



**SIEMENS**



**Lern-/Lehrunterlage**

Siemens Automation Cooperates with Education (SCE) | Ab NX MCD V12/TIA Portal V15.0

**DigitalTwin@Education Modul 150-006**  
Signalerstellung für ein dynamisches 3D-Modell im CAE-System Mechatronics Concept Designer

[siemens.de/sce](https://siemens.de/sce)

**SIEMENS**

Global Industry  
Partner of  
WorldSkills  
International



worldskills

## Passende SCE Trainer Pakete zu dieser Lern-/Lehrunterlage

### SIMATIC STEP 7 Software for Training (inkl. PLCSIM Advanced)

- **SIMATIC STEP 7 Professional V15 - Einzel-Lizenz**  
Bestellnr.: 6ES7822-1AA05-4YA5
- **SIMATIC STEP 7 Professional V15 - 6er Klassenraum-Lizenz**  
Bestellnr.: 6ES7822-1BA05-4YA5
- **SIMATIC STEP 7 Professional V15 - 6er Upgrade-Lizenz**  
Bestellnr.: 6ES7822-1AA05-4YE5
- **SIMATIC STEP 7 Professional V15 - 20er Studenten-Lizenz**  
Bestellnr.: 6ES7822-1AC05-4YA5

### Software SIMATIC WinCC Engineering/Runtime Advanced im TIA Portal

- **SIMATIC WinCC Advanced V15 - 6er Klassenraum-Lizenz**  
6AV2102-0AA05-0AS5
- **Upgrade SIMATIC WinCC Advanced V15 - 6er Klassenraum-Lizenz**  
6AV2102-4AA05-0AS5
- **SIMATIC WinCC Advanced V15 - 20er Studenten-Lizenz**  
6AV2102-0AA05-0AS7

### NX V12.0 Educational Bundle (Schulen, Hochschulen, nicht für betriebliche Ausbildungsstätten)

- **Ansprechpartner:** [academics.plm@siemens.com](mailto:academics.plm@siemens.com)

## Weitere Informationen rund um SCE

[siemens.de/sce](https://www.siemens.de/sce)

## Verwendungshinweis

Die SCE Lern-/Lehrunterlage für die durchgängige Automatisierungslösung Totally Integrated Automation (TIA) wurde für das Programm “**Siemens Automation Cooperates with Education (SCE)**“ speziell zu Ausbildungszwecken für öffentliche Bildungs- und F&E-Einrichtungen erstellt. Siemens übernimmt bezüglich des Inhalts keine Gewähr.

Diese Unterlage darf nur für die Erstausbildung an Siemens Produkten/Systemen verwendet werden. D. h. Sie kann ganz oder teilweise kopiert und an die Auszubildenden/Studierenden zur Nutzung im Rahmen deren Ausbildung/Studiums ausgehändigt werden. Die Weitergabe sowie Vervielfältigung dieser Unterlage und Mitteilung Ihres Inhalts ist innerhalb öffentlicher Aus- und Weiterbildungsstätten für Zwecke der Ausbildung oder im Rahmen des Studiums gestattet.

Ausnahmen bedürfen der schriftlichen Genehmigung durch Siemens. Alle Anfragen hierzu an [scsupportfinder.i-ia@siemens.com](mailto:scsupportfinder.i-ia@siemens.com).

Zu widerhandlungen verpflichten zu Schadensersatz. Alle Rechte auch der Übersetzung sind vorbehalten, insbesondere für den Fall der Patentierung oder GM-Eintragung.

Der Einsatz für Industriekunden-Kurse ist explizit nicht erlaubt. Einer kommerziellen Nutzung der Unterlagen stimmen wir nicht zu.

Wir danken der HS Darmstadt, besonders Herrn Heiko Webert, M.Sc. und Herrn Prof. Dr.-Ing. Stephan Simons und allen weiteren Beteiligten für die Unterstützung bei der Erstellung dieser SCE Lern-/Lehrunterlage.

# Inhaltsverzeichnis

1	Zielstellung.....	7
2	Voraussetzung .....	7
3	Benötigte Hardware und Software .....	8
4	Theorie .....	9
4.1	Kommunikation mit externen Quellen.....	9
4.2	Signal-Eigenschaften in Mechatronics Concept Designer.....	10
5	Aufgabenstellung .....	13
6	Planung .....	13
7	Strukturierte Schritt-für-Schritt-Anleitung.....	14
7.1	Erstellen der Signale für das dynamische Modell .....	15
7.2	Erzeugen einer Signal-Verbindung zwischen virtueller SPS und digitalem Zwilling .....	31
7.3	Testen des digitalen Zwillings mit der virtuellen SPS.....	37
8	Checkliste – Schritt-für-Schritt-Anleitung.....	38
9	Weiterführende Informationen .....	39

# Abbildungsverzeichnis

Abbildung 1: Überblick benötigter Soft- und Hardwarekomponenten in diesem Modul .....	8
Abbildung 2: Anwendung "Mechatronics Concept Designer" in NX mit Kennzeichnungen zur Erläuterungen der Bereiche im Text.....	10
Abbildung 3: Befehlssuche im NX-Menü, in Orange umrandet.....	14
Abbildung 4: Öffnen einer Baugruppe in NX.....	15
Abbildung 5: Hinzufügen von Parametern dynamischer Eigenschaften für Signale im Signaladapter .....	16
Abbildung 6: Lesen/Schreiben Eigenschaft eines Parameters .....	17
Abbildung 7: Erstellen eines passenden Signals für einen Parameter .....	18
Abbildung 8: Definition einer Formel zwischen Signal und Parameter .....	19
Abbildung 9: Erstellen eines Ausgabe-Signals für einen Lichttaster .....	20
Abbildung 10: Formel für das Signal des Lichttaster Systems "csLightSensorCylinder".....	21
Abbildung 11: Erstellen eines Geschwindigkeitssignals vom Datentyp "double" .....	24
Abbildung 12: Erstellen eines Parameters für eine Transportfläche .....	26
Abbildung 13: Erstellen des Signaladapters "saSortingPlant" .....	27
Abbildung 14: Erstellung einer neuen Symboltabelle für den Signaladapter initiieren .....	28
Abbildung 15: Erstellung einer neuen Symboltabelle für Signaladapter abschließen.....	29
Abbildung 16: Abschließen der Symbolzuordnung für Signaladapter .....	30
Abbildung 17: Auswahl einer Signalzuordnung über PLCSIM Advanced.....	32
Abbildung 18: Variablen aus PLCSIM Advanced-Instanz für Signalzuordnung freigeben .....	33
Abbildung 19: Zuordnen eines MCD-Signals zu einem externen Signal.....	34
Abbildung 20: Alle Signale durch automatische Zuordnung verbinden .....	35
Abbildung 21: Signalzuordnung wieder auftrennen.....	36
Abbildung 22: Signalzuordnung zwischen dynamischen Modell und virtueller SPS bestätigen .....	36

# Tabellenverzeichnis

Tabelle 1: Checkliste der "Signalerstellung für ein dynamisches 3D-Modell im CAE-System Mechatronics Concept Designer" .....38

# Signalerstellung für ein dynamisches 3D-Modell im CAE-System Mechatronics Concept Designer

## 1 Zielstellung

In Modul 4 der Workshop-Reihe zu DigitalTwin@Education haben Sie das statische 3D-Modell einer Sortieranlage vollkommen eigenständig konstruiert. Als Resultat entstand eine Baugruppe, in welcher die nötigen Einzelkomponenten der Sortieranlage eingefügt und im Raum korrekt positioniert sind. Aufbauend dazu hat sich Modul 5 mit der Dynamisierung des 3D-Modells beschäftigt. Durch das Zuweisen von physikalische Eigenschaften können die Komponenten der Sortieranlage miteinander interagieren.

Damit der digitale Zwilling im Zusammenspiel mit einer virtuellen SPS funktioniert, braucht es im letzten Schritt eine Verbindung zwischen dem Mechatronics Concept Designer (MCD) und PLCSIM Advanced, womit die SPS mit dem Automatisierungsprogramm simuliert wird. Das Ziel dieses Moduls liegt darin, Signale zu erstellen und diese zu beiden Programmen passend zuzuordnen. Anschließend sollen Sie mit dem Automatisierungsprogramm aus Modul 1 der DigitalTwin@Education-Reihe die korrekte Funktionsweise Ihres digitalen Zwillings validieren.

## 2 Voraussetzung

Für dieses Modul benötigen Sie das Wissen über die dynamischen Eigenschaften des Modells, welche in Modul 5 verwendet wurden. Außerdem sollten Sie aus den Modulen 1 - 2 die Funktionsweise des Automatisierungsprogramms verstanden haben, da Sie in diesem Modul davon wieder Gebrauch machen sollen.

### 3 Benötigte Hardware und Software

Für dieses Modul werden folgende Komponenten benötigt:

- 1 **Engineering Station:** Voraussetzungen sind Hardware und Betriebssystem (für weitere Informationen: siehe ReadMe/Liesmich auf den TIA Portal Installations-DVDs sowie im NX-Softwarepaket)
- 2 **Software SIMATIC STEP 7 Professional im TIA Portal** – ab V15.0
- 3 **Software SIMATIC WinCC Runtime Advanced im TIA Portal** – ab V15.0
- 4 **Software SIMATIC S7-PLCSIM Advanced** – ab V2.0
- 5 **Software NX mit Erweiterung Mechatronics Concept Designer** – ab V12.0

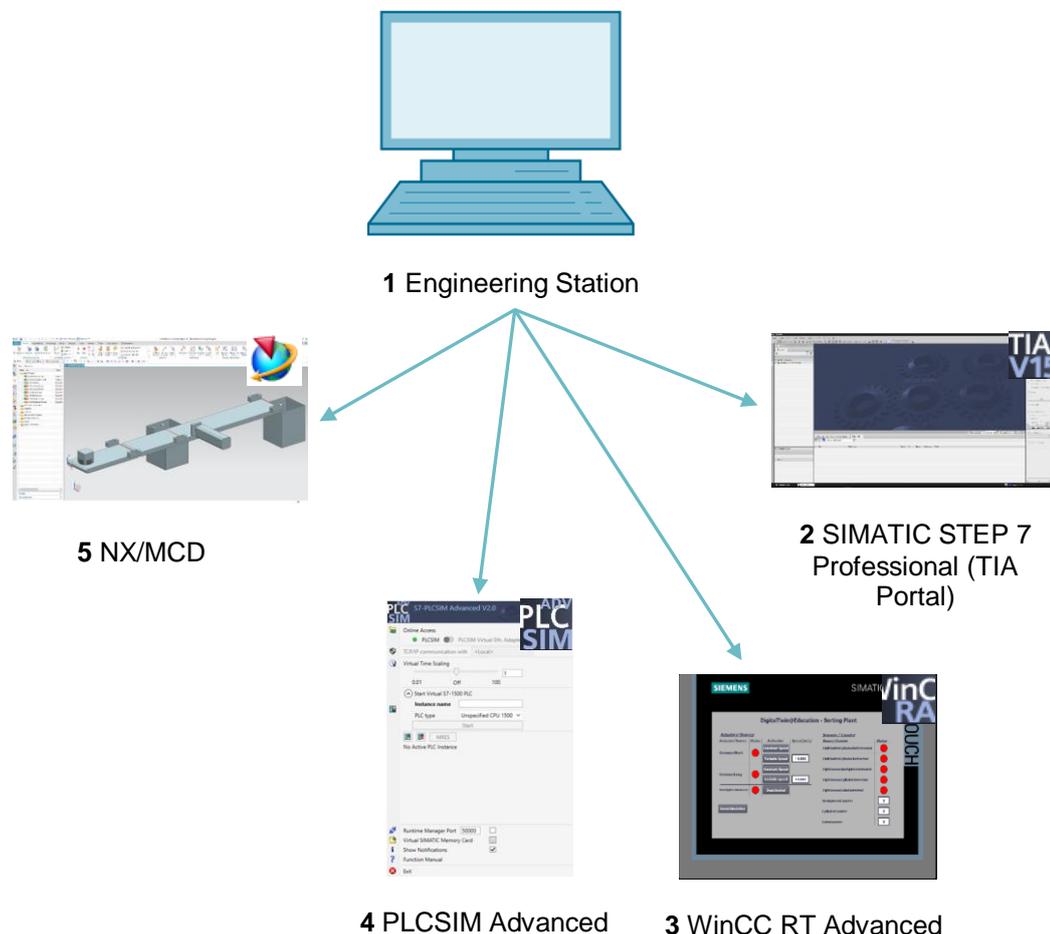


Abbildung 1: Überblick benötigter Soft- und Hardwarekomponenten in diesem Modul

Anhand von [Abbildung 1](#) wird deutlich, dass die Engineering Station die einzige Hardwarekomponente des Systems darstellt. Die restlichen Komponenten basieren ausschließlich auf Software.

## 4 Theorie

### 4.1 Kommunikation mit externen Quellen

In Modul 5 dieser Workshop-Reihe haben Sie die dynamischen Eigenschaften erstellt und mithilfe der Laufzeitüberwachung in Mechatronics Concept Designer auf Funktionalität überprüft. Zielführend bei einem digitalen Zwilling ist jedoch, eine Verbindung zu einer Steuerung herzustellen, sodass die dynamischen Eigenschaften im MCD durch die Steuerung geändert und die Resultate aus dem MCD der Steuerung zur Verfügung gestellt werden.

MCD bietet dabei diverse Möglichkeiten an, um mit externen Programmen zu kommunizieren (siehe [Kapitel 9](#), Link [1]). Darunter zählen:

- Kommunikation mit MATLAB über das MATLAB-Protokoll
- Verbindung zu einem OPC-Server (ab MCD V12.01 auch zu einem OPC UA-Server)
- S7-Kommunikation über PLCSIM Advanced oder direkt mittels PROFINET-Protokoll
- Anbindung zu einem geteilten Speicher (*Shared Memory*) für bspw. SIMIT
- TCP/UDP-Verbindungen

In diesem Modul soll die Kommunikation zu PLCSIM Advanced konfiguriert werden, wie bereits in den Modulen 1 – 3 dieser Workshop-Reihe angewandt. Dadurch lässt sich eine virtuelle SPS mit NX/MCD verbinden.

## 4.2 Signal-Eigenschaften in Mechatronics Concept Designer

Die Kommunikation zu externen Programmen funktioniert in der NX-Erweiterung Mechatronics Concept Designer über die Definition und Zuordnung von Signalen.

