.2.1.1 IEEE 802.11p

Die Standardisierung für WLAN erfolgte zu Beginn mit dem 802.11 Standard, welcher anfangs die Erweiterungen a und b vorsah [15]. Mittlerweile basieren herkömmliche WLAN Geräte im Bereich der Unterhaltungselektronik nahezu ausschließlich auf dem 802.11 Standard mit den Erweiterungen a, b, g oder n. Um Kompatibilität zwischen den Geräten unterschiedlicher Hersteller zu gewährleisten, werden diese über die Wireless Ethernet Compatibiltiy Allaince (Wi-Fi Alliance) [16] zertifiziert, wodurch sich der Begriff Wi-Fi als Synonym für WLAN etablierte. Der IEEE 802.11a Standard erwies sich als geeignete Basis für den IEEE 802.11p Standard, sieht jedoch den Einsatz in stationärer Umgebung vor. Aus diesem Grund wurden die Spezifikationen der ursprünglichen IEEE 802.11a für den mobilen Einsatz optimiert [17]. Der Fokus von IEEE 802.11p liegt in der Inter-Fahrzeug Kommunikation und spezifiziert wie in Abbildung 2.4 ersichtlich die Bitübertragungsschicht und die Sicherungsschicht des ISO/OSI Schichtenmodells [18]. Tabelle 2.1 zeigt die

Parameter	**IEEE 802.11a**	**IEEE 802.11p**
Frequenzband	5,180 GHz - 5,825 GHz	5,850 GHz - 5,925 GHz
Bandbreiten	20 MHz	5 MHz, 10 MHz oder 20 MHz
Datenrate	6, 9, 12, 18, 24, 36, 48, 54 Mbps	3, 4.5, 6, 9, 12, 18, 24, 27 Mbps
Modulation	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM
OFDM Symboldauer	4,0 μs	8,0 μs
Sicherheitsabstand	0,8 μs	1,6 μs
Ratifizierung	1999	2010

Tabelle 2.1: Gegenüberstellung IEEE 802.11a und IEEE 802.11p

Gegenüberstellung von IEEE 802.11a zu IEEE 802.11p. Durch doppelten Sicherheitsabstand und doppelter Symboldauer von IEEE 802.11p wird eine geringere Intersymbolinterferenz (ISI) [19] sowie eine höhere Robustheit gegen Mehrwegeausbreitung erzielt. Wie Tabelle 2.1 zeigt, ist die maximale Datenrate auf 27 Mbps begrenzt. In den USA liegt das 5,9 GHz-Band in einem Bereich von 5,85 bis 5,925 GHz.

Aufbauend auf dem IEEE 802.11p Standard werden die darüberliegenden Schichten über die IEEE 1609.x Protokollfamilie definiert. Wie Abbildung 2.4 zeigt, unterteilen sich die Schichten ab der Logical Link Control (IEEE 802.2 [20]) in einen linken und rechten Zweig. Der Linke repräsentiert die Schichten, die für sicherheitsrelevante Kommunikation (Basic Safety Messages) wesentlich sind. Der rechte Zweig jene Schichten, die für die Kommunikation von nicht sicherheitsrelevanten Nachrichten (Infotainment) verantwortlich sind.

2.2.1.2 IEEE 1609.x

IEEE 1609.1 - Resource Manager [21]
: Der Resource Manager entspricht dem WAVE Betriebssystem. Hier werden unter anderem bereitgestellte Dienste und die Formate von Nachrichten innerhalb der WAVE Architektur spezifiziert.

IEEE 1609.2 - Security Services [10]
: Dieser Standard spezifiziert alle sicherheitsrelevanten Maßnahmen der WAVE Kommunikation. In Abschnitt 2.3 wird auf die Sicherheitsanforderungen und die grundlegende Systemarchitektur eingegangen.

IEEE 1609.3 - Networking Serivces [22]
: Networking Services definiert Netzwerk- und Transportschicht wie auch das WAVE Short Message Protokoll (WSMP), welches den Transport sicherheitsrelevanter Nachrichten, der Basic Safety Messages (BSM) mit geringem Overhead vorsieht.

IEEE 1609.4 - Multi-Channel Operation [23]
: Der 5,9 GHz Bereich ist auf 7 Kanäle zu je 10 MHz aufgeteilt. Bei diesen Kanälen sind ein Steuerkanal und sechs Servicekanäle vorgesehen.

2.2.1.3 SAE J2735

SAE J2745 [24] definiert die Basic Safety Messages (BSM). Die BSMs sind Nachrichten eines Fahrzeugs und beinhalten sicherheitsrelevante Informationen für andere Verkehrsteilnehmer. Die übertragenen Informationen entsprechen inhaltlich den Informationen welche nach dem europäischen ETSI Standard in den Cooperative Awareness Message (CAM) beziehungsweise in den Decentralized Environmental Notification Message (DENM) enthalten sind. Eine ausführliche Beschreibung dieser Nachrichten findet sich in Abschnitt 2.2.2.3.

2.2.2 Europa

Im Gegensatz zu den USA liegt in Europa der Fokus auf infrastrukturloser Kommunikation [25]. Das Car2Car Communication Consortium [26] und das ETSI [27] arbeiten an

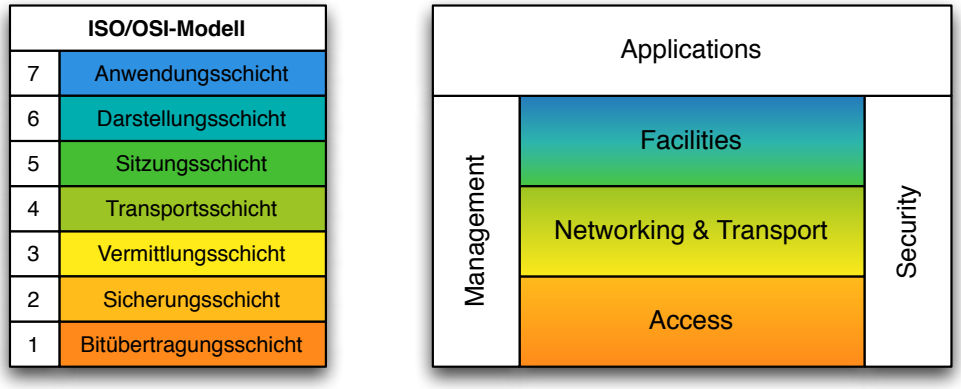

Abbildung 2.5: Aufbau des C-ITS Protokollstacks im Vergleich zum ISO/OSI-Modell

der Umsetzung des Cooperative-Intelligent Transportation System (ITS) (C-ITS) - der europäischen Spezifikation für Car-to-X. Abbildung 2.5 zeigt den Aufbau des C-ITS Protokollstacks im Vergleich zum ISO/OSI-Modell. Die Farbgebung dient der Zuordnung zu den Schichten des ISO/OSI-Modells. Wie zu sehen ist, werden die sieben Schichten des ISO/OSI Modells in den drei Schichten Access, Networking and Transport und Facilities des C-ITS Protokollstacks abgebildet. Der Fokus liegt in Europa zwar auf infrastrukturloser Kommunikation, die Standards sehen jedoch auch infrastrukturbasierte Kommunikation vor. In diesem Abschnitt werden die einzelnen Schichten des C-ITS Protokollstacks erläutert und es wird auf die wichtigsten Unterschiede zum amerikanischen WAVE Standard eingegangen. Eine detaillierter Übersicht zu den einzelnen ETSI-Standards findet sich in Anhang B.

2.2.2.1 Access

In dieser Schicht sind jene Standards definiert, welche die Bitübertragungs- und Sicherungsschicht des ISO/OSI-Modells betreffen. Für die drahtlose Kommunikation existiert dem amerikanischen IEEE 802.11p Standard entsprechend in Europa ITS-G5. Dabei handelt es sich um den IEEE 802.11p Standard, welcher durch ETSI an die europäischen Spezifikationen angepasst wurde. Tabelle 2.2 zeigt die Zuteilung der Kanäle nach ETSI ES 202 663 [28]. In Europa beginnt das 5,9 GHz-Band bei 5,875 GHz und endet bei 5,925 GHz, wobei sich dieser Bereich auf 6 Kanäle zu je 10 MHz aufteilt. Der Steuerkanal G5CC (Tabelle 2.2) dient zur Übermittlung von Nachrichten, die die Straßensicherheit

oder den Verkehrsfluss betreffen. Im Gegensatz zum Standard IEEE 802.11p, wo über einen Transceiver zwischen den Kanälen zeitlich umgeschaltet wird, sieht die ITS-G5 Spezifikation zwei Transceiver vor [29]. Ein Transceiver ist für den Control Channel, der zweite für die Service Channels vorgesehen. Die Servicekanäle G5SC1 und G5SC2 werden von Straßensicherheits- oder Verkehrsflussanwendungen verwendet. Die verbleibenden Servicekanäle G5SC3, G5SC34 und G5SC3 sind für Benutzeranwendungen reserviert. Wie bei den Sendeleistungen (Spalte TX power limit) zu sehen ist, wird grundsätzlich mit einer Leistung von 23 dBm EIRP (200 mW) gesendet. Für die sicherheitsrelevanten Anwendungen von Einsatzfahrzeugen ist für die Kanäle G5CC und G5SC1 eine maximale Sendeleistung von 33 dBm EIRP (2 W) vorgesehen. Der Kanal G5SC5 kann für Dynamic Frequency Selection (DFS) [30] verwendet werden und besitzt je nach DFS Mode eine Sendeleistung von 33 dBm EIRP beziehungsweise 23 dBm EIRP.

Channel type	**Centre frequency**	**IEEE ch. No.**	**Channel spacing**	**Default data rate**	**TX power limit**
G5CC	5900 MHz	180	10 MHz	6 Mbit/s	33 dBM EIRP
G5SC2	5890 MHz	178	10 MHz	12 Mbit/s	23 dBM EIRP
G5SC1	5880 MHz	176	10 MHz	6 Mbit/s	33 dBM EIRP
G5SC3	5870 MHz	174	10 MHz	6 Mbit/s	23 dBM EIRP
G5SC4	5860 MHz	172	10 MHz	6 Mbit/s	0 dBM EIRP
G5SC5	5470 MHz to 5725 MHz		several		33 dBM EIRP (DFS master) 23 dBm EIRP (DFS slave)

Tabelle 2.2: Zuteilung der Kanäle nach ETSI ES 202 663 [28]

2.2.2.2 Networking and Transport

Die Schicht Networking and Transport definiert jene Standards, welche die Vermittlungs-, Transport- und Sitzungsschicht des ISO/OSI-Modells betreffen. Abbildung 2.6 zeigt die

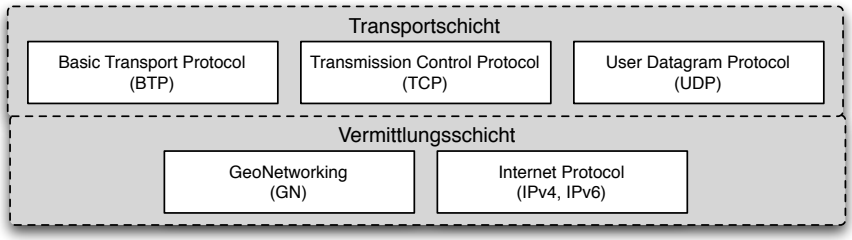

Abbildung 2.6: Protokolle der Vermittlungs- und Transportschicht

Protokolle der Vermittlungs- und Transportschicht. Wie zu sehen ist, sieht die Schicht Standard-Netzwerkprotokolle wie Transmission Control Protocol (TCP), User Datagram Protocol (UDP) und Internet Protocol (IP) vor. Die Protokolle GeoNetworking (GN) und Basic Transport Protocol (BTP) sind speziell für Anwendung in VANETs vorgesehen.

GeoNetworking

Das ETSI GeoNetworking Protokoll ist ein Netzwerkprotokoll, das die Möglichkeit einer geografischen Adressierung und Weiterleitung von Nachrichten bietet. Somit besteht die Möglichkeit, Nachrichten innerhalb eines geografisch beschränkten Bereichs zu senden. Die Netzwerkarchitektur ist dabei nicht von Relevanz, das heißt der Nachrichtenaustausch kann innerhalb eines infrastrukturlosen Netzwerks, eines infrastrukturbasierten Netzwerks oder in einem gemischten Netzwerk erfolgen. [31]

Basic Transport Protocol

Das Basic Transport Protocol (BTP) ermöglicht innerhalb des VANETs eine verbindungslose Ende zu Ende Verbindung. Nachrichten wie CAM oder DENM der Facility-Schicht werden vom BTP durch Multiplexing für das GeoNetworking Protokoll aufbereitet, sodass diese über GeoNetworking übertragen werden und am Ziel durch De-Multiplexing weiter verarbeitet werden können. Abbildung 2.7 zeigt den

Abbildung 2.7: Struktur eines BTP Pakets

Aufbau der BTP Paket Struktur. Der MAC Header ist der entsprechende Header einer ITS Zugriffstechnologie - beispielsweise ITS-G5. Auf den MAC Header folgt der GeoNetworking Header und optional der GeoNetworking Security Header. Das BTP-Protokoll besteht lediglich aus einem 4-Byte Header der das Handling der Daten zwischen Facility Layer und GeoNetworking bestimmt. Dafür beinhaltet der BTP-Header den entsprechenden Source- und Destinationport. Auf den Header folgen die eigentlichen Nutzdaten (Payload), welche die Daten von CAM oder DENM (siehe Abschnitt 2.2.2.3) enthalten [32].

2.2.2.3 Facilities

Die Facility-Schicht liegt zwischen Anwendungs- und Transportschicht und ist im amerikanischen WAVE Standard in dieser Weise nicht vorgesehen. Die Facility-Schicht definiert Nachrichten wie Cooparative Awareness Messages (CAM) und Decentralized Environmental Nofification Messages (DENM).

Cooperative Awareness Message (CAM)

Die CAMs werden innerhalb des VANETs von allen mobilen ITS-Stationen wie PKWs, Einsatzfahrzeuge und öffentlichen Fahrzeugen gesendet und enthalten Information über Position und Status eines sendenden Fahrzeugs. Die CAM wird von einem Fahrzeug ausgesendet und von ITS-Stationen empfangen, die sich in Reichweite befinden. Eine empfangene CAM wird jedoch nicht weitergeleitet, was bedeutet dass diese Informationen nur direkt zwischen Nachbarn ausgetauscht werden (single hop distance). Die Idee der CAM ist die Übermittlung der "Hier bin ich"-Information. Ein Empfänger entscheidet über Relevanz der Informationen und kann gegebenenfalls reagieren. Abbildung 2.8 zeigt die Struktur der CAM spezifiziert nach ETSI TS

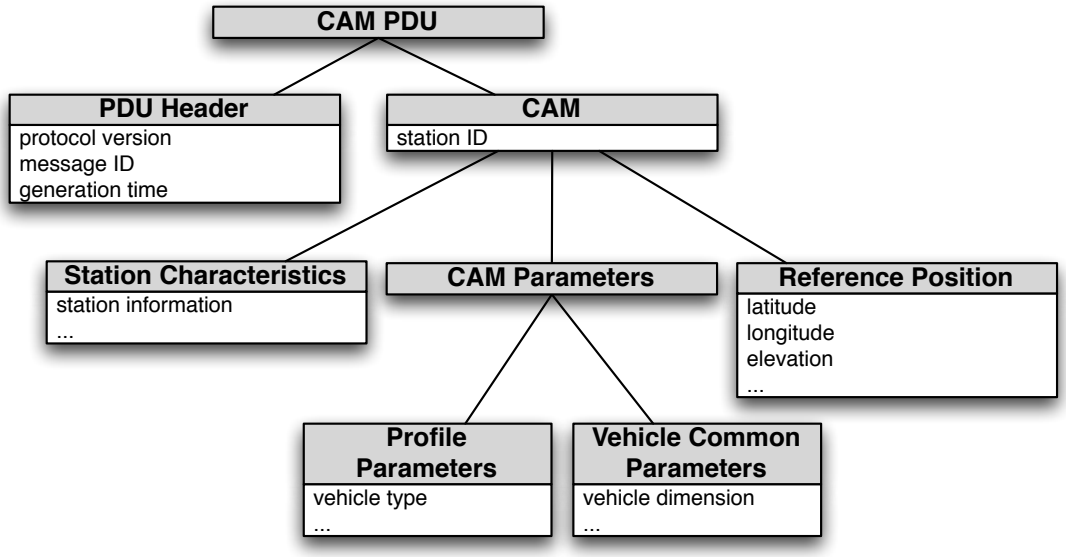

Abbildung 2.8: CAM Struktur mit dazugehörigen Daten. Die drei Punkte symbolisieren weitere Variablen, die im Standard [33] ersichtlich sind.

102 637-2 [33]. Die CAM Protocol Data Unit (PDU) enthält die Objekte PDU Header und CAM. Der PDU Header enthält Grundinformationen zur CAM. Das Objekt CAM enthält die ID der Station sowie die Objekte Station Characteristics, CAM Parameters und Reference Position. Station Characteristics gibt Auskunft über stationsrelevante Daten, wie beispielsweise ob die Station mobil ist oder nicht. Das

Objekt CAM Parameters enthält die beiden Objekte Vehicle Common Parameters und Profile Parameters. Vehicle Common Parameters enthält allgemeine Fahrzeugdaten wie Fahrzeugabmessungen oder Geschwindigkeit. Die Profile Parameters sind abhängig vom Fahrzeugtyp und enthalten beispielsweise für den Typ Einsatzfahrzeug die Information ob die Warneinrichtungen eingeschaltet sind. Die Reference Position enthält Positionsdaten wie Längen-, Breitengrad und Fahrtrichtung.

Decentralized Environmental Notification Message (DENM)

Decentralized Environmental Notification Messages werden durch bestimmte Ereignisse ausgelöst und dienen zur Benachrichtigung der Verkehrsteilnehmer. Im Gegensatz zu CAM werden DENM nicht permanent gesendet, sondern nur wenn ein Ereignis auftritt. DENMs werden im Vergleich zu CAMs nicht nur von mobilen ITS Stationen sondern auch von stationären ITS Stationen wie beispielsweise einer temporären Baustelle gesendet. Abbildung 2.9 zeigt die Struktur der DENM spezifiziert

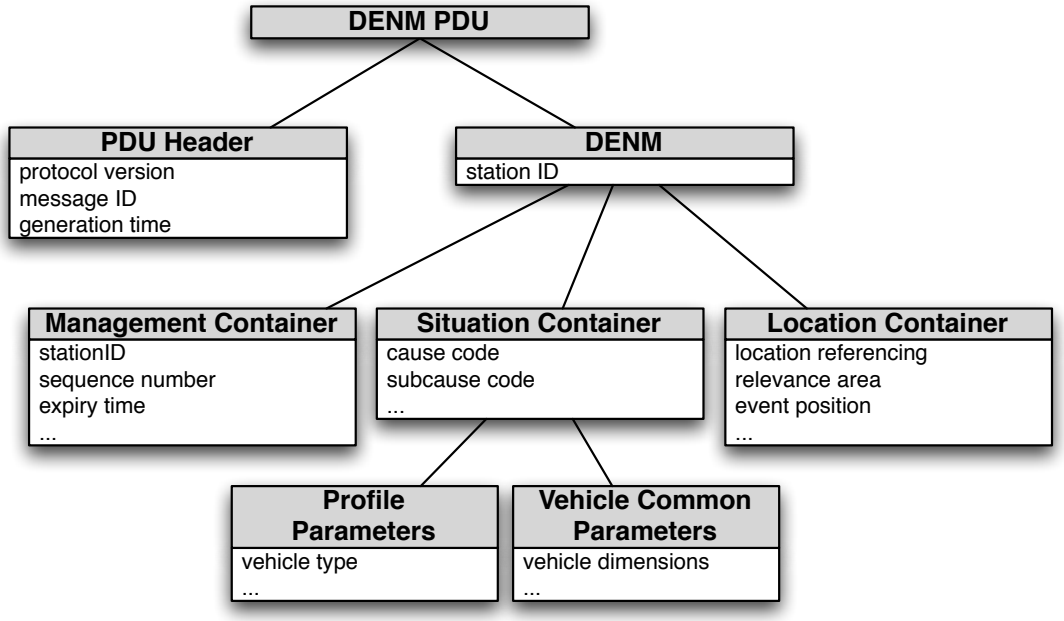

Abbildung 2.9: DENM Struktur mit dazugehörigen Daten. Die drei Punkte symbolisieren weitere Variablen, die im Standard [34] ersichtlich sind.

nach ETSI TS 102 637-3 [34]. Der DENM Header entspricht dem CAM Header. Das DENM Objekt enthält die drei Container Management, Situation und Location. Der Management Container enthält die ID der Station, die Sequenznummer und Gültigkeitsdauer der Nachricht. Der Situation Container beinhaltet Informationen darüber wodurch die DEN Nachricht ausgelöst wurde. Diese Informationen sind

im Cause Code und Subcause Code hinterlegt. Die Objekte Vehicle Common Parameters und Profile Parameters entsprechen den gleichnamigen Objekten der CAM. Der Location Container enthält Informationen zum Ereignisort wie Längengrad und Breitengrad und wieweit sich das Ereignis lokal auswirkt.

