

Virtuelle Inbetriebnahme der HLF Produktion auf Basis von Simulationsmodellen

--- Skript für Selbststudium ---

1.	DIGITALER ZWILLING	2
2.	CIROS STUDIO	3
3.	ÜBUNGEN.....	4
3.1.	Übung 1: Simulationsmodell erstellen.....	4
3.2.	Übung 2: UA-Expert und DigitalTwin	8
3.3.	Übung 3: Condition Monitoring mit DigitalTwin.....	12
3.4.	Übung 4: MApp mit Devices	14
4.	AUFGABE	18
4.1.	Aufgabenbeschreibung	18
4.2.	Virtuelle Inbetriebnahme.....	19
4.3.	Reale Inbetriebnahme	20
5.	ANHANG	21
5.1.	Klasse Device	21
5.2.	Klasse Order.....	21
5.3.	Klasse MAppModel	22
5.4.	Unified Automation UA Expert.....	22
5.5.	CIROS Studio	23

1. Digitaler Zwilling

Aus der Industrie 4.0 Initiative hat sich der Begriff des digitalen Zwillings (engl. *digital twin*) für komplexere Systeme entwickelt. Die Visualisierung der Daten als virtuell ermöglicht insbesondere interdisziplinären Teams die Beurteilung komplexer Zusammenhänge in einer gewohnten Umgebung: Dabei ist der digitale Zwilling eine digitale Repräsentanz eines materiellen oder immateriellen Objekts aus der realen Welt in der digitalen Welt. Sie sind mehr als reine Daten und bestehen aus **Modellen des repräsentierten Objekts** und können daneben Simulationen, Algorithmen und Services enthalten¹.

Ein **digitaler Zwilling für Anlagen** ist demnach ein virtuelles Abbild einer physischen Anlage, wobei als virtuelle Repräsentation ein CAD-Modell genutzt werden kann. Um ein genaues Abbild der Anlage zu gewährleisten, wird der digitale Zwilling mithilfe von Daten aus der realen Welt erstellt und aktualisiert (z.B. *durch eine Steuerungsapplikation*). Für die Bereitstellung von Zustandsinformationen sind geeignete Schnittstellen erforderlich, dafür existieren eine Reihe von Standardschnittstellen (z.B. *OPC-UA*²), die jedoch in der Praxis häufig nur bedingt alle erforderlichen Informationen bereitstellen. Zustände und reale Objekte müssen zudem häufig zusätzlich verknüpft werden. So kann beispielsweise ein Anlagenteil oder eine Fertigungszelle einer Anlage häufig durch mehrere Sensoren überwacht werden. Dabei stellt jeder Sensor wiederum mehrere Sensorinformationen bereit, so dass eine Zuweisung zwischen Betrachtungseinheit und den Identifikatoren der Sensorinformationen bestehen muss. Des Weiteren liegen Zustandsinformationen in unterschiedlichen Aktualisierungsraten vor. Dies kann insbesondere bei Erfassung, Speicherung und Analyse zu Herausforderungen führen.

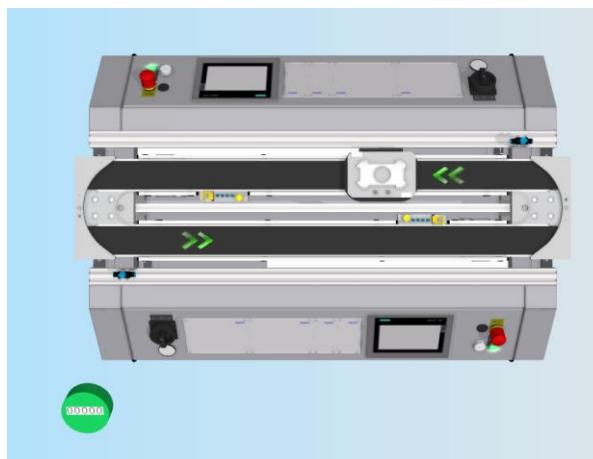



Abbildung 1: Digitaler Zwilling

¹ Siehe [Digitaler Zwilling – Wikipedia](#)

² Siehe [OPC Unified Architecture – Wikipedia](#)

	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 3 von 24

Der digitalen Zwillinge für Anlagen kann sowohl für die Planung und Entwicklung als auch für den Betrieb und die Wartung eingesetzt werden. Bei der sogenannten **virtuellen Inbetriebnahme**³ wird der digitale Zwilling der Anlage oder Maschine verwendet, um die Inbetriebnahme in der realen Welt zu simulieren. Dadurch können frühzeitig Diskrepanzen und Fehler erkannt werden, was Zeit- und Kosteneinsparungen ermöglicht.

Die Projektdauer eines technischen Systems wird signifikant von der Inbetriebnahme beeinflusst, wobei hier das Beseitigen von Fehlern in der Steuerungsapplikationen den dominierenden Aufwand darstellt. Durch die Entwicklung immer komplexerer Anlagen, die horizontal und vertikal vernetzt sind, steigt der Anteil an Software innerhalb eines technischen Systems deutlich an und übt somit einen erheblichen Einfluss auf die Inbetriebnahme-Zeiten aus. Um die Inbetriebnahme-Zeiten zu reduzieren, stellt die virtuelle Inbetriebnahme ein geeignetes Mittel dar. Darüber hinaus kann mit dieser Methode auch gleichzeitig die Softwarequalität von Steuerungsapplikationen verbessert werden.

2. CIROS Studio

CIROS Studio⁴ ist eine Softwaresystem für die 3D-Anlagensimulation. Mit CIROS Studio modellieren Sie Layouts und Prozesse, simulieren Roboterzellen und automatisierte Fertigungsanlagen und visualisieren Abläufe.

Das System ermöglicht

- die Modellierung von Layouts und Prozessen,
- die Simulation von Roboterzellen und automatisierte Fertigungsanlagen und
- die Visualisierung von Fertigungsabläufen.

CIROS Studio wird für die Anlagenplanung, Konstruktion, Elektroplanung, Steuerungsprogrammierung, Inbetriebnahme, Bedienertraining, Vertrieb und Marketing eingesetzt.

³ Siehe [Virtuelle Inbetriebnahme – Wikipedia](#)

⁴ Siehe [CIROS® \(festo-didactic.com\), CIROS Studio for 3D Factory Simulation - VEROSIM Solutions](#)

 OST Ostschweizer Fachhochschule	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 4 von 24

3. Übungen

Das Ziel der folgenden Übungen ist es, ein umfassenderes Verständnis im Umgang mit Simulationsmodellen für die virtuelle Inbetriebnahme zu erlangen.

3.1. Übung 1: Simulationsmodell erstellen

In dieser Übung erstellen Sie ein Modell mit einem Förderband (engl. Conveyor Belt) sowie zwei Umlenkteilen (engl. Deflection), welches als Teilmodell der Hybriden Lernfabrik zählt.

1. Öffnen Sie das CIROS Studio und wählen Sie den *Modellassistent* aus.

