Requirements / Design and Test Documentation

**(RDT)**

*Version 0.7*

Teilnehmer

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Matrikelnummer** | **E-Mail** | **Kürzel** |
| Ismail, Muhammad Aiman Bin | 2307874 | muhammadaimanbin.ismail@haw-hamburg.de | AI |
| Budagov, Vadim | 2305722 | vadim.budagov@haw.hamburg.de | BGV |
| Mathia, Paul | 2294123 | paul.mathia@haw-hamburg.de | PM |
| Agdas, Adem-Can | 2317217 | Adem-can.agdas@haw-hamburg.de | ACA |
| Kunow, Torben | 2316163 | Torben.kunow@haw-hamburg.de | KNW |
| Kirdas, Ömer | 2304575 | oemer.kirdas@haw-hamburg.de | KIR |

ESEP – Praktikum – WS18/19

Im Rahmen des Praktikums ‘Embedded Systems Engineering’ (kurz ESEP) wird eine Software für eine Werkstücks-Sortieranlage modelliert und entwickelt. Realisiert wird das Projekt in einem Team von sechs Studenten im vierten Semester. Dieses Dokument stellt die wesentlichen Requirements, das Design und die Tests der Software im Laufe des Projektes dar.

Änderungshistorie:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Datum** | **Anmerkungen/Änderungen** |
| 0.1 | PM, BGV, ACA | 01.10.2018 | Erstellen der Requirements |
| 0.2 | KIR | 03.10.2018 | Requirements überarbeitet (nicht vollständig) |
| 0.3 | ACA | 04.10.2018 | Teilnehmerliste aktualisiert |
| 0.4 | BGV | 05.10.2018 | Use Case Nr. 01 Kalibrierung |
| 0.5 | KIR | 08.10.2018 | Requirements überarbeitet und neue hinzugefügt |
| 0.6 | PM | 10.10.2018 | Vorläufige Version des Komponenten Diagrams |
| 0.7 | PM, ACA | 15.10.2018 | Komplettüberarbeitung des RDT (Templates entfernt, Grundstruktur hinzugefügt) |
| 0.8 | BGV | 29.10.2018 | Layout angepasst. Struktur des Dokuments bearbeitet. Einleitungen für die ersten Hauptpunkte geschrieben. |
| 0.9 | PM, AI | 29.10.2018 | Use Case und Requirement für Remote PC angefertigt und hinzugefügt. Use Case Diagramme, Komponentendiagramm und Klassendiagramm eingefügt. |
| 0.10 | PM | 31.10.2018 | Überarbeitung des RDT  Spezifkation: Embedded Recorder + Remote Lan Control  Überarbeitung + Ergänzung: USE-Case Diagramme |
| 0.11 | PM | 31.10.2018 | Vorläufige Version PSP eingefügt |
| 0.12 | BGV | 29.11.2018 | Objektdiagramm eingefügt |
| 0.13 | PM | 05.12.2018 | USE-Case überarbeitet, PSP überarbeitet |
|  |  |  |  |

NEUE VERSIONEN NUR BEI PUSH INS REPOSITORY!

**Inhaltsverzeichnis**

1. [**Teamorganisation**](#_1fob9te) **4**

1.1 [Verantwortlichkeiten](#_1id46iskcush) 4

1.2 [Teamzusammenstellung](#_vm2fpr51hzoe) 4

1.3 [Absprachen](#_tc4m4v6903bh) 4

1.4 [Repository-Konzept](#_gkin6pa1rye6) 5

2. [**Projektmanagement**](#_d8k7q89ve53b) **5**

2.1 [Prozess](#_2i93qctaevzj) 5

2.2 [PSP](#_bpjkdh8h9zs9) 5

2.3 [Qualitätssicherung](#_qkj352rhhi35) 7

3. [**Randbedingungen**](#_vifmf4a8oev7) **7**

3.1 [Arbeitsumgebung](#_4dydbc1g21kq) 7

3.2 [Programmiersprachen](#_apy9ojlteqoq) 7

4. [**Requirements und Use Cases**](#_dzi4o5uaqd06) **7**

4.1 [Requirements](#_dd987kuw7m70) 7

4.2 [Use Cases](#_7alu51sh518i) 12

5. [**Design**](#_5en5532r7rx4) **21**

5.1 [Auflistung und Beschreibung der Komponenten](#_jo5lai4ewfkw) 21

5.2 [Komponentendiagramm](#_gh70von4w2v) 22

5.3 [Klassendiagramm](#_p3t2gbv2h2wd) 23

5.4 [Objektdiagramm](#_g50ai3m0x3dw) 24

5.5 [State Machines](#_hj5xwuoh9fer) 24

6. [**Testen**](#_c58woifg6qif) **25**

6.1 [Testplan](#_9jjs5dart4ss) 25

6.2 [Tracematrix](#_5g3oluj79csl) 25

7. [**Review**](#_hlbns1r1qpzf) **25**

8. [**Anhang**](#_slpeya5jml40) **26**

8.1 [Glossar](#_a5bijv66u3bz) 26

8.2 [Abkürzungen](#_kyi0tmlqrlal) 26

# Teamorganisation

Hier wird die Teamorganisation während des Projektes beschrieben. Im folgenden Punkt sind generelle Verantwortlichkeiten aufgelistet. Dabei handelt es sich nicht um alleinige Zuständigkeiten. Jeder Teilnehmer wird in verschiedenen Themengebieten tätig sein. Des Weiteren wurden die Teammitglieder in zweier Teams aufgeteilt, um an einer Aufgabe zusammenzuarbeiten. Dadurch wird das Vier Augen Prinzip gewährleistet.

## Verantwortlichkeiten

|  |  |
| --- | --- |
| **Name** | **Generelle Zuständigkeit** |
| Paul | Projektleiter, Doku, Softwarearchitektur |
| Vadim | Softwarearchitektur, Doku |
| Ömer | Implementation, Tests |
| Torben | Implementation, Tests |
| Aiman | Implementation, Design, Git, Kommunikation |
| Adem | Implementation, Design, Lektor, Git |

## Teamzusammenstellung

|  |  |
| --- | --- |
| Mitglieder | Teamkürzel |
| Paul Mathia, Adem-Can Agdas | PA |
| Ismail, Muhammad Aiman Bin, Vadim Budagov | VI |
| Ömer Kirdas, Torben Kunow | ÖT |

## Absprachen

Es wurde im Team beschlossen, dass mindestens jeden Donnerstag ein Meeting stattfindet. Bei Bedarf bzw. Abweichungen können Meetings auf andere Tage verschoben oder um weitere Tage ergänzt werden. Bei jedem Meeting hält ein vorher gewählter Protokollant sämtliche Information und Beschlüsse im Meeting-Protokoll fest. Die Meeting-Protokolle sind dem Repository zu entnehmen. Hauptkommunikationsplatformen für die externe Kommunikation abseits der Meetings sind hierbei die Plattform Slack und eine extra angelegte WhatsApp-Gruppe.

