

Requirements / Design and Test Documentation (RDT)

Version 1.1

ESEP – Praktikum – Sommersemester 2023

Lorenz, Maik, 2542513, maik.lorenz@haw-hamburg.de

Schukow, Dominik, 2441109, dominik.schukow@haw-hamburg.de

Malik, Sulaiman, 2441151, sulaiman.malik@haw-hamburg.de

Änderungshistorie:

Version	Erstellt	Autor	Kommentar
0.1	2018-03-12	LMN	Initiale Version des Templates.
0.2	2020-03-15	DAI	Überarbeitung wegen Corona.
0.3	2022-02-24	LMN	Anpassungen für Sommersemester. Anforderungen an Requirements reduziert auf Ergänzungen.
0.4	2022-11-22 ff.	CHRS	Neustrukturierung des Templates, Schriftgrößen vereinheitlicht, Erweiterungen: Hinweise am Anfang des Dokuments, Unterkapitel Hardware und technische Gegebenheiten, Unterkapitel Analyse des Kundenwunsches , Unterkapitel Nachrichten und Signale, allg. Abnahmetest Text + Tabelle, Unterkapitel Abbildungsverzeichnis
0.5	2023-03-28	DOM, SUL, MAI	Absprachen, Requirement Analysis, Project management, System context diagram, Abkürzungen, Glossar
0.6	2023-04-13	DOM, SUL, MAI	Requirement Analysis (Absprachen), Requirement (Software and Hardware perspective) Uses Cases, Warning, Hardware Analysis, Software Architecture
0.7	2023-04-27	DOM,SUL, MAI	Software-Architektur als Komponentendiagramm, Sequenzdiagramme und erste Verhaltensmodellierung (FSMs) ergänzt
0.8	2023-05-11	DOM, SUL, MAI	Alle FSMs modelliert, Signalnamen für Lichtschranken geändert. Abgabe für Praktikumstermin 4
0.9	2023-05-31	DOM, SUL, MAI	FSMs abgeändert und Beschreibung der Funktionalitäten hinzugefügt
1.0	2023-06-14	DOM, SUL, MAI	FSMs, Event-Liste und Klassendiagramme gemäß aktueller Implementierung aktualisiert
1.1	2023-06-23	DOM, SUL, MAI	Praktikumstermin 6 ½: Technische Schulden, Implementierung, Testprotokolle und Abnahmetests aktualisiert

Inhaltsverzeichnis:

1	Teamorganisation.....	5
1.1	Verantwortlichkeiten	5
1.2	Absprachen.....	5
1.3	Repository-Konzept.....	5
2	Projektmanagement.....	6
2.1	Prozess.....	6
2.2	Projektplan	6
2.3	Risiken	7
2.4	Qualitätssicherung	8
3	Problemanalyse	9
3.1	Analyse des Kundenwunsches	9
3.1.1	Stakeholder	9
3.1.2	Systemkontext des Systems	10
3.1.3	Anforderungen	11
3.1.4	Use Cases / User Stories	15
3.1.5	Absprachen.....	18
3.2	Hardware: Analyse der technischen Gegebenheiten	20
3.2.1	Technischer Aufbau und Hardwarekomponenten.....	20
3.2.2	Werkstücke.....	21
3.2.3	Anforderungen aus dem Verhalten und technischen Besonderheiten	22
3.3	Softwareebene	23
3.3.1	Systemkontext der Software.....	23
3.3.2	Resultierende Anforderungen an die Software	23
3.3.3	Nachrichten und Signale	25
3.3.4	Sequenz Diagramm (Events)	28
3.3.5	Warnungen und Fehler	32
4	Software-Design	34
4.1	Software Architektur.....	34
4.1.1	Beschreibung der Komponenten	36
4.2	Software Struktur	37
4.2.1	Logic mit WorkpieceManager	37

4.2.2	HAL, EventManager, Watchdog und Logger	38
4.2.3	Configuration.....	39
4.3	Verhaltensmodellierung.....	40
4.3.1	MainFSM	40
4.3.2	MotorFSM	45
4.3.3	HeightSensorFSM	46
4.3.4	EStopFSM	47
4.3.5	ErrorFSM.....	48
4.3.6	ServiceModeFSM.....	49
5	Implementierung.....	50
5.1	Höhenmessung.....	50
5.2	EventManager	51
6	Qualitätssicherung	52
6.1	Teststrategie.....	52
6.2	Testszenarien/Abnahmetest	53
6.3	Testprotokolle und Auswertungen	61
6.3.1	Unit Tests und Integrationstest.....	61
6.3.2	Systemtests	62
7	Technische Schulden	64
8	Lessons Learned	67
9	Anhang	67
9.1	Glossar.....	67
9.2	Abkürzungen	67
9.3	Abbildungsverzeichnis.....	68

1 Teamorganisation

1.1 Verantwortlichkeiten

Verantwortlichkeit	Person/en
Projektmanager	Lorenz, Maik
Implementierung, Test	Schukow, Dominik
Requirements Analyst	Malik, Sulaiman

1.2 Absprachen

Dokumentation

- Source-Code auf [GitHub](#)
- Arbeitsversion RDT im SharePoint
- Abgabefertiges RDT im MS-Teams Raum für Gruppe 2.1

Kommunikation

- Feste Zeiten für Meetings im Labor: Donnerstag ab 12:00 Uhr bzw. nach dem Praktikum
- Freitag 13:00 Sprint-Planung / Standup
- Montag 19:00 online
- Meetings je nach aktuellen Themen in MS-Teams – Absprache über WhatsApp
- Besprechungsprotokolle werden in [Confluence](#) dokumentiert
- Anfragen und Absprachen über WhatsApp (max. Reaktionszeit 1 Stunde). Wenn jemand verhindert ist oder nicht weiterhelfen kann, wird das kommuniziert und ein Termin für eine Antwort genannt oder die Anfrage delegiert

Aufgabenverteilung

- Über das [Scrum-Board von JIRA](#) werden zu erledigende Aufgaben in Issues definiert. Es ist immer eine "Definition of Done" (vorweggenommener Endzustand) anzugeben, die spezifiziert wann eine Aufgabe als abgeschlossen gilt

1.3 Repository-Konzept

Der Source-Code liegt auf GitHub im Projekt [ESEP-2023SoSe-Team-2-1](#).

Wir arbeiten nach dem [GitFlow Workflow](#). Im main-Branch dürfen nur funktionsfähige Versionen liegen. Im develop-Branch ist der Arbeitsstand für die nächste Version. Neue Features werden in eigenen feature-Banches implementiert und danach in den develop-Branch gemerged. Auslieferungsbereite Versionen werden mit Versionsnummern getaggt.

2 Projektmanagement

2.1 Prozess

Da das Team nach dem Scrum-Modell arbeitet, werden zu bearbeitende Aufgaben immer in Sprints eingeplant. Sprints finden immer jeweils zwischen zwei Praktikumsterminen statt, dauern also in der Regel zwei Wochen.

Ein Review des gerade abgeschlossenen sowie die Planung eines neuen Sprints findet immer am Freitag 13:00 nach einem Praktikumstermin statt. In Wochen ohne Praktikum wird dieser Termin dazu genutzt, um den Stand der zu bearbeitenden Aufgaben des aktuellen Sprints zu besprechen.

Besprechungen werden immer schriftlich in Confluence dokumentiert. Sich daraus ergebende Absprachen werden ebenso dokumentiert und falls notwendig direkt in JIRA-Tasks eingeplant.

Wichtige Absprachen mit den Betreuern werden in diesem Dokument festgehalten.

2.2 Projektplan

Meilensteine

Zeitpunkt	Ziele
Praktikum 1	Organisation innerhalb des Teams definiert. Anforderungsanalyse und Systemkontextdiagramm erstellt. Projektplan und Projektstruktur erstellt. Momentics und Repository ist eingerichtet. Ein Programm kann auf die Anlage geladen werden und diese ansteuern
Praktikum 2	Vollständige Anforderungsanalyse und Abmachungen sind dokumentiert. Die Aktorik der HAL ist implementiert. Beispiel zur Datenübertragung via QNET ist implementiert. Abnahmetests sind formuliert. Erstes Dokument der Software Architektur ist ausgearbeitet.
Praktikum 3	Überarbeitetes Dokument mit dem Entwurf der Software Architektur liegt vor. FSMs sind grob modelliert. Die Sensorik der HAL ist implementiert. Konzept der Übergabe der Daten von HAL zu FSM liegt vor. Präsentation der Architektur als Vortrag.

Praktikum 4	Das Dokument der Software Architektur ist final und kann implementiert werden.
Praktikum 5	FSMs sind ohne Fehlerbehandlung modelliert. Grundfunktionalität ohne Fehlerbehandlung implementiert (Werkstücke können sortiert werden)
Praktikum 6	FSMs sind vollständig modelliert. Die Anlage ist vollständig implementiert. Abgabe des finalen Requirement Design Dokument.
Praktikum 7	Alle nicht realisierten Funktionalitäten sind dokumentiert und begründet. Gesamtanlage ist bereit für die Abnahmetests durch den Kunden. Fehlerzustände sind dokumentiert. "Lessons Learned" ausgefüllt. Abgabe von Dokumenten, Planung, Code und Protokollen

Zur Visualisierung des Projektplans wird die [Jira Roadmap](#) verwendet. Das Projekt ist in verschiedene Abschnitte aufgeteilt, die hier auf der oberen Ebene mittels sogenannter "Epics" dargestellt werden (vgl. Gantt-Chart). Der Name jedes Epics ist als vorweggenommener Endzustand formuliert, damit auf den ersten Blick klar ist, was das Ziel ist. Jedes Epic wird auf User Stories und Aufgaben herunter gebrochen, die erledigt werden müssen für die Erreichung des (Teil-)Ziels. Ziel ist es eine Granularität zu schaffen, damit Aufgaben möglichst unabhängig voneinander bearbeitet und somit gut auf die Teammitglieder verteilt werden können.

2.3 Risiken

Risikobeschreibung	Hypothetisch / bekannt	Wahrscheinlichkeit	Maßnahme
Teammitglied bricht das Praktikum ab	bekannt	gering	Absprache mit Kunde über wegfallende Requirements
Teammitglied ist krank / nicht verfügbar	bekannt	normal	Gute Dokumentation, Verteilung der Aufgaben an andere Teammitglieder
Kein Zugang zum Labor	bekannt	normal	Nutzung der Simulation zum Testen der Software
Verzug durch technische Schwierigkeiten	bekannt	normal	Technische Beratung anfragen bei Profs.

2.4 Qualitätssicherung

Um die Qualität der umgesetzten Features sicherzustellen, werden Unit- und Modultests mit der GoogleTest Suite erstellt. Vor dem Abschluss von Feature-Branches müssen alle Tests bestanden werden.

Mit den Abnahmetests wird die korrekte Funktion des Gesamtsystems aus Kundensicht getestet.

3 Problemanalyse

3.1 Analyse des Kundenwunsches

Der Kunde stellt ein fertiges System zur Verfügung, das so programmiert werden soll, dass aufgelegte Werkstücke am Ende eines FBM in vorgegebener Reihenfolge ankommen sollen. Zur Ermittlung der Werkstücktypen können deren Eigenschaften durch an der Anlage montierte Sensoren bestimmt werden. Werkstücke, die nicht in die Reihenfolge passen, sollen auf Rutschen aussortiert werden.

3.1.1 Stakeholder

Stakeholder	Interessen
Kunde	Fehlerfreie und richtig sortierte Bausteine.
Entwickler	Fehlerfreie und termingerechte Implementierung der Software
Anwender	Benutzerfreundliche Sortieranlage, welche vollautomatisch nach auflegen eines Werkstückes läuft
Tester	Testen / Wartung der Software

3.1.2 Systemkontext des Systems

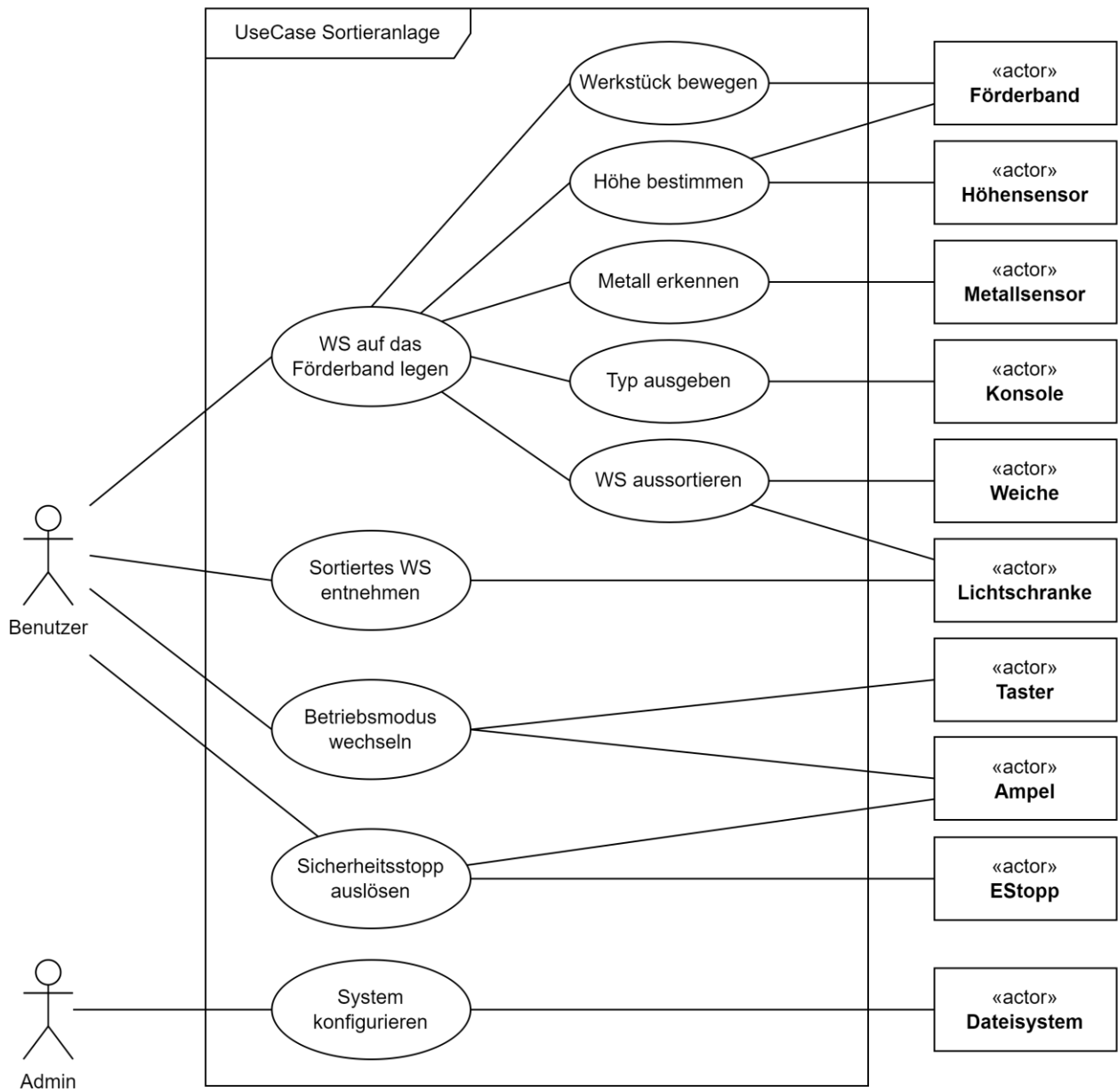


Figure 1 Use_Case_diagram

3.1.3 Anforderungen

Lfd. Nr. / ID	Beschreibung	Fußnote
ANF01	Auf der Anlage sollen die Werkstücke in folgender vorgegebener Reihenfolge sortiert werden: <Type A> → <Type B> → <Type C>	12
ANF02	Die Sortier-Reihenfolge soll aus allen nicht-binären Werkstücken über eine auf dem System abgelegten Datei konfigurierbar sein. Die Konfigurationsdatei soll benutzerfreundlich gestaltet sein.	15, 43, 44
ANF03	Flache Werkstücke werden von FBM1 erkannt und aussortiert, sofern sie nicht der Konfiguration entsprechen oder die Rutsche an FBM1 voll ist.	17
ANF04	Werkstücke, die nicht der vorgegebenen Reihenfolge entsprechen, werden spätestens an FBM2 aussortiert.	18
ANF05	Auf dem FBM1 können sich mehrere Werkstücke befinden.	22
ANF06	Auf dem FBM2 darf sich maximal 1 Werkstück befinden. Die Übergabe von FBM1 an FBM2 erfolgt daher auch vereinzelt.	23
ANF07	Auf der Anlage (beide FBM) sollen die Werkstücke langsam durch die Höhenmessung transportiert werden	25
ANF08	Es darf kein Werkstück von der Anlage fallen.	26
ANF09	Sind beide Rutschen voll, läuft der Sortierbetrieb so lange weiter, bis eine Aussortierung eines Werkstückes nicht mehr erfolgen kann.	28
ANF10	Ist die Rutsche auf FBM1 voll, so soll die Aussortierung über FBM2 erfolgen	38
ANF11	Ist die Rutsche auf FBM2 voll, so soll die Aussortierung über FBM1 erfolgen.	39
ANF12	Wenn sich auf FBM1 kein Werkstück befindet, soll FBM1 anhalten.	37
ANF13	Wenn sich auf FBM2 kein Werkstück befindet, soll FBM2 anhalten.	37