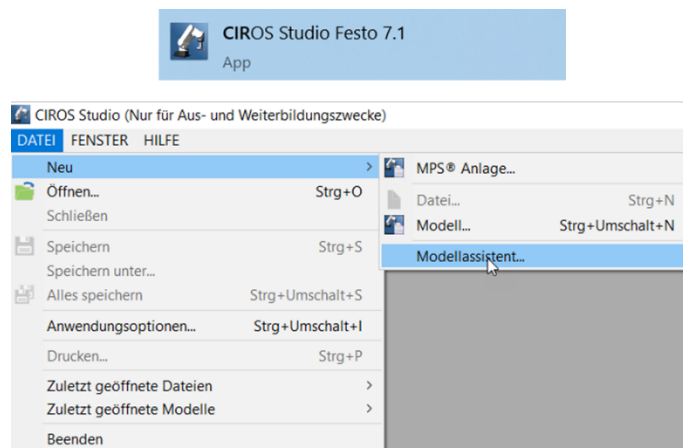




Abbildung 2: *CIROS Studio*

2. Leeres Projekt erstellen: In der Rubrik *Speicherort* den Speicherort aus und geben Sie unter *Modellname* MyFirstDigitalTwin ein. Unter Rubrik *Roboterauswahl* wählen Sie *Leeres Model* und unter *Zusammenfassung* Drücken Sie *Fertigstellen*.
3. Anlage modellieren: Unter Menu **MODELLIERUNG** wählen Sie  *Editiermodus* aus. Danach öffnen Sie unter **MODELLIERUNG** die Ansicht der  *Modellbibliotheken*.

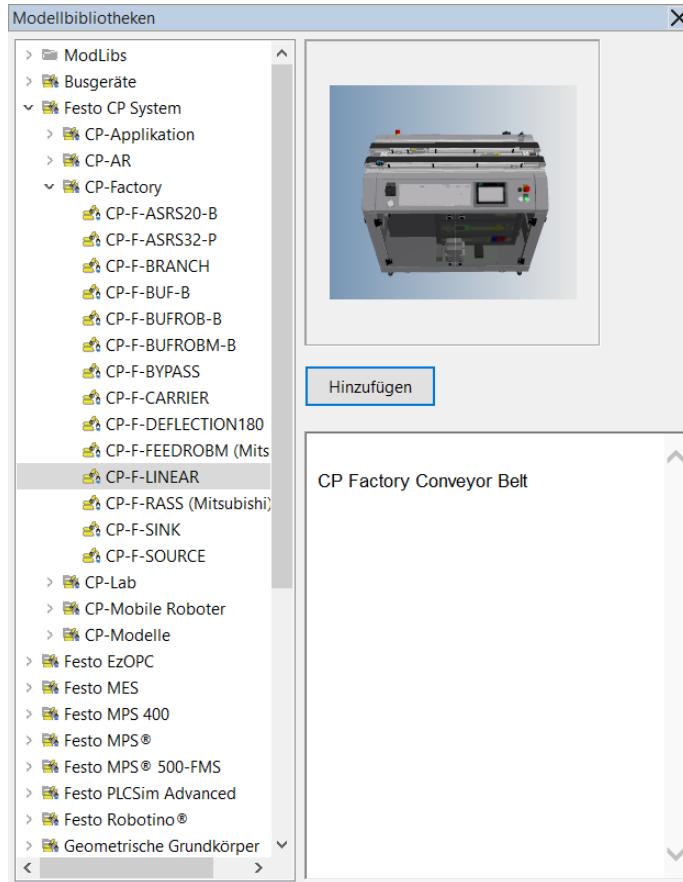


Abbildung 3: Modellbibliotheken

4. Fügen Sie ein Förderband unter Festo CP System/CP-Factory/CP-FLINEAR hinzu.

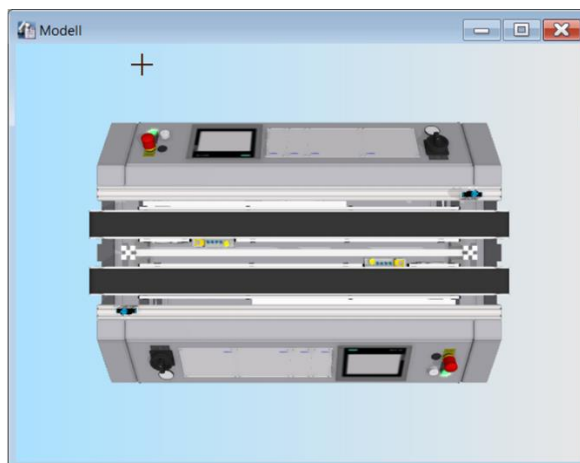
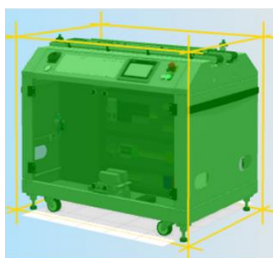


Abbildung 4: Modellfenster mit Förderband

5. Um weitere Modellteile zu einem Gesamtmodell optimal verbinden zu können, wählen Sie unter Menu *ANSICHT/Standardansichten Aufsicht* aus.
6. Wählen Sie in der Ansicht Modellbibliotheken ein Umlenkteil unter *Festo CP System/CP-Factory/CP-DEFLECTION180* aus und bestätigen Sie mit *Hinzufügen*.

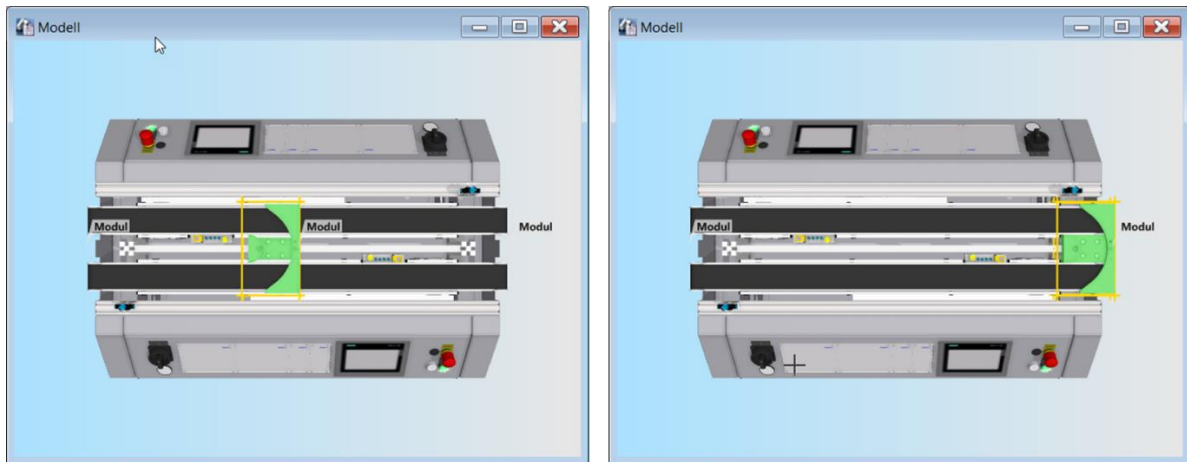
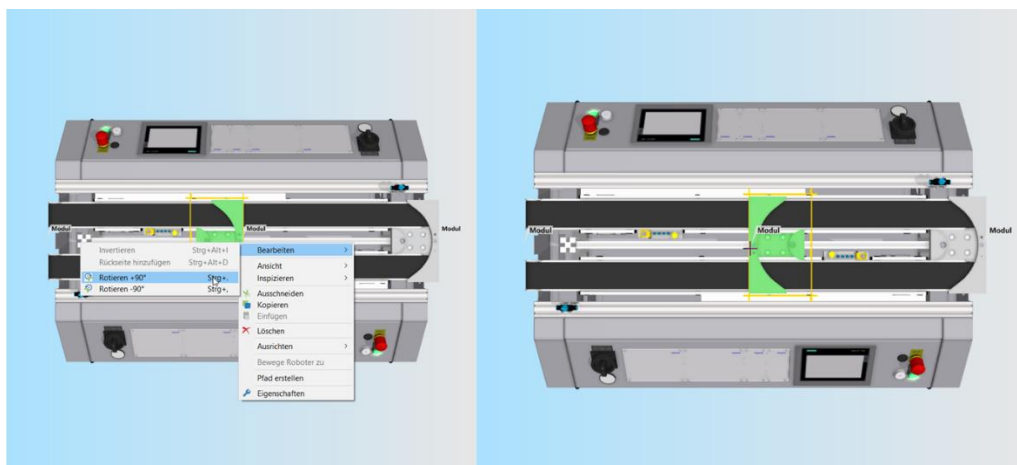


Abbildung 5: Modellfenster mit Umlenkteil

7. Bewegen Sie das Umlenkteil so weit nach rechts, bis sich die beiden Ports *Modul* verbinden.
8. Fügen Sie ein weiteres Umlenkteil hinzu, drehen Sie das Teil um 180°. Anschliessend bewegen Sie das Teil so weit nach links, bis sich die beiden Ports *Modul* verbinden.



