

Handbuch | DE

TS8040

TwinCAT 2

Supplement |
Building Automation



Inhaltsverzeichnis

1	Vorwort.....	11
1.1	Hinweise zur Dokumentation	11
1.2	Zu Ihrer Sicherheit.....	12
1.3	Hinweise zur Informationssicherheit	13
2	Einleitung.....	14
3	Zielgruppen.....	15
4	Setup	16
5	Anforderungsprofil.....	17
6	Hardwareanforderungen	18
7	Grundlagen und Definitionen.....	19
7.1	Sequenzregler.....	19
7.2	Verschattungskorrektur	21
7.3	Sonnenschutz: Grundlagen und Definitionen.....	29
7.3.1	Effektiver Elevationswinkel.....	35
8	TcBA PLC-Bibliothek	37
8.1	Allgemeine Informationen	37
8.2	Funktionsbausteine	37
8.2.1	Verschattungskorrektur	44
8.2.2	Sonnenschutzautomatik.....	45
8.2.3	Liste der Verschattungselemente.....	46
8.2.4	Liste der Fassadenelemente.....	46
8.2.5	FB_BACnetAI1201.....	46
8.2.6	FB_BACnetAI1203.....	49
8.2.7	FB_BACnetAO1201	51
8.2.8	FB_BACnetAO1203.....	53
8.2.9	FB_BACnetAO1205.....	56
8.2.10	FB_BACnetAV1201	59
8.2.11	FB_BACnetAV1202	60
8.2.12	FB_BACnetAV1203	63
8.2.13	FB_BACnetAV1204	67
8.2.14	FB_BACnetAVDisplay.....	69
8.2.15	FB_BACnetAVSetpoint	70
8.2.16	FB_BACnetBI1201.....	71
8.2.17	FB_BACnetBI1203.....	72
8.2.18	FB_BACnetBI1205.....	74
8.2.19	FB_BACnetBO1201	76
8.2.20	FB_BACnetBO1202.....	78
8.2.21	FB_BACnetBO1203.....	82
8.2.22	FB_BACnetBO1205.....	84
8.2.23	FB_BACnetBV1201	87
8.2.24	FB_BACnetBV1202	89
8.2.25	FB_BACnetBV1203	91

8.2.26	FB_BACnetBV1204	94
8.2.27	FB_BACnetBVDisplay	96
8.2.28	FB_BACnetBVSetpoint	97
8.2.29	FB_BACnetCAL1201	98
8.2.30	FB_BACnetLoop1201	99
8.2.31	FB_BACnetLoop1202	103
8.2.32	FB_BACnetLoopSeq1201	107
8.2.33	FB_BACnetLoopSeq1202	112
8.2.34	FB_BACnetMI1203	115
8.2.35	FB_BACnetMO1202	117
8.2.36	FB_BACnetMO1203	120
8.2.37	FB_BACnetMV1201	123
8.2.38	FB_BACnetMV1202	125
8.2.39	FB_BACnetMV1203	127
8.2.40	FB_BACnetMVDisplay	130
8.2.41	FB_BACnetMVSetpoint	131
8.2.42	FB_BACnetSchedB1201	132
8.2.43	FB_BACnetSchedBinPV	133
8.2.44	FB_BACnetSchedR1201	134
8.2.45	FB_BACnetSchedUdi1201	136
8.2.46	FB_BACnetTLog1201	137
8.2.47	FB_BA_Anlg3Pnt	138
8.2.48	FB_BA_Cont4Stp01	139
8.2.49	FB_BA_RampLmt	143
8.2.50	FB_BA_SldgLmtMonit	145
8.2.51	FB_BA_StpDly	147
8.2.52	FB_BA_Swi2P	148
8.2.53	FB_BA_Swi2P_Dly	149
8.2.54	FB_BA_SwiHys2P	152
8.2.55	FB_BA_SwiHys2P_Dly	153
8.2.56	FB_BA_SwiMonit	155
8.2.57	FB_BA_FltrPT1	156
8.2.58	FB_BA_PIDCtrl	157
8.2.59	FB_BA_PIDCtrlEx	161
8.2.60	FB_BA_PISync1201	164
8.2.61	FB_BA_PISync1202	165
8.2.62	FB_BA_PWM	166
8.2.63	FB_BA_SeqCtrl	167
8.2.64	FB_BA_SeqLink	171
8.2.65	FB_BA_Chrct02	173
8.2.66	FB_BA_Chrct04	175
8.2.67	FB_BA_Chrct07	176
8.2.68	FB_BA_Chrct32	178
8.2.69	FB_BA_TiAavg	180
8.2.70	FB_BA_Alarm	182
8.2.71	FB_BA_AlarmMgnr	184

8.2.72	FB_BA_AlarmPlt	186
8.2.73	FB_BA_Alm.....	188
8.2.74	FB_BA_AlmColt04	189
8.2.75	FB_BA_AlmColt08	191
8.2.76	FB_BA_AlmColt12	193
8.2.77	FB_BA_AlmColt16	196
8.2.78	FB_BA_ComnMsg	200
8.2.79	FB_BA_ComnMsgTermt	203
8.2.80	FB_BA_DMUX_XX	206
8.2.81	FB_BA_MMUX_XX.....	208
8.2.82	FB_BA_MultiCalc_XX	211
8.2.83	FB_BA_MUX_XX.....	213
8.2.84	FB_BA_PrioSwi_XX.....	216
8.2.85	FB_BA_AntBlkg	218
8.2.86	FB_BA_DHW2P.....	219
8.2.87	FB_BA_FIFO04.....	221
8.2.88	FB_BA_FIFO08.....	222
8.2.89	FB_BA_FnctSel.....	225
8.2.90	FB_BA_FrstPrtc	228
8.2.91	FB_BA_HX.....	230
8.2.92	FB_BA_LglPrev.....	231
8.2.93	FB_BA_LmtCtrl	233
8.2.94	FB_BA_NgtCol.....	236
8.2.95	FB_BA_RcvMonit.....	237
8.2.96	FB_BA_RmTAdj.....	240
8.2.97	FB_BA_SpRmT.....	243
8.2.98	FB_BA_SpSupvis.....	246
8.2.99	FB_BA_StepCtrl08.....	249
8.2.100	FB_BA_StepCtrl12.....	252
8.2.101	FB_BA_BldPosEntry	257
8.2.102	FB_BA_BrtnsHysDly	259
8.2.103	FB_BA_CalcSunPos	260
8.2.104	FB_BA_FcdElemEntry	261
8.2.105	FB_BA_InRngAzm	265
8.2.106	FB_BA_InRngElv	268
8.2.107	FB_BA_RdFcdElemLst	270
8.2.108	FB_BA_RdShdObjLst	274
8.2.109	FB_BA_RolBldActr.....	279
8.2.110	FB_BA_ShdCorr	281
8.2.111	FB_BA_ShdObjEntry	284
8.2.112	FB_BA_SunBldActr.....	288
8.2.113	FB_BA_SunBldPosDly.....	293
8.2.114	FB_BA_SunBldEvt.....	295
8.2.115	FB_BA_SunBldPrioSwi4.....	295
8.2.116	FB_BA_SunBldPrioSwi8.....	297
8.2.117	FB_BA_SunBldScn.....	298

8.2.118	FB_BA_SunBldSwi	301
8.2.119	FB_BA_SunBldTwiLgtAuto	303
8.2.120	FB_BA_SunBldWthrPrtc	304
8.2.121	FB_BA_SunPrtc	306
8.2.122	FB_BA_LgtSwi	310
8.2.123	FB_BA_CnstLgtCtrl	313
8.2.124	FB_BA_AnlgLgtActr	316
8.2.125	FB_BA_DALILgtActr	318
8.2.126	FB_BA_Blink	322
8.2.127	FB_BA_CnvtTiSt	323
8.2.128	FB_BA_ExtTiSt	324
8.2.129	FB_BA_GetTime	325
8.2.130	FB_BA_SetTime	327
8.3	Aufzählungen und Strukturen	329
8.3.1	E_BA_AlmSta	329
8.3.2	E_BA_PosMod	330
8.3.3	E_BA_ShdObjType	330
8.3.4	ST_BA_BldPosTab	330
8.3.5	ST_BA_Cnr	331
8.3.6	ST_BA_ComnMsg	331
8.3.7	ST_BA_ComnMsgTermt	332
8.3.8	ST_BA_FcdElem	332
8.3.9	ST_BA_SeqLink / ST_BA_SeqLinkData	333
8.3.10	ST_BA_ShdObj	335
8.3.11	ST_BA_SpRmT	336
8.3.12	ST_BA_Sunbld	337
8.3.13	ST_BA_SunBldScn	337
8.3.14	E_BA_CtrlSttupMod	338
8.3.15	E_BA_OnMod	338
8.3.16	ST_BA_LgtCmd	339
8.3.17	E_BA_DALIMod	339
8.4	Fehlercodes	340
9	TwinCAT BA PLC Templates	347
9.1	Global_Variablen_Alarming	353
9.2	BAC_Gen_01	354
9.3	BAC_GenAlm_01	356
9.4	BAC_GenComnMsg_01	360
9.5	BAC_GenDvc_01	363
9.6	Global_Variables_General	364
9.7	BAC_GenNC_01	368
9.8	BAC_GenSys_01	370
9.9	BAC_GenWthT_01	371
9.10	BAC_PltAlm_01	372
9.11	BAC_PltComnMsg_01	376
9.12	BAC_AC_Humf_01	380
9.13	BAC_AC_Humf_PID_01	383

9.14	BAC_AC_SteamGenerator_01_xx.....	388
9.15	BAC_AC_FireDmp_01_xx	392
9.16	BAC_AC_ErcPI_01	398
9.17	BAC_AC_ErcPI_02	402
9.18	BAC_AC_ErcRecup_01	407
9.19	BAC_AC_ErcRot_01.....	412
9.20	BAC_AC_ErcRot_02.....	417
9.21	BAC_AC_ErcT_PID_01	420
9.22	BAC_ErclcPrt_01	426
9.23	BAC_ErclcPrt_02.....	428
9.24	BAC_AC_ExtAFan_FC_01	430
9.25	BAC_AC_ExtAFan1st_01	432
9.26	BAC_AC_SuAFan_FC_01	435
9.27	BAC_AC_SuAFan1st_01	438
9.28	BAC_DiffPrssMonit_01	441
9.29	BAC_DiffPrssMonit_02	443
9.30	BAC_AC_Filter_01.....	444
9.31	BAC_AC_ExhADmp2P_01_xx.....	446
9.32	BAC_AC_OuADmp2P_01_xx.....	454
9.33	BAC_AC_ColH_PID_01.....	460
9.34	BAC_AC_ColT_01	465
9.35	BAC_AC_ColT_02.....	468
9.36	BAC_AC_ColT_PID_01	471
9.37	BAC_AC_ColTH_01.....	476
9.38	BAC_AC_ColTH_02.....	480
9.39	BAC_AC_MixAT_01.....	483
9.40	BAC_AC_MixAT_PID_01.....	488
9.41	BAC_AC_ReHtr_01	493
9.42	BAC_AC_ReHtr_PID_01	496
9.43	BAC_AC_PreHtr_01	501
9.44	BAC_AC_PreHtr_PID_01	506
9.45	BAC_AC_RetWtrCtrl_01.....	511
9.46	BAC_FrstPrt_01	515
9.47	BAC_AC_VAV_01_xx.....	517
9.48	BAC_AC_OpMod_01	524
9.49	BAC_AC_SeqH_01.....	528
9.50	BAC_AC_SeqT_01	532
9.51	BAC_AC_StartT_01	538
9.52	BAC_AC_StartTH_01	543
9.53	BAC_AC_SumNgtCol_01	549
9.54	BAC_H_HtgCir_01	551
9.55	BAC_H_HtgCirSp_01.....	553
9.56	BAC_DHW_01	556
9.57	BAC_DHW_Ctrl_01.....	557
9.58	BAC_HW_LglPrev_01.....	560
9.59	BAC_Uni_FC_01_xx.....	563

9.60	BAC_Uni_Dmp_01_xx	568
9.61	BAC_Uni_Dmp2P_01_xx	572
9.62	BAC_Uni_Mot1st_01_xx	576
9.63	BAC_Uni_Pu1st_01_xx	579
9.64	BAC_Uni_SmokeDetc_001	586
9.65	BAC_Cont4Stp_01	588
9.66	BAC_HX_01	592
9.67	BAC_Hys_01	594
9.68	BAC_Hys_02	595
9.69	BAC_PID_01	597
9.70	BAC_PID_02	599
9.71	BAC_PID_03	601
9.72	BAC_Ramp_01	602
9.73	BAC_Ramp_02	603
9.74	BAC_Scale_02	605
9.75	BAC_Scale_04	606
9.76	BAC_Scale_07	607
9.77	BAC_Uni_Vlv_01_xx	608
9.78	BAC_Uni_Vlv3P_01_xx	612
9.79	Lüftungsanlagen	618
9.79.1	BAC_AC_KennzeichnungssystemAnlagenschlüssel	618
9.79.2	BAC_AC_SE_3_4_1_1_1_0	619
9.79.3	BAC_AC_SE_4_4_1_1_0_1	625
9.79.4	BAC_AC_SE_4_4_1_1_3_0	631
9.79.5	BAC_AC_Sx_001	638
9.80	Sollwerte der Sequenz-Regler	641
9.80.1	BAC_AC_CasCtrlH_01	642
9.80.2	BAC_AC_CasCtrlH_02	645
9.80.3	BAC_AC_CasCtrlT_01	649
9.80.4	BAC_AC_CasCtrlT_02	653
9.80.5	BAC_AC_SpRmT_01	658
9.80.6	BAC_AC_SpRmT_02	661
9.80.7	BAC_AC_SpRmTH_01	663
9.80.8	BAC_AC_SpRmTH_02	666
9.80.9	BAC_AC_SpRmTH_03	670
9.80.10	BAC_AC_SpRmTH_04	674
9.80.11	BAC_AC_SpSuAT_01	678
9.80.12	BAC_AC_SpSuAT_02	681
9.81	BACnet Objekte	683
9.81.1	BAC_AI_01	683
9.81.2	BAC_AI_02	685
9.81.3	BAC_AI_Enthalpy_01	687
9.81.4	BAC_AI_WthT_01	689
9.81.5	BAC_AO_01	691
9.81.6	BAC_AO_02	693
9.81.7	BAC_AO_03	695

9.81.8	BAC_AV_01	697
9.81.9	BAC_AV_02	699
9.81.10	BAC_AV_03	700
9.81.11	BAC_AV_04	702
9.81.12	BAC_BI_01	703
9.81.13	BAC_BI_02	704
9.81.14	BAC_BI_CMD_01	706
9.81.15	BAC_BO_01	707
9.81.16	BAC_BO_02	710
9.81.17	BAC_BV_01	712
9.81.18	BAC_BV_02	713
9.81.19	BAC_CMD_01	714
9.81.20	BAC_MV_01	715
9.81.21	BAC_MV_02	716
9.81.22	BAC_SchedBinPV_01	717
9.81.23	BAC_SchedB_01	718
9.81.24	BAC_SchedR_01	719
9.81.25	BAC_SchedUdj_01	720
9.81.26	BAC_TL_01	721
9.82	IO	722
9.82.1	P_KL1501	722
9.82.2	P_KL27x1	723
9.82.3	P_KL320x	724
9.82.4	P_KL3204	726
9.82.5	P_KL3208	727
9.82.6	P_KL3228	728
9.82.7	P_KL8519	729
9.82.8	P_KL8524	732
9.82.9	P_KL8528	734
9.82.10	P_KL8548	735
10	TwinCAT BA Project Builder	737
10.1	Erste Schritte	737
10.2	Hauptfenster	742
10.3	Optionen	751
10.4	AddIns erstellen	755
10.4.1	Erste Schritte - WPF	755
10.4.2	Erste Schritte - WinForms	760
10.4.3	Beispiele	765
11	Support und Service	770

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zuwendungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

2 Einleitung

Die Grundvoraussetzung, um die hohen Anforderungen an die Gebäudeautomation, wie Komfort, Energieeinsparung, geringe Investitions- und Betriebskosten und ein schneller Return-of-Investment, zu erfüllen, ist ein durchgängiges, aufeinander abgestimmtes Steuerungssystem für die Automation aller technischen Ausbaugewerke. Mit TwinCAT Building Automation hat Beckhoff ein Softwareprodukt entwickelt, das die Engineeringzeit reduziert und alle wesentlichen Funktionen für alle Gewerke einer modernen Gebäudeautomation integriert. Umfangreiche Softwarebibliotheken und Supplements setzen den Gedanken des modularen Beckhoff-Automatisierungsbaukastens auch auf der Softwareebene fort. Die neue Softwaresuite umfasst im Wesentlichen drei Grundfunktionen:

1. TwinCAT BA PLC Libraries: Basisfunktionen für alle Gewerke.
2. TwinCAT BA PLC Templates: Funktionsvorlagen für alle Gewerke.
3. TwinCAT BA Project Builder: Konfigurationstool, das Templates, Hardware und BACnet-Objekte miteinander verknüpft.

3 Zielgruppen

Diese Software ist für Gebäudeautomations-Systempartner der Beckhoff Automation GmbH & Co. KG. Die Systempartner sind tätig in dem Bereich Gebäudeautomation und beschäftigen sich mit Errichtung, Inbetriebsetzung, Erweiterung, Wartung und Service von mess-, steuer- und regelungstechnischen Anlagen der technischen Gebäudeausrüstung.

4 Setup

Systemvoraussetzungen

- Beckhoff TwinCAT PLC V2.11 Build 2254 oder höher
- Microsoft .NET Framework 4.5.2 oder höher
- Windows 7 SP1, Windows 10 oder höher

Installation

1. Ausführen der *.exe Datei des TwinCAT Supplements.
2. Klicken Sie auf *Weiter*, lesen Sie die Lizenz Vereinbarung sorgfältig, akzeptieren Sie diese und klicken auf *Weiter*.
3. Tragen Sie den Benutzernamen, Name der Firma und die Seriennummer ein.
4. Klicken Sie auf *Installieren* zum Starten der Installation.

Deinstallation

1. Öffnen Sie die Systemsteuerung und wählen *Programme und Funktionen* aus.
2. Wählen Sie *TwinCAT Building Automation* aus und klicken auf *Deinstallieren*.
3. Zum Starten der Deinstallation mit *Ja* bestätigen.

5 Anforderungsprofil

Für den Nutzer werden die folgenden Grundkenntnisse vorausgesetzt.

- TwinCAT PLC Control
- TwinCAT System Manager
- PC und Netzwerkkennnisse
- Aufbau und Eigenschaften der Beckhoff Embedded-PC und deren Busklemmensystem
- Kenntnisse in Heizungs-, Lüftungs-, Klima-, Sanitäreanlagen sowie der Raumautomation
- Einschlägige Sicherheitsvorschriften der technischen Gebäudeausrüstung

6 Hardwareanforderungen

Die Software ist auf allen PC basierenden Hardware-Plattformen einsetzbar. Ideale Zielplattform für die Anwendungen der Heizungs-, Lüftungs-, Klima- und Sanitärtechnik sind die Embedded PC's der Baureihe CX.

7 Grundlagen und Definitionen

7.1 Sequenzregler

In der Heizungs-, Lüftungs- und Klimatechnik kommt es häufig vor, dass zum Erreichen einer Regelgröße mehrere Stellglieder verwendet werden, die in einer so genannten Reglersequenz arbeiten.

In der unten dargestellten Klimaanlage sind an der Zulufttemperaturregelung drei Stellorgane beteiligt. In der BA-Bibliothek wird für jedes dieser Stellorgane ein eigener Sequenzregler instanziiert.

Bei aktiver Regelung ist immer nur einer dieser Sequenzregler aktiv. Die anderen nicht aktiven Regler fixieren ihr Stellsignal so, wie es energetisch für die Temperierung der Zulufttemperatur optimal ist.

Das bedeutet in Abhängigkeit des Wirksinns des einzelnen Reglers entweder das Maximum oder das Minimum für die Stellgröße I_rY .

Wenn die Wirkung des aktiven Stellgliedes (Reglers) bei dem Erreichen einer Endlage nicht ausreicht, wird von dem aktiven Regler auf den links oder rechts benachbarten Regler umgeschaltet.

Dieser übernimmt damit die Regelung. Der zuvor aktive Regler verharrt je nach Wirksinn in der Endlage von I_rY_{Max} oder rY_{Min} . Ebenso wird mit den weiteren Stellgliedern verfahren, bis der Sollwert oder das rechte oder linke Ende der Sequenz erreicht ist.

In der Sequenz der dargestellten Raumlufttechnischen Anlage sind alle Stellglieder, die die Regelgröße beeinflussen, von links nach rechts dargestellt. Ganz links steht das Stellglied, das die größtmögliche Erhöhung der Zulufttemperatur ermöglicht, ganz rechts das Stellglied, das die größtmögliche Verringerung der Zulufttemperatur bewirkt.

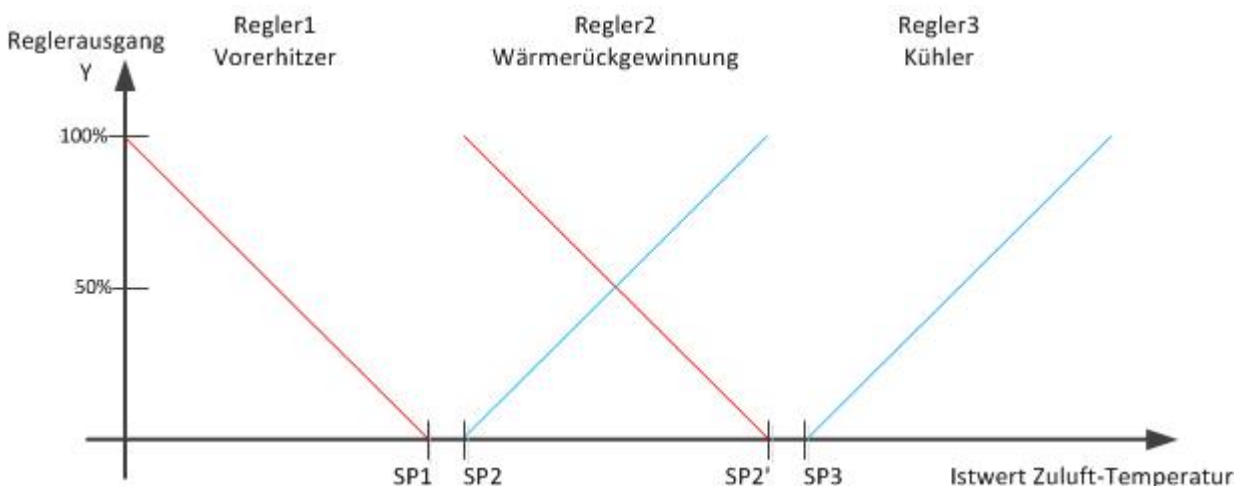
Manche Stellglieder, wie z.B. eine Umluftklappe oder eine Wärmerückgewinnung wechseln während des Betriebs ihre Wirkrichtung. (indirekt = heizen, direkt = kühlen)

Stellglieder mit wechselnden Wirksinn, wie z.B. Außenluft- und Umluftklappe oder Wärmerückgewinnung, werden nur einmal aufgeführt.

- Regler 1: Vorerhitzer
- Regler 2: Mischluft
- Regler 3: Kühler

Schematische Darstellung

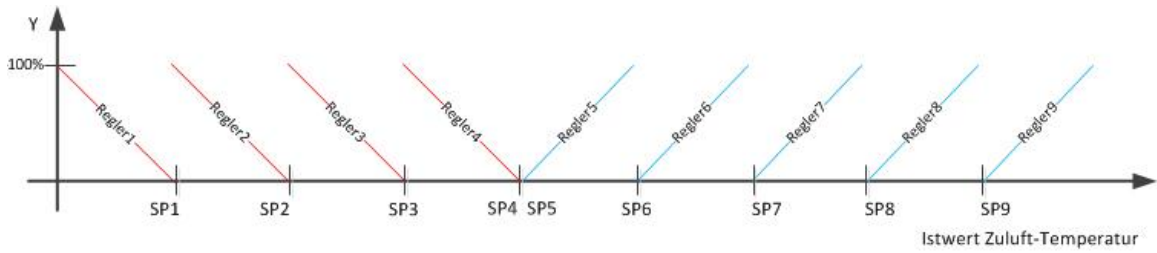
Diese Anlage wird schematisch wie folgt dargestellt:



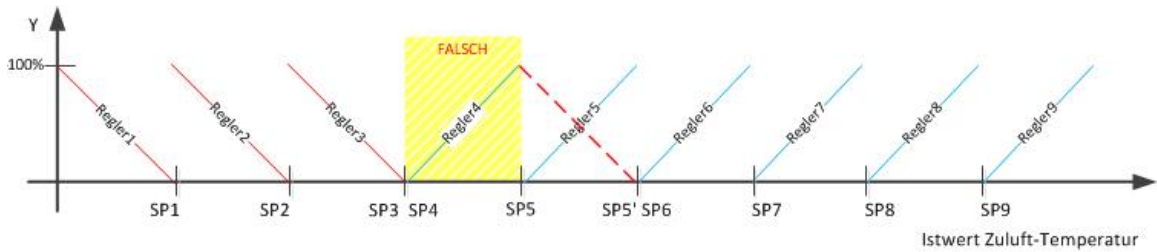
Regeln für die Aufstellung einer Sequenz

Für die Aufstellung der Sequenzen sind folgende Regeln zu beachten, wobei thematisch auf eine Zuluftregelung Bezug genommen wird:

1. Die Nummerierung der Sequenzregler erfolgt von den Heizsequenzen mit niedrigen Ordnungszahlen hin zu den Kühlsequenzen mit hohen Ordnungszahl kontinuierlich steigend.

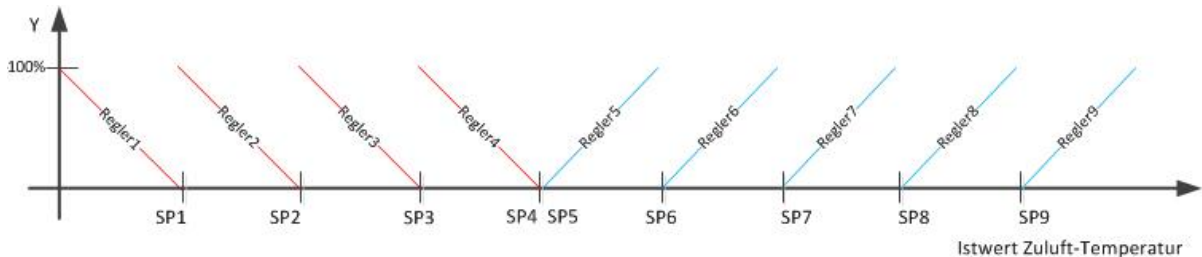


2. In eine Folge von Heizsequenzen sollte keine Kühlsequenz eingebunden werden. Und in eine Folge von Kühlsequenzen sollte keine Heizsequenz eingebunden werden. Sequenzen mit einer Wirksinnumkehr für ein Mischluftsystem oder eine Wärmerückgewinnung sind zwischen den Heiz- und den Kühlsequenzen zu positionieren.



In diesem Bild wäre Regler 4 falsch platziert, wenn Regler 5 in den Heizbetrieb wechseln würde. Oder: Regler 4 ist richtig, Regler 5 müsste jedoch ein reiner Kühlregler sein. In beiden Fällen wären 2 Wechsel von Heizen nach Kühlen vorhanden.

3. Die Sollwerte innerhalb der Sequenz müssen monoton steigend sein. Diese Forderung ergibt sich aus dem oben erläuterten Umschaltverhalten: Ist der Sollwert eines niedrigeren Reglers höher als der des nächst höheren, so kann es zu einem ständigen Hin- und Herschalten zwischen zwei Reglern führen. Wie oben bereits erwähnt, haben jedoch Regler mit gleichem Wirksinn üblicherweise den gleichen Sollwert.



$$SP1 \leq SP2 \leq SP3 \leq SP4 \leq SP5 \leq SP6 \leq SP7 \leq SP8 \leq SP9$$

Sequenzregler in der PLC

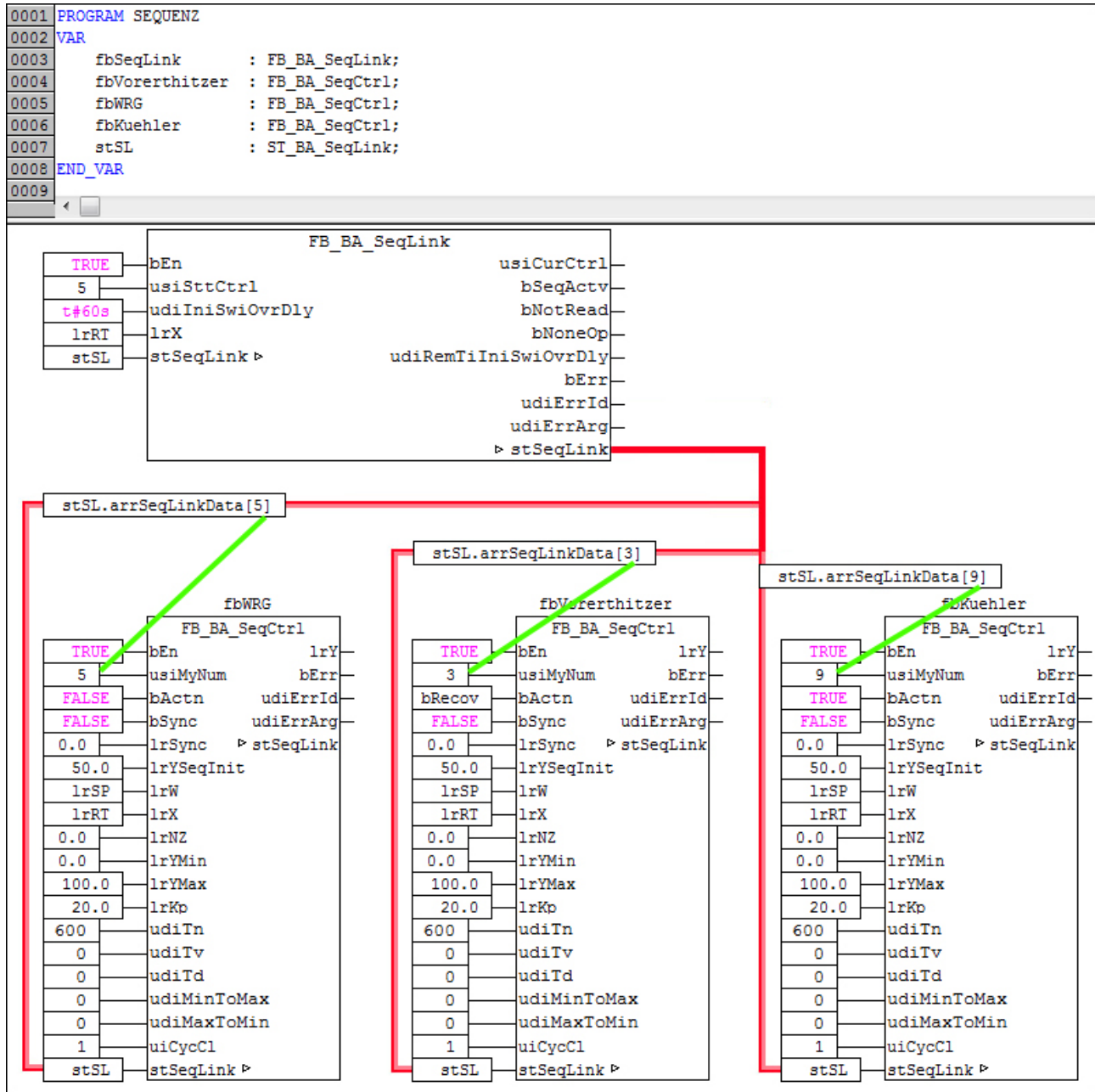
Für die Realisierung einer Sequenzregelung im SPS-Programm stellt die BA-Bibliothek zwei Bausteine zur Verfügung:

- Der Baustein **FB_BA_SeqCtrl** [► 167]. Er stellt einen einzelnen Regler als Bestandteil einer Sequenz von maximal 16 Reglern dar.
- Der Baustein **FB_BA_SeqLink** [► 171]. Dieser Baustein ist der Kontrollbaustein der Sequenz und existiert demnach pro Sequenz nur einmal. Er entscheidet, welcher Regler der Sequenz gerade aktiv ist und überprüft die Sequenz auf bestimmte Fehlerzustände, wie beispielsweise die doppelte Vergabe von Ordnungszahlen an den Reglern.

Die Strukturvariable **ST_BA_SeqLink** [► 333] dient zur Verbindung der Sequenzregler mit dem Sequenzlinker **FB_BA_SeqLink** [► 171].

Diese Strukturvariable ist pro Sequenzregelung einmal zu deklarieren.

Die Freigabe der Sequenzregelung erfolgt am Eingang *bEn* des Funktionsbausteins **FB_BA_SeqLink** [► 171]. Mit der Variablen *usiStartCtrl* wird bestimmt, mit welchem Regler nach dem Start des Regelbetriebs begonnen wird zu regeln. Im Beispiel wird mit dem Sequenzregler mit der Nr. 5 gestartet. Das Umschalten von dem Regler 5 auf einen anderen Regler in der Sequenz ist nach dem Neustart der Regelung um den Wert der Eingangsvariable *tniSwiOvrDly* um 60 Sekunden gesperrt.



7.2 Verschattungskorrektur

Die Verschattungskorrektur ist in Verbindung mit der Sonnenautomatik oder Lamellennachführung nutzbar. Die Funktion prüft, ob ein Fenster oder eine Gruppe von Fenstern, die z.B. einem Raum zugeordnet sind, temporär durch umliegende Bebauung oder eigene Gebäudeteile verschattet werden. Für Fenster, welche im Schatten umliegender Gebäude oder Bäume stehen ist ein Sonnenschutz nicht notwendig, unter Umständen sogar störend. Die Verschattungskorrektur ermittelt anhand von eingetragenen Daten der Fassade und ihrer Umgebung, welche Teile der Fassade verschattet werden. Damit ist es dann möglich, für einzelne Fenster oder Fenstergruppen zu entscheiden, ob der Sonnenschutz aktiv sein soll. Neben dem aktuellen Sonnenstand hängt die Verschattung der einzelnen Fenster von drei Dingen ab:

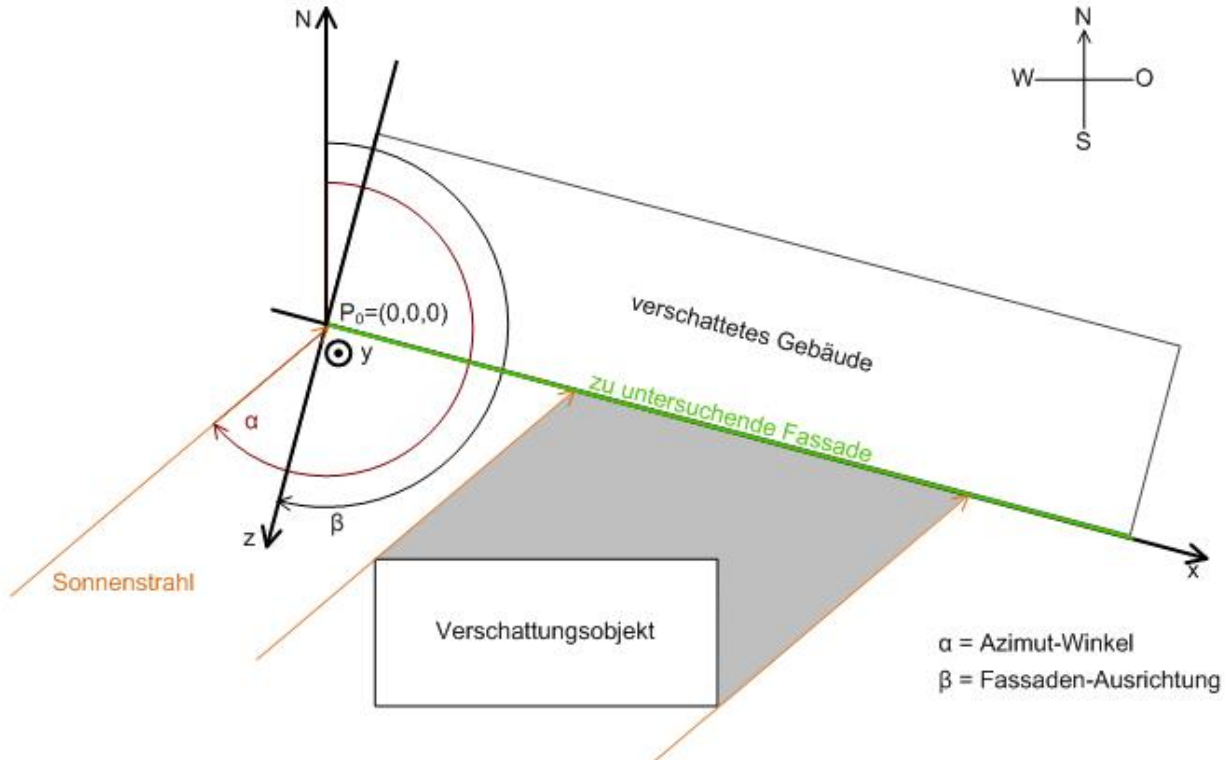
- der Ausrichtung der Fassade
- der Lage der Fenster

- der Positionierung der Verschattungsobjekte

Die folgenden Abbildungen sollen diese Zusammenhänge erläutern und die einzutragenden Parameter vorstellen.

Ausrichtung der Fassade

Betrachtung von oben



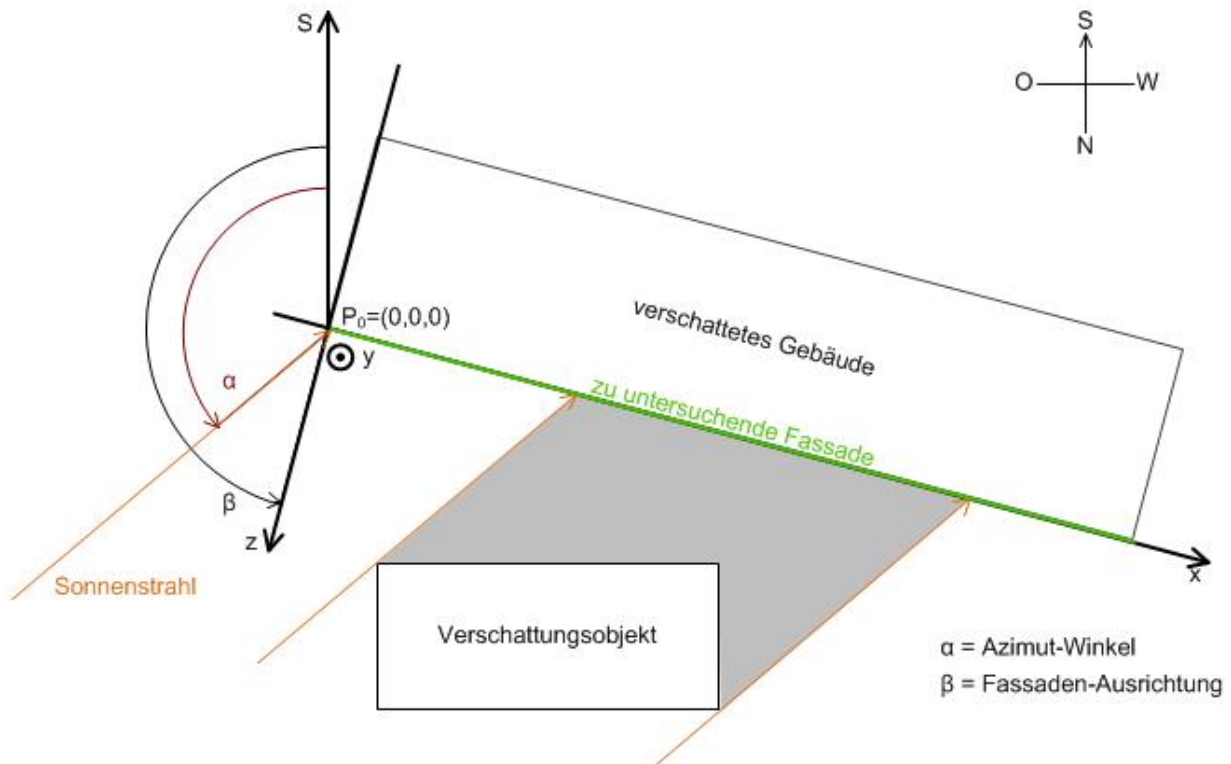
Für die reine Betrachtung des Schattenwurfes auf die Fassade ist letztendlich ein zweidimensionales Koordinatensystem erforderlich, daher wurden die x- und y-Achse auf die Fassade gelegt. Der Nullpunkt liegt dabei im Fußpunkt links unten, so als würde man die Fassade von vorne betrachten. Zur Bemessung der verschattenden Objekte kommt dann noch die Z-Komponente hinzu. Deren Achse weist von der Fassade weg und hat denselben Nullpunkt, wie die x- und y-Achse.

Der horizontale Sonnenstand (Azimutwinkel) ist auf der Nordhalbkugel per Definition von der Nordrichtung aus bemessen. Die Fassadenausrichtung richtet sich ebenfalls nach der Nordrichtung, wobei die Blickrichtung aus einem Fenster der Fassade gilt:

Blickrichtung	Fassadenausrichtung
Nord	$\beta=0^\circ$
Ost	$\beta=90^\circ$
Süd	$\beta=180^\circ$
West	$\beta=270^\circ$

Auf der Südhalbkugel ist der Sonnenverlauf anders herum: Sie geht zwar auch im Osten auf, hat ihren Mittagsstand jedoch im Norden. Die Fassadenausrichtung wird diesem Verlauf angepasst:

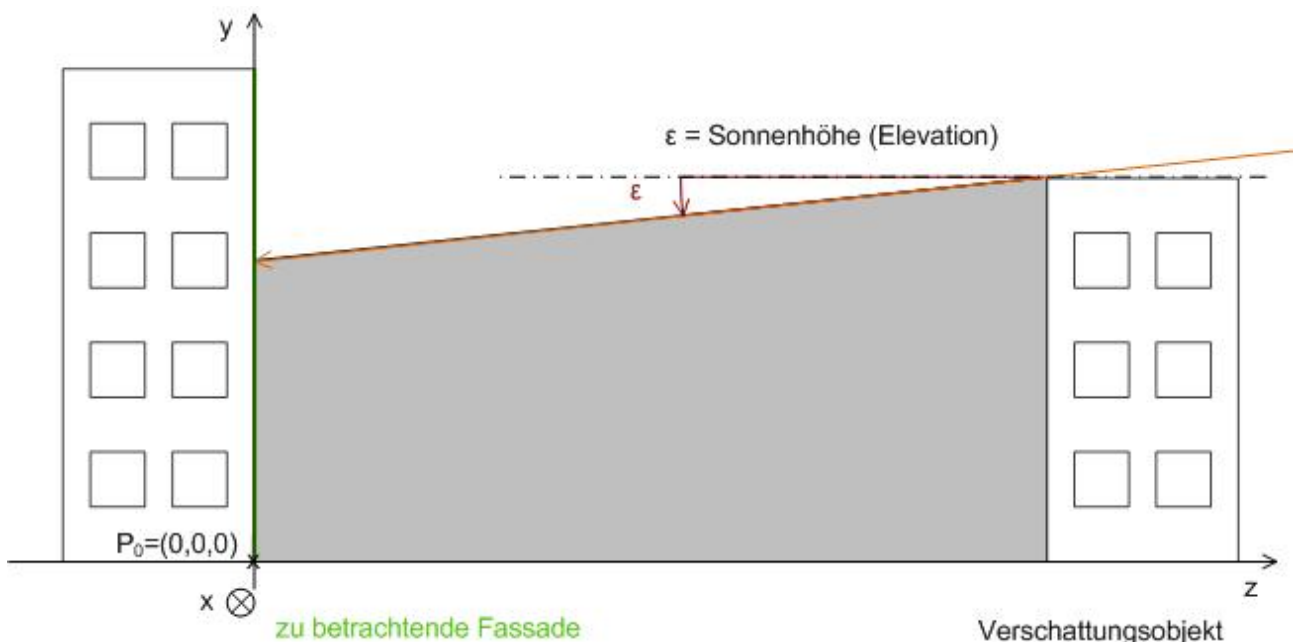
Blickrichtung	Fassadenausrichtung
Süd	$\beta=0^\circ$
Ost	$\beta=90^\circ$
Nord	$\beta=180^\circ$
West	$\beta=270^\circ$



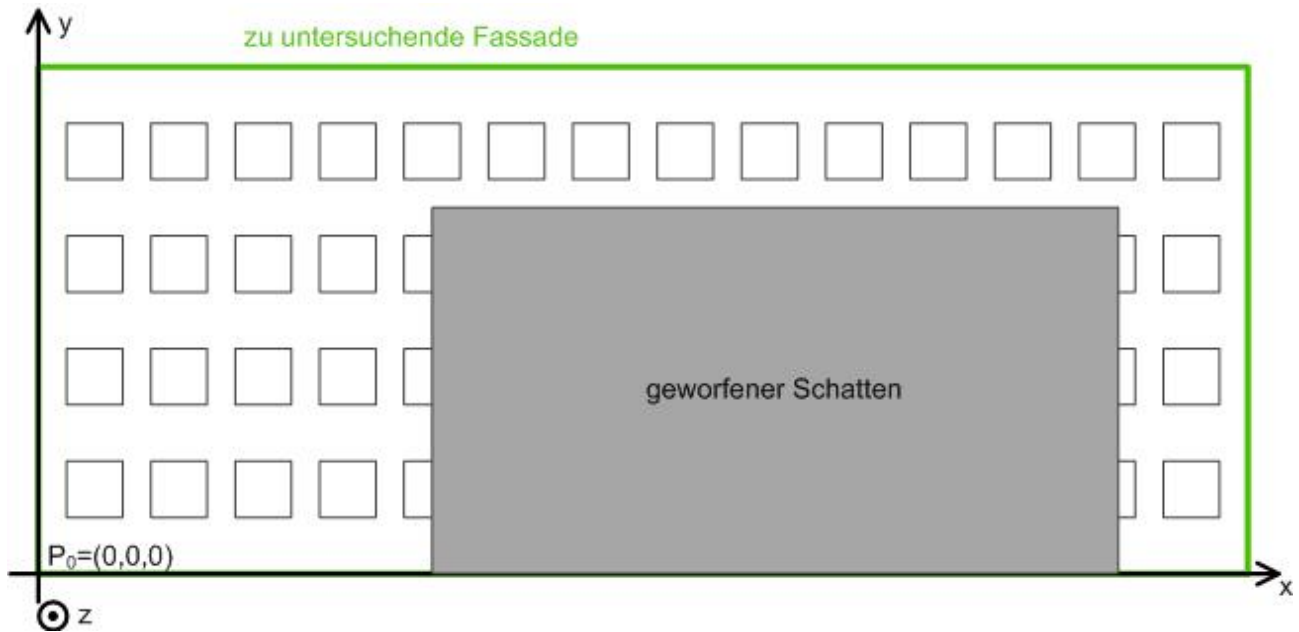
Die weiteren Ausführungen beziehen sich jedoch der Einfachheit halber nur auf die Anwendung auf der Nordhalbkugel. Die Berechnungen auf der Südhalbkugel gestalten sich analog und werden bei der Parametrierung des Bausteines FB_BA_ShdCorr [► 281] (Verschattungskorrektur) durch einen booleschen Eingang, *bSouth*, aktiviert

Die beiden folgenden Abbildungen sollen die Lage des Ursprungspunktes P_0 sowie die Ausrichtung des Koordinatensystems weiter verdeutlichen:

Betrachtung von der Seite



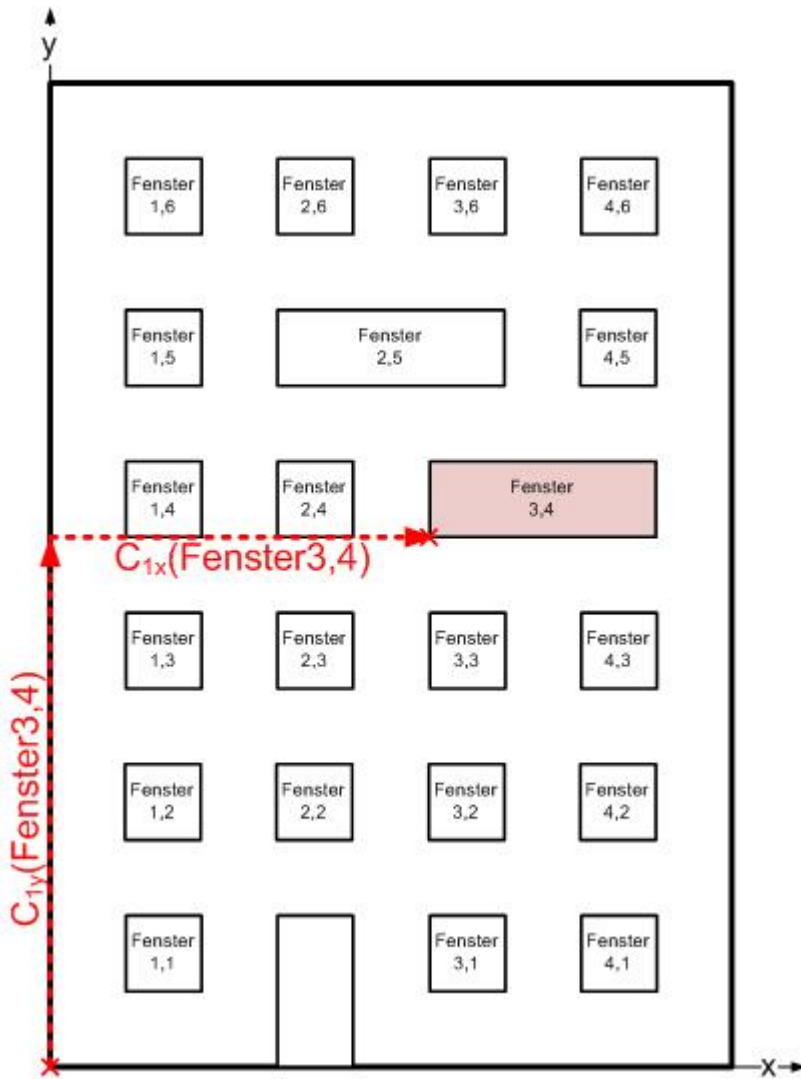
Anhand dieser Abbildung lässt sich auch der Elevationswinkel (Sonnenhöhe) darstellen: per Definition ist dieser bei Sonnenaufgang 0° (horizontaler Lichteinfall) und kann maximal 90° erreichen, dies jedoch nur an Orten innerhalb des nördlichen und südlichen Wendekreises.

Betrachtung von vorne

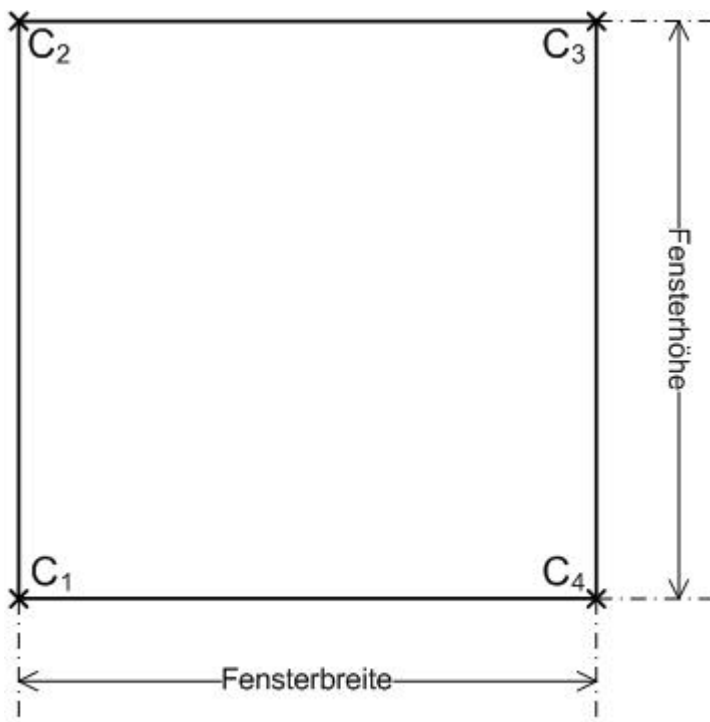
Hier ist die Lage des Koordinatenursprungs, P_0 , am linken unteren Fußpunkt der Fassade noch einmal besonders deutlich. Darüber hinaus ist die x-y-Ausrichtung dargestellt, die später für den Eintrag der Fensterelemente wichtig ist.

Lage der Fenster

Die Lage der Fenster wird durch die Angabe ihres linken unteren Eckpunktes in Bezug auf das Fassaden-Koordinatensystems definiert. Da ein Fenster plan auf der Fassade liegt, ist die Eingabe auf die x- und die y-Koordinate beschränkt.



Zusätzlich sind die Breite und die Höhe anzugeben.



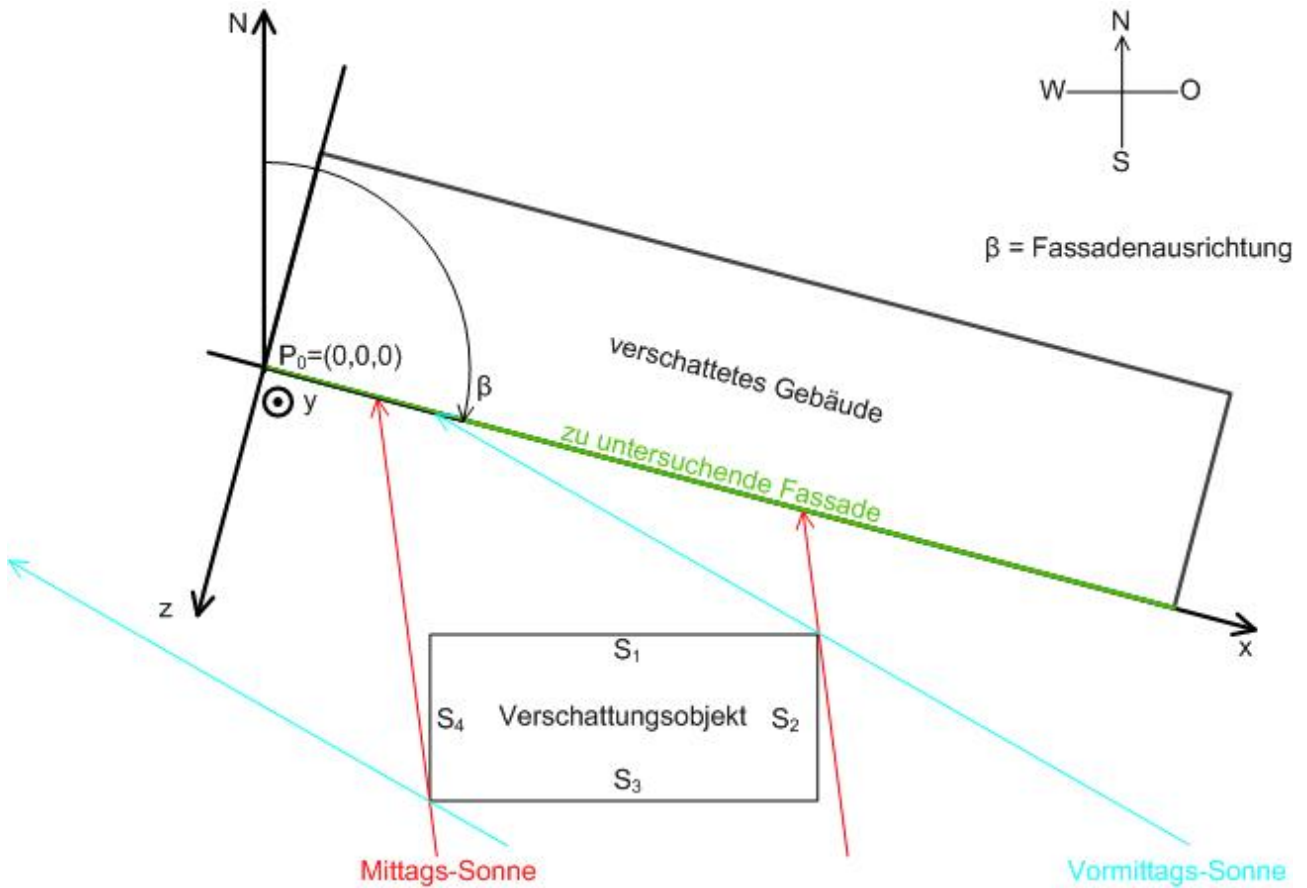
C_n =Ecke n (Corner)

Aus den eingetragenen Werten wird intern die Lage jedes Fenstereckpunktes auf der Fassade ermittelt. Ein Fenster gilt dann als verschattet, wenn alle Eckpunkte im Schatten liegen.

Positionierung der Verschattungsobjekte

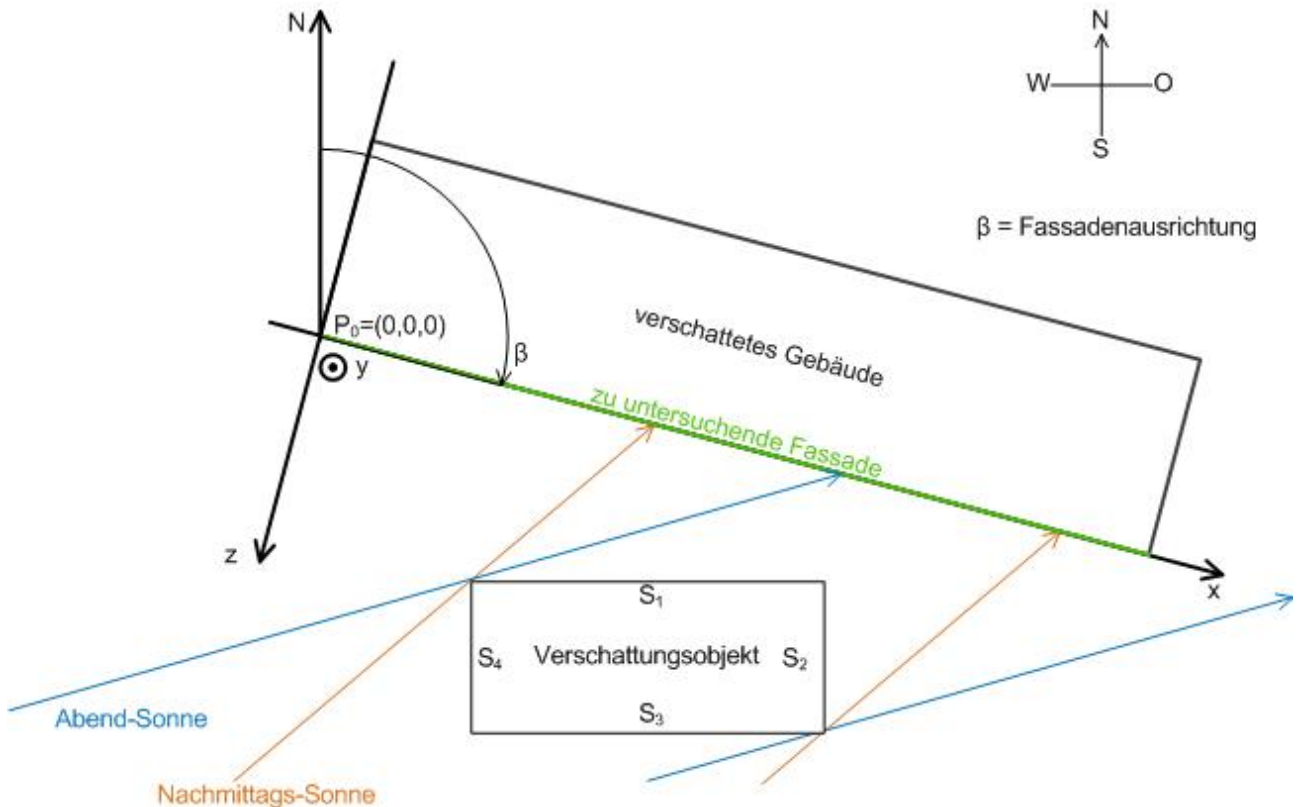
Bei der Beschreibung der Verschattungsobjekte wird zwischen eckigen Objekten (Gebäude, Pfeiler) und Objekten, die annähernd kugelförmig sind (z.B. Bäume), unterschieden. Eckige Objekte lassen sich ihrem Schattenwurf nach in viereckige Schatten werfende Fassaden unterteilen, wobei überlegt werden muss, welche über den Tag hinweg den Hauptschatten werfen:

Morgens/Mittags

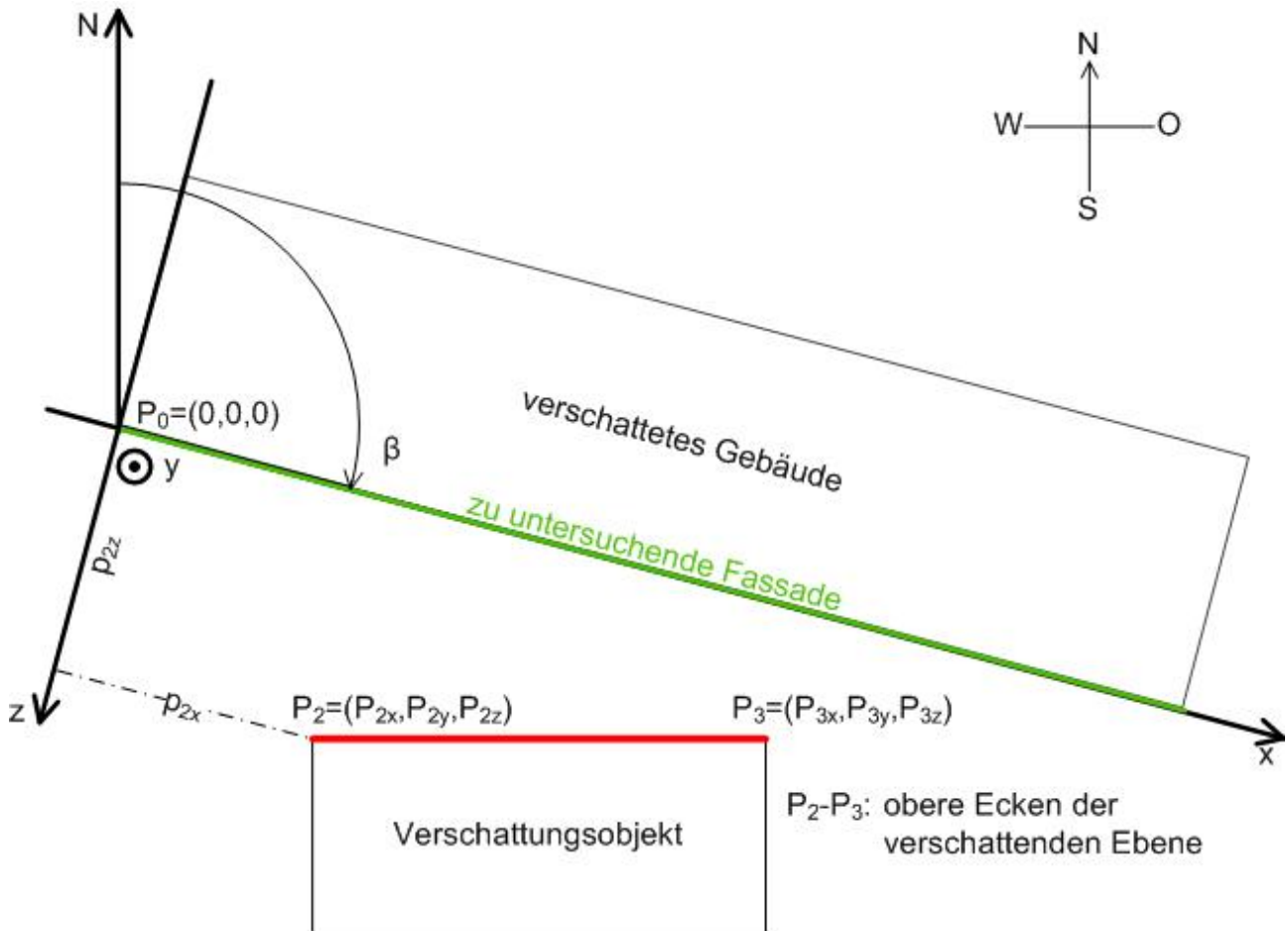


Morgens und Mittags würde der geworfene Schatten hauptsächlich durch die Seiten S_1 und S_4 gegeben sein, eine Betrachtung von S_2 und S_3 , sollten sie nicht höher sein, wäre nicht nötig.

Nachmittags/Abends



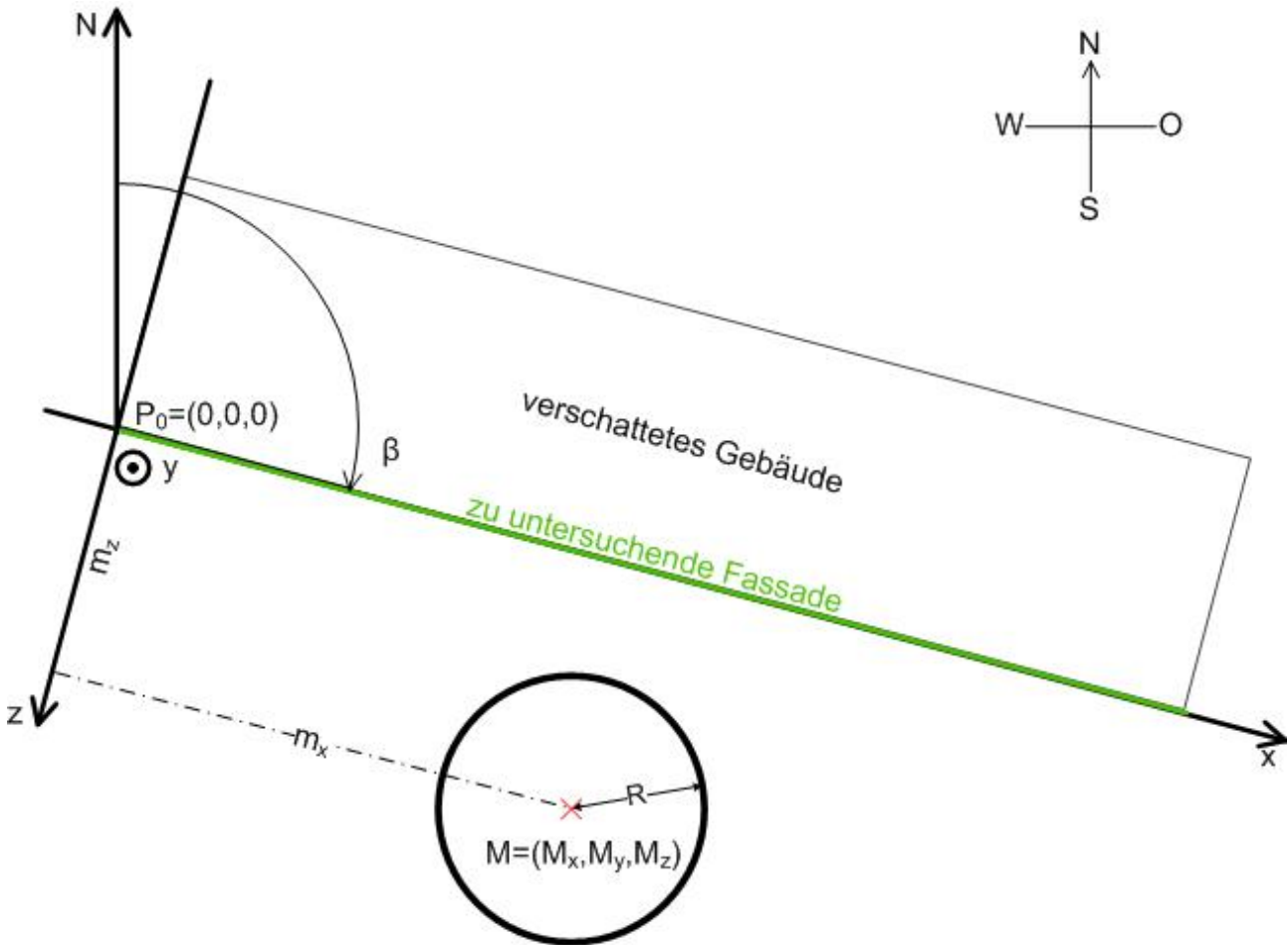
Am Nachmittag und am Abend lässt sich der Summen-Schatten allein durch die Betrachtung von S_1 und S_2 ermitteln. Es reicht also in diesem Fall S_1, S_2 und S_4 als Schattenwerfer anzugeben. Die Eingabe erfolgt dabei anhand der vier Eckpunkte bzw. deren Koordinaten in Bezug auf den Fassaden-Nullpunkt:



In dieser Skizze sind wegen der Draufsicht nur die oberen Punkte, P_2 und P_3 dargestellt. Der untere Punkt P_1 liegt unter P_2 und P_4 unter P_3 .

Die Eingabe von verschattenden Kugелеlementen erfolgt durch die Eingabe des Kugelmittelpunktes und des Radius:

Kugelelemente



Eine "Einteilung" des Kugelelementes, wie beim eckigen Gebäude, ist freilich nicht notwendig, da der Schattenwurf einer Kugel sich nur in seiner Richtung, nicht aber in seiner Größe ändert.

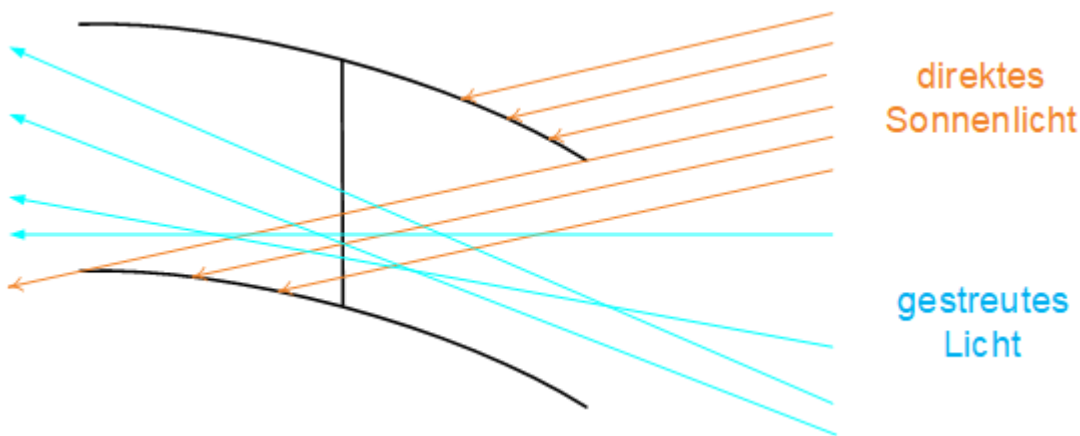
7.3 Sonnenschutz: Grundlagen und Definitionen

Direkter Einfall von Tageslicht wird von in Räumen befindlichen Personen als störend angesehen. Auf der anderen Seite jedoch empfindet der Mensch das natürliche Licht als angenehmer im Vergleich zum künstlichen Licht. Zwei Möglichkeiten des Blendschutzes sollen hier vorgestellt werden:

- Lamellennachführung
- Höhenverstellung

Lamellennachführung

Eine Jalousie mit Lamellen, welche sich nachführen lassen, bietet hier die Möglichkeit eines intelligenten Sonnenschutzes. Dabei wird die Stellung der Lamellen zyklisch dem aktuellen Sonnenstand angepasst, so dass kein direktes Tageslicht durch die Jalousien fällt, jedoch möglichst viel diffuses Tageslicht genutzt werden kann.



Die Abbildung zeigt, dass gestreutes Licht von unten noch einfallen kann, während vom direkten Tageslicht gerade nichts mehr, bzw. theoretisch nur noch ein Strahl hindurchtritt. Zur Berechnung des Lamellenwinkels sind folgende Parameter notwendig:

- die aktuelle Sonnenhöhe (Elevationswinkel)
- der Sonnenstand, d.h. der Azimut Winkel
- die Fassadenausrichtung
- die Lamellenbreite
- der Lamellenabstand

Effektiver Elevationswinkel

Wird die Jalousie im Schnitt wie oben betrachtet, so hängt der Lichteinfallswinkel nicht allein von der Sonnenhöhe (Elevation) ab, sondern auch von der Sonnenrichtung:

- Sind Fassadenausrichtung und Sonnenstand (Azimut) gleich, trifft die Sonne also direkt auf die Fassade, so ist der effektive Lichteinfallswinkel gleich dem aktuellen Elevationswinkel.
- Fällt die Sonne jedoch von der Sonnenrichtung aus gesehen schräg auf die Fassade, so ist bei gleichem Elevationswinkel der effektive Winkel größer.

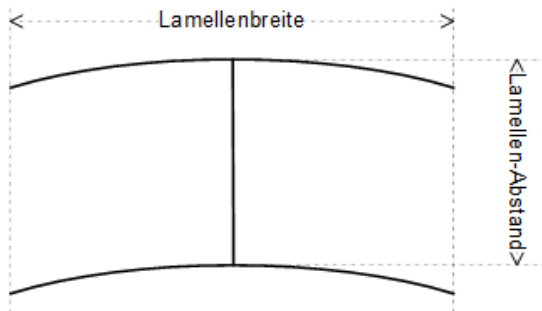
Dieser Zusammenhang kann leicht mit einem rechtwinkligen Zeichendreieck, welches hochkant auf dem Tisch steht, verdeutlicht werden: Direkt von der Seite ist ein Dreieck mit zwei 45°-Winkeln und einem 90° Winkel zu sehen. Wird das Dreieck gedreht, so wird die auf dem Tisch liegende Seite scheinbar kürzer und die beiden ehemaligen 45°-Winkel ändern sich. Das Dreieck scheint steiler zu werden.

Im Folgenden soll daher vom „effektiven Elevationswinkel“ gesprochen werden, der Teil des Lichtes, welcher direkt auf die Jalousie trifft.

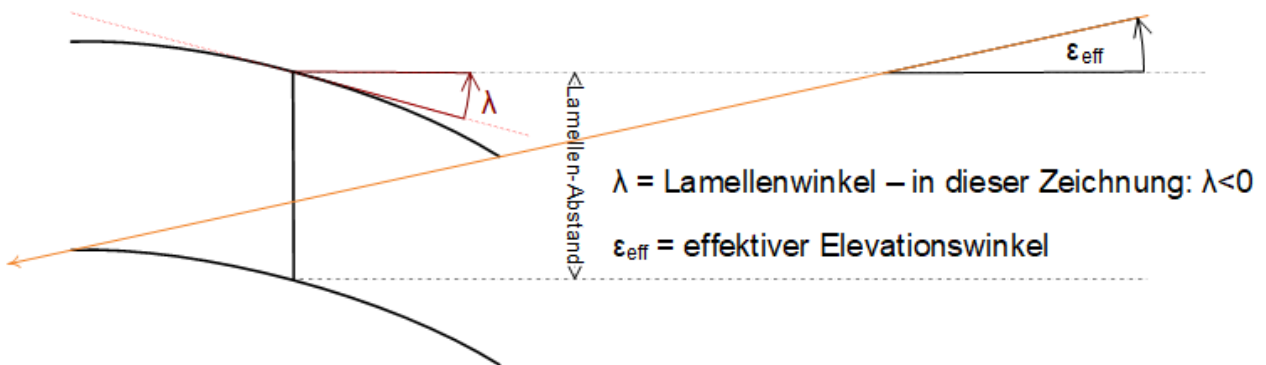
Die folgenden drei Bilder zeigen den Zusammenhang zwischen effektivem Elevationswinkel und den Jalousiemaßen und wie sich der resultierende Lamellenwinkel λ im Laufe des Tages ändert:

Lamellenwinkel

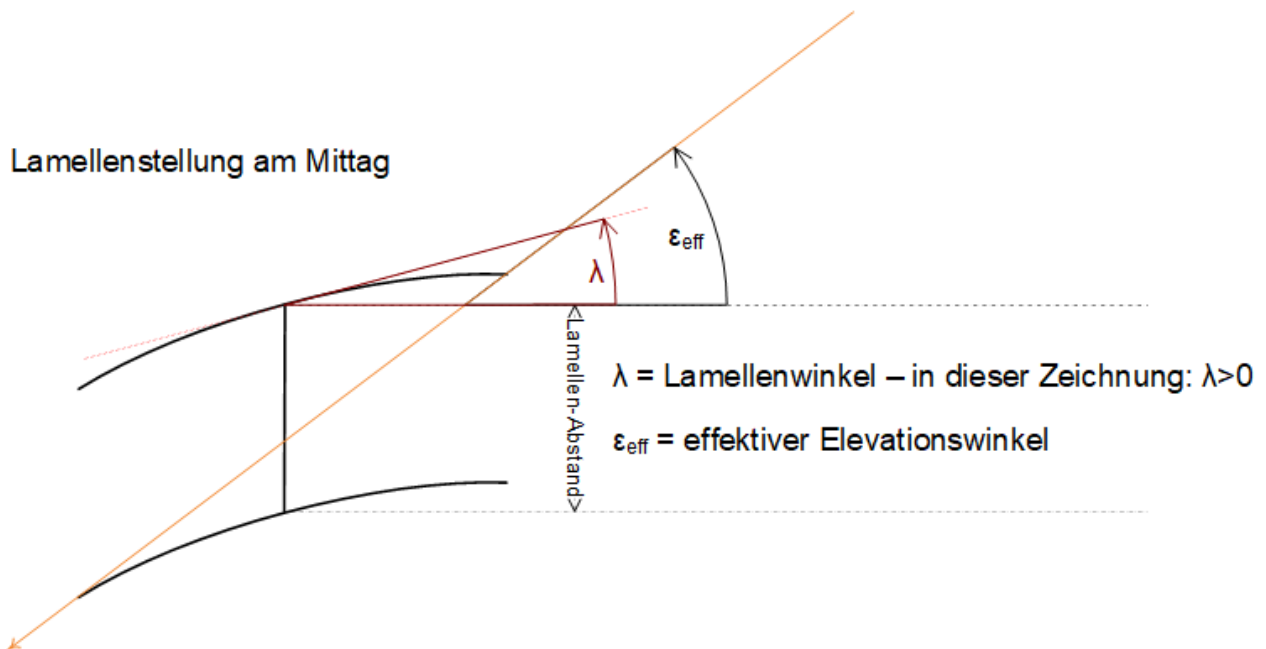
Lamellen bei einen Winkel von $\lambda=0$



Lamellenstellung am Morgen und am Abend



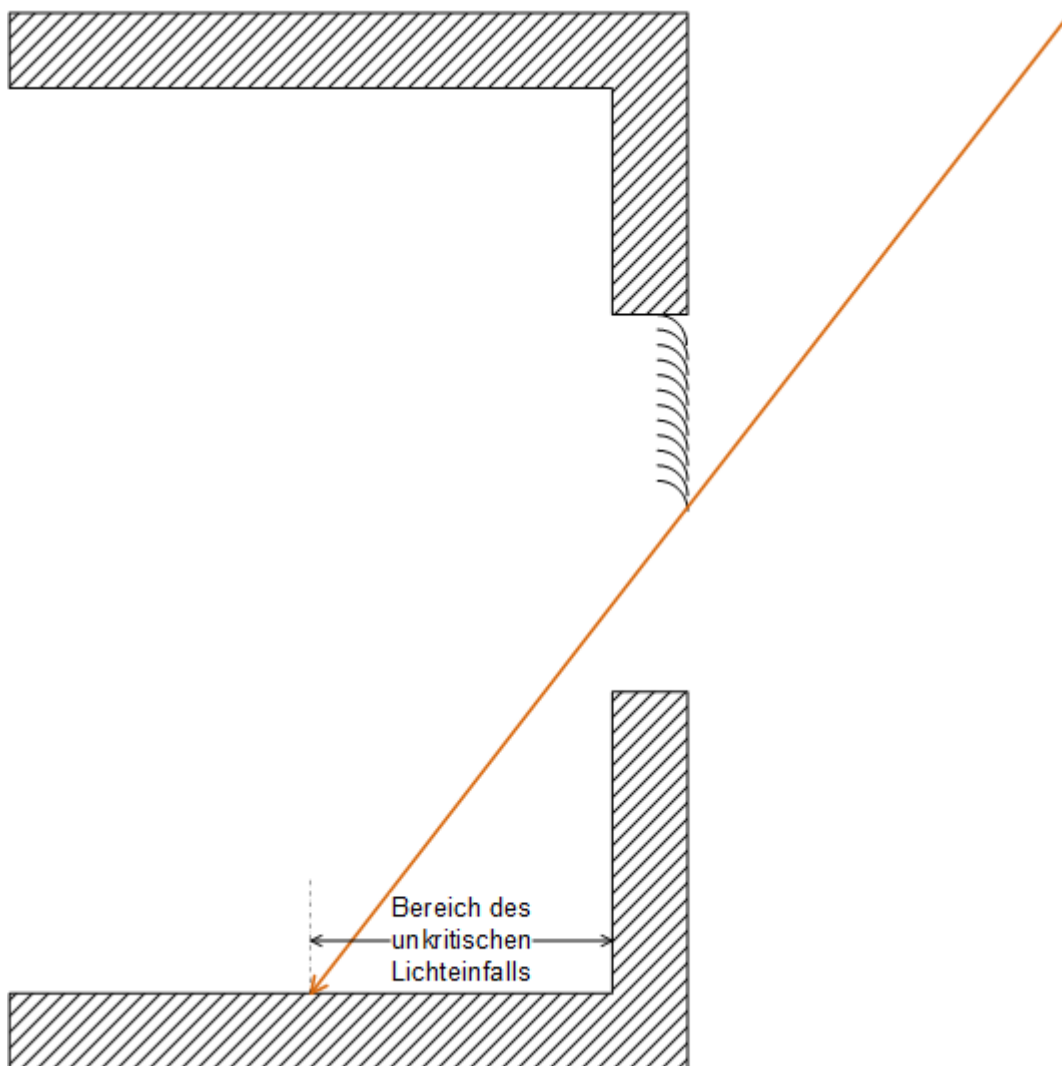
Lamellenstellung am Mittag



Tiefergehende Informationen zu dieser Thematik enthält das Kapitel Effektiver Elevationswinkel [► 35].

Höhenverstellung

Bei einem hohen Sonnenstand in der Mittagszeit dringen die direkten Sonnenstrahlen nicht in die volle Tiefe des Raumes ein. Wenn direkte Sonnenstrahlen im Bereich der Fensterbrüstung als unkritisch betrachtet werden, kann die Höhe des Sonnenschutzes automatisch so angepasst werden, dass die Sonnenstrahlen immer nur bis zu einer unkritischen Tiefe in den Raum eindringen.

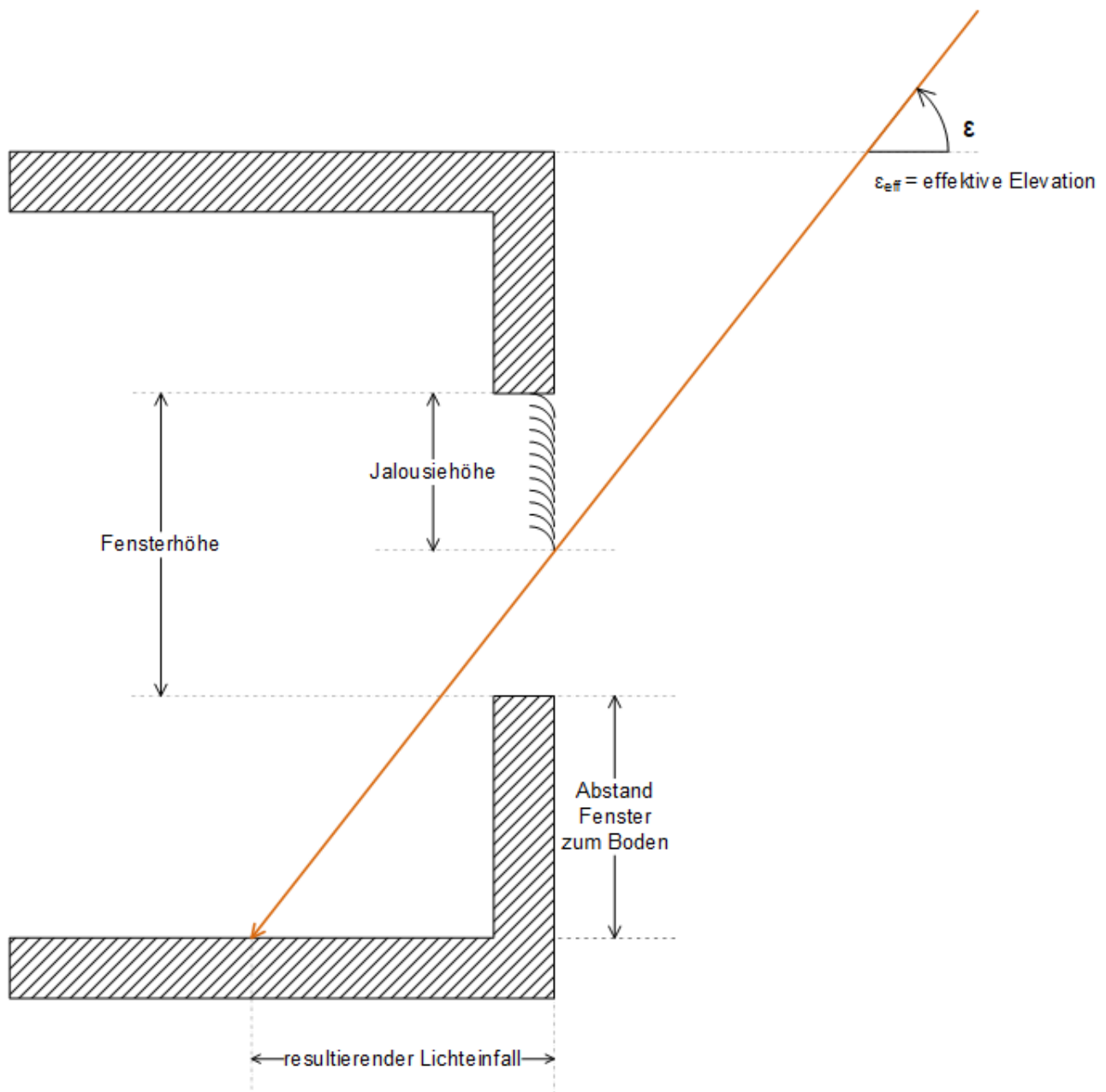


Um zu jedem Zeitpunkt die entsprechende Jalousiehöhe berechnen zu können, die sicherstellt, dass die Sonneneinstrahlung einen gewissen Wert nicht überschreitet, sind folgende Werte nötig.

Zur Berechnung der jeweiligen Jalousiehöhe erforderlich:

- Sonnenhöhe (Elevation)
- Fensterhöhe
- Abstand Fenster zum Boden

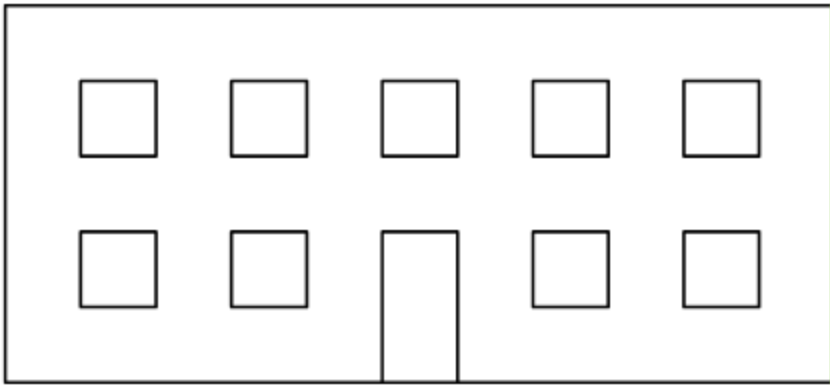
Die folgende Abbildung veranschaulicht, wo diese Parameter einzuordnen sind:



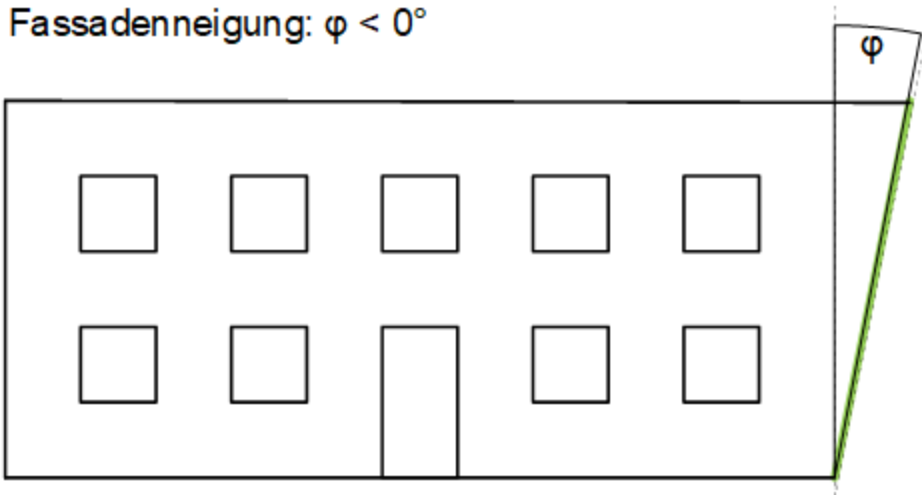
Einfluss der Fassadenneigung

Bei den beiden beschriebenen Methoden des Sonnenschutzes wurde davon ausgegangen, dass die Fassade und damit die Fenster senkrecht zum Boden stehen. Bei einer geneigten Fassade jedoch ändert sich der Lichteinfall, so dass dieser Einfluss berücksichtigt wird. Die Fassadenneigung wird wie folgt definiert:

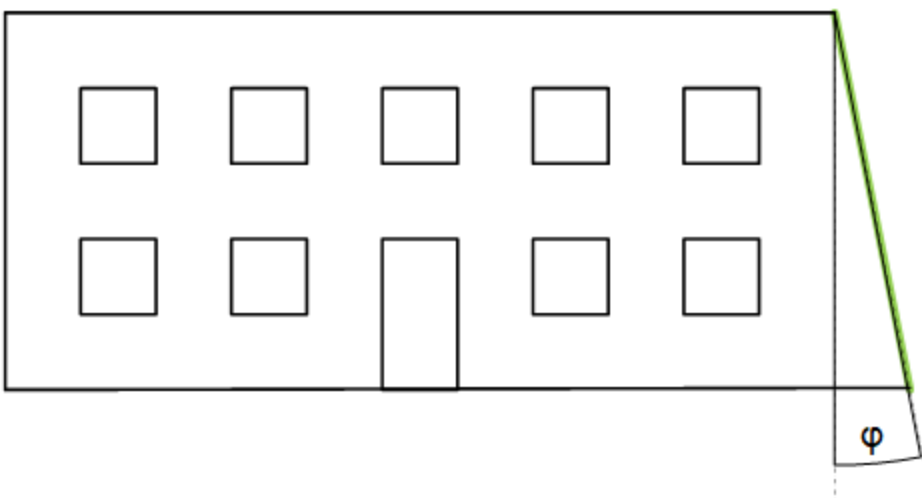
Fassadenneigung: $\varphi = 0^\circ$



Fassadenneigung: $\varphi < 0^\circ$

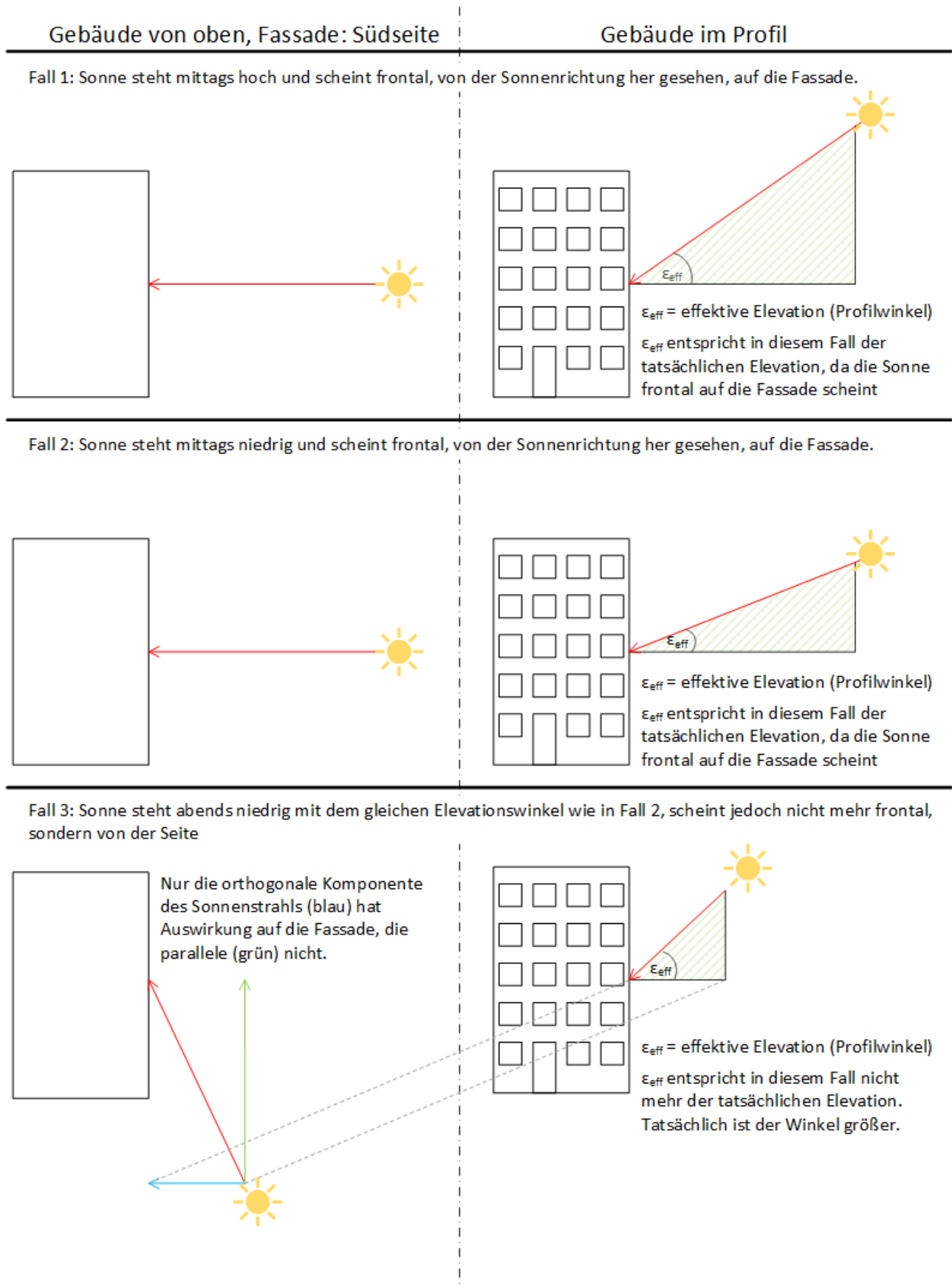


Fassadenneigung: $\varphi > 0^\circ$



7.3.1 Effektiver Elevationswinkel

Dieses Kapitel beschreibt das Verhältnis vom Sonneneinfall auf die Fassade zum Elevationswinkel.



Fall 1 zeigt einen typischen Sonneneinfall mittags auf eine Südfassade.

Die Sonne steht im Vergleich zu morgens und abends hoch.

Das grün schraffierte Dreieck stellt den Elevationswinkel ϵ dar.

Von der Sonnenrichtung (Azimut) her gesehen wirkt nur die orthogonale Komponente direkt auf die Fassade!

Daher ist in diesem Fall der Elevationswinkel ϵ gleich dem für die Berechnung effektiven Elevationswinkel ϵ_{eff} bzw. dem Profilwinkel.

Fall 2 zeigt einen typischen Sonneneinfall mittags auf eine Südfassade, jedoch niedriger.

Das grün schraffierte Dreieck stellt den Elevationswinkel ϵ dar, der im Vergleich zu Fall1 kleiner ausfällt.

Auch hier ist der Elevationswinkel ϵ gleich dem für die Berechnung effektiven Elevationswinkel ϵ_{eff} bzw. dem Profilwinkel.

Fall 3 zeigt einen typischen Sonneneinfall abends auf eine Südfassade. Der Elevationswinkel ist gleich dem in Fall 2. Damit soll Fall 3 die zeitliche Fortsetzung von Fall1 am selben Tag darstellen.

Das grün schraffierte Dreieck aus Fall 2 (Gebäude im Profil) ist nun nach vorne hin gekippt.

Es stellt jedoch die orthogonale Beleuchtungskomponente dar. Der Elevationswinkel für diese Komponente ist sichtbar größer als in Fall 2, wo die Sonne orthogonal auf die Fassade scheint.

8 TcBA PLC-Bibliothek

Durch den Einsatz der TcBA-Bibliothek sind alle SPS-Programme, von der Heizungszentrale über die Vollklimaanlage bis hin zu den Raumautomationsfunktionen, mit TwinCAT PLC Control programmierbar und stehen innerhalb der Gebäudeautomationsbibliotheken als Bausteine zur Verfügung.

Die objektorientierte Kapselung der Gebäudeautomationsfunktionen bietet folgende Vorteile:

- Schnelle Erstellung der Anlagenprogramme.
- Schnelle Parametrierung und Inbetriebnahme der Anlagen.
- Gewährleistung eines stets sehr hohen Anlagenfunktionsumfang.
- Gute Lesbarkeit der Programme (Voraussetzung für eine langjährige Wartbarkeit und Erweiterbarkeit der Anlagen).
- Gute Wiederverwendbarkeit einmal erstellter Vorlagen für Anlagen oder Anlagenbaugruppen.
- Leichte Einarbeitung des Personals.
- Leichte Erweiterung und Änderung bereits existierender Anlagen.
- Vorgaben für eine klare objektorientierte Struktur zur Erstellung von Visualisierungsobjekten in MMI- und SCADA-Systemen.
- Bessere Dokumentierbarkeit der Programme.

8.1 Allgemeine Informationen

Weitere erforderliche Bibliotheken

Für PC-Systeme (x86) und Embedded-PCs (CXxxxx):

- Standard.lib
- TcBase.lib
- TcSystem.lib
- TcUtilities.lib
- TcTestAndSet.lib
- TcBACnetRev12.lib

Für Busklemmen-Controller der Serie BCxx00, BCxx50, BCxx20, BC9191 und BXxx00:

- nicht verfügbar

● Speicherauslastung

I Durch Einbinden der Bibliothek wird bereits SPS-Programmspeicher verbraucht. Abhängig vom Applikationsprogramm kann daher der verbleibende Speicher nicht ausreichend sein.

8.2 Funktionsbausteine

BACnet

Name	Beschreibung
FB_BACnetAI1201 [► 46]	Analog-Input. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetAI1203 [► 49]	Analog-Input. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetAO1201 [► 51]	Analog-Output. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.

Name	Beschreibung
FB_BACnetAO1203 [▶ 53]	Analog-Output. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetAV1201 [▶ 59]	Analog-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetAV1202 [▶ 60]	Analog-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Mittlere Version.
FB_BACnetAV1203 [▶ 63]	Analog-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetAV1204 [▶ 67]	Analog-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetAVDisplay [▶ 69]	Analog-Value. BACnet-Objekt-generierender Funktionsbaustein welcher zur Anzeige eines Wertes aus der PLC im BACnet verwendet werden kann.
FB_BACnetAVSetpoint [▶ 70]	Analog-Value. BACnet-Objekt-generierender Funktionsbaustein welcher das BACnet Property Present Value in der PLC abbildet.
FB_BACnetBI1201 [▶ 71]	Binary-Input. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetBI1203 [▶ 72]	Binary-Input. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetBI1205 [▶ 74]	Binary-Input. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetBO1201 [▶ 76]	Binary-Output. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetBO1202 [▶ 78]	Binary-Output. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Mittlere Version.
FB_BACnetBO1203 [▶ 82]	Binary-Output. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetBO1205 [▶ 84]	Binary-Output. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetBV1201 [▶ 87]	Binary-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetBV1202 [▶ 89]	Binary-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Mittlere Version.
FB_BACnetBV1203 [▶ 91]	Binary-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetBV1204 [▶ 94]	Binary-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.

Name	Beschreibung
FB_BACnetBVDisplay [▶ 96]	Binary-Value. BACnet-Objekt-generierender Funktionsbaustein welcher zur Anzeige eines Wertes aus der PLC im BACnet verwendet werden kann.
FB_BACnetBVSetpoint [▶ 97]	Binary-Value. BACnet-Objekt-generierender Funktionsbaustein welcher das BACnet Property Present Value in der PLC abbildet.
FB_BACnetCAL1201 [▶ 98]	Kalender. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus.
FB_BACnetLoop1201 [▶ 99]	PID-Regler. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetLoop1202 [▶ 103]	PID-Regler. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetLoopSeq1201 [▶ 107]	PID-Sequenz-Regler. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetLoopSeq1202 [▶ 112]	PID-Sequenz-Regler. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetMI1203 [▶ 115]	Multistate-Input. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus.
FB_BACnetMO1202 [▶ 117]	Multistate-Output. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetMO1203 [▶ 120]	Multistate-Output. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetMV1201 [▶ 123]	Multistate-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Kleine Version.
FB_BACnetMV1202 [▶ 125]	Multistate-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Mittlere Version.
FB_BACnetMV1203 [▶ 127]	Multistate-Value. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus. Große Version.
FB_BACnetMVDisplay [▶ 130]	Multistate-Value. BACnet-Objekt-generierender Funktionsbaustein welcher zur Anzeige eines Wertes aus der PLC im BACnet verwendet werden kann.
FB_BACnetMVSetpoint [▶ 131]	Multistate-Value. BACnet-Objekt-generierender Funktionsbaustein welcher das BACnet Property Present Value in der PLC abbildet.
FB_BACnetSchedB1201 [▶ 132]	Zeitplan mit booleschen Ausgabewerten. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus.
FB_BACnetSchedBinPV [▶ 133]	Zeitplan mit booleschen Ausgabewerten vom Typ Binary Present Value mit der Funktion "Vorausrechnende Ein- und Ausschaltzeit". BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus.
FB_BACnetSchedR1201 [▶ 134]	Zeitplan mit REAL-Ausgabewerten. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus.

Name	Beschreibung
FB_BACnetSchedUdi1201 [▶ 136]	Zeitplan mit Ganzzahl-Ausgabewerten. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus.
FB_BACnetTLog1201 [▶ 137]	Trend-Log. BACnet-Objekt-generierender Funktionsbaustein zur zusätzlichen Parametrierung aus der SPS heraus.

Steuerung

Name	Beschreibung
FB_BA_Anlg3Pnt [▶ 138]	Analogwert zu Dreipunkt-Wandler
FB_BA_Cont4Stp_01 [▶ 139]	Stufenschalter mit 4 Stufen
FB_BA_RampLmt [▶ 143]	Rampenbegrenzung
FB_BA_SldgLmtMonit [▶ 145]	gleitende Grenzwertüberwachung
FB_BA_StpDly [▶ 147]	verzögerte Ausgabe von Schaltstufen
FB_BA_Swi2P [▶ 148]	Zweipunktschalter
FB_BA_Swi2P_Dly [▶ 149]	Zweipunktschalter mit zeitlicher Verzögerung.
FB_BA_SwiHys2P [▶ 152]	Zweipunktschalter um einen Schaltpunkt
FB_BA_SwiHys2P_Dly [▶ 153]	Zweipunktschalter um einen Schaltpunkt mit zeitlicher Verzögerung.
FB_BA_SwiMonit [▶ 155]	Überwachung von Endlagenschaltern, z. B. bei einer Zweipunktklappe

Regelung

Name	Beschreibung
FB_BA_FltrPT1 [▶ 156]	Filter erster Ordnung
FB_BA_PIDCtrl [▶ 157]	Universeller PID-Regler
FB_BA_PIDCtrlEx [▶ 161]	Erweiterter PID-Regler mit paralleler PID-Struktur oder vorangestelltem P-Glied , nur unter Tc2_BA verfügbar!
FB_BA_PISync1201 [▶ 164]	PI-Synchronisation von zwei Reglern des Typs FB_BACnetLoop1201 [▶ 99]
FB_BA_PISync1202 [▶ 165]	PI-Synchronisation von zwei Reglern des Typs FB_BACnetLoop1202 [▶ 103]
FB_BA_PWM [▶ 166]	Puls-Weiten-Modulations-Baustein
FB_BA_SeqCtrl [▶ 167]	Sequenzregler, einleitende Erläuterungen zu Sequenzreglern siehe hier [▶ 19].
FB_BA_SeqLink [▶ 171]	Sequenzregler-Steuerbaustein

Mathematische Funktionen

Name	Beschreibung
FB_BA_Chrct02 [▶ 173]	Linearinterpolation für 2 Stützstellen
FB_BA_Chrct04 [▶ 175]	Linearinterpolation für 4 Stützstellen
FB_BA_Chrct07 [▶ 176]	Linearinterpolation für 7 Stützstellen
FB_BA_Chrct32 [▶ 178]	Linearinterpolation für 32 Stützstellen
FB_BA_TiAvrg [▶ 180]	zeitlich arithmetischer Mittelwert

Meldungen

Name	Beschreibung
FB_BA_Alarm [▶ 182]	Alarmsammelbaustein mit zuschaltbarem Alarmspeicher und interner Quittierung
FB_BA_AlarmMgnr [▶ 184]	Funktionsbaustein sammelt die Alarmermeldungen sämtlicher Anlagen ein und bündelt diese zu einem Sammelalarm
FB_BA_AlarmPlt [▶ 186]	Funktionsbaustein sammelt die Alarmermeldungen einer Anlage ein und bündelt diese zu einem Sammelalarm
FB_BA_Alm [▶ 188]	Alarmbaustein mit zuschaltbarem Alarmspeicher und Quittierung
FB_BA_AlmColt04 [▶ 189]	Alarmsammelbaustein, 4 Alarmermeldungen
FB_BA_AlmColt08 [▶ 191]	Alarmsammelbaustein, 8 Alarmermeldungen
FB_BA_AlmColt12 [▶ 193]	Alarmsammelbaustein, 12 Alarmermeldungen
FB_BA_AlmColt16 [▶ 196]	Alarmsammelbaustein, 16 Alarmermeldungen
FB_BA_ComnMsg [▶ 200]	Bildung von Sammelmeldungen
FB_BA_ComnMsgTermt [▶ 203]	globaler Sammelbaustein von Sammelmeldungen (FB_BA_ComnMsg [▶ 200])

Operatoren

Name	Beschreibung
FB_BA_DMUX_XX [▶ 206]	Demultiplexer-Bausteine
FB_BA_MMux_XX [▶ 208]	Funktionsbaustein schaltet in Abhängigkeit eines Selektors und der entsprechenden Eingangs-Selektorbedingung einen Eingangswert auf den Ausgang
FB_BA_MultiCalc_XX [▶ 211]	Multi-Kalkulations-Bausteine
FB_BA_MUX_XX [▶ 213]	Multiplexer-Bausteine
FB_BA_PrioSwi_XX [▶ 216]	Prioritätenschalter

Anlagensteuerung

Name	Beschreibung
FB_BA_AntBlkg [▶ 218]	Blockierschutz von Pumpen oder Stellantrieben
FB_BA_DHW2P [▶ 219]	Steuerung der Ladung eines Warmwasserspeichers mittels eines Zweipunkt-Reglers
FB_BA_FIFO04 [▶ 221]	Folgesteuerung von bis zu vier Aggregaten
FB_BA_FIFO08 [▶ 222]	Folgesteuerung von bis zu acht Aggregaten
FB_BA_FnctSel [▶ 225]	Funktionsanwahl (Heizen und/oder Kühlen) in einen Zwei- oder Vierleiternetz
FB_BA_FrstPrtc [▶ 228]	Überwachung Frostalarm und Noterhitzung
FB_BA_HX [▶ 230]	Berechnung Taupunkttemperatur, spezifische Enthalpie und die absolute Feuchte
FB_BA_LglPrev [▶ 231]	Funktionsbaustein zur Desinfektion des Brauchwassers und zum Abtöten von Legionellen
FB_BA_LmtCtrl [▶ 233]	Funktionsbaustein zur Ermittlung der Grenzwerte und dessen Freigabe
FB_BA_NgtCol [▶ 236]	Sommernachtkühlung
FB_BA_RcvMonit [▶ 237]	Funktionsbaustein zur Berechnung des Wirkungsgrades einer Energie-Rückgewinnung
FB_BA_RmTAdj [▶ 240]	Anpassung des Raumtemperatursollwerts

Name	Beschreibung
FB_BA_SpRmT [▶ 243]	Anpassung des Raumtemperatursollwerts
FB_BA_SpSupVis [▶ 246]	Funktionsbaustein zur Verarbeitung und Kontrolle des unteren und oberen Sollwerts einer Zuluftfeuchte- oder Zulufttemperaturregelung
FB_BA_StepCtrl08 [▶ 249]	Schrittkettenbaustein, 8 Schritte
FB_BA_StepCtrl12 [▶ 252]	Schrittkettenbaustein, 12 Schritte

Raumfunktionen Verschattung

[Übersicht Verschattungskorrektur \[▶ 44\]](#)

[Verschattungskorrektur: Grundlagen und Definitionen \[▶ 21\]](#)

[Übersicht Sonnenschutzautomatik \[▶ 45\]](#)

Sonnenschutz: Grundlagen und Definitionen

Name	Beschreibung
FB_BA_BldPosEntry [▶ 257]	Sonnenschutzfunktion: Eingabe von Jalousiepositionen.
FB_BA_BrtnsHysDly [▶ 259]	Schwellwertschalter für Helligkeit
FB_BA_CalcSunPos [▶ 260]	Berechnung Sonnenstand
FB_BA_FcdElemEntry [▶ 261]	Verschattungskorrektur: Eingabe von Fassadenelementen per Baustein.
FB_BA_InRngAzM [▶ 265]	Überprüfung gültiger Sonnenstand Sonnenrichtungsbereich (Azimutwinkel)
FB_BA_InRngElv [▶ 268]	Überprüfung gültiger Sonnenstand Sonnenhöhenbereich (Elevationswinkel)
FB_BA_RdFcdElemLst [▶ 270]	Verschattungskorrektur: Eingabe von Fassadenelementen per Datenliste (csv).
FB_BA_RdShdObjLst [▶ 274]	Verschattungskorrektur: Eingabe von Verschattungsobjekten per Datenliste (csv).
FB_BA_RolBldActr [▶ 279]	Rollladen-Aktor
FB_BA_Shdcorr [▶ 281]	Verschattungskorrektur-Baustein
FB_BA_ShdcObjEntry [▶ 284]	Verschattungskorrektur: Eingabe von Verschattungsobjekten per Baustein.
FB_BA_SunBldActr [▶ 288]	Jalousie-Aktor
FB_BA_SunBldPosDly [▶ 293]	Einschaltverzögerung von Jalousien/Jalousiegruppen
FB_BA_SunBldEvt [▶ 295]	Ausgabe eines vorgegebenen prozentualen Jalousieposition und -stellung
FB_BA_SunBldPrioSwi4 [▶ 295]	Prioritätssteuerung, 4 Eingänge
FB_BA_SunBldPrioSwi8 [▶ 297]	Prioritätssteuerung, 8 Eingänge
FB_BA_SunBldScn [▶ 298]	Handbedienung mit Szenenanwahl und -programmierung
FB_BA_SunBldSwi [▶ 301]	Handbedienung
FB_BA_SunBldTwiLgtAuto [▶ 303]	Dämmerungsautomatik
FB_BA_SunBldWthrPrtc [▶ 304]	Witterungsschutz-Funktion
FB_BA_SunPrtc [▶ 306]	Sonnenschutzfunktion, siehe Übersicht Sonnenschutzautomatik (Verschattungskorrektur)

Raumfunktionen Beleuchtung

Name	Beschreibung
FB_BA_LgtSwi [▶ 310]	Schalt- und Dimmbaustein - Funktion für zwei Lichttaster.
FB_BA_CnstLgtCtrl [▶ 313]	Konstantlichtregelung für einen Raum.
FB_BA_AnlgLgtActr [▶ 316]	Funktionsbaustein zum Ansteuern eines analogen Lichtaktors, beispielsweise über eine KL2751.
FB_BA_DALILgtActr [▶ 318]	Funktionsbaustein zum Ansteuern eines DALI-Lichtaktors.

Spezial

Name	Beschreibung
FB_BA_Blink [▶ 322]	Einfacher Oszillator-Baustein

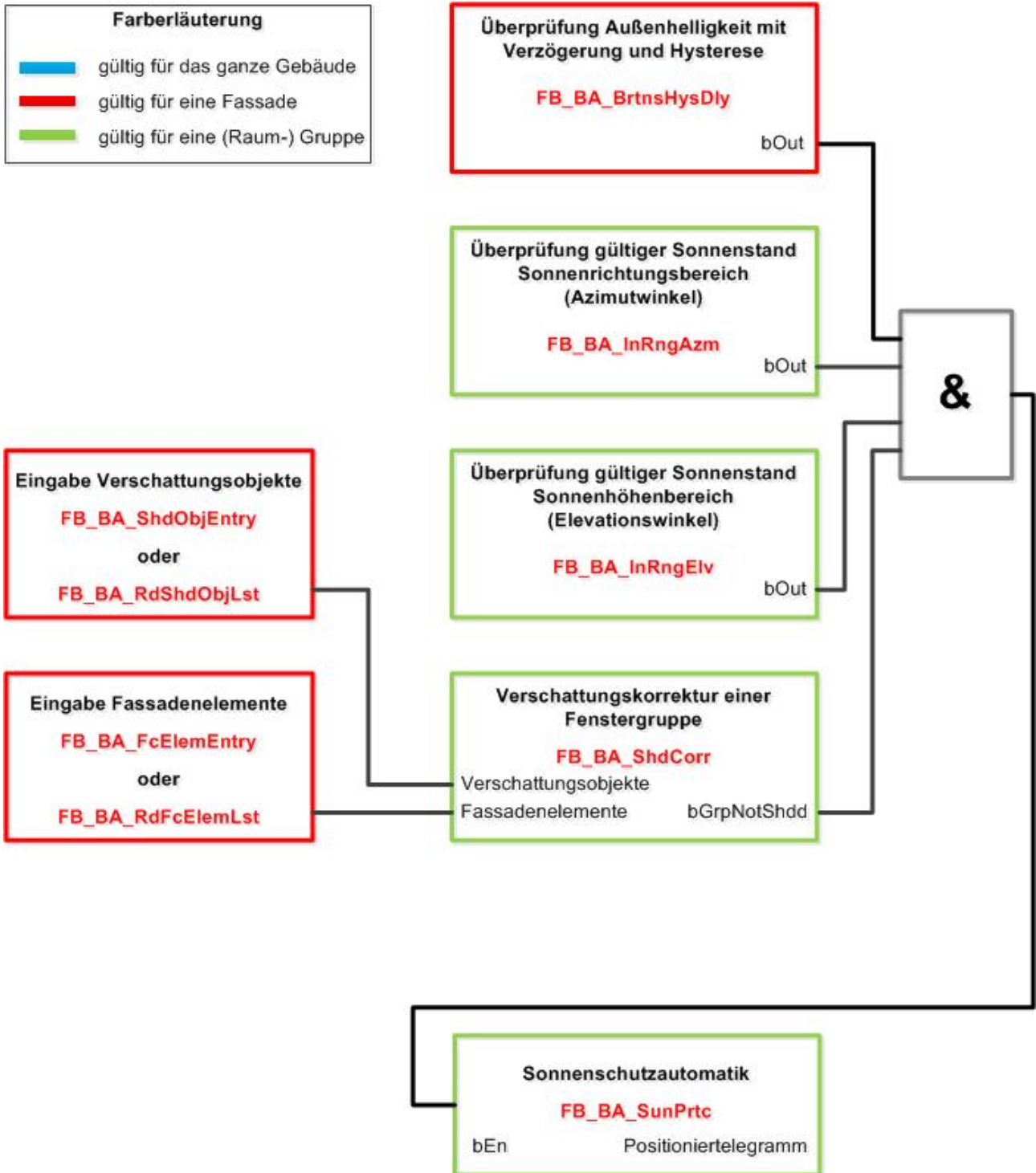
System

Name	Beschreibung
FB_BA_CnvtTiSt [▶ 323]	Umwandlung Jahr, Monat, Tag, Stunde, Minute und Sekunde in Zeitstruktur
FB_BA_ExtTiSt [▶ 324]	Umwandlung Zeitstruktur in Jahr, Monat, Tag, Stunde, Minute und Sekunde
FB_BA_GetTime [▶ 325]	Interne Uhr mit Zeitinformationen - synchronisierbar mit Systemzeit
FB_BA_SetTime [▶ 327]	Setzen der Systemzeit

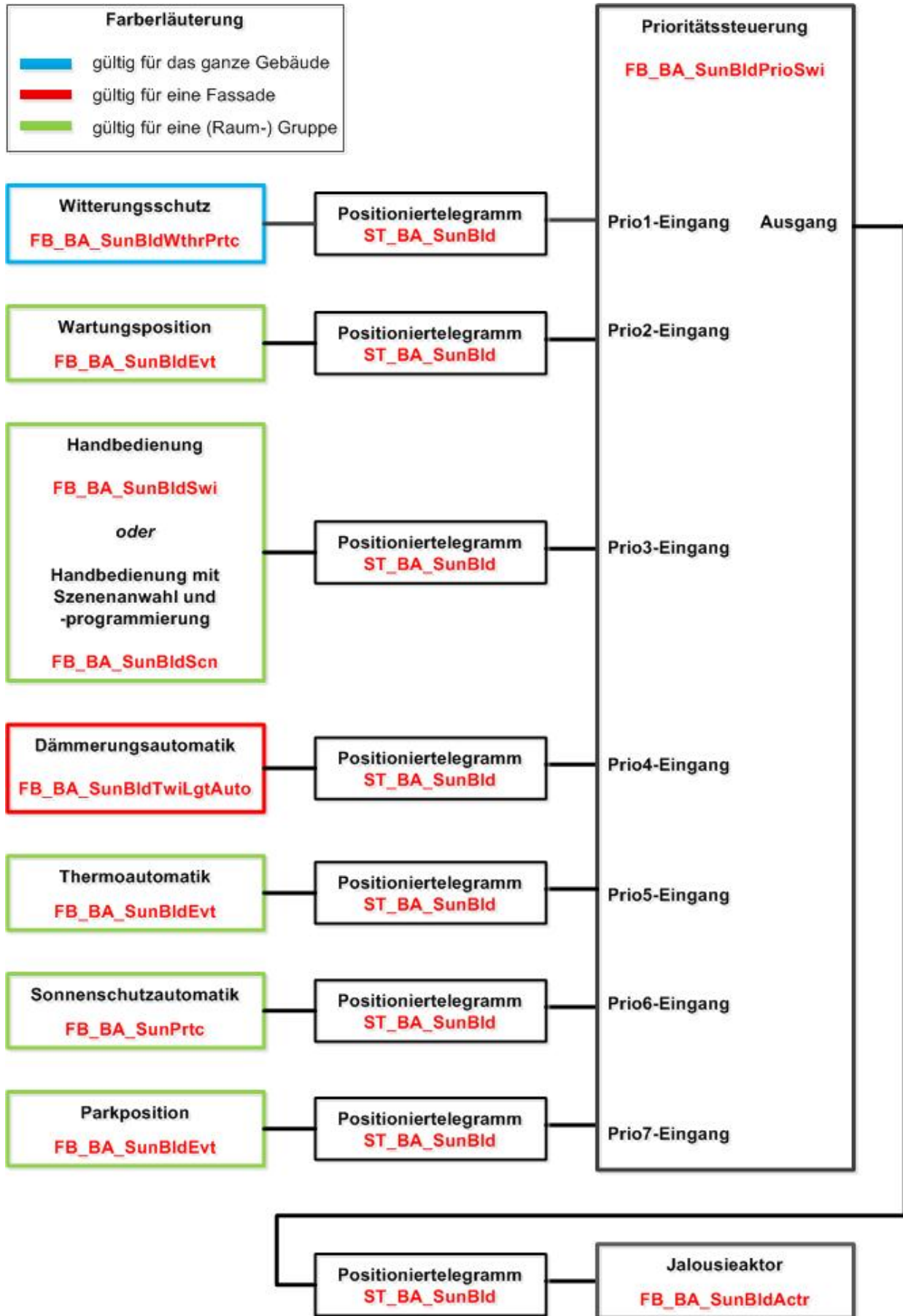
Übersicht Aufzählungen und Strukturen

Name	Beschreibung
E_BA_AlmSta [▶ 329]	Enumerator Status der Alarmmeldungen
E_BA_PosMod [▶ 330]	Enumerator zur Definition des Positioniermodus
E_BA_ShdObjType [▶ 330]	Enumerator zur Auswahl des Verschattungsobjekt-Typs
ST_BA_BldPosTab [▶ 330]	Struktur der Stützpunkteinträge für die Höhenverstellung der Jalousie
ST_BA_Cnr [▶ 331]	Information über Fenster-Eckpunkte
ST_BA_CmnMsg [▶ 331]	Struktur des Kommunikationstelegrammes an die Bausteine FB_BA_CmnMsg [▶ 200]
ST_BA_CmnMsgTermt [▶ 332]	Struktur des Kommunikationstelegrammes an die Bausteine FB_BA_CmnMsgTermt [▶ 203]
ST_BA_FcdElem [▶ 332]	Listeneintrag eines Fassadenelementes (Fenster).
ST_BA_SeqLink [▶ 333]	Struktur des Daten- und Befehlsaustausches zwischen dem Steuerbaustein FB_BA_SeqLink [▶ 171] und den Sequenzreglern FB_BA_SeqCtrl [▶ 167] .
ST_BA_ShdObj [▶ 335]	Listeneintrag eines Verschattungsobjekts
ST_BA_SpRmT [▶ 336]	Raum-Temperatursollwerte
ST_BA_SunBld [▶ 337]	Struktur des Jalousie-Positioniertelegramms
ST_BA_SunBldScn [▶ 337]	Tabelleneintrag einer Jalousie-Szene

8.2.1 Verschattungskorrektur



8.2.2 Sonnenschutzautomatik



8.2.3 Liste der Verschattungselemente

Die Daten aller Verschattungsobjekte (Gebäudeteile, Bäume, etc.) pro Fassade werden innerhalb des Programms in einem Feld von Strukturelementen des Typs `ST_BA_ShObj` [▶ 335] hinterlegt.

Die Verschattungskorrektur `FB_BA_ShCorr` [▶ 281] liest die Informationen dieser Liste zwar nur, der Verwaltungsbaustein `FB_BA_ShObjEntry` [▶ 284] jedoch hat liest und beschreibt sie als Ein-/Ausgangsvariable.

Es empfiehlt sich daher, diese Liste global zu deklarieren:

```
VAR_GLOBAL
    arrShdObj : ARRAY[1..gBA_cMaxShdObj] OF ST_BA_ShObj;
END_VAR
```

Die Variable `gBA_cMaxShdObj` stellt dabei die Obergrenze der zur Verfügung stehenden Elemente dar und ist global als Konstante innerhalb der Programmbibliothek deklariert:

```
VAR_GLOBAL CONSTANT
    gBA_cMaxShdObj : INT := 20;
END_VAR
```

8.2.4 Liste der Fassadenelemente

Die Daten aller Fenster (Fassadenelemente) pro Fassade werden innerhalb des Programms in einem Feld von Strukturelementen des Typs `ST_BA_FcdElem` [▶ 332] hinterlegt.

Der Verwaltungsbaustein `FB_BA_FcdElemEntry` [▶ 261] sowie die Verschattungskorrektur `FB_BA_ShCorr` [▶ 281] lesen und beschreiben diese Liste (letzterer setzt die Verschattungsinformation) und greifen somit als Ein-/Ausgangsvariable auf dieses Feld zu.

Es empfiehlt sich daher, diese Liste global zu deklarieren:

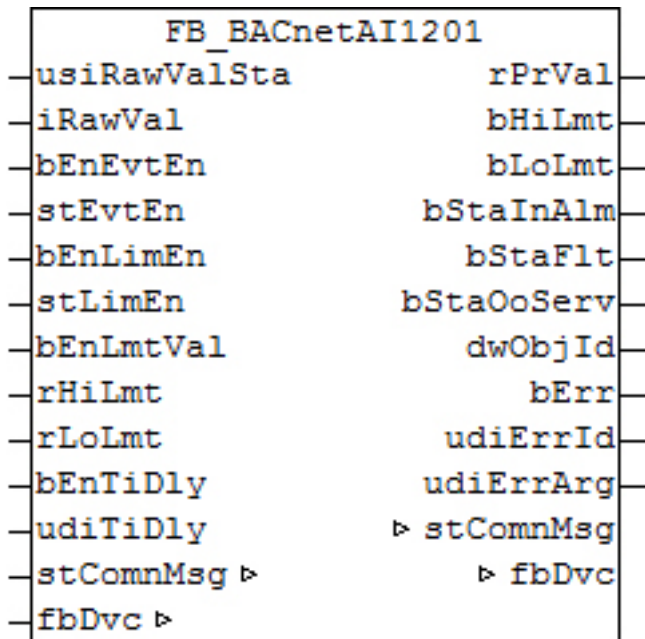
```
VAR_GLOBAL
    arrFcdElem : ARRAY[1..gBA_cMaxColumnFcd, 1..gBA_cMaxRowFcd] OF ST_BA_FcdElem;
END_VAR
```

Die Variablen `gBA_cMaxColumnFcd` und `gBA_cMaxRowFcd` definieren dabei die Obergrenzen der zur Verfügung stehenden Elemente und sind global als Konstante innerhalb der Programmbibliothek deklariert:

```
VAR_GLOBAL CONSTANT
    gBA_cMaxRowFcd : INT :=10;
    gBA_cMaxColumnFcd : INT :=20;
END_VAR
```

8.2.5 FB_BACnetAI1201

BACnet Analog-Input



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Input-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung. Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleine" Version. Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAI1203 \[▶ 49\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnLimEn	Freigabe zum Beschreiben des Bitmusters der Grenzwert-Freigaben	
stLimEn	Property-Wert Bitmuster Grenzwert-Freigaben	Limit-Enable (52)
bEnLmtVal	Freigabe zum Beschreiben der Grenzwerte	
rHiLmt	Property-Wert HighLimit	HighLimit (45)
rLoLmt	Property-Wert LowLimit	LowLimit (59)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)

Ein-Ausgänge

VAR_INPUT

```

usiRawValSta : USINT;
iRawVal      : INT;
bEnEvtEn    : BOOL;
stEvtEn     : ST_BACnet_EventTransitionBits;
bEnLimEn    : BOOL;
stLimEn     : ST_BACnet_LimitEnable;
  
```

```

bEnLmtVal    : BOOL;
rHiLmt      : REAL;
rLoLmt      : REAL;
bEnTiDly    : BOOL;
udiTiDly    : UDINT;

```

usiRawValSta: Eingang zur Verbindung mit dem Statusbyte (State) der Klemme

iRawVal: Eingang zur Verbindung mit dem Datenwort (Data In) der Klemme

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnLimEn / stLimEn: Freigabe/Property-Wert Bitmuster LimitEnable

bEnLimVal: Freigabe Beschreiben High- und Low-Limit

rHiLmt: Property-Wert High-Limit

rLoLmt: Property-Wert Low-Limit

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

VAR_OUTPUT

```

rPrVal      : REAL;
bHiLmt      : BOOL;
bLoLmt      : BOOL;
bStaInAlm   : BOOL;
bStaFlt     : BOOL;
bStaOoServ  : BOOL;
dwObjId     : DWORD;
bErrs       : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;

```

rPrVal: Aktueller Wert des Analog-Input-Objekts - direkt aus dem BACnet eingelesen

bHiLim: Meldung oberer Grenzwert erreicht

bLoLim: Meldung unterer Grenzwert erreicht

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Analog-Input-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Analog-Input-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Analog-Input-Objekts an.

dwObjId: BACnet-Objekt-ID des Analog-Input-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```

stComnMsg   : ST_BA_ComnMsg;
fbDvc       : FB_BACnet_Device;

```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_ComMsg](#) [► 200].

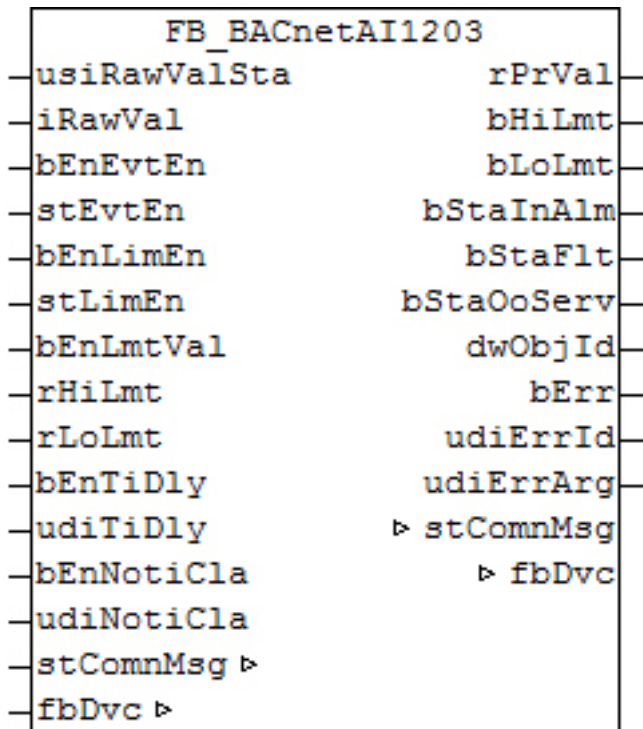
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.6 FB_BACnetAI1203

BACnet Analog-Input



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Input-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "große" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAI1201](#) [► 46]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnLimEn	Freigabe zum Beschreiben des Bitmusters der Grenzwert-Freigaben	
stLimEn	Property-Wert Bitmuster Grenzwert-Freigaben	Limit-Enable (52)
bEnLmtVal	Freigabe zum Beschreiben der Grenzwerte	
rHiLmt	Property-Wert HighLimit	HighLimit (45)
rLoLmt	Property-Wert LowLimit	LowLimit (59)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

usiRawValSta : USINT;
iRawVal      : INT;
bEnEvtEn    : BOOL;
stEvtEn     : ST_BACnet_EventTransitionBits;
bEnLimEn    : BOOL;
stLimEn     : ST_BACnet_LimitEnable;
bEnLmtVal   : BOOL;
rHiLmt      : REAL;
rLoLmt      : REAL;
bEnTiDly    : BOOL;
udiTiDly    : UDINT;
bEnNotiCla  : BOOL;
udiNotiCla  : UDINT;

```

usiRawValSta: Eingang zur Verbindung mit dem Statusbyte (State) der Klemme

iRawVal: Eingang zur Verbindung mit dem Datenwort (Data In) der Klemme

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnLimEn / stLimEn: Freigabe/Property-Wert Bitmuster LimitEnable

bEnLimVal: Freigabe Beschreiben High- und Low-Limit

rHiLmt: Property-Wert High-Limit

rLoLmt: Property-Wert Low-Limit

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

rPrVal      : REAL;
bHiLmt      : BOOL;
bLoLmt      : BOOL;
bStaInAlm   : BOOL;
bStaFlt     : BOOL;
bStaOoServ  : BOOL;
dwObjId     : DWORD;
bErrs       : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;

```

rPrVal: Aktueller Wert des Analog-Input-Objekts - direkt aus dem BACnet eingelesen

bHiLim: Meldung oberer Grenzwert erreicht

bLoLim: Meldung unterer Grenzwert erreicht

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Analog-Input-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Analog-Input-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Analog-Input-Objekts an.

dwObjId: BACnet-Objekt-ID des Analog-Input-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc      : FB_BACnet_Device;
```

stComnMsg: Referenz auf die Verbindungsstruktur [[▶ 331](#)] zum Melde-Sammelbaustein FB_BA_ComMsg [[▶ 200](#)].

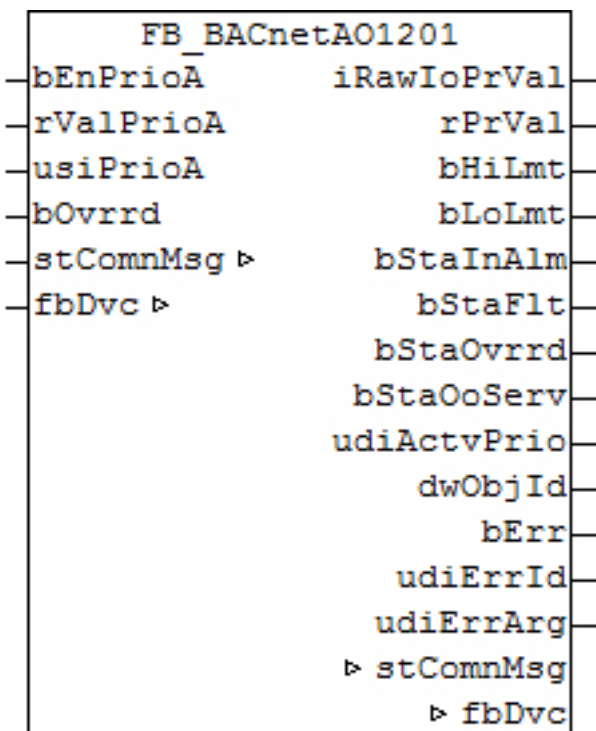
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.7 FB_BACnetAO1201

BACnet Analog-Output



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleine" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAO1203](#) [[▶ 53](#)]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
rValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

Ein-Ausgänge

VAR_INPUT

```
bEnPrioA : BOOL;
rValPrioA : REAL;
usiPrioA : USINT;
bOvrrd : BOOL;
```

bEnPrioA: Freigabe zum Beschreiben

rValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bOvrrd: An diesem Eingang kann das Eingangssignal einer lokale Vorrangbedienung z. B. von einem KM4602 angelegt werden. Steht dieser Eingang dann auf "TRUE", so wird das im BACnet angezeigt.

VAR_OUTPUT

```
iRawIoPrVal : INT;
rPrVal : REAL;
bHiLmt : BOOL;
bLoLmt : BOOL;
bStaInAlm : BOOL;
bStaFlt : BOOL;
bStaOvrrd : BOOL;
bStaOoServ : BOOL;
udiActvPrio : UDINT;
dwObjId : DWORD;
bErrs : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

iRawIoPrVal: Ausgang zur Verbindung mit dem Datenwort (Data Out) der analogen Ausgangs-Klemme

rPrVal: Aktueller Wert des Analog-Output-Objekts - direkt aus dem BACnet eingelesen

bHiLim: Meldung oberer Grenzwert erreicht

bLoLim: Meldung unterer Grenzwert erreicht

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Analog-Output-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Analog-Output-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Overridden“ des Analog-Output-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Analog-Output-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Analog-Output-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc : FB_BACnet_Device;
```

stCommMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_ComMsg \[► 200\]](#)

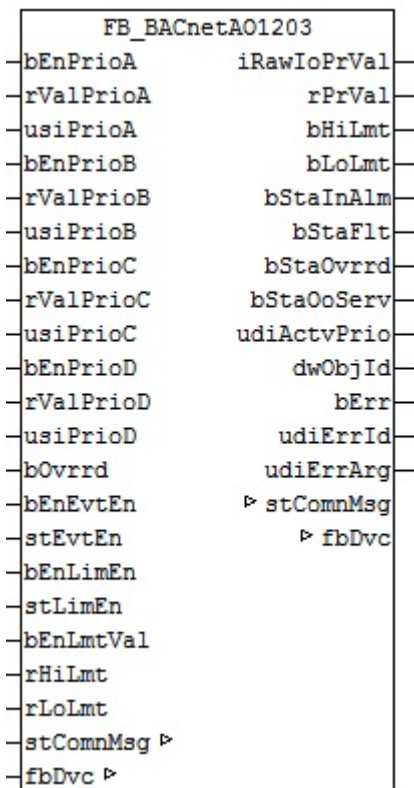
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.8 FB_BACnetAO1203

BACnet Analog-Output



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "große" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAO1201 \[► 51\]](#)
- [FB_BACnetAO1205 \[► 56\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
rValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
rValPrioB	Zu schreibender Wert	Priority-Array (87)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
rValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
rValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnLimEn	Freigabe zum Beschreiben des Bitmusters der Grenzwert-Freigaben	
stLimEn	Property-Wert Bitmuster Grenzwert-Freigaben	Limit-Enable (52)
bEnLmtVal	Freigabe zum Beschreiben der Grenzwerte	
rHiLmt	Property-Wert HighLimit	HighLimit (45)
rLoLmt	Property-Wert LowLimit	LowLimit (59)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA   : BOOL;
rValPrioA  : REAL;
usiPrioA   : USINT;
bEnPrioB   : BOOL;
rValPrioB  : REAL;
usiPrioB   : USINT;
bEnPrioC   : BOOL;
rValPrioC  : REAL;
usiPrioC   : USINT;
bEnPrioD   : BOOL;
rValPrioD  : REAL;
usiPrioD   : USINT;
bOvrrd    : BOOL;
bEnEvtEn   : BOOL;
stEvtEn    : ST_BACnet_EventTransitionBits;
bEnLimEn   : BOOL;
stLimEn    : ST_BACnet_LimitEnable;
bEnLmtVal  : BOOL;
rHiLmt     : REAL;
rLoLmt     : REAL;

```

bEnPrioA: Freigabe zum Beschreiben

rValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

rValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

rValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioD: Freigabe zum Beschreiben

rValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bOvrrd: An diesem Eingang kann das Eingangssignal einer lokale Vorrangbedienung z. B. von einem KM4602 angelegt werden. Steht dieser Eingang dann auf "TRUE", so wird das im BACnet angezeigt.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnLimEn / stLimEn: Freigabe/Property-Wert Bitmuster LimitEnable

bEnLimVal: Freigabe Beschreiben High- und Low-Limit

rHiLmt: Property-Wert High-Limit

rLoLmt: Property-Wert Low-Limit

VAR_OUTPUT

```
iRawIoPrVal : INT;
rPrVal      : REAL;
bHiLmt      : BOOL;
bLoLmt      : BOOL;
bStaInAlm   : BOOL;
bStaFlt     : BOOL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
udiActvPrio : UDINT;
dwObjId     : DWORD;
bErrs       : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

iRawIoPrVal: Ausgang zur Verbindung mit dem Datenwort (Data Out) der analogen Ausgangs-Klemme.

rPrVal: Aktueller Wert des Analog-Output-Objekts - direkt aus dem BACnet eingelesen

bHiLim: Meldung oberer Grenzwert erreicht

bLoLim: Meldung unterer Grenzwert erreicht

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Analog-Output-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Analog-Output-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Overridden“ des Analog-Output-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Analog-Output-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Analog-Output-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc      : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_ComMsg \[► 200\]](#).

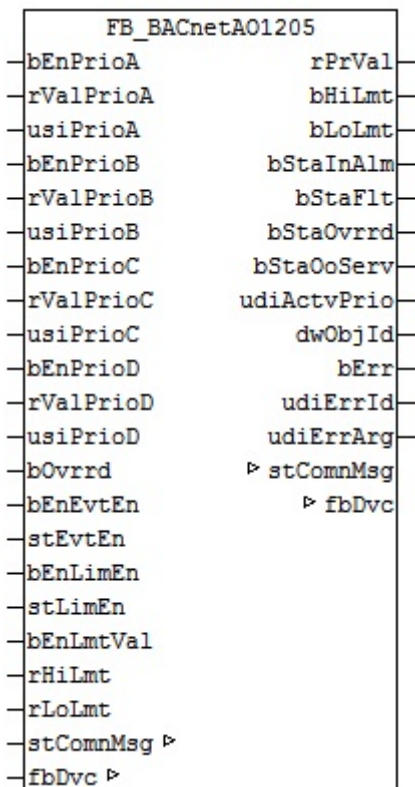
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.9 FB_BACnetAO1205

BACnet Analog-Output

**Funktionsbeschreibung**

Dieser Baustein generiert ein BACnet-Analog-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "große" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAO1201 \[► 51\]](#)
- [FB_BACnetAO1203 \[► 53\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt

(steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
rValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
rValPrioB	Zu schreibender Wert	Priority-Array (87)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
rValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
rValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnLimEn	Freigabe zum Beschreiben des Bitmusters der Grenzwert-Freigaben	
stLimEn	Property-Wert Bitmuster Grenzwert-Freigaben	Limit-Enable (52)
bEnLmtVal	Freigabe zum Beschreiben der Grenzwerte	
rHiLmt	Property-Wert HighLimit	HighLimit (45)
rLoLmt	Property-Wert LowLimit	LowLimit (59)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
rValPrioA     : REAL;
usiPrioA      : USINT;
bEnPrioB      : BOOL;
rValPrioB     : REAL;
usiPrioB      : USINT;
bEnPrioC      : BOOL;
rValPrioC     : REAL;
usiPrioC      : USINT;
bEnPrioD      : BOOL;
rValPrioD     : REAL;
usiPrioD      : USINT;
bOvrrd       : BOOL;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
bEnLimEn      : BOOL;
stLimEn       : ST_BACnet_LimitEnable;
bEnLmtVal     : BOOL;
rHiLmt        : REAL;
rLoLmt        : REAL;
    
```

bEnPrioA: Freigabe zum Beschreiben

rValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

rValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

rValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioD: Freigabe zum Beschreiben

rValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bOvrrd: An diesem Eingang kann das Eingangssignal einer lokale Vorrangbedienung z. B. von einem KM4602 angelegt werden. Steht dieser Eingang dann auf "TRUE", so wird das im BACnet angezeigt.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnLimEn / stLimEn: Freigabe/Property-Wert Bitmuster LimitEnable

bEnLimVal: Freigabe Beschreiben High- und Low-Limit

rHiLmt: Property-Wert High-Limit

rLoLmt: Property-Wert Low-Limit

VAR_OUTPUT

```
rPrVal      : REAL;
bHiLmt      : BOOL;
bLoLmt      : BOOL;
bStainAlm   : BOOL;
bStaFlt     : BOOL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
udiActvPrio : UDINT;
dwObjId     : DWORD;
bErrs       : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

rPrVal: Aktueller Wert des Analog-Output-Objekts - direkt aus dem BACnet eingelesen

bHiLim: Meldung oberer Grenzwert erreicht

bLoLim: Meldung unterer Grenzwert erreicht

bStainAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Analog-Output-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Analog-Output-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Analog-Output-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Analog-Output-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Analog-Output-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc      : FB_BACnet_Device;
```

stComnMsg: Referenz auf die Verbindungsstruktur [[▶ 331](#)] zum Melde-Sammelbaustein FB_BA_ComMsg [[▶ 200](#)].

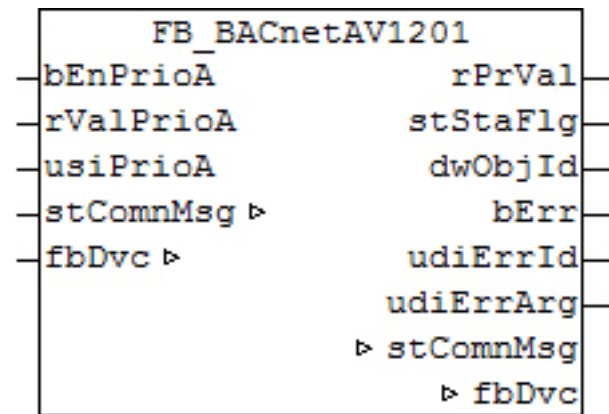
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.10 FB_BACnetAV1201

BACnet Analog-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleinste" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAV1202](#) [[▶ 60](#)]
- [FB_BACnetAV1203](#) [[▶ 63](#)]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
rValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

Ein-Ausgänge

VAR_INPUT

```
bEnPrioA : BOOL;
rValPrioA : REAL;
usiPrioA : USINT;
```

bEnPrioA: Freigabe zum Beschreiben

rValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

VAR_OUTPUT

```
rPrVal : REAL;
stStaFlg : ST_BACnet_StatusFlags;
dwObjId : DWORD;
bErr : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

rPrVal: Aktueller Wert des Analog-Value-Objekts - direkt aus dem BACnet eingelesen

stStaFlg: Ausgabestruktur des BACnet-Status

dwObjId: BACnet-Objekt-ID des Analog-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_CmnMsg \[► 200\]](#).

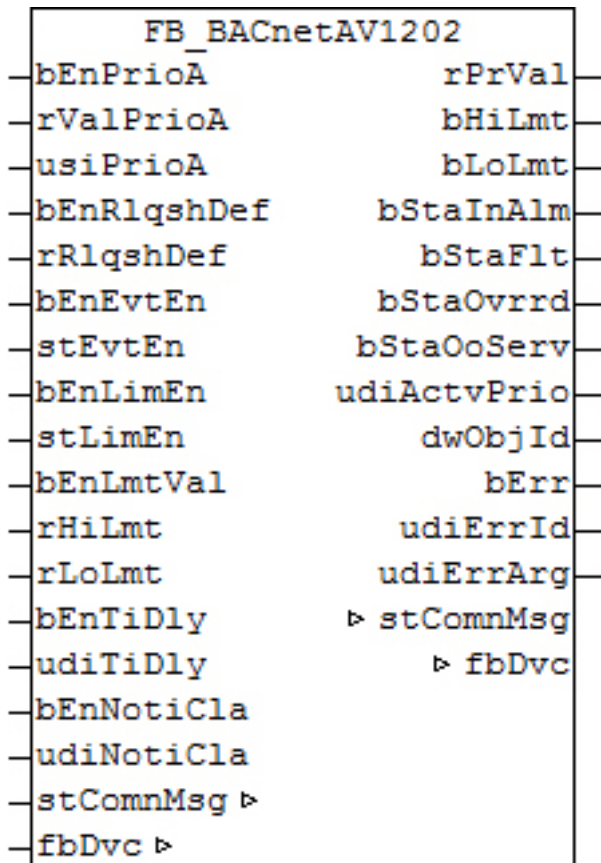
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.11 FB_BACnetAV1202

BACnet Analog-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "mittlere" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAV1201 \[▶ 59\]](#)
- [FB_BACnetAV1203 \[▶ 63\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
rValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnRlqshDef	Freigabe zum Beschreiben des Relinquish-Defaults	
rRlqshDef	Zu schreibender Wert auf das Property Relinquish-Default	Relinquish-Default (104)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnLimEn	Freigabe zum Beschreiben des Bitmusters der Grenzwert-Freigaben	
stLimEn	Property-Wert Bitmuster Grenzwert-Freigaben	Limit-Enable (52)
bEnLmtVal	Freigabe zum Beschreiben der Grenzwerte	
rHiLmt	Property-Wert HighLimit	HighLimit (45)
rLoLmt	Property-Wert LowLimit	LowLimit (59)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
rValPrioA     : REAL;
usiPrioA      : USINT;
bEnRlqshDef   : BOOL;
rRlqshDef     : REAL;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
bEnLimEn      : BOOL;
stLimEn       : ST_BACnet_LimitEnable;
bEnLmtVal     : BOOL;
rHiLmt        : REAL;
rLoLmt        : REAL;
bEnTiDly      : BOOL;
udiTiDly      : UDINT;
bEnNotiCla    : BOOL;
udiNotiCla    : UDINT;

```

bEnPrioA: Freigabe zum Beschreiben

rValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnRlqshDef / rRlqshDef: Freigabe/Property-Wert Relinquish-Default

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnLimEn / stLimEn: Freigabe/Property-Wert Bitmuster LimitEnable

bEnLimVal: Freigabe Beschreiben High- und Low-Limit

rHiLmt: Property-Wert High-Limit

rLoLmt: Property-Wert Low-Limit

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

rPrVal        : REAL;
bHiLmt        : BOOL;
bLoLmt        : BOOL;
bStaInAlm     : BOOL;
bStaFlt       : BOOL;
bStaOvrrd     : BOOL;
bStaOoServ    : BOOL;
udiActvPrio   : UDINT;
dwObjId       : DWORD;
bErrs         : BOOL;
udiErrId      : UDINT;
udiErrArg     : UDINT;

```

rPrVal: Aktueller Wert des Analog-Value-Objekts - direkt aus dem BACnet eingelesen

bHiLim: Meldung oberer Grenzwert erreicht

bLoLim: Meldung unterer Grenzwert erreicht

bStalnAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Analog-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Analog-Value-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Analog-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Analog-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Analog-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [[▶ 340](#)].

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [[▶ 331](#)] zum Melde-Sammelbaustein [FB_BA_CmnMsg](#) [[▶ 200](#)].

fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.12 FB_BACnetAV1203

BACnet Analog-Value

FB_BACnetAV1203	
– bEnPrioA	rPrVal
– rValPrioA	bHiLmt
– usiPrioA	bLoLmt
– bEnPrioB	bStaInAlm
– rValPrioB	bStaFlt
– usiPrioB	bStaOvrrd
– bEnPrioC	bStaOoSrv
– rValPrioC	udiActvPrio
– usiPrioC	dwObjId
– bEnPrioD	bErr
– rValPrioD	udiErrId
– usiPrioD	udiErrArg
– bEnRlqshDef	▷ stCommMsg
– rRlqshDef	▷ fbDvc
– bEnEvtEn	
– stEvtEn	
– bEnLimEn	
– stLimEn	
– bEnLmtVal	
– rHiLmt	
– rLoLmt	
– bEnTiDly	
– udiTiDly	
– bEnNotiCla	
– udiNotiCla	
– stCommMsg ▷	
– fbDvc ▷	

Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "größte" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAV1201 \[▶ 59\]](#)
- [FB_BACnetAV1202 \[▶ 60\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
rValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
rValPrioB	Zu schreibender Wert	Priority-Array (87)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
rValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
rValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnRlqshDef	Freigabe zum Beschreiben des Relinquish-Defaults	
rRlqshDef	Zu schreibender Wert auf das Property Relinquish-Default	Relinquish-Default (104)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnLimEn	Freigabe zum Beschreiben des Bitmusters der Grenzwert-Freigaben	
stLimEn	Property-Wert Bitmuster Grenzwert-Freigaben	Limit-Enable (52)
bEnLmtVal	Freigabe zum Beschreiben der Grenzwerte	
rHiLmt	Property-Wert HighLimit	HighLimit (45)
rLoLmt	Property-Wert LowLimit	LowLimit (59)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
rValPrioA     : REAL;
usiPrioA      : USINT;
bEnPrioB      : BOOL;
rValPrioB     : REAL;
usiPrioB      : USINT;
bEnPrioC      : BOOL;
rValPrioC     : REAL;
usiPrioC      : USINT;
bEnPrioD      : BOOL;
rValPrioD     : REAL;
usiPrioD      : USINT;
bEnRlqshDef   : BOOL;
rRlqshDef     : REAL;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
bEnLimEn      : BOOL;
    
```

```

stLimEn      : ST_BACnet_LimitEnable;
bEnLmtVal   : BOOL;
rHiLmt      : REAL;
rLoLmt      : REAL;
bEnTiDly    : BOOL;
udiTiDly    : UDINT;
bEnNotiCla  : BOOL;
udiNotiCla  : UDINT;

```

bEnPrioA: Freigabe zum Beschreiben

rValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

rValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

rValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioD: Freigabe zum Beschreiben

rValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bEnRlqshDef / rRlqshDef : Freigabe/Property-Wert Relinquish-Default

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnLimEn / stLimEn: Freigabe/Property-Wert Bitmuster LimitEnable

bEnLimVal: Freigabe Beschreiben High- und Low-Limit

rHiLmt: Property-Wert High-Limit

rLoLmt: Property-Wert Low-Limit

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

rPrVal      : REAL;
bHiLmt      : BOOL;
bLoLmt      : BOOL;
bStaInAlm   : BOOL;
bStaFlt     : BOOL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
udiActvPrio : UDINT;
dwObjId     : DWORD;
bErrs       : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;

```

rPrVal: Aktueller Wert des Analog-Value-Objekts - direkt aus dem BACnet eingelesen

bHiLim: Meldung oberer Grenzwert erreicht

bLoLim: Meldung unterer Grenzwert erreicht

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Analog-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Analog-Value-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Analog-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Analog-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Analog-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [[▶ 340](#)].

