



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Sebastian Karel Lembcke

**Automobile Cloud-Dienste am Beispiel eines Dienstes zur
Erfassung der Luftgüte entlang der Fahrbahn**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

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

Sebastian Karel Lembcke

**Automobile Cloud-Dienste am Beispiel eines Dienstes zur
Erfassung der Luftgüte entlang der Fahrbahn**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

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

Betreuender Prüfer: Prof. Dr. Franz-Josef Korf
Zweitgutachter: Prof. Dr. Tim Tiedemann

Eingereicht am: 8. August 2017

Sebastian Karel Lembcke

Thema der Arbeit

Automobile Cloud-Dienste am Beispiel eines Dienstes zur Erfassung der Luftgüte entlang der Fahrbahn

Stichworte

Cloud Computing, Automobil, Internet of Things, Internet of Vehicles, Connected Car, Vehicle-to-Everything, Sensorik, Luftgütemessung, Geoinformationssystem

Kurzzusammenfassung

In dieser Bachelorarbeit werden die aktuell verfügbaren und von Autoherstellern und anderen Unternehmen antizipierten automobilen Cloud-Dienste analysiert, wobei Grundideen und Funktionsprinzipien zusammengefasst vorgestellt werden. Weiterhin werden eigene Ideen für Neuentwicklungen beschrieben. Als Beispiel wird hierzu ein Dienst zur Erfassung der Luftgüte entlang der Fahrbahn entwickelt. Hardware und Software werden fahrzeugunabhängig ausgelegt, wobei der Einsatz typischer Fahrzeugsensorik und Bordsysteme in diesem Kontext diskutiert wird.

Sebastian Karel Lembcke

Title of the paper

Automotive cloud services using the example of a service to measure air quality along roadways

Keywords

Cloud Computing, automobil, Internet of Things, Internet of Vehicles, Connected Car, Vehicle-to-Everything, sensor technology, air quality measurement, geographic information system

Abstract

In this bachelor thesis, currently available automotive cloud services and those being anticipated by car manufacturers and other companies are analysed and underlying ideas and operating principles are presented in summary. In addition, own ideas for new developments are described. As an example, a service to measure air quality along roadways is developed. Hardware and software are designed to work independently, although the utilization of typical vehicle sensors and on-board systems in this context is discussed.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	2
2.1	Cloud Computing	2
2.1.1	Cloud Computing im automobilen Kontext	3
2.2	Internet of Vehicles	3
2.3	Fahrzeugsensorik	4
2.3.1	Sensorik zum Erfassen des Zustands von Fahrzeug und Insassen	4
2.3.2	Sensorik zum direkten Erfassen der Fahrzeugumgebung	6
2.3.3	Indirektes Erfassen der Fahrzeugumgebung	8
3	Analyse aktuell betriebener und aufkommender Dienste	10
3.1	Infotainment, Social-Media und E-Commerce	12
3.2	Navigation	12
3.3	Remote-Dienste	13
3.4	Fleet Tracking und Vehicle Analytics	13
3.5	Car-2-X	13
3.6	Emergency Vehicle Priority	14
3.7	Freigabe des Kofferraums zur Anlieferung von Sendungen	14
3.8	Verwaltung von Benutzern für privates Car-Sharing	14
3.9	HD ¹ Maps und Connected ADAS ²	15
3.10	Community-based Parking	16
3.11	Erfassen der Straßenbeschaffenheit mit der Fahrzeugsensorik	16
3.12	Unfallverhütung durch Analyse von Bewegungsmustern	16
3.13	Kommunikation mit Ampelanlagen	17
3.14	Zugriff auf Verkehrsüberwachungskameras	17
3.15	Cloud-basiertes Deep Learning	17
4	Beispiele für Neuentwicklungen	19
4.1	Vehicular CCTV ³	19
4.1.1	Anforderungen	20
4.1.2	Stakeholder	21
4.1.3	Konzept	21

¹High Definition

²Advanced Driver Assistance Systems

³Closed Circuit Television

4.1.4	Use Case	22
4.2	Gefahrenstellenerkennung durch Meldung von außergewöhnlichen Fahrmanövern	23
4.2.1	Anforderungen	23
4.2.2	Stakeholder	23
4.2.3	Konzept	24
4.3	Abholdienst Car-Collect für autonome Fahrzeuge	25
4.3.1	Anforderungen	25
4.3.2	Stakeholder	25
4.3.3	Konzept	25
4.3.4	Use Cases	27
4.4	Effiziente Parkplatznutzung	29
4.4.1	Anforderungen	29
4.4.2	Stakeholder	30
4.4.3	Konzept	30
5	Entwicklung eines Dienstes zum Erfassen der Luftgüte entlang der Fahrbahn	32
5.1	Motivation	32
5.2	Stakeholder	34
5.3	Analyse der Problemstellung	35
5.3.1	Schadstoffbelastungen in der Luft	35
5.3.2	Anforderungen an die Sensorik	37
5.3.3	Anforderungen an die Signalverarbeitungskette	38
5.3.4	Anbindung an ein Geoinformationssystem	38
5.3.5	Bewertung der Anforderungen bezüglich der Verwendung der Fahrzeugsensorik und Bordsysteme	39
5.4	Konzeption eines fahrzeugunabhängigen Prototyps	41
5.4.1	Auswahl von Hardwarekomponenten	41
5.4.2	Entwurf der Hardware	46
5.4.3	Entwurf einer Mikrocontroller-Software zum Auslesen der Sensoren	48
5.4.4	Entwurf einer Android-Applikation zur Datenweiterverarbeitung	51
5.4.5	Entwurf eines Gehäuses	55
5.4.6	Auswahl eines Geoinformationssystems	56
5.4.7	Qualitätssicherungsmaßnahmen	57
5.5	Umsetzung des Prototyps	61
5.5.1	Hardwareseitige Umsetzung und Einbau in das Gehäuse	61
5.5.2	Implementierung der Mikrocontroller-Software	62
5.5.3	Implementierung der Android-Applikation	62
5.6	Evaluation	64
5.6.1	Durchführung einer Messung	64
5.6.2	Ergebnis der Messung	65
5.7	Methode zur Absolutwertbestimmung und Kalibrierung	67

6 Zusammenfassung und Fazit	69
6.1 Zusammenfassung	69
6.2 Fazit	70
Literaturverzeichnis	71
Abkürzungsverzeichnis und Glossar	77
Verzeichnisstruktur des Datenträgers	78

Tabellenverzeichnis

3.1	Eigenschaften der vorgestellten Dienste	11
4.1	Use Case zum Dienst Vehicular CCTV nach UML-Schema	22
4.2	Use Case zum Dienst Car Collect (Einkauf mit Car-Collect) nach UML-Schema	27
4.3	Use Case zum Dienst Car-Collect (Abfertigen einer Bestellung) nach UML-Schema	28
5.1	Werte- und Messbereich zum Testen der Mikrocontroller-Software	58
5.2	Zustände der Komponenten der Android-Applikation bezüglich Eingaben . . .	59
5.3	Fehlerfälle und Systemverhalten der Android-Applikation	60
5.4	Ergebnisse der Luftgütemessung an verschiedenen Orten	66

Abbildungsverzeichnis

2.1	Vielfalt der Fahrzeugsysteme mit Sensoren [7, S. 2]	5
2.2	Darstellung der Umfelderkennung eines Fahrzeugs [5, S. 350]	8
5.1	Kennlinien des Mehrkanal-Gassensors MiCS-6814 [55]	43
5.2	Kennlinie des Figaro Gassensors TGS 3870-B00 [4]	44
5.3	Blockschaltbild der Hardware des Prototyps	47
5.4	Mikrocontroller-Programmstruktur im Überblick als UML-Aktivitätsdiagramm	48
5.5	Sensoren-Lese- und Bluetooth-Sende-UML-Aktivität aus Abbildung 5.4	49
5.6	UML-Klassendiagramm der Android Applikation	51
5.7	UML-Sequenzdiagramm der Handler-Kommunikation	53
5.8	UML-Zustandsautomat des ReceiverHandlers	54
5.9	Schematische Darstellung des Gehäuses	55
5.10	Foto des Messgeräts	61
5.11	Grafische Oberfläche der Android-Applikation	63

1 Einleitung

Bereits heute wird in Fahrzeugen eine Vielzahl komplexer elektronischer Systeme verbaut. Hierzu zählen Sicherheits-, Komfort- und Unterhaltungssysteme, sowie auch Fahrerassistenzsysteme. Zunehmend verfügen Pkw und Lkw auch über Module, mit denen eine Internetverbindung hergestellt werden kann. So können aus dem Fahrzeug heraus Webinhalte abgerufen oder andere online verfügbare Dienste in Anspruch genommen werden. Ebenso lassen sich über die Netzwerkverbindung Daten vom Fahrzeug abrufen oder Fahrzeugfunktionen aktivieren und deaktivieren. Die steigende Vernetzung der Automobile eröffnet unter anderem die Möglichkeit Sensordaten der Fahrzeuge massenhaft zu sammeln und zu analysieren, sowie auch Verkehrsteilnehmer zu koordinieren. Hierbei spielen Cloud-Server eine zentrale Rolle, da sie zum einen als Kommunikationsknotenpunkt und zum anderen als Plattformen für die verschiedensten Dienste dienen. Ziel dieser Arbeit ist daher das breite Spektrum an automobilen Cloud-Diensten darzustellen, sowie ein konkretes Beispiel für die Realisierung eines Dienstes zu bieten.

Im Folgenden werden die aktuell verfügbaren, sowie die von verschiedenen Industrien antizipierten automobilen Cloud-Dienste analysiert. Die Konzepte der einzelnen Dienste werden zusammengefasst in eigenen Abschnitten dargestellt und erläutert. Um das breite Spektrum an möglichen zukünftigen Diensten zu veranschaulichen, werden in einem weiteren Kapitel Beispiele für Neuentwicklungen aufgezeigt. Die einzelnen Konzepte werden erläutert und gegebenenfalls anhand von Anwendungsfällen dargestellt. Als konkretes Beispiel für die Realisierung eines Dienstes wird weiterhin ein Dienst zur Erfassung der Luftgüte entlang der Fahrbahn entwickelt. Es wird die Problemstellung analysiert, Hardware ausgewählt und das System konzeptioniert und umgesetzt. Das fertige System besteht aus einem Messgerät, einer Android-Applikation und einer Server-Plattform, womit es fahrzeugunabhängig eingesetzt werden kann. Trotz der Fahrzeugunabhängigkeit wird der Einsatz typischer Fahrzeugsensorik und Bordsysteme diskutiert. Die Ergebnisse einer Messung, die mit dem System durchgeführt wurde, werden abschließen dokumentiert und bewertet.

2 Grundlagen

2.1 Cloud Computing

Im Allgemeinen wird mit Cloud Computing die Verwendung von IT-Infrastrukturen über Rechnernetze, üblicherweise das Internet, beschrieben. Bei den Infrastrukturen kann es sich unter anderem um Speicher, Rechenleistung oder Software handeln. Diese werden von Dienstanbietern zur Verfügung gestellt und sind für Nutzer über entsprechende Netzwerkschnittstellen zugänglich. Dabei gibt es einige Charakteristika die Cloud Computing auszeichnen.

Laut der NIST Definition [10] wird Cloud Computing durch fünf Merkmale charakterisiert. Diese sind on-demand self-Service, broad network access, resource pooling, rapid elasticity und measured service. Hiernach sind Nutzer von Cloud-Diensten in der Lage selbständig und nach Bedarf auf entsprechende Ressourcen zuzugreifen. Der Zugriff erfolgt über ein Netzwerk und ist gegebenenfalls von verschiedenen Client-Plattformen aus möglich. Ressourcen sind üblicherweise redundant vorhanden und werden automatisch und dynamisch zugewiesen, wobei für den Nutzer Orts-, Skalierungs- und Replikationstransparenz gegeben ist. Die Verwaltung der Ressourcen wird vom Dienstanbieter übernommen. Aufzeichnung und Analyse der Aktivitäten erlauben es die Verwaltung effizienter vorzunehmen und Vorhersagen über die Nutzung zu treffen.

Weiterhin wird zwischen drei Dienst-Modellen unterschieden: Software as a Service, Platform as a Service und Infrastructure as a Service. Software as a Service ist die Bereitstellung von Anwendungssoftware, die auf der Cloud-Infrastruktur ausgeführt wird, Platform as a Service ist die Bereitstellung einer Rechner-Plattform, um Anwendungssoftware des Nutzers auf Betriebssystemebene auszuführen und Infrastructure as a Service ist die Bereitstellung der grundlegenden Rechner-Ressourcen, um Nutzer-Software auch unterhalb der Betriebssystemebene auszuführen. Der Zugriff ist entweder exklusiv für eine Person (Private Cloud), für eine Gruppe von Personen (Community Cloud) oder für die Öffentlichkeit (Public Cloud) freigegeben. Mischformen werden als Hybrid Cloud bezeichnet.

2.1.1 Cloud Computing im automobilen Kontext

Aktuell betreiben viele Fahrzeughersteller und kooperierende Unternehmen Backends, mit denen Daten aus Fahrzeugen beziehungsweise Daten der Fahrzeugsensorik gesammelt und analysiert werden können. Über sie läuft auch die Kommunikation zwischen Fahrzeugen und Frontends, wie Smartphone-Apps oder Web-Portalen ab, genauso wie die Kommunikation zwischen Infotainment-Systemen im Auto und externen Content-Providern. Wegen der Eigenschaften der Backends wird in diesem Zusammenhang oft der Begriff Cloud-Server verwendet [58, Kap. 2.2.1]. Ein Beispiel für eine solche Cloud für fahrzeugbezogene Dienste ist das Daimler Vehicle Backend, welches von T-Systems betrieben wird [19].

2.2 Internet of Vehicles

Im Zuge der Vernetzung von Dingen des täglichen Lebens, also dem Aufkommen des Internet of Things, werden auch Fahrzeuge zunehmend in das globale Netz integriert [59, S. 55-57]. So liefern Sensoren in modernen Pkw und Lkw Daten über den Zustand des Fahrzeugs, der Insassen und der Umgebung, welche dann über eine Netzwerkverbindung an anderer Stelle genutzt werden können. Andersherum können auch Fahrer und Fahrzeugsysteme auf Daten von Außerhalb zugreifen, wobei die Datenquelle ein anderer, sich in Sichtweite befindender Verkehrsteilnehmer, eine Verkehrsüberwachungskamera oder aber ein weit entfernter Server sein kann. Daher wird in diesem Zusammenhang auch oft von Vehicle-to-Everything, V2X, oder Car-to-X gesprochen. Dies umfasst die Kommunikationsmodelle Vehicle-to-Vehicle, Vehicle-to-Infrastructure, Vehicle-to-Pedestrian und Vehicle-to-Network [59, S. 55-57] [8] [17]. Die Teilnehmer des IoV sind demnach nicht nur Fahrzeuge, sondern auch Fußgänger, Radfahrer, die Verkehrsinfrastruktur, Gebäude und entfernte Rechner-Systeme. Die Netzwerkverbindung verläuft dabei über das Mobilfunknetz und über Backend-Server, wobei die direkte Kommunikation zwischen Endpunkten in Zukunft möglich sein soll [17].

Über die reine Konnektivität der einzelnen Teilnehmer hinaus, bedarf es Mechanismen zur Verwaltung und Aufrechterhaltung des Systems. Bei entsprechender Komplexität fallen große Mengen an Daten an, die unterschiedlich relevant sind. Ebenso erfordern stark wechselnde Bedingungen eine besondere Dynamik und Stabilität. Daher ist denkbar, dass die Verwaltung des IoV sowohl auf der kooperativen Selbstverwaltung der einzelnen Teilnehmer als auch der zentralen Auswertung großer gesammelter Datenmengen beruhen wird [61].

2.3 Fahrzeugsensorik

In heutigen Fahrzeugen ist eine Vielzahl an unterschiedlichen Sensoren verbaut, die der Funktion des Antriebsstrangs, der Sicherheit, der Fahrdynamik und dem Komfort der Insassen dienen. Fahrzeuge mit Fahrerassistenzsystemen oder gar teilautonome und autonome Fahrzeuge haben darüber hinaus weitere Sensorsysteme wie Nah- und Fernbereichs-LIDAR¹ und Radarsysteme, Ultraschallsensoren, oder Bildsensoren, um die Umgebung des Fahrzeugs zu erfassen. Dieser Abschnitt bietet hierzu einen Überblick. Typische Einbauorte der Sensoren in Fahrzeugen können Abbildung 2.1 entnommen werden.

Mit Bezug auf das Thema der Arbeit wird unterschieden zwischen Sensoren die Daten über den Fahrzeugzustand und den Zustand der Insassen liefern und solchen die der Erfassung der Umgebung dienen. Erstere sind vor allem für Ferndiagnosen durch den Eigentümer, Werkstätten, Pannendienste und Rettungsdienste von Bedeutung, aber auch für statistische Verschleiß- oder Fehleranalysen, die von den Herstellern durchgeführt werden können. Informationen über die Umgebung sind vielseitig von Nutzen. Unter anderem können diese Daten von Kartenbeziehungsweise Navigationsdienstbetreibern oder auch Umweltbehörden verarbeitet werden. Anzumerken ist an dieser Stelle, dass Sensoren, die den Fahrzeugzustand überwachen meist indirekt auch Daten über die Umgebung liefern. Dies wird im entsprechenden Abschnitt erläutert.

2.3.1 Sensorik zum Erfassen des Zustands von Fahrzeug und Insassen

Zu den Sensoren, die den Fahrzeugzustand und den Zustand der Insassen erfassen, zählen im Wesentlichen die des Antriebsstranges und der Sicherheits- und Fahrdynamiksysteme, aber auch bestimmte Sensoren der Komfortsysteme. Der Inhalt dieses Abschnitts beruht im Wesentlichen auf der Bosch Fachliteratur von Konrad Reif [7] und dem Volkswagen Techniklexikon [52].

Die Sensorik im Antriebsstrang von Fahrzeugen mit Verbrennungsmotor dient zum einen dazu die Funktion von Motor und Getriebe zu Überwachen und zu Regeln, zum anderen aber auch die Steuerungsentention des Fahrers oder der Fahrerassistenzsysteme korrekt umzusetzen. Hierzu ist eine Vielzahl verschiedenster Sensoren nötig. Um den Verbrennungsprozess zu Regeln werden die Temperatur von angesaugter Luft, Kraftstoff, Kühlmittel und Motorenöl, die geförderte Kraftstoffmenge und der Einspritzpumpendruck, sowie die angesaugte Luftmenge und gegebenenfalls der Ladedruck erfasst. Ebenso werden die Temperatur und die Zusam-

¹Light Detection and Ranging

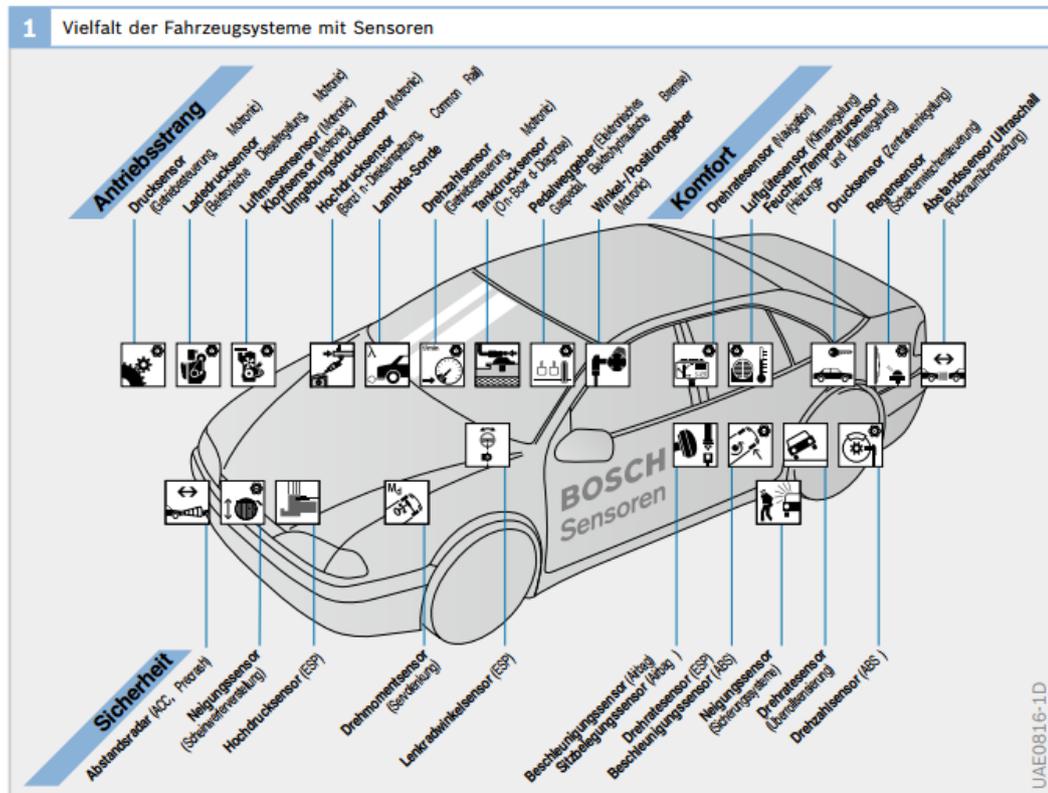


Abbildung 2.1: Vielfalt der Fahrzeugsysteme mit Sensoren [7, S. 2]

menetzung der Abgase gemessen und ausgewertet, was nicht nur für das Motormanagement, sondern auch für Abgasnachbehandlungssysteme notwendig sein kann. Für die korrekte Arbeitsweise des Motors müssen zusätzlich die Motorendrehzahl, die Stellung der Kurbel- und Nockenwelle, sowie die Drosselklappenstellung erfasst werden. Klopfensensoren überwachen dabei, ob ungewollte Selbstzündungen in den Zylindern stattfinden. Für die Steuerung von Automatikgetrieben kommen Wählhebelpositions- und Getriebedrehzahlsensoren zum Einsatz. In neueren Fahrzeugen wird typischerweise die Stellung des Fahrpedals ebenfalls mit einem Sensor erfasst, um die Stellung der Drosselklappe beziehungsweise die Einspritzanlage entsprechend einzustellen.

Die Sensorik der Sicherheitssysteme variiert stark je nach Fahrzeugmodell und Ausstattungsvariante. Ein gutes Beispiel hierfür bietet die Palette an Systemen zum Schutz der Insassen. So werden Front-Airbags und gegebenenfalls auch Gurtstraffer, von mehreren Beschleunigungs-

sensoren und unter Umständen auch Drehratensensoren ausgelöst, die im Steuergerät, an der Karosserie oder an Trägern installiert sein können. Hierzu können auch Drucksensoren in den Türen für Seiten-Airbags kommen. Komplexere Systeme erkennen durch Gewichtssensoren zusätzlich die Sitzbelegung oder das Gewicht der einzelnen Insassen, um beispielsweise bei Kindern anders zu reagieren, als bei Erwachsenen. Weiterhin kann die Sitzposition der einzelnen Insassen durch weitere Sensoren im Innenraum oder durch den Einsatz von mehreren Gewichtssensoren pro Sitz noch genauer erfasst werden.

Für die Funktion von Fahrdynamiksystemen messen Sensoren den vom Fahrer eingestellten Lenkwinkel, den aktuellen Bremsdruck oder auch Bremspedalweg und die einzelnen Raddrehzahlen. Zusätzlich können Beschleunigungs-, Neigungs-, Drehraten- oder Achssensoren zum Einsatz kommen. Auch hier variiert die Zahl und Art der eingesetzten Sensoren, je nach Fahrzeughersteller, -typ und Ausstattungsvariante. Ein einfaches Antiblockiersystem erfasst beispielsweise lediglich die Raddrehzahlen und regelt entsprechend den Bremsdruck. Komplexe Fahrdynamiksysteme errechnen wiederum aus allen Daten der genannten Sensoren Soll- und Istzustand des Fahrzeugs in Bezug auf Position und Bewegungsrichtung und regeln Bremsen, Motor und Getriebe des Fahrzeugs entsprechend.

Bestimmte Sensoren der Komfortsysteme liefern ebenfalls Informationen über den Zustand von Fahrzeug und Insassen. Ein Beispiel sind Kohlendioxidsensoren, die die entsprechende Gaskonzentration im Innenraum überwachen, und Innenraumkameras, die die Augenbewegung des Fahrers aufnehmen, um Übermüdung vorzubeugen beziehungsweise diese zu erkennen. Weiterhin verfügt die automatische Klimaanlage über Temperatursensoren.

2.3.2 Sensorik zum direkten Erfassen der Fahrzeugumgebung

Aus der Fahrzeugumgebung werden zahlreiche Daten aufgenommen, wobei die beteiligten Sensoren vor allem den Komfort- und Fahrerassistenzsystemen zuzuordnen sind. Der Inhalt dieses Abschnitts ist an die Angaben im VW Techniklexikon [52] und den Ausführungen von C. Singer in "Handbuch Fahrerassistenzsysteme" [5] angelehnt.

Zu der Sensorik der Komfortsysteme zählen in diesem Zusammenhang Regen- und Lichtsensoren, aber auch Temperatursensoren für die Regulierung der Klimaanlage und der Anzeige der Außentemperatur im Cockpit. Zusätzlich verfügen einige Autos über Luftgütesensoren, die Schadstoffe in der Außenluft feststellen und den Umluftbetrieb aktivieren.

