

Assisted Driving

SE-Projekt

Systems Engineering



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

Oleg Tydynyan 53 33 13

Max Wahl 53 33 26

Robert Ledwig 53 33 16

28. Oktober 2015

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Listingverzeichnis	V
Formelverzeichnis	VI
Abkürzungsverzeichnis	VII
1 Einleitung	1
1.1 Inhalt der Arbeit und Gliederung	1
2 Lastenheft	2
2.1 Einführung	2
2.2 Zielsetzung	2
2.3 Anforderungen	2
2.3.1 Definition der Schilder	3
2.4 Spezifikation des Systems	4
2.4.1 Hauptkomponenten	4
2.4.2 Systemumgebung	4
2.4.3 Systemspezifikationen	4
2.5 Nicht unterstützte Informationssysteme	4
2.6 Grenzen des Systems	5
2.7 Gesamtarchitektur	5
2.7.1 Prinzipieller Ablauf	5
3 Pflichtenheft	6
3.1 Projektziel/Einführung	6
3.2 Anforderungsanalyse	6
3.2.1 Muss-Anforderungen	6
3.2.2 Soll-Anforderungen	7
3.2.3 Kann-Anforderungen	7
3.3 Entwicklungstools	8

3.4	Hauptkomponenten	8
3.4.1	Algorithmischer Ansatz	8
3.4.2	Hardware	9
3.4.3	Kamera	9
3.5	Probleme	9
3.5.1	Potentielle Probleme im Entwicklungsprozess	9
3.5.2	Externe Einflüsse im Betrieb	9
3.6	Ablaufplan	10
3.7	Phasenplan	11
4	Systemarchitektur	13
4.1	funktionale Systemarchitektur	13
4.2	Ableitung der Hardwareanforderungen	13
4.3	Ableitung der Softwareanforderungen	13
4.4	Architektur des Gesamtsystems	13
4.4.1	Technische Systemarchitektur	13
4.4.1.1	statisch	13
4.4.1.2	dynamisch	13
5	Softwarearchitektur	14
5.1	Theoretische Grundlagen der verwendeten Algorithmik	14
5.2	Softwarearchitektur	14
6	Systemtest und Optimierung	15
7	Zusammenfassung und Ausblicke	16
	Literaturverzeichnis	17
A	Anhang Code	i
A.1	Code 1	i
A.2	Code 2	i
B	Anhang Bilder	ii
B.1	Bilder	ii

Abbildungsverzeichnis

2.1	Vorfahrt gewähren	3
2.2	Halt! Vorfahrt gewähren	3
2.3	Vorfahrt (nur an der nächsten Kreuzung / Einmündung)	3
2.4	Vorfahrtsstraße	3
2.5	Verlauf der Vorfahrtsstraße	3
2.6	geplanter Ablauf	5
3.1	Modell	6
3.2	Ablaufplan	10

Tabellenverzeichnis

3.1	Phasenplan	11
-----	----------------------	----

Listingverzeichnis

Formelverzeichnis

Abkürzungsverzeichnis

1 Einleitung

1.1 Inhalt der Arbeit und Gliederung

2 Lastenheft

2.1 Einführung

Die nachfolgende Analyse enthält alle an das zu entwickelnde System verbindlich gestellten Anforderungen. Es ist die wichtigste Grundlage für die Anfertigung der Systemspezifikation (Pflichtenheft) und somit essentieller Bestandteil des Projekts. Dies beinhaltet sowohl die funktionalen und nicht-funktionalen Anforderungen, die Zielsetzung, das Verwendungsfeld, als auch weitere Eckdaten für das Projekt.

2.2 Zielsetzung

Ziel des Projekts ist es ein System zu entwickeln, welches in der Lage ist (Vorfahrts-) Schilder im Straßenverkehr selbstständig zu erfassen. Diese Informationen sollen verarbeitet und an den Fahrer ausgegeben werden. Das System wird so ausgelegt, dass es um die Erkennung weiterer Schilder erweitert werden kann. Die Systementwicklung schließt die Bereitstellung einer selbst entworfenen Software auf einer gewählten Hardwareplattform mit ein. Das finale System soll unter realen Bedingungen (im Fahrzeug, wahlweise in einem Modell) verwendet werden können.

