ESEP Dokument

Entwicklungsentwurfsdokument für eine FESTO Sortieranlage

Johannes Kruber, Luis Nickel, Leo Peters, Malte Schuler

2018

ESP-Bericht zu Conveyor Belts

Inhalt

Inhalt

[1 Dokumentorganisation 8](#_Toc517286008)

[1.1 Autorenliste 8](#_Toc517286009)

[1.2 Versionen 8](#_Toc517286010)

[2 Projekt Struktur 8](#_Toc517286011)

[2.1 Projektvariablen 8](#_Toc517286012)

[2.2 Teamstruktur 9](#_Toc517286013)

[2.3 Entwicklungsstruktur 9](#_Toc517286014)

[2.4 Git Struktur 9](#_Toc517286015)

[3 Requirements und Use Cases 9](#_Toc517286016)

[3.1 Projektplanung 9](#_Toc517286017)

[3.1.1 Stakeholder 9](#_Toc517286018)

[3.1.2 Anforderungen 10](#_Toc517286019)

[3.1.3 Systemkontext 13](#_Toc517286020)

[3.1.4 Use Cases (klein) 13](#_Toc517286021)

[3.1.5 Use Cases (groß) 14](#_Toc517286022)

[3.2 Systemanalyse 16](#_Toc517286088)

[4 Design 16](#_Toc517286089)

[4.1 Systemarchitektur 16](#_Toc517286090)

[4.2 Threadingmodell 16](#_Toc517286091)

[4.3 Threadkommunikation 16](#_Toc517286092)

[4.4 Datenmodellierung 17](#_Toc517286093)

[4.4.1 Interne Messages 17](#_Toc517286094)

[4.4.2 Serial Packages 18](#_Toc517286095)

[4.5 Serielle Kommunikation 18](#_Toc517286096)

[4.6 HAL 19](#_Toc517286097)

[4.7 FSM für Conveyor Belt 1 19](#_Toc517286098)

[4.8 FSM für Conveyor Belt 2 20](#_Toc517286099)

[5 Umsetzung 21](#_Toc517286100)

[5.1 Conveyor Belts 21](#_Toc517286101)

[5.1.1 Laufband 1 21](#_Toc517286102)

[5.1.2 Laufband 2 21](#_Toc517286103)

[5.2 Serielle Schnittstelle 21](#_Toc517286104)

[5.3 Höhen Messung 22](#_Toc517286105)

[6 Testing 22](#_Toc517286106)

[6.1 Einführung 22](#_Toc517286107)

[6.2 Zu testendes System und Testobjekte 22](#_Toc517286108)

[6.3 Vom Test auszuschließende Objekte 22](#_Toc517286109)

[6.4 Organisation der Tests 22](#_Toc517286110)

[6.4.1 Testprozess 23](#_Toc517286111)

[6.5 Risikobetrachtung und Priorisierung 23](#_Toc517286112)

[6.6 Testziele, Testthemen und Teststrategie 23](#_Toc517286113)

[6.7 Qualitätsmaßnahmen 23](#_Toc517286114)

[6.7.1 Feature Driven Development 24](#_Toc517286115)

[6.7.2 Benutzte Agile Methoden zur Verbesserung der Software Qualität 24](#_Toc517286116)

[6.7.3 Umgesetzte Qualitätskriterien nach ISO 25010 24](#_Toc517286117)

[6.7.4 Komponententests 25](#_Toc517286118)

[6.7.5 Integrationstests 25](#_Toc517286119)

[6.7.6 Abnahmetests 26](#_Toc517286120)

[6.7.7 Komponententest Checkliste 26](#_Toc517286121)

[6.7.8 Integration Tests 27](#_Toc517286122)

[6.7.9 Abnahmetests 27](#_Toc517286123)

[6.8 Test Protokoll 35](#_Toc517286124)

[6.9 Test Metrik (Relevante Tests zur Requirementabdeckung) 36](#_Toc517286125)

[7 Bedienungsanleitung 37](#_Toc517286126)

[7.1 Vorwort 37](#_Toc517286127)

[7.2 Merkmale 37](#_Toc517286128)

[7.3 Sicherheitshinweise 37](#_Toc517286129)

[7.3.1 Betriebsanleitung beachten 37](#_Toc517286130)

[7.3.2 Haftung für Funktion und Schäden 37](#_Toc517286131)

[7.4 Anwendungsbereich 38](#_Toc517286132)

[7.5 Inbetriebnahme 38](#_Toc517286133)

[7.6 Bedienung 38](#_Toc517286134)

[7.6.1 Kalibrierung des Laufbandes starten 38](#_Toc517286135)

[7.6.2 Einzelne Sortieranlage zurücksetzen 39](#_Toc517286136)

[7.6.3 E-Schnellstopp benutzen und Sortierung neu starten 39](#_Toc517286137)

[7.6.4 Quittieren 39](#_Toc517286138)

[7.6.5 Sortieranlage nach Inbetriebnahme starten 39](#_Toc517286139)

[7.6.6 Sortieranlage nach einem Reset starten 39](#_Toc517286140)

[7.6.7 Puks dem Laufband zuführen 39](#_Toc517286141)

[7.7 Störungssuche / Fehlerbehebung 39](#_Toc517286142)

[8 Elemente 41](#_Toc517286143)

[8.1 Laufband 41](#_Toc517286144)

[8.2 Puks 41](#_Toc517286145)

[8.2.1 Bohrung mit Metall 41](#_Toc517286146)

[8.2.2 Bohrung ohne Metall 42](#_Toc517286147)

[8.2.3 Binärcodiert 1 42](#_Toc517286148)

[8.2.4 Binärcodiert 2 43](#_Toc517286149)

[8.2.5 Binärcodiert 5 43](#_Toc517286150)

[8.2.6 Flach 44](#_Toc517286151)

[8.2.7 Ohne Bohrung 44](#_Toc517286152)

[9 Lessons Learned 45](#_Toc517286153)

[9.1 Langsam anfangen und dann größer werden. 45](#_Toc517286154)

[9.2 Einzelne Sachen ändern und nicht viele auf einmal 45](#_Toc517286155)

[9.3 Früh Logging zum System hinzufügen 45](#_Toc517286156)

[9.4 Einzelne Teile testen bevor man es zum Projekt hinzufügt 45](#_Toc517286157)

[9.5 Alles braucht länger als man denkt 45](#_Toc517286158)

[9.6 Mit einem Kommilitonen zu reden hilft mehr als man denkt 45](#_Toc517286159)

[9.7 Von Angesicht zu Angesicht zu reden ist das Produktivste 45](#_Toc517286160)

[9.8 Über ein Problem schlafen 45](#_Toc517286161)

[9.9 „Wenn du willst, dass es funktioniert mach es selbst“ 46](#_Toc517286162)

[9.10 QNX Compiler ist sehr labil 46](#_Toc517286163)

[10 Mängelliste 46](#_Toc517286164)

# Dokumentorganisation

## Autorenliste

|  |  |
| --- | --- |
| Kürzel | Name |
| LMN | Prof. Dr. Thomas Lehmann |
| JK | Johannes Kruber |
| LN | Luis Nickel |
| LP | Leo Peters |
| MS | Malte Schuler |

## Versionen

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Erstellt | Autor | Kommentar |
| 0.1 | 2018-03-12 | LMN | Initiale Version der Templates. |
| 0.2 | 2018-04-12 | MS | Struktur des Templates hinzufügen |
| 0.3 | 2018-04-12 | MS | Use Cases und Stakeholder |
| 0.4 | 2018-04-26 | LP | FSM und weitere Use Cases |
| 0.41 | 2018-05-03 | MS | Anforderungsanalyse und Systemarchitektur |
| 0.42 | 2018-05-03 | MS | Datenmodelle |
| 0.5 | 2018-05-15 | LP | Serielle Schnittstelle |
| 0.6 | 2018-05-16 | MS | Architektur Kapitel Aufbau verändern |
| 0.7 | 2018-05-23 | MS | Abnahmetests hinzufügen |
| 0.71 | 2018-05-24 | MS | Weitere Abnahmetests hinzufügen |
| 0.8 | 2018-06-11 | MS | Qualitätsmaßnahmen hinzugefügt |
| 0.81 | 2018-06-13 | MS | Umsetzung hinzugefügt |
| 0.9 | 2018-06-14 | MS | Bedienungsanleitung hinzugefügt |
| 0.91 | 2018-06-14 | MS | Elemente eingefügt |
| 0.92 | 2018-06-14 | MS | HöhenMessungsAutomat hinzugefügt |
| 0.93 | 2018-06-18 | MS | Mängelliste eingefügt / Bedienungsanleitung angepasst |
| 0.94 | 2018-06-20 | MS | Test Metrik / Fehler behoben |
| 1.0 | 2018-06-21 | MS | Anpassungen für Abnahme |

# Projekt Struktur

## Projektvariablen

|  |  |
| --- | --- |
| Variable | Variablenname |
| sensors: | sn\_ |
| lightbarrier start: | sn\_lb\_start |
| lightbarrier height sensor: | sn\_lb\_heightsn |
| lightbarrier switch: | sn\_lb\_switch |
| lightbarrier ramp: | sn\_lb\_ramp |
| lightbarrier end: | sn\_lb\_end |
| height sensor: | sn\_height |
| height control: | sn\_height\_control |
| switch (Weiche) open: | sn\_switch\_open |
| metal sensor: | sn\_metal |
| buttons: | btn\_ |
| button start: | btn\_start |
| button stop: | btn\_stop |
| button reset: | btn\_reset |
| button E-Stop: | btn\_emergency |
| actors: | ac\_ |
| motor: | ac\_motor |
| switch (Weiche): | ac\_switch |
| activity lamp: | ac\_lamp |
| LEDs: | led\_ |
| LED start button: | led\_start |
| LED reset button: | led\_reset |
| LED Q1: | led\_q1 |
| LED Q2: | led\_q2 |
| Leo | Protokollant und Digitalisierung von Absprachen / Debugging |
| Johannes | Programmierung und Design/Prototyping |
| Luis | Programmierung und Prototyping/Design |
| Malte | Projektorganisation / Doku / Design / Tests |

## Teamstruktur

|  |  |
| --- | --- |
| Version 0.3 (Stand 12.06.2018) | Aufgaben |
| Leo | Protokollant und Digitalisierung von Absprachen |
| Johannes | Programmierung und Design/Prototyping |
| Luis | Programmierung und Prototyping/Design |
| Malte | Projektorganisation / Doku / Design |

## Entwicklungsstruktur

Absprache 🡪 Design 🡪 Prototyping 🡪 Komponentenschnittstellen umsetzen 🡪 Komponente umsetzen 🡪 Mit anderen Komponenten integrieren 🡪Kundenabnahme

## Git Struktur

Nach Gitflow:

-Masterbranch für Releases.

-Develop branch für Zwischenstadien

-Featurebranches für Entwicklung einzelner Features.

# Requirements und Use Cases

## Projektplanung

### Stakeholder

|  |  |
| --- | --- |
| **Stakeholder** | **Beschreibung** |
| **Anwender** | Diejenige Person, die das Laufband bedient |
| **Kunde** | Die Person, welche die Maschine bezahlt |
| **Projektleiter** | Die Person die das Projektmanagement übernimmt |
| **Produktmanager** | Die Person, welche das Produktverhalten bestimmt |
| **Programmierer** | Die Person, die das bestimmte Produktverhalten umsetzen |

### Anforderungen

Die Zahlen am Ende der Sätze verweisen auf die Nummerierung in der Aufgabenstellung. **Fett gedruckte Stellen** weisen auf eine Änderung/genauere Beschreibung der Aufgabenstellung hin.