2.2.2.4 Applications

Die Applications-Schicht beinhaltet die Standards der ITS-Anwendungen. Dabei sind die ITS-Anwendungen der Application-Schicht im Standard ETSI TR 102 638 [35] definiert. Abbildung 2.10 zeigt die Klassifizierung der ITS-Anwendungen nach ETSI. Die

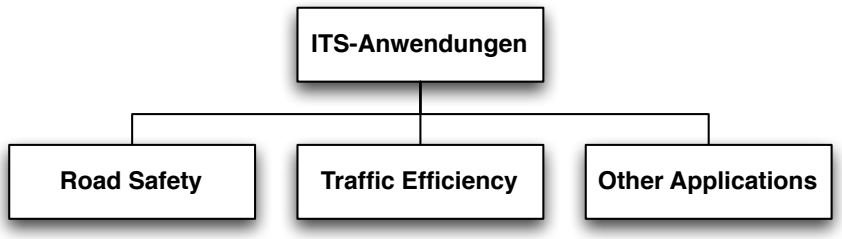

Abbildung 2.10: Klassifizierung der ITS-Anwendungen nach ETSI

ITS-Anwendungen werden in Road Safety, Traffic Efficiency und Other Applications klassifiziert.

- **Road Safety**
  Road Safety Anwendungen dienen der Vermeidung von Unfällen, indem die Anwendung den Fahrer frühzeitig vor einem Unfall warnt oder bei hoher Wahrscheinlichkeit eines unvermeidbaren Unfalls autonom in das Fahrzeug eingreift. Letztere fordern ein hohes Maß an Sicherheit und sind daher für eine erste Markteinführung noch nicht vorgesehen.

- **Traffic Efficiency**
  Traffic Efficiency Anwendungen optimieren den Verkehrsfluss um Staus zu vermeiden und somit einen Beitrag zum Umweltschutz zu leisten.

- **Other Applications**
  Zu den restlichen Anwendungen zählen Service- und Komfortanwendungen wie beispielsweise Zahldienste, Zugangskontrollen oder Internetzugang.

2.2.2.5 Security

Im Bereich der Sicherheitsstandardisierungen sind derzeit die amerikanischen Standards für WAVE durch IEEE 1609.2 weiter fortgeschritten als die europäischen Sicherheitsstan-

dards für ITS nach ETSI. Der IEEE 1609.2 Standard spezifiziert Systemarchitektur und Nachrichtenformat im Kontext Sicherheit [36]. ETSI definiert die Sicherheitsstandards, wie in Abbildung 2.5 ersichtlich, schichtübergreifend und spezifiziert die Sicherheitsmaßnahmen in Anlehnung an IEEE 1609.2. Der ETSI Standard TS 102 867 [37] gibt Aufschluss darüber, welche Sicherheitsmaßnahmen vom IEEE 1609.2 übernommen beziehungsweise adaptiert werden. Anhang B gibt einen Überblick der ETSI Standards. Der Technische Report TR 102 893 [38] beinhaltet die grundsätzliche Sicherheits-Risikoanalyse. Die übrigen Standards in Anhang B beziehen sich auf diese Risikoanalyse und definieren entsprechende Maßnahmen. In Abschnitt 2.3 wird auf die Sicherheitsanforderungen und die grundlegende Systemarchitektur eingegangen.

2.2.2.6 Management

Die Management-Schicht definiert schichtübergreifende Standards wie Adressierungs- und Kommunikationsschemen. Einen wichtigen Standard stellt vor allem die Maßnahme des Decentralized Congestion Control dar:

Decentralized Congestion Control

Grundsätzlich gilt für das VANET, dass es stets für zeitkritische Anwendungen verfügbar sein muss. Das VANET darf auch bei einer großen Anzahl von Teilnehmern nie voll ausgelastet sein, um die Übertragung von zeitkritischen Nachrichten zu gewährleisten. Die Anzahl der Teilnehmer eines VANETs ist jedoch nicht beschränkt. Laut [25] kann es durch das CSMA Zugriffsverfahren zu einer beträchtlichen Verzögerung der Datenübertragung kommen, da bei einer hohen Teilnehmerdichte nicht gewährleistet werden kann, dass jeder Teilnehmer Zugriff bekommt. Diese Tatsache ist für zeitkritische Anwendungen inakzeptabel und fordert daher Maßnahmen, die die Überlastung des VANETs verhindern. Decentralized Congestion Control (DCC) stellt in der Management Schicht (Abbildung 2.5) eine schichtübergreifende Funktionalität dar, die eine Überlastung des VANETs verhindert. Der Standard ETSI TS 102 687 [39] spezifiziert DCC-Maßnahmen wie Transmit Power Control (TPC), Transmit Rate Control (TRC) und Transmit Datarate Control (TDC), welche in der Access Schicht angesiedelt sind. Konkrete Algorithmen für DCC, welche in höheren Schichten angesiedelt sind, werden Ende 2013 im ETSI Standard TS 103 175 (siehe Anhang B) veröffentlicht [39].

2.2.3 Zusammenfassung

In diesem Abschnitt wurden die Standardisierungen nach IEEE (USA) und ETSI (Europa) behandelt. Wie beschrieben, verfolgen die Länder dabei unterschiedliche Ziele. In den USA liegt das Hauptziel von Car-to-X auf infrastrukturbasierter Kommunikation,

während in Europa der Fokus auf infrastrukurloser Kommunikation liegt. Als Sammelbezeichnung für den USA spezifischen Protokollstack wird das Akronym WAVE (Wireless Access for the Vehicular Environment) verwendet. Im europäischen Raum wird nach ETSI das Gesamtsystem als Cooperative-Intelligent Transportation System C-ITS bezeichnet. Im Bezug auf das ISO/OSI-Schichtenmodell teilen die beiden Standards die Schichten des Protokollstacks nach unterschiedlichen Kriterien ein (Vergleich Abbildung 2.4 zu Abbildung 2.5). Die Wireless Übertragung nach ETSI ITS-G5 ist vom IEEE 802.11p abgeleitet. Zu den größten Unterschieden zählen die unterschiedlich spezifizierten Kanäle. Weiters werden nach IEEE 802.11p die einzelnen Kanäle über einen Transceiver zeitlich umgeschaltet, während die ITS-G5 Spezifikation zwei Transceiver vorsieht, um gleichzeitig auf zwei Kanälen zu kommunizieren. In den höheren Schichten unterscheiden sich die Kommunikationsprotokolle, auch wenn inhaltlich ähnliche Informationen übermittelt werden.

2.3 Sicherheitsarchitektur und Kryptographie

Wie in Kapitel 1 beschrieben, bringt die Car-to-X Kommunikation große Vorteile in Punkto Sicherheit und Effizienz mit sich. Die Vernetzung von Fahrzeugen birgt jedoch auch die Gefahr von Datenmissbrauch und Manipulation in sich. Bis dato wirken sich vorhandene Sicherheitslücken auf einzelne Fahrzeuge aus. Mit Einführung der Car-to-X Kommunikation könnten Sicherheitslücken alle Fahrzeuge innerhalb eines VANETs betreffen. Auch Datenschutz ist ein wichtiges Thema, denn CAM und DENM enthalten Informationen, die beispielsweise zur Verfolgung von Fahrzeugen missbraucht werden könnten und daher dem Datenschutz unterliegen.

2.3.1 Grundlegendes

Eine Vielzahl von Publikationen befasst sich mit dem Thema Sicherheit von VANETs. Die Publikation „Security Analysis of Vehicular Ad Hoc Networks" [40] analysiert Sicherheitsproblematiken und definiert wie auch die Publikation „How to Secure ITS Applications?" [36] folgende Basisziele.

- **Anonymität**
  Ein Empfänger darf nicht in der Lage sein, anhand einer Nachricht die wahre Identität des Senders festzustellen.
- **Authentizität**
  Der Empfänger einer Nachricht muss überprüfen können, ob die Nachricht von einem vertrauenswürdigen Sender stammt. Die eigentliche Identität des Senders ist jedoch nicht von Bedeutung.
- **Integrität**
  Es muss sichergestellt sein, dass empfangene Nachrichten nicht manipuliert worden sind.
- **Nichtabstreitbarkeit**
  Gesendete Nachrichten müssen anonymisiert sein, jedoch muss bei zu reglementierenden Umständen die Möglichkeit bestehen, die wahre Identität des Senders festzustellen.
- **Nichtnachverfolgbarkeit**
  Es darf keine Möglichkeit bestehen, Fahrzeuge anhand der gesendeten Nachrichten verfolgen zu können.

Spezielle ITS-Anwendungen erfordern auch noch weitere Ziele wie Autorisierung verschiedener Protokolle, Plausibilitätsüberprüfungen, Vertraulichkeit, Verfügbarkeit und Echtzeitfähigkeit. Die Publikation [36] gibt einen Überblick über die ITS-Anwendungen und

die dafür notwendigen Sicherheitsziele. Die prinzipielle Public Key Infrastructure (PKI) zur Absicherung der Ziele ist in Abbildung 2.11 dargestellt. Zertifizierung und Signierung funktionieren folgender Maßen: [41]

Die OBU X eines Fahrzeugs besitzt eine einzigartige Identität ID_X, die mit dem geheimen Schlüssel SK_X und dem öffentlichen Schlüssel PK_X verknüpft ist. Die Zertifizierungsstelle CA bestätigt anhand der Schlüssel und den technischen Eigenschaften der On-Board-Unit ein Teilnehmerzertifikat $Cert_A(X)$. Die OBU erstellt neue Schlüsselpaare $\{SK_X^1, PK_X^1 \cdots SK_X^n, PK_X^n\}$, welche die CA durch die technischen Eigenschaften bestätigt und zertifiziert. Die OBU erhält von der CA das Schlüsselpaar (SK_X^i, PK_X^i) und das Zer-

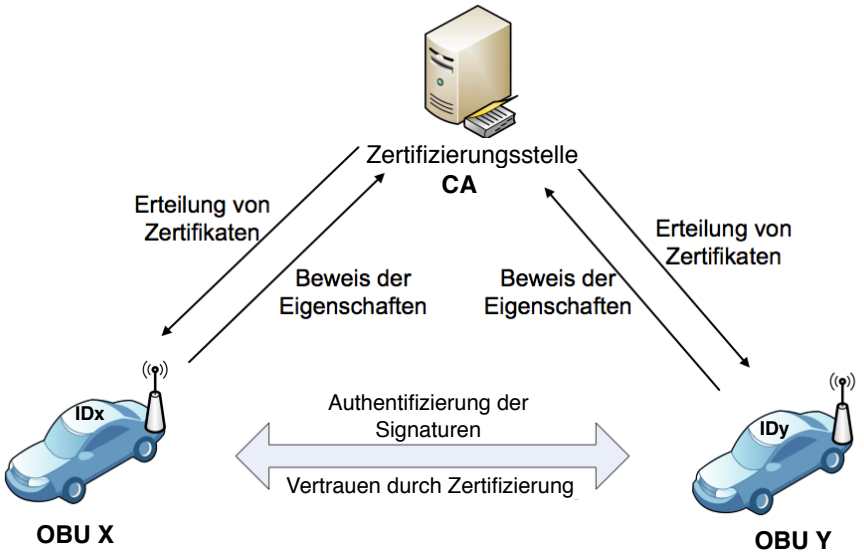

Abbildung 2.11: Vereinfachte Sicherheitsarchitektur des VANETs [41]

tifikat $Cert_A(PK_X^i)$, die Pseudonyme darstellen. Für den Vorgang der Erteilung von Zertifikaten und dem Beweis der Eigenschaften zwischen CA und OBU muss Vertraulichkeit herrschen. Eine Nachricht innerhalb des VANETs wird nun mit Hilfe der Pseudonyme des geheimen Schlüssels SK_X^i vom Sender signiert und mit dem dazugehörigen Zertifikat $Cert_A(PK_X^i)$ versendet. Der Empfänger kann mit Hilfe des öffentlichen Schlüssels PK_X^i die Nachricht überprüfen.

Durch digitale Signaturen werden die Ziele Authentizität, Integrität und Nichtabstreitbarkeit erreicht [42]. Teilnehmer innerhalb des VANETs erhalten anhand ihrer Identität von der Zertifizierungsstelle Pseudonyme. Diese Pseudonyme werden regelmäßig durchgetauscht. Nur die Zertifizierungsstelle kennt die Identiät ID_X. Somit sind Anonymität und Nichtnachverfolgbarkeit gewährleistet. Nichtabstreitbarkeit ist gegeben, da die Zertifizie-

rungsstelle bei zu reglementierenden Umständen den Zusammenhang zwischen Identität und Pseudonym wiederherstellen kann.

2.3.2 Spezifizierte Verschlüsselungsverfahren

Die Sicherheit des Car-to-X Systems muss grundsätzlich über die gesamte Lebensdauer eines Fahrzeugs gewährleistet werden können. Im Bereich der Sicherheitsstandardisierungen sind derzeit die amerikanischen Standards für WAVE durch IEEE 1609.2 weiter fortgeschritten als die europäischen Sicherheitsstandards für ITS nach ETSI [36]. Prinzipiell führt eine Erhöhung der Sicherheitsstufe durch aufwändigere Verschlüsselungsverfahren zu einer längeren Latenzzeit. Um die Anforderungen im Hinblick auf Sicherheit und Latenz erfüllen zu können, sieht der IEEE 1609.2 Standard laut [43] folgende Verschlüsselungsverfahren vor:

- **Elliptic Curve Digital Signature Standard (ECDSA)**
  Als Hauptalgorithmus ist der Digitale Signatur Algorithmus ECDSA [44] vorgesehen. Jede Broadcast-Nachricht wird mit dem Private Key des Senders signiert. Das Zertifikat wird zusammen mit der signierten Nachricht übermittelt. Der Empfänger verifiziert die Nachricht mit Hilfe des Public Key und kann somit die Integrität der Daten überprüfen. Über den Zeitstempel der Nachricht wird zusätzlich die Aktualität der Daten sichergestellt.

- **Elliptic Curve Integrated Encryption Scheme (ECIES)**
  ECIES [45] ist wie ECDSA ein asymmetrisches Verschlüsselungsverfahren. ECIES könnte prinzipiell zur Verschlüsselung von Nutzdaten verwendet werden, der IEEE 1609.2 Standard sieht ECIES zum Austausch von symmetrischen Schlüsseln vor. Dieser symmetrische Schlüssel wird für das Authenticated Encryption Verfahren verwendet. Durch den zuvor stattfindenden Schlüsselaustausch über ECIES wird der darauf folgende Datenaustausch über das Authenticated Encryption Verfahren beschleunigt.

- **Authenticated Encryption (AES-CCM)**
  Das Authenticated Encryption Verfahren ist ein rein symmetrisches Verschlüsselungsverfahren und ist in erster Linie zur effizienten Überprüfung der Datenintegrität vorgesehen. Optional kann auch die Datenverschlüsselung erfolgen. Vorteil dieses Verfahren ist der geringe Overhead. IEEE 1609.2 sieht AES-CCM [46] für Unicast-Nachrichten und Softwaredownloads vor.

2.3.3 Anforderungen an die Sicherheitsarchitektur

Die beiden Verfahren ECDSA und ECIES sind asymmetrische Verfahren und beruhen auf der Elliptic Curve Cryptographie (ECC). Die Sicherheit der ECC basiert mathematisch auf der Schwierigkeit der Lösung des diskreten Logarithmus. Zudem hat das ECC-Verfahren gegenüber dem klassischen RSA-Verfahren den Vorteil, dass mit kleiner Schlüssellänge die gleiche Sicherheitsstufe erreicht werden kann [41]. Um die möglichen Anforderungen an die Sicherheitsarchitektur abzuschätzen, wird in [43] von folgendem Worst-Case-Szenario ausgegangen:
Auf einer Autobahnkreuzung mit vier mal drei Spuren kommt es zu einem Stau. Bei einer 500 Meter Reichweite des 802.11p WLANs und angenommenen 400 Fahrzeugen muss eine OBU durch die regelmäßig empfangenen CAM-Nachrichten (Abschnitt 2.2.2.3) der anderen Verkehrsteilnehmer trotz adäquaten DCC (Abschnitt 2.2.2.6) bis zu 4000 Signaturen pro Sekunde verifizieren. Diese Tatsache stellt hohe Performance-Anforderungen an die Sicherheitsarchitektur und ist laut [43] nur mit geeigneter Hardwareunterstützung durchführbar.

2.3.4 Zusammenfassung

Das Thema Sicherheit und Kryptographie ist eine der essenziellen Grundlagen der Car-to-X Kommunikation. Die Standardisierungen nach IEEE 1609.2 sind derzeit weiter fortgeschritten als die europäischen Sicherheitsstandards für ITS nach ETSI. In den Standards ist die Verwaltung der Zertifikate durch eine PKI noch nicht vorgesehen. Die PKI muss jedoch zur Car-to-X Markteinführung vorhanden sein, um potentiellen Gefährdungen durch Datenmanipulation und -missbrauch entgegenzuwirken. Dieser Abschnitt behandelte die grundlegenden Sicherheitsbasisziele sowie die spezifierten Verschlüsselungsverfahren nach IEEE 1609.2. Die Publikation [43] geht auf die grundlegenden Anforderungen der Sicherheitsarchitektur ein. Aus dieser Publikation geht hervor, dass die kryptographischen Operationen in Abhängigkeit von Verkehrsaufkommen sehr hohe Performance-Anforderungen an die Hardware der OBU stellen können. Eine garantierte Signaturverifikation ist laut [43] nur mit geeigneter Hardwareunterstützung durchführbar.

Kapitel 3

Plattformauswahl

Realisierungen im Bereich der Car-to-X Kommunikation beschränken sich bislang auf Versuchsaufbauten und Feldtests. Wie bereits in Abschnitt 2.1.1 beschrieben, bilden im sim^{TD}-Feldtest PC und Router zusammen eine OBU (Abbildung 2.3). Für die Implementation dieser Arbeit wird eine integrierte Plattform gewählt, welche im Bezug auf Hardware- und Softwarekomponenten eine hohe Eignung für die Integration in zukünftigen Serienfahrzeugen aufweist. In Abschnitt 3.1 dieses Kapitels werden die Kriterien zur Auswahl der Plattform spezifiziert. Abschnitt 3.2 beschreibt die derzeit erhältlichen Plattformen. Hardware- und Softwarearchitektur der gewählten Plattform sowie die dazugehörige Entwicklungsumgebung sind in Abschnitt 3.3 beschrieben.

3.1 Auswahlkriterien

Auf der Plattform wird das Use-Case Szenario (siehe Abschnitt 4.1) implementiert. Zur Auswahl der Hardware und der zugehörigen Software werden folgende Kriterien definiert:

- **Serienreife**
  Die Plattform soll nach Möglichkeit Potential zur Serienreife haben. Dahinter steht die Zielsetzung, das Konzept der Hardwareplattform in zukünftigen Serienfahrzeugen zu verwenden. Die Serienreife bezieht sich sowohl auf die Hardwarearchitektur wie auch auf die Softwarearchitektur.

- **Referenzen**
  Das Kriterium beschreibt, ob auf Grund vorhandener Referenzen Einschätzungen zur Eignung der Plattform gemacht werden können. Dazu zählen Publikationen über Laboraufbauten oder Feldtests, die vorab Rückschlüsse auf die Eignung erlauben.

- **ITS-G5**
  Für dieses Kriterium wird die ITS-G5 Tranceiver-Einheit der Plattform bewertet. Das Hauptaugenmerk liegt einerseits auf dem implementierten Chipset (Chip-on-Board oder miniPCI), andererseits auf mögliche Konfigurationsmodi für Einkanal- und Zweikanal-Betrieb. Die ITS-G5 Spezifikation sieht die Möglichkeit des simultanen Zweikanal-Betriebs (ohne zeitliches Umschalten) vor. Aus diesem Grund soll nicht nur ein Einkanal-Betrieb, sondern auch der Zweikanal-Betrieb möglich sein.

- **Schnittstellen**
  Abgesehen von der ITS-G5-Schnittstelle, zählt zur wichtigsten Schnittstelle die CAN-Schnittstelle, wodurch ein Datenaustausch mit dem Fahrzeug möglich wird. Für CAM und DENM sind Positionsdaten über GPS notwendig, daher sollte nach Möglichkeit bereits ein interner GPS-Receiver verfügbar sein.

- **Kryptographie**
  Wie in Abschnitt 2.3.3 beschrieben, stellen kryptographische Operationen hohe Performance-Anforderungen an die Hardware. Um diese Anforderungen auch bei hoher Verkehrsdichte zu erfüllen, muss die Signaturverifizierung Hardware unterstützt erfolgen.

- **Protokollstack**
  Abschnitt 2.2 beschreibt die regionalen Unterschiede in den USA und Europa. Die Protokollstacks für Europa (C-ITS) und den USA (WAVE) müssen standardkonform verfügbar sein. Das ermöglicht die Implementation von Anwendungen in den oberen Schichten des ISO/OSI-Schichtenmodells, ohne in tiefere Schichten wie beispielsweise die Netzwerkschicht eingreifen zu müssen.

