

# Fortentwicklung und Neugestaltung der Software des LMS4000 Demokoffers

## PROJEKTARBEIT

für die Prüfung zum

Bachelor of Science

des Studienganges Informatik / Informationstechnik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

**Lorenz Scherrer**

Abgabedatum 1. Oktober 2022

Bearbeitungszeitraum

12 Wochen

Matrikelnummer

4711

Kurs

ka-tinf21b3

Ausbildungsfirma

SICK

Waldkirch

Betreuer der Ausbildungsfirma

Titel Vorname Nachname

Gutachter der Studienakademie

Titel Vorname Nachname

## Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: »Fortentwicklung und Neugestaltung der Software des LMS4000 Demokoffers« selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

\_\_\_\_\_  
Ort      Datum

\_\_\_\_\_  
Unterschrift

C:/Users/loren/DHBW\_Karlsruhe/T1000/T1000/bericht.sty

## **Zusammenfassung**

Dieses L<sup>A</sup>T<sub>E</sub>X-Dokument kann als Vorlage für einen Praxis- oder Projektbericht, eine Studien- oder Bachelorarbeit dienen.

Zusammengestellt von Prof. Dr. Jürgen Vollmer <juergen.vollmer@dhbw-karlsruhe.de>  
<https://www.karlsruhe.dhbw.de>. Die jeweils aktuellste Version dieses L<sup>A</sup>T<sub>E</sub>X-Paketes ist immer auf der *FAQ-Seite* des Studiengangs Informatik zu finden: <https://www.karlsruhe.dhbw.de/inf/studienverlauf-organisatorisches.html> → *Formulare und Vorlagen*.

Stand \$Date: 2020/03/13 15:07:45 \$

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
1.1	Vorwort . . . . .	6
1.2	Einleitung . . . . .	6
1.3	SICK AG . . . . .	8
1.3.1	Geschichte . . . . .	8
1.3.2	Unternehmensstruktur . . . . .	8
1.4	Ausgangssituation . . . . .	9
1.5	Aufgabenstellung . . . . .	11
<b>2</b>	<b>Grundlagen</b>	<b>12</b>
2.1	Appstudio . . . . .	12
2.2	LUA . . . . .	14
2.3	LMS4000 . . . . .	15
2.4	Encoder . . . . .	15
2.5	Clean Coding . . . . .	15
<b>3</b>	<b>Umsetzung</b>	<b>16</b>
3.1	Kommunikation zwischen ScanDriver und DK App . . . . .	16
3.2	Neugestaltung der GUI . . . . .	16
3.3	Behebung des Überlauf des Encoders . . . . .	16
<b>4</b>	<b>Umsetzung</b>	<b>17</b>
4.1	Fazit und Ausblick . . . . .	17
4.2	Fazit . . . . .	17
4.3	Ausblick . . . . .	17
	<b>Anhang</b>	<b>18</b>
	<b>Index</b>	<b>18</b>

# Abbildungsverzeichnis

1.1	Hauptstandort der SICK AG in Waldkirch . . . . .	9
1.2	Demokoffer . . . . .	10

# Tabellenverzeichnis

1.1	Legende zur Abbildung 1.2 . . . . .	10
-----	-------------------------------------	----

# Liste der Algorithmen

# Formelverzeichnis



# Kapitel 1

## Einleitung

### 1.1 Vorwort

Dieser Bericht handelt von den Tätigkeiten während der Praxisphase vom 18.10.2021 bis zum 10.01.2022 in der Abteilung Distance Ranging BU83 bei der SICK AG in Waldkirch. Ziel war die Fortentwicklung und Neugestaltung der Software des LMS4000 Demokoffers.

### 1.2 Einleitung

Der LMS4000 ist ein 2D-LiDAR(Light Detection and Ranging)-Sensor der es ermöglicht präzise Vermessungen schnell bewegender, kleiner und großer Objekte unabhängig von ihrer, Farbe oder Oberflächenbeschaffenheit.

