

Von C# nach Python:  
Software-Konzeptionierung einer  
robotergestützten Lagerverwaltung  
Analyse bestehender Software und  
Konzeptionierung einer integrierten  
Python-Anwendung mit  
kameragestützten  
Validierungsprozessen in der Industrie  
4.0-Plattform Modellfabrik  $\mu$ Plant

Lennart Schink



Matrikelnummer:

33237484

Gutachter:

Univ.-Prof. Dr.-Ing. Andreas Kroll

Betreuer:

Dip.-Ing. Axel Dürrbaum

Tag der Abgabe:

8. Juni 2023

MRT-Nr.:

N.N



# Inhaltsverzeichnis

---

Abkürzungsverzeichnis	XI
Symbolverzeichnis	XIII
Index	XV
1. Motivation und Zielsetzung	1
2. Softwarearchitektur der bestehenden Software	3
3. Weiteres Kapitel	5
3.1. Unterkapitel . . . . .	5
3.1.1. Abschnitt . . . . .	5
4. Tips & Tricks	7
4.1. Makros . . . . .	7
4.2. Tabellen . . . . .	7
4.3. Symbolverzeichnis . . . . .	8
4.4. Abkürzungsverzeichnis . . . . .	8
4.5. Index . . . . .	9
4.6. Nützliche Umgebungen . . . . .	9
4.6.1. Programm-Quellcode . . . . .	9
4.7. Mathematische Umgebungen . . . . .	10
5. Zusammenfassung und Ausblick	11
A. Dies ist der erste Anhang	XVII
Literaturverzeichnis	XIX



# Tabellenverzeichnis

---

4.1. Neuer Spaltentyp . . . . .	7
4.2. Neue Spaltentypen . . . . .	8



# Abbildungsverzeichnis

---





# Listings

---

Listings/Demo1.py . . . . .	9
Listings/Demo2.py . . . . .	9







# Symbolverzeichnis

---

Symbol	Bedeutung	Einheit	Bemerkung
$\pi$	Kreiszahl		
$c$	Lichtgeschwindigkeit	$\text{m/s}^2$	im Vakuum
$g$	Erdbeschleunigung		
$g$	Erdbeschleunigung	$\text{m/s}^2$	



# Index

---

listings, [9](#)

Abkürzungsverzeichnis, [8](#)

Glossar, [8](#)

Index, [9](#)

Makros, [7](#)

Symbolverzeichnis, [8](#)

Tabellen, [7](#)

Vektor, [10](#)





# 1 Motivation und Zielsetzung

---

Das Institut für Mess- und Regelungstechnik an der Universität Kassel hat in den letzten Jahren eine Modellfabrik  $\mu$ Plant gebaut. Aus über 70 Einzelarbeiten ist ein modernes Industrie-4.0 Konzept geschaffen worden. Teil der  $\mu$ Plant ist ein vollautomatisiertes Lager. Das Lager besteht aus einem abgetrennten Raum, dessen Zugang über eine Tür mit einem Türschalter überwacht ist. In diesen Bereich können Turtlebots einfahren. In dem abgetrennten Bereich steht ein Industrieroboter und ein Lagerregal mit ausgewiesenen 18 Lagerplätzen. Außerdem befindet sich neben einer Andockstation für den Turtlebot auch noch eine Werkbank.

Ein pneumatischer Greifer des Industrieroboters kann Paletten, die je mit bis zu zwei Bechern bestückt werden können, zwischen dem mobilen Roboter und dem Lagerregal frei bewegen. Von einem PC-Arbeitsplatz aus können mittels Software die Lagerprozesse überwacht werden. Außerdem kann im Fehlerfall eingeschritten werden und es können manuell Prozesse ausgelöst werden.

Die Software ist derzeit in 3 Teile aufgeteilt: Einerseits gibt es die Lagerverwaltung - die Hauptsoftware. Sie bildet die automatisierten Prozesse ab und verfügt über ein GUI welches u.A. den Bestand visualisiert. Daneben gibt es den Warehouse Controller, der dazu verwendet wird manuell Lagerprozesse auszulösen, und ein RFID-Tool was für manuelle RFID - Prozesse benutzt wird.

Mit dem Wechsel des Betriebssystems von Windows 7 auf Windows 10 ist die Kompatibilität der in C# implementierten Software nicht mehr gegeben. Außerdem laufen Teilfunktionen des Programms nicht fehlerfrei oder tolerieren kaum Fehlbedienungen. Die Dreiteilung der Software ist im Allgemeinen auch nicht mehr erwünscht.