## Repository-Konzept

Nachstehend wird der Link zum Repository bereitgestellt: (https://gitlab.informatik.haw-hamburg.de/ace722/esep-ws18.git). Jede Änderung des Projektes wird in einem neuen Branch festgehalten und erst nach Teamabsprache gemerged. Die Absprachen finden bei jedem Meeting statt, gleichzeitig werden die erledigten Aufgaben besprochen und im Team diskutiert. Jede Woche werden neue Issues eröffnet, mit deren Hilfe die neu vergebenen Aufgaben überwacht werden.

# Projektmanagement

Im Folgenden wird beschrieben, wie und mit welchen Mitteln das Projekt realisiert werden soll.

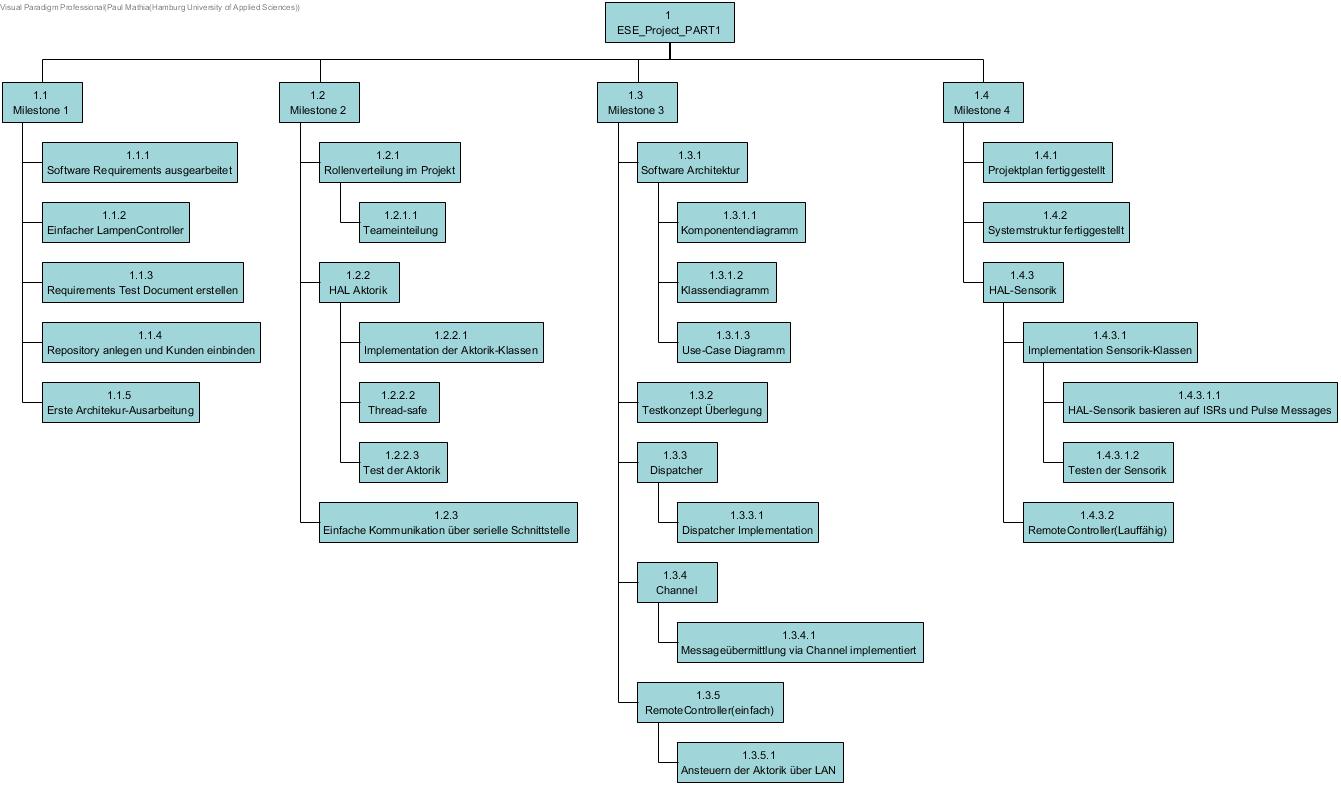
## Prozess

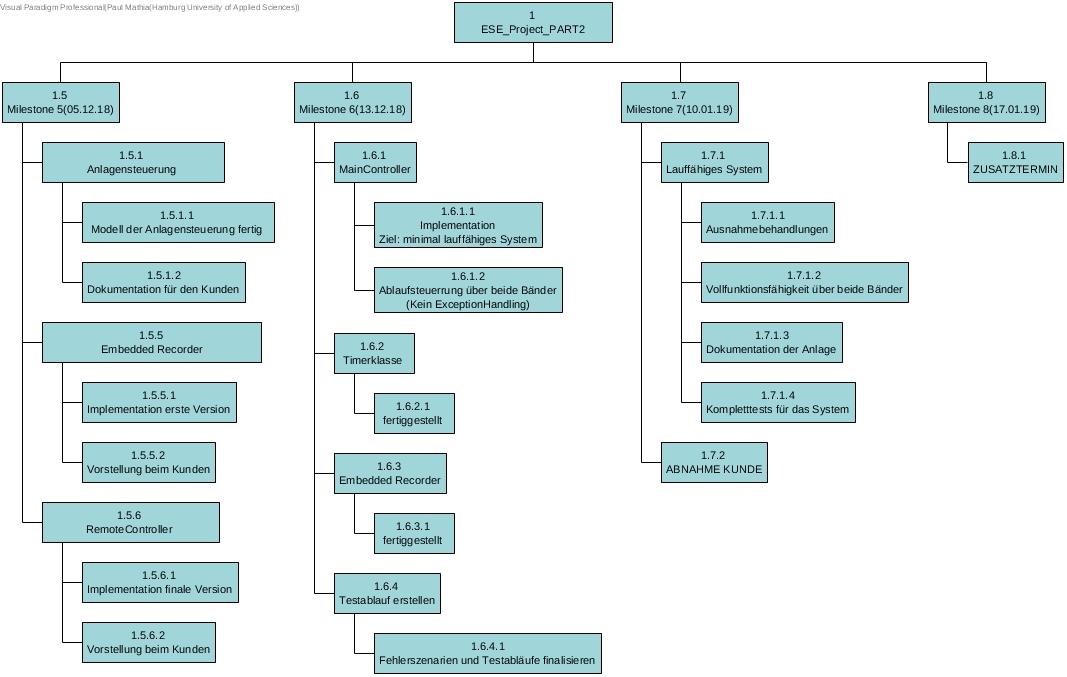
Bei Meetings werden einzelne Aufgaben an die vorher vorgestellten 2er-Gruppen verteilt. Beim nächsten Meeting werden die Ergebnisse präsentiert und diskutiert, wie fortgefahren werden soll, ggf. neue Aufgabenverteilung.