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc      : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [[▶ 331](#)] zum Melde-Sammelbaustein [FB_BA_ComMsg](#) [[▶ 200](#)].

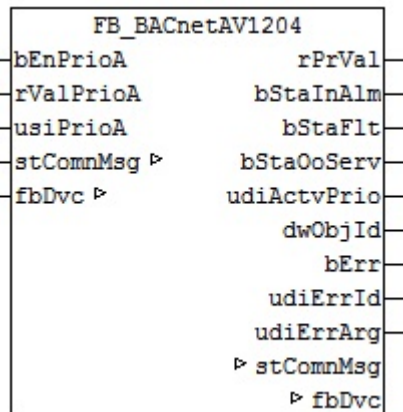
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.13 FB_BACnetAV1204

BACnet Analog-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleinste" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAV1201](#) [[▶ 59](#)]
- [FB_BACnetAV1202](#) [[▶ 60](#)]
- [FB_BACnetAV1203](#) [[▶ 63](#)]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
rValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

Ein-Ausgänge

VAR_INPUT

```
bEnPrioA : BOOL;
rValPrioA : REAL;
usiPrioA : USINT;
```

bEnPrioA: Freigabe zum Beschreiben

rValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

VAR_OUTPUT

```
rPrVal : REAL;
bStaInAlm : BOOL;
bStaFlt : BOOL;
bStaOoServ : BOOL;
udiActvPrio : UDINT;
dwObjId : DWORD;
bErrs : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

rPrVal: Aktueller Wert des Analog-Value-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Analog-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Analog-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Analog-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Analog-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_CmnMsg \[► 200\]](#).

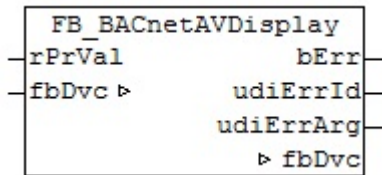
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.14 FB_BACnetAVDisplay

BACnet Analog-Value Objekt, welches zur Anzeige eines Wertes aus der PLC im BACnet verwendet werden kann.



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAV1201 \[▶ 59\]](#)
- [FB_BACnetAV1202 \[▶ 60\]](#)
- [FB_BACnetAV1203 \[▶ 63\]](#)
- [FB_BACnetAV1204 \[▶ 67\]](#)
- [FB_BACnetAVSetpoint \[▶ 70\]](#)

Ein-Ausgänge

VAR_INPUT

```
rPrVal : REAL;
```

rPrVal: Wert aus der PLC, der bei Wertänderung auf das BACnet Property Present Value geschrieben wird.

VAR_OUTPUT

```
bErrs : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[▶ 340\]](#).

VAR_IN_OUT

```
fbDvc : FB_BACnet_Device;
```

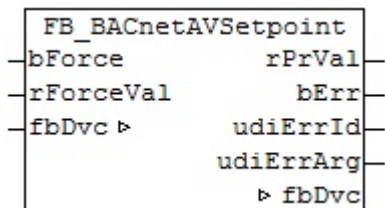
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.15 FB_BACnetAVSetpoint

BACnet Analog-Value Objekt, welches das BACnet Property Present Value in der PLC abbildet.



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Analog-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetAV1201 \[► 59\]](#)
- [FB_BACnetAV1202 \[► 60\]](#)
- [FB_BACnetAV1203 \[► 63\]](#)
- [FB_BACnetAV1204 \[► 67\]](#)
- [FB_BACnetAVDisplay \[► 69\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface.

Mit jeder steigenden Flanke an bForce wird einmalig der Wert von rForceVal auf den Present Value des AV-Objekts geschrieben.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bForce	Mit jeder steigenden Flanke an bForce wird einmalig der Wert von rForceVal auf den Present Value des AV-Objekts geschrieben.	
rForceVal	Zu schreibender Wert auf das BACnet Property Present Value	Present Value (85)

Ein-Ausgänge

VAR_INPUT

```

bForce      : BOOL;
rForceVal   : REAL;
  
```

bForce: Mit jeder steigenden Flanke an bForce wird einmalig der Wert von rForceVal auf den Present Value des AV-Objekts geschrieben.

rForceVal: Wert, der auf das BACnet Property Present Value geschrieben wird.

VAR_OUTPUT

```
rPrVal      : REAL;
bErrs      : BOOL;
udiErrId   : UDINT;
udiErrArg  : UDINT;
```

rPrVal: Aktueller Wert des BACnet Property Present Value - direkt aus dem BACnet eingelesen

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
fbDvc      : FB_BACnet_Device;
```

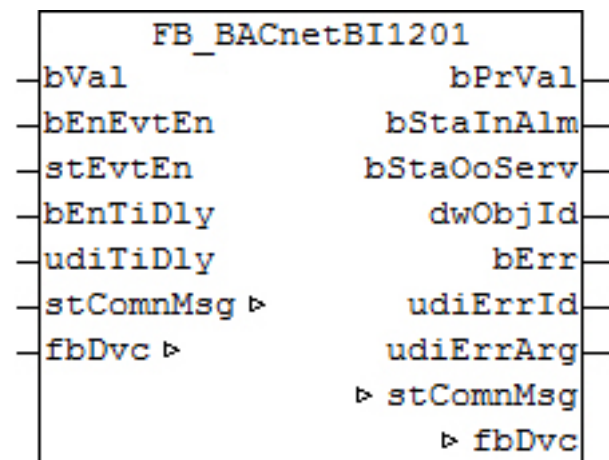
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.16 FB_BACnetBI1201

BACnet Binary-Input



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Input-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "große" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBI1203](#) [► 72]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)

Ein-Ausgänge

VAR_INPUT

```
bVal      : BOOL;
bEnEvtEn  : BOOL;
stEvtEn   : ST_BACnet_EventTransitionBits;
bEnTiDly  : BOOL;
udiTiDly  : UDINT;
```

bVal: Eingang zur Verbindung mit dem Eingangsbit (Input) der Klemme

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

VAR_OUTPUT

```
bPrVal    : BOOL;
bStAlm    : BOOL;
bStaOoSrv : BOOL;
dwObjId   : DWORD;
bErr      : BOOL;
udiErrId  : UDINT;
udiErrArg : UDINT;
```

bPrVal: Aktueller Wert des Binary-Input-Objekts - direkt aus dem BACnet eingelesen

bStAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Input-Objekts an.

bStaOoSrv: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Input-Objekts an.

dwObjId: BACnet-Objekt-ID des Binary-Input-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_CmnMsg](#) [► 200].

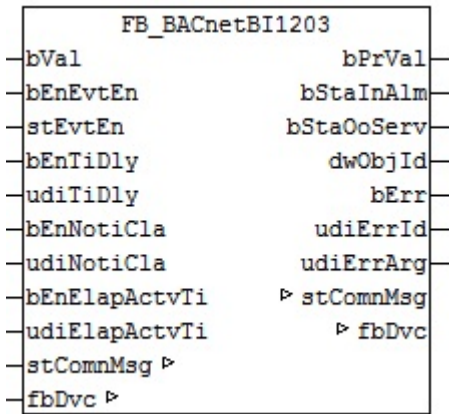
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.17 FB_BACnetBI1203

BACnet Binary-Input



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Input-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "große" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBI1201 \[► 71\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)
bEnElapActvTi	Freigabe zum Beschreiben der abgelaufenen Aktiv-Zeit	
udiElapActvTi	Property-Wert abgelaufene Aktiv-Zeit	ElapsedActiveTime (33)

Ein-Ausgänge

VAR_INPUT

```

bVal      : BOOL;
bEnEvtEn  : BOOL;
stEvtEn   : ST_BACnet_EventTransitionBits;
bEnTiDly  : BOOL;
udiTiDly  : UDINT;
bEnNotiCla : BOOL;
udiNotiCla : UDINT;
bEnElapActvTi : BOOL;
udiElapActvTi : UDINT;
    
```

bVal: Eingang zur Verbindung mit dem Eingangsbit (Input) der Klemme

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

bEnElapActvTi / udiElapActvTi: Freigabe/Property-Wert abgelaufene Aktiv-Zeit

VAR_OUTPUT

```
bPrVal      : BOOL;
bStaInAlm   : BOOL;
bStaOoServ  : BOOL;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bPrVal: Aktueller Wert des Binary-Input-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Input-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Input-Objekts an.

dwObjId: BACnet-Objekt-ID des Binary-Input-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_ComMsg](#) [► 200].

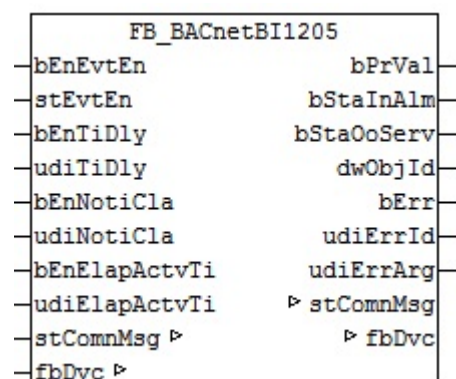
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.18 FB_BACnetBI1205

BACnet Binary-Input



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Input-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "große" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBI1201 \[► 71\]](#)
- [FB_BACnetBI1203 \[► 72\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)
bEnElapActvTi	Freigabe zum Beschreiben der abgelaufenen Aktiv-Zeit	
udiElapActvTi	Property-Wert abgelaufene Aktiv-Zeit	ElapsedActiveTime (33)

Ein-Ausgänge

VAR_INPUT

```

bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
bEnTiDly      : BOOL;
udiTiDly      : UDINT;
bEnNotiCla    : BOOL;
udiNotiCla    : UDINT;
bEnElapActvTi : BOOL;
udiElapActvTi : UDINT;
    
```

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

bEnElapActvTi / udiElapActvTi: Freigabe/Property-Wert abgelaufene Aktiv-Zeit

VAR_OUTPUT

```

bPrVal        : BOOL;
bStaInAlm     : BOOL;
bStaOoServ    : BOOL;
dwObjId       : DWORD;
bErr          : BOOL;
udiErrId      : UDINT;
udiErrArg     : UDINT;
    
```

bPrVal: Aktueller Wert des Binary-Input-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Input-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Input-Objekts an.

dwObjId: BACnet-Objekt-ID des Binary-Input-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [[▶ 340](#)].

VAR_IN_OUT

```
stCommMsg : ST_BA_CommMsg;
fbDvc      : FB_BACnet_Device;
```

stCommMsg: Referenz auf die [Verbindungsstruktur](#) [[▶ 331](#)] zum Melde-Sammelbaustein [FB_BA_CommMsg](#) [[▶ 200](#)].

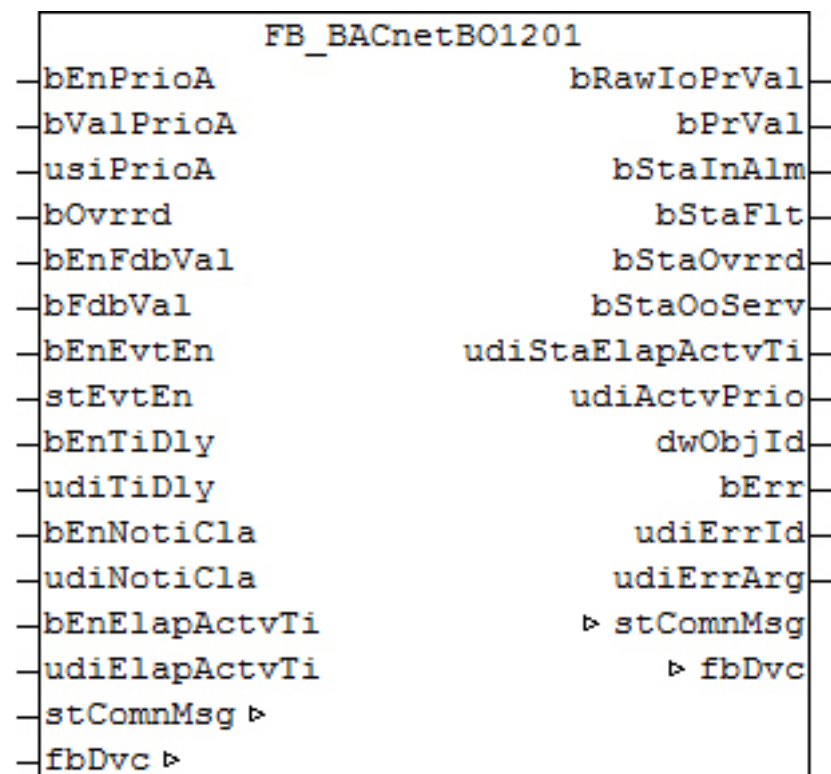
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.19 FB_BACnetBO1201

BACnet Binary-Output



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleinste" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBO1202](#) [[▶ 78](#)]

- FB BACnetBO1203 [▶ 82]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
bValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
bValPrioA     : BOOL;
usiPrioA      : USINT;
bOvrrd       : BOOL;
bEnFdbVal    : BOOL;
bFdbVal      : BOOL;
bEnEvtEn     : BOOL;
stEvtEn      : ST_BACnet_EventTransitionBits;
bEnTiDly     : BOOL;
udiTiDly     : UDINT;
bEnNotiCla   : BOOL;
udiNotiCla   : UDINT;
bEnElapActvTi : BOOL;
udiElapActvTi : UDINT;
    
```

bEnPrioA: Freigabe zum Beschreiben

bValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bOvrrd: An diesem Eingang kann das Eingangssignal einer lokale Vorrangbedienung z. B. von einem KM2652 angelegt werden. Steht dieser Eingang dann auf "TRUE", so wird das im BACnet angezeigt.

bEnFdbVal / bFdbVal: Freigabe/Property-Wert Feedback-Value. Dieser Wert wird aus der PLC heraus auf das Property FeedbackValue (Property Id 40) geschrieben, jedoch **nicht** per ADS, sondern per zyklischem Interface.

Es dient einer Rückmelde-Überwachung: Ist *bEnFdbVal* auf TRUE gesetzt, so muss an *bFdbVal* innerhalb der Zeit, die im BACnet unter TIME-Delay (Property Id 113) eingetragen ist, den gleichen Wert haben, wie der Ausgang Present-Value. Ansonsten wird im EventState (Property Id 36) durch den Eintrag "offnormal" eine Rückmelde-Diskrepanz angezeigt. Ist *bEnFdbVal* auf FALSE gesetzt, so wird intern *bFdbVal* gleich dem Present-Value gesetzt, und es kann so niemals zu einer Rückmeldediskrepanz kommen.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

bEnElapActvTi / udiElapActvTi: Freigabe/Property-Wert abgelaufene Aktiv-Zeit

VAR_OUTPUT

```

bRawIoPrVal   : BOOL;
bPrVal        : BOOL;
bStainAlm     : BOOL;
    
```

```

bStaFlt      : BOOL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
udiStaElapActvTi : UDINT;
udiActvPrio : UDINT;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;

```

bRawIoPrVal: Ausgabewert an die digitale Ausgangs-Klemme

bPrVal: Aktueller Wert des Binary-Output-Objekts - direkt aus dem BACnet eingelesen

bStalnAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Output-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Output-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Binary-Output-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Output-Objekts an.

udiStaElapActvTi: Zeigt die verstrichene aktiv-Zeit des Binary-Output-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Binary-Output-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```

stComnMsg : ST_BA_ComnMsg;
fbDvc     : FB_BACnet_Device;

```

stComnMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_ComMsg \[► 200\]](#).

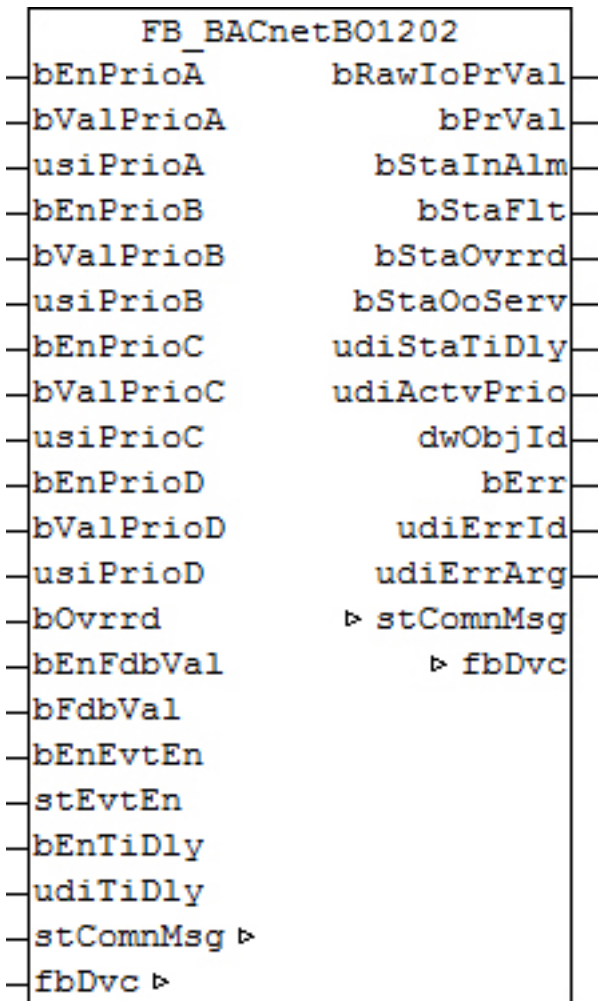
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.20 FB_BACnetBO1202

BACnet Binary-Output



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "mittlere" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBO1201 \[▶ 76\]](#)
- [FB_BACnetBO1203 \[▶ 82\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
bValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
bValPrioB	Zu schreibender Wert	Priority-Array (87)

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
bValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
bValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
bValPrioA     : BOOL;
usiPrioA      : USINT;
bEnPrioB      : BOOL;
bValPrioB     : BOOL;
usiPrioB      : USINT;
bEnPrioC      : BOOL;
bValPrioC     : BOOL;
usiPrioC      : USINT;
bEnPrioD      : BOOL;
bValPrioD     : BOOL;
usiPrioD      : USINT;
bOvrrd       : BOOL;
bEnFdbVal     : BOOL;
bFdbVal       : BOOL;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
bEnTiDly      : BOOL;
udiTiDly      : UDINT;

```

bEnPrioA: Freigabe zum Beschreiben

bValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

bValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

bValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioD: Freigabe zum Beschreiben

bValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bOvrrd: An diesem Eingang kann das Eingangssignal einer lokale Vorrangbedienung z. B. von einem KM2652 angelegt werden. Steht dieser Eingang dann auf "TRUE", so wird das im BACnet angezeigt.

bEnFdbVal / bFdbVal: Freigabe/Property-Wert Feedback-Value. Dieser Wert wird aus der PLC heraus auf das Property FeedbackValue (Property Id 40) geschrieben, jedoch **nicht** per ADS, sondern per zyklischem Interface.

Es dient einer Rückmelde-Überwachung: Ist *bEnFdbVal* auf TRUE gesetzt, so muss an *bFdbVal* innerhalb der Zeit, die im BACnet unter TIME-Delay (Property Id 113) eingetragen ist, den gleichen Wert haben, wie der Ausgang Present-Value. Ansonsten wird im EventState (Property Id 36) durch den Eintrag "offnormal" eine Rückmelde-Diskrepanz angezeigt. Ist *bEnFdbVal* auf FALSE gesetzt, so wird intern *bFdbVal* gleich dem Present-Value gesetzt, und es kann so niemals zu einer Rückmeldediskrepanz kommen.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

VAR_OUTPUT

```
bRawIoPrVal : BOOL;
bPrVal      : BOOL;
bStaInAlm   : BOOL;
bStaFlt     : BOOL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
udiStaTiDly : UDINT;
udiActvPrio : UDINT;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bRawIoPrVal: Ausgabewert an die digitale Ausgangs-Klemme

bPrVal: Aktueller Wert des Binary-Output-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Output-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Output-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Overridden“ des Binary-Output-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Output-Objekts an.

udiStaTiDly: Zeigt den aktuell parametrisierten TimeDelay (Property Id 113) an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Binary-Output-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_ComMsg](#) [► 200].

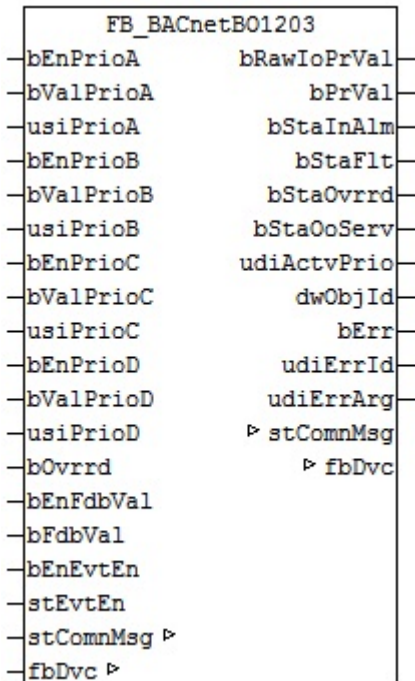
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.21 FB_BACnetBO1203

BACnet Binary-Output



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "größte" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBO1201 \[▶ 76\]](#)
- [FB_BACnetBO1202 \[▶ 78\]](#)
- [FB_BACnetBO1205 \[▶ 84\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
bValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
bValPrioB	Zu schreibender Wert	Priority-Array (87)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
bValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
bValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
bValPrioA     : BOOL;
usiPrioA      : USINT;
bEnPrioB      : BOOL;
bValPrioB     : BOOL;
usiPrioB      : USINT;
bEnPrioC      : BOOL;
bValPrioC     : BOOL;
usiPrioC      : USINT;
bEnPrioD      : BOOL;
bValPrioD     : BOOL;
usiPrioD      : USINT;
bOvrrd       : BOOL;
bEnFdbVal     : BOOL;
bFdbVal       : BOOL;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
    
```

bEnPrioA: Freigabe zum Beschreiben

bValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

bValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

bValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioD: Freigabe zum Beschreiben

bValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bOvrrd: An diesem Eingang kann das Eingangssignal einer lokale Vorrangbedienung z. B. von einem KM2652 angelegt werden. Steht dieser Eingang dann auf "TRUE", so wird das im BACnet angezeigt.

bEnFdbVal / bFdbVal: Freigabe/Property-Wert Feedback-Value. Dieser Wert wird aus der PLC heraus auf das Property FeedbackValue (Property Id 40) geschrieben, jedoch **nicht** per ADS, sondern per zyklischem Interface.

Es dient einer Rückmelde-Überwachung: Ist *bEnFdbVal* auf TRUE gesetzt, so muss an *bFdbVal* innerhalb der Zeit, die im BACnet unter TIME-Delay (Property Id 113) eingetragen ist, den gleichen Wert haben, wie der Ausgang Present-Value. Ansonsten wird im EventState (Property Id 36) durch den Eintrag "offnormal" eine Rückmelde-Diskrepanz angezeigt. Ist *bEnFdbVal* auf FALSE gesetzt, so wird intern *bFdbVal* gleich dem Present-Value gesetzt, und es kann so niemals zu einer Rückmeldediskrepanz kommen.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

VAR_OUTPUT

```
bRawIoPrVal : BOOL;
bPrVal      : BOOL;
bStaInAlm   : BOOL;
bStaFlt     : BOOL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
udiActvPrio : UDINT;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bRawIoPrVal: Ausgabewert an die digitale Ausgangs-Klemme

bPrVal: Aktueller Wert des Binary-Output-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Output-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Output-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Binary-Output-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Output-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Binary-Output-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_CmnMsg](#) [► 200].

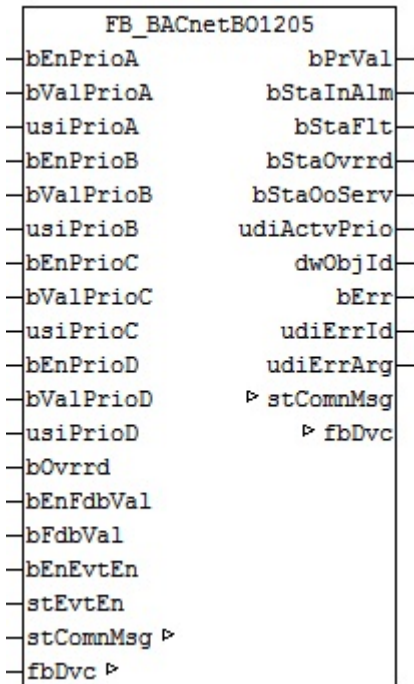
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.22 FB_BACnetBO1205

BACnet Binary-Output



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "größte" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBO1201 \[▶ 76\]](#)
- [FB_BACnetBO1202 \[▶ 78\]](#)
- [FB_BACnetBO1203 \[▶ 82\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
bValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
bValPrioB	Zu schreibender Wert	Priority-Array (87)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
bValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
bValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
bValPrioA     : BOOL;
usiPrioA      : USINT;
bEnPrioB      : BOOL;
bValPrioB     : BOOL;
usiPrioB      : USINT;
bEnPrioC      : BOOL;
bValPrioC     : BOOL;
usiPrioC      : USINT;
bEnPrioD      : BOOL;
bValPrioD     : BOOL;
usiPrioD      : USINT;
bOvrrd        : BOOL;
bEnFdbVal     : BOOL;
bFdbVal       : BOOL;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;

```

bEnPrioA: Freigabe zum Beschreiben

bValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

bValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

bValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioD: Freigabe zum Beschreiben

bValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bOvrrd: An diesem Eingang kann das Eingangssignal einer lokale Vorrangbedienung z. B. von einem KM2652 angelegt werden. Steht dieser Eingang dann auf "TRUE", so wird das im BACnet angezeigt.

bEnFdbVal / bFdbVal: Freigabe/Property-Wert Feedback-Value. Dieser Wert wird aus der PLC heraus auf das Property FeedbackValue (Property Id 40) geschrieben, jedoch **nicht** per ADS, sondern per zyklischem Interface.

Es dient einer Rückmelde-Überwachung: Ist *bEnFdbVal* auf TRUE gesetzt, so muss an *bFdbVal* innerhalb der Zeit, die im BACnet unter TIME-Delay (Property Id 113) eingetragen ist, den gleichen Wert haben, wie der Ausgang Present-Value. Ansonsten wird im EventState (Property Id 36) durch den Eintrag "offnormal" eine Rückmelde-Diskrepanz angezeigt. Ist *bEnFdbVal* auf FALSE gesetzt, so wird intern *bFdbVal* gleich dem Present-Value gesetzt, und es kann so niemals zu einer Rückmeldediskrepanz kommen.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

VAR_OUTPUT

```
bPrVal      : BOOL;
bStaInAlm  : BOOL;
bStaFlt    : BOOL;
bStaOvrrd  : BOOL;
bStaOoSrv  : BOOL;
udiActvPrio : UDINT;
dwObjId    : DWORD;
bErr       : BOOL;
udiErrId   : UDINT;
udiErrArg  : UDINT;
```

- bPrVal:** Aktueller Wert des Binary-Output-Objekts - direkt aus dem BACnet eingelesen
- bStaInAlm:** Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Output-Objekts an.
- bStaFlt:** Zeigt den Zustand des Statusflags „Fault“ des Binary-Output-Objekts an.
- bStaOvrrd:** Zeigt den Zustand des Statusflags „Override“ des Binary-Output-Objekts an.
- bStaOoSrv:** Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Output-Objekts an.
- udiActvPrio:** Zeigt an, welche Priorität die aktive ist.
- dwObjId:** BACnet-Objekt-ID des Binary-Output-Objekts
- bErr:** Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.
- udiErrId / udiErrArg:** Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc     : FB_BACnet_Device;
```

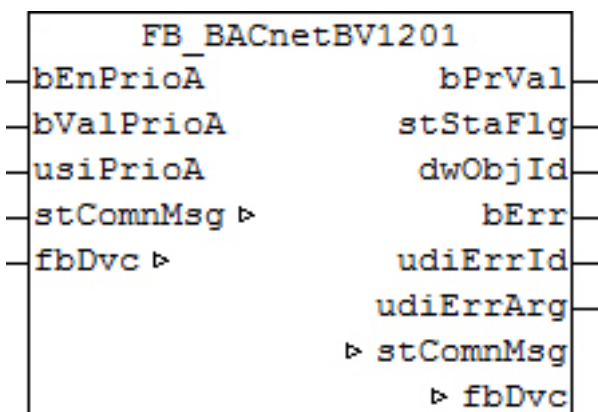
- stComnMsg:** Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_CmnMsg \[► 200\]](#).
- fbDvc:** Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.23 FB_BACnetBV1201

BACnet Binary-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleinste" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBV1202 \[► 89\]](#)
- [FB_BACnetBV1203 \[► 91\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
bValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

Ein-Ausgänge

VAR_INPUT

```
bEnPrioA : BOOL;
bValPrioA : BOOL;
usiPrioA : USINT;
```

bEnPrioA: Freigabe zum Beschreiben

bValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

VAR_OUTPUT

```
bPrVal : BOOL;
stStaFlg : ST_BACnet_StatusFlags;
dwObjId : DWORD;
bErr : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

bPrVal: Aktueller Wert des Binary-Value-Objekts - direkt aus dem BACnet eingelesen

stStaFlg: Ausgabestruktur des BACnet-Status

dwObjId: BACnet-Objekt-ID des Binary-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_CmnMsg \[► 200\]](#).

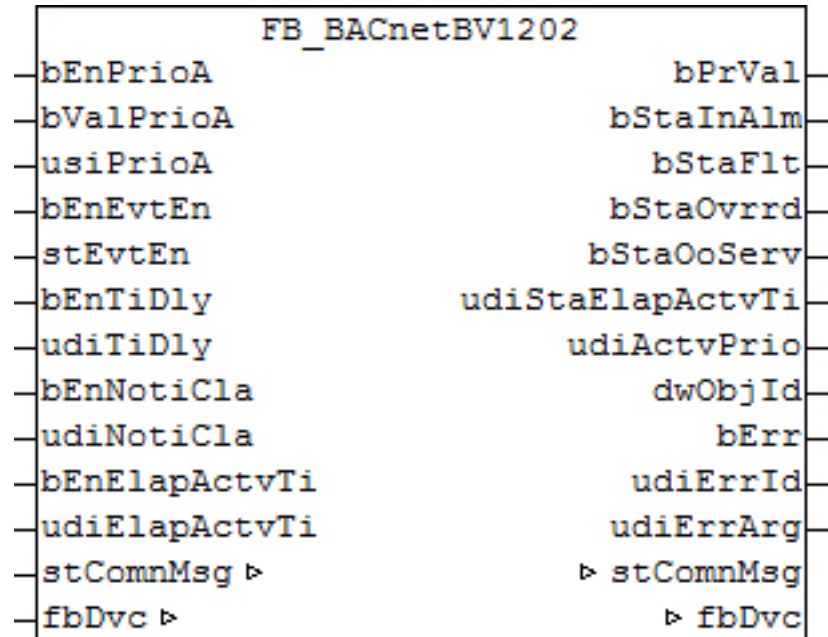
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.24 FB_BACnetBV1202

BACnet Binary-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "mittlere" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBV1201](#) [▶ 87]
- [FB_BACnetBV1203](#) [▶ 91]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
bValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)
bEnElapActvTi	Freigabe zum Beschreiben der abgelaufenen Aktiv-Zeit	
udiElapActvTi	Property-Wert abgelaufene Aktiv-Zeit	ElapsedActiveTime (33)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
bValPrioA     : BOOL;
usiPrioA      : USINT;
bEnEvtEn     : BOOL;
stEvtEn      : ST_BACnet_EventTransitionBits;
bEnTiDly     : BOOL;
udiTiDly     : UDINT;
bEnNotiCla   : BOOL;
udiNotiCla   : UDINT;
bEnElapActvTi : BOOL;
udiElapActvTi : UDINT;

```

bEnPrioA: Freigabe zum Beschreiben

bValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

bEnElapActvTi / udiElapActvTi: Freigabe/Property-Wert abgelaufene Aktiv-Zeit

VAR_OUTPUT

```

bPrVal       : BOOL;
bStaInAlm    : BOOL;
bStaFlt      : BOOL;
bStaOvrrd    : BOOL;
bStaOoServ   : BOOL;
udiStaElapActvTi : UDINT;
udiActvPrio  : UDINT;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;

```

bPrVal: Aktueller Wert des Binary-Value-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Value-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Overridden“ des Binary-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Value-Objekts an.

udiStaElapActvTi: Zeigt die verstrichene aktiv-Zeit des Binary-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Binary-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stConnMsg : ST_BA_ConnMsg;
fbDvc     : FB_BACnet_Device;
```

stConnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA ComMsg](#) [► 200].

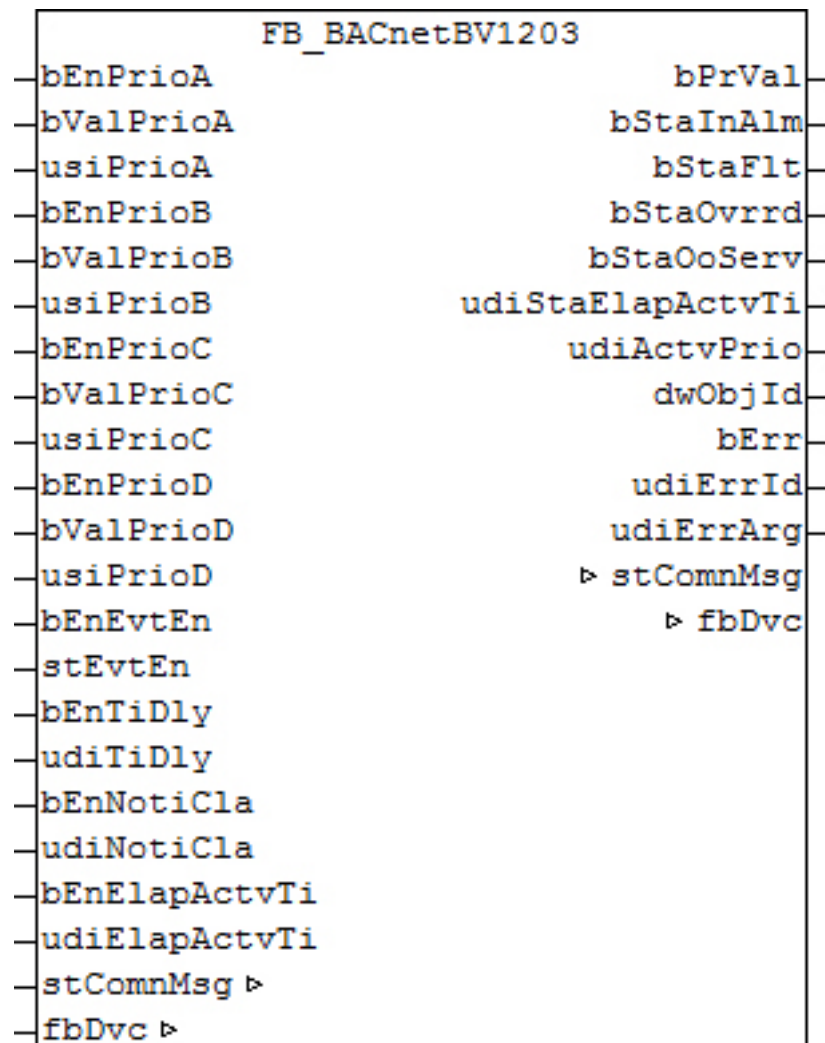
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.25 FB_BACnetBV1203

BACnet Binary-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "größte" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBV1201 \[▶ 87\]](#)
- [FB_BACnetBV1202 \[▶ 89\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
bValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
bValPrioB	Zu schreibender Wert	Priority-Array (87)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
bValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
bValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)
bEnElapActvTi	Freigabe zum Beschreiben der abgelaufenen Aktiv-Zeit	
udiElapActvTi	Property-Wert abgelaufene Aktiv-Zeit	ElapsedActiveTime (33)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
bValPrioA     : BOOL;
usiPrioA      : USINT;
bEnPrioB      : BOOL;
bValPrioB     : BOOL;
usiPrioB      : USINT;

```

```

bEnPrioC      : BOOL;
bValPrioC     : BOOL;
usiPrioC      : USINT;
bEnPrioD      : BOOL;
bValPrioD     : BOOL;
usiPrioD      : USINT;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
bEnTiDly      : BOOL;
udiTiDly      : UDINT;
bEnNotiCla    : BOOL;
udiNotiCla    : UDINT;
bEnElapActvTi : BOOL;
udiElapActvTi : UDINT;

```

bEnPrioA: Freigabe zum Beschreiben

bValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

bValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

bValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioD: Freigabe zum Beschreiben

bValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

bEnElapActvTi / udiElapActvTi: Freigabe/Property-Wert abgelaufene Aktiv-Zeit

VAR_OUTPUT

```

bPrVal        : BOOL;
bStaInAlm     : BOOL;
bStaFlt       : BOOL;
bStaOvrrd     : BOOL;
bStaOoServ    : BOOL;
udiStaElapActvTi : UDINT;
udiActvPrio   : UDINT;
dwObjId       : DWORD;
bErr          : BOOL;
udiErrId      : UDINT;
udiErrArg     : UDINT;

```

bPrVal: Aktueller Wert des Binary-Value-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Value-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Overridden“ des Binary-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Value-Objekts an.

udiStaElapActvTi: Zeigt die verstrichene aktiv-Zeit des Binary-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Binary-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc      : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_ComMsg](#) [► 200].

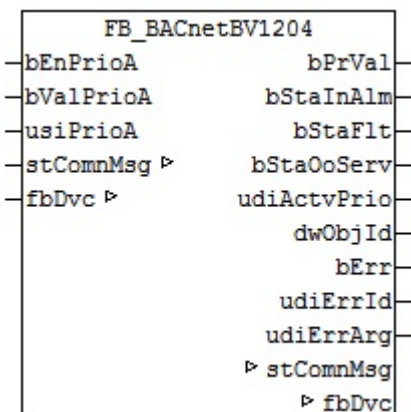
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.26 FB_BACnetBV1204

BACnet Binary-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleinste" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBV1201](#) [► 87]
- [FB_BACnetBV1202](#) [► 89]
- [FB_BACnetBV1203](#) [► 91]
- [FB_BACnetBVDisplay](#) [► 96]
- [FB_BACnetBVSetpoint](#) [► 97]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt

(steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
bValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

Ein-Ausgänge

VAR_INPUT

```
bEnPrioA : BOOL;
bValPrioA : BOOL;
usiPrioA : USINT;
```

bEnPrioA: Freigabe zum Beschreiben

bValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

VAR_OUTPUT

```
bPrVal : BOOL;
bStaInAlm : BOOL;
bStaFlt : BOOL;
bStaOoServ : BOOL;
udiActvPrio : UDINT;
dwObjId : DWORD;
bErr : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

bPrVal: Aktueller Wert des Binary-Value-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Binary-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Binary-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

dwObjId: BACnet-Objekt-ID des Binary-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[▶ 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[▶ 331\]](#) zum Melde-Sammelbaustein [FB_BA_ComMsg \[▶ 200\]](#).

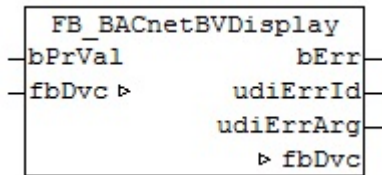
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.27 FB_BACnetBVDisplay

BACnet Binary-Value Objekt, welches zur Anzeige eines Wertes aus der PLC im BACnet verwendet werden kann.



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBV1201 \[► 87\]](#)
- [FB_BACnetBV1202 \[► 89\]](#)
- [FB_BACnetBV1203 \[► 91\]](#)
- [FB_BACnetBV1204 \[► 94\]](#)
- [FB_BACnetSetpoint \[► 97\]](#)

Ein-Ausgänge

VAR_INPUT

```
bPrVal : BOOL;
```

bPrVal: Boolescher Wert aus der PLC, der bei Wertänderung auf das BACnet Property Present Value geschrieben wird.

VAR_OUTPUT

```
bErrs : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
fbDvc : FB_BACnet_Device;
```

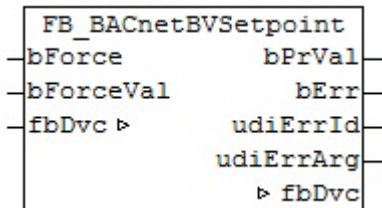
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.28 FB_BACnetBVSetpoint

BACnet Binary-Value Objekt, welches das BACnet Property Present Value in der PLC abbildet.



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Binary-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetBV1201](#) [▶ 87]
- [FB_BACnetBV1202](#) [▶ 89]
- [FB_BACnetBV1203](#) [▶ 91]
- [FB_BACnetBV1204](#) [▶ 94]
- [FB_BACnetBVDisplay](#) [▶ 96]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface.

Mit jeder steigenden Flanke an bForce wird einmalig der Wert von rForceVal auf den Present Value des BV-Objekts geschrieben.

PLC-Vari- ble	Bemerkung	BACnet-Property (Property ID)
bForce	Mit jeder steigenden Flanke an bForce wird einmalig der Wert von bForceVal auf den Present Value des BV-Objekts geschrieben.	
bForceVal	Zu schreibender Wert auf das BACnet Property Present Value	Present Value (85)

Ein-Ausgänge

VAR_INPUT

```

bForce      : BOOL;
bForceVal   : BOOL;
    
```

bForce: Mit jeder steigenden Flanke an bForce wird einmalig der Wert von bForceVal auf den Present Value des BV-Objekts geschrieben.

bForceVal: Wert, der auf das BACnet Property Present Value geschrieben wird.

VAR_OUTPUT

```
bPrVal      : BOOL;
bErrs       : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bPrVal: Aktueller Wert des BACnet Property Present Value - direkt aus dem BACnet eingelesen

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
fbDvc       : FB_BACnet_Device;
```

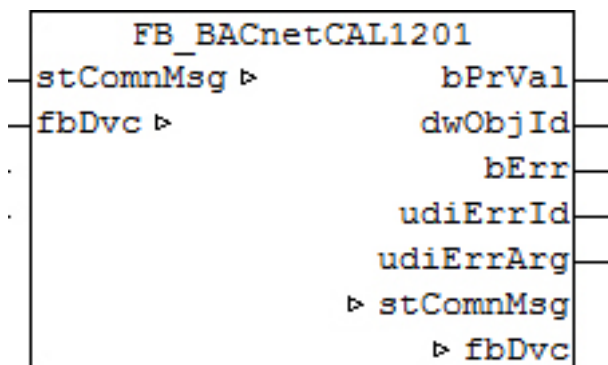
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.29 FB_BACnetCAL1201

BACnet Kalender

**Funktionsbeschreibung**

Dieser Baustein generiert ein BACnet-Calendar-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Ein-Ausgänge**VAR_OUTPUT**

```
bPrVal      : BOOL;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bPrVal: Ein TRUE an diesem Ausgang zeigt an, dass das aktuelle Datum innerhalb der parametrisierten Perioden liegt.

dwObjId: BACnet-Objekt-ID des Calendar-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [▶ 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [▶ 331] zum Melde-Sammelbaustein [FB_BA_ComMsg](#) [▶ 200].

fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.30 FB_BACnetLoop1201

Universeller BACnet PID-Regler (große Version)

FB_BACnetLoop1201	
-bEn	rPrVal
-bEnActn	rX
-bActn	rW
-bEnPropConst	bStaActn
-rPropConst	rStaYMin
-bEnIntgConst	rStaYMax
-rIntgConst	bStaOvrrd
-bEnDervConst	bStaOoServ
-rDervConst	bStaFlt
-bEnEvtEn	bStaInAlm
-stEvtEn	bOpLp
-bEnErrLmt	bOthFlt
-rErrLmt	stStaFlg
-bEnTiDly	dwObjId
-udiTiDly	bErr
-bEnNotiCla	udiErrId
-udiNotiCla	udiErrArg
-bEnMinOut	▷ stCommMsg
-rMinOut	▷ fbDvc
-bEnMaxOut	
-rMaxOut	
-bSync	
-lrSync	
-lrNz	
-udiDampConst	
-udiMinToMax	
-udiMaxToMin	
-uiCycCl	
-stCommMsg	▷
-fbDvc	▷

Funktionsbeschreibung

Dieser Baustein bildet den Baustein [FB_BA_PIDCtrl](#) [▷ 157] als Loop-Objekt im BACnet ab. Das Loop-Objekt auf der BACnet-Seite hat dabei im Wesentlichen die Aufgabe, Parameter und Werte aus der PLC heraus darzustellen und in die PLC zu übertragen. Die Reglerlogik liegt allein in der PLC im untergeordneten [FB_BA_PIDCtrl](#) [▷ 157]. Es ist möglich, die wesentlichen Parameter von beiden Seiten - PLC und BACnet - zu verändern wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnActn	Freigabe zum Beschreiben des ReglerWirksinns	

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bActn	Property-Wert ReglerWirksinn bActn=TRUE, BACnet direct → ReglerWirksinn: E=X-W (Kühlen) bActn=FALSE, BACnet reverse → ReglerWirksinn: E=W-X (Heizen)	Action (2)
bEnPropConst	Freigabe zum Beschreiben der Proportionalitätskonstante (K-Faktor)	
rPropConst	Property-Wert Proportionalitätskonstante (K-Faktor)	ProportionalConstant (93)
bEnIntgConst	Freigabe zum Beschreiben der Integrationszeit (I-Anteil)	
rIntgConst	Property-Wert Integrationszeit [s] (I-Anteil)	IntegralConstant (49)
bEnDervConst	Freigabe zum Beschreiben der Vorhaltezeit (D-Anteil)	
rDervConst	Property-Wert Vorhaltezeit [s] (D-Anteil)	DerivativeConstant (26)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnErrLmt	Freigabe zum Beschreiben des Meldewerts "Maximalwert Regelabweichung"	
rErrLmt	Property-Wert Meldewert "Maximalwert Regelabweichung"	ErrorLimit (34)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)
bEnMinOut	Freigabe zum Beschreiben der Minimalausgabe am Reglerausgang	
rMinOut	Property-Wert Minimalausgabe am Reglerausgang	MinimumOutput (68)
bEnMaxOut	Freigabe zum Beschreiben der Maximalausgabe am Reglerausgang	
rMaxOut	Property-Wert Maximalausgabe am Reglerausgang	MaximumOutput (61)

Ein-Ausgänge

VAR_INPUT

```

bEn          : BOOL;
bEnActn     : BOOL;
bActn       : BOOL;
bEnPropConst : BOOL;
rPropConst  : REAL;
bEnIntgConst : BOOL;
rIntgConst  : REAL;
bEnDervConst : BOOL;
rDervConst  : REAL;
bEnEvtEn    : BOOL;
stEvtEn     : ST_BACnet_EventTransitionBits;
bEnErrLmt   : BOOL;
rErrLmt     : REAL;
bEnTiDly    : BOOL;
udiTiDly    : UDINT;
bEnNotiCla  : BOOL;
udiNotiCla  : UDINT;
bEnMinOut   : BOOL;
rMinOut     : REAL;
bEnMaxOut   : BOOL;
rMaxOut     : REAL;
bSync       : BOOL;
lrSync      : LREAL;
lrNz        : LREAL;
udiDampConst : UDINT;
udiMinToMax : UDINT;
udiMaxToMin : UDINT;
uiCycCl     : UINT;
    
```

bEn: Regleraktivierung

bEnActn / bActn: Freigabe/Property-Wert Reglerwirksinn: „direct“ oder „reverse“

bEnPropConst / rPropConst: Freigabe/Property-Wert Proportionalitätskonstante (K-Faktor)

bEnIntgConst / rIntgConst: Freigabe/Property-Wert Integrationszeit [s] (I-Anteil)

bEnDervConst / rDervConst: Freigabe/Property-Wert Vorhaltezeit [s] (D-Anteil)

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnErrLmt / rErrLmt: Freigabe/Property-Wert Meldewert Maximalausgabe

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

bEnMinOut / rMinOut: Freigabe/Property-Wert Maximalausgabe am Reglerausgang

bEnMaxOut / rMaxOut: Freigabe/Property-Wert Minimalausgabe am Reglerausgang

bSync: Setzen des Ausgangs auf *IrSync*

IrSync: Synchronisationswert, auf den die Stellgröße bei einer steigenden Flanke am Eingang *bSync* gesetzt wird.

IrNz: Neutrale Zone

udiDampConst: Dämpfungszeit des D-Anteiles [s]

udiMinToMax: Steigungsbegrenzung des Reglerausgangs für Anstieg: *udiMinToMax* [s] bezogen auf eine Änderung von *IrMinOut* auf *rMaxOut*.

udiMaxToMin: Steigungsbegrenzung des Reglerausgangs für Abfall: *udiMaxToMin* [s] bezogen auf eine Änderung von *IrMaxOut* auf *rMinOut*.

uiCycCl: Aufrufzyklus des Bausteins als Vielfaches der Zykluszeit. Ein Nulleintrag wird automatisch als *uiCycleCall* = 1 gewertet.

Beispiel: *tTaskCycleTime* = 20ms, *uiCtrlCycleCall* = 10 -> Der Regelalgorithmus wird alle 200ms aufgerufen. Damit werden aber auch nur alle 200ms die Ausgänge aktualisiert.

VAR_OUTPUT

```
rPrVal      : REAL;
rX          : REAL;
rW          : REAL;
bStaActn    : BOOL;
rStaYMin    : REAL;
rStaYMax    : REAL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
bStaFlt     : BOOL;
bStaInAlm   : BOOL;
bOpLp      : BOOL;
bOthFlt     : BOOL;
stStaFlg    : ST_BACnet_StatusFlags;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

rPrVal: Stellwert - Reglerausgang

rX: Aktuell gültiger Istwert aus dem BACnet gelesen.

rW: Aktuell gültiger Sollwert aus dem BACnet gelesen.

bStaActn: Zeigt den Zustand des Wirksinns an.

bStaYMin: Zeigt den unteren Wert der Regler-Ausgangsbegrenzung an.

bStaYMax: Zeigt den oberen Wert der Regler-Ausgangsbegrenzung an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Overridden“ des Loop-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Loop-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Loop-Objekts an.

bStalnAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Loop-Objekts an.

bOpLp: Zeigt den Zustand „OpenLoop“ des Loop-Objekt-Properties „Reliability“ (Zuverlässigkeit) an.

bOthFlt: Zeigt den Zustand „OtherFault“ des Loop-Objekt-Properties „Reliability“ (Zuverlässigkeit) an.

stStaFig: Ausgabestruktur des BACnet-Status

dwObjId: BACnet-Objekt-ID des Loop-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_ComMsg \[► 200\]](#).

fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.31 FB_BACnetLoop1202

Universeller BACnet PID-Regler. Im Vergleich zum [FB_BACnetLoop1201 \[► 99\]](#) lässt dieser Baustein das Beschreiben folgender BACnet-Properties aus der SPS heraus nicht zu:

- Notification Class
- Minimum Output
- Maximum Output

FB_BACnetLoop1202	
bEn	rPrVal
bEnActn	rX
bActn	rW
bEnPropConst	bStaActn
rPropConst	rStaYMin
bEnIntgConst	rStaYMax
rIntgConst	bStaOvrrd
bEnDervConst	bStaOoServ
rDervConst	bStaFlt
bEnEvtEn	bStaInAlm
stEvtEn	bOpLp
bEnErrLmt	bOthFlt
rErrLmt	stStaFlg
bEnTiDly	dwObjId
udiTiDly	bErr
bSync	udiErrId
lrSync	udiErrArg
lrNz	▷ stCommMsg
udiDampConst	▷ fbDvc
udiMinToMax	
udiMaxToMin	
uiCycCl	
stCommMsg	▷
fbDvc	▷

Funktionsbeschreibung

Dieser Baustein bildet den Baustein [FB BA_PIDCtrl \[▷ 157\]](#) als Loop-Objekt im BACnet ab. Das Loop-Objekt auf der BACnet-Seite hat dabei im Wesentlichen die Aufgabe, Parameter und Werte aus der PLC heraus darzustellen und in die PLC zu übertragen. Die Reglerlogik liegt allein in der PLC im untergeordneten [FB BA_PIDCtrl \[▷ 157\]](#). Es ist möglich, die wesentlichen Parameter von beiden Seiten - PLC und BACnet - zu verändern wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnActn	Freigabe zum Beschreiben des ReglerWirksinns	
bActn	Property-Wert ReglerWirksinn bActn=TRUE, BACnet direct → ReglerWirksinn: E=X-W (Kühlen) bActn=FALSE, BACnet reverse → ReglerWirksinn: E=W-X (Heizen)	Action (2)
bEnPropConst	Freigabe zum Beschreiben der Proportionalitätskonstante (K-Faktor)	
rPropConst	Property-Wert Proportionalitätskonstante (K-Faktor)	ProporttionalConstant (93)

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnIntgConst	Freigabe zum Beschreiben der Integrationszeit (I-Anteil)	
rIntgConst	Property-Wert Integrationszeit [s] (I-Anteil)	IntegralConstant (49)
bEnDervConst	Freigabe zum Beschreiben der Vorhaltezeit (D-Anteil)	
rDervConst	Property-Wert Vorhaltezeit [s] (D-Anteil)	DerivativeConstant (26)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnErrLmt	Freigabe zum Beschreiben des Meldewerts "Maximalwert Regelabweichung"	
rErrLmt	Property-Wert Meldewert "Maximalwert Regelabweichung"	ErrorLimit (34)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)

Ein-Ausgänge

VAR_INPUT

```

bEn          : BOOL;
bEnActn     : BOOL;
bActn       : BOOL;
bEnPropConst : BOOL;
rPropConst  : REAL;
bEnIntgConst : BOOL;
rIntgConst  : REAL;
bEnDervConst : BOOL;
rDervConst  : REAL;
bEnEvtEn    : BOOL;
stEvtEn     : ST_BACnet_EventTransitionBits;
bEnErrLmt   : BOOL;
rErrLmt     : REAL;
bEnTiDly    : BOOL;
udiTiDly    : UDINT;
bSync       : BOOL;
lrSync      : LREAL;
lrNz        : LREAL;
udiDampConst : UDINT;
udiMinToMax : UDINT;
udiMaxToMin : UDINT;
uiCycCl     : UINT;
    
```

bEn: Regleraktivierung

bEnActn / bActn: Freigabe/Property-Wert Reglerwirksinn: „direct“ oder „reverse“

bEnPropConst / rPropConst: Freigabe/Property-Wert Proportionalitätskonstante (K-Faktor)

bEnIntgConst / rIntgConst: Freigabe/Property-Wert Integrationszeit [s] (I-Anteil)

bEnDervConst / rDervConst: Freigabe/Property-Wert Vorhaltezeit [s] (D-Anteil)

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnErrLmt / rErrLmt: Freigabe/Property-Wert Meldewert Maximalausgabe

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bSync: Setzen des Ausgangs auf *lrSync*.

lrSync: Synchronisationswert, auf den die Stellgröße bei einer steigenden Flanke am Eingang *bSync* gesetzt wird.

lrNz: Neutrale Zone

udiDampConst: Dämpfungszeit des D-Anteiles [s]

udiMinToMax: Steigungsbegrenzung des Reglerausgangs für Anstieg: *udiMinToMax* [s] bezogen auf eine Änderung von *lrMinOut* auf *rMaxOut*.

udiMaxToMin: Steigungsbegrenzung des Reglerausgangs für Abfall: *udiMaxToMin* [s] bezogen auf eine Änderung von *lrMaxOut* auf *rMinOut*.

uiCycCl: Aufrufzyklus des Bausteins als Vielfaches der Zykluszeit. Ein Nulleintrag wird automatisch als *uiCycleCall* =1 gewertet.

Beispiel: *tTaskCycleTime* = 20ms, *uiCtrlCycleCall* =10 -> Der Regelalgorithmus wird alle 200ms aufgerufen. Damit werden aber auch nur alle 200ms die Ausgänge aktualisiert.

VAR_OUTPUT

```
rPrVal      : REAL;
rX          : REAL;
rW          : REAL;
bStaActn    : BOOL;
rStaYMin    : REAL;
rStaYMax    : REAL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
bStaFlt     : BOOL;
bStaInAlm   : BOOL;
bOpLp       : BOOL;
bOthFlt     : BOOL;
stStaFlg    : ST_BACnet_StatusFlags;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

rPrVal: Stellwert - Reglerausgang

rX: Aktuell gültiger Istwert aus dem BACnet gelesen.

rW: Aktuell gültiger Sollwert aus dem BACnet gelesen.

bStaActn: Zeigt den Zustand des Wirksinns an.

bStaYMin: Zeigt den unteren Wert der Regler-Ausgangsbegrenzung an.

bStaYMax: Zeigt den oberen Wert der Regler-Ausgangsbegrenzung an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Loop-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Loop-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Loop-Objekts an.

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Loop-Objekts an.

bOpLp: Zeigt den Zustand „OpenLoop“ des Loop-Objekt-Properties „Reliability“ (Zuverlässigkeit) an.

bOthFlt: Zeigt den Zustand „OtherFault“ des Loop-Objekt-Properties „Reliability“ (Zuverlässigkeit) an.

stStaFlg: Ausgabestruktur des BACnet-Status

dwObjId: BACnet-Objekt-ID des Loop-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
stComnMsg   : ST_BA_CmnMsg;
fbDvc       : FB_BACnet_Device;
```

stComnMsg: Referenz auf die Verbindungsstruktur [► 331] zum Melde-Sammelbaustein FB_BA_ComMsg [► 200].

fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.32 FB_BACnetLoopSeq1201

BACnet Sequenz-Regler

FB_BACnetLoopSeq1201	
usiMyNum	rPrVal
bEn	rX
bEnActn	rW
bActn	bStaSeqCtrlActv
bEnPropConst	bStaActn
rPropConst	rStayMin
bEnIntgConst	rStayMax
rIntgConst	bStaOvrrd
bEnDervConst	bStaOoServ
rDervConst	bStaFlt
bEnEvtEn	bStaInAlm
stEvtEn	bOpLp
bEnErrLmt	bOthFlt
rErrLmt	stStaFlg
bEnTiDly	dwObjId
udiTiDly	bErr
bEnNotiCla	udiErrId
udiNotiCla	udiErrArg
bEnMinOut	▷ stCommMsg
rMinOut	▷ stSeqLink
bEnMaxOut	▷ fbDvc
rMaxOut	
lrYSeqInit	
bSync	
lrSync	
lrNz	
udiDampConst	
udiMinToMax	
udiMaxToMin	
uiCycCl	
stCommMsg	▷
stSeqLink	▷
fbDvc	▷

Funktionsbeschreibung

Dieser Baustein bildet den Baustein [FB_BA_SeqCtrl](#) [▶ 167] als Loop-Objekt im BACnet ab. Das Loop-Objekt auf der BACnet-Seite hat dabei im Wesentlichen die Aufgabe, Parameter und Werte aus der PLC heraus darzustellen und in die PLC zu übertragen. Die Reglerlogik liegt allein in der PLC im internen [FB_BA_SeqCtrl](#) [▶ 167]. Es ist möglich, die wichtigsten Parameter von beiden Seiten - PLC und BACnet - zu verändern wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der

am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

Wie bei dem "normalen" Sequenzregler vom Typ [FB_BA_SeqCtrl](#) [► 167], welcher eine reine PLC-Lösung darstellt, funktioniert die BACnet-Version ebenfalls nur mit einem Sequenz-Link-Baustein [FB_BA_SeqLink](#) [► 171].

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnActn	Freigabe zum Beschreiben des ReglerWirksinns	
bActn	Property-Wert ReglerWirksinn bActn=TRUE, BACnet direct → ReglerWirksinn: E=X-W (Kühlen) bActn=FALSE, BACnet reverse → ReglerWirksinn: E=W-X (Heizen)	Action (2)
bEnPropConst	Freigabe zum Beschreiben der Proportionalitätskonstante (K-Faktor)	
rPropConst	Property-Wert Proportionalitätskonstante (K-Faktor)	ProporttionalConstant (93)
bEnIntgConst	Freigabe zum Beschreiben der Integrationszeit (I-Anteil)	
rIntgConst	Property-Wert Integrationszeit [s] (I-Anteil)	IntegralConstant (49)
bEnDervConst	Freigabe zum Beschreiben der Vorhaltezeit (D-Anteil)	
rDervConst	Property-Wert Vorhaltezeit [s] (D-Anteil)	DerivativeConstant (26)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnErrLmt	Freigabe zum Beschreiben des Meldewerts "Maximalwert Regelabweichung"	
rErrLmt	Property-Wert Meldewert "Maximalwert Regelabweichung"	ErrorLimit (34)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)
bEnMinOut	Freigabe zum Beschreiben der Minimalausgabe am Reglerausgang	
rMinOut	Property-Wert Minimalausgabe am Reglerausgang	MinimumOutput (68)
bEnMaxOut	Freigabe zum Beschreiben der Maximalausgabe am Reglerausgang	
rMaxOut	Property-Wert Maximalausgabe am Reglerausgang	MaximumOutput (61)

Ein-Ausgänge

VAR_INPUT

```

bEn          : BOOL;
usiMyNum    : USINT;
bEnActn     : BOOL;
bActn       : BOOL;
bEnPropConst : BOOL;
rPropConst  : REAL;
bEnIntgConst : BOOL;
rIntgConst  : REAL;
bEnDervConst : BOOL;
rDervConst  : REAL;
bEnEvtEn    : BOOL;
stEvtEn     : ST_BACnet_EventTransitionBits;
bEnErrLmt   : BOOL;
    
```

```

rErrLmt      : REAL;
bEnTiDly    : BOOL;
udiTiDly    : UDINT;
bEnNotiCla  : BOOL;
udiNotiCla  : UDINT;
bEnMinOut   : BOOL;
rMinOut     : REAL;
bEnMaxOut   : BOOL;
rMaxOut     : REAL;
bSync       : BOOL;
lrYSeqInit  : LREAL;
lrSync      : LREAL;
lrNz        : LREAL;
udiDampConst : UDINT;
udiMinToMax : UDINT;
udiMaxToMin : UDINT;
uiCycCl     : UINT;

```

usiMyNum: Ordnungszahl des Sequenzreglers

bEn: Regleraktivierung

bEnActn / bActn: Freigabe/Property-Wert ReglerWirksinn: „direct“ oder „reverse“

bEnPropConst / rPropConst: Freigabe/Property-Wert Proportionalitätskonstante (K-Faktor)

bEnIntgConst / rIntgConst: Freigabe/Property-Wert Integrationszeit [s] (I-Anteil)

bEnDervConst / rDervConst: Freigabe/Property-Wert Vorhaltezeit [s] (D-Anteil)

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnErrLmt / rErrLmt: Freigabe/Property-Wert Meldewert Maximalausgabe

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

bEnMinOut / rMinOut: Freigabe/Property-Wert Maximalausgabe am Reglerausgang

bEnMaxOut / rMaxOut: Freigabe/Property-Wert Minimalausgabe am Reglerausgang

lrYSeqInit: Start-Synchronisationswert. Ist dieser Regler derjenige, der bei Aktivierung der Regelsequenz als erster freigegeben wird, so wird dieser Regler mit diesem Wert am Ausgang gestartet.

bSync: Setzen des Ausganges auf *lrSync*.

lrSync: Synchronisationswert, auf den die Stellgröße bei einer steigenden Flanke am Eingang *bSync* gesetzt wird.

lrNz: Neutrale Zone

udiDampConst: Dämpfungszeit des D-Anteiles [s]

udiMinToMax: Steigungsbegrenzung des Reglerausgangs für Anstieg: *udiMinToMax* [s] bezogen auf eine Änderung von *lrMinOut* auf *rMaxOut*.

udiMaxToMin: Steigungsbegrenzung des Reglerausgangs für Abfall: *udiMaxToMin* [s] bezogen auf eine Änderung von *lrMaxOut* auf *rMinOut*.

uiCycCl: Aufrufzyklus des Bausteins als Vielfaches der Zykluszeit. Ein Nulleintrag wird automatisch als *uiCycleCall* = 1 gewertet.

Beispiel: *tTaskCycleTime* = 20ms, *uiCtrlCycleCall* = 10 -> Der Regelalgorithmus wird alle 200ms aufgerufen. Damit werden aber auch nur alle 200ms die Ausgänge aktualisiert.

VAR_OUTPUT

```

rPrVal      : REAL;
rX          : REAL;
rW          : REAL;
bStaSeqCtrlActv : BOOL;
bStaActn    : BOOL;
rStaYMin    : REAL;
rStaYMax    : REAL;

```

```

bStaOvrrd      : BOOL;
bStaOoServ     : BOOL;
bStaFlt       : BOOL;
bStaInAlm     : BOOL;
bOpLp        : BOOL;
bOthFlt      : BOOL;
stStaFlg     : ST_BACnet_StatusFlags;
dwObjId      : DWORD;
bErr         : BOOL;
udiErrId     : UDINT;
udiErrArg    : UDINT;
    
```

rPrVal: Stellwert - Reglerausgang

rX: Aktuell gültiger Istwert aus dem BACnet gelesen.

rW: Aktuell gültiger Sollwert aus dem BACnet gelesen.

bStaSeqCtrlActv: Sequenzregler ist freigegeben und betriebsbereit (nicht im Fehlerzustand).

bStaActn: Zeigt den Zustand des Wirksinns an.

bStaYMin: Zeigt den unteren Wert der Regler-Ausgangsbegrenzung an.

bStaYMax: Zeigt den oberen Wert der Regler-Ausgangsbegrenzung an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Loop-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Loop-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Loop-Objekts an.

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Loop-Objekts an.

bOpLp: Zeigt den Zustand „OpenLoop“ des Loop-Objekt-Properties „Reliability“ (Zuverlässigkeit) an.

bOthFlt: Zeigt den Zustand „OtherFault“ des Loop-Objekt-Properties „Reliability“ (Zuverlässigkeit) an.

stStaFlg: Ausgabestruktur des BACnet-Status

dwObjId: BACnet-Objekt-ID des Loop-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```

stComnMsg : ST_BA_CmnMsg;
stSeqLink : ST_BA_SeqLink;
fbDvc     : FB_BACnet_Device;
    
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_CmnMsg](#) [► 200].

stSeqLink: Daten- und [Befehlsstruktur](#) [► 333] zwischen den einzelnen Sequenzreglern und dem Steuerbaustein [FB_BA_SeqLink](#) [► 171].

fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.33 FB_BACnetLoopSeq1202

BACnet Sequenz-Regler. Im Vergleich zum [FB_BACnetLoopSeq1201](#) [▶ 107] lässt dieser Baustein das Beschreiben folgender BACnet-Properties aus der SPS heraus nicht zu:

- Notification Class
- Minimum Output
- Maximum Output

FB_BACnetLoopSeq1202	
usiMyNum	rPrVal
bEn	rX
bEnActn	rW
bActn	bStaSeqCtrlActv
bEnPropConst	bStaActn
rPropConst	rStaYMin
bEnIntgConst	rStaYMax
rIntgConst	bStaOvrrd
bEnDervConst	bStaOoServ
rDervConst	bStaFlt
bEnEvtEn	bStaInAlm
stEvtEn	bOpLp
bEnErrLmt	bOthFlt
rErrLmt	stStaFlg
bEnTiDly	dwObjId
udiTiDly	bErr
lrYSeqInit	udiErrId
bSync	udiErrArg
lrSync	▷ stCommMsg
lrNz	▷ stSeqLink
udiDampConst	▷ fbDvc
udiMinToMax	
udiMaxToMin	
uiCycCl	
stCommMsg	▷
stSeqLink	▷
fbDvc	▷

Funktionsbeschreibung

Dieser Baustein bildet den Baustein [FB_BA_SeqCtrl](#) [▶ 167] als Loop-Objekt im BACnet ab. Das Loop-Objekt auf der BACnet-Seite hat dabei im Wesentlichen die Aufgabe, Parameter und Werte aus der PLC heraus darzustellen und in die PLC zu übertragen. Die Reglerlogik liegt allein in der PLC im internen [FB_BA_SeqCtrl](#) [▶ 167]. Es ist möglich, die wichtigsten Parameter von beiden Seiten - PLC und BACnet - zu verändern wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der

am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

Wie bei dem "normalen" Sequenzregler vom Typ [FB_BA_SeqCtrl](#) [► 167], welcher eine reine PLC-Lösung darstellt, funktioniert die BACnet-Version ebenfalls nur mit einem Sequenz-Link-Baustein [FB_BA_SeqLink](#) [► 171].

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnActn	Freigabe zum Beschreiben des ReglerWirksinns	
bActn	Property-Wert ReglerWirksinn bActn=TRUE, BACnet direct → ReglerWirksinn: E=X-W (Kühlen) bActn=FALSE, BACnet reverse → ReglerWirksinn: E=W-X (Heizen)	Action (2)
bEnPropConst	Freigabe zum Beschreiben der Proportionalitätskonstante (K-Faktor)	
rPropConst	Property-Wert Proportionalitätskonstante (K-Faktor)	ProporttionalConstant (93)
bEnIntgConst	Freigabe zum Beschreiben der Integrationszeit (I-Anteil)	
rIntgConst	Property-Wert Integrationszeit [s] (I-Anteil)	IntegralConstant (49)
bEnDervConst	Freigabe zum Beschreiben der Vorhaltezeit (D-Anteil)	
rDervConst	Property-Wert Vorhaltezeit [s] (D-Anteil)	DerivativeConstant (26)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnErrLmt	Freigabe zum Beschreiben des Meldewerts "Maximalwert Regelabweichung"	
rErrLmt	Property-Wert Meldewert "Maximalwert Regelabweichung"	ErrorLimit (34)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)

Ein-Ausgänge

VAR_INPUT

```

usiMyNum      : USINT;
bEn           : BOOL;
bEnActn      : BOOL;
bActn        : BOOL;
bEnPropConst : BOOL;
rPropConst   : REAL;
bEnIntgConst : BOOL;
rIntgConst   : REAL;
bEnDervConst : BOOL;
rDervConst   : REAL;
bEnEvtEn     : BOOL;
stEvtEn      : ST_BACnet_EventTransitionBits;
bEnErrLmt    : BOOL;
rErrLmt      : REAL;
bEnTiDly     : BOOL;
udiTiDly     : UDINT;
lrYSeqInit   : LREAL;
bSync        : BOOL;
lrSync       : LREAL;
lrNz         : LREAL;
udiDampConst : UDINT;
udiMinToMax  : UDINT;
udiMaxToMin  : UDINT;
uiCycCl      : UINT;
    
```

usiMyNum: Ordnungszahl des Sequenzreglers

bEn: Regleraktivierung

bEnActn / bActn: Freigabe/Property-Wert ReglerWirksinn: „direct“ oder „reverse“

bEnPropConst / rPropConst: Freigabe/Property-Wert Proportionalitätskonstante (K-Faktor)

bEnIntgConst / rIntgConst: Freigabe/Property-Wert Integrationszeit [s] (I-Anteil)

bEnDervConst / rDervConst: Freigabe/Property-Wert Vorhaltezeit [s] (D-Anteil)

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnErrLmt / rErrLmt: Freigabe/Property-Wert Meldewert Maximalausgabe

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

IrYSeqInit: Start-Synchronisationswert. Ist dieser Regler derjenige, der bei Aktivierung der Regelsequenz als erster freigegeben wird, so wird dieser Regler mit diesem Wert am Ausgang gestartet.

bSync: Setzen des Ausgangs auf *IrSync*.

IrSync: Synchronisationswert, auf den die Stellgröße bei einer steigenden Flanke am Eingang *bSync* gesetzt wird.

IrNz: Neutrale Zone

udiDampConst: Dämpfungszeit des D-Anteiles [s]

udiMinToMax: Steigungsbegrenzung des Reglerausgangs für Anstieg: *udiMinToMax* [s] bezogen auf eine Änderung von *IrMinOut* auf *rMaxOut*.

udiMaxToMin: Steigungsbegrenzung des Reglerausgangs für Abfall: *udiMaxToMin* [s] bezogen auf eine Änderung von *IrMaxOut* auf *rMinOut*.

uiCycCl: Aufrufzyklus des Bausteins als Vielfaches der Zykluszeit. Ein Nulleintrag wird automatisch als *uiCycleCall* =1 gewertet.

Beispiel: *tTaskCycleTime* = 20ms, *uiCtrlCycleCall* =10 -> Der Regelalgorithmus wird alle 200ms aufgerufen. Damit werden aber auch nur alle 200ms die Ausgänge aktualisiert.

VAR_OUTPUT

```
rPrVal      : REAL;
rX          : REAL;
rW          : REAL;
bStaSeqCtrlActv : BOOL;
bStaActn    : BOOL;
rStaYMin    : REAL;
rStaYMax    : REAL;
bStaOvrrd  : BOOL;
bStaOoServ  : BOOL;
bStaFlt     : BOOL;
bStaInAlm   : BOOL;
bOpLp      : BOOL;
bOthFlt     : BOOL;
stStaFlg    : ST_BACnet_StatusFlags;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

rPrVal: Stellwert - Reglerausgang

rX: Aktuell gültiger Istwert aus dem BACnet gelesen.

rW: Aktuell gültiger Sollwert aus dem BACnet gelesen.

bStaSeqCtrlActv: Sequenzregler ist freigegeben und betriebsbereit (nicht im Fehlerzustand).

bStaActn: Zeigt den Zustand des Wirksinns an.

bStaYMin: Zeigt den unteren Wert der Regler-Ausgangsbegrenzung an.

- bStaYMax:** Zeigt den oberen Wert der Regler-Ausgangsbegrenzung an.
- bStaOvrrd:** Zeigt den Zustand des Statusflags „Override“ des Loop-Objekts an.
- bStaOoServ:** Zeigt den Zustand des Statusflags „OutOfService“ des Loop-Objekts an.
- bStaFlt:** Zeigt den Zustand des Statusflags „Fault“ des Loop-Objekts an.
- bStalnAlm:** Zeigt den Zustand des Statusflags „InAlarm“ des Loop-Objekts an.
- bOpLp:** Zeigt den Zustand „OpenLoop“ des Loop-Objekt-Properties „Reliability“ (Zuverlässigkeit) an.
- bOthFlt:** Zeigt den Zustand „OtherFault“ des Loop-Objekt-Properties „Reliability“ (Zuverlässigkeit) an.
- stStaFig:** Ausgabestruktur des BACnet-Status
- dwObjId:** BACnet-Objekt-ID des Loop-Objekts
- bErr:** Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.
- udiErrId / udiErrArg:** Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[▶ 340\]](#).

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
stSeqLink : ST_BA_SeqLink;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[▶ 331\]](#) zum Melde-Sammelbaustein [FB_BA_CmnMsg \[▶ 200\]](#).

stSeqLink: Daten- und [Befehlsstruktur \[▶ 333\]](#) zwischen den einzelnen Sequenzreglern und dem Steuerbaustein [FB_BA_SeqLink \[▶ 171\]](#).

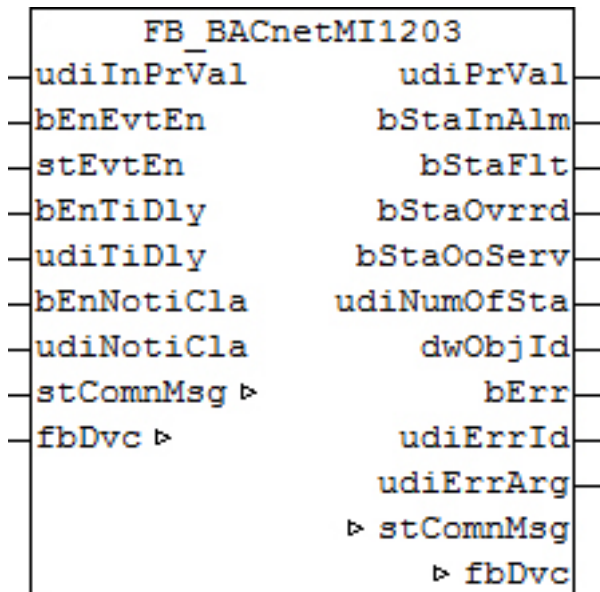
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.34 FB_BACnetMI1203

BACnet Multistate-Input



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Multistate-Input-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

udiInPrVal : UDINT
bEnEvtEn   : BOOL;
stEvtEn    : ST_BACnet_EventTransitionBits;
bEnTiDly   : BOOL;
udiTiDly   : UDINT;
bEnNotiCla : BOOL;
udiNotiCla : UDINT;

```

udiInPrVal: Eingabewert von der Hardware. Der Wertebereich der Status ist 1 bis 1000. Unter Umständen müssen die Eingangssignale der Hardware (z. B. n digitale Eingänge für einen Drehschalter) durch eine Logik so vorkodiert, dass Werte im Rahmen des gültigen Bereiches dargestellt werden.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

udiPrVal      : UDINT;
bStaInAlm    : BOOL;
bStaFlt      : BOOL;
bStaOvrrd    : BOOL;
bStaOoServ   : BOOL;
udiNumOfSta  : UDINT;
dwObjId      : DWORD;
bErr         : BOOL;
udiErrId     : UDINT;
udiErrArg    : UDINT;
    
```

udiPrVal: Aktueller Wert des Multistate-Input-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Multistate-Input-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Multistate-Input-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Overriden“ des Analog-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Multistate-Input-Objekts an.

udiNumOfSta: Zeigt die eingestellte Anzahl der Status an (NumberOfStates, Property Id 74).

dwObjId: BACnet-Objekt-ID des Multistate-Input-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```

stComnMsg : ST_BA_CmnMsg;
fbDvc     : FB_BACnet_Device;
    
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_CmnMsg \[► 200\]](#).

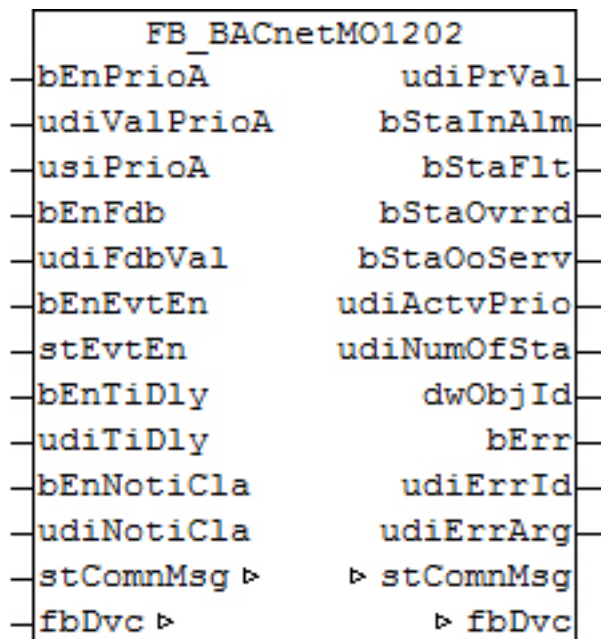
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.35 FB_BACnetMO1202

BACnet Multistate-Output



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Multistate-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleine" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetMO1203](#) | [1201](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
udiValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
udiValPrioA   : UDINT;
usiPrioA      : USINT;
bEnFdb       : BOOL;
udiFdbVal    : UDINT;
bEnEvtEn     : BOOL;

```

```

stEvtEn      : ST_BACnet_EventTransitionBits;
bEnTiDly    : BOOL;
udiTiDly    : UDINT;
bEnNotiCla  : BOOL;
udiNotiCla  : UDINT;

```

bEnPrioA: Freigabe zum Beschreiben

udiValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnFdbVal / udiFdbVal: Freigabe/Property-Wert Feedback-Value. Dieser Wert wird aus der PLC heraus auf das Property FeedbackValue (Property Id 40) geschrieben, jedoch **nicht** per ADS, sondern per zyklischem Interface.

Es dient einer Rückmelde-Überwachung: Ist *bEnFdbVal* auf TRUE gesetzt, so muss an *udiFdbVal* innerhalb der Zeit, die im BACnet unter TIME-Delay (Property Id 113) eingetragen ist, den gleichen Wert haben, wie der Ausgang Present-Value. Ansonsten wird im EventState (Property Id 36) durch den Eintrag "offnormal" eine Rückmelde-Diskrepanz angezeigt. Ist *bEnFdbVal* auf FALSE gesetzt, so wird intern *udiFdbVal* gleich dem Present-Value gesetzt, und es kann so niemals zu einer Rückmeldediskrepanz kommen.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

udiPrVal     : UDINT;
bStaInAlm   : BOOL;
bStaFlt     : BOOL;
bStaOvrrd   : BOOL;
bStaOoServ  : BOOL;
udiActvPrio : UDINT;
udiNumOfSta : UDINT;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;

```

udiPrVal: Aktueller Wert des Multistate-Output-Objekts - direkt aus dem BACnet eingelesen. Dieser Wert bedarf u.U. einer nachfolgenden Logik, wenn er mit einer Hardware verknüpft werden soll (Anzeigetafel, etc.).

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Multistate-Output-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Multistate-Output-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Multistate-Output-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Multistate-Output-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

udiNumOfSta: Zeigt die eingestellte Anzahl der Status an (NumberOfStates, Property Id 74).

dwObjId: BACnet-Objekt-ID des Multistate-Output-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```

stComnMsg   : ST_BA_ComnMsg;
fbDvc       : FB_BACnet_Device;

```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_ComMsg](#) [► 200].

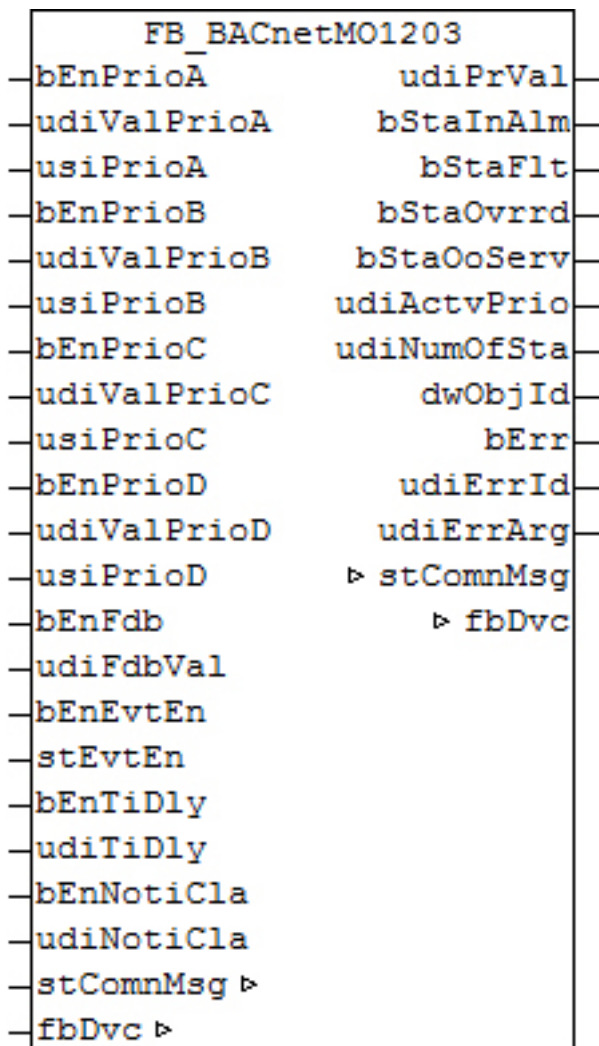
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.36 FB_BACnetMO1203

BACnet Multistate-Output



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Multistate-Output-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "große" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetMO1202](#) [▶ 117]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt

(steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
udiValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
udiValPrioB	Zu schreibender Wert	Priority-Array (87)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
udiValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
udiValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
udiValPrioA   : UDINT;
usiPrioA      : USINT;
bEnPrioB      : BOOL;
udiValPrioB   : UDINT;
usiPrioB      : USINT;
bEnPrioC      : BOOL;
udiValPrioC   : UDINT;
usiPrioC      : USINT;
bEnPrioD      : BOOL;
udiValPrioD   : UDINT;
usiPrioD      : USINT;
bEnFdb        : BOOL;
udiFdbVal     : UDINT;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
bEnTiDly      : BOOL;
udiTiDly      : UDINT;
bEnNotiCla    : BOOL;
udiNotiCla    : UDINT;
    
```

bEnPrioA: Freigabe zum Beschreiben

udiValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

udiValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

udiValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16).

bEnPrioD: Freigabe zum Beschreiben

udiValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bEnFdbVal / udiFdbVal: Freigabe/Property-Wert Feedback-Value. Dieser Wert wird aus der PLC heraus auf das Property FeedbackValue (Property Id 40) geschrieben, jedoch **nicht** per ADS, sondern per zyklischem Interface.

Es dient einer Rückmelde-Überwachung: Ist *bEnFdbVal* auf TRUE gesetzt, so muss an *udiFdbVal* innerhalb der Zeit, die im BACnet unter TIME-Delay (Property Id 113) eingetragen ist, den gleichen Wert haben, wie der Ausgang Present-Value. Ansonsten wird im EventState (Property Id 36) durch den Eintrag "offnormal" eine Rückmelde-Diskrepanz angezeigt. Ist *bEnFdbVal* auf FALSE gesetzt, so wird intern *udiFdbVal* gleich dem Present-Value gesetzt, und es kann so niemals zu einer Rückmeldediskrepanz kommen.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

udiPrVal      : UDINT;
bStaInAlm    : BOOL;
bStaFlt      : BOOL;
bStaOvrrd    : BOOL;
bStaOoServ   : BOOL;
udiActvPrio  : UDINT;
udiNumOfSta  : UDINT;
dwObjId      : DWORD;
bErr         : BOOL;
udiErrId     : UDINT;
udiErrArg    : UDINT;

```

udiPrVal: Aktueller Wert des Multistate-Output-Objekts - direkt aus dem BACnet eingelesen. Dieser Wert bedarf u.U. einer nachfolgenden Logik, wenn er mit einer Hardware verknüpft werden soll (Anzeigetafel, etc.).

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Multistate-Output-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Multistate-Output-Objekts an.

bStaOvrrd: Zeigt den Zustand des Statusflags „Overridden“ des Multistate-Output-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Multistate-Output-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

udiNumOfSta: Zeigt die eingestellte Anzahl der Status an (NumberOfStates, Property Id 74).

dwObjId: BACnet-Objekt-ID des Multistate-Output-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc      : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[▶ 331\]](#) zum Melde-Sammelbaustein [FB_BA_ComMsg \[▶ 200\]](#).

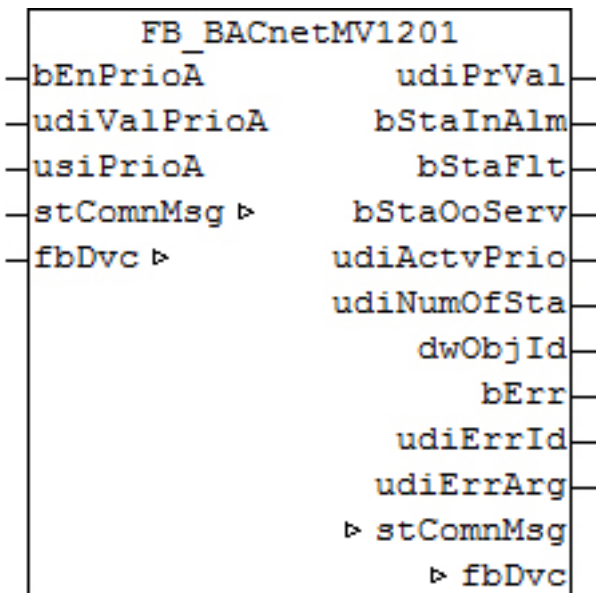
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.37 FB_BACnetMV1201

BACnet Multistate-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Multistate-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "kleinste" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetMV1202 \[▶ 125\]](#)
- [FB_BACnetMV1203 \[▶ 127\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
udiValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	

Ein-Ausgänge

VAR_INPUT

```
bEnPrioA      : BOOL;
udiValPrioA   : UDINT;
usiPrioA      : USINT;
```

bEnPrioA: Freigabe zum Beschreiben

udiValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

VAR_OUTPUT

```
udiPrVal      : UDINT;
bStaInAlm     : BOOL;
bStaFlt       : BOOL;
bStaOoServ    : BOOL;
udiActvPrio   : UDINT;
udiNumOfSta   : UDINT;
dwObjId       : DWORD;
bErrs         : BOOL;
udiErrId      : UDINT;
udiErrArg     : UDINT;
```

udiPrVal: Aktueller Wert des Multistate-Value-Objekts - direkt aus dem BACnet eingelesen

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Multistate-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Multistate-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Multistate-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

udiNumOfSta: Zeigt die eingestellte Anzahl der Status an (NumberOfStates, Property Id 74).

dwObjId: BACnet-Objekt-ID des Multistate-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_CmnMsg](#) [► 200].

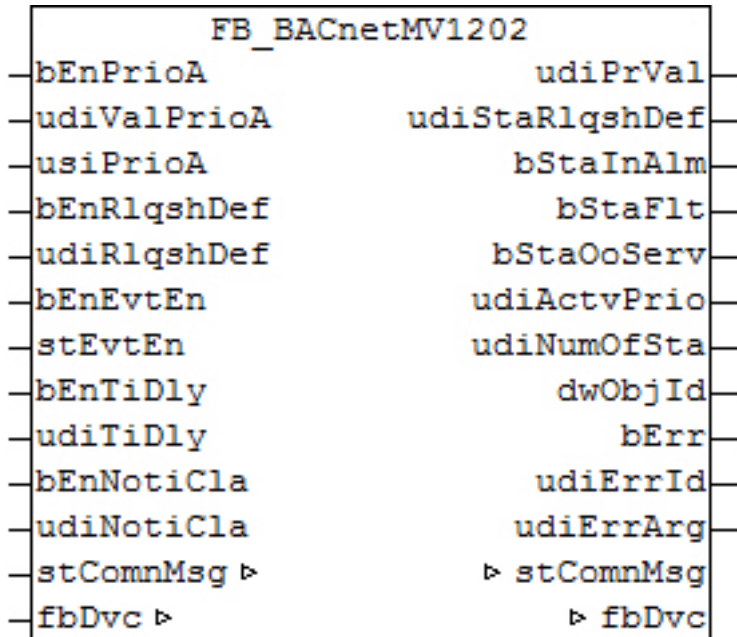
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.38 FB_BACnetMV1202

BACnet Multistate-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Multistate-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "mittlere" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetMV1201](#) [▶ 123]
- [FB_BACnetMV1203](#) [▶ 127]

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
udiValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnRlqshDef	Freigabe zum Beschreiben des Relinquish-Defaults	

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
udiRlqshDef	Zu schreibender Wert auf das Property Relinquish-Default	Relinquish-Default (104)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
udiValPrioA   : UDINT;
usiPrioA      : USINT;
bEnRlqshDef  : BOOL;
udiRlqshDef   : UDINT;
bEnEvtEn     : BOOL;
stEvtEn      : ST_BACnet_EventTransitionBits;
bEnTiDly     : BOOL;
udiTiDly     : UDINT;
bEnNotiCla   : BOOL;
udiNotiCla   : UDINT;

```

bEnPrioA: Freigabe zum Beschreiben

udiValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnRlqshDef / udiRlqshDef: Freigabe/Property-Wert Relinquish-Default

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

udiPrVal      : UDINT;
udiStaRlqshDef : UDINT;
bStaInAlm     : BOOL;
bStaFlt       : BOOL;
bStaOoServ    : BOOL;
udiActvPrio   : UDINT;
udiNumOfSta   : UDINT;
dwObjId       : DWORD;
bErr          : BOOL;
udiErrId      : UDINT;
udiErrArg     : UDINT;

```

udiPrVal: Aktueller Wert des Multistate-Value-Objekts - direkt aus dem BACnet eingelesen

udiStaRlqshDef: Zeigt den Zustand des Relinquish-Default an.

bStaInAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Multistate-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Multistate-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Multistate-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

udiNumOfSta: Zeigt die eingestellte Anzahl der Status an (NumberOfStates, Property Id 74).

dwObjId: BACnet-Objekt-ID des Multistate-Value-Objekts

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [▶ 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_ConnMsg;
fbDvc     : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [▶ 331] zum Melde-Sammelbaustein [FB_BA ComMsg](#) [▶ 200].

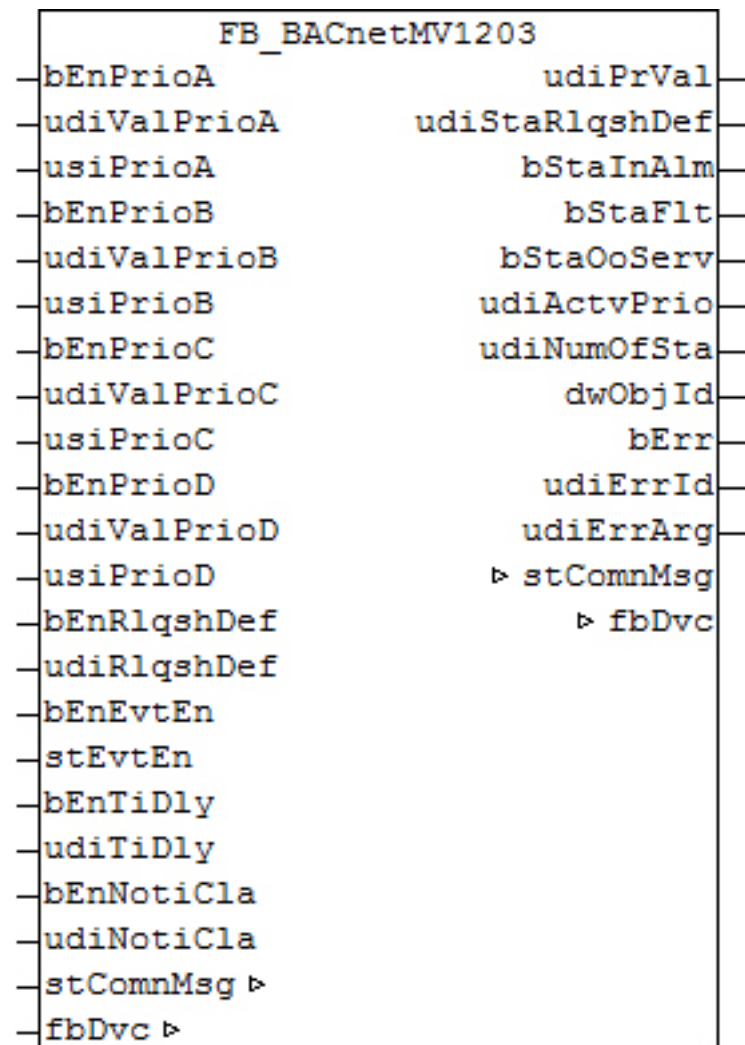
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.39 FB_BACnetMV1203

BACnet Multistate-Value



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Multistate-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Bei diesem Baustein handelt es sich vom Funktionsumfang her um die "größte" Version.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetMV1201 \[► 123\]](#)
- [FB_BACnetMV1202 \[► 125\]](#)
- [FB_BACnetMVDisplay \[► 130\]](#)
- [FB_BACnetMVSetpoint \[► 131\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnPrioA	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioA</i> eingetragen ist	
udiValPrioA	Zu schreibender Wert	Priority-Array (87)
usiPrioA	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioB	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioB</i> eingetragen ist	
udiValPrioB	Zu schreibender Wert	Priority-Array (87)
usiPrioB	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioC	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioC</i> eingetragen ist	
udiValPrioC	Zu schreibender Wert	Priority-Array (87)
usiPrioC	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnPrioD	Freigabe zum Beschreiben des Priority-Arrays auf die Priorität, welche unter <i>usiPrioD</i> eingetragen ist	
udiValPrioD	Zu schreibender Wert	Priority-Array (87)
usiPrioD	Auswahl der zu beschreibenden Priorität (1..16) im Priority-Array	
bEnRlqshDef	Freigabe zum Beschreiben des Relinquish-Defaults	
udiRlqshDef	Zu schreibender Wert auf das Property Relinquish-Default	Relinquish-Default (104)
bEnEvtEn	Freigabe zum Beschreiben des Bitmusters EventEnable	
stEvtEn	Property-Wert Bitmuster EventEnable	EventEnable (35)
bEnTiDly	Freigabe zum Beschreiben der Meldeverzögerung [s]	
udiTiDly	Property-Wert Meldeverzögerung [s]	TimeDelay (113)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

bEnPrioA      : BOOL;
udiValPrioA   : UDINT;
usiPrioA      : USINT;
bEnPrioB      : BOOL;
udiValPrioB   : UDINT;
usiPrioB      : USINT;
bEnPrioC      : BOOL;
udiValPrioC   : UDINT;
usiPrioC      : USINT;
bEnPrioD      : BOOL;
udiValPrioD   : UDINT;
usiPrioD      : USINT;
bEnRlqshDef   : BOOL;
udiRlqshDef   : UDINT;
bEnEvtEn      : BOOL;
stEvtEn       : ST_BACnet_EventTransitionBits;
bEnTiDly      : BOOL;
udiTiDly      : UDINT;
bEnNotiCla    : BOOL;
udiNotiCla    : UDINT;

```

bEnPrioA: Freigabe zum Beschreiben

udiValPrioA: Wert, der in das Priority-Array auf die Priorität *usiPrioA* geschrieben wird.

usiPrioA: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioB: Freigabe zum Beschreiben

udiValPrioB: Wert, der in das Priority-Array auf die Priorität *usiPrioB* geschrieben wird.

usiPrioB: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioC: Freigabe zum Beschreiben

udiValPrioC: Wert, der in das Priority-Array auf die Priorität *usiPrioC* geschrieben wird.

usiPrioC: Auswahl der zu beschreibenden Priorität (1..16)

bEnPrioD: Freigabe zum Beschreiben

udiValPrioD: Wert, der in das Priority-Array auf die Priorität *usiPrioD* geschrieben wird.

usiPrioD: Auswahl der zu beschreibenden Priorität (1..16)

bEnRlqshDef / udiRlqshDef: Freigabe/Property-Wert Relinquish-Default.

bEnEvtEn / stEvtEn: Freigabe/Property-Wert Bitmuster EventEnable

bEnTiDly / udiTiDly: Freigabe/Property-Wert Meldeverzögerung [s]

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

udiPrVal      : UDINT;
udiStaRlqshDef : UDINT;
bStaInAlm     : BOOL;
bStaFlt       : BOOL;
bStaOoServ    : BOOL;
udiActvPrio   : UDINT;
udiNumOfSta   : UDINT;
dwObjId       : DWORD;
bErr          : BOOL;
udiErrId      : UDINT;
udiErrArg     : UDINT;

```

udiPrVal: Aktueller Wert des Multistate-Value-Objekts - direkt aus dem BACnet eingelesen

udiStaRlqshDef: Zeigt den Zustand des Relinquish-Default an.

bStalnAlm: Zeigt den Zustand des Statusflags „InAlarm“ des Multistate-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Multistate-Value-Objekts an.

bStaOoServ: Zeigt den Zustand des Statusflags „OutOfService“ des Multistate-Value-Objekts an.

udiActvPrio: Zeigt an, welche Priorität die aktive ist.

udiNumOfSta: Zeigt die eingestellte Anzahl der Status an (NumberOfStates, Property Id 74).

dwObjId: BACnet-Objekt-ID des Multistate-Value-Objekts.

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_CmnMsg;
fbDvc      : FB_BACnet_Device;
```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_CmnMsg](#) [► 200].

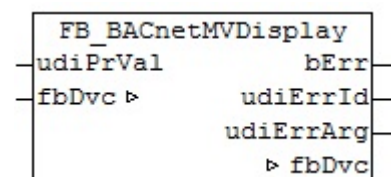
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.40 FB_BACnetMVDisplay

BACnet MultiState-Value Objekt, welches zur Anzeige eines Wertes aus der PLC im BACnet verwendet werden kann.



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-MultiState-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetMV1201](#) [► 123]
- [FB_BACnetMV1202](#) [► 125]
- [FB_BACnetMV1203](#) [► 127]
- [FB_BACnetMVSetpoint](#) [► 131]

Ein-Ausgänge

VAR_INPUT

```
udiPrVal : UDINT;
```

udiPrVal: Wert aus der PLC, der bei Wertänderung auf das BACnet Property Present Value geschrieben wird.

VAR_OUTPUT

```
bErrs      : BOOL;
udiErrId   : UDINT;
udiErrArg  : UDINT;
```

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
fbDvc      : FB_BACnet_Device;
```

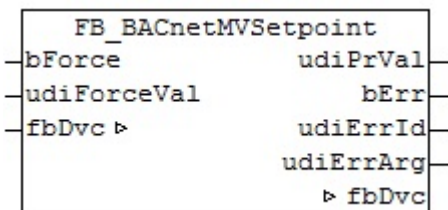
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.41 FB_BACnetMVSetpoint

BACnet MultiState-Value Objekt, welches das BACnet Property Present Value in der PLC abbildet.



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-MultiState-Value-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Alternativ stehen folgende Versionen zur Verfügung:

- [FB_BACnetMV1201 \[► 123\]](#)
- [FB_BACnetMV1202 \[► 125\]](#)
- [FB_BACnetMV1203 \[► 127\]](#)
- [FB_BACnetMVDisplay \[► 130\]](#)

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface.

Mit jeder steigenden Flanke an bForce wird einmalig der Wert von udiForceVal auf den Present Value des MV-Objekts geschrieben.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bForce	Mit jeder steigenden Flanke an bForce wird einmalig der Wert von udiForceVal auf den Present Value des MV-Objekts geschrieben.	

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
udiForceVal	Zu schreibender Wert auf das BACnet Property Present Value	Present Value (85)

Ein-Ausgänge

VAR_INPUT

```
bForce      : BOOL;
udiForceVal : UDINT;
```

bForce: Mit jeder steigenden Flanke an bForce wird einmalig der Wert von rForceVal auf den Present Value des AV-Objekts geschrieben.

udiForceVal: Wert, der auf das BACnet Property Present Value geschrieben wird.

VAR_OUTPUT

```
udiPrVal    : UDINT;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

udiPrVal: Aktueller Wert des BACnet Property Present Value - direkt aus dem BACnet eingelesen

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
fbDvc      : FB_BACnet_Device;
```

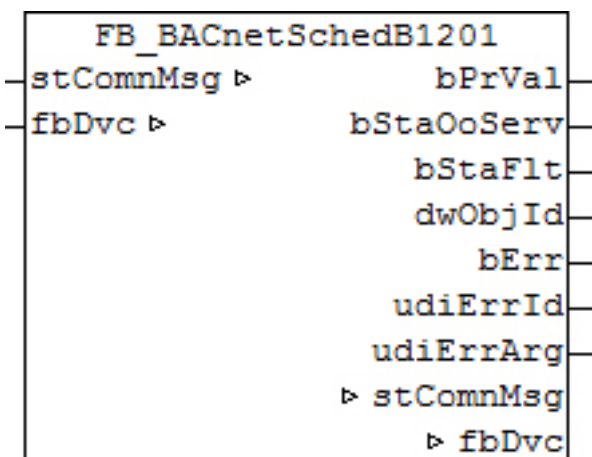
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.42 FB_BACnetSchedB1201

BACnet Scheduler vom Ausgabebetyp BOOL



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Scheduler-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung. Der Ausgabotyp des Objekts ist vom Typ BOOL.

Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Ein-Ausgänge

VAR_OUTPUT

```
bPrVal      : BOOL;
bStaOoServ  : BOOL;
bStaFlt     : BOOL;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bPrVal: Ausgabewert des Zeitplans in Anhängigkeit des aktuell eingestellten Datums/Uhrzeit und der Zeitplaneinträge.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Binary-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Value-Objekts an.

dwObjId: BACnet-Objekt-ID des Binary-Value-Objekts.

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [▶ 340].

VAR_IN_OUT

```
stComnMsg  : ST_BA_CmnMsg;
fbDvc      : FB_BACnet_Device;
```

stComnMsg: Referenz auf die Verbindungsstruktur [▶ 331] zum Melde-Sammelbaustein FB_BA_CmnMsg [▶ 200].

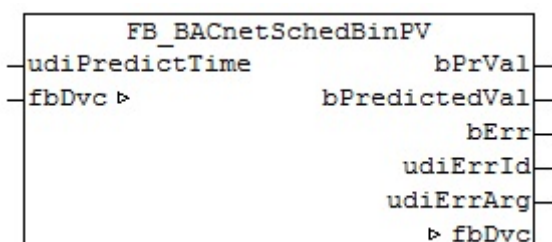
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.43 FB_BACnetSchedBinPV

BACnet Scheduler vom Typ Binary Present Value mit der Funktion "Vorausberechnende Ein- und Ausschaltzeit".



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Scheduler-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung. Der Ausgabebetyp des Objekts ist vom Typ BOOL.

Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Ein-Ausgänge

VAR_INPUT

```
udiPredictTime : UDINT;
```

udiPredictTime: Vorausberechnete Ein- und Ausschaltzeit in Sekunden. Der hier angegebene Zeitwert schaltet in Abhängigkeit der Zeitschaltkanäle des BACnet Property WeeklySchedule den Ausgang bPredictedVal vorher ein bzw. aus.

VAR_OUTPUT

```
bPrVal          : BOOL;
bPredictedVal   : BOOL;
bErr            : BOOL;
udiErrId        : UDINT;
udiErrArg       : UDINT;
```

bPrVal: Ausgabewert des Zeitplans in Abhängigkeit des aktuell eingestellten Datums/Uhrzeit und der Zeitplaneinträge.

bPredictedVal: Ausgang der in Abhängigkeit der Ein- und Ausschaltzeit udiPredictTime und des BACnet Property WeeklySchedule geschaltet wird.

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```
fbDvc          : FB_BACnet_Device;
```

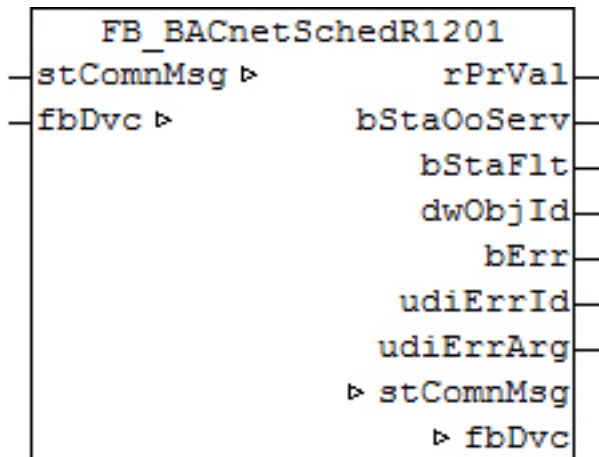
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.44 FB_BACnetSchedR1201

BACnet Scheduler vom Ausgabebetyp REAL



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Scheduler-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung. Der Ausgabebetyp des Objekts ist vom Typ REAL.

Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Ein-Ausgänge

VAR_OUTPUT

```

rPrVal      : REAL;
bStaOoServ  : BOOL;
bStaFlt     : BOOL;
dwObjId     : DWORD;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
  
```

rPrVal: Ausgabewert des Zeitplans in Anhängigkeit des aktuell eingestellten Datums/Uhrzeit und der Zeitplaneinträge.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Binary-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Value-Objekts an.

dwObjId: BACnet-Objekt-ID des Binary-Value-Objekts.

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

VAR_IN_OUT

```

stComnMsg   : ST_BA_CmnMsg;
fbDvc       : FB_BACnet_Device;
  
```

stComnMsg: Referenz auf die [Verbindungsstruktur \[► 331\]](#) zum Melde-Sammelbaustein [FB_BA_CmnMsg \[► 200\]](#).

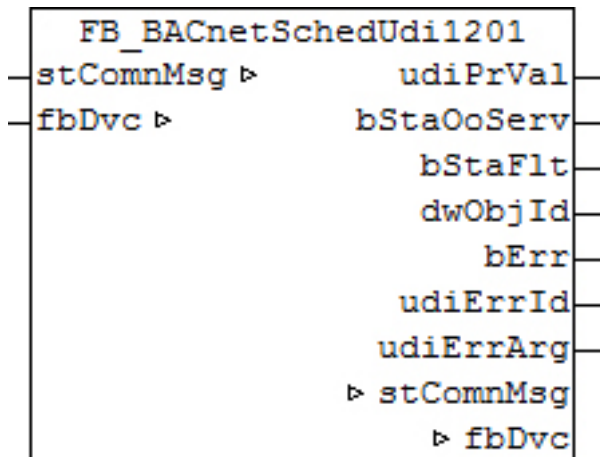
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.45 FB_BACnetSchedUdi1201

BACnet Scheduler vom Ausgabebetyp UDINT



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Scheduler-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zu Verfügung. Der Ausgabebetyp des Objekts ist vom Typ UDINT.

Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Ein-Ausgänge

VAR_OUTPUT

```
udiPrVal   : UDINT;
bStaOoServ : BOOL;
bStaFlt    : BOOL;
dwObjId    : DWORD;
bErr       : BOOL;
udiErrId   : UDINT;
udiErrArg  : UDINT;
```

udiPrVal: Ausgabewert des Zeitplans in Anhängigkeit des aktuell eingestellten Datums/Uhrzeit und der Zeitplaneinträge.

bStaOvrrd: Zeigt den Zustand des Statusflags „Override“ des Binary-Value-Objekts an.

bStaFlt: Zeigt den Zustand des Statusflags „Fault“ des Binary-Value-Objekts an.

dwObjId: BACnet-Objekt-ID des Binary-Value-Objekts.

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
stComnMsg : ST_BA_ComnMsg;
fbDvc     : FB_BACnet_Device;
```


stCommMsg: Referenz auf die [Verbindungsstruktur \[▶ 331\]](#) zum Melde-Sammelbaustein [FB_BA_ComMsg \[▶ 200\]](#).

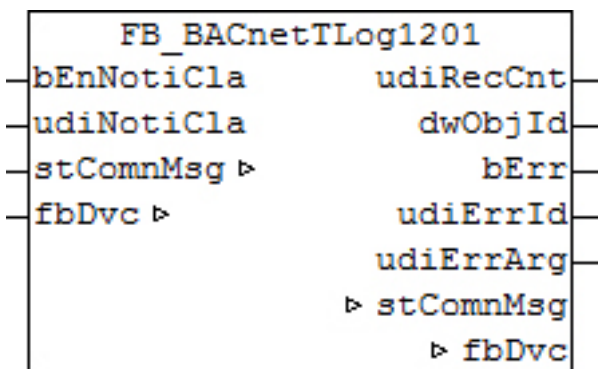
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.46 FB_BACnetTLog1201

BACnet Trend-Log



Funktionsbeschreibung

Dieser Baustein generiert ein BACnet-Trend-Log-Objekt und stellt innerhalb der PLC Schreibe- und Lesevariablen für das Objekt zur Verfügung.

Es ist möglich, die Parameter, welche von der PLC aus beschreibbar sind, auch von der BACnet-Seite her zu beschreiben, wobei die letzte Änderung immer die gültige ist. Die Übertragung dieser Parameter erfolgt aus der PLC heraus per ADS und in die PLC hinein per zyklischem Interface. PLC-seitig haben alle Parameter zusätzlich einen Enable-Eingang: Ist der betreffende Enable-Eingang erstmalig gesetzt (steigende Flanke), so wird der am Parameter-Eingang stehende Wert in jedem Fall übertragen, im weiteren Verlauf – bei gesetztem Enable – nur noch bei Werteänderung, um so den ADS-Verkehr zu mindern. Ist der Enable-Eingang nicht gesetzt, so erfolgt keine Übertragung.

PLC-Variable	Bemerkung	BACnet-Property (Property ID)
bEnNotiCla	Freigabe zum Beschreiben der Meldeklasse	
udiNotiCla	Property-Wert Meldeklasse	NotificationClass (17)

Ein-Ausgänge

VAR_INPUT

```

bEnNotiCla    : BOOL;
udiNotiCla    : UDINT;
    
```

bEnNotiCla / udiNotiCla: Freigabe/Property-Wert Meldeklasse

VAR_OUTPUT

```

udiRecCnt : UDINT;
dwObjId   : DWORD;
bErr      : BOOL;
udiErrId  : UDINT;
udiErrArg : UDINT;

```

udiRecCnt: Anzahl der geloggen Werte (Property ID 141).

dwObjId: BACnet-Objekt-ID des Trend-Log-Objekts.

bErr: Zeigt allgemein einen Fehler im Baustein an. Die Ursache kann in BACnet, im ADS-Datenaustausch oder auch an falscher Parametrierung liegen.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```

stComnMsg : ST_BA_ComnMsg;
fbDvc     : FB_BACnet_Device;

```

stComnMsg: Referenz auf die [Verbindungsstruktur](#) [► 331] zum Melde-Sammelbaustein [FB_BA_ComMsg](#) [► 200].

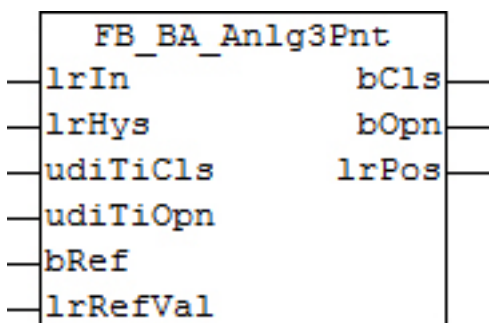
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.47 FB_BA_Anlg3Pnt

Der Funktionsbaustein ist für die Ansteuerung von Dreipunktstellantrieben zur Betätigung von Ventilen oder Klappen vorgesehen.

**Funktionsbeschreibung**

Der Baustein wandelt ein stetiges Stellsignal für die Positionierung des Stellantriebs in die binären Befehle für das Auf- und Zufahren um.

Ist die Abweichung zwischen dem Positionssollwert *lrIn* und dem errechneten Positionswert *lrPos* des Stellorgans größer als der mit der Variable *lrHys/2* eingestellte Schwellwert, dann beginnt der Funktionsbaustein abhängig von dem Betrag der Regelabweichung durch Schalten der Ausgänge *bOpn* oder *bCls* die Position zu korrigieren:

	bOpn	bCls
$lrIn - lrPos > lrHys/2$	TRUE	FALSE
$lrIn - lrPos < - lrHys/2$	FALSE	TRUE

Die Eingabe *lrIn* wird intern automatisch auf den Bereich von 0..100% begrenzt.

Eine steigende Flanke an *bRef* löst einen Referenzierbefehl (Setzen der errechneten Istposition auf *lrRefVal*) aus.

Falls der Antrieb über Endlagenschalter verfügt, können diese auch direkt mittels Digitaleingang erfasst und zur Referenzierung an *bRef* verwendet werden.

Ein-/Ausgänge

VAR_INPUT

```
lrIn      : LREAL;
lrHys     : LREAL;
udiTiCls  : UDINT;
udiTiOpn  : UDINT;
bRef      : BOOL;
lrRefVal  : LREAL;
```

lrIn: Sollwert für die Position des Stellantriebs [0 - 100%]

lrHys: Hysterese für die Position des Stellantriebs [0 - 100%]

udiTiCls: Fahrzeit des Stellantriebs von auf nach zu [ms]

udiTiOpn: Fahrzeit des Stellantriebs von zu nach auf [ms]

bRef: Flanke referenziert den internen Positionsspeicher des Antriebs auf Wert von *lrRefVal* [0 - 100%]

lrRefVal: Wert für die Referenzierung des Stellantrieb mit *bRef* [0 - 100%]

VAR_OUTPUT

```
bCls      : BOOL;
bOpn      : BOOL;
lrPos     : LREAL;
```

bCls: Ausgang für das Zufahren des Stellantriebs

bOpn: Ausgang für das Auffahren des Stellantriebs

lrPos: aktuelle errechnete Position des Stellantriebs [0 - 100%]

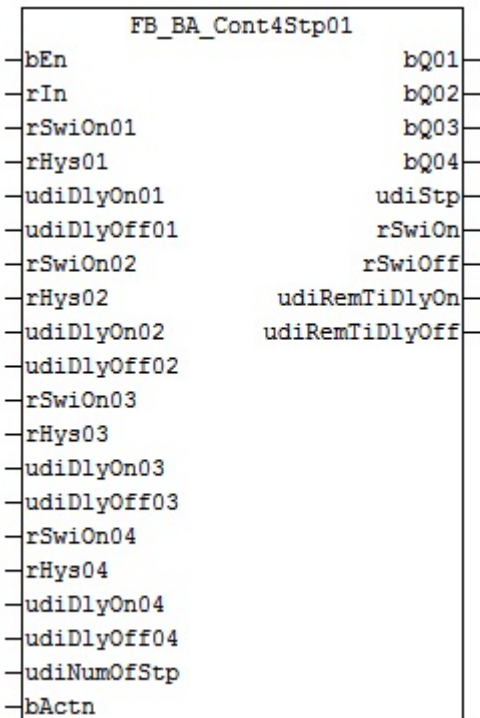
Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.48 FB_BA_Cont4Stp01

Stufenschalter mit 4 Stufen

Schnittstelle



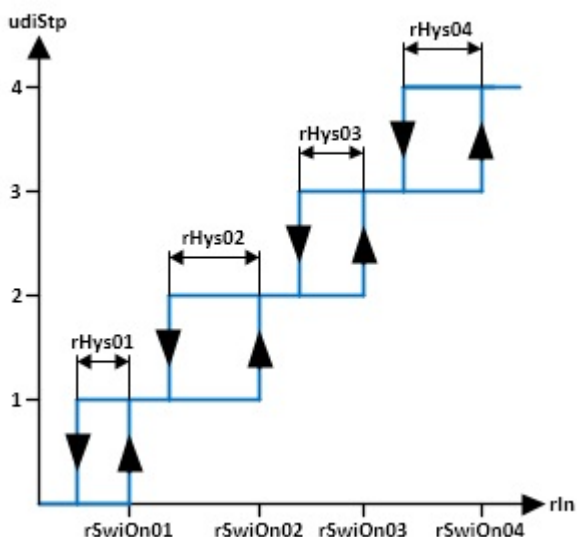
Funktionsbeschreibung

Der Funktionsbaustein ermittelt abhängig vom Eingangssignal die resultierenden Schaltstufen eines mehrstufigen Aggregats.

Es sind vier Einschaltsschwellen und vier Hysteresen parametrierbar.

Diagramm 01

Wirksinn $bActn = FALSE = Reverse = Heizen$

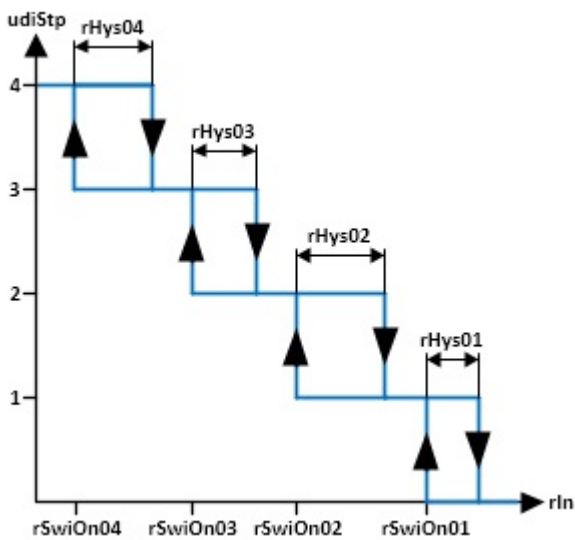


udiStp	udiNum OfStp	rSwiOn	rSwiOff	udiRem TiDlyOn	udiRem TiDlyOff	bQ01	bQ02	bQ03	bQ04
0	0	rSwiOn0 1	rSwiOn0 1 - rHys01	udiDlyOn 01	0	FALSE	FALSE	FALSE	FALSE

1	>= 1	rSwiOn0 2	rSwiOn0 1 - rHys01	udiDlyOn 02	udiDlyOff 01	TRUE	FALSE	FALSE	FALSE
2	>= 2	rSwiOn0 3	rSwiOn0 2 - rHys02	udiDlyOn 03	udiDlyOff 02	TRUE	TRUE	FALSE	FALSE
3	>= 3	rSwiOn0 4	rSwiOn0 3 - rHys03	udiDlyOn 04	udiDlyOff 03	TRUE	TRUE	TRUE	FALSE
4	>= 4	rSwiOn0 4	rSwiOn0 4 - rHys04	0	udiDlyOff 04	TRUE	TRUE	TRUE	TRUE

Diagramm 02

Wirksinn *bActn* =TRUE = Direct = Kühlen



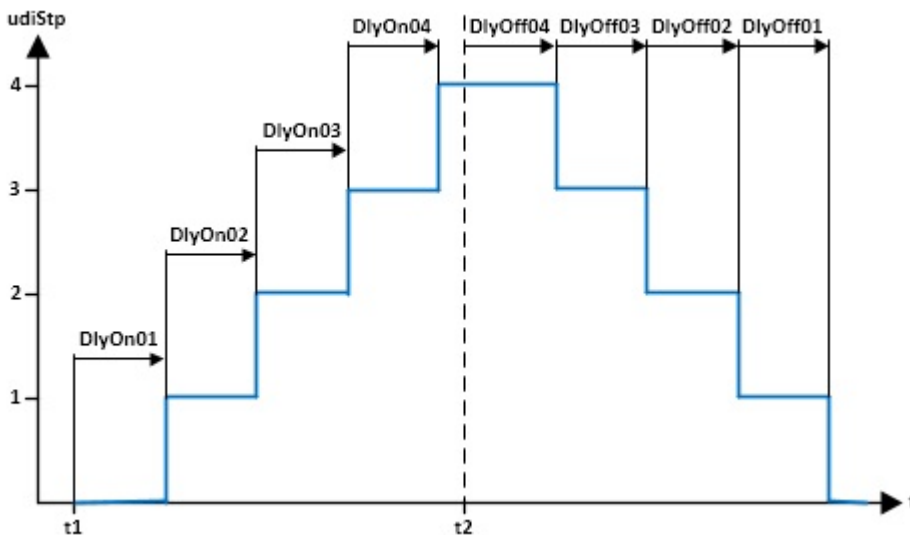
udiStp	udiNum OfStp	rSwiOn	rSwiOff	udiRem TiDlyOn	udiRem TiDlyOff	bQ01	bQ02	bQ03	bQ04
0	0	rSwiOn0 1	rSwiOn0 1 + rHys01	udiDlyOn 01	0	FALSE	FALSE	FALSE	FALSE
1	>= 1	rSwiOn0 2	rSwiOn0 1 + rHys01	udiDlyOn 02	udiDlyOff 01	TRUE	FALSE	FALSE	FALSE
2	>= 2	rSwiOn0 3	rSwiOn0 2 + rHys02	udiDlyOn 03	udiDlyOff 02	TRUE	TRUE	FALSE	FALSE
3	>= 3	rSwiOn0 4	rSwiOn0 3 + rHys03	udiDlyOn 04	udiDlyOff 03	TRUE	TRUE	TRUE	FALSE
4	4	rSwiOn0 4	rSwiOn0 4 + rHys04	0	udiDlyOff 04	TRUE	TRUE	TRUE	TRUE

Diagramm 03

Zeitverhalten der Ein- und Ausschaltverzögerungen

Zum Zeitpunkt t1 springt *rIn* von £*rSwiOn01* auf *rSwiOn04*

Zum Zeitpunkt t2 springt *rIn* von *rSwiOn04* auf £ *rSwiOn01* – *rHys01*



Ein-/Ausgänge

VAR_INPUT

```

bEn      : BOOL;
rIn      : REAL;
rSwiOn01 : REAL;
rHys01   : REAL;
udiDlyOn01 : UDINT;
udiDlyOff01 : UDINT;
rSwiOn02 : REAL;
rHys02   : REAL;
udiDlyOn02 : UDINT;
udiDlyOff02 : UDINT;
rSwiOn03 : REAL;
rHys03   : REAL;
udiDlyOn03 : UDINT;
udiDlyOff03 : UDINT;
rSwiOn04 : REAL;
rHys04   : REAL;
udiDlyOn04 : UDINT;
udiDlyOff04 : UDINT;
udiNumOfStp : UDINT;
bActn    : BOOL;

```

bEn: Generelle Freigabe des Funktionsbausteins. Ist $bEn = FALSE$, so sind sämtliche Ausgänge auf 0 gesetzt.

rIn: Eingangswert von dem der Schaltzustand abgeleitet wird.

rSwiOn01: Einschaltpunkt Stufe 01

rHys01: Absolutwert Hysterese Stufe 01

udiDlyOn01: Einschaltverzögerung Stufe 01

udiDlyOff01: Ausschaltverzögerung Stufe 01

rSwiOn02: Einschaltpunkt Stufe 02

rHys02: Absolutwert Hysterese Stufe 02

udiDlyOn02: Einschaltverzögerung Stufe 02

udiDlyOff02: Ausschaltverzögerung Stufe 02

rSwiOn03: Einschaltpunkt Stufe 03

rHys03: Absolutwert Hysterese Stufe 03

udiDlyOn03: Einschaltverzögerung Stufe 03

udiDlyOff03: Ausschaltverzögerung Stufe 03

rSwiOn04: Einschaltpunkt Stufe 04

rHys04: Absolutwert Hysterese Stufe 04

udiDlyOn04: Einschaltverzögerung Stufe 04

udiDlyOff04: Ausschaltverzögerung Stufe 04

udiNumOfStp: Eingabe der Anzahl der Stufen, die benötigt werden.
Die Eingabe ist begrenzt von 0 bis 4

bActn: Eingangsvariable mit der der Wirksinn des Stufenschalters bestimmt wird.
TRUE = Direct = Kühlen; FALSE = Reverse = Heizen

VAR_OUTPUT

```
bQ01      : BOOL;
bQ02      : BOOL;
bQ03      : BOOL;
bQ04      : BOOL;
udiStp    : UDINT;
rSwiOn    : REAL;
rSwiOff   : REAL;
udiRemTiDlyOn : UDINT;
udiRemTiDlyOff : UDINT;
```

bQ01: Anzeige des Status Stufe 01
TRUE = EIN; FALSE = AUS
udiStp >= 1

bQ02: Anzeige des Status Stufe 02
TRUE = EIN; FALSE = AUS
udiStp >= 2

bQ03: Anzeige des Status Stufe 03
TRUE = EIN; FALSE = AUS
udiStp >= 3

bQ04: Anzeige des Status Stufe 04
TRUE = EIN; FALSE = AUS
udiStp >= 4

udiStp: Anzeige in welcher Stufe sich der Stufenschalter befindet

rSwiOn: Anzeige des nächsten Einschaltpunktes

rSwiOff: Anzeige des nächsten Ausschaltpunktes

udiRemTiDlyOn: Ist der Einschaltpunkt für das Weiterschalten auf die nächste Stufe erfüllt, dann wird hier der Ablauf der Einschaltverzögerungszeit angezeigt.

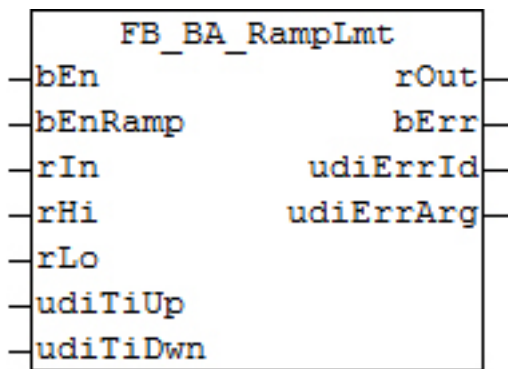
udiRemTiDlyOff: Ist der Ausschaltpunkt für das Herunterschalten auf die nächste Stufe erfüllt, dann wird hier der Ablauf der Ausschaltverzögerungszeit angezeigt.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

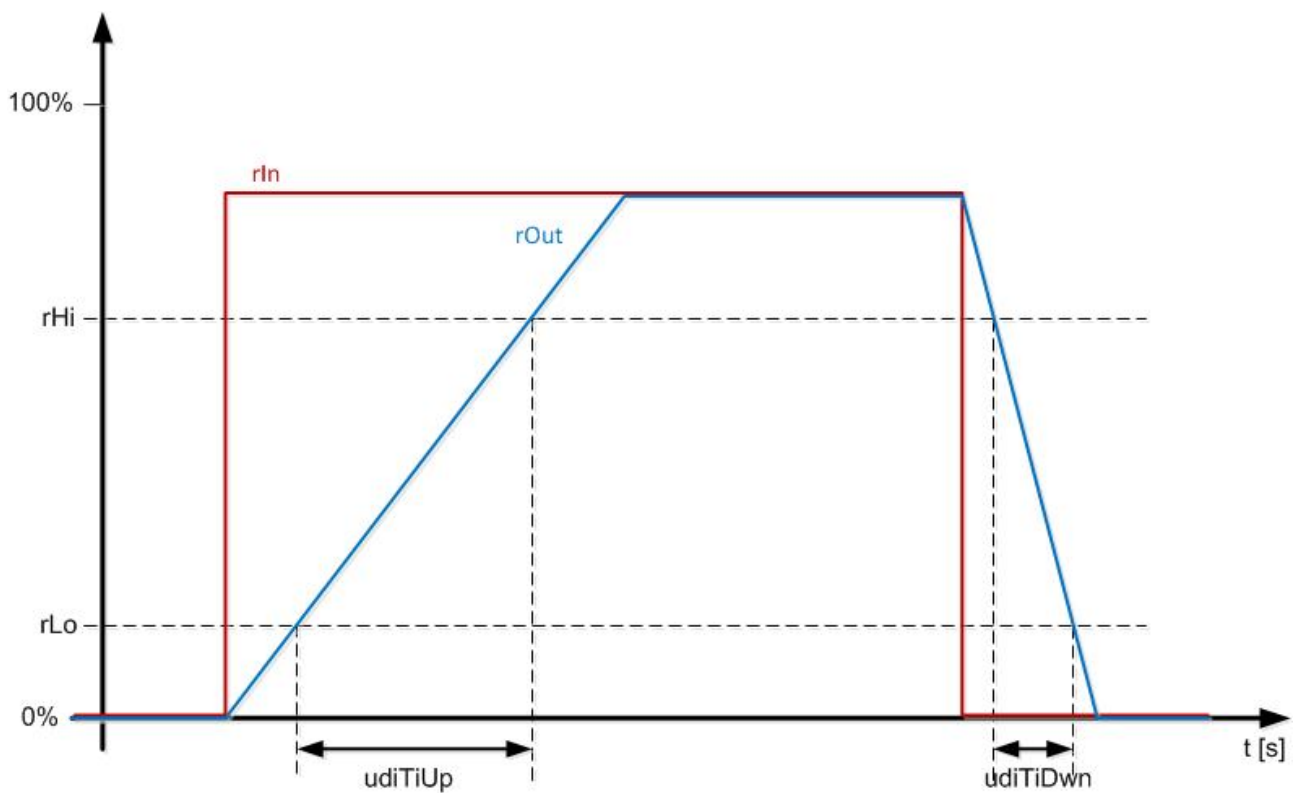
8.2.49 FB_BA_RampLmt

Rampenbegrenzung



Funktionsbeschreibung

Der Funktionsbaustein limitiert die Anstiegs- bzw. Abfallgeschwindigkeit eines Eingangssignals. Beim Anstieg von rIn wird der Ausgang $rOut$ auf die Steigung von $(rHi-rLo)/udiTiUp$ begrenzt. Beim Abfallen von rIn wird der Ausgang $rOut$ auf die Steigung von $(rHi-rLo)/udiTiDwn$ begrenzt.



Ein-/Ausgänge

VAR_INPUT

```

bEn      : BOOL;
bEnRamp  : BOOL;
rIn      : REAL;
rHi      : REAL;
rLo      : REAL;
udiTiUp  : UDINT;
udiTiDwn : UDINT;

```

bEn: Freigabe Baustein, wenn FALSE, dann ist $rOut = 0.0$

bEnRamp: Freigabe Rampenbegrenzung, wenn FALSE, dann ist $rOut = rIn$.

rIn: Eingangswert der Rampenfunktion

rHi: Obere Stützstelle zur Berechnung der Rampen.

rLo: untere Stützstelle zur Berechnung der Rampen. *rHi* muss größer als *rLo* sein, ansonsten wird ein Fehler ausgegeben!

udiTiUp: Anstiegszeit [s]

udiTiDwn: Abfallzeit [s]

VAR_OUTPUT

```
rOut      : REAL;
bErr      : BOOL;
udiErrId  : UDINT;
udiErrArg : UDINT;
```

rOut: durch die Rampen steigungs-begrenztes Ausgangssignal

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

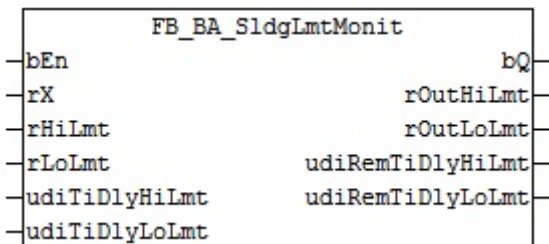
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.50 FB_BA_SldgLmtMonit

Funktionsbaustein für eine gleitende Grenzwertüberwachung



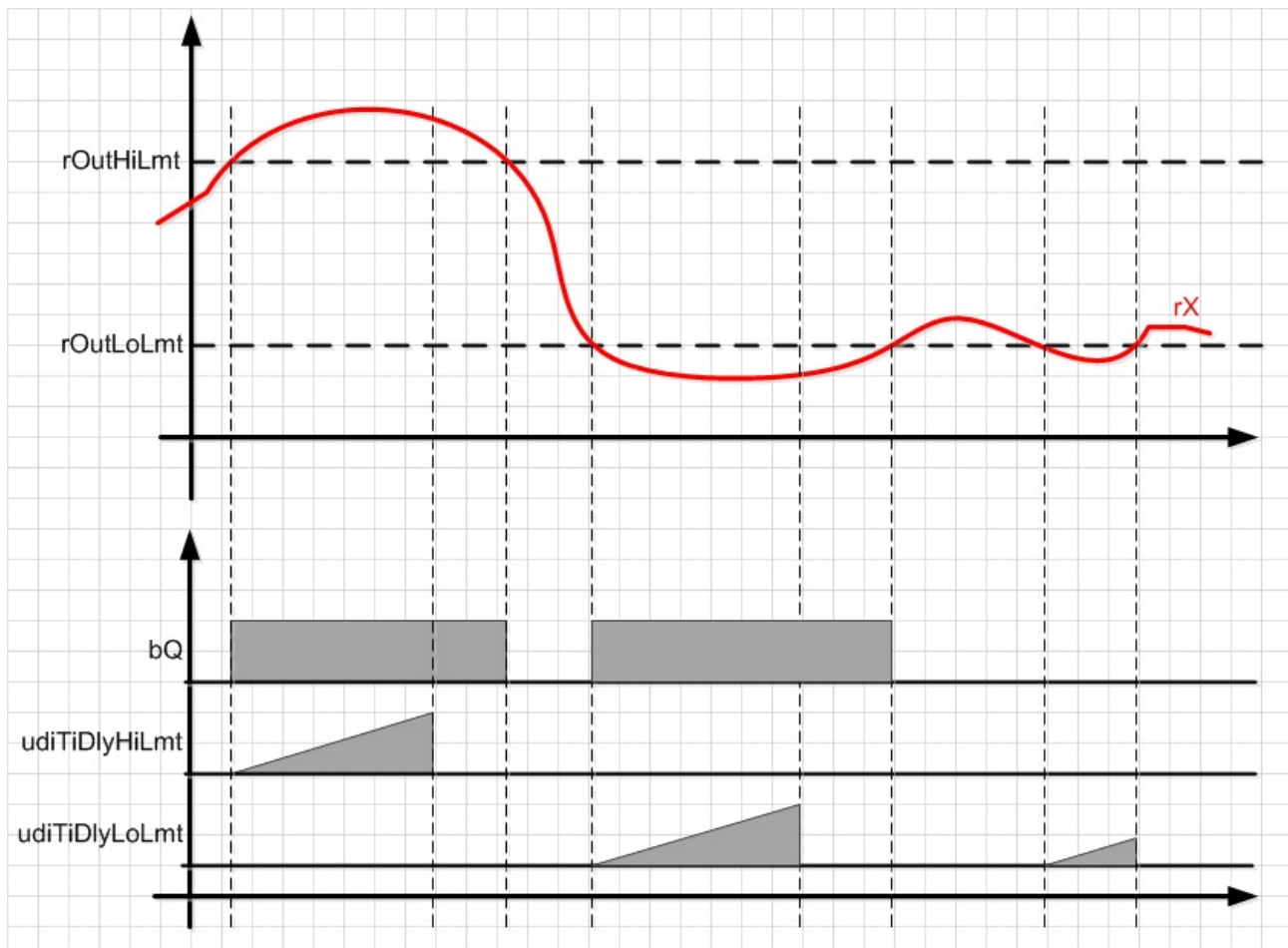
Funktionsbeschreibung

Der Funktionsbaustein kann für eine gleitende Grenzwertüberwachung eingesetzt werden. Dieses kann eine Funktionsüberwachung eines Regelventils durch Vergleich des Stellausgangs und der Stellungsrückmeldung sein.

Wichtig ist, dass nicht größer gleich sein darf. Ist dieses der Fall, dann ist = Lmt und = - 0,05. rHiLmt rLoLmt rOutHiLmt rHi rOutLoLmt rHiLmt

Diagramm 1

Diagramm 1



Ein-/Ausgänge

VAR_INPUT

```

bEn      : BOOL;
rX       : REAL;
rHiLmt  : REAL;
rLoLmt  : REAL;
udiTiDlyHiLmt : UDINT; [s]
udiTiDlyLoLmt : UDINT; [s]

```

bEn: Freigabe Baustein

rX: Istwert

rHiLmt: Oberer Grenzwert. $rOutHiLmt = rHiLmt$

rLoLmt: Unterer Grenzwert. $rOutLoLmt = rLoLmt$

udiTiDlyHiLmt : Ist rX größer $rOutHiLmt$ und die Zeitverzögerung $udiTiDlyHiLmt$ abgelaufen, so wird bQ gesetzt, siehe [Diagramm 1 \[► 145\]](#).

udiTiDlyLoLmt : Ist rX kleiner $rOutLoLmt$ und die Zeitverzögerung $udiTiDlyLoLmt$ abgelaufen, so wird bQ gesetzt, siehe [Diagramm 1 \[► 145\]](#).

VAR_OUTPUT

```

bQ       : BOOL;
rOutHiLmt : REAL;
rOutLoLmt : REAL;
udiRemTiDlyHiLmt : UDINT; [s]
udiRemTiDlyLoLmt : UDINT; [s]

```

bQ: Grenzwertüberwachung ist frei gegeben. bQ kann nur über $bEn = FALSE$ zurück gesetzt werden.

rOutHiLmt: Ausgabe oberer Grenzwert

rOutLoLmt : Ausgabe unterer Grenzwert

udiRemTiDlyHiLmt : Countdown der Verzögerung *udiTiDlyHiLmt*

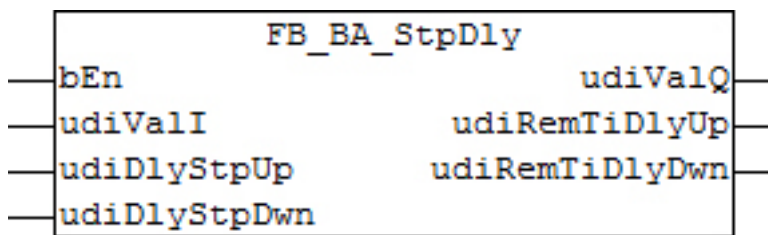
udiRemTiDlyLoLmt : Countdown der Verzögerung *udiTiDlyLoLmt*

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.51 FB_BA_StpDly

Dieser Baustein dient zur verzögerten Ausgabe von Schaltstufen.



Funktionsbeschreibung

Die Funktion wird über den Eingang *bEn* aktiviert. Ist dies der Fall, die Zahl die am Eingang *udiValI* anliegt verzögert am den Ausgang *udiValQ* ausgegeben. Ist die Zahl am Eingang größer als die am Ausgang, so verzögert sich die erneute Ausgabe um *udiDlyStpUp* [s], ist sie kleiner, dann um *udiDlyStpDwn*.

i Ändert sich während des Timerablaufs nach oben die Zahl am Ausgang auch weiter nach oben, so wird der Zeitgeber **NICHT** erneut gestartet. Das Gleiche gilt für den Ablauf nach unten. Der jeweilige Schaltzyklus gilt als abgeschlossen, wenn der dazugehörige Countdown abgelaufen ist oder wenn während des Ablaufes am Eingang (wieder) die gleiche Zahl wie am Ausgang anliegt.

Steht *bEn* auf FALSE, so wird der Eingangswert *udiValI* unverzögert an *udiValQ* ausgegeben.

Ein-/Ausgänge

VAR_INPUT

```
bEn          : BOOL;
udiValI     : UDINT;
udiDlyStpUp : UDINT;
udiDlyStpDwn : UDINT;
```

bEn: Ein TRUE-Signal an diesem Eingang aktiviert den Baustein. Der Wert am Eingang *udiValI* wird nur noch verzögert an *udiValQ* ausgegeben. Steht *bEn* auf FALSE, so wird der Eingangswert *udiValI* unverzögert ausgegeben.

udiValI: Eingangswert

udiDlyStpUp: Verzögerungszeit zum Hochschalten [s]

udiDlyStpDwn: Verzögerungszeit zum Herunterschalten [s]

VAR_OUTPUT

```
udiValQ      : UDINT;
udiRemTiDlyUp : UDINT;
udiRemTiDlyDwn : UDINT;
```

udiValQ: Ausgangswert.

udiRemTiDlyUp: Countdown zum Hochschalten [s]

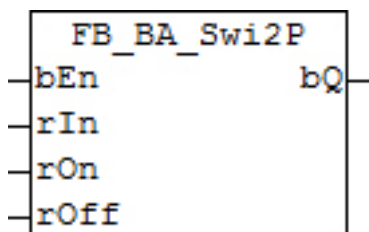
udiRemTiDlyDwn: Countdown zum Herunterschalten [s]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.52 FB_BA_Swi2P

Zweipunktschalter

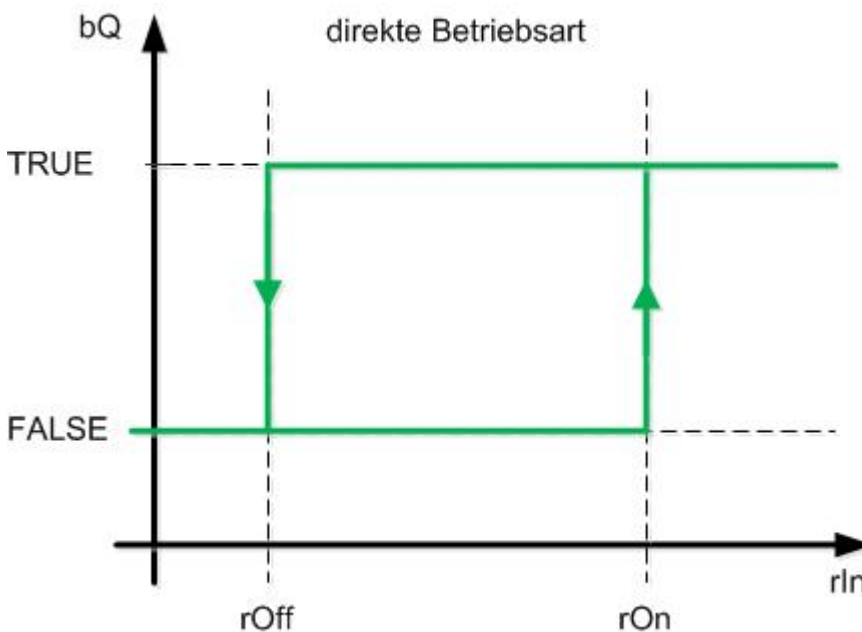


Funktionsbeschreibung

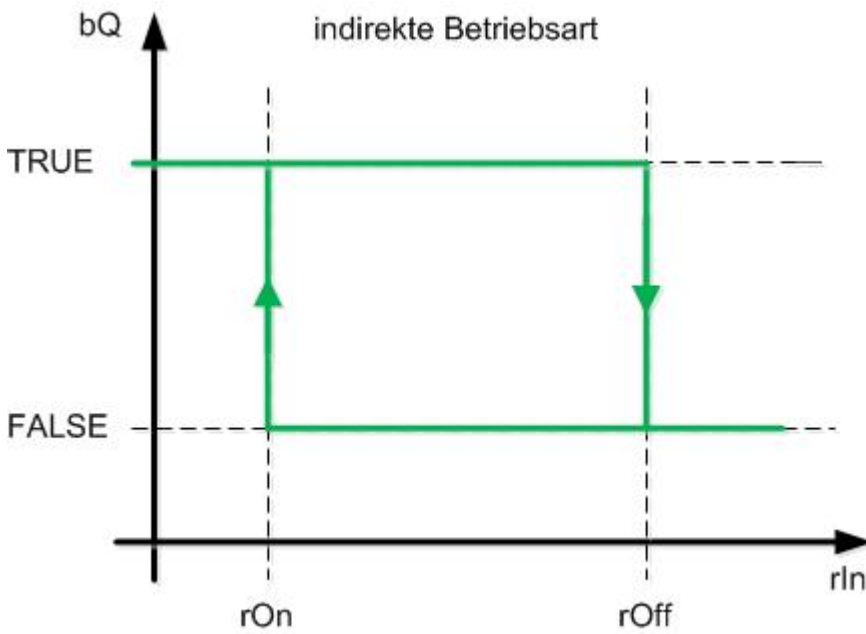
Der Funktionsbaustein FB_BA_Swi2P ist ein Zweipunktschalter mit einem Ein- und einem Ausschaltpunkt.

Eine generelle Freigabe des Funktionsbausteins erfolgt am Eingang *bEn*. Solange *bEn* FALSE ist ist der Ausgang *bQ* FALSE. Der Wirksinn des Bausteins hängt von der relativen Lage des Einschaltpunktes und des Ausschaltpunktes ab.

Ist der Einschaltpunkt größer als der Ausschaltpunkt dann ist der Wirksinn direkt bzw. gleichläufig (Kühlbetrieb).



Ist der Ausschaltpunkt größer als der Einschaltpunkt dann ist der Wirksinn direkt bzw. gegenläufig (Heizbetrieb).



Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
rIn      : REAL;
rOn      : REAL;
rOff     : REAL;
```

bEn: Generelle Freigabe des Funktionsbausteins

rIn: Eingangswert

rOn: Einschaltpunkt

rOff: Ausschaltpunkt

VAR_OUTPUT

```
bQ       : BOOL;
```

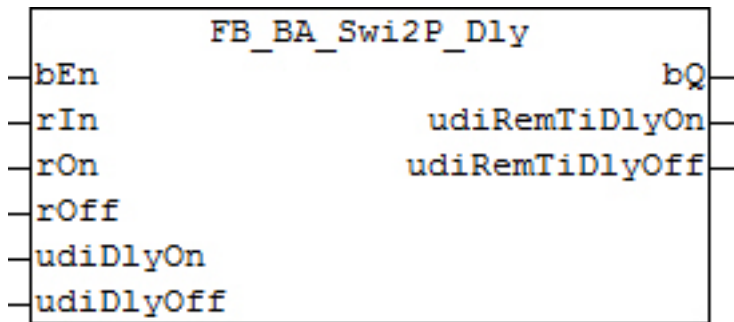
bQ: Steuerausgang

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.53 FB_BA_Swi2P_Dly

Zweipunktschalter mit zeitlicher Verzögerung

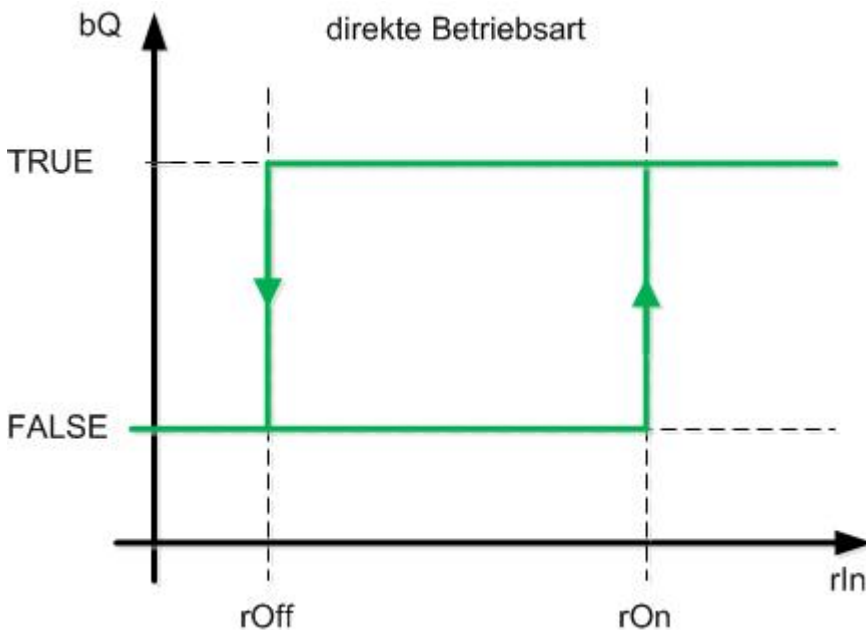


Funktionsbeschreibung

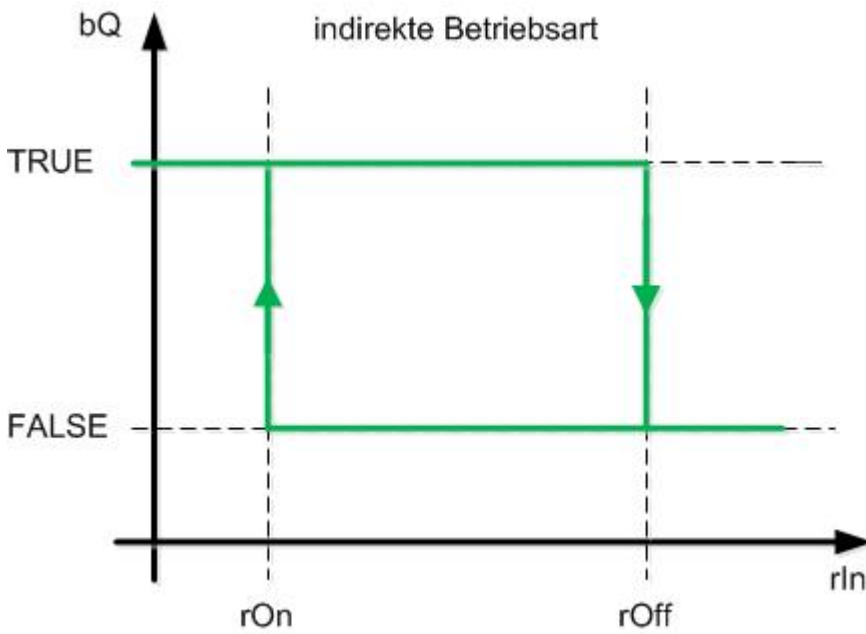
Der Funktionsbaustein FB_BA_Swi2P_Dly ist ein Zweipunktschalter mit einem Ein- und einem Ausschaltzeitpunkt. Im Gegensatz zum [FB_BA_Swi2P \[► 148\]](#) wird hier eine Änderung des Ausgangssignals zeitlich verzögert.

