

Aufgabe ESEP

Prof. Dr. Franz Korf
Prof. Dr. Thomas Lehmann
M. Sc. Enrico Christophers

Sommersemester 2023

Rev: 23SoSe-final

1. Ziel des Praktikums

In diesem Praktikum entwerfen Sie Steuerungs-Software im Team mit den aktuellen Methoden und Techniken der Softwareentwicklung für Eingebettete Systeme und führen ihren Entwurf in funktionierende Software-Produkt über. Hierdurch sollen Sie lernen, geeignete Methoden und Techniken der Softwareentwicklung auszuwählen, anzuwenden und ein funktionierendes Software-Produkt zu erstellen. Weiterhin soll die Koordinierung und das Arbeiten im Team geübt werden.

2. Aufgabenstellung

Bauen Sie aus zwei Förderbandmodulen (Festo-Transfersystem) eine **Werkstück-Sortieranlage**. Jedes Förderbandmodul wird durch einen eigenen Steuerungscomputer (Embedded System) gesteuert. Die beiden Computer sind über Ethernet gekoppelt und können das Qnet Protokoll verwenden.

Für die Implementierung der Grundfunktionalität wird ein 4er-Team benötigt. Bei einem größeren Team sind, in Absprache mit der Betreuung, Zusatzfunktionalitäten zu realisieren.

3. Problembeschreibung/Anforderungen

Im Folgenden sind die zu erfüllenden Anforderungen aufgeführt. Zur besseren Referenzierung sind die Aussagen am Ende mit einer fortlaufenden Nummer versehen. Trotz des Umfangs dieses Dokumentes sind die Anforderungen *unterspezifiziert* oder vielleicht auch *widersprüchlich*. Ein Teil Ihrer Tätigkeit besteht also darin, mit Ihrer Betreuung die Spezifikation zu *präzisieren*. Alle Absprachen sind schriftlich festzuhalten!

Ihnen steht eine Simulation der Anlage (Simulator) zur Verfügung (siehe Teams-Raum), der das Arbeiten Zuhause ermöglicht. Weitere Informationen über die Förderbänder und die Werkstücke finden Sie ebenfalls im Teams-Raum.

Die Endabnahme findet im Labor in Präsenz statt. Sollte es wieder pandemiebedingte Einschränkungen geben, so erfolgt alternativ die Abnahme online im Simulator. Letzteres wird rechtzeitig und gemeinsam im Praktikum festgelegt.

3.1. Grundfunktionalität Sortieren

Die Sortieranlage soll Werkstücke so sortieren, dass Werkstücke mit bestimmten Eigenschaften in die Rutschen aussortiert werden und am Ende von Förderbandmodul 2 Werkstücke in einer vorgegebenen Reihenfolge ankommen₍₁₎. Es gibt dabei mehrere Typen von Werkstücken, die auch unterschiedlich auf das Band des Förderbandmodul 1 gelegt werden können₍₂₎.

- Flache Werkstücke₍₃₎
- Hohe Werkstücke mit Bohrung und Metalleinsatz₍₄₎.
- Hohe Werkstücke mit Bohrung ohne Metalleinsatz₍₅₎.
- Hohe Werkstücke ohne Bohrung (dreht man ein hohes Werkstück mit Bohrung um, so erhält man ein Werkstück ohne Bohrung)₍₆₎.
- Binär-codierte Werkstücke₍₇₎.

Die Werkstücke haben unterschiedliche Farben: rot, schwarz und weiss₍₈₎. Entsprechend des Volumens und der Einarbeitung eines Metallrings haben sie auch unterschiedliches Gewicht₍₉₎. Codierte Werkstücke bestehen aus unterschiedlichem Material, sind grau oder semitransparent, je nach verwendetem Material für den 3D-Druck₍₁₀₎. Das Gewicht und Gleiteigenschaften variieren entsprechend₍₁₁₎.

Ziel der Sortierung ist es, dass am Ende von Förderbandmodul 2 die Werkstücke vereinzelt in der folgenden Reihenfolge ankommen:

- $\langle \text{Type A} \rangle \rightarrow \langle \text{Type B} \rangle \rightarrow \langle \text{Type C} \rangle$ ₍₁₂₎

Die Reihenfolge muss für nachfolgende Maschinen im Fertigungsprozess eingehalten werden, ansonsten können diese beschädigt werden₍₁₃₎. Für die Entnahme ist im Produktivbetrieb ein Pick-and-Place-Roboter vorgesehen₍₁₄₎. Die genauen Typen der Werkstücke für $\langle \text{Type A} \rangle$, $\langle \text{Type B} \rangle$ und $\langle \text{Type C} \rangle$ können aus der Menge der nicht binär-codierten Werkstücke konfiguriert werden₍₁₅₎.

Weiterhin gelten folgende Sortierregeln₍₁₆₎:

1. Flache Werkstücke sollen auf Förderbandmodul 1 erkannt und aussortiert werden₍₁₇₎.
2. Werkstücke, die nicht der gewünschten Reihung entsprechen, sollen spätestens auf Förderbandmodul 2 erkannt und aussortiert werden₍₁₈₎.

Weiterhin ist zu beachten:

- Die Zuführung erfolgt durch Einlegen des Werkstücks am Anfang des Bandes von Förderbandmodul 1₍₁₉₎. Die Lichtschranke wird hierdurch unterbrochen₍₂₀₎.
- Es dürfen stets Werkstücke auf das Band gelegt werden, wenn der Anfang von Förderbandmodul 1 frei ist₍₂₁₎. Auf diesem Förderbandmodul können sich also mehrere Werkstücke befinden₍₂₂₎.
- Die Übergabe an das Förderbandmodul 2 soll *vereinzelte* erfolgen. Es darf sich auf Förderbandmodul 2 maximal ein Werkstück auf dem Transportband befinden₍₂₃₎.
- Bei der Übergabe des Werkstücks von Förderbandmodul 1 auf Förderbandmodul 2 kann es passieren, dass die Werkstücke sich "überschlagen", d. h. eine oben liegende Bohrung liegt dann auf Förderbandmodul 2 unten₍₂₄₎.
- Auf beiden Bändern sollen die Werkstücke *langsam* durch die Höhenmessung transportiert werden₍₂₅₎.
- Es darf kein Werkstück von den Bändern fallen₍₂₆₎.
- Eine volle Rutsche ist an der entsprechenden Anlage zu signalisieren₍₂₇₎. Sind beide Rutschen voll, läuft der Sortierbetrieb solange weiter, bis eine Aussortierung eines Werkstückes nicht mehr erfolgen kann₍₂₈₎.
- Wenn ein Werkstück das Ende von Förderbandmodul 2 erreicht, sollen bis dahin folgende Daten von dem Werkstück erfasst sein₍₂₉₎:
 - Werkstück-ID₍₃₀₎
 - Typ₍₃₁₎
 - mittlere Höhe an Förderbandmodul 1 in Millimeter₍₃₂₎
 - Höhenmesswert an Förderbandmodul 2₍₃₃₎

Sollte sich das Werkstück "überschlagen" haben, so soll dieses ebenfalls erfasst werden₍₃₄₎.

Die Werkstück-ID vergibt Ihr System beim Erkennen des Werkstücks am Anfang von Förderbandmodul 1₍₃₅₎. Im Regelfall ist auch der Typ schon nach Durchlauf von Förderbandmodul 1 bekannt₍₃₆₎.

- Beide Bänder sollen jeweils stoppen, wenn sich kein Werkstück auf ihnen befindet₍₃₇₎.
- Ist die Rutsche auf Förderbandmodul 1 voll, so soll die Aussortierung über Förderbandmodul 2 erfolgen₍₃₈₎. Umgekehrt, ist die Rutsche auf Förderbandmodul 2 voll, so soll die Aussortierung bereits auf Förderbandmodul 1 erfolgen₍₃₉₎. Diese Situation ist dem Bediener zu signalisieren₍₄₀₎. Die Bedingung an die Reihenfolge am Ende von Förderbandmodul 2 muss weiterhin eingehalten werden₍₄₁₎.

- Insgesamt soll ein möglichst hoher Durchsatz an Werkstücken erreicht werden⁽⁴²⁾.
- Die Konfiguration von <Type A>, <Type B> und <Type C> soll in einer Datei auf dem System abgelegt sein und aus dieser eingelesen werden⁽⁴³⁾. Diese Konfigurationsdatei soll benutzerfreundlich gestaltet sein⁽⁴⁴⁾.

3.2. Besonderheiten der Anlagen

Ansteuerung der Sortiermechanik: Bezüglich der Umleitung von Werkstücken gibt es zwei Varianten von Förderbandmodulen⁽⁴⁵⁾. Die eine Variante verfügt über eine Weiche zum Aussortieren der Werkstücke, die zweite Variante über einen Auswerfer⁽⁴⁶⁾.

Die Weichen sind im stromlosen Zustand *geschlossen*⁽⁴⁷⁾. Beim Öffnen fließt Strom durch die Magnetspule, die die Weiche betätigt⁽⁴⁸⁾. Wenn der Strom zu lange fließt, überhitzt die Spule, und die Weiche wird beschädigt⁽⁴⁹⁾. **Daher bitte die Weichen *nicht minutenlang* auf Durchgang stellen**⁽⁵⁰⁾!

Der Auswerfer lässt im stromlosen Zustand Werkstücke passieren⁽⁵¹⁾. Fließt Strom durch den Auswerfer, so fährt dieser aus und drückt ein Werkstück aktiv in die Rutsche⁽⁵²⁾. Im ausgefahrenen Zustand können keine Werkstücke passieren und werden auch nicht in die Rutsche befördert⁽⁵³⁾.

Ihre Lösung soll mit beiden Varianten, Weiche und Auswerfer, arbeiten können⁽⁵⁴⁾. Die Abnahme erfolgt mit einer beliebigen Kombination an Anlagenvarianten⁽⁵⁵⁾.

Höhenmessung: Der Laser der Höhenmessung ist gegenüber der Senkrechten in Richtung Bandanfang verkippt⁽⁵⁶⁾. Die Verkipfung ist für das Funktionieren des Messverfahrens erforderlich (ansonsten Eigenblendung)⁽⁵⁷⁾. Die Lichtschranke an der Höhenmessung steht im Produktivsystem nicht zur Verfügung⁽⁵⁸⁾.

3.3. Fehlererfassung

Generell soll der Betrieb der Anlage sicher (Safety) sein⁽⁵⁹⁾.

Folgende Fehlerzustände beim Betrieb der Anlage sollen erfasst werden:

- Verschwinden von Werkstücken⁽⁶⁰⁾.
Reaktion: Bandstopp, Fehlermeldung
- Hinzufügen von Werkstücken außerhalb des Anfangsbereiches⁽⁶¹⁾.
Reaktion: Bandstopp, Fehlermeldung
- Beide Rutschen voll und ein notwendiges Aussortieren ist nicht mehr möglich⁽⁶²⁾.
Reaktion: Bandstopp, Fehlermeldung

Überlegen Sie selber ein Vorgehen, wie die Anlage nach Erkennung und Behebung eines Fehlers weitermachen soll⁽⁶³⁾. Falls eine Erweiterung der Fehlererfassung technisch sinnvoll ist oder wesentlich besser in das Lösungskonzept passt, so ist dieses in Absprache möglich⁽⁶⁴⁾.

Fehlermeldungen und Warnings sollen per MQTT gemeldet werden⁽⁶⁵⁾. Das Topic setzt sich aus dem String "Festo", dem letzten Teil der IP-Adresse der Anlage, sowie der Unterkategorie "incident" zusammen⁽⁶⁶⁾. Beispielsweise ist von der Anlage mit der IP

192.168.101.104 das Topic `/Festo/104/incident`₍₆₇₎. Die Nachricht ist ein JSON-codierter String.₍₆₈₎

3.4. Bedientaster

Mittels der Elemente des Bedienpannels sollen folgende Funktionalität realisiert werden₍₆₉₎:

- **Start** Anlage wechselt in den Betriebszustand₍₇₀₎. Wird der Taster lange gedrückt, kommt die Anlage in einen Service-Mode und führt beispielsweise Kalibrierungen oder Selbsttests durch₍₇₁₎.
- **Stop** Die Anlage wechselt wieder in den Ruhezustand₍₇₂₎. Der Wechsel ist nur möglich, wenn keine Fehler oder Warnungen vorliegen₍₇₃₎.
- **Reset** Fehlerquittierung (siehe unten)₍₇₄₎.
- **E-Stopp** Schnellabschaltung. Wird ein E-Stopp-Schalter gedrückt, steht die *ganze* Anlage still, ähnlich einem Not-Aus₍₇₅₎. Es steht also auch das Förderbandmodul still, an dessen Bedienpanel der E-Stopp-Schalter *nicht* gedrückt wurde₍₇₆₎. Wenn der Schalter anschließend wieder herausgezogen wird, bleibt die Anlage weiterhin so lange stehen, bis der Reset-Taster gedrückt wurde₍₇₇₎.

Geben Sie dem Benutzer sinnvolle Hinweise zur Bedienung der Anlage über die LEDs an den Tastern oder über andere Signalisierungsmöglichkeiten der Anlage.₍₇₈₎

3.5. Zustandsanzeigen

Die farbigen Anzeigeleuchten (Ampel) signalisieren mindestens folgende Anlagenzustände₍₇₉₎:

Grün Im Betriebszustand soll die Lampe dauerhaft leuchten₍₈₀₎. Im Service-Mode soll die Lampe blinken₍₈₁₎.

Gelb Blinklicht bei Warnungen₍₈₂₎.

Rot Fehler- und Quittierungszustand₍₈₃₎:

Bei Fehlermeldesystemen müssen unterschiedliche Zustände signalisiert werden₍₈₄₎. Wenn ein Fehler neu aufgetreten ist, hat dieser den Zustand *anstehend unquittiert*₍₈₅₎. Nun sieht ein Bediener den Fehler (der immer noch ansteht), und teilt dies dem Meldesystem durch Drücken der *Quittierungstaste* mit₍₈₆₎. Der Fehler wechselt in den Zustand *anstehend quittiert*₍₈₇₎. Wenn nun der Fehler behoben wird, der Fehler-Signaleingang (siehe unten) also den Wert *OK* hat, wechselt die Fehlersignalisierung wieder in den Zustand *OK*₍₈₈₎. Dieser Zustand wird nicht gesondert angezeigt₍₈₉₎. Weiterhin ist noch denkbar, dass sich der Fehler von selbst repariert, ohne dass das Bedienpersonal ihn zur Kenntnis genommen hat₍₉₀₎. Dies ist der Zustand *gegangen unquittiert*₍₉₁₎. Er wird durch quittieren verlassen₍₉₂₎.

Design-Hinweis: In dieser vollen Breite lässt sich die Fehlersignalisierung nur durchführen, wenn es einen allgemeinen *Signaleingang* gibt, der einen Fehler signalisiert₍₉₃₎.

Die Anzeige für aufgetauchte/verschwundene Werkstücke geht nach dem Drücken der Quittungstaste direkt in den Zustand *OK*, wenn kein weiterer Fehler vorliegt₍₉₄₎.

Die Fehlerzustände sollen folgendermaßen signalisiert werden₍₉₅₎:

kein Fehler Leuchte aus₍₉₆₎
anstehend unquittiert schnelles Blinken (1 Hz)₍₉₇₎
anstehend quittiert dauerhaft leuchten₍₉₈₎
gegangen unquittiert langsames Blinken (0,5 Hz)₍₉₉₎

4. Erweiterte Anforderungen

Ist ein Team größer als vier Studierende, so wird die zusätzliche Kapazität durch Zusatzfunktionalität kompensiert. Die folgenden Anforderungen stellen mögliche Erweiterungen dar. Die jeweils für das Team relevante Funktionalität ist mit der Betreuung abzusprechen. Auch hier gilt, alle Absprachen sind schriftlich festzuhalten. Die erweiterten Requirements sind spätestens zum 3. Praktikumstermin final dokumentiert.

4.1. Integrated Recorder/Replay

Alle Eingangssignale in ihre Software und alle Reaktionen sollen durch den Integrated Recorder aufgezeichnet werden₍₁₀₀₎. Die Abspeicherung soll für Menschen lesbar bzw. editierbar erfolgen₍₁₀₁₎. Hierdurch kann im Fehlerfall eine genaue Analyse der Signalreihenfolge durchgeführt werden₍₁₀₂₎.

Ihre Software soll dann durch ein Zusatzelement diese Aufzeichnung einlesen und die Eingangssignale in der Reihenfolge, und falls erforderlich dem korrekten Timing, wieder in Ihre Software einspeisen (Replay)₍₁₀₃₎. Hierdurch kann dann mittels Debugger das System beobachtet und Fehler gefunden werden₍₁₀₄₎.

Die aufgezeichneten Sequenzen sollen graphisch dargestellt werden₍₁₀₅₎.

Die Codierung der aufgezeichneten Sequenzen sollen so aufgebaut werden, dass auch Sequenzen von Hand erstellt werden können₍₁₀₆₎. Die Funktionalität der Software soll dann über den Replay-Mechanismus mit diesen künstlichen Sequenzen getestet werden₍₁₀₇₎. Hinweis: Setzen Sie das Recorder-System für das Testen ein. Es ist daher sinnvoll, das System frühzeitig im Projekt zu realisieren.

4.2. Remote Control

Die Sortieranlage soll wie auf einem Maschinenleitstand von einem remote PC gesteuert und überwacht werden₍₁₀₈₎. Eine genaue Vorstellung über solch eine Remote Control Anwendung hat der Kunde noch nicht, somit müssen Sie die Funktionalität mit ihm erarbeiten und dokumentieren₍₁₀₉₎.

Einige Punkte sind schon fest:

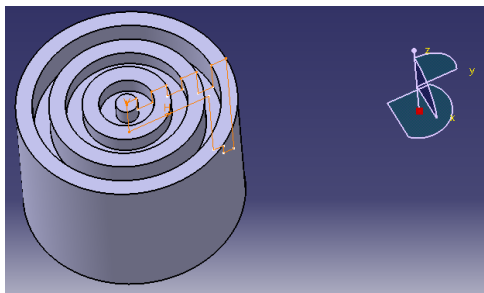
- Die Kommunikation soll auf MQTT basieren₍₁₁₀₎.
- Die Remote Control Anwendung soll zumindest eine graphische Oberfläche haben, die auf einem PC läuft₍₁₁₁₎. Eine graphische Oberfläche, die im Webbrowser läuft, ist für den Kunden optimal₍₁₁₂₎.
- Die Button-Panel beider Anlagen sollen in der Oberfläche sichtbar und benutzbar sein. Damit ist die Anlage über die Remote Control Anwendung zumindest ein- und ausschaltbar₍₁₁₃₎.

- Relevante Informationen, beispielsweise wie volle sind die Rutschen, sollen graphisch angezeigt werden₍₁₁₄₎.
- Die Konsolenmeldungen sollen auch in der Remote Control Anwendung ausgegeben werden₍₁₁₅₎.

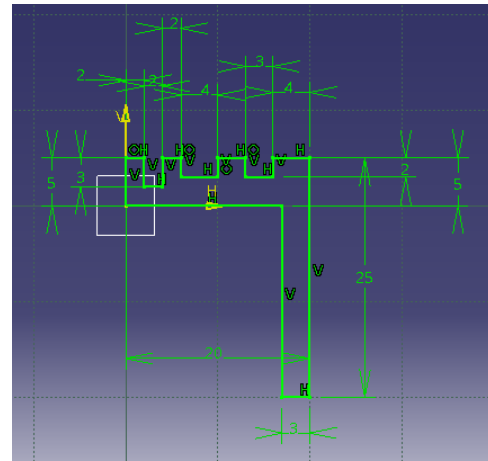
4.3. Codierte Werkstücke

Das Sortiment der Werkstücke wird um codierte Werkstücke erweitert₍₁₁₆₎. Diese sollen ebenfalls von der Anlage erkannt und aussortiert werden₍₁₁₇₎.

Die Ringe auf der Oberseite eines codierten Werkstücks stellen eine binär-codierte Typkennung dar (vgl. Abb. 1)₍₁₁₈₎. Ein mitteltiefer Ring stellt eine logische 1 dar, ein tiefer Ring eine logische 0₍₁₁₉₎. Ringe auf gleicher Höhe wie der Außenring können als Trennzeichen angesehen werden₍₁₂₀₎. Weitere Fotos und Beschreibungen dazu finden Sie im Teams-Raum.



(a) 3D-Ansicht



(b) Schnitt durch Werkstück

Abbildung 1: Prinzipieller Aufbau der codierten Werkstücke

Wird ein Werkstück mit einer binär-codierten Typkennung erkannt, so sind auf der Konsole der jeweiligen Anlage folgende Daten auszugeben₍₁₂₁₎:

- Werkstück-ID₍₁₂₂₎
- Binärcode₍₁₂₃₎

Für die binär-codierten Werkstücke gelten die gleichen Sortierregeln wie für die anderen Werkstücke₍₁₂₄₎.

4.4. Farberkennung

Mittels einer Kamera soll auf dem Förderbandmodul 1 an der Weiche die Farbe des Werkstücks erkannt werden₍₁₂₅₎. Die Farbe ist dann Teil der Werkstückdaten und hat keinen Einfluss auf die Sortierung₍₁₂₆₎.

Für die Farberkennung ist eine Anlage mit einer Kamerahalterung und Farbtafeln für die dynamische Farbwertkalibrierung ausgestattet₍₁₂₇₎. Als Kamerasysteme steht ein

Raspberry Pi mit Kamera zur Verfügung₍₁₂₈₎. Eine rudimentäre Farberkennung auf Basis von openCV ist bereits implementiert₍₁₂₉₎. Aufgabe ist, die Systeme zu integrieren und die Präzision der Farberkennung zu verbessern₍₁₃₀₎. Das Farberkennungssystem kann über UDP oder über MQTT angebunden werden₍₁₃₁₎.

4.5. Individuelle Erweiterung

Schlagen Sie dem Auftraggeber eine sinnvolle Erweiterung vor und stimmen Sie die nötigen Requirements ab₍₁₃₂₎.

5. Projektablauf

Zu den Praktikumsterminen soll der Projektfortschritt ihrer Betreuung demonstriert werden. Auch wenn Sie vermutlich nach einem agilen Prozess vorgehen, so sind zu bestimmten Terminen Mindestfunktionalitäten **am Beginn des Praktikumstermins** vorzuweisen. Weiterhin sind zu bestimmten Terminen zum Beginn des Praktikumstermins Dokumentationen abzuliefern. Wenn Sie hiervon abweichen wollen, so ist das mit der Betreuung rechtzeitig schriftlich zu vereinbaren.

Die Zeitaufwände der bearbeiteten Aufgaben und Tätigkeiten sollen erfasst und für das gesamte Teams zusammengefasst werden. Die Tracking-Daten sollen zum Beginn eines jeden Praktikumstermins aktuell sein.

5.1. Projektabnahme

Folgende Leistungen müssen für den erfolgreichen Abschluss erbracht werden:

- Die geforderte Funktionalität am Ende des Praktikums ist in ausreichendem Umfang umgesetzt. Die Implementierung wurde ausführlich getestet und die Tests protokolliert. Die Implementierung entspricht dem dokumentierten Design. Die Anlage lässt sich sicher betreiben.
- Die Dokumente lagen pünktlich vor und wurden abgenommen.
- Tracking-Daten über Schätzungen und geleisteten Aufwand lagen vor.

Das Praktikum kann durch die Betreuung ggf. vorzeitig abgebrochen werden, wenn der Projektfortschritt als nicht ausreichend angesehen wird. Es soll hierdurch verhindert werden, dass weiterhin Zeit investiert wird, die nicht zu einem erfolgreichen Abschluss führen wird.

5.2. Praktikumstermine

In diesem Abschnitt finden Sie Vorgaben und Empfehlungen für die Bearbeitung des Projektes über das Semester. Beachten Sie, dass die Zusatzaufgaben nicht aufgelistet sind. Diese müssen Sie bei der Projektplanung eigenständig mit berücksichtigen.

Wenn Sie die Reihenfolge der Tätigkeiten/Ablieferungen umstellen wollen, so ist dies mit dem Betreuer abzusprechen und schriftlich festzuhalten. Wir empfehlen, eine Abnahmeliste zu den jeweiligen Terminen zu erstellen und diese bei der Abnahme gemeinsam mit

dem Betreuer durchzugehen. Die Betreuer behalten sich vor, ihre eigene Abnahmeliste zu erstellen.

Gewährleisten Sie im Team, dass Sie Ihre Ergebnisse und den aktuellen Stand zu jedem Termin ad-hoc präsentieren können.

5.2.1. 1. Praktikumstermin

- Aus dem Workshop gegebenenfalls zu vervollständigen:
 - Sie haben ein Team gebildet.
 - Jedes Teammitglied hat eine verantwortliche Rolle innerhalb des Projekts bekommen.
 - Sie haben sich intensiv mit der Aufgabenstellung beschäftigt und haben die ersten System- und Anforderungsanalyse durchgeführt und dokumentiert.
 - Sie haben mit einem Projektplan angefangen und eine grobe Projektstruktur erstellt.
 - Sie haben sich mit eventuellen Zusatzaufgaben beschäftigt.
 - Sie können eine Anlage vom Beaglebone Black ansprechen (z.B. Ampel ansteuern).
- Sie haben ein Projekt in Momentics aufgesetzt.
- Die Source-Code-Verwaltung ist bei Github aufgesetzt. Dabei ist folgendes Namensschema zu verwenden:
`ESEP-2023SoSe-Team-<Gruppe>-<Team>`
Beispielsweise für das 2. Team der Praktikumsgruppe 03:
`ESEP-2023SoSe-Team-3-2`.
Ihre Betreuung hat Zugriff auf das Repository. Das gleiche Namensschema mit dem Postfix `__RDD` gilt für den Dateinamen für Ihr RDD.

5.2.2. 2. Praktikumstermin

- Die vollständige Anforderungsanalyse liegt als Dokument vor. Eventuelle individuelle Abmachungen mit dem Betreuer sind ausführlich zu dokumentieren. Die Analyse enthält eine Aufzählung von auftretenden Warnings und Errors.
- Die Aktorik kann über einen Hardware Abstraction Layer (HAL) angesteuert werden. Die Schnittstellen sind dokumentiert.
- Beispiel-Code für Qnet ist lauffähig. Ein Test prüft die korrekte Datenübertragung über Qnet.
- Erste Abnahmetests anhand der Systemanforderungen sind definiert. Mindestens ein Abnahmetest ist konkret, vollständig ausformuliert.
- Eine erste Software Architektur ist ausgearbeitet und dokumentiert. Sie diskutieren diese mit Ihrer Betreuung.

5.2.3. 3. Praktikumstermin

- **Pflicht:** Sie legen zu Beginn des Praktikumstermins ein Dokument mit einem Entwurf der Software-Architektur vor. Das Dokument soll erste Design-Elemente mit beinhalten. Das Dokument dient hier noch als Diskussionsgrundlage mit Ihrer Betreuung. Weiterhin enthält das Dokument Überlegungen zu den Qualitätssicherungsmaßnahmen im Projekt.
- Erste Modellierung der Anlagensteuerung beider Förderbänder mit mittels Zustandsautomaten inklusive Ausnahmebehandlung wurde durchgeführt.
- HAL der Sensorik ist entworfen, dokumentiert und implementiert.
- Ein Konzept für die Weiterleitung der Sensorsignale zu verarbeitenden Komponenten liegt vor.

5.2.4. 4. Praktikumstermin

- **Quality Gate:** Die Software-Architektur liegt dokumentiert vor. Die Architektur sollte soweit ausgereift sein, dass diese zügig umgesetzt werden kann. Das Dokument soll auch schon das Design der Steuerung beinhalten, die in den nächsten Wochen umgesetzt werden soll.

5.2.5. 5. Praktikumstermin

- **Quality Gate:** Eine Anlage sortiert bereits einige Werkstücke. Eine Fehlerbehandlung ist noch nicht implementiert.
- Die Modellierung ist weitgehend abgeschlossen.
- Grundfunktionalität ohne Fehlerbehandlung ist weitgehend auf beiden Anlagen implementiert.
- Meldungen zu Fehlern und Warnings können über MQTT verschickt werden.

5.2.6. 6. Praktikumstermin

- Die Modellierung ist vollständig abgeschlossen.
- Funktionalität ist weitgehend auf beiden Anlagen inklusive Fehlerbehandlung implementiert. Letzte Systemtests und Bug-Fixes müssen noch durchgeführt werden.

5.2.7. Praktikumstermin 6 $\frac{1}{2}$

- **Pflicht:** Sie reichen die finale Version des Requirement-Design-Documents bei Ihrer Betreuung ein.

5.2.8. 7. Praktikumstermin (Endabnahme)

An diesem Termin erfolgt die Endabnahme durch die Betreuer und es erfolgt eine Reflexion über das Projekt.

- Gesamtanlage ist bereit für die Abnahmetests durch den Kunden.
- Alle nicht realisierten Funktionalitäten sind dokumentiert und fehlende Umsetzung ist begründet.
- Bekannte Fehler sind dokumentiert.
- Alle Dokumente und der Projektplan sind auf aktuellem Stand.
- Lessons Learned dokumentiert.
- Abgabe aller Artefakten des Projekts (d.h. Dokumente, Plan, Code, Protokolle etc.).

A. Anhang

A.1. Port-Belegung

Die diskrete Sensorik kann über parallele GPIO-Ports des in der Anlage verbauten Beaglebone Black eingelesen werden. Ebenso kann die Aktorik über die GPIO-Ports angesteuert werden.

Festo Signale am Beaglebone Black

GPIO 0, folgende Bits sind Eingänge

Bit	Funktion
2	Werkstück im Einlauf = low when true
3	Werkstück in Höhenmessung = low when true
4	Werkstück Höhe OK = high when true
5	Werkstück in Weiche = low when true
7	Werkstück Metall = high when true
14	Weiche offen = high when true
15	Rutsche voll = low when true
20	Werkstück im Auslauf = low when true
22	TasteStart = high, wenn betätigt
23	TasteStop = low, wenn betätigt
26	TasteReset = high, wenn betätigt
27	E-Stop = low, wenn betätigt

GPIO 1, folgende Bits sind Ausgänge:

Bit	Funktion
12	Motor Rechtslauf
13	Motor Linkslauf
14	Motor langsam
15	Motor Stop
16	Rote Lampe an
17	Gelbe Lampe an
18	Grüne Lampe an
19	Weiche öffnen

GPIO 2, folgende Bits sind Ausgänge:

Bit	Funktion
2	LED Taste Start
3	LED Taste Reset
4	Signalleuchte Q1
5	Signalleuchte Q2