Requirements and Design Documentation

(RDD)

Version 1.8

SE2P – Praktikum – SS 2013

Matthiessen, Erik, 2024025, erik.matthiessen@haw-hamburg.de

Güngör, Nilüfer, 2007833, niluefer.guengoer@haw-hamburg.de

Ahmad, Maschhood, 1979158,maschhood.ahmad@haw-hamburg.de

Rycka, Denis, 2038475, denis.rycka@haw-hamburg.de

Änderungshistorie:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Datum** | **Anmerkungen** |
| 1.1 | Erik Matthiessen | 04.04.2013 | ErsteEintragungen |
| 1.2 | Maschhod Ahmad | 09.04.2013 | Randbedingung und Anforderung eingetragen |
| 1.3 | Nilüfer Güngör | 15.04.2013 | Ergänzungen (Use-Case-Diagramm) |
| 1.4. | Nilüfer Güngör | 27.05.2013 | Klassendiagramm und Automaten hinzugefügt. |
| 1.5 | Denis Rycka | 04.06.2013 | Automaten aktualisiert, KomponentenDiagramm, Design |
| 1.6 | Nilüfer Güngör | 15.06.2013 | Ergänzungen |
| 1.7 | Erik Matthiessen | 17.06.2013 | Lessons learned |
| 1.8 | Erik Matthiessen | 18.06.2013 | Letzte Eintragungen.  Abnahme.  Bekannte Fehler |

Inhalt

Motivation 3

Randbedingungen 3

Entwicklungsumgebung 3

Werkzeuge **Fehler! Textmarke nicht definiert.**

Sprachen **Fehler! Textmarke nicht definiert.**

Requirements und Use Cases 3

Anforderungen 3

Use-Case-Diagramm **Fehler! Textmarke nicht definiert.**

Design 8

System Architektur 9

Datenmodell 9

Verhaltensmodell 9

Implementierung 15

Algorithmen 15

Patterns 15

Mapping Rules 15

Testen 16

Unit Test/Komponenten Test 16

Integration Test/System Test 16

Regressionstest **Fehler! Textmarke nicht definiert.**

Abnahmetest **Fehler! Textmarke nicht definiert.**

Testplan 17

Testprotokolle und Auswertungen 17

Projektplan 19

Verantwortlichkeiten 19

PSP und Zeitplan 19

Lessons Learned 20

Glossar 20

Abkürzungen **Fehler! Textmarke nicht definiert.**

Anhänge 20

1. Motivation

Im Rahmen des SE2 Praktikums bauen wir ein Werkstück-Sortieranlage aus zwei Förderbandmodulen. Jedes dieser Fördermodule wird durch einen eigenen GEME-Rechner gesteuert. Die beiden Rechner werden über die serielle Schnittstelle gekoppelt.

Ziel des Praktikums ist es, eine vollfunktionstüchtige Sortieranlage zu entwickeln, welche Werkstücke nach bestimmten Kriterien sortieren bzw. aussortieren kann.

Stakeholder:

Entwickler, Vertrieb, Kunden.

(Studierende und Professoren)

1. Randbedingungen
   1. Entwicklungsumgebung

QNX Momentix IDE 4.7

EclipseC++ Juno /MinGW GCC

Visual Paradigm 10.1 Enterprise Edition

VirtualBox

* 1. Werkzeuge

GitHub

TortoiseGit

GitBash

* 1. Sprachen

C++

1. Requirements und Use Cases
   1. Anforderungen

# 3.1.1 Akzeptierte Werkstücke

**Akteur:** Bedienpersonal

**Ziel:** Werkstück erreicht das Ende des zweiten Förderbands

**Auslöser:** Bedienpersonal legt Werkstück an den Anfang des ersten Förderbands

**Vorbedingungen:**

Förderband 1 im Betrieb und bereit (Grüne Ampel)

Anfang des Förderband 1 ist frei (Unterbrechung Lichtschranke)