ANF14	<p>Wenn ein Werkstück das Ende von FBM 2 erreicht, werden folgende Daten auf der Konsole ausgegeben:</p> <ul style="list-style-type: none"> • Werkstück-ID • Werkstück-Typ • Mittlerer Höhenmesswert aus der Mitte des Werkstücks von FBM1 in Millimeter • Mittlerer Höhenmesswert aus der Mitte des Werkstücks von FBM2 in Millimeter • Überschlagen (ja/nein) 	29-34
ANF15	Eine volle Rutsche ist an der entsprechenden Anlage zu signalisieren.	27
ANF16	An einem FBM können unterschiedliche Varianten der Sortiermechanik eingebaut sein (Auswerfer oder Weiche), die beide vom System unterstützt werden müssen	45-55
ANF17	Die Lichtschranke an der Höhenmessung darf nicht im Betriebsmodus verwendet werden.	58
ANF18	<p>Auf die Anlage werden mehrere Typen von WS unterschiedlich auf das Band gelegt, die alle erkannt werden müssen:</p> <ul style="list-style-type: none"> • Flache WS • Hohe WS mit Bohrung und Metall • Hohe WS mit Bohrung ohne Metall • Hohe WS ohne Bohrung <p>Alle anderen Werkstücke werden als "Unbekannt" eingeordnet.</p>	2-11
ANF19	Die sortierten WS am Ende von FBM2 müssen so bereitgestellt werden, dass ein Pick-and-Place Roboter sie entnehmen kann.	14
ANF20	Wenn die Lichtschranke am Anfang von FBM1 frei ist, können neue WS hier eingelegt werden.	19-21
ANF21	Eine Änderung der Aussortierung bedingt durch eine volle Rutsche (siehe ANF10 und ANF11) ist dem Bediener zu signalisieren.	40
ANF22	Es soll ein möglichst hoher Durchsatz an Werkstücken erreicht werden	42

ANF23	Der Betrieb der Anlage soll jederzeit sicher sein und darf keine Gefährdung des Bedieners hervorrufen	59
ANF24	<p>Folgende Fehlerzustände sollen erfasst und dem Bediener signalisiert werden:</p> <ul style="list-style-type: none"> • Beide Rutschen voll und ein notwendiges Aussortieren ist nicht mehr möglich <p>Die weiteren Fehlerfälle werden nach Absprache nicht umgesetzt</p>	62
ANF25	Durch Drücken des "Start"-Tasters (< 2sec) wechselt die Anlage in den Betriebszustand, durch langes (>= 2sec) Drücken in den Service-Mode.	70-71
ANF26	Durch Drücken des "Stop"-Tasters (< 2sec) wechselt die Anlage in den Ruhezustand (Wenn keine Fehler oder Warnungen vorliegen).	72-73
ANF27	Durch Drücken eines "E-Stopp" Schalters, steht die ganze Anlage (beide FBM!) still. Sind beide E-Stopp Schalter herausgezogen, bleibt die Anlage weiterhin so lange stehen, bis der Reset-Taster gedrückt wurde.	75-77
ANF28	Dem Benutzer sollen sinnvolle Hinweise zur Bedienung der Anlage angezeigt werden durch Nutzung der LEDs an den Tastern oder anderer Anzeigeelemente.	78
ANF29	Die grüne Ampel soll im Betriebsmodus dauerhaft leuchten. Im Service-Mode soll sie grün blinken.	80-81
ANF30	Die gelbe Ampel soll bei anliegenden Warnungen blinken.	82
ANF31	<p>Die rote Ampel soll anliegende Fehler wie folgt anzeigen:</p> <ul style="list-style-type: none"> • Anstehend unquittiert: Schnelles Blinken (1 Hz) • Anstehend quittiert: Dauerhaftes Leuchten • Gegangen unquittiert: Langsames Blinken (0,5 Hz) 	95-99

ANF32	<p>Mit anliegenden Fehlern soll wie folgt umgegangen werden:</p> <ul style="list-style-type: none"> • Neu aufgetreten: Anstehend unquittiert • Fehler von Bediener durch Drücken der "Reset"-Taster quittiert: Anstehend quittiert • Anstehend quittiert und Fehler behoben: Wechsel in den Zustand "OK / Kein Fehler" • Automatisch behobene Fehler: Wechsel in den Zustand "Gegangen unquittiert". Dieser wird durch quittieren verlassen 	83-93
-------	---	-------

3.1.4 Use Cases / User Stories

ID	UC-01
Titel	E-Stopp Funktion auslösen
Beschreibung	Das Use Case beschreibt die Funktionsweise des E-stopp Schalters.
Akteure	Anlage (Lamp, E-Stopp, ButtonReset), Benutzer.
Vorbedingung	<ul style="list-style-type: none"> Die Anlage ist im Betriebsmodus
Hauptszenario	<ol style="list-style-type: none"> Ein E-Stopp Schalter an FBM1 oder FBM2 wird gedrückt. Die Anlage stoppt. Der E-Stopp Schalter wird herausgezogen. Vorderband wird ausgeräumt. Reset-Button an FBM1 und FBM2 wird betätigt.
Nachbedingung	Die Anlage ist im Betriebsmodus.

ID	UC-02
Titel	Geflippte Werkstücke erkennen
Beschreibung	Das Use Case beschreibt die Funktion der Anlage, wenn ein Werkstück bei Übergabe zwischen beiden FBM geflippt wird.
Akteure	FBM2 (LBA, LBW, LBE, LBR, Motor, Weiche), Console.
Vorbedingung	<ul style="list-style-type: none"> Anlage ist im Betriebsmodus. Am Ende des FBM2 soll ein Werkstück mit Bohrung kommen. Werkstück mit BoM wird am Anfang des FBM1, so dass die Bohrung nach unten zeigt. Das Werkstück wird bei Übergabe von FBM 1 auf FBM 2 geflippt. Das Werkstück muss sortiert werden.
Hauptszenario	<ol style="list-style-type: none"> FBM2 erkennt, dass das Werkstück geflippt wurde Das Werkstück wird nicht aussortiert Das Werkstück wird zum Ende des FBM2 transportiert. Die Daten des Werkstücks werden auf der Konsole mit, Überschlagen“ zusätzlich angezeigt.
Nachbedingung	<ul style="list-style-type: none"> Anlage ist weiter im Betriebsmodus. Kein Werkstück befindet sich auf FBM2

ID	UC-03
Titel	Rutsche 1 voll erkennen.
Beschreibung	Das Use Case beschreibt die Funktion der Anlage, wenn die Rutsche auf FBM 1 voll ist.
Akteure	Anlage (LBA, LBW, LBE, LBR, Motor, MD, HM, Weiche)
Vorbedingung	<ul style="list-style-type: none"> Anlage ist im Betriebsmodus. Rutsche 1 ist voll. Rutsche 2 ist leer. Ein flaches Werkstück befindet sich am MD des FBM1.
Hauptszenario	<ol style="list-style-type: none"> System erkennt, dass das Werkstück aussortiert werden muss. Das Werkstück wird zum FBM 2 transportiert. Das Werkstück wird in Rutsche auf FBM 2 aussortiert.
Nachbedingung	Das Werkstück liegt in der Rutsche von FBM2

ID	UC-04
Titel	Rutsche 2 voll erkennen
Beschreibung	Das Use Case beschreibt die Funktion der Anlage, wenn die Rutsche auf FBM 2 voll ist.
Akteure	Anlage (LBA, LBW, LBE, LBR, Motor, HM, MD, Weiche)
Vorbedingung	<ul style="list-style-type: none"> Anlage ist im Betriebsmodus. Rutsche 2 ist voll. Rutsche 1 ist leer. Das (nicht flaches) Werkstück befindet sich am MD des FBM1 und passt nicht in die Sortierreihenfolge.
Hauptszenario	<ol style="list-style-type: none"> System erkennt, dass das Werkstück aussortiert werden muss. Das Werkstück wird in Rutsche auf FBM 1 aussortiert
Nachbedingung	Das Werkstück liegt in der Rutsche von FBM1

ID	UC-05
Titel	Beide Rutschen voll erkennen
Beschreibung	Das Use Case beschreibt die Fehlerbehandlung, wenn beide Rutschen voll sind und ein Werkstück aussortiert wird.

Akteure	Anlage (LBA, LBW, LBE, LBR, Motor, Weiche, ButtonReset, ButtonStart, Lamp).
Vorbedingung	<ul style="list-style-type: none"> Anlage ist im Betriebsmodus. Beide Rutschen sind voll. Ein Werkstück muss aussortiert werden
Hauptszenario	<ol style="list-style-type: none"> Grüne Lampe aus. Die Anlage stoppt. Rote Lampe blinkt (1Hz). Benutzer drückt Button-Reset(<= 2 Sek.). Rote Lampe leuchtet dauerhaft. Benutzer behebt den Fehler. Benutzer drückt Button-Start (<=2 Sek.). Rote Lampe aus. Grüne Lampe an.
Nachbedingung	Anlage ist im Betriebsmodus

ID	UC-06
Titel	Sortiervorgang starten
Beschreibung	Das Use Case beschreibt den Sortiervorgang der Anlage.
Akteure	Anlage (LBA, LBW, LBE, LBR, Motor, Weiche, HM, MD), Benutzer
Vorbedingung	<ul style="list-style-type: none"> Anlage ist im Betriebsmodus. Beide Rutsche sind leer. Die von Anlage angenommene Reihenfolge ist: WS-F → WS-BOM → WS-BM. Reihenfolge der Werkstücke auf Anlage: <ul style="list-style-type: none"> WS-BOM → WS-F → WS-F → WS-BOM → Werkstück BuM → WS-BOM. Die Werkstücke werden nacheinander am Anfang des FBM1 gelegt.
Hauptszenario	<ol style="list-style-type: none"> WS-BOM wird in Rutsche auf FBM 1 aussortiert. WS-F wird zum Ende transportiert. WS-F wird in Rutsche auf FBM 1 aussortiert. WS-BOM wird zum Ende transportiert. WS-BM wird zum Ende transportiert. WS-BOM wird in Rutsche auf FBM 1 aussortiert.
Nachbedingung	<ul style="list-style-type: none"> 2 Werkstück BoM sind in Rutsche auf FBM 1. 1 WS-F ist in Rutsche auf FBM 2. 1* WS-F → 1* WS-BOM → 1* WS-BM wurden zum Ende des FBM 2 transportiert

ID	UC-07
Titel	<i>Sortiervorgang starten für ein Werkstück</i>
Beschreibung	<i>Das Use Case beschreibt den Sortiervorgang der Anlage für ein Werkstück.</i>
Akteure	<i>Anlage (LBA, LBW, LBE, LBR, Motor, Weiche, HM. MD), Benutzer</i>
Vorbedingung	<ul style="list-style-type: none"> • <i>Anlage ist im Betriebsmodus.</i> • <i>Beide Rutschen sind leer.</i> • <i>Ein Werkstück wird am Anfang des FBM1 gelegt oder von FBM1 an FBM2 übergeben.</i>
Hauptszenario	<ol style="list-style-type: none"> 1. <i>Das Werkstück unterbricht LBA</i> 2. <i>Das Werkstück wird zur HD transportiert</i> 3. <i>Das Werkstück wird zur MD transportiert</i> 4. <i>Das Werkstück wird erkannt, ob es aussortiert werden muss</i> 5. <i>Das Werkstück unterbricht LBW</i> 6. <i>Das Werkstück muss nicht aussortiert werden</i> 7. <i>Das Werkstück unterbricht LBE</i>
Nachbedingung	<p><i>6a) Das Werkstück muss aussortiert werden</i></p> <p><i>7a) Das Werkstück wird von Weiche aussortiert</i></p> <p><i>8a) Das Werkstück unterbricht LBR</i></p>

ID	UC-08
Titel	
Beschreibung	
Akteure	
Vorbedingung	
Hauptszenario	
Nachbedingung	

3.1.5 Absprachen

Nachfolgend sind die Absprachen ausgeführt, die mit dem Kunden getroffen wurden bzgl. der Anforderungen:

ID	Absprache
----	-----------

ASP-01	Wenn auf FB1 ein WS als "ungültig" erkannt wird (passt nicht zu der gewünschten Sortier-Reihenfolge), aber auf FB2 als "gültig" oder umgekehrt, soll immer aufgrund des an FBM2 erkannten WS-Typs entschieden werden, ob das WS aussortiert oder durchgelassen wird.
ASP-02	<p>Vom Kunden wurde folgender Vorschlag gemacht: "Wenn ein WS as FBM2 aussortiert werden muss, die Rutsche jedoch voll ist: Können wir nicht zurück fahren und an FB1 aussortieren, wenn frei?"</p> <p>Dieses Feature wird nach Absprache nicht umgesetzt aufgrund hohem Programmieraufwand und kein Produktivitätsgewinn. In diesem Fall erfolgt eine Fehlermeldung und Bandstopp.</p>
ASP-03	Der Höhenmesswert an FBM2 soll als "maximale Höhe in Millimeter" ausgegeben werden. Höhenangaben sollen generell immer in Millimeter erfolgen.
ASP-04	Werkstücke sollen bevorzugt an FBM2 aussortiert werden, da sich sonst alle schnell in der Rutsche stauen. Ausnahme: Aussortierung an FBM1 ist fest vorgegeben (flache WS).
ASP-05	Der Anfang von FBM1, an dem neue Werkstücke aufgelegt werden, ist definiert durch den Bereich in dem die erste Lichtschranke unterbrochen wird.
ASP-06	<p>Die Lichtschranke der Höhenmessung wird zum Abnahmetest abgeschaltet.</p> <p>In unserem Prototypen ist diese noch vorhanden, im Produktivsystem wird die Hardware jedoch eingespart, da der Systemarchitekt der Meinung ist, das bekommen die Softwerker auch ohne hin.</p>
ASP-07	Eine volle Rutsche wird am jeweiligen FBM als Warnung signalisiert (gelbe Ampel blinkt).
ASP-08	Da die Bewertung, ob ein WS der gewünschten Reihenfolge entspricht, auf Grundlage von FBM2 erfolgt (siehe ASP-01), wird dieser WS-Typ auch als "finaler Typ" ausgegeben. Zusätzlich wird angegeben, dass sich das WS überschlagen hat.
ASP-09	Wenn eine Rutsche voll ist und die Aussortierung deshalb am jeweils anderen FBM erfolgt, wird diese Situation neben dem Blinken der gelben Ampel an der vollen Rutsche (Warnung) zusätzlich über das Leuchten von Q1 (Aussortierung an FBM1) bzw. Q2 (Aussortierung an FBM2) signalisiert.

3.2 Hardware: Analyse der technischen Gegebenheiten

3.2.1 Technischer Aufbau und Hardwarekomponenten

Die Festo-Transfersysteme verfügen über unterschiedliche Sensoren und Aktoren, mit dessen Hilfe unsere Sortieranlage realisiert wird.

Sensorik:

Für die Registrierung der Position von WS werden Lichtschranken verwendet. Damit Entscheidungen zur Sortierung anhand der Höhe von WS getroffen werden können, werden Höhsensoren verwendet. Um das Metall in WS zu erkennen wird ein Metallsensor verwendet. Zudem befindet sich an einem Transfersystem ein Bedienfeld mit den Tasten Start, Stop und Reset. Rechts daneben befindet sich der E-Stopp Schalter. Es handelt sich um einen üblichen E-Stopp-Schalter, der bei Betätigung einklinkt. Durch Herausziehen wird er wieder in seine ursprüngliche Position versetzt. Hierbei handelt es sich nicht um einen klassischen E-Stopp Schalter, der die Anlage stromlos setzt. Der sichere Betrieb nach Betätigung des Schalters muss deshalb über die Steuerungssoftware realisiert werden.

Aktorik:

Die Förderbandmodule werden durch einen eigenen Steuerungscomputer gesteuert mit einem Beaglebone Black. Die beiden Computer sind über Ethernet gekoppelt.

Das FB lässt sich durch entsprechende Ansteuerung des Motors in langsamer und schneller Geschwindigkeit und dazu nach links oder rechts bewegen.

Zur Aussortierung von WS in eine Rutsche existieren zwei Varianten des Hardwareaufbaus. Die eine Variante verfügt über eine Weiche, die zweite Variante über einen Kicker. Die Weiche ist im stromlosen Zustand geschlossen. Beim Öffnen fließt Strom durch die Magnetspule, die die Weiche betätigt. Der Kicker lässt im stromlosen Zustand WS passieren. Fließt Strom durch den Kicker, so fährt dieser aus und drückt ein Werkstück aktiv in die Rutsche. Im ausgefahrenen Zustand können keine WS passieren und werden auch nicht in die Rutsche befördert.

Um den Benutzer über den Zustand des Systems zu informieren, verfügt ein Transfersystem über eine Konsolenausgabe (mit verbundener Momentics IDE über "stdout") und für die zusätzliche Visualisierung über eine Ampel. Die Ampel kann die Farben grün, orange und rot darstellen.

Darüber hinaus verfügt das Bedienfeld über Status-LED's (jeweils an den Tastern Start und Reset sowie Q1 und Q2).

Das FB hat eine Länge von 700 mm mit einer Tiefe von 70 mm. Der Abstand zu den äußeren Förderbandbegrenzungen beträgt 5 mm. Somit beträgt die effektive nutzbare Gesamthöhe des FB 80 mm. Die am Transfersystem angebrachte Rutsche ist 180 mm lang und 70 mm breit. Sie weist eine Verengung von 30 mm an der zulaufenden Seite vom FB auf.

