Requirements / Design and Test Documentation

(RDT)

Version 0.9

ESEP – Praktikum – WS 22/23

Dichte, Daniel, MatrikelNr, daniel.dichte@haw-hamburg.de

Leuendorf, Björn, 2506436, bjoern.leuendorf@haw-hamburg.de

Oehlers, Johannes, 2574219, johannes.oehlers@haw-hamburg.de

Kießler, Pascal, 2265544, pascal.kiessler@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 | 2022-10-06 | PKI | Requirements erarbeitet |
| 0.5 | 2022-10-13 | LDF | RDTs zusammengefasst und aufgeräumt, Glossar |
| 0.6 | 2022-10-26 | ALL | Kapitel 3 bearbeitet und ersten Abnahmetest |
| 0.7 | 2022-11-18 | PKI | Vorschläge von Enrico Christophers eingebracht |
| 0.8 | 2022-12-09 | LDF | FSM eingefügt |
| 0.9 | 2022-12-14 | LDF | Kleine anpassungen |
| 0.10 | 2023-01-02 | LDF |  |

Legende für Kürzel der Änderungshistorie

LMN: Prof. Dr. Thomas Lehmann  
DAI: Prof. Dr. Zhen Ru Dai  
PKI: Pascal Kießler  
LDF: Björn Leuendorf  
JOE: Johannes Oehlers  
DDI: Daniel Dichte  
ALL: Bearbeitung durch DDI, JOE, LDF, PKI

Inhaltsverzeichnis

[1 Teamorganisation 3](#_Toc123561544)

[1.1 Verantwortlichkeiten 3](#_Toc123561545)

[1.2 Absprachen im Team 3](#_Toc123561546)

[1.3 Repository-Konzept 3](#_Toc123561547)

[2 Projektmanagement 4](#_Toc123561548)

[2.1 Prozess 4](#_Toc123561549)

[2.2 Projektplan 4](#_Toc123561550)

[2.3 Risiken 4](#_Toc123561551)

[2.4 Qualitätssicherung 5](#_Toc123561552)

[3 Problemanalyse 5](#_Toc123561553)

[3.1 Absprachen mit den Betreuern Enrico Christophers und Prof. Dr. Franz Korf 5](#_Toc123561554)

[3.2 Systemebene 6](#_Toc123561555)

[3.2.1 Stakeholder 6](#_Toc123561556)

[3.2.2 Systemkontext 6](#_Toc123561557)

[3.2.3 Use Cases / User Stories 6](#_Toc123561558)

[3.3 Technischer Prozess 11](#_Toc123561559)

[3.4 Softwareebene 11](#_Toc123561560)

[3.4.1 Systemkontext 11](#_Toc123561561)

[3.4.2 Anforderungen 11](#_Toc123561562)

[3.4.3 Safe State 12](#_Toc123561563)

[4 Design 13](#_Toc123561564)

[4.1 Software Architektur 13](#_Toc123561565)

[4.2 Datenmodellierung 13](#_Toc123561566)

[4.3 Verhaltensmodellierung 14](#_Toc123561567)

[4.3.1 Eingabealphabet 14](#_Toc123561568)

[4.3.2 State Machiene 17](#_Toc123561569)

[5 Implementierung 24](#_Toc123561570)

[6 Testen 24](#_Toc123561571)

[6.1 Testplan 24](#_Toc123561572)

[6.2 Testszenarien/Abnahmetest 24](#_Toc123561573)

[6.3 Testprotokolle und Auswertungen 26](#_Toc123561574)

[7 Lessons Learned 26](#_Toc123561575)

[8 Anhang 28](#_Toc123561576)

[8.1 Glossar 28](#_Toc123561577)

[8.2 Abkürzungen 30](#_Toc123561578)

# Teamorganisation

Hier wird im Kurzem erläutert, wer für welchen Bereich zuständig ist und wie Absprachen und das angelegte Repository aussehen.

## Verantwortlichkeiten

|  |  |
| --- | --- |
|  | Ansprechpartner |
| Projektmanager | Daniel |
| Testmanager | Pascal |
| SW-Architekt | Johannes |
| Protokollant | Björn |

## **Absprachen im Team**

Wir treffen uns als Team regelmäßig und verfolgen dabei Ansätze wie es bei Scrum üblich ist. Einmal in der Woche findet ein festes, wöchentliches Meeting (Weekly) statt. Dieser Termin ist immer auf Dienstag gelegt. Bei diesem Termin besteht, sofern nicht anders abgesprochen, Anwesenheitspflicht, mit Absprache auch aus dem Homeoffice. In diesem Meeting wird einmal der aktuelle Arbeitsstand von jedem Teammitglied berichtet und eine Art Retrospektive abgehalten. Nach dieser gehen wir in die Planung für den Zeitraum bis zum nächsten Weekly. Hier werden sich die weiteren Aufgaben angeschaut und konkretisiert, bzw. auch verteilt im Team. Des Weiteren werden ein bis zwei weitere Meetings die nicht zeitlich festgelegt sind in der Woche im Weekly geplant, an denen man Zwischenstände absprechen und Probleme ansprechen kann. Hier können dann auch diese Probleme eventuell direkt gemeinsam gelöst werden. Alle Termine die geplant sind und wurden werden in einem extra dafür erstellten Kalender festgehalten. Des Weiteren werden Arbeitszeiten für Aufgaben die geleistet wurden getrackt und in einer dafür erstellten Excel-Tabelle eingetragen um nachvollziehen zu können wer wann an welcher Aufgabe gearbeitet hat.

## Repository-Konzept

In dem Repository wird es die folgende Ordnerstruktur geben.

Es gibt einen Ordner, in dem die verschiedenen Dokumente liegen. Dieser Ordner befindet sich in einer Cloud, zu der alle Teammitglieder Zugriff haben. In diesem sind das Lasten-/ Pflichtenheft abgelegt. Auch konkrete Aufgabenbeschreibung, Modellierungen und Ähnliches werden in dem Ordner in der Cloud abgelegt und sind zugänglich für jeden Im Team. Das Repository wird via Git-lab (HAW-Hamburg) verwaltet. In diesem Repository wird der gesamte Quellcode verwaltet und bearbeitet.

In dem zuvor erwähnten Git-lab Repository gibt es einen Master-Branch, welcher immer funktionsfähig bzw. immer in einem sauberen Zustand ist. Das bedeutet, der Code muss kompilierbar und auf der Festo ausführbar sein. Auf den Master-Branch wird nur gepushed, wenn es zu dem Code ein Review gibt. Das heißt es wird mindestens mit einem vier Augen Prinzip gearbeitet. Nachdem ein Pull-Request erstellt worden ist, muss dieser durch einen Reviewer approved werden. Erst nachdem das Approve gegeben wurde, kann der Branch gemerged werden. Es wurde sich hierbei ein zweites Mal abgesichert. Wenn ein neues Feature oder eine Entwicklung erfolgreich durch das Review gekommen ist und completed werden kann, wird diese zuerst in einen Dev-Branch gemerged. Dieser Dev-Branch bildet die Schnittstelle zwischen dem Master und neuen Entwicklungen. Erst wenn die zusammenspielenden Entwicklungen laufen und ohne Bugs funktionieren, wird ein Merge-Request vom Dev auf den Master erstellt. Dieser Request muss erneut gesondert reviewed werden. Nachdem das Review erfolgreich war und das Approve gegeben wurde, kann dieser Dev-Branch in den Master gemerged werden.

Von dem Dev-Branch werden dann weitere Branches abgeleitet an denen das Team arbeiten wird. Es wird zwei Arten von Branches geben. Die eine Art mit dem Präfix „feat/…“ repräsentieren dabei Neuentwicklungen. Dabei steht das „feat“ für Feature und die Punkte repräsentieren den Namen des Features. Ein Feature kann auch eine schon bestehende Entwicklung erweitern. Die zweite Art von Branches sind Bugs. Dafür wurde sich auf den Präfix „bug/…“ geeinigt. Auch hier besteht der Präfix für einen Bug an dem gearbeitet wird und die Punkte für den Namen des Bugs.

Wenn ein Merge-Request erstellt wird, reviewed mindestens einer der anderen Entwickler im Team den Code und testet bei sich diese Entwicklung. Sollte alles korrekt funktionieren, wird der Merge-Request „approved” und der Ersteller des Merge-Requests schließt diesen, wodurch der Dev-Branch aktualisiert wird. Für jedes Review muss eine kleine Dokumentation vorliegen, in welcher festgehalten wird, was exakt geändert und hinzugefügt wurde, damit der Reviewer schnell weiß, worum es geht und was zu prüfen ist. Eine kurze Dokumentation des Reviews wird dem Merge beigefügt. Durch diese Struktur kann später nachvollzogen werden, was genau passiert ist und wo eventuelle Probleme ihren Ursprung haben.

# Projektmanagement

In diesem Kapitel werden organisatorische Punkte beschrieben und festgelegt.

## Prozess

Es wird eine Art abgewandeltes Scrum eingesetzt. Das bedeutet, dass wöchentliche Meetings am Dienstag abgehalten werden, damit wir zwischen den ESEP-Terminen genügend Zeit haben, um beim Meeting auch produktiv sein zu können. Retrospektive fällt weg und ein Kanban Board wird vorbereitet, auf welchen wir die Aufgaben im Backlog Tracken. Die Sprints beschreiben die Zeit von einem Praktikumstermin zu dem Nächsten.

## Projektplan

Zeiten werden über ein Excel-Tool erfasst. Geplant ist, im Zeitplan niemals hinter der Deadline hinterher zu hängen. Sollte man nach Scrum arbeiten, werden in einem Backlog die entsprechenden User Storys erstellt, die in einem Sprint geplant werden und abgearbeitet werden.

## Risiken

Wenn Sie eine Risk-Matrix für Ihr Projekt erstellen, dann fügen Sie die Tabelle hier ein.

