Bachelorarbeit

**LiDAR basierte Umgebungserfassung**

vorgelegt von Michael Hoffmann



Fakultät für Elektrotechnik und Informationstechnik

Regenerative Energien – Elektrotechnik

|  |  |
| --- | --- |
| Betreuer: | Prof. Dr. Gerhard Schillhuber |
| Laufende Nummer: | 1499 |
| Bearbeitungsbeginn: | 15.03.2017 |
| Abgabetermin: | 18.08.2017 |



|  |  |
| --- | --- |
| Firma: | Objective Software GmbH |
| Betreuer: | Alexander Dratva |

Bachelorarbeit

**LiDAR basierte Umgebungserfassung mit Schwenk-/Neigemotor und Orientierungseinheit**

**LiDAR based environment scanning with  
 pan-tilt motor and orientation unit**

vorgelegt von Michael Hoffmann

an der

Hochschule München

Fakultät für Elektrotechnik und Informationstechnik

Regenerative Energien – Elektrotechnik

|  |  |
| --- | --- |
| Betreuer: | Prof. Dr. Gerhard Schillhuber |
| Laufende Nummer: | 1499 |
| Bearbeitungsbeginn: | 15.03.2017 |
| Abgabetermin: | 18.08.2017 |

|  |  |
| --- | --- |
| Firma: | Objective Software GmbH |
| Betreuer: | Alexander Dratva |

Erklärung des Bearbeiters

Name: Hoffmann  
Vorname: Michael

Ich erkläre hiermit, dass ich die vorliegende Bachelorarbeit selbständig verfasst und noch nicht anderweitig zu Prüfungszwecken vorgelegt habe.

Sämtliche benutzte Quellen und Hilfsmittel sind angegeben, wörtliche und sinngemäße Zitate sind als solche gekennzeichnet.

Ottobrunn, den 18.08.2017 Unterschrift

Kurzfassung

Meine Bachelorarbeit wurde bei der Firma Objective Software GmbH in München durchgeführt. Die Firma hat ein neues Konzept zur Parkraumerfassung auf Freiflächen entwickelt.

Ziel der Arbeit war es, einen Teil des Entwurfes mit einem echten Prototyp zu evaluieren. Das Konzept zur Parkraumerfassung bestand nur aus Anforderungen an das System. Sämtliche Planungs-, Architektur- und Designarbeiten sollten als erster Schritt der Bachelorarbeit durchgeführt werden. Darüber hinaus sollte der Hardwareaufbau und Softwareimplementierung entwickelt werden.

Das Gesamtsystem sollte durch motorische Schwenkbewegungen einen möglichst großen Parkplatzbereich erfassen. Hierzu wurden ein LiDAR, ein Motor und eine Orientierungseinheit verwendet. Des Weiteren wurde ein Algorithmus zum Errechnen der Punkte und für die Motorsteuerung entworfen. Für Test- und Präsentationszwecke wurde außerdem eine Halterung angefertigt.

Abstract

I did my bachelor thesis at Objective Software GmbH, a company located in Munich. The company created a new concept to detect outdoor parking space.

The goal of the thesis was to implement a proof of concept with a real prototype. The original concept of parking lot detection contained only system requirements. All planning work, architecture and design, should be done as a first step. Additionally the hardware installation and software implementation should be developed.

The complete system should scan a large parking area by motor-driven pivoting. Therefore a LiDAR, a motor and an orientation unit were used. Also an algorithm for point cloud calculation and the motor-control was engineered. Besides this a frame was constructed for testing and showcases.

Sperrvermerk

Diese vorliegende Bachelorarbeit mit dem Titel *LiDAR basierte Umgebungserfassung* enthält vertrauliche Daten des Unternehmens Objective Software GmbH. Diese Arbeit darf Dritten, mit Ausnahme des betreuenden Dozenten und befugten Mitgliedern des Prüfungsausschusses, ohne ausdrückliche Zustimmung des Unternehmens nicht zugänglich gemacht werden. Eine Vervielfältigung und Veröffentlichung der Bachelorarbeit – auch in Auszügen – ist grundsätzlich nicht erlaubt.