2.3 Anforderungen

In diesem Abschnitt werden die gewünschten Funktionalitäten bzw. das Verhalten des Systems erläutert. Es werden Aspekte erläutert, wie Leitung des Systems, Eingaben-Verarbeitung-Ausgaben, sowie die Grenzen des Systems. Ebenso wird auf die Qualität des Systems eingegangen. Die (gewünschte/geforderte) Funktionalität an sich spielt dabei eine ebenso entscheidende Rolle, wie die Bedingungen unter denen die Funktionalität ausgeführt wird.

2.3.1 Definition der Schilder

Das System soll in der Lage sein, alle in der deutschen Straßenverkehrsordnung festgelegten Vorfahrtszeichen korrekt zu erkennen. Die Trefferquote muss bei mindestens 50% liegen. Zunächst sollen die folgenden fünf Verkehrszeichen erkannt werden:



Abbildung 2.1: Vorfahrt gewähren



Abbildung 2.2: Halt! Vorfahrt gewähren



Abbildung 2.3: Vorfahrt (nur an der nächsten Kreuzung / Einmündung)



Abbildung 2.4: Vorfahrtsstraße



Abbildung 2.5: Verlauf der Vorfahrtsstraße

Anschließend ist eine Erweiterung auf Sonderregelungen (Kreisverkehr) sowie weitere Verkehrszeichen (z.B. Geschwindigkeitsbeschränkungen) vorgesehen.

2.4 Spezifikation des Systems

2.4.1 Hauptkomponenten

Das finale System soll aus folgenden Hauptkomponenten bestehen:

- hochauflösende (Stereo-)Kamera
- Recheneinheit
- Ausgabegerät

Die Kamera kann frei gewählt werden, muss jedoch mit der verbauten Recheneinheit zusammenarbeiten. Als Recheneinheit wäre ein Einplatinencomputer (Bsp. Raspberry Pi) denkbar. Die Ausgabe des erkannten Schildes soll optisch und akustisch erfolgen (Display, HUD, Warnsignal).

2.4.2 Systemumgebung

Das System wird so entwickelt, dass es in einem beliebigen Auto verbaut und genutzt werden kann. Die Kamera wird hinter der Windschutzscheibe mit ?Blick? auf die Straße platziert und muss so angebracht werden, dass die Bilderfassung durch Vibrationen während der Fahrt nicht gestört wird.

Als spätere Erweiterung ist eine Anbindung an die CAN-Kommunikation des Auto-Steuergerätes denkbar (z.B. zur Geschwindigkeitserfassung).

2.4.3 Systemspezifikationen

Die Erfassung der Verkehrszeichen soll bis mind. 60 km/h erfolgen. Je nach erfasstem Zeichen muss die Anzeigedauer auf einem Ausgabegerät (z.B. Display) unterschiedlich lang erfolgen. Es muss sichergestellt werden, dass keine anderen von der Kamera erfassten Objekte (z.B. Aufkleber, Werbetafeln) als Verkehrszeichen fehlinterpretiert werden.

2.5 Nicht unterstützte Informationssysteme

Nicht unterstützt werden elektronische Anzeige-Tafeln, sowie Kombischilder. Beides ist jedoch als spätere Erweiterung denkbar.

2.6 Grenzen des Systems

Das finale System muss nur unter idealen Bedingungen einwandfrei funktionieren. Dies schließt den Betrieb am Tag unter idealen Belichtungen ein. Verkehrszeichen dürfen nicht verdreht, verdreht oder aus anderen Gründen unleserlich sein. Verkehrszeichen dürfen einen Mindestabstand von 20m zueinander besitzen. Fahrzeuggeschwindigkeiten über 60 km/h garantieren keine fehlerfreie Erfassung mehr.

2.7 Gesamtarchitektur

2.7.1 Prinzipieller Ablauf

Das nachfolgende Diagramm verdeutlicht den Ablauf des geplanten Programms.