Eine generelle Freigabe des Funktionsbausteins erfolgt am Eingang *bEn*. Solange *bEn* FALSE ist ist der Ausgang *bQ* FALSE. Der Wirksinn des Bausteins hängt von der relativen Lage des Einschaltzeitpunktes und des Ausschaltzeitpunktes ab.

Ist der Einschaltzeitpunkt größer als der Ausschaltzeitpunkt dann ist der Wirksinn direkt bzw. gleichläufig (Kühlbetrieb).



Ist der Ausschaltzeitpunkt größer als der Einschaltzeitpunkt dann ist der Wirksinn direkt bzw. gegenläufig (Heizbetrieb).



Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
rIn      : REAL;
rOn      : REAL;
rOff     : REAL;
udiDlyOn : UDINT;
udiDlyOff : UDINT;
```

bEn: generelle Freigabe des Funktionsbausteins

rIn: Eingangswert

rOn: Einschaltpunkt

rOff: Ausschaltpunkt

udiDlyOn: Einschaltverzögerung [s]

udiDlyOff: Ausschaltverzögerung [s]

VAR_OUTPUT

```
bQ      : BOOL;
udiRemTiDlyOn : UDINT;
udiRemTiDlyOff : UDINT;
```

bQ: Steuerausgang

udiRemTiDlyOn: Countdown Einschaltverzögerung [s]

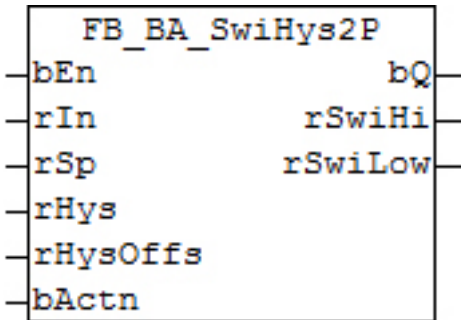
udiRemTiDlyOff: Countdown Ausschaltverzögerung [s]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.54 FB_BA_SwiHys2P

Zweipunktschalter um einen Schaltpunkt



Funktionsbeschreibung

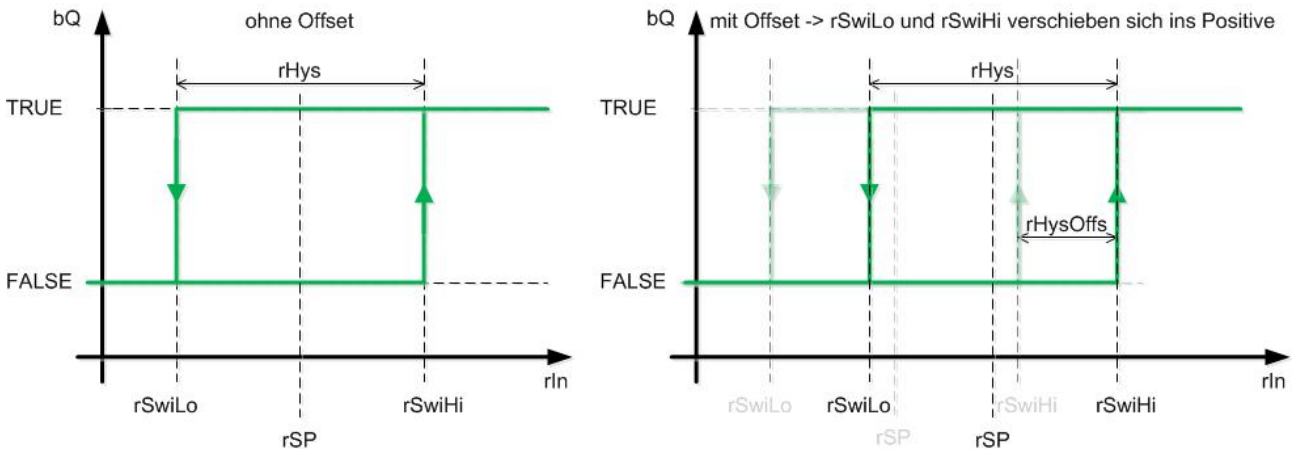
Der Funktionsbaustein FB_BA_SwiHys2P ist ein Zweipunktschalter mit einstellbarer Hysterese und Hystereseoffset.

Eine generelle Freigabe des Funktionsbausteins erfolgt am Eingang *bEn*. Bei gesperrten Funktionsbaustein ist der Ausgang *bQ* FALSE. Am Eingang *rSp* wird der Sollwert für den Zweipunktschalter angeschlossen. Der Wirksinn des Bausteins hängt von der Eingangsvariablen *bActn* ab.

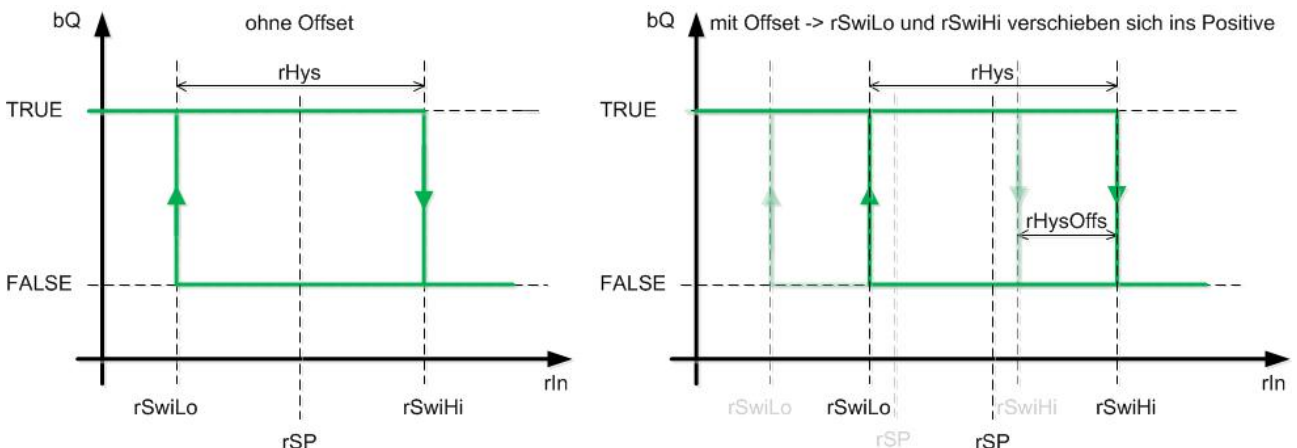
Die aktiven Schaltpunkte ergeben sich aus dem Sollwert, der Hysterese und dem Offset für die Hysterese. Diese werden an den Ausgängen *rSwiHi* und *rSwiLo* ausgegeben.

- Der obere Schaltpunkt ergibt sich aus $rSp + rHys/2 + rHysOffs$.
- Der untere Schaltpunkt ergibt sich aus $rSp - rHys/2 + rHysOffs$.

Ist *bActn* TRUE ergibt sich ein direkter bzw. gleichläufiger Wirksinn (Kühlbetrieb).



Ist *bActn* FALSE ergibt sich ein indirekter bzw. gegenläufiger Wirksinn (Heizbetrieb).



Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
rIn      : REAL;
rSp      : REAL;
rHys     : REAL;
rHysOffs : REAL;
bActn    : BOOL;
```

bEn: generelle Freigabe des Funktionsbausteins

rIn: Eingangswert

rSp: Sollwerteingang

rHys: Hysterese

rHysOffs: Offset für die Hysterese

bActn: Wirksinn

VAR_OUTPUT

```
bQ       : BOOL;
rSwiHi   : REAL;
rSwiLo   : REAL;
```

bQ: Ausgang

rSwiHi: Oberer Schalterpunkt

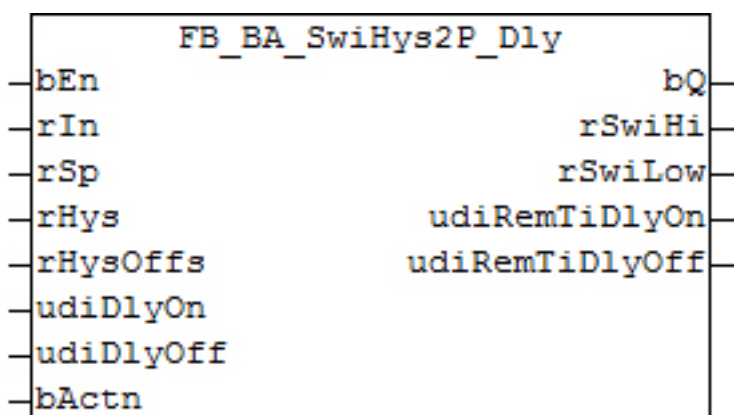
rSwiLo: Unterer Schalterpunkt

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.55 FB_BA_SwiHys2P_Dly

Zweipunktschalter um einen Schalterpunkt mit zeitlicher Verzögerung



Funktionsbeschreibung

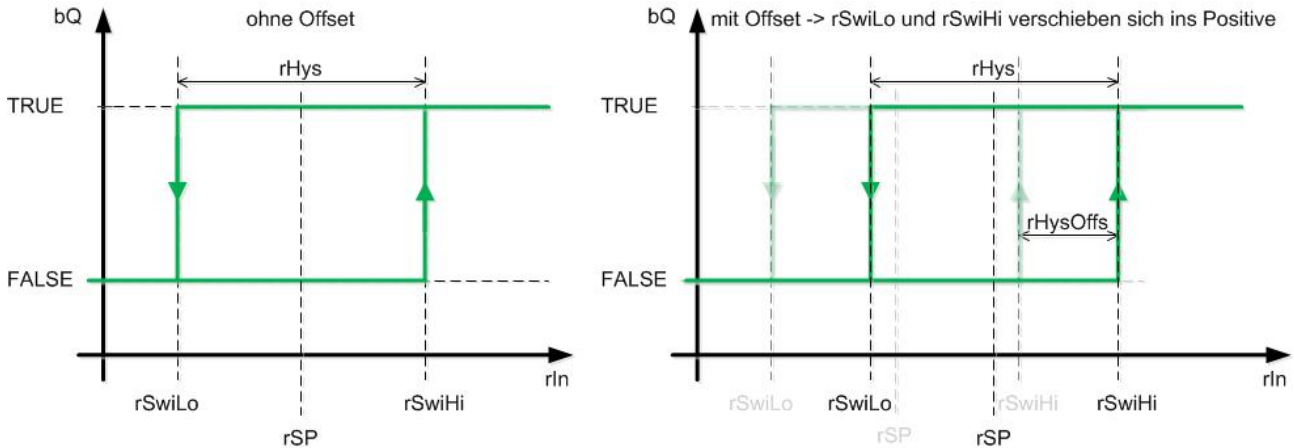
Der Funktionsbaustein FB_BA_SwiHys2P_Dly ist ein Zweipunktschalter mit Hysterese. Im Gegensatz zum FB_BA_SwiHys2P [► 152] wird hier eine Änderung des Ausgangssignals zeitlich verzögert.

Eine generelle Freigabe des Funktionsbausteins erfolgt am Eingang bEn . Bei gesperrten Funktionsbaustein ist der Ausgang bQ FALSE. Am Eingang rSp wird der Sollwert für den Zweipunktschalter angeschlossen. Der Wirksinn des Bausteins hängt von der Eingangsvariablen $bActn$ ab.

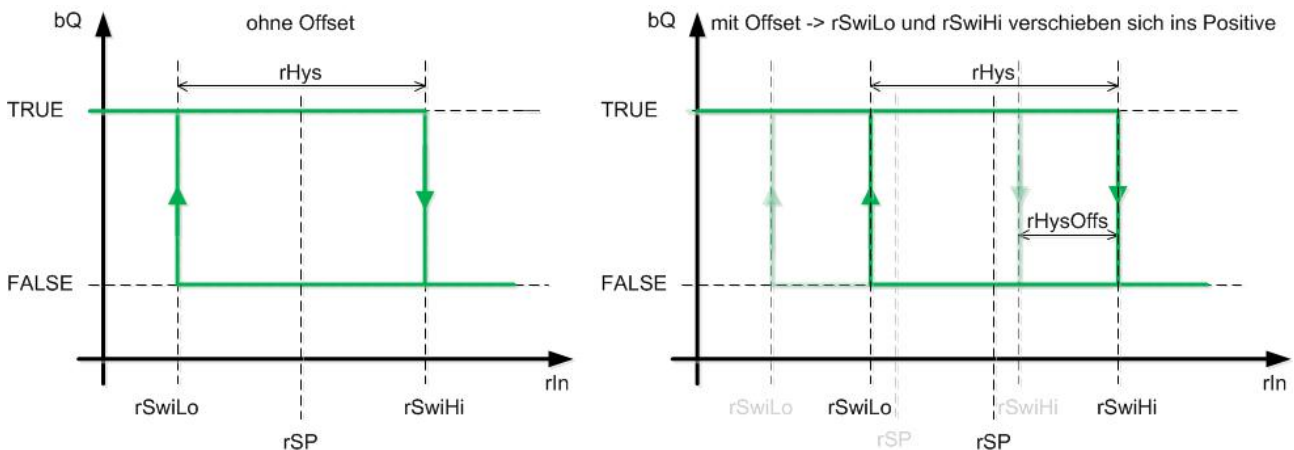
Die aktiven Schaltpunkte ergeben sich aus dem Sollwert, der Hysterese und dem Offset für die Hysterese. Diese werden an den Ausgängen $rSwiHi$ und $rSwiLo$ ausgegeben.

- Der obere Schaltpunkt ergibt sich aus $rSp + rHys/2 + rHysOffs$.
- Der untere Schaltpunkt ergibt sich aus $rSp - rHys/2 + rHysOffs$.

Ist $bActn$ TRUE ergibt sich ein direkter bzw. gleichläufiger Wirksinn (Kühlbetrieb).



Ist $bActn$ FALSE ergibt sich ein indirekter bzw. gegenläufiger Wirksinn (Heizbetrieb).



Ein-/Ausgänge

VAR_INPUT

```

bEn      : BOOL;
rIn      : REAL;
rSp      : REAL;
rHys     : REAL;
rHysOffs : REAL;
udiDlyOn : UDINT;
udiDlyOff : UDINT;
bActn    : BOOL;

```

bEn: Generelle Freigabe des Funktionsbausteins

rIn: Eingangswert

rSp: Sollwerteingang

rHys: Hysterese

rHysOffs: Offset für die Hysterese

udiDlyOn: Einschaltverzögerung [s]

udiDlyOff: Ausschaltverzögerung [s]

bActn: Wirksinn

VAR_OUTPUT

```
bQ      : BOOL;
rSwiHi  : REAL;
rSwiLo  : REAL;
```

bQ: Ausgang

rSwiHi: Oberer Schalterpunkt

rSwiLo: Unterer Schalterpunkt

udiRemTiDlyOn: Countdown Einschaltverzögerung [s]

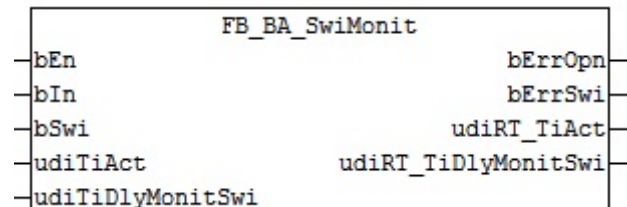
udiRemTiDlyOff: Countdown Ausschaltverzögerung [s]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.56 FB_BA_SwiMonit

Dieser Baustein dient zur Überwachung von Endlagenschaltern, z. B. bei einer Zweipunktklappe



Funktionsbeschreibung

Der Funktionsbaustein besteht aus 2 Schritten um einen Endlagenschalter zu überwachen.

Schritt 1 beinhaltet die Überwachung während der Fahrzeit des Aktors.

Schritt 2 überwacht den geöffneten Zustand des Aktors.

Die Funktion wird über den Eingang *bEn* aktiviert. Ist *bEn* FALSE, so sind die Ausgänge *bErrOpn*/*bErrSwi* ebenfalls FALSE.

Schritt 1: Steht am Eingang *bIn* steht ein TRUE an, dann läuft die Verfahrzeit des Aktors *udiTiAct*/*udiRT_TiAct* ab. Steht innerhalb dieser Zeit am Eingang *bSwi* kein TRUE an, so wird über den Ausgang *bErrOpn* eine Störung angezeigt. Wird *bSwi* aber innerhalb der Zeitangabe *udiTiAct* TRUE, so ist Schritt 1 abgearbeitet und Schritt 2 wird aktiv.

Schritt 2: *bIn* und *bSwi* sind TRUE. Wird *bIn* FALSE, so wird Schritt 1 wieder aktiv. Wird *bSwi* FALSE und innerhalb der Zeitangabe *udiTiDlyMonitSwi*/*udiRT_TiDlyMonitSwi* nicht wieder TRUE, so wird über den Ausgang *bErrSwi* eine Störung angezeigt.

Die Störungen *bErrOpn*/*bErrSwi* werden zurück gesetzt, wenn entweder *bEn* oder *bIn* FALSE sind.

Ein-/Ausgänge**VAR_INPUT**

```

bEn          : BOOL;
bIn          : BOOL;
bSwi        : BOOL;
udiTiAct    : UDINT;
udiTiDlyMonitSwi : UDINT;

```

bEn: Ein TRUE-Signal an diesem Eingang aktiviert den Baustein. Der Wert am Eingang *udiValI* wird nur noch verzögert an *udiValQ* ausgegeben. Steht *bEn* auf FALSE, so wird der Eingangswert *udiValI* unverzögert ausgegeben.

bIn: Die Ansteuerung des Aktors muss an diesem Eingang angelegt werden.

bSwi: Endlagenschalter Aktor

udiTiAct: Verfahrzeit des Aktors [s]

udiTiDlyMonitSwi: Verzögerungszeit des Endlagenschalters im geöffneten Zustand [s]. Die Zeit wird aktiv, wenn *bIn* TRUE ist und der Endlagenschalter *bSwi* seinen Zustand von TRUE nach FALSE ändert.

VAR_OUTPUT

```

bErrOpn     : BOOL;
bErrSwi     : BOOL;
udiRT_TiAct : UDINT;
udiRT_TiDlyMonitSwi : UDINT;

```

bErrOpn: Störung Schritt 1 Endlagenschalter hat während des Verfahrens nicht die Endlage erreicht

bErrSwi: Störung Schritt 2 Endlagenschalter bzw. der Aktor hat im geöffneten Zustand seine Endlage verlassen

udiRT_TiAct: Countdown um den Ausgang *bErrOpn* zu setzen [s]. Die Vorgabe kommt von *udiTiAct*.

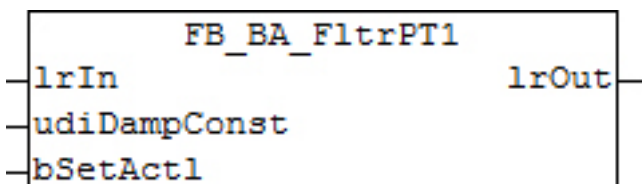
udiRT_TiDlyMonitSwi: Countdown zum um den Ausgang *bErrSwi* zu setzen [s]. Die Vorgabe kommt von *udiTiDlyMonitSwi*.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.57 FB_BA_FltrPT1

Filter erster Ordnung

**Ein-/Ausgänge****VAR_INPUT**

```

lrIn        : LREAL;
udiDampConst : UDINT;
bSetAct1    : BOOL;

```

Irln: Eingangssignal

udiDampConst: Filterzeitkonstante [s]

bSetActl: Eine steigende Flanke an diesem Eingang setzt den Ausgangswert *IrOut* auf den Eingangswert *Irln*.

VAR_OUTPUT

IrOut : LREAL;

IrOut: Gefiltertes Ausgangssignal



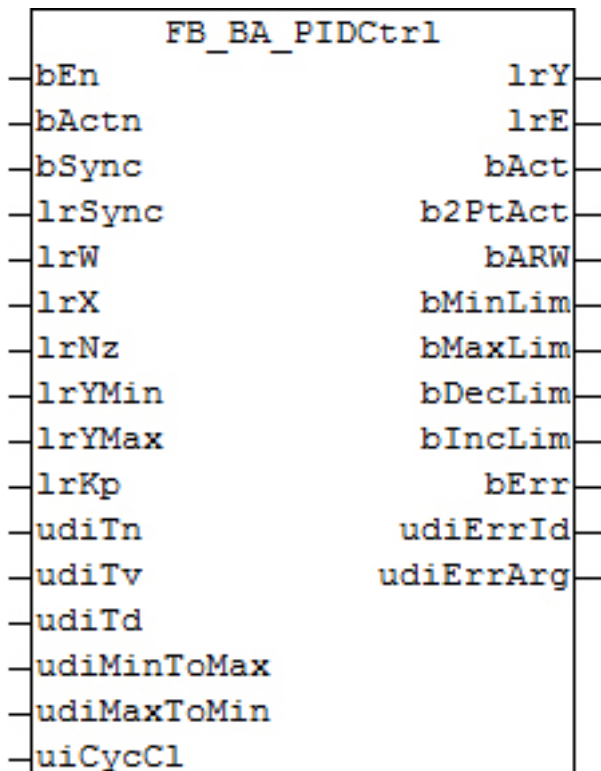
Beim ersten Aufruf des Bausteins (Systemstart) wird der Ausgang *IrOut* einmalig automatisch gleich dem Eingang *Irln* gesetzt.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.58 FB_BA_PIDCtrl

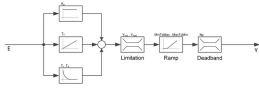
Universeller PID-Regler



Funktionsbeschreibung

Dieser Regler ist als Standard-PID-Regler in Parallelstruktur ausgelegt. Für einen Regler mit wählbaren P-Glied (vorgelagert oder parallel zu I- und D) siehe bitte [FB_BA_PIDCtrlEx \[► 161\]](#).

Wirkungsplan



Passiv-Verhalten ($bEn = FALSE$ oder $bErr = TRUE$)

Die Ausgänge werden wie folgt gesetzt:

IrY	0,0
IrE	0,0
bAct	FALSE
bARW	FALSE
bMinLim	FALSE
bMaxLim	FALSE
bIncLim	FALSE
bDeclim	FALSE

Im Fehlerfall steht $bErr$ auf TRUE - $udiErrId$ und $udiErrArg$ beschreiben den Fehler. Die internen Werte für P-, I-, und D-Anteil werden auf 0 gesetzt, ebenso die Werte für den I- und D-Anteil vom vorhergehenden Zyklus. Damit wird die Stellgröße bei einem Neustart im ersten Zyklus "sauber", das heißt ohne Vergangenheitswerte berechnet.

Aktiv-Verhalten ($bEn = TRUE$ und $bErr = FALSE$)

Im ersten Zyklus werden I- und D-Anteil wie bereits erwähnt ohne Vergangenheitswerte berechnet und somit "sauber" aufgestartet. Ein positives Signal an $bSync$ setzt den I-Anteil so, dass die Stellgröße den Wert $IrSync$ annimmt. Diese Methode kann, wenn bEn und $bSync$ gleichzeitig gesetzt werden, zum Setzen eines Initialwerts genutzt werden, von dem aus die Regelung "losläuft". Ist der I-Anteil nicht aktiv, so wird der D-Anteil entsprechend gesetzt. Zu beachten ist, dass intern nur die steigende Flanke von $bSync$ ausgewertet wird, handelt es sich doch um eine Setz-Aktion. Für ein erneutes Synchronisieren, etwa auf einen Übergabewert, muss am Eingang $bSync$ ein erneutes TRUE-Signal angelegt werden. Ist der I-Anteil aktiv so sorgt der Regler dafür, dass dieser fest gehalten wird, sollte sich der Reglerausgang IrY an den Grenzen $IrYMin$ oder $IrYMax$ befinden und noch weiter fallen bzw. steigen wollen. Dieses Verfahren wird Anti-Reset-Wind-Up genannt und trägt dafür Sorge, dass der I-Anteil immer nur so groß ist, dass die Stellgröße bei entsprechender Regelabweichung sofort wieder Werte innerhalb der Grenzen annehmen kann, ohne dass ein zu groß gewordener Integralanteil zunächst noch abgebaut werden muss.

Wirksinn

Wirksinn

Mit $bActn = FALSE$ wird der Wirksinn des Reglers so umgekehrt, dass eine Regelabweichung kleiner als 0 eine Stellgrößenänderung ins Positive bewirkt. Dies wird dadurch erreicht, dass die Regelabweichung negativ berechnet wird:

bActn	rXW (Regelabweichung)	Wirksinn
TRUE	$IrX - IrW$ (Istwert-Sollwert)	direkt (Kühlen)
FALSE	$IrW - IrX$ (Sollwert-Istwert)	indirekt (Heizen)

Anti-Reset-Windup bei Erreichen des Maximal- bzw Minimalwerts

Erreicht der Regler am Ausgang seine Obergrenze und ist die Regelabweichung weiterhin positiv, so wird der Integralanteil sich so lange weiter erhöhen, bis die Regelabweichung wieder kleiner oder gleich Null ist. Dies kann dazu führen, dass sich unnötigerweise ein sehr hoher Integralanteil aufbaut, der bei einer Vorzeichenänderung der Regelabweichung erst wieder abgebaut werden muss und das Regelverhalten träge macht. Das gleiche gilt auch bei Erreichen des Minimalwerts am Ausgang bei einer weiterhin negativen Regelabweichung. Um diesem vorzubeugen wird, sobald einer der beiden Fälle eintritt, der I-Anteil nicht weiter berechnet, sondern auf dem Wert des vorhergehenden SPS-Zyklus gehalten. Tritt eine Vorzeichenumkehr ein, also Maximalwert erreicht und Regelabweichung kleiner als 0.0 bzw. Minimalwert erreicht und Regelabweichung größer als 0.0, so wird der I-Anteil wieder berechnet.

Steigungsbegrenzung

Ist der Regler schneller eingestellt, als das Stellglied, so kann dieses dem Regler nicht folgen, was zu einem Schwingverhalten führen kann. Daher gibt es die Möglichkeit die Steigung der Stellgröße zu begrenzen.

Als Maß dafür gelten die Zeiten:

udiMinToMax: Steigungsbegrenzung des Reglerausgangs für Anstieg: *udiMinToMax* [s] bezogen auf eine Änderung von *lrYMin* auf *lrYMax*.

udiMaxToMin: Steigungsbegrenzung des Reglerausgangs für Abfall: *udiMaxToMin* [s] bezogen auf eine Änderung von *lrYMax* auf *lrYMin*.

Hieraus lässt sich jeweils die maximale Änderung pro SPS-Zyklus - das maximale Inkrement oder Dekrement - errechnen. Ist die berechnete Änderung des Stellsignals über einen SPS-Zyklus nun höher als die durch *udiMinToMax* bzw. *udiMaxToMin* eingestellte, so wird das Stellsignal lediglich um das maximale Inkrement oder Dekrement erhöht bzw. erniedrigt. Der I-Anteil wird dabei intern auf dieselbe Art nachgeführt (I-Anteil des letzten SPS-Zyklus + maximales Inkrement bzw. I-Anteil des letzten SPS-Zyklus - maximales Dekrement).

Neutrale Zone (Deadband)

Ein Wert von *lrNZ* > 0.0 gibt die Funktion der neutralen Zone (Deadband) frei. Ein Wert gleich Null deaktiviert das Totbandglied und die Werte am Eingang werden direkt durchgereicht.

Wenn das Totbandglied aktiv ist wird der Eingangswert erst dann als neuer Ausgangswert durchgereicht, wenn der Absolutwert der Differenz vom Eingangswert zum jetzigen Ausgangswert größer oder gleich *lrNZ/2* ist.

Beispiel: *lrNZ* = 1, *lrYin* = 55.0, *lrYout*= 55.0 (Wert wurde gerade durchgereicht)

PLC-Zyklus+1	<i>lrYin</i> =55.2	<i>lrYout</i> =55.0
PLC-Zyklus+2	<i>lrYin</i> =55.3	<i>lrYout</i> =55.0
PLC-Zyklus+3	<i>lrYin</i> =55.1	<i>lrYout</i> =55.0
PLC-Zyklus+4	<i>lrYin</i> =55.6	<i>lrYout</i> =55.6
PLC-Zyklus+5	<i>lrYin</i> =55.4	<i>lrYout</i> =55.6
PLC-Zyklus+6	<i>lrYin</i> =55.3	<i>lrYout</i> =55.6
PLC-Zyklus+7	<i>lrYin</i> =55.1	<i>lrYout</i> =55.1

Durch diese Funktion sollen unnötig viele Stellimpulse vermieden werden.

Zweipunkt-Regelverhalten

- nicht mehr vorhanden -

Autokorrektur von Parametern

Bei einigen Parametern macht es Sinn, sie automatisch zu korrigieren bzw. zu begrenzen, ohne eine Fehlermeldung auszugeben, da die Erwartung des Anwenders eindeutig vorhersehbar ist:

- *lrSync* > *lrYMax* -> *lrSync* := *lrYMax*
- *lrSync* < *lrMin* -> *lrSync* := *lrMin*
- *uiCycCl* = 0 -> *uiCycCl* := 1

Ein-Ausgänge

VAR_INPUT

```

bEn      : BOOL;
bActn    : BOOL;
bSync    : BOOL;
lrSync   : LREAL;
lrW      : LREAL;
lrX      : LREAL;
lrNZ     : LREAL;
lrYMin   : LREAL;
    
```

```

lrYMax      : LREAL;
lrKp        : LREAL;
udiTn       : UDINT;
udiTv       : UDINT;
udiTd       : UDINT;
udiMinToMax : UDINT;
udiMaxToMin : UDINT;
uiCycCl     : UINT;

```

bEn: Regleraktivierung

bActn: Wirksinn [► 158] des Reglers

bSync / IrSync: Synchronisationsbefehl: Ausgangswert *lrY* auf *IrSync* setzen

lrW: Sollwert

lrX: Istwert

lrNZ: Neutrale Zone

lrYMin: Untere Ausgabebegrenzung des Reglers

lrYMax: Obere Ausgabebegrenzung des Reglers

lrKp: Reglerverstärkung. Wirkt nur auf den P-Anteil

udiTn: Nachstellzeit des I-Anteiles [ms] . Ein Nullwert an diesem Parameter schaltet den I-Anteil ab.

udiTv: Vorhaltezeit des D-Anteiles [ms] . Ein Nullwert an diesem Parameter schaltet den D-Anteil ab.

udiTd: Dämpfungszeit des D-Anteiles [ms]

udiMinToMax: Steigungsbegrenzung des Reglerausgangs für Anstieg: *udiMinToMax* [s] bezogen auf eine Änderung von *lrYMin* auf *lrYMax*.

udiMaxToMin: Steigungsbegrenzung des Reglerausgangs für Abfall: *udiMaxToMin* [s] bezogen auf eine Änderung von *lrYMax* auf *lrYMin*.

uiCycCl: Aufrufzyklus des Bausteins als Vielfaches der Zykluszeit. Ein Nulleintrag wird automatisch als *uiCycleCall*=1 gewertet.

Beispiel: *tTaskCycleTime* = 20ms, *uiCtrlCycleCall* =10 -> Der Regelalgorithmus wird alle 200ms aufgerufen. Damit werden aber auch nur alle 200ms die Ausgänge aktualisiert.

VAR_OUTPUT

```

lrY         : LREAL;
lrE         : LREAL;
bAct        : BOOL;
b2PtAct     : BOOL;
bARW        : BOOL;
bMinLim     : BOOL;
bMaxLim     : BOOL;
bDecLim     : BOOL;
bIncLim     : BOOL;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;

```

lrY: Stellgröße. Bereich durch *lrYMin* und *lrYMax* eingeschränkt.

lrE: Regelabweichung (die Berechnung ist abhängig vom Wirksinn [► 158])

bAct: Der Regler ist aktiv, das heißt freigegeben (*bEn* = TRUE) und nicht im Fehlerzustand (*bErr* = FALSE).

bARW: Anti-Reset-Windup-Funktion ist aktiv

b2PtAct: wird nicht mehr benötigt.

bMaxLim: Die Stellgröße hat ihren oberen Grenzwert erreicht.

bMinLim: Die Stellgröße hat ihren unteren Grenzwert erreicht.

bDecLim: Die Stellgrößen-Steigung hat ihren Grenzwert für das maximale Abfallen erreicht, siehe *udiMaxToMin* (VAR_INPUT).

bIncLim : Die Stellgrößen-Steigung hat ihren Grenzwert für den maximale Anstieg erreicht, siehe *udiMinToMax* (VAR_INPUT).

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

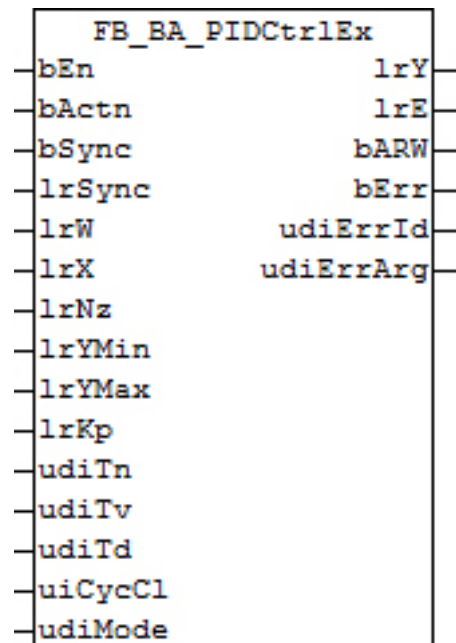
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.59 FB_BA_PIDCtrlEx

Universeller PID-Regler, wahlweise in Parallelstruktur oder mit vorgelagertem P-Glied

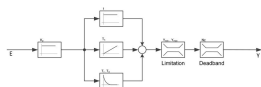


Funktionsbeschreibung

Dieser Regler ist kann wahlweise in Parallelstruktur, als auch mit vorgelagertem P-Glied betrieben werden. Dieses wird durch den Eingang *udiMode* vorgegeben.

Wirkungsplan

udiMode=0 (P-Anteil vorgelagert):



udiMode=1 (Parallelstruktur):



Passiv-Verhalten (bEn = FALSE oder bErr = TRUE)

Die Ausgänge werden wie folgt gesetzt:

IrY	0,0
IrE	0,0
bAct	FALSE
bARW	FALSE
bMinLim	FALSE
bMaxLim	FALSE
bIncLim	FALSE
bDecLim	FALSE

Im Fehlerfall steht *bErr* auf TRUE - *udiErrId* und *udiErrArg* beschreiben den Fehler. Die internen Werte für P-, I-, und D-Anteil werden auf 0 gesetzt, ebenso die Werte für den I- und D-Anteil vom vorhergehenden Zyklus. Damit wird die Stellgröße bei einem Neustart im ersten Zyklus "sauber", das heißt ohne Vergangenheitswerte berechnet.

Aktiv-Verhalten (bEn = TRUE und bErr = FALSE)

Im ersten Zyklus werden I- und D-Anteil wie bereits erwähnt ohne Vergangenheitswerte berechnet und somit "sauber" aufgestartet. Ein positives Signal an *bSync* setzt den I-Anteil so, dass die Stellgröße den Wert *IrSync* annimmt. Diese Methode kann, wenn *bEn* und *bSync* gleichzeitig gesetzt werden, zum Setzen eines Initialwerts genutzt werden, von dem aus die Regelung "losläuft". Ist der I-Anteil nicht aktiv, so wird der D-Anteil entsprechend gesetzt. Zu beachten ist, dass intern nur die steigende Flanke von *bSync* ausgewertet wird, handelt es sich doch um eine Setz-Aktion. Für ein erneutes Synchronisieren, etwa auf einen Übergabewert, muss am Eingang *bSync* ein erneutes TRUE-Signal angelegt werden. Ist der I-Anteil aktiv so sorgt der Regler dafür, dass dieser fest gehalten wird, sollte sich der Reglerausgang *IrY* an den Grenzen *IrYMin* oder *IrYMax* befinden und noch weiter fallen bzw. steigen wollen. Dieses Verfahren wird Anti-Reset-Wind-Up genannt und trägt dafür Sorge, dass der I-Anteil immer nur so groß ist, dass die Stellgröße bei entsprechender Regelabweichung sofort wieder Werte innerhalb der Grenzen annehmen kann, ohne dass ein zu groß gewordener Integralanteil zunächst noch abgebaut werden muss.

Wirksinn

Mit *bActn* = FALSE wird der Wirksinn des Reglers so umgekehrt, dass eine Regelabweichung kleiner als 0 eine Stellgrößenänderung ins Positive bewirkt. Dies wird dadurch erreicht, dass die Regelabweichung negativ berechnet wird:

bActn	rXW (Regelabweichung)	Wirksinn
TRUE	IrX-IrW (Istwert-Sollwert)	direkt (Kühlen)
FALSE	IrW-IrX (Sollwert-Istwert)	indirekt (Heizen)

Anti-Reset-Windup bei Erreichen des Maximal- bzw Minimalwerts

Erreicht der Regler am Ausgang seine Obergrenze und ist die Regelabweichung weiterhin positiv, so wird der Integralanteil sich so lange weiter erhöhen, bis die Regelabweichung wieder kleiner oder gleich Null ist. Dies kann dazu führen, dass sich unnötigerweise ein sehr hoher Integralanteil aufbaut, der bei einer Vorzeichenänderung der Regelabweichung erst wieder abgebaut werden muss und das Regelverhalten träge macht. Das gleiche gilt auch bei Erreichen des Minimalwerts am Ausgang bei einer weiterhin negativen Regelabweichung. Um diesem vorzubeugen wird, sobald einer der beiden Fälle eintritt, der I-Anteil nicht weiter berechnet, sondern auf dem Wert des vorhergehenden SPS-Zyklus gehalten. Tritt eine Vorzeichenumkehr ein, also Maximalwert erreicht und Regelabweichung kleiner als 0.0 bzw. Minimalwert erreicht und Regelabweichung größer als 0.0, so wird der I-Anteil wieder berechnet.

Neutrale Zone (Deadband)

Ein Wert von *IrNZ* > 0.0 gibt die Funktion der neutralen Zone (Deadband) frei. Ein Wert gleich Null deaktiviert das Totbandglied und die Werte am Eingang werden direkt durchgereicht.

Wenn das Totbandglied aktiv ist wird der Eingangswert erst dann als neuer Ausgangswert durchgereicht, wenn der Absolutwert der Differenz vom Eingangswert zum jetzigen Ausgangswert größer oder gleich *IrNZ/2* ist.

Beispiel: *IrNZ* = 1, *IrYin* = 55.0, *IrYout*= 55.0 (Wert wurde gerade durchgereicht)

PLC-Zyklus+1	<i>IrYin</i> =55.2	<i>IrYout</i> =55.0
PLC-Zyklus+2	<i>IrYin</i> =55.3	<i>IrYout</i> =55.0
PLC-Zyklus+3	<i>IrYin</i> =55.1	<i>IrYout</i> =55.0
PLC-Zyklus+4	<i>IrYin</i> =55.6	<i>IrYout</i> =55.6
PLC-Zyklus+5	<i>IrYin</i> =55.4	<i>IrYout</i> =55.6
PLC-Zyklus+6	<i>IrYin</i> =55.3	<i>IrYout</i> =55.6
PLC-Zyklus+7	<i>IrYin</i> =55.1	<i>IrYout</i> =55.1

Durch diese Funktion sollen unnötig viele Stellimpulse vermieden werden.

Autokorrektur von Parametern

Bei einigen Parametern macht es Sinn, sie automatisch zu korrigieren bzw. zu begrenzen, ohne eine Fehlermeldung auszugeben, da die Erwartung des Anwenders eindeutig vorhersehbar ist:

- *IrSync* > *IrYMax* -> *IrSync* := *IrYMax*
- *IrSync* < *IrMin* -> *IrSync* := *IrMin*
- *uiCycCl* = 0 -> *uiCycCl* := 1

Ein-Ausgänge

VAR_INPUT

```

bEn      : BOOL;
bActn    : BOOL;
bSync    : BOOL;
lrSync   : LREAL;
lrW      : LREAL;
lrX      : LREAL;
lrNZ     : LREAL;
lrKp     : LREAL;
udiTn    : UDINT;
udiTv    : UDINT;
udiTd    : UDINT;
uiCycCl  : UINT;
udiMode  : UDINT;
    
```

bEn: Regleraktivierung

bActn: Wirksinn [► 162] des Reglers

bSync / IrSync: Synchronisationsbefehl: Ausgangswert *IrY* auf *IrSync* setzen.

lrW: Sollwert

lrX: Istwert

lrNZ: Neutrale Zone

IrYMin: Untere Ausgabebegrenzung des Reglers

IrYMax: Obere Ausgabebegrenzung des Reglers

lrKp: Reglerverstärkung. Wirkt nur auf den P-Anteil

udiTn: Nachstellzeit des I-Anteiles [ms] . Ein Nullwert an diesem Parameter schaltet den I-Anteil ab.

udiTv: Vorhaltezeit des D-Anteiles [ms] . Ein Nullwert an diesem Parameter schaltet den D-Anteil ab.

udiTd: Dämpfungszeit des D-Anteiles [ms] . **uiCycCl:** Aufrufzyklus des Bausteins als Vielfaches der Zykluszeit. Ein Nulleintrag wird automatisch als *uiCycleCall*=1 gewertet.

Beispiel: *tTaskCycleTime* = 20ms, *uiCtrlCycleCall* =10 -> Der Regelalgorithmus wird alle 200ms aufgerufen. Damit werden aber auch nur alle 200ms die Ausgänge aktualisiert.

udiMode: *udiMode*=0: Regler mit vorgelagertem P-Anteil, *udiMode*=1: Regler in Parallelstruktur.

VAR_OUTPUT

```
lrY      : LREAL;
lrE      : LREAL;
bARW     : BOOL;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

lrY: Stellgröße. Bereich durch *lrYMin* und *lrYMax* eingeschränkt.

lrE: Regelabweichung (die Berechnung ist abhängig vom [Wirksinn](#) [[▶](#) 162])

bARW: Anti-Reset-Windup-Funktion ist aktiv.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

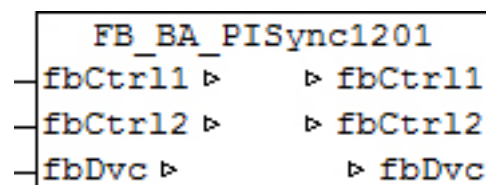
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [[▶](#) 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.60 FB_BA_PISync1201

Parameter-Synchronisation für 2 Regler vom Typ [FB_BACnetLoop1201](#) [[▶](#) 99]. Dieser Baustein kann beispielsweise zum Abgleich der Führungsregler dienen, welche zur Erzeugung der Zuluftsollwerte für einen raumlufttechnische Anlage mit einem Luftkühler und einem Luffterhitzer eingesetzt werden.



Funktionsbeschreibung

Dieser Baustein synchronisiert automatisch die P- und I- Parameter sowie den minimalen und maximalen Ausgabewert *lrYMin* und *lrYMax* von 2 Reglern des Typs [FB_BACnetLoop1201](#) [[▶](#) 99]. Es ist dabei egal, wo die Änderung stattfindet: An einem der beiden Regler in der PLC oder im BACnet.

Der Baustein erkennt automatisch eine Änderung und überträgt diese dann per ADS an den jeweilig anderen Regler. Zu diesem Zweck müssen beide Regler über die IN-OUT-Variablen *fbCtrl1* und *fbCtrl2* am Baustein referenziert werden - *fbDvc* ist die Referenz auf das BACnet-Device unter dem beide Bausteine laufen.

Ein-Ausgänge

VAR_IN_OUT

```
fbCtrl1 : FB_BACnetLoop1201;
fbCtrl2 : FB_BACnetLoop1201;
fbDvc   : FB_BACnet_Device;
```

fbCtrl1: Referenz auf den Regler Nr. 1 des Parameterabgleichs

fbCtrl2: Referenz auf den Regler Nr. 2 des Parameterabgleichs

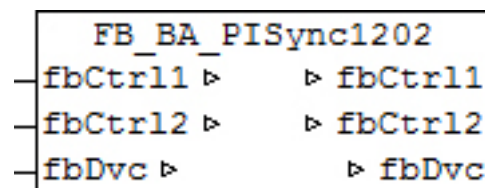
fbDvc: Referenz auf den Baustein des BACnet-Device-Objekts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.61 FB_BA_PISync1202

Parameter-Synchronisation für 2 Regler vom Typ `FB_BACnetLoop1202` [▶ 103]. Dieser Baustein kann beispielsweise zum Abgleich der Führungsregler dienen, welche zur Erzeugung der Zuluftsollwerte für einen raumluftechnische Anlage mit einem Luftkühler und einem Luftherhitzer eingesetzt werden.



Funktionsbeschreibung

Dieser Baustein synchronisiert automatisch die P- und I- Parameter sowie den minimalen und maximalen Ausgabewert *IrYMin* und *IrYMax* von 2 Reglern des Typs `FB_BACnetLoop1202` [▶ 103]. Es ist dabei egal, wo die Änderung stattfindet: An einem der beiden Regler in der PLC oder im BACnet.

Der Baustein erkennt automatisch eine Änderung und überträgt diese dann per ADS an den jeweilig anderen Regler. Zu diesem Zweck müssen beide Regler über die IN-OUT-Variablen *fbCtrl1* und *fbCtrl2* am Baustein referenziert werden - *fbDvc* ist die Referenz auf das BACnet-Device unter dem beide Bausteine laufen.

Ein-Ausgänge

VAR_IN_OUT

```
fbCtrl1 : FB_BACnetLoop1202;
fbCtrl2 : FB_BACnetLoop1202;
fbDvc   : FB_BACnet_Device;
```

fbCtrl1: Referenz auf den Regler Nr. 1 des Parameterabgleichs

fbCtrl2: Referenz auf den Regler Nr. 2 des Parameterabgleichs

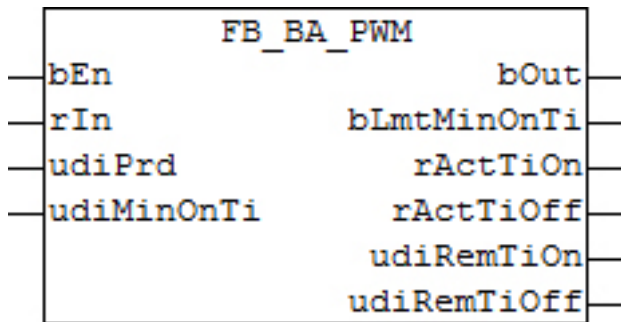
fbDvc: Referenz auf den Baustein des BACnet-Device-Objektes

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.62 FB_BA_PWM

Puls-Weiten-Modulations-Baustein



Funktionsbeschreibung

Der Funktionsbaustein errechnet aus einem analogen Eingangssignal rIn (0..100%, **intern fest begrenzt**) und einer Periodendauer $udiPrd$ [s] eine Ein- und eine Ausschaltzeit, welche an den Ausgängen als $rActTiOn$ und $rActTiOff$ in Sekunden angezeigt werden.

Dabei entsprechen:

- 100% am Eingang einer Einschaltzeit von der gesamten Periodendauer $udiPrd$ und einer Ausschaltzeit von 0s.
- 0% am Eingang einer Einschaltzeit von 0s und einer Ausschaltzeit von der gesamten Periodendauer $udiPrd$.

Darüber hinaus gibt es die Möglichkeit über $udiMinOnTi$ [s] die Einschaltdauer nach unten hin zu begrenzen, um Schäden an Antrieben durch zu kurze Stellimpulse zu vermeiden. Dieses Verhalten gilt jedoch nur für $0 > rIn > 100!$

Für die Eingangssignale:

- $rIn=0$: Einschaltzeit=0, Ausschaltzeit= $udiPrd$ -> $bOut$ bleibt dauerhaft FALSE
- $rIn=100$: Einschaltzeit= $udiPrd$, Ausschaltzeit=0 -> $bOut$ bleibt dauerhaft TRUE

Schaltverhalten

1. Ein FALSE-Signal am Eingang bEn setzt den Baustein außer Funktion und bQ auf FALSE. Es werden lediglich die Ein- und Ausschaltzeiten kontinuierlich berechnet und an den Ausgängen $rActTiOn$ und $rActTiOff$ [s] angezeigt.
2. Eine steigende Flanke am Eingang bEn aktiviert den Baustein: Er springt zunächst in einen Entscheidungs-Schritt. Je nach vorherigem Zustand vom Schaltausgang bQ wird nun in den umschaltenden Schritt gesprungen, **es sei denn**, der Eingang rIn steht auf 0.0. Dann erfolgt ein Sprung in den Aus-Schritt ($bQ=FALSE$).
3. In dem jeweils aktiven Schritt (EIN oder AUS) läuft nun ein Countdown-Timer mit dem aktuell errechneten Startwert ab, der sich aus dem Puls-Pause-Verhältnis ergibt. Der Ein- bzw. Aus-Schritt wird mit der errechneten Zeit durchlaufen, egal, ob sich währenddessen das Puls-Pause-Verhältnis ändert. Der jeweilige Countdown wird an den Ausgängen $udiRemTiOn$ und $udiRemTiOff$ in vollen Sekunden angezeigt.
4. Nach Ablauf des Ein- bzw. des Aus-Schrittes erfolgt ein Rücksprung in den Entscheidungsschritt (Punkt 2).

Ein-Ausgänge

VAR_INPUT

```
bEn      : BOOL;
rIn      : REAL;
udiPrd   : UDINT;
udiMinOnTi : UDINT;
```

bEn : Aktivierung der Puls-Weiten-Modulation

rIn: Eingangssignal, intern auf 0..100% fest begrenzt

udiPrd: Periodenzeit [s]

udiMinOnTi: Minimale Einschaltzeit [s]

VAR_OUTPUT

```
bOut     : BOOL;
bLmtSwiTi : BOOL;
rActTiOn : REAL;
rActTiOff : REAL;
udiRemOnTi : UDINT;
udiRemOffTi : UDINT;
```

bOut: PWM-Ausgang.

bLmtSwiTi: Informationsausgang, dass das Eingangssignal so niedrig ist, dass mit der minimalen Einschaltzeit begrenzt wird.

rActTiOn: Informationsausgang: Errechnete Einschaltzeit

rActTiOff: Informationsausgang: Errechnete Ausschaltzeit

udiRemOnTi: Countdown Einschalttimer

udiRemOffTi: Countdown Ausschalttimer

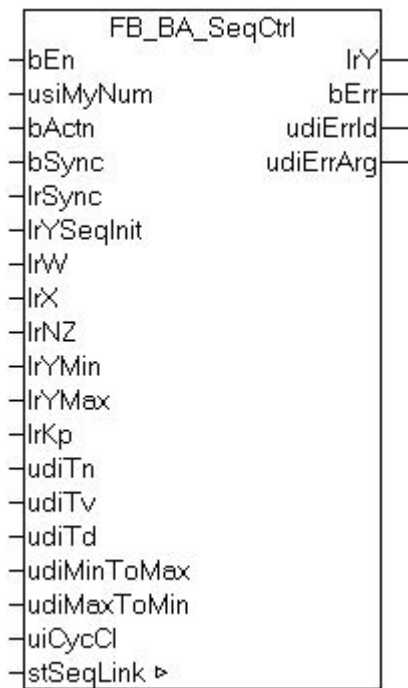
Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.63 FB_BA_SeqCtrl

PID-Regler als Bestandteil einer Sequenz. Dieser Regler ist in seinen Funktionalitäten identisch mit dem [FB_BA_PIDCtrl \[▶ 157\]](#). Er wird jedoch, wenn er durch *bEn*=TRUE freigegeben ist, über eine übergeordneten Kontrollbaustein [FB_BA_SeqLink \[▶ 171\]](#) gesteuert.

Der Datenaustausch zwischen dem Steuerbaustein [FB_BA_SeqLink \[▶ 171\]](#) und den Sequenzreglern [FB_BA_SeqCtrl](#) erfolgt über die Strukturvariable [stSeqLink \[▶ 333\]](#).



Funktionsbeschreibung

Heiz- Kühlfolge

Die Reglersequenz ist so zu konzipieren, dass die Sequenzregler mit niedriger Ordnungszahl zum Heizen und die mit höherer zum Kühlen zu nutzen sind. Dabei ist nur ein Wechsel zulässig:

- Sequenzregler n ($usiMyNum=n$, $bActn=TRUE$)
- Sequenzregler $n+1$ ($usiMyNum=n+1$, $bActn=FALSE$)

Möglich ist aber auch die ausschließliche Programmierung von Kühl- bzw. Heizreglern.

Eine Parametrierung, die dieser Konvention widerspricht wird erkannt und als Fehler am Steuerbaustein [FB_BA_SeqLink \[► 171\]](#) angezeigt.

Reglerausgabe

Während der Steuerbaustein [FB_BA_SeqLink \[► 171\]](#) vorgibt, welcher Sequenzregler aktiv ist, entscheidet jeder Sequenzregler selbst, was er am Steuerausgang IrY ausgibt. Da jeder Sequenzregler über die In-Out-Variable [stSeqLink \[► 333\]](#) über den Zustand der anderen Regler Bescheid weiß kann er 4 Fälle unterscheiden:

1. Der Sequenzregler erkennt, dass er selbst und alle anderen Sequenzregler nicht freigegeben sind, sei es durch fehlende Freigaben (bEn) am Eingang oder durch einen erkannten Fehler am Steuerbaustein [FB_BA_SeqLink \[► 171\]](#)
-> Der Sequenzregler deaktiviert seinen internen PID-Regler und gibt am Steuerausgang IrY 0.0 aus.
2. Der Sequenzregler erkennt, dass er freigegeben und er selbst vom Steuerbaustein [FB_BA_SeqLink \[► 171\]](#) als aktiv gesetzt ist
-> Der Sequenzregler aktiviert seinen internen PID-Regler. Dessen Ausgangssignal wird am Steuerausgang IrY ausgegeben.
3. Der Sequenzregler erkennt, dass er freigegeben ist, jedoch ist ein Sequenzregler **höherer** Ordnungszahl vom Steuerbaustein [FB_BA_SeqLink \[► 171\]](#) als aktiv gesetzt
-> Der Sequenzregler deaktiviert seinen internen PID-Regler. Befindet sich der Sequenzregler im Heizbetrieb ($bActn=FALSE$), so wird er seinen Minimalwert $IrYMin$ am Steuerausgang IrY ausgeben. Befindet er sich hingegen im Kühlbetrieb ($bActn=TRUE$), so gibt er seinen Maximalwert $IrYMax$ am Steuerausgang IrY aus.

- Der Sequenzregler erkennt, dass er freigegeben ist, jedoch ist ein Sequenzregler **niedrigerer** Ordnungszahl vom Steuerbaustein [FB_BA_SeqLink \[► 171\]](#) als aktiv gesetzt
-> Der Sequenzregler deaktiviert seinen internen PID-Regler. Befindet sich der Sequenzregler im Heizbetrieb (*bActn*=FALSE), so wird er seinen Maximalwert *lrYMax* am Steuerausgang *lrY* ausgeben. Befindet er sich hingegen im Kühlbetrieb (*bActn*=TRUE), so gibt er seinen Minimalwert *lrYMin* am Steuerausgang *lrY* aus.

Synchronisation

Wird ein Sequenzregler von der übergeordneten Steuerung aktiv geschaltet, so erfolgt immer eine Synchronisation, d.h. der Regler beginnt mit einem festen Startwert am Ausgang *lrY*. Dabei werden 3 Fälle unterschieden:

- Die gesamte Sequenzregelung wurde gerade über den Eingang *bEn* der übergeordneten Steuerung [FB_BA_SeqLink \[► 171\]](#) eingeschaltet. Der Regler, mit der Ordnungszahl *usiStartSeq* am Eingang von ist [FB_BA_SeqLink \[► 171\]](#) ist der Startregler.
-> Der Sequenzregler wird mit dem Wert, der an seinem Eingang *lrYSeqInit* eingetragen ist, synchronisiert.
- Der Sequenzregler, der gerade aktiviert wurde, hatte einen Regler höherer Ordnungszahl als "Vorgänger"
-> Befindet sich der Sequenzregler im Heizbetrieb (*bActn*=FALSE), so wird er mit seinem Minimalwert *lrYMin* synchronisiert. Befindet er sich hingegen im Kühlbetrieb (*bActn*=TRUE), so ist der Synchronisationswert sein Maximalwert *lrYMax*.
- Der Sequenzregler, der gerade aktiviert wurde, hatte einen Regler niedrigerer Ordnungszahl als "Vorgänger"
-> Befindet sich der Sequenzregler im Heizbetrieb (*bActn*=FALSE), so wird er mit seinem Maximalwert *lrYMax* synchronisiert. Befindet er sich hingegen im Kühlbetrieb (*bActn*=TRUE), so ist der Synchronisationswert sein Minimalwert *lrYMin*.

Jeder Sequenzregler kann auch über Wertvorgabe *lrSync* und Aktivierung *bSync* extra synchronisiert werden, wenn er gerade von der übergeordneten Steuerung aktiv gesetzt ist. Ein ständiges TRUE-Signal am Eingang *bSync* (etwa durch Versehen) wird intern durch Flankenbildung abgefangen und behindert dadurch nicht die oben beschriebene Synchronisation bei Aktiv-Setzung.

Startverhalten

Um ein "vernünftiges" Einregeln der gesamten Regelsequenz zu ermöglichen, wird der Startregler mindestens für die Zeit *udiIniswiOvrDly* [s], welche am Baustein [FB_BA_SeqLink \[► 171\]](#) eingetragen ist, aktiv gehalten, d.h. in dieser Zeit findet kein Umschalten auf einen anderen Regler dieser Sequenz statt. Der Ausgang *lrY* des Startreglers wird **einmalig** auf seinen Wert *lrYSeqInit* synchronisiert.

Ein-Ausgänge

VAR_INPUT

```

bEn          : BOOL;
usiMyNum     : USINT;
bActn       : BOOL;
bSync       : BOOL;
lrSync      : LREAL;
lrYSeqInit  : LREAL;
lrW         : LREAL;
lrX         : LREAL;
lrNZ        : LREAL;
lrYMin      : LREAL;
lrYMax      : LREAL;
lrKp        : LREAL;
udiTn       : UDINT;
udiTv       : UDINT;
udiTd       : UDINT;
udiMinToMax : UDINT;
udiMaxToMin : UDINT;
uiCycCl     : UINT;

```

bEn: Aktivierung des Sequenzreglers

usiMyNum: Ordnungszahl des Sequenzreglers

bActn: Wirksinn-Umkehrung des Reglers. Bei Heiz-Kühleinsatz: *bActn*=FALSE entspricht Heizbetrieb, *bActn*=TRUE entspricht Kühlbetrieb.

bSync / IrSync: Synchronisationsbefehl: Ausgangswert *IrY* auf *IrSync* setzen.

IrYSeqInit: Startwert des Reglers nach dem Neustart der gesamten Regelsequenz.

IrW: Sollwert

IrX: Istwert

IrNZ: Neutrale Zone

IrYMin: Untere Ausgabebegrenzung des Reglers [%]. Wählbarer Bereich: 0..100%.

IrYMax: Obere Ausgabebegrenzung des Reglers [%]. Wählbarer Bereich: 0..100%.

IrKp: Reglerverstärkung. Wirkt nur auf den P-Anteil.

udiTn: Nachstellzeit des I-Anteiles [ms] . Ein Nullwert an diesem Parameter schaltet den I-Anteil ab.

udiTv: Vorhaltezeit des D-Anteiles [ms] . Ein Nullwert an diesem Parameter schaltet den D-Anteil ab.

udiTd: Dämpfungszeit des D-Anteiles [ms]

udiMinToMax: Steigungsbegrenzung des Reglerausgangs für Anstieg: *udiMinToMax* [s] bezogen auf eine Änderung von *IrYMin* auf *IrYMax*.

udiMaxToMin: Steigungsbegrenzung des Reglerausgangs für Abfall: *udiMaxToMin* [s] bezogen auf eine Änderung von *IrYMax* auf *IrYMin*.

uiCycCl: Aufrufzyklus des Bausteins als Vielfaches der Zykluszeit. Ein Nulleintrag wird automatisch als *uiCycleCall*=1 gewertet.

Beispiel: *tTaskCycleTime* = 20ms, *uiCtrlCycleCall* =10 -> Der Regelalgorithmus wird alle 200 ms aufgerufen. Damit werden aber auch nur alle 200ms die Ausgänge aktualisiert.

VAR_IN_OUT

```
stSeqLink : ST_BA_SeqLink;
```

stSeqLink: Daten- und Befehlsstruktur zwischen den einzelnen Sequenzreglern und dem Steuerbaustein FB_BA_SeqLink [► 171].



Tragen mehrere Sequenzregler dieselbe Nummer (*usiMyNum*), so wird dies erkannt und als Fehler sowohl am Sequenzregler als auch am Steuerbaustein FB_BA_SeqLink [► 171] ausgegeben.

VAR_OUTPUT

```
IrY      : LREAL;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

IrY: Stellgröße. Bereich: 0..100%, soweit nicht weiter durch *IrYMin* und *IrYMax* eingeschränkt.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

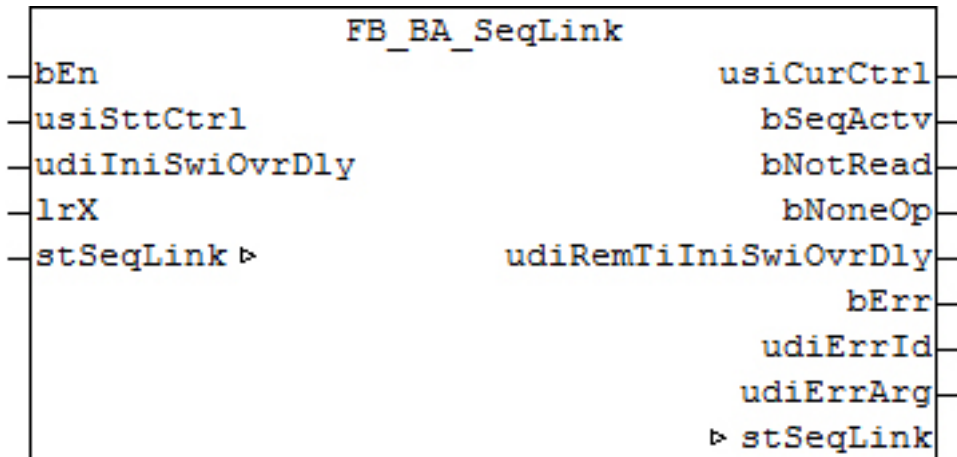
Siehe auch:

[ST_BA_SeqLink / ST_BA_SeqLinkData](#) [▶ 333]

8.2.64 FB_BA_SeqLink

Dieser Baustein stellt die übergeordnete Steuereinheit dar, die vorgibt, welcher Sequenzregler gerade aktiv ist.

Der Datenaustausch zwischen dem Steuerbaustein FB_BA_SeqLink und den Sequenzreglern [FB_BA_SeqCtrl](#) [▶ 167] erfolgt über die Strukturvariable [stSeqLink](#) [▶ 333].



Funktionsbeschreibung

Startverhalten

Eine TRUE-Signal am Eingang *bEn* aktiviert die komplette Sequenzregelung. Der Baustein wird zunächst denjenigen Sequenzregler aktiv schalten, welcher an *usiStartSeq* benannt ist. Alle anderen Sequenzregler richten Ihren Ausgabewert nach der Rangordnung des aktiven Reglers, siehe [FB_BA_SeqCtrl](#) [▶ 167]. Der Startregler wird beim Sequenz-Start einmalig auf seinen Wert *lrSyncVal* gesetzt.

Um ein "vernünftiges" Einregeln der gesamten Regelsequenz zu ermöglichen, wird der Startregler mindestens für die Zeit *udiIniSwiOvrDly* lang aktiv gehalten, d.h. in dieser Zeit findet kein Umschalten auf eine anderen Regler dieser Sequenz statt.

Umschaltverhalten

Erreicht ein Sequenzregler seinem Maximal- oder Minimalwert, so wird je nach Regler-Wirksinn auf den nächsten Regler in der Rangfolge umgeschaltet, sofern der Istwert den Sollwert des nächsten Reglers unterschreitet bzw. überschreitet.

Dabei werden 4 Fälle unterschieden:

- Der noch aktive Regler hat direkten Wirksinn (Kühlen) und befindet sich auf seinem Maximalwert: Es wird in der Rangfolge der nächst höhere Regler gewählt, wenn der Istwert den Sollwert dieses Reglers überschreitet.
- Der noch aktive Regler hat direkten Wirksinn (Kühlen) und befindet sich auf seinem Minimalwert: Es wird in der Rangfolge der nächst niedrigere Regler gewählt, wenn der Istwert den Sollwert dieses Reglers unterschreitet.
- Der noch aktive Regler hat indirekten Wirksinn (Heizen) und befindet sich auf seinem Maximalwert: Es wird in der Rangfolge der nächst niedrigere Regler gewählt, wenn der Istwert den Sollwert dieses Reglers unterschreitet.
- Der noch aktive Regler hat indirekten Wirksinn (Heizen) und befindet sich auf seinem Minimalwert: Es wird in der Rangfolge der nächst höhere Regler gewählt, wenn der Istwert den Sollwert dieses Reglers überschreitet.

Abschaltverhalten

Wird einem Regler der Sequenz seine Freigabe weggenommen oder geht er in Störung, so ist er für die gesamte Sequenz nicht mehr verfügbar.

Handelt es sich dabei nicht um den vorher aktiven Regler, so kann es, je nachdem welche Stellgröße dieser Regler ausgegeben hat, zu einer Temperaturänderung kommen, die durch die Reglersequenz nach Möglichkeit wieder ausgeglichen wird.

Handelt es sich jedoch um den aktiven Regler, dessen Freigabe weg genommen wird, so muss auf den nächst "sinnvollen" umgeschaltet werden. Der Sequenz-Link-Baustein geht dabei nach folgenden Regeln vor:

- Der gerade deaktivierte Regler hatte direkten Wirksinn (Kühlen)
Es gibt einen betriebsbereiten Regler mit höherer Ordnungszahl → auf den nächst höheren betriebsbereiten Regler schalten. Es gibt nur einen betriebsbereiten Regler mit niedrigerer Ordnungszahl → auf den nächst niedrigeren betriebsbereiten Regler schalten. Es gibt keinen betriebsbereiten Regler mehr → Störmeldung.
- Der gerade deaktivierte Regler hatte indirekten Wirksinn (Heizen)
- Es gibt einen betriebsbereiten Regler mit niedrigerer Ordnungszahl → auf den nächst niedrigeren betriebsbereiten Regler schalten. Es gibt nur einen betriebsbereiten Regler mit höherer Ordnungszahl → auf den nächst höheren betriebsbereiten Regler schalten. Es gibt keinen betriebsbereiten Regler mehr → Störmeldung.

Zuschaltverhalten

Wird ein Regler der Sequenz hinzugefügt, so ist er in jedem Fall zunächst inaktiv und wird, je nach Wirksinn und Positionierung im Sinne der Sequenzreihenfolge, seinen Minimal- bzw. Maximalwert ausgeben. Die daraus resultierende Temperaturänderung wird durch die Reglersequenz nach Möglichkeit wieder ausgeglichen.

Fehlererkennung

Folgende Fehler werden durch den Sequenz-Link-Baustein erkannt:

1. Es existieren mindestens zwei Regler mit derselben Ordnungszahl, die der Sequenz zugeordnet sind.
2. Es wurde in der Reglerfolge mehr als nur ein Wechsel von indirektem auf direkten Wirksinn (Heizen auf Kühlen) festgestellt.
3. In der Reglerfolge sind die Sollwerte der Regler nicht monoton steigend.
4. Der Regler, welcher durch den Eingang *usiStartSeq* als Startregler definiert wurde, existiert nicht bzw. ist nicht dieser Sequenz zugeordnet (es wird dann der nächstgültige Regler gewählt).
5. Es wurde ein Startregler mit der Ordnungszahl 0 bzw. größer der maximal zulässigen Anzahl Regler festgelegt (es wird dann der nächstgültige Regler gewählt).

Nur der erste Fehler lässt den Sequenz-Link-Baustein in Störung gehen bzw. sperrt seine Abarbeitung (=FALSE). *bSeqActv* Keiner der zugehörigen Regler ist dann mehr aktiv und alle Regler geben die Stellgröße "0" aus. Der Baustein ist nicht aktiv:

Alle anderen Fehler werden zwar auch durch *bErr*=TRUE angezeigt, haben jedoch eine Fehlernummer (*udiErrId*), die sie lediglich als Warnung kennzeichnen.

Ein-Ausgänge

VAR_INPUT

```
bEn           : BOOL;
usiStartCtrl  : USINT;
udiIniSwiOvrDly : UDINT;
lrX           : LREAL
```

bEn: Aktivierung des Sequenzreglers

usiStartSeq: Ordnungszahl des Sequenzreglers, welcher bei der Gesamt-Aktivierung der Start-Regler sein soll.

udiIniSwiOvrDly: Wie eingangs bereits erwähnt, benötigt eine Sequenzregelung am Anfang eine gewisse Zeit um sich mit dem Startregler einzuregeln. Der Startregler bleibt für mindestens die Zeit *udiIniSwiOvrDly* [s] in der Sequenz aktiv, bis andere Kriterien auf einen anderen Regler schalten lassen. Siehe auch "Startverhalten [▶ 171]" und "Umschaltverhalten [▶ 171]".

IrX: Istwert der Regelung

VAR_IN_OUT

```
stSeqLink : ST_BA_SeqLink;
```

stSeqLink: Daten- und Befehlsstruktur zwischen den einzelnen Sequenzreglern und dem Steuerbaustein FB_BA_SeqLink . Über diese Struktur empfängt der Sequenz-Link-Baustein alle relevanten Daten der Sequenzregler und teilt diesen gleichzeitig mit, welcher der aktive ist.



Tragen mehrere Sequenzregler dieselbe Nummer (*usiMyNum*), so wird dies erkannt und als Fehler sowohl am Sequenzregler als auch am Steuerbaustein ausgegeben.

VAR_OUTPUT

```
usiCurCtrl      : USINT;
bSeqActv        : BOOL;
bNotRead        : BOOL;
bNoneOp         : BOOL;
udiRemTiIniSwiOvrDly : UDINT;
bErr            : BOOL;
udiErrId        : UDINT;
udiErrArg       : UDINT;
```

usiCurCtrl: Ordnungszahl des aktuell aktiven Sequenzreglers. Ist keiner aktiv, so wird hier 0 ausgegeben.

bSeqActv: Der Sequenzbaustein ist freigegeben (*bEn*) und hat keinen abschaltenden Fehler, siehe Fehlererkennung [▶ 172].

bNotRead: Jeder Sequenzregler übermittelt dem Steuerbaustein Daten über die Struktur *stSeqLink*. Solange noch keine Daten übermittelt wurden - dies ist beim Einschalten der SPS der Fall - steht dieser Ausgang auf TRUE.

bNoneOp: Dieser Ausgang wird auf TRUE geschaltet, wenn keiner der Sequenzregler seine eigene Freigabe (*bEn*=TRUE) hat.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind. Dieser Baustein stellt im Fehlerfall seine Abarbeitung u.U. nicht ein, siehe Fehlererkennung [▶ 172].

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [▶ 340].

Voraussetzungen

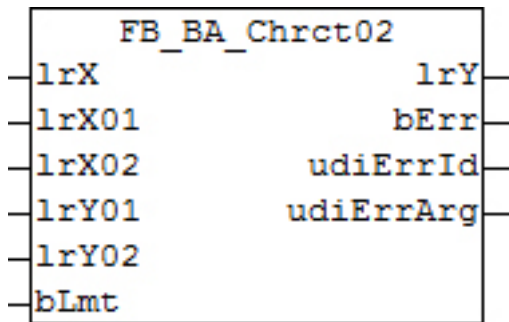
Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

Siehe auch:

[ST_BA_SeqLink / ST_BA_SeqLinkData \[▶ 333\]](#)

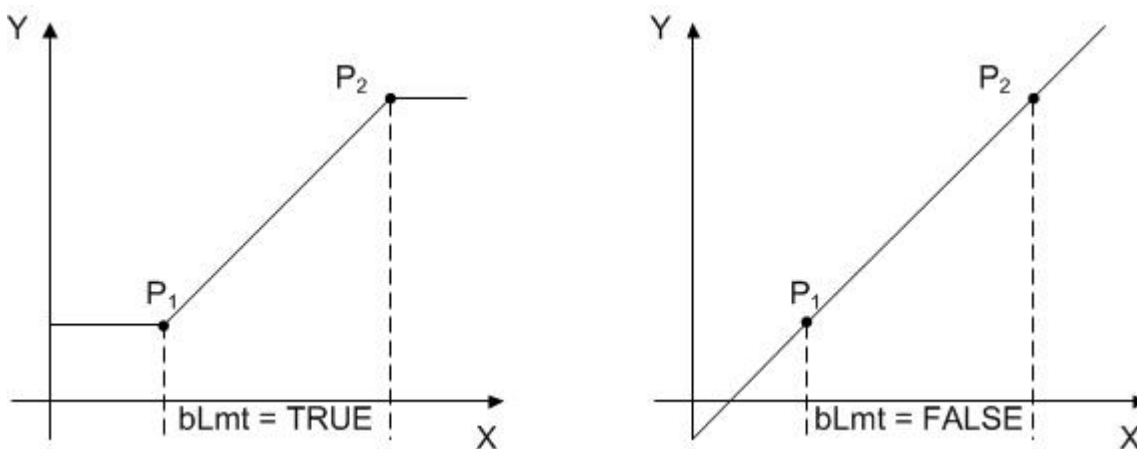
8.2.65 FB_BA_Chrct02

Lineare Interpolation mit 2 Stützstellen



Funktionsbeschreibung

Der Baustein FB_BA_Chrct02 stellt eine Linear-Interpolation mit 2 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen. Die Kennlinie wird durch die Stützpunkte $[lrX1/lrY1]$ und $[lrX2/lrY2]$ bestimmt. Wenn die Eingangsvariable *bLmt* TRUE ist, wird *lrY* von *lrY01* und *lrY02* begrenzt. Bei *bLmt* gleich FALSE erfolgt keine Begrenzung von *lrY*.



Fehlerbehandlung

Die Eingabewerte für $lrX[n+1]$ müssen immer mindestens um den Betrag von 0.0000001 größer als von $lrX[n]$ sein.

Im Fehlerfall zeigt die Variable *udiErrId* an, an welchem Punkt der Kennlinie die Werte nicht monoton steigend sind.

Ein-/Ausgänge

VAR_INPUT

```
lrX   : LREAL;
lrX01 : LREAL;
lrX02 : LREAL;
lrY01 : LREAL;
lrY02 : LREAL;
bLmt  : BOOL;
```

lrX: Eingangswert der Kennlinie.

lrX01: X-Wert für Stützpunkt P1.

lrX02: X-Wert für Stützpunkt P2.

lrY01: Y-Wert für Stützpunkt P1.

lrY02: Y-Wert für Stützpunkt P2.

bLmt: Begrenzung des Ausgangswerts *lrY*.

VAR_OUTPUT

```
lrY      : LREAL;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

lrY: Errechneter Ausgangswert der Kennlinie

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

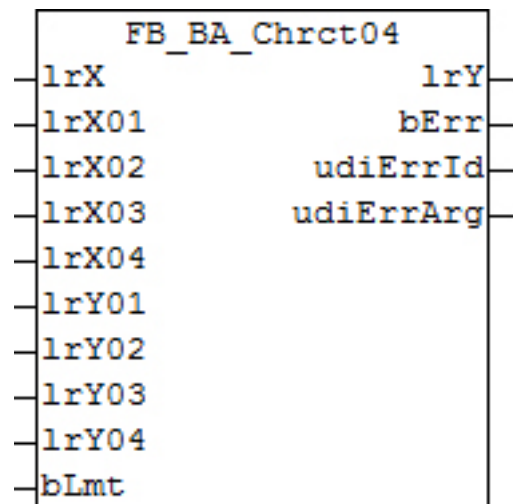
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

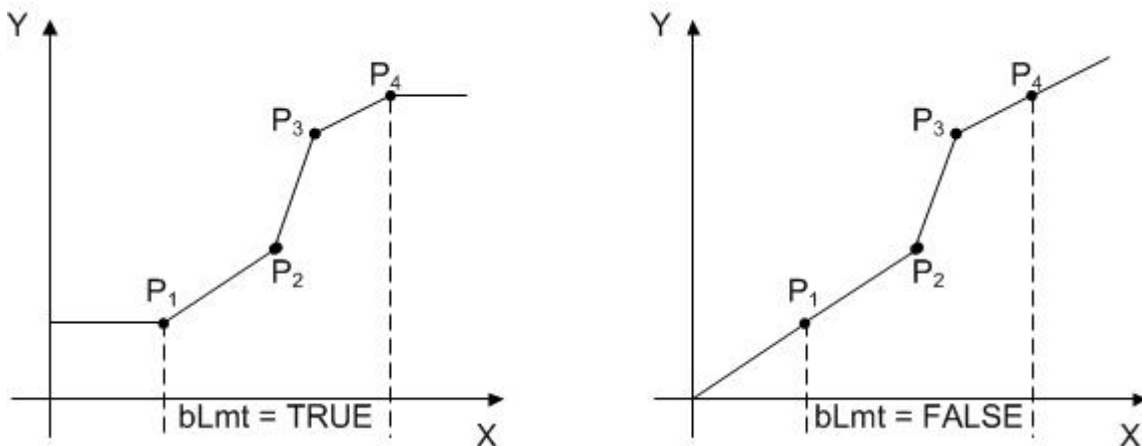
8.2.66 FB_BA_Chrct04

Lineare Interpolation mit 4 Stützstellen



Funktionsbeschreibung

Der Baustein FB_BA_Chrct04 stellt eine Linear-Interpolation mit 4 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen. Die Kennlinie wird durch die Stützpunkte [lrX1/lrY1] bis [lrX4/lrY4] bestimmt. Wenn die Eingangsvariable bLmt TRUE ist, wird lrY von lrY01 und lrY04 begrenzt. Bei bLmt gleich FALSE erfolgt keine Begrenzung von lrY.



Fehlerbehandlung

Die Eingabewerte für $lrX[n+1]$ müssen immer mindestens um den Betrag von 0.0000001 größer als von $lrX[n]$ sein.

Im Fehlerfall zeigt die Variable *udiErrId* an, an welchem Punkt der Kennlinie die Werte nicht monoton steigend sind.

VAR_INPUT

```
lrX    : LREAL;
lrX01  : LREAL;
lrX02  : LREAL;
lrX03  : LREAL;
lrX04  : LREAL;
lrY01  : LREAL;
lrY02  : LREAL;
lrY03  : LREAL;
lrY04  : LREAL;
bLmt   : BOOL;
```

lrX: Eingangswert der Kennlinie

lrX01: X-Wert für Stützpunkt P1

lrX02: X-Wert für Stützpunkt P2

lrX03: X-Wert für Stützpunkt P3

lrX04: X-Wert für Stützpunkt P4

lrY01: Y-Wert für Stützpunkt P1

lrY02: Y-Wert für Stützpunkt P2

lrY03: Y-Wert für Stützpunkt P3

lrY04: Y-Wert für Stützpunkt P4

bLmt: Begrenzung des Ausgangswerts *lrY*

Ein-/Ausgänge

VAR_OUTPUT

```
lrY      : LREAL;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

lrY: Errechneter Ausgangswert der Kennlinie

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

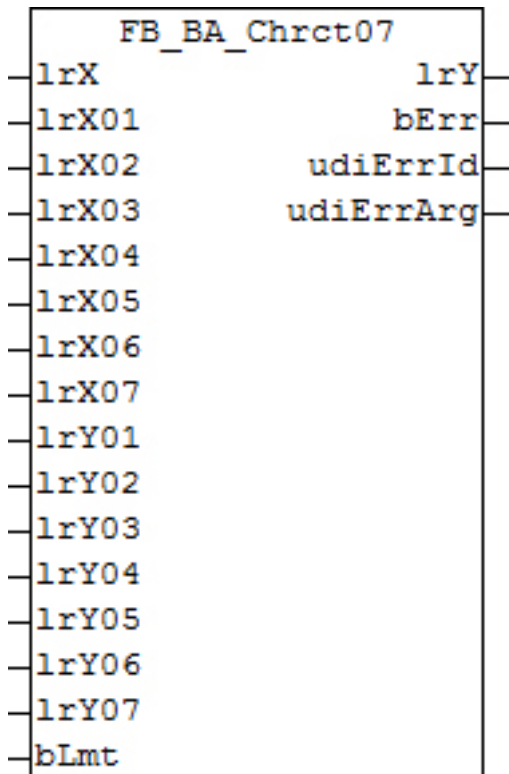
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

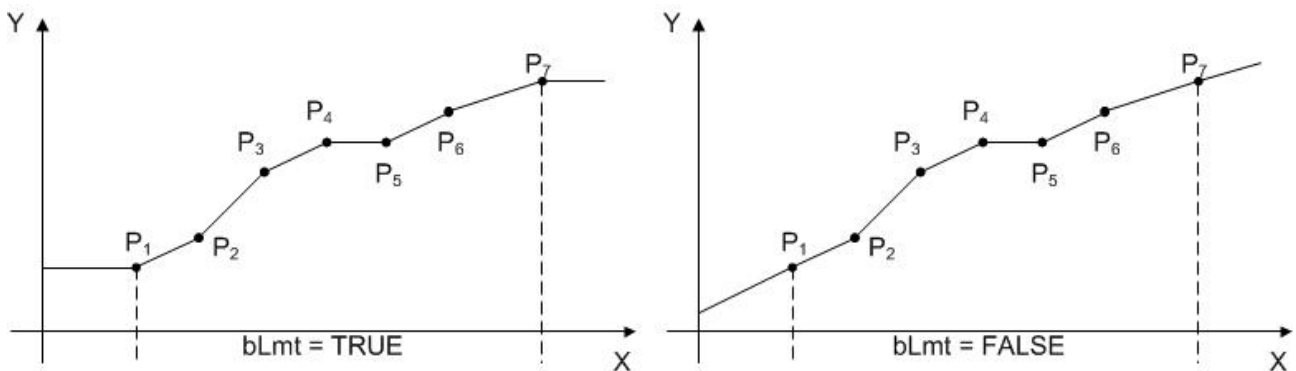
8.2.67 FB_BA_Chrct07

Lineare Interpolation mit 7 Stützstellen



Funktionsbeschreibung

Der Baustein FB_BA_Chrc07 stellt eine Linear-Interpolation mit 7 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen. Die Kennlinie wird durch die Stützpunkte [lrX1/lrY1] bis [lrX7/lrY7] bestimmt. Wenn die Eingangsvariable bLmt TRUE ist, wird lrY von lrY01 und lrY07 begrenzt. Bei bLmt gleich FALSE erfolgt keine Begrenzung von lrY.



Fehlerbehandlung

Die Eingabewerte für lrX[n+1] müssen immer mindestens um den Betrag von 0.0000001 größer als von lrX[n] sein.

Im Fehlerfall zeigt die Variable udiErrId an, an welchem Punkt der Kennlinie die Werte nicht monoton steigend sind.

Ein-/Ausgänge

VAR_INPUT

- lrX : LREAL;
- lrX01 : LREAL;
- lrX02 : LREAL;
- lrX03 : LREAL;
- lrX04 : LREAL;
- lrX05 : LREAL;
- lrX06 : LREAL;

```

lrX07 : LREAL;
lrY01 : LREAL;
lrY02 : LREAL;
lrY03 : LREAL;
lrY04 : LREAL;
lrY05 : LREAL;
lrY06 : LREAL;
lrY07 : LREAL;
bLmt  : BOOL;

```

lrX: Eingangswert der Kennlinie

lrX01: X-Wert für Stützpunkt P1
lrX02: X-Wert für Stützpunkt P2
lrX03: X-Wert für Stützpunkt P3
lrX04: X-Wert für Stützpunkt P4
lrX05: X-Wert für Stützpunkt P5
lrX06: X-Wert für Stützpunkt P6
lrX07: X-Wert für Stützpunkt P7

lrY01: Y-Wert für Stützpunkt P1
lrY02: Y-Wert für Stützpunkt P2
lrY03: Y-Wert für Stützpunkt P3
lrY04: Y-Wert für Stützpunkt P4
lrY05: Y-Wert für Stützpunkt P5
lrY06: Y-Wert für Stützpunkt P6
lrY07: Y-Wert für Stützpunkt P7

bLmt: Begrenzung des Ausgangswerts *lrY*

VAR_OUTPUT

```

lrY      : LREAL;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;

```

lrY: Errechneter Ausgangswert der Kennlinie

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

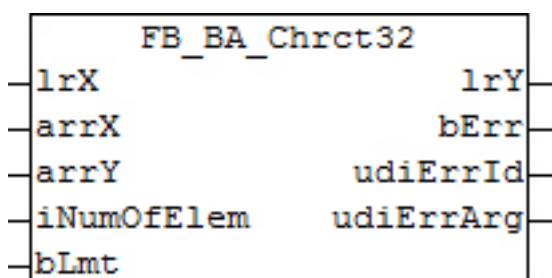
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

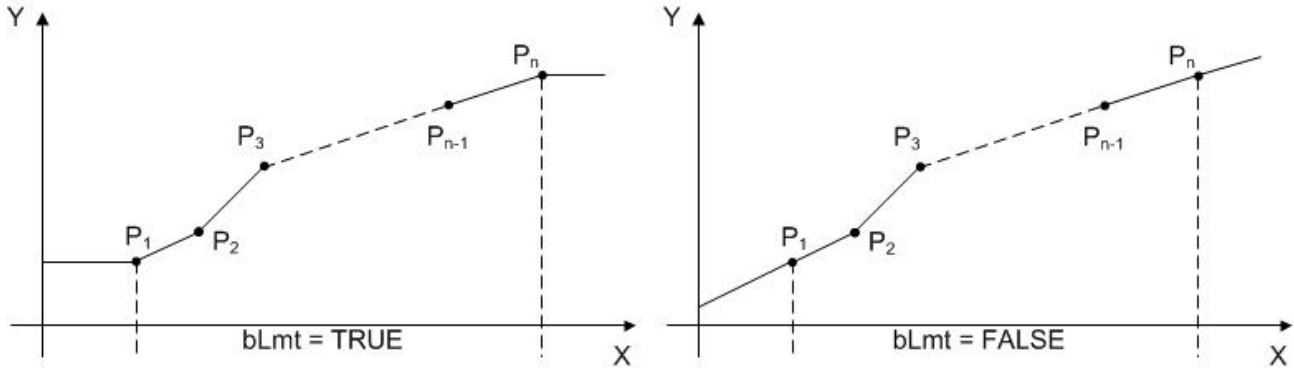
8.2.68 FB_BA_Chrct32

Lineare Interpolation mit bis zu 32 Stützstellen (parametrierbar)



Funktionsbeschreibung

Der Baustein FB_BA_Chrct32 stellt eine Linear-Interpolation mit bis zu 32 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen. Die Stützstellen werden hier, anders als bei den "kleineren" Interpolationsbausteinen FB_BA_Chrct02 [▶ 173], FB_BA_Chrct04 [▶ 175] und FB_BA_Chrct07 [▶ 176] der Übersicht halber über Feldvariablen [arrX[1]/arrY[1] bis [arrX[n]/arrY[n]] bestimmt. Wenn die Eingangsvariable bLmt TRUE ist, wird lrY von arrY[1] und arrY[n] begrenzt. Bei bLmt gleich FALSE erfolgt keine Begrenzung von lrY.



Fehlerbehandlung

Die Eingabewerte für lrX[n+1] müssen immer mindestens um den Betrag von 0.0000001 größer als von lrX[n] sein.

Im Fehlerfall zeigt die Variable udiErrId an, an welchem Punkt der Kennlinie die Werte nicht monoton steigend sind.

Der Parameter für die Anzahl der Stützstellen, iNumOfElem, muss im Bereich 2..32 liegen.

Ein-/Ausgänge

VAR_INPUT

```
lrX      : LREAL;
arrX     : ARRAY [1..gBA_cChrct32_NumOfElem] OF LREAL;
arrY     : ARRAY [1..gBA_cChrct32_NumOfElem] OF LREAL;
iNumOfElem : INT;
bLmt     : BOOL;
```

lrX: Eingangswert der Kennlinie

arrX : Feld mit den X-Werten für die Stützpunkte

iNumOfElem: Anzahl der Stützstellen - diese Wert muss im Bereich von 2-32 liegen, ansonsten wird ein Fehler ausgegeben.

arrY : Feld mit den Y-Werten für die Stützpunkte

bLmt: Begrenzung des Ausgangswerts lrY

VAR_OUTPUT

```
lrY      : LREAL;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

lrY: Errechneter Ausgangswert der Kennlinie

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

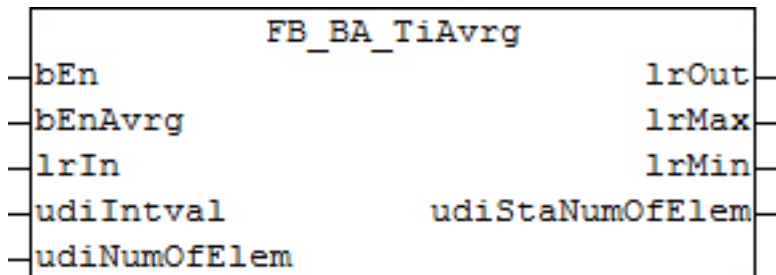
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [▶ 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.69 FB_BA_TiAavg

Mittelwertbildung über ein Zeitintervall



Funktionsbeschreibung

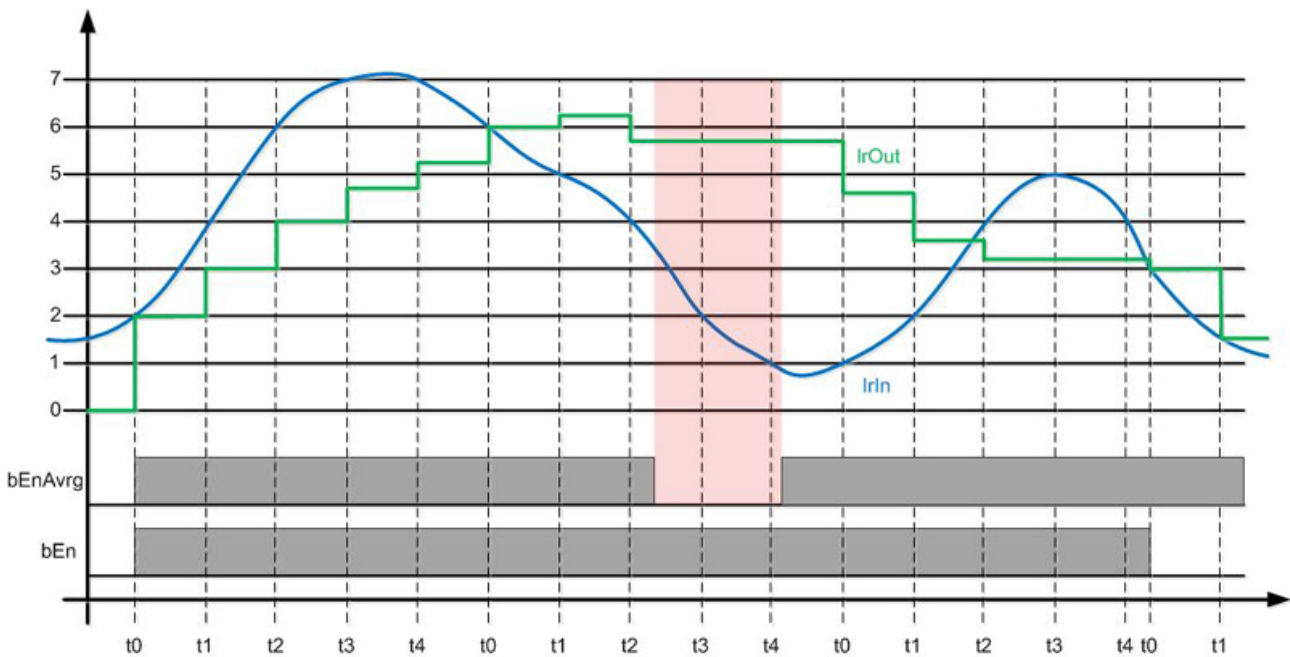
Der Funktionsbaustein FB_BA_TiAavg errechnet den zeitlich arithmetischen Mittelwert eines aufgenommenen Analogwerts. Dazu werden zeitlich diskret Werte in einen FIFO-Speicher hineingeschrieben. *udiIntval* gibt das Zeitintervall [s] an, in dem die Werte aufgenommen und in den FIFO geschrieben werden. Das Schreiben der Werte erfolgt jedoch nur wenn der Eingang *bEnAavg* = TRUE ist. Mit der Variablen *udiNumOfElem* wird die Größe des FIFO-Speichers bestimmt. Diese ist auf 1..256 begrenzt. Bei einem Eintrag außerhalb dieser Grenzen wird die Anzahl automatisch auf einen Defaultwert von 48 gesetzt. Die tatsächliche Anzahl wird zur Kontrolle am Ausgang *udiStaNumOfElem* ausgegeben. Der Funktionsbaustein kann zum Beispiel für die Berechnung einer mittleren Außentemperatur im stündlichen Tagesmittel verwendet werden. In dem Fall wäre *udiNumOfElem* = 24 und *udiIntval* = 3600 Sekunden. *bEn* ist die allgemeine Freigabe des Funktionsbausteins. Bei *bEn* = FALSE wird der FIFO-Speicher innerhalb des Funktionsbausteins komplett gelöscht und es werden keine Daten aufgezeichnet.

Beispiel:

udiNumOfElem = 5

Der rosa eingefärbte Bereich kennzeichnet die Phase, in der *bEnAavg* auf FALSE gesetzt ist. Es werden keine Werte in den FIFO geschrieben und der Ausgangswert *lrOut* gehalten.

	1. Zyklus		2. Zyklus		3. Zyklus		4. Zyklus	
	lrIn	lrOut	lrIn	lrOut	lrIn	lrOut	lrIn	lrOut
t0	2	2/1 = 2	6	(4+6+7+7+6)/5 = 6	1	(7+6+5+4+1)/5 = 4.6	3	lrIn = 3
t1	4	(2+4)/2 = 3	5	(6+7+7+6+5)/5 = 6.25	2	(6+5+4+1+2)/5 = 3.6	1.5	lrIn = 1.5
t2	6	(2+4+6)/3 = 4	4	(7+7+6+5+4)/5 = 5.8	4	(5+4+1+2+4)/5 = 3.2		
t3	7	(2+4+6+7)/4 = 4.75	2	(7+7+6+5+4)/5 = 5.8	5	(4+1+2+4+5)/5 = 3.2		
t4	7	(2+4+6+7+7)/5 = 5.2	1	(7+7+6+5+4)/5 = 5.8	4	(1+2+4+5+4)/5 = 3,2		



Ein-/Ausgänge

VAR_INPUT

```
bEn          : BOOL;
bEnAvrg     : BOOL;
lrIn        : LREAL;
udiIntVal   : UDINT;
udiNumOfElem : UDINT;
```

bEn: Freigabe des Bausteins

bEnAvrg: Freigabe der Mittelwertbildung, *bEnAvrg* = TRUE zeichnet auf, bei *bEnAvrg* = FALSE bleibt der errechnete Wert am Ausgang.

lrIn: Eingangswert zur Mittelwertbildung

udiIntVal: Zeitintervall [s] für das Schreiben neuer Werte in den FIFO

udiNumOfElem: Größe des FIFO-Wertespeichers. Eine Änderung setzt die vorangegangene Mittelwertbildung zurück. Die Anzahl der Werte ist auf 1..256 automatisch begrenzt.

VAR_OUTPUT

```
lrOut       : LREAL;
lrMax       : LREAL;
lrMin       : LREAL;
udiStaNumOfElem : UDINT;
```

lrOut: Mittelwert

lrMax: größter Wert im FIFO-Speicher

lrMin: kleinster Wert im FIFO-Speicher

udiStaNumOfElem: Anzahl der Einträge im Speicher

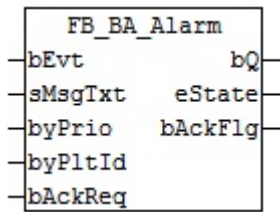
Voraussetzungen

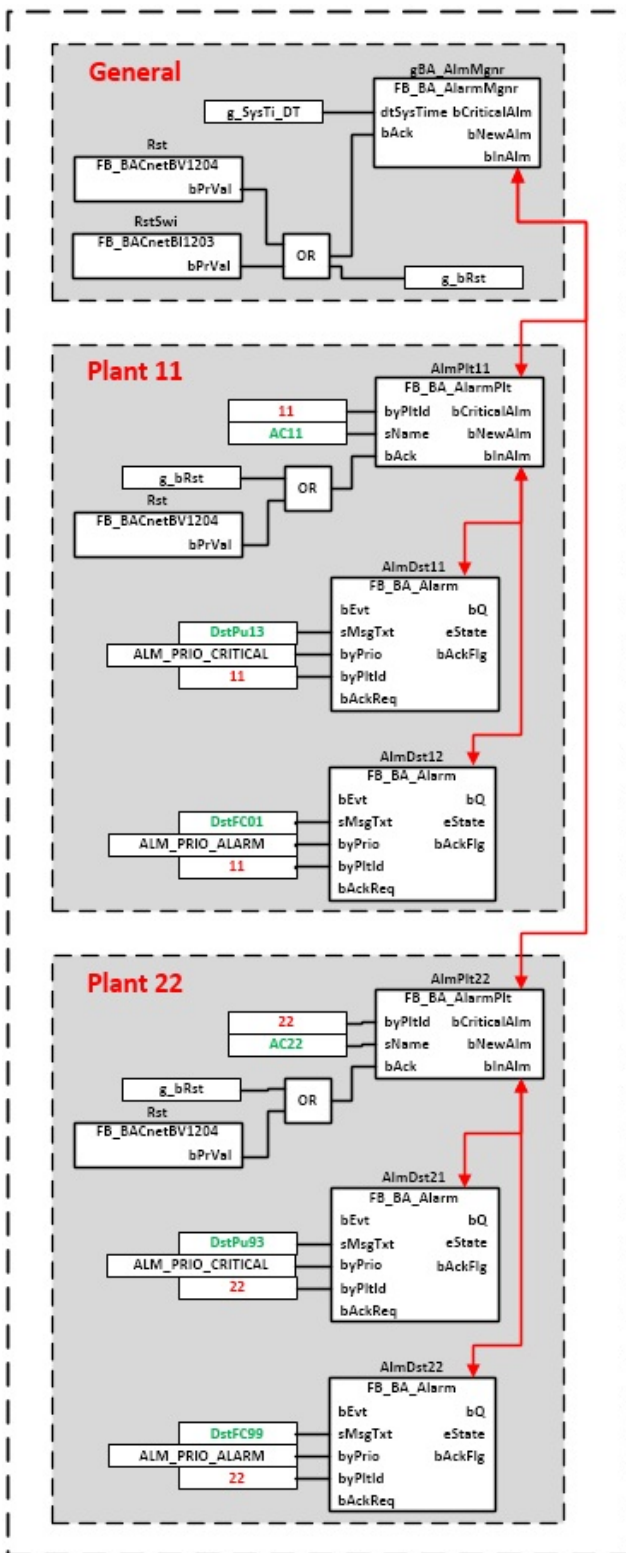
Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.70 FB_BA_Alarm

Alarmausgabebaustein mit zuschaltbarem Alarmspeicher und interner Quittierung.

Der Impuls der Quittierung wird im [FB_BA_AlarmPlt \[► 186\]](#) erzeugt und intern an die Alarmsammelbausteine einer Anlage durch gereicht.





Ein-/Ausgänge

VAR_INPUT

```

bEvt      : BOOL;
sMsgTxt   : STRING;
byPrio    : BYTE;
byPltId   : BYTE;
bAckReq   : BOOL;
    
```

bEvt: Eingang zum Erfassen des Alarms

sMsgTxt: Eingabefeld für Meldetext (konfigurierbar über Project-Builder oder Excel-Import)

byPrio: Bestimmt die Aktion, die beim Eingehen des Alarms bezogen auf die Steuerung der jeweiligen Anlage ausgelöst werden soll.

byPrio = 0 = keine Meldung = ALM_PRIO_EMPTY

byPrio = 1 = diverse Meldungen = ALM_PRIO_NOTE

byPrio = 2 = Warnung = ALM_PRIO_WARNING

byPrio = 3 = Alarm ohne Abschaltung = ALM_PRIO_ALARM

byPrio = 4 = Alarm mit Abschaltung = ALM_PRIO_CRITICAL, setzt den Ausgang *bCriticalAlm* vom Funktionsbaustein [FB_BA_AlarmPlt](#) [▶ 186].

byPrtId: Nummer der zugehörigen Anlage. **Wichtig: Alle Templates einer Anlage müssen der gleichen Anlagennummer in der PLC zugewiesen sein!**

bAckReq: TRUE = quittierpflichtig. Ist der Alarm quittierpflichtig erlischt dieser in der Alarmliste nur nach einer Alarmquittierung des Anlagenbetreibers.

VAR_OUTPUT

```
bQ           : BOOL;
eState      : E_BA_AlmSta;
bAckFlg    : BOOL;
```

bQ: Alarmausgang

eState: [Enumerator](#) [▶ 329] Status der Alarmmeldungen

bAckFlg: Bei einer ausgelösten Quittierung steht dieser Ausgang für einen Zyklus an. Der Impuls kann zum Quittieren von Störungen genutzt werden. Der Impuls der Quittierung wird im [FB_BA_AlarmPlt](#) [▶ 186] erzeugt und intern an die Alarmsammelbausteine einer Anlage durchgereicht.

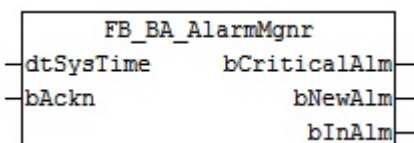
Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

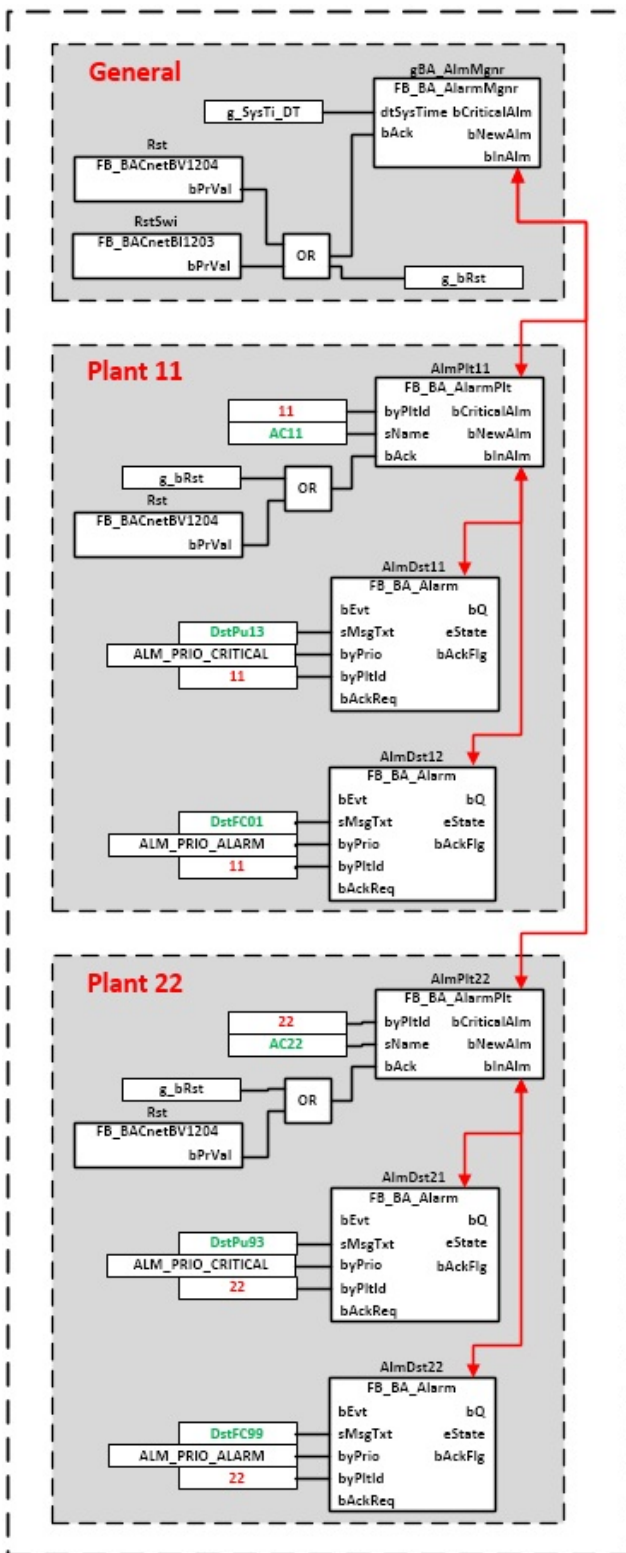
8.2.71 FB_BA_AlarmMgnr

Der Funktionsbaustein Alarmmanager sammelt die Alarmergebnisse sämtlicher Anlagen ein und bündelt diese zu einem Sammelalarm.

Er leitet intern die Systemzeit an die Anlagen und den dazu gehörigen Alarmausgabebausteinen weiter, damit im Alarmfall ein Zeitstempel ausgegeben werden kann.



Der Alarmmanager ist das Bindeglied zwischen sämtlichen Alarmen einer Steuerung und deren Darstellung in der Target Visualisierung. Die Anzeige der Alarmergebnisse in der Target Visualisierung wird automatisch mit dem Template BAC_GenAlm_01 angelegt.



Ein-/Ausgänge

VAR_INPUT

```
dtSysTime: DT;
sName     : STRING;
bAckn     : BOOL;
```

dtSysTime: Systemzeit welche im Alarmfall als Zeitstempel genutzt wird. Diese wird intern an die Anlagen **FB_BA_AlarmPit** und den dazu gehörigen Alarmausgabebausteinen **FB_BA_Alarm** weiter geleitet.

bAckn: Eingang für die zentrale Alarmquittierung sämtlicher Anlagen. Diese Quittierung wird intern an die Alarmausgabebausteine **FB_BA_Alarm** durch gereicht.

VAR_OUTPUT

```
bCriticalAlm : BOOL;
bNewAlm      : BOOL;
bInAlm       : BOOL;
```

bCriticalAlm: Kritischer Alarm innerhalb einer Anlage. Dieser wird in den Templates für das Abschalten einer Anlage genutzt. Dazu muss in dem Alarmausgabebausteine **FB_BA_Alarm** der Eingang *byPrio* den Wert 4 haben.

bNewAlm: Zeigt die Neuwertmeldung eines Alarms innerhalb einer Anlage an.

bInAlm: Zeigt an, dass ein Alarm innerhalb einer Anlage ansteht.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.72 FB_BA_AlarmPlt

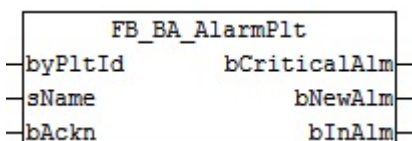
Der Funktionsbaustein sammelt die Alarmer einer Anlage ein und bündelt diese zu einem Sammelalarm. Erzeugt werden die Alarmer innerhalb der anlagenzugehörigen Templates durch den Baustein FB_BA_Alarm [► 182].

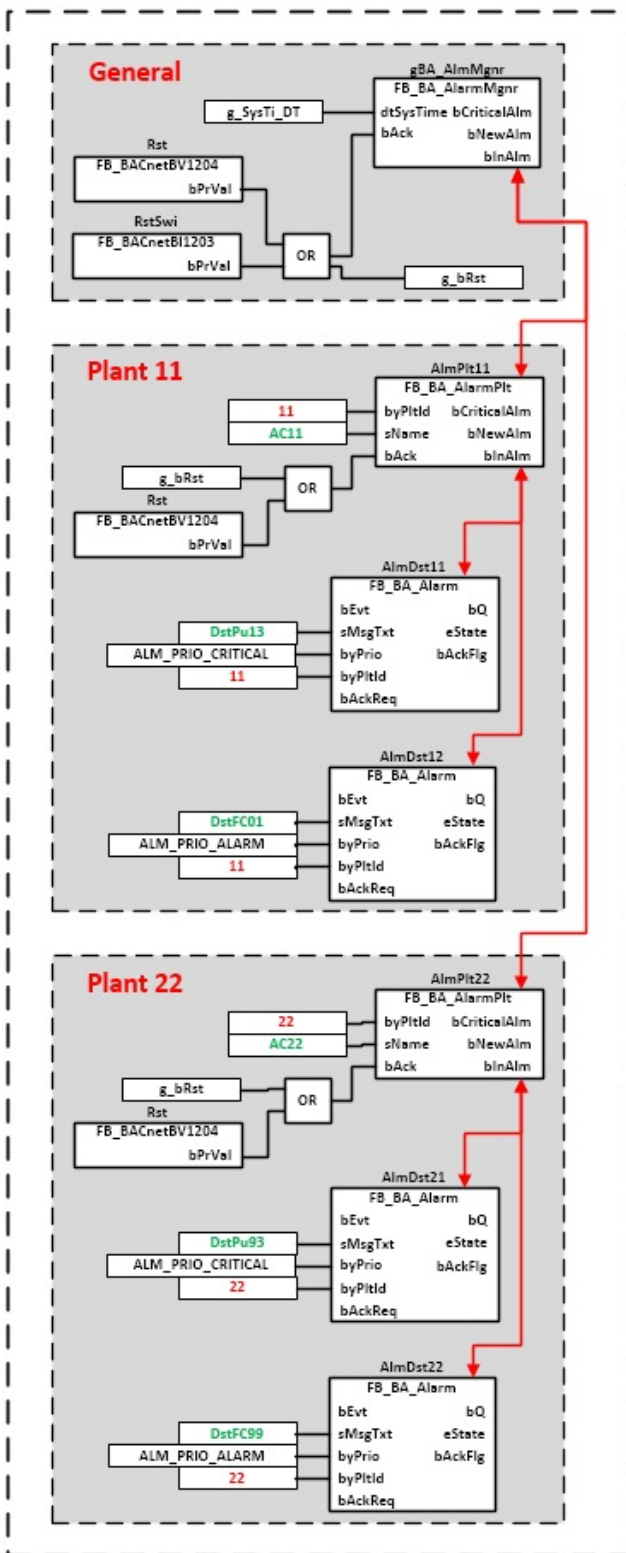
Die Zuordnung der Alarmer zu einer Anlage erfolgt über die Anlagennummer.

Die Priorisierung bzw. Wertigkeit der Alarmer erfolgt im Alarmausgabebaustein **FB_BA_Alarm** durch den Eingang *byPrio*.

Wichtig ist, dass der Alarmsammler **FB_BA_AlarmPlt und die Alarmausgabebausteine FB_BA_Alarm [► 182] innerhalb der Templates einer Anlage auf die gleiche Anlagennummer eingestellt sind!**

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.





Ein-/Ausgänge

VAR_INPUT

byPltId : BYTE;
 sName : STRING;
 bAck : BOOL;

byPltId: Nummer der zugehörigen Anlage. **Wichtig: Alle Templates einer Anlage müssen der gleichen Anlagennummer in der PLC zugewiesen sein!**

sName: Eingabefeld für die Anlagenbezeichnung (konfigurierbar über Project-Builder oder Excel-Import)

bAck: Eingang für die Alarmquittierung der Anlage. Diese Quittierung wird intern an die Alarmausgabebausteine **FB_BA_Alarm** durch gereicht.

VAR_OUTPUT

```
bCriticalAlm : BOOL;
bNewAlm      : BOOL;
bInAlm       : BOOL;
```

bCriticalAlm: Kritischer Alarm. Dieser wird in den Templates für das Abschalten einer Anlage genutzt. Dazu muss in dem Alarmausgabebausteine **FB_BA_Alarm** der Eingang *byPrio* den Wert 4 haben.

bNewAlm: Zeigt die Neuwertmeldung eines Alarms an

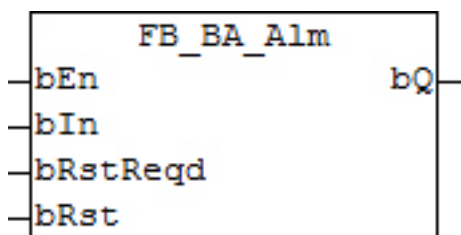
bInAlm: Zeigt an, dass ein Alarm ansteht

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

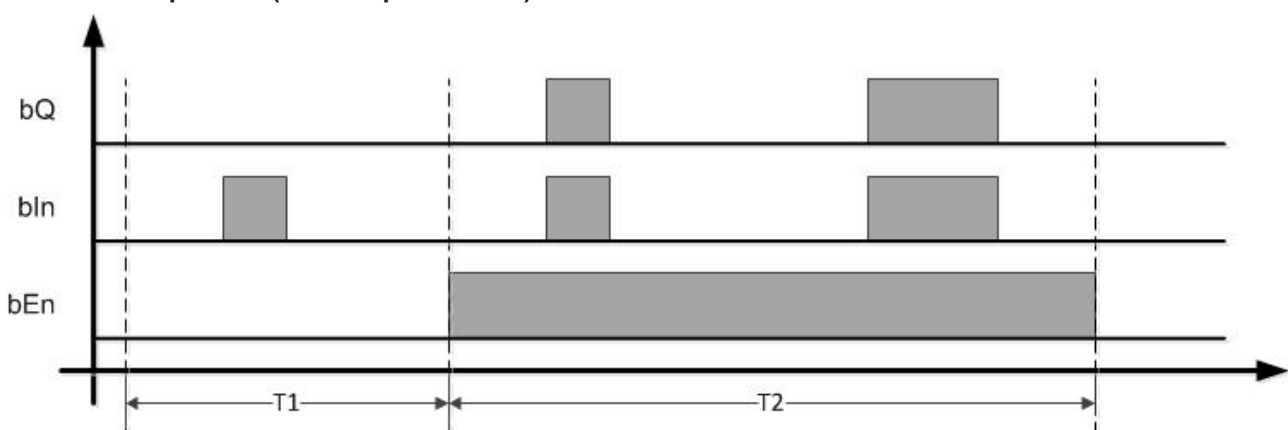
8.2.73 FB_BA_Alarm

Alarmbaustein mit zuschaltbarem Alarmspeicher und Quittierung



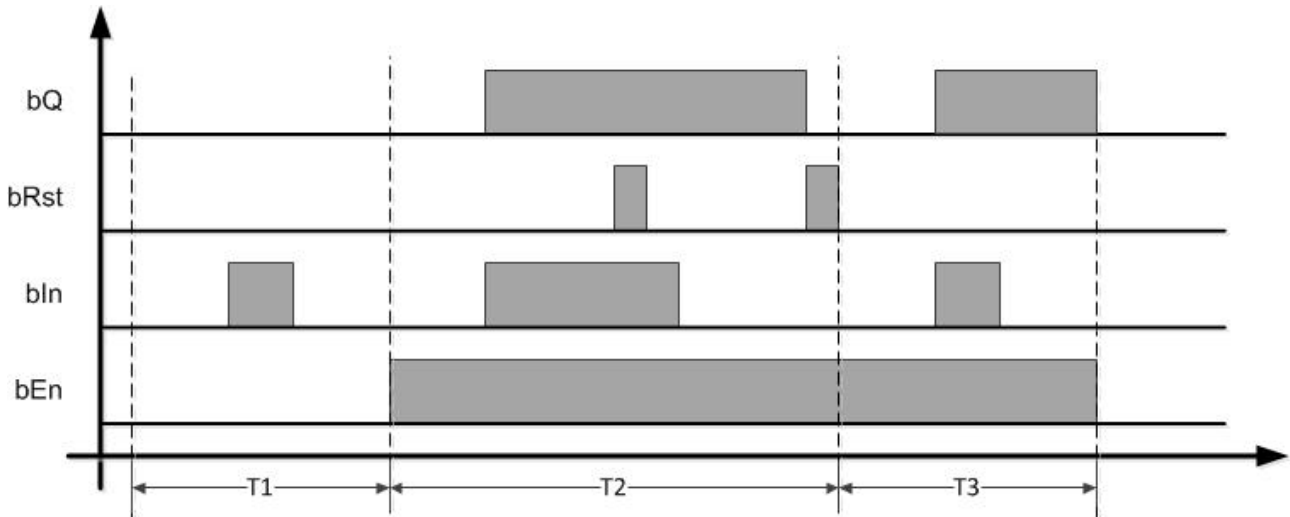
Funktionsbeschreibung

ohne Alarmspeicher ($bRstReqd = FALSE$)



- T1: Der Baustein ist nicht aktiviert ($bEn=FALSE$). Eine Meldung am Eingang *bIn* hat keinerlei Auswirkung auf den Alarmausgang *bQ*.
- T2: Der Baustein ist aktiviert ($bEn=TRUE$). Der Alarmausgang *bQ* bleibt immer so lange gesetzt, wie die Meldung am Eingang *bIn* ansteht.

mit Alarmspeicher (bRstReqd = TRUE)



- T1: Der Baustein ist nicht aktiviert ($bEn=FALSE$). Eine Meldung am Eingang bIn hat keinerlei Auswirkung auf den Alarmausgang bQ .
- T2: Der Baustein ist aktiviert ($bEn=TRUE$). Eine steigende Flanke am Meldeeingang bIn setzt den Alarmausgang bQ . Dieser kann durch eine steigende Flanke am Reset-Eingang $bRst$ zurück gesetzt werden, jedoch nur, wenn die Meldung an bIn nicht mehr ansteht.
- T3: Alarmmeldungen an bQ stehen grundsätzlich nur so lange an, wie der Baustein an bEn freigegeben ist.

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
bIn      : BOOL;
bRstReqd : BOOL;
bRst     : BOOL;
```

bEn: Freigabe des Funktionsbausteins

bIn: Meldeeingang

bRstReqd: Aktivierung Alarmspeicher (Quittierpflichtigkeit)

bRst: Reset Alarm

VAR_OUTPUT

```
bQ      : BOOL;
```

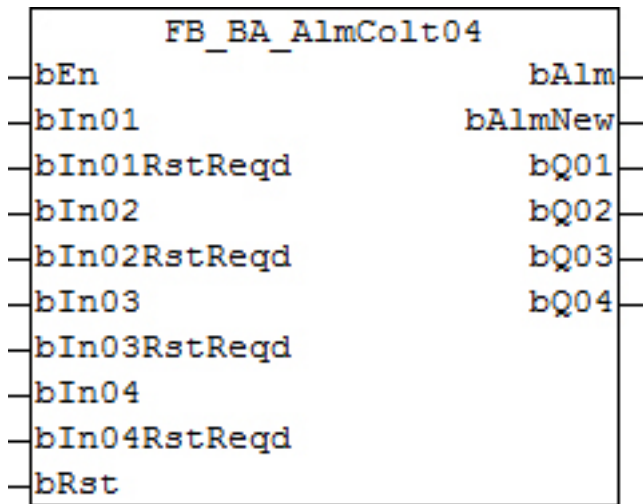
bQ: Alarmausgang

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.74 FB_BA_AlmColt04

Alarmsammelbaustein, 4 Alarme



Funktionsbeschreibung

Dieser Baustein stellt eine Alarmsammlung von bis zu 4 Einzel-Alarmen dar. Jeder dieser Einzel-Alarme *bIn01..bIn04* wird bausteinintern mit einem [FB_BA_Alm \[► 188\]](#) behandelt und hat entsprechend die Wahl zur Alarmspeicherung an den Eingängen *bIn01RstReqd..bIn04RstReqd*. Die Status der 4 Einzel-Alarme werden jeweils an den Ausgängen *bQ01* bis *bQ04* ausgegeben.

Der Ausgang des Sammelalarms *bAlm* ergibt sich aus einer ODER-Verknüpfung der einzelnen Alarme und kann dementsprechend auch unquittiert gehen, wenn kein Einzel-Alarm mit Quittierpflichtigkeit parametrierung wurde. Ist hingegen über die Eingänge *bIn01RstReqd..bIn04RstReqd* eine Alarmspeicherung angewählt, so können die Einzelalarms sowie der Sammelalarm global über den Eingang *bRst* zurück gesetzt werden. Wie in [FB_BA_Alm \[► 188\]](#) bereits erwähnt, kann ein Alarm natürlich nur dann zurück gesetzt werden, wenn seine Auslösemeldung an *bIn* nicht mehr existiert.

Der Ausgang *bAlmNew* wird immer dann gesetzt, wenn an den Eingängen *bIn01..bIn04* ein erneutes Meldeereignis auftritt. Er ist in jedem Fall - unabhängig von den Parametrierungen an *bIn01RstReqd..bIn04RstReqd* - speichernd und wird nur durch eine steigende Flanke an *bRst* bis zum nächsten Meldeereignis zurück gesetzt.

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
bIn01   : BOOL;
bIn01RstReqd : BOOL;
bIn02   : BOOL;
bIn02RstReqd : BOOL;
bIn03   : BOOL;
bIn03RstReqd : BOOL;
bIn04   : BOOL;
bIn04RstReqd : BOOL;
```

bEn: Freigabe des Funktionsbausteins

bIn01: Alarmeingang 1

bIn01RstReqd: Quittierpflichtigkeit Alarm am Eingang 1

bIn02: Alarmeingang 2

bIn02RstReqd: Quittierpflichtigkeit Alarm am Eingang 2

bIn03: Alarmeingang 3

bIn03RstReqd: Quittierpflichtigkeit Alarm am Eingang 3

bIn04: Alarmeingang 4

bIn04RstReqd: Quittierpflichtigkeit Alarm am Eingang 4

bRst: Rücksetzen aller als quittierpflichtig parametrierung Alarme sowie der Sammelalarms *bAlm* und *bAlmNew*.

VAR_OUTPUT

```
bAlm      : BOOL;
bAlmNew   : BOOL;
bQ01      : BOOL;
bQ02      : BOOL;
bQ03      : BOOL;
bQ04      : BOOL;
```

bAlm: Sammelalarm - ODER-Verknüpfung der Einzel-Alarme an den Eingängen

bAlmNew: Merker neues Alarmereignis

bQ01: Alarmausgang 1

bQ02: Alarmausgang 2

bQ03: Alarmausgang 3

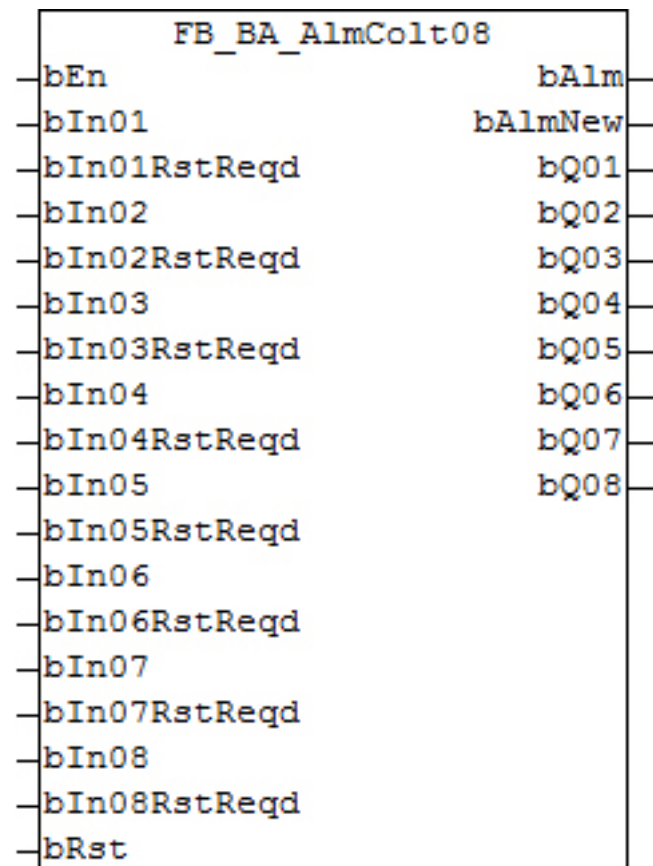
bQ04: Alarmausgang 4

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.75 FB_BA_AlmColt08

Alarmsammelbaustein, 8 Alarme



Funktionsbeschreibung

Dieser Baustein stellt eine Alarmsammlung von bis zu 8 Einzel-Alarmen dar. Jeder dieser Einzel-Alarme *bln01..bln08* wird bausteinintern mit einem `FB_BA_Alm` [► 188] behandelt und hat entsprechend die Wahl zur Alarmspeicherung an den Eingängen *bln01RstReqd..bln08RstReqd*. Die Status der 8 Einzel-Alarme werden jeweils an den Ausgängen *bQ01* bis *bQ08* ausgegeben.

Der Ausgang des Sammelalarms *bAlm* ergibt sich aus einer ODER-Verknüpfung der einzelnen Alarme und kann dementsprechend auch unquittiert gehen, wenn kein Einzel-Alarm mit Quittierpflichtigkeit parametrierung wurde. Ist hingegen über die Eingänge *bln01RstReqd..bln08RstReqd* eine Alarmspeicherung angewählt, so können die Einzelalarms sowie der Sammelalarm global über den Eingang *bRst* zurück gesetzt werden. Wie in `FB_BA_Alm` [► 188] bereits erwähnt, kann ein Alarm natürlich nur dann zurück gesetzt werden, wenn seine Auslösemeldung an *bln* nicht mehr existiert.

Der Ausgang *bAlmNew* wird immer dann gesetzt, wenn an den Eingängen *bln01..bln08* ein erneutes Meldeereignis auftritt. Er ist in jedem Fall - unabhängig von den Parametrierungen an *bln01RstReqd..bln08RstReqd* - speichernd und wird nur durch eine steigende Flanke an *bRst* bis zum nächsten Meldeereignis zurück gesetzt.

Ein-Ausgänge

VAR_INPUT

```
bEn          : BOOL;
bIn01        : BOOL;
bIn01RstReqd : BOOL;
bIn02        : BOOL;
bIn02RstReqd : BOOL;
bIn03        : BOOL;
bIn03RstReqd : BOOL;
bIn04        : BOOL;
bIn04RstReqd : BOOL;
bIn05        : BOOL;
bIn05RstReqd : BOOL;
bIn06        : BOOL;
bIn06RstReqd : BOOL;
bIn07        : BOOL;
bIn07RstReqd : BOOL;
bIn08        : BOOL;
bIn08RstReqd : BOOL;
```

bEn: Freigabe des Funktionsbausteins

bln01: Alarmeingang 1

bln01RstReqd: Quittierpflichtigkeit Alarm am Eingang 1

bln02: Alarmeingang 2

bln02RstReqd: Quittierpflichtigkeit Alarm am Eingang 2

bln03: Alarmeingang 3

bln03RstReqd: Quittierpflichtigkeit Alarm am Eingang 3

bln04: Alarmeingang 4

bln04RstReqd: Quittierpflichtigkeit Alarm am Eingang 4

bln05: Alarmeingang 5

bln05RstReqd: Quittierpflichtigkeit Alarm am Eingang 5

bln06: Alarmeingang 6

bln06RstReqd: Quittierpflichtigkeit Alarm am Eingang 6

bln07: Alarmeingang 7

bln07RstReqd: Quittierpflichtigkeit Alarm am Eingang 7

bln08: Alarmeingang 8

bln08RstReqd: Quittierpflichtigkeit Alarm am Eingang 8

bRst: Rücksetzen aller als quittierpflichtig parametrierung Alarme sowie der Sammelalarms *bAlm* und *bAlmNew*.

VAR_OUTPUT

```

bAlm      : BOOL;
bAlmNew   : BOOL;
bQ01      : BOOL;
bQ02      : BOOL;
bQ03      : BOOL;
bQ04      : BOOL;
bQ05      : BOOL;
bQ06      : BOOL;
bQ07      : BOOL;
bQ08      : BOOL;
    
```

bAlm: Sammelalarm - ODER-Verknüpfung der Einzel-Alarme an den Eingängen

bAlmNew: Merker neues Alarmereignis

bQ01: Alarmausgang 1

bQ02: Alarmausgang 2

bQ03: Alarmausgang 3

bQ04: Alarmausgang 4

bQ05: Alarmausgang 5

bQ06: Alarmausgang 6

bQ07: Alarmausgang 7

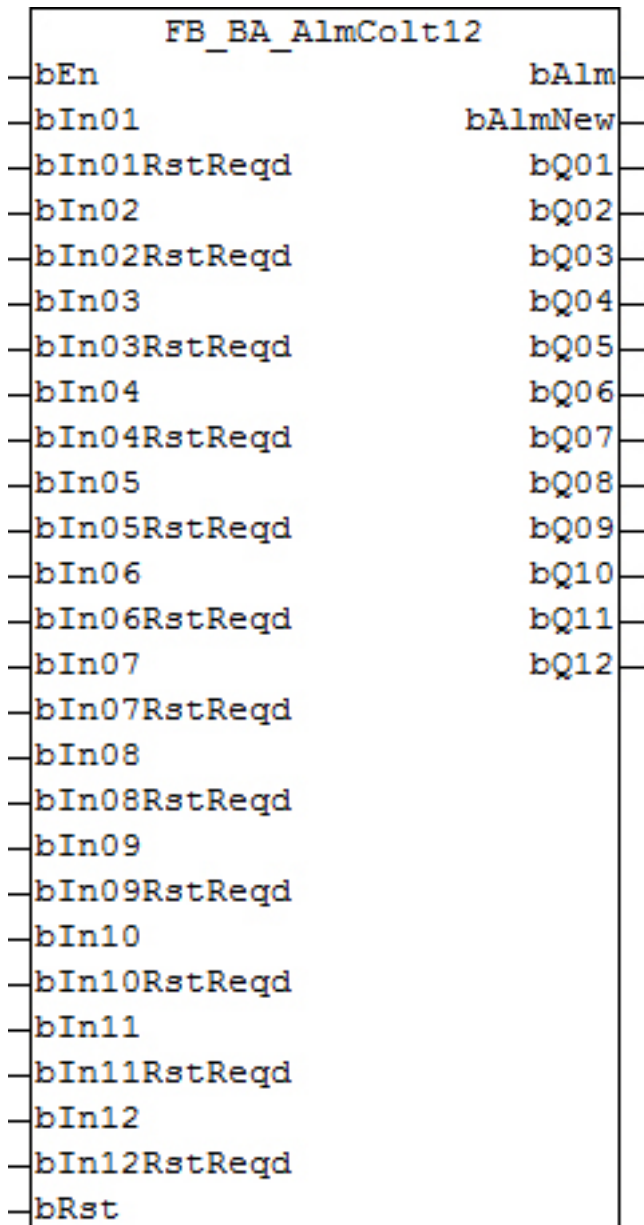
bQ08: Alarmausgang 8

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.76 FB_BA_AlmColt12

Alarmsammelbaustein, 12 Alarme



Funktionsbeschreibung

Dieser Baustein stellt eine Alarmsammlung von bis zu 12 Einzel-Alarmen dar. Jeder dieser Einzel-Alarme *bIn01..bIn12* wird bausteinintern mit einem [FB_BA_Alm](#) [► 188] behandelt und hat entsprechend die Wahl zur Alarmspeicherung an den Eingängen *bIn01RstReqd..bIn12RstReqd*. Die Status der 12 Einzel-Alarme werden jeweils an den Ausgängen *bQ01* bis *bQ12* ausgegeben.

Der Ausgang des Sammelalarms *bAlm* ergibt sich aus einer ODER-Verknüpfung der einzelnen Alarme und kann dementsprechend auch unquittiert gehen, wenn kein Einzel-Alarm mit Quittierpflichtigkeit parametrierung wurde. Ist hingegen über die Eingänge *bIn01RstReqd..bIn12RstReqd* eine Alarmspeicherung angewählt, so können die Einzelalarms sowie der Sammelalarm global über den Eingang *bRst* zurück gesetzt werden. Wie in [FB_BA_Alm](#) [► 188] bereits erwähnt, kann ein Alarm natürlich nur dann zurück gesetzt werden, wenn seine Auslösemeldung an *bIn* nicht mehr existiert.

Der Ausgang *bAlmNew* wird immer dann gesetzt, wenn an den Eingängen *bIn01..bIn12* ein erneutes Meldeereignis auftritt. Er ist in jedem Fall - unabhängig von den Parametrierungen an *bIn01RstReqd..bIn12RstReqd* - speichernd und wird nur durch eine steigende Flanke an *bRst* bis zum nächsten Meldeereignis zurück gesetzt.

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;  
bIn01   : BOOL;  
bIn01RstReqd : BOOL;  
bIn02   : BOOL;  
bIn02RstReqd : BOOL;  
bIn03   : BOOL;  
bIn03RstReqd : BOOL;  
bIn04   : BOOL;  
bIn04RstReqd : BOOL;  
bIn05   : BOOL;  
bIn05RstReqd : BOOL;  
bIn06   : BOOL;  
bIn06RstReqd : BOOL;  
bIn07   : BOOL;  
bIn07RstReqd : BOOL;  
bIn08   : BOOL;  
bIn08RstReqd : BOOL;  
bIn09   : BOOL;  
bIn09RstReqd : BOOL;  
bIn10   : BOOL;  
bIn10RstReqd : BOOL;  
bIn11   : BOOL;  
bIn11RstReqd : BOOL;  
bIn12   : BOOL;  
bIn12RstReqd : BOOL;
```

bEn: Freigabe des Funktionsbausteins

bIn01: Alarmeingang 1

bIn01RstReqd: Quittierpflichtigkeit Alarm am Eingang 1

bIn02: Alarmeingang 2

bIn02RstReqd: Quittierpflichtigkeit Alarm am Eingang 2

bIn03: Alarmeingang 3

bIn03RstReqd: Quittierpflichtigkeit Alarm am Eingang 3

bIn04: Alarmeingang 4

bIn04RstReqd: Quittierpflichtigkeit Alarm am Eingang 4

bIn05: Alarmeingang 5

bIn05RstReqd: Quittierpflichtigkeit Alarm am Eingang 5

bIn06: Alarmeingang 6

bIn06RstReqd: Quittierpflichtigkeit Alarm am Eingang 6

bIn07: Alarmeingang 7

bIn07RstReqd: Quittierpflichtigkeit Alarm am Eingang 7

bIn08: Alarmeingang 8

bIn08RstReqd: Quittierpflichtigkeit Alarm am Eingang 8

bIn09: Alarmeingang 9

bIn09RstReqd: Quittierpflichtigkeit Alarm am Eingang 9

bIn10: Alarmeingang 10

bIn10RstReqd: Quittierpflichtigkeit Alarm am Eingang 10

bIn11: Alarmeingang 11

bIn11RstReqd: Quittierpflichtigkeit Alarm am Eingang 11

bIn12: Alarmeingang 12

bIn12RstReqd: Quittierpflichtigkeit Alarm am Eingang 12

bRst: Rücksetzen aller als quittierpflichtig parametrisierten Alarme sowie der Sammelalarme *bAlm* und *bAlmNew*.

VAR_OUTPUT

```

bAlm      : BOOL;
bAlmNew   : BOOL;
bQ01      : BOOL;
bQ02      : BOOL;
bQ03      : BOOL;
bQ04      : BOOL;
bQ05      : BOOL;
bQ06      : BOOL;
bQ07      : BOOL;
bQ08      : BOOL;
bQ09      : BOOL;
bQ10      : BOOL;
bQ11      : BOOL;
bQ12      : BOOL;

```

bAlm: Sammelalarm - ODER-Verknüpfung der Einzel-Alarme an den Eingängen

bAlmNew: Merker neues Alarmereignis

bQ01: Alarmausgang 1

bQ02: Alarmausgang 2

bQ03: Alarmausgang 3

bQ04: Alarmausgang 4

bQ05: Alarmausgang 5

bQ06: Alarmausgang 6

bQ07: Alarmausgang 7

bQ08: Alarmausgang 8

bQ09: Alarmausgang 9

bQ10: Alarmausgang 10

bQ11: Alarmausgang 11

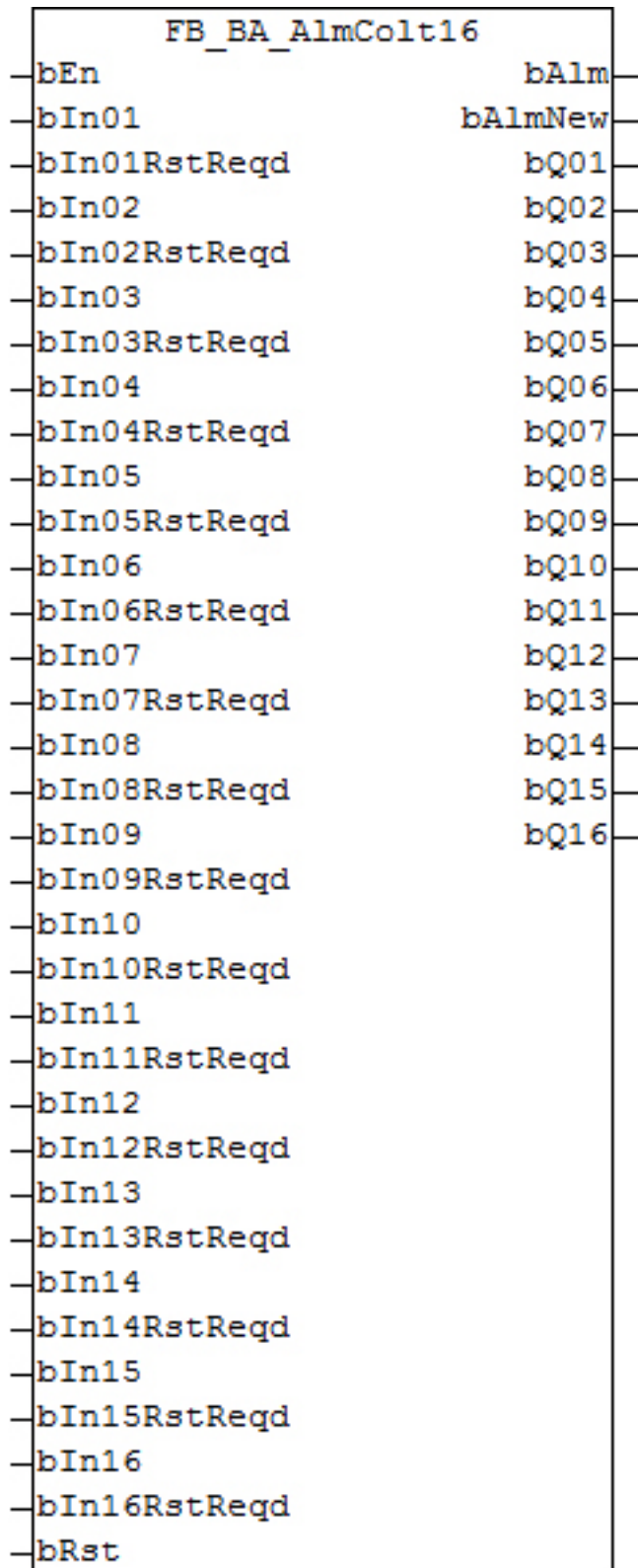
bQ12: Alarmausgang 12

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.77 FB_BA_AlmColt16

Alarmsammelbaustein, 16 Alarme



Funktionsbeschreibung

Dieser Baustein stellt eine Alarmsammlung von bis zu 16 Einzel-Alarmen dar. Jeder dieser Einzel-Alarme *bIn01..bIn16* wird bausteinintern mit einem `FB_BA_Alm` [► 188] behandelt und hat entsprechend die Wahl zur Alarmspeicherung an den Eingängen *bIn01RstReqd..bIn16RstReqd*. Die Status der 16 Einzel-Alarme werden jeweils an den Ausgängen *bQ01* bis *bQ16* ausgegeben.

Der Ausgang des Sammelalarms *bAlm* ergibt sich aus einer ODER-Verknüpfung der einzelnen Alarme und kann dementsprechend auch unquittiert gehen, wenn kein Einzel-Alarm mit Quittierpflichtigkeit parametrierung wurde. Ist hingegen über die Eingänge *bln01RstReqd..bln16RstReqd* eine Alarmspeicherung angewählt, so können die Einzelalarme sowie der Sammelalarm global über den Eingang *bRst* zurück gesetzt werden. Wie in [FB BA Alm \[► 188\]](#) bereits erwähnt, kann ein Alarm natürlich nur dann zurück gesetzt werden, wenn seine Auslösemeldung an *bln* nicht mehr existiert.

Der Ausgang *bAlmNew* wird immer dann gesetzt, wenn an den Eingängen *bln01..bln16* ein erneutes Meldeereignis auftritt. Er ist in jedem Fall - unabhängig von den Parametrierungen an *bln01RstReqd..bln16RstReqd* - speichernd und wird nur durch eine steigende Flanke an *bRst* bis zum nächsten Meldeereignis zurück gesetzt.

Ein-/Ausgänge

VAR_INPUT

```

bEn          : BOOL;
bIn01        : BOOL;
bIn01RstReqd : BOOL;
bIn02        : BOOL;
bIn02RstReqd : BOOL;
bIn03        : BOOL;
bIn03RstReqd : BOOL;
bIn04        : BOOL;
bIn04RstReqd : BOOL;
bIn05        : BOOL;
bIn05RstReqd : BOOL;
bIn06        : BOOL;
bIn06RstReqd : BOOL;
bIn07        : BOOL;
bIn07RstReqd : BOOL;
bIn08        : BOOL;
bIn08RstReqd : BOOL;
bIn09        : BOOL;
bIn09RstReqd : BOOL;
bIn10        : BOOL;
bIn10RstReqd : BOOL;
bIn11        : BOOL;
bIn11RstReqd : BOOL;
bIn12        : BOOL;
bIn12RstReqd : BOOL;
bIn13        : BOOL;
bIn13RstReqd : BOOL;
bIn14        : BOOL;
bIn14RstReqd : BOOL;
bIn15        : BOOL;
bIn15RstReqd : BOOL;
bIn16        : BOOL;
bIn16RstReqd : BOOL;
bRst         : BOOL;

```

bEn: Freigabe des Funktionsbausteins

bln01: Alarmeingang 1

bln01RstReqd: Quittierpflichtigkeit Alarm am Eingang 1

bln02: Alarmeingang 2

bln02RstReqd: Quittierpflichtigkeit Alarm am Eingang 2

bln03: Alarmeingang 3

bln03RstReqd: Quittierpflichtigkeit Alarm am Eingang 3

bln04: Alarmeingang 4

bln04RstReqd: Quittierpflichtigkeit Alarm am Eingang 4

bln05: Alarmeingang 5

bln05RstReqd: Quittierpflichtigkeit Alarm am Eingang 5

bln06: Alarmeingang 6

bln06RstReqd: Quittierpflichtigkeit Alarm am Eingang 6

bln07: Alarmeingang 7

bln07RstReqd: Quittierpflichtigkeit Alarm am Eingang 7

bln08: Alarmeingang 8

bln08RstReqd: Quittierpflichtigkeit Alarm am Eingang 8

bln09: Alarmeingang 9

bln09RstReqd: Quittierpflichtigkeit Alarm am Eingang 9

bln10: Alarmeingang 10

bln10RstReqd: Quittierpflichtigkeit Alarm am Eingang 10

bln11: Alarmeingang 11

bln11RstReqd: Quittierpflichtigkeit Alarm am Eingang 11

bln12: Alarmeingang 12

bln12RstReqd: Quittierpflichtigkeit Alarm am Eingang 12

bln13: Alarmeingang 13

bln13RstReqd: Quittierpflichtigkeit Alarm am Eingang 13

bln14: Alarmeingang 14

bln14RstReqd: Quittierpflichtigkeit Alarm am Eingang 14

bln15: Alarmeingang 15

bln15RstReqd: Quittierpflichtigkeit Alarm am Eingang 15

bln16: Alarmeingang 16

bln16RstReqd: Quittierpflichtigkeit Alarm am Eingang 16

bRst: Rücksetzen aller als quittierpflichtig parametrisierten Alarme sowie der Sammelalarme *bAlm* und *bAlmNew*.

VAR_OUTPUT

```
bAlm      : BOOL;  
bAlmNew   : BOOL;  
bQ01     : BOOL;  
bQ02     : BOOL;  
bQ03     : BOOL;  
bQ04     : BOOL;  
bQ05     : BOOL;  
bQ06     : BOOL;  
bQ07     : BOOL;  
bQ08     : BOOL;  
bQ09     : BOOL;  
bQ10     : BOOL;  
bQ11     : BOOL;  
bQ12     : BOOL;  
bQ13     : BOOL;  
bQ14     : BOOL;  
bQ15     : BOOL;  
bQ16     : BOOL;
```

bAlm: Sammelalarm - ODER-Verknüpfung der Einzel-Alarme an den Eingängen

bAlmNew: Merker neues Alarmereignis

bQ01: Alarmausgang 1

bQ02: Alarmausgang 2

bQ03: Alarmausgang 3

bQ04: Alarmausgang 4

bQ05: Alarmausgang 5

bQ06: Alarmausgang 6

bQ07: Alarmausgang 7

bQ08: Alarmausgang 8

bQ09: Alarmausgang 9

bQ10: Alarmausgang 10

bQ11: Alarmausgang 11

bQ12: Alarmausgang 12

bQ13: Alarmausgang 13

bQ14: Alarmausgang 14

bQ15: Alarmausgang 15

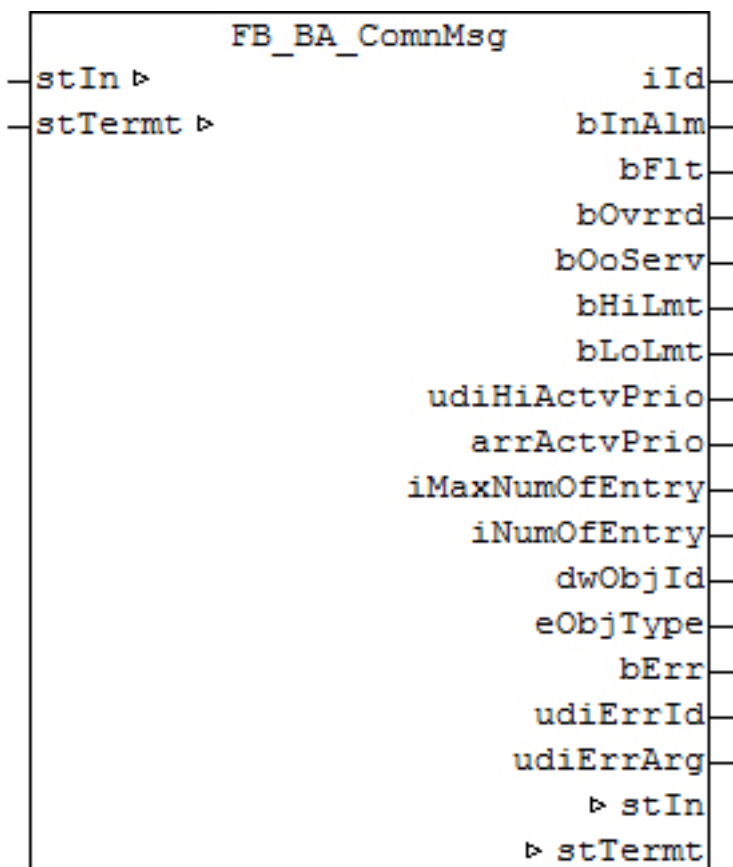
bQ16: Alarmausgang 16

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

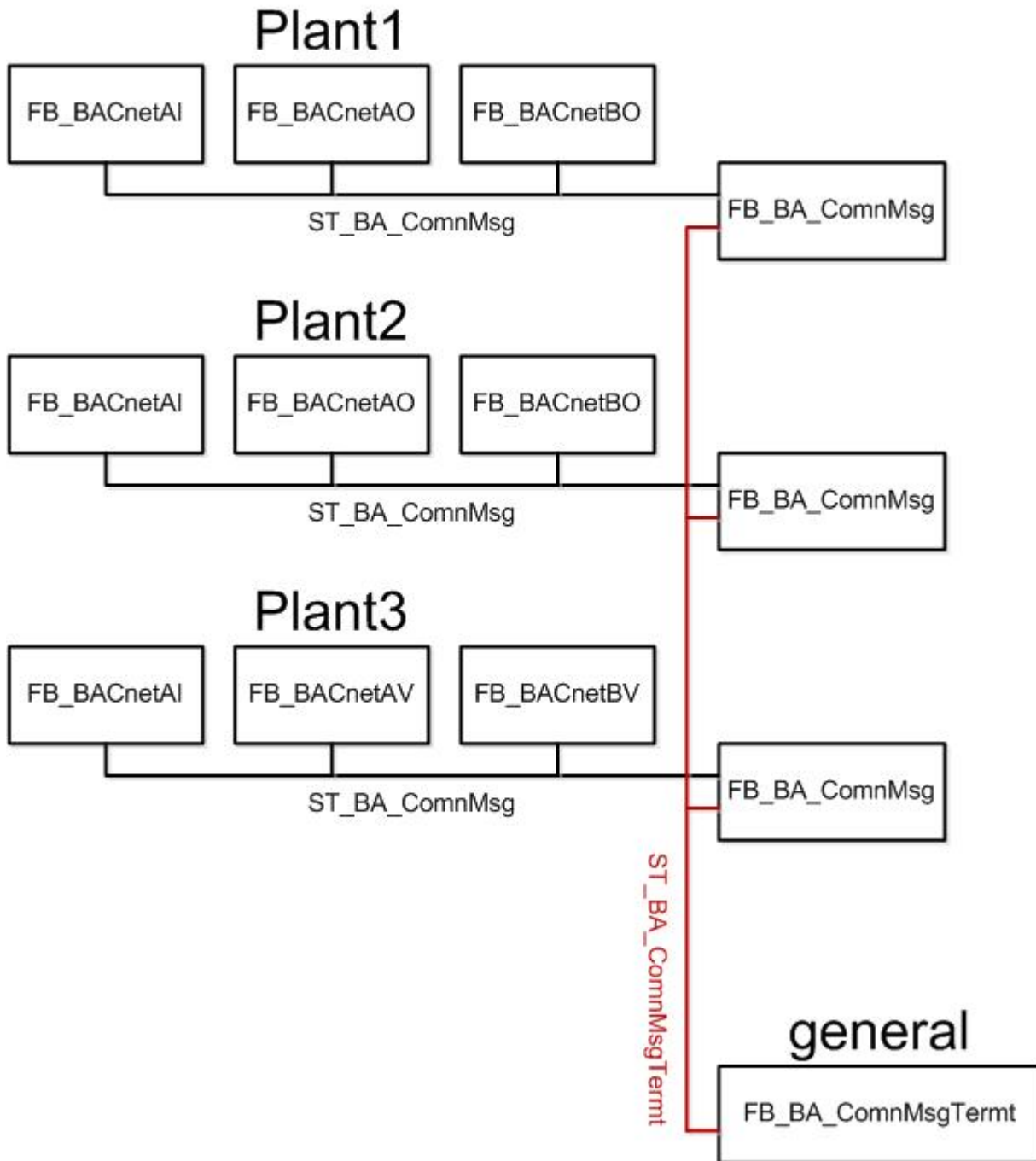
8.2.78 FB_BA_ComnMsg

Bildung von Sammelmeldungen



Funktionsbeschreibung

Die Funktionsbausteine `FB_BA_CmnMsg` und `FB_BA_CmnMsgTermt` [▶ 203] dienen zur Bildung von Sammelmeldungen in der Anlagenebene und in der Controllerebene. Die Informationen der BACnet-Objekte werden in der Struktur `ST_BA_CmnMsg` an die Funktionsbausteine `ST_BA_CmnMsg` [▶ 331] übertragen. Mit der Variablen `stOut` werden die Sammelmeldungen der Anlagen an den Funktionsbaustein `FB_BA_CmnMsgTermt` [▶ 203] weiter geleitet. Hier sind alle Meldungen der BACnet-Objekte eines BACnet-Controllers zusammengefasst.