- **Graphical User Interface (GUI)**
  Der spezifizierte Use-Case in Abschnitt 4.1 benötigt ein Human Machine Interface (HMI). Wie in Abbildung 4.1 zu sehen ist, soll der Fahrer visuell über eine Meldung am Monitor informiert werden. Daher müssen Hardware- und Softwareschnittstellen zur Interaktion mit dem Fahrer vorhanden sein.

- **Software Development Kit (SDK)**
  Zur Umsetzung des in Abschnitt 4.1 spezifierten Use-Case wird die entsprechende ITS-Anwendung implementiert. Zur Entwicklung der ITS-Anwendung soll ein SDK verfügbar sein. Im optimalen Fall gibt es über das SDK die Möglichkeit, entwickelte Anwendungen über Remote-Debugging direkt auf der Hardware zu testen.

3.2 Verfügbare Plattformen und Bewertung

Die derzeit erhältlichen OBU-Plattformen werden in diesem Abschnitt im Bezug auf die Auswahlkriterien K_i (Abschnitt 3.1) bewertet. In den jeweiligen Tabellen sind N einzelne Kriterien K_i mit einer kurzen Erläuterung und einer Bewertung G_i angeführt. Den Kriterien K_i wird ein Gewicht $G_i \in \{0, 0.5, 1\}$ zugeteilt wobei 0 das Kriterium K_i nicht erfüllt und 1 das Kriterium K_i zur Gänze erfüllt. Die Gesamtbewertung G_{ges} berechnet sich zu:

$$G_{ges} = \frac{\sum_{i=1}^{N} G_i}{N} \cdot 100$$

3.2.1 AutoTalks

Die israelische Firma AutoTalks [47] vertreibt die Hardwareplattform Pangaea-3. Die Plattform besteht aus einem von AutoTalks entworfenen Chip, der eine 360 MHz ARM-Prozessorarchitektur besitzt. Des Weiteren verfügt der Chip über einen Co-Prozessor, der für die Verschlüsselungsverfahren (Abschnitt 2.3.2) optimiert ist. Der AutoTalks ITS-G5 Transceiver ist für den Car-to-X Einsatz entwickelt worden und unterstützt sowohl den Einkanal- als auch den Zweikanal-Betrieb. Über eine Tablet-Anwendung werden die ent-

AutoTalks Pangaea-3		
Kriterium K_i	**Erläuterung**	G_i
Serienreife	ja (Embedded System)	1
Referenzen	SafetyPilot, sim^{TD}	1
ITS-G5	Ein-/Zweikanal (Chip-on-Board)	1
CAN/Ethernet/GPS	ja/ja/intern	1
Kryptographie	Hardware unterstützt	1
Protokollstack	C-ITS, WAVE	1
GUI	Tablet-Anwendung	1
SDK	ja	1
Gesamtbewertung G_{ges}		100 %

Tabelle 3.1: Bewertung AutoTalks Pangaea-3

sprechenden ITS-Anwendungen visualisiert. Durch das On-Chip Design ist Pangaea-3 eine im entsprechenden Vergleich weit entwickelte Plattform, die Potential zur Serienreife hat. Die Plattform erfüllt alle Kriterien K_i und erlangt daher eine Gesamtwertung G_{ges} von 100%. Bedingt durch die späte Verfügbarkeit (April 2013) stellt sie jedoch für die Implementierung in dieser Arbeit keine Option dar.

3.2.2 Cohda

Die MK-2 Plattform von Cohda [48] ist mit einem 533 MHz ARM-Prozessor ausgestattet. Verschlüsselungsverfahren werden hardwareunterstützt durchgeführt. Die CohdaMobility Einheit der MK-2 Plattform ermöglicht den Betrieb im Einkanal- und Zweikanal-Betrieb. Anwendungen können über den VGA-Ausgang am Monitor visualisiert werden. Das Hard-

Cohda MK-2		
Kriterium K_i	**Erläuterung**	G_i
Serienreife	ja (Embedded System)	1
Referenzen	SafetyPilot, sim^{TD}	1
ITS-G5	Ein-/Zweikanal (Chip-on-Board)	1
CAN/Ethernet/GPS	ja/ja/intern	1
Kryptographie	Hardware unterstützt	1
Protokollstack	C-ITS, WAVE	1
GUI	Monitor via VGA	1
SDK	ja (Remote-debugging)	1
Gesamtbewertung G_{ges}		100 %

Tabelle 3.2: Bewertung Cohda MK-2

waredesign der MK-2 Plattform kann wie AutoTalks Pangaea-3 als seriennah eingestuft werden, da die einzelnen Komponenten als Chip-On-Board Implementierungen vorliegen. MK-2 erfüllt alle spezifizierten Kriterien K_i und erreicht in der Gesamtwertung G_{ges} ebenfalls 100%. Aufgrund der ausgezeichneten Gesamtwertung und der Verfügbarkeit wird die Plattform MK-2 als Implementationsplattform herangezogen. Eine detaillierte Beschreibung der Plattform findet sich in Abschnitt 3.3.

3.2.3 Fraunhofer

Fraunhofer ESK stellt mit dem ARTiS-XT [49] eine Plattform für die Entwicklung von Infotainment- und Telematikanwendungen zur Verfügung. ARTiS-XT besteht aus einem Embedded-PC, der über USB mit der dazugehörigen Echtzeitplattform kommuniziert. Der Embedded-PC besitzt einen 1,33 GHz Intel-Atom Prozessor auf dem die ITS-Anwendungen ausgeführt werden, die Echtzeitplattform verfügt über einen 132 MHz Freescale Prozessor. Über die Echtzeitplattform werden die Schnittstellen zu CAN und Ethernet bereit gestellt. Einkanal und Zweikanal-Betrieb für ITS-G5 werden über zwei miniPCI-Slots des Embedded-PCs realisiert in denen sich jeweils eine miniPCI Karte mit einem Qualcomm Atheros Chipset befindet. ARTiS-XT eignet sich für PC-Demonstratorlösungen kann jedoch auf Grund des Hardwareaufbaus nicht in dieser Art in eine seriennahe Hardwareplattform umgesetzt werden. Darüber hinaus sind die miniPCI-Chipsets nur bedingt au-

Fraunhofer ARTiS-XT		
Kriterium K_i	**Erläuterung**	G_i
Serienreife	nein (PC)	0
Referenzen	keine	0
ITS-G5	Ein-/Zweikanal (miniPCI)	0,5
CAN/Ethernet/GPS	ja/ja/intern	1
Kryptographie	keine Hardwareunterstüzung	0
Protokollstack	C-ITS, WAVE	1
GUI	ja via DVI	1
SDK	keine Angaben	0
Gesamtbewertung G_{ges}		44 %

Tabelle 3.3: Bewertung Fraunhofer ARTiS-XT

tomotive tauglich. ARTiS-XT erreicht aus diesen Gründen in der Gesamtbewertung G_{ges} 44 %. Die genaue Aufschlüsselung ist in Tabelle 3.4 ersichtlich.

3.2.4 NEC

NEC bietet mit dem LinkBird-MX-4 [50] eine Prototypenplattform in der mittlerweile 4. Generation. LinkBird besitzt einen 266 MHz Mikroprozessor, unterstützt keine hardwareunterstützte Verschlüsselung. Zur Visualisierung von ITS-Anwendungen ist weitere Hardware notwendig, da auf die Plattform nur über ein Webinterface zugegriffen werden kann. Wie beim Fraunhofer ARTiS-XT ist die ITS-G5 Kommunikation über zwei miniPCI Karten mit jeweils einem Qualcomm Atheros Chipset realisiert. Die LinkBird Plattform eignet sich beispielsweise zur Protokollevalierung, ist jedoch auf Grund der mini-PCI Lösung und der fehlenden Verschlüsselungsunterstützung keine seriennahe Hardwareplattform. Daher

NEC LinkBird-MX-4		
Kriterium K_i	**Erläuterung**	G_i
Serienreife	nein	0
Referenzen	sim^{TD}	0,5
ITS-G5	Ein-/Zweikanal (miniPCI)	0,5
CAN/Ethernet/GPS	ja/ja/extern	1
Kryptographie	keine Hardwareunterstüzung	0
Protokollstack	C-ITS	0,5
GUI	Webinterface	0,5
SDK	ja	1
Gesamtbewertung G_{ges}		50 %

Tabelle 3.4: Bewertung NEC LinkBird-MX-4

erreicht die Plattform in der Gesamtbewertung G_{ges} 50 %. Auf Anfrage bei NEC wurde bekanntgegeben, dass die Plattform nicht mehr lieferbar ist.

3.2.5 UNEX

UNEX ist Anbieter von Industriefunklösungen und vertreibt im Sektor Car-to-X die Plattformen OBE-101 (WAVE) und OBE-102 (C-ITS). Die Plattform OBE-102 [51] verfügt über einen 400 MHz Freescale MPC. Verschlüsselungsverfahren werden von der Plattform nicht unterstützt. Zur Kommunikation besitzt die Plattform den selben Atheros miniPCI

UNEX OBE-102		
Kriterium K_i	**Erläuterung**	G_i
Serienreife	nein	0
Referenzen	nein	0
ITS-G5	Einkanal (miniPCI)	0
CAN/Ethernet/GPS	ja/ja/extern	1
Kryptographie	keine Hardwareunterstüzung	0
Protokollstack	C-ITS	0,5
GUI	Webinterface	0,5
SDK	ja	1
Gesamtbewertung G_{ges}		38 %

Tabelle 3.5: Bewertung UNEX OBE-102

Chipset wie Fraunhofer ARTiS-XT und NEC LinkBird-MX-4. Bedingt durch die miniPCI Karte, der fehlenden Möglichkeit des Zweikanal-Betriebs und der fehlenden Unterstützung von Kryptographie wird die Plattform als nicht seriennah eingestuft und erhält eine Gesamtwertung G_{ges} von 38 %.

3.2.6 Zusammenfassung

Tabelle 3.6 liefert eine Zusammenfassung der Gesamtbewertungen der einzelnen Hardwareplattformen aus Abschnitt 3.2. Die Plattform Pangaea-3 erfüllt alle Kriterien, stellt jedoch auf Grund der späten Verfügbarkeit (April 2013) für die Implementierung in dieser Arbeit keine Option dar. Cohda bietet mit der Plattform MK-2 ein seriennahes Gerät mit einem eigens entwickelten Chip-on-Bord Transceiver, welcher die 802.11p und ITS-G5 Spezifikation erfüllt. Die MK-2 Plattform erfüllt alle Kriterien und wird daher als Implementierungsplattform herangezogen. ARTiS-XT von Fraunhofer stellt eine PC-Demonstratorlösung dar, welche die Anforderungen eines seriennahen Embedded-Systems nicht erfüllt. NEC LinkBird-MX-4 kommuniziert über eine für den Automobileinsatz ungeeignete 802.11p miniPCI Karte, bietet keine Verschlüsselungsunterstützung und ist in absehbarer Zeit nicht lieferbar. OBE-102 von UNEX ist aus Hardwaresicht ähnlich der LinkBird-MX-4 Plattform, bietet keine Möglichkeit zum Zweikanal-Betrieb und erreicht 38 % in der Gesamtwertung.

Hersteller	**Plattform**	G_{ges}
AutoTalks	Pangaea-3	100 %
Cohda	MK-2	100 %
Fraunhofer	ARTiS-XT	44 %
NEC	LinkBird-MX-4	50 %
UNEX	OBE-102	38 %

Tabelle 3.6: Gesamtbewertung der Plattformen

3.3 Gewählte Plattform Cohda MK-2

Die Plattform Cohda MK-2 wurde anhand der Kriterien in Abschnitt 3.1 als Implementationsplattform gewählt. MK-2 ist eine gemeinsame Entwicklung der Firmen Cohda und NXP. Cohda entwickelt die prototypische Hardware und die dazugehörige Software. Die Signalverarbeitung ist von Cohda in FPGAs (Field Programmable Gate Arrays) umgesetzt. Die längerfristige System-on-Chip Lösung entwirft NXP. Im Folgenden wird detailliert auf die Software- und Hardwarearchitektur der Plattform eingegangen.

3.3.1 Hardwarearchitektur

Abbildung 3.1 zeigt den Überblick der MK-2 Hardwarearchitektur. Die Hauptkomponenten bestehen aus dem Power Supply-, dem Radio RF-, dem FPGA und dem Application Processor Sub-System. Radio RF- und FPGA Sub-system bilden gemeinsam die Cohda Mobility 802.11p Kommunikationseinheit. Auf dem Application Processor Sub-system läuft das Betriebssystem mit den ITS-Anwendungen. Der Datenaustausch zwischen Application Processor und FPGA erfolgt über einen Speicherbus.

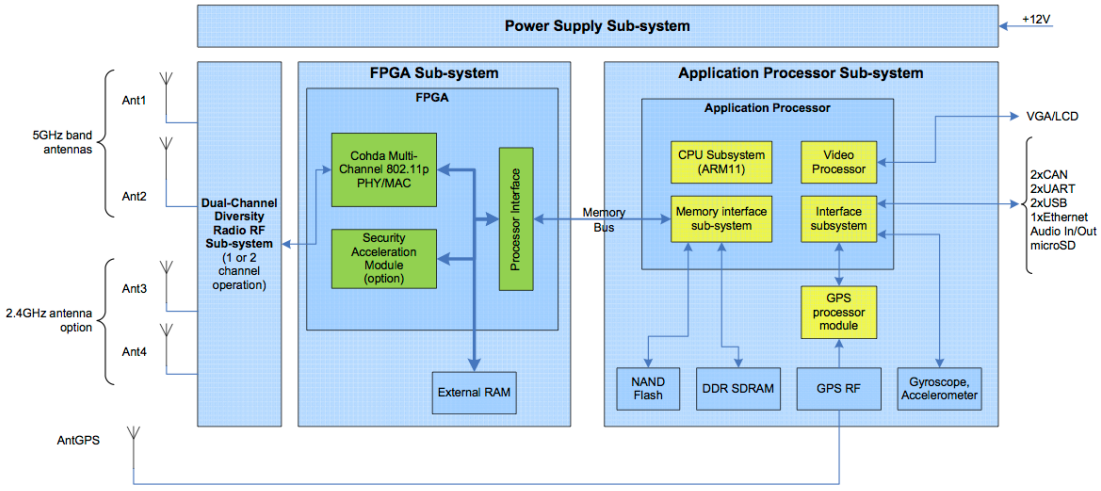

Abbildung 3.1: Hardwarearchitektur Cohda MK-2 [52]

FPGA und Radio RF Sub-system

Die Bitübertragungs- und Sicherungsschicht der Cohda Mobility Einheit entsprechen der 802.11p Spezifikation sowie der ETSI TC-ITS Spezifikation (Abschnitt 2.2.1.1 bzw. 2.2.2.1). Wie in Abbildung 3.1 zu sehen ist, verfügt die Plattform über vier Antennenbuchsen. Jeweils zwei Buchsen sind für den Betrieb im 2,4 GHz Bereich und für den Betrieb im 5,9 GHz Bereich vorgesehen. Bei den erworbenen Geräten sind für den Betrieb nach ETSI TC-ITS nur die beiden Antennen für den 5,9 GHz Bereich

aktiv. Die Unterstützung zweier Antennen für den 5,9 GHz Bereich ermöglicht eine sehr flexible Konfiguration der Kommunikationsmodi. Im Einkanal Modus erfolgt die Datenübertragung nur über einen aus den in Tabelle 2.2 spezifizierten Kanälen. Im Zweikanal Modus hingegen erfolgt die Kommunikation simultan auf zwei unterschiedlichen Kanälen. In [53] werden die Vorteile des verbesserten Wireless-Empfangs unter Verwendung von Antennen-Diversity beschrieben. Beide Kanalmodi können sowohl ohne Antennen-Diversity als auch mit Antennen-Diversity betrieben werden. Folglich ergeben sich vier Kommunikationsmodi:

i) Einkanal ohne Antennen-Diversity

ii) Einkanal mit Antennen-Diversity

iii) Zweikanal ohne Antennen-Diversity

iv) Zweikanal mit Antennen-Diversity

Die maximale Sendeleistung der Kommunikationsmodi ist auf +21dBm Equivalent Isotropically Radiated Power (EIRP) beschränkt.

Application Processor Sub-system

Das Herzstück des MK-2 Application Processor Sub-systems bildet der 533 MHz ARM11-Prozessor. Dem Prozessor stehen 64 MB Arbeitsspeicher zur Verfügung. Das Dateisystem befindet sich auf einem NAND Flash, wobei die Partition /mnt/ubi dem User eine Größe von 460 MB bietet. Zusätzlich besteht die Möglichkeit das Dateisystem über einen microSD-Kartenslot mit einer microSD-Karte auf maximal 4 GB zu erweitern. Die Zeit- und Standortbestimmung erfolgt über das GPS Processor Modul mit integriertem GPS-RF. Zur generellen Verbesserung der GPS-Standortbestimmung ist die Plattform mit einem Gyroskop ausgestattet. Zusätzlich kann bei schlechtem GPS-Empfang zur Verbesserung der Standortbestimmung Dead-Reckoning [6] verwendet werden. Hierbei werden auch Daten vom CAN-Bus beziehungsweise von den Digital Inputs in die Standortbestimmung miteinbezogen.

Hardware Schnittstellen

Zur Kommunikation mit externen peripheren Geräten, aber auch zur Interaktion mit dem Fahrer verfügt die MK-2 Plattform über folgende Hardwareschnittstellen:

- **Audio In/Out**
  Die Audio In/Out-Schnittstelle ist zur Verwendung von Mikrofon und Lautsprecher vorgesehen und über jeweils einen Stereo-Klinkenstecker ausgeführt. Diese Schnittstelle dient als HMI und kann beispielsweise dazu verwendet werden, den Fahrer akustisch zu warnen.

- **CAN**
  Über die CAN-Schnittstelle werden dynamische Fahrzeugdaten abgegriffen und können in weiterer Folge von ITS-Anwendungen verarbeitet werden. Die MK-2 Plattform verfügt über die Schnittstelle CAN0, welche eine Datenrate von bis zu 1Mbit/s unterstützt und eine weitere CAN1-Schnittstelle, welche die Datenrate von 125kBit/s verwendet. Dabei wird von beiden Schnittstellen sowohl der Standard CAN 2.0 A als auch der Standard CAN 2.0 B mit erweitertem Statusfeld unterstützt.
- **Digital Inputs**
  Die drei 12 Volt Digital Inputs sind universell einsetzbar. Ein besonderer Anwendungsfall der Digital Inputs ist die Verwendung für Dead-Reckoning. Dabei werden alternativ zum CAN-Signal digitale Signale wie Geschwindigkeit oder Lenkwinkel verwendet, um die Ortsbestimmung bei schlechtem GPS-Empfang zu verbessern.
- **Ethernet**
  Der 10BASE-T/100BASE-TX Ethernet-Schnittstelle sind zwei Hauptfunktionen zuzuordnen. Während der Entwicklungsphase von Programmen kann über diese Schnittstelle mit dem entsprechenden SDK Remote Debugging durchgeführt werden. Im Betrieb als OBU oder RSU stellt die Schnittstelle mit den dazugehörigen Linux Ethernet Treibern vollständige IPv4 und IPv6 Netzwerkkonektivität her.
- **RS232**
  Die RS232-Schnittstelle ermöglicht zum einen den Zugriff auf die MK-2 Plattform über die Konsole, zum anderen kann die universelle RS232-Schnittstelle verwendet werden, um beispielsweise einen externen GPS-Receiver zu verbinden.
- **USB**
  Die USB-Schnittstelle kann als USB 2.0 (480 MBit/s) oder als USB 1.1 (12 MBit/s) konfiguriert werden. Dies ermöglicht den Anschluss zahlreicher Peripheriegeräte wie eines USB-UMTS Modems, Keyboards oder Touch-Screen Controllers. Standardmäßig ist die USB-Schnittstelle als USB-Ethernet Device konfiguriert, sodass Remote-Debugging alternativ zur Ethernet-Schnittstelle über USB möglich ist.
- **VGA**
  Das Application Processer Sub-system verfügt über ein Bildverarbeitungsmodul, welches eine Auflösung von 800 x 600 Pixel und eine Farbtiefe von 15 Bit pro Pixel unterstützt. Die VGA-Schnittstelle kann zusammen mit einem Touch-Screen Controller als HMI verwendet werden.