**Nachbedingungen:**

Werkstück befindet sich am Ende des zweiten Förderbands

**Erfolgsszenario 1:**

1. Erkennen der Höhe des Werkstückes mit der Höhenmessung

2. Werkstück hat akzeptierte Höhe und Bohrung zeigt nach oben, Werkstück wurde akzeptiert

3. Weiche wird geöffnet und das Werkstück durchgelassen

4. Weiche wird geschlossen

5. Werkstück erreicht Ende des ersten Förderbands

6. Übergabe auf das Förderband 2, da dieses frei ist

7. Werkstück wird mit Bohrung nach oben erkannt

8. Werkstück hat keinen Metallkern

9. Weiche wird geöffnet und das Werkstück durchgelassen

10. Weiche wird geschlossen

11. Werkstück erreicht das Ende des zweiten Förderbands

12. Bedienpersonal entnimmt das Werkstück vom zweiten Förderband

**Erfolgsszenario 2:**

1. Erkennen der Höhe des Werkstückes mit der Höhenmessung

2. Werkstück hat akzeptierte Höhe und Bohrung mit Metallkern und zeigt nach oben, Werkstück wurde akzeptiert

3. Weiche wird geöffnet und das Werkstück durchgelassen

4. Weiche wird geschlossen

5. Werkstück erreicht Ende des ersten Förderbands (Gelb Blinkende Ampel, Förderband 1 angehalten)

6. Werkstück wird vom Bedienpersonal umgedreht und zurückgelegt

6. Übergabe auf das Förderband 2, da dieses frei ist

7. Werkstück wird mit akzeptierter Höhe und Bohrung mit Metallkern nach unten erkannt

8. Weiche wird geöffnet und das Werkstück durchgelassen

9. Weiche wird geschlossen

10. Werkstück erreicht das Ende des zweiten Förderbands

11. Bedienpersonal entnimmt das Werkstück vom zweiten Förderband

**Erfolgsszenario 3:**

1. Erkennen der Höhe des Werkstückes mit der Höhenmessung

2. Werkstück hat akzeptierte Höhe und Bohrung zeigt nach unten, Werkstück wurde akzeptiert

3. Weiche wird geöffnet und das Werkstück durchgelassen

4. Weiche wird geschlossen

5. Werkstück erreicht Ende des ersten Förderbands (Gelb Blinkende Ampel, Förderband 1 angehalten)

6. Werkstück wird vom Bedienpersonal umgedreht und zurückgelegt

7. Übergabe auf das Förderband 2, da dieses frei ist

8. Werkstück wird mit Bohrung nach oben erkannt

9. Werkstück hat keinen Metallkern

10. Weiche wird geöffnet und das Werkstück durchgelassen

11. Weiche wird geschlossen

12. Werkstück erreicht das Ende des zweiten Förderbands

13. Bedienpersonal entnimmt das Werkstück vom zweiten Förderband

**Fehlerfälle:**

Verschwinden von Werkstücken

Hinzufügen von Werkstücken mitten auf dem Band

# 3.1.2 Aussortieren von zu flachen Werkstücken

**Akteur:** -

**Ziel:** Werkstück aussortieren

**Auslöser:** Höhenmessung

**Vorbedingungen:**

Förderband 1 im Betrieb und bereit (Grüne Ampel)

**Nachbedingungen:**

Werkstück befindet sich auf der Rutsche

**Erfolgsszenario:**

1. Höhe des Werkstückes wird mit der Höhenmessung erkannt

2. Weiche wird nicht geöffnet

3. Werkstück landet auf der Rutsche

**Fehlerfälle:**

Rutsche Voll

# 3.1.3 Aussortieren von Werkstücken mit Bohrung nach unten

**Akteur:** -

**Ziel:** Werkstück deren Bohrung nach unten liegt, werden von Förderband 2 aussortiert

**Auslöser:** Höhenmessung

**Vorbedingungen:**

Bandanlage im Betrieb (Grüne Ampel)

Nur ein Werkstück befindet sich auf dem Förderband

**Nachbedingungen:**

Werkstück befindet sich auf der Rutsche

**Erfolgsszenario:**

1. Höhe des Werkstückes wird mit der Höhenmessung erkannt

2. Werkstück mit Bohrung nach unten wird erkannt

2. Weiche wird nicht geöffnet

3. Werkstück landet auf der Rutsche

**Fehlerfälle:**

Rutsche Voll

# 3.1.4 Zurücktransport von Werkstücken mit Bohrung und Metallkern nach oben zum Band Anfang

**Akteur:** -

**Ziel:** Werkstücke mit Metallkern, deren Bohrung nach oben liegt, werden vom Förderband 2 zurück zum Anfang transportiert

**Auslöser:** Sensor erkennt Metall

**Vorbedingungen:**

Bandanlage im Betrieb (Grüne Ampel)

Nur ein Werkstück befindet sich auf dem Förderband

**Nachbedingungen:**

Werkstück befindet sich am Anfang des zweiten Förderbands

**Erfolgsszenario:**

1. Höhe des Werkstückes wird mit der Höhenmessung erkannt

2. Werkstück mit Metallkern und Bohrung nach oben erkannt

2. Werkstück wird zurück zum Anfang des Förderband 2 transportiert (Gelb Blinkende Ampel)

**Fehlerfälle:**

# 3.1.5 Rutsche Voll „Fehlerszenario“

**Akteur:** Bedienpersonal

**Ziel:** Fehlerbehebung durch Entleerung der Rutsche

**Auslöser:** Sensor meldet Rutsche voll

**Vorbedingungen:**

Bandanlage im Betrieb (Grüne Ampel)

Werkstück wurde auf die Rutsche geschoben

**Nachbedingungen:**

Bandanlage wieder im Betrieb (Grüne Ampel)