Fahrerassistenzsysteme und vollautonome Fahrzeuge benötigen zur Erfassung der Fahrzeugumgebung beispielsweise Ultraschallabstandssensoren, LIDAR- und Radarsysteme, sowie Bildsensoren. Ein einfaches Fahrerassistenzsystem ist eine Ultraschalleinparkhilfe, die die Entfernung zu Objekten im Nahbereich um das Fahrzeug in Höhe der Stoßstange misst und dem Fahrer mitteilt. Bei adaptiven Geschwindigkeitsregel-, Notbremsassistenten- und Fahrstreifenwechselassistentensystemen kommt oft Radartechnologie zum Einsatz. Mit Radarsystemen können der Abstand zu Objekten und die Geschwindigkeit anderer Verkehrsteilnehmer ermittelt werden. Dies kann sowohl im Nah-, als auch im Fernbereich sein, wie es bei der Geschwindigkeitsregelung der Fall ist. Ebenfalls zur Abstandsmessung werden LIDAR-Systeme eingesetzt, die im Gegensatz zu Radar-Systemen keine Mikrowellen, sondern infrarotes oder ultraviolettes Licht aussenden und empfangen. LIDAR-Systeme können zwar in der Regel das Umfeld eines Fahrzeugs deutlich exakter als Radar-Systeme erfassen, sind jedoch aufgrund der Dämpfung der Signale bei schlechter Witterung nur eingeschränkt nutzbar. Bildsensoren, beziehungsweise Kameras, werden vielfältig in Fahrzeugen eingesetzt. Im einfachsten Fall wird der Fahrer durch eine Rückfahrkamera unterstützt. Auch gibt es in Fahrzeuge integrierte Nachtsichtgeräte auf Kamerabasis. In teil- und vollautonomen Fahrzeugen werden Kameras für die Fahrspur- und Straßenschilderkennung verwendet. Sie eignen sich wegen der präzisen Wiedergabe des realen Umfelds besonders gut für derartige Aufgaben. Die Tiefeninformation und somit der Abstand zu Objekten kann durch den Einsatz von Stereokameras ermittelt werden. Auch **TOF²**-Kameras liefern diese Information, wobei im Gegensatz zu Stereokameras keine algorithmisch aufwendige Verarbeitung der Bilder notwendig ist. Die Umfelderkennung eines Fahrzeugs mit den genannten Systemen wird in [Abbildung 2.2](#) veranschaulicht.

Meist werden die Daten der einzelnen Sensoren fusioniert, um das Fahrzeugumfeld zuverlässiger erfassen zu können. Dies ist besonders im Hinblick auf autonome Fahrzeuge wichtig, da sich nicht auf die Wahrnehmung des Menschen verlassen werden kann.

²Time of Flight (Laufzeitverfahren)

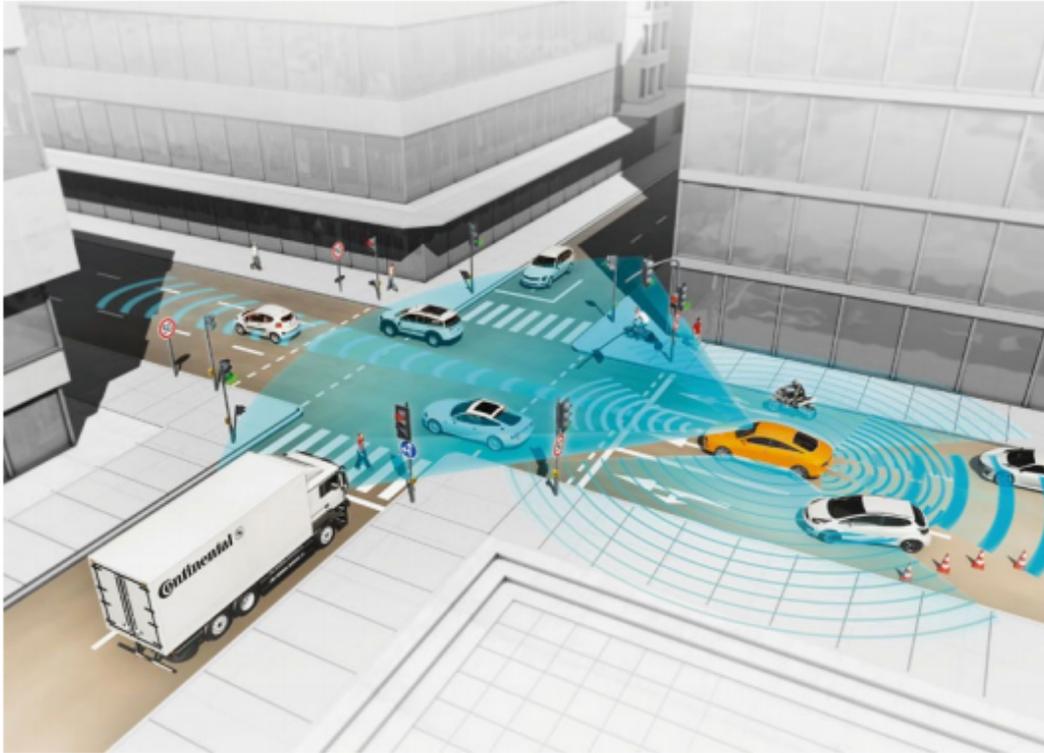


Abbildung 2.2: Darstellung der Umfelderkennung eines Fahrzeugs [5, S. 350]

2.3.3 Indirektes Erfassen der Fahrzeugumgebung

Wie eingangs bereits angedeutet liefern Sensoren, die in erster Linie den Zustand von Fahrzeug und Insassen überwachen, indirekt auch Informationen über die Umgebung. Die Grundsätzliche Überlegung hierzu ist, dass Interaktionen zwischen Insassen und Umgebung oder Fahrzeug und Umgebung beidseitig verändernd wirken können. Daher ist es in vielen Fällen möglich durch sich verändernde Größen im und am Auto einen Rückschluss auf die Veränderung der Umgebung zu ziehen. Die Änderungen in der Umgebung sind dann gegebenenfalls für andere Verkehrsteilnehmer durch ihre Außensensorik erfassbar. Ein einfaches Beispiel hierfür ist ein am Straßenrand stehendes Auto, bei welchem die Tür geöffnet wird. Position, Geschwindigkeit und das Öffnen der Tür werden von der fahrzeuginternen Sensorik erfasst. Der Rückschluss, der gezogen werden kann, ist dass die geöffnete Tür einen Teil der Fahrbahn blockieren wird. Dies könnte dann unter anderem durch Distanzsensoren anderer Fahrzeuge registriert werden. Das Prinzip gilt beispielsweise auch für die Sensorik des Antriebsstranges. Die Menge der angesaugten Luft, des verbrauchten Treibstoffs und die Abgaszusammensetzung sind

2 Grundlagen

Indikatoren für die Umweltbelastung, wenn die Daten vieler Pkw und Lkw in einem Bereich erfasst werden.

3 Analyse aktuell betriebener und aufkommender Dienste

In diesem Kapitel wird zum einen das derzeitige Angebot an verfügbaren automobilen Cloud-Diensten analysiert und zusammengefasst, zum anderen werden aufkommende und antizipierte Technologien vorgestellt, die momentan von Fahrzeugherstellern oder IT-Unternehmen beworben werden. Aufgrund des raschen technologischen Fortschritts und der Schnelllebigkeit des Marktes wird auf eine Trennung zwischen bereits bestehenden und künftigen Technologien verzichtet.

In Tabelle 3.1 werden den Diensten charakteristische Eigenschaften zugeordnet, um einen besseren Überblick zu verschaffen. In der Übersichtstabelle sind die Eigenschaften Plattformübergreifend, Echtzeit, Crowdsourcing und Aktorenfernzugriff eingetragen. Plattformübergreifend beschreibt, dass die Möglichkeit besteht den entsprechenden Dienst auf einer anderen Hardware-Plattform als der im Fahrzeug verbauten zu betreiben. Dies gilt insbesondere für Smartphones, die aufgrund der hohen Rechenleistung und vielfältigen Sensorik viele Dienste nutzbar machen. Die Eigenschaft Echtzeit weist darauf hin, dass Echtzeitanforderungen bestehen, also dass Deadlines eingehalten werden müssen. Meist handelt es sich um weiche Echtzeitanforderungen, wie beim Media-Streaming. Dienste mit harten Echtzeitanforderungen können technisch noch nicht realisiert werden. In Zukunft ist jedoch damit zu rechnen, dass dies der Fall sein wird. Die entsprechenden Tabelleneinträge sind in Klammern gesetzt. Die Eigenschaft Crowdsourcing weist darauf hin, dass der Dienst auf die Daten vieler Fahrzeuge angewiesen ist. Die aufgenommenen Daten werden meist mit statistischen Mitteln in der Cloud analysiert, um dann darauf basierende Informationen zur Verfügung zu stellen. Aktorenfernzugriff beschreibt die Möglichkeit über Frontends auf die Aktorik von Fahrzeugen zuzugreifen. Dies stellt eine besondere Art von Dienst dar, da der Zustand des Fahrzeugs verändert werden kann, ohne dass sich eine Person im Auto oder in der Nähe befindet, um die Vorgänge zu überwachen. Die aufgeführten Dienste sind zeitlich aufsteigend sortiert, d.h. die bereits betriebenen sind oben in der Auflistung, gefolgt von aktuell erprobten. Weiter unten befinden sich Dienste, die sich noch in der Entwicklung befinden.

Tabelle 3.1: Eigenschaften der vorgestellten Dienste

	Plattform- übergreifend	Echtzeit	Crowd- sourcing	Aktorenfern- zugriff	Seite
Infotainment	Ja	Ja	Nein	Nein	12
Navigation	Ja	Ja	Ja	Nein	12
Remote-Dienste zur Statusabfrage und zum Fernsteuern von Fahrzeugfunktionen	Nein	Nein	Nein	Ja	13
Fleet Tracking und Vehicle Analytics	Nein	Ja	Nein	Nein	13
Car-2-X	Nein	Nein	Ja	Nein	13
Emergency Vehicle Priority	Nein	Ja	Nein	Nein	14
Kofferraumfreigabe für Lieferdienste	Nein	Nein	Nein	Nein	14
Car-Sharing Nutzerverwaltung	Nein	Nein	Nein	Nein	14
HD-Maps und Connected ADAS	Nein	(Ja)	Ja	Nein	15
Community-based Parking	Nein	(Ja)	Ja	Nein	16
Erfassen der Beschaffenheit der Straße	Nein	Nein	Ja	Nein	16
Unfallverhütung durch Analyse von Bewegungsmustern	Ja	(Ja)	Nein	Nein	16
Kommunikation mit Ampeln	Nein	Ja	Nein	Nein	17
Zugriff auf Verkehrsüberwachungskameras	Nein	Ja	Nein	Nein	17
Cloud-basiertes Deep Learning	Nein	Nein	Ja	Nein	17

3.1 Infotainment, Social-Media und E-Commerce

Zum Infotainment-Angebot, welches in Fahrzeugen nutzbar ist, zählen vor allem Media-Streaming-Dienste. So können mit dem System AudiConnect verschiedene Internetradiosender gehört und Musikstücke über den Streaming-Dienst Napster abgerufen werden. Auch auf Social-Media- und Email-Konten kann mit AudiConnect zugegriffen werden [29]. Grundsätzlich ist es hier, sowie auch über andere Frontends, möglich Käufe im Auto abzuwickeln. So kann an Tankstellen oder in Parkhäusern gezahlt oder eine Veranstaltung oder ein Hotel gebucht werden. Buyer-to-Buyer-Marktplätze, wie das SAP Vehicles Network ermöglichen es entsprechenden Betreibern ihre Geschäftsvorgänge über In-Car-Apps abzuwickeln [21]. In einigen Autos, wie auch in einigen Modellen von BMW mit ConnectedDrive, ist es möglich Webinhalte über einen Browser im Fahrzeug abzurufen [30]. Hierdurch kann dementsprechend auf alle Dienste zugegriffen werden, die über ein Frontend-Interface zur Verfügung stehen.

Verfügt ein Fahrzeug über keine **Connectivity-Unit**¹, so kann dafür die Internetverbindung eines Smartphones genutzt werden. Für die Kopplung existierender Verbindungsstandards wie MirrorLink, Apple CarPlay oder Android Auto, die auch von vielen Zubehör-**Head-Unit**²s unterstützt werden. Hierbei werden Applikationen auf dem Smartphone ausgeführt, jedoch erfolgt die Bedienung über die Bedienelemente des Fahrzeugs. Ebenfalls wird das User-Interface auf dem fahrzeuginternen Display angezeigt [42] [38] [28].

3.2 Navigation

Im Gegensatz zu klassischen Navigationssystemen bieten aktuelle Systeme Echtzeit-Verkehrsinformationen und automatische Updates, wie etwa Google Maps und Google Street View bei AudiConnect. Ebenso ist es möglich Routen über die entsprechenden Webseiten zu planen und an das Navigationssystem im Auto zu senden. Neben der Navigationsfunktionalität und dem Abruf von Verkehrsinformationen können auch freie Parkplätze angezeigt werden. Dies gilt unter anderem für Parkhäuser mit entsprechender automatischer Überwachung und Netzwerkanbindung [29]. Bei Fahrzeugen von BMW mit ConnectedDrive wird dieser Dienst über die Applikation ParkNow bereitgestellt [30].

Auch Navigationssysteme wie Google Maps können ohne Connectivity-Unit im Auto verwen-

¹Hardware Modul für die Internetanbindung

²Steuereinheit des Mediensystems in Fahrzeugen

det werden, indem ein Smartphone gekoppelt wird. Dazu stehen die in Abschnitt 3.1 genannten Schnittstellen zur Verfügung.

3.3 Remote-Dienste

Durch eine Internetverbindung im Pkw ist es nicht nur möglich aus dem Fahrzeug heraus auf Daten von entfernten Servern zuzugreifen, sondern auch von externen Plattformen aus Daten von Fahrzeugen abzurufen oder Fahrzeugfunktionen in Anspruch zu nehmen. So kann der Fahrer beziehungsweise der Eigentümer bei Fahrzeugen mit AudiConnect unter anderem den Stand des Treibstofftanks, die bisher gefahrene Strecke oder die Position des Fahrzeugs aus der Ferne abrufen. Ebenso kann die Klimaanlage oder die Standheizung eingeschaltet und die Türen oder der Kofferraum geöffnet werden. Weiterhin können Werkstätten erste Diagnosen vornehmen, indem Fehlerberichte online abgerufen werden [29]. Grundsätzlich erfolgt hier, wie auch bei anderen Herstellern, die Verbindung zum Auto über das Backend des Fahrzeugherstellers. Dies dient dem Schutz vor unbefugten Zugriffen auf das Fahrzeug und es ermöglicht dem Hersteller Nutzungsstatistiken aufzustellen [58, S. 24] [15]

3.4 Fleet Tracking und Vehicle Analytics

Zur Überwachung von Fahrzeugflotten sind seit einigen Jahren verschiedene Systeme im Einsatz. Die Vernetzung von Pkw und Lkw ermöglicht es immer detailliertere Daten der einzelnen Fahrzeuge einzuholen und zu analysieren. Ein aktuelles Beispiel für eine fahrzeugherstellerunabhängige Plattform ist Mojio. Das Betreiberunternehmen bietet zum einen Hardwarelösungen an, um in Fahrzeugen nachträglich eine Internetanbindung herzustellen. Zum anderen bietet es Software und Backendlösungen an, mit denen gesammelte Daten, wie Informationen aus den Steuergeräten oder gefahrene Strecken, analysiert werden können [44]. Ähnliche Systeme werden bereits unter anderem von Lieferdiensten wie UPS eingesetzt. Diese dienen sowohl der Optimierung des Treibstoffverbrauchs und der zum Liefern benötigten Zeit, wie auch der Überwachung der Belastung der mechanischen Bauteile der Fahrzeuge, was durch entsprechende Sensoren erfasst wird [50].

3.5 Car-2-X

Mercedes-Benz verbaut als erster Hersteller die Car-2-X Technologie serienmäßig in einigen Fahrzeugmodellen. Gefahrenstellen, wie Stauenden, oder bestimmte Verkehrssituationen, die

von Fahrzeugsystemen oder dem Fahrer gemeldet werden, werden in der Cloud gespeichert und sind für andere Abrufbar. Befindet sich ein ein Auto in der Nähe der gemeldeten Gefahrenstelle, so wird der Fahrer alarmiert [31].

3.6 Emergency Vehicle Priority

Emergency Vehicle Priority ist eine aufkommende Technologie, die unter anderem in Australien bereits in ausgewählten Städten eingesetzt wird. Einsatzfahrzeuge nutzen die Vehicle-to-Infrastructure Kommunikation, um Ampeln entlang ihrer Route auf Grün zu schalten. Dies verhindert Verzögerungen und reduziert Risiken, die beim Überfahren Roter Ampeln entstehen. Anhand der Position und Geschwindigkeit von Polizei-, Kranken- oder Feuerwehrwagen wird die Ankunftszeit an den entsprechenden Ampeln ermittelt, um diese rechtzeitig umzuschalten. Hat das Fahrzeug die Kreuzung passiert, so schalten diese wieder auf Normalbetrieb um [22].

3.7 Freigabe des Kofferraums zur Anlieferung von Sendungen

Aktuell wird in Pilotprojekten von verschiedenen Autobauern, wie Smart [49] und Audi [12], ein Konzept erforscht, welches es Paketzustellern ermöglichen soll Sendungen im Kofferraum der Autos der Empfänger zu hinterlegen, ohne einen Schlüssel zu benötigen [16]. Der Lieferant erhält hierfür einmalig eine digitale Zugangsberechtigung zum Öffnen des Kofferraums auf das Smartphone gesendet. Diese erlischt entsprechend wieder nach Verschließen des Kofferraums.

3.8 Verwaltung von Benutzern für privates Car-Sharing

Der Autohersteller Smart plant Fahrzeuge technisch ab Werk für Car-Sharing vorzubereiten. Der Fahrzeugeigentümer kann Personen einladen sein Fahrzeug zu nutzen, welche hierzu eine Email erhalten. Nach einer Registrierung kann das Auto per Smartphone-App gebucht werden. In welchem Zeitraum dies möglich ist wird vom Eigentümer festgelegt. Zum Öffnen des Fahrzeugs dient das Smartphone mit der entsprechenden App, welche auch die Aktivitäten der einzelnen Nutzer aufzeichnet. Gestartet wird das Auto mit einem im Fahrzeug hinterlegten Schlüssel [9] [23].

3.9 HD Maps und Connected ADAS

Um die Sicherheit und Effizienz beim autonomen und teilautonomen Fahren zu erhöhen arbeiten zahlreiche Unternehmen derzeit an sogenannten HD Maps [51], auch HD Live Maps [37] oder HAD (Highly Automated Driving) Maps genannt [41]. Fahrzeuge die mit Laserscannern, Kameras und anderen Sensoren konstant ihre Umgebung abtasten, senden hierbei ihre Daten an ein entsprechendes Backend, wo aus der Datensammlung aller Fahrzeuge dreidimensionale Karten mit einer Genauigkeit von wenigen Zentimetern entstehen. Zusätzlich zu den stationären Objekten, also Kantsteinen, Schildern und Ampeln, Fahrbahnmarkierungen oder Erhöhungen und Vertiefungen in der Fahrbahn, werden auch dynamische Objekte beziehungsweise temporäre Ereignisse, wie etwa Staus und Gefahrenstellen aufgenommen und in die Karten eingetragen. Diese aggregierten Karten können dann von den einzelnen Fahrzeugen abgerufen werden, um jeweils die eigenen Sensordaten zu ergänzen. Durch die Redundanzen erhöht sich nicht nur die Genauigkeit, mit der Fahrzeuge ihre Umgebung erkennen, sondern es ermöglicht auch, dass Fahrzeuge und Fahrer Objekte und Situationen registrieren, auch wenn diese nicht in Sichtweite sind. Oft wird in diesem Zusammenhang von Connected ADAS, Electronic Horizon [36] oder ADAS Horizon [24] gesprochen.

Das Unternehmen MobileEye arbeitet daran die zeitliche Auflösung der HD Maps so zu erhöhen, dass diese nahezu Echtzeit- oder Time-To-Reflect-Reality-Qualität erreichen, um so zum Beispiel ein redundantes System zur Erkennung der Fahrspuren zu erhalten, auch wenn Fahrzeuge sich mit hohen Geschwindigkeiten fortbewegen [47]. Weiterhin sollen HD Maps dazu verwendet werden die Genauigkeit bei der Positionsbestimmung von Fahrzeugen zu erhöhen, da GPS-Systeme allein zu ungenau sind, um sie in Kombination mit solchen Karten einzusetzen. Ein entsprechendes Produkt ist RoadDNA von TomTom. Durch Transformation der dreidimensionalen Kartendaten in Längs- und Queransichten können Fahrzeuge ihre Sensordaten effizient mit den Karten abgleichen, um so ihre Position exakt zu bestimmen [51].

Das ADASIS Forum [24] definiert Standards und Protokolle für Sensordaten HD Maps und Connected ADAS, damit Entwickler und Hersteller entsprechender Systeme voneinander profitieren können. Zu den Mitgliedern gehören Autohersteller, Navigationskarten Anbieter, Navigationssystementwickler und Fahrerassistenzsystementwickler.

3.10 Community-based Parking

Bosch und Daimler erproben derzeit das Community Based Parking. Fahrzeuge, die mit Ultraschallsensoren ausgestattet sind, scannen im Vorbeifahren den Straßenrand nach Lücken ab und übermitteln die Daten an ein entsprechendes Backend (Daimler Vehicle Backend, Bosch IoT³ Cloud). Hier werden die Daten, also die Größe und Position der Lücken, aggregiert und zu einer Parkraumkarte weiterverarbeitet. Um Irrtümer auszuschließen, beispielsweise die Verwechslung von Einfahrten und Parklücken, werden die Messungen vieler Fahrzeuge zu verschiedenen Uhrzeiten miteinbezogen und mit Data-Mining-Verfahren analysiert. So kann beispielsweise davon ausgegangen werden, dass es sich bei Lücken, die immer frei sind, um Einfahrten handelt. Die Karten können dann abgerufen und im Fahrzeug oder auf dem Smartphone angezeigt werden. Aufbauend auf die Parkraumkarten ist geplant Parkprognosen sowie Echtzeitbelegungskarten zu erstellen [34] [20].

3.11 Erfassen der Straßenbeschaffenheit mit der Fahrzeugsensorik

Durch die Auswertung der Daten heute bereits vielfach verbauter Sensoren ist es möglich Rückschlüsse auf die Straßenbeschaffenheit zu ziehen. Durch messen der Vertikalbeschleunigung und des Stoßdämpferhubs an den einzelnen Rädern können beispielsweise Schlaglöcher erkannt werden. Die Daten können dann zusammen mit der Position in die Cloud übertragen werden. In der Cape-to-Cape-Challenge in Süd-Afrika wurde dies mit einem technisch angepassten VW Touareg erprobt [1]. In Land Rover Fahrzeugen wird diese Technologie bereits lokal unter dem Namen Pothole Detection Technology eingesetzt, um das Fahrwerk dynamisch anzupassen wenn Schlaglöcher durchfahren werden. In Zukunft sollen dann die Daten vieler Autos zur Verarbeitung an Server gesendet werden. Möglich ist auch, dass zusätzlich zu Beschleunigungs- und Hubsensoren, Stereokameras zum Erfassen der Straßenbeschaffenheit zum Einsatz kommen [46].

3.12 Unfallverhütung durch Analyse von Bewegungsmustern

Die Kommunikation mit anderen Verkehrsteilnehmern, darunter auch Smartphones von Fußgängern und Radfahrern, kann nicht nur dazu verwendet werden, um vor Gefahrenstellen zu warnen, die nicht in Sichtweite sind, sondern auch zur Verhütung von Unfällen zwischen

³Internet of Things

regulären Verkehrsteilnehmern. So sollen künftig Bewegungsmuster beziehungsweise die antizipierte Bahn jedes einzelnen errechnet und bevorstehende Kollisionen erkannt werden. Radfahrer, Autofahrer, Fußgänger und die Software von autonomen Fahrzeugen können so frühzeitig informiert werden, um entsprechend zu reagieren [17].

3.13 Kommunikation mit Ampelanlagen

Vernetzte Ampeln sollen in Zukunft Abgasemissionen reduzieren, indem den Fahrern der bei Rot wartenden Fahrzeugen mitgeteilt wird wann die nächste Grünphase eintritt, damit diese entsprechend ihre Motoren ausschalten können. Ebenso sollen sie dazu beitragen stockenden Verkehr in Städten zu beseitigen. Hierzu werden Fahrern beim Anfahren an rote Ampeln mitgeteilt bei welcher Geschwindigkeit sie fahren müssen, um nicht anhalten zu müssen. Das Konzept wurde zwischen 2013 und 2015 unter anderem von Siemens in Newcastle erforscht [13].

3.14 Zugriff auf Verkehrsüberwachungskameras

Damit Fahrer schlecht zu überblickende Kreuzungen besser einsehen können ist es möglich, dass die Aufnahmen von Überwachungskameras in Echtzeit aus dem Auto heraus abrufbar werden. Diese würden dann per Video-Stream auf das Display des Infotainment-Systems übertragen werden. Zum einen dient dies der Sicherheit, zum anderen aber auch der Energieeffizienz, da das Verkehrsaufkommen schnell überblickt und eingeschätzt werden könnte [17].

3.15 Cloud-basiertes Deep Learning

Der Computerhardware-Entwickler NVIDIA hat kürzlich die sogenannte DRIVE PX 2 Hardware-Plattform vorgestellt, die für den Einsatz in autonomen und teilautonomen Fahrzeugen bestimmt ist. Die Plattform ist speziell für die Verarbeitung von Sensordaten und deren Fusion, sowie die Kopplung mit HD Maps entwickelt worden [43]. Durch den Einsatz von leistungsstarken CPUs und GPUs eignet sie sich auch für Anspruchsvolle Anwendungen wie Deep Learning. Ein Beispiel wäre der Einsatz von Convolutional Neural Networks zur Bilderkennung. Zum Trainieren der Netzwerke kann eine Daten-Pipeline zwischen Fahrzeug und Backend, bestehend aus NVIDIA DGX-1 Servern aufgebaut werden. Diese Server sind mit 8 NVIDIA Tesla P100 GPUs und zwei 20-Kern Intel Xeon CPUs ausgestattet [33], was sie besonders

Leistungsfähig macht und die Trainingszeiten im Gegensatz zu konventionellen Servern deutlich verkürzt. Deep Learning in der Cloud entlastet nicht nur die Hardware in Fahrzeugen, sondern ermöglicht auch Daten vieler Fahrzeuge für das Trainieren einzusetzen. Aktuell wird das Konzept jedoch noch nicht eingesetzt.

