

LB/FB-PROFINET-Gateway

CFC für STEP 7

Handbuch



Your automation, our passion.

 **PEPPERL+FUCHS**

Es gelten die Allgemeinen Lieferbedingungen für Erzeugnisse und Leistungen der Elektroindustrie, herausgegeben vom Zentralverband Elektroindustrie (ZVEI) e. V. in ihrer neuesten Fassung sowie die Ergänzungsklausel: "Erweiterter Eigentumsvorbehalt".

Weltweit

Pepperl+Fuchs-Gruppe

Lilienthalstr. 200

68307 Mannheim

Deutschland

Telefon: +49 621 776 - 0

E-Mail: info@de.pepperl-fuchs.com

<https://www.pepperl-fuchs.com>

1	Version	5
2	Grundlagen.....	6
2.1	Allgemeines.....	6
2.1.1	Konzept Treiberbausteine	6
2.1.2	PCS 7-Version	6
2.1.3	GSDML.....	6
2.1.4	Bausteinnummern	6
2.2	Funktion der Bausteine	7
2.2.1	Ersatzwerte.....	7
2.2.2	Quality Codes	7
2.2.3	Skalierung.....	7
2.2.4	HART-Nebenvariablen	8
2.2.5	Simulation.....	8
3	Modulbaustein - RIOMOD	9
3.1	Beschreibung von RIOMOD	9
3.2	Anschlüsse von RIOMOD.....	10
3.3	Meldetexte und Begleitwerte von RIOMOD	11
4	Kanalbausteine	13
4.1	Hinweise zum Einsatz der Kanalbausteine	13
4.2	RIOAnIn - Analog-Eingangskanalbaustein.....	14
4.2.1	Beschreibung/Funktionen von RIOAnIn	14
4.2.2	Betriebsarten von RIOAnIn	15
4.2.3	Fehlerbehandlung von RIOAnIn.....	15
4.2.4	Melden von RIOAnIn.....	15
4.2.5	Anschlüsse von RIOAnIn	16
4.3	RIOAnOu – Analog-Ausgangskanalbaustein	18
4.3.1	Beschreibung/Funktionen von RIOAnOu	18
4.3.2	Betriebsarten von RIOAnOu	19
4.3.3	Fehlerbehandlung von RIOAnOu	19
4.3.4	Melden von RIOAnOu.....	19
4.3.5	Anschlüsse von RIOAnOu	20
4.4	RIOCount - Analog-Eingangskanalbaustein (Zähler Module).....	22
4.4.1	Beschreibung/Funktionen von RIOCount.....	22
4.4.2	Betriebsarten von RIOCount.....	23
4.4.3	Fehlerbehandlung von RIOCount	23
4.4.4	Melden von RIOCount	23
4.4.5	Anschlüsse von RIOCount.....	24
4.5	RIODiIn – Digital-Eingangskanalbaustein	25
4.5.1	Beschreibung/Funktionen von RIODiIn.....	25
4.5.2	Betriebsarten von RIODiIn	25
4.5.3	Fehlerbehandlung von RIODiIn.....	25
4.5.4	Melden von RIODiIn	25
4.5.5	Anschlüsse von RIODiIn	26

4.6	RIODiOu – Digital-Ausgangskanalbaustein	27
4.6.1	Beschreibung/Funktionen von RIODiOu	27
4.6.2	Betriebsarten von RIODiOu	27
4.6.3	Fehlerbehandlung von RIODiOu	27
4.6.4	Melden von RIODiOu	27
4.6.5	Anschlüsse von RIODiOu	28
4.7	Fehlerbehandlung	29
5	Projektierung.....	31
5.1	HW Konfig	31
5.2	CFC.....	33

1 Version

Dokumentenversion	Änderungen	Siehe
DOCT-6416	Erste Version	-
DOCT-6416A	PCS7-V9.1-Version ergänzt	Kapitel 2.1.2
	GSDML-Version geändert	Kapitel 2.1.3

2 Grundlagen

Dieses Kapitel enthält Informationen zu folgenden Themen:

- Allgemeines
- Funktionen der Bausteine



Hinweis!

Für die Projektierung werden fundierte Kenntnisse im Umgang mit der Siemens SIMATIC PCS 7 vorausgesetzt. Nähere Informationen zu SIMATIC PCS 7 finden Sie in der zugehörigen Onlinehilfe bzw. Dokumentation.

2.1 Allgemeines

Dieser Abschnitt enthält Informationen zu folgenden Themen:

- Konzept Treiberbaustein
- PCS 7-Version
- GSDML
- Bausteinnummern

2.1.1 Konzept Treiberbausteine

Das Bausteinkonzept basiert auf dem PCS 7-Standard. Hierdurch muss am Kanalbaustein nur die logische Adresse des Prozesswerts des Submoduls am Eingang "PV_In" bzw. am Ausgang "PV_Out" verschaltet werden. Beim Kompilieren der CFC-Pläne mit der Option "Baugruppentreiber erzeugen" fügt der Treiberwizard auf Basis der Hardwarekonfiguration automatisch alle weiteren Blöcke ein und verschaltet diese entsprechend.

Diagnosemeldungen werden über das PCS 7-Meldesystem gemeldet. Die zyklischen Diagnosedaten der Submodule werden als Statuswort an den verschalteten Kanalbausteinen zur Verfügung gestellt.

2.1.2 PCS 7-Version

Die Bausteinbibliotheken wurden für **PCS 7 V9.0+SP1** und **PCS 7 V9.1** erstellt und sind nur mit diesen Versionen kompatibel.

2.1.3 GSDML

Die Bibliotheken wurden für die GSDML-Datei "**GSDML-V2.33-PF-RIO-LBFB-20200825.xml**" entwickelt. Ein fehlerfreier Betrieb mit einer anderen GSDML-Version kann nicht gewährleistet werden.

2.1.4 Bausteinnummern

Es werden folgende Bausteinnummern verwendet:

Baustein	FB-Nummer
RIOMOD	FB 300
RIODiIn	FB 301
RIODiOu	FB 302
RIOAnIn	FB 303
RIOAnOu	FB 304
RIOCount	FB 305

2.2 Funktion der Bausteine

Dieser Abschnitt enthält Informationen zu folgenden Themen:

- Ersatzwerte
- Quality Codes
- Skalierung
- HART-Nebenvariablen
- Simulation

2.2.1 Ersatzwerte

Einige Kanalbausteine bieten die Möglichkeit im Fehlerfall Ersatzwerte an den Prozesswertausgängen auszugeben. Diese Funktionalität ist nicht für alle Kanalbausteine gleich und ist in den jeweiligen Kapiteln genauer beschrieben.

2.2.2 Quality Codes

Die Kanalbausteine stellen an vielen ihrer Ausgänge, neben dem eigentlichen Wert, einen Quality Code zur Verfügung. Dieser gibt den Status des Wertes an. Es werden folgende drei Quality Codes verwendet:

Quality Code	Bedeutung	Verwendung
16#80	Gültiger Wert	Es liegt kein Fehler vor und der Baustein befindet sich nicht im Simulationsbetrieb.
16#60	Simulierter Wert	Der Baustein befindet sich im Simulationsbetrieb.
16#00	Ungültiger Wert	Es liegt ein Fehler vor (Ausgang "Bad" = true) und der Baustein befindet sich nicht im Simulationsbetrieb. 16#00 wird ebenfalls für nicht verfügbare HART-Ausgänge ausgegeben.

2.2.3 Skalierung

Die Skalierung der analogen Prozesswerte erfolgt in den Modulen selbst und die Prozesswerte werden bereits als Gleitkommazahl übertragen. Die Parametrierung kann über die Parameter der Submodule in HW Konfig vorgenommen werden.

2.2.4 HART-Nebenvariablen

Der Modulbaustein RIOMOD erkennt über die Submodul ID, ob ein HART-Submodul verwendet wird. In diesem Fall werden neben den Informationen des analogen Ein-/Ausgangs-Submoduls, auch die Submodul ID und die logische Eingangsadresse des HART-Submoduls über die Struktur "RIO_Conn" an den verschalteten Kanalbaustein weitergegeben.

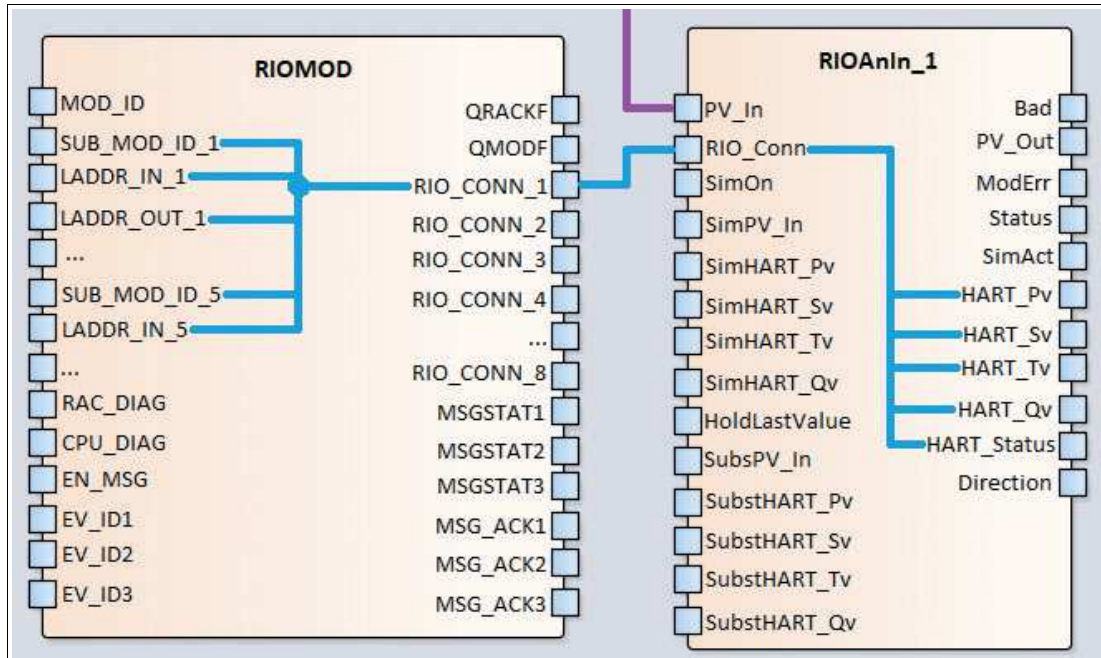


Abbildung 2.1 RIOMOD - Interne Informationsverteilung für HART-Module

Im Kanalbaustein werden die HART-Nebenvariablen incl. Statusbyte gelesen und an den entsprechenden Ausgängen des zugehörigen analogen Kanalbausteins ausgegeben. Falls ein HART-Submodul weniger als vier Nebenvariablen liefert, dann wird der Quality Code der nicht verfügbaren HART-Ausgänge auf 16#0 "Invalid Value" gesetzt.