## Qualitätssicherung



Qualitätssicherung via Git. Die Features werden mit dem 4 Augen Prinzip überprüft. Das heißt, der Entwickler prüft vor dem Merge-Request die korrekte Funktion und erstellt ein Update-Log, damit das zweite Teammitglied als Prüfer das Codereview erstellen kann. Jederzeit muss darauf geachtet werden, dass wir uns nach wie vor im Zeitplan befinden. Zu erkennen ist das an den Meilensteinen und visualisiert auch im Ganttdiagramm, welches mit den Sprints und Aufgaben ausgefüllt wurde. Aufgabenstellungen sollen auch zwischendurch noch einmal durchgegangen werden. Das RDT wird ausführlich ausarbeitet, damit rechtzeitig das erste Review durchgeführt werden kann. Ein Glossar wird möglichst früh ausgearbeitet und kontinuierlich um neue Begriffe erweitert.

Um zu prüfen, ob die entwickelte Software erwartungsgemäß läuft und nicht nur kompiliert, werden Abnahmetests erarbeitet, die auf den Anforderungen und Use Cases basieren. Die Abnahmetests werden akribisch durchgeführt und die Bugs in einem Kanban Board festgehalten, damit diese bei weiterer Entwicklung der Software nicht aus den Augen geraten. Der aktuelle Stand sämtlicher Bugs und ob einzelne Teammitglieder Hilfe bei der Behebung dieser Bugs benötigen, wird auch bei jedem Meeting besprochen.

# Problemanalyse

In diesem Kapitel werden Stichprobenartig die Absprachen mit den Betreuern festgehalten.

## Absprachen mit den Betreuern Enrico Christophers und Prof. Dr. Franz Korf

* Wenn kein Fehler vorliegt und der Reset button gedrückt wird soll nichts passieren und der klick auf den Button soll ignoriert werden.
* Der Motor der das Förderband ansteuert stoppt, wenn sich kein Werkstück mehr auf dem Laufband befinden sollte.   
  **(Ist kein WP mehr auf dem conveyor belt, soll der Motor stoppen)**
* Die Config die wir im Servoce modus ändern können wird im JSON Format geschrieben.
* Es sollen Werkstücke von dem Festotransfersystem erkannt werden. Wenn der Gegenstand der auf dem Band liegt kein Werkstück ist, geht das Festotransfersystem in einen Fehlerzustand der wieder quittiert werden muss.  
  **(WPs sollen erkannt werden und wenn ein Gegenstand, der kein WP ist, auf dem conveyor belt liegt, soll erkannt werden, dass es KEIN WP ist. Dann gibt es einen Fehler.)**
* Die Lichtschranke für die Höhenmessung wird nicht beachtet.
* Wenn der E-stop gedrückt wird, wechselt das ganze Festotransfersystem in einen Safe state (Wird in Kapitel 3.4.3 erörtert). Wenn sich das Festotransfersystem in diesm Safe state befindet, muss alles in seine Ausgangslage gebracht werden. Wenn die Ausgangslage erreicht ist, kann der Resetbutton gedrückt werden, dass signalisiert dem Festotransfersystem das die Gefahr behoben wurde  
  **(Sollte der E-Stop gedrückt sein, wechseln wir in den safe state und mit dem Drücken des reset buttons wird bestätigt, dass alle WPs vom belt genommen wurden und die Gefahr behoben wurde)**
* Im Service Mode gibt es einen klar definierten Ablauf, welcher durchlaufen wird um die korrekte Funktionsweise der Hardware zu überprüfen
* In der Kalibrierung wird einmal die Art der Switch und die Zeit zwischen den Abschnitten mit Motor und Motor-Slow erfasst. Außerdem wird der Abstand vom Height sensor zum Belt gemessen und die config wird eingelesen.
* Ob ein WP dazugelegt wurde oder weggenommen wurde wird mit Hilfe von Timern erkannt.

## Systemebene

Die Anforderungen aus der Aufgabenstellung sind nicht vollständig. Die Struktur der nachfolgenden Kapitel soll Sie bei der Strukturierung der Analyse unterstützen.

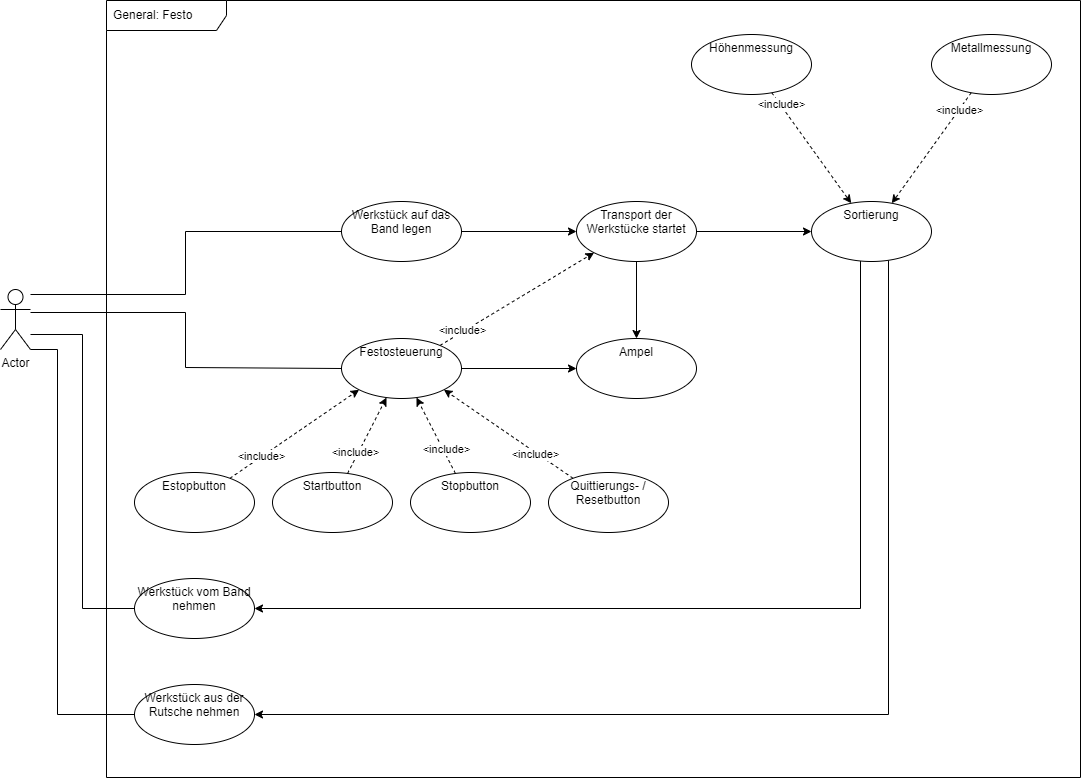
### Stakeholder

|  |  |
| --- | --- |
| Stakeholder | Interessen |
| Kunde | Funktionierende, fehlerfreie Software Absicherung gegen Fehler Kommentierter Code Wartbarer Code Software und Hardware greifen ineinander |
| Entwickler | Entwicklung der Software Kommentierter und Wartbarer Code Zufriedenheit des Kunden |
| Tester | Testen der Komponenten und des Systems |
| Berater | Entwicklung seiner Anforderungen Steht bereit für Fragen zum Projekt |