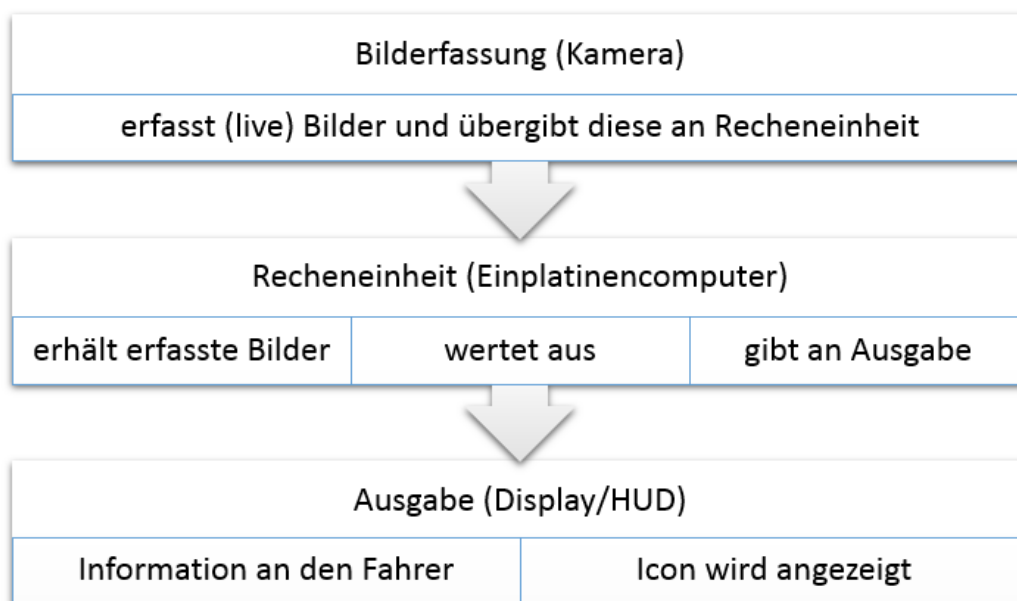


Abbildung 2.6: geplanter Ablauf

3 Pflichtenheft

3.1 Projektziel/Einführung

Ziel des Projekts ist es ein System zu entwickeln, welches in der Lage ist (Vorfahrts-) Schilder im Straßenverkehr autonom zu erfassen. Die Erfassung erfolgt ? während der Fahrt ? mit Hilfe einer Kamera. Ist das System aktiv, läuft die Kamera stets mit und es werden alle erfassten Bilder in die zentrale Recheneinheit (CPU) weitergeleitet. Dort werden diese mittels einer Bilderfassungssoftware analysiert und die Informationen der erkannten Verkehrsschilder an den Fahrer weitergeleitet. Die Ausgabe erfolgt visuell; auf einem Display wird das erfasste (Vorfahrts-) Schild durch ein Icon dargestellt.

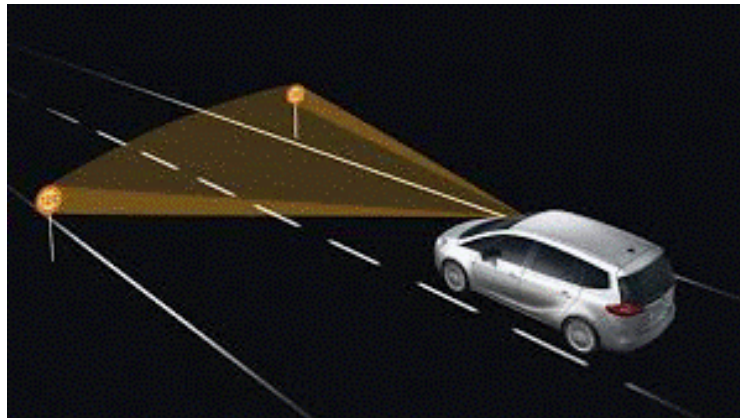


Abbildung 3.1: Modell

3.2 Anforderungsanalyse

3.2.1 Muss-Anforderungen

Die hier beschriebenen Punkte müssen erfüllt werden, um das Projekt erfolgreich abzuschließen.

1. Es werden mindestens zwei deutsche Vorfahrtsschilder (Stopp-Schild und ?Vorfahrt beachten?) korrekt erkannt.

2. 51% aller Schilder werden korrekt erkannt.
3. Die Bilderfassung erfolgt korrekt (siehe M2) bis zu einer Fahrzeuggeschwindigkeit von 30 km/h.
4. Die Bilderfassung erfolgt mit einer Abtastrate von mindestens 1 Hz (max. möglicher Schilder-Abstand dadurch ca. 10m bei 30 km/h).
5. Die von der Kamera aufgenommenen Bilder werden in eine niedrigere Auflösung umgerechnet. Diese und die maximal mögliche Auflösung der aufgenommenen Bilder sind zu bestimmen (unter Beachtung von M2 und M4).
6. Die Ausgabe des erkannten Zeichens erfolgt über eine LED (eine je erkanntes Schild) nach maximal 1 Sekunde.