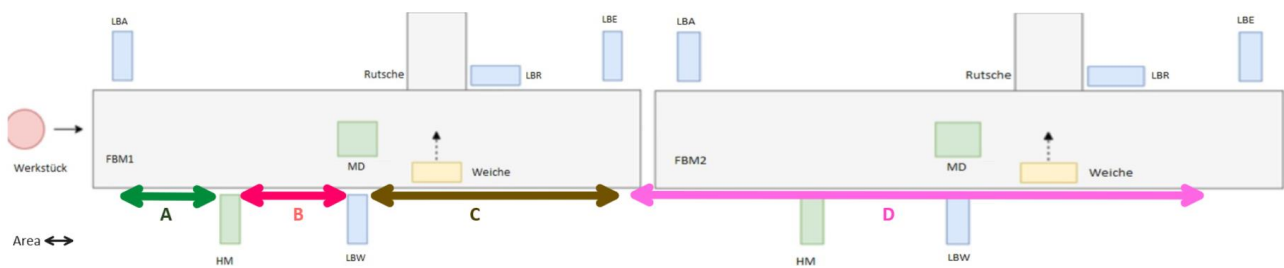
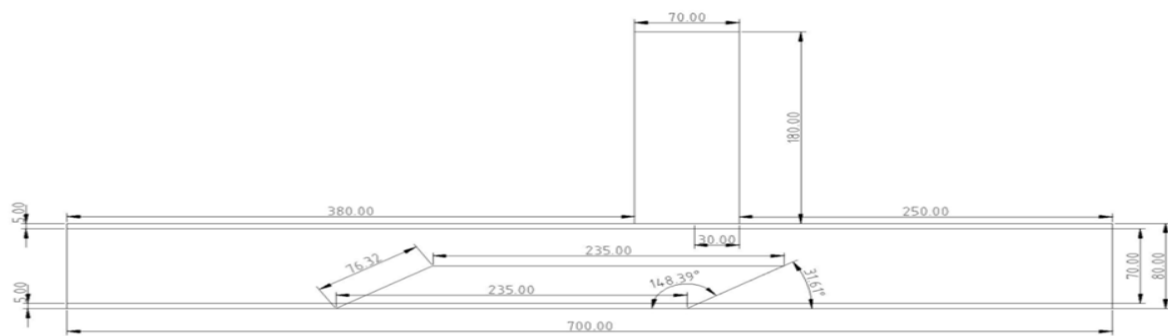


Figure 2 Hardware Darstellung der Sortieranlage



Angaben in mm

Figure 3 Veranschaulichung der Messwerte (Quelle: <https://reposit.haw-hamburg.de/handle/20.500.12738/8463>)

Beaglebone Black:

Bei diesem Transfersystem kommt ein Beaglebone Black mit bereits installiertem QNX Neutrino zum Einsatz. Auf dem Beaglebone Black sitzt ein AM3358/9-SoC von TI, der ein ARM Cortex-A8-Prozessor implementiert. Für die Installation bietet QNX ein [BSE \(Board Support Package\) an](#). Dazu ist es ratsam den entsprechenden User Guide zu [suchen und zu finden](#). Alternativ ist eine Virtualisierung auf zwei VMware-VMs verfügbar. Die korrekte Funktionsweise des Beaglebones wird über ein blaues Lauflicht auf der linken Seite des Frontpanels signalisiert.

3.2.2 Werkstücke

Vom Nutzer der Anlage werden verschiedenartige Werkstücke auf das FB gelegt:

Typ ID	Beschreibung	Maße	Besonderheiten / zu beachten
WS-F	Flache WS		
WS-BM	Hohe WS mit Bohrung und Metalleinsatz		Wird das WS gedreht, erhält man den Typ WS-OB
WS-BOM	Hohe WS mit Bohrung ohne Metalleinsatz		Wird das WS gedreht, erhält man den Typ WS-OB
WS-OB	Hohe WS ohne Bohrung		
WS-BIN	Binär-codierte WS		Die WS verfügen über Rillen. Während der Höhenmessung sind also schnell wechselnde

			<p>Messwerte zu erwarten.</p> <p>Zur Erkennung der Binärcodierung ist auch entscheidend, ab wann die Messung gestartet wird (Mitte bis Ende oder Anfang bis Mitte des WS)</p>
--	--	--	---

3.2.3 Anforderungen aus dem Verhalten und technischen Besonderheiten

Lfd. Nr. / ID	Beschreibung
HW_REQ_001	Das unterschiedliche Gewicht und je nach Anzahl der auf dem FB befindlichen WS kann dies die Motorgeschwindigkeit beeinflussen. Dies muss bei Funktionen berücksichtigt werden, die Annahmen bzgl. der aktuellen Motorgeschwindigkeit treffen.
HW_REQ_002	Binär codierte WS haben die gleiche Höhe wie hohe WS. Diese müssen voneinander unterschieden werden können.
HW_REQ_003	Die Lichtschranke, die erkennt, wenn sich ein WS unter dem Höhenmesser befindet, steht im Produktivbetrieb nicht zur Verfügung. Die Erkennung der Präsenz von Werkstücken unter der Höhenmessung muss deshalb anders erfolgen.

3.3 Softwareebene

3.3.1 Systemkontext der Software

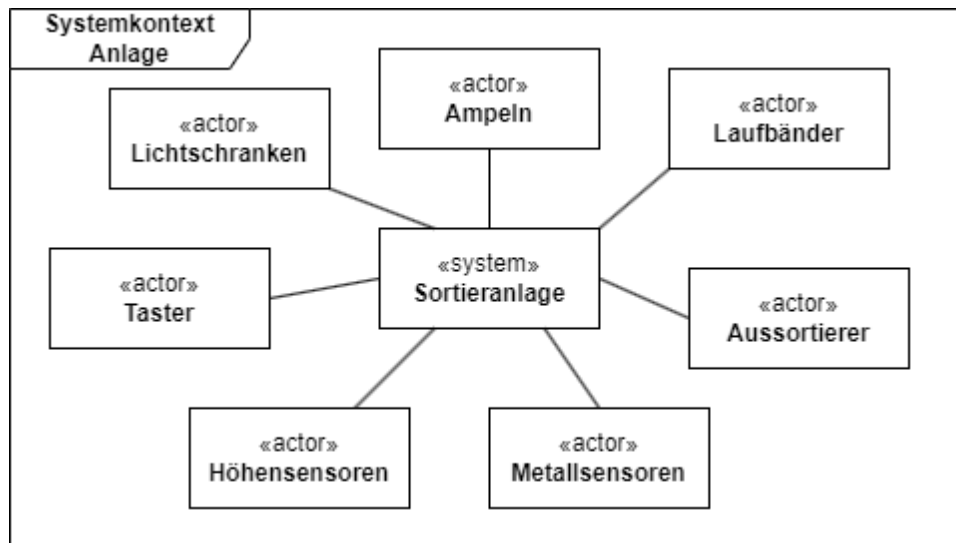


Figure 4 System Context diagramm

3.3.2 Resultierende Anforderungen an die Software

Lfd. Nr. / ID	Beschreibung
REQS_001	Beim Auflegen eines WS am Anfang von FBM1 wird der Motor aktiviert und das FB bewegt sich.
REQS_002	Wenn ein aufgelegtes WS den Bereich der Höhenmessung erreicht, bewegt sich der Motor am jeweiligen FBM in langsamer Geschwindigkeit. Beim Verlassen des Bereichs bewegt sich der Motor wieder in schneller Geschwindigkeit.
REQS_003	Ein flaches WS, das nicht der gewünschten Reihenfolge entspricht, wird an FBM1 aussortiert. Bei voller Rutsche wird das WS durchgelassen und an FBM2 aussortiert. Ein nicht-flaches WS, das nicht der gewünschten Reihenfolge entspricht, wird an FBM2 übergeben und dort aussortiert. Bei voller Rutsche an FBM2 erfolgt die Aussortierung an FBM1. Bei gewünschter Aussortierung und falls beide Rutschen voll sind, erfolgt ein Bandstopp und Fehlermeldung an beiden Anlagen.
REQS_004	Bei voller Rutsche erfolgt eine Warnmeldung. Hierbei blinkt die gelbe Ampel am betroffenen FBM und es wird eine Warnung auf der Konsole ausgegeben mit der Aufforderung, die Rutsche freizumachen.
REQS_005	Bei der Übergabe eines WS von FBM1 an FBM2 (LS am Anfang von FBM2 unterbrochen) wird der Motor an FBM2 aktiviert und das FB bewegt sich.

REQS_006	<p>Wenn ein WS das Ende von FBM2 erreicht, stoppt der Motor. Der Motor kann erst wieder anlaufen, wenn die LB wieder frei ist.</p> <p>Der Motor stoppt auch, wenn ein WS aussortiert wurde.</p>
REQS_007	Beim Drücken des E-Stopp muss ein sofortiger Bandstopp beider FBM erfolgen.
REQS_008	<p>Im Ruhezustand wechselt die Anlage durch das kurze Drücken des Start-Tasters vom Ruhezustand in den Betriebszustand.</p> <p>Im Ruhezustand wechselt die Anlage durch das lange Drücken des Start-Tasters vom Ruhezustand in den Service-Mode.</p>
REQS_009	Im Betriebszustand wechselt die Anlage durch das Drücken des Stopptasters in den Ruhezustand. Dieser Wechsel ist nur möglich, wenn keine Fehler oder Warnungen vorliegen.
REQS_010	<p>Im Betriebszustand wechselt die Anlage beim Auftreten eines Fehlers, der sich nicht von selbst beheben lässt, in den Fehlermodus, der sich wie folgt verhält:</p> <ol style="list-style-type: none"> 1. Rote Ampel blinkt schnell (1 Hz) 2. Drücken des Reset-Tasters: Rote Ampel leuchtet dauerhaft 3. Drücken des Start-Tasters: Wechsel in den Betriebszustand <p>Im Betriebszustand wechselt die Anlage beim Auftreten eines Fehlers, der sich von selbst behebt, in den Fehlermodus. Wenn er sich danach von selbst behebt, verhält sich die Anlage wie folgt:</p> <ol style="list-style-type: none"> 1. Rote Ampel blinkt langsam (0,5 Hz) 2. Drücken des Start-Tasters: Wechsel in den Betriebszustand
REQS_011	<p>Das System muss folgende Konfiguration ermöglichen, die beim Programmstart aus einer Konfigurationsdatei eingelesen wird:</p> <ul style="list-style-type: none"> • Modus: Master oder Slave • Sortiermechanik: Weiche oder Auswerfer • Gewünschte Sortierreihenfolge durch Einlesen einer Konfigurationsdatei auf dem Master-System <p>Wird eine ungültige oder fehlende Konfiguration festgestellt, erfolgt eine Fehlermeldung und ein Wechsel in den Betriebszustand ist nicht möglich.</p>

REQS_012	<p>Wenn ein WS an der Weiche von FBM1 angekommen ist, werden folgende Informationen auf der Konsole ausgegeben:</p> <ul style="list-style-type: none"> • WS-ID • WS-Typ • Mittlere Höhe in Millimetern an FBM1 <p>Wenn ein WS an der Weiche von FBM2 angekommen ist, werden folgende Informationen auf der Konsole ausgegeben:</p> <ul style="list-style-type: none"> • WS-ID • WS-Typ • Mittlere Höhe in Millimetern an FBM1 • Maximale Höhe in Millimetern an FBM2 • WS hat sich überschlagen: ja/nein <p>An beiden FBM erfolgt zusätzlich die Ausgabe, ob das WS gültig ist (der gewünschten Reihenfolge entspricht) oder eine Aussortierung erfolgt.</p>
----------	--

3.3.3 Nachrichten und Signale

Anmerkung: Viele Signale können an Master und auch an Slave auftreten. Um die folgenden Tabellen zu vereinfachen, wurde deshalb ein X als Platzhalter innerhalb des Signalnamens verwendet, das für den Systemtyp steht (Events jeweils 1x mit M für Master, 1x mit S für Slave).

HAL

Folgende Events werden jeweils von der HAL FBM1 (Master) und FBM2 (Slave) an die Logic (Master) über den EventManager gesendet.

Name	Beschreibung
ESTOP_X_PRESSED	E-Stopp an FBM1/2 wurde gedrückt
ESTOP_X_RELEASED	E-Stopp an FBM1/2 wurde gelöst
START_X_SHORT	An einer der beiden Anlagen wurde die Start-Taste gedrückt (<=2 sec)
START_X_LONG	An einer der beiden Anlagen wurde die Start-Taste gedrückt (>2 sec)
STOP_X_SHORT	An einer der beiden Anlagen wurde die Stop-Taste gedrückt (<=2 sec)

RESET_X _ SHORT	An einer der beiden Anlagen wurde die Reset -Taste gedrückt (<=2 sec)
RESET_X _ LONG	An einer der beiden Anlagen wurde die Reset-Taste gedrückt (>2 sec)
LBA_X_BLOCKED	LB Start FBM1 unterbrochen
LBA_X_UNBLOCKED	LB Start FBM1 wieder frei
LBW_X_BLOCKED	LB an der Weiche unterbrochen
LBW_X_UNBLOCKED	LB an der Weiche wieder frei
LBE_X_BLOCKED	LB Ende FBM1 unterbrochen
LBE_X_UNBLOCKED	LB Ende FBM1 wieder frei
LBR_X_BLOCKED	LB oben an der Rutsche unterbrochen
LBR_X_UNBLOCKED	LB oben an der Rutsche wieder frei
MD_X_PAYLOAD	Payload= ist es ein Metal (True oder False)
HM_X_PAYLOAD	Payload= die Height des WS in mm
HM_X_WS_F	An der Höhenmessung wurde ein Werkstück vom Typ "Flach" erkannt
HM_X_WS_OB	An der Höhenmessung wurde ein Werkstück vom Typ "Hoch ohne Bohrung" erkannt
HM_X_WS_BOM	An der Höhenmessung wurde ein Werkstück vom Typ "Bohrung ohne Metall" erkannt
HM_X_WS_UNKNOWN	An der Höhenmessung wurde ein unbekanntes Werkstück erkannt

Höhensensor

Folgende Events werden jeweils von der Logic (Master) an den Höhsensor gesendet:

Name	Beschreibung
HM_X_CAL_OFFSET	Führt eine Kalibrierung des Offsets durch (Laufbandhöhe = 0,0 mm)
HM_X_CAL_REF	Führt eine Kalibrierung des Referenzwertes durch (Höhe eines hohen Werkstücks = 25,0 mm).

Folgende Events werden jeweils von Logic (Master) an HAL FBM1 und FBM2 über Eventmanager gesendet.

Name	Beschreibung
MOTOR_X_FAST	Startet den Motor mit Normaler Geschwindigkeit.
MOTOR_X_SLOW	Startet den Motor mit Langsamer Geschwindigkeit.
MOTOR_X_STOP	Hält den Motor an.
LAMP_X_GREEN	Schaltet den Zustand der grünen Lampe: 0: aus 1: an 2: blinkt langsam
LAMP_X_YELLOW	Schaltet den Zustand der gelben Lampe: 0: aus 1: an 2: blinkt langsam
LAMP_X_RED	Schaltet den Zustand der roten Lampe: 0: aus 1: an 2: blinkt langsam 3: blinkt schnell
LED_X_Q1	Schaltet den Zustand der LED Q1: 0: aus 1: an
LED_X_Q2	Schaltet den Zustand der LED Q2: 0: aus 1: an
LED_X_START	Schaltet den Zustand der LED am Start-Taster: 0: aus 1: an
LED_X_RESET	Schaltet den Zustand der LED am Reset-Taster: 0: aus 1: an
SORT_X_OUT	Sortiert das Werkstück aus: 0: passieren lassen (Weiche öffnen) 1: aussortieren (Pusher kurz ausfahren)

Die MotorFSM existiert zur Laufzeit zwei Mal, einmal für Steuerung von FBM1 und einmal für FBM2. Folgende Events werden an die entsprechende MotorFSM gesendet, um die Flags "Start", "Stop" und "Fast" zu setzen bzw. rückzusetzen.

Name	Beschreibung
MOTOR_X_RIGHT_REQ	Stellt einen Request zum Steuern der Motorrichtung (rechts): 0: nicht nach rechts fahren 1: nach rechts fahren
MOTOR_X_SLOW_REQ	Stellt einen Request zum Steuern der langsamen Geschwindigkeit: 0: langsam aus 1: langsam an
MOTOR_X_STOP_REQ	Stellt einen Request zum Anhalten des Motors: 0: anhalten aufheben 1: anhalten

Watchdog

Der Watchdog läuft auf FBM1 und FBM2 und sendet all seine Events an den eigenen EventManager.

Name	Beschreibung
WD_CONN_ESTABLISHED	Nach einer Verbindungsaufforderung (z. B. nach dem Programmstart) hat die Partneranlage innerhalb der maximal erlaubten Zeit geantwortet
WD_CONN_LOST	Die Verbindung zur Partneranlage wurde unterbrochen
WD_CONN_REESTABLISHED	Nach einem vorigen ConnectionLost Event wurde die Verbindung wiederhergestellt
WD_M_HEARTBEAT	Heartbeat Message von FBM1 an FBM2
WD_S_HEARTBEAT	Heartbeat Message von FBM2 an FBM1

3.3.4 Sequenz Diagramm (Events)

Die folgenden Sequenzdiagramme zeigen den Signalaustausch zwischen Komponenten.

E-STOP Seq-Diagramm

Folgendes Diagramm zeigt den Signalaustausch zwischen Komponenten beim Betätigen des Estop an FBM1

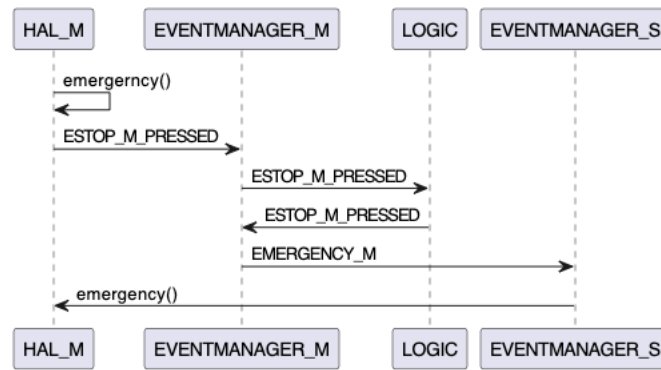


Figure 5 Seq_diagram_E-stop_FBM1

Folgendes Diagramm zeigt den Signalaustausch zwischen Komponenten beim Betätigen des Estop an FBM2

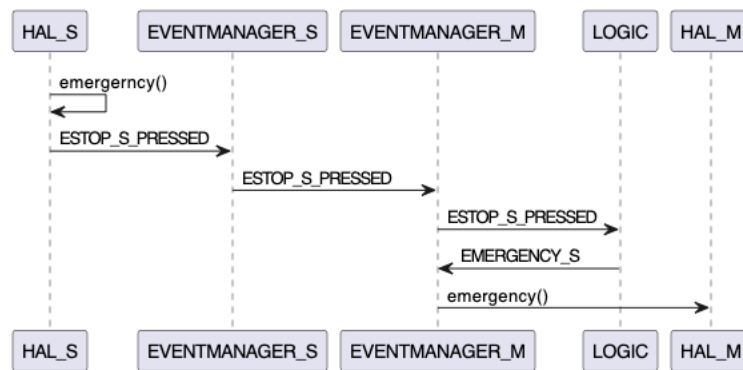


Figure 6 Seq_diagram_E-stop_FBM2

Der Signalaustausch zwischen Komponenten findet immer über den Eventmanager statt, wird aber in folgenden Diagrammen nicht gezeigt, um die Diagramme übersichtlicher zu machen.