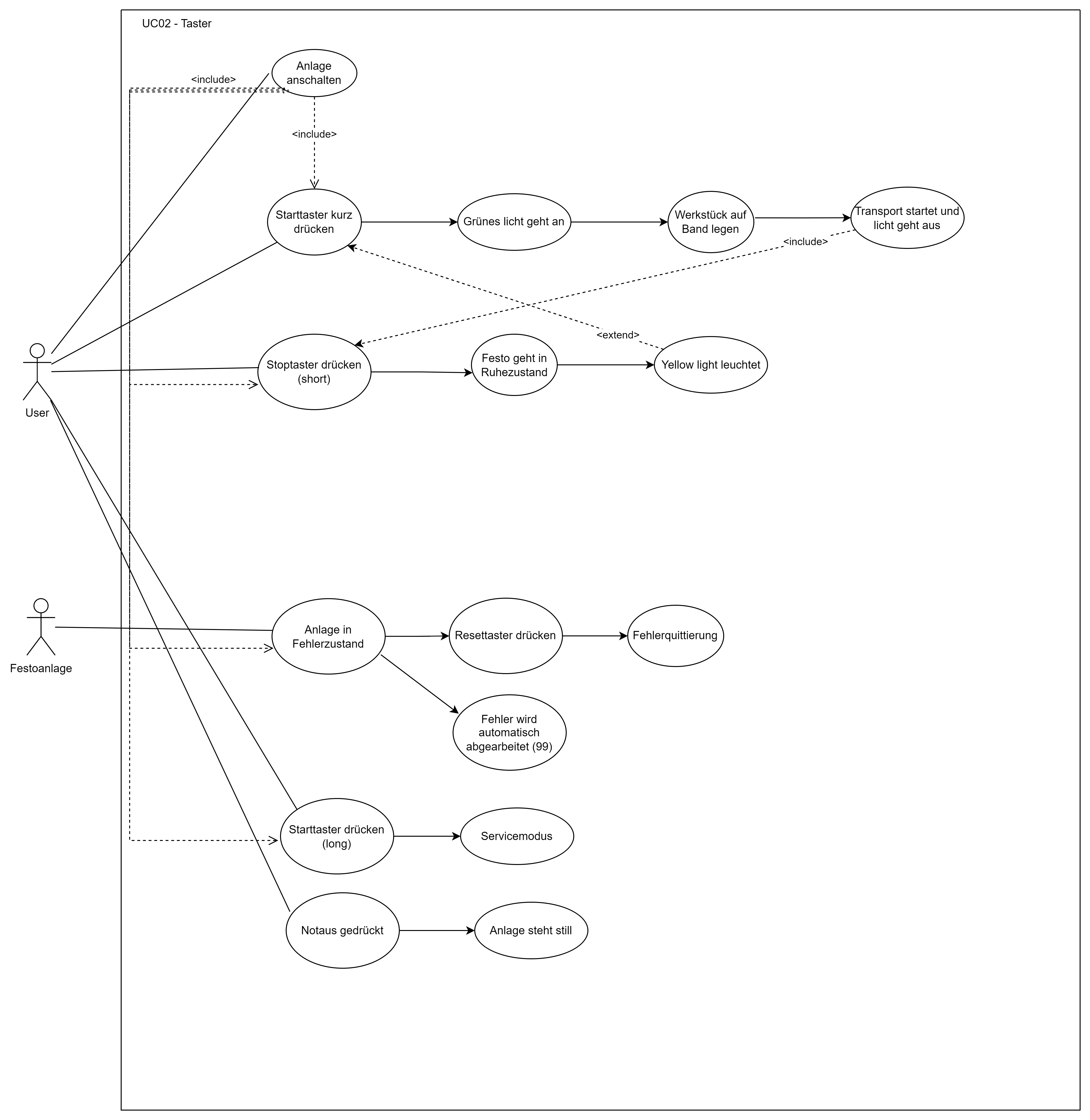
### Systemkontext

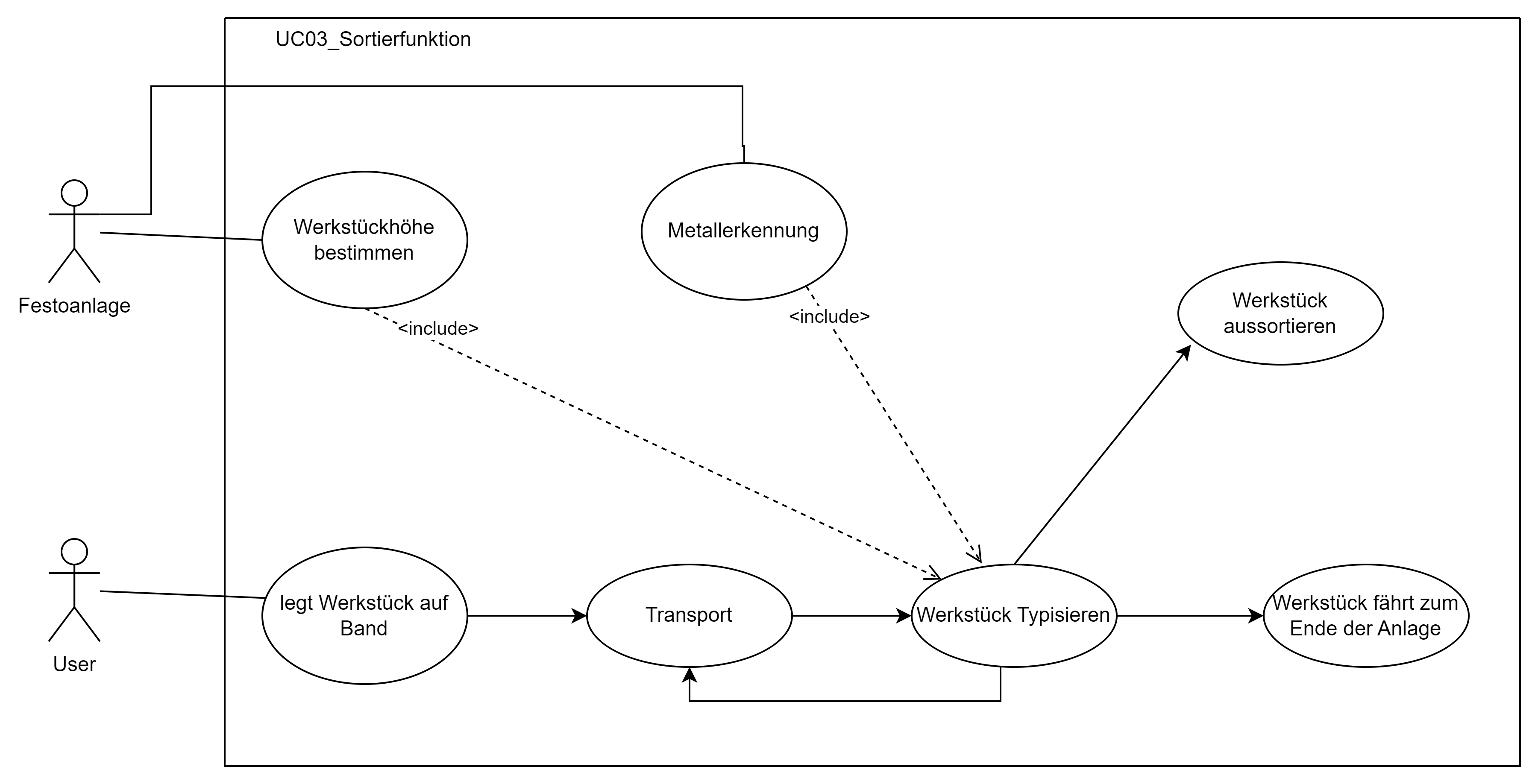
Der Systemkontext wird genauer beleuchtet in den Use cases.

### Use Cases / User Stories

Im Folgenden werden die Use cases als Diagramm festgehalten.

UC01:  
Dieser Use Case zeigt die grundlegenden Interaktionsmöglichkeiten des Users mit der Festo. Die folgenden Use cases dieses präzisieren, damit auch kleinteiligere Schritte beleuchtet werden. Diese Aufteilung wird vorgenommen, damit die Übersichtlichkeit nicht verloren geht.



UC02:  
UC02 beschränkt sich auf das Handling der Buttons. Je nach Button sollte sich die Festo anders verhalten.UC03:  
UC0 bezieht sich auf den generellen Sortieralgorithmus.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | UC\_02\_01 | **Name** | Inbetriebnahme der Anlage | **Priorität** |  |
| **Beschreibung** | Die Anlage wird gestartet und kann Werkstücke annehmen. Nach Auflegen eines Werkstücks an den Anfang Anlage startet das Förderband. | | | | |
| **Aktoren** | * Start button * Motor * Light-barrier-Inlet * Ampel | | | | |
| **Vorbedingung** |  | | | | |
| **Main flow** | 1. Start button wird gedrückt 2. Ampel schaltet auf Grün 3. Werkstück wird auf das Förderband gelegt | | | | |
| **Nachbedingung** | 1. Der Motor des ersten Förderbandmoduls der Anlage startet und das Werkstück wird befördert 2. Ampel geht aus | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | UC\_02\_02 | **Name** | Stoppen der Anlage | **Priorität** |  |
| **Beschreibung** | Während ein Werkstück auf dem Förderband transportiert wird, kann der Motor jederzeit gestoppt werden. | | | | |
| **Aktoren** | * Stopp button * Ampel | | | | |
| **Vorbedingung** | 1. UC\_02\_01 | | | | |
| **Main flow** | 1. Stopp button wird gedrückt | | | | |
| **Nachbedingung** | 1. Beide Motoren der Anlage werden gestoppt 2. Ampel schaltet auf Gelb | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | UC\_02\_03 | **Name** | Fehlerquittierung | **Priorität** |  |
| **Beschreibung** | Während der Laufzeit der Anlage kann es immer zu einem Fehler kommen. Ursache dafür können verschieden sein, von einem Falschen Werkstück, falsche Bedienung oder Ähnliches. Ziel dieses Use cases ist es die automatische Fehlerhandhabung- und Quittierung zu beschreiben. | | | | |
| **Aktoren** | * Reset button * Ampel * Motor | | | | |
| **Vorbedingung** | 1. UC\_02\_01 gedrückt 2. Ein Fehler ist aufgetaucht der quittiert werden muss | | | | |
| **Main Flow** | 1. Anlage wechselt in den Fehlerzustand “anstehend unquittiert” 2. Die Ampel beginnt Rot zu blinken (2Hz) 3. Der Reset button wird gedrückt 4. Der Fehler wird quittiert und Festo wechselt in den Zustand “anstehend quittiert” 5. Die Ampel schaltet auf Rot 6. Der Start Button wird kurz (<3Sekunden) | | | | |
| **Altern. Flow 1** | 4.1. Fehler behebt sich selber  5.1. Anlage geht in den Zustand “gegangen unquittiert”  6.1. Die Ampel beginnt Rot zu blinken (1Hz) | | | | |
| **Altern. Flow 2** | 5.2. Anlage kann Zustand beheben, weil Werkstück dazugekommen oder verloren | | | | |
| **Nachbedingung** | 1. Anlage geht in den Zustand zurück, in dem sie sich vor Auftreten des Fehlers befunden hat 2. Die Ampel geht aus. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | UC\_02\_04 | **Name** | Servicemodus starten | **Priorität** |  |
| **Beschreibung** | Die Anlage kann in den Servicemodus gehen. | | | | |
| **Aktoren** | * Start button * Ampel | | | | |
| **Vorbedingung** | 1. Anlage wurde angeschaltet und wartet auf Eingabe des Benutzers | | | | |
| **Main Flow** | 1. Der Start button wird lange (>3Sekunden) gedrückt, während die Anlage weder mit Kalibrierung noch mit Transport von Werkstücken beschäftigt, ist | | | | |
| **Nachbedingung** | 1. Die Festo geht in den Servicemodus 2. Die Ampel Blinkt Grün (1HZ) | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | UC\_02\_05 | **Name** | E-Stop switch handling (betätigen) | **Priorität** |  |
| **Beschreibung** | Bei einem Gefahrenfall kann jederzeit der E-Stop Schalter gedrückt werden. Anlage stoppt, wenn der E-Stop Schalter betätigt wurde. | | | | |
| **Aktoren** | * E-Stop Schalter * Ampel * Motor | | | | |
| **Vorbedingungen** | 1. UC\_02\_01 | | | | |
| **Main Flow** | 1. Drücken des E-Stop Schalter | | | | |
| **Nachbedingung** | 1. Die Motoren der Anlage stoppen 2. Die Ampel beginnt abwecheselnt Gelb und Rot zu blinken (1Hz) | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | UC\_03\_01 | **Name** | Werkstück Typisieren | **Priorität** |  |
| **Beschreibung** | Der Höhensensor misst einen Abstand, der nicht dem Abstand zu dem Förderband entspricht. Anhand der nun folgenden Messungen kann in drei unterschiedliche Werkstücktypen unterschieden werden: “Flach”, “ohne Bohrung” und “mit Bohrung”. Der Metallsensor misst ein Metallgegenstand. Anhand dieser Messung können die Werkstücke “mit Bohrung” in “Bohrung mit Metallkern” oder “Bohrung ohne Metallkern” typisiert werden. | | | | |
| **Aktoren** | * Höhensensor * Metallsensor | | | | |
| **Vorbedingung** | 1. UC\_02\_01 2. Das Werkstück wurde durch die Rotation des Förderbandes an den Höhensensor transportiert | | | | |
| **Main Flow** | 1. Die Höhe des Werkstücks wird ausgewertet 2. Der Metallsensor stellt fest, ob es sich um ein Werkstück mit Metall handelt | | | | |
| **Nachbedingung** | 1. Der Werkstücktyp wurde erkannt | | | | |