Die Arbeitsoberfläche von Mechatronics Concept Designer ist in [Abbildung 2](#) dargestellt. Zum Öffnen dieser Anwendung in NX suchen Sie über die bereits bekannte Befehlsuche, im oberen rechten Bildschirmteil, nach der Anwendung "**Mechatronics Concept Designer**".

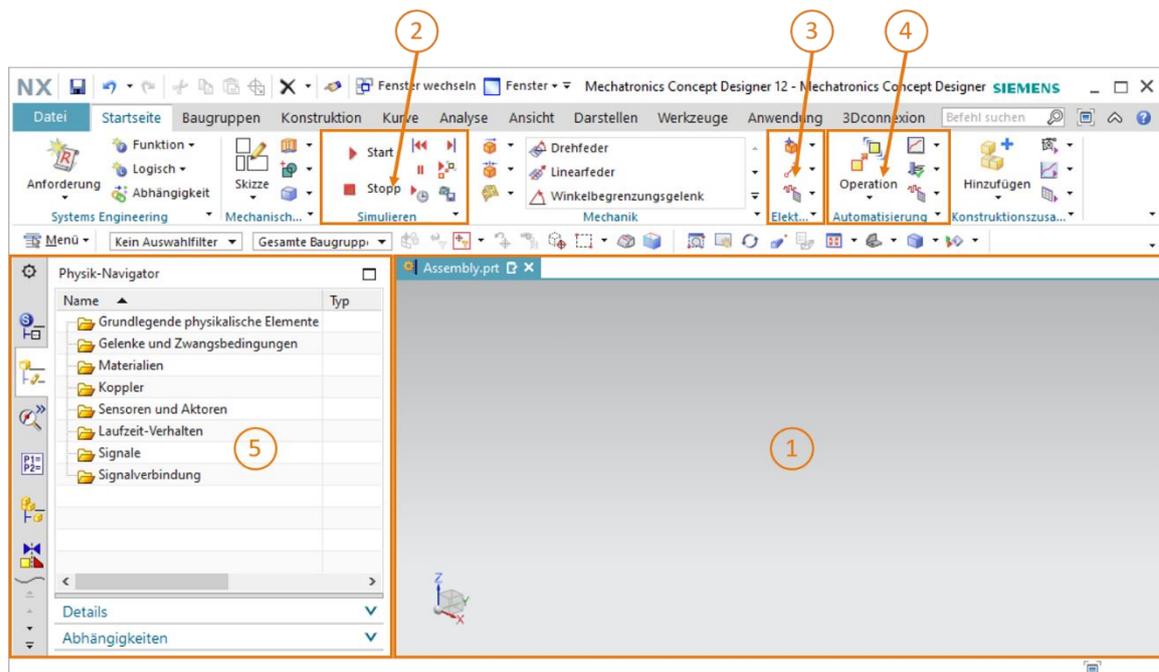


Abbildung 2: Anwendung "Mechatronics Concept Designer" in NX mit Kennzeichnungen zur Erläuterungen der Bereiche im Text

Zur Definition von Signalen und zum Testen des digitalen Zwillings finden in dieser Anwendung folgende Fenster Verwendung:

- Im zentralen Bildschirm (siehe [Abbildung 2](#), Bereich 1) befindet sich die dreidimensionale Arbeitsoberfläche, worüber Sie die Funktionsweise Ihres dynamischen 3D-Modells im Zusammenspiel mit der virtuellen SPS während einer Simulation nachverfolgen können.
- Im mittleren Teil der Menüleiste (siehe [Abbildung 2](#), Bereich 2) können Sie die Simulation Ihres Modells steuern. Von diesen Funktionen werden Sie in [Kapitel 7.3](#) Gebrauch machen.

- Die Signal-Eigenschaften aus dem Bereich der Elektrik finden Sie in der Menüleiste direkt neben den dynamisch-mechanischen Eigenschaften (siehe [Abbildung 2](#), Bereich 3). Hier können Sie Signale und Tabellen erstellen. Einige der Befehle werden im Folgenden kurz erläutert.

- Über den Befehl **Signal**  können Sie in Ihrem Modell ein Signal erstellen, um physikalische Eigenschaften eines Objekts durch einen Laufzeitausdruck zu kontrollieren. Ein Laufzeitausdruck ist ein nicht-statischer Wert, welcher sich zur Laufzeit einer Simulation ändern kann. Dieser Ausdruck ist entweder intern an MCD gekoppelt oder er wird durch Anbindung an ein Signal aus einer externen Quelle, wie PLCSIM Advanced, bestimmt.

- Mit dem Erstellen einer **Symboltabelle**  definieren Sie eine Auflistung von Symbolen, welche für eine eindeutige Benennung der Signale genutzt werden. Sie können eine Symboltabelle auch aus einer externen Quelle importieren, wie z.B. STEP 7.

- Mittels des **Signaladapters**  können Signale und Laufzeitausdrücke miteinander verknüpft werden. Es ist möglich pro Signaladapter mehrere Signale und Laufzeitausdrücke zu verwenden. Außerdem lassen sich über diesen Befehl sowohl Signale als auch Laufzeitausdrücke erstellen.

- Die Signal-Eigenschaften aus dem Bereich der Automatisierung finden sich ebenfalls in der Menüleiste von MCD (siehe [Abbildung 2](#), Bereich 4). Hier sollen sie die folgende Eigenschaft nutzen:

- Die **Signalzuordnung**  schafft eine Verknüpfung von Signalen aus MCD mit Signalen aus externen Programmen. Darunter fällt auch PLCSIM Advanced.

- Über die Ressourcenleiste auf dem linken Bildschirmteil (siehe [Abbildung 2](#), Bereich 5) im MCD ist unter anderem der Physik-Navigator abrufbar. Dort werden Ihre Signale und Verbindungen hinterlegt.



### HINWEIS

Für weitere Informationen zu den Signal-Eigenschaften in Mechatronics Concept Designer können Sie in der Online-Hilfe nach entsprechenden Einträgen recherchieren (siehe [Kapitel 9](#), Link [2]).

Hier empfiehlt es sich jedoch nach den englischen Begrifflichkeiten zu suchen, da die deutschen Beiträge nur unvollständig sind.

## 5 Aufgabenstellung

Im Folgenden sollen Sie Ihr in Modul 5 erstelltes dynamisches 3D-Modell der Sortieranlage um Signale erweitern und eine Verbindung zu einer virtuellen SPS aufbauen. Weiterhin sollen Sie das Automatisierungsprogramm aus Modul 1 der Workshop-Reihe von DigitalTwin@Education in die virtuelle SPS laden und Ihren eigenen digitalen Zwilling selbstständig validieren.

Dazu benötigen Sie erneut die NX-Anwendung Mechatronics Concept Designer (MCD). Nun sollen Sie sich jedoch vollständig auf die Anbindung Ihres dynamischen 3D-Modells zu externen Programmen konzentrieren.

## 6 Planung

Für die Zuordnung von Signalen für ein dynamisches 3D-Modell und dessen Inbetriebnahme wird das CAD-System **NX**, mindestens in der Version **V12.0** vorausgesetzt. Das Zusatzmodul **Mechatronics Concept Designer (MCD)** muss ebenfalls in NX vorhanden sein.

Sie benötigen Kenntnisse über **statische** und **dynamische 3D-Modelle**, welche Sie aus den Modulen 4 und 5 entnehmen können. Weiterhin sollten Sie Ihr Wissen über die **Funktionsweise des Automatisierungsprogramms** aus den **Modulen 1 – 2** der DigitalTwin@Education-Reihe auffrischen. Bei Unklarheiten bzgl. der Arbeitsweise der Sortieranlage sollten Sie sich insbesondere den Theorieteil im **Kapitel 4.2** aus **Modul 1** erneut ansehen.

Machen Sie sich außerdem erneut mit der **Interaktion zwischen virtueller SPS und digitalem Zwilling** aus **Modul 1** vertraut und halten Sie die Beschreibung von Modul 1 dieser Workshop-Reihe bereit, da Sie diese insbesondere in [Kapitel 7.2](#) und [7.3](#) benötigen werden.

Bei der Benennung der Signale wurde auf den "**Leitfaden Standardisierung**" von Siemens zurückgegriffen. Diesen können Sie unter dem in [Kapitel 9](#) angegebenen Link [3] finden.

## 7 Strukturierte Schritt-für-Schritt-Anleitung

Mit diesem Modul wird der Ordner "150-006\_DigitalTwinAtEducation\_NX\_dynModelSignals" bereitgestellt. Der Ordner beinhaltet drei Unterordner:

- "fullDynModel" beinhaltet das gesamte dynamische 3D-Modell der Sortieranlage aus **Modul 5**. Sie können dieses Modell als Ausgangspunkt für dieses Modul verwenden, sofern Ihre Ergebnisse aus Modul 5 unvollständig sind.
- "fullDigTwin" enthält mit dem vollständigen digitalen Zwilling die Lösung zu diesem Modul. Dies dient als Hilfestellung, falls Sie an einem Schritt nicht weiterkommen.
- "fullPlcBasic" stellt das aus **Modul 1** bekannte Automatisierungsprogramm mit integriertem HMI bereit. Dieses benötigen Sie zum Testen Ihres digitalen Zwillings.

Sofern Sie im Laufe des Moduls einen Befehl oder eine Anwendung nicht in der Entwicklungsumgebung finden, sei an dieser Stelle nochmals auf die Befehlssuche hingewiesen. Diese befindet sich, wie in [Abbildung 3](#) dargestellt, im oberen rechten Bildschirmteil der NX-Benutzeroberfläche.

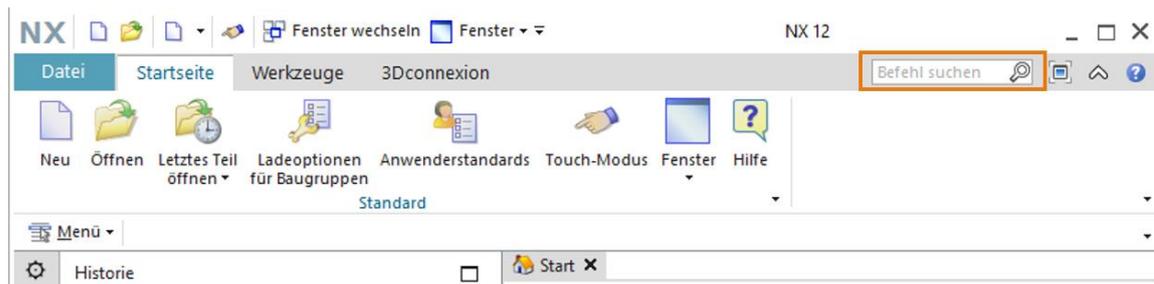


Abbildung 3: Befehlssuche im NX-Menü, in Orange umrandet

Aus den gefundenen Vorschlägen können Sie sich den passenden Befehl aussuchen. NX zeigt Ihnen dabei auch an, wo Sie den Befehl finden, sodass Sie ihn in Zukunft auch direkt aus dem Menü anwählen können.

**WICHTIG:** Mit neuen Versionen von NX verändert sich die Oberfläche und die Anordnung verschiedener Befehle in den Menüs. Außerdem kann jeder Benutzer eine benutzerdefinierte Oberfläche erstellen. Während die nachfolgenden Beschreibungen die Standard-Oberfläche von NX12.0 abbilden, kann sich das in Ihrer Version unterscheiden. **Sollten Sie daher einen Befehl nicht an den beschriebenen Positionen im Fenster finden, nutzen Sie die Befehlssuche.**

Weiterhin sollten Sie beachten, dass diese Beschreibung nur ein Lösungsvorschlag ist. Es wurde versucht ein nachvollziehbares Vorgehen zu beschreiben, womit Sie einfach Ihren digitalen Zwilling mit einer virtuellen CPU aus den Modulen 1 bis 3 interagieren lassen können.

Beachten Sie, dass bestimmte Stellen in Form von Abschnitten hervorgehoben sind. Da im Laufe dieser Beschreibung häufig auf diese Bereiche verwiesen wird, sollen diese Markierungen als Orientierungshilfe dienen.

## 7.1 Erstellen der Signale für das dynamische Modell

In diesem Kapitel sollen Sie für Ihre Sortieranlage alle nötigen Signale erstellen, welche durch eine SPS extern steuerbar sein sollen. Gehen Sie dabei wie folgt vor:

- Erstellen Sie mit Ihrem Betriebssystem eine Kopie Ihrer in Modul 5 verwendeten Modelle. Speichern Sie diese in einem neuen Ordner in Ihrem Dateisystem ab. Sofern Sie ein unvollständiges dynamisches Modell haben, können Sie auch, wie in [Kapitel 7](#) erwähnt, auf das Projekt "fullDynModel" zurückgreifen und von diesem Ordner eine Arbeitskopie erstellen.
- Starten Sie NX und warten Sie, bis sich das Programm geöffnet hat und Sie die Startseite sehen. Klicken Sie auf die Schaltfläche "Öffnen" (siehe [Abbildung 4](#), Schritt 1) und navigieren Sie zu Ihrem zuvor erstellten Ordner. Sie sehen nun die in Modul 5 verwendeten Teile. Selektieren Sie die Baugruppe "assSortingPlant", welche das vollständige dynamische 3D-Modell der Sortieranlage enthält (siehe [Abbildung 4](#), Schritt 2). Wählen Sie die Option "Teilweise geladen" aus (siehe [Abbildung 4](#), Schritt 3), damit nur die Modelle und dynamischen Eigenschaften der Einzelkomponenten der Baugruppe geladen werden, nicht aber zusätzliche Zeichnungen oder Koordinatensysteme. Bestätigen Sie Ihre Auswahl schließlich mit einem Klick auf "OK" (siehe [Abbildung 4](#), Schritt 4).