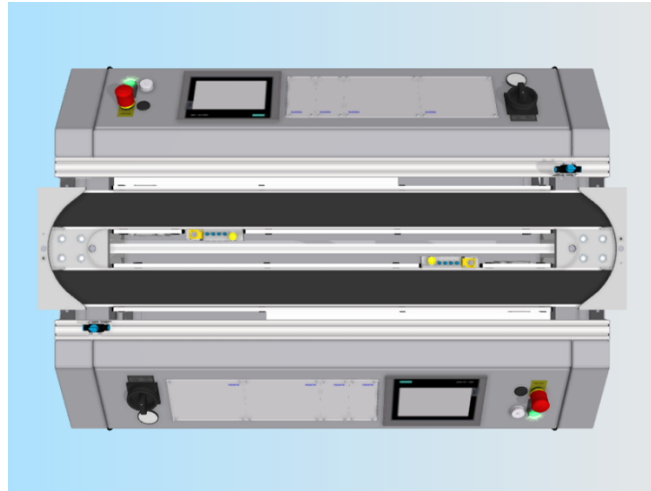


Abbildung 6: Modellfenster mit zwei Umlenkteilen

9. Für die Simulation fehlt nun noch eine Quelle. Wählen Sie in der Ansicht Modellbibliotheken die Quelle unter *Festo CP System/CP-Factory/CP-SOURCE* aus und bestätigen Sie mit *Hinzufügen*. Anschliessend bewegen Sie die Quelle so weit nach links, bis sich die beiden Ports *Modul* verbinden.

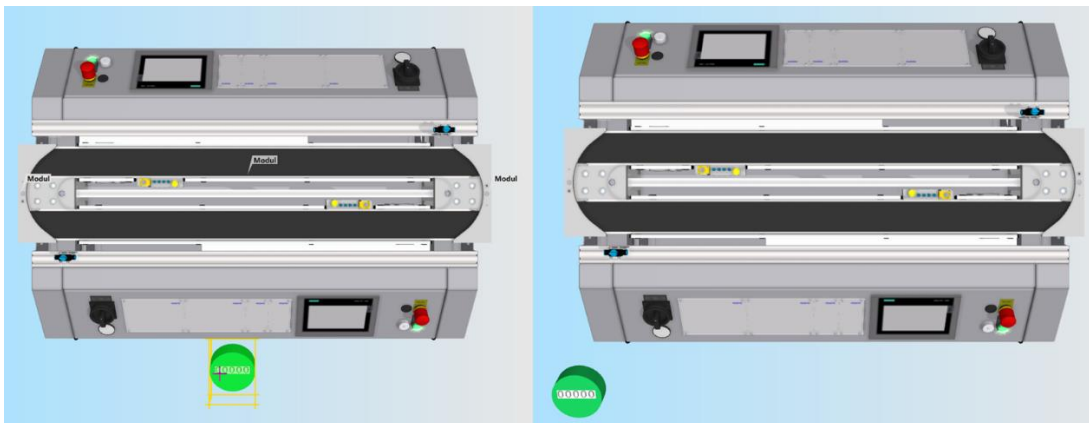













Abbildung 7: Modellfenster mit einer Quelle

10. Schliessen Sie die Modellbibliothek und schalten Sie unter Menu **MODELLIERUNG** den  *Editiermodus* aus.
11. Starten Sie die Simulation: Wählen Sie unter Menu *Extras* die Option  *OPC-UA-Server* aus, und starten Sie die Simulation unter  *Simulation* mit *Start/Stop*. Anschliessend klicken Sie im Modell auf den  Source-Button, um ein Werkstückträger (engl. *Carrier*) zu generieren.

	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 8 von 24

3.2. Übung 2: UA-Expert und DigitalTwin

In dieser Übung Durchsuchen und Erkunden Sie den OPC-UA Server und deren Datenmodell im Projekt *DigitalTwin* in *CIROS*. Dabei führen Sie mit dem OPC-UA Testclient von [Unified Automation UA Expert](#) Experimente zur Visualisierung und Steuerung des virtuellen Simulationsmodells *DigitalTwin* durch.

1. Öffnen Sie mit  *CIROS* aus dem Projektordner ...*HLF**DigitalTwin* die Projektvorlage *DigitalTwin.modx*.
2. Erkunden Sie *Ciros Studio*: Öffnen Sie mit dem *Model-Explorer* das Modell und wählen Sie die einzelnen Objekte aus. Wieviel und welche Modellteile gehören zum Modell?
3. Starten Sie die *Simulation*: Wählen Sie unter Menu *Extras* die Option  *OPC-UA-Server* aus, und starten Sie die Simulation unter  *Simulation* mit *Start/Stop*. Anschliessend klicken Sie im Modell auf den  Source-Button, um ein Werkstückträger (engl. *Carrier*) zu generieren.
4. Starten Sie auf ihrer Entwicklungsumgebung den Testclient  *Unified Automation UA Expert*.
5. Öffnen Sie unter Menu *Server*  *Add...* das Dialogfenster *Add Server*. Geben Sie Sie unter *Configuration Name* *Digital Twin Server* ein, wählen Sie den *OPC-UA Server* vom Modell *DigitalTwin* und bestätigen Sie mit *OK*. Falls kein Server auftaucht, überprüfen sie mit Rechtsklick auf *Local->Edit URL* ob sie mit dem in **Abbildung 9** übereinstimmt (Localhost funktioniert solange *CIROS* auf demselben Computer läuft wie der UA-Expert).