Sortierung WS-F in FBM1

Folgendes Diagramm zeigt den Signalaustausch zwischen Komponenten bei Sortierung eines WS-F an FBM1

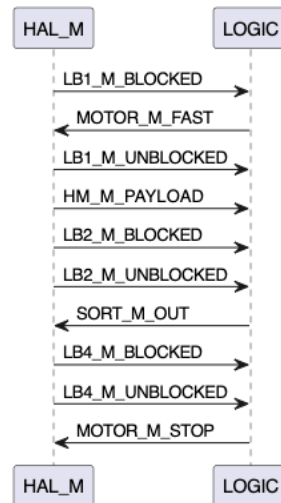


Figure 7 Seq_diagram_WS-F_FBM1

Sortierung WS-F RUTSCHE 1 VOLL

Folgendes Diagramm zeigt den Signalaustausch zwischen Komponenten bei Sortierung eines WS-F an FBM2, wenn die Rutsche an FBM1 voll ist.

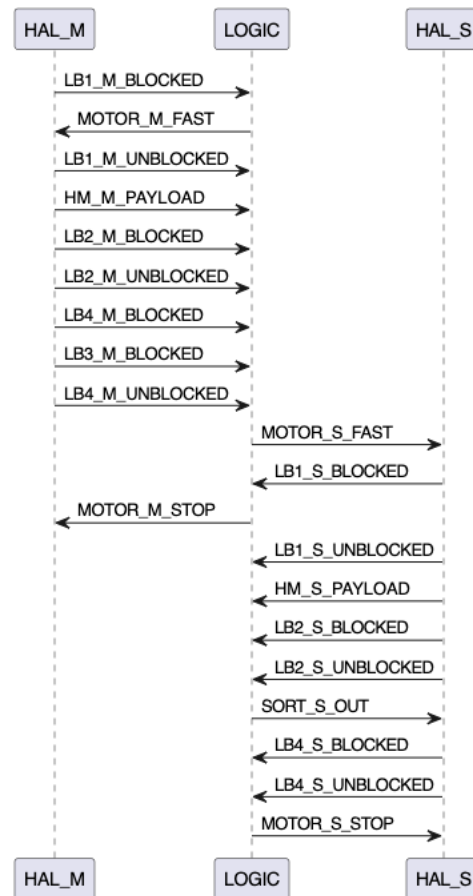


Figure 8 Seq_diagram_WS-F_FBM2

Watchdog Seq-Diagram

Folgendes Diagramm zeigt den Signalaustausch zwischen Watchdogs, um die Verbindungen zwischen beiden FBMs zu überwachen.

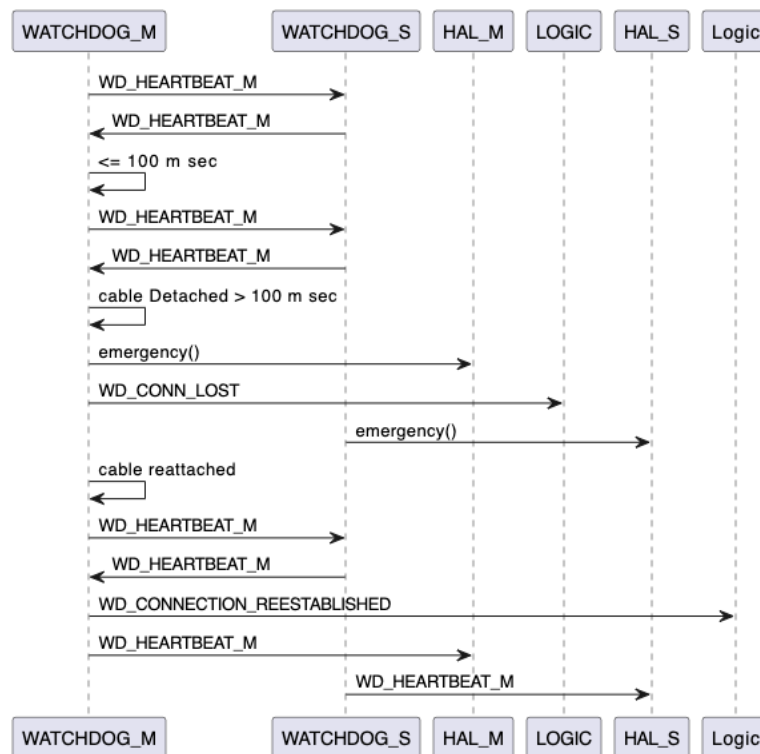


Figure 9 Seq_diagram_WATCHDOG

3.3.5 Warnungen und Fehler

Kategorie	Beschreibung	Anzeige
Warnung	Rutsche an FBM X voll	Gelbe Ampel blinkt am jeweiligen FBM
Warnung	Rutsche an FBM voll. Die Aussortierung erfolgt an anderem FBM.	Gelbe Ampel blinkt am FBM mit voller Rutsche. Zusätzliche Anzeige, dass Aussortierung an anderem FBM erfolgt: Leuchten der LED Q1 (für Master) bzw. Q2 (für Slave)
Fehler	Die Rutschen an beiden FBMs sind voll und ein Werkstück muss aussortiert werden.	Rote Ampel blinkt 1Hz an beiden FBM
Fehler	Netzwerk zwischen FBM1 und FBM2 unterbrochen.	Rote Ampel blinkt 1Hz an beiden FBM

4 Software-Design

4.1 Software Architektur

Master and Slave

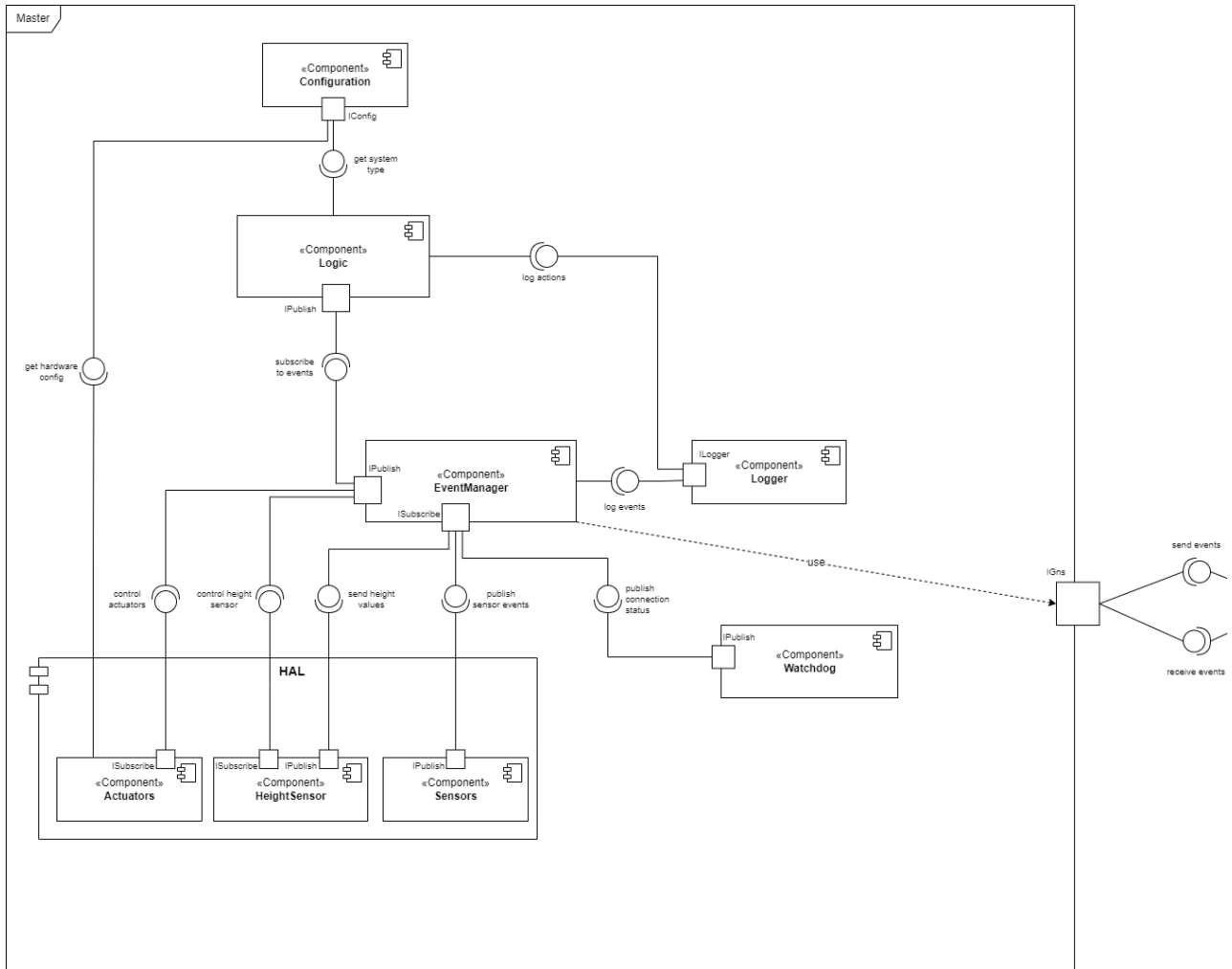


Figure 10 Component_diagram_master

4.1.1 Beschreibung der Komponenten

Der **EventManager** (EVM) ist die zentrale Schnittstelle, die alle im System auftretenden Events empfängt und an interessierte Komponenten weiterleitet. Dem EventManager angeschlossene Komponenten können sich für bestimmte Events anmelden ("subscribe"). Wenn diese auftreten, benachrichtigt der EventManager die Komponente darüber über den Aufruf einer Funktion ("publish").

Nachfolgend ist die Funktionsweise der einzelnen Komponenten kurz beschrieben:

Configuration

Beinhaltet die Konfiguration des Systems, auf dem die aktuelle Software läuft:

- Master oder Slave
- Auswerfer/Weiche
- WS-Reihenfolge (wird bei Programmstart eingelesen)
- Kalibrierwerte (wird in einer Datei gespeichert nach der Kalibrierung im Service-Mode)

EventManager

Empfängt Events von anderen Komponenten (Publishers) und leitet sie an Interessenten (Subscribers) weiter (an andere Anlage per GNS).

Watchdog

Überwacht Verfügbarkeit der Partneranlage, meldet Fehler und wenn ein voriger Verbindungsabbruch wieder behoben ist.

Logic

Beinhaltet die FSM's (Finite State Machines) zur Steuerung der Sortieranlage sowie Informationen über die auf der Anlage befindlichen Werkstücke. Dies wird über eine Klasse "WorkpieceManager" realisiert.

HAL

- Sensorik meldet Sensor-Events (z. B. LB unterbrochen/wieder frei, WS in Höhenmessung, Metall detektiert) an den EventManager
- Sensorik beinhaltet auch den Höhenmesser (ADC), der in einem eigenen Thread läuft und auf Anfrage eine kontinuierliche Höhenmessung startet bzw. stoppt. Die Messergebnisse werden nach der Messung als Event an den EVM geschickt (Messwerte in mm)
- Aktorik lässt sich ansteuern per Methodenaufruf (explizit oder HAL-Events beim EVM)

Logger

- Loggt Infos, Warnungen, Fehler und Debug-Meldungen auf die Konsole
- Alle wichtigen Events werden auf Master geloggt, was zum Debuggen auch ausreichend ist
- Ist auf alle benötigten Events subscribed, um diese zu loggen
- Logic schick bei Statewechsel changeEvents, welche ebenfalls geloggt werden

4.2 Software Struktur

4.2.1 Logic mit WorkpieceManager

Die Logic besteht aus mehreren FSM's, von denen nachfolgend die MainFSM dargestellt ist. Diese empfängt Events, wenn Taster gedrückt wurden und steuert den aktuellen Betriebsmodus.

Zusätzlich werden auch Events der Sensoren verarbeitet, die Informationen der auf dem Band befindlichen Werkstücke aktualisiert und ggf. Aktionen ausgelöst (z. B. Schließen der Weiche, um Werkstück auszusortieren). Dafür wird die Klasse WorkpieceManager benutzt, der alle Werkstücke, deren aktuelle Position und Daten in einer internen Liste verwaltet.

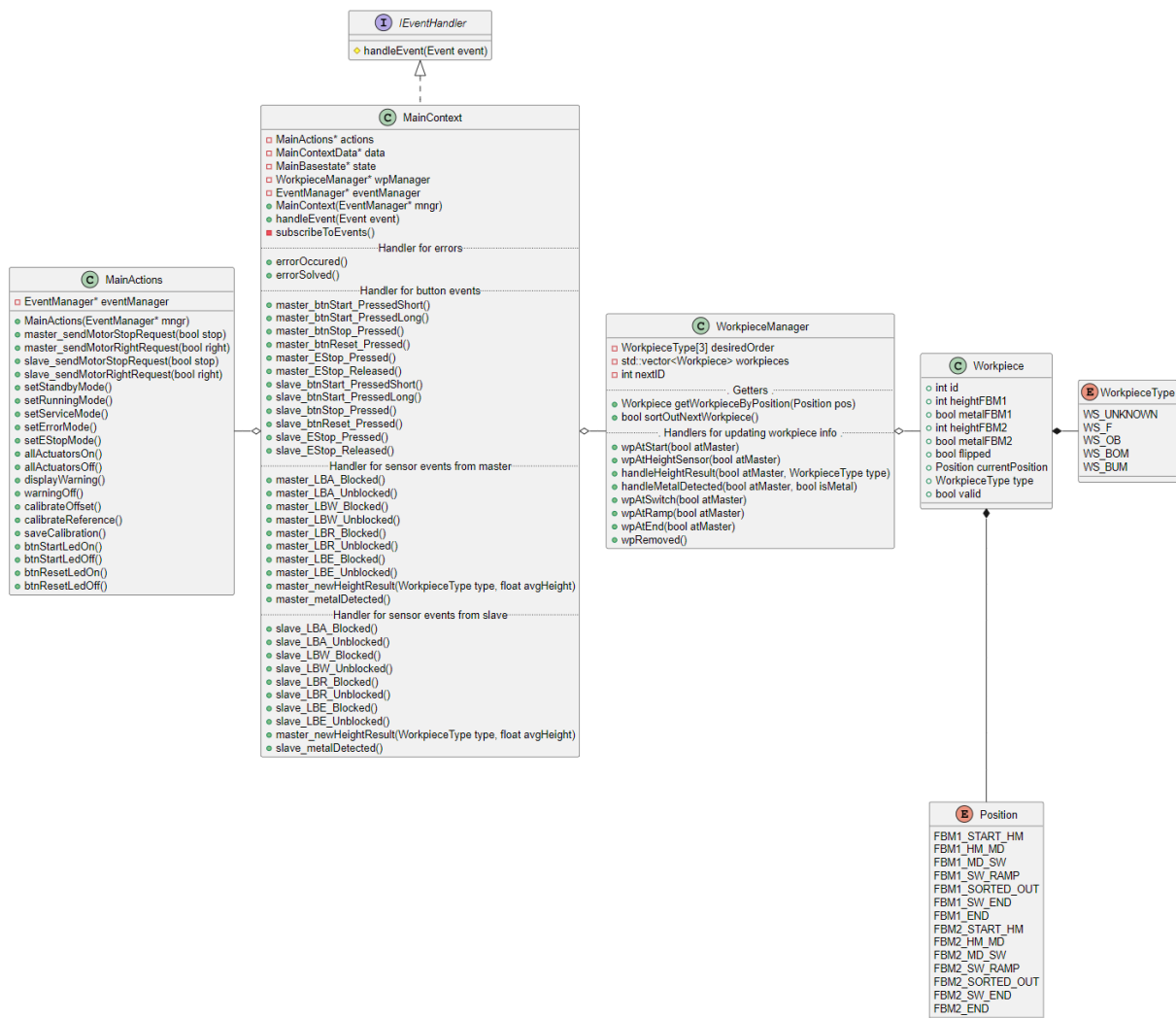


Figure 12 MainFSM mit WorkpieceManager

4.2.2 HAL, EventManager, Watchdog und Logger

Nachfolgend ist ein Klassendiagramm der HAL, des EventManagers und Loggers aufgeführt. Der EventManager agiert auch mit anderen Komponenten, allerdings eignet sich die HAL sehr gut, um das Konzept zum Behandeln von Events anhand des Reactor Patterns darzustellen.

Es existiert eine abstrakte Klasse ("Interface") *IEventHandler*, die lediglich über eine Methode *notify* verfügt. Jede Klasse, die dieses Interface nun implementiert, kann sich beim EventManager über die Methode *register* anmelden. Beim Auftreten dieses Events ruft der EventManager die Callback-Methode *notify* auf. Das weitere Verarbeiten des Events geschieht nun in der Klasse des spezifischen EventHandlers (Beispiel: HAL Aktorik).

Auf dem Klassendiagramm ist auch der Logger aufgeführt, welcher auch über ein Interface implementiert wird. Dies lässt Raum dafür, den *ConsoleLogger*, der Meldungen auf die Konsole loggt, einfach auszutauschen, z. B. über einen Logger, der die Log-Meldungen per MQTT sendet oder in eine Datei schreibt.