## PSP

Projektstrukturplan auf der folgenden Seite.

PSP wurde in zwei Teile geteilt um es übersichtlicher darzustellen.





## Qualitätssicherung

//TODO

# Randbedingungen

## Arbeitsumgebung

Im Rahmen des Projektes wird folgende Software / die folgenden Dienste verwendet:

* QNX Momentics (IDE)
* Google Test (Test Framework)
* Catch2(Test Framework)
* Doxygen (Dokumentation)
* Visual Paradigm (Design)
* Google Drive (RDT)
* HAW-internes gitlab (Version Control System)

## Programmiersprachen

Das Projekt wird in C++ umgesetzt (C++14 kompatibel).

# Requirements und Use Cases

Eine Auflistung aller Requirements des Systems.

## Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr** | **Typ** | **Beschreibung** | **Referenz** |
| REQ-001 | Sortierung flacher Werkstücke | Flache Werkstücke müssen auf die Rutsche von Band 1 aussortiert werden. | Seite 2 Liste Punkt 1 |
| REQ-002 | Sortierung von Werkstücken ohne Bohrung oben | Werkstücke, die auf der Oberseite keine Bohrung haben,  müssen auf die Rutsche von Band 1 aussortiert werden. | Seite 2 Liste Punkt 2 |
| REQ-003 | Reihung | Werkstücke mit Bohrung müssen am Ende von Band 2 in der Reihenfolge   **Bohrung oben ohne Metall →  Bohrung oben mit Metall →  Bohrung oben ohne Metall** ankommen. Werkstücke, die dieser Reihung nicht entsprechen, müssen auf die Rutsche von Band 2 aussortiert werden. | Seite 2 Satz 4 |
| REQ-004 | Codierte Werkstücke | Codierte Werkstücke mit dem Code "1" und "4" müssen auf die Rutsche von Band 1 aussortiert werden.  Codierte Werkstücke mit dem Code "2" und "5" müssen auf die Rutsche von Band 1 aussortiert werden. | Seite 3 Liste Punkt 4/5 |
| REQ-005 | Kalibrierung | Aufgrund von inkonsistentem Verhalten (Bandgeschwindigkeit, Position der Sensoren) muss eine Kalibrierung des Systems durchgeführt werden, um ein konsistentes Verhalten des Systems zu bewerkstelligen, unabhängig davon, auf welchen konkreten Geräten das System zu einem beliebig gegebenen Zeitpunkt operiert. Während der Kalibrierung muss die Ampel Grün blinken. | Seite 3 Punkt A |
| REQ-006 | Last von Band 1 | Werkstücke dürfen stets an den Anfang von Band 1 gelegt werden, wenn der Anfang frei ist. Das bedeutet, dass das System damit klar kommen muss, wenn sich mehrere Werkstücke auf einmal auf Band 1 befinden. Der des Bandes ist dann frei, wenn die Lichtschranke nicht mehr durch ein anderes Werkstück unterbrochen wird. | Seite 3 Punkt C |
| REQ-007 | Übergabe an Band 2 | Es darf nur ein Werkstück von Band 1 auf Band 2 übertragen werden, wenn Band 2 frei ist. Band 2 gilt erst dann als frei, wenn der Benutzer das Werkstück am ende von Band 2 entfernt oder ein Werkstück über die Rutsche aussortiert wird. | Seite 3 Punkt D |
| REQ-008 | Übergabe an Band 1 | Das Werkstück muss an den Anfang von Band 1 gelegt werden. Damit das System den Sortiervorgang starten kann, muss das Werkstück so platziert werden, dass es die erste Lichtschranke durchbricht. |  |
| REQ-009 | Höhenmessung | Während der Höhenmessung muss das Band langsam laufen. | Seite 3 Punkt F |
| REQ-010 | Herunterfallen | Es darf kein Werkstück von den Bändern fallen. Das gilt auch, wenn das Werkstück das Ende von Band 2 erreicht. | Seite 3 Punkt G |
| REQ-011 | Rutschenlast | Ist die Rutsche auf Band 1 voll, so muss die Aussortierung über Band 2 erfolgen. Ist die Rutsche auf Band 2 voll, so muss die Aussortierung bereits auf Band 1 erfolgen. Diese Situation muss dem Bediener zu signalisiert werden. Die Bedingung an die Reihenfolge am Ende von Band 2 (REQ-003) muss nach wie vor eingehalten werden. | Seite 3 Punkt H |
| REQ-012 | Leerlauf | Leerlauf muss vermieden werden. Wenn sich auf einem einzelnen Band kein Werkstück befindet, muss es stoppen. | Seite 3 Punkt I |
| REQ-013 | Typkennung | Die Typkennung eines codierten Werkstücks muss auf der Konsole ausgegeben werden, sobald diese erkannt wird. | Seite 3 Punkt J |
| REQ-014 | Werkstücksdaten | Wenn ein Werkstück das Ende von Band 2 erreicht, müssen auf der Konsole folgende Werkstücksdaten ausgegeben werden: – ID – Typ – Höhen-Messwert von Band 1 – Höhen-Messwert von Band 2 Die ID muss das System beim Erkennen des Werkstücks am Anfang von Band 1 vergeben. | Seite 3 Punkt K |
| REQ-015 | Weichensteuerung | Die Weiche darf nicht minuten- oder stundenlang auf Durchgang gestellt werden. | Seite 4 Erster Absatz |
| REQ-016 | Fehlererfassung | Folgende Fehlerzustände beim Betrieb der Anlage müssen erfasst werden: • Verschwinden von Werkstücken Reaktion: Bandstopp, Fehlermeldung • Hinzufügen von Werkstücken mitten auf dem BandReaktion: Bandstopp, Fehlermeldung • Beide Rutschen voll Reaktion: Bandstopp, Fehlermeldung | Seite 4 Zweiter Absatz |
| REQ-017 | Bedientaster | Die Bedientaster müssen folgende Funktionen erfüllen: • Start: Die Anlage wechselt in den Betriebszustand. Wird der Taster 3 Sekunden lang durchgehend gedrückt, so hat das System in einen Service-Mode überzugehen, in dem Selbsttests und Kalibrierungen durchgeführt werden. • Stop: Die Anlage wechselt wieder in den Ruhezustand. Der Wechsel darf nur möglich sein, wenn keine Fehler oder Warnungen vorliegen. • Reset Fehlerquittierung (siehe REQ-018). • E-Stopp Schnellabschaltung. Wird der E-Stopp-Taster gedrückt, hat die ganze Anlage stillzustehen. Es muss also auch der andere Anlagenteil stillstehen, an dessen Bedienpanel der E-Stopp-Taster nicht gedrückt wurde. Wenn der Taster anschließend wieder herausgezogen wird, muss die Anlage weiterhin so lange stehen, bis der Reset- Taster gedrückt wird. Dem Benutzer müssen Hinweise zur Bedienung der Anlage über die LEDs an den Tastern gegeben werden. | Seite 4 Dritter Absatz |
| REQ-018 | Zustandsanzeigen | Die farbigen Anzeigeleuchten müssen folgende Zustände signalisieren:   * **Grün:** Bandanlage in Betrieb. Dauerleuchten bei Normalbetrieb. Blinklicht beim Kalibrierungsvorgang. * **Gelb:** Blinklicht beim Aussortieren der Werkstücke und anderen Warnungen. * **Rot:** Bei Fehlermeldesystemen müssen 4 verschiedene Zustände signalisiert werden. Wenn ein Fehler neu aufgetreten ist, hat er den Zustand **anstehend unquittiert** (schnelles Blinken 1Hz). Quittiert der Bediener den Fehler durch das drücken der Quittiertaste, muss der Fehler in den Zustand **anstehend quittiert** (Dauerlicht)wechseln. Wird der Fehler behoben, muss der Fehler in den Zustand **OK** (Licht aus)wechseln. Wenn sich der Fehler von selbst repariert, ohne dass der Bediener ihn zu Kenntnis genommen hat, muss der Fehler in den Zustand **gegangen unquittiert** (langsames Blinken 0,5Hz)wechseln. | Seite 5 |
| REQ-019 | Übergabe an Band 1 | Das Werkstück muss an den Anfang von Band 1 gelegt werden. Damit das System den Sortiervorgang starten kann, muss das Werkstück so platziert werden, dass es die erste Lichtschranke durchbricht. |  |
| REQ-020 | Überschlagung | Das System muss erkennen, wenn sich ein Werkstück bei der Übergabe von Band 1 auf Band 2 überschlägt und muss mit dieser Situation entsprechend umgehen können. (Gelbes Licht blinkt, Aussortieren) | Seite 3 Punkt E |
| REQ-021 | ID-Vergabe | Wird die erste Lichtschranke durchbrochen, muss das Werkstück eine ID zugeteilt bekommen. Die ID muss eindeutig sein, das heißt eine ID darf nicht an mehrere Werkstücke verteilt werden. |  |
| REQ-022 | Verschwinden und Hinzufügen von Werkstücken | Wenn Werkstücke auf einem beliebigen Band unerwartet verschwinden oder auftauchen, muss das System damit entsprechend umgehen. (Frage im Praktikum)(Warum gab es nicht bereits ein Requirement dafür?) |  |
| REQ-023 | Rutschen Leeren | Wenn beide Rutschen voll sind, muss das System dies als Fehler erkennen und entsprechend in den Fehlermodus übergehen. Die Rutschen müssen dann manuell von Hand geleert werden und der Fehler vom Nutzer anschließend quittiert werden. |  |
| REQ-024 | Remote Controller | Das System wird über eine Remote-PC überwacht und gesteuert.  Der aktuelle Zustand, mögliche Fehlerzustände der Anlage und weitere Informationen wie z.B Werkstücks-Informationen werden auf einer GUI-Applikation/Konsolen-Applikation angezeigt. Über die GUI-Applikation kann das Personal die Aktorik der Anlage steuern und überwachen. Umschalten auf die  Manuelle Steuerung überschreibt das automatisierte Sortierverfahren der Anlage. Das umschalten zurück in den automatisierten Modus passiert nach schließen der Applikation. |  |
| REQ-025 | Embedded Recorder | Aufgabe des Embedded Recorders ist es nach Systemstart alle Ereignisse der Anlage aufzuzeichnen. Alle Ereignisse werden dabei in einer txt-file gespeichert und können anschließend vom Anwender eingelesen und über eine Replay-Funktion rekonstruiert werden. Hauptfunktion des Embedded Recorders ist es also sich auch ohne Werkstücke anhand der Ereignisdaten genauso zu verhalten wie zum Zeitpunkt der der Datenaufnahme. |  |