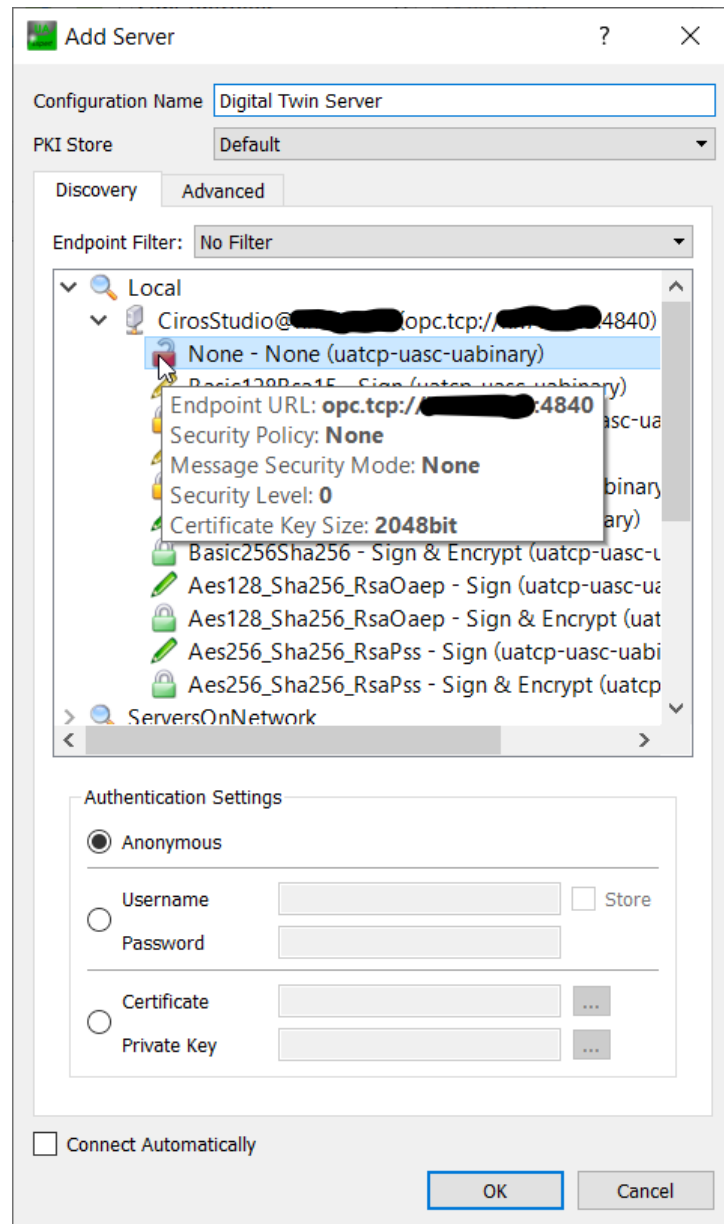


Abbildung 8: OPC-UA Server vom Modell DigitalTwin

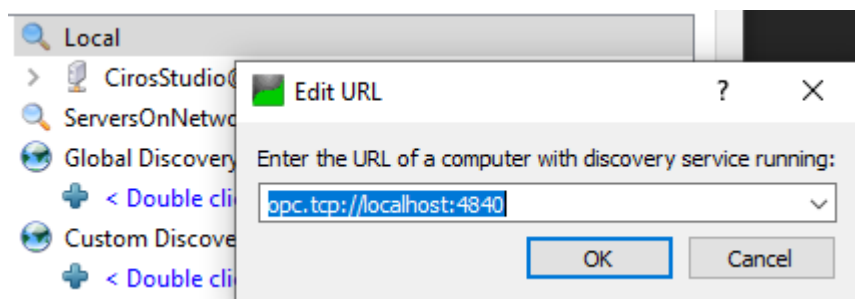



Abbildung 9 Edit URL

6. Öffnen Sie nun unter Menu *Server*  *Properties...* das Eigenschaftsfenster der Serververbindung und merken Sie sich den *Endpoint URL* für weitere Anwendungsbeispiele.

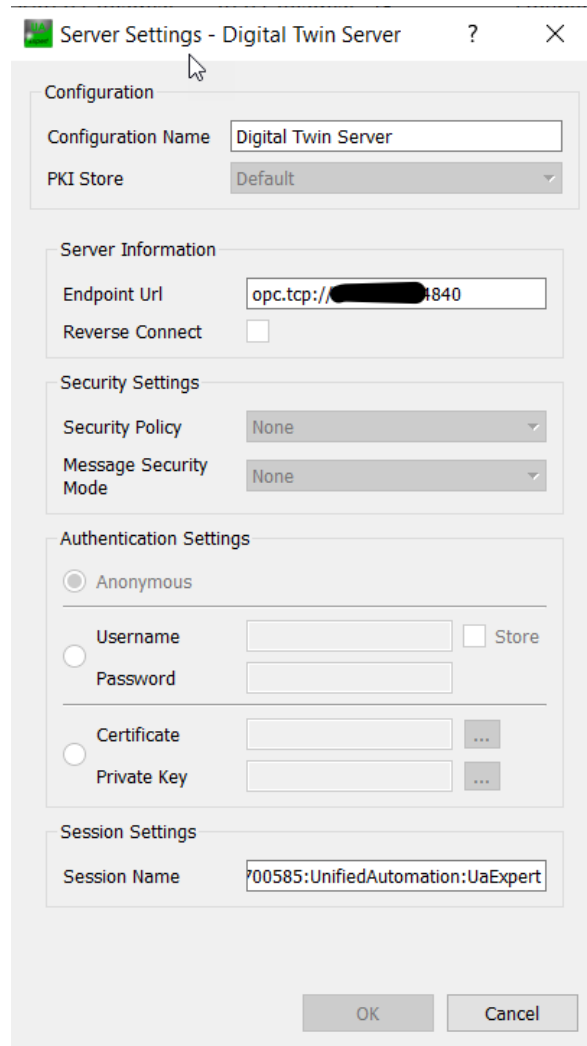



Abbildung 10: Eigenschaftsfenster vom *OPC-UA Server*

7. Um das Modell des OPC-UA Servers *Address Space* zu erhalten, wählen Sie nun unter Menu *Server*  *Connect*.
8. Erkunden Sie *UA-Expert*: Suchen Sie im Fenster *Address Space* die Datenvariable *Root/Objects/Simulation/Simulation time* und stellen Sie diese mittels *drag&drop* in den *Access Data View*. Überprüfen Sie, ob sich die Simulationszeit fortlaufend ändert.

9. Überprüfen Sie die Zustände Lichtschranken: Um eine Eingangslichtschranke im Modell zu visualisieren, ist im *Ciros Studio* den *Model-Explorer* zu öffnen und die *Lichtschranke* unter *Model/Objekte/CP-F-LINEAR/SPS_A/Sensor_Band_links* zu selektieren. Anschliessend wählen Sie im *UA-Expert*, Fenster *Address Space* die Datenvariable *Root/Objects/ENVIRONEMENT/CP-F-LINEAR/Objects@CP-F-LINEAR/SPS_A/Objects@SPS_A/Sensor_Band_links/Outputs@Sensor_Band_links/Erkannt* aus und stellen Sie diese mittels *drag&drop* in den *Access Data View*. Die *NodeId* muss mit dem in **Abbildung 11** übereinstimmen.






Attributes	
Attribute	Value
▼ Nodell	ns=3;s= OUT:Sensor_Band_links Erkannt
NamespaceIndex	3
IdentifierType	String
Identifier	OUT:Sensor_Band_links Erkannt

Abbildung 11 Sensor_Band_links NodeId

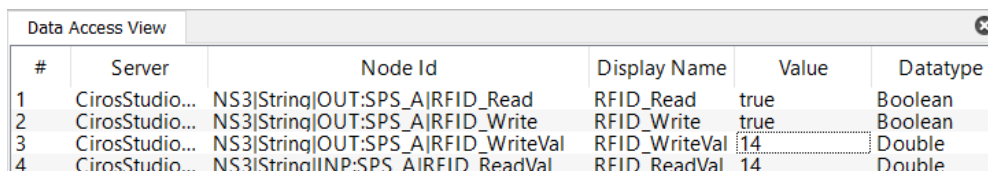
10. Beobachten Sie nun, ob sich der Variablen-Zustand beim Lichtschranken-Eintritt und -Austritt des Werkstückträgers ändert. Überprüfen Sie ebenfalls die weiteren Lichtschranken.
11. Verändern Sie Bandgeschwindigkeit: Wählen Sie im *UA-Expert*, Fenster *Address Space* die Datenvariable *Root/Objects/ENVIRONEMENT/CP-F-LINEAR/Objects@CP-F-LINEAR/SPS_A/Objects@SPS_A /Outputs@SPS_A/DOUT0_x6* aus und stellen Sie diese mittels *drag&drop* in den *Access Data View*. Setzen den *Value* der Datenvariable auf *true* und überprüfen Sie, ab sich die Bandgeschwindigkeit reduziert hat. Führen Sie das Experiment ebenfalls mit dem zweiten Band durch.