Im Simulationsbetrieb werden die Eingänge "SimHART_Pv", "SimHART_Sv", usw. an die jeweiligen Ausgänge "HART_PV", "HART_Sv", usw. durchgereicht. Der Quality Code der Ausgänge wird auf 16#60 "Simulation" gesetzt.

Falls der Eingang "HoldLastValue" von RIOAnIn = "false" oder der Eingang "HlvHART" von RIOAnOu = "false" ist, werden im Fehlerfall die Ersatzwerte "SubstHART_Pv", "SubstHART_Sv", etc. an die jeweiligen Ausgänge "HART_PV", "HART_Sv", etc. durchgereicht. Der Quality Code wird auf 16#0 "Invalid Value" gesetzt.

Simulation und Ersatzwertmodus prüfen nicht, wie viele Nebenvariablen das verschaltete Submodul liefert und reichen jeweils die Werte alle vier Eingänge an die, die entsprechenden Ausgänge durch.

2.2.5 Simulation

Der Simulationsbetrieb wird von allen Kanalbausteinen unterstützt.

Im Simulationsbetrieb ("SimOn" = 1) werden die simulierten Werte wie z.B. "SimPV_In" direkt an die zugehörigen Ausgänge z.B. "PV_Out" durchgereicht. Die Ausgänge "Status", "Bad", "ModErr" und "HART_Status" werden im Simulationsbetrieb auf 0 gesetzt. Der Ausgang "SimAct" wird auf 1 gesetzt.

Der Quality Code aller Ausgänge wird auf 16#60 "Simulation" gesetzt. Das Simulationsverhalten der HART-Nebenvariablen ist in Kapitel 2.2.4 beschrieben.

3 Modulbaustein - RIOMOD

Dieses Kapitel enthält Informationen zu folgenden Themen:

- Beschreibung von RIOMOD
- Anschlüsse von RIOMOD
- Meldetexte und Begleitwerte von RIOMOD

3.1 Beschreibung von RIOMOD

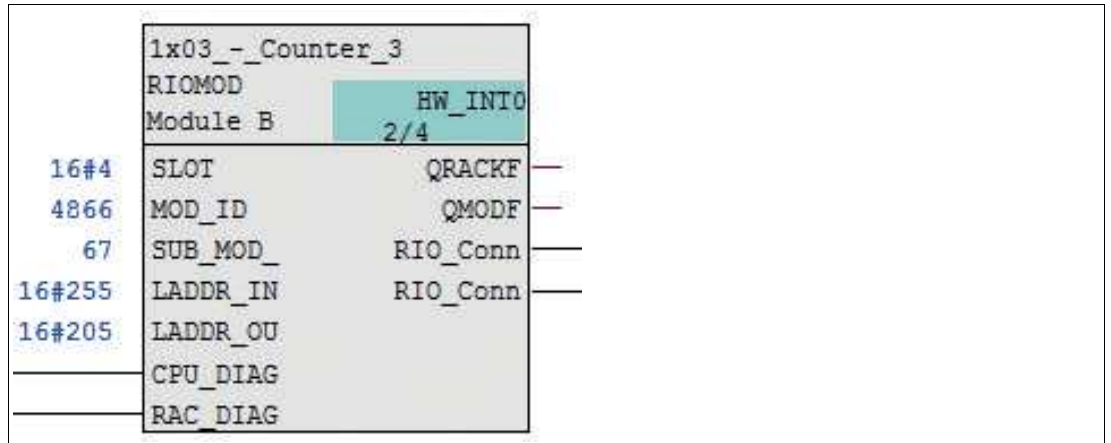


Abbildung 3.1 Modulbaustein RIOMOD

Für jedes in HW Konfig projektierte Modul wird durch den Treiberwizard eine eigene Instanz von RIOMOD eingebaut unabhängig vom Typ des Moduls. Dabei werden alle relevanten Informationen des Moduls und aller zugehörigen Submodule an die Eingänge des Bausteins geschrieben. Diese werden unter Berücksichtigung der Modul ID und der Submodul ID über die Strukturausgänge "RIO_CONN_1" bis "RIO_CONN_8" an die entsprechenden Kanalbausteine weitergeleitet.

3.2 Anschlüsse von RIOMOD

Eingänge von RIOMOD

Eingang	Typ	Sichtbar	Beschreibung
EN	BOOL	X	Enable
SLOT	INT	X	Einbauposition des Moduls
MOD_ID	DWORD	X	Modul ID
SUB_MOD_ID_1	DWORD	Falls vorhanden	Submodul ID - Submodul 1
LADDR_IN_1	DWORD	Falls vorhanden	Logische Eingangsadresse - Submodul 1
LADDR_OUT_1	DWORD	Falls vorhanden	Logische Ausgangsadresse - Submodul 1
...			
SUB_MOD_ID_8	DWORD	Falls vorhanden	Submodul ID - Submodul 8
LADDR_IN_8	DWORD	Falls vorhanden	Logische Eingangsadresse - Submodul 8
LADDR_OUT_8	DWORD	Falls vorhanden	Logische Ausgangsadresse - Submodul 8
EV_ID1	DWORD	-	Event ID 1
EV_ID2	DWORD	-	Event ID 2
EV_ID3	DWORD	-	Event ID 3
EV_ID4	DWORD	-	Event ID 4
EN_MSG	BOOL	-	1 = Enable Alarm

Ausgänge von RIOMOD

Ausgang	Typ	Sichtbar	Beschreibung
ENO	BOOL	-	Enable Out
QRACKF	BOOL	X	1 = Übergeordneter Fehler
QMODF	BOOL	X	1 = Modul gezogen/defekt
MSGSTAT1	WORD	-	Meldungsfehler-Information
MSGSTAT2	WORD	-	Meldungsfehler-Information
MSGSTAT3	WORD	-	Meldungsfehler-Information
MSGSTAT4	WORD	-	Meldungsfehler-Information
MSG_ACK1	WORD	-	Meldungsquittierung
MSG_ACK2	WORD	-	Meldungsquittierung
MSG_ACK3	WORD	-	Meldungsquittierung
MSG_ACK4	WORD	-	Meldungsquittierung
RIO_Conn_1	STRUCT	Falls verschaltet	Struktur zur Übertragung der relevanten Daten an Kanalbaustein - Submodul 1
...			
RIO_Conn_8	STRUCT	Falls verschaltet	Struktur zur Übertragung der relevanten Daten an Kanalbaustein - Submodul 8

IN_OUTs von RIOMOD

In/Out	Typ	Sichtbar	Beschreibung
CPU_DIAG	STRUCT	X	CPU Diagnose - Wird verwendet, um auf Interrupts zu reagieren
RAC_DIAG	STRUCT	X	Rack Diagnose - Auswertung von Rack Fehlern

3.3 Meldetexte und Begleitwerte von RIOMOD

RIOMOD meldet relevante Meldungen der Modul- und Kanaldiagnose an das PCS 7-Meldesystem. Die Rack Diagnose wird von "OB_DIAG1_PN" bearbeitet.

Es ist eine Alarmunterdrückung implementiert, die sicherstellt, dass keine unterlagerten Fehler im Meldesystem gemeldet werden, sobald ein überlagerter Fehler anliegt. Bei einem Rackfehler werden z.B. keine Modulfehler gemeldet und bei einem Modulfehler werden keine Kanalfehler des Moduls gemeldet. Außerdem werden alle unterlagerten Meldungen auf "gehend" gesetzt, sobald ein übergeordneter Fehler auftritt.