4 Beispiele für Neuentwicklungen

In diesem Kapitel werden eigene Ideen für Neuentwicklungen aus dem Bereich automobiler Cloud-Dienste vorgestellt. Es soll anhand von Beispielen ein Verständnis dafür geschaffen werden, welche Dienste in Zukunft betrieben werden könnten. Daher stehen bei den Beispielen nicht die konkreten Umsetzungen, sondern jeweils die grundlegenden Konzepte im Vordergrund. Die Ideen für die im Folgenden vorgestellten Dienste entstammen der Überlegung wofür die Sensorik, die Aktorik und andere Systeme in Fahrzeugen neben ihrem Haupteinsatzzweck verwendet werden können. Dies bezieht sich sowohl auf die Komponenten, die aktuell in Fahrzeugen verbaut sind, als auch auf solche, die erwartungsgemäß zukünftig in Fahrzeugen vorzufinden sein werden.

4.1 Vehicular CCTV

Die Videoüberwachung öffentlich zugänglicher Orte ist heutzutage gängige Praxis bei der Prävention und Aufklärung von Verbrechen. Besonders in urbanen Gebieten werden Überwachungskameras eingesetzt, die meist an Bahnhöfen, Kreuzungen, Parkhäusern und öffentlichen Plätzen angebracht sind. Ihre Aufnahmen helfen unter anderem Fälle von Raub oder Vandalismus aufzuklären und wirken gleichzeitig abschreckend. Überwachungskameras sind jedoch meist nur vereinzelt angebracht oder haben einen begrenzten Erfassungsbereich. Flächendeckende Videoüberwachung ist zum einen aufgrund der hohen Kosten, vor allem aber aus Datenschutzgründen nicht realisierbar. Da das Gefahrenpotential in bestimmten Bereichen oft nur temporär gegeben und nicht vorhersehbar ist, ist die strategische Positionierung von Kameras oft zeitlich nicht umsetzbar oder unrentabel. So kommt es zum Beispiel besonders bei Großveranstaltungen vor, dass sich laut oder aggressiv verhaltende Personengruppen auf öffentlichen Wegen aufhalten. Wenn es zu Vorfällen kommt, können sich Einsatzkräfte erst bei Ankunft ein Bild von der Situation machen. Ein weiteres Beispiel sind von Raubüberfällen oder Diebstählen flüchtende Täter, die nur schwer verfolgt werden können, da sie schnell den Sichtbereich von Überwachungskameras verlassen.

Durch den Zugriff auf Außenkameras von parkenden Fahrzeugen wäre es möglich urbane Gebiete nahezu flächendeckend zu überwachen, wobei keine Zusatzkosten entstünden. Dabei wird davon ausgegangen, dass ein Großteil der Pkw und Lkw in Zukunft über die nötige Hardware verfügen. Überall, wo entsprechende Fahrzeuge stehen, könnten Bilder der Umgebung abgerufen werden. Dies würde auch Einsatzkräften ermöglichen Situationen vor dem Eintreffen einzuschätzen. Weiterhin könnten bei Feuerwehreinsätzen die Lokalitäten vor dem Losfahren begutachtet werden, um so gegebenenfalls passende Ausrüstung mit an Bord zu nehmen. Ein solcher Dienst soll hier konzeptioniert werden.

4.1.1 Anforderungen

Zu den technischen Voraussetzungen zählen vor allem Außenkameras an Fahrzeugen. Diese sind meist hinter der Windschutzscheibe in Fahrtrichtung oder auch am Heck nach hinten gerichtet angebracht. Einige Pkw Modelle haben zusätzlich Kameras, die die Umgebung neben dem Fahrzeug erfassen. Auch die Seitenspiegel zählen zu typischen Installationsorten. Neben Kameras muss auch Hardware vorhanden sein, die leistungsfähig genug ist, die Anfragen zu verarbeiten und den Stream bereitzustellen. Zusätzlich ist eine Internetverbindung erforderlich, wobei die Übertragungsrate hoch genug sein muss, um einen Video-Stream zu ermöglichen. Auch die fahrzeuginterne Kommunikation über die Bussysteme muss dazu in der Lage sein die entstehenden Datenmengen verarbeiten zu können. Dies gilt auch dann wenn die Datenverbindung durch andere Dienste in Anspruch genommen wird. Ebenso muss ein System zur Positionsbestimmung verbaut sein.

Um die Skalierbarkeit des Systems zu gewährleisten, muss das Backend entsprechend leistungsfähig sein. Die großen Datenmengen, die durch viele Video-Streams gleichzeitig entstehen, stellen dabei besondere Anforderungen an die Netzwerkverbindung. Auch muss die Verfügbarkeit des Dienstes besonders hoch sein. Der Dienst stellt also besonders hohe Quality of Service Anforderungen.

Neben den technischen Voraussetzungen müssen auch rechtliche Voraussetzungen gegeben sein. So ist es erforderlich, dass nicht gegen geltende Datenschutzgesetze verstoßen wird. Besonders umstritten könnte dabei die Festlegung der Umstände sein, unter denen auf die Außenkameras privater Fahrzeuge zugegriffen werden kann. Basieren könnte die Befugnis jeweils darauf, ob im Moment des Zugriffs der Schutz der Privatsphäre der Betroffenen dem Ziel des angestrebten Einsatzes überwiegt oder nicht. So könnte beispielsweise immer dann ein Videostream abgerufen werden dürfen, wenn dies dem Zweck dient unmittelbare Lebens-

gefahr abzuwenden. Auch wie eingangs beschrieben könnte die Nutzung des Dienstes in speziell gekennzeichneten Zonen gestattet sein. In jedem Fall aber müsste verhindert werden, dass auf Pkw zugegriffen wird, die sich auf privatem Gelände befinden, oder dass für den Fahrzeugbesitzer Kosten durch die Datenverbindung entstehen.

4.1.2 Stakeholder

Da der Dienst nur von Polizei und Feuerwehr in Anspruch genommen werden könnte, sind dies auch die Hauptnutzergruppen. An der technischen Umsetzung wären zudem die Autohersteller und Mobilfunkanbieter beteiligt. Es ist weiterhin zu beachten, dass es für einen solchen Dienst eine eigene Gesetzgebung geben müsste und dass es gegebenenfalls erforderlich wäre, dass sein Einsatz allgemein akzeptiert wird. Daher zählen auch der Gesetzgeber und die Öffentlichkeit zu den Interessengruppen, womit eine Auflistung isoliert betrachteter Stakeholder zu diesem Zeitpunkt als wenig sinnvoll erscheint.

4.1.3 Konzept

Kernfunktion des Dienstes ist der Zugriff auf die Außenkameras aller Fahrzeuge, die sich in Bereichen aufhalten, in denen ein besonders hohes Gefahrenpotential herrscht oder in denen ein Einsatz stattfindet. Die Nutzer, also Polizei- und Feuerwehrbeamte, würden beispielsweise über eine Desktop-Applikation den Bereich auswählen, den sie einsehen möchten und den Video-Stream starten, sofern der Bereich einsehbar ist. Die Software abstrahiert dabei den Zugriff so, dass nicht bestimmte Fahrzeuge ausgewählt werden können, sondern nur der gewünschte Sichtbereich und Winkel. Sollte die Sicht verdeckt sein, kann ein anderer Blickwinkel ausgewählt werden. Entsprechende Berechnungen werden in der Cloud ausgeführt.

Wird ein Fahrzeug in einer der Zonen, indem der Zugriff auf die Kameras stattfinden kann, abgestellt, so authentifiziert es sich beim Backend und meldet sich beim Dienst an. Hierzu sendet es unter anderem die Anzahl und Ausrichtung der Kameras, sowie die Position und Lage des Fahrzeugs an den Server. Dieser errechnet welche Bereiche aus welchen Winkeln einsehbar sind. Bei Anforderung eines Video-Streams sendet der Server eine Anfrage an das Auto, welches daraufhin den Stream startet.

Sollte eine Person in das Fahrzeug einsteigen, während auf dieses zugegriffen wird, wird dies im Auto auf dem Display angezeigt. Die Person kann dann darüber entscheiden, ob sie losfahren oder stehen bleiben möchte. Bei einer Bewegung des Fahrzeugs bricht der Stream

ab und das Fahrzeug sendet eine entsprechende Nachricht an das Backend. Dem Nutzer wird dann automatisch der beste alternative Video-Stream zu Verfügung gestellt, der vorhanden ist.

4.1.4 Use Case

Tabelle 4.1: Use Case zum Dienst Vehicular CCTV nach UML-Schema

Name	Anforderung auf Einsicht eines Bereiches, indem ein Feuerwehreinsatz stattfinden soll
Akteure	Der Benutzer, abgestellte Pkw
Beschreibung	Bei einem Feuerwehreinsatz soll vor dem Abfahren geprüft werden mit welchen Löschfahrzeugen die Örtlichkeit angefahren werden kann. Hierzu wird eine Einsicht des Bereiches angefordert.
Auslöser	Das Anfordern des Video-Streams durch den Benutzer über eine Desktop-Applikation
Vorbedingungen	Es sind Pkw im Bereich abgestellt, es handelt sich um eine Zone, die zur Überwachung freigegeben ist
Nachbedingungen	Der Benutzer hat den gewünschten Bereich einsehen können
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer wählt in der Desktop-Applikation auf einer digitalen Karte den gewünschten Bereich und den Blickwinkel aus. 2. Ein Video-Bild der Örtlichkeit wird übertragen, 3. Der Benutzer beendet die Übertragung.
Alternative Ablaufschritte	<ol style="list-style-type: none"> 2a. Der Benutzer stellt fest, dass die Sicht verdeckt ist und wählt den nächsten verfügbaren Blickwinkel aus, 2b. Dem Benutzer wird mitgeteilt, dass die momentan verwendete Kamera nicht länger verfügbar ist, da das Fahrzeug bewegt wird. 2c. Es wird ein anderer Video-Stream des Bereiches übertragen.

4.2 Gefahrenstellenerkennung durch Meldung von außergewöhnlichen Fahrmanövern

Bedingt durch die jeweilige Anordnung von Straßen, Gebäuden, Parkplätzen, Radwegen und Fußgängerüberwegen ist die Übersichtlichkeit und damit das Unfallrisiko stets unterschiedlich. Auch die Breite von Straßen, Parkgewohnheiten und typisches Verhalten von Fußgängern ändern das lokale Unfallrisiko. Durch die eigene Erfahrung lernen Fahrer welches Verhalten jeweils angebracht ist. Ebenso werden Warnschilder und Geschwindigkeitsbegrenzungen von Behörden aufgestellt, wenn das Gefahrenpotential bekannt ist. Zur Abschätzung werden unter anderem Unfallstatistiken herangezogen.

Mit Hilfe der Sensorik von Pkw und Lkw lassen sich Fahrmanöver wie Notbremsungen oder schnelles Ausweichen feststellen. Häuft sich dieses Fahrverhalten verschiedener Fahrer in einem Bereich, so kann davon ausgegangen werden, dass dies beispielsweise mit der Straßenführung oder lokalen Parkgewohnheiten zusammenhängt. Entsprechende bauliche Maßnahmen oder Eintragungen in Karten können dann vorgenommen werden. Auch denkbar ist ein Fahrverbot für autonome Fahrzeuge. In diesem Abschnitt soll ein entsprechender Dienst entworfen werden.

4.2.1 Anforderungen

Viele Autos sind aufgrund der Fahrdynamiksysteme mit Beschleunigungs-, Bremsdruck-, und Lenkwinkelsensoren ausgestattet. Mit diesen ist es möglich außergewöhnliche Fahrmanöver festzustellen, wobei eine entsprechende Software hierzu die Daten analysieren muss. Zusätzlich müssen Fahrzeuge über eine Internetverbindung und über ein System zur Positionsbestimmung verfügen. Da der Dienst nicht zeitkritisch ist und pro gemeldetem Fahrmanöver nur wenige Byte übertragen werden müssen, bestehen keine besonders hohen Anforderungen an die Netzwerkverbindung. Es ist jedoch zu beachten, dass potentiell viele Meldungen gemacht werden, was bedeutet, dass der Dienst skalierbar sein muss. Ebenfalls müssen im Backend große Datenmengen gespeichert und Ausgewertet werden.

4.2.2 Stakeholder

Zu den Stakeholdern zählen vor allem Behörden, beziehungsweise Verkehrsämter. Fahrzeughersteller und Mobilfunkanbieter sind jedoch für die technische Umsetzung verantwortlich und somit auch zu nennen. Möglich ist auch, dass Versicherungsunternehmen an der Umsetzung

eines solchen Dienstes Interesse haben, da unter anderem Unfallhergänge besser nachvollzogen werden könnten.

4.2.3 Konzept

Im Wesentlichen besteht das System aus den Fahrzeugen mit entsprechender Hard- und Software, dem Backend und einem Interface, worüber die Daten abgerufen werden können.

Eine Software im Auto überwacht kontinuierlich die Daten der Sensoren der Fahrdynamiksysteme und registriert charakteristische Muster oder die Überschreitung bestimmter Grenzwerte. Wird ein außergewöhnliches Fahrmanöver erkannt, so wird das erkannte Manöver und die Position, sowie der Fahrzeugtyp und die zuvor gefahrene Geschwindigkeit an das Backend übertragen. Der Vorgang ist für den Fahrer transparent.

Im Backend werden alle Meldungen in einer Datenbank gespeichert und in festgelegten Zeitintervallen analysiert. Die Art und Anzahl der Meldungen werden aufbereitet, um in einer Karte dargestellt zu werden. Sollte eine Häufung an einer Position erkannt werden, wird dies gesondert gespeichert.

Der Zugriff auf die Daten erfolgt beispielsweise über ein Webinterface. Dem Nutzer wird eine Karte angezeigt, auf welcher die Anzahl und Art der Fahrmanöver eingetragen sind. Ebenso werden die Positionen angezeigt, bei denen besonders viele Vorfälle gemeldet wurden. Weiterhin besteht die Möglichkeit Details über die einzelnen Meldungen angezeigt zu bekommen.

4.3 Abholdienst Car-Collect für autonome Fahrzeuge

Autonome Fahrzeuge könnten in Zukunft zum Alltag gehören. Bereits heute arbeiten viele Autohersteller daran diese Vision in die Realität umzusetzen. Neben der reinen Nutzung von Autos als Personentransportmittel ermöglicht der Selbstfahrbetrieb viele neue Funktionen. So könnten private Pkw künftig selbstständig zu Geschäften fahren, um dort Ware abzuholen, die zuvor online bestellt wurde. Angestellte hätten hierzu die Möglichkeit den Kofferraum zu öffnen, um die Artikel hineinzulegen. Im Folgenden wird ein solcher Dienst entwickelt.

4.3.1 Anforderungen

Voraussetzung für den Betrieb des Dienstes ist, dass Autos technisch und auch rechtlich in der Lage dazu sind sich autonom und unbemannt auf Straßen fortzubewegen. Es ist davon auszugehen, dass ein solches Fahrzeug ohnehin über eine Internetverbindung verfügen würde, was ebenfalls eine Voraussetzung wäre. Zusätzlich müsste es möglich sein, online Berechtigungen zum Öffnen des Kofferraums zu vergeben. Zum Überwachen des Beladevorgangs könnte auch eine Kamera im Kofferraum angebracht werden.

Abgesehen von den Anforderungen an die Pkw müssten Geschäfte entsprechend vorbereitet sein. Hierzu muss es einen gesonderten Bereich zum Beladen oder eine Durchfahrt geben. Gegebenenfalls ist auch ein Leitsystem erforderlich, um die Autos zu den richtigen Beladepätzen zu führen.

An das Backend sind keine besonderen Anforderungen zu stellen, da der Dienst sich in dieser Hinsicht nur unwesentlich von heutigen Online-Shops unterscheidet. Dementsprechend würde auf eine für diese Zwecke eingesetzte Hard- und Software zurückgegriffen werden.

4.3.2 Stakeholder

Zu den Stakeholdern zählen vor allem die Betreiber von Geschäften, die den Dienst anbieten könnten. Zusätzlich könnten auch Autohersteller an der technischen Umsetzung beteiligt sein.

4.3.3 Konzept

Am Betrieb des Dienstes sind verschiedene Akteure beteiligt, die vom Backend koordiniert werden. Dies sind zum einen die Fahrzeuge der Besteller, zum anderen sind es die Angestellten des Geschäfts, die die Fahrzeuge beladen und die Bestellungen zusammenstellen.

Wenn ein Geschäft den Dienst Car-Collect anbietet, wird dies dem Käufer als Option im Online-Shop angezeigt. Bei Abschluss einer Bestellung wird der Standort eines Geschäfts mit der geringsten Distanz zum Wohnort ermittelt, wobei geprüft wird, ob hier alle Artikel verfügbar sind. Den Mitarbeitern wird die Bestellliste übermittelt, wobei je nach Auslastung eine Zeit geschätzt wird, zu welcher die Bestellung abgeholt werden kann. An das Auto des Käufers wird die Position des Ladens und die Abholzeit übermittelt. Dieses fordert den Käufer, beispielsweise über eine Smartphone-App, dazu auf den Auftrag zu bestätigen. Ebenso wird die Abfahrtzeit geschätzt. Ist diese erreicht und wurde der Auftrag bestätigt, so fährt das Auto zur entsprechenden Adresse. Hier wird es durch ein Leitsystem entweder zu einem Beladeplatz oder in eine Durchfahrt geführt. Der genaue Beladeplatz beziehungsweise die Route auf dem Gelände wird dabei online übertragen. Wenn das Auto am gewünschten Platz ist, wird der Kofferraum elektronisch geöffnet, von einem Mitarbeiter beladen und von diesem wieder verschlossen. Der Mitarbeiter bestätigt anschließend den Abschluss des Auftrags, woraufhin das Auto zum Wohnort des Bestellers zurückfährt. Während der einzelnen Vorgänge kann der Käufer den Status über den Online-Shop abrufen.

4.3.4 Use Cases

Tabelle 4.2: Use Case zum Dienst Car Collect (Einkauf mit Car-Collect) nach UML-Schema

Name	Einkauf mit Car-Collect
Akteure	Der Käufer, der Pkw des Käufers
Beschreibung	Der Käufer gibt online eine Bestellung auf und lässt diese von seinem Auto abholen.
Auslöser	Das Aufgeben der Bestellung
Vorbedingungen	Das Auto kann ungehindert über öffentliche Verkehrswege vom Wohnort des Käufers zum Geschäft und wieder zurück gelangen.
Nachbedingungen	Das Auto des Käufers ist mit der Bestellten Ware im Kofferraum am Wohnort des Käufers.
Standardablauf	<ol style="list-style-type: none"> 1. Der Käufer stellt im Online-Shop des Geschäfts seine Bestellung zusammen. 2. Der Käufer schließt den Auftrag ab und wählt die Option Car-Collect aus. 3. Über eine entsprechende Smartphone-App, wird der Käufer dazu aufgefordert den Auftrag und die Zeit, in der sein Auto die Bestellung abholt, zu bestätigen. 4. Das Auto des Käufers fährt zum Geschäft. 5. Das Auto des Käufers kehrt vom Geschäft wieder. 6. Der Käufer erhält eine Benachrichtigung über die Rückkehr seines Autos auf seinem Smartphone.
Alternative Ablaufschritte	<ol style="list-style-type: none"> 3a. Der Käufer steigt in sein Auto, fährt los und schiebt damit den Abholauftrag auf. 3b. Der Käufer kehrt zurück und veranlasst über die Smartphone-App die Wiederaufnahme des Abholauftrags.

Tabelle 4.3: Use Case zum Dienst Car-Collect (Abfertigen einer Bestellung) nach UML-Schema

Name	Abfertigen einer Bestellung
Akteure	Der Mitarbeiter, das Auto des Käufers
Beschreibung	Der Mitarbeiter fertigt eine Bestellung ab, indem er das Auto des Käufers belädt.
Auslöser	Der Auftrag wird dem Mitarbeiter zugeteilt.
Vorbedingungen	Das Auto kann ungehindert über öffentliche Verkehrswege vom Wohnort des Käufers zum Geschäft und wieder zurück gelangen.
Nachbedingungen	Das Auto des Käufers ist mit der bestellten Ware beladen.
Standardablauf	<ol style="list-style-type: none"> 1. Dem Mitarbeiter wird die Bestelldatei digital auf seinen Hand-Held-Computer übermittelt. 2. Der Mitarbeiter sucht im Geschäft die entsprechenden Artikel aus. 3. Der Mitarbeiter bringt die Artikel zum angegebenen Beladeplatz. 4. Das Auto des Käufers trifft ein. 5. Über seinen Hand-Held-Computer öffnet der Mitarbeiter den Kofferraum des Autos. 6. Der Mitarbeiter belädt den Kofferraum. 7. Der Mitarbeiter verschließt den Kofferraum. 8. Der Mitarbeiter bestätigt den Abschluss des Auftrags über seinen Hand-Held-Computer. 9. Das Auto des Käufers verlässt den Beladeplatz.
Alternative Ablaufschritte	<p>Alternative 1:</p> <ol style="list-style-type: none"> 3a. Dem Mitarbeiter wird über seinen Hand-Held-Computer mitgeteilt, dass der Auftrag verschoben wurde. 3b. Der Mitarbeiter bringt die Artikel in ein Zwischenlager. <p>Alternative 2:</p> <ol style="list-style-type: none"> 1a. Dem Mitarbeiter wird über seinen Hand-Held-Computer mitgeteilt, dass ein aufgeschobener Auftrag wiederaufgenommen wurde. 2a. Der Mitarbeiter holt die Bestellung aus dem Zwischenlager.

4.4 Effiziente Parkplatznutzung

Besonders in Städten sind oft zu wenige Parkplätze vorhanden, um den Bedarf zu decken. Dies gilt sowohl für die Innenstädte, als auch für Wohngebiete. Die Parkplatzsuche kostet Zeit und erhöht die Umweltbelastung, da umher gefahren werden muss bis eine passende Stelle zum Abstellen des Fahrzeugs gefunden ist. Eine Erhöhung der Anzahl der Parkplätze ist meist nicht realisierbar, da hierfür freier Raum fehlt. Die vorhandenen Parkplätze könnten jedoch durch eine effizientere Nutzung mehr Fahrzeuge aufnehmen. Dies gilt für Parkbuchten und Parkflächen am Straßenrand, wo Autos ohne feste Positionen fließend hintereinander abgestellt werden. Durch die unterschiedlichen Größen der Autos und durch verschiedene Ankunfts- und Stehzeiten entstehen zwischen den Fahrzeugen Lücken, die nicht nutzbar sind. In Anbetracht des Aufkommens autonomer Fahrzeuge ist denkbar, dass Pkw und Lkw künftig automatisch aufrücken, um so zusammenhängende Lücken zu schaffen, die groß genug sind, um weitere Autos aufzunehmen. Das System könnte ebenso dazu dienen bei zu wenig Platz zum Ausparken genügend Freiraum vor oder hinter dem Auto zu schaffen. So könnten Fahrzeuge besonders dicht hintereinander geparkt werden. Ein entsprechender Dienst soll hier entworfen werden.

4.4.1 Anforderungen

Die Grundvoraussetzung für einen solchen Dienst ist, neben dem Vorhandensein von Systemen zur Abstandsmessung vor und hinter dem Fahrzeug, der autonome Betrieb. Bereits heute gibt es zahlreiche Pkw und auch Nutzfahrzeuge, die autonom einparken können. Da das Aufrücken in Parkbuchten lediglich ein Vor- und Zurücksetzen erfordert, wäre die technische Anforderung für die Umsetzung des Dienstes in dieser Hinsicht bereits erfüllt. Rechtlich ist dies jedoch bislang nicht der Fall, da ein Fahrer am Steuer sitzen muss, um wenn nötig einzuschreiten. Für diesen Dienst müssen sich Autos allerdings ohne Fahrer fortbewegen können.

Weiterhin müssen die Autos über eine Internetverbindung verfügen, um Steuerbefehle entgegen zu nehmen und gemessene Abstände vor und hinter dem Fahrzeug in die Cloud zu senden. Denkbar ist in diesem Zusammenhang auch eine Selbstorganisation der Fahrzeuge, wofür lediglich eine Punkt-zu-Punkt-Verbindung erforderlich wäre. Dies hätte den Vorteil, dass der Dienst unabhängig von der Internetverbindung arbeiten könnte. Nachteilig ist jedoch zu bewerten, dass die Selbstorganisation unter Umständen leichter zu manipulieren wäre, da eine Authentifizierung der Fahrzeuge beim Backend nicht stattfinden würde.

4.4.2 Stakeholder

Da die Autohersteller für die technische Umsetzung des Dienstes verantwortlich wären, sind diese auch als Hauptinteressengruppe zu nennen. Zusätzlich könnten Stadtverwaltungen oder Verkehrsämter beteiligt sein, da diese eventuell bestimmen müssten, auf welchen Parkplätzen der Dienst Anwendung finden darf.

4.4.3 Konzept

Grundsätzlich erfüllt der Dienst zwei Aufgaben. Zum einen werden parkende Autos aufgerückt, um ungenutzte Parkfläche freizugeben, zum anderen werden Autos bewegt, um Freiraum zum Ausparken zu schaffen.

Die Bereiche, in denen Aufgerückt werden soll, sind als Schicht in der Navigationskarte realisiert. Wird ein Auto in einem solchen Bereich abgestellt, so meldet es sich für den Dienst beim Backend an. Einmalig wird der Abstand nach vorn und nach hinten gemessen und an den Server übermittelt. Dieser sendet an alle anderen Fahrzeuge in dem Bereich eine Anfrage zum Messen der Abstände. Aus den Daten wird dann der ungenutzte Freiraum errechnet. Sollte dieser groß genug sein, damit ein weiteres Auto abgestellt werden kann, wird errechnet welche Autos nach vorn und welche nach hinten bewegt werden müssen. Ebenso wird die Reihenfolge ermittelt, in der die einzelnen Aktionen durchgeführt werden müssen. Bei der Erstellung des Ablaufplans werden auch Daten wie die Antriebsart der Autos, ob ein Gefälle besteht und wie oft die Autos bereits bewegt wurden, sowie der Kraftstoffstand miteinbezogen. Dem Plan entsprechend wird den Autos nacheinander der Befehl gegeben den Abstand zum nächsten Hindernis nach vorn oder nach hinten auf einen definierten Mindestabstand zu reduzieren. Hierbei wird eine Lücke offengelassen, wodurch ein Ausparken ermöglicht wird. Tauchen unerwartet neue Hindernisse auf oder ist ein Fahrzeug blockiert, wird der Vorgang abgebrochen.