3.3. Übung 3: Condition Monitoring mit DigitalTwin

Im Modul „Vernetzung und Kommunikation“ der [Hybriden Lernfabrik](#) (kurz HLF) haben Sie das Beispielprojekt „Condition Monitoring mit OPC-UA“ erfolgreich umgesetzt. Diese Anwendung können Sie ebenfalls mit dem digitalen Zwilling überprüfen:

1. Öffnen Sie dazu mit  CIROS aus dem Projektordner ...*HLF**DigitalTwin* die Projektvorlage *DigitalTwin.modx*.
2. Starten Sie die Simulation: Wählen Sie unter Menu *Extras* die Option  *OPC-UA-Server* aus, und starten Sie die Simulation unter  *Simulation* mit *Start/Stop*. Anschliessend klicken Sie im Modell auf den  Source-Button, um ein Werkstückträger (engl. *Carrier*) zu generieren.
3. Starten Sie auf ihrer Entwicklungsumgebung den Testclient  *Unified Automation UA Expert* und verbinden Sie sich mit dem *OPC-UA Server* von *DigitalTwin*.
4. Definieren Sie eine RFID-ID: Wählen Sie im *UA-Expert*, Fenster *Address Space* folgende *Datenvariablen* aus und stellen Sie diese mittels *drag&drop* in den *Access Data View*:
 - a. *Root/Objects/ENVIRONEMENT/CP-F-LINEAR/Objects@CP-F-LINEAR/SPS_A/Objects@SPS_A/Outputs@SPS_A/RFID_Read*
 - b. *Root/Objects/ENVIRONEMENT/CP-F-LINEAR/Objects@CP-F-LINEAR/SPS_A/Objects@SPS_A/Outputs@SPS_A/RFID_Write*
 - c. *Root/Objects/ENVIRONEMENT/CP-F-LINEAR/Objects@CP-F-LINEAR/SPS_A/Objects@SPS_A/Outputs@SPS_A/RFID_WriteVal*
 - d. *Root/Objects/ENVIRONEMENT/CP-F-LINEAR/Objects@CP-F-LINEAR/SPS_A/Objects@SPS_A/Inputs@SPS_A/RFID_ReadVal*


Achten Sie darauf dass sie Inputs und Outputs von Outputs/Inputs@SPS_A nehmen und nicht von Objects@SPS_A.




#	Server	Node Id	Display Name	Value	Datatype
1	CirosStudio...	NS3 String OUT:SPS_A RFID_Read	RFID_Read	true	Boolean
2	CirosStudio...	NS3 String OUT:SPS_A RFID_Write	RFID_Write	true	Boolean
3	CirosStudio...	NS3 String OUT:SPS_A RFID_WriteVal	RFID_WriteVal	14	Double
4	CirosStudio...	NS3 String INP:SPS_A RFID_ReadVal	RFID_ReadVal	14	Double

Abbildung 12: Datenvariablen vom Modell *DigitalTwin*

5. Setzen Sie den *Value* von *RFID_Read* und *RFID_Write* auf *true* sowie *RFID_WriteVal* auf 14. Warten Sie, bis sich der Werkstückträger auf der Lesestation befindet und auf die Freigabe für die Weiterfahrt wartet. Für das Schreiben der RFID-ID setzen Sie folgende Aktionen:

 OST Ostschweizer Fachhochschule	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 13 von 24

- a. Setzen Sie zuerst *RFID_Write* auf *false*, dann wieder auf *true* zurück.
 - b. Setzen Sie weiters *RFID_Read* auf *false*, dann wieder auf *true* zurück. Nun sollten *RFID_WriteVal* und *RFID_ReadVal* die gleichen Werte haben (siehe oben: Abbildung 12).
 - c. Setzen Sie für die Freigabe der Weiterfahrt *RFID_Write* und *RFID_Read* auf *false*.
6. Testen Sie *Condition Monitoring*: Öffnen Sie mit  BlueJ aus dem Projektordner ...*HLF**DigitalTwin* die Projektvorlage *BlueJ*.
 7. Kompilieren Sie die Klassen *ConMon* und *OPCUAClientAPI*.
 8. Starten die Anwendung *ConMonitoring* in der Klasse *ConMon* mit der Methode *main*.

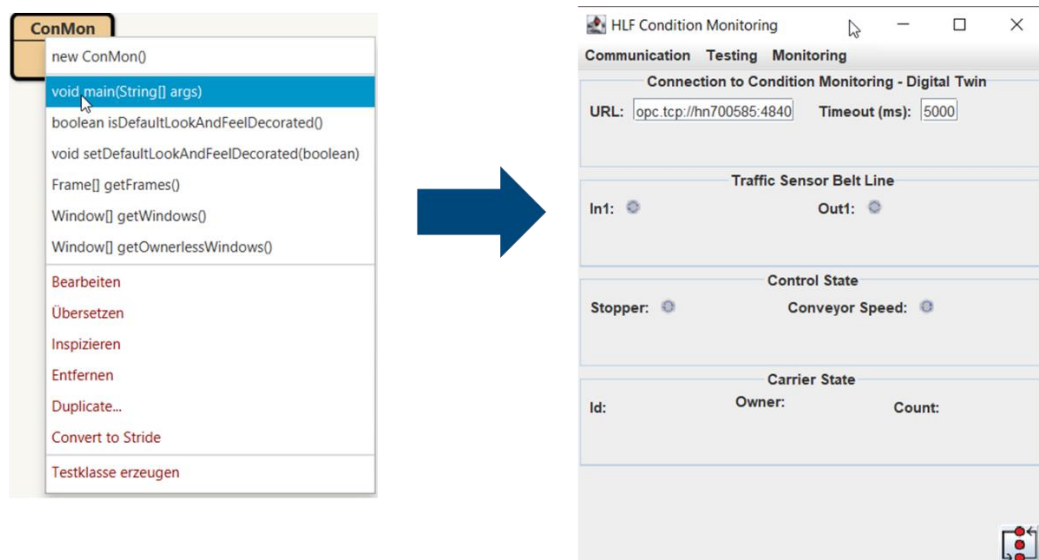









Abbildung 13: Start der Anwendung *ConditionMonitoring* mit der Methode *main*

9. Setzen Sie in der Anwendung die URL und starten Sie anschliessend die Verbindung mit  *Connect-Button*. Bei erfolgreicher Verbindung wird der  *Connect-Button* auf grün gesetzt.
10. *Testen Sie die Überwachung unter Menu Monitoring/Start*: Überprüfen Sie den Zustand der *Lichtsensoren (In1 und Out1)*, *Stopper* und die bereits gesetzte *RFID-ID (Id)*. Setzen Sie die Bandgeschwindigkeit unter *Menu Testing/Conveyor Half* auf *halb*.
11. Setzen Sie eine ungerade RFID-ID (siehe oben unter Punkt 4.) und überprüfen sie den neuerlich den Owner.

 OST Ostschweizer Fachhochschule	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 14 von 24

3.4. Übung 4: MApp mit Devices

Im Modul „Vertikale Vernetzung“ der HLF wird die HLF-Anwendung [Manufacturing Serviceapplikation](#) (kurz MApp) beschrieben. Probieren Sie die Anwendung zusammen mit dem digitalen Zwilling aus:

1. Öffnen Sie dazu mit  CIROS aus dem Projektordner ...*HLF*\Devices die Projektvorlage *Devices.modx*.
2. Starten Sie die Simulation: Wählen Sie unter Menu *Extras* die Option  *OPC-UA-Server* aus, und starten Sie die Simulation unter  *Simulation* mit *Start/Stop*. Anschliessend klicken Sie im Modell auf den  Source-Button, um den Werkstückträger mit dem Produkt (engl. *Carrier*) zu generieren und auf die Vorbereitungsstation zu positionieren.