Aus zwei Förderbandmodulen soll eine Werkstück-Sortieranlage gebaut werden. Jedes Förderband (im weiteren FB 1 bzw. FB 2) besitzt sein eigenes Steuerungssystem, beide Systeme sind über eine serielle Schnittstelle gekoppelt und können darüber kommunizieren.

Die Sortieranlage soll bestimmte Werkstücke über die vorhandenen Rampen aussortieren. Am Ende von FB 2, sollen Werkstücke in vorgegebener Reihenfolge ankommen(1).  
Es gibt vier Typen von Werkstücken:

* Flache Werkstücke(3)
* Werkstücke mit Bohrung und Metalleinsatz(4) (Bohrung oben mit Metall, im weiteren BOM)
* Werkstücke mit Bohrung ohne Metalleinsatz(5) (Bohrung oben ohne Metall, im weiteren BOOM)
* Binärcodierte Werkstücke(6) (im weiteren mit Typkennung, wobei die Typnummer der binären Codierung entspricht(7))

Für binärcodierte Werkstücke gilt, ein mitteltiefer Ring im Werkstück stellt eine logische 0 dar, ein tiefer Ring eine logische 1(8). **Die binäre Codierung wird von der Mitte des Werkstücks nach außen gemessen.(1)**

Bei der Sortierung müssen folgende Regeln beachtet werden(10):

* Reihenfolge am Ende des FB 2: BOM -> BOOM -> BOOM(9)
* Flache Werkstücke sollen auf FB 1 erkannt und aussortiert werden(11)
* Werkstücke mit der Bohrung unten sollen auf beiden FB erkannt **und frühestmöglich(2)** aussortiert werden(13)
* Werkstücke vom Typ 1 & 4 sollen auf dem FB 1 aussortiert werden(14)
* Werkstücke vom Typ 2 & 7 sollen auf dem FB 2 aussortiert werden(15)

Die Werkstücke werden am Anfang von FB 1 auf die Sortieranlage **zugeführt(3)**(16), wodurch die Lichtschranke („Start“) unterbrochen wird(17). Es dürfen dem FB 1 stets **einzeln** Werkstücke **zugeführt(4)** werden, **solange sich keine anderen Werkstücke in der Lichtschranke befinden(5)**(18) und ein Abstand von 2 Pukdurchmesser eingehalten wird(Absprache)**.** Es können sich auf FB 1 also mehrere Werkstücke gleichzeitig befinden**(35)**(19).   
Die Übergabe an das FB 2 soll einzeln erfolgen. Es wird nur ein Werkstück übergeben, wenn auf FB 2 **kein Werkstück vorhanden ist (Rampe ausgenommen)(6)** (20)**.** Manchmal kann es vorkommen, dass Werkstücke sich bei der Übergabe von FB 1 nach FB 2 „überschlagen“. Eine auf FB 1 oben liegende Bohrung liegt dann eventuell unten(21). **Unabhängig der vorherigen Messungen auf FB 1, wird dieses Werkstück auf FB 2 aussortiert werden.(7)**

Auf beiden Bändern sollen Werkstücke **langsam(8)** durch die Höhenmessung transportiert werden(22). Es dürfen vorne an FB 1 und hinten an FB 2 **keine Werkstücke von den Bändern fallen(9)**(24).

Ist ein Werkstück am Ende von FB 2 angekommen, sollen auf der Konsole folgende Werkstückdaten **ausgegeben werden(10)**(25):

* Werkstück-ID(26)
* Typ(27)
* Höhenmesswert von FB 1(28)
* Höhenmesswert von FB 2(29)

Die Werkstück-ID wird beim auflegen des Werkstücks auf FB 1 vergeben(30). Im Normalfall ist der Typ des Werkstücks schon nach Durchlauf von FB 1 bekannt(31).

Sobald ein binär-codiertes Werkstück auf einem der beiden Bänder **erkannt wird(11)**, wird auf der zugehörigen Konsole folgendes ausgegeben(32):

* Zeitstempel der Erkennung(33)
* Werkstück-ID(34)
* Binärcode(35)
* Höhenmesswert(36)

Wenn sich auf einem FB keine Werkstücke befinden, soll dieses **anhalten(12)**(37). Wenn die Rampe von FB 1 voll ist, wird das auszusortierende **Werkstück stattdessen(13)** auf der Rampe des FB 2 aussortiert(38). **Andersherum(14)** soll ein Werkstück bereits auf der Rampe des FB 1 aussortiert werden, wenn die Rampe des FB 2 voll ist(39). Wenn ein Werkstück **auf dem jeweils anderen Band(15)** aussortiert wird, *soll dies dem Bediener signalisiert werden(40) (Auf welche Art wird das Signal gegeben?).* Die Weiche darf nicht **minutenlang(16)** auf Durchgang gestellt sein(45).

Die folgenden Fehlerzustände beim Betrieb der Anlage sollen erfasst werden:

* Verschwinden von Werkstücken**(17)**(48).
* Hinzufügen von Werkstücken in der Mitte eines FB**(18)**(49).
* Beide Rampen sind voll**(19)**(50).

Auf die oben genannten Fehler wird reagiert, indem **beide FB gestoppt** werdenund eine Fehlermeldung ausgegeben wird. (*Wie wird nach dem stoppen der Anlage weiter vorgegangen?)* (51)  
Im Folgenden sind die Funktionen der Tasten der Anlage aufgelistet(52):

* Start: Die Anlage wechselt in den Betriebszustand**(20)**(53). Wird die Taste länger gedrückt gehalten, wechselt die Anlage in den Service-Modus**(21)**. Im Service-Modus führt die Anlage die vorgefertigten Selbsttests durch(54).
* Stop: Die Anlage wechselt in den Ruhezustand**(22)**(55). Dies ist nur möglich, wenn keine Fehler oder Warnungen vorliegen**(23)**(56).
* Reset: Fehlerquittierung**(24)**(57) s.u.
* E-Stop: Schnellabschaltung. Die gesamte Anlage wird stillgesetzt**(25)**(59). Wenn der Schalter herausgezogen wird, bleibt die Anlage im Stillstand bis die Reset-Taste gedrückt wird**(26)**(60).

*Die LEDs an den Tasten sollen dem Bediener Hinweise zur Bedienung geben. (*Welche LED soll wann an/aus sein?) (61).

Die Ampellichter der Anlage sollen folgende Zustände signalisieren(62):

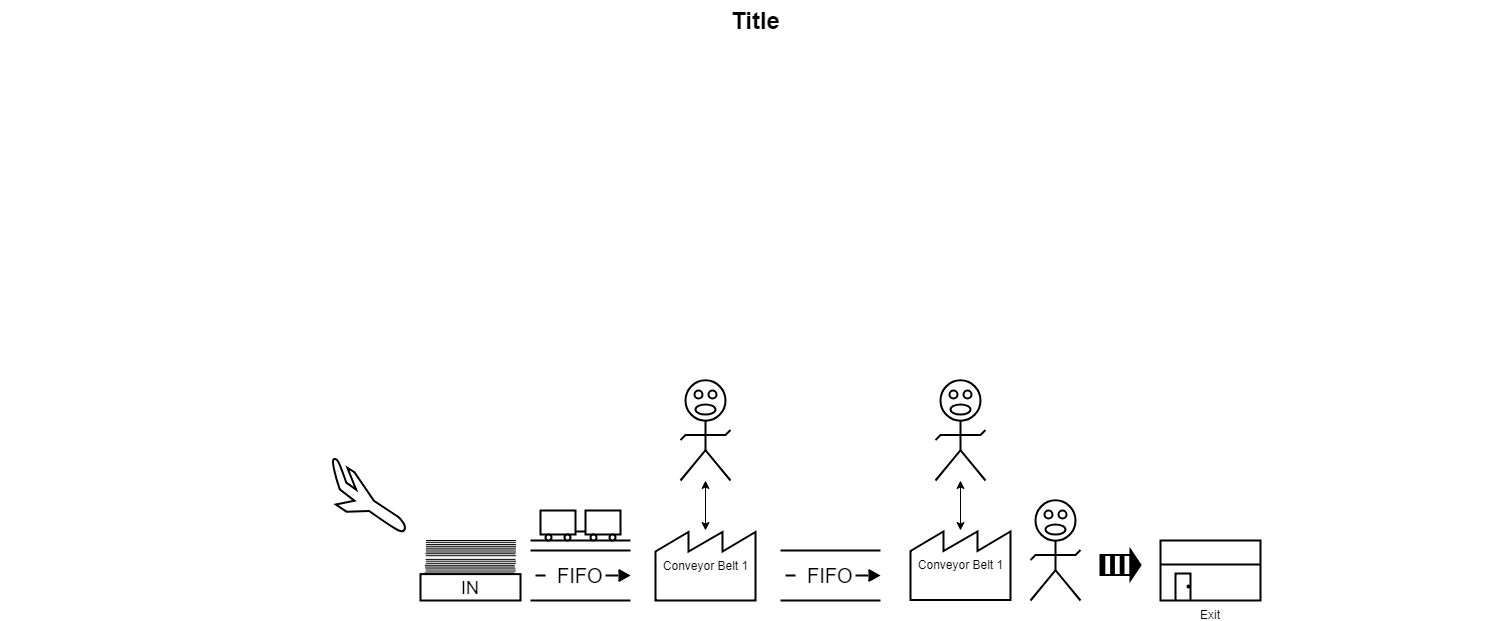
* Grün: Im Betriebszustand leuchtet die Lampe dauerhaft**(27)**(63). Im Service-Modus blinkt die  
  Lampe gleichmäßig**(28)**(64).
* Gelb: Bei Warnungen blinkt die Lampe gleichmäßig**(29)**(65).
* Rot: Fehler- und Quittierzustände(66):  
  - anstehend unquittiert(68), wenn ein Fehler gerade erst aufgetreten ist. Sobald der Bediener den Fehler bemerkt, muss er die Quittierungstaste drücken**(30)**(69). Nun wechselt der Fehler in den Zustand:  
  - anstehend quittiert(70). Wenn der Fehler-Signaleingang den Wert OK hat, wechselt der Fehler in den Zustand  
  - OK(71).  
  - gegangen unquittiert**(37)**(74) ist der Fehlerzustand, wenn der Fehler behoben wurde, ohne dass der Bediener ihn zur Kenntnis genommen hat(73).

Es gibt einen allgemeinen Signaleingang, der einen Fehler signalisiert(75). Die Anzeige für aufgetauchte oder verschwundene Werkstücke geht nach dem Drücken der Quittierungstaste direkt in den Zustand OK**(31)**(76).

Folgende Signale sollen für die verschiedenen Fehlerzustände ausgegeben werden(77).

Kein Fehler: Lampe aus**(32)**(78)  
anstehend unquittiert: schnelles Blinken (1 Hz)**(33)** (79)  
anstehend quittiert: Lampe leuchtet dauerhaft**(34)**(80)  
gegangen unquittiert: langsames Blinken (0,5 Hz)**(35)** (81)

### Systemkontext



### Use Cases (klein)

Epic:

Als Anwender, möchte ich Bausteine, am Anfang, auf das Förderband 1 legen, um am Ende des zweiten Bandes die richtigen Bausteine in der richtigen Reihenfolge wieder herunternehmen zu können.

Story:

- Als Anwender möchte ich (an der Kontrollleuchte) erkennen können, ob das Förderband sich gerade bewegt, um effizient arbeiten zu können.

- Als Anwender möchte ich (an der Kontrollleuchte) erkennen können, ob sich das Förderband in einem Fehlerzustand befindet, um diesen schnell erkennen und beheben zu können.

- Als Anwender möchte ich (an der Kontrollleuchte) erkennen können, ob sich das Förderband gerade in einem "Selbsttest" befindet.