## Technischer Prozess

Die Pins der Komponenten sind in einer vorgegebenen und festgelegten Weise angeschlossen, die dem Aufgabenblatt entnommen werden kann.

Die einzige Interaktion zwischen Menschen und Festo ist, nach starten des Programms, das Auflegen des WPs auf dem Inlet der Festo1 und das Bedienen der Button und des Switches auf dem Kontrollfeld.

Das Band darf sich nur nach vorne bewegen und niemals rückwärts. Außerdem gibt es zwei Geschwindigkeiten, mit denen die Motoren das Belt bewegen können. Hier muss beim tracking der WPs aufgepasst werden.

Festo1 und Festo2 können zur Kommunikation mit einem Ethernet-Kabel verbunden werden.

Jede der möglichen Festos aus dem Labor ist minimal anders. Die Motoren laufen unterschiedlich schnell, die Höhensensoren haben unterschiedliche Abstände zum Band und die light-barrier haben alle eine andere Position auf dem Rahmen. Dies ist in dem calibration mode zu beachten.

Um eine Beschädigung der Weiche zu verhindern, muss diese möglichst schnell wieder geschlossen werden.

## Softwareebene

Der folgende Abschnitt beschreibt die Steuerung des technischen Systems die aus der Anforderungsanalyse entstanden ist.

### Systemkontext

Wie sieht der Kontext Ihrer Software aus? Wie erfolgt die Kommunikation mit Nachbarsystemen? Liste der ein- und ausgehenden Signale/Nachrichten.

Da es sich um ein Master-Master System handelt, wird es eine Kommunikation über die Dispatcher geben. Dies ist im Punkt „Software Architektur“ genauer erklärt.

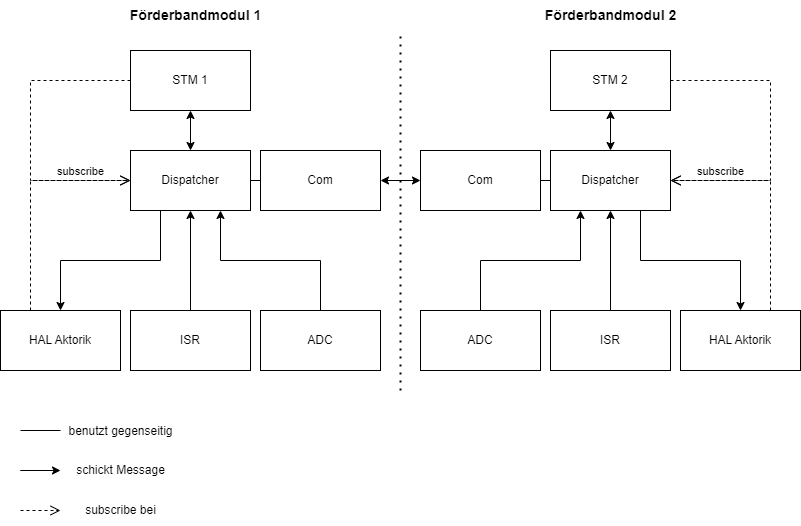
### Anforderungen

Es existiert eine Config-Datei in der die Sortierung vorgegeben wird. Der Aufbau des Systems wird erkannt und damit soll unterscheiden werden können, um was für einen Switch es sich bei der Festo1 und der Festo 2 handelt. Ein Switch kann sowohl eine Weiche als auch ein Auswerfer sein. Die HAL darf das Band nicht in die Fahrtrichtung drehen lassen, sondern muss immer in die Richtung drehen, die das Ende des Systems darstellt. Fehler und Warnungen sollen User schnellstmöglich erkennbar gemacht werden. Die Inkenntnisssetzung geschieht über verschiedene Lichtsignale durch die Ampel & LEDs des Systems. Die Betätigung des E-Stops sorgt für einen unverzüglichen Stopp des gesamten Systems, auch soll das System dann in einen Save State wechseln. Das System soll Werkstücke so sortieren können, wie es in der Config vorgegeben wird. Dabei ist die Aufgabe von der Festo1 die Höhe der Workpieces zu kontrollieren. Die Aufgabe von Festo2 ist es zu kontrollieren, ob die Accepted order aus der Config eingehalten wird und ob sich Werkstücke zwischen den Festo geflipped haben. Das System muss in der Lage sein, Interrupts über die ISR verarbeiten zu können. Die Quittierung von Fehlern soll über das Drücken des Reset buttons erfolgen. Das System stoppt, wenn der Stop Button betätigt wird. Warnungen, die die Möglichkeit bieten automatisch quittiert zu werden, sollten auch automatisch quittiert werden. Durch ein Long press des Start buttons, wird die Anlage in den Calibration mode versetzt. Während des regulären Betriebs des Systems sollte das grüne Licht der Ampel leuchten. Wenn das System sich in dem Servicemodus befindet, soll das grüne Licht blinken. Bei einer Wartung soll das gelbe Licht der Ampel blinken. Sollte ein Werkstück am Ende des Bands angekommen sein oder sich kein Werkstücke auf dem Band befinden, hört das selbige Band auf zu drehen.

### Safe State

# Design

## Software Architektur



Die Architektur ist nach dem Master-Master Prinzip aufgebaut. Die beiden einzelnen Förderbandmodule sind somit identisch modelliert. Signale von einem Modul werden als Interrupts in der ISR abgefangen und als Message an den Dispatcher geschickt. Anders empfängt die HAL Aktorik Messages vom Dispatcher, interpretiert diese und steuert das Modul an. Der Dispatcher erhält Messages und schickt diese an die für die Message relevanten Komponenten weiter. Die State Machine empfängt Messages vom Dispatcher. In dieser wird die Reaktion auf Message bestimmt, welche dann wieder als Message an den Dispatcher geschickt wird. Die Kommunikation der Module findet zwischen den Com-Klassen, die von den Dispatchern benutzt werden, statt. Der Dispatcher arbeitet nach dem Publish Subscribe Pattern. Aktorik und STM sind hierbei die Subscriber, die sich auf Events beim Dispatcher registrieren. Der ADC schickt selbstständig regelmäßig aktuelle Werte vom Höhensensor an den Dispatcher.

## Datenmodellierung

Bestimmen Sie das Datenmodell und dokumentieren Sie es hier mit Hilfe von UML Klassendiagrammen unter Beachtung der Designprinzipien. Die Modelle können mit Hilfe eines UML-Tools erstellt werden. Hier ist dann ein Übersichtsbild einzufügen.

Geben Sie eine kurze textuelle Beschreibung des Datenmodells und deren wichtigsten Klassen und Methoden an.

### Eingabealphabet

Overview

|  |  |
| --- | --- |
| conEstablished | Eine Verbindung zwischen den Förderbändern konnte hergestellt werden |
| conLost | Die Verbindung zwischen den Förderbändern ist unterbrochen worden |
| eStop\_Self | Der EStop der eigenen Anlage wurde gedrückt |
| eStop\_Self || eStop\_Ext | Ein oder beide Estops wurden reingedrückt |
| eStop\_Done [connection] | Reset wird kurz gedrückt, kein Estop ist gedrückt und die Verbindung wurde unterbrochen. |
| eStop\_Done [!connection] | Reset wird kurz gedrückt, kein Estop ist gedrückt die Verbindung ist noch vorhanden. |

Connection

|  |  |
| --- | --- |
| start\_lp | Start wird lang gedrückt |
| reset\_sp | Reset wird kurz gedrückt |
| reset\_lp | Reset wird lang gedrückt |
| start\_sp[loadedConf] | Start wird kurz gedrückt und eine Konfiguration für die Anlage ist vorhanden |
| stop\_sp | Stop wird lang gedrückt |
| error | In OperatingMode ist es zu einem Error gekommen |

E-Stop

|  |  |
| --- | --- |
| eStop\_Self\_pressed | Der Estop wurde auf der Anlage gedrückt, auf welcher das Programm läuft |
| eStop\_Self\_pressed | Der Estop wurde auf der Anlage gelöst, auf welcher das Programm läuft |
| eStop\_Self\_released [!Estop\_Ext] | Der eigene Estop wurde gelöst und der externe Estop ist es bereits auch |
| eStop\_Ext\_pressed | Der Estop wurde auf einer extern angeschlossenen Festo-Anlage gedrückt |
| eStop\_Ext\_released | Der Estop wurde auf einer extern angeschlossenen Festo-Anlage gelöst |
| eStop\_Ext\_released [!Estop\_Self] | Der externe Estop wurde gelöst und der eigene Estop ist es bereits auch |
| reset\_sp | Reset wird kurz gedrückt |
| reser\_sp [!connection && !Estop\_Self] | Reset wird kurz gedrückt, die Verbindung zur fremden Festo ist verloren & der eigene Estop wurde rausgezogen |