3.3.2 Softwarearchitekur

Auf der MK-2 Plattform wird als Betriebssystem Linux 2.6 ausgeführt. Grundsätzlich werden bei Unix basierenden Betriebssystemen wie Linux zwei Bereiche unterschieden: Kernelspace und Userspace. Im Kernelspace wird der Betriebssystemkernel ausgeführt, der u.a. für Scheduling, Prozess- und Hardwareverwaltung verantwortlich ist und somit die Grundlage des Betriebssystems darstellt. Anwenderprogramme wie ITS-Anwendungen werden im Userspace ausgeführt. Über Systemcalls können Programme im Userspace auf bereitgestellte Funktionen des Betriebssystemkernels zugreifen. Die Ausführung der Systemcall-Funktionalität wird dadurch vom Userspace Programm an den Betriebssystemkernel übergeben. Abbildung 3.2 zeigt die Softwarearchitektur der MK-2 Plattform. Cohda unterteilt die Architektur in die Sub-systeme Platform-Services, HMI-Services, MobilityPHY and MobilityMAC, IEEE 1609 and ETSI TC-ITS sowie Positioning Services. Linux stellt Systemcalls APIs in Form von zahlreichen Bibliotheksfunktionen zur

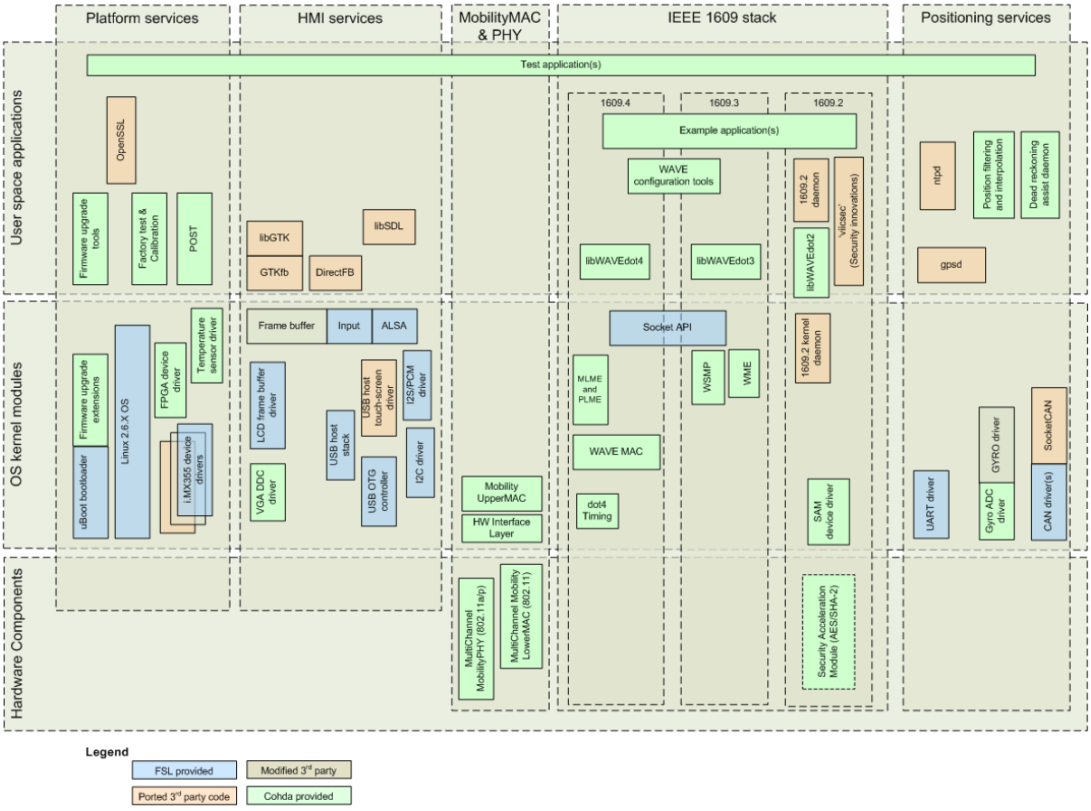

Abbildung 3.2: Softwarearchitektur Cohda MK-2 [52]

Verfügung. Eine besondere Form von APIs stellen Sockets dar. Über Sockets findet eine bidirektionale Kommunikation zwischen Prozessen des Userspace und des Kernelspace statt. Berkeley Sockets sind im Linux-Betriebssystem verankerte Netzwerksockets und sind

die Schnittstelle zwischen der Netzwerkprotokollimplementierung im Kernelsapce und der Anwendung im Userspace. Die APIs, welche für die Use-Case Implementation (Kapitel 4) relevant sind, werden im Abschnitt 4.4 detailliert beschrieben.

3.3.3 Entwicklungsumgebung

Die Entwicklungsumgebung der Cohda MK-2 Plattform befindet sich auf einer von Cohda vorbereiteten Virtuelle Maschine (VM). Auf dieser VM läuft das Betriebssystem Ubuntu 2.30.2. Die VM enthält neben den benötigten Bibliotheken das SDK Eclipse Ganymede. Die Anwendung wird auf Eclipse in der Programmiersprache C programmiert. Abbildung

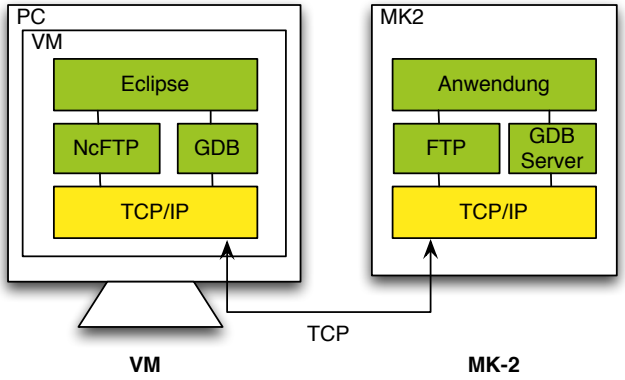

Abbildung 3.3: Entwicklungsumgebung bestehend aus PC und Embedded-Plattform

3.3 zeigt die Entwicklungsumgebung bestehend aus Entwicklungs-PC mit dazugehöriger VM sowie der MK-2 Plattform. Eine Anwendung wird auf der VM in Eclipse entwickelt. Von Eclipse aus erfolgt die Übertragung der entwickelten Anwendung via File Transfer Protocol (FTP) auf das MK-2 Zielsystem. Remote-Debugging wird über den GNU-Debugger GDB ermöglicht. Auf dem MK-2 System läuft dafür die GDB-Server Anwendung. Debug-Befehle werden von Eclipse aus an den GDB weitergegeben. Dieser kommuniziert über TCP/IP mit der GDB-Serveranwendung, welche die Debug-Befehle entgegennimmt und den Programmablauf steuert. Das Debugging der Anwendung auf der ARM-CPU (MK-2) funktioniert in gleicher Weise, als würde die Anwendung auf der CPU des Entwicklung-PC debugged werden.

3.3.4 Zusammenfassung

In diesem Abschnitt wurde die Hardware- und Softwarearchitektur der Cohda MK-2 Plattform näher gebracht. Weiters wurde die dazugehörige Entwicklungsumgebung beschrieben. Die Informationen wurden dem Cohda Wireless MK2 Wiki [52] entnommen.

Kapitel 4

Implementation

Dieses Kapitel behandelt die Integration der MK-2 Plattform in die E/E-Architektur eines Fahrzeugs. Ziel ist die Analyse der benötigten Hardware- und Softwareschnittstellen. Zur Analyse der notwendigen Schnittstellen wird auf der MK-2 Plattform eine ITS-Anwendung implementiert. Als ITS-Anwendung wird der Use-Case des elektronischen Bremslichts bei Notbremsung (Emergency Electronic Brake Light (EEBL)) implementiert. Die folgenden Abschnitte beschreiben zunächst den Use-Case, danach wird auf die Systemanwendungen der MK-2 Plattform eingegangen. Im Anschluss wird die Implementierung des Use-Case anhand von Ablaufdiagrammen erläutert. Abschnitt 4.4 analysiert die benötigten Hardware- und Softwareschnittstellen. Der letzte Abschnitt zeigt die Umsetzung der Implementation im Laboraufbau.

4.1 Use-Case

In Kapitel 2.2.2.4 wurden ITS-Anwendungen nach ETSI klassifiziert. Der Use-Case EEBL zählt zur Klasse der Road Safety Anwendungen. Ziel dieser ITS-Anwendung ist es, den Fahrer auch bei Sichtbehinderung (LKWs, Nebel) frühzeitig vor stark bremsenden Vorderfahrzeugen zu warnen um so Auffahrunfälle zu verhindern. Abbildung 4.1 zeigt schematisch die Komponenten des Use-Cases. Der Ablauf ist folgender: Bei plötzlicher und langanhaltender Verzögerung von Fahrzeug 1 durch Betätigung der Fußbremse legt das Steuergerät (STG) die CAN-Nachricht einer Notbremsung auf den CAN-Bus des Fahrzeuges. Dieses CAN-Signal ist bei aktuellen Fahrzeugen bereits vorhanden und dient zur Herbeiführung eines minimalen Bremswegs. Motor- und Bremssteuergerät agieren so, dass maximale Verzögerung erreicht wird. Die MK-2 Plattform registriert diese Nachricht am CAN-Bus und generiert dadurch eine DENM, welche über ITS-G5 versendet wird. Diese nach ETSI TS 102 637-3 [34] genormte Nachricht enthält wie in Abschnitt 2.2.2.3 beschrieben folgende für den Use-Case relevante Daten: Die aktuelle Position des Fahrzeuges, wel-

che über GPS ermittelt wird, die Fahrtrichtung und den Cause Code „Notbremsung“, der das DENM-Ereignis beschreibt. Der Folgeverkehr (Fahrzeug 2), ebenfalls mit der MK-2 Plattform ausgestattet, empfängt die DENM. Die ITS-Anwendung entscheidet aufgrund

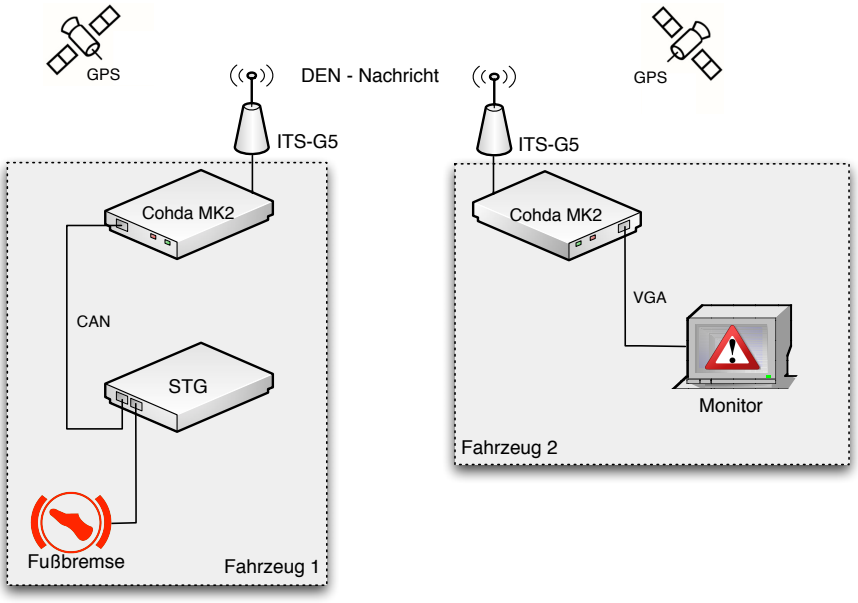

Abbildung 4.1: Use-Case elektronisches Bremslicht bei Notbremsung

der eigenen GPS-Position und der in der DENM enthaltenen GPS-Position, sowie des Zeitstempels ob die Information der Notbremsung für den Fahrer noch relevant ist. Sind GPS-Position und Zeitstempel innerhalb des definierten Gültigkeitsbereiches wird der Fahrer visuell über den Monitor gewarnt. Optional kann die Warnung des Fahrers auch über die Headunit des Fahrzeugs erfolgen, indem die CAN- oder Ethernetschnittstelle der MK-2 Plattform verwendet wird.

4.2 Systemanwendungen

Die MK-2 Plattform bietet zum Erstellen und Ausführen von Anwendungen mehrere Modi, wodurch es möglich ist, entwickelte Anwendungen direkt auf der MK-2 Hardware (Modus A) ausschließlich in der VM (Modus B) oder in der Kombination von VM und MK-2 (Modus C) auszuführen. Im Folgenden sind die Funktionsweisen der drei Ausführungsmodi beschrieben. Eine ITS-Anwendung, die Informationen auf Basis von CAM oder DENM erhält beziehungsweise zur Verfügung stellt, benötigt Schnittstellen um auf GPS- und ITS-G5 Daten Zugriff zu haben. Die Schnittstellen in Richtung Hardware werden in allen Ausführungsmodi durch folgende zwei Anwendungen realisiert:

- **ETSA**: Die ETS-Anwendung (ETSA) ist die von Cohda verfügbare C-ITS Protokollstack-Anwendung. ETSA ist die Schnittstelle zwischen ITS-Anwendung und dem ITS-G5 Transceiver.
- **GPSD**: Die GPSD-Anwendung ist ein Hintergrunddienst, welcher GPS-Daten vom GPS-Receiver anderen Anwendungen in Form von UDP-Datenpakete zur Verfügung stellt.

Auf die detaillierte Funktionalität der Schnittstellen wird in Abschnitt 4.4 eingegangen. Die beiden Schnittstellen sind entscheidend für die Ausführungsmodi der zu implementierenden ITS-Anwendung. Durch die Hardware-Emulationsumgebung der VM ist es möglich, Softwareentwicklung auch ohne Vorhandensein der MK-2 Plattform durchzuführen. Das Makefile kompiliert aus dem Quellcode der ITS-Anwendung ausführbare Dateien für die x86-Architektur der VM sowie für die ARM-Architektur der MK-2 Plattform.

Ausführungsmodus A

Für Ausführungsmodus A wurde die ITS-Anwendung wie in 3.3.3 beschrieben, zuerst über Make für die ARM11-Architektur kompiliert und über FTP auf beide MK-2 Plattformen transferiert. Abbildung 4.3 zeigt beide Plattformen MK-2-A und MK-2-B im Ausführungs-

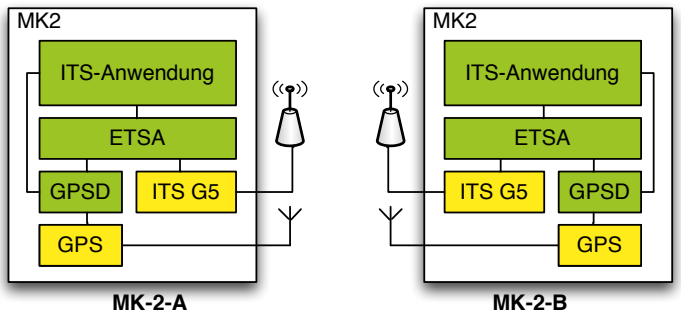

Abbildung 4.2: Modus A: ITS-Anwendung und ETSA MK-2 (ARM11)

modus A. Wie zu sehen ist, werden ITS-Anwendung und ETSA auf der ARM-11 Architektur der MK-2 Plattform ausgeführt. Beide ITS-Anwendungen kommunizieren über die Cohda ITS-G5 Hardware miteinander. Zusätzlich werden die GPS-Daten vom eingebauten GPS-Receiver empfangen. Ausführungsmodus A eignet sich durch den reinen Embedded-Einsatz für Anwendungen, die bereits fertig entwickelt worden sind.

Ausführungsmodus B

Im Modus B laufen alle Anwendungen auf der x86-Architektur des Entwicklungs-PCs. Durch die Hardware-Emulationsumgebung der VM können Anwendungen auch ohne Vor-

handensein einer Hardware entwickelt und getestet werden. Abbildung 4.3 zeigt den Ausführungsmodus B. Wie zu sehen ist, wird die ITS-G5 Hardware der MK-2 Plattform durch eine Dummy-ITS-G5 Hardware simuliert. Drahtlose ITS-G5 Kommunikation zwischen VM-A und VM-B wird durch TCP/IP-Kommunikation ersetzt. Somit können CAM

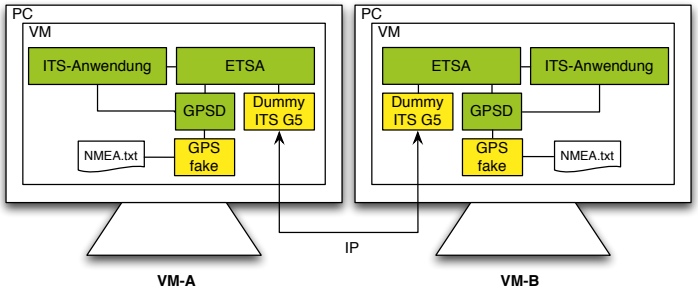

Abbildung 4.3: Modus B: ITS-Anwendung und ETSA auf VM (x86)

und DENM zwischen den VMs ausgetauscht werden. Bei GPSfake handelt es sich um eine Testanwendung die aus einem GPS-Logfile (NMEA.txt) GPS-Koordinaten ausliest und GPSD über eine serielle Pseudoschnittstelle zur Verfügung stellt. Somit wird zur Simulation der reale GPS-Receiver durch GPSfake ersetzt. Erstellte ITS-Anwendungen, die im Modus B ausgeführt wurden, können nach dem Kompilieren für die ARM11-Architektur ohne weitere Änderungen auf der MK-2 Plattform ausgeführt werden. Modus B dient daher zur hardwarelosen Anwendungsentwicklung.

Ausführungsmodus C

Modus C ist eine Kombination der Modi A und B. Wie in Abbildung 4.4 zu sehen ist, wird die ITS-Anwendung in der VM ausgeführt, ETSA und GPSD laufen auf der MK-2 Plattform. Die ITS-Anwendung sendet und empfängt CAM- und DENM-Daten, indem sie

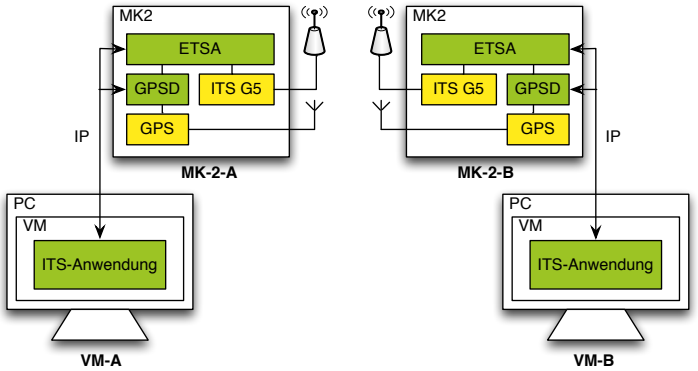

Abbildung 4.4: Modus C: ITS-Anwendung auf VM (x86) und ETSA auf MK-2 (ARM11)

mit der ETSA über IP Daten austauscht. Über die ITS-G5 Hardware der MK-2 Plattform leitet die ETSA die Daten weiter. Modus C erlaubt Anwendungsentwicklung in der VM, bei gleichzeitiger Möglichkeit der Ausführung auf der Zielhardware.

4.3 ITS-Anwendung

Als ITS-Anwendung wurde der Use-Case EEBL im Rahmen dieser Masterarbeit umgesetzt. Die Implementierung erfolgte unter Verwendung der Eclipse Entwicklungsumgebung. Abbildung 4.5 zeigt die Entwicklungsumgebung mit dem Hauptprogramm *ets − shell.c*. Bei der Implementierung wurde auf dem bereits von Cohda vorhandenem Framework aufgebaut.