- Als Anwender möchte ich einen Baustein am Anfang des Förderbandes auflegen können, damit dieser sortiert wird.

- Als Anwender möchte ich auch Bausteine, die sich auf der Rutsche befinden vorne auf das Förderband auflegen können, um diesen erneut zu sortieren.

- Als Anwender möchte ich beide Bänder bei Bedarf sofort anhalten können, um für meine Sicherheit zu garantieren.

- Als Anwender möchte ich ein Band bei Bedarf anhalten können, um bei Fehlern schneller agieren zu können.

- Als Anwender möchte ich bei einem Fehlerfall die Bänder wieder in den Anfangszustand versetzen können, um diesen eigenständig zu beheben.

- Als Anwender möchte ich erkennen können, in welche Richtung sich das Band bewegt, um effizient arbeiten zu können.

- Als Anwender möchte ich zu jedem Zeitpunkt erkennen, wie schnell sich das Band aktuell bewegt, um für meine Sicherheit zu garantieren.

- Als Anwender möchte ich einen flachen Baustein vorne auf das erste Förderband legen und dieser soll möglichst auf die erste Rampe transportiert werden, um den falschen Baustein auszusortieren.

- Als Anwender möchte ich, dass Bausteine, die auf dem Förderband umkippen o.ä. auf die nächstmögliche Rampe transportiert werden, um diesen erneut auf das Förderband aufzulegen.

- Als Anwender möchte ich am Ende des zweiten Förderbandes nur Bausteine abnehmen müssen, die der gewünschten Reihenfolge entsprechen, um weiter mit diesen zu arbeiten.

### Use Cases (groß)

| **Abschnitt** | **Inhalt** |
| --- | --- |
| Bezeichner | Emergency Stop |
| Titel | Emergency Stop |
| Kurzbeschreibung | Wenn der Not-Aus Schalter betätigt wird, müssen beide Förderbänder sofort anhalten. |
| Autoren | Leo |
| Priorität (Projekt) | Sehr hoch |
| Priorität (Technik) | Sehr hoch |
| Verantwortlich |  |
| Akteure | Anwender |
| Auslösendes Ereignis | Betätigen des Not-Aus Schalters |
| Vorbedingung | Betrieb des Förderbandes |
| Nachbedingung | Zurücksetzten des Förderbandes |
| Ergebnis | Die beiden Förderbänder halten sofort an. Die Ampel springt auf rot. Alle weiteren Aktoren bleiben in ihrem aktuellen Zustand. |
| Hauptszenario | Der Anwender betreibt das Förderband. Aus beliebigem Grund (zum Beispiel ein Notfall im Betrieb) betätigt der Anwender den Not-Aus Schalter. Beide Laufbänder halten sofort an, die Ampel springt auf rot. Alle weiteren Aktoren bleiben in ihrem aktuellen Zustand. Der Anwender kann jetzt das Problem beheben. Anschließend müssen beide Förderbänder wieder in den Ausgangszustand gebracht und zurückgesetzt werden. |
| Alternativszenario 1 |  |
| Alternativszenario 2 |  |
| Alternativszenario 3 |  |
| Ausnahmeszenarien |  |
| Qualitäten |  |

| **Abschnitt** | **Inhalt** |
| --- | --- |
| Bezeichner | RE FB2 |
| Titel | Reihenfolge einhalten, Förderband 2 |
| Kurzbeschreibung | Am Ende des zweiten Förderbandes sollen nur Werkstücke in der gewünschten Reihenfolge ankommen. Werkstücke, die nicht der Reihenfolge entsprechen werden vorher auf den Rampen aussortiert. |
| Autoren | Leo |
| Priorität (Projekt) | Sehr hoch |
| Priorität (Technik) | Sehr hoch |
| Verantwortlich | Programmierer |
| Akteure | Anwender |
| Auslösendes Ereignis | Eintreffen eines Werkstückes auf das zweite Förderband |
| Vorbedingung | Inbetriebnahme des Förderbandsystems |
| Nachbedingung | - |
| Ergebnis | Am Ende des zweiten Förderbandes werden Werkstücke in der gewünschten Reihenfolge entnommen. Werkstücke, welche nicht des gewünschten Reihenfolge entsprechen, werden über die Rampen aussortiert. |
| Hauptszenario | Ein Puk, welcher der gewünschten Reihenfolge entspricht,trifft während des Betriebs des Förderbandsystems am zweiten Förderband ein. Dieser läuft über das Förderband und wird nicht auf die Rampe aussortiert. |
| Alternativszenario 1 | Ein Puk, welcher nicht der gewünschten Reihenfolge entspricht, trifft während des Betriebs des Förderbandsystems am zweiten Förderband ein. Dieser läuft über das Förderband und wird auf die Rampe aussortiert. |
| Alternativszenario 2 | Ein Puk, welcher nicht der erwarteten Typkonfiguration entspricht, trifft während des Betriebs des Förderbandsystems am zweiten Förderband ein. Dieser läuft über das Förderband und wird auf die Rampe aussortiert. |
| Ausnahmeszenarien | Der Puk wird während des Vorgangs vom Laufband entfernt. |

## Systemanalyse

# Design

## Systemarchitektur

Die Systemarchitektur beschreibt den Aufbau unserer Komponenten. So gehen wir davon aus dieses über folgendes Modell darzustellen.

(Visual Paradigm File)

Die Logik soll dabei über Automaten abarbeitet werden.

Für Förderband 2 kann die Logik zur Sortierung der Puks innerhalb eines NEA umgesetzt werden, da nur ein Puk zur gleichen Zeit auf dem Förderband vorhanden sein darf.

Für Förderband 1 haben wir vorgesehen eine State Machine für jeden einzelnen Puk umzusetzen. Zwischen zwei Sensoren werden die State Machines der jeweiligen Puks durch eine Queue nach einem FIFO Prinzip abgearbeitet.

## Threadingmodell

In einer State Maschine sind die erwarteten State Modelle der Threads angesprochen.



## Threadkommunikation

Folgendes Dokument hilft die Kommunikation zwischen den einzelnen Threads darzustellen.



## Datenmodellierung

### Interne Messages

#### Modell:

#define ACTOR\_ARRAY\_SIZE 12  
#define SENSOR\_ARRAY\_SIZE 12  
#define BLINK\_ARRAY\_SIZE 3

int senderId;  
 int receiverId;  
 bool actorStatus[ACTOR\_ARRAY\_SIZE];  
 bool blink[BLINK\_ARRAY\_SIZE];  
 bool sensorData[SENSOR\_ARRAY\_SIZE];  
 bool adcEnable;  
int hightSensor;  
int pulseType;

#### Defines:

#define MSG\_PULS\_ALTIMER 0  
#define MSG\_PULS\_TIKTIMER 1  
#define MSG\_PULS\_SERIAL 2  
#define MSG\_PULS\_GPIOISR 3  
#define MSG\_PULS\_ADCISR 4  
#define MSG\_PULS\_NOPULS -1  
  
#define MSG\_SN\_LB\_START 0  
#define MSG\_SN\_LB\_HIGHTSN 1  
#define MSG\_SN\_HIGHTCONTROLE 2  
#define MSG\_SN\_LB\_SWITCH 3  
#define MSG\_SN\_METALL 4  
#define MSG\_SN\_SWITCHOPEN 5  
#define MSG\_SN\_LB\_RAMP 6  
#define MSG\_SN\_LB\_END 7  
#define MSG\_BTN\_START 8  
#define MSG\_BTN\_STOP 9  
#define MSG\_BTN\_RESET 10  
#define MSG\_BTN\_EMERGENCY 11

// GPIO1  
#define MSG\_MOTOR\_RIGHT 0  
#define MSG\_MOTOR\_LEFT 1  
#define MSG\_MOTOR\_SLOW 2  
#define MSG\_MOTOR\_STOP 3  
#define MSG\_ACTLAMP\_RED 4  
#define MSG\_ACTLAMP\_YELLOW 5  
#define MSG\_ACTLAMP\_GREEN 6  
#define MSG\_SWITCH 7

// GPIO2  
#define MSG\_LED\_START 8 //LED Taste Start  
#define MSG\_LED\_RESET 9 //LED Taste Reset  
#define MSG\_LED\_Q1 10 //Signalleuchte Q1  
#define MSG\_LED\_Q2 11 //Signalleuchte Q2

### Serial Packages

#### Modell:

##### Header:

type, // Type of sent Message (See TypeDefinitions)  
version, // Count for rejected messages in a row  
msgNumber, // Number of sent Message  
ackNumber, // Ackknowledgement number  
checkSum; // Sum for checking for complete message

##### Body:

Message // Content

#### Definitionen:

sizeOf(type) = char;

type0 = ping; // Check for working signal  
type1 = pukMessage; // Data for Puk that will be transferred from CB1 to CB2  
type2 = emergencyStop; // Signal for Emergency Stop  
type3 = emergencyStopClear // Signal for Emergency Stop cleared  
type4 = rampFull; // Signal that Ramp is full  
type5 = rampFree; // Signal that Ramp is free  
type6 = beltFree; // Answer to BeltStatusRequest: Belt is free  
type7 = beltInUse; // Answer to BeltStatusRequest: Belt is in use  
type8 = beltStatusRequest; // Checking status of CB  
type9 = sendWarning; // Send Warning signal  
type9 = MetalAccepted // Metal Puk Accepted  
type10= MetalRejected // Metal Puk Rejected

Checksumme == 15  
sizeOf(version) = char  
sizeOf(msgNumber) = int;

sizeOf(ackNumber) = int;  
sizeOf(timeStamp) = long;

## Serielle Kommunikation

Die serielle Kommunikation soll über 2 Devices, eines zum Lesen und eines zum Schreiben realisiert werden. Ein Controller, welcher beide Devices steuern kann verarbeitet dann die ankommenden und abgehenden Nachrichten. Ankommende Nachrichten werden dann verarbeitet und an den MainController weitergeleitet. Ausgehende Nachrichten vom MainController werden in ein Übertragungspaket umgewandelt und dann über die serielle Schnittstelle geschrieben.

Zusätzlich werden in regelmäßigen Zeitabständen Pings gesendet, sodass ein vorhanden sein der Verbindung garantiert wird. Jedes Paket wird von jeweiligen Partner quittiert.

Eine Paketformatvorlage mit entsprechenden Headern sowie Bitgrößen unter 4.4.2

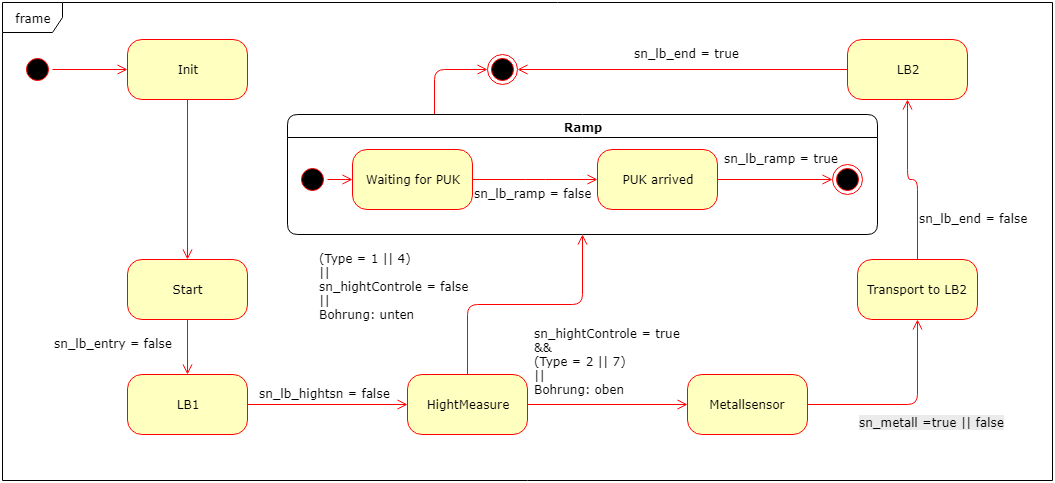
## HAL

Die Bibliotheken werden von ein Hal klasse gekapselt. Auf diese Hal klasse wird mit entsprechenden Sensor- und Aktorklassen zugegriffen.

