

Requirements and Design Documentation

(RDD)

Version 0.5

ESEP - Praktikum - Wintersemester 2016/2017

Lüdemann	Mona	2212744	mona.luedemann1@haw-hamburg.de
Butkerei	Marvin	2247550	marvin.butkerei@haw-hamburg.de
Schumacher	Wilhelm	2245216	wilhelm.schumacher@haw-hamburg.de
Melkonyan	Anushavan	2243668	anushavan.melkonyan@haw-hamburg.de
Colbow	Marco	2177095	marco.colbow@haw-hamburg.de
Cakir	Mehmet	2195657	mehmet.cakir@haw-hamburg.de

8. Januar 2017

Änderungshistorie:

Version	Author	Datum	Anmerkungen/Änderungen
0.1	Mehmet Cakir	2016-10-18	Kapitel 1-4 und Testkonzept
0.2	Mehmet Cakir	2016-10-26	Korrekturen an Formulierung, Visualisierungen noch nicht festgelegt.
0.3	Mehmet Cakir	2016-11-03	Testtabellen umformatiert. Tests zu Grundfunktionen, HAL-UML, Systemgrenzen, Systemarchitektur und Visualisierungsentscheidung sowie entsprechend kurzen Text hinzugefügt.
0.4	Mehmet Cakir	2016-11-16	Neugliederung der Kapitel 4 und 7, Systemkontexte zusammengeführt, verwendete Werkzeuge ergänzt, Zeitmessung und FSM/HSM eingepflegt, Abbildung 6 zur Zeiterfassung aktualisiert, diverse Umformulierungen.
0.5	Mehmet Cakir	2016-11-17	Aktualisiertes UML-Klassendiagramm der HAL und Tests der Sensorik eingefügt.

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Einleitung	3
2 Teamorganisation	3
2.1 Verantwortlichkeiten	3
2.2 Absprachen	3
2.3 Repository-Konzept	4
3 Projektmanagement	4
3.1 Prozess	4
3.2 PSP/Zeitplan/Tracking	4
3.3 Qualitätssicherung	5
4 Randbedingungen	5
4.1 Entwicklungsumgebung	5
4.2 Werkzeuge	5
4.3 Sprachen	5
5 Requirements and Use Cases	6
5.1 Stakeholder	6
5.2 Anforderungen	7

1 Einleitung

Diese Dokumentation beschreibt für dieses Projekt im Rahmen des ESEP Praktikum im Wintersemester 2016/2017 sämtliche Beschlüsse, Schritte und Maßnahmen die während des Projekt- bzw. Entwicklungszeitraums getroffen wurden. Das Projekt umfasst die Implementierung von Software zur Ansteuerung von drei baugleichen Förderbändern des Unternehmens Festo, womit eine Werkstück-Sortieranlage realisiert werden soll. Die Software stellt die Kommunikation der drei Förderbänder über serielle Schnittstellen sicher.

2 Teamorganisation

Grundsätzlich kann jedes Teammitglied eine Aufgabe seiner Wahl übernehmen. Bei jedem Meeting werden die Aufgaben verteilt, worüber im folgenden Meeting über den Fortschritt diskutiert wird. Falls ein Mitglied seine Aufgabe fertiggestellt hat, übernimmt er eine Neue. Bei Nichteinhaltung des Zeitplans werden entsprechend der Zeitpuffer andere Aufgaben zurückgestellt. Die Aufgaben richten sich nach den zu bewältigenden Milestones(siehe [?]) zum jeweiligen Praktikumstermin. Für die Projektleitung und die Pflege des RDD-Dokuments wurde jeweils eine Person bestimmt, welche im Unterkapitel 2.1 eingesehen werden kann.