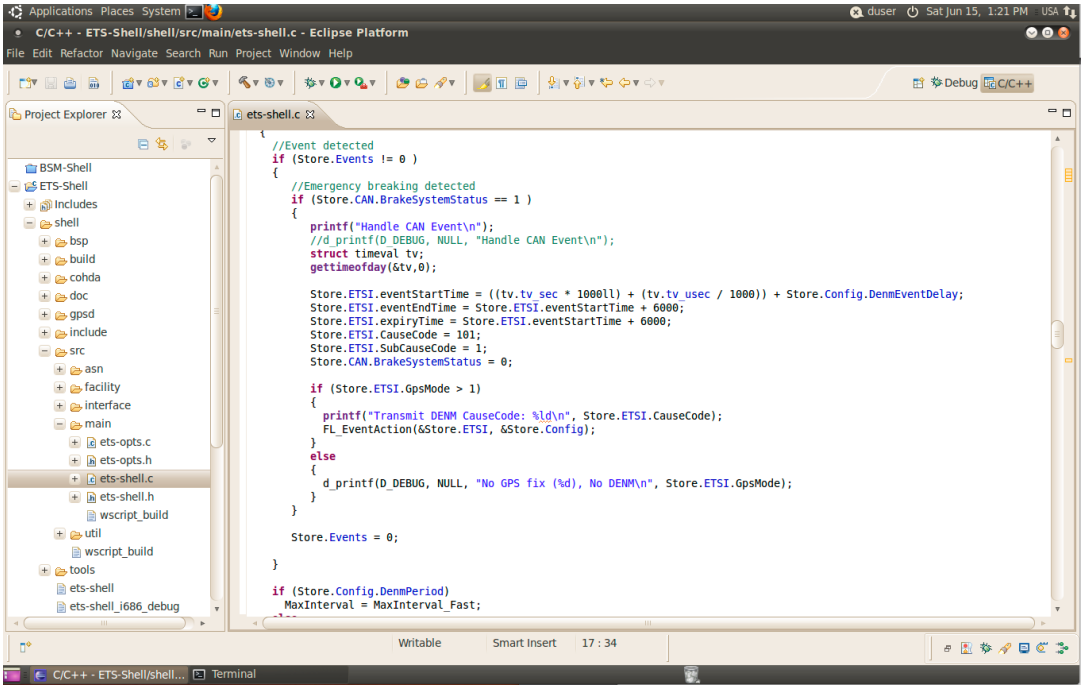

Abbildung 4.5: Entwicklungsumgebung Eclipse

Im Folgendem wird nun das Programm der ITS-Anwendung für den Use-Case EEBL anhand von Ablaufdiagrammen erläutert:

Abbildung 4.6 zeigt das Ablaufdiagramm des Hauptprogramms. Beim Starten der Anwendung wird zuerst die CAN-, GPS- und UDP-Schnittstelle initialisiert. Auf die Schnittstellen kann nach erfolgter Initialisierung über einen File-Descriptor zugegriffen werden. Details zu den Schnittstellen finden sich unter Abschnitt 4.4. Die Schnittstellen werden über Polling solange abgefragt, bis auf einer Schnittstelle Daten anliegen. In diesem Fall wird das Polling sofort verlassen und die Verarbeitung der Daten weitergeführt. Nach erfolgter Da-

tenverarbeitung wird die Variable *TimeOut* um die benötigte Zykluszeit (Schleifendurchlaufzeit) verringert. Die Variable *TimeOut* dient der regelmäßigen CAM-Generierung, und entspricht zu Beginn des Programmablaufs dem CAM-Nachrichtenintervall. *TimeOut* beschränkt zudem auch das maximale Pollingintervall. Durch Verringern von *TimeOut* nach jedem Zyklus wird respektive auch das Pollingintervall kleiner, wodurch es irgendwann so klein ist, dass während des kurzen Pollingintervalls keine Daten mehr an einem der

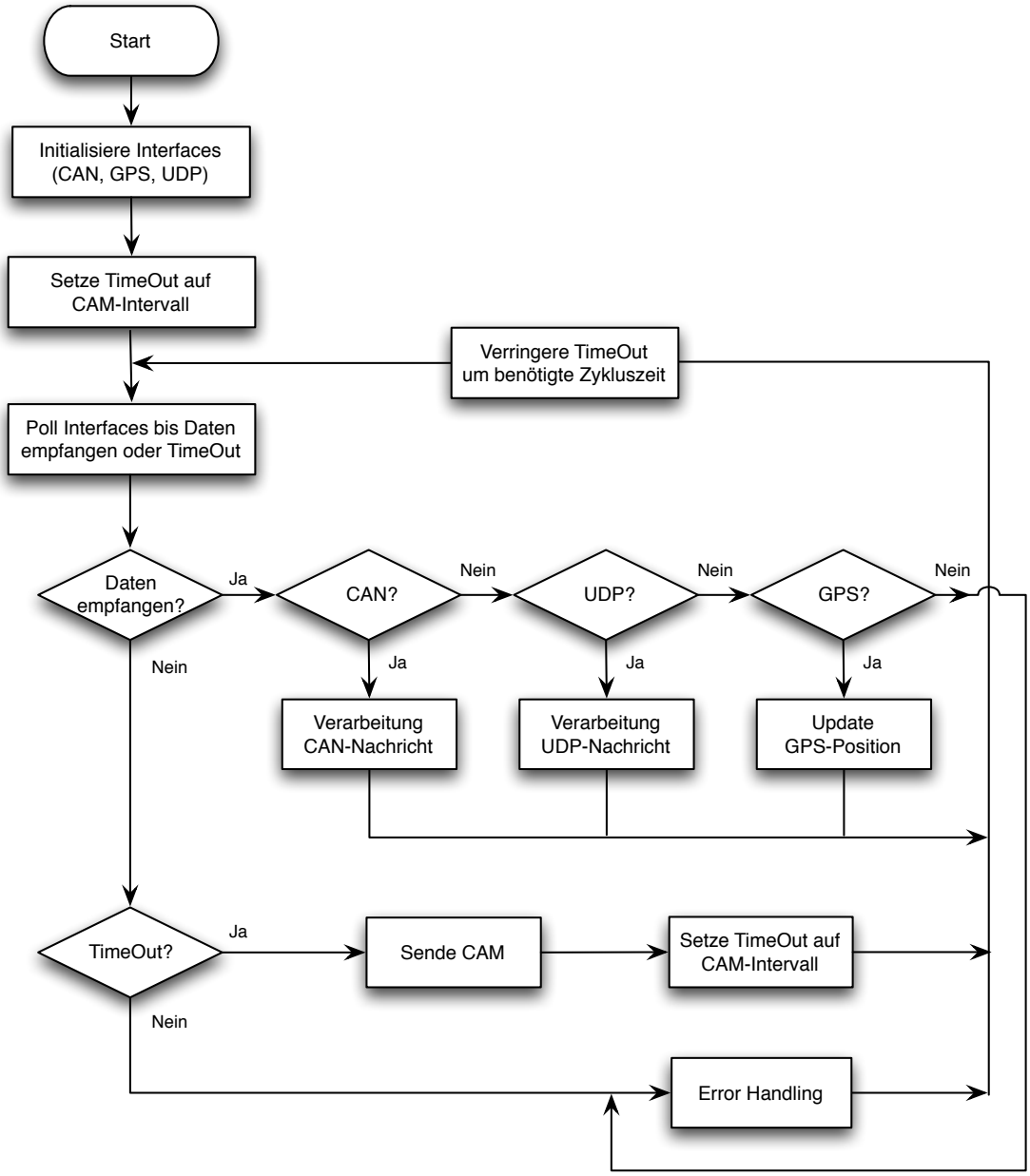

Abbildung 4.6: Ablaufdiagramm Hauptprogramm

File-Descriptoren anliegen und es zum $TimeOut$ kommt. Beim $TimeOut$ wird eine CAM generiert und danach $TimeOut$ wieder auf den Wert des CAM-Nachrichtenintervalls gesetzt. Mit anderen Worten zählt $TimeOut$ die Zeitdauer seit der letzten CAM-Generierung. Innerhalb der Zeitdauer von $TimeOut$ werden über Polling Daten empfangen und verarbeitet.

Liegen im Hauptprogramm beim Polling Daten der CAN-Schnittstelle an, wird im weiteren Programmablauf die CAN-Verarbeitung aufgerufen. Das Ablaufdiagramm der CAN-Verarbeitung ist in Abbildung 4.7 zu sehen. Zu Beginn wird der CAN-Identifier überprüft.

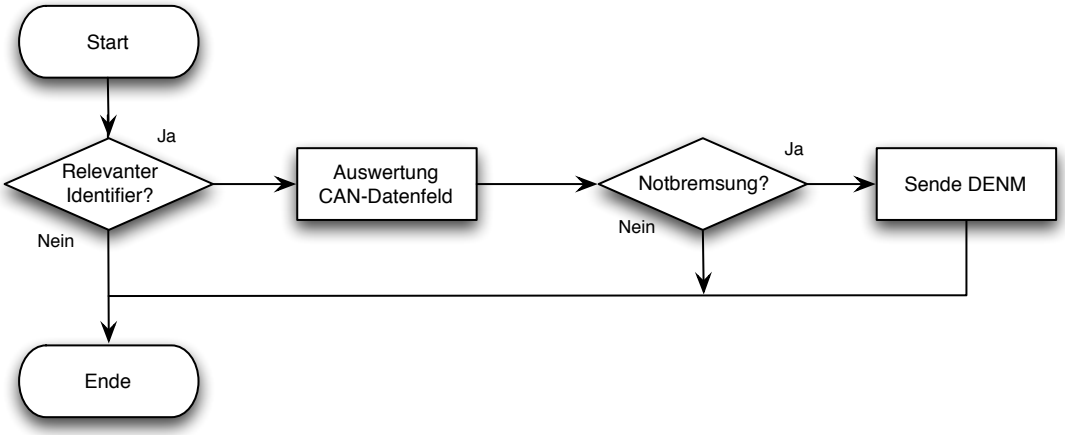

Abbildung 4.7: Ablaufdiagramm CAN-Verarbeitung

Entspricht dieser der CAN-Bremsnachricht wird das CAN-Datenfeld ausgewertet. Ist der CAN-Identifier unbekannt, wird die Nachricht verworfen. Falls das Datenfeld die Botschaft der Notbremsung enthält, wird die DENM mit der Botschaft EEBL generiert.

Handelt es sich beim Polling um UDP-Daten wird die UDP-Verarbeitung aufgerufen. In der UDP-Nachricht befindet sich der BTP-Header. Über die Port-Nummer des BTP-Headers kann die empfangene Nachricht als CAM oder DENM identifiziert werden. Die

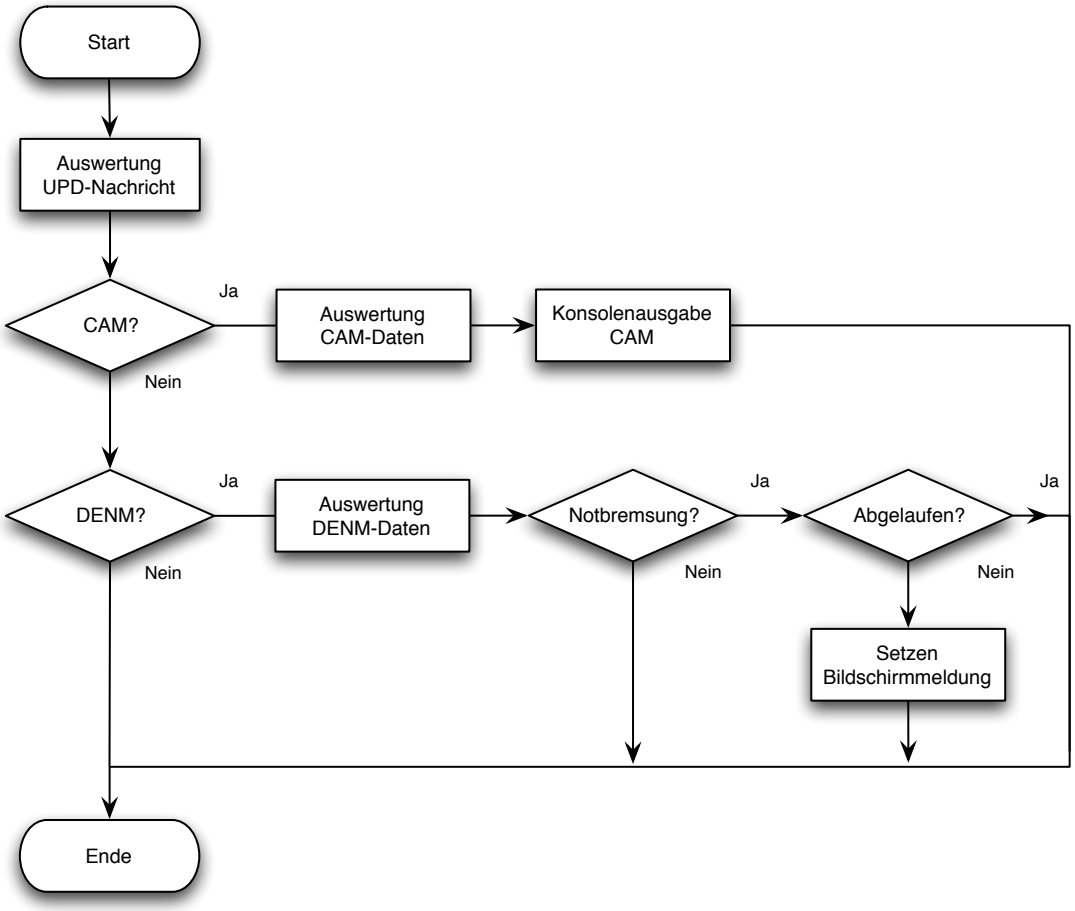

Abbildung 4.8: Ablaufdiagramm UDP-Verarbeitung

CAM-Daten haben keine Auswirkung auf die Funktionalität des EEBL Use-Cases, sie werden lediglich über die Konsole eines angeschlossenen Rechners ausgegeben. Bei der DENM Auswertung wird überprüft ob der DENM-*CauseCode* (siehe 2.2.2.3) dem Notbremsung-Code entspricht. Ist dies der Fall wird auf dem Bildschirm eine Warnung angezeigt, um den Fahrer zu informieren. Die DENM-*Duration* gibt an wielang die DENM relevant ist. Die Warnung am Bildschirm bleibt entsprechend der DENM-*Duration* aktiv. Nachdem diese Dauer vorüber ist, wechselt die Bildschirmanzeige wieder in Standard-Bildanzeige.

Abbildung 4.9 zeigt links das Ablaufdiagramm zum Versenden einer CAM. Auf der rechten Seite ist das Versenden einer DENM zu sehen. Beim Senden der jeweiligen Nachricht werden anhand des GPS-Signals die Ortskoordinaten sowie die aktuelle Zeit für den Zeitstempel in der Nachricht gespeichert. Für die DENM wird zusätzlich der Grund

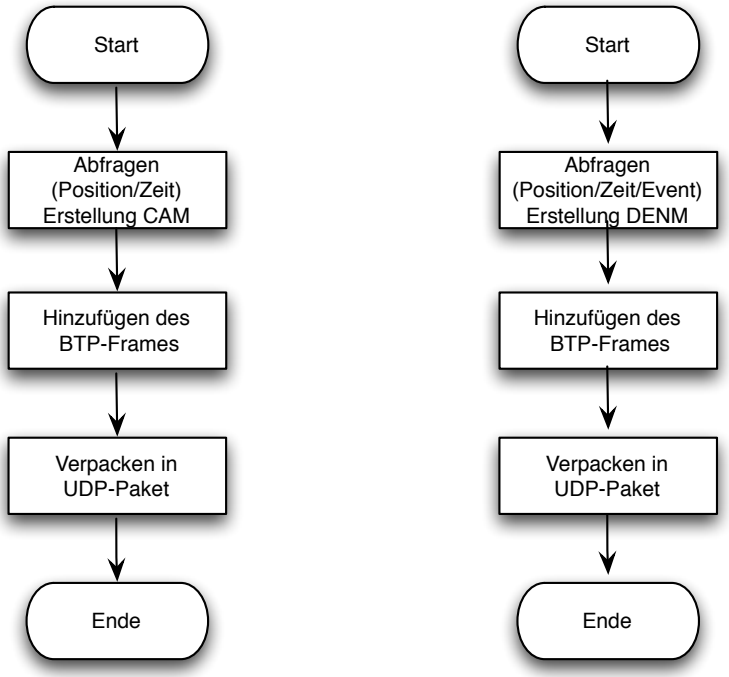

Abbildung 4.9: Ablaufdiagramm Senden CAM (links) und Senden DENM (rechts)

(Event) der DENM-Generierung abgefragt, um den entsprechenden *CauseCode* laut ETSI TS 102 637-3 [34] in der Nachricht zu hinterlegen. Weitere Parameter der CAM beziehungsweise der DENM werden entsprechend der Struktur in Abschnitt 2.2.2.3 gesetzt. Für den Laboraufbau (Abschnitt 4.5) wurden die fahrzeugspezifischen Parameter wie beispielsweise Fahrzeugabmessungen oder Fahrzeugtyp mit Dummy-Daten ersetzt. Für eine CAM beinhaltet der BTP-Frame die Portnummer 2001, für eine DENM die Portnummer 2002. Die Nachricht bestehend aus BTP-Frame und CAM- beziehungsweise DENM-Frame werden in ein UDP-Paket verpackt und an die ETSA (siehe Abschnitt 4.4) übermittelt.

4.4 Schnittstellen

Wie bereits in Abschnitt 3.3.2 beschrieben, greifen Userspace-Anwendungen über bereitgestellte APIs auf Funktionalitäten des Betriebssystemkernels zu. Zu den bereitgestellten Funktionalitäten zählen unter anderem der Zugriff auf die MK-2 Hardwareschnittstellen. Abbildung 4.10 zeigt die hardwarerelevanten APIs, welche für den Use-Case EEBL von Bedeutung sind. Wie zu sehen ist benötigt die EEBL-Anwendung Zugriff auf vier Hardwa-

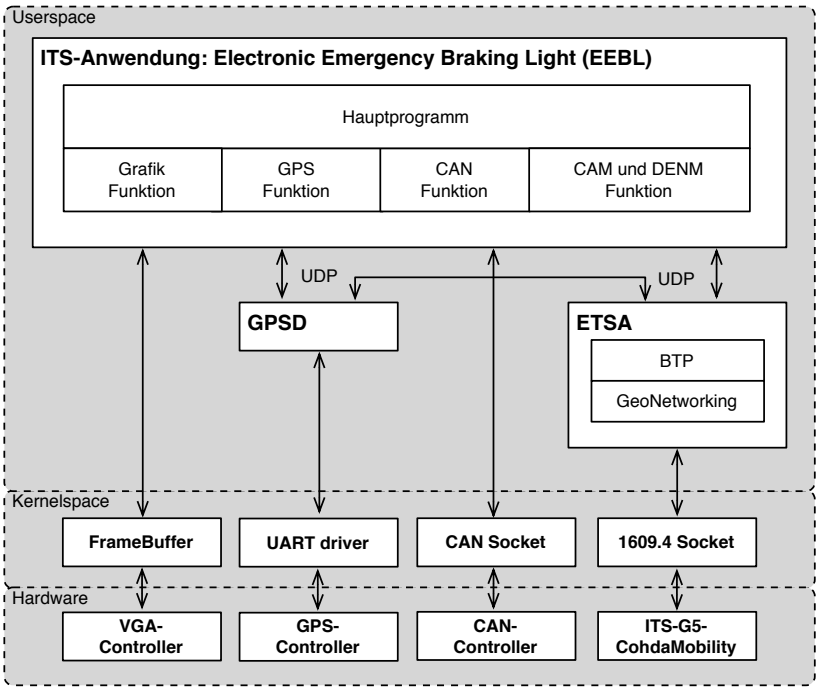

Abbildung 4.10: Relevante Schnittstellen des Use-Case

rekomponenten. Die Grafik-Funktion stellt über den FrameBuffer Bilddateien am VGA-Controller zur Verfügung, um im Fall der EEBL-DENM eine Warnung am Bildschirm anzuzeigen. Die GPS-Funktionalität wird verwendet, um die aktuellen Positionsdaten über die DENM und CAM zu versenden. Dies geschieht über den GPSD-Hintergrunddienst, welcher über die UART-Schnittstelle auf den GPS-Controller zugreift. Die ETSA-Protokollstack Anwendung empfängt ebenfalls UDP-Nachrichten vom GPSD, um den geographischen Gültigkeitsbereich des GeoNetworking Protokolls (Abschnitt 2.2.2.2) errechnen zu können. Der CAN-Controller Zugriff ist notwendig, um die CAN-Nachricht der Notbremsung zu empfangen und in weiterer Folge die EEBL-DENM zu generieren. Die Kommunikation zwischen ITS-Anwendung und CohdaMobility-Einheit stellt die Basis der ITS-G5 Kommunikation dar. Sie erfolgt wie bereits erwähnt über die ETSA. Die folgenden Punkte beschreiben die Schnittstellen im Detail.

- **CAN-Socket**
  CAN-Socket ist eine Erweiterung der Berkeley Sockets für Linux. Die Kommunikation mit dem CAN-Bus erfolgt vom Userspace aus wie eine Kommunikation mit dem Socket für das Internetprotokoll PF_INET. Durch read- und write-Befehle wird auf den CAN-Netzwerkadapter zugegriffen. CAN-Socket stellt dafür im Kernelspace Treiber für unterschiedliche CAN-Controller zur Verfügung. Die CAN-Protokollfamilie PF_CAN bietet darüber hinaus verschiedene Socket-Typen für Paketfilterung (BCM) oder ungefilterte Weiterleitung (RAW). Anstelle des CAN-Sockets kann der CAN-Zugriff auch direkt über den CAN-Treiber (Controller) erfolgen. Dabei wird der CAN-Treiber als Zeichengerät realisiert, wobei der CAN-Zugriff dem Zugriff auf eine serielle Schnittstelle ähnelt. Abbildung 4.11 zeigt den Vergleich zwischen Zeichentreiber und Socket. Die Realisierung über Sockets anstelle eines Zei-

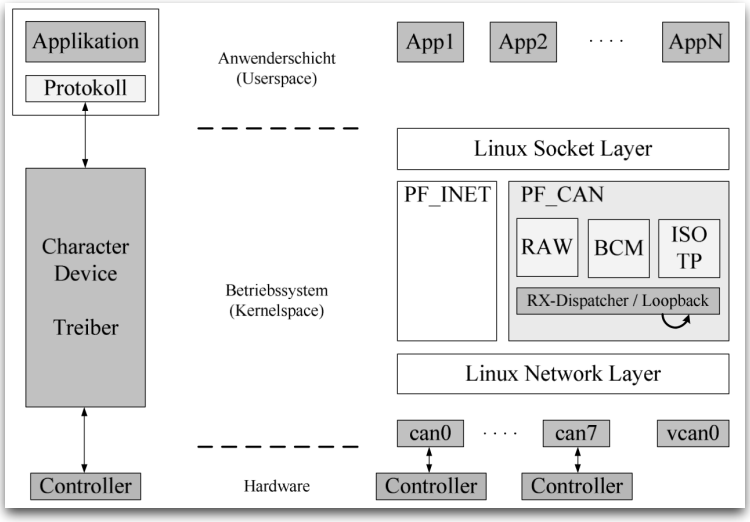

Abbildung 4.11: CAN-Zugriff realisiert als Zeichentreiber und Socket [54]

chengeräts hat deutliche Vorteile: Es ist mehreren Anwendungen gleichzeitig möglich, auf mehrere CAN-Busse lesend und schreibend zuzugreifen. Auch eine Echtzeitfähigkeit des Betriebssystems muss nicht gegeben sein. Funktionalität wie das Puffern von Botschaften und Vergabe von Zeitstempeln sind bereits in der Linux Netzwerkschicht vorhanden und müssen daher nicht neu implementiert werden. Darüber hinaus gibt es keine Probleme mit systembedingten Abhängigkeiten zwischen einzelnen CAN-Anwendungen wie es alternativ mit einem Zeichengerät der Fall wäre. Dazu registrieren sich Anwendungen für bestimmte CAN-IDs beim PF_CAN Modul. Der RX-Dispatcher sendet über ein Loopback die empfangen Nachrichten an jene Anwendungen die sich vorher für die CAN-ID registriert haben [54].

