

Requirements and Design Documentation

Version 5.00

SE2P – Praktikum – WS2012

Jan-Tristan Rudat, 2007852, jan-tristan.rudat@haw-hamburg.de

Martin Slowikowski, 1999166, martin.slowikowski@haw-hamburg.de

Chris Addo, 2010200, christopher.addo@haw-hamburg.de

Jens Eberwein, 2007797, jens.eberwein@haw-hamburg.de

Changelog:

Version	Author	Datum	Anmerkungen
0.01	Rudat	14.10.2012	RDD erstellt + Requirements
0.02	Slowikowski	16.10.2012	UML
1.00	Slowikowski	17.10.2012	Aufgabenplan, Milestone 1
1.01	Rudat	17.10.2012	Klassen- + Komponentendiags
2.00	Rudat	22.10.2012	PSP grob
3.01	Slowikowski	07.11.2012	Regressionstests
3.02	Eberwein, Rudat	10.11.2012	Fertigstellung PSP
3.03	Eberwein, Addo	13.11.2012	Erstellung Zustandsautomaten
4.01	Addo, Slowikowski	28.11.2012	Regressionstest State Pattern, Ablauf eingefügt
4.02	Addo, Slowikowski	01.12.2012	FSMs aktualisiert im RDD
4.03	Rudat, Eberwein	01.12.2012	Klassendiagramme aktualisiert
5.00	Addo, Slowikowski	01.12.2012	Timerkonzept eingefügt

Inhaltsverzeichnis

1 Motivation	4
2 Randbedingungen	4
2.1 Entwicklungsumgebung	4
2.2 Werkzeuge	4
2.3 Sprachen	4
3 Requirements und Use Cases (Sequenzdiagramme).....	4
3.1 Allgemeine Anforderungen	4
3.2 Anforderungen	5
3.2.1 Durchlauf akzeptierter Werkstücke	5
3.2.2 Aussortieren von zu flachen Werkstücken.....	6
3.2.3 Aussortieren von Werkstücken mit Bohrung nach oben mit Metalleinsatz	6
3.3 Fehlerszenarien	6
3.3.1 Fehlermeldung „Rutsche voll“	6
3.3.2 Werkstück wurde vom Band genommen	7
3.3.3 Werkstück wurde mitten auf dem Band hinzugefügt	7
3.4 Diagramme	8
3.4.1 Durchlauf akzeptierter Werkstücke	8
3.4.2 Aussortieren von zu flachen Werkstücken.....	9
3.4.3 Aussortieren von Werkstücken mit Bohrung nach oben mit Metalleinsatz	10
3.4.4 Fehlerbehandlung	11
3.5 Timer-Konzept	11
4 Design	12
4.1 System	12
4.2 Datenmodell	13
4.3 Verhaltensmodell	13
5 Implementierung.....	13
5.1 Algorithmen.....	13
5.2 Patterns	13
5.3 Mapping Rules	13
6 Testen	13
6.1 Unit-Test/Komponenten Test.....	13
6.2 Integration Test/System Test	13
6.3 Regressionstest.....	13
6.3.1 Aktorik HAL Testablauf	13

6.3.2 RS232 Testablauf	14
6.3.3 Ampelkontrollthread Testablauf	15
6.3.4 Sensorik HAL Testablauf	15
6.3.5 State Pattern Testablauf.....	18
6.4 Abnahmetest	19
6.5 Testplan	19
6.6 Testprotokolle und Auswertungen.....	19
7 Bedienung der Anlage	19
7.1 Funktionsweise des Laufbands.....	19
7.2 Funktionsweise der Buttons.....	19
7.3 Verhalten im Fehlerfall	19
8 Projektplan	19
8.1 Verantwortlichkeiten.....	19
8.2 Projektstrukturplan	19
9 Lessons Learned	19
Glossar	19
Abkürzungen	19
Anhänge.....	19

1 Motivation

Im Rahmen des Studienganges "Technische Informatik" an der HAW Hamburg, soll im Rahmen des vierten Semesters ein Kurs namens Software Engineering 2 absolviert werden.

In diesem Kurs werden vertiefende Grundlagen des Software Engineering vermittelt, sowie eine Aufgabe erteilt, in welcher die Software für eine Werkstücksortieranlage entwickelt werden soll.

Die Werkstücksortieranlage besteht aus zwei Förderbandmodulen, welche über zwei GEME Rechner gesteuert werden. Beide Rechner sind über eine serielle Schnittstelle miteinander verbunden.

Aus diesem Kontext ist das vorliegende Dokument entstanden.

2 Randbedingungen

2.1 Entwicklungsumgebung

- Visual Paradigm 10.0 Enterprise
- Momentics 4.70 IDE
- QNX 6.5

2.2 Werkzeuge

- GIT
- TortoiseGIT
- Notepad++

2.3 Sprachen

- C
- C++
- Shellscript

3 Requirements und Use Cases (Sequenzdiagramme)

3.1 Allgemeine Anforderungen

Werkstücke werden in gewissen Zeitabständen aufs Band gelegt und Sensoren sortieren bestimmte Werkstücke aus.

Diese Werkstücke können auf das Band gelegt werden:

- mit richtiger Höhe
 - o Mit Metalleinsatz
 - Öffnung nach oben
 - Öffnung nach Unten
 - o Ohne Metalleinsatz
 - Öffnung nach oben
 - Öffnung nach Unten
- mit falscher Höhe

Folgende sollen aussortiert werden:

Höhe	Metall	Öffnung	Aussortieren
richtige Höhe	Mit Metalleinsatz	Öffnung nach oben	Ja (Band 2)
		Öffnung nach Unten	Ja (Band 2, wenden Band 1)
	Ohne Metalleinsatz	Öffnung nach oben	Nein
		Öffnung nach Unten	Nein (wenden Band 1)

mit falsche Höhe			Ja (Band 1)
------------------	--	--	-------------

3.2 Anforderungen

3.2.1 Durchlauf akzeptierter Werkstücke

Akteur: Arbeiter am Förderband

Ziel: Akzeptiertes Werkstück erreicht Ende des zweiten Förderbandes

Auslöser: Arbeiter legt ein Werkstück an den Anfang des ersten Förderbandes

Vorbedingung:

- Förderband 1 in Betrieb und bereit (Ampel: Grün)
- Anfang des ersten Förderbandes ist frei (Schranke 1 nicht unterbrochen)

Erfolgsszenario 1:

1. Ermittlung der Höhe des Werkstückes mit Höhenmessung
2. Werkstück hat akzeptierte Höhe und Bohrung zeigt nach **oben**, Werkstück wird angenommen
3. Öffnen der Weiche, Werkstück wird durchgelassen
4. Weiche wird geschlossen
5. Werkstück erreicht Ende des ersten Förderbandes
6. Transport des Werkstück auf Förderband 2, da dieses frei ist
7. Bohrung des Werkstücks zeigt nach oben
8. Werkstück enthält kein Metallkern
9. Weiche wird geöffnet, Werkstück wird durchgelassen
10. Weiche wird wieder geschlossen
11. Werkstück erreicht das Ende von Band 2, Band zwei bleibt stehen
12. Arbeiter nimmt das Werkstück vom Band 2

Erfolgsszenario 2:

1. Ermittlung der Höhe des Werkstückes mit Höhenmessung
2. Werkstück hat akzeptierte Höhe und Bohrung zeigt nach **unten**, Werkstück wird angenommen
3. Öffnen der Weiche, Werkstück wird durchgelassen
4. Weiche wird geschlossen
5. Werkstück erreicht Ende des ersten Förderbandes (Zustandsanzeige: Gelb blinkend)
6. Arbeiter nimmt Werkstück aus der Lichtschranke, wendet es und legt es zurück
7. Transport des Werkstück auf Förderband 2, da dieses frei ist
8. Bohrung des Werkstücks zeigt nach oben
9. Werkstück enthält kein Metallkern
10. Weiche wird geöffnet, Werkstück wird durchgelassen
11. Weiche wird wieder geschlossen
12. Werkstück erreicht das Ende von Band 2, Band zwei bleibt stehen
13. Arbeiter nimmt das Werkstück vom Band 2

Nachbedingung: Werkstück wird nach Erreichen der Lichtschranke am Ende von Band 2 entnommen

Fehlerfälle: siehe Fehlerszenarien:

- Werkstück wurde vom Band genommen

- Werkstück wurde mitten auf dem Band hinzugefügt

3.2.2 Aussortieren von zu flachen Werkstücken

Akteur: -

Ziel: Werkstücke, deren Höhe kleiner XXmm beträgt, werden von Band 1 aussortiert

Auslöser: Sensor meldet die Höhe des Werkstücks

Vorbedingung:

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Werkstücke befinden sich auf dem Förderband

Erfolgsszenario:

1. Höhe des Werkstückes wird mit Hilfe der Höhenmessung erkannt
2. Werkstück zu flach
3. Weiche bleibt zu
4. Werkstück wird aussortiert

Nachbedingung: Aussortiertes Werkstück befindet sich auf der Rutsche

Fehlerfälle: siehe Fehlerszenarien:

- Rutsche ist voll

3.2.3 Aussortieren von Werkstücken mit Bohrung nach oben mit Metalleinsatz

Akteur: -

Ziel: Werkstücke mit Metalleinsatz, deren Bohrung nach oben liegt, werden von Band 2 aussortiert

Auslöser: Sensor meldet Metall im Werkstück

Vorbedingung:

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Ein Werkstück befindet sich auf dem Förderband

Erfolgsszenario:

1. Metallsensor auf Band 2 erkennt Metallkern im Werkstück
2. Weiche bleibt geschlossen
3. Werkstück wird aussortiert

Nachbedingung: Aussortiertes Werkstück befindet sich auf der Rutsche

Fehlerfälle: siehe Fehlerszenarien:

- Rutsche ist voll

3.3 Fehlerszenarien

3.3.1 Fehlermeldung „Rutsche voll“

Akteur: Arbeiter am Förderband

Ziel: Behebung des Fehlers: Rutsche entleeren

Auslöser: Sensor meldet Rutsche voll

Vorbedingung:

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Ein Werkstück wurde auf die Rutsche geschoben

Ablauf der Fehlerbehebung:

1. Bandstopp, Zustandsanzeige blinkt rot (Fehlerzustand: „anstehend unquittiert“, schnelles Blinken 1 Hz)
2. Arbeiter sieht den Fehler
3. Arbeiter drückt die Quittierungstaste
4. Zustandsanzeige: rotes Dauerlicht (Fehlerzustand: „anstehend quittiert“)
5. Arbeiter nimmt Werkstücke von der Rutsche
6. Arbeiter betätigt die Starttaste
7. Rote Leuchte erlischt

Nachbedingung: Bandanlage wieder in Betrieb (Zustandsanzeige: Grün)

3.3.2 Werkstück wurde vom Band genommen

Akteur: Arbeiter am Förderband

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

Auslöser: Sensor meldet, dass ein Werkstück fehlt

Vorbedingung:

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Ein Werkstück wird vom Band genommen

Ablauf der Fehlerbehebung:

1. Bandstopp, Zustandsanzeige blinkt rot (Fehlerzustand: „anstehend unquittiert“, schnelles Blinken 1Hz)
2. Arbeiter sieht den Fehler
3. Arbeiter drückt die Quittierungstaste
4. Zustandsanzeige: rotes Dauerlicht (Fehlerzustand: „anstehend quittiert“)
5. Arbeiter legt das vom Band genommene Werkstück an den Anfang von Band 1
6. Arbeiter betätigt die Starttaste
7. Rote Leuchte erlischt

Nachbedingung: Bandanlage wieder in Betrieb (Zustandsanzeige: Grün)

3.3.3 Werkstück wurde mitten auf dem Band hinzugefügt

Akteur: Arbeiter am Förderband

Ziel: Werkstück wird wieder vom Band genommen

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

Vorbedingung:

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Ein Werkstück wird mitten auf dem Band hinzugefügt

Ablauf der Fehlerbehebung:

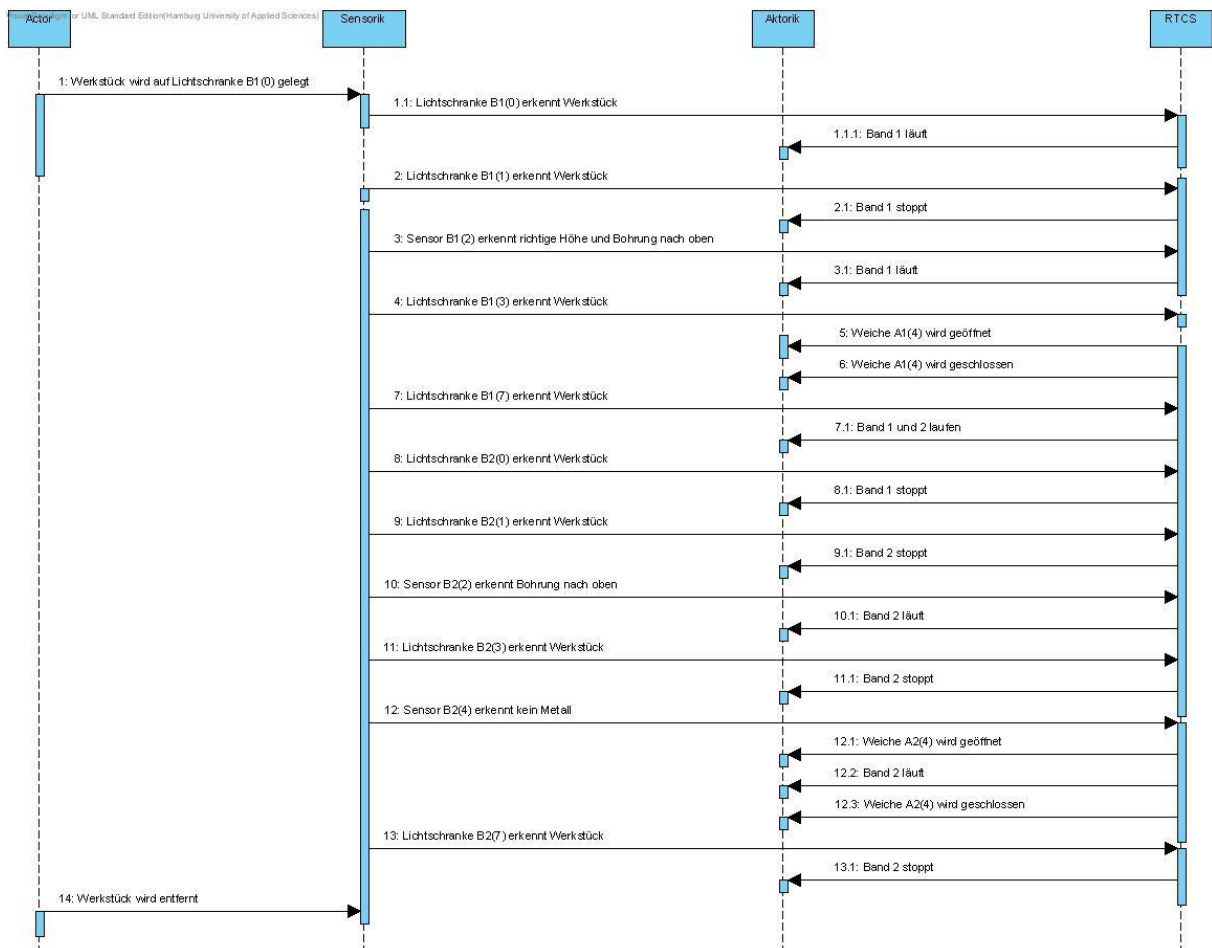
1. Bandstopp, Zustandsanzeige blinkt rot (Fehlerzustand: „anstehend unquittiert“, schnelles Blinken 1Hz)
2. Arbeiter sieht den Fehler
3. Arbeiter drückt die Quittierungstaste
4. Zustandsanzeige: rotes Dauerlicht (Fehlerzustand: „anstehend quittiert“)
5. Arbeiter entfernt das hinzugefügte Werkstück wieder vom Band
6. Arbeiter betätigt die Starttaste
7. Rote Leuchte erlischt

Nachbedingung: Bandanlage wieder in Betrieb (Zustandsanzeige: Grün)