3.2.2 Soll-Anforderungen

Die folgenden Anforderungen sollen umgesetzt werden. Sie sind fest in den Projektplan integriert, sind jedoch nicht projektentscheidend.

1. Es werden alle deutschen Vorfahrtszeichen korrekt erkannt.
2. Die Bilderfassung erfolgt korrekt (siehe M2) bis zu einer Fahrzeuggeschwindigkeit von 60 km/h.
3. Die Bilderfassung erfolgt mit einer Abtastrate von mindestens 2 Hz (max. möglicher Schilder-Abstand dadurch ca. 10m bei 60 km/h).
4. Die Geschwindigkeit des Fahrzeuges, in dem das System verbaut wird, soll über ein CAN-Bus-System erfasst werden. Bei der Überschreitung von 60 km/h soll eine Warnung ausgegeben werden.
5. Die Anzeige des erkannten Zeichens soll über ein Display erfolgen, in welchem das erkannte Verkehrszeichen abgebildet wird (nach max. 0,5 Sekunden), das Icon erscheint 15 s oder bis zur Erkennung eines neuen Verkehrszeichens.

3.2.3 Kann-Anforderungen

Die folgenden Anforderungen sind optional und können im Laufe des Projektes erreicht werden, bzw. dienen als Ausgangspunkt für mögliche Erweiterungen. Sie sind nicht notwendig, um das Projekt zu einem erfolgreichen Abschluss zu führen.

1. Das System kann weitere Verkehrsschilder erkennen.

2. Die Bilderfassung wird mit einem Smartphone durchgeführt, welches vom Nutzer gestellt wird.
3. Die erkannten Verkehrszeichen werden über ein HUD (Head-up-Display) in die Windschutzscheibe in den Sichtbereich des Fahrers projiziert.
4. Zusätzlich zur Anzeige wird ein akustischer Warnton ausgelöst, wenn z.B. ein Stoppschild ohne Anhalten überfahren wurde.
5. Durch Erweiterung auf eine Car-2-Car Communication (WLAN- und GPS-basiert) wird vor einem bevorstehenden Unfall gewarnt, andere Verkehrsteilnehmer in der Nähe werden z.B. bei überfahrenen Stopp-Schildern gewarnt und ggf. automatisch gebremst.

3.3 Entwicklungstools

Die zu entwerfende Software wird mit Hilfe der Entwicklungsumgebung "Visual Studio 2013" ¹ entworfen, unter Nutzung der Open-Source-Bibliothek "OpenCV" ². Das fertige System soll auf einer für diese Anwendung optimierten und mit dem Build-Werkzeug "yocto" ³ erstellten Linux-Distribution in Betrieb genommen werden. Für deren Start- und Flashvorgang auf der gewählten Hardware kommt der Open-Source-Bootloader "U-Boot" ⁴ zum Einsatz.

3.4 Hauptkomponenten

3.4.1 Algorithmischer Ansatz

Die von der Kamera erfassten Bilder werden unter Verwendung geeigneter Filter-Algorithmen vorbearbeitet und skaliert. Anschließend durchlaufen die nun erhaltenen Daten einen für die einzelnen Verkehrszeichen spezifischen Farbfilter. Daraufhin können geeignete Objekterkennende Algorithmen (Feature-Tracking und ggf. Viola-Jones) genutzt werden, um die gewünschten Schilder eindeutig zu identifizieren. Die genutzten Algorithmen werden so gewählt, dass auch nur teilweise erkennbare sowie

¹<https://www.visualstudio.com/>
letzter Zugriff: 25.10.2015

²<http://opencv.org/>
letzter Zugriff: 25.10.2015

³<https://www.yoctoproject.org/>
letzter Zugriff: 25.10.2015

⁴<http://www.denx.de/wiki/U-Boot/WebHome>
letzter Zugriff: 25.10.2015

farblich variierende Zeichen (z.B. bei unterschiedlichen Lichtstimmungen) erkannt werden können und dabei gleichzeitig ausreichend Performance garantieren.