## Use Cases

|  |
| --- |
| Use Case Nr: 01 |
| Name | E-Stopp-Taste drücken |
| Autor | TK |
| Priorität |  |
| Requirements | REQ-017, REQ-018 |
| Auslöser | Der Anwender drückt die E-Stopp-Taste |
| Akteure | Anwender |
| Vorbedingung | Das System ist in Betrieb |
| Beschreibung | Das System befindet sich im Normalbetrieb und die E-Stopp-Taste wird gedrückt. |
| Ergebnis | Die gesamte Anlage steht still, das heißt auch das Laufband an dem die E-Stopp-Taste nicht gedrückt wurde. Das System steht so lange still bis die Taste wieder herausgezogen wurde und die Reset-Taste gedrückt wurde. Dem Nutzer werden Hinweise zur Bedienung über die LED’s an den Tasten angezeigt(z.B. Blinken der Reset-Taste). |
| Hauptszenario | 1. Der Anwender drückt die E-Stopp-Taste an einem der Laufbänder 2. Beide Anlagenteile stehen komplett still 3. Die E-Stopp-Taste wird wieder herausgezogen 4. LED hinter der Reset-Taste fängt an zu blinken 5. Die Reset-Taste wird gedrückt 6. Das System kehrt in den Normalbetrieb zurück |
| Alternativszenarien |  |
| Fehlerszenarien | 1. Nur eines der Laufbänder bleibt beim Drücken der E-Taste stehen 2. Das System bleibt beim Drücken der E-Stopp-Taste im Normalbetrieb 3. Das System kehrt nach dem Drücken der Reset-Taste nicht in den Normalbetrieb zurück |

|  |
| --- |
| Use Case Nr: 02 |
| Name | Unerlaubtes Entfernen eines Werkstückes |
| Autor | TK |
| Priorität |  |
| Requirements | REQ-016, REQ-022 |
| Auslöser | Ein Werkstück wird während des Betriebs vom Laufband entfernt |
| Akteure | Anwender |
| Vorbedingung | Das System befindet sich im Normalbetrieb, ein Werkstück wurde am Anfang des Laufbandes aufgelegt und befindet sich auf einem der Laufbänder. Das Werkstück hat das Ende des letzten Laufbandes noch nicht erreicht. |
| Beschreibung | Ein Werkstück wird, während es sich auf dem Laufband befindet und bevor es das Ende des letzten Laufbandes erreicht hat, vom Nutzer entfernt. |
| Ergebnis | Die Laufbänder bleiben stehen. Es wird eine Fehlermeldung angezeigt. |
| Hauptszenario | 1. Der Nutzer legt ein Werkstück an den Anfang des ersten Bandes 2. Der Nutzer entfernt das Werkstück bevor es das Ende des zweiten Bandes erreicht hat 3. Die Laufbänder bleiben stehen 4. Eine Fehlermeldung wird ausgegeben(?) |
| Alternativszenarien |  |
| Fehlerszenarien | 1. Die Laufbänder stoppen nicht nachdem das Fehlen des Werkstückes erkannt wurde 2. Das Fehlen des Werkstückes wird nicht erkannt 3. Es wird keine Fehlermeldung ausgegeben |