- **ETSA**
  Die ETS-Anwendung (ETSA) ist die von Cohda verfügbare C-ITS Protokollstack Implementierung. Die Nutzung der ETSA ist für Testzwecke kostenfrei. Soll die ETSA jedoch in kommerziellen Systemen Anwendung finden, muss die Nutzungslizenz separat erworben werden. Wie in Abbildung 4.10 ersichtlich, ist die ETSA die Schnittstelle zwischen ITS-Anwendung und dem ITS-G5 CohdaMobility-Transceiver. Eine CAM oder DENM die von der ITS-Anwendung generiert wurde, wird in ein UDP-Paket verpackt und an die UDP-Adresse der ETSA mit der BTP Port-Nummer gesendet. Die ETSA fügt den GeoNetworking-Header an und gibt die Nachricht über den 1609.4 Socket an die Linux-Netzwerkschicht weiter. Die Abfolge beim Empfang von ITS-G5 Nachrichten funktioniert in umgekehrter Reihenfolge, wobei die ETSA die entpackten Daten an die UDP-Adresse und Port-Nummer der ITS-Anwendung weiterleitet. Grundsätzlich wird die ETSA über eine Konfigurationsdatei konfiguriert, wodurch UDP-Adressen und GeoNetworking-Verhalten eingestellt werden.

- **1609.4 Socket**
  Die 1609.4 Socket Funktionalität umfasst die Verarbeitung der ITS-G5 Frames. Dazu zählt zum einen die Wahl des Kanals - Steuerkanal (G5CC) oder Servicekanal (G5SC1 - G5SC5) - zum anderen das Routing und Multiplexing der Pakete anhand des verwendeten Protokolls. Der 1609.4 Socket stellt der Anwendungsschicht ITS-G5 Funktionalität über einen Netzwerkadapter zur Verfügung. Im Userspace wird die Schnittstelle wie beim CAN-Socket über den Netzwerkadapter durch read- und write-Befehle angesprochen. CAMs, DENMs aber auch IPv4/IPv6 Pakete, die über ITS-G5 ausgetauscht werden sollen, werden über den bereitgestellten Netzwerkadapter geroutet. Wie in Abbildung 4.10 zu sehen ist, wird aus der ITS-Anwendungssicht nicht direkt auf den 1609.4 Socket zugegriffen. Der Zugriff erfolgt indirekt über die von Cohda bereitgestellte C-ITS Protokollstackimplementierung ETSA.

- **DirectFB**
  Linux stellt grundsätzlich eine große Anzahl von Grafiklibraries zur Verfügung. Das Linux Betriebssystem der Cohda MK-2 Plattform unterstützt standardmäßig die Libraries libGTK, libSDL, GTKfb und DirectFB, die je nach Anwendungsfall genutzt werden können. Für die ITS-Anwendung EEBL muss vom GUI keine Funktionalität für User-Inputs vorhandne sein. Zur reinen Output-Visualisierung eignet sich daher DirectFB. DirectFB ermöglicht es, den Framebuffer der VGA-Schnittstelle als Device im Userspace anzusprechen. Die zu verwendenden Datenstrukturen sind in der Library „fb.h“ definiert. Um Bilder am VGA-Ausgang über DirectFB anzuzeigen, wird auf den Framebuffer mit open- und close-Befehlen zugegriffen, wobei jedem einzelnen Pixel ein 15-Bit RGB Farbwert zugewiesen wird. Als Bildvorlage

bietet sich daher das Bitmap-Format (.bmp) an, da pro Pixel die Farbinformation als RGB-Farbwert gespeichert ist.

- **GPSD**
  GPSD-Server ist ein Hintergrunddienst der im Userspace läuft und die GPS-Signale vom integrierten GPS-Receiver auswertet. Die GPS-Daten werden von der ITS-Anwendung und der ETSA über den UDP-Socket Port 2947 abgefragt.

4.5 Laboraufbau der Implementation

Die ITS-Anwendung mit der Use-Case Implementierung EEBL wurde wie in Abschnitt 3.3.3 beschrieben programmiert und auf die Embedded-MK-2 Plattform übertragen. Da zu Beginn der Implementierungsphase nicht beide Hardwareplattformen verfügbar waren, wurde zuerst die ITS-Anwendung in Ausführungsmodus B (Abschnitt 4.2) simuliert. Nachdem beide Plattformen verfügbar waren, konnte der Laboraufbau in Ausführungsmodus C aufgebaut werden. Abbildung 4.12 zeigt den Laboraufbau in Ausführungsmodus C. Der

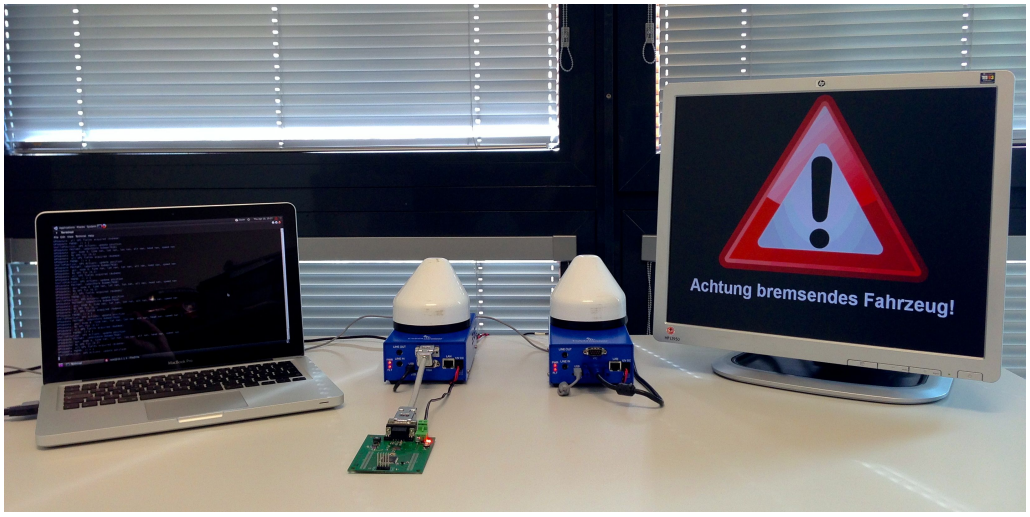

Abbildung 4.12: Laboraufbau der Implementation

Laptop dient zur Steuerung und Kontrolle des Programmablaufs. Die linke Cohda-Einheit ist über den CAN-Bus mit einer CAN-Testplatine verbunden. Die CAN-Testplatine simuliert im Use-Case (Abbildung 4.1) das Steuergerät (STG) von Fahrzeug 1, welches die Nachricht einer Notbremsung auf den CAN-Bus legt. Die rechte Cohda-Einheit simuliert das Fahrzeug 2 im Folgeverkehr, welches die DENM der Notbremsung empfängt und die Nachricht am Bildschirm ausgibt. Die Ausführung des Laboraufbaus läuft wie folgt: Wird die Spannungsversorgung der Cohda-Plattformen hergestellt, dauert es zunächst 60 Sekunden bis das Linux-Betriebssystem hochfährt und einsatzbereit ist. Der Telnet-Dienst

ermöglicht vom Laptop aus Zugang zur Linux-Konsole der Cohda-Plattformen. Über die Linux-Konsole werden ETSA und die implementierte ITS-Anwendung gestartet. Auf der linken Plattform muss zuvor noch der CAN-Dienst über ein Skript gestartet werden, damit der CAN-Socket verfügbar ist. Wenn sich beide Plattformen über Beaconing gegenseitig registriert haben, beginnt die CAM Generierung im Intervall von 2 Sekunden. Das CAM-Intervall ist abhängig vom CAM Use-Case, welche in [33] definiert sind. Wird auf der CAN-Testplatine der Taster betätigt, wird die CAN-Nachricht einer Notbremsung auf den CAN-Bus gelegt. Die linke MK2-Plattform empfängt die CAN-Nachricht und generiert eine DENM mit dem DENM-*CauseCode* EEBL. Die DENM wird von der rechten MK-2 Plattform empfangen. Handelt es sich um den EEBL-*CauseCode* wird die Warnung am Bildschirm angezeigt. Nach Ablauf der Gültigkeitsdauer wechselt der Bildschirm wieder in den Standard-Anzeigemodus.

4.6 Zusammenfassung

Eine Analyse der notwendigen Hardware- und Softwareschnittstellen zur Integration der Cohda-Plattform in die E/E-Architektur eines Fahrzeugs wurde durch die Use-Case Implementierung EEBL umgesetzt. Bei diesem Use-Case wird die DENM einer Notbremsung von einem Fahrzeug über ITS-G5 ausgesendet. Das nachfolgende Fahrzeug empfängt die Nachricht und informiert den Fahrer über eine visuelle Meldung am Monitor. Während der Entwicklungsphase der ITS-Anwendung können unterschiedliche Ausführungsmodi verwendet werden. So kann die ITS-Anwendung in der VM, auf der MK-2 Hardwareplattform oder im Kombinationsmodus ausgeführt werden. Die EBBL-Anwendung greift über die vom Betriebssystem bereitgestellten APIs auf die Hardwareressourcen der MK-2 Plattform zu. Folgende APIs finden in der Implementierung Anwendung: Zur Detektion der Notbremsung wertet die ITS-Anwendung die CAN-Bus Nachrichten des Fahrzeugs über den CAN-Socket aus. Das Versenden und Empfangen einer CAM beziehungsweise DENM erfolgt über die Protokollstack-Implementierung ETSA, welche über den 1609.4 Socket auf die Transceivereinheit der MK-2 Plattform zugreift. GPS-Signale zur Standortbestimmung werden vom internen GPS-Receiver durch den Hintergrunddienst GPSD per UDP an die ITS- und ETS-Anwendung weitergeleitet. Wird die DENM einer Notbremsung empfangen, erfolgt die Warnung visuell am Monitor. Um am VGA-Ausgang ein Bild zu visualisieren, wird über den FrameBuffer auf den VGA-Controller zugegriffen. Für den Laboraufbau wurde der CAN-Fahrzeugbus durch eine CAN-Testplatine simuliert, welche das CAN-Notbremssignal des Fahrzeugs generiert.

Kapitel 5

Feldtest

Ziel des Feldtests ist es, die Reichweite der Car-to-X Kommunikation unter Einsatz der Cohda MK-2 Plattform zu analysieren. Dieses Kapitel beschreibt zu Beginn die Rahmenbedingungen zur Durchführung des Feldtests (Abschnitt 5.1), den Ablauf des Testszenarios (Abschnitt 5.2) sowie die Ergebnisse des Feldtests (Abschnitt 5.3).

5.1 Rahmenbedingungen des Tests

Der Feldtest erfolgt am Campusgelände Inffeldgasse und findet zwischen zwei stehenden Fahrzeugen statt. Beide Fahrzeuge sind mit der Cohda MK-2 Plattform ausgestattet. Fahrzeug 1 befindet sich an einem fixen Standort. Fahrzeug 2 befindet sich an veränderlichen Standorten. Pro Standort/Messpunkt verändert sich die Distanz zwischen den Fahrzeugen. Die Übertragungsreichweite wird anhand der Packet Error Rate (PER) ermittelt. Die PER beschreibt in der Drahtloskommunikation die Qualität des Drahtlosnetzwerkes und ist definiert durch die Anzahl der fehlerhaft empfangenen Datenpakete dividiert durch die Anzahl aller gesendeten Datenpakete. Der Feldtest finden in zwei unterschiedlichen Szenarien statt:

- **Testszenario A**: Sichtverbindung Line-of Sight (LOS)
- **Testszenario B**: keine Sichtverbindung Non-line-of Sight (NLOS)

Beide Testszenarien werden unter folgenden technischen Rahmenbedingungen durchgeführt:

Cohda MK-2

Die CohdaMobility-Einheit der MK-2 Plattform erlaubt wie in Abschnitt 3.3.1 beschrieben, vier unterschiedliche Betriebsmodi. Die Durchführung des Feldtests erfolgt unter folgenden Einstellungen: Aus den Betriebsmodi wird für den Feldtest der

Einkanal-Modus gewählt, da die PER eines Übertragungskanals gemessen wird. Die Übertragung der Testdaten erfolgt zum Vergleich einmal ohne Antennen-Diversity und einmal mit Antennen-Diversity. Aus den nach ETSI ES 202 663 [28] spezifizierten Kanälen (Abschnitt 2.2.2.1, Tabelle 2.2) werden die Kanäle 174 und 178 im Feldtest getestet. Die beiden Kanäle unterscheiden sich durch die Datenrate von 6 Mbit/s beziehungsweise 12 Mbit/s. Der Feldtest für die Kanäle 180 und 176 ist aufgrund der spezifizierten Sendeleistung von 33 dBm EIRP nicht möglich, denn wie in Abschnitt 3.3.1 beschrieben, beträgt die maximale Sendeleistung der MK-2 Plattform +21 dBm EIRP.

Antenne

Als Antenneneinheit wird die mitgelieferte Antenne SMW-303 von Mobile Mark verwendet. Die Antenneneinheit integriert zwei 5 GHz Rundstrahlantennen zur ITS-G5 Kommunikation sowie eine 1575,42 MHz GPS-Antenne. Die beiden 5 GHz Antennen ermöglichen laut Herstellerspezifikation einen Antennengewinn von 5 dBi. Abbildung 5.1 zeigt die Abstrahlcharakteristik der Antenne in Richtung der Höhe links und in azimutaler Richtung rechts. Wie zu sehen ist besitzt die Antenne eine gleichmäßige Abstrahlcharakteristik in azimutaler Richtung. Während des Feldtests ist die magnetische Antenne zentriert am Fahrzeugdach angebracht und über ein 4,5 m langes RF-195 Koaxialkabel mit der MK-2 Plattform verbunden.

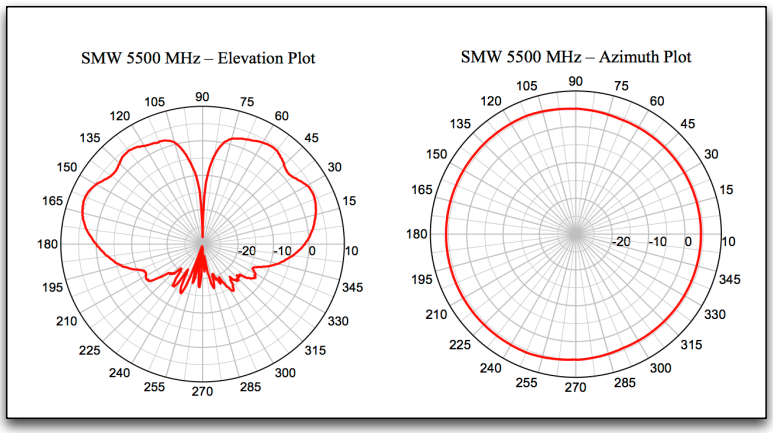

Abbildung 5.1: Abstrahlcharakteristik der Antenne [55]

Testanwendung

Die Anwendung zum Senden der Testpakete (TX-App) generiert beim Ausführen 140 Testpakete pro Sekunde und stoppt nach 10 Sekunden, was einer Übertragung von insgesamt 1400 Testpaketen pro Messung entspricht. Jedes übertragene Testpaket besteht aus einem 802.11 Header, dem Subnetwork Access Protocol (SNAP), den

Nutzdaten und der Frame Check Sequence (FCS)-Prüfzeichenfolge. Die TX-App generiert Testpakete, wobei die Nutzdaten mit Zufallsbytes gefüllt werden. Dabei variiert die TX-App die Paketgröße während des Testablaufs, um etwaige Schwankungen der PER in Bezug auf die Paketgröße zu analysieren. Die unterschiedlichen Testpakete werden mit einer Rate P_{rate} von 20 Paketen pro Sekunde gesendet. Die Nutzdaten variieren zwischen 50, 111, 174, 350, 700, 1050 und 1400 Byte. Das Paket mit Sequenznummer 0 beinhaltet Nutzdaten in einer Größe von 50 Byte, Sequenznummer 1 in der Größe von 111 Bytes usw. Die Nutzdatengröße von 111 Byte entspricht einer CAM-Übertragung, 174 Byte der DENM-Übertragung. Um mögliche Auswirkungen der PER in Abhängigkeit der Paketgröße feststellen zu können, wird zum Vergleich die Nutzdatengröße in Schritten von 350 Byte erhöht. Die Gesamtgröße P_{size} eines Testpakets errechnet sich aus der Größe des Headers Hdr, der Größe des 802.2-SNAP $Snap$, den Nutzdaten $Data$ und der FCS Fcs.

Somit ergibt sich für ein Testpaket folgender Overhead P_{over} :

$$\begin{aligned} P_{over} &= Hdr + Snap + Fcs \\ P_{over} &= 30 + 8 + 4 \\ P_{over} &= 42\ Bytes \end{aligned}$$

Die benötigte Datenübertragungsrate pro Sekunde C errechnet sich zu:

$$\begin{aligned} C &= \sum_{i=1}^{7} (Data[i] + P_{over}) \cdot P_{rate} \cdot 8 \\ C &= 4129 \cdot 20 \cdot 8 \\ C &= 661\ kBit/s \end{aligned}$$

Somit ergibt sich für Kanal 174 (Bandbreite $B = 6\ MBit/s$) und Kanal 178 ($B = 12\ MBit/s$) folgende prozentuelle Auslastung l:

$$\begin{aligned} l_{174} &= \frac{C}{B_{174}} & l_{178} &= \frac{C}{B_{178}} \\ l_{174} &= \frac{661}{6000} & l_{178} &= \frac{661}{12000} \\ l_{174} &= 11\ \% & l_{178} &= 5,5\ \% \end{aligned}$$

Durch die geringe Auslastung der Übertragungskanäle ($l <= 11\%$) kann sichergestellt werden, dass die Übertragung der Testpakete nicht von Bandbreitenbegrenzungen beeinflusst wird.

5.2 Testablauf

Zur Ermittlung der PER werden auf zwei Cohda MK-2 Plattform unterschiedliche Anwendungen implementiert. Auf einer Plattform läuft die Anwendung zum Senden der Testpakete (TX-App), auf der zweiten die Anwendung zum Empfangen der Testpakete (RX-App). Zur Ermittlung der PER sendet die TX-App Testpakete mit fortlaufender Sequenznummer. Die RX-App wertet die empfangenen Testpakete aus und berechnet anhand der Sequenznummer die PER. Für den Aufbau des Testszenarios werden zwei Fahrzeuge mit jeweils einer MK-2 Plattform ausgestattet. Abbildung 5.2 zeigt den Aufbau des Testszenarios. Fahrzeug 1 wird mit der TX-App Plattform ausgerüstet, während Fahrzeug 2 mit der RX-App Plattform ausgestattet wird. Das Starten der Testanwendungen (RX-

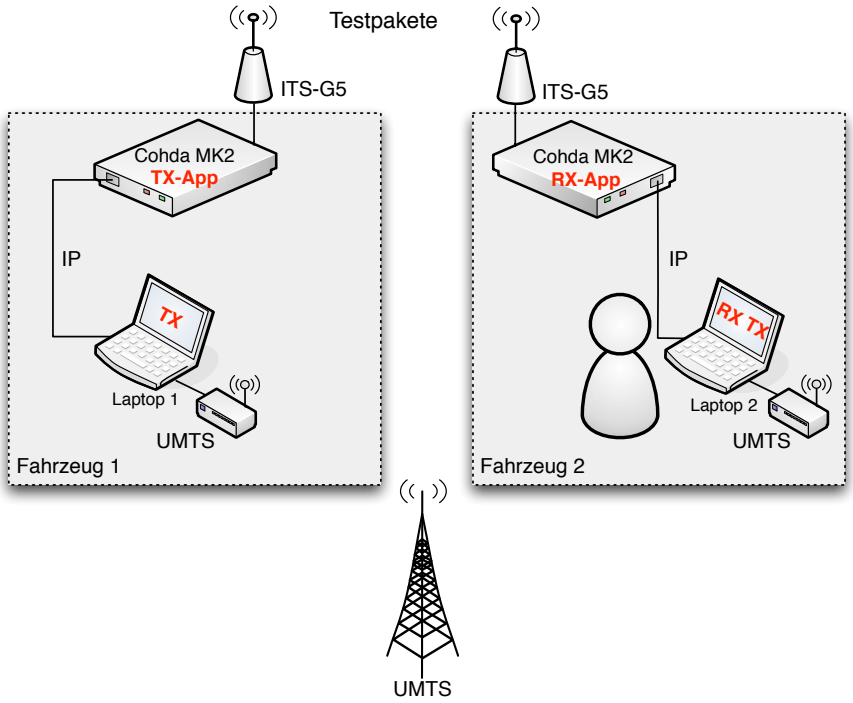

Abbildung 5.2: Schematischer Aufbau des Feldtests

App und TX-App) an jedem einzelnen Messpunkt erfolgt über den Telnet-Dienst vom Laptop aus. Dazu befindet sich in jedem Fahrzeug ein Laptop. Das UMTS-Modem wird zur Herstellung einer Internetverbindung für den Remote-Zugriff auf Laptop 1 benötigt. Das Testszenario wird vom User in Fahrzeug 2 aus gesteuert, indem am Laptop 2 lokal über Telnet die RX-App gestartet wird. Anschließend wird über Remote-Zugriff auf den entfernten Laptop 1 in Fahrzeug 1 zugegriffen und von dort aus die TX-App über Telnet gestartet. Nach erfolgreicher Messung werden die Ergebnisse beider Apps in einem Log-File auf der jeweiligen MK-2 Plattform hinterlegt.