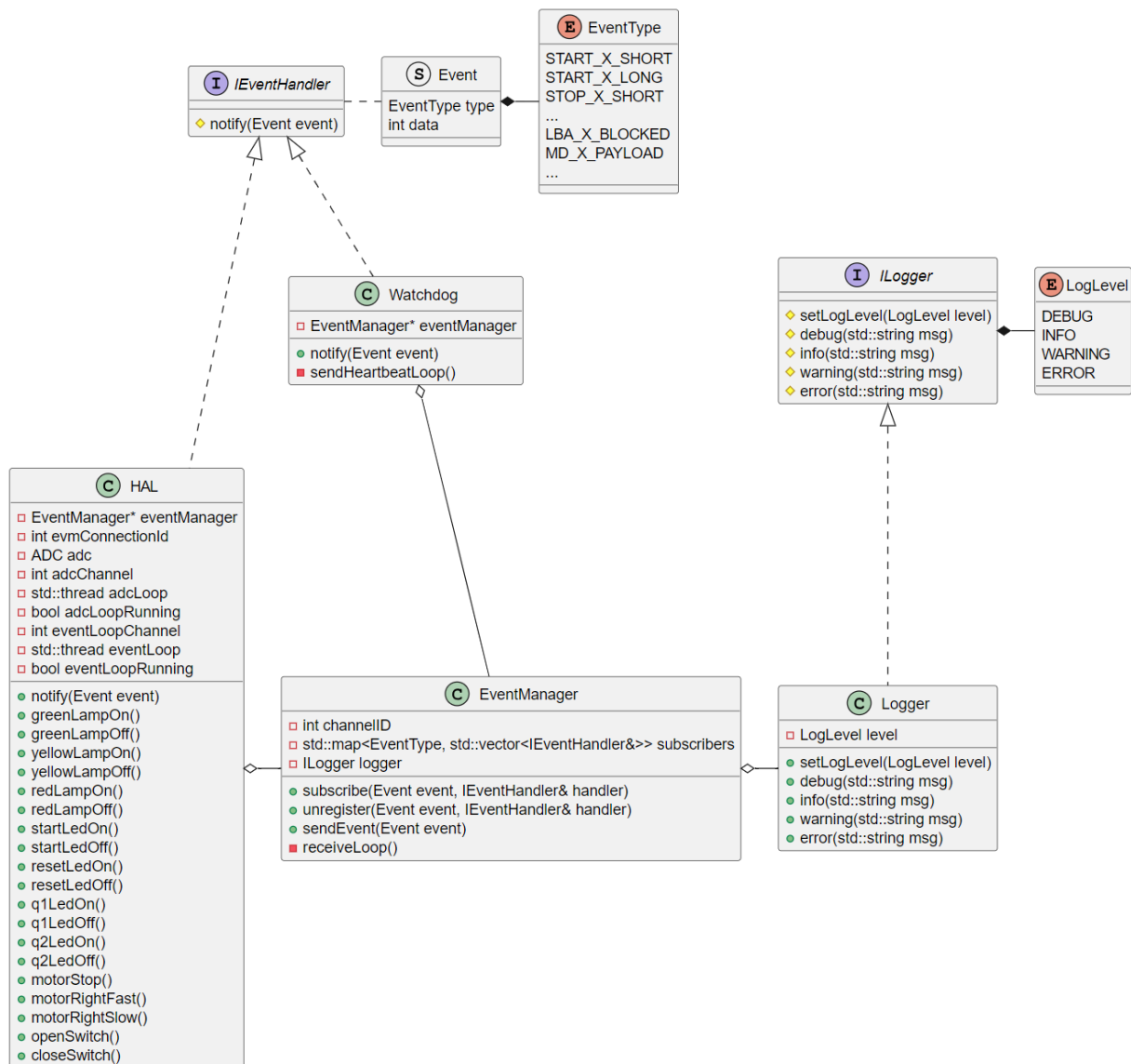


Figure 13 HAL, EventManager, Watchdog und Logger

4.2.3 Configuration

Die Configuration Komponente ist als Singleton implementiert, es existiert damit nur ein Objekt davon zur Laufzeit. Zum Programmstart wird die aktuelle Config aus der Datei eingelesen und intern gespeichert. Komponenten, die die aktuelle Konfiguration benötigen können diese über eine Referenz auf das Objekt einlesen.

Die Konfiguration enthält folgende Informationen:

- isMaster – true: Configuration läuft auf dem Master-System, false: Slave
- pusher – true: Zum Aussortieren wird der Auswerfer verwendet, false: Weiche
- configuredWorkpieceOrder: Die gewünschte Reihenfolge der Sortierung
- calOffset: Das kalibrierte Offset (Laufband-Höhe)
- adcIncPerMillimeter: Die kalibrierte Höhe eines flachen Werkstücks (wird zur zukünftigen Berechnung benötigt)

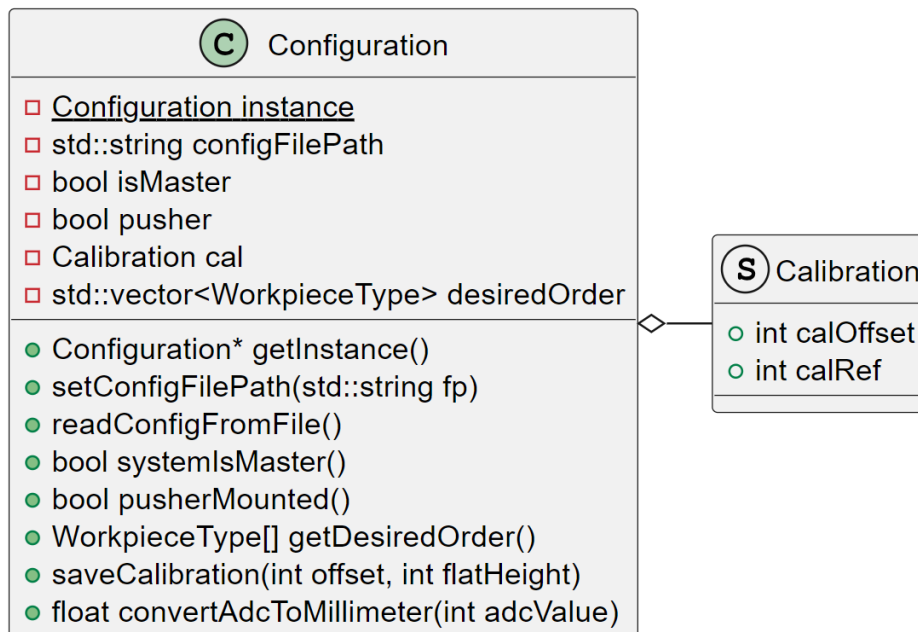


Figure 14 Configuration

4.3 Verhaltensmodellierung

Nachfolgend sind die FSM's aufgeführt, die alle auf dem Master-System (FBM1) parallel laufen:

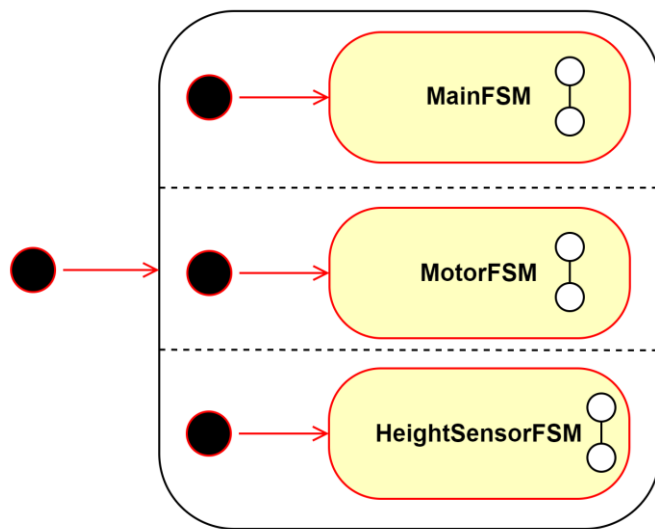


Figure 15 All FSMs

4.3.1 MainFSM

Zur Steuerung des aktuellen Betriebsmodus verwenden wir eine MainFSM. Diese reagiert im Wesentlichen auf die Tastendrücke (Start, Stop, E-Stop) und wechselt die Betriebsmodi entsprechend.

Im Betriebszustand (Running) wird auch auf Events von den Sensoren reagiert. Diese müssen so verarbeitet werden, dass die Informationen zu den auf den Bändern befindlichen Werkstücken aktualisiert werden. Dafür verwenden wir einen internen "WorkpieceManager", der die aktuellen Informationen und Positionen der Werkstücke kennt.

- **Standby:** Ruhezustand – alles aus
- **Running:** Betriebszustand - Grüne Lampe leuchtet, bereit zum Einlegen neuer Werkstücke. E
- **ServiceModeFSM:** Hier werden Selbsttests und Kalibrierungen durchgeführt. Da dies ein spezieller Modus ist, der nur von befähigten Personen durchgeführt werden kann, behandeln wir nicht das Drücken des E-Stopps
- **EStopFSM:** alles aus – die Sub-Statemachine ist dafür zuständig, dass dieser Modus wieder verlassen werden kann, wenn die Anlage wieder sicher ist
- **ErrorFSM:** Fehlerzustand – die Sub-Statemachine ist dafür zuständig, dass der Fehler gemäß Anforderung quittiert und behoben wird

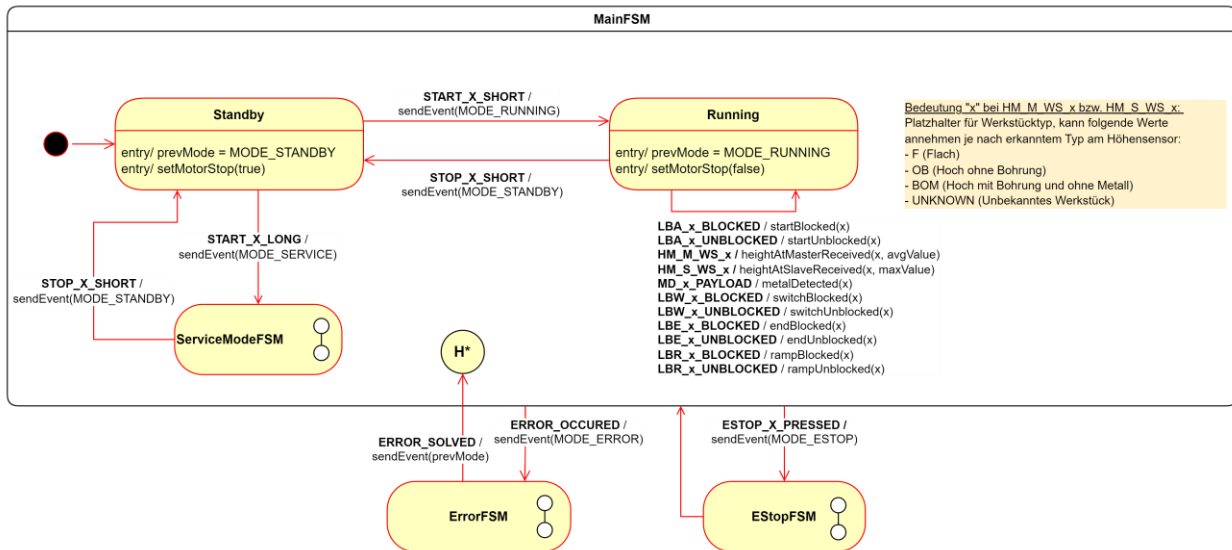


Figure 16 MainFSM

Bei jedem Zustands-Wechsel wird der neue Betriebsmodus per Event gemeldet (MODE_x). Die HAL ist auf diese Events registriert und schaltet die Aktoren je nach Zustand (Lampen und Motor).

Falls in den Fehlermodus (ErrorFSM) gewechselt wird und danach wieder in den letzten Zustand gewechselt wird, muss der letzte Modus gemerkt werden, um ein neues Event zu verschicken, das dann nochmals den Wechsel des Betriebsmodus meldet. Dafür benutzen wir die Variable **prevMode**.

Modus Running:

Im Modus Running (Betriebsmodus) werden alle empfangenen SensorEvents mithilfe des WorkpieceManager verarbeitet.

Hinweis: Der Platzhalter "x" steht für den Systemtyp (M für Master, S für Slave).

- startBlocked(MASTER):
 - Anlegen eines neuen Werkstücks
 - Wenn vorher noch kein WS auf FBM1 war: Versenden des Events FBM1_OCCUPIED
 - setMotorFast(true): Setzt das Flag „Motor fast“ und lässt den Motor anlaufen
- heightAtMasterReceived(x, avgValue) / heightAtSlaveReceived(x, maxValue):
 - Beim Werkstück, das auf die Höhenmessung wartet, wird der erkannte Typ und der durchschnittliche (FBM1) bzw. maximale (FBM2) Höhenmesswert gesetzt
- metalDetected(x):
 - Beim Werkstück das am Metallsensor angekommen ist, wird der erkannte Typ geändert
- switchBlocked(x):
 - Bei dem Werkstück, das sich an der Weiche befindet, wird geprüft, ob es aussortiert werden soll.
 - Wenn ja und Rampe frei: Schließen der Weiche. Ansonsten: Öffnen der Weiche
- rampBlocked(x):
 - Das WS, das auf sein Aussortieren wartet, wird aus dem WorkpieceManager entfernt
 - Der Status der jeweiligen Rutsche wird auf "belegt" gesetzt und es erscheint eine Warnung
 - Wenn sich auf FBM1 bzw. FBM2 kein WS mehr befindet, wird durch Aufruf von setMotorFast(false) der Motor angehalten

- FBM2 verschickt das Event FBM2_FREE, um im Falle einer anstehenden Übergabe von FBM1 den Transfer einzuleiten
- rampUnblocked(x):
 - Der Status der jeweiligen Rutsche wird auf "frei" gesetzt und die Warnung „Rutsche voll“ verschwindet
- endBlocked(MASTER):
 - Wenn FBM2 frei: Versenden des Events TRANSFER_START (Motor FBM1 und FBM2 nach rechts) und setMotorFast(true)
 - Wenn FBM2 belegt: Versenden des Events WAIT_FOR_TRANSFER und Stopp des Motors durch Aufruf von setMotorStop(true)
- endBlocked(SLAVE):
 - Versenden des Events WAIT_FOR_REMOVAL (Motor FBM2 stoppen und warten, bis WS entnommen wurde)
 - Stoppen des Motors durch Aufruf von setMotorStop(true)
 - Ausgabe der Werkstück-Daten
- endUnblocked(SLAVE):
 - WS, das auf die Entnahme wartet aus dem WorkpieceManager entfernen
 - Versenden des Events FBM2_FREE, um im Falle einer anstehenden Übergabe von FBM1 den Transfer einzuleiten
- startBlocked(SLAVE):
 - Wenn ein aktueller Transfer stattfindet (ein WS wartet auf das Ankommen an FBM2), wird der Zähler für Anzahl der Werkstücke auf FBM1 dekrementiert. Wenn dieser 0 ist, wird durch Aufruf von setMotorFast(false) der Motor an FBM1 gestoppt.
 - Der Status von FBM2 wird aktualisiert auf "belegt" und das Event FBM2_OCCUPIED wird versendet
 - Starten des Motors an FBM2 durch Aufruf von setMotorFast(true)

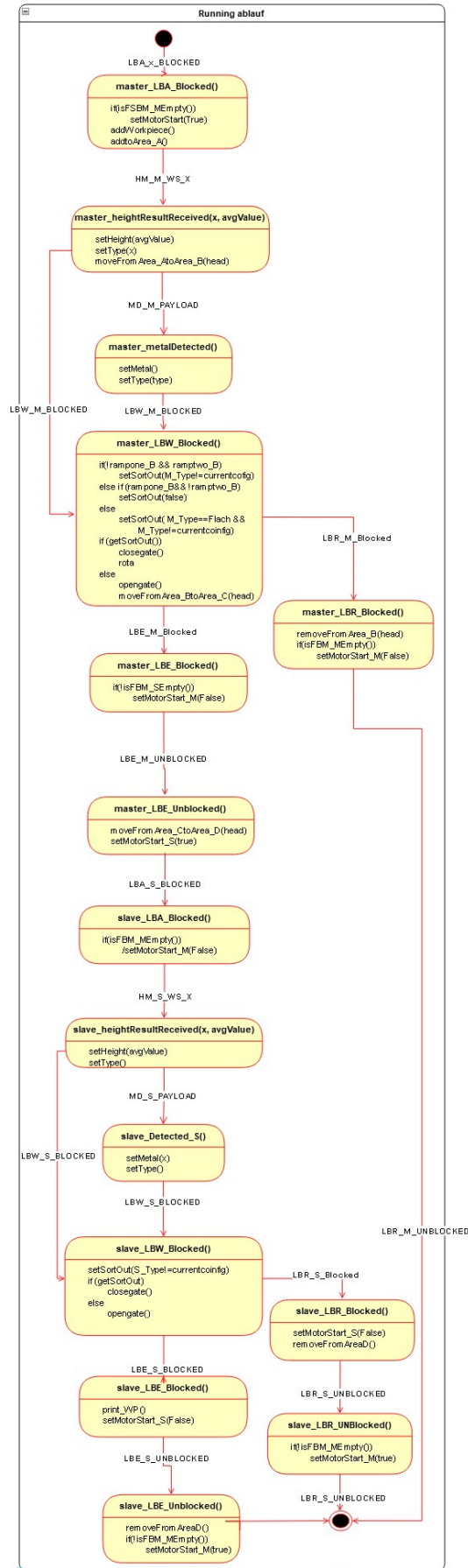


Figure 17 Running Ablauf

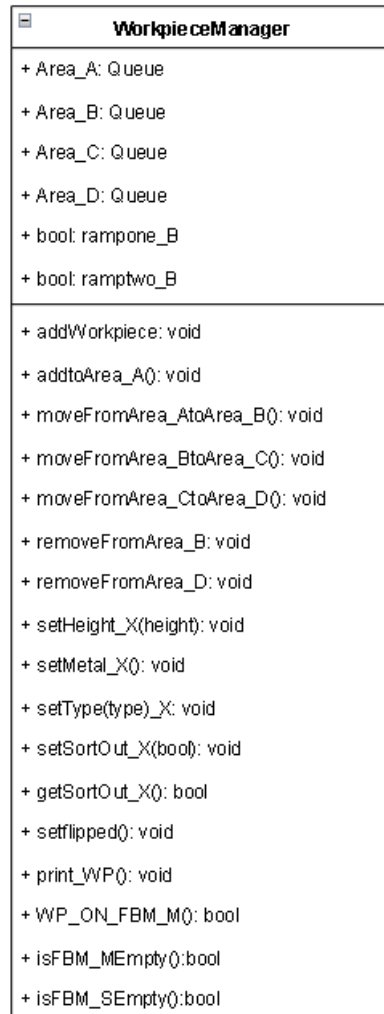


Figure 18 Class Workpiecemanager



Figure 19 Class Workpiece

4.3.2 MotorFSM

Da es möglich ist, dass mehrere Komponenten zur gleichen Zeit Anfragen zur Änderung der Geschwindigkeit stellen, ist es notwendig diese zu priorisieren. Dafür verwenden wir eine MotorFSM, von der jeweils für FBM1 und FBM2 eine Instanz auf dem Master erstellt wird.