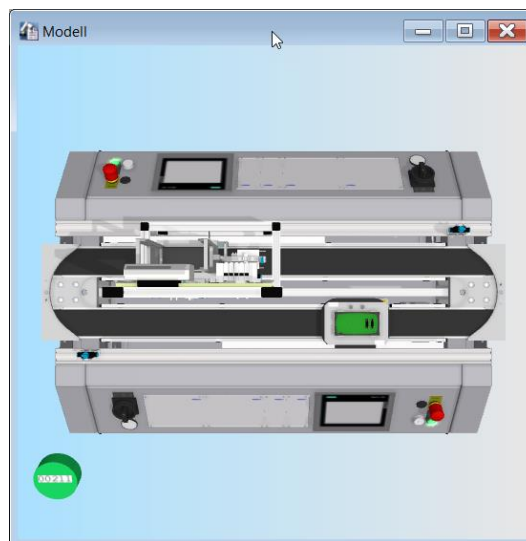



Abbildung 14: Simulationsmodell *Devices*

3. Passen Sie die Konfigurationsdateien an: Öffnen Sie die Konfigurationsdatei *config.properties* aus dem Projektordner ...*HLF*\Devices*BlueJ*\data*preparationDevice* mit einem Editor und tragen Sie die [URL](#) ein. Anschliessend speichern Sie die Datei mit dem neuen Eintrag. Führen Sie den gleichen Eintrag von der Konfigurationsdatei *config.properties* durch, die sich im Projektordner ... data*proessingDevice* und wenn vorhanden, im Projektordner ... data*executionDevice* befinden.

 OST Ostschweizer Fachhochschule	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 15 von 24

