



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Bachelorarbeit

Medieninformatik

Ampelphasen-Informationssystem für FahrradfahrerInnen
auf Grundlage persistenter geo- und zeitbasierter Daten

Berlin, den 22. Januar 2015

Autorin:

Jacoba BRANDNER

Matrikelnummer:

786635

Betreuerin:

Frau Prof. Dr. Gudrun GÖRLITZ

Gutachterin:

Frau Prof. Dr. Petra SAUER

INHALT

1 Einführung	4
1.1 Motivation	4
1.2 Zielstellung	4
1.3 Aufbau der Arbeit	5
2 Bestehende Konzepte	6
2.1 Ampelinformationssysteme	6
2.1.1 Grüne Welle auf Radwegen	6
2.1.2 Individuelle Ampelinformationssysteme im Auto	7
2.2 Ampelinformationssysteme als mobile Applikation	10
2.3 Fahrraderweiterungen	11
2.3.1 Displaylose Fahrradnavigation	11
2.3.2 Intelligente Fahrradlenker	11
2.3.3 Das Samsung Smart Bike	12
2.3.4 Der COBI Fahrradcomputer	13
2.4 Analyseergebnis	13
3 Lösungsansätze	15
4 Grundlagen	17
4.1 Technische Grundlagen	17
4.1.1 Android	17
4.1.2 Relationale Datenbanken !!! muss das da rein?	19
4.1.3 Mobile Sensorik <– Ich benutze nur GPS?	21
4.1.4 Geolokation mittels Global Positioning System (GPS)	22
4.2 Berechnung der Geschwindigkeitsempfehlung	23
5 Szenarien im Ampelbereich	24
6 Anforderungsdefinition	26
6.1 Funktionalität	26
6.2 Die graphische Oberfläche	27
7 Konzeption	28
7.1 Anwendungsaufbau	28
7.2 Das Design	28

7.3	Datengrundlage	29
7.3.1	Ampeldaten	29
7.3.2	Positionsdaten + Geschwindigkeit	29
7.4	Architektur	29
7.4.1	Use-Cases	29
7.4.2	Aktivitäten	30
7.4.3	Klassenstruktur UML	30
7.5	Theorie ?	30
7.6	Entwicklungsumgebung ?	30
8	Der Prototyp	31
8.1	Android-Manifest	31
8.2	Verwendete Bibliotheken?	31
8.2.1	googlebla	31
8.3	MainActivity	31
8.4	Umsetzung Szenarien	31
8.5	Lokalisierung	31
9	Evaluierung	32
9.1	Systemtest	32
	Akronyme	33
	Glossar	34
	Abbildungsverzeichnis	35
	Literaturverzeichnis	36

1 EINFÜHRUNG

1.1 MOTIVATION

Im Berliner Verkehrswesen ist ein deutlicher Trend zu bemerken. Das Fahrrad wird zum ökologischen und gesundheitlichen, aktiven Lebensstil und wird dem hohen Verkehrsaufkommen der Automobile, insbesondere in der Stadtregion, entgegenwirken. „Fahrradfahren boomt in Berlin stärker als bislang bekannt“ [J.A14]

Neue Fahrradwege und Vergrößerung des Fahrradstraßennetzes sind regionale Baumaßnahmen, die dabei aktuell diesen Fahrradtrend unterstützen. Grund der neuen Fahrradeuphorie ist nicht zuletzt die erfolgreiche Etablierung der E-Bikes¹. E-Bikes erfreuen sich großer Beliebtheit und ermöglichen auch längere Touren ohne große Anstrengung.

Die Digitalisierung der Autoinnenräume mit Navigation und Bordelektronik sowie die Verbindungen zu Smartphones stellen aktuell keine Besonderheit mehr dar. Wird das Fahrrad nun als „vollwertiges“ Mitglied im Straßenverkehr angesehen, kann zusätzliche Elektronik wie Navigation die FahrradfahrerInnen unterstützen.

Sicherheit und eine rechtzeitige Ankunft am Ziel sind die Hauptaspekte der VerkehrsteilnehmerInnen. Das Halten an der Ampel kann dabei schnell zu Verzögerungen führen. Doch wer die Restzeit im Voraus kennt, kann sich darauf einstellen und so die verlorenen Zeitabschnitte reduzieren.

1.2 ZIELSTELLUNG

Der Fahrtfluss der RadfahrerInnen soll nicht unnötig unterbrochen werden. Rote Ampeln zwingen einen zum Anhalten – das Anfahren kostet Kraft und ist deshalb unbeliebt. So kommt es, dass viele RadfahrerInnen die Straße bei rot überqueren und die Verkehrssicherheit aller gefährden, wo doch das Radfahren an sich zur Gesundheit beiträgt und auch gut für die Umwelt ist. Angesichts des Nutzenpotentials eines Ampelinformationssystems lässt sich die Zielstellung klar und deutlich formulieren. Durch das reibungslosen Passieren der Ampeln wird der Verkehr sicherer und das Radfahren attraktiver.

Um die Ampeldaten zu erfassen, gibt es verschiedene Möglichkeiten. Eine 100prozentige Deckung erreicht man nicht einmal durch manuelle Ablesung jeder Ampel, da circa 20 Prozent der Lichtsignalanlagen in Berlin manuell gesteuert werden. Die Berliner Verkehrslenkungszentrale stellt für diese Arbeit verkehrstechnische Unterlagen wie einen Lageplan, Signalzeitenpläne und Daten der verkehrsabhängigen Steuerung von Lichtsignalanlagen von circa zehn Anlagen als Basis für den zu

¹ Elektrofahrrad. Ein Fahrrad mit elektrischem Hilfsmotor

entwickelnden Prototypen zur Verfügung. Für die Auswertung der Daten werden die potentiellen Wartezeiten an der nächsten Ampel vorzeitig errechnet und den FahrerInnen mitgeteilt. Resultierend kann der Nutzer die Geschwindigkeit anpassen und die verbleibende Wegstrecke zur Ampel nutzen, um bei Grün ohne anzuhalten die Kreuzung zu überqueren. Für die Datenerhebung werden zugleich die mobilen Systeme der RadfahrerInnen genutzt. Wenn man das mit Ampeln auf gegebener Teststrecke umsetzt, kann zunächst der Prototyp des Ampelhinweissystem entwickelt werden.

Das Ziel der Arbeit ist ein Konzept und dessen prototypische Anwendung eines Ampelhinweissystem, welches einem auf Basis der zu erstellenden Ampeldatenbank Informationen über die Ampelschaltung zukommen lässt und ihn so interaktiv durch das Verkehrsnetz führt.

1.3 AUFBAU DER ARBEIT

Zunächst wird analysiert welche Studien, Projekte oder Anwendungen es zu diesem Thema bereits gibt. Dann wird erläutert, ob sich eine mobile Anwendung oder eher eine Arduinoinstallation anbietet. Basierend darauf werden im vierten Kapitel die technischen Grundlagen erklärt. Hier werden also Definitionen und Entwicklungswerkzeuge beschrieben und ein Überblick über mögliche Einsatzgebiete gegeben. Für das Verständnis der Umsetzung ist die Klärung der theoretischen Berechnungsgrundlagen erforderlich.

Im Anforderungsanalysekapitel werden die Anforderungen an Funktionalität und Usability Kriterien für die Anwendung ermittelt und strukturiert.

Es werden Personas, fiktive Benutzer, eingeführt und schließlich eine Zusammenfassung der herausgearbeiteten Anforderungen gestellt.

Das sechste Kapitel bildet mit der Konzipierung den Kern dieser Arbeit. *Architektur, Design, (Theorie — funktioniert nicht wie in Konzeption beschrieben, weil...) es wird auf.. einnengangen.... mobile Anwendung. Applikation (App). die NutzerInnen auf Ampeln hinweist und die Dauer der Phase. Zusammenfassend wird das Konzept am Ende von allen Personas noch einmal kritisch betrachtet und evaluiert.*

Kapitel sieben beschreibt die Umsetzung des exemplarischen Prototyps. Dieser wird dann in Architektur, Funktionalität und Design erläutert und schließlich in mehreren Testreihen evaluiert. Anhand einiger Systemtests wird eine Optimierung der Anwendung herausgearbeitet und gegebenenfalls umgesetzt.

Den Abschluss dieser Arbeit bildet eine Zusammenfassung der Ereignisse dieser Arbeit und einen Ausblick auf zukünftige Entwicklung hinsichtlich des Themas.

2 BESTEHENDE KONZEPTE

Die Verkehrsstrategie des Senats sieht vor, dass das Radfahren bis zum Jahr 2025 20 Prozent des Gesamtverkehrs ausmachen soll. (Vgl. [J.A14]) "Wir brauchen eine intelligente Konstruktion, die alle Verkehrsarten verbindet", sagte Berlins derzeitiger Bürgermeister Michael Müller (SPD). Sowohl statisch an Radwegen, als auch für den Einsatz in Kraftfahrzeugen gibt es bereits Projekte zu Ampelassistenten in Bordcomputern, Navigationssystemen, oder aber auch als App die rote Ampeln erkennen und die optimale Fahrgeschwindigkeit für die Grüne Welle ermitteln. Auch Erweiterungen für's Rad direkt werden vielfältiger — vom einfachen Navigationssystem bis hin zu intelligenten Aufsätzen, die an das Smartphone gekoppelt sind.

2.1 AMPEL INFORMATIONSSYSTEME

Unter dem Prinzip "Grüne Welle" wird die Abstimmung der Ampelschaltzustände, sodass ein Fahrzeug in einer bestimmten Geschwindigkeit mehrere Ampeln passieren kann ohne anzuhalten, verstanden. Der folgende Abschnitt soll die existierenden Lösungen und Ansätze für Ampelinformati-onssysteme darstellen.

2.1.1 Grüne Welle auf Radwegen

In Kopenhagen unterstützen grüne Licht-emittierende Dioden (LEDs) auf Radwegen die RadfahreInnen indem sie wenn diese mit einem Tempo von 20 km/h fahren, sie begleiten und so signalisieren, dass sie sich auf der Grünen Welle befinden.

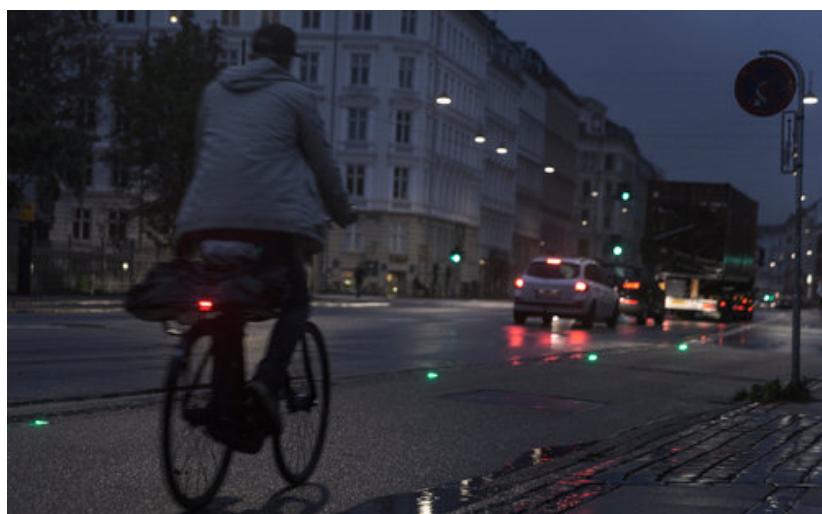


Abbildung 2.1 Kopenhagen: LEDs signalisieren die Grüne Welle bei 20 km/h Quelle: [Car14]

Zusätzlich erkennen Sensoren im Radweg Fahrradgruppen und veranlassen dann die Ampel zu einer längeren Grünphase. In einem anderen Stadtteil sind Leuchttafeln, die die verbleibende Zeit der Ampelphase anzeigen, am Radwegrand installiert [Sch14].

Kopenhagen als Vorbild hat Berlin mit vier Ampeln in Schöneberg eine Grüne Welle für RadfahrerInnen umgesetzt und plant bereits die zweite [Neu14]. Auch hier möchte man die Benutzung des Rades attraktiver machen und den Fahrradverkehr beschleunigen.

2.1.2 Individuelle Ampelinformationssysteme im Auto

In den letzten Jahrzehnten gab es immer wieder Intentionen, eine Anzeige von Geschwindigkeitsempfehlungen im Fahrzeug umzusetzen. Der folgende Abschnitt stellt existierende Lösungen und Ansätze für Ampelinformationssysteme im Auto vor.

Projekt Wolfsburger Welle

Die VW-Forschung initiierte in den 80er Jahren mit dem Projekt "Wolfsburger Welle" die ersten Untersuchungen zur "Grünen Welle" Informationen im Fahrzeug; mit der Idee, beim Annähern an eine Ampel die optimale Geschwindigkeit im Fahrzeug zu geben. [Zim84] "Dazu sendet die Ampelanlage ihren aktuellen Phasenzustand und eine Prognose für den nächsten Zustandwechsel an alle Fahrzeuge, die sich annähern. Der Fahrzeugcomputer setzt dann die aktuelle Fahrzeuggeschwindigkeit mit dem Abstand zur Ampel und der aktuellen Ampelphase in Bezug. Daraus wird errechnet, ob das Fahrzeug im Moment mit der grünen Welle 'mitschwimmt' oder ob die Geschwindigkeit außerhalb des optimalen Bereichs liegt"([Tho09]).

Projekt Travolution

Im Sommer 2008 wurde das Projekt TRAVOLUTION (TRAffic & eVOLUTION), von den Projektpartnern¹ abgeschlossen. Es besteht aus den Teilprojekten VERKEHRSADAPTIVE NETZSTEUERUNG MIT GENETISCHEN ALGORITHMEN und DER INFORMIERTE FAHRER. Im Netzsteuerungsprojekt wurden 46 Lichtsignalanlagen in Ingolstadt mit der Netzsteuerungssoftware BALANCE ausgestattet, wodurch sie intelligent auf den Verkehr reagieren und die Schaltung an den Verkehr anpassen. Ziel des zweiten Teilprojektes war es, die Autofahrer über die Ampelphasen zu informieren. Die Car-to-Infrastructure (C2I) auf Basis von Wireless Local Area Network (WLAN) umsetzend, senden mit Kommunikationsmodulen ausgestattete Ampeln die Grünphasen an den Bordcomputer der Autos, welcher dann die Geschwindigkeit für ein reibungsloses Passieren errechnet [BBK⁺⁰⁹] und wie in Abbildung 2.2 zu sehen ist, anzeigt. Im Rahmen des Projektes wurden zwei Anwendungsfälle umgesetzt. Die Restrotanzeige – die die Dauer der verbleibenden Rotphase angibt, und die "Dynamische Grüne Welle" – die Anzeige der Progressionsgeschwindigkeit. Die Vorhersage der Schaltbilder ist aufgrund der verkehrsabhängigen Logik bei nicht festzeitgesteuerten Lichtsignalanlage (LSA) nur mit einer gewissen Wahrscheinlichkeit möglich. So werden sekündlich die Grünwahrscheinlichkeiten vom LSA-Kommunikationsmodul aktualisiert und an das Fahrzeugmodul gesendet, welches diese in Relation zu eigener Position, Geschwindigkeit und Richtung setzt und die entsprechende Empfehlung ausgibt.

¹ [tra]



Abbildung 2.2 Der Bordcomputer zeigt die optimale Geschwindigkeit an. Quelle: [tra]

Fundierend auf TRAVOLUTION sind Folgeprojekte wie zum Beispiel das ebenfalls von Audi ins Leben gerufene “Ampelinfo online“ entstanden. Über Mobilfunk ist in der Car-to-X (C2X)-Anwendung das Auto mit dem zentralen Verkehrsrechner, welcher die Ampelanlagen steuert, vernetzt und visualisiert die entsprechenden Informationen im Bordcomputer. [Amp14]

Projekt Kolibri

In Bayern wurde im April 2011 das Pilot-Projekt KOLIBRI² mit den Teststrecken der B13 bei München mit sieben und der St2145 in der Nähe von Regensburg mit acht ampelgeregelten Kreuzungen gestartet. Gemeinsam untersuchten die Projektpartner³ die Funktionen und Auswirkungen eines Ampelassistenten außerhalb von Ortschaften ([kol]). “Per Mobilfunk übertragen die Ampeln ihre Daten an die Zentrale der TRANSVER GmbH. Dort wertet sie ein Computer aus und sendet die Ergebnisse an die Fahrzeuge. Ein Anzeigefeld im Bordcomputer oder eine Applikation auf dem Smartphone zeigt an, ob sich das Fahrzeug in der Grünen Welle bewegt.“ ([Bat13])



Abbildung 2.3 Anzeige der Geschwindigkeitsempfehlung. Quelle: [kol]

Die FahrerInnen wurden sowohl fahrzeugintegriert⁴ als auch via Smartphone, wie Abbildung 2.3 zeigt, über die Schaltung der nächsten Ampel informiert und erhielten Empfehlungen über die aktu-

² Kooperative Lichtsignaloptimierung – Bayrisches Pilotprojekt

³ <http://www.kolibri-projekt.de/Sites/kolibri3.html>

⁴ On-Board-Computer

elle Progressionsgeschwindigkeit. Nach zwei Jahren erfolgreicher Arbeit war das Pilotprojekt abgeschlossen

Projekt Testfeld Telematik

Ende des Jahres 2013 wurde in Wien das Projekt TESTFELD TELEMATIK – Feldversuch zur Stärkung österreichischen Know-Hows im Bereich umweltverträglicher Mobilität erfolgreich abgeschlossen. Per C2X-Kommunikation bringt das Projekt Kooperative Dienste wie Ampelinformationen direkt ins Auto.



Abbildung 2.4 Mobile Anwendung des Projekts Testfeld-Telematik Quelle: [Jan14]

Über Navigationssysteme, integrierte Systeme, Nachrüst-Plattformen oder mobile Endgeräte erreicht die FahrerInnen die Information der optimalen Geschwindigkeit sowie die Dauer der jeweiligen Ampelphase [Jan14]. Um an die Informationen zu kommen wurden unter anderem Kameras und Sensoren, beispielsweise als Induktionsschleife in die Fahrbahn eingelassen. Andere Autohersteller wie BMW, Volvo und Volkswagen kooperieren als Forschungsprojekt "Car 2 Car Communication Consortium" mit TESTFELD TELEMATIK, ebenfalls mit dem Ziel die Sicherheit an Kreuzungen zu verbessern. Im Auto installierte Sensoren kommunizieren mit Kameras und Scanner in der Ampel. Allerdings funktioniert das System nur mit dem ambitionierten Ziel, wenn alle Autohersteller zusammenarbeiten und sich auf den gleichen Standard einigen. [Elf13]

Toyota

Auch Toyota hat ein System entwickelt, welches eine spezielle Infrastruktur an Kreuzungen, die Installation von Infrarot-Sendern, die mit dem Toyota-Navigationssystem kommunizieren erfordert. An roter Ampel werden die Fahrer über die verbleibende Wartezeit informiert. Die ausgestatteten Navigationssysteme wurden bis jetzt jedoch ausschließlich in Japan getestet. [Toy11]

2.2 AMPEL INFORMATIONSSYSTEME ALS MOBILE APPLIKATION

Ampelassistenten als mobile Anwendung sind recht einfach umzusetzen. Smartphones sind bereits mit einem GPS-Empfänger ausgestattet und haben Internetzugang. Die hier vorgestellten mobilen Anwendungen existieren bereits oder befinden sich in der Testphase.

Mobile Applikation EnLighten

ENLIGHTEN erkennt rote Ampeln und visualisiert die Dauer dieser Phase. Die mobile Anwendung nutzt GPS zur Lokalisierung des Autos und verwendet die Dedicated Short Range Communication (DS-RC)-Kommunikation zu Ampelphasenprognose. An Autos und Verkehrsinfrastruktur wie Ampeln wird entsprechende Hardware installiert, die gesammelten Daten an die Verkehrszentrale gesendet, sodass dort die Ampelphasen prognostiziert werden können. Dafür werden Komponenten wie Höchstgeschwindigkeit, Ampelschaltpläne, Tageszeit und Fahrtrichtung beachtet. Mit der Verkehrszentrale verbunden, visualisiert ENLIGHTEN die Restrotdauer.

Aufgrund von hohen Installationskosten und -Aufwand ist ENLIGHTEN erst in einigen amerikanischen Städten funktionstüchtig und verfügbar⁵.

Mobile Applikation Signal Guru

Signal Guru wurde von den Wissenschaftlern des MIT⁶ und der Universität von Princeton entwickelt. Unter den vorgestellten Projekten hebt sich Signal Guru insofern ab, dass die Informationen nicht direkt von einer Vermittlung (Server oder LSA) kommt, sondern von der Anwendung selbst errechnet wird. Die App errechnet über die Smartphones vieler Nutzer - welche miteinander kommunizieren - die Wahrscheinlichkeit, wann eine Ampel grün wird und errechnen ein daraus ein Zeitmuster zur Voraussage. Wie in Abbildung 2.5 ist zu sehen ist, muss die eingebaute Kamera durch die Windschutzscheibe die Ampel registrieren. Bei Testläufen im Straßenverkehr vielen die Ergebnisse bei statisch geschalteten Ampeln deutlich besser aus als bei angepassten Ampelschaltungen [KPM11]



Abbildung 2.5 Signal Guru muss in der Lage sein die Ampel zu 'sehen'. Quelle: [KPM11]

Dieser Ansatz ist für Fahrräder jedoch nicht umsetzbar, da das Smartphone in der Halterung am Lenker die LSA nicht erfassen kann.

⁵ [EnL14]

⁶ Massachusetts Institute of Technology

2.3 FAHRRADERWEITERUNGEN

Als Schnittstelle das Smartphone nutzend gibt es ausgeklügelte Systeme mit reichlich Funktionen. Das einfache Navigationssystem für Fahrräder ist kaum noch notwendig, wo doch zum Beispiel die hier aufgeführten um einiges umfangreicher sind.

2.3.1 Displaylose Fahrradnavigation

Das HAMMERHEAD ist ein “Hammer“, oder einfach ein “T“, an den Lenker angebracht. Gespickt mit verschiedenfarbigen LEDs zeigt es den Weg zeigen, warnt vor Hindernissen und ersetzt die vorderen Scheinwerfer.

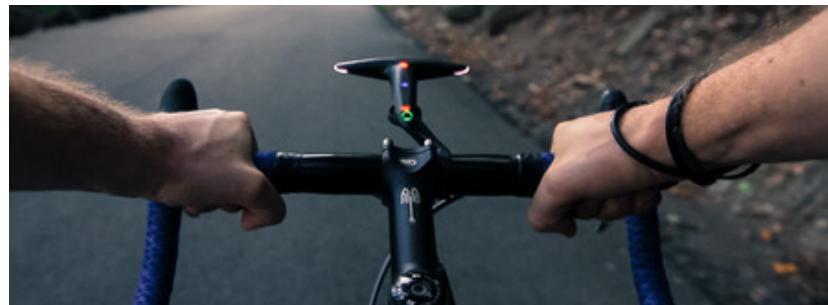


Abbildung 2.6 Hammerhead – LEDs zeigen den Weg. Quelle: [ham]

Via Bluetooth ist HAMMERHEAD an das Smartphone gekoppelt, auf dem die zugehörige Navigationsanwendung läuft mit der man Routen eingeben, teilen und speichern kann [ham].

Ein sehr ähnliches Prinzip verfolgt das CYCLENAV von der Firma Schwinn. Untersiede findet man hier im Design und einem integrierten Lautsprecher, der Abbiegehinweise ausgibt und auf Wunsch wiederholt [cyc14].

2.3.2 Intelligente Fahrradlenker

Mehr High-Tech, aber auch umfangreichere Funktionen bietet der vom amerikanischen Startup HELIOS-BIKES entwickelte HELIOS-Lenker. Neben dem Frontlicht hat der Lenker wie in Abbildung 2.7 zu sehen, an den Enden LEDs die einen zum gewünschten Standort leiten.



Abbildung 2.7 Helios-Lenker Quelle: [hel14]

Sie passen ihre Farbe der Geschwindigkeit an und haben auf Wunsch auch eine Blinkfunktion. Verbindet man den Lenker mit einem Smartphone, lässt sich die Farbe der LEDs individualisieren. Die Verbindung zum Handy hat weitere Vorteile: dank des eingebauten GPS-Trackers und eingesteckter SIM-Karte lässt sich das Fahrrad per SMS über den derzeitigen Standort abfragen [hel14], was im Falle eines Diebstahl sehr hilfreich sein kann.

VANHAWKS VALOUR heißt das Rad, das ab April 2015 lieferbar ist. Wie im HELIOS-Lenker steckt auch hier ein über das Smartphone steuerbares Navigationssystem, das die Abbiegehinweise per LED signalisiert, im Lenker. Auf den gefahrenen Routen merkt sich das Rad durch einen Erschütterungssensor erfassste Hindernisse wie Unebenheiten in der Fahrbahn und ermittelt beim nächsten Mal darauf rücksicht nehmend eine andere Route. Es ist darüber hinaus in der Lage mit anderen VANHAWKS VALOUR-Rädern zu kommunizieren und dessen Routenbegebenheiten ebenfalls zu berücksichtigen. Mittels Radarsensoren registriert das Fahrrad Autos im toten Winkel und benachrichtigt die FahrerInnen durch ein vibrierenden Lenker [van].

2.3.3 Das Samsung Smart Bike

Auf der Mailänder Designwoche hat Samsung ein Smartbike vorgestellt, das mit verschiedenen intelligenten Komponenten wie Bluetooth, einer Kamera und Laserprojektoren ausgestattet. Der Rahmen ist aus Aluminium und leicht geschwungen, was Vibrationen abfangen soll. Wie Abbildung 2.8 zeigt, zeichnen vier Laserprojektoren den eigenen, begleitenden Fahrradweg auf die Straße und sollen so die Sicherheit erhöhen, indem sie den Sicherheitsabstand markieren und aus dem toten Winkel sichtbar sind.

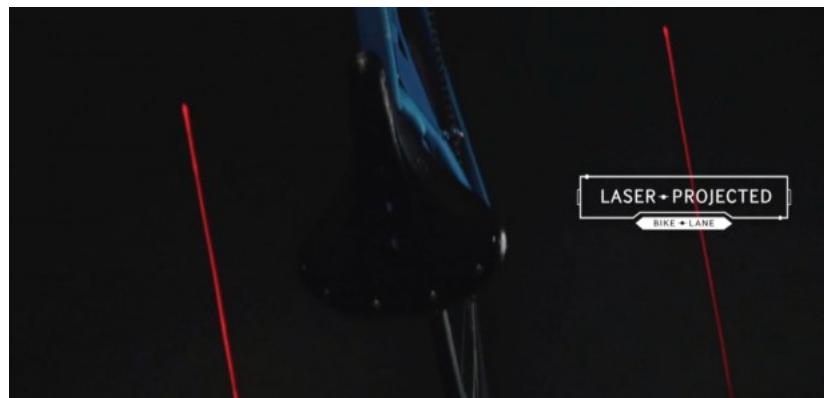


Abbildung 2.8 Samsung Smart Bike Quelle: [sma14]

Natürlich ist auch dieses Fahrrad mit dem Smartphone verbunden, das sich dank eines Magneten einfach am Lenker anbringen lässt. Darauf kann man die Laserprojektoren ein- und ausschalten, dafür einen Timer bestimmen und über die eingebaute Kamera unter dem Sattel den Verkehr hinter sich im Auge behalten. Das Smartphone fungiert außerdem als Navigationsgerät und durch den eingebauten GPS-Empfänger lassen eigene und Routen von anderen Nutzern speichern und intelligent verarbeiten [sma14]. Wenn also viele Menschen mit einem Samsung Smartbike unterwegs sind, erkennt das Rad die Route als angenehm und navigiert dort entlang.

2.3.4 Der COBI Fahrradcomputer

Ein Kickstarterprojekt aus Frankfurt am Main entwickelt das System COBI (Connected Biking), das alle standardisierten Fahrradsysteme wie Lampen, Navigation, Tachometer etc. vereinen soll. COBI ist ein Modul mit integrierter Frontleuchte in das man das Smartphone, welches dann mit der installierten COBI-App als Fahrradcomputer dient, legt. Durch eine wasser- und stoßfeste Hülle ist es vor Umwelteinflüssen geschützt. Zu dem Lenkersystem gibt es auch Rückstrahler die beim Bremsen intensiver leuchten und eine Blinkfunktion haben.



(a) Frontlicht und Smartphonehalterung

(b) Bremslicht und Blinker

Abbildung 2.9 COBI – Das smarte Fahrradsystem. Quelle: [cob14b]

Möchte man das Smartphone trotzdem nicht am Lenker haben, bleibt die Verbindung zum Modul über Funk bestehen. Steuern lässt sich das System dann über einen Controller, den man am Lenker angebracht, mit dem Daumen bedienen kann. Ist es jedoch in der Halterung, wird das Smartphone über den E-Bike-Akku oder einen zusätzlich integrierten Akku aufgeladen. Wie bei den anderen genannten Systemen ist in der COBI-App eine Navigationsanwendung, wie auch die tracking&share Funktion inklusive. Darüber hinaus verfügt es über einen Diebstahlschutz, Fitnesstracker sowie die Möglichkeit einer Anbindung an Spotify⁷.

Das Projekt ist bereits voll finanziert und der Versand der vorbestellten Systeme beginnt vorrausichtlich im Frühjahr 2015 [cob14a].

2.4 ANALYSEERGEBNIS

Das Thema der Geschwindigkeitsempfehlung zum Erreichen der Grünen Welle ist sehr interessant und wird in den nächsten Jahren weiter untersucht werden. Auch die Nachfrage an Fahrraderweiterungen steigt und die EntwicklerInnen solcher Systeme werden kreativer, wodurch immer mehr Produkte mit erweiternden Funktionen entstehen.

Es existieren noch andere Studien und Produkte zu diesem Gebiet, doch um den Rahmen dieser Arbeit nicht zu sprengen, werden nicht alle aufgezählt. Sie haben zur Realisierung dieser Arbeit beigetragen

⁷ Digitaler Musikstreaming Dienst

indem sie deutlich zeigen, dass sich Forschung zu diesem Thema lohnt.

Gerade eine Ampelassists für das Fahrrad bewegt Menschen dazu, sich für das Radfahren zu begeistern. So wird der Verkehr flüssiger, die Teilnehmer entspannter und die Luft sauberer. AutofahreInnen sind schon lange nicht mehr allein auf der Straße und so gilt es, dieses erfolgreiche Konzept für alle VerkehrsteilnehmerInnen zu erweitern.

3 LÖSUNGSANSÄTZE

Zur Umsetzung der beschriebenen Ampelinformationsanwendung kommen zwei Möglichkeiten in die engere Wahl. In diesem Kapitel werden die Realisierung durch eine Smartphone App und die einer Arduino-Anwendung gegenübergestellt. Zu beachten sind die Komponenten wie Sensorik, Internetverbindung, Stromversorgung und Darstellung der Informationen.

GPS

Als Grundlage aller modernen Navigations- und Ortungssysteme im Bereich der Navigation ist GPS für die Fahrradpositionsbestimmung obligatorisch. In einem Smartphone ist ein GPS Empfänger inklusive, für eine Arduino-Anwendung ein entsprechendes Modul vonnöten¹.

Datenbankanbindung

Ein weiterer wichtiger Aspekt ist die Datenbankanbindung. Mindestens die Position der Ampeln und die Phasen der Schaltpläne werden in einer Datenbankbank gespeichert und dort von der Anwendung angefragt und ausgewertet. Während man für die Arduino-Anwendung eine Client-Server Architektur aufbauen und via Internet auf die Datenbank zugreifen muss, liefert Android die native Datenbank SQLite mit.

Internetverbindung

Durch mobile Breitbandverbindung oder auch wahlweise per WLAN ist beim Smartphone eine Internet-Anbindung vorhanden. Das Arduino-Board benötigt für die Internetverbindung eine Erweiterung um das Ethernet-Shield².

Stromversorgung

Die Stromversorgung ist im Smartphone durch den integrierten Akku gegeben. Die Laufzeit ist vom Typ abhängig, genügt jedoch für die alltägliche Radstrecke. Für die mobile Stromversorgung des Arduino-Boards wird eine 9-Volt Batterie benötigt, die zusätzlichen Platz beansprucht und wassergeschützt und gut erreichbar angebracht werden muss. Sicher gibt es weiterhin die Möglichkeit den Strom aus dem Nabendynamo zu gewinnen – dies bedarf jedoch zusätzlicher Arbeit. Außerdem sollte dann der Arduino in der Lage sein Strom zu speichern, sodass die Anwendung beim Halt an der Ampel nicht ausschaltet.

¹ Vgl. [Som10] S. 227

² Vgl. [Som10] S. 36

Darstellung

Ein Darstellungskonzept muss bei beiden Möglichkeiten erstellt werden. Auf dem Smartphone ist besonders auf Erkennbarkeit bei schlechten Witterungsbedingungen, das ggf. spiegelnde Display berücksichtigend zu achten. Dafür sind aufgrund des vorhandenen Displays in gewisser Größe wesentlich mehr Informationen darstellbar. Als Arduino-Anwendung sind lediglich ein paar helle LEDs am bzw. im Lenker erforderlich. Diese müssen doch eindeutig und intuitiv lesbar sein, um den vollen Informationsumfang zu gewährleisten.

Ergebnis

Auch wenn das Ampelinformationssystem als Arduino-Anwendung minimalistischer ist sprechen die anderen Punkte dagegen. Beginnend bei der einfachen Stromversorgung bis hin zur vorhandenen Sensorik und internen Datenbank steht das Smartphone weiter vorn und überzeugt durch seine Einfachheit in der Anwendung. Auch die vielen Erweiterungen die es bereits für das Fahrrad in Form einer mobilen Anwendung gibt, zeigen dass Lösungen für oben genannte Probleme wie die Nutzung bei schlechten Witterungsbedingungen existieren.

4 GRUNDLAGEN

Dieses Kapitel befasst sich mit sowohl den mathematischen als auch den technischen Grundlagen der zu behandelnden Thematik, welche für das weitere Verständnis der Arbeit beitragen.

4.1 TECHNISCHE GRUNDLAGEN

Im Zuge dieser Arbeit wird eine Smartphone Anwendung erstellt, deren Grundlage für die Implementierung die Software-Plattform Android und der im Smartphone integrierte GPS-Sensor ist. Die eigene Position und Geschwindigkeit wird mittels GPS ermittelt, um in Verbindung mit der festen Position der nächsten LSA die optimale Geschwindigkeit für das Erreichen der “Grünen Welle“ zu errechnen. Im folgenden Abschnitt werden Funktionsweise und Besonderheiten der verwendeten Technologien beschrieben.

4.1.1 Android

Die umfassende Open-Source-Plattform Android stellt eine vollständige Ausstattung für Mobilgeräte dar. Android-Anwendungen werden mit der Programmiersprache Java und der Auszeichnungssprache Extensible Markup Language (XML) entwickelt. Mit dem Android Software Development Kit (SDK)¹ werden die Werkzeuge und Application Programming Interface (API) zur Verfügung gestellt, die erforderlich sind Mobilanwendungen auf der Android-Plattform erzeugen zu können.

Zu den wichtigsten SDK Werkzeugen gehören der Android SDK-Manager, der AVD-Manager, der Emulator und der Dalvik Debug Monitor Server(Ddms). Der SDK-Manager verwaltet die SDK-Pakete, sowie die installierten Pakete und System-Images. Der AVD-Manager bietet eine grafische Oberfläche in der Android Virtuell Devices verwaltet, und im Emulator ausgeführt werden können. Mithilfe des Ddms können Android Anwendungen auf Fehler untersucht werden. [anda]

Ursprünglich gab es die Virtuelle Maschine Dalvik, die unter Berücksichtigung von Rechenleistung und der Lebensdauer von Batterien speziell für Android entworfen wurde. [QUELLE] Diese wurde mit Android 5.0 von der Android Runtime (ART) als effizientere Laufzeit komplett ersetzt. [andb]

Mit dem Android Native Development Kit (NDK)² existiert auch ein Tool, mit dem Teile einer Anwendung in systemeigenen Programmiersprachen wie C oder C++ implementiert werden können. Programmcode der in solchen Sprachen geschrieben ist, eignet sich zum Beispiel bei CPU-intensiven Operationen wie Signalverarbeitungen oder Physik-Simulationen besonders gut. Hier ist allerdings

¹ Das Android SDK steht unter <https://developer.android.com/sdk/index.html> zum Download bereit

² Das Android NDK steht unter <https://developer.android.com/tools/sdk/ndk/index.html> zum Download bereit

sicherzustellen, ob die erforderlichen Bibliotheken in dem SDK auch verfügbar sind. [andd] Einen Überblick über die komplexe Android-Systemarchitektur, welche nachfolgend (nach [KC13] S. 3ff) kurz beschrieben wird, zeigt die folgende Abbildung.

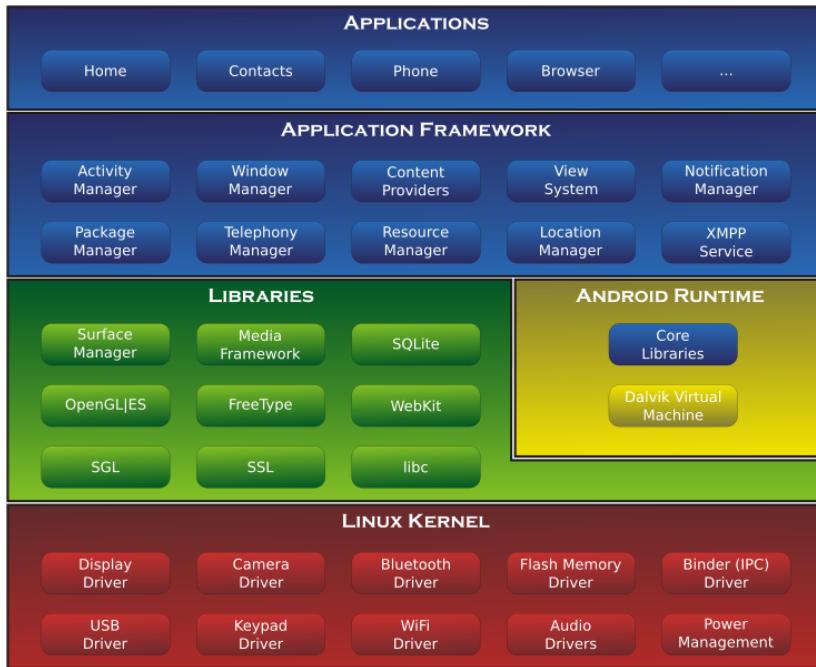


Abbildung 4.1 Die Android-Systemarchitektur Quelle: <http://en.wikipedia.org/wiki/File:Android-System-Architecture.svg>

Linux Kernel: Android basiert auf dem Linux 3.1-Kernel. Dieser eine bewährte Betriebssystemgrundlage indem er die erforderlichen Hardware-Treiber zur Verfügung stellt.

Bibliotheken: Systemeigenen Bibliotheken sind C/C++ Bibliotheken und vorinstalliert. Dazu gehören alle Bibliotheken im grünen Bereich von Abbildung 4.1:

- SURFACE MANAGER Der für die Displayverwaltung verantwortliche Oberflächen-Manager
- OPENGL/ES Eine 2D und 3D Grafikbibliothek
- SGL Eine 2D Grafikbibliothek
- MEDIA-FRAMEWORK eine Medien-Bibliothek zur Wiedergabe von Audio- und Video-Daten
- FREETYPE eine Bibliothek zur Darstellung von Computerschriften als Rastergrafik
- SSL Das Secure-Socket-Layer für die Internet-Sicherheit
- SQLITE Ist eine ausgereifte Datenbank die den internen Gerätespeicher nutzt
- WEBKIT WebKit ist die Standard-Browser-Engine und erlaubt das schnelle Rendern und Anzeigen von HTML Seiten
- LIBC Eine C-Bibliothek

Android Runtime: Die Android Laufzeitumgebung nutzt die Java-Core-Bibliotheken und bis zur Version 5.0 auch die Dalvik Virtual Maschine (VM), weswegen diese noch auf der Grafik abgebildet ist. *Die Dalvik VM ist Googles Implementation von Java, zur Anwendung auf mobilen Geräten*

optimiert. Jede gestartete Android-Anwendung läuft in einem eigenen Prozess und bekommt darüber hinaus seine eigene Dalvik VM. Da die Anwendungen über keinen gemeinsamen Speicher verfügen erhöht das die Sicherheit und Verfügbarkeit und ist somit optimaler, obwohl mehr Ressourcen benötigt werden. Denn ein sterbender Prozess nimmt so nur seine "eigene" Anwendung mit. Die Anwendungen werden zunächst von einem normalen Java-Compiler in Java-Bytecode übersetzt und dann von dem Dex-Compiler in den Dalvik-Bytecode, welcher schließlich von der Dalvik VM ausgeführt wird. [andc]

Application Framework: Androids Application-Framework ist eine Umgebung die unterschiedliche Dienste zur Verfügung stellt. Sie bietet EntwicklerInnen Zugriff auf die im Kern verwendeten APIs sowie auf die Java-Bibliotheken die für Android erstellt wurden.

Applications: Auf der obersten Ebene in Abbildung 4.1 befinden sich die Anwendungen die den täglichen Telefon-Bedarfs wie Adressbuch, Messenger, E-Mail, Internet-Browser etc. decken. Zusätzlich unterstützt Android verschiedene Anwendungen von Drittanbietern. Diese sind hauptsächlich in Java geschrieben und werden am häufigsten über den Google Play Store verteilt.

4.1.2 Relationale Datenbanken !!! muss das da rein?

Eine relationale Datenbank ist ein Konstrukt zum Ablegen von Daten. Die Organisation der Daten erfolgt in Form von Tabellen, also Datenstrukturen mit Zeilen und Spalten. Die Daten in verschiedenen Tabellen können sich gegenseitig referenzieren (über IDs, die man Schlüssel nennt) und dadurch Beziehungen (Relationen) aufweisen. Jede Zeile einer Tabelle bildet einen sogenannten Datensatz und setzt sich aus den Werten mehrerer Felder (Spalten) zusammen. Typisch ist, dass es in jeder Tabelle eine Spalte gibt, deren Werte innerhalb der Tabelle eindeutig sind (d. h., es gibt in der Tabelle keine zweiten Datensätze, die in der betreffenden Spalte denselben Wert stehen haben). Diese Spalten werden als Primärschlüssel bezeichnet. Wird eine solche Primärschlüssel-Spalte in einer anderen Tabelle verwendet, spricht man von einem sogenannten Fremdschlüssel. Über Fremdschlüssel kann ein Datensatz eine Verbindung zu einem Datensatz einer anderen Tabelle herstellen. In unserem Beispiel ergänzt z. B. der Fremdschlüssel die Bestellinformationen um die Information, welcher Kunde die Bestellung aufgegeben hat. Wie eingangs erwähnt, bietet Android das Datenbanksystem SQLite an, mit dessen Hilfe man von einer App aus Daten in Form einer Datenbank, d. h. einer definierten Menge von Tabellen, ablegen und auslesen kann.

Für die Kommunikation mit relationalen Datenbanken wie SQLite gibt es eine spezielle Kommandosprache namens Structured Query Language (SQL). Im Falle von SQLite können via SQL folgende Operationen durchgeführt werden:

- Definition der Tabellenstruktur (Tabellename, Spaltenanzahl, Datentyp pro Spalte)
- Anlegen und Löschen einer Tabelle
- Einfügen, Ändern, Löschen von Datensätzen
- Suchen von Daten aus einer oder mehreren Tabellen

Die meisten dieser Operationen lassen sich aber auch ohne Rückgriff auf SQL ausführen.

SQLITE

SQLite ist eine weitgehend in sich abgeschlossene Datenbank-Engine die auch bei Android sehr beliebt ist. Es benötigt nur minimale Unterstützung von externen Bibliotheken oder vom Betriebssystem. So eignet es sich gut für den Einsatz in Systemen, denen die Infrastruktur eines Desktop-Computers fehlt. Der Code ist Open Source und frei für den privaten oder gewerblichen Einsatz.

Im Gegensatz zu den meisten anderen SQL-Datenbanken ist keine Client-Server-Architektur vorhanden. Um in die Datenbank zu schreiben oder daraus zu lesen greift der Prozess auf die interne Festplatte, und nicht auf einen Server zu. Es ist also keine Netzwererverbindung vornötigen.

Der Hauptvorteil, keinen separaten Serverprozess zu installieren, einzurichten, zu konfigurieren und zu verwalten, macht SQLite zu einem konfigurationsfreien Datenbanksystem. Es besteht keine administrative Notwendigkeit um eine neue Datenbankinstanz zu erstellen oder Zugriffsrechte zuzuweisen.

“SQLite just works.“ [SQLb]

Als Transaktionsdatenbank implementiert SQLite serialisierbare Transaktionen die auch bei einem Programm- oder Betriebssystemabsturz atomar, konsistent, isoliert und dauerhaft sind. Wird also die Transaktion durch z.B. einen Programmabsturz unterbrochen, werden alle Änderungen entweder gar nicht oder komplett angezeigt.

SQLite ist außerdem eine Ein-Datei-Datenbank. Das bedeutet die vollständige SQL-Datenbank mit mehreren Tabellen, Indizes und Ansichten wird in nur einer Datei gespeichert. So kann jedes Programm, das in der Lage ist auf die Festplatte zuzugreifen, die SQLite-Datenbank verwenden. [SQLa] Aufgrund eben dieser Eigenschaften eignet sich SQLite ausgezeichnet für die Entwicklung von Android-Apps.

SQLITE UNTER ANDROID

Android stellt mit dem Paket `android.database.sqlite` eine API zur Verfügung, die den effektiven Einsatz von SQLite ermöglicht. Dort findet sich die Hilfsklasse `SQLiteOpenHelper`, welche eine Instanz von `SQLiteDataBase` liefert. Der Konstruktor der Hilfsklasse erwartet ein Context-Objekt und den Namen der Datenbank. Beim Aufruf der geerbten Methoden `getReadableDatabase()` zum Datenauslesen oder `getWritableDatabase()` zum Verändern oder auslesen der Daten wird geprüft, ob eine Datenbank mit diesem Namen bereits existiert. Ist das der Fall erhält man eine Referenz auf Datenbankobjekt, wenn nicht wird eines erstellt.

Wird eine Datenbank neu angelegt, geschieht das durch den Aufruf der `onCreate()` Methode. Hier wird ein SQL-String erstellt, der den Tabellennamen und die enthaltenden Spalten angibt. Dieser String wird via `execSQL()` dem `SQLite-Database`-Objekt gegeben, in Folge dessen eine Tabellenstruktur angelegt wird.

```
1 public void onCreate(SQLiteDatabase db) {  
  
3     String sql = "CREATE TABLE schiffe" +  
4             "(id INTEGER PRIMARY KEY AUTOINCREMENT , " +  
5             "bezeichnung VARCHAR(20) NOT NULL , " +  
6             "captain VARCHAR(20) NOT NULL , " +
```

```
7     "geburtstatum DATE)";  
9     db.execSQL(db);  
10 }
```

Listing 4.1 Anlegen einer Tabellenstruktur

Im obigen Codebeispiel soll eine Tabelle namens *schiffe* mit den vier Spalten *id*, *bezeichnung*, *captain* und *geburtstatum* mit den entsprechenden Datentypen angelegt werden. Die Spalte *id* fungiert hier als Primärschlüssel, kann also zur eindeutigen Referenzierung der Datensätze verwendet werden. Mit der Angabe von `AUTOINCREMENT` setzt SQLite automatisch die nächste verfügbare ID als Wert ein. Die Datensätze ohne den Zusatz `NOT NULL` sind nicht obligatorisch.

Jede SQLite-Datenbank die in einer App erzeugt wird, wird auch dort abgelegt. So ist sie auch nur für die Anwendung zugreifbar, die sie angelegt hat. Zu erwähnen ist noch, dass der Umfang der Datenbank die interne Speicherkapazität nicht überschreiten kann. Hat man die Datenbank erfolgreich erstellt, lassen sich die vier **CRUD**³ Operationen darauf anwenden. Hierfür wird die entsprechende Methode mit den gewünschten Kriterien auf die Instanz der Datenbank angewandt.

4.1.3 Mobile Sensorik <– Ich benutze nur GPS?

Ein Sensor⁴ ist ein Bauelement, das physikalische Eigenschaften wie Helligkeit, Temperatur oder Beschleunigung sowohl quantitativ als auch qualitativ erfassen kann. Die GPS-Sensoren in Smartphones sind kostengünstiger, kleiner und haben einen geringeren Stromverbrauch. Dafür allerdings eine geringere Messgenauigkeit. In der zu entwickelnden Anwendung kommt es auf jeden Meter an. Ob die Genauigkeit des integrierten GPS-Sensors genügt, muss also im Rahmen dieser Arbeit getestet werden.

MOBILE SENSORIK UNTER ANDROID

Die meisten Android-Mobilgeräte verfügen über integrierte Sensoren, die die Bewegung, Ausrichtung und verschiedene Umgebungsbedingungen messen. Diese Sensoren sind praktisch wenn man dreidimensionale Gerätbewegungen, Positionierungen oder Änderungen in der Umgebung des Gerätes überwachen möchte. So können zum Beispiel Spieleanwendungen den Beschleunigungssensor nutzen, um komplexe BenutzerInnengesten und Bewegungen wie Neigung, Erschütterung, Drehung oder Schwenkung erfassen.

Die Android-Plattform unterstützt Bewegungssensoren zum Messen von Beschleunigungen und Drehungen in drei Achsen, Umgebungssensoren zur Ermittlung verschiedener Umweltparameter wie Luftdruck und -feuchtigkeit, oder Beleuchtung und Temperatur, und Positionssensoren zum Messen der physikalischen Position des Gerätes. Android bietet mit dem Android Sensor Framework eine Sammlung von Klassen und Schnittstellen an mithilfe dessen man auf diese Sensoren zugreifen und deren Daten erfassen kann. [andf]

³ Create, Read, Update, Delete

⁴ aus dem Lateinischen, deutsch: “fühlen“

4.1.4 Geolokation mittels GPS

Die satellitengestützte Positionsbestimmung GPS gewährleistet die Bestimmung des exakten Standpunktes und ist so wesentlicher Bestandteil ortsgebundener Anwendungen wie zum Beispiel die in Kapitel 2 beschriebenen.

GPS wurde ursprünglich vom US-Militär entwickelt und dann Mitte der 90er Jahre der Öffentlichkeit zur Verfügung gestellt. Noch heute bleibt es mit einer Genauigkeit von bis zu vier Metern die genauste Lokalisierungstechnologie. Das Globale Positionsbestimmungssystem bestimmt mittels 31 Satelliten die die Erde umkreisen und ihre aktuelle Position mit der genauen Uhrzeit ausstrahlen. Daraus errechnet ein GPS-Empfänger die eigene Position. Daten von einem einzigen Satelliten lässt die Position des GPS-Empfängers auf deinen großen Bereich der Erde einschränken. Das Hinzufügen eines zweiten engt die Position auf den Bereich, in dem sich die zwei Bereiche überlappen ein. Mit den Daten eines dritten Satelliten bekommt man bereits eine relativ genaue Positionierung. Mit jedem weiteren Satelliten wird die Präzision erhöht. GPS-Empfänger benutzen regelmäßig vier bis sieben, oder gar mehr Satelliten. Trotzdem hat GPS einige Nachteile. Es verbraucht viel Energie, was die Akkulaufzeit des Mobiltelefons beeinträchtigt und benötigt nach dem Kaltstart eine Weile, den ersten Satelliten zu erreichen. ([FA11] S. 24ff)

GPS UNTER ANDROID

Android unterstützt mit dem `android.location` Paket den Zugriff auf die Ortungsdienste. Als zentrale Komponente des Location Frameworks stellt der `LocationManager` APIs zur Lokalisierung des Geräts bereit. Mit dem `LocationManager` ist die Anwendung in der Lage alle Location Provider⁵ des letzten bekannten Standortes abzufragen, sich für regelmäßige Updates zur Position des Gerätes anzumelden und sich wieder abzumelden wenn sich das Gerät außerhalb gegebener Parameter befindet. [ande]

Die geographische Positionsangabe besteht aus Längengrad (Langitude) und Breitengrad (Latitude). Beide werden unter anderem vom `LocationManager`-Objekt als Gleitkommawert geliefert. Daneben auch Informationen wie die Höhe in Metern über der Meereshöhe, Peilung, Zeitstempel und die Geschwindigkeit.

⁵ deutsch: Standortanbieter. Ein Standortanbieter bietet regelmäßige Berichte über die geographische Lage des Gerätes

4.2 BERECHNUNG DER GESCHWINDIGITSEMPFEHLUNG

Präsentiert das System während der Anwendung eine Geschwindigkeitsempfehlung, ist diese abhängig von der Fahrgeschwindigkeit und vom Abstand zur Ampel. *Geschwindigkeit ist per Definition die erste Ableitung des Weges nach der Zeit.* Für einen Streckenabschnitt zwischen zwei Punkten (Position der Ampel und Position der Rades) wird die Zeitspanne Δt benötigt. Diese errechnet man mit $t_2 - t_1$, woraus sich dann die Durchschnittsgeschwindigkeit im untersuchten Streckenabschnitt ergibt. Angenommen die Progressionsgeschwindigkeit v wird zum Zeitpunkt t_1 ermittelt, die LSA schaltet zum Zeitpunkt t_2 auf Rot und Abstand zur Ampel beträgt s , dann gilt:

$$v = \frac{s}{t_2 - t_1}$$

Der Abstand zur Ampel wird also durch die verbleibende Zeit dividiert. Die von der Berliner Verkehrsleitzentrale zur Verfügung gestellten Ampelsignalpläne und Position der angesteuerten Ampel dienen als Grundlage dieser Berechnung und sind aus der Datenbank zu holen. Die aktuelle Position des Fahrrads wird vom GPS Sensor des Smartphones ermittelt und daraus der Abstand zur Ampel errechnet. Die Abbildung 4.2 soll die Berechnungsgrundlagen veranschaulichen:

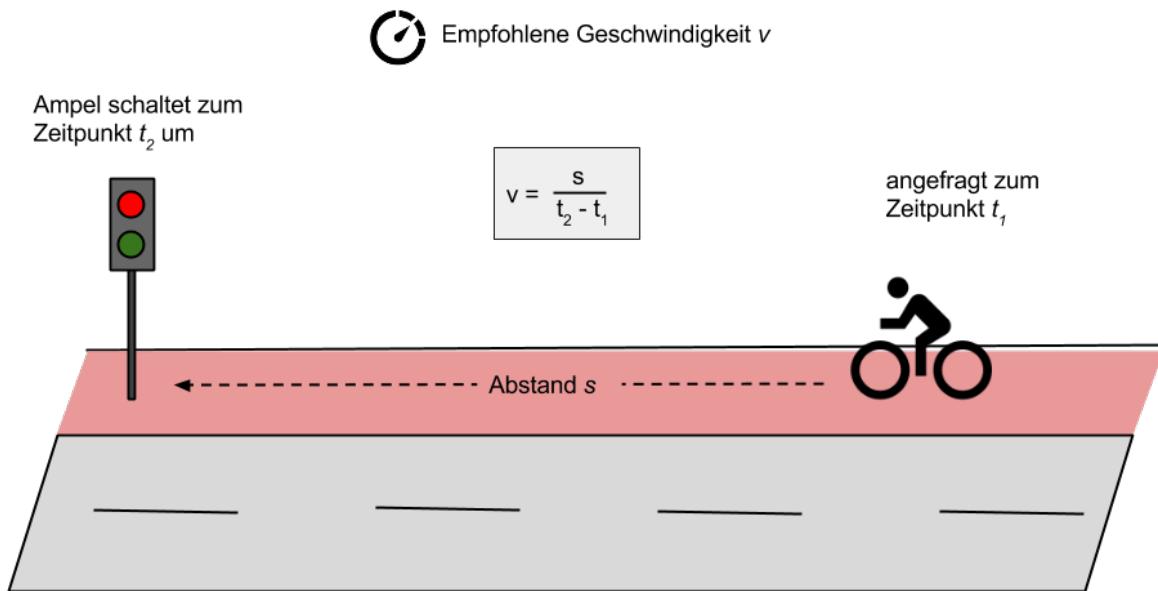


Abbildung 4.2 Veranschaulichung der Berechnung

Um die entsprechende LSA während der Grünphase zu passieren, muss letztendlich die empfohlene Geschwindigkeit v eingehalten werden.

5 Szenarien im Ampelbereich

Alle in Kapitel 2 angeführten Studien zu Ampelinformationssystemen und Konzepte zu Fahrraderweiterungen haben die Gemeinsamkeit des selbstkontrollierten Fahrverhaltens der FahrerInnen. Ausgesprochen werden lediglich Empfehlungen, die möglichst intuitiv und schnell vermittelt werden. Grundlegend sollte die Anwendung in der Lage sein, die passende Empfehlung oder Handlungsauforderung anzuzeigen, die sich aus folgenden Szenarien ergeben.

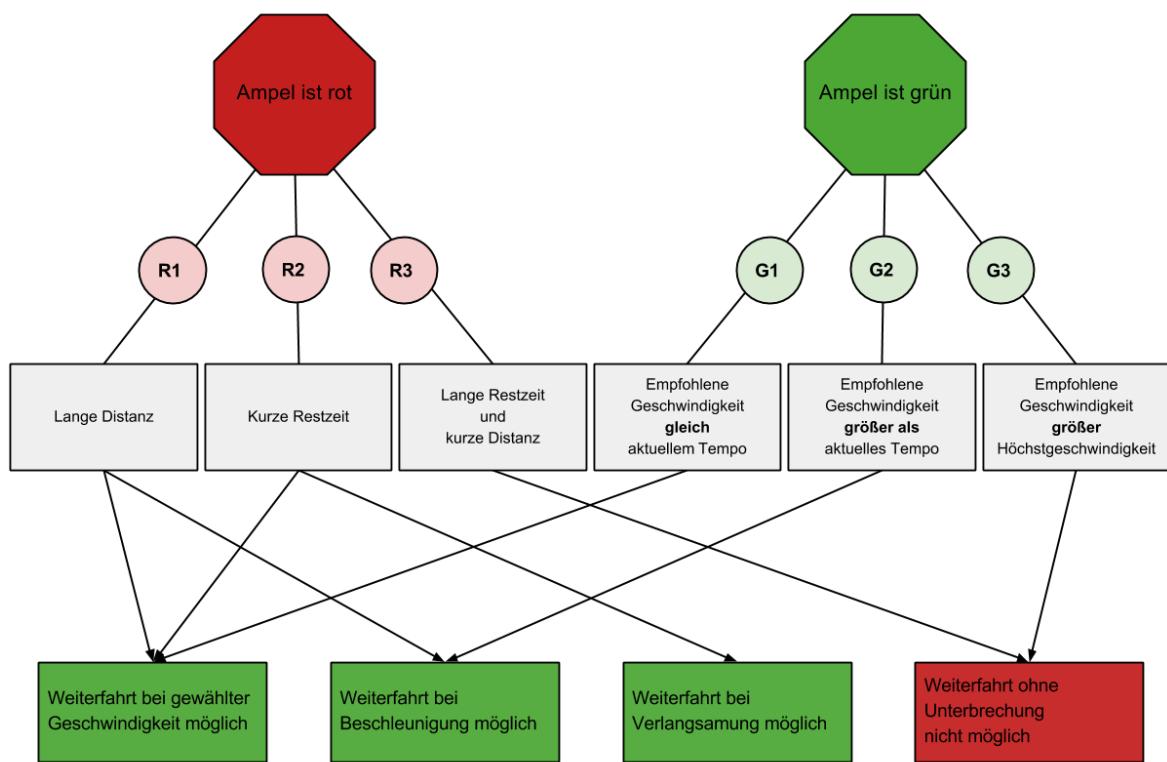


Abbildung 5.1 Szenarien im Ampelbereich

Szenario R1:

Zeigt die Ampel im Moment und noch eine ganze Weile auf Rot ist je nach Distanz zwischen Ampel und Fahrrad ist ein reibungsloses Passieren der Ampel durch Beschleunigung oder ohne Änderung der Geschwindigkeit zu erreichen.

Szenario R2:

Die Ampel zeigt im Moment auf Rot, aber die Restzeit ist kurz. Auch hier gilt: Je nach Entfernung zwischen Fahrrad und LSA, ist ein reibungsloses Passieren der Ampel durch Verzögerung oder ohne Änderung des Tempos zu erreichen.

Szenario R3:

Zeigt die Ampel im Moment Rot, die Restrotzeit ist lang, doch die Entfernung zwischen Fahrrad und LSA hoch, so ist keine Weiterfahrt ohne Unterbrechung möglich.

Szenario G1:

Ist die empfohlene Geschwindigkeit gleich der aktuellen, ist ein reibungsloses Passieren bei beibehaltenem Tempo möglich. Es besteht kein Aktionsbedarf.

Szenario G2:

Zeigt die Ampel im Moment Grün und ist die empfohlene Geschwindigkeit höher als die aktuelle, ist ein reibungsloses Passieren durch Beschleunigung zu erreichen. Bei der Anzeige der Progressionsgeschwindigkeit ist selbstverständlich die geltende Höchstgeschwindigkeitsbegrenzung oder die eingestellte Höchstgeschwindigkeit zu beachten.

Szenario G3:

Zeigt die Ampel im Moment Grün und ist die empfohlene Geschwindigkeit höher als die aktuelle und gleichzeitig auch höher als die zugelassene, bzw. eingestellte Höchstgeschwindigkeit, ist die Distanz zur Ampel zu groß und ein reibungsloses Passieren nicht möglich.

Resultierend aus den Szenarien im Ampelbereich ergeben sich die folgenden Systemzustände.

- Zustand a: Keine Weiterfahrt ohne Unterbrechung möglich
- Zustand b: Kein Aktionsbedarf
- Zustand c: Weiterfahrt durch Verlangsamung möglich
- Zustand d: Weiterfahrt durch Beschleunigung möglich

Im weiteren Verlauf wird unter anderem beschrieben wie diese Systemzustände in den Designprozess eingebunden werden.

6 ANFORDERUNGSDEFINITION

Dieses Kapitel soll helfen eine Vorstellung für die Anforderungen an die Ampelhinweissystem -App zu bekommen. Im heutigen High-Technologie Zeitalter spielt die Benutzbarkeit und Funktionalität des Produkts eine große Rolle. Es geht nicht nur darum, Interesse zu wecken, sondern auch die NutzerInnen für das Produkt zu begeistern.

Aus den oben genannten Szenarien werden im Folgenden die Anforderungen hergeleitet, die die zu entwickelnde Smartphone-Anwendung erfüllen soll. Für die Umsetzung einer solchen Applikation ist die Positions- und Geschwindigkeitsbestimmung obligatorisch. Im Zusammenhang mit den Ampeldaten, bestehend aus Lage- und Schaltpunkt sollen die notwendigen Berechnungen durchgeführt und deren Ergebnisse als Empfehlung ausgesprochen.

6.1 FUNKTIONALITÄT

Die Bestimmung der Position, Fahrtrichtung und Geschwindigkeit des Fahrrads ist die zentrale Voraussetzung für eine zeitgerechte Empfehlungsaussprechung. Sie sollte ebenso schnell wie präzise erfolgen, sodass es keine Verzögerung der berechneten Ergebnisanzeige gibt und die Informationen eine hohe Genauigkeit betragen.

Von ebenso hoher Relevanz sind die Ampeldaten bestehend aus der genauen Position der LSA und deren Schaltpläne. Es muss ermittelt werden, welche Ampel die am nächsten Passierte und daraus die Entfernung zu eben solcher. Die Prognose findet statt, sobald die Ampel circa 300 Meter voraus liegt. Bei einer größeren Entfernung von zum Beispiel einem Kilometer kann die Gefahr der Ablenkung bestehen. Sollte die Anwendung auf der Strecke durchgängig vorschlagen schneller zu fahren, könnte die Person dies als Ansporn sehen und so nicht mehr auf den Verkehr achten. Bei dieser doch recht kurzen Entfernung genügt für den zu entwickelnden Prototyp der Abstand als Berechnungsbestandteil in Luftlinie. Bei Optimierungsbedarf bietet es sich hierfür an, Kartenmaterial einzubinden, um den genauen Abstand ermitteln zu können. Hierbei werden dann beeinflussende Elemente wie Hügel oder Kurven berücksichtigt.

Die Positionsdaten und Schaltpläne, im Uhrzeit- oder Datumsformat befinden sich in einer Datenbank. Diese sollte allein aus Performancegründen keinen Internetzugriff voraussetzen (nativ sein). Auch unter Berücksichtigung des erhöhten Akkuverbrauchs bei der Verwendung von GPS ist es von Vorteil auf die WLAN-Netzsuche oder mobile Breitbandverbindung zu verzichten. Ist das Datenvolumen nämlich aufgebraucht, ist die Verbindung zu langsam die benötigsten Daten zeitnah anzufragen und zu erhalten.

Die von Android bereitgestellte systemeigene Datenbank SQLite nutzt den internen Speicher des Geräts und ermöglicht so einen schnellen Zugriff auf diese Daten. Die Datenbank wird als Datei in der

Anwendung abgelegt, wodurch sich Abfragen zu externen Servern, die eine Verbindung mit einem Netzwerk voraussetzen erübrigen.

6.2 DIE GRAPHISCHE OBERFLÄCHE

Ebenso wie die Bestimmung der Position und Geschwindigkeit spielt auch die graphische Darstellung eine große Rolle. Gerade bei dem Gebrauch im Verkehr ist die Eindeutigkeit und schnelle Erfassbarkeit der zu übermittelnden Informationen bedeutend.

Um nicht von dem Verkehr abzulenken und diesen somit zu gefährden muss die Information, sowohl in Bedeutung als auch in Darstellung auf einen kurzen Blick erkennbar sein. Demnach sollte die Oberfläche möglichst einfach gehalten werden.

Da die Anwendung während der Fahrt nicht bedienbar sein muss – denn auch das würde vom Verkehr ablenken – besteht mehr Raum für die Informationsanzeige. Es ist davon auszugehen, dass die Anwendung ausschließlich im Freien Gebrauch findet. Dies und die dort herrschenden wechselnden Helligkeiten, zum Beispiel bei der Fahrt durch Schatten, setzen eine Verwendung von hohen Kontrasten voraus.

EMPFEHLUNGSANZEIGE??? <– FRAU GÖRLITZ FRAGEN

Im Allgemeinen sollte die Anwendung in der Lage sein die sich aus Kapitel 5 resultierenden vier Systemzustände zu visualisieren und gegebenenfalls eine entsprechende Empfehlung auszusprechen. Zur Verdeutlichung und einfacher Erfassbarkeit bietet es sich an, die entsprechenden Farben zu nehmen. Also ist für Zustand *b*, der das Erreichen der Grünen Welle ohne Veränderung der eigenen Geschwindigkeit steht Grün und für den Zustand *a*, der ausdrückt dass keine Weiterfahrt ohne Absteigen möglich ist, Rot zu verwenden. Die Zustände *c* und *d* drücken aus, ein Erreichen der Grünen Welle ist möglich, jedoch nur bei Veränderung der eigenen Fahrgeschwindigkeit. Zur inhaltlichen Abgrenzung ist hier eine andere Farbe zu wählen, sodass bereits aus dem Augenwinkel erkennbar ist “Ich muss etwas tun“.

7 KONZEPTION

Die erarbeiteten Anforderungen an ein Ampelinformationssystem für FahrradfahrerInnen werden in diesem Kapitel für die Konzeption angewendet. Beginnend mit dem Aufbau der Anwendung werden in den folgenden Abschnitten das Design, die von der Anwendung genutzten Daten, Anwendungsfälle, die Architektur und schließlich die Komponenten der Entwicklungsumgebung, welche eingesetzt werden aufgeführt.

7.1 ANWENDUNGSAUFBAU

Android.

Activities bzw 1 Activity.

Action Bar / Einstellungen?

7.2 DAS DESIGN

Aus den Anforderungen an die graphische Oberfläche in Kapitel 6 entsteht das Design. Durch die überschaubare Anzahl an Funktionalitäten kann der Aufbau einfach gehalten werden. Aus den Empfehlungsanzeigen, welche den Systemzuständen entsprechen sind vier Fenster abzuleiten. Die folgende Abbildung zeigt den Entwurf der Benutzeroberfläche.

Abbildung 7.1a zeigt die grafische Umsetzung des Zustands *a*, in dem die Ampelschaltung keine Weiterfahrt ermöglicht. Es wird ein großer roter Kreis mit einem schwarzen Kreuz verwendet. Rot ist eine Signalfarbe und steht auch bei Ampeln für “Halt“ oder “Stop“. Auch das Kreuz wird häufig als *verweigerndes* Symbol eingesetzt und ist somit intuitiv als solches erkennbar.

Abbildung 7.1b setzt die Visualisierung des Zustands *b* um, bei dem man mit der aktuellen Geschwindigkeit in der Grünen Welle mitschwimmt, um. Hier wird ein großer grüner Kreis verwendet, in der Mitte steht “ok“. Durch die nicht alleinige Benutzung der Ampelfarben Rot und Grün ist die Ansicht auch für Menschen mit einer Rot-Grün-Sehschwäche eindeutig interpretierbar.

Die Abbildungen 7.1c und 7.1d für die Zustände *c* und *d* sind in der Anordnung der Elemente identisch. Mittig im Bild ist eine eingekreiste rote Zahl die den Countdown der Restrotanzeige darstellt. Sie wird also mit jeder Sekunde aktualisiert. Ober- und unterhalb des Countdowns befinden sich jeweils zwei hellblaue Pfeile. Je nach Differenz der aktuellen zur berechneten Geschwindigkeit sind ein oder zwei Pfeile ausgefüllt. Wird also empfohlen viel schneller zu fahren, sind die oberen Pfeile aktiv, bei der Aufforderung etwas das Tempo zu drosseln der untere. Als Farbe für die Pfeile wurde Hellblau gewählt. Sie unterscheidet sich sowohl farblich als auch in der Helligkeitsstufe zu den ande-

ren Farben und steht im hohen Kontrast zu dem Hintergrund. Auf schwarzem Hintergrund wirken die Farben intensiver und sind so auch aus dem Augenwinkel leichter zu erkennen und unterscheiden.



Abbildung 7.1 Entwurf des Designs anhand der Systemzustände

Die aktuelle Geschwindigkeit wird bewusst nicht angezeigt. Auch nicht die Progressionsgeschwindigkeit. Die Differenz ist beim Fahrrad nicht so hoch wie bspw. im Auto. Bei einer Varianz von wenigen km/h genügt die Anzeigevariante “schneller“ und “noch schneller“.

Auf die Anzeigevarianten in Projekten wie zB

KOLIBRI (2.1.2) und TRAVOLUTION (2.1.2)

eingehen?

7.3 DATENGRUNDLAGE

7.3.1 Ampeldaten

Von der Verkehrsleitzentrale zur Verfügung gestellt.

Position und Signalpläne

Sind ggf. umzuwandeln, bzw Datenbankformat anzupassen + manuell zu übernehmen da Format = .pdf.

7.3.2 Positionsdaten + Geschwindigkeit

7.4 ARCHITEKTUR

7.4.1 Use-Cases

7.4.2 Aktivitäten

7.4.3 Klassenstruktur UML

MainActivity, util = Hilfsklassen -> GPSTracker, SpeedAdvisory

7.5 THEORIE ?

7.6 ENTWICKLUNGSUMGEBUNG ?

Für die Erstellung wird Folgendes verwendet:

- Android Studio
- ADT Bundle..., SDK,
- SQLite
- Diagramme und Abbildungen

8 DER PROTOTYP

Prototyp zeit, wie mittels GPS ... realisiert werden kann. Design und Funktionalitäten werden ebenfalls vorgestellt

8.1 ANDROID-MANIFEST

8.2 VERWENDETE BIBLIOTHEKEN?

8.2.1 googlebla

8.3 MAINACTIVITY

8.4 UMSETZUNG SZENARIEN

8.5 LOKALISIERUNG

9 EVALUIERUNG

9.1 SYSTEMTEST

ist das GPS schnell/genau genug fürs Radfahren? Optimierung ggf. umsetzen

ABKÜRZUNGEN

API	Application Programming Interface.
App	Applikation.
C2X	Car-to-Infrastructure oder Vehicle-to-Infrastructure.
C2X	Car-to-X oder Vehicle-to-X.
DSRC	Dedicated Short Range Communication.
GPS	Global Positioning System.
LED	Licht-emittierende Diode.
LSA	Lichtsignalanlage.
NDK	Native Development Kit.
REST	REpresentational State Transfer.
SDK	Software Development Kit.
VM	Virtual Maschine.
WLAN	Wireless Local Area Network.
XML	Extensible Markup Language.

GLOSSAR

Arduino

Die Arduino-Plattform ist eine quelloffene, aus Soft- und Hardware bestehende Physical-Computing-Plattform. Die Hardware besteht aus einem einfachen I/O-Board mit einem Mikrocontroller und analogen und digitalen Ein- und Ausgängen. Die Entwicklungsumgebung basiert auf Processing, die auch technisch weniger Versierten den Zugang zur Programmierung und zu Mikrocontrollern erleichtern soll.

C2I

direkter, drahtloser Datenaustausch zwischen Fahrzeugen jeglicher Art und infrastrukturellen Einrichtungen wie Funkbaken und Lichtsignalanlagen auf Basis von WLAN, Bluetooth oder DSRC.

C2X

direkter Informationsaustausch zwischen Fahrzeugen jeglicher Art, Verkehrsleittechnik wie z.B. Lichtsignalanlagen und Verkehrsleitzentralen.

DSRC

funkgestützte sicherheitsrelevante und private Dienste, die in der Automotive-Technik von mobilen Stationen ausgeführt werden können.

Smartphone

Mobiltelefon, das sich von einem klassischen Mobiltelefon durch einen größeren berührungs-empfindlichen Bildschirm (Touchscreen) und diverse Sensoren, wie dem GPS unterscheidet. So ist eine Interaktion mit der Umgebung und den AnwenderInnen möglich.

ABBILDUNGSVERZEICHNIS

2.1	Grüne Welle durch LEDs	6
2.2	Projekt Travolution	8
2.3	Projekt Kolibri	8
2.4	Projekt Testfeld-Telematik Ampelinformation	9
2.5	Signal Guru	10
2.6	Hammerhead	11
2.7	Helios-Lenker	11
2.8	Samsung Smart Bike	12
2.9	COBI	13
4.1	Android-Systemarchitektur	18
4.2	Berechnung Progressionsgeschwindigkeit	23
5.1	Szenarien	24
7.1	Systemzustände im Ampelbereich	29

QUELLCODEVERZEICHNIS

4.1 Anlegen einer Tabellenstruktur	20
--	----

LITERATURVERZEICHNIS

- [Amp14] Audi connect – Intelligent durch den Stadtverkehr mit Ampelinfo online. http://www.audi.de/de/brand/de/vorsprung_durch_technik/content/2014/06/ampelinfo-online.html, Juni 2014
- [anda] Android Tools Help. <https://developer.android.com/tools/help/index.html>, . – Zugriff: 15.01.2015
- [andb] Android Lollipop. <http://developer.android.com/about/versions/lollipop.html>, . – Zugriff: 16.01.2015
- [andc] Introducing ART. <https://source.android.com/devices/tech/dalvik/art.html>, . – Zugriff: 15.01.2015
- [andd] Android NDK. <https://developer.android.com/tools/sdk/ndk/index.html>, . – Zugriff: 15.01.2015
- [ande] Location and Maps. <http://developer.android.com/guide/topics/location/index.html>, . – Zugriff: 17.01.2015
- [andf] Sensors Overview. http://developer.android.com/guide/topics/sensors/sensors_overview.html, . – Zugriff: 17.01.2015
- [Bat13] BATTENBERG, Dr. A.: Pilotprojekt zur Verbesserung des Verkehrsflusses präsentiert Ergebnisse: Grüne Welle auf der Landstraße spart Zeit und Sprit. (2013), Juni. – Pressemitteilung
- [BBK⁺09] BRAUN, Robert ; BUSCH, Fritz ; KEMPER, Carsten ; HILDEBRANDT, Robert ; WEICHENMEIER, Florian ; MENIG, Cornelius ; PAULUS, Ingrid ; PRESSLEIN-LEHLE, Renate: TRAVOLUTION – Netzweite Optimierung der Lichtsignalsteuerung und LSA-Fahrzeug-Kommunikation. In: *Straßenverkehrstechnik* 10 (2009), S. 365–374
- [Car14] CARDWELL, Diane: Copenhagen Lighting the Way to Greener, More Efficient Cities. In: *The New York Times* (2014), Dezember
- [cob14a] COBI. World's Smartest Connected Biking System. <https://www.kickstarter.com/projects/cobi/cobi-worlds-smallest-connected-biking-system>, 2014. – Zugriff: 28.12.2014
- [cob14b] COBI. <http://www.cobi.bike/press.html>, 2014. – Zugriff: 28.12.2014

- [cyc14] *CycleNav Smart Bike Navigator.* <http://www.schwinnbikes.com/usa/news/cyclenav-smart-bike-navigator/>, März 2014. – Zugriff: 28.12.2014
- [Elf13] ELFLEIN, Nicole: Der perfekte Beifahrer. In: *Pictures of the Future* (2013), Frühjahr, S. 104–106
- [EnL14] *EnLighten.* <http://connectedsignals.com/index.php>, 2014. – Zugriff: 28.11.2014
- [FA11] FERRARO, Richard ; AKTIHANOGLU, Murat: *Location-Aware Applications*. Shelter Island : Manning Publications Co., 2011. – ISBN 978-1-935182-33-7
- [ham] *Hammerhead.* <http://hammerhead.io/>, . – Zugriff: 28.12.2014
- [hel14] *Helios.* <http://www.ridehelios.com/>, 2014. – Zugriff: 28.12.2014
- [J.A14] J.ANKER: Fahrradfahren boomt in Berlin stärker als bislang bekannt. In: *Berliner Morgenpost* (2014), Juni
- [Jan14] JANDRISITS, Dipl.-Ing.(FH) M.: Testfeld Telematik – Publizierbarer Endbericht. 2014. – Forschungsbericht
- [KC13] KRAJCI, Iggy ; CUMMINGS, Darren: *Android on x86: An Introduction to Optimizing for Intel Architecture*. New York : Apress Media L.L.C., 2013. – ISBN 978-1-4302-6130-8
- [kol] *Kooperative Lichtsignaloptimierung.* <http://www.kolibri-projekt.de/>, . – Zugriff: 28.12.2014
- [KPM11] KOUKOUMIDIS, Emmanouil ; PEH, Li-Shiuan ; MARTONOSI, Margaret: SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory / MIT and Princeton University. 2011. – Forschungsbericht
- [Neu14] NEUMANN, Peter: Berlin bekommt zweite grüne Welle für Radler. In: *Berliner Zeitung* (2014), September
- [Sch14] SCHOENEBECK, Gudrun von: *Kopenhagen verwöhnt Radfahrer mit Grünen Wellen.* <http://www.ingenieur.de/Themen/Fahrrad/Kopenhagen-verwoehnt-Radfahrer-Gruenen-Wellen>, August 2014. – Zugriff: 28.12.2014
- [sma14] *Samsung Smart Bike.* <http://www.maestrosacademy.it/progetto-sbike>, 2014. – Zugriff: 28.12.2014
- [Som10] SOMMER, Ulli: *Arduino – Mikrocontroller-Programmierung mit Arduino/Freeduino*. 85586 Poing : Franzis Verlag GmbH, 2010. – ISBN 978-3-446-43823-1
- [SQLa] *About SQLite.* <http://www.sqlite.org/about.html>, . – Zugriff: 19.01.2015
- [SQLb] *SQLite Is A Zero-Configuration Database.* <http://www.sqlite.org/zeroconf.html>, . – Zugriff: 19.01.2015

- [Tho09] THOMA, Stephan: *Mensch-Maschine-Interaktion*, Technische Universität München, Diss., Oktober 2009
- [Toy11] TOYOTA: *TMC Develops Navigation System Compatible with Vehicle-Infrastructure Co-operative Safety System*. <http://www2.toyota.co.jp/en/news/11/06/0629.html>, 2011. – Zugriff: 08.12.2014
- [tra] *Verkehrsoptimierung mit Genetischen Algorithmen – Projektpartner*. <http://www.travolution-ingolstadt.de/index.php?id=73>, . – Zugriff: 30.12.2014
- [van] *Vanhawks*. <https://www.vanhawks.com/>, . – Zugriff: 28.12.2014
- [Zim84] ZIMDAHL, Walter: Guidelines and some developments for a new modular driver information system / Volkswagenwerk AG. 1984. – Forschungsbericht. – 178 – 182 S.