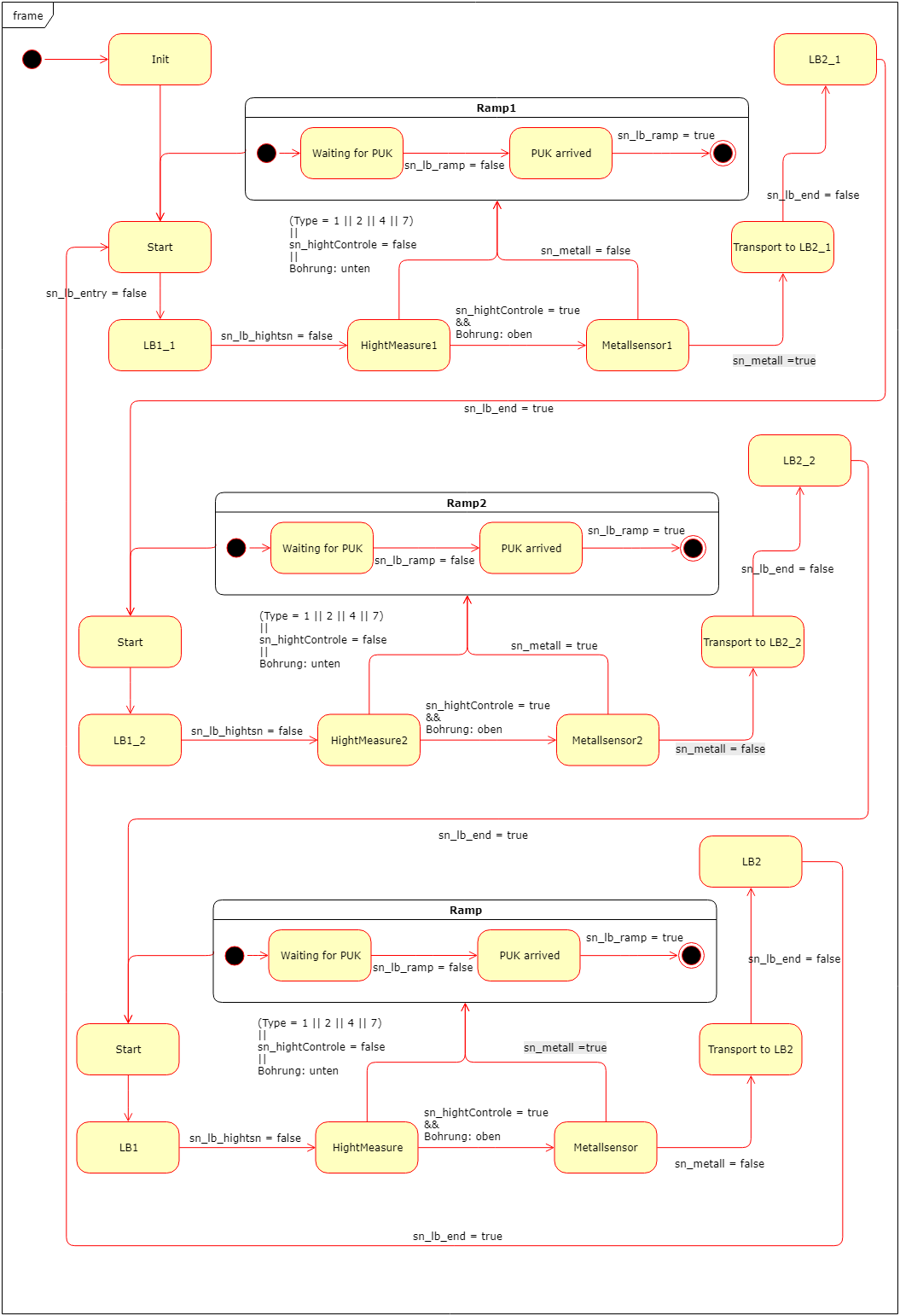
Eine Controllerklasse verwaltet die HAL und alle zugehörigen Sensor- und Aktorobjekte. Diese gewährt nach außen den Zugriff auf die Hardware.

Die sensorwerte werden innerhalb des HAL-Objektes gespeichert und können auf Wunsch aktualisiert werden. Einzelne Sensorwerte können dann nach außen gegeben werden.

## FSM für Conveyor Belt 1



## FSM für Conveyor Belt 2



D

# Umsetzung

## Conveyor Belts

Die Grundlage für beide Laufbänder ist eine StateMachine Klasse, welche Funktionalitäten für beide Laufbänder bietet. Das wären Initialisierungen sowie eine FSM die einen globalen Zustand des Laufbandes herstellt und Übergangsfunktionen für alle States und Pukstates.

Die einzelnen StateMachines der Laufbänder werden jeweils vom StateMachine Client gestartet, die Client kann auch einen Kalibrierungslauf starten um die Zeitabstände zwischen einzelnen States und der Gesamtlaufzeit auszumessen. Diese Daten werden dann gespeichert und stehen während der normalen Ausführung für Vergleiche bereit.

Weiterhin existieren im Gegensatz zu den FSMs noch weitere interne Timer zur Fehlerüberwachung, um den Fortschritt einzelner Puks auf dem Laufband zu protokollieren und die Höhenmessung entsprechend auswerten zu können. Diese Timer können auch einen Statusübergang verursachen. (Errorstate)

### Laufband 1

Das Laufband 1 hat keine eigene FSM, sondern bezieht sich ihren Status aus den aufgelegten Puks welche alle eine eigene FSM haben. Diese FSMs werden vom Laufband aktualisiert, wenn ein Interrupt für die State Machine ausgelöst wird. Interrupts werden vor allem durch Gerät interne Nachrichten ausgelöst. Zusätzlich zu den einzelnen FSMs aus 4.7 existieren nun 4 selbstdefinierte Abschnitte auf dem Laufband, welche durch einzelne FIFO Segmente dargestellt werden. Diese existieren jeweils zwischen zwei Lichtschranken. Diese FIFOs existieren um die PUK Reihenfolge korrekt darzustellen.

### Laufband 2

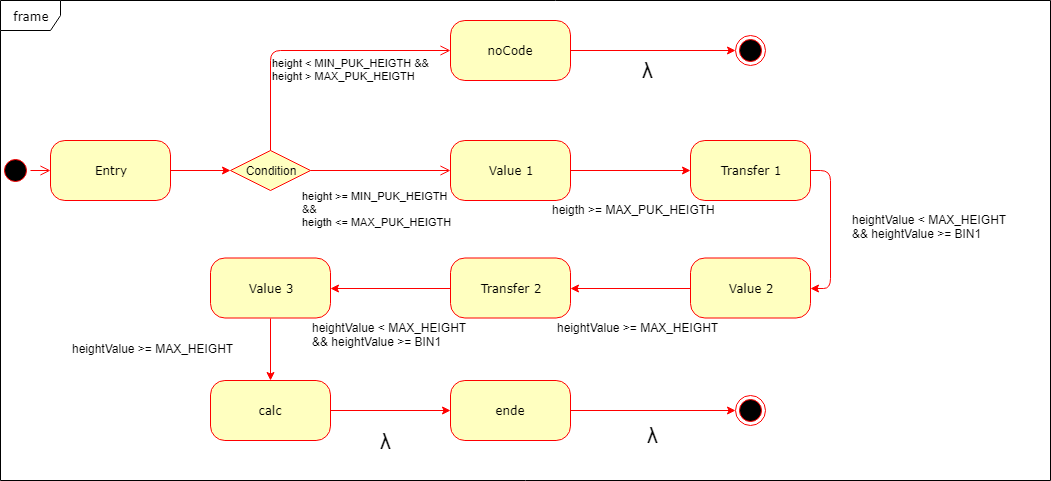
Das Laufband 2 basiert auf dem FSM aus 4.8 und besitzt die States des FSMs und separat noch einmal alle Übergangsfunktionen des FSM zusätzlich zu Lamda Übergängen bei Fehlern.

## Serielle Schnittstelle

Die serielle Schnittstelle ist über einen separaten Client (also Thread) umgesetzt und verbindet sich nach Möglichkeit mit einer anderen Maschine automatisch.

Die serielle Schnittstelle besitzt somit vom System eine sehr lose Kopplung. Durch ein gemeinsames Speichermedium weist die serielle Schnittstelle zur State Machine jedoch eine starke Kupplung auf. Unter 4.4.2 das Kommunikationsmodell aufgeführt, welches verschickt wird.

## Höhen Messung



Die Höhen Messung wird durch einen eigenen Automaten umgesetzt welcher sich an mehreren Referenz Punkten Höhen Messwerte nimmt. Zunächst wird festgestellt ob eine Bohrung vorhanden ist, falls nicht wird der Puk aussortiert.

Als nächstes wird festgestellt ob ein Binärcode existiert, falls ja werden die Werte zwischengespeichert und nachher kombiniert um den Binärcode auszugeben/zu berechnen.

# Testing

## Einführung

Es soll eine Werkstück Sortieranlage programmiert werden, diese soll bzgl. Requirements im RDD Dokument abgeschlossen sein. Dies behandelt z.B. das Einführen von verschiedenen Elementen in den Ablauf inmitten der Anlage.

Fehlerhafte Abläufe sollen erkannt und per Anzeige bekannt gemacht werden um dann vom Nutzer quittiert zu werden.

## Zu testendes System und Testobjekte

Es ist die Werkstück Sortieranlage auf die korrekte Bedienung der zu bedienenden Elemente der Puks zu testen.

Zusätzlich wird die Werkstück Sortieranlage auf die Interaktion mit dem Nutzer getestet.

## Vom Test auszuschließende Objekte

Äußere Einflüsse die die Anlage beschädigen.

Im Betrieb ausfallende Hardwareelemente die nicht digital erkannt werden können.

Hardware Objekte die Teilfunktional sind und die durch Tests nicht unbedingt ersichtlich sind. (z.B. Wackelkontakte an den Kabeln/unebene Puks)

## Organisation der Tests

Die Organisation und Implementierung der Tests findet durch den Qualitätsmanager statt.

### Testprozess

Der Testprozess baut auf verschiedenen Testgebieten auf. Dazu gehören

* Umsetzung von Use Cases
* Testen einzelner Komponenten wie Aktoren und Sensoren
* Testen der Interaktion mit dem Nutzer
* Testen einzelner Subsysteme

Diese werden auf verschiedene Art und Weise getestet. So werden Komponenten z.B. zunächst mit White- und Grey-Box-Testing auf ihre Funktionen getestet, um danach mit Black-Box-Testing im Zusammenspiel mit anderen Subsystemen in Integrationstests getestet zu werden.

Bedienbarkeit – in dem Falle Interaktionen – kann z.B. nicht allein durch Komponenten-Tests und Integrationstests getestet werden, sondern wird durch den Eindruck von Qualitätsmanager und Kunden beeinflusst.

Letztendlich wird das System für den Kunden entwickelt und von ihm im sogenannten Abnahmetest abgenommen.

## Risikobetrachtung und Priorisierung

Die Risikobetrachtung bezieht sich hauptsächlich auf den Verzug durch nicht erwünschtes Verhalten von Code die beim Testen fertig gestellt wird. Um dies frühzeitig zu erkennen und pünktlich für das Workpackage fertig zu stellen ist es nötig grundlegende Funktionalitäten des Systems Elemente frühzeitig zu testen.