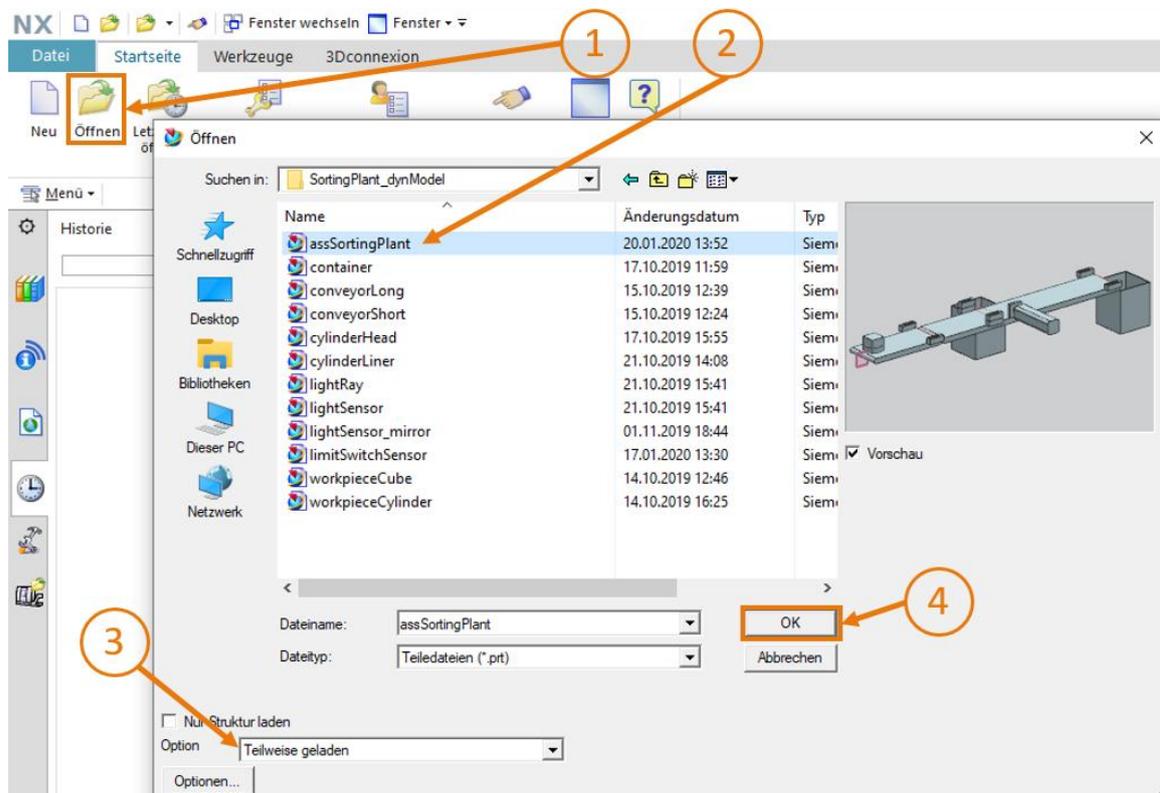


Abbildung 4: Öffnen einer Baugruppe in NX

## Abschnitt: Signalerstellung und -verknüpfung mit dem Signaladapter

→ Nachdem sich die Baugruppe in der NX-Anwendung **"Mechatronics Concept Designer"** geöffnet hat, erstellen Sie Ihr erstes Beispiel. Hierfür soll nun ein Signal zum Aktivieren der Objektquelle erstellt und verknüpft werden, welches zur Erzeugung von quaderförmigen Werkstücken benötigt wird. Zum Hinzufügen von Signalen und deren Verknüpfung mit dynamischen Eigenschaften öffnen Sie zunächst den Befehl **"Signaladapter"** aus dem Bereich **"Elektrik"**, wie in [Abbildung 5](#), Schritt 1 dargestellt. Es erscheint daraufhin das Befehlsfenster **"Signaladapter"**. Zuerst muss nun ein Parameter einer dynamischen Eigenschaft ausgewählt werden, welcher mit einem Signal verbunden werden soll. Klicken Sie hierfür unter dem Befehlsreiter **"Parameter"** auf die Schaltfläche **"Physik-Objekt auswählen"** (siehe [Abbildung 5](#), Schritt 2). Navigieren Sie in der Ressourcenleiste zum **„Physik-Navigator“** (siehe [Abbildung 5](#), Schritt 3) und wählen Sie darin als ersten Parameter die Objektquelle **"osWorkpieceCube"** aus (siehe [Abbildung 5](#), Schritt 4). Nachdem Sie diesen selektiert haben, können Sie nun im Befehlsfenster bei dem Punkt **"Parametername"** den entsprechenden Parameter wählen, welchen Sie auf ein Signal zuweisen möchten. Wählen Sie in diesem Fall den Parametername **"Aktiv"** der selektierten Objektquelle aus (siehe [Abbildung 5](#), Schritt 5). Klicken Sie auf die Schaltfläche **"Parameter hinzufügen"** (siehe [Abbildung 5](#), Schritt 6), wodurch der Parameter in diesen Signaladapter eingefügt wird.

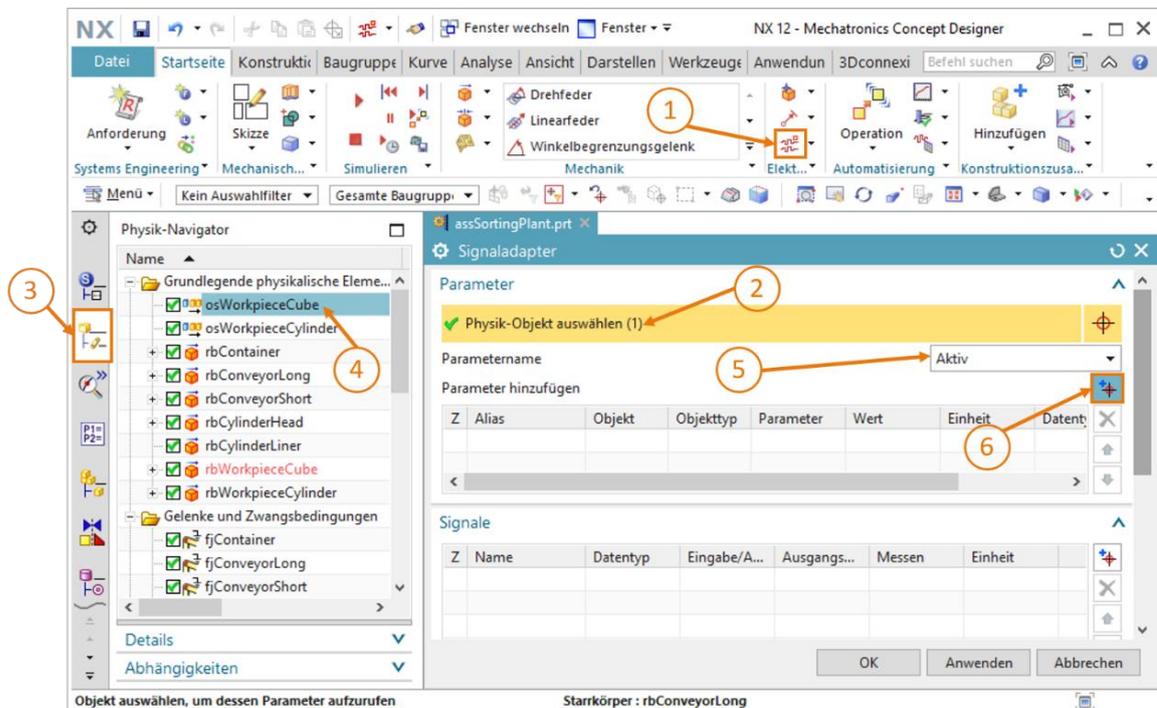


Abbildung 5: Hinzufügen von Parametern dynamischer Eigenschaften für Signale im Signaladapter

- In der Tabelle unter dem Befehlspunkt "**Parameter**" finden Sie anschließend den soeben ausgewählten Parameter. Benennen Sie dessen Alias-Name in "**paOsWorkpieceCube\_SetActive**" um (siehe [Abbildung 7](#), Schritt 1). Das Präfix "**pa**" soll hierbei für "**Parameter**" stehen, um eine klare Unterscheidung gegenüber einem Signalnamen zu haben. Klicken Sie außerdem auf das Kästchen am Anfang der Tabelle, um dem Parameter später ein Signal zuweisen zu können. Dies wird durch ein Häkchen  dargestellt. Wenn Sie im Befehlsregister "**Parameter**" nach rechts scrollen, können Sie weitere Eigenschaften des Parameters, z. B. die Eigenschaft "**Lesen/Schreiben**", die anzeigt, ob ein Parameter gelesen ("**R**" für Read) oder geschrieben ("**W**" für Write) werden kann. Der aktuelle Parameter "**paOsWorkpieceCube\_SetActive**" kann nur geschrieben werden (siehe [Abbildung 6](#), Schritt 1).

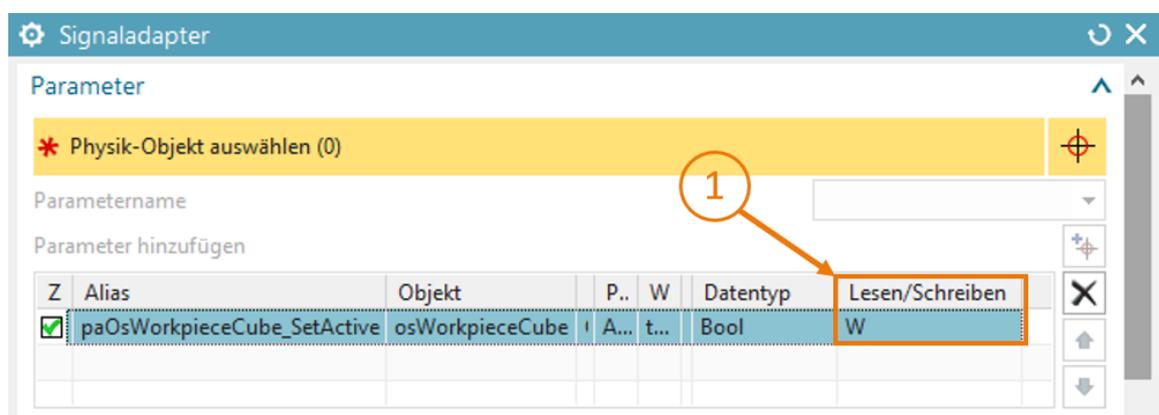


Abbildung 6: Lesen/Schreiben Eigenschaft eines Parameters

- Nun benötigen Sie noch ein zugehöriges Signal, mit dem der Parameter verbunden werden soll. Klicken Sie hierfür auf die Schaltfläche "**Hinzufügen**" unter dem Befehlsreiter "**Signale**" (siehe [Abbildung 7](#), Schritt 2). Es erscheint ein neues Signal. Passen Sie die Eigenschaften des Signals entsprechend des Parameters an. Führen Sie dazu einen Doppelklick auf den Standardnamen des Signals "**Signal\_0**" aus und benennen Sie das Signal in "**osWorkpieceCube\_SetActive**" um. Als Datentyp muss derselbe Typ gewählt werden, wie für den zugehörigen Parameter. Dies ist in diesem Fall ein "**Bool**". Der Wert der Eigenschaft "**Ein-/Ausgabe**" muss entsprechend der Eigenschaft "**Lesen/Schreiben**" des Parameters gewählt werden. Für einen Parameter, der geschrieben werden muss, muss das Signal aus Sicht des MCD ein Eingang von einer externen Quelle sein. Für einen Parameter der gelesen wird, muss das Signal ein Ausgang zu einem externen Programm sein. Da der aktuelle Parameter "**paOsWorkpieceCube\_SetActive**" geschrieben wird, muss für das Signal "**osWorkpieceCube\_SetActive**" demnach der Wert "**Eingabe**" gewählt werden. Der gewählte Ausgangswert in der Tabelle für das Signal ist gleichzusetzen mit dem Initialwert des Signals. Dieser soll in diesem Fall "**false**" sein (siehe [Abbildung 7](#), Schritt 3).

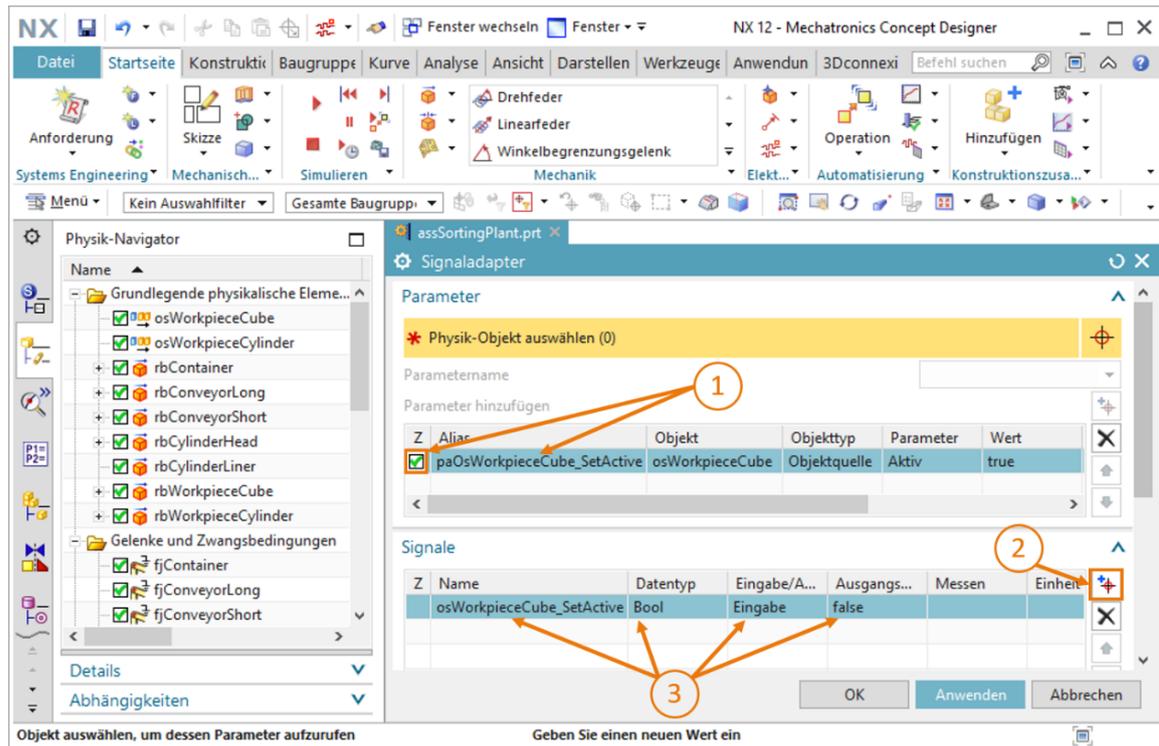


Abbildung 7: Erstellen eines passenden Signals für einen Parameter

→ Der Parameter und das zugehörige Signal müssen nun logisch miteinander verknüpft werden. Scrollen Sie dazu in dem Befehlsfenster nach unten zu dem Befehlsreiter **"Formeln"**. Dort sehen Sie, dass Sie dem Parameter **"paOsWorkpieceCube\_SetActive"** eine Formel zuweisen können. Klicken Sie hierfür auf die entsprechende Zeile in der Tabelle (siehe [Abbildung 8](#), Schritt 1). In dem Eingabefeld **"Formel"** können Sie nun eine geeignete Zuordnung treffen. In diesem Fall genügt eine einfache Zuordnung des Signals **"osWorkpieceCube\_SetActive"** zu dem Parameter, wie in [Abbildung 8](#), Schritt 2 dargestellt. Nach einem Klick auf die Eingabe-Taste Ihrer Tastatur sehen Sie in der Tabelle unter der Spalte **"Formel"** die soeben getätigte Zuweisung.