Dieser LiDAR-Sensor wird mithilfe des Demokoffers für den Vertrieb demonstriert. Der Koffer beinhaltet mit dem LMS4000 noch drei weitere Sensoren. Zwei von diesen Sensoren sind Metallsensoren, sogenannte Inits, sie werden dazu verwendet, die genaue Winkelposition des LMS4000 zu bestimmen. Der letzte Sensor ist ein Encoder, dieser bestimmt die Veränderung der Position des LMS4000. Mit diesen drei Sensoren kann die Winkelposition des LMS4000 bestimmt werden. Eine SIM1000 wird dazu genutzt, die Daten, die die vier Sensoren liefern, zu erfassen, auszuwerten und zu archivieren (Beschreibung des Demokoffers erweitern).

Mit dem Programm AppStudio ist es möglich, eine App für die SIM1000 zu schreiben. Diese App weist jedem 2D-Scan des LMS4000 eine Winkelposition zu. Somit macht die App es möglich 3D-Bilder mit 60Hz von den möglichen 600Hz des LMS4000 aufzunehmen für 10 Sekunden. Die

Bildrate schöpft das Potenzial des LMS4000 nicht aus.

(Um den LMS4000 besser im Vertrieb zu präsentieren wäre es günstig die Bildrate zu erhöhen). Um die Bildrate des Demokoffers zu erhöhen, sollen zwei Apps zusammen arbeiten. Der ScanDriver der von Chris Jorna und die Demokoffer App von Manuel Würzburger..[ANSMANN op. 1997; BASLER 2016; FUJII und FUKUCHI 2005; IERUSALIMSKY op. 2016; IERUSALIMSKY, FIGUEIREDO und CELES FILHO 2006; JUNG und BROWN 2007; KÜHNEL und ZWIRNER 2012; *Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere* o. D.; MARTIN 2013; YOUNG 2014]

## 1.3 SICK AG

### 1.3.1 Geschichte

Bereits 1946 wurde die heutige SICK AG von Erwin Sick in München gegründet. 1952 hatte das Unternehmen dann seinen ersten wirtschaftlichen Durchbruch mit der Vorstellung eines Unfallschutz-Lichtvorhangs auf der Internationalen Werkzeugmesse in Hannover. Der Erfolg stellte sich ein und die SICK AG ist heute einer der führenden Herstellern und Entwickler von Sensoren und Sensorlösungen für die Bereiche Fabrik-, Logistik- und Prozessautomation. 1956 zieht das damals 25 Mitarbeiter zählende Unternehmen von München an den heutigen Standort Waldkirch. Im Jahre 1988 verstirbt Erwin Sick und seine Frau Gisela Sick führt das Unternehmen als Hauptgesellschafterin fort. 1996 wird eine Umfirmierung des Unternehmens von der Erwin Sick GmbH in eine Aktiengesellschaft durchgeführt. Der Hauptteil der Aktien (95% Unternehmen nicht Börsen notiert und kann weiterhin nach Vorstellungen der Gründerfamilie geführt werden mit nachhaltiger Wachstums Orientierung. Heute umfasst die SICK AG über 50 Tochtergesellschaften und Beteiligungen sowie zahlreiche Vertretungen rund um den Globus. Im Geschäftsjahr 2021 erzielte das Unternehmen einen Umsatz von 1.964 Millionen Euro und zählt Weltweit über 10.000 Mitarbeiter.

### 1.3.2 Unternehmensstruktur

Wie jedes größere Unternehmen ist auch die SICK AG in verschiedene Bereiche eingeteilt. Neben einigen Zentralbereichen, Corporate Departments (CDs) und Corporate Units (CUs), gibt es neun unterschiedliche Global Business Centers (GBCs). Jedes GBC hat eine weltweite Geschäftsverantwortung für einen bestimmten Technologiebereich. Hier liegt ebenso die Verantwortung für die Entwicklung, Produktion und das Produktmanagement von Produkten und Serviceleistungen. Für jede Region gibt es eine zuständige Stelle, welche die Verantwortung für den Vertrieb und das Servicegeschäft übernimmt. Dieser Bereich ist in den Sales and Service Clusters (SSCs) zusammengefasst. Neben diesen Bereichen gibt es noch Global Industry Centers (GICs), welche unter anderem die Verantwortung für das International Key Account Management (IKAM) und für Branchen in der Fabrik-, Logistik- und Prozessautomation innehaben. (SICK AG 2020d) Die