Es bestehen drei boolsche Variablen („Flags“), die bei Empfang der Events MOTOR_X_[STOP|FAST|SLOW] auf true bzw. false gesetzt werden. Nach jedem empfangenen Event wird der Status aller Variablen geprüft und je nach Kombination der Zustand gewechselt.

Beim Wechsel in einen neuen State verschickt die MotorFSM dann Events zur Steuerung des aktuellen Modus über den EventManager an die Aktorik.

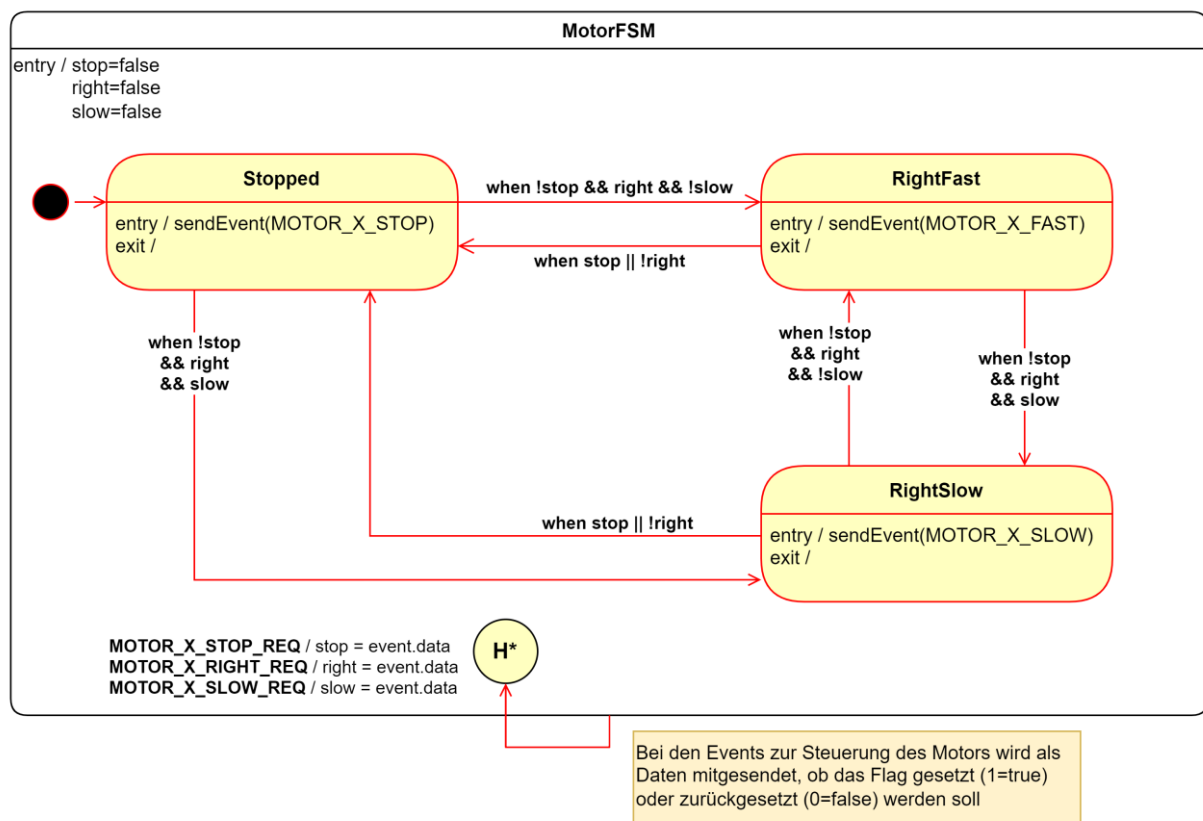


Figure 20 MotorFSM

4.3.3 HeightSensorFSM

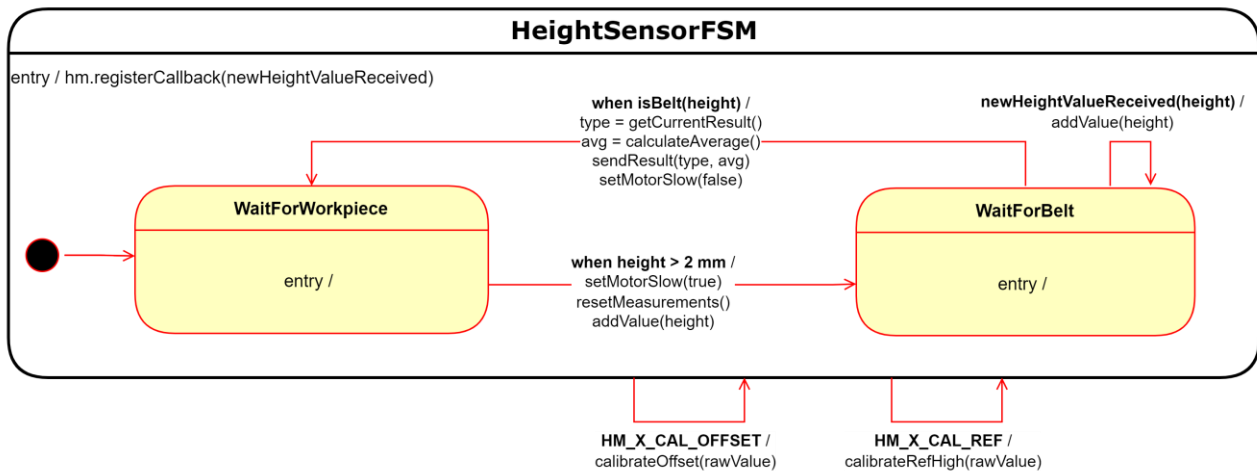


Figure 21 HeightSensorFSM

Für den Höhsensor existiert eine eigene FSM, die dafür zuständig ist, den Ablauf während der Messung von Werkstücken zu steuern. Anfangs ist die FSM im Zustand WaitForWorkpiece und empfängt kontinuierlich neue Messwerte vom Höhsensor. Bei jedem neuen Wert des ADC wird die Funktion newHeightValueReceived(int val) aufgerufen. Der Rohwert des ADC wird in mm umgewandelt und die FSM prüft, in welchem Bereich der Messwert liegt.

Wenn eine Höhe größer als die Laufbandhöhe (+ 2mm Toleranz) erkannt wird, wird in den Zustand WaitForBelt gewechselt, in dem alle Messwerte zwischengespeichert werden. Wenn wieder Laufbandhöhe erkannt wird, wird zurück in den Zustand WaitForWorkpiece gewechselt und anhand der aufgenommenen Messwerte der erkannte Werkstücktyp sowie der durchschnittliche Höhenmesswert bestimmt. Das Ergebnis der gesamten Messung wird über den EventManager an die MainFSM versendet, die daraufhin die Werkstückdaten aktualisiert.

Kalibrierung des Höhsensors:

Durch das Senden der Events HM_X_CAL_OFFSET bzw. HM_X_CAL_REF kann der Offset bzw. Referenzwert (25,0mm) einkalibriert werden.

Danach wird der Multiplikator-Wert "ADC-Inkrement pro mm" wie folgt berechnet:

$$\text{Step}[1/\text{mm}] = (\text{Offset} - \text{Referenz}) / 25,0 \text{ mm}$$

Während der Messungen wird die aktuell gemessene Höhe dann wie folgt berechnet:

$$\text{Höhe}[\text{mm}] = (\text{Offset} - \text{Rohwert}) / \text{Step}$$

4.3.4 EStopFSM

Die Sub-Statemachine "EStopFSM" wird betreten, wenn einer der beiden E-Stopp Schalter gedrückt wurde. Um aus diesem Modus wieder raus zu kommen, müssen die E-Stopp beider Förderbandmodule wieder herausgezogen sein und an beiden Bändern der Reset-Taster gedrückt werden. Danach wird wieder in den Ruhezustand (Standby) gewechselt.

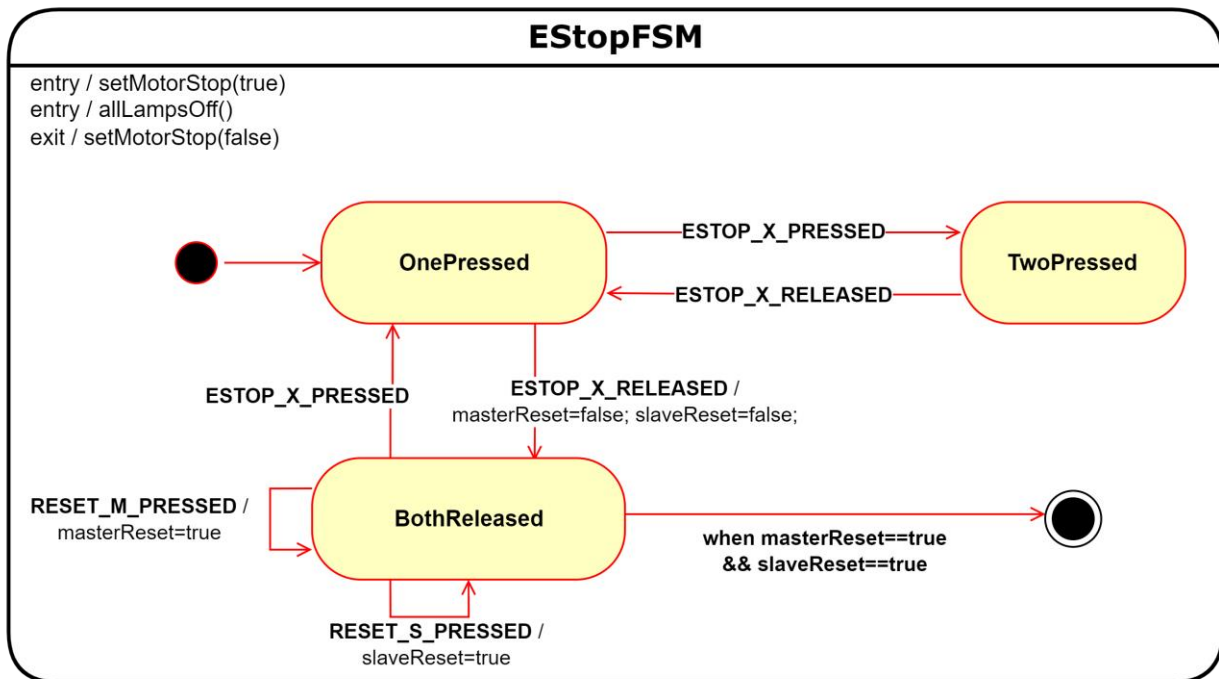


Figure 22 EStopFSM

4.3.5 ErrorFSM

Die Sub-Statemachine "ErrorFSM" wird betreten, wenn ein Fehler auftritt. Der Fehlermodus kann erst verlassen werden, wenn der Fehler behoben und quittiert wurde.

Dabei wird berücksichtigt, dass sich ein Fehler von selbst beheben kann (State "Solved Unresigned") oder manuell behoben werden muss (State "Pending Resigned").

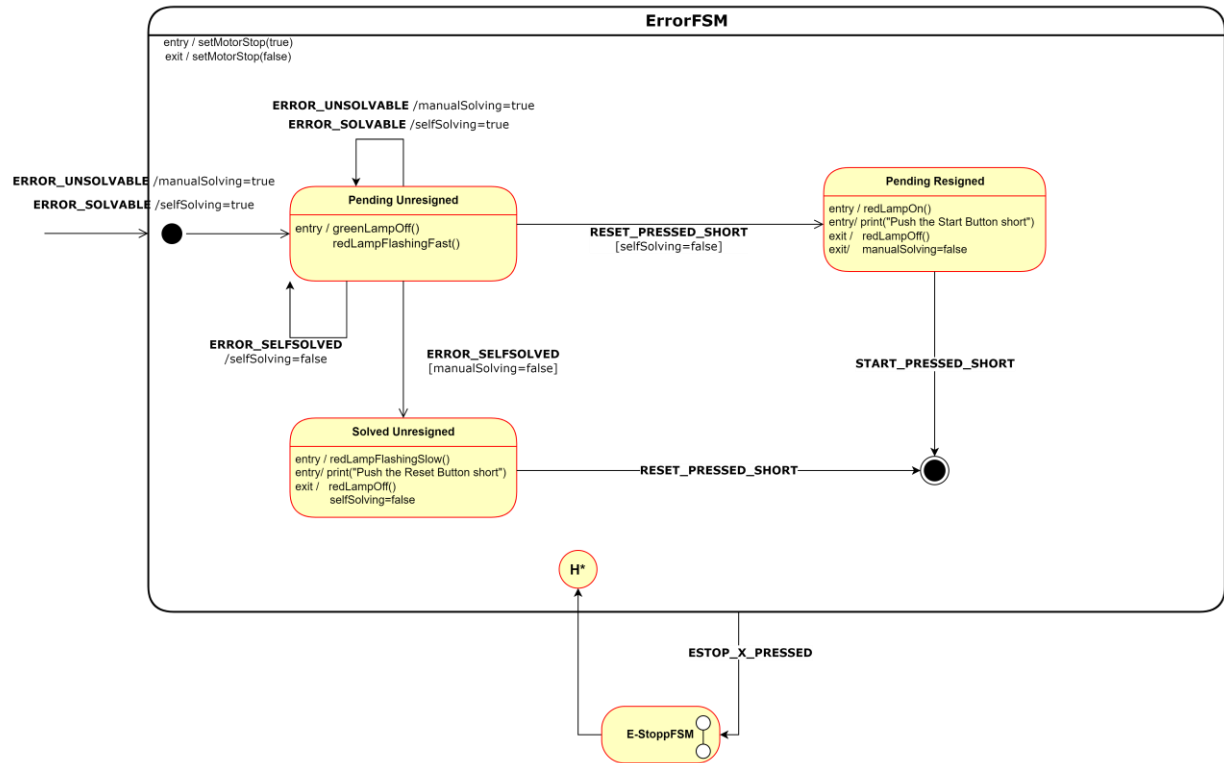


Figure 23 ErrorFSM

4.3.6 ServiceModeFSM

Die ServiceModeFSM wird betreten, wenn man länger als 2 Sekunden den Start-Taster drückt und kann wieder verlassen werden, wenn man den Stop-Taster drückt.

- **SelftestSensors:** Testet die Funktion der Sensoren durch Unterbrechen durch den Benutzer und Ausgabe auf der Konsole
- **SelftestActuators:** Schaltet alle Lampen und die Weiche an und fragt den Benutzer, ob die Funktion gegeben ist
- **CalHeightFSM:** Kalibrierung des Höhensensors
- **CalMotorFSM:** Kalibrierung der Motorgeschwindigkeit

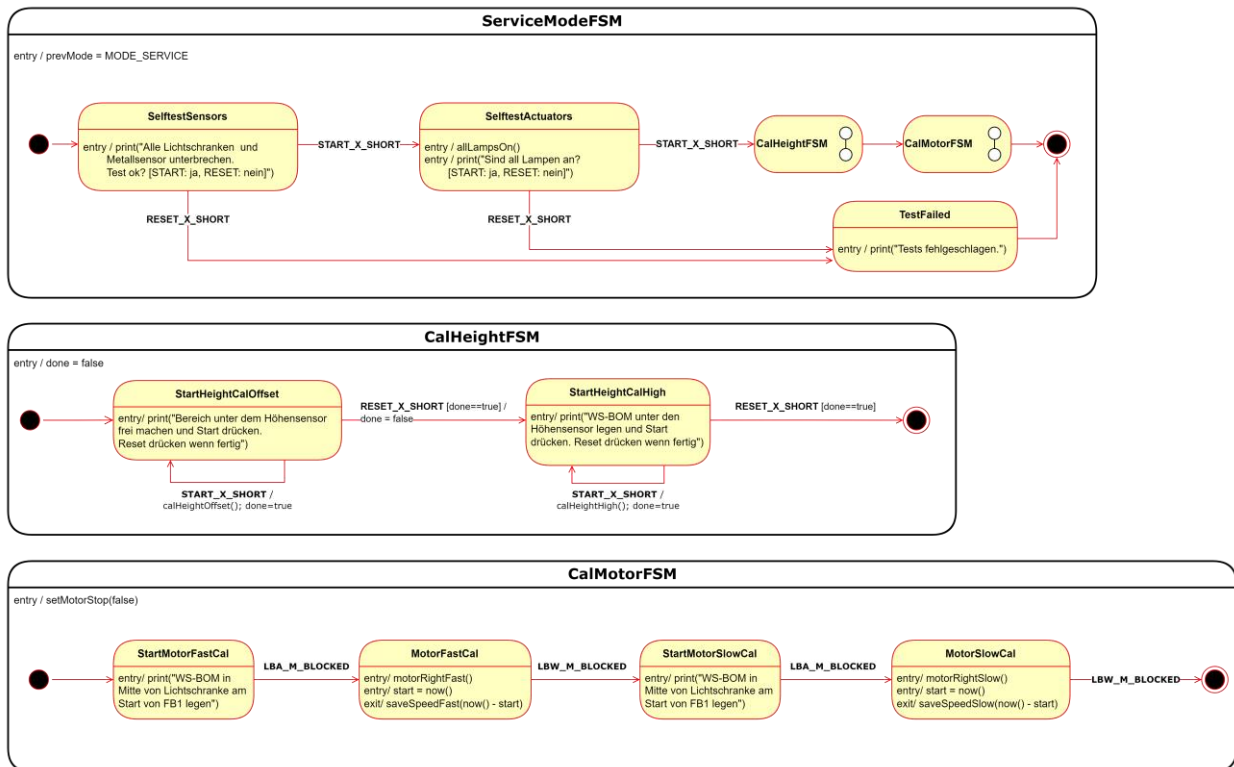


Figure 24 ServiceModeFSM

5 Implementierung

Im Kapitel [Beschreibung der Komponenten](#) ist bereits einiges über die Implementierung beschrieben.