**Fehlerbehebung:**

1. Bandstopp, Rote Signalleuchte schnelles Blinken 1 Hz,( Fehlerzustand: anstehend unquittiert)

2. Bedienpersonal sieht Fehler

3. Bedienpersonal und drückt Quittierungstaste

4. Rote Signalleuchte Dauerlicht, (Fehlerzustand: anstehend quittiert)

5. Bedienpersonal entfernt Werkstücke von der Rutsche

6. Bedienpersonal betätigt die Starttaste

7. Rote Signalleuchte erlischt (Fehlerzustand: OK)

# 3.1.6 Titel: Verschwinden von Werkstücken „Fehlerszenario“

**Akteur:** Bedienpersonal

**Ziel:** Das entnommene Werkstück wird an den Anfang von Band eins gelegt

**Auslöser:** Sensor meldet, dass ein Werkstück fehlt

**Vorbedingung:**

Bandanlage in Betrieb (Grüne Ampel)

Ein Werkstück wird vom Band genommen

**Nachbedingung:**

Bandanlage wieder in Betrieb (Grüne Ampel)

**Fehlerbehebung:**

1. Bandstopp, Rote Signalleuchte schnelles Blinken 1 Hz,( Fehlerzustand: anstehend unquittiert)

2. Bedienpersonal sieht Fehler

3. Bedienpersonal und drückt Quittierungstaste

4. Rote Signalleuchte Dauerlicht, (Fehlerzustand: anstehend quittiert)

5. Bedienpersonal legt das vom Band genommene Werkstück an den Anfang von Band 1

6. Bedienpersonal betätigt die Starttaste

7. Rote Signalleuchte erlischt (Fehlerzustand: OK)

# 3.1.7 Hinzufügen von Werkstücken mitten auf dem Band „Fehlerszenario“

**Akteur:** Bedienpersonal

**Ziel:** Werkstück wird wieder vom Band genommen

**Auslöser:** Sensor meldet, dass ein Werkstück zu viel auf dem Band ist

**Vorbedingung:**

Bandanlage in Betrieb (Grüne Ampel)

Ein Werkstück wird mitten auf dem Band hinzugefügt

**Nachbedingung:**  
Bandanlage wieder in Betrieb (Grüne Ampel)

**Fehlerbehebung:**

1. Bandstopp, Rote Signalleuchte schnelles Blinken 1 Hz,( Fehlerzustand: anstehend unquittiert)

2. Bedienpersonal sieht Fehler

3. Bedienpersonal und drückt Quittierungstaste

4. Rote Signalleuchte Dauerlicht, (Fehlerzustand: anstehend quittiert)

5. Bedienpersonal entfernt das hinzugefügte Werkstück wieder vom Band

6. Bedienpersonal betätigt die Starttaste

7. Rote Signalleuchte erlischt (Fehlerzustand: OK)

* 1. Use-Case-Diagramm

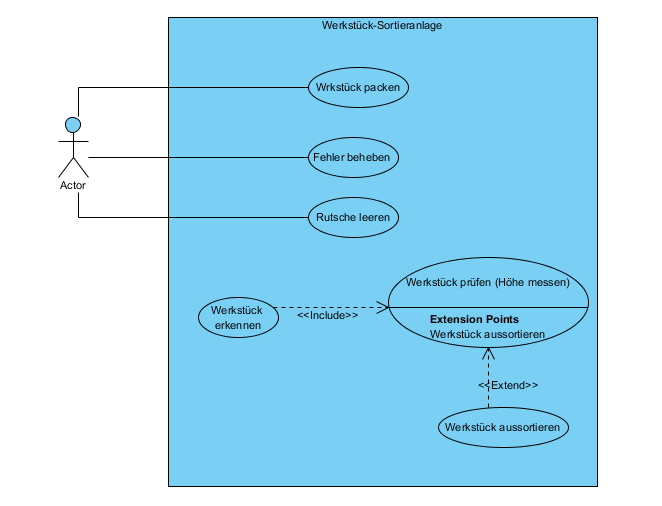


Abbildung 3.2.1: Use-Case-Diagramm der Fließband-Sortieranlage

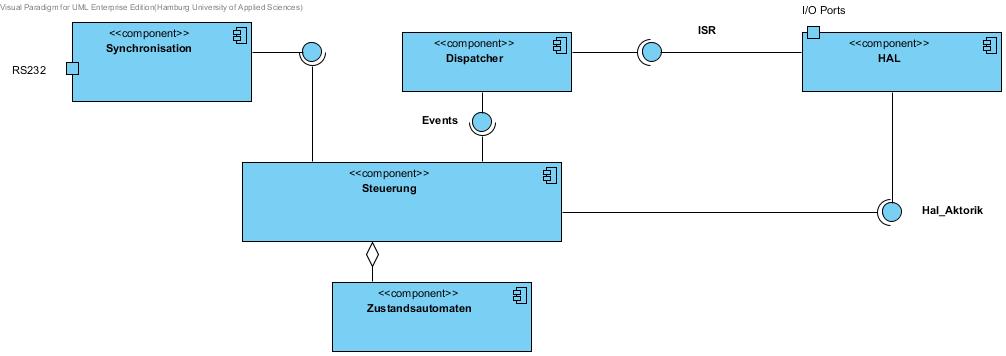
1. Design

**Hardware Abstraktion Schicht (HAL)**

Die Hardware Abstraktion Schicht, kurz HAL, dient zur Kapselung der Hardware.