Es sind folgende Meldungen in Deutsch und in Englisch angelegt:

Meldebau- stein ALARM_8P	Meldungs- nummer	Vorbereitungsmeldetext	Beschreibung
EV_ID1	SIG_1	Englisch: Module @1%d@/@2%d@/@3%d@ Module pulled Deutsch: Modul @1%d@/@2%d@/@3%d@ Modul gezogen	Modul gezogen Typ: AS-Leittechnik Mel- dung - Störung
EV_ID2	SIG_1	Englisch: Module @1%d@/@2%d@/@3%d@: Open Wire at Channel 1 Deutsch: Modul @1%d@/@2%d@/@3%d@: Draht- bruch an Kanal 1	Drahtbruch an Kanal 1. Typ: AS-Leittechnik Mel- dung - Störung
EV_ID2
EV_ID2	SIG_8	Englisch: Module @1%d@/@2%d@/@3%d@: Open Wire at Channel 8 Deutsch: Modul @1%d@/@2%d@/@3%d@: Draht- bruch an Kanal 8	Drahtbruch an Kanal 8. Typ: AS-Leittechnik Mel- dung - Störung
EV_ID3	SIG_1	Englisch: Module @1%d@/@2%d@/@3%d@: Short Cir- cuit at Channel 1 Deutsch: Modul @1%d@/@2%d@/@3%d@: Kurz- schluss an Kanal 1	Kurzschluss an Kanal 1. Typ: AS-Leittechnik Mel- dung - Störung
EV_ID3
EV_ID3	SIG_8	Englisch: Module @1%d@/@2%d@/@3%d@: Short Cir- cuit at Channel 8 Deutsch: Modul @1%d@/@2%d@/@3%d@: Kurz- schluss an Kanal 8	Kurzschluss an Kanal 8. Typ: AS-Leittechnik Mel- dung - Störung
EV_ID4	SIG_1	Englisch: Module @1%d@/@2%d@/@3%d@: Line Fault at Channel 1 Deutsch: Modul @1%d@/@2%d@/@3%d@: Leitungs- fehler an Kanal 1	Line Fault an Kanal 1. Wird nur gemeldet, wenn NICHT Kurzschluss und NICHT Unterbrechung am selben Kanal. Typ: AS-Leittechnik Mel- dung - Störung

Meldebau- stein ALARM_8P	Meldungs- nummer	Vorbesetzungsmeldetext	Beschreibung
EV_ID4
EV_ID4	SIG_8	Englisch: Module @1%d@/@2%d@/@3%d@: Line Fault at Channel 8	Line Fault an Kanal 8. wird nur gemeldet, wenn NICHT Kurschluss und NICHT Unterbrechung am selben Kanal.

Beschreibung der Begleitwerte

Nummer	Bedeutung	Datentyp
@1%d@	ID des PN IO Systems	BYTE
@2%d@	Rack Nummer	BYTE
@3%d@	Slot Nummer	BYTE

4 Kanalbausteine

4.1 Hinweise zum Einsatz der Kanalbausteine

Die Kanalbausteine orientieren sich am Siemens APL-Standard und bieten Funktionalitäten wie beispielsweise Simulation, Ersatzwerte und Quality Codes. Diese sind näher in den Kapiteln der einzelnen Kanalbausteine beschrieben.

Die Bibliothek umfasst fünf Kanalbausteine, die den Einsatz aller verfügbaren Submodule des Gateways ermöglichen.

- RIODiIn: Digitale Eingänge
- RIODiOu: Digitale Ausgänge
- RIOAnIn: Analoge Eingänge
- RIOAnOu: Analoge Ausgänge
- RIOCount: Zähler

Die folgende Tabelle listet alle in der GSDML vorhandenen Submodule auf und verweist auf den zugehörigen Kanalbaustein.

Typ	Submodul ID GSDML	I/O in Byte	Zugehöriger Kanalbaustein
Digital Input (Submodule unterscheiden sich durch Parameter)	0x01 0x02 0x03 0x04	1/0	RIODiIn
Digital Output (Submodule unterscheiden sich durch Parameter und das Fehlerbit "Linefault")	0x011 0x012 0x013	1/1	RIODiOu
Analog Input	0x21 0x22 (AI mit HART-Option)	6/0	RIOAnIn
Analog Output	0x31 0x32 (AO mit HART-Option)	5/4	RIOAnOu
Counter Input Frequency Input Frequency/Counter Input	0x41 0x42 0x43	6/5 7/0 11/5	RIOCount RIOAnIn RIOCount + RIOAnIn
RTD/Resist. Input predefined (5x04) TC/Voltage Input predefined (5x02, 5x05) Voltage Input predefined (5x06) RTD Input user defined (5x04) Resist. Input user defined (5x01, 5x04) RTD/Resist. Input predefined (5x01) RTD Input user defined (5x01) TC Input user defined (5x02, 5x05) Voltage Input user defined mV (5x02, 5x05) Voltage Input user defined (5x06)	0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x58 0x59 0x5A	6/0	RIOAnIn
HART Dyn Var Pv HART Dyn Var PvSv HART Dyn Var PvSvTv HART Dyn Var PvSvTvQv HART Dyn Var Sv HART Dyn Var SvTv HART Dyn Var SvTvQv	0x61 0x62 0x63 0x64 0x65 0x66 0x67	5/0 9/0 13/0 17/0 5/0 9/0 13/0	Kanalbaustein des zugehörigen AI/AO Siehe 2.2.4

4.2 RIOAnIn - Analog-Eingangskanalbaustein

Dieser Abschnitt enthält Informationen zu folgenden Themen:

- Beschreibung/Funktionen von RIOAnIn
- Betriebsarten von RIOAnIn
- Fehlerbehandlung von RIOAnIn
- Melden von RIOAnIn
- Anschlüsse von RIOAnIn

4.2.1 Beschreibung/Funktionen von RIOAnIn

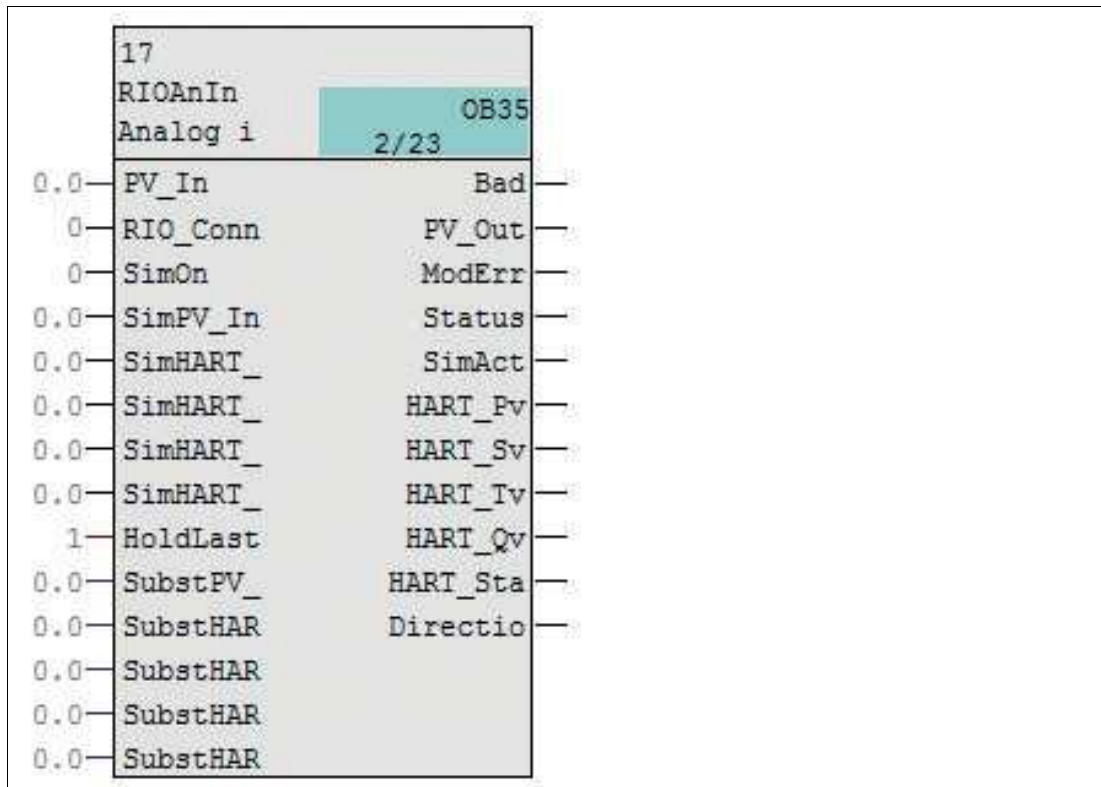


Abbildung 4.1 Kanalbaustein RIOAnIn

Zur Projektierung muss der Eingang PV_In von RIOAnIn auf den Prozesswert des gewünschten Submoduls verschaltet werden. Hierfür muss zuvor ein Symbol in der Symboltabelle vom Typ REAL angelegt werden und mit der Eingangsadresse des Prozesswerts (DWORD) verknüpft werden. Die Projektierung der Kanalbausteine ist genauer in Kapitel 5 beschrieben.

Die hier aufgeführten Funktionen sind im Detail in Kapitel 2.2 beschrieben.

Der Kanalbaustein RIOAnIn liest das verschaltete Doppelwort vom Typ Real. Falls kein Fehler vorliegt, wird der gelesene Prozesswert an "PV_Out" ausgegeben. Das Diagnosewort des verschalteten Moduls wird ebenfalls zyklisch gelesen und wie in Kapitel 5.7 beschrieben am Ausgang "Status" ausgegeben. Falls ein Fehler auftritt, wird der Ausgang "Bad" auf 1 gesetzt, was bedeutet, dass der Prozesswert ungültig ist. Falls ein Rack- oder Modulfehler vorliegt, werden "ModErr" und "Bad" auf 1 gesetzt.

Der Kanalbaustein bietet die Option "HoldLastValue", die über den gleichnamigen Eingang aktiviert/deaktiviert werden kann. Falls "HoldLastValue" aktiv ist ("HoldLastValue" = 1), wird im Fehlerfall (Bad = 1) der letzte gültige Prozesswert an "PV_Out" gehalten. Falls "HoldLastValue" inaktiv ist ("HoldLastValue" = 0), werden im Fehlerfall die an "SubstPV_In", "SubstHART_PV", usw. angegebenen Wert an "PV_Out", "HART_PV", usw. ausgegeben.

Im Simulationsbetrieb ("SimOn" = 1) wird der an "SimPV_In" angegebene Wert an "PV_Out" ausgegeben. Die Ausgänge "Status", "Bad", "ModErr" und "HART_Status" werden im Simulationsbetrieb auf 0 gesetzt. Der Ausgang "SimAct" wird auf 1 gesetzt. Die Eingänge "SimHART_Pv", "SimHART_Sv", etc. werden an die jeweiligen Ausgänge "HART_PV", "HART_Sv", etc. durchgereicht.

Falls RIOAnIn in Kombination mit einem HART-Submodul verwendet wird, bekommt der Kanalbaustein die logische Eingangsadresse des HART-Submoduls und dessen Submodul-ID über die Struktur "RIO_Conn" mitgeteilt (Siehe 2.2.4). Mit Hilfe dieser Informationen können die HART-Nebenvariablen incl. Statusbyte gelesen und an den entsprechenden Ausgängen ausgegeben werden.

Die Skalierung des Prozesswerts findet bereits im Submodul selbst statt.

Der Ausgang "Direction" ist nur für das Frequenzmodul gültig und gibt das gleichnamige zyklische Eingangsbit des Submoduls aus. Für alle anderen Submodule ist der Ausgang immer 0.

4.2.2 Betriebsarten von RIOAnIn

Dieser Baustein verfügt über keine Betriebsarten.

4.2.3 Fehlerbehandlung von RIOAnIn

Die Fehlerbehandlung der Kanalbausteine ist in Kapitel 4.7 beschrieben.

4.2.4 Melden von RIOAnIn

Dieser Baustein hat kein Meldeverhalten.

4.2.5 Anschlüsse von RIOAnIn

Eingänge RIOAnIn

Eingang	Typ	Startwert	Sichtbar	Beschreibung
EN	BOOL	1	-	1 = Aufgerufener Baustein wird bearbeitet
PV_In	REAL	0.0	X	Verschaltung auf Prozesswert
RIO_Conn	STRUCT	-	X	Enthält Informationen über das Submodul. Verschaltung zu RIOMOD wird automatisch per Treiberwizard eingefügt.
SimOn	BOOL + ST	0 16#80	X	1 = Simulation eingeschaltet
SimPV_In	REAL + ST	0.0 16#80	-	Prozesswert, der bei SimOn = 1 verwendet wird
Sim-HART_Pv	REAL + ST	0.0 16#80	-	Wert, der bei SimOn = 1 an HART_Pv ausgegeben wird
Sim-HART_Sv	REAL + ST	0.0 16#80	-	Wert, der bei SimOn = 1 an HART_Sv ausgegeben wird
Sim-HART_Tv	REAL + ST	0.0 16#80	-	Wert, der bei SimOn = 1 an HART_Tv ausgegeben wird
Sim-HART_Qv	REAL + ST	0.0 16#80	-	Wert, der bei SimOn = 1 an HART_Qv ausgegeben wird
HoldLast-Value	BOOL	1	X	1 = Letzter Prozesswert wird im Fehlerfall gehalten 0 = Ersatzwert wird verwendet
SubsPV_In	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an PV_Out ausgegeben, falls HoldLastValue = 0 ist
Subs-HART_Pv	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an HART_Pv ausgegeben, falls HoldLastValue = 0 ist
Subs-HART_Sv	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an HART_Sv ausgegeben, falls HoldLastValue = 0 ist
Subs-HART_Tv	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an HART_Tv ausgegeben, falls HoldLastValue = 0 ist
Subs-HART_Qv	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an HART_Qv ausgegeben, falls HoldLastValue = 0 ist

Ausgänge RIOAnIn

Ausgang	Typ	Startwert	Sichtbar	Beschreibung
ENO	BOOL	0	-	1 = Bausteinalgorithmus ist fehlerfrei durchlaufen
Bad	BOOL + ST	0 16#80	X	1 = Prozesswert ist ungültig
PV_Out	REAL + ST	0.0 16#80	X	Prozesswert
ModErr	BOOL + ST	0 16#80	-	1 = Gerät / Baugruppe ist gestört
Status	WORD + ST	0 16#80	-	Status-Wort Siehe 4.7
SimAct	BOOL + ST	0 16#80	-	1 = Simulation aktiv
HART_Pv	REAL + ST	0.0 16#80	-	HART Nebenvariable Pv
HART_Sv	REAL + ST	0.0 16#80	-	HART Nebenvariable Sv
HART_Tv	REAL + ST	0.0 16#80	-	HART Nebenvariable Tv
HART_Qv	REAL + ST	0.0 16#80	-	HART Nebenvariable Qv
HART_Status	BYTE + ST	0 16#80	-	HART Status-Byte
Direction	BOOL + ST	0 16#80	-	Nur gültig für Frequenzsubmodul. Gibt das gleichnamige zyklische Eingangsbit aus.

4.3 RIOAnOu – Analog-Ausgangskanalbaustein

Dieser Abschnitt enthält Informationen zu folgenden Themen:

- Beschreibung/Funktionen von RIOAnOu
- Betriebsarten von RIOAnOu
- Fehlerbehandlung von RIOAnOu
- Melden von RIOAnOu
- Anschlüsse von RIOAnOu

4.3.1 Beschreibung/Funktionen von RIOAnOu

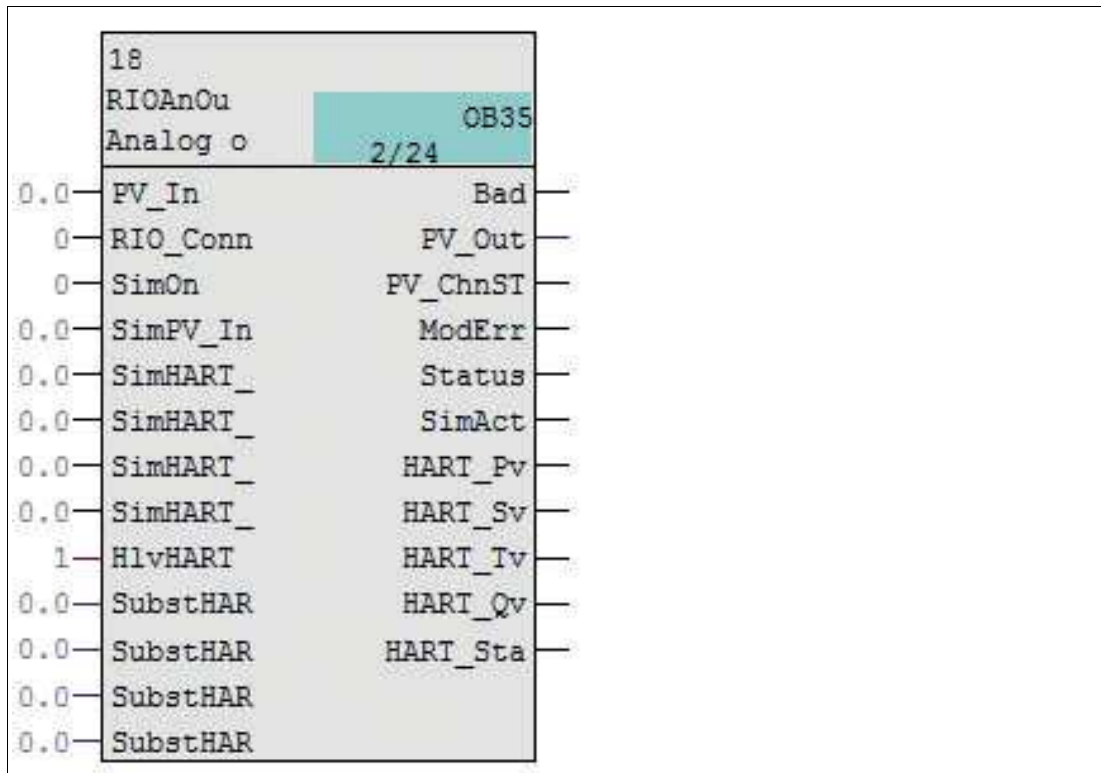


Abbildung 4.2 Kanalbaustein RIOAnOu

Zur Projektierung muss der Ausgang PV_Out von RIOAnOu auf den Prozesswert des gewünschten Submoduls verschaltet werden. Hierfür muss zuvor ein Symbol in der Symboltabelle vom Typ REAL angelegt werden und mit der Ausgangsadresse des Prozesswerts (DWORD) verknüpft werden. Die Projektierung der Kanalbausteine ist genauer in Kapitel 5 beschrieben. Die hier aufgeführten Funktionen sind im Detail in Kapitel 2.2 beschrieben.

Der Kanalbaustein RIOAnOu schreibt das an "PV_In" verschaltete Doppelwort an die über "PV_Out" verschaltete logische Ausgangsadresse. Über die Struktur "RIO_Conn" bekommt der Kanalbaustein die logische Eingangsadresse des zugehörigen Submoduls übergeben. Die Diagnoseinformationen des Submoduls werden zyklisch gelesen und die darin enthaltenen Informationen werden wie in Kapitel 4.7 beschrieben, am Ausgang "Status" ausgegeben. Falls "ModErr" = 1 ist, wird "Bad" ebenfalls auf 1 gesetzt.

Der Kanalbaustein bietet die Option "HoldLastValue" für HART-Variablen, die über den Eingang "HlvHART" aktiviert/deaktiviert werden kann. Falls "HlvHART" = 1 ist, werden im Fehlerfall (Bad = 1) die letzten gültigen Werte an den HART-Ausgängen gehalten. Falls "HlvHART" inaktiv ist, werden im Fehlerfall die an "SubstHART_Pv", "SubstHART_Sv", usw. angegebenen Wert an "HART_Pv", "HART_Sv", usw. ausgegeben.

Im Simulationsbetrieb ("SimOn" = 1) wird der an "SimPV_In" angegebene Wert an "PV_Out" ausgegeben. Die Ausgänge "Status", "Bad", "ModErr" und "HART_Status" werden im Simulationsbetrieb auf 0 gesetzt. Der Ausgang "SimAct" wird auf 1 gesetzt. Die Eingänge "SimHART_Pv", "SimHART_Sv", etc. werden an die jeweiligen Ausgänge "HART_PV", "HART_Sv", etc. durchgereicht.

Falls RIOAnOu in Kombination mit einem HART-Submodul verwendet wird, bekommt der Kanalbaustein die logische Eingangsadresse des HART-Submoduls und dessen Submodul-ID über die Struktur "RIO_Conn" mitgeteilt (Siehe 2.2.4). Mit Hilfe dieser Informationen können die HART-Nebenvariablen incl. Statusbyte gelesen und an den entsprechenden Ausgängen ausgegeben werden.

4.3.2 Betriebsarten von RIOAnOu

Dieser Baustein verfügt über keine Betriebsarten.

4.3.3 Fehlerbehandlung von RIOAnOu

Die Fehlerbehandlung der Kanalbausteine ist in Kapitel 4.7 beschrieben.

4.3.4 Melden von RIOAnOu

Dieser Baustein hat kein Meldeverhalten.

4.3.5 Anschlüsse von RIOAnOu

Eingänge RIOAnOu

Eingang	Typ	Startwert	Sichtbar	Beschreibung
EN	BOOL	1	-	1 = Aufgerufener Baustein wird bearbeitet
PV_In	REAL + ST	0.0 16#80	X	Eingangswert
RIO_Conn	STRUCT	-	X	Enthält Informationen über das Submodul. Verschaltung zu RIOMOD wird automatisch per Treiberwizard eingefügt.
SimOn	BOOL + ST	0 16#80	X	1 = Simulation eingeschaltet
SimPV_In	REAL + ST	0.0 16#80	-	Prozesswert, der bei SimOn = 1 verwendet wird
Sim-HART_Pv	REAL + ST	0.0 16#80	-	Wert, der bei SimOn = 1 an HART_Pv ausgegeben wird
Sim-HART_Sv	REAL + ST	0.0 16#80	-	Wert, der bei SimOn = 1 an HART_Sv ausgegeben wird
Sim-HART_Tv	REAL + ST	0.0 16#80	-	Wert, der bei SimOn = 1 an HART_Tv ausgegeben wird
Sim-HART_Qv	REAL + ST	0.0 16#80	-	Wert, der bei SimOn = 1 an HART_Qv ausgegeben wird
HlvHART	BOOL	1	-	1 = Letzte Werte der HART-Ausgänge werden im Fehlerfall gehalten. 0 = Ersatzwert wird verwendet
Subs-HART_Pv	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an HART_Pv ausgegeben, falls HlvHART = 0 ist
Subs-HART_Sv	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an HART_Sv ausgegeben, falls HlvHART = 0 ist
Subs-HART_Tv	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an HART_Tv ausgegeben, falls HlvHART = 0 ist
Subs-HART_Qv	REAL	0.0	-	Dieser Ersatzwert wird im Fehlerfall an HART_Qv ausgegeben, falls HlvHART = 0 ist

Ausgänge RIOAnOu

Ausgang	Typ	Startwert	Sichtbar	Beschreibung
ENO	BOOL	0	-	Enable Out
Bad	BOOL + ST	0 16#80	X	1 = Prozesswert ist ungültig
PV_Out	REAL	0.0	X	Verschaltung auf Prozesswert
ModErr	BOOL + ST	0 16#80	-	1 = Gerät / Baugruppe ist gestört
Status	WORD + ST	0 16#80	-	Status-Wort Siehe 4.7
PV_ChnST	REAL + ST	0.0 16#80	X	Signalstatus des Ausgangskanals und Wert des PV_Out
SimAct	BOOL + ST	0 16#80	-	1 = Simulation aktiv
HART_Pv	REAL + ST	0.0 16#80	-	HART Nebenvariable Pv
HART_Sv	REAL + ST	0.0 16#80	-	HART Nebenvariable Sv
HART_Tv	REAL + ST	0.0 16#80	-	HART Nebenvariable Tv
HART_Qv	REAL + ST	0.0 16#80	-	HART Nebenvariable Qv
HART_Status	BYTE + ST	0 16#80	-	HART Status-Byte

4.4 RIOCount - Analog-Eingangskanalbaustein (Zähler Module)

Dieses Kapitel enthält Informationen zu folgenden Themen:

- Beschreibung/Funktionen von RIOCount
- Betriebsarten von RIOCount
- Fehlerbehandlung von RIOCount
- Melden von RIOCount
- Anschlüsse von RIOCount

4.4.1 Beschreibung/Funktionen von RIOCount

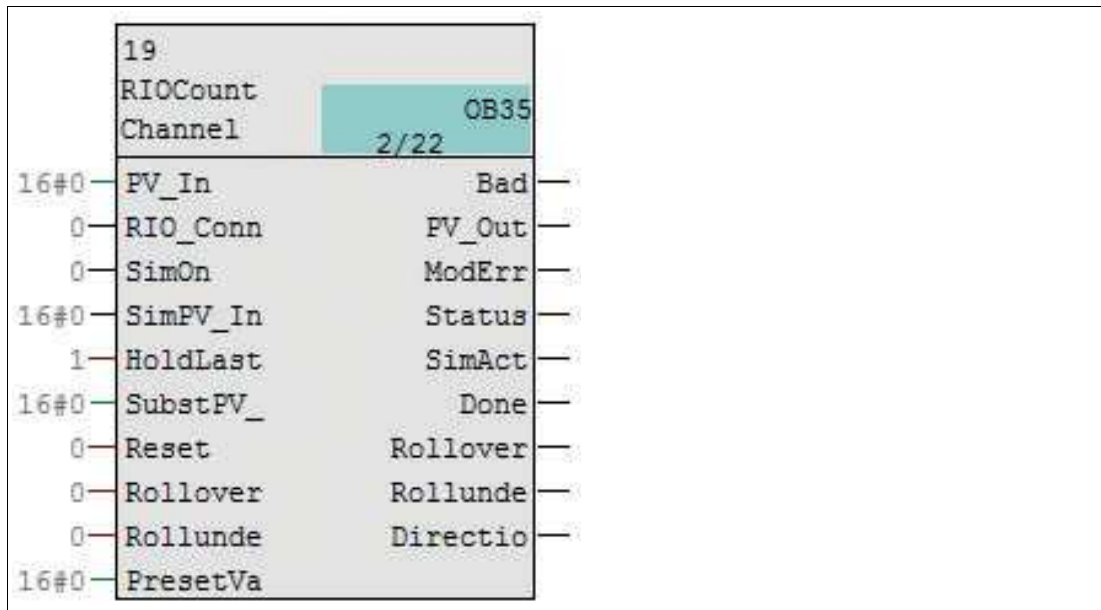


Abbildung 4.3 Kanalbaustein RIOCount

Zur Projektierung muss der Eingang PV_In von RIOCount auf den Zählwert des gewünschten Submoduls vom Typ DWORD verschaltet werden. Die Projektierung der Kanalbausteine ist genauer in Kapitel 5 beschrieben.

Die hier aufgeführten Funktionen sind im Detail in Kapitel 2.2 beschrieben.

Der Kanalbaustein RIOCount wird zum Lesen von Zählwerten verwendet und kommt bei den beiden Submodulen "Counter Input" (Submodul ID: 0x41) und "Frequency/Counter Input" (Submodul ID: 0x43) zum Einsatz. Wobei für das Modul "Frequency/Counter Input" zum Lesen der Frequenz zusätzlich ein RIOAnIn-Kanalbaustein benötigt wird.

Die Diagnose-Informationen des Submoduls werden zyklisch gelesen und die darin enthaltenen Informationen werden wie in Kapitel 4.7 beschrieben, am Ausgang "Status" ausgegeben. Falls "ModErr" = 1 ist, wird "Bad" ebenfalls auf 1 gesetzt.

Unabhängig vom verwendeten Submodul, muss die logische Eingangsadresse des Zählwertes auf PV_In verschaltet werden. Falls kein Fehler vorliegt, wird der gelesene Prozesswert an "PV_Out" ausgegeben.

Die Eingänge "Reset", "RolloverAck", "RollunderAck" und "PresetValue" werden zyklisch auf die zugehörige logische Ausgangsadresse geschrieben.

Die Ausgänge "Done", "Rollover", "Rollunder" und "Direction" werden zyklisch von den zugehörigen logischen Eingangsadressen gelesen. Im Fehlerfall wird der letzte gültige Wert dieser 4 Ausgänge gehalten.

Der Kanalbaustein bietet die Option "HoldLastValue", die über den gleichnamigen Eingang aktiviert/deaktiviert werden kann. Falls "HoldLastValue" aktiv ist ("HoldLastValue" = 1), wird im Fehlerfall (Bad = 1) der letzte gültige Werte an "PV_Out" gehalten. Falls "HoldLastValue" inaktiv ist ("HoldLastValue" = 0), wird im Fehlerfall der an "SubstPV_In" angegebenen Wert an "PV_Out" ausgegeben.

Im Simulationsbetrieb ("SimOn" = 1) wird der an "SimPV_In" angegebene Wert an "PV_Out" ausgegeben. Die Ausgänge "Status", "Bad" und "SimAct" werden im Simulationsbetrieb auf 0 gesetzt.

4.4.2 Betriebsarten von RIOCount

Dieser Baustein verfügt über keine Betriebsarten.

4.4.3 Fehlerbehandlung von RIOCount

Die Fehlerbehandlung der Kanalbausteine ist in Kapitel 4.7 beschrieben.

4.4.4 Melden von RIOCount

Dieser Baustein hat kein Meldeverhalten.

4.4.5 Anschlüsse von RIOCount

Eingänge RIOCount

Eingang	Typ	Startwert	Sichtbar	Beschreibung
EN	BOOL	1	-	Enable
PV_In	DWORD	0	X	Verschaltung auf erstes Eingangs-Bit des zugehörigen Submoduls.
RIO_Conn	STRUCT	-	X	Enthält Informationen über das Submodul. Verschaltung zu RIOMOD wird automatisch per Treiberwizard eingefügt.
SimOn	BOOL + ST	0 16#80	X	1 = Simulation eingeschaltet
SimPV_In	DWORD + ST	0 16#80	X	Prozesswert, der bei SimOn = 1 verwendet wird
HoldLast-Value	BOOL	0	X	1 = Letzter Prozesswert wird im Fehlerfall gehalten. 0 = Ersatzwert wird verwendet
SubsPV_In	DWORD	0	-	Dieser Ersatzwert wird an PV_Out ausgegeben, falls HoldLastValue = 0 ist.
Reset	BOOL	0	X	Dieser Wert wird direkt auf das zyklische Ausgangsbit "Reset" geschrieben.
RolloverAck	BOOL	0	X	Dieser Wert wird direkt auf das zyklische Ausgangsbit "Rollover_Ack" geschrieben.
Rollunder-Ack	BOOL	0	X	Dieser Wert wird direkt auf das zyklische Ausgangsbit "Rollunde_Ack" geschrieben.
PresetValue	DWORD	0	X	Dieser Wert wird direkt auf das zyklische Ausgangs-Doppelwort "PresetValue" geschrieben.

Ausgänge RIOCount

Ausgang	Typ	Startwert	Sichtbar	Beschreibung
ENO	BOOL	0	-	Enable Out
Bad	BOOL + ST	0 16#80	X	1 = Prozesswert ist ungültig
PV_Out	DWORD + ST	0 16#80	X	Prozesswert
ModErr	BOOL + ST	0 16#80	-	1 = Gerät / Baugruppe ist gestört
Status	WORD + ST	0 16#80	-	Status-Wort Siehe 4.7
SimAct	BOOL + ST	0 16#80	-	1 = Simulation aktiv
Done	BOOL + ST	0 16#80	X	Zyklisches Eingangsbit "Done" wird hier ausgegeben.
Rollover	BOOL + ST	0 16#80	X	Zyklisches Eingangsbit "Rollover" wird hier ausgegeben.
Rollunder	BOOL + ST	0 16#80	X	Zyklisches Eingangsbit "Rollunder" wird hier ausgegeben.
Direction	BOOL + ST	0 16#80	-	Zyklisches Eingangsbit "Direction" wird hier ausgegeben.

4.5 RIODiIn – Digital-Eingangskanalbaustein

Dieses Kapitel enthält Informationen zu folgenden Themen:

- Beschreibung/Funktionen von RIODiIn
- Betriebsarten von RIODiIn
- Fehlerbehandlung von RIODiIn
- Melden von RIODiIn
- Anschlüsse von RIODiIn

4.5.1 Beschreibung/Funktionen von RIODiIn

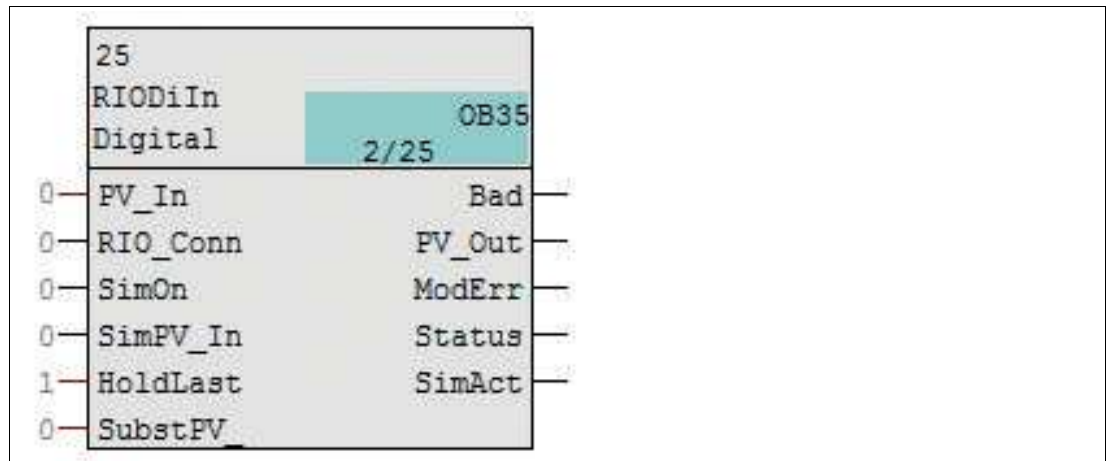


Abbildung 4.4 Kanalbaustein RIODiIn

Zur Projektierung muss der Eingang PV_In von RIODiIn auf den Prozesswert des gewünschten Submoduls vom Typ BOOL verschaltet werden. Die Projektierung der Kanalbausteine ist genauer in Kapitel 5 beschrieben.

Die hier aufgeführten Funktionen sind im Detail in Kapitel 2.2 beschrieben.

Der Kanalbaustein RIODiIn liest das am Eingang PV_In verschaltete Eingangsbit und gibt es an PV_Out aus, falls kein Fehler vorliegt. Die Diagnosebits des verschalteten Moduls werden ebenfalls gelesen und wie in Kapitel 4.7 beschrieben am Ausgangstatus ausgegeben. Falls ein Fehler auftritt, wird der Ausgang "Bad" auf 1 gesetzt wodurch der Quality Code aller Ausgänge wird auf 16#00 "Ungültiger Wert" gesetzt wird. Falls ein Rack- oder Modulfehler vorliegt, werden "ModErr" und "Bad" auf 1 gesetzt.

Der Kanalbaustein bietet die Option "HoldLastValue", die über den gleichnamigen Eingang aktiviert/deaktiviert werden kann. Falls "HoldLastValue" aktiv ist ("HoldLastValue" = 1), wird im Fehlerfall (Bad = 1) der letzte gültige Prozesswert an "PV_Out" gehalten. Falls "HoldLastValue" inaktiv ist ("HoldLastValue" = 0), wird im Fehlerfall der an "SubstPV_In" angegebene Wert an "PV_Out" ausgegeben.

Im Simulationsbetrieb ("SimOn" = 1) wird der an "SimPV_In" angegebene Wert an "PV_Out" ausgegeben. Die Ausgänge "Status", "Bad" und "ModErr" werden im Simulationsbetrieb auf 0 gesetzt. Der Ausgang "SimAct" wird auf 1 gesetzt.

4.5.2 Betriebsarten von RIODiIn

Dieser Baustein verfügt über keine Betriebsarten.

4.5.3 Fehlerbehandlung von RIODiIn

Die Fehlerbehandlung der Kanalbausteine ist in Kapitel 4.7 beschrieben.

4.5.4 Melden von RIODiIn

Dieser Baustein hat kein Meldeverhalten.

4.5.5 Anschlüsse von RIODiIn

Eingänge RIODiIn

Eingang	Typ	Startwert	Sichtbar	Beschreibung
EN	BOOL	1	-	1 = Aufgerufener Baustein wird bearbeitet
PV_In	BOOL	0	X	Verschaltung auf Prozesswert
RIO_Conn	STRUCT	-	X	Enthält Informationen über das Submodul. Verschaltung zu RIOMOD wird automatisch per Treiberwizard eingefügt.
SimOn	BOOL + ST	0 16#80	X	1 = Simulation eingeschaltet
SimPV_In	BOOL + ST	0 16#80	X	Prozesswert, der bei SimOn = 1 verwendet wird
HoldLast-Value	BOOL	1	X	1 = Letzter Prozesswert wird im Fehlerfall gehalten. 0 = Ersatzwert wird verwendet
SubsPV_In	BOOL	0	-	Dieser Ersatzwert wird im Fehlerfall an PV_Out ausgegeben, falls HoldLastValue = 0 ist.

Ausgänge RIODiIn

Ausgang	Typ	Startwert	Sichtbar	Beschreibung
ENO	BOOL	0	-	1 = Bausteinalgorithmus ist fehlerfrei durchlaufen
Bad	BOOL + ST	0 16#80	X	1 = Prozesswert ist ungültig
PV_Out	BOOL + ST	0 16#80	X	Prozesswert
ModErr	BOOL + ST	0 16#80	-	1 = Gerät / Baugruppe ist gestört
Status	WORD + ST	0 16#80	-	Status-Wort Siehe 4.7
SimAct	BOOL + ST	0 16#80	-	1 = Simulation aktiv

4.6 RIODiOu – Digital-Ausgangskanalbaustein

Dieses Kapitel enthält Informationen zu folgenden Themen:

- Beschreibung/Funktionen von RIODiOu
- Betriebsarten von RIODiOu
- Fehlerbehandlung von RIODiOu
- Melden von RIODiOu
- Anschlüsse von RIODiOu

4.6.1 Beschreibung/Funktionen von RIODiOu

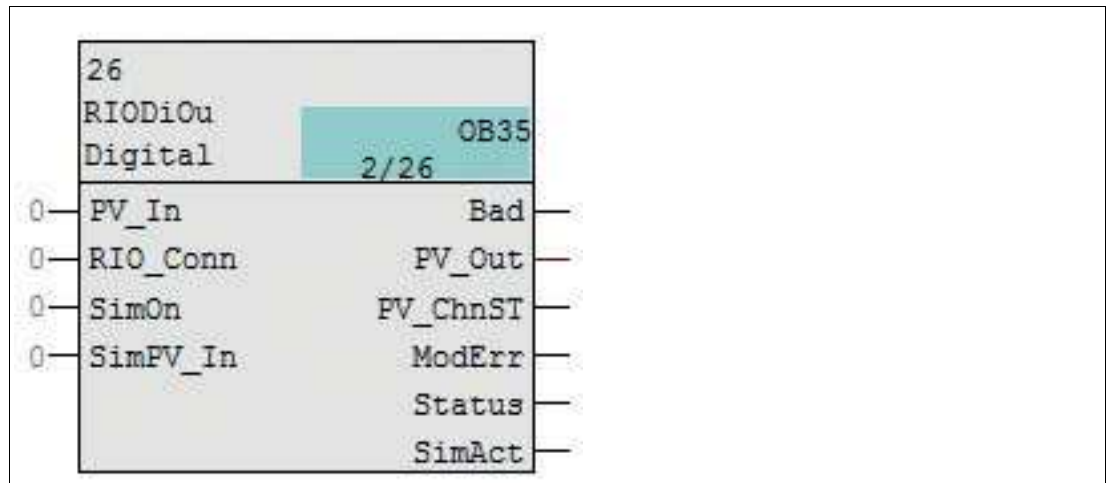


Abbildung 4.5 Kanalbaustein RIODiOu

Zur Projektierung muss der Ausgang PV_Out von RIODiOu auf den Prozesswert des gewünschten Submoduls vom Typ BOOL verschaltet werden. Die Projektierung der Kanalbausteine ist genauer in Kapitel 5 beschrieben.

Die hier aufgeführten Funktionen sind im Detail in Kapitel 2.2 beschrieben.

Der Kanalbaustein RIODiOu schreibt das an "PV_In" anliegende Bit an die über "PV_Out" verschaltete logische Adresse. Über die Struktur "RIO_Conn" bekommt der Kanalbaustein die logische Eingangsadresse des zugehörigen Submoduls übergeben. Das Eingangsbyte des Submoduls wird zyklisch gelesen und die darin enthaltenen Statusinformationen werden wie in Kapitel 4.7 beschrieben, am Ausgang Status ausgegeben. Falls "ModErr" = 1 ist, wird "Bad" ebenfalls auf 1 gesetzt.

Im Simulationsbetrieb ("SimOn" = 1) wird der an "SimPV_In" angegebene Wert an "PV_Out" ausgegeben. Die Ausgänge "Status", "Bad" und "ModErr" werden im Simulationsbetrieb auf 0 gesetzt. Der Ausgang "SimAct" wird auf 1 gesetzt.

4.6.2 Betriebsarten von RIODiOu

Dieser Baustein verfügt über keine Betriebsarten.

4.6.3 Fehlerbehandlung von RIODiOu

Die Fehlerbehandlung der Kanalbausteine ist in Kapitel 4.7 beschrieben.

4.6.4 Melden von RIODiOu

Dieser Baustein hat kein Meldeverhalten.

4.6.5 Anschlüsse von RIODiOu

Eingänge RIODiOu

Eingang	Typ	Startwert	Sichtbar	Beschreibung
EN	BOOL	1	-	1 = Aufgerufener Baustein wird bearbeitet
PV_In	BOOL	0	X	Eingangswert
RIO_Conn	STRUCT	-	X	Enthält Informationen über das Submodul. Verschaltung zu RIOMOD wird automatisch per Treiberwizard eingefügt.
SimOn	BOOL + ST	0 16#80	X	1 = Simulation eingeschaltet
SimPV_In	BOOL + ST	0 16#80	X	Prozesswert, der bei SimOn = 1 verwendet wird

Ausgänge RIODiOu

Ausgang	Typ	Startwert	Sichtbar	Beschreibung
ENO	BOOL	0	-	1 = Bausteinalgorithmus ist fehlerfrei durchlaufen
Bad	BOOL + ST	0 16#80	X	1 = Prozesswert ist ungültig
PV_Out	BOOL	0	X	Verschaltung auf Prozesswert
PV_ChnST	BOOL + ST	0 16#80	X	Signalstatus des Ausgangskanals und Wert von PV_Out
ModErr	BOOL + ST	0 16#80	-	1 = Gerät / Baugruppe ist gestört
Status	WORD + ST	0	-	Status-Wort Siehe 4.7
SimAct	BOOL + ST	0 16#80	-	1 = Simulation aktiv

4.7 Fehlerbehandlung

Die Kanalbausteine geben die zyklischen Diagnosedaten der Submodule am Ausgang "Status" und ggf. über den Ausgang "Status_HART" aus. Das Senden von Meldungen an das PCS 7 Meldesystem wird von RIOMOD übernommen (siehe Kapitel 3.3).

Zusätzlich zur zyklischen Diagnose werden folgende Fehler behandelt:

- Prüfung der Submodul ID, ob das verschaltete Submodul dem Kanalbaustein bekannt ist und ob es unterstützt wird. Z.B: Fehler bei Verschaltung eines Analogen Ausgangs auf RIOAnIn.
- Fehler beim Lesen der zyklischen Diagnosedaten mittels "BLKMOV" (SFB20).

Die folgenden zwei Tabellen zeigen die Bytereihenfolge und den Aufbau des Status-Ausgangs. Der Aufbau und die Bytereihenfolge sind für alle Kanalbausteine gleich. Die Spalte "Bad = true" gibt an, ob die entsprechende Diagnosemeldung den Ausgang "Bad" des Kanalbausteins auf "true" setzt und damit den Prozesswert als ungültig kennzeichnet. Nicht alle Submodul verfügen über alle Diagnosebits. Falls ein Diagnosebit von einem Submodul nicht bereitgestellt wird, wird dessen Wert auf 0 gesetzt.

Ausgangswort "Status" - Byteorder

Byte 0								Byte 1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Ausgangswort "Status" - Aufbau

Byte	Bit	Name	Bad = true
0	0	Nicht verwendet	-
0	1	Fault	X
0	2	Line Fault	X
0	3	Nicht verwendet	-
0	4	Open Wire	X
0	5	Short Circuit	X
0	6	Simulation Active	-
0	7	Nicht verwendet	-
1	0	Over Range	-
1	1	Under Range	-
1	2	High High Alarm	-
1	3	High Alarm	-
1	4	LowAlarm	-
1	5	Low Low Alarm	-
1	6	Das verschaltete Submodul wird nicht vom Kanalbaustein unterstützt.	X
1	7	Fehler beim lesen/schreiben der zyklischen Daten mittels "BLKMOV" (SFB20)	X

Der Status von HART-Submodulen wird am Ausgang "HART_Status" des Kanalbausteins ausgegeben. Die HART-Diagnosen haben keinen Einfluss auf den Ausgang "Bad" oder den Quality Code der HART-Ausgänge. Kommt es allerdings zu einem Fehler beim Lesen der zyklischen Daten des HART-Submoduls mittels "BLKMOV" (SFB20), wird 16#FF an HART_Status ausgegeben und der Quality Code aller HART Ausgänge wird auf 16#0 (Invalid value) gesetzt.

Ausgangsbyte HART_Status - Aufbau

Byte	Bit	Name
0	0	Non Primary Variable out of Limits
0	1	Primary Variable out of Limits
0	2	Analog Output Saturated
0	3	Analog Output Fixed
0	4	More Status Available
0	5	Cold Start
0	6	Configuration Changed
0	7	Field Device Malfunction

5 Projektierung

5.1 HW Konfig

Zur Projektierung der Kanalbausteine muss zunächst das Gateway incl. aller verwendeten Module und Submodule in HW Konfig eingebaut und wie gewünscht parametrieren werden. Hierzu muss zuvor die GSDML-Datei importiert werden. Die unterstützte Version der GSDML-Datei ist in Kapitel 2.1.3 aufgeführt.

The image shows two screenshots from a hardware configuration tool. The top screenshot displays a rack (UR2) with the following modules:

- 1 PS 407 10A
- 3 CPU 410E
- X1 DP
- IF1
- IF2
- X5 PN-IO-X5
- X5 P1 R Port 1
- X5 P2 R Port 2
- X8 PN-IO-X8
- X8 P1 R Port 1
- X8 P2 R Port 2
- 5

A connection line leads from the rack to a module labeled (3) LB-8122 with a status of '?????'. The bottom screenshot shows the detailed configuration for (3) LB-8122:

Steckplatz	Baugruppe	Bestellnumm...	E-Adresse	A-Adresse	Diagnoseadresse
0	LB-8122				16361*
X1	PN-IO				16360*
X1 P1	Port 1 - RJ45				16359*
X1 P2	Port 2 - RJ45				16358*
1	3x05 - Analog Input (4A~				0*
Channel 1	Analog Input		0...5		
Channel 2	Analog Input		6...11		
Channel 3	Analog Input		12...17		
Channel 4	Analog Input		18...23		
HART Channel 1	HART Dyn Var Pv Sv Tv Qv		191...207		
HART Channel 2	HART Dyn Var Pv Sv		219...227		
HART Channel 3	HART Dyn Var Sv		208...212		
HART Channel 4	HART Dyn Var Sv Tv Qv		228...240		
2					
3	6x17 - Digital Output (~				13**
Channel 1	Digital Output		41	13	
Channel 2	Digital Output		215	14	
4	2x17 - Solenoid Driver ~				15**
Channel 1	Digital Output		218	15	
Channel 2	Digital Input		216		
Channel 3	Digital Input		217		

Abbildung 5.1 Gateway in HW Konfig



Symboltabelle anlegen

Die hier beschriebene Vorgehensweise zeigt Ihnen, wie Sie beim Zusammenstellen Ihrer Submodule im Projekt Symbole über Dialogfelder ändern oder neu definieren können.

Wählen Sie das Submodul aus der Konfigurationstabelle, dem Sie ein Symbol vergeben wollen. Klicken Sie dazu mit der rechten Maustaste auf das gewünschte Submodul und wählen Sie im Dialogfenster die Funktion "Symbole bearbeiten".

↳ Das Eingabefenster für das angewählte Submodul öffnet sich. In der nachfolgenden Abbildung wurde unter anderem das Symbol "ED645_Value" angelegt, das mit dem Eingangsdoppelwort "ED645" verknüpft ist.

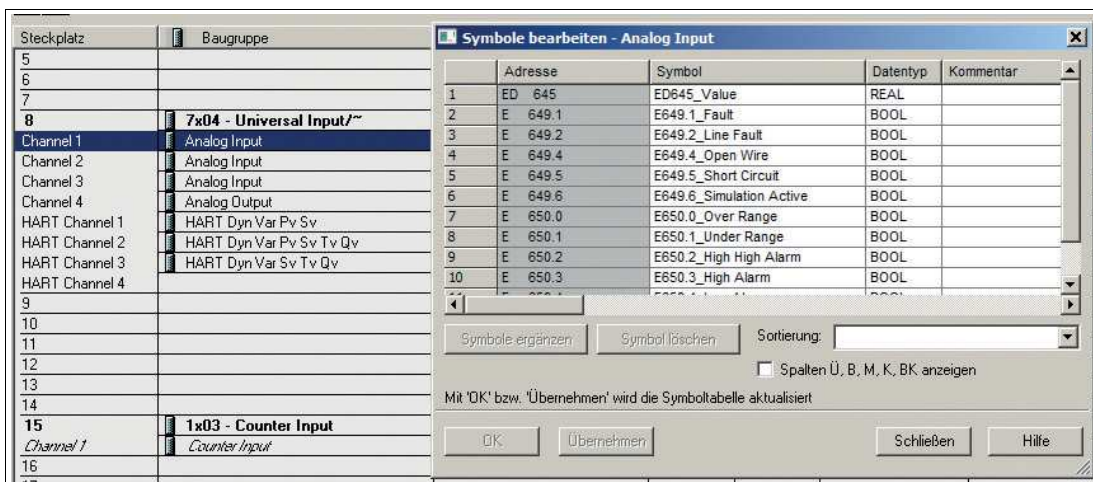


Abbildung 5.2 Symbole erzeugen



Hinweis!

Die so erzeugten Symbole erscheinen nach dem Übersetzen, wie in der nachfolgenden Abbildung zu sehen ist, in der Symboltabelle. Symbole können auch direkt in der Symboltabelle angelegt und mit einer beliebigen Adresse verknüpft werden. Wichtig ist, dass der Datentyp stimmt. Um z.B. den analogen Eingangstreiber RIOAnIn zu verwenden, muss der Eingang PV_IN auf das Eingangsdoppelwort des Submodul-Prozesswertes verschaltet werden und als Datentyp muss "REAL" gewählt sein. Daher kann RIOAnIn auch nicht direkt im CFC-Plan auf ein Doppelwort verschaltet werden, das mit keinem Symbol vom Datentyp "REAL" verknüpft ist.

S7-Programm(1) (Symbole) -- FP_RIO_PN\SIMATIC 400(1)\CPU 410E					
	Status	Symbol	Adresse	Datentyp	Kommentar
268		ED621_Value	ED 621	REAL	
269		ED645_Value	ED 645	REAL	
270		ED656_Value	ED 656	REAL	
271		ED671_Value	ED 671	REAL	
272		ED707_Counter Value	ED 707	DWORD	

Abbildung 5.3 Symbole in der Symboltabelle

5.2 CFC

Im CFC Plan kann der Kanalbaustein durch einen Rechtsklick auf "PV_In" > "Verschaltung zu Operand" mit dem gewünschten Symbol verschaltet werden.

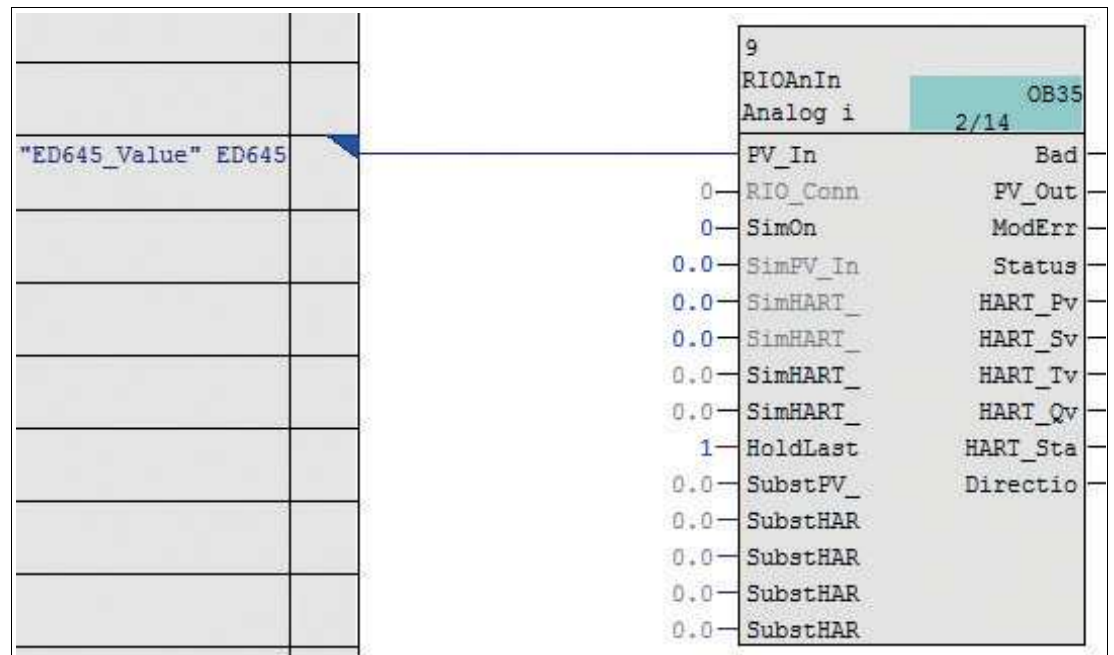


Abbildung 5.4 Verschaltung eines Kanalbausteins mit einem Symbol

Durch Übersetzen mit der gewählten Option "Baugruppentreiber erzeugen" wird der Treiberwizard ausgeführt, der automatisch in den @-Plänen alle benötigten Bausteine und Verschaltungen einfügt und den Eingang "RIO_Conn" des Kanalbausteins mit dem zugehörigen Modulbaustein verschaltet.

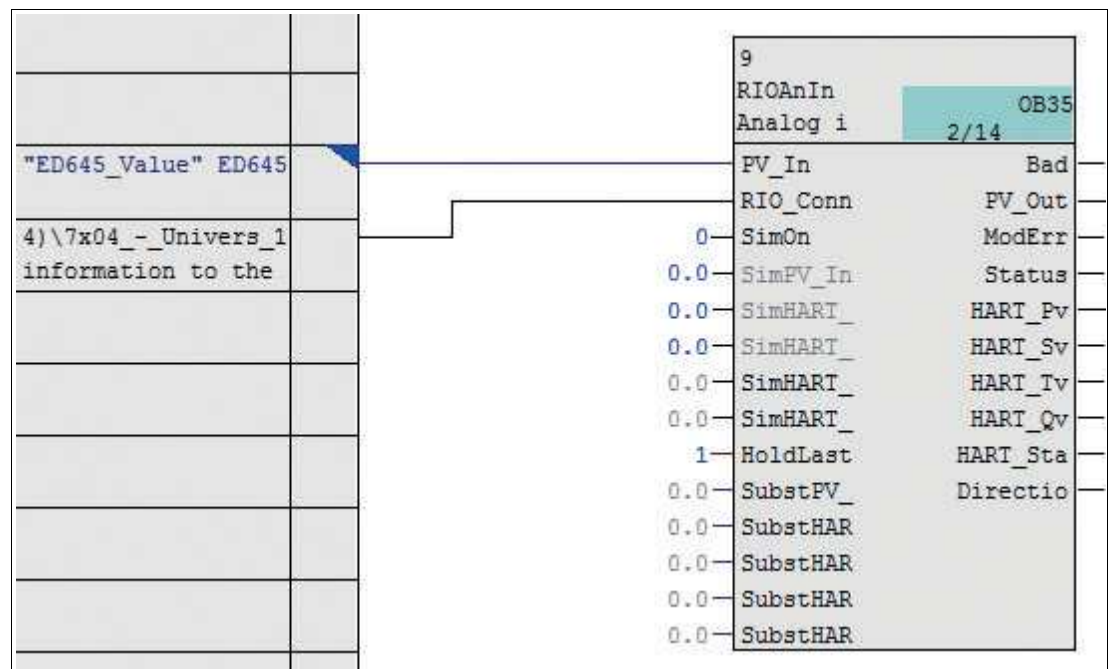


Abbildung 5.5 Der Eingang "RIO_Conn" wurde durch den Treiberwizard mit dem zugehörigen Modulbaustein verschaltet

Dieses Vorgehen ist für alle Kanalbausteine gleich. Das heißt "PV_In" (bzw. "PV_Out" für RIODiOu und RIOAnOu) muss auf den Prozesswert des gewünschten Submoduls verschaltet werden und alle zusätzlichen Bausteine und Verknüpfungen werden durch den Treiberwizard eingefügt.

Your automation, our passion.

Explosionsschutz

- Eigensichere Barrieren
- Signaltrenner
- Feldbusinfrastruktur FieldConnex®
- Remote-I/O-Systeme
- Elektrisches Ex-Equipment
- Überdruckkapselungssysteme
- Bedien- und Beobachtungssysteme
- Mobile Computing und Kommunikation
- HART Interface Solutions
- Überspannungsschutz
- Wireless Solutions
- Füllstandsmesstechnik

Industrielle Sensoren

- Näherungsschalter
- Optoelektronische Sensoren
- Bildverarbeitung
- Ultraschallsensoren
- Drehgeber
- Positioniersysteme
- Neigungs- und Beschleunigungssensoren
- Feldbusmodule
- AS-Interface
- Identifikationssysteme
- Anzeigen und Signalverarbeitung
- Connectivity

Pepperl+Fuchs Qualität

Informieren Sie sich über unsere Qualitätspolitik:

www.pepperl-fuchs.com/qualitaet