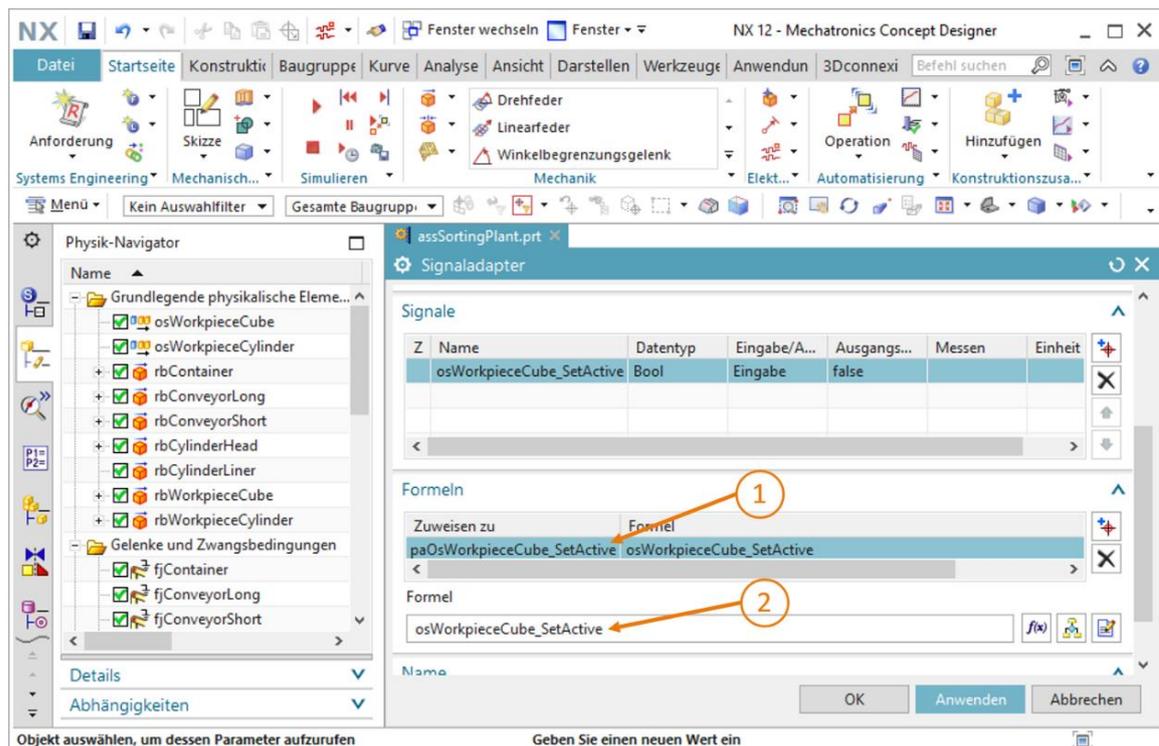


Abbildung 8: Definition einer Formel zwischen Signal und Parameter

## HINWEIS

Es können auch komplizierte Formeln gewählt werden, welche von mehreren Parametern und/oder Signalen abhängig sind. Hier sollte man jedoch zugunsten der Nachvollziehbarkeit des digitalen Zwillings darauf achten, so einfache Formeln wie möglich im Signaladapter zu definieren. Die Logik sollte vorrangig immer Teil des Automatisierungsprogramms und nicht des digitalen Zwillings sein.

Damit haben Sie das erste Signal Ihres digitalen Zwillings eigenständig erstellt. Erstellen Sie nun die verbleibenden Signale nach dem Vorgehen aus [Kapitel 7.1](#), "**Abschnitt: Signalerstellung und -verknüpfung mit Signaladapter**". Nutzen Sie dafür folgende Kenndaten:

- Aus der Objektquelle "**osWorkpieceCylinder**" soll der Parameter "**Aktiv**" mit dem Alias-Namen "**paOsWorkpieceCylinder\_SetActive**" im Signaladapter erstellt werden. Setzen Sie wiederum das Häkchen  vor dem Parameter, um ein Signal zuweisen zu können. Das zugehörige Signal soll den Namen "**osWorkpieceCylinder\_SetActive**" tragen und vom Datentyp "**Bool**" sein. Dieses Signal ist als "**Eingabe**" zu definieren und besitzt einen Ausgangswert von "**false**". Weisen Sie als Formel für den Parameter "**paOsWorkpieceCylinder\_SetActive**" die direkte Zuordnung des Signals "**osWorkpieceCylinder\_SetActive**" zu.
- Erzeugen Sie für den Parameter "**getriggert**" des Kollisionssensors "**csLightSensorCube**" einen neuen Parameter mit dem Alias-Namen "**paCsLightSensorCube\_Detected**" im Signaladapter. Für das zugehörige Signal geben Sie ein **boolesches** Signal mit dem Namen "**csLightSensorCube\_Detected**" an. Dieses Signal muss jedoch als "**Ausgabe**" definiert werden, da der Parameter "**paCsLightSensorCube\_Detected**" in der Spalte "**Lesen/Schreiben**" den Wert "**R**" aufzeigt, was bedeutet, dass er gelesen werden muss. Weisen Sie dem Signal einen Ausgangswert von "**false**" zu. Setzen Sie hier das Häkchen  vor dem Signal, wie in [Abbildung 9](#), Schritt 1 dargestellt, um dem Ausgabesignal eine Formel zuweisen zu können. Nutzen Sie für das Signal "**csLightSensorCube\_Detected**" als Formel die direkte Zuweisung des Parameters "**paCsLightSensorCube\_Detected**".

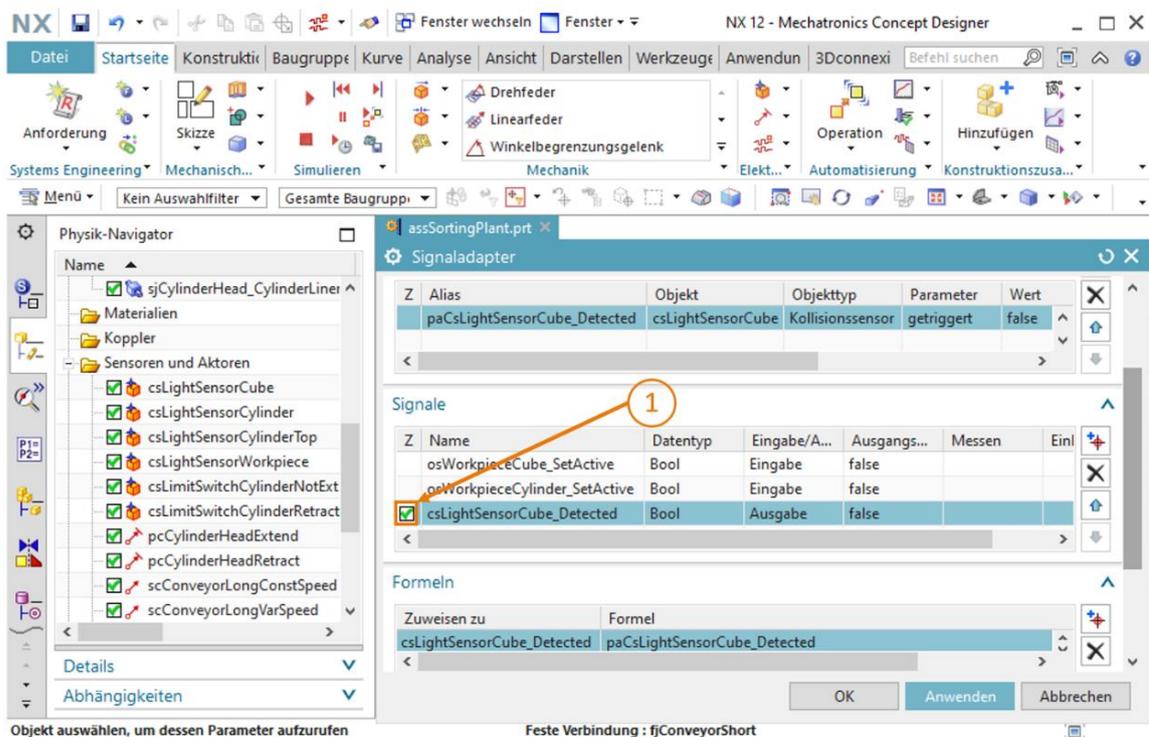


Abbildung 9: Erstellen eines Ausgabe-Signals für einen Lichttaster

→ Das nächste Lichttaster-System in der Mitte des Förderprozesses besteht aus zwei Kollisionssensoren. Deshalb müssen dafür zwei Parameter erstellt werden. Definieren Sie zuerst für den Parameter "**getriggert**" des Kollisionssensors "**csLightSensorCylinder**" einen Parameter mit dem Alias-Namen "**paCsLightSensorCylinder\_Detected**" im Signaladapter. Erstellen Sie für den Parameter "**getriggert**" des zweiten Kollisionssensors "**csLightSensorCylinderTop**" einen Parameter mit dem Alias-Namen "**paCsLightSensorCylinderTop\_Detected**" im Signaladapter. Es soll nun ein kombiniertes Signal erzeugt werden, welches auf beide Parameter reagiert. Dieses soll den Namen "**csLightSensorCylinder\_Detected**" erhalten und vom Datentyp "**Bool**" sein. Da auch in diesem Fall beide oben genannte Parameter gelesen werden müssen, soll das kombinierte Signal als "**Ausgabe**" konfiguriert werden und den Ausgangswert "**false**" besitzen. Geben Sie für "**csLightSensorCylinder\_Detected**", wie in [Abbildung 10](#), Schritt 1 gezeigt, folgende Formel vor:

**"((paCsLightSensorCylinderDetected) & (!paCsLightSensorCylinderTop\_Detected))".**

Diese Formel stellt eine UND-Verknüpfung beider Parameter dar, wobei der zweite Parameter negiert wird, d. h. das Ausgangssignal "**csLightSensorCylinder\_Detected**" wird "**true**", wenn "**paCsLightSensorCylinderDetected**" den Wert "**true**" annimmt und gleichzeitig "**paCsLightSensorCylinderTop\_Detected**" "**false**" ist. Den Erklärungen in Modul 1 dieser Workshop-Reihe bzgl. der Funktionsweise der Sortieranlage zufolge, wird aufgrund der Höhe der Werkstücke ein zylinderförmiges Werkstück nur dann erkannt, wenn der untere Lichttaster dieses Lichttaster-Systems auslöst, aber zum selben Zeitpunkt der obere Lichttaster keine Kollision erkennt. Diese Logik wird mit dieser Formel dargestellt.

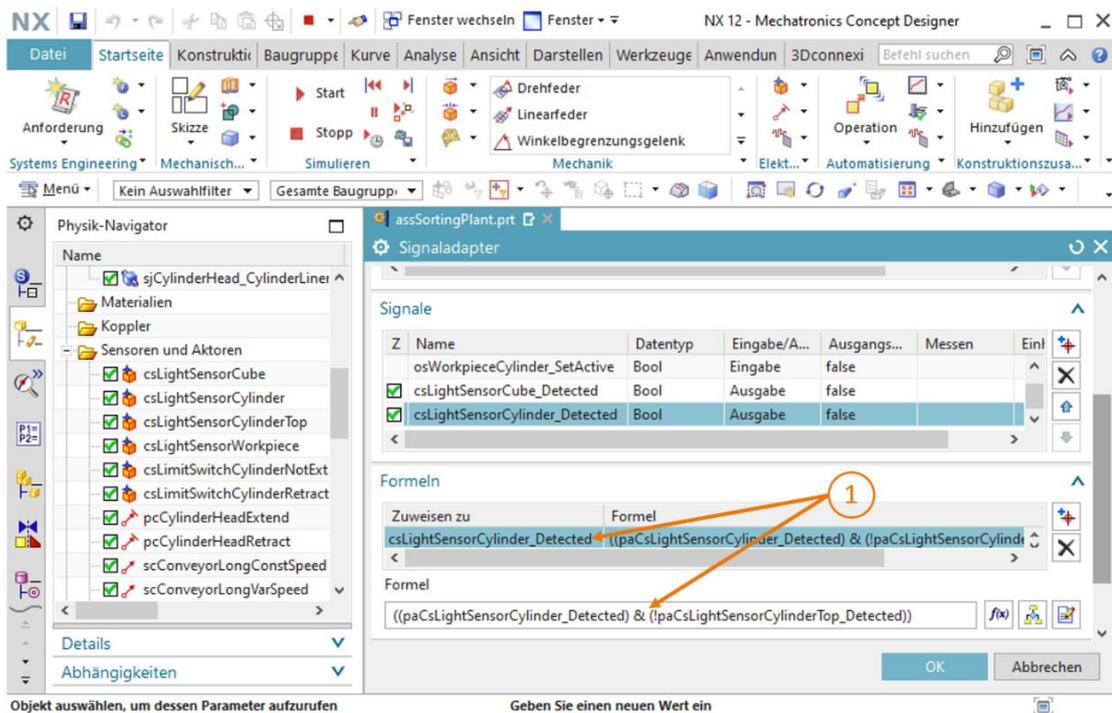


Abbildung 10: Formel für das Signal des Lichttaster Systems "csLightSensorCylinder"



## HINWEIS

Immer wenn einem schreibbaren Parameter ein Eingangssignal zugewiesen werden soll, muss das Häkchen  vor dem Parameter gesetzt werden. Ebenso muss sobald einem Ausgangssignal ein lesbarer Parameter zugewiesen werden soll, das Häkchen  vor dem Signal gesetzt werden.

Erst wenn ein Häkchen  zu sehen ist, können Sie die entsprechende Formel definieren.