Calibration

|  |  |
| --- | --- |
| lb\_I | Lightbarrier Inlet durchbrochen |
| hs\_WP | Height Sensor erfasst Werkstück |
| calc\_done | Berechnung der cm/s ist abgeschlossen |

Service Mode

|  |  |
| --- | --- |
| loading\_completed | Laden der Konfigurationsdatei ist abgeschlossen |
| reset\_sp | Reset wird kurz gedrückt |

Error

|  |  |
| --- | --- |
| error\_gone | Der Fehler hat sich von selbst behoben |
| reset\_sp | Reset wird kurz gedrückt |
| reset\_sp [appeared] | Reset wird kurz gedrückt und es ist ein Werkstück aufgetaucht |
| reset\_sp [disappeared] | Reset wird kurz gedrückt und es ist ein Werkstück verschwunden |
| printed | Das Printen auf der Konsole ist abgeschlossen |
| start\_sp | Start wird kurz gedrückt |

Operating Mode

|  |  |
| --- | --- |
| lb\_I | Lightbarrier Inlet durchbrochen |
| empty\_Belt | Es befindet sich kein Werkstück mehr auf dem Band |
| Lb\_O || lb\_SL | Entweder die Lightbarrier Outlet oder auf der Slide durchbrochen |

Throw

|  |  |
| --- | --- |
| lb\_SW [WP\_expected] | Lightbarrier Switch durchbrochen und es wird ein Werkstück erwartet |
| lb\_SW\_free | Lightbarrier Switch ist wieder frei |
| in\_Order | Das aktuelle Werkstück entspricht der erwarteten Reihenfolge |
| out\_of\_Order | Das aktuelle Werkstück entspricht nicht der erwarteten Reihenfolge |
| flat | Das aktuelle Werkstück wurde in der Höhenmessung als Flach erkannt |
| slSelf\_full [festo == 1] | Die Rutsche von Anlage 1 ist voll und die aktuelle Anlage wurde als Festo 1 initialisiert |
| slExt\_full [festo == 1] | Die Rutsche von Anlage 2 ist voll und die aktuelle Anlage wurde als Festo 1 initialisiert |
| both\_full [festo == 1] | Beide Rutschen sind voll und die aktuelle Anlage wurde als Festo 1 initialisiert |
| sl\_free [festo == 1] | Beide Rutschen sind frei und in der Config wurde der Wert festo == 1 hinterlegt |
| slSelf\_full [festo == 2] | Die Rutsche von Anlage 1 ist voll und die aktuelle Anlage wurde als Festo 2 initialisiert |
| slExt\_full [festo == 2] | Die Rutsche von Anlage 2 ist voll und die aktuelle Anlage wurde als Festo 2 initialisiert |
| both\_full [festo == 2] | Beide Rutschen sind voll und die aktuelle Anlage wurde als Festo 2 initialisiert |
| sl\_free [festo == 2] | Beide Rutschen sind frei und in der Config wurde der Wert festo == 2 hinterlegt |
| signal\_receipted |  |

Measure Height

|  |  |
| --- | --- |
| hs\_WP | Height Sensor erfasst Werkstück |
| hs\_Belt | Height Sensor erfasst Belt |

Metal

|  |  |
| --- | --- |
| metal\_detected | Metall wurde erkannt |
| no\_metal | Es wird kein Metall mehr erkannt |

Slide

|  |  |
| --- | --- |
| slSelf\_free | Die Rutsche der Festo 1 ist frei und hat Platz für weitere Werkstücke |
| slSelf\_full | Die Rutsche der Festo 1 ist voll |
| slExt\_free | Die Rutsche der Festo 2 ist frei und hat Platz für weitere Werkstücke |
| slExt\_full | Die Rutsche der Festo 2 ist voll |
| wp\_expected | Es wird ein Werkstück vor der Rutsche erwartet |

Manage\_WP

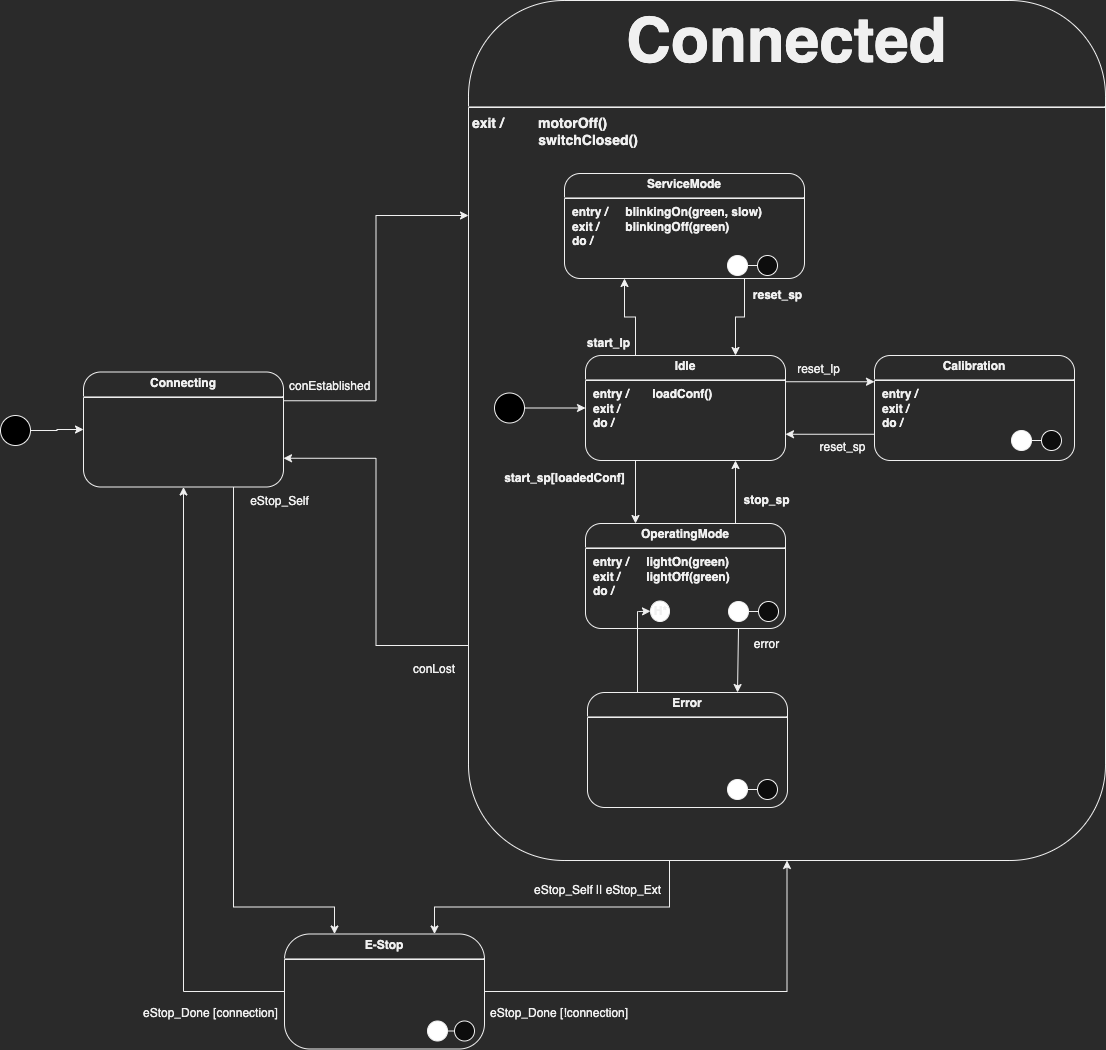
|  |  |
| --- | --- |
| lb\_I | Lightbarrier Inlet durchbrochen |

Watch

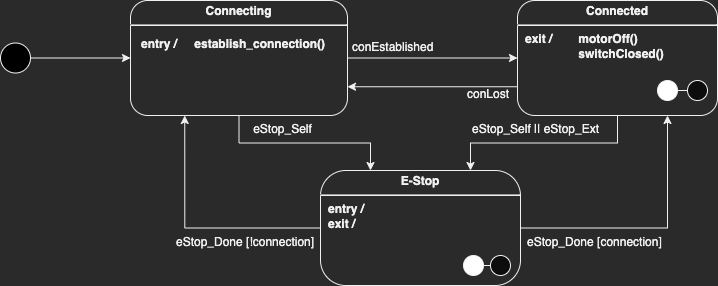
|  |  |
| --- | --- |
| height | Ein Werkstück wurde in der Höhenmessung erkannt |
| lb\_O | Lightbarrier Outlet durchbrochen |
| lb\_SW | Lightbarrier Switch durchbrochen |
| done | Prüfung auf zu früh oder zu spät ist abgeschlossen |