Die HAL stellt Funktionen bereit, mit denen die Hardware angesprochen werden kann, ohne dass sich der Anwender über die Funktion und die eigentliche Ansteuerung der Hardware Gedanken machen muss. Die HAL ist mit dem Singleton-Pattern implementiert worden, so dass es nur eine Objekt-Instanz gibt, die auf die Hardware zugreifen kann.

* 1. System Architektur



* 1. Datenmodell

Klassendiagramm befindet sich in der Datei „Klassendiagramm.jpg“

Abbildung 4.2.1: Klassendiagramm der Fließband-Sortieranlage

* 1. Verhaltensmodell

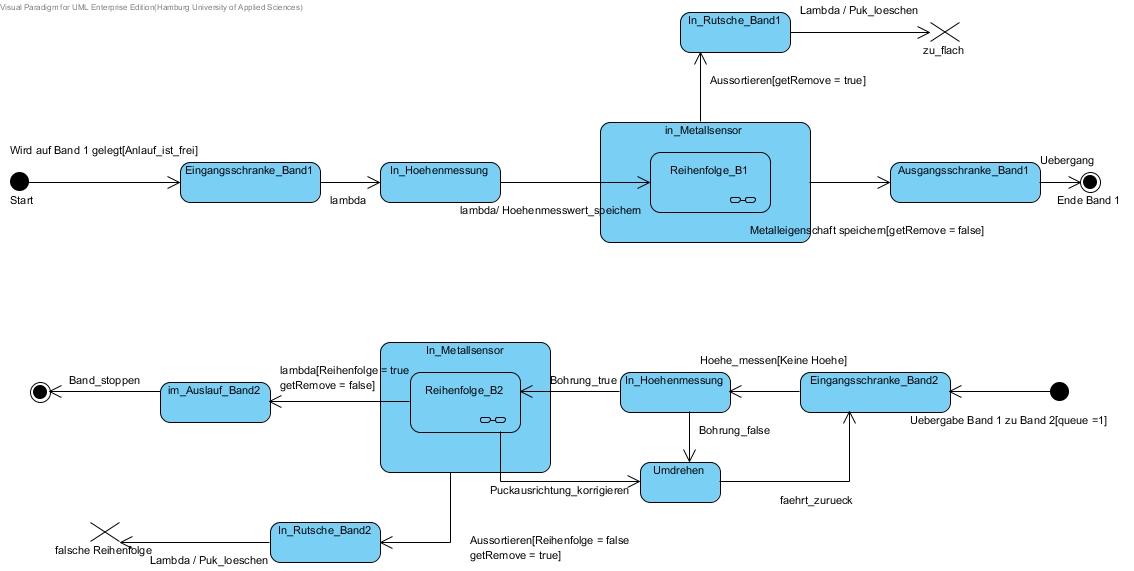


Abbildung 4.3.1: Ablaufdiagramm

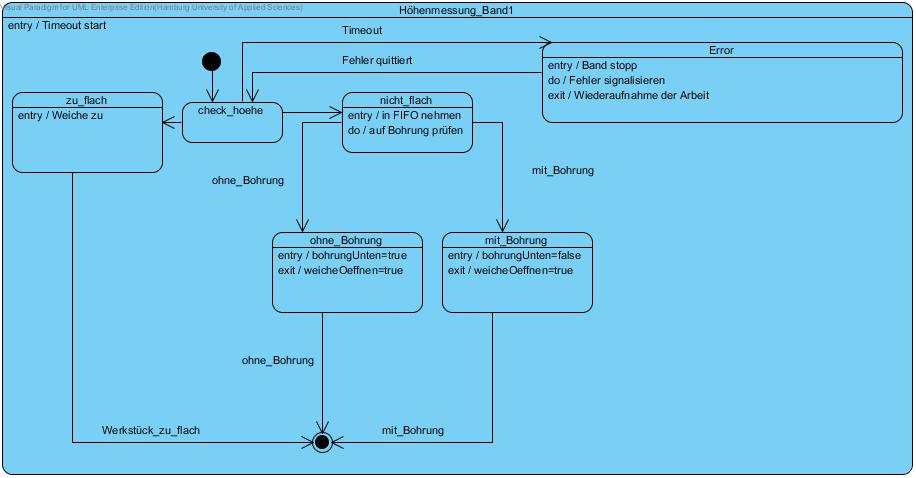


Abbildung 4.3.2: Höhenmessung auf Band 1

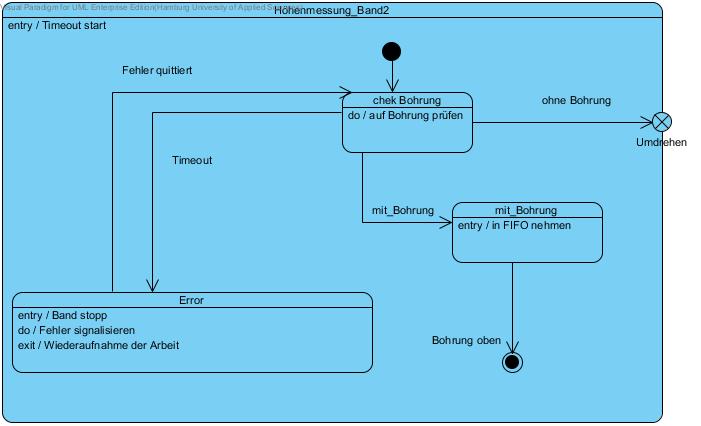


Abbildung 4.3.3: Höhenmessung auf Band 1

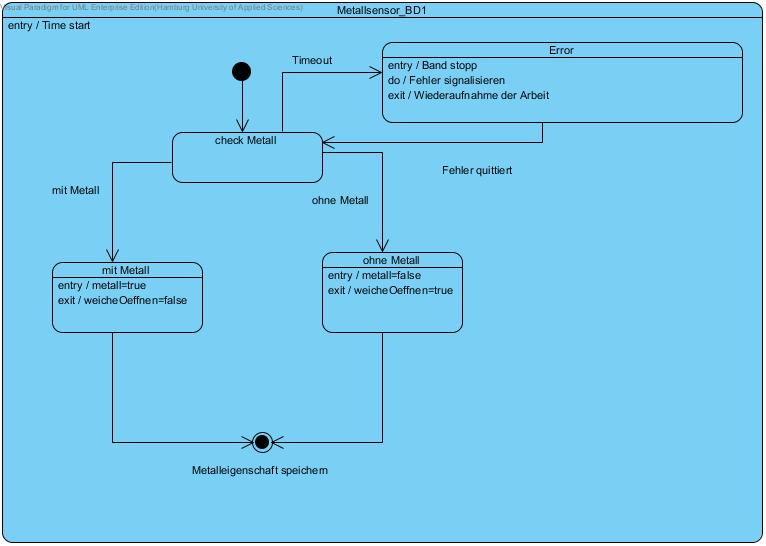


Abbildung 4.3.4: Metallsensor auf Band 1

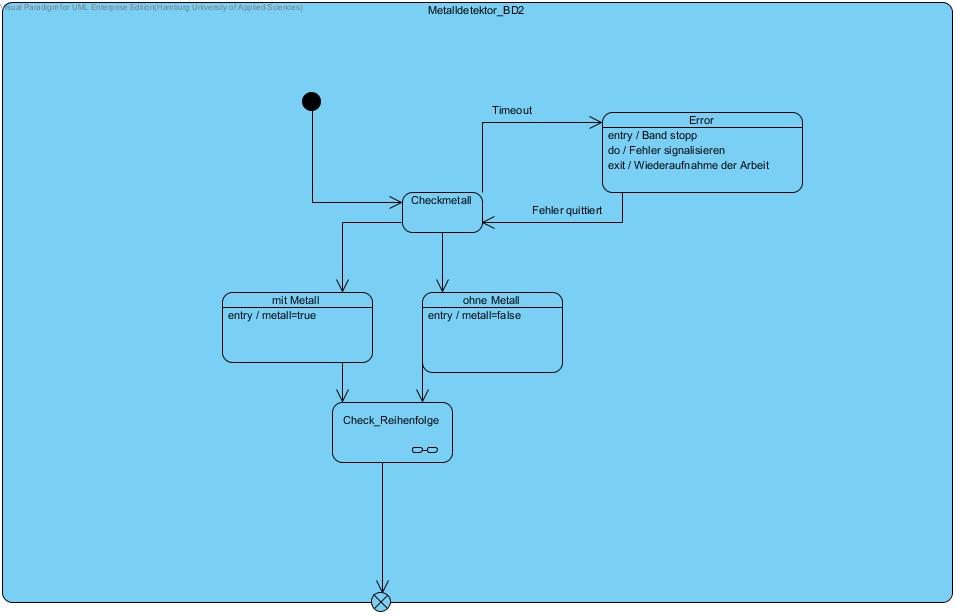


Abbildung 4.3.5: Metallsensor auf Band 2

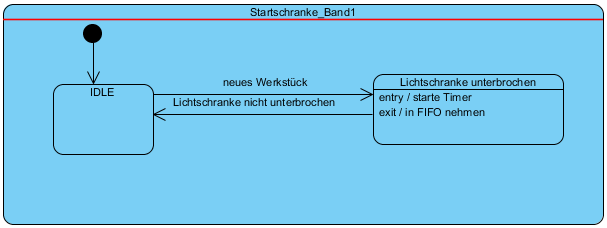


Abbildung 4.3.6: Startschranke auf Band 1

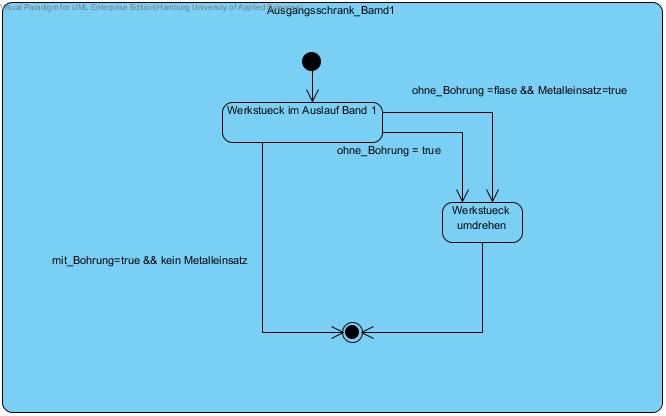


Abbildung 4.3.7: Ausgangschranke auf Band 1

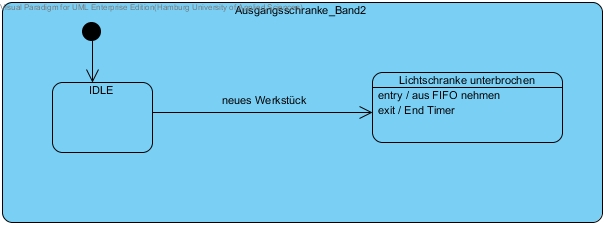