Diese Seminararbeit beschäftigt sich mit der Analyse der bestehenden Software: Es wird ermittelt, aus welchen Programmteilen und Funktionen die Software besteht. Aus den Erkenntnissen wird ein Konzept entwickelt wie die Drei Software Teile zusammengeführt werden könnten um so die Grundlage für eine Migration der Software nach Python zu schaffen.

Erkenntnisse aus der studentische Arbeit von [Hügler] sollen überprüft und vertieft werden um Anforderungen an Kameras und arUco Marker zu ermitteln, die später eine automatisierte Inventur ermöglichen sollen.



# 2 Softwarearchitektur der bestehenden Software

---

Analysemethoden der Informatik für Software sind in der Regel für die verschiedenen Design-Phasen einer Software entwickelt worden. Eine von mir durchgeführte Recherche ergab, dass sich Analyse-Tools und Methoden für bestehende Software vor Allem darauf fokussieren die Performance, Speichermanagement und Benutzererfahrung zu bewerten. Die Architektur einer Software spielt dabei eine untergeordnete Rolle. Für die Neuentwicklung der bestehenden Software werden im Folgenden die Klassen und Objekte sowie ihre Wechselwirkungen dargestellt und anschließend analysiert und bewertet.

In einem C# - Projekt sind UI und Logik in getrennten Dateien implementiert. XAML-Dateien sind eine angepasste Form von XML-Dateien. Sie legen fest wie etwas gerendert wird während XAML.CS- Dateien die dahinterliegende Business-Logik abbilden. Am Beispiel des Haupt-GUI des Programms soll der prinzipielle Aufbau gezeigt werden. Wegen der Komplexität wird der Umfang jedoch nicht für das gesamte Programm fortgesetzt.



# 3 Weiteres Kapitel

---

## 3.1. Unterkapitel

### 3.1.1. Abschnitt

#### Unterabschnitt

Die Unterabschnittsebene sollte nicht mehr im Inhaltsverzeichnis auftauchen.



# 4 Tips & Tricks

---

Do's and Dont's in  $\text{\LaTeX}$ : <https://mast.queensu.ca/~andrew/LaTeX/latex-dos-and-donts.pdf>

## 4.1. Makros

$\text{\TeX}$ -Makros def oder  $\text{\LaTeX}$ -Makros newcommand definieren für einheitliche Schreibweisen im Dokument:

```
\newcommand{\Matlab}{\textsc{Matlab}\xspace}
\def\squared#1{\ensuremath{\#1^2}}
\newcommand{\Makro}[3]{Argument #1 mit Argument#2 und Argument #3}
```

Superskript mitten im Text ohne  $\backslash\$\$$  ->  $\backslash\squared{\pi}$

Mit  $\backslash\text{\Matlab}$  kann man es berechnen!

Superskript mitten im Text ohne  $\mathbb{\$}$  ->  $\pi^2$

Mit  $\text{\texttt{MATLAB}}$  kann man es berechnen!

## 4.2. Tabellen

Eigene Formate für Spalten in Tabellen mit dem Paket tabularx definieren:

```
\newcolumntype{R}{>\raggedleft\arraybackslash}X}
```

**Tabelle 4.1.:** Neuer Spaltentyp

label 1	label 2	label 3	label 4
item 1	item 2	item 3	item 4

Zahlen ausrichten in Spalten mit Paket dcolumn (und gleichzeitig ersetzen des Dezimaltrenners "." durch ein Komma ",")

```
\newcolumntype{d}[1]{D{.}{,}{\#1} }
```

Hilfe: <https://en.wikibooks.org/wiki/LaTeX/Tables>

---

Symbol	Wert
$\pi$	3,1415
-e	-2,71

**Tabelle 4.2.:** Neue Spaltentypen

### 4.3. Symbolverzeichnis

Ein Symbolverzeichnis kann automatisch mit dem Paket `nomencl` angelegt werden:

Mit dem Befehl `\Symbol{symbol}{bedeutung}` wird das Symbol im Text angegeben und gleichzeitig in das Symbolverzeichnis aufgenommen:

Die Zahl `\Symbol{$\pi$}{Kreiszahl}` ist transzendent.

Die Zahl  $\pi$  ist transzendent.

Es können auch Einheit und Kommentar angegeben werden:

Symbol mit Text: Die  
`\SymbolT{$g$}{Erdbeschleunigung}`  
 wirkt immer.

Symbol mit Text und Einheit: Die  
`\SymbolU{$g$}{Erdbeschleunigung}{m/s^2}`  
 wirkt immer.

Mit Text, Einheit und Bemerkung: Die  
`\SymbolB{$c$}{Lichtgeschwindigkeit}{m/s^2}{im Vakuum}`  
 ist maximal.

Symbol mit Text: Die Erdbeschleunigung  $g$  wirkt immer.

Symbol mit Text und Einheit: Die Erdbeschleunigung  $g$  wirkt immer.

Mit Text, Einheit und Bemerkung: Die Lichtgeschwindigkeit  $c$  ist maximal.

Das Symbolverzeichnis muss mit dem Befehl `makeindex` manuell erstellt werden:

```
latex MRT-Bericht.tex
makeindex MRT-Bericht.nlo -s nomencl.ist -o MRT-Bericht.nls
latex MRT-Bericht.tex
```

Hilfe: <https://www.ctan.org/pkg/nomencl?lang=de>

### 4.4. Abkürzungsverzeichnis

Ein Abkürzungsverzeichnis wird in  $\LaTeX$  als Glossar mit dem Befehl

`\Glossar{abkuerzung}{kurz}{lang}` erstellt:



---

Hier ein Eintrag für das AbkVz:

```
\Glossar{svm}{SVM}{Support Vector Machine}
```

Hier ein Eintrag für das AbkVz: Support Vector Machine (SVM)

Hilfe: <https://en.wikibooks.org/wiki/LaTeX/Glossary>

## 4.5. Index

L<sup>A</sup>T<sub>E</sub>X erstellt automatisch einen Index mit dem Befehl `\Index{eintrag}` angelegt:

L<sup>A</sup>T<sub>E</sub>X erstellt automatisch einen Index-Eintrag mit dem Befehl  
“`\Index{Index}`”.

L<sup>A</sup>T<sub>E</sub>X erstellt automatisch einen Index-Eintrag mit dem Befehl “Index”.

Hilfe: <https://en.wikibooks.org/wiki/LaTeX/Indexing>

## 4.6. Nützliche Umgebungen

### 4.6.1. Programm-Quellcode

Listings können mit dem Paket listings in ein Dokument aufgenommen werden.

```
a = 1                                     1
b = 2                                     2
c = sqrt( a^2 + b^2 )                     3
```

Man kann auch Befehle wie `var i = 0;` direkt im Text angeben!

Dateien automatisch in das Dokument einfügen:

```
\lstinputlisting[language=Python]{Demo1.py}
```

Demo 1 als Python-Quelltext :

```
# This program prints Hello , world!

print( 'Hello , world!' )
```

Demo 2 mit Zeilennummern:

```
1 # This program adds two numbers
2
3 num1 = 1.5
4 num2 = 6.3
5
6 # Add two numbers
7 sum = float(num1) + float(num2)
```

---

```

8
9 # Display the sum
10 print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))

```

Hilfe: [https://en.wikibooks.org/wiki/LaTeX/Source\\_Code\\_Listings](https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings)

## 4.7. Mathematische Umgebungen

Schreibweise von Vektoren/Matrizen vereinfachen:

Vektor

$$\mathbf{v}_s^T = \begin{pmatrix} 11, 12, 13 \end{pmatrix} \quad (4.1)$$

$$\mathbf{v}_z = \begin{pmatrix} 11 \\ 21 \\ 31 \end{pmatrix} \quad (4.2)$$

$g$  ist  $9.8055$  in  $\text{m/s}^2$

Die Umgebungen des Paketes `amsmath` verwenden weniger vertikalen Platz und können ausgerichtet werden

```

\begin{align}\label{eq:1}
  a &= 1\\
  xyz &= \mathrm{i}
\end{align}

```

```

\begin{alignat*}{2}
  \dot{x} &= A \cdot x + B \cdot u \\
  y &= C \cdot x + D \cdot u
\end{alignat*}

```

$$\dot{x} = A(t) \cdot x + B \cdot u \quad (4.3)$$

$$y = C \cdot x + D \cdot u \quad (4.4)$$

$$\dot{\mathbf{x}} = \mathbf{A}(t) \cdot \mathbf{x} + \mathbf{B} \cdot u$$

$$y = \mathbf{C} \cdot \mathbf{x} + D \cdot u$$

# 5 Zusammenfassung und Ausblick

---

Hier wird die Arbeit zusammengefasst und ein Ausblick auf offene Fragestellungen gegeben.



# **A** Dies ist der erste Anhang

---

Hier Text einfügen.



## Literaturverzeichnis

---