### State Machiene

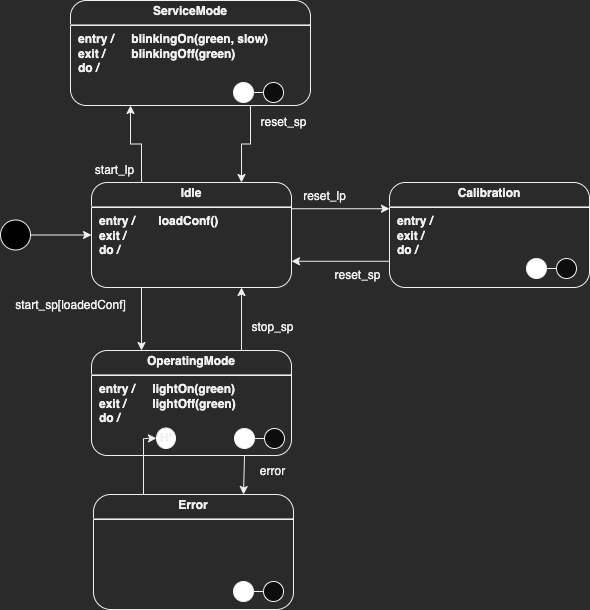
Big Overview



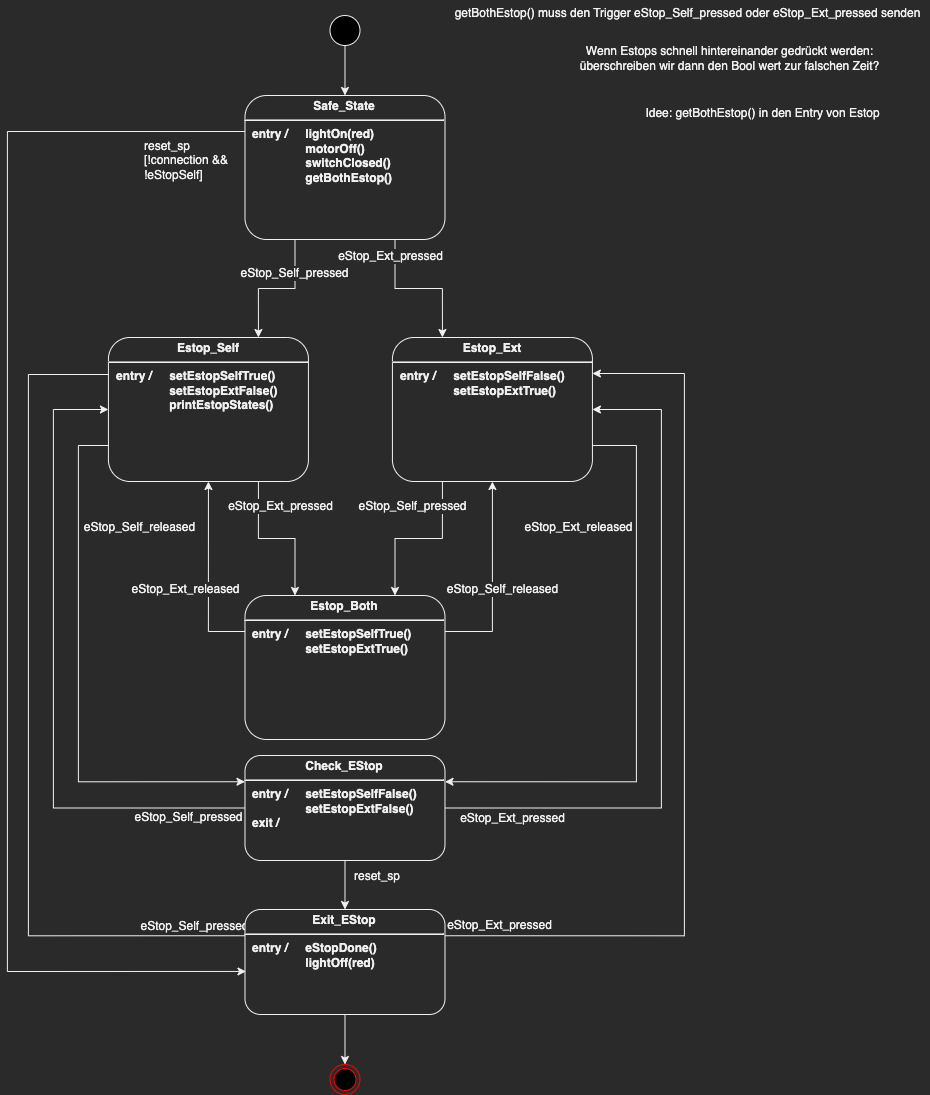
Overview



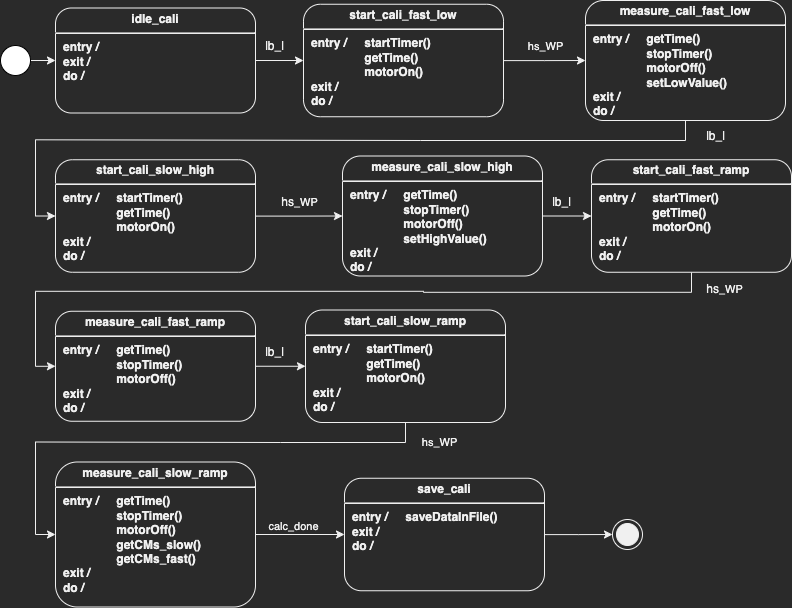
Connection



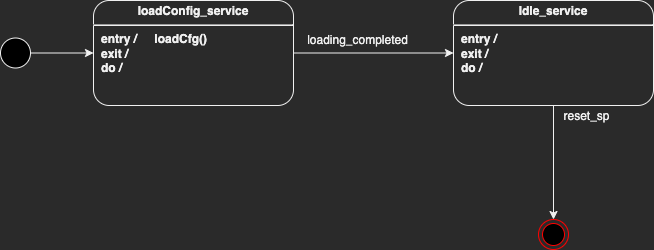
E-Stop



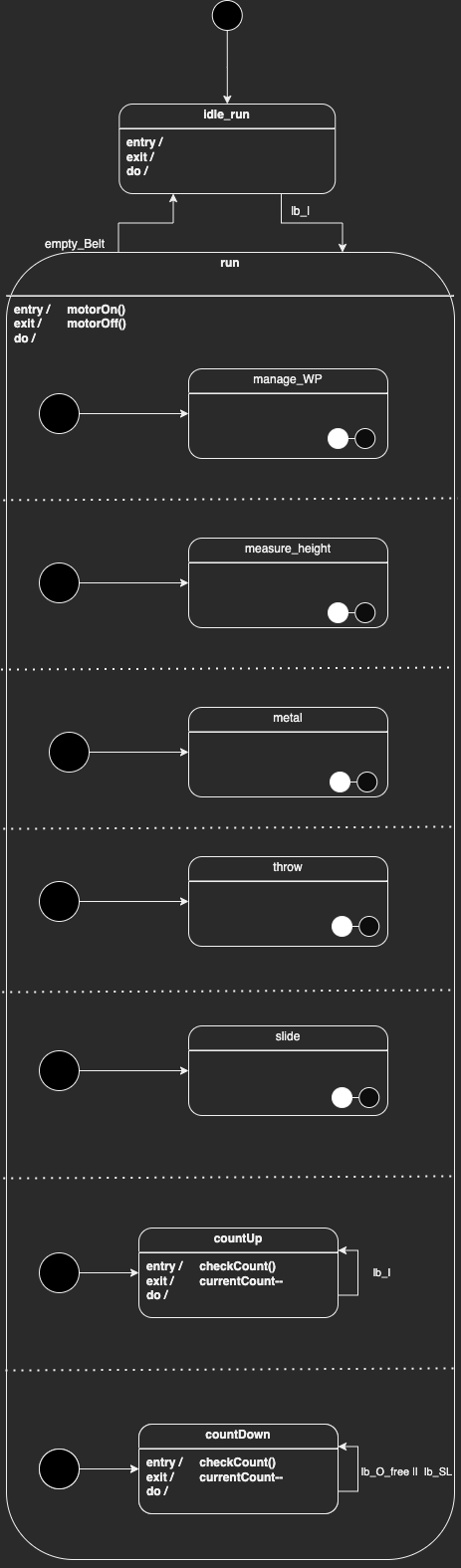
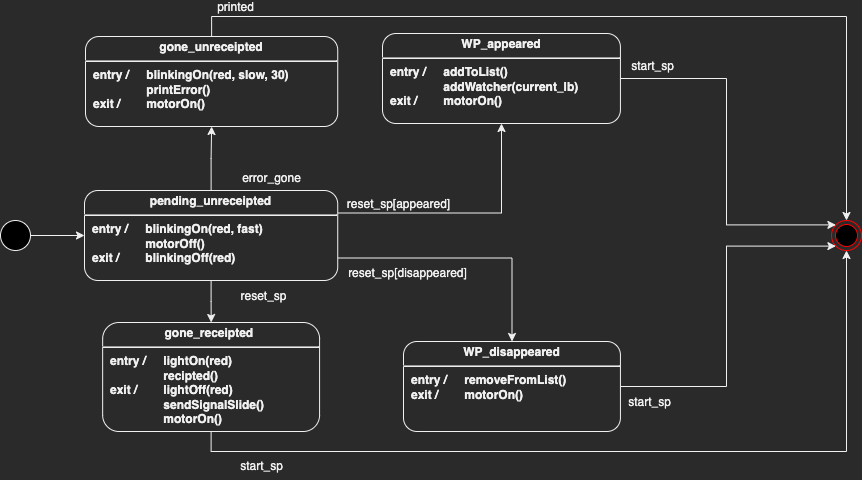
Calibration