Sollten ggf. Elemente falsch Priorisiert werden, so kann eine Verzögerung des gesamten Projektes eintreten.

Im Falle des Projektes ist dies durch Kundenmeilensteine und Coaching Meilensteine in den Projektvorgaben vorgeschrieben was zuerst fertiggestellt werden sollte, sofern keine anderen Absprachen getroffen wurden. In dieser Reihenfolge ist eine Priorisierung in ungefährer Reihenfolge vorgeschrieben. Es entsteht dadurch z.B. ein geringeres Risiko am Ende der Implementierung der Sortieranlage einen Fehler in den Aktoren oder Sensoren zu finden.

## Testziele, Testthemen und Teststrategie

Beim Testen soll darauf geachtet werden, dass Mehrfachprüfungen – also von einer anderen Person – vorhanden sind, sodass Grenzen, inverse Relationen, Error Zustände und Geschwindigkeit getestet werden.

Letztendlich soll jedoch das System zunächst vor dem Abnahmetest auf die Einhaltung der Requirements geprüft werden.

Beim Start des Systems soll es zudem die Möglichkeit geben einen Komponententest der HAL vorzunehmen um nicht funktionierende Komponenten zu erkennen.

## Qualitätsmaßnahmen

In jedem Projekt gibt es bestimmte Methoden und Prozesse, um Qualitätsanforderungen zur Qualitätssicherung auch Qualitätsmaßnahmen genannt. Folgende Prozesse haben wurde zur Entwicklung des Conveyor Belt Systems angewandt um eine Qualitätssicherung des Systems sicherzustellen.

Die Qualitätsanforderungen an das Projekt spiegeln sich zum größten Teil in den Abnahmetests unter 6.7.9 wieder und richten sich stark nach den Anforderungen des Kunden.

Die Qualitätssicherung wird dann mit diesen Abnahmetests durchgeführt um die Funktionsfähigkeit für den Kunden zu garantieren aber auch zu zeigen.

Folgende Qualitätsmaßnahmen haben dann jedoch zur Umsetzung des Projektes geholfen.

### Feature Driven Development

Das Projekt wurde nach den Requirements in einzelne Features aufgeteilt die die Entwicklung voranzubringen sollen. Features können dann zusammengefasst werden um Meilensteine zu bilden und Fortschritte gegenüber dem Kunden darzubieten.

### Benutzte Agile Methoden zur Verbesserung der Software Qualität

#### Code Reviews

Code Reviews dienen dazu, dass der programmierte Code eines Entwicklers von einem anderen Entwickler gesichtet wird und dort auf Code Smells, Style Guidelines und mögliche Defekte hinweist. Der Programmierer muss die Fehler vor dem Mergen des Feature Branches auf develop korrigieren.

#### Pair Programming

Im Pair Programming sitzen min. 2 Entwickler zusammen wobei einer der Entwickler Programmiert und der andere auf mögliche Defekte, Code Smells und mehr schon während der Entwicklung hinweist. Eine solche Methode führt zu deutlich weniger Anfälligkeiten während des Code Reviews und zu einem schneller umgesetzten Feature.

Bei diesem Projektablauf geschah das Pair Programming vor allem dann, wenn zwei Komponenten mit einander integriert wurden. Dabei ist einer der Entwickler als Schreiber des Codes tätig und der andere als Begutachter oder auch als Tester an der Hardware. Dies wurde vor allem sinnvoll als die Entwicklung durch begrenzte Mittel (zu wenig Laufbänder um diese mit der Konkurrenz zu teilen) ins Stocken begann.

#### Kanban

Im Projekt für den Kunden wird Feature Orientiert an einem Kanban Board gearbeitet. An diesem können alle Entwickler und auch der Kunde den Projektfortschritt begutachten. Zudem könne Behinderungen für nicht am Feature arbeitende im Standup geklärt werden. Falls Probleme auftreten können diese mit Unterstützung im Pair Programming gelöst werden.

Das Kanban Board des Projektes ist im Gitlab Projekt für den Kunden einsehbar. Dort werden vor Abnahme auch noch die vorhandenen Bugs eingepflegt um den Kunden einen Stand zu übergeben.

### Umgesetzte Qualitätskriterien nach ISO 25010

#### Funktional

Dieses Produkt weist nach Anforderungen des Kunden eine funktional korrekte Software auf, dies wird bestätigt durch funktionierende Abnahmetests.  
Weiterhin weist das Produkt keine vom Kunden ungewünschte Softwareelemente auf.

#### Performance

Die Umsetzung des Produktes ist darauf angesetzt eine am Hardwarelimit arbeitendes Laufband umzusetzen. Dabei wird am optimalsten nach Wegen gesucht die gewünschte Sortierung auszugeben.

#### Sicherheit

Das Produkt schützt sich und den Kunden vor Manipulationen im Rahmen des Möglichen, so werden hinzugefügte Puks, entfernte Puks oder auch Beeinflussung von Puks auf dem Laufband erkannt. Weiterhin erkennt das Produkt ob sich ein Status von Laufband 1 zu Laufband 2 verändert hat.

#### Kompatibel

Aufgrund der Spezialisierung des Laufbandes auf ein festgelegtes System ist eine Spezialisierung nicht auszuschließen, jedoch erlaubt es die Kapselung von Modulen und auch Hardwareschnittstellen das Produkt auf anderen Systemen mit geringer Anpassung zum Laufen zu bringen.

#### Verlässlich

Das Systemversucht im weites gehenden eine hohe Fehlertoleranz aufzuweisen, jedoch ist das System sehr anfällig gegenüber menschlichen und auch Hardware technischen Fehlern, da diese nicht selbst behoben werden können. Das Produkt kann seinen eigenen Status ab einem bestimmten Moment nicht wiederherstellen.

#### Usability

Das System muss von geschulten Personal benutzt werden, eine Bedienungsanleitung liegt bei. Schulungen können angefordert werden.

#### Wartbarkeit

Das System basiert auf mehreren State Machines und ist somit sehr modular aufgebaut. Weitere Features können durch eine weitere State Machine umgesetzt werden. Diese State Machines sind als Client integriert. Die Clients steuern alle weiteren Hardware Elemente und sind sehr modular gehalten, was eine hohe Wartbarkeit gewährleistet.

#### Portierbarkeit

Eine Installation des Produktes ist leider nur durch geschultes Personal möglich.

### Komponententests

Es werden Komponententests angewendet um einzelne Komponenten wie die serielle Schnittstelle auf ihre Funktionalität zu überprüfen. So kann die Funktionalität der seriellen Schnittstelle ohne das Zusammenspiel von anderen Komponenten bestätigt werden.

### Integrationstests

Im Projekt wohldefinierte Integrationstests sind dafür da die Kommunikation zwischen den Komponenten zu testen und ein Zusammenspiel durchzuführen. So wird die Ampel dieses Projekt im Zusammenspiel mit der HAL, dem Timer und dem AmpelController getestet. Hierunter fallen auch Systemtests die im Zusammenhang mit den State Machines stattfindet um die gesamt Funktionalität des Systems sicherzustellen. Relevant ist es vor allem, da eine Umsetzung der Systemlogik ohne kleinere Komponenten nicht möglich ist.

### Abnahmetests

Es werden Abnahmetests auf Basis der Requirements des Kunden auserarbeitet. Diese Tests sorgen dafür, dass vor der Übergabe an den Kunden, Kundenspezifische Tests zur Überprüfung der Funktionalität vorhanden sind. Diese sollen dem Kunden zeigen, dass die Anlage zuverlässig auf unterschiedliche Szenarios (auch Fehler Szenarios) reagieren kann.

### Komponententest Checkliste

Komponententest Checkliste

Diese Checkliste soll sicherstellen, dass die Hardware der Anlage und dessen Steuerung ohne Probleme funktioniert.

Beim Testen der Aktorik wird jedes Element für 3 Sekunden getestet.

|  |  |  |
| --- | --- | --- |
| Funktioniert folgende Hardware Komponente? | Ja | Nein |
| Beispiel: Ist die Sortieranlage vorhanden: | ☑ | ☐ |
| Anlage im seriellen Modus | ☐ | ☐ |
| Leuchtet Ampel Grün | ☐ | ☐ |
| Leuchtet Ampel Gelb | ☐ | ☐ |
| Leuchtet Ampel Rot | ☐ | ☐ |
| Blinkt Ampel Grün Schnell | ☐ | ☐ |
| Blinkt Ampel Grün Langsam | ☐ | ☐ |
| Blinkt Ampel Gelb Schnell | ☐ | ☐ |
| Blinkt Ampel Gelb Langsam | ☐ | ☐ |
| Blinkt Ampel Rot Schnell | ☐ | ☐ |
| Blinkt Ampel Rot Langsam | ☐ | ☐ |
| Motor Rechtslauf | ☐ | ☐ |
| Motor Rechtslauf Langsam | ☐ | ☐ |
| Motor Linkslauf | ☐ | ☐ |
| Motor Linkslauf Langsam | ☐ | ☐ |
| Motor Stoppt | ☐ | ☐ |
| Weiche öffnet | ☐ | ☐ |
| LED Starttaste | ☐ | ☐ |
| LED Resettaste | ☐ | ☐ |
| LED Q1 | ☐ | ☐ |
| LED Q2 | ☐ | ☐ |

Beim Testen der Sensorik wird nach Aktivierung des Sensors die grüne Ampel LED für 2 Sekunden aktiviert.

|  |  |  |
| --- | --- | --- |
| Einlauf Werkstück | ☐ | ☐ |
| Werkstück in Höhenmessung | ☐ | ☐ |
| Höhenmessung | ☐ | ☐ |
| Werkstück in Weiche | ☐ | ☐ |
| Werkstück Metall | ☐ | ☐ |
| Weiche offen | ☐ | ☐ |
| Rutsche voll | ☐ | ☐ |
| Auslauf Werkstück | ☐ | ☐ |
| Taste Start | ☐ | ☐ |
| Taste Stop | ☐ | ☐ |
| Taste Reset | ☐ | ☐ |
| Taste E-Stop | ☐ | ☐ |