Nachfolgend wird noch mehr ins Detail eingegangen.

5.1 Höhenmessung

Zu Beginn hatten wir eine FSM modelliert, die ihren Zustand wechselt, wenn ein anderes Höhenlevel erkannt wird. Dies hat sich während Tests als sehr fehleranfällig erwiesen. Deshalb wurde die FSM geändert und diese hat nur noch zwei Zustände. Wenn Laufbandhöhe gemessen wird, ist die Höhenmessung „inaktiv“. Wenn ein Werkstück unter dem Sensor erkannt wird, wird begonnen, alle Messwerte zwischenzuspeichern, bis wieder Laufbandhöhe erkannt wird. Durch Aufruf der Funktion `getAverageHeight()` wird der durchschnittliche Wert der letzten 100 ADC-Samples aufgerufen. Danach wird die Höhengsensor FSM über den neuen Wert über eine Callback-Funktion benachrichtigt.

```
float HeightSensor::adcValueToMillimeter(int adcValue) {  
    int abs = adcOffset - adcValue;  
    float mm = ((float) abs / adcIncPerMillimeter);  
    if (mm < 0)  
        return 0.0;  
    return mm;  
}  
  
float HeightSensor::getAverageHeight() {  
    if (window.empty())  
        return 0.0;  
    long sum = 0;  
    for (int val : window) {  
        sum += val;  
    }  
    int avg = sum / window.size();  
    return adcValueToMillimeter(avg);  
}
```

Figure 25 Auszug: Bestimmung der Höhenmesswerte in mm anhand der ADC-Rohwerte

Wenn die FSM Laufbandhöhe erkennt, wird der aktuelle Typ anhand der empfangenen Messwerte bestimmt. Da es im Randbereich meist Fehlmessungen gibt, werden die oberen und unteren 5% der Messungen verworfen. Dann wird anhand der Messwerte am Anfang, Mitte und Ende des Werkstücks der Typ bestimmt.

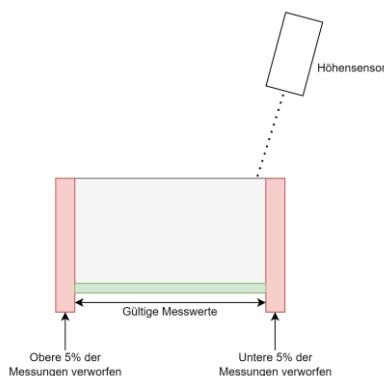


Figure 26 Skizze Höhenmessung

5.2 EventManager

Die Komponenten kommunizieren miteinander mit dem Publish/Subscribe Pattern. Der EventManager empfängt alle internen und externen Events über QNX PulseMessages. Daraufhin ruft er die Callback-Funktionen an allen Subscribern auf.

Auf Events „subscribed“ werden kann wie folgt am EventManager (Beispiel):

```
eventManager->subscribe(EventType::START_M_SHORT,  
    std::bind(&MainContext::handleEvent, this, std::placeholders::_1)  
);
```

Figure 27 Subscribe auf Events mit Callback-Funktion

Die Callbackfunktionen der Subscriber zu einem Event werden vom EventManager wie folgt aufgerufen:

```
if (subscribers.find(event.type) != subscribers.end()) {  
    for (const auto &callback : subscribers[event.type]) {  
        callback(event);  
    }  
}
```

Figure 28 Aufruf der Callback-Funktionen bei den Subscribern

6 Qualitätssicherung

6.1 Teststrategie

Phase 1: Modultest

Während der ersten Entwicklungsphase werden die Teilkomponenten (Module) des Systems unabhängig voneinander entwickelt. Deren Funktionalität wird mithilfe automatisierter Unit-Tests überprüft. Bevor ein Modul fertig gestellt werden kann, muss es als „Black Box“ getestet werden, d. h. die Schnittstellen, die ein Modul für andere Komponenten zur Verfügung stellt, werden getestet anhand der Design-Spezifikation. Dabei ist es ggf. erforderlich, dass das Verhalten anderer Komponenten simuliert wird (bspw. das Versenden von Events und Vergleich der Reaktion des Moduls darauf).

Phase 2: Integrationstest

Sobald alle Einzelmodule entwickelt sind, wird deren Zusammenwirken miteinander getestet. Dabei muss das Verhalten des Gesamt-Systems simuliert werden. Unterstützung dabei kann die zur Verfügung gestellte Simulationssoftware bieten, mit der Abläufe an der realen Anlage simuliert und reproduzierbar wiederholt werden können.

Phase 3: Systemtest

Mit dem Systemtest wird die fertige Software auf das Zielsystem aufgespielt und die Funktionalität des Systems als Ganzes getestet. Die Einhaltung der Kundenanforderungen, Sicherheitsfunktionen sowie Qualitätsanforderungen werden getestet. Abweichungen sind festzuhalten und zu beseitigen, bevor die Abnahme durch den Kunden erfolgen kann.

Phase 4: Abnahmetests

Nach erfolgreichem Bestehen des Systemtests und Beseitigen letzter Bugs, erfolgt die Abnahme durch den Kunden durch Abnahmetests, bei denen die Einhaltung der Kundenanforderungen überprüft werden. Die Testfälle sind unter **6.2 Testscenarien/Abnahmetest** definiert. Während der Abnahme werden die Testergebnisse protokolliert

6.2 Testszenarien/Abnahmetest

Test ID	T1	Getestete Anforderungen	ANF25, ANF28, ANF29	Priorität	normal
Beschreibung	ServiceMode: Selbsttests und Kalibrierung des Systems				
Vorbedingung	<ul style="list-style-type: none"> Die Anlage ist aus Auf beiden FBM befindet sich keine Konfigurationsdatei (unter „tmp/ese_2.1/conf.txt“) Die Funktion aller Sensoren und Aktoren der beiden Anlagen ist gegeben 				
Input Daten	Höhensensor, Lichtschranken				
Ablaufbeschreibung	Aktion		Erwartung	Erfüllt	
	Programm an Master und Slave starten		Alle Lampen sind aus. Die LED am Start-Taster leuchtet. In der Konsole erscheint der Hinweis, dass eine neue Konfigurationsdatei erstellt wurde und eine Kalibrierung erforderlich ist.		
	Start-Taster lange drücken (≥ 2 Sekunden)		Die grüne Lampe blinkt. In der Konsole erscheint der Hinweis, dass sich kein Werkstück unter dem Höhengsensor an beiden Anlagen befinden soll.		
	Start-Taster drücken		Die LED am Reset-Taster leuchtet		
	Reset-Taster drücken		Die LED am Reset-Taster ist aus. In der Konsole erscheint der Hinweis, dass sich ein hohes Werkstück (25 mm) unter dem Höhengsensor an beiden Anlagen befinden soll.		
	Hohes Werkstück (25 mm ± 1mm) unter den Höhengsensor an beiden Anlagen legen und Start-Taster drücken		Die LED am Reset-Taster leuchtet		
	Reset-Taster drücken		Die LED am Reset-Taster ist aus. In der Konsole erscheint der Hinweis, dass ein beliebiges Werkstück eingelegt werden soll		
	Beliebiges Werkstück am Anfang von FBM1 einlegen		Das Werkstück fährt bis an das Ende von FBM2 und hält dort an.		

		Die Lichtschranken an der Rampe von FBM1 und FBM2 manuell unterbrechen. Danach Start-Taster drücken	In der Konsole erscheint die Meldung, dass alle Sensoren OK sind. Es wird übergegangen zum „Aktorik Selbsttest“, alle Lampen leuchten und die Weiche öffnet sich		
		Start-Taster zwei Mal nacheinander drücken	Das System wechselt in den Ruhezustand und danach in den Betriebszustand		
		Flaches Werkstück auflegen	Das Werkstück fährt durch bis an das Ende von FBM2		

Test ID	T2	Getestete Anforderungen	ANF23, ANF27, ANF28	Priorität	<i>kritisch</i>
Beschreibung	Funktionsweise des E-Stopp				
Vorbedingung	Die Anlage ist aus. Der E-Stopp an FBM1 ist gedrückt.				
Input Daten	E-Stopp, Reset, Start-Taster				
Ablaufbeschreibung	Aktion	Erwartung		Erfüllt	
	Anlage einschalten und Start drücken	Die Anlage verbleibt im E-Stopp Modus			
	E-Stopp an FBM1 herausziehen	Die LED an beiden Reset-Tastern leuchtet			
	Reset an beiden FBM drücken	Die LED an beiden Start-Tastern leuchtet			
	Start-Taster an einer der FBM drücken	Die grüne Lampe an beiden FBM leuchtet			
	Zwei Werkstücke nacheinander einlegen und warten, bis beide Motoren laufen. Danach E-Stopp an FBM2 drücken	Die Förderbänder an beiden FBM stehen still. Alle Lampen und LED's sind aus			
	Reset an beiden FBM und Start-Taster drücken	Die Anlage verbleibt im aktuellen Zustand			
	E-Stopp an FBM2 herausziehen, Bänder freiräumen, Reset an beiden FBM und einen Start-Taster drücken	Die Anlage wechselt in den Betriebszustand. Die grüne Lampe an beiden FBM leuchtet			

Test ID	T3	Getestete Anforderungen	ANF01, ANF02	Priorität	<i>hoch</i>
Beschreibung	Sortierung der Werkstücke in vorgegebener Reihenfolge				
Vorbedingung	Konfigurierte Reihenfolge: <WS-F> → <WS-BuM> → <WS-OB> (In Konfigurationsdatei: <i>ORDER=F,BUM,OB</i>) Anlage befindet sich im Betriebsmodus.				
Input Daten	Werkstücke, Lichtschranken, Höhensensor				
Ablaufbeschreibung	Aktion	Erwartung	Erfüllt		
	Flaches Werkstück (WS-F) auflegen	Das Werkstück fährt durch bis an das Ende von FBM2. Die Werkstückdaten werden an FBM1 und FBM2 jeweils spätestens an der Weiche ausgegeben.			
	Werkstück am Ende entnehmen. Hohes Werkstück mit Bohrung und Metall (WS-BuM) auflegen	Das Werkstück fährt durch bis an das Ende von FBM2. Die Werkstückdaten werden an FBM1 und FBM2 jeweils spätestens an der Weiche ausgegeben.			
	Werkstück am Ende entnehmen. Hohes Werkstück ohne Bohrung (WS-OB) auflegen	Das Werkstück fährt durch bis an das Ende von FBM2. Die Werkstückdaten werden an FBM1 und FBM2 jeweils spätestens an der Weiche ausgegeben.			

Test ID	T4	Getestete Anforderungen	ANF01, ANF02, ANF03, ANF04, ANF18	Priorität	<i>hoch</i>
Beschreibung	Aussortierung von Werkstücken, die nicht der vorgegebenen Reihenfolge entsprechen				
Vorbedingung	<ul style="list-style-type: none"> Konfigurierte Reihenfolge: <WS-F> → <WS-BuM> → <WS-OB> (In Konfigurationsdatei: <i>ORDER=F,BUM,OB</i>) Anlage befindet sich im Betriebsmodus Die Rutschen an beiden FBM sind frei Als nächstes Werkstück wird WS-F erwartet (nach dem Einschalten) 				
Input Daten	Werkstücke, Lichtschranken, Höhengsensor				
Ablaufbeschreibung	Aktion	Erwartung		Erfüllt	
	Werkstück WS-OB (Hoch ohne Bohrung) auflegen	Werkstück wird an der Rutsche von FBM2 aussortiert			
	Werkstück WS-BUM (Hoch mit Bohrung und Metall) auflegen	Werkstück wird an der Rutsche von FBM2 aussortiert			
	Werkstück WS-BOM (Hoch mit Bohrung ohne Metall) auflegen	Werkstück wird an der Rutsche von FBM2 aussortiert			
	Binär-codiertes Werkstück auflegen	Werkstück wird an der Rutsche von FBM2 aussortiert			
	Werkstück WS-F (Flach) auflegen	Werkstück wird bis an das Ende von FBM2 befördert			
	Werkstück WS-F (Flach) auflegen	Werkstück wird an der Rutsche von FBM1 aussortiert			

Test ID	T5	Getestete Anforderungen	ANF05, ANF06	Priorität	<i>normal</i>
Beschreibung	Mehrere Werkstücke dürfen sich auf FBM1 befinden. Auf FBM2 darf sich maximal ein Werkstück befinden.				
Vorbedingung	<ul style="list-style-type: none"> Anlage befindet sich im Betriebsmodus Konfigurierte Reihenfolge: <WS-F> → <WS-BuM> → <WS-OB> (In Konfigurationsdatei: <i>ORDER=F,BUM,OB</i>) 				
Input Daten	Werkstücke, Lichtschranken, Höhengsensor				
Ablaufbeschreibung	Aktion	Erwartung		Erfüllt	
	Zwei Werkstücke (WS-F und WS-BUM) nacheinander auf FBM1 legen	Der Motor läuft nach Einlegen des ersten Werkstücks an			
	Warten, bis sich das erste Werkstück auf FBM2 befindet	Der Motor an FBM1 läuft weiter und hält an, wenn das zweite Werkstück am Ende von FBM1 angekommen ist			

	Warten, bis das erste Werkstück am Ende von FBM2 angekommen ist. Dann entnehmen	Der Motor an FBM1 läuft an und das Werkstück wird an das Ende von FBM2 befördert. Dann stehen beide Motoren still	
--	---	---	--

Test ID	T6	Getestete Anforderungen	ANF07	Priorität	<i>normal</i>
Beschreibung	Werkstücke sollen langsam durch die Höhenmessung transportiert werden				
Vorbedingung	<ul style="list-style-type: none"> Anlage befindet sich im Betriebsmodus 				
Input Daten	Werkstücke, Hözensensor				
Ablaufbeschreibung	Aktion	Erwartung	Erfüllt		
	Nächstes Werkstück in der Reihenfolge am Anfang von FBM1 einlegen	Der Motor läuft an FBM1 und FBM2 jeweils in langsamer Geschwindigkeit, wenn sich das Werkstück darunter befindet			

Test ID	T7	Getestete Anforderungen	ANF09, ANF10, ANF11, ANF15	Priorität	<i>normal</i>
Beschreibung	Bei voller Rutsche soll am anderen FBM aussortiert werden. Wenn beide Rutschen voll sind, soll der Betrieb so lange weiter laufen, bis nicht mehr aussortiert werden kann.				
Vorbedingung	<ul style="list-style-type: none"> Anlage befindet sich im Betriebsmodus Als nächstes Werkstück wird WS-BuM erwartet Die Rutschen an FBM1 und FBM2 sind nicht voll 				
Input Daten	Werkstücke, Höhensensor				
Ablaufbeschreibung	Aktion		Erwartung		Erfüllt
	Rutsche an FBM2 voll machen		An FBM2 blinkt die gelbe Lampe, die LED Q1 leuchtet		
	Werkstück WS-OB auflegen		Werkstück wird an FBM1 aussortiert		
	Rutsche an FBM2 frei machen und FBM1 voll machen		An FBM1 blinkt die gelbe Lampe, die LED Q2 leuchtet		
	Flaches Werkstück am Anfang von FBM1 einlegen		Werkstück wird an FBM2 aussortiert		
	Beide Rutschen voll machen und WS-BuM einlegen		Werkstück wird bis an das Ende von FBM2 befördert		
	Flaches Werkstück am Anfang von FBM1 einlegen		Wenn das Werkstück an der Weiche von FBM2 angekommen ist, geht die Anlage in den Fehlerzustand		
	Reset-Taster drücken		In der Konsole erscheint die Meldung, dass die Rampe erst frei gemacht werden muss. Die Anlage verbleibt im aktuellen Zustand		
	Rampe an FBM2 frei machen und Start drücken		Das Werkstück wird aussortiert		

Test ID	T8	Getestete Anforderungen	ANF04, ANF14	Priorität	<i>normal</i>
Beschreibung	Erkennung geflippter Werkstücke und Aussortierung				
Vorbedingung	<ul style="list-style-type: none"> Anlage befindet sich im Betriebsmodus Als nächstes Werkstück wird WS-BuM erwartet 				
Aktion	<p>Ein Werkstück <WS-BuM> an LBA auf FBM 1 gelegt.</p> <p>Das Werkstück wird bei der Übergabe auf FBM2 geflippt.</p>				
Erwartung	Das Werkstück wird bei FBM2 aussortiert.				
Erfüllt					

Test ID	T9	Getestete Anforderungen	ANF09, ANF24	Priorität	<i>normal</i>
Beschreibung	Wenn die Rutschen bei beiden Anlagen voll sind und ein Werkstück aussortiert werden muss, wechseln beide FBM in den Error-Zustand				
Vorbedingung	<ul style="list-style-type: none"> Anlage befindet sich im Betriebsmodus. Beide Rutschen sind voll. 				
Aktion	Aktion	Erwartung	Erfüllt		
	Ein Werkstück < WS-F > an LBA auf FBM 1 gelegt.	Das Werkstück wird zu Ende des FBM2 transportiert			
	Ein Werkstück < WS-F > an LBA auf FBM 1 gelegt.	Das Werkstück wird bis zur Weiche des FBM2's transportiert, Motor stoppt bei beide FBM und die rote Lampen blinken auf 1 hz.			
	Reset Taster wird gedrückt	Rote Lampen blinken auf 0.5 hz			
	Rutsche 1 und 2 wird leer gemacht. Start Taster gedrückt	Das Werkstück wird an FBM2 aussortiert. Das System befindet sich wieder im Betriebszustand (grüne Lampen leuchten)			

Test ID	T10	Getestete Anforderungen		Priorität	<i>kritisch</i>
Beschreibung	Wenn die Verbindung zwischen beiden Anlagen getrennt wird, dann stoppt der Motor bei beiden FBM.				
Vorbedingung	<ul style="list-style-type: none"> Anlage befindet sich im Betriebsmodus. 				
Aktion	Aktion		Erwartung	Erfüllt	
	Ein Werkstück < WS-F > an LBA auf FBM 1 gelegt. Wenn das Werkstück bei FBM2 ankommt wird ein weiteres Werkstück < WS-F > an LBA auf FBM1 gelegt es wird dann sofort die Verbindungs-Kabel zwischen FBM1 und FBM2 getrennt.		Der Motor bei beiden FBM hält an.		