Abbildung 4.3.8: Ausgangschranke auf Band 1

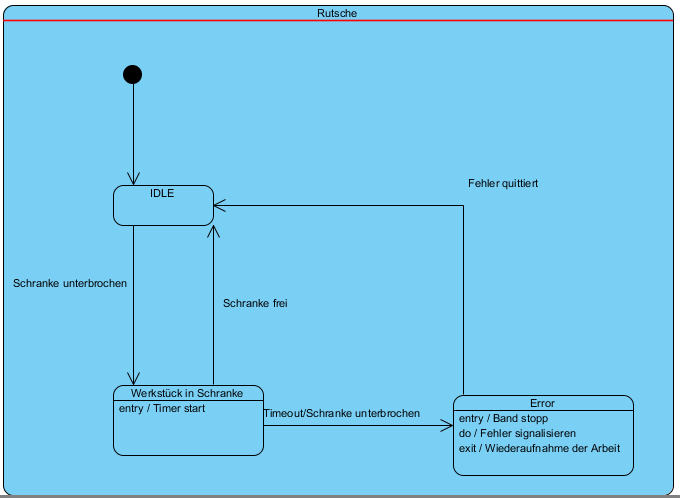


Abbildung 4.3.8: Rutsche

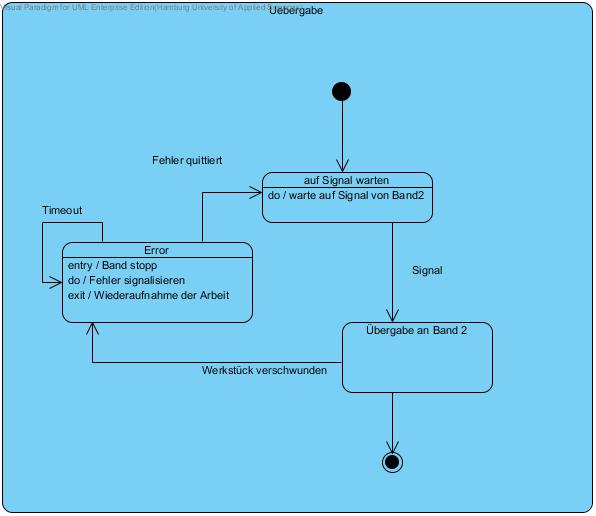


Abbildung 4.3.9: Übergabe von Band 1 nach Band 2

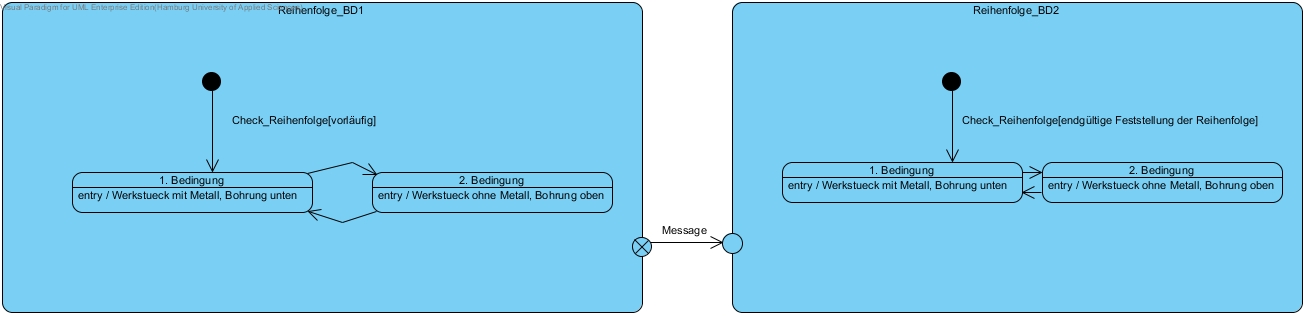


Abbildung 4.3.10: Reihenfolge der akzeptierten Puks

1. Implementierung
   1. Algorithmen

-

* 1. Patterns
* State Pattern
  + FSM (States, aus Pucksicht)
* Facade Pattern
  + Mutex
  + Timer
  + TimerHandler
  + RS232\_1
  + Controller
* Singleton Pattern
  + Dispatcher
  + ISRHandler
  + PuckHandler
  + ErrorFSM
  + ActorHAL
  + SensorHAL
  + LightController
  + RS232\_1
  + Receiver
  + TimerHandler
* Observer Pattern
  + Dispatcher

1. Testen
   1. Regressions- und Komponententests

Um Fehler in Modifikationen bereits getesteter Software innerhalb dieses Projektes zu finden, ist es unerlässlich, über einen Grundstock an Regressionstests zu verfügen.

* 1. Integration Test/System Test

Es wird ein Puk auf Laufband 1 gelegt.

Der Prozess beginnt, indem das Laufband nach rechts fährt.

Der Puk gelang an das Ende von Laufband 2 und seine Werte werden korrekt ausgegeben.