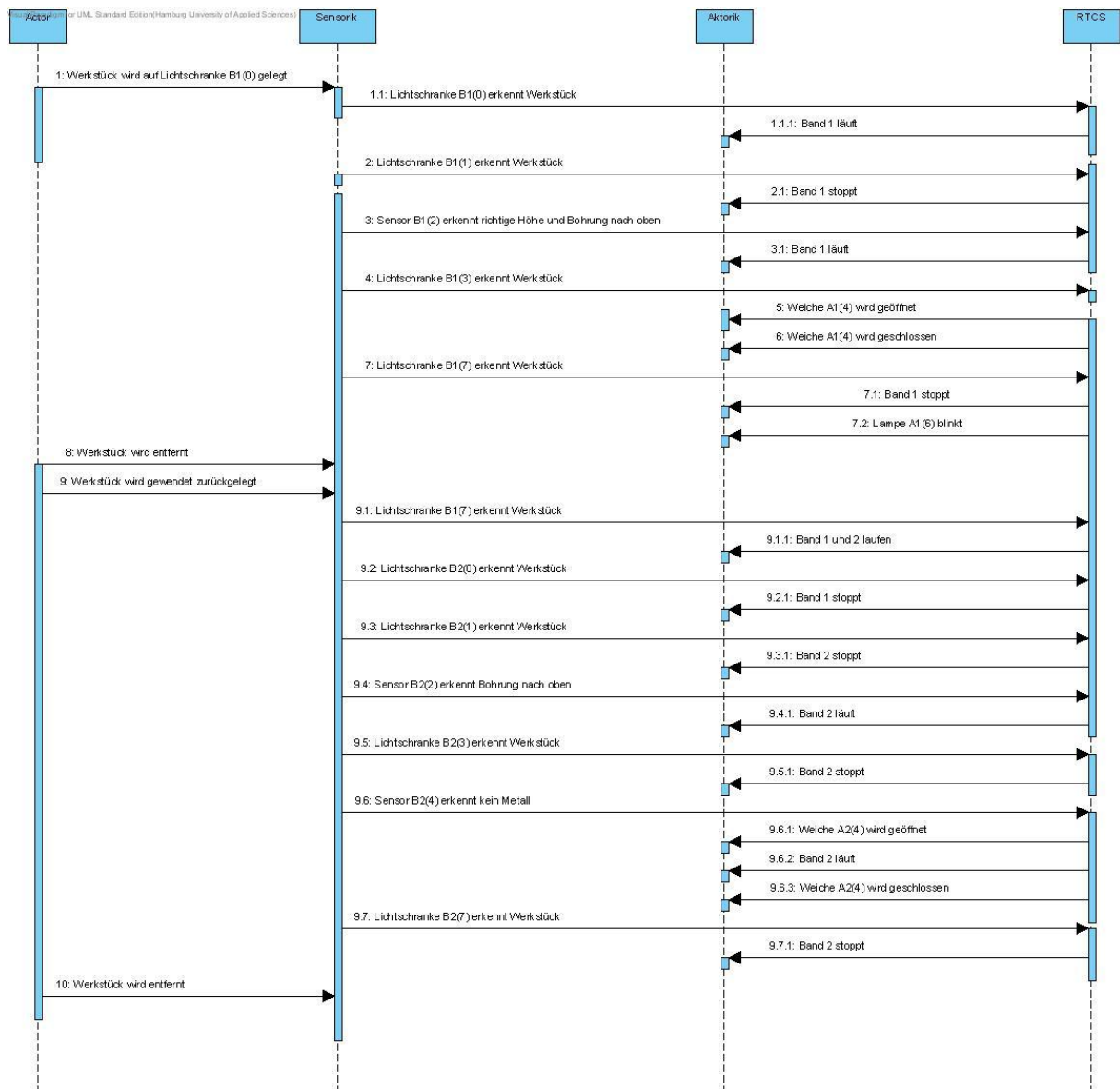
3.4 Diagramme

3.4.1 Durchlauf akzeptierter Werkstücke

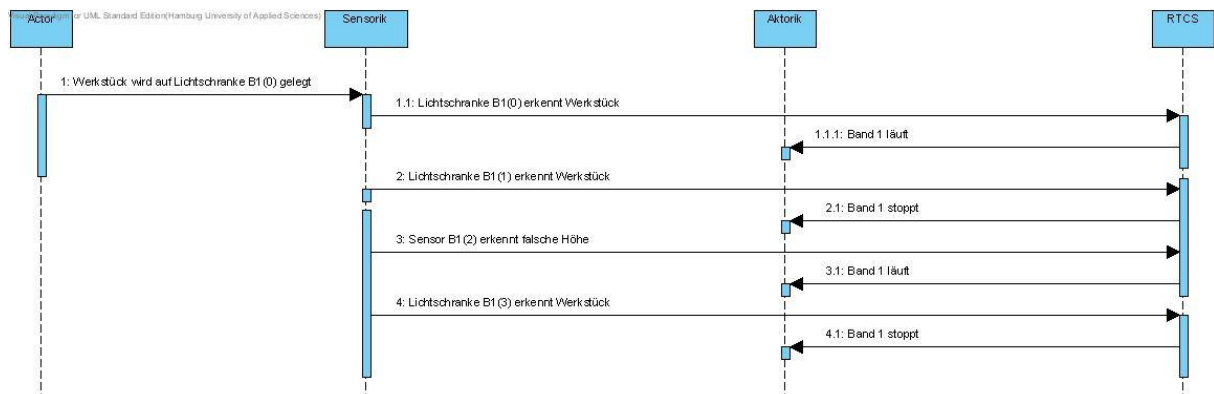
Szenario 1, richtige Höhe, Bohrung nach oben, kein Metall:



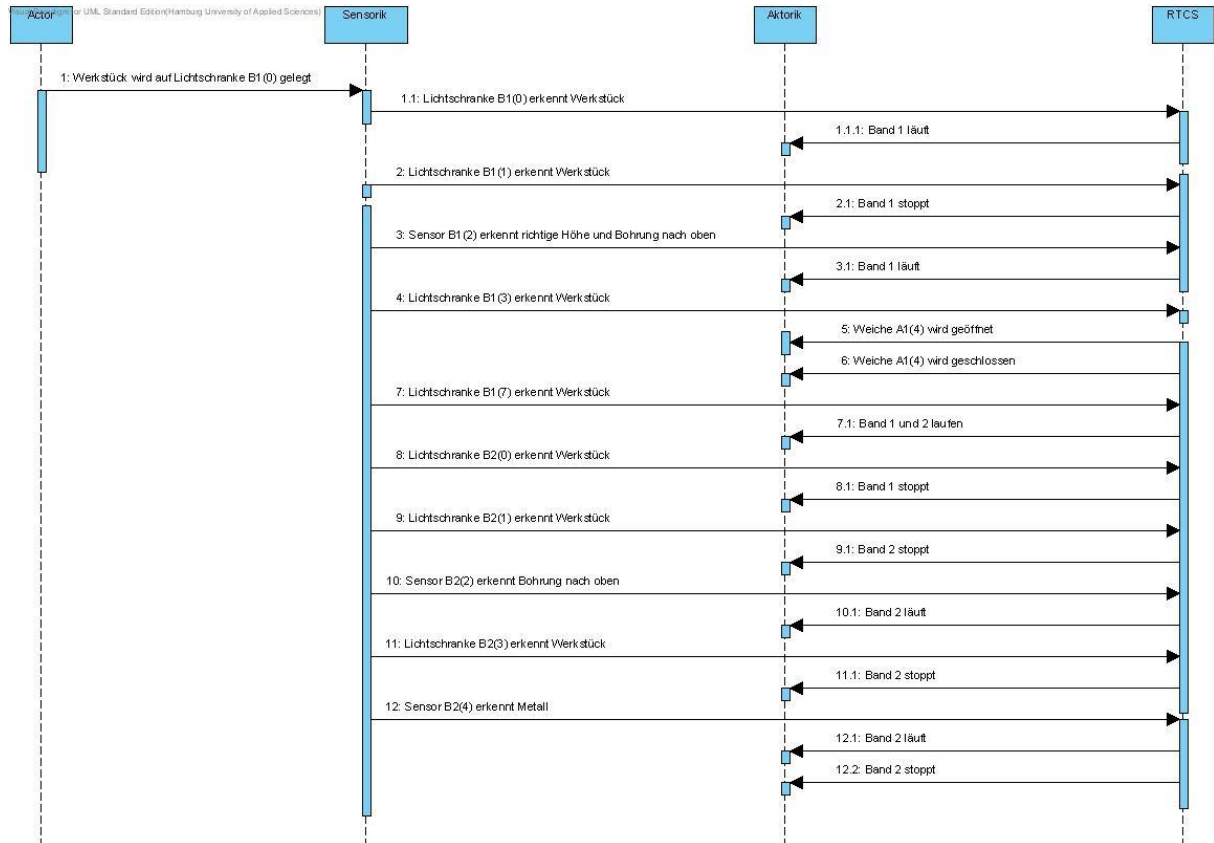
Szenario 2, richtige Höhe, Bohrung nach unten, kein Metall:



3.4.2 Aussortieren von zu flachen Werkstücken

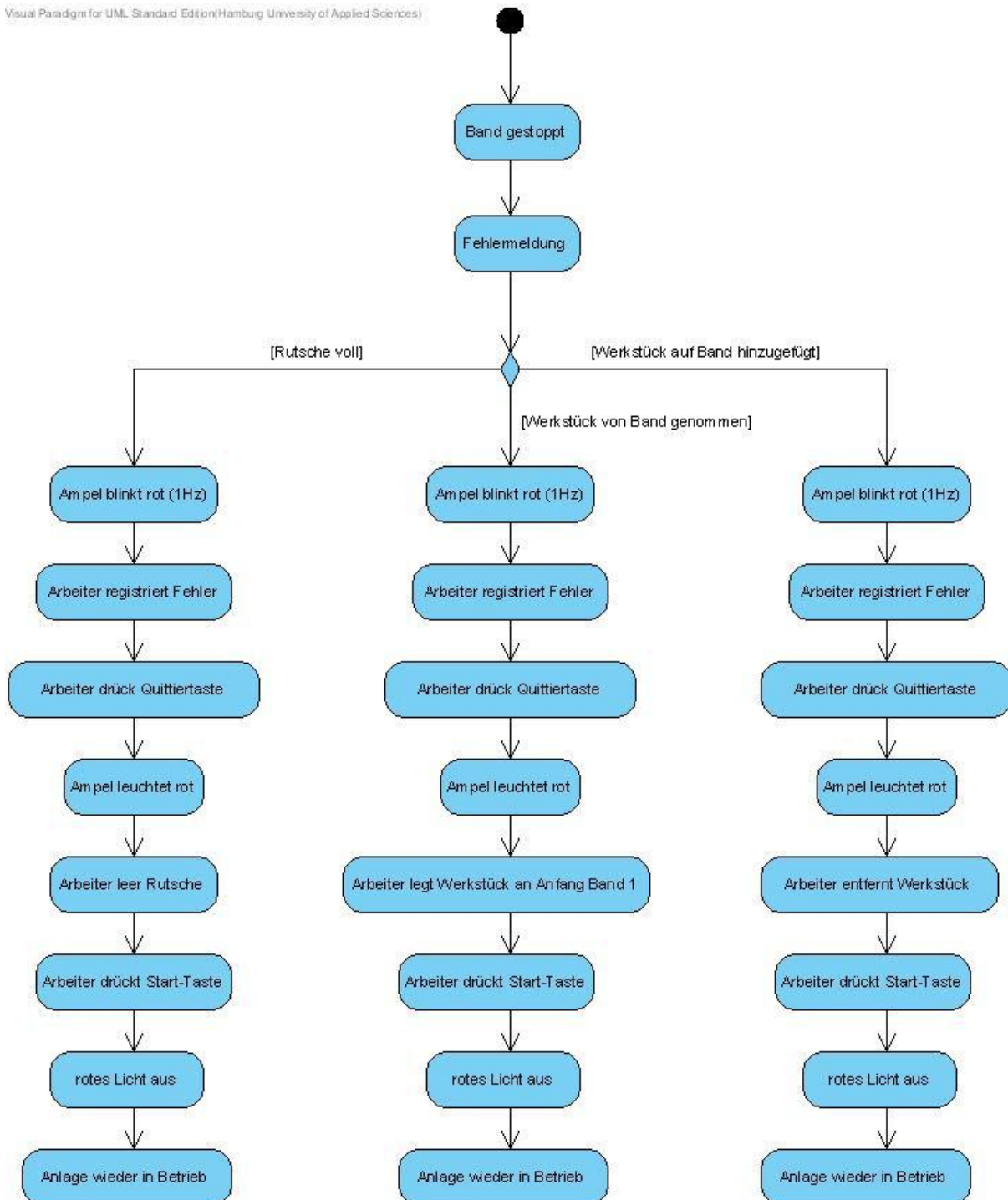


3.4.3 Aussortieren von Werkstücken mit Bohrung nach oben mit Metalleinsatz



3.4.4 Fehlerbehandlung

Visual Paradigm for UML Standard Edition (Hamburg University of Applied Sciences)



3.5 Timer-Konzept

Timer werden benötigt, um gewisse Fehlerszenarien darstellen zu können, um einen geregelten Bandablauf sicherstellen zu können. Dafür werden an verschiedenen Stellen des Bandes Timer eingesetzt.