6.3 Testprotokolle und Auswertungen

6.3.1 Unit Tests und Integrationstest

Es wurden Unit Test und in der finalen Phase vor der Abnahme Integrationstests hinzugefügt, die den korrekten Ablauf der MainFSM im Modus Running testet. Dafür wurde die GoogleTest Suite verwendet.

Finished after 0,562 seconds		
Runs: 32	Errors: 0	Failures: 0
<div><div>UnitTest_HeightSensor (0.069 s)</div><div><div>InitialStateAfterStartup (0.003 s)</div><div>WaitForWorkpieceWhenMotorRunning (0.007 s)</div><div>HighDetected (0.011 s)</div><div>FlatDetected (0.014 s)</div><div>HighWithHoleDetected (0.01 s)</div><div>UnknownDetected (0.004 s)</div><div>CalculateAverageAndMaxValue (0.01 s)</div><div>CalculateAverageAndMaxValueWhenMotorStoppedInBetween (0.01 s)</div></div></div> <div><div>UnitTest_Actuators (0.012 s)</div><div><div>GreenLampOnOffFlashing (0.006 s)</div><div>MotorFastSlowStopped (0.006 s)</div></div></div> <div><div>UnitTest_WorkpieceManager (0.0 s)</div><div><div>InitiallyNoWorkpiecesAddNewAndIncrementId (0.0 s)</div></div></div> <div><div>IntegrationTest_Running (0.204 s)</div><div><div>RampBlockedWhenRunningStarted (0.027 s)</div><div>NewWorkpieceInsertedStartMotor (0.023 s)</div><div>WorkpieceGetHeadOfArea (0.031 s)</div><div>WorkpieceDefaultValues (0.015 s)</div><div>SetHeightResult (0.018 s)</div><div>SetMetalResult (0.018 s)</div><div>WorkpieceInOrderLetPass (0.018 s)</div><div>WorkpieceNotInOrderSortOutAtFBM1 (0.035 s)</div><div>WorkpieceNotInOrderSortOutAtFBM2 (0.019 s)</div></div></div> <div><div>UnitTest_MainFSM (0.224 s)</div><div><div>StartStateStandby (0.009 s)</div><div>StateRunningAfterStartPressedShort (0.023 s)</div><div>StateServiceModeAfterStartPressedLong (0.072 s)</div><div>StateStandbyAfterRunningAndStopPressed (0.027 s)</div><div>StateEStopAfterRunningAndEStopPressed (0.033 s)</div><div>ErrorManualSolvedAfterRunning (0.031 s)</div><div>ErrorSelfSolvedAfterRunning (0.029 s)</div></div></div> <div><div>UnitTest_MotorFSM (0.053 s)</div><div><div>StartStateStopped (0.006 s)</div><div>StoppedToFastToStopped (0.013 s)</div><div>StoppedToSlowToStopped (0.013 s)</div><div>FastToSlow (0.009 s)</div><div>SlowToFast (0.012 s)</div></div></div>		

6.3.2 Systemtests

Wir haben auch Systemtests definiert, die am Komplettsystem ausgeführt wurden. Dadurch konnten wir gut sehen, was bereits funktioniert und wo es noch etwas zu optimieren gibt bzw. Bugfixing.

Erster Durchlauf:

Name	Status	Actual End Date
ESEP-T13 (1.0) - Aussortierung Mit Auswerfer Und Weiche Möglich	PASS	23/Jun/23 7:32 Pm
ESEP-T18 (1.0) - Anzeige Sinnvoller Hinweise Zur Bedienung	PASS	21/Jun/23 7:57 Pm
ESEP-T5 (1.0) - Erkennung Der Werkstücktypen An Der Höhenmessung	PASS	21/Jun/23 7:56 Pm
ESEP-T9 (1.0) - Nicht-Flache Werkstücke Sollen An FBM2 Aussortiert Werden.	PASS	21/Jun/23 7:56 Pm
ESEP-T16 (1.0) - Umgang Mit Geflippten Werkstücken	FAIL	21/Jun/23 7:55 Pm
ESEP-T10 (1.0) - Auf FBM2 Darf Sich Maximal Ein Werkstück Befinden	PASS	21/Jun/23 7:54 Pm
ESEP-T8 (1.0) - Aussortierung Flacher Werkstücke An FBM1	PASS	21/Jun/23 7:51 Pm
ESEP-T17 (1.0) - Flache Werkstücke Sollen An FBM1 Aussortiert Werden	PASS	21/Jun/23 7:51 Pm
ESEP-T15 (1.0) - Konfiguration Der Sortierreihenfolge Und Kalibrierwerte	PASS	21/Jun/23 7:50 Pm
ESEP-T7 (1.0) - Erzeugung Eines Fehlers An FBM2, Wenn Beide Rutschen Voll Sind	FAIL	21/Jun/23 7:44 Pm
ESEP-T6 (1.0) - Erzeugung Eines Fehlers An FBM1, Wenn Beide Rutschen Voll Sind	FAIL	21/Jun/23 7:44 Pm
ESEP-T14 (1.0) - Der Motor Stoppt, Wenn Sich Kein Werkstück Auf Dem Förderband Befindet	PASS	21/Jun/23 7:44 Pm
ESEP-T4 (1.0) - Änderung Der Sortierung Bei Voller Rutsche	PASS	21/Jun/23 7:43 Pm
ESEP-T12 (1.0) - Fehlerbehandlung Bei Abgebrochener Kommunikation	FAIL	21/Jun/23 7:29 Pm
ESEP-T3 (1.0) - Sortierung Nach Vorgegebener Reihenfolge	FAIL	21/Jun/23 7:28 Pm
ESEP-T2 (1.0) - Wechsel Der Betriebsmodi	PASS	21/Jun/23 7:19 Pm
ESEP-T11 (1.0) - Verbinden Der Beiden Anlagen Beim Start	PASS	21/Jun/23 7:19 Pm
ESEP-T1 (1.0) - EStopp	PASS	21/Jun/23 7:12 Pm

Zweiter Durchlauf:

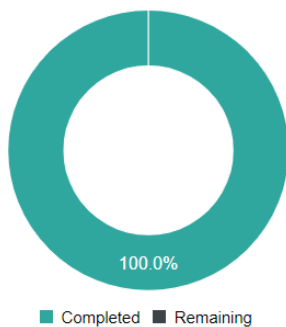
ESEP-T12 ist „BLOCKED“, da die Fehlerbehandlung für diesen Fall noch in Arbeit war

Name	Status	Actual End Date
ESEP-T9 (1.0) - Nicht-Flache Werkstücke Sollen An FBM2 Aussortiert Werden.	PASS	22/Jun/23 7:28 Pm
ESEP-T17 (1.0) - Flache Werkstücke Sollen An FBM1 Aussortiert Werden	PASS	22/Jun/23 7:28 Pm
ESEP-T13 (1.0) - Aussortierung Mit Auswerfer Und Weiche Möglich	PASS	22/Jun/23 7:28 Pm
ESEP-T8 (1.0) - Aussortierung Flacher Werkstücke An FBM1	PASS	22/Jun/23 7:28 Pm
ESEP-T7 (1.0) - Erzeugung Eines Fehlers An FBM2, Wenn Beide Rutschen Voll Sind	PASS	22/Jun/23 7:28 Pm
ESEP-T6 (1.0) - Erzeugung Eines Fehlers An FBM1, Wenn Beide Rutschen Voll Sind	PASS	22/Jun/23 7:27 Pm
ESEP-T5 (2.0) - Erkennung Der Werkstücktypen An Der Höhenmessung Und Metallsensor	PASS	22/Jun/23 7:24 Pm
ESEP-T14 (1.0) - Der Motor Stoppt, Wenn Sich Kein Werkstück Auf Dem Förderband Befindet	PASS	22/Jun/23 7:14 Pm
ESEP-T10 (1.0) - Auf FBM2 Darf Sich Maximal Ein Werkstück Befinden	FAIL	22/Jun/23 7:14 Pm
ESEP-T4 (1.0) - Änderung Der Sortierung Bei Voller Rutsche	FAIL	22/Jun/23 7:09 Pm
ESEP-T3 (1.0) - Sortierung Nach Vorgegebener Reihenfolge	PASS	22/Jun/23 7:03 Pm
ESEP-T12 (1.0) - Fehlerbehandlung Bei Abgebrochener Kommunikation	BLOCKED	22/Jun/23 7:03 Pm
ESEP-T1 (2.0) - EStopp Funktionalität	PASS	22/Jun/23 5:53 Pm

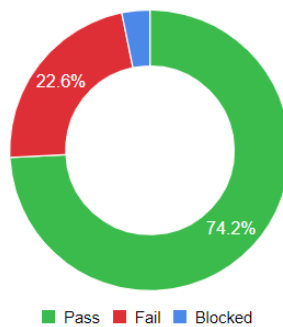
Ergebnis nach zwei Tagen Systemtest:

Test execution results (summary)

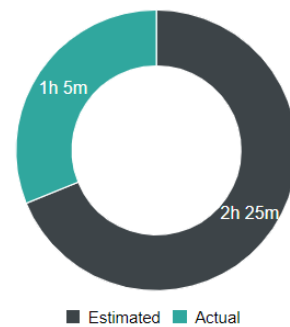
Overall Progress



Test Execution Results



Test Execution Effort



Die fehlerhaften Tests werden in der letzten Woche vor der Abnahme noch verstärkt angegangen und je nach Priorität behoben.

7 Technische Schulden

ID	Mangel	Vorgeschlagene Maßnahmen zur Behebung	Geschätzter Aufwand
MA-01	Bei getrennter Verbindung zwischen den Anlagen wird in Fehler gewechselt, der sich selbst beheben kann. Ein erneutes Verbinden über GNS ist jedoch aktuell nicht möglich und das Programm muss neu gestartet werden	Bei getrennter Verbindung ein Wiederverbinden anstoßen	6h
MA-04	Nach dem Start des Programms werden manchmal einige Events verarbeitet, die vom vorigen Lauf noch verarbeitet werden	Beim Start des Programms werden für 3 Sekunden Events vom Kernel empfangen. Der Fehler tritt jedoch in seltenen Fällen noch auf und das System ist z. B. nach dem Start im ServiceMode	4h
MA-07	Nach Absprache haben wir festgelegt, dass der erkannte Typ an FBM2 entscheidend ist für die Aussortierung bzw. durchlassen. Allerdings ändern wir unser nächst erwartetes Werkstück dann auch erst an der Weiche von FBM2. Dies hat zur Folge, dass Werkstücke die kurz nacheinander aufgelegt werden und eigentlich in der Reihenfolge sind, aussortiert werden	Bei Werkstücken, deren Typ an FBM1 schon final feststeht, können wir das nächst-erwartete Werkstück schon an FBM1 ändern.	1h
MA-11	Die CalMotorFSM (im ServiceMode) wurde aus Zeit- und Prioritätsgründen nicht umgesetzt	Eine Kalibrierung der Motorgeschwindigkeit ist für eine korrekte Funktionalität unserer Anlage nicht notwendig. Die korrekte Funktion des Motors wird im "Selbsttest Sensorik" geprüft. Falls der Kunde in einer weiteren Version das Feature "Erkennung von Verschwinden von Werkstücken" wünscht, kann die Kalibrierung des Motors umgesetzt werden.	3h
MA-12	Wenn die Werkstücke in einem bestimmten Abstand zueinander sind, wird das nachfolgende aussortiert, da die	Problem könnte durch Timing gelöst werden. Dieses Feature musste nach Absprache jedoch nicht umgesetzt werden.	20h

	Weiche zu gemacht wird, wenn das erste WS am Ende von FBM1 angekommen ist		
MA-13	Für einen reibungslosen Start müssen beide FBM's komplett neu gestartet werden. Falls nur ein FBM neu gestartet wird, verbindet es sich wieder zum anderen FBM, was zu unerwünschtem Verhalten führt	Wenn das Programm beendet wird, soll sich das Programm auf der anderen FBM auch beenden	4h
MA-14	In sehr seltenen Fällen geht die Verbindung zwischen den FBM verloren	Timeout beim Empfang von PulseMessages erhöhen	8h

7.1 Behobene Mängel

ID	Mangel	Umgesetzte Maßnahmen
MA-02	Die Erkennung eines Verbindungsverlustes zum anderen FBM funktioniert nur unzuverlässig und wird meist nur sehr spät erkannt.	Watchdog sendet alle 100ms einen Heartbeat und meldet einen Fehler, wenn innerhalb dieser Zeit kein Heartbeat des anderen FBM empfangen wurde
MA-03	Werkstück BOM (Hoch mit <u>kleiner</u> Bohrung, schwarz) wird manchmal als OB (Hoch ohne Bohrung) erkannt	Die Höhenmessung wurde optimiert (Ausreißer werden herausgefiltert, Laufband wird erst nach 5 Messwerten in Folge erkannt)
MA-05	An FBM2 wird der durchschnittliche Höhenmesswert angezeigt. Nach Absprache soll der maximale Wert und mittlere Wert angezeigt werden.	Ausgabe des mittleren und maximalen Wert zu aufwendig, da bei PulseMessages nur ein int-Feld verschickt werden kann. Deshalb wird wie an FBM1 nur der mittlere Wert ausgegeben
MA-06	Im Service-Mode "Selbsttest Aktorik" wird aktuell keine Unterscheidung gemacht, ob ein Auswerfer montiert ist. Dieser bleibt während des Tests eingefahren und fährt kurz aus, wenn die Funktion der Aktorik mit "Start" bestätigt wurde	Im Selbsttest wird geprüft, ob ein Auswerfer montiert ist. Wenn ja, wird dieser ausgefahren. Eine Weiche wird geöffnet.
MA-08	Nach dem Fehlerzustand wird wieder in den Ruhezustand gewechselt anstatt in den letzten Zustand	Der Fehler lässt sich erst quittieren, wenn die Rutsche an FBM2 frei ist. Nachdem der Fehlermodus verlassen wurde, wird der laufende Betrieb fortgesetzt. Das aktuell an der Rutsche befindliche Werkstück wird aussortiert.
MA-09	Im ServiceMode beim Sensorik Selbsttest tritt ein SIGSEV auf, jedoch meist an unterschiedlichen Stellen. Der Fehler tritt im Integrationstest nicht auf	Ursache waren undefinierte Member durch fehlerhafte Speicherallokierung in der „ServiceMode State“-Klasse.
MA-10	Bei anliegender Warnung dauert es ca. eine Sekunde nachdem Stop gedrückt wurde bis der Zustand gewechselt wird.	Ursache bereits gefunden. Der "Blinking Thread" muss so angepasst werden, dass der Thread sofort terminiert sobald die Anfrage kommt

8 Lessons Learned

Die Ziele die für wichtige Zwischenschritte (Milestones) im Projekt gesetzt wurden, müssen unbedingt eingehalten werden. In jedem Fall sollte immer ein Teammeeting einberufen werden, in dem besprochen wird, wie wir damit umgehen und evtl. mehr Support anfragen bei anderen Teammitgliedern.

Es ist sehr wichtig, sich intensiv mit der Anlage und Blackberry QNX auseinanderzusetzen. Das haben vor allem die Teammitglieder bemerkt, die das Projekt schon zum zweiten Mal machen.

Wir haben auch gelernt, dass es wichtig ist, die Funktionen einzelner Softwarekomponenten genauer zu spezifizieren, da während des Projekts oft Funktionalitäten nachgeliefert werden mussten, da sie zu Beginn nicht bedacht wurden.

Sehr hilfreich war auch die Nutzung von GoogleTest. Es muss zwar relativ viel Aufwand betrieben werden, um Mocks zu erstellen. Allerdings lohnt es sich im Vergleich dazu, wenn am realen System getestet werden muss.

9 Anhang

9.1 Glossar

Abkürzung	Bedeutung
Anlage	FBM1 und FBM2
FBM1	Erstes Förderband Modul (vorderes)
FBM2	Zweites Förderband Modul (hinteres)
EVM	EventManager

9.2 Abkürzungen

Abkürzung	Bedeutung
HAL	Hardware Abstraktion Layer.
FBM	Förderband Modul (Eine gesamte Anlage mit Beaglebone und Hardware)
FB	Förderband eines FBM
Betriebsmodus	Die Anlage sortiert die Werkstücke.
WS	Werkstück, der auf die Anlage zur Sortierung gelegt wird
LB	Light-Barrier (Lichtschranke)
LBA	Light-Barrier Start
LBW	Light-Barrier Weiche
LBE	Light-Barrier Ende

LBR	Light-Barrier Rutsche
BuM	Werkstücktyp: Hoch mit Bohrung und Metall
BoM	Werkstücktyp: Hoch mit Bohrung ohne Metall
OB	Werkstücktyp: Hoch ohne Bohrung
MD	Metal Detector
HM	Höhen Messsensor

9.3 **Abbildungsverzeichnis**

-