### Integration Tests

### Abnahmetests

Voraussetzungen für die Abnahmetests: Das Laufband wurde initialisiert und eingemessen. Nach einem Abnahmetest wird das Laufband resettet und mit dem Start Button gestartet um einen weiteren Abnahmetest ausführen zu können.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 1 | **Name** | Reihenfolge einhalten FB2 | **Priorität** | hoch |
| **Input Daten** | Bohrung oben mit Metall -> Bohrung oben ohne  Metall -> Bohrung oben mit Metall -> Bohrung oben ohne Metall | | | | |
| **Output Daten** | Bohrung oben mit Metall, Bohrung oben ohne  Metall, Bohrung oben ohne Metall | | | | |
| **Ablaufbeschreibung** | Am Förderband 2 kommen die oben genannten Werkstücke in genannter Reihenfolge an. Das erste und das zweite Werkstück entsprechen der gewünschten Reihenfolge und gelangen an das Ende des Förderbandes. Das dritte Werkstück nicht, daher wird es über die Rampe aussortiert. Das vierte Werkstück ist das letzte gewünschte Werkstück und wird auch bis zum Ende des Förderbandes transportiert. | | | | |
| **Abdeckung Requirement** | RE FB2 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 2 | **Name** | Rampe 1 voll | **Priorität** | hoch |
| **Input Daten** | Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 | | | | |
| **Output Daten** | Keine Puks kommen ans Ende des Laufbandes an. Die Rampe sollte mit 4 Puks voll sein. Eine Warnung sollte angezeigt werden auf Laufband 1. | | | | |
| **Ablaufbeschreibung** | Am Förderband 1 sollen die Inputdaten in Reihenfolge einzeln mit Abstand (3 Puks Abstand) aufgelegt werden. Alle Puks sollen auf Rampe 1 aussortiert werden bis die Lichtschranke der Rampe 1 dauerhaft durch einen Puk unterbrochen ist. Sobald dies für mehrere Sekunden der Fall ist soll die Gelbe Warnleuchte des Förderbandes 1 angehen, und keine Aussortierung mehr auf Förderband 1 erfolgen. Nach der Abnahme der Puks durch einen Mitarbeiter soll die Warnung automatisch wieder erlöschen. Der fünfte Puk sollte vor der Abnahme der Puks auf Laufband 2 aussortiert werden. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 3 | **Name** | Rampe 2 voll | **Priorität** |  |
| **Input Daten** | Werkstück Typ 2 🡪 Werkstück Typ 2 🡪 Werkstück Typ 2 🡪 Werkstück Typ 2 🡪 Werkstück Typ 2 | | | | |
| **Output Daten** | Laufbände 1 und 2 halten an und es blinkt eine zu quittierende Warnung auf Laufband 2 auf | | | | |
| **Ablaufbeschreibung** | Es werden die Werkstücke wie die Input Daten beschrieben einzeln auf Laufband 1 in Lichtschranke 1 gesetzt (mit Abstand zueinander). Die Werkstücke werden allesamt einzeln auf Laufband 2 aussortiert bis die Rampe 2 voll ist. Eine Aussortierung auf Laufband 1 ist nicht vorgesehen, da der fünfte Puk in der letzten Lichtschranke des ersten Laufbandes liegt. Da keine weiteren Werkstücke auf Laufband 2 mehr sortiert werden können halten die Laufbänder an bis die Rampe 2 geleert wurde. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 4 | **Name** | Werkstück im Betrieb verschwunden | **Priorität** |  |
| **Input Daten** | Werkstück Typ 2 🡪 Werkstück Bohrung oben mit Metall | | | | |
| **Output Daten** | Die Laufbänder halten und beide Laufbänder sind im Error Zustand | | | | |
| **Ablaufbeschreibung** | Es wird Werkstück Typ 2 auf das Laufband 1 in die Lichtschranke 1 gesetzt. Das Laufband fährt nun los. Während das Werkstück Typ 2 die Lichtschranke des Höhensensors durchquert wird ein weiteres Werkstück Bohrung oben mit Metall auf das Laufband gepackt. Nach durchqueren des LB\_1 wird dieses Werkstück wieder entfernt.  Nach einigen Sekunden merkt der Automat, dass das Werkstück nicht am Höhensensor angekommen ist und wechselt in den Error Zustand. Der Errorzustand wird danach quittiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 5 | **Name** | Werkstück im Betrieb aufgetaucht | **Priorität** |  |
| **Input Daten** | Werkstück Typ 2 | | | | |
| **Output Daten** | Error stoppt beide Laufbänder und die Ampel leuchtet rot | | | | |
| **Ablaufbeschreibung** | Es wird das Werkstück Typ 2 auf das Laufband 1 in die Lichtschranke 1 gesetzt. Der Puk läuft über das Laufband 1. Nachdem die Schranke öffnet und Typ 2 durchfährt wird die Lichtschrank des Höhensensors unterbrochen. Danach wird die Unterbrechung wieder aufgehoben. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 6 | **Name** | Es befindet sich immer nur ein Puk auf Laufband 2 | **Priorität** | Niedrig (Redundant) |
| **Input Daten** | Werkstück Typ 2 🡪 Werkstück Typ 2 | | | | |
| **Output Daten** | Es werden beide Werkstücke auf Laufband 2 aussortiert | | | | |
| **Ablaufbeschreibung** | Es werden beide Werkstücke auf Laufband 1 mit Abstand (3 Pukdurchmesser) aufgelegt. Beide sollen auf Laufband 2 aussortiert werden. Der erste Puk wird erfolgreich auf Laufband 2 übergeben, dann stopp Laufband 1 wenn der zweite Puk in die Lichtschranke am Ende des Bandes eintritt. Sobald der Puk auf Laufband 2 aussortiert wurde und das Band somit wieder frei ist, wird der zweite Puk auf das Laufband 2 übergeben und dort aussortiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 7 | **Name** | Nebenläufigkeit Band 1 | **Priorität** | Sehr hoch |
| **Input Daten** | Werkstück Typ 2 🡪Werkstück Typ 1 | | | | |
| **Output Daten** | Laufband läuft und sortiert Werkstücke bis Werkstück Typ 2 die Lichtschranke Richtung Laufband 2 unterbricht. | | | | |
| **Ablaufbeschreibung** | Es werden die Inputdaten mit einem Abstand von 2 Puks auf das Laufband 1 aufgelegt. Typ 2 wird durch die Schranke gelassen. Werkstück Typ 1 wird aussortiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 8 | **Name** | Werkstück Daten werden ausgegeben am Ende von Laufband 2 | **Priorität** | Niedrig |
| **Input Daten** | Werkstück Bohrung oben mit Metall | | | | |
| **Output Daten** | Werkstück-ID Typ Höhenmesswert von FB 1 Höhenmesswert von FB 2 | | | | |
| **Ablaufbeschreibung** | Es werden die Input Daten in die Lichtbarriere 1 auf Laufband 1 gelegt. Beim Eintreffen in Lichtbarriere 2 auf Laufband 2 werden die Output Daten des Werkstücks angezeigt. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 9 | **Name** | Werkstück Daten eines Typ Bausteins wird ausgegeben | **Priorität** | niedrig |
| **Input Daten** | Werkstück Typ 1 | | | | |
| **Output Daten** | Zeitstempel der Erkennung  Werkstück-ID  Binärcode  Höhenmesswert | | | | |
| **Ablaufbeschreibung** | Es werden die Input Daten auf Laufband 1 aufgelegt. Beim Austreten aus dem Bereich der Höhenmessung werden die Output Daten über die Konsole ausgegeben. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 10 | **Name** | Werkstück Überschlag | **Priorität** |  |
| **Input Daten** | Werkstück Bohrung oben mit Metall | | | | |
| **Output Daten** | Werkstück wird aussortiert | | | | |
| **Ablaufbeschreibung** | Es werden die Input Daten auf Laufband 1 gelegt. Während des Übergebens des Werkstückes von Laufband 1 auf Laufband 2 wird das Werkstück gedreht. In der Höhenmessung wird erkannt, dass das Werkstück keine Bohrung mehr besitzt und das Werkstück wird danach aussortiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 11 | **Name** | Höhenmessung Band läuft langsam | **Priorität** |  |
| **Input Daten** | Werkstück Typ 2 | | | | |
| **Output Daten** | Werkstück Typ 2 wird langsam durch die Höhenmessung transportiert und auf Laufband 2 aussortiert | | | | |
| **Ablaufbeschreibung** | Die Input Daten werden auf das Laufband 1 gelegt. Während des Transportes durch die Höhenmessung 1 läuft Band 1 im langsam Modus. Während des Transportes durch die Höhenmessung 2 läuft Band 2 im langsam Modus. Danach wird das Werkstück aussortiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 12 | **Name** | Rampe 1 voll Aussortierung auf Band 2 | **Priorität** | hoch |
| **Input Daten** | Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 | | | | |
| **Output Daten** | Keine Puks kommen ans Ende des Laufbandes 2 an. Die Rampe sollte mit 4 Puks voll sein. Eine Warnung sollte angezeigt werden auf Laufband 1. Laufband 2 empfängt die restlichen Puks auf Rampe 2. | | | | |
| **Ablaufbeschreibung** | Am Förderband 1 sollen die Inputdaten in Reihenfolge einzeln mit Abstand (3 Puks Abstand) aufgelegt werden. Alle Puks sollen auf Rampe 1 aussortiert werden bis die Lichtschranke der Rampe 1 dauerhaft durch einen Puk unterbrochen ist. Sobald dies für mehrere Sekunden der Fall ist soll die Gelbe Warnleuchte des Förderbandes 1 angehen, und keine Aussortierung mehr auf Förderband 1 erfolgen. Nach der Abnahme der Puks durch einen Mitarbeiter soll die Warnung automatisch wieder entfernen.  Das Laufband übergibt nun die Puks an Laufband 2 auf welchem die restlichen Werkstücke des Typs 1 aussortiert werden. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 13 | **Name** | Rampe 1 voll Aussortierung auf Band 2 Reihenfolge aber halten | **Priorität** | hoch |
| **Input Daten** | Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Bohrung oben mit Metall 🡪 Werkstück Typ 1 🡪 Werkstück Bohrung oben ohne Metall 🡪 Werkstück Bohrung oben mit Metall 🡪 Werkstück Bohrung oben ohne Metall | | | | |
| **Output Daten** | Keine Puks kommen ans Ende des Laufbandes an. Die Rampe sollte mit 4 Puks voll sein. Eine Warnung sollte angezeigt werden auf Laufband 1. | | | | |
| **Ablaufbeschreibung** | Am Förderband 1 sollen die Inputdaten in Reihenfolge einzeln mit Abstand (3 Puks Abstand) aufgelegt werden. Alle Puks sollen auf Rampe 1 aussortiert werden bis die Lichtschranke der Rampe 1 dauerhaft durch einen Puk unterbrochen ist. Sobald dies für mehrere Sekunden der Fall ist soll die Gelbe Warnleuchte des Förderbandes 1 angehen, und keine Aussortierung mehr auf Förderband 1 erfolgen. Nach der Abnahme der Puks durch einen Mitarbeiter soll die Warnung automatisch wieder entfernen.  Das Laufband übergibt nun die Puks an Laufband 2 auf welchem die restlichen Werkstücke des Typs 1 aussortiert werden. Das Werkstück mit Bohrung oben mit Metall trifft in der letzten Lichtschranke ein und wird von einem Mitarbeiter abgenommen. Werkstück Typ 1 wird wieder aussortiert. Werkstück Bohrung oben ohne Metall kommt auch in Lichtschranke 2 an und wird von einem Mitarbeiter abgenommen. Werkstück oben mit Metall wird nun auch aussortiert um die Reihenfolge zu halten. Der letzte Puk wird nun an Laufband 2 übergeben und kommt nach Erkennung auch erfolgreich an der letzten Lichtschranke des Laufbandes 2 an und wird von einem Mitarbeiter abgenommen. Sollte Rampe 2 während des Aussortierens voll sein halten beide Bänder, da auch Rampe 1 voll ist. Ein Mitarbeiter leert dann Rampe 2 aber nicht Rampe 1. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 14 | **Name** | Kalibrierung und Selbsttests | **Priorität** | niedrig |
| **Input Daten** | Start Knopf wird 4 Sekunden gedrückt | | | | |
| **Output Daten** | Selbsttests und Kalibrierung wird ausgeführt und mit einer grün leuchtenden Ampel quittiert | | | | |
| **Ablaufbeschreibung** | Es wird der Start Knopf für 4 Sekunden gedrückt. Danach wird die Routine für die Kalibrierung und die Selbsttests ausgeführt. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 15 | **Name** | E-Stop | **Priorität** | mittel |
| **Input Daten** | E-Notaus wird gedrückt | | | | |
| **Output Daten** | Laufbänder sind beide gestoppt bis Reset ausgeführt wird | | | | |
| **Ablaufbeschreibung** | Es wird der Notaus an Laufband 1 betätigt. Beide Bänder stoppen und die rote Lampe leuchtet dauerhaft. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 16 | **Name** | Stop | **Priorität** |  |
| **Input Daten** | Stop Taste wird gedrückt | | | | |
| **Output Daten** | Laufband ruht | | | | |
| **Ablaufbeschreibung** | Durch das betätigen des Stop Tasters wird das Laufband in den Ruhezustand geschaltet. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 17 | **Name** | Stop während Error | **Priorität** |  |
| **Input Daten** | E-Notaus 🡪 Stop | | | | |
| **Output Daten** | Es passiert nichts, der Errorzustand bleibt erhalten | | | | |
| **Ablaufbeschreibung** | Es wird ein Notaus betätigt und 2 Sekunden danach die Stoptaste gedrückt. Die Ampel leuchtet weiterhin und der Automat bleibt im Error State. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 18 | **Name** | E-Stop Quittierung | **Priorität** |  |
| **Input Daten** | E-Notaus wird gedrückt, Notaus wird wieder gezogen und quittiert | | | | |
| **Output Daten** | Laufband läuft wieder | | | | |
| **Ablaufbeschreibung** | Es wird der Notaus an Laufband 2 betätigt. Beide Bänder stoppen und die rote Lampe leuchtet. Der Notaus wird an Laufband 2 wieder gezogen und danach mit Reset quittiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 19 | **Name** | E-Stop an beiden Bändern | **Priorität** |  |
| **Input Daten** | E-Notaus wird an beiden Bändern gedrückt, an Laufband 1 gezogen und quittiert | | | | |
| **Output Daten** | Beide Laufbänder befinden sich immer noch im Error unquittiert Zustand. | | | | |
| **Ablaufbeschreibung** | Es wird einzeln das Notaus an beiden Laufbändern gedrückt. An einem Laufband wieder gezogen und mit dem Quittierungsbutton quittiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 20 | **Name** | Serielle Schnittstelle ist Disconnected | **Priorität** |  |
| **Input Daten** | Es wird bei Betrieb das Serielle Kabel gezogen | | | | |
| **Output Daten** | Beide Laufbändern befinden sich im Error Zustand | | | | |
| **Ablaufbeschreibung** | Beide Laufbänder sind im Betrieb (Ampel leuchtet dauerhaft grün) dann wird das serielle Kabel gezogen. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 21 | **Name** | Bohrung unten | **Priorität** |  |
| **Input Daten** | Werkstück Bohrung unten mit Metall | | | | |
| **Output Daten** | Ein Werkstück in Rampe auf Laufband 1 | | | | |
| **Ablaufbeschreibung** | Es wird das Werkstück mit Bohrung unten auf Laufband 1 nach Bedienungsanleitung aufgelegt und auf Laufband 1 sofort aussortiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 22 | **Name** | Beide Rampen voll | **Priorität** |  |
| **Input Daten** | Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 🡪 Werkstück Typ 1 | | | | |
| **Output Daten** | Es wird ein Error ausgegeben auf beiden Laufbändern | | | | |
| **Ablaufbeschreibung** | Es werden alle Werkstücke nach Bedienungsanleitung aufgelegt. Beide Rampen sollen voll sein. | | | | |
| **Abdeckung Requirement** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. / Test-ID** | 23 | **Name** | Rampe 2 voll auf Laufband 1 aussortieren | **Priorität** |  |
| **Input Daten** | Werkstück Typ 2 🡪 Werkstück Typ 2 🡪 Werkstück Typ 2 🡪 Werkstück Typ 2 🡪 Werkstück Typ 2 🡪 | | | | |
| **Output Daten** | Warnung wird auf Laufband 2 angezeigt | | | | |
| **Ablaufbeschreibung** | Es werden 4 Werkstücke Typ 2 nach Bedienungsanleitung aufgelegt. Sobald diese 4 Werkstücke Typ 2 auf Laufband 2 aussortiert wurden wird ein weiteres Werkstück Typ 2 auf Laufband 1 aufgelegt und auf Laufband 1 aussortiert. | | | | |
| **Abdeckung Requirement** |  | | | | |