- Für den Parameter **"getriggert"** des Kollisionssensors **"csLightSensorWorkpiece"** benötigen Sie im Signaladapter einen Parameter, dem Sie den Alias-Namen **"paCsLightSensorWorkpiece\_Detected"** geben sollen. Erstellen Sie dazu ein **boolesches** Signal mit dem Namen **"csLightSensorWorkpiece\_Detected"**. Auch dieses wird als **Ausgabe**-Signal mit einem Anfangswert von **"false"** definiert. Die Formel für das Signal **"csLightSensorWorkpiece\_Detected"** lautet **"paCsLightSensorWorkpiece\_Detected"**.
- Erstellen Sie für den Parameter **"getriggert"** des Kollisionssensors **"csLimitSwitchCylinderNotExtended"** im Signaladapter einen Parameter mit dem Alias-Namen **"paCsLimitSwitchCylinderNotExtended\_Activated"**. Fügen Sie ein **boolesches** Signal namens **"csLimitSwitchCylinderNotExtended\_Activated"** hinzu. Dieses Signal soll als **Ausgabe**-Typ definiert sein und einen Ausgangswert von **"false"** betragen. Geben Sie als Formel für das Signal **"csLimitSwitchCylinderNotExtended\_Activated"** die Zuordnung **"paCsLimitSwitchCylinderNotExtended\_Activated"** an.
- Der Parameter **"getriggert"** des letzten verbleibenden Kollisionssensors **"csLimitSwitchCylinderRetracted"** soll ebenfalls als Parameter im Signaladapter mit dem Alias-Namen **"paCsLimitSwitchCylinderRetracted\_Activated"** angelegt werden. Das zugehörige Signal **"csLimitSwitchCylinderRetracted\_Activated"** soll den Datentyp **"Bool"** erhalten. Außerdem soll dieses als **Ausgabe**-Signal mit einem Ausgangswert von **"false"** definiert werden. Die Formel für **"csLimitSwitchCylinderRetracted\_Activated"** ist als einfache Zuordnung des Parameters **"paCsLimitSwitchCylinderRetracted\_Activated"** anzugeben.
- Für den Parameter **"Aktiv"** des Positionsreglers **"pcCylinderHeadExtend"** soll im Signaladapter ein Parameter mit dem Alias-Namen **"paPcCylinderHeadExtend\_SetActive"** erstellt werden. Erzeugen Sie weiterhin ein neues Signal namens **"pcCylinderHeadExtend\_SetActive"** vom Datentyp **"Bool"**. Definieren Sie das Signal als **"Eingabe"** mit einem Ausgangswert von **"false"**. Als Formel für den Parameter **"paPcCylinderHeadExtend\_SetActive"** geben Sie die Zuweisung des Signals **"pcCylinderHeadExtend\_SetActive"** vor.

- Im Folgenden sollen Sie für den Parameter "**Aktiv**" des Positionsreglers "**pcCylinderHeadRetract**" einen Parameter im Signaladapter mit dem Alias-Namen "**paPcCylinderHeadRetract\_SetActive**" erzeugen. Das dazu korrespondierende Signal, soll den Namen "**pcCylinderHeadRetract\_SetActive**" tragen. Definieren Sie dieses Signal als **Boolean** und als **Eingabe**-Signal mit einem Ausgangswert von "**false**". Die Formel von "**paPcCylinderHeadRetract\_SetActive**" lautet "**pcCylinderHeadRetract\_SetActive**".
- Der Parameter "**Aktiv**" des Geschwindigkeitsreglers "**scConveyorLongConstSpeed**" bedarf eines Parameters im Signaladapter mit dem Alias-Namen "**paScConveyorLongConstSpeed\_SetActive**". Das dazugehörige Signal soll den Namen "**scConveyorLongConstSpeed\_SetActive**" erhalten. Dieses Signal soll vom Datentyp "**Bool**" sein, als **Eingabe**-Signal definiert werden und einen Ausgangswert von "**false**" besitzen. Als Formel für "**paScConveyorLongConstSpeed\_SetActive**" soll eine einfache Zuweisung von "**scConveyorLongConstSpeed\_SetActive**" verwendet werden.
- Für den Geschwindigkeitsregler "**scConveyorLongVarSpeed**" müssen zwei Parameter und zwei Signale im Signaladapter definiert werden.

Davon sollen der erste Parameter und das zugehörige Signal im Signaladapter zur Aktivierung des Geschwindigkeitsreglers dienen. Erstellen Sie dazu für den Parameter "**Aktiv**" des Geschwindigkeitsreglers "**scConveyorLongVarSpeed**" im Signaladapter einen neuen Parameter mit dem Alias-Namen "**paScConveyorLongVarSpeed\_SetActive**". Erzeugen Sie zusätzlich das Signal "**scConveyorLongVarSpeed\_SetActive**" und deklarieren Sie den Datentypen als "**Bool**". Das Signal soll zur **Eingabe** dienen. Der Ausgangswert soll "**false**" sein. Geben Sie schließlich als Formel für "**paScConveyorLongVarSpeed\_SetActive**" das Signal "**scConveyorLongVarSpeed\_SetActive**" vor.

Mit dem zweiten Signal soll es möglich sein, die Sollgeschwindigkeit des Geschwindigkeitsreglers variabel vorzugeben. Erzeugen Sie hierfür im Signaladapter einen neuen Parameter, welcher mit dem Parameter **"Geschwindigkeit"** des Geschwindigkeitsreglers **"scConveyorLongVarSpeed"** verknüpft sein soll und verwenden Sie den Alias-Name **"paScConveyorLongVarSpeed\_SetSpeed"**. Das zugehörige Signal soll den Namen **"scConveyorLongVarSpeed\_SetSpeed"** erhalten. Da mit diesem Signal eine Geschwindigkeit vorgegeben werden soll, ist der Datentyp als **"double"** zu deklarieren. Bei einem Signal, welches nicht dem Datentyp **"Bool"** entspricht, muss der physikalische Typ unter der Rubrik **"Messen"** und die zugehörige physikalische Einheit des Werts unter der Rubrik **"Einheit"** eingegeben werden. Stellen Sie in diesem Fall sicher, dass Sie für das aktuelle Signal in der Spalte **"Messen"** den Wert **"Geschwindigkeit"** sowie in der Spalte **"Einheit"** den Ausdruck **"mm/s"** vorgeben (siehe [Abbildung 11](#), Schritt 1). Es handelt sich auch hierbei um ein **Eingabe**-Signal. Legen Sie überdies als Ausgangswert **"0.0"** noch fest. Als Formel für den Parameter **"paScConveyorLongVarSpeed\_SetSpeed"** genügt die Zuweisung des Signals **"scConveyorLongVarSpeed\_SetSpeed"**.

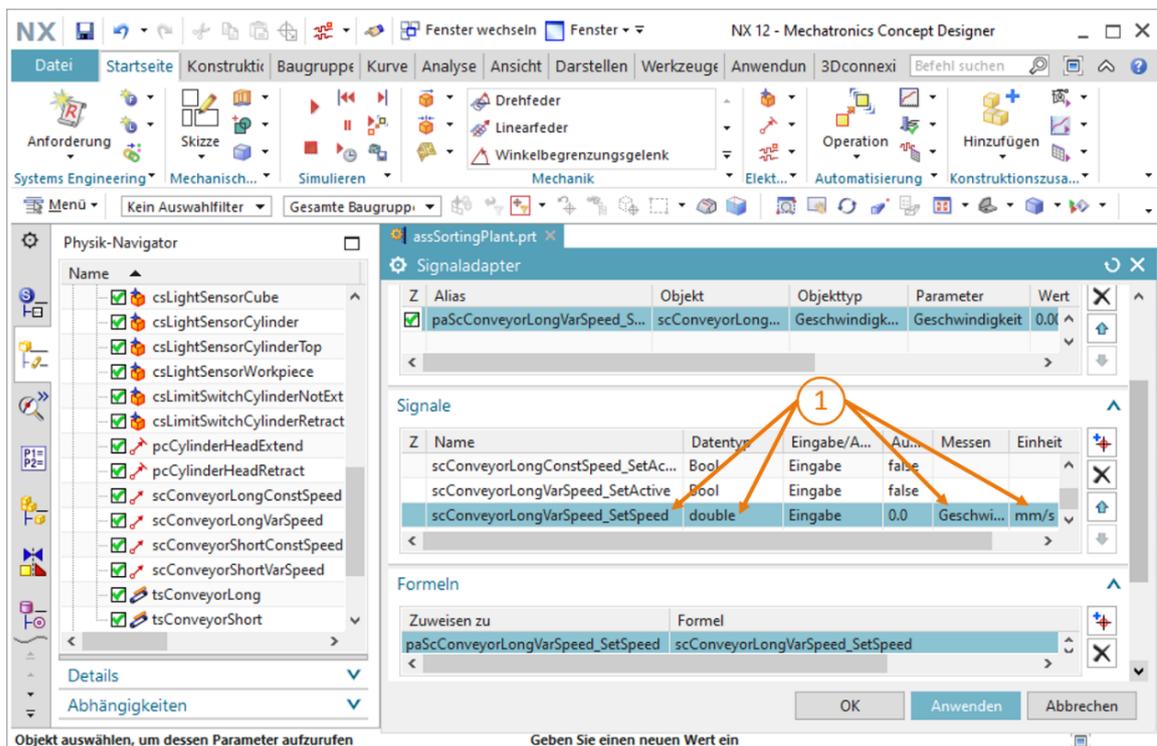


Abbildung 11: Erstellen eines Geschwindigkeitssignals vom Datentyp "double"

- Für den Geschwindigkeitsregler "**scConveyorShortConstSpeed**" soll der Parameter "**Aktiv**" als neuer Parameter im Signaladapter eingefügt werden. Geben Sie diesen Parameter den Alias-Namen "**paScConveyorShortConstSpeed\_SetActive**". Erstellen Sie das korrespondierende Signal "**scConveyorShortConstSpeed\_SetActive**" vom Datentyp "**Bool**". Es ist als **Eingabe**-Signal zu kennzeichnen und soll einen Anfangswert von "**false**" besitzen. Vergeben Sie schließlich als Formel für "**paScConveyorShortConstSpeed\_SetActive**" eine direkte Zuweisung des Signals "**scConveyorShortConstSpeed\_SetActive**".
- Der Geschwindigkeitsregler "**scConveyorShortVarSpeed**" benötigt ebenfalls zwei Signale im Signaladapter.

Der erste Parameter im Signaladapter soll sich auf den Parameter "**Aktiv**" des Geschwindigkeitsreglers "**scConveyorShortVarSpeed**" beziehen und den Alias-Namen "**paScConveyorShortVarSpeed\_SetActive**" tragen. Als zugehöriges Signal soll ein **boolesches** Signal namens "**scConveyorShortVarSpeed\_SetActive**" erzeugt werden. Dieses soll ein **Eingabe**-Signal sein und mit einem Ausgangswert von "**false**" beginnen. Dem Parameter "**paScConveyorShortVarSpeed\_SetActive**" soll das Signal "**scConveyorShortVarSpeed\_SetActive**" zugewiesen werden.

Mit dem zweiten Signal soll die Sollgeschwindigkeit für den Geschwindigkeitsregler vorgeben werden. Erzeugen Sie dazu einen neuen Parameter im Signaladapter, welcher auf den Parameter "**Geschwindigkeit**" des Geschwindigkeitsreglers "**scConveyorShortVarSpeed**" basiert. Geben Sie diesem Parameter den Alias-Namen "**paScConveyorShortVarSpeed\_SetSpeed**". Beim Definieren des neuen Signals "**scConveyorShortVarSpeed\_SetSpeed**" ist erneut darauf zu achten, dass als Datentyp "**double**", in der Rubrik "**Messen**" "**Geschwindigkeit**" und als Einheit "**mm/s**" angegeben wird. Es handelt sich um ein **Eingabe**-Signal mit einem Ausgangswert von "**0.0**". Die Formel für den Parameter "**paScConveyorShortVarSpeed\_SetSpeed**" lautet "**scConveyorShortVarSpeed\_SetSpeed**".

→ Damit für die Transportfläche "**tsConveyorLong**" gewährleistet ist, dass Sie nur bei einem aktiven Signal eines der beiden zugehörigen Geschwindigkeitsreglers verfährt, muss ein weiterer Parameter dem Signaladapter hinzugefügt werden. Selektieren Sie für diese Transportfläche den Parameternamen "**Aktiv**" und geben Sie dem neuen Parameter den Alias-Namen "**paTsConveyorLong\_SetActive**". Weisen Sie schließlich, wie in [Abbildung 12](#), Schritt 1 angegeben, für diesen Parameter folgende Formel zu:

**"((scConveyorLongConstSpeed\_SetActive) | (scConveyorLongVarSpeed\_SetActive))"**

Hierbei steht das Zeichen "|" für die **Oder**-Verknüpfung. Somit ist gewährleistet, dass das Transportband nur verfährt, wenn mindestens ein Geschwindigkeitsregler für dieses Transportband aktiviert wurde. Aufgrund der Logik im entwickelten Automatisierungsprogramm kann im regulären Betriebsablauf die Transportfläche jedoch nur durch einen Geschwindigkeitsregler gleichzeitig betrieben werden.

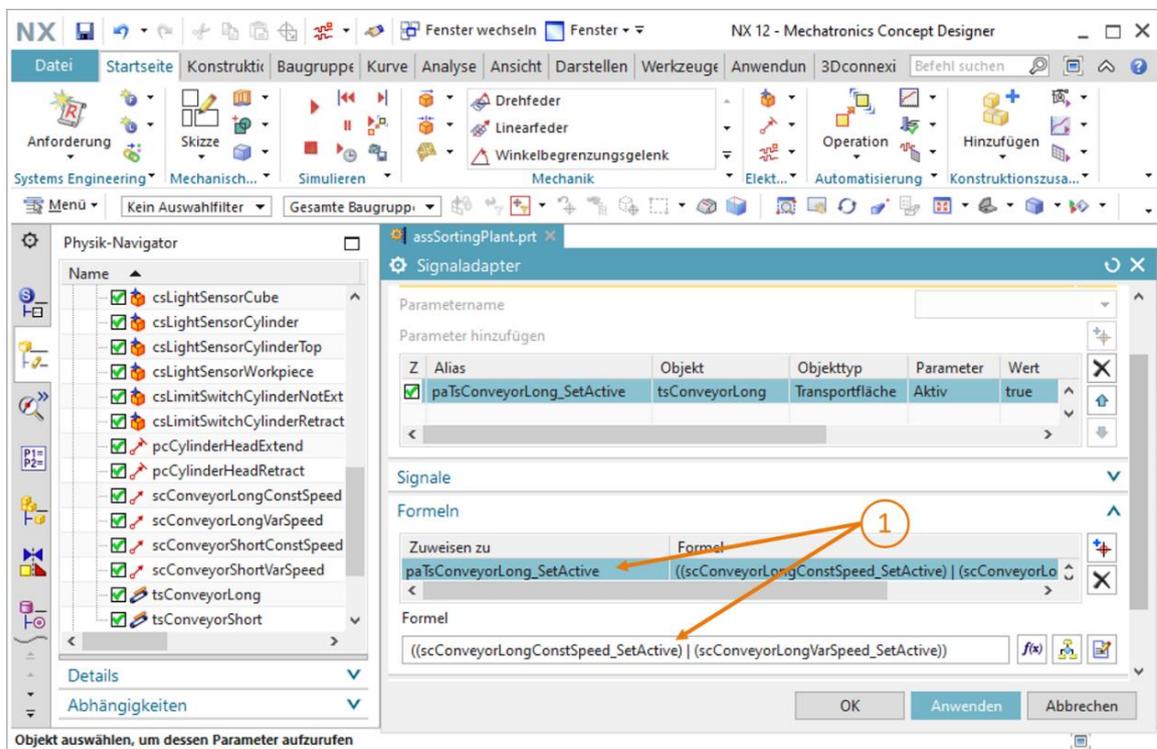


Abbildung 12: Erstellen eines Parameters für eine Transportfläche

→ Für die Transportfläche "**tsConveyorShort**" gilt selbiges Verhalten, wie zuvor für "**tsConveyorLong**" beschrieben. Wählen Sie den Status "**Aktiv**" für den neuen Parameter namens "**paTsConveyorShort\_SetActive**" aus.