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="name">Camera</entry>
  <entry key="url">opc.tcp://[REDACTED]:4840</entry>
</properties>
```

Abbildung 15: Eintrag der [URL](#) in der Konfigurationsdatei *config.properties*.

4. Testen Sie die Anwendung *MApp*: Öffnen Sie mit  BlueJ aus dem Projektordner *...\HLF\Devices* die Projektvorlage *BlueJ*.
5. Kompilieren Sie die Klasse *MAppView*, *MAppModel*, *Device* und *Order*.
6. Starten die Anwendung *MApp* in der Klasse *MAppView* mit der Methode *main*.

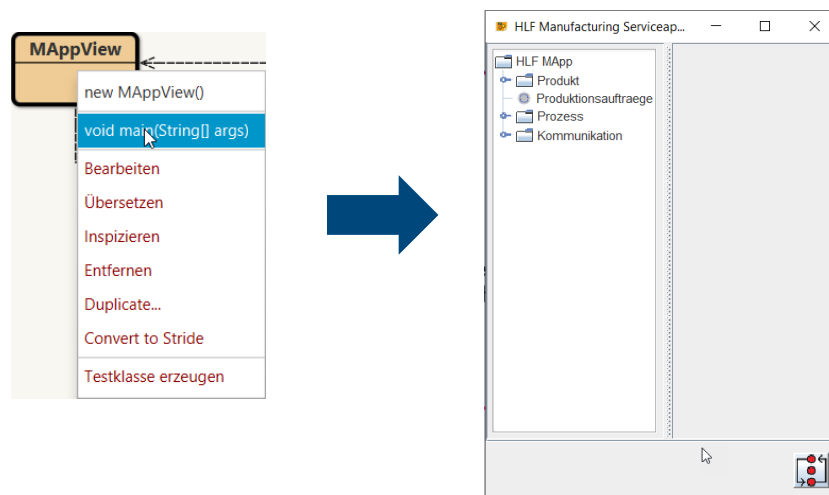




Abbildung 16: Start der Anwendung *MApp* mit der Methode *main*

7. Starten Sie die Verbindung mit  *Connect-Button*. Bei erfolgreicher Verbindung wird der  *Connect-Button* auf grün gesetzt.
8. Fügen einen neuen Produktionsauftrag hinzu:

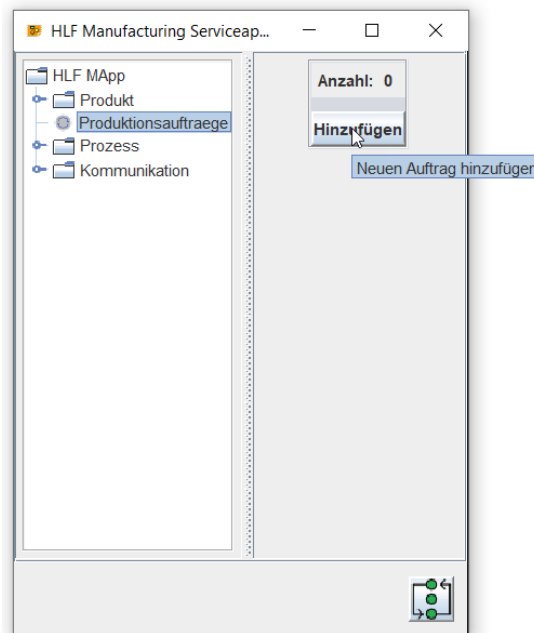


Abbildung 17: Produktionsauftrag hinzufügen

- Öffnen Sie den Produktionsauftrag, tragen Sie unter *RFID-No* 14 ein und schreiben Sie den Wert auf den virtuellen Transponder mit Button *Schreiben*. Dafür muss im CIROS ein Carrier am RFID-Reader sein.

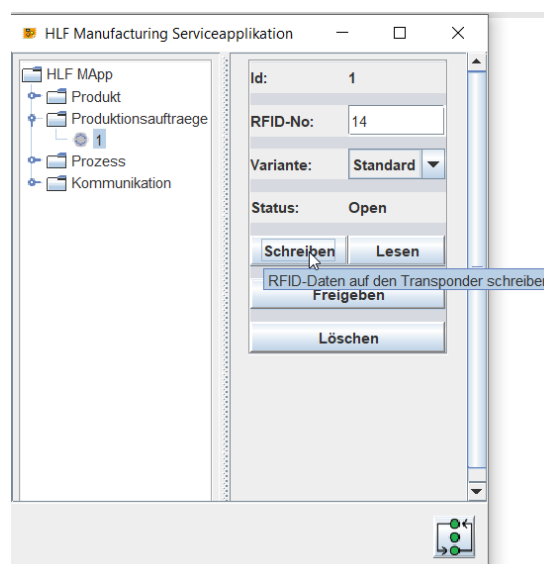




Abbildung 18: Schreiben der RFID-No

- Starten Sie auf ihrer Entwicklungsumgebung den Testclient  *Unified Automation UA Expert* und überprüfen Sie die Datenvariablen für die [RFID-No](#) auf deren Korrektheit.

 OST Ostschweizer Fachhochschule	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 17 von 24

11. Geben Sie den Produktionsauftrag mit *Freigeben* frei und öffnen Sie anschliessend den Ordner *Prozess*. Mit Button *Neuer Auftrag* wird der definierte Auftrag ausgelöst und auf dem digitalen Zwilling durchgeführt, sofern der [Produktionsablauf](#) in der Klasse `MAppModel` implementiert ist.

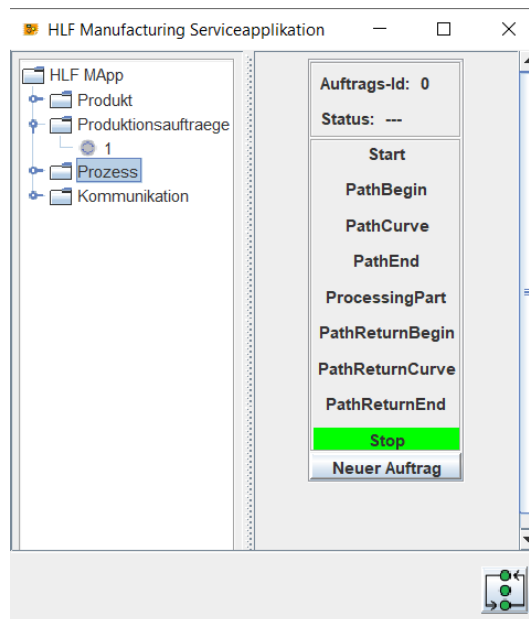


Abbildung 19: Prozessansicht von Mapp

4. Aufgabe

In dieser Aufgabe wird die Implementierung der Manufacturing Serviceapplikation (kurz MApp) vervollständigt und zeitgleich eine **virtuelle Inbetriebnahme** von zwei HLF-Stationen für Tests durchgeführt. Im nächsten Schritt kann in der HLF die **reale Inbetriebnahme** unter Verwendung der finalen MApp durchgeführt werden.

4.1. Aufgabenbeschreibung

Der HLF soll künftig eine **Manufacturing Serviceapplikation** zur Verfügung stehen: Dabei werden zwei Stationen (Devices) durch passive Werkstückträgerumlenkung zu einer Produktionseinheit (Plant) zusammengestellt. Auf der der Vorbereitungs-Station (Preparation-Device) wird der Werkstückträger (Carrier) für die Bearbeitung vorbereitet und auf der Bearbeitungs-Station (Processing-Device) wird das Werkstück bearbeitet.

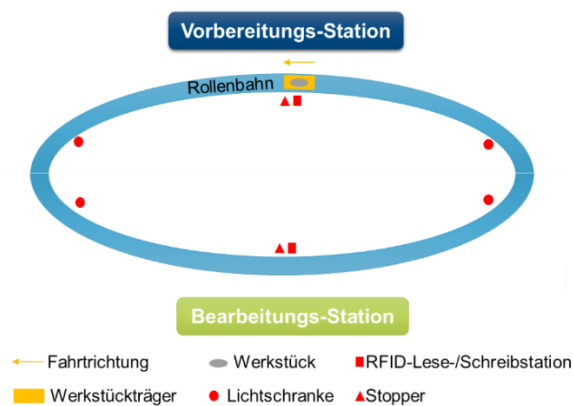


Abbildung 20: HLF-Anlage mit zwei Stationen

Die Anlage lässt sich logisch in Vorbereitung, Bearbeitung, Strecke und Rückführstrecke mit Anfang, Kurve und Ende unterteilen.

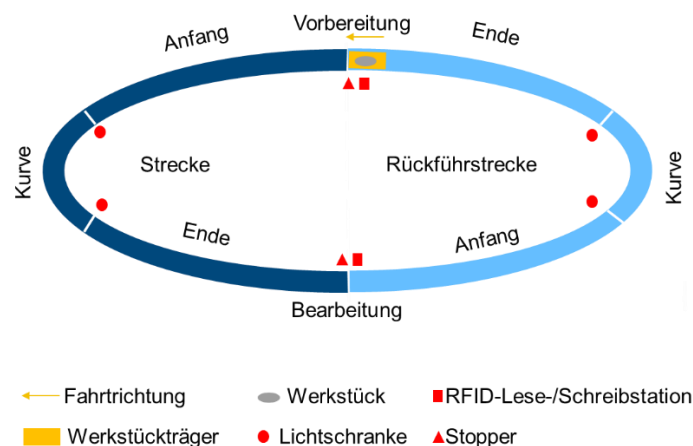



Abbildung 21: Logische Unterteilung der HLF-Anlage

4.2. Virtuelle Inbetriebnahme

Wählen Sie für die zu lösende Aufgabe eine unten angeführte Anlage zur Implementierung von MApp aus:

Anlage	Stationsnamen	Beschreibung
Magazin	CAM MAGBACK	Auflegen der Oberschale
Bohren	DRILL TURN	Bohren der Unterschale (links, rechts, beidseitig)
Presse	MPRESS OUTPUT	Oberschale andrücken

- Öffnen Sie mit  BlueJ aus dem Projektordner ...*HLF*\Devices\... die gewünschte Projektvorlage *BlueJ*.
- Öffnen Sie die Klasse `MAppModel` und navigieren Sie zur Methode `processOrder`: Diese Methode bildet den in Abbildung 22 beschriebenen [Zustandsautomaten](#) für MApp ab.
- Studieren Sie in Abbildung 22 den Ablauf des Zustandsautomaten und implementieren Sie möglichen Zustände, wobei `doStart` und `doStop` schon finalisiert sind.