Abbildung 1.1: Hauptstandort der SICK AG in Waldkirch

GBCs sind weiterhin in kleinere Einheiten, den sogenannten Business Units (BUs) aufgeteilt. Hier wird das Geschäftsfeld der GBC in verschiedene Produkt- und Dienstleistungskategorien unterteilt. Je nach Größe und Ausrichtung einer BU, kann es vorkommen, dass eine weitere Unterteilung erfolgt. Hier wird Beispielsweise in Fachdisziplinen wie Entwicklung und Marketing aufgeschlüsselt.

## 1.4 Ausgangssituation

Die Hardware, mit der in diesem Projekt gearbeitet wird, ist der Demokoffer des LMS4000. Dieser wird im Vertrieb genutzt, um die Funktionen des LMS4000 zu demonstrieren. Auf dem Demokoffer lief bis vor kurzem die App von Silas Geschwender. Diese App nutzt nicht das volle Potenzial des LMS4000. Weshalb Manuel Würzburger mithilfe des ScanDrivers von Chris Jona das Potenzial des LMS4000 auszureisen. Der Ersteller hat an dieser neuen App weitergearbeitet, um sie fertig zustellen.

An dem Koffer sind vier Sensoren, zwei Metall Sensoren, die, die genau Gradzahl des

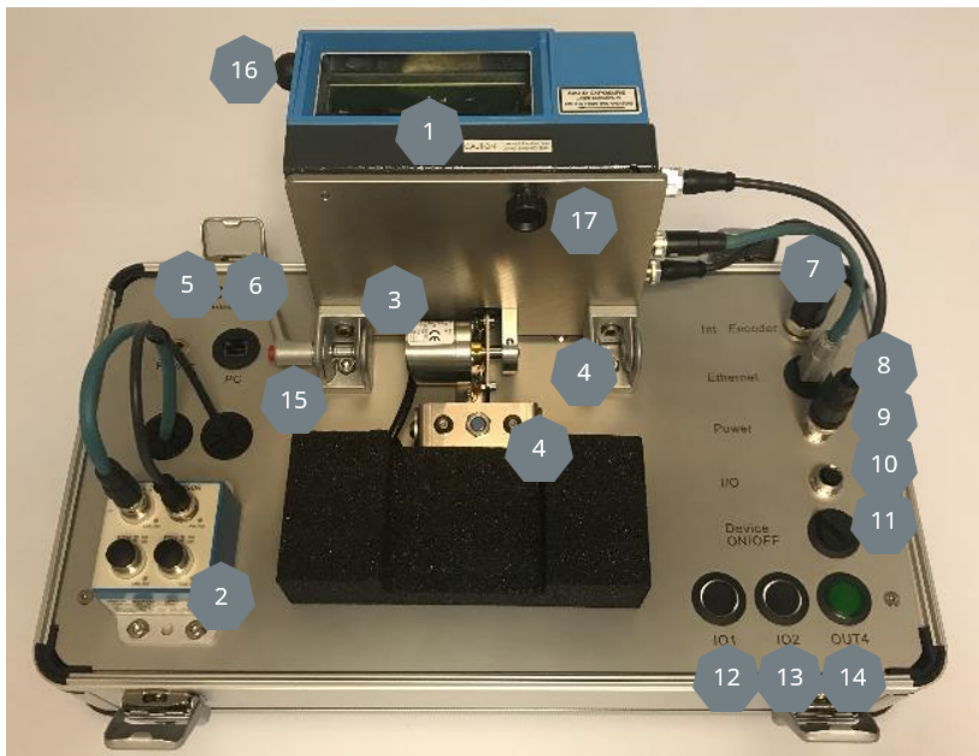


Abbildung 1.2: Demokoffer

- 1 LM4111
- 2 SIM1004
- 3 Incremental Encoder
- 4 Inductive Proximity Sensors
- 5 Power supply DC 24V
- 6 Ethernet PC
- 7 Internal Encoder
- 8 Ethernet LMS4111
- 9 Power supply LMS4111
- 10 I/Os LMS4111
- 11 Switch ON / OFF LMS4111
- 12 IO1 LMS4111
- 13 IO2 LMS4111
- 14 OUT4 LMS4111
- 15 Locking lever for different positions
- 16 Grab handle

Tabelle 1.1: Legende zur Abbildung 1.2

LMS4000 bestimmen, einmal  $0^\circ$  und um die  $90^\circ$  Position zu bestimmen. Der Encoder kann Rotation feststellen und damit die Veränderung der Gradzahl. Der LMS4000 nimmt ein 2D-Bild

auf, welches einer Gradzahl zugewiesen wird. Die SIM1004 verarbeitet diese Daten zu einer Punktwolke.

## 1.5 Aufgabenstellung

Das Hauptziel ist es, die App auf eine stabile Version für den Vertrieb zu bringen. Fehler und Störungen sollen korrigiert werden. Ein weiteres Ziel ist es zu erforschen, welche die höchst mögliche Frequenzzahl ist, mit welcher aufgenommen werden kann. Die ursprüngliche App von Silas Geschwender