Fehlerbehandlung:

Jedes BACnet-Objekt reserviert sich im ersten Zyklus nach einem Neustart der Steuerung ein Feld in einem eindimensionalen Array innerhalb der Meldestruktur `ST_BA_CmnMsg` [▶ 331]. Die Größe des Arrays ist mit der globalen Konstante `gBa_cMaxArrCmnMsg` festgelegt. Die Standardeinstellung dieser Konstante ist 256. Falls mehr als 256 BACnet-Objekte an eine Instanz von `FB_BA_CmnMsg` angeschlossen werden,

muss diese erhöht werden. Die Weiterleitung der Meldungen an den [FB_BA_CmnMsgTermt \[► 203\]](#) ist innerhalb der Struktur [ST_BA_CmnMsgTermt \[► 332\]](#) auf 64 festgelegt. Für eine Erhöhung muss die Konstante `gBa_cMaxArrCmnMsgTermt` vergrößert werden.

Address	Constant Name	Value
0001	VAR_GLOBAL CONSTANT	
0002		
0003	gBA_clrClsZero	: LREAL := 0.00001;
0004	gBA_cMaxSeqCtrl	: USINT := 16;
0005	gBA_cMaxSunBldScn	: UDINT := 20;
0006	gBA_cMaxRowFc	: INT := 10;
0007	gBA_cMaxColumnFc	: INT := 20;
0008	gBA_cMaxShdObj	: INT := 20;
0009	gBA_cMaxDataFileSize	: UINT := 65534; (*NOT allowed t
0010	gBA_cChrct32_NumOfElem	: UDINT := 32;
0011	gBA_cMaxNumOfStp	: UDINT := 32;
0012	gBa_cMaxArrCmnMsg	: INT := 256;
0013	gBa_cMaxArrCmnMsgTermt	: INT := 64;
0014		
0015	gBA_PropHighLimit	: UDINT := 1;
0016	gBA_PropLowLimit	: UDINT := 2;
0017	gBA_PropTimeDelay	: UDINT := 3;
0018	gBA_PropEventEnableBits	: UDINT := 4;
0019	gBA_PropLimitEnableBits	: UDINT := 5;
0020	gBA_PropNotificationClass	: UDINT := 6;

Ein-Ausgänge

VAR_IN_OUT

```
stIn      : ST_BA_CmnMsg;
stTermt   : ST_BA_CmnMsgTermt;
```

stIn: Datenstruktur zum Anschluss der BACnet-Objekte in der Anlagenebene

stTermt: [Datenstruktur \[► 332\]](#) zur Weitergabe der Sammelmeldungen von der Anlagenebene an den Funktionsbaustein [FB_BA_CmnMsgTermt \[► 203\]](#).

VAR_OUTPUT

```
iId       : INT;
bInAlm    : BOOL;
bFlt      : BOOL;
bOvrrd    : BOOL;
bOoServ   : BOOL;
bHiLmt    : BOOL;
bLoLmt    : BOOL;
udiHiActvPrio : UDINT;
arrActvPrio : ARRAY [1..16] OF BOOL;
iMaxNumOfEntry : INT;
iNumOfEntry : INT;
dwObjId   : DWORD;
eObjType  : E_BACnetObjectType;
bErr      : BOOL;
udiErrId  : UDINT;
udiErrArg : UDINT;
```

iId: Die Kennzeichnung des zuletzt an den [FB_BA_CmnMsgTermt](#) gesendeten Informations-Telegramms (`stTermt.arrData[iId]`)

bInAlm: Das zuletzt sendende BACnet-Objekt ist im Status Alarm.

bFlt: Das zuletzt sendende BACnet-Objekt ist im Status Fault.

bOvrrd: Beim zuletzt sendenden BACnet-Objekt ist die lokale mechanische Vorrangbedienung aktiviert.

bOoSrv: Das zuletzt sendende BACnet-Objekt ist "out of service".

bHiLmt: Beim zuletzt sendenden BACnet-Objekt ist das Hi-Limit überschritten.

bLoLmt: Beim zuletzt sendenden BACnet-Objekt ist das Lo-Limit unterschritten.

udiHiActvPrio: gibt die höchste geschriebene Priorität aller BACnet-Objekte an.

arrActvPrio: Das Array gibt eine Übersicht der aktiven Prioritäten aller angeschlossenen BACnet-Objekte.

iMaxNumOfEntry: Zeigt an wie viele BACnet-Objekte an den Funktionsbaustein angeschlossen werden können.

iNumOfEntry: Zeigt an wie viele BACnet-Objekte an den Funktionsbaustein angeschlossen sind.

dwObjId: Die Id des Objektes, welches zuletzt in den Fehlerzustand gegangen ist.

eObjType: Der Objekttyp des Objektes, welches zuletzt in den Fehlerzustand gegangen ist.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

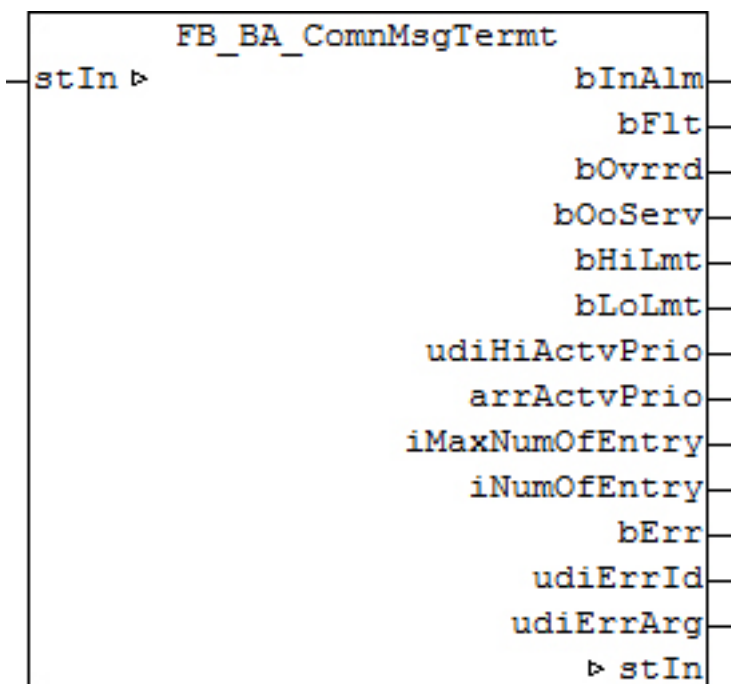
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.79 FB_BA_ComnMsgTermt

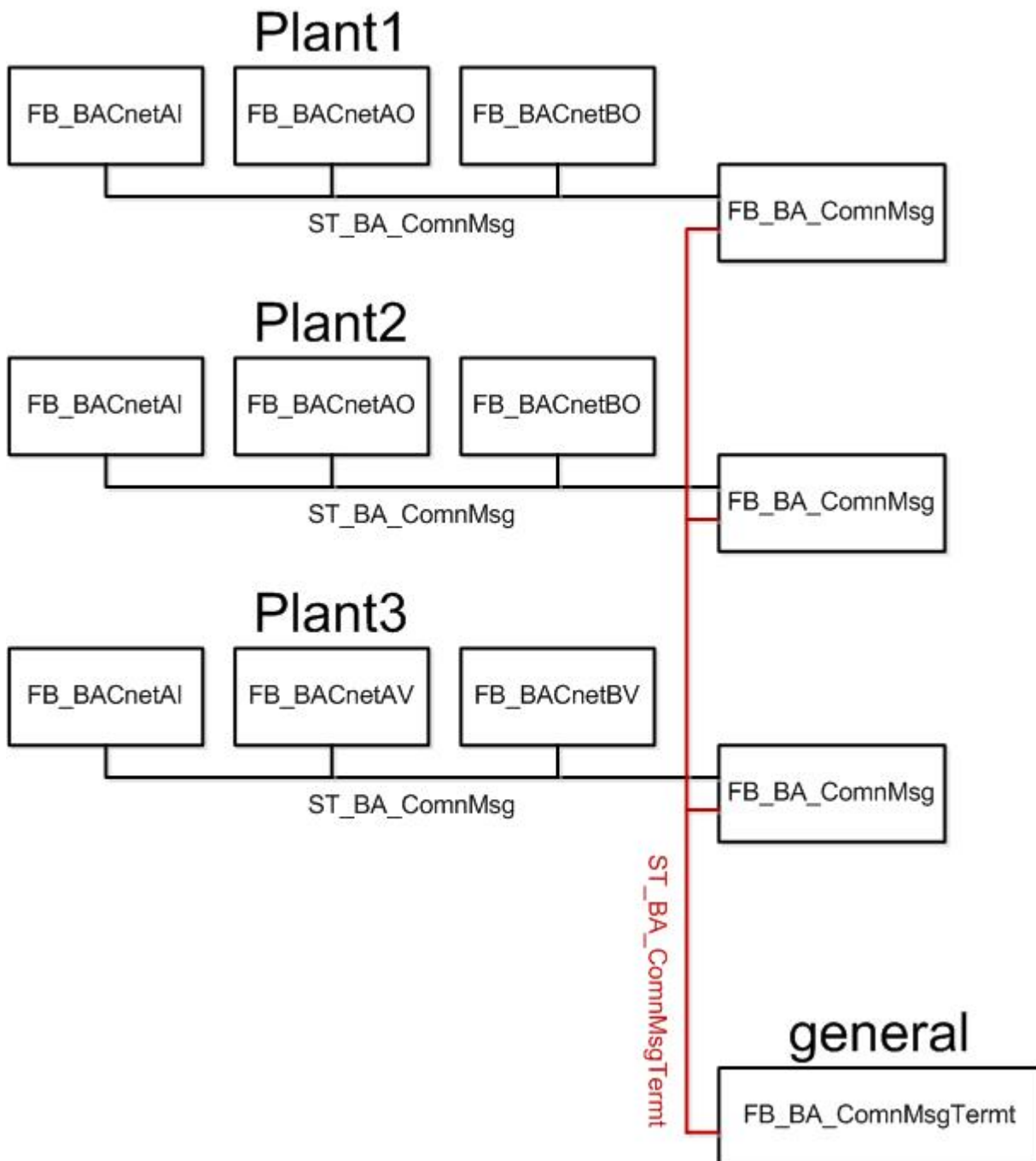
Bildung von Sammelmeldungen



Funktionsbeschreibung

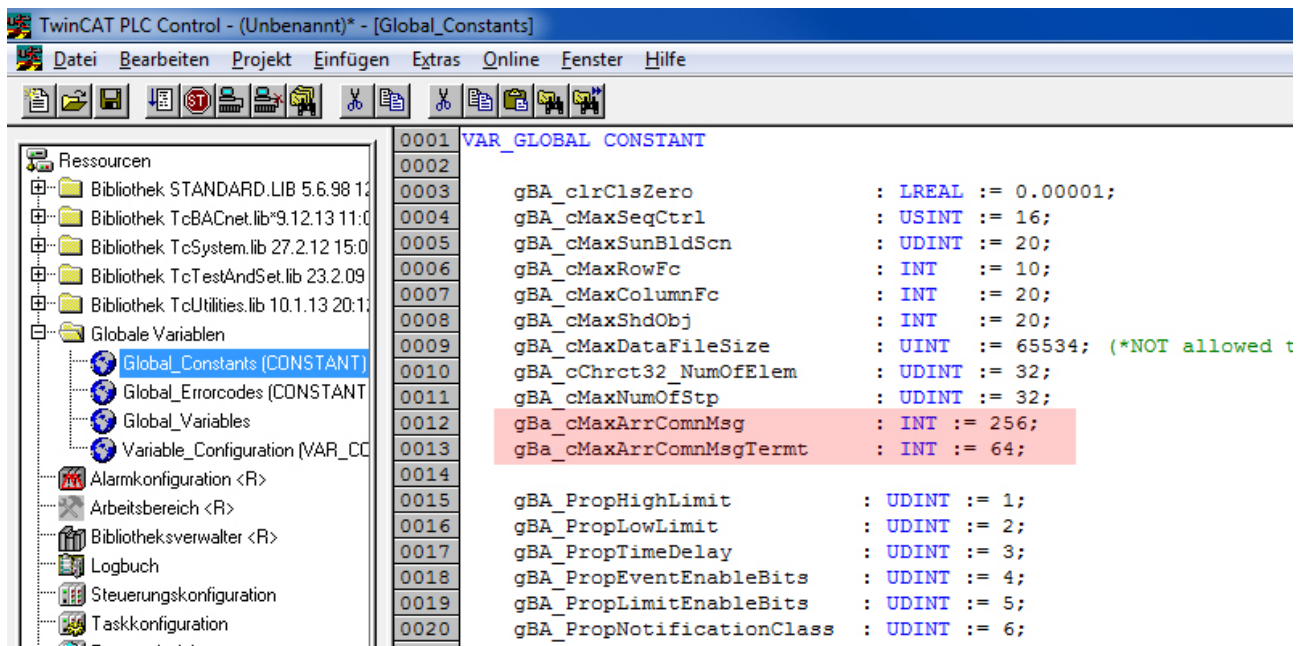
Die Funktionsbausteine [FB_BA_ComnMsg](#) [► 200] und [FB_BA_ComnMsgTermt](#) dienen zur Bildung von Sammelmeldungen in der Anlagenebene und in der Controllerebene.

Die Informationen der BACnet-Objekte werden in der Struktur [ST_BA_ComnMsg](#) [► 331] an die Funktionsbausteine [FB_BA_ComnMsg](#) [► 200] übertragen. Mit der Variablen *stOut* werden die Sammelmeldungen der Anlagen an den Funktionsbaustein [FB_BA_ComnMsgTermt](#) weiter geleitet. Hier sind alle Meldungen der BACnet-Objekte eines BACnet-Controllers zusammengefasst.



Fehlerbehandlung:

Jedes BACnet-Objekt reserviert sich im ersten Zyklus nach einem Neustart der Steuerung ein Feld in einem eindimensionalen Array innerhalb der Meldestruktur `ST_BA_CmnMsg` [▶ 331]. Die Größe des Arrays ist mit der globalen Konstante `gBa_cMaxArrCmnMsg` festgelegt. Die Standardeinstellung dieser Konstante ist 256. Falls mehr als 256 BACnet-Objekte an eine Instanz von `FB_BA_CmnMsg` [▶ 200] angeschlossen werden, muss diese erhöht werden. Die Weiterleitung der Meldungen an den `FB_BA_CmnMsgTermt` ist innerhalb der Struktur `ST_BA_CmnMsgTermt` auf 64 festgelegt. Für eine Erhöhung muss die Konstante `gBa_cMaxArrCmnMsgTermt` vergrößert werden.



Ein-Ausgänge

VAR_IN_OUT

stIn : ST_BA_ComnMsgTermt;

stIn: Datenstruktur [► 332] zur Entgegennahme der Sammelmeldungen von der Anlagenebene an den Funktionsbaustein FB_BA_ComnMsgTermt.

VAR_OUTPUT

bInAlm : BOOL;
 bFlt : BOOL;
 bOvrrd : BOOL;
 bOoServ : BOOL;
 bHiLmt : BOOL;
 bLoLmt : BOOL;
 udiHiActvPrio : UDINT;
 arrActvPrio : ARRAY [1..16] OF BOOL;
 iMaxNumOfEntry : INT;
 iNumOfEntry : INT;
 dwObjId : DWORD;
 eObjType : E_BACnetObjectType;
 bErr : BOOL;
 udiErrId : UDINT;
 udiErrArg : UDINT;

bInAlm: Das zuletzt sendende BACnet-Objekt ist im Status Alarm.

bFlt: Das zuletzt sendende BACnet-Objekt ist im Status Fault.

bOvrrd: Beim zuletzt sendenden BACnet-Objekt ist die lokale mechanische Vorrangbedienung aktiviert.

bOoServ: Das zuletzt sendende BACnet-Objekt ist "out of service".

bHiLmt: Beim zuletzt sendenden BACnet-Objekt ist das Hi-Limit überschritten.

bLoLmt: Beim zuletzt sendenden BACnet-Objekt ist das Lo-Limit unterschritten.

udiHiActvPrio: gibt die höchste geschriebene Priorität aller BACnet-Objekte an.

arrActvPrio: Das Array gibt eine Übersicht der aktiven Prioritäten aller angeschlossenen BACnet-Objekte.

iMaxNumOfEntry: Zeigt an wie viele BACnet-Objekte an den Funktionsbaustein angeschlossen werden können.

iNumOfEntry: Zeigt an wie viele BACnet-Objekte an den Funktionsbaustein angeschlossen sind.

eObjType: Der Objekttyp des Objektes, welches zuletzt in den Fehlerzustand gegangen ist.

dwObjId: Die Id des Objektes, welches zuletzt in den Fehlerzustand gegangen ist.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [► 340].

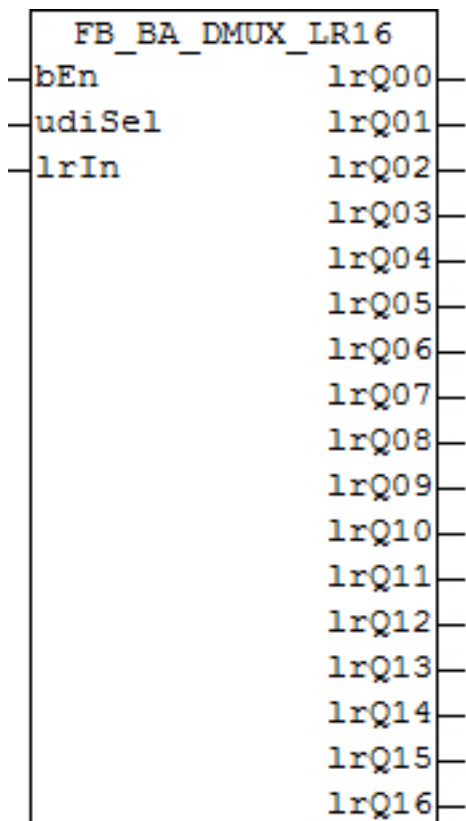
Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.80 FB_BA_DMUX_XX

Die Demultiplexer-Bausteine existieren für verschiedene Variablentypen (BOOL, INT, LREAL, REAL, USINT, UINT und UDINT) und in verschiedenen Ausgangsgrößen (5, 9, 13 und 17), haben jedoch alle die gleiche Funktionalität.

Exemplarisch wird hier der Baustein FB_BA_DMUX_LR16 beschrieben.



Funktionsbeschreibung

Der Funktionsbaustein gibt im aktivierten Zustand ($bEn=TRUE$) den Wert am Eingang $lrIn$ an demjenigen Ausgang $lrQ01..lrQ16$ aus dessen Nummer am Eingang $udiSel$ eingetragen ist. Alle anderen Ausgänge werden auf 0 gesetzt (bei den booleschen Demultiplexern auf FALSE).

Beispiel:

Eingänge	Ausgänge
$bEn = TRUE$	$lrQ00 = 0.0$

Eingänge	Ausgänge
udiSel = 5	IrQ01 = 0.0
lrIn = 32.5	IrQ02 = 0.0
	IrQ03 = 0.0
	IrQ04 = 0.0
	IrQ05 = 32.5
	IrQ06 = 0.0
	IrQ07 = 0.0
	IrQ08 = 0.0
	IrQ09 = 0.0
	IrQ10 = 0.0
	IrQ11 = 0.0
	IrQ12 = 0.0
	IrQ13 = 0.0
	IrQ14 = 0.0
	IrQ15 = 0.0
	IrQ16 = 0.0

Ist der eingetragene Wert an *udiSel* größer als die Anzahl der Ausgänge, so wird am "höchsten" Ausgang der Wert von *lrIn* ausgegeben:

Eingänge	Ausgänge
bEn = TRUE	IrQ00 = 0.0
udiSel = 25	IrQ01 = 0.0
lrIn = 32.5	IrQ02 = 0.0
	IrQ03 = 0.0
	IrQ04 = 0.0
	IrQ05 = 0.0
	IrQ06 = 0.0
	IrQ07 = 0.0
	IrQ08 = 0.0
	IrQ09 = 0.0
	IrQ10 = 0.0
	IrQ11 = 0.0
	IrQ12 = 0.0
	IrQ13 = 0.0
	IrQ14 = 0.0
	IrQ15 = 0.0
	IrQ16 = 32.5

Bei *bEn*=FALSE wird an allen Ausgängen 0.0 bzw. FALSE bei den booleschen Demultiplexern ausgegeben.

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
udiSel   : UDINT;
lrIn     : LREAL;
```

bEn: Aktivierung der Bausteinfunktion.

udiSel: Nummer des Ausgangs *IrQ00...IrQ16*, der den Wert des Eingangs *lrIn* annehmen soll.

lrIn: Wert, der zur Ausgabe kommen soll.

VAR_OUTPUT

```

lrQ00 : LREAL;
lrQ01 : LREAL;
lrQ02 : LREAL;
lrQ03 : LREAL;
lrQ04 : LREAL;
lrQ05 : LREAL;
lrQ06 : LREAL;
lrQ07 : LREAL;
lrQ08 : LREAL;
lrQ09 : LREAL;
lrQ10 : LREAL;
lrQ11 : LREAL;
lrQ12 : LREAL;
lrQ13 : LREAL;
lrQ14 : LREAL;
lrQ15 : LREAL;
lrQ16 : LREAL;

```

lrQ00...lrQ16: Wertausgänge.

Voraussetzungen

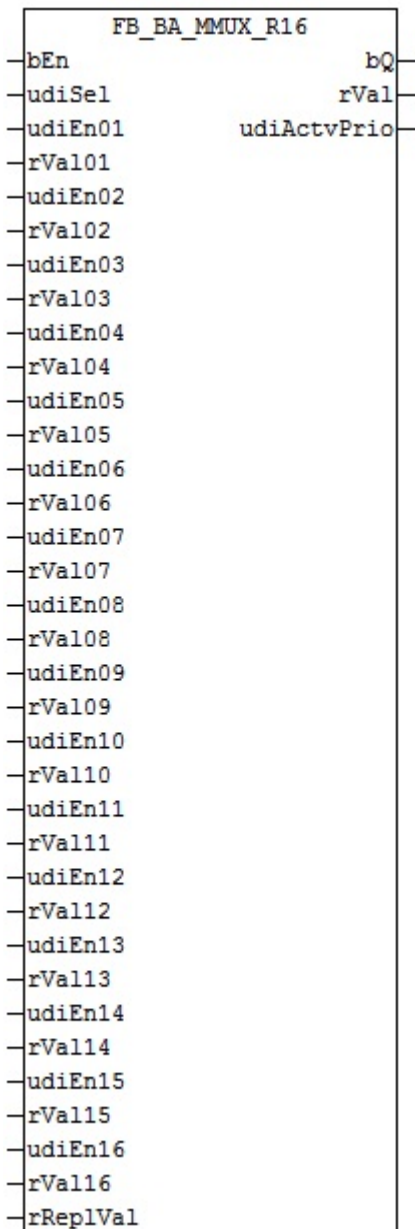
Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.81 FB_BA_MMUX_XX

Der Funktionsbaustein schaltet in Abhängigkeit eines Selektors und der entsprechenden Eingangselektorbedingung einen Eingangswert auf den Ausgang.

Die Multiplexer-Bausteine existieren für verschiedene Variablentypen (BOOL, INT, LREAL, REAL, USINT, UINT und UDINT) und in verschiedenen Eingangsgrößen (4, 8, 12, 16 und 24), haben jedoch alle die gleiche Funktionalität.

Exemplarisch wird hier der Baustein FB_BA_MMUX_R16 beschrieben.



Funktionsbeschreibung

Der Funktionsbaustein schaltet im aktivierten Zustand (*bEn*=TRUE) in Abhängigkeit eines Selektors *udiSel* und der entsprechenden Eingangs-Selektorbedingung *udiEnxx* einen von den Eingangswerten *rValxx* auf den Ausgang *rVal*.

Wenn mehrere Eingangs-Selektorbedingungen *udiEn01...udiEn16* gleich sind und der Selektor *udiSel* einer Bedingung entspricht, dann wird der Eingangswert *rVal01...rVal16* der tiefsten aktiven Selektorbedingung auf den Ausgang *rVal* geschaltet. *udiEn01* ist die tiefste, *udiEn16* die höchste Selektorbedingung.

Die Ausgangsvariable *bQ* zeigt an, dass der Selektor *udiSel* mit einer Eingangs-Selektorbedingung *udiEnxx* übereinstimmt.

Die Ausgangsvariable *udiActvPrio* zeigt die aktive Selektorbedingung an.

Ist keine Selektorbedingung aktiv, so wird *rReplVal* an *rVal* ausgegeben. *bQ* ist dann FALSE und *udiActvPrio* zeigt eine 255 an.

Beispiel:

Eingänge		Ausgang	
Variable	Wert	Variable	Wert

Eingänge		Ausgang	
bEn	TRUE	bQ	TRUE
udiSel	5	rVal	1,123
udiEn01	4	udiActvPrio	7
rVal01	123		
udiEn02			
rVal02			
udiEn03	3		
rVal03	321		
udiEn04			
rVal04			
udiEn05	8		
rVal05	345		
udiEn06			
rVal06			
udiEn07	5		
rVal07	1,123		
udiEn08			
rVal08			
udiEn09	5		
rVal09	5,4321		
udiEn10			
rVal10			
udiEn11			
rVal11			
udiEn12			
rVal12			
udiEn13			
rVal13			
udiEn14			
rVal14			
udiEn15			
rVal15			
udiEn16			
rVal16			
rReplVal			

Ein-/Ausgänge

VAR_INPUT

```

bEn      : BOOL;
udiSel   : UDINT;
udiEn01  : UDINT := 255;
rVal01   : REAL;
udiEn02  : UDINT := 255;
rVal02   : REAL;
udiEn03  : UDINT := 255;
rVal03   : REAL;
udiEn04  : UDINT := 255;
rVal04   : REAL;
udiEn05  : UDINT := 255;
rVal05   : REAL;
udiEn06  : UDINT := 255;
rVal06   : REAL;
udiEn07  : UDINT := 255;
rVal07   : REAL;

```

```

udiEn08 : UDINT := 255;
rVal08  : REAL;
udiEn09 : UDINT := 255;
rVal09  : REAL;
udiEn10 : UDINT := 255;
rVal10  : REAL;
udiEn11 : UDINT := 255;
rVal11  : REAL;
udiEn12 : UDINT := 255;
rVal12  : REAL;
udiEn13 : UDINT := 255;
rVal13  : REAL;
udiEn14 : UDINT := 255;
rVal14  : REAL;
udiEn15 : UDINT := 255;
rVal15  : REAL;
udiEn16 : UDINT := 255;
rVal16  : REAL;
rReplVal : REAL;
    
```

bEn: Aktivierung der Bausteinfunktion

udiSel: Selektor

udiEn01..udiEn16: Eingangs-Selektorbedingung
Die Eingangsvariablen sind auf den Wert 255 vorinitialisiert.

rVal01...rVal16: Eingangswerte, aus denen gewählt werden soll.

rReplVal: Ersatzwert, wenn keine Eingangs-Selektorbedingung aktiv ist.

VAR_OUTPUT

```

bQ      : BOOL;
rVal    : REAL;
udiActvPrio : UDINT;
    
```

bQ: Ist TRUE, wenn der Selektor *udiSel* mit einer Eingangs-Selektorbedingung *udiEnxx* übereinstimmt.

rVal: Wert der gewählten Eingangs-Selektorbedingung

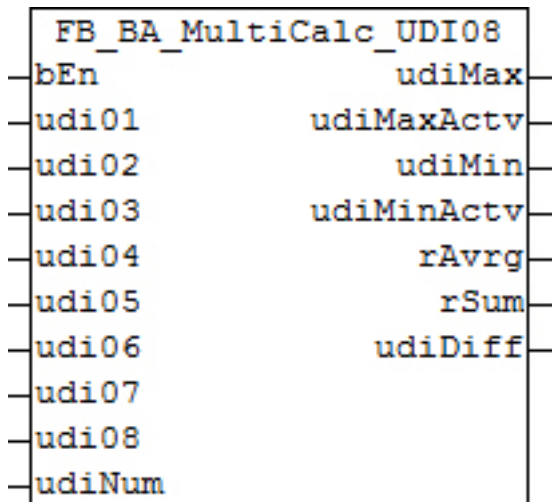
udiActvPrio: Zeigt an, welche Eingangs-Selektorbedingung aktiv ist.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.82 FB_BA_MultiCalc_XX

Die Multi-Kalkulations-Bausteine existieren für verschiedene Variablentypen (LREAL, REAL, INT, UINT und UDINT) und in verschiedenen Eingangsgrößen (4 und 8), haben jedoch alle die gleiche Funktionalität. Exemplarisch wird hier der Baustein FB_BA_UDI08 beschrieben.



Funktionsbeschreibung

Der Funktionsbaustein ermittelt im aktivierten Zustand ($bEn=TRUE$) aus den 8 Eingangswerten $udi01...udi08$ folgendes:

- den Maximalwert von allen Eingängen $udiMax$
- den Eingang, an dem dieser Maximalwert anliegt $udiMinActv$
- den Minimalwert von allen Eingängen $udiMin$
- den Eingang, an dem dieser Minimalwert anliegt $udiMinActv$
- den Mittelwert aller Eingänge $rAavg$
- die Summe aller Eingänge $rSum$
- die Differenz zwischen dem Maximal- und Minimalwert $udiDiff$

Sollen nicht alle Eingänge zur Berechnung kommen, so lässt sich die Anzahl durch einen Eintrag an $udiNum$ einschränken: mit $udiNum=6$ beispielsweise werden die Berechnungen nur für die Eingänge $udi01...udi06$ durchgeführt.

Ein Eintrag größer als 8 wird automatisch auf 8 limitiert, ein Eintrag kleiner als 1 automatisch auf 1.

Beispiel:

Eingänge	Ausgang
$bEn = TRUE$	$udiMax = 32$
$udi01 = 32$	$udiMaxActv = 1$
$udi02 = 17$	$udiMin = 5$
$udi03 = 5$	$udiMinActv = 3$
$udi04 = 9$	$rAavg = 18.5$
$udi05 = 16$	$rSum = 111$
$udi06 = 32$	$udiDiff = 27$
$udi07 = 25$	
$udi08 = 44$	
$udiNum = 6$	

Bei $bEn=FALSE$ wird an allen Ausgängen 0 ausgegeben.

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
udi01    : UDINT;
udi02    : UDINT;
udi03    : UDINT;
```

```
udi04 : UDINT;
udi05 : UDINT;
udi06 : UDINT;
udi07 : UDINT;
udi08 : UDINT;
udiNum : UDINT;
```

bEn: Aktivierung der Bausteinfunktion

udi01...udi08: Eingangswerte, aus denen berechnet werden soll.

udiNum: Anzahl der Eingangswerte, mit denen gerechnet wird.

VAR_OUTPUT

```
udiMax : UDINT;
udiMaxActv : UDINT;
udiMin : UDINT;
udiMinActv : UDINT;
rAavg : REAL;
rSum : REAL;
udiDiff : UDINT;
```

udiMax: Maximalwert von allen Eingängen

udiMaxActv: Eingang, an dem der Maximalwert anliegt.

udiMin: Minimalwert von allen Eingängen

udiMinActv: Eingang, an dem der Minimalwert

rAavg: Mittelwert aller Eingänge

rSum: Summe aller Eingänge

udiDiff: Differenz zwischen dem Maximal- und Minimalwert

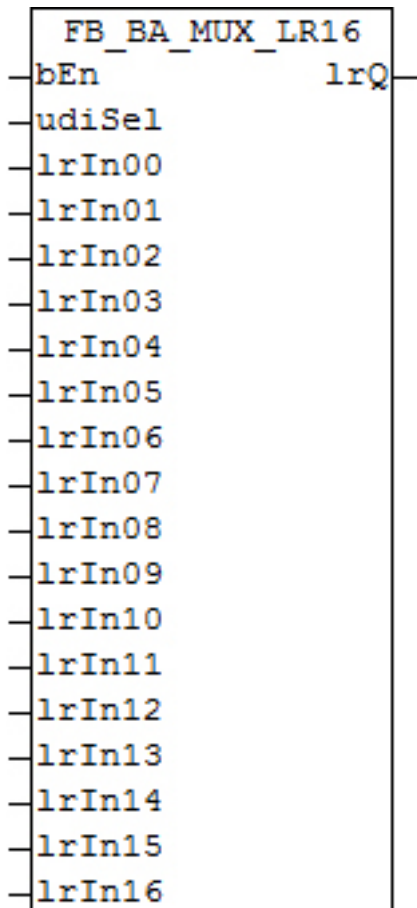
Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.83 FB_BA_MUX_XX

Die Multiplexer-Bausteine existieren für verschiedene Variablentypen (BOOL, INT, LREAL, REAL, USINT, UINT und UDINT) und in verschiedenen Eingangsgrößen (5, 9, 13 und 17), haben jedoch alle die gleiche Funktionalität.

Exemplarisch wird hier der Baustein FB_BA_MUX_LR16 beschrieben.



Funktionsbeschreibung

Der Funktionsbaustein gibt im aktivierten Zustand ($bEn=TRUE$) denjenigen Eingangswert $lrIn00..lrIn16$ am Ausgang lrQ aus, dessen Nummer am Eingang $udiSel$ eingetragen ist.

Beispiel:

Eingänge	Ausgang
bEn = TRUE	lrQ = 16.5
udiSel = 5	
lrIn00 = 15.9	
lrIn01 = 32.5	
lrIn02 = 17.4	
lrIn03 = 5.84	
lrIn04 = 9.56	
lrIn05 = 16.5	
lrIn06 = 32.781	
lrIn07 = 25.4	
lrIn08 = 44.5	
lrIn09 = 66.1	
lrIn10 = 45.5	
lrIn11 = 83.3	
lrIn12 = 54.56	
lrIn13 = 33.8	
lrIn14 = 98.5	
lrIn15 = 71.3	

Eingänge	Ausgang
lrIn16 = 2.3	

Ist der eingetragene Wert an *udiSel* größer als die Anzahl der Eingänge, so wird der "höchstrangige" Eingang an *lrQ* ausgegeben:

Eingänge	Ausgang
bEn = TRUE	lrQ = 2.3
udiSel = 25	
lrIn00 = 15.9	
lrIn01 = 32.5	
lrIn02 = 17.4	
lrIn03 = 5.84	
lrIn04 = 9.56	
lrIn05 = 16.5	
lrIn06 = 32.781	
lrIn07 = 25.4	
lrIn08 = 44.5	
lrIn09 = 66.1	
lrIn10 = 45.5	
lrIn11 = 83.3	
lrIn12 = 54.56	
lrIn13 = 33.8	
lrIn14 = 98.5	
lrIn15 = 71.3	
lrIn16 = 2.3	

Bei *bEn*=FALSE wird am Ausgang *lrQ* 0.0 bzw. FALSE bei den booleschen Multiplexern ausgegeben.

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
udiSel  : UDINT;
lrIn00  : LREAL;
lrIn01  : LREAL;
lrIn02  : LREAL;
lrIn03  : LREAL;
lrIn04  : LREAL;
lrIn05  : LREAL;
lrIn06  : LREAL;
lrIn07  : LREAL;
lrIn08  : LREAL;
lrIn09  : LREAL;
lrIn10  : LREAL;
lrIn11  : LREAL;
lrIn12  : LREAL;
lrIn13  : LREAL;
lrIn14  : LREAL;
lrIn15  : LREAL;
lrIn16  : LREAL;
```

bEn: Aktivierung der Bausteinfunktion

udiSel: Nummer des Eingangs, dessen Wert am Ausgang *lrQ* ausgegeben werden soll.

lr00...lr16: Eingangswerte, aus denen gewählt werden soll

VAR_OUTPUT

```
lrQ     : UDINT;
```

IrQ: Wert des gewählten Eingangs

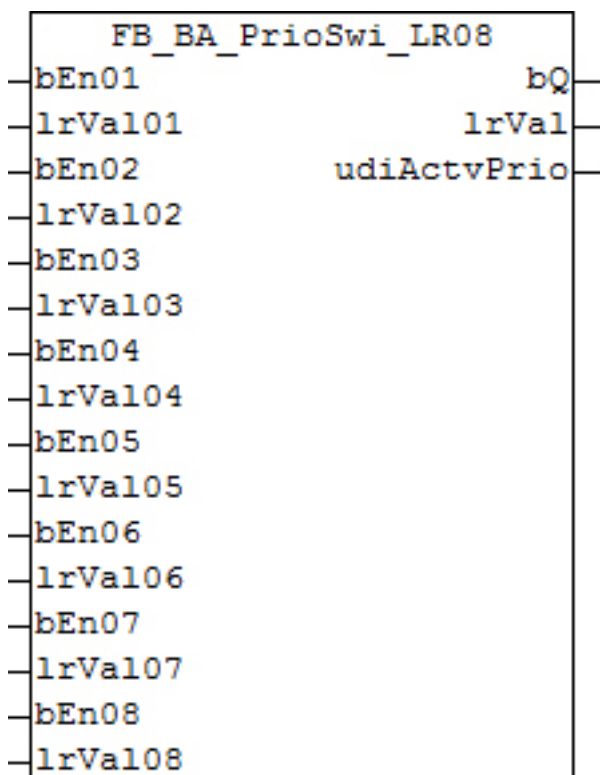
Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.84 FB_BA_PrioSwi_XX

Die Prioritätenschalter existieren für verschiedene Variablentypen (BOOL, INT, LREAL, REAL, USINT, UINT und UDINT) und in verschiedenen Ausgangsgrößen (4, 8, 12 und 16 bzw. 24), haben jedoch alle die gleiche Funktionalität.

Exemplarisch wird hier der Baustein FB_BA_PrioSwi_LR08 beschrieben.



Funktionsbeschreibung

Zur Selektion verschiedener Werte stehen Prioritätenschalter zur Verfügung. Zum Ausgang *lrVal* wird derjenige Wert mit der höchsten Priorität durchgeschaltet dessen Eingang *bEnxx* TRUE ist.

Beispiel:

Eingänge			Ausgänge		
bEn01	FALSE		bQ	TRUE	
lrVal01		32.5	lrVal		5.84
bEn02	FALSE		udiActvPrio		3
lrVal02		17.4			
bEn03	TRUE				
lrVal03		5.84			
bEn04	TRUE				
lrVal04		9.56			

Eingänge		Ausgänge			
bEn05	FALSE				
lrVal05		16.5			
bEn06	TRUE				
lrVal06		32.781			
bEn07	FALSE				
lrVal07		25.4			
bEn08	TRUE				
lrVal08		44.5			

Ist keine der Prioritäten freigegeben, so fällt der Ausgang *bQ* auf FALSE. An den Ausgängen *lrVal* und *udiActvPrio* wird jeweils 0 ausgegeben. Bei einen booleschen Prioritätenschalter wird an dem Ausgang *bVal* dann FALSE ausgegeben.

Eingänge		Ausgänge			
bEn01	FALSE		bQ	FALSE	
lrVal01		32.5	lrVal		0.0
bEn02	FALSE		udiActvPrio		0
lrVal02		17.4			
bEn03	FALSE				
lrVal03		5.84			
bEn04	FALSE				
lrVal04		9.56			
bEn05	FALSE				
lrVal05		16.5			
bEn06	FALSE				
lrVal06		32.781			
bEn07	FALSE				
lrVal07		25.4			
bEn08	FALSE				
lrVal08		44.5			

Ein-/Ausgänge

VAR_INPUT

```

bEn01 : BOOL;
lrVal01 : LREAL;
bEn02 : BOOL;
lrVal02 : LREAL;
bEn03 : BOOL;
lrVal03 : LREAL;
bEn04 : BOOL;
lrVal04 : LREAL;
bEn05 : BOOL;
lrVal05 : LREAL;
bEn06 : BOOL;
lrVal06 : LREAL;
bEn07 : BOOL;
lrVal07 : LREAL;
bEn08 : BOOL;
lrVal08 : LREAL;
    
```

bEn01...bEn08: Freigabe des Prioritätswerts.

lrVal01...lrVal08: Prioritätswert.

VAR_OUTPUT

```
bQ      : BOOL;
lrVal   : LREAL;
udiActvPrio : UDINT;
```

bQ: Ausgabe, ob überhaupt eine Priorität freigegeben ist.

lrVal: Ausgabe des Wertes der aktuellen (höchsten) Priorität, die freigegeben ist.

udiActvPrio: Aktuelle (höchste) Priorität, die freigegeben ist.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.85 FB_BA_AntBlkg

Blockierschutz für Pumpen und Stellantrieben

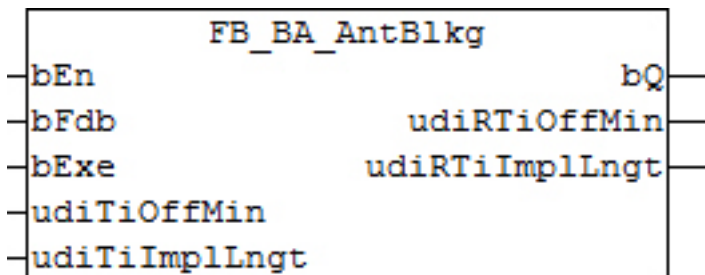


Abb. 1: FB_BA_AntBlkg

Funktionsbeschreibung

Dieser Funktionsbaustein verhindert das Blockieren von Pumpen oder Stellantrieben nach längeren Zeiten ohne Bewegung durch die Ausgabe eines Einschaltimpulses.

Die maximale Dauer des Stillstands bis zur Ausgabe eines Impulses wird durch den Wert der Variablen *udiTiOffMin* bestimmt. Zur Erfassung der Stillstandszeit muss der Eingang *bFdb* mit der Betriebsrückmeldung des Aggregates verbunden werden. Die Länge des Impulses wird mit der Variablen *udiTiImplLngt* parametrisiert. Falls die Blockierschutzimpulse nicht in Abhängigkeit der Stillstandszeiten, sondern zyklisch von einem Zeitschaltplan ausgehen sollen, ist hierfür der Eingang *bExe* zu verwenden. Bei einer steigenden Flanke an *bExe* wird unmittelbar ein Impuls an *bQ* ausgegeben. Eine Impulsausgabe erfolgt generell nur wenn der Funktionsbaustein an *bEn* frei geschaltet ist.

Ein-/Ausgänge**VAR_INPUT**

```
bEn      : BOOL;
bFdb     : BOOL;
bExe     : BOOL;
udiTiOffMin : UDINT;
udiTiImplLngt : UDINT;
```

bEn: Freigabe des Funktionsbausteins

bFdb: Eingang zum Anschluss des Feedbacksignals eines Motors oder Ventils

bExe: Steigende Flanke forciert die Ausgabe eines Impulses

udiTiOffMin: Mindestausschaltzeit [s]: Nach dem Ablauf der Zeit von *udiTiOffMin* ohne Bewegung des Aggregates erfolgt die Ausgabe eines Impulses.

udiTImplLngt: Länge des Blockieschutzimpulses [s] an *bQ*

VAR_OUTPUT

```
bQ : BOOL;
udiRTiOffMin : UDINT;
udiRTiImplLngt : UDINT;
```

bQ: Ausgang zur Ausgabe des Impulses

udiRTiOffMin: Verbleibende Zeit [s] bis zur Ausgabe des nächsten Impulses bei andauerndem Stillstand.

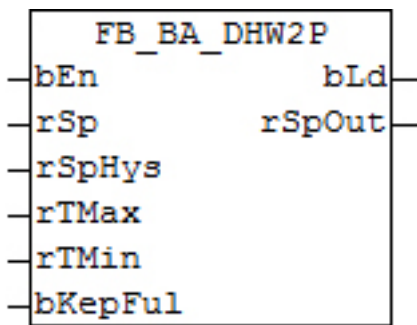
udiTImplLngt: Verbleibende Restzeit [s] des Impulses an *bQ*

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.86 FB_BA_DHW2P

Warmwasserspeicher-Steuerung



Funktionsbeschreibung

Der Funktionsbaustein steuert die Ladung eines Warmwasserspeichers mittels eines Zweipunkt-Reglers. An dem Eingang *bEn* wird die Speicherladung frei gegeben. Bei aktiver Speicherladung ist der Ausgang *bLd* TRUE. Mit der Variablen *rSp* wird dem Funktionsbaustein der Sollwert für die Brauchwassertemperatur übergeben. An dem Eingang *rTMin* wird eine Minimalauswahl, am Eingang *rTMax* eine Maximalauswahl aller Temperaturfühler des Warmwasserspeichers angeschlossen.

Bedingt durch die Temperaturschichtung im Warmwasserspeicher ist der oberste Fühler in allgemeinen der mit der höchsten Temperatur und der untere derjenige mit der niedrigsten.

Die Speicherladung kann mittels der Variablen *bKepFul* in zwei Arten erfolgen:

bKepFul = FALSE

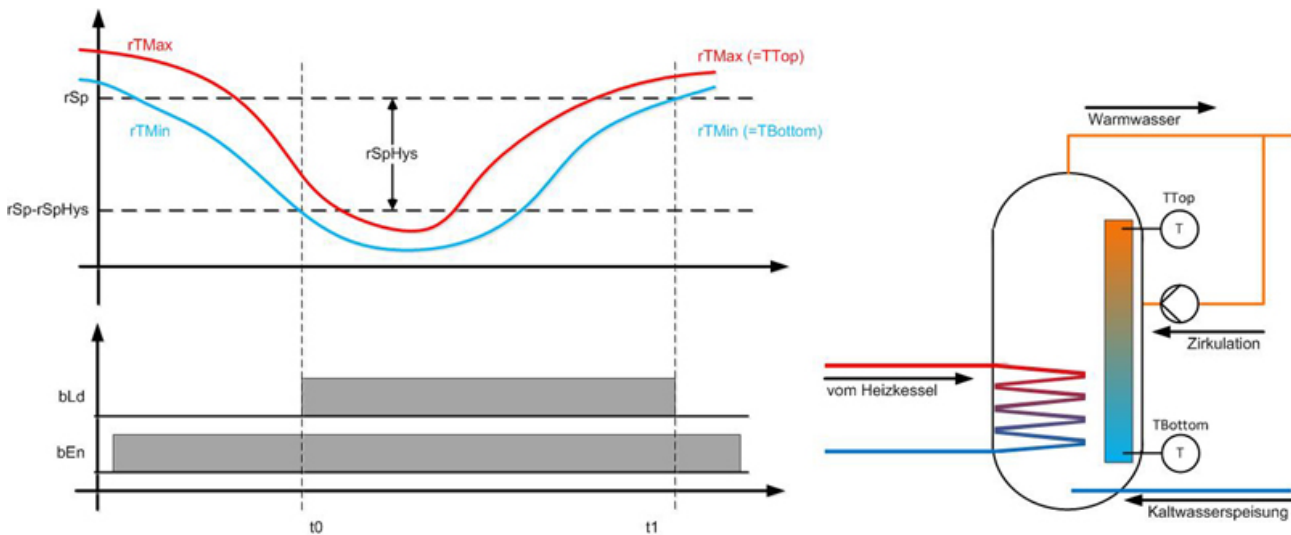
Die Ladung wird angefordert wenn *rTMax* unter dem Wert von *rSp-rSpHys* liegt. Die Ladung wird deaktiviert wenn *rTMin* über dem Sollwert von *rSp* liegt.

Dadurch, dass im allgemeinen der oberste Fühler die höchste Temperatur misst, erfolgt eine Einschaltung der Speicherladung deshalb erst bei entladem Warmwasserspeicher.

bKepFul = TRUE

Die Ladung wird angefordert wenn *rTMin* unter dem Wert von *rSp-rSpHys* liegt. Die Ladung wird deaktiviert wenn *rTMin* wieder über dem Sollwert ist.

Durch die Minimalauswahl aller Speichertemperaturen, wird der kälteste Punkt des Speichers für die Regelung verwendet. Eine Nachladung erfolgt sobald der Speicher nicht mehr ganz gefüllt ist.



Ein-/Ausgänge

VAR_INPUT

```

bEn      : BOOL;
rSp      : REAL;
rSpHys   : REAL;
rTMax    : REAL;
rTMin    : REAL;
bKepFul  : BOOL;

```

bEn: Freigabe Boiler Ladung

rSp: Temperatursollwert des Brauchwassers [°C]

rSpHys: Hysterese, empfohlen 1°K bis 5°K

rTMax: Maximalauswahl aller Speichertemperaturfühler [°C]

rTMin: Minimalauswahl aller Speichertemperaturfühler [°C]

bKepFul: Steuertemperaturwahl:

FALSE = mit *rTMax* wird *bLd* gefordert, mit *rTMin* ausgeschaltet

TRUE = *rTMin* allein steuert das Ein-Ausschalten von *bLd*

VAR_OUTPUT

```

bLd      : BOOL;
rSpOut   : REAL;

```

bLd: Freigabe des Ladebetriebs

rSpOut: Sollwertweitergabe an Ladeschaltung:

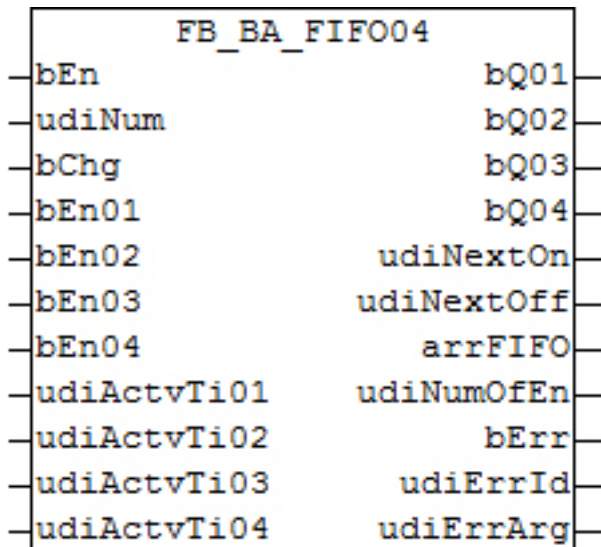
- *rSpOut* = *rSp* (Eingang) wenn der Baustein aktiviert ist
- *rSpOut* = 0 wenn der Baustein nicht aktiviert ist

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.87 FB_BA_FIFO04

Folgesteuerung von bis zu vier Aggregaten



Funktionsbeschreibung

Der Funktionsbaustein FB_BA_FIFO04 ermöglicht eine Folgesteuerung von bis zu vier Aggregaten mit automatischem Wechsel der Einschaltreihenfolge nach Betriebsstunden.

Der Funktionsbaustein steht in zwei Ausführungen zur Verfügung: für eine Sequenz von vier und von acht [► 222] Aggregaten.

In der Abfolge haben die Aggregate mit den wenigen Betriebsstunden Vorrang vor den mit vielen Betriebsstunden.

Eine steigende Flanke an *bChg* erzwingt einen Folgewechsel. Die Aggregate mit den geringsten Betriebsstunden werden in der FIFO nach vorne gesetzt und somit vorrangig eingeschaltet.

In der Folge werden nur Aggregate eingetragen, welche an den Eingängen *bEn01*..*bEn04* frei gegeben sind.

udiNum gibt die Anzahl der angeforderten Aggregate an.

An den Eingängen *udiActvTi01* bis *udiActvTi04* werden die Betriebsstunden der Aggregate eingetragen.

Werden diese Eingänge alle konstant auf Null gesetzt, erfolgt der Folgewechsel nur in Abhängigkeit von *bChg* zyklisch gesteuert.

Dabei fällt immer das erste Aggregat aus dem FIFO heraus, die anderen werden aufgeschoben, und das erste Aggregat hinten in dem FIFO wieder angehängt. Somit rotieren die Aggregate in ihrer Folge.

Werden am Eingang *udiNum* mehr Aggregate angefordert als an den Eingängen *bEn01* bis *bEn04* zur Verfügung stehen, wird dieses an *bErr* mit einem TRUE signalisiert.

Fehlerbehandlung

Werden am Eingang *udiNum* mehr Aggregate angefordert als an den Eingängen *bEn01* bis *bEn04* zur Verfügung stehen, wird dieses an *bErr* mit einem TRUE signalisiert.

Ein-/Ausgänge

VAR_INPUT

```

bEn      : BOOL;
udiNum   : UDINT;
bChg     : BOOL;
bEn01    : BOOL;
bEn02    : BOOL;
bEn03    : BOOL;
bEn04    : BOOL;
udiActvTi01 : UDINT;

```

```

udiActvTi02 : UDINT;
udiActvTi03 : UDINT;
udiActvTi04 : UDINT;

```

bEn: Freigabe des Bausteins

udiMyNum: Anzahl der Aggregate

bChg: Folgewechsel erzwingen

bEn01...bEn04: Freigabe Aggregat 1...Freigabe Aggregat 4

udiActvTi01...udiActvTi04: Betriebsstunden Aggregat 1...Betriebsstunden Aggregat 4

VAR_OUTPUT

```

bQ01      : BOOL;
bQ02      : BOOL;
bQ03      : BOOL;
bQ04      : BOOL;
udiNextOn  : UDINT;
udiNextOff : UDINT;
arrFIFO    : ARRAY [1..4] OF UDINT;
udiNumOfEn : UDINT;
bErr       : BOOL;
udiErrId   : UDINT;
udiErrArg  : UDINT;

```

bQ01...bQ04: Schaltet Aggregat 1..4

udiNextOn: Nummer des Aggregates, welches als nächstes eingeschaltet wird.

udiNextOff: Nummer des Aggregates, welches als nächstes eingeschaltet wird.

arrFIFO: FIFO-Speicher als Feld

udiNumOfEn: Anzahl Teilnehmer in Abhängigkeit der einzelnen Freigaben

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

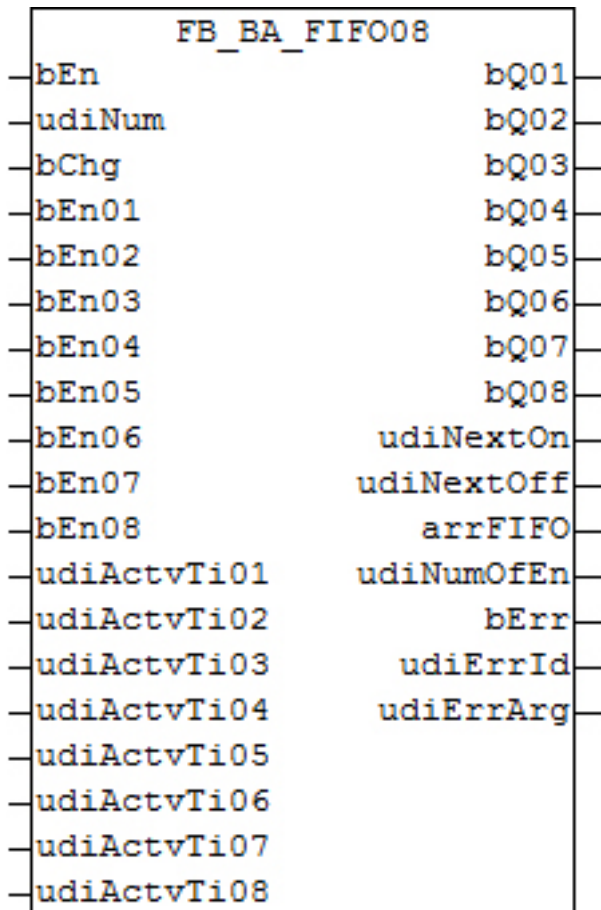
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.88 FB_BA_FIFO08

Folgesteuerung von bis zu acht Aggregaten



Funktionsbeschreibung

Der Funktionsbaustein FB_BA_FIFO08 ermöglicht eine Folgesteuerung von bis zu acht Aggregaten mit automatischem Wechsel der Einschaltreihenfolge nach Betriebsstunden.

Der Funktionsbaustein steht in zwei Ausführungen zur Verfügung: für eine Sequenz von [vier](#) [► 221] und von acht Aggregaten.

In der Abfolge haben die Aggregate mit den wenigen Betriebsstunden Vorrang vor den mit vielen Betriebsstunden.

Eine steigende Flanke an *bChg* erzwingt einen Folgewechsel. Die Aggregate mit den geringsten Betriebsstunden werden in der FIFO nach vorne gesetzt und somit vorrangig eingeschaltet.

In der Folge werden nur Aggregate eingetragen, welche an den Eingängen *bEn01*..*bEn08* frei gegeben sind. *udiNum* gibt die Anzahl der angeforderten Aggregate an.

An den Eingängen *udiActvTi01* bis *udiActvTi08* werden die Betriebsstunden der Aggregate eingetragen. Werden diese Eingänge alle konstant auf Null gesetzt, erfolgt der Folgewechsel nur in Abhängigkeit von *bChg* zyklisch gesteuert.

Dabei fällt immer das erste Aggregat aus dem FIFO heraus, die anderen werden aufgeschoben, und das erste Aggregat hinten in dem FIFO wieder angehängt. Somit rotieren die Aggregate in ihrer Folge.

Werden am Eingang *udiNum* mehr Aggregate angefordert als an den Eingängen *bEn01* bis *bEn08* zur Verfügung stehen, wird dieses an *bErr* mit einem TRUE signalisiert.

Fehlerbehandlung

Werden am Eingang *udiNum* mehr Aggregate angefordert als an den Eingängen *bEn01* bis *bEn08* zur Verfügung stehen, wird dieses an *bErr* mit einem TRUE signalisiert.

Ein-/Ausgänge**VAR_INPUT**

```

bEn      : BOOL;
udiNum   : UDINT;
bChg    : BOOL;
bEn01   : BOOL;
bEn02   : BOOL;
bEn03   : BOOL;
bEn04   : BOOL;
bEn05   : BOOL;
bEn06   : BOOL;
bEn07   : BOOL;
bEn08   : BOOL;
udiActvTi01 : UDINT;
udiActvTi02 : UDINT;
udiActvTi03 : UDINT;
udiActvTi04 : UDINT;
udiActvTi05 : UDINT;
udiActvTi06 : UDINT;
udiActvTi07 : UDINT;
udiActvTi08 : UDINT;

```

bEn: Freigabe des Bausteins

udiMyNum: Anzahl der Aggregate

bChg: Folgewechsel erzwingen

bEn01...bEn08: Freigabe Aggregat 1...Freigabe Aggregat 8

udiActvTi01...udiActvTi08: Betriebsstunden Aggregat 1...Betriebsstunden Aggregat 8

VAR_OUTPUT

```

bQ01    : BOOL;
bQ02    : BOOL;
bQ03    : BOOL;
bQ04    : BOOL;
bQ05    : BOOL;
bQ06    : BOOL;
bQ07    : BOOL;
bQ08    : BOOL;
udiNextOn : UDINT;
udiNextOff : UDINT;
arrFIFO  : ARRAY [1..8] OF UDINT;
udiNumOfEn : UDINT;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;

```

bQ01...bQ08: Schaltet Aggregat 1..8

udiNextOn: Nummer des Aggregates, welches als nächstes eingeschaltet wird.

udiNextOff: Nummer des Aggregates, welches als nächstes eingeschaltet wird.

arrFIFO: FIFO-Speicher als Feld

udiNumOfEn: Anzahl Teilnehmer in Abhängigkeit der einzelnen Freigaben

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

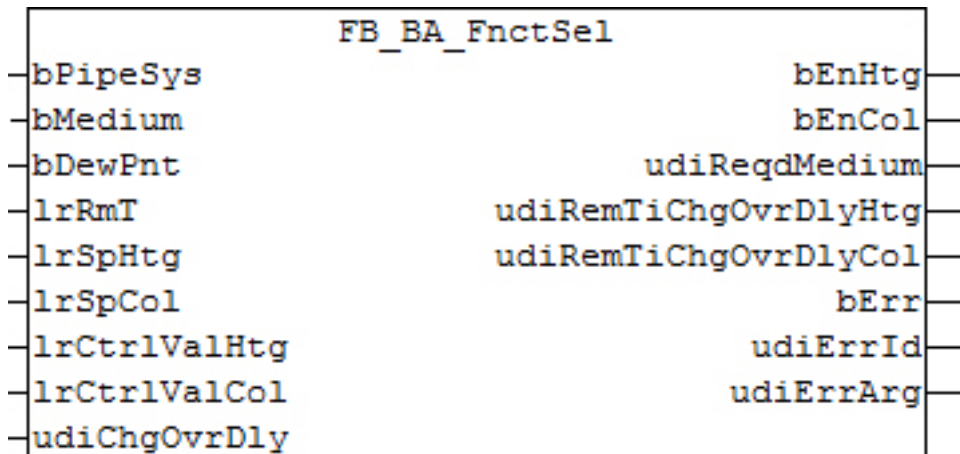
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.89 FB_BA_FunctSel

Automatische Anwahl Heiz- oder Kühlregler



Funktionsbeschreibung

Der Funktionsbaustein dient der Heiz-Kühlreglerfreigabe für einen Raum als Teil einer klimatechnischen Anlage. Dabei spielt die Art des Verteilungsnetzes eine große Rolle: handelt es sich um ein Zweileitersystem, so können alle Räume der Anlage immer nur entweder beheizt oder gekühlt werden. In einem Vierleitersystem hingegen kann die Klimatisierung der Räume bedarfsgerecht erfolgen, d.h. ein Teil der Räume kann beheizt, der andere von derselben Anlage gekühlt werden.

Der Funktionsbaustein, der wie bereits erwähnt pro Raum Anwendung findet, wählt damit je nach vorhandenem Rohrleitungssystem seine Regler unterschiedlich an:

Zweileiternetz

Das Zweileitersystem ist angewählt, wenn der Baustein am Eingang *bPipeSys* einen FALSE-Eintrag hat. Da alle Räume der Anlage nur entweder beheizt oder gekühlt werden können, wird die Anwahl von zentraler Stelle für alle Räume gleich über den Eingang *bMedium* vorgegeben. Ist *bMedium* FALSE, so wird der Raum-Heizregler angewählt - ist der Eingang TRUE, so ist es der Kühlregler. Die Reglerfreigaben *bEnHtg* und *bEnCol* werden immer gegenseitig verzögert um *uiChgOvrDly* [s] ausgegeben, das heißt: erst wenn die Freigabe Kühlen *bEnCol* für *uiChgOvrDly* auf FALSE stand, kann die Freigabe Heizen gegeben werden und umgekehrt. Zusätzlich zum Ablauf dieser Umschaltzeit wird darauf geachtet, dass der abzuschaltende Regler 0.0 ausgibt. Dies geschieht durch Rückmeldung an den Eingängen *lrCtrlValHtg* und *lrCtrlValCol*. So wird ein drastischer Wechsel von Heizen auf Kühlen und umgekehrt vermieden.

Vierleiternetz

Das Vierleitersystem ist angewählt, wenn der Baustein am Eingang *bPipeSys* einen TRUE-Eintrag hat. Die Regleranwahl kann nun für alle Räume nach Bedarf unterschiedlich sein und richtet sich daher nach der Raumtemperatur *lrRmT* und den eingestellten Sollwerten *lrSpHtg* für Heizen und *lrSpCol* für Kühlen. Übersteigt die Raumtemperatur den Kühlsollwert, so wird der Kühlregler freigegeben (*bEnCol*), fällt sie unter den Heizsollwert, wird der Heizregler freigegeben (*bEnHtg*). Befindet sich die Temperatur zwischen beiden Sollwerten, so sind beide Regler abgeschaltet (energieneutrale Zone). Auch hier werden die Reglerfreigaben *bEnHtg* und *bEnCol* immer gegenseitig verzögert um *uiChgOvrDly* [s] ausgegeben (siehe Zweileiternetz) und zusätzlich zum Ablauf dieser Umschaltzeit darauf geachtet, dass der abzuschaltende Regler 0.0 ausgibt. Dies geschieht durch Rückmeldung an den Eingängen *lrCtrlValHtg* und *lrCtrlValCol*. So wird ein drastischer Wechsel von Heizen auf Kühlen und umgekehrt vermieden, sollte die Umschaltzeit nicht ausreichend hoch bemessen sein.

Taupunktwärter (*bDewPnt*)

In beiden Systemen - Zwei- und Vierleiter - hat der Taupunktwärter die Aufgabe das Kühlen unmittelbar zu verriegeln. Da es sich hierbei um eine Sicherheitsfunktion handelt, ist der betreffende Eingang *bDewPnt* nach dem Ruhestromprinzip, also invertiert ausgelegt: *bDewPnt* = FALSE: Kühlregler ist verriegelt.

Programmablauf

Der Baustein kennt 3 Zustände:

1. Warten auf Freigabe Heizen bzw. Kühlen
2. Freigabe Heizen
3. Freigabe Kühlen

Im ersten Schritt wartet der Baustein auf die Erfüllung der Bedingungen, die für das Heizen bzw. das Kühlen notwendig sind:

Heizen	Kühlen
Ausgabe Kühlregler = 0 ($IrCtrlValCol$)	Ausgabe Heizregler = 0 ($IrCtrlValHtg$)
Raumtemperatur ($IrRmT$) < Sollwert Heizen ($IrSpHtg$)	Raumtemperatur ($IrRmT$) > Sollwert Kühlen ($IrSpCol$)
Reglerfreigabe Kühlen ($bEnCol$) steht mindestens für die Umschaltzeit $udiChgOvrDel$ [s] auf FALSE	Reglerfreigabe Heizen ($bEnHtg$) steht mindestens für die Umschaltzeit $udiChgOvrDel$ [s] auf FALSE
Vierleitersystem ist angewählt ($bPipesys=TRUE$) ODER Zweileitersystem ist angewählt und es steht Heizmedium an ($bPipeSys=FALSE$ AND $bMedium=FALSE$)	Vierleitersystem ist angewählt ($bPipesys=TRUE$) ODER Zweileitersystem ist angewählt und es steht Kühlmedium an ($bPipeSys=FALSE$ AND $bMedium=TRUE$)
	Der Taupunktwärter (Ruhestromprinzip) spricht nicht an ($bDewPnt=TRUE$)

Ist nun eine Bedingungskette erfüllt, springt der Baustein in den jeweiligen Zustand Heizen oder Kühlen und beendet diesen erst, wenn der entsprechende Regler am Bausteineingang ($IrCtrlValHtg/IrCtrlValCol$) 0 ausgibt. Damit ist sicher gestellt, dass immer nur ein Regler im Eingriff ist, auch wenn eine zu hohe Reglerausgabe, z. B. Heizen, u.U. ein kurzfristiges Kühlen erforderlich machen würde (Überschwingen). Das Heizen, bzw. Kühlen wird dann erst abgeschlossen, wenn kein Bedarf mehr ansteht.

Allerdings gibt es 3 Ausnahmen, bei denen das Heizen bzw. Kühlen unmittelbar unterbrochen wird:

1. Im Zweileitersystem ($bPipeSys=FALSE$) wird geheizt ($bEnHtg$), es wurde jedoch auf Kühlmedium umgestellt $bMedium=TRUE$
2. Im Zweileitersystem ($bPipeSys=FALSE$) wird gekühlt ($bEnCol$), es wurde jedoch auf Heizmedium umgestellt $bMedium=FALSE$
3. Im Kühlbetrieb (Zwei- oder Vierleitersystem) hat der Taupunktwärter angesprochen ($bDewPnt=FALSE$)

In diesen Fällen werden die Heiz- und Kühlfreigaben weg genommen und zurück in den Wartezustand gesprungen.

Bedarfsmitteilung ($udiReqdMedium$)

Um der Anlage mitzuteilen, wie der aktuelle Bedarf an Wärme oder Kühlung ist, wird am Bausteinausgang, also pro Raum, in Abhängigkeit der Ist- und Solltemperatur eine Bedarfskennung ausgegeben. Diese kann zentral gesammelt und ausgewertet werden. Die Auswertung erfolgt immer, also unabhängig vom vorhandene Netzsystem (Zwei- oder Vierleiter).

$udiReqdMedium$	Medium	Raumtemperatur
1	kein Medium ist angefordert	$rTRm > SpHtg$ UND $rRTm < SpCol$
2	Heizmedium ist angefordert	$rTRm < SpHtg$
3	Kühlmedium ist angefordert	$rTRm > SpCol$

Fehlerbehandlung

Der Heiz-Sollwert darf nicht größer oder gleich dem Kühl-Sollwert sein, da damit ein Temperaturbereich entstehen würde, in dem gleichzeitig Heiz- und Kühlbedarf entsteht. Da der Baustein jedoch immer nur eine Freigabe, Heizen oder Kühlen ausgibt, ist der Fall anlagentechnisch unbedenklich. Es wird für diesen Fall nur eine Warnmeldung ($bErr=TRUE$, $udiErrId=gBA_WarnIdFnctSel$) ausgegeben - der Baustein unterbricht seine Abarbeitung nicht.

Ein-/Ausgänge

VAR_INPUT

```
bPipeSys      : BOOL;
bMedium       : BOOL;
bDewPnt      : BOOL;
lrRmT        : LREAL;
lrSpHtg      : LREAL;
lrSpCol      : LREAL;
lrCtrlValHtg : LREAL;
lrCtrlValCol : LREAL;
udiChgOvrDel : UINT;
```

bPipeSys: Bei Zweileitersystem ist *bPipeSys* FALSE, bei Vierleitersystem TRUE

bMedium: Aktuelle Versorgung des gesamten Zweileiternetzes mit Kühl- oder Heizmittel. Wenn Heizmedium ansteht, ist *bMedium* FALSE.

bDewPnt: Taupunktwärter im **Ruhestromprinzip** (negative Logik): Wenn *bDewPnt* = FALSE, dann wird der Kühlregler gesperrt.

lrTRm: Raumtemperatur

lrSpHtg: Heiz-Sollwert

lrSpCol: Kühl-Sollwert

lrCtrlValHtg: Aktueller Ausgabewert vom Heizregler. Wird intern als Umschaltkriterium von Heizen auf Kühlen verwendet: *lrCtrlValHtg* muss 0 sein.

lrCtrlValCol: Aktueller Ausgabewert vom Kühlregler. Wird intern als Umschaltkriterium von Kühlen auf Heizen verwendet: *lrCtrlValCol* muss 0 sein.

udiChgOvrDel: Umschaltverzögerung [s] von Heizen auf Kühlen und umgekehrt

VAR_OUTPUT

```
bEnHtg      : BOOL;
bEnCol      : BOOL;
udiReqdMedium : UDINT;
udiRemTiChgOvrDlyHtg : UDINT;
udiRemTiChgOvrDlyCol : UDINT;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bEnHtg: Freigabe des Reglers Heizen

bEnCol: Freigabe des Reglers Kühlen

udiReqdMedium:

udiReqdMedium	Medium	Raumtemperatur
1	kein Medium ist angefordert	rTRm > SpHtg UND rRTm < SpCol
2	Heizmedium ist angefordert	rTRm < SpHtg
3	Kühlmedium ist angefordert	rTRm > SpCol

udiRemTiChgOvrDlyHtg: Countdown [s] Umschaltverzögerung von Kühlen auf Heizen

udiRemTiChgOvrDlyCol: Countdown [s] Umschaltverzögerung von Heizen auf Kühlen

bErr: Bei fehlerhafter Abarbeitung, bzw. bei Warnzuständen wird dieser Ausgang auf TRUE gesetzt.

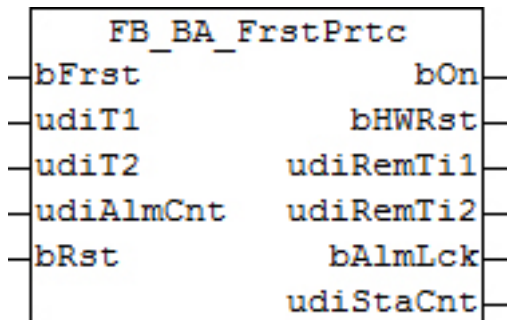
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2244	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.90 FB_BA_FrstPrtc

Überwachung Frostalarm und Noterhitzung mit 2 Zeitüberwachungen: T1 Alarm geht wieder und T2 Alarm kommt nicht erneut.



Funktionsbeschreibung

Der Funktionsbaustein dient zur Frostüberwachung eines Heizregister in einer raumlufttechnischen Anlage.

Frostgefahr steht an, wenn der Eingang *bFrst* TRUE ist. Der Frostalarm muss im Anlagenprogramm so verknüpft werden, dass die Anlage sofort abschaltet, das Erhitzervertil öffnet und die Erhitzerpumpe eingeschaltet wird.

Bei Frostgefahr wird der Ausgang *bOn* gesetzt und der Timer T1 gestartet. Ist nach Ablauf von *udiT1* (Sekunden) nach wie vor Frostgefahr (*bFrst*=TRUE) dann bleibt *bOn* gesetzt. Ein Rücksetzen kann nur an dem Eingang *bRst* erfolgen.

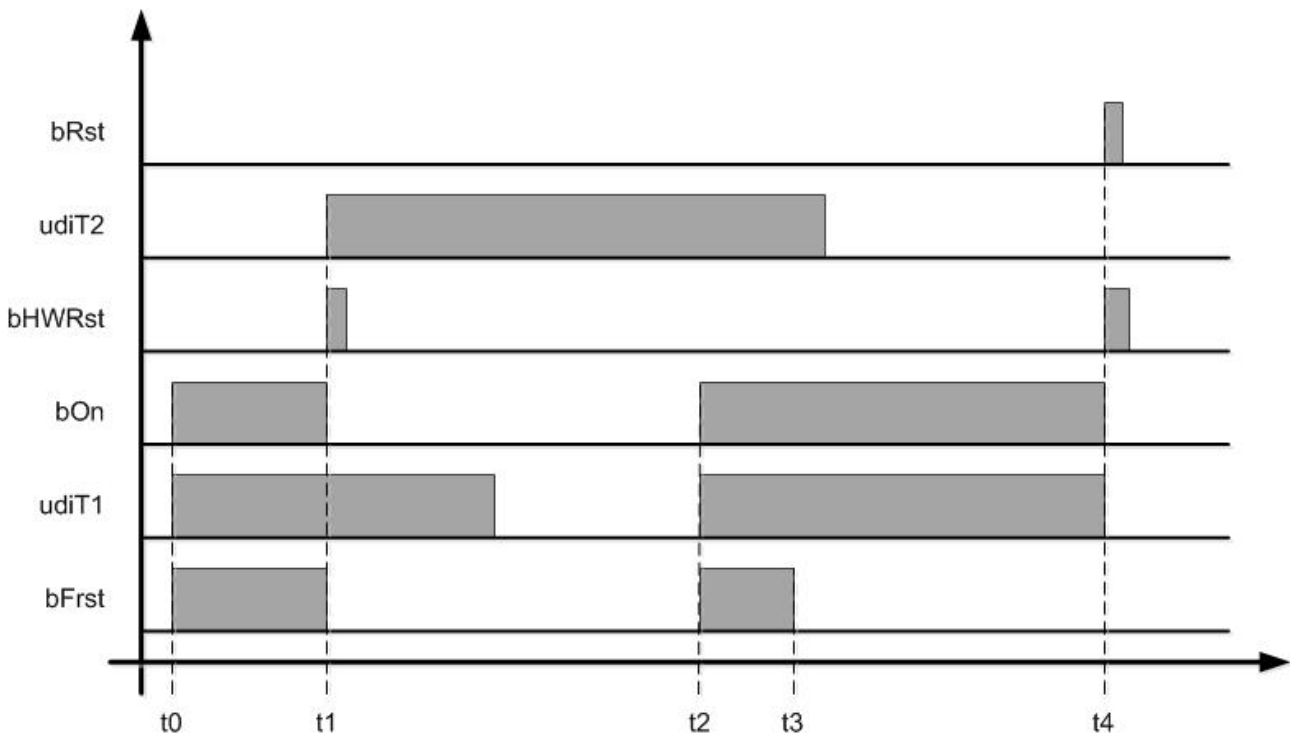
Erlischt der Frostalarm durch das Aufheizen des Heizregisters innerhalb des Zeitraums von *udiT1* (*bFrst*=FALSE), so erfolgt ein automatischer Neustart der Anlage. Für den Anlagenneustart wird *bOn* FALSE und am Ausgang *bHWRst* wird ein Impuls zur Quittierung einer Selbsthaltungsschaltung im Schaltschrank ausgegeben. Mit dem Neustart wird der Timer T2 gestartet. Erfolgt innerhalb des Zeitraums von *udiT2* (Sekunden) erneut ein Frostalarm wird die Anlage endgültig verriegelt. *bOn* bleibt so lange gesetzt bis der Frostalarm beseitigt und an *bRst* quittiert ist.

Falls die Frostalarme aufeinander folgend, aber immer mit einem Zeitversatz größer als die Zeit von *udiT2* erfolgen, würde theoretisch beliebig oft automatisch neu gestartet werden. Um dieses zu vermeiden werden die Neustarts innerhalb des Bausteins gezählt. Mit *udiAlmCnt* ist die Zahl der möglichen automatischen Neustarts zwischen 0 und 4 einstellbar.

Eine Quittierung am Eingang *bRst* setzt den Alarmspeicher innerhalb des Bausteins auf Null zurück.

Beispiel:

t0 = Frostalarm an Eingang *bFrst*, Alarmmeldung an Ausgang *bOn*, Starten des Timers T1 (*udiT1* [s])
 t1 = Frostalarm erloschen, Rücksetzen von *bOn*, Ausgabe Hardwareimpuls, Starten von Timer T2 (*udiT2* [s]), Anlagenneustart
 t2 = Erneuter Frostalarm innerhalb von T2, Alarmmeldung an *bOn*, Starten des Timers T1, verriegeln des Frostalarms
 t3 = Frostalarm geht.
 t4 = Quittierung des Alarms an *bRst*, Rücksetzen von *bOn*.



Ein-/Ausgänge

VAR_INPUT

```
bFrst      : BOOL;
udiT1      : UDINT;
udiT2      : UDINT;
udiAlmCnt  : UDINT;
bRst       : BOOL;
```

bFrst: Anschluss für die Frostereignisse luft- und wasserseitig

udiT1: Timer zur Verzögerung von Neustarts [s]

udiT2: Timer Überwachungszeit [s]

udiAlmCnt: Maximale Zahl der automatischen Anlagenneustarts ohne Rücksetzen. Die maximal mögliche Eingabe ist 4 und wird intern begrenzt.

bRst: Rücksetzen bzw. quittieren des Frostalarms

VAR_OUTPUT

```
bOn        : BOOL;
bHWRst     : BOOL;
udiRemTi1  : UDINT;
udiRemTi2  : UDINT;
bAlmLck    : BOOL;
udiStaCnt  : UDINT;
```

bOn: Frostalarm aktiv

bHWRst: Ausgabe eines Impulses zur Quittierung der Frostschutzhardware

udiRemTi1: Verbleibende Zeit bis Anlagenneustart nach Frostalarm

udiRemTi2: Verbleibende Überwachungszeit

bAlmLck: Alarm Lock - gespeicherter Alarm

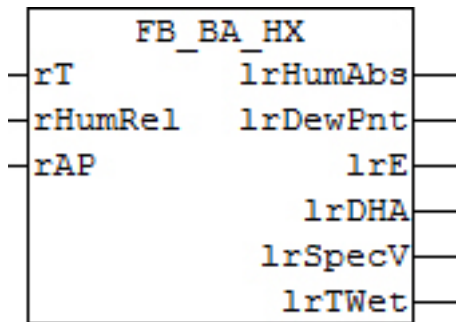
udiStaCnt: Status Counter – akt. Anzahl nicht quittierter Fehlstarts

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.91 FB_BA_HX

Enthalpieberechnung

**Funktionsbeschreibung**

Mit diesem Funktionsbaustein werden die Taupunkttemperatur, die spezifische Enthalpie und die absolute Feuchte berechnet. Für die Berechnung der Größen werden die Temperatur, die relative Feuchte und der barometrische Luftdruck benötigt.

Die Enthalpie ist ein Maß für die Energie eines thermodynamischen Systems.

Ein-/Ausgänge**VAR_INPUT**

```
rT      : REAL;
rHumRel : REAL;
rAP     : REAL;
```

rT: Temperatur [°C]

rHumRel: Relative Feuchte [%]

rAP: Hydrostatischer Luftdruck 1013.25 hPa

VAR_OUTPUT

```
lrHumAbs : LREAL;
lrDewPnt : LREAL;
lrE      : LREAL;
lrDHA    : LREAL;
lrSpecV  : LREAL;
lrTWet   : LREAL;
```

lrHumAbs: Absolute Feuchte g Wasser je kg trockener Luft [g/Kg]

lrDewPnt: Taupunkttemperatur [°C]

lrE: Enthalpie [kJ/kg]

lrDHA: Dichte feuchter Luft ρ [kg Gemisch/m³]

lrSpecV: spezifisches Volumen [m³/kg]

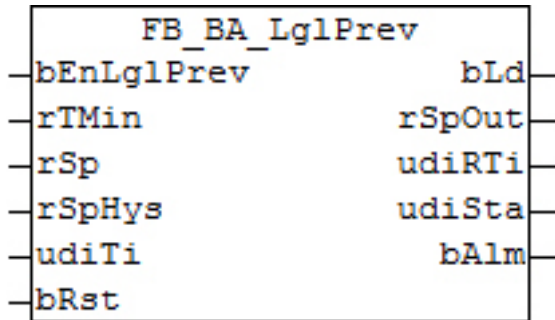
lrTWet: Feuchtkugeltemperatur [°C]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.92 FB_BA_LglPrev

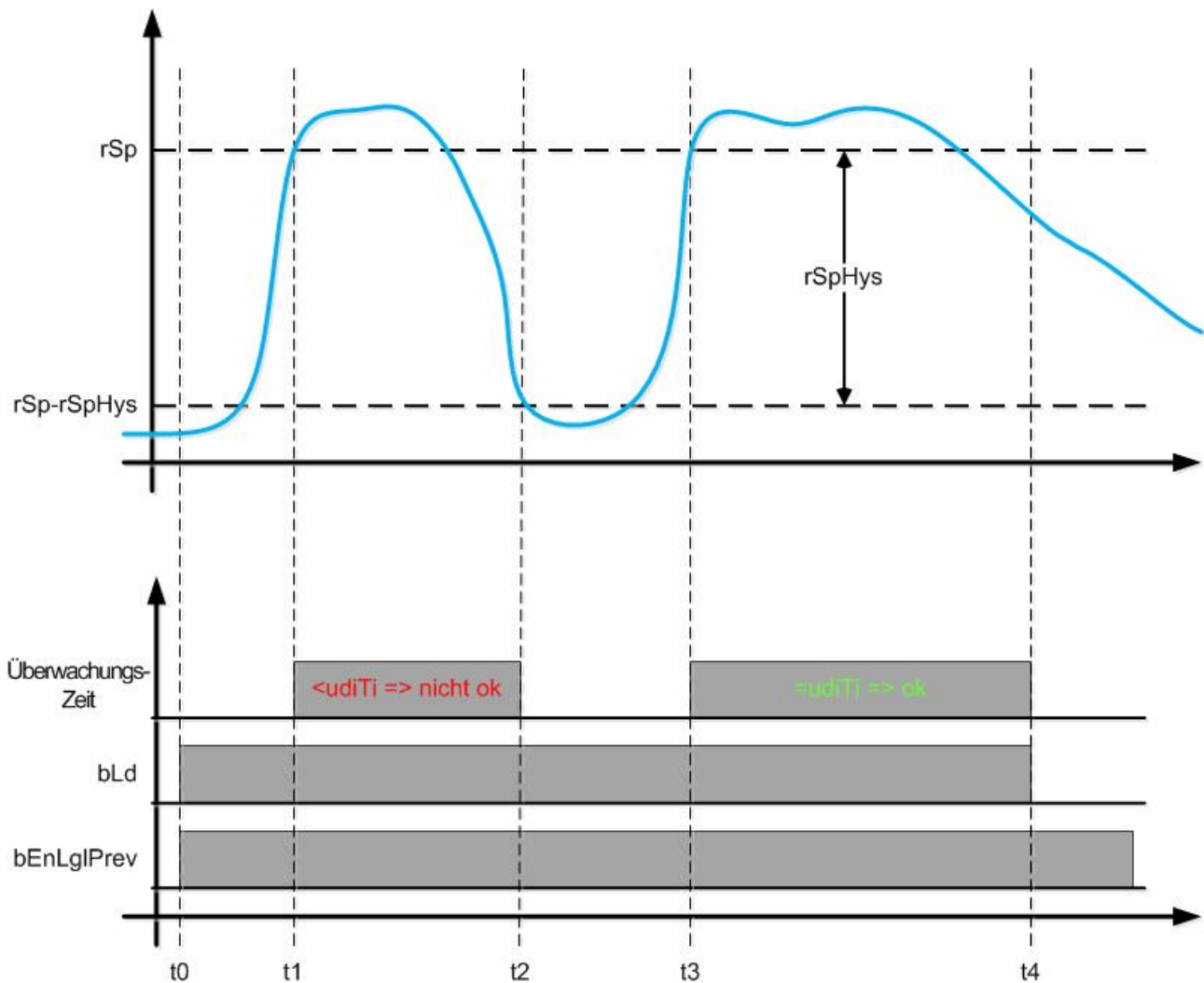
Legionellenschutz



Funktionsbeschreibung

Der Funktionsbaustein dient zur Desinfektion des Brauchwassers und zum Abtöten von Legionellen. Die Freigabe des Desinfektionsbetriebes erfolgt an dem Eingang *bEnLglPrev* durch die Anschaltung eines Zeitschaltprogrammes. Zu empfehlen ist die Desinfektion mindestens einmal in der Woche nachts durchzuführen. Die Temperatur sollte mindestens 70°C sein. Das Einschaltintervall an *bEnLglPrev* muss dafür ausreichend lang gewählt werden. Der Ausgang *bLd* aktiviert dabei die Speicherladung. Bei Warmwasserspeichern mit zwei Temperaturfühler muss an *rTMin* eine Min-Auswahl beider Fühler angeschlossen werden.

Überschreitet *rTMin* den Wert von *rSp*, wird ein Überwachungstimer mit einer Zeit von *udiTi* [s] gestartet. Bleibt die minimale Speichertemperatur *rTMin* bis zum Ablauf des Timers oberhalb von *rSp-rSpHys* ist eine ausreichende Aufheizung des Speichers erfolgt. Bei einer vorhandenen Zirkulation muss der Ausgang *bLd* mit der Freigabe der Zirkulationspumpe verknüpft sein damit auch die Wasserleitung innerhalb des Brauchwassersystems desinfiziert wird. Ist die Temperatur jedoch während es Desinfektionsprozesses unter *rSp-rSpHys* gefallen muss der Desinfektionsprozess neu gestartet werden bis die Zeit von *udiTi* einmal komplett abgelaufen ist. Bei einer erfolgreichen Desinfektion wird der Ausgang *bLd* zurück gesetzt. Wurde während der Bausteinaktivierung (*bEnLglPrev*) kein vollständiger Desinfektionsprozess abgeschlossen, wird dieses mit dem Ausgang *bAlm* signalisiert. Der Ausgang muss mit dem Eingang *bRst* zurückgesetzt werden.



Erläuterung zur Grafik:

t_0 Starten des Legionellenprogramms und Schalten des Ausgangs bLd . Aufheizen der Warmwasserspeichers.

t_1 Der Speicher hat die Temperatur rSp erreicht. Der Timer für die Aufheizzeit wird gestartet.

t_2 Die minimale Speichertemperatur ist unterhalb von $rSp - rSpHys$ gefallen. Der Timer für die Aufheizzeit wird wieder zurück gesetzt.

t_3 Die Temperatur steigt erneut über rSp und der Aufheiztimer wird wieder gestartet.

t_4 Die Minimale Speichertemperatur war über den Zeitraum von $udiTi$ hinweg oberhalb der Grenze $rSp - rSpHys$, die Desinfektion war erfolgreich. bLd wird zurück gesetzt und der Warmwasserspeicher geht wieder in den Normalbetrieb.

Ein-/Ausgänge

VAR_INPUT

```

bEnLglPrev : BOOL;
rTMin      : REAL;
rSp        : REAL;
rSpHys     : REAL;
udiTi      : UDINT;
bRst       : BOOL;

```

bEnLglPrev: Freigabe des Desinfektionsbetriebes über ein Zeitschaltprogramm

rTMin: Minimale Speichertemperatur [°C]. Minimalauswahl von Temperaturfühler oben und Temperaturfühler unten.

rSp: Sollwert Desinfektion [°C]

rSpHys: Temperaturspreizung [°K] untere Grenze - wird immer absolut gerechnet

udiTi: Überwachungszeitraum [s]

bRst: Rücksetzen des Legionellenalarms

VAR_OUTPUT

```
bLd      : BOOL;
rSpOut   : REAL;
udiRTi   : UDINT;
udiSta   : UDINT;
```

bLd: Anti-Legionellenbetrieb aktiv

rSpOut: Sollwertweitergabe an Ladeschaltung:

- rSp (Eingang) wenn der Baustein aktiviert ist
- 0 wenn der Baustein nicht aktiviert ist

udiRTi: Countdown Desinfektions-Betriebstimer

udiSta: Status Desinfektionsprogramm:

1. Der Desinfektionsbetrieb war erfolgreich.
2. Die Desinfektion ist erfolgreich abgeschlossen. Zum Abschluss und für eine neue Aktivierung der Legionellenvorsorge muss *bEnLglPrev* FALSE sein.
3. Der Desinfektionsbetrieb ist aktiv.
4. Desinfektion war nicht erfolgreich. Alarm steht an.
5. Desinfektion war nicht erfolgreich, der Alarm wurde quittiert.
6. Bedeutet ein Neustart der Steuerung bzw. es wurde noch kein Legionellenbetrieb angefordert.

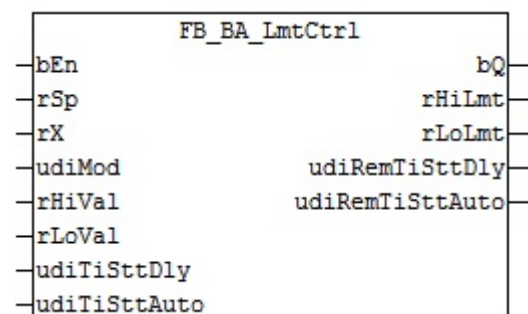
bAlm: Temperatursollwert wurde nicht durchgängig über den Zeitraum von *udiTime* hinweg erreicht, so dass eine ausreichende Desinfektion nicht gewährleistet ist.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.93 FB_BA_LmtCtrl

Baustein zur Ermittlung der Grenzwerte und dessen Freigabe



Funktionsbeschreibung

Der Funktionsbaustein dient zur Ermittlung von Grenzwerten ($rHiLmt$ / $rLoLmt$) in Abhängigkeit der eingestellten Betriebsart und dessen Freigabe (bQ).

Der Funktionsbaustein kann für folgende Bedingungen eingesetzt werden, z. B. bei einem Anlagenstillstand, im Moment des Starts und bis zum eingeregelter Zustand einer Anlage. In diesen Phasen soll das Melden z. B. eines Loop-Objektes unterdrückt werden damit keine falschen Meldungen an die MBE gesendet werden. Erst wenn die RLT-Anlage vollständig hochgefahren ist und die Regelung sich eingependelt hat, soll das Melden des Loop-Objektes freigegeben werden.

Wichtig ist, dass nur über = FALSE zurück gesetzt werden kann. bQ bEn

Diagramm 1

Diagramm 1

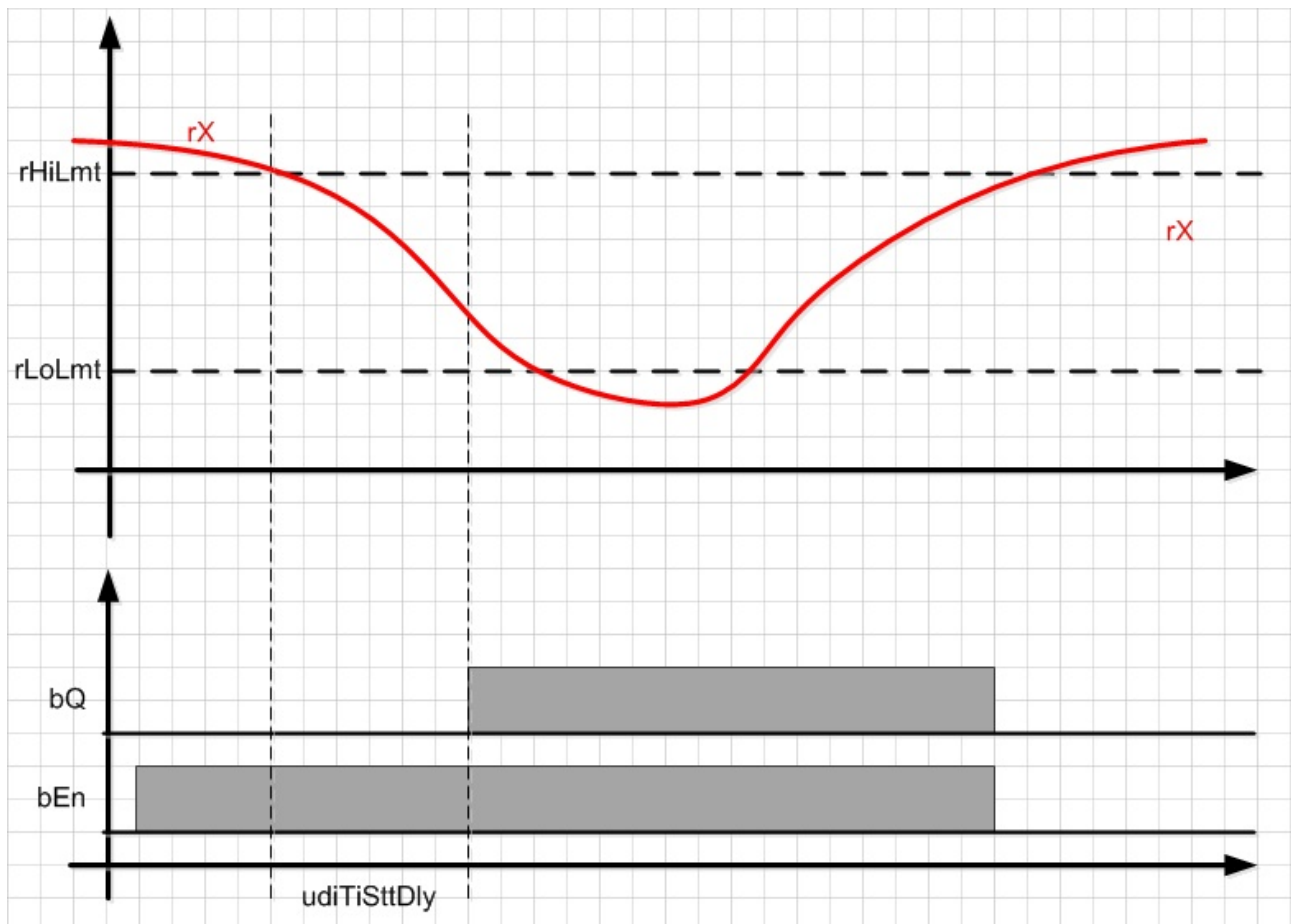
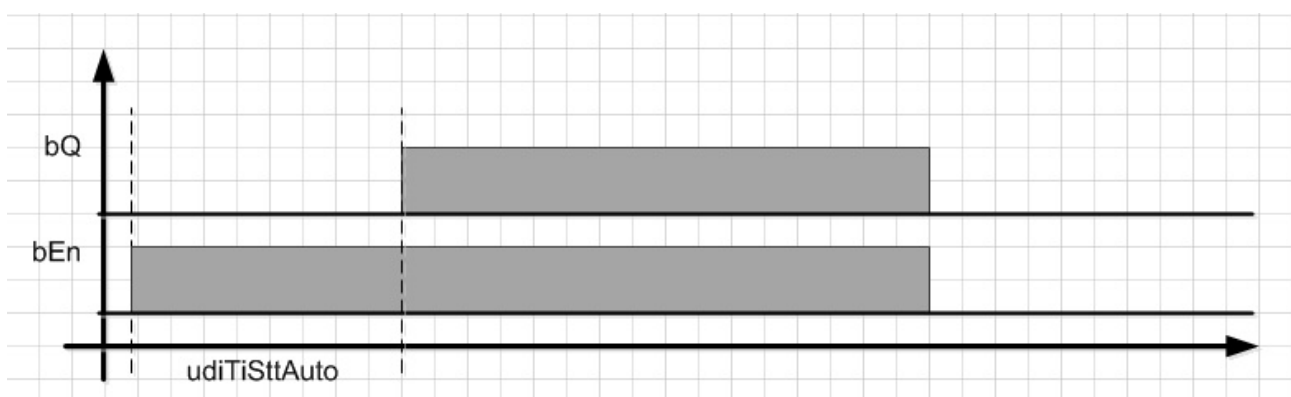


Diagramm 2

Diagramm 2



Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
rSp      : REAL;
rX       : REAL;
udiOpMod : UDINT;
rHiVal   : REAL;
rLoVal   : REAL;
udiTiSttDly : UDINT; [s]
udiTiSttAuto : UDINT; [s]
```

bEn: Freigabe Baustein. Ist $bQ = TRUE$, so kann der Ausgang nur zurück gesetzt werden, wenn bEn auf FALSE gesetzt wird.

rSp: Sollwert

rX: Istwert

udiOpMod: Vorgabe Betriebsart. Es kann zwischen 3 Betriebsarten gewählt werden. Daraus resultiert dann:

udiOpMod = 1 = fest/fixed: $rHiLmt = rHiVal$ und $rLoLmt := rLoVal$. **Wichtig ist, dass $rLoVal$ nicht größer gleich $rHiVal$ sein darf. Ist dieses der Fall, dann ist $rHiLmt=rHiVal$ und $rLoLmt:=rHiVal-0,05$**

udiOpMod = 2 = gleitend absolut/sliding absolut: Es wird über die Funktion ABS der Absolutwert für die beiden Eingangsvariablen $rHiVal / rLoVal$ bestimmt. $rHiLmt = rSp + rHiVal$ und $rLoLmt = rSp - rLoVal$

udiOpMod = 3 = gleitend prozentual/sliding percent: Es wird über die Funktion ABS der Absolutwert für die beiden Eingangsvariablen $rHiVal / rLoVal$ bestimmt. Die Formeln zur Bestimmung der Grenzwerte lauten: $rHiLmt = rSp + ABS(((rSp \times rHiVal) / 100))$; $rLoLmt = rSp - ABS(((rSp \times rLoVal) / 100))$

rHiVal: Oberer Wert zur Berechnung des oberen Grenzwerts $rHiLmt$. Die Berechnung des oberen Grenzwert ist abhängig von der Betriebsart $udiOpMod$.

rLoVal: Unterer Wert zur Berechnung des unteren Grenzwerts $rLoLmt$. Die Berechnung des unteren Grenzwert ist abhängig von der Betriebsart $udiOpMod$.

udiTiSttDly: Startverzögerung für bQ , wenn Funktion erfüllt ist, siehe [Diagramm 1](#) [▶ 234].

udiTiSttAuto: Automatische Startverzögerung. Ist $bEn = TRUE$, so wird bQ nach Ablauf der automatischen Zeitverzögerung TRUE, siehe [Diagramm 2](#) [▶ 234].

VAR_OUTPUT

```
bQ      : BOOL;
rHiLmt  : REAL;
rLoLmt  : REAL;
udiRemTiSttDly : UDINT;
udiRemTiSttAuto : UDINT;
```

bQ: Grenzwertüberwachung ist frei gegeben. bQ kann nur über $bEn = FALSE$ zurück gesetzt werden.

rHiLmt: Oberer Grenzwert für die Freigabe der Grenzwertüberwachung. Der Wert ist abhängig von der Betriebsart $udiOpMod$

rLoLmt : Unterer Grenzwert für die Freigabe der Grenzwertüberwachung. Der Wert ist abhängig von der Betriebsart $udiOpMod$

udiRemTiSttDly: Countdown der Startverzögerung $udiTiSttDly$

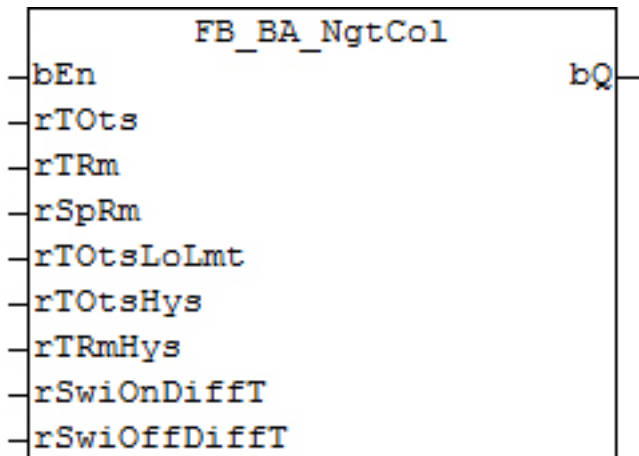
udiRemTiSttAuto: Countdown der Automatischen Startverzögerung $udiTiSttAuto$

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.94 FB_BA_NgtCol

Sommernachtkühlung



Funktionsbeschreibung

Dieser Funktionsbaustein wird verwendet um nachts mit kühler Außenluft die tags zuvor aufgeheizten Räume herunter zu kühlen. Die Funktion der Sommernachtkühlung dient zur Verbesserung der Luftqualität und zur Einsparung elektrischer Energie. In den ersten Stunden des nächsten Sommertages wird elektrische Energie zur Kälteerzeugung gespart.

Durch Parametrierung des Funktionsbausteins FB_BA_NgtCol werden die Startbedingungen für die Sommernachtkühlung definiert. Der Baustein kann verwendet werden, um motorisch betätigte Fenster zu öffnen oder Klimaanlage außerhalb ihrer normalen Betriebszeiten in den Sommernachtkühlbetrieb zu schalten.

Folgende Bedingungen müssen für die Aktivierung der Sommernachtkühlung erfüllt sein:

- Der Baustein selbst ist aktiviert ($bEn = TRUE$).
- Die Außentemperatur ist nicht zu niedrig ($rTOts > rTOtsLoLmt$).
- Die Außentemperatur ist im Vergleich zur Raumtemperatur ausreichend niedrig ($rTRm - rTOts > rSwiOnDifft$).
- Die Raumtemperatur ist so hoch, so dass es sich lohnt die Sommernachtkühlung einzuschalten. $rTRm > rSpRm + rTRmHys$.

Bei den folgenden Bedingungen wird die Sommernachtkühlung deaktiviert:

- Der Baustein selbst ist deaktiviert ($bEn = FALSE$).
- Die Außentemperatur ist zu niedrig ($rTOts < rTOtsLoLmt$).
- Die Außentemperatur ist im Vergleich zur Raumtemperatur zu hoch ($rTRm - rTOts < rSwiOffDifft$).
- Die Raumtemperatur ist niedriger als der Sollwert. $rTRm \leq rSpRm$.

Ein-/Ausgänge

VAR_INPUT

```

bEn          : BOOL;
rTOts        : REAL;
rTRm         : REAL;
rSpRm        : REAL;
rTOtsLoLmt   : REAL;
rTOtsHys     : REAL;
rTRmHys      : REAL;
rSwiOnDifft  : REAL;
rSwiOffDifft : REAL;

```

bEn: Freigabe des Funktionsbausteins

rTOts: Außentemperatur [°C]

rTRm: Außentemperatur [°C]

rSpRm: Raumtemperatur-Sollwert

rTOtsLoLmt: Untere Grenze der Außentemperatur [°C], verhindert ein zu starkes Auskühlen.

rTOtsHys: Hysterese für die minimale Außentemperatur [°K]. Diese Hysterese, welche nach unten hin intern auf 0.5 °K begrenzt ist, soll ein Flattern von *bQ* verhindern, falls die Außentemperatur genau um den Wert von *rTOtsLoLmt* pendelt.

rTRmHys: Hysterese für die Raumtemperatur [°K]. Diese Hysterese, welche nach unten hin intern auf 0.5 °K begrenzt ist, soll ein unnötiges Schwanken von *bQ* verhindern, falls die Raumtemperatur genau um den Sollwert *rSpRm* pendelt.

rSwiOnDiffT: Differenz zwischen der Raumtemperatur und der Außentemperatur ab der die Sommernachtkühlung frei gegeben wird [°K].

rSwiOffDiffT: Differenz zwischen der Raumtemperatur und der Außentemperatur ab der die Sommernachtkühlung gesperrt wird [°K].

VAR_OUTPUT

bQ : BOOL;

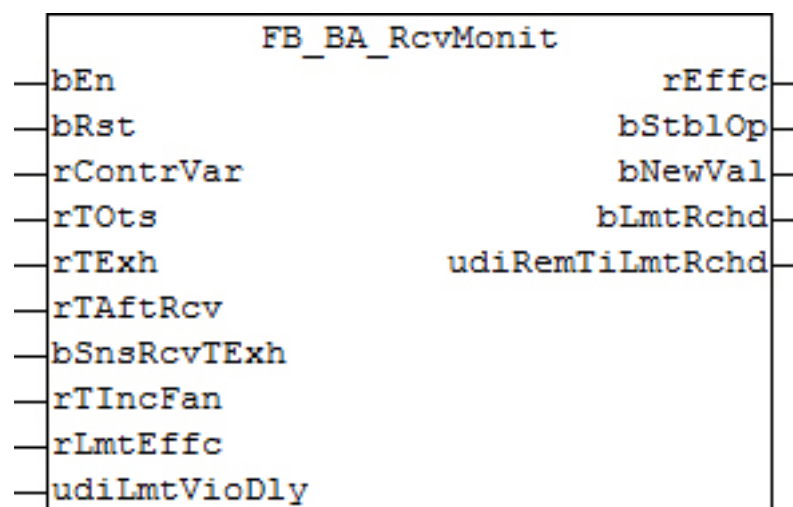
bQ: Sommernachtkühlung Ein

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.95 FB_BA_RcvMonit

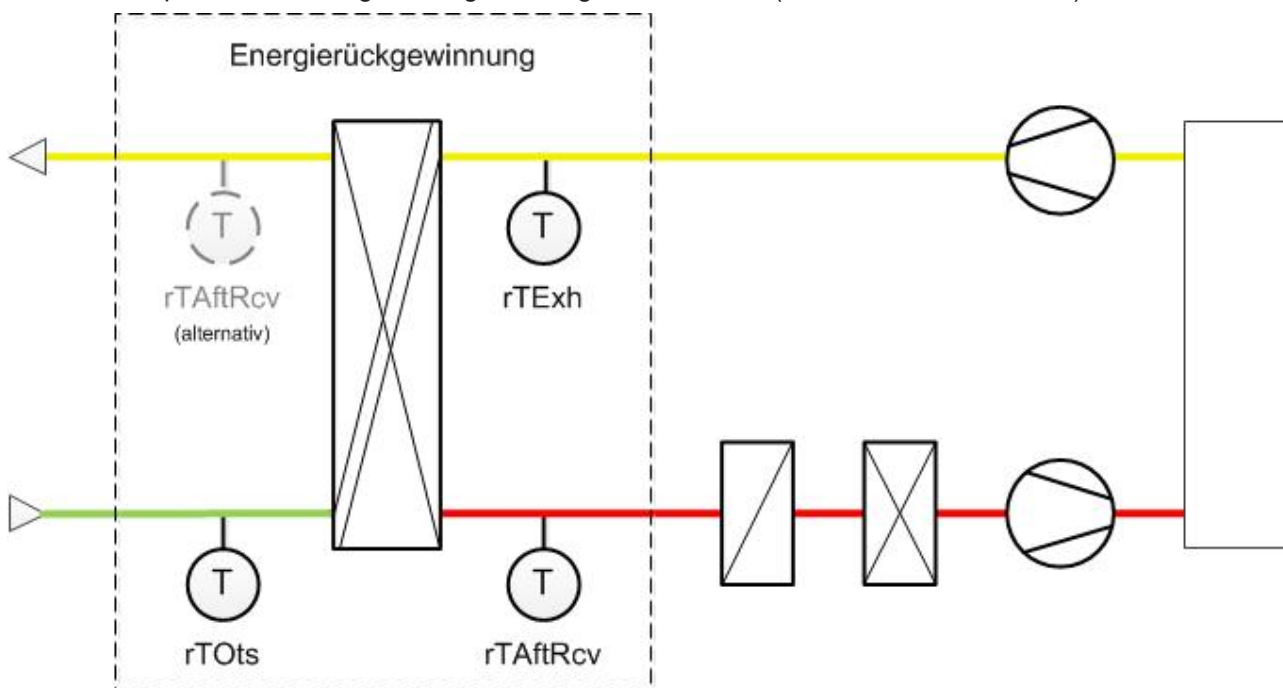
Funktionsbaustein zur Berechnung des Wirkungsgrades einer Energie-Rückgewinnung



Funktionsbeschreibung

Der Baustein benötigt zur Berechnung des Wirkungsgrades (der sogenannten Rückwärmzahl) folgende Temperaturmesswerte:

- Außenlufttemperatur $rTOts$
- Fortlufttemperatur $rTEh$
- Lufttemperatur der Energie-Rückgewinnung im Zuluftkanal (alternativ: im Abluftkanal) $rTAftRcv$



Der Baustein nimmt nun alle 10s die Temperatur-Messwerte auf und bildet aus jeweils 6 Werten den minütlichen Mittelwert. Danach wird beurteilt, ob die Anlage sich in einem eingeregelt, "stabilen" Zustand befindet:

- Die aufgenommenen Temperaturen Außenluft, Fortluft und Luft nach Energierückgewinnung sind annähernd konstant, d.h. keiner der 6 Einzelwerte weicht mehr als $0,5^{\circ}\text{K}$ vom jeweiligen Mittelwert ab.
- Die Temperaturdifferenz Außenluft zu Fortluft beträgt mindestens 5°K .

Ist dies der Fall, so wird dieser Messzyklus mit einem TRUE-Signal am Ausgang $bStblOp$ quittiert und der ermittelte Wirkungsgrad an $rEffc$ ausgegeben. Ist der Zustand hingegen nicht "stabil", so erscheint am Ausgang $bStblOp$ ein FALSE-Signal und $rEffc$ wird auf 0 gesetzt.

In jedem Fall jedoch wird jeder Mess- und Auswerte-Zyklus mit einem Trigger (ein Ein-PLC-Zyklus langes TRUE-Signal) an $bNewVal$ als beendet gekennzeichnet.

Freigabe (bEn) und Reset ($bRst$)

Der Baustein arbeitet nur dann, wenn ein TRUE-Signal an bEn anliegt. ansonsten stoppt er seine Abarbeitung und alle Ausgänge werden auf FALSE bzw 0.0 gesetzt.

Während der Abarbeitung kann ein Mess- und Auswertezyklus jederzeit durch ein TRUE-Signal an $bRst$ abgebrochen werden. Alle Ausgänge werden auf FALSE bzw. 0.0 gesetzt und der Messzyklus startet automatisch neu.

Auswahl des Temperaturwerts "nach Rückgewinnung" ($bSnsRcvTEh$)

Ein FALSE-Eintrag an $bSnsRcvTEh$ bedeutet, dass zur Berechnung des Wirkungsgrades die Temperaturmessung nach der Wärmerückgewinnung im **Zuluftkanal** verwendet wird. Soll hingegen die Temperaturmessung nach der Wärmerückgewinnung im **Abluftkanal** angewendet werden, so muss an $bSnsRcvTEh$ ein TRUE angelegt werden.

Grenzwertüberschreitung ($rContrVar$, $rLmtEffc$, $bLmtRchd$)

Eine Grenzwertüberschreitung liegt dann vor, wenn der ermittelte Wirkungsgrad kleiner ist als der eingetragene Grenzwert $rLmtEffc$ und gleichzeitig der Stellgröße für die Wärmerückgewinnung bei 100% liegt. Dazu muss die Stellgröße mit dem Eingang $rContrVar$ verbunden werden.

Die Meldung der Grenzwertüberschreitung lässt sich durch einen Eintrag an $udiLmtVioDly$ verzögern: Liegen

die beiden Kriterien, Überschreitung und Übersteuerung länger als *udiLmtVioDly* [s] vor, so wird dieses mit einem TRUE-Signal an *bLmtRchd* angezeigt. Das Auftreten lässt sich am Countdown-Ausgang *udiRemTiLmtRchd* verfolgen.

Eine aufgetretene Warnmeldung erlischt, wenn ein kompletter Messzyklus "gute" Werte erbringt oder bei einer steigenden Flanke an *bRst* bzw einer Deaktivierung des Bausteins.



Diese Warnmeldung kann nur zustande kommen, wenn die Anlage in einem stabilen Betrieb ist (*bStblOp*=TRUE).

Berücksichtigung Temperaturanstieg der Abluft durch Ventilatormotor (*rTIncFan*)

Es ist möglich, dass die Abluft durch einen Ventilatormotor erwärmt und die Messung damit verfälscht wird. Diese Temperaturerhöhung kann durch den Parameter *rTIncFan* angegeben werden. Intern wird dann die gemessene Ablufttemperatur um diesen Wert gemindert.

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
bRst     : BOOL;
rContrVar : REAL;
rTOts    : REAL;
rTExh    : REAL;
rTAftRcv : REAL;
bSnsRcvTExh : BOOL;
rTIncFan : REAL;
rLmtEffc : REAL;
udiLmtVioDly : UDINT;
```

bEn: Freigabe des Bausteins

bRst: Reset - alle ermittelten Werte werden gelöscht

rContrVar: Stellgröße für die Wärmerückgewinnung, d.h. ihr Istwert

rTOts: Außentemperatur

rTExh: Fortlufttemperatur

rTAftRcv: Temperatur nach Energierückgewinnung

bSnsRcvTExh: Messort Temperatur nach Energierückgewinnung: FALSE -> im Zuluftkanal (SupplyAir) - TRUE -> im Abluftkanal (ExhaustAir).

rTIncFan: Temperaturanstieg durch Ventilator

rLmtEffc: Wirkungsgrad Grenzwert

udiLmtVioDly: Zeitverzögerung Grenzwertüberschreitung [s]

VAR_OUTPUT

```
rEffc    : REAL;
bStblOp  : BOOL;
bNewVal  : BOOL;
bLmtRchd : BOOL;
udiRemTiLmtRchd : UDINT;
```

rEffc: Effektivität

bStblOp: Stabiler Betrieb

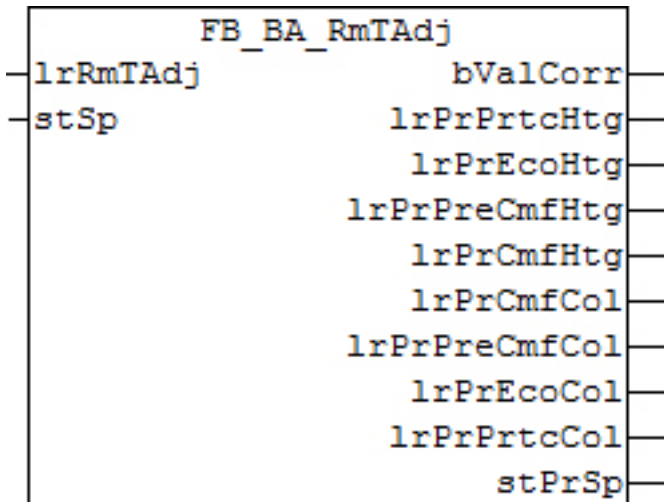
bNewVal: Ausgabetrigger neuer Wert *rEffc*

bLmtRchd: Grenzwert erreicht

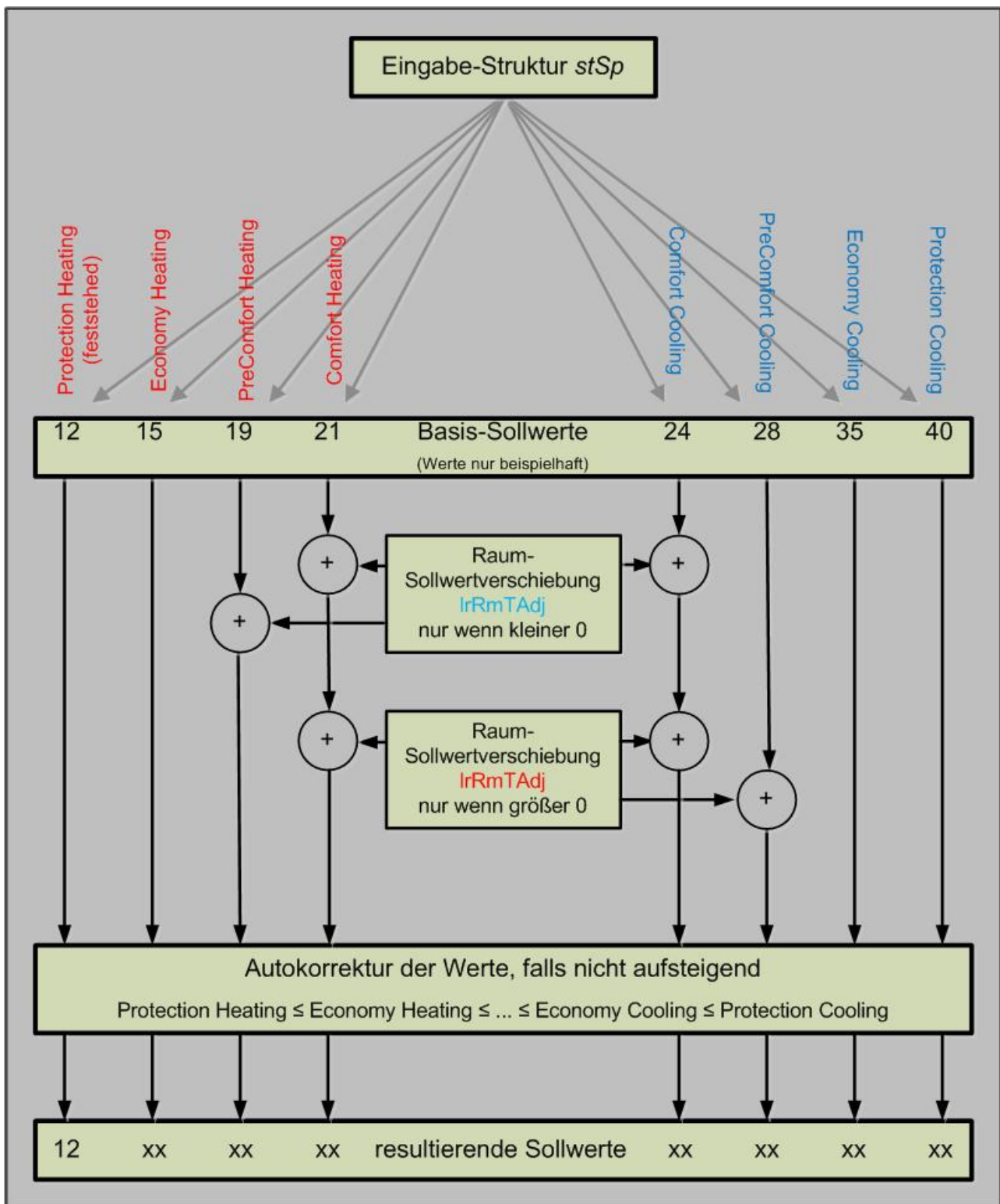
udiRemTiLmtRchd: Countdown Timer Grenzwert erreicht

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.96 FB_BA_RmTAdj**Funktionsbeschreibung**

Der Funktionbaustein *FB_BA_RmTAdj* dient der Anpassung des Raumtemperatursollwerts vom Nutzer im Raum. Er verschiebt die Sollwerte am Eingang des Funktionsbausteins in Abhängigkeit eines Offsets *rRmTAdj*, wie in der folgenden Skizze dargestellt. Am Eingang *rRmTAdj* kann für die Sollwertkorrektur z. B. der Wert eines Widerstandspotentiometers, oder eines busfähigen Feldgeräts verwendet werden.



Ist der Einstellwert *rRmTAdj* größer als Null, so wird eine Raumtemperatur-Erwärmung gewünscht: Der Wert Comfort Heating wird um den Wert *rRmTAdj* angehoben. Gleichzeitig werden auch die Werte Comfort Cooling und Precomfort Cooling erhöht. Ist der Wert *rRmTAdj* hingegen kleiner als Null, so wird eine Raumtemperatur-Absenkung erwünscht. Analog zum Erwärmungsfall werden nun die Werte Comfort Cooling, Comfort Heating und PreComfort Heating um den Wert *rRmTAdj* herabgesetzt.

Auto-Korrektur

Die Temperaturanpassung ist dazu gedacht, die Werte in kleinem Rahmen zu beeinflussen. Natürlich können beliebige Eingangswerte eingegeben werden. Eine Heizungsanlage wird jedoch nur dann sinnvoll funktionieren, wenn die Sollwerte in folgender Reihenfolge aufsteigende Werte besitzen:

- Protection Heating
- Economy Heating
- Precomfort Heating
- Comfort Heating
- Comfort Cooling
- Precomfort Cooling
- Economy Cooling
- Protection Cooling

Die Auto-Korrektur arbeitet nach dem Prinzip, daß beginnend mit dem Wert Economy Heating geprüft wird, ob dieser Wert kleiner als der niedrigere, in diesem Fall Protection Heating ist. Ist das der Fall, wird der Wert Economy Heating dem Wert Protection Heating gleich gesetzt. Es folgt dann die Prüfung, ob der Wert Precomfort Heating kleiner ist als Economy Heating und so weiter bis zur Prüfung, ob Protection Cooling kleiner ist als Economy Cooling. Sind ein oder mehrere Werte korrigiert worden, so wird dies am Ausgang *bValCorr* mit einem TRUE-Signal angezeigt.

Ein-Ausgänge

VAR_INPUT

```
rRmTAdj : REAL;
stSp    : ST_BA_SpRmT;
```

rRmTAdj: Raumtemperatur-Verschiebungswert

stSp : Eingabe-Struktur [► 336] der Sollwerte

VAR_OUTPUT

```
bValCorr      : BOOL;
rPrPrtcHtg    : REAL;
rPrEcoHtg     : REAL;
rPrPreCmfHtg  : REAL;
rPrCmfHtg     : REAL;
rPrPrtcCol    : REAL;
rPrEcoCol     : REAL;
rPrPreCmfCol  : REAL;
rPrCmfCol     : REAL;
stPrSp        : ST_BA_SpRmT;
```

bValCorr: Autokorrektur der Werte wurde durchgeführt, siehe oben

rPrPrtcHtg: Resultierender Sollwert Protection Heating

rPrEcoHtg: Resultierender Sollwert Economy Heating

rPrPreCmfHtg: Resultierender Sollwert PreComfort Heating

rPrCmfHtg: R Resultierender Sollwert Comfort Heating

rPrCmfCol: Resultierender Sollwert Comfort Cooling

rPrPreCmfCol: Resultierender Sollwert PreComfort Cooling

rPrEcoCol: Resultierender Sollwert Economy Cooling

rPrPrtcCol : Resultierender Sollwert Protection Cooling

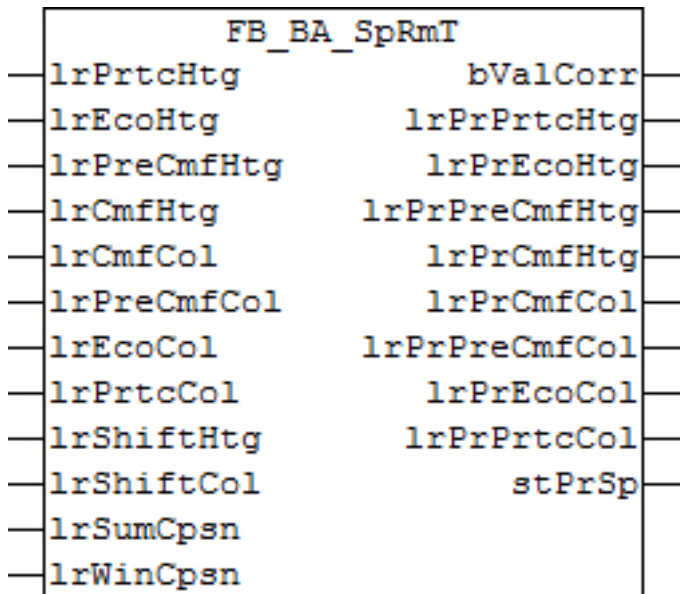
stPrSp: Zusammengefasste Ausgabe der resultierenden Werte in einer Struktur

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2241	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

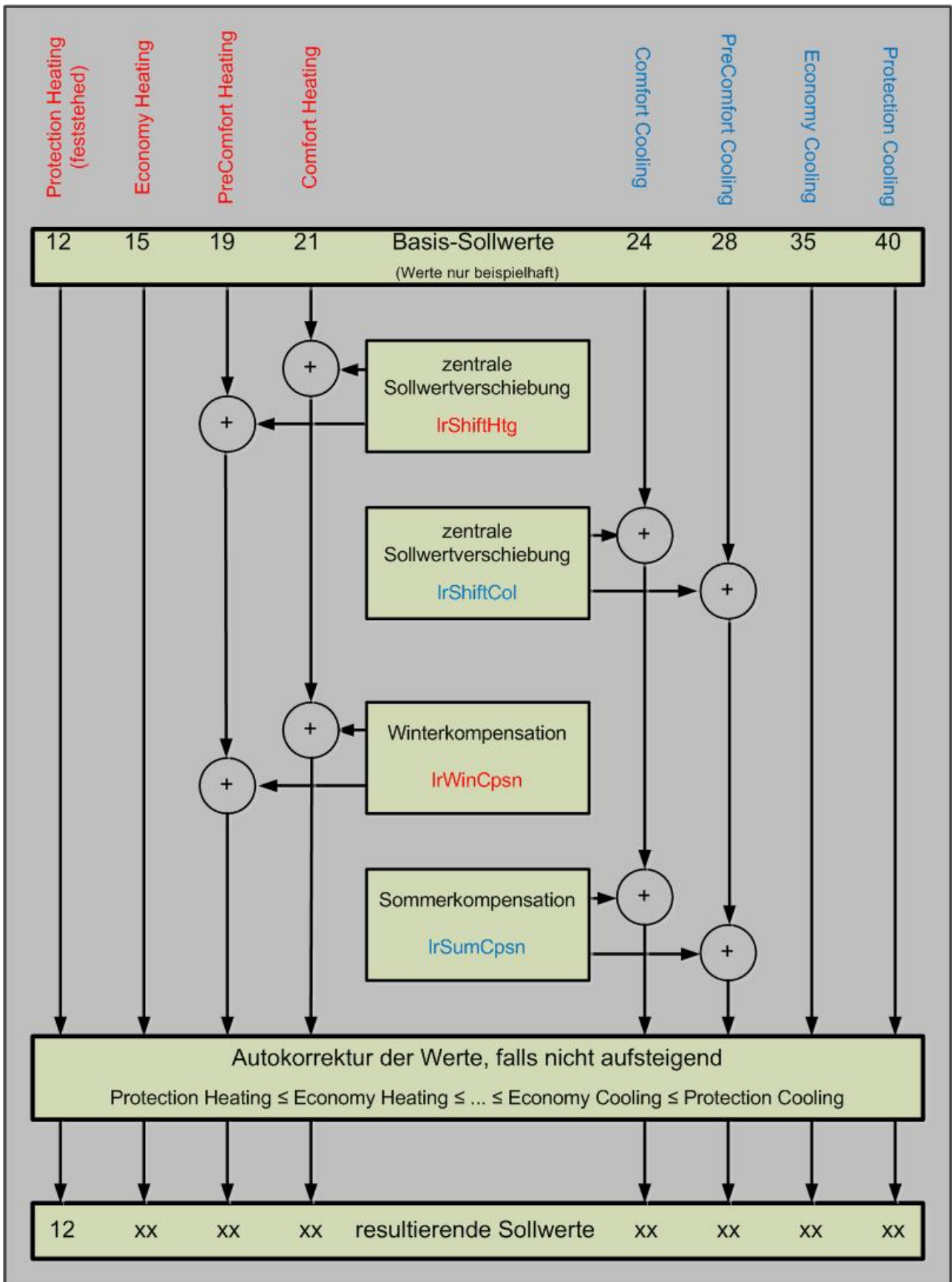
8.2.97 FB_BA_SpRmT

Bildung Raumtemperatur-Sollwerte



Funktionsbeschreibung

Der Funktionsbaustein *FB_BA_SpRmT* weist den Energieniveaus Protection, Economy, PreComfort und Comfort jeweils einen Sollwert für den Kühl- und Heizbetrieb zu. Die folgende Grafik beschreibt das Verhalten des Funktionsbausteins, wobei die eingetragenen Werte als Beispiel zu sehen sind:



Die Comfort- und PreComfort-Werte des Heizbetriebes werden mit dem Wert *IrShiftHeating* als zentrale Sollwertverschiebung beaufschlagt. Unabhängig davon wird noch Winterkompensation *IrWinCpsn* hinzuaddiert.

Für den Kühlbetrieb gilt analog: Die Comfort- und PreComfort-Werte werden mit dem Wert *IrShiftCooling* und unabhängig davon mit dem Wert der Sommerkompensation *IrSumCpsn* beaufschlagt.

Auto-Korrektur

Die Sollwert-Verschiebung ist dazu gedacht, die Werte in kleinem Rahmen zu beeinflussen. Natürlich können beliebige Eingangswerte eingegeben werden. Eine Heizungsanlage wird jedoch nur dann sinnvoll funktionieren, wenn die Sollwerte in folgender Reihenfolge aufsteigende Werte besitzen:

- Protection Heating
- Economy Heating
- Precomfort Heating
- Comfort Heating
- Comfort Cooling
- Precomfort Cooling
- Economy Cooling
- Protection Cooling

Die Auto-Korrektur arbeitet nach dem Prinzip, dass beginnend mit dem Wert Economy Heating geprüft wird, ob dieser Wert kleiner als der niedrigere, in diesem Fall Protection Heating ist. Ist das der Fall, wird der Wert Economy Heating dem Wert Protection Heating gleich gesetzt. Es folgt dann die Prüfung, ob der Wert Precomfort Heating kleiner ist als Economy Heating und so weiter bis zur Prüfung, ob Protection Cooling kleiner ist als Economy Cooling. Sind ein oder mehrere Werte korrigiert worden, so wird dies am Ausgang *bValCorr* mit einem TRUE-Signal angezeigt.

Ein-Ausgänge

VAR_INPUT

```
lrPrtcHtg : LREAL;  
lrEcoHtg : LREAL;  
lrPreCmfHtg : LREAL;  
lrCmfHtg : LREAL;  
lrCmfCol : LREAL;  
lrPreCmfCol : LREAL;  
lrEcoCol : LREAL;  
lrPrtcCol : LREAL;  
lrShiftHtg : LREAL;  
lrShiftCol : LREAL;  
lrSumCpsn : LREAL;  
lrWrWinCpsn : LREAL;
```

bValCorr: Autokorrektur: mindestens einer der resultierenden Sollwerte wurde so angepasst, dass die Werte weiterhin monoton steigend sind.

lrPrtcHtg : Basis-Sollwert Protection Heating

lrEcoHtg: Basis-Sollwert Economy Heating

lrPreCmfHtg: Basis-Sollwert PreComfort Heating

lrCmfHtg: Basis-Sollwert Comfort Heating

lrCmfCol: Basis-Sollwert Comfort Cooling

lrPreCmfCol: Basis-Sollwert PreComfort Cooling

lrEcoCol: Basis-Sollwert Economy Cooling

lrPrtcCol : Basis-Sollwert Protection Cooling

lrShiftHtg : Sollwertverschiebung Heizen

lrShiftCol: Sollwertverschiebung Kühlen

lrSumCpsn: Wert Sommerkompensation

lrWinCpsn: Wert Winterkompensation

VAR_OUTPUT

```

lrPrPrtcHtg : LREAL;
lrPrEcoHtg : LREAL;
lrPrPreCmfHtg : LREAL;
lrPrCmfHtg : LREAL;
lrPrPrtcCol : LREAL;
lrPrEcoCol : LREAL;
lrPrPreCmfCol : LREAL;
lrPrCmfCol : LREAL;
stPrSp : ST_BA_SpRmT;

```

lrPrPrtcHtg : Resultierender Sollwert Protection Heating

lrPrEcoHtg: Resultierender Sollwert Economy Heating

lrPrPreCmfHtg: Resultierender Sollwert PreComfort Heating

lrPrCmfHtg: R Resultierender Sollwert Comfort Heating

lrPrCmfCol: Resultierender Sollwert Comfort Cooling

lrPrPreCmfCol: Resultierender Sollwert PreComfort Cooling

lrPrEcoCol: Resultierender Sollwert Economy Cooling

lrPrPrtcCol : Resultierender Sollwert Protection Cooling

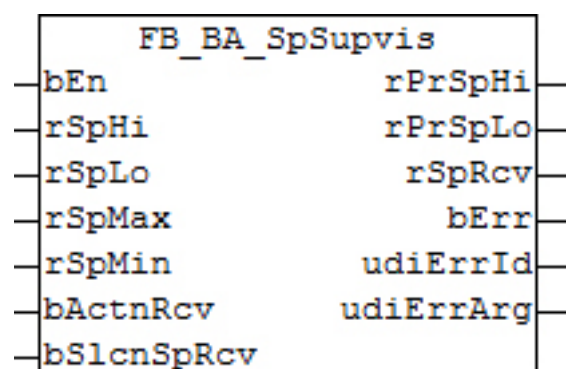
stPrSp: Zusammengefasste Ausgabe der resultierenden Werte in einer [Struktur](#) [► 336].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.98 FB_BA_SpSupvis

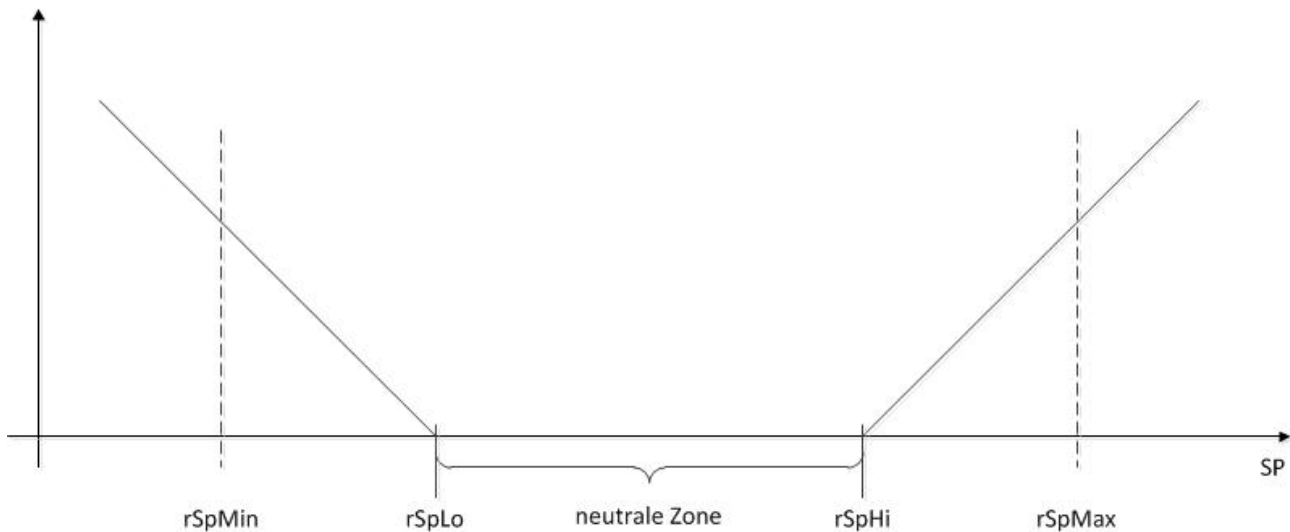
Funktionsbaustein zur Verarbeitung und Kontrolle des unteren und oberen Sollwerts einer Zuluftfeuchte- oder Zulufttemperaturregelung.

**Funktionsbeschreibung****Überprüfung und Limitierung der Sollwerte**

Der Funktionsbaustein limitiert und begrenzt die Sollwerte. Die beiden folgenden Tabellen stellen dar was überprüft wird und wie im Fehler reagiert wird.

Überprüfung		Aktion
rSpLo > rSpHi		letzten gültigen Werte von rSpLo und rSpHi werden genommen
rSpMin >= rSpMax		letzten gültigen Werte von rSpMin und rSpMax werden genommen
rSpHi > rSpMax		rPrSpHi = rSpMax
rSpLo < rSpMin		rPrSpLo = rSpMin
Überprüfung	bErr	Aktion
rSpMin >= rSpMax	TRUE	rSpErr = ((rSpMin + rSpMax) / 2)
rSpHi < rSpMin		rPrSpHi = rPrSpLo = rPrRcv = rSpErr
rSpLo > rSpMax		

Der Abstand zwischen den Sollwerten beschreibt eine Energieneutrale Zone. Bei der Verwendung in einer Zuluftregelung würde innerhalb der Neutralen Zone nicht geheizt und nicht gekühlt werden.



Die überprüften und u.U. limitierten Sollwerte werden am Bausteinenausgang als *rPrSpHi* und *rPrSpLo* (Present Setpoint) ausgegeben.

Sollwert für Wärmerückgewinnung

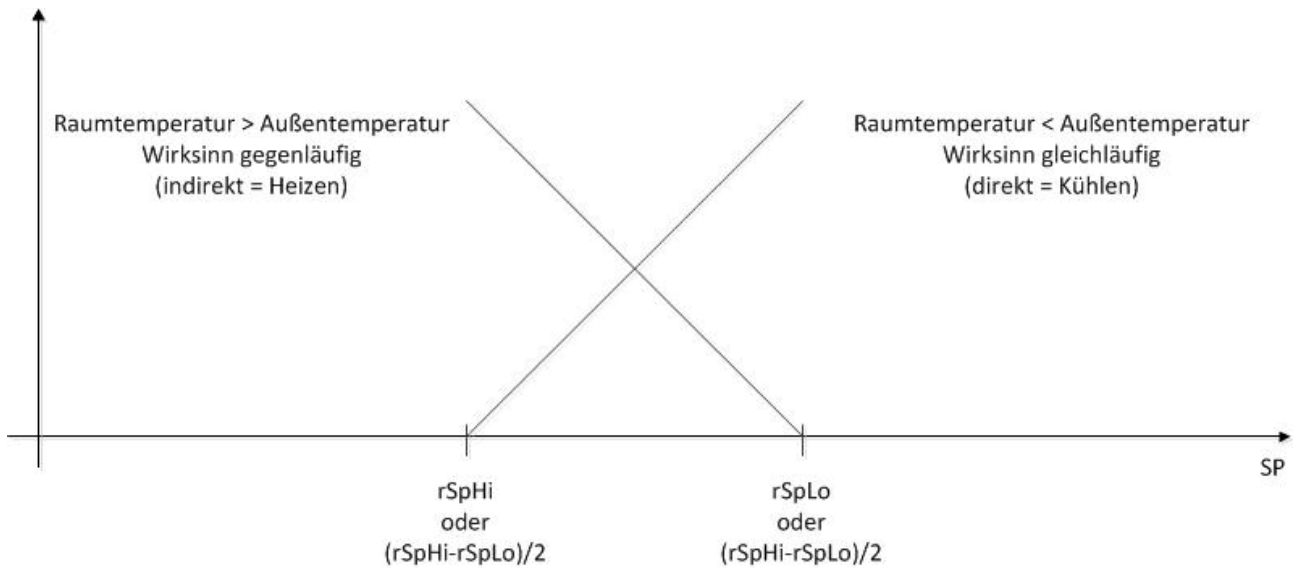
Für eine Wärmerückgewinnung wird der Sollwert *rSpRcv* wahlweise aus dem Mittelwert des oberen und unteren Sollwerts, *rSoHi* und *rSpLo* berechnet oder aber in Abhängigkeit vom Wirksinn der WRG. Die Methode wird durch die Eingangsvariable *bSlcnSpRcv* definiert:

b SlcnSpRcv	rSpRcv
TRUE	Mittelwert aus rSpLo und rSpHi
FALSE	Wirksinnabhängig, definiert durch Eingang bActRcv

Im Falle der Wirksinnabhängigen Definition des Sollwerts gilt dann:

bActRcv	Wirksinn	rSpRcv
TRUE	direkt (Kühlen)	lrSpHi
FALSE	indirekt (Heizen)	lrSpLo

Wärmerückgewinnung (WRG)



Ein-Ausgänge

VAR_INPUT

```

bEn      : BOOL;
rSpHi    : REAL;
rSpLo    : REAL;
rSpMax   : REAL;
rSpMin   : REAL;
bActnRcv : BOOL;
bSlcnSpRcv : BOOL;

```

bEn : Bausteinfreigabe. Wenn $bEn = FALSE$, stehen alle Ausgabeparameter auf 0.0

rSpHi: Zu überprüfender Eingabewert des oberen Sollwerts

rSpLo: Zu überprüfender Eingabewert des unteren Sollwerts

rSpMax: Maximaler Sollwert

rSpMin: Minimaler Sollwert

bActnRcv: Wirkrichtung der nachgeschalteten Wärmerückgewinnung

bSlcnSpRcv: Sollwert-Wahl der nach geschalteten Wärmerückgewinnung

VAR_OUTPUT

```

rPrSpHi  : REAL;
rPrSpLo  : REAL;
rSpRcv   : REAL;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;

```

rPrSpHi: Ausgabewert des oberen Sollwerts

rPrSpLo: Ausgabewert des unteren Sollwerts

rSpRcv: Ausgabewert des resultierenden Sollwerts für die Wärmerückgewinnung

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

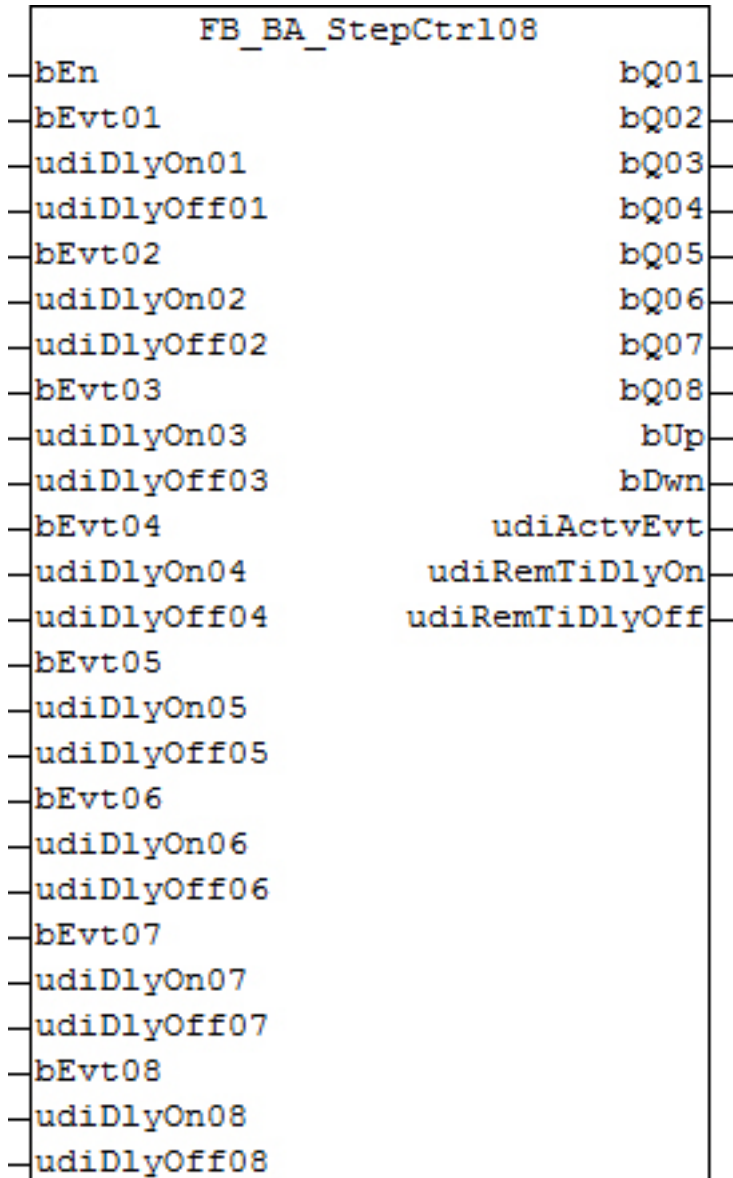
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.99 FB_BA_StepCtrl08

Schaltstufenbaustein, 8-fach



Funktionsbeschreibung

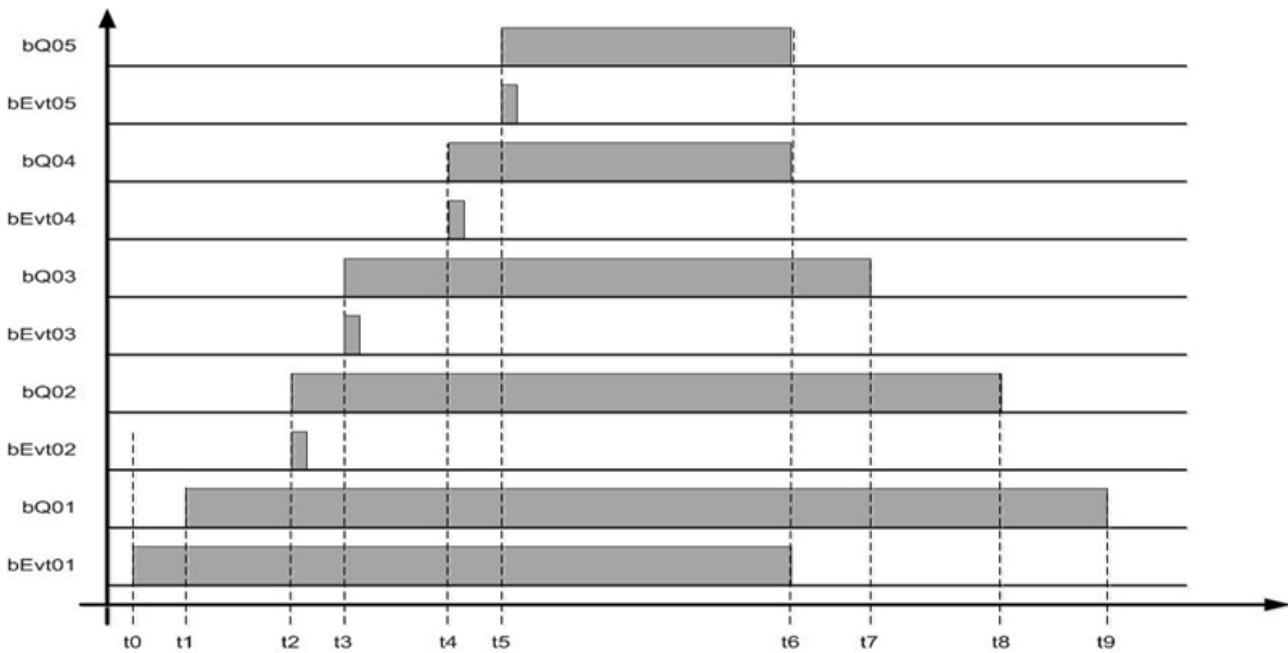
Der Funktionsbaustein dient zur Ausgabe sequenziell folgender Steuerungsbefehle. Eine typische Anwendung dieses Bausteins ist der Startvorgang einer raumlufttechnischen Anlage. *bEn* ist für die generelle Freigabe des Funktionsbausteins. Bei *bEn* = FALSE werden alle Ausgänge von *bQ01* bis *bQ08* auf FALSE gesetzt. Das Starten der Steuersequenz erfolgt am Eingang *bEvt01*. Nach dem Ablauf des Timers *udiDlyOn01* wird der zugehörige Ausgang *bQ01* gesetzt. Weitere Stufen schalten nach einer steigenden Flanke an den Eingängen *bEvt02* bis *bEvt08* jeweils Zeitverzögert über die Timer *udiDlyOn02* bis

udiDlyOn08 zu. Wird *bEvt01* bei hochgelaufener Steuerkette FALSE dann schalte die Steuersequenz in umgekehrter Reihenfolge zurück. Das Abschalten der Ausgänge wird durch die Timer *udiDlyOff01* bis *udiDlyOff08* verzögert.

Die Ausgänge *bUp* und *bDwn* zeigen an ob sich die Steuerkette im absteigenden oder fallenden Zustand befindet. Die Variable *udiActvEvt* zeigt an in welchen Schritt sich die Steuerkette aktuell befindet. Dabei bedeutet "0", dass die Schrittkette nicht aktiv ist. Für die Verwendung mit einem BACnet-Multistate-Output-Objekt, welches keine "0" anzeigen kann, steht der Ausgang *udiStep* zur Verfügung, der den Ausgang *udiActvEvt+1* darstellt.

udiRemTiDlyOn zeigt beim Hochschalten der Steuerkette die verbleibenden Zeit bis zum nächsten Schritt an. *udiRemTiDlyOff* zeigt beim Runterschalten der Steuerkette die verbleibende Zeit bis zum Umschalten in den nächst niedrigeren Schritt an.

Beispiel



t0 Einschalten der Schrittkette

t1 Einschalten Schritt 1 *udiDlyOn01* = t1 - t0

t2 Ereignis Freigabe Schritt 2, Einschalten Schritt 2, *udiDlyOn02* = 0

t3 Ereignis Freigabe Schritt 3, Einschalten Schritt 3, *udiDlyOn03* = 0

t4 Ereignis Freigabe Schritt 4, Einschalten Schritt 4, *udiDlyOn04* = 0

t5 Ereignis Freigabe Schritt 5, Einschalten Schritt 5, *udiDlyOn05* = 0

t6 Abschalten der Schrittkette, Abschalten Schritt 5, Abschalten Schritt 4; *udiDlyOff05* = 0, *udiDlyOff04* = 0

t7 Abschalten Schritt 3, *udiDlyOff03* = t7 -t6

t8 Abschalten Schritt 2, *udiDlyOff02* = t8 -t7

t9 Abschalten Schritt 1, *udiDlyOff01* = t9 -t8

Ein-/Ausgänge

VAR_INPUT

```

bEn          : BOOL;
bEvt01       : BOOL;
udiDlyOn01   : UDINT;
udiDlyOff01  : UDINT;
bEvt02       : BOOL;
udiDlyOn02   : UDINT;
udiDlyOff02  : UDINT;
bEvt03       : BOOL;
udiDlyOn03   : UDINT;
udiDlyOff03  : UDINT;
bEvt04       : BOOL;
udiDlyOn04   : UDINT;
udiDlyOff04  : UDINT;
bEvt05       : BOOL;

```

```
udiDlyOn05 : UDINT;  
udiDlyOff05 : UDINT;  
bEvt06 : BOOL;  
udiDlyOn06 : UDINT;  
udiDlyOff06 : UDINT;  
bEvt07 : BOOL;  
udiDlyOn07 : UDINT;  
udiDlyOff07 : UDINT;  
bEvt08 : BOOL;  
udiDlyOn08 : UDINT;  
udiDlyOff08 : UDINT;
```

bEn: Freigabe des Funktionsbausteins

bEvt01: Einschalten der Steuerkette

udiDlyOn01: Einschaltverzögerung für Ausgang *bQ01* [s]

udiDlyOff01: Ausschaltverzögerung für Ausgang *bQ01* [s]

bEvt02: Weiterschaltbefehl Schritt 2

udiDlyOn02: Einschaltverzögerung für Ausgang *bQ02* [s]

udiDlyOff02: Ausschaltverzögerung für Ausgang *bQ02* [s]

bEvt03: Weiterschaltbefehl Schritt 3

udiDlyOn03: Einschaltverzögerung für Ausgang *bQ03* [s]

udiDlyOff03: Ausschaltverzögerung für Ausgang *bQ03* [s]

bEvt04: Weiterschaltbefehl Schritt 4

udiDlyOn04: Einschaltverzögerung für Ausgang *bQ04* [s]

udiDlyOff04: Ausschaltverzögerung für Ausgang *bQ04* [s]

bEvt05: Weiterschaltbefehl Schritt 5

udiDlyOn05: Einschaltverzögerung für Ausgang *bQ05* [s]

udiDlyOff05: Ausschaltverzögerung für Ausgang *bQ05* [s]

bEvt06: Weiterschaltbefehl Schritt 6

udiDlyOn06: Einschaltverzögerung für Ausgang *bQ06* [s]

udiDlyOff06: Ausschaltverzögerung für Ausgang *bQ06* [s]

bEvt07: Weiterschaltbefehl Schritt 7

udiDlyOn07: Einschaltverzögerung für Ausgang *bQ07* [s]

udiDlyOff07: Ausschaltverzögerung für Ausgang *bQ07* [s]

bEvt08: Weiterschaltbefehl Schritt 8

udiDlyOn08: Einschaltverzögerung für Ausgang *bQ08* [s]

udiDlyOff08: Ausschaltverzögerung für Ausgang *bQ08* [s]

VAR_OUTPUT

```
bQ01 : BOOL;  
bQ02 : BOOL;  
bQ03 : BOOL;  
bQ04 : BOOL;  
bQ05 : BOOL;  
bQ06 : BOOL;  
bQ07 : BOOL;  
bQ08 : BOOL;  
bUp : BOOL;  
bDwn : BOOL;
```

```

udiActvEvt      : UDINT;
udiRemTiDlyOn  : UDINT;
udiRemTiDlyOff : UDINT;

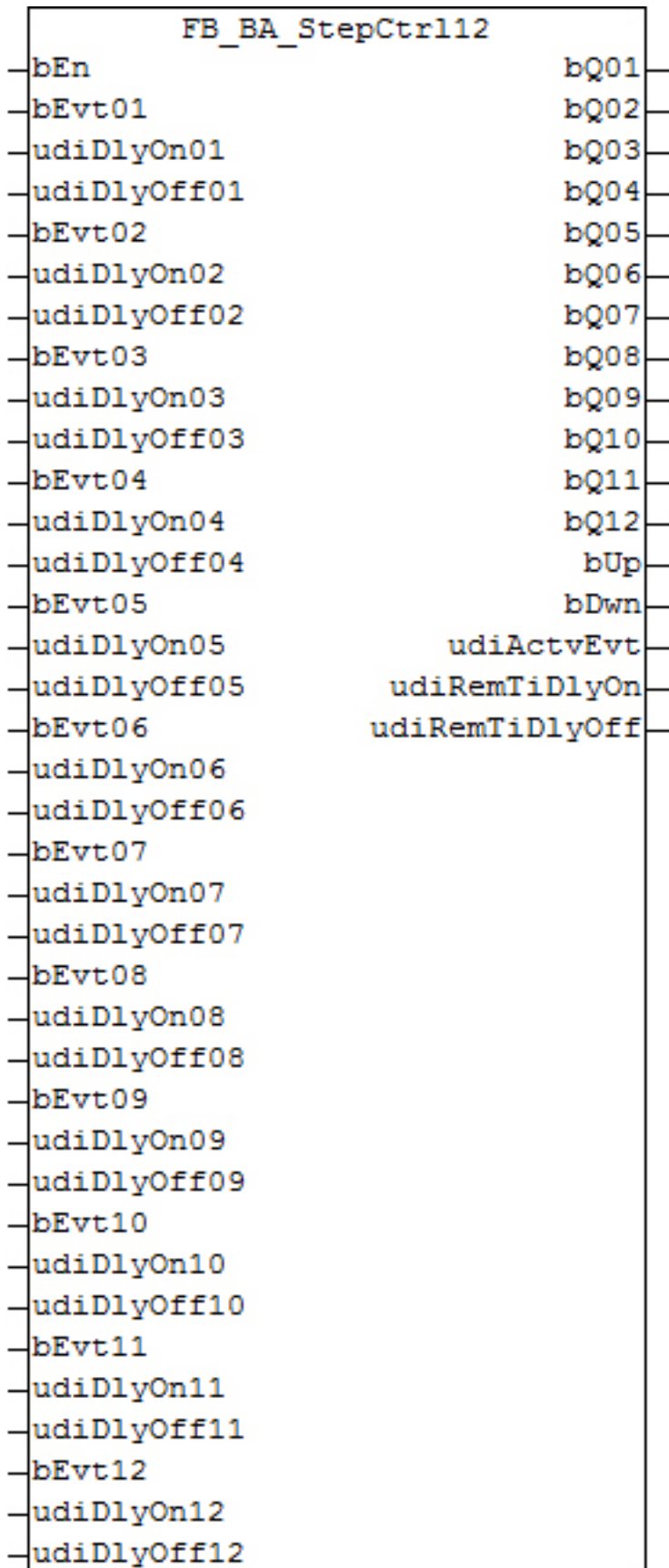
```

bQ01: Schritt 1 Ein**bQ02:** Schritt 2 Ein**bQ03:** Schritt 3 Ein**bQ04:** Schritt 4 Ein**bQ05:** Schritt 5 Ein**bQ06:** Schritt 6 Ein**bQ07:** Schritt 7 Ein**bQ08:** Schritt 8 Ein**bUp:** Steuerkette ist im aufsteigenden Zustand**bDwn:** Steuerkette ist im fallenden Zustand**udiActvEvt:** Aktiver Schritt, Anzeige 0..8, wobei "0" für eine nicht aktive Schrittkette steht.**udiRTiDlyOn:** Restzeit bis zum Hochschalten in den nächsten Schritt [s]**udiRTiDlyOff:** Restzeit bis zum Runterschalten in den vorherigen Schritt [s]**Voraussetzungen**

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.100 FB_BA_StepCtrl12

Schaltstufenbaustein, 12-fach



Funktionsbeschreibung

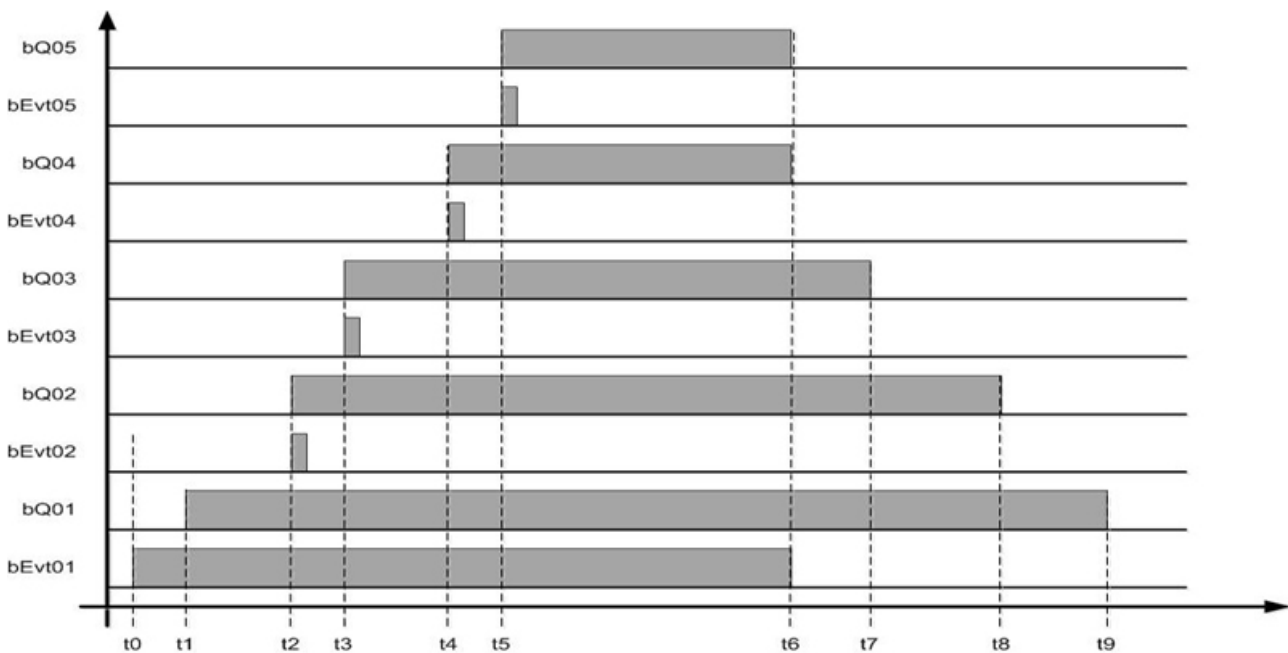
Der Funktionsbaustein dient zur Ausgabe sequenziell folgender Steuerungsbefehle. Eine typische Anwendung dieses Bausteins ist der Startvorgang einer raumlufotechnischen Anlage. *bEn* ist für die generelle Freigabe des Funktionsbausteins. Bei *bEn* = FALSE werden alle Ausgänge von *bQ01* bis *bQ12*

auf FALSE gesetzt. Das Starten der Steuersequenz erfolgt am Eingang *bEvt01*. Nach dem Ablauf des Timers *udiDlyOn01* wird der zugehörige Ausgang *bQ01* gesetzt. Weitere Stufen schalten nach einer steigenden Flanke an den Eingängen *bEvt02* bis *bEvt12* jeweils Zeitverzögert über die Timer *udiDlyOn02* bis *udiDlyOn12* zu. Wird *bEvt01* bei hochgelaufener Steuerkette FALSE dann schalte die Steuersequenz in umgekehrter Reihenfolge zurück. Das Abschalten der Ausgänge wird durch die Timer *udiDlyOff01* bis *udiDlyOff12* verzögert.

Die Ausgänge *bUp* und *bDwn* zeigen an ob sich die Steuerkette im absteigenden oder fallenden Zustand befindet. Die Variable *udiActvEvt* zeigt an in welchen Schritt sich die Steuerkette aktuell befindet. Dabei bedeutet "0", dass die Schrittkette nicht aktiv ist. Für die Verwendung mit einem BACnet-Multistate-Output-Objekt, welches keine "0" anzeigen kann, steht der Ausgang *udiStep* zur Verfügung, der den Ausgang *udiActvEvt+1* darstellt.

udiRemTiDlyOn zeigt beim Hochschalten der Steuerkette die verbleibende Zeit bis zum nächsten Schritt an. *udiRemTiDlyOff* zeigt beim Runterschalten der Steuerkette die verbleibende Zeit bis zum Umschalten in den nächst niedrigeren Schritt an.

Beispiel



t0 Einschalten der Schrittkette

t1 Einschalten Schritt 1, $udiDlyOn01 = t1 - t0$

t2 Ereignis Freigabe Schritt 2, Einschalten Schritt 2, $udiDlyOn02 = 0$

t3 Ereignis Freigabe Schritt 3, Einschalten Schritt 3, $udiDlyOn03 = 0$

t4 Ereignis Freigabe Schritt 4, Einschalten Schritt 4, $udiDlyOn04 = 0$

t5 Ereignis Freigabe Schritt 5, Einschalten Schritt 5, $udiDlyOn05 = 0$

t6 Abschalten der Schrittkette, Abschalten Schritt 5, Abschalten Schritt 4; $udiDlyOff05 = 0$, $udiDlyOff04 = 0$

t7 Abschalten Schritt 3, $udiDlyOff03 = t7 - t6$

t8 Abschalten Schritt 2, $udiDlyOff02 = t8 - t7$

t9 Abschalten Schritt 1, $udiDlyOff01 = t9 - t8$

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
bEvt01   : BOOL;
udiDlyOn01 : UDINT;
udiDlyOff01 : UDINT;
bEvt02   : BOOL;
udiDlyOn02 : UDINT;
udiDlyOff02 : UDINT;
bEvt03   : BOOL;
udiDlyOn03 : UDINT;
udiDlyOff03 : UDINT;
bEvt04   : BOOL;
```

```
udiDlyOn04 : UDINT;  
udiDlyOff04 : UDINT;  
bEvt05 : BOOL;  
udiDlyOn05 : UDINT;  
udiDlyOff05 : UDINT;  
bEvt06 : BOOL;  
udiDlyOn06 : UDINT;  
udiDlyOff06 : UDINT;  
bEvt07 : BOOL;  
udiDlyOn07 : UDINT;  
udiDlyOff07 : UDINT;  
bEvt08 : BOOL;  
udiDlyOn08 : UDINT;  
udiDlyOff08 : UDINT;  
bEvt09 : BOOL;  
udiDlyOn09 : UDINT;  
udiDlyOff09 : UDINT;  
bEvt10 : BOOL;  
udiDlyOn10 : UDINT;  
udiDlyOff10 : UDINT;  
bEvt11 : BOOL;  
udiDlyOn11 : UDINT;  
udiDlyOff11 : UDINT;  
bEvt12 : BOOL;  
udiDlyOn12 : UDINT;  
udiDlyOff12 : UDINT;
```

bEn: Freigabe des Funktionsbausteins

bEvt01: Einschalten der Steuerkette

udiDlyOn01: Einschaltverzögerung für Ausgang *bQ01* [s]

udiDlyOff01: Ausschaltverzögerung für Ausgang *bQ01* [s]

bEvt02: Weiterschaltbefehl Schritt 2

udiDlyOn02: Einschaltverzögerung für Ausgang *bQ02* [s]

udiDlyOff02: Ausschaltverzögerung für Ausgang *bQ02* [s]

bEvt03: Weiterschaltbefehl Schritt 3

udiDlyOn03: Einschaltverzögerung für Ausgang *bQ03* [s]

udiDlyOff03: Ausschaltverzögerung für Ausgang *bQ03* [s]

bEvt04: Weiterschaltbefehl Schritt 4

udiDlyOn04: Einschaltverzögerung für Ausgang *bQ04* [s]

udiDlyOff04: Ausschaltverzögerung für Ausgang *bQ04* [s]

bEvt05: Weiterschaltbefehl Schritt 5

udiDlyOn05: Einschaltverzögerung für Ausgang *bQ05* [s]

udiDlyOff05: Ausschaltverzögerung für Ausgang *bQ05* [s]

bEvt06: Weiterschaltbefehl Schritt 6

udiDlyOn06: Einschaltverzögerung für Ausgang *bQ06* [s]

udiDlyOff06: Ausschaltverzögerung für Ausgang *bQ06* [s]

bEvt07: Weiterschaltbefehl Schritt 7

udiDlyOn07: Einschaltverzögerung für Ausgang *bQ07* [s]

udiDlyOff07: Ausschaltverzögerung für Ausgang *bQ07* [s]

bEvt08: Weiterschaltbefehl Schritt 8

udiDlyOn08: Einschaltverzögerung für Ausgang *bQ08* [s]

udiDlyOff08: Ausschaltverzögerung für Ausgang *bQ08* [s]

bEvt09: Weiterschaltbefehl Schritt 9

udiDlyOn09: Einschaltverzögerung für Ausgang *bQ09* [s]

udiDlyOff09: Ausschaltverzögerung für Ausgang *bQ09* [s]

bEvt10: Weiterschaltbefehl Schritt 10

udiDlyOn10: Einschaltverzögerung für Ausgang *bQ10* [s]

udiDlyOff10: Ausschaltverzögerung für Ausgang *bQ10* [s]

bEvt11: Weiterschaltbefehl Schritt 11

udiDlyOn11: Einschaltverzögerung für Ausgang *bQ11* [s]

udiDlyOff11: Ausschaltverzögerung für Ausgang *bQ11* [s]

bEvt12: Weiterschaltbefehl Schritt 12

udiDlyOn12: Einschaltverzögerung für Ausgang *bQ12* [s]

udiDlyOff12: Ausschaltverzögerung für Ausgang *bQ12* [s]

VAR_OUTPUT

```
bQ01      : BOOL;  
bQ02      : BOOL;  
bQ03      : BOOL;  
bQ04      : BOOL;  
bQ05      : BOOL;  
bQ06      : BOOL;  
bQ07      : BOOL;  
bQ08      : BOOL;  
bQ09      : BOOL;  
bQ10      : BOOL;  
bQ11      : BOOL;  
bQ12      : BOOL;  
bUp       : BOOL;  
bDwn      : BOOL;  
udiActvEvt : UDINT;  
udiRemTiDlyOn : UDINT;  
udiRemTiDlyOff : UDINT;
```

bQ01: Schritt 1 Ein

bQ02: Schritt 2 Ein

bQ03: Schritt 3 Ein

bQ04: Schritt 4 Ein

bQ05: Schritt 5 Ein

bQ06: Schritt 6 Ein

bQ07: Schritt 7 Ein

bQ08: Schritt 8 Ein

bQ09: Schritt 9 Ein

bQ10: Schritt 10 Ein

bQ11: Schritt 11 Ein

bQ12: Schritt 12 Ein

bUp: Steuerkette ist im aufsteigenden Zustand

bDwn: Steuerkette ist im fallenden Zustand

udiActvEvt: Aktiver Schritt, Anzeige 0..12, wobei "0" für eine nicht aktive Schrittkette steht.

udiRTidlyOn: Restzeit bis zum Hochschalten in den nächsten Schritt [s]

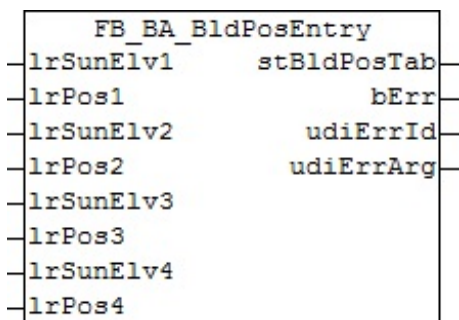
udiRTidlyOff: Restzeit bis zum Runterschalten in den vorherigen Schritt [s]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.101 FB_BA_BldPosEntry

Dieser Baustein dient zur Eingabe von Stützstellen für den Baustein [FB_BA_SunPrtc \[▶ 306\]](#), sollte dieser im Modus der Höhenpositionierung mit Hilfe einer Tabelle betrieben werden, siehe [E_BA_PosMod \[▶ 330\]](#).



Funktionsbeschreibung

Der Baustein [FB_BA_SunPrtc \[▶ 306\]](#) bietet neben den Betriebsarten "fixe Jalousiehöhe" und "maximaler Lichteinfall" auch die Möglichkeit, die Jalousiehöhe per Tabelleneinträge in Abhängigkeit von der Sonnenhöhe zu steuern. Durch die Eingabe mehrerer Stützpunkte wird linear interpoliert die betreffende Jalousiehöhe zum entsprechenden Sonnenstand errechnet. Da falsch eingetragene Werte jedoch zu Fehlfunktionen beim [FB_BA_SunPrtc \[▶ 306\]](#) führen können ist diesem der Baustein [FB_BA_BldPosEntry](#) voranzustellen. Es lassen sich vier Stützpunkte an diesem Baustein parametrieren, wobei ein fehlender Eintrag als Nulleintrag gewertet wird.

Der Baustein sortiert die eingegebenen Werte nicht selbstständig, sondern achtet darauf, dass die eingetragenen Sonnenstände der jeweiligen Stützstellen in aufsteigender Reihenfolge eingegeben wurden. Unbeabsichtigt fehlerhafte Einträge fallen dadurch schneller auf.

Die gewählten Werte für *lrSunElv1 .. lrSunElv4* müssen auch eindeutig sein, es darf beispielsweise nicht gelten:

[*lrSunElv1* = 10 ; *rBldPos1* = 50] und gleichzeitig [*lrSunElv2* = 10 ; *rBldPos2* = 30].

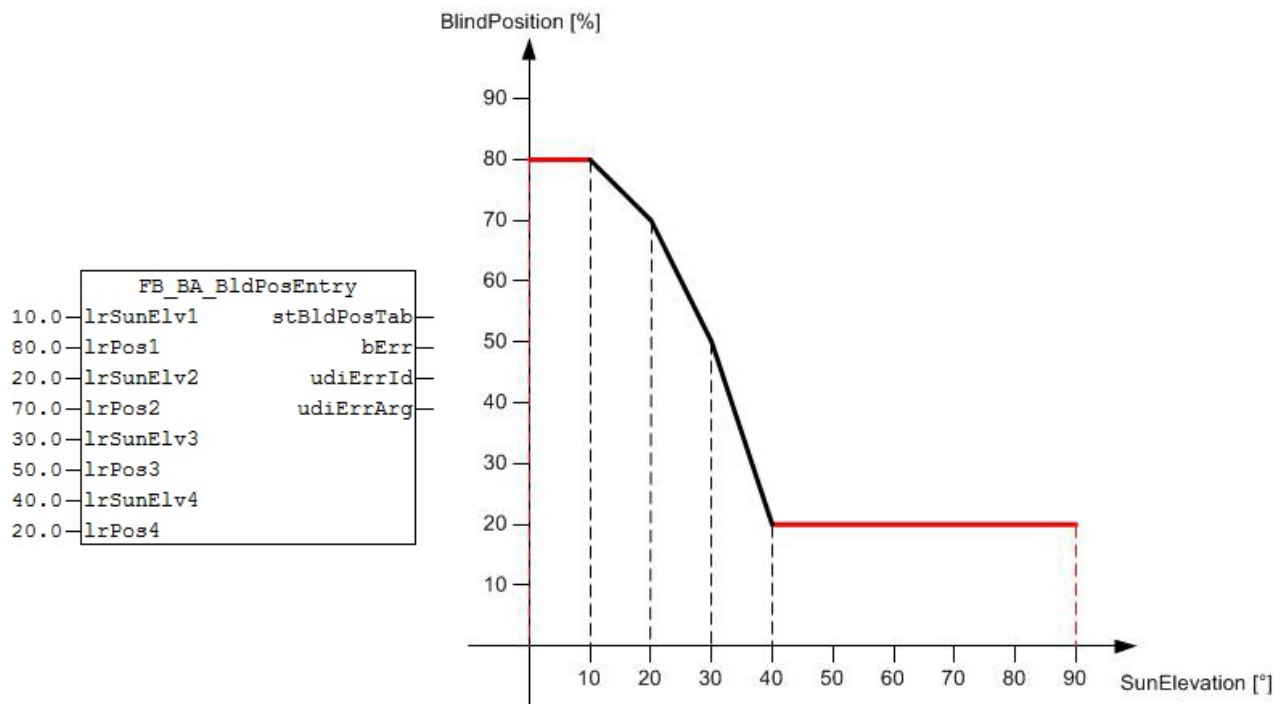
Das würde bedeuten, dass für ein und denselben Wert zwei verschiedene Zielwerte bestünden, was keinen eindeutigen funktionalen Zusammenhang bilden lässt.

Darüber hinaus müssen die Einträge für Sonnenstand und Jalousiehöhe im gültigen Bereich liegen. Das bedeutet mathematisch, dass folgende Bedingungen erfüllt sein müssen:

- $lrSunElv1 < lrSunElv2 < lrSunElv3 < lrSunElv4$ - (Werte aufsteigend und ungleich)
- $0 \leq rSunElv \leq 90$ ([°] - Gültigkeitsbereich Quellwerte)
- $0 \leq rBldPos \leq 100$ (in Prozent - Gültigkeitsbereich Zielwerte)

Der Baustein überprüft die eingetragenen Werte auf diese Bedingungen hin und gibt bei Nicht-Erfüllen einen Fehlercode [\[▶ 340\]](#) aus. Zusätzlich wird der Ausgang *bValid* auf FALSE gesetzt.

Des Weiteren sorgt der Baustein selbstständig für ein Ausfüllen der Randbereiche: Intern wird ein weiterer Stützpunkt bei *rSunElv* = 0 mit *rBldPos1* und ein weiterer oberhalb von *lrSunElv4* bei *rSunElv* = 90 mit *rBldPos4* aufgestellt. Damit wird sichergestellt, dass für alle gültigen Eingabewerte $0 \leq rSunElv \leq 90$ ein sinnvoller Zielwert vorhanden ist **ohne** dass ein Eintrag für *rSunElv* = 0 und *rSunElv* = 90 vom Anwender zwingend vergeben werden muss:



```

FB_BA_BldPosEntry
10.0 lrSunElv1      stBldPosTab
80.0 lrPos1         bErr
20.0 lrSunElv2     udiErrId
70.0 lrPos2         udiErrArg
30.0 lrSunElv3
50.0 lrPos3
40.0 lrSunElv4
20.0 lrPos4

```

Die tatsächliche Anzahl an Stützstellen, welche an den Baustein [FB_BA_SunPrtc](#) [► 306] übergeben wird, erhöht sich damit auf 6, siehe [ST_BA_BldPosTab](#) [► 330].

Die Interpolation der Werte erfolgt im Blendschutz-Baustein.

Ein-/Ausgänge

VAR_INPUT

```

lrSunElv1 : LREAL;
lrPos1    : LREAL;
lrSunElv2 : LREAL;
lrPos2    : LREAL;
lrSunElv3 : LREAL;
lrPos3    : LREAL;
lrSunElv4 : LREAL;
lrPos4    : LREAL;

```

lrSunElv1: Sonnenstand des 1. Stützpunktes (0°..90°)

rBldPos1: Jalousieposition (Grad der Schließung) des 1. Stützpunktes (0%..100%)

lrSunElv2: Sonnenstand des 2. Stützpunktes (0°..90°)

rBldPos2: Jalousieposition (Grad der Schließung) des 2. Stützpunktes (0%..100%)

lrSunElv3: Sonnenstand des 3. Stützpunktes (0°..90°)

rBldPos3: Jalousieposition (Grad der Schließung) des 3. Stützpunktes (0%..100%)

lrSunElv4: Sonnenstand des 4. Stützpunktes (0°..90°)

rBldPos4: Jalousieposition (Grad der Schließung) des 4. Stützpunktes (0%..100%)

VAR_OUTPUT

```

bValid      : BOOL;
uiErrorId   : UDINT;
stBldPosTab : ST_BA_BldPosTab;

```

stBldPosTab: Übergabestruktur der Stützstellen, siehe [ST_BA_BldPosTab](#) [► 330]

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

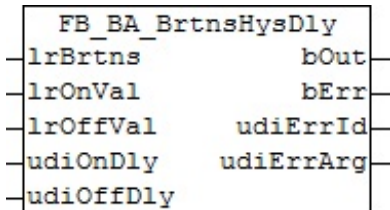
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.102 FB_BA_BrtnsHysDly

Helligkeits-Schwellwertschalter



Funktionsbeschreibung

Dieser Baustein stellt einen Schwellwertschalter für Helligkeit dar. Das Ein- bzw. Ausschaltverhalten kann zusätzlich zeitlich verzögert werden.

Ein-/Ausgänge

VAR_INPUT

```
lrBrtns      : LREAL;
lrOnVal      : LREAL;
lrOffVal     : LREAL;
udiOnDly     : UDINT;
udiOffDly    : UDINT;
```

lrBrtns: Außenhelligkeit [lx]

lrOnVal: Einschaltsschwellwert [lx]. Dieser muss größer sein als der Ausschaltsschwellwert *usiOffValue*.

lrOffVal: Ausschaltsschwellwert [lx]. Dieser muss kleiner sein als der Einschaltsschwellwert *usiOnValue*.

udiOnDly: Einschaltverzögerung [s]

udiOffDly: Ausschaltverzögerung [s]

VAR_OUTPUT

```
bOut         : BOOL;
bErr         : BOOL;
udiErrId     : UDINT;
udiErrArg    : UDINT;
```

bOut: Binärer verzögerter Ausgang des Schwellwertschalters.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

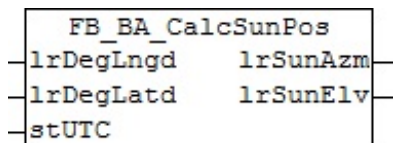
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.103 FB_BA_CalcSunPos

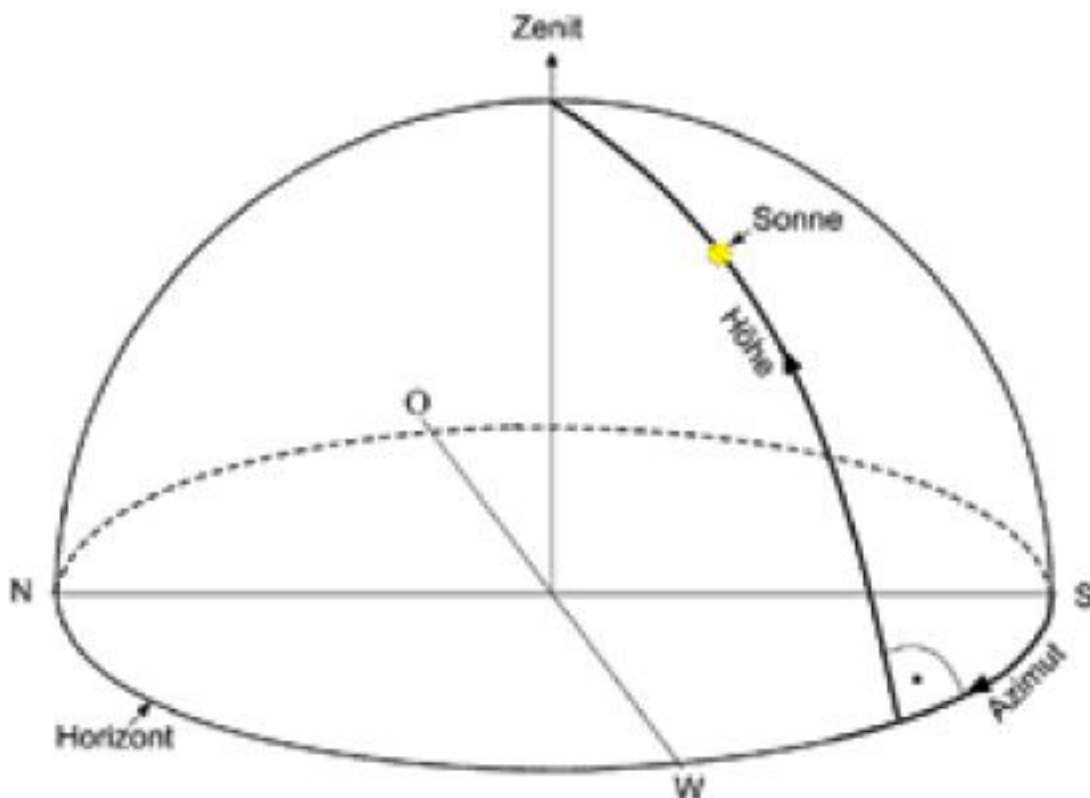
Berechnung des Sonnenstandes durch die Angabe von Datum, Uhrzeit, geographischer Länge und geographischer Breite.



Funktionsbeschreibung

Der Sonnenstand für einen gegebenen Zeitpunkt lässt sich nach gängigen Methoden mit definierter Genauigkeit berechnen. Für Anwendungen mit mäßigen Anforderungen genügt der hier vorliegende Baustein. Als Grundlage wurde hierfür der SUNAE-Algorithmus verwendet, der einen günstigen Kompromiss zwischen Genauigkeit und Rechenaufwand darstellt.

Der Stand der Sonne an einem festen Beobachtungsort wird normalerweise mit zwei Winkelangaben bestimmt. Die eine gibt die Höhe über dem Horizont an, wobei 0° bedeutet, dass sich die Sonne in der Horizontalebene des Standortes befindet und ein Wert von 90° , dass sie sich senkrecht über dem Beobachter befindet. Die andere Winkelangabe gibt die Richtung an, in der die Sonne steht. Bei dem SUNAE-Algorithmus wird unterschieden, ob der Beobachter auf der nördlichen (Längengrad > 0) oder auf der südlichen (Längengrad < 0) Erdhalbkugel steht. Ist der Beobachtungspunkt auf der nördlichen Erdhalbkugel, so wird ein Wert von 0° für die nördliche Sonnenrichtung zugeordnet und läuft dann im Uhrzeigersinn um den Kompass, d.h. 90° ist Osten, 180° ist Süden, 270° Westen etc. Ist der Beobachtungspunkt auf der südlichen Erdhalbkugel, so entspricht 0° der südlichen Richtung und verläuft dann entgegen dem Uhrzeigersinn, d.h. 90° ist Osten, 180° ist Norden, 270° Westen etc.



Bei der Angabe der Uhrzeit muss die Zeit als koordinierte Weltzeit (UTC, Universal Time Coordinated - im älteren Sprachgebrauch auch GMT, Greenwich-Mean-Time) angegeben werden.

Die geographische Breite ist die im Winkelmaß (also $[\circ]$) angegebene nördliche oder südliche Entfernung eines Ortes der Erdoberfläche vom Äquator. Die Breite kann Werte von 0° (am Äquator) bis $\pm 90^\circ$ (an den Polen) annehmen. Dabei gibt ein positives Vorzeichen die nördliche Richtung und ein negatives Vorzeichen

die südliche Richtung an. Die geographische Länge ist ein Winkel, der ausgehend vom Nullmeridian 0° (künstlich festgelegte Nord-Süd-Linie) Werte bis ±180° annehmen kann. Ein positives Vorzeichen gibt die Länge in östlicher Richtung und ein negatives Vorzeichen in westlicher Richtung an. Beispiele:

Ort	Geographische Länge	Geographische Breite
Sydney, Australien	151,2°	-33,9°
New York, USA	-74,0°	40,7°
London, England	-0,1°	51,5°
Moskau, Russland	37,6°	55,7°
Peking, China	116,3°	39,9°
Dubai, Vereinigte Arabische Emirate	55,3°	25,4°
Rio de Janeiro, Brasilien	-43,2°	-22,9°
Hawai, USA	-155,8°	20,2°
Verl, Deutschland	8,5°	51,9°

Gibt der Baustein *FB_BA_CalcSunPos* für die Sonnenhöhe *lrSunElv* einen negativen Wert zurück, so ist die Sonne nicht sichtbar. Dieses kann zur Bestimmung von Sonnenaufgang und Sonnenuntergang genutzt werden.

Ein-/Ausgänge

VAR_INPUT

```
lrDegLngd : LREAL;
lrDegLatd : LREAL;
stUTC     : TIMESTRUCT;
```

lrDegLngd: Geographische Länge [°]

lrDegLatd: Geographische Breite [°]

stUTC: Eingabe der aktuellen Uhrzeit als koordinierte Weltzeit. Mit Hilfe des Bausteins *FB_BA_GetTime* [▶ 325] kann diese Zeit aus einem Zielsystem gelesen werden.

VAR_OUTPUT

```
lrSunAzm : LREAL;
lrSunElv : LREAL;
```

lrSunAzm: Sonnenrichtung (nördliche Erdhalbkugel: 0° Norden ... 90° Osten ... 180° Süden ... 270° Westen ... / südliche Erdhalbkugel: 0° Süden ... 90° Osten ... 180° Norden ... 270° Westen ...)

lrSunElv: Sonnenhöhe (0° horizontal ... 90° senkrecht)

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.104 FB_BA_FcdElemEntry

Dieser Baustein dient zur Verwaltung aller Fassadenelemente (Fenster) einer Fassade, welche global in einer Liste von Fassadenelementen [▶ 46] hinterlegt ist. Er soll die Eingabe der Elementinformationen - auch im Hinblick auf die Nutzung der Target-Visualisierung - erleichtern. Eine schematische Darstellung der Objekte mit Beschreibung der Koordinaten ist unter Verschattungskorrektur: Grundlagen und Definitionen [▶ 21] gegeben.

FB_BA_FcdElemEntry	
iColumn	lrCnr2X
iRow	lrCnr2Y
bWrt	lrCnr3X
bRd	lrCnr3Y
usiGrp	lrCnr4X
lrCnr1X	lrCnr4Y
lrCnr1Y	bErr
lrWdwWdth	udiErrId
lrWdwHght	udiErrArg
arrFcdElem ▶	▶ arrFcdElem

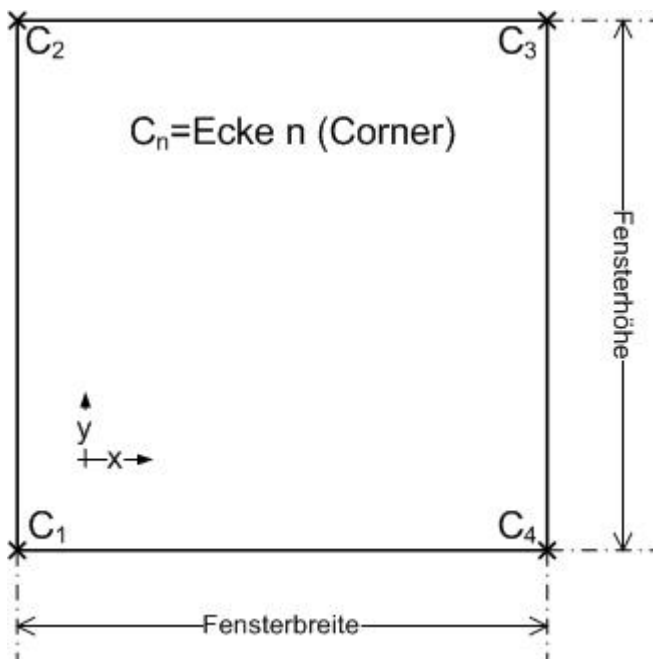
Funktionsbeschreibung

Die Deklaration der Fassadenelemente erfolgt in den globalen Variablen als zweidimensionales Feld über die Fensterspalten (Columns) und -reihen (Rows):

```
VAR_GLOBAL
    arrFcdElem : ARRAY[1..gBA_cMaxColumnFcd, 1..gBA_cMaxRowFcd] OF ST_BA_FcdElem;
END_VAR
```

Jedes einzelne Element $arrFcdElem[x,y]$ trägt die Informationen für jeweils ein Fassadenelement (ST_BA_FcdElem [▶ 332]). Dazu gehören die Gruppenzugehörigkeit, die Abmessungen (Breite, Höhe) und die Koordinaten der Eckpunkte. Der Baustein greift dabei über die IN-OUT-Variable $arrFcdElem$ direkt auf dieses Feld zu.

Anmerkung: Die Tatsache, dass die Koordinaten der Eckpunkte C2 bis C4 Ausgangswerte sind, ergibt sich daraus, dass sie aus den Eingabeparametern gebildet werden und für die Verwendung in einer Visualisierung zur Verfügung stehen sollen:



Alle Angaben [m]!

$lrCnr2X = lrCnr1X$
 $lrCnr2Y = lrCnr1Y + lrWdwHght$ (Fensterhöhe)
 $lrCnr3X = lrCnr1X + lrWdwWdth$ (Fensterbreite)
 $lrCnr3Y = lrCnr2Y$
 $lrCnr4X = lrCnr1X + lrWdwWdth$ (Fensterbreite)
 $lrCnr4Y = lrCnr1Y$

Die Verwendung des Bausteins erfolgt in drei Schritten:

- Auslesen
- Ändern
- Schreiben

Auslesen

Mit den Einträgen an *iColumn* und *iRow* wird das entsprechende Element aus der Liste, *arrFcdElem[iColumn,iRow]*, ausgewählt. Eine steigende Flanke an *bRd* liest folgende Daten aus dem Listenelement aus:

- *usiGrp* Gruppenzugehörigkeit,
- *IrCnr1X* X-Koordinate des Eckpunktes1 [m]
- *IrCnr1Y* Y-Koordinate des Eckpunktes1 [m]
- *IrWdwWdth* Fensterbreite [m]
- *IrWdwHght* Fensterhöhe [m]

Diese werden dann den entsprechenden Eingangsvariablen des Bausteins zugewiesen, der daraus nach dem oben erläuterten Zusammenhang die Koordinaten der Eckpunkte C2-C4 als Ausgangsvariablen errechnet. Wichtig ist hierbei, dass im Schritt des Auslesens die Eingangswerte nicht überschrieben werden. So lassen sich alle Werte zunächst in einer Visualisierung anzeigen.

Ändern

In einem nächsten Programmschritt können dann die aufgeführten Eingabewerte verändert werden. Die eingegebenen Werte werden dabei ständig auf Plausibilität überprüft. Der Ausgang *bErr* zeigt an, ob die Werte gültig sind (*bErr=FALSE*). Sollte dies nicht der Fall sein, wird am Ausgang *udiErrId/udiErrArg* ein entsprechender Fehlercode [► 340] ausgegeben. Siehe auch unten "Fehler (*bErr=TRUE*)".

Schreiben

Mit einer positiven Flanke an *bWrt* werden die parametrisierten Daten in das Listenelement mit dem Index *nId* geschrieben, unabhängig davon, ob sie gültige Werte darstellen oder nicht. Daher ist innerhalb der Elementstruktur *ST_BA_FcdElem* [► 332] ebenfalls ein Plausibilitätsbit *bVld* vorhanden, das genau diese Information an den Baustein *FB_BA_ShdCorr* [► 281] weiterreicht und dort Fehlberechnungen vorbeugt.

Diese Vorgehensweise soll nur als Vorschlag angesehen werden. Es ist natürlich auch möglich, den Baustein ganz normal in einem Schritt zu parametrieren und die eingetragenen Werte mit einer steigenden Flanke an *bWrt* in das entsprechende Listenelement zu schreiben.

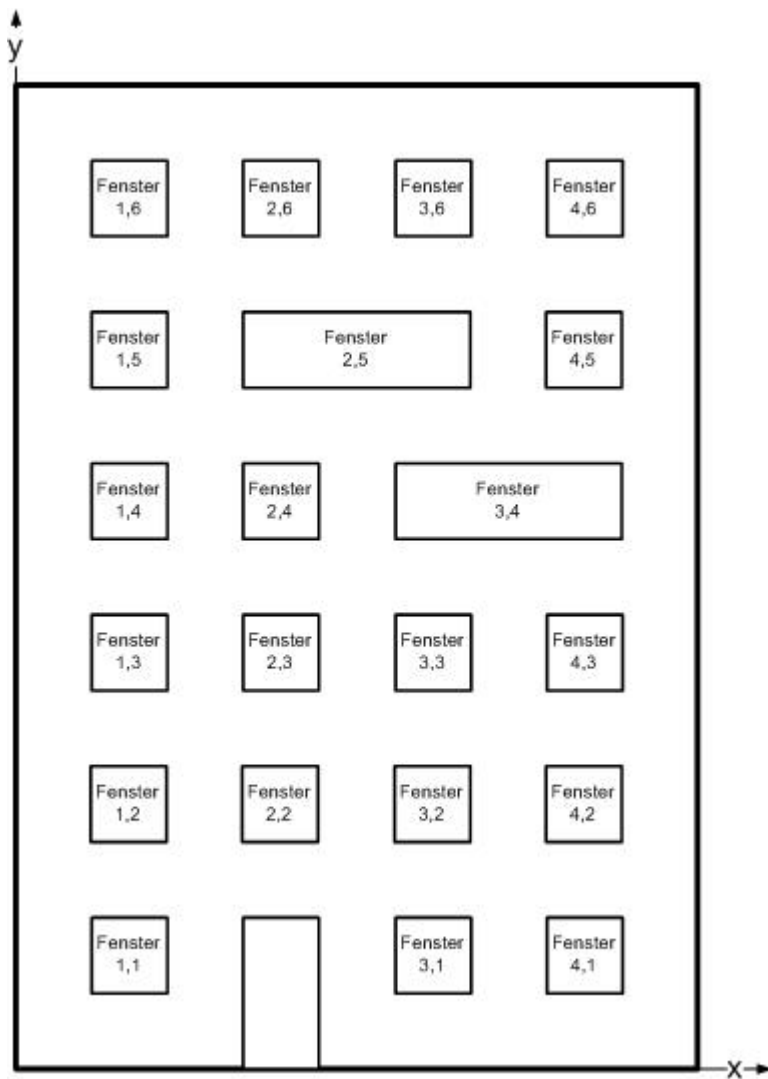
Fehler (*bErr=TRUE*)

Der Baustein *FB_BA_ShdCorr* [► 281], welcher beurteilt, ob alle Fenster einer Gruppe verschattet sind, wird nur dann seine Aufgabe ausführen, wenn alle Fenster der betrachteten Gruppe gültige Einträge haben. Das bedeutet:

- *usiGrp* muss größer als 0 sein
- *IrCnr1X* muss größer oder gleich 0.0 sein
- *IrCnr1Y* muss größer oder gleich 0.0 sein
- *IrWdwWdth* muss größer als 0 sein
- *IrWdwHght* muss größer als 0 sein

Ist eines dieser Kriterien nicht erfüllt, so wird dies als Falscheingabe interpretiert und der Fehlerausgang *bErr* am Bausteinausgang von *FB_BA_FcdElemEntry* gesetzt. Innerhalb des Fensterelementes *ST_BA_FcdElem* [► 332] wird das Plausibilitätsbit *bVld* auf FALSE gesetzt.

Sind hingegen **alle** Einträge eines Fassadenelementes Null, wird es als gültiges, gewollt ausgelassenes Fassadenelement angesehen:



Bei einer Fassade von 6x4 Fenstern wären hier die Elemente Fenster (2,1), Fenster (3,5) und Fenster (4,4) Leerelemente.

Ein-Ausgänge

VAR_INPUT

```
iColumn : INT;
iRow    : INT;
bWrt   : BOOL;
bRd    : BOOL;
usiGrp  : USINT;
lrCnr1X : LREAL;
lrCnr1Y : LREAL;
lrWdwWdth : LREAL;
lrWdwHght : LREAL;
```

iColumn: Spalten-Index des gewählten Elementes auf der Fassade. Dies bezieht sich auf die Auswahl eines Feldelementes des unter der IN-OUT-Variablen *arrFcdElem* angelegten Arrays.

iRow: dto. Reihen-Index. **und dürfen nicht Null sein!** *iRowiColumn* Das ergibt sich aus der Felddefinition, welche jeweils mit 1 beginnt, siehe oben.

bRd: Mit einer positiven Flanke an diesem Eingang werden die Informationen des gewählten Elementes, *arrFcdElem[iColumn,iRow]* in den Baustein gelesen und den Eingangsvariablen *usiGrp* bis *lrWdwHght* zugewiesen. Daraus ergeben sich dann die Ausgangsvariablen *lrCnr2X* bis *lrCnr4Y*. Sind zu Zeitpunkt des Auslesens schon Daten an den Eingängen *usiGrp* bis *lrWdwHght* angelegt, so werden die zuvor ausgelesenen Daten sofort mit diesen überschrieben.

bWrt: Eine positive Flanke schreibt die eingegebenen sowie errechneten Werte in das gewählte Feldelement *arrFcdElem[iColumn,iRow]*.

usiGrp: Gruppenzugehörigkeit.

IrCnr1X: X-Koordinate des Eckpunktes1 [m].

IrCnr1Y: Y-Koordinate des Eckpunktes1 [m].

IrWdwWdth: Fensterbreite [m].

IrWdwHght: Fensterhöhe [m].

Ein-/Ausgänge

VAR_OUTPUT

```
lrCnr2X : LREAL;
lrCnr2Y : LREAL;
lrCnr3X : LREAL;
lrCnr3Y : LREAL;
lrCnr4X : LREAL;
lrCnr4Y : LREAL;
bErr    : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

IrCnr2X: Ermittelte X-Koordinate des Eckpunktes 2 des Fensters [m]. Siehe "[Anmerkung \[▶ 262\]](#)" oben.

IrCnr2Y: Ermittelte Y-Koordinate des Eckpunktes 2 des Fensters [m]. Siehe "[Anmerkung \[▶ 262\]](#)" oben.

IrCnr3X: Ermittelte X-Koordinate des Eckpunktes 3 des Fensters [m]. Siehe "[Anmerkung \[▶ 262\]](#)" oben.

IrCnr3Y: Ermittelte Y-Koordinate des Eckpunktes 3 des Fensters [m]. Siehe "[Anmerkung \[▶ 262\]](#)" oben.

IrCnr4X: Ermittelte X-Koordinate des Eckpunktes 4 des Fensters [m]. Siehe "[Anmerkung \[▶ 262\]](#)" oben.

IrCnr4Y: Ermittelte Y-Koordinate des Eckpunktes 4 des Fensters [m]. Siehe "[Anmerkung \[▶ 262\]](#)" oben.

bErr : Ergebnis Kontrolle für die eingegebenen Werte.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[▶ 340\]](#).

VAR_IN_OUT

```
arrFcdElem : ARRAY[1..gBA_cMaxColumnFcd, 1..gBA_cMaxRowFcd] OF ST_BA_FcdElem;
```

arrFcdElem: [Liste von Fassadenelementen \[▶ 46\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

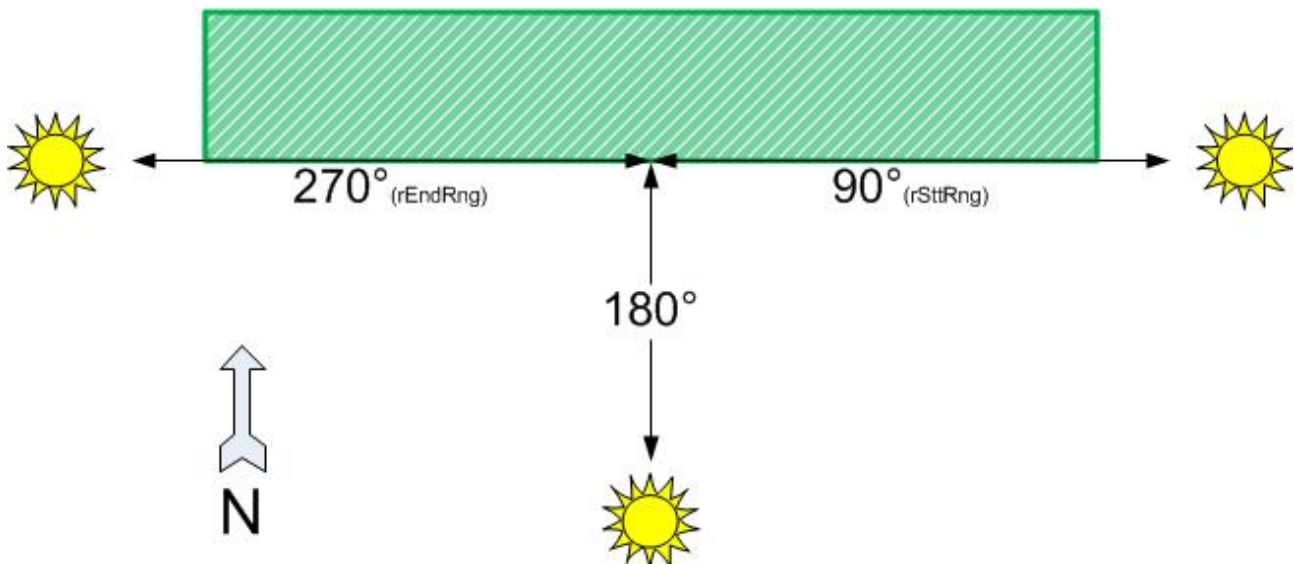
8.2.105 FB_BA_InRngAzm

Dieser Baustein prüft, ob der aktuelle Azimutwinkel (horizontaler Sonnenstand) innerhalb der eingetragenen Grenzen liegt. Wie in der [Übersicht \[▶ 45\]](#) erkennbar, gibt der Baustein eine zusätzliche Bewertung, ob der Sonnenschutz einer Fenstergruppe aktiviert werden soll. Daher gelten die Betrachtungen im weiteren Text immer für eine Fenstergruppe.

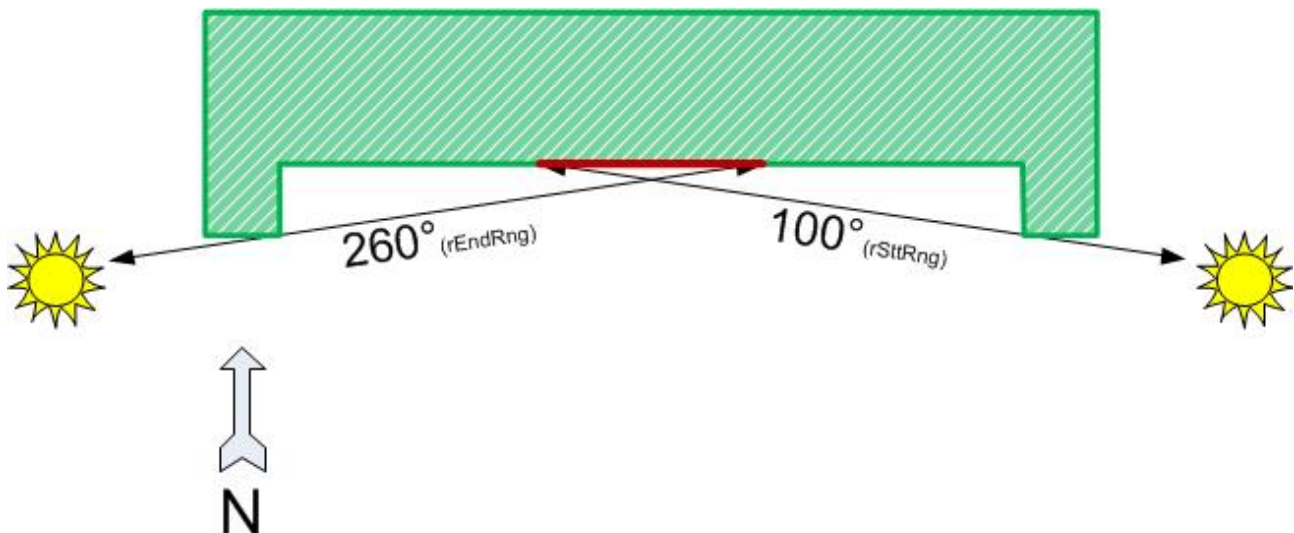
FB_BA_InRngAzm	
lrAzm	bOut
lrSttRng	bErr
lrEndRng	udiErrId
	udiErrArg

Funktionsbeschreibung

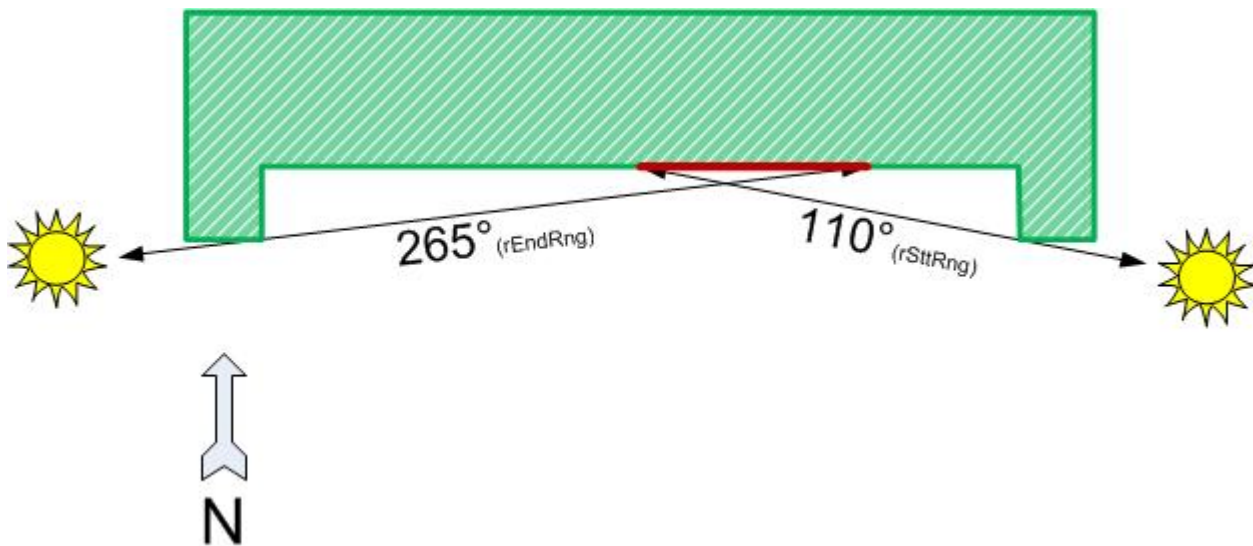
Eine glatte Fassade wird von der Sonne immer in einem Azimutwinkel von *Fassadenausrichtung-90°*... *Fassadenausrichtung+90°* bestrahlt.



Hat die Fassade jedoch seitliche Vorsprünge, so wird dieser Bereich eingeschränkt. Diese Einschränkung lässt sich mit Hilfe dieses Bausteins überprüfen. Dabei spielt aber auch die Lage der Fenstergruppe auf der Fassade eine Rolle. Liegt sie zentral, so ergibt sich folgende Situation (Die Werte sind dabei nur beispielhaft):



Für eine Gruppe am Rand ändern sich die Werte:



Der Anfang des Bereiches *lrSttRng* darf dabei größer sein als das Ende *lrEndRng*, es wird dann über 0° hinaus betrachtet.

Tab. 1: Beispiel:

<i>lrAzm</i>	10.0°
<i>lrSttRng</i>	280.0°
<i>lrEndRng</i>	20.0°
bOut	TRUE

Der betrachtete Bereich darf jedoch nicht größer als 180° oder gleich 0° sein, dieses wäre unrealistisch. Derartige Eingaben ergeben einen Fehler am Ausgang *bErr* - der Prüfausgang *bOut* wird dabei zusätzlich auf FALSE gesetzt.

Ein-/Ausgänge

VAR_INPUT

```
lrAzm : LREAL;
lrSttRng : LREAL;
lrEndRng : LREAL;
```

lrAzm: Aktueller Azimutwinkel

lrSttRng: Bereichsanfang [°]

lrEndRng: Bereichsende [°]

VAR_OUTPUT

```
bOut : BOOL;
bErr : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

bOut: Das Fassadenelement liegt in der Sonne wenn der Ausgang TRUE ist.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

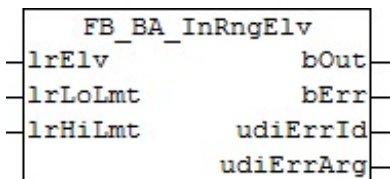
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

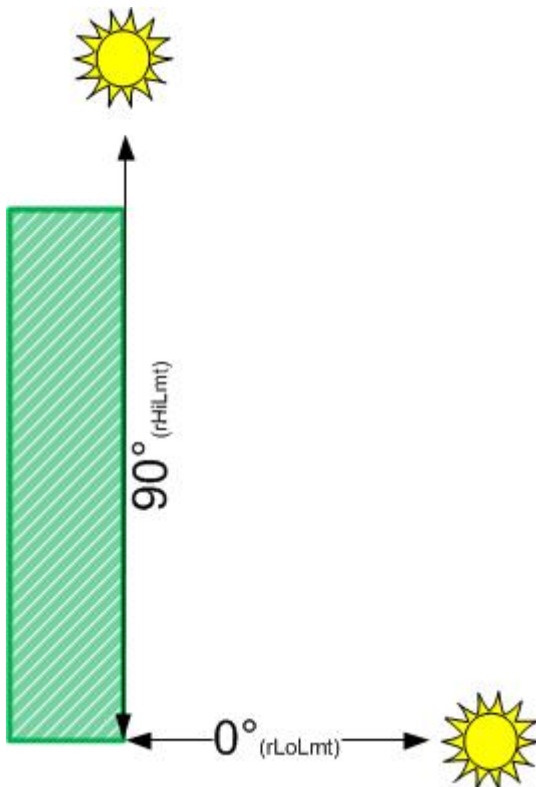
8.2.106 FB_BA_InRngElv

Dieser Baustein prüft, ob der aktuelle Elevationswinkel (vertikaler Sonnenstand) innerhalb der eingetragenen Grenzen liegt. Wie in der [Übersicht \[► 45\]](#) erkennbar, gibt der Baustein eine zusätzliche Bewertung, ob der Sonnenschutz einer Fenstergruppe aktiviert werden soll. Daher gelten die Betrachtungen im weiteren Text immer für eine Fenstergruppe.

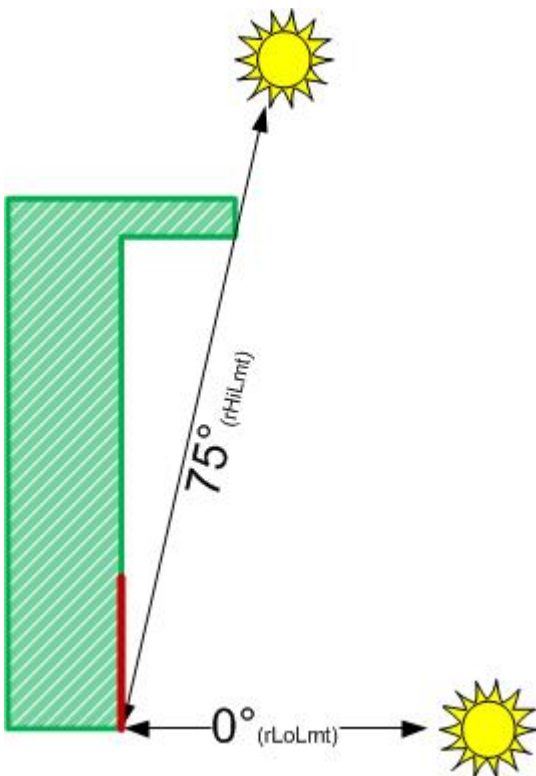


Funktionsbeschreibung

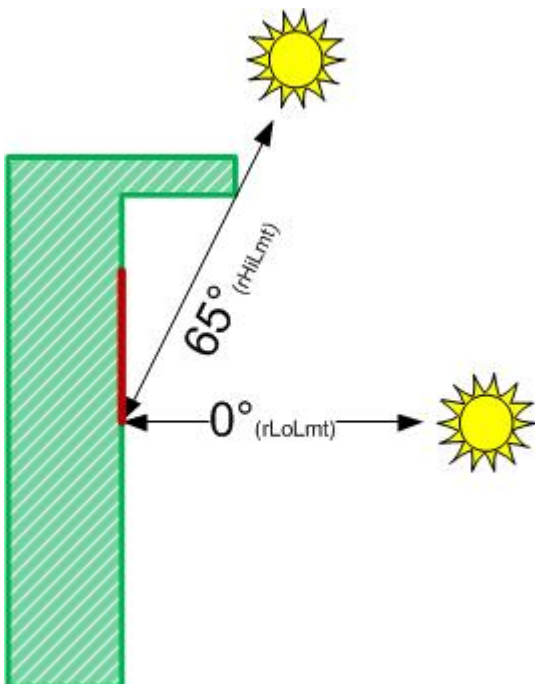
Eine normale senkrechte Fassade wird von der Sonne immer in einem Elevationswinkel von 0° bis maximal 90° bestrahlt.



Hat die Fassade jedoch Vorsprünge, so wird dieser Bereich eingeschränkt. Diese Einschränkung lässt sich mit Hilfe dieses Bausteins überprüfen. Dabei spielt aber auch die Lage der Fenstergruppe auf der Fassade eine Rolle. Liegt sie im unteren Bereich, so ergibt sich folgende Situation (Die Werte sind dabei nur beispielhaft):



Für eine Gruppe unterhalb des Vorsprunges ändern sich die Werte:



Die untere Betrachtungsgrenze, $lrLoLmt$, darf dabei nicht größer oder gleich der oberen, $lrHiLmt$, sein. Derartige Eingaben ergeben einen Fehler am Ausgang $bErr$ - der Prüfausgang $bOut$ wird dabei zusätzlich auf FALSE gesetzt.

Ein-/Ausgänge

VAR_INPUT

```
lrElv : LREAL;
lrLoLmt : LREAL;
lrHiLmt : LREAL;
```

lrElv: Aktueller Elevationswinkel [°]

IrLoLmt: Unterer Grenzwert [°]

IrHiLmt: Oberer Grenzwert [°]

VAR_OUTPUT

```
bOut      : BOOL;
bErr      : BOOL;
udiErrId  : UDINT;
udiErrArg : UDINT;
```

bOut: Das Fassadenelement liegt in der Sonne.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

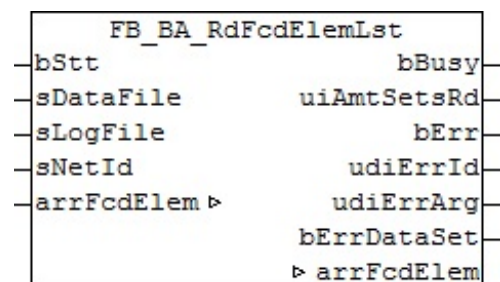
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.107 FB_BA_RdFcdElemLst

Mit Hilfe dieses Bausteins lassen sich Daten für Fassadenelemente (Fenster) aus einer vordefinierten Excel-Tabelle im csv-Format in die [Liste der Fassadenelemente](#) [► 46] importieren. Zusätzlich werden die importierten Daten auf Plausibilität überprüft und Fehler in eine Log-Datei geschrieben.



Funktionsbeschreibung

Excel-Tabelle

Das folgende Beispiel zeigt die Excel-Tabelle mit den Einträgen der Fensterelemente.

Alle Textfelder sind frei beschreibbar, wichtig sind die grün markierten Felder, wobei dort jede Zeile einen Datensatz kennzeichnet.

Folgende Regeln sind zu beachten:

- Ein Datensatz muss immer mit einem '@' beginnen.
- Die Indizes *IndexColumn* und *IndexRow* müssen innerhalb der definierten Grenzen liegen, siehe [Liste von Fassadenelementen](#) [► 46]. Mit diesen Indizes wird direkt beschrieben, in welches Fassadenelement der Liste *arrFcdElem*s die Daten des Satzes hineingelegt werden.
- Fensterbreite und Fensterhöhe müssen größer Null sein
- Die Eck-Koordinaten P1x und P1y müssen größer oder gleich Null sein.
- Jedes Fensterelement muss einer Gruppe 1..255 zugeordnet sein.
- Die Gesamtgröße der Tabelle darf systembedingt 65534 Bytes nicht überschreiten.
- Die Tabelle muss in Excel als Dateityp "CSV (Trennzeichen-getrennt) (*.csv)" abgespeichert worden sein.

Es ist nicht notwendig, alle Fensterelemente zu beschreiben, die per Definition bzw. Deklaration möglich wären. Bevor die neue Liste eingelesen wird, löscht der Baustein die gesamte alte Liste im Programm. Alle Elemente, welche dann nicht durch Einträge in der Excel-Tabelle beschrieben werden, haben reine Null-Einträge und sind dadurch als nicht-vorhanden gekennzeichnet und auch nicht auswertbar, da der Baustein zur Verschattungskorrektur, [FB_BA_ShdCorr](#) [▶ 281], Elemente mit Gruppeneintrag '0' nicht annimmt.

DE_FacadeElements.xls [Kompatibilitätsmodus]										
	A	B	C	D	E	F	G	H	I	J
1	Nummer	Beschreibung		IndexColumn	IndexRow	Fensterbreit	Fensterhoeh	P1x	P1y	Gruppe
2				(Achse)	(Stockwerk)	[m]	[m]	[m]	[m]	
3		Text								
4	1	Beschreibung	@	1	1	1,2	1,3	1,5	1	2
5	2	Beschreibung	@	0	1	1,2	1,3	2,7	1	2
6	3	Beschreibung	@	3	1	1,2	1,3	4,4	1	2
7	4	Beschreibung	@	4	1	1,2	1,3	6,1	1	2
8	5	Beschreibung	@	5	1	1,2	1,3	7,8	1	2
9	6	Beschreibung	@	6	1	1,2	1,3	9,5	1	2
10	7	Beschreibung	@	7	1	1,2	1,3	11,2	1	2
11	8	Beschreibung	@	8	1	1,2	1,3	12,9	1	2
12	9	Beschreibung	@	9	1	1,2	1,3	14,6	1	2
13	10	Beschreibung	@	10	1	1,2	1,3	16,3	1	2
14	11	Beschreibung	@	1	1	1,2	1,3	1,5	4	2
15	12	Beschreibung	@	0	1	1,2	1,3	2,7	4	2
16	13	Beschreibung	@	3	1	1,2	1,3	4,4	4	2
17	14	Beschreibung	@	4	1	1,2	1,3	6,1	4	2
18	15	Beschreibung	@	5	1	1,2	1,3	7,8	4	2
19	16	Beschreibung	@	6	1	1,2	1,3	9,5	4	2
20	17	Beschreibung	@	7	1	1,2	1,3	11,2	4	2
21	18	Beschreibung	@	8	1	1,2	1,3	12,9	4	2
22	19	Beschreibung	@	9	1	1,2	1,3	14,6	4	2
23	20	Beschreibung	@	10	1	1,2	1,3	16,3	4	2
24	21	Beschreibung	@	1	1	1,2	1,3	1,5	7	2
25	22	Beschreibung	@	0	1	1,2	1,3	2,7	7	2
26	23	Beschreibung	@	3	1	1,2	1,3	4,4	7	2
27	24	Beschreibung	@	4	1	1,2	1,3	6,1	7	2
28	25	Beschreibung	@	5	1	1,2	1,3	7,8	7	2
29	26	Beschreibung	@	6	1	1,2	1,3	9,5	7	2
30	27	Beschreibung	@	7	1	1,2	1,3	11,2	7	2
31	28	Beschreibung	@	8	1	1,2	1,3	12,9	7	2
32	29	Beschreibung	@	9	1	1,2	1,3	14,6	7	2
33	30	Beschreibung	@	10	1	1,2	1,3	16,3	7	2
34	31	Beschreibung	@	1	1	1,2	1,3	1,5	10	2
35	32	Beschreibung	@	0	1	1,2	1,3	2,7	10	2
36	33	Beschreibung	@	3	1	1,2	1,3	4,4	10	2
37	34	Beschreibung	@	4	1	1,2	1,3	6,1	10	2
38	35	Beschreibung	@	5	1	1,2	1,3	7,8	10	2
39	36	Beschreibung	@	6	1	1,2	1,3	9,5	10	2
40	37	Beschreibung	@	7	1	1,2	1,3	11,2	10	2
41	38	Beschreibung	@	8	1	1,2	1,3	12,9	10	2
42	39	Beschreibung	@	9	1	1,2	1,3	14,6	10	2
43	40	Beschreibung	@	10	1	1,2	1,3	16,3	10	2
44										

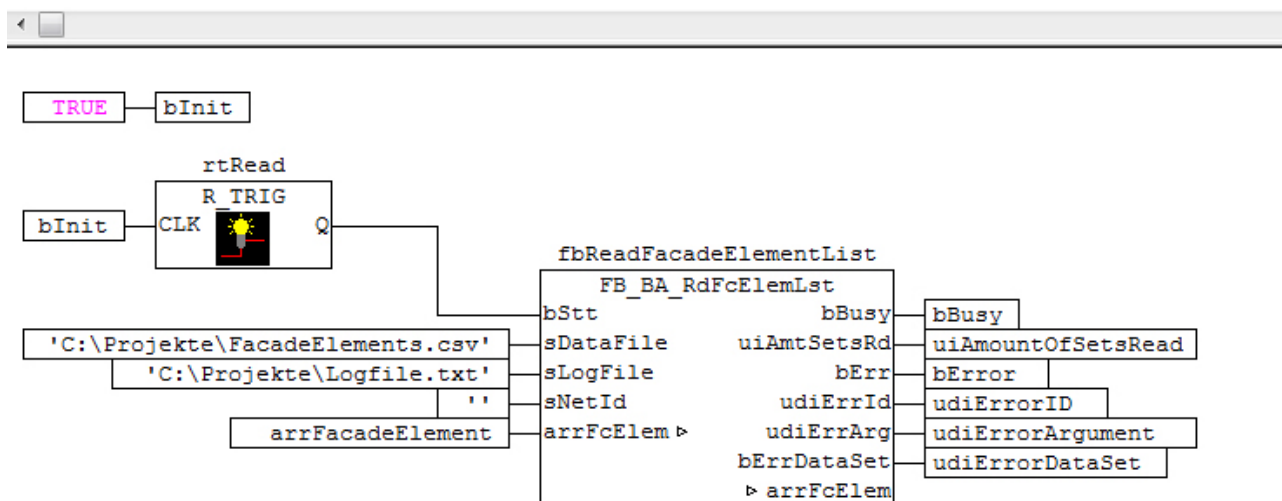
Log-Datei

Bei jedem Start des Lesebausteins wird die Logdatei neu beschrieben und der alte Inhalt gelöscht. Ist die Log-Datei nicht vorhanden, so wird sie zunächst automatisch erstellt. Die Log-Datei enthält dann entweder eine OK Meldung oder eine Auflistung aller aufgetretenen Fehler. Fehler, welche mit dem Öffnen, Beschreiben oder Schließen der Logdatei selbst in Zusammenhang stehen, können nicht mitgeschrieben werden. Daher sind immer auch die Ausgänge *udiErrId* / *udiErrArg* des Lesebausteins zu beachten, welche den letzten Fehlercode anzeigen. Da im Ablauf des Lesens die Log-Datei immer als Letztes geschlossen wird, ist ein entsprechender Hinweis im Fehlerfall gewährleistet.

Programmbeispiel

```
PROGRAM ReadFacadeElements
VAR
  bInit          : BOOL;
  rtRead        : R_TRIG;
  fbReadFacadeElementList : FB_BA_RdFcElemLst;
  arrFacadeElement : ARRAY[1..gBA_cMaxColumnFc, 1..gBA_cMaxRowFc] OF ST_BA_FcElem;

  bBusy        : BOOL;
  uiAmountOfSetsRead : UINT;
  bError       : BOOL;
  udiErrorID   : UDINT;
  udiErrorArgument : UDINT;
  udiErrorDataSet : UDINT;
END_VAR
```



In diesem Beispiel wird bei PLC-Start die Variable *bInit* zunächst auf TRUE gesetzt. Damit erhält der Eingang *bStt* am Baustein *fbReadFacadeElementList* einmalig eine steigende Flanke, welcher den Lesevorgang auslöst. Gelesen wird die Datei "FacadeElements.csv", welche sich im Ordner "C:\Projekte\" befindet. In demselben Ordner wird dann die Logdatei "Logfile.txt" hinterlegt. Ist diese Logdatei noch nicht vorhanden wird sie erstellt, andernfalls wird der bestehende Inhalt überschrieben. Das Lesen und Schreiben erfolgt auf demselben Rechner, auf dem sich auch die PLC befindet. Das wird durch den Eingang *sNetID* = "" (=lokal) definiert. Alle Daten werden in die im Programm deklarierte Liste *arrFcdElem* geschrieben. Solange gelesen und geschrieben wird ist der Ausgang *bBusy* auf TRUE. Der zuletzt auftretende Fehler in Bezug auf das Dateihandling wird an *udiErrId* / *udiErrArg* angezeigt, *bErr* steht dann auf TRUE. Wird hingegen ein Fehler im Datensatz erkannt wird dies an *bErrDataSet* angezeigt und in der Log-Datei näher beschrieben. Die Anzahl der gefundenen und gelesenen Datenzeilen wird an *uiAmtSetsRd* zur Kontrolle angezeigt.

In die folgende Excel Liste wurden die markierten Fehler "eingebaut". Dadurch ergibt sich das gezeigte Log-File:

	A	B	C	D	E	F	G	H	I	J
1	Nummer	Beschreibung	@	IndexColumn (Achse)	IndexRow (Stockwerk)	Fensterbreite [m]	Fensterhoehe [m]	P1x [m]	P1y [m]	Gruppe
3		Text								
4	1	Beschreibung	@	1	1	1,2	1,3	1,5	1,4	2
5	2	Beschreibung	@	0	1	1,2	1,3	2,7	1	2
6	3	Beschreibung	@	3	1	1,2	1,3	4,4	1	2
7	4	Beschreibung	@	4	1	1,2	1,3	6,1	1	2
8	5	Beschreibung	@	5	1	1,2	1,3	7,8	1	2
9	6	Beschreibung	@	6	1	0	1,3	9,5	1	2
10	7	Beschreibung	@	7	1	1,2	1,3	11,2	1	2
11	8	Beschreibung	@	8	1	1,2	1,3	12,9	1	2
12	9	Beschreibung	@	9	1	1,2	1,3	14,6	1	2
13	10	Beschreibung	@	10	1	1,2	1,3	16,3	1	5
14	11	Beschreibung	@	1	2	1,2	1,3	1	1	5
15	12	Beschreibung	@	2	2	1,2	1,3	2,7	3	5
16	13	Beschreibung	@	3	2	1,2	1,3	4,4	4	5
17	14	Beschreibung	@	4	2	1,2	1,3	4,4	4	5
18	15	Beschreibung	@	5	2	1,2	1,3	7,8	4	5
19	16	Beschreibung	@	6	2	1,2	1,3	9,5	4	5
20	17	Beschreibung	@	7	2	1,2	1,3	11,2	4	5
21	18	Beschreibung	@	8	2	1,2	1,3	12,9	4	5
22	19	Beschreibung	@	9	2	1,2	1,3	14,6	4	3
23	20	Beschreibung	@	10	2	1,2	1,3	16,3	4	3
24										
25	31	Beschreibung	@	1	3	1,2	1,3	1	7	3
26	32	Beschreibung	@	2	3	1,2	1,3	-1	6	3
27	33	Beschreibung	@	3	3	1,2	1,3	4,4	7	3
28	34	Beschreibung	@	4	3	1,2	1,3	6,1	7	0
29	35	Beschreibung	@	5	3	1,2	1,3	7,8	7	3
30	36	Beschreibung	@	6	3	1,2	1,3	9,5	7	3
31	37	Beschreibung	@	7	3	1,2	1,3	11,2	7	3
32	38	Beschreibung	@	8	3	1,2	1,3	12,9	7	7
33	39	Beschreibung	@	9	3	1,2	1,3	14,6	7	7
34	40	Beschreibung	@	10	3	1,2	1,3	16,3	7	7

LogFacade.txt - Editor

Datei Bearbeiten Format Ansicht ?

Index-Error in Data-Set #2
 Validation-Error in Data-Set #6, ErrId=33056, ErrArg=5
 Validation-Error in Data-Set #22, ErrId=33056, ErrArg=3
 Validation-Error in Data-Set #24, ErrId=33056, ErrArg=2

Der erste Fehler ist im Datensatz 2 und ein Index-Fehler, da "0" nicht erlaubt ist. Der nächste Fehler in Datensatz 6 wurde nach Validierung der Daten mit dem intern genutzten Baustein [FB_BA_ShdObjEntry](#) [▶ 284] gefunden und daher näher mit einer Fehlernummer versehen, welche unter [Fehlercodes](#) [▶ 340] näher aufgeschlüsselt wird. Der dritte und der vierte Fehler trat ebenfalls nach der internen Validierung auf.

Wichtig hierbei ist, dass sich die Datensatz-Nummern (hier 22 und 24) nicht nach den eingetragenen Nummern in der Liste, sondern nach den tatsächlichen laufenden Nummern richten: eingelesen wurden hier lediglich 30 Datensätze.

Ein-/Ausgänge

VAR_INPUT

```
bStt      : BOOL;
sDataFile : STRING;
sLogFile  : STRING;
sNetId    : T_AmsNetId;
```

bStt: Eine TRUE-Flanke an diesem Eingang startet den Lesevorgang.

sDataFile: Enthält den Pfad- und Dateinamen der zu öffnenden Daten-Datei. Diese muss in Excel als Dateityp "CSV (Trennzeichen-getrennt) (*.csv)" abgespeichert worden sein. Öffnet man die Datei mit einem einfachen Text-Editor, so müssen die Werte durch Semikola getrennt dargestellt sein. Beispiel für einen Eintrag: *sDataFile:= 'C:\Projekte\FacadeElements.csv'*

sLogFile: dto. Log-File für die auflaufenden Fehler. Diese Datei wird bei jeder neuen Bausteinaktivierung überschrieben, so dass nur aktuelle Fehler enthalten sind.

sNetId: Hier kann ein String mit der AMS Net Id des TwinCAT-Rechners angegeben werden, auf dem die Dateien geschrieben/gelesen werden sollen. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.



Die Speicherung der Daten kann nur auf dem Steuerungsrechner selbst und den Rechnern erfolgen, die mit dem Steuerungsrechner per ADS verbunden sind. Verweise auf lokale Festplatten dieser Rechner sind möglich, nicht jedoch auf verbundene Netzwerkfestplatten.

VAR_OUTPUT

```
bBusy      : BOOL;
uiAmtSetsRd : UINT;
bErr       : BOOL;
udiErrId   : UDINT;
udiErrArg  : UDINT;
bErrDataSet : BOOL;
```

bBusy: Dieser Ausgang steht auf TRUE, solange Elemente aus der Datei ausgelesen werden.

uiAmtSetsRd: Anzahl der gelesenen Datensätze

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn ein Datei-Schreib- bzw. Lesefehler aufgetreten ist.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

bErrDataSet: Dieser Ausgang wird auf TRUE geschaltet, wenn die gelesenen Datensätze fehlerhaft sind. Nähere Einzelheiten werden in die Log-Datei geschrieben.

VAR_IN_OUT

```
arrFcdElem : ARRAY[1..gBA_cMaxColumnFcd, 1..gBA_cMaxRowFcd] OF ST_BA_FcdElem;
```

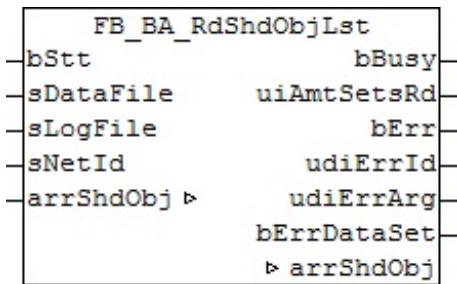
arrFcdElem: [Liste von Fassadenelementen](#) [► 46].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.108 FB_BA_RdShdObjLst

Mit Hilfe dieses Bausteins lassen sich Daten für Verschattungsobjekte aus einer vordefinierten Excel-Tabelle im csv-Format in die [Liste der Verschattungsobjekte](#) [► 46] importieren. Zusätzlich werden die importierten Daten auf Plausibilität überprüft und Fehler in eine Log-Datei geschrieben.



Funktionsbeschreibung

Excel-Tabelle

Das folgende Beispiel zeigt die Excel-Tabelle mit den Einträgen der Fensterelemente. Alle Textfelder sind frei beschreibbar, wichtig sind die grün markierten Felder, wobei dort jede Zeile einen Datensatz kennzeichnet. Je nachdem ob es sich um den Typ Viereck oder Kugel handelt, haben die Spalten G bis J eine unterschiedliche Bedeutung. Die Spalten K bis M sind bei Kugeln freizulassen. Bezüglich der Vierecks-Koordinaten werden nur die relevanten Daten eingegeben und die restlichen intern berechnet, siehe [FB_BA_ShObjEntry \[▶ 284\]](#).

Folgende Regeln sind zu beachten:

- Ein Datensatz muss immer mit einem '@' beginnen.
- Die Monatseinträge dürfen nicht 0 und größer 12 sein, andernfalls sind alle Kombinationen möglich.
Beispiele:
 Beginn=1, Ende=1: Verschattung im Januar.
 Beginn=1, Ende=5: Verschattung von Anfang Januar bis Ende Mai.
 Beginn=11, Ende=5: Verschattung von Anfang November bis Ende Mai (des folgenden Jahres).
- Fensterbreite und Fensterhöhe müssen größer Null sein
- Die z-Koordinaten P1z und P3z bzw. Mz müssen größer Null sein.
- Der Radius muss größer Null sein.
- Die Gesamtgröße der Tabelle darf systembedingt 65534 Bytes nicht überschreiten.
- Die Tabelle muss in Excel als Dateityp "CSV (Trennzeichen-getrennt) (*.csv)" abgespeichert worden sein.

Es ist nicht notwendig, alle Verschattungsobjekte, welche pro Fassade möglich sind, zu beschreiben. Nur die, die in der Liste aufgeführt werden, kommen letztendlich zum Tragen.

DE_ShadingObjects.csv													
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Nummer	Beschreibung		Typ	Beginn	Ende	P1x/Mx	P1y/My	P1z/Mz	P2y/R	P3x	P3y	P3z
2				0 - Viereck	(Monat)	(Monat)	[m]	[m]	[m]	[m]	[m]	[m]	[m]
3				1 - Kugel									
4		Text											
5	1	Beschreibung	@	0	1	2	-94,75	0	36,06	11	-70,71	11	68,59
6	2	Beschreibung	@	0	1	2	-23,33	0	9,9	10,5	-3,54	10,5	22,62
7	3	Beschreibung	@	0	1	2	62,23	0	0	14,47	62,23	14,47	8
8	4	Beschreibung	@	0	1	2	46	0	13	14,47	62,23	14,47	8
9	5	Beschreibung	@	0	1	2	46	0	13	14,47	46	14,47	38,89
10	6	Beschreibung	@	0	1	2	0	0	14	9	35	9	14
11	7	Beschreibung	@	0	1	2	0	0	14	9,8	16	9,8	14
12	8	Beschreibung	@	0	1	2	23,6	0	14	9,8	25	9,8	14
13	9	Beschreibung	@	0	1	2	27,8	0	14	9,8	35	9,8	14
14													
15	10	Beschreibung	@	1	1	2	27	15	40	6			
16	11	Beschreibung	@	1	1	2	38	15	36	6			
17	12	Beschreibung	@	1	1	2	-14	4	4	1,5			
18	13	Beschreibung	@	1	1	2	-6,5	6	6	3,2			
19	14	Beschreibung	@	1	1	2	-7	9	6	1,2			
20	15	Beschreibung	@	1	1	2	-1	6	8	3,2			
21	16	Beschreibung	@	1	1	2	-1	9	8	1,2			

Log-Datei

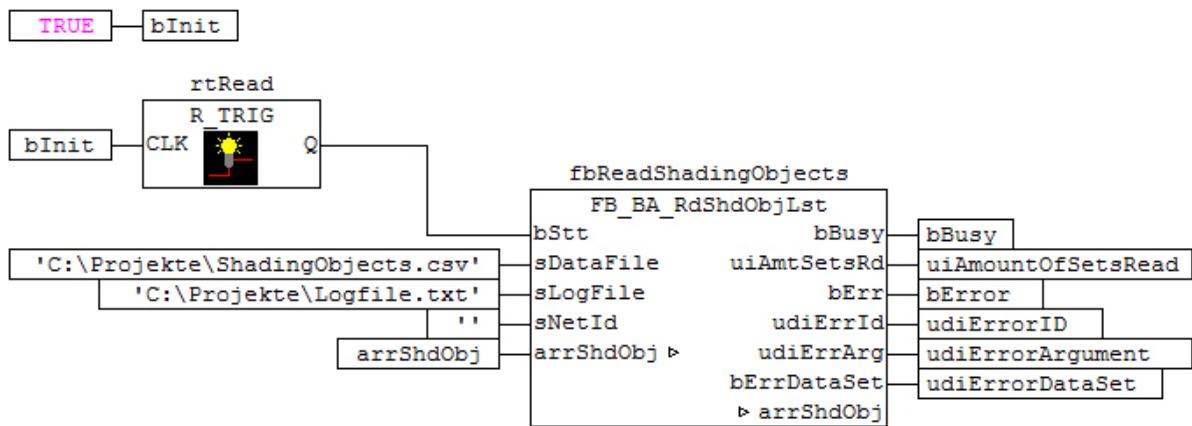
Bei jedem Start des Lesebausteins wird die Logdatei neu beschrieben und der alte Inhalt gelöscht. Ist die Log-Datei nicht vorhanden, so wird sie zunächst automatisch erstellt. Die Log-Datei enthält dann entweder eine OK Meldung oder eine Auflistung aller aufgetretenen Fehler. Fehler, welche mit dem Öffnen, Beschreiben oder Schließen der Logdatei selbst in Zusammenhang stehen, können nicht mitgeschrieben werden. Daher sind immer auch die Ausgänge *udiErrld* / *udiErrArg* des Lesebausteins zu beachten, welche den letzten Fehlercode anzeigen. Da im Ablauf des Lesens die Log-Datei immer als Letztes geschlossen wird, ist ein entsprechender Hinweis im Fehlerfall gewährleistet.

Programmbeispiel

```

PROGRAM ReadShadingObjects
VAR
    bInit          : BOOL;
    rtRead         : R_TRIG;
    fbReadShadingObjects : FB_BA_RdShdObjLst;
    arrShdObj      : ARRAY[1..gBA_cMaxShdObj] OF ST_BA_ShdObj;

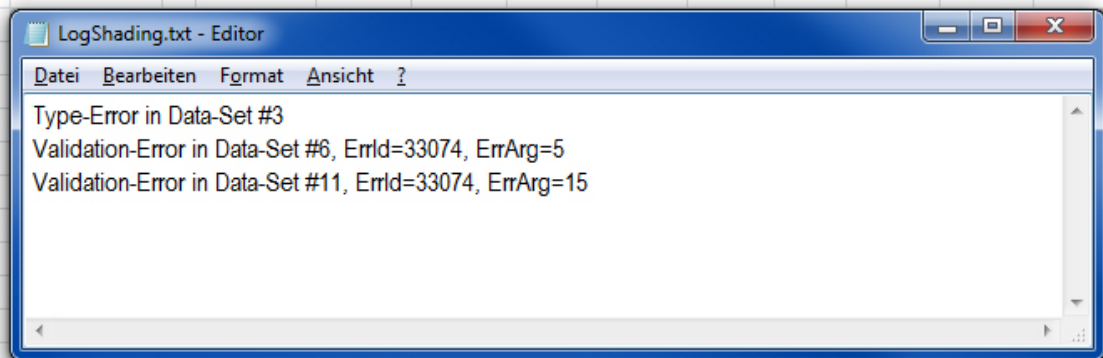
    bBusy         : BOOL;
    uiAmountOfSetsRead : UINT;
    bError        : BOOL;
    udiErrorID    : UDINT;
    udiErrorArgument : UDINT;
    udiErrorDataSet : UDINT;
END_VAR
    
```



In diesem Beispiel wird bei PLC-Start die Variable *bInit* zunächst auf TRUE gesetzt. Damit erhält der Eingang *bStt* am Baustein *fbReadShadingObjects* einmalig eine steigende Flanke, welche den Lesevorgang auslöst. Gelesen wird die Datei "ShadingObjects.csv", welche sich im Ordner "C:\Projekte\" befindet. In demselben Ordner wird dann die Logdatei "Logfile.txt" hinterlegt. Ist diese Logdatei noch nicht vorhanden wird sie erstellt, andernfalls wird der bestehende Inhalt überschrieben. Das Lesen und Schreiben erfolgt auf demselben Rechner, auf dem sich auch die PLC befindet. Das wird durch den Eingang *sNetID* = " (=lokal) definiert. Alle Daten werden in die im Programm deklarierte Liste *arrShdObj* geschrieben. Solange gelesen und geschrieben wird ist der Ausgang *bBusy* auf TRUE. Der zuletzt auftretende Fehler in Bezug auf das Dateihandling wird an *udiErrId* / *udiErrArg* angezeigt, *bErr* steht dann auf TRUE. Wird hingegen ein Fehler im Datensatz erkannt wird dies an *bErrDataSet* angezeigt und in der Log-Datei näher beschrieben. Die Anzahl der gefundenen und gelesenen Datenzeilen wird an *uiAmtSetsRd* zur Kontrolle angezeigt.

In die folgende Excel Liste wurden die markierten Fehler eingebaut. Dadurch ergibt sich das gezeigte Log-File:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Nummer	Beschreibung		Typ	Beginn	Ende	P1x/Mx	P1y/My	P1z/Mz	P2y/R	P3x	P3y	P3z
2				0 - Viereck	(Monat)	(Monat)	[m]	[m]	[m]	[m]	[m]	[m]	[m]
3				1 - Kugel									
4		Text											
5	1	Beschreibung	@	0	1	2	-94,75	0	36,06	11	-70,71	11	68,59
6	2	Beschreibung	@	0	1	2	-23,33	0	9,9	10,5	-3,54	10,5	22,62
7	3	Beschreibung	@	2	1	2	62,23	0	0	14,47	62,23	14,47	8
8	4	Beschreibung	@	0	1	2	46	0	13	14,47	62,23	14,47	8
9	5	Beschreibung	@	0	1	2	46	0	13	14,47	46	14,47	38,89
10	6	Beschreibung	@	0	1	2	0	0	14	9	35	9	-14
11	7	Beschreibung	@	0	1	2	0	0	14	9,8	16	9,8	14
12	8	Beschreibung	@	0	1	2	23,6	0	14	9,8	25	9,8	14
13	9	Beschreibung	@	0	1	2	27,8	0	14	9,8	35	9,8	14
14													
15	11	Beschreibung	@	1	1	2	27	15	40	6			
16	12	Beschreibung	@	1	1	13	38	15	36	6			
17	13	Beschreibung	@	1	1	2	-14	4	4	1,5			
18	14	Beschreibung	@	1	1	2	-6,5	6	6	3,2			
19	15	Beschreibung	@	1	1	2	-7	9	6	1,2			
20	16	Beschreibung	@	1	1	2	-1	6	8	3,2			
21	17	Beschreibung	@	1	1	2	-1	9	8	1,2			
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													
33													



Der erste Fehler ist im Datensatz 3 und ein Typ-Fehler, da "2" nicht definiert ist.

Der nächste Fehler in Datensatz 6 wurde nach Validierung der Daten mit dem intern genutzten Baustein [FB_BA_ShdObjEntry](#) [► 284] gefunden und daher näher mit einer Fehlernummer versehen, welche unter [Fehlercodes](#) [► 340] näher aufgeschlüsselt wird. Der dritte Fehler trat ebenfalls nach der internen Validierung auf.

Wichtig hierbei ist, dass sich die Datensatz-Nummer (hier 11) nicht nach der eingetragenen Nummer in der Liste, sondern nach der tatsächlichen laufenden Nummer richtet: eingelesen wurden hier lediglich 16 Datensätze.

Ein-/Ausgänge

VAR_INPUT

```
bStt      : BOOL;
sDataFile : STRING;
sLogFile  : STRING;
sNetId    : T_AmsNetId;
```

bStt: Eine TRUE-Flanke an diesem Eingang startet den Lesevorgang.

sDataFile: Enthält den Pfad- und Dateinamen der zu öffnenden Daten-Datei. Diese muss in Excel als Dateityp "CSV (Trennzeichen-getrennt) (*.csv)" abgespeichert worden sein. Öffnet man die Datei mit einem einfachen Text-Editor, so müssen die Werte durch Semikola getrennt dargestellt sein. Beispiel für einen Eintrag: `sDataFile:= 'C:\Projekte\ShadingObjects.csv'`

sLogFile: dto. Log-File für die auflaufenden Fehler. Diese Datei wird bei jeder neuen Bausteinaktivierung überschrieben, so dass nur aktuelle Fehler enthalten sind.

sNetId: Hier kann ein String mit der AMS Net Id des TwinCAT-Rechners angegeben werden, auf dem die Dateien geschrieben/gelesen werden sollen. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.



Die Speicherung der Daten kann nur auf dem Steuerungsrechner selbst und den Rechnern erfolgen, die mit dem Steuerungsrechner per ADS verbunden sind. Verweise auf lokale Festplatten dieser Rechner sind möglich, nicht jedoch auf verbundene Netzwerkfestplatten.

VAR_OUTPUT

```
bBusy      : BOOL;
uiAmtSetsRd : UINT;
bErr       : BOOL;
udiErrId   : UDINT;
udiErrArg  : UDINT;
bErrDataSet : BOOL;
```

bBusy: Dieser Ausgang steht auf TRUE, solange Elemente aus der Datei ausgelesen werden.

uiAmtSetsRd: Anzahl der gelesenen Datensätze

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn ein Datei-Schreib- bzw. Lesefehler aufgetreten ist.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [▶ 340].

bErrDataSet: Dieser Ausgang wird auf TRUE geschaltet, wenn die gelesenen Datensätze fehlerhaft sind. Nähere Einzelheiten werden in die Log-Datei geschrieben.

VAR_IN_OUT

```
arrShdObj: ARRAY[1..gBA_cMaxShdObj] OF ST_BA_ShdObj;
```

arrShdObj: Liste der Verschattungsobjekte [▶ 46].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

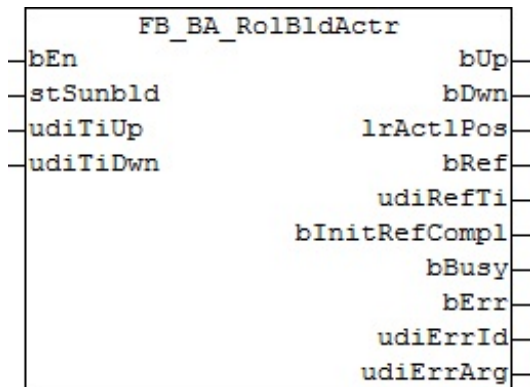
8.2.109 FB_BA_RoIBldActr

Dieser Baustein dient zur Positionierung einer Rollladen-Jalousie über zwei Ausgänge: Hoch- und Herunterfahren. Über das Positioniertelegramm [stSunBld \[▶ 337\]](#) kann die Jalousie auf eine beliebige Position gesteuert werden. Darüber hinaus beinhaltet das Positioniertelegramm [stSunBld \[▶ 337\]](#) auch Handbefehle, mit denen die Jalousie individuell auf bestimmte Positionen bewegt werden kann. Diese Handbefehle werden von dem Baustein [FB_BA_SunBldSwi \[▶ 301\]](#) angesteuert.

Der Baustein besitzt intern eine feste Umschaltverriegelung (Ausgang *bUp* zu Ausgang *bDwn*) von **500ms**.



Dieser Baustein muss in jedem SPS-Zyklus aufgerufen werden, da die SPS-Zykluszeit in die Berechnung der Positionen mit einfließt.



Funktionsbeschreibung

Struktur des Jalousie-Positioniertelegramms [stSunBld](#) [► 337].

```

TYPE ST_BA_SunBld:
STRUCT
  lrPos      : LREAL;
  lrAnagl   : LREAL;
  bManUp    : BOOL;
  bManDwn   : BOOL;
  bManMod   : BOOL;
  bActv     : BOOL;
END_STRUCT
END_TYPE

```

Die aktuelle Höhenposition und der Lamellenwinkel werden dabei nicht durch einen zusätzlichen Encoder eingelesen, sondern intern durch die Laufzeit der Jalousie ermittelt.

Durch die beiden unterschiedlichen Laufzeitparameter *udiTiUp* (Laufzeit Jalousie hoch [ms]) und *udiTiDwn* (Laufzeit Jalousie herunter [ms]) wird den unterschiedlichen Fahrcharakteristiken Rechnung getragen.

Der Baustein steuert die Jalousie grundsätzlich über die Informationen, aus dem Positioniertelegramm [stSunBld](#) [► 337]. Ist der Automatikmodus aktiv (*bManMod*=FALSE), so wird immer die aktuelle Position angefahren, wobei Änderungen sofort berücksichtigt werden. Im Handbetrieb (*bManMod*=TRUE) steuern die Befehle *bManUp* und *bManDwn* die Jalousie.

Referenzieren

Ein sicheres Referenzieren ist gegeben, wenn die Jalousie länger als ihre komplette Hochlaufzeit nach oben hin angesteuert wird. Die Position ist dann auf jeden Fall "0". Da eine Jalousiepositionierung ohne Encoder naturgemäß immer fehlerbehaftet ist, ist es wichtig möglichst oft automatisch zu referenzieren: jedes mal, wenn die Position "0" angefahren werden soll, fährt die Jalousie zunächst ganz normal mit kontinuierlicher Positionsberechnung nach oben. Erreicht sie den errechneten Positionswert 0%, so wird der Ausgang *bUp* noch einmal für die weiterhin gehalten und zwar noch einmal für die komplette Hochlaufzeit + 5s.

Aus Gründen der Flexibilität gibt es nun zwei Möglichkeiten, den Referenziervorgang zu unterbrechen: Bis zum Erreichen der errechneten 0%-Position wird eine Positionsänderung immer noch angenommen und ausgeführt, nach Erreichen dieser 0%-Position kann die Jalousie noch mit dem Handbefehl "herunterfahren" anders bewegt werden. Diese beiden sinnvollen Einschränkungen machen es nötig, dass der Nutzer selbst dafür Sorge trägt, die Jalousie so oft wie möglich sicher referenzieren zu lassen.

Beim Neustart der Anlage referenziert der Baustein in jedem Fall. Die abgeschlossene Initial-Referenzierung wird durch ein TRUE-Signal an Ausgang *bInitRefCmpl* angezeigt. Auch die Initial-Referenzierung kann durch einen Handbefehl "herunterfahren" vorzeitig abgeschlossen werden.

Ein-/Ausgänge

VAR_INPUT

```

bEn      : BOOL;
stSunBld : ST_BA_Sunblind;
udiTiUp  : UDINT;
udiTiDwn : UDINT;

```


bEn: Freigabeeingang für den Baustein. Solange dieser Eingang auf TRUE steht, nimmt der Aktorbaustein Befehle entgegen und arbeitet, wie oben beschrieben. Ein FALSE-Signal an diesen Eingang setzt die Steuerausgänge *bUp* und *bDwn* zurück und der Funktionsbaustein verharrt in einem Ruhezustand.

stSunBld: Positioniertelegramm, siehe [ST_BA_SunBld \[► 337\]](#)

udiTiUp: Komplette Hochfahrzeit [ms]

udiTiDwn: Komplette Zeit zum Herunterfahren in ms

VAR_OUTPUT

```
bUp          : BOOL;
bDwn        : BOOL;
lrActlPos   : LREAL;
udiRefTi    : UDINT;
bInitRefCompl : BOOL;
bRef        : BOOL;
bBusy       : BOOL;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bUp: Steuerausgang Jalousie hoch

bDwn: Steuerausgang Jalousie herunter

lrActlPos: Aktuelle Position in Prozent

udiRefTi: Referenzier-Countdown-Anzeige [s]

blnitRefCompl: Initial-Referenziervorgang abgeschlossen.

bRef: Die Jalousie befindet sich in der Referenzierung, d.h. für die die komplette Hochlaufzeit + 5s wird der Ausgang *bUp* gesetzt. Nur ein Handbefehl "herunter" kann die Jalousie in Gegenrichtung bewegen und diesen Modus beenden.

bBusy: Ein Positionier- oder Referenziervorgang findet statt.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

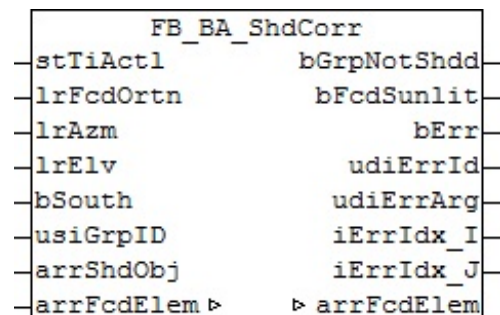
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.110 FB_BA_ShdCorr

Funktionsbaustein zur Verschattungsbeurteilung einer Fenstergruppe auf einer Fassade



Funktionsbeschreibung

Der Baustein FB_BA_ShdCorr berechnet für eine Gruppe von Fenstern, ob sich diese im Schatten von umliegenden Objekten befindet. Mit dem Ergebnis, welches am Ausgang *bGrpNotShdd* ausgegeben wird, kann beurteilt werden, ob ein Sonnenschutz für diese Fenstergruppe sinnvoll ist.

Dabei greift der Baustein auf zwei zu definierende Listen zu:

- Die Parameter, welche die Verschattungselemente beschreiben, die für die Fassade maßgeblich sind, auf der sich die Fenstergruppe befindet.. Diese Liste der Verschattungsobjekte [► 46] wird als Eingangsvariable *arrShdObj* an den Baustein angelegt, da die Informationen lediglich gelesen werden.
- Die Daten der Elemente (Fenster) der Fassade, in der sich die zu betrachtende Gruppe befindet. Auf diese Liste der Fassadenelemente [► 46] wird über die IN-OUT-Variable *arrFcdElem* zugegriffen, da nicht nur die Koordinaten der Fenster gelesen werden, sondern von dem Baustein FB_BA_ShdCorr in diese Liste auch jeweils die Verschattungsinformation für jede Fensterecke hinterlegt wird. Diese Information kann so in weiteren Teilen des Applikationsprogrammes genutzt werden.

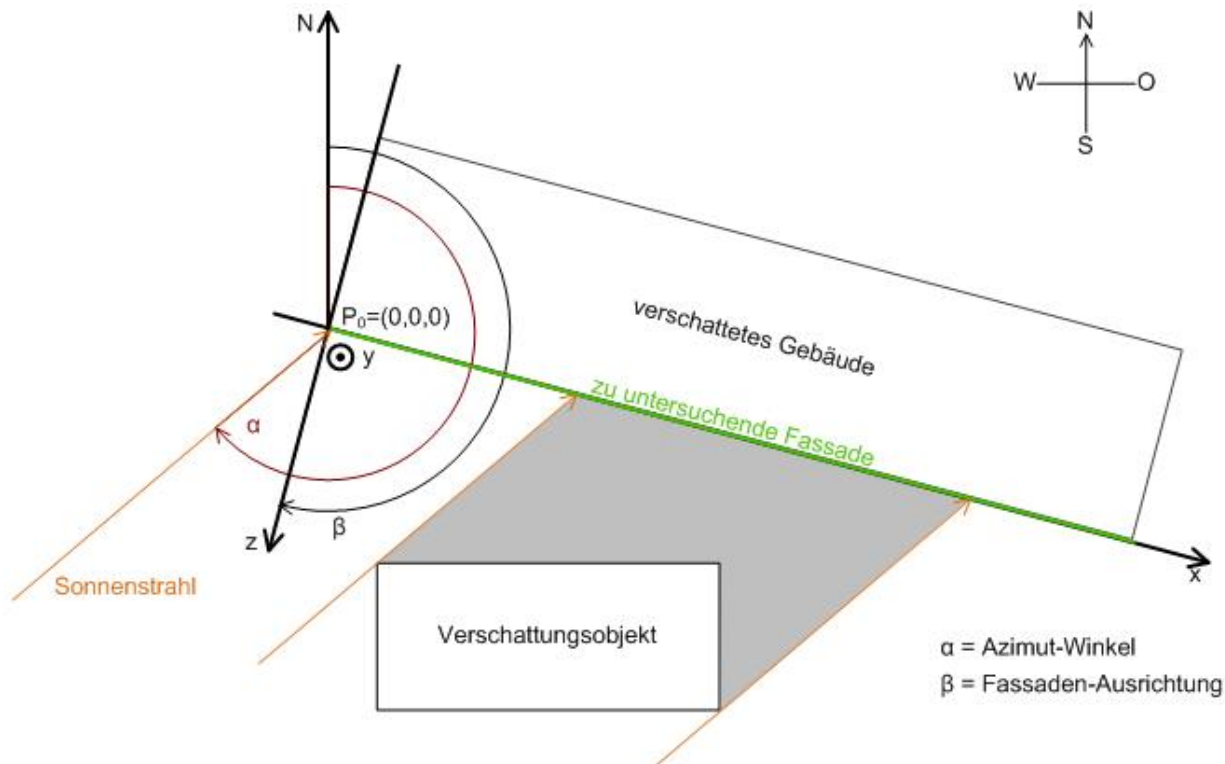
Anhand der Fassadenausrichtung (*IrFcdOrtn*), der Sonnenrichtung (*IrAzm*) und der Sonnenhöhe (*IrElv*) kann dann für jeden Eckpunkt eines Fensters errechnet werden, ob sich dieser im Bereich eines Schattens befindet. Eine Gruppe von Fenstern gilt dann als komplett verschattet, wenn alle Eckpunkte verschattet sind.

Dabei gilt auf der Nordhalbkugel für die Fassadenausrichtung (Blick aus dem Fenster):

Blickrichtung	Fassadenausrichtung
Nord	$\beta=0^\circ$
Ost	$\beta=90^\circ$
Süd	$\beta=180^\circ$
West	$\beta=270^\circ$

Der Baustein führt seine Berechnungen nur dann durch, wenn die Sonne tatsächlich auf die Fassade scheint. Betrachtet man die in der Einleitung vorgestellte Zeichnung, so ist dies gegeben wenn gilt:

Fassadenausrichtung < Azimutwinkel < Fassadenausrichtung + 180°

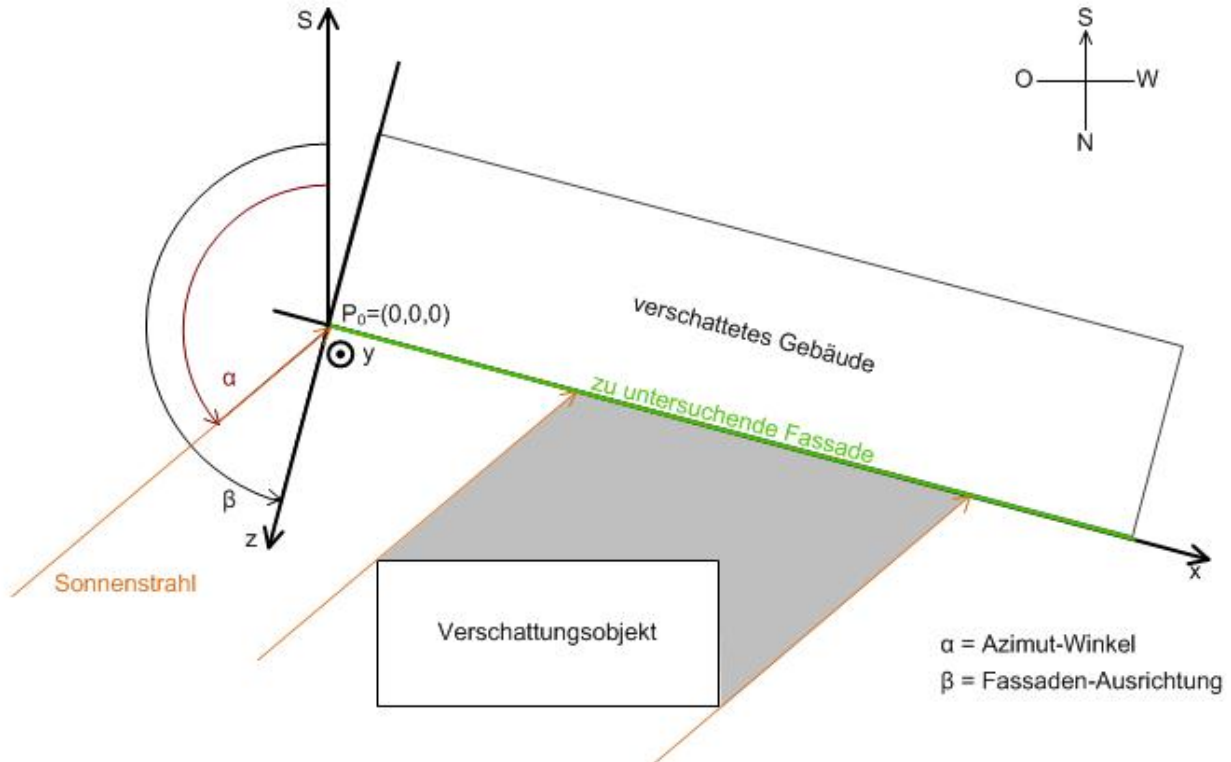


Darüber hinaus ist eine Berechnung auch dann nicht nötig, wenn die Sonne noch nicht aufgegangen ist, die Sonnenhöhe (Elevation) also unter 0° liegt. In beiden Fällen wird der Ausgang *bFcdSunlit* auf FALSE gesetzt.

Für die Südhalbkugel ändern sich die Verhältnisse. Es gilt für die Fassadenausrichtung (Blick aus dem Fenster):

Blickrichtung	Fassadenausrichtung
Süd	$\beta=0^\circ$
Ost	$\beta=90^\circ$
Nord	$\beta=180^\circ$
West	$\beta=270^\circ$

Auch ändert sich die interne Berechnung Fassade zu Sonnenstrahl:



Zur Unterscheidung der Verhältnisse - Nord- oder Südhalbkugel - ist der Eingangsparameter *bSouth* auf FALSE zu setzen (Nordhalbkugel) oder auf TRUE (Südhalbkugel)

Ein-/Ausgänge

VAR_INPUT

```

stTiAct1 : TIMESTRUCT;
lrFcdOrtn : LREAL;
lrAzm : LREAL;
lrElv : LREAL;
bSouth : BOOL;
usiGrpID : USINT;
arrShdObj : ARRAY[1..gBA_cMaxShdObj] OF ST_BA_ShdObj;
    
```

stTiAct1: Eingabe der aktuellen Uhrzeit - hier die Ortszeit, da mithilfe dieser Zeit die verschattenden Monate berücksichtigt werden. Bei Verwendung der UTC-Zeit (bzw. GMT) kann, je nach Lage auf der Erde, der Monat mitten am Tag "umspringen".

lrFcdOrtn: Fassadenausrichtung, siehe Abbildung oben

lrAzm: Sonnenrichtung zum Betrachtungszeitpunkt [°]

lrElv: Sonnenhöhe zum Betrachtungszeitpunkt [°]

bSouth: FALSE: Berechnungen beziehen sich auf Verhältnisse der Nordhalbkugel - TRUE: auf Südhalbkugel

usiGrpId: Betrachtete Fenstergruppe. Die Gruppe 0 ist hierbei für nicht verwendete Fensterelemente reserviert, siehe [FB_BA_FcdElemEntry](#) [► 261]. Ein 0-Eintrag würde zu einer Fehlerausgabe führen (bErr=TRUE). Der Baustein wird dann nicht weiter abgearbeitet und *bGrpNotShdd* auf FALSE gesetzt.

arrShdObj: [Liste der Verschattungsobjekte](#) [► 46].

VAR_OUTPUT

```
bGrpNotShdd : BOOL;
bFcdSunlit  : BOOL;
bErr       : BOOL;
udiErrId   : UDINT;
udiErrArg  : UDINT;
```

bGrpNotShdd : Steht auf TRUE, solange die Fenstergruppe als nicht verschattet errechnet wird.

bFcdSunlit: Dieser Ausgang ist dann auf TRUE gesetzt, wenn die Sonne auf die Fassade scheint. Siehe Beschreibung oben.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
arrFcdElem : ARRAY[1..gBA_cMaxColumnFcd, 1..gBA_cMaxRowFcd] OF ST_BA_FcdElem;
```

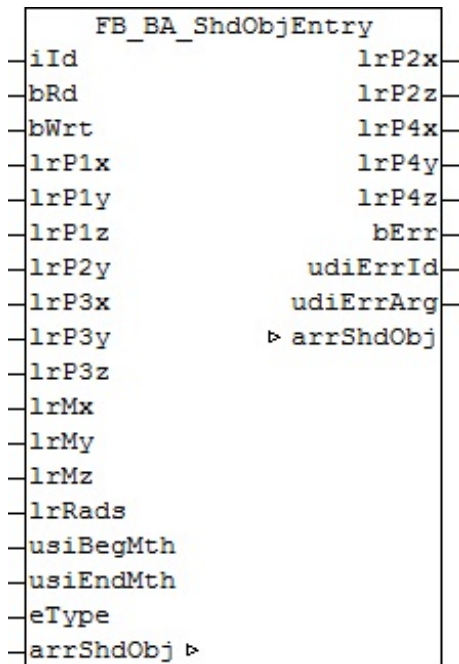
arrFcdElem: [Liste der Fassadenelemente](#) [► 46].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.111 FB_BA_Sh ObjEntry

Dieser Baustein dient zur Verwaltung aller Verschattungselemente einer Fassade, welche global in einer [Liste von Verschattungselementen](#) [► 46] hinterlegt ist. Er soll die Eingabe der Elementinformationen - auch im Hinblick auf die Nutzung der Target-Visualisierung - erleichtern. Eine schematische Darstellung der Objekte mit Beschreibung der Koordinaten ist unter [Verschattungskorrektur: Grundlagen und Definitionen](#) [► 21] gegeben.



Funktionsbeschreibung

Die Deklaration der Verschattungselemente erfolgt in der globalen Variable:

```
VAR_GLOBAL
    arrShadingObject : ARRAY[1..iShadingObjects] OF ST_BA_ShObj;
END_VAR
```

Jedes einzelne Element *arrShadingObject[1]* bis *arrShadingObject[gBA_cMaxShdObj]* trägt die Informationen für jeweils ein Verschattungselement (*ST_BA_ShObj* [▸ 335]). Dies sind neben dem gewählten Verschattungstyp (Viereck oder Kugel) die jeweils dazugehörigen Koordinaten. Im Falle eines Vierecks sind es die Eckpunkte (*lrP1x*, *lrP1y*, *lrP1z*), (*lrP2x*, *lrP2y*, *lrP2z*), (*lrP3x*, *lrP3y*, *lrP3z*) und (*lrP4x*, *lrP4y*, *lrP4z*) und im Falle einer Kugel der Mittelpunkt (*lrMx*, *lrMy*, *lrMz*) und der Radius *lrRads*. Zusätzlich kann über die Eingänge *usiBegMth* und *usiEndMth* die Phase der Verschattung definiert werden, was bei Objekten, wie z. B. Bäume, die im Winter kein Laub tragen, wichtig ist.

Der Baustein greift dabei über die IN-OUT-Variable *arrShadingObject* direkt auf das Feld dieser Informationen zu.

Anmerkung: Die Tatsache, dass die Viereckskoordinaten *lrP2x*, *lrP2z*, *lrP4x*, *lrP4y*, und *lrP4z* Ausgangswerte sind, ergibt sich daraus, dass sie aus den Eingabeparametern gebildet werden:

$$lrP2x = lrP1x; lrP2z = lrP1z; lrP4x = lrP3x; lrP4y = lrP1y; lrP4z = lrP3z;$$

Das schränkt die Eingabe eines Vierecks soweit ein, dass die seitlichen Kanten senkrecht auf dem Boden stehen ($lrP2x = lrP1x$ und $lrP4x = lrP3x$), dass das Viereck keine Neigung hat ($lrP2z = lrP1z$ und $lrP4z = lrP3z$) und nur nach "oben" also in positive y-Richtung eine unterschiedliche Höhe haben kann ($lrP4y = lrP1y$).

Die Verwendung des Bausteins erfolgt in drei Schritten:

- Auslesen
- Ändern
- Schreiben

Auslesen

Mit dem Eintrag an *ild* wird das entsprechende Element aus der Liste, *arrShadingObject[ild]*, ausgewählt. Eine steigende Flanke an *bRd* liest die Daten aus. Diese Werte werden den Ein- und Ausgangsvariablen des Bausteins zugewiesen. Es handelt sich hierbei um die Eingangswerte *lrP1x*, *lrP1y*, *lrP1z*, *lrP2y*, *lrP3x*, *lrP3y*,

lrP3z, *lrMx*, *lrMy*, *lrMz*, *rRadius* und dem Objekt-Enumerator *eType* und um die Ausgangswerte *lrP2x*, *lrP2z*, *lrP4x*, *lrP4y*, und *lrP4z*. Wichtig ist hierbei, dass im Schritt des Auslesens die Eingangswerte nicht überschrieben werden. So lassen sich alle Werte zunächst in einer Visualisierung anzeigen.

Ändern

In einem nächsten Programmschritt können dann die aufgeführten Eingabewerte verändert werden. Sollte am Eingang *eType* [► 330] über den Wert "*eObjectTypeTetragon*" die Verwendung eines Vierecks vorgewählt sein, ergeben sich die Ausgangswerte *lrP2x*, *lrP2z*, *lrP4x*, *lrP4y*, und *lrP4z* aus den eingegebenen Koordinaten des Vierecks, siehe oben.

Die eingegebenen Werte werden ständig auf Plausibilität überprüft. Der Ausgang *bErr* zeigt an, ob die Werte gültig sind (*bErr*=FALSE). Sollte dies nicht der Fall sein, wird am Ausgang *udiErrId/udiErrArg* ein entsprechender Fehlercode [► 340] ausgegeben.

Wird ein Viereck definiert, müssen lediglich die Eingänge *lrP1x*, *lrP1y*, *lrP1z*, *lrP2y*, *lrP3x*, *lrP3y* und *lrP3z* beschrieben werden, die Eingänge *lrMx*, *lrMy*, *lrMz* und *lrRads* brauchen nicht verknüpft zu werden. Im Falle einer Kugeldefinition müssen nur *lrMx*, *lrMy*, *lrMz* und *lrRads* beschrieben werden und die Viereckkoordinaten können unverknüpft bleiben

Schreiben

Mit einer positiven Flanke an *bWrt* werden die parametrisierten Daten in das Listenelement mit dem Index *ild* geschrieben, unabhängig davon, ob sie gültige Werte darstellen oder nicht. Daher ist innerhalb der Elementstruktur *ST_BA_ShdObj* [► 335] ein Plausibilitätsbit *bVld* vorhanden, das genau diese Information an den Baustein *FB_BA_ShdCorr* [► 281] weiterreicht und dort Fehlberechnungen vorbeugt.

Diese Vorgehensweise soll nur als Vorschlag angesehen werden. Es ist natürlich auch möglich, den Baustein ganz normal in einem Schritt zu parametrieren und die eingetragenen Werte mit einer steigenden Flanke an *bWrt* in das entsprechende Listenelement zu schreiben.

Ein-/Ausgänge

VAR_INPUT

```
iId      : INT;
bRd      : BOOL;
bWrt     : BOOL;
lrP1x    : LREAL;
lrP1y    : LREAL;
lrP1z    : LREAL;
lrP2y    : LREAL;
lrP3x    : LREAL;
lrP3y    : LREAL;
lrP3z    : LREAL;
lrMx     : LREAL;
lrMy     : LREAL;
lrMz     : LREAL;
lrRads   : LREAL;
usiBegMth : USINT;
usiEndMth : USINT;
eType    : E_BA_ShdObjType;
```

ild: Index des gewählten Elementes. Dies bezieht sich auf die Auswahl eines Feldelementes des unter der IN-OUT-Variablen *arrShdObj* angelegten Arrays. **ild darf nicht Null sein!** Das ergibt sich aus der Felddefinition, welche mit 1 beginnt, siehe oben.

bRd: Mit einer positiven Flanke an diesem Eingang werden die Informationen des gewählten Elementes, *arrShdObj[ild]* in den Baustein gelesen und den Eingangsvariablen *lrP1x* bis *eType* sowie den Ausgangsvariablen *lrP2x* bis *lrP4z* zugewiesen. Sind zu diesem Zeitpunkt schon Daten an den Eingängen *lrP1x* bis *eType* angelegt, so werden die zuvor ausgelesenen Daten sofort mit diesen überschrieben.

bWrt: Eine positive Flanke schreibt die an den Eingängen *lrP1x* bis *eType* angelegten sowie die ermittelten und den Ausgängen *lrP2x* bis *lrP4z* zugewiesenen Werte in das gewählte Feldelement *arrShdObj[ild]*.

lrP1x: X-Koordinate des Punktes 1 des Verschattungselementes (Viereck) [m]

lrP1y: Y-Koordinate des Punktes 1 des Verschattungselementes (Viereck) [m]

lrP1z: Z-Koordinate des Punktes 1 des Verschattungselementes (Viereck) [m]

lrP2y: Y-Koordinate des Punktes 2 des Verschattungselementes (Viereck) [m]

lrP3x: X-Koordinate des Punktes 3 des Verschattungselementes (Viereck) [m]

lrP3y: Y-Koordinate des Punktes 3 des Verschattungselementes (Viereck) [m]

lrP3z: Z-Koordinate des Punktes 3 des Verschattungselementes (Viereck) [m]

lrMx: X-Koordinate des Mittelpunktes des Verschattungselementes (Kugel) [m]

lrMy: Y-Koordinate des Mittelpunktes des Verschattungselementes (Kugel) [m]

lrMz: Z-Koordinate des Mittelpunktes des Verschattungselementes (Kugel) [m]

lrRads: Radius des Verschattungselementes (Kugel) [m]

usiBegMth: Anfang der Verschattungsperiode (Monatszahl)

usiEndMth: Ende der Verschattungsperiode (Monatszahl)

eType: Gewählter Elementtyp: Viereck oder Kugel. Siehe [E_BA_ShObjType](#) [► 330].

Bemerkung zur Verschattungsperiode:

Die Monateinträge dürfen nicht 0 und größer 12 sein, andernfalls sind alle Kombinationen möglich.

Beispiele:

Beginn=1, Ende=1: Verschattung im Januar.

Beginn=1, Ende=5: Verschattung von Anfang Januar bis Ende Mai.

Beginn=11, Ende=5: Verschattung von Anfang November bis Ende Mai (des folgenden Jahres).

VAR_OUTPUT

```
lrP2x      : LREAL;
lrP2z      : LREAL;
lrP4x      : LREAL;
lrP4y      : LREAL;
lrP4z      : LREAL;
bErr       : BOOL;
udiErrId   : UDINT;
```

lrP2x: Ermittelte X-Koordinate des Punktes 2 des Verschattungselementes (Viereck) [m]. Siehe "[Anmerkung](#) [► 285]" oben.

lrP2z: Ermittelte Z-Koordinate des Punktes 2 des Verschattungselementes (Viereck) [m]. Siehe "[Anmerkung](#) [► 285]" oben.

lrP4x: Ermittelte X-Koordinate des Punktes 4 des Verschattungselementes (Viereck) [m]. Siehe "[Anmerkung](#) [► 285]" oben.

lrP4y: Ermittelte Y-Koordinate des Punktes 4 des Verschattungselementes (Viereck) [m]. Siehe "[Anmerkung](#) [► 285]" oben.

lrP4z: Ermittelte Z-Koordinate des Punktes 4 des Verschattungselementes (Viereck) [m]. Siehe "[Anmerkung](#) [► 285]" oben.

bErr: Ergebnis der Plausibilitätskontrolle für die eingegebenen Werte. In Bezug auf ein Viereck ist verlangt, dass der Innenwinkel 360° beträgt und die Punkte in einer Ebene liegen und *vor* der betrachteten Fassade liegen. Bei einer Kugel muss der Mittelpunkt ebenfalls vor der Fassade liegen und der Radius muss größer Null sein.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

VAR_IN_OUT

```
arrShdObj : ARRAY[1..gBA_cMaxShdObj] OF ST_BA_ShObj;
```

arrShdObj: Liste der Verschattungsobjekte [► 46].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

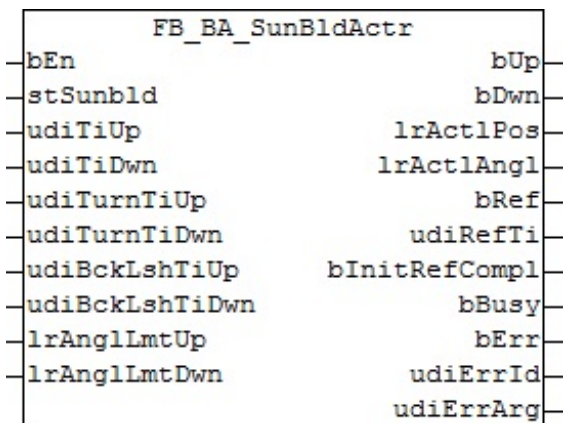
8.2.112 FB_BA_SunBldActr

Dieser Baustein dient zur Positionierung einer Lamellen-Jalousie über zwei Ausgänge: hoch- und herunterfahren. Über das Positioniertelegramm [stSunBld \[▶ 337\]](#) kann die Jalousie auf eine beliebige (Höhen-) Position und einen Lamellenwinkel gefahren werden. Darüber hinaus beinhaltet das Positioniertelegramm [stSunBld \[▶ 337\]](#) auch Handbefehle, mit denen die Jalousie individuell auf bestimmte Stellungen bewegt werden kann. Diese Handbefehle werden von dem Baustein [FB_BA_SunBldSwi \[▶ 301\]](#) angesteuert.

Der Baustein besitzt intern eine feste Umschaltverriegelung (Ausgang *bUp* zu Ausgang *bDwn*) von **500ms**.



Dieser Baustein muss in jedem SPS-Zyklus aufgerufen werden, da die SPS-Zykluszeit in die Berechnung der Positionen mit einfließt.



Funktionsbeschreibung

Struktur des Jalousie-Positioniertelegramms [stSunBld \[▶ 337\]](#)

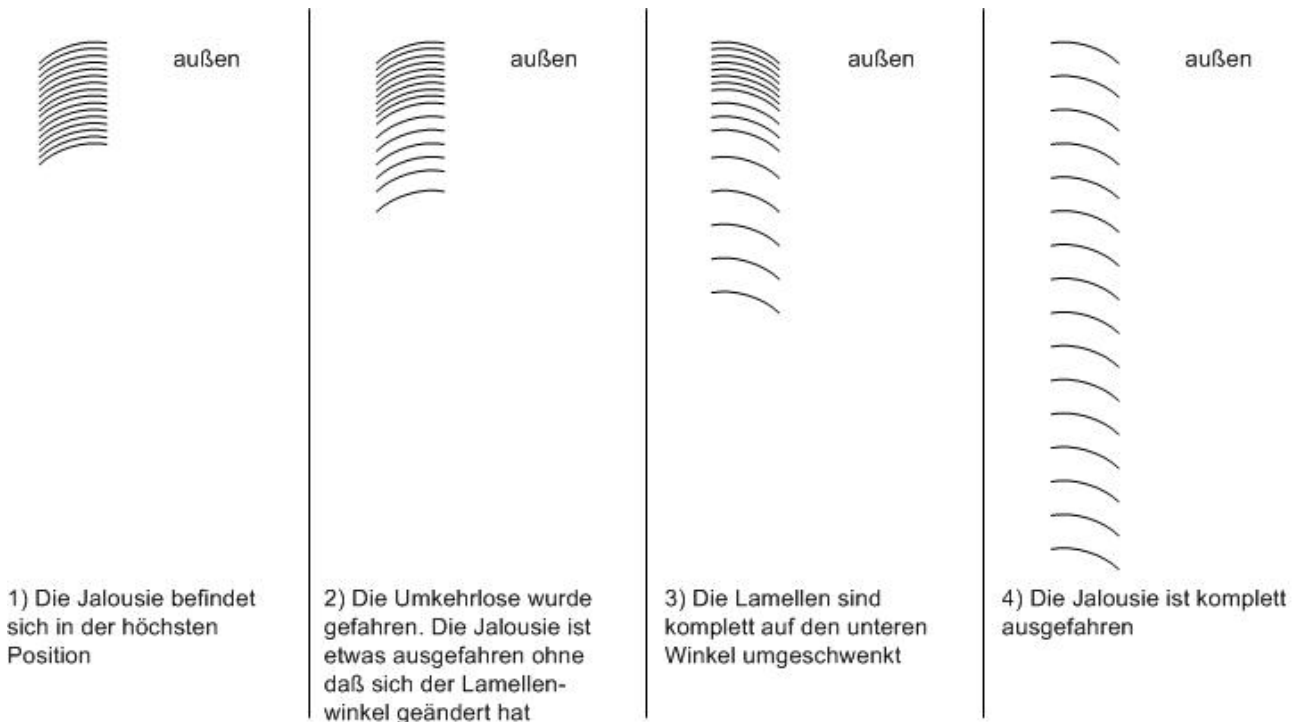
```

TYPE ST_BA_SunBld:
STRUCT
    lrPos      : LREAL;
    lrAngl    : LREAL;
    bManUp    : BOOL;
    bManDwn   : BOOL;
    bManMod   : BOOL;
    bActv     : BOOL;
END_STRUCT
END_TYPE

```

Die aktuelle Höhenposition und der Lamellenwinkel werden dabei nicht durch einen zusätzlichen Encoder eingelesen, sondern intern durch die Laufzeit der Jalousie ermittelt. Der Berechnung liegt folgendes Fahrprofil zugrunde (von der höchsten und niedrigsten Position der Jalousie aus betrachtet):

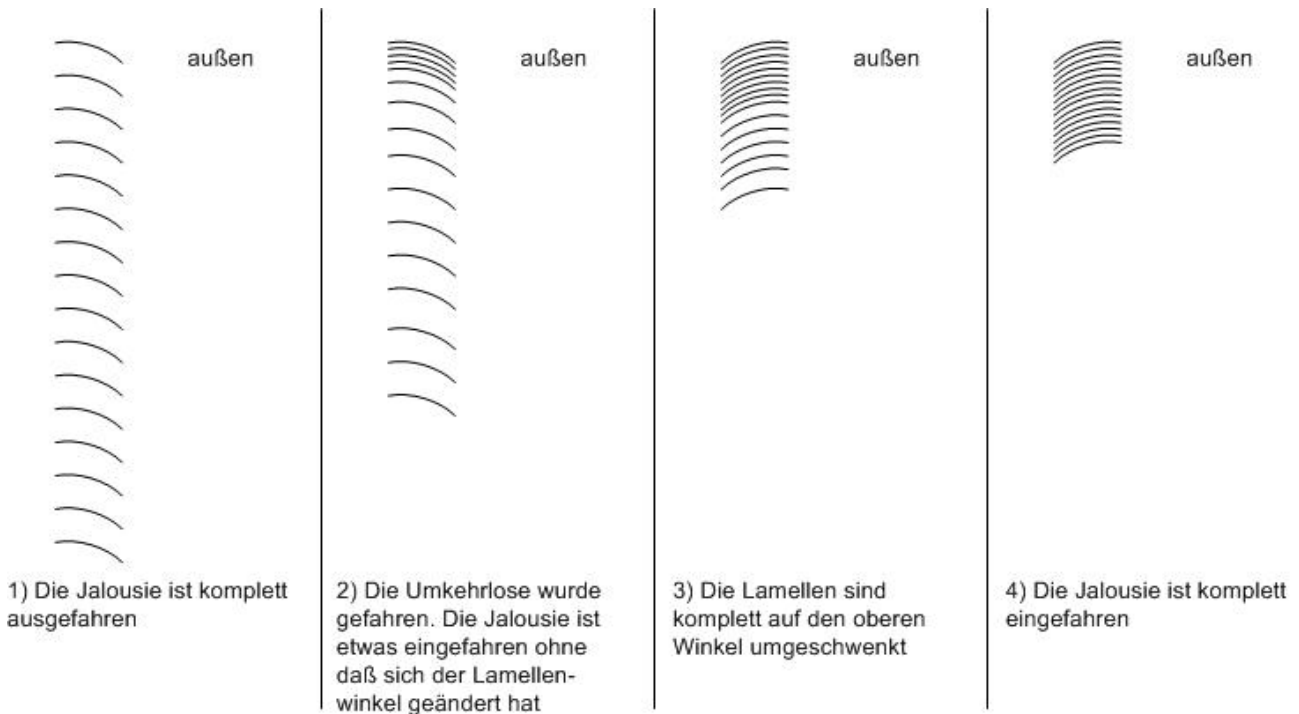
Fahrprofil abwärts:



Nähere Erläuterung zu den Begriffen "Umkehrlose" und "Umschwenken", hier in der Abwärtsbewegung:

Die Jalousie beschreibt ihre Abwärtsbewegung normalerweise mit dem Lamellentiefpunkt nach außen gerichtet, wie in Bild 3.

Befindet sich die Jalousie in einer Ausgangsstellung mit dem Tiefpunkt nach innen (d.h. nach Abschluss einer Aufwärtsbewegung), so vergeht bei einer erneuten Abwärtsfahrt eine gewisse Zeit bis die Lamellen vom "Tiefpunkt innen" bis zu "Tiefpunkt außen" zu schwenken beginnen. Während dieser Zeit ändert sich der Lamellenwinkel nicht, die Jalousie fährt nur herunter (Bild1 und Bild2). Diese Zeit ist ein wichtiger Parameter zur Bewegungsberechnung und wird am Baustein unter *udiBckLshTiDwn* [ms] eingetragen. Da man an einem x-beliebigen Punkt nach einer beliebig langen Jalousiebewegung nicht weiß, ob schon etwas von der Umkehrlosen ausgefahren wurde, misst man die Umkehrlose der Abwärtsbewegung bzw. deren Ausfahrzeit am Sichersten, wenn die Jalousie zunächst ganz nach oben gefahren wurde. Ein weiterer wichtiger Parameter ist die Zeitspanne des darauf folgenden Umschwenkens der Lamellen von "Tiefpunkt innen" bis zu "Tiefpunkt außen". Diese Zeit ist als *udiTurnTiDwn* [ms] am Baustein einzutragen.

Fahrprofil aufwärts:**Nähere Erläuterung zu den Begriffen "Umkehrlose" und "Umschwenken", hier in der Aufwärtsbewegung:**

Dieser Sachverhalt ist analog zur oben beschriebenen Abwärtsbewegung: Die Jalousie beschreibt ihre Aufwärtsbewegung normalerweise mit dem Lamellentiefpunkt nach innen gerichtet, wie in Bild 3. Befindet sich die Jalousie in einer Ausgangsstellung mit dem Tiefpunkt nach außen (d.h. nach Abschluss einer Abwärtsbewegung), so vergeht bei einer erneuten Aufwärtsfahrt eine gewisse Zeit bis die Lamellen vom "Tiefpunkt außen" bis zu "Tiefpunkt innen" zu schwenken beginnen. Während dieser Zeit ändert sich der Lamellenwinkel nicht, die Jalousie fährt nur herauf (Bild1 und Bild2). Auch diese Zeit ist ein wichtiger Parameter zur Bewegungsberechnung und wird am Baustein unter *udiBckLshTiUp* [ms] eingetragen. Da man an einem x-beliebigen Punkt nach einer beliebig langen Jalousiebewegung nicht weiß, ob schon etwas von der Umkehrlosen ausgefahren wurde, misst man die Umkehrlose der Aufwärtsbewegung bzw. deren Ausfahrzeit am Sichersten, wenn die Jalousie zunächst ganz nach unten gefahren wurde. Ein weiterer wichtiger Parameter ist die Zeitspanne des darauf folgenden Umschwenkens der Lamellen von "Tiefpunkt außen" bis zu "Tiefpunkt innen". Diese Zeit ist als *udiTurnTiUp* [ms] am Baustein einzutragen.

Parametrierung

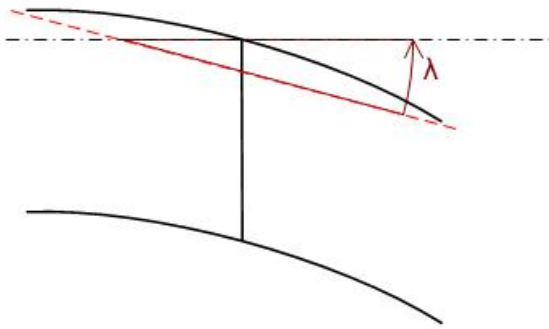
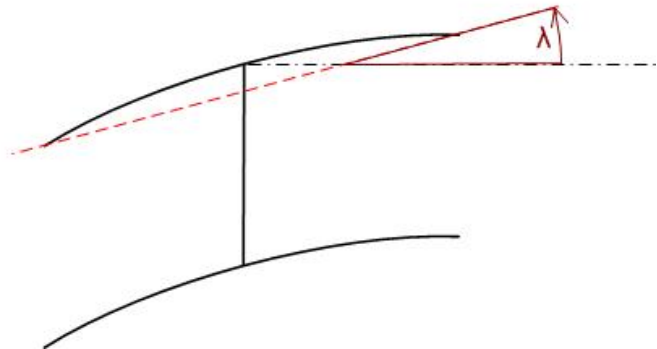
Zur Berechnung der (Höhen-)Position und des Lamellenwinkels sind nun jeweils für die Auf- und Abwärtsbewegung folgende Zeiten zu ermitteln:

- die Verfahrdauer der Umkehrlose (*udiBckLshTiUp* / *udiBckLshTiDwn* [ms])
- die Verfahrdauer des Umschwenkens (*udiTurnTiUp* / *udiTurnTiDwn* [ms])
- die Gesamt-Verfahrdauer (*udiTiUp* / *udiTiDwn* [ms])

Des Weiteren sind zur Berechnung erforderlich:

- der höchste Lamellenwinkel nach einem Umschwenken nach oben (*IrAngLmtUp* [°])
- der niedrigste Lamellenwinkel nach einem Umschwenken nach unten (*IrAngLmtDwn* [°])

Der Lamellenwinkel λ ist dabei durch eine gedachte Gerade durch die Endpunkte der Lamelle zur Horizontalen definiert.

Lamellenwinkel $\lambda < 0$ Lamellenwinkel $\lambda > 0$ 

Funktionsweise

Der Baustein steuert die Jalousie grundsätzlich über die Informationen, aus dem Positioniertelegramm [stSunBld](#) [► 337]. Ist der Automatikmodus aktiv ($bManMod=FALSE$), so wird immer die aktuelle Position und Lamellenwinkel angefahren, wobei Änderungen sofort berücksichtigt werden. Die Positionierung auf die Höhe hat dabei Vorrang: Es wird zunächst die eingegebene Höhe und danach der Lamellenwinkel angefahren. Aus Gründen der Einfachheit bleibt dabei der Positionsfehler durch das Winkel-Verfahren unberücksichtigt. Im Handbetrieb ($bManMod=TRUE$) steuern die Befehle $bManUp$ und $bManDwn$ die Jalousie. Bei einem Wechsel vom Hand- in den Automatikmodus wird in jedem Fall ein Automatik-Fahrbehl ausgelöst.

Referenzieren

Ein sicheres Referenzieren ist gegeben, wenn die Jalousie länger als ihre komplette Hochlaufzeit nach oben hin angesteuert wird. Die Position ist dann auf jeden Fall "0" und der Lamellenwinkel auf seinem Maximum. Da eine Jalousiepositionierung ohne Encoder naturgemäß immer fehlerbehaftet ist, ist es wichtig möglichst oft automatisch zu referenzieren: jedes mal, wenn die Position "0" angefahren werden soll (der Winkel spielt dabei keine Rolle) fährt die Jalousie zunächst ganz normal mit kontinuierlicher Positionsberechnung nach oben. Erreicht sie den errechneten Positionswert 0%, so wird der Ausgang bUp gehalten und zwar noch einmal für die komplette Hochlaufzeit + 5s.

Aus Gründen der Flexibilität gibt es nun zwei Möglichkeiten, den Referenziervorgang zu unterbrechen: Bis zum Erreichen der errechneten 0%-Position wird eine Positionsänderung immer noch angenommen und ausgeführt, nach Erreichen dieser 0%-Position kann die Jalousie noch mit dem Handbefehl "herunterfahren" anders bewegt werden. Diese beiden sinnvollen Einschränkungen machen es nötig, dass der Nutzer selbst dafür Sorge trägt, die Jalousie so oft wie möglich sicher referenzieren zu lassen.

Beim Neustart der Anlage referenziert der Baustein in jedem Fall. Die abgeschlossene Initial-Referenzierung wird durch ein TRUE-Signal an Ausgang $bInitRefCmpl$ angezeigt. Die Initial-Referenzierung kann durch einen Handbefehl "herunterfahren" **nicht** vorzeitig abgeschlossen werden.

Zielgenauigkeit

Zielgenauigkeit

Da der Baustein die Position der Jalousie alleine über Laufzeiten ermittelt, spielt die Zykluszeit der SPS-Task eine entscheidende Rolle bei der Positioniergenauigkeit. Besteht beispielsweise eine Umschwenkzeit von 1s für einen Lamellen-Winkelbereich von -70° bis 10° , so liegt bei einer Zykluszeit von 50ms die Genauigkeit bei $\pm 4^\circ$.

Ein-/Ausgänge

VAR_INPUT

```

bEn          : BOOL;
stSunbld     : ST_BA_SunBld;
udiTiUp      : UDINT;
udiTiDwn     : UDINT;
udiTurnTiUp  : UDINT;
udiTurnTiDwn : UDINT;
udiBckLshTiUp : UDINT;

```

```

udiBckLshTiDwn : UDINT;
lrAnglLmtUp    : LREAL;
lrAnglLmtDwn   : LREAL;

```

bEn: Freigabeeingang für den Baustein. Solange dieser Eingang auf TRUE steht, nimmt der Aktorbaustein Befehle entgegen und arbeitet, wie oben beschrieben. Ein FALSE-Signal an diesen Eingang setzt die Steuerausgänge *bUp* und *bDwn* zurück und der Funktionsbaustein verharrt in einem Ruhezustand.

stSunbld: Positioniertelegramm, siehe [ST_BA_SunBld](#) [► 337]

udiTiUp: Komplette Hochfahrzeit [ms]

udiTiDwn: Komplette Zeit zum Herunterfahren [ms]

udiTurnTiUp: Zeit zum Umschwenken der Lamellen in obere Richtung [ms]

udiTurnTiDwn: Zeit zum Umschwenken der Lamellen in untere Richtung [ms]

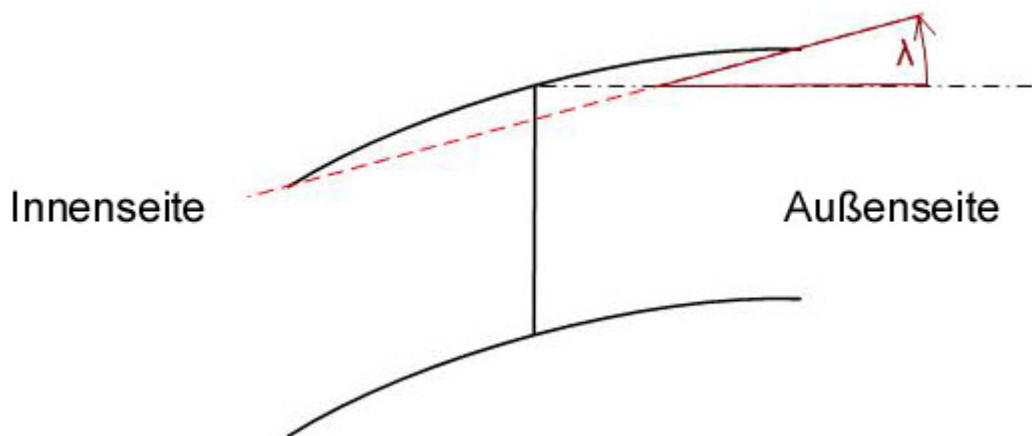
udiBckLshTiUp: Zeit zum Ausfahren der Umkehrlose in obere Richtung [ms]

udiBckLshTiDwn: Zeit zum Ausfahren der Umkehrlose in untere Richtung [ms]

lrAnglLmtUp: Höchste Stellung der Lamellen [°]

Diese Stellung ist erreicht, wenn die Jalousie ganz hochgefahren ist.

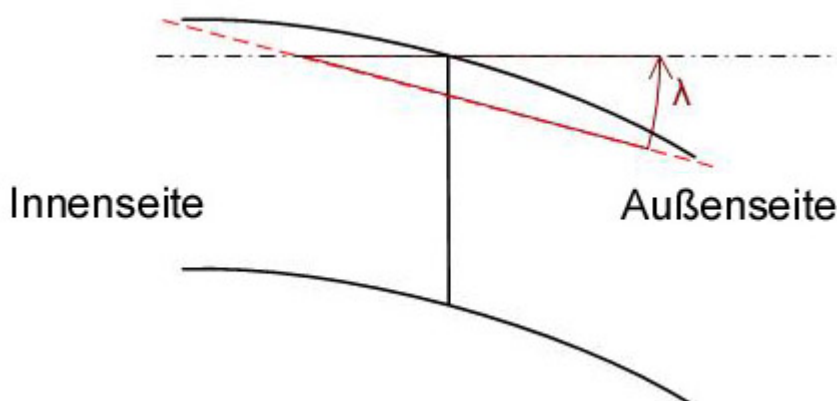
Der Lamellenwinkel λ , so wie er einleitend definiert ist, ist dann typischerweise größer Null.



lrAnglLmtDwn: Niedrigste Stellung der Lamellen [°].

Diese Stellung ist erreicht, wenn die Jalousie ganz heruntergefahren ist.

Der Lamellenwinkel λ , so wie er einleitend definiert ist, ist dann typischerweise kleiner Null.



VAR_OUTPUT

```
bUp      : BOOL;
bDwn     : BOOL;
lrActlPos : LREAL;
lrActlAngl : LREAL;
bRef     : BOOL;
bInitRefCompl : BOOL;
bRef     : BOOL;
bBusy    : BOOL;
bErr     : BOOL;
udiErrId : UDINT;
udiErrArg : UDINT;
```

bUp: Steuerausgang Jalousie hoch

bDwn: Steuerausgang Jalousie herunter

lrActlPos: Aktuelle Position in Prozent

lrActlAngl: Aktueller Lamellenwinkel [°]

bRef: Die Jalousie befindet sich in der Referenzierung, d.h. für die die komplette Hochlaufzeit + 5s wird der Ausgang *bUp* gesetzt. Nur ein Handbefehl "herunter" kann die Jalousie in Gegenrichtung bewegen und diesen Modus beenden.

udiRefTi: Referenzier-Countdown-Anzeige [s]

blnitRefCompl: Initial-Referenziervorgang abgeschlossen

bBusy: Ein Positionier- oder Referenziervorgang findet statt.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

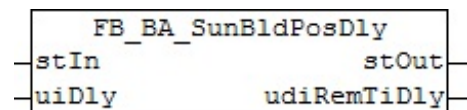
udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [▶ 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.113 FB_BA_SunBldPosDly

Dieser Baustein verzögert Positionsänderungen von Automatik-Befehlen.



Funktionsbeschreibung

Wenn durch ein Ereignis, zum Beispiel Wetterschutz, zu viele Jalousieantriebe gleichzeitig gestartet werden, kann es sein, dass die Summe der hohen Motor-Anzugsströme Sicherungen auslösen. Es ist daher wünschenswert, die Jalousieantriebe kurz hintereinander zu starten um den Gesamtstrom zu minimieren.

Dieser Baustein gibt Automatikbefehle vom Eingangstelegramm *stIn* [▶ 337] verzögert an das Ausgangstelegramm *stOut* [▶ 337] weiter. Er beachtet dazu drei Fälle:

1. Die Jalousieposition *lrPos* hat sich im Automatikmodus verändert (*bManMode* = FALSE im Telegramm *stIn*).
2. Der Lamellenwinkel *lrAngl* hat sich im Automatikmodus verändert (*bManMode* = FALSE im Telegramm *stIn*).

- Der Hand-Modus wurde gerade verlassen, d.h. der Automatikmodus gerade aktiv (fallende Flanke *bManMode* im Telegramm *stIn*).

Das Ausgangstelegramm *stOut*, ist immer eine direkte Kopie des Eingangstelegramms *stIn*. In diesen drei Fällen jedoch werden im Ausgangstelegramm *stOut* für die Zeit von *uiDly* [ms]:

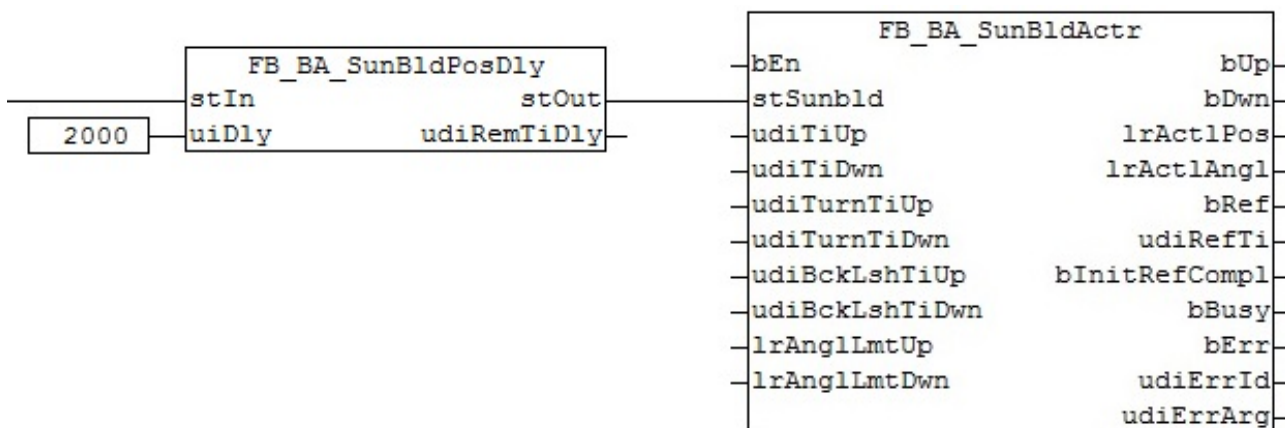
- *bManMode* = TRUE
- *bManUp* = FALSE
- *bManDwn* = FALSE

gesetzt. Dadurch wird die über den Baustein *FB_BA_SunBldActr* [► 288] angesteuerte Jalousie für den Zeitraum der Verzögerung auf ihrer Position gehalten. Jede weitere Änderung nach den oben genannten Kriterien innerhalb der Verzögerungszeit startet den Zeitgeber neu.

Ein Wechsel auf Hand im Eingangstelegramm (*bManMode* = TRUE) jedoch löscht den Warte-Timer unmittelbar. Das (Hand-)Telegramm wird unverzögert durchgereicht. So werden **nur** Automatiktelegramme verzögert.

Anwendung

Vorzugsweise direkt vor dem Jalousie-Aktorbaustein:



Ein-/Ausgänge

VAR_INPUT

```
stIn      : ST_BA_Sunblind;
uiDly     : UINT;
```

stIn: Eingangs-Positioniertelegramm, siehe [ST_BA_SunBld \[► 337\]](#)

uiDly: Verzögerungszeit des Aktiv-Bits im Positioniertelegramm [ms]

VAR_OUTPUT

```
stOut     : ST_BA_Sunblind;
```

stOut: Ausgangs-Positioniertelegramm, siehe [ST_BA_SunBld \[► 337\]](#)

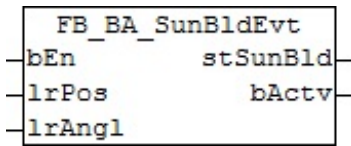
udiRemTiDly: Anzeigeausgang abgelaufene Verzögerungszeit [s]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.114 FB_BA_SunBldEvt

Dieser Baustein dient zur Positions- und Winkelvorgabe bei einem beliebigen Ereignis. Sie kann beispielsweise genutzt werden, um eine Parkposition anzufahren oder im Wartungsfall die Jalousie hochfahren zu lassen.



Funktionsbeschreibung

Die Funktion wird über den Eingang *bEn* aktiviert. Ist dies der Fall, so wird der Aktiv-Merker im Positioniertelegramm (*bActv* in *stSunBld*) am Ausgang *stSunBld* [► 337] gesetzt und die an den In-Out-Variablen eingetragenen Werte *lrPos* für die Jalousiehöhe [%] und *lrAngl* für den Lamellenwinkel [°] in diesem Telegramm weiter gereicht. Ist die Funktion durch Rücksetzen von *bEn* nicht mehr aktiv, so wird der Aktiv-Merker im Positioniertelegramm *stSunBld* [► 337] zurück und die Positionen für Höhe und Winkel auf "0" gesetzt. Nutzt man den Prioritätenbaustein, so kann dann eine Funktion niedrigerer Priorität die Steuerung übernehmen.

Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
lrPos    : LREAL;
lrAngl   : LREAL;
```

bEn: Ein TRUE-Signal an diesem Eingang aktiviert den Baustein und übergibt die eingetragenen Sollwerte im Positioniertelegramm *ST_BA_SunBld* [► 337] zusammen mit dem Aktivmerker. Ein FALSE-Signal setzt den Aktivmerker wieder zurück, sowie Position und Winkel auf Null.

lrPos: Höhenposition der Jalousie [%] im Falle einer Aktivierung

lrAngl: Lamellenwinkel der Jalousie [°] im Falle einer Aktivierung

VAR_OUTPUT

```
stSunBld : ST_BA_SunBld;
bActv    : BOOL;
```

bActv: Entspricht dem booleschen Wert *bActv* im Jalousie-Telegramm *ST_BA_SunBld* [► 337] und dient zur reinen Anzeige, ob der Baustein ein aktives Telegramm sendet.

stSunBld: Ausgabestruktur der Jalousiestellungen, siehe *ST_BA_SunBld* [► 337]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.115 FB_BA_SunBldPrioSwi4

Prioritätssteuerung für bis zu 4 Positioniertelegramme (*stSunBld_Prio1* ... *stSunBld_Prio4*) des Typs *ST_BA_SunBld* [► 337].