3.4.2 Hardware

Die fertig entwickelten Programme werden auf ein ?MarS-Board? der Firma Embest Technologies überspielt. Dieses stellt neben einem ausreichend schnellen ARM Cortex-A9-Prozessor und zwei USB-Schnittstellen an denen direkt die benötigte Kamera angebunden werden kann, auch einen speziellen Grafikprozessor (Vivante GC2000) der zur Vorverarbeitung der eingehende Bilder genutzt werden kann. Eine Ethernet-Schnittstelle erlaubt ebenfalls den Zugriff auf Kameras über ein Netzwerk.

3.4.3 Kamera

Verwendet wird die USB3.0-Kamera mxBlueFox3 der Firma Matrix Vision. Diese bietet mit 10 Megapixel ein Vielfaches der geforderten Auflösung und steht im Labor der HTW zur Verfügung.

3.5 Probleme

3.5.1 Potentielle Probleme im Entwicklungsprozess

Größte Herausforderung im Entwicklungsprozess wird die Anpassung der gewählten Algorithmen sein, so dass diese einen akzeptablen Kompromiss zwischen Performance (maximale Bildübertragungsrate bei gewünschter Auflösung) und Erfolgsrate der Objekterkennung bietet. Es ist zu untersuchen unter welchen Bedingungen das System noch einwandfrei arbeitet und welche Verkehrszeichen besonders schwierig zu erkennen sind. Die folgende Optimierung des Codes wird aufgrund der Komplexität der Bildverarbeitenden Algorithmen viel Zeit in Anspruch nehmen.

3.5.2 Externe Einflüsse im Betrieb

Das finale System wird stets nur eine begrenzte Erfolgsquote bei der Erkennung der Verkehrszeichen aufweisen. Im Straßenverkehr können Schilder verdreht oder verdreckt sein und je nach Tageszeit und Wetter unterschiedlich schwer wahr zu nehmen sein. Ziel muss es sein, diese Störeinflüsse zu identifizieren und klare Regeln zu definieren, unter denen das System noch einwandfrei arbeitet.