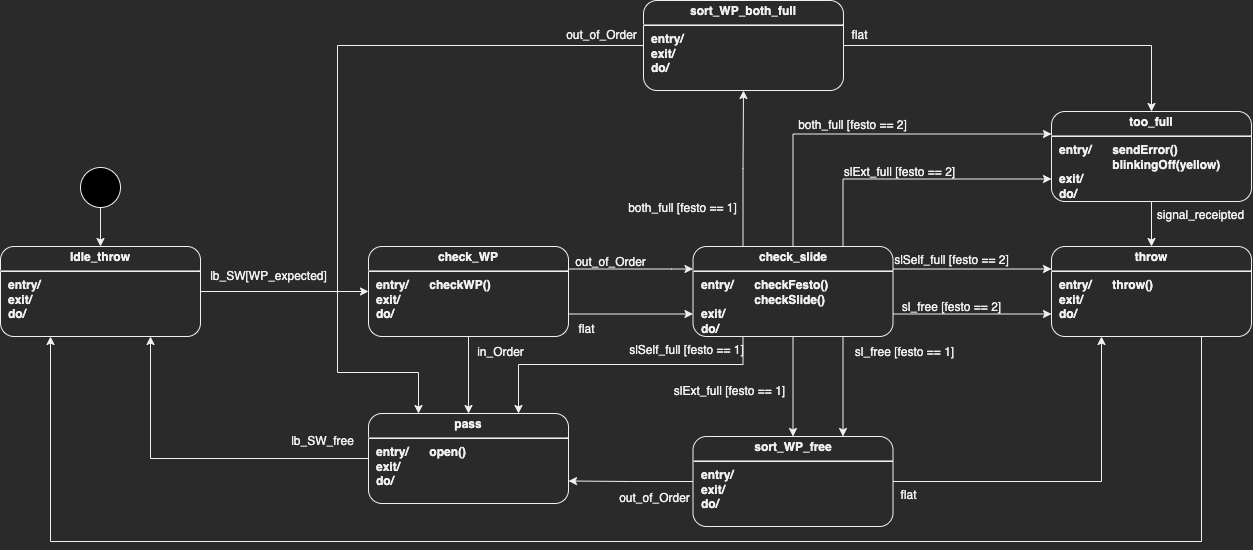
Service-Mode



Error

Operating-Mode

Throw

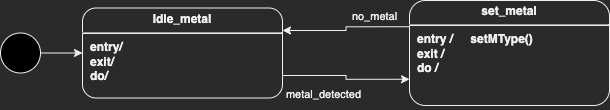


Measure Height

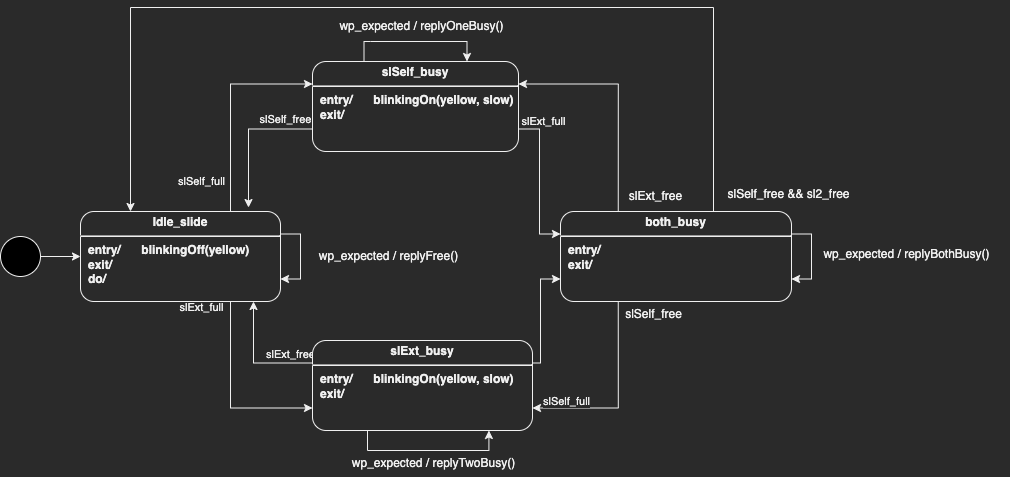
Ein Bild, das Text, Gerät, Screenshot, Messanzeige enthält.

Automatisch generierte Beschreibung

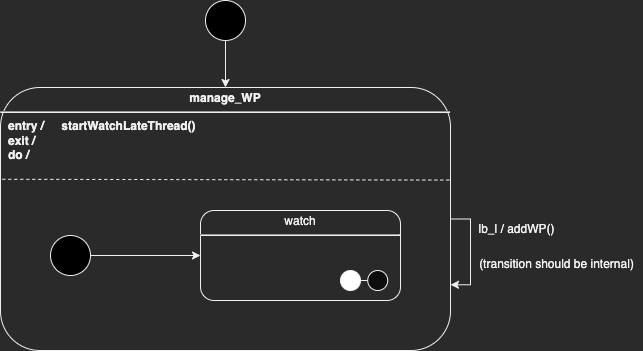
Metal



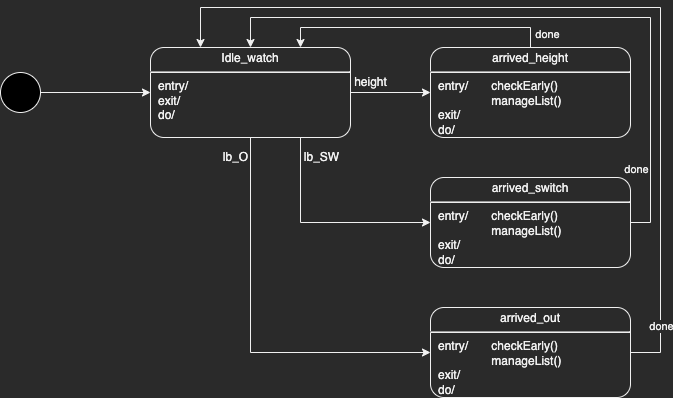
Slide



Manage\_WP



Watch



# Implementierung

Anmerkung: Nur wichtige Implementierungsdetails sollen hier erklärt werden. Code-Beispiele (snippets) können hier aufgelistet werden, um der Erklärung zu dienen. Welche Patterns haben Sie für Ihre Implementierung benutzt.

Anmerkung: Bitte KEINE ganze Programme hierhin kopieren!

# Testen

Machen Sie sich auf Basis Ihrer Überlegungen zur Qualitätssicherung Gedanken darüber, wie Sie die Erfüllung der Anforderungen möglichst automatisiert im Rahmen von Teststufen (Unit-Test, Komponententest, Integrationstest, Systemtest, Regressionstest und Abnahmetest) überprüfen werden.

## Testplan

Definieren Sie Zeitpunkte für die jeweiligen Teststufen in Ihrer Projektplanung. Dazu können Sie die Meilensteine zu Hilfe nehmen. Überlegen Sie, wie die Test-Architektur der jeweiligen Teststufen aussehen. Verwenden Sie Testmethoden wie z.B. Grenzwertanalyse, 100% Zustandsabdeckung, 100% Transitionsüberdeckung, Pfadüberdeckung, Tiefensuche, Breitensuche, etc. Versuchen Sie, so gut wie möglich, Ihre Tests zu automatisieren.

## Testszenarien/Abnahmetest

Leiten Sie die Abnahmebedingungen aus den Kunden-Anforderungen her. Dokumentieren Sie hier, welche Schritte für die einzelnen Abnahmetests erforderlich sind und welches Ergebnis jeweils erwartet wird (Test Cases).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | AT\_01 | **Name** | E-Stop Kommunikation | **Priorität** |  |
| **Voraussetzung:** | * Die Festo sind verbunden und können kommunizieren | | | | |
| **Eingabe:** | * Einer der beiden E-Stops wird betätigt * Der betätigte E-Stop wird wieder herausgezogen * Reset Button an dem Modul dessen E-Stop betätig wurde gedrückt. | | | | |
| **Erwartetes Ergebnis:** | Nachdem der E-Stop gedrückt wurde:   * Belt steht still * Red und Yellow light blinken abwechselnd   Solange bis ... :   * … Beide E-Stops gelöst sind * … Reset button an den Festo an dem der E-Stop gedrückt wurde, betätigt wurden   Wenn die Festos … :   * … die Verbindung zueinander verloren haben, dann versuchen beide Festos eine neue Verbindung zueinander aufzubauen. * … eine bestehende Verbindung haben, dann ist die Anlage wieder bereit für den Betrieb. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | AT\_02 | **Name** | Start und Stop | **Priorität** |  |
| **Voraussetzung:** | * Die Festos sind miteinander verbunden und bilden eine Anlage. | | | | |
| **Eingabe:** | * Einer der Start buttons wird gedrückt * Workpiece wird auf das Belt gelegt und durchbricht die lb\_I des ersten Moduls * Stop Button wird gedrückt | | | | |
| **Erwartetes Ergebnis:** | Nachdem ein Start button gedrückt wurde:   * Green light leuchtet   Wenn ein Workpiece auf das Belt gelegt wird und die lb\_I des ersten Moduls durchbricht:   * Motor startet und das Belt läuft   Solange bis Stop Button gedrückt wird:   * Motor stoppt und Belt steht still | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | AT\_03 | **Name** | Gegangenquittiert | **Priorität** |  |
| **Voraussetzung:** | * Die Festo ist im Betriebszustand * Die lb\_SL signalisieren das die Rutschen voll sind * Ein Workpiece das nicht der Reinfolge entspricht, wird auf das Belt gelegt und durchbricht die lb\_I des ersten Moduls | | | | |
| **Eingabe:** | * Die Rutschen und das Workpiece werden geräumt * Der Reset button wird gedrückt | | | | |
| **Erwartetes Ergebnis:** | Das Workpiece wird als falsch erkannt und muss aussortiert werden.   * Das Green light geht aus * Das Yellow light geht aus * Das Red light blinkt (fast blinking)   Solange bis:   * Die Rutschen und das Workpiece werden geräumt * Der Reset button wird gedrückt   Dann:   * Der Fehler wird quittiert * Das Red light leuchtet   Kurz danach:   * Das Red light geht aus * Das Green light leuchtet | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | AT\_04 | **Name** | Gegangenunquittiert | **Priorität** |  |
| **Voraussetzung:** | * Die Festo ist im Betriebszustand * Die lb\_SL signalisieren, dass die Rutschen voll sind * Ein Workpiece das nicht der Reinfolge entspricht, wird auf das Belt gelegt und durchbricht die lb\_I des ersten Moduls | | | | |
| **Eingabe:** | * Die Rutschen und das Workpiece werden geräumt | | | | |
| **Erwartetes Ergebnis:** | Das Workpiece wird als falsch erkannt und soll aussortiert werden.   * Das Green light geht aus * Das Yellow light geht aus * Das Red light blinkt (fast blinking)   Solange bis:   * Die Rutschen und das Workpiece werden geräumt   Dann:   * Das Red light blinkt (slow blinking)   Kurz danach:   * Das Red light geht aus * Das Green light leuchtet | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | AT\_05 | **Name** | Servicemodus | **Priorität** |  |
| **Voraussetzung:** | * Die Festo ist im Betriebszustand | | | | |
| **Eingabe:** | * Start button wird gedrückt (long press) | | | | |
| **Erwartetes Ergebnis:** | Nachdem der Start button lange gedrückt wurde:   * Green light blinkt (fast blinking) | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | AT\_06 | **Name** | Höhensensor | **Priorität** |  |
| **Voraussetzung:** | * Die Festo ist im Betriebszustand * Ein Workpiece das der nächsten erwarteten Höhe entspricht, wird auf das Belt gelegt und durchbricht die lb\_I des ersten Moduls | | | | |
| **Eingabe:** | * Einer der Start buttons wird gedrückt * Workpiece wird auf das Belt gelegt und durchbricht die lb\_I der Festo1 | | | | |
| **Erwartetes Ergebnis:** | * Das Workpiece passiert den HS | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | AT\_07 | **Name** | Metallsensor | **Priorität** |  |
| **Voraussetzung:** | * Die Festo ist im Betriebszustand * Ein Workpiece das dem nächsten Typ entspricht, wird auf das Belt gelegt und durchbricht die lb\_I von Festo2 | | | | |
| **Eingabe:** | * Workpiece wird auf das Belt gelegt und durchbricht die lb\_Id es Festo2 | | | | |
| **Erwartetes Ergebnis:** | * Das Workpiece passiert den MS | | | | |

## Testprotokolle und Auswertungen

Hier fügen Sie die Test Protokolle bei, auch wenn Fehler bereits beseitigt worden sind, ist es schön zu wissen, welche Fehler einst aufgetaucht sind. Eventuelle Anmerkung zur Fehlerbehandlung kann für weitere Entwicklungen hilfreich sein.

Das letzte Testprotokoll ist das Abnahmeprotokoll, das bei der abschließenden Vorführung erstellt wird. Es enthält eine Auflistung der erfolgreich vorgeführten Funktionen des Systems sowie eine Mängelliste mit Erklärungen der Ursachen der Fehlfunktionen und Vorschlägen zur Abhilfe

# Lessons Learned

Führen Sie ein Teammeeting durch, in dem gesammelt wird, was gut gelaufen war, was schlecht gelaufen war und was man im nächsten Projekt (z.B. im PO) besser machen will. Listen Sie für die Aspekte jeweils mindestens drei Punkte auf. Weitere Erfahrungen und Erkenntnisse können hier ebenso kommentiert werden, auch Anregungen für die Weiterentwicklung des Praktikums.

# Anhang

## Glossar

Tasten & Schalter:

|  |  |
| --- | --- |
| **Name** | **Beschreibung** |
| Start button | Die grüne Taste der Festo, welche mit „Start“ beschriftet ist. |
| Stop button | Die rote Taste der Festo, welche mit „Stop“ beschriftet ist. |
| Reset button | Die graue Taste der Festo, welche mit „Reset“ beschriftet ist. |
| E-Stop switch | Der rote Schalter der Festo, welcher mit „Not-Aus“ beschriftet ist. |

Lampen, LEDs & Leuchten:

|  |  |
| --- | --- |
| **Name** | **Beschreibung** |
| Red light | Die rote Lampe der Ampel. |
| Yellow light | Die gelbe Lampe der Ampel. |
| Green light | Die grüne Lampe der Ampel. |
| Start LED | Die kleine LED, welche sich in der rechten oberen Ecke der Start Button befindet. |
| Reset LED | Die kleine LED, welche sich in der rechten oberen Ecke der Reset Button befindet. |
| Q1 lamp | Das Leuchtfeld, welches sich unter der Reset Button befindet. |
| Q2 lamp | Das Leuchtfeld, welches sich neben der Q1 Lamp befindet. |

Lichtschranken:

|  |  |
| --- | --- |
| **Name** | **Bescheibung** |
| Light-barrier -Inlet | Die Lichtschranke, welche sich links (am Anfang) auf der Festo befindet und das Bit 2 in GPIO 0 für „Werkstück im Einlauf“ setzt. |
| Light-barrier-Hight-Measurement | Die Lichtschranke, welche sich hinter dem HS auf der Festo befindet und das Bit 3 in GPIO 0 für „Werkstück in Höhenmessung“ setzt. |
| Light-barrier -Switch | Die Lichtschranke, welche sich mittig vor der Weiche auf der Festo befindet und das Bit 5 in GPIO 0 für „Werkstück in Weiche“ setzt. |
| Light-barrier -Slide | Die Lichtschranke, welche sich am oberen Ende der Rutsche auf der Festo befindet und das Bit 15 in GIPO 0 für „Rutsche voll“ setzt. |
| Light-barrier -Outlet | Die Lichtschranke, welche sich rechts (am Ende) auf der Festo befindet und das Bit 20 in GPIO 0 für „Werkstück in Auslauf“ setzt. |

Sensoren:

|  |  |
| --- | --- |
| **Name** | **Beschreibung** |
| Height sensor | Ein Sensor, welcher links-mittig über dem Förderband angebracht wurde, um die Höhenwerte der passierenden Werkstücke zu messen. |
| Metal sensor | Ein Sensor, welcher rechts-mittig über dem Förderband angebracht wurde, um das Bit 7 von GPIO 0 auf 1 zu setzen, wenn das passierende Werkstück einen Metallring eingelassen hat. |
| Switch sensor | Ein Metallsensor, welcher erkennt, ob die Weiche offen oder geschlossen ist. Im Falle der Weiche setzt der Weichensensor das Bit 14 in GPIO 0 auf 1, wenn die Weiche mit Strom versorgt wird und die Werkstücke Richtung Outlet durchlässt. Im Falle des Auswerfers setzt der Weichensensor das Bit 14 in GPIO 0 auf 1 wenn der Auswerfer **nicht** mit Strom versorgt wird und die Werkstücke passieren können. |

Aktorik:

|  |  |
| --- | --- |
| **Name** | **Beschreibung** |
| Motor | Der Motor, welcher das Förderband der Festo antreibt, fährt im Rechtslauf. |
| Motor-Slow | Der Motor, welcher das Förderband der Festo antreibt, fährt im langsamen Rechtslauf. |
| Motor-Stop | Der Motor, welcher das Förderband der Festo antreibt, hält an. |
| Switch | Die Weiche **oder** der Auswerfer, welche/r die Werkstücke von dem Förderband auf die Rutsche befördern können. Das Wort Switch steht für beide Komponenten, da unser Code für beides austauschbar funktionieren muss. |

Anderes:

|  |  |
| --- | --- |
| **Name** | **Beschreibung** |
| Festo | Das gesamte Anlagensystem (Beide Förderbandmodule zusammen) |
| conveyor belt | Das Förderband der Festo |
| First conveyor module | Das vordere Förderbandmodul |
| Second conveyor module | Das hintere Förderbandmodul |
| Short press | Drücken mit einer Dauer < 3 Sekunden |
| Long press | Drücken mit einer Dauer >= 3 Sekunden |
| Fast blinking | 1 Hz |
| Slow blinking | 0,5 Hz |
| Service-Mode | Ein Modus zum Testen der Hardware und Teile der Software |
| Operating state | Die Festo ist bereit Werkstücke anzunehmen und zu sortieren |
| Idle state | Die Festo wartet auf Arbeitsaufträge |
| Accepted order | Typ A, Typ B, Typ C |
| lights | Alle ansteuerbaren LEDs, Lamps und Lights |
| components | Alle Sensoren und Aktoren |
| Festo-warnings | Warnungen der Festoanlage (Yellow Light) |

## Abkürzungen

Start button: **Start**  
Stop button: **Stop**  
Reset button: **Reset**  
E-Stop switch: **E-Stop**

Red light: **Red**   
Yellow light: **Yellow**  
Green light: **Green**  
Start LED: **S\_LED**  
Reset LED: **R\_LED**  
Q1 lamp: **Q1**  
Q2 lamp: **Q2**  
**Motor:** **motor**

Light-barrier: **lb**  
-Inlet: **lb\_I**  
-Switch: **lb\_SW**  
-Slide: **lb\_SL**  
-Outlet: **lb\_O**

Height sensor: **HS**  
Metal sensor: **MS**  
Switch sensor: **SwS**

Calibration mode: **CM**Service Mode: **SM**Workpiece: **WP**

First conveyor module: **Festo1**  
Second conveyor module: **Festo2**

**State Machine Begrifflichkeiten**

**Aufbauen einer Verbindung :** **Connecting**

**Verbindung herststellt:** **Connected**

**Verbindung verloren:** **Connection lost**

**Kalibrierung:** **Calibration**

**Ok Zustand:** **OK**

**Fehlerzustand:** **Error**

**Estop:** **E-Stop**

**Ruhezustand:** **Idle**

**Service Modus:** **Service mode**

**Betriebszustand:** **Operating mode**

**Gegangen unquittiert:** **gone unreceipted**

**Anstehend unquittiert:** **pending unreceipted**

**Gegangen quittiert:** **gone receipted**