konnte mit 60Hz für 10 Sekunden aufnehmen. Es muss bestimmt werden, mit welcher Herzfrequenz aufgenommen werden kann und die SIM noch stabil läuft. Auch soll das Design der App maßgeblich verändert werden. Die App besteht aus zwei Seiten, auf der ersten Seite stehen die Daten der Komponenten, Status des Demokoffers und hier können auch neue Scans aufgenommen werden. Der aktuelle Scann ist dort auch zu sehen, die älteren Scans sind auch der zweiten Seite zu sehen, wo sie auch heruntergeladen oder gespeichert werden können. Die Seite zum Aufnehmen der Scans und die Seite zum Einsehen der Scans sollen zusammengefasst werden.

# Kapitel 2

## Grundlagen

### 2.1 Appstudio

SICK AppSpace - Engineering-Framework für Ihre individuellen Sensoranwendungen

SICK AppSpace ist ein Ökosystem rund um programmierbare Sensoren und Geräte von SICK und individualisierte SensorApps. Als Teil einer dynamischen Entwickler-Community können Kunden eigenständig oder gemeinsam mit den Experten von SICK SensorApps erstellen. Mit diesen SensorApps lassen sich alle Anwendungen mit unterschiedlichsten Technologien lösen. Individualisierte SensorApps werden mit intelligenten Softwaretools und Algorithmen erstellt. Bestehende Lösungen für Track Trace, Positionieraufgaben, Roboterführung oder Qualitätskontrolle können an die individuellen Bedürfnisse der Kunden angepasst werden; und völlig neue SensorApps können nach individuellen Anforderungen und absolut maßgeschneidert für bestehende Systeme erstellt werden. SICK AppSpace unterstützt heute eine Reihe von Geräten und Technologien, wie 2D-Vision, 3D-Vision, LiDAR, RFID oder Integrationsprodukte. Das SICK-AppSpace-Ökosystem besteht aus drei Hauptkomponenten. Zum einen aus der Hardware, das heißt programmierbare Sensoren, Sensorintegrationsmaschinen und andere Geräte. Zum anderen Software und Tools, das heißt die Werkzeuge zum Erstellen, Verteilen und Bereitstellen von SensorApps. Und als letzten Hauptkomponenten die Community, das heißt die Mitglieder des SICK AppSpace-Entwicklerclubs, die sich im SICK Support Portal und Konferenzen austauschen.

Die programmierbaren Geräte im SICK-AppSpace-Ökosystem bieten Raum für die Integration

der kundenspezifischen SensorApps und ermöglichen so maßgeschneiderte Applikationslösungen. So werden je nach Anwendung völlig neue Lösungen im Bereich der Automatisierung möglich - und die SensorApps können jederzeit angepasst oder ausgetauscht werden. SICK AppSpace-Sensoren und -Geräte bieten eine "Dual TalkFunktion, die eine gleichzeitige Kommunikation mit einer SPS sowie mit übergeordneten Enterprise-Level-Systemen und sogar Cloud-Services ermöglicht. Sie unterstützen damit den Aspekt der vertikalen Integration von Industrie 4.0.

Das AppEngine-Framework ist das Herzstück der Firmware bei allen programmierbaren Geräten. Es bietet eine umfangreiche Anwendungsprogrammierschnittstelle (SICK Algorithm API) mit einem breiten, gerätespezifischen Satz an vordefinierten Funktionen und Operatoren. Geräte mit Bildverarbeitung bieten zwei Methoden der Bildverarbeitung - entweder mit den 2D- und 3D-Operatoren der SICK Algorithm API oder mit der integrierten leistungsfähigen HALCON-Vision-Bibliothek.

Die zentralen Werkzeuge im SICK AppSpace-Ökosystem sind das SICK AppStudio, der SICK AppManager und der SICK AppPool.

Das SICK AppStudio ist eine integrierte Anwendungsentwicklungsumgebung zur Erstellung von SensorApps. Mit diesen SensorApps können Entwickler kundenspezifische Applikationslösungen auf programmierbaren Geräten erstellen. SensorApps können mit Standardkomponenten und Programmiersprachen programmiert werden. Anwendungsentwickler mit weniger Interesse am Programmieren können auch Datenflüsse modellieren, wobei ein tieferes Eintauchen in die Programmierung immer möglich ist. Die Benutzeroberflächen der SensorApps sind webbasiert, so dass sie in jedem Browser angezeigt werden können. Sie können vom SensorApp-Entwickler über einen grafischen UI-Builder oder mit Standard-Webtechnologien individuell gestaltet werden.

SensorApps werden mit dem Deployment-Tool SICK AppManager installiert, aktualisiert und verwaltet. Es integriert sich direkt in den SICK AppPool und unterstützt das automatische Deployment über ein CLI. Der vollständige SICK AppManager ist kostenlos und für Windows (x64) verfügbar, die reine CLI-Version ist auch für Windows-32bit-, Linux-64bit- und ARM-32bit-Systeme verfügbar.

Der SICK AppPool ist der zentrale und sichere Cloud-Service zum Austausch von SensorApps. SensorApp-Entwickler können ihre SensorApps veröffentlichen und entweder mit einer ausge-



wählten Gruppe von Nutzern oder mit der ganzen Welt teilen. Interessierte können SensorApps nach Stichworten, kompatiblen Geräten, Applikationen, Herausgebern und zahlreichen weiteren Filtern finden. Die Veröffentlichung im SICK AppPool ist ein Privileg der Mitglieder des SICK AppSpace Developers Club.

Um SensorApps zu entwickeln, ist eine Mitgliedschaft im SICK AppSpace Developers Club erforderlich. Diese Mitgliedschaft beinhaltet eine Volllizenz für das SICK AppStudio. Für die Dauer von einem Jahr haben Mitglieder außerdem Zugriff auf das Ticket-Supportsystem im SICK Support Portal, Entwicklerschulungen, exklusive Downloads und viele weitere Vorteile. Darüber hinaus werden sie zu den jährlichen SICK AppSpace-Entwicklerkonferenzen und anderen Veranstaltungen eingeladen, wo sie ihre Arbeit präsentieren, Ideen austauschen und die Entwicklung des SICK AppSpace-Ökosystems mitgestalten können.

## 2.2 LUA

Lua ist eine imperative und erweiterbare Skriptsprache zum Einbinden in Programme, um diese leichter weiterentwickeln und warten zu können. Eine der besonderen Eigenschaften von Lua ist die geringe Größe des kompilierten Skript-Interpreters.

Lua-Programme sind meist plattformunabhängig und werden vor der Ausführung in Bytecode übersetzt. Obwohl man mit Lua auch eigenständige Programme schreiben kann, ist sie vorrangig als eingebettete Skriptsprache für andere Programme konzipiert. Vorteile von Lua sind die geringe Größe von 120 kB, die Erweiterbarkeit und die hohe Geschwindigkeit, verglichen mit anderen Skriptsprachen.

Der Lua-Interpreter kann über eine C-Bibliothek angesprochen werden, die auch ein API<sup>1</sup> für die Laufzeitumgebung des Interpreters für Aufrufe vom C-Programm aus enthält. Mittels des API können verschiedene Teile des Programmes in C (oder C++) und Lua geschrieben werden, während Variablen und Funktionen in beiden Richtungen erreichbar bleiben (d. h. eine Funktion in Lua kann eine Funktion in C/C++ aufrufen und umgekehrt).

---

<sup>1</sup>von englisch application programming interface, wörtlich „Anwendungsprogrammierschnittstelle

## **2.3 LMS4000**

## **2.4 Encoder**

## **2.5 Clean Coding**

## Kapitel 3

# Umsetzung

3.1 Kommunikation zwischen ScanDriver und DK App

3.2 Neugestaltung der GUI

3.3 Behebung des Überlauf des Encoders

## Kapitel 4

# Umsetzung

### 4.1 Fazit und Ausblick

### 4.2 Fazit

### 4.3 Ausblick

# Literatur

- ANSMANN, Albert [op. 1997]. *Advances in atmospheric remote sensing with lidar: Selected papers of the 18th International Laser Radar Conference (ILRC), Berlin, 22-26 July 1996*. Berlin [etc.]: Springer. ISBN: 3-540-61887-2 [siehe S. 7].
- BASLER, Stefan [2016]. *Encoder und Motor-Feedback-Systeme: Winkellage- und Drehzahlerfassung in der industriellen Automation*. Wiesbaden: Springer Vieweg. ISBN: 9783658128449 [siehe S. 7].
- FUJII, Takashi und Tetsuo FUKUCHI [2005]. *Laser remote sensing*. Boca Raton [etc.]: CRC Press. ISBN: 0-8247-4256-7 [siehe S. 7].
- IERUSALIMSKY, Roberto [op. 2016]. *Programming in Lua*. 4th ed. Rio de Janeiro: Lua.org. ISBN: 978-8-59037-986-7 [siehe S. 7].
- IERUSALIMSKY, Roberto, Luiz Henrique de FIGUEIREDO und Waldemar CELES FILHO [2006]. *Lua 5.1 reference manual*. Rio de Janeiro: Lua.org. ISBN: 85-903798-3-3 [siehe S. 7].
- JUNG, Kurt und Aaron BROWN [2007]. *Beginning Lua Programming (Wrox beginning guides)*. Wrox. ISBN: 0-470-06917-1 [siehe S. 7].
- KÜHNEL, Claus und Daniel ZWIRNER [2012]. *Lua: Einsatz von Lua in Embedded Systems*. 2., bearb. und erw. Aufl. Altendorf: Skript Verl. Kühnel. ISBN: 978-3-90785-715-1 [siehe S. 7].
- Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere* [o.D.] Scholars Portal. ISBN: 0-387-40075-3 [siehe S. 7].
- MARTIN, Robert C. [2013]. *Clean Code - Refactoring, Patterns, Testen und Techniken für sauberen Code: Deutsche Ausgabe*. Frechen: mitp. ISBN: 9783826696381 [siehe S. 7].

YOUNG, David [2014]. *Learning game AI programming with Lua: Leverage the power of Lua programming to create game AI that focuses on motion, animation, and tactics*. Community experience distilled. Birmingham, UK: Packt Publishing. ISBN: 1-783-28133-2 [siehe S. 7].