2.1 Verantwortlichkeiten

Aufgabe	Zuständige/r	Bemerkung
Projektleitung	Mona	Die Projektleitung überwacht den Projektfortschritt und benachrichtigt insbesondere bei Nichteinhalten des Zeitplans alle Teammitglieder. Außerdem hat die Projektleitung bei Unstimmigkeiten immer das letzte Wort.
RDD-Pflege	Mehmet	Der Zuständige ist für die Gestaltung und für die Vollständigkeit des RDDs verantwortlich. Er kann andere Gruppenmitglieder dazu auffordern Inhalte für das Dokument zu erarbeiten und ihm bereit zu stellen.
Protokollführung	Alle Teammitglieder	Die Protokollführung wird reihum von Gruppenmitgliedern übernommen. Dabei wird folgende Reihenfolge eingehalten: <i>Mona</i> → <i>Marvin</i> → <i>Marco</i> → <i>Wilhelm</i> → <i>Mehmet</i> → <i>Anushavan</i>

Tabelle 1: Zuteilung von Verantwortlichkeiten

2.2 Absprachen

Zur Kommunikation außerhalb der Praktikumstermine werden die Messengerdienste Slack und WhatsApp verwendet. Unstimmigkeiten, Fragen und Inkenntnisetzungen können somit interaktiv geklärt bzw. mitgeteilt werden. Es wird erwartet, dass jedes Teammitglied in einem Zeitfenster von 24 Stunden auf eine Nachricht entsprechend mit einer Nachricht antwortet. In folgender Abbildung 1 werden die Termine der Meetings dargestellt:

Terminplan für Meetings				
Oktober	Mi, 05.10.	Do, 13.10.	Mi, 19.10.	Mi, 26.10.
	ab 16:00 Uhr	ab 12:00 Uhr	ab 16:00 Uhr	ab 16:00 Uhr
November	Do, 03.11.	Do, 10.11.	Mi, 16.11.	Mi, 23.11.
	ab 12:00 Uhr	ab 12:00 Uhr	ab 16:00 Uhr	ab 16:00 Uhr
Dezember	Do, 01.12.	Mi, 07.12.	Mi, 14.12.	Do, 22.12.
	ab 12:00 Uhr	ab 16:00 Uhr	ab 16:00 Uhr	ab 12:00 Uhr
Weitere Termine können/müssen je nach Bedarf in der Gruppe vereinbart werden.				

Abbildung 1: Terminplan der Meetings

2.3 Repository-Konzept

Das Projekt wird mit dem Versionskontrollsystem Git verwaltet. Zentral wurde ein Repository auf GitHub angelegt. Erreichbar ist das Repository unter <https://github.com/mbutkerei/conveyor>. Änderungen werden lokal auf einem Branch vorgenommen, jedoch nicht auf dem Master. Sind die Änderungen erfolgreich abgeschlossen, kann der Master mit dem lokalen Branch zusammengeführt werden. Bevor ein push durchgeführt wird, muss gepullt werden. Nachdem ggf. Mergekonflikte gelöst wurden, kann vom Masterbranch aus auf das Repository gepusht werden.

3 Projektmanagement

Für die Gewährleistung einer guten Teamarbeit, werden in den folgenden Kapiteln erklärt wie die Teammitglieder mit ihren Aufgaben umgehen bzw. wann eine gegenseitige Benachrichtigung über ihren Fortschritt spätestens stattfinden sollte.

3.1 Prozess

Das Projekt wird auf Grundlage der vorgeschlagenen Milestones umgesetzt. Für jede Implementierung ist zuvor ein geeignetes, sowie selbsterklärendes bzw. verständliches, aber auch möglichst vollständiges Diagramm anzufertigen. Die Visualisierung sollte vor der Implementierung allen anderen Teammitgliedern vorgestellt werden, um mögliche Verbesserungen einzuholen und ggf. Konflikte früh zu erkennen, sowie sie zu lösen. In der Tabelle 2 sind für die jeweiligen Spezifikationen die festgelegte Modellierung aufgelistet.

Spezifikation	Modellierung
Klassen	UML Diagramm
Verhalten bzw. logische Abläufe	Zustandsautomat
Systemarchitektur	Komponentendiagramm

Tabelle 2: Festgelegte Modellierung zur jeweiligen Spezifikation

3.2 PSP/Zeitplan/Tracking

Zu jedem Praktikumstermin wird erwartet, dass die verteilten Aufgaben bzw. Milestones erfüllt werden. Um dies zu gewährleisten, muss jedes Teammitglied bei Schwierigkeiten die Projektleitung darüber sofort in Kenntnis setzen, damit frühzeitig ausgeholfen werden kann. Dazu wurden Arbeitspakete definiert und als Milestones in einem Gantt-Diagramm festgehalten.

3.3 Qualitätssicherung

Hinsichtlich der Qualitätssicherung, werden die vier Punkte Team, Modellierung, Code und Förderband herangezogen.

