

**SIEMENS**

*Ingenuity for life*

24/7

Industry Online Support

Home

**Digitalisierung mit TIA Portal:  
Virtuelle Inbetriebnahme mit  
SIMATIC und Simulink**

**Anwendungsfall 2:  
Anbindung von Simulink-  
Modellen an SIMATIC PLCSIM  
Advanced über OPC UA**

SIMATIC S7-PLCSIM Advanced  
Simulink

<https://support.industry.siemens.com/cs/ww/de/view/109749187>

Siemens  
Industry  
Online  
Support



# Rechtliche Hinweise

## Nutzung der Anwendungsbeispiele

In den Anwendungsbeispielen wird die Lösung von Automatisierungsaufgaben im Zusammenspiel mehrerer Komponenten in Form von Text, Grafiken und/oder Software-Bausteinen beispielhaft dargestellt. Die Anwendungsbeispiele sind ein kostenloser Service der Siemens AG und/oder einer Tochtergesellschaft der Siemens AG („Siemens“). Sie sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit und Funktionsfähigkeit hinsichtlich Konfiguration und Ausstattung. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern bieten lediglich Hilfestellung bei typischen Aufgabenstellungen. Sie sind selbst für den sachgemäßen und sicheren Betrieb der Produkte innerhalb der geltenden Vorschriften verantwortlich und müssen dazu die Funktion des jeweiligen Anwendungsbeispiels überprüfen und auf Ihre Anlage individuell anpassen.

Sie erhalten von Siemens das nicht ausschließliche, nicht unterlizenzierbare und nicht übertragbare Recht, die Anwendungsbeispiele durch fachlich geschultes Personal zu nutzen. Jede Änderung an den Anwendungsbeispielen erfolgt auf Ihre Verantwortung. Die Weitergabe an Dritte oder Vervielfältigung der Anwendungsbeispiele oder von Auszügen daraus ist nur in Kombination mit Ihren eigenen Produkten gestattet. Die Anwendungsbeispiele unterliegen nicht zwingend den üblichen Tests und Qualitätsprüfungen eines kostenpflichtigen Produkts, können Funktions- und Leistungsmängel enthalten und mit Fehlern behaftet sein. Sie sind verpflichtet, die Nutzung so zu gestalten, dass eventuelle Fehlfunktionen nicht zu Sachschäden oder der Verletzung von Personen führen.

## Haftungsausschluss

Siemens schließt seine Haftung, gleich aus welchem Rechtsgrund, insbesondere für die Verwendbarkeit, Verfügbarkeit, Vollständigkeit und Mangelfreiheit der Anwendungsbeispiele, sowie dazugehöriger Hinweise, Projektierungs- und Leistungsdaten und dadurch verursachte Schäden aus. Dies gilt nicht, soweit Siemens zwingend haftet, z.B. nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der schuldhaften Verletzung des Lebens, des Körpers oder der Gesundheit, bei Nichteinhaltung einer übernommenen Garantie, wegen des arglistigen Verschweigens eines Mangels oder wegen der schuldhaften Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegen oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist mit den vorstehenden Regelungen nicht verbunden. Von in diesem Zusammenhang bestehenden oder entstehenden Ansprüchen Dritter stellen Sie Siemens frei, soweit Siemens nicht gesetzlich zwingend haftet.

Durch Nutzung der Anwendungsbeispiele erkennen Sie an, dass Siemens über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann.

## Weitere Hinweise

Siemens behält sich das Recht vor, Änderungen an den Anwendungsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in den Anwendungsbeispielen und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Ergänzend gelten die Siemens Nutzungsbedingungen (<https://support.industry.siemens.com>).

## Securityhinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z.B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter:

<https://www.siemens.com/industrialsecurity>.

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter: <http://www.siemens.com/industrialsecurity>.

# Inhaltsverzeichnis

<b>Rechtliche Hinweise</b>	<b>2</b>
<b>1 Einführung</b>	<b>4</b>
1.1 Überblick	4
1.2 Funktionsweise	5
1.3 Verwendete Komponenten	6
<b>2 Engineering</b>	<b>7</b>
2.1 Hardwarekonfiguration in TIA Portal	7
2.2 Codegenerierung mit SIMATIC Target 1500S	12
2.2.1 Einstellungen für Codegenerierung	12
2.2.2 Codierung mit SIMATIC Target 1500S	18
2.2.3 Beschreibung der generierten Funktionsbausteine	22
2.3 Simulationsmodell in Simulink	22
2.3.1 MATLAB-Funktion	22
2.3.2 MATLAB-Funktion einsetzen	24
2.4 STEP 7 Programm erstellen	25
2.5 Inbetriebnahme der Beispielapplikation	31
2.5.1 Projekt laden	32
2.5.2 Simulink-Simulation starten	33
<b>3 Bedienung</b>	<b>34</b>
3.1 PID-Regler bedienen	34
3.2 Regelergebnis beobachten	35
3.3 Anwender-Website nutzen	36
<b>4 Alternative Lösung</b>	<b>38</b>
<b>5 Code MATLAB-Funktion</b>	<b>39</b>
<b>6 Anhang</b>	<b>40</b>
6.1 Service und Support	40
6.2 Links und Literatur	41
6.3 Änderungsdokumentation	41

# 1 Einführung

## 1.1 Überblick

Die Software Simulink von MathWorks wird in der Automatisierungs- und Regelungstechnik häufig zur Simulation von Prozessen und zur Erstellung von Algorithmen verwendet. Die Anforderung besteht darin, in wenigen Schritten das Modell, den Algorithmus oder die Funktion in einer virtuellen Umgebung über PLCSIM Advanced oder hardwarebasierend über einen Software Controller zu simulieren.

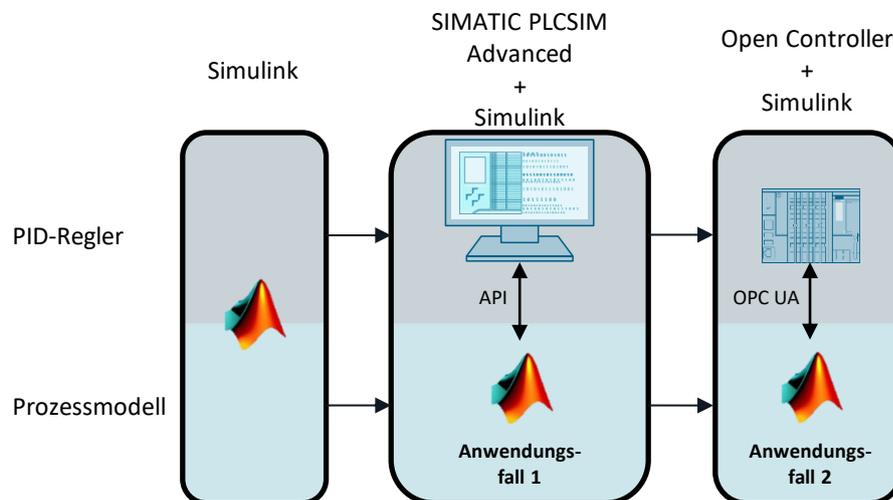
Dieses Anwendungsbeispiel zeigt Ihnen, wie Sie eine MATLAB-Funktion zur Kommunikation mit einem Software Controller über OPC UA erstellen. Die Einbindung der Funktion in das Simulink-Modell ermöglicht die Validierung und virtuelle Inbetriebnahme eines PID-Reglers, welcher auf einem virtuellen Controller läuft, im Kontext eines simulierten Prozesses in Simulink.

Die notwendigen Schritte zur Projektierung, Inbetriebnahme und Bedienung des Anwendungsbeispiels werden in diesem Dokument beschrieben.

Das gesamte Anwendungsbeispiel besteht aus folgenden Dokumenten:

- Hauptdokument: Überblick über die zwei Anwendungsfälle und das Simulink-Modell.
- Anwendungsfall 1: Anbindung von Simulink-Modellen an SIMATIC PLCSIM Advanced über API.
- Anwendungsfall 2: Anbindung von Simulink-Modellen an SIMATIC S7-1500 Software Controller über OPC UA (dieses Dokument).

Abbildung 1-1: Anwendungsfälle im Überblick



## 1.2 Funktionsweise

Mit dem Add-On OPC Toolbox für MATLAB lassen sich Funktionen in Simulink einbinden, welche einen Datenaustausch mit einem OPC UA Server ermöglichen. Mit diesem Anwendungsbeispiel wird Ihnen eine fertig ausprogrammierte MATLAB-Funktion "Read\_OPC\_Func" zur Verfügung gestellt. Die Funktion ermöglicht das Schreiben und Lesen von Variablen eines Controllers über OPC UA.

Die Funktion wird in Simulink über den Block "Interpreted MATLAB Fcn" integriert und in einem Regelkreis anstelle eines Reglers platziert.

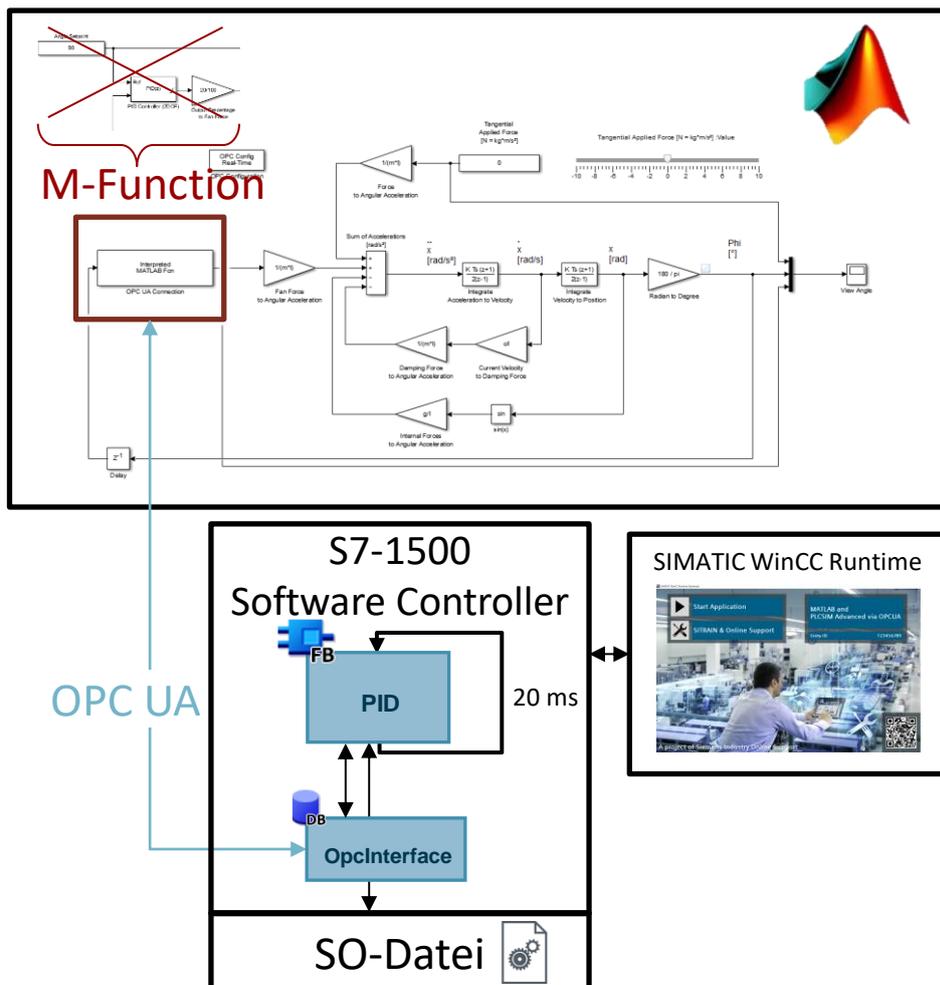
Das Simulink-Modell eines PID-Reglers wird mit SIMATIC Target 1500S codiert und auf einem S7-1500 Software Controller zum Ablaufen gebracht. Damit dieses anschließend im Kontext des Simulationsmodells getestet werden kann.

Der Variablen austausch erfolgt über einen globalen Datenbaustein des virtuellen Controllers. Dabei findet der Austausch im Zyklus des Simulationsmodells statt.

Die Visualisierung des Reglerergebnisses und Steuerung des Reglers erfolgt über ein simuliertes HMI-Bild der WinCC Runtime.

**Hinweis** Die Projektierung des HMIs ist nicht Bestandteil dieses Anwendungsbeispiels.

Abbildung 1-2



## 1.3 Verwendete Komponenten

Dieses Anwendungsbeispiel wurde mit diesen Hard- und Softwarekomponenten erstellt:

Tabelle 1-1: Hard- und Software

Komponente	Artikelnummer / Hinweis / Link
(R2019b) MATLAB V9.7 MATLAB Coder V4.3 Simulink V10.0 Simulink Coder V9.2 OPC Toolbox V4.0.8	MathWorks Online Dokumentation: <a href="http://de.mathworks.com/help/">http://de.mathworks.com/help/</a>
STEP 7 Professional V16	6ES7822-1..06-.. Handbuch: <a href="https://support.industry.siemens.com/cs/ww/de/view/109773506">https://support.industry.siemens.com/cs/ww/de/view/109773506</a>
SIMATIC S7-1500 ODK 1500S V2.5 SP1	6ES7806- 2CD03-0YA0 Handbuch: <a href="https://support.industry.siemens.com/cs/ww/de/ps/13914/man">https://support.industry.siemens.com/cs/ww/de/ps/13914/man</a>
SIMATIC Target 1500S for Simulink V4.0	6ES7823-1BE03-0YA5 Handbuch: <a href="https://support.industry.siemens.com/cs/ww/de/ps/24443/man">https://support.industry.siemens.com/cs/ww/de/ps/24443/man</a>
SIMATIC ET 200SP Open Controller CPU 1515SP PC2	6ES7677-2DB42-0GB0 Handbuch SIMATIC S7-1500 CPU 150xS <a href="https://support.industry.siemens.com/cs/ww/de/view/109740725">https://support.industry.siemens.com/cs/ww/de/view/109740725</a> Handbuch: SIMATIC ET 200SP Open Controller CPU 1515SP PC <a href="https://support.industry.siemens.com/cs/ww/de/view/109758551">https://support.industry.siemens.com/cs/ww/de/view/109758551</a>

Dieses Anwendungsbeispiel besteht aus folgenden Komponenten:

Tabelle 1-2: Komponenten des Anwendungsbeispiels

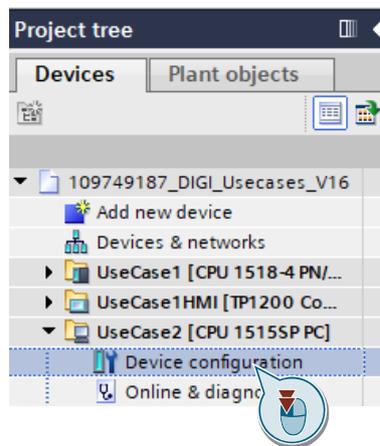
109749187_DIGI_Usecases_MAIN_DOC_V20_de.pdf	Hauptdokument.
109749187_DIGI_Usecases_OPC_DOC_V20_de.pdf	Dieses Dokument.
109749187_DIGI_Usecases_TIAV16_PROJ_V20.zip	TIA Portal-Projekt für die Anwendungsfälle 1&2.
109749187_DIGI_Usecases_Simulink_PROJ_V20.zip	Simulink-Modelle für die Anwendungsfälle 1&2.

## 2 Engineering

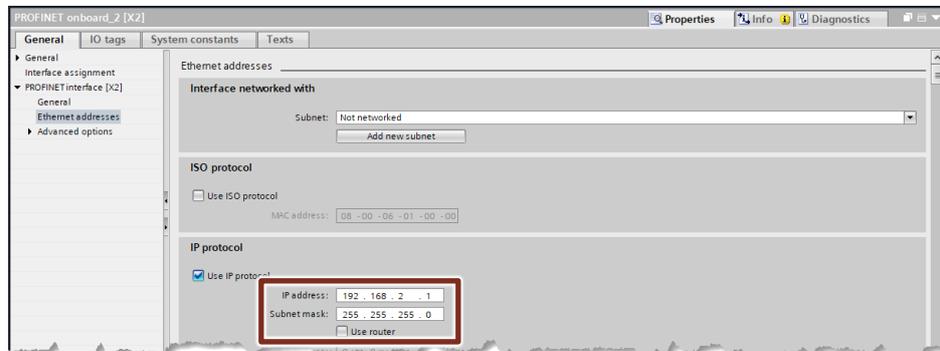
### 2.1 Hardwarekonfiguration in TIA Portal

Führen Sie folgende Schritte durch, um die Hardwarekonfiguration für Ihre PLC einzurichten.

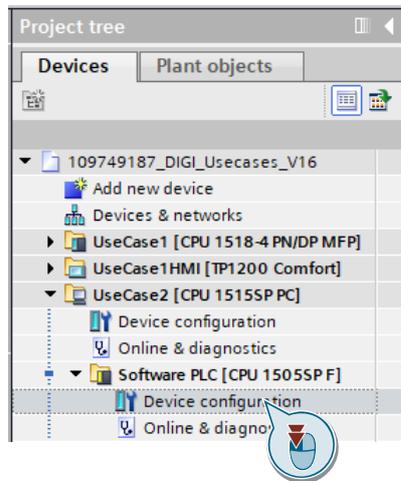
1. Fügen Sie im Projekt eine CPU 1515SP PC2 ein. Zum Beispiel mit einer CPU 1505 SP V2.5 mit einer CPU 1505SP F V2.7. Benennen Sie die PLC "UseCase2".
2. Öffnen Sie mit einem Doppelklick die Gerätekonfiguration ("Device configuration") der CPU.



3. Stellen Sie im Inspektorfenster unter "Einstellungen > Allgemein > PROFINET-Schnittstelle [X2] > Ethernet-Adressen" ("Properties > General > PROFINET interface [X2] > Ethernet addresses") die IP-Adresse für Ihre CPU ein.



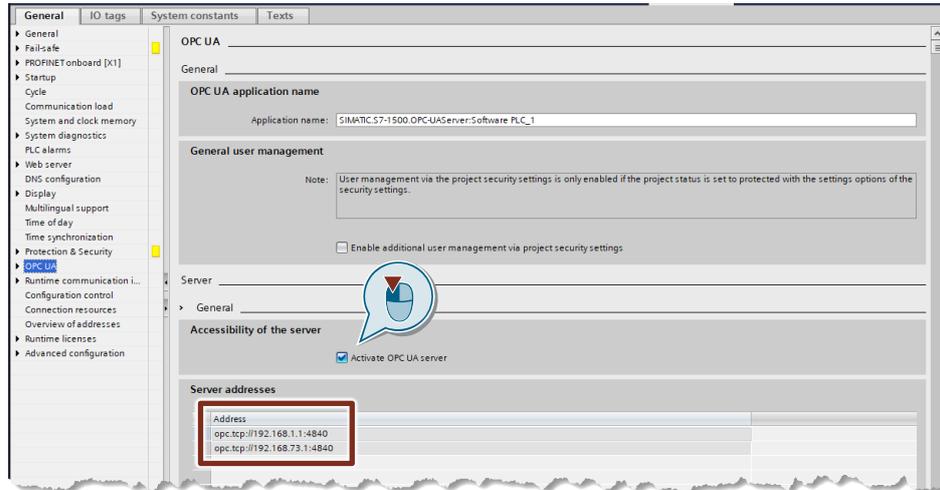
- Öffnen Sie mit einem Doppelklick die Gerätekonfiguration ("Device configuration") der Software PLC.



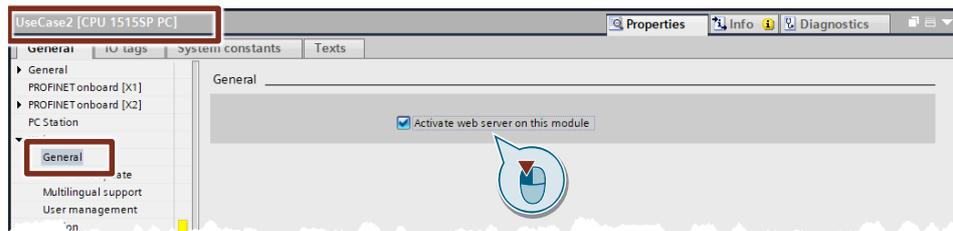
- Stellen Sie im Inspektorfenster unter "Einstellungen > Allgemein > PROFINET-Schnittstelle [X1] > Ethernet-Adressen" ("Properties > General > PROFINET interface [X1] > Ethernet addresses") die IP-Adresse für Ihre CPU ein.



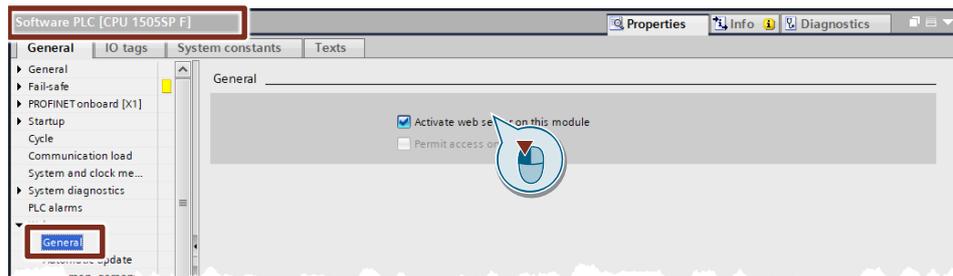
6. Aktivieren Sie unter "OPC UA" den OPC UA Server für Ihre CPU. Unter "Server-Adressen:" ("Server addresses:") finden Sie die IP-Adresse und die Portnummer, unter der der Server erreicht werden kann.



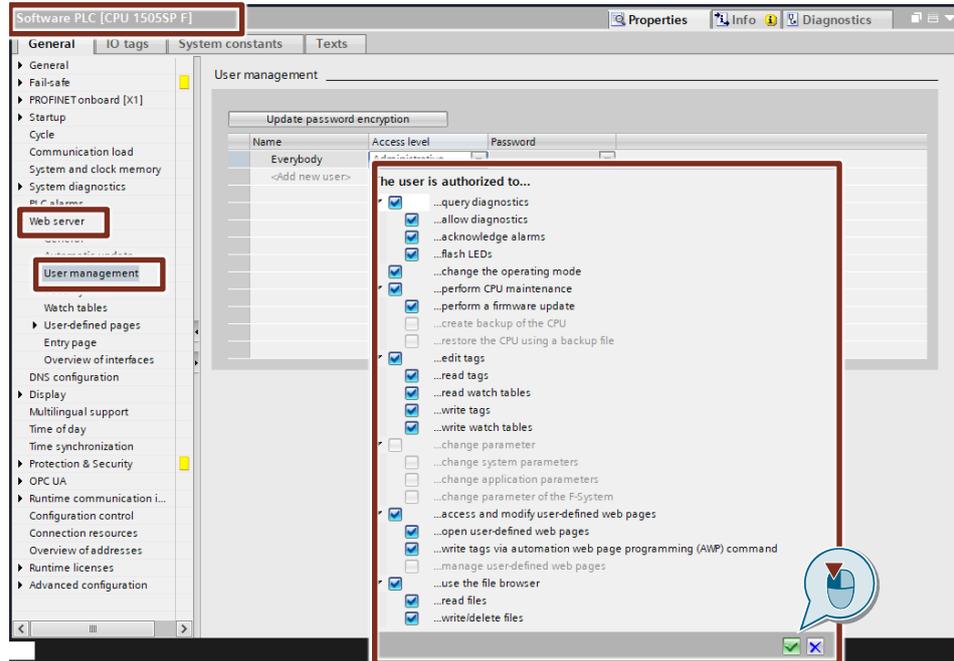
7. Stellen Sie unter "Runtime-Lizenzen > OPC UA" ("Runtime licences > OPC UA") den Typ der erworbenen Lizenz, entsprechend dem Typ der benötigten Lizenz ein.
8. Aktivieren Sie den Webserver für das PC-System. Quittieren Sie den Security-Dialog.



9. Klicken Sie auf die "CPU 1505SP F" und aktivieren Sie "Webserver auf dieser Baugruppe aktivieren" ("Activate web server on this module"). Quittieren Sie den Security-Dialog.



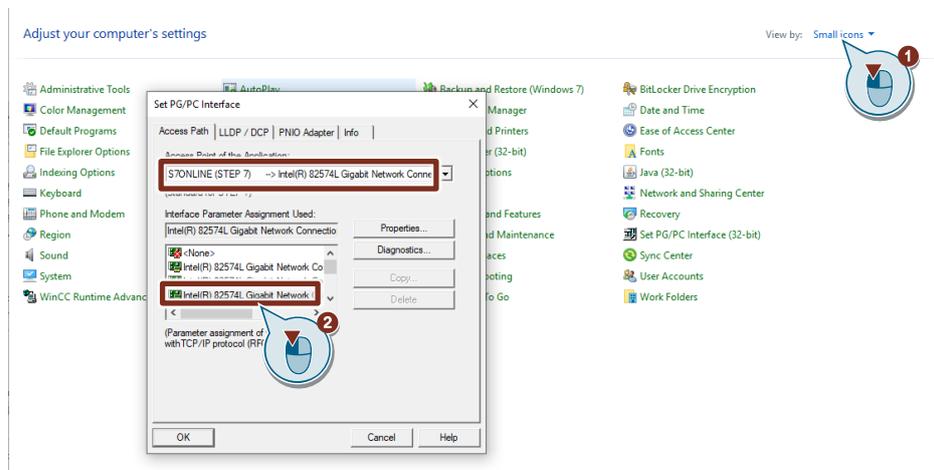
10. Navigieren Sie zu "Webserver > Benutzerverwaltung" ("Web server > User management") und aktivieren Sie unter "Zugriffsstufe" ("Access level") alle Optionskästchen für vollen Zugriff. Erstellen Sie bei Bedarf ein Passwort.



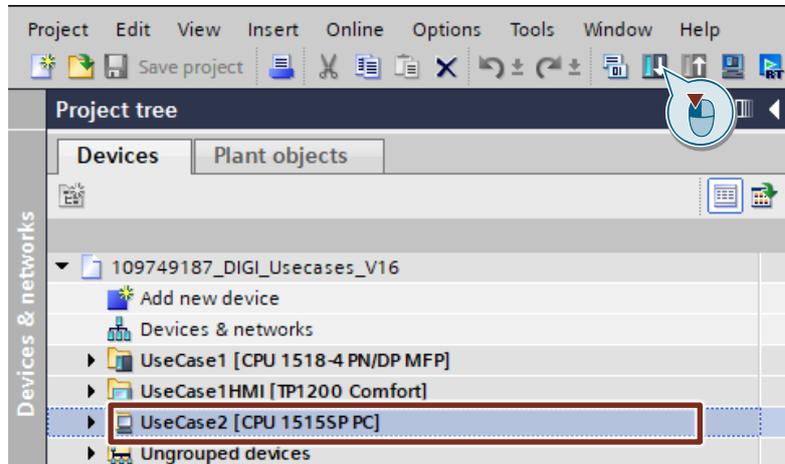
**Hinweis**

Um die SO-Datei in den Webserver zu laden oder zu löschen, müssen die Parameter "...Dateien zu lesen" ("read files") und "...Dateien zu schreiben/löschen" ("write/delete files") aktiviert sein. Alle weiteren Parameter sind optional.

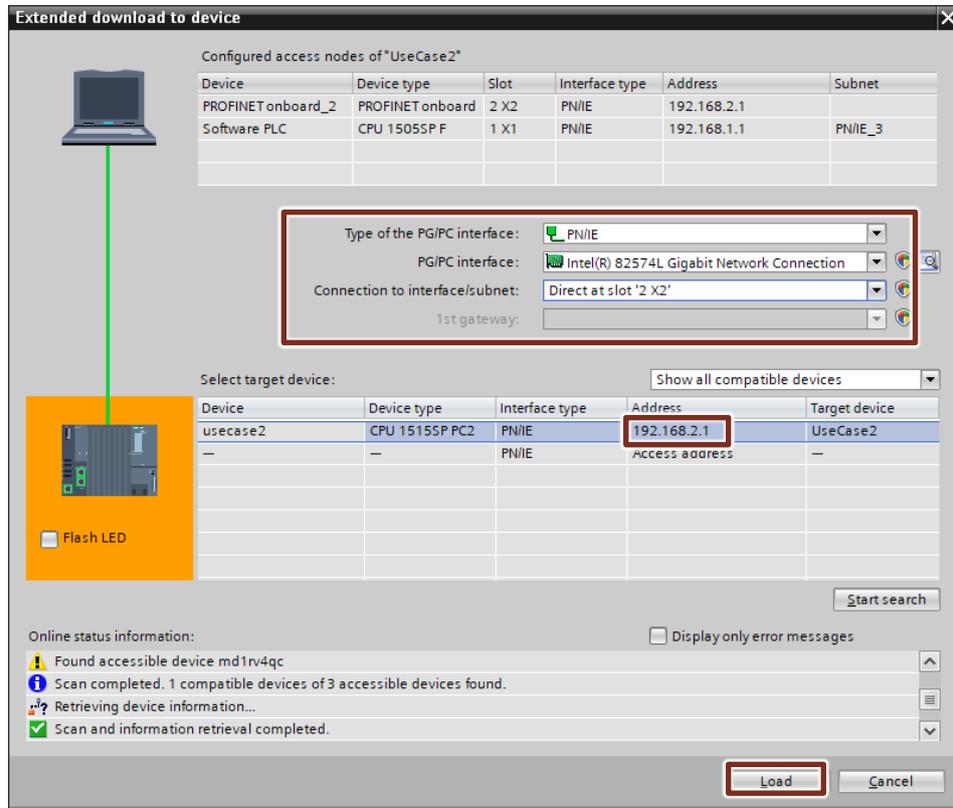
11. Stellen Sie die PG/PC Schnittstelle ein.
  1. Öffnen Sie hierfür ihre Systemsteuerung
  2. Stellen sie die Anzeige auf "Kleine Symbole"
  3. Wählen Sie "PG/PC Schnittstelle einstellen"
  4. Wählen Sie für die S7ONLINE (STEP 7) die Ethernet-Schnittstelle ihres PG/PCs mit TCP/IP.1 aus.



12. Laden Sie die Hardwarekonfiguration auf den ET 200SP Open Controller.
  - Selektieren Sie "Usecase2" in der Projektnavigation.
  - Klicken Sie auf die Schaltfläche "Laden in Gerät" ("Download to device").



- Wählen Sie "PN/IE" aus.
- Wählen Sie die Ethernet-Schnittstelle ihres PG/PCs aus.
- Wählen Sie "Direkt an Steckplatz `2 X2" ("Direct at Slot `2 X2") aus.
- Klicken Sie auf "Suche starten" ("Start search").
- Klicken Sie auf "Laden" ("Load").



13. Bestätigen Sie alle folgenden Dialoge bis der Download abgeschlossen ist.

**Hinweis**

Wenn Sie die Hardwarekonfiguration des SIMATIC ET 200SP Open Controllers das erste Mal laden, müssen Sie die Windows-Schnittstelle (hier IP: 192.168.2.1) verwenden.

Achten Sie darauf, dass die IP-Adresse der Windows-Schnittstelle der Parametrierung im TIA Portal entspricht.

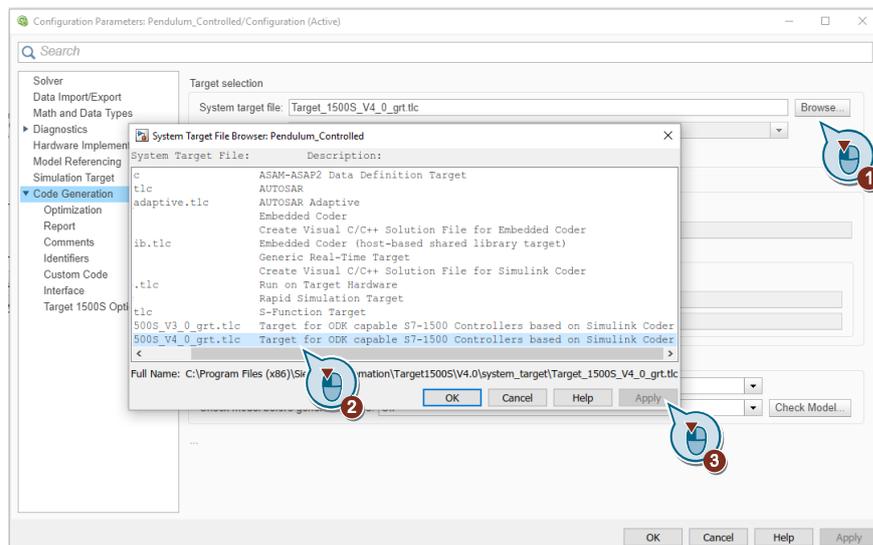
## 2.2 Codegenerierung mit SIMATIC Target 1500S

Folgendes Kapitel beschreibt, wie Sie das zu simulierende Modell des PID-Reglers mit Simulink für die Codierung mit Target 1500S vorbereiten und diese anschließend durchführen.

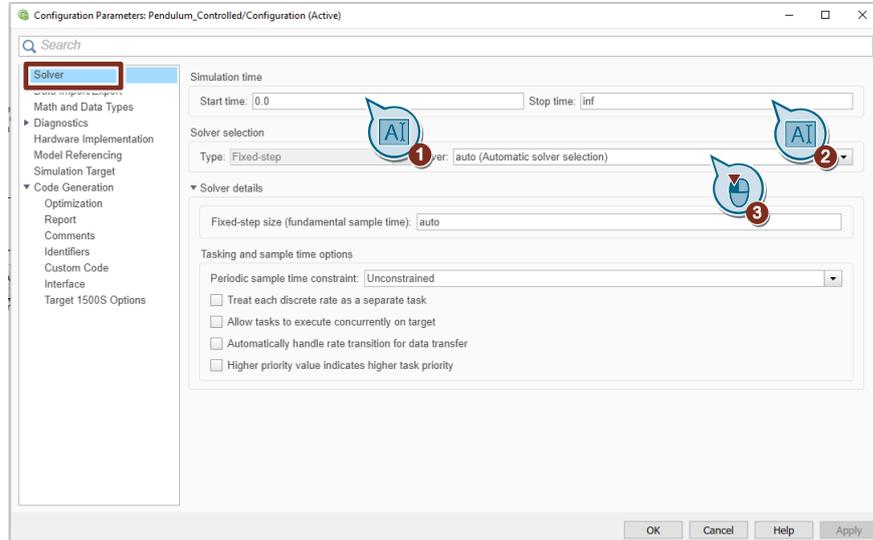
### 2.2.1 Einstellungen für Codegenerierung

Führen Sie folgende Einstellungen der Codegenerierung für Ihr Modell durch.

1. Öffnen Sie Ihr Modell "Pendulum\_Controlled.slx".
2. Klicken Sie in der Menüleiste auf "Code > Settings > C/C++ Code generation settings...".
3. Navigieren Sie zu "Code Generation".
  1. Öffnen Sie über die Schaltfläche "Browse..." das Target-Auswahlfenster.
  2. Wählen Sie "Target\_1500S\_V4\_0\_grt.tlc" aus.
  3. Bestätigen Sie die Einstellung mit "Apply".

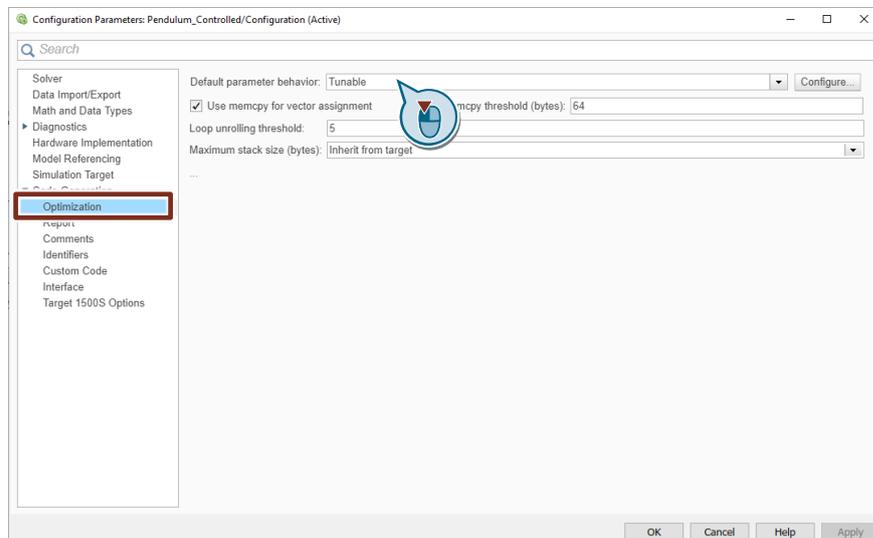


4. Navigieren Sie zu "Solver".
  1. Belassen Sie "Start time" bei "0".
  2. Ändern Sie "Stop time" auf "inf". Damit ist sichergestellt, dass der External Mode ohne definiertes Ende verwendet wird.
  3. Belassen Sie "Solver" defaultmäßig auf "auto (Automatic solver selection)".



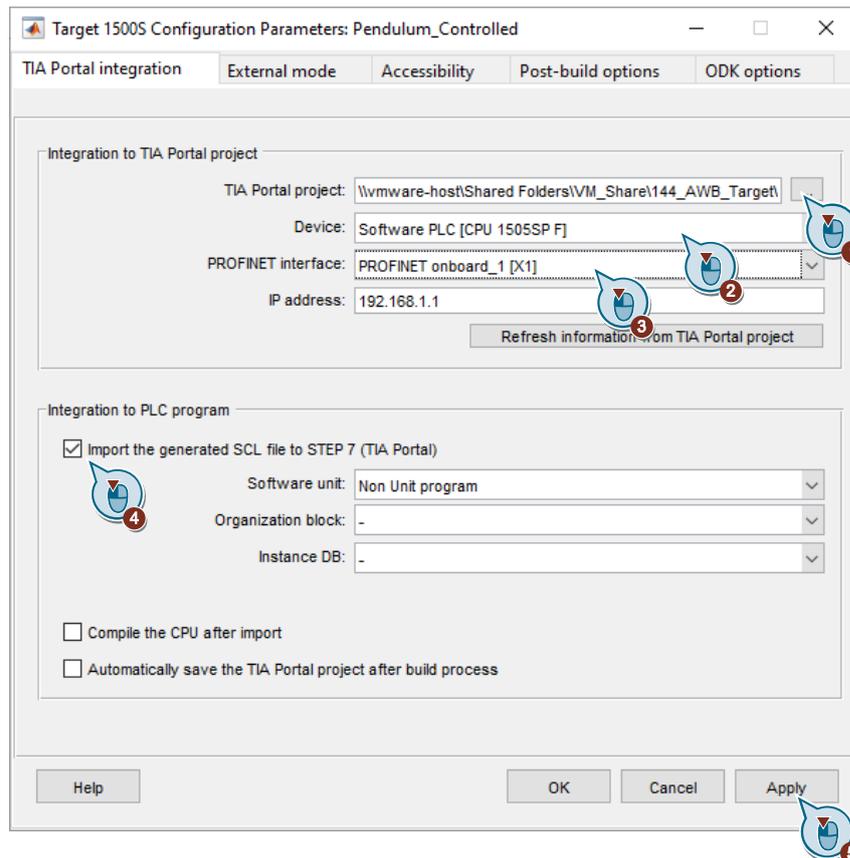
5. Navigieren Sie zu "Code Generation > Optimization".
  - Setzen Sie den Parameter "Default parameter behaviour:" auf "Tunable".

Mit dieser Einstellung haben Sie nach der Codegenerierung Zugriff mit STEP 7 (TIA Portal) auf alle internen Parameter des Modells.

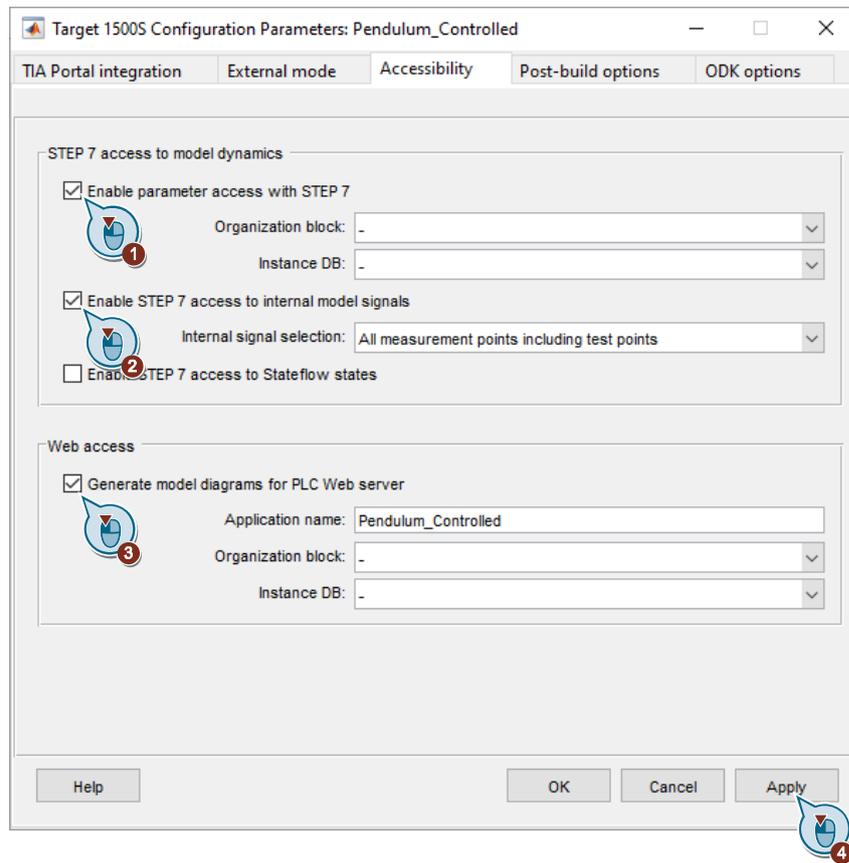


6. Klicken Sie auf "Target 1500S Options > Target 1500s Options".

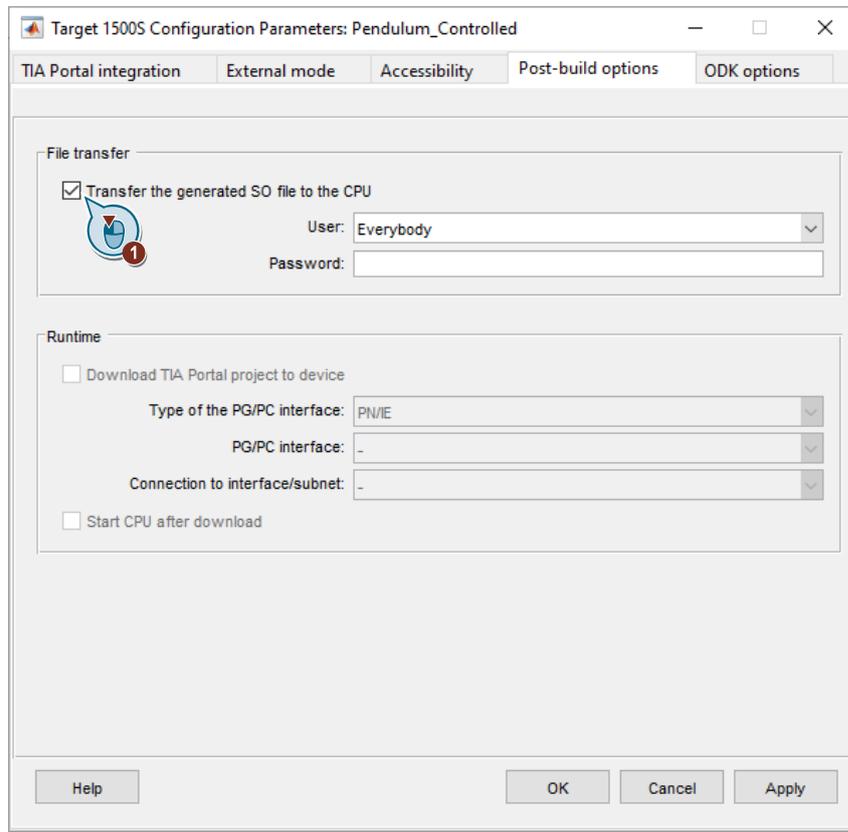
7. Navigieren Sie zu "TIA Portal integration"
  1. Wählen Sie unter "TIA Portal project" Ihr TIA Projekt aus.
  2. Stellen Sie bei "Device" die Software PLC ein.
  3. Wählen Sie bei "PROFINET interface" die [X1]-Schnittstelle. Die IP-Adresse wird automatisch angepasst.
  4. Überprüfen Sie, dass der Haken bei "Import the generated SCL file to STEP 7" gesetzt ist.
  5. Bestätigen Sie mit "Apply".



8. Navigieren Sie zu "Accessibility"
  1. Setzen Sie den Haken bei "Enable parameter access with Step 7"
  2. Setzen Sie den Haken bei "Enable Step 7 access to internal model signals"
  3. Überprüfen Sie, dass der Hake bei "Generate model diagrams for PLC Web server" gesetzt ist.
  4. Bestätigen Sie mit "Apply".

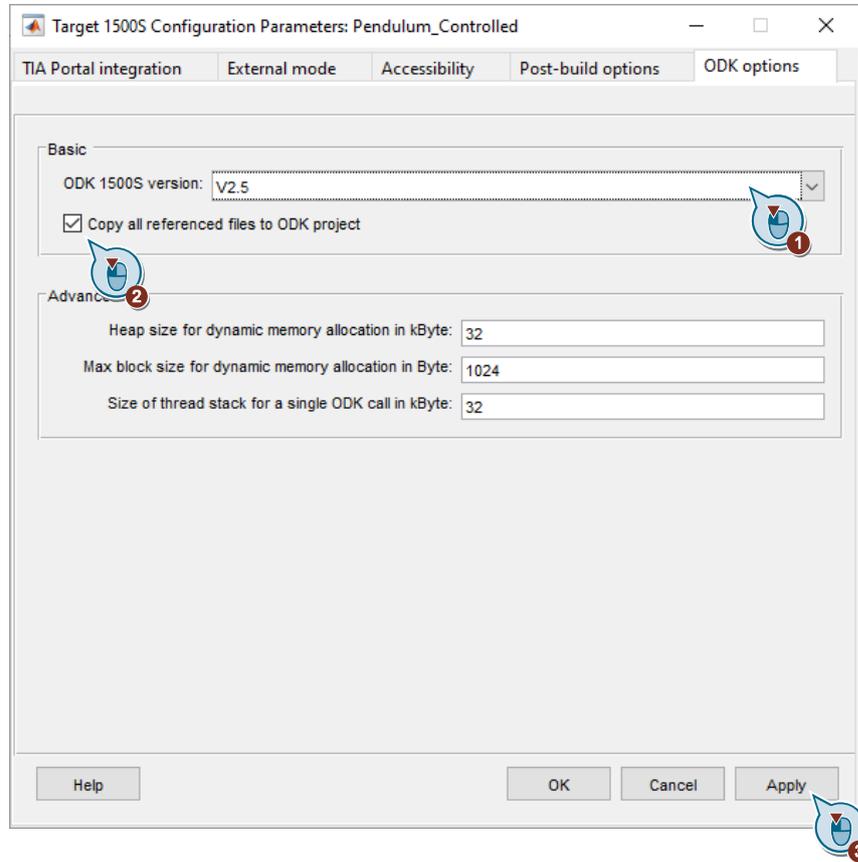


9. Navigieren Sie zu "Post-build options" und stellen Sie sicher, dass der Haken bei "Transfer the generated SO file to the CPU" gesetzt ist.



10. Navigieren Sie zu "ODK options"

1. Wählen Sie bei "ODK 1500S version" V2.5 aus.
2. Setzen Sie den Haken bei "Copy all referenced files to ODK project"
3. Bestätigen Sie mit "Apply".



**Hinweis**

Die Parameter "Heap size for dynamic memory allocation in kByte" und "Max block size for dynamic memory allocation in Byte" müssen nur bei Problemen mit der Code-Generierung angepasst werden.

Weitere Informationen finden Sie dazu im Handbuch Target 1500S

<https://support.industry.siemens.com/cs/ww/de/ps/24443/man>

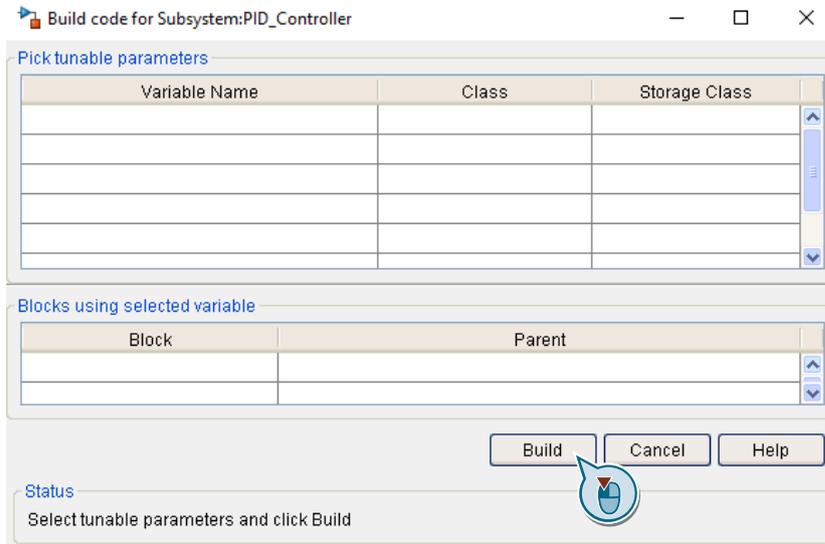
und im Handbuch SIMATIC S7-1500 ODK 1500S

<https://support.industry.siemens.com/cs/ww/de/ps/13914/man>.

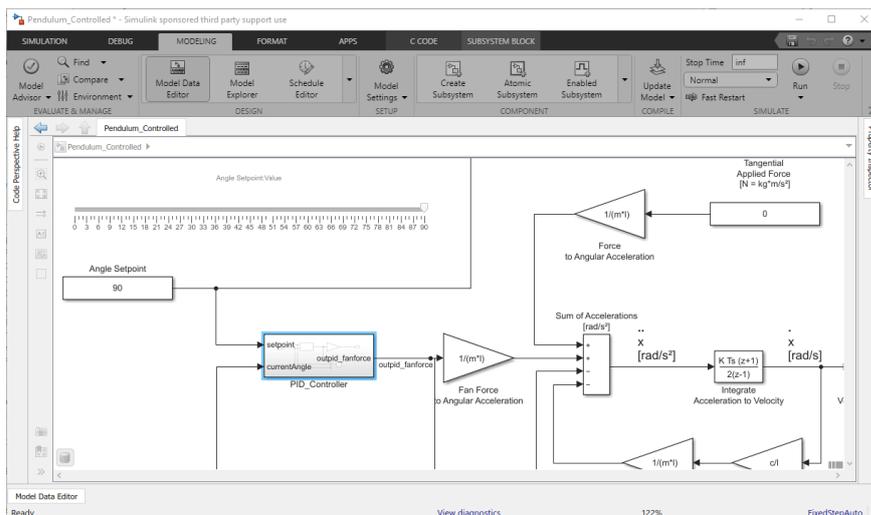
11. Schließen Sie den Dialog mit "OK".

## 2.2.2 Codierung mit SIMATIC Target 1500S

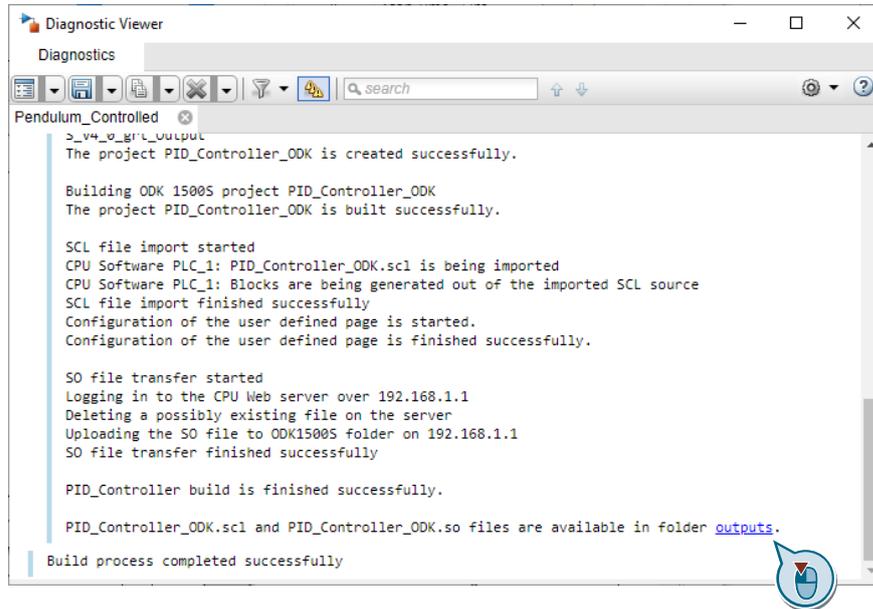
1. Öffnen Sie Ihr eigenes oder das mitgelieferte Modell "Pendulum\_Controlled.slx".
2. Wählen Sie mit Rechtsklick das Subsystem "PID\_Controller" an und klicken Sie in dem Auswahlfenster auf "Code > C/C++ Code > Build This Subsystem".
3. Das Fenster "Build code for Subsystem:PID\_Controller" öffnet sich. Bestätigen Sie dieses mit "Build".



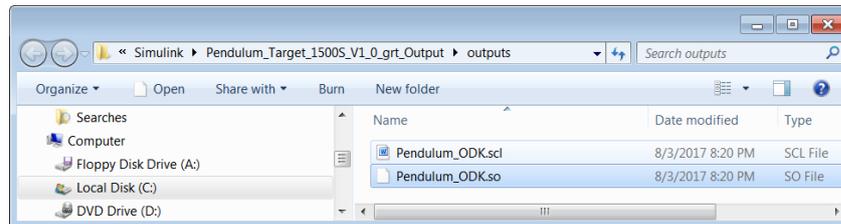
- Während der Codegenerierung ist Simulink nicht bedienbar.
  - SIMATIC Target 1500S übersetzt das Simulink-Modell "Pendulum\_Controlled" mit seinen In- und Output-Variablen in "C/C++" Code.
  - Aus diesem Quellcode wird anschließend mit dem ODK Compiler des SIMATIC ODK 1500S die SO-Datei und SCL-Quelle generiert.
4. Sobald die Codegenerierung durchgeführt wird, können Sie im Simulink-Fenster auf den Link "View..." in der Statuszeile klicken.



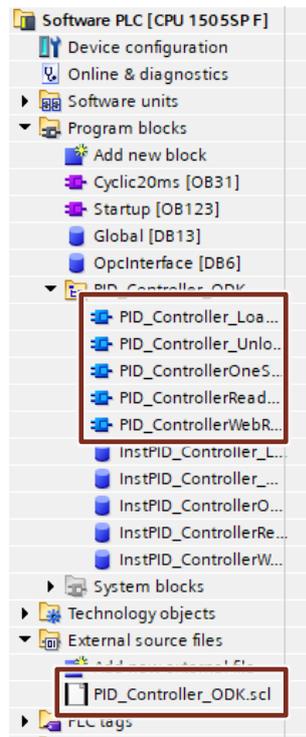
5. Der "Diagnostic Viewer" öffnet sich.
  - Klicken Sie auf den Link "outputs". Sie gelangen direkt in das Speicherverzeichnis der SO-Datei und der SCL-Quelle.



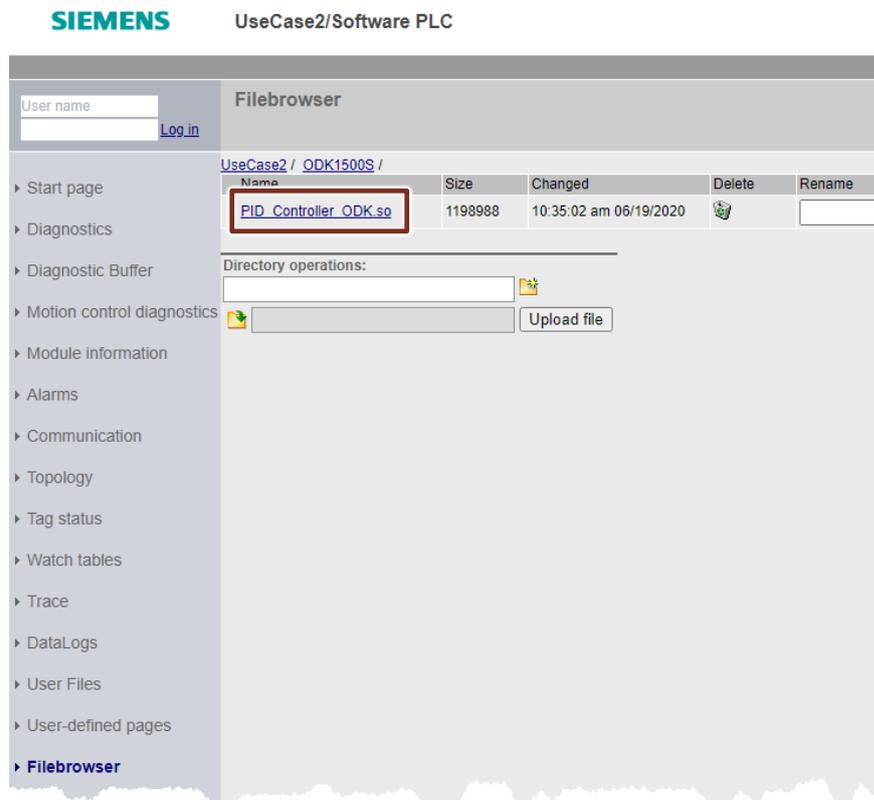
6. Im Verzeichnis des Simulink-Modells "...Pendulum\_Target\_1500S\_V1\_0\_grt\_Output\outputs" befinden sich die generierten Dateien.



- Überprüfen Sie im TIA Portal Projekt, ob die SCL-Dateien automatisch importiert worden sind. Andernfalls fügen Sie die aus Matlab generierte SCL-Datei manuell hinzu. Folgen Sie dabei den Schritten [SCL-Quelle importieren in Step 7 Programm erstellen](#)



8. Überprüfen Sie ob die SO-Datei automatisch in den Webserver unter "Filebrowser > ODK1500S" abgelegt wurde. Andernfalls vollziehen Sie diesen Upload manuell ("SO-Datei auf Webserver laden" in "[Step 7 Programm erstellen](#)").



### 2.2.3 Beschreibung der generierten Funktionsbausteine

Folgende Tabelle beschreibt alle Bausteine, die aus der SCL-Quelle generiert wurden.

Tabelle 2-1: Bausteinerklärung

Generierter Baustein	Funktionsbeschreibung
FB PID_Controller_Unload	Baustein löscht die SO-Datei aus dem Arbeitsspeicher der CPU.
FB PID_Controller_Load	Baustein lädt die SO-Datei vom Webserver in den Arbeitsspeicher der CPU.
FB PID_ControllerOneStep	Baustein ruft die SO-Datei auf. Die Schnittstellen entsprechen dem ursprünglichen Simulink-Modell.
FB PID_ControllerWebReadWrite (optional)	Baustein wird zum Lesen oder Schreiben durch die Benutzerdefinierte Webseite aufgerufen.
FB PID_ControllerIrReadWriteParameters (optional)	Baustein wird zum Lesen oder Schreiben der zu manipulierenden Modellparameter aufgerufen.

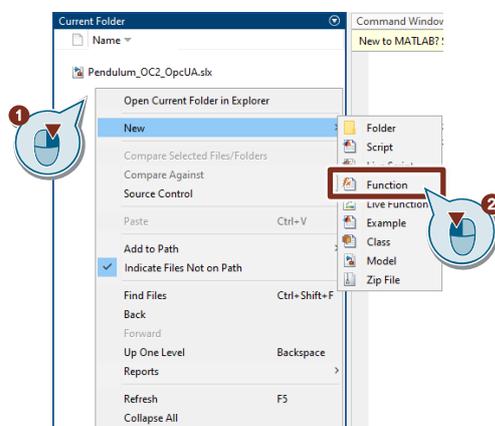
## 2.3 Simulationsmodell in Simulink

Folgendes Kapitel beschreibt, wie Sie eine benutzerdefinierte MATLAB-Funktion "Read\_OPC\_Func" für einen Datenaustausch mit einem OPC-Server in MATLAB erstellen und über den Block "Interpreted MATLAB Function" in das Simulink-Modell einbinden.

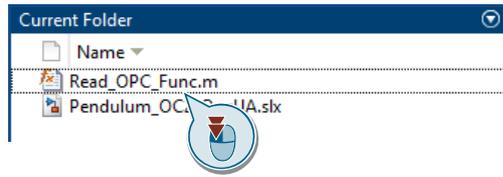
### 2.3.1 MATLAB-Funktion

Führen Sie folgende Schritte durch, um eine MATLAB-Funktion zu erstellen.

1. Navigieren Sie in MATLAB zum Ablageort Ihres Simulink-Modells und fügen Sie dort eine neue Funktion ein.



- Benennen Sie die Funktion "Read\_OPC\_Func" und öffnen Sie mit einem Doppelklick auf die Funktion den Funktions-Editor.



- Geben Sie im Funktions-Editor folgenden Code ein. Eine Kopiervorlage finden Sie im Anhang unter Kapitel 5.

Passen Sie folgende Parameter entsprechend Ihrer Anwendung an:

- Geben Sie hier die IP-Adresse und den Port des OPC-Servers an.
- Geben Sie hier den Namen des Schnittstellen-DBs an, über welchen der Datenaustausch zwischen der PLC und dem Modell stattfindet.
- Geben Sie hier den Namen der Variablen an, dessen Wert die Funktion aus dem Schnittstellen-DB lesen soll.
- Geben Sie hier den Namen der Variablen an, dessen Werte die Funktion in dem Schnittstellen-DB schreiben soll.

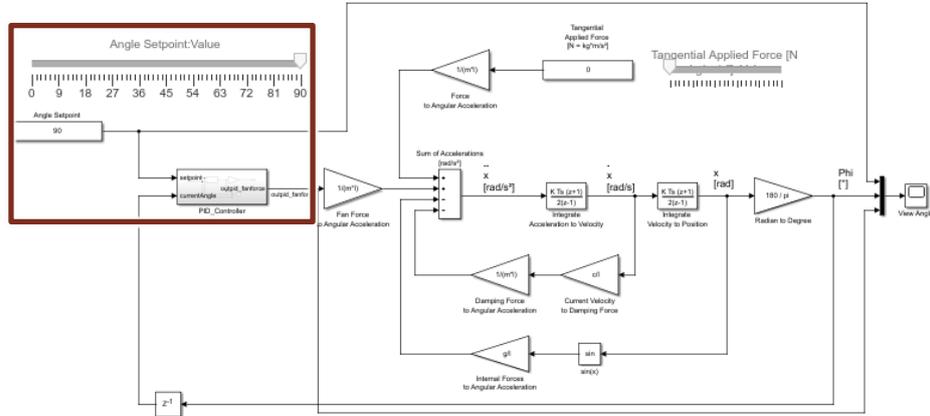
```

1 function [x] = Read_OPC_Func(y)
2
3 % variables
4 persistent init_Server;
5 persistent init_Nodes;
6 persistent uaClient;
7 persistent Var_Node_In;
8 persistent Var_Node_Out;
9 persistent testVal;
10
11 % initialize variables
12 if (isempty(init_Server))
13     testVal = 0;
14     init_Server = 0;
15     init_Nodes = 0;
16 end
17
18 % OPC UA server (PLC) address and connecting client (Simulink) to the
19 % server
20 if init_Server == 0
21     init_Server = 1;
22     uaClient = opcua('192.168.1.1',4840);
23     connect(uaClient);
24 end
25
26 % define variable nodes in the server
27 if uaClient.isConnected == 1 && init_Nodes == 0
28     init_Nodes = 1;
29     % read out the variables of the OPC UA server
30     Var_Node_In = opcuanode(3, ""OpcInterface"."fanForce"", uaClient);
31     Var_Node_Out = opcuanode(3, ""OpcInterface"."currentAngle"", uaClient);
32 end
33
34 % read and write variables of the server
35 if uaClient.isConnected == 1 && init_Nodes == 1
36     % read "fanForce" value from server and store in "val"
37     [val, ~, ~] = readValue(uaClient, Var_Node_In);
38     % assign input y of the function to "currentAngle" variable
39     writeValue(uaClient, Var_Node_Out, y);
40     % assign "val" to variable "testVal"
41     testVal = val;
42 end
43
44 % assign "fanForce" ("testVal") value to the output x of the function
45 x = double(testVal);
46 end
    
```

- Speichern Sie die erstellte Funktion.

### 2.3.2 MATLAB-Funktion einsetzen

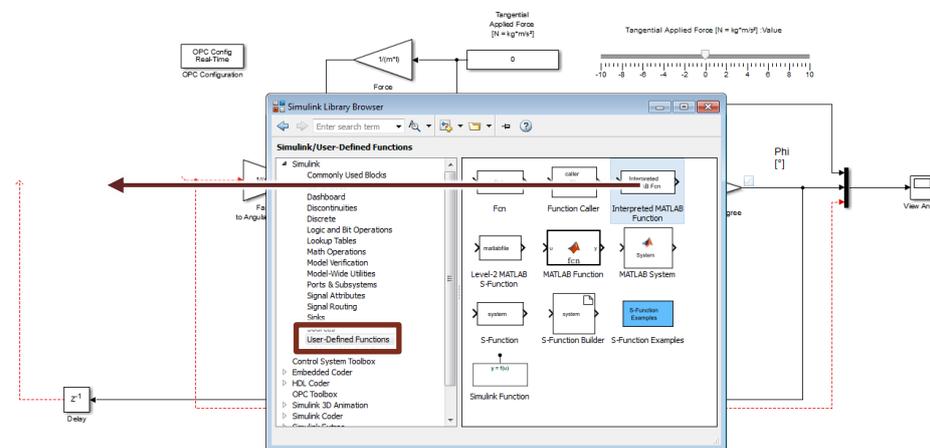
1. Öffnen Sie Ihr Simulink-Modell mit dem Regelkreis und entfernen Sie den PID-Regler und die markierten Funktionen, die mit vom virtuellen Controller ausgeführt werden.



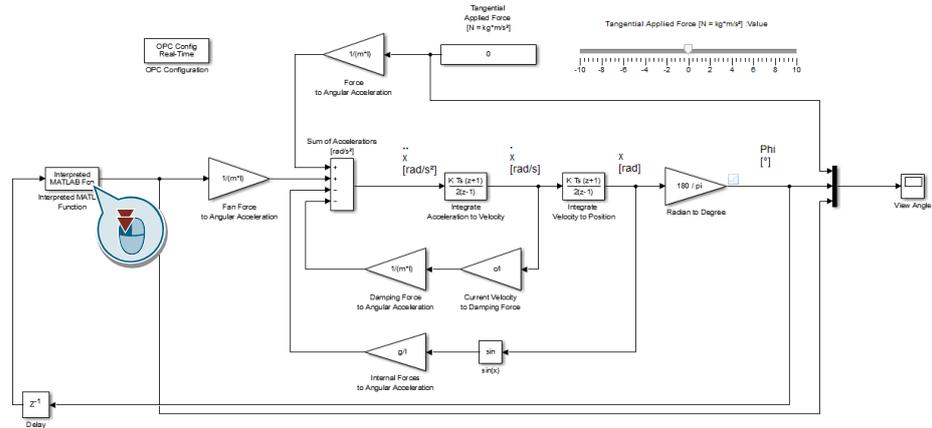
2. Öffnen Sie den Simulink Library Browser.



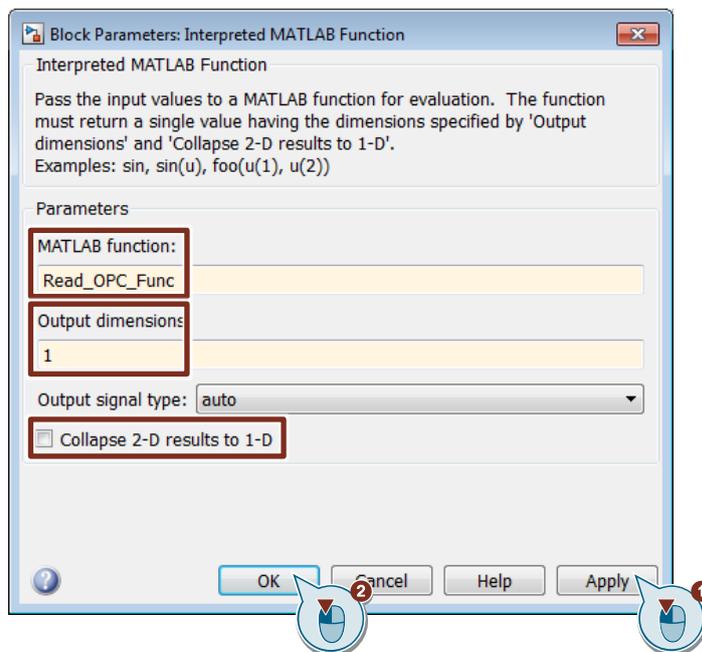
3. Ziehen Sie per Drag&Drop den Block "Interpreted MATLAB Function" aus dem Verzeichnis "Simulink > User-Defined Functions" in das Simulink-Modell und verbinden Sie diesen mit dem Modell.



- Öffnen Sie mit einem Doppelklick die Blockeigenschaften des Block "Interpreted MATLAB Function".



- Geben Sie als Funktions-Namen den Namen Ihrer Funktion ein, hier "Read OPC Func", und stellen Sie weitere Parameter, wie in der folgenden Abbildung dargestellt ein. Übernehmen Sie die Änderung mit "Apply" und bestätigen Sie mit "OK".



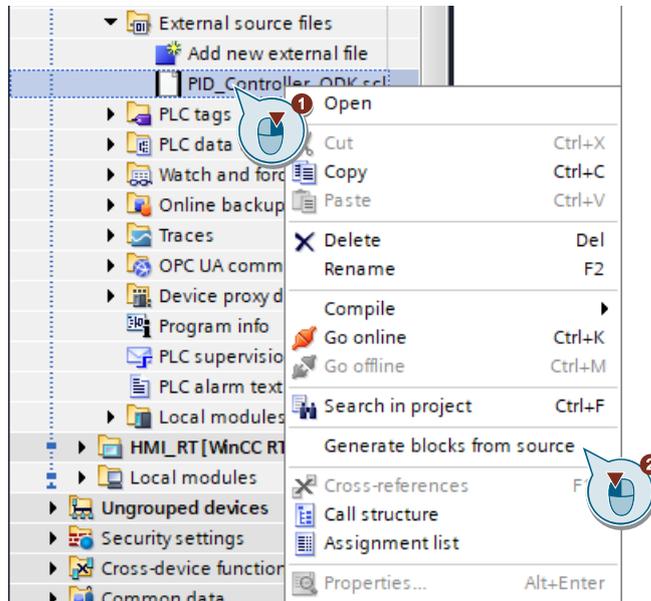
## 2.4 STEP 7 Programm erstellen

### SCL-Quelle importieren

Falls das automatische Importieren der SCL-Quelle nicht funktioniert hat, führen Sie folgende Schritte durch, um die SCL-Quelle, welche durch das Target 1500S generiert wurde, zu importieren.

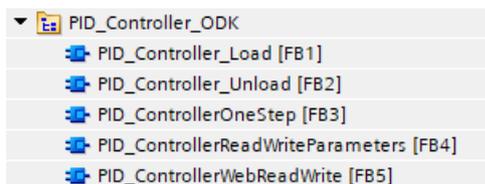
- Navigieren Sie in der "Projektnavigation" ("Project tree") zu "Externe Quelle" ("External source files").
- Klicken Sie auf "Neue externe Datei hinzufügen" ("Add new external file").

3. Wählen Sie die generierte SCL-Quelle "PID\_Controller\_ODK.scl" aus und importieren Sie die Datei in das S7-Programm.
4. Generieren:
  1. Klicken Sie mit der rechten Maustaste auf die Datei "PID\_Controller\_ODK.scl".
  2. Klicken Sie auf "Bausteine aus Quelle generieren" ("Generate blocks from source") und bestätigen Sie den Dialog.

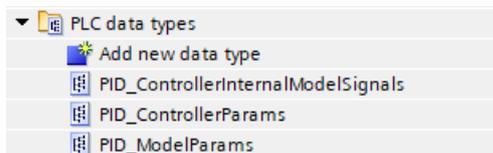


Das TIA Portal generiert aus der SCL-Quelle die von Target 1500S erstellten Bausteine.

5. Unter "Programmbausteine" ("Program blocks") finden Sie die generierten Bausteine.



6. Unter "PLC-Datentypen" ("PLC data types") finden Sie die generierten PLC-Datentypen.



### OBs und globale DBs

Führen Sie folgende Schritte durch, um die OBs und globalen DBs zu erstellen.

1. Fügen Sie Ihrem S7-Programm folgende Bausteine hinzu:
  - Cyclic Interrupt OB: "Cyclic20ms" (mit einer Weckalarm-Zeit von 20 ms)
  - Startup OB: "Startup"
  - Global DB: "Global"
  - Global DB: "OpclInterface"

#### Hinweis

Der Zyklus des OB "Cyclic20ms" muss dem Zyklus Ts des Simulink-Modell entsprechen (hier 0.02 bzw. 20 ms) entsprechen.

2. Öffnen Sie den DB "Global" und fügen Sie folgende Variablen hinzu. Die Variablen werden durch eine Struktur zusammengefasst.
  - pendulumController [Struct]
    - angleSetpoint [Int]
    - currentAngle [LReal]
    - fanforce [LReal]
    - operatingMode [Int] (Startwert = 3)
  - internalsignals [PID\_ControllerInternalModelSignals]
  - modelParams [PID\_ControllerParams]

Global			
	Name	Data type	Start value
1	Static		
2	pendulumController	Struct	
3	angleSetpoint	LReal	0.0
4	currentAngle	LReal	0.0
5	fanforce	LReal	0.0
6	operatingMode	Int	3
7	internalsignals	*PID_ControllerInternalModelSignals*	
8	modelParams	*PID_ControllerParams*	

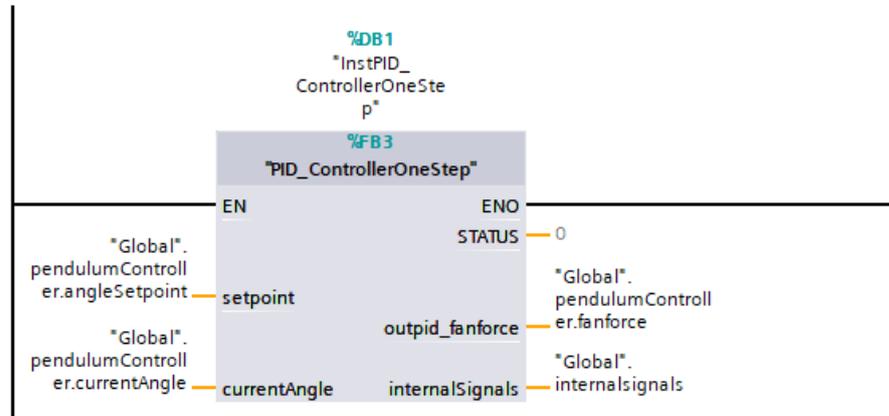
3. Öffnen Sie den DB "OpclInterface" und fügen Sie folgende Variablen für den Datenaustausch mit dem Simulink-Modell hinzu:
  - fanForce [Real]
  - currentAngle [LReal]

OpclInterface			
	Name	Data type	Start value
1	Static		
2	fanForce	Real	0.0
3	currentAngle	LReal	0.0

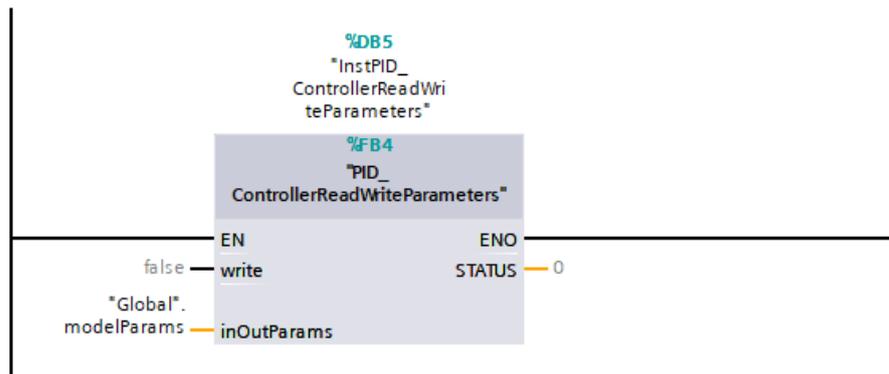
**OB "Cyclic20ms"**

Führen Sie folgende Schritte durch, um den "Cyclic20ms" OB zu programmieren.

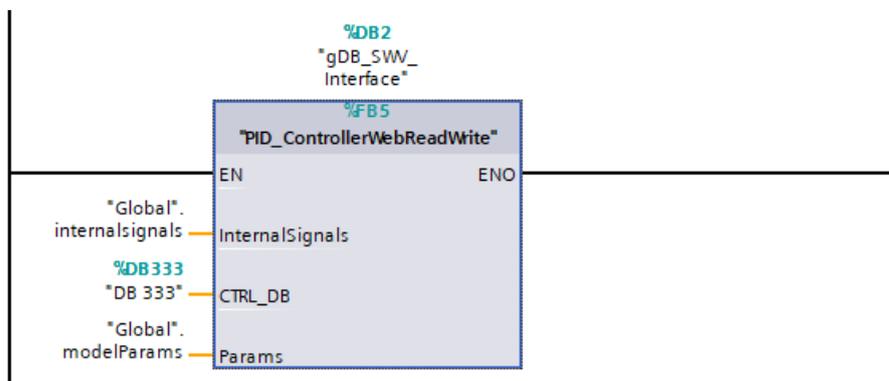
1. Öffnen Sie den Weckalarm OB "Cyclic20ms".
2. Rufen Sie den Baustein "PID\_ControllerOneStep" auf und verschalten Sie ihn wie in folgender Abbildung.



3. Rufen Sie den Baustein "PID\_ControllerReadWriteParameters" auf und verschalten Sie ihn wie in folgender Abbildung.



4. Rufen Sie den Baustein "PID\_ControllerWebReadWrite" auf und verschalten Sie ihn wie in folgender Abbildung. Dabei muss der Instance-Name "gDB\_SWV\_Interface" heißen.



5. Weisen Sie den OPCUA Variablen "fanForce" und "currentAngle" ihre Werte zu.

```

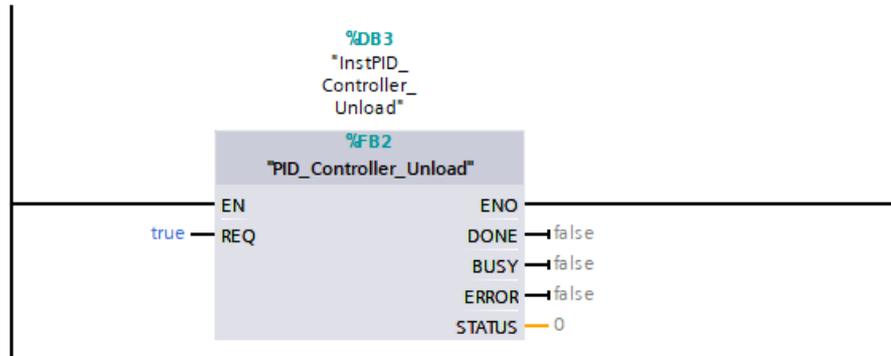
1 "Global".pendulumController.currentAngle := "OpcInterface".currentAngle;
2 "OpcInterface".fanForce := LREAL_TO_REAL("Global".pendulumController.fanforce );

```

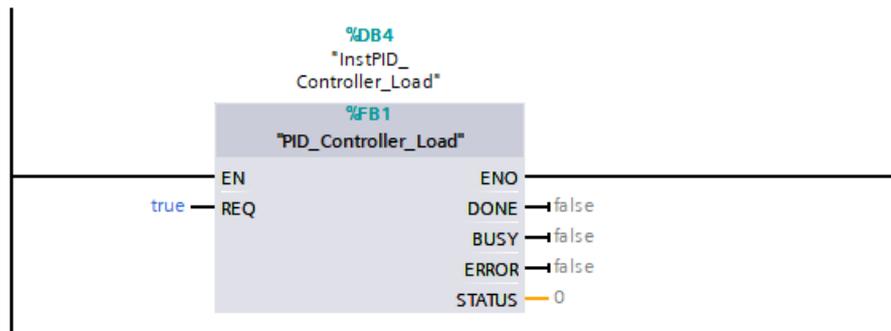
**OB "Startup"**

Führen Sie folgende Schritte durch, um den "Startup" OB zu programmieren.

1. Rufen Sie den Baustein "PID\_Controller\_Unload" auf und verschalten Sie ihn wie in folgender Abbildung.



2. Rufen Sie den Baustein "PID\_Controller\_Load" auf und verschalten Sie ihn wie in folgender Abbildung.

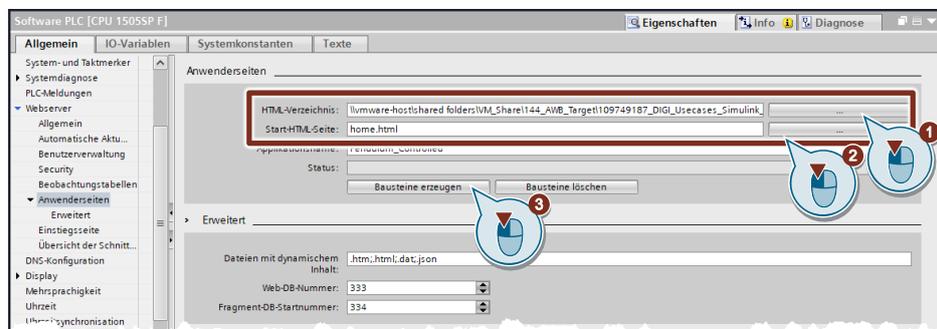


© Siemens AG 2020. All rights reserved

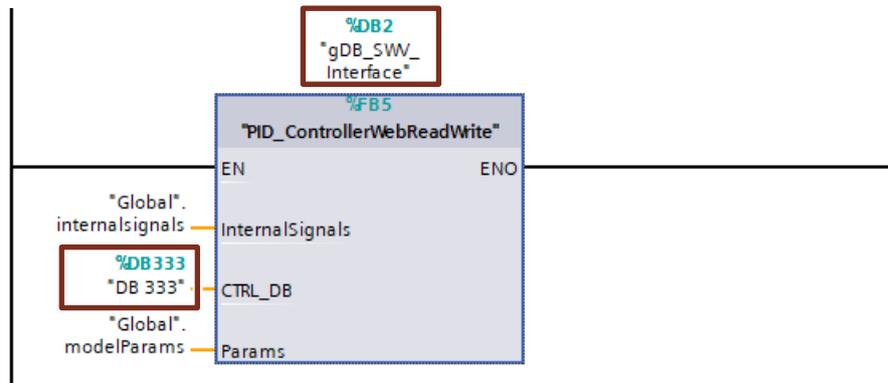
**Webserver Anwenderseiten**

Führen Sie folgende Schritte durch, um die Anwenderseite dem Webserver zuzufügen.

1. Wählen Sie das Verzeichnis der aus dem Matlab-Modell generierten Website aus: "...\Pendulum\_Target\_1500S\_V1\_0\_grt\_Output\website".
2. Hinterlegen Sie "home.html" als Start-HTML-Seite.
3. Erzeugen Sie die Bausteine

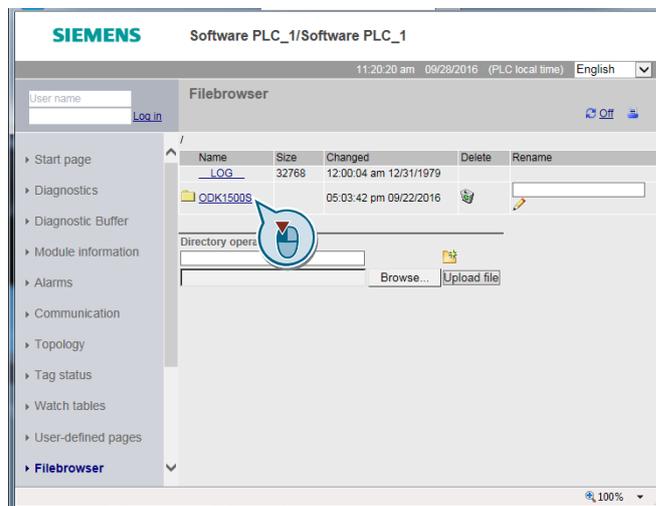


- Prüfen Sie, ob im OB "Cyclic20ms" beim Aufruf des FB "PID\_ControllerWebReadWrite" die gewählte Web-DB-Nummer (hier 333) hinterlegt ist und die Instanz mit "gDB\_SWV\_Interface" benannt ist.

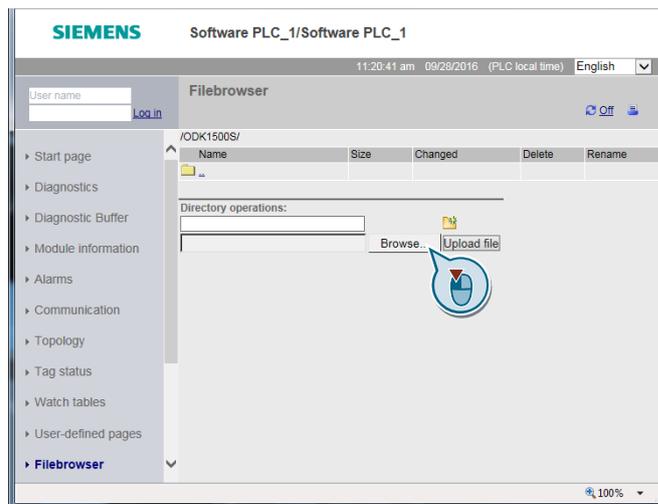


### SO-Datei auf Webserver laden

- Öffnen Sie einen beliebigen Browser (Internet Explorer, Firefox usw.).
- Öffnen Sie die Webseite <http://192.168.2.1:81>. Die Startseite des CPU-Webserver öffnet sich.
- Klicken Sie auf "Weiter" ("Enter").
- Navigieren Sie zu "Filebrowser".
- Klicken Sie auf den Ordner "ODK1500S".



6. Klicken Sie auf die Schaltfläche "Durchsuchen" ("Browse").



7. Navigieren Sie zu der Datei "PID\_Controller\_ODK.so", die von Target 1500S erstellt wurde. Die Datei befindet sich im Ordner "outputs" des generierten Target 1500S Codes.
8. Markieren Sie die Datei und klicken Sie auf "Öffnen" ("Open").
9. Klicken Sie auf die Schaltfläche "Datei laden" ("Upload file").
10. Schließen Sie den Browser.

### Hinweis

Nachdem Sie die SO-Datei in den Webserver geladen haben, müssen Sie die CPU neu starten.

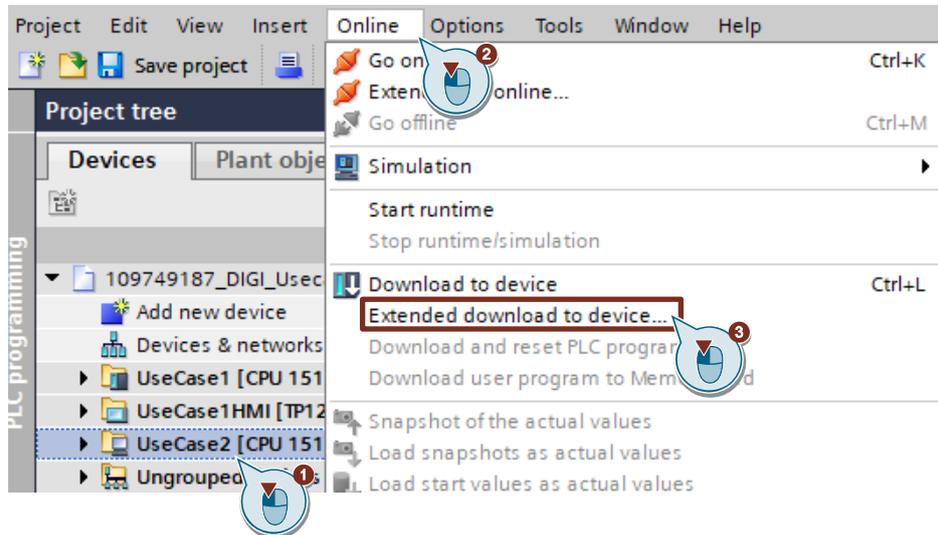
Erst durch den Aufruf des Bausteins "PID\_Controller\_Load" im OB "Startup" wird die SO-Datei in den Arbeitsspeicher der CPU geladen.

## 2.5 Inbetriebnahme der Beispielapplikation

Führen Sie folgende Schritte durch, um die Simulation mit dem mitgelieferten TIA Portal Projekt und dem mitgelieferten Simulink-Modell in Betrieb zu nehmen. In dem Simulink-Modell ist die OPC-Funktion bereits eingefügt.

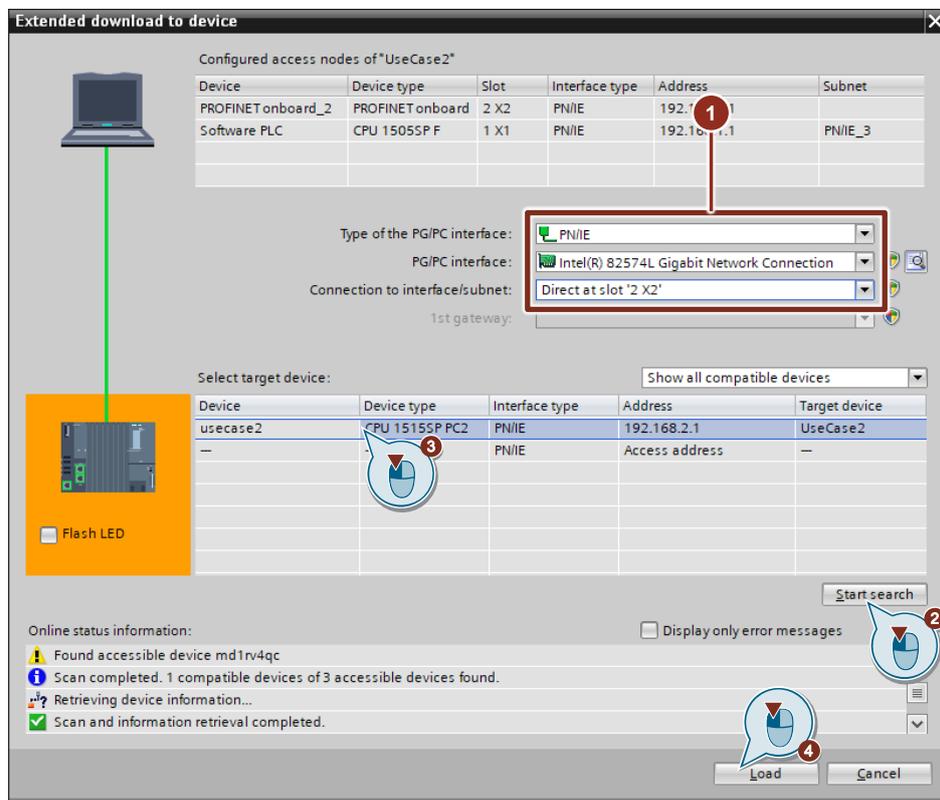
## 2.5.1 Projekt laden

1. Öffnen Sie das Beispielprojekt oder Ihr TIA Portal-Projekt.
2. Markieren Sie in der Projektnavigation die Steuerung "UseCase2" und wählen Sie in der Menüleiste von TIA Portal "Online > Erweitertes Laden in Gerät..." ("Online > Extended download to device") aus.

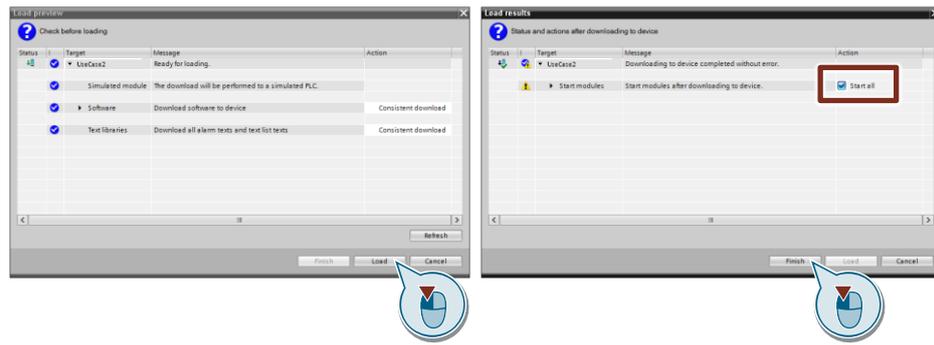


3. Laden:

1. Stellen Sie die Schnittstelle zum Controller ein.
2. Starten Sie die Suche.
3. Wählen Sie aus der Liste den gefundenen virtuellen Controller aus.
4. Laden Sie das Projekt.

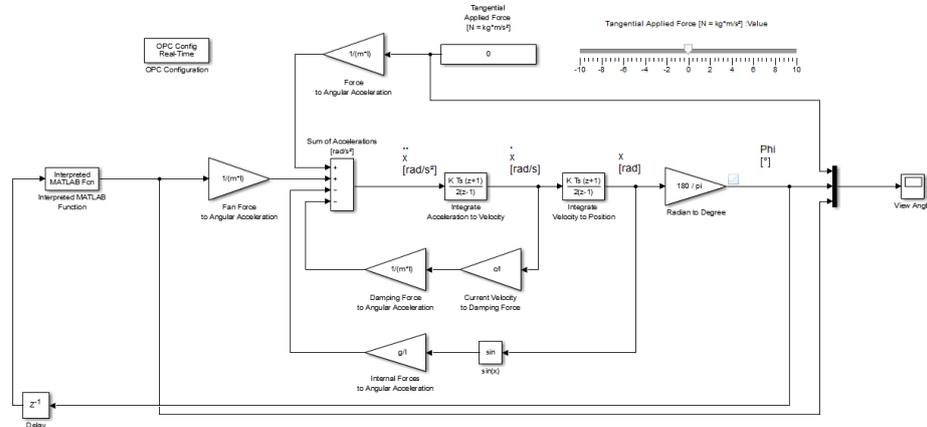


4. Folgen Sie weiteren TIA Portal-Meldungen und starten Sie den Controller.



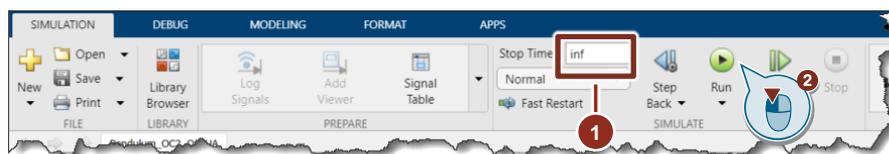
### 2.5.2 Simulink-Simulation starten

1. Öffnen Sie das mitgelieferte Simulink-Modell "Pendulum\_OC2\_OpcUA.slx".



2. Starten Sie die Simulation des Simulink-Modells.

1. Stellen Sie die Simulationsdauer auf "inf" (= unbegrenzt).
2. Klicken Sie auf das Symbol "Run".

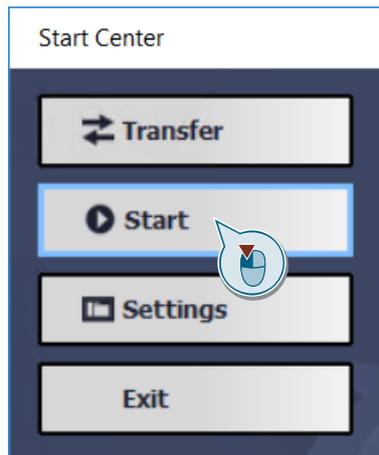


## 3 Bedienung

### 3.1 PID-Regler bedienen

#### HMI starten

1. Verbinden Sie sich über RemoteDesktop auf die Windowsumgebung des Open Controllers.
2. Starten Sie über den WinCC Runtime Loader die HMI-Runtime.



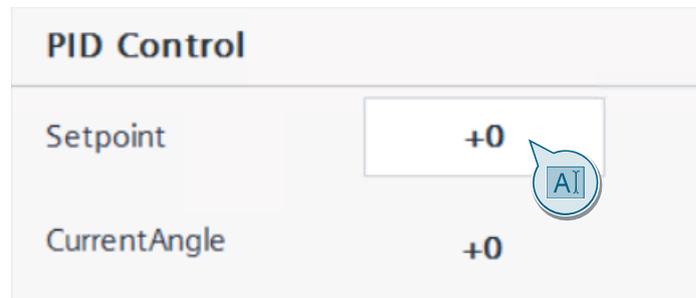
3. Wählen Sie "Start Application" an.



### Sollwert vorgeben

Klicken Sie in das Eingabefeld "Setpoint" und geben Sie einen Sollwert für den Auslenkungswinkel des Pendels ein. Bestätigen Sie Ihre Eingabe mit Enter.

Abbildung 3-1: Sollwert vorgeben



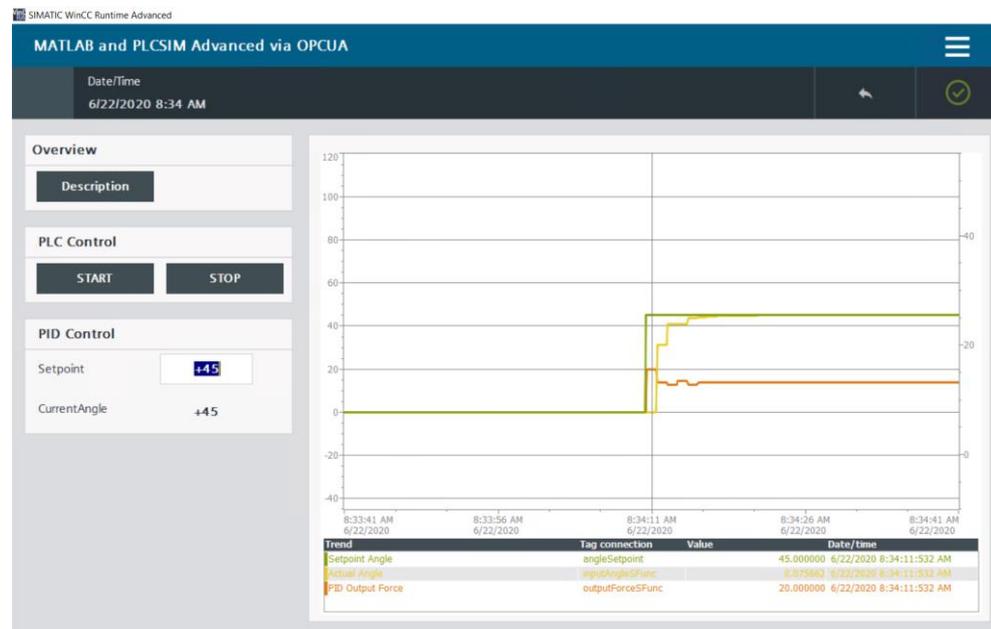
## 3.2 Regelergebnis beobachten

Zum Beobachten des Regelergebnisses stehen Ihnen zwei Möglichkeiten zur Verfügung.

Zum einen stellt der Trace in der Simulation des HMIs den Verlauf des Sollwertes, des Istwertes (Auslenkungswinkel) und der Stellgröße (Kraft) des PID-Reglers graphisch dar.

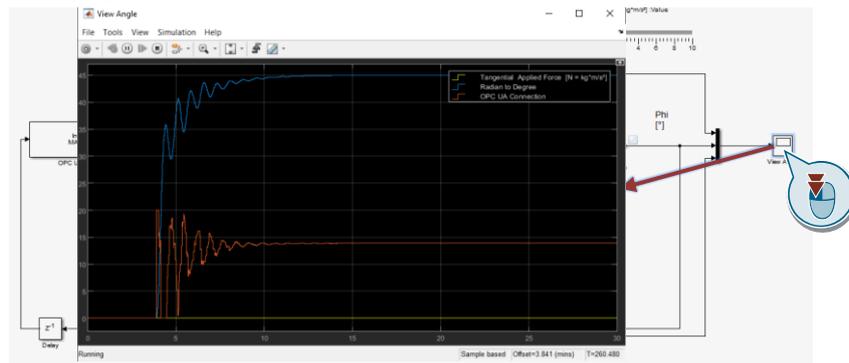
Die Abbildung zeigt einen Verlauf für einen eingestellten Sollwinkel von 45°.

Abbildung 3-2: Simulation des HMIs



Zum anderen können Sie das Regelergebnis auch über ein Anzeigefenster in Simulink beobachten. Durch einen Doppelklick auf die "Scope"-Funktion im Simulink-Modell öffnen Sie das Anzeigefenster.

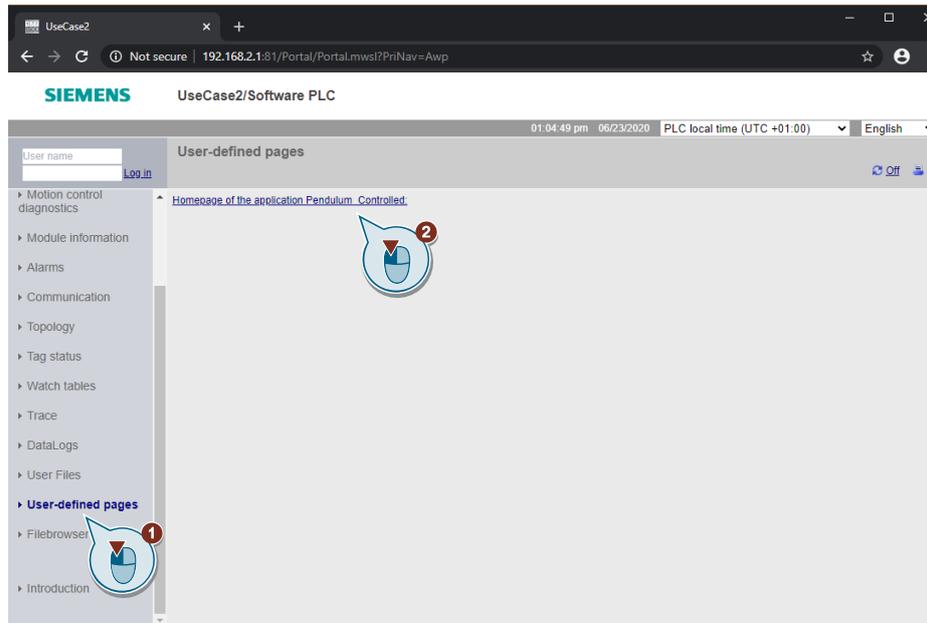
Abbildung 3-3: "Scope"-Funktion



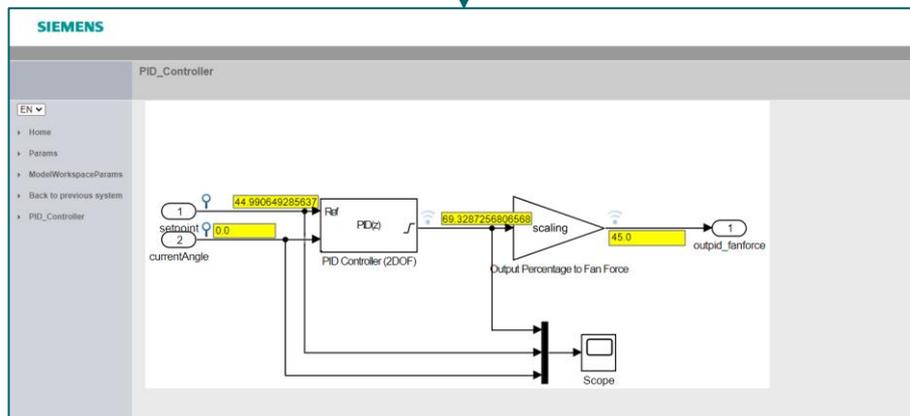
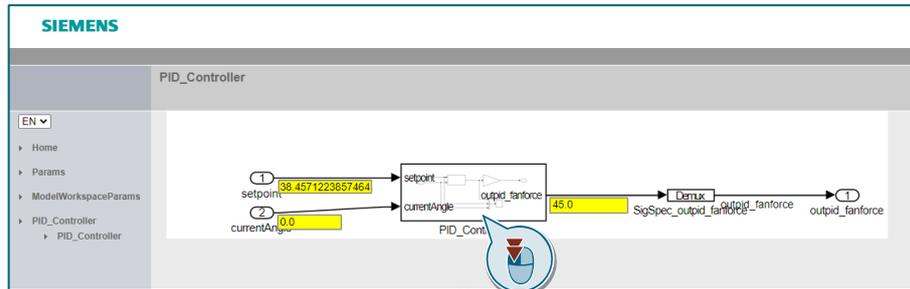
## 3.3 Anwender-Website nutzen

Die Anwenderseite des Webbrowsers stellt eine weitere Möglichkeit zur Bedienung und Beobachtung des PID-Reglers dar.

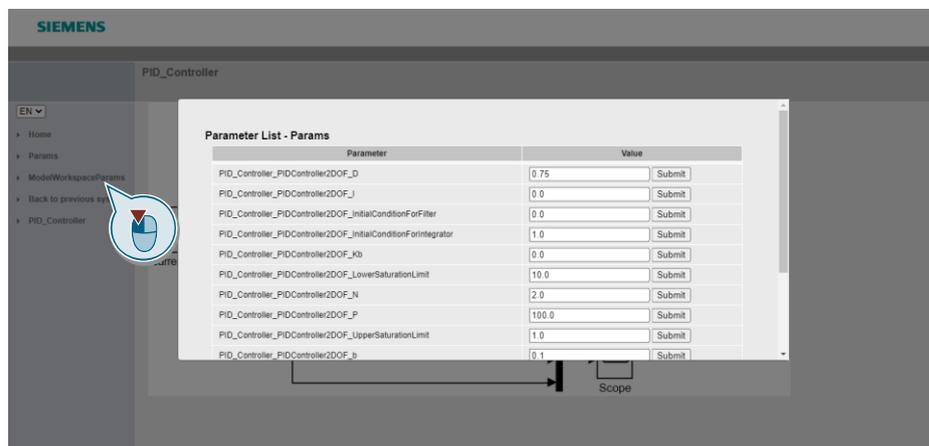
1. Öffnen Sie die Anwenderseite im Webserver.



- In der Anwenderseite können die Werte am Eingang / Ausgang und innerhalb des PID\_Controller eingesehen. Für die Werte innerhalb klicken Sie auf das Subsystem.



- Die Parameter des PID\_Controller können unter ModelWorkspaceParams angepasst werden.



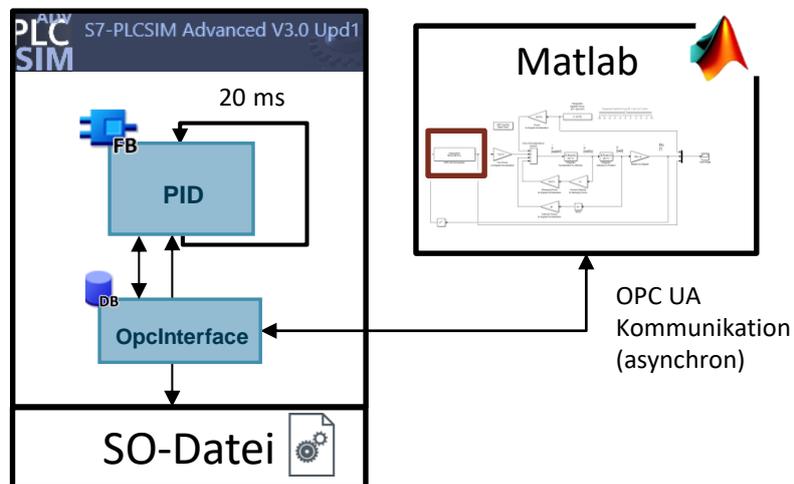
## 4 Alternative Lösung

Alternativ können Sie Ihr Programm und die Hardwarekonfiguration in eine virtuelle CPU laden und so die Funktion des PID-Reglers im Zusammenspiel mit dem Simulationsmodell testen.

Laden Sie hierfür die im TIA-Projekt bereits vorbereitete CPU "UseCase2PLCSimAdv" in eine virtuelle CPU auf PLCSIM Advanced. Legen Sie die SO-Datei "PID\_:Controller\_ODK.so" über den Webserver oder über die "Virtual SIMATIC Memory Card" auf der virtuellen CPU ab. Starten Sie anschließend sowohl die Simulation des HMIs "UseCase2HMIPLCSimAdv" als auch das Simulationsmodell "Pendulum\_OC2\_OpcUA.slx" und testen Sie das Zusammenspiel.

**Hinweis** Sie müssen die PG/PC-Schnittstelle in der Systemsteuerung auf den Siemens PLCSIM Virtual Ethernet Adapter einstellen.

Abbildung 4-1: Überblick der alternativen Lösung



## 5 Code MATLAB-Funktion

```

function [x] = Read OPC_Func(y)
% variables
persistent init_Server;
persistent init_Nodes;
persistent uaClient;
persistent Var_Node_In;
persistent Var_Node_Out;
persistent testVal;
% initialize variables
if (isempty(init_Server))
    testVal = 0;
    init_Server = 0;
    init_Nodes = 0;
end
% OPC UA server (PLC) address and connecting client (Simulink) to
the server
if init_Server == 0
    init_Server = 1;
    uaClient = opcua('192.168.1.1',4840);
    connect(uaClient);
end
% define variable nodes in the server
if uaClient.isConnected == 1 && init_Nodes == 0
    init_Nodes = 1;
    % read out the variables of the OPC UA server
    Var_Node_In =
opcuanode(3,"""OpcInterface"".""fanForce""",uaClient);
    Var_Node_Out=
opcuanode(3,"""OpcInterface"".""currentAngle""",uaClient);
end
% read and write variables of the server
if uaClient.isConnected == 1 && init_Nodes == 1
    % read "fanForce" value from server and store in "val"
    [val, ~, ~] = readValue(uaClient, Var_Node_In);
    % assign input y of the function to "currentAngle" variable
    writeValue(uaClient, Var_Node_Out, y);
    % assign "val" to variable "testVal"
    testVal = val;
end
% assign "fanForce" ("testVal") value to the output x of the
function
x = double(testVal);
end

```

## 6 Anhang

### 6.1 Service und Support

#### Industry Online Support

Sie haben Fragen oder brauchen Unterstützung?

Über den Industry Online Support greifen Sie rund um die Uhr auf das gesamte Service und Support Know-how sowie auf unsere Dienstleistungen zu.

Der Industry Online Support ist die zentrale Adresse für Informationen zu unseren Produkten, Lösungen und Services.

Produktinformationen, Handbücher, Downloads, FAQs und Anwendungsbeispiele – alle Informationen sind mit wenigen Mausklicks erreichbar:

[support.industry.siemens.com](https://support.industry.siemens.com)

#### Technical Support

Der Technical Support von Siemens Industry unterstützt Sie schnell und kompetent bei allen technischen Anfragen mit einer Vielzahl maßgeschneiderter Angebote – von der Basisunterstützung bis hin zu individuellen Supportverträgen.

Anfragen an den Technical Support stellen Sie per Web-Formular:

[support.industry.siemens.com/cs/my/src](https://support.industry.siemens.com/cs/my/src)

#### SITRAIN – Digital Industry Academy

Mit unseren weltweit verfügbaren Trainings für unsere Produkte und Lösungen unterstützen wir Sie praxisnah, mit innovativen Lernmethoden und mit einem kundenspezifisch abgestimmten Konzept.

Mehr zu den angebotenen Trainings und Kursen sowie deren Standorte und Termine erfahren Sie unter:

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

#### Serviceangebot

Unser Serviceangebot umfasst folgendes:

- Plant Data Services
- Ersatzteilservices
- Reparaturservices
- Vor-Ort und Instandhaltungsservices
- Retrofit- und Modernisierungsservices
- Serviceprogramme und Verträge

Ausführliche Informationen zu unserem Serviceangebot finden Sie im Servicekatalog:

[support.industry.siemens.com/cs/sc](https://support.industry.siemens.com/cs/sc)

#### Industry Online Support App

Mit der App "Siemens Industry Online Support" erhalten Sie auch unterwegs die optimale Unterstützung. Die App ist für iOS und Android verfügbar:

[support.industry.siemens.com/cs/ww/de/sc/2067](https://support.industry.siemens.com/cs/ww/de/sc/2067)

## 6.2 Links und Literatur

Tabelle 6-1

Nr.	Thema
\1\	Siemens Industry Online Support <a href="https://support.industry.siemens.com">https://support.industry.siemens.com</a>
\2\	Link auf die Beitragsseite des Anwendungsbeispiels <a href="https://support.industry.siemens.com/cs/ww/de/view/109749187">https://support.industry.siemens.com/cs/ww/de/view/109749187</a>
\3\	Handbuch: SIMATIC S7-PLCSIM Advanced <a href="https://support.industry.siemens.com/cs/ww/de/view/109773484">https://support.industry.siemens.com/cs/ww/de/view/109773484</a>
\4\	MathWorks Online Dokumentation: <a href="http://de.mathworks.com/help/">http://de.mathworks.com/help/</a>
\5\	Handbuch: STEP 7 Professional V16 <a href="https://support.industry.siemens.com/cs/ww/de/view/109773506">https://support.industry.siemens.com/cs/ww/de/view/109773506</a>

## 6.3 Änderungsdokumentation

Tabelle 6-2

Version	Datum	Änderung
V1.0	12/2017	Erste Ausgabe
V2.0	09/2020	Überarbeitung Anwendungsfälle 1&2