



PA
Doku

2023

PA: Transportarm

2. Juli 2023

Emily Antosch 2519935

INHALTSVERZEICHNIS

Abbildungsverzeichnis	3
1 PA - Dokumentation des Transportarms	4
1.1 Dokumentation der Visualisierung	4
1.1.1 Beschreibung	4
1.1.2 Softwarekonzept	4
1.1.3 Weitere Hinweise	5

ABBILDUNGSVERZEICHNIS

1.1 Visualisierung der Lackieranlage 5

1.1 Dokumentation der Visualisierung

1.1.1 Beschreibung

Das Programm ist in der Lage, die Funktion einer Lackiereinrichtung mit Transportarm zu simulieren. Die Visualisierung ist in Abbildung 1.1 zu sehen. Die Anlage besteht aus zwei Bändern *A* und *E* und drei Stationen *L1*, *L2* und *K*. Auf einem Transportband kommen die zu lackierenden Teile an. Die Station *L1* ist die erste Lackierstation, an der die Teile lackiert werden. Die Station *K* ist die Konservierstation, an der die Teile konserviert werden. Die Stationen *L1*, *L2* und *K* sind jeweils mit einem Roboterarm ansteuerbar. Nachdem die Teile lackiert wurden, werden sie auf dem zweiten Transportband *E* abgelegt. Die Station *L2* ist die zweite Lackierstation, die die gleiche Funktion wie *L1* übernimmt.

Werkstücke können nur vorwärts durch die Anlage mit einem Transportarm bewegt werden. Es gibt, in der Version der Anlage, zwei Werkstücke, die in einem Durchlauf verarbeitet werden können. Ein Transport von einer Station zur nächsten dauert zwei Sekunden und jede Bearbeitung eines Werkstückes an einer Station dauert 5 Sekunden. Der Stand der Bearbeitung wird über eine Ampelanzeige erreicht. Die Ampel zeigt an, ob die Station frei ist, ob ein Werkstück bearbeitet wird oder ob ein Werkstück fertig bearbeitet worden ist. Dabei bedeutet Grün, dass die Station frei ist, Gelb, dass die Station die Arbeit abgeschlossen hat und Rot, dass die Bearbeitung des Werkstücks noch nicht beendet ist. Nur ein Werkstück kann gleichzeitig an einer Station liegen. Die Ampelanzeigen an den Bändern haben nur die Zustände Grün und Rot. Grün an der Station *E* bedeutet, dass ein Werkstück zur Abholung bereit ist, während Rot bedeutet, dass das Band leer ist. Umgekehrt heißt Grün an der Station *A*, dass die Station frei zur Abgabe eines Werkstücks ist und Rot, dass die Station noch nicht bereit für eine weitere Annahme ist (Band voll).

1.1.2 Softwarekonzept

Die Steuerung nimmt für die Visualisierung des Prozesses einen objektorientierten Ansatz. Der FB *ArmModel* stellt dabei das Herz der Anwendung dar und managet die Bewegung des Arm innerhalb der Laufzeit des Programms. Der Status der Anlage wird über die ENUM *ArmModelState* geregelt. In *ArmModel* werden beim Start zunächst alle Werte zurückgesetzt und dann wird direkt in den Zustand *NormalOperation* gewechselt. Dort wird fortlaufend überprüft, ob eine der Trigger gesetzt wurden, die das Bewegen von einem Werkstück von A nach B einleiten. Ist dies der Fall, werden zunächst verschiedene Informationen über den Transport gesammelt, um die richtige Bewegung ausführen zu können. Dazu gehören die Ausgangsposition des Werkstücks und das Ziel. Dann

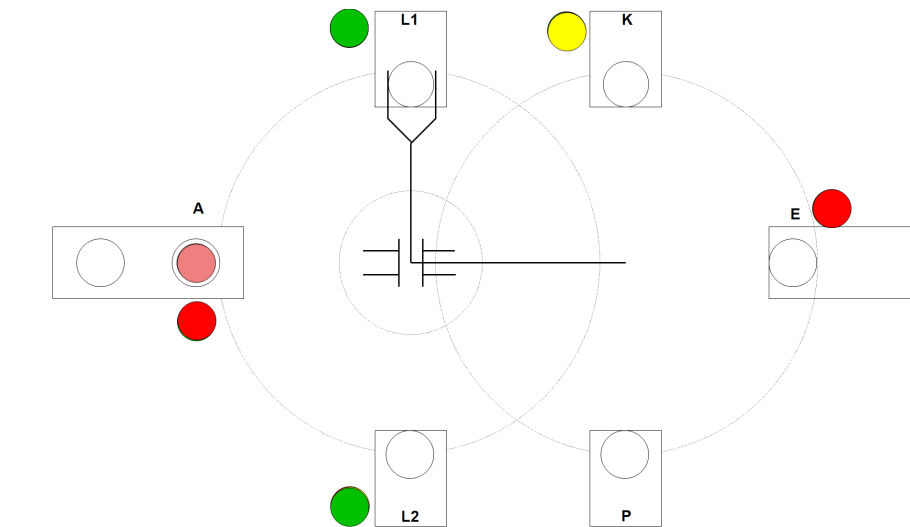


Abbildung 1.1: Visualisierung der Lackieranlage

wird eine Animation ausgeführt, die in dem FB *ProductionObject* definiert ist, ausgeführt. Jede weiterer Transport wird gesperrt, sodass sich immer nur ein Werkstück im Transport befinden kann. Am Ende des Transports werden die jeweiligen Zustände des Werkstücks, wie zum Beispiel die derzeitige Position, aktualisiert und die entsprechenden Stationen (*ProductionStation*) werden darüber informiert, dass sie entweder ein Werkstück verloren haben oder ein neues dazugewonnen haben. Gleichzeitig führt die neue Station des Werkstücks ihre jeweilige Funktion aus. Ein Öffnen der Klammern um ein Objekt ist nicht notwendig um einen Transport zu veranlassen. Das könnte jedoch durch eine Erweiterung hinzugefügt werden. Zurzeit muss ein solcher Bewegungsbefehl über die Force-Tabelle erfolgen. Eine Steuerung könnte also nun über die Visualisierung gelegt werden, die diesen Prozess steuert und auch automatisiert.

Die Variablen, die gesetzt werden können, um einen Transport von einer Station zur nächsten zu veranlassen, sind:

- *ArmModel.MoveToL1*
- *ArmModel.MoveToL2*
- *ArmModel.MoveToKfromL1*
- *ArmModel.MoveToKfromL2*
- *ArmModel.MoveToE*

Bestimmte Konzepte zum Ablauf des Prozess sind bereits implementiert. So wird ein Bewegungsbefehl nicht ausgeführt, wenn dieser zu einer Kollision von zwei Werkstücken führen würde.

1.1.3 Weitere Hinweise

Sollte es ein Problem mit dem Programm und Unklarheiten bei der Bedienung geben, erreichen Sie mich unter: emilylucia.antosch@haw-hamburg.de.