|  |  |
| --- | --- |
| Use Case Nr: 03 |  |
| Name | Fehlerquittierung |
| Autor | AI |
| Priorität |  |
| Requirements | REQ-018 |
| Auslöser | Fehlersignal eingegangen |
| Akteure | System, Anwender |
| Vorbedingung | System befindet sich im Normalbetrieb |
| Ergebnis | Der Fehler wird behoben und das System befindet sich wieder im Normalbetrieb zurück |
| Hauptszenario | 1. Fehlersignal wird von System ausgelöst. 2. Das System befindet sich im Zustand „anstehend unquittiert“. Die rote Lampe leuchtet mit 1Hz Geschwindigkeit. 3. Der Bediener drückt die Quittierungstaste. Das System geht in den Zustand „anstehend quittiert“. Die rote Lampe leuchtet dauerhaft. 4. Der Fehler wird vom Bediener behoben. Das System geht in Zustand „OK“. Die rote Lampe leuchtet aus. |
| Alternativszenarien | 1. Fehlersignal wird von System ausgelöst. 2. Das System befindet sich im Zustand „anstehend unquittiert“. Die rote Lampe leuchtet mit 1 GHz Geschwindigkeit. 3. Das Problem wird vom System selbst behoben. Das System befindet sich jetzt im Zustand „gegangen unquittiert“. Die rote Lampe leuchtet mit 0,5 GHz Geschwindigkeit. 4. Der Bediener drückt die Quittierungstaste. Das System geht in Zustand „OK“. Die Lampe leuchtet aus. |
| Fehlerszenarien | Das System bleibt im Fehlerzustand und kann nicht mit dem Betrieb weitermachen. |

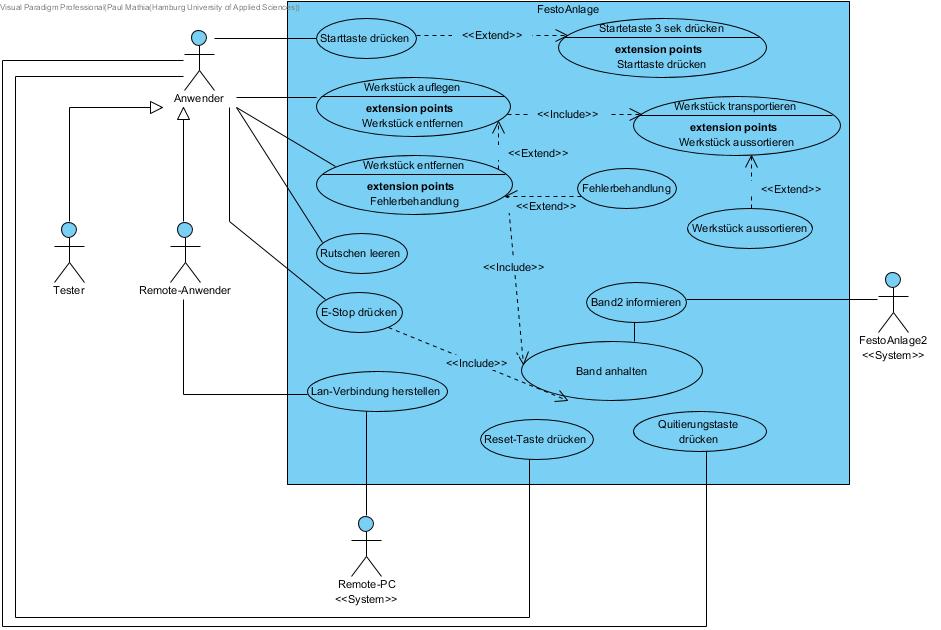
|  |
| --- |
| Use Case Nr: 04 |
| Name | Kalibrierung |
| Autor | BGV |
| Priorität |  |
| Requirements | REQ-017, REQ-005 |
| Auslöser | Die Start-Taste wird 3 Sekunden lang gedrückt. |
| Akteure | System, Anwender |
| Vorbedingung | System befindet sich im Ruhezustand. (siehe Glossar) |
| Ergebnis | Das System hat die Anlagen neu kalibriert. |
| Hauptszenario | 1. Der Anwender drückt lange die Start-Taste eines Bandes 3 Sekunden lang gedrückt. 2. Das System wechselt in den Service-Mode. 3. Die grünen Lampen beider Bänder blinken mit 1 Hz. 4. Das System führt Kalibrierung durch. 5. Die grünen Lichter beider Bänder hören auf zu blinken und sind ausgeschaltet. 6. Das System kehrt in Ruhestand zurück. |
| Alternativszenarien |  |
| Fehlerszenarien | Bei erfolgreicher Kalibrierung gibt das System eine Rückmeldung über die Konsole aus. Bei einer fehlerhaften oder abgebrochenen Kalibrierung wird die Meldung auf der Konsole ausgegeben. Es folgt ein erneuter Kalibrierung Versuch. |

|  |
| --- |
| Use Case Nr: 05 |
| Name | Rutschen leeren |
| Autor | BGV |
| Priorität |  |
| Requirements | REQ-023 |
| Auslöser | Beide Rutschen voll. |
| Akteure | System, Anwender |
| Vorbedingung | Das System befindet sich im Betriebszustand. |
| Ergebnis | Die Rutschen sind leer und das System nimmt den Betriebszustand wieder auf. |
| Hauptszenario | 1. Das System stellt fest, dass beide Rutschen voll sind. 2. Beide Bänder stoppen. 3. Die rote Lampe blinkt schnell mit 1Hz (anstehend unquittiert) 4. Der Anwender leert beide Rutschen und drückt anschließend auf Reset-Taste. 5. Das System signalisiert den Wert OK. 6. Die rote Lampe ist aus. 7. Das System nimmt den betriebszustand wieder auf. |
| Alternativszenarien |  |
| Fehlerszenarien |  |