Inhaltsverzeichnis

[Erklärung des Bearbeiters III](#_Toc105195048)

[Kurzfassung IV](#_Toc105195049)

[Abstract IV](#_Toc105195050)

[Sperrvermerk V](#_Toc105195051)

[Inhaltsverzeichnis VI](#_Toc105195052)

[Abbildungsverzeichnis VIII](#_Toc105195053)

[Tabellenverzeichnis IX](#_Toc105195054)

[Formelverzeichnis IX](#_Toc105195055)

[1 Einleitung 1](#_Toc105195056)

[1.1 Smart Home 1](#_Toc105195057)

[1.2 Ziel der Arbeit / Requirements 1](#_Toc105195058)

[1.3 Gliederung der Arbeit 2](#_Toc105195059)

[2 Hardware 3](#_Toc105195060)

[2.1 Beschreibung des Komplettsystems 4](#_Toc105195061)

[2.1.1 KNX - Bus 4](#_Toc105195062)

[2.1.2 Lüfter 4](#_Toc105195063)

[2.1.3 Feuchtigkeitssensor 4](#_Toc105195064)

[2.1.4 Treiber 4](#_Toc105195065)

[2.1.4.1 Ansteuerprinzipien 4](#_Toc105195066)

[2.1.4.2 Vorgehen 4](#_Toc105195067)

[2.1.4.3 Schaltplan 4](#_Toc105195068)

[2.1.4.4 Layout 4](#_Toc105195069)

[2.1.4.5 Bauteilauswahl 4](#_Toc105195070)

[3 Software 4](#_Toc105195071)

[3.1 Mikrocontroller 4](#_Toc105195072)

[3.2 Softwarestruktur 4](#_Toc105195073)

[3.3 Uart 4](#_Toc105195074)

[3.4 Timer 4](#_Toc105195075)

[3.5 Interrupt/Callback 4](#_Toc105195076)

[3.6 Programmablauf 4](#_Toc105195077)

[3.6.1 statemachine\_process 4](#_Toc105195078)

[3.6.2 statemachine\_uart 4](#_Toc105195079)

[3.7 HAL\_UART\_Receive\_IT in detail? 4](#_Toc105195080)

[3.8 Regler 4](#_Toc105195081)

[3.8.1 Allgemein 4](#_Toc105195082)

[3.8.2 Reglertypen 5](#_Toc105195083)

[3.8.2.1 Zwei Punkt Regler 5](#_Toc105195084)

[3.8.2.2 P-Regler 5](#_Toc105195085)

[3.8.2.3 PI-Regler 5](#_Toc105195086)

[3.8.2.4 PID-Regler 5](#_Toc105195087)

[3.8.3 Realisierung 5](#_Toc105195088)

[4 Ausblick 5](#_Toc105195089)

[4.1.1 Aktueller Stand der Arbeit 5](#_Toc105195090)

[4.1.2 Empfehlungen für Hardware 2.0 5](#_Toc105195091)

[4.1.3 Fazit 5](#_Toc105195092)

[4.1.4 CMake Hauptdatei 6](#_Toc105195093)

Abbildungsverzeichnis

[Abbildung 1: Systemkonzept 2](#_Toc105195103)

[Abbildung 4: Überblick des Systemaufbaues 5](#_Toc105195104)

[Abbildung 6: Programmablauf des Konfigurationsprogramms 5](#_Toc105195105)

Tabellenverzeichnis

[Tabelle 1: LiDAR - vertikale Winkel 6](#_Toc105195098)

Formelverzeichnis

[Formel 1: Distanz zu gescanntem Objekt 5](#_Toc105195094)

# Einleitung

In diesem Kapitel wird kurz das Unternehmen Objective Software GmbH vorgestellt. Anschließend wird das Problem der derzeitigen Parkraumüberwachung aufgezeigt und ein Konzept zur Lösung beschrieben. Im nächsten Unterkapitel wird die vorhandene, zu nutzende Hardware vorgestellt, gefolgt von der Zielsetzung der Arbeit. Zuletzt wird die Gliederung der Bachelorarbeit dargelegt.

## Smart Home

Meine Bachelorarbeit habe ich bei der Objective Software GmbH durchgeführt. Diese wurde 1998 von Clemens Dannheim in München gegründet.

Derzeit hat das Unternehmen 4 Standorte in 2 Ländern. Die Standorte sind München, Frankfurt, Magdeburg und Turin. Insgesamt beschäftigt Objective Software 150 Mitarbeiter. Unter den mehr als 110 Kunden befinden sich Firmen wie BMW, MAN, Krauss-Maffei Wegmann, Bosch, Intel, Rohde & Schwarz, um nur einige zu nennen.

Die Spezialgebiete des Unternehmens sind Autonomes Fahren, Fahrerassistenzsysteme, die Erstellung von hochgenauen digitalen Karten, In-Car Cyber Security und Implementierung von Embedded-Software. Des Weiteren werden eine Vielzahl von Entwicklungs- und Analyse-Tools für IT-Plattformen angeboten. [1]

## Ziel der Arbeit / Requirements

Das Suchen von Parkplätzen kostet jeden Autofahrer sehr viel Zeit. Auf Grund dessen wird in den meisten modernen Parkhäusern die Parkplatzbelegung mit Hilfe von Sensoren überwacht. Hierdurch kann der Autofahrer schneller zu einem Parkplatz gelotst werden. Für die Parkraumüberwachung wird für jeden Parkplatz ein Sensor verwendet. Dies ist mit einem hohen Material- und Wartungskostenaufwand verbunden. Auf Freiflächen-Parkplätzen wird meistens keine Parkraumüberwachung verwendet. Die in Parkhäusern genutzte Technik wäre hier noch teuer und schwieriger zu realisieren. Die Gründe hierfür sind fehlende Montagemöglichkeiten für die Sensoren, große Entfernungen, Wetterbedingungen.

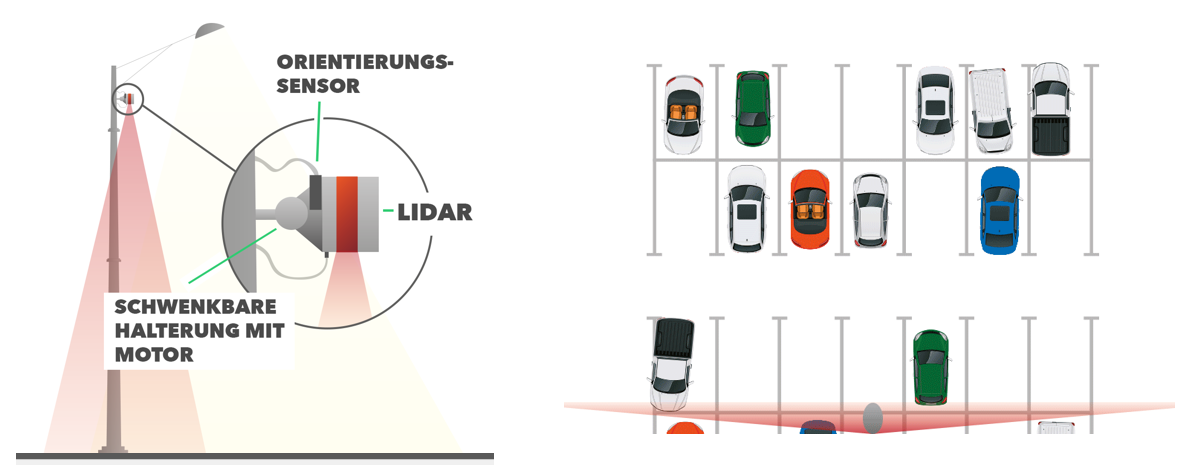


Abbildung : Systemkonzept

## Gliederung der Arbeit

# Hardware

Dieses Kapitel gibt eine Übersicht über die verwendeten Softwaretools und Programmiersprachen bei der Softwareentwicklung.

## Beschreibung des Komplettsystems

### KNX - Bus

### Lüfter

### Feuchtigkeitssensor

### Treiber

#### Ansteuerprinzipien

#### Vorgehen

#### Schaltplan

#### Layout

#### Bauteilauswahl

# Software

## Mikrocontroller

## Softwarestruktur

## Uart

## Timer

## Interrupt/Callback

## Programmablauf

### statemachine\_process

### statemachine\_uart

## HAL\_UART\_Receive\_IT in detail?

## Regler

### Allgemein

### Reglertypen

#### Zwei Punkt Regler

#### P-Regler

#### PI-Regler

#### PID-Regler

### Realisierung

# Ausblick

### Aktueller Stand der Arbeit

### Empfehlungen für Hardware 2.0

### Fazit

Abbildung : Überblick des Systemaufbaues

Abbildung : Programmablauf des Konfigurationsprogramms

.

Formel : Distanz zu gescanntem Objekt

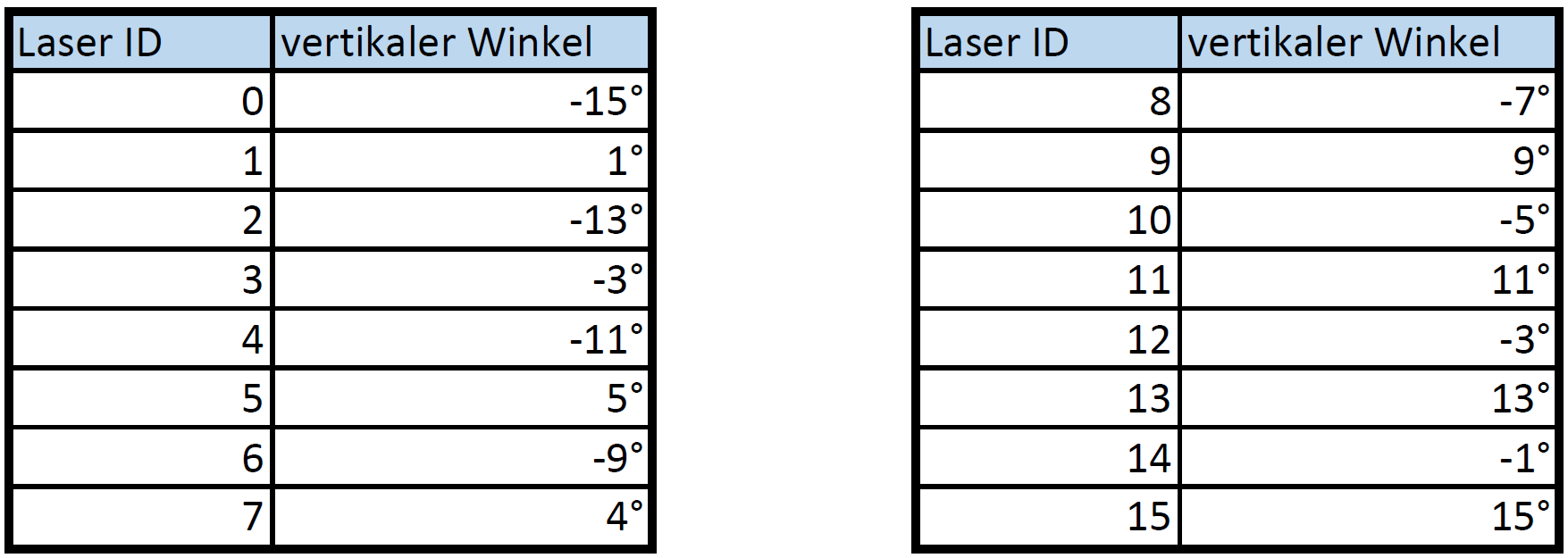


Tabelle : LiDAR - vertikale Winkel

### CMake Hauptdatei

Die CMake Hauptdatei CMakeLists.txt im Root-Verzeichnis muss erstellt werden und es wurden folgende Parameter eingetragen: die minimale CMake Version, Projektname, verwendete C++ Version.

Des Weiteren wurden projektspezifische Einstellungen eingefügt.

1. include\_directories(Source)
2. include\_directories(${xsensInc})
3. include\_directories(${gtest\_SOURCE\_DIR/include ${gtest\_SOURCE\_DIR})

Als nächstes werden die Pfade zu den Headerdateien angegeben. Dazu wird die CMake-Anweisung include\_directories verwendet. Hierdurch wird dem Präprozessor mitgeteilt, an welchen Orten er nach den eingebunden Headerdateien suchen soll. Auf Grund der Übersichtlichkeit wurde für jede verwendete Bibliothek ein eigener CMake-Befehl verwendet. Es wäre auch möglich, alle Pfade in einem Befehl zu schreiben. Um die Pfade nicht komplett ausschreiben zu müssen, wurde für die Pfade Variablen mit dem set-Befehl von CMake erzeugt. Diese können danach mit ${Variablenname} verwendet werden. Der Vorteil hiervon ist, dass bei Pfadänderungen zu den Bibliotheken, diese nur an einer zentralen Stelle geändert werden müssen.

1. add\_subdirectory(Source)
2. add\_subdirectory(Test)
3. add\_subdirectory(Lib/gtest)

Mit dem Befehl add\_subdirectory werden die Ordner angegeben, in denen weitere zu berücksichtigende CMake-Dateien mit Einstellungen zu finden sind. Im Ordner Lib/gtest befindet sich das GoogleTest Framework. Dabei handelt es sich selbst um ein CMake Projekt, das seine eigene CMake-Datei mit allen Einstellungen für die Verwendung der Bibliothek bereitstellt.

In den anderen beiden Ordnern wurde die CMake-Textdatei angelegt. Dazu später mehr.

1. add\_executable(mainprogram main.cpp)
2. add\_executable(configprogram configuration.cpp)
3. add\_executable(tests testRunner.cpp)

Als nächstes werden die Namen der auszuführenden Dateien festgelegt und die dazu verwendete cpp-Datei.

An jedes ausführbare Programm werden die verwendeten Bibliotheken gebunden. Dies geschieht über den Befehl target\_link\_libraries. Dadurch wird dem Linker mitgeteilt, an welchen Stellen die jeweiligen Implementierungen zu den eingebunden Headerdateien zu finden sind.

1. target\_link\_libraries(
2. mainprogram
3. Source
4. ${xsensLibPath}xsensdeviceapi64.lib
5. ${xsensLibPath}xstypes64.lib
6. )

Das Linken wird an Hand des Hauptprogramms gezeigt. Zuerst wird der zuvor festgelegte Name der ausführbaren Datei dem Befehl übergeben, gefolgt von den Bibliotheken. Bei Source handelt es sich um eine selbst erstellte Bibliothek aller Dateien aus dem Source Ordner. Die Erstellung wird im späteren Verlauf beschrieben. Zusätzlich wurden zwei Teile der XDA-Library eingebunden. Die eingebundenen lib-Dateien der

Literaturangaben

[1] Objective Software GmbH: Unternehmen. http://www.objective.de/de/unternehmen/, abgerufen am: 24.05.2017

[2] Axis Communications AB.: Netzteil PS-24. https://www.axis.com/de/de/products/power-adapters/mains-adaptor-ps-24/, abgerufen am: 02.08.2017

[3] Axis Communications AB.: Schwenk-/Neigemotro YP3040. 41755/DE/R1/1101. 2011. https://www.axis.com/files/datasheet/ds\_yp3040\_ptmotor\_41755\_de\_1101\_lo.pdf, abgerufen am: 05.07.2017

[4] Velodyne LiDAR, Inc. - VLP-16 Data Sheet: 63-9229 Rev-F. http://velodynelidar.com/docs/datasheet/63-9229\_Rev-F\_Puck%20\_Spec%20Sheet\_Web.pdf, abgerufen am: 11.06.2017

[5] Xsens Technologies B.V. - MTi User Manual: MTi 10-series and MTi 100-series. Document MT0605P, Revision I

[6] Xsens Technologies B.V. - MTi 100-series Datasheet. https://www.xsens.com/download/pdf/documentation/mti-100/mti-100-series.pdf, abgerufen am: 12.07.2017

[7] Breymann, U.: Der C++-Programmierer. C++ lernen ; professionell anwenden ; Lösungen nutzen. München: Hanser 2015

[8] Wolf, J.: C++. Das umfassende Handbuch ; [das Lehr- und Nachschlagewerk zu C++ ; inkl. der neuerungen von C++11 ; Sprachgrundlagen, OOP, Multithreading u.v.m ; CD-ROM: Quellcode der Beispiele und Entwicklunsumgebungen]. Galileo Computing, Bd. 2021. Bonn: Galileo Press 2014

[9] CMake: Overview. https://cmake.org/overview/, abgerufen am: 24.05.2017

[10] Geirhos, M.: Entwurfsmuster. Das umfassende Handbuch. Rheinwerk Computing. Bonn: Rheinwerk-Verlag 2015

[11] Winner, H., Hakuli, S., Lotz, F. u. Singer, C. (Hrsg.): Handbuch Fahrerassistenzsysteme. Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort. ATZ/MTZ-Fachbuch. Wiesbaden: Springer Vieweg 2015

[12] Velodyne LiDAR, Inc. - User's Manual and Programming Guide VLP-16: Velodyne LiDAR Puck. Rev. A. http://velodynelidar.com/docs/manuals/63-9243%20Rev%20B%20User%20Manual%20and%20Programming%20Guide,VLP-16.pdf, abgerufen am: 11.06.2017

[13] Schreiner, R.: Computernetzwerke. Von den Grundlagen zur Funktion und Anwendung. München: Hanser 2016

[14] Stenger, J.: Technische Optik: Laser. Technische Universität Braunschweig. Institut für Hochfrequenztechnik. 2009. https://www.tu-braunschweig.de/Medien-DB/ihf/laser\_skript\_ws0809.pdf, abgerufen am: 21.07.2017

[15] Xsens Technologies B.V. - MT Low Level Communication Documentation: MTi 1-series/10-series/100-series. Document MT0101P, Revision X2. 2017

[16] Sommer, E.: Anlagensteuerung und Monitoring (v0.6). Fakultät Elektro- und Informationstechnik. University Of Applied Sciences - Munich. 2015

[17] Müther, H.: Physik. http://www.tphys.physik.uni-tuebingen.de/muether/physik1/tpart31.pdf, abgerufen am: 13.07.2017

[18] Bartelmann, M., Feuerbacher, B., Krüger, T., Lüst, D., Rebhan, A., Wipf, A., Modler, F. u. Kreh, M.: Theoretische Physik. Berlin: Springer Spektrum 2015

[19] Maruhn, J.: Mechanik 2. Gothe Universiät Frankfurt am Main. http://th.physik.uni-frankfurt.de/~maruhn/Mechanik2/SS25.pdf, abgerufen am: 19.07.2017

[20] Weber, W.: Industrieroboter. Methoden der Steuerung und Regelung. München: Fachbuchverlag Leipzig im Carl Hanser Verlag 2017

[21] Thomas Janz, W. S.: Stationäre Kreisfahrt und Lenkwinkelsprung. Laborversuch: Fahrdynamik. http://home.htw-berlin.de/~stedtnit/pdf/Laborversuch%20Fahrdynamik.pdf, abgerufen am: 19.07.2017