Ein zweiter (zu flacher) Puk wird auf Laufband 1 gelegt.

Er erreicht nicht das Ende des Laufbandes, da er auf Band 1 aussortiert wird.

Band zwei meldet keine Fehlfunktion aufgrund eines fehlenden Puks.

* 1. Regressionstest

Für den Regressionstest gibt es die Datei Regressionstest.cpp.

Für den Test muss die Methode TestLauf() aufgerufen werden.

Der Test lässt einen Puck das gesamte Fließband durchlaufen und damit alle Sensoren passieren.

Weiterhin müssen am Ende alle Knöpfe einmal gedrückt werden.

Ablauf:

1: Eingang

Puk auf den Eingang legen.

Laufband fährt nach rechts.

Grüne Lampe leuchtet.

2: Höhe

Puk erreicht die Höhenmessung.

Laufband fährt nach rechts.

Grüne Lampe leuchtet nicht.

Gelbe Lampe leuchtet.

3: Schranke

Puk erreicht die Schranke.

Schranke geht auf.

Laufband fährt nach rechts.

Gelbe Lampe leuchtet nicht.

Rote Lampe Leuchtet.

4: Ende

Puk erreicht das Ende

Laufband bleibt stehen.

Laufband fährt nach links.

Rote Lampe leuchtet nicht.

LED Q1 leuchtet.

5: Schranke 2

Puk passiert das Laufband in Richtung links.

Laufband bbleibt stehen.

Schranke wird geschlossen

LED Q1 leuchtet nicht.

LED Q2 leuchtet.

Laufband fährt nach rechts.