|  |
| --- |
| Use Case Nr: 06 |
| Name | Sortierung von Werkstück Typen |
| Autor | ACA |
| Priorität |  |
| Requirements | REQ-001, REQ-002, REQ-003, REQ-004 |
| Auslöser | Ein Werkstück wird zum Sortieren auf das Band gelegt |
| Akteure | System, Anwender |
| Vorbedingung | Das System befindet sich im Anfangszustand // ist betriebsbereit |
| Ergebnis | Werkstücke wurden korrekt aussortiert. |
| Werkstücktypen | Im folgenden werden folgende Werkstücktypen definiert:   |  |  | | --- | --- | | Typ 1 | Flaches Werkstück | | Typ 2.1 | Werkstück mit Bohrung und Metalleinsatz | | Typ 2.2 | Typ 2.1 auf dem Kopf | | Typ 3.1 | Werkstück mit Bohrung ohne Metalleinsatz | | Typ 3.2 | Typ 3.1 auf dem Kopf | | Typ 4.0 bis 4.7 | Codierte Werkstücke, jeweils mit ihrer binären Codierung. (4.2 hat zum Beispiel die Codierung  „0-1-0“ | |
| Hauptszenario | 1. Das Werkstück wird auf das Band gelegt, sodass es die Lichtschranke unterbricht. 2. Band läuft los 3. Werkstück kommt in die Höhenmessung von Band 1 4. Werkstück kommt in den Metalldetektor von Band 1 5. →Sortierszenario |
| Sortierszenario Typ 1, Typ 2.2, Typ 3.2, Typ 4.1, Typ 4.4 | 1. Das Werkstück wird auf die Rutsche von Band 1 aussortiert. |
| Sortierszenario Typ 4.2, Typ 4.5 | 1. Das Werkstück wird an Band 2 übergeben. 2. Das Werkstück wird auf die Rutsche von Band 2 aussortiert. |
| Sortierszenario Typ 2.1, Typ 3.1 | 6. Das Werkstück wird an Band 2 übergeben.  7.1.1 Wenn das Werkstück der Reihung gemäß REQ-003 entspricht, dann wird das Werkstück bis an das Ende von Band 2 gefahren.  7.1.2 Band stoppt  7.1.3 Werkstück wird manuell entfernt.  7.2.1 Wenn das Werkstück der Reihung gemäß REQ-003 nicht entspricht, so wird das Werkstück auf die Rutsche von Band 2 aussortiert. |
| Fehlerszenarien | |  | | --- | | 6. Das Werkstück wird auf Band 2 übergeben.  6.1 Das Werkstück überschlägt sich.  6.2 Das Werkstück wird auf die Rutsche von Band 2 aussortiert. | | 1. Das Werkstück wird auf das Band gelegt, sodass es die Lichtschranke unterbricht. 2. Band läuft los.   2.1Werkstück verschwindet.  2.2 Fehlermeldung (am Band, an dem es verschwunden ist) | |

|  |
| --- |
| Use Case Nr: 07 |
| Name | Manuelle Steuerung über den Remote PC |
| Autor | AI |
| Priorität |  |
| Requirements | REQ-024 |
| Auslöser | Manueller Steuerungsmodus auf Remote PC |
| Akteure | Anwender, Remote PC, Anlage |
| Vorbedingung | Das System befindet sich im Sortiermodus oder im Ruhezustand. |
| Ergebnis | Die Aktorik der Anlage kann manuell vom Remote PC gesteuert werden. |
| Hauptszenario | 1. Der Anwender öffnet/startet die GUI-Applikation/Konsole auf dem eingerichteten Remote PC und verbindet sich mit der Anlage. 2. Der Anwender stellt die Anlage in den manuellen Steuerungsmodus um. 3. Der Anwender nutzt die GUI-Applikation/Konsole um die Aktorik zu steuern. |
| Alternativszenarien |  |
| Fehlerszenarien | 1. Der Anwender öffnet die GUI-Applikation auf dem Remote PC und verbindet sich mit der Anlage. 2. Die Verbindung zwischen Remote PC und der Anlage ist während das manuelle Steuerungsmodus abgebrochen. |

|  |
| --- |
| Use Case Nr: 08 |
| Name | Überwachung der Anlage vom Remote PC |
| Autor | AI |
| Priorität |  |
| Requirements | REQ-024 |
| Auslöser | Verbindung vom Remote PC und Anlage erstellt |
| Akteure | Anwender, Remote PC, Anlage |
| Vorbedingung | Der Remote PC und die Anlage sind durch LAN verbunden. |
| Ergebnis | Die (Fehler)zustände der Anlage, die Informationen über die Werkstücke und die aktuellen Ereignisse auf dem Laufband werden auf dem Remote PC angezeigt. |
| Hauptszenario | 1. Anwender startet die GUI-Applikation/Konsolen-Applikation auf dem Remote PC und verbindet sich mit der Anlage. 2. Die Anlage sendet ihre aktuellen Ereignisse in Form von Log-Informationen an den Remote PC. 3. Die empfangenen Informationen werden auf der GUI vom Remote PC angezeigt. |
| Alternativszenarien |  |
| Fehlerszenarien | 1. Anwender startet die GUI-Applikation/Konsolen-Applikation auf dem Remote PC und verbindet sich mit der Anlage. 2. Die Verbindung zwischen dem Remote PC und der Anlage wurde während der Überwachung abgebrochen. |