## Test Protokoll

|  |  |  |  |
| --- | --- | --- | --- |
| Abnahme Test Id | Problem? | Funktioniert:  Ja | Funktioniert:  Nein |
| Beispiel 0 | Test läuft nicht durch | ☐ | ☑ |
| 1 |  | ☐ | ☐ |
| 2 |  | ☐ | ☐ |
| 3 |  | ☐ | ☐ |
| 4 |  | ☐ | ☐ |
| 5 |  | ☐ | ☐ |
| 6 |  | ☐ | ☐ |
| 7 |  | ☐ | ☐ |
| 8 |  | ☐ | ☐ |
| 9 |  | ☐ | ☐ |
| 10 |  | ☐ | ☐ |
| 11 |  | ☐ | ☐ |
| 12 |  | ☐ | ☐ |
| 13 |  | ☐ | ☐ |
| 14 |  | ☐ | ☐ |
| 15 |  | ☐ | ☐ |
| 16 |  | ☐ | ☐ |
| 17 |  | ☐ | ☐ |
| 18 |  | ☐ | ☐ |
| 19 |  | ☐ | ☐ |
| 20 |  | ☐ | ☐ |
| 21 |  | ☐ | ☐ |
| 22 |  | ☐ | ☐ |
| 23 |  | ☐ | ☐ |
| 24 |  | ☐ | ☐ |
| 25 |  | ☐ | ☐ |
| 26 |  | ☐ | ☐ |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Require-ment | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| Test |
| 1 |  |  | X | X | X | X |  | X | X | X |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |
| 2 | X |  | X | X | X |  |  | X |  | X | X | X | X |  | X | X |  |  |  |  |  |  |  |  |  |  | X |  | X |  |  | X |  |  |  | X |  |  |
| 4 | X |  | X | X | X |  |  | X |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  | X |  |  | X |  |  | X | X | X | X | X |  | X |  |  |
| 5 | X |  | X |  |  |  |  | X |  |  | X | X |  |  |  | X |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  |  | X |  | X |  | X |  |
| 10 |  | X\* |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |
| 21 |  | X\* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 22 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23 |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Test Metrik (Relevante Tests zur Requirementabdeckung)

# Bedienungsanleitung

## Vorwort

Sehr geehrter Kunde,

wir freuen uns, dass Sie sich für dieses Produkt entschieden haben. Es bietet ihnen höchste Qualität und neueste Technologie.

Um die volle Leistungsfähigkeit des Produktes nutzen zu können, lesen Sie sich bitte vor dem Anschließen und der Inbetriebnahme diese Bedienungsanleitung sorgfältig durch. Die Betriebssicherheit und auch die Funktion des Gerätes können nur dann gewährleistet werden, wenn die Sicherheitsvorschriften sowie auch die Sicherheitshinweise in dieser Bedienungsanleitung beachtet werden. Wir übernehmen keine Haftung für Schäden, die durch unsachgemäßen Gebrauch bzw. fehlerhafte Bedienung entstehen.

Bitte stellen Sie sicher, dass alle Personen, die dieses Gerät bedienen diese Bedienungsanleitung gelesen und auch verstanden haben.

Bewahren Sie diese Betriebsanleitung an einem sicheren und zugänglichen Ort auf um Sie jederzeit Griffbereit zu haben.

## Merkmale

Das 2-Laufband System welches mit diesem Produkt in Verbindung steht dient zur Sortierung von Bausteinen mit verschiedenen Eigenschaften.

Das Laufband System hat gewisse Einschränkungen bzgl. Genauigkeit der Sensoren und Hardware, zusätzlich sind verschiedene Anlagen nicht aufeinander geeicht, sodass unterschiedliche Laufgeschwindigkeiten und Höhenmesswerte vorhanden sind.

All dies wird vom dem Produkt abgefangen und führt zu einem möglichst optimalen Zusammenspiel zweier Laufbänder zur Sortierung von Werkstücken.

## Sicherheitshinweise

### Betriebsanleitung beachten

Jede Inbetriebnahme und Handhabung des Produktes setzt die genaue Kenntnis und Beachtung dieser Bedienungsanleitung voraus. Das Produkt ist nur für die in der Betriebsanleitung beschriebenen Abläufe bestimmt.

### Haftung für Funktion und Schäden

Die Haftung für die Funktion des Produktes geht in jedem Fall auf den Betreiber über, wenn das Produkt nicht von autorisiertem geschulten Personal bedient wird. Das Gerät muss gemäß der Betriebsanleitung betrieben werden und die Haftung bleibt bei nicht Beachtung auf jeden Fall beim Betreiber.

## Anwendungsbereich

Das Produkt ist eine Software zur Sortierung auf 2 Laufbändern für Werkstücke in Art eines Puks. Das Produkt soll ausschließlich zur Sortierung der beiliegenden Puks verwendet werden und ist nicht für andere Elemente ausgelegt.

## Inbetriebnahme

Diese Inbetriebnahme ist nach jedem Stromverlust durchzuführen.

1. Laufband 1 wird mit einem Stromkabel an das Stromnetz angeschlossen
2. Laufband 2 wird mit einem Stromkabel an das Stromnetz angeschlossen.
3. Bevor die zum Produkt gehörenden Laufbänder zusammen mit dem Produkt in Betrieb genommen werden können, muss das Produkt von autorisierten Fachpersonal auf die Gerätschaften aufgespielt werden. Eines der Laufbänder wird in diesem Falle als Laufband 1 konfiguriert und eines als Laufband 2.

*Zum autorisierten Fachpersonal kann auch ein Mitarbeiter des Betreibers gehören, sofern er eine ausführliche Schulung zum Produkt erhalten hat welche ein Aufspielen des Produktes erlaubt.*

Es ist folgender Ablauf zur weiteren Inbetriebnahme einzuhalten:

1. Es ist Laufband 1 mit einem seriellen Kabel mit Laufband 2 zu verbinden.
2. Laufband 1 und Laufband 2 in den Kalibrierungsmodus zu stellen. (Siehe Kapitel Bedienung
   1. Sobald der Kalibrierungslauf auf beiden Laufbändern läuft wird auf den Laufbändern die serielle Kommunikation überprüft. Eine erfolgreiche Verbindung wird mit einer dauerhaft grün blinkenden Ampel auf beiden Laufbändern signalisiert. Eine noch nicht vorhandene Verbindung wird mit einer rot blinkenden Ampel auf beiden Laufbändern signalisiert.
   2. Sofern eine rot blinkende Ampel auftaucht muss der Kalibrierungslauf neu gestartet werden.
3. Auf Laufband 1 und 2 jeweils einen roten Puk (Siehe Kapitel Elemente) mit Bohrung nach unten in die Lichtbarriere 1 stellen.
4. Auf Laufband 1 und 2 mit dem Knopf „Stop“ jeweils die Kalibrierung auf den Laufbändern starten.
5. „Start“ Knopf zwei Mal Betätigen bis die Anlage startet und die grüne Ampel blinkt
6. Mit einer weiteren Betätigung des „Start“ Knopfes kann der Sortierungslauf beginnen.

## Bedienung

### Kalibrierung des Laufbandes starten

1. „Stop“ für mehrere Sekunden gedrückt halten und „Start“ drücken.  
   Sobald die grüne Ampel leuchtet einen roten Puk mit Bohrung ohne Metall umgedreht (Siehe Kapitel Elemente) in erste Lichtschranke bis ganz an die Seite des Bedienfeldes stellen.
2. Den „Start“ Knopf betätigen
3. Abwarten bis der Puk von der Machine aussortiert wird.

### Einzelne Sortieranlage zurücksetzen

Nach einer Betätigung des Reset Schalters, schaltet sich das Laufband in den Startzustand der Sortieranlage zurück. Es kann zu unvorhersehbaren Verhalten kommen, wenn das Laufband nicht geräumt wird, zudem kann die Verbindung mit Laufband 2 Probleme verursachen, wenn dieses nicht ebenfalls resetet wird.