Nutzen Sie hierzu die nachfolgende Formel:

**„((scConveyorShortConstSpeed\_SetActive))(scConveyorShortVarSpeed\_SetActive))"**,  
um den ordnungsgemäßen Betrieb der Transportfläche zu gewährleisten.

Damit sind alle notwendigen Parameter und Signale im Signaladapter definiert. Vergeben Sie abschließend als Namen für den Signaladapter "saSortingPlant" (siehe [Abbildung 13](#), Schritt 1). Das Präfix "sa" steht hierbei für die englische Bezeichnung "signal adapter". Bestätigen Sie die Konfiguration Ihres neuen Signaladapters mit einem Klick auf die Schaltfläche "OK" (siehe [Abbildung 13](#), Schritt 2).

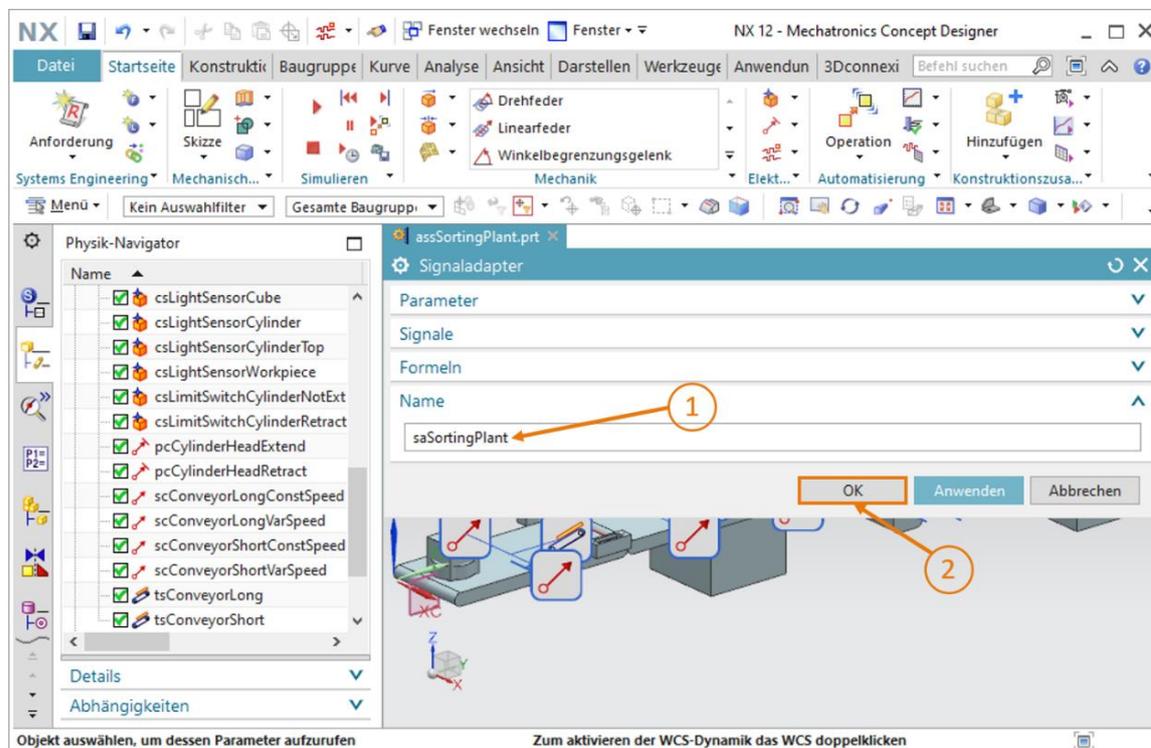


Abbildung 13: Erstellen des Signaladapters "saSortingPlant"

Im Nachfolgenden erscheint ein neues Fenster "**Symbole zu Symboltabelle hinzufügen**", in dem Sie dazu aufgefordert werden, anzugeben, in welche Symboltabelle die Signale Ihres Signaladapters als Symbole hinzugefügt werden sollen. Hier besteht die Möglichkeit eine bereits bestehende Symboltabelle zu erweitern oder eine neue Symboltabelle anzulegen. Da Sie in Ihrem derzeitigen Projekt noch keine Symboltabelle erzeugt haben, betätigen Sie die Schaltfläche "**Neue Symboltabelle erstellen**" (siehe [Abbildung 14](#), Schritt 1).

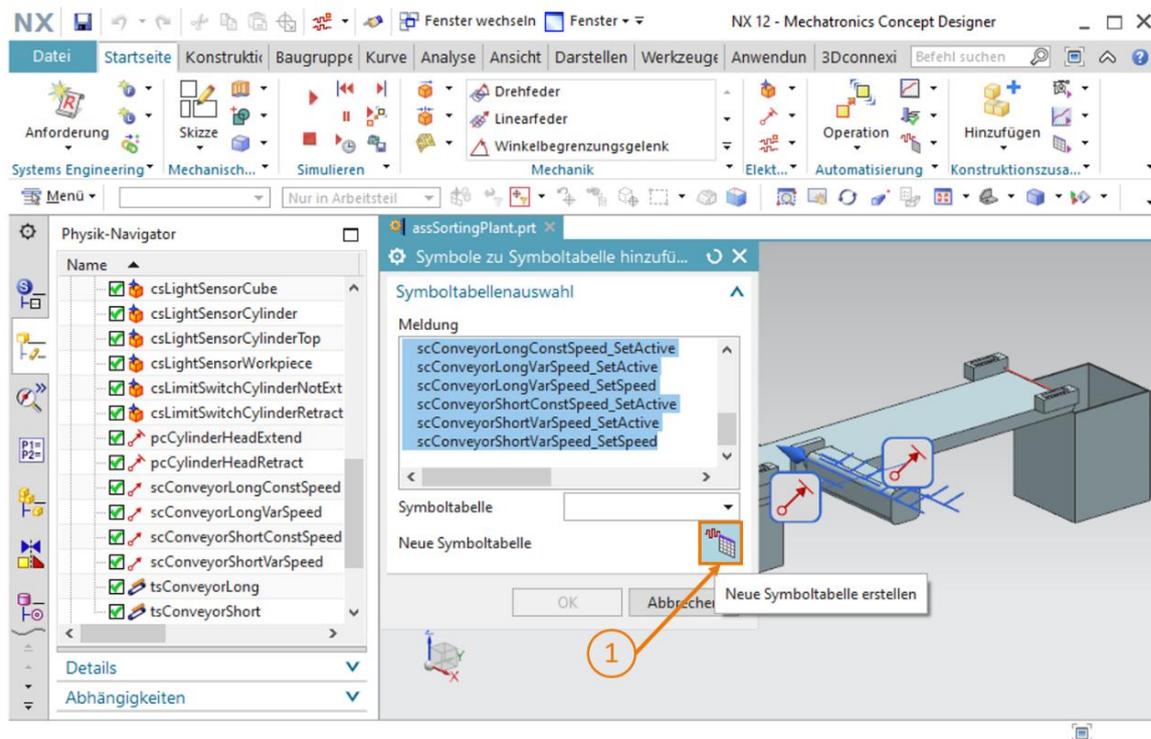


Abbildung 14: Erstellung einer neuen Symboltabelle für den Signaladapter initiieren

Es erscheint nun das Befehlsfenster "Symboltabelle". Hier können Sie zum einen neue Symbole definieren, zum anderen der Symboltabelle einen Namen zuweisen. Da Sie die Signale aus Ihrem Signaladapter vollständig übernehmen können, brauchen Sie hier keine neuen Signale zu definieren. Geben Sie der Symboltabelle den Namen "stSortingPlant" (siehe [Abbildung 15](#), Schritt 1) und klicken Sie auf die Schaltfläche "OK" (siehe [Abbildung 15](#), Schritt 2). Das Präfix "st" steht hierbei für die englische Bezeichnung "signal table".

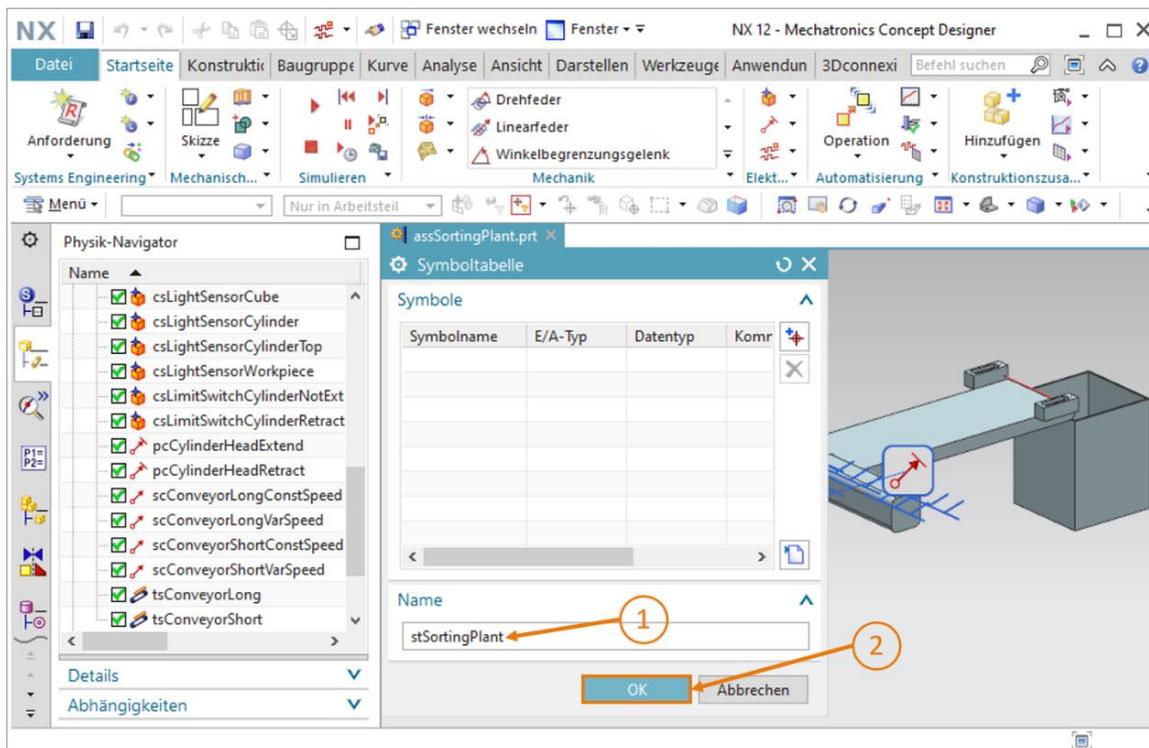


Abbildung 15: Erstellung einer neuen Symboltabelle für Signaladapter abschließen

Nun gelangen Sie wieder zurück in das Fenster **"Symbole zu Symboltabelle hinzufügen"**. Hier sollten Sie, sofern noch nicht geschehen, Ihre soeben erstellte Symboltabelle **"stSortingPlant"** auswählen, wie in [Abbildung 16](#), Schritt 1 dargestellt. Schließen Sie den Erstellprozess durch Betätigen der Schaltfläche **"OK"** ab (siehe [Abbildung 16](#), Schritt 2).

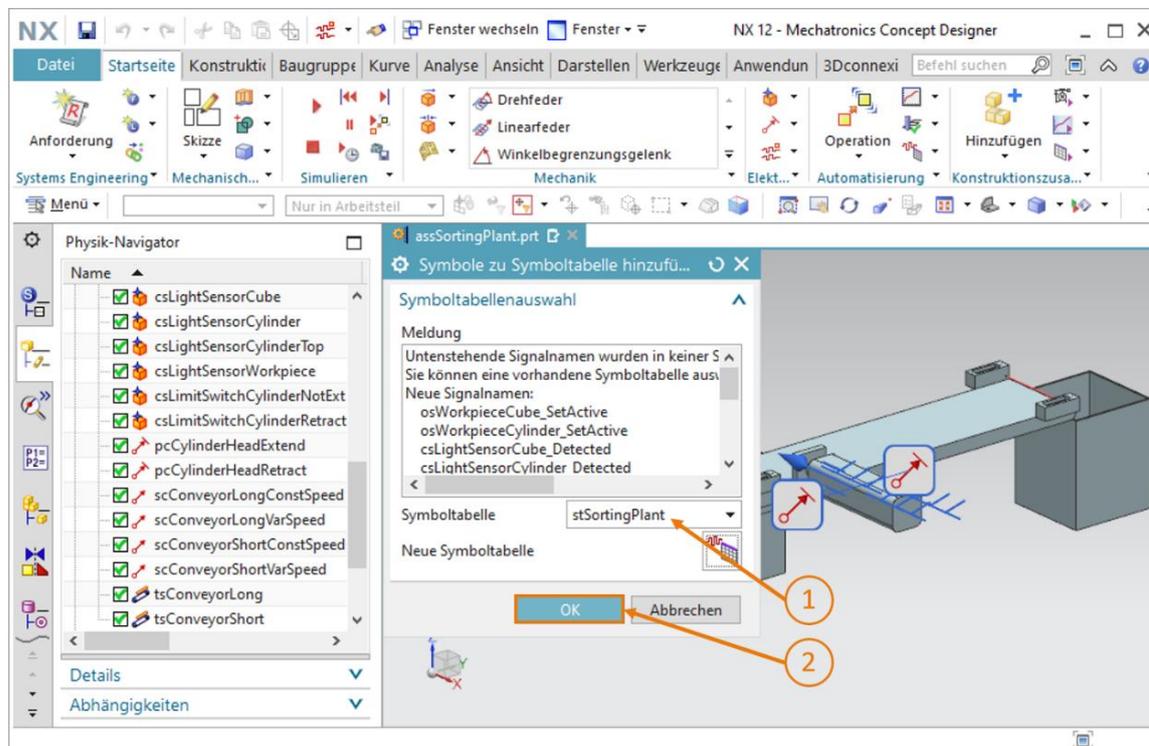


Abbildung 16: Abschließen der Symbolzuordnung für Signaladapter

Damit sind alle nötigen Signale in Ihrem dynamischen 3D-Modell eingefügt, sodass Sie im Nachfolgenden eine Signalverbindung zu einer virtuellen SPS aufbauen können. Speichern Sie aber zunächst die Änderungen an dem Modell durch einen Klick auf die Schaltfläche