- Testen Sie die Implementierung indem Sie die Anwendung *MApp* in der Klasse `MAppView` mit der Methode `main` starten. Folgen Sie den Anweisungen unter [Übung 4](#).

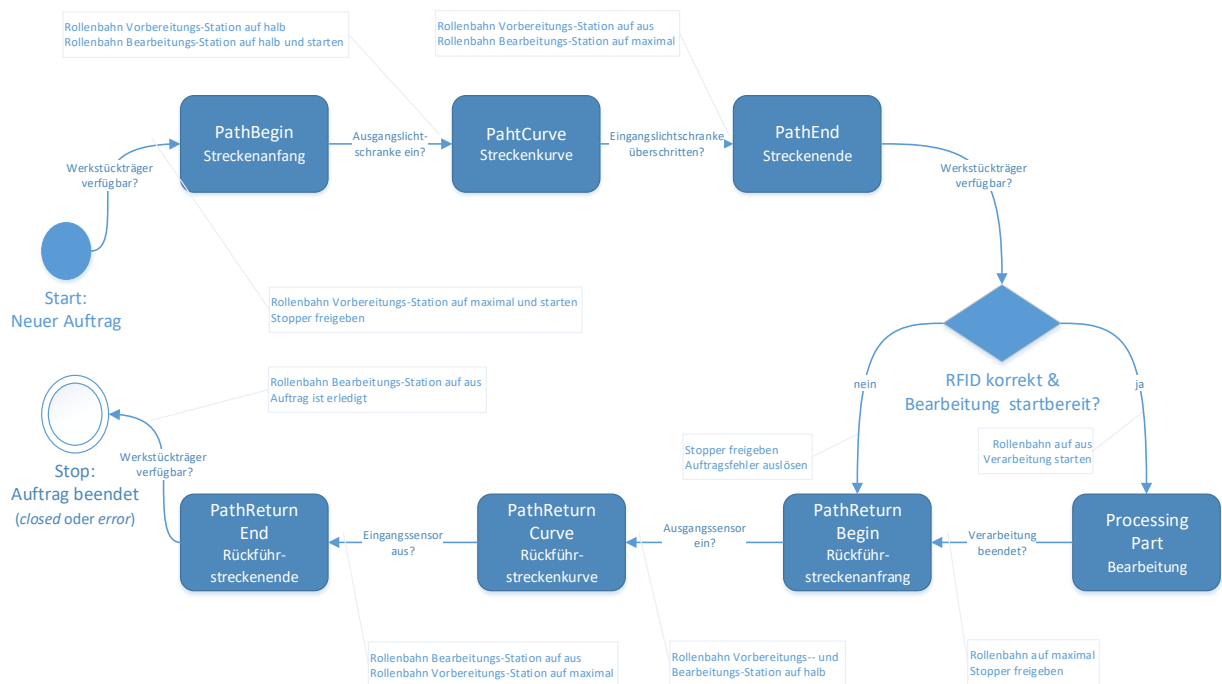




Abbildung 22 Zustandsautomat Mapp

	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 20 von 24

4.3. Reale Inbetriebnahme

Die finalisierte *MApp* kann nach der virtuellen Inbetriebnahme mit *Ciros Studio* in der HLF in Betrieb genommen und getestet werden.

 OST Ostschweizer Fachhochschule	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 21 von 24

5. Anhang

In der Klasse [MAppModel](#) stehen die Methoden `getExecutionDevice`, `getPreparationDevice` und `getExecutionDevice` für die Stationen (Devices) zur Verfügung, wobei letzteres nur die Anlage *Drehen* benötigt.

In der Klasse [Device](#) stehen Methoden für die Abfrage und das Setzen der Stationszustände zur Verfügung. Der in den Methoden übergebene *String* ist ein *Key* welcher in dem File „*data/****device/data-points.properties*“ hinterlegt ist.


Mit der Methode `getOrder` in der Klasse [Order](#) kann der aktuelle Auftrag abgefragt werden.

5.1. Klasse Device

Methods Summary	
boolean	<u><code>readBoolean(String name)</code></u> returns state from node name
void	<u><code>writeBoolean(String name, boolean value)</code></u> write state from node name
int	<u><code>readInteger(String name)</code></u> returns value from node name
int	<u><code>writeInteger(String name, int value)</code></u> write state from node name
int	<u><code>readRFID()</code></u> returns RFID-Value from carrier
void	<u><code>writeRFID(int rfid)</code></u> write RFID-Value to carrier

5.2. Klasse Order

Nested Class Summary	
static enum	<u><code>Order.State</code></u> Enum Constants: Closed, Error, InProcess, Open, Released

 OST Ostschweizer Fachhochschule	Hybride Lernfabrik (HLF)	Autoren: Robert Schöch, Robin Kuhn
		Datum: 4.12.2024
		Version: 1.2
		Seite: 22 von 24

Methods Summary	
Order.State	<u>getState</u>
	get order status
void	<u>setState(Order.State state)</u>
	set order status

5.3. Klasse MAppModel

Nested Class Summary	
static enum	<u>MAppModel.State</u>
	Enum Constants: PathBegin, PathCurve, PathEnd, PathReturnBegin, PathReturnCurve, PathReturnEnd, ProcessingPart, Start, Stop

Methods Summary	
Device	<u>getExecutionDevice()</u>
	returns execution device
Device	<u>getPreparationDevice()</u>
	returns preparation device
Device	<u>IntegerProcessingDevice()</u>

5.4. Unified Automation UA Expert

Unified Automation UA Expert

Unified Automation [UA Expert](#) ist ein OPC UA Testclient, der alle OPC UA Funktionen unterstützt. Dieser kann zum Testen von OPC UA Servern oder für die Entwicklung von OPC UA Client-Anwendungen verwendet werden.

Wenn die *NodeId* bekannt ist kann über Rechtsklick im «Data Access View» Add Custom Node dieser Node einfach hinzugefügt werden wie in Abbildung 23 gezeigt.

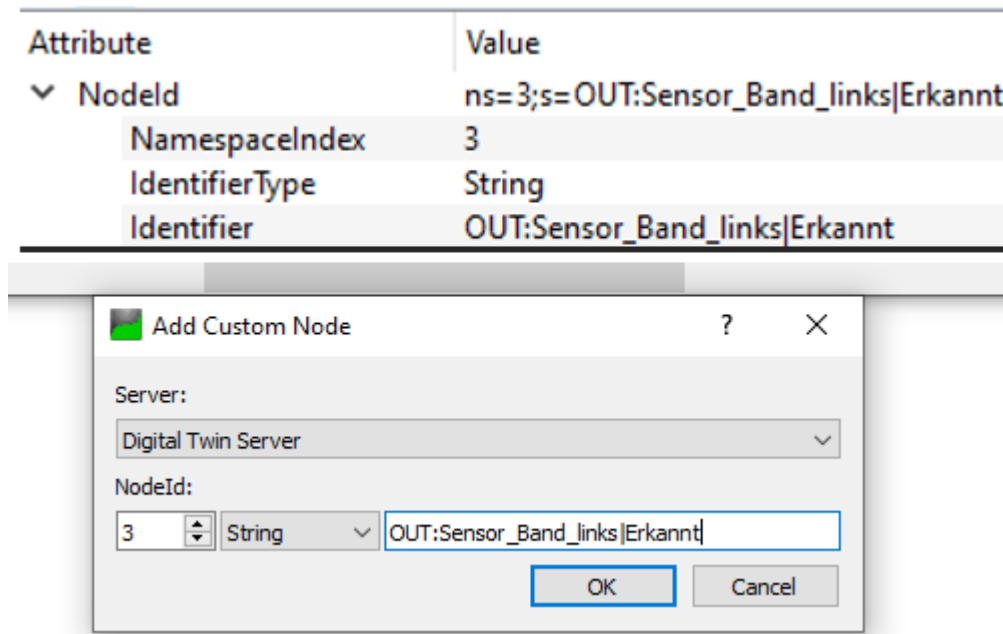


Abbildung 23 UaExpert Node über NodeId einfügen

5.5. CIROS Studio

Version

Die für dieses Dokument verwendete CIROS Version ist 7.3.3. Aus Erfahrung kann sich der in Abbildung 23 angezeigte *Address Space* trotz unveränderter *NodeId* unterscheiden.

Objekte löschen

Während der Simulation können Objekte gelöscht werden. Dies wird am Beispiel Magazin gezeigt, bei dem die Rückschale wieder entfernt werden kann. Dafür wird wie in Abbildung 24 gezeigt, das gewünschte Objekt ausgewählt und mit der «del» -Taste gelöscht. Anschliessend kann in einem nächsten Produktionsauftrag eine neue Rückschale angebracht werden.

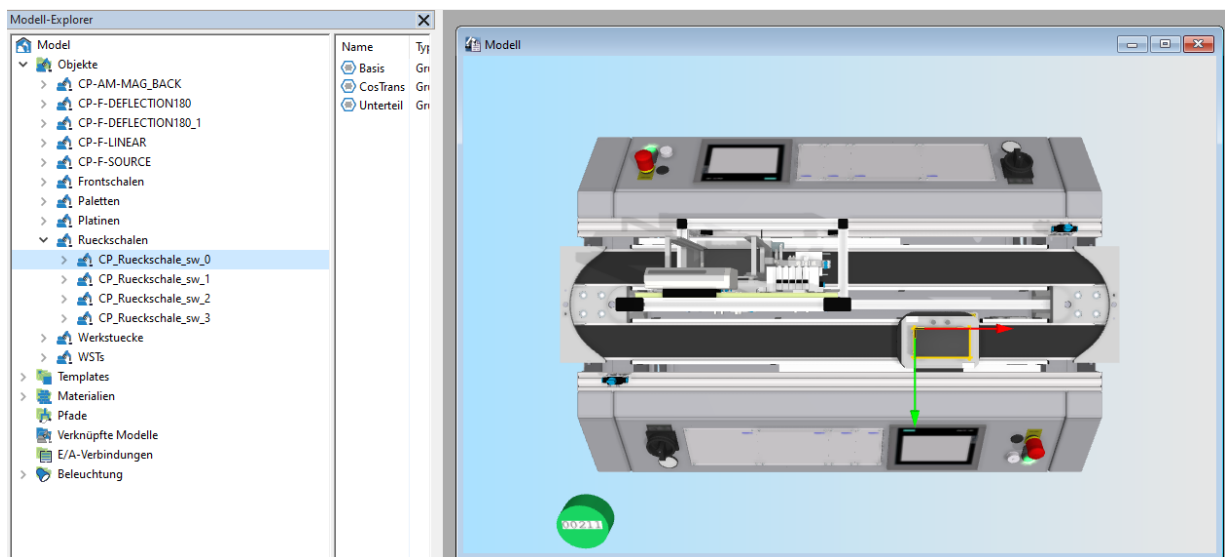


Abbildung 24 Objekt in Simulation auswählen