1. **Team:** Jedes Teammitglied sollte über seine eigenen Fähigkeiten im Klaren sein und möglichst nur Aufgaben übernehmen, wofür es sich am besten geeignet fühlt. Darüber hinaus muss jedes Teammitglied bei Möglichkeit stets seine Unterstützung anbieten. Bei Problemen oder Überforderung müssen alle anderen Teammitglieder darüber unterrichtet und Aufgaben ggf. neu verteilt werden.
2. **Modellierung:** Vor der Implementierung muss eine geeignete Visualisierung erstellt, anderen Teammitgliedern vorgestellt und diskutiert werden.
3. **Code:** Für den Code werden bekannte Pattern eingesetzt und verständliche sowie übersichtliche Realisierungen angestrebt. Den Maßstab hierfür setzen die Teammitglieder. Treten beim Code Review keine schwerwiegenden Anmerkungen bzw. Verständnisprobleme auf, gilt der Code als verständlich und übersichtlich.
4. **Förderband:** Um hohen Durchsatz sowie Effizienz bei der Aussortierung zu erzielen, werden die Komponenten mit der höchstmöglichen Geschwindigkeit für die jeweilige Situation angetrieben, während die Sicherheit des Bedieners im Vordergrund steht. Dabei werden Fehler- bzw. Ausnahmezustände ggf. durch einfache Signalcodes mithilfe der Ampel dem Bediener mitgeteilt.

4 Randbedingungen

In diesem Kapitel werden die Bedingungen genannt unter denen das Projekt umgesetzt wird und die Mittel, die für die Umsetzung herangezogen werden.

4.1 Entwicklungsumgebung

Die drei Förderbänder werden über drei QNX Systeme gesteuert, die über eine serielle Schnittstelle verbunden sind. Als IDE wird QNX Momentics auf Windows 7 verwendet.

4.2 Werkzeuge

- QNX Momentics IDE 5.0
- Latex(MiKTeX 2.9, Texmaker 4.5)
- Git 2.8.1
- Visual Paradigm 13.2
- Gantt Project 2.8.1
- Microsoft Visio 2016

4.3 Sprachen

Das System wird im C++03 Standard programmiert. Dabei werden vorgegebene Bibliotheken verwendet, welche in folgender Tabelle 3 aufgelistet sind:

Name	Version	Autor
HWaccess.h	Unknown	Prof. Dr. Stephan Pareigis
HAWThread.h	Unknown	Prof. Dr. Stephan Pareigis
Lock.h	0.1	Simon Brummer

Tabelle 3: Verwendete Programmierbibliotheken

5 Requirements and Use Cases

Mithilfe der Requirements werden die Anforderungen an die einzelnen Komponenten des Förderbandes ermittelt. Dabei werden die Interessen der Stakeholder berücksichtigt.

5.1 Stakeholder

Stakeholder	Interessen
Kunde	<ul style="list-style-type: none">- fehlerfreie Umsetzung der Anforderungen- erfolgreiche Beendigung des Projektes
Designer	<ul style="list-style-type: none">- übersichtliches, leicht erweiterbares Design- sorgfältige Dokumentation
Entwickler	<ul style="list-style-type: none">- präzises Design- sinnvolle Kommentare- lesbarer Code
Tester	<ul style="list-style-type: none">- übersichtliches, vollständiges Testkonzept
Bediener (Mitarbeiter, die das Laufband später bedienen sollen)	<ul style="list-style-type: none">- einfache und intuitive Bedienung
Instandhalter	<ul style="list-style-type: none">- robustes System
Andere Mitarbeiter	<ul style="list-style-type: none">- Kenntnis über System und Funktionsweise

Tabelle 4: Stakeholder und ihre Interessen

5.2 Anforderungen

Titel	Beschreibung
Ansteuerung der Ampeln	Die Software soll die Ampeln aller Förderbänder für folgende Fälle entsprechend ansteuern können: <ul style="list-style-type: none"> - grünes Licht bei Normalbetrieb, fehlerfrei - gelbes Licht bei Warnungen - rotes Licht bei Fehler
Ansteuerung der Motoren	Die Motoren der Förderbänder sollen in folgenden Varianten ansteuerbar sein: <ul style="list-style-type: none"> - Rechtslauf langsam/schnell - Linkslauf langsam/schnell - Stopp
Ansteuerung der Weichen	Die Stellungen „offen“ und „geschlossen“ der Weichen müssen angesteuert werden. Außerdem soll beachtet werden, dass die Weichen nur für kurze Zeit die Stellung „offen“ halten, um eine Beschädigung der Weichen zu vermeiden.
Erkennung von Werkstücken	Das erste und zweite Förderband müssen drei Arten von Werkstücken erkennen können: <ul style="list-style-type: none"> - Flache Werkstücke - Werkstücke mit Metalleinsatz (Bohrung liegt nach oben oder unten) - Werkstücke ohne Metalleinsatz (Bohrung liegt nach oben oder unten)
Aussortierung von Werkstücken	Flache Werkstücke und Werkstücke, bei der die Bohrung nach unten liegt, sollen auf dem ersten und zweiten Förderband aussortiert werden.
Reihenfolge der Werkstücke	Am Ende vom zweiten Förderband sollen die Werkstücke vereinzelt in folgender Reihenfolge ankommen: Bohrung oben ohne Metall → Bohrung oben ohne Metall → Bohrung oben mit Metall

Tabelle 5: Anforderungen(Teil 1)

Titel	Beschreibung
Erkennung von Überschlagen der Werkstücke + Aussortierung des betreffenden Werkstücks	Das zweite Förderband muss eine erneute Prüfung des aktuellen Werkstücks durchführen, um es im Falle eines Überschlagens auszusortieren.
Langsamer Transport bei Höhenmessung	Wenn ein Werkstück durch die Höhenmessung transportiert wird, soll das Förderband langsam laufen.
Konsolenausgabe am Ende vom zweiten Förderband	Wenn ein Werkstück das Ende vom zweiten Förderband erreicht, sollen auf der Konsole vom zweiten Förderband folgende Werkstückdaten ausgegeben werden: <ul style="list-style-type: none"> - ID - Typ - Ermittelter Höhen-Messwert vom ersten Förderband - Ermittelter Höhen-Messwert vom zweiten Förderband
Konsolenausgabe am Ende vom dritten Förderband	Am Ende des dritten Förderbandes sollen die Werkstückdaten ankommender Werkstücke auf der Konsole des dritten Förderbandes ausgegeben werden.
Stopp der Förderbänder bei keinen Werkstücken	Alle drei Förderbänder sollen jeweils stoppen, wenn sich kein Werkstück auf ihnen befindet.
Erkennung voller Rutschen	Volle Rutschen müssen mithilfe des Sensors am Rutscheneingang erkannt werden.
Rutschen koordinieren	Ist die Rutsche vom ersten Förderband voll, so soll die Aussortierung über das zweite Förderband erfolgen. Umgekehrt, ist die Rutsche vom zweiten Förderband voll, so soll die Aussortierung bereits auf dem ersten Förderband erfolgen. Da das dritte Förderband nur Werkstücke bündelt und weiterleitet, entfällt das Berücksichtigen der Rutsche des dritten Förderbandes.
Gebündelter Transport von Werkstückgruppen auf drittem Förderband	Die drei sortierten Werkstücke sollen gebündelt (im Abstand von 1,5cm) an das Ende des dritten Förderbandes transportiert werden.
Fehlererfassung: Verschwinden von Werkstücken + Reaktion	Mittels Zeitmessung soll das Verschwinden von Werkstücken erfasst werden. Wenn an einer nachfolgend benachbarten Lichtschranke kein Werkstück erfasst wird und dabei zuviel Zeit vergeht, tritt folgende Reaktion auf: Bandstopp, Fehlermeldung.

Tabelle 6: Anforderungen(Teil 2)

Titel	Beschreibung
Fehlererfassung: Hinzufügen von Werkstücken + Reaktion	Mittels Zeitmessung soll das zu schnelle oder fehlerhafte Hinzufügen von Werkstücken erfasst werden. Wenn zwischen zwei benachbarten Lichtschranken die erwartete Zeit unterschritten wird, in der ein Werkstück erfasst werden müsste, dann tritt folgende Reaktion auf: Bandstopp, Fehlermeldung
Fehlererfassung: Beide Rutschen voll + Reaktion	Es soll erkannt werden, wenn beide Rutschen vom ersten und zweiten Förderband voll sind. Reaktion: Bandstopp, Fehlermeldung

Tabelle 7: Anforderungen(Teil 3)