Den einzelnen Fahrzeugbesitzern wird über eine Smartphone-App mitgeteilt, dass ihr Auto bewegt wurde, um Freiraum zu schaffen. Über die App haben sie die Möglichkeit anzukündigen wann sie ihr Auto ausparken möchten, damit die freigelassene Lücke vor ihrem Fahrzeug positioniert werden kann. Über eine In-Car-App kann der Vorgang auch sofort initiiert werden. Der Server errechnet hierzu erneut einen Ablaufplan, nachdem alle Fahrzeuge bewegt werden müssen und arbeitet diesen entsprechend ab. Kollidieren die Zeiten zweier Fahrzeugbesitzer, so wird der zuerst eingetroffene Wunsch priorisiert. Ein begonnener Vorgang kann durch den Wunsch nach sofortigem Ausparken eines anderen Fahrers nicht abgebrochen werden. Ist

4 Beispiele für Neuentwicklungen

genügend Freiraum vorhanden, um mehr als einem Fahrer das Ausparken zu ermöglichen, so werden entsprechend mehrere Lücken freigelassen und positioniert.

5 Entwicklung eines Dienstes zum Erfassen der Luftgüte entlang der Fahrbahn

5.1 Motivation

Luftverschmutzung ist aktuell ein viel diskutiertes Thema. Emissionen, die durch Industrieanlagen, Kraftwerke, Abfallverwertungsanlagen und den Güter- und Personentransport entstehen belasten die Gesundheit vieler Millionen Menschen weltweit. Besonders Städte in Niedriglohnländern leiden unter Luftverschmutzung. Laut WHO [14] würden 98% der Städte mit mehr als 100.000 Einwohnern in diesen Ländern die Richtlinien der Organisation überschreiten. In Hochlohnländern seien es immer noch 56%. Eine Schätzung der WHO habe auch ergeben, dass im Jahr 2012 Weltweit 3 Millionen frühzeitige Tode auf Luftverschmutzung zurückzuführen seien. Auch in Deutschen Städten wie Stuttgart oder Hamburg werden Grenzwerte regelmäßig überschritten, wie anhand der Überschreitungstabelle des Umweltbundesamtes zu sehen ist [27]. Die Emissionen entstehen dabei vor allem durch den Kfz-Verkehr. In Hamburg entstehen zusätzlich durch Container- und Kreuzfahrtschiffe große Belastungen.

Besonders Feinstäube mit Durchmessern von maximal 10 Mikrometern (PM^{10}) beziehungsweise maximal 2,5 Mikrometern ($PM_{2.5}$) würden laut Weltgesundheitsorganisation [14] eine große Belastung für die Gesundheit darstellen. Sie trügen zur Entstehung von kardiovaskulären und respiratorischen Krankheiten bei und förderten das Lungenkrebsrisiko. Weiterhin sei das Gas Ozon in Bodennähe eine Ursache für Lungenkrankheiten und eine Abnahme der Lungenfunktion, sowie ein Auslöser von Asthma. Es sei einer der Hauptbestandteile fotochemischen Smogs, welcher durch das Einwirken des Sonnenlichts unter anderem auf Stickoxide und flüchtige organische Verbindungen, sogenannte VOC^2 , entstünde. Stickstoffdioxid, welches zu den Stickoxiden gehört, fördere besonders bei Kindern die Symptome von Asthma. Des Weiteren würde Schwefeldioxidgas die Augen und Atemwege reizen und in Verbindung mit Wasser Schwefelsäure bilden. Diese sei der Hauptbestandteil von saurem Regen, welcher eine

¹Particulate Matter

²Volatile Organic Compounds

Ursache für die Zerstörung von Wäldern sei.

Es gibt bereits zahlreiche Open-Source-Projekte mit dem Ziel die Luftverschmutzung zu messen und zu dokumentieren, um die Menschen für das Thema zu sensibilisieren und auf das Problem aufmerksam zu machen. Die Ergebnisse der Projekte sind meist öffentlich im Internet einsehbar und die eingesetzten Messgeräte frei erhältlich. Die Smart Citizen Plattform [48] bietet beispielsweise die Möglichkeit selbst gemessene Daten auf einer Karte einzutragen, die online eingesehen werden kann. Die Messgeräte verfügen über Sensoren und eine WLAN-Verbindung und können direkt über die Webseite des Projekts erworben werden. Sie messen unter anderem Kohlenmonoxid und Stickstoffdioxid. Ein Projekt zur Messung von Feinstaubbelastung wird derzeit vom OK Lab Stuttgart betrieben [40]. Auf der Webseite stehen unter anderem Bauanleitungen für Messgeräte zur Verfügung. Die erfassten Daten können auch hier über das Internet an entsprechende Server gesendet werden, wo sie in einer Karte visualisiert werden. Die AirCasting Plattform [26] bietet eine ähnliche Funktionalität. Die Messgeräte, die unter anderem Feinstaub, Kohlenmonoxid und Stickstoffdioxid messen, können auch hier online erworben werden. Anders als die erst genannten Geräte, die über WLAN ans Internet verbunden werden, müssen diese mit einem Smartphone gekoppelt werden, um Daten zu senden.

Auch Google betreibt mit seinen Street View Autos in ausgewählten Städten Luftqualitätsmessungen [25]. Im Gegensatz zu den oben genannten Messgeräten liefern die hier eingesetzten jedoch deutlich präzisere Ergebnisse. Weiterhin repräsentieren die Daten, die durch die Messung auf der Straße beim Entlangfahren erfasst werden, die Schadstoffbelastung für Fußgänger, sowie Rad- und Autofahrer. Die Daten, die im Rahmen der Open-Source-Projekte erfasst werden, stammen meist von vereinzelt Sensoren, die an unbekannter Stelle angebracht sind. Es ist daher nicht eindeutig bestimmbar inwiefern sie die lokale Luftqualität widerspiegeln, zumal nicht ausgeschlossen werden kann, dass die Messungen teilweise in Innenräumen stattfinden. Sinnvoller erscheint daher der Ansatz die Luftgüte beim Autofahren zu erfassen.

In vielen Fahrzeugen wird eine solche Messung bereits vom Fahrer unbemerkt von der automatischen Klimaanlage durchgeführt. Luftgütesensoren messen bestimmte Gaskonzentrationen und schalten den Umluftbetrieb ein, sobald ein Grenzwert überschritten wurde, was besonders in Tunneln oder im Stau der Fall sein kann. Bislang werden lediglich Gase wie Stickoxide oder Kohlenmonoxid gemessen [52]. Es ist jedoch denkbar, dass in Zukunft auch die Feinstaubbelastung erfasst wird. So hat beispielsweise der Sensorhersteller Paragon entsprechende

Feinstaubsensoren speziell für den Einsatz in Fahrzeugen entwickelt [45]. Die Daten der Sensoren könnten kontinuierlich ausgelesen und an ein entsprechendes Backend gesendet werden, um die Luftqualität so zu kartografieren.

In dieser Arbeit wird ein fahrzeugunabhängiger Prototyp entwickelt, dessen Daten an einen Server zur zentralen Auswertung und Darstellung in einer Karte durch ein Geoinformationssystem übertragen werden. Das System wird dabei mit stetem Bezug zum Einsatz im Fahrzeug und die Nutzung typischer Fahrzeugsensorik entworfen und umgesetzt. Daher werden die einzelnen Komponenten so gewählt, dass ihre Eigenschaften mit denen der im Fahrzeug verbauten Hardware vergleichbar sind. Obwohl dieser Bezug gegeben ist, ist der generelle Einsatz eines fahrzeugunabhängigen Messgerätes durchaus denkbar. So würde vermieden werden, dass Autohersteller die Hoheit über die gesammelten Daten haben. Beispielsweise könnten städtische Fahrzeugflotten, wie etwa die Einsatzwagen der Feuerwehr, mit entsprechenden Geräten ausgestattet werden. In dieser Arbeit erfolgt die Aufnahme von Messwerten nur mit einem Gerät, wobei der Einsatz in einem crowdsourcing-basierten System angedacht ist.

In den folgenden Abschnitten werden zunächst Stakeholder identifiziert und Anforderungen aufgestellt, die bezüglich der Verwendung von typischer Fahrzeugsensorik bewertet werden. Anschließend werden nach Auswahl der Komponenten Hardware, Software und Gehäuse des Prototyps entworfen, sowie eine passende Serverplattform ausgewählt. In den darauffolgenden Abschnitten werden dann die Umsetzung und die Aufnahme von Messwerten beschrieben und die Ergebnisse analysiert, wobei anzumerken ist, dass es sich bei der Messung um eine Vergleichsmessung handelt. Abschließend wird eine Methode zur Absolutwertbestimmung und Kalibrierung vorgestellt.

5.2 Stakeholder

Zu den Stakeholdern zählen vor allem Umweltbehörden, da diese besonders in urbanen Gebieten Schadstoffbelastungen in der Luft messen, um die Einhaltung gesetzlicher Richtlinien zu überwachen. Auch private Institutionen und Umweltschutzorganisationen könnten von den erfassten Daten Gebrauch machen. Möglich ist auch, dass Wohnungsunternehmen eine Interessengruppe darstellen, da die lokale Luftqualität den Wohnungsmarkt beeinflussen könnte. Mit Bezug auf die Nutzung der Luftgütesensoren der Klimaanlage im Fahrzeug ist es weiterhin denkbar, dass die Autohersteller an der technischen Umsetzung beteiligt sind.

5.3 Analyse der Problemstellung

5.3.1 Schadstoffbelastungen in der Luft

Industrieanlagen, Kraftwerke, Fahrzeuge und landwirtschaftliche Betriebe stoßen täglich große Mengen an Schadstoffen aus, die Einfluss auf Tiere, Menschen und Pflanzen nehmen. Welche Schadstoffe dies sind, welche Auswirkungen sie haben, wer die Hauptemittenten sind und welche Grenzwerte existieren wird im Folgenden beschrieben. Dabei basieren die Ausführungen auf den Angaben des Umweltbundesamtes [18].

Zu den Schadstoffen in der Luft zählen Kohlenmonoxid, Stickstoffdioxid, Benzol, Ammoniak, Schwefeldioxid, Ozon und Feinstaub. Ammoniak trägt vor allem zur Eutrophierung und Versauerung von Böden und Gewässern bei, wodurch Schäden an der Vegetation entstehen und sich die Artenvielfalt verändert. Die Hauptemittenten sind landwirtschaftliche Betriebe. Besonders die Rinderhaltung, beziehungsweise die Lagerung von Wirtschaftsdünger, trägt zur Ammoniakbelastung bei. Grenzwerte existieren bislang nicht, jedoch gilt laut einer EU-Richtlinie eine Konzentration von unter $10 \mu\text{g}/\text{m}^3$ als unbedenklich für die Vegetation.

Benzol, welches Bestandteil der Abgase von Fahrzeugen ist, führt bei langfristiger Aufnahme zur Schädigung von Knochenmark und Organen. Außerdem sind die Abbauprodukte des Stoffes krebserregend. Im Jahresmittel darf der europaweite Grenzwert von $5 \mu\text{g}/\text{m}^3$ nicht überschritten werden.

Auch die Kohlenmonoxidbelastung entstammt überwiegend dem Straßenverkehr. Es ist ein starkes Atemgift und hat Auswirkungen auf das zentrale Nervensystem. Der höchste 8-Stunden-Mittelwert eines Tages darf den Grenzwert von $10 \mu\text{g}/\text{m}^3$ nicht überschreiten.

Ozon in Bodennähe wird nicht direkt freigesetzt, sondern entsteht durch Sonneneinstrahlung auf Vorläuferschadstoffe, wie Stickstoffdioxid und flüchtige organische Verbindungen (VOC), wozu auch Benzol zählt. Es verursacht entzündliche Reaktionen der Atemwege und führt zu einer Verminderung der Lungenfunktion. Dies ist besonders für Asthmatiker schädlich. Neben Menschen und Tieren können auch die Blattorgane von Pflanzen geschädigt werden. Ab einer Konzentration von $180 \mu\text{g}/\text{m}^3$ über einen Zeitraum von einer Stunde sollten besonders empfindliche Personen Aktivitäten im Freien meiden. Die Alarmschwelle, bei der generell Aktivitäten im Freien gemieden werden sollten, liegt bei $240 \mu\text{g}/\text{m}^3$.

Volatile Organic Compounds, also flüchtige organische Verbindungen, entstammen zum Teil der Verbrennung von fossilen Brennstoffen, aber auch der Verwendung von Lösemitteln. VOC werden aber auch zu einem großen Teil von Lebewesen emittiert.

Stickstoffdioxid wird hingegen überwiegend durch Verbrennungsmotoren und Feuerungsanlagen ausgestoßen. Es ist schädigend für Pflanzen und auch für Menschen mit Vorerkrankungen der Lunge. Europaweit gilt ein Grenzwert von $200 \mu\text{g}/\text{m}^3$, der nicht öfter als 18-mal im Jahr überschritten werden darf. Der Jahresmittelgrenzwert liegt bei $40 \mu\text{g}/\text{m}^3$.

Schwefeldioxid, welches bei der Verbrennung von Kohle und Öl freigesetzt wird, führt bei Menschen und Tieren zu Reizungen der Atemwege und Augen. Vor allem wirkt es aber schädigend auf Ökosysteme, da Böden und Gewässer versauern. Außerdem tragen die Sulfatpartikel zur Feinstaubbelastung bei. Es gilt ein europaweiter Tagesgrenzwert von $125 \mu\text{g}/\text{m}^3$, der nicht öfter als 3-mal im Jahr überschritten werden darf. Der 1-Stunden Grenzwert liegt bei $350 \mu\text{g}/\text{m}^3$ und darf höchstens 24-mal im Jahr überschritten werden. In Deutschland liegt die Belastung jedoch deutlich unterhalb dieser Grenzen.

Feinstäube sind ein komplexes Gemisch aus flüssigen und festen Partikeln, bei denen unterschieden wird zwischen PM10 (Partikel kleiner als 10 Mikrometer) und PM2.5 (Partikel kleiner als 2,5 Mikrometer). Sie entstehen bei der Verbrennung in Motoren, in Öfen und Kraftwerken, sowie bei der Stahlerzeugung oder beim Umschlagen von Schuttgütern. PM10 Feinstaub dringt beim Menschen bis in die Nasenhöhle vor und kann dort Reizungen verursachen. PM2.5 Feinstaub kann bis in die Bronchien, das Lungengewebe und den Blutkreislauf vordringen, wodurch eine erhöhte Thromboseneigung hervorgerufen wird. Ebenso kann Einfluss auf die Regulierungsfunktion des Vegetativen Nervensystems genommen werden. Europaweit gilt ein Tagesgrenzwert von $50 \mu\text{g}/\text{m}^3$, der nicht öfter als 35-mal im Jahr überschritten werden darf. Im Jahresmittel darf ein Wert von $40 \mu\text{g}/\text{m}^3$ nicht überschritten werden. Der einzuhaltende Jahresmittelwert für PM2.5 liegt bei $20 \mu\text{g}/\text{m}^3$.

Zusammengefasst bedeutet dies, dass im Straßenverkehr und in der Nähe von Industrieanlagen oder Kraftwerken vor allem Kohlenmonoxid, Feinstäube und Ozon zu messen sind, wobei auch die Ozon-Vorläuferschadstoffe Stickstoffdioxid und flüchtige organische Verbindungen, wie Benzol, alternativ als Indikator gemessen werden können. In ländlichen Gebieten könnte weiterhin Ammoniak eine Größe sein, die es zu erfassen gilt. Hierauf basierend werden die Anforderungen an die Sensorik definiert.

5.3.2 Anforderungen an die Sensorik

Grundlegend gilt es mit Sensoren den Anteil der Gase Kohlenmonoxid, Stickstoffdioxid, Ammoniak und flüchtige Organische Verbindungen, darunter Benzol und Toluol, in der Luft zu erfassen. Zusätzlich soll die Konzentration an Feinstaub, also PM10 und PM2.5, erfasst werden. Des Weiteren muss im Kontext des Gesamtsystems die Position bestimmt werden können, an der Schadstoffkonzentrationen gemessen werden.

Der Erfassungsbereich der Sensoren soll den Bereich der in der Luft vorkommenden Schadstoffkonzentrationen mindestens bis zu einem Wert abdecken, der für Menschen über längere Zeit als Gesundheitsgefährdend eingestuft wird. Dies bedeutet konkret, dass der Erfassungsbereich laut National Institute for Occupational Safety and Health [57] für Kohlenmonoxid bei 0 bis 400 ppm, für Stickstoffdioxid bei 0 bis 10 ppm, für Ammoniak bei 0 bis 100 ppm und Benzol bei 0 bis 50 ppm liegen soll. Hierbei beziehen sich die Werte jeweils auf eine Aussetzungsdauer von einer Stunde. Für die Messung der Feinstaubbelastung soll basierend auf Richtlinien der WHO [14] jeweils der Bereich 0 bis $150 \mu\text{g}/\text{m}^3$ erfassbar sein. Da mittels Crowdsourcing die Daten vieler Hundert oder Tausend Sensoren erfasst werden sollen, über welche im Anschluss gemittelt wird, sind keine besonderen Anforderungen an die Genauigkeit der Sensoren zu stellen, wobei sich dies auf zufällige Abweichungen bezieht.

Ein weiterer Aspekt ist die Reaktionszeit der Sensoren. Es ist davon auszugehen, dass von Fahrzeugen während der Fahrt ausgestoßene Emissionen nicht lokal in einem Bereich von wenigen Metern verbleiben, sondern durch die Bewegung der Autos und durch Wind verteilt werden. Es soll daher genügen mindestens alle 500 Metern eine Messung durchzuführen. Daraus lässt sich eine maximale Reaktionszeit der Sensoren von 35 Sekunden ableiten, wobei sich dies auf eine Geschwindigkeit von 50 km/h bezieht. Bei geringerer Geschwindigkeit findet entsprechend in geringeren Abständen eine Messung statt. Aufgrund der angenommenen Verteilung der Luft soll hier für die Positionsbestimmung eine Genauigkeit von 50 Metern als ausreichend betrachtet werden.

Da das Gesamtsystem inklusive der Sensoren vom Bordsystem eines Fahrzeugs mit Energie versorgt wird und davon auszugehen ist, dass die Leistungsaufnahme sich nicht in einem Bereich bewegen wird, welcher das Bordnetz nennenswert belastet, sind bezüglich der Leistungsaufnahme keine besonderen Anforderungen zu stellen.

5.3.3 Anforderungen an die Signalverarbeitungskette

Bezüglich der Aufbereitung der Sensordaten sind die Aspekte Digitalisierung, Datenübertragung und Werteumrechnung zu berücksichtigen.

Da davon auszugehen ist, dass die Sensoren sowohl über analoge als auch digitale Schnittstellen verfügen können, kann es notwendig sein, dass eine Analog-Digital-Wandlung vorgenommen werden muss. In diesem Fall erscheint eine 8-Bit Quantisierung als sinnvoll, da eine Unterteilung der Messbereiche in 256 Stufen für den gegebenen Anwendungsfall ausreichend ist.

Die Hardware zur Berechnung der Messwerte muss zum einen in der Lage sein über die entsprechenden Bussysteme die Kommunikation mit den Sensoren herzustellen, als auch die Rohdaten der Sensoren anhand der Kennlinien in Messwerte umzurechnen und gegebenenfalls Störeinflüsse rechnerisch zu kompensieren. Weiterhin müssen die aufbereiteten Daten an ein Übertragungsmodul übermittelt werden, von wo aus sie zu einem Server gesendet werden. Genaue Anforderungen an die Berechnungshardware können an dieser Stelle ohne konkrete Sensortypen jedoch nicht aufgestellt werden.

5.3.4 Anbindung an ein Geoinformationssystem

Bei der Analyse der Anbindung an ein Geoinformationssystem sind die Aspekte Datendurchsatz und Quality of Service, Verschlüsselung, zentrale Verarbeitung, Speicherung, sowie die Skalierbarkeit des Systems zu betrachten.

Da pro Fahrzeug in einem Messintervall nur wenige Bytes übertragen werden müssen, ist davon auszugehen, dass die Daten über das bestehende Mobilfunknetz an einen Server übertragen werden können. Selbst unter Berücksichtigung jeweils zu übertragener Authentifizierungsschlüssel und Prüfsummen ist auszuschließen, dass die notwendige Übertragungsrate in einem Bereich von mehr als einem Kilobyte pro Sekunde liegt, was sich deutlich unter den heute üblichen Übertragungsraten im Mobilfunk befindet.

Hohe Latenzen, Paketverluste oder kurzzeitiger Verbindungsabbruch stellen keine potentielle Quelle für Fehler dar, da keine Echtzeitanforderungen bestehen. Auch dauerhaft ausfallende Verbindungen werden durch die große Anzahl an Klienten kompensiert. Durch Prüfsummen muss jedoch sichergestellt werden, dass Daten korrekt übertragen wurden, was durch entsprechende Protokollschichten sichergestellt werden muss.

Auch zu berücksichtigen ist, dass Klienten sich gegenüber dem Server Authentifizieren müssen, um Manipulation auszuschließen. Möglich ist hier auch der Einsatz einer Verschlüsselung.

Die zum Server gesendeten Daten müssen zum einen zuverlässig in einer Datenbank gespeichert, zum anderen aber auch analysiert und verarbeitet werden, um beispielsweise zeitliche Verläufe und Mittelwerte zu berechnen. Der Server muss daher entsprechend Leistungsfähig sein und über eine angemessene Netzwerkanbindung verfügen. Wegen der starken Abhängigkeit von der Anzahl der Klienten erscheint es hier nicht als sinnvoll genaue Anforderungen diesbezüglich zu definieren.

Ein weiterer wichtiger Aspekt ist die Skalierbarkeit des Systems. Es muss davon ausgegangen werden, dass mehrere hunderttausend Fahrzeuge am Dienst teilnehmen und Daten bereitstellen. Die Zahl der aktiven Nutzer kann zum Beispiel zwischen Tag und Nacht sehr stark variieren. Daher muss das System zwingend hochskalierbar sein. Die Hard- und Software des Servers muss dementsprechend ausgelegt werden.

5.3.5 Bewertung der Anforderungen bezüglich der Verwendung der Fahrzeugsensorik und Bordsysteme

Wie eingangs beschrieben sind Luftgütesensoren ein Bestandteil der automatischen Klimaanlage. Laut Volkswagen Techniklexikon [52] erfassen diese oxidierbare Gase, wie Kohlenmonoxid und Benzoldämpfe, und reduzierbare Gase, wie Stickoxide. Aus den Angaben in Pkw-Klimatisierung 4, von Dieter Schlenz, [3, S. 67] geht hervor, dass Luftgütesensoren in automatischen Klimaanlage meist aus mehreren Sensorelementen auf Basis von Halbleitertechnologie bestehen. Diese verändern in Abhängigkeit der Konzentration der genannten Gase ihren Widerstand. Die Sensorsysteme haben bereits eine eigene Auswertlogik mit Mikroprozessoren integriert mit der sie die Außenluft zu einer Qualitätsstufe zuordnen. Es ist daher denkbar, dass diese Art Sensorsystem mit entsprechender Modifikation für die crowdsourcing-basierte Erfassung der Luftgüte geeignet ist. Da der Sensorhersteller Paragon bereits Partikelsensoren speziell zur Erfassung von Feinstäuben für Fahrzeuge anbietet, soll eine Bewertung der Eignung hier nicht stattfinden.

Technisch ist es möglich ein Steuergerät zur Auswertung der Sensorsignale im Fahrzeug zu installieren, zumal die genannten Sensorsysteme bereits über eine eigene Auswertlogik verfügen. Zu beachten ist jedoch, dass dies mit zusätzlichen Kosten verbunden ist. Daher ist

dies vielmehr aus wirtschaftlicher Perspektive zu betrachten. Möglich ist aber auch, dass die Daten auf bereits bestehender Hardware, wie dem Infotainment-System verarbeitet werden. Zur Datenübertragung an den Server eignen sich die in neueren Fahrzeugen installierten Connectivity-Units mit Mobilfunkanbindung. Eine auch von anderen Diensten im Auto oft genutzte Alternative ist die Kopplung an ein Smartphone. Somit ist in einem ersten Ansatz anzunehmen, dass der Dienst potentiell auch in Serienfahrzeuge integrierbar ist. Dies hängt jedoch vom Nutzen und wirtschaftlichen Aspekten ab.

5.4 Konzeption eines fahrzeugunabhängigen Prototyps

In diesem Abschnitt wird ein fahrzeugunabhängiger Prototyp zur Erfassung der Luftgüte konzipiert, der als Beispiel für eine konkrete Realisierung des Dienstes dienen soll. Hierzu wird ein Messgerät entworfen, welches aufgenommene Daten an ein Smartphone sendet, wo diese weiterverarbeitet werden. Die Messwerte werden dann an einen Server mit Geoinformationssystem gesendet, wo die Daten gespeichert und in einer Karte dargestellt werden. Da eine Kalibrierung des Messgerätes unter Laborbedingungen und mit Referenzgasen den Rahmen dieser Arbeit überschreiten würde, werden die Messwerte hier nur auf Basis der Kennlinien der Sensorhersteller berechnet. Unter Beibehaltung möglichst gleichbleibender Störfaktoren kann so eine relative Messung stattfinden. Eine Methode zur Absolutwertbestimmung und Kalibrierung wird jedoch im Abschnitt 5.7 vorgestellt.

5.4.1 Auswahl von Hardwarekomponenten

Im Folgenden werden Gas- und Feinstaubsensoren, sowie Hardwareplattformen zur Berechnung und Datenübertragung ausgewählt, wobei Sensoren und Mikrocontroller-Platinen in einem Gehäuse verbaut werden sollen. Die Entscheidungen für die Festlegung auf die einzelnen Sensortypen basiert jeweils auf der besten Übereinstimmung mit den Anforderungen, die durch Vergleiche ermittelt wurden. Weiterhin werden relevante Eigenschaften der Sensoren dargestellt und diskutiert.

Zum Erfassen der Konzentrationen der Gase Kohlenmonoxid, Stickstoffdioxid und Ammoniak in der Luft wird der Dreikanal-Metalloxid-Halbleitersensordatensatz MiCS-6814 von SGX Sensortech eingesetzt. Dieser ist insbesondere für die Erfassung von Abgasen und durch Landwirtschaftliche Betriebe entstehende Luftbelastungen ausgelegt. Laut Datenblatt liegt der Erfassungsbereich für Kohlenmonoxid bei 1 bis 1000 ppm, für Stickstoffdioxid bei 0,05 bis 10 ppm und für Ammoniak bei 1 bis 500 ppm, womit die Anforderungen an den Erfassungsbereich erfüllt sind. Weiterhin kann der Sensor bei Temperaturen von -30 bis 85 °C und einer relativen Luftfeuchtigkeit von 5 bis 95 % betrieben werden, wodurch er sich zur Messung an der Außenluft eignet. Hierbei ist jedoch zu beachten, dass der Sensor vor Wasser und Verschmutzung geschützt werden muss. Dies wird beim Entwurf des Gehäuses berücksichtigt.

Weiterhin sind Selektivität und Sensitivität der einzelnen Kanäle zu beachten. So reagieren diese nicht nur auf die zu messenden Gase, sondern auch mit unterschiedlichen Empfindlichkeiten auf andere Gase. Abbildung 5.1 zeigt die Kennlinien der drei Kanäle des MiCS-6814,

anhand welcher erkennbar ist, dass Kohlenmonoxid- und Ammoniakkanal auch mit hoher Empfindlichkeit auf Wasserstoff reagieren. Vergleiche mit anderen Sensoren zeigen, dass diese geringe Selektivität typisch für Halbleitersensoren ist. Als Beispiel soll der Figaro TGS 3870-B00 dienen, dessen Kennlinie in [Abbildung 5.2](#) zu sehen ist. Ohne parallele Messung von Störgrößen kann die Ermittlung von Absolutwerten daher nicht oder nur mit großer Ungenauigkeit vorgenommen werden. Wenn nicht mit einer stabilen Konzentration von beispielsweise Wasserstoff gerechnet werden kann muss durch Nutzung der Messwerte aller Kanäle und bestenfalls weiterer Sensoren das Ergebnis rechnerisch korrigiert werden. Alters- oder Produktionsbedingte Abweichungen werden kompensiert, indem stets das Verhältnis von Nullwiderstand zu aktuellem Widerstand als Signal genommen wird. Der Nullwiderstand ist dabei der Widerstand, der am Sensor gemessen werden kann wenn er keinem Gas ausgesetzt ist auf welches er anspricht, also welches eine Veränderung der Leitfähigkeit der Messoberfläche hervorruft. Eine entsprechende Kalibrierung bei reiner Luft ohne Schadstoffbelastung ist daher in jedem Fall erforderlich.

Ebenfalls zu berücksichtigen ist, dass der Sensor jeweils erst nach einer Aufwärmphase die gewünschten Messwerte liefert. Auch weist der Hersteller darauf hin, dass die Lufttemperatur und die Luftfeuchtigkeit Einfluss auf die Messergebnisse nehmen. So ruft eine Veränderung dieser Größen ein Offset der Messwerte hervor. Laut Hersteller beträgt die typische Reaktionszeit der Sensoren 10 Sekunden. Konkret wird der Sensor als Teil eines Sensorsystems mit drei 8-Bit Analog-Digital-Wandlern und einem I2C-Interface verwendet.

Zur Erfassung flüchtiger organischer Verbindungen kommt der Winsen WSP2110 zum Einsatz. Der Erfassungsbereich liegt zwischen 1 und 50 ppm, womit auch hier der geforderte Erfassungsbereich abgedeckt ist. Ebenso muss dieser Sensor vor Wasser und Verschmutzungen geschützt werden. Wie auch der MiCS-6814 weist dieser Sensor nur eine geringe Selektivität auf. Außerdem rufen auch hier Veränderungen der Temperatur und der Luftfeuchtigkeit ein Offset der Messwerte hervor. Die Reaktionszeit ist mit 20 Sekunden angegeben, was unterhalb der geforderten Zeit liegt. Die Schnittstelle des Sensors ist passiv analog.

Um Temperatur- und Luftfeuchtigkeitsänderungen als Störfaktoren bei der Messung zu überwachen, werden diese Größen separat mit einem Seedstudio High Accuracy Temperature and Humidity Sensor mit TH02-Chip gemessen, welcher über einen I2C-Bus angebunden wird.

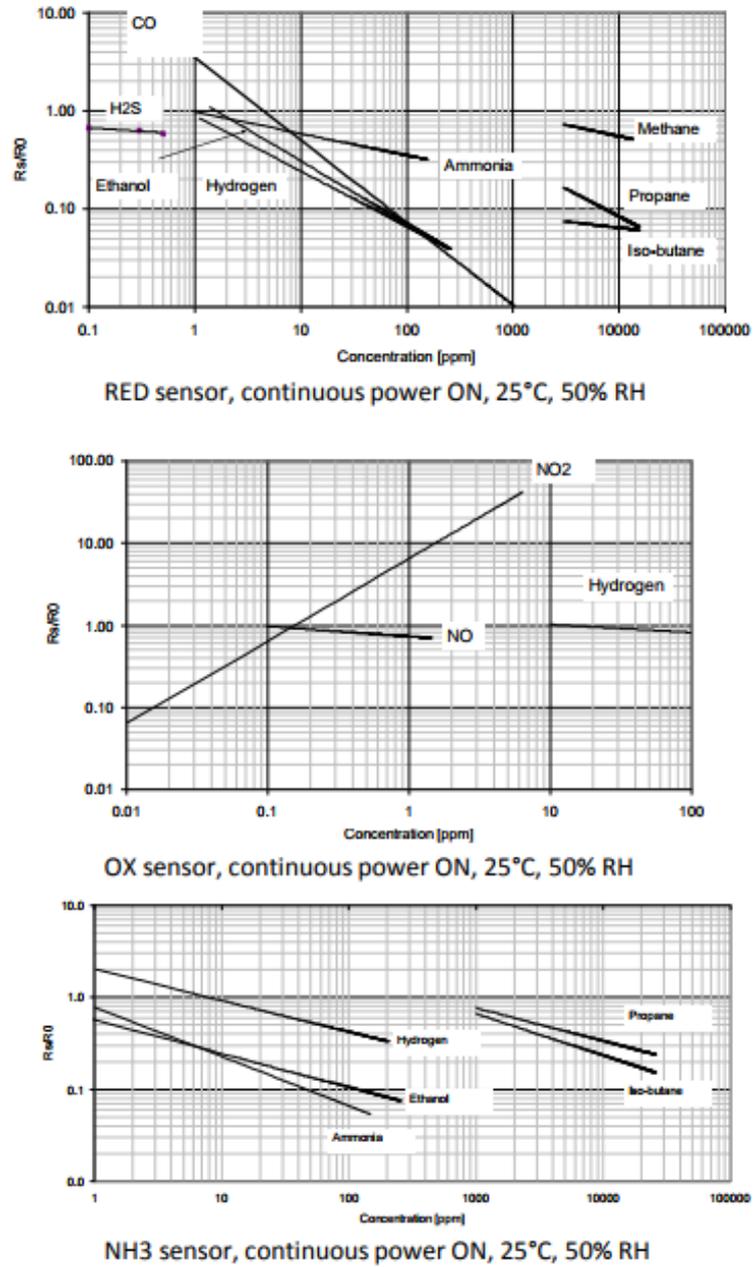


Abbildung 5.1: Kennlinien des Mehrkanal-Gassensors MiCS-6814 [55]

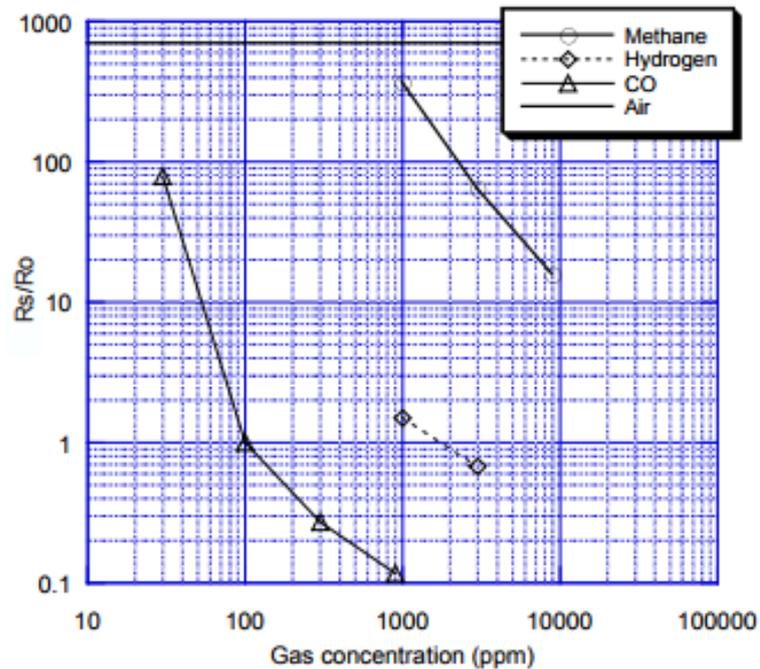


Abbildung 5.2: Kennlinie des Figaro Gassensors TGS 3870-B00 [4]

Feinstaub Konzentrationen werden mit dem Nova Fitness SDS011 Sensorsystem erfasst [11]. Dieses arbeitet mit einem Laser-Basierten Zählverfahren und kann die Feinstäube PM10 und PM2.5 mit einer Auflösung von $0,3 \mu\text{g}/\text{m}^3$ erfassen, wobei es für den Temperaturbereich von -10 bis $50 \text{ }^\circ\text{C}$ eingesetzt werden kann. Die maximale relative Luftfeuchtigkeit beträgt 70 %.

Der Erfassungsbereich liegt bei 0 bis $999 \mu\text{g}/\text{m}^3$ womit der geforderte Bereich abgedeckt ist. Die Reaktionszeit beträgt nach Herstellerangaben 10 Sekunden. Anzumerken ist auch, dass ein maximaler relativer Fehler von $\pm 10 \mu\text{g}/\text{m}^3$ einzurechnen ist, wobei dieser durch die Vielzahl an Messungen vernachlässigbar wird. Angebunden wird der Sensor über eine UART-Schnittstelle, über die jede Sekunde der aktuelle Messwert übertragen wird. Diese Arbeitet mit einer Spannung von 3,3 V weshalb beim Anschluss an einen 5 V Mikrocontroller ein Pegelwandler eingesetzt werden muss.

Entscheidungskriterium für die Auswahl dieses Sensors war unter anderem auch der erfolgreiche Einsatz im Projekt des OK Lab Stuttgart [40]. Im Vergleich mit anderen Sensoren, wie dem Shinyei PPD42 [56], bietet dieser den Vorteil, dass die Messwerte bereits in $\mu\text{g}/\text{m}^3$ ausgegeben

werden und nicht in Teilchen pro Volumeneinheit. Eine Umrechnung und Kalibrierung muss diesbezüglich nicht mehr vorgenommen werden. Zudem sorgt die aktive Belüftung für stabilere Messdaten.

Wegen der umfangreichen Peripherie und zahlreichen Hardwareschnittstellen mit entsprechenden Bibliotheken wird ein Arduino UNO R3 mit ATmega328P zum Auslesen der Sensordaten verwendet. Einer der eingebauten 10-Bit Analog-Digital-Wandler wird zum Erfassen des Widerstandes des Winsen Sensors verwendet. Die anderen Sensoren werden über Busse verbunden. Für die Weiterverarbeitung der aufgenommenen Werte und für die Serveranbindung kann das Arduino Uno Board nicht verwendet werden, weshalb auf leistungsstärkere Hardware zurückgegriffen werden muss. Diese soll über eine drahtlose Bluetooth-Schnittstelle mit dem Arduino Uno Board kommunizieren, weshalb ein Bluetooth-Modul zum weiteren Umfang der Hardware gehört. Das ausgewählte Modul, Seedstudio Bluetooth V3, verfügt über einen CSR Bluecore-04-Chip, welcher den Bluetooth 2.0+EDR Standard erfüllt. Da die Übertragungsraten im Megabit Bereich liegen, ist diese in jedem Fall als ausreichend einzuschätzen. Der Anschluss des Moduls erfolgt über eine UART-Schnittstelle.

Zur weiteren Verarbeitung der Sensordaten wird ein Android Smartphone eingesetzt. Dies bietet neben einem leistungsstarken Prozessor in der Regel die nötige Sensorik zur Positionsbestimmung, wie auch die Möglichkeit Daten drahtlos an einen Server zu übertragen. So kann der Hardwareaufwand des zu entwerfenden Prototyps stark reduziert werden ohne Peripheriegeräte wie ein Modem zu benötigen, wobei ein Smartphone als gängiges, ohnehin vorhandenes Gerät betrachtet wird. Dieses Konzept wird auch bei zahlreichen anderen Geräten wie Multimediasystemen eingesetzt.

Zum Ansaugen der Außenluft wird ein Arctic F8 Axiallüfter verwendet, welcher mit 12 V betrieben werden muss. Zu beachten ist, dass je nach Fördermenge die am Messort angesaugte Luft erst verzögert im Gehäuse zur Messung bereit steht. Dies wird im Abschnitt 5.4.5 diskutiert.

Intern soll das Gerät mit 9 V Spannung versorgt werden. Dazu wird die Spannung des Pkw-Bordnetzes mit einem Spannungsregler herunter geregelt.

Zusammengefasst werden vier Gassensoren, ein Mehrkanal- und ein Einkanal-Sensor, ein Temperatur- und Luftfeuchtesensor, ein Feinstaubsensor und ein Bluetooth Modul zusammen mit einer Mikrocontroller-Platine in ein Gehäuse verbaut. Intern werden weiterhin zwei Span-

nungsregler und ein Logik-Pegelwandler benötigt. Außerhalb des Gehäuses wird ein weiterer Spannungsregler verwendet, um die Versorgungsspannung bereit zu stellen.

5.4.2 Entwurf der Hardware

Die im letzten Abschnitt genannten Hardwarekomponenten werden gemäß des Blockschaltbildes 5.3 verbunden, wobei nur das Herunterregeln der Bordnetzspannung auf 9 V außerhalb des Gehäuses stattfindet. Das Arduino Board wird direkt an die 9 V angeschlossen. Für den Lüfter wird die Spannung entsprechend auf 12 V hochgeregelt. Bluetooth-Modul und Feinstaubsensor werden jeweils über eine UART-Schnittstelle an das Arduino Uno Board verbunden. Das Bluetooth-Modul wird vom Spannungsregler der Mikrocontroller-Platine mit 5 V Spannung versorgt. Wegen der hohen Leistungsaufnahme muss der Feinstaubsensor separat an einen Spannungsregler angeschlossen werden. Ein Logik-Pegelwandler wird eingesetzt, um die 5 V des ATmega328P auf die 3,3 V des Feinstaubensors herunterzuregeln und umgekehrt. Temperatur- und Luftfeuchtesensor, sowie Mehrkanal-Gassensor werden jeweils über einen I2C-Bus angeschlossen, wobei die Versorgungsspannung von 5 V von der Mikrocontroller-Platine stammt. Da der TH02 Sensor laut Herstellerangaben abgeschaltet werden muss wenn mit anderen Geräten auf dem I2C-Bus kommuniziert wird, wurde dieser an einen eigenen Bus angeschlossen, um ständiges Ein- und Ausschalten zu umgehen. Der Widerstand des VOC-Sensors wird über einen der integrierten 10-Bit Analog-Digital-Wandler des ATmega328P gemessen. Die Versorgungsspannung von 5 V wird auch hier vom Arduino Board bereitgestellt. Entsprechende Widerstände sind bereits auf der Sensorplatine verbaut.

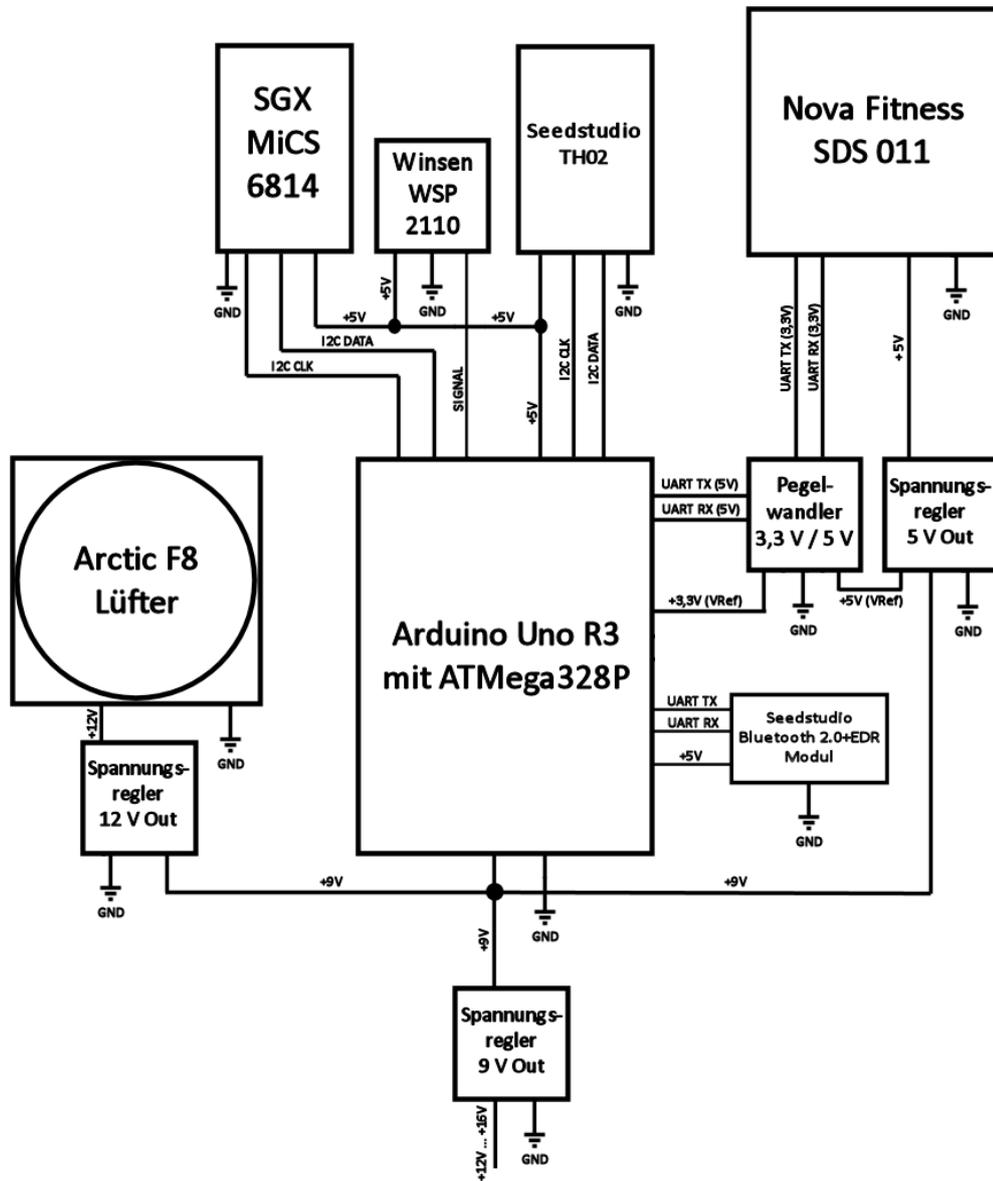


Abbildung 5.3: Blockschaltbild der Hardware des Prototyps

5.4.3 Entwurf einer Mikrocontroller-Software zum Auslesen der Sensoren

Zum Auslesen der Sensoren und senden der Daten an das Android Smartphone wird eine Mikrocontroller-Software entworfen. Die Programmstruktur im Überblick wird in Abbildung 5.4 dargestellt. Die Subaktivität "Lese Sensoren aus und sende per Bluetooth" wird anhand von Abbildung 5.5 illustriert.

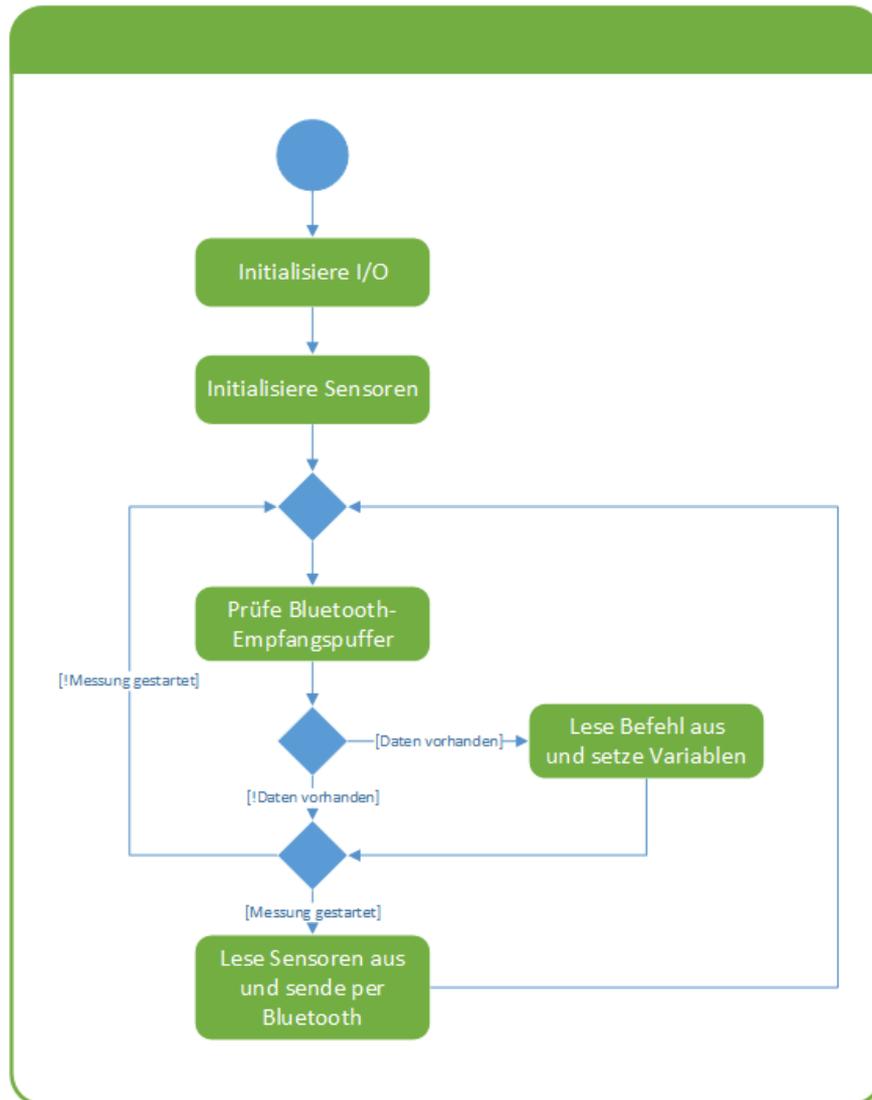


Abbildung 5.4: Mikrocontroller-Programmstruktur im Überblick als UML-Aktivitätsdiagramm

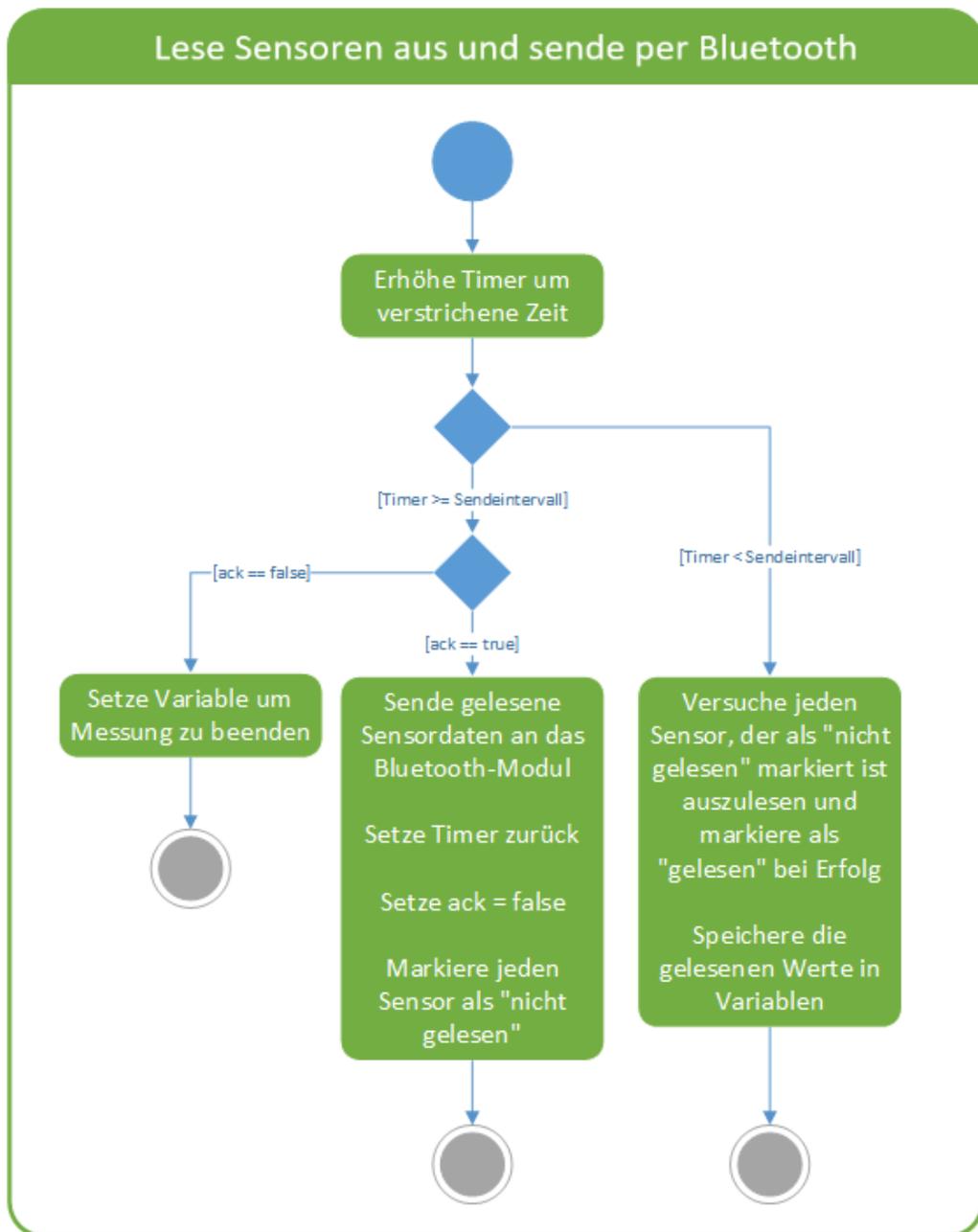


Abbildung 5.5: Sensoren-Lese- und Bluetooth-Sende-UML-Aktivität aus Abbildung 5.4

Grundsätzlich werden in einer Endlosschleife zwei Aktionen ausgeführt. Die erste ist das Auslesen des Bluetooth-Empfangspuffers und die Verarbeitung der ankommenden Steuerbefehle vom Smartphone. Die zweite Aktion ist das Auslesen der Sensoren und das Versenden der Daten an das Smartphone, wobei diese Aktion nur bei gestarteter Messung ausgeführt wird.