6: Rutsche

Puk rutscht in die Rutsche.

Laufband bleibt stehen.

Lampen sind aus.

LEDs sind aus.

7: Start

Start-Knopf drücken

LED Start leuchtet.

8: Reset

Reset-Knopf drücken.

LED Reset leuchtet.

9: Stop

Stop-Knopf drücken.

LED Start leuchtet nicht.

10: Qickstop

Qickstop drücken.

LED Reset leuchtet nicht.

* 1. Abnahmetest

Für den Abnahmetest benutzen sie den Testplan.

* 1. Testplan

### Mögliche Puckvarianten

|  |  |
| --- | --- |
| NP | Normaler Puck, Bohrung oben |
| NPT | Normaler Puck, Bohrung unten |
| FP | Zu flacher Puck |
| MP | Puck enthält Metall, Bohrung oben |
| MPT | Puck enthält Metall, Bohrung unten |

### Testreihen

#### Testreihe 1

Kombination: Alle Pucks werden einzeln auf das Band gelegt.

Erwartetes Ergebnis: FP wird auf Band 1 aussortiert, NPT und MPT verursachen am Ende von Band 1 jeweils eine Turnover-Ausnahme. NP und MPT erreichen das Ende von Band 2.

#### Testreihe 2

Kombination: MP+MP+MP+MP

Erwartetes Ergebnis: Alle vier Pucks werden auf Band 2 erkannt und der Benutzer wird aufgefordert den 1. Und 3 MP umzudrehen. 2. Und 4. Puk wird aussortiert, da es nicht der Reihenfolge entspricht.