**"Speichern"** .

## 7.2 Erzeugen einer Signal-Verbindung zwischen virtueller SPS und digitalem Zwilling

Zum Erzeugen einer Signal-Verbindung muss eine virtuelle SPS bereits in Betrieb sein. In diesem Abschnitt müssen Sie deshalb wieder auf das TIA Portal und PLCSIM Advanced zurückgreifen. Gehen Sie zur Anbindung wie folgt vor:

- Entpacken Sie in Ihrem Betriebssystem das mit diesem Modul bereitgestellte Archiv (siehe [Kapitel 7](#)) und speichern Sie den Inhalt des Ordners "**fullPlcBasic**" in einen Ordner Ihrer Wahl. Der Ordner enthält das in Modul 1 bereits verwendete und in Modul 2 beschriebene Automatisierungsprogramm.
- Öffnen Sie nun das **TIA Portal** und dearchivieren Sie das Projekt "**150-006\_DigitalTwinAtEducation\_TIAP\_Basic.zap15**" aus Ihrem soeben erstellten Ordner. Gehen Sie dafür, wie in **Kapitel 7.1 des Moduls 1 der Workshop-Reihe zu DigitalTwin@Education** beschrieben, vor.
- Übersetzen Sie sowohl die Hardwarekonfiguration, als auch die Software des Automatisierungsprogramms. Befolgen Sie dazu die Erläuterungen aus **Kapitel 7.2 in Modul 1 dieser Workshop-Reihe**.
- Öffnen Sie das Programm "**S7-PLCSIM Advanced**" und starten Sie eine neue Instanz einer virtuellen SPS. Nennen Sie diese Instanz "**DigTwinAtEdu\_PLCSIM**". Laden Sie anschließend Ihr Automatisierungsprogramm in die virtuelle SPS und warten Sie bis der CPU Status in "**Start**" wechselt, d. h. ein grünes Kästchen vor dem Instanznamen erscheint. Gehen Sie hier nach den Beschreibungen aus **Kapitel 7.3 des Moduls 1 dieser Workshop-Reihe** vor.

Damit ist die virtuelle SPS betriebsbereit, sodass Sie nun die Signal-Verbindung zum dynamischen 3D-Modell konfigurieren können. Wechseln Sie zurück in den Mechatronics Concept Designer zu Ihrem dynamischen 3D-Modell mit Signalen und gehen Sie nach folgenden Schritten vor:

- Führen Sie in der Menüleiste "**Automatisierung**" den Befehl "**Signalzuordnung**" aus (siehe [Abbildung 17](#), Schritt 1). Es öffnet sich das Befehlsfenster "**Signalzuordnung**". Dort müssen Sie zunächst die externe Signalquelle auswählen. Gehen Sie dazu in den Reiter "**Externer Signaltyp**" und selektieren Sie als Typ "**PLCSIM-Assistent**", da Sie eine Verbindung mit PLCSIM Advanced anstreben (siehe [Abbildung 17](#), Schritt 2). Zu diesem Zeitpunkt ist Ihrem dynamischen Modell noch nicht bekannt, zu welcher Instanz in PLCSIM Advanced eine Verbindung aufgebaut werden soll. Daher klicken Sie unter dem Punkt "**PLCSIMAdv-Instanzen**" auf die Schaltfläche "**Einstellungen**" (siehe [Abbildung 17](#), Schritt 3).

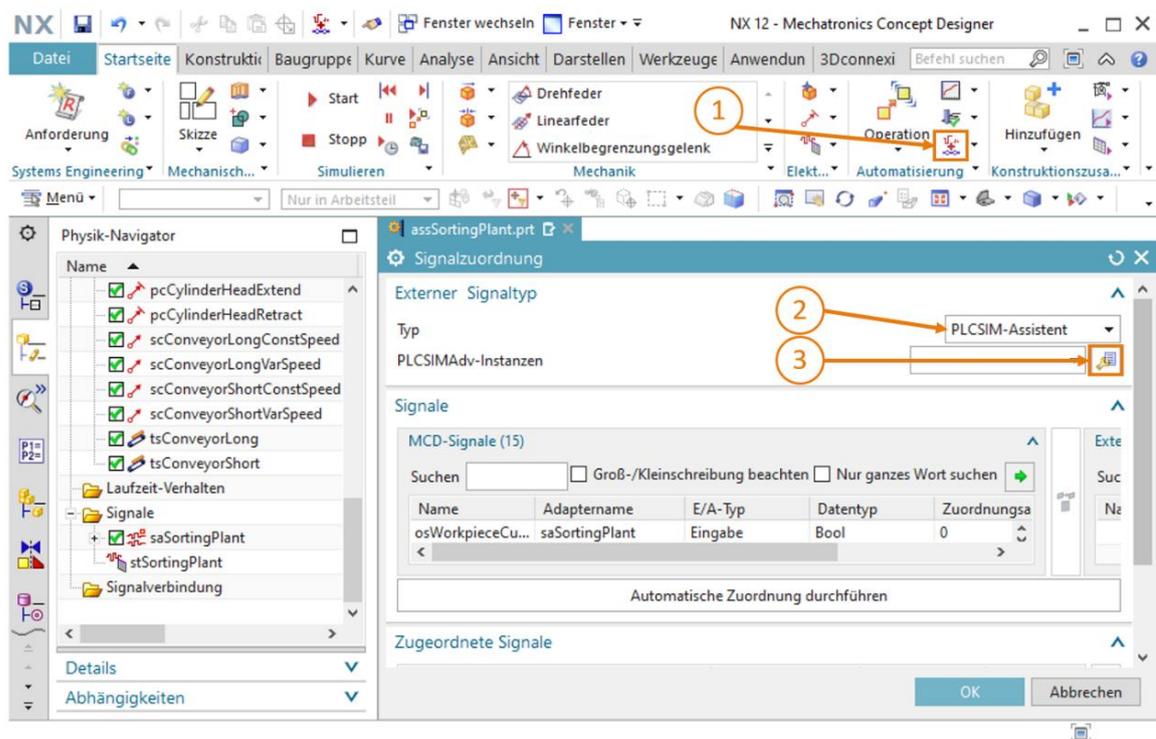


Abbildung 17: Auswahl einer Signalzuordnung über PLCSIM Advanced

## HINWEIS

Sollten Sie bereits vorher Signal-Verbindungen für das dynamische Modell angelegt haben, erscheinen in der Auswahlleiste unter dem Punkt "**PLCSIMAdv-Instanzen**" **alle jemals zuvor für dieses Modell verwendeten PLCSIM Advanced-Instanzen**. Beachten Sie jedoch, dass diese Instanzen nicht zwangsläufig noch existieren bzw. gültig sind. Zur Überprüfung klicken Sie auf die Schaltfläche "**Einstellungen**" und sehen Sie sich den derzeitigen Status der jeweiligen Instanz an.

→ Nun öffnet sich das Fenster **"Externe Signalkonfiguration"**. Hier können Sie die gewünschte Instanz auswählen und die dazugehörigen Variablen für die Signalzuordnung freigeben. Klicken Sie zunächst auf die Schaltfläche **"Registrierte Instanzen aktualisieren"** (siehe [Abbildung 18](#), Schritt 1). Es erscheint Ihre zuvor gestartete und geladene virtuelle SPS-Instanz. Der Status **"Run"** gibt an, dass diese virtuelle SPS erreichbar ist. Nach Selektion dieser Instanz, wie in [Abbildung 18](#), Schritt 2 hervorgehoben, werden Ihnen die E/A-Signale des Automatisierungsprogramms angezeigt. Fügen Sie alle verfügbaren Variablen ein, indem Sie das Häkchen bei **"Alle auswählen"** durch Klicken auf das Kästchen setzen (siehe [Abbildung 18](#), Schritt 3). Bestätigen Sie die Auswahl mit einem Klick auf **"OK"** (siehe [Abbildung 18](#), Schritt 4).

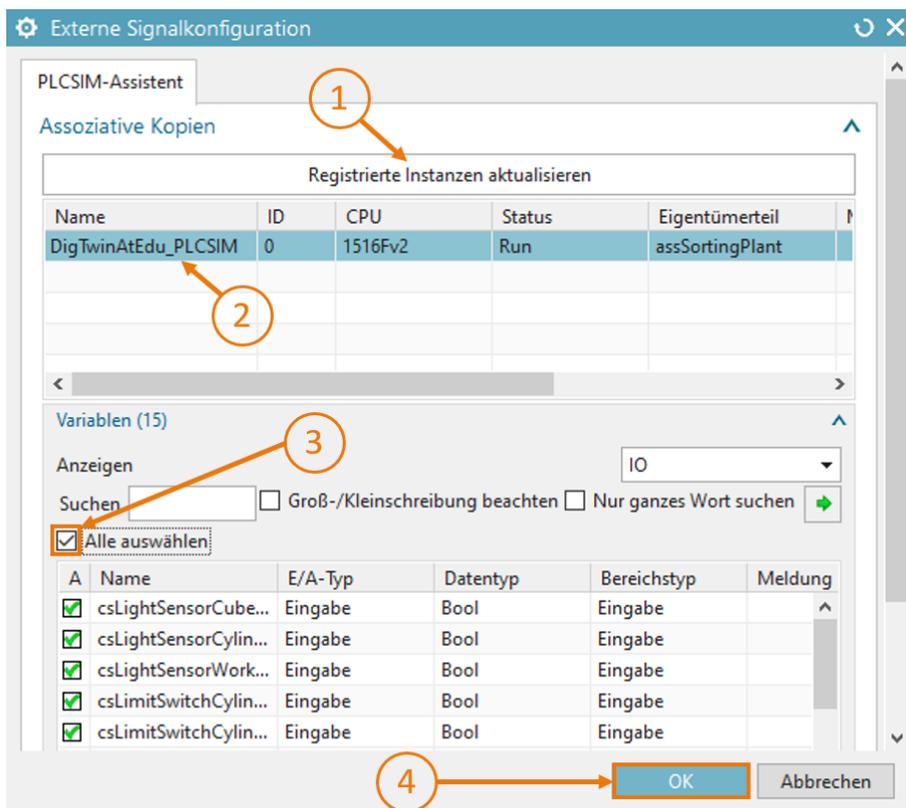


Abbildung 18: Variablen aus PLCSIM Advanced-Instanz für Signalzuordnung freigeben



**HINWEIS**

Wann immer eine Änderung oder Erweiterung des Automatisierungsprogramms ausgeführt wurde, müssen die registrierte Instanz für die Signalzuordnung aktualisiert und ggf. weitere Signal hinzugefügt werden.

→ Sie gelangen zurück zum Befehlsfenster **"Signalzuordnung"**. Dort finden Sie die soeben ausgewählte virtuelle SPS sowie die damit verfügbaren externen Signale im rechten Teil des Fensters. Sie können nun mit dem Zuordnen der Signale beginnen. Selektieren Sie zuerst in der Tabelle **"MCD-Signale"** im linken Teil des Fensters das Signal **"osWorkpieceCube\_SetActive"** (siehe [Abbildung 19](#), Schritt 1). Suchen Sie anschließend in der Tabelle **"Externe Signale"** nach dem korrespondierenden Signal aus dem Automatisierungsprogramm. Die Bezeichnungen aus beiden Programmen wurden hier sinnvollerweise identisch gewählt, wie [Abbildung 19](#), Schritt 2 verdeutlicht. Klicken Sie auf die Schaltfläche **"Signal zuordnen"**, um eine Verbindung zwischen den beiden Signalen herzustellen (siehe [Abbildung 19](#), Schritt 3). Es ist hierbei wichtig zu erwähnen, dass sich Eingabe-Signale in MCD nur mit Ausgabe-Signalen einer SPS verbinden lassen und umgekehrt..

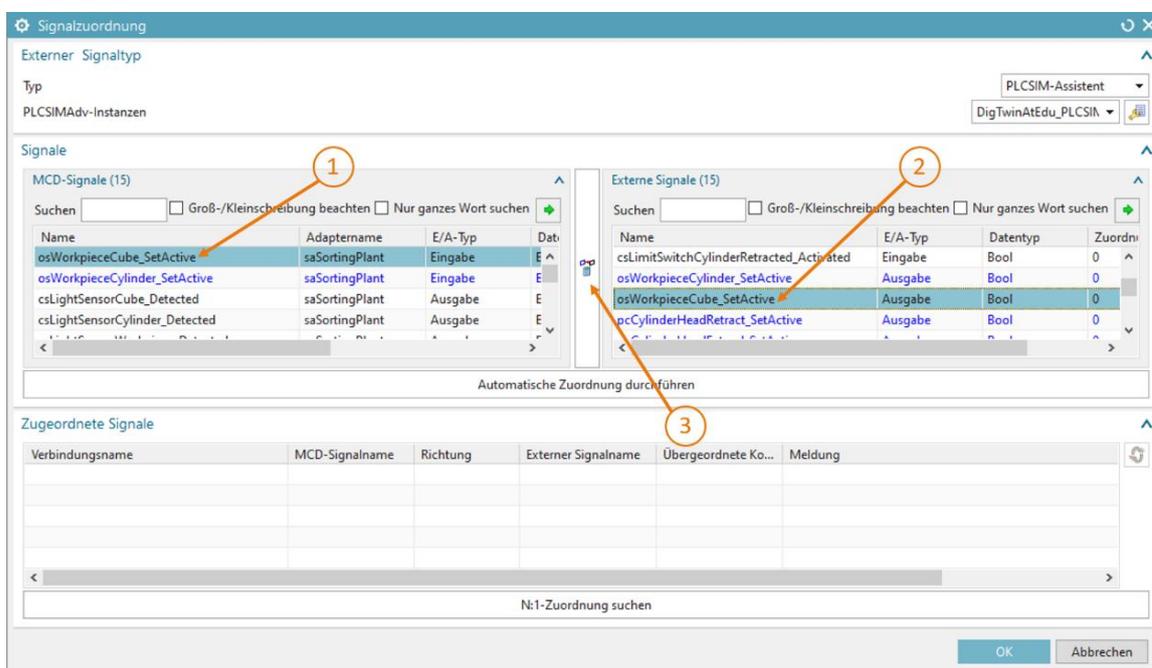


Abbildung 19: Zuordnen eines MCD-Signals zu einem externen Signal

→ In der Tabelle unter dem Befehlspunkt **"Zugeordnete Signale"** können Sie jetzt die soeben getätigte Signalzuordnung sehen. Fügen Sie nun die weiteren Zuordnungen ein. Da in diesem Modell die Namen der MCD-Signale mit den Variablennamen des Automatisierungsprogramms übereinstimmen, können Sie die Schaltfläche **"Automatische Zuordnung durchführen"** auswählen, um diesen Vorgang automatisch durch das Programm durchführen zu lassen (siehe [Abbildung 20](#), Schritt 1).