3.6 Ablaufplan

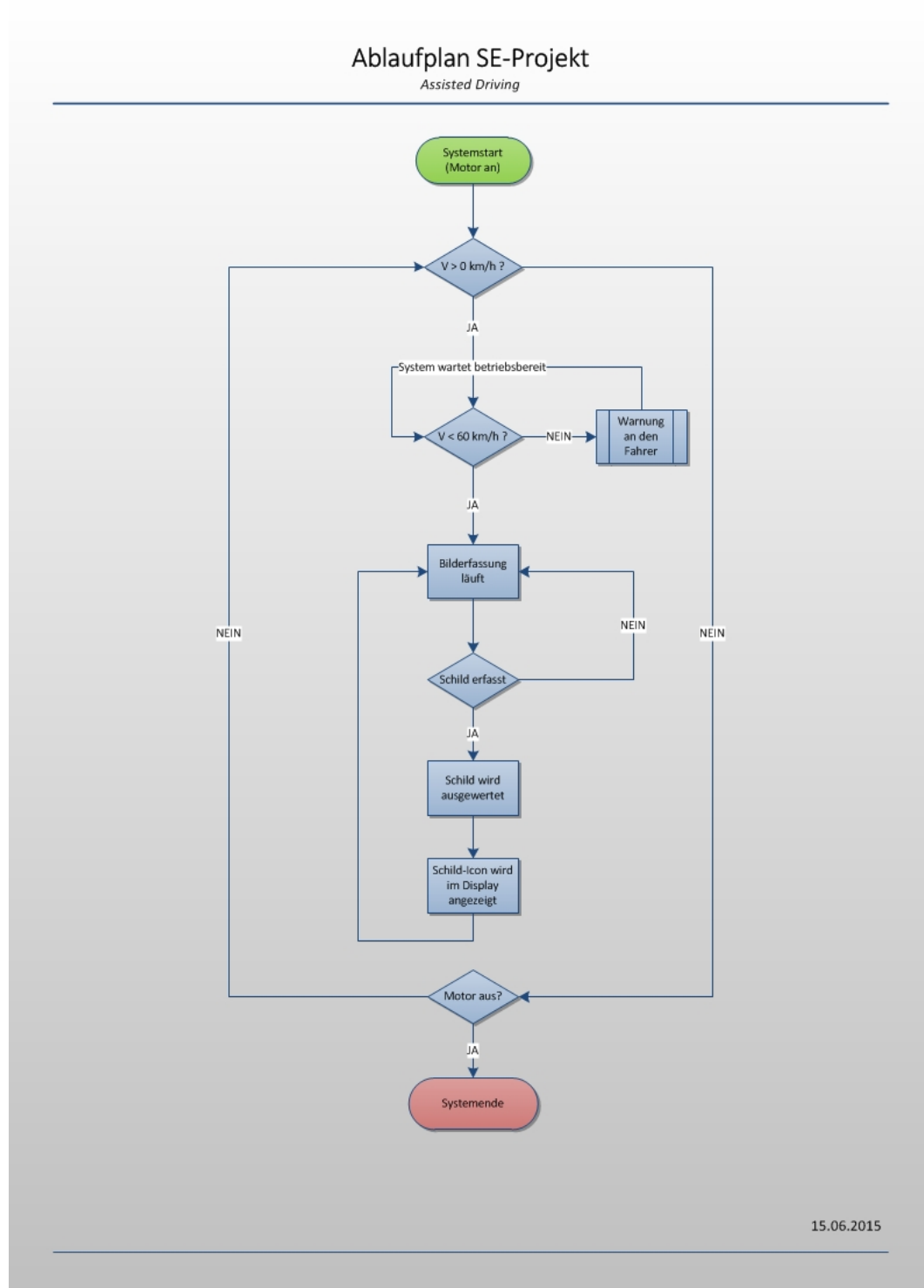


Abbildung 3.2: Ablaufplan

3.7 Phasenplan

Phase	Ziel/Meilenstein	Zeitraum
Recherche	Lastenheft	25.04.15 - 04.05.15
Konzeptphase	Pflichtenheft	28.04.15 - 18.06.15
Entwurfsphase	Fertigstellung Algorithmen	05.06.15 - 03.07.15
Fertigungsphase	Lauffähiges System	25.06.15 - 11.12.15
Betriebsphase	Abgeschlossenes Projekt	13.11.15 - 07.01.16
Präsentationsphase	Abgabe Projekt	21.05.15 - 11.02.16

Tabelle 3.1: Phasenplan

Der Phasenplan des Projektes besteht insgesamt aus sieben Phasen. Diese Phasen besitzen teils einen fließend zeitlichen Übergang zueinander.

Zuerst erfolgt in der Recherchephase (April bis Mai 2015) das Vertraut-Machen der Projektmitglieder mit der Entwicklungsumgebung "Microsoft Visual Studio in Kombination mit den OpenCV Bibliotheken. Außerdem werden die Anforderungen (Spezifikationen) an das Projekt definiert; das Lastenheft entsteht.

Zeitlich versetzt zur Recherchephase erfolgt bereits die Konzeptphase (April bis Juni 2015), die in dem Pflichtenheft endet. Die Projektanforderungen aus dem Lastenheft werden validiert und es erfolgt eine Aufschlüsselung nach Muss-, Soll-, und Kann-Kriterien hinsichtlich Umsetzbarkeit. Ebenso wird nach diesen Kriterien die Hard- und Software abgeleitet sowie ein erster algorithmischer Ansatz zur Problemlösung erstellt.

Von Juni bis Juli 2015 verläuft die Entwurfsphase. Hierin informieren sich die Projektteilnehmer über die Algorithmik zur Objekterkennung. Erste Testmuster zur Detektierung bestimmter Bildausschnitte werden erstellt. Diese Vorabuntersuchung erfolgt zunächst unabhängig von der später zu verwendenden Hardware.

In der Ende Juni 2015 startenden Fertigungsphase erfolgt zunächst die Beschaffung der Hardware, deren anschließende Inbetriebnahme (Schaffung Softwarebasis Mars-Board) sowie die Code-Implementierung. Ist die Fertigungsphase abgeschlossen, ist der Meilenstein "lauffähiges System" erreicht (Dezember 2015).

Die im November 2015 beginnende Betriebsphase umfasst die Implementierung des Gesamtsystems in der Fahrzeugumgebung. Praktische Tests werden durchgeführt sowie eventuelle softwaretechnische Fehler behoben. Im Januar 2016 ist der Abschluss der Betriebsphase geplant.

Die Präsentationsphase erfolgt parallel zu allen anderen erwähnten Phasen. Es erfolgen hierbei die Präsentationen zu den Zwischenständen, finalen Abschlüssen des

3 Pflichtenheft

Projektes und das Pflegen der Dokumentation. Die Präsentationsphase endet im Februar 2016 mit der Projektabgabe.

4 Systemarchitektur

4.1 funktionale Systemarchitektur

4.2 Ableitung der Hardwareanforderungen

4.3 Ableitung der Softwareanforderungen

4.4 Architektur des Gesamtsystems

4.4.1 Technische Systemarchitektur

4.4.1.1 statisch

4.4.1.2 dynamisch

5 Softwarearchitektur

5.1 Theoretische Grundlagen der verwendeten Algorithmik

5.2 Softwarearchitektur

6 Systemtest und Optimierung

7 Zusammenfassung und Ausblicke

Literaturverzeichnis

Abaxor.de

ABAXOR.DE: *FPGA-Entwicklung: Vorteile von FPGAs gegenüber Software/Prozessoren*. <http://www.abaxor.de/vorteile-fpga.html>. – letzter Zugriff: 22.01.2015

AlteraDatasheet

ALTERADATASHEET: *Cyclone Family Data Sheet*. <http://www.datasheetarchive.com/dlmain/SFDatasheet-4/sf-00095700.pdf>. – letzter Zugriff: 05.01.2015

AlteraVol1

ALTERAVOL1: *Cyclone Device Handbook, Volume 1*. http://www.altera.com/literature/hb/cyc/cyc_c5v1.pdf. – letzter Zugriff: 05.01.2015

AlteraVol2

ALTERAVOL2: *Cyclone Device Handbook, Volume 2*. http://users.ece.gatech.edu/~hamblen/UP3/cyc_c5v2.pdf. – letzter Zugriff: 05.01.2015

Diller-Technologies

DILLER-TECHNOLOGIES: *I2C*. <http://www.diller-technologies.de/stm32.html>. – letzter Zugriff: 05.01.2015

KT-electronic

KT-ELECTRONIC: *Ultraschall Messmodul HC-SR04*. http://www.mikrocontroller.net/attachment/218122/HC-SR04_ultraschallmodul_beschreibung_3.pdf. – letzter Zugriff: 05.01.2015

Mikrocontroller.net a

MIKROCONTROLLER.NET: *Ausgangsstufen Logik-ICs*. http://www.mikrocontroller.net/articles/Ausgangsstufen_Logik-ICs. – letzter Zugriff: 05.01.2015

Mikrocontroller.net b

MIKROCONTROLLER.NET: *Field Programmable Gate Array*. <http://www.mikrocontroller.net/articles/FPGA>. – letzter Zugriff: 05.01.2015

RN-Wissen.de

RN-WISSEN.DE: *I2C*. <http://rn-wissen.de/wiki/index.php/I2C>. – letzter Zugriff: 05.01.2015

STMicroelectronicsData

STMICROELECTRONICS DATA: *STM32F4 Datasheet*. <http://www.st.com/web/en/resource/technical/document/datasheet/DM00037051.pdf>. – letzter Zugriff: 05.01.2015

STMicroelectronicsRef

STMICROELECTRONICS REF: *RM0090 Reference Manual*. http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf. – letzter Zugriff: 05.01.2015

Tuttas

TUTTAS, Jörg: *Automaten*. <http://www.rts.uni-hannover.de/labor/pneu/skript/node4.html>. – letzter Zugriff: 28.01.2015

Wikipedia.org

WIKIPEDIA.ORG: *Field Programmable Gate Array*. http://de.wikipedia.org/wiki/Field_Programmable_Gate_Array. – letzter Zugriff: 05.01.2015

Winner 2009

WINNER, Hermann: *Handbuch Fahrerassistenzsysteme*. Vieweg+Teubner, 2009. – ISBN 978

A Anhang Code

Die Codedateien befinden sich unter folgenden Pfaden im Anhang auf CD:

A.1 Code 1

- Pfad 1: \Code\STM\STM32F4_Project_BA\application\hc_sr04_app.h

A.2 Code 2

- Pfad 2: \Code\STM\STM32F4_Project_BA\application\hc_sr04_app.h

B Anhang Bilder

B.1 Bilder

Die Bilder befinden sich in dem Ordner `Ordnername` auf CD.