#### Testreihe 3

Kombination: NP+NP+NP+NP

Erwartetes Ergebnis: Alle Puks werden auf Band 2 aussortiert, da sie nicht der Reihenfolge entsprechen.

#### Testreihe 4

Kombination: FP+FP+FP+FP

Erwartetes Ergebnis:Alle vier Pucks werden auf Band 1 aussortiert und es tritt der Fehler „Rutsche voll“ auf. Der Fehler wird quittiert, die Rutsche geleert und anschließend der normale Betrieb wieder gestartet.

#### Testreihe 5

Kombination: MPT+FP+MP+FP+NP+NP

Erwartetes Ergebnis:Der erste Puck verursacht am Ende von Band 1 eine Turnover-Ausnahme und erreicht danach das Ende von Band 2. Die beiden FP Pucks werden auf Band 1 aussortiert. MP wird auf Band 2 aussortiert und der erste NP erreicht das Ende von Band 2. Das zweite NP wird am Ende von Band 2 aussortiert.

#### Testreihe 6

Kombination: NP+NP+NP

Diese drei NP’s kommen auf Band 1 durch aber auf Band 2 werden sie aussortiert, da sie nicht der Reihenfolge entsprechen.

* 1. Testprotokolle und Auswertungen

Es wurde festgestellt, dass noch folgende Fehler vorhanden sind und als bekannte Fehler aufgeführt werden können.

Fehler:

Es kann vorkommen das es beim umdrehen des Puks am Ende von Band zwei zu einem Bandabsturz kommt.

Mögliche Ursache:

Es scheint daran zu liegen, dass es zu einem Timeout-Fehler gekommen ist, da der Puk zu lange vom Band entfernt war.

Behebung:

Quickstop drücken, alle Puks vom Band entfernen und Fehlerreset durchführen.

Fehler:

Ein Puk mit Metallern wir durchgelassen, obwohl er bei beiden Messungen mit der Bohrung nach unten gelegen hat. Er hätte eigentlich zurückfahren müssen um umgedreht zu werden.

Mögliche Ursache:

Es kann einen Fehler bei der Übertragung des Zustandes gegeben haben. Dabei wird der falsche Zustand des Puks angegeben. Das würde erklären warum am Ende von Band 2 ausgegeben wurde, das der Puk aus Metall ist obwohl das nie gemessen wurde.

Behebung:

Keine Behebung notwendig. Der Puk kann einfach am Ende von Band 2 entfernt werden.

Fehler:

Während die Rutsche voll war und darauf gewartet wurde, dass der Timer dies erkennt, trat ein zweiter Fehler auf. Dieser wurde quittiert und behoben. Es wurde anschließend nicht mehr festgestellt, dass die Rutsche voll ist.

Mögliche Ursache:

Es kann sein, dass die Lichtschranke der Rutsche unterbrochen wurde und damit der Timer gestartet wurde. Bevor die Rutsche für voll erklärt wurde trat der zweite Fehler auf. Die Behebung dauerte zu lange und so wurde der Rutschen Fehler nicht mehr erkannt.

Behebung:

Rutsche leeren. Bei erneuter Überfüllung löst sie wieder den Fehler aus.

Fehler:

Wenn die RS\_232 Verbindung unterbrochen wir kommt es zum Systemstillstand.

Mögliche Ursachen:

Das erste Fließband wartet darauf das es vom zweiten band eine Antwort erhält, da diese jedoch nie kommt passiert auch nichts mehr.

Behebung:

RS\_232 Verbindung wieder herstellen und mit Quckstop und Fehlerreset Normalzustand wieder herstellen.

1. Projektplan
   1. Verantwortlichkeiten

Innerhalb der Gesprächssitzung zur Organisation des Teams wurde sich darauf verständigt, eine demokratische Grundordnung zu verfolgen. Entscheidungen werden gemeinsam im Team getroffen.

* 1. PSP und Zeitplan

Der fertige Projektstrukturplan liegt in der Datei „SE2P\_Projekt\_Planung.mpp“ vor.

1. Lessons Learned

Was haben wir gelernt:

Es ist sehr wichtig eine aktive Kommunikation in der Gruppe zu haben. Die größten Schwierigkeiten hatten wir, da wir uns, vor allem am Anfang des Projektes, nicht richtig Abgesprochen haben. Je weiter das Projekt voran geschritten ist, desto mehr haben wir versucht diesen Fehler zu vermeiden und zu verbessern.

Aus dem Punkt resultierte auch unsere zweite Erkenntnis. Es ist sehr wichtig die Aufgaben klar zu verteilen. Sonst kommt es leicht vor das manche Arbeiten doppelt gemacht werden und andere gar nicht. Es kam auch vor das manchmal einer gar nicht wusste was seine Aufgaben sind und nichts gemacht hat.

Glossar

Fehlerreset – Im Fehlerfall Reset drücken um den Fehler zu quittieren und danach Start drücke um das Band erneut zu starten.

Anhänge

-SE2P\_Projekt\_Planung.mpp