An folgenden Positionen sollen Timer zum Einsatz kommen:

1. Rutsche
 - Wurde die Lichtschranke unterbrochen, wurde aber nach x Sekunden nicht wieder geschlossen, ist die Rutsche voll

2. Weiche

- Wurde die Weiche für einen Puck geöffnet, muss sie nach x Sekunden wieder geschlossen werden

3. Band 1 am Ende

- Bei der Übergabe eines Pucks von Band 1 nach Band 2 muss das Band 1 noch x Sekunden weiterlaufen, damit der Puck auf das Band 2 transportiert wird

4. Band 1 am Ende

- Wartet am Ende von Band 1 ein Puck auf die Übergabe, und kein ACK wird nach x Sekunden empfangen, ist Band 2 vermutlich gestört (z.B. EStop, Fehler, nicht aktiv)

5. Band 1 und Band 2 für jedes der drei Segmente

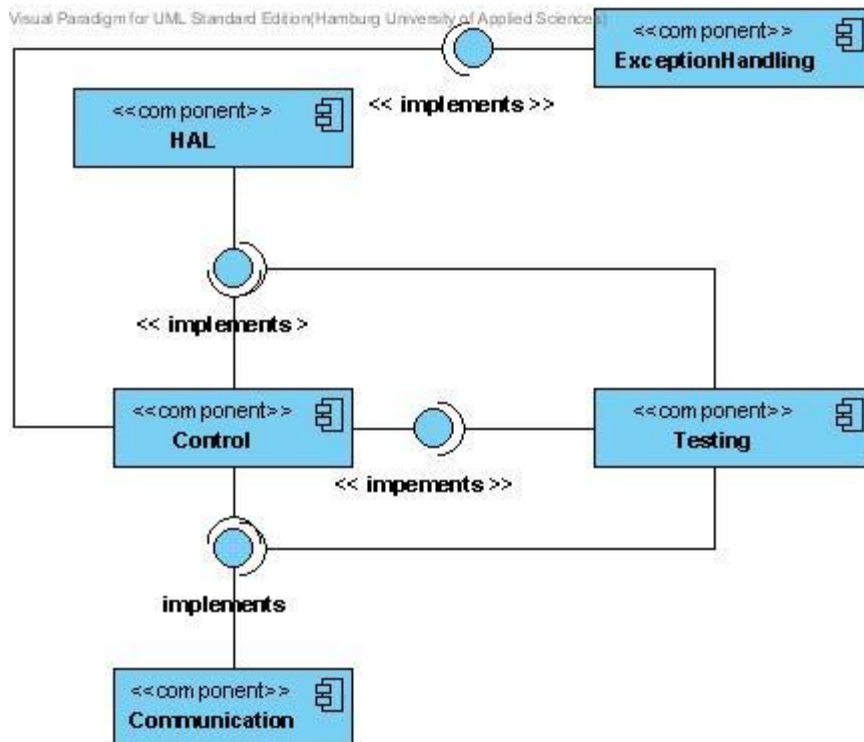
- Min/Max Timer für die Fahrtzeit durch die drei Segmente pro Werkstück (z.B. vom Bandanfang (LS1 wieder geschlossen) bis zur Höhenkontrolle (LS2 unterbrochen))

Das Auslösen eines Timers hat das Absenden eines Pulses zur Folge. Je nachdem, welcher Timer ausgelöst wurde, sind für die oben genannten Timer verschiedene Ziele der Pulse zur Abarbeitung zu definieren:

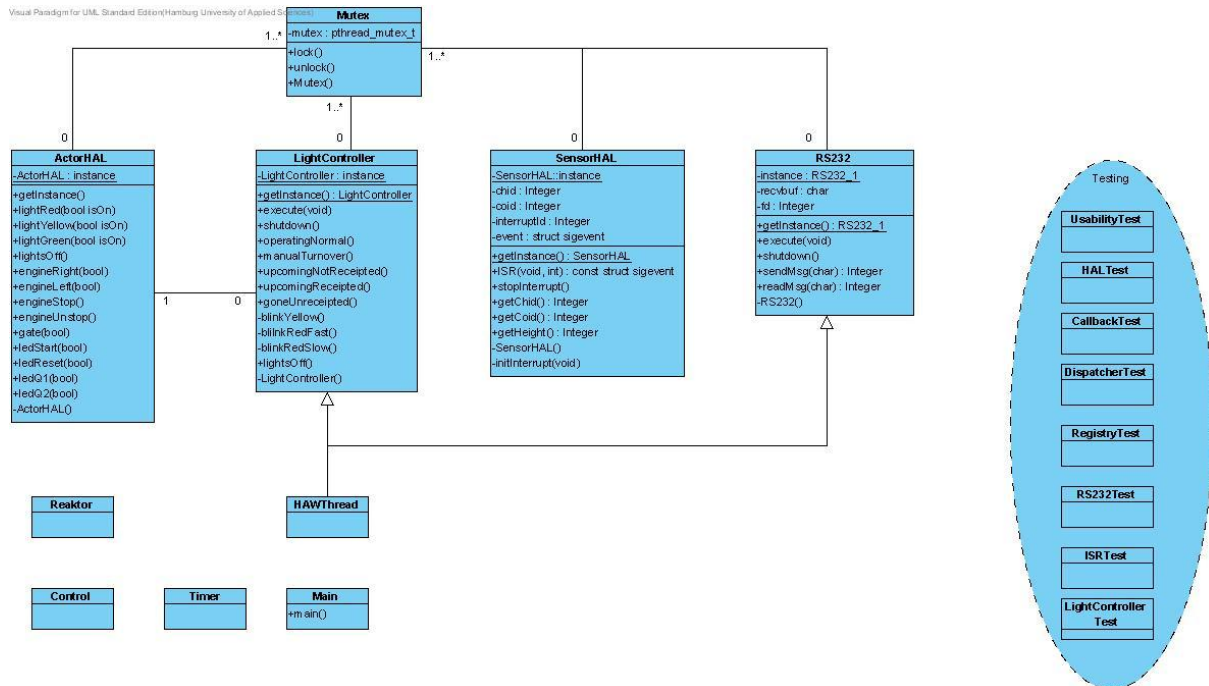
1. Der Dispatcher erhält den Pulse, dass die Rutsche voll ist
2. Der Dispatcher erhält den Pulse, dass die Weiche geschlossen werden muss
3. Der Dispatcher von Band 1 erhält den Pulse, dass das Band noch weiterlaufen muss
4. Die Error FSM erhält den Pulse, dass am Ende von Band 1 kein ACK von Band 2 kam
5. Die Error FSM erhält den Pulse, dass ein Timer unter- oder überschritten wurde