1. Reset drücken
2. Laufband leeren
3. Start drücken

### E-Schnellstopp benutzen und Sortierung neu starten

1. Not-Aus Schalter betätigen
2. Laufband hält
3. Not-Aus Schalter ziehen
4. Laufband hält
5. Laufband leeren
6. Quittierung betätigen

### Quittieren

1. „Reset“ Knopf drücken

### Sortieranlage nach Inbetriebnahme starten

Nach einer Inbetriebnahme muss ein Laufbandverbund gestartet werden.

1. An Laufband den „Start“ Knopf drücken

### Sortieranlage nach einem Reset starten

Nach einem Reset muss ein einzelnes Laufband neu gestartet werden.

1. Laufband leeren
2. Am betroffenen Laufband Start drücken

### Puks dem Laufband zuführen

Damit die Sortierung ordnungsgemäß funktioniert muss ein Abstand von 2 Puklängen zwischen Puks existieren.

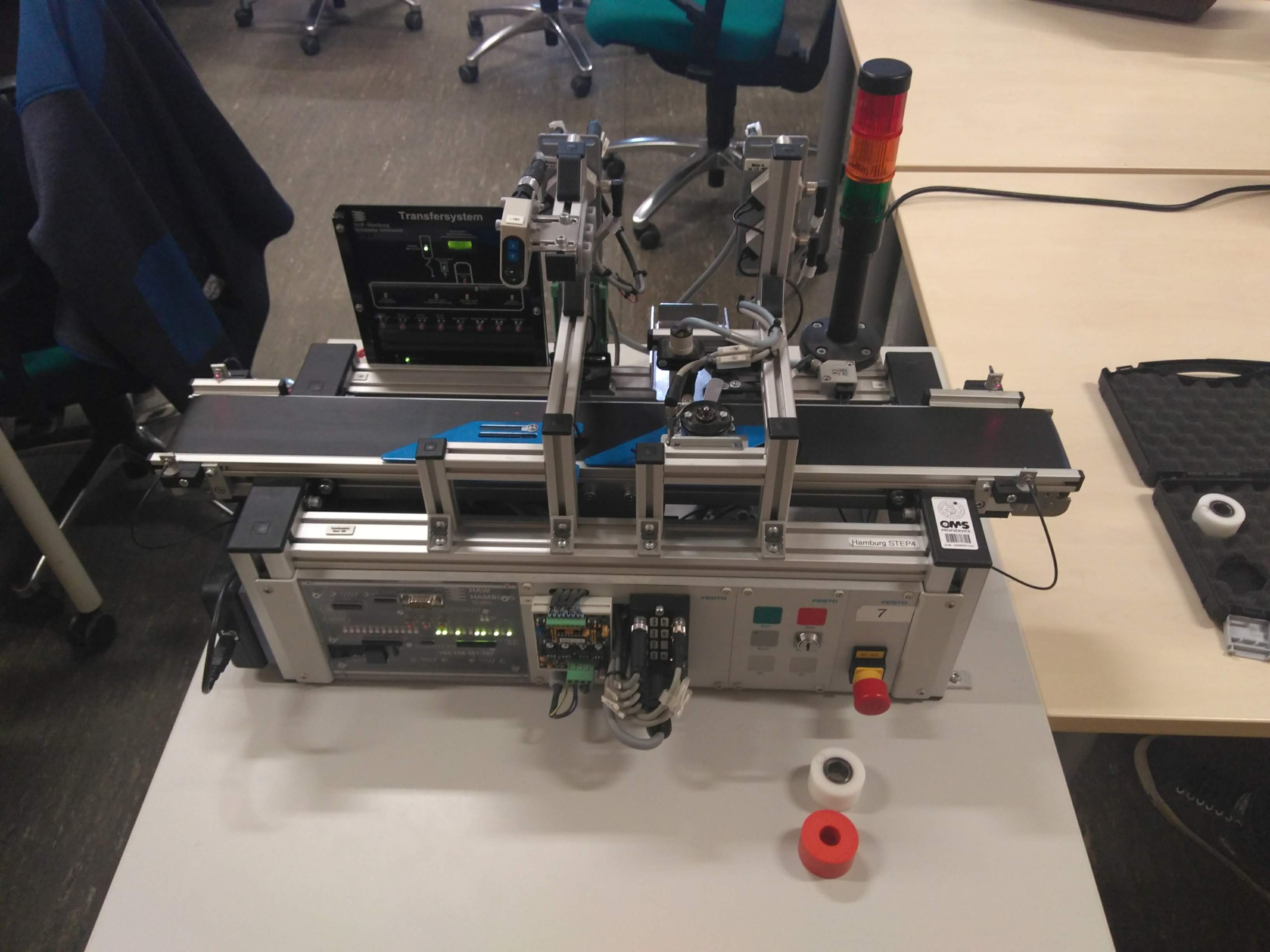
1. Ersten Puk auflegen
2. Zweiten Puk erst auflegen wenn zwischen den beiden Puks eine Leerfläche der Länge von ca. 2 Puks existiert.

## Störungssuche / Fehlerbehebung

|  |  |  |
| --- | --- | --- |
| **Fehler / Störung** | **Mögliche Ursache** | **Mögliche Lösung** |
| Ampel leuchtet Rot | Es befindet sich ein Puk in der zweiten Lichtschranke von Laufband 2 | Puk abnehmen |
| Ampel blinkt Rot | Beide Rampen sind voll | Werkstücke aus einer Rampe entfernen |
| E-Stop wurde ausgelöst | E-Stop an beiden Maschinen lösen und quittieren |
| Die serielle Verbindung wurde unterbrochen | Serielles Kabel wieder verbinden |
| Beide Laufbänder neu kalibrieren |
| Ampel leuchet Gelb | Nach dem Kalibrierungslauf konnte keine serielle Verbindung zum zweiten Laufband herstellen | Serielles Kabel an beide Laufbänder anschließen |
| Zweites Laufband durch die Kalibrierung laufen lassen |
| Eine der Rampen ist voll | Werkstücke aus voller Rampe entfernen. |
| Falsche Puk Reihenfolge wird sortiert | Puks werden falsch erkannt | Bitte eine erneute Kalibrierung ausführen |
| Es wird ein fehlender Puk angezeigt, er ist jedoch noch vorhanden. | Äußere Einflüsse haben den Puk aus der Toleranzmarge gedrängt | Bitte eine erneute Kalibrierung ausführen. Falls dies nicht hilft Fachpersonal von Festo das Laufband überprüfen lassen |
| Es wird ein fehlender Puk angezeigt | Es wurde ein Puk vom Laufband entfernt | Das Laufband leeren und per Reset das Laufband zurücksetzen |
| Es wird ein dazugekommener Puk angezeigt | Ein Puk wurde auf das Laufband gelegt | Puk vom Laufband entfernen |
| Ampel blinkt grün | Kalibrierungslauf wurde gestartet | Kalibrierungslauf nach Bedienungsanleitung abschließen |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Elemente

## Laufband



## Puks

### Bohrung mit Metall



### Bohrung ohne Metall



### Binärcodiert 1



### Binärcodiert 2



### Binärcodiert 5



### Flach



### Ohne Bohrung



# Lessons Learned

## Langsam anfangen und dann größer werden.

Wir haben unsere Arbeit in großen Teilen auf Basis des Tutoriums erarbeitet und dadurch festgestellt, dass langsames Anfangen mehr brachte als wildes rumprobieren mit Elementen mit denen wir vorher nichts zu tun hatten. Nachdem wir unsere Grundlagen für das System gefestigt hatten haben wir unsere Systemplanung dementsprechend erweitert.   
Ein passendes Zitat dazu kommt von John Gall: „*A complex system that works is invariably found to have evolved from a simple system that worked“* [1]

## Einzelne Sachen ändern und nicht viele auf einmal

Für das Debugging ist uns bewusstgeworden, dass man nicht mehrere Sachen auf einmal in ein System einbauen sollte was noch sehr labil und anfällig gegenüber Fehler ist. Man weiß am Ende leider nicht mehr wo der Fehler herkommt.

## Früh Logging zum System hinzufügen

Nach ersten Versuchen das System zum Laufen zu bekommen ist uns bewusstgeworden, dass man ohne Logging kein kompliziertes System mehr Debuggen kann. Noch besser wäre es gewesen ein ausführliches Error System frühzeitig umzusetzen, aber das konnte Zeitlich nicht mehr rechtzeitig umgesetzt werden.

## Einzelne Teile testen bevor man es zum Projekt hinzufügt

Bevor man einzelne Komponenten zum System hinzufügt sollte man Komponenten testen und nicht nachher bei Integrationstests nicht richtig festzustellen was jetzt wirklich kaputt ist.

## Alles braucht länger als man denkt

Es wurde uns leider erst in der Mitte des Praktikums bewusst, jedoch alles braucht sobald es komplexer ist sehr viel mehr Zeit zur Umsetzung. Somit sind wir letztendlich auch durch großen Zeitaufwand im Zeitplan sehr nach hinten gerutscht.

## Mit einem Kommilitonen zu reden hilft mehr als man denkt

Auch wenn man nur auf jemanden einredet ist sehr hilfreich sein Problem einem Kommilitonen mitzuteilen. Meistens kommt man während des Vortragens schon auf das Problem.

## Von Angesicht zu Angesicht zu reden ist das Produktivste

Die Kommunikation über Textnachrichten Dienste, Telefon oder Videokonferenz sind alle nicht so gut als wenn man sich gegenübersteht.

## Über ein Problem schlafen

Manchmal ist einfach sinnvoll am Abend aufzuhören und sich nicht mehr mit dem Problem zu beschäftigen. Wir mussten feststellen, dass Probleme am nächsten Tag mit einer neuen Herangehensweise viel besser gelöst werden konnten als man vorher überhaupt dachte.

## „Wenn du willst, dass es funktioniert mach es selbst“

Leider musste eines unserer Gruppenmitglieder feststellen, dass wenn etwas richtig funktionieren soll dann sollte man es selbst machen. Man selbst hat bei bestimmten System einfach eine höhere Kompetenz die man miteinander leider nicht immer Kommunizieren kann.

## QNX Compiler ist sehr labil

Wir mussten feststellen, dass der QNX Compiler teils sehr interessante Ergebnisse produziert. So wurden Funktionen ausgeführt, obwohl diese nach Debugger gar nicht erreicht wurden.

## Lieber Enums statt Defines

Während der Entwicklung ist uns bewusstgeworden, dass Defines im Code sehr viel weniger übersichtlich sind als Enums, leider haben wir es nicht geschafft alle Defines durch Enums zu ersetzen.

# Mängelliste

|  |  |  |  |
| --- | --- | --- | --- |
| Fehler | Ursache | Lösung | Umsetzungsdauer |
| Laufband bewegt nicht langsam | Funktion nicht benötigt aber implementiert | Kundenwunsch nachfragen | 1 Std. Laufband 1  4 Std. Laufband 2 |
| Höhensensor Toleranz unzureichend | Puks sind leider nicht genau genug hergestellt | Toleranz für zu hohe Puks einfügen | 3h beide Laufbänder |
| Puks im Übergang zwischen beiden Bändern, können verschwinden | Die Laufbänder ignorieren einen beim Übergang verlorenen Puk | Implementation eines verlorenen Puks auf Laufband 2 | 2 Std. Laufband 2 |
| Laufband 2 erkennt erst nach dem ersten Puk ob Rampe 2 voll | Es wird nicht dauerhaft überprüft ob die Rampe belegt ist oder nicht | Es wird kein separater Prozess zur Überprüfung von Rampe 2 gestartet | 1 Std. Laufband 2 |
| Ausgabe entspricht nicht den Requirements | Nicht umgesetzt | Implementieren | ½ Std. Laufband 1  ½ Std. Laufband 2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |