



WITTENSTEIN

alpha

cynapse<sup>®</sup>  
SIEMENS Steuerung – ifm IO-Link Master

Implementierungsbeispiel



## WITTENSTEIN alpha GmbH

Walter-Wittenstein-Straße 1  
D-97999 Igersheim  
Germany

## Cybertronic Support

Bei Fragen zu diesem Implementierungsbeispiel wenden Sie sich bitte direkt an:  
[cybertronic-support@wittenstein.de](mailto:cybertronic-support@wittenstein.de)

## Customer Service

		✉	☎
Deutschland	WITTENSTEIN alpha GmbH	service@wittenstein-alpha.de	+49 7931 493-12900
Benelux	WITTENSTEIN BVBA	service@wittenstein.biz	+32 9 326 73 80
Brasil	WITTENSTEIN do Brasil	vendas@wittenstein.com.br	+55 15 3411 6454
中国	威騰斯坦（杭州）实业有限公司	service@wittenstein.cn	+86 571 8869 5856
Österreich	WITTENSTEIN GmbH	office@wittenstein.at	+43 2256 65632-0
Danmark	WITTENSTEIN AB	info@wittenstein.dk	+45 4027 4151
France	WITTENSTEIN sarl	info@wittenstein.fr	+33 134 17 90 95
Great Britain	WITTENSTEIN Ltd.	sales.uk@wittenstein.co.uk	+44 1782 286 427
Italia	WITTENSTEIN S.P.A.	info@wittenstein.it	+39 02 241357-1
日本	ヴィッテンシュタイン株式会社	sales@wittenstein.jp	+81-3-6680-2835
North America	WITTENSTEIN holding Corp.	technicalsupport@wittenstein-us.com	+1 630-540-5300
España	WITTENSTEIN S.L.U.	info@wittenstein.es	+34 93 479 1305
Sverige	WITTENSTEIN AB	info@wittenstein.se	+46 40-26 50 10
Schweiz	WITTENSTEIN AG Schweiz	sales@wittenstein.ch	+41 81 300 10 30
台湾	威騰斯坦有限公司	info@wittenstein.tw	+886 3 287 0191
Türkiye	WITTENSTEIN Güç Aktarma Sistemleri Tic. Ltd. Şti.	info@wittenstein.com.tr	+90 216 709 21 23

© WITTENSTEIN alpha GmbH 2023

Inhaltliche und technische Änderungen vorbehalten.

# Inhaltsverzeichnis

<b>1</b>	<b>Zu dieser Anleitung</b> .....	<b>2</b>
1.1	Informationssymbole und Querverweise .....	2
<b>2</b>	<b>Hardwareaufbau</b> .....	<b>3</b>
<b>3</b>	<b>Inbetriebnahme im SIEMENS TIA Portal V15.1</b> .....	<b>4</b>
3.1	Hardwarekonfiguration der Komponenten.....	4
3.2	Integration von cynapse® .....	10
<b>4</b>	<b>Prozessdaten</b> .....	<b>12</b>
4.1	Definition.....	12
4.2	Prozessdaten mit Hilfe des „cynapse Prozessdaten“ FB lesen .....	12
<b>5</b>	<b>Parameter</b> .....	<b>16</b>
5.1	Definition.....	16
5.2	Siemens Baustein für Parameter lesen/schreiben in Programm einbinden .....	16
5.3	Parameter lesen .....	22
5.4	Parameter schreiben.....	25
<b>6</b>	<b>Events</b> .....	<b>28</b>
6.1	Definition.....	28
6.2	Events auslesen .....	28
<b>7</b>	<b>Blob-Daten</b> .....	<b>34</b>
7.1	Definition.....	34
7.2	Blob-Daten mit Hilfe des „Blob_Transfer“ FB lesen.....	34

# 1 Zu dieser Anleitung

Diese Anleitung enthält Vorgehensweisen zur beispielhaften Verwendung des WITTENSTEIN Sensors cynapse®.

In dieser Anleitung wird mit Beispielcode gearbeitet. Falls Sie entsprechende Codebeispiele benötigen, wenden Sie sich bitte an:  
cybertronic-support@wittenstein.de

Das Original dieser Anleitung wurde in Deutsch erstellt, alle anderen Sprachversionen sind Übersetzungen dieser Anleitung.

## 1.1 Informationssymbole und Querverweise

Folgende Informationssymbole werden verwendet:

- fordert Sie zum Handeln auf
- ➔ zeigt die Folge einer Handlung an
- ❗ gibt Ihnen zusätzliche Informationen zur Handlung

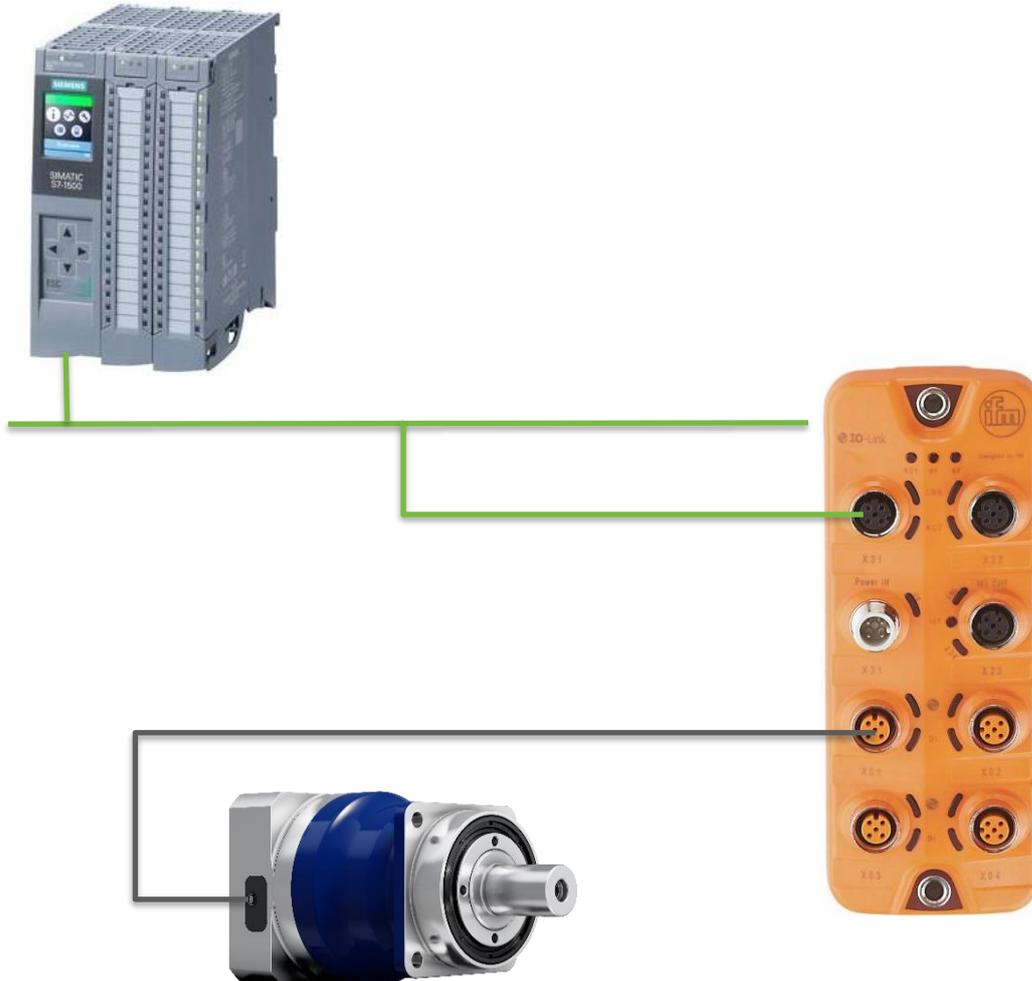
Ein Querverweis bezieht sich auf die Kapitelnummer und die Überschrift des Zielabschnittes (z. B. Kapitel 5 „Parameter“).

Ein Querverweis auf eine Tabelle bezieht sich auf die Tabellenummer (z. B. Tabelle „Tbl - 1“).

## 2 Hardwareaufbau

Der Hardwareaufbau des Beispielprojekts besteht aus folgenden Komponenten:

- Steuerung: SIEMENS S7-1500 (6ES7511-1AK02-0AB0)
- IO-Link Master: ifm AL1300
- IO-Link Device: WITTENSTEIN cynapse®



Die Siemens Steuerung ist mittels PROFINET mit dem ifm IO-Link Master verbunden (grün). cynapse® wird an einen der IO-Link Ports des Masters angeschlossen (schwarz).

Kenntnisse zur korrekten Verdrahtung aller Komponenten werden vorausgesetzt und in dieser Beispielbeschreibung nicht behandelt.

### 3 Inbetriebnahme im SIEMENS TIA Portal V15.1

#### Voraussetzung

Für die Durchführung der Inbetriebnahme von cynapse® benötigen Sie ein offenes Projekt im TIA Portal.

- Der Aufbau der Hardware ist erfolgt.
- Für die vorhandene CPU wurde bereits eine IP-Adresse und die Subnetzmaske vergeben.
- Die GSD Datei des IO-Link Masters wurde über die Website des Master-Hersteller bezogen und liegt vor.

Falls Sie entsprechende Codebeispiele benötigen, wenden Sie sich bitte an [cybertronic-support@wittenstein.de](mailto:cybertronic-support@wittenstein.de)

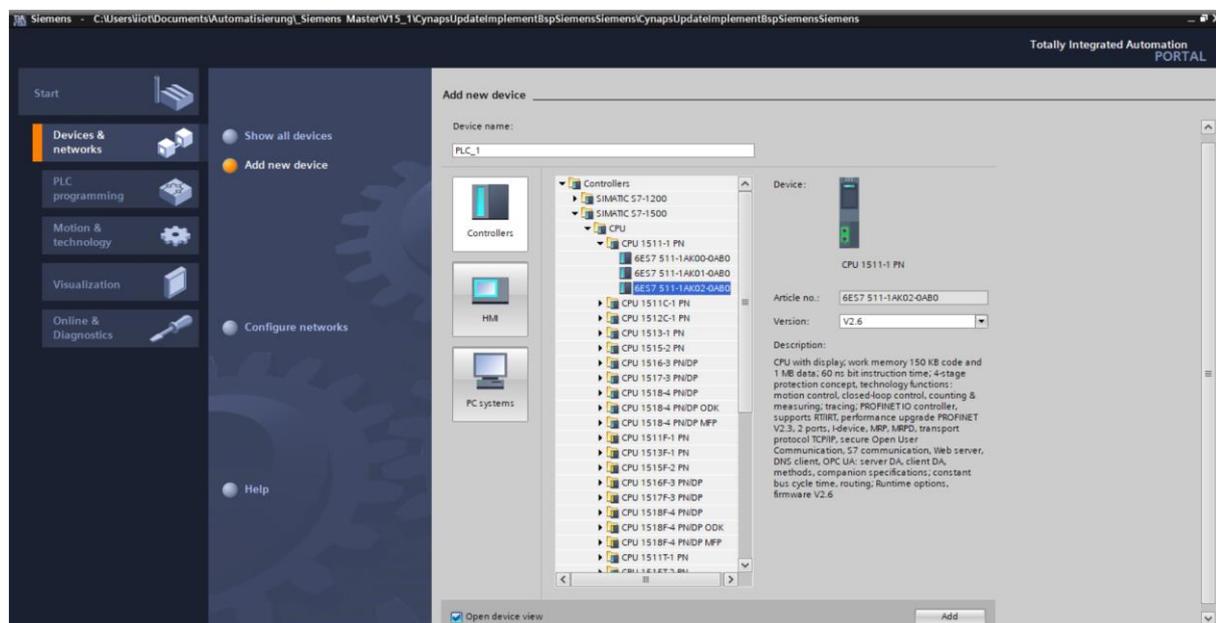
### 3.1 Hardwarekonfiguration der Komponenten

#### Einleitung

Im Folgenden legen Sie die CPU und den IO-Link Master in der Hardware-Konfiguration an und vernetzen diese miteinander.

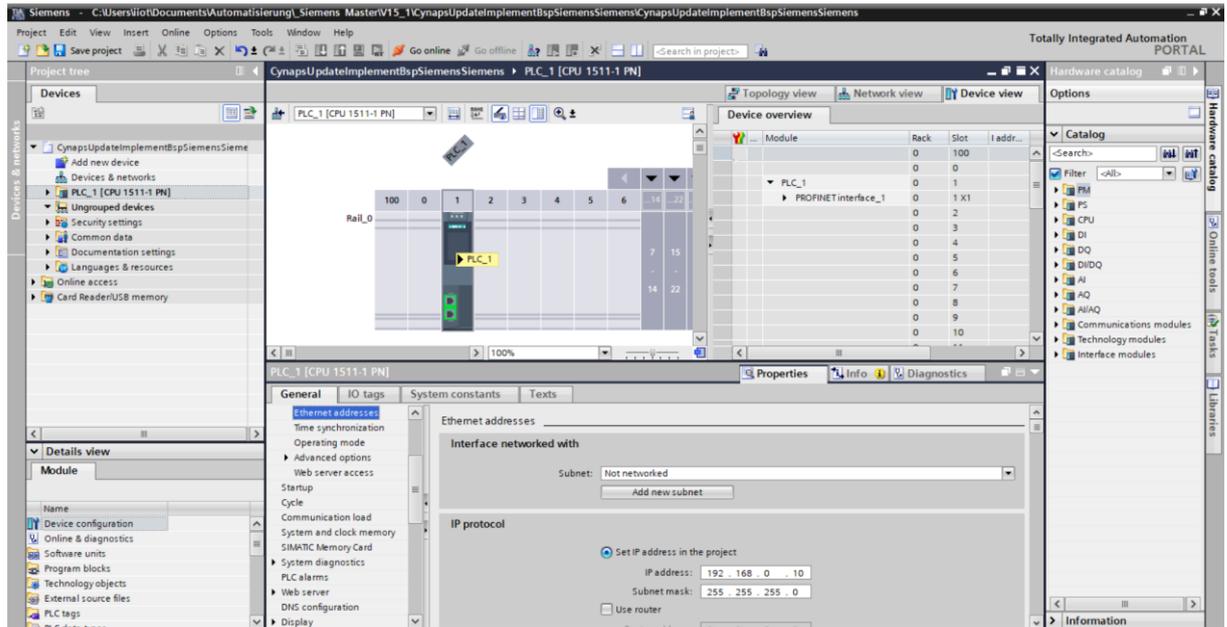
#### Vorgehen

1. Öffnen Sie das Portal „Devices & networks“.
2. Fügen Sie ein neues Gerät ein.
3. Öffnen Sie den Ordner „SIMATIC S7-1500“.
4. Wählen Sie die verwendete CPU aus.
5. Passen sie ggf. die Version Ihrer Hardware an.

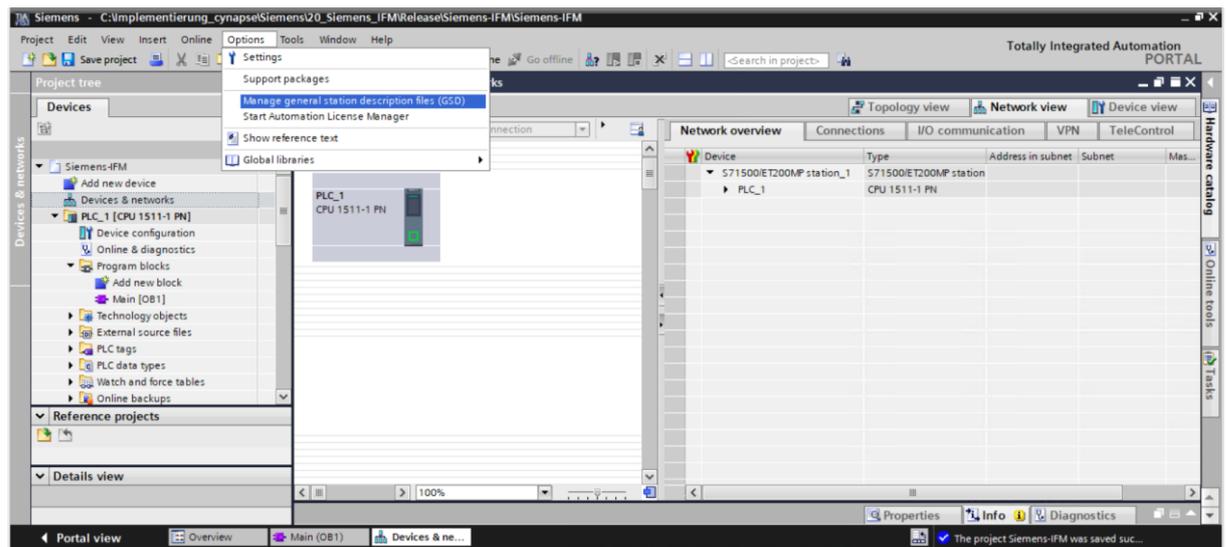


6. Legen Sie die CPU über Doppelklick auf den Namen an.
7. Springen Sie über einen Doppelklick auf die CPU in der automatisch geöffneten Projektansicht in die Einstellungen der CPU.

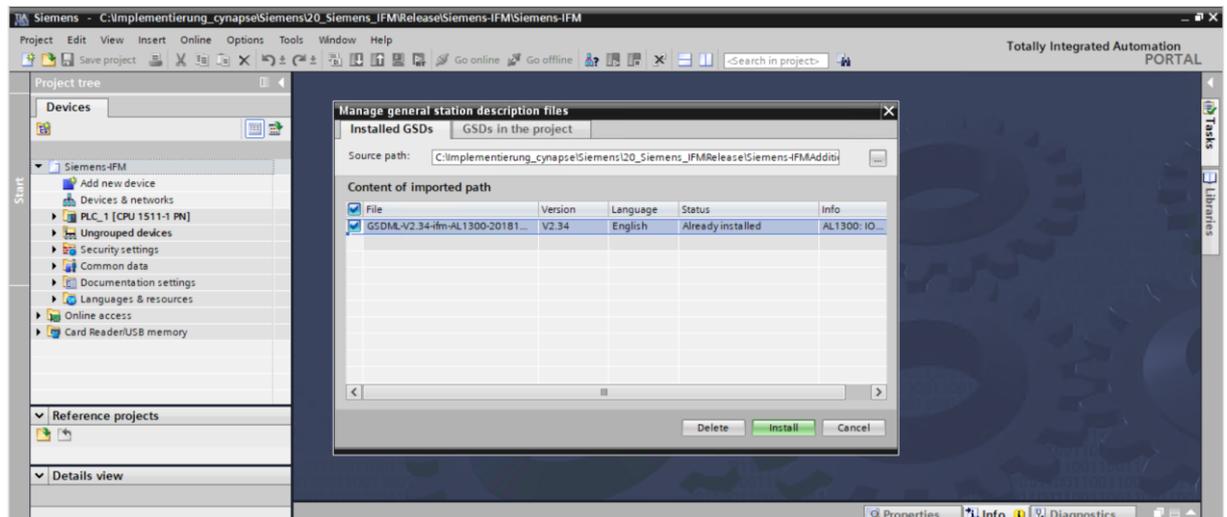
8. Tragen Sie unter „Ethernet addresses“ die vergebene IP-Adresse und die Subnetzmaske ein.



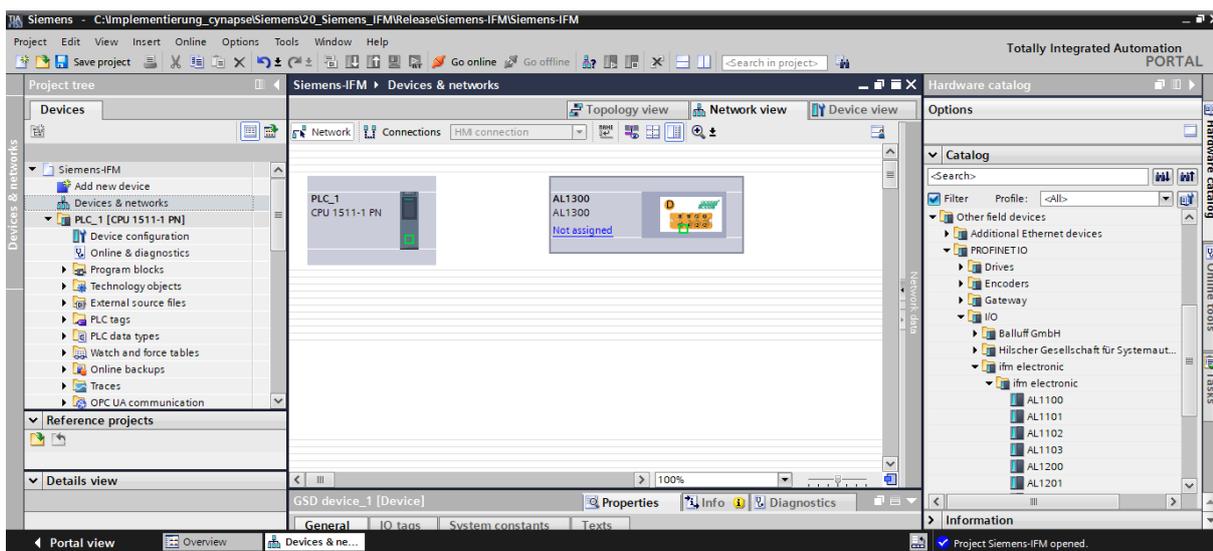
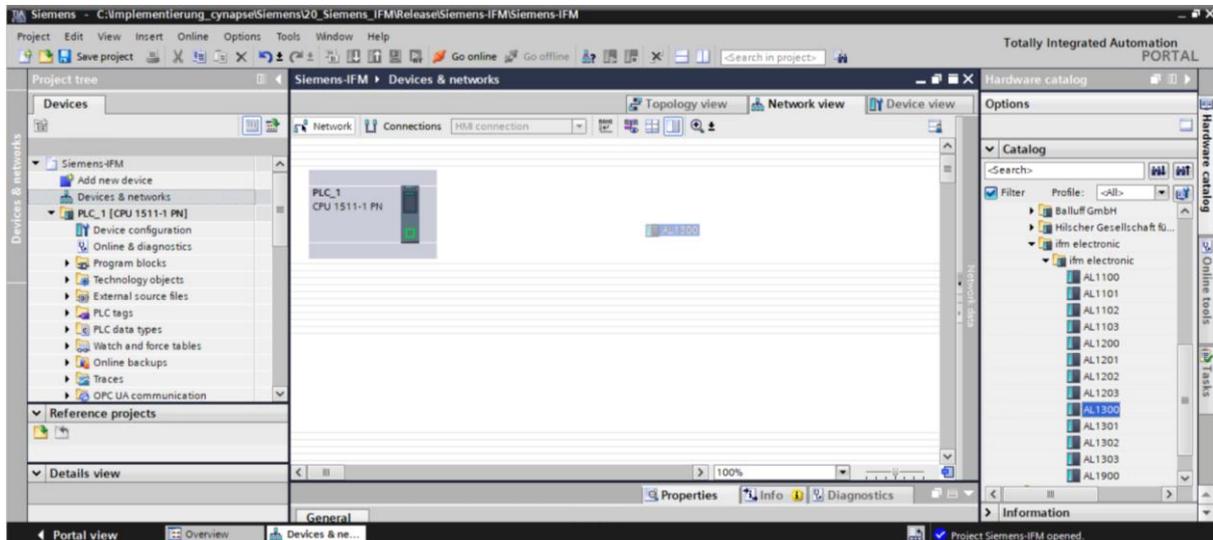
9. Importieren Sie über „Options“ die GSD-Datei des IO-Link Masters.



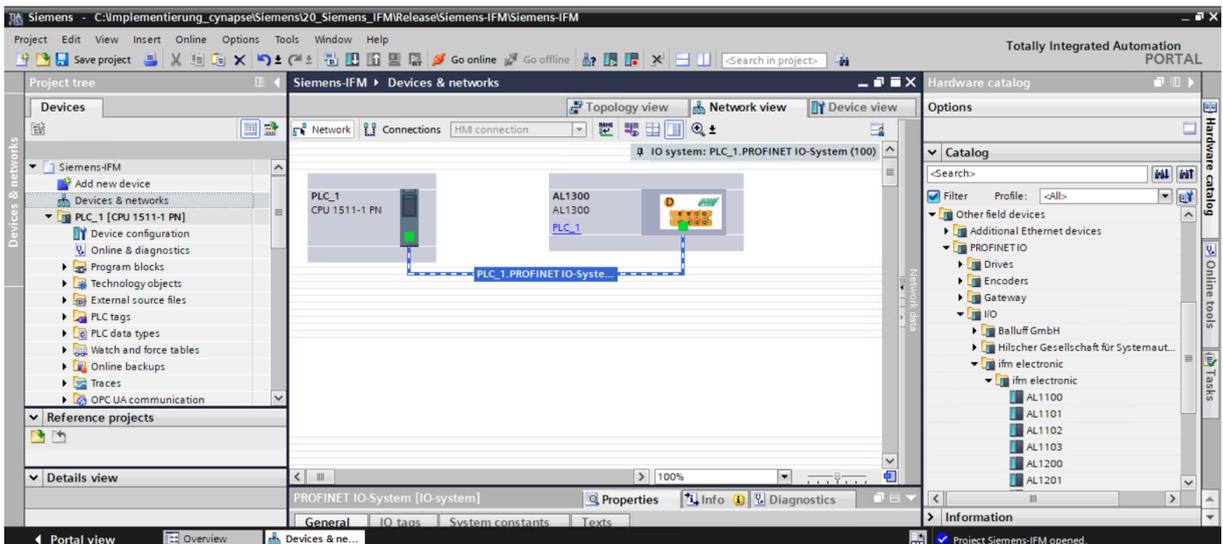
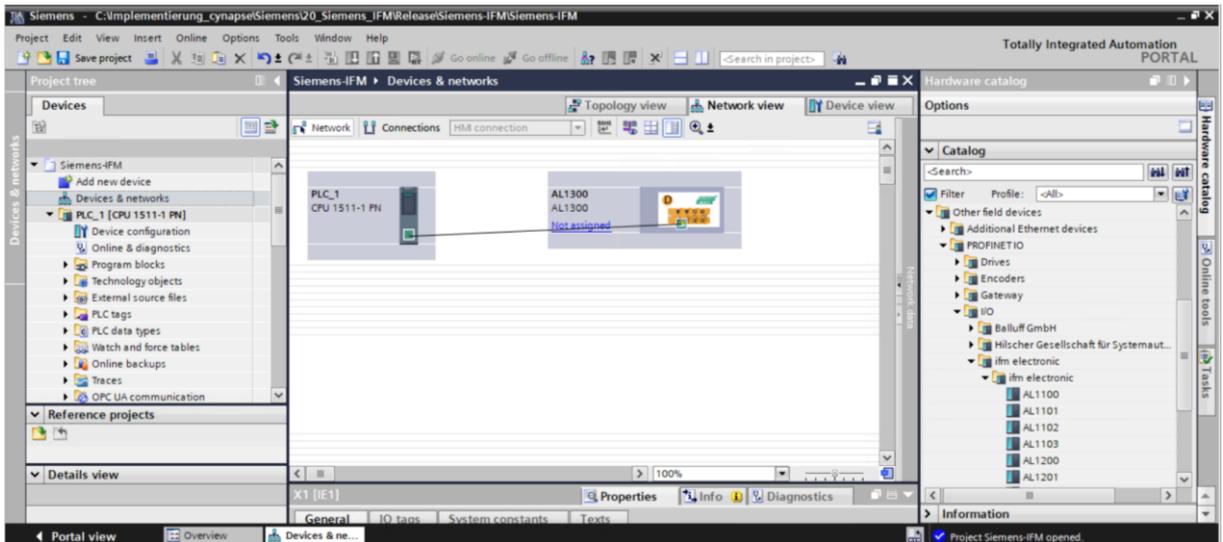
10. Öffnen Sie den GSD-Manager und installieren Sie die GSD-Datei.



11. Öffnen Sie den „Hardware catalog“.
12. Wechseln Sie in die „Network view“.
13. Öffnen Sie den Ordner „Other Field Devices“ und den Ordner „Profinet IO“.
14. Öffnen Sie die Ordner „I/O“, „ifm Electronic“ und den Ordner „ifm Electronic“.
15. Ziehen Sie das verwendete Interfacemodul per Drag & Drop in den weißen Hintergrund der Netzsicht.

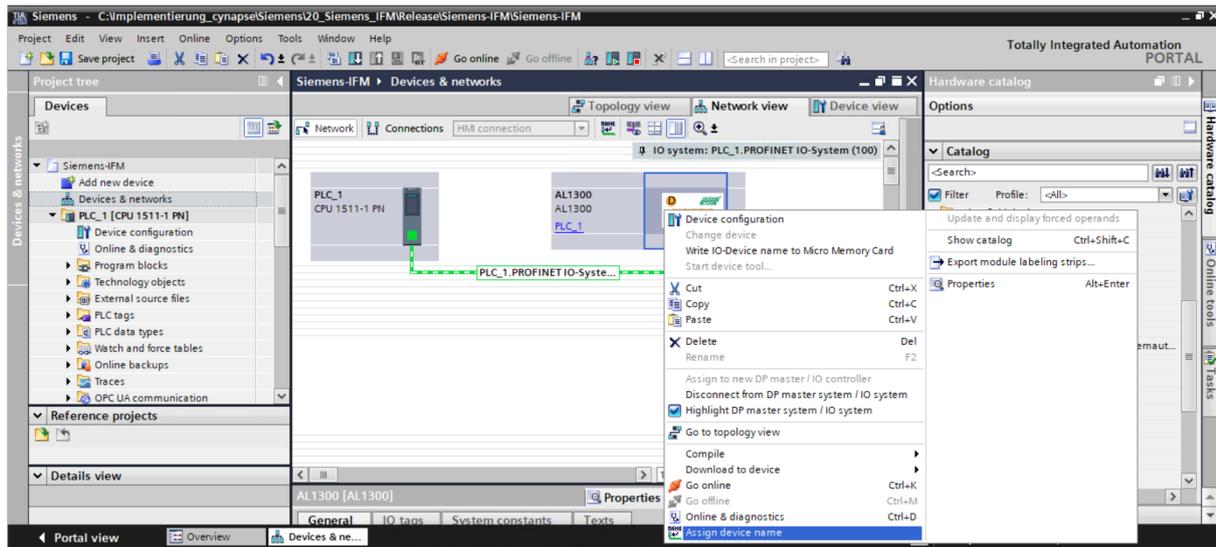


16. Ziehen Sie per Drag & Drop eine Verbindung von der Schnittstelle der CPU zur Schnittstelle des Interfacemoduls, um diese per PROFINET zu vernetzen.

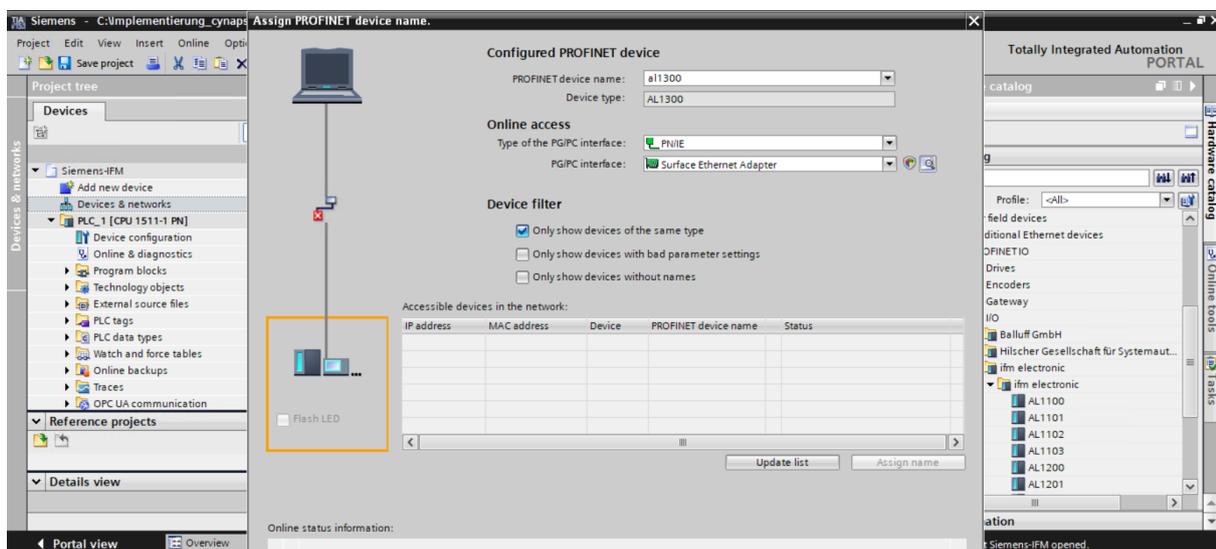


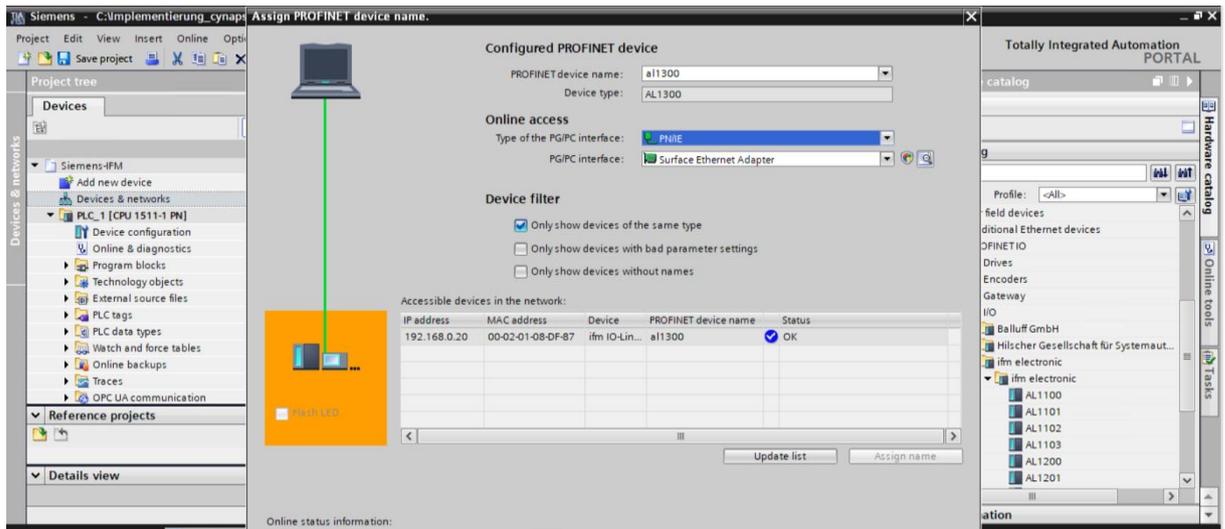
17. Für eine erfolgreiche Kommunikation muss der Gerätenamen des erreichbaren Teilnehmers mit dem der Hardwarekonfiguration übereinstimmen. Überprüfen Sie diesen wie folgt:

- Rechtsklick auf das IO Device.
- Anwahl Gerätenamen zuweisen über „Assign device name“.
- ① Der Name wird von der Entwicklungsumgebung vorgeschlagen. Dieser kann aber auch umbenannt werden.



- Klick auf „Update list“.





## Ergebnis

- ➔ Die Hardwarekomponenten sind alle verbunden und eine erfolgreiche Kommunikation zwischen TIA Portal, CPU und IO-Link Master ist möglich.

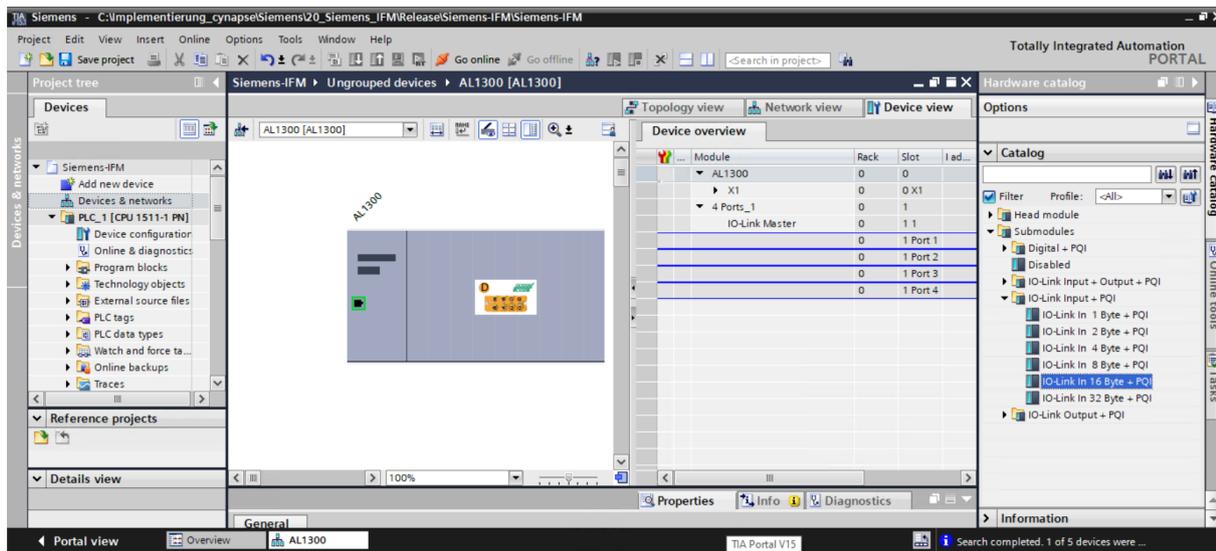
## 3.2 Integration von cynapse®

### Einleitung

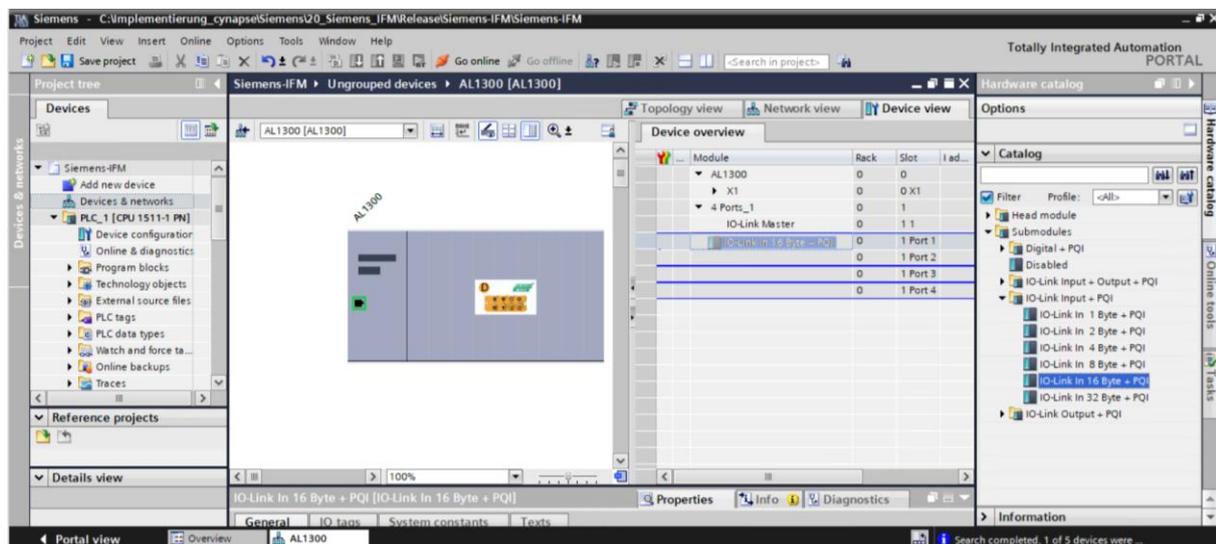
Im Folgenden vervollständigen Sie die Hardwarekonfiguration mit der Integration von cynapse®.

### Vorgehen

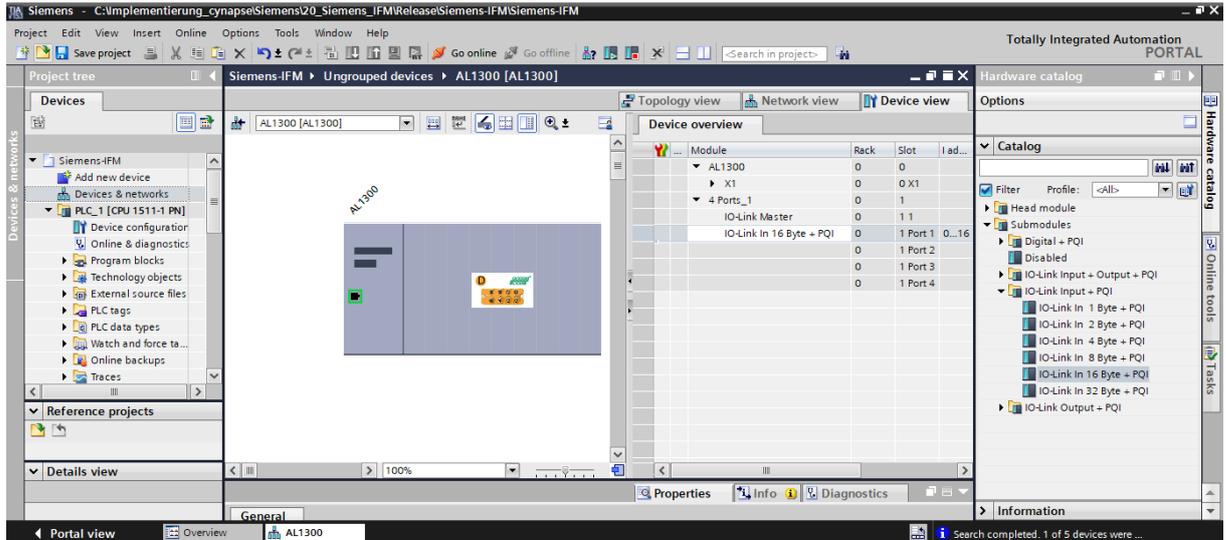
1. Öffnen Sie den Reiter „Device View“ und den Hardwarekatalog.



2. Ziehen Sie per Drag and Drop „IO-Link IN 16 Byte+PQI“, gemäß der Betriebsanleitung, an den Port, an dem cynapse® angeschlossen ist.

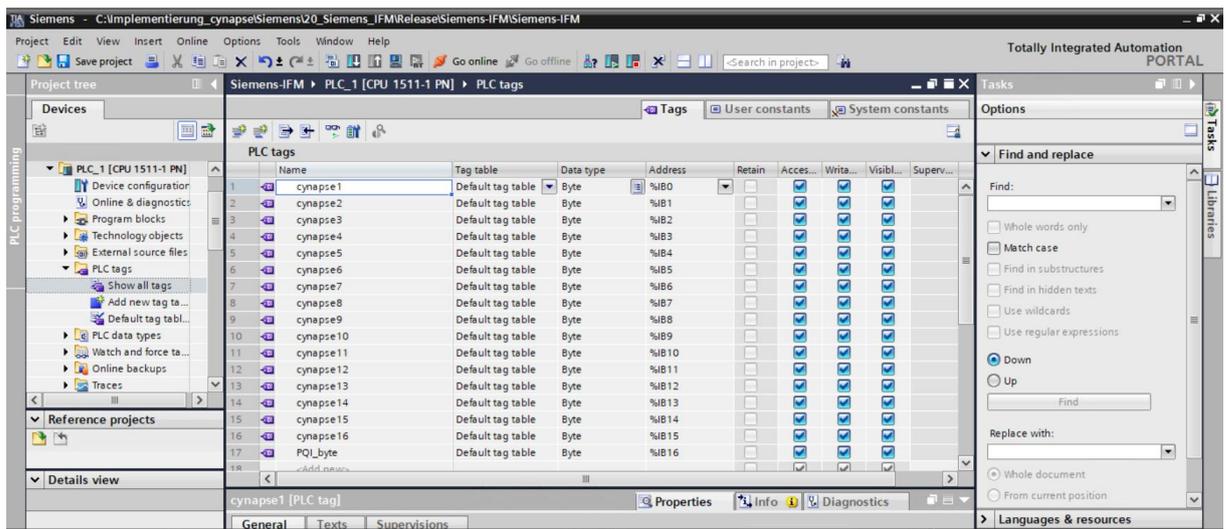


3. Als Ergebnis können nun die Daten vom Sensor übertragen werden.



4. Um cynapse® Daten im TIA Portal verwenden zu können muss eine Variablen-tabelle angelegt werden.

5. Gehen Sie im Projektbaum auf PLC tags und legen Sie folgende Variablen an.



**Ergebnis**

- ➔ Die Hardwarekonfiguration ist mit den korrekten Datentypen verknüpft und eine Variablen-tabelle wurde angelegt. Nun ist die Integration von cynapse® abgeschlossen und die Verbindung zu cynapse® ist hergestellt.

## 4 Prozessdaten

### 4.1 Definition

Unter Prozessdaten versteht man zyklisch kommunizierte Daten zwischen IO-Link Master und Steuerung. In jedem Zyklus werden diese Daten übergeben. Die von cynapse® gesendeten Prozessdaten sind abhängig vom Versionsstand der Hard- und Software. Nähere Informationen dazu finden Sie in der Betriebsanleitung cynapse®.

### 4.2 Prozessdaten mit Hilfe des „cynapse Prozessdaten“ FB lesen

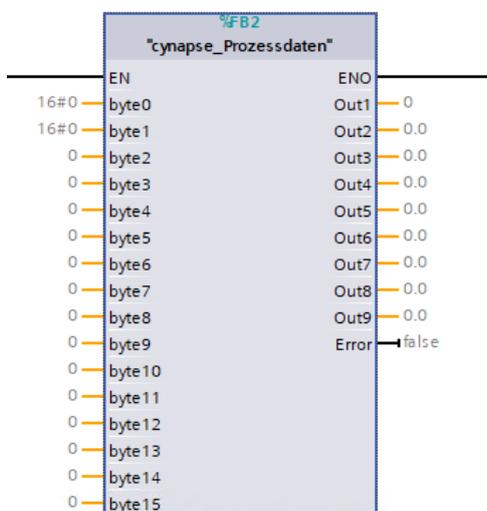
#### Voraussetzung

- ☞ Sie haben ein Beispielprojekt für das Auslesen der Prozessdaten über folgende Quelle bezogen:  
cybertronic-support@wittenstein.de

#### Einleitung

Im Folgenden lesen Sie Prozessdaten mit Hilfe eines Funktionsbausteins aus einem Beispielprojekt aus. Dieser übernimmt die Skalierung der Messwerte.

cynapse® bietet verschiedene Prozessdatenformate an, um bei gleichbleibender Prozessdatenlänge verschiedene Daten zur weiteren Verarbeitung anzubieten. Diese Prozessdaten können durch den Parameter Settings ausgewählt werden. Nähere Informationen hierzu finden Sie in der Betriebsanleitung cynapse® und im Kapitel 5.4 „Parameter schreiben“.

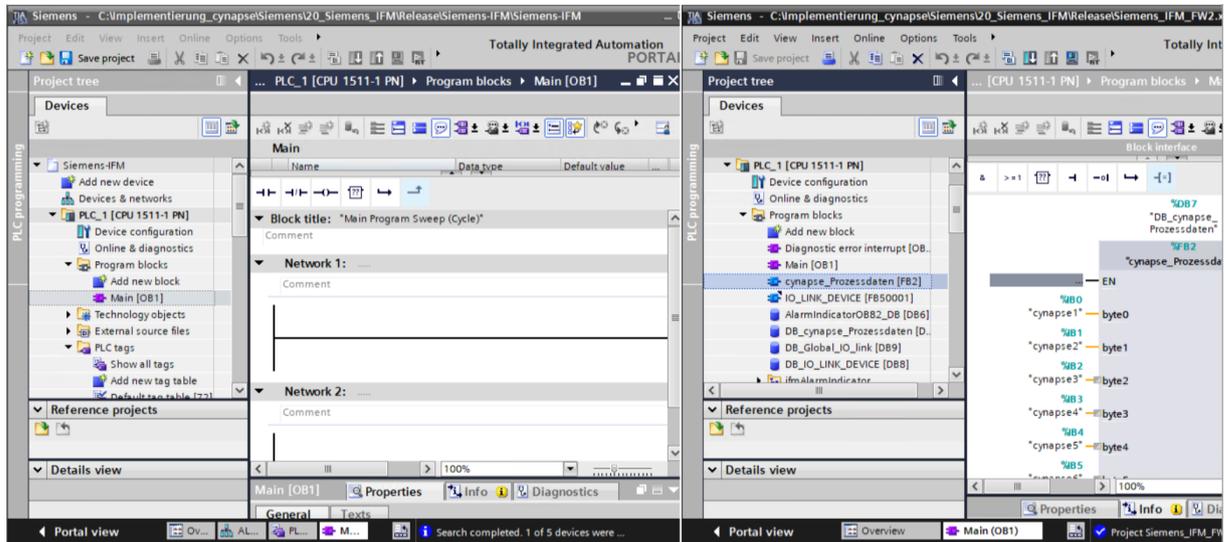


Ein-/Ausgang	Datentyp	Funktion
<b>Input 0</b>	BYTE	Byte 0 Reserviert
<b>Input 1</b>	BYTE	Byte 1 Prozessdatenprofil
<b>Input 2-14</b>	INT	Prozessdatenprofilabhängige Eingangsvariablen - nähere Informationen hierzu in der Betriebsanleitung
<b>Out 1</b>	INT	-
<b>Out 2</b>	REAL	Aktuell aktives Prozessdatenprofil
<b>Out 3-9</b>	REAL	Skalierte Werte der Inputs 2-14

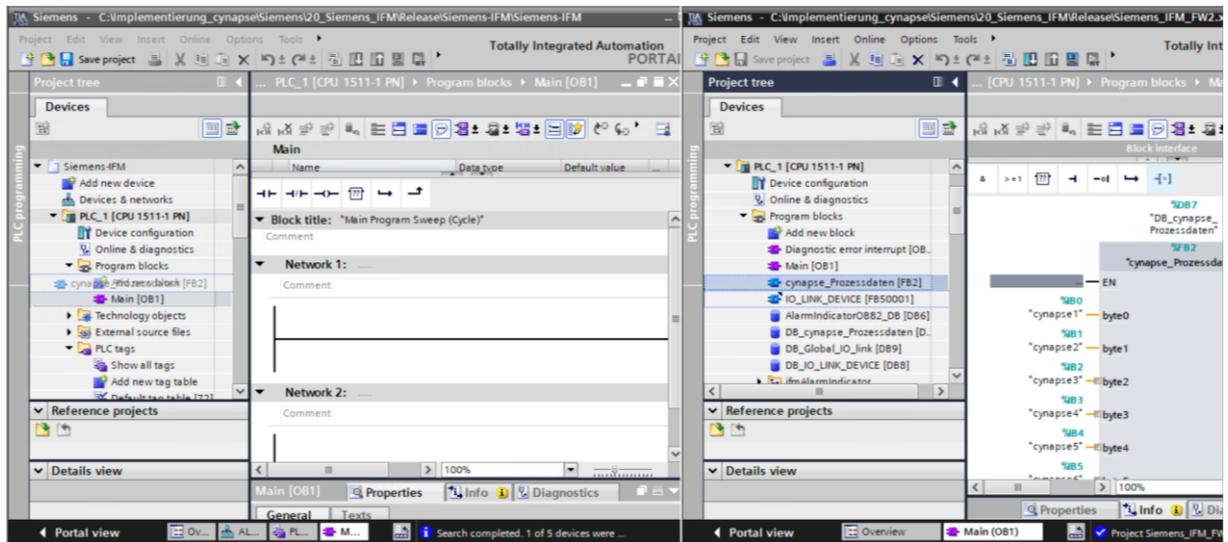
Tbl - 1

## Vorgehen

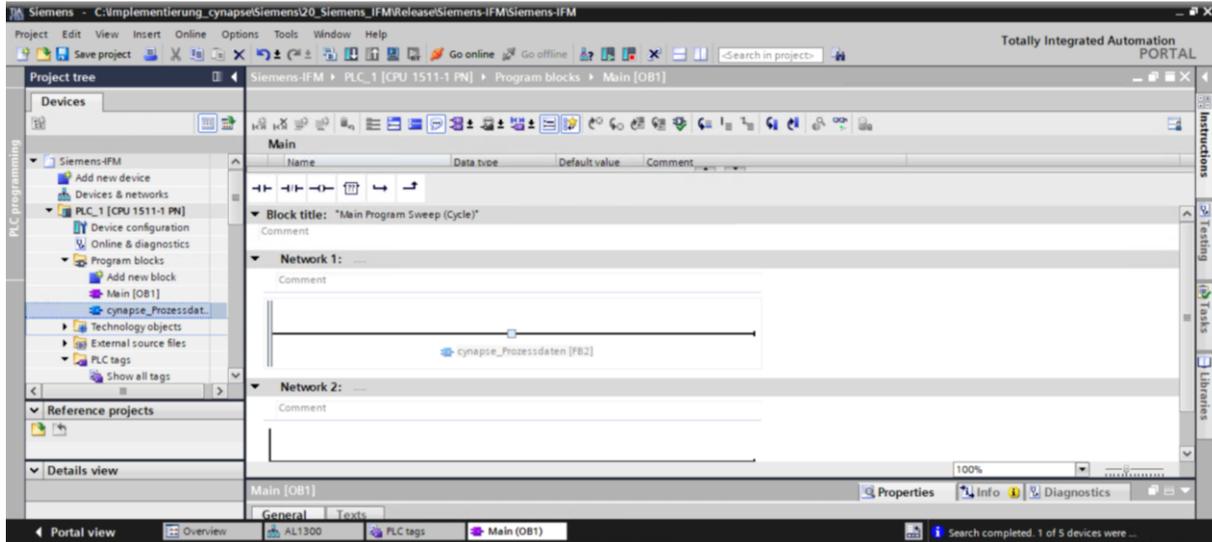
1. Öffnen Sie das erhaltene Beispielprojekt.
2. Öffnen Sie parallel das Projekt, in welchem Sie Prozessdaten lesen möchten.
3. Markieren Sie den FB „cynapse\_Prozessdaten“ im Beispielprojekt.



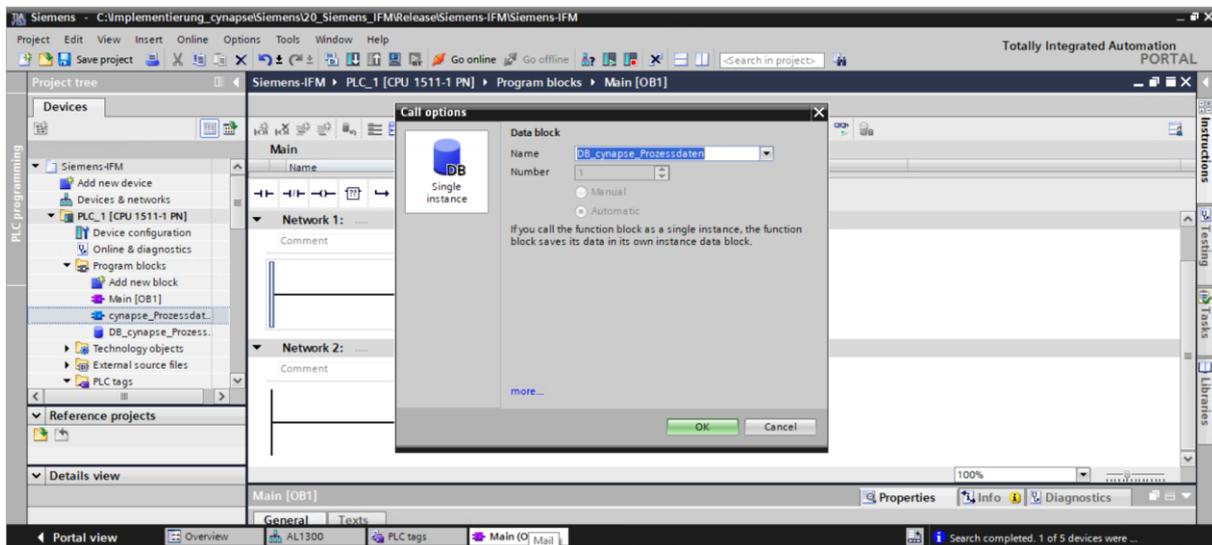
4. Ziehen Sie diesen per Drag and Drop unter „Program blocks“ in Ihr Projekt.



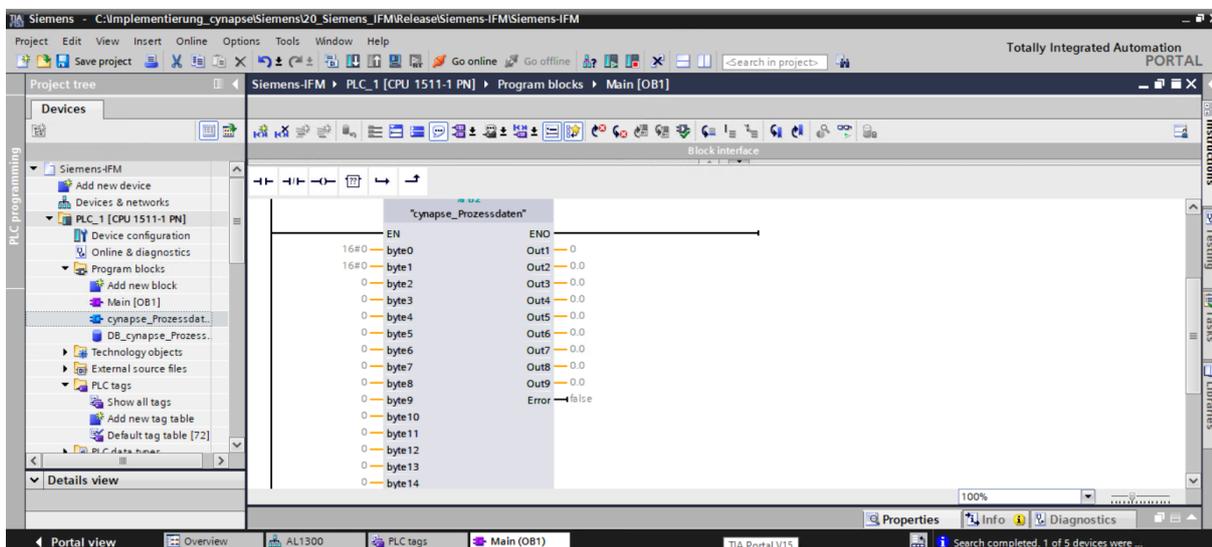
5. Ziehen Sie den FB per Drag and Drop in den Main-Baustein.



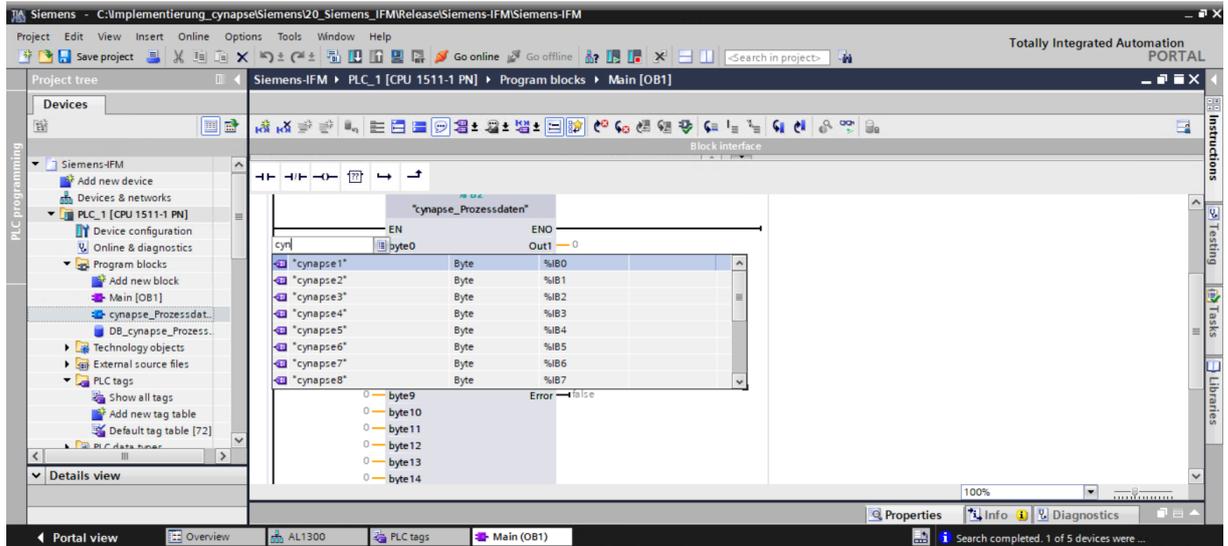
6. Legen Sie den DB durch Klick auf „OK“ an.



7. Der Baustein ist in einem Netzwerk des Main-Bausteins enthalten.



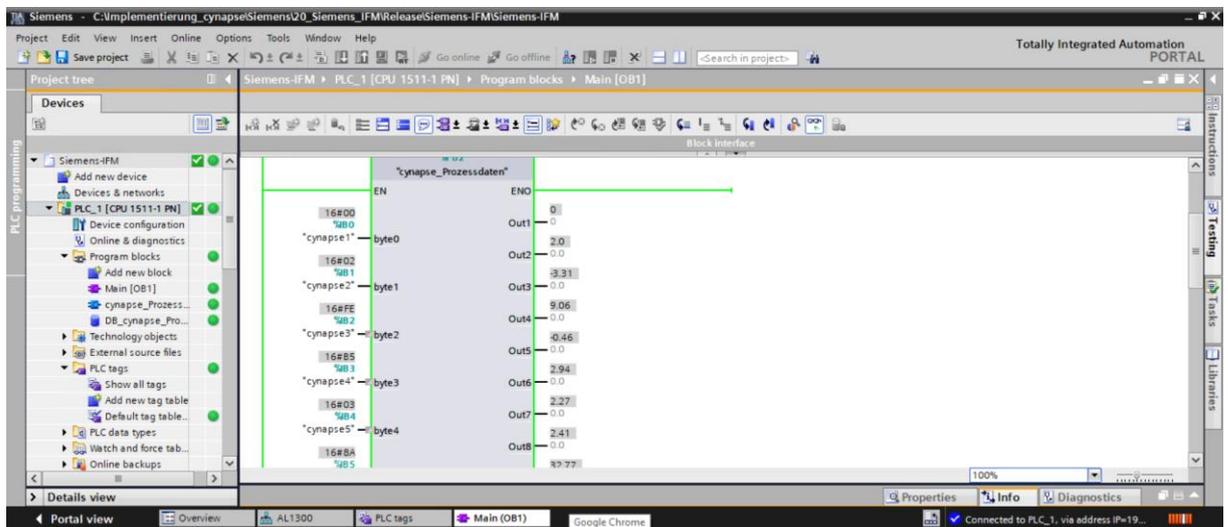
## 8. PLC tags aus der Variablen-tabelle den Eingängen zuweisen.



### Ergebnis

- ➔ An den Ausgängen des Bausteins liegen die skalierten Prozessdaten nach gewähltem Prozessdatenabbild an und können im Programm weiterverwendet werden.

Nach Kompilieren  und Download  lassen sich die Daten online  beobachten. 



## 5 Parameter

### 5.1 Definition

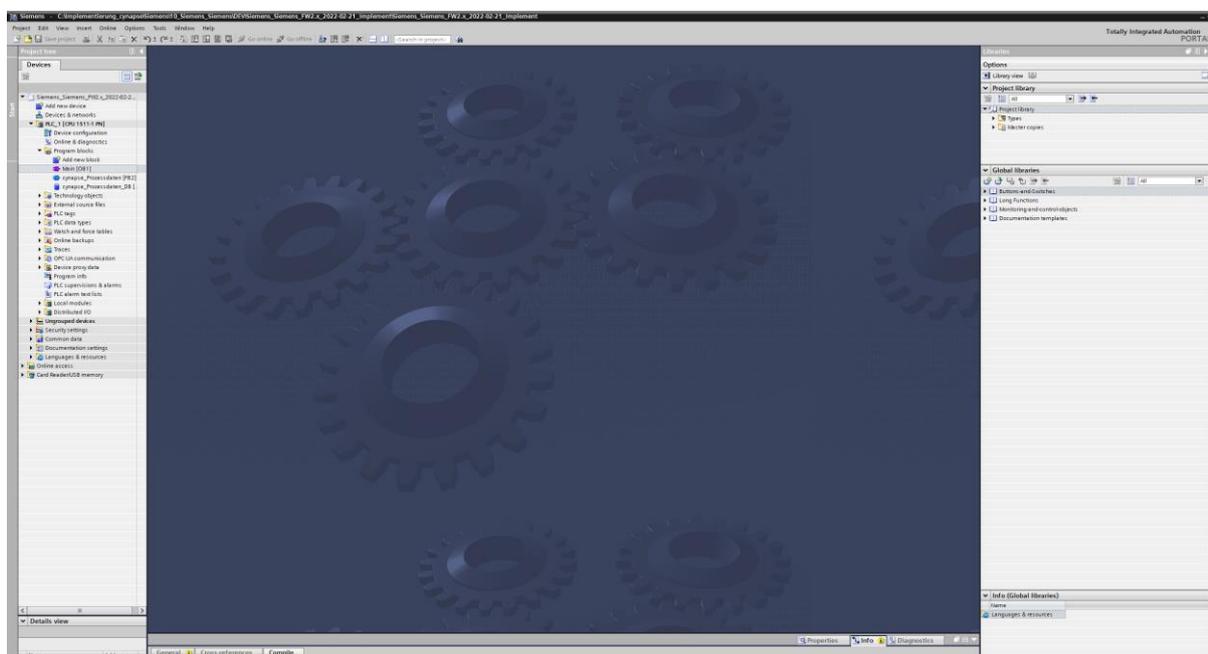
Unter Parametern versteht man azyklisch kommunizierte Daten. Hierdurch können Geräteparameter wie Geräteinformationen, Schwellwerte oder Diagnosedaten eines IO-Link Device (z.B. cynapse®) gelesen beziehungsweise geschrieben werden. Die Daten auf dem Device werden mit Index und Subindex eindeutig adressiert.

Nähere Informationen zu Index und Subindex sowie den Aufbau des Datensatzes finden Sie in der Betriebsanleitung cynapse®.

### 5.2 Siemens Baustein für Parameter lesen/schreiben in Programm einbinden

#### Voraussetzung

- ➔ Sie haben die Bibliothek LIOLink über die Website von Siemens bezogen und kennen den Ablageort.
- ➔ Ihr TIA Projekt ist geöffnet und die Task Card Bibliotheken ist geöffnet.



- ⓘ **Hinweis:** Eine doppelte Anfrage an ein Device über den IO\_LINK\_DEVICE Baustein ist nicht möglich. Da Parameter lesen, Parameter schreiben und Blob-Daten auf diesen Baustein zugreifen sind diese gegeneinander zu verriegeln.

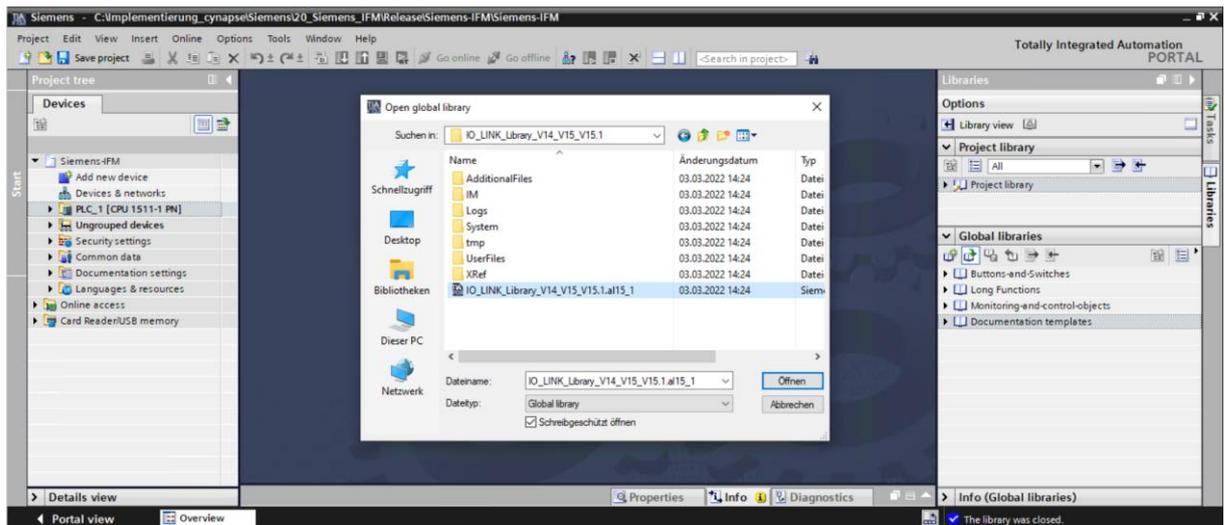
## Einleitung

Im Folgenden binden Sie den von Siemens veröffentlichten Funktionsbaustein "FBIOLinkDevice" in Ihr Projekt ein und ermitteln projektspezifische Eingangsgrößen des Bausteins.

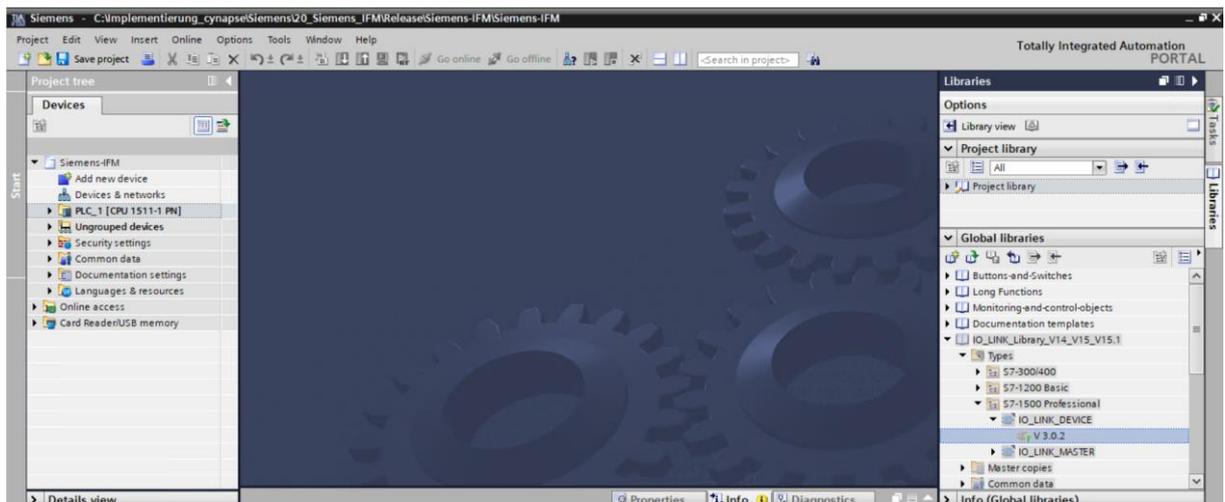
Mit Hilfe dieses Funktionsbausteins können Sie Programmparameter, Messwerte und Diagnosedaten von einem IO-Link Device lesen bzw. Geräteparameter zu einem IO-Link Device schreiben oder Systemkommandos senden.

## Vorgehen

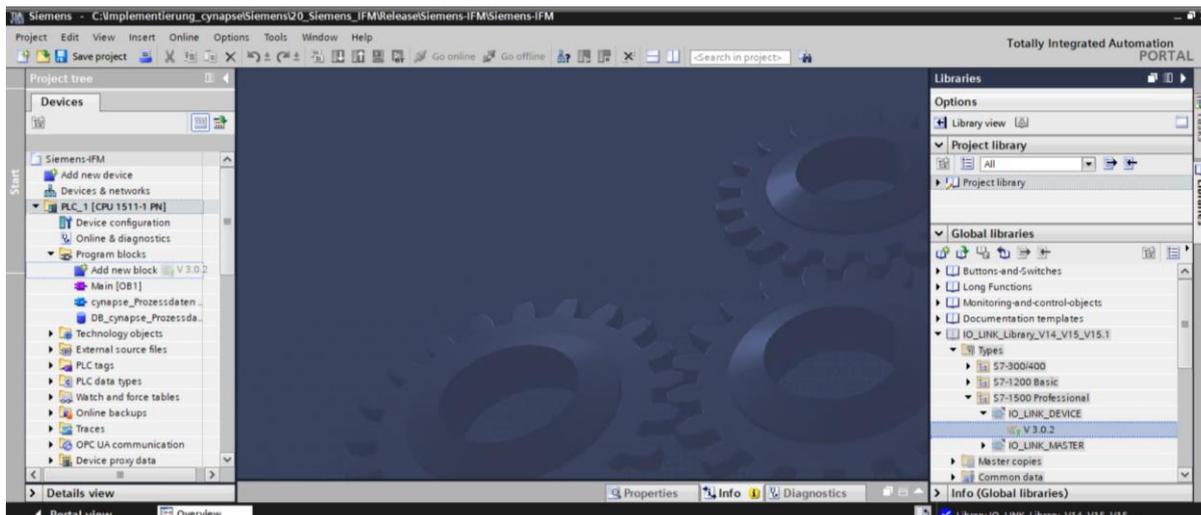
1. Klicken Sie auf die Schaltfläche „Open global library“.
2. Wählen Sie die Bibliothek am bekannten Speicherort aus.
3. Klicken Sie auf „Öffnen“.



4. Die Bibliothek erscheint unter „Global libraries“ und kann aufgeklappt werden.

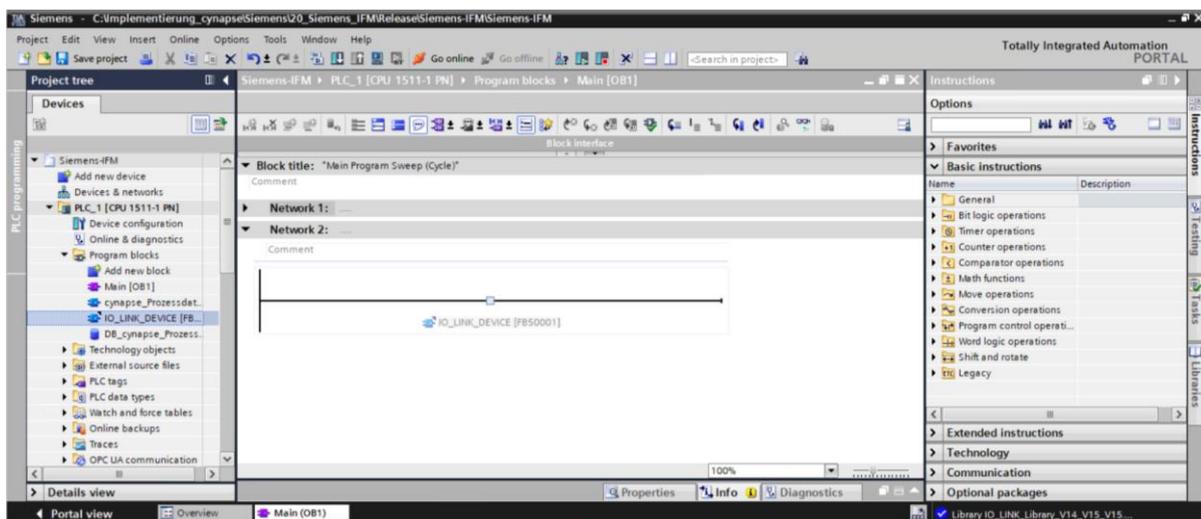


5. Ziehen Sie den Baustein IO\_LINK\_DEVICE per Drag and Drop in Programmbausteine.

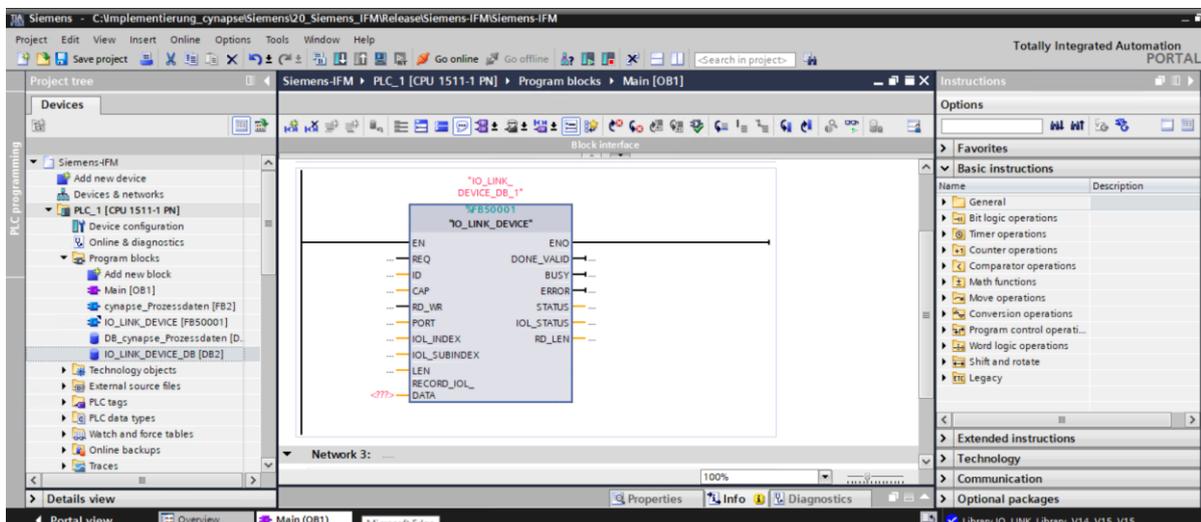


6. Öffnen Sie mit einem Doppelklick den Main-Baustein.

7. Ziehen Sie den IO\_LINK\_DEVICE Baustein per Drag and Drop in das Netzwerk.

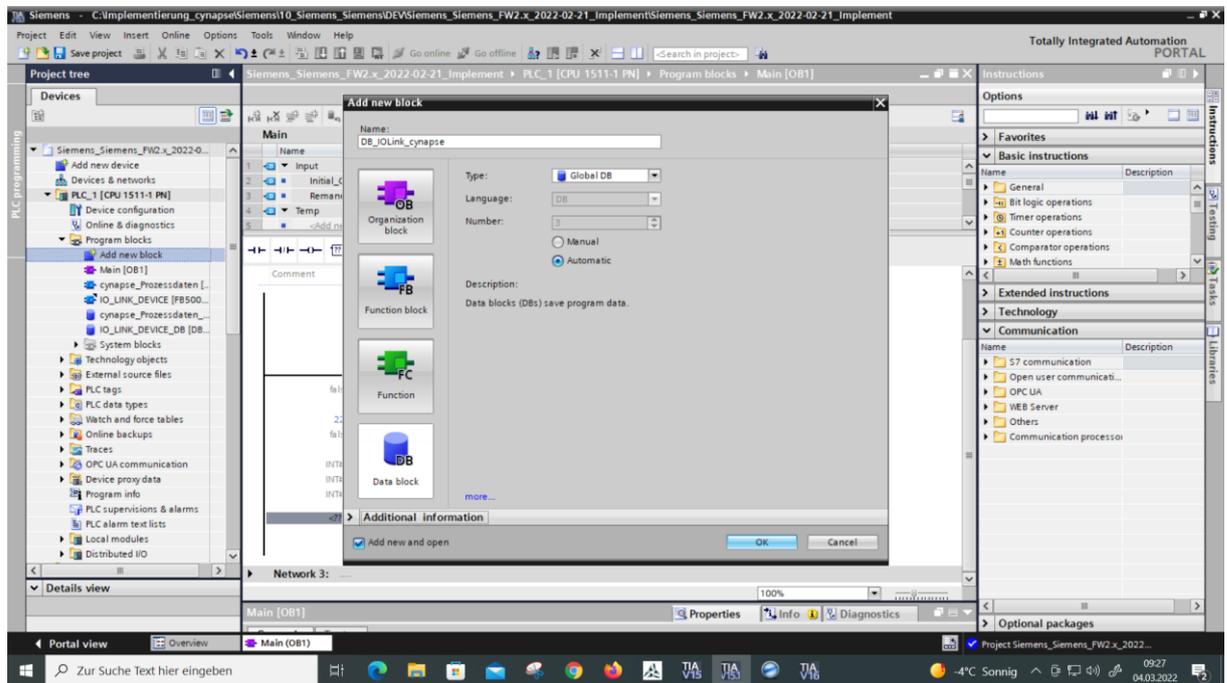


8. Lassen Sie den zugehörigen DB durch Anweisung des TIA Portals erstellen.

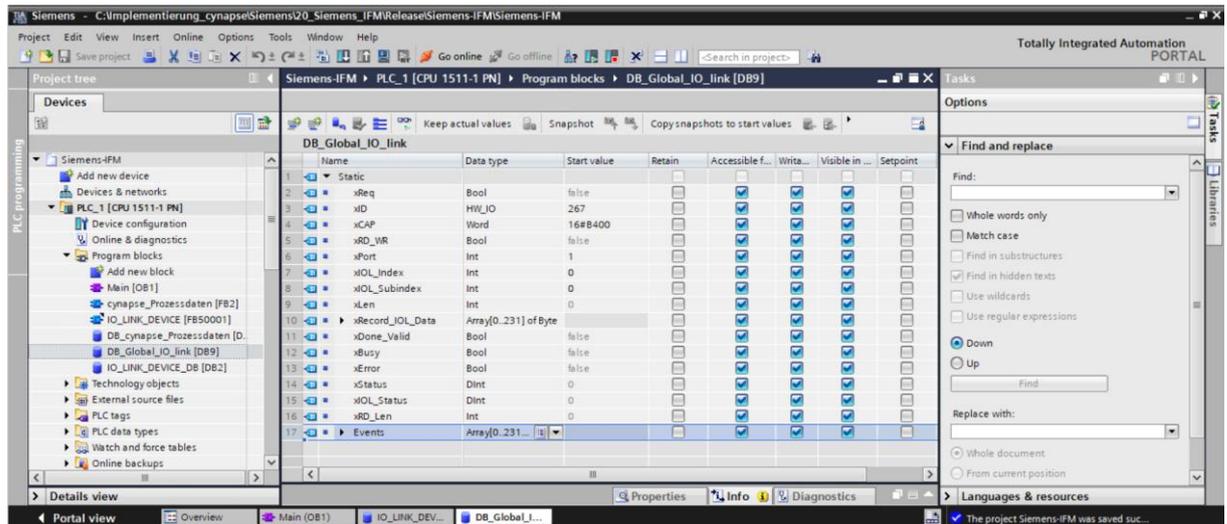


9. Legen Sie den globalen Datenbaustein DB für Ein- und Ausgänge an:

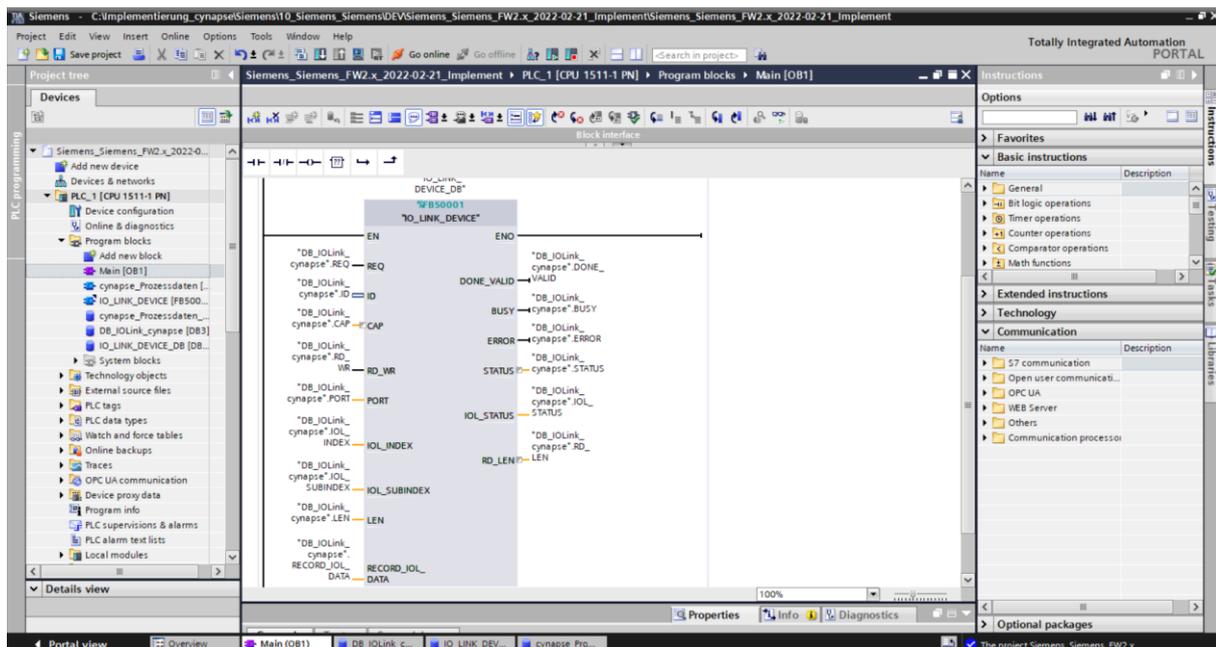
- Unter „Program blocks“ Klick auf „Add new block“.
- DB auswählen und benennen.
- „OK“ klicken.



- Globalen DB durch Doppelklick auf diesen öffnen.
- Variablen in den geforderten Datentypen gemäß der Bausteinbeschreibung von Siemens im DB anlegen.

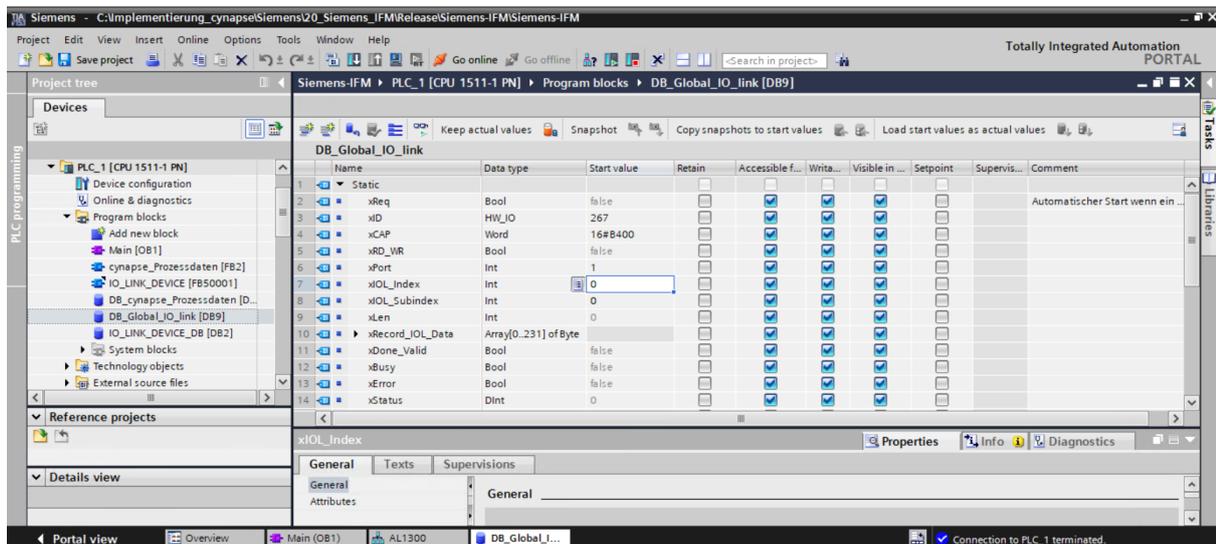


10. Verknüpfen Sie Ein- und Ausgänge des Funktionsbausteins in dem Main-Baustein mit den erstellten Variablen im globalen Datenbaustein.

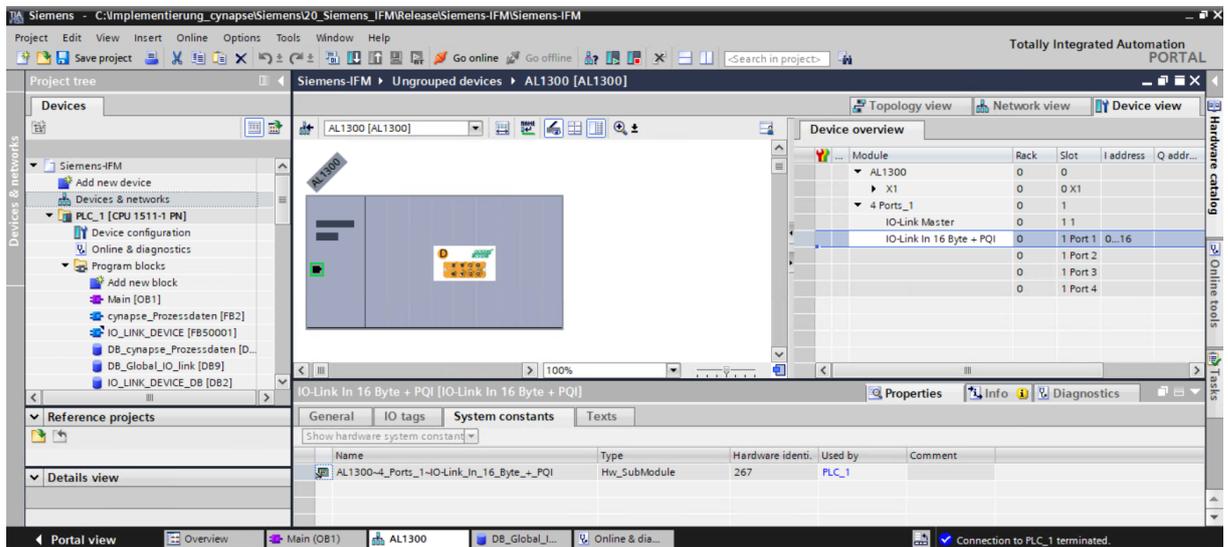


11. Öffnen Sie den erstellten globalen DB durch Doppelklick.

12. Setzen Sie projektspezifische Eingangsgrößen:

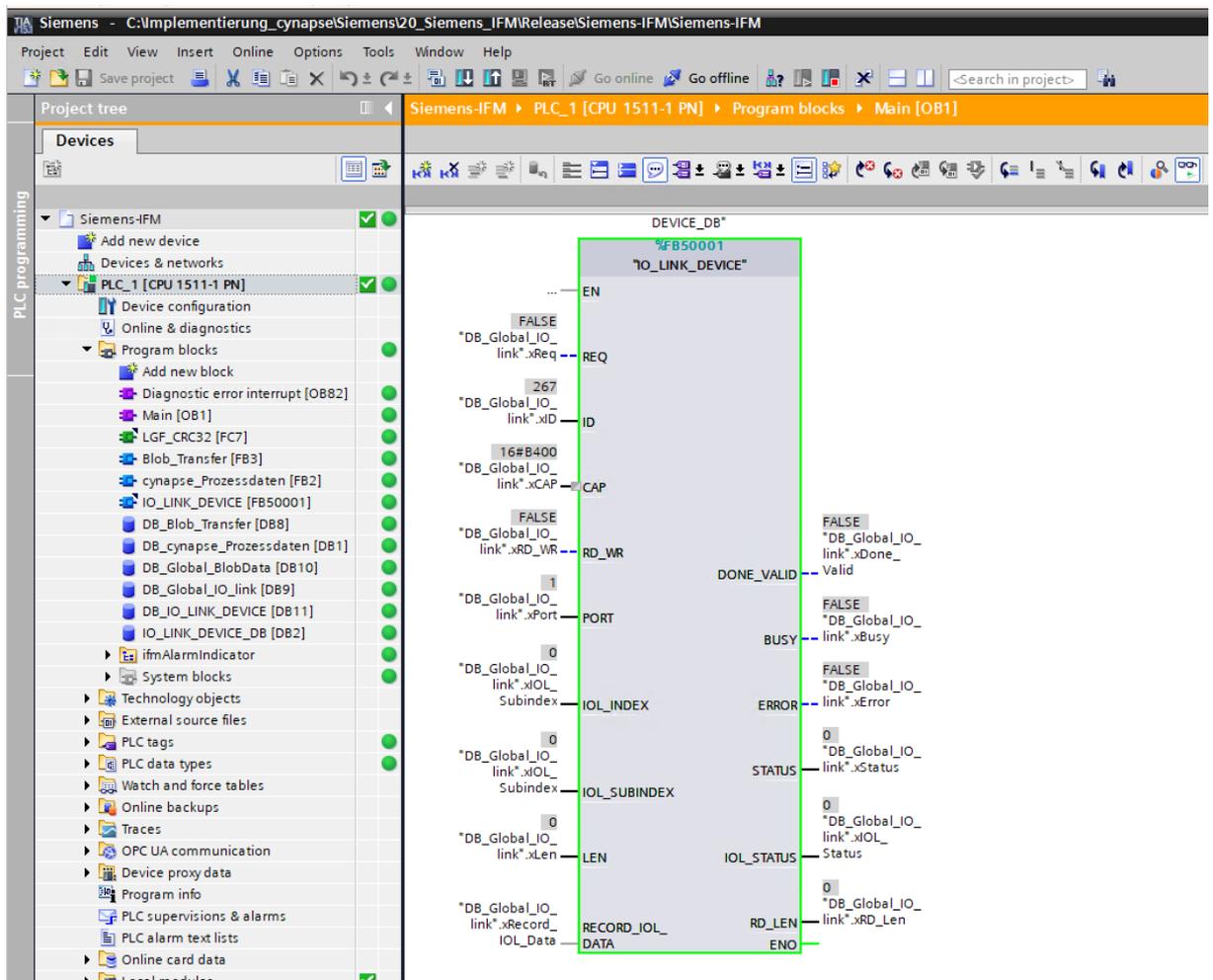


- ID: Hardwareerkennung des IO-Link Kommunikationsmoduls: Diese Information ist in den Systemkonstanten in der Hardwaresicht zu finden = 267
- CAP: Zugangspunkt (Client Access Point): Diese Information ist in der Masterdokumentation zu finden = 16#B400
- PORT: Portnummer an dem das IO-Link Device angeschlossen ist.



### Ergebnis

- ➔ Der Funktionsbaustein ist in dem Main-Baustein eingebunden.
- ➔ Alle Ein- und Ausgänge sind mit dem erstellten Datenbaustein verknüpft.
- ➔ Projektspezifische Eingangsgrößen wurden ermittelt und als Startwerte im DB gesetzt.



## 5.3 Parameter lesen

### Voraussetzung

- ➔ Das Projekt ist geöffnet und der Funktionsbaustein IO\_Link Device wurde wie in Kapitel 5.2 „Siemens Baustein für Parameter lesen/schreiben in Programm einbinden“ beschrieben in dieses eingebunden.
  - ➔ Das Projekt wurde erfolgreich in die Hardware geladen und ein Onlinezugriff ist möglich.
  - ➔ Die Informationen zu Index und Subindex des gewünschten Parameters wurden ermittelt. Die allgemeinen Indizes sind der IO-Link Spezifikation zu entnehmen. Die cynapse® spezifischen Indizes sind in der Betriebsanleitung zu finden.
- ① **Hinweis:** Eine doppelte Anfrage an ein Device über den IO\_LINK\_DEVICE Baustein ist nicht möglich. Da Parameter lesen, Parameter schreiben und Blob-Daten auf diesen Baustein zugreifen, sind diese gegeneinander zu verriegeln.

### Einleitung

Im Folgenden lesen Sie mit Hilfe des eingebundenen Funktionsbausteins "FBloLinkDevice" beispielhaft das aktuell ausgegebene Prozessdatenformat über den Parameter Settings aus cynapse® aus.

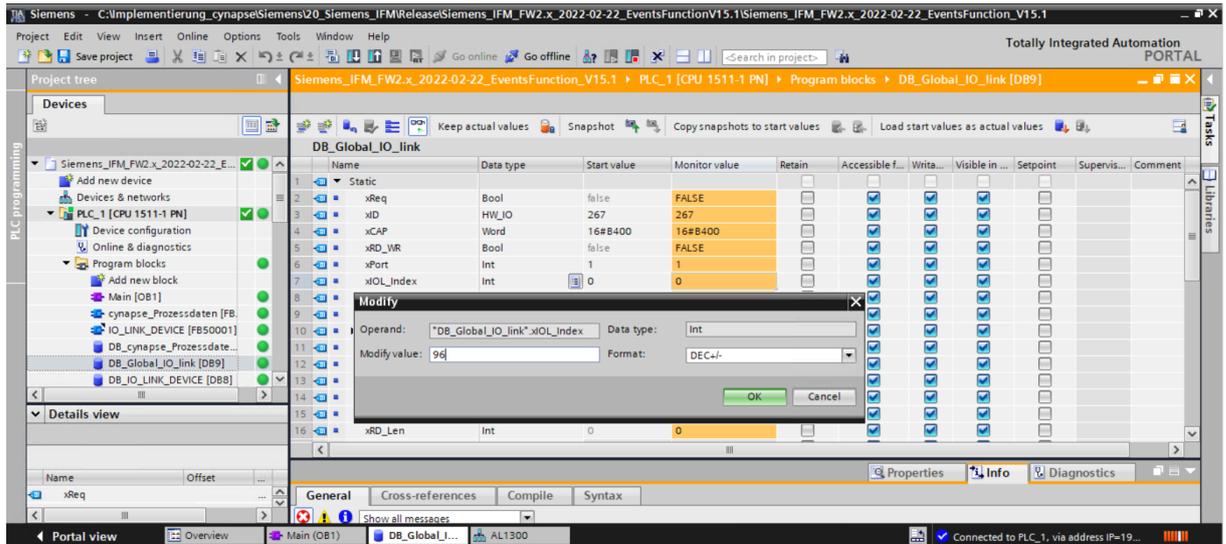
Vorgehend einige Informationen zum Baustein:

- Die Datenübertragung erfolgt in Form von Rohdaten (ARRAY of Byte)
- Ist "RD\_WR" = FALSE, werden Daten ausgelesen und an "RECORD\_IOL\_DATA" ausgegeben.
- Solange noch keine gültigen Antwortdaten eingetroffen sind, wird dies über den Ausgang "BUSY" = TRUE signalisiert.
- Der Wert TRUE des Ausgangs "DONE\_VALID" zeigt, dass die Übertragung erfolgreich durchgeführt wurde. Bei einem Leseauftrag liegen die Daten nun konsistent am Ein-/Ausgang "RECORD\_IOL\_DATA" vor und der Ausgang "RD\_LEN" zeigt die Länge der empfangenen Daten an.
- Der Wert TRUE des Ausgangs "ERROR" zeigt, dass ein Fehler aufgetreten ist. Solange der Eingang REQ = TRUE ist, behalten die Ausgangsparameter ihren Wert bei. Ist der Eingang REQ = FALSE, bevor die Bearbeitung des FB abgeschlossen ist, werden die Werte der Ausgangsparameter nach der Bearbeitung des Auftrages nur für einen Zyklus lang gehalten.

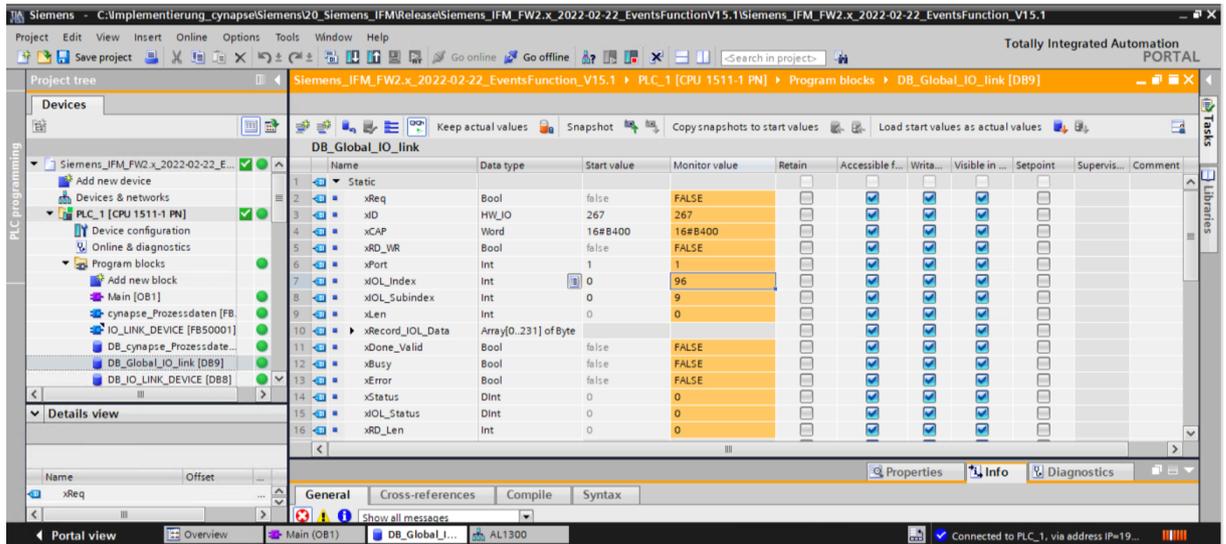
### Vorgehen

1. Öffnen Sie den Datenbaustein mit Ein-/Ausgangsvariablen des Bausteins durch Doppelklick.
2. Betätigen Sie die Schaltfläche „Go online“ .
3. Starten Sie das Onlinemonitoring. .
4. Ändern Sie den Index mit einem Doppelklick auf „Monitor value“ des zu ändernden Eingangs.
5. Geben Sie den Index ein.

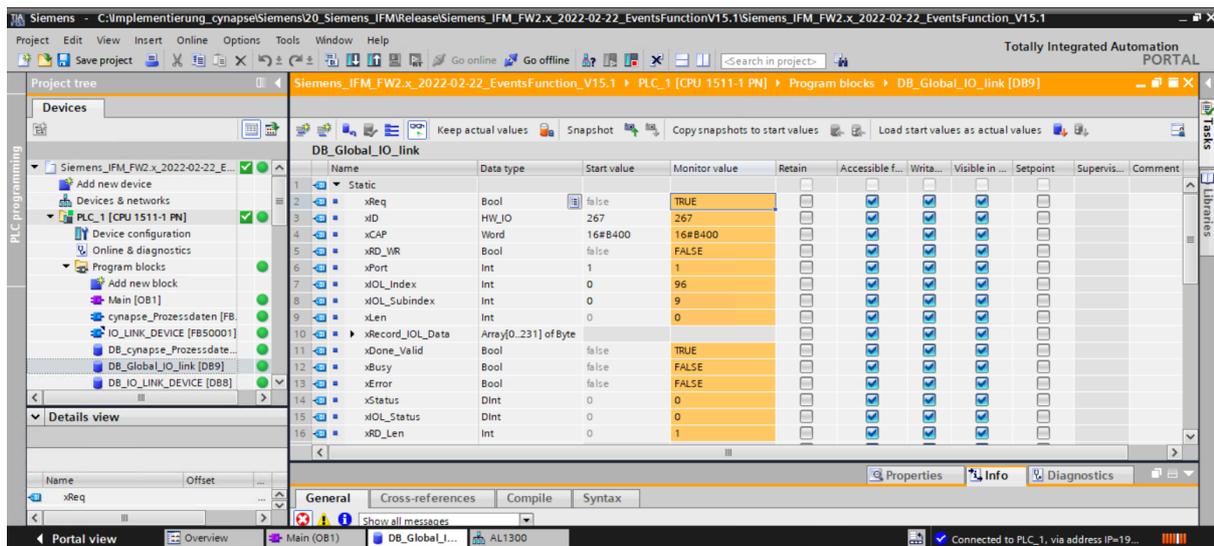
6. Bestätigen Sie mit „OK“.



7. Geben Sie den Subindex nach gleichem Schema ein.



8. Setzen Sie den Eingang REQ durch einen Doppelklick auf „Monitor value“ von FALSE auf TRUE.



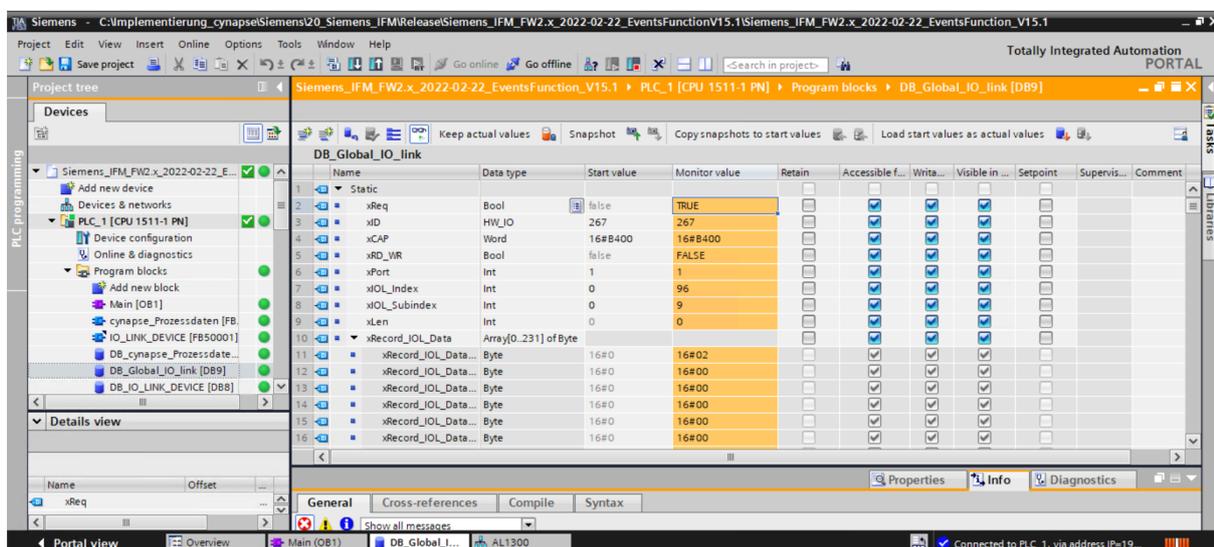
9. DONE\_VALIDE TRUE zeigt erfolgreiches Auslesen an.

10. RD\_LEN zeigt Länge des gelesenen Parameters an.

11. RECORD\_IOL\_DATA aufklappen.

## Ergebnis

➔ Im Array RECORD\_IOL\_DATA werden die gelesenen Parameter codiert angezeigt.



Dieser gibt in diesem Beispiel das aktuell ausgegebene Prozessdatenformat an.

## 5.4 Parameter schreiben

### Voraussetzung

- ➔ Das Projekt ist geöffnet und der Funktionsbaustein IO\_Link Device wurde wie in Kapitel 5.2 „Siemens Baustein für Parameter lesen/schreiben in Programm einbinden“ beschrieben in dieses eingebunden.
  - ➔ Das Projekt wurde erfolgreich in die Hardware geladen und ein Onlinezugriff ist möglich.
  - ➔ Die Informationen zu Index, Subindex und Länge des gewünschten Parameters wurden ermittelt. Die allgemeinen Indizes sind der IO-Link Spezifikation zu entnehmen. Die cynapse® spezifischen Indizes sind in der Betriebsanleitung zu finden.
- ① **Hinweis:** Eine doppelte Anfrage an ein Device über den IO\_LINK\_DEVICE Baustein ist nicht möglich. Da Parameter lesen, Parameter schreiben und Blob-Daten auf diesen Baustein zugreifen sind diese gegeneinander zu verriegeln.
- ① **Hinweis:** Ist die Länge des zu schreibenden Parameters größer als 1 Byte, muss dieser in ein Bytearray konvertiert werden. Parameter des Typs Float müssen z.B. mit Hilfe des Befehls „REAL\_TO\_DWORD“ in ein 4 Byte langes Array konvertiert werden.

### Einleitung

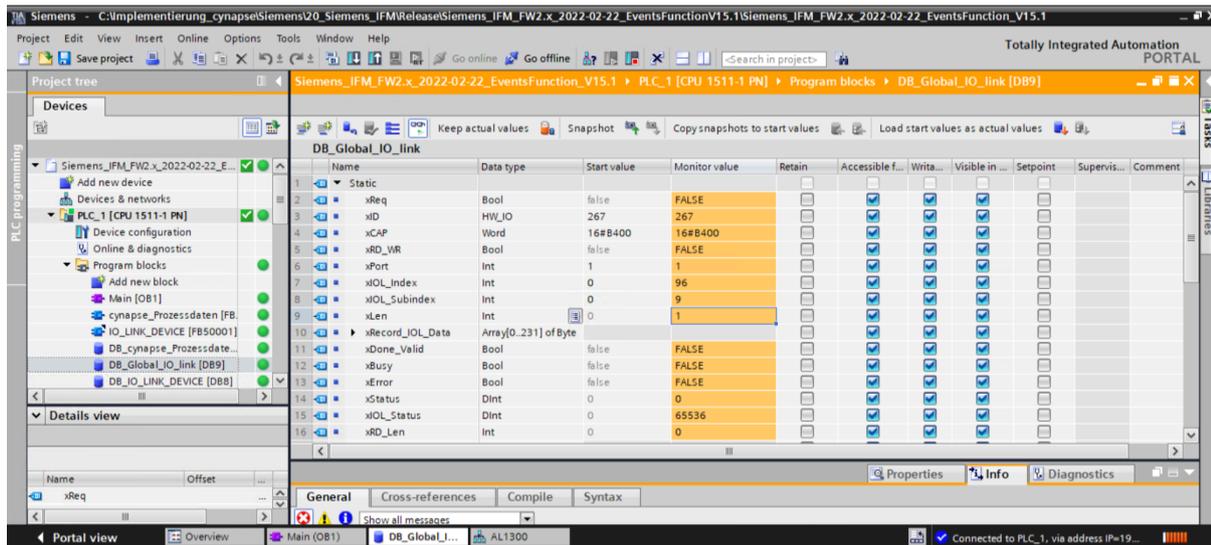
Im Folgenden ändern Sie mit Hilfe des eingebundenen Funktionsbausteins "FBloLinkDevice" beispielhaft das aktuell ausgegebene Prozessdatenformat über den Parameter Settings von cynapse®. Hierfür wird ein Parameter geschrieben.

Vorgehend einige Informationen zum Baustein:

- Die Datenübertragung erfolgt in Form von Rohdaten (ARRAY of Byte)
- Ist "RD\_WR" = TRUE, werden die Daten aus "RECORD\_IOL\_DATA" in cynapse® geschrieben.
- Für einen Schreibauftrag muss die Länge des zu sendenden Parameters unter LEN angegeben werden.
- Solange noch keine gültigen Antwortdaten eingetroffen sind, wird dies über den Ausgang "BUSY" = TRUE signalisiert.
- Der Wert TRUE des Ausgangs "DONE\_VALID" zeigt, dass die Übertragung erfolgreich durchgeführt wurde.
- Der Wert TRUE des Ausgangs "ERROR" zeigt, dass ein Fehler aufgetreten ist. Solange der Eingang REQ = TRUE ist, behalten die Ausgangsparameter ihren Wert bei. Ist der Eingang REQ = FALSE, bevor die Bearbeitung des FB abgeschlossen ist, werden die Werte der Ausgangsparameter nach der Bearbeitung des Auftrages nur für einen Zyklus lang gehalten.

## Vorgehen

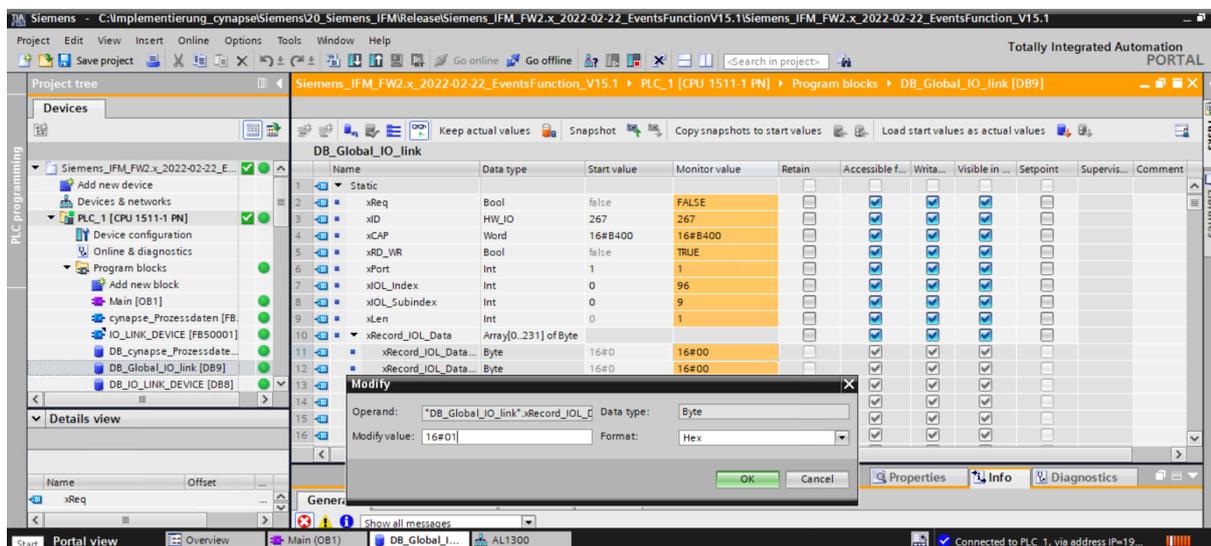
1. Öffnen Sie den globalen Datenbaustein mit Ein-/Ausgangsvariablen des Bausteins durch Doppelklick.
2. Betätigen Sie die Schaltfläche „Go online“.
3. Starten Sie das Onlinemonitoring.
4. Setzen Sie den Wert für die Eingangsvariable RD\_WR durch Doppelklick auf die Variable in der orangenen „Monitor value“ Spalte auf TRUE.
5. Tragen Sie die Länge des zu schreibenden Parameters in LEN ein.



The screenshot shows the Siemens TIA Portal interface. The main window displays the configuration for the data block 'DB\_Global\_IO\_link'. The table below represents the data block structure:

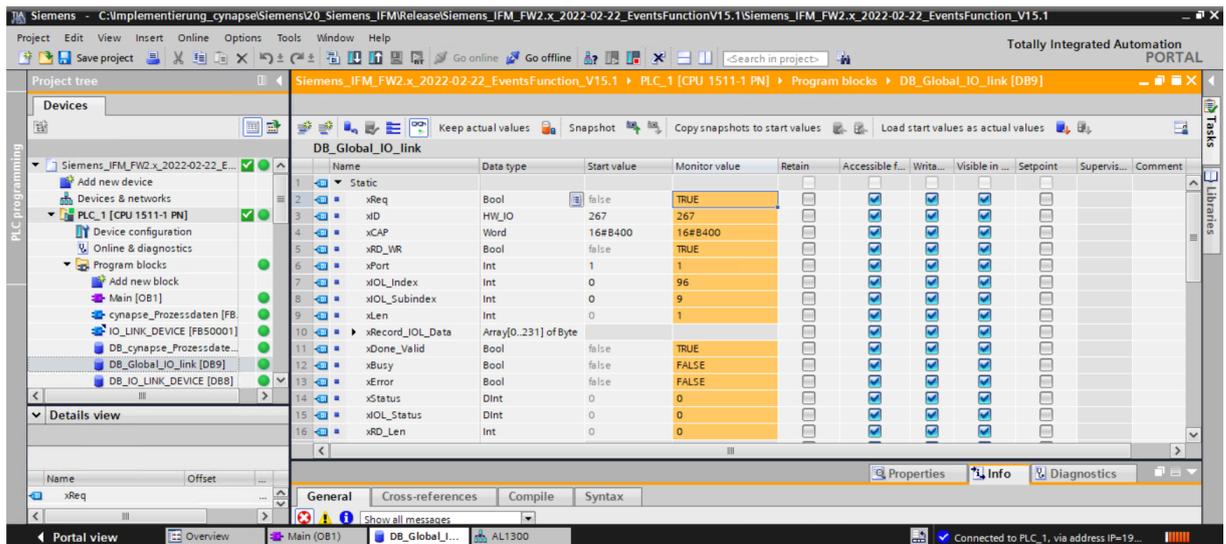
Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
Static										
xReq	Bool	false	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xID	HW_IO	267	267		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xCAP	Word	16#B400	16#B400		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xRD_WR	Bool	false	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xPort	Int	1	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xIOL_Index	Int	0	96		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xIOL_Subindex	Int	0	9		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xLen	Int	0	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xRecord_IOL_Data	Array[0..231] of Byte				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xDone_Valid	Bool	false	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xBusy	Bool	false	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xError	Bool	false	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xStatus	Dint	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xIOL_Status	Dint	0	65536		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
xRD_Len	Int	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			

6. Tragen Sie in Array RECORD\_IOL\_DATA im ersten Byte das gewünschte Prozessdatenformat = 1 ein.



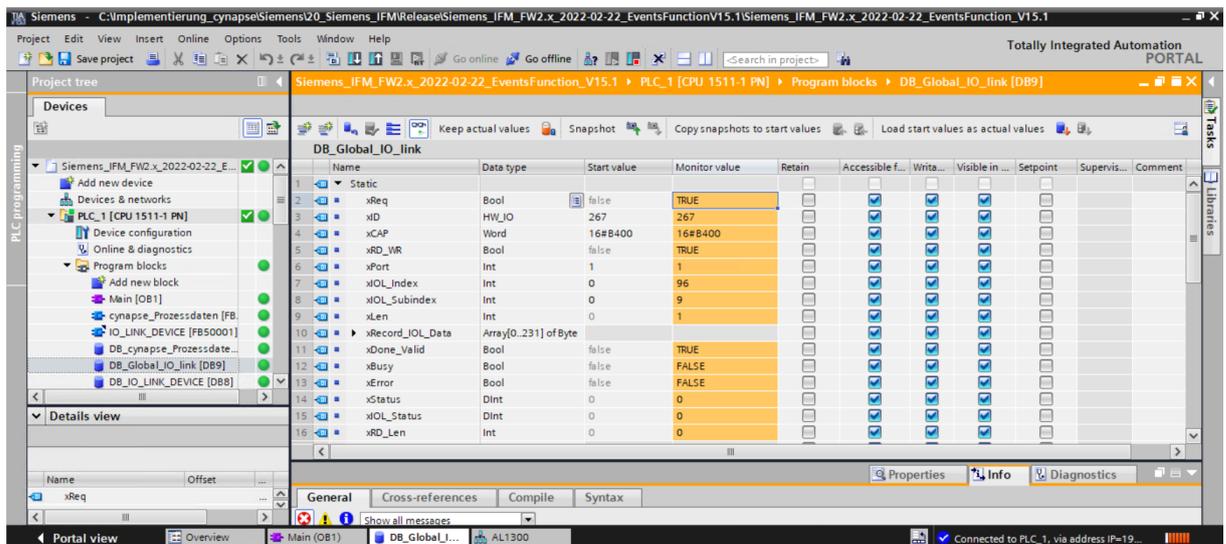
The screenshot shows the same TIA Portal interface as above, but with a 'Modify' dialog box open. The dialog box is used to edit the value of the first byte of the 'xRecord\_IOL\_Data' array. The 'Operand' is set to 'DB\_Global\_IO\_link'.xRecord\_IOL\_D' and the 'Data type' is 'Byte'. The 'Modify value' is entered as '16#01' and the 'Format' is set to 'Hex'.

7. Setzen Sie den REQ Eingang durch einen Doppelklick auf „Monitor value“ von FALSE auf TRUE.



Ergebnis

➔ DONE\_VALID TRUE zeigt erfolgreiches Schreiben des Parameters an.



Die Änderung kann durch Auslesen des Parameters wie in Kapitel 5.3 „Parameter lesen“ beschrieben überprüft werden.

## 6 Events

### 6.1 Definition

Unter Events versteht man die Meldung eines nicht korrekten Betriebszustands des IO-Link-Device. Beispiele hierfür sind zu hohe Betriebstemperatur, Vibrationen oder ein detektierter defekt am Gerät. Dabei liegen Warnungen oder Fehler in Form von Events immer nur dann an, wenn Schwellwerte über- bzw. unterschritten sind und werden automatisch zurückgesetzt. cynapse® unterscheidet zwischen von WITTENSTEIN definierten Grenzwerten und benutzerdefinierten Grenzwerten. Letztere können über Parameter schreiben geändert werden. Detaillierte Informationen zu Parametern für benutzerdefinierten Grenzwerten und fehlerspezifische Codes finden Sie in der Betriebsanleitung cynapse®.

### 6.2 Events auslesen

#### Voraussetzung

- ➔ Sie haben ein Beispielprojekt für das Auslesen von Events über folgende Quelle bezogen: [cybertronic-support@wittenstein.de](mailto:cybertronic-support@wittenstein.de)

#### Einleitung

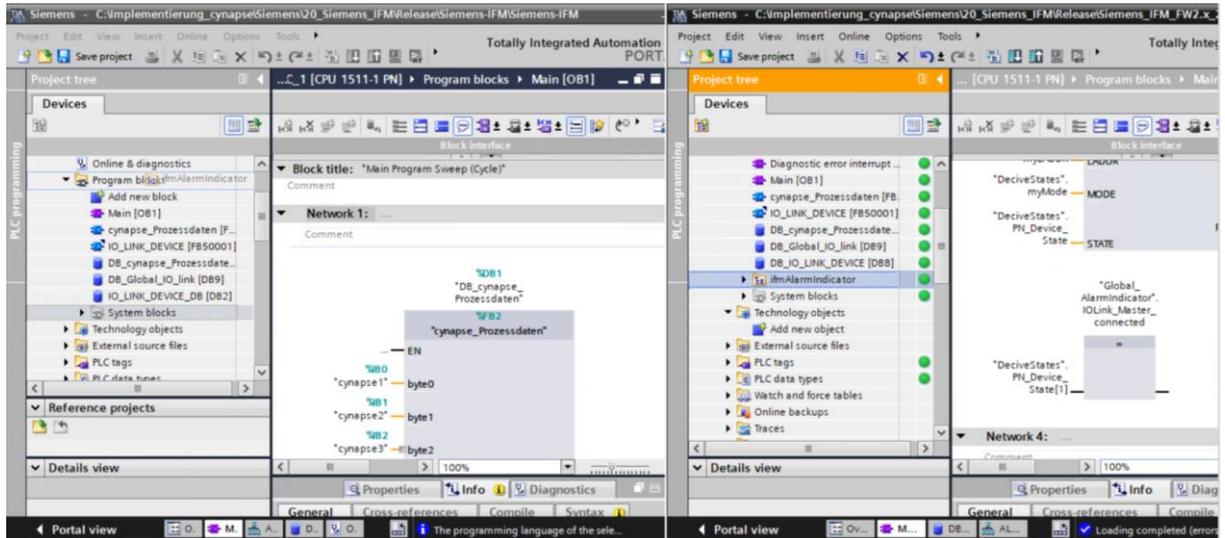
Im Folgenden lesen Sie anstehende Events mit Hilfe eines Funktionsbausteins aus einem Beispielprojekt aus. Durch Bewegungen von cynapse® werden hierfür mehrere Events provoziert.

Um Events auslesen zu können, muss in cynapse® eine generelle Eventfreigabe erfolgen. Diese Freigabe wird im Parameter Index = 96, Subindex = 1 erteilt. Außerdem werden Events über die Subindexe 2, 3, 5 und 7 zur Meldung freigeschalten. Nähere Informationen hierfür finden Sie in der Betriebsanleitung cynapse®.

- ⓘ **Hinweis:** Eine doppelte Anfrage an ein Device über den IO\_LINK\_DEVICE Baustein ist nicht möglich. Da Parameter lesen, Parameter schreiben und Blob-Daten auf diesen Baustein zugreifen sind diese gegeneinander zu verriegeln.

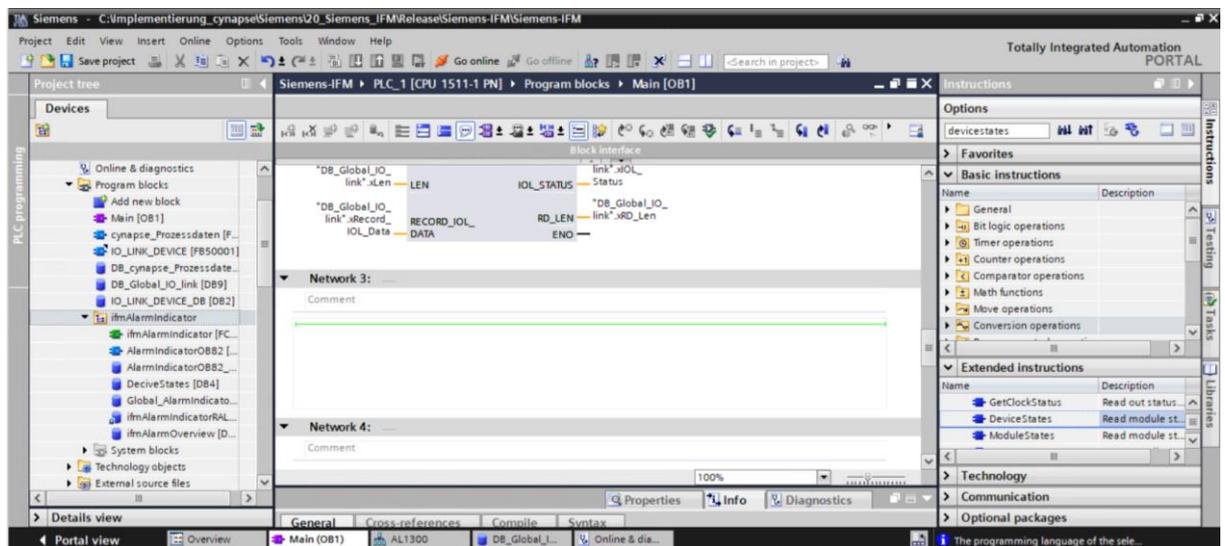
### Vorgehen

1. Öffnen Sie das erhaltene Beispielprojekt.
2. Öffnen Sie parallel das Projekt, in welchem Sie Events überwachen möchten.
3. Markieren Sie den Ordner IFMAlarmindicator.
4. Ziehen Sie diesen per Drag and Drop unter „Program blocks“ in Ihr Projekt.

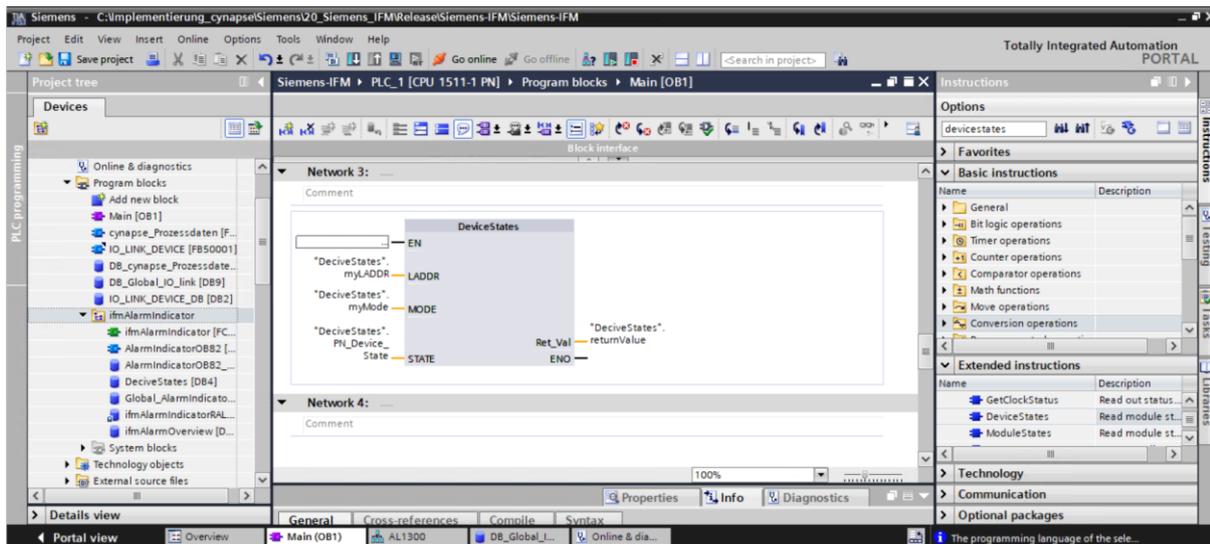


5. Ziehen Sie per Drag and Drop den Baustein „DeviceStates“ aus den Extended Instructions in ein beliebiges Netzwerk.

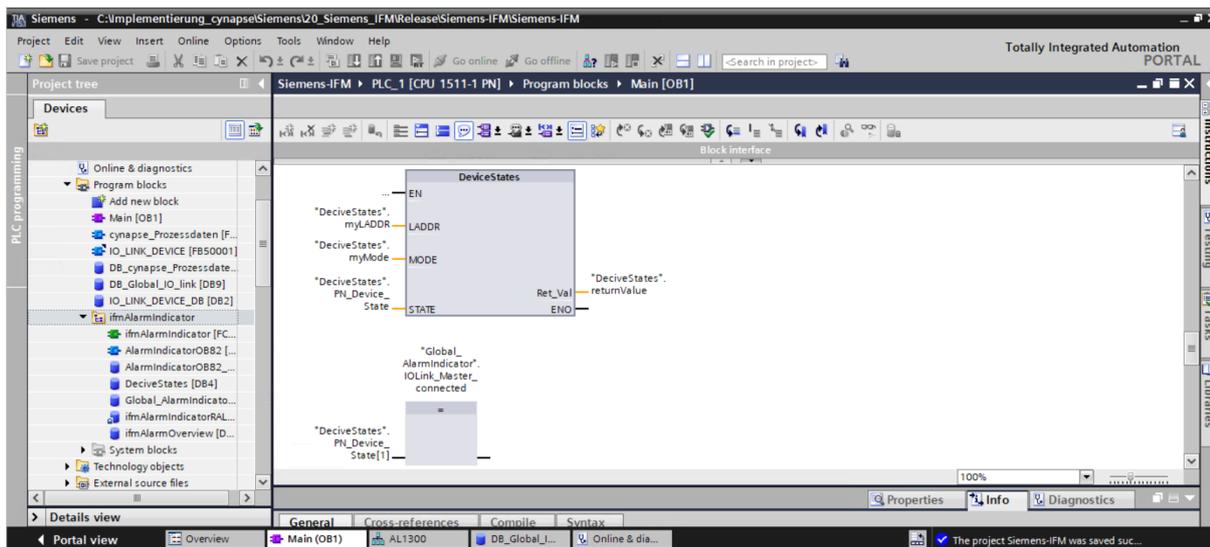
ⓘ Dieser Baustein überprüft die Aktivität des angeschlossenen Ports.



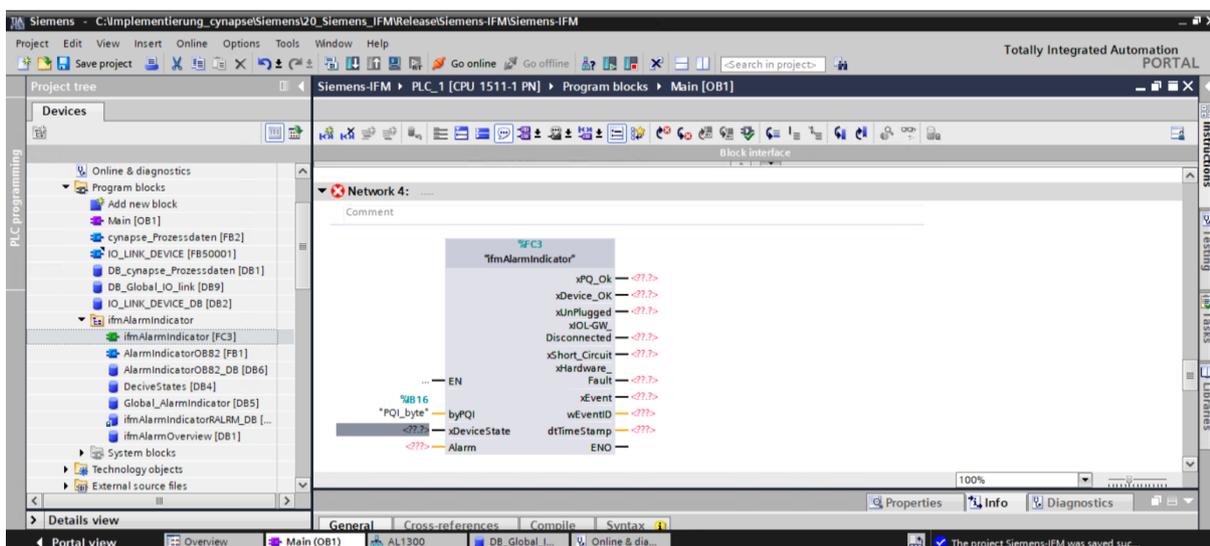
6. Verknüpfen Sie die Eingänge mit den entsprechenden Variablen aus dem Globalen DB „DeviceStates“.



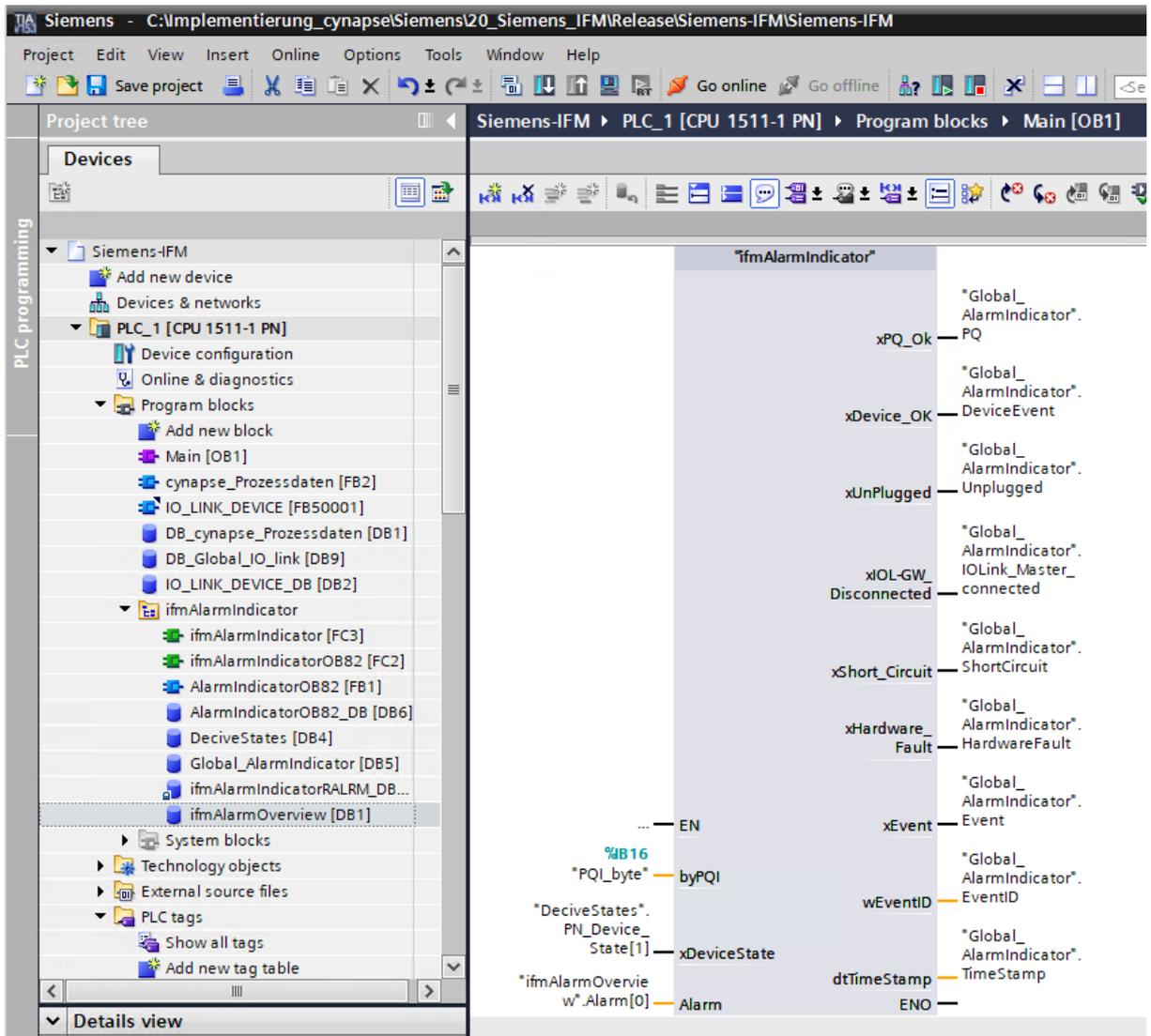
7. Weisen Sie anschließend den angeschlossenen Port zu.



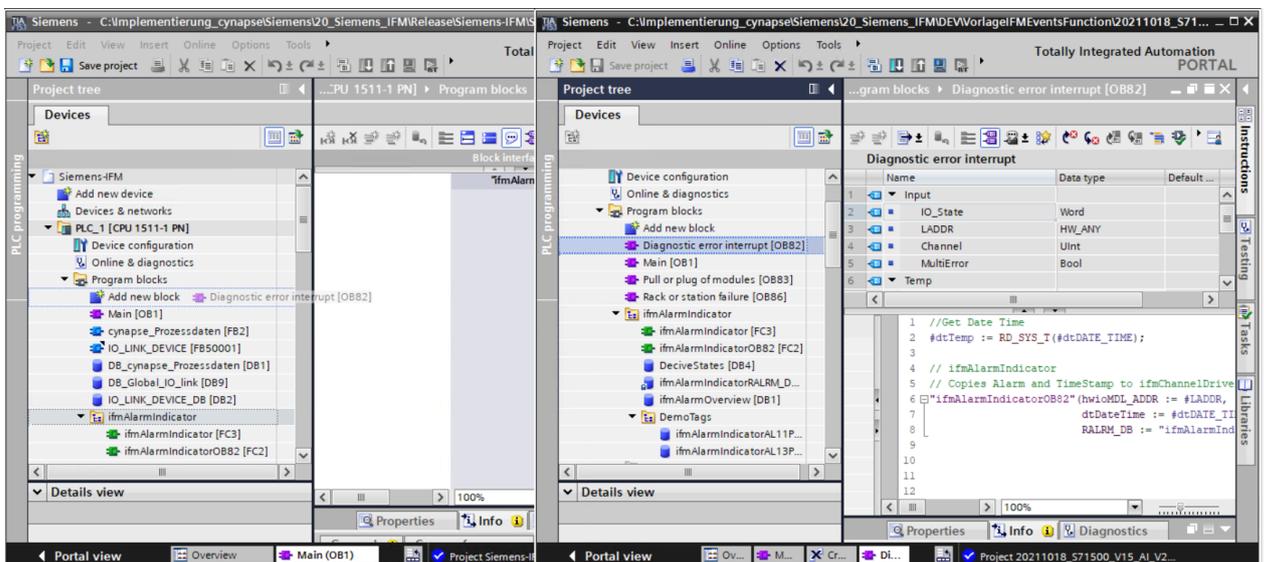
8. Ziehen Sie den Baustein „ifmAlarmIndicator“ in ein freies Netzwerk.



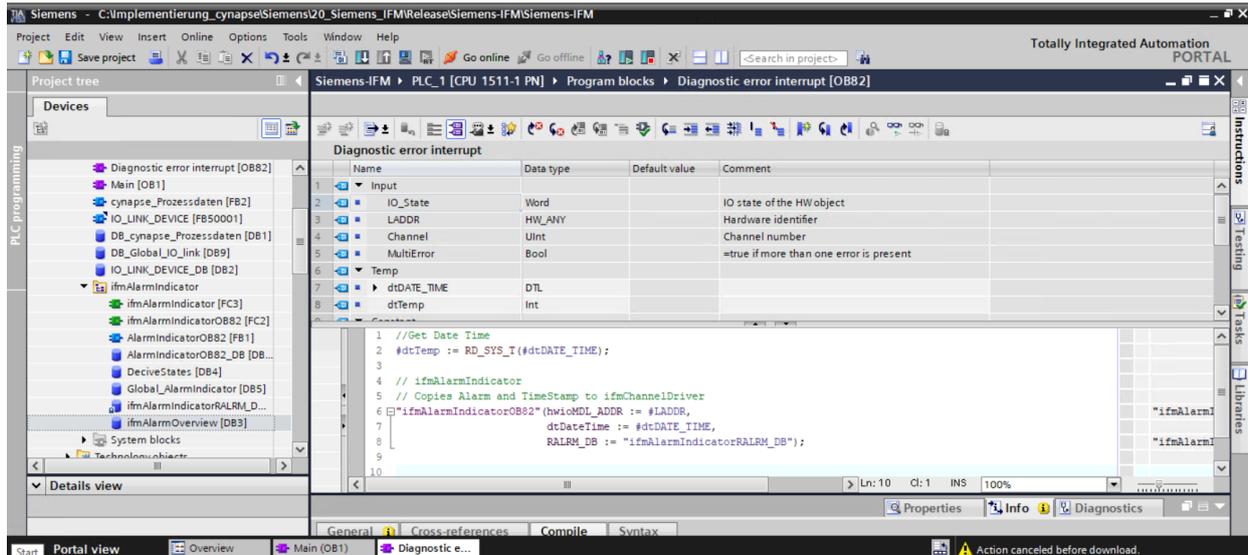
9. Verknüpfen Sie die Ein- und Ausgänge mit den entsprechenden Variablen aus dem globalen DB „Global\_AlarmIndicator“.



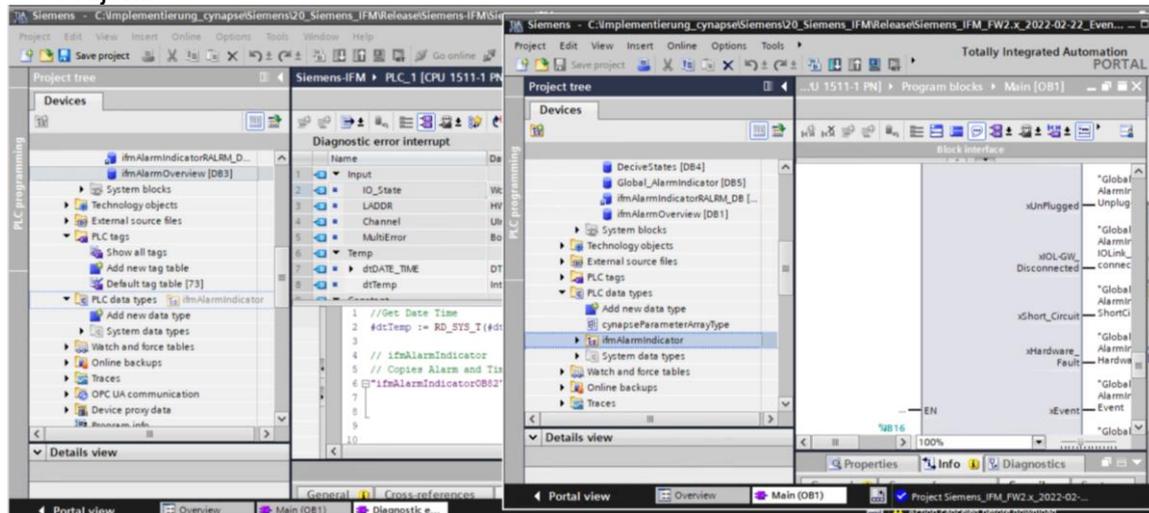
10. Ziehen Sie per Drag an Drop den OB82 „Diagnostic Error Interrupt“ in Ihren Projektbaum.



11. Der OB82 wird aktiv, sobald ein Event anliegt und steuert den Baustein an, um das Event auszulesen.

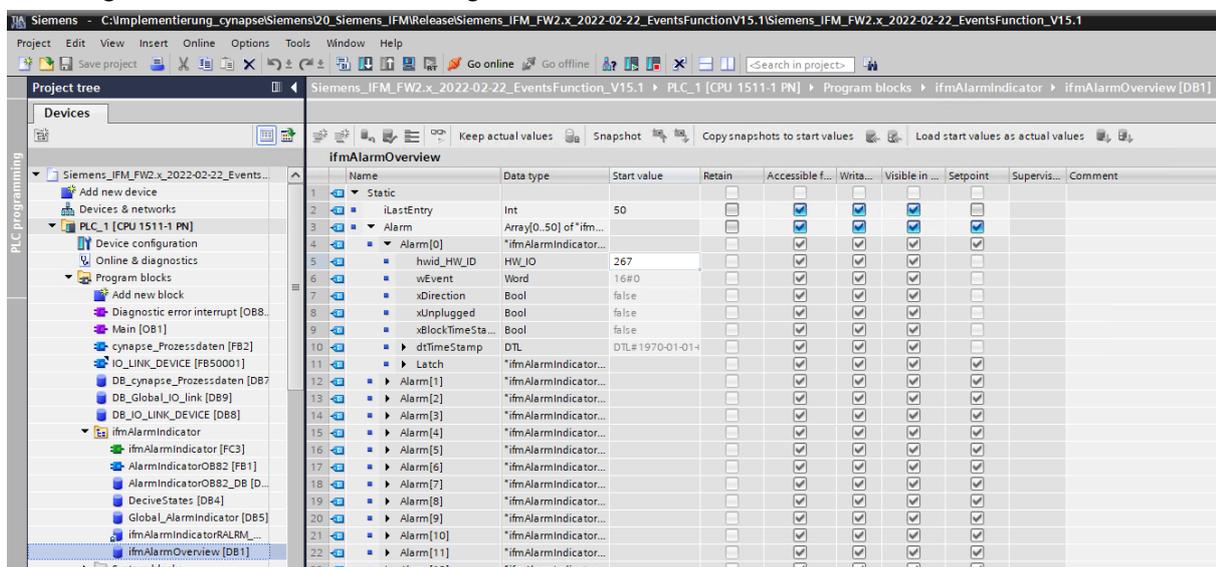


12. Ziehen Sie per Drag and Drop den Ordner „ifmAlarmIndicator“ aus den PLC data Types in Ihr Projekt.



13. Tragen Sie die korrekte HardwareID in den DB „ifmAlarmOverview“ ein.

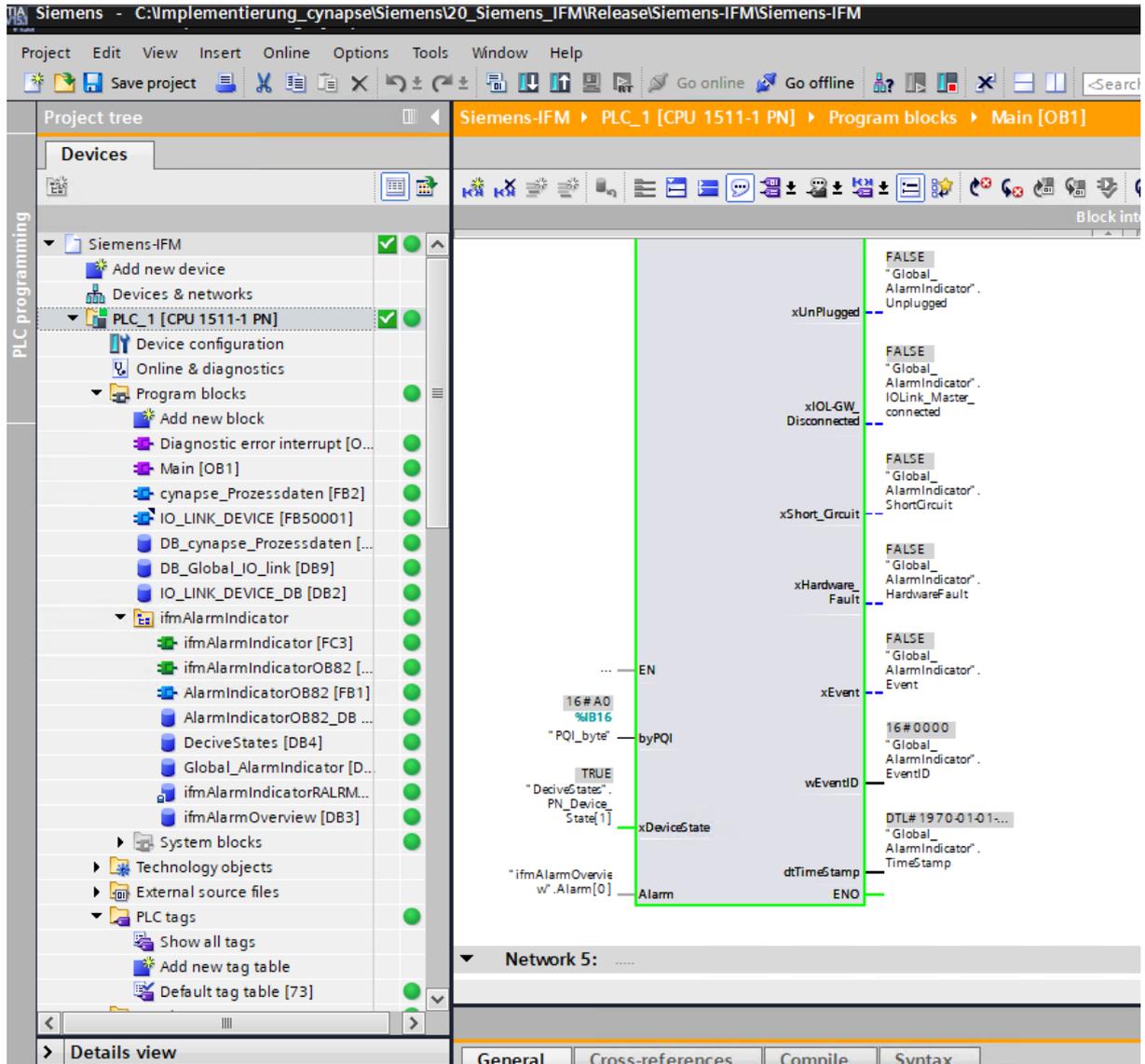
① Diese finden Sie in Kapitel 5.2 „Siemens Baustein für Parameter lesen/schreiben in Programm einbinden“, Handlungsschritt 12.



14. Kompilieren Sie das Programm.
15. Spielen Sie das Programm auf die Hardware.
16. Betätigen Sie die Schaltfläche „Go online“.
17. Starten Sie das Onlinemonitoring.

### Ergebnis

➔ Der anstehende Eventcode ist an dem Ausgang wEventID zu finden.



## 7 Blob-Daten

### 7.1 Definition

IO-Link definiert den Transfer größerer Datenmengen (**Binary large object**) durch das BLOB Transfer Profil. Hierbei wird der Typ des zu übertragenden Datenblocks über die BLOB\_ID zwischen 1 und 32767 identifiziert. Das Vorzeichen der ID gibt die Richtung des Transfers an; ein positives Vorzeichen kennzeichnet die Datenflussrichtung vom Master zum Device, ein negatives Vorzeichen bedeutet Datenfluss vom Device zum Master.

cynapse® bietet verschiedene per BLOB Transfer transportierte Datenpakete an. Nähere Informationen zur BLOB\_ID und zur Decodierung der ausgelesenen Datenpakete finden Sie in der Betriebsanleitung cynapse®.

### 7.2 Blob-Daten mit Hilfe des „Blob\_Transfer“ FB lesen

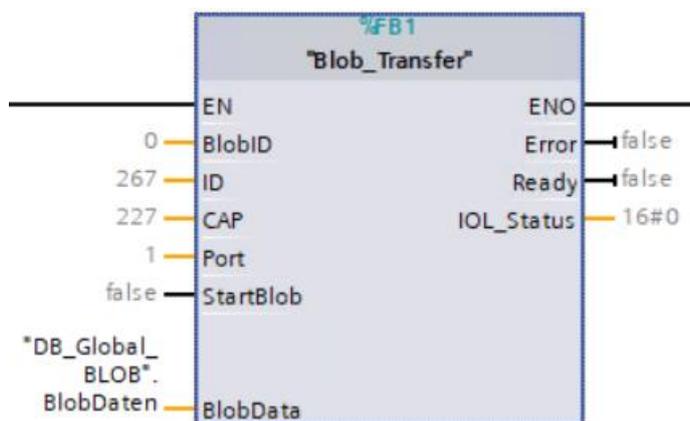
#### Voraussetzung

➔ Sie haben ein Beispielprojekt für Blob-Transfer über folgende Quelle bezogen:  
cybertronic-support@wittenstein.de

ⓘ **Hinweis:** Eine doppelte Anfrage an ein Device über den IO\_LINK\_DEVICE Baustein ist nicht möglich. Da Parameter lesen, Parameter schreiben und Blob-Daten auf diesen Baustein zugreifen sind diese gegeneinander zu verriegeln.

#### Einleitung

Im Folgenden lesen Sie Blob-Daten, mit Hilfe eines Funktionsbausteins, aus einem Beispielprojekt aus. Dieser übernimmt den spezifikationskonformen Ablauf des Blob-Transfers. Nähere Informationen hierzu finden sie in der IO-Link Spezifikation.

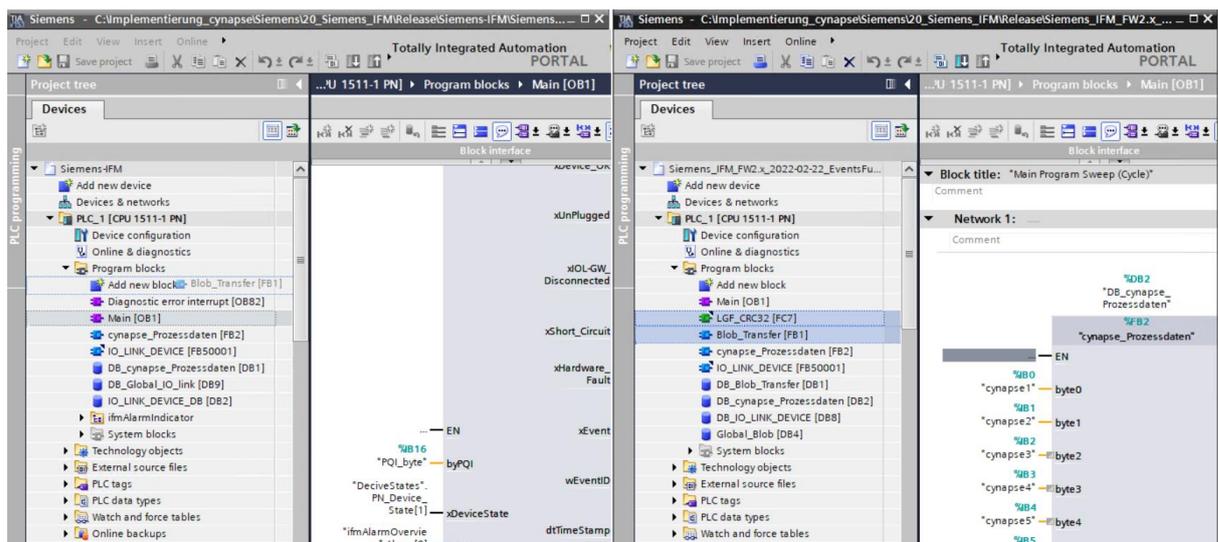


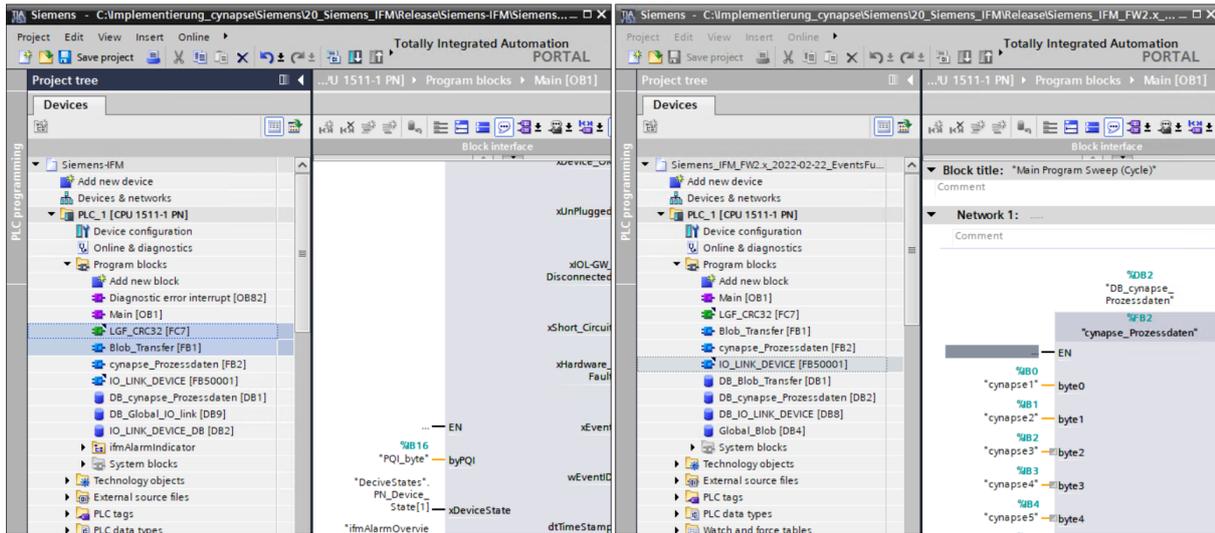
Ein/Ausgang	Datentyp	Funktion
<b>BlobID</b>	Integer	ID des zu übertragenden Datenblocks
<b>ID</b>	HW_IO	Hardwareerkennung des IO-Link Kommunikationsmoduls
<b>CAP</b>	Integer	Zugangspunkt (Client Access Point)
<b>Port</b>	Integer	Portnummer an dem das IO-Link Device betrieben wird
<b>StartBlob</b>	Bool	Positive Flanke: Blob-Transfer starten
<b>BlobData</b>	Array of Byte	Ausgelesene Daten
<b>Error</b>	Bool	Fehlerstatus (0: kein Fehler)
<b>Ready</b>	Bool	Ready-Status (1: erfolgreich abgeschlossener Blob-Transfer)
<b>IOL_Status</b>	DWord	IO-Link Fehlerstatus des IO_LINK_DEVICE Bausteins

Tbl - 2

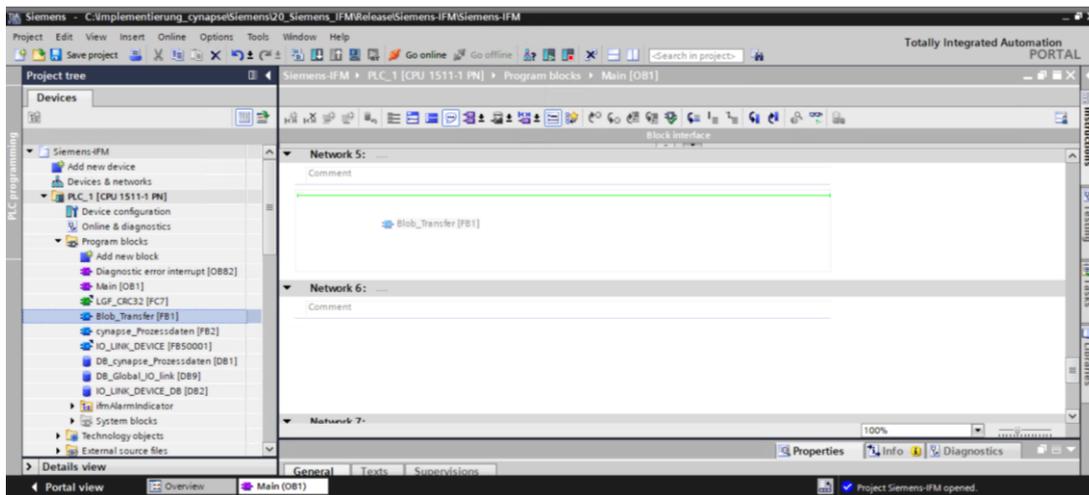
### Vorgehen

1. Öffnen Sie das erhaltene Code Beispielprojekt.
2. Öffnen Sie parallel das Projekt, in welchem Sie Blob-Daten lesen möchten.
3. Markieren Sie den Funktionsblock Blob-Transfer und die Funktion LGF\_CRC32 im Beispielprojekt
- ① **Hinweis:** Wichtig: LGF\_CRC32 aus Beispielprojekt ist leicht modifiziert. Wird dieser nicht genutzt, muss im Ursprungsbaustein die Länge des Arrays eingebunden werden.
4. Ziehen Sie diese per Drag and Drop unter „Program blocks“ in Ihr Projekt.

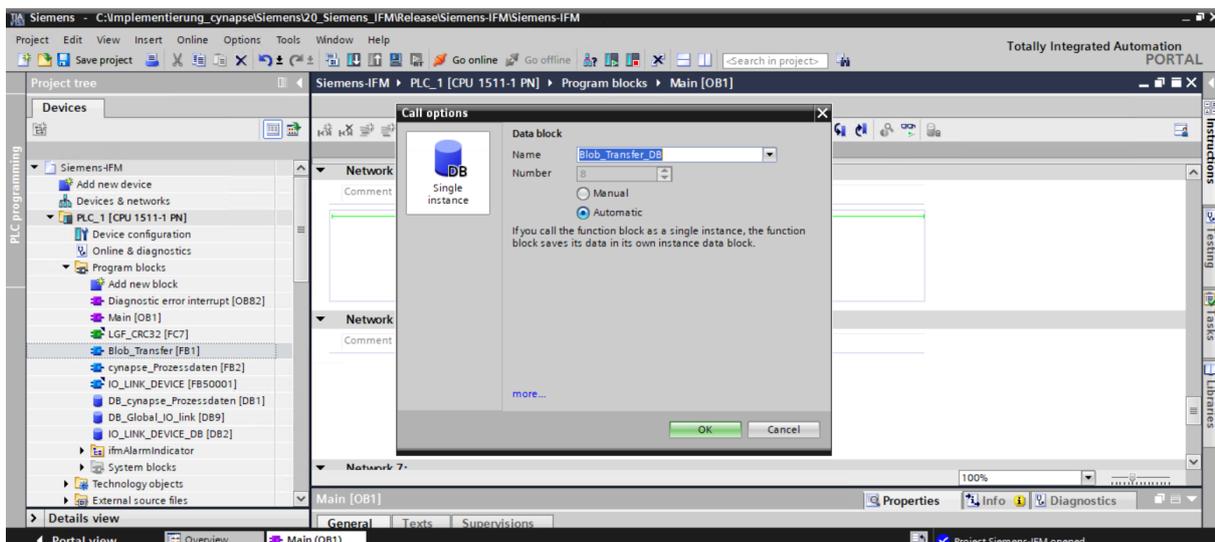




5. Ziehen Sie den FB per Drag and Drop in den Main-Baustein.

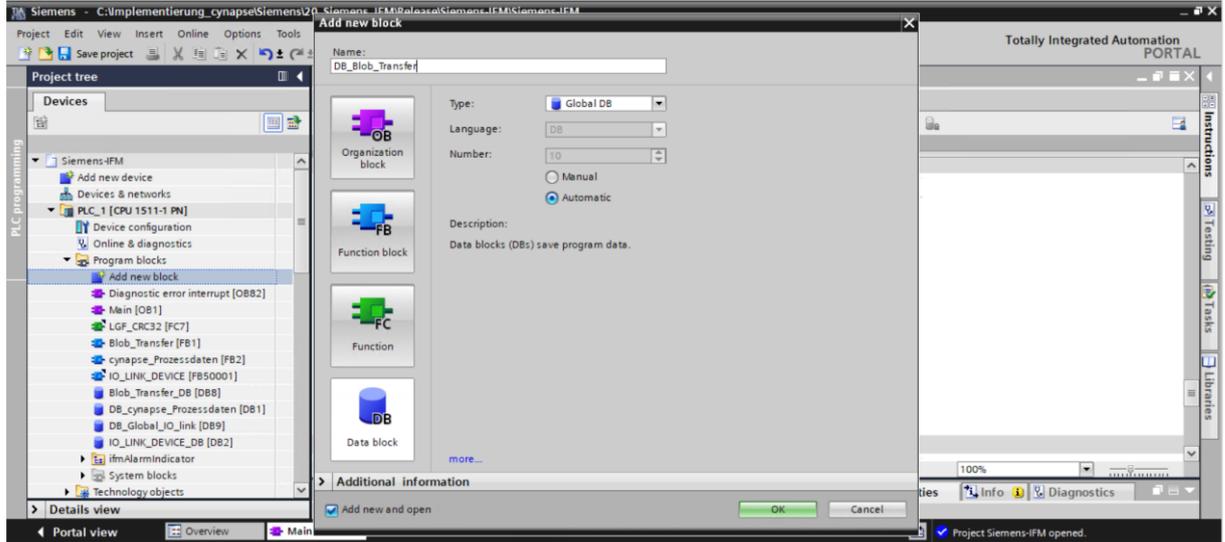


6. Legen Sie den DB durch Klick auf „OK“ an.

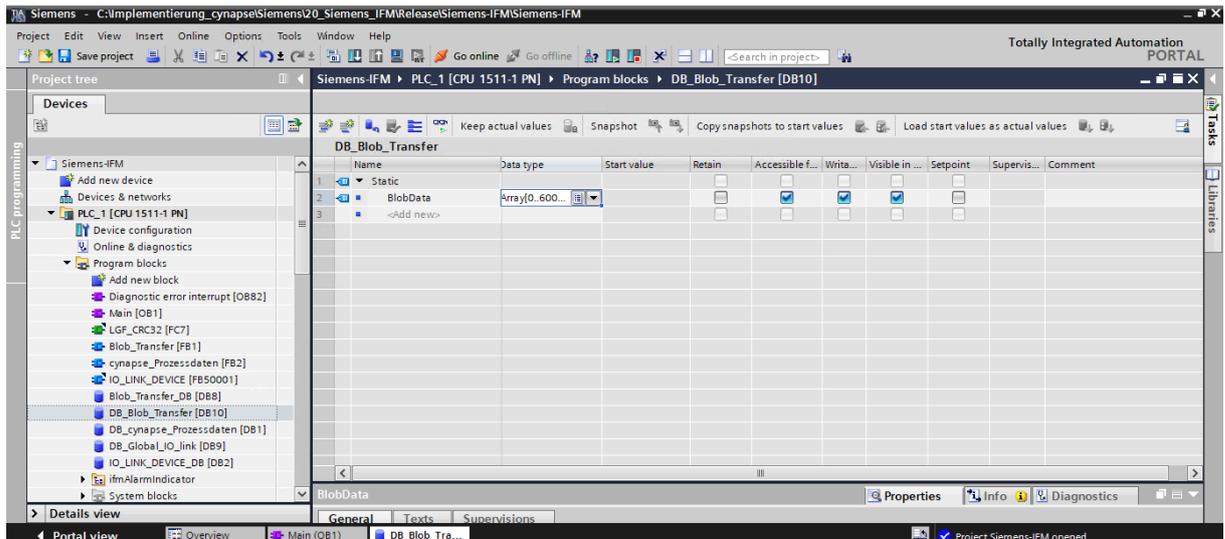


7. Der Baustein ist im Netzwerk in dem Main-Baustein enthalten.

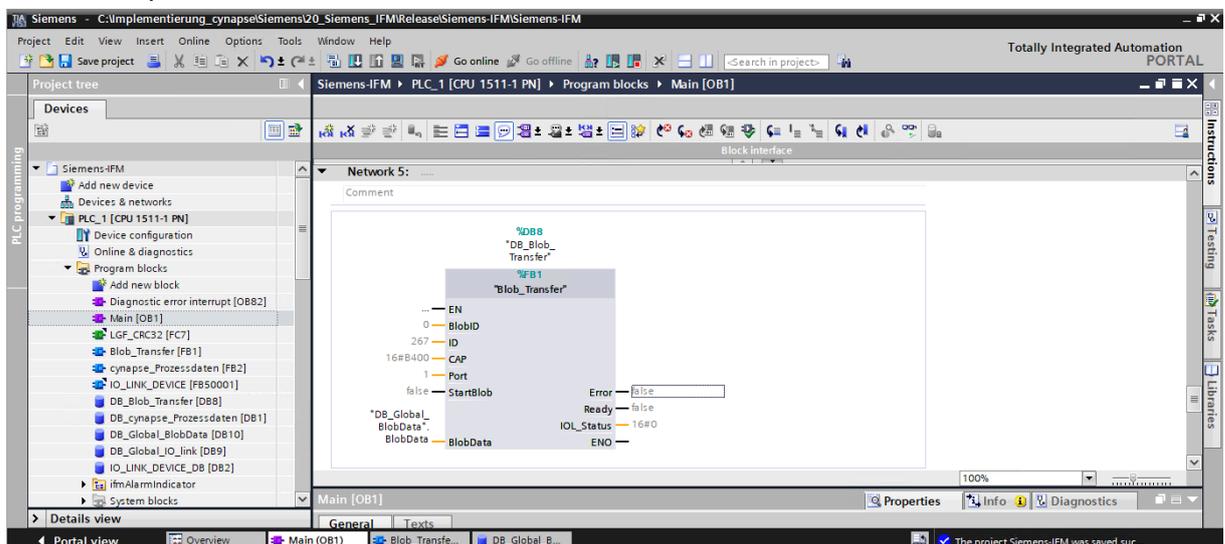
8. Um diesen betreiben zu können muss in einem globalen Datenbaustein ein Wertebereich für die ausgelesenen BLOB-Daten reserviert werden.
  - Doppelklick auf „Add new block“
  - Globalen Datenbaustein anlegen.



- Array in der Größe von 0-600000 Byte anlegen.



### 9. Verknüpfen Sie den Baustein.



## 10. Setzen Sie projektspezifische Eingangsgrößen:

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Input								
2	BlobID	Int	0						
3	ID	HW_IO	267						
4	CAP	Int	16#B400						
5	Port	Int	1						
6	StartBlob	Bool	false						
7	Output								
8	Error	Bool	false						
9	Ready	Bool	false						
10	IOL_Status	DWord	16#0						
11	InOut								
12	BlobData	Array[*] of Byte							
13	Static								
14	Start	Bool	false						
15	Blob_Length	Dint	0						
16	iStep	Int	10						
17	REQ	Bool	false						
18	RD_WR	Bool	false						

- ID: Hardwareerkennung des IO-Link Kommunikationsmoduls: Diese Information ist in den Systemkonstanten in der Hardwaresicht zu finden = 267

11. Laden Sie das Projekt auf die Hardware und gehen Sie online.

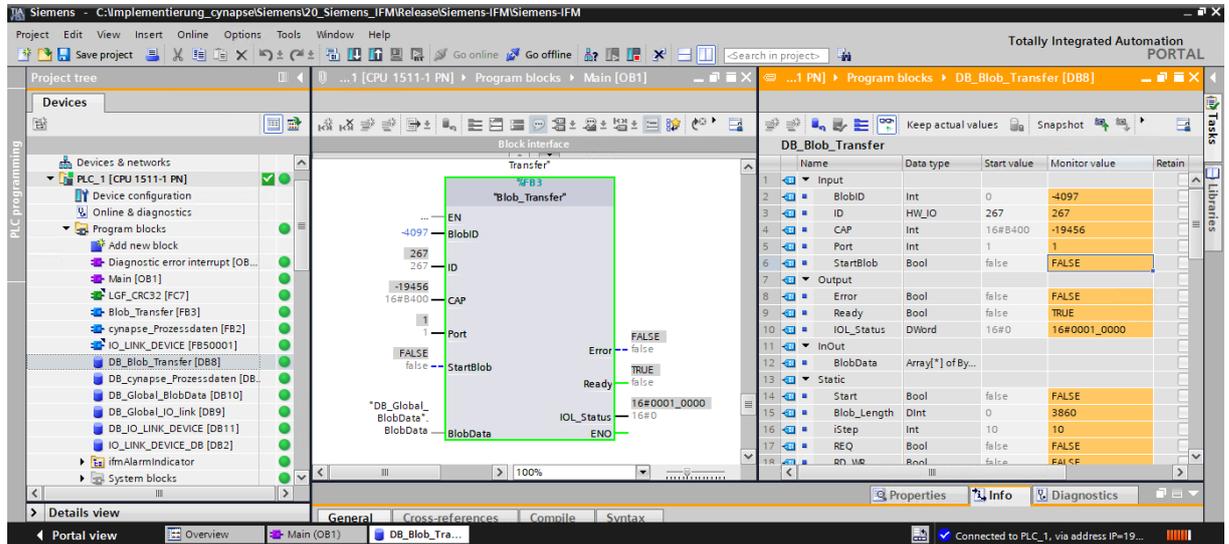
12. Geben Sie die BlobID vor.

13. Stoßen Sie den Blob-Transfer durch den Starttrigger an.

Name	Data type	Start value	Monitor value	Retain
1	Input			
2	BlobID	Int	0	
3	ID	HW_IO	267	
4	CAP	Int	16#B400	
5	Port	Int	1	
6	StartBlob	Bool	false	
7	Output			
8	Error	Bool	false	
9	Ready	Bool	false	
10	IOL_Status	DWord	16#0	
11	InOut			
12	BlobData	Array[*] of By...		
13	Static			
14	Start	Bool	false	
15	Blob_Length	Dint	0	
16	iStep	Int	10	
17	REQ	Bool	false	
18	RD_WR	Bool	false	

## Ergebnis

- ➔ Durch den Ausgang Ready = true zeigt der Baustein einen erfolgreich durchgeführten Blob-Transfer an.
- ➔ Die ausgelesenen Daten befinden sich im mit dem Ausgang BlobData verknüpften Array und können weiterverwendet werden.



Name	Data type	Start value	Monitor value	Retain
1	Input			
2	BlobID	Int	0	-4097
3	ID	HWLJO	267	267
4	CAP	Int	16#B400	19456
5	Port	Int	1	1
6	StartBlob	Bool	false	FALSE
7	Output			
8	Error	Bool	false	FALSE
9	Ready	Bool	false	TRUE
10	IOL_Status	DWord	16#0	16#0001_0000
11	InOut			
12	BlobData	Array[*] of By...		
13	Static			
14	Start	Bool	false	FALSE
15	Blob_Length	Dint	0	3860
16	iStep	Int	10	10
17	REQ	Bool	false	FALSE
18	RD_WR	Bool	false	FALSE



## Revisionshistorie

Revision	Datum	Kommentar	Kapitel
01	18.11.2019	Neuerstellung	Alle
02	15.07.2022	cynapse® Trademark, Überarbeitung	Alle
03	20.06.2023	Übersetzung EN	Alle



alpha

WITTENSTEIN alpha GmbH · Walter-Wittenstein-Straße 1 · 97999 Igersheim · Germany  
Tel. +49 7931 493-12900 · [info@wittenstein.de](mailto:info@wittenstein.de)

**WITTENSTEIN – eins sein mit der Zukunft**  
[www.wittenstein-alpha.de](http://www.wittenstein-alpha.de)