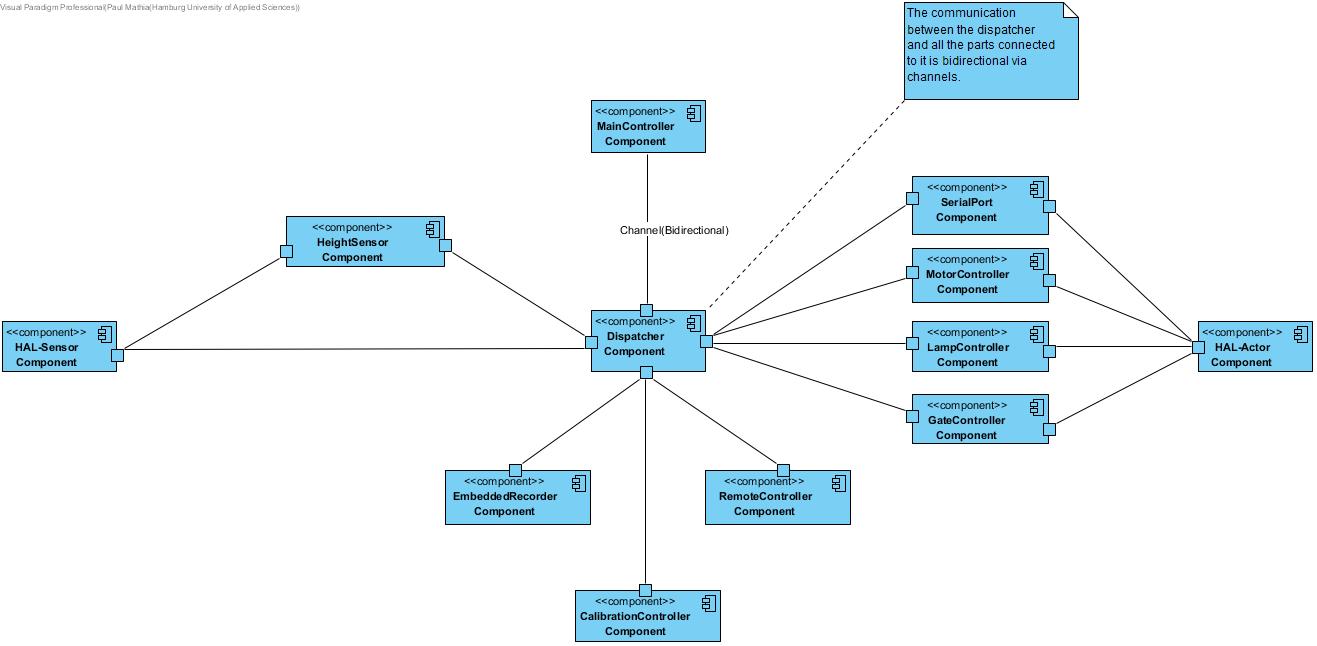


# Design

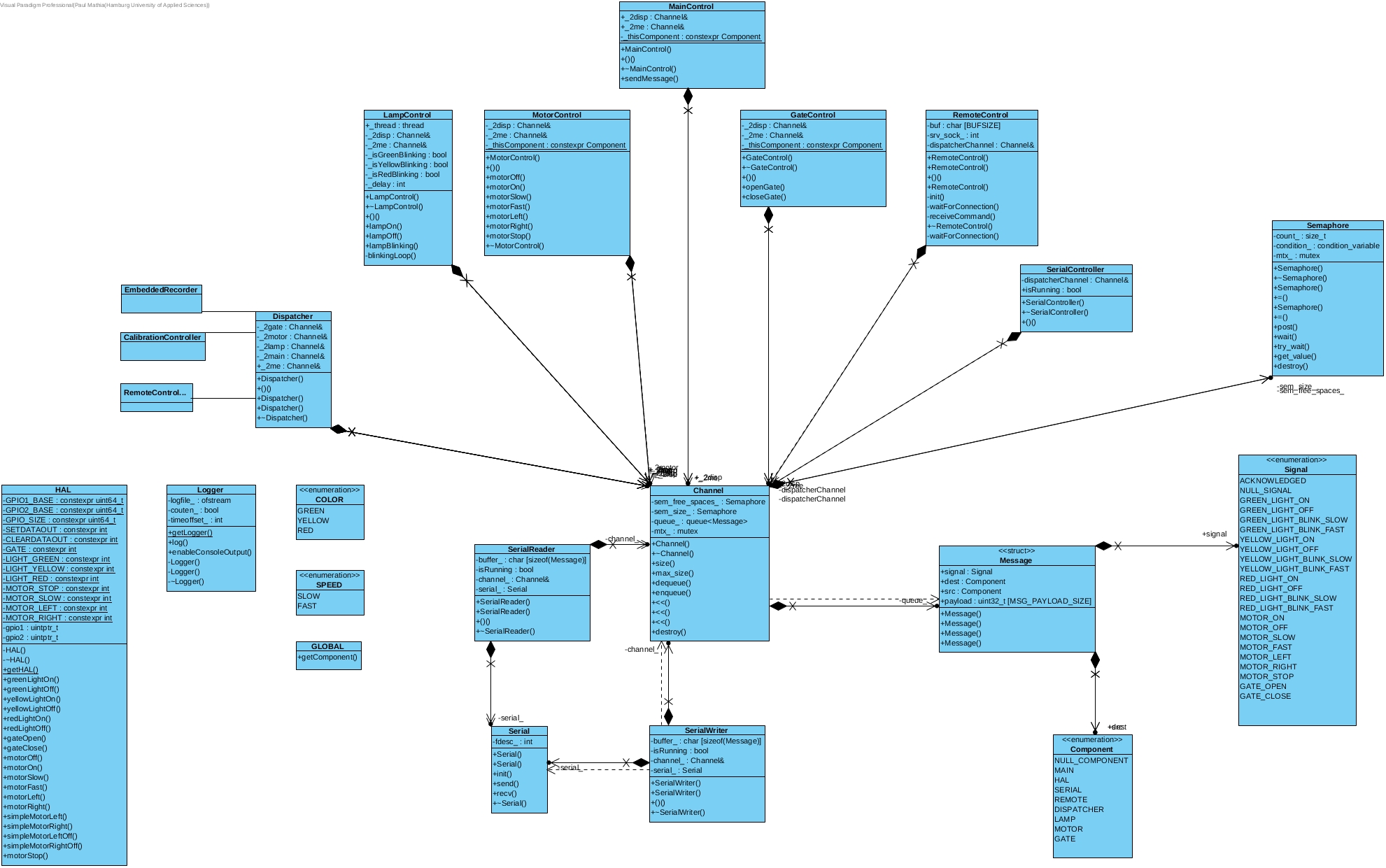
## Auflistung und Beschreibung der Komponenten

|  |  |
| --- | --- |
| **Komponent** | **Beschreibung** |
| MainController | Logik-Kern der Software (implementiert als State-Machine). Steuert das komplette System. |
| Dispatcher | Mittelsmann des Systems. Verteilt anhand der Messages des MainControllers die Aufgaben an die Controller. |
| Aktorik-/Sensorik Controller | Steuern die einzelnen Aktor und Sensor-Komponenten des Systems.  **LampController**: verantwortlich für das ein- und ausschalten der Lampen.  MotorController: verantwortlich für die Steuerung des Laufbandes.  **GateController**: verantwortlich für das öffnen und schließen der Weiche.  **SerialPortController**: verantwortlich für das übermitteln von Messages an die zweite angeschlossene Anlage.  **HeightSensorController**: versorgt den MainController mit den nötigen Höhendaten eines Werkstückes.  **MetalSensorController**: versorgt den MainController mit den nötigen Daten ob ein Werkstück Metal enthält. |
| RemoteController | Überschreibt die automatisierte Sortierfunktion des Systems mit einer manuellen. Siehe Requirement: REQ-024 |
| HAL-Sensorik/Aktorik | Hardware Abstraction Layer(Hardwareabstraktionsschicht) für Sensorik und Aktorik. Zuständig für das ansteuern der Hardware. Übersetzung der “Software Sprache” in “Hardware Sprache”. |
| Embedded Recorder | Sämtliche Kommunikation die über den Dispatcher geht wird an den Embedded Recorder geschickt der diese in einer Textfile speichert die für die Replay-Funktion des Systems benötigt werden. Siehe Requirement: REQ-025 |
| CalibrationController | Software-Einheit die einmal nach Systemsstart und/oder nach Aufforderung des Nutzers die Systemdaten neu kalibriert. Systemdaten sind Daten, welche das System benötigt um richtig zu laufen wie z.B.: die Zeit die verstreichen darf wenn ein Werkstück die erste Lichtschranke durchbrochen hat. (Min- und Max-Zeit). *(Detailierte Spezifikation folgt.)* |