4 Design

4.1 System



4.2 Datenmodell



4.3 Verhaltensmodell

5 Implementierung

5.1 Algorithmen

5.2 Patterns

- State Pattern
- Facade Pattern
- Singleton Pattern

5.3 Mapping Rules

6 Testen

6.1 Unit-Test/Komponenten Test

6.2 Integration Test/System Test

6.3 Regressionstest

Um Fehler in Modifikationen bereits getesteter Software innerhalb dieses Projektes zu finden, ist es unerlässlich, über einen Grundstock an Regressionstests zu verfügen. Es werden folgende Testfälle ausgeführt:

6.3.1 Aktorik HAL Testablauf

Für einen Test der gesamten Aktorik werden folgende Zeilen Code verwendet:

```

39     HALTest halTest;
40     halTest.testHal();

```

Folgende Ausgabe auf der Konsole wird erwartet, analog zur Ausgabe soll das Verhalten am Festo-System zu beobachten sein:

Debug Hal: New HAL instance created Debug Hal: red light on Debug Hal: red light off Debug Hal: yellow light on Debug Hal: yellow light off Debug Hal: green light on Debug Hal: green light off Debug Hal: green light on Debug Hal: yellow light on Debug Hal: red light on Debug Hal: all lights off Debug Hal: engine right with normal speed Debug Hal: engine right with slow speed Debug Hal: engine stopped Debug Hal: engine revert stop Debug Hal: engine stopped Debug Hal: gate open Debug Hal: gate closed Debug Hal: start led on Debug Hal: reset led on Debug Hal: Q1 led on Debug Hal: Q2 led on Debug Hal: start led off Debug Hal: reset led off Debug Hal: Q1 led off Debug Hal: Q2 led off	rotes Licht an Ampel geht an rotes Licht an Ampel geht aus gelbes Licht an Ampel geht an gelbes Licht an Ampel geht aus grünes Licht an Ampel geht an grünes Licht an Ampel geht aus grünes Licht an Ampel geht an gelbes Licht an Ampel geht an rotes Licht an Ampel geht an alle Lichter an Ampel gehen aus Laufband fährt rechts Laufband fährt langsam rechts Laufband hält an Laufband fährt weiter rechts Laufband hält an Weiche öffnet sich Weiche schließt sich LED Start geht an LED Reset geht an LED Q1 geht an LED Q2 geht an LED Start geht aus LED Reset geht aus LED Q1 geht aus LED Q2 geht aus
--	--

6.3.2 RS232 Testablauf

Für einen Test der seriellen Schnittstelle werden folgende Zeilen Code verwendet, weiterhin müssen die beiden COM-Ports des GEME-PC mit einem Null-Modem Kabel verbunden werden:

```

42     RS232Test rs232Test;
43     rs232Test.testRS232();

```

Folgende Ausgabe ist bei korrekter Funktion auf der Konsole zu erwarten:

Debug RS232_1: opening devfile1 SUCCEEDED Debug RS232_1: New RS232_1 instance created Debug RS232_2: opening devfile2 SUCCEEDED Debug RS232_2: New RS232_2 instance created Debug RS232_1: Unknown msg recved: b Debug RS232_1: Timeout or EAGAIN Debug RS232_1: Timeout recved Debug RS232_1: Timeout or EAGAIN Debug RS232_1: Timeout recved Testmessage recved on devfile1: a Debug RS232_1: Timeout or EAGAIN Debug RS232_1: Timeout recved Debug RS232_1: Timeout or EAGAIN Debug RS232_1: Timeout recved	COM1 initialisiert und geöffnet COM2 initialisiert und geöffnet Lese auf COM1, schreibe auf COM2 Unbekannte Nachricht ,b' empfangen Zyklisch generierter Timeout Gültige Nachricht ,a' empfangen
---	---

1. Zu kleines Werkstück (landet auf der Rutsche)
2. Akzeptiertes Werkstück, Bohrung nach oben (Band stoppt, wenn WS das Ende erreicht)
3. Akzeptiertes Werkstück, Bohrung nach unten (Band stoppt, wenn WS das Ende erreicht, Ampel blinkt gelb)
4. Werkstück mit Metall (WS landet auf der Rutsche)

Anschließend werden die Knöpfe bedient:

1. Reset
2. Stop
3. E-Stop drücken
4. Start
5. E-Stop lösen
6. Start

Diese Reihenfolge führt zu folgender Ausgabe auf der Konsole:

Debug SensorHAL: New SensorHAL instance created Debug Hal: New HAL instance created Debug LightController: New LC instance created Werkstueck im Einlauf Debug Hal: engine right with normal speed Debug Hal: engine revert stop Kein Werkstueck im Einlauf Werkstueck in Hoehenmessung Werkstueck Hoehe: 2712 Debug Hal: engine stopped Debug Hal: engine revert stop Kein Werkstueck in Hoehenmessung Werkstueck in Weiche Kein Werkstueck in Weiche Rutsche voll Rutsche nicht voll Debug Hal: engine stopped Werkstueck im Einlauf Debug Hal: engine right with normal speed Debug Hal: engine revert stop Kein Werkstueck im Einlauf Werkstueck im Toleranzbereich: 4040 Werkstueck zu klein/gross: 2525 Werkstueck in Hoehenmessung Werkstueck Hoehe: 3526 Debug Hal: engine stopped Debug Hal: engine revert stop Werkstueck im Toleranzbereich: 4025 Werkstueck zu klein/gross: 2548 Kein Werkstueck in Hoehenmessung Werkstueck in Weiche Debug Hal: gate open Weiche offen Kein Werkstueck in Weiche Werkstueck im Auslauf Debug Hal: gate closed Debug Hal: engine stopped Weiche geschlossen	 Werkstück auf Band gelegt Band fährt los Werkstück ist in Höhenmessung Messung Band stoppt Nach Messung/Prüfung weiterfahren Werkstück an der Weiche Werkstück wird aussortiert Neues Werkstück auf Band gelegt Werkstück erreicht Weiche Werkstück ok, daher Weiche öffnen Werkstück erreicht Ende des Bands Weiche wieder schließen
---	---

