

INHALT

1 Grundlagen	2
1.1 Technische Grundlagen	2
1.1.1 Android	2
1.1.2 SQLite	4
1.1.3 Mobile Sensorik <– Ich benutze nur GPS?	5
1.1.4 Geolokation mittels Global Positioning System (GPS)	5
1.2 Berechnung der Geschwindigkeitsempfehlung	7
 Akronyme	 8
 Glossar	 9
 Abbildungsverzeichnis	 10
 Literaturverzeichnis	 11

1 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.

1.1 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 Lichtsignalanlage (LSA) die optimale Geschwindigkeit für das Erreichen der “Grünen Welle“ zu errechnen. Im folgenden Abschnitt werden Funktionsweise und Besonderheiten der verwendeten Technologien beschrieben.

1.1.1 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 1.1 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 1.1:

- **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
- **FREE TYPE** 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 eigene 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 1.1 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.

1.1.2 SQLite

SQLite ist eine eingebettete Datenbank-Engine die auch bei Android sehr beliebt ist. Der Code ist Open Source und frei für den privaten oder gewerblichen Einsatz.

Im Gegensatz zu den meisten anderen Structured Query Language (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 Netzwerkkommunikation vonnöten.

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-Applikationen (Apps).

RELATIONALE DATENBANKEN

SQLITE UNTER ANDROID

1.1.3 Mobile Sensorik ←– Ich benutze nur GPS?

Ein Sensor³ 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 diese Sensoren zugreifen und deren Daten erfassen kann. [andf]

1.1.4 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 ?? 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 genaueste 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)

Android unterstützt mit dem `android.location` Paket den Zugriff auf die Ortungsdienste. Als zen-

³ aus dem Lateinischen, deutsch: "fühlen"

trale Komponente des Location Frameworks stellt der `LocationManager` APIs zur Lokalisierung des Geräts bereit. Mit dem `LocationManager` ist die Anwendung in der Lage alle Location Provider⁴ 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 (Longitude) 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.

⁴ deutsch: Standortanbieter. Ein Standortanbieter bietet regelmäßige Berichte über die geographische Lage des Gerätes

1.2 BERECHNUNG DER GESCHWINDIGKEITSEMPFEHLUNG

Präsentiert das System während der Anwendung eine Geschwindigkeitsempfehlung, ist diese abhängig von der Fahrtgeschwindigkeit und vom Abstand zur Ampel. 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 Ampelschaltpläne 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 1.2 soll die Berechnungsgrundlagen veranschaulichen:

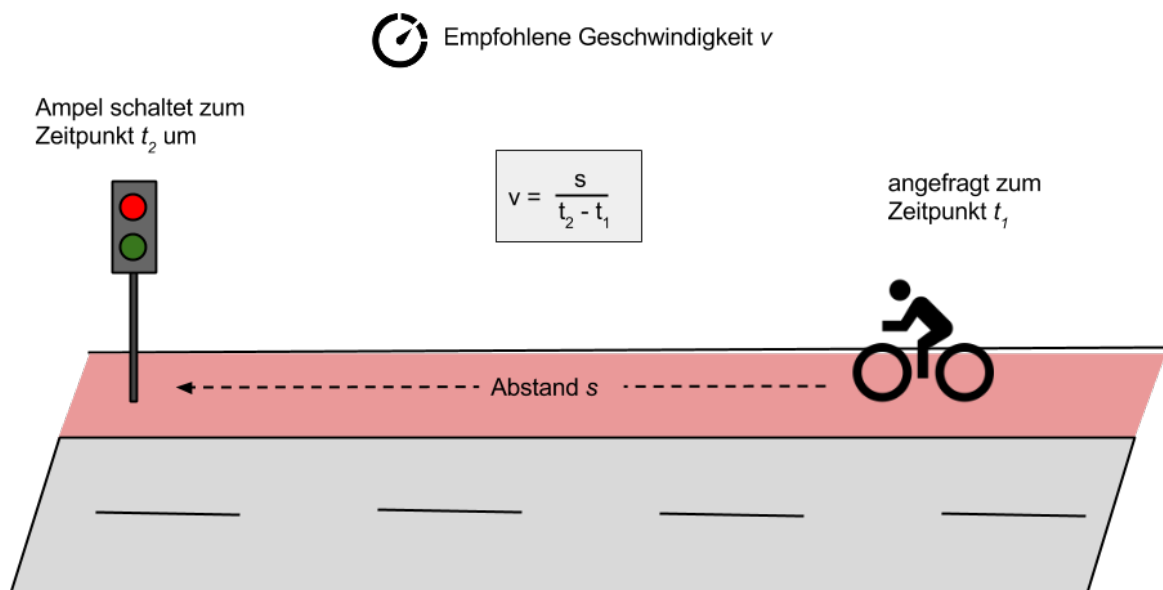


Abbildung 1.2 Veranschaulichung der Berechnung

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

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 Wireless Local Area Network (WLAN), Bluetooth oder Dedicated Short Range Communication (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 berührungsempfindlichen Bildschirm (Touchscreen) und diverse Sensoren, wie dem GPS unterscheidet. So ist eine Interaktion mit der Umgebung und den AnwenderInnen möglich.

ABBILDUNGSVERZEICHNIS

1.1	Android-Systemarchitektur	3
1.2	Berechnung Progressionsgeschwindigkeit	7

QUELLCODEVERZEICHNIS

LITERATURVERZEICHNIS

- [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
- [FA11] FERRARO, Richard ; AKTIHANOGLU, Murat: *Location-Aware Applications*. Shelter Island : Manning Publications Co., 2011. – ISBN 978–1–935182–33–7
- [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
- [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