## Komponentendiagramm

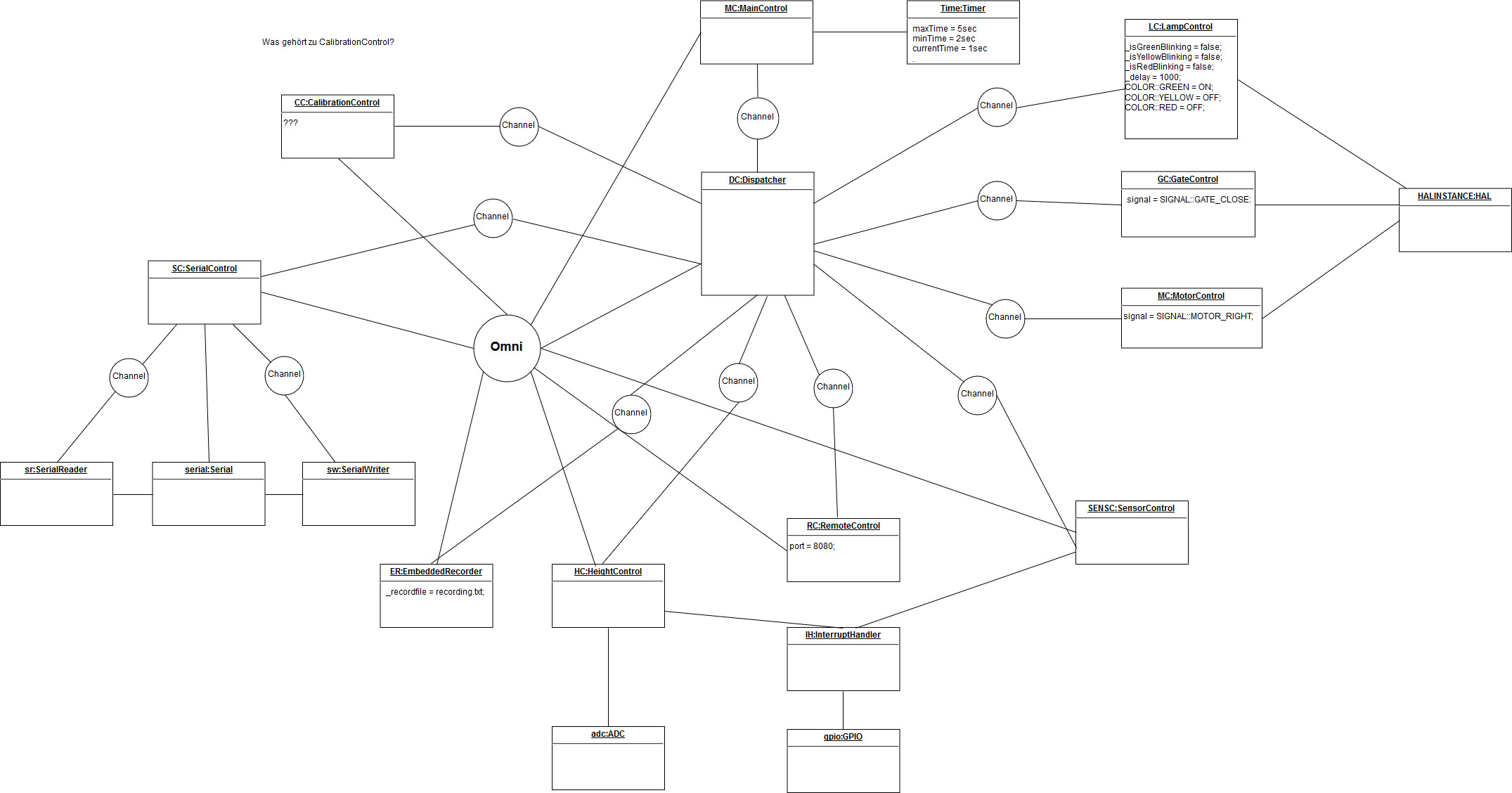


## Klassendiagramm(Wird im Projektverlauf ergänzt)



## Objektdiagramm

Das Objektdiagramm zeigt zur bestimmten Laufzeit, in welchem Zustand sich das System befindet. Dieses kann noch um einige Objekte erweitert werden. Siehe CalibrationControl.



## State Machines

siehe Bilder

# Anhang

## Glossar

|  |  |
| --- | --- |
| **Begriff** | **Erklärung** |
| Ruhezustand | Beide Bänder stehen still. Die Lampen sind aus. Zwischenschritt zwischen Betrieb und Aus. Das System weist keine Fehler nach. |
| Service-Mode | Das System führt Kalibrierung und/oder Selbsttest durch. |
| Kalibrierung | Alle Werte werden mit einem Initialwert versehen. |
| Anfangszustand, betriebsbereit | Gerät ist eingeschaltet und kalibriert, es liegen keine Fehlermeldungen vor, beide Bänder sind leer, beide Rutschen sind leer. |
| Normalbetrieb | Das System läuft wie in der Spezifikation beschrieben |
| Fehler | Ein Zustand des Systems, das nicht in der Spezifikation steht |
| Fehlersignal | Ein vom System ausgelösten Signal, das zur Erkenntnis alle beteiligten Komponenten, genutzt wird |
| Quittierungstaste | Reset Taste, gedrückt zur Erkennung, dass ein Fehler passiert ist. |
| Bediener | Ein Person, die für die Fehlerbehebung des Systems zuständig ist |
| GUI | Grafische Benutzeroberfläche oder auch grafische Benutzerschnittstelle (Abk. GUI von [englisch](https://de.wikipedia.org/wiki/Englische_Sprache) graphical user interface) bezeichnet eine Form von [Benutzerschnittstelle](https://de.wikipedia.org/wiki/Benutzerschnittstelle) eines [Computers](https://de.wikipedia.org/wiki/Computer). Sie hat die Aufgabe, [Anwendungssoftware](https://de.wikipedia.org/wiki/Anwendungssoftware) auf einem Rechner mittels grafischer [Symbole](https://de.wikipedia.org/wiki/Symbol), [Steuerelemente](https://de.wikipedia.org/wiki/Steuerelement) oder auch [Widgets](https://de.wikipedia.org/wiki/Widget) genannt, bedienbar zu machen. |

## Abkürzungen