Die Steuerbefehle die empfangen werden können sind "START", "STOP" und "ACK". Sie sind jeweils als 8-Bit Integer-Zahl kodiert. "START" und "STOP" starten beziehungsweise beenden die Messung. "ACK" wird vom Smartphone zum Bestätigen empfangener Daten versendet. Wenn der Startbefehl empfangen wird muss zum einen das "Start Flag" gesetzt werden, welches signalisiert, dass die Messung aktiv ist. Zum anderen muss einmalig das "Acknowledge Flag" gesetzt werden, da die jeweils letzten empfangenen Daten vom Smartphone bestätigt werden. Bei einem Stoppbefehl wird das "Start Flag" gelöscht.

Bei gesetztem "Start Flag" werden zusätzlich zum Lesen des Empfangspuffers die Sensoren ausgelesen und die Daten entsprechend versendet. Hierzu wird ein Timer bei jedem Schleifendurchlauf um die verstrichene Zeit erhöht. Ist das Sendeintervall von 20 Sekunden erreicht, so wird geprüft, ob die zuletzt verschickten Daten bestätigt wurden. Ist dies nicht der Fall, wird das "Start Flag" gelöscht und die Messung beendet, da von einem Fehler auszugehen ist. Wurde hingegen das "Acknowledge Flag" gesetzt, so werden die letzten gelesenen Sensorwerte an das Bluetooth-Modul übertragen und sofort verschickt. Wenn das Sendeintervall noch nicht erreicht wurde, werden die letzten Sensorwerte aktualisiert, indem die Sensoren ausgelesen werden. Pro Sendeintervall wird jeder Sensor jedoch nur einmal ausgelesen. Um die Daten der Sensoren, die über I2C-Busse verbunden sind, auszulesen, muss mit diesen über entsprechende Unterfunktionen kommuniziert werden. Die bereitgestellten Bibliotheks-Funktionen geben bei Fehlern entsprechende Rückgabewerte. Weiterhin muss der aktuelle Wert des gewählten Analog-Digital-Wandlers ausgelesen werden. Für die Werte des Feinstaubensors muss der Puffer der Seriellen Schnittstelle ausgelesen werden. Sekündlich trifft bei korrekter Funktion eine 10-Byte Sequenz ein. Daher werden wenn verfügbar die zuletzt angekommenen 10-Byte aus dem Puffer gelesen. Um ein Verschieben der Sequenz auszuschließen werden Head- und Tail-Byte gelesener Daten überprüft. Bei einem Fehler wird der Puffer geleert.

5.4.4 Entwurf einer Android-Applikation zur Datenweiterverarbeitung

Aufgabe der Android-Applikation ist es Daten vom Messgerät zu empfangen, Berechnungen auf ihnen auszuführen und an den Server zu versenden. Das Diagramm in Abbildung 5.6 zeigt die Klassen mit den wichtigsten Methoden.

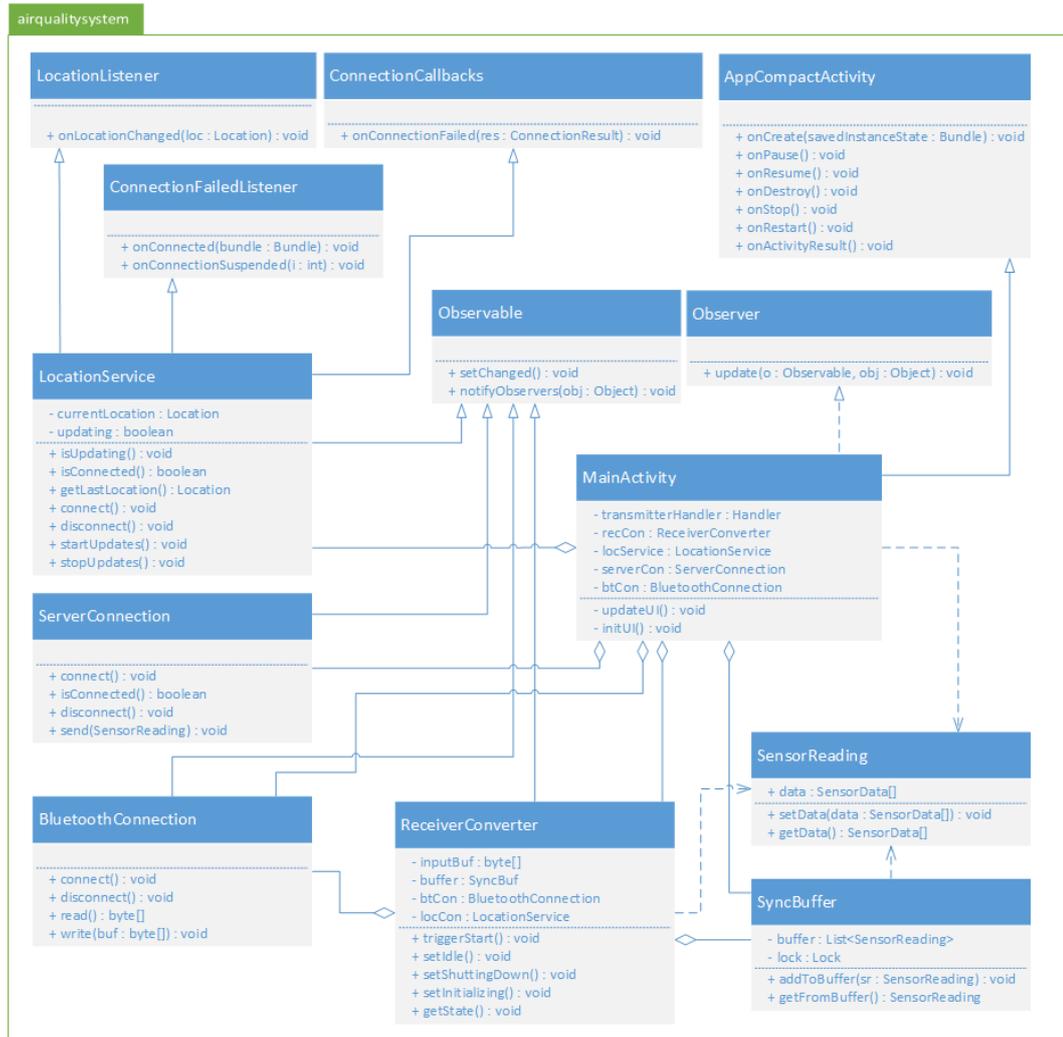


Abbildung 5.6: UML-Klassendiagramm der Android Applikation

Die entworfene Software basiert weitestgehend auf asynchroner Kommunikation einzelner Handler, die periodisch Daten empfangen und verarbeiten, sowie die Messwerte an den Server übermitteln. Die grafische Oberfläche mit entsprechenden Schaltflächen wird innerhalb der Klasse "MainActivity" initialisiert und aktualisiert. Sämtliche Aktionen werden dann von hier aus gestartet und beendet.

In der Klasse "LocationService", die den Google API Fused Location Service kapselt, wird über einen Call-Back-Mechanismus alle 2 Sekunden die Position aktualisiert, oder so schnell eine neue Position vom darunterliegenden System bereitgestellt wird. Die aktuellste Position kann dann über einen Methodenaufruf abgerufen werden. Kann zum Beispiel bei Tunnelfahrten die Position nicht aktualisiert werden, werden die Positionen nachträglich linear interpoliert, sobald das nächste Positionsupdate verfügbar ist. Für die Fehlererkennung werden Listener eingesetzt, die beim Fused Location Service angemeldet werden. Über das Observer-Pattern wird bei einem Fehler der geänderte Verbindungsstatus an die "MainActivity" weitergeleitet, von wo aus die Messung beendet und der Benutzer informiert wird.

Ankommende Daten werden vom "ReceiverHandler" in der Klasse "ReceiverConverter" gelesen und in Messwerte umgerechnet. Auch die Steuerbefehle für den Mikrocontroller werden durch den Handler verschickt. Vergehen mehr als 40 Sekunden bevor neue Daten empfangen werden ist dies als Fehler zu bewerten. Auch das erfolglose Schreiben auf den Output-Stream oder Lesen vom Input-Stream stellt einen Fehler dar. Dies wird der "MainActivity" über die "notifyObservers" Methode mitgeteilt, was ein Beenden der Messung zur Folge hat. Empfangene einzelne unplausible Daten, wie etwa die negativen Rückgabewerte der Funktionen der Mikrocontroller-Software, werden verworfen. Mehr als 10 Fehler werden jedoch dem Nutzer mitgeteilt und die Messung wird beendet, da ein Hardwarefehler vorliegen könnte. Die Bluetooth-Verbindung wird von der Klasse "BluetoothConnection" gekapselt, welche über die Rückgabewerte der "read" und "write" Methoden Fehler signalisiert. Da der Verbindungsaufbau Nutzereingaben erfordert, die wiederum einen Wechsel des Verbindungsstatus bewirken, ist diese Klasse auch vom Typ "Observable". Veränderungen des Verbindungsstatus werden so an die "MainActivity" weitergeleitet.

Von der Klasse "MainActivity" aus werden durch den "TransmitterHandler" die Messwert-Objekte an den Server verschickt. Die Serververbindung wird von der Klasse "ServerConnection" gekapselt. Dem verwendeten Protokoll entsprechend wird hier eine Client-Bibliothek eingesetzt. Konkret wird dies in Abschnitt 5.4.6 thematisiert. Ab Start der Applikation wird eine

Verbindung zum Server hergestellt und bei laufender Messung alle 10 s, wenn verfügbar, ein Datenobjekt verschickt. Das Intervall ist halb so lang wie das Sendeintervall des Mikrocontrollers, um mögliche Verbindungsprobleme zu kompensieren. Bei einem Abriss der Verbindung wird diese automatisch versucht wiederherzustellen. Tritt beim Senden ein Fehler auf, so wird dies erneut über das Observer-Pattern bekannt gegeben. Mehr als 10 Sendefehler führen zu einem Beenden der Messung.

Der Ablauf einer Messung und die Kommunikation der einzelnen Handler wird in Abbildung 5.7 dargestellt.

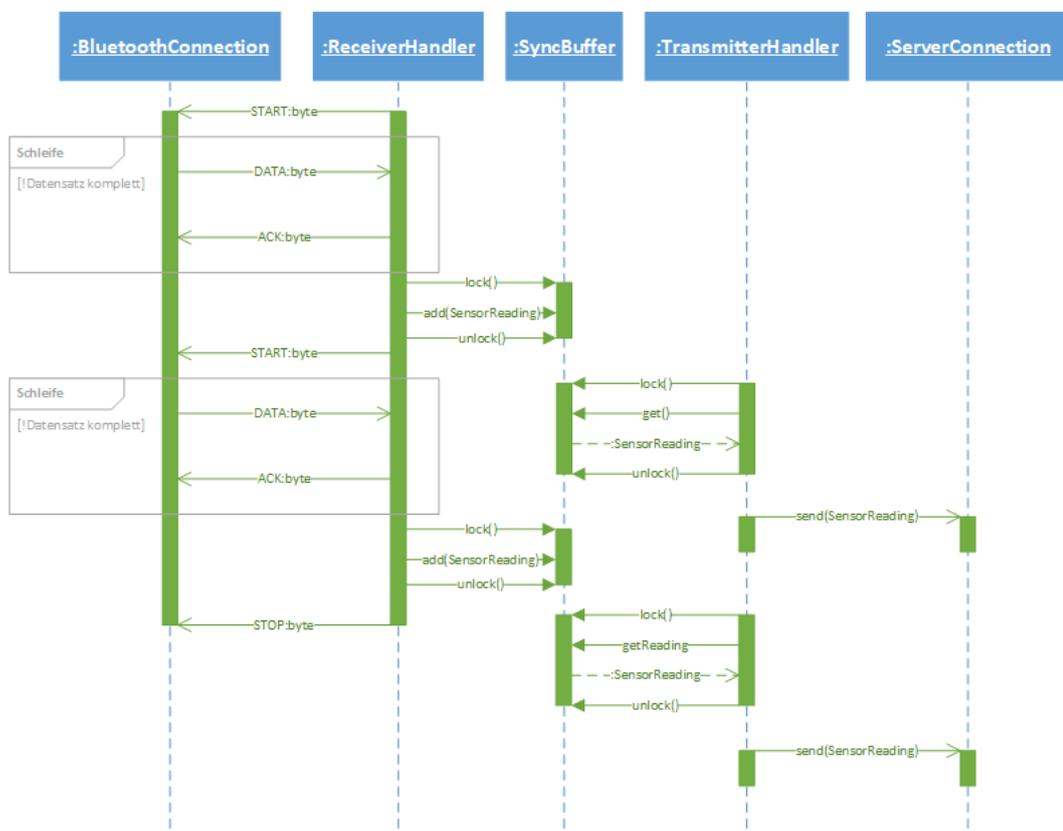


Abbildung 5.7: UML-Seqenzdiagramm der Handler-Kommunikation

In Abbildung 5.8 wird der "ReceiverHandler" als Zustandsautomat dargestellt. Hieran wird das Schema zur Fehlerbehandlung veranschaulicht. Tritt eine Zeitüberschreitung beim Empfang von Daten auf oder kann nicht auf Input- oder Output-Stream zugegriffen werden, so wechselt

der Handler in den Zustand "ShuttingDown" und informiert die "MainActivity" über den Fehler. In der hierdurch aufgerufenen "update" Methode wird dann der Nutzer über den Fehler informiert und die Verbindung wird ordentlich geschlossen, sodass ein erneuter Verbindungsaufbau stattfinden kann. Der "TransmitterHandler" arbeitet jedoch weiter bis alle Daten versendet wurden. Die Aktualisierung der Position wird hiervon nicht betroffen.

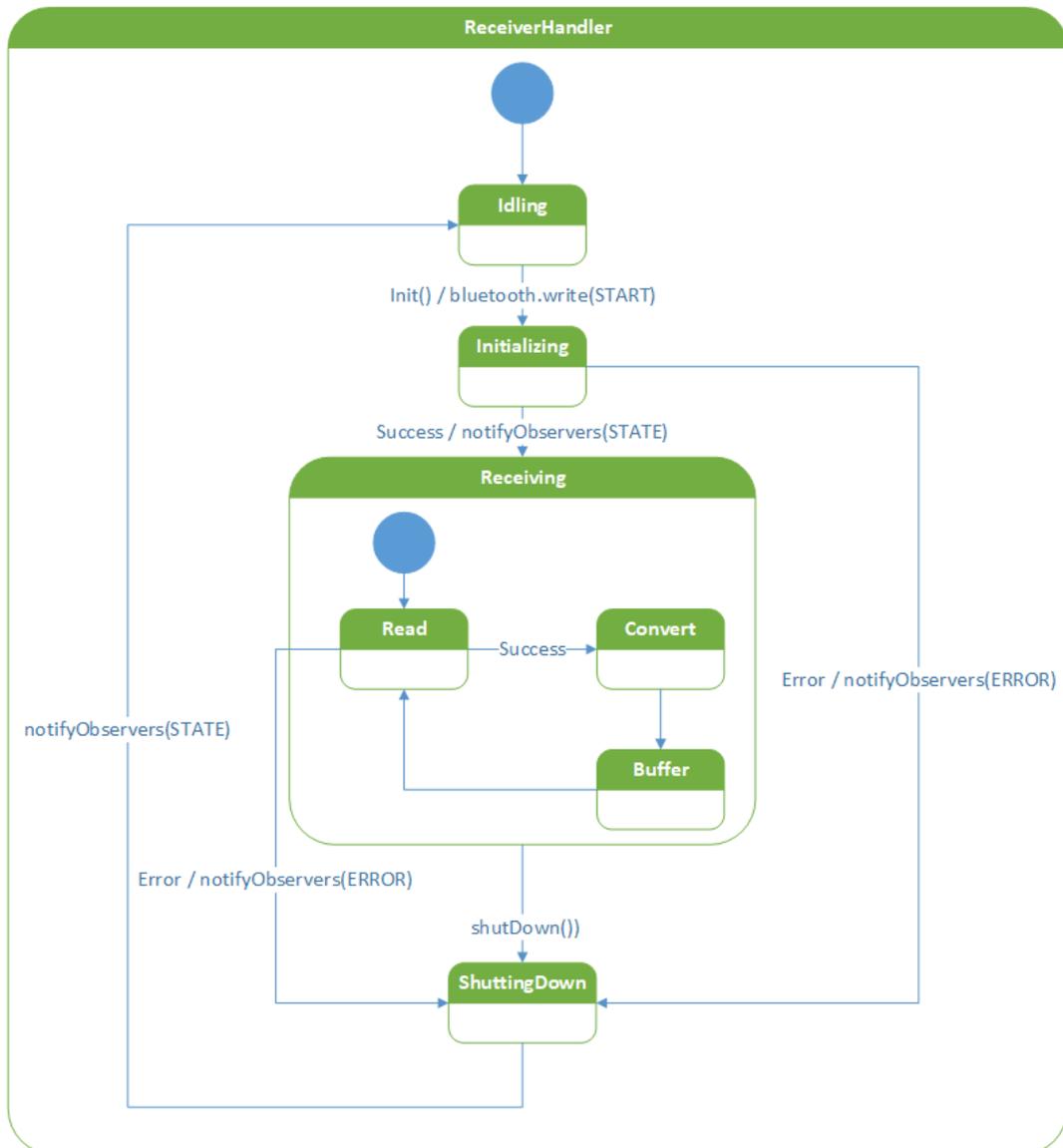


Abbildung 5.8: UML-Zustandsautomat des ReceiverHandlers

5.4.5 Entwurf eines Gehäuses

Für das Gehäuse für Sensoren und Mikrocontroller soll ein Universal-Kunststoffgehäuse mit den Maßen 220 x 125 x 110 mm zum Einsatz kommen. An der Oberseite wird über einer Öffnung der Arctic F8 80 mm Lüfter angebracht, welcher die Luft aus dem Gehäuse zieht. Durch eine Öffnung an der Seite, an der ein Schlauch mit einem Durchmesser von 30 mm und einer Länge von 150 cm montiert ist, wird die Luft entsprechend nachgezogen. Hierbei strömt sie im unteren Teil des Gehäuses an den Gassensoren vorbei, wobei schwere Partikel wie Wassertropfen und Verschmutzungen auf dem Boden zurückbleiben. Der Feinstaubsensor mit eigener Belüftung zieht einen Teil der Luft aus dem Strom. Die Abluft wird in der Nähe des Schachtes, der in den oberen Gehäuseteil führt, ausgestoßen. Die schematische Darstellung 5.9 veranschaulicht dies. Die Luftströmung ist durch grüne Pfeile markiert.

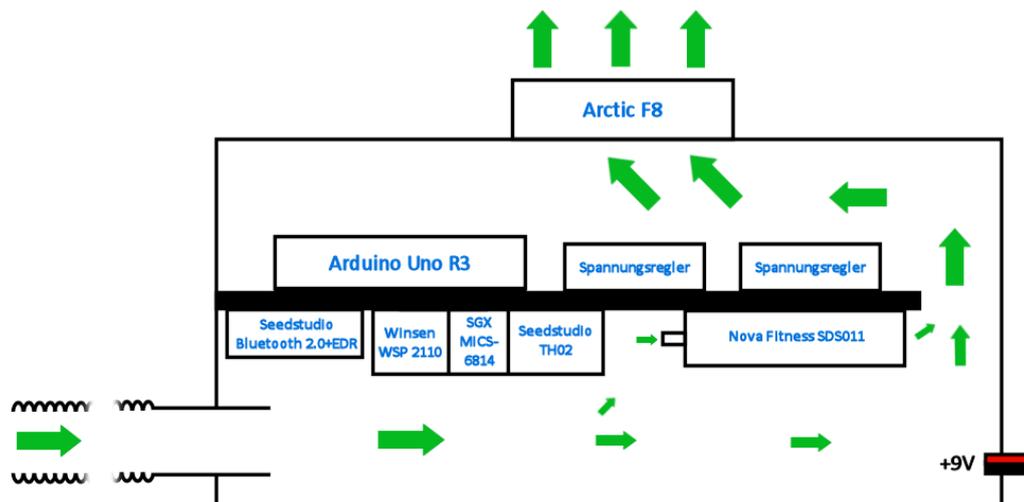


Abbildung 5.9: Schematische Darstellung des Gehäuses

Der eingesetzte Lüfter hat laut Datenblatt eine Fördermenge von $47,6 \text{ m}^3/\text{h}$ bei 12 V. Folglich ist maximal mit einem Durchfluss von etwa 13 Litern pro Sekunde zu rechnen. Bei Einberechnung des Luftvolumens im Gehäuse und dem Ansaugschlauch, welches kombiniert etwa 4 Liter beträgt, würden vernachlässigbare 330 ms vergehen bis die Außenluft das Innere des Gehäuses durchflossen hat. Unter Berücksichtigung von Verlusten durch Reibung und Druck ist jedoch mit einer deutlich größeren Verzögerung bis zum Erreichen der Außenluft im Gehäuse zu

rechnen. Auch eventuelle Verwirbelungen sind nicht einberechnet. Die Strömungstechnischen Eigenschaften des Gehäuses sollen jedoch nicht Gegenstand dieser Arbeit sein.

5.4.6 Auswahl eines Geoinformationssystems

Aufgrund der Anforderungen an die Skalierbarkeit des Systems wird für den Dienst eine IoT-Plattform gewählt. Beispiele hierfür sind Kaa IoT [39] und Crossbar.io [32]. Sie bilden das Bindeglied zwischen eingebetteten Systemen und Datenbanken oder Anwendungen zur Verarbeitung und Visualisierung von Daten. Besonders sind sie für verteilte Systeme und Microservice-Anwendungen ausgelegt, wobei die massenhafte Auswertung von Sensordaten einen typischen Anwendungsfall darstellt.

Konkret wird hier die von Y. Wilkening im Rahmen seiner Bachelorarbeit [60] entwickelte Plattform verwendet, welche er für diese Arbeit angepasst hat. Sie basiert auf der Crossbar.io Plattform, an welche eine SQL-Datenbank und eine Webapplikation zur grafischen Darstellung angebunden ist.

Für die Kommunikation zwischen Android-Gerät und Serverplattform muss wegen der Einschränkung durch die Crossbar.io Software das Web Application Messaging Protocol [53] verwendet werden. Hierfür gibt es Android-Client-Bibliotheken, wodurch die Implementierung der Serveranbindung erleichtert wird. Das Protokoll bietet zwei Kommunikationsmuster an. Diese sind Publish and Subscribe und Remote Procedure Call. Da der hier konzeptionierte Dienst periodisch selbständig Daten versendet, wird vom Publish-and-Subscribe-Schema Gebrauch gemacht. Das bedeutet, dass vom Android-Gerät aus die Messwerte jeweils veröffentlicht und über die crossbar.io-Plattform an die Datenbank weitergeleitet werden.

Aufgrund der Verwendung dieses standardisierten Protokolls entfällt die Implementierung zusätzlicher Sicherheitsschichten auf dem Android-Smartphone. Das Webinterface, also die Kommunikationsschnittstelle, wurde weitestgehend offengelassen. Das bedeutet, dass es keine Einschränkungen bezüglich bestimmter gemessener Gase gibt und somit auch andere Hardware eingesetzt werden könnte.

Die Webapplikation bietet die Möglichkeit den Ort und den Umkreis anzugeben, zu dem die gemessenen Werte angezeigt werden sollen. Da der Dienst als Crowdsourcing-Dienst ausgelegt ist, werden jeweils die Mittelwerte aller Messwerte im jeweiligen Umkreis gebildet.

Die Darstellung erfolgt auf einer Karte von Google Maps [35], welche über eine entsprechende API eingebunden ist. Zu erreichen ist die Webapplikation unter <http://141.22.10.81/webApp/>.

5.4.7 Qualitätssicherungsmaßnahmen

Für die Qualitätssicherung werden die einzelnen Komponenten der Hard- und Software getestet. Dies umfasst das Verhalten der Mikrocontroller-Software im normalen Betrieb und im Fehlerfall, sowie das Verhalten der Android-Applikation in beiden Fällen. Ein Test der Server-Plattform soll nicht stattfinden. Es wird lediglich überprüft, ob die gesendeten Daten korrekt in der Datenbank aufgenommen werden.

Zum Testen des Mehrkanal-Gassensors, des Temperatur- und Luftfeuchtesensors und des Feinstaubensors werden diese über I2C-Busse beziehungsweise eine serielle Schnittstelle an den Mikrocontroller verbunden. Es werden dann in einem ersten Schritt die Rohdaten ausgelesen und über eine serielle Verbindung an einen Desktop-Rechner gesendet wo sie auf Plausibilität überprüft werden. Im Falle des Mehrkanal-Gassensors wird hierfür die entsprechende Bibliotheksfunktion verwendet. Für den Temperatur- und Luftfeuchtesensor werden Unterfunktionen geschrieben, mit denen die Werte gemäß Datenblatt [54] ausgelesen werden. Der Feinstaubsensor sendet periodisch Daten über eine serielle Schnittstelle, deren Empfangspuffer lediglich ausgelesen wird. Der interne Analog-Digital-Wandler, an den der VOC-Sensor angeschlossen wird, wird nicht überprüft, da von der korrekten Funktion der Arduino-Bibliotheksfunktion ausgegangen wird. Für die übrigen oben genannten Sensoren sind in Tabelle 5.1 die Wertebereiche der Rohdaten und die korrespondierenden Messbereiche aufgelistet, anhand welcher die Plausibilität der Daten überprüft werden kann.

In einem zweiten Schritt werden die Rohdaten des Temperatur- und Luftfeuchtesensors gemäß Datenblatt auf dem Mikrocontroller in Messwerte umgerechnet und versendet. Mit einem externen Messgerät kann dann die korrekte Funktion des Sensors überprüft werden. Auch die Rohdaten des Feinstaubensors werden bereits auf dem Mikrocontroller umgerechnet. Hierzu wird ebenfalls gemäß Datenblatt vorgegangen [11]. Eine Prüfung der korrekten Funktion des Sensors findet jedoch nicht statt. Die Messwerte, die durch den Mehrkanal-Gassensor ermittelt werden, werden erst auf dem Android-Gerät berechnet. Daher wird der zweite Prüfschritt hier nicht durchgeführt.

Tabelle 5.1: Werte- und Messbereich zum Testen der Mikrocontroller-Software

Sensortyp	Wertebereich	Messbereich
Temperatur	[320, 4320]	[-40°C, +85°C]
Relative Luftfeuchtigkeit	[384, 1984]	[0%, 100%]
PM10	[0, 3] [0, 255] (High und Low Byte)	[0 $\mu\text{g}/\text{m}^3$, 793,5 $\mu\text{g}/\text{m}^3$]
PM2.5	[0, 3] [0, 255] (High und Low Byte)	[0 $\mu\text{g}/\text{m}^3$, 793,5 $\mu\text{g}/\text{m}^3$]
Kohlenmonoxid	[0, 1023]	-
Stickstoffdioxid	[0, 1023]	-
Ammoniak	[0, 1023]	-

Um das Verhalten bei Busfehlern oder Defekten der Sensoren zu testen wird die Kommunikation manuell systematisch für die einzelnen Sensoren unterbrochen. Das erwartete Verhalten ist ein Fehlercode als Rückgabewert der jeweiligen Funktionen in der Mikrocontroller-Software. Alle Werte, darunter auch die Fehlercodes, werden unverändert in die Android-Applikation weitergereicht.

In Bezug auf die Android-Applikation muss getestet werden, ob die übermittelten Daten korrekt interpretiert werden. Hierzu werden vom Mikrocontroller aus per Bluetooth definierte Testsequenzen übermittelt und über das grafische Interface überprüft. Da es sich bei allen Werten um 32-Bit Floating Point Zahlen handelt, können jeweils die Äquivalenzklassen im Bereich vom Mindestwert des Datentypen bis 0 und von 0 bis zum Maximalwert überprüft werden, wobei der Nulldurchgang, also der Bereich von -1 bis +1, schrittweise durchlaufen wird. Um das korrekte Verhalten der einzelnen Software-Komponenten zu testen wird per USB-Debugging eine Log-Ausgabe auf einem Desktop-Rechner ausgegeben, wobei der Zustand der Handler, der Bluetooth- und Serververbindung, sowie der Status der Positionsbestimmung ausgegeben werden. Über die Bedienelemente der grafischen Oberfläche wird dann die Software gesteuert und die Log-Ausgabe überwacht. In Tabelle 5.2 sind die Eingaben und erwarteten Zustände der Komponenten definiert. Hierbei wird von Betriebsbereitem Messgerät und Server ausgegangen. Auch Bluetooth-Verbindung, Internetverbindung und GPS-Positionsbestimmung werden als eingeschaltet und fehlerfrei betrachtet.

Tabelle 5.2: Zustände der Komponenten der Android-Applikation bezüglich Eingaben

Eingabe	Bluetooth-Verbindung	Server-Verbindung	Receiver-Handler	Transmitter-Handler
Keine	Nicht verbunden	Verbunden	Zustand "Idling"	Nicht aktiv
Verbindung zum Messgerät Herstellen	1. Verbindungsaufbau 2. Verbunden	Verbunden	Zustand "Idling"	Nicht aktiv
Messung starten	Verbunden	Verbunden	1. Zustand "Initializing" 2. Zustand "Receiving"	Aktiv
Keine (aktive Messung)	Verbunden	Verbunden	Zustand "Receiving"	
Messung beenden	Verbunden	Verbunden	1. Zustand "Shutting Down" 2. Zustand "Idling"	1. Aktiv 2. Nicht aktiv
Verbindung zum Messgerät trennen	Nicht verbunden	Verbunden	Zustand "Idling"	Nicht aktiv

Um zu testen, ob die Daten korrekt an den Server übermittelt werden, werden Testwerte basierend auf den oben definierten Äquivalenzklassen an einer beliebigen Position übermittelt. Über das Webinterface kann dann überprüft werden, ob diese korrekt angezeigt werden.

Die Positionsinterpolation wird an einem konkreten Anwendungsfall überprüft. Hierzu wird eine festgelegte Strecke abgefahren und die GPS-Funktion des Smartphones abgeschaltet und nach einer Strecke von mindestens 1 km wieder eingeschaltet, wobei die genaue Uhrzeit festgehalten wird. Auf der Karte des Webinterfaces kann dann nachvollzogen werden, ob die Positionen korrekt interpoliert wurden.

Um das Verhalten im Fehlerfall zu überprüfen wird nach dem gleichen Schema vorgegangen. Über das Betriebssystem lassen sich die Hardwarefunktionen, also Bluetooth-Verbindung, GPS-Positionsbestimmung und die Internetverbindung abschalten. Dies wird genutzt um entsprechende Fehlerfälle zu simulieren. In Tabelle 5.3 wird der jeweilige Fehlerfall und das erwartete Verhalten des Systems definiert.

Tabelle 5.3: Fehlerfälle und Systemverhalten der Android-Applikation

Initialer Systemzustand	Fehlerfall	Verhalten des Systems	Anschließendes Zustand des Systems
Das Messgerät ist verbunden, Der Server ist verbunden, Der ReceiverHandler ist im Zustand "Receiving", Der Transmitter-Handler ist aktiv	Zeitüberschreitung beim Empfang von Daten	1. Der ReceiverHandler wechselt in den Zustand "Shutting Down" 2. Die Bluetooth-Verbindung wird getrennt 3. Ein Flag wird gesetzt, sodass der TransmitterHandler inaktiv wird sobald der Puffer geleert ist 4. Dem Nutzer wird eine Fehlermeldung angezeigt	Das Messgerät ist nicht verbunden, Der Server ist verbunden, ReceiverHandler ist im Zustand "Idling", Der TransmitterHandler ist nicht aktiv
Das Messgerät ist verbunden, Der Server ist verbunden, Der ReceiverHandler ist im Zustand "Receiving", Der Transmitter-Handler ist aktiv	Abriss der Bluetooth-Verbindung	1. Der ReceiverHandler wechselt in den Zustand "Shutting Down" 2. Die Bluetooth-Verbindung wird getrennt 3. Ein Flag wird gesetzt, sodass der TransmitterHandler inaktiv wird sobald der Puffer geleert ist 4. Dem Nutzer wird eine Fehlermeldung angezeigt	Das Messgerät ist nicht verbunden, Der Server ist verbunden, ReceiverHandler ist im Zustand "Idling", Der TransmitterHandler ist nicht aktiv
Das Messgerät ist verbunden, Der Server ist verbunden, Der ReceiverHandler ist im Zustand "Receiving", Der Transmitter-Handler ist aktiv	Mehr als 10 unplausible Messwerte oder Fehlercodes erhalten	1. Der ReceiverHandler wechselt in den Zustand "Shutting Down" 2. Die Bluetooth-Verbindung wird getrennt 3. Ein Flag wird gesetzt, sodass der TransmitterHandler inaktiv wird sobald der Puffer geleert ist 4. Dem Nutzer wird eine Fehlermeldung angezeigt	Das Messgerät ist nicht verbunden, Der Server ist verbunden, ReceiverHandler ist im Zustand "Idling", Der TransmitterHandler ist nicht aktiv
Das Messgerät ist verbunden, Der Server ist verbunden, Der ReceiverHandler ist im Zustand "Receiving", Der Transmitter-Handler ist aktiv	Mehr als 10 Sendefehler bei der Übertragung zum Server	1. Der ReceiverHandler wechselt in den Zustand "Shutting Down" 2. Die Bluetooth-Verbindung wird getrennt 3. Der TransmitterHandler wird angehalten 4. Die Serververbindung wird getrennt 5. Dem Nutzer wird eine Fehlermeldung angezeigt	Das Messgerät ist nicht verbunden, Der Server ist nicht verbunden, ReceiverHandler ist im Zustand "Idling", Der TransmitterHandler ist nicht aktiv

5.5 Umsetzung des Prototyps

5.5.1 Hardwareseitige Umsetzung und Einbau in das Gehäuse

Die Hardware wurde nach dem Schaltbild aus Abschnitt 5.4.2 umgesetzt. Für die 9 V Spannungsversorgung wurde ein Kfz-DC-DC-Netzteil verwendet. Sensoren und Mikrocontroller-Platine wurde auf einer Acrylglasplatte montiert, welche mittig im Gehäuse eingepasst wurde. Zur Fixierung wurden zugeschnittene Kunststoffprofile verwendet, wie es in Abbildung 5.10 zu sehen ist. Eines der Schlauchenden ist gummiert, sodass es passgenau in die gebohrte Öffnung des Gehäuses eingeführt werden konnte. Weiterhin wurde die Öffnung an der Oberseite, über der der Lüfter montiert ist, abgedichtet. Ein Schalter an der Gehäuseoberseite ermöglicht es das Gerät ein- und auszuschalten.

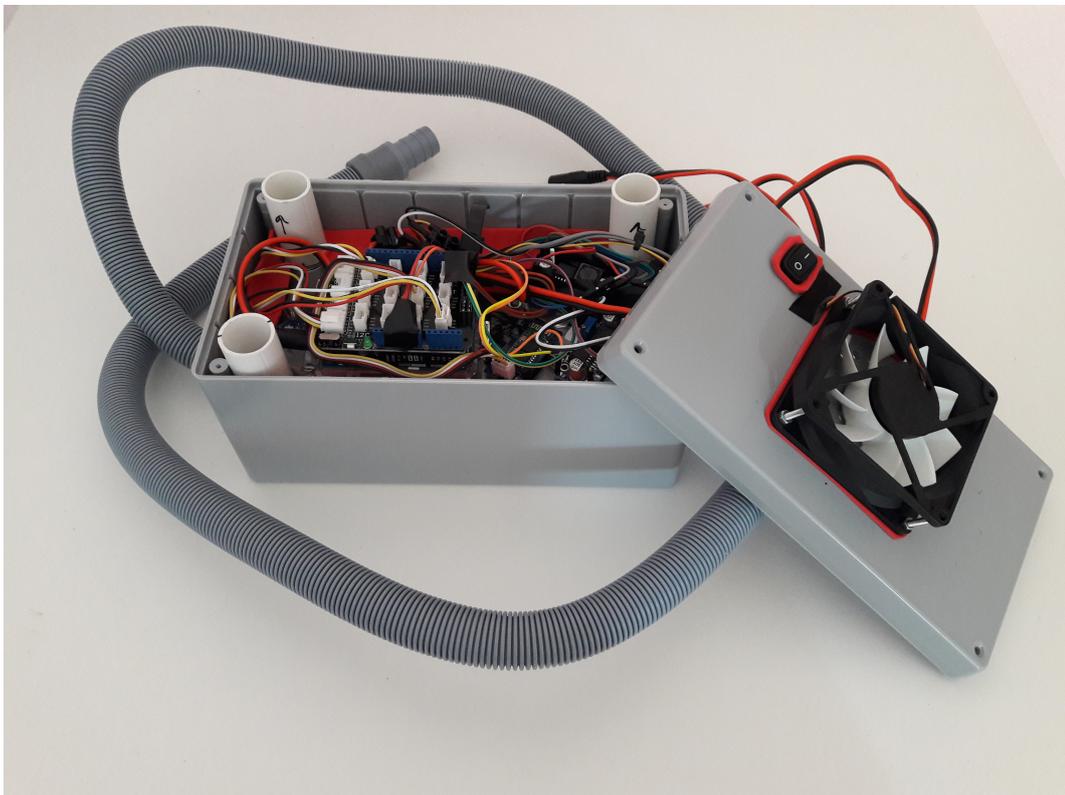


Abbildung 5.10: Foto des Messgeräts

5.5.2 Implementierung der Mikrocontroller-Software

Die Software des Mikrocontrollers wurde in der Sprache C++ in der Arduino Entwicklungsumgebung programmiert, welche automatisch Bibliotheken für die Funktionen des Boards einbindet. Zum Auslesen des Mehrkanal-Gassensors wurde die Bibliothek des Sensorherstellers verwendet. Für den Seedstudio TH02 Temperatur- und Luftfeuchtesensor [54] und den Nova SDS011 Feinstaubsensor [11] wurden gemäß den Datenblättern Unterfunktionen zum Auslesen der Sensoren erstellt. Beim Nova SDS011 werden die über das UART-Interface ankommenden Daten anhand von Head- und Tail-Byte, sowie einer Checksumme überprüft. Für die Kommunikation wird die Arduino-Bibliothek "Serial" verwendet. Zum Auslesen des Seedstudio TH02 wird eine Software-I2C-Bibliothek eingesetzt. Die Messwerte werden durch Abfrage entsprechender Register des Sensors gelesen. Zum Auslesen des ADC, an den der VOC-Sensor angeschlossen ist, wird die Funktion "AnalogRead" der Arduino-Bibliothek verwendet. An das Bluetooth-Modul verschickt werden die Daten über eine serielle Verbindung. Hierfür wird die Arduino-Bibliothek "SoftwareSerial" eingesetzt. Die Hauptschleife des Programms wurde gemäß des Aktivitätsdiagramms aus 5.4.3 programmiert.

5.5.3 Implementierung der Android-Applikation

Die Android-Applikation wurde dem Entwurf aus 5.4.4 nach mit der Entwicklungsumgebung Android-Studio in der Programmiersprache Java implementiert. Das Layout der Benutzeroberfläche ist im XML-Format. Die Zielplattform ist ein Gerät mit Android 7.0 oder höher.

Die Abbildung 5.11 zeigt die Bedienoberfläche der Applikation. Mit der Schaltfläche im linken oberen Rand lässt sich ein Messgerät per Bluetooth verbinden und trennen. Die Schaltfläche im rechten oberen Rand startet und beendet eine Messung. Mit dem Umschalter neben dem "Transmit data" Feld kann das Senden der Messwerte zum Server ein- und ausgeschaltet werden. Im Bereich darunter werden die aktuell gemessenen Werte, sowie die Position und die Uhrzeit angezeigt. Unter der Positionsinformation werden die Status von aktueller Messung, Serververbindung und Bluetooth-Verbindung zum Messgerät angezeigt. Mit dem Umschalter "Restriction" kann die Messung auf einen Temperatur- und Luftfeuchtigkeitsbereich eingeschränkt werden, wobei die Intervalle in die Eingabefelder darunter eingetragen werden müssen. Alle Messwerte, die bei einer Temperatur oder bei einer Luftfeuchtigkeit außerhalb des Bereiches aufgenommen wurden, werden verworfen. Mit dem Umschalter "Get R0" kann bei einer Messung der Nullwiderstand der Gassensoren zur Kalibrierung aufgenommen werden.

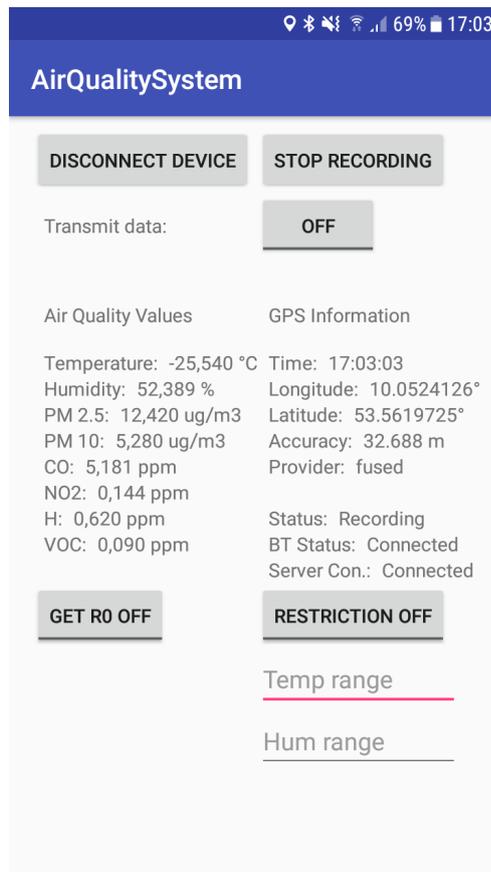


Abbildung 5.11: Grafische Oberfläche der Android-Applikation

Für das Observer-Pattern wurde die Implementierung aus der Java Util Bibliothek verwendet. Die Benachrichtigungen an die Klasse "MainActivity" sind stets String Konstanten, die dort definiert sind. In der "update" Methode werden diese dann interpretiert und entsprechende Aktionen ausgelöst.

Die Listener für die Schaltflächen der Benutzeroberfläche werden in entsprechenden Methoden der "MainActivity" initialisiert. Im Falle der Schaltfläche zum Verbinden eines Messgeräts wird über den Listener ein Handler gestartet, der die einzelnen Schritte des Verbindungsaufbaus durchläuft, da dies mit Nutzerinteraktion verbunden ist. Würde dies vom Listener der Schaltfläche gemacht werden, würde der UI-Thread anhalten, was zu vermeiden ist.

Um die Positionsinterpolation bei Tunnelfahrten umzusetzen, wurde der Klasse "SyncBuffer"

ein zweiter Puffer hinzugefügt. Ist die aktuelle Position seit mehr als 5 Sekunden nicht aktualisiert worden, werden alle Messungen mit der letzten, nicht mehr aktuellen Position, aber mit der aktuellen Zeit dort gespeichert. Sobald die Position wieder aktualisiert wurde, wird ein Handler gestartet, der in allen Datenobjekten des Puffers anhand der Zeit- und Wegdifferenz die interpolierte Position berechnet. Anschließend werden die Datenobjekte in den Hauptpuffer übertragen, von wo aus sie gesendet werden können.

Um die Nullwiderstände der Gassensoren zu messen, wird beim Berühren der Schaltfläche "Get R0" in der Klasse "ReceiverConverter" die Umrechnung der Messwerte ausgeschaltet und es werden keine Datenobjekte zum Versenden in den Puffer eingefügt. Alle vom Mikrocontroller ankommenden Werte werden gespeichert, um bei erneuter Betätigung des Umschalters Mittelwerte zu bilden.

5.6 Evaluation

5.6.1 Durchführung einer Messung

Das umgesetzte System wurde unter anderem in der Hamburger Innenstadt, den Stadtteilen St. Georg, Sternschanze, Stellingen, Schnelsen, dem Vorort Norderstedt und der ländlichen Gemeinde Kisdorf eingesetzt, wobei mehrere kontinuierliche Messung durchgeführt wurden. Die Messungen fanden am Freitag, den 28.07.2017 zwischen 13:00 Uhr und 17:00 Uhr statt.

Das Messgerät wurde auf dem Beifahrersitz des verwendeten Pkw platziert, wobei der Ansaugschlauch aus dem Seitenfenster gehangen wurde. Das Ende des Schlauches ragte etwa 5 cm aus der Fensteröffnung, sodass sich das Ende des Schlauches in einer Höhe von etwa 1,6 m befand. Dabei war die Öffnung des Schlauches zur Seite gerichtet. Über das Bordnetz des Pkw wurde das Messgerät mit Strom versorgt. Als Smartphone wurde ein Samsung Galaxy A3 2016 eingesetzt, welches über die nötigen Funktionen verfügt.

Vor den Messungen musste das Gerät jeweils etwa 20 Minuten eingeschaltet sein, damit die Sensoren vorwärmen und sich eine stabile Temperatur im Gehäuse einstellen konnte. Danach wurden die Nullwiderstände der Gassensoren gemessen, um sie für den bestehenden Temperatur- und Luftfeuchtigkeitsbereich zu kalibrieren. Die Temperatur im Gehäuse durfte bei den Messungen zwischen 25 °C und 35 °C, die relative Luftfeuchtigkeit zwischen 40 % und 60 % schwanken. Messungen außerhalb dieser Bereiche wurden nicht gewertet, da davon auszugehen ist, dass dies die Gasmessung zu stark beeinflusst.

5.6.2 Ergebnis der Messung

Das System arbeitete bei den Messungen wie erwartet. Alle 20 Sekunden wurde ein Satz an Messwerten vom Messgerät zum Smartphone übertragen und an den Server gesendet. Obwohl keine Fehler auftraten fiel auf, dass die per GPS ermittelte Position teilweise mehr als 40 Meter vom tatsächlichen Standort abwich. Auch fiel das GPS-Signal mehrmals aus, wobei die Software in diesen Fällen die Position nachträglich linear interpolierte. Die Verbindung zum Server und die Bluetooth-Verbindung zwischen Messgerät und Smartphone waren stets stabil.

Die gemessenen Stickstoffdioxid-Werte waren durchschnittlich insgesamt höher im Stadtgebiet als im Vorort und im ländlichen Bereich. In der Stadt wurden Werte von 0,15 ppm bis 0,3 ppm gemessen, wogegen die Werte im Vorort und in der ländlichen Gemeinde zwischen 0,07 ppm und 0,1 ppm lagen. Die Stickstoffdioxidbelastung schien jedoch auch stark in Verbindung mit dem Verkehrsaufkommen und dem Verkehrsfluss zu hängen. So wurde auch im Vorort bei hohem Verkehrsaufkommen im Stau ein Wert von 0,32 ppm gemessen. Auf der Kieler Straße in Hamburg mit drei voll belegten Spuren bei leicht stockendem Verkehr wurde ebenfalls ein Wert von 0,31 ppm gemessen.

Die Feinstaubbelastung war bei Stau besonders hoch. Auf der Ohechaussee in Norderstedt, wo sich zum Zeitpunkt der Messung der Verkehr mehrere Minuten lang staute, reichte der PM10 Messwert bis auf $2,2 \mu\text{g}/\text{m}^3$. Auch im Stadtteil St.Georg wurden bei ähnlichen Verkehrsbedingungen $1,41 \mu\text{g}/\text{m}^3$ PM10 gemessen. Abgesehen von Messspitzen im Stau lag die Feinstaubbelastung jedoch meist unter $1 \mu\text{g}/\text{m}^3$.

Zwischen den Kohlenmonoxid-Werten und dem Verkehrsaufkommen konnte kein sinnvoller Zusammenhang gefunden werden. Der Wert schwankte meist zwischen 5,8 ppm und 6 ppm. Lediglich in der Innenstadt und in St. Georg fiel der Wert kurzzeitig um 1 ppm. Hierfür konnte jedoch keine Erklärung gefunden werden. Auch der VOC-Wert schien in keinem Zusammenhang zum Ort und dem Verkehrsaufkommen zu stehen. In Bezug auf die Ammoniakbelastung ließ sich feststellen, dass diese in der Stadt im Durchschnitt niedriger war als im Vorort oder im ländlichen Gebiet. Es fiel jedoch auf, dass eine Korrelation zwischen den Ergebnissen der Sensoren besteht, da sie stets gemeinsam anstiegen oder sanken.

In Tabelle 5.4 sind die gemessenen Werte auszugsweise aufgelistet, wobei dies stets die Mittelwerte der Messungen eines Bereiches sind. Hierbei steht T für die Temperatur und HUM für die relative Luftfeuchtigkeit im Gehäuse des Messgerätes. Kohlenmonoxid wird mit CO abgekürzt,

Stickstoffdioxid mit NO₂ und Ammoniak mit NH₃. Das Verkehrsaufkommen beschreibt die Belegung der vorhandenen Fahrspuren und die Geschwindigkeit des Verkehrs.