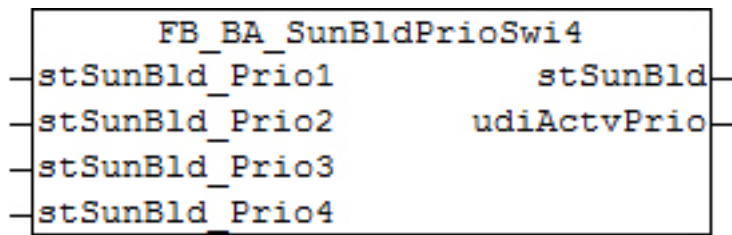


Abb. 2: FB_BA_SunBldPrioSwi4

Funktionsbeschreibung

Struktur des Jalousie-Positioniertelegramms [stSunBld](#) [[▶ 337](#)].

```

TYPE ST_BA_SunBld:
STRUCT
  lrPos      : LREAL;
  lrAngl     : LREAL;
  bManUp     : BOOL;
  bManDwn    : BOOL;
  bManMod    : BOOL;
  bActv      : BOOL;
END_STRUCT
END_TYPE
  
```

An diesem Baustein lassen sich bis zu 4 Positioniertelegramme verschiedener Steuerbausteine anlegen. Dabei hat das Telegramm an *stSunBld_Prio1* die höchste und das an *stSunBld_Prio4* die niedrigste Priorität. Das aktive Telegramm der höchsten Priorität wird am Ausgang *stSunBld* ausgegeben. "Aktiv" bedeutet, dass innerhalb der Struktur des Positioniertelegramms die Variable *bActv* gesetzt ist.

i Dieser Baustein ist so zu programmieren, dass immer eines der angelegten Telegramme aktiv ist. Sollte dies nicht der Fall sein, so wird am Ausgang ein Leertelegramm ausgegeben, d.h. *lrPos=0*, *lrAngl=0*, *bManUp=FALSE*, *bManDwn=FALSE*, *bManMod=FALSE*, *bActv=FALSE*. Da der Jalousiebaustein [FB_BA_SunBldActr](#) [[▶ 288](#)] bzw. der Rollladenbaustein [FB_BA_RolBldActr](#) [[▶ 279](#)] seinerseits nicht auf den Merker *bActv* achtet, würde dieses Telegramm als Fahrbefehl auf Position "0", also vollständig geöffnet, gewertet werden. Das Fehlen eines aktiven Telegramms stellt damit kein Sicherheitsrisiko für die Jalousie dar.

Ein-/Ausgänge

VAR_INPUT

```

stSunBld_Prio1 : ST_BA_SunBld;
stSunBld_Prio2 : ST_BA_SunBld;
stSunBld_Prio3 : ST_BA_SunBld;
stSunBld_Prio4 : ST_BA_SunBld;
  
```

stSunBld_Prio1..stSunBld_Prio4: Zur Auswahl stehende Positioniertelegramme. Dabei hat *stSunBld_Prio1* die höchste und *stSunBld_Prio4* die niedrigste Priorität.

VAR_OUTPUT

```

stSunBld      : ST_BA_SunBld;
udiActvPrio   : UDINT;
  
```

stSunBld: Resultierendes Positioniertelegramm

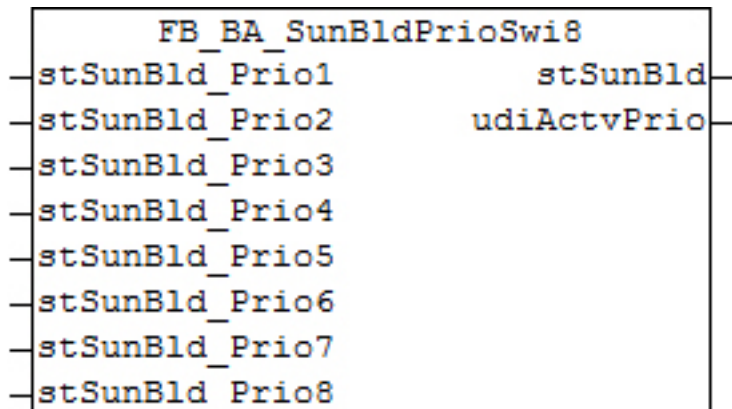
udiActvPrio: Aktives Positioniertelegramm. Ist keines aktiv, so wird "0" ausgegeben

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.116 FB_BA_SunBldPrioSwi8

Prioritätssteuerung für bis zu 8 Positioniertelegramme (*stSunBld_Prio1* ... *stSunBld_Prio8*) des Typs *ST_BA_SunBld* [► 337].



Funktionsbeschreibung

Struktur des Jalousie-Positioniertelegramms *stSunBld* [► 337].

```

TYPE ST_BA_SunBld:
STRUCT
    lrPos      : LREAL;
    lrAngl     : LREAL;
    bManUp     : BOOL;
    bManDwn   : BOOL;
    bManMod    : BOOL;
    bActv      : BOOL;
END_STRUCT
END_TYPE
  
```

An diesem Baustein lassen sich bis zu 8 Positioniertelegramme verschiedener Steuerbausteine anlegen. Dabei hat das Telegramm an *stSunBld_Prio1* die höchste und das an *stSunBld_Prio8* die niedrigste Priorität. Das aktive Telegramm der höchsten Priorität wird am Ausgang *stSunBld* ausgegeben. "Aktiv" bedeutet, dass innerhalb der Struktur des Positioniertelegramms die Variable *bActv* gesetzt ist.

i Dieser Baustein ist so zu programmieren, dass immer eines der angelegten Telegramme aktiv ist. Sollte dies nicht der Fall sein, so wird am Ausgang ein Leertelegramm ausgegeben, d.h. *lrPos*=0, *lrAngl*=0, *bManUp*=FALSE, *bManDwn*=FALSE, *bManMod*=FALSE, *bActv*=FALSE. Da der Jalousiebaustein *FB_BA_SunBldActr* [► 288] bzw. der Rollladenbaustein *FB_BA_RolBldActr* [► 279] seinerseits nicht auf den Merker *bActv* achtet, würde dieses Telegramm als Fahrbefehl auf Position "0", also vollständig geöffnet, gewertet werden. Das Fehlen eines aktiven Telegramms stellt damit kein Sicherheitsrisiko für die Jalousie dar.

Ein-/Ausgänge

VAR_INPUT

```

stSunBld_Prio1 : ST_BA_SunBld;
stSunBld_Prio2 : ST_BA_SunBld;
stSunBld_Prio3 : ST_BA_SunBld;
stSunBld_Prio4 : ST_BA_SunBld;
stSunBld_Prio5 : ST_BA_SunBld;
stSunBld_Prio6 : ST_BA_SunBld;
stSunBld_Prio7 : ST_BA_SunBld;
stSunBld_Prio8 : ST_BA_SunBld;
  
```

stSunBld_Prio1..stSunBld_Prio8: Zur Auswahl stehende Positioniertelegramme. Dabei hat *stSunBld_Prio1* die höchste und *stSunBld_Prio8* die niedrigste Priorität.

VAR_OUTPUT

```

stSunBld      : ST_BA_SunBld;
udiActvPrio   : UDINT;
  
```

stSunBld: Resultierendes Positioniertelegramm

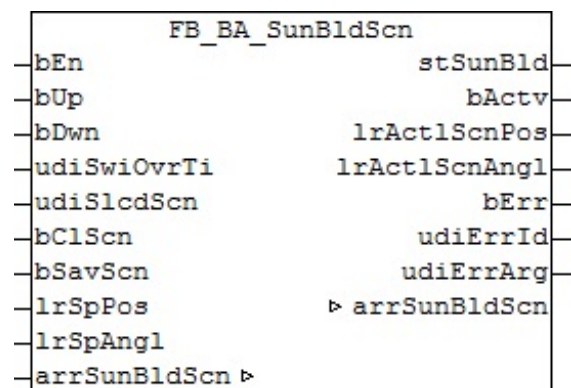
udiActvPrio: Aktives Positioniertelegramm. Ist keines aktiv, so wird "0" ausgegeben.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.117 FB_BA_SunBldScn

Dieser Baustein stellt eine Erweiterung der Handbedienung [FB_BA_SunBldSwi](#) [[▶ 301](#)] um eine Szenen-Speicher- und Aufruf-Funktionalität dar. Damit lässt sich die Jalousieansteuerung [FB_BA_SunBldActr](#) [[▶ 288](#)] bzw. die Rollladenansteuerung [FB_BA_RolBldActr](#) [[▶ 279](#)] sowohl im Handbedienmodus ansteuern, als auch zuvor gespeicherte Positionen (Szenen) direkt anfahren. Es können bis zu 21 Szenen gespeichert werden.



Funktionsbeschreibung

Struktur des Jalousie-Positioniertelegramms [stSunBld](#) [[▶ 337](#)]

```

TYPE ST_BA_SunBld:
STRUCT
    lrPos      : LREAL;
    lrAngl    : LREAL;
    bManUp    : BOOL;
    bManDwn   : BOOL;
    bManMod   : BOOL;
    bActv     : BOOL;
END_STRUCT
END_TYPE

```

Betrieb

Der Funktionsbaustein steuert über die Befehlseingänge *bUp* und *bDwn* den Jalousiebaustein [FB_BA_SunBldActr](#) [[▶ 288](#)] bzw. den Rollladenbaustein [FB_BA_RolBldActr](#) [[▶ 279](#)] im Handbetrieb an, wobei *bUp* Vorrang hat. Die Befehle werden an die jeweiligen Kommandos *bManUp* und *bManDwn* des Positioniertelegramms weiter gereicht. Ist ein Befehlseingang länger als die eingetragene Zeit *udiSwiOvrTi* [ms] aktiviert, so geht der entsprechende Steuerbefehl in Selbsthaltung. Ein erneutes Aktivieren eines Befehlseinganges löscht diese Selbsthaltung wieder.

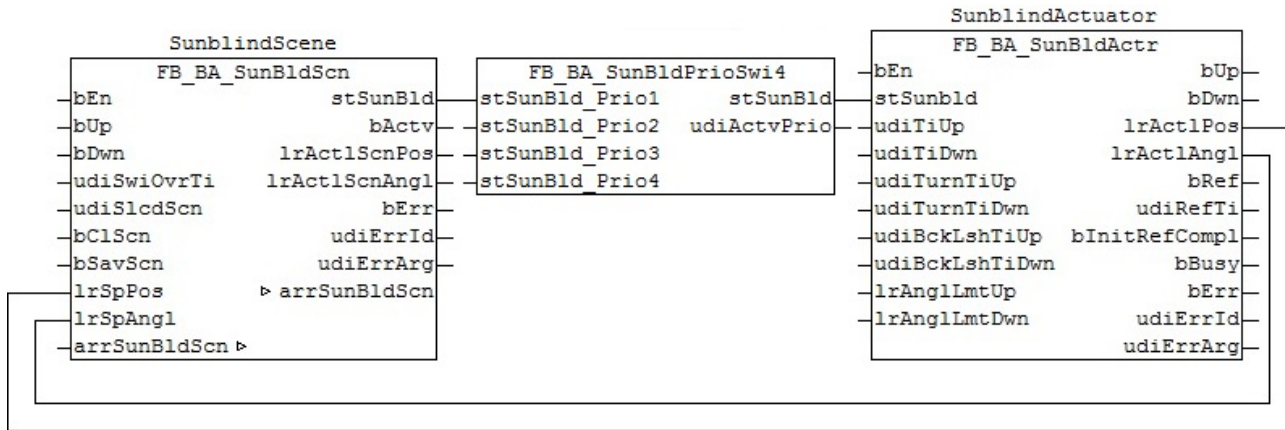
Eine steigende Flanke an *bSavScn* speichert die aktuelle Position und den Lamellenwinkel in die unter *udiSlcdScn* angewählte Szene. Dieser Vorgang ist jederzeit möglich, auch während einer aktiven Positionierung. Mit *bClScn* wird die angewählte Szene aufgerufen, das heißt, die gespeicherten Werte von Position und Winkel angefahren.

Mit Aktivierung des Bausteins über *bEn* wird im Positioniertelegramm das Bit *bActv* und *bManMod* sofort gesetzt. Damit meldet der Baustein an einem Prioritätsschalter ([FB_BA_SunBldPrioSwi4](#) [[▶ 295](#)] bzw. [FB_BA_SunBldPrioSwi8](#) [[▶ 297](#)]) seinen Vorrang gegenüber niedrigen Prioritäten an und durch das Setzen von *bManMod* weiß der Aktorbaustein, dass kein Automatikbefehl auf eine Position zu fahren ist.

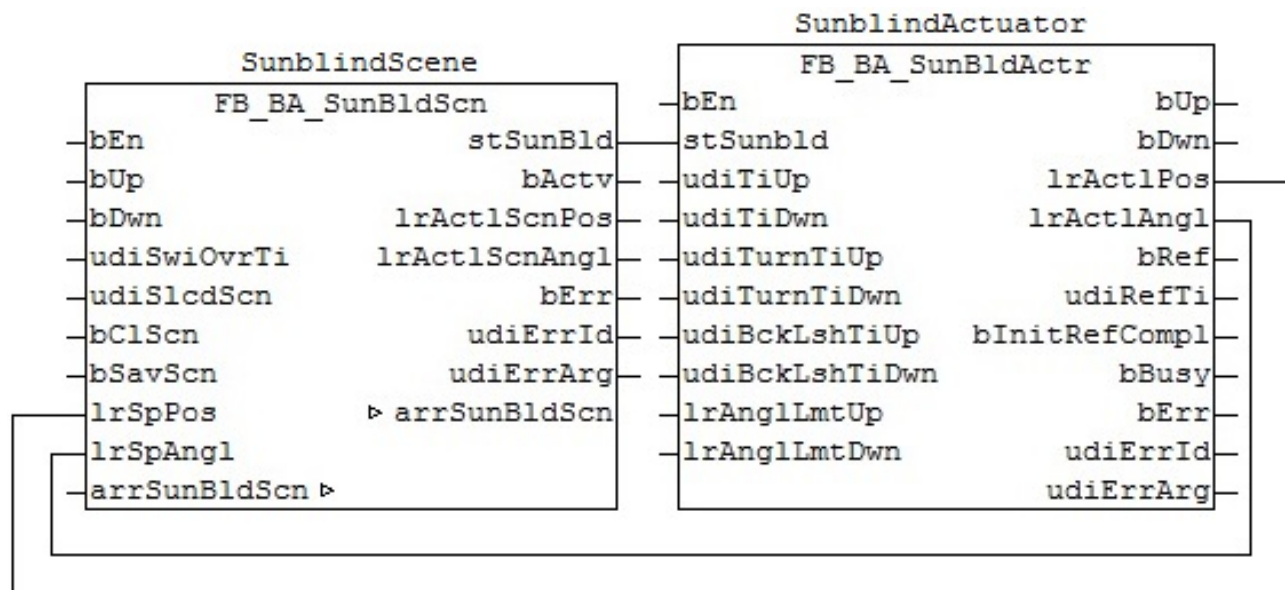
Verknüpfung an Jalousiebaustein

Der Szenenanwahlbaustein kann, wie der "normale" Handbedienbaustein [FB_BA_SunBldSwi](#) [▶ 301], entweder über eine voran gestellte Prioritätssteuerung [FB_BA_SunBldPrioSwi4](#) [▶ 295] bzw. [FB_BA_SunBldPrioSwi8](#) [▶ 297] oder aber direkt an den Jalousiebaustein angeschlossen werden. Die Verbindung erfolgt dabei über das Positioniertelegramm [stSunBld](#) [▶ 337]. Des weiteren benötigt der Szenenbaustein die aktuelle Positionen aus dem Jalousiebaustein der Referenz-Jalousie:

Verwendung einer Prioritätssteuerung:



Direkte Beschaltung:



Ein-/Ausgänge

VAR_INPUT

- bEn : BOOL;
- bUp : BOOL;
- bDwn : BOOL;
- udiSwiOvrTi : UDINT;
- udiSlcdScn : UDINT;
- bClScn : BOOL;
- bSavScn : BOOL;
- lrSpPos : LREAL;
- lrSpAngl : LREAL;

bEn: Steht dieser Eingang auf FALSE, so ist der Baustein ohne Funktion. Im Positioniertelegramm stSunBld [▶ 337] werden für Position und Winkel jeweils 0 ausgegeben - *bManualMode* und *bActv* stehen jeweils auf FALSE. Das bedeutet für eine Beschaltung mit Prioritätssteuerung, dass eine andere Funktionalität die Jalousieansteuerung übernimmt. Eine direkte Beschaltung hingegen lässt die Jalousie direkt auf Position 0, also ganz nach oben fahren, da der Aktorbaustein das Bit *bActv* selbst nicht auswertet.

bUp: Befehlseingang Jalousie hoch

bDwn: Befehlseingang Jalousie herunter

udiSwiOvrTi: Zeit [ms] bis bei dauerhaft aktiviertem Befehlseingang der entsprechende Handbefehl im Positioniertelegramm stSunBld [▶ 337] in Selbsthaltung geht.

udiSlcdScn: Angewählte Szene, welche entweder gespeichert (*bSaveScene*) oder aufgerufen (*bInvokeScene*) werden soll.

bClScn: Angewählte Szene aufrufen

bSavScn: Angewählte Szene speichern

lrSpPos: Sollposition [%], welche in der gewählten Szene gesichert werden soll. Ist mit der Ist-Position des Aktorbausteins FB_BA_SunBldActr [▶ 288] bzw. FB_BA_RolBldActr [▶ 279] der Referenz-Jalousie/Rolllade zu verknüpfen, um dadurch eine vorher manuell angefahrne Position speichern zu können.

lrSpAngl: dto. Lamellenwinkel [°]

VAR_IN_OUT

```
arrSunBldScn : ARRAY[0..cMaxSunBldScn] OF ST_BA_SunBldScn;
```

arrSunBldScn: Tabelle mit den Szeneneinträgen vom Typ ST_BA_SunBldScn [▶ 337]. Es können bis zu 21 Szenen gespeichert werden (0..20).

VAR_OUTPUT

```
stSunBld      : ST_BA_SunBld;
bActv         : BOOL;
lrActlScnPos  : LREAL;
lrActlScnAngl : LREAL;
bErr          : BOOL;
udiErrId      : UDINT;
udiErrArg     : UDINT;
```

stSunBld: Positioniertelegramm, siehe ST_BA_SunBld [▶ 337]

bActv: Entspricht dem booleschen Wert *bActv* im Jalousie-Telegramm ST_BA_SunBld [▶ 337] und dient zur reinen Anzeige, ob der Baustein ein aktives Telegramm sendet.

lrActlScnPos: Zeigt die gespeicherte relative Jalousiehöhenposition [%] der aktuell angewählten Szene an.

lrActlScnAngl: dto. Lamellenwinkel [°]

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe Fehlercodes [▶ 340].



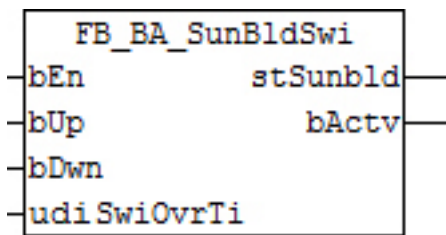
Sollte ein Fehler anstehen, so wird diese Automatik deaktiviert und Position und Winkel auf 0 gesetzt. Das bedeutet, dass bei Verwendung einer Prioritätssteuerung automatisch eine andere Funktion niedrigerer Priorität (siehe Übersicht [▶ 45]) die Steuerung der Jalousie übernimmt. Bei direkter Beschaltung hingegen wird die Jalousie auf Position/Winkel 0 fahren.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.118 FB_BA_SunBldSwi

Mit Hilfe dieses Bausteins lassen sich die Jalousieansteuerung [FB_BA_SunBldActr](#) [▶ 288] bzw. die Rollladensteuerung [FB_BA_RolBldActr](#) [▶ 279] im Handbedienmodus ansteuern. Die Verbindung erfolgt dabei über das Positioniertelegramm [stSunBld](#) [▶ 337] entweder direkt oder mit einer zusätzlichen Prioritätssteuerung.



Funktionsbeschreibung

Struktur des Jalousie-Positioniertelegramms [stSunBld](#) [▶ 337].

```

TYPE ST_BA_SunBld:
STRUCT
    lrPos      : LREAL;
    lrAngl     : LREAL;
    bManUp     : BOOL;
    bManDwn    : BOOL;
    bManMod    : BOOL;
    bActv      : BOOL;
END_STRUCT
END_TYPE
    
```

Betrieb

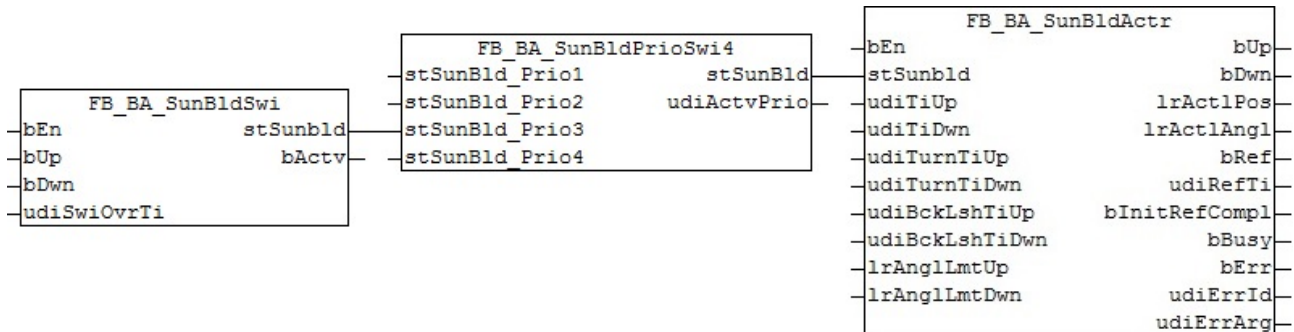
Der Funktionsbaustein steuert über die Befehlseingänge *bUp* und *bDwn* den Jalousiebaustein [FB_BA_SunBldActr](#) [▶ 288] bzw. den Rollladenbaustein [FB_BA_RolBldActr](#) [▶ 279] im Handbetrieb an, wobei *bUp* Vorrang hat. Die Befehle werden an die jeweiligen Kommandos *bManUp* und *bManDwn* des Positioniertelegramms weiter gereicht. Ist ein Befehlseingang länger als die eingetragene Zeit *udiSwiOvrTi* [ms] aktiviert, so geht der entsprechende Steuerbefehl in Selbsthaltung. Ein erneutes Aktivieren eines Befehlseinganges löscht diese Selbsthaltung wieder.

Mit Aktivierung des Bausteins über *bEn* wird im Positioniertelegramm das Bit *bActv* sofort gesetzt. Damit meldet der Baustein an einem Prioritätsschalter ([FB_BA_SunBldPrioSwi4](#) [▶ 295] bzw. [FB_BA_SunBldPrioSwi8](#) [▶ 297]) seinen Vorrang gegenüber niedrigen Prioritäten an. Der Baustein kennt nun 2 Modi: reines Handfahren über *bUp* und *bDwn* - diese setzen im Positioniertelegramm das Bit *bManMod* und blockieren damit am Aktorbaustein die Annahme von Automatikbefehlen auf Position. Der zweite Modus ist der Szenenaufwurf über *bCIScn*. Dieser gibt über das Positioniertelegramm eine Position und einen Winkel der über *udiSlcdScn* gewählten Szene an (Auswahl des Feldes *arrSunBldScn*). Dabei wird das Handmodus-Bit im Positioniertelegramm *bManMod* abgelöscht und der Aktorbaustein weiß, dass er auf eine bestimmte Position zu fahren hat. Der Eingang *bCIScn* wird gegenüber den Handbefehlen nachrangig betrachtet. Ist im ersten Zyklus der Bausteinaktivierung (*bEn*) weder *bUp*, noch *bDwn*, noch *bCIScn* gesetzt, so "springt" der Baustein in den Hand-Modus (Positioniertelegramm: Bit *bManMod*) und verhält sich damit für diesen Fall genau so, wie der [FB_BASunBldSwi](#) [▶ 301].

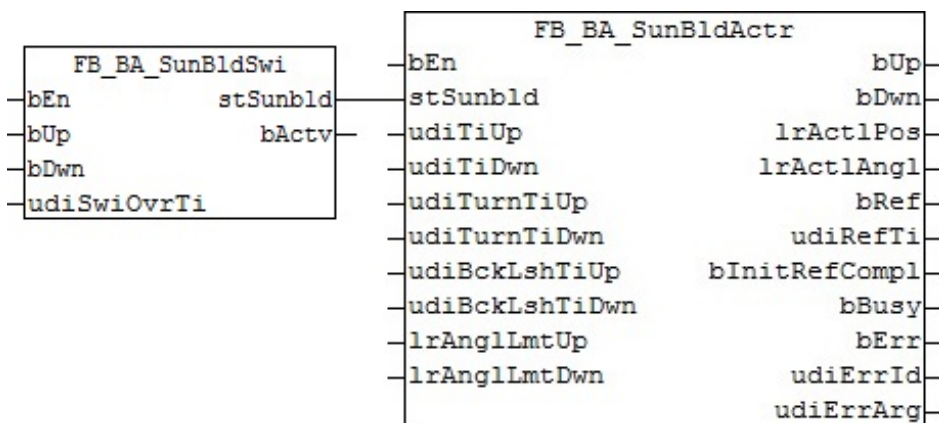
Verknüpfung an Jalousiebaustein

Der Handbedienbaustein kann entweder über eine voran gestellte Prioritätssteuerung [FB_BA_SunBldPrioSwi4](#) [▶ 295] bzw. [FB_BA_SunBldPrioSwi8](#) [▶ 297] oder aber direkt an den Jalousiebaustein angeschlossen werden. Die Verbindung erfolgt dabei über das Positioniertelegramm [stSunBld](#) [▶ 337].

Verwendung einer Prioritätssteuerung:



Direkte Beschaltung:



Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
bUp      : BOOL;
bDwn     : BOOL;
udiSwiOvrTi : UDINT;
```

bEn: Steht dieser Eingang auf FALSE, so ist der Baustein ohne Funktion. Im Positioniertelegramm [stSunBld](#) [▶ 337] werden für Position und Winkel jeweils 0 ausgegeben - `bManMod` und `bActv` stehen jeweils auf FALSE. Das bedeutet für eine Beschaltung mit Prioritätssteuerung, dass eine andere Funktionalität die Jalousieansteuerung übernimmt. Eine direkte Beschaltung hingegen lässt die Jalousie direkt auf Position 0, also ganz nach oben fahren, da der Aktorbaustein das Bit `bActv` selbst nicht auswertet.

bUp: Befehlseingang Jalousie hoch

bDwn: Befehlseingang Jalousie herunter

udiSwiOvrTi: Zeit [ms] bis bei dauerhaft aktiviertem Befehlseingang der entsprechende Handbefehl im Positioniertelegramm [stSunBld](#) [▶ 337] in Selbsthaltung geht.

VAR_OUTPUT

```
stSunBld : ST_BA_SunBld;
bActv    : BOOL;
```

stSunBld: Positioniertelegramm, siehe [ST_BA_SunBld](#) [▶ 337]

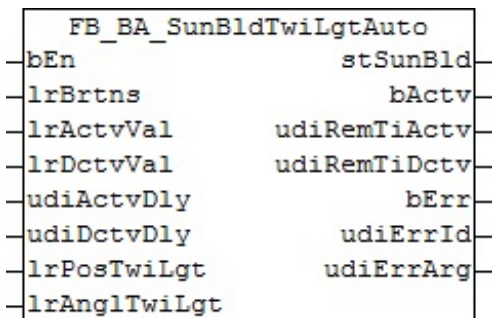
bActv: Entspricht dem booleschen Wert *bActv* im Jalousie-Telegramm ST_BA_SunBld [► 337] und dient zur reinen Anzeige, ob der Baustein ein aktives Telegramm sendet.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.119 FB_BA_SunBldTwiLgtAuto

Dieser Baustein steuert die Jalousie, sollte die Außenhelligkeit einen Grenzwert unterschritten haben.



Funktionsbeschreibung

Die Dämmerungsautomatik arbeitet mit einer Werte- und einer zeitlichen Hysterese: Unterschreitet der Außenhelligkeitswert *lrBrtns* [lux] für die Zeit *udiActvDly* [s] den Wert *lrActvVal* [lux], so ist der Baustein aktiv und wird die an den IN-Variablen angegebenen Jalousiepositionen *lrPosTwiLgt* (Höhe [%]) und *lrAnglTwiLgt* (Lamellenwinkel [°]) am Ausgang im Positioniertelegramm stSunBld [► 337] bereitstellen. Überschreitet die Außenhelligkeit hingegen für die Zeit *udiDctvDly* [s] den Wert *lrDctvVal* [lux], so ist die Automatik nicht mehr aktiv. Der Aktiv-Merker im Positioniertelegramm stSunBld [► 337] wird zurück und die Positionen für Höhe und Winkel auf "0" gesetzt. Eine Funktion niedrigerer Priorität kann dann die Steuerung übernehmen.

Ein-/Ausgänge

VAR_INPUT

```

bEn          : BOOL;
lrBrtns     : LREAL;
lrActvVal   : LREAL;
lrDctvVal   : LREAL;
udiActvDly  : UDINT;
udiDctvDly  : UDINT;
lrPosTwiLgt : LREAL;
lrAnglTwiLgt : LREAL;
    
```

bEn: Steht dieser Eingang auf FALSE, so ist der Baustein ohne Funktion. Im Positioniertelegramm stSunBld [► 337] werden für Position und Winkel jeweils 0 ausgegeben und *bActv* steht auf FALSE. Das bedeutet, dass über die Prioritätssteuerung eine andere Funktionalität die Jalousieansteuerung übernimmt.

lrBrtns: Außenhelligkeit [lx]

lrActvVal: Aktivierungsgrenzwert [lx]

lrDctvVal: Deaktivierungsgrenzwert [lx]

udiActvDly: Aktivierungsverzögerung [s]

udiDctvDly: Deaktivierungsverzögerung [s]

lrPosTwiLgt: Höhenposition der Jalousie [%], sollte die Dämmerungsautomatik aktiv sein

lrAnglTwlLgt: Lamellenwinkel der Jalousie [°], sollte die Dämmerungsautomatik aktiv sein

VAR_OUTPUT

```
stSunBld      : ST_BA_SunBld;
bActv         : BOOL;
udiRemTiActv  : UINT;
udiRemTiDctv  : UINT;
bErr          : BOOL;
udiErrId      : UDINT;
udiErrArg     : UDINT;
```

stSunBld: Ausgabestruktur der Jalousiestellungen, siehe [ST_BA_SunBld \[► 337\]](#)

bActv : Entspricht dem booleschen Wert *bActv* im Jalousie-Telegramm [ST_BA_SunBld \[► 337\]](#) und dient zur reinen Anzeige, ob der Baustein ein aktives Telegramm sendet.

udiRemTiActv: Zeigt die verbleibende Zeit an nach Unterschreitung des Schaltwerts *lrActvVal* bis zur Aktivierung der Automatik [s] an. Solange kein Herunterzählen der Zeit stattfindet, steht dieser Ausgang auf 0.

udiRemTiDctv: Zeigt die verbleibende Zeit an nach Überschreitung des Schaltwerts *lrDctvVal* bis zur Abschaltung der Automatik [s] an. Solange kein Herunterzählen der Zeit stattfindet, steht dieser Ausgang auf 0.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).



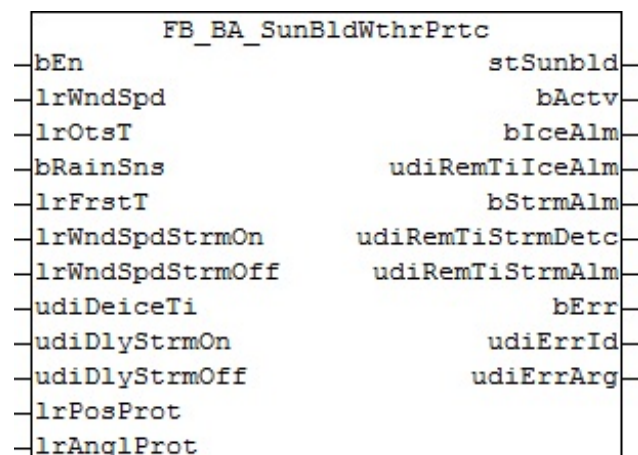
Sollte ein Fehler anstehen, so wird diese Automatik deaktiviert und Position und Winkel auf 0 gesetzt. Das bedeutet, dass bei Verwendung einer Prioritätssteuerung automatisch eine andere Funktion niedrigerer Priorität (siehe [Übersicht \[► 45\]](#)) die Steuerung der Jalousie übernimmt. Bei direkter Beschaltung hingegen wird die Jalousie auf Position/Winkel 0 fahren.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.120 FB_BA_SunBldWthrPrtc

Der Witterungsschutz hat bei der Jalousiesteuerung die höchste Priorität (siehe [Übersicht \[► 45\]](#)) und soll sicherstellen, dass die Jalousie weder durch Eis, noch durch Wind beschädigt wird.



Funktionsbeschreibung

Die Witterungsschutzautomatik hat die Aufgabe, die Jalousie vor zwei unmittelbar bevorstehenden Gefahren zu schützen und sie dafür in eine sichere Position zu fahren:

- **Vereisung:** Eine bevorstehende Vereisung wird dadurch erkannt, dass während einer Niederschlagserkennung an *bRainSns* die gemessene Außentemperatur *lrOtsT* unterhalb des Frost-Grenzwerts *lrFrstT* liegt. Dieses Ereignis wird intern gespeichert und bleibt dann solange bestehen, bis sichergestellt ist, dass das Eis wieder abgetaut ist. Dazu muss die Außentemperatur den Frost-Grenzwert für die eingetragene Enteisungszeit *udiDeiceTi* [s] überschritten haben. Aus Sicherheitsgründen wird das Vereisungsereignis persistent, also über einen SPS-Ausfall hinweg gespeichert. Fällt die Steuerung also während einer Ver- bzw. Enteisungsperiode aus, so gilt die Jalousie nach Wiederanlauf der Steuerung als neu vereist und der Enteisungszeitmesser startet von Neuem.
- **Sturm:** Liegt die gemessene Windgeschwindigkeit für die Zeit *udiDlyStrmOn* [s] über dem Wert *lrWndSpdStrmOn*, so wird davon ausgegangen, dass ein Sturm unmittelbar bevor steht. Erst wenn die Windgeschwindigkeit den Wert *lrWndSpdStrmOff* für die Zeit *udiDlyStrmOff* [s] unterschreitet, gilt der Sturm als abgeflaut und das Fahren der Jalousie als sicher. Aus Sicherheitsgründen wird auch das Sturm-Ereignis persistent gespeichert. Fällt die Steuerung also während eines Sturmes aus, so wird nach Wiederanlauf der Steuerung Ablauf-Zeitgeber von Neuem gestartet.

In beiden Gefahr-Fällen wird die Jalousie in die Schutzposition gefahren die durch *lrPosProt* (Höhenposition in Prozent) und *lrAnglProt* (Lamellenwinkel [°]) vorgegeben wird.

Ein-/Ausgänge

VAR_INPUT

```
bEn           : BOOL;
lrWndSpd      : LREAL;
lrOtsT        : LREAL;
bRainSns     : BOOL;
lrFrstT       : LREAL;
lrWndSpdStrmOn : LREAL;
lrWndSpdStrmOff : LREAL;
udiDeiceTi    : UINT;
udiDlyStrmOn  : UDINT;
udiDlyStrmOff : UDINT;
lrPosProt     : LREAL;
lrAnglProt    : LREAL;
```

bEn: Steht dieser Eingang auf FALSE, so ist der Baustein ohne Funktion. Im Positioniertelegramm [stSunBld](#) [► 337] werden für Position und Winkel jeweils 0 ausgegeben und *bActv* steht auf FALSE. Das bedeutet, dass über die Prioritätssteuerung eine andere Funktionalität die Jalousieansteuerung übernimmt.

lrWndSpd: Windgeschwindigkeit. Die Einheit der Eingabe ist beliebig, jedoch ist es wichtig, dass es keine Werte kleiner als 0 gibt, und die Werte mit zunehmender Geschwindigkeit größer werden.

lrOtsT: Außentemperatur [°C]

bRainSns: Eingang für einen Niederschlagssensor

lrFrstT: Vereisungstemperatur-Grenzwert [°] Celsius. Dieser Wert darf nicht größer als 0 sein. Anderenfalls wird ein Fehler ausgegeben.

lrWndSpdStrmOn: Windgeschwindigkeits-Grenzwert zur Aktivierung des Sturmalarms. Dieser Wert darf nicht kleiner als 0 sein und muss oberhalb des Wertes für die Deaktivierung liegen. Anderenfalls wird ein Fehler ausgegeben. Die Einheit der Eingabe muss die gleiche sein wie die des Einganges *lrWndSpd*. Ein Wert größer als dieser Grenzwert löst nach der eingetragenen Zeit *udiDlyStrmOn* den Alarm aus.

lrWndSpdStrmOff: Windgeschwindigkeits-Grenzwert zur Deaktivierung des Sturmalarms. Dieser Wert darf nicht kleiner als 0 sein und muss unterhalb des Wertes für die Aktivierung liegen. Anderenfalls wird ein Fehler ausgegeben. Die Einheit der Eingabe muss die gleiche sein wie die des Einganges *lrWndSpd*. Ein Wert kleiner oder gleich diesem Grenzwert setzt nach der eingetragenen Zeit *udiDlyStrmOff* den Alarm zurück.

udiDeiceTi: Zeit zum Abtauen der Jalousie nach Vereisung [s]. Danach wird der Vereisungsalarm zurück gesetzt.

udiDlyStrmOn: Verzögerungszeit zur Auslösung des Sturmalarms [s]

udiDlyStrmOff: Verzögerungszeit zum Rücksetzen des Sturmalarms [s]

lrPosProt: Höhenposition der Jalousie [%] im Schutzfall

lrAnglProt: Lamellenwinkel der Jalousie [°] im Schutzfall

VAR_OUTPUT

```
stSunBld      : ST_BA_SunBld;
bActv        : BOOL;
bIceAlm      : BOOL;
udiRemTiIceAlm : UDINT;
bStrmAlm     : BOOL;
udiRemTiStrmDetc : UDINT;
udiRemTiStrmAlm : UDINT;
bErr         : BOOL;
udiErrId     : UDINT;
udiErrArg    : UDINT;
```

stSunBld: Ausgabestruktur der Jalousiestellungen, siehe [ST_BA_SunBld \[► 337\]](#)

bActv: Entspricht dem booleschen Wert *bActv* im Jalousie-Telegramm [ST_BA_SunBld \[► 337\]](#) und dient zur reinen Anzeige, ob der Baustein ein aktives Telegramm sendet.

bIceAlm: Zeigt den Vereisungsalarm an.

udiRemTiIceAlm: Bei aufkommenden Vereisungsfall (*bIceAlm*=TRUE) wird dieser Sekundenzähler auf die Enteisungszeit gesetzt. Sobald die Temperatur über dem eingetragenen Frostpunkt (*lrFrstT*) liegt, werden hier die verbleibenden Sekunden bis zur Entwarnung (*bIceAlm*=FALSE) angezeigt. Solange kein Herunterzählen der Zeit stattfindet, steht dieser Ausgang auf 0.

bStrmAlm: Zeigt den Sturmalarm an.

udiRemTiStrmDetc: Im unkritischen Fall zeigt dieser Sekundenzähler konstant die Alarmverzögerungszeit *udiDlyStrmOn* an. Liegt die gemessene Windstärke *lrWndSpd* über dem Aktivierungsgrenzwert *lrWndSpdStrmOn*, so werden die Sekunden bis zum Alarm herunter gezählt. Solange kein Herunterzählen der Zeit stattfindet, steht dieser Ausgang auf 0.

udiRemTiStrmAlm: Sobald der Sturmalarm ausgelöst wird, zeigt dieser Sekundenzähler zunächst konstant die Deaktivierungsverzögerungszeit *udiDlyStrmOff* des Sturmalarms an. Sinkt die gemessene Windstärke *lrWndSpd* unter den Deaktivierungsgrenzwert *lrWndSpdStrmOff*, so werden die Sekunden bis zur Entwarnung (*bStrmAlm*=FALSE) herunter gezählt. Solange kein Herunterzählen der Zeit stattfindet, steht dieser Ausgang auf 0.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).



Sollte ein Fehler anstehen, so wird diese Automatik deaktiviert und Position und Winkel auf 0 gesetzt. Das bedeutet, dass bei Verwendung einer Prioritätssteuerung automatisch eine andere Funktion niedrigerer Priorität (siehe [Übersicht \[► 45\]](#)) die Steuerung der Jalousie übernimmt. Bei direkter Beschaltung hingegen wird die Jalousie auf Position/Winkel 0 fahren.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.121 FB_BA_SunPrtc

Baustein zur Blendschutzsteuerung mit Hilfe einer Lamellen-Jalousie

FB_BA_SunPrtc	
bEn	stSunBld
stUTC	bActv
udiPosIntval	bErr
lrDegLngd	udiErrId
lrDegLatd	udiErrArg
lrFcdOrtn	
lrFcdAngl	
lrLamWdth	
lrLamDstc	
lrFixPos	
lrMaxLgtIndc	
lrWdwHght	
lrDstcWdwFlr	
stBldPosTab	
ePosMod	

Funktionsbeschreibung

Der Blendschutz wird durch die Variation des Lamellenwinkels und durch die Position der Jalousiehöhe realisiert.

Der Lamellenwinkel wird in Abhängigkeit des Sonnenstandes so eingestellt, dass direkte Blendung unterbunden, aber trotzdem ein maximaler Einfall natürlichen Lichts möglich ist.

Für das Variieren der Jalousiehöhe stehen drei verschiedenen Betriebsarten zur Verfügung:

1. Die Jalousie fährt bei aktiven Sonnenschutz in eine fest vorgegebene Höhe. Der Wert für die Höhe wird mit der Variablen *lrFixPos* vorgegeben.
2. Die Position der Jalousie wird in Abhängigkeit des Sonnenstandes variiert. Die Position wird in der Tabelle (*stBldPosTab* [► 330]) vorgegeben. Siehe auch Beschreibung des *FB_BA_BldPosEntry* [► 257].
3. Die höher der Jalousie wird anhand der Fenstergeometrie so berechnet, dass die Sonnenstrahlen bis zu einer vorgegebenen Tiefe in den Raum einfallen. Die Einfalltiefe der Sonnenstrahlen wird mit der Variablen *lrMaxLgtIndc* definiert.

Damit das Neupositionieren des Lamellenwinkels nicht zu häufig erfolgt, kann mittels der Variablen *udiPosIntval* ein Zeitintervall bestimmt werden, innerhalb dessen keine Anpassung des Lamellenwinkels erfolgt. Um trotzdem eine Blendwirkung auszuschließen, wird der Winkel immer so weiter geändert, dass er bis zum Ablauf des Zeitintervalls ausreicht.

Für die Positionierung der Jalousie und der Einstellung des Lamellenwinkels müssen die folgenden Bedingungen erfüllt sein.

1. Der Eingang *bEn* muss TRUE sein.
2. Die Sonne muss aufgegangen sein. (Elevation > 0)
3. Der Baustein ist richtig parametrisiert (*bErr*=False)

Der Baustein *FB_BA_SunPrtc* ermöglicht einen Blendschutz auf 2 verschiedene parallel arbeitende Weisen:

- Lamellennachführung, so dass das direkt einfallende Licht durch den heruntergelassenen Teil der Jalousie gerade nicht mehr hindurch treten kann.
- Steuerung der Jalousiehöhe, mit 3 verschiedenen Möglichkeiten (Einstellbar über den Enumerator an *ePosMod*):
 - 1) fixe Jalousiehöhe, d.h. keine Änderung (voreingestellt)
 - 2) Jalousiehöhe in Abhängigkeit des Sonnenstandes, definiert über eine Tabelle (*stBldPosTab* [► 330]), siehe auch Beschreibung des *FB_BA_BldPosEntry* [► 257]
 - 3) maximal erwünschter Lichteinfall

Anhand der eingetragenen Parameter, welche weiter unten beschrieben werden, errechnet der Baustein die notwendige Lamellenstellung und Jalousie-Position und übergibt sie in die Ausgabestruktur *stSunBld* [► 337]. Die Ausgabe erfolgt freilich nicht kontinuierlich, da eine ständige Jalousiebewegung als störend empfunden werden würde. Am Eingang *udiPosIntval* lässt sich in Minuten einstellen, in welchem Abstand neue Positionswerte ausgegeben werden sollen.

Jedoch muss zwischen zwei Positionierzeitpunkten die Verschattungskriterien immer erfüllt sein: kein direktes Licht darf durch die Lamellen treten und der erwünschte Lichteinfall durch die Jalousiehöhe muss begrenzt bleiben, geht man zunächst davon aus, dass die Jalousiehöhe über den Modus "maximal erwünschter Lichteinfall" gesteuert wird. Daher werden intern zwei Jalousie- und Lamellenstellungen errechnet: die für den jetzigen und die für den nächsten Schaltpunkt. Diejenige Stellung, bei der die Jalousie mehr geschlossen ist, ist dann die gültige.

Die Positionierung in Intervallen beginnt genau dann, wenn folgende 3 Bedingungen erfüllt sind:

- Der Eingang *bEn* muss auf TRUE stehen.
- Der Baustein darf nicht durch falsche Parametrierung im Fehlerzustand sein (*bErr*=TRUE).
- Die Sonne muss aufgegangen sein, d.h. die Sonnenhöhe (Elevation) muss größer 0° sein. Hierbei handelt es sich um eine interne Sicherheitsabfrage, da die Begrenzung auf min. 0° vom Anwender eigentlich über die Programmierung des Einganges *bEn* erfolgen soll, siehe [Übersicht Sonnenschutzautomatik \(Verschattungskorrektur\)](#) [▶ 45].

Sind diese 3 Bedingungen nicht erfüllt, so wird in der Positionierstruktur das Aktiv-Bit (*bActv*) auf FALSE, die Jalousiehöhe auf 0% und der Lamellenwinkel auf 0% gestellt.

Fehlerbehandlung

Folgende Fehleingaben werden erkannt:

Immer:

- Die Dauer des Positionierintervalls ist gleich Null oder überschreitet 720 min.
- Der eingetragene Längengrad ist nicht im gültigen Bereich von -180°..180°.
- Der eingetragene Breitengrad ist nicht im gültigen Bereich von -90°..90°.
- Der eingetragene Wert für die Fassadenneigung *lrFcdAngl* ist außerhalb des gültigen Bereiches von -90°..90°.
- Der Wert für den Lamellenabstand (*lrLamDstc*) ist größer oder gleich dem Wert für die Lamellenbreite (*lrLamWdth*). Dies stellt keine "gültige" Jalousie dar, da die Lamellen nicht vollständig schließen könnten. Mathematisch gesehen würde dieses zu Fehlern führen.
- Der eingetragene Wert für den Lamellenabstand *lrLamDstc* ist gleich Null.
- Der eingetragene Wert für die Lamellenbreite *lrLamWdth* ist gleich Null.

Nur wenn Positionierung "fixe Jalousiehöhe" angewählt - *ePosMod*=*ePosModFix*:

- Der eingetragene Wert für die fixe Jalousiehöhe (*lrFixPos*) ist größer als 100 oder kleiner als 0.

Nur wenn Positionierung "maximaler Lichteinfall" angewählt - *ePosMod*=*ePosModMaxIndc*:

- Das Bit "Werte gültig" (*bVld*) in der Positioniertabelle *stBldPosTab* ist nicht gesetzt - ungültige Werte: siehe [FB_BA_BldPosEntry](#).

Nur wenn Positionierung "maximaler Lichteinfall" angewählt - *ePosMod*=*ePosModMaxIndc*.

- Der eingetragene Wert für die Fensterhöhe *lrWdwHght* ist kleiner oder gleich Null.
- Der eingetragene Abstand Fensterunterkante zu Boden *lrDstcWdwFlr* ist kleiner Null.
- Der eingetragene Wert für den maximal gewünschten Lichteinfall *lrMaxLgtIndc* ist kleiner oder gleich Null.

Ein-/Ausgänge

VAR_INPUT

```
bEn          : BOOL;
stUTC       : TIMESTRUCT;
udiPosIntval : UDINT;
lrDegLngd   : LREAL;
lrDegLatd   : LREAL;
lrFcdOrtn   : LREAL;
lrFcdAngl   : LREAL;
lrLamWdth   : LREAL;
lrLamDstc   : LREAL;
```

```
lrFixPos      : LREAL;
lrMaxLgtIndc  : LREAL;
lrWdwHght     : LREAL;
lrDstcWdwFlr  : LREAL;
stBldPosTab   : ST_BA_BldPosTab;
ePosMod       : E_BA_PosMod;
```

bEn: Wenn dieser Eingang auf FALSE gesetzt wird, ist die Positionierung nicht aktiv, d.h. in der Positionierstruktur stSunBld vom Typ ST_BA_SunBld [▶ 337] wird das aktiv-Bit (*bActv*) zurückgesetzt und der Baustein selbst verharrt in einem Stillstands-Modus. Ist der Baustein hingegen aktiviert, so ist das aktiv-Bit auf TRUE und der Baustein gibt in der Positionierstruktur zu den entsprechenden Zeiten seine Stellwerte durch (*rPos*, *rAngl*).

stUTC: Eingabe der aktuellen Uhrzeit als koordinierte Weltzeit (UTC, Universal Time Coordinated - im älteren Sprachgebrauch auch GMT, Greenwich-Mean-Time). Mit Hilfe des Bausteins FB_BA_GetTime [▶ 325] kann diese Zeit aus einem Zielsystem gelesen werden.

i Ein Zeitrücksprung von mehr als 300s führt, wenn die Jalousie nach den o.a. Kriterien in der Sonne steht und der Blendschutz aktiv ist, zu einer sofortigen Neupositionierung. Diese Funktionalität ist eingefügt worden, um einen nachvollziehbaren Programmablauf zu gewährleisten.

udiPosIntval: Positionierintervall in Minuten - Zeitspanne zwischen zwei Ausgaben von Jalousiestellungen. Gültiger Bereich: 1min..720min.

lrDegLngd: Geographische Länge (Längengrad) [°]. Gültiger Bereich: -180°..180°

lrDegLatd: Geographische Breite (Breitengrad) [°]. Gültiger Bereich: -90°..90°

lrFcdOrtn: Fassadenausrichtung [°]

Dabei gilt auf der Nordhalbkugel für die Fassadenausrichtung (Blick aus dem Fenster):

Blickrichtung	Fassadenausrichtung
Nord	$\beta=0^\circ$
Ost	$\beta=90^\circ$
Süd	$\beta=180^\circ$
West	$\beta=270^\circ$

Für die Südhalbkugel gilt hingegen:

Blickrichtung	Fassadenausrichtung
Süd	$\beta=0^\circ$
Ost	$\beta=90^\circ$
Nord	$\beta=180^\circ$
West	$\beta=270^\circ$

lrFcdAngl: Fassadenneigung [°]. Siehe Fassadenneigung

lrLamWdth: Breite der Lamellen in mm, siehe Skizze

lrLamDstc: Lamellenabstand in mm, siehe Skizze

lrFixPos: Fixe (konstante) Jalousiehöhe [0..100%]. Gültig, wenn *ePosMod* = *ePosModFix* (siehe Enumerator E_BA_PosMod [▶ 330]).

lrMaxLgtIndc: Maximal gewünschter Lichteinfall in mm gemessen ab Außenseite der Wand (siehe Höhenverstellung) Mit Hilfe der Parameter *lrWdwHght* und *lrDstcWdwFlr* wird in Abhängigkeit des Sonnenstandes errechnet, wie hoch die Jalousie stehen muss, damit der Lichteinfall den Wert *lrMaxLgtIndc* nicht überschreitet. Gültig, wenn *ePosMod* = *ePosModeMaxIncidence* (siehe Enumerator E_BA_PosMod [▶ 330]).

lrWdwHght: Fensterhöhe in mm zur Errechnung des der Jalousiehöhe wenn der Modus des "Maximal gewünschter Lichteinfalls" gewählt ist.

IrDstcWdwFlr: Abstand Boden - Fenstersims in mm zur Errechnung des der Jalousiehöhe wenn der Modus des "Maximal gewünschter Lichteinfalls" gewählt ist.

stBldPosTab: Tabelle von 6 Stützpunkten, davon 4 parametrierbar, aus denen dann durch lineare Interpolation eine Jalousieposition in Abhängigkeit des Sonnenstandes gegeben wird. Gültig, wenn $ePosMod = ePosModFix$ (siehe Enumerator [E_BA_PosMod](#) [► 330]). Weitere Beschreibung siehe [FB_BA_BldPosEntry](#) [► 257].

ePosMod: Auswahl des Positioniermodus, siehe Enumerator [E_BA_PosMod](#) [► 330].

VAR_OUTPUT

```
stSunBld      : ST_BA_SunBld;
bActv        : BOOL;
bErr         : BOOL;
udiErrorId   : UDINT;
udiErrArg    : UDINT;
```

stSunBld: Ausgabestruktur der Jalousiestellungen, siehe [ST_BA_SunBld](#) [► 337]

bActv: Der Baustein ist im aktiv-Zustand, das heißt, es liegt kein Fehler an, der Baustein ist freigegeben und der Sonnenstand befindet sich im eingetragenen Fassadenbereich (die Fassade wird beschienen).

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [► 340].

i Sollte ein Fehler anstehen, so wird diese Automatik deaktiviert und Position und Winkel auf 0 gesetzt. Das bedeutet, dass bei Verwendung einer Prioritätssteuerung automatisch eine andere Funktion niedrigerer Priorität (siehe [Übersicht](#) [► 45]) die Steuerung der Jalousie übernimmt. Bei direkter Beschaltung hingegen wird die Jalousie auf Position/Winkel 0 fahren.

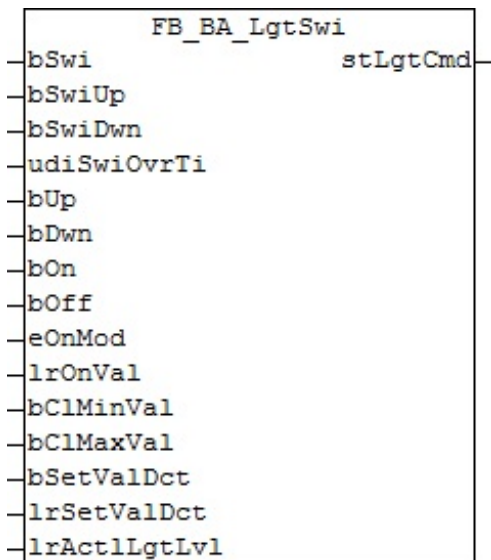
Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.122 FB_BA_LgtSwi

Schalt- und Dimmbaustein - Funktion für zwei Lichttaster

i Dieser Baustein benötigt zwingend eine Lichtwertrückmeldung am Eingang `IrActlLgtLvl` um einen Lichtwechsel (toggle-Befehl) realisieren zu können. Ohne diese Rückmeldung, die entweder von einem Einzelgerät oder als Mittelwert einer Gruppe kommt, weiß der Schalterbaustein nicht um den logischen Zustand des Lichts und kann bei Kurztastendruck an `bSwi` nicht in den gegensätzlichen Zustand springen lassen. Ein Beispiel findet sich unten.



Funktionsbeschreibung

Dieser Baustein ist ein kombinierter Schalter/Dimmer.

Für das zentrale Einschalten an *bOn* sind verschiedene Modi (*eOnMod*) definiert und über diesen Enumerator-Eingang wählbar:

- *eBA_CIMaxVal* : Das Licht wird unmittelbar auf den am Lichtaktor-Baustein gültigen Maximalwert gestellt.
- *eBA_CIMinVal* : Das Licht wird unmittelbar auf den am Lichtaktor-Baustein gültigen Minimalwert gestellt.
- *eBA_CIOnVal* : Das Licht wird unmittelbar auf den am Eingang *lrOnVal* anstehenden Wert gestellt.
- *eBA_CIRstVal* : Das Licht wird unmittelbar auf den am Lichtaktor-Baustein gültigen Reset-Wert gestellt. Dies ist der Wert, der dort vor dem letzten Ausschalten gültig war. Anwendungsbeispiel: Wenn in einem Konferenzraum nach einer Präsentation das Licht wieder eingeschaltet wird.

Alle weiteren Befehle siehe unten.

Der Baustein sendet seine Kommandos über das Befehlstelegramm *stLgtCmd*. [▶ 339](#)

Ein-/Ausgänge

VAR_INPUT

```

bSwi      : BOOL;
bSwiUp    : BOOL;
bSwiDwn   : BOOL;
udiSwiOvrTi : UDINT;
bUp       : BOOL;
bDwn     : BOOL;
bOn      : BOOL;
bOff     : BOOL;
eOnMod   : E_BA_LgtOnMode;
lrOnValue : LREAL;
bClMinVal : BOOL;
bClMaxVal : BOOL;
bSetValDct : BOOL;
lrSetValDct : LREAL;
lrActlLgtLvl : LREAL;

```

bSwi: Schaltereingang Ein/Aus, Auf-/Abdimmen: Ein kurzes Signal (< *udiSwiOvrTi*) schaltet das Licht je nach Zustand aus oder ein. Ein langer Tastendruck dimmt das Licht abwechselnd auf oder ab. Das Aufdimmen endet bei Erreichen des Maximalwerts, das Abdimmen bei Erreichen des Minimalwerts. Eine automatische Dimmrichtungsumkehr erfolgt nicht. Ist das Licht ausgeschaltet, so wird zunächst auf den Minimalwert geschaltet und dann aufgedimmt.

bSwiUp: Schaltereingang Ein/Aus, Aufdimmen: Ein kurzes Signal ($< udiSwiOvrTi$) schaltet das Licht je nach Zustand aus oder ein. Ein langer Tastendruck dimmt das Licht auf. Das Aufdimmen endet bei Erreichen des Maximalwerts. Ist das Licht ausgeschaltet, so wird zunächst auf den Minimalwert geschaltet und dann aufgedimmt.

bSwiDwn: Schaltereingang Ein/Aus, Abdimmen: Ein kurzes Signal ($< udiSwiOvrTi$) schaltet das Licht je nach Zustand aus oder ein. Ein langer Tastendruck dimmt das Licht ab. Das Abdimmen endet bei Erreichen des Minimalwerts. Ist das Licht ausgeschaltet, so wird zunächst auf den Maximalwert geschaltet und dann abgedimmt.

udiSwiOvrTi: Umschaltzeit [ms] zwischen Tastendruckerkennung kurz und lang und damit zwischen Schalten und Dimmen.

bUp: Aufdimmen des Lichts. Dieser Eingang hat **keine** Schaltfunktion und wird durch die Umschaltzeit $udiSwiOvrTi$ nicht beeinflusst. Die Funktion ist bei ausgeschaltetem Licht unwirksam.

bDwn: Abdimmen des Lichts. Dieser Eingang hat **keine** Schaltfunktion und wird durch die Umschaltzeit $udiSwiOvrTi$ nicht beeinflusst. Die Funktion ist bei ausgeschaltetem Licht unwirksam.

bOn: Zentrales Einschalten. Der Einschaltwert richtet sich nach der Auswahl an $eOnMod$, s.u..

bOff: Zentrales Ausschalten

eOnMod: Auswahl Einschaltwert ([E_BA_LgtOnMode](#) ▶ 338)

IrOnVal: [0..100%] Einschaltwert, wenn ein zentraler Einschaltbefehl über bOn kommt und an $eOnMod$ der Modus $eBA_ClOnVal$ angewählt ist.

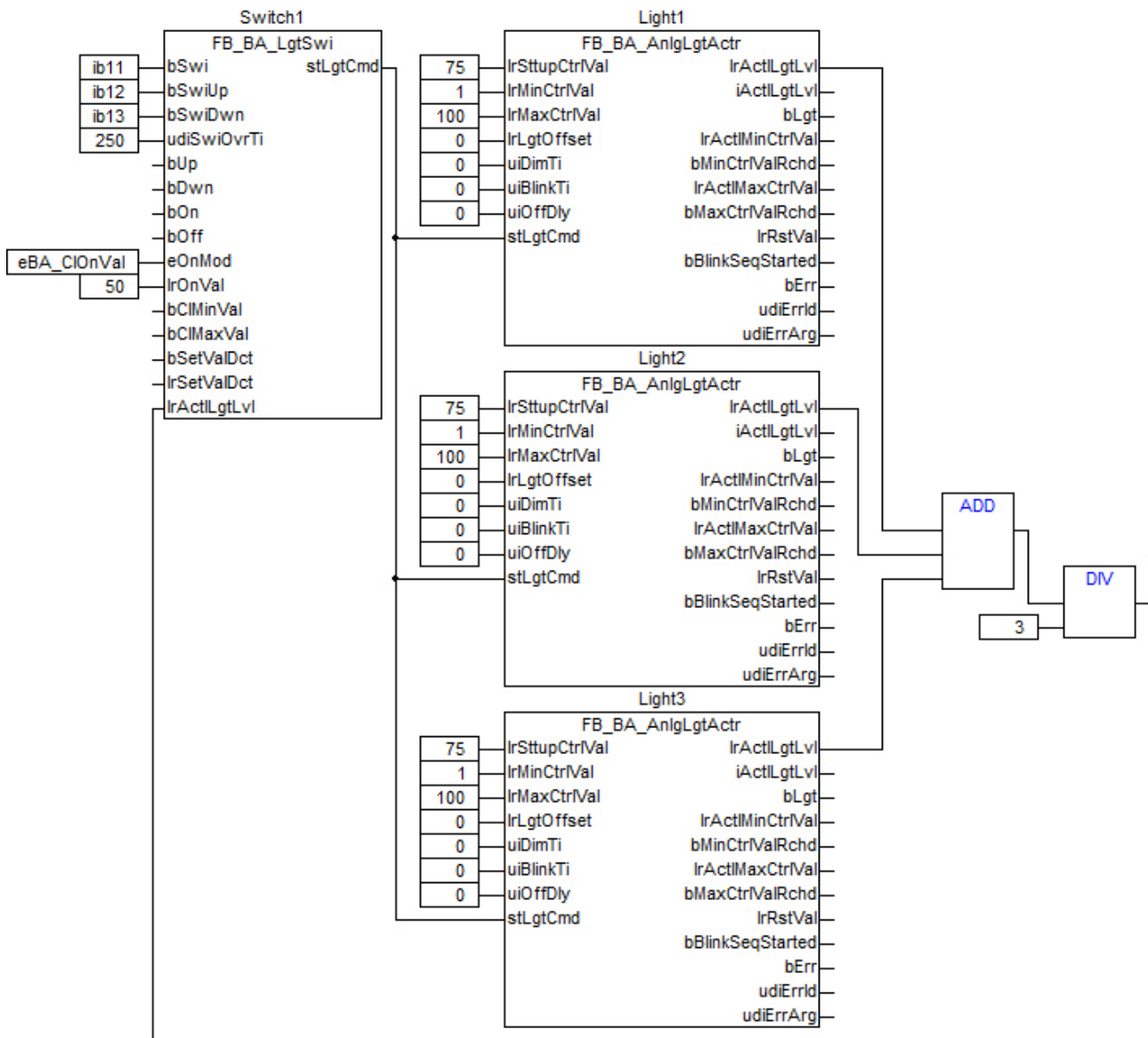
bCIMinVal: Der Lichtaktor-Baustein bekommt den Befehl auf den Minimalwert zu schalten.

bCIMaxVal: Der Lichtaktor-Baustein bekommt den Befehl auf den Maximalwert zu schalten.

bSetValDct/IrSetValDct: Der Lichtaktor-Baustein bekommt den Befehl auf den Wert $IrSetValDct$ [0..100%] zu schalten. **Dieser Wert wird am Lichtaktor noch mit dem Wert $IrLgtOffset$ beaufschlagt.**

IrActILgtLvl: [0..100%] Dieser Eingang dient zur Umschalt- (Toggle-) Beurteilung. Bei einer Gruppe von Lampen empfiehlt es sich entweder den Mittelwert aller Aktoren oder den Lichtwert des Gruppenmasters anzulegen. Die Belegung dieses Einganges ist zwingend notwendig, da der Schalterbaustein ohne diese Rückmeldung nicht um den logischen Lichtzustand des Einzelgeräts bzw. der Gruppe weiß.

Beispiel:



VAR_OUTPUT

stLgtCmd : ST_BA_LgtCmd;

stLgtCmd: Befehlstelegramm vom Typ ST_BA_LgtCmd [▶ 339](#)

Voraussetzungen

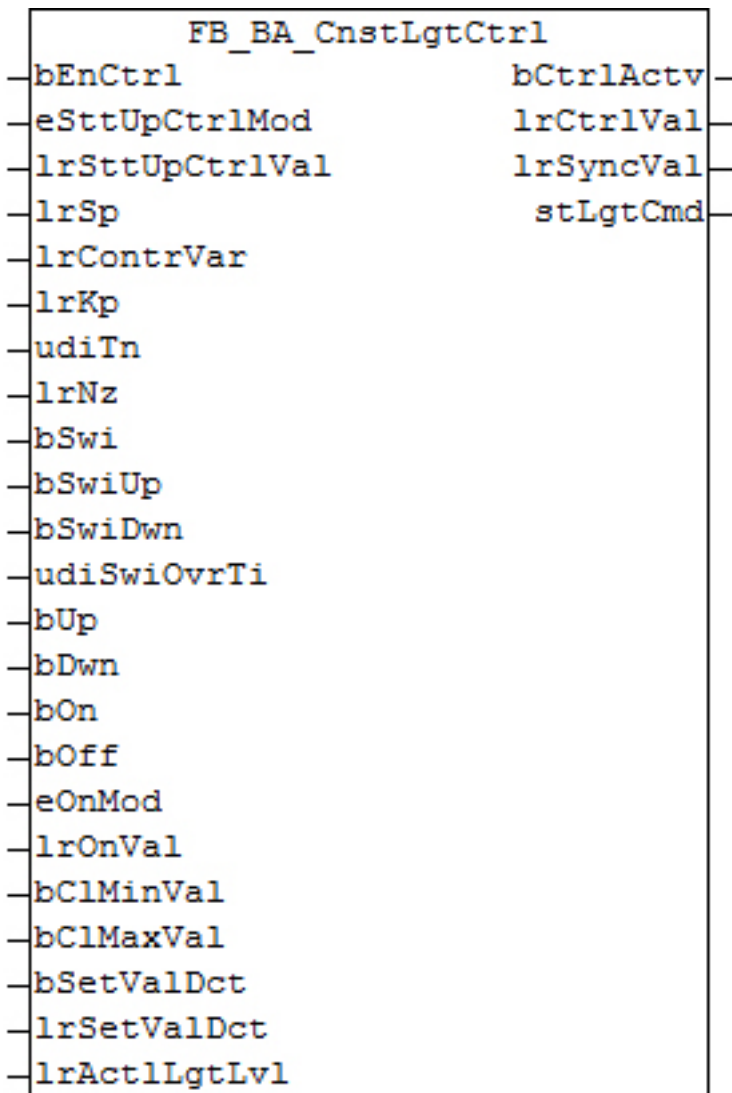
Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.123 FB_BA_CnstLgtCtrl

Konstantlichtregelung für einen Raum



Dieser Baustein benötigt zwingend eine Lichtwertrückmeldung am Eingang IrActLgtLvl um einen Lichtwechsel (toggle-Befehl) realisieren zu können. Ohne diese Rückmeldung, die entweder von einem Einzelgerät oder als Mittelwert einer Gruppe kommt, weiß der Schalterbaustein nicht um den logischen Zustand des Lichts und kann bei Kurztastendruck an bSwi nicht in den gegensätzlichen Zustand springen lassen. Siehe auch [FB_BA_LgtSwi](#) [▶ 310](#).



Funktionsbeschreibung

Dieser Baustein ist ein kombinierter Schalter/Dimmer mit einem zusätzlichen Konstantlichtregel-Modus. Die Konstantlichtregelung arbeitet intern mit einem PI-Regler.

Für das zentrale Einschalten an *bOn* sind verschiedene Modi (*eOnMod*) definiert und über diesen Enumerator-Eingang wählbar:

- *eBA_CIMaxVal* : Das Licht wird unmittelbar auf den am Lichtaktor-Baustein gültigen Maximalwert gestellt.
- *eBA_CIMinVal* : Das Licht wird unmittelbar auf den am Lichtaktor-Baustein gültigen Minimalwert gestellt.
- *eBA_CIOnVal* : Das Licht wird unmittelbar auf den am Eingang *lrOnVal* anstehenden Wert gestellt.
- *eBA_CIRstVal* := Das Licht wird unmittelbar auf den am Lichtaktor-Baustein gültigen Reset-Wert gestellt. Dies ist der Wert, der dort vor dem letzten Ausschalten gültig war. Anwendungsbeispiel: Wenn in einem Konferenzraum nach einer Präsentation das Licht wieder eingeschaltet wird.

Alle weiteren Befehle siehe unten.

Der Baustein sendet seine Kommandos über das Befehlstelegramm *stLgtCmd* [► 339].

Ein-/Ausgänge

VAR_INPUT

```

bEnCtrl      : BOOL;
eSttUpCtrlMod : E_BA_CtrlSttupMod;
lrSttUpCtrlVal : LREAL;
lrSp         : LREAL;
lrContrVar   : LREAL;
lrKp         : LREAL;
udiTn        : UDINT;
lrNz         : LREAL;
bSwi         : BOOL;
bSwiUp       : BOOL;
bSwiDwn      : BOOL;
udiSwiOvrTi  : UDINT;
bUp          : BOOL;
bDwn         : BOOL;
bOn          : BOOL;
bOff         : BOOL;
eOnMod       : E_BA_LgtOnMode;
lrOnValue    : LREAL;
bClMinVal    : BOOL;
bClMaxVal    : BOOL;
bSetValDct   : BOOL;
lrSetValDct  : LREAL;
lrActILgtLvl : LREAL;

```

bEnCtrl: Eine positive Flanke an diesem Eingang schaltet die Regelung aktiv. **Werden die Schalteingänge *bSwi...bSetValDct* betätigt, so wird die Regelung wieder deaktiviert, egal ob am Eingang *bEnCtrl* ein TRUE-Signal anliegt.**

eSttUpCtrlMod/IrSttUpCtrlVal: Gibt an, mit welchem Wert der Regler aufstarten soll: siehe [E_BA_CtrlSttupMod \[► 338\]](#).

lrSp: Sollwert-Eingang [lx]

lrContrVar: Istwert-Eingang [lx]

lrKp: Verstärkungsfaktor des internen PI-Reglers

udiTn: Nachstellzeit (Integrationszeit) des internen PI-Reglers [ms]

lrNz: Neutrale Zone: Ist der Absolutwert der Regelabweichung kleiner als $lrNz/2$, so stoppt der Regler intern seine Stellgrößenberechnung, bleibt jedoch weiterhin aktiv.

bSwi: Schaltereingang Ein/Aus, Auf-Abdimmen: Ein kurzes Signal ($< udiSwiOvrTi$) schaltet das Licht je nach Zustand aus oder ein. Ein langer Tastendruck dimmt das Licht abwechselnd auf oder ab. Das Aufdimmen endet bei Erreichen des Maximalwerts, das Abdimmen bei Erreichen des Minimalwerts. Eine automatische Dimmrichtungsumkehr erfolgt nicht. Ist das Licht ausgeschaltet, so wird zunächst auf den Minimalwert geschaltet und dann aufgedimmt.

bSwiUp: Schaltereingang Ein/Aus, Aufdimmen: Ein kurzes Signal ($< udiSwiOvrTi$) schaltet das Licht je nach Zustand aus oder ein. Ein langer Tastendruck dimmt das Licht auf. Das Aufdimmen endet bei Erreichen des Maximalwerts. Ist das Licht ausgeschaltet, so wird zunächst auf den Minimalwert geschaltet und dann aufgedimmt.

bSwiDwn: Schaltereingang Ein/Aus, Abdimmen: Ein kurzes Signal ($< udiSwiOvrTi$) schaltet das Licht je nach Zustand aus oder ein. Ein langer Tastendruck dimmt das Licht ab. Das Abdimmen endet bei Erreichen des Minimalwerts. Ist das Licht ausgeschaltet, so wird zunächst auf den Maximalwert geschaltet und dann abgedimmt.

udiSwiOvrTi: Umschaltzeit [ms] zwischen Tastendruckerkennung kurz und lang und damit zwischen Schalten und Dimmen.

bUp: Aufdimmen des Lichts. Dieser Eingang hat **keine** Schaltfunktion und wird durch die Umschaltzeit *udiSwiOvrTi* **nicht** beeinflusst. Die Funktion ist bei ausgeschaltetem Licht unwirksam.

bDwn: Abdimmen des Lichts. Dieser Eingang hat **keine** Schaltfunktion und wird durch die Umschaltzeit *udiSwiOvrTi* **nicht** beeinflusst. Die Funktion ist bei ausgeschaltetem Licht unwirksam.

bOn: zentrales Einschalten. Der Einschaltwert richtet sich nach der Auswahl an *eOnMod*, s.u..

bOff: zentrales Ausschalten.

eOnMod: Auswahl Einschaltwert (E_BA_LgtOnMode [► 338]).

IrOnVal: [0..100%] Einschaltwert, wenn ein zentraler Einschaltbefehl über *bOn* kommt und an *eOnMod* der Modus *eBA_CiOnVal* angewählt ist.

bCIMinVal: Der Lichtaktor-Baustein bekommt den Befehl auf den Minimalwert zu schalten.

bCIMaxVal: Der Lichtaktor-Baustein bekommt den Befehl auf den Maximalwert zu schalten.

bSetValDct/IrSetValDct: Der Lichtaktor-Baustein bekommt den Befehl auf den Wert *IrSetValDct* [0..100%] zu schalten. **INFO:** dieser Wert wird am Lichtaktor noch mit dem Wert *IrLgtOffset* beaufschlagt.

IrActlLgtLvl: [0..100%] Dieser Eingang dient zur Umschalt- (Toggle-) Beurteilung. Bei einer Gruppe von Lampen empfiehlt es sich entweder den Mittelwert aller Aktoren oder den Lichtwert des Gruppenmasters anzulegen.

VAR_OUTPUT

```
bCtrlActv : BOOL;
lrCtrlVal : LREAL;
stLgtCmd  : ST_BA_LgtCmd;
```

bCtrlActv: Die Konstantlichtregelung ist aktiv

lrCtrlVal: Im Befehlstelegramm *stLgtCmd* (vom Typ *ST_BA_LgtCmd*) [► 339] aktuell ausgegebener Wert.

lrSyncVal: [0..100%] Zur Kontrolle : Durch den Modus *eSttUpCtrlMod* (siehe Eingänge) aktuell gewählter Aufstartwert.

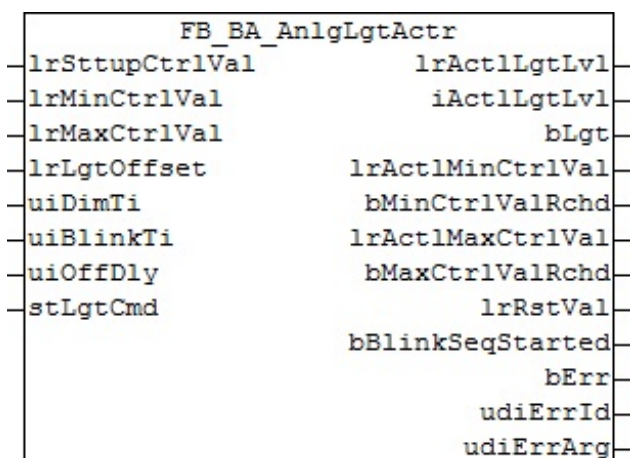
stLgtCmd: Befehlstelegramm vom Typ *ST_BA_LgtCmd*. [► 339]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.124 FB_BA_AnlgLgtActr

Funktionsbaustein zum Ansteuern eines analogen Lichtaktors, beispielsweise über eine KL2751



Funktionsbeschreibung

Der Baustein empfängt seine Kommandos über das Befehlstelegramm `stLgtCmd`. [▶ 339](#)

Als konventioneller Analog-Ansteuerungsbaustein besitzt er Lichtwert-Ausgänge als LREAL-Wert [0..100%], als INT-Wert [0..32767] und als booleschen Wert (EIN/AUS), der bei bestimmten Lampenarten das Schalten eines AUS-Relais ermöglicht.



Dieser Baustein muss in jedem SPS-Zyklus aufgerufen werden, da die SPS-Zykluszeit in die Berechnung der Dimmrampe mit einfließt.

Ein-/Ausgänge

VAR_INPUT

```
lrSttupCtrlVal : LREAL;
lrMinCtrlVal   : LREAL;
lrMaxCtrlVal   : LREAL;
lrLgtOffset    : LREAL;
udiDimTi       : UDINT;
uiBlinkTi      : UINT;
uiOffDly       : UINT;
stLgtCmd       : ST_BA_LgtCmd;
```

lrSttupCtrlVal : Lichtwert nach Neustart. [*lrMinCtrlVal*..*lrMaxCtrlVal*]. Eingabe als Prozentwert (0%..100%). Dieser Wert wird nur im ersten Aufruf-Zyklus übernommen. Fehlerhafte Einträge kleiner *lrMinCtrlVal* oder größer *lrMaxCtrlVal* werden automatisch begrenzt.

lrMinCtrlVal: Minimaler Lichtausgabewert. [*1*..*lrMaxCtrlVal*]. Eingabe als Prozentwert (1%..100%). Eine Änderung dieses Wertes wirkt unmittelbar. Fehlerhafte Einträge kleiner 1.0 oder größer 100.0 werden automatisch begrenzt.

Zur Kontrolle wird der resultierende Wert am Ausgang *lrActlMinCtrlVal* ausgegeben, wobei zu beachten ist, dass auch Wertänderungen über das Befehlstelegramm `stLgtCmd` angenommen werden. Die letzte Änderung ist jeweils gültig.

lrMaxCtrlVal: Maximaler Lichtausgabewert. [*lrMinCtrlVal*..100%]. Eingabe als Prozentwert (1%..100%). Eine Änderung dieses Wertes wirkt unmittelbar. Fehlerhafte Einträge kleiner *lrMinCtrlVal* oder größer 100.0 werden automatisch begrenzt.

Zur Kontrolle wird der resultierende Wert am Ausgang *lrActlMaxCtrlVal* ausgegeben, wobei zu beachten ist, dass auch Wertänderungen über das Befehlstelegramm `stLgtCmd` angenommen werden. Die letzte Änderung ist jeweils gültig.

lrLgtOffset: Möglichkeit den Helligkeitswert zu erhöhen oder zu erniedrigen. Dieser Effekt ist nur in Verbindung mit dem Befehl `bSetCtrlValDct` der Kommandostruktur `stLgtCmd` wirksam. In diesem Fall wird der Lichtfaktor auf den Wert $lrSetCtrlValDct + lrLgtOffset$ gesetzt. Derr Wert kann dabei nicht den eingestellten Maximalwert überschreiten bzw. den Minimalwert unterschreiten. Ein Abschalten jedoch ist möglich, wenn $lrSetCtrlValDct + lrLgtOffset$ kleiner oder gleich Null ist.

uiDimTi: Dimmrampe [ms]: Zeit in der von 0 auf 100% gedimmt wird.

uiBlinkTi / udiOffDly: Mit diesen beiden Zeiten [s] lässt sich eine Ausschaltwarnung, beispielsweise für eine Treppenhausbeleuchtung erzielen. Zu dem Zeitpunkt, an dem die Leuchte normalerweise abschaltet, kommen diese beiden Intervalle **zusätzlich** zum Tragen: für die Zeit *uiBlinkTi* [s] wird das Licht ausgeschaltet und danach für die Zeit *uiOffDly* [s] noch einmal eingeschaltet.

stLgtCmd: Befehlstelegramm vom Typ `ST_BA_LgtCmd` [▶ 339](#)

VAR_OUTPUT

```
lrActlLgtLvl   : LREAL;
iActlLgtLvl    : INT;
bLgt           : BOOL;
lrActlMinCtrlVal : LREAL;
bMinCtrlValRchd : BOOL;
lrActlMaxCtrlVal : LREAL;
bMaxCtrlValRchd : BOOL;
lrRstVal       : LREAL;
```

```

bErr      : BOOL;
udiErrId  : UDINT;
udiErrArg : UDINT;

```

IrActILgtLvl: Aktuelle Stellgröße [0..100%]. Berechnung siehe unten.

iActILgtLvl: Aktuelle Stellgröße [0..32767]

bLgt: Aktuelle Stellgröße (EIN / AUS)

IrActlMinCtrlVal: Aktueller Min-Wert (über Eingang oder Kommandostruktur beschrieben)

bMinCtrlValRchd: Der Aktor hat seinen Minimalwert erreicht.

IrActlMaxCtrlVal: Aktueller Max-Wert (über Eingang oder Kommandostruktur beschrieben)

bMaxCtrlValRchd: Der Aktor hat seinen Maximalwert erreicht.

IrRstVal: Letzter Wert vor Aus als Wiedereinschaltwert, d.h. vor dem Ausschalten wird der aktuelle Stellwert hier gespeichert.

bErr: Lichtbaustein im Fehlerzustand

udiErrId / udiErrArg: Fehlernummer und -argument zur Bestimmung der Ursache, siehe [Fehlercodes](#) [► 340].

Voraussetzungen

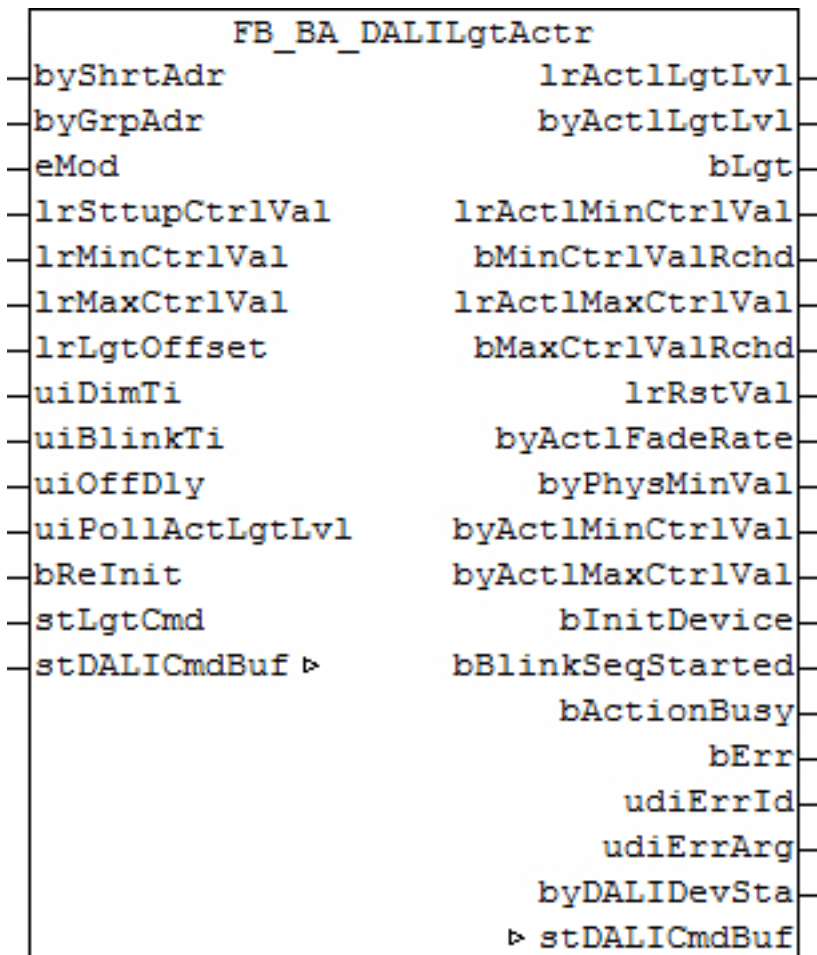


Die Warnmeldungen zur Autokorrektur der Minimal- und Maximalwerte liegen nur für einen SPS-Zyklus an, da die Korrektur den Fehler selbst behebt.

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.125 FB_BA_DALILgtActr

Funktionsbaustein zum Ansteuern eines DALI-Lichtaktors



Funktionsbeschreibung

Der Baustein empfängt seine Kommandos über das Befehlstelegramm `stLgtCmd`. [▶ 339]

Als Licht-Ansteuerungsbaustein für das DALI-Subsystem besitzt er als IN-OUT-Variable eine Referenz auf die Befehlsspeicher-Struktur `stDALICmdBuf`.

Ein-/Ausgänge

VAR_INPUT

```

byShrtAdr      : BYTE;
byGrpAdr       : BYTE;
eMod           : BOOL;
lrSttupCtrlVal : LREAL;
lrMinCtrlVal   : LREAL;
lrMaxCtrlVal   : LREAL;
lrLgtOffset    : LREAL;
uiDimTi        : UINT;
uiBlinkTi      : UINT;
uiOffDly       : UINT;
uiPollActLgtLvl : UINT;
bReinit        : BOOL;
stLgtCmd       : ST_BA_LgtCmd;
    
```

byShrtAdr : Kurzadresse des anzusprechenden DALI-Vorschaltgeräts. Diese Adresse muss im gültigen Bereich von 0..63 sein, da der Baustein ein vorhandenes Vorschaltgerät braucht um die Status abzufragen. Bei einer ungültigen Adresse wird der Baustein nicht ausgeführt und ein entsprechender Fehlercode ausgegeben. Die Kurzadresse bezieht sich auf die DALI-Befehle, welche sich durch die Eingänge an diesem Baustein ergeben:

- lrSttupCtrlVal (Lichtwert nach Neustart)
- lrMinCtrlVal (Setzen minimaler Lichtausgabewert)
- lrMaxCtrlVal (Setzen maximaler Lichtausgabewert)

- `uiDimTi` (Definieren der Dimmrampe -> Änderung `FadeRate`)

byGrpAdr: Gruppenadresse. Wird nur berücksichtigt, wenn das Vorschaltgerät durch diesen Baustein als Master betrieben wird: `eMod=eBA_DALIModGrpMst` oder `eMod=eBA_DALIModGrpSgl`.



Auch wenn dieser Baustein als Master eine Gruppe ansteuern sollte, ist in jedem Fall die Kurzadresse des Masters unter `byShrtAdr` einzutragen, da aus dem entsprechenden Vorschaltgerät Daten stellvertretend für die ganze Gruppe ausgelesen werden.

eMod: Auswahl über das Einzel-/Gruppen bzw. Master-/Slaveverhalten des Bausteins, siehe [E_BA_DALIMod](#) [[▶ 339](#)].

IrSttupCtrlVal: Lichtwert nach Neustart

IrMinCtrlVal: Minimaler Lichtausgabewert. [0..100%]. Eine Änderung am Bausteineingang bewirkt eine Umparametrierung (Kurzadresse) im Vorschaltgerät. Dort wird es nach unten hin auf den physikalischen Min-Wert und nach oben hin auf den Maximalwert begrenzt.

IrMaxCtrlVal: Maximaler Lichtausgabewert. [0..100%]. Eine Änderung am Bausteineingang bewirkt eine Umparametrierung (Kurzadresse) im Vorschaltgerät. Dort wird es nach unten hin auf den physikalischen Min-Wert begrenzt.

IrLgtOffset: Möglichkeit den Helligkeitswert zu erhöhen oder zu erniedrigen. Dies kann in einer Lampen-Gruppenschaltung bei den Leuchten sinnvoll sein, die in der Nähe von Tageslicht liegen und weniger hell sein müssen, als die Lampen, welche die dunklen Bereiche des Raumes ausleuchten. Der Lichtwert kann durch diesen Parameter jedoch nicht über `IrMaxCtrlVal` oder unter `IrMinCtrlVal` gesteuert werden. Dieser Offset wirkt sich nur auf Befehle aus, welche über die Kommandostruktur als `bSetCtrlValDct` an den Baustein gelangen.

udiDimTi: Dimmrampe [s]: Zeit in der von 0 auf 100% gedimmt wird. Wird bausteinintern auf eine DALI-`FadeRate` umgerechnet und im Vorschaltgerät abgespeichert.

uiBlinkTi / uiOffDly: Mit diesen beiden Zeiten [ms] lässt sich eine Ausschaltwarnung, beispielsweise für eine Treppenhausbeleuchtung erzielen. Zu dem Zeitpunkt, an dem die Leuchte normalerweise abschaltet kommen diese beiden Intervalle **zusätzlich** zum Tragen: für die Zeit `uiBlinkTi` [s] wird das Licht ausgeschaltet und danach für die Zeit `uiOffDly` [s] noch einmal eingeschaltet.

uiPollActLgtLvl: Zykluszeit [s], mit der im Hintergrund der aktuelle Istwert (ACTUAL DIM LEVEL) ausgelesen wird. Damit das Dimmen der Lampen nicht gestört wird, wird das Auslesen immer in der niedrigsten Priorität durchgeführt. Wird der Wert auf 0 gesetzt, so wird das Auslesen gesperrt. Zusätzlich zum aktuellen Lichtwert wird auch das Statusbyte des Vorschaltgeräts ausgelesen und an den Ausgang `byDALIDevSta` geschrieben.

bReInit: Flankengesteuerter Eingang: Startet die [Initialisierung](#) [[▶ 320](#)] des Bausteins erneut.

stLgtCmd: Befehlstelegramm vom Typ `ST_BA_LgtCmd`. [[▶ 339](#)]

VAR_OUTPUT

```
lrActlLgtLvl      : LREAL;
byActlLgtLvl     : BYTE;
bLgt             : BOOL;
lrActlMinCtrlVal : LREAL;
bMinCtrlValRchd  : BOOL;
lrActlMaxCtrlVal : LREAL;
bMaxCtrlValRchd  : BOOL;
lrRstVal         : LREAL;
byActlFadeRate   : BYTE;
byPhysMinVal     : BYTE;
byActlMinCtrlVal : BYTE;
byActlMaxCtrlVal : BYTE;
bInitDevice      : BOOL;
bBlinkSeqStarted : BOOL;
bActionBusy     : BOOL;
bErr             : BOOL;
udiErrId         : UDINT;
udiErrArg        : UDINT;
byDALIDevSta     : BYTE;
```


IrActILgtLvl: Aktueller Lichtausgabewert [0..100%]

byActILgtLvl: Aktueller Lichtausgabewert [0..254]

bLgt: Licht ist ein- oder ausgeschaltet

IrActlMinCtrlVal: Aktuell gültiger Minimalwert. Die Ausgabe dieses Wertes wird nur in der Initialisierungsphase des Bausteins und damit des betreffenden Vorschaltgeräts durch eine QueryMinLevel-Abfrage verifiziert. Im weiteren Verlauf wird der Wert an diesem Ausgang ausgegeben, der über den Bausteineingang bzw. über die Kommandostruktur in das Vorschaltgerät einprogrammiert wird. In diesem kann eine Begrenzung auf den physikalischen Min-Wert oder auf den Max-Wert stattfinden.

bMinCtrlValRchd: Der Aktor hat seinen Minimalwert erreicht.

IrActlMaxCtrlVal: Aktuell gültiger Maximalwert. Die Ausgabe dieses Wertes wird nur in der Initialisierungsphase des Bausteins und damit des betreffenden Vorschaltgeräts durch eine QueryMinLevel-Abfrage verifiziert. Im weiteren Verlauf wird der Wert an diesem Ausgang ausgegeben, der über den Bausteineingang bzw. über die Kommandostruktur in das Vorschaltgerät einprogrammiert wird. In diesem kann eine Begrenzung auf den Min-Wert oder auf 100% stattfinden.

bMaxCtrlValRchd: Der Aktor hat seinen Maximalwert erreicht.

IrRstVal: In dieser Variable wird der Lichtwert vor Ausschalten gesichert. Über den Befehl *bCIRstVal* in der Kommandostruktur kann dieser Wert wieder hergestellt werden. Anwendungsbeispiel: Wenn in einem Konferenzraum nach einer Präsentation das Licht wieder eingeschaltet wird.

byActIFadeRate: Aktuell gültige DALI-FadeRate, welche sich aus der Dimmzeit (Eingang *uiDimTi*) ergibt.

byPhysMinVal: Während der Initialisierungsphase des Bausteins ausgelesener physikalischer Minimalwert [1..254].

byActlMinCtrlVal: siehe *IrActlMinCtrlVal*. Von [0..100%] auf den Wertebereich von [0..254] umgerechnet.

byActlMaxCtrlVal: siehe *IrActlMaxCtrlVal*. Von [0..100%] auf den Wertebereich von [0..254] umgerechnet.

blnitDevice: Lichtbaustein in der Initialisierungsphase: Es werden folgende Aktionen ausgeführt:

- Die am Eingang anliegende Lichttrampe (*uiDimTi*) wird als DALI-FadeRate in das Vorschaltgerät (Kurzadresse) einprogrammiert.
- Der am Eingang anliegende Mindestwert (*IrMinCtrlVal*) wird als DALI-MinValue in das Vorschaltgerät (Kurzadresse) einprogrammiert.
- Der am Eingang anliegende Maximalwert (*IrMaxCtrlVal*) wird als DALI-MaxValue in das Vorschaltgerät (Kurzadresse) einprogrammiert.
- Der physikalische Mindestwert des DALI-Vorschaltgeräts wird gelesen und am Bausteinausgang ausgegeben.
- Der tatsächlich gültige Mindestwert des DALI-Vorschaltgeräts wird gelesen und am Bausteinausgang ausgegeben.
- Der tatsächlich gültige Maximalwert des DALI-Vorschaltgeräts wird gelesen und am Bausteinausgang ausgegeben.
- Das Vorschaltgerät (Kurzadresse) wird auf den am Eingang anliegende Startwert (*IrSttupCtrlVal*) gesetzt.
- Der tatsächlich gültige Lichtausgabewert des DALI-Vorschaltgeräts wird gelesen und am Bausteinausgang ausgegeben.

bBlinkSeqStarted: Die Blinksequenz wurde gestartet.

bActionBusy: Ein DALI-Befehl wird ausgeführt.

bErr: Lichtbaustein im Fehlerzustand

udiErrld / udiErrArg: Fehlernummer und -argument zur Bestimmung der Ursache, siehe [Fehlercodes](#) [► 340].

byDALIDevSta: Statusbyte des DALI-Vorschaltgeräts. Wird zusammen mit der automatischen Abfrage des Lichtwerts im Hintergrund ausgelesen, siehe Eingang *uiPollActLgtLvl*.

VAR_IN_OUT

```
stDALICmdBuf : ST_DALIV2CommandBuffer;
```

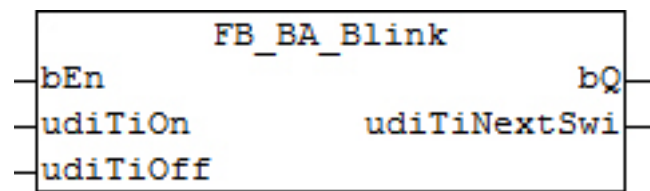
stDALICmdBuf: Referenz auf die Befehlsspeicher-Struktur stDALICmdBuf

Voraussetzungen

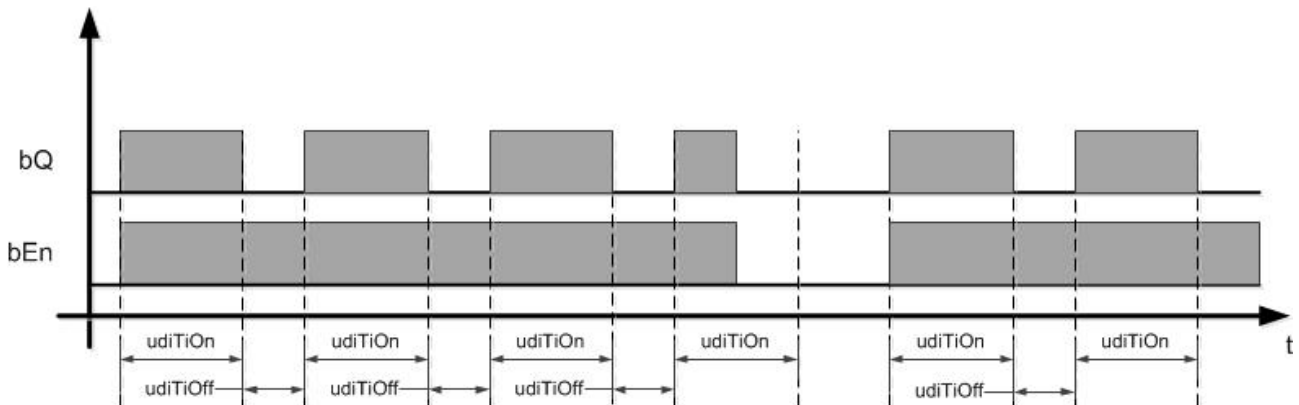
Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBASubsystems-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.126 FB_BA_Blink

Einfacher Oszillator-Baustein

**Funktionsbeschreibung**

Dieser Baustein ist ein Oszillator mit einstellbarer Puls- und Pausenzeit, *udiTiOn* und *udiTiOff* [s]. Er wird mit einem TRUE-Signal an *bEn* freigegeben und beginnt mit der Puls-Phase.



udiTiNextSwi ist ein Countdown [ms] zum nächsten Wechsel von *bQ*.

Ein-Ausgänge**VAR_INPUT**

```
bEn      : BOOL;
udiTiOn  : UDINT;
udiTiOff : UDINT;
```

bEn: Bausteinfreigabe

udiTiOn: Puls-Zeit [ms]

udiTiOff: Pausenzeit [ms]

VAR_OUTPUT

```
bQ      : BOOL;
udiTiNextSwi : UDINT;
```

bQ: Oszillatorausgang.

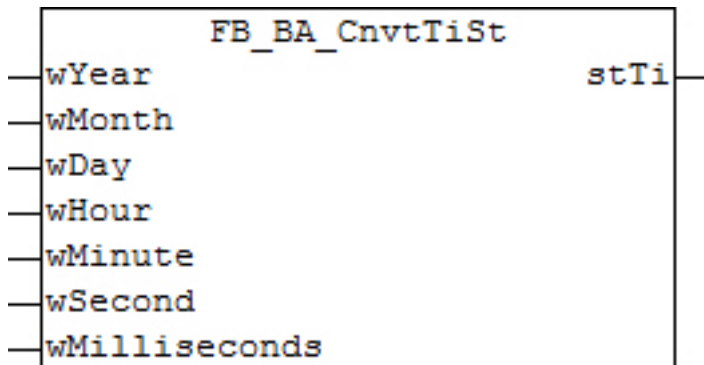
udiTiNextSwi: Countdown bis zum nächsten Wechsel von *bQ* [s]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.127 FB_BA_CnvtTiSt

Zusammenfügen einer Zeitstruktur



Funktionsbeschreibung

Mit dem Funktionsbaustein *FB_BA_CnvtTiSt* können die verschiedenen Bestandteile einer Zeitstruktur zu dieser selbst zusammen gefasst werden.

i Der Baustein verfügt über keine Überprüfung fehlerhafter Eingaben, etwa eine Stundeneingabe von 99, da diese Überprüfung sinnvollerweise in den angeschlossenen Bausteinen erfolgt, die ihrerseits die Zeitstruktur ohnehin überprüfen müssen. Die zulässigen Grenzen werden hier jedoch bei der Variablenerläuterung aufgezeigt.

VAR_INPUT

```
wYear      : WORD;
wMonth     : WORD;
wDay       : WORD;
wHour      : WORD;
wMinute    : WORD;
wSecond    : WORD;
wMilliseconds : WORD;
```

wYear: Jahr (1970..2106)

wMonth: Monat (1..12)

wDay: Tag im Monat (1..31)

wHour: Stunde (0..23)

wMinute: Minuten (0..59)

wSecond s: Sekunden (0..59)

wMillisecond: Millisekunden (0..999)

VAR_OUTPUT

```
stTi      : TIMESTRUCT;
```

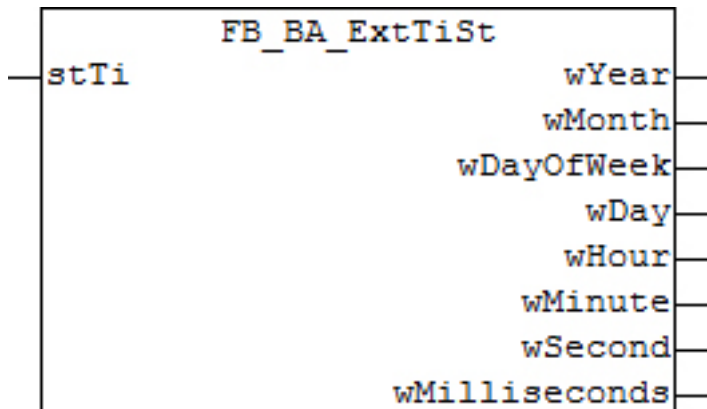
stTi: Ausgabe: Zeitstruktur

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.128 FB_BA_ExtTiSt

Zeit-Struktur-Auflösung

**Funktionsbeschreibung**

Der Funktionsbaustein FB_BA_ExtTiSt löst eine Zeitstruktur in die verschiedenen Bestandteile auf, um sie so etwa für Zeitbedingungen nutzbar zu machen.

Ein-/Ausgänge**VAR_INPUT**

```
stTi      : TIMESTRUCT;
```

stTi: Eingabe: Zeitstruktur

VAR_OUTPUT

```
wYear      : WORD;
wMonth     : WORD;
wDay       : WORD;
wHour      : WORD;
wMinute    : WORD;
wSecond    : WORD;
wMilliseconds : WORD;
```

wYear: Jahr (1970..2106)

wMonth: Monat (1..12)

wDay: Tag im Monat (1..31)

wHour: Stunde (0..23)

wMinute: Minuten (0..59)

wSecond ys: Sekunden (0..59)

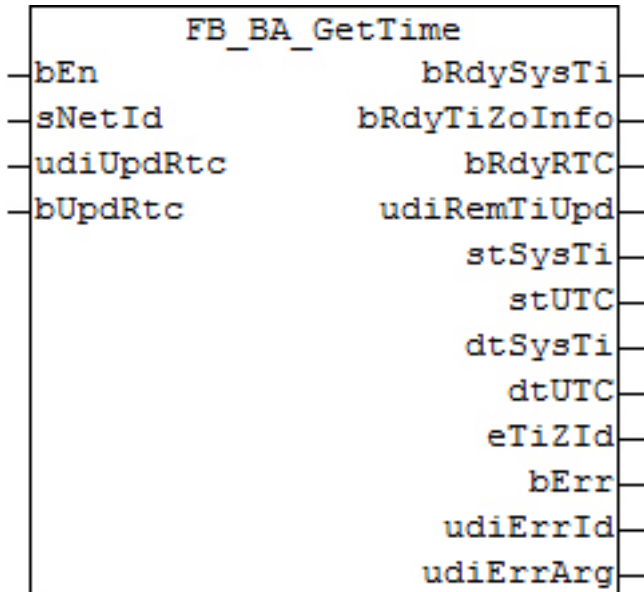
wMillisecond: Millisekunden (0..999)

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.129 FB_BA_GetTime

Systemzeit auslesen



Funktionsbeschreibung

Mit diesem Funktionsbaustein kann eine interne Uhr (Real Time Clock RTC) in der TwinCAT SPS realisiert werden. Die RTC-Uhr wird mit der Freigabe des Funktionsbausteins über *bEn* mit der aktuellen NT-Systemzeit initialisiert. Es wird ein Systemtakt der CPU benutzt um die aktuelle RTC-Zeit zu berechnen. Der Funktionsbaustein muss in jedem Zyklus der SPS einmal aufgerufen werden, damit die aktuelle Zeit berechnet werden kann. Intern wird in dem Funktionsbaustein eine Instanz der Funktionsbausteine NT_GetTime, FB_GetTimeZoneInformation und RTC_EX2 aufgerufen. Die Ausgabe der Zeit erfolgt an den Ausgängen *stSysTi* für die gelesene Systemzeit und *stUtcTi* für die koordinierte Weltzeit (Coordinated Universal Time). Diese wird intern aus der Systemzeit und der Zeitzone ermittelt. Ist die Systemzeit und/oder die Zeitzone fehlerhaft eingegeben worden, so wird auch die UTC-Zeit nicht richtig sein.

Die Systemzeit wird über den einzustellenden Timer (*udiUpdRTC* [sec]) zyklisch ausgelesen und damit die interne RTC-Uhr synchronisiert. In demselben Zyklus werden auch die Zeitinformationen (Zeitzone, Zeitverschiebung zu UTC und Sommer-/ Winterzeit) ausgelesen. Der Ausgang *udiRemTiUpd* zeigt die verbleibenden Sekunden bis zum nächsten Lesezyklus. Die ausgegebenen Zeitstrukturen *stSysTi* und *stUtcTi* können mit Hilfe des Bausteins FB_BA_ExtTiSt [▶ 324] weiter in die Bestandteile - Tag, Monat, Stunde, Minute etc. - aufgelöst werden.

i Lese-/ Wartezyklus

Während des Lesezyklus fallen die Ausgänge `bRdySysTi` und `bRdyTiZoInfo` auf FALSE, der Enumerator `eTiZId` zeigt 0 = `eTimeZoneID_Unknown`. Wurde erfolgreich gelesen, so gehen die Ausgänge wieder auf TRUE bzw. zeigen die jeweilige Information Sommer- oder Winterzeit an, sofern sie verfügbar ist. War das Lesen nicht erfolgreich - es wird intern 5s lang auf eine Antwort gewartet - so bleiben die Ausgänge auf FALSE bzw. 0 stehen und ein erneuter Wartezyklus bis zum nächsten Lesezyklus wird gestartet. Im Fehlerfall wird die interne RTC-Uhr zwar nicht synchronisiert und kann immer noch eine richtige Zeit anzeigen, die Zeitinformationen jedoch können fehlerhaft sein und damit auch die UTC-Zeit. Fehler während des Lesezyklus schlagen sich in jedem Fall in der Anzeige an `bErr` und `udiErrId` / `udiErrArg` nieder. Der Countdown-Ausgang wird erst dann neu gestartet, wenn der Wartezyklus beginnt.

Ein-Ausgänge

VAR_INPUT

```
bEn      : BOOL;
sNetId   : T_AmsNetId;
udiUpdRtc : UDINT;
bUpdRtc  : BOOL;
```

bEn: Freigabe des Bausteins. Ist `bEn` = TRUE, so wird die RTC-Uhr mit der NT-Systemzeit initialisiert.

sNetId : Hier kann die `AmsNetId` des TwinCAT-Rechners angegeben werden dessen NT-Systemzeit als Zeitbasis gelesen werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

udiUpdRtc: Zeitangabe [s], mit der die RTC-Uhr mit der NT-Systemzeit regelmäßig synchronisiert wird. Dieser Wert wird intern auf ein Minimum von 5s begrenzt, um die Abarbeitung der internen Bausteine zu gewährleisten.

bUpdRtc: Parallel zu der Zeitangabe `tGetSystemTime` kann die RTC-Uhr über eine positive Flanke an diesem Eingang synchronisiert werden.

VAR_OUTPUT

```
bRdySysTi   : BOOL;
bRdyTiZoInfo : BOOL;
bRdyRTC     : BOOL;
udiRemTiUpd  : UDINT;
stSysTi     : TIMESTRUCT;
stUTC       : TIMESTRUCT;
dtSysTi     : DT;
dtUTC       : DT;
eTiZId      : E_TimeZoneID;
bErr        : BOOL;
udiErrId    : UDINT;
udiErrArg   : UDINT;
```

bRdySysTi: Die Systemzeit wurde erfolgreich aus dem Zielsystem gelesen.

bRdyTiZoInfo: Die zusätzlichen Zeitinformationen (Zeitzone, Zeitverschiebung zu UTC und Sommer-/ Winterzeit) wurden erfolgreich gelesen.

bRdyRTC: Wurde der Funktionsbaustein mindestens einmal initialisiert, so wird dieser Ausgang gesetzt. Ist dieser Ausgang gesetzt, dann sind die Werte für das Datum, Uhrzeit und Millisekunden an den Ausgängen gültig.

udiRemTiUpd: Countdown zur nächsten Synchronisation bzw. Aktualisierung der Zeitinformationen.

stSysTi: Systemzeit des ausgelesenen Zielsystems. Die Zeitstruktur kann mit Hilfe des Bausteins `FB_BA_ExtTiSt` [▶ 324] weiter in die Bestandteile - Tag, Monat, Stunde, Minute etc. - aufgelöst werden.

i Ist der Baustein nicht aktiviert (`bEn=FALSE`), so zeigen der Ausgang `stSysTi` in seinen Teilelementen (Tag Monat, etc.) jeweils 0 an.

stUTC: Koordinierte Weltzeit. Diese wird intern aus der Systemzeit und den gelesenen Zeitinformationen des Zielsystems ermittelt. Die Zeitstruktur kann mit Hilfe des Bausteins [FB_BA_ExtTiSt](#) [▶ 324] weiter in die Bestandteile - Tag, Monat, Stunde, Minute etc. - aufgelöst werden.



Ist der Baustein nicht aktiviert (*bEn=FALSE*), so zeigen der Ausgang *stUTC* in seinen Teilelementen (Tag Monat, etc.) jeweils 0 an.

dtSysTi / dtUTC: Wie *stSysTi / stUTC*, jedoch im DATE-AND-TIME-Format: Jahr-Monat-Tag-Stunden-Minuten-Sekunden.



Ist der Baustein nicht aktiviert (*bEn=FALSE*), so zeigen die Ausgänge *dtSysTi* und *dtUTC* jeweils DT#1970-01-01-00:00 an, da dies die Untergrenze ist und es den Nullen in der Strukturdarstellung von *stSysTi / stUTC* entspricht.

eTiZld: Enumerator zur Sommer-/ Winterzeitinformation

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn die eingetragenen Parameter fehlerhaft sind.

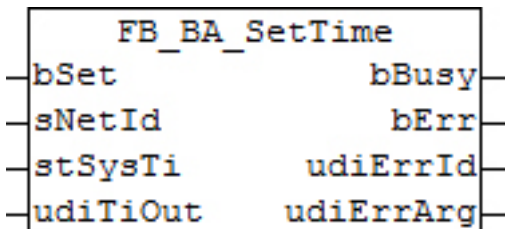
udiErrld / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes](#) [▶ 340].

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.2.130 FB_BA_SetTime

Systemzeit setzen



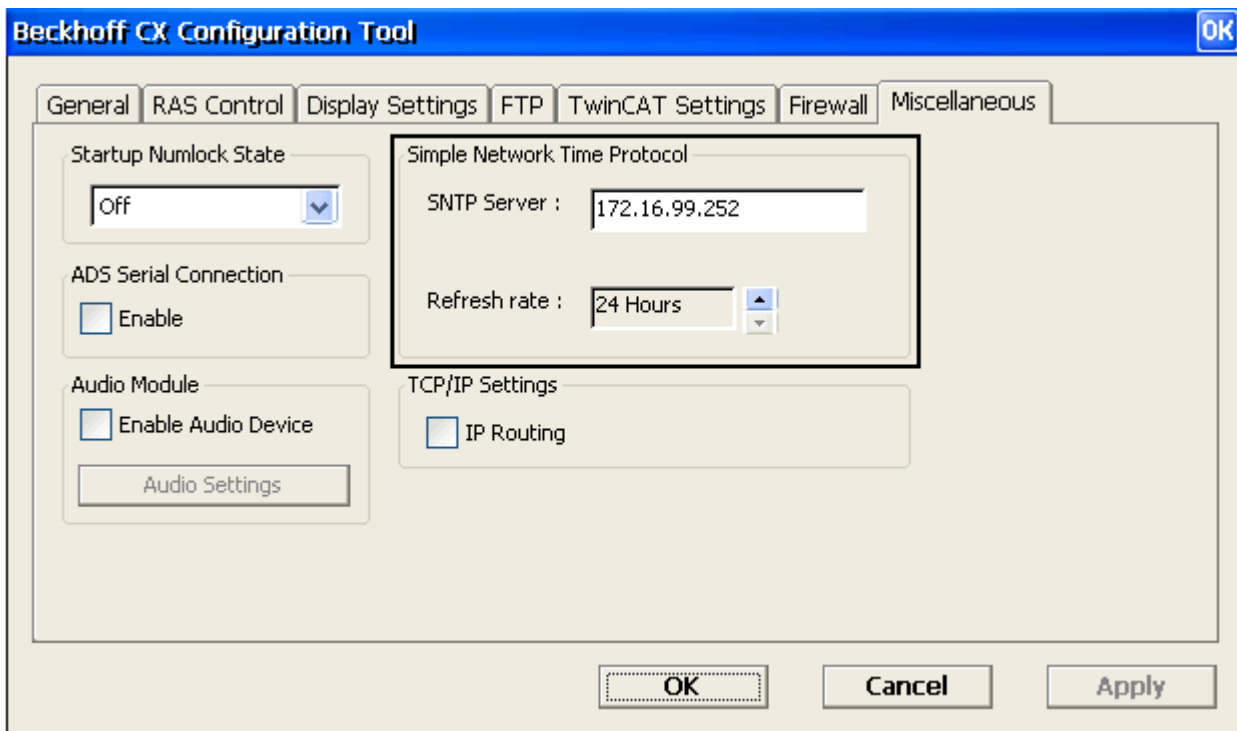
Funktionsbeschreibung

Mit dem Funktionsbaustein *FB_BA_SetTime* kann die lokale NT-Systemzeit und das Datum eines TwinCAT-Systems gesetzt werden (die lokale NT-Systemzeit wird in der Taskleiste eingeblendet). Die Systemzeit wird über die Struktur *stSysTi* vorgegeben.

Intern wird in dem Funktionsbaustein eine Instanz des Funktionsbausteins **NT_SetLocalTime** aus der TcUtilities-Bibliothek aufgerufen.



Die lokale NT-Systemzeit können Sie auch mit der Hilfe des SNTP-Protokolls mit einer Referenzzeit synchronisieren. Mehr Informationen dazu finden Sie im Beckhoff Information System unter: **Beckhoff Information System > Embedded-PC > Betriebssysteme > CE > SNTP: Simple Network Time Protocol**



Ein-/Ausgänge

VAR_INPUT

```
bSet      : BOOL;
sNetId    : T_AmsNetId;
stSysTi   : TIMESTRUCT;
udiTiOut  : UDINT;
```

bSet: Aktivierung des Funktionsbausteins mit einer steigenden Flanke

sNetId: Hier kann die AmsNetId des TwinCAT-Rechners angegeben werden dessen lokale NT-Systemzeit gesetzt werden soll. Für den lokalen Rechner kann auch ein Leerstring *sNetId := ""*; angegeben werden.

stS ysTi Struktur mit der neuen lokalen NT-Systemzeit. Ist die Zeit als Struktur nicht vorhanden, so empfiehlt sich die Verwendung des Bausteins [FB_BA_CnvtTiSt \[► 323\]](#), welcher die Teilvariablen von Datum und Uhrzeit in einer Struktur zusammen führt.

udiTiOut: Gibt die Timeout-Zeit [s] an, die bei der Ausführung nicht überschritten werden darf.

VAR_OUTPUT

```
bBusy     : BOOL;
bError    : BOOL;
udiErrorId : UDINT;
bInvalidParameter : BOOL;
```

bBusy: Bei der Aktivierung des Funktionsbausteins über eine steigende Flanke an *bSetLocalTime* wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung der NT-Systemzeit erfolgen, dann wird dieser Ausgang gesetzt. *bError* wird mit der Aktivierung des Funktionsbausteins über die Eingangsvariable *bSetLocalTime* zurück gesetzt.

bErr: Dieser Ausgang wird auf TRUE geschaltet, wenn entweder die zu übertragene Systemzeit fehlerhaft ist, siehe unten oder aber ein ADS-Fehler in der Übertragung stattfindet. Beide Fehlerarten schlagen sich in einer eindeutigen Beschreibung der Ausgangsvariablen *udiErrId / udiErrArg* nieder.

udiErrId / udiErrArg: Enthält die Fehlernummer und das Fehlerargument. Siehe [Fehlercodes \[► 340\]](#).

Grenzen der Zeitvorgabe

Die angelegte Zeitstruktur *stSysTi* wird bausteinintern auf Grenzen geprüft:

Jahr: 1970..2106 - dieser Bereich bezieht sich auf den maximal darstellbaren Jahres-Zeitbereich im TwinCAT-System

Monat: 1..12

Monatstag: 1..28/29/30/31, je nach Monat und Schaltjahr - beides wird ermittelt und berücksichtigt

Stunde: 0..23

Minute: 0..59

Sekunde: 0..59

Millisekunde: 0..999

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3 Aufzählungen und Strukturen

8.3.1 E_BA_AlmSta

Enumerator Status der Alarmmeldungen

```

TYPE E_BA_AlmSta :
(
    E_BA_AlmSta_UNKOWN      := 0,
    E_BA_AlmSta_OK         := 1,
    E_BA_AlmSta_TRIGGERED  := 2,
    E_BA_AlmSta_GONE       := 3,
    E_BA_AlmSta_ACKNOWLEDGED := 4,
    E_BA_AlmSta_OFFLINE    := 5
);
END_TYPE
    
```

E_BA_AlmSta_UNKOWN: Der Alarm wurde keiner Anlage zugewiesen.

E_BA_AlmSta_OK: Es steht kein Alarm an.

E_BA_AlmSta_TRIGGERED: Ein Alarm wurde ausgelöst.

E_BA_AlmSta_GONE: Der Alarm ist gegangen.

E_BA_AlmSta_ACKNOWLEDGED: Der Alarm wurde quittiert, steht aber noch an.

E_BA_AlmSta_OFFLINE: Der Alarm ist offline.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.2 E_BA_PosMod

Enumerator zur Definition des Positioniermodus

```
TYPE E_BA_PosMod :
(
    eBA_PosModFix:= 0,
    eBA_PosModTab,
    eBA_PosModMaxIndc
);
END_TYPE
```

eBA_PosModFix: Die Jalousiehöhe nimmt einen festen Wert ein, welcher am Baustein FB_BA_SunPrtc [► 306] über den Wert *lrFixPos* eingestellt wird [%].

eBA_PosModTab: Die Höhenpositionierung erfolgt mit Hilfe einer Tabelle von 6 Stützpunkten, davon 4 parametrierbar. Aus diesen Punkten wird dann durch lineare Interpolation eine Jalousieposition in Abhängigkeit des Sonnenstandes errechnet. Weitere Beschreibung siehe FB_BA_BldPosEntry [► 257].

eBA_PosModMaxIndc: Die Positionierung erfolgt unter Angabe des maximal gewünschten Lichteinfalls.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.3 E_BA_ShObjType

Enumerator zur Auswahl des Verschattungsobjekt-Typs

```
TYPE E_BA_ShObjType :
(
    eBA_ObjTypeTetragon := 0,
    eBA_ObjTypeGlobe := 1
);
END_TYPE
```

eBA_ObjTypeTetragon: Objekttyp ist ein Viereck

eBA_ObjTypeGlobe: Objekttyp ist eine Kugel

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.4 ST_BA_BldPosTab

Struktur der Stützpunkteinträge für die Höhenverstellung der Jalousie

```
TYPE ST_BA_BldPosTab:
STRUCT
    lrSunElv : ARRAY[0..5] OF LREAL;
    lrBlindPos : ARRAY[0..5] OF LREAL;
    bValid : BOOL;
    byDummy : ARRAY[1..3] OF BYTE;
END_STRUCT
END_TYPE
```

lrSunElv / lrBlindPos: Die 6 Stützstellen welche übergeben werden, wobei die Array-Elemente 0 und 5 die oben erwähnten automatisch generierten Rand-Elemente darstellen.

bValid: Gültigkeitsflag für den Baustein [FB_BA_SunPrtc](#) [▶ 306]. Wird vom Baustein [FB_BA_BldPosEntry](#) [▶ 257] auf TRUE gesetzt, wenn die eingegebenen Daten den beschriebenen Gültigkeitskriterien entsprechen.

byDummy: Unbenutzte Füllvariablen zur Herstellung einer 4-Byte-Anordnung. Diese Anordnung ist dann wichtig, wenn diese Struktur Teil eines Datenaustauschs zwischen Controllern unterschiedlicher Architektur (ARM/I486) ist.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.5 ST_BA_Cnr

Information über Fenster-Eckpunkte

```

TYPE ST_BA_Cnr :
STRUCT
  lrX      : LREAL;
  lrY      : LREAL;
  bShdd    : BOOL;
  byDummy  : ARRAY[1..3] OF BYTE;
END_STRUCT
END_TYPE
    
```

lrX: X-Koordinate des Fensters (auf der Fassade)

lrY: Y-Koordinate des Fensters (auf der Fassade)

bShdd: Information, ob dieser Eckpunkt verschattet ist: *bShdd*=TRUE: Eckpunkt ist verschattet.

byDummy: Unbenutzte Füllvariablen zur Herstellung einer 4-Byte-Anordnung. Diese Anordnung ist dann wichtig, wenn diese Struktur Teil eines Datenaustauschs zwischen Controllern unterschiedlicher Architektur (ARM/I486) ist.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.6 ST_BA_ComnMsg

Struktur des Kommunikationstelegrammes an die Bausteine vom Typ [FB_BA_ComMsg](#) [▶ 200].

```

TYPE ST_BA_ComnMsg :
STRUCT
  arrData      : ARRAY [1..gBa_cMaxArrComnMsg] OF ST_BA_ComnMsgData;
  iCntEntryArr : INT;
  bDummy01    : BOOL;
  bDummy02    : BOOL;
END_STRUCT
END_TYPE

TYPE ST_BA_ComnMsgData :
STRUCT
  arrActvPrio : ARRAY [1..16] OF BOOL;
  udiActvPrio : UDINT;
  dwObjId     : DWORD; (*neu*)
  eObjType    : E_BACnetObjectType;
  iId         : INT;
  bAck        : BOOL;
    
```

```

bInAlm      : BOOL;
bOoServ     : BOOL;
bOvrrd      : BOOL;
bFlt        : BOOL;
bHiLmt      : BOOL;
bLoLmt      : BOOL;
bErr         : BOOL;
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.7 ST_BA_CmnMsgTermt

Struktur des Kommunikationstelegramms an die Bausteine vom Typ FB_BA_CmnMsgTermt [► 203].

```

TYPE ST_BA_CmnMsgTermt :
STRUCT
  arrData      : ARRAY [1..gBa_cMaxArrCmnMsgTermt] OF ST_BA_CmnMsgTermtData;
  iCntEntryArr : INT;
  bDummy01     : BOOL;
  bDummy02     : BOOL;
END_STRUCT
END_TYPE

```

```

TYPE ST_BA_CmnMsgTermtData :
STRUCT
  arrActvPrio : ARRAY [1..16] OF BOOL;
  udiActvPrio : UDINT;
  iId         : INT;
  bAck        : BOOL;
  bInAlm      : BOOL;
  bOoServ     : BOOL;
  bOvrrd      : BOOL;
  bFlt        : BOOL;
  bHiLmt      : BOOL;
  bLoLmt      : BOOL;
  bErr        : BOOL;
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.8 ST_BA_FcdElem

Listeneintrag eines Fassadenelementes (Fenster)

```

TYPE ST_BA_FcdElem:
STRUCT
  lrWdwWdth : LREAL;
  lrWdwHght : LREAL;
  stCnr      : ARRAY [1..4] OF ST_BA_Cnr;
  usiGrp     : USINT;
  bVld       : BOOL;
  byDummy    : ARRAY[1..2] OF BYTE;
END_STRUCT
END_TYPE

```

lrWdwWdth: Breite des Fensters

IrWdwHght: Höhe des Fensters

stCnr: Koordinaten der Fenster-Eckpunkte und Information, ob dieser Eckpunkt verschattet ist, siehe [ST_BA_Cnr](#) [▶ 331].

bVld: Plausibilität der eingetragenen Daten: *bVld*=TRUE: Daten sind plausibel.

usiGrp: Gruppenzugehörigkeit des Elements

byDummy: Unbenutzte Füllvariablen zur Herstellung einer 4-Byte-Anordnung. Diese Anordnung ist dann wichtig, wenn diese Struktur Teil eines Datenaustauschs zwischen Controllern unterschiedlicher Architektur (ARM/I486) ist.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

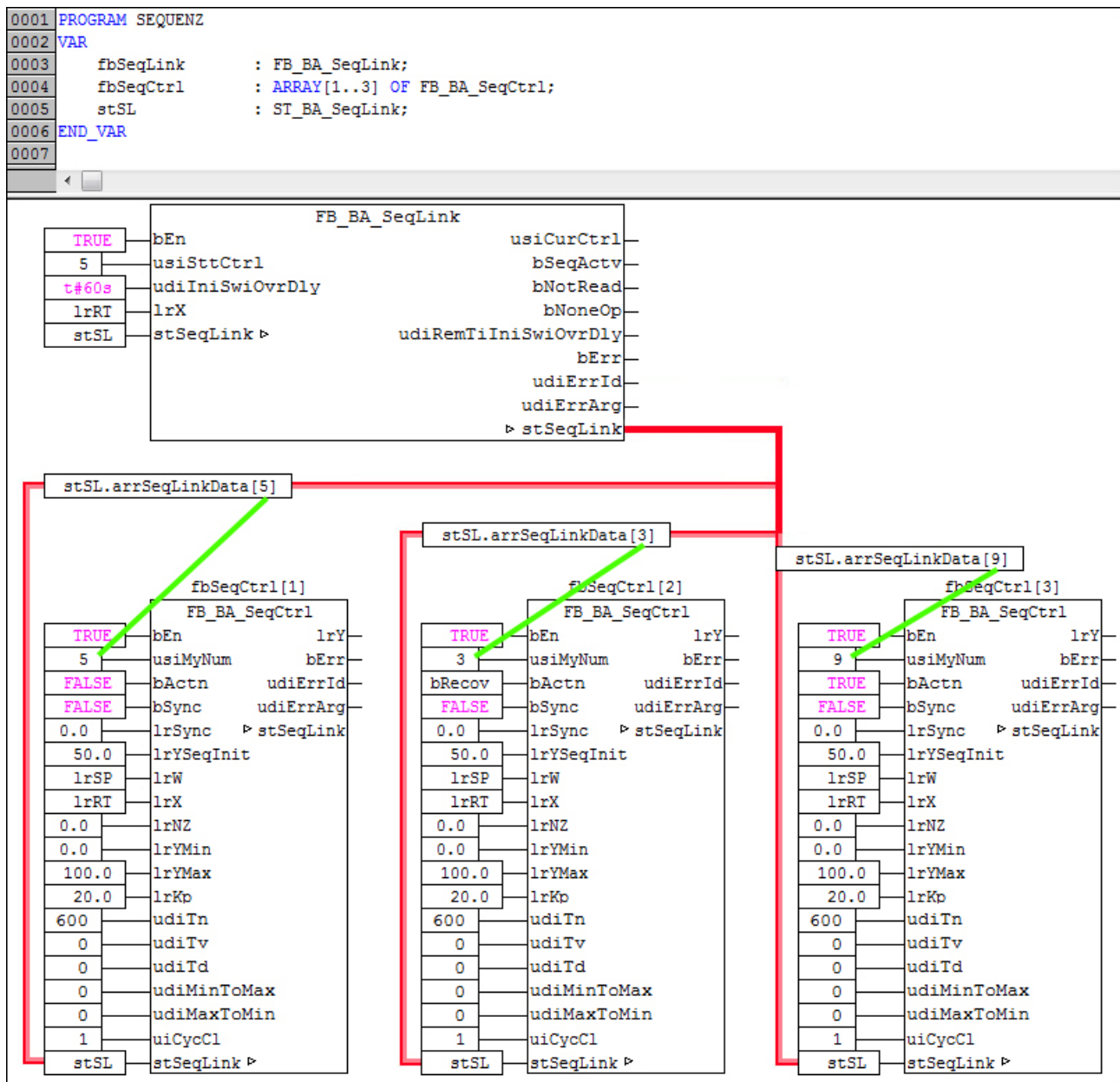
8.3.9 ST_BA_SeqLink / ST_BA_SeqLinkData

Struktur des Daten- und Befehlsaustausches zwischen dem Steuerbaustein [FB_BA_SeqLink](#) [▶ 171] und den Sequenzreglern [FB_BA_SeqCtrl](#) [▶ 167].

Diese Struktur ist pro Sequenzregelung einmal anzulegen:

```
stSeqLink : ST_BA_SeqLink;
```

Innerhalb dieser Struktur ist automatisch eine weitere Feldstruktur deklariert, über die der Sequenz-Link-Baustein auf der einen Seite und die einzelnen Sequenzregler auf der anderen alle relevanten Daten miteinander austauschen. Jeder Sequenzregler schreibt dabei seine Daten in das seiner Ordnungszahl (Eintrag am Eingang *usiMyNum* am Sequenzreglerbaustein) entsprechende Feldelement. An den Bausteinen angelegt wird immer die komplette Struktur mit allen Feldelementen.



Die Strukturen haben den folgenden Aufbau:

```

TYPE ST_BA_SeqLink :
STRUCT
  arrSeqLinkData : ARRAY[1..cMaxSeqCtrl] OF ST_BA_SeqLinkData;
  bSeqActv      : BOOL;
  usiCurSeq    : USINT;
  byDummy      : ARRAY[1..2] OF BYTE;
END_STRUCT
END_TYPE

```

bSeqActv: Die Sequenzregelung ist freigegeben und aktiv.

usiCurSeq: Vom FB_BA_SeqLink: Vorgabe aktueller Sequenzregler

byDummy: Unbenutzte Füllvariablen zur Herstellung einer 4-Byte-Anordnung. Diese Anordnung ist dann wichtig, wenn diese Struktur Teil eines Datenaustauschs zwischen Controllern unterschiedlicher Architektur (ARM/I486) ist.

```

TYPE ST_BA_SeqLinkData:
STRUCT
  lrY      : LREAL;
  lrYMin   : LREAL;
  lrYMax   : LREAL;
  lrW      : LREAL;
  bActn    : BOOL;
  bOp      : BOOL;

```

```
bPresence      : BOOL;
bErrDouble    : BOOL;
usiCurSeq    : USINT;
byDummy       : ARRAY[1..3] OF BYTE;
END_STRUCT
END_TYPE
```

IrY: Vom FB_BA_SeqCtrl: Übermittlung aktueller Stellwert

IrYMin: Vom FB_BA_SeqCtrl: Übermittlung minimaler Stellwert

IrYMax: Vom FB_BA_SeqCtrl: Übermittlung maximaler Stellwert

IrW: Vom FB_BA_SeqCtrl: Übermittlung aktueller Sollwert

bActn: Vom FB_BA_SeqCtrl: Übermittlung Wirksinn invers (*bActn* = FALSE: Heizbetrieb - *bActn* = TRUE: Kühlbetrieb).

bOp: Vom FB_BA_SeqCtrl:Sequenzregler ist freigegeben, d.h. sein Eingang *bEn* ist auf TRUE gesetzt.

bPresence: Vom FB_BA_SeqCtrl: Prüfbit, s.u.

bErrDouble: Vom FB_BA_SeqCtrl: Fehler beim Prüfen der Nummern: es existieren mindestens 2 Sequenzregler gleicher Ordnungszahl *usiMyNum*.

usiCurSeq: Vom FB_BA_SeqLink: Vorgabe aktueller Sequenzregler

byDummy: Unbenutzte Füllvariablen zur Herstellung einer 4-Byte-Anordnung. Diese Anordnung ist dann wichtig, wenn diese Struktur Teil eines Datenaustauschs zwischen Controllern unterschiedlicher Architektur (ARM/I486) ist.

Bemerkung zum Prüfbit:

Jeder Sequenzcontroller setzt in der für ihn gültigen seiner Struktur das *bPresence*-Flag. Ist dieses jedoch schon gesetzt, so muss *usiMyNum* zwangsläufig doppelt vergeben worden sein, greifen doch zwei Sequenzregler auf dieselbe Struktur zu. Der Sequenz-Link-Baustein setzt alle Prüfbits nach Auswertung wieder zurück, so dass dieser Test zyklisch erfolgt. Dadurch kann ein Fehler automatisch per online-change behoben, bzw. auch neue Sequenzregler hinzugefügt werden

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.10 ST_BA_ShdObj

Listeneintrag eines Verschattungsobjekts

```
TYPE ST_BA_ShdObj :
STRUCT
  lrP1x      : LREAL;
  lrP1y      : LREAL;
  lrP1z      : LREAL;
  lrP2x      : LREAL;
  lrP2y      : LREAL;
  lrP2z      : LREAL;
  lrP3x      : LREAL;
  lrP3y      : LREAL;
  lrP3z      : LREAL;
  lrP4x      : LREAL;
  lrP4y      : LREAL;
  lrP4z      : LREAL;
  lrMx       : LREAL;
  lrMy       : LREAL;
  lrMz       : LREAL;
  lrRads     : LREAL;
  usiBegMth  : USINT;
```

```

    usiEndMth : USINT;
    eType     : E_BA_ShdObjType;
    bVld     : BOOL;
END_STRUCT
END_TYPE

```

IrP1x .. IrP4z: Eckkoordinaten. Nur von Bedeutung, wenn das Element ein Viereck ist.

IrMx .. IrMz: Mittelpunktkoordinaten. Nur von Bedeutung, wenn das Element eine Kugel ist.

IrRads: Radius der Kugel. Nur von Bedeutung, wenn das Element eine Kugel ist.

usiBegMth: Anfang der Verschattungsperiode (Monatszahl)

usiEndMth: Ende der Verschattungsperiode (Monatszahl)

eType: Objekttyp, siehe [E_BA_ShdObjType](#) [► 330]

bVld: Plausibilität der Daten: *bVld*=TRUE: Daten sind plausibel.

Bemerkung zur Verschattungsperiode:

Die Monatseinträge dürfen nicht 0 und größer 12 sein, andernfalls sind alle Kombinationen möglich.

Beispiele:

Beginn=1, Ende=1: Verschattung im Januar.

Beginn=1, Ende=5: Verschattung von Anfang Januar bis Ende Mai.

Beginn=11, Ende=5: Verschattung von Anfang November bis Ende Mai (des folgenden Jahres).

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.11 ST_BA_SpRmT

Raum-Temperatursollwerte

```

TYPE ST_BA_SpRmT :
STRUCT
    lrPrtcHtg : LREAL := 12.0;
    lrEcoHtg  : LREAL := 15.0;
    lrPreCmfHtg : LREAL := 19.0;
    lrCmfHtg  : LREAL := 21.0;
    lrPrtcCol : LREAL := 40.0;
    lrEcoCol  : LREAL := 35.0;
    lrPreCmfCol : LREAL := 28.0;
    lrCmfCol  : LREAL := 24.0;
END_STRUCT
END_TYPE

```

Die Werte in der Struktur sind mit den Voreinstellwerten deklariert.

Die Variablen haben folgende Bedeutung:

lrPrtcHtg: Protection Heating

lrEcoHtg: Economy Heating

lrPreCmfHtg: Pre-Comfort Heating

lrCmfHtg: Comfort Heating

lrPrtcCol: Protection Cooling

lrEcoCol: Economy Cooling

IrPreCmfCol: Pre-Comfort Cooling

IrCmfCol: Comfort Cooling

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.12 ST_BA_Sunbld

Struktur des Jalousie-Positioniertelegramms

```

TYPE ST_BA_SunBld:
STRUCT
    lrPos      : LREAL;
    lrAngl     : LREAL;
    bManUp     : BOOL;
    bManDwn    : BOOL;
    bManMod    : BOOL;
    bActv      : BOOL;
END_STRUCT
END_TYPE
    
```

IrPos: Übergebene Jalousiehöhe [%]

IrAngl: Übergebene Lamellenstellung [°]

bManUp: Handbefehl: Jalousie hoch

bManDwn: Handbefehl: Jalousie herunter

bManMod: TRUE: Der Handbedienmodus ist aktiv. FALSE: Der Automatikmodus ist aktiv.

bActv: Der Absender des Telegramms ist aktiv. Dieses Bit wird nur von der Prioritätssteuerung [FB_BA_SunBldPrioSwi4 \[▶ 295\]](#) bzw. [FB_BA_SunBldPrioSwi8 \[▶ 297\]](#) ausgewertet. Die Sonnenschutzaktoren [FB_BA_SunBldActr \[▶ 288\]](#) und [FB_BA_RolBldActr \[▶ 279\]](#) beachten es nicht.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.13 ST_BA_SunBldScn

Tabelleneintrag einer Jalousie-Szene

```

TYPE ST_BA_SunBldScn:
STRUCT
    lrPos      : LREAL;
    lrAngl     : LREAL;
END_STRUCT
END_TYPE
    
```

IrPos: Jalousiehöhe [%]

IrAngl: Lamellenstellung [°]

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.14 E_BA_CtrlSttupMod

Gibt an, mit welchem Lichtwert die Konstantlichtregelung aufstarten soll.

```

TYPE E_BA_CtrlSttupMod :
(
eBA_SttMaxVal := 0,
eBA_SttMinVal := 1,
eBA_SttDctVal := 2
);
END_TYPE

```

eBA_SttMaxVal: Startet mit dem Maximalwert. Die Regelung startet mit 100% - dies lässt alle angeschlossenen Lichtaktoren auf ihren Maximalwert springen.

eBA_SttMinVal: Startet mit dem Minimalwert. Die Regelung startet mit 1% - dies lässt alle angeschlossenen Lichtaktoren auf ihren Minimalwert (nicht AUS) springen.

eBA_SttDctVal: Startet mit dem Wert, der unter *IrSttDctVal* angegeben ist.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.15 E_BA_OnMod

Beschreibt das Einschaltverhalten über den Befehl *bOn* bzw. über Kurzklick an den Eingängen *bSwi*, *bSwiUp* und *bSwiDwn*.

```

TYPE E_BA_OnMod :
(
eBA_ClMaxVal := 0,
eBA_ClMinVal := 1,
eBA_ClOnVal := 2,
eBA_ClRstVal := 3
);
END_TYPE

```

eBA_ClMaxVal: Schaltet den Lichtaktor auf den Maximalwert.

eBA_ClMinVal: Schaltet den Lichtaktor auf den Minimalwert.

eBA_ClOnVal: Schaltet den Lichtaktor auf den Wert *IrOnVal*.

eBA_ClRstVal: Schaltet den Lichtaktor auf den Wert, den der Schaltaktor vor dem letzten Ausschalten innehatte.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.16 ST_BA_LgtCmd

Befehlstelegramm von den Licht-Schaltfunktionen an die Lichtaktoren

```

TYPE ST_BA_LgtCmd :
STRUCT
    bOff          : BOOL;
    bClMinCtrlVal : BOOL;
    bClMaxCtrlVal : BOOL;
    bClOnVal      : BOOL;
    lrOnVal       : LREAL;
    bClRstVal     : BOOL;
    bSetCtrlValDct : BOOL;
    lrSetCtrlValDct : LREAL;
    bChgMaxCtrlVal : BOOL;
    lrChgMaxCtrlVal : LREAL;
    bChgMinCtrlVal : BOOL;
    lrChgMinCtrlVal : LREAL;
    bDimUp        : BOOL;
    bDimDwn       : BOOL;
    bStepUp       : BOOL;
    bStepDwn      : BOOL;
END_STRUCT
END_TYPE
    
```

- bOff:** Schaltet den Lichtaktor unmittelbar aus.
- bClMinCtrlVal:** Schaltet den Lichtaktor auf den Minimalwert.
- bClMaxCtrlVal:** Schaltet den Lichtaktor auf den Maximalwert.
- bClOnVal / lrOnVal:** Schaltet den Lichtaktor auf den gewählten On-Value.
- bClRstCtrlVal:** Schaltet den Lichtaktor auf den Wert, den der Schaltaktor vor dem letzten Ausschalten innehatte.
- bSetCtrlValDct / lrSetCtrlValDct :** Schaltet den Lichtaktor auf den Wert *lrSetCtrlValDct*.
- bChgMaxCtrlVal / lrChgMaxCtrlVal:** Gibt dem Lichtaktor einen neuen Maximalwert *lrChgMaxCtrlVal* vor.
- bChgMinCtrlVal / lrChgMinCtrlVal:** Gibt dem Lichtaktor einen neuen Maximalwert *lrChgMinCtrlVal* vor.
- bDimUp:** statisches Signal zum Hochdimmen, bei DALI-Lichtaktoren wird hiermit alle 200ms der Befehl "UP" ausgegeben.
- bDimDwn:** statisches Signal zum herunterdimmen, bei DALI-Lichtaktoren wird hiermit alle 200ms der Befehl "DOWN" ausgegeben.
- bStepUp:** Schaltet den Lichtwert um einen Schritt höher. Bei DALI-Lichtaktoren wird hiermit EIN Schritt herauf gesetzt, was dem DALI-Befehl "STEP UP" entspricht. Bei analogen Lichtaktoren entspricht dies einem Erhöhen um (100/254) %.
- bStepDwn:** Schaltet den Lichtwert um einen Schritt herunter. Bei DALI-Lichtaktoren wird hiermit EIN Schritt herunter gesetzt, was dem DALI-Befehl "STEP DOWN" entspricht. Bei analogen Lichtaktoren entspricht dies einem Absenken um (100/254) %

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.3.17 E_BA_DALIMod

Vorgabe der Funktionsweise des DALI-Lichtaktorbausteins

```

TYPE E_BA_DALIMod :
(
eBA_DALIModGrpMst := 0,
eBA_DALIModGrpSlv := 1,
eBA_DALIModGrpSgl := 2,
eBA_DALIModSglDvc := 3
);
END_TYPE

```

eBA_DALIModGrpMst: Master einer Gruppe mit ein oder mehreren Lichtaktoren, welche Slaves (*eBA_DALIModGrpSlv*) darstellen. Der Master schaltet die Gruppenbefehle durch, die Parametrierung über die Lichtaktoreingänge jedoch wird für alle Geräte individuell ausgeführt.

eBA_DALIModGrpSlv: Slave einer Gruppe mit ein oder mehreren Lichtaktoren, welche Slaves (*eBA_DALIModGrpSlv*) darstellen. Der Master schaltet die Gruppenbefehle durch, die Parametrierung über die Lichtaktoreingänge jedoch wird für alle Geräte individuell ausgeführt.

eBA_DALIModGrpSgl: Master einer Gruppe, bei denen die Slaves **nicht** als Bausteine aufgeführt werden. Der Master schaltet die Gruppenbefehle durch, die Parametrierung über die Lichtaktoreingänge wird für alle Geräte als Gruppenbefehl des Masters ausgeführt.

eBA_DALIModSglDvc: Ein Lichtaktorbaustein schaltet ein DALI-Gerät über eine Kurzadresse.

Voraussetzungen

Entwicklungsumgebung	Zielsystem	erforderliche Bibliothek	erforderliches Supplement
TwinCAT 2.11 R3/x64	PC/CX	TcBA-Bibliothek ab V1.0.0	TS8040 TwinCAT Building Automation ab V1.0.0

8.4 Fehlercodes

ErrId (hex) Warnung	ErrId (hex) stoppender Fehler	ErrArg	Beschreibung
0x0	0x0		Kein Fehler
0x40008001	0x80008001	1	Der untere Grenzwert der Stellgröße <i>IrMinOut</i> ist größer als der obere Grenzwert <i>IrMaxOut</i>
		2	Der Wert der neutralen Zone (<i>IrNz</i>) darf nicht kleiner als 0.0 sein
		3	Der Wert der Proportionalitätskonstanten (<i>IrKp</i>) darf nicht kleiner als 0.0 sein
0x40008002	0x80008002	1	Eine Reglerordnungszahl <i>usiMyNum</i> ist doppelt vergeben
		2	Eine Reglerordnungszahl <i>usiMyNum</i> ist größer als die maximal erlaubte Anzahl <i>gBA_cMaxSeqCtrl</i> .
		3	Der untere Grenzwert der Stellgröße <i>IrMinOut</i> ist größer als der obere Grenzwert <i>IrMaxOut</i> .
		4	Der Wert der neutralen Zone (<i>IrNz</i>) darf nicht kleiner als 0.0 sein.
		5	Der Wert der Proportionalitätskonstanten (<i>IrKp</i>) darf nicht kleiner als 0.0 sein.
0x40008003	0x80008003	6	Die Reglerordnungszahl <i>usiMyNum</i> eines freigegebenen Reglers ist 0. Das ist nur für nicht verwendete und damit nicht freigegebene Regler erlaubt.
		1	Dem SequenzLink wurde mitgeteilt, dass eine Reglerordnungszahl <i>usiMyNum</i> ist doppelt vergeben worden ist.
		2	Doppelter Wirksinn-Wechsel in der Reglersequenz..

ErrId (hex) Warnung	ErrId (hex) stoppender Fehler	ErrArg	Beschreibung
		3	In der Reglersequenz hat ein Regler höherer Ordnungszahl einen geringeren Sollwert als sein "Vorgänger". Es erfolgt keine Korrektur - die Regesequenz läuft mit den eingegebenen Parametern.
		4	Der Sequenzregler, der als Startregler definiert ist (<i>usiSttCtrl</i>) ist gar nicht parametrierbar, d.h. vorhanden. Es wird der Regler mit der geringsten Ordnungszahl als Startregler verwendet.
		5	Die Ordnungszahl des Start-Reglers ist höher als die maximal erlaubte Anzahl an Reglern oder Null. Es wird der Regler mit der geringsten Ordnungszahl als Startregler verwendet.
		6	Der Sequenzregler, der als Startregler definiert ist (<i>usiSttCtrl</i>) ist nicht frei gegeben, d.h. vorhanden. Es wird der Regler mit der geringsten Ordnungszahl als Startregler verwendet.
0x40008004	0x80008004	1	Der untere Grenzwert der Stellgröße <i>IrMinOut</i> ist größer als der obere Grenzwert <i>IrMaxOut</i>
0x40008005	0x80008005	1	Das minimale Eingangssignal <i>IrMinIn</i> ist größer als das maximale Eingangssignal <i>IrMaxIn</i>
0x40008010	0x80008010	1	Wrong Mapping, das Mapping zum BACnet Device ist fehlerhaft und muss überprüft werden
		2	Operational, das BACnet Device ist nicht betriebsfähig
		3	ADS Verbindung gestört
		4	kein gültiger Ams Port, muss größer 1000 sein
0x40008011	0x80008011	1	Falsche Eingabe Priorität A, PrioA darf nicht <1 oder =6 oder >16 sein
		2	Falsche Eingabe Priorität A, PrioA darf nicht <1 oder >16 sein
		3	Falsche Eingabe Priorität B, PrioB darf nicht <1 oder =6 oder >16 sein
		4	Falsche Eingabe Priorität B, PrioB darf nicht <1 oder >16 sein
		5	Falsche Eingabe Priorität C, PrioC darf nicht <1 oder =6 oder >16 sein
		6	Falsche Eingabe Priorität C, PrioC darf nicht <1 oder >16 sein
		7	Falsche Eingabe Priorität D, PrioD darf nicht <1 oder =6 oder >16 sein
		8	Falsche Eingabe Priorität D, PrioD darf nicht <1 oder >16 sein
		9	Falsche Eingabe Priorität Array, arrPrio[1] oder arrPrio[2] oder arrPrio[3] darf nicht > 255 sein
0x40008013	0x80008013	ADS-Nr	ADS-Fehler FB_BACnet_WriteProp. Der Baustein wird weiter abgearbeitet.
0x40008014	0x80008014	ADS-Nr	ADS-Fehler FB_BACnet_ReadProp. Der Baustein wird weiter abgearbeitet.
0x40008015	0x80008015	1	Die Struktur ST_BA_ComnMsg ist überfüllt. Es sind zu viele Einträge von den einzelnen BACnet Objekten in die Struktur Common Message getätigt worden. Abhilfe durch neuen FB_BA_ComnMsg.
		2	Die Struktur ST_BA_ComnMsgTermt ist überfüllt. Es sind zu viele Einträge von den Auswertebausteinen FB_BA_ComnMsg in die Struktur Common Message Terminated getätigt worden. Abhilfe durch neuen FB_BA_ComnMsgTermt und globaler Struktur
0x40008016	0x80008016	1	Falsche Eingabe PresentValue Priorität A, PrioA darf nicht < 1 OR > NumberOfState sein
		2	Falsche Eingabe PresentValue Priorität B, PrioB darf nicht < 1 OR > NumberOfState sein

ErrId (hex) Warnung	ErrId (hex) stoppender Fehler	ErrArg	Beschreibung
		3	Falsche Eingabe PresentValue Priorität C, PrioC darf nicht < 1 OR > NumberOfState sein
		4	Falsche Eingabe PresentValue Priorität D, PrioD darf nicht < 1 OR > NumberOfState sein
0x40008017	0x80008017	Lokalität	Zeigt die Id des FB_BA_ComnMsg-Funktionsbaustein an wo der Fehler anliegt
0x40008018	0x80008018	1	Falsche Eingabe Relinquish Default, darf nicht < 1 oder > NumberOfStates sein
		2	Falsche Eingabe HighLimit/LowLimit, HighLimit ist kleiner = LowLimit
		3	FB_BACSchrcr: Falsche Angabe <i>usiNumOfPnt</i>
0x40008019	0x80008019	Lokalität	Zeigt das Feld im Array an wo der Fehler zu suchen ist. Die Einträge in das Array arrX müssen so aussehen: arr[1] > arrX[2] > arrX[n] ODER arr[1] < arrX[2] < arrX[n]
0x40008020	0x80008020	1	Welches Objekt einen Fehler hat, wird mit den beiden Ausgängen dwObjId / eObjType angezeigt
0x40008021	0x80008021	1	lr01 überschreitet den Bereich von LREAL
		2	Addition lr01/lr02 überschreitet den Bereich von LREAL
		3	Addition lr01/lr02/lr03 überschreitet den Bereich von LREAL
		4	Addition lr01/lr02/lr03/lr04 überschreitet den Bereich von LREAL
		5	Addition lr01/lr02/lr03/lr04/lr05 überschreitet den Bereich von LREAL
		6	Addition lr01/lr02/lr03/lr04/lr05/lr06 überschreitet den Bereich von LREAL
		7	Addition lr01/lr02/lr03/lr04/lr05/lr06/lr07 überschreitet den Bereich von LREAL
		8	Addition lr01/lr02/lr03/lr04/lr05/lr06/lr07/lr08 überschreitet den Bereich von LREAL
		9	Differenz MAX zu MIN überschreitet den Bereich von LREAL
0x40008027	0x80008027	Lokalität	udiErrArg Zeigt die Variable lrX0x an wo der Fehler zu suchen ist. Folgende Reihenfolge muss eingehalten werden: lrX01 > lrX02 > lrXn ODER lrX01 < lrX02 < lrXn
0x40008028	0x80008028	1	Falsche Eingabe PresentValue, darf nicht <1 oder >udiNumOfSta (NumberOfStates) sein
0x4000802F	0x8000802F	1	Setpoint Heating ist größer oder gleich Setpoint Cooling
0x40008030	0x80008030	1	Falscher Parameter <i>udiNum</i> > <i>udiNumOfEn</i>
0x40008030	0x80008030	1	Fehler intern MEMSET
		2	Fehler intern <i>tTaskCycleTime</i>
		3	Fehler intern <i>tCtrlCycleTime</i>
		4	Fehler intern <i>InvalidParam_tTi</i>
		5	Fehler intern <i>InvalidParam_fBaseTime</i>
		6	Fehler intern Error_MEMCPY
0x40008033	0x80008033	1	Falscher Parameter <i>udiNumOfStep</i>
0x40008034	0x80008034	ADS-Nr	ADS-Fehler beim Lesen der Zeit
0x40008035	0x80008035	ADS-Nr	ADS-Fehler beim Setzen der Zeit
0x40008036	0x80008036	1	Fehler Bereichsüberschreitung Jahr
		2	Fehler Bereichsüberschreitung Monat
		3	Fehler Bereichsüberschreitung Monatstag
		4	Fehler Bereichsüberschreitung Stunde

ErrId (hex) Warnung	ErrId (hex) stoppender Fehler	ErrArg	Beschreibung
		5	Fehler Bereichsüberschreitung Minute
		6	Fehler Bereichsüberschreitung Sekunde
		7	Fehler Bereichsüberschreitung Millisekunde
0x40008100	0x80008100	1	Die x-Werte (Elevationswerte) in der Tabelle sind entweder nicht in aufsteigender Reihenfolge oder doppelt vorhanden.
		2	Ein eingetragener Elevationswert ist außerhalb des gültigen Bereiches von 0°..90°.
		3	Ein eingetragener Positionswert ist außerhalb des gültigen Bereiches von 0%..100%.
0x40008101	0x80008101	1	Der Einschalthysteresewert <i>IrOnVal</i> ist kleiner oder gleich dem Ausschalthysteresewert <i>IrOffVal</i> .
		2	Der Einschalthysteresewert <i>IrOnVal</i> oder der Ausschalthysteresewert <i>IrOffVal</i> ist kleiner Null.
0x40008102	0x80008102	1	Parameterfehler: <i>IrSttRng=IrEndRng</i> .
		2	Parameterfehler: <i>IrSttRng</i> bzw. <i>rEndRng</i> kleiner 0° oder größer 360°.
		3	Parameterfehler: Die Differenz von <i>IrSttRng</i> und <i>IrEndRng</i> ist größer als 180Grad. Für die Betrachtung des Sonneneinfalles auf eine Fassade ist dieser Bereich zu groß gewählt.
0x40008103	0x80008103	1	Parameterfehler: <i>IrHiLmt</i> kleiner oder gleich <i>rLoLmt</i> .
		2	Parameterfehler: <i>IrLoLmt</i> ist kleiner 0° oder <i>IrHiLmt</i> ist größer 90°.
0x40008104	0x80008104	1	Die Gesamt-Hoch- oder Herunterfahrzeit (<i>udiTiUp/udiTiDwn</i>) ist Null.
0x40008105	0x80008105	1	Parameterfehler: Rolladenhöhe ist kleiner oder gleich 0.
		2	Parameterfehler: Hoch/Herunter-Timer = 0.
		3	Parameterfehler: Turning/Wendetimer = 0.
		4	Parameterfehler: Lamellenwinkelgrenzen: Die obere Grenze ist kleiner oder gleich der Unteren (<i>IrAnglLmtUp<=IrAnglLmtDwn</i>).
		5	Einer der Hysterese-Offset-Werte für das Nachpositionieren, <i>IrHysIncPos</i> oder <i>IrHysIncAngl</i> , ist kleiner Null
0x40008106	0x80008106	1	Die eingetragene Szenen-Nummer <i>udiSlcdScn</i> ist höher als die maximal zulässige Anzahl an Szenen.
		2	Der Positionssollwert <i>IrSpPos</i> ist größer als 100 oder kleiner als 0.
0x40008107	0x80008107	1	Die über die Struktur <i>stRmTSp</i> übertragenen Sollwerte sind nicht in aufsteigender Reihenfolge. Richtig: <i>stRmTSp.rPreCmfHtg <= stRmTSp.rCmfHtg <= stRmTSp.rCmfCol <= stRmTSp.rPreCmfCol</i> .
		2	Der Einschalthysteresewert <i>IrBrtnsActvVal</i> ist kleiner oder gleich dem Ausschalthysteresewert <i>IrBrtnsDctvVal</i> .
		3	Die Jalousieposition im Heizfall <i>IrPosHtg</i> ist größer als die im Kühlfall <i>IrPosCol</i> . Dies macht keinen Sinn.
		4	Einer der Schaltwerte, <i>IrBrtnsActvVal</i> oder <i>IrBrtnsDctvVal</i> , ist kleiner als 0.
		5	Einer der Positionssollwerte, <i>IrPosHtg</i> oder <i>IrPosCol</i> , ist größer als 100 oder kleiner als 0.
0x40008108	0x80008108	1	Der Einschalthysteresewert <i>IrActvVal</i> ist größer oder gleich dem Ausschalthysteresewert <i>IrDctvVal</i> .
		2	Einer der Schaltwerte, <i>IrActvVal</i> oder <i>IrDctvVal</i> , ist kleiner als 0.
		3	Der Positionssollwert <i>IrPosTwiLgt</i> ist größer als 100 oder kleiner als 0.

ErrId (hex) Warnung	ErrId (hex) stoppender Fehler	ErrArg	Beschreibung
0x40008109	0x80008109	1	Die eingetragene Frosttemperatur liegt über 10°C.
		2	Die eingetragene Windgeschwindigkeit zum Ausschalten des Alarmes ist größer oder gleich der zum Einschalten.
		3	Die eingetragene Windgeschwindigkeit zum Ein- bzw. zum Ausschalten des Alarmes ist kleiner als 0.
		4	Der Positionssollwert <i>IrPosProt</i> ist größer als 100 oder kleiner als 0.
0x4000810A	0x8000810A	1	Die Dauer des Positionierintervalls ist gleich Null oder überschreitet 720 min.
		2	Der eingetragene Längengrad ist nicht im gültigen Bereich von -180°..180°.
		3	Der eingetragene Breitengrad ist nicht im gültigen Bereich von -90°..90°.
		4	Der eingetragene Wert für die Fassadenneigung <i>IrFcdAngl</i> ist außerhalb des gültigen Bereiches von -90°..90°.
		5	Der Wert für den Lamellenabstand (<i>IrLamDstc</i>) ist größer oder gleich dem Wert für die Lamellenbreite (<i>IrLamWdth</i>). Dies stellt keine "gültige" Jalousie dar, da die Lamellen nicht vollständig schließen könnten. Mathematisch gesehen würde dieses zu Fehlern führen.
		6	Der eingetragene Wert für den Lamellenabstand <i>IrLamDstc</i> ist gleich Null.
		7	Der eingetragene Wert für die Lamellenbreite <i>IrLamWdth</i> ist gleich Null.
		8	Der eingetragene Wert für die fixe Jalousiehöhe (<i>IrFixPos</i>) ist größer als 100 oder kleiner als 0. Gleichzeitig ist Positionierung "fixe Jalousiehöhe" angewählt - <i>ePosMod=ePosModFix</i>.
		9	Das Bit "Werte gültig" (<i>bVld</i>) in der Positioniertabelle stBldPosTab [▶ 330] ist nicht gesetzt - ungültige Werte: siehe FB_BA_BldPosEntry [▶ 257] . Gleichzeitig ist Positionierung "Tabelle" angewählt - <i>ePosMod=ePosModTab</i>.
		10	Der eingetragene Wert für die Fensterhöhe <i>IrWdwHght</i> ist kleiner oder gleich Null. Gleichzeitig ist Positionierung "maximaler Lichteinfall" angewählt - <i>ePosMod=ePosModMaxIndc</i>.
11	Der eingetragene Abstand Fensterunterkante zu Boden <i>IrDstcWdwFir</i> ist kleiner Null. Gleichzeitig ist Positionierung "maximaler Lichteinfall" angewählt - <i>ePosMod=ePosModMaxIndc</i>.		
12	Der eingetragene Wert für den maximal gewünschten Lichteinfall <i>IrMaxLgtIndc</i> ist kleiner oder gleich Null. Gleichzeitig ist Positionierung "maximaler Lichteinfall" angewählt - <i>ePosMod=ePosModMaxIndc</i>.		
0x40008120	0x80008120	1	Indexfehler! <i>iColumn</i> und/oder <i>iRow</i> liegen außerhalb der zulässigen Grenzen <i>1..gBA_cMaxColumnFcd</i> bzw. <i>1..gBA_cMaxRowFcd</i> . Siehe Liste der Fassadenelemente [▶ 46] .
		2	Der Gruppenindex ist 0, gleichzeitig ist jedoch ein anderer Eintrag des Fassadenelementes ungleich Null. Nur wenn alle Einträge eines Fassadenelementes Null sind, wird es als gültiges, gewollt ausgelassenes Fassadenelement angesehen, ansonsten jedoch als Fehleingabe interpretiert.
		3	Die X-Komponente ersten Eckpunktes (Corner1) ist kleiner Null.

Errld (hex) Warnung	Errld (hex) stoppender Fehler	ErrArg	Beschreibung
		4	Die Y-Komponente ersten Eckpunktes (Corner1) ist kleiner Null.
		5	Die Fensterbreite ist kleiner oder gleich Null.
		6	Die Fensterhöhe ist kleiner oder gleich Null.
0x40008121	0x80008121	1	Fehler beim Lesen der Daten-Datei: Datei ist zu groß (Anzahl Bytes größer als <i>cMaxDataFileSize</i>)
0x40008122	0x80008122	ADS-Nr	Dateihandling-Fehler: Open Logfile
0x40008123	0x80008123	ADS-Nr	Dateihandling-Fehler: Open Datafile
0x40008124	0x80008124	ADS-Nr	Dateihandling-Fehler: Read Datafile
0x40008125	0x80008125	ADS-Nr	Dateihandling-Fehler: Write Logfile
0x40008126	0x80008126	ADS-Nr	Dateihandling-Fehler: Close Datafile
0x40008127	0x80008127	ADS-Nr	Dateihandling-Fehler: Write Logfile
0x40008128	0x80008128	ADS-Nr	Dateihandling-Fehler: Close Logfile
0x40008129	0x80008129	1	Fehler beim Lesen der Daten-Datei: Datei ist zu groß (Anzahl Bytes größer als <i>cMaxDataFileSize</i>)
0x4000812A	0x8000812A	ADS-Nr	Dateihandling-Fehler: Open Logfile
0x4000812B	0x8000812B	ADS-Nr	Dateihandling-Fehler: Open Datafile
0x4000812C	0x8000812C	ADS-Nr	Dateihandling-Fehler: Read Datafile
0x4000812D	0x8000812D	ADS-Nr	Dateihandling-Fehler: Write Logfile
0x4000812E	0x8000812E	ADS-Nr	Dateihandling-Fehler: Close Datafile
0x4000812F	0x8000812F	ADS-Nr	Dateihandling-Fehler: Write Logfile
0x40008130	0x80008130	ADS-Nr	Dateihandling-Fehler: Close Logfile
0x40008131	0x80008131	1	Der Index der zu betrachtenden Fenstergruppe <i>usiGrpId</i> ist 0.
		2	Ein Element der Fensterliste, <i>arrFcdElem[i,j]</i> , ist nicht gültig. Siehe Ausgänge <i>iErrldx_I</i> und <i>iErrldx_J</i> .
0x40008132	0x80008132	1	Indexfehler! <i>ild</i> liegt außerhalb der zulässigen Grenzen 1..gBA_cMaxShdObj.
		2	Die Summe der Winkel des Vierecks ist nicht 360°. Das bedeutet, dass die Eckpunkte nicht in der Reihenfolge P1, P2, P3 und P4 sondern P1, P3, P2 und P4 sind. Dies ergibt ein über Kreuz geschlagenes Viereck.
		3	Die Eckpunkte des Vierecks liegen nicht auf der selben Ebene.
		4	Die z-Komponente von P1 ist kleiner Null. Damit läge dieser Eckpunkt hinter der Fassade.
		5	Die z-Komponente von P3 ist kleiner Null. Damit läge dieser Eckpunkt hinter der Fassade.
		6	P1 ist gleich P2. Damit ist das eingetragene Objekt kein Viereck.
		7	P1 ist gleich P3. Damit ist das eingetragene Objekt kein Viereck.
		8	P1 ist gleich P4. Damit ist das eingetragene Objekt kein Viereck.
		9	P2 ist gleich P3. Damit ist das eingetragene Objekt kein Viereck.
		10	P2 ist gleich P4. Damit ist das eingetragene Objekt kein Viereck.
		11	P3 ist gleich P4. Damit ist das eingetragene Objekt kein Viereck.
		12	Der eingetragene Radius ist gleich Null.

ErrId (hex) Warnung	ErrId (hex) stoppender Fehler	ErrArg	Beschreibung
		13	Die z-Komponente des Kugelmittelpunktes ist kleiner Null. Damit läge dieser Punkt hinter der Fassade.
		14	Fehler Objekttyp eType - weder Viereck noch Kugel.
		15	Fehler Monateingabe.
0x40008200	0x80008200	1	Parameter " <i>IrMinCtrlVal</i> " (minimaler Lichtausgabewert) unterhalb 1.0. Der Wert wurde automatisch auf 1.0 korrigiert.
		2	Parameter " <i>IrMinCtrlVal</i> " (minimaler Lichtausgabewert) oberhalb 100.0. Der Wert wurde automatisch auf 100.0 korrigiert.
		3	Parameter " <i>IrMaxCtrlVal</i> " (maximaler Lichtausgabewert) unterhalb 1.0. Der Wert wurde automatisch auf 1.0 korrigiert.
		4	Parameter " <i>IrMinCtrlVal</i> " (maximaler Lichtausgabewert) oberhalb 100.0. Der Wert wurde automatisch auf 100.0 korrigiert.
		5	Parameter " <i>IrMaxCtrlVal</i> " (maximaler Lichtausgabewert) unterhalb " <i>IrMinCtrlVal</i> " (maximaler Lichtausgabewert). Der Wert wurde automatisch auf " <i>IrMinCtrlVal</i> " korrigiert.
0x40008201	0x80008201	DALI- ErrId	Trägt das direkt herausgeführte Fehlerargument aus der Errorcode-Liste der Bibliothek DALIV2, siehe TcPlcLibDALIV2_Errorcodes
0x40008202	0x80008202	1	Ungültige Kurzadresse, oberhalb 63.
		2	Ungültige Gruppenadresse, oberhalb 15.

9 TwinCAT BA PLC Templates

Einleitung

Die TwinCAT BA PLC Templates bestehen aus fertigen TwinCAT Programmbausteinen für Sensoren und Aktoren, für komplette Baugruppen, sowie für Anlagenteile oder ganze Anlagen der Heizungs-, Klima- und Lüftungstechnik und der Raumautomation. Im Gegensatz zu einer regulären SPS-Bibliothek, werden TwinCAT BA PLC Templates als Programmbausteine in die SPS-Programme importiert. Notwendige Anpassungen kann der Systemintegrator bei Bedarf selber durchführen. Darüber hinaus besteht die Möglichkeit, eigene TwinCAT BA PLC Templates zu erstellen. Besondere Werkzeuge sind hierfür nicht notwendig.

General

Name	Beschreibung
BAC_Gen_01 [▶ 354]	Aufruftemplate, welches die Basisfunktion eines BACnet Controllers liefert.
BAC_GenAlm_01 [▶ 356]	Sammlung sämtlicher Alarme eines Controllers
BAC_GenComnMsg_01 [▶ 360]	Sammlung sämtlicher Ereignismeldungen eines Controllers
BAC_GenDvc_01 [▶ 363]	Template, welches das BACnet Device instanziiert. Für eine BACnet-Analge ist dieses Template zwingend erforderlich und nur einmal zu platzieren.
BAC_GenNC_01 [▶ 368]	Das Template stellt für das Absetzen der objektintegrierten Meldungen (intrinsic reporting) acht BACnet-Meldeklassen (Notification Class) bereit.
BAC_GenSys_01 [▶ 370]	Template Systemfunktionen, z. B. Blinksequenz für Störmeldeanzeigen und Bereitstellung der Systemzeit.
BAC_GenWthT_01 [▶ 371]	Außentemperatur und gemittelte Außentemperatur

Anlagenaufruftemplates

Name	Beschreibung
BAC_AC_KennzeichnungssystemAnlagenschlüssel [▶ 618]	Kennzeichnungssystem Anlagentemplates Lüftungs- und Klimatechnik
BAC_AC_SE_3_4_1_1_1_0 [▶ 619]	Aufruftemplate Lüftungsanlagen mit Zu- und Abluftventilator und thermischer Luftbehandlung: Ab-/Zuluftkaskade einfach - Ventilator Druck - Vorerhitzer - Kühler - ERG Platte
BAC_AC_SE_4_4_1_1_0_1 [▶ 625]	Aufruftemplate Lüftungsanlagen mit Zu- und Abluftventilator und thermischer Luftbehandlung: Ab-/Zuluftkaskade - Fan Druck - Vorerhitzer - Kühler - Mischluft
BAC_AC_SE_4_4_1_1_3_0 [▶ 631]	Aufruftemplate Lüftungsanlagen mit Zu- und Abluftventilator und thermischer Luftbehandlung: Ab-/Zuluftkaskade - Ventilator Druck - Vorerhitzer - Kühler - ERG Rotation
BAC_AC_Sx_001 [▶ 638]	Anlagengrundprogramm für Lüftungsanlagen mit thermischer Luftbehandlung

Basisprogramme einer Anlage

Name	Beschreibung
BAC_PltAlm_01 [▶ 372]	Sammlung und Quittierung aller Alarme einer Anlage

Name	Beschreibung
BAC_PltComnMsg_01 [▶ 376]	Sammlung sämtlicher Ereignismeldungen einer Anlage

Heizen

Aggregat	Name	Beschreibung
Heizkreis	BAC_H_HtgCir_01 [▶ 551]	Aufruftemplate eines Witterungstemperatur geführten Heizkreises
	BAC_H_HtgCirSp_01 [▶ 553]	Das Template ist eine Witterungstemperatur geführte Heizkreissteuerung. Es beinhaltet eine Heizkurve, einen Heizgrenzschalter und die Betriebsarten Frost / Nacht / Tag / Auto mit zugehörigem Sollwert.
Warmwasser	BAC_DHW_01 [▶ 556]	Aufruftemplate Regelung und Ladung eines Warmwasserspeichers
	BAC_DHW_Ctrl_01 [▶ 557]	Das Template steuert die Ladung eines Warmwasserspeichers.
	BAC_HW_LglPrev_01 [▶ 560]	Das Template ist eine Schutzfunktion zur Verhinderung der Legionellen-Bildung im Warmwasser.

Klimaanlage

Aggregat	Name	Beschreibung
Befeuchtung	BAC_AC_Humf_01 [▶ 380]	Aufruftemplate für die Ansteuerung eines Dampfbefeuchters
	BAC_AC_Humf_PID_01 [▶ 383]	Sequenzregler Befeuchten
	BAC_AC_SteamGenerator_01_xx [▶ 388]	Dampferzeuger/Dampfgenerator
Brandschutzklappen	BAC_AC_FireDmp_01_xx [▶ 392]	Aufruftemplate Ansteuerung und Überwachung einer motorischen Brandschutzklappe mit binären Schaltausgang
Energierückgewinnung	BAC_AC_ErcPI_01 [▶ 398]	Aufruftemplate für die Ansteuerung eines Plattenwärmetauschers
	BAC_AC_ErcPI_02 [▶ 402]	Aufruftemplate für die Ansteuerung eines Plattenwärmetauschers mit Bypassklappe
	BAC_AC_ErcRecup_01 [▶ 407]	Aufruftemplate für die Ansteuerung eines Kreislaufverbundsystem
	BAC_AC_ErcRot_01 [▶ 412]	Aufruftemplate für die Ansteuerung eines Rotationswärmetauschers mit Bypassklappe
	BAC_AC_ErcRot_02 [▶ 417]	Aufruftemplate für die Ansteuerung eines Rotationswärmetauschers
	BAC_AC_ErcT_PID_01 [▶ 420]	Sequenzregler Energierückgewinnung Temperatur
	BAC_ErcIcPrt_01 [▶ 426]	Das Template dient zum Schutz vor Vereisung der Energierückgewinnungs-Aggregate über einen Differenzdruckschalter

Aggregat	Name	Beschreibung
	BAC ErclcPrt 02 [▶ 428]	Das Template dient zum Schutz vor Vereisung der Energierückgewinnungs-Aggregate über einen Differenzdruckfühler
Filter	BAC AC Filter 01 [▶ 444]	Das Template realisiert eine Filterüberwachung innerhalb einer raumlufttechnischen Anlagen mittels eines Differenzdruckschalters.
Klappen	BAC AC ExhADmp2P 01 xx [▶ 446]	Aufruftemplate für die Ansteuerung und Überwachung einer Fortluftklappe mit einem Federrücklaufantrieb und Endlagenkontrolle
	BAC AC OuADmp2P 01 xx [▶ 454]	Aufruftemplate für die Ansteuerung und Überwachung einer Außenluftklappe mit einem Federrücklaufantrieb und Endlagenkontrolle
Kühler	BAC AC CoIT 01 [▶ 465]	Aufruftemplate eines temperaturgesteuerten Kaltwasser-Luftkühlers (Ventil, Pumpe)
	BAC AC CoIT 02 [▶ 468]	Aufruftemplate eines temperaturgesteuerten Kaltwasser-Luftkühlers (Ventil)
	BAC AC CoIT PID 01 [▶ 471]	Sequenzregler für einen Kühler Temperatur
	BAC AC CoITH 01 [▶ 476]	Aufruftemplate eines temperatur- und feuchtegesteuerten Kaltwasser-Luftkühlers mit Ventil und Pumpe
	BAC AC CoITH 02 [▶ 480]	Aufruftemplate eines temperatur- und feuchtegesteuerten Kaltwasser-Luftkühlers mit Ventil
	BAC AC CoIH PID 01 [▶ 460]	Sequenzregler für einen Kühler Entfeuchtung
Mischluft	BAC AC MixAT 01 [▶ 483]	Aufruftemplate Mischluftklappen temperaturgesteuert für die Ansteuerung von drei stetigen Klappen
	BAC AC MixAT PID 01 [▶ 488]	Sequenzregler Mischluftklappensteuerung
Nacherhitzer	BAC AC ReHtr 01 [▶ 493]	Aufruftemplate eines Nacherhitzers
	BAC AC ReHtr PID 01 [▶ 496]	Sequenzregler Nacherhitzer
Ventilatoren	BAC AC ExtAFan FC 01 [▶ 430]	Aufruftemplate mit integriertem Druckregler für einen Drehzahl gesteuerten Abluftventilator
	BAC AC ExtAFan1st 01 [▶ 432]	Aufruftemplate für einen einstufigen Abluftventilator
	BAC AC SuAFan FC 01 [▶ 435]	Aufruftemplate mit integriertem Druckregler für einen Drehzahl gesteuerten Zuluftventilator
	BAC AC SuAFan1st 01 [▶ 438]	Aufruftemplate für einen einstufigen Zuluftventilator
	BAC DiffPrssMonit 01 [▶ 441]	Differenzdrucküberwachung mittels eines Differenzdruckwächters

Aggregat	Name	Beschreibung
	BAC_DiffPrssMonit_02 [▶ 443]	Differenzdrucküberwachung mittels eines Differenzdruckfühlers
Vorerhitzer	BAC_AC_PreHtr_01 [▶ 501]	Aufruftemplate eines Warmwasserluftherhitzer
	BAC_AC_PreHtr_PID_01 [▶ 506]	Sequenzregler Vorerhitzer
	BAC_AC_RetWtrCtrl_01 [▶ 511]	Aufruftemplate Rücklauftemperaturregelung des Vorerhitzers
	BAC_FrstPrt_01 [▶ 515]	Frostschutzüberwachung Vorerhitzer
Volumenstromregler	BAC_AC_VAV_01_xx [▶ 517]	Aufruftemplate Ansteuerung eines Volumenstromreglers mittels analogen Ausgang

Setpoint	Beschreibung
BAC_AC_CasCtrlH_01 [▶ 642]	Kaskadenregler für die Zuluftfeuchte bestehend aus einem Führungsregler zur Sollwertberechnung für das Be- und Entfeuchten
BAC_AC_CasCtrlH_02 [▶ 645]	Kaskadenregler für die Zuluftfeuchte bestehend aus zwei Führungsregler für eine Raum- bzw. Abluft/ Zuluft-Kaskadenfeuchteregeung.
BAC_AC_CasCtrlT_01 [▶ 649]	Kaskadenregler Zulufttemperatur bestehend aus einem Führungsregler zur Sollwertberechnung für Heizen, Kühlen und Rückgewinnung
BAC_AC_CasCtrlT_02 [▶ 653]	Kaskadenregler Zulufttemperatur bestehend aus zwei Führungsreglern zur Sollwertberechnung für Heizen, Kühlen und Rückgewinnung
BAC_AC_SpRmTH_01 [▶ 663]	Sollwertprogramm für eine Abluft/Zuluftkaskade mit jeweils einen Sollwert für die Relative und absolute Feuchte und einem Raumtemperatursollwert inklusive Sommer und Winterkompensation
BAC_AC_SpRmTH_02 [▶ 666]	Sollwertprogramm für eine Abluft/Zuluftkaskade mit zwei Sollwerten für die Relative und absolute Feuchte und einem Raumtemperatursollwert inklusive Sommer und Winterkompensation
BAC_AC_SpRmTH_03 [▶ 670]	Sollwertprogramm für eine Abluft/Zuluftkaskade mit jeweils einen Sollwert für die Relative und absolute Feuchte und zwei Raumtemperatursollwerte inklusive Sommer und Winterkompensation
BAC_AC_SpRmTH_04 [▶ 674]	Sollwertprogramm für eine Abluft/Zuluftkaskade mit jeweils zwei Sollwerten für die Relative und absolute Feuchte und zwei Raumtemperatursollwerte inklusive Sommer und Winterkompensation
BAC_AC_SpRmT_01 [▶ 658]	Sollwertprogramm für eine Abluft/Zuluftkaskade mit nur einem Raumtemperatursollwert inklusive Sommer und Winterkompensation
BAC_AC_SpRmT_02 [▶ 661]	Sollwertprogramm für eine Abluft/Zuluftkaskade mit jeweils einem Raumtemperatursollwert für den Heiz- und Kühlbetrieb inklusive Sommer und Winterkompensation
BAC_AC_SpSuAT_01 [▶ 678]	Sollwertprogramm für eine Zulufttemperatur-Regelung mit einem Zulufttemperatur Sollwert inklusive Sommer und Winterkompensation über eine Kennlinie.

Setpoint	Beschreibung
BAC AC SpSuAT 02 [▶ 681]	Sollwertprogramm für eine Zulufttemperatur-Regelung mit separaten Zuluftsollwerten für das Heizen, Kühlen und die Energierückgewinnung inklusive zwei separate Sollwertkennlinien mit Sommer- und Winterkompensation
General	Beschreibung
BAC AC OpMod 01 [▶ 524]	Betriebsart einer Lüftungsanlage
BAC AC SeqH 01 [▶ 528]	Starten und Steuern der Zuluftsequenzregelung Feuchte einer RLT-Anlage
BAC AC SeqT 01 [▶ 532]	Starten und Steuern der Zuluftsequenzregelung Temperatur einer RLT-Anlage
BAC AC StartT 01 [▶ 538]	Schrittweises Anfahren eine RLT-Anlage ohne Be- und Entfeuchtung.
BAC AC StartTH 01 [▶ 543]	Schrittweises Anfahren eine RLT-Anlage mit Be- und Entfeuchtung.
BAC AC SumNgtCol 01 [▶ 549]	Sommernachtkühlung

Universal

	Name	Beschreibung
Frequenzumformer	BAC Uni FC 01_xx [▶ 563]	Ansteuerung eines Frequenzumrichters
Klappe	BAC Uni Dmp 01_xx [▶ 568]	Ansteuerung eines stetigen Klappenantriebs
	BAC Uni Dmp2P 01_xx [▶ 572]	Ansteuerung einer Zweipunktklappe
Motor	BAC Uni Mot1st 01_xx [▶ 576]	Ansteuerung eines einstufigen Motors
Pumpe	BAC Uni Pu1st 01_xx [▶ 579]	Ansteuerung einer einstufigen Pumpe
Rauchmelder	BAC Uni SmokeDetc 001 [▶ 586]	Ansteuerung und Überwachung eines Kanalrauchmelders
Steuerung	BAC Cont4Stp 01 [▶ 588]	Stufenschalter mit 4 Stufen um mehrstufige Aggregate zu steuern
	BAC HX 01 [▶ 592]	Berechnung der Taupunkttemperatur, Enthalpie, Feuchtkugeltemperatur und der absoluten Feuchte
	BAC Hys 01 [▶ 594]	Hysterese-Funktion mit festen Schaltpunkten
	BAC Hys 02 [▶ 595]	Hysterese-Funktion mit dynamischen Schaltpunkten
	BAC PID 01 [▶ 597]	Universeller PID-Regler
	BAC PID 02 [▶ 599]	Universeller PID-Regler
	BAC PID 03 [▶ 601]	Universeller PID-Regler ohne Referenzierungsobjekte
	BAC Ramp 01 [▶ 602]	Fallende Rampenbegrenzung
	BAC Ramp 02 [▶ 603]	Steigende Rampenbegrenzung
	BAC Scale 02 [▶ 605]	Linear-Interpolation mit 2 Stützstellen
BAC Scale 04 [▶ 606]	Linear-Interpolation mit 4 Stützstellen	

	Name	Beschreibung
	BAC Scale 07 [▶ 607]	Linear-Interpolation mit 7 Stützstellen
Ventil	BAC Uni Vlv 01_xx [▶ 608]	Ansteuerung eines stetigen Regelventils
	BAC Uni Vlv3P 01_xx [▶ 612]	Ansteuerung eines Dreipunktventils

BACnet Objekte

Name	Beschreibung
BAC AI 01 [▶ 683]	Analog Input Objekt mit Alarmaufnahme
BAC AI 02 [▶ 685]	Analog Input Objekt
BAC AI Enthalpy 01 [▶ 687]	Das Template erfasst die beiden analogen Eingangswerte Temperatur und Relative Feuchte . Mit dem Funktionsbaustein HX werden die Taupunkttemperatur , die spezifische Enthalpie , Feuchtkugeltemperatur und die Absolute Feuchte berechnet.
BAC AO 01 [▶ 691]	Analog Ausgangs Objekt mit manuellen Handeingriff mittels eines digitalen Eingangs LocSwi und Trend Objekt
BAC AO 02 [▶ 693]	Analog Ausgangs Objekt
BAC AO 03 [▶ 695]	Analog Ausgangs Objekt mit manuellen Handeingriff mittels eines digitalen Eingangs LocSwi
BAC AV 01 [▶ 697]	Analog Wert Objekt mit Priorisierung eines REAL-Prozesswertes
BAC AV 02 [▶ 699]	Analog Wert Objekt für die Eingabe eines Sollwertes oder Parameters
BAC AV 03 [▶ 700]	Analog Wert Objekt mit Priorisierung eines REAL-Prozesswertes mit Trend Objekt
BAC AV 04 [▶ 702]	Analog Wert Objekt für die Eingabe eines Sollwertes oder Parameters mit Trend Objekt
BAC BI 01 [▶ 703]	Binäres Eingangsobjekt mit Alarmaufnahme
BAC BI 02 [▶ 704]	Binäres Eingangsobjekt
BAC BI CMD 01 [▶ 706]	Binäres Eingangsobjekt mit Alarmaufnahme und Gruppenauftrags-Objekt für das Schreiben auf spezifizierte Properties anderer Objekte
BAC BO 01 [▶ 707]	Binäres Ausgangsobjekt mit Priorisierung des Schaltwertes und manuellen Handeingriff mittels eines digitalen Eingangs LocSwi
BAC BO 02 [▶ 710]	Binäres Ausgangsobjekt mit Priorisierung des Schaltwertes
BAC BV 01 [▶ 712]	Binäres Werte Objekt mit Priorisierung des Prozesswertes
BAC BV 02 [▶ 713]	Binäres Werte Objekt für die Eingabe eines Sollwertes oder Parameters
BAC CMD 01 [▶ 714]	Gruppenauftrags-Objekt
BAC MV 01 [▶ 715]	Multistate-Wert Objekt mit einfacher Priorisierung
BAC MV 02 [▶ 716]	Multistate-Wert Objekt um einen Prozesswert zu bedienen und zu beobachten
BAC SchedBinPV 01 [▶ 717]	Das Template beinhaltet einen BACnet Scheduler vom Typ Binary Present Value. Zudem enthält das Template die Funktion "Vorausrechnende Ein- und Ausschaltzeit"

Name	Beschreibung
BAC_SchedB_01 [▶ 718]	Template beinhaltet einen BACnet Scheduler vom Ausgabetyt BOOL
BAC_SchedR_01 [▶ 719]	Template beinhaltet einen BACnet Scheduler vom Ausgabetyt REAL
BAC_SchedUdi_01 [▶ 720]	Template beinhaltet einen BACnet Scheduler vom Ausgabetyt UDINT
BAC_TL_01 [▶ 721]	Trendlog-Speicher-Objekt

IO-Templates

Name	Beschreibung
P_KL1501 [▶ 722]	IO-Template zur Parametrierung einer KL1501: 1-Kanal Auf-Abwärtszähler.
P_KL27x1 [▶ 723]	IO-Template zur Parametrierung einer KL2751 / KL2761: 1-Kanal Dimmerklemme.
P_KL320x [▶ 724]	IO-Template zur Parametrierung einer KL3201 oder KL3202: Eingangsklemme für Widerstandssensoren.
P_KL3204 [▶ 726]	IO-Template zur Parametrierung einer KL3204: 4-Kanal Eingangsklemme für Widerstandssensoren.
P_KL3208 [▶ 727]	IO-Template zur Parametrierung einer KL3208-0010: 8-Kanal Eingangsklemme für Widerstandssensoren.
P_KL3228 [▶ 728]	IO-Template zur Parametrierung einer KL3228: 8-Kanal Eingangsklemme für Widerstandssensoren.
P_KL8519 [▶ 729]	IO-Template zur Parametrierung einer KL8519: 16-Kanal-Digital-Eingangs-Meldemodul.
P_KL8524 [▶ 732]	IO-Template zur Parametrierung einer KL8524: 4 x 2-Kanal-Digital-Ausgangsmodul.
P_KL8528 [▶ 734]	IO-Template zur Parametrierung einer KL8528: 8-Kanal-Digital-Ausgangsmodul.
P_KL8548 [▶ 735]	IO-Template zur Parametrierung einer KL8548: 8-Kanal-Analog-Ausgangsmodul 0...10 V.

9.1 Global_Variablen_Alarming

In der SPS sind unter **Resources\GlobalVariables** die folgenden Funktionsbausteine, Variablen und Konstanten zu finden:

VAR_GLOBAL			
gBA_AlmMgnr	FB_BA_AlarmMgnr		Alarmsammler mit Neuwertmeldung//Alarm collector with new value message
END_VAR			
VAR_GLOBAL CONSTANT			
Alarming Priority			
ALM_PRIO_EMPTY	BYTE	0	keine Meldung//no message
ALM_PRIO_NOTE	BYTE	1	diverse Meldungen// various message
ALM_PRIO_WARNING	BYTE	2	Warnung//warning
ALM_PRIO_ALARM	BYTE	3	Alarm ohne Abschaltung// Alarm without shutdown

ALM_PRIO_CRITICAL	BYTE	4	Alarm mit Abschaltung// Alarm with shutdown setzt den Ausgang <i>bCriticalAlm</i> vom Funktionsbaustein AlarmPlt in Template BAC_PltAlm_01 [▶ 372]// sets the output <i>bCriticalAlm</i> by the function blockAlarmPltinTemplateB AC_PltAlm_01 [▶ 372]
END_VAR			

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.2 BAC_Gen_01

Anwendung

Das Template `BAC_Gen_01` generiert Basisfunktionen, Systemdaten und globale Variablen. Es ist Voraussetzung für den Betrieb vieler anderer Aufruf- und Sub-Templates.

Es ist deshalb zwingend erforderlich das Aufruftemplates `BAC_Gen_01` einmalig pro Automationsstation zu laden!

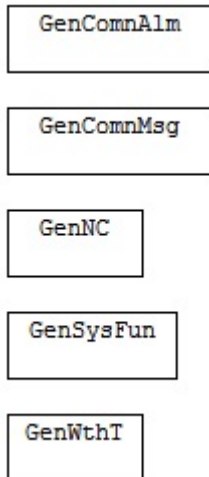
Das Template `BAC_Gen_01` ist ein Aufruftemplates und beinhaltet die folgenden Sub-Templates:

- **GenComnAlm** Das Sub-Template `BAC_GenAlm_01` [▶ 356] sammelt alle Alarmer innerhalb einer Automationsstation.
- **GenComnMsg** Das Sub-Template `BAC_GenComnMsg_01` [▶ 360] sammelt alle Ereignismeldungen der BACnet-Objekte innerhalb einer Automationsstation.
- `BAC_GenDvc_01` [▶ 363] Das Sub-Template ist für die Anbindung des PLC Programmes an ein lokales BACnet Device Objekt (Server) zuständig.
- **GenNC** Das Sub-Template `BAC_GenNC_01` [▶ 368] stellt für das Absetzen der objektintegrierten Meldungen (intrinsic reporting) acht BACnet-Meldeklassen bereit.
- **GenSysFun** Das Sub-Template `BAC_GenSys_01` [▶ 370] liest die Systemzeit aus dem PC aus und bildet die Zeitinformationen in der PLC auf Variablen ab. Zudem wird ein Blinkimpuls erzeugt.
- **GenWthT** Das Sub-Template `BAC_GenWthT_01` [▶ 371] bildet via AV-Objekte die Außentemperatur und die gemittelte Außentemperatur dar.
- `BAC_GenGlobal_01` [▶ 364] Das Sub-Template erzeugt eine Liste mit globalen Variablen und Konstanten für den Betrieb anderer Sub- und Aufruftemplates.

Schnittstelle

BAC_Gen_01

Blockschaltbild



Programmbeschreibung

Instanz	Typ	Aufgabe
GenConnAlm	BAC_GenAlm_01 [▶ 356]	Das Sub-Template BAC_GenAlm_01 [▶ 356] sammelt alle Alarme in einer Automationsstation
GenConnMsg	BAC_GenConnMsg_01 [▶ 360]	Das Sub-Template BAC_GenConnMsg_01 [▶ 360] sammelt alle Ereignismeldungen der BACnet-Objekte innerhalb einer Automationsstation.
BAC_GenDvc_01 [▶ 363]		Das Sub-Template ist für die Anbindung des PLC Programmes an ein lokales BACnet Device Objekt (Server) zuständig.
GenNC	BAC_GenNC_01 [▶ 368]	Das Sub-Template BAC_GenNC_01 [▶ 368] stellt für das Absetzen der objektintegrierten Meldungen (intrinsic reporting) acht BACnet-Meldeklassen bereit
GenSysFun	BAC_GenSys_01 [▶ 370]	Das Sub-Template BAC_GenSys_01 [▶ 370] liest die Systemzeit aus dem PC aus und bildet die Zeitinformationen in der PLC auf Variablen ab. Zudem wird ein Blinkimpuls erzeugt.
GenWthT	BAC_GenWthT_01 [▶ 371]	Das Sub-Template BAC_GenWthT_01 [▶ 371] bildet via AV-Objekte die Außentemperatur und die gemittelte Außentemperatur dar.
	BAC_GenGlobal_01 [▶ 364]	Das Sub-Template erzeugt die globalen Variablen und Konstanten innerhalb einer Automationsstation.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.3 BAC_GenAlm_01

Funktionsbeschreibung

Das Template **BAC_GenAlm_01** sammelt alle Alarmer in einer Automationsstation. Ein Sammelarm wird zur Ansteuerung einer Sammelstörmeldelampe an einen BO-Objekt und zur Anzeige in der MBE als BV-Objekt weitergegeben.

Ein weiteres BV-Objekt erlaubt ein zentrales Quittieren aller Alarmer von der MBE oder einem lokalen Bediengerät. Für das Quittieren der Alarmer mit einem Quittiertaster im Schaltschrank ist ein BI-Objekt vorhanden. Der Quittierimpuls wird zum hereinwischen von Relaisschaltungen (z. B. Frostschutzrelais) mit einem BO-Objekt verknüpft.

Sämtliche Alarmer werden in der Target Visualisierung des PLC Controls dargestellt.

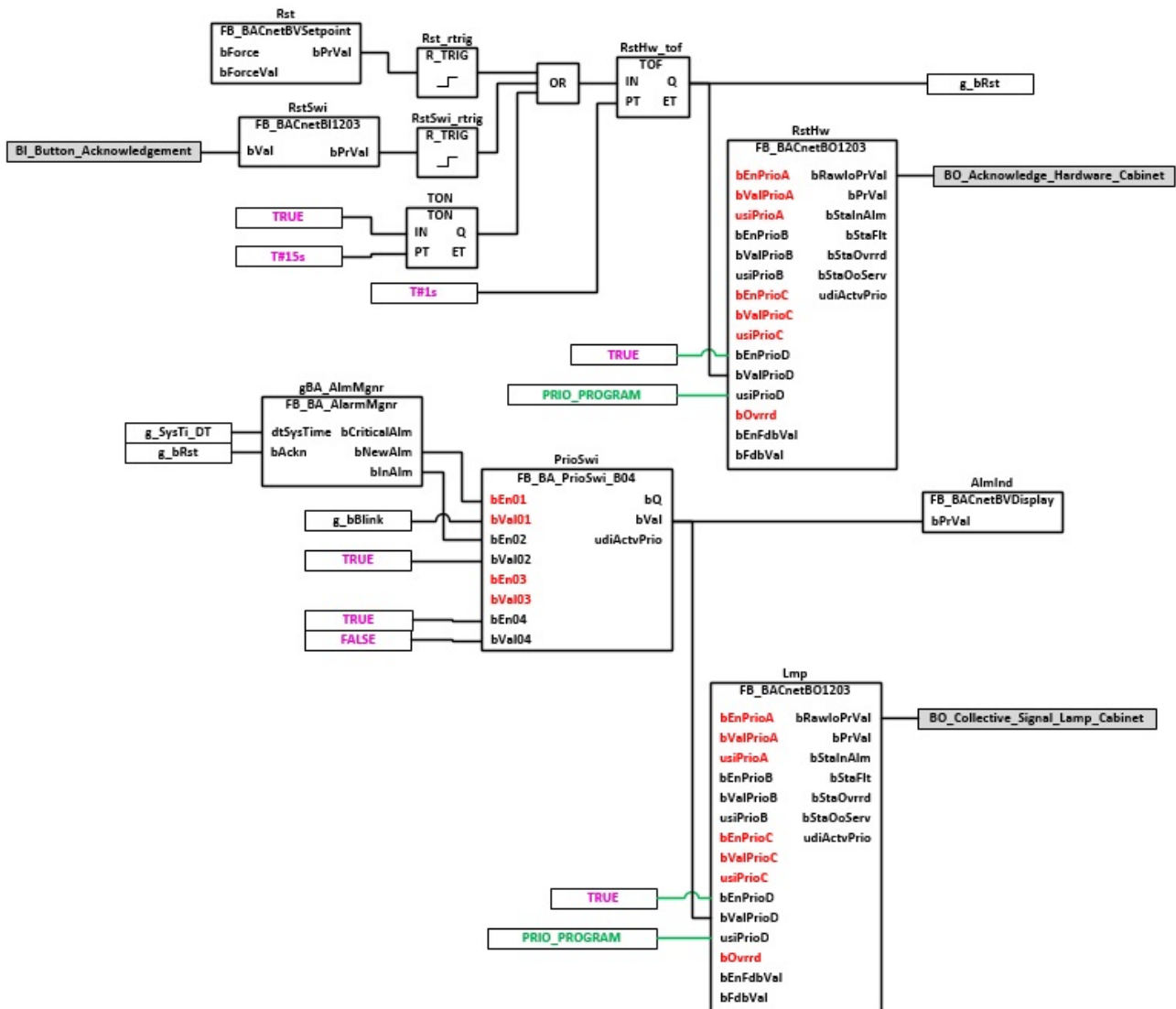
Das Template **BAC_GenAlm_01** legt die für das Alarming benötigten Globale Variablen Alarming [► 353] an.

Unter **IO-Verknüpfung** sind die Variablen zu finden, die mit dem Prozessabbild der Eingangs- und Ausgangsebene in der SPS verknüpft werden.

Schnittstelle

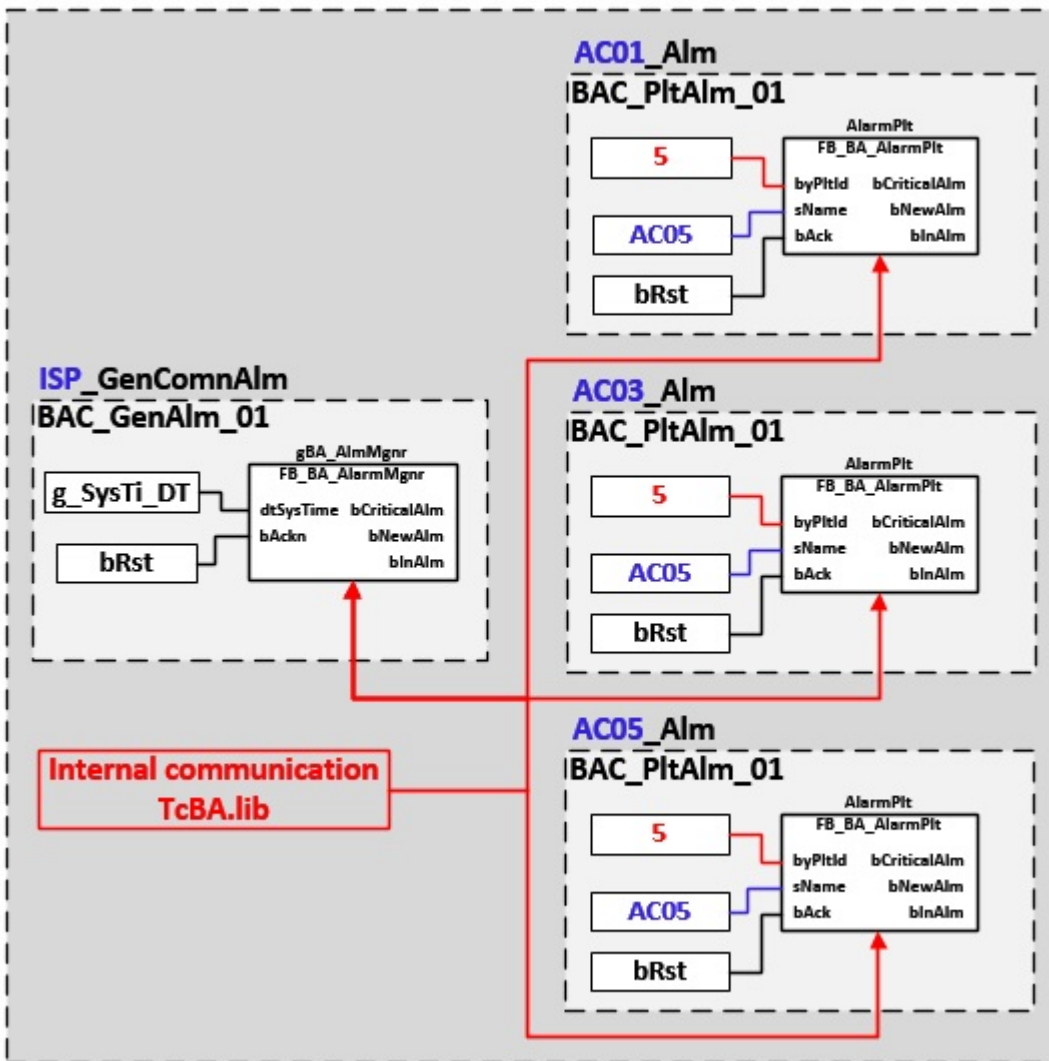
BAC_GenAlm_01

Blockschaltbild

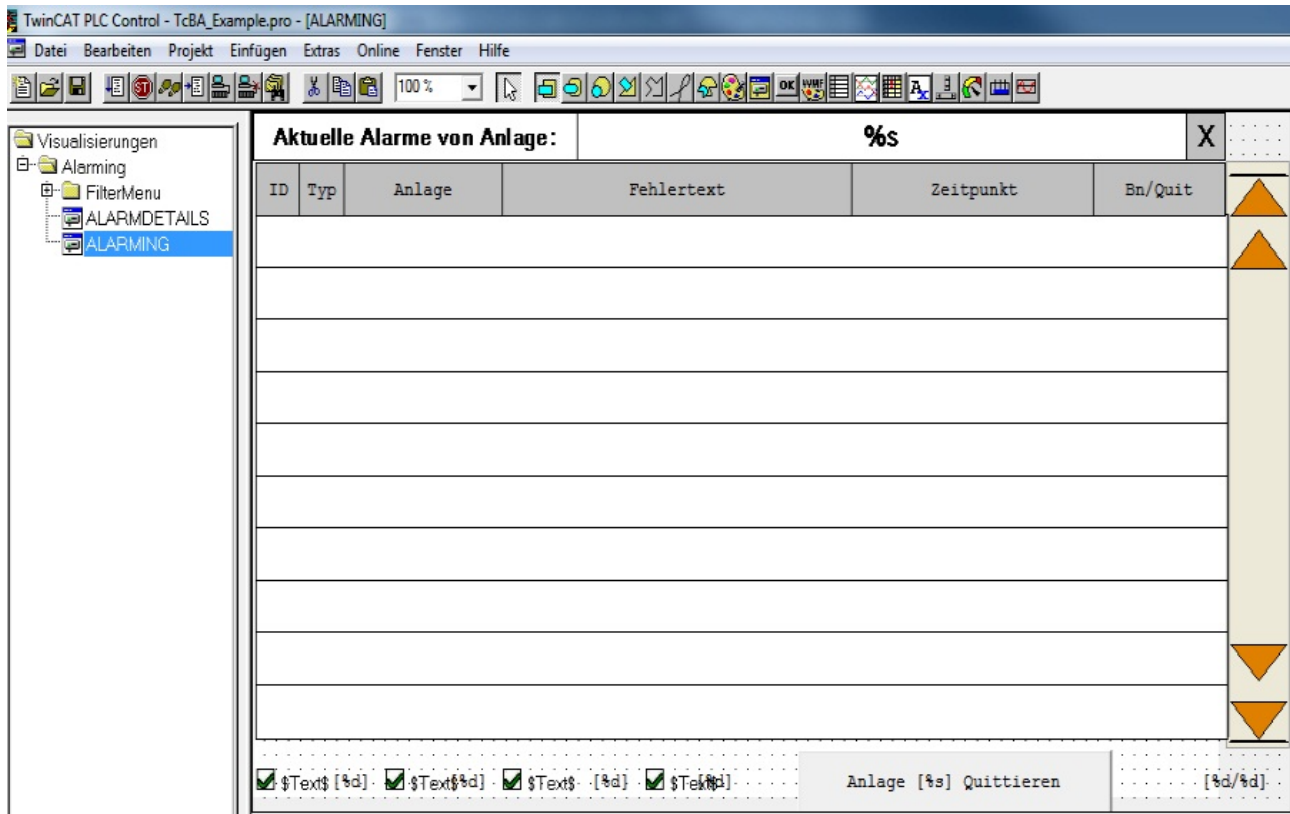


Blockschaltbild der internen Kommunikation

Der Funktionsbaustein **FB_BA_AlarmPlt** [► 186] sammelt die Alarmer einer Anlage. Mit der Kommunikationsstruktur **gBA_AlarmMgnr** werden die Sammelalarmer der Anlagen, an den Funktionsbaustein **FB_BA_AlarmMgnr** [► 184] übergeben. Dieser bildet daraus wiederum einen Sammelalarm und fast damit die Anlagen-Alarmer zu einem globalen Sammelalarm aller Anlagen zusammen. Die Kommunikationsstruktur **gBA_AlarmMgnr** dient zur Übertragung eines globalen Quittierimpulses an alle Alarmbausteine innerhalb des SPS-Programms. Außerdem wird die Systemzeit zur Bildung eines Zeitstempels an die Alarmbausteine übertragen. Die Kommunikationsstruktur ist ein interner Bestandteil der **TcBa.lib**



Target Visualisierung PLC Control



VAR CONSTANT

```
PLT_NUM : BYTE := 0;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmerbausteins **FB_BA_Alarm**. [▶ 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [▶ 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
Rst	FB_BACnetBVSetpoint [▶ 97]	BV-Objekt für das Quittieren aller Alarmer in einem Controller
RstSwi	FB_BACnetBI1203 [▶ 72]	BI-Objekt für den Anschluss eines Quittiertasters.
Rst_rtrig	R_TRIG	Flankenbewertung des Quittiersignals vom BV-Objekt Rst
RstSwi_rtrig	R_TRIG	Flankenbewertung des Quittiersignals vom BI-Objekt RstSwi
RstAuto	TON	Einschaltverzögerung führt 15 Sekunden nach einem Neustart der PLC ein automatisches Quittieren aller Alarmer aus.

Instanz	Typ	Aufgabe
	OR	Fast die Quittierbefehle vom automatischen Quittieren nach PLC-Neustart, BV-Objekt (Quittieren von der MBE) oder BI-Objekt (Quittiertaste im Schaltschrank) zusammen
RstHw_tof	TOF	Verlängerung des Quittierimpulses auf die Dauer von einer Sekunde
RstHw	FB_BACnetBO1203 [► 82]	Das BO-Objekt dient zur Ausgabe des Quittierimpulses an einen digitalen Ausgang z. B. für das Hereinwischen eines Frostschutzrelais im Schaltschrank.
gBA_AlmMgnr	FB_BA_AlarmMgnr [► 184]	Der Funktionsbaustein FB_BA_AlarmMgnr [► 184] ist der zentrale Baustein dieses Templates. Er sammelt die Alarmer des gesamten Controllers ein und bündelt diese zu einem Sammelalarm. Erzeugt werden die Alarmer innerhalb der anlagenzugehörigen Templates durch den Baustein FB_BA_Alarm. [► 182].
PrioSwi	FB_BA_PrioSwi_B04 [► 216]	Bei einem nicht quitierten Alarm wird der Ausgang bNewAlarm des Funktionsbausteins TRUE. In diesem Fall wird vom Prioritätenschalter PrioSwi ein Blinksignal und den Ausgang bVal durchgeschaltet. Die nachgeschaltete Störmeldelampe am BO-Objekt und das BV-Objekt blinken. Nach dem Quittieren des Alarms geht das Blinken in Dauerlicht über. Geht der Sammelalarm, dann wird der Ausgang von PrioSwi FALSE gesetzt. Beim Eingang eines neuen Alarms beginnt die Sammelmeldelemp wieder zu blinken bis ein erneutes Quittieren erfolgt. (Neuwertmeldung)
AlmInd	FB_BACnetBVDisplay [► 96]	BV-Objekt zur Anzeige eines Sammelalarms
Lmp	FB_BACnetBO1203 [► 82]	BO-Objekt zur Ansteuerung einer Sammelstörmeldelampe im Schaltschrank

IO-Verknüpfung

Variablen zur Verknüpfung mit den Busklemmen

Parameter	Typ	optional	Prozessab- bild	
BI_Button_Acknowledgement	BOOL		Eingang	Digitaleingang - Taster Quittierung - Meldung - ausgelöst
BO_Acknowledge_Hardware_Cabinet	BOOL		Ausgang	Digitalausgang - Quittierung Hardware Schaltschrank - Schaltbefehl - Ein/Aus
BO_Collective_Signal_Lamp_Cabinet	BOOL		Ausgang	Digitalausgang - Sammelmeldung Lampe Schaltschrank - Schaltbefehl - Ein/Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.4 BAC_GenComnMsg_01

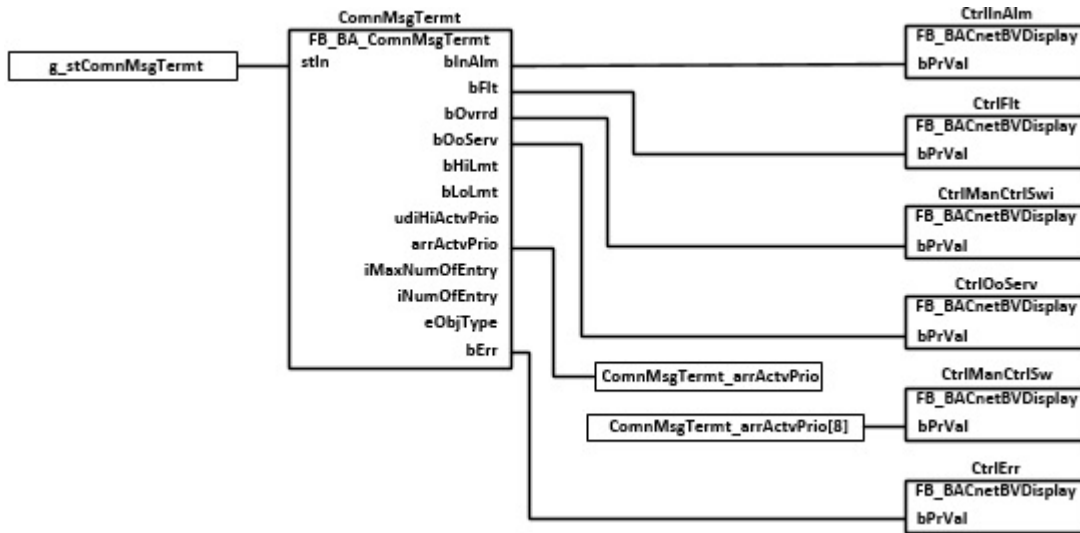
Funktionsbeschreibung

Das Template **BAC_GenComnMsg_01** sammelt alle Ereignismeldungen der BACnet-Objekte innerhalb einer Automationsstation. Zur Anzeige dieser Meldungen in der Management- und Bedienebene (MBE) oder auf einem lokalen Bediendisplay, sind die wichtigsten Meldungen mit BV-Objekten verknüpft.

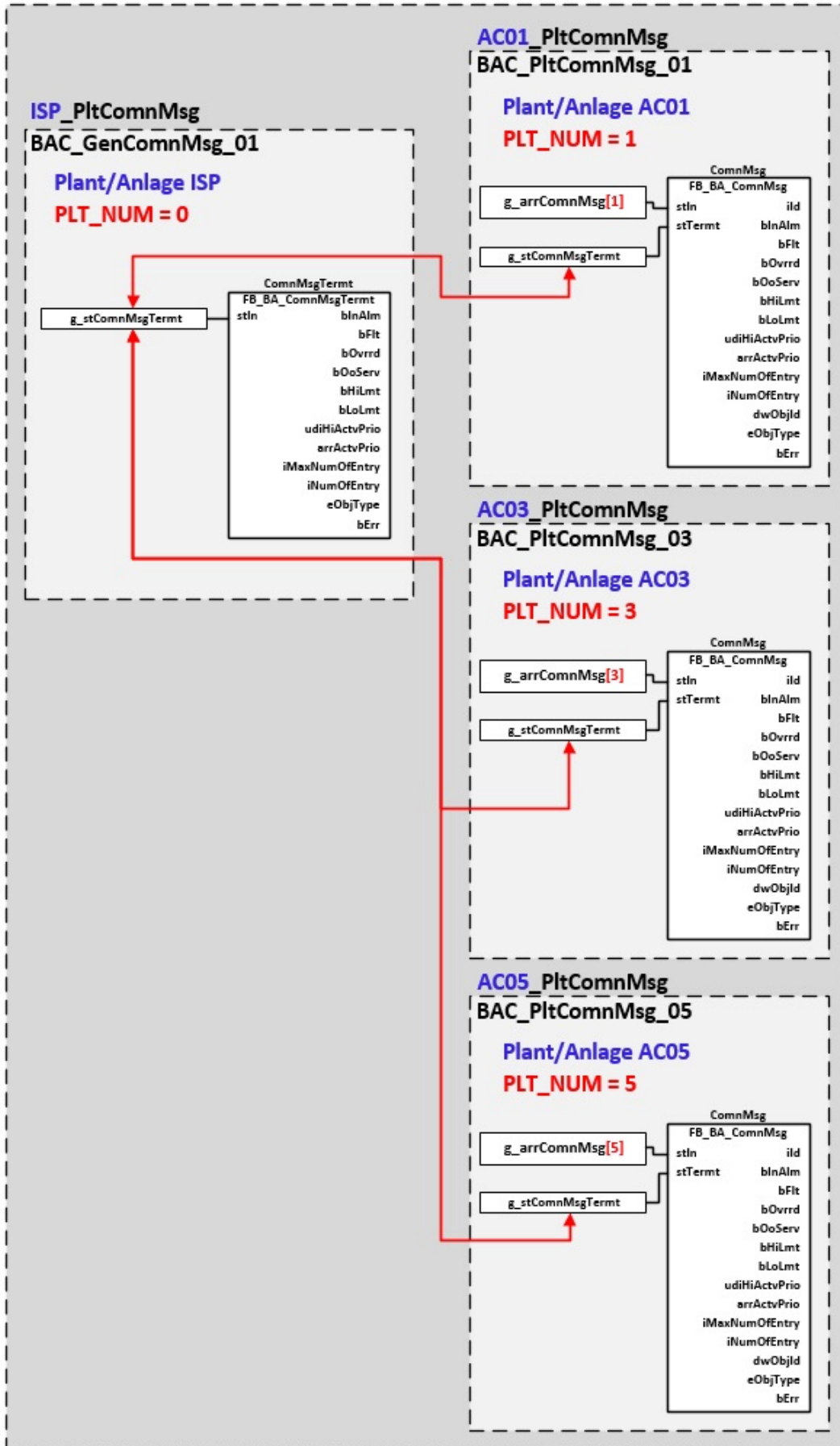
Schnittstelle

BAC_GenCommMsg_01

Blockschaltbild



Blockschaltbild Vernetzung



Programmbeschreibung

Instanz	Typ	Aufgabe
ComnMsgTermt	FB BA ComnMsgTermt [▶ 203]	Der Funktionsbaustein ComnMsgTermt sammelt Ereignismeldungen der BACnet-Objekte im kompletten Device bzw. Controller. Mit der global deklarierten Variablen <u>g_stComnMsgTermt</u> [▶ 364] erhält der Funktionsbaustein die Meldungen aller Meldungssammler auf der Anlagenebene. Zur Anzeige dieser Sammelmeldungen sind einige Informationen des Funktionsbausteins ComnMsgTermt auf BV-Objekte verknüpft.
CtrlInAlm	FB BACnetBVDisplay [▶ 96]	Controllerebene - Sammelmeldung des Statusflags aller BACnet-Objekte in dem Device
CtrlFit	FB BACnetBVDisplay [▶ 96]	Controllerebene - Sammelmeldung der BACnet-Objekte Fault in dem Device
CtrlManCtrlSw	FB BACnetBVDisplay [▶ 96]	Controllerebene - Sammelmeldung Handübersteuerung eines der BACnet-Objekte in dem Device ist auf Prio 8
CtrlManCtrlSwi	FB BACnetBVDisplay [▶ 96]	Controllerebene - an einem der BACnet-Ausgangsobjekt in dem Device ist die örtliche mechanische Vorrangbedienung aktiviert
CtrlOoServ	FB BACnetBVDisplay [▶ 96]	Controllerebene - eines der BACnet-Objekte in dem Device ist OutOfService
CtrlErr	FB BACnetBVDisplay [▶ 96]	Controllerebene - eines der BACnet-Objekte in dem Device ist gestört. z. B. ADS-Fehler

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

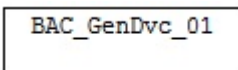
9.5 BAC_GenDvc_01

Funktionsbeschreibung

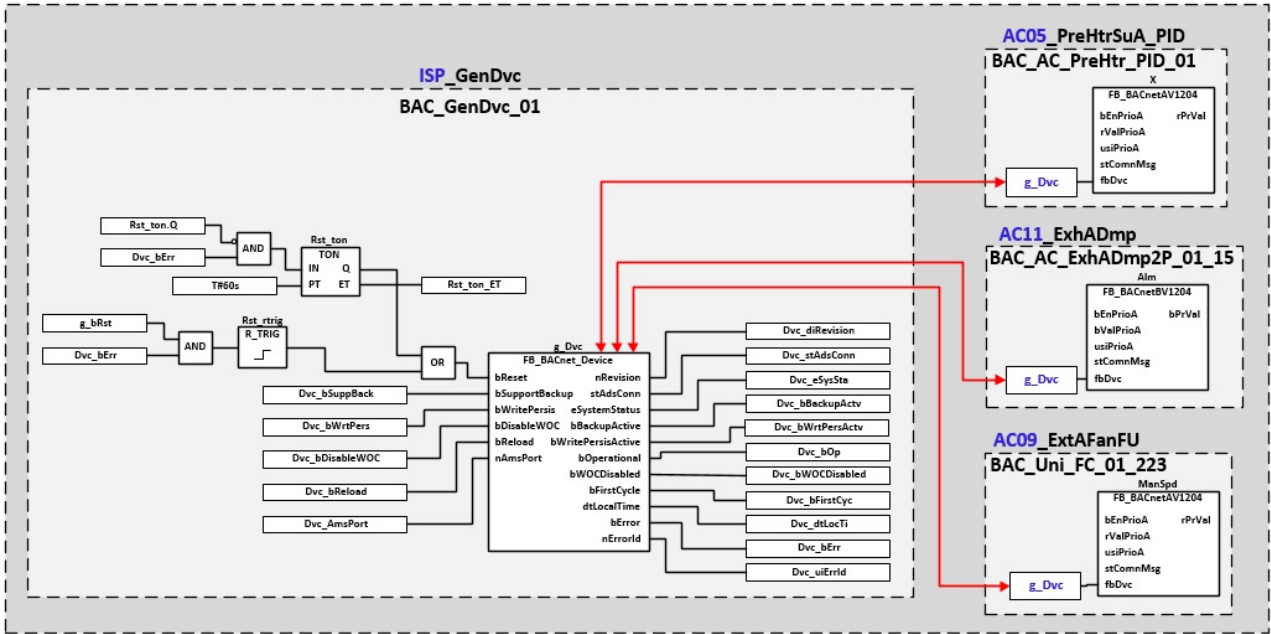
Das Template BAC_GenDvc_01 beinhaltet den Funktionsbaustein FB_BACnet_Device. Es ist für die Anbindung der SPS an das BACnet Device Objekt (Server) zuständig.

Es ist deshalb zwingend erforderlich dieses Template einmalig im SPS-Programm zu platzieren!

Schnittstelle



Blockschaltbild



Programmbeschreibung

Instanz	Typ	Aufgabe
g_Dvc	FB_BACnet_Device	Mit Hilfe des Funktionsbausteins "FB_BACnet_Device" wird der Zustand des lokalen BACnet Device Objects gelesen (System_Status) und im SPS-Programm ausgegeben. Zudem wird mit Hilfe des Prozessdatums "Server Control" der lokale BACnet Server gesteuert (Aktivierung und Handhabung des SPS-Backups)
OR, Rst_ton, Rst_trig		Im Fehlerfall Dvc_bErr wird das BACnet Device Objekt entweder automatisch über das Zeitglied Rst_ton oder über die globale Quitterung g_bRst zurück gesetzt.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.6 Global_Variables_General

In der SPS sind unter **Resources\GlobalVariables\Globale_Variablen_General** die folgenden Funktionsbausteine, Variablen und Konstanten zu finden:

VAR_GLOBAL			
gBA_AlmMgnr	FB_BA_AlarmMgnr		Alarmsammler mit Neuwertmeldung//Alarm collector with new value message
g_stComnMsgTerm	ST_BA_ComnMsgTerm	t [▶ 332]	CommonMessage Abschluss / Terminated Auf diese Struktur darf pro Steuerung aus einem Template heraus nur einmal zugegriffen werden! Folgende Templates greifen auf die globale Struktur zu: BAC_GenComnMsg_01 [▶ 360]

g_bRst	BOOL		Reset Die Variable darf pro Steuerung nur einmal beschrieben werden! Folgende Templates beschreiben die globale Variable: - BAC_GenAlm_01 [▶ 356]
g_bBlink	BOOL		Blinksequenz // Flash sequence Die Variable darf pro Steuerung nur einmal beschrieben werden! Folgende Templates beschreiben die globale Variable: - BAC_GenSys_01 [▶ 370]
g_WthT_rPrVal	REAL		Witterungstemperatur//weather temperature Die Variable darf pro Steuerung nur einmal beschrieben werden! Folgende Templates beschreiben die globale Variable: - BAC_AI_WthT_01 [▶ 689]
g_WthTDamp_rPrVal	REAL		Witterungstemperatur gedämpft // Weather temperature damped Die Variable darf pro Steuerung nur einmal beschrieben werden! Folgende Templates beschreiben die globale Variable: - BAC_GenWthT_01 [▶ 371]
g_WthTLmtCrit	REAL		Witterungstemperatur Grenzwert kritisch // Weather temperature limit critical Die Variable darf pro Steuerung nur einmal beschrieben werden!
g_H_Sp_Frost	REAL		Heizung Sollwert Frost // Heating setpoint frost Die Variable darf pro Steuerung nur einmal beschrieben werden!
g_SysTi_DT	DT		Systemzeit des Zielsystems // System time of the target system Die Variable darf pro Steuerung nur einmal beschrieben werden. Folgende Templates beschreiben die globale Variable: - BAC_GenSys_01 [▶ 370]
g_arrComnMsg	ARRAY [MIN_PLT_NUM..MAX_PLT_NUM] OF ST_BA_ComnMsg [▶ 331]		Array Aufnahme der Objektinformationen // Array Recording the object information
g_stSeqLinkT	ARRAY [MIN_PLT_NUM..MAX_PLT_NUM] OF ST_BA_SeqLink [▶ 333]		Array Sequenz Link Struktur Temperatur // Array Sequence link structure temperature
g_stSeqLinkH	ARRAY [MIN_PLT_NUM..MAX_PLT_NUM] OF ST_BA_SeqLink [▶ 333]		Array Sequenz Link Struktur Feuchte // Array Sequence link structure humidity
END_VAR			
VAR_GLOBAL CONSTANT			
Gemeinsame Meldungen//Common message			
gBA_cMaxArrComnMsg	INT	100	Gemeinsame Meldungen - Maximale Anzahl pro Sammelmeldung
gBA_cMaxArrComnMsgTermt	INT	12	Maximale Anzahl an Sammelmeldungen
Alarmsystem//Alarming			
gBA_MAX_NUM_OF_ALARMS	UDINT	100	Alarmsystem - Maximale Anzahl an Alarmen im Alarmsystem

gBA_MAX_NUM_OF_PLANT_DEFS	BYTE	10	Alarmsystem - Maximale Anzahl an Anlagen
Grenzwerte Anlagennummer//Limit value plant number			
MAX_PLT_NUM	BYTE	1	Maximale Anlagennummer//Maximum Plant Number
Betriebsarten Klimaanlage//Operation Mode Air Conditioning			
OPMOD_AC_OFF	UDINT	1	Aus//Off
OPMOD_AC_ON	UDINT	2	Ein//On
OPMOD_AC_EMERG	UDINT	3	Notfall//Emergency
OPMOD_AC_MANOFF	UDINT	4	Hand aus//Manual off
OPMOD_AC_MANON	UDINT	5	Hand ein//Manual on
OPMOD_AC_FROST	UDINT	6	Frost//Frost
OPMOD_AC_SMOKE_EXTPRG	UDINT	7	Entrauchung Programm//Smoke extraction programm
OPMOD_AC_SMOKE_EXTSUA	UDINT	8	Entrauchung Zuluft//Smoke extraction supply
OPMOD_AC_SMOKE_EXTEXHA	UDINT	9	Entrauchung Fortluft//Smoke extraction exhaust
OPMOD_AC_FIRE	UDINT	10	Feuer//Fire
OPMOD_AC_NIGHTCOOL	UDINT	11	Nachtkühlung//Night cooling
OPMOD_AC_COOLDOWNPRTC	UDINT	12	Stützbetrieb,Auskühlschutz//Cool down protection
OPMOD_AC_OVERHEATINGPRTC	UDINT	13	Überhitzungsschutz//Over heating protection
OPMOD_AC_ALARM	UDINT	14	Störung//Alarm
OPMOD_AC_FORCEDVENT	UDINT	15	Zwangsbelüftung//Forced ventilation
OPMOD_AC_CENTRALSWIOFF	UDINT	16	Zentralabschaltung//Central switch-off
Anlagenschritte Startprogramm Klimaanlage//Plant step start programm Air Conditioning			
PLTSTP_AC_OFF	UDINT	1	Aus//Off
PLTSTP_AC_PRERI	UDINT	2	Vorspülen//pre-rinse
PLTSTP_AC_DAMP	UDINT	3	Klappe//Damper
PLTSTP_AC_FAN_SUA	UDINT	4	Zuluftventilator//Fan supply air
PLTSTP_AC_FAN_EXT	UDINT	5	Abluftventilator//Fan extract air
PLTSTP_AC_ENABLEMPCTRL	UDINT	6	Freigabe Temperaturregelung, Sequenz Link//Enable temperature control, Sequence link
PLTSTP_AC_ENABLEHUMCTRL	UDINT	7	Freigabe Feuchteregelung, Sequenz Link//Enable humidity control, Sequence link
PLTSTP_AC_ENABLEMTMONIT	UDINT	8	Freigabe Grenzwertüberwachung//Enable limit monitoring
PLTSTP_AC_ON	UDINT	9	Ein//On
Betriebsarten Schalter//Operation Mode switch			
OPMOD_SWI_AUTOTO	UDINT	1	Auto//Auto

OPMOD_SWI_OF F	UDINT	2	Aus//Off
OPMOD_SWI_ST P01	UDINT	3	Stufe 1//Step 1
OPMOD_SWI_ST P02	UDINT	4	Stufe 2//Step 2
OPMOD_SWI_ST P03	UDINT	5	Stufe 3//Step 3
Betriebsarten Manuell//Operation Mode manual			
OPMOD_MAN_A UTO	UDINT	1	Auto//Auto
OPMOD_MAN_O FF	UDINT	2	Aus//Off
OPMOD_MAN_ST P01	UDINT	3	Stufe 1//Step 1
OPMOD_MAN_ST P02	UDINT	4	Stufe 2//Step 2
OPMOD_MAN_ST P03	UDINT	5	Stufe 3//Step 3
Betriebsarten Manuell//Operation Mode manual			
OPMOD_AUTO	UDINT	1	Auto//Auto
OPMOD_OFF	UDINT	2	Manual//Manuell
Betriebsarten Heizverteiler//Operation Mode heat distribution			
OPMOD_HTGCI R_AUTO	UDINT	1	Auto//Auto
OPMOD_HTGCI R_OFF	UDINT	2	Aus//Off
OPMOD_HTGCI R_DAY	UDINT	3	Tag//Day
OPMOD_HTGCI R_NIGHT	UDINT	4	Nacht//Night
OPMOD_HTGCI R_FROST	UDINT	5	Frost//Frost
Betriebsarten 3P Ventil//Operation Mode 3P valve			
OPMOD_3PVLV_ AUTO	UDINT	1	Auto//Auto
OPMOD_3PVLV_ OFF	UDINT	2	Aus//Off
OPMOD_3PVLV_ OPEN	UDINT	3	Auf//Open
OPMOD_3PVLV_ CLOSE	UDINT	4	Zu//Close
Betriebsart 2P Klappe//Operation Mode 2P Damper			
OPMOD_2P_AUT O	UDINT	1	Betriebsart 2P Klappe Auto//Operation Mode 2P Damper Auto
OPMOD_2P_CLO SE	UDINT	2	Betriebsart 2P Klappe Close//Operation Mode 2P Damper Close
OPMOD_2P_OPE N	UDINT	3	Betriebsart 2P Klappe Open//Operation Mode 2P Damper Open
Sequenznummer Regler//Sequence Number controller Die Reihenfolge der Sequenznummern muss beachtet werden//The order of the sequence numbers must be observed			
SEQNUM_T_REH TR	USINT	1	Sequenznummer Nacherhitzer//Sequence number reheater

SEQNUM_T_PRE HTR	USINT	2	Sequenznummer Vorerhitzer//Sequence number preheater
SEQNUM_T_MIX	USINT	3	Sequenznummer Mischluft//Sequence number mixed air
SEQNUM_T_ERC	USINT	4	Sequenznummer Energierückgewinnung//Sequence number energy recovery
SEQNUM_T_COL	USINT	5	Sequenznummer Kühler//Sequence number cooler
SEQNUM_T_OFF	USINT	6	kein Sequenzregler aktiv//no sequence controller active
SEQNUM_H_HU MF	USINT	1	Sequenznummer Befeuchter//Sequence number humidifier
SEQNUM_H_DEH UMF	USINT	2	Sequenznummer Entfeuchter//Sequence number dehumidifier
SEQNUM_H_OFF	USINT	3	kein Sequenzregler aktiv//no sequence controller active
Hydrostatischer Luftdruck//atmospheric pressure			
AP	REAL	1013.25	
BACnet Priority//BACnet Priorität			
PRIO_SAFETY	USINT	1	Priority Safety//Priorität Sicherheit
PRIO_DISTURBA NCE	USINT	3	Priority Disturbance//Priorität Störung
PRIO_LOCAL	USINT	8	Priority Local Access, panel//Priorität lokaler Zugriff, Panel
PRIO_PROGRAM	USINT	15	Priority Program, Auto//Priorität Program, Auto
Alarming Priority			
ALM_PRIO_EMPT Y	BYTE	0	keine Meldung//no message
ALM_PRIO_NOTE	BYTE	1	diverse Meldungen//various message
ALM_PRIO_WAR NING	BYTE	2	Warnung//warning
ALM_PRIO_ALAR M	BYTE	3	Alarm ohne Abschaltung//Alarm without shutdown
ALM_PRIO_CRITI CAL	BYTE	4	Alarm mit Abschaltung//Alarm with shutdown setzt den Ausgang <i>bCriticalAlm</i> vom Funktionsbaustein AlarmPlt in Template BAC_PltAlm_01 [▶ 372] // sets the output <i>bCriticalAlm</i> by the function block AlarmPlt in Template BAC_PltAlm_01 [▶ 372]
END_VAR			

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.7 BAC_GenNC_01

Funktionsbeschreibung

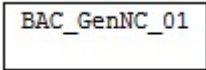
Das Template BAC_GenNC_01 stellt für das Absetzen der objektintegrierten Meldungen (intrinsic reporting) acht BACnet-Meldeklassen bereit. Das Intrinsic Reporting der BACnet-Objekte innerhalb der Templates ist auf die Verwendung dieser Meldeklassen abgestimmt.

Die Definition der Meldeklassen ist dem AMEV Standard entnommen. (Arbeitskreis Maschinen- und Elektrotechnik staatlicher und kommunaler Verwaltungen)

Das Laden des Templates BAC_Gen_NC_01 ist zwingend erforderlich! Das Template BAC_Gen_NC_01 darf nur einmal geladen werden!

Beispiel- Notification-Class und -Sink

Schnittstelle



VAR CONSTANT

```
PLT_NUM : BYTE := 0;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA Alarm](#). [[182](#)]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [[372](#)] mittels des Funktionsbausteins [FB_BA AlarmPlt](#). [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA ComnMsg](#) [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
NC10	FB_BACnetNC1202	Meldeklasse 10 Gefahr für Leben
NC20	FB_BACnetNC1202	Meldeklasse 20 Sicherheitsmeldungen
NC30	FB_BACnetNC1202	Meldeklasse 30 Meldung signalisiert Anlagenausfall oder erfordert sofortigen Eingriff
NC40	FB_BACnetNC1202	Meldeklasse 40 Störungsmeldung
NC50	FB_BACnetNC1202	Meldeklasse 50 Wartungsmeldung
NC60	FB_BACnetNC1202	Meldeklasse 60 Systemmeldung
NC70	FB_BACnetNC1202	Meldeklasse 70 Handeingriff
NC80	FB_BACnetNC1202	Meldeklasse 80 Sonstige Meldungen

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.8 BAC_GenSys_01

Funktionsbeschreibung

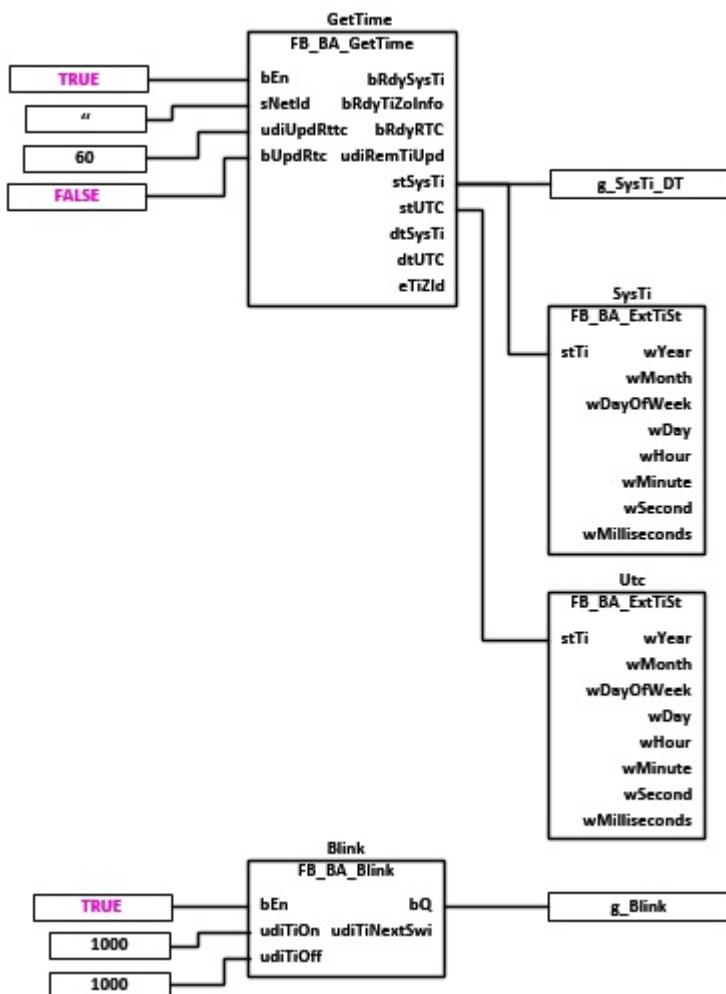
Das Template **BAC_GenSys_01** ist ein Systemfunktions-Template.

Das Template liest die Systemzeit aus dem PC aus und bildet die Zeitinformationen in der SPS auf Variablen ab. Verschiedene Zeitangaben werden innerhalb der Templates für die Anlagen- und Raumautomation benötigt.

Außerdem wird ein Blinkimpuls erzeugt und über eine globale Variable anderen Templates zur Verfügung gestellt, z. B. siehe dem Blockschaltbild im Template [BAC_PltAlm_01](#) [▶ 372].

Das Laden des Templates BAC_GenSys_01 ist zwingend erforderlich! Das Template BAC_GenSys_01 darf nur einmal geladen werden!

Blockschaltbild



Schnittstelle

BAC_GenSys_01

Programmbeschreibung

Instanz	Typ	Aufgabe
GetTime	FB BA GetTime [▶ 325]	Mit diesem Funktionsbaustein wird eine interne Uhr (Real Time Clock RTC) in der TwinCAT SPS realisiert. Die RTC-Uhr wird mit der Freigabe des Funktionsbausteins über bEn mit der aktuellen NT-Systemzeit initialisiert. Es wird ein Systemtakt der CPU benutzt um die aktuelle RTC-Zeit zu berechnen. Der Funktionsbausteins muss in jedem Zyklus der SPS ein Mal aufgerufen werden, damit die aktuelle Zeit berechnet werden kann. Die Ausgabe der Zeit erfolgt an den Ausgängen stSysTi für die gelesene Systemzeit und stUtcTi für die koordinierte Weltzeit (Coordinated Universal Time). Diese wird intern aus der Systemzeit und der Zeitzone ermittelt. Ist die Systemzeit und/oder die Zeitzone fehlerhaft eingegeben worden, so wird auch die UTC-Zeit nicht richtig sein.
SysTi	FB BA ExtTiSt [▶ 324]	Der Funktionsbaustein SysTi löst die Zeitstruktur des ausgelesenen Zielsystems in die verschiedenen Bestandteile auf.
Utc	FB BA ExtTiSt [▶ 324]	Der Funktionsbaustein Utc löst die Koordinierte Weltzeit (Universal Time Coordinated - im älteren Sprachgebrauch auch GMT, Greenwich-Mean-Time) in die verschiedenen Bestandteile auf.
Blink	FB BA Blink [▶ 322]	Der Funktionsbaustein Blink erzeugt einen Blinkimpuls und stellt diesen über eine globale Variable g_Blink [▶ 364] anderen Templates zur Verfügung, z. B. Sammelstörmeldelampe

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.9 BAC_GenWthT_01

Funktionsbeschreibung

Das Template **BAC_GenWthT_01** liest die gemessene Außentemperatur mittels der globalen Variable **g_WthT_rPrVal** [▶ 364] ein.

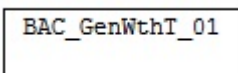
Aus der gemessenen Außentemperatur wird eine gemittelte bzw. gedämpfte Außentemperatur berechnet. Diese wird anderen Templates durch die Globalvariable **g_WthTDamp_rPrVal** [▶ 364] zur Verfügung gestellt.

Das BACnet-Objekt **WthTLmtCrit** beschreibt die globale Variable **g_WthTLmtCrit**. Sie beschreibt einen Außentemperaturwert bei dessen Unterschreitung in den Anlagen der Frostschutzbetrieb aktiviert werden sollte.

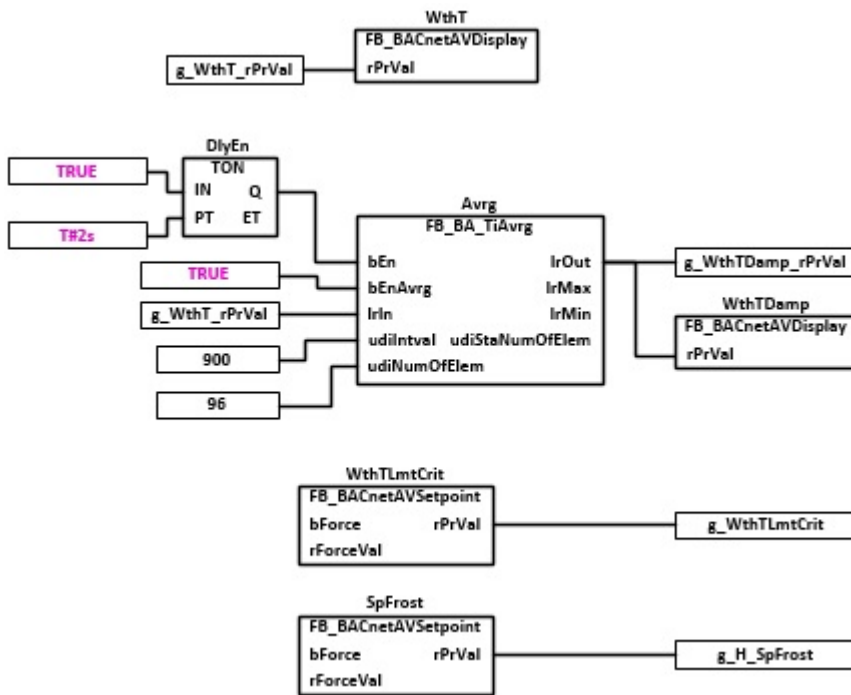
Mit dem AV-Objekt **SpFrost** wird die globale Variable **g_H_SpFrost** beschrieben. Mit diesem globalen Wert kann einer oder mehreren Anlagen ein Sollwert für den Frostschutzbetrieb zugewiesen werden.

Für die Verwendung vieler anderer Templates ist es erforderlich dieses Template einmal zu platzieren!

Schnittstelle



Blockschaltbild



Programmbeschreibung

Instanz	Typ	Aufgabe
WthT	FB_BACnetAVDisplay [▶ 69]	AV-Objekt zur Anzeige der aktuellen Außentemperatur
WthLmtCrit	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe und Bereitstellung eines globalen Wertes für die kritische Außentemperatur
SpFrost	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe und Bereitstellung eines globalen Frostschutzwertes
DlyEn	TON	verzögert das Einlesen des ersten Temperaturwertes nach dem Aufstart der PLC um 2 Sekunden
Avrg	FB_BA_TiAvrg [▶ 180]	Der Funktionsbaustein Avrg misst alle 15 Minuten (900 Sekunden) die Außentemperatur und schreibt den Wert in einen FIFO-Speicher (First In First Out). Innerhalb des Speichers werden 96 Elemente abgelegt. So ergibt sich ein Zeitfenster für die Mittelwertbildung der Außentemperatur von 24 Stunden.
WthTDamp	FB_BACnetAVDisplay [▶ 69]	Ausgabe des Wertes der gemittelten Außentemperatur

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.10 BAC_PltAlm_01

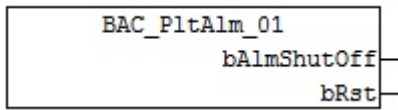
Anwendung

Das Template **BAC_PltAlm_01** hat in einer Anlage die folgenden Aufgaben:

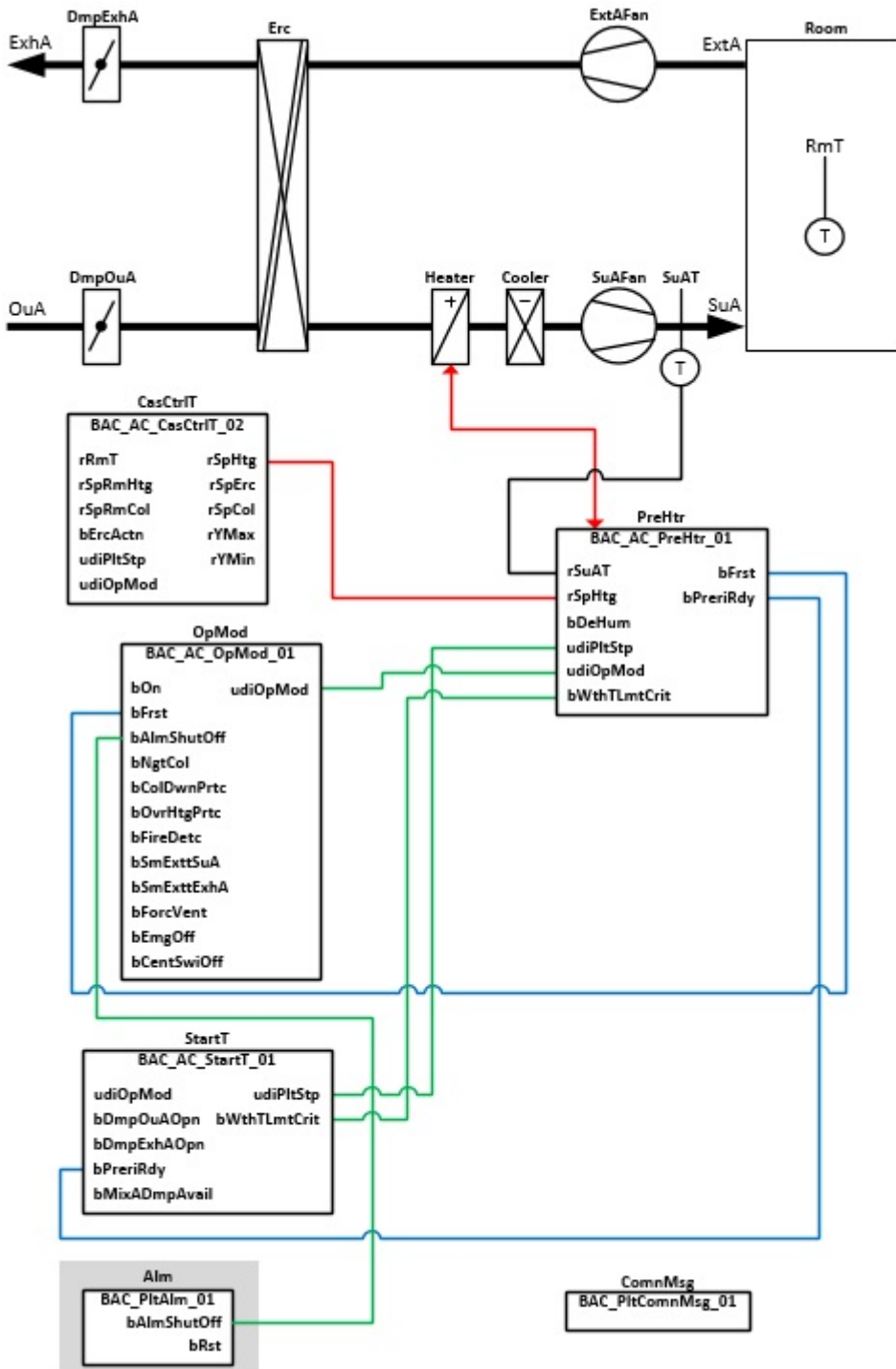
- Sammeln der Alarme einer Anlage und Bündelung dieser zu einem Sammelalarm bzw. anlagenabschaltenden Alarm **bAlmShutOff**

- Anzeige, dass ein Alarm in einer Anlage ansteht.
- Anlagenbezogenes Quittieren und Rücksetzen der Alarme

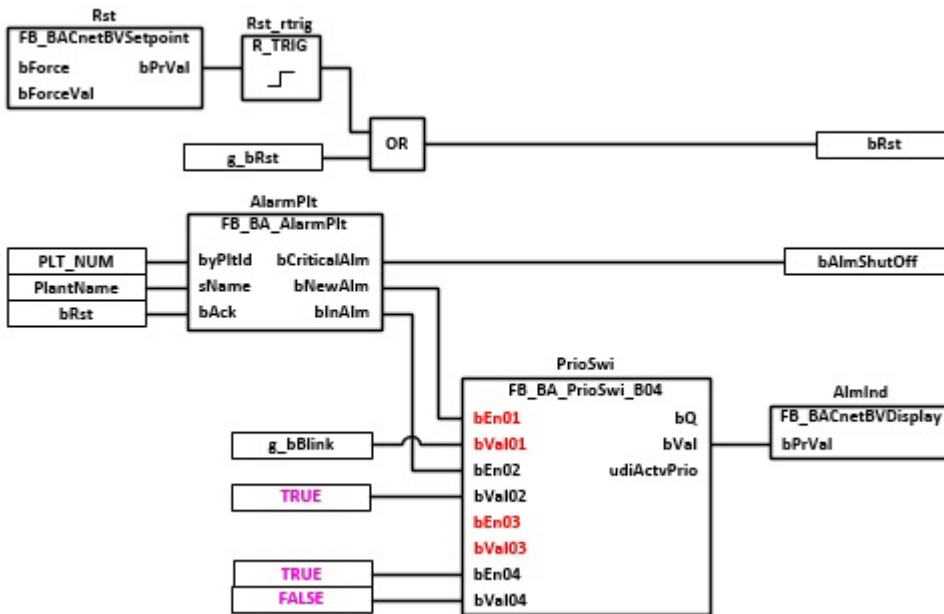
Schnittstelle



Anlagenbild

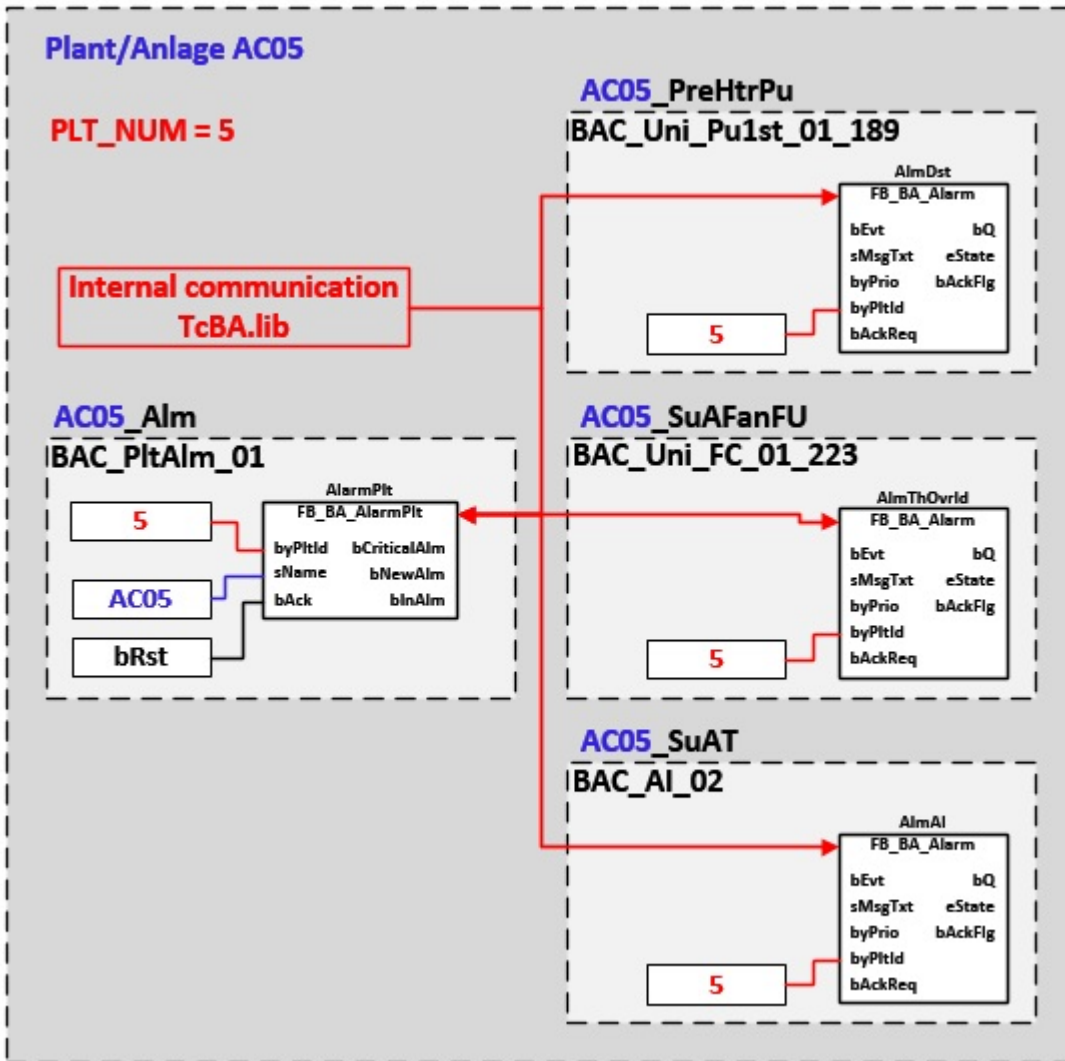


Blockschaltbild



Blockschaltbild interne Kommunikationsstruktur

Die Aufnahme der Alarme innerhalb der Templates erfolgt mit dem Funktionsbaustein `FB_BA_Alarm` [▶ 182]. Die Kommunikation zwischen dem Funktionsbaustein `FB_BA_Alarm` [▶ 182] und dem Alarmsammler `FB_BA_AlarmPlt` [▶ 186] erfolgt innerhalb der Bibliothek mittels einer Datenstruktur. Wichtig ist, dass der Wert von der Variablen `byPltId` beim Sender des Alarms `FB_BA_Alarm` [▶ 182] und beim Empfänger des Alarms `FB_BA_AlarmPlt` [▶ 186] gleich sind.



VAR_OUTPUT

```
bAlmShutOff : BOOL;
bRst        : BOOL;
```

bAlmShutOff: Ein anlagenabschaltemder Alarm ist aktiv.

bRst: Bei einer ausgelösten Quittierung steht dieser Ausgang für einen Zyklus an. Der Impuls kann zum Quittieren von Störungen genutzt werden.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins FB_BA_Alarm. [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates BAC_PltAlm_01 mittels des Funktionsbausteins FB_BA_AlarmPlt. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates BAC_PltComnMsg_01 durch den Funktionsbaustein FB_BA_ComnMsg [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
Rst	FB_BACnetBVSetpoint [▶ 97]	BV-Objekt zur lokalen Quittierung des Anlagensammelalarms.
Rst_rtrig	R_TRIG	Der Baustein Rst_rtrig wertet die steigende Flanke der lokalen Quittierung vom BV-Objekt Rst aus.
	OR	Die Oderverknüpfung fasst die lokale Quittierung vom BACnet-Objekt und die globale Störmeldungsquittierung zusammen. Das globale Quittieren von Alarmen sämtlicher Anlagen eines Controllers über die Variable <i>g_bRst</i> [▶ 364] wird ausgelöst in dem Template <i>BAC_GenAlm_01</i> [▶ 356].
AlarmPlt	FB_BA_AlarmPlt [▶ 186]	Der Funktionsbaustein AlarmPlt ist der zentrale Baustein dieses Templates. Er sammelt die Alarmer einer Anlage ein und bündelt diese zu einem Sammelalarm.
PrioSwi	FB_BA_PrioSwi_B04 [▶ 216]	Bei einem nicht quittierten Alarm wird der Ausgang <i>bNewAlarm</i> des Funktionsbausteins <i>AlarmPlt</i> TRUE. In diesem Fall wird vom Prioritätenschalter <i>PrioSwi</i> ein Blinksignal an den Ausgang <i>bVal</i> durchgeschaltet. Ein neuer nicht quittierter Alarm steht an. (Neuwertmeldung). Bei einem anstehenden aber quittierten Alarm wird vom Prioritätenschalter ein TRUE durchgeschaltet. Der Ausgang <i>bVal</i> ist dauerhaft an.
AlmInd	FB_BACnetBVDisplay [▶ 96]	Anzeige der Sammelalarmer einer Anlage mit Neuwertmeldung.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.11 BAC_PltComnMsg_01

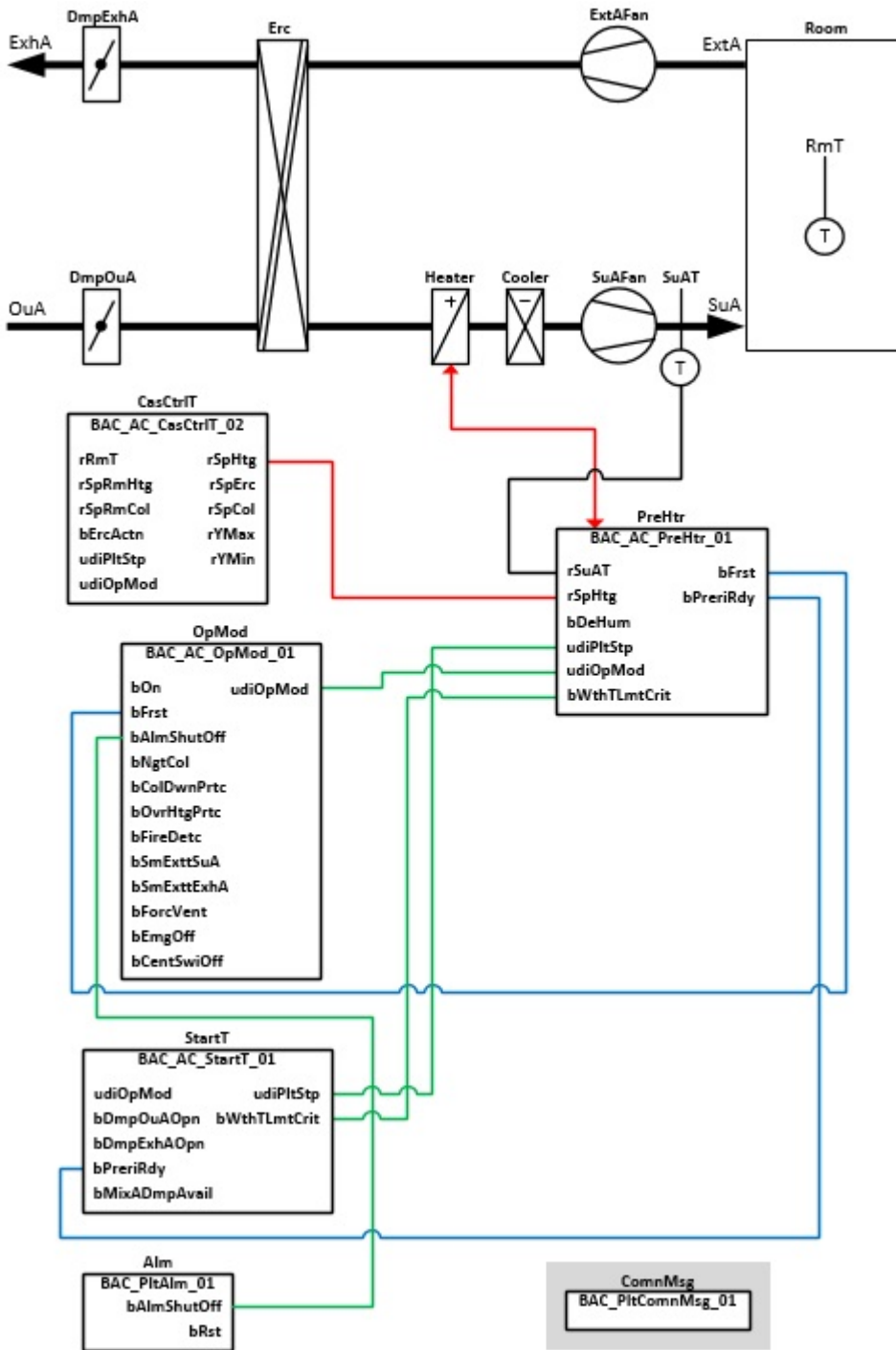
Funktionsbeschreibung

Das Template **BAC_PltComnMsg_01** sammelt alle Ereignismeldungen der BACnet-Objekte innerhalb einer Anlage. Zur Anzeige dieser Meldungen in der Management- und Bedienebenen (MBE) oder auf einem lokalen Bediendisplay sind die wichtigsten Meldungen mit einem BV-Objekt verknüpft.

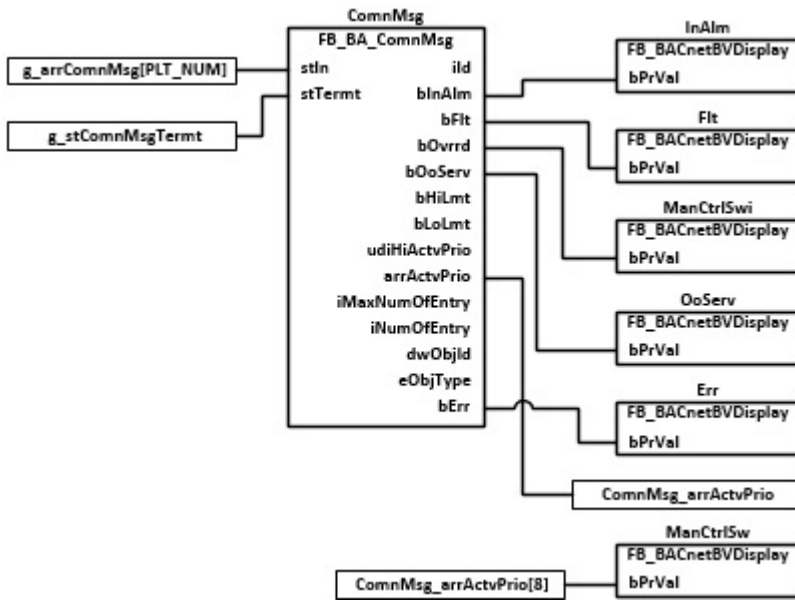
Schnittstelle

BAC_PltComnMsg_01

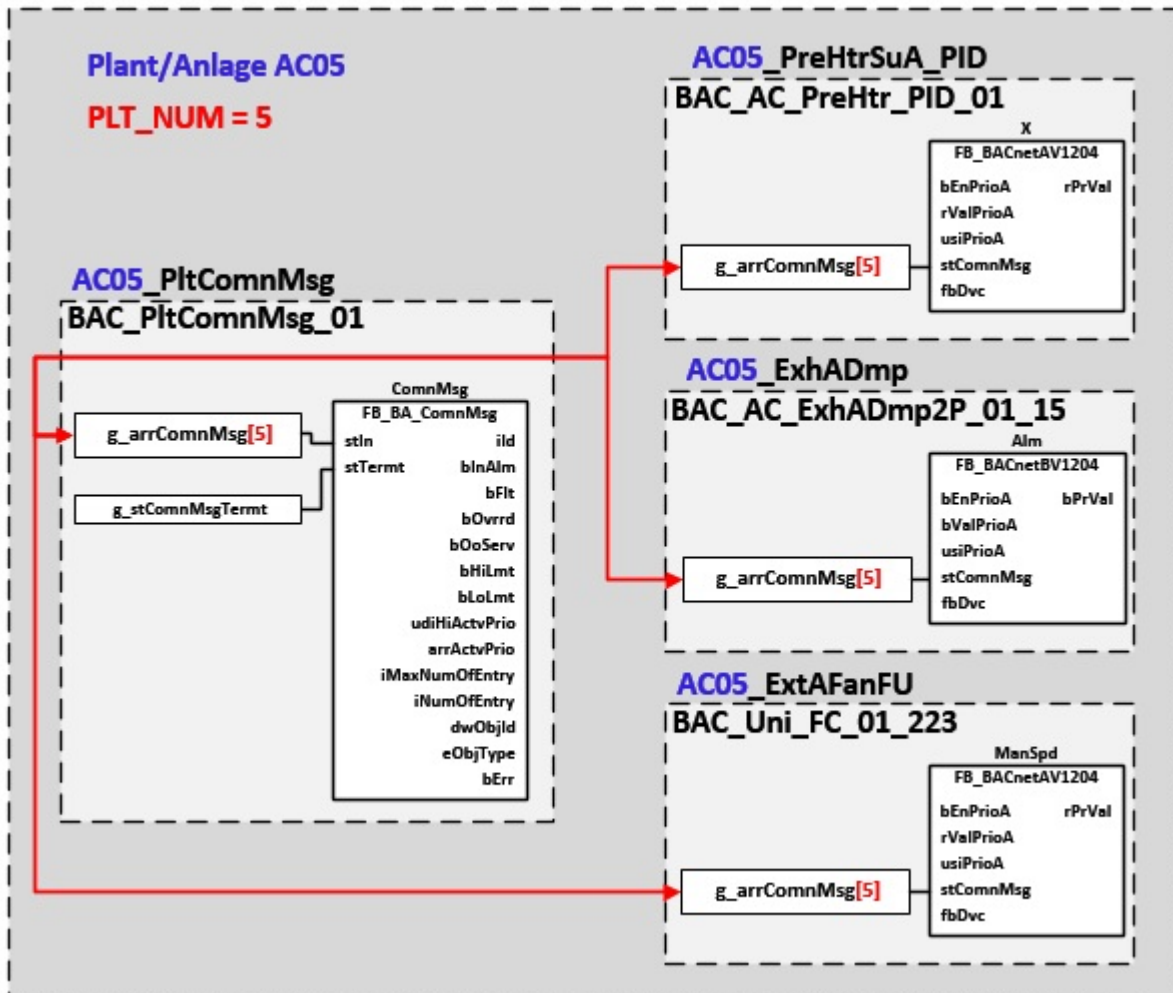
Anlage



Blockschaltbild 01



Blockschaltbild 02



VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [[182](#)]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [[372](#)] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
ComnMsg	FB_BA_ComnMsg [200]	<p>Der Funktionsbaustein ComnMsg sammelt Ereignismeldungen der BACnet-Objekte einer Anlage. Zur Anzeige dieser Sammelmeldungen sind einige Informationen des Funktionsbausteins ComnMsg auf ein BV-Objekten verknüpft.</p> <p>Die Übertragung der Meldungen von den BACnet-Objekten zum Meldebaustein ComnMsg erfolgt mittels eines eindimensionalen Arrays der Struktur ST_BA_ComMsg. Über die Anlagennummer reserviert sich jede Anlage ein Element dieses Arrays. Das Array für das Sammeln der Meldungen aus den BACnet-Objekten innerhalb eines Controllers ist in dem Ordner Globale Variablen General [364] global deklariert.</p> <p>Wichtig ist, dass der Meldesammer FB_BA_ComnMsg und die BACnet-Objekte innerhalb der Templates einer Anlage auf die gleiche Anlagennummer eingestellt sind!</p> <pre>VAR CONSTANT PLT_NUM : BYTE := 1; (*Anlagennummer//Plant Number*) END_VAR</pre> <p>Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.</p>
InAlm	FB_BACnetBVDisplay [96]	Sammelmeldung des Statusflags der BACnet-Objekte
Flt	FB_BACnetBVDisplay [96]	Sammelmeldung der BACnet-Objekte Fault
ManCtrlSw	FB_BACnetBVDisplay [96]	Sammelmeldung Handübersteuerung eines der BACnet-Objekte ist auf Prio 8
ManCtrlSw i	FB_BACnetBVDisplay [96]	An einem der BACnet-Ausgangsobjekt ist die örtliche mechanische Vorrangbedienung aktiviert
OoSrv	FB_BACnetBVDisplay [96]	Eines der BACnet-Objekte in der Anlage ist OutOfService
Err	FB_BACnetBVDisplay [96]	Eines der BACnet-Objekte ist gestört. z. B. ADS-Fehler

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.12 BAC_AC_Humf_01

Anwendung

Das Template **BAC_AC_Humf_01** dient der Steuerung und Regelung des Befeuchters.

Das Template **BAC_AC_Humf_01** ist ein Aufruftemplate.

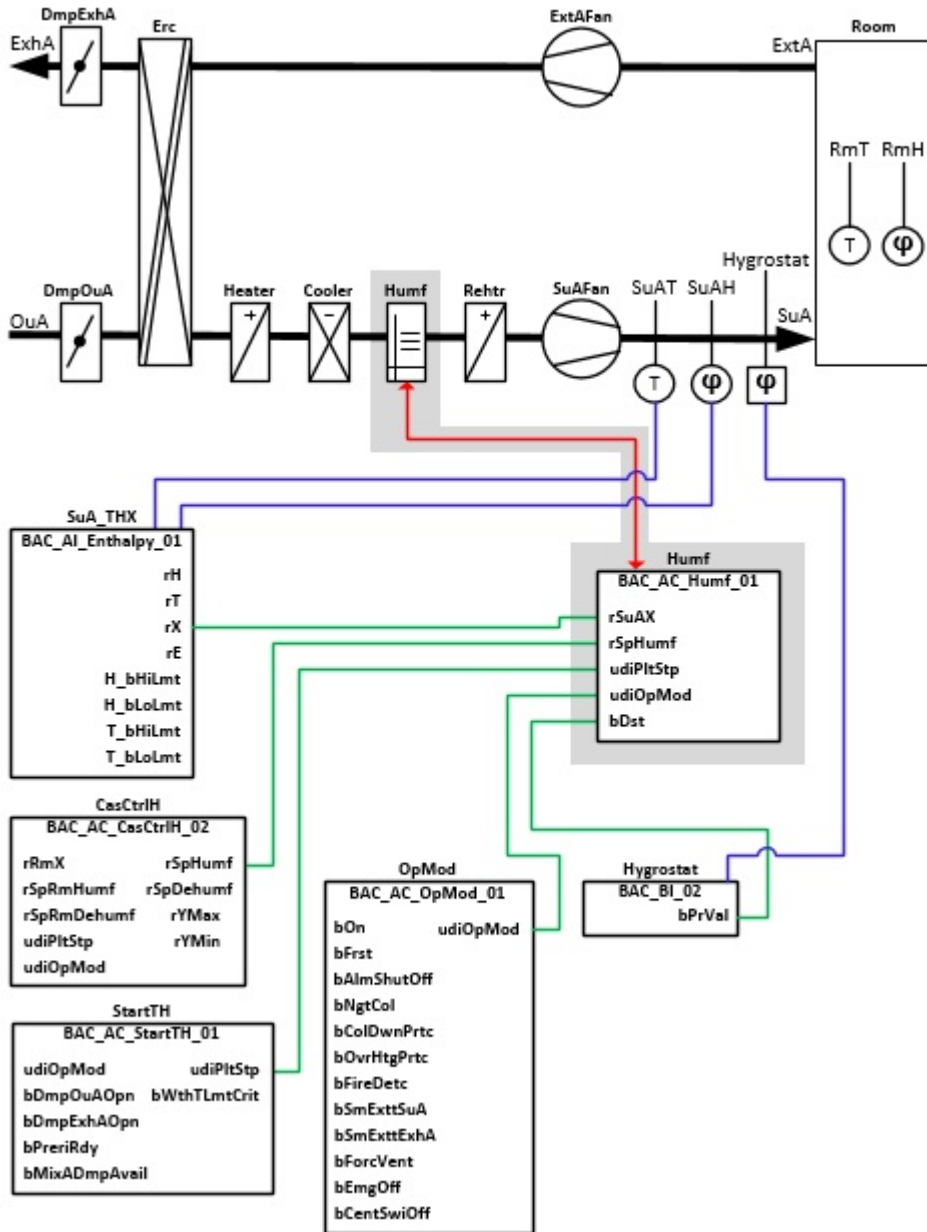
Innerhalb des Aufruftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

- **HumfPID** Regelung der absoluten Zuluftfeuchte
- **Steam** Ansteuerung des Dampferzeugers

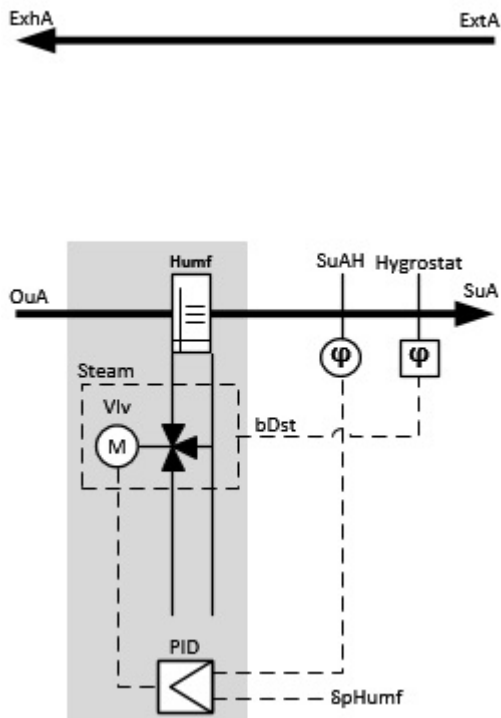
Schnittstelle

BAC_AC_Humf_01	
-rSuAX	
-rSpHumf	
-udiPltStp	
-udiOpMod	
-bDst	

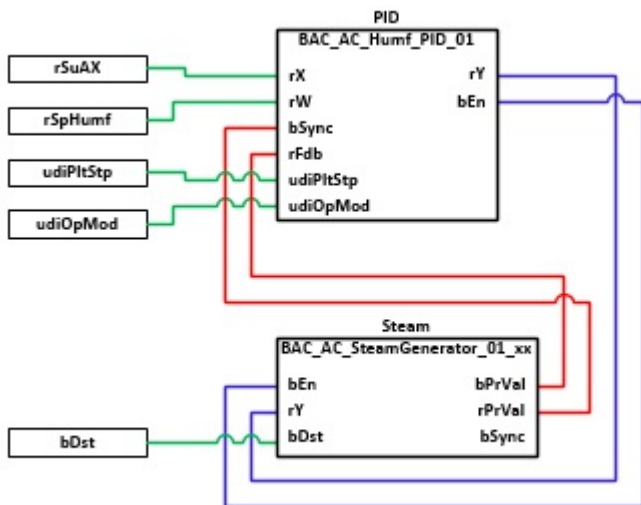
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

rSuAX      : REAL;
rSpHumf    : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
bDst       : BOOL;

```

rSuAX: Errechneter Wert der absoluten Zulufffeuchte. Dieser Wert entsteht im Template [BAC_AI_Enthalpy_01](#) [▶ 687].

rSpDehumf: Sollwert für die absolute Zulufffeuchte

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC AC StartTH_01](#) [▶ 543].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC AC OpMod_01](#) [▶ 524].

bDst: Eingang für eine externe Störung des Befeuchters, z. B. ein Hygrostat

Programmbeschreibung

Instanz	Typ	Aufgabe
HumfPID	BAC_AC_Humf_PID_01 [▶ 383]	Sub-Template inklusive des PID-Sequenzreglers für die absolute Zuluftfeuchteregelung.
HumfSteam	BAC_AC_SteamGenerator_01_03 [▶ 388]	Sub-Template zur Ansteuerung eines Dampfgenerators

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.13 BAC_AC_Humf_PID_01

Funktionsbeschreibung

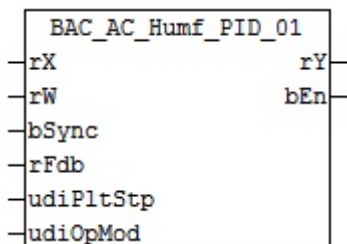
Das Sub-Template **BAC_AC_Humf_PID_01** ist der Sequenzregler für einen Befeuchter. Die Referenzierung des Sollwertes, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte **X**, **W** und **Y**.

Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart **udiOpMod** und der globalen Feuchte-Kommunikationsstruktur **g_stSeqLinkH[PLT_NUM]**. Diese Daten- und Befehlsstruktur ist das Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein [FB_BA_SeqLink](#) [[▶ 171](#)] einer Anlage.

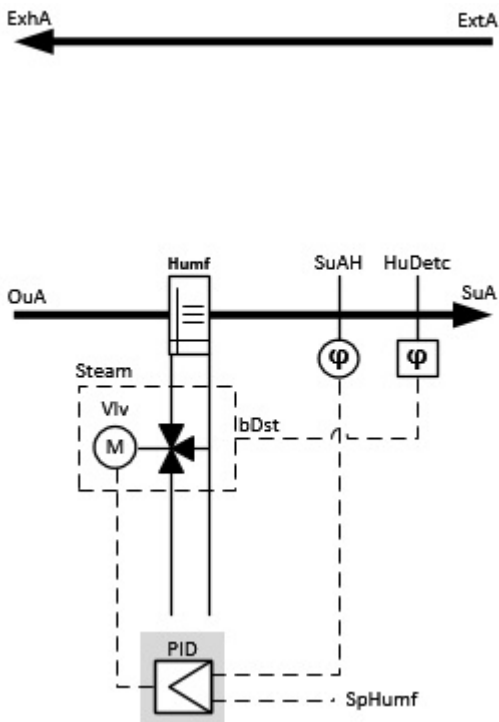
Zur Anzeige der Reglerfreigabe dient das BACnet-BV-Objekt **En**.

Die Grenzwertüberwachung des PID-Reglers wird in Abhängigkeit des Anlagenanfahrprozesses von dem Funktionsbaustein **EnEvtEn** gesteuert.

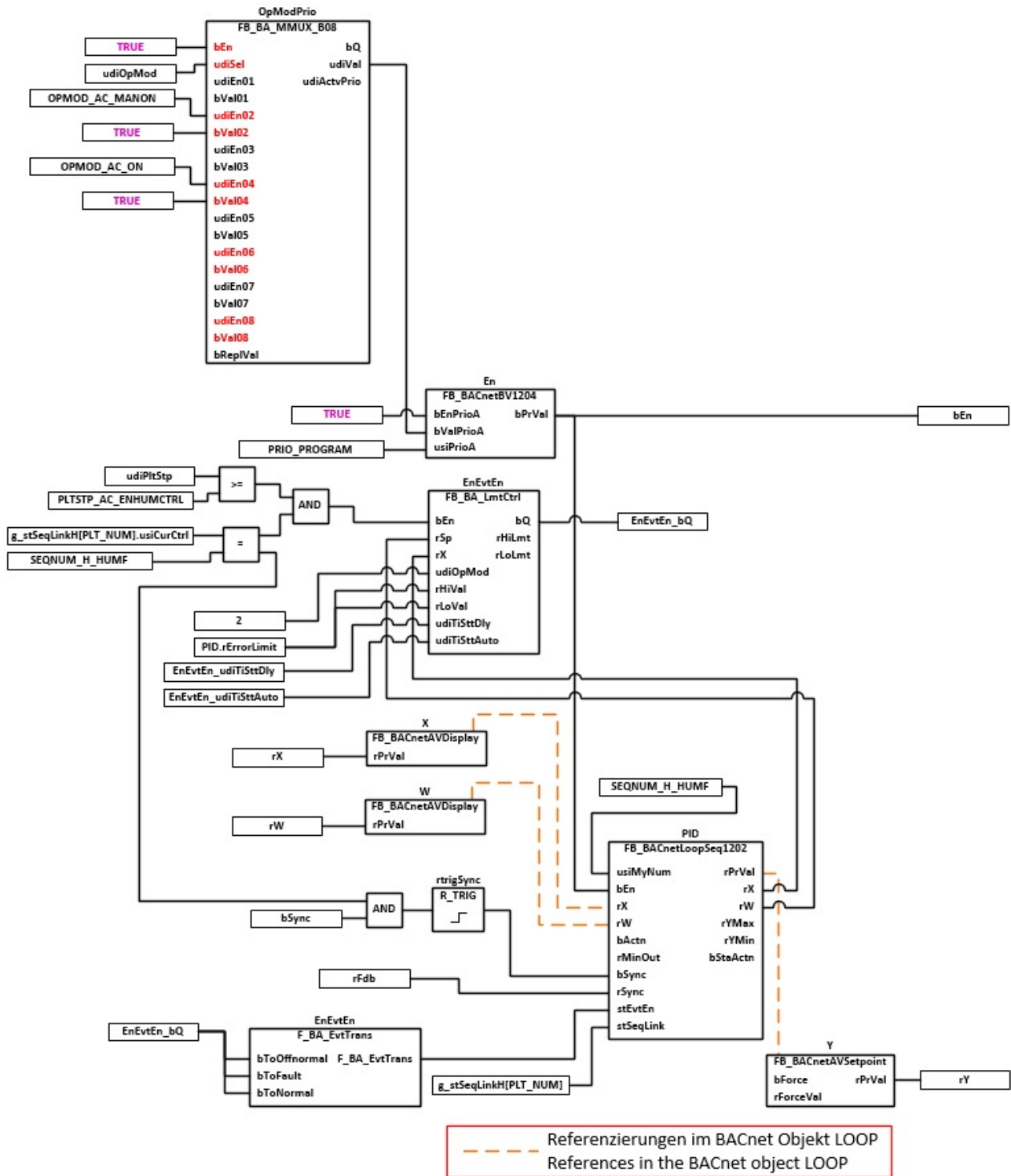
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

```

rX      : REAL;
rW      : REAL;
bSync   : BOOL;
rFdbVlv : REAL;
udiPltStp : UDINT;
udiOpMod : UDINT;
  
```

rX: Errechneter Wert der absoluten Zulufffeuchte

rW: Sollwert der absoluten Zulufffeuchte

bSync: Eingang für die Synchronisation des Reglers

rFdb: Stellungsrückmeldung Aktor

udiPltStp: Schritte beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartTH_01 \[▶ 543\]](#).

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod_01 \[▶ 524\]](#)

VAR_OUTPUT

```
rY : REAL;
```

rY: Ausgabe der Stellgröße

bEn: Ausgabe der Freigabe des PID-Sequenzreglers

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm_ \[▶ 182\]](#)

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01 \[▶ 372\]](#)

mittels des Funktionsbausteins [FB BA AlarmPlt_ \[▶ 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg \[▶ 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur **g_stSeqLinkH[PLT_NUM]** dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein **FB_BA_SeqLink** zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe	Freigabe	Bemerkung
X	FB_BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts		
W	FB_BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts		
OpModPri o	FB_BA_MMUX_B08 [▶ 208]	Der Multiplexer definiert die Freigabebedingungen des Sequenzreglers in Abhängigkeit der Anlagenbetriebsart.		
		udiOpMod		
		OPMOD AC MANON [▶ 364]	Hand ein	TRUE Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet
		OPMOD AC ON [▶ 364]	Ein	TRUE Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm
En	FB_BACnetBV1204 [▶ 94]	Das BV-Objekt dient zur Anzeige und Aktivierung der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay. Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart udiOpMod und der Kommunikationsstruktur stSeqLink .		

Instanz	Typ	Aufgabe
EnEvtEn	<p>FB_BA_LmtCtrl ▶ 233</p>	<p>Das BACnet-Loop-Objekt PID überwacht die Funktion der Regelung in dem es den Sollwert W und den Istwert X miteinander vergleicht. Ist die Abweichung $W-X$ größer als das Property ErrorLimit, dann sendet das Loop-Objekt eine Meldung an die MBE.</p> <p>Beim Anlagenstillstand, im Moment des Starts und bis zum eingeregelter Zustand der Anlage wird das Melden des Loop-Objekts unterdrückt, damit keine falschen Meldungen an die MBE gesendet werden. Erst wenn die RLT-Anlage vollständig hochgefahren ist und die Regelung sich eingependelt hat, wird das Melden des Loop-Objekts freigegeben. Zusätzlich wird das Melden aktiviert, wenn die Regelung nach langer Zeit nicht in dem vom Property ErrorLimit definierten Bereich um den Sollwert gelangt ist.</p> <p>Die Freigabe des objektinternen Meldens erfolgt indem auf die BACnetEventTransitionBits des Loop-Objekts geschrieben wird.</p> <p>Für die Freigabe des Meldens vom Loop-Objekt müssen die folgenden Bedingungen erfüllt sein:</p> <p>1. Das Anlagenstartprogramm BAC_AC_StartTH_01 ▶ 543 hat die Überwachung der Regelung und die Grenzwertüberwachung der Fühler freigegeben udiPitStp >= PLTSTP_AC_ENHUMCTRL und Der Entfeuchterregler ist der aktive Regler in der Regelsequenz. g_stSeqLinkH[PLT_NUM] ▶ 364].usiCurCtrl = SEQNUM_H_HUMF und Die absolute Zuluftfeuchte hat sich dem Sollwert so weit angenähert, dass sie sich in einem Bereich zwischen rSp - ErrorLimit und rSp + ErrorLimit eingependelt hat. und Die absolute Zuluftfeuchte muss mindestens für die Dauer von EnEvtEn_udiTiSttDly innerhalb des Bereiches von rSp - ErrorLimit und rSp + ErrorLimit verblieben sein.</p> <p>2. Der Timer EnEvtEn_udiTiSttAuto ist abgelaufen und die Regelung hat ihren Sollwertbereich nicht erreicht.</p>
	F_BA_EvtTrans	<p>schreibt auf die BACnetEventTransitionBits to_offnormal, to_fault und to_normal des Loop-Objekts. Dazu muss der Eingang bEnEvtEn = TRUE sein</p>
PID	<p>FB_BACnetLoopSeq1 202 ▶ 103</p>	<p>Sequenzregler Befeuchten.</p>
rtrigSync	R_TRIG	<p>Bei einer steigenden Flanke am Eingang bSync wird das Loop-Objekt auf den Wert von IrSync aufsynchronisiert. Wenn das Regelventil des Kühlers durch Schreiben einer höheren Priorität auf das zugehörige AO-Objekt von der MBE oder durch Betätigen der örtlichen Vorrangbedienung übersteuert wurde, weicht die aktuelle Position des Regelventils von dem Ausgang des Loop-Objekts ab. Mit den Variablen bSync und rFdb kann die Synchronität zwischen der Position des Regelventils und dem Regler wieder hergestellt werden.</p>
Y	<p>FB_BACnetAVSetpoint ▶ 70</p>	<p>Dass AV-Objekt ist referenziert auf den Stellgrößen Ausgang des BACnet-Loop-Objekts</p>

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.14 BAC_AC_SteamGenerator_01_xx

Funktionsbeschreibung

Das Template **BAC_AC_SteamGenerator_01_xx** ist für die Ansteuerung eines Dampferzeugers mit binären und analogen Ein- und Ausgängen. Es besteht im Wesentlichen aus jeweils einem AO- und BO-Objekt für die Ansteuerung des Dampfgenerators und einem MV-Objekt zur manuellen Steuerung und dem dazu gehörigen AV-Objekt zur Eingabe des Stellwertes. Das Template wird ergänzt durch optionale BACnet Objekte, siehe Tabelle Ausstattungsvarianten.

Unter **IO-Verknüpfung** sind die Variablen zu finden, die mit dem Prozessabbild der Eingangs- und Ausgangsebene in der SPS verknüpft werden.



Die beiden Ausgangsvariablen rPrVal / bSync sind nur dann aktiv, wenn in dem eingesetzten Template die mechanische Vorrangbedienung FdbOutAO vorhanden ist. Ist dieses nicht der Fall, so geben die beiden Variablen den Wert Null aus.

Ausstattungsvarianten

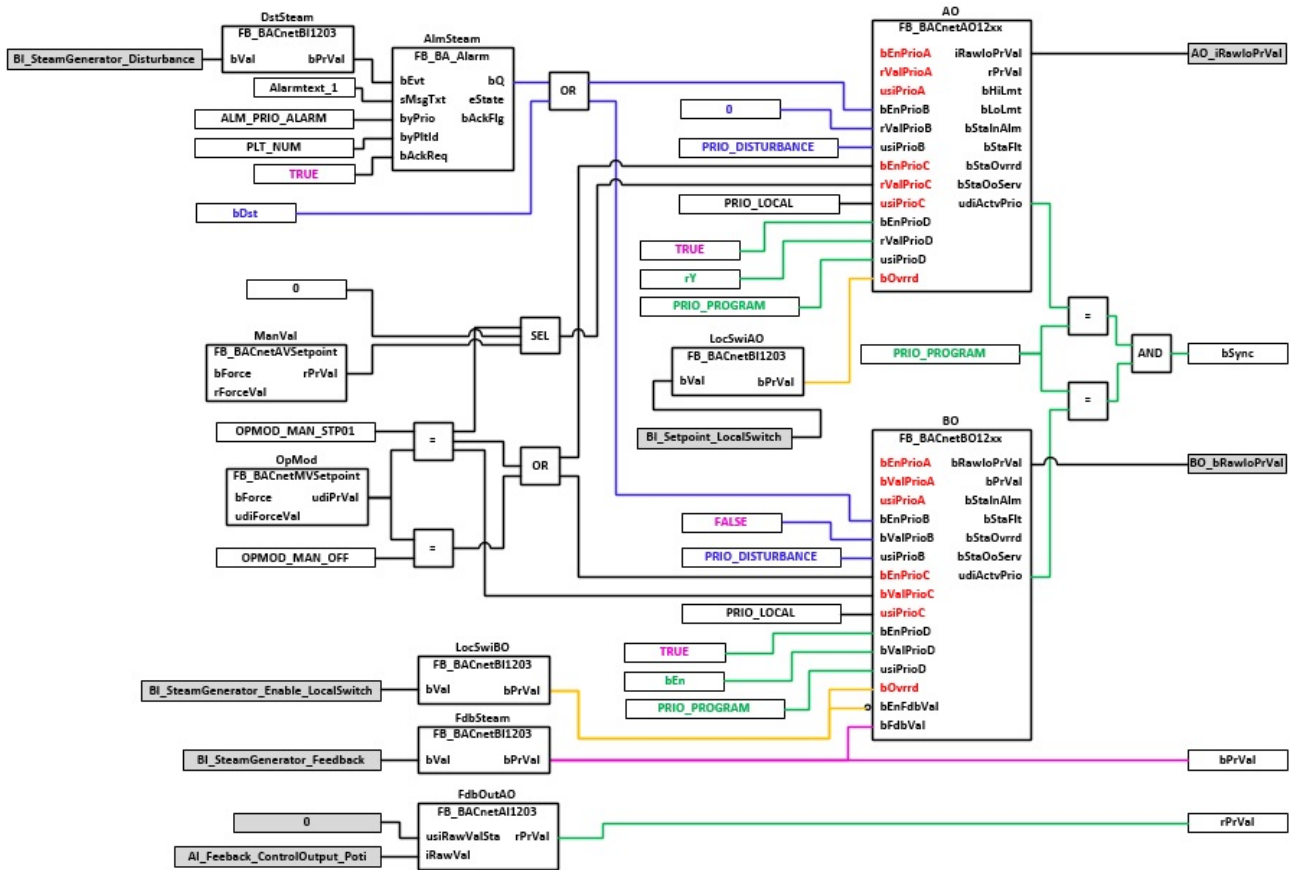
Das Template **BAC_AC_SteamGenerator_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten des Templates werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

Optionen	mechani- sche Vor- rangbedie- nung Freiga- be Dampfbfeuc- hter Rück- meldung Handschal- ter (Frg A-0-H)	mechani- sche Vor- rangbedie- nung Freiga- be Dampfbfeuc- hter Rück- meldung Re- lais (Rm-Ausg)	mechani- sche Vor- rangbedie- nung Soll- wert Dampfbfeuc- hter Rück- meldung Handschal- ter (Rm-A-H)	mechani- sche Vor- rangbedie- nung Stell- wert Dampfbfeuc- hter Rück- meldung Stellung Poti (Rm-Poti)	Betriebs- rückmel- dung Dampfbfe- uchter (Betr-Dampf- Gen)	Störmel- dung Dampfbfe- uchter (Stör-Dampf- Gen)
Instanz	LocSwiBO	FdbOutBO	LocSwiAO	FdbOutAO	FdbSteam	DstSteam
Datenpunkt Typ	BI	BI	BI	AI	BI	BI
	32	16	8	4	2	1
BAC_AC_Ste- amGenerator _01_03	0	0	0	0	1	1
BAC_AC_Ste- amGenerator _01_47	1	0	1	1	1	1

Schnittstelle

BAC_AC_SteamGenerator_01_xx	
—bEn	bPrVal
—rY	rPrVal
—bDst	bSync

Blockschaltbild Variante BAC_AC_SteamGenerator_01_47



VAR_INPUT

bEn : BOOL;
 rY : REAL;
 bDst : BOOL;

bEn: Freigabe des Dampferzeugers; Programmpriorität

rY: Stellwert des Dampferzeugers

bDst: Eingang externe Störung des Dampferzeugers.

VAR_OUTPUT

bPrVal : BOOL;
 rPrVal : REAL;
 bSync : BOOL;

bPrVal: Betriebsrückmeldung Dampferzeugers

rPrVal : Aktueller Stellwert des Dampferzeugers

bSync: Ausgabe eines Impulses um den zum Dampferzeuger zugehörigen Regler beim Rücksetzen vom Hand- in den Automatikbetrieb auf die aktuelle Stellgröße des Dampfgenerators zu synchronisieren. Der Synchronisationsimpuls **bSync** sollte nur dann eingesetzt werden, wenn in dem eingesetzten Template die mechanische Vorrangbedingung **FdbOutAO** vorhanden ist.

VAR CONSTANT

PLT_NUM : BYTE := 1;

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB BA Alarm**. [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt.](#) [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe			
DstSteam	FB_BACnetBI1203 [► 72]	X	BI-Objekt Störung Dampfgenerator			
LocSwiBO	FB_BACnetBI1203 [► 72]	X	BI-Objekt mechanische Vorrangbedienung Rückmeldung Handschalter binär			
LocSwiAO	FB_BACnetBI1203 [► 72]	X	BI-Objekt mechanische Vorrangbedienung Rückmeldung Handschalter analog			
FdbSteam	FB_BACnetBI1203 [► 72]	X	BI-Objekt Betriebsrückmeldung Dampfgenerator			
FdbOutAO	FB_BACnetAI1203 [► 49]	X	Rückgabewert Mechanische Vorrangbedienung Stellwert Dampfgenerator			
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung des Dampfgenerators von der MBE oder einem lokalen Bediendisplay			
ManVal	FB_BACnetAVSetpoint [► 70]		AV-Objekt zur Eingabe des Stellwertes des Dampfgenerators bei manueller Übersteuerung			
AlmSteam	FB_BA_Alarm [► 182]	x	Erfassung und Weiterverarbeitung eines Fehlers des Dampfgenerators			
AO	FB_BACnetAO1203 [► 53]		AO-Objekt für die Vorgabe des Drehzahlsollwertes an den Dampfgenerator			
			Priorität:	Freigabe	Wert	Bemerkung
			PRIO_SAFETY (1)	FALSE	0	
			PRIO_DISTURBAN CE (3)	Eingang bDst OR DstSteam	0	
			PRIO_LOCAL (8)	OpMod_udiPrVal = OPMOD_MAN_OFF OPMOD_MAN_OFF	Selecto r 0 ORMan Val_rPr Val	Bei Handbetrieb Wert von AV-Objekt ManVal
PRIO_PROGRAM (15)	TRUE	rY	Wert von Eingang rY (z. B. Stellgröße vom Regler)			
BO	FB_BACnetBO1203 [► 82]		BO-Objekt für die Vorgabe der Freigabe an den Frequenzumrichter			
			Priorität:	Freigabe	Wert	Bemerkung
			PRIO_SAFETY (1)	FALSE	FALSE	

Instanz	Typ	optional	Aufgabe			
			PRIO_DISTURBANCE (3)	Eingang bDst OR DstSteam	FALSE	
			PRIO_LOCAL (8)	OpMod_udiPrVal = OPMOD_MAN_OFFFOROPMOD_MAN_STP01	TRUE, wenn OpMod_udiPrVal = OPMOD_MAN_STP01	Bei Handbetrieb Wert von AV-Objekt ManVal
			PRIO_PROGRAM (15)	TRUE	bEn	Wert von Eingang bEn
	EQ, EQ, AND		Wert des Netzwerkes ist TRUE, wenn die Aktive Priorität PRIO_PROGRAM (15) ist. Kann zur Synchronisation des Reglers bei Rückkehr in den Automatikbetrieb benutzt werden			
TLogAO	FB_BACnetTLog1201 [► 137]		Trendaufzeichnung des AO-Objekts für die Vorgabe des Stellwertes an den Dampfgenerator			

IO-Verknüpfung

In der zum Template gehörigen XML-Description sind in dem Bereich **Parameter** Variablen mit der Kennung **INPUT** oder **Output** deklariert. Diese Parameter können im Project Builder oder über die Excel-Import-Schnittstelle mit dem Prozessabbild der Eingangs- und Ausgangsebene in der SPS verknüpft werden.

Parameter	Typ	Optional	Prozessabbild	
BI_SteamGenerator_Disturbance	BOOL		Eingang	Digitaleingang - Dampfgenerator Störung - Meldung - ausgelöst
BI_SteamGenerator_Enable_LocalSwitch	BOOL	x	Eingang	Digitaleingang - Schalter Hand Freigabe Dampfgenerator - Meldung - Hand/ Auto
BI_Setpoint_LocalSwitch	BOOL	x	Eingang	Digitaleingang - Schalter Stellwert - Meldung - Hand/ Auto
BI_SteamGenerator_Feedback	BOOL		Eingang	Digitaleingang - Dampfgenerator - Meldung - Betrieb
AI_Feedback_ControlOutput_Poti	INT	x	Eingang	Analogeingang - Hand Potentiometer – Rückmeldung - Stellausgang
AO_SteamGenerator_ControlCommand	INT		Ausgang	Analogausgang - Dampfgenerator Stellbefehl
BO_SteamGenerator_Enable	BOOL		Ausgang	Digitalausgang - Dampfgenerator Schaltbefehl - Freigabe Ein/Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.15 BAC_AC_FireDmp_01_xx

Funktionsbeschreibung

Das Template **BAC_AC_FireDmp_01_xx** ist für die Ansteuerung und Überwachung einer motorischen Brandschutzklappe mit binären Schaltausgang.



Die beiden Ausgangsvariablen bSwiOpn / bSwiCls geben nur dann den tatsächlichen Zustand der Brandschutzklappe aus, wenn in dem eingesetzten Template die Rückmeldung der Endlagenschalter SwiOpn/SwiCls vorhanden sind. Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang bSwiOpn = TRUE. Bei nicht angesteuerter Klappe wird der Ausgang bSwiCls = TRUE. Dieses kann zur Folge haben, dass in dem Startprogramm einer Lüftungsanlage eine Verzögerungszeit zum Auffahren der Brandschutzklappen angegeben werden muss.

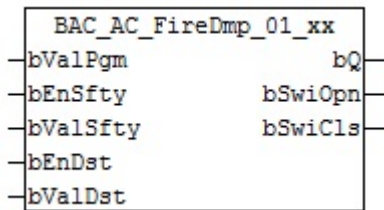
Ausstattungsvarianten

Das Template **BAC_AC_FireDmp_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten der Klappen werden nach einer Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

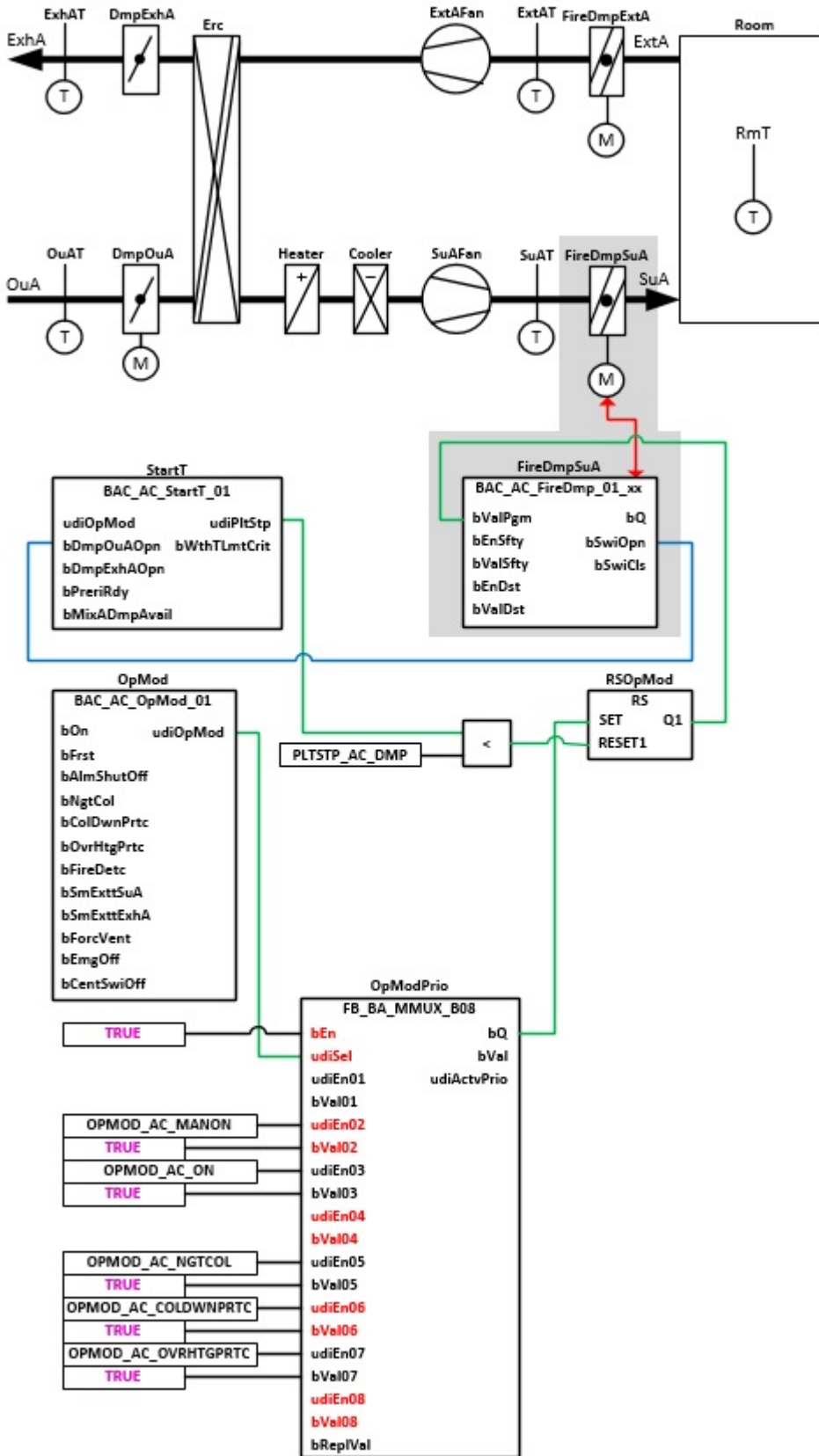
Optionen	Meldung thermische Auslösung (ThermAuslösung)	mechanische Vorrangbedien- ung Rückmeldung Handscharter (A-0-H)	mechanische Vorrangbedien- ung Rückmeldung Relaisausgang (Rm-Ausg)	Endlage Auf (SchalterAuf)	Endlage Zu (SchalterZu)
Instanz	ThTrigger	LocSwi	FdbOut	SwiOpn	SwiCls
Datenpunkt Typ	BI	BI	BI	BI	BI
	16	8	4	2	1
BAC_AC_FireDmp_01_001	0	0	0	0	1
BAC_AC_FireDmp_01_002	0	0	0	1	0
BAC_AC_FireDmp_01_003	0	0	0	1	1
BAC_AC_FireDmp_01_013	0	1	1	0	1
BAC_AC_FireDmp_01_014	0	1	1	1	0
BAC_AC_FireDmp_01_015	0	1	1	1	1
BAC_AC_FireDmp_01_017	1	0	0	0	1
BAC_AC_FireDmp_01_018	1	0	0	1	0
BAC_AC_FireDmp_01_019	1	0	0	1	1
BAC_AC_FireDmp_01_029	1	1	1	0	1
BAC_AC_FireDmp_01_030	1	1	1	1	0

Optionen	Meldung thermische Auslösung (ThermAuslösung)	mechanische Vorrangbedien- ung Rückmeldung Handschal- ter (A-0-H)	mechanische Vorrangbedie- nung Rückmeldung Relaisausgang (Rm-Ausg)	Endlage Auf (SchalterAuf)	Endlage Zu (SchalterZu)
Instanz	ThTrigger	LocSwi	FdbOut	SwiOpn	SwiCls
Datenpunkt Typ	BI	BI	BI	BI	BI
	16	8	4	2	1
BAC_AC_FireD mp_01_031	1	1	1	1	1

Schnittstelle



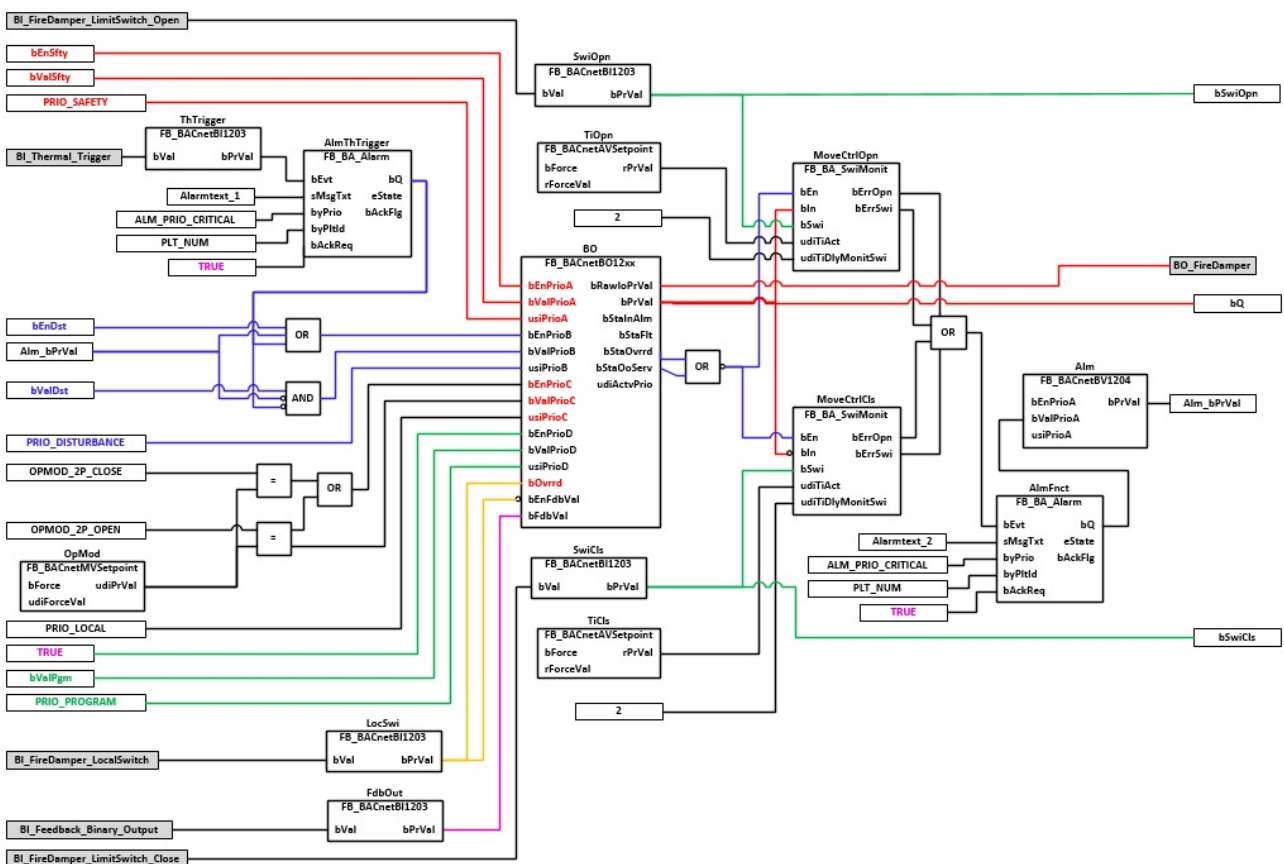
Anlagenschema 01



Anlagenschema 02



Blockschaltbild Variante BAC_AC_FireDmp_01_031



VAR_INPUT

```

bValPgm      : BOOL;
bEnSfty      : BOOL;
bValSfty     : BOOL;
bEnDst       : BOOL;
bValDst      : BOOL;
    
```

bValPgm: Binärer Wert Programmpriorität

bEnfty: Freigabe Sicherheitspriorität

bValSfty: Binärer Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität

bValDst: Binärer Wert Störungspriorität

VAR_OUTPUT

```
bQ      : BOOL;
bSwiOpn : BOOL;
bSwiCls  : BOOL;
```

bQ: Status Ausgang Ansteuerung der Brandschutzklappe

bSwiOpn: Endlage Offen der Brandschutzklappe ist erreicht. Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang **bSwiOpn** = TRUE. Bei nicht angesteuerter Klappe wird der Ausgang **bSwiCls** = TRUE.

bSwiCls: Endlage Zu der Brandschutzklappe ist erreicht. Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang **bSwiOpn** = TRUE. Bei nicht angesteuerter Klappe wird der Ausgang **bSwiCls** = TRUE.

VAR CONSTANT

```
PLT_NUM : BYTE := 10;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [► 182]

Die Auswertung der Alarme einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
SwiOpn	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss des Endlagenschalters Offen
SwiCls	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss des Endlagenschalters Zu
ThTrigger	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss der thermischen Auslösung, z. B. des Schmelzlots
AlmThTrigger	FB_BA_Alarm [► 182]	x	Der Funktionsbaustein AlmThTrigger erfasst das Ereignis der thermischen Auslösung, z. B. des Schmelzlots. Aktionen, die nach dem Eingang der thermischen Auslösung erfolgen soll, können im Template am Funktionsbaustein AlmThTrigger parametrisiert werden.
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung der Brandschutzklappe von der MBE oder einem lokalen Bediendisplays
LocSwi	FB_BACnetBI1203 [► 72]	X	BI-Objekt für für die Rückmeldung einer mechanischen Vorrangbedienung. (Hand/Not-Bedienebene)
FdbOut	FB_BACnetBI1203 [► 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Relais
TiOpn	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe des Wertes für die Auffahrzeit

Instanz	Typ	optional	Aufgabe		
TiCls	FB_BACnetAVSetpoint [70]	X	AV-Objekt zur Eingabe des Wertes für die Zufahrzeit		
MoveCtrlOpn	FB_BA_SwiMonit [155]	X	Funktionsbaustein welcher die Endlage Auf der Klappe überwacht		
MoveCtrlCls	FB_BA_SwiMonit [155]	X	Funktionsbaustein welcher die Endlage Zu der Klappe überwacht		
BO	FB_BACnetBO1203 [82]		BO-Objekt für die Ansteuerung der Brandschutzklappe		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty
			PRIO_DISTURBANCE (3)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Störung (Disturbance) des nachgeschalteten BO-Objekts aktivieren. Ereignisse: 1. Eingang Template bEnDst 2. Endlagenschalterstörung vom Funktionsbaustein AlmFct 3. Thermische Auslösung Schmelzlot vom Funktionsbaustein AlmThTrigger	Am Eingang des Templates bEnDst lässt sich die Brandschutzklappe zwangsweise Öffnen oder Schließen, z. B. bei Entrauchung oder Feueralarm. Das zwangsweise Einschalten ist jedoch am AND-Modul verriegelt bei vorhandener thermischer Auslösung des Schmelzlot oder einer Endlagenschalterstörung.
			PRIO_LOCAL (8)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt hat den Wert OPMOD_2P_CLOSE (Brandschutzklappe schließen) 2. Das MV-Objekt hat den Wert OPMOD_2P_OPEN (Brandschutzklappe öffnen)	TRUE, wenn OpMod_udiPrVal = OPMOD_2P_OPEN
PRIO_PROGRAM (15)	TRUE	Eingang bValPgm			
AlmFct	FB_BA_Alarm [182]	x	Der Funktionsbaustein AlmFct erfasst das Ereignis der Überwachung der Endlagenschalter. Aktionen, die nach dem Eingang der Endlagenschalterstörung erfolgen soll, können im Template am Funktionsbaustein AlmFct parametrieren werden.		

Instanz	Typ	optional	Aufgabe
Alm	FB_BACnetBV1204 [► 94]	x	BV-Objekt zur Anzeige der Klappenstörung in der MBE

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_FireDamper_Li mitSwitch_Open	BOOL	X	Eingang	Digitaleingang - Schalter Brandschutzklappe Auf - Meldung - Betätigt/Nicht betätigt
BI_FireDamper_Li mitSwitch_Close	BOOL	X	Eingang	Digitaleingang - Schalter Brandschutzklappe Zu - Meldung - Betätigt/Nicht betätigt
BI_FireDamper_Lo calSwitch	BOOL	X	Eingang	Digitaleingang - Schalter Hand Brandschutzklappe - Meldung - Hand/ Auto
BI_Feedback_Bina ry_Output	BOOL	X	Eingang	Digitaleingang - Brandschutzklappe Schaltbefehl - Rückmeldung - Ein/ Aus
BI_Thermal_Trige r	BOOL	X	Eingang	Digitaleingang - thermische Auslösung Schmelzlot - Meldung - ausgelöst
BO_FireDamper	BOOL		Ausgang	Digitalausgang - Brandschutzklappe - Schaltbefehl - Ein/ Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.16 BAC_AC_ErcPI_01

Anwendung

Das Template BAC_AC_ErcPI_01 dient der Steuerung, Regelung und dem Vereisungsschutz einer Energierückgewinnung mit Plattenwärmetauscher.

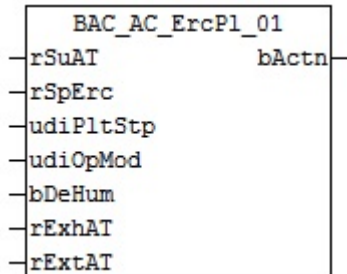
Innerhalb des Aufruftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

- **ErcSuA_PID**: Zulufttemperaturregelung
- **ErcExhA_PID**: Minimalbegrenzung der Fortlufttemperatur

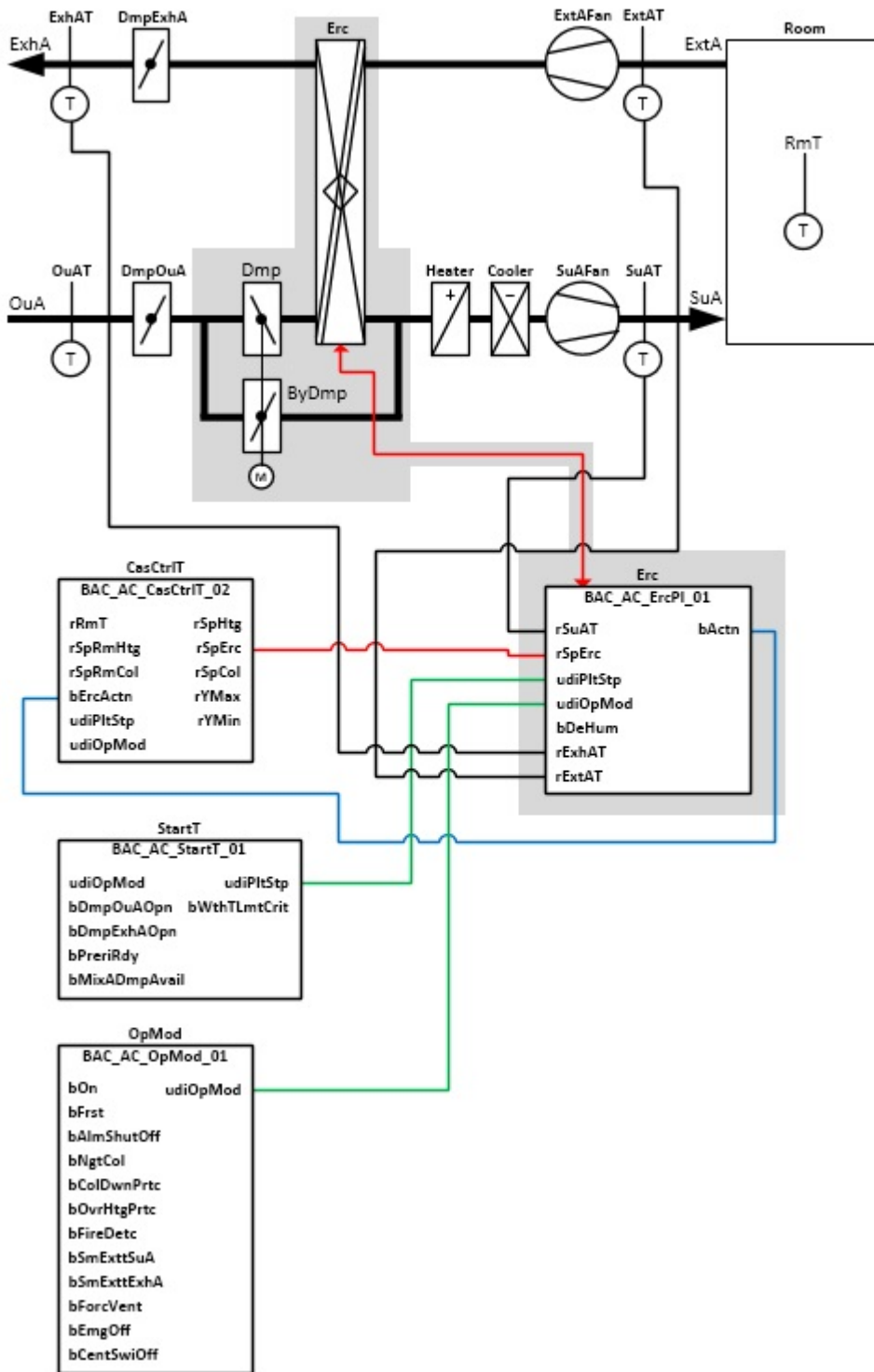
- **ErcIcPrt**: Vereisungsschutz des Wärmetauschers mittels eines Differenzdruckschalters
- **ErcDmp**: Ansteuerung eines Bypass-Klappensystems

Der Fortluftminimalbegrenzer **ErcExhA_PID** und das Frostschutzprogramm **ErcIcPrt** begrenzen die Stellgröße des Zulufttemperaturreglers **ErcSuA_PID** der Energierückgewinnung über die Minimalauswahl, um einer Vereisung des Wärmetauschers vorzubeugen.

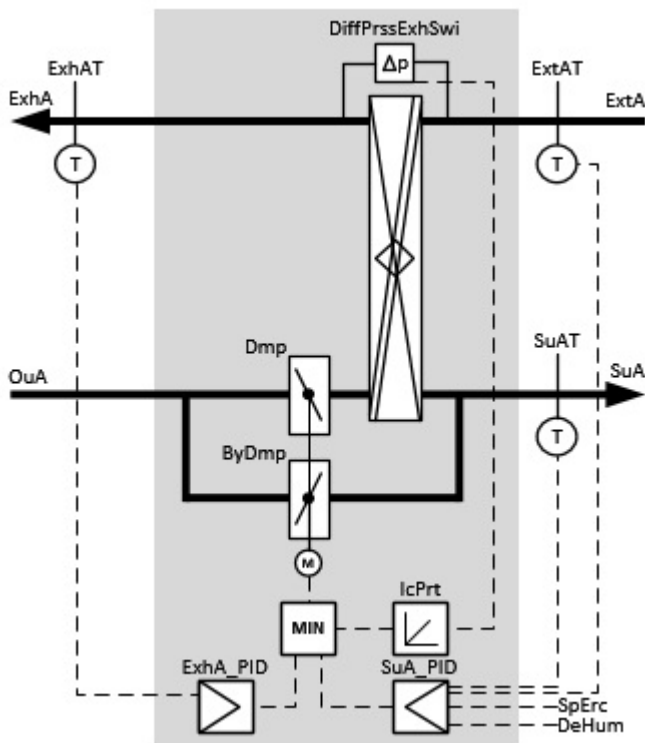
Schnittstelle



Anlagenschema 01

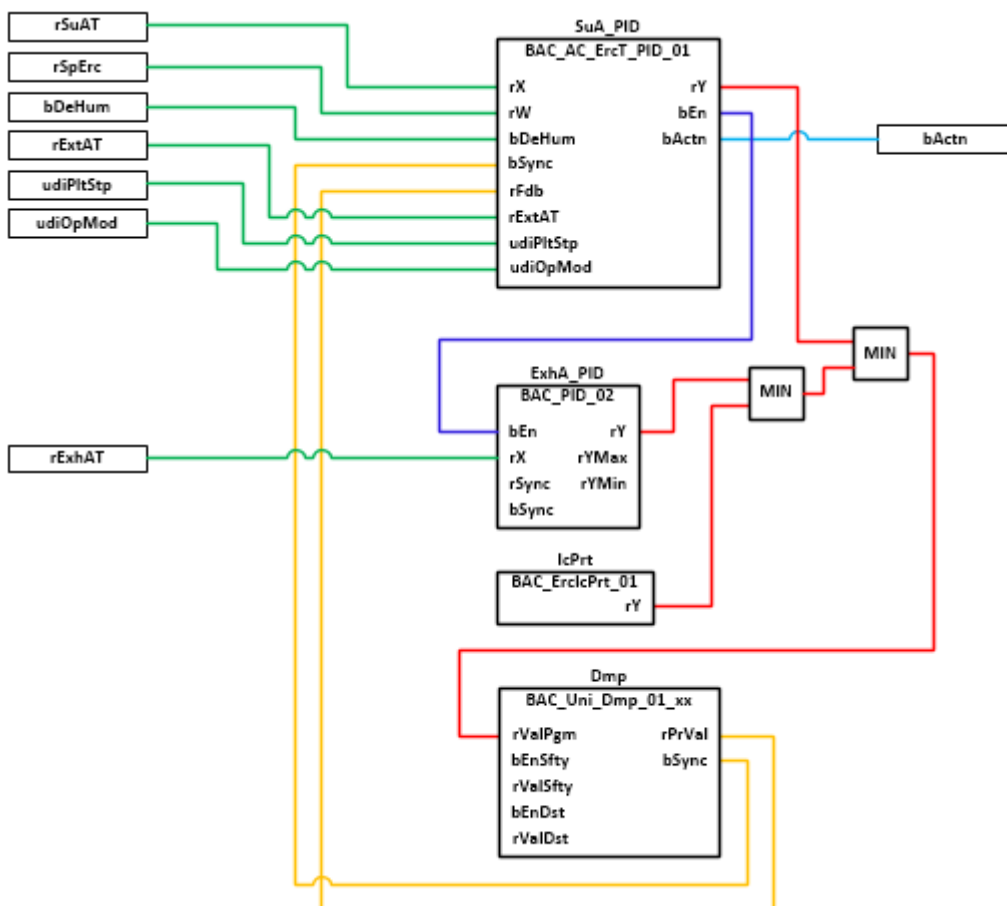


Anlagenschema 02



Die Ausgabe des Stellsignals zur Leistungsregelung der Energierückgewinnung bezieht sich auf die Klappe **Dmp** und nicht auf die Klappe **ByDmp** (Siehe Grafik). Bei der Inbetriebnahme der RLT-Anlage ist darauf zu achten, dass Klappe **Dmp** mit einem direkten Wirksinn und die Klappe **ByDmp** mit einen invertierten Wirksinn betrieben wird.

Blockschaltbild



VAR_INPUT

```

rSuAT      : REAL;
rSpErc     : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
bDeHum     : BOOL;
rExhAT     : REAL;
rExtAT     : REAL;

```

rSuAT: Messwert Zulufttemperatur

rSpErc: Sollwert Zulufttemperatur

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC AC StartT 01](#) [[▶ 538](#)].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC AC OpMod 01](#) [[▶ 524](#)].

bDeHum: Betriebsmeldung Entfeuchtebetrieb. Sperrung der Energierückgewinnung im Heizbetrieb (Außentemperatur > Ablufttemperatur) .

rExhAT: Messwert Fortlufttemperatur

rExtAT: Messwert Ablufttemperatur

VAR_OUTPUT

```

bActn      : BOOL;

```

bActn: Wirksinn der Zulufttemperaturregelung **SuA_PID**. Der Wirksinn der Energierückgewinnung wird von verschiedenen Sollwertprogrammen oder Kaskadenreglern zur Bestimmung des Sollwertes für die Zulufttemperatur der Energierückgewinnung benötigt, z. B. [BAC AC SpSuAT 02](#) [[▶ 681](#)], [BAC AC CasCtrlT 02](#) [[▶ 653](#)].

Programmbeschreibung

Instanz	Typ	Aufgabe
SuA_PID	BAC AC ErcT PID 01 [▶ 420]	Sub-Template Zulufttemperaturregelung. Die Temperaturregelung erfolgt stetig mittels eines PID-Sequenzreglers
ExhA_PID	BAC PID 02 [▶ 599]	Sub-Template Fortlufttemperaturregelung. Die Regelung dient dem Vereisungsschutz und soll als Prävention dienen, bevor die Rampenfunktion lcPrt aktiv wird.
lcPrt	BAC ErclcPrt 01 [▶ 426]	Sub-Template Vereisungsschutz mittels Differenzdruckschalter
Dmp	BAC Uni Dmp 01_05 [▶ 568]	Sub-Template zur Ansteuerung einer stetigen Klappe
	MIN, MIN	MIN-Auswahl der Stellsignale Zulufttemperaturregelung <i>SuA_PID</i> , Fortlufttemperaturregelung <i>ExhA_PID</i> und der Rampenfunktion Frostschutz

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.17 BAC_AC_ErcPI_02**Anwendung**

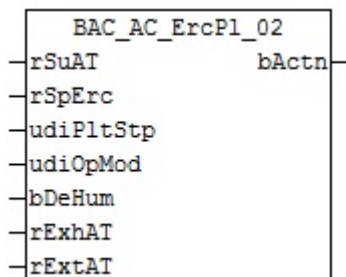
Das Template [BAC_AC_ErcPI_02](#) dient der Steuerung, Regelung und dem Vereisungsschutz einer Energierückgewinnung mit Plattenwärmetauscher.

Innerhalb des Aufrufftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

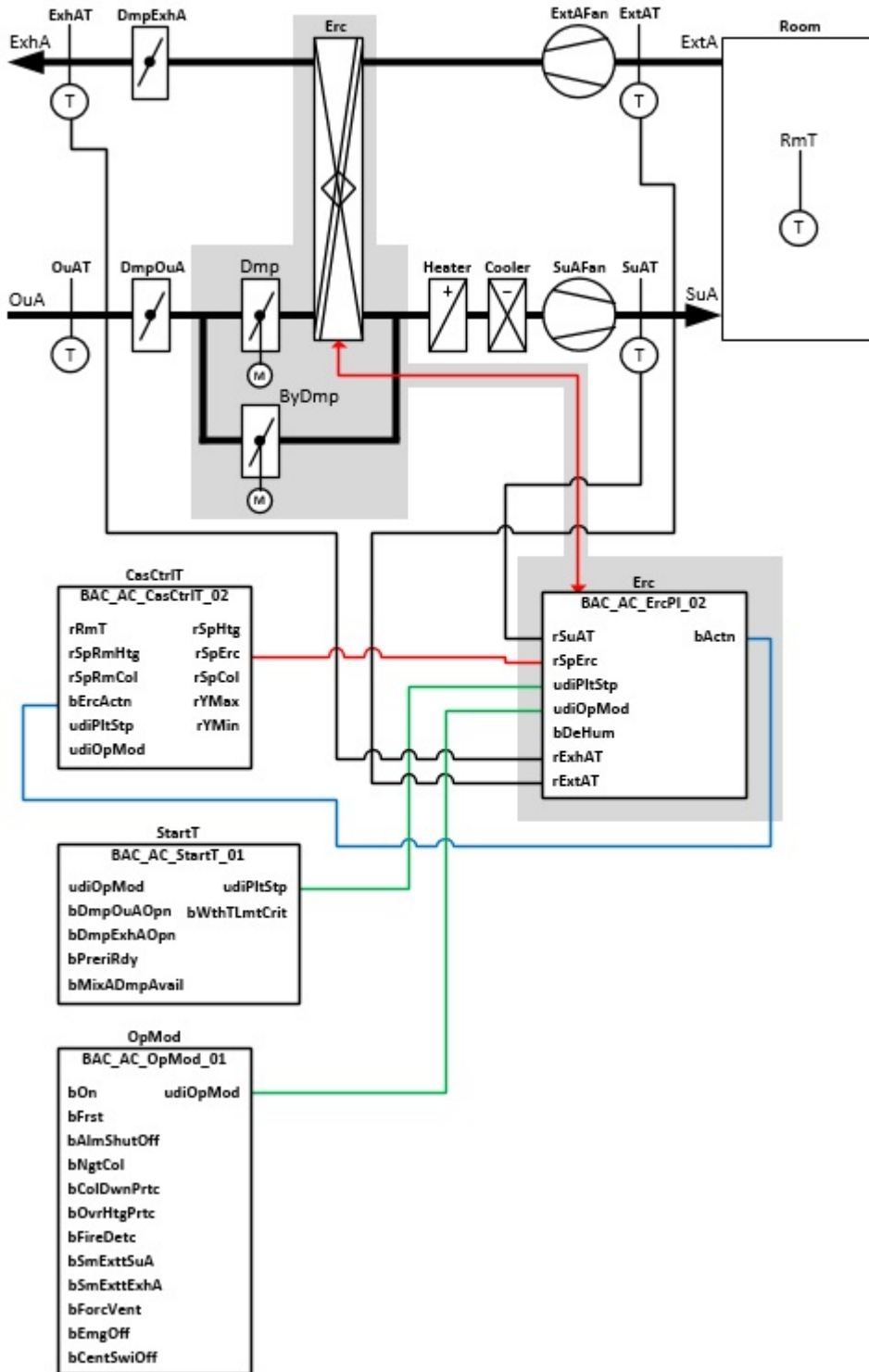
- **ErcSuA_PID**: Zulufttemperaturregelung
- **ErcExhA_PID**: Minimalbegrenzung der Fortlufttemperatur
- **ErcIcPrt**: Vereisungsschutz des Wärmetauschers mittels eines Differenzdruckschalters
- **ErcDmp**: Ansteuerung einer Klappe
- **ErcByDmp**: Ansteuerung einer Bypass-Klappe

Der Fortluftminimalbegrenzer **ErcExhA_PID** und das Frostschutzprogramm **ErcIcPrt** begrenzen die Stellgröße des Zulufttemperaturreglers **ErcSuA_PID** der Energierückgewinnung über die Minimalauswahl, um einer Vereisung des Wärmetauschers vorzubeugen.

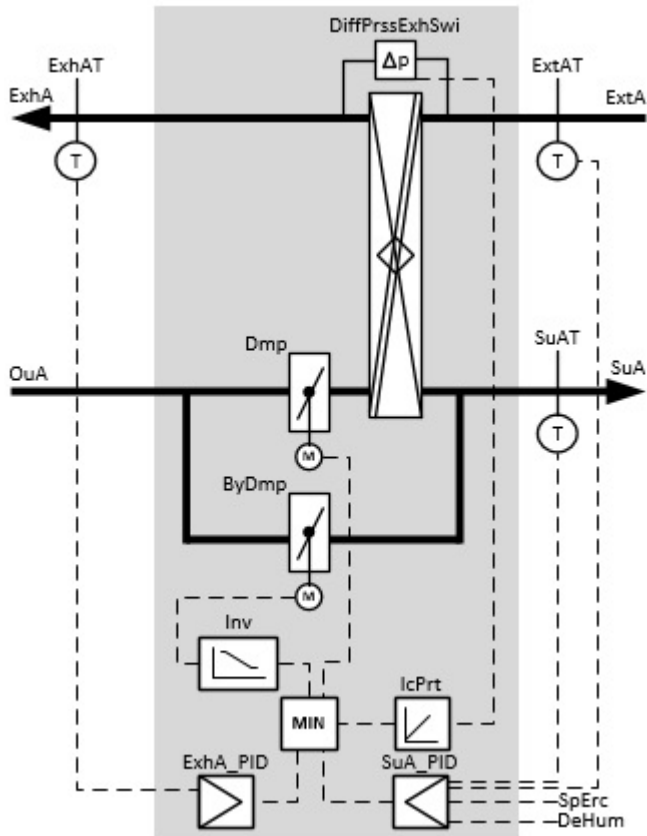
Schnittstelle



Anlagenschema 01

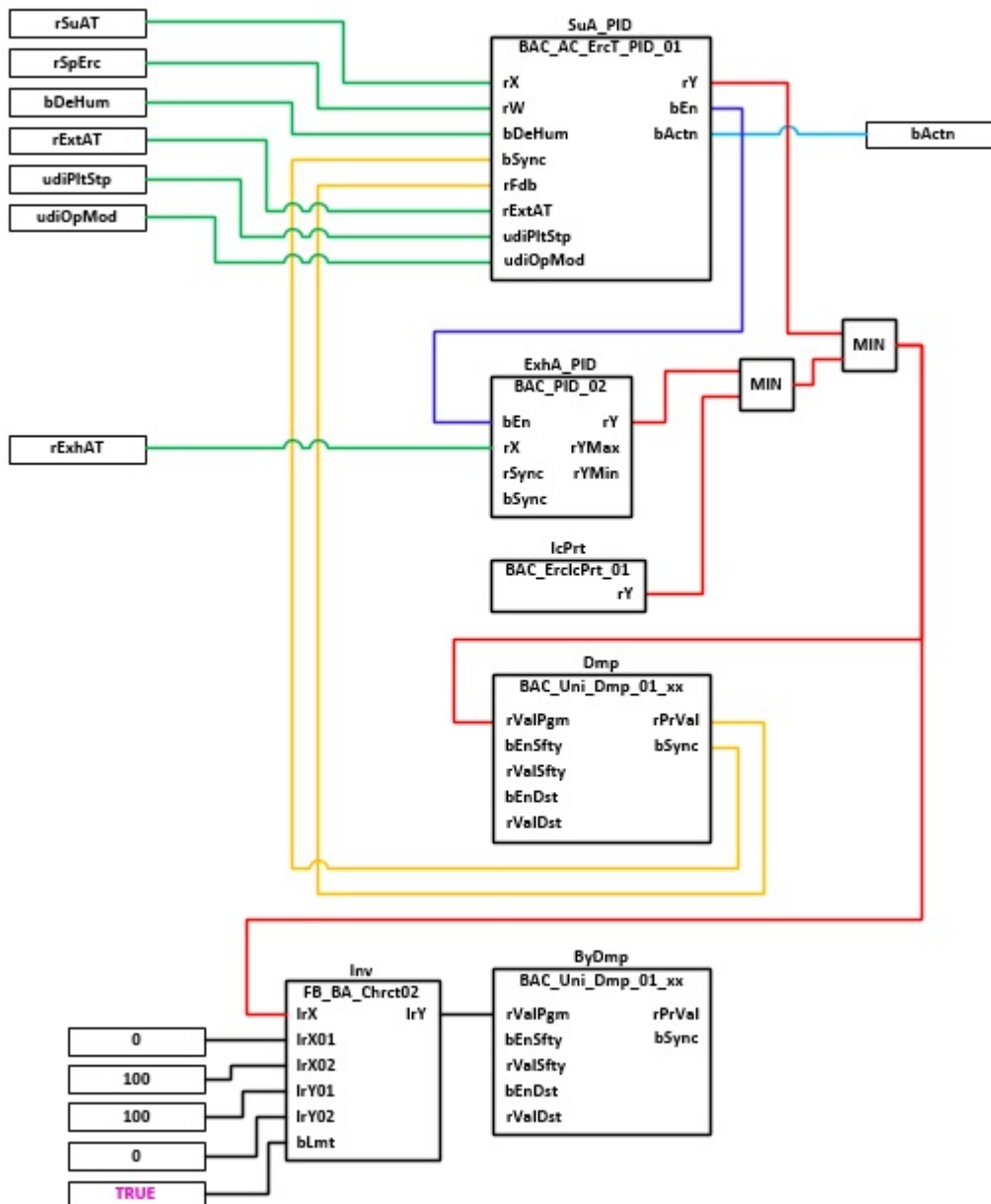


Anlagenschema 02



Die Ausgabe des Stellsignals zur Leistungsregelung der Energierückgewinnung bezieht sich auf die Klappe **Dmp** (Siehe Grafik). Das Stellsignal der Bypass-Klappe **ByDmp** wird invertiert. Bei der Inbetriebnahme der RLT-Anlage ist darauf zu achten, dass beide Klappen mit einem direkten Wirksinn betrieben werden.

Blockschaltbild



VAR_INPUT

```

rSuAT      : REAL;
rSpErc     : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
bDeHum     : BOOL;
rExhAT     : REAL;
rExtAT     : REAL;

```

rSuAT: Messwert Zulufttemperatur

rSpErc: Sollwert Zulufttemperatur

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm BAC AC StartT 01 [► 538].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl BAC AC OpMod 01 [► 524].

bDeHum: Betriebsmeldung Entfeuchtebetrieb. Sperrung der Energierückgewinnung im Heizbetrieb (Außentemperatur > Ablufttemperatur) .

rExhAT: Messwert Fortlufttemperatur

rExtAT: Messwert Ablufttemperatur

VAR_OUTPUT

bActn : BOOL;

bActn: Wirksinn der Zulufttemperaturregelung **SuA_PID**. Der Wirksinn der Energierückgewinnung wird von verschiedenen Sollwertprogrammen oder Kaskadenreglern zur Bestimmung des Sollwertes für die Zulufttemperatur der Energierückgewinnung benötigt, z. B. [BAC_AC_SpSuAT_02](#) [▶ 681], [BAC_AC_CasCtrlT_02](#) [▶ 653].

Programmbeschreibung

Instanz	Typ	Aufgabe
SuA_PID	BAC AC ErcT PID_01 [▶ 420]	Sub-Template Zulufttemperaturregelung. Die Temperaturregelung erfolgt stetig mittels eines PID-Sequenzreglers
ExhA_PID	BAC PID_02 [▶ 599]	Sub-Template Fortlufttemperaturregelung. Die Regelung dient dem Vereisungsschutz und soll als Prävention dienen, bevor die Rampenfunktion lcPrt aktiv wird.
lcPrt	BAC ErcIcPrt_01 [▶ 426]	Sub-Template Vereisungsschutz mittels Differenzdruckschalter
Dmp	BAC Uni Dmp_01_05 [▶ 568]	Sub-Template zur Ansteuerung einer stetigen Klappe
Inv	FB BA Chrct02 [▶ 173]	Invertierung des Stellsignals der stetigen Klappe für Ansteuerung der Bypass-Klappe.
ByDmp	BAC Uni Dmp_01_05 [▶ 568]	Sub-Template zur Ansteuerung einer stetigen Bypass-Klappe
	MIN, MIN	MIN-Auswahl der Stellsignale Zulufttemperaturregelung <i>SuA_PID</i> , Fortlufttemperaturregelung <i>ExhA_PID</i> und der Rampenfunktion Frostschutz

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.18 BAC_AC_ErcRecup_01

Anwendung

Das Template **BAC_AC_ErcRecup_01** dient der Steuerung, Regelung und dem Vereisungsschutz einer Energierückgewinnung mit einem Kreislaufverbundsystems.

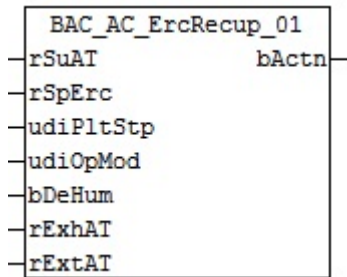
Das Template **BAC_AC_ErcRecup_01** ist ein Aufrufftemplate.

Innerhalb des Aufrufftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

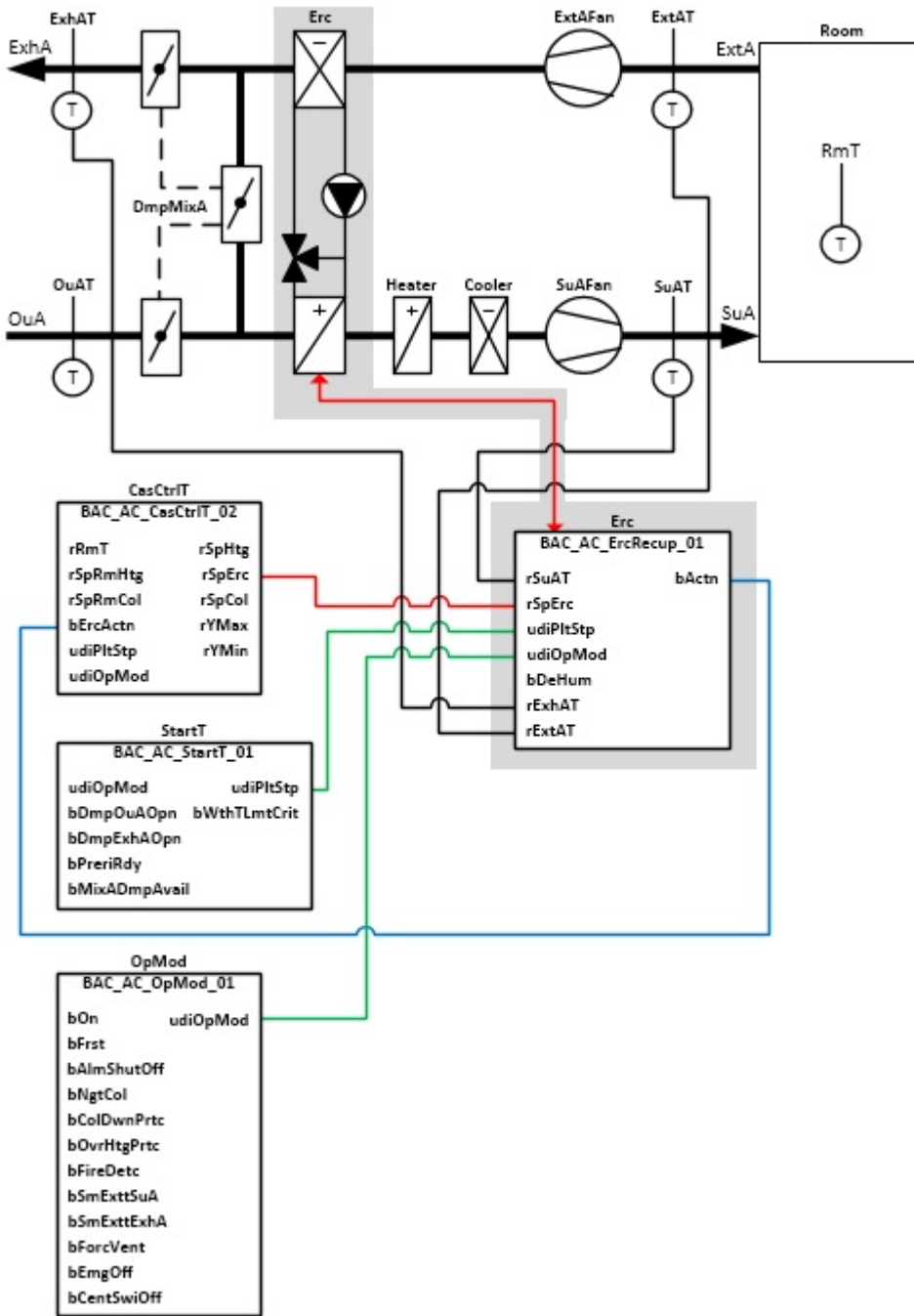
- **ErcRetWtrT:** Rücklafterperatur des Kreislaufverbundsystems
- **ErcSuA_PID:** Regelung der Zulufttemperatur
- **ErcExhA_PID:** Minimalbegrenzung der Fortlufttemperatur
- **ErcPu** Ansteuerung der Zirkulationspumpe
- **ErcRetWtr_PID:** Minimalbegrenzung der Wassertemperatur am Rücklauf des Wärmetauschers in der Fortluft des Kreislaufverbundsystems
- **ErcIcPrt_01:** Vereisungsschutz des Wärmetauschers in der Fortluft des Kreislaufverbundsystems mit Differenzdruckschalter
- **ErcVlv:** Ansteuerung eines stetigen Regelventils im Wasser/Glykol-Kreislauf

Der wasserseitige Rücklauftemperaturregler **ErcRetWtr_PID**, der Fortluftminimalbegrenzer **ErcExhA_PID** und das Frostschutzprogramm **ErcIcPrt** begrenzen die Stellgröße des Zulufttemperaturreglers **ErcSuA_PID** der Energierückgewinnung über die Minimalauswahl um einer Vereisung des Wärmetauschers vorzubeugen.

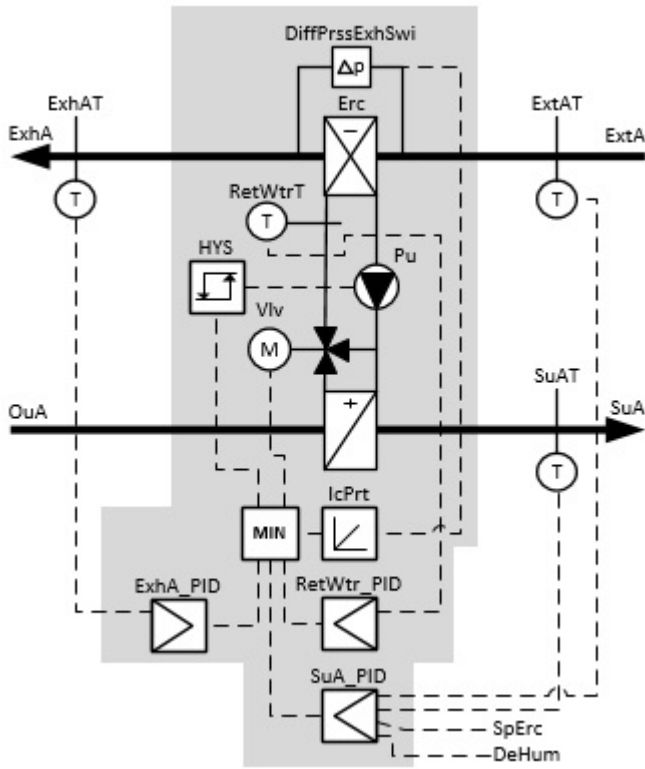
Schnittstelle



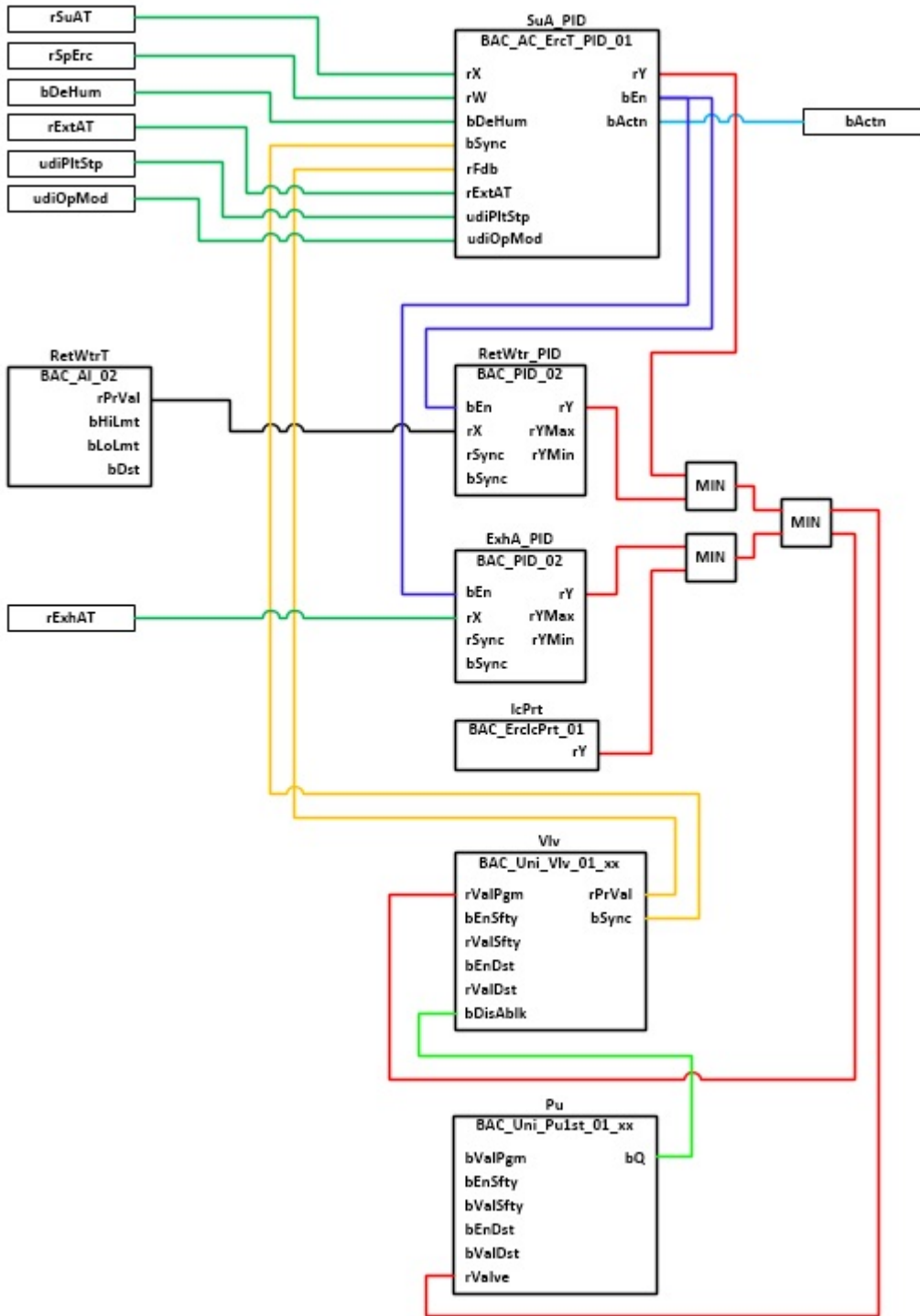
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

rSuAT      : REAL;
rSpErc     : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
bDeHum     : BOOL;
rExhAT     : REAL;
rExtAT     : REAL;
    
```

rSuAT: Messwert Zulufttemperatur

rSpErc: Sollwert Zlufttemperatur

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm BAC AC StartT 01 [► 538].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC_AC_OpMod_01](#) [[▶ 524](#)].

bDeHum: Betriebsmeldung Entfeuchtebetrieb. Sperrung der Energierückgewinnung im Heizbetrieb (Außentemperatur > Ablufttemperatur) .

rExhAT: Messwert Fortlufttemperatur

rExtAT: Messwert Ablufttemperatur

VAR_OUTPUT

bActn : BOOL;

bActn: Wirksinn der Zulufttemperaturregelung **SuA_PID**. Der Wirksinn der Energierückgewinnung wird von verschiedenen Sollwertprogrammen oder Kaskadenreglern zur Bestimmung des Sollwertes für die Zulufttemperatur der Energierückgewinnung benötigt, z. B. [BAC_AC_SpSuAT_02](#) [[▶ 681](#)], [BAC_AC_CasCtrlT_02](#) [[▶ 653](#)].

Programmbeschreibung

Instanz	Typ	Aufgabe
RetWtrT	BAC_AI_02 [▶ 685]	Sub-Template Rücklauftemperatur des Kreislaufverbundsystems
SuA_PID	BAC_AC_ErcT_PID_01 [▶ 420]	Sub-Template Zulufttemperaturregelung. Die Temperaturregelung erfolgt stetig mittels eines PID-Sequenzreglers
ExhA_PID	BAC_PID_02 [▶ 599]	Sub-Template Fortlufttemperaturregelung für den Vereisungsschutz
RetWtrT_PID	BAC_PID_02 [▶ 599]	Sub-Template Rücklauftemperaturregelung für den Vereisungsschutz
lcPrt	BAC_ErcIcPrt_01 [▶ 426]	Sub-Template für den Vereisungsschutz einer Energierückgewinnung mittels Differenzdruckschalter
Vlv	BAC_Uni_Vlv_01_13 [▶ 608]	Template zur Ansteuerung eines stetigen Ventils
Pu	BAC_Uni_Pu1st_01_189 [▶ 579]	Template zur Ansteuerung einer einstufigen Pumpe
	MIN, MIN, MIN	MIN-Auswahl der Stellsignale Zulufttemperaturregelung SuA_PID , Fortlufttemperaturregelung ExhA_PID , Rücklauftemperaturregelung RetWtrT_PID und der Rampenfunktion Frostschutz

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.19 BAC_AC_ErcRot_01

Anwendung

Das Template **BAC_AC_ErcRot_01** dient zur Steuerung, Regelung und dem Vereisungsschutz einer Energierückgewinnung mit Rotationswärmetauscher.

Das Template **BAC_AC_ErcRot_01** ist ein Aufruftemplate.

Innerhalb des Aufruftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

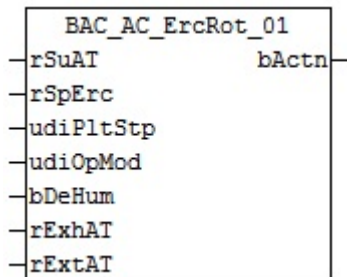
- **ErcSuA_PID:** Regelung der Zulufttemperatur
- **ErcExhA_PID:** Minimalbegrenzung der Fortlufttemperatur
- **ErcMot:** Ansteuerung WRG-Steuergerätes
- **ErcIcPrt:** Vereisungsschutz des Wärmetauschers mittels Differenzdruckschalter

- **ErcByDmp**: Ansteuerung einer Bypass-Klappe

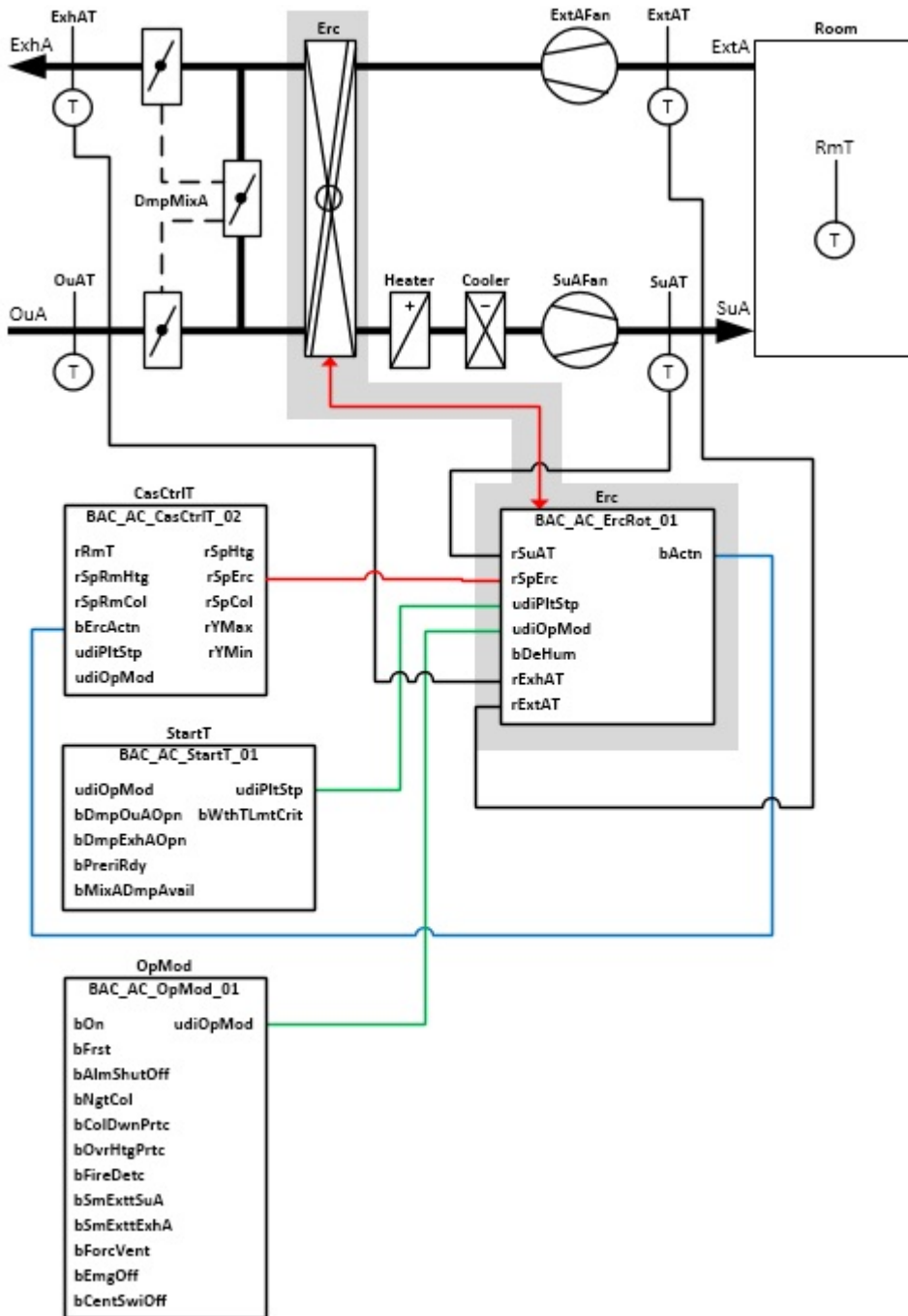
Der Fortluftminimalbegrenzer **ErcExhA_PID** und das Frostschutzprogramm **ErcIcPrt** begrenzen die Stellgröße des Zulufttemperaturreglers **ErcSuA_PID** der Energierückgewinnung über die Minimalauswahl, um einer Vereisung des Wärmetauschers vorzubeugen.

Die Freigabe des Zuluftreglers im Sub-Template [BAC_AC_ErcT_PID_01 \[► 420\]](#) erfolgt in Abhängigkeit der Anlagenbetriebsart. Bei Betriebsarten, wie beispielsweise der Sommernachtkühlung, ist der Zuluftregler gesperrt. Mittels des Hystereseomoduls **HysByDmp** wird die Bypassklappe bei gesperrter Energierückgewinnung aufgeföhren. So werden unnötig hohe Strömungswiderstände vermieden und der Energieverbrauch der Ventilatoren reduziert.

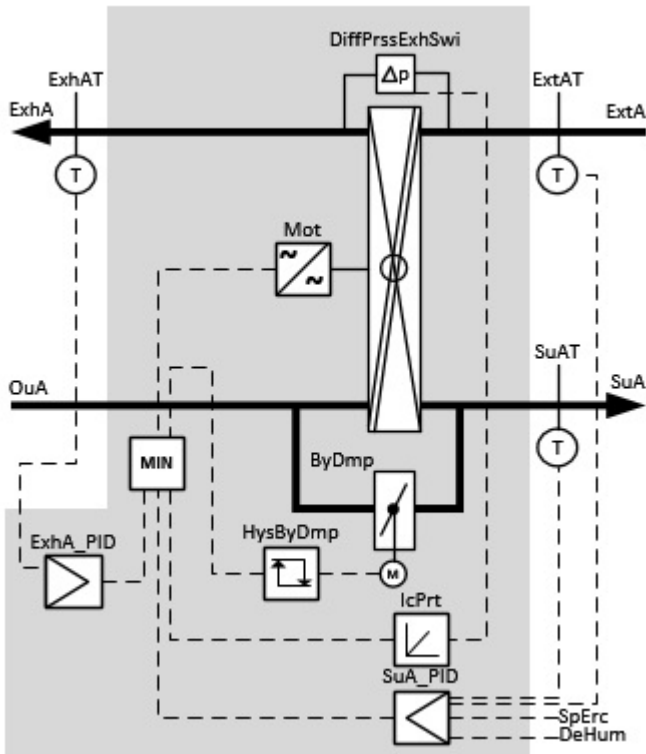
Schnittstelle



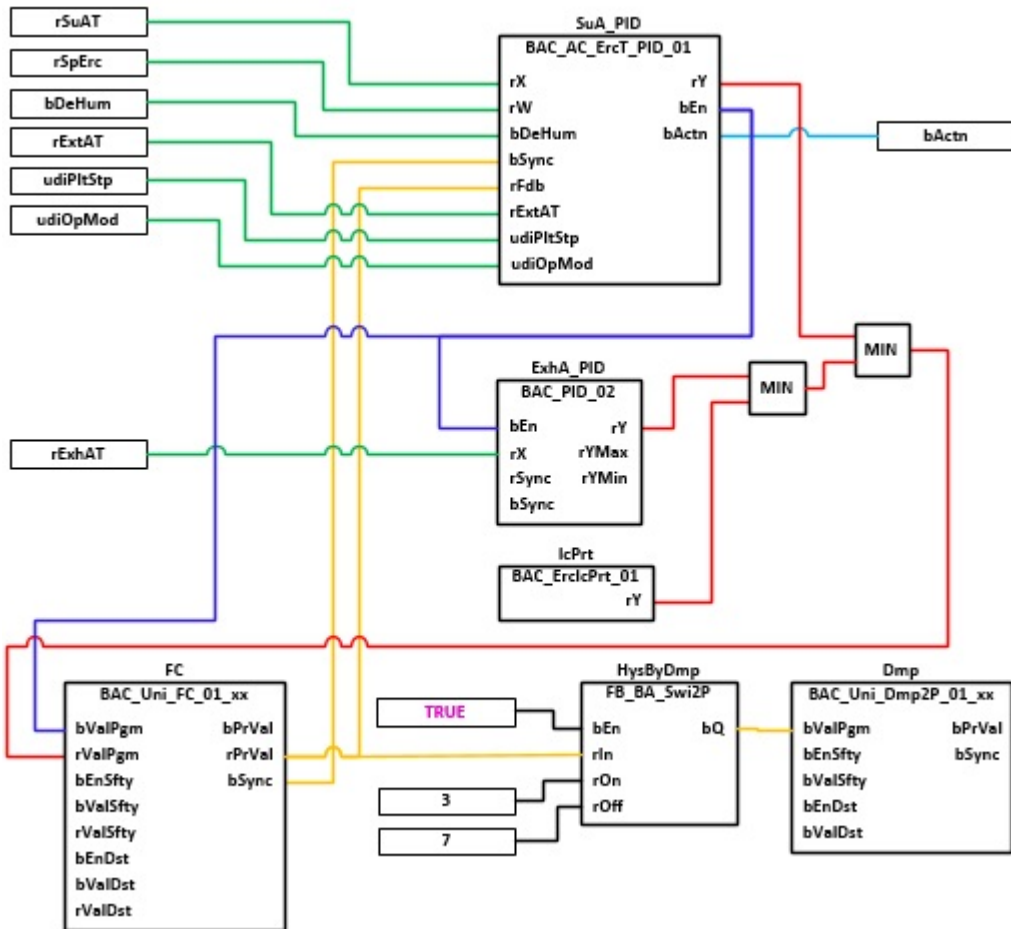
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

rSuAT      : REAL;
rSpErc     : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
bDeHum     : BOOL;
rExhAT     : REAL;
rExtAT     : REAL;

```

rSuAT: Messwert Zulufttemperatur

rSpErc: Sollwert Zulufttemperatur

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC AC StartT 01](#) [► 538].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC AC OpMod 01](#) [► 524].

bDeHum: Betriebsmeldung Entfeuchtebetrieb. Bei aktiver Entfeuchtung wird die Energierückgewinnung im Heizbetrieb gesperrt. Gewinnt die Wärmerückgewinnung Kälte aus der Abluft zurück, dann wird sie im Entfeuchtebetrieb auf 100% geschaltet.

rExhAT: Messwert Fortlufttemperatur

rExtAT: Messwert Ablufttemperatur

VAR_OUTPUT

```

bActn      : BOOL;

```

bActn: Wirksinn der Zulufttemperaturregelung **SuA_PID**. Der Wirksinn der Energierückgewinnung wird von verschiedenen Sollwertprogrammen oder Kaskadenreglern zur Bestimmung des Sollwertes für die Zulufttemperatur der Energierückgewinnung benötigt, z. B. [BAC AC SpSuAT 02](#) [► 681], [BAC AC CasCtrlT 02](#) [► 653].

Programmbeschreibung

Instanz	Typ	Aufgabe
SuA_PID	BAC AC ErcT PID 01 [► 420]	Sub-Template Zulufttemperaturregelung. Die Temperaturregelung erfolgt stetig mittels eines PID-Sequenzreglers
ExhA_PID	BAC PID 02 [► 599]	Sub-Template Fortlufttemperaturregelung für den Vereisungsschutz
IcPrt	BAC ErcIcPrt 01 [► 426]	Sub-Template für den Vereisungsschutz einer Energierückgewinnung mittels Differenzdruckschalter
Mot	BAC Uni FC 01 223 [► 563]	Sub-Template zur Ansteuerung eines Frequenzumformers. Dieser steuert dann den Motor des Rotationswärmetauschers.
HysByDmp	FB BA Swi2P [► 148]	Der Funktionsbaustein HysByDmp ist ein Zweipunktschalter und steuert die Bypassklappe in Abhängigkeit einer Hysterese auf und zu. Die Stellgröße für den Motor Mot des Rotationswärmetauschers dient als Eingangsgröße für den Zweipunktschalter.
ByDmp	BAC Uni Dmp2P 01 08 [► 572]	Sub-Template zur Ansteuerung der Bypassklappe.
	MIN, MIN	MIN-Auswahl der Stellsignale Zulufttemperaturregelung SuA_PID , Fortlufttemperaturregelung ExhA_PID und der Rampenfunktion Frostschutz

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.20 BAC_AC_ErcRot_02

Anwendung

Das Template **BAC_AC_ErcRot_02** dient zur Steuerung, Regelung und dem Vereisungsschutz einer Energierückgewinnung mit Rotationswärmetauscher.

Das Template **BAC_AC_ErcRot_02** ist ein Aufruftemplate.

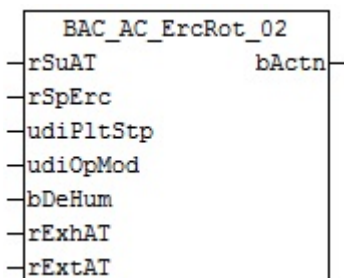
Innerhalb des Aufruftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

- **ErcSuA_PID**: Regelung der Zulufttemperatur
- **ErcExhA_PID**: Minimalbegrenzung der Fortlufttemperatur
- **ErcMot**: Ansteuerung WRG-Steuergerät
- **ErcIcPrt**: Vereisungsschutz des Wärmetauschers mittels Differenzdruckschalter

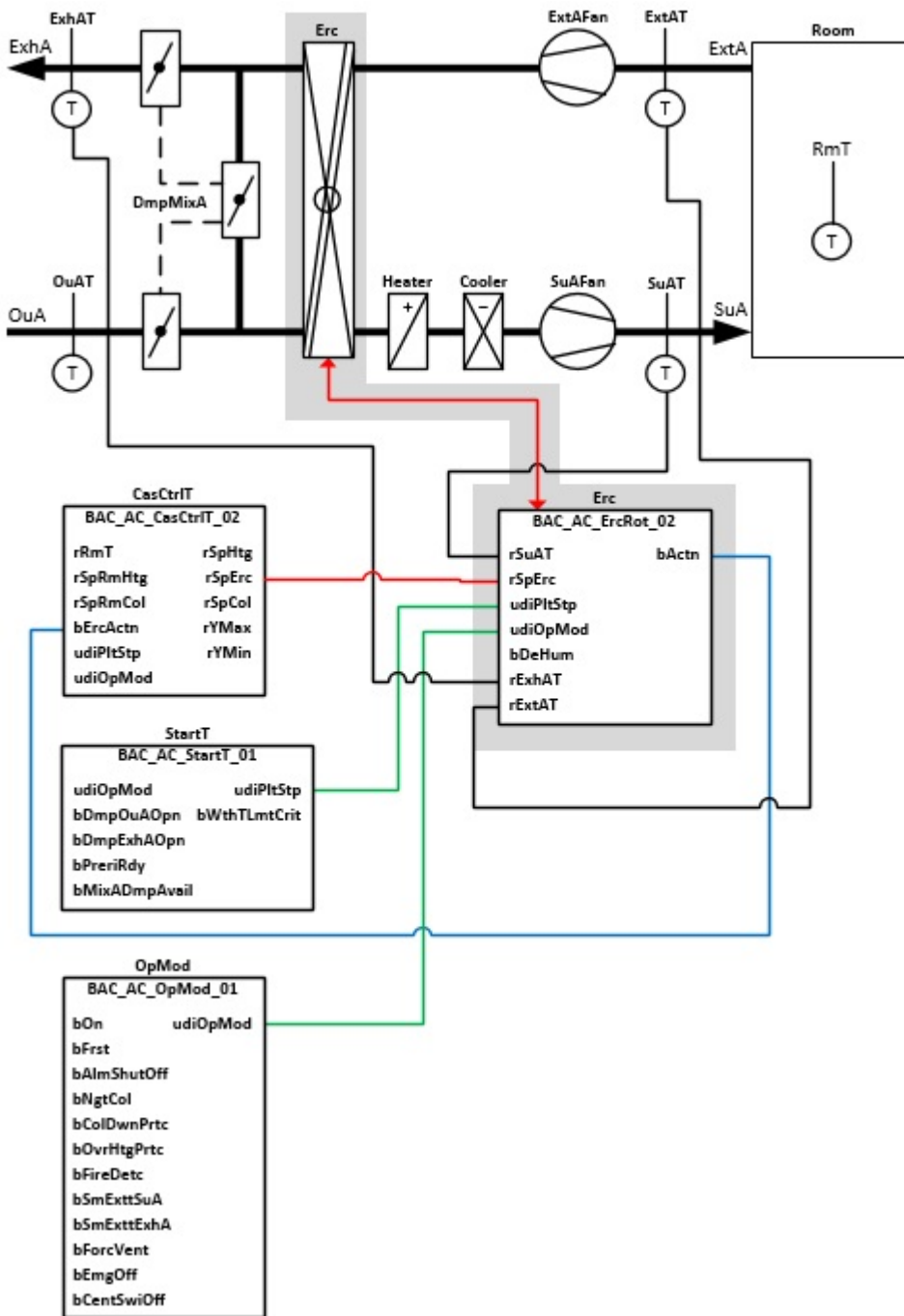
Der Fortluftminimalbegrenzer **ErcExhA_PID** und das Frostschutzprogramm **ErcIcPrt** begrenzen die Stellgröße des Zulufttemperaturreglers **ErcSuA_PID** der Energierückgewinnung über die Minimalauswahl, um einer Vereisung des Wärmetauschers vorzubeugen.

Die Freigabe des Zuluftreglers im Sub-Template [BAC AC ErcT PID_01](#) [► 420] erfolgt in Abhängigkeit der Anlagenbetriebsart. Bei Betriebsarten, wie beispielsweise der Sommernachtkühlung, ist der Zuluftregler gesperrt.

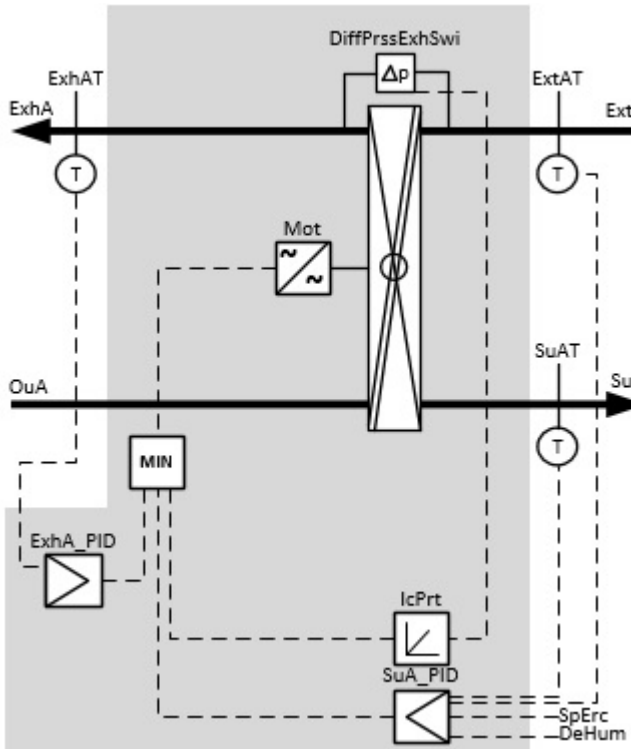
Schnittstelle



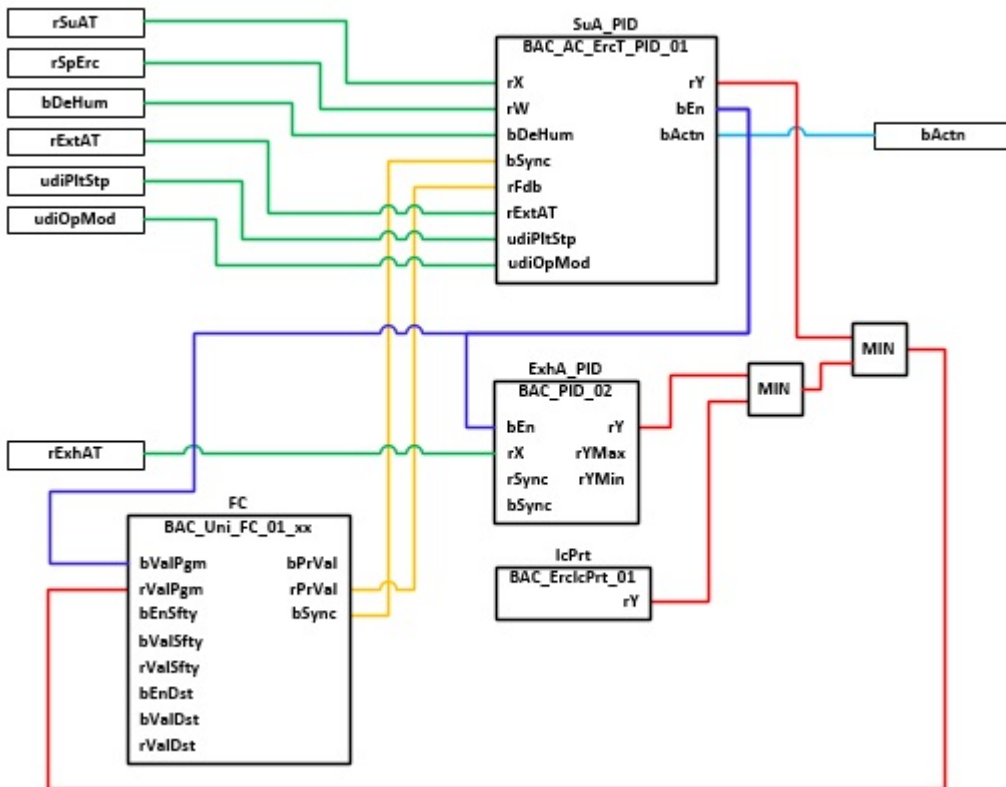
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

rSuAT      : REAL;
rSpErc     : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
bDeHum     : BOOL;
rExhAT     : REAL;
rExtAT     : REAL;
    
```

rSuAT: Messwert Zulufttemperatur

rSpErc: Sollwert Zlufttemperatur

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC AC StartT 01 \[► 538\]](#).

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC AC OpMod 01 \[► 524\]](#).

bDeHum: Betriebsmeldung Entfeuchtebetrieb. Bei aktiver Entfeuchtung wird die Energierückgewinnung im Heizbetrieb gesperrt. Gewinnt die Wärmerückgewinnung Kälte aus der Abluft zurück, dann wird sie im Entfeuchtebetrieb auf 100% geschaltet.

rExhAT: Messwert Fortlufttemperatur

rExtAT: Messwert Ablufttemperatur

VAR_OUTPUT

bActn : BOOL;

bActn: Wirksinn der Zulufttemperaturregelung **SuA_PID**. Der Wirksinn der Energierückgewinnung wird von verschiedenen Sollwertprogrammen oder Kaskadenreglern zur Bestimmung des Sollwertes für die Zulufttemperatur der Energierückgewinnung benötigt, z. B. [BAC AC SpSuAT 02 \[► 681\]](#), [BAC AC CasCtrlT 02 \[► 653\]](#).

Programmbeschreibung

Instanz	Typ	Aufgabe
SuA_PID	BAC AC ErcT PID 01 [► 420]	Sub-Template Zulufttemperaturregelung. Die Temperaturregelung erfolgt stetig mittels eines PID-Sequenzreglers
ExhA_PID	BAC PID 02 [► 599]	Sub-Template Fortlufttemperaturregelung für den Vereinsungsschutz
IcPrt	BAC ErcIcPrt 01 [► 426]	Sub-Template für den Vereinsungsschutz einer Energierückgewinnung mittels Differendruckschalter
Mot	BAC Uni FC 01 215 [► 563]	Sub-Template zur Ansteuerung eines Frequenzumformers. Dieser steuert dann den Motor des Rotationswärmetauschers.
	MIN, MIN	MIN-Auswahl der Stellsignale Zulufttemperaturregelung SuA_PID , Fortlufttemperaturregelung ExhA_PID und der Rampenfunktion Frostschutz

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.21 BAC_AC_ErcT_PID_01

Funktionsbeschreibung

Das Sub-Template **BAC_AC_ErcT_PID_01** ist ein Zulufttemperatur-Sequenzregler für eine Energierückgewinnung.

Die Referenzierung des Sollwertes, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte **X**, **W** und **Y**.

Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart **udiOpMod** und der globalen Temperatur-Kommunikationsstruktur **g_stSeqLinkT[PLT_NUM]**. Diese Daten- und Befehlsstruktur ist das Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein [FB BA_SeqLink](#)

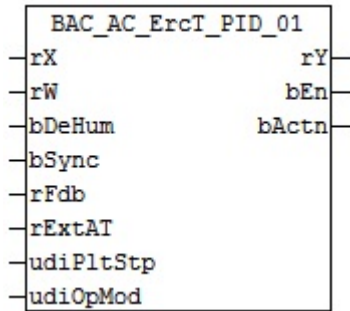
[►_171] einer Anlage.

Zur Anzeige der Reglerfreigabe dient das BACnet-BV-Objekt **En**.

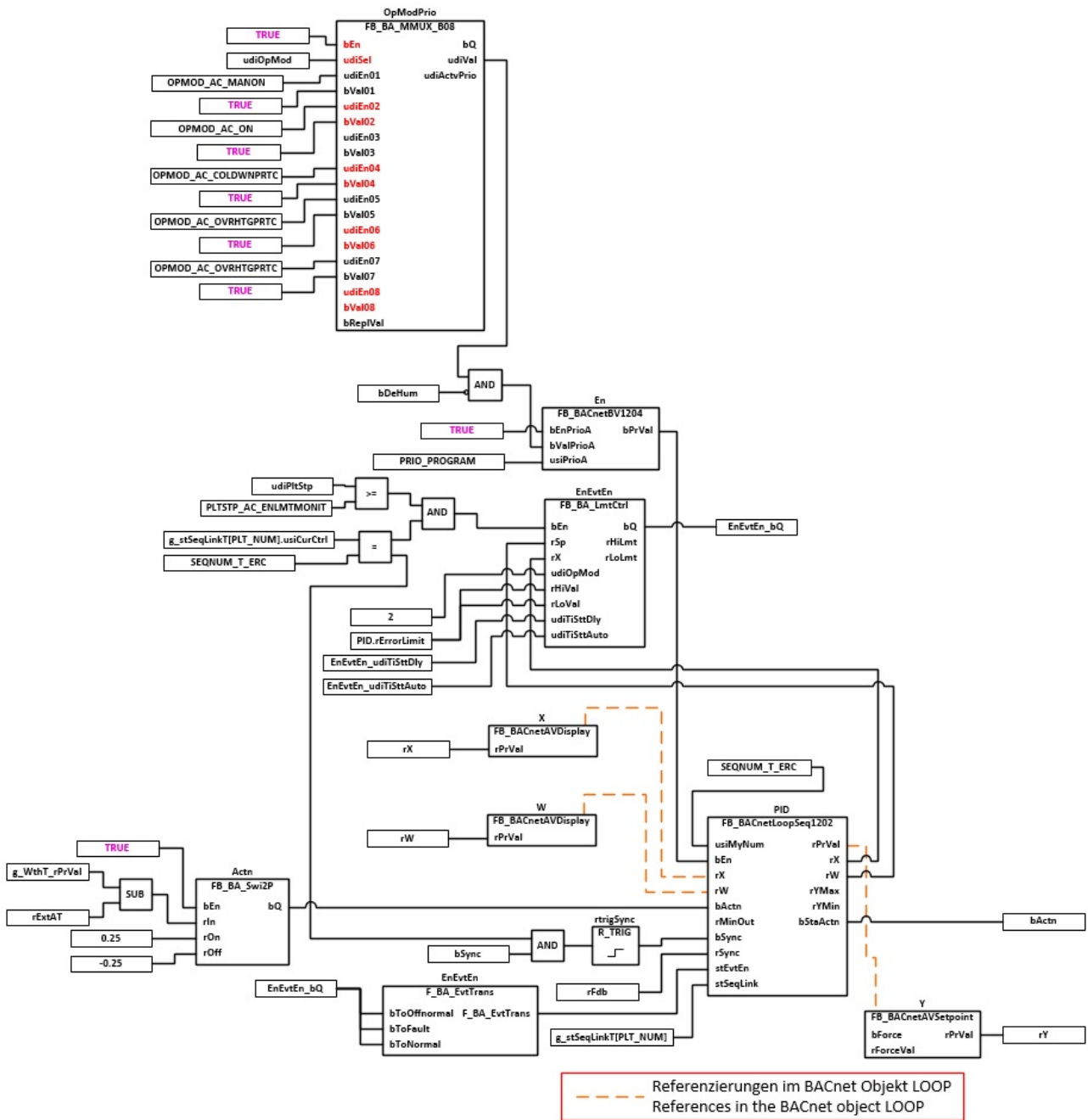
Die Grenzwertüberwachung des PID-Reglers wird in Abhängigkeit des Anlagenanfahrprozesses von dem Funktionsbaustein **EnEvtEn** gesteuert.

Im Entfeuchtebetrieb **bDeHum** = TRUE wird der PID-Sequenzregler gesperrt.

Schnittstelle

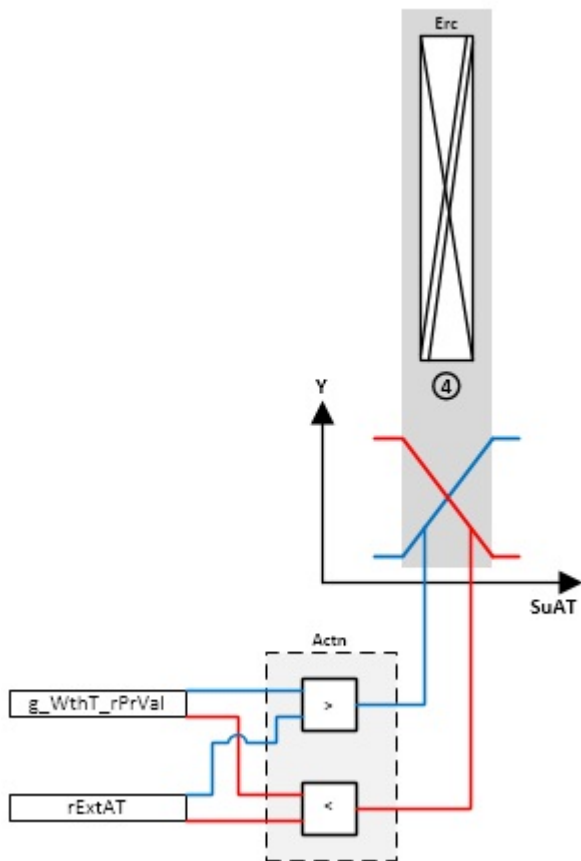


Blockschaltbild



Wirksinn

Die Auswahl des Wirksinns des PID-Sequenzreglers erfolgt anhand eines Vergleichs von der Außentemperatur mit der Ablufttemperatur.
 Ist die Außentemperatur kleiner als die Ablufttemperatur dann ist der Wirksinn des PID-Reglers indirekt (Heizbetrieb)
 Ist die Außentemperatur größer als die Ablufttemperatur dann ist der Wirksinn des PID-Reglers direkt (Kühlbetrieb)



VAR_INPUT

```
rX      : REAL;
rW      : REAL;
bDeHum  : BOOL;
bSync   : BOOL;
rFdb    : REAL;
rExtAT  : REAL;
udiPltStp : UDINT;
udiOpMod : UDINT;
```

rX: Messwert Zulufttemperatur

rW: Sollwert der Zulufttemperatur

bDeHum: Entfeuchtung aktiv, Sperrung der Freigabe des Zulufttemperaturreglers. Bei aktiver Entfeuchtung wird die Energierückgewinnung im Heizbetrieb gesperrt. Gewinnt die Wärmerückgewinnung Kälte aus der Abluft zurück, dann wird sie im Entfeuchtebetrieb auf 100% geschaltet.

bSync: Eingang für die Synchronisation des Reglers

rFdb: Stellungsrückmeldung Aktor

rExtAT: Messwert Ablufttemperatur

udiPltStp: Schritt beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartT_01 \[► 538\]](#)

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod_01 \[► 524\]](#)

VAR_OUTPUT

```
rY      : REAL;
bEn     : BOOL;
bActn   : BOOL;
```

rY: Ausgabe der Stellgröße

bEn: Freigabe der Regelung Energierückgewinnung

bActn: Wirksinn des Zuluftreglers, wird benötigt für die Sollwertstrategie; TRUE = Direct = Kühlen; FALSE = Reverse = Heizen

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [▶ 182]

Die Auswertung der Alarmergebnisse einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [▶ 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur [g_stSeqLinkT\[PLT_NUM\]](#) dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein [FB_BA_SeqLink](#) zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe			
X	FB_BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts			
W	FB_BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts			
OpModPrio	FB_BA_MMUX_B08 [▶ 208]	Der Multiplexer definiert die Freigabebedingungen des Sequenzreglers in Abhängigkeit der Anlagenbetriebsart.			
		udiOpMod	Freigabe	Bemerkung	
		OPMOD AC MANON [▶ 364]	Hand ein	TRUE	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet
		OPMOD AC ON [▶ 364]	Ein	TRUE	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm
		OPMOD AC OVRHTGPRTC [▶ 364]	Überhitzungsschutz	TRUE	Die Anlage läuft in der Betriebsart Überhitzungsschutz
		OPMOD AC COLDWNPRTC [▶ 364]	Stützbetrieb, Auskühlschutz	TRUE	Die Anlage läuft in der Betriebsart Auskühlschutz
		OPMOD AC FORCVENT [▶ 364]	Zwangsbelüftung	TRUE	Die Anlage läuft in der Betriebsart Zwangsbelüftung

Instanz	Typ	Aufgabe
En	FB_BACnetBV1204 [▶ 94]	Mit dem BV-Objekt wird die Freigabe des Sequenzreglers in der MBE angezeigt.
	AND	Im Entfeuchtebetrieb (bDeHum = TRUE) und gleichzeitiger Freigabe durch OpModPrio wird der Energierückgewinnung die Freigabe genommen.
Actn	FB_BA_Swi2P [▶ 148]	<p>Der Funktionsbaustein FB_BA_Swi2P [▶ 148] bestimmt den Wirksinn des Sequenzreglers.</p> <p>Ist die Subtraktion von der Außentemperatur und der Ablufttemperatur $< - 0,25$, dann ist der Ausgang des Funktionsbausteins Actn FALSE. Der Wirksinn des Loop-Objekts ist damit indirekt und die Energierückgewinnung befindet sich im Heizbetrieb.</p> <p>Ist die Subtraktion von der Außentemperatur und der Ablufttemperatur $> + 0,25$, dann ist der Ausgang des Funktionsbausteins Actn TRUE. Der Wirksinn des Loop-Objekts ist damit direkt und die Energierückgewinnung befindet sich im Kühlbetrieb.</p> <p>Die Hysterese zwischen $- 0.25$ und $+ 0,25$ des Funktionsbausteins Actn verhindert ein häufiges Umschalten des Reglerwirksinns bei annähernd gleicher Außen- und Ablufttemperatur.</p>
EnEvtEn	FB_BA_LmtCtrl [▶ 233]	<p>Das BACnet-Loop-Objekt PID überwacht die Funktion der Regelung in dem es den Sollwert W und den Istwert X miteinander vergleicht. Ist die Abweichung $W-X$ größer als das Property ErrorLimit, dann sendet das Loop-Objekt eine Meldung an die MBE.</p> <p>Beim Anlagenstillstand, im Moment des Starts und bis zum eingeregelt Zustand der Anlage wird das Melden des Loop-Objekts unterdrückt, damit keine falschen Meldungen an die MBE gesendet werden. Erst wenn die RLT-Anlage vollständig hochgefahren ist und die Regelung sich eingependelt hat, wird das Melden des Loop-Objekts freigegeben. Zusätzlich wird das Melden aktiviert, wenn die Regelung nach langer Zeit nicht in dem vom Property ErrorLimit definierten Bereich um den Sollwert gelangt ist.</p> <p>Die Freigabe des objektinternen Meldens erfolgt indem auf die BACnetEventTransitionBits des Loop-Objekts geschrieben wird.</p> <p>Für die Freigabe des Meldens vom Loop-Objekt müssen die folgenden Bedingungen erfüllt sein:</p> <ol style="list-style-type: none"> <p>Das Anlagenstartprogramm BAC_AC_StartT_01 [▶ 538] hat die Überwachung der Regelung und die Grenzwertüberwachung der Fühler freigegeben udiPltStp \geq PLTSTP_AC_ENLMTMONIT und Die Energierückgewinnung ist der aktive Regler in der Regelsequenz. g_stSeqLinkT[PLT_NUM] [▶ 364].usiCurCtrl = SEQNUM_T_ERC und Die Zulufttemperatur hat sich dem Sollwert so weit angenähert, dass sie sich in einem Bereich zwischen rSp - ErrorLimit und rSp + ErrorLimit eingependelt hat. und Die Zulufttemperatur muss mindestens für die Dauer von EnEvtEn_udiTiSttDly innerhalb des Bereiches von rSp - ErrorLimit und rSp + ErrorLimit verblieben sein.</p> <p>Der Timer EnEvtEn_udiTiSttAuto ist abgelaufen und die Regelung hat Ihren Sollwertbereich nicht erreicht.</p>

Instanz	Typ	Aufgabe
	F_BA_EvtTrans	schreibt auf die BACnetEventTransitionBits to_offnormal, to_fault und to_normal des Loop-Objekts. Dazu muss der Eingang bEnEvtEn = TRUE sein
PID	FB_BACnetLoopSeq12 02 [▶ 112]	Sequenzregler Zulufttemperatur Energierückgewinnung.
rtrigSync	R_TRIG	Bei einer steigenden Flanke am Eingang bSync wird das Loop-Objekt auf den Wert von lrSync aufsynchronisiert. Wenn der Aktor der Energierückgewinnung durch Schreiben einer höheren Priorität auf das zugehörige AO-Objekt, von der MBE oder durch Betätigen der örtlichen Vorrangbedienung übersteuert wurde, weicht die aktuelle Position des Aktors von dem Ausgang des Loop-Objekts ab. Mit den Variablen bSync und rFdbVlv kann die Synchronität zwischen der Position des Aktors und dem Regler wieder hergestellt werden.
Y	FB_BACnetAVSetpoint [▶ 70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts
	AND, SEL, MAX	Netzwerk welches den Ausgabewert rY durch die MAX -Auswahl ausgibt. An der MAX -Auswahl wird zum einen der Wert des Zulufttemperatur-Reglers PID ausgegeben. Zum anderen wird in Abhängigkeit des Entfeuchtebetriebs bDeHum und des Vergleichs der Außentemperatur mit der Ablufttemperatur ein Wert von 100 oder 0 ausgegeben.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

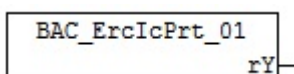
9.22 BAC_ErcIcPrt_01

Funktionsbeschreibung

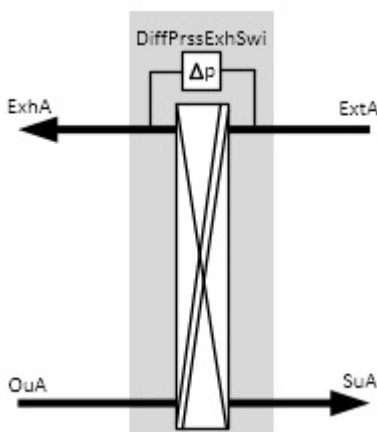
Das Template dient zum Schutz vor der Vereisung einer Energierückgewinnung.

Beim Ansprechen des Differenzdruckschalters **DiffPrssExhSwi** wird eine Rampenfunktion aktiviert welche ein analoges Begrenzungssignal ausgibt.

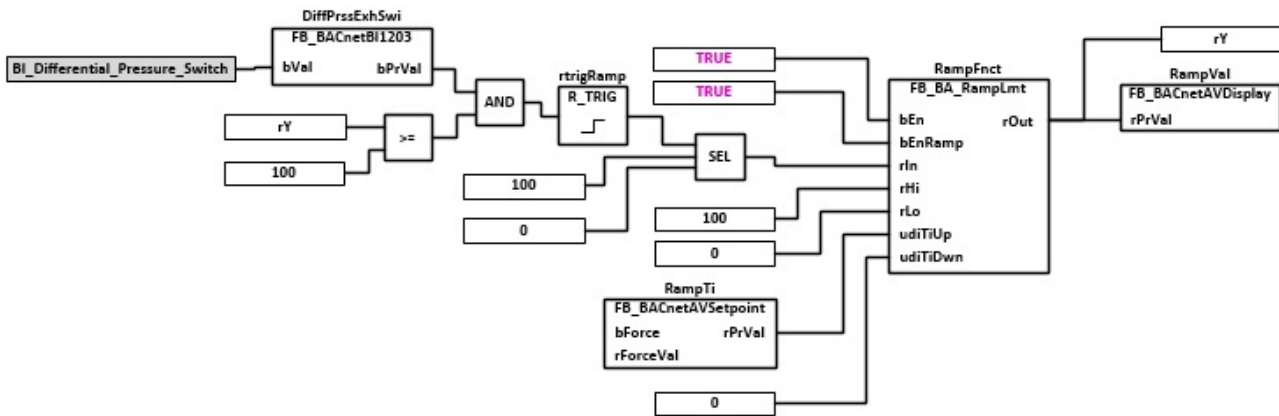
Die Begrenzung der Energierückgewinnung bewirkt eine Aubtauung durch den warmen Abluftstrom.



Anlagenschema



Blockschaltbild



VAR_OUTPUT

rY : REAL;

rY: Ausgabe des Rampensignals. Bei nicht ausgelöstem Differenzdruckschalter wird an rY 100% ausgegeben

VAR CONSTANT

PLT_NUM : BYTE := 1;

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt. Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm](#). [► 182] Die Auswertung der Alarme einer Anlage, z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB BA AlarmPlt](#). [► 186] Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
DiffPrssExh Swi	FB_BACnetBI1203 [► 72]	BI-Objekt Differenzdruckschalter über dem Fortluft seitigen Wärmetauscher
RampTi	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe der Anstiegszeit des Rampensignals
RampFunct	FB_BA_RampLmt [► 143]	Rampenfunktion welche ein von 0% ausgehendes steigendes Rampensignal erzeugt, wenn der Differenzdruckwächter angesprochen hat.
RampVal	FB_BACnetAVDisplay [► 69]	Das AV-Objekt zeigt das Rampensignal an.
rtrigRamp	R_TRIG SEL	Gibt der Rampenfunktion für einen Zyklus den Startwert 0% vor, wenn der Differenzdruckschalter ausgelöst hat.

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Differential_Pressure_Switch	BOOL		Eingang	Digitaleingang - Meldung - Differenzdruckschalter

Versionshistorie

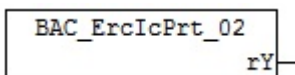
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.23 BAC_ErcIcPrt_02

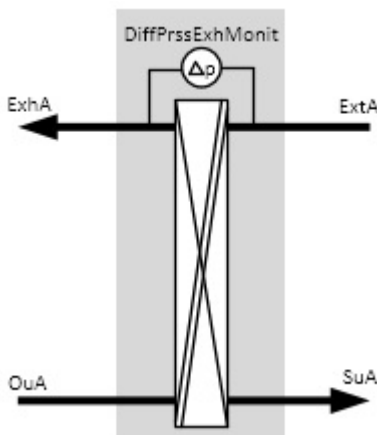
Funktionsbeschreibung

Das Template dient zum Schutz vor der Vereisung einer Energierückgewinnung. Beim Ansprechen des HighLimit des Differenzdruckfühlers **DiffPrssExhMonit** wird eine Rampenfunktion aktiviert, welche ein analoges Begrenzungssignal erzeugt. Die Begrenzung der Energierückgewinnung bewirkt eine Abtaugung durch den warmen Abluftstrom.

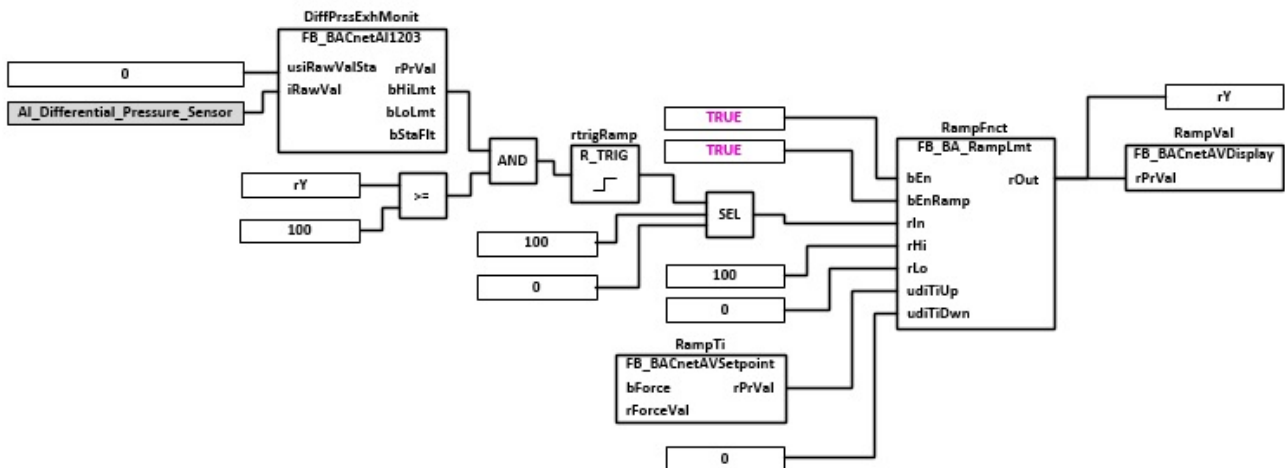
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_OUTPUT

rY : REAL;

rY: Ausgabe des Rampensignals. Bei nicht ausgelöstem HighLimit des Differenzdruckfühlers wird an rY 100% ausgegeben

VAR CONSTANT

PLT_NUM : BYTE := 1;

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [► 182]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
DiffPrssExhMonit	FB_BACnetAI1203 [► 49]	Differenzdruckfühler über dem Fortluft seitigen Wärmetauscher
RampTi	FB_BACnetAVSetpoint [► 70]	Eingabe der Anstiegszeit des Rampensignals
RampFnc	FB_BA_RampLmt [► 143]	Rampenfunktion welche ein von 0% ausgehendes steigendes Rampensignal erzeugt, wenn der Differenzdruckfühler angesprochen hat.
RampVal	FB_BACnetAVDisplay [► 69]	Anzeige des Rampensignals
rtrigRamp	R_TRIG SEL	Gibt der Rampenfunktion für einen Zyklus den Startwert 0% vor, wenn der Differenzdruckschalter ausgelöst hat.

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
AI_Differential_Pressure_Sensor	INT		Eingang	Analogeingang - Messwert - Differenzdruck Fortluft

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.24 BAC_AC_ExtAFan_FC_01

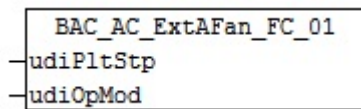
Anwendung

Das Template **BAC_AC_ExtAFan_FC_01** dient zur Steuerung und Regelung eines druckgeregelten Abluftventilators.

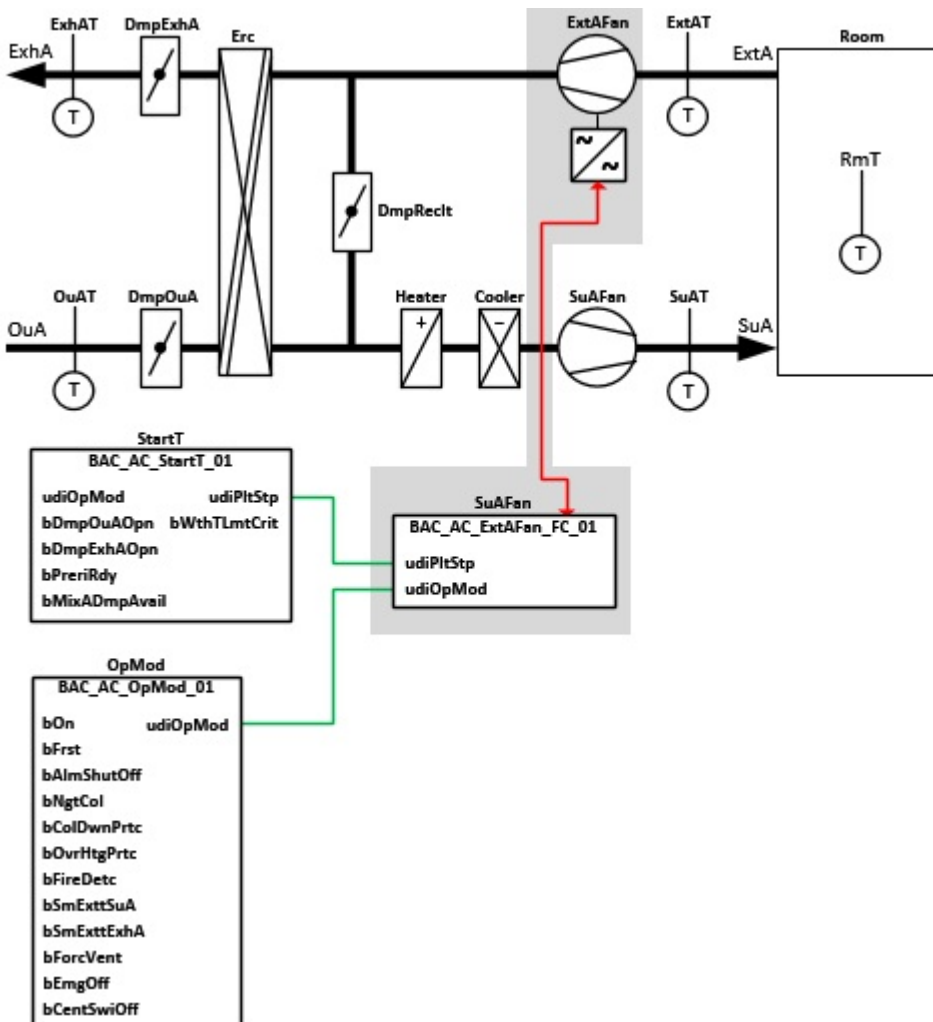
Die wesentlichen Aufgaben des Templates sind:

- Druckregelung des Abluftventilators
- Ansteuerung des Frequenzumformers
- Differenzdrucküberwachung des Ventilators

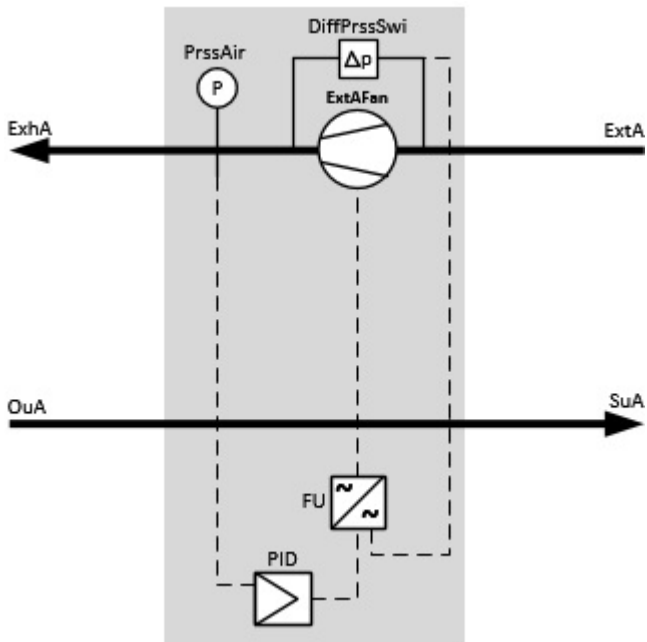
Schnittstelle



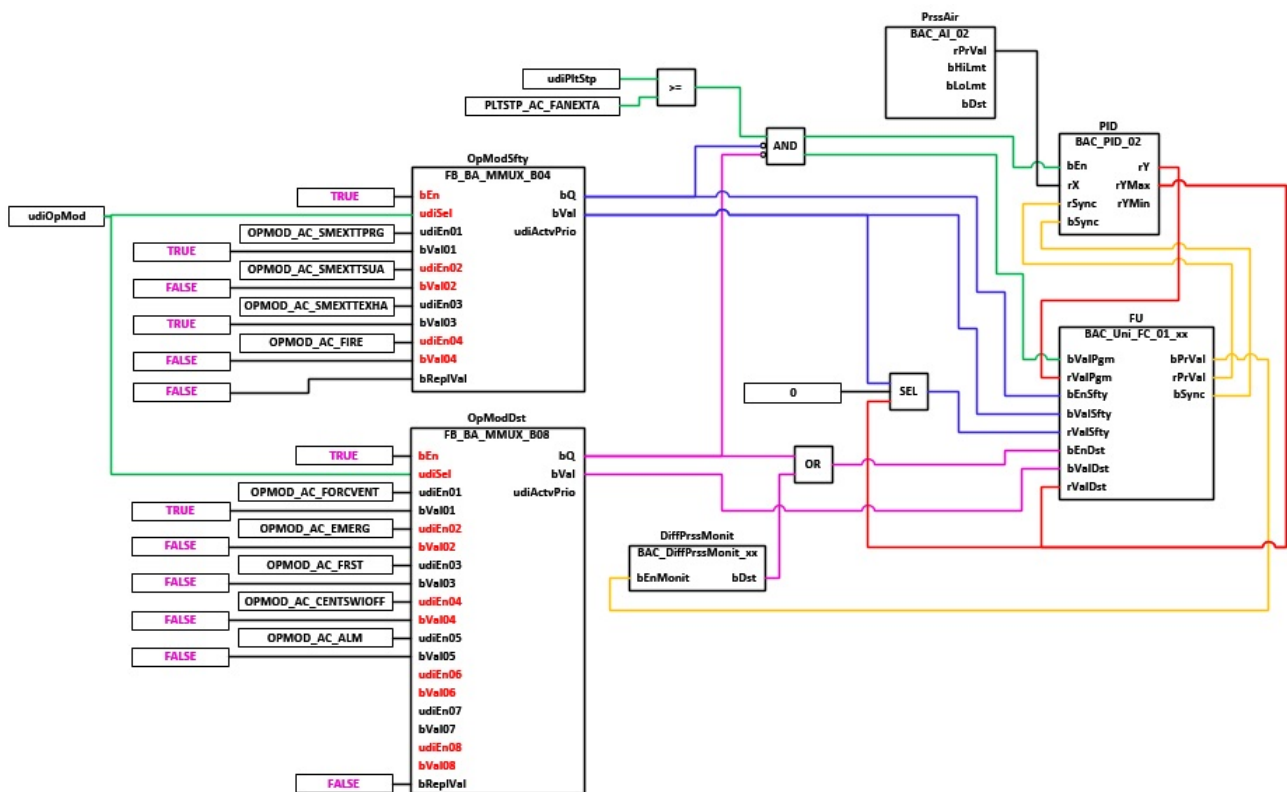
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

udiPltStp      : UDINT;
udiOpMod       : UDINT;
    
```

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm BAC AC StartT 01 [▶ 538].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl BAC AC OpMod 01 [▶ 524].

Programmbeschreibung

Instanz	Typ	Aufgabe		
PrssAir	BAC_AI_02 [▶_685]	Sub-Template Druckfühler PrssAir ist die Regelgröße des Druckreglers.		
PID	BAC_PID_02 [▶_599]	Sub-Template Druckregelung.		
DiffPrssMo nit	BAC_DiffPrssMonit_01 [▶_441]	Sub-Template Differenzdrucküberwachung mittels Differenzdruckschalter		
FU	BAC_Uni_FC_01_223 [▶_563]	Sub-Template Ansteuerung eines Frequenzumformers inklusive Motorlogik.		
OpModSfty	FB_BA_MMUX_B04 [▶_208]	Der Multiplexer definiert anhand der Anlagenbetriebsart udiOpMod die Sicherheitspriorität für die Ansteuerung des Abluftventilators FU		
		udiOpMod	Freigabe	
		OPMOD_AC_SMEXTTPRG	Entrauchung Programm	TRUE
		OPMOD_AC_SMEXTTSA	Entrauchung Zuluft	FALSE
		OPMOD_AC_SMEXTTEXHA	Entrauchung Fortluft	TRUE
		OPMOD_AC_FIRE	Feuer	FALSE
OpModDst	FB_BA_MMUX_B08 [▶_208]	Der Multiplexer definiert anhand der Anlagenbetriebsart udiOpMod die Priorität Störung für die Ansteuerung des Abluftventilators FU		
		udiOpMod	Freigabe	
		OPMOD_AC_FORCVENT	Zwangsbelüftung	TRUE
		OPMOD_AC_EMERG	Notfall	FALSE
		OPMOD_AC_FRST	Frost	FALSE
		OPMOD_AC_CENTS WIOFF	Zentralabschaltung	FALSE
		OPMOD_AC_ALM	Störung/Alarm	FALSE
	>=, AND	Netzwerk welches anhand der Anlagenschritte udiPitStp die Freigabe für den Abluftventilator als Ergebnis hat		
	SEL	SEL-Auswahl welches den Wert für die Sicherheitspriorität des Template FU vorgibt		
	OR	Das Ergebnis der OR-Operation ist die Freigabe der Störungspriorität des Templates FU .		

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.25 BAC_AC_ExtAFan1st_01

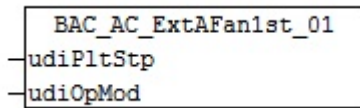
Anwendung

Das Aufruftemplate **BAC_AC_ExtAFan1st_01** dient zur Steuerung eines einstufigen Abluftventilators.

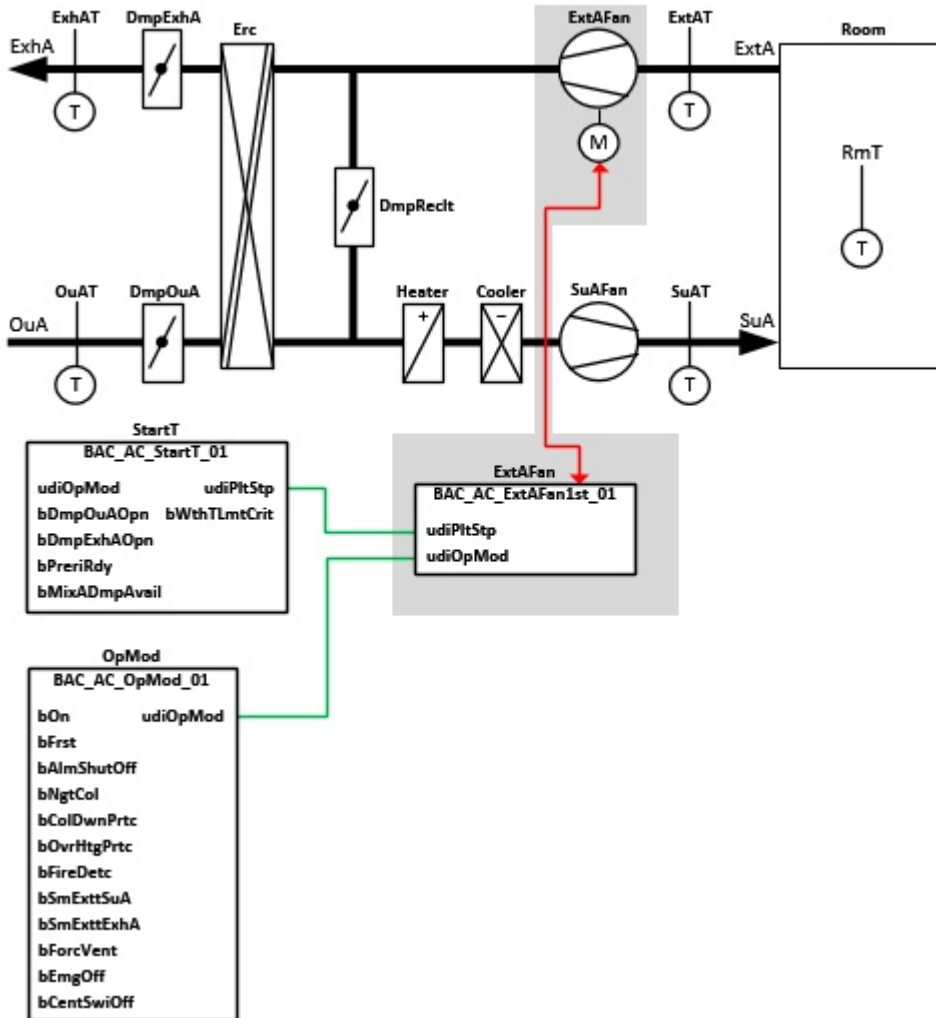
Die wesentlichen Aufgaben des Templates sind:

- Ansteuerung des Abluftventilators
- Differenzdrucküberwachung

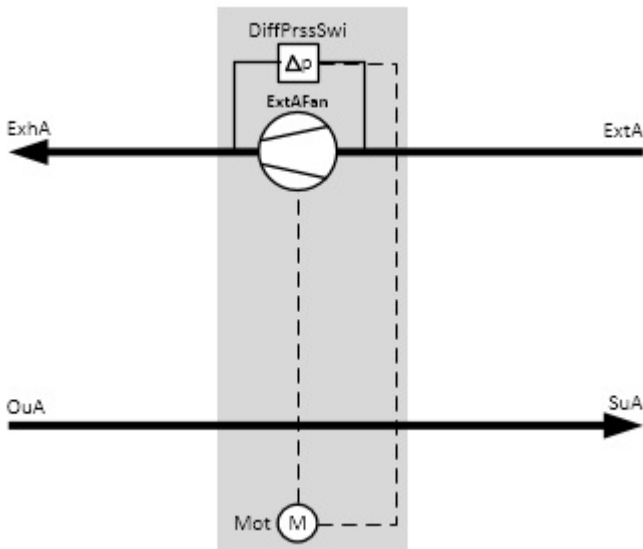
Schnittstelle



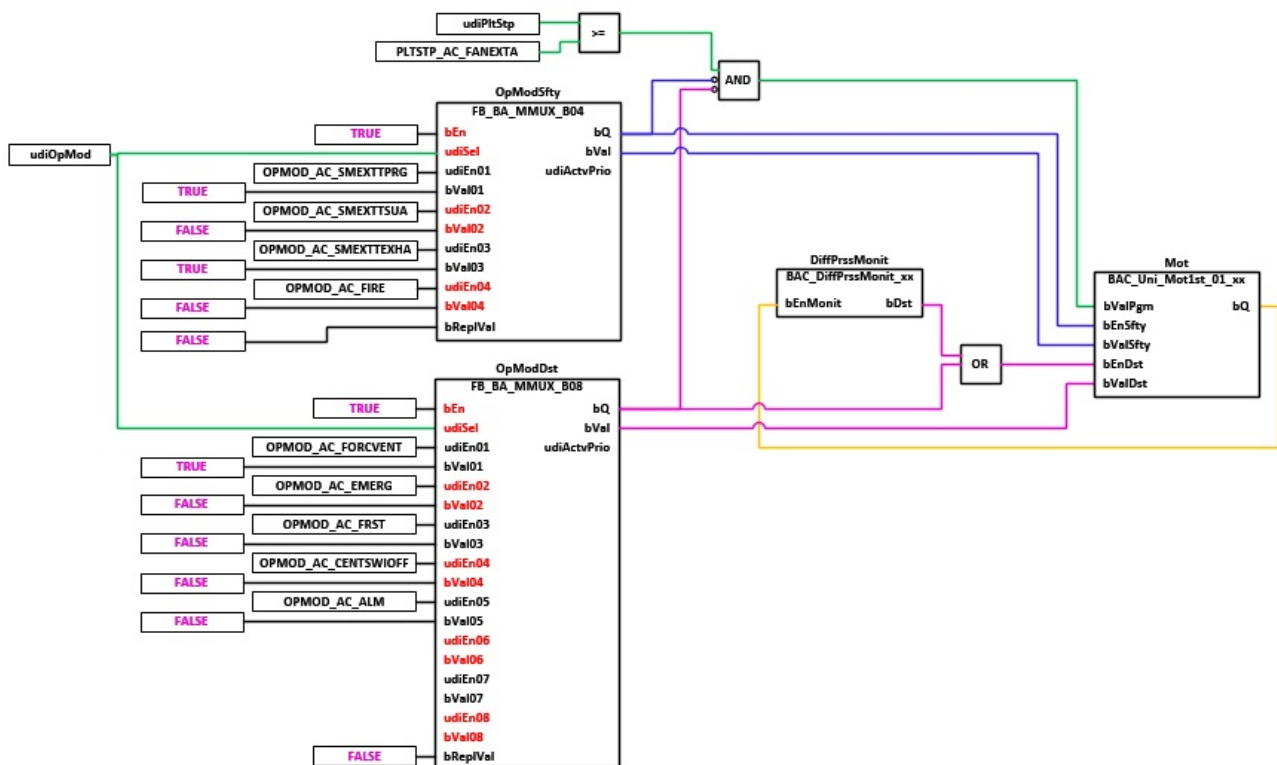
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

udiPltStp      : UDINT;
udiOpMod       : UDINT;

```

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC AC StartT 01](#) [[▶ 538](#)].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC AC OpMod 01](#) [[▶ 524](#)].

Programmbeschreibung

Instanz	Typ	Aufgabe		
DiffPrssMonit	BAC_DiffPrssMonit_01 [▶ 441]	Sub-Template Differenzdrucküberwachung mittels Differenzdruckschalter		
Mot	BAC_Uni_Mot1st_01_29 [▶ 576]	Sub-Template Ansteuerung eines einstufigen Motors inklusive Motorlogik.		
OpModSfty	FB_BA_MMUX_04 [▶ 208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Sicherheitspriorität für die Ansteuerung des Abluftventilators Mot		
		udiOpMod	Freigabe	
		OPMOD_AC_SMEXTTPRG	Entrauchung Programm	TRUE
		OPMOD_AC_SMEXTTSA	Entrauchung Zuluft	FALSE
		OPMOD_AC_SMEXTTEXHA	Entrauchung Fortluft	TRUE
		OPMOD_AC_FIRE	Feuer	FALSE
OpModDst	FB_BA_MMUX_08 [▶ 208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Priorität Störung für die Ansteuerung des Abluftventilators Mot		
		udiOpMod	Freigabe	
		OPMOD_AC_FORCVENT	Zwangsbelüftung	TRUE
		OPMOD_AC_EMERG	Notfall	FALSE
		OPMOD_AC_FRST	Frost	FALSE
		OPMOD_AC_CENTS WIOFF	Zentralabschaltung	FALSE
		OPMOD_AC_ALM	Störung/Alarm	FALSE
	AND >=	Das Ergebnis des Netzwerkes ist die Freigabe für den Abluftventilator anhand der Anlagenschritte udiPltStp		
	OR	Das Ergebnis der OR-Operation ist die Freigabe der Störungspriorität des Templates Mot .		

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.26 BAC_AC_SuAFan_FC_01

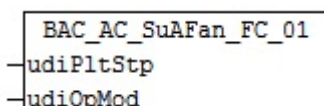
Anwendung

Das Aufruftemplate BAC_AC_SuAFan_FC_01 dient zur Steuerung eines druckgeregelten Zuluftventilators.

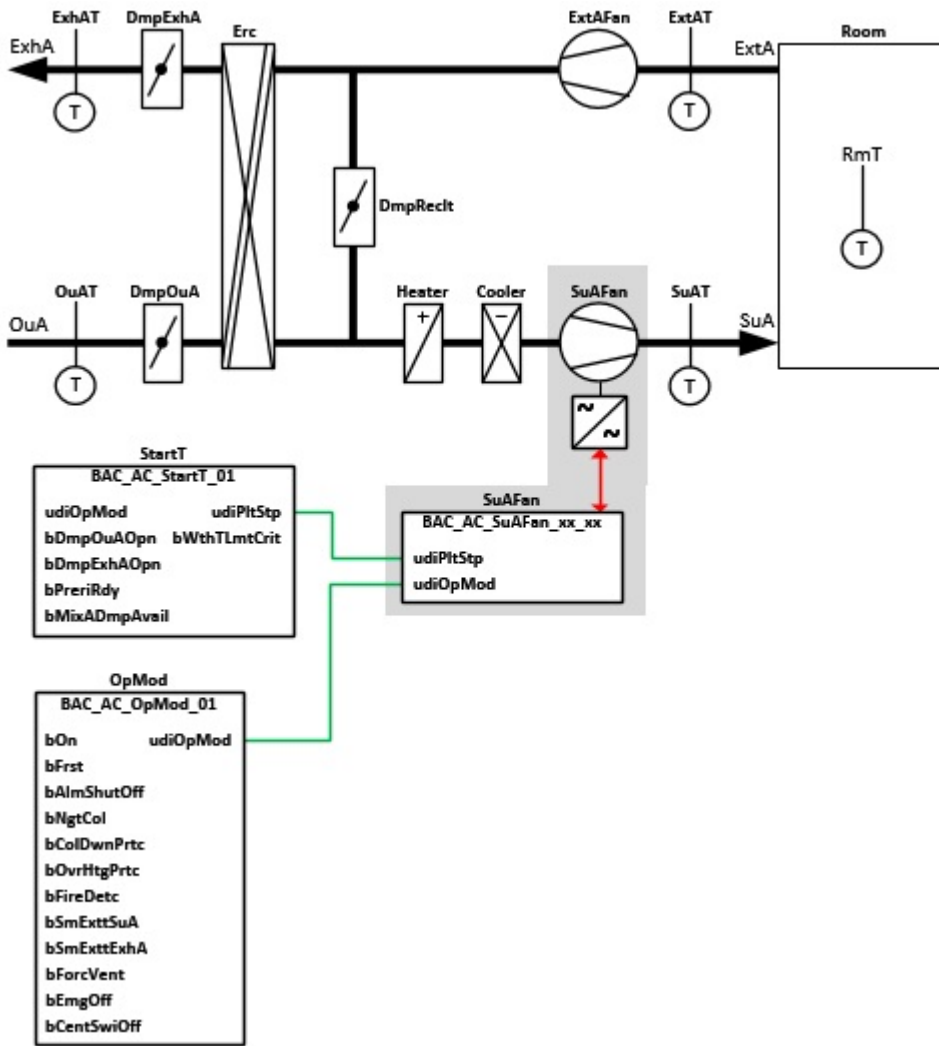
Die wesentlichen Aufgaben des Templates sind:

- Druckregelung des Zuluftventilators
- Ansteuerung des Frequenzumformers
- Differenzdrucküberwachung

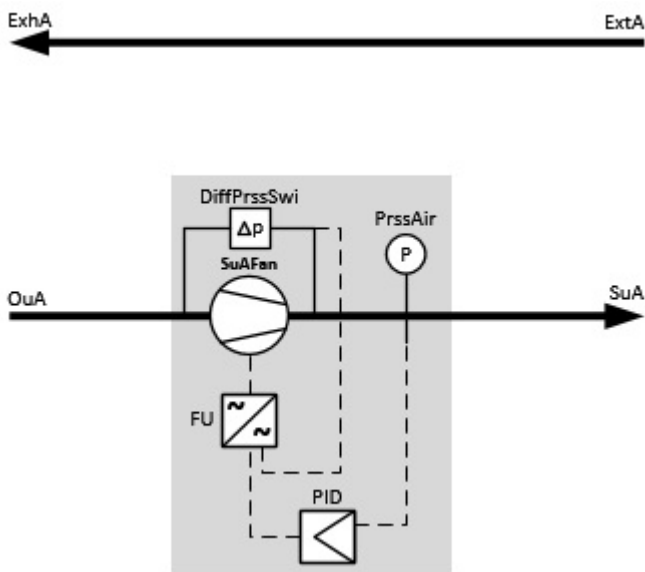
Schnittstelle



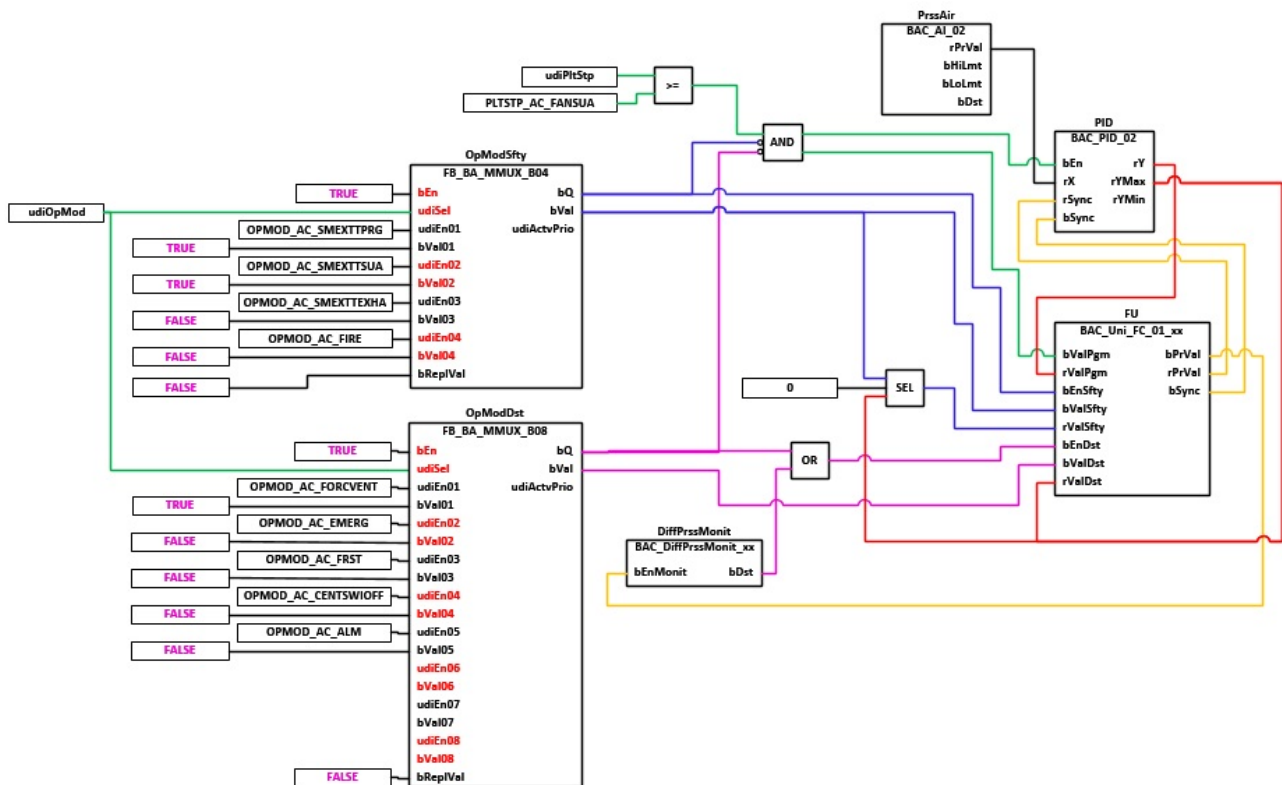
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```
udiPltStp      : UDINT;
udiOpMod       : UDINT;
```

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm BAC AC StartT 01 [▶ 538].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl BAC AC OpMod 01 [▶ 524].

Programmbeschreibung

Instanz	Typ	Aufgabe		
PrssAir	BAC AI 02 [▶ 685]	Sub-Template Druckfühler PrssAir ist die Regelgröße des Druckreglers PID .		
PID	BAC PID 02 [▶ 599]	Sub-Template Druckregelung. Die Regelung erfolgt stetig mittels eines PID-Reglers		
DiffPrssMonit	BAC DiffPrssMonit 01 [▶ 441]	Sub-Template Differenzdrucküberwachung mittels Differenzdruckschalter		
FU	BAC Uni FC 01 223 [▶ 563]	Template Ansteuerung eines Frequenzumformers inklusive Motorlogik.		
OpModSfty	FB BA MMUX B04 [▶ 208]	Der Multiplexer definiert anhand der Anlagenbetriebsart udiOpMod die Sicherheitspriorität für die Ansteuerung des Zuluftventilators FU		
		udiOpMod	Freigabe	
		OPMOD_AC_SMEXTTPRG	Entrauchung Programm	TRUE
		OPMOD_AC_SMEXTTSUA	Entrauchung Zuluft	TRUE
		OPMOD_AC_SMEXTTEXHA	Entrauchung Fortluft	FALSE
OPMOD_AC_FIRE	Feuer	FALSE		

Instanz	Typ	Aufgabe		
OpModDst	FB BA MMUX B08 [▶ 208]	Der Multiplexer definiert anhand der Anlagenbetriebsart udiOpMod die Priorität Störung für die Ansteuerung des Zuluftventilators FU		
		udiOpMod	Freigabe	
		OPMOD_AC_FORCVENT	Zwangsbelüftung	TRUE
		OPMOD_AC_EMERG	Notfall	FALSE
		OPMOD_AC_FRST	Frost	FALSE
		OPMOD_AC_CENTS WIOFF	Zentralabschaltung	FALSE
		OPMOD_AC_ALM	Störung/Alarm	FALSE
	>=, AND	Netzwerk welches anhand der Anlagenschritte udiPltStp die Freigabe für den Zuluftventilator als Ergebnis hat		
	SEL	SEL-Auswahl welches den Wert für die Sicherheitspriorität des Template FU vorgibt		
	OR	Das Ergebnis der OR-Operation ist die Freigabe der Störungspriorität des Templates FU .		

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.27 BAC_AC_SuAFan1st_01

Anwendung

Das Aufruftemplate **BAC_AC_SuAFan1st_01** dient zur Steuerung eines einstufigen Zuluftventilators.

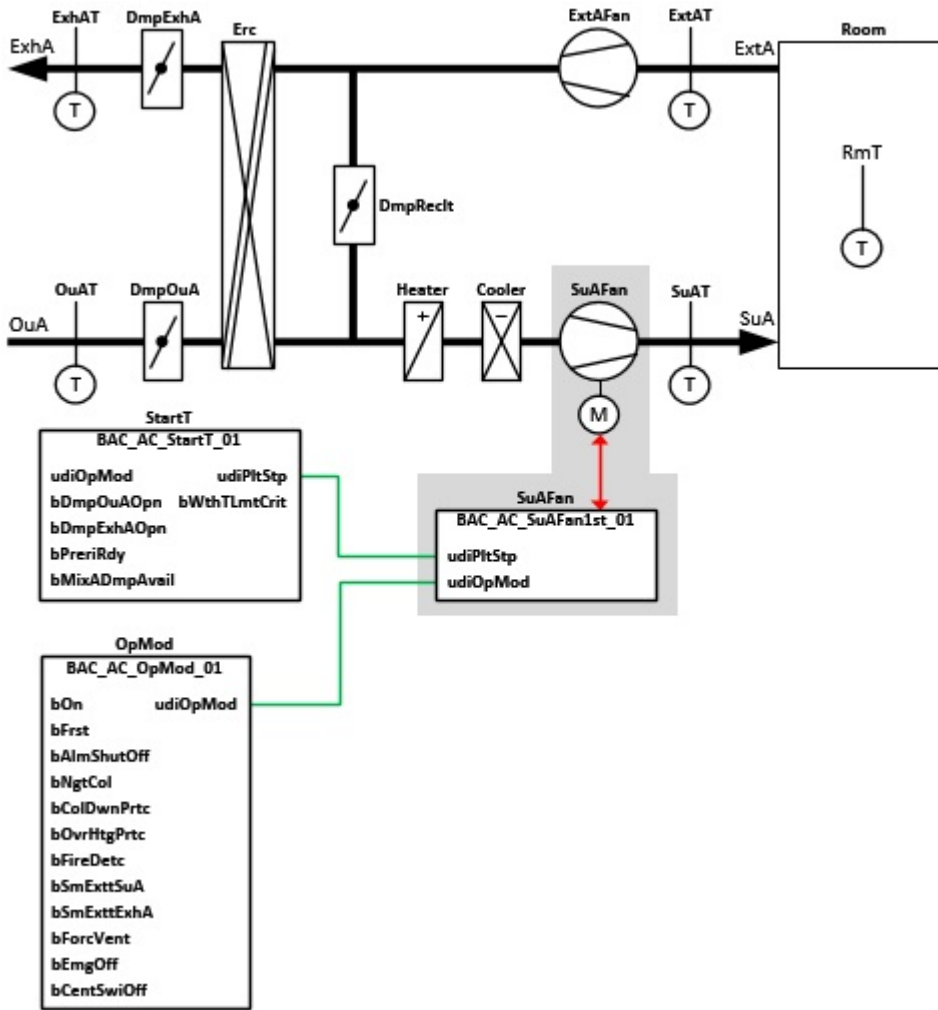
Die wesentlichen Aufgaben des Templates sind:

- Ansteuerung des Zuluftventilators
- Differenzdrucküberwachung

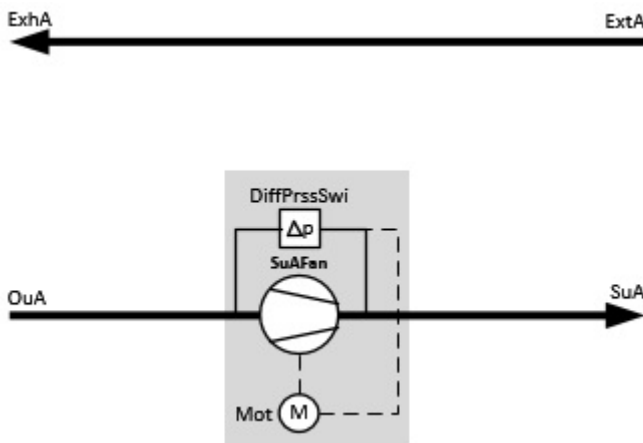
Schnittstelle

BAC_AC_SuAFan1st_01
-udiPltStp
-udiOpMod

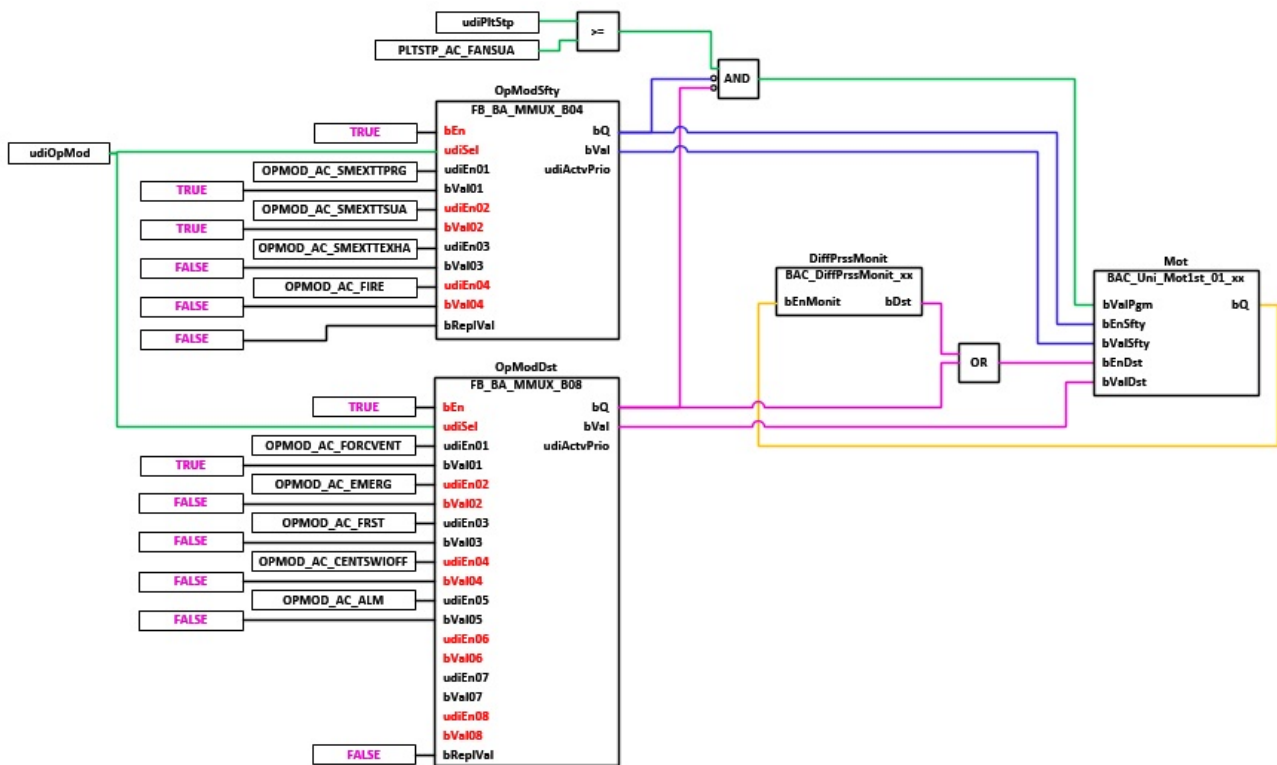
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

udiPltStp      : UDINT;
udiOpMod       : UDINT;
    
```

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm BAC AC StartT 01 [▶ 538].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl BAC AC OpMod 01 [▶ 524].

Programmbeschreibung

Instanz	Typ	Aufgabe		
DiffPrssMonit	BAC_DiffPrssMonit_01 [▶ 441]	Sub-Template Differenzdrucküberwachung mittels Differenzdruckschalter		
Mot	BAC_Uni_Mot1st_01_29 [▶ 576]	Sub-Template Ansteuerung eines einstufigen Motors inklusive Motorlogik.		
OpModSfty	FB_BA_MMUX_B04 [▶ 208]	Der Multiplexer definiert anhand der Anlagenbetriebsart udiOpMod die Sicherheitspriorität für die Ansteuerung des Zuluftventilators Mot		
		udiOpMod	Freigabe	
		OPMOD_AC_SMEXTTPRG	Entrauchung Programm	TRUE
		OPMOD_AC_SMEXTTSUA	Entrauchung Zuluft	TRUE
		OPMOD_AC_SMEXTTEXHA	Entrauchung Fortluft	FALSE
	OPMOD_AC_FIRE	Feuer	FALSE	
OpModDst	FB_BA_MMUX_B08 [▶ 208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Priorität Störung für die Ansteuerung des Zuluftventilators Mot see "Extracted nested table 49"		
		udiOpMod	Freigabe	
		OPMOD_AC_FORCVENT	Zwangsbelüftung	TRUE

Instanz	Typ	Aufgabe
		OPMOD_AC_EMERG Notfall FALSE
		OPMOD_AC_FRST Frost FALSE
		OPMOD_AC_CENTS WIOFF Zentralabschaltung FALSE
		OPMOD_AC_ALM Störung/Alarm FALSE
	>= AND	Das Ergebnis des Netzwerkes ist die Freigabe für den Zuluftventilator anhand der Anlagenschritte udiPltStp
	OR	Das Ergebnis der OR-Operation ist die Freigabe der Störungspriorität des Templates Mot .

Versionshistorie

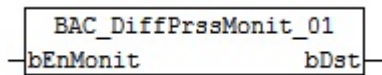
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.28 BAC_DiffPrssMonit_01

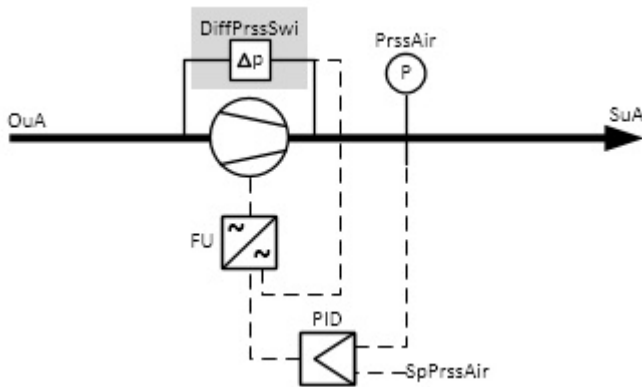
Funktionsbeschreibung

Das Template realisiert eine Differenzdrucküberwachung eines Ventilators oder einer Pumpe mittels eines Differenzdruckschalters.

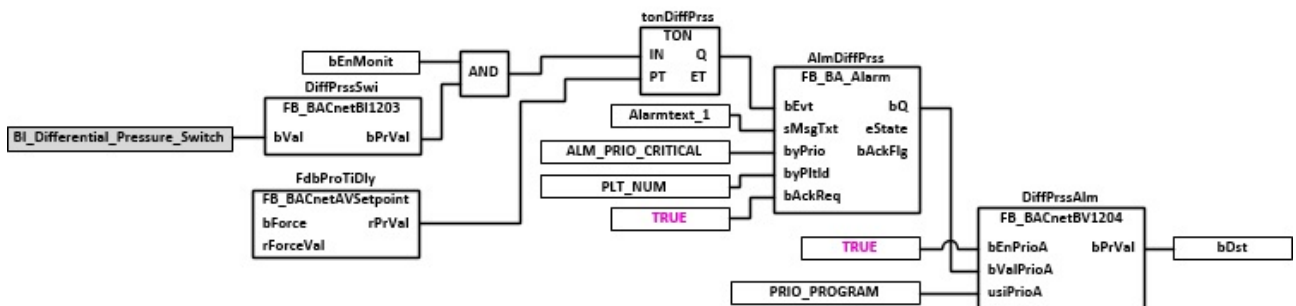
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

```
bEnMonit      : BOOL;
```

bEnMonit: Freigabe der Differenzdrucküberwachung. Der Ventilator oder Pumpe ist eingeschaltet

VAR_OUTPUT

```
bDst          : BOOL;
```

bDst: Anzeige Störmeldung Differenzdrucküberwachung

VAR CONSTANT

```
PLT_NUM       : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [► 182]

Die Auswertung der Alarmergebnisse einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
DiffPrssSwi	FB_BACnetBI1203 [► 72]	BI-Objekt Differenzdruckwächter für die Differenzdrucküberwachung
FdbProTiDly	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe der Zeitverzögerung Prozessrückmeldung <i>FdbProTiDly</i> . Die Meldung des Differenzdruckwächters kann verzögert werden um Druckschwankungen zu überbrücken.
tonDiffPrss	TON	Zeitglied um die Prozessrückmeldung des Differenzdruckwächters verzögert als Alarm auszugeben. Das Zeitglied kann hilfreich sein um Druckschwankungen zu überbrücken.
AlmDiffPrss	FB_BA_Alarm [► 182]	Der Alarmsammelbaustein dient der Einbindung der Differenzdruckstörung in den Sammelalarm der zugehörigen Anlage.
DiffPrssAlm	FB_BACnetBV1204 [► 94]	BV-Objekt zur Anzeige des Alarms Differenzdrucküberwachung

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Differential_Pressure_Switch	BOOL		Eingang	Digitaleingang - Meldung - Differenzdruckschalter

Versionshistorie

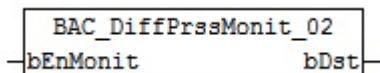
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.29 BAC_DiffPrssMonit_02

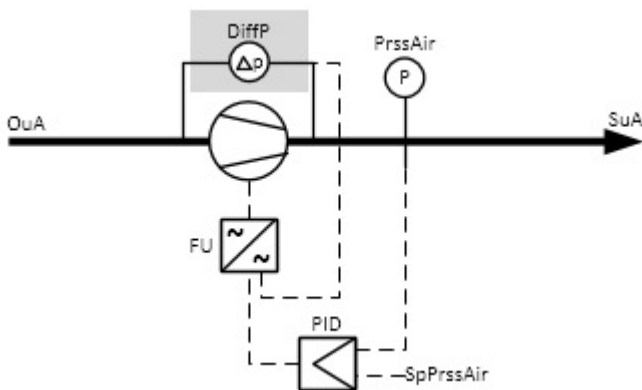
Funktionsbeschreibung

Das Template realisiert eine Differenzdrucküberwachung eines Ventilators oder einer Pumpe mittels eines Differenzdruckfühlers.

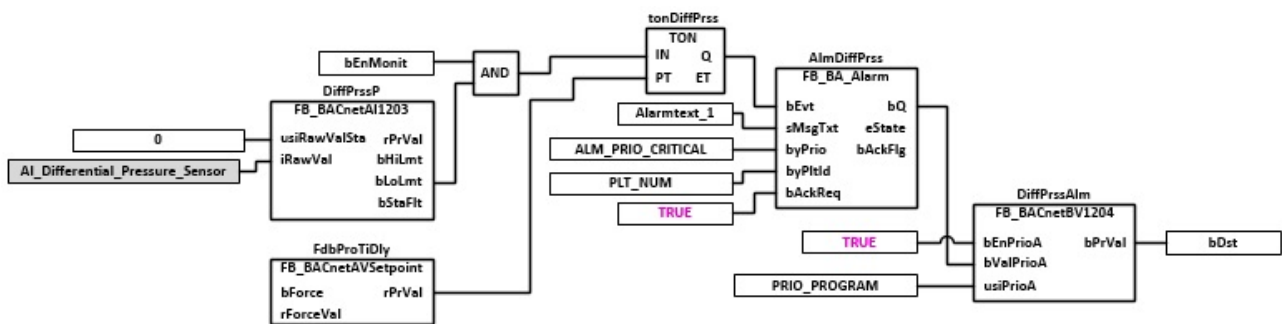
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

bEnMonit : BOOL;

bEnMonit: Freigabe der Differenzdrucküberwachung. Der Ventilator oder Pumpe ist eingeschaltet.

VAR_OUTPUT

bDst : BOOL;

bDst: Anzeige Störmeldung Differenzdrucküberwachung

VAR CONSTANT

PLT_NUM : BYTE := 1;

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmerbausteins FB BA Alarm. [► 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates BAC PltAlm_01 [► 372] mittels des Funktionsbausteins FB BA AlarmPlt. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates BAC_PltComnMsg_01 durch den Funktionsbaustein FB BA ComnMsg [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
DiffP	FB_BACnetAI1203 [► 49]	AI-Objekt Differenzdruckfühler für die Differenzdrucküberwachung. Das BACnet-Property Low Limit löst die Aktion der Differenzdrucküberwachung aus.
FdbProTiDly	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe der Zeitverzögerung Prozessrückmeldung FdbProTiDly. Die Meldung des Differenzdruckfühlers kann verzögert werden um Druckschwankungen zu überbrücken.
tonDiffPrss	TON	Zeitglied um die Prozessrückmeldung des Differenzdruckfühlers verzögert als Alarm auszugeben.
AlmDiffPrss	FB_BA_Alarm [► 182]	Der Alarmerbaustein dient der Einbindung der Differenzdruckstörung in den Sammelalarm der zugehörigen Anlage.
DiffPrssAlm	FB_BACnetBV1204 [► 94]	BV-Objekt zur Anzeige des Alarms Differenzdrucküberwachung

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
AI_Differential_Pressure_Sensor	INT		Eingang	Analogeingang - Messwert - Differenzdruck

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.30 BAC_AC_Filter_01

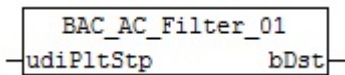
Funktionsbeschreibung

Das Template realisiert eine Filterüberwachung innerhalb einer raumlufttechnischen Anlage mittels eines Differenzdruckschalters. Beim Ansprechen des Differenzdruckschalters wird eine Filterverschmutzung erfasst.