kein Werkstueck im Auslauf Debug Hal: all lights off Werkstueck im Einlauf Debug Hal: engine right with normal speed Debug Hal: engine revert stop Kein Werkstueck im Einlauf Werkstueck im Toleranzbereich: 4023 Werkstueck in Hoehenmessung Werkstueck Hoehe: 2471 Debug Hal: engine stopped Debug Hal: engine revert stop Werkstueck zu klein/gross: 2551 Kein Werkstueck in Hoehenmessung Werkstueck in Weiche Debug Hal: gate open Weiche offen Kein Werkstueck in Weiche Werkstueck im Auslauf Debug Hal: gate closed Debug Hal: engine stopped manualTurnover: 1 Debug Hal: all lights off Debug Hal: yellow light on Weiche geschlossen Debug Hal: yellow light off Debug Hal: yellow light on kein Werkstueck im Auslauf Debug Hal: all lights off Werkstueck im Einlauf Debug Hal: engine right with normal speed Debug Hal: engine revert stop Kein Werkstueck im Einlauf Werkstueck im Toleranzbereich: 2663 Werkstueck zu klein/gross: 2510 Werkstueck in Hoehenmessung Werkstueck Hoehe: 3465 Debug Hal: engine stopped Debug Hal: engine revert stop Werkstueck im Toleranzbereich: 3800 Werkstueck zu klein/gross: 2553 Kein Werkstueck in Hoehenmessung Werkstueck Metall Werkstueck in Weiche Werkstueck kein Metall Kein Werkstueck in Weiche Rutsche voll Rutsche nicht voll Debug Hal: engine stopped Resettaste gedrueckt Resettaste losgelassen Stoptaste gedrueckt Stoptaste losgelassen E-Stoptaste gedrueckt System angehalten mittels EStop Debug Hal: red light on Starttste gedrueck Starttste losgelassen E-Stoptaste geloest E-Stoptaste gedrueckt	Neues Werkstück auf Band gelegt Werkstück erreicht Ende des Bands Werkstück muss gewendet werden Gelbes blinken Neues Werkstück auf Band gelegt Werkstück enthält Metall Werkstück aussortieren Reset Taster Stop Taster E-Stop Taster Rotes Licht geht an Start Taster gedrückt, rotes Licht bleibt an E-Stop Taster zurückgesetzt, Schalter kann prellen, an dieser Stelle mit
---	--

```
System angehalten mittels EStop
Debug Hal: red light on
E-Stoptaste geloest
E-Stoptaste gedrueckt
System angehalten mittels EStop
Debug Hal: red light on
E-Stoptaste geloest
E-Stoptaste gedrueckt
System angehalten mittels EStop
Debug Hal: red light on
E-Stoptaste geloest
E-Stoptaste gedrueckt
System angehalten mittels EStop
Debug Hal: red light on
E-Stoptaste geloest
Starttste gedrueck
System rennt weiter, EStop geloest und
start
Debug Hal: red light off
Debug Hal: green light on
Starttste losgelassen
```

bool und Start Taster verbunden

Nach rücksetzen von E-Stop Start Taster gedrückt, System läuft wieder, rotes Licht geht aus, grünes Licht geht an

6.3.5 State Pattern Testablauf

Für einen Test des State Patterns muss in der Datei „bandselection.h“ folgendes define gesetzt werden:

```
20 //valid defines are BAND_1 BAND_2 BAND_TEST
21 #define BAND_TEST
```

Weiterhin wird folgendes Setup in der „main.cpp“ benötigt:

```
64 ErrorFSM* errfsm = ErrorFSM::getInstance();
65 errfsm->start(0);
66
67 Dispatcher* disp = Dispatcher::getInstance();
68 disp->start(0);
69
70 ISRHandler* isrhandler = ISRHandler::getInstance();
71 isrhandler->start(0);
72
73 PuckHandler::getInstance()->initializePucks(disp);
```

Für den Test müssen nach und nach alle Lichtschranken unterbrochen und wieder geschlossen werden. Folgende Reihenfolge ist für den Test vorgesehen:

1. Lichtschranke am Bandanfang unterbrechen/schließen
2. Lichtschranke an der Höhenmessung unterbrechen/schließen
3. Lichtschranke an der Weiche unterbrechen/schließen
4. Lichtschranke an der Rutsche unterbrechen/schließen
5. Lichtschranke am Bandende unterbrechen/schließen

Dieser Ablauf bewirkt folgende Ausgabe auf der Konsole:

Entering State 1: LS1	LS am Bandanfang unterbrochen (Konstruktor)
-----------------------	---

Leaving State 1: LS1	LS am Bandanfang geschlossen (new State 2)
Entering State 2: LS2	LS an Höhenkontrolle unterbrochen (Konstruktor)
Leaving State 2: LS2	LS an Höhenkontrolle geschlossen (new State 3)
Entering State 3: LS3	LS an der Weiche unterbrochen (Konstruktor)
Leaving State 3: LS3	LS an Weiche geschlossen (new State 4)
Entering State 4: LS4	LS an Rutsche unterbrochen (Konstruktor)
Leaving State 4: LS4	LS an Rutsche geschlossen (new State 5)
Entering State 5: LS5	LS am Bandende unterbrochen (Konstruktor)
Leaving State 5: LS5	LS am Bandende geschlossen (Test beendet)
time to kill test program ...	

6.4 Abnahmetest

6.5 Testplan

6.6 Testprotokolle und Auswertungen

7 Bedienung der Anlage

7.1 Funktionsweise des Laufbands

7.2 Funktionsweise der Buttons

7.3 Verhalten im Fehlerfall

8 Projektplan

8.1 Verantwortlichkeiten

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

8.2 Projektstrukturplan

Der fertige Projektstrukturplan liegt in der Datei „se2psp.pdf“ vor.

9 Lessons Learned

Glossar

Abkürzungen

Anhänge