Tabelle 5.4: Ergebnisse der Luftgütemessung an verschiedenen Orten

Messort	Verkehrsaufkommen	T	HUM	PM2.5	PM10	CO	NO ₂	NH ₃	VOC
Steintorwall (Innenstadt)	Hoch, stockender Verkehr	30,36 °C	45,56 %	0,81 µg/m ³	0,93 µg/m ³	4,87 ppm	0,15 ppm	0,62 ppm	0,13 ppm
Berlinertor-Damm (St. Georg)	Hoch, stau mehrere Minuten lang	27,07 °C	54,51 %	0,98 µg/m ³	1,41 µg/m ³	4,83 ppm	0,25 ppm	0,56 ppm	0,08 ppm
Stresemannstraße (Sternschanze)	Hoch, leicht stockender Verkehr	30,92 °C	47,09 %	0,6 µg/m ³	0,7 µg/m ³	5,87 ppm	0,21 ppm	0,97 ppm	0,26 ppm
Kieler Straße (Stellingen)	Sehr hoch, leicht stockender Verkehr	30,37 °C	43,83 %	0,67 µg/m ³	0,7 µg/m ³	5,76 ppm	0,31 ppm	0,85 ppm	0,18 ppm
Holsteiner Chaussee (Schnelsen)	Hoch, stockender Verkehr	28,37 °C	50,19 %	0,69 µg/m ³	0,7 µg/m ³	5,8 ppm	0,26 ppm	0,86 ppm	0,22 ppm
Ohechaussee (Norderstedt)	Hoch, Stau mehrere Minuten lang	28,16 °C	54,38 %	1,1 µg/m ³	2,2 µg/m ³	5,73 ppm	0,32 ppm	0,78 ppm	0,12 ppm
Ulzburger Straße (Norderstedt)	Geringes Aufkommen	31,18 °C	43,23 %	0,67 µg/m ³	0,7 µg/m ³	5,92 ppm	0,09 ppm	1,48 ppm	0,22 ppm
Kisdorf	Sehr geringes Aufkommen	31,33 °C	42,53 %	0,69 µg/m ³	0,7 µg/m ³	5,95 ppm	0,07 ppm	1,93 ppm	0,23 ppm

Zusammengefasst kann gesagt werden, dass die Feinstaub- und Stickstoffdioxid-Messungen nachvollziehbare Ergebnisse lieferten. Wie erwartet waren diese vom jeweiligen Verkehrsaufkommen und dem Verkehrsfluss abhängig. Denkbar ist, dass diese Ergebnisse zustande kamen, da sowohl Feinstaubsensor, als auch Stickstoffdioxid-Sensor sich in nur geringem Maße von Störfaktoren beeinflussen ließen. Schlüssige Zusammenhänge zwischen den gemessenen Kohlenmonoxid-Werten und dem Verkehr konnten nicht hergestellt werden. Ursache hierfür

könnte das geringe Vorkommen des Zielgases im Verhältnis zu Störgrößen sein. Auch lieferte der VOC-Sensor keine direkt nachvollziehbaren Ergebnisse. Erwartungsgemäß war jedoch, dass die gemessene Ammoniakbelastung im ländlichen Bereich deutlich größer war als im Stadtgebiet. Es fiel auf, dass die Messwerte aller Gassensoren, die oxidierbare Gase messen, also Kohlenmonoxid-, Ammoniak- und VOC-Sensor, stets gemeinsam anstiegen oder fielen. Diese Beobachtung stimmt mit der geringen Selektivität der Sensoren überein, da jeder Sensor auch auf die Zielgase der anderen beiden reagiert.

5.7 Methode zur Absolutwertbestimmung und Kalibrierung

Aufgrund verschiedener Eigenschaften von Halbleiter-Gassensoren ist es mit ihnen nicht möglich ohne eine Kalibrierung zuverlässig die Absolutwerte von Gaskonzentrationen zu messen. Diese Eigenschaften sind die Abhängigkeit von den Umwelteinflüssen Temperatur und Luftfeuchtigkeit sowie die geringe Selektivität bezogen auf die verschiedenen Gase, die neben dem jeweils zu messenden Zielgas vorkommen können. Alters- oder Produktionsbedingte Abweichungen des Sensorwiderstandes werden rechnerisch kompensiert, indem das Sensorsignal für die Weiterverarbeitung stets das Verhältnis von aktuellem Widerstand des Sensors zu seinem Nullwiderstand ist. Dabei ist der Nullwiderstand jener Widerstand, der am Sensor gemessen werden kann, wenn er keinem Gas ausgesetzt ist, welches die Leitfähigkeit der Messoberfläche beeinflusst.

Wie anhand der Sensorkennlinien in Abbildung 5.1 des Abschnittes 5.4.1 zu sehen ist, reagieren die Sensoren mit unterschiedlichen Sensitivitäten auf die Verschiedenen Gaskonzentration in der Luft. Somit entstehen je nach Gaszusammensetzung charakteristische Kombination der Sensorsignale. Mit einer Funktion über diese Signale, können die spezifischen Gaskonzentrationen näher bestimmt werden. In der Arbeit von H. Baba und Z. Dibi [6], wird eine Methode beschrieben wie künstliche neuronale Netze dazu eingesetzt werden können, anhand der Signale verschiedener Halbleiter-Gassensoren jeweils die Konzentration eines spezifischen Gases zu bestimmen. Das neuronale Netz dient hierbei der Funktionsapproximation, wobei zu jedem Zeitschritt als Argument ein Vektor mit allen Sensorsignalen, der Temperatur und der Luftfeuchtigkeit übergeben wird. Die Ausgabe des Netzes ist wiederum ein Vektor mit den einzelnen Gaskonzentrationen. In der Arbeit aus [6] wurden für die Kompensation der Umwelteinflüsse und der Bestimmung der spezifischen Gaskonzentrationen zwei verschiedene Netze eingesetzt, was jedoch nicht zwingend erforderlich ist. Zum Bestimmen der Parameter für die Funktionsapproximation, also das Einstellen der Gewichte des neuronalen Netzes, muss das

Netz trainiert werden. Hierzu wird ein Datensatz erzeugt, indem die Sensoren verschiedenen bekannten Gaszusammensetzungen ausgesetzt werden, wobei jeweils die Sensorsignale und die korrespondierenden Gaszusammensetzungen gespeichert werden. Ebenso wird das Verhalten bei unterschiedlichen Temperaturen und Luftfeuchtheitswerten aufgenommen. Die Datensätze werden anschließend dazu verwendet die Netze mit dem Backpropagation-Verfahren zu trainieren. In der Arbeit von D. Lee et al. [2], wird ein ähnliches Verfahren beschrieben. Anders als in [6] werden hierzu jedoch keine im Handel erhältlichen Sensoren verwendet, sondern es wird ein Gassensor-Array hergestellt, welches geringere Abmessungen aufweist als einzelne Sensoren.

Bezogen auf das hier entworfene System ließe sich ein künstliches neuronales Netz sowohl als Teil der Android-Applikation als auch serverseitig implementieren. In Beiden Fällen müssten entsprechende Datensätze mit den verwendeten Sensoren zur Kalibrierung erzeugt werden. Auf diese Weise könnten die Absolutwerte der Gaskonzentrationen in einem gewissen Toleranzbereich erfasst werden.

6 Zusammenfassung und Fazit

6.1 Zusammenfassung

In der vorliegenden Arbeit wurden aktuell verfügbare und von der Industrie antizipierte automobiler Cloud-Dienste analysiert und zusammenfassend dargestellt. Das Spektrum reicht von heute gängigen Diensten wie Infotainment- und Navigationsdiensten bis hin zu sich noch in der Entwicklung befindenden Technologien wie etwa das Cloud-basierte Deep Learning zum Optimieren der Systeme von autonomen und teilautonomen Fahrzeugen.

Um Beispiele für eine Erweiterung dieses Spektrums aufzuzeigen, wurden eigene Ideen entwickelt und die Dienste entsprechend konzeptioniert. Dabei lag der Fokus darauf die grundlegenden Funktionsprinzipien darzustellen. Hierzu wurden in einigen Beispieldiensten auch mögliche Anwendungsfälle herangezogen. Die konzeptionierten Neuentwicklungen basieren sowohl auf heute in Fahrzeugen verbauten Bordsystemen, darunter die Sensorik und Aktorik, sowie auch auf Technologien, die zukünftig zu erwarten sind.

Als konkretes Beispiel für die Umsetzung einer Idee aus dem Bereich automobiler Cloud-Dienste wurde ein Dienst zur Erfassung der Luftgüte entlang der Fahrbahn entwickelt. Hierzu wurde ein Messgerät mit entsprechender Sensorik und eine Android-Applikation entworfen und umgesetzt. Für das Backend wurde eine bereits bestehende Plattform verwendet. Obwohl das System fahrzeugunabhängig ausgelegt wurde, wurde dennoch der Einsatz typischer Fahrzeugsensorik und anderer Bordsysteme diskutiert. Mit dem entwickelten Messsystem wurde dann in der Hamburger Innenstadt und der Umgebung eine relative Messung durchgeführt. Anschließend wurden die Ergebnisse ausgewertet. Die Ergebnisse zeigten wie erwartet, dass die Stickstoffdioxid- und Feinstaubbelastung im innerstädtischen Bereich höher ist, als in ländlichen Bereichen. Vor allem aber wurde gezeigt, dass die Belastung im Stau und bei hohem Verkehrsaufkommen um ein Vielfaches steigt.

6.2 Fazit

Zurzeit existiert bereits eine Vielzahl automobiler Cloud-Dienste, wobei diese technologisch oft noch nicht ausgereift sind. Besonders solche, die auf Crowdsourcing angewiesen sind, können wegen der kleinen Anzahl an dafür vorbereiteten Fahrzeugen ihr volles Potential nicht ausschöpfen. Ausgenommen hiervon sind Navigations- und Infotainmentdienste, die plattformübergreifend arbeiten, also beispielsweise auch mit dem Smartphone zugänglich sind. Mit der steigenden Vernetzung von Fahrzeugen, der Vielzahl an beteiligten Unternehmen und dem breiten Spektrum an Möglichkeiten ist zu erwarten, dass in Zukunft weit mehr Dienste verfügbar sein werden. Bei Betrachtung der von der Industrie antizipierten Dienste wird klar, dass hierzu auch immer mehr die Fahrzeugsensorik als Informationsquelle dienen wird. Die massenhaft gesammelten Daten können zentral auf Servern ausgewertet werden, um so Erkenntnisse über den Straßenverkehr und die Verkehrsteilnehmer zu gewinnen. Auch werden Fahrzeuge immer mehr autonom untereinander und mit ihrer Umgebung kommunizieren, um sich zu koordinieren.

Die Analyse der aktuellen und aufkommenden Dienste, sowie auch die Entwicklung eigener Ideen hat gezeigt, dass sich automobiler Cloud-Dienste nicht nur auf den Straßenverkehr beziehen müssen. Es ist denkbar, dass Pkw und Lkw in Zukunft mehr als ein Transportmittel darstellen und für die verschiedensten Aufgaben eingesetzt werden können. Beispiele hierfür sind die Nutzung des Kofferraums als Paketstation und die Ergänzung von städtischen Überwachungskameras durch die Außenkameras von Fahrzeugen.

Mit der Entwicklung des Dienstes zur Erfassung der Luftgüte wurde demonstriert wie ein solches System aufgebaut sein kann. Mit der entwickelten Hard- und Software gelang es eine relative Messung der Luftqualität aus dem Auto heraus durchzuführen. Trotz des prototypischen Charakters des Systems lieferte es weitestgehend nachvollziehbare Ergebnisse. Eine Weiterentwicklung, wie die Anwendung der hier beschriebenen Methode zur Absolutwertbestimmung und Kalibrierung, erscheint daher sinnvoll. Hierbei könnte es sowohl als fahrzeugunabhängiges Messsystem eingesetzt werden, oder aber ein fester Bestandteil bestimmter Fahrzeuge sein.

Literaturverzeichnis

- [1] AGNES VOGT: Big Data: Von Kap zu Kap mit einem Auto voller Daten. (2015), März.
– URL <http://www.automobilwoche.de/article/20150312/NACHRICHTEN/150319981/big-data-von-kap-zu-kap-mit-einem-auto-voller-daten>. – [Abgerufen am 10.05.2017]
- [2] DAE-SIK LEE, SANG-WOO BAN, MINHO LEE, DUK-DONG LEE: Micro Gas Sensor Array With Neural Network for Recognizing Combustible Leakage Gases. In: *IEEE Sensors Journal* 5 (2005), Juni
- [3] DIETER SCHLENZ (Hrsg.): *Pkw-Klimatisierung 4, Klimakonzepte, Zuheizkonzepte, Regelungsstrategien und Entwicklungsmethoden*. Renningen : Expert-Verlag, 2015
- [4] FIGARO Sensors (Veranst.): *TGS 3870-B00*. O.J.. – Rev. 12/13
- [5] H. WINNER, S. HAKULI, F. LOTZ, C. SINGER (Hrsg.): *Handbuch Fahrerassistenzsysteme, Grundlagen, Komponenten und Systeme für Aktive Sicherheit und Komfort*. 3. Wiesbaden : Springer Fachmedien, 2015
- [6] HAKIM BABA, ZOHIR DIBI: A Novel Neural Network-Based Technique for Smart Gas Sensors Operating in a Dynamic Environment. In: *MDPI Journal* (2009), Oktober
- [7] K. REIF (Hrsg.): *Sensoren im Kraftfahrzeug, Bosch Fachinformation Automobil*. 3. Wiesbaden : Springer Fachmedien, 2016
- [8] LIU NANJIE: Internet of Vehicles: Your Next Connections. (O.J.). – URL <http://www1.huawei.com/enapp/28/hw-110836.htm>. – [Abgerufen am 03.05.2017]
- [9] MARTIN BAITINGER: smart ready to share: privates Carsharing. (2017), Januar. – URL <http://www.smartpit.de/smart-ready-to-share-privates-carsharing/>. – [Abgerufen am 09.05.2017]

- [10] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, PETER MELL, TIMOTHY GRANCE: The NIST Definition of Cloud Computing. (2011), September. – URL <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>. – [Abgerufen am 03.05.2017]
- [11] Nova Fitness Co, Ltd. (Veranst.): *Laser PM2.5 Sensor specification*. Oktober 2015. – V. 1.3
- [12] O.V.: Audi liefert mit DHL und Amazon das Komfort-Paket. (2015), April. – URL <https://www.audi-mediacycenter.com/de/pressemitteilungen/audi-liefert-mit-dhl-und-amazon-das-komfort-paket-347>. – [Abgerufen am 09.05.2017]
- [13] O.V.: Green Light for Vehicle-to-Infrastructure Communications. (2015), September. – URL <https://www.siemens.com/innovation/en/home/pictures-of-the-future/mobility-and-motors/urban-mobility-green-light-for-vehicle-to-infrastructure-communications.html>. – [Abgerufen am 18.05.2017]
- [14] O.V.: Ambient (outdoor) air quality and health. (2016), September. – URL <http://www.who.int/mediacentre/factsheets/fs313/en/>. – [Abgerufen am 08.06.2017]
- [15] O.V.: Anhang Datenschutz und Datensicherheit. (2016), August. – URL https://www.audi-mediacycenter.com/de/techday_connectivity-6597/anhang-datenschutz-und-datensicherheit-6604. – [Abgerufen am 08.05.2017]
- [16] O.V.: Kofferraum als Briefkasten: DHL und Smart gehen zusammen. (2016), Juli. – URL <http://www.sueddeutsche.de/news/wirtschaft/auto-kofferraum-als-briefkastendhlund-smart-gehen-zusammen-dpa.urn-newsml-dpa-com-20090101-160724-99-801509>. – [Abgerufen am 09.05.2017]
- [17] O.V.: Leading the World to 5G, Cellular Vehicle-to-Everything (C-V2X) technologies. (2016), Juni. – URL <https://www.qualcomm.com/documents/path-5g-cellular-vehicle-everything-c-v2x>. – [Abgerufen am 04.05.2017]
- [18] O.V.: Luftschadstoffe im Überblick. (2016). – URL <http://www.umweltbundesamt.de/themen/luft/luftschadstoffe-im-ueberblick>. – [Abgerufen am 11.07.2017]

- [19] O.V.: Partner für Globale Fahrzeugvernetzung. (2016), September. – URL <https://www.t-systems.com/de/de/ueber-uns/unternehmen/newsroom/news/news/daimlerconnected-car-plattform-384902>. – [Abgerufen am 04.05.2017]
- [20] O.V.: Pilotprojekt von Bosch und Daimler zu Community-based Parking: Ihr Mercedes-Benz wird zur Parkplatz-Suchmaschine. (2016), September. – URL <http://media.daimler.com/marsMediaSite/de/instance/ko/Pilotprojekt-von-Bosch-und-Daimler-zu-Community-based-Parking-Ihr-Mercedes-Benz-wird-zur-Parkplatz-Suchmaschine.xhtml?oid=13521746>. – [Abgerufen am 09.05.2017]
- [21] O.V.: SAP Solution Brief, SAP Vehicles Network. (2016). – URL <https://www.sap.com/documents/2016/03/aa90c769-627c-0010-82c7-eda71af511fa.html>. – [Abgerufen am 08.05.2017]
- [22] O.V.: Emergency Vehicle Priority. (2017), Februar. – URL <http://www.tmr.qld.gov.au/Safety/Road-safety/Emergency-Vehicle-Priority.aspx>. – [Abgerufen am 18.05.2017]
- [23] O.V.: smart startet privates Carsharing: Mein smart, Dein smart, unser smart. (2017), Januar. – URL <http://media.daimler.com/marsMediaSite/de/instance/ko/smart-startet-privates-Carsharing-Mein-smart-Dein-smart-unser-smart.xhtml?oid=15178276>. – [Abgerufen am 09.05.2017]
- [24] O.V.: ADASIS - Advancing map-enhanced driver assistance systems. (O.J.). – URL <http://adasis.org/>. – [Abgerufen am 09.05.2017]
- [25] O.V.: Air View | Google. (O.J.). – URL <https://environment.google/projects/airview/>. – [Abgerufen am 08.06.2017]
- [26] O.V.: AirCasting. (O.J.). – URL <http://aircasting.org/>. – [Abgerufen am 08.06.2017]
- [27] O.V.: Aktuelle Luftdaten | Umweltbundesamt. (O.J.). – URL http://www.umweltbundesamt.de/daten/luftbelastung/aktuelle-luftdaten#/. – [Abgerufen am 26.07.2017]
- [28] O.V.: Android Auto. (O.J.). – URL https://www.android.com/intl/de_de/auto/. – [Abgerufen am 07.05.2017]

- [29] O.V.: Audi Connect. (O.J.). – URL <https://www.audi.de/connect>. – [Abgerufen am 07.05.2017]
- [30] O.V.: BMW ConnectedDrive : BMW Connected. (O.J.). – URL <https://www.bmw.de/de/topics/faszination-bmw/connecteddrive/bmw-connected.html>. – [Abgerufen am 07.05.2017]
- [31] O.V.: Car-To-X Kommunikation, Nicht länger Science Fiction. (O.J.). – URL http://www.mercedes-benz.de/content/germany/mpc/mpc_germany_website/de/home_mpc/passengercars/home/world/innovation/news/car_to_x.html. – [Abgerufen am 19.05.2017]
- [32] O.V.: Crossbar.io. (O.J.). – URL <http://crossbar.io/>. – [Abgerufen am 23.07.2017]
- [33] O.V.: Der Deep Learning Server DGX-1. (O.J.). – URL <http://www.nvidia.de/object/deep-learning-system-de.html>. – [Abgerufen am 09.05.2017]
- [34] O.V.: Gemeinsam schneller zur freien Parklücke mit Community-based Parking. (O.J.). – URL <http://www.bosch-mobility-solutions.de/de/highlights/vernetzte-mobilit%C3%A4t/community-based-parking/>. – [Abgerufen am 09.05.2017]
- [35] O.V.: Google Maps. (O.J.). – URL <http://maps.google.com/>. – [Abgerufen am 23.07.2017]
- [36] O.V.: HERE Electronic Horizon. (O.J.). – URL <https://here.com/en/products-services/products/here-electronic-horizon>. – [Abgerufen am 09.05.2017]
- [37] O.V.: HERE HD Live Map | On the road towards anonomous driving. (O.J.). – URL <https://here.com/en/products-services/products/here-hd-live-map>. – [Abgerufen am 09.05.2017]
- [38] O.V.: iOS - CarPlay - Apple(DE). (O.J.). – URL www.apple.com/de/ios/carplay. – [Abgerufen am 07.05.2017]
- [39] O.V.: Kaa IoT Product Development Platform - IoT Application Enablement. (O.J.). – URL <https://www.kaaiot.io/>. – [Abgerufen am 23.07.2017]

- [40] O.V.: luftdaten.info - Feinstaub selber messen - Open Data und Citizen Science aus Stuttgart. (O.J.). – URL <http://luftdaten.info/>. – [Abgerufen am 08.06.2017]
- [41] O.V.: Maps for Self-Driving Cars | Maps for Automated Driving. (O.J.). – URL <http://www.sanborn.com/highly-automated-driving-maps-for-autonomous-vehicles/>. – [Abgerufen am 09.05.2017]
- [42] O.V.: Mirror Link. (O.J.). – URL www.mirrorlink.com. – [Abgerufen am 07.05.2017]
- [43] O.V.: NVIDIA Drive PX 2 | Entwicklungsplattform für selbstfahrende Autos. (O.J.). – URL <http://www.nvidia.de/object/drive-px-de.html>. – [Abgerufen am 09.05.2017]
- [44] O.V.: Open Platform - Mojio. (O.J.). – URL <https://www.moj.io/connected-car-platform/>. – [Abgerufen am 08.05.2017]
- [45] O.V.: Paragon - PM 2.5 Partikelsensor. (O.J.). – URL <http://www.paragon.ag/geschaeftsbereiche/sensoren/luftguetemanagement/pm-25-partikelsensor.html>. – [Abgerufen am 08.06.2017]
- [46] O.V.: POTHOLE DETECTION TECHNOLOGY RESEARCH ANNOUNCED BY JAGUAR LAND ROVER. (O.J.). – URL <https://www.landrover.com/experiences/news/pothole-detection.html>. – [Abgerufen am 10.05.2017]
- [47] O.V.: Road Experience Manager (REM) - MobileEye. (O.J.). – URL <http://www.mobileye.com/our-technology/rem/>. – [Abgerufen am 09.05.2017]
- [48] O.V.: Smart Citizen, Open source technology for citizens political participation in smarter cities. (O.J.). – URL <https://smartcitizen.me/>. – [Abgerufen am 08.06.2017]
- [49] O.V.: Smart ready to drop, Macht den Kofferraum zur Paketbox. (O.J.). – URL <https://www.smart.com/de/de/index/ready-to-drop.html>. – [Abgerufen am 09.05.2017]
- [50] O.V.: Telematics : UPS. (O.J.). – URL <https://www.ups.com/content/us/en/bussol/browse/leadership-telematics.html>. – [Abgerufen am 08.05.2017]
- [51] O.V.: TomTom HD Map & RoadDNA | TomTom Automotive. (O.J.). – URL <http://automotive.tomtom.com/products-services/hd-map-roaddna/>. – [Abgerufen am 09.05.2017]

- [52] O.V.: Volkswagen Technik Lexikon. (O.J.). – URL <http://www.volkswagen.de/de/technologie/technik-lexikon.html>. – [Abgerufen am 04.05.2017]
- [53] O.V.: WAMP - Web Application Messaging Protocol. (O.J.). – URL <http://wamp-proto.org/>. – [Abgerufen am 01.08.2017]
- [54] Seedstudio (Veranst.): *Si7005 Datasheet*. O.J.. – Rev. 0.1
- [55] SGX Sensortech (Veranst.): *MiCS-6814 Datasheet*. O.J.. – Rev. 7
- [56] Shinyei (Veranst.): *Specification sheet of PPD42NS Particle Sensor*. 3. 2013
- [57] THE NATIONAL INSTITUTE FOR OCCUPATIONAL SAFETY AND HEALTH: Immediately Dangerous to Life or Health Values. (1994). – URL <https://www.cdc.gov/niosh/idlh/intridl4.html>. – [Abgerufen am 11.07.2017]
- [58] VOLKER JOHANNING, ROMAN MILDNER (Hrsg.): *Car IT Kompakt, Das Auto der zukunft - Vernetzt und autonom fahren*. Wiesbaden : Springer Fachmedien, 2015
- [59] V.P. ANDELFINGER, T. HÄNISCH (Hrsg.): *Internet der Dinge, Technik, Trends, Geschäftsmodelle*. Wiesbaden : Springer Fachmedien, 2015
- [60] WILKENING, Yannic: *Fahrzeuge als Sensorknoten für Big Data Cloud-Anwendungen am Beispiel Parkraumvorhersage*, HAW Hamburg, Bachelorarbeit, April 2017
- [61] Y. FANGCHUN, W. SHANGGUANG, L. JINGLIN, L. ZHIHAN, S. QIBO: An Overview of Internet of Vehicles. (2014), Oktober. – URL <http://www.sguangwang.com/PDF/An%20Overview%20of%20Internet%20of%20Vehicles.pdf>. – [Abgerufen am 03.05.2017]

Abkürzungsverzeichnis und Glossar

CCTV Closed Circuit Television

VOC Volatile Organic Compounds

PM Particulate Matter

LIDAR Light Detection and Ranging

TOF Time of Flight (Laufzeitverfahren)

Connectivity-Unit Hardware Modul für die Internetanbindung

Head-Unit Steuereinheit des Mediensystems in Fahrzeugen

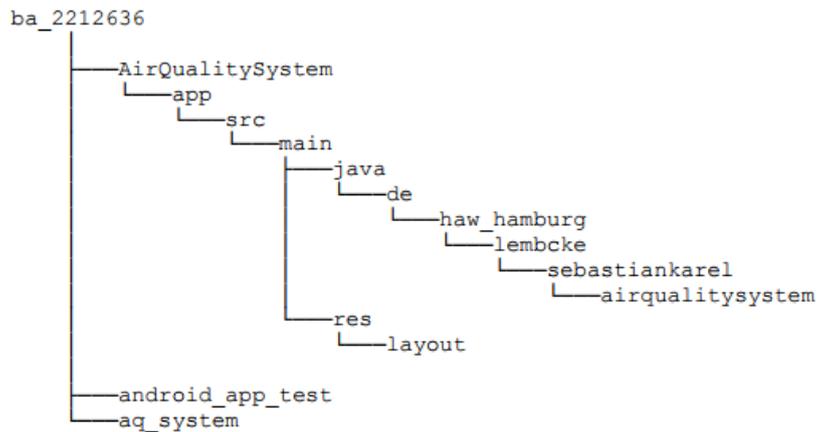
HD High Definition

ADAS Advanced Driver Assistance Systems

IoT Internet of Things

Verzeichnisstruktur des Datenträgers

Dargestellt sind die relevanten Verzeichnisse des Datenträgers. Der Quelltext der Android-Applikation befindet sich im Verzeichnis "airqualitysystem". Das Layout der grafischen Oberfläche ist im Verzeichnis "layout". Im Verzeichnis "android_app_test" ist die Arduino-Software zum Testen der Übertragung an das Smartphone. Die Arduino-Software, die im Messgerät eingesetzt wurde, befindet sich im Verzeichnis "aq_system".



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

Hamburg, 8. August 2017

Sebastian Karel Lembcke