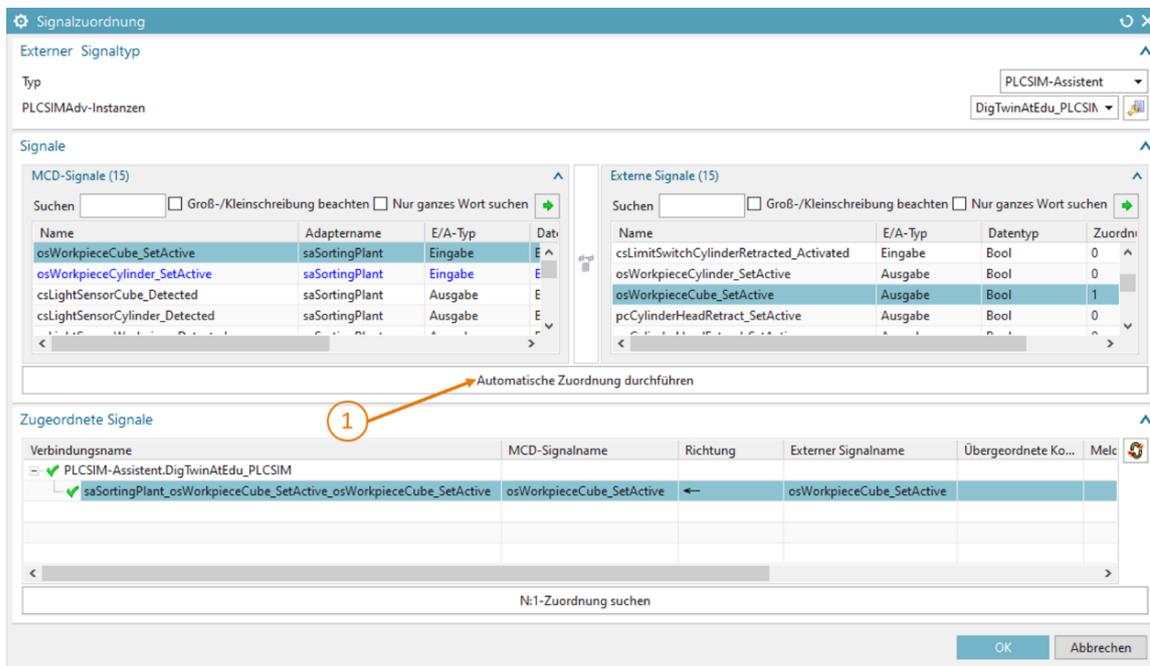


Abbildung 20: Alle Signale durch automatische Zuordnung verbinden



**HINWEIS**

Sollten Sie ein Signal falsch zugeordnet haben, können Sie die entsprechende Zuordnung in der Tabelle unter dem Befehlspunkt **"Zugeordnete Signale"** selektieren und durch Betätigen der Schaltfläche **"Fase erzeugen"**  die Verbindung wieder auftrennen (siehe [Abbildung 21](#), Schritt 1). Anschließend müssen Sie die korrekte Zuordnung neu anlegen.

Zugeordnete Signale

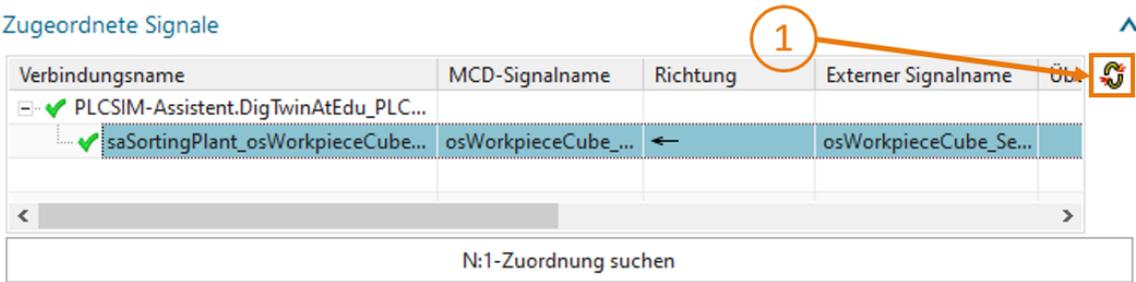


Abbildung 21: Signalzuordnung wieder auftrennen

→ Der Vorgang hat nun alle 15 Signale zwischen dem dynamischen 3D-Modell und der virtuellen SPS miteinander verbunden. Prüfen Sie die Zuordnungen auf Ihre Richtigkeit und schließen Sie die Signalzuordnung durch einen Klick auf die Schaltfläche "OK" ab (siehe [Abbildung 22](#), Schritt 1).

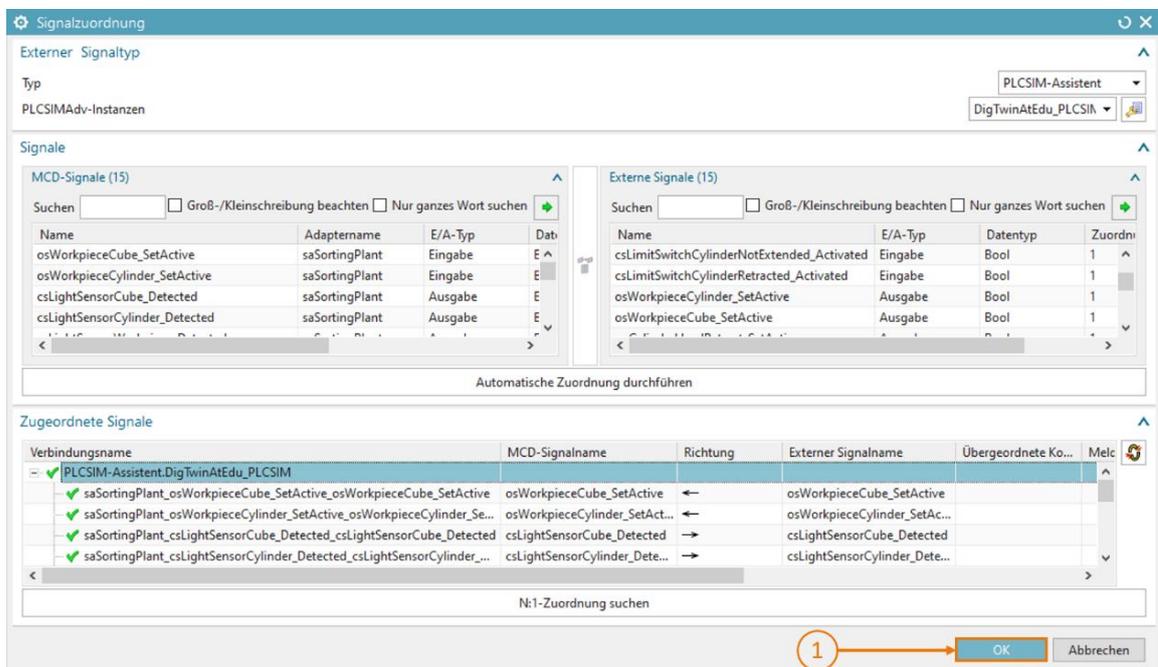


Abbildung 22: Signalzuordnung zwischen dynamischen Modell und virtueller SPS bestätigen

Damit ist die Verbindung zwischen dem dynamischen 3D-Modell in NX/MCD und dem Automatisierungsprogramm in Ihrer virtuellen SPS hergestellt. Speichern Sie Ihr Modell durch

Betätigen der "Speichern"-Schaltfläche .

## 7.3 Testen des digitalen Zwillings mit der virtuellen SPS

In diesem Kapitel sollen Sie Ihren digitalen Zwilling im Zusammenspiel mit einem Automatisierungsprogramm in einer virtuellen SPS betreiben und die Funktionalität validieren. Gehen Sie dazu nach folgendem Schema vor:

- Nachdem Sie das Automatisierungsprogramm in [Kapitel 7.2](#) bereits in eine Instanz einer virtuellen SPS hineingeladen haben, starten Sie nun das HMI über das Simulationstool "**WinCC Runtime Advanced**". Dies soll über das TIA Portal geschehen. Nutzen Sie hierfür das Vorgehen aus **Kapitel 7.4 von Modul 1 der Workshop-Reihe zu DigitalTwin@Education**.
- Wechseln Sie anschließend in das Programm "**Mechatronics Concept Designer**" und starten Sie für Ihren digitalen Zwilling eine Simulation. Betätigen Sie dafür in der Menüleiste "**Simulation**" den Befehl "**Start**" .
- Führen Sie für Ihren digitalen Zwilling die **beiden Testszenerien aus dem ersten Modul dieser Workshop-Reihe** durch und validieren Sie die Funktionsweise Ihres digitalen Zwillings. Befolgen Sie dabei die Beschreibungen aus **Kapitel 7.6 aus Modul 1 dieser Workshop-Reihe**. Sie können erkennen, dass sich Ihr selbst erstellter digitaler Zwilling aus den Modulen 4 – 6 dieser Workshop-Reihe gleich verhält, wie das vorgegebene Modell, welches Sie für die ersten drei Module verwendet haben. Stoppen Sie am Ende Ihrer Testreihe die Simulation in MCD, beenden Sie die simulierte HMI-Instanz und schließen Sie Ihre virtuelle SPS.

Ihnen steht selbstverständlich nun frei, Ihren digitalen Zwilling mit Ihrem optimierten Automatisierungsprogramm aus **Modul 3** zu überprüfen.

Damit ist dieses Schulungsmodul am Ende angelangt. Mit dem erlangten Wissen können Sie nun selbstständig eigene digitale Zwillinge erzeugen und für Ihr Automatisierungsprojekt eine virtuelle Inbetriebnahme durchführen.

## 8 Checkliste – Schritt-für-Schritt-Anleitung

Die nachfolgende Checkliste hilft den Auszubildenden/Studierenden selbstständig zu überprüfen, ob alle Arbeitsschritte der Schritt-für-Schritt-Anleitung sorgfältig abgearbeitet wurden und ermöglicht eigenständig das Modul erfolgreich abzuschließen.

Nr.	Beschreibung	Geprüft
1	Ihr dynamisches Modell aus Modul 5 wurde um die benötigten Signale erfolgreich erweitert.	
2	Es wurde eine gültige Signal-Verbindung zwischen Ihrem digitalen Zwilling und der virtuellen SPS angelegt.	
3	Durch Simulation der Testszenarien aus Modul 1 dieser Workshop-Reihe konnte der selbst erstellte digitale Zwilling vollständig und erfolgreich validiert werden.	

Tabelle 1: Checkliste der "Signalerstellung für ein dynamisches 3D-Modell im CAE-System Mechatronics Concept Designer"

## 9 Weiterführende Informationen

Zur Einarbeitung bzw. Vertiefung finden Sie als Orientierungshilfe weiterführende Informationen, wie z.B.: Getting Started, Videos, Tutorials, Apps, Handbücher, Programmierleitfaden und Trial Software/Firmware, unter nachfolgenden Links:

### **Vorsicht “Weiterführende Informationen“ – In Vorbereitung**

Hier vorab interessante Links:

- [1] [support.industry.siemens.com/cs/document/90885040/programmierleitfaden-f%C3%BCr-s7-1200-s7-1500?dti=0&lc=de-DE](https://support.industry.siemens.com/cs/document/90885040/programmierleitfaden-f%C3%BCr-s7-1200-s7-1500?dti=0&lc=de-DE)
- [2] [support.industry.siemens.com/cs/document/109756737/leitfaden-standardisierung?dti=0&lc=de-DE](https://support.industry.siemens.com/cs/document/109756737/leitfaden-standardisierung?dti=0&lc=de-DE)
- [3] [omg.org/spec/UML/2.5.1/PDF](https://omg.org/spec/UML/2.5.1/PDF)
- [4] [geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/](https://geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/)
- [5] [geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/](https://geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/)

## Weitere Informationen

Siemens Automation Cooperates with Education  
**[siemens.de/sce](https://www.siemens.de/sce)**

SCE Lern-/Lehrunterlagen  
**[siemens.de/sce/module](https://www.siemens.de/sce/module)**

SCE Trainer Pakete  
**[siemens.de/sce/tp](https://www.siemens.de/sce/tp)**

SCE Kontakt Partner  
**[siemens.de/sce/contact](https://www.siemens.de/sce/contact)**

Digital Enterprise  
**[siemens.de/digital-enterprise](https://www.siemens.de/digital-enterprise)**

Totally Integrated Automation (TIA)  
**[siemens.de/tia](https://www.siemens.de/tia)**

TIA Portal  
**[siemens.de/tia-portal](https://www.siemens.de/tia-portal)**

TIA Selection Tool  
**[siemens.de/tia/tia-selection-tool](https://www.siemens.de/tia/tia-selection-tool)**

SIMATIC Controller  
**[siemens.de/controller](https://www.siemens.de/controller)**

SIMATIC Technische Dokumentation  
**[siemens.de/simatic-doku](https://www.siemens.de/simatic-doku)**

Industry Online Support  
**[support.industry.siemens.com](https://support.industry.siemens.com)**

Katalog- und Bestellsystem Industry Mall  
**[mall.industry.siemens.com](https://mall.industry.siemens.com)**

Siemens  
Digital Industries, FA  
Postfach 4848  
90026 Nürnberg  
Deutschland

Änderungen und Irrtümer vorbehalten  
© Siemens 2020

**[siemens.de/sce](https://www.siemens.de/sce)**