5.3 Auswertung

Die Distanzen zwischen den Fahrzeugen werden aus einem vorhandenen maßstabsgetreuen Plan (AutoCAD-Datei) der Inffeldgasse entnommen. Im Testszenario A wird die Übertragung mit Sichtverbindung entlang der Inffeldgasse gemessen. Für Testszenario B werden Messpunkte um den Gebäudebereich Inffeldgasse 24 aufgenommen.

5.3.1 Testszenario A (LOS)

Abbildung 5.3 zeigt das Campusgelände Inffeldgasse. Für den LOS-Test wird Fahrzeug 1 am westlichen Ende der Inffeldgasse bei Referenzmesspunkt MP_{ref} positioniert. Mit Fahrzeug 2 werden entlang der Inffeldgasse Messpunkte abgefahren, um in Abhängigkeit der Distanz zwischen den zwei Fahrzeugen die PER zu messen. Zu Beginn der Messung wer-

Abbildung 5.3: Messpunkte Testszenario A

den zum Einschätzen der Reichweite Messvorgänge in größeren Intervallen ($\approx$ 150 Meter) durchgeführt. Eine feinere Auflösung der Messpunkte erfolgt in jenem Distanz-Bereich wo es zur Änderung der PER kommt. Beim Durchführen der Messung wurden für den LOS-Test 9 Messpunkte (MP_1 bis MP_9) aufgenommen. Abbildung 5.4 zeigt den Verlauf der PER für Kanal 174 in Abhängigkeit der Distanz zwischen den zwei Fahrzeugen, wobei sich die PER auf die Gesamtanzahl der überertragenen Nutzdaten (50 bis 1400 Byte) bezieht. Der Referenzmesspunkt MP_{ref} ist im Verlauf bei 0 Meter eingezeichnet. Messpunkt MP_1 befindet sich in einer Entfernung von 230 Metern, der Messpunkt MP_5 bei 532 Metern. Wie zu sehen ist, liegt die PER bei den Messpunkten MP_1 bis MP_5 mit minimalen

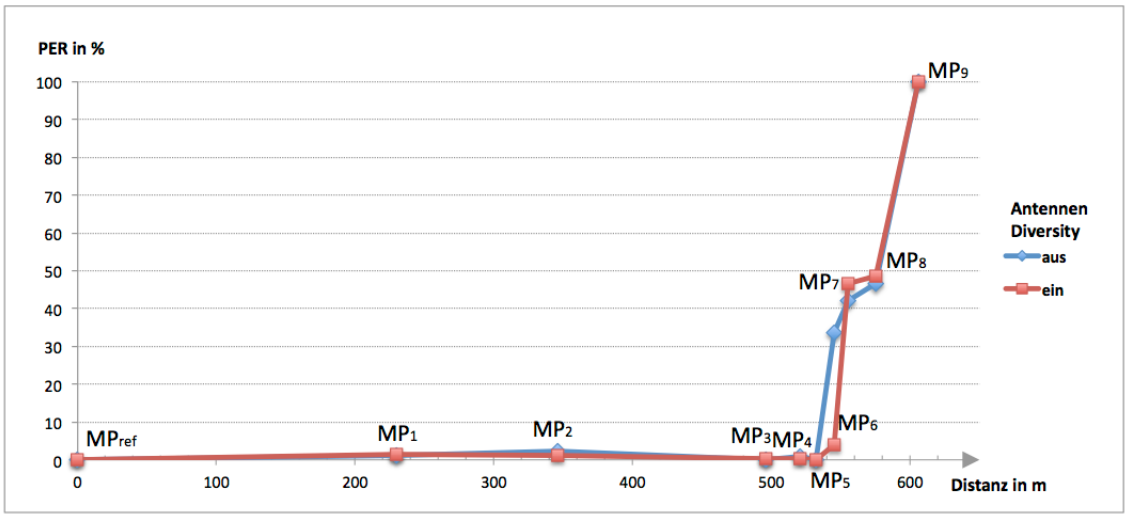

Abbildung 5.4: Verlauf der PER für Kanal 174 bezogen auf gesamte Nutzdatengröße

Abweichungen bei 0 %. Bei Messpunkt MP_6, der 15 Meter von MP_5 entfernt ist, kann ein erstes Ansteigen der PER registriert werden. Ohne Antennen-Diversity wird hier eine PER von 33,5 %, mit aktiver Antennen-Diversity eine PER von 3,9 % festgestellt. In einer Entfernung von 555 bis 575 Meter (MP_7 und MP_8) schwankt die PER bei 45 %. Ab 606 Meter (MP_9) werden keine Testpakete mehr empfangen wodurch die PER bei 100 % liegt. Für Kanal 178 (12 MBit/s) beträgt die PER bei MP_1 bereits 100 %, während sich die PER für Kanal 174 noch bei 0 % befindet. Für Kanal 178 werden auf Grund der geringen Reichweite keine weiteren Messungen durchgeführt. Abbildung 5.5 zeigt den Verlauf der PER für Kanal 174 in Abhängigkeit der Distanz zwischen den zwei Fahrzeugen, wobei sich

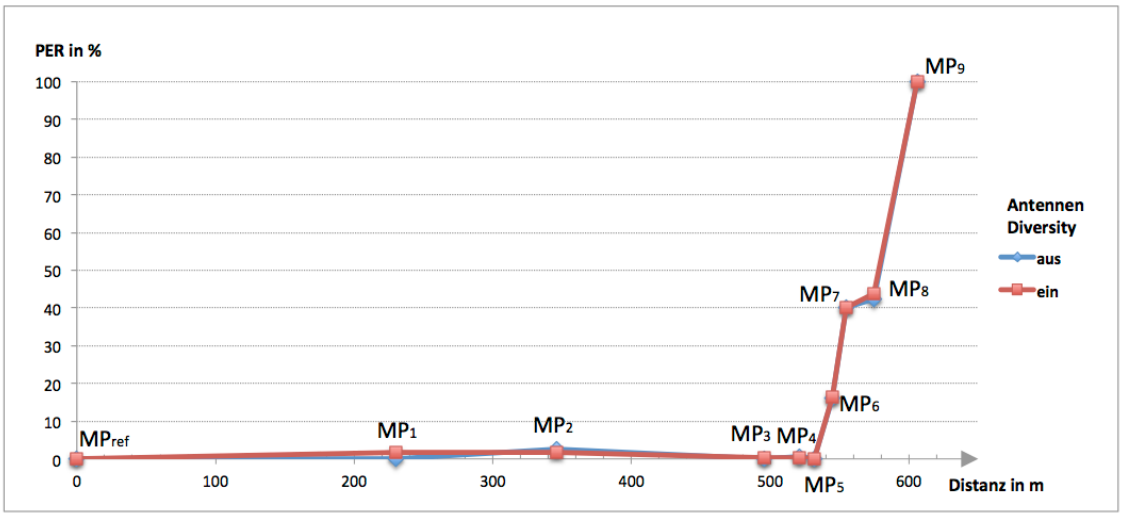

Abbildung 5.5: Verlauf der PER für Kanal 174 bezogen auf CAM und DENM Größe

die PER nur auf die Datengröße im CAM und DENM Bereich (111 und 174 Byte) bezieht. Im Vergleich zu Abbildung 5.4, bei der Nutzdaten bis zu einer Größe von 1400 Byte übertragen werden, sind nur minimale Unterschiede festzustellen. Die Größe der übertragenen Pakete hat demnach nur wenig Einfluss auf die PER. Die Messungen wurden bei stehenden Fahrzeugen durchgeführt, wodurch die Vorteile des Antennen-Diversity Betriebs nur wenig Auswirkung haben.

5.3.2 Testszenario B (NLOS)

Testszenario B wird um den Gebäudebereich Inffeldgasse 24 durchgeführt. Für den NLOS-Test befindet sich Fahrzeug 1 auf der westlichen Seite des Gebäudes. Dieser Standort dient als Referenzmesspunkt MP_{ref}. Rund um das Gebäude werden 9 Messpunkte aufgenommen. Auf Grund der schlechten Reichweitenperformance von Kanal 178 bei Testszenario A wird die Messung nur mit Kanal 174 durchgeführt, wobei die Messung mit Antennen-Diversity und ohne Antennen-Diversity erfolgt. Abbildung zeigt den Gebäudebereich der Inffeldgasse 24 mit den Messpunkten MP_1 bis MP_9. In der Abbildung sind pro Messpunkt

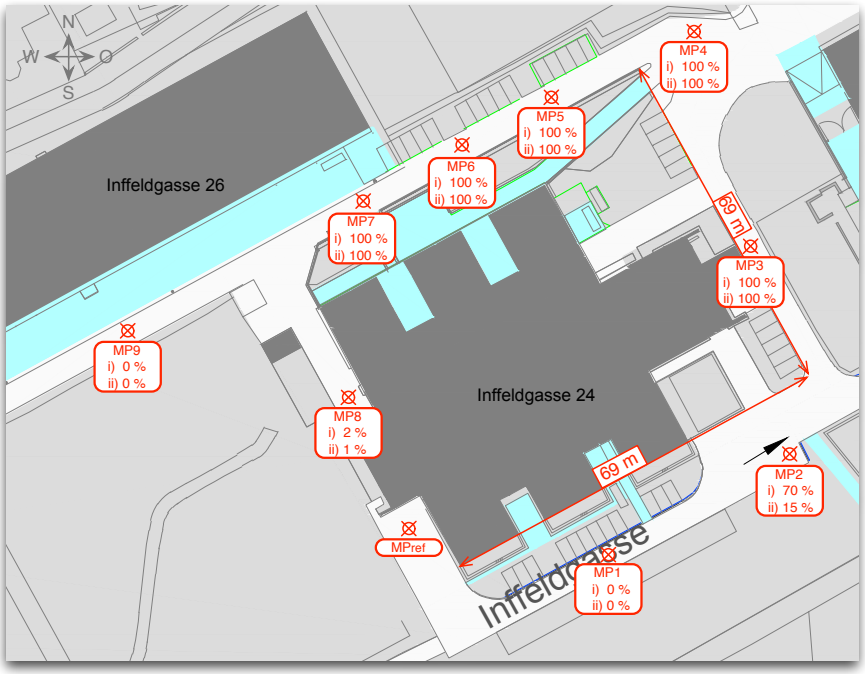

Abbildung 5.6: Messpunkte Testszenario B

MP zwei PER-Werte ersichtlich: Wert i) bezieht sich auf die Messung ohne Antennen-Diversity, Wert ii) auf die Messung mit Antennen-Diversity. Bei den Messpunkten MP_3 bis MP_7 befindet sich das Gebäude Inffeldgasse 24 zur Gänze zwischen den einzelnen Messpunkten und dem Bezugspunkt MP_{ref}. Die PER bei diesen Messpunkten liegt bei 100

%, was einer völligen Abschattung des Übertragungssignals entspricht. Bei kleinflächiger Abschattung des Signals durch das Gebäude, kommt es zu geringen Paketverlusten. So liegt die PER für MP_1, MP_8 und MP_9 mit kleinen Abweichungen bei 0 %. Im Randbereich bei MP_2 lässt sich eine deutlich niedrigere PER im Antennen-Diversity Betrieb feststellen. So beträgt die PER ohne Diversity 70 %, mit Diversity 15 %. Dieses Ergebnis lässt darauf zurückführen, dass das Signal über Reflexionen am MP_2 ankommt, da keine Sichtverbindung besteht. Im Diversity-Betrieb wird das Signal eher wiederhergestellt.

5.4 Zusammenfassung

Testszenario A zeigt, dass bei Sichtverbindung unter den in Abschnitt 5.1 angeführten Rahmenbedingungen eine nahezu fehlerfreie Kommunikation bis zu einer Distanz von $\approx$ 530 Metern möglich ist. In diesem Bereich liegt die PER abgesehen von kleinen Messschwankungen bei 0 %. Im Bereich von 555 Metern bis 575 Metern schwankt die PER in einem Bereich von 45 %. Die Übertragung wird in diesem Bereich zunehmend instabil. Ab einem Bereich von 600 Metern liegt die PER bei 100 %. Es können keine Pakete mehr erfolgreich übertragen werden. Insgesamt zeigt die Messung, dass die Übertragung in einem weiten Bereich (530 Meter) sehr stabil ist. Innerhalb einer Distanz, die kleiner als 100 Meter ist, steigt die PER jedoch von 0 % auf 100 % an. Im Bereich der fehlerfreien Übertragung ist während der Messung ein deutliches Ansteigen der PER festzustellen, wenn Objekte wie Fahrzeuge oder Menschenansammlungen die Sichtverbindung behindern.

Testszenario B zeigt, dass die Reichweite zwischen zwei Fahrzeugen bei nicht vorhandener Sichtverbindung im urbanen Bereich sehr gering ist, denn das Gebäude schattet das Signal sehr stark ab. Diese Tatsache ist vor allem für Car-to-X Kreuzungsszenarien im urbanen Bereich von großem Nachteil. Als Beispiel eines Kreuzungsszenarios sei die Warnung bei Vorrangverletzung durch einen anderen Verkehrsteilnehmer aufgrund überhöhter Geschwindigkeit angeführt. Wie die Messung in Testszenario B zeigt, kann eine Warnung bedingt durch die geringe Reichweite, unter Umständen erst zu spät erfolgen.

Kapitel 6

Zusammenfassung und Ausblick

Ziel dieser Arbeit war es zunächst den aktuellen Stand der Technik im Bereich der Car-to-X Kommunikation zu evaluieren. Dazu wurde zu Beginn der Feldversuch sim^{TD} näher beleuchtet. Über zahlreiche Literatur- und Standardrecherchen sowie der Teilnahme am 5th ETSI Workshop on ITS in Wien konnten die notwendigen Informationen erarbeitet werden. Für die Einbindung eines Car-to-X Systems wurde vorab eine Plattform gewählt. Anhand der Informationen aus den vorangegangenen Recherchen konnten Anforderungen an die Car-to-X Plattform definiert werden. Aus den verfügbaren Plattformen erwies sich die Cohda MK-2 Plattform als jene Plattform, die das Anforderungsprofil am geeignetsten erfüllte. Zur praktischen Analyse der Softwareschnittstellen wurde der Car-to-X Use-Case des elektronischen Bremslichts bei Notbremsung (EEBL) gewählt. Um die notwendigen Hardware- und Softwareschnittstellen in praxisbezogener Funktion zu testen, wurde im Labor der EEBL Use-Case zwischen zwei Plattformen aufgebaut. Die Evaluierung ergab, dass das Linux-Betriebssystem mit den entsprechenden Softwarekomponenten essenzielle Bestandteile der Car-to-X Plattform sind. Dieser Umstand bedeutet, dass erstmals Komponenten des Linux-Betriebssystems in sicherheitskritischen Funktionen des Fahrzeugs Anwendung finden. Um die Sicherheitsanforderungen erfüllen zu können, werden daher in Zukunft Maßnahmen zur Absicherung dieser Komponenten unabdingbar sein. Aus Sicht der Systemarchitektur weist die Cohda MK-2 Plattform aufgrund des eigens entwickelten ITS-G5 Chipsets, ein hohes Potential zur Serienreife auf. Daher wurde abschließend ein Straßenfeldtest durchgeführt, um die Reichweite zu analysieren. Das Testszenario bei Sichtverbindung im urbanen Bereich ergab eine fehlerfreie Kommunikation bis zu einer Distanz von 530 Metern. Bei einer Entfernung von 606 Metern konnte keine Kommunikation mehr stattfinden. Diese Tatsache zeigt, dass die Kommunikation innerhalb der relativ kleinen Distanz von 70 Metern abbricht. Im Testszenario ohne Sichtverbindung wurde festgestellt, dass Gebäudeabschattungen die Reichweite sehr stark einschränken, was vor allem für Car-to-X Kreuzungsszenarien besonders nachteilig ist.

6.1 Ergebnisse

Die Recherchen zeigten dass Forschungs- und Standardisierungsprozesse im Bereich der Car-to-X Kommunikation derzeit noch nicht abgeschlossen sind. In Europa und den USA finden länderspezifische Standardisierungsprozesse statt. Während das ETSI Standards und Richtlinien für das europäische Car-to-X System C-ITS definiert, übernimmt das IEEE diese Aufgabe in den USA. Zur Unterscheidung ist in der Literatur für das amerikanische Car-to-X System der Begriff WAVE üblich.

In beiden Ländern spielen gegenwärtig die Themen Sicherheit und Kryptographie eine wichtige Rolle bei der Definition der Standards. Grundlegende Sicherheitsthemen wie Verschlüsselungsverfahren und Signatur der Nachrichten sind im Standard definiert. Vorgaben zur PKI, die für Verwaltung der Zertifikate zuständig ist, sind in den Standards noch nicht vorgesehen. Für die Car-to-X Markteinführung müssen jedoch auch diese Infrastrukturen vorhanden sein, da ohne entsprechenden Sicherheitsmaßnahmen ein zu hohes Gefährdungspotential für Manipulation und Datenmissbrauch vorherrscht.

Aus Gründen der Hardware-Performance wurde im sim^{TD}-Feldversuch ein schwächerer Algorithmus zur Signierung der Nachrichten implementiert. Die Car-to-X Hardwareplattform besteht im sim^{TD}-Feldversuch aus PC mit Router. Generell wurde der Feldversuch unter Einhaltung der ETSI Standards und unter möglichst realen Rahmenbedingungen durchgeführt, um realitätsnahe Ergebnisse zu erhalten. Zum Zeitpunkt des Verfassens dieser Arbeit liegen noch keine offiziellen Ergebnisse zum sim^{TD}-Feldversuch vor.

In dieser Arbeit wurde die Einbindung von Car-to-X Kommunikation in die bestehende E/E-Architektur von Fahrzeugen analysiert. Die Hardwareplattform zur Einbindung soll im Bezug auf Hardware- und Softwarearchitektur eine hohe Eignung für zukünftige Serienreife aufweisen. In einer Vorauswahl erwiesen sich fünf Plattformen als prinzipiell geeignet und wurden für die engere Bewertung herangezogen. Aus dieser Bewertung ging die Cohda MK-2 Plattform als Geeignetste hervor, wobei der eigens entwickelte Chip-on-Board Transceiver zur Drahtloskommunikation besonders hervorzuheben ist. Im Gegensatz dazu verwendet die Mehrheit der Car-to-X Prototypenhersteller einen miniPCI-Chipset als Transceiver-Einheit, der üblicherweise im Bereich der Unterhaltungselektronik Anwendung findet. Bei allen Car-to-X Plattformen findet sich Linux als Betriebssystem wieder, was bis dato für sicherheitskritische Anwendungen im Automotive-Bereich eine Ausnahme darstellte.

Zur Analyse der notwendigen Hardware- und Softwareschnittstellen wurde für die ITS-Anwendung der Use-Case des elektronischen Bremslichts bei Notbremsung EEBL implementiert. Bei diesem Use-Case werden Fahrzeuge über eine DENM vor einem notbremsenden Fahrzeug gewarnt. Zur Integration in die bestehende E/E-Architektur des Fahrzeugs wurde auf folgende Software- und Hardwareschnittstellen zurückgegriffen:

- **CAN**
  Über den CAN-Bus des Fahrzeugs wird die Notbremsung von der MK-2 Plattform empfangen. Softwareseitig kommuniziert die implementierte Anwendung über einen Socket mit dem CAN-Controller der MK-2 Plattform. Der Zugriff auf den CAN-Bus erfolgt über den CAN-Socket wie ein Zugriff auf einen Ethernet-Socket.

- **GPS**
  Die GPS-Daten werden in allen Car-to-X Anwendungsszsenarien benötigt, denn im ITS-Netzwerkprotokoll GN sind die Standortkoordinaten zur Eingrenzung des Routings obligat vorhanden. Zusätzlich benötigt die ITS-Anwendung zur DENM Generierung den aktuellen Standort, da diese Nachricht ebenfalls Standortkoordinaten enthält. Der Linux Hintergrunddienst GPSD stellt den Anwendungen die Daten des GPS-Receivers anhand von UDP-Paketen zur Verfügung.

- **ITS-G5**
  Die essenzielle Schnittstelle zur drahtlosen Car-to-X Kommunikation ist durch die Cohda Transceiver-Einheit realisiert. Die ITS-Anwendung sendet und empfängt Pakete von der Cohda C-ITS-Protokollstack-Anwendung (ETSA). Die ETSA übernimmt das Routing der Pakete und greift über den 1609.4 Socket des Linux Betriebssystems auf die Coda Transceiver-Einheit zu. Durch die Tranceiver-Einheit werden die Nachrichten drahtlos über ITS-G5 versendet beziehungsweise empfangen.

- **VGA**
  Die VGA-Schnittstelle wird als HMI verwendet, um von der ITS-Anwendung die Warnung der Notbremsung am Bildschirm zu visualisieren. Für die Implementierung wurde DirectFB verwendet, wodurch die VGA-Schnittstelle direkt von der ITS-Anwendung angesprochen wird.

Die erforderlichen Schnittstellen zur Umsetzung des Use-Case EEBL zeigen, dass bedingt durch das Linux-Betriebssystem für die Softwareimplementierung auf bestehende Linux-Bibliotheken zurückgegriffen wird.

Der Feldtest wurde zur grundsätzlichen Einschätzung der Car-to-X Reichweite unter Verwendung des seriennahen Cohda ITS-G5 Chipsets durchgeführt. Die Reichweite wurde zwischen zwei Fahrzeugen anhand der PER ermittelt. Im urbanen Bereich ergab sich bei Sichtverbindung eine fehlerfreie Kommunikation bis zu einer Distanz von $\approx$ 530 Metern. Innerhalb weiterer 70 Metern steigt die Anzahl der fehlerhaft übermittelten Pakete an, wobei bei der Distanz von 606 Metern keine Pakete mehr empfangen werden. Für den Testfall ohne Sichtverbindung wurde festgestellt, dass bedingt durch Gebäudeabschattungen die Reichweite sehr eingeschränkt ist. Diese Tatsache erweist sich für Car-to-X Kreuzungsszenarien als besonders nachteilig.

6.2 Ausblick

Im Fokus dieser Arbeit steht die Integration eines seriennahen Car-to-X Systems in die bestehende Fahrzeugarchitektur. Die Arbeit zeigt, dass das Linux-Betriebssystem und die dazugehörigen APIs essenzielle Komponenten einer Car-to-X Plattform sind. Diese Tatsache impliziert, dass bisherige Linux-Softwarekomponenten, die ihren Ursprung im Bereich der Unterhaltungselektronik haben, Einzug in den Automotive-Sektor halten. Die früher strikt getrennten Welten der Unterhaltungselektronik und des Automotive-Bereichs verschmelzen dadurch zusehends. Linux findet sich bereits in Serienfahrzeugen als Betriebssystem auf Infotainment-Komponenten wieder [56]. Da das Car-to-X System eine äußerst sicherheitskritische Komponente im Fahrzeug darstellen wird, werden zukünftig Maßnahmen zur Absicherung des Linux Systems unabdingbar sein. Nur dadurch ist es möglich, den hohen Sicherheitsanforderungen des Automotive-Bereichs gerecht zu werden.

Anhang A

Abkürzungsverzeichnis

AES	Authenticated Encryption
API	Application Programming Interface
BTP	Basic Transport Protocol
CAM	Cooperative Awareness Message
CAN	Controller Area Network
C-ITS	Cooperative-ITS
DCC	Decentralized Congestion Control
DENM	Decentralized Environmental Notification Message
ECDSA	Elliptic Curve Digital Signature Standard
ECIES	Elliptic Curve Integrated Encryption Scheme
EEBL	Emergency Electronic Brake Light
EIRP	Equivalent Isotropically Radiated Power
ETSI	European Telecommunications Standards Institute
FCS	Frame Check Sequence
FTP	File Transfer Protocol
GN	GeoNetworking
GPS	Global Positioning System

GUI	Graphical User Interface
HMI	Human Machine Interface
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ITS	Intelligent Transportation System
LOS	Line-of Sight
MAC	Medium Access Control
NLOS	Non-line-of Sight
OBU	On Board Unit
PER	Packet Error Rate
PKI	Public Key Infrastructure
RSU	Road Side Unit
SDK	Software Development Kit
SNAP	Subnetwork Access Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Telecommunication System
VANET	Vehicular Ad-Hoc Network
VM	Virtuelle Maschine
WAVE	Wireless Access for the Vehicular Environment
WLAN	Wireless Local Area Network

Anhang B

ETSI Standards

Der Anhang enthält einen Überblick der ETSI Standards für die Entwicklung von Intelligent Transport Systems. Die Listen basieren auf Release 1 - Revision 1 und beinhalten die veröffentlichten und geplanten Standards mit Titel, Nummer und aktuellem Status. Die Dokumente der einzelnen Standards können anhand der Nummer im Web unter [57] aufgerufen werden.

B.1 General Standards

Standard Title	**Standard Number**	**Version**	**Status**
Communications Architecture	EN 302 665	V1.1.1	Published
Framework for Public Mobile Networks in C-ITS	TR 102 962	V1.1.1	Published
Security Architecture	TS 102 731	V1.1.1	Published
Access control, secure and privacy preserving services	TS 102 942	V1.1.1	Published
Common data dictionary	TS 102 894-2		Q2 2013
Facility layer architecture	TS 102 894		Q3 2013
Network Architecture	TS 102 636	V1.1.1	Published
Network Architecture - revised TS	EN 302 636-3		Q4 2013
ITS testing framework	EG 202 798	V1.1.1	Published

Tabelle B.1: ETSI General Standards

B.2 Application requirements

Standard Title	Standard Number	Version	Status
Road Hazard Signalling	TS 101 539-1		Q2 2013
Longitudinal Collision Risk Warning	TS 101 539-3		Q2 2013
Intersection Collision Risk Warning	TS 101 539-2		Q2 2013
Electrical Vehicle charging spot notification	TS 101 556-1		Published
Tyre Pressure monitoring systems	TS 101 556-2		Q2 2013
Planning and Reservation of EV Energy Supply	TS 101 556-3		Q2 2013
Basic Set of Applications	TR 102 638	V1.1.1	Published

Tabelle B.2: ETSI Application requirements

B.3 Facilites

Standard Title	Standard Number	Version	Status
Cooperative Awareness Message (CAM)	TS 102 637	V1.1.1	Published
Cooperative Awareness Message (CAM)	EN 302 637-2		Q3 2013
Decentralized Environmental Notification Message (DENM)	TS 102 637	V1.1.1	Published
Decentralized Environmental Notification Message (DENM)	EN 302 637-3		Q3 2013
Local Dynamic Maps	EN 302 895		Q2 2013
ITS station position and time	TS 102 890-3		Q2 2013

Tabelle B.3: ETSI Facilities

Standard Title	Standard Number	Version	Status
CAM (ATS, TSS&TP. PICS)	TS 102 859-1 TS 102 859-2 TS 102 859-3	V1.1.1 V1.1.1 V1.1.1	Published
DENM (ATS, TSS&TP. PICS)	TS 102 869-1 TS 102 869-2 TS 102 869-3	V1.1.1 V1.1.1 V1.1.1	Published
Validation of CAM	TR 103 061-1		Published
Validation of DENM	TR 103 061-2		Published

Tabelle B.4: ETSI Facilities (Testing Standards)

B.4 Network and Transport

Standard Title	Standard Number	Version	Status
GeoNetworking Requirements and Scenarios	TS 102 636	V1.1.1	Published
GeoNetworking Requirements and Scenarios rev. TS	EN 302 636-1/2		Q4 2013
GeoNetworking Media independent	TS 102 636-4-1	V1.1.1	Published
GeoNetworking Media independent rev. TS	EN 302 636-4-1		Q4 2013
GeoNetworking Media dependent (G5A)	TS 102 636-4-2		Q4 2013
Transmission IPv6 over GeoNetworking	TS 102 636-6-1	V1.1.1	Published
Basic Transport Protocols	TS 102 636-5-1	V1.1.1	Published
Basic Transport Protocols - rev. TS	EN 302 636-5-1		Q4 2013

Tabelle B.5: ETSI Network and Transport

Standard Title	Standard Number	Version	Status
Basic Transport Protocol (ATS, TSS&TP, PICS)	TS 102 870-1 TS 102 870-2 TS 102 870-3	V1.1.1 V1.1.1 V1.1.1	Published
GeoNetworking ITS G5 (ATS, TSS&TP, PICS)	TS 102 871-1 TS 102 871-2 TS 102 871-3	V1.1.1 V1.1.1 V1.1.1	Published
IP packets over GeoNetworking (ATS, TSS&TP, PICS)	TS 102 859-1 TS 102 859-2 TS 102 859-3	V1.1.1 V1.1.1 V1.1.1	Published
GeoNetworking Validation	TR 103 061-2	V1.1.1	Published
Basic Transport Protocol Validation	TR 103 061-4	V1.1.1	Published
IPv6 over GeoNetworking Validation	TR 103 061-5	V1.1.1	Published

Tabelle B.6: ETSI Network and Transport (Testing Standards)

B.5 Access and Media

Standard Title	Standard Number	Version	Status
European Profile on ITS G5	ES 202 663	V1.1.1	Published
Profile Standard on ITS G5 - rev ES	EN 302 663		Q3 2012
PHY/MAC Congestion Control	TS 102 687	V1.1.1	Published
Mitigation DSRC 5.8/5.9 GHz	TS 102 792	V1.1.1	Published
ITS G5 Channel Configuration	TS 102 724	V1.1.1	Published

Tabelle B.7: ETSI Access and Media

Standard Title	Standard Number	Version	Status
Channel Congestion 5.9 GHz (ATS, TSS&TP, PICS)	TS 102 917-1 TS 102 917-2 TS 102 917-3	V1.1.1 V1.1.1 V1.1.1	Published
Coexistence Methods DSRC/ITS G5 (ATS, TSS&TP, PICS)	TS 102 916-1 TS 102 916-2 TS 102 916-3	V1.1.1 V1.1.1 V1.1.1	Published

Tabelle B.8: ETSI Access and Media (Testing Standards)

B.6 Management

Standard Title	Standard Number	Version	Status
Decentralized Congestion Control Cross Layer	TS 103 175		Q4 2013
ITS Object Identifier Tree	TR 102 707	V1.1.1	Published
Classification of Applications	TS 102 860	V1.1.1	Published
Addressing Schemes	TS 102 723-1	V1.1.1	Published
Management Information Base	TS 102 723-2	V1.1.1	Published
ETSI TC ITS Registration List	TS 102 965		Q2 2013
Facility Communication Management	TS 102 890-1		
Facility Service Announcement	TS 102 890-2		Published

Tabelle B.9: ETSI Management

B.7 Security

Standard Title	Standard Number	Version	Status
Threat Vulnerability and Risk Analysis	TR 102 893	V1.1.1	Published
Security Mapping for IEEE 1609.2	TS 102 867	V1.1.1	Published
Confidentiality Servies	TS 102 943	V1.1.1	Published
Identity, Trust and Privacy	TS 102 943	V1.1.1	Published
Access Control, Secure and Privacy Preserving Services	TS 102 942	V1.1.1	Published
Security Header and Certificate Formats for ITS G5	TS 103 097	V1.1.1	Published

Tabelle B.10: ETSI Security

Literaturverzeichnis

[1] *DRIVE C2X Project*, 2012 (Zugriff am 10.11.2012). URL www.drive-c2x.eu.

[2] R. Stahlmann, A. Festag, A. Tomatis, I. Radusch, and Fischer. Starting european field tests for car-2-x communication: The drive c2x framework. *Proceedings of 18th ITS World Congress and Exhibition 2011*, page 9, 2011.

[3] *Pre-Drive C2X Project*, 2012 (Zugriff am 08.12.2012). URL http://wiki.fot-net.eu/index.php?title=PRE-DRIVE_C2X.

[4] Katrin Pudenz. SimTD: Wenn Autos und Motorräder miteinander sprechen. *ATZ online*, 10 2012. URL http://www.atzonline.de/Aktuell/Nachrichten/1/16791/Simtd-Wenn-Autos-und-Motorraeder-miteinander-sprechen.html.

[5] C Weiss. Konsolidierter systemarchitekurentwurf. Deliverable 21.2, Sichere Intelligente Mobilitaet Testfeld Deutschland, 09 2009.

[6] Wei-Wen Kao. Integration of GPS and dead-reckoning navigation systems. In *Vehicle Navigation and Information Systems Conference*, volume 2, pages 635–643, 1991. doi: 10.1109/VNIS.1991.205808.

[7] O. Hartkopp. Fahrzeuganbindungen durch Standard-IT-Verfahren. In *GI Jahrestagung (2)*, pages 546–550, 2007.

[8] H. Stübing, H.bing, M. Bechler, D. Heussner, T. May, I. Radusch, H. Rechner, and P. Vogel. simtd: a car-to-x system architecture for field operational tests [topics in automotive networking]. *Communications Magazine, IEEE*, 48(5):148 –154, may 2010. ISSN 0163-6804. doi: 10.1109/MCOM.2010.5458376.

[9] H. Wieker, S. Weber, J. Vogt, A. Hinsberger, T. Baum, B. Allani, and M. Fuenfrocken. Management of roadside units for the sim-td field test. *16th World Congress Exhibition ITS Services*, 2009.

[10] IEEE 1609.2: Trial-use standard for wireless access in vehicular environments - security services for applications and management messages, 2006.

[11] N. Bissmeyer, H. Stübing, M. Matthesz, J.P. Stotz, J. Schuette, M. Gerlach, and F. Friederici. Simtd Security Architecure. Technical report, simTD-Projekt, 2009.

[12] T. Leinmüller et al. A global trend for car-to-X communications. *In Proceedings of FISITA 2008 World Automotive Congress*, 2008.

[13] M. Hassnaa and Z. Yan. Vehicular Networks - Techniques, Standards and Applications. *CRC Press*, 2009.

[14] IEEE 802.11p, IEEE standard for wireless lan medium access control and physical layer specifications: Wireless access in vehicular environments (WAVE), 2010.

[15] J. Rech. *Wireless LANs: 802.11-WLAN-Technologie und praktische Umsetzung im Detail*. dpunkt, 2012.

[16] *Webpage*, 2013 (Zugriff am 15.04.2013). URL www.wi-fi.org.

[17] A. Carter. The status of vehicle-to-vehicle communication as a means of improving crash prevention performance. Technical report, National Highway Traffic Safety Administration United States of America, 2005.

[18] H. Zimmermann. OSI reference modell - the ISO model of architecure for open systems interconnection. *IEEE Transactions on Communications*, 28:425 – 432, 1980.

[19] J. D. Dally and J.W Poulton. *Digital Systems Engineering*. Cambridge University Press, 1998.

[20] IEEE 802.2: Logical link control, 1985.

[21] IEEE 1609.1: Trial-use standard for wireless access in vehicular environments - resource manager, 2006.

[22] IEEE 1609.3: Trial-use standard for wireless access in vehicular environments - networking services, 2007.

[23] IEEE 1609.4: Trial-use standard for wireless access in vehicular environments - multichannel operation, 2006.

[24] SAE J2735 - dedicated short range communications (DSRC) message set dictionary, 2009.

[25] Katrin Sjöberg. Standardization of wireless vehicular communications within IEEE and ETSI. In *IEEE VTS Workshop on Wirelss Vehicular Communications Halmstad University*, 2011.

[26] *CAR 2 CAR Communication Consortium*, 2012 (Zugriff am 20.11.2012). URL http://www.car-to-car.org.

[27] *European Telecommunications Standards Institute ETSI: Workgroup Intelligent Transportation Systems ITS*, 2012 (Zugriff am 13.11.2012). URL www.etsi.org/technologies-clusters/technologies/intelligent-transport/cooperative-its.

[28] ETSI ES 202 663 - v1.1.0: Intelligent transport systems ITS; european profile standard for the physical and medium access control layer of intelligent transport systems operating in the 5 GHz frequency band, 2010.

[29] Cooperative Mobility Systems and Services for Energy Efficiency. Deliverable 2.4 - eCoMove Cooperative Communication Protocols Specification, 07 2011. URL http://ecomove-project.eu/assets/Documents/Deliverables/110727-DEL-SP2-WP2.4.1-D2.4communicationprotocolsspecification.pdf.

[30] Guojun Dong, Hong Chen, Min Yang, and Jufeng Dai. Dynamic frequency selection (dfs) in ieee802.16e ofdm system working at unlicensed bands. In *Advanced Communication Technology, The 9th International Conference on*, volume 2, pages 1330 –1334, feb. 2007. doi: 10.1109/ICACT.2007.358603.

[31] ETSI TS 102 636-6-1 - v1.1.1: Intelligent transport systems ITS; vehicular communications; geonetworking; part 6: Internet integration; sub-part 1: Transmission of IPv6 packets over geonetworking, 2011.

[32] ETSI TS 102 636-5-1 - v1.1.1: Intelligent transport systems ITS; vehicular communications; geonetworking; part 5 transport protocols; sub-part 1: Basic transport protocol, 2011.

[33] ETSI TS 102 637-2- v1.2.1: Intelligent transport systems ITS; vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic services, 2011.

[34] ETSI TS 102 637-3- v1.1.1: Intelligent transport systems ITS; vehicular communications; basic set of applications; part 3: Specification of decentralized environmental notification basic service, 2010.

[35] ETSI TS 102 638 v1.1.1: Intelligent transport systems ITS; vehicular communications; basic set of applications; definitions, 2009.

[36] R. Moalla, B. Lonc, H. Labiod, and N. Simoni. How to secure ITS applications? In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, pages 113 –118, june 2012. doi: 10.1109/MedHocNet.2012.6257110.

[37] ETSI TS 102 867 v1.1.1: Intelligent transport systems ITS; security; stage 3 mapping of IEEE 1609.2, 2012.

[38] ETSI TS 102 893 v1.1.1: Intelligent transport systems ITS; security; threat, vulnerability and risk analysis TVRA, 2010.

[39] ETSI TS 102 687-5-1 - v1.1.1: Intelligent transport systems ITS; decentralized congestion control mechanisms for intelligent transport systems operating in the 5 GHz range; access layer part, 2011.

[40] G. Samara, W.A.H. Al-Salihy, and R. Sures. Security analysis of vehicular ad hoc nerworks (vanet). In *Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on*, pages 55 –60, sept. 2010. doi: 10.1109/NETAPPS.2010.17.

[41] B. Glas. *Trusted computing für adaptive Automobilsteuergeräte im Umfeld der Inter-Fahrzeug-Kommunikation*. Steinbuch series on advances in information technology. KIT Scientific Publishing, 2010. ISBN 9783866446021.

[42] M. Raya and J.P. Hubaux. The security of vehicular ad-hoc networks. *SASN*, pages 11–21, 2005.

[43] T. Schütze. Automotive security: Cryptograhy for car2x communication. *Rohde und Schwarz*, 3 2011.

[44] FIPS 186-3 federal information processing standards publication digital signature standard (DSS), June 2009.

[45] Information technology - security techniques - encryption algorithmus - part 2: Asymmetric ciphers, 2006.

[46] M. Dworkin. Recommendation for block cipher modes of operation: The CCM mode for confidentiality and authentication. *NIST Special Publicaton*, May 2004.

[47] AutoTalks. *Company webpage*, 2012 (Zugriff am 02.12.2012). URL www.auto-talks.com.

[48] Cohda Wireless. *Company webpage*, 2013 (Zugriff am 10.01.2013). URL www.cohdawireless.com.

[49] Fraunhofer ESK. *ARTiS-XT Produktbeschreibung*, 2013 (Zugriff am 02.02.2013). URL www.esk.fraunhofer.de/content/dam/esk/de/documents/PDB_ARTiS-XT_dt_web.pdf.

[50] NEC. *NEC LinkBird MX-4*, 2012 (Zugriff 10.12.2012). URL http://www.nec.co.jp/press/en/0811/images/1301-01.pdf.

[51] UNEX. *OBE-102 Produktbeschreibung*, 2013 (Zugriff am 05.02.2013). URL http://unex.com.tw/product/obe-102.

[52] *Cohda Wireless MK2 Wiki*, 06 2013. URL http://mk2wiki.cohdawireless.com.

[53] J. Moon and Y. Kim. Antenna Diversity Strengthens Wireless LANs. *Communication Systems Design*, pages 15–22, 2003.

[54] O. Hartkopp. Fahrzeuganbindung durch IT-Standardverfahren. In *Fahrzeugsystemelektronik*, 2007.

[55] *Mobile Mark Produktbeschreibung*, 2013 Zugriff am 15.04.2013. URL www.mobilemark.com.

[56] Alexander Kocher and Peter Kleiner. Innovationstreiber linux. *ATZelektronik*, 3(6): 28–35, 2008. ISSN 1862-1791. doi: 10.1007/BF03223934. URL http://dx.doi.org/10.1007/BF03223934.

[57] ETSI. *Standards*, März 2013 (Zugriff am 25.01.2013). URL http://webapp.etsi.org/workprogram/SimpleSearch/QueryForm.asp.

[58] *Sichere Intelligente Mobilität Testfeld Deutschland - (simTD)*, 2012 (Zugriff am 15.11.2012). URL www.simtd.de. [Online].