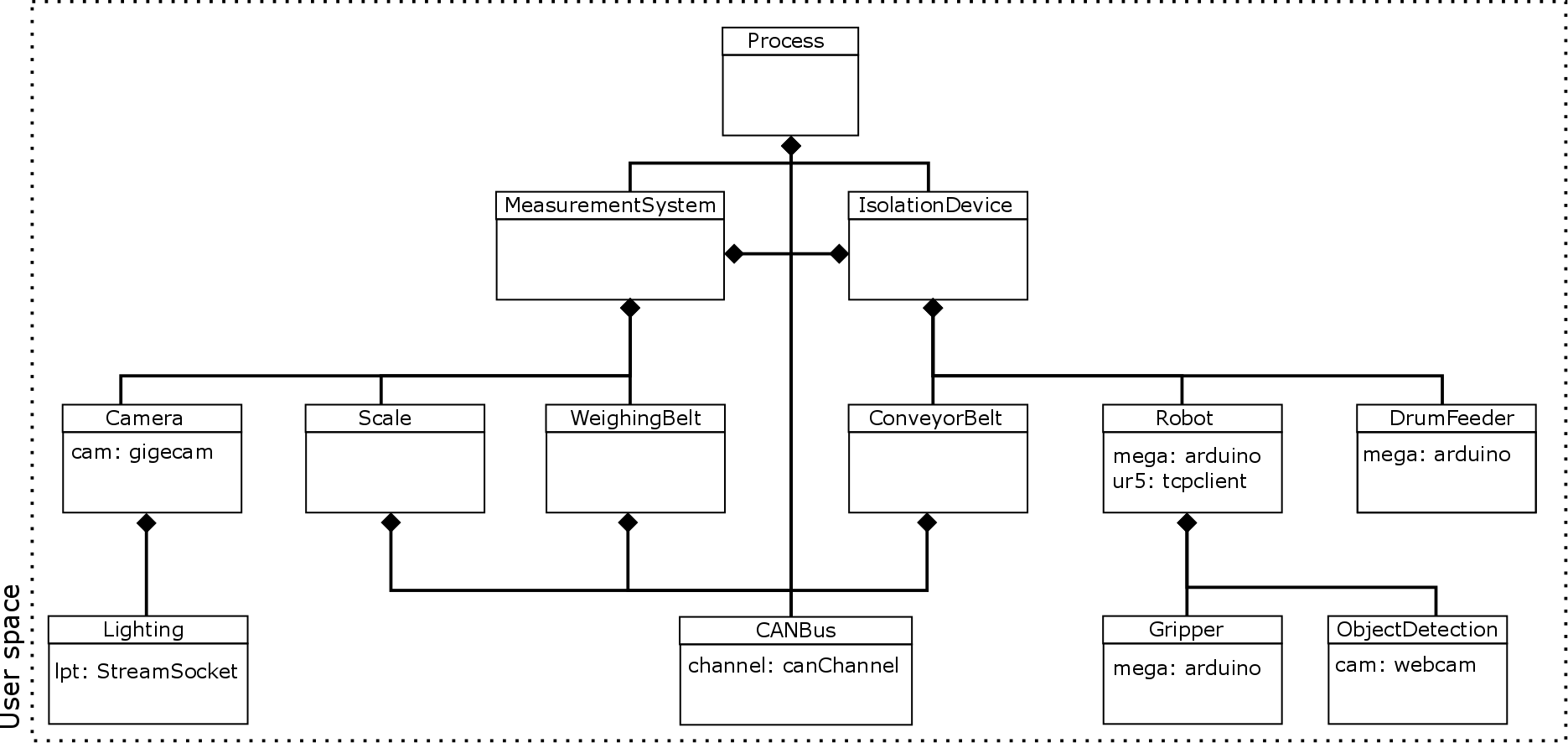
# Software

1. **Stand der Anlage**
   1. **Jede Komponente ein Objekt**
      1. **Baumstruktur**

****

* 1. **Kommunikationsschnittstellen**
     1. **CAN**
     2. **TCP**
     3. **Arduino**
     4. **LPT1**
     5. **Webcam**

1. **Zielsetzung**
   1. **Erweiterung**
      1. **Einfache Bedienung + Sichere Überwachung**
      2. **Gesamter Prozess + Einzelne Komponente**
   2. **Verbesserung**
      1. **Prozess**
2. **Einfache Bedienung + Sichere Überwachung → GUI**
   1. **Gesamter Prozess + Einzelne Komponente → Getrenntes Konzept**
   2. **Einfache Bedienung → Bisherige Funktionen durch Interaktive Elemente Abbilden u. erweitern**
      1. **View Model Controller Ansatz**
         1. **Allg. Erklärung**
         2. **Bisheriger Stand → Controller**
            1. **Funktionsaufrufe**
         3. **Bestehendes Objekt um Model erweitert**
            1. **Sowohl Lowlevel (Winkel etc.)**
            2. **Als auch Highlevel (State Object)**
         4. **GUI → View**
   3. **Sichere Überwachung**
      1. **Bisheriger Stand**

Das bisherige Überwachungskonzept stützt sich auf drei Methoden zur Bestimmung des Anlangenzustands.

* + - 1. **Interaktion mit den Objektrepräsentationen der Anlagenkomponenten**

Primär kann die Anlage über Iteraktion mit den Komponentenobjekten überwacht werden.

* + - 1. **Einfaches Textlogging**

Sekundär werden über ein rudimentäres Loggingsystem Zustandsnachrichten in der Matlabkonsole wiedergegeben. SIEHE ABBILDUNG IRGENDWAS

* + - * 1. **Nachteile**

**Nicht Filterbar**

**Nicht Persistent**

* + - 1. **Visuelle Überwachung**

Tertiär wird die Anlage durch den Anlagenbenutzer visuell überwacht.

* + - * 1. **Nachteile**

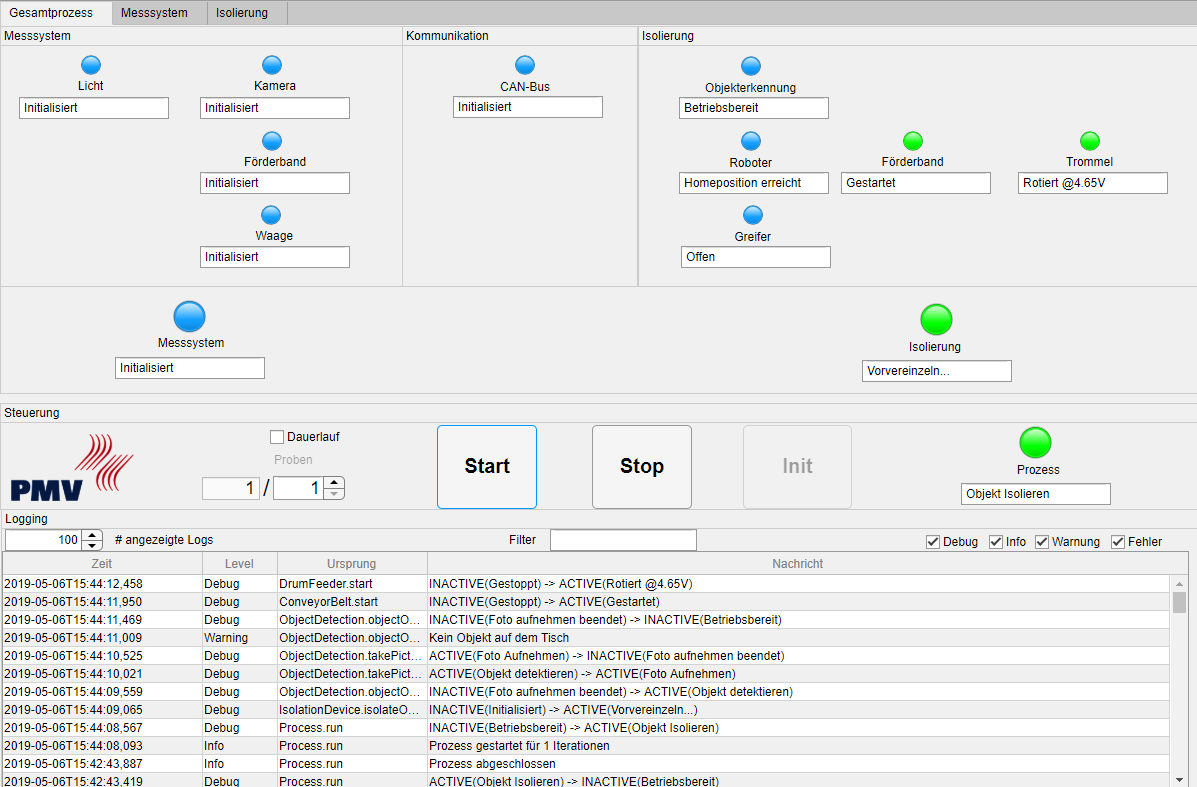
**Anwesenheit**

Dies erfordert die permanente Anwesenheit des Nutzers, wodurch das Betreiben der Anlage sehr Aufwandsintensiv ist.

**Innere Zustände nicht erkennbar → Fehler Abwarten**

Ebenso sind die internen Zustände der Objektrepräsentationen und der Komponentensoftware selbst nicht direkt überwacht werden. Es ist somit erst das Auftreten eines Fehlers feststellbar, jedoch nicht das Auftreten seiner Ursache.

* + 1. **Neuer Stand**
       1. **Konzept verschiedener Detailierungstiefe**



* + - * 1. **Logger**

**Ziel**

**Konsole**

**Datei**

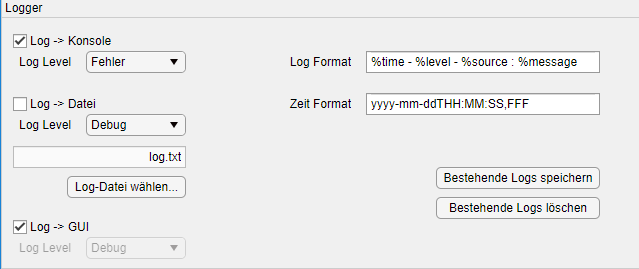
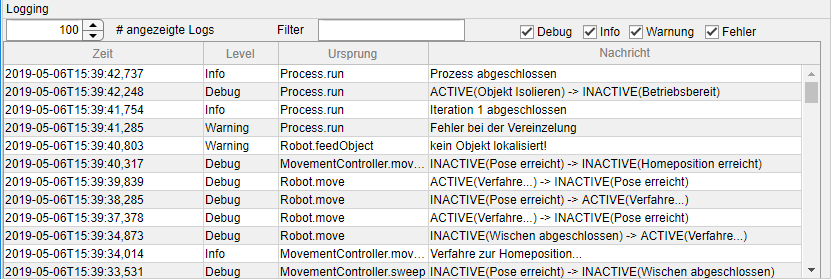
**Funktion (→ GUI/Tabelle)**

**Filterbarkeit**

**Zeit**

**Level**

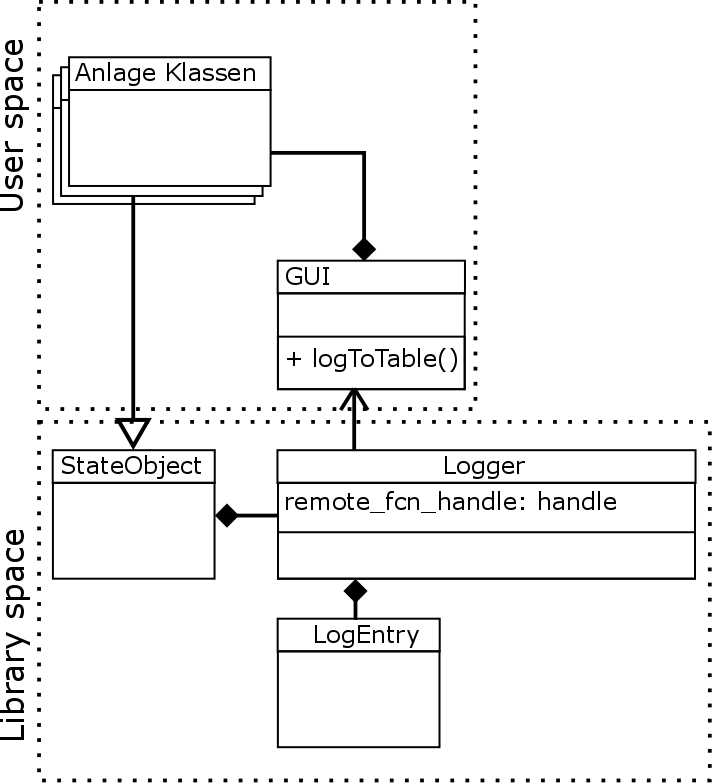
**Ursprung**



* + - * 1. **Zustandsüberwachung**

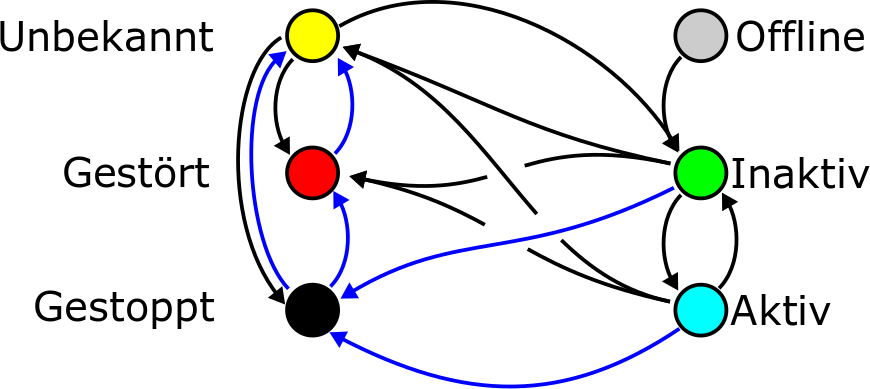
**Lampen**

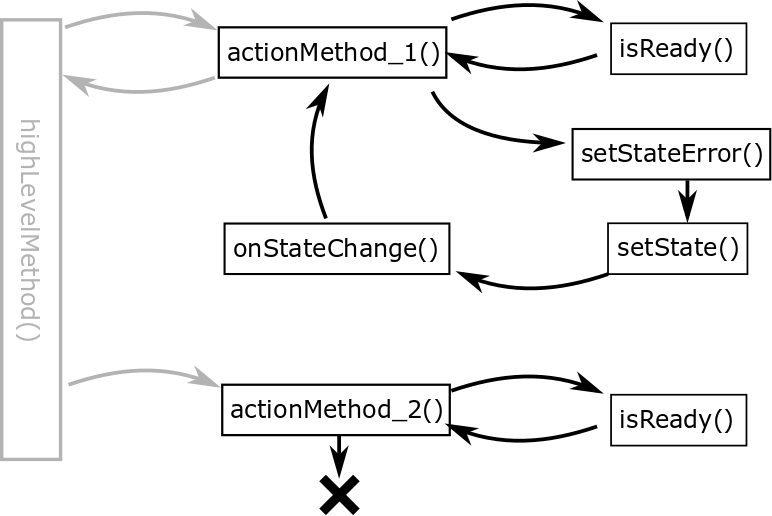
**Stateobject**

****

**Zustände**

**Zustandsübergänge**



****

|  |
| --- |
| function actionMethod(this,varargin)  if ~this.isReady; return; end  ... |

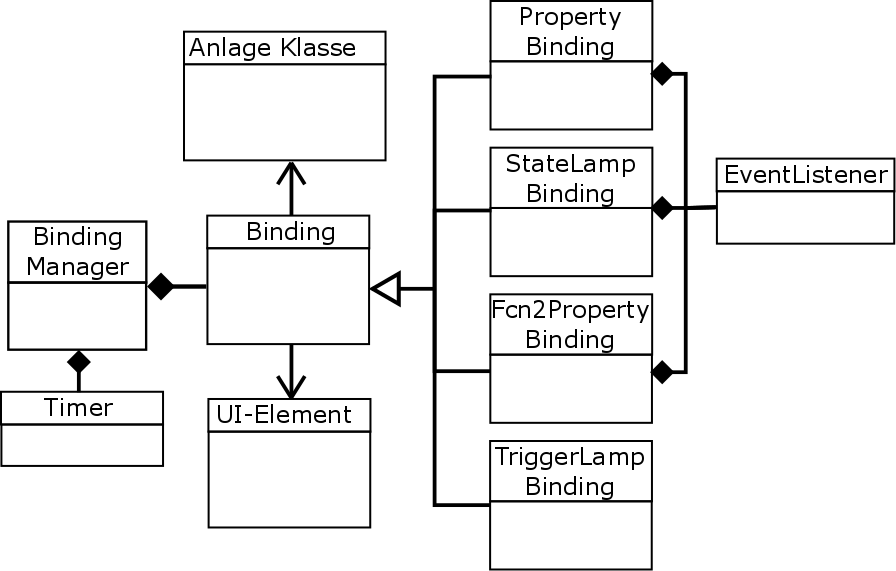
**Erlaubte / Verbotene**

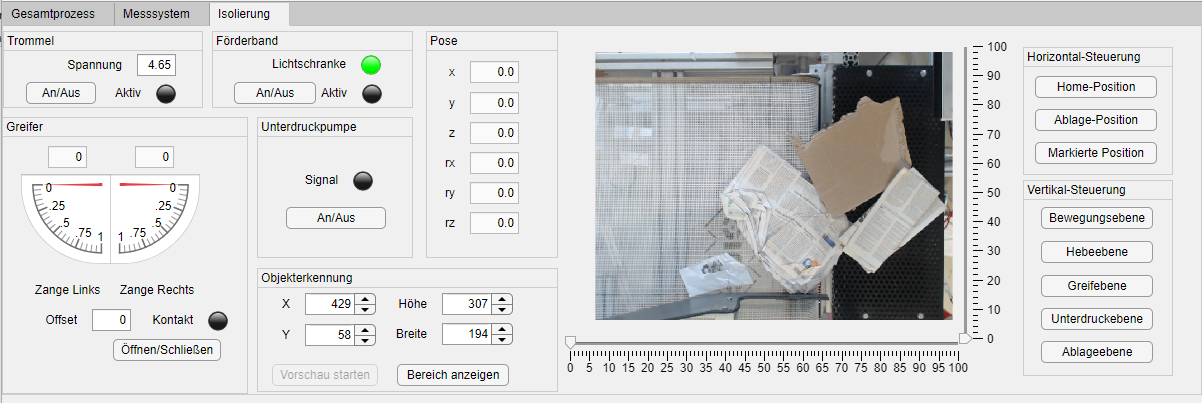
**Maschinelle / Nutzer**

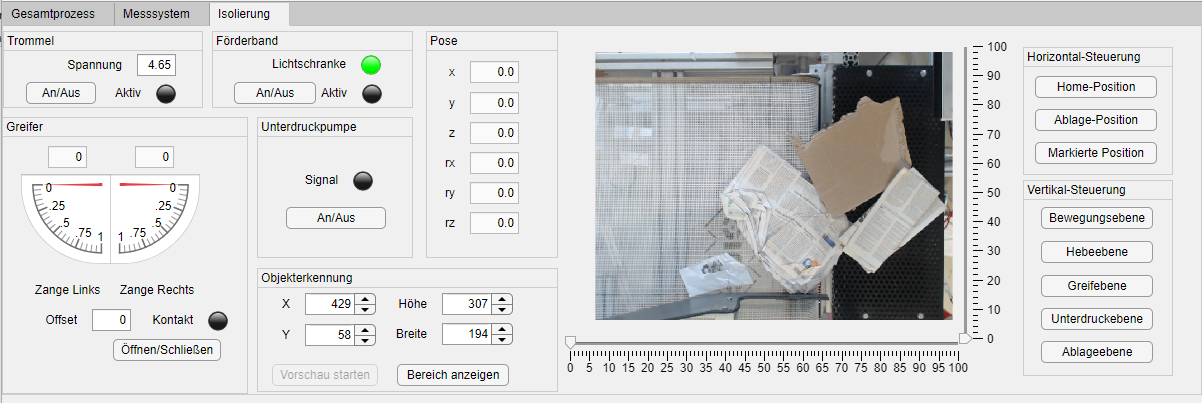
**Trennung Erstellung / Initialisierung**

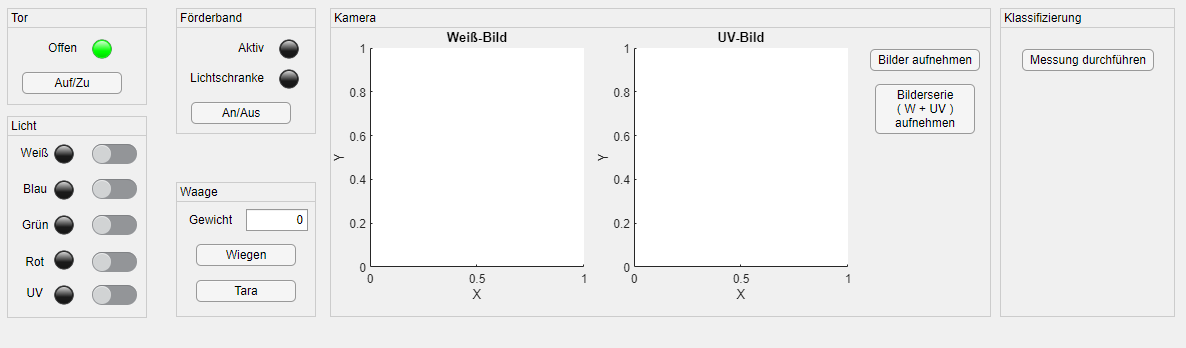
**Überwachbarkeit während Initialisierung**

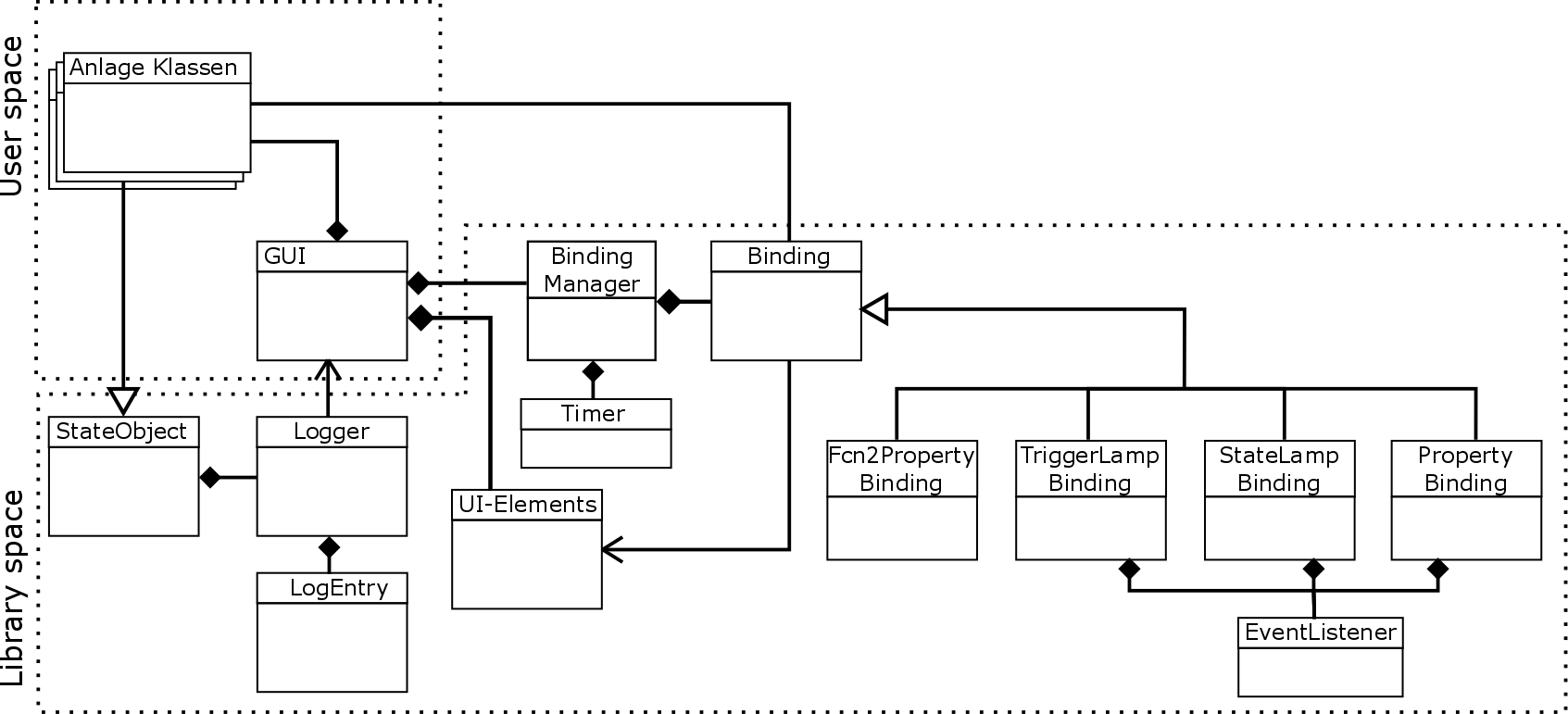
* + - * 1. **Detailüberwachung**

****

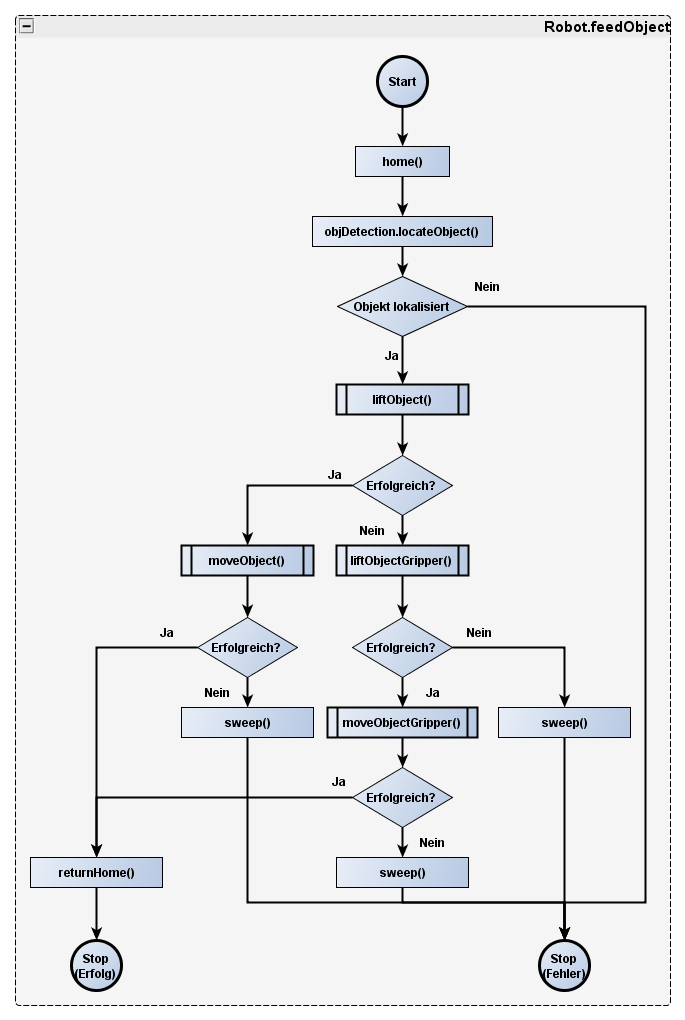






****

1. **Verbesserung**
   1. **Prozessanalyse**
      1. **Programmablaufplan des IST-Zustands**

****

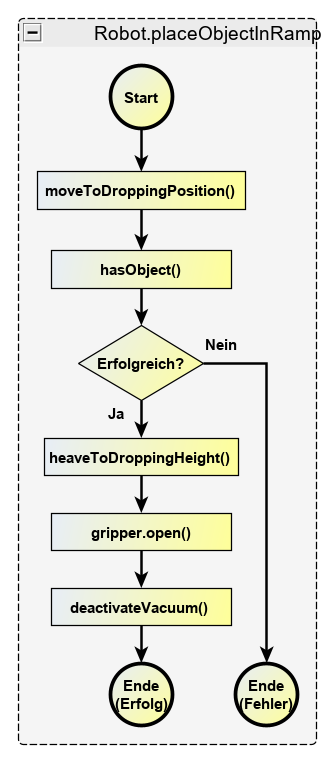
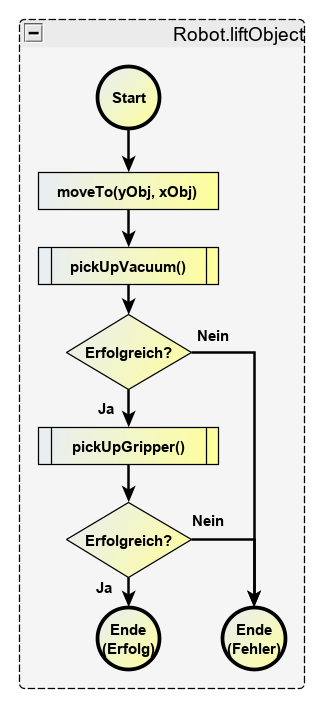
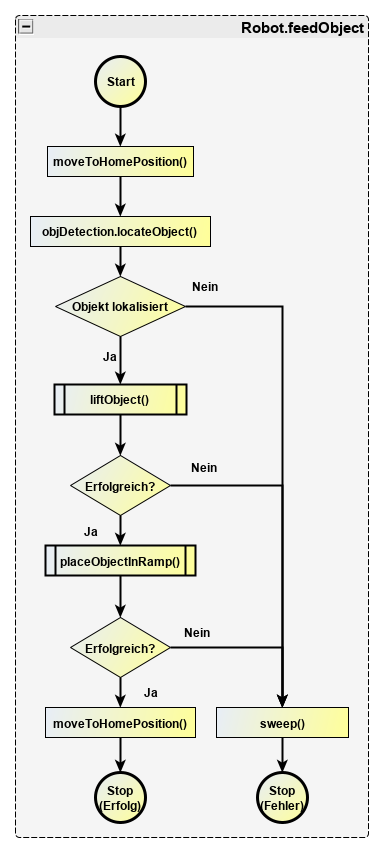
****

Abbildung 1 Test

* + - 1. **Redundanz**
      2. **Nicht definierte/abgefangene Fälle**
    1. **Vereinfachter logischer Programmablaufplan**
    2. **Programmablaufplan des SOLL-Zustands**
  1. **Roboter Bewegungskonzept**
     1. **Problemstellung**
        1. **Einfahren in gefährliche Positionen nicht verhindert**
        2. **Schwere Nutzbarkeit durch freie Verfügbarkeit des Raumes**
     2. **Lösung**
        1. **Ebenenmodell**

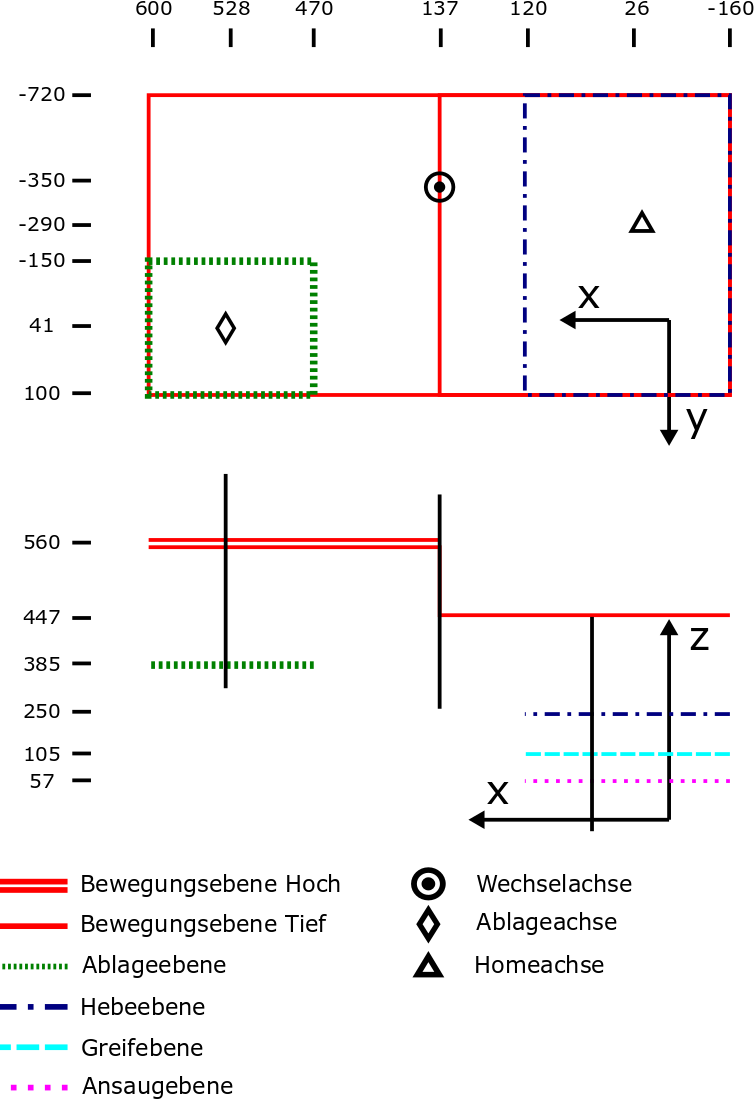
****

Abbildung 2 Ebenenbasiertes Bewegungskonzept

* + - * 1. **Einbeziehung Kopfdrehung (6 parameter, davon 4 fest)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ebene** | **Rotation X [°]** | **Rotation Y [°]** | **Rotation Z [°]** |
| Bewegungsebene Hoch | 135 | 120 | 0 |
| Ablageebene | -15 | 167 | 0 |
| Restliche Ebenen | 180 | 0 | 0 |

*Tabelle 1 Ebenenbedingte Kopfrotation*

* + - * 1. **Nur eine Bewegungsebene**

**Sonst nur vertikal**

**Definierte Bewegungen**

Im Folgenden wird das Bewegungskonzept anhand eines Beispiels verdeutlicht. Es wird der Bewegungsvorgang der beim Zuführen eines Objektes zur Messzellenrampe verwendet. Die gezeigten Befehle sind in der fertigen Implementierung in Methoden eines höheren Funktionslevels enthalten. Einzelne Befehl, wie ID 4 und ID 6 werden durch das Bewegungskonzept in mehrteilige Bewegungsabläufe übersetzt um ein sicheres Erreichen der Endposition zu gewährleisten.

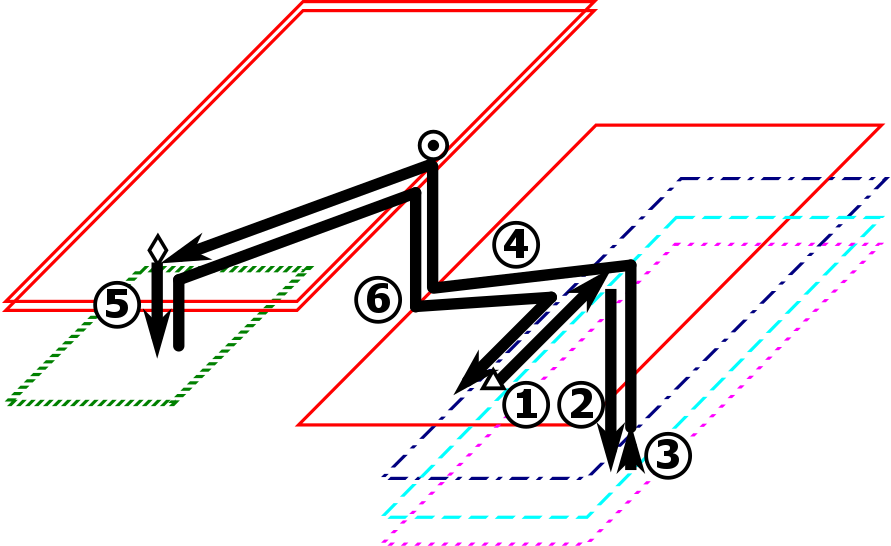
****

Abbildung 3 Beispiel Bewegungskonzept

|  |  |  |
| --- | --- | --- |
| **ID** | **Befehl** | **Aktion** |
| **1** | moveTo(X\_object,Y\_object) | *Verfährt horizontal zur Position (X\_object, Y\_object)* |
| **2** | heaveToGrippingHeight() | *Verfährt vertikal zur Greifebene* |
| **3** | heaveToLiftingHeight() | *Verfährt vertikal zur Hebeebene* |
| **4** | moveTo(X\_ramp,Y\_ramp) |  |
| **4.1** |  | *Verfährt horizontal zur Wechselachse* |
| **4.2** |  | *Verfährt vertikal entlang Wechselachse zur Bewegungsebene Hoch* |
| **4.3** |  | *Verfährt horizontal zur Ablageachse* |
| **5** | heaveToDroppingHeight() | *Verfährt vertikal entlang Ablageachse zur Ablageebene* |
| **6** | moveToHomePosition() |  |
| **6.1** |  | *Verfährt vertikal entlang Ablageachse zur Bewegungsebene Hoch* |
| **6.2** |  | *Verfährt horizontal zur Wechselachse* |
| **6.3** |  | *Verfährt vertikal entlang Wechselachse zur Bewegungsebene Tief* |
| **6.4** |  | *Verfährt horizontal zur Homeachse* |

*Tabelle 2 Befehls* → *Aktions Zusammenhang im Beispiel Bewegungskonzept*

* 1. **Förderbandkalibrierung**

Für Kamerabasierte Objekterkennungs- und Klassifikationsalgorithmen ist es wichtig möglichst genau die Position des zu erkennenden, bzw. des zu klassifizierenden Objektes zu kennen. Sowohl bei der Klassifizierung in der Messzelle als auch beim Roboter, muss daher eine möglichst genaue Positionierung stattfinden. Bei beiden System findet die letzte Bewegung des Altpapierobjektes durch ein Förderband statt. Diese werden jeweils durch einen Start- und einen Stop-Befehl, welche zeitlich versetzt erfolgen, gesteuert.Um ein Objekt mit den Förderbändern genau positionieren zu können, wird eine rudimentäre Kalibrierung vorgenommen.

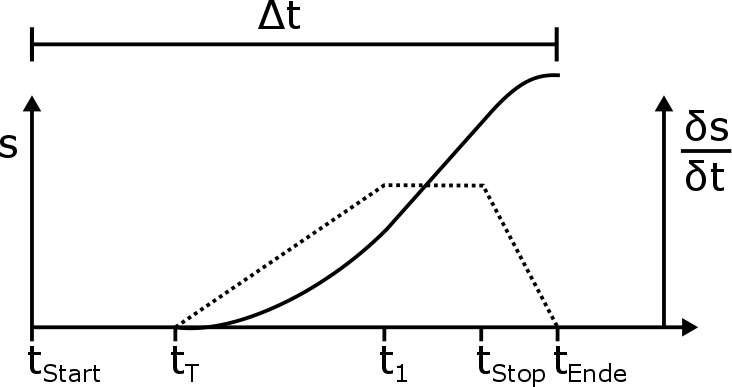


Abbildung 4 Angenommener Geschwindigkeits- und Positionsverlauf der Förderbänder

Es wird angenommen, dass nach dem Ausführen des Start-Befehls bei tStart eine Totzeit bis tT eintritt, bis es zur Bewegung des Förderbandes kommt. Nach einer Anlaufzeit, mit linearem Geschwindigkeitsverlauf, bis t1, hat das Förderband einen konstanten Geschwindigkeitszustand erreicht. Nach Ausführen des Stop-Befehls bei tStop wird fällt die Geschwindigkeit linear auf null. Die geförderte Distanz zum Zeitpunkt tEnde ist hiernach konstant und dient als Messgröße für die Kalibrierung.

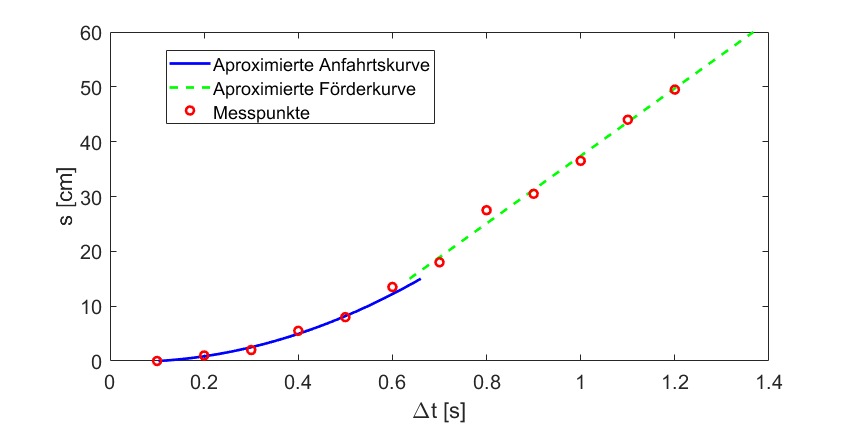
* + 1. **Förderband (Isolierung)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dauer Δt [s]** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 | 1.1 | 1.2 |
| **Distanz s [cm]** | 0 | 1 | 2 | 5.5 | 8 | 13.5 | 18 | 27.5 | 30.5 | 36.5 | 44 | 49.5 |

Für das Förderband der Isolierung wurden die Werte aus TABELLE X gemessen und aus ihnen wurden mit Hilfe des Curve Fitting Tools (Platzhalter1) von Mathworks die Funktionen aus (1) gebildet. In ABBILDUNG X ist diese zusammen mit den gemessenen Daten zu sehen.

|  |  |
| --- | --- |
|  | (1) |

|  |  |
| --- | --- |
|  | (2) |

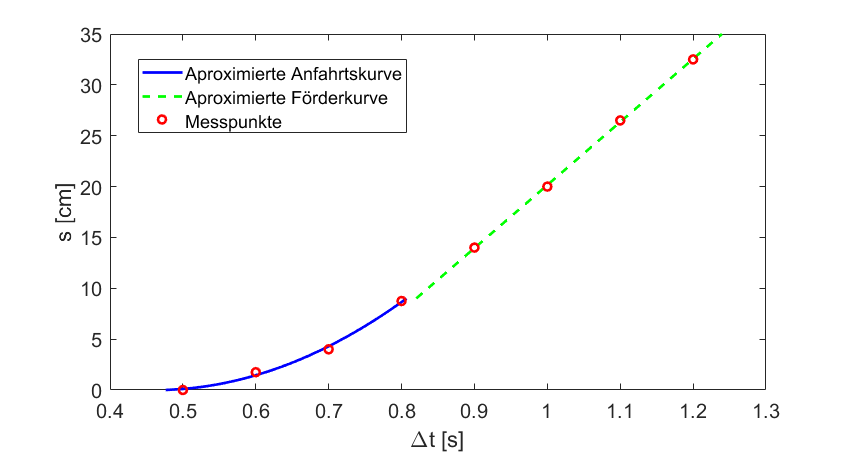


* + 1. **Förderband (Messzelle)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dauer Δt [s]** | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 | 1.1 | 1.2 |
| **Distanz s [cm]** | 0 | 1.75 | 4 | 8.75 | 14 | 20 | 26.5 | 32.5 |

|  |  |
| --- | --- |
|  | (3) |

|  |  |
| --- | --- |
|  | (4) |

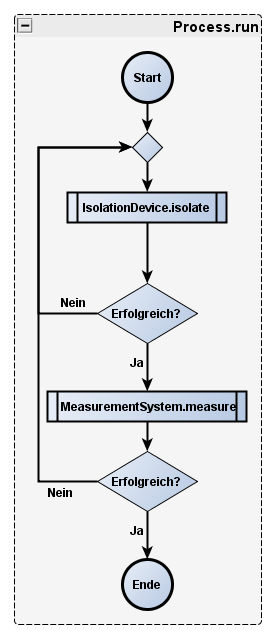


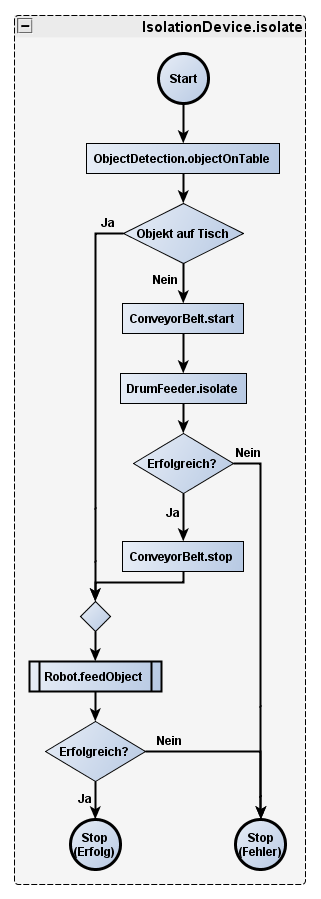
* + 1. **Ansteuerung Trommel???**

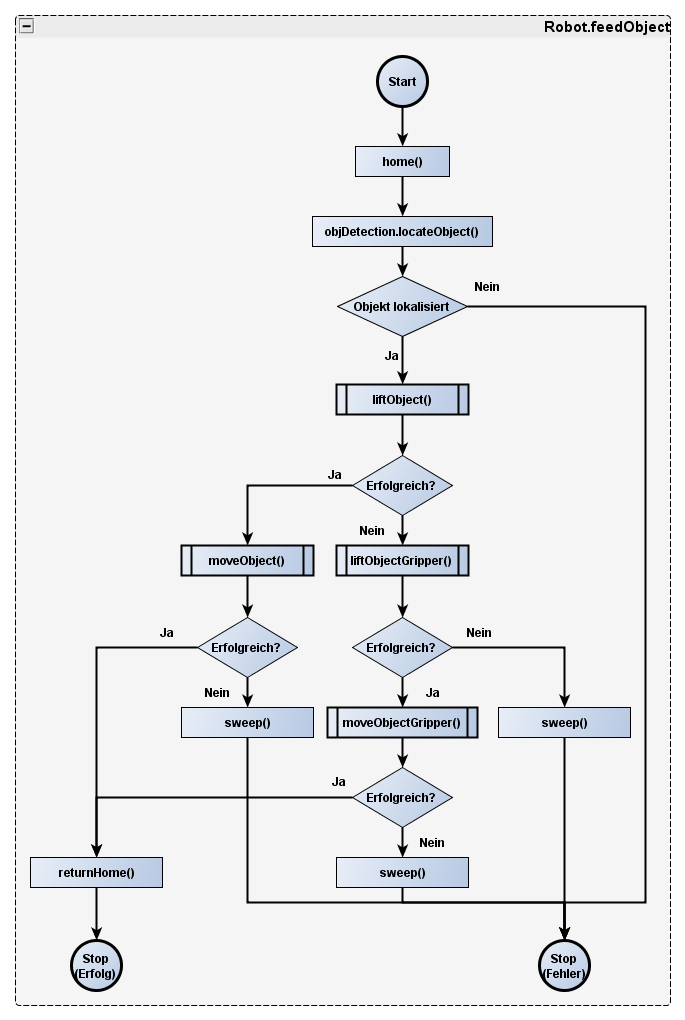
1. **Ausblick**
   1. **Ausbau Rückführgrößen von Anlagekomponenten**
   2. **Ausbau Messsensorik**
   3. **Ausarbeitung Sicherheitskonzept für unüberwachten Betrieb???**
   4. **Weitere Prozessanalyse**

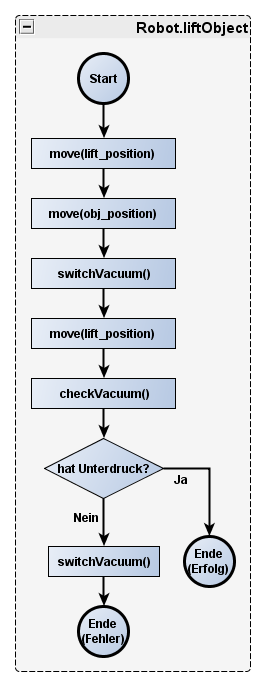
ANHANG

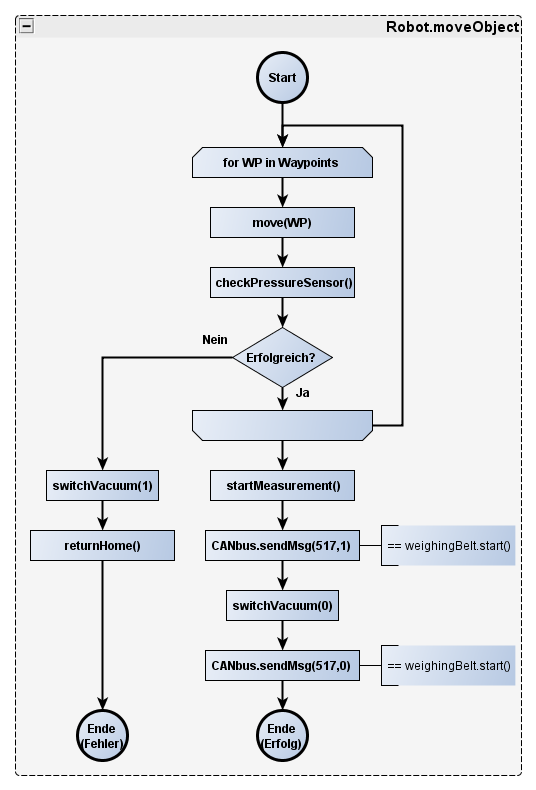
**Prozess alt**

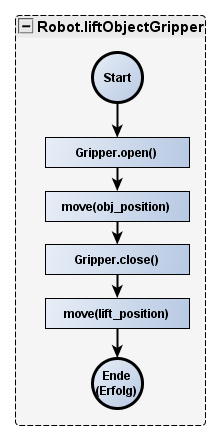
****

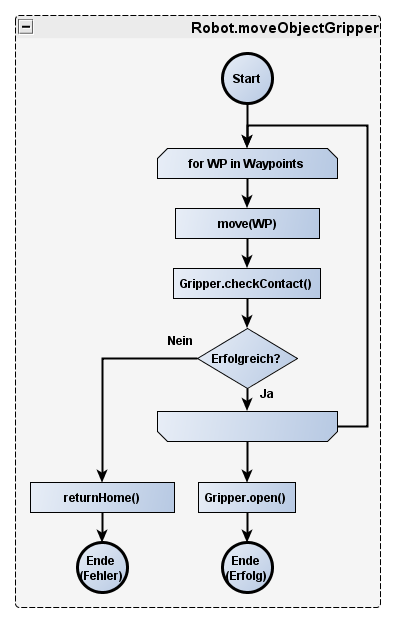
****

****

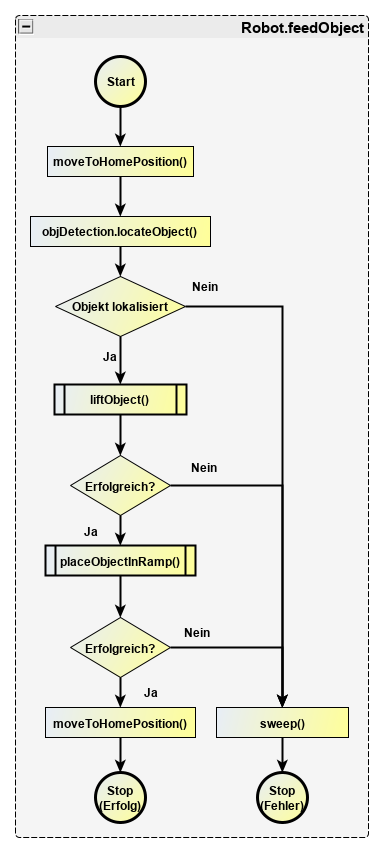
****

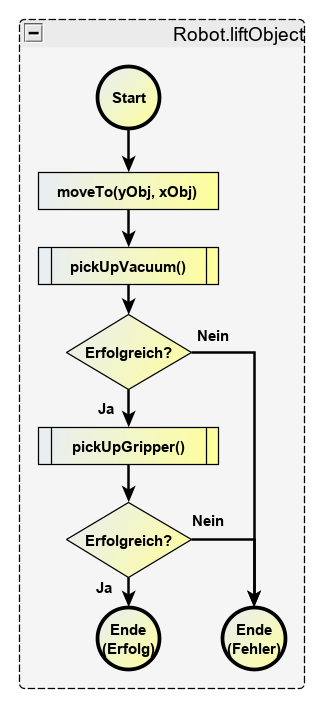
****

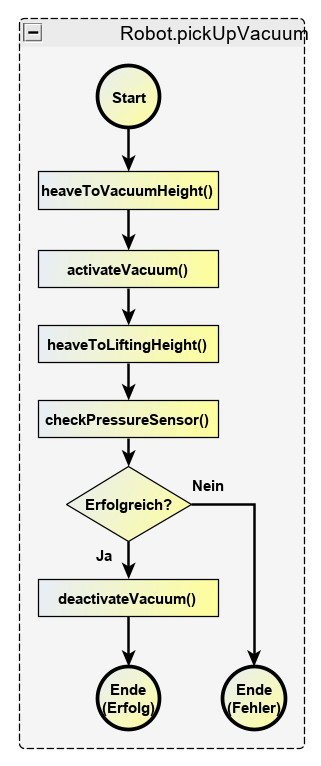
****

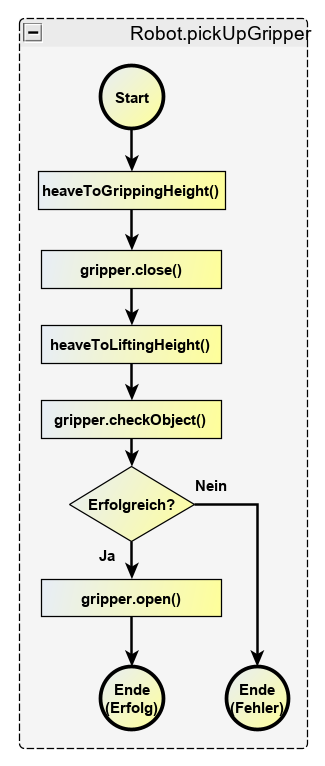
****

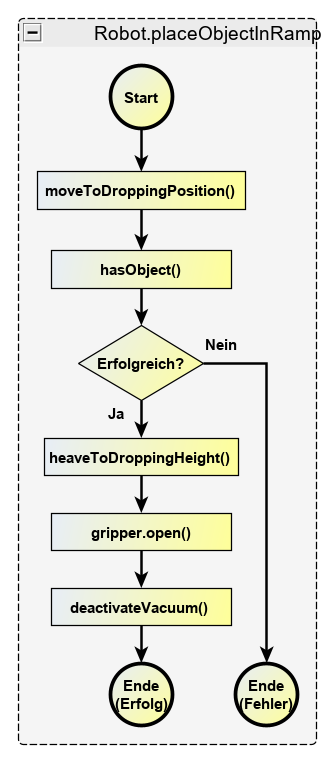
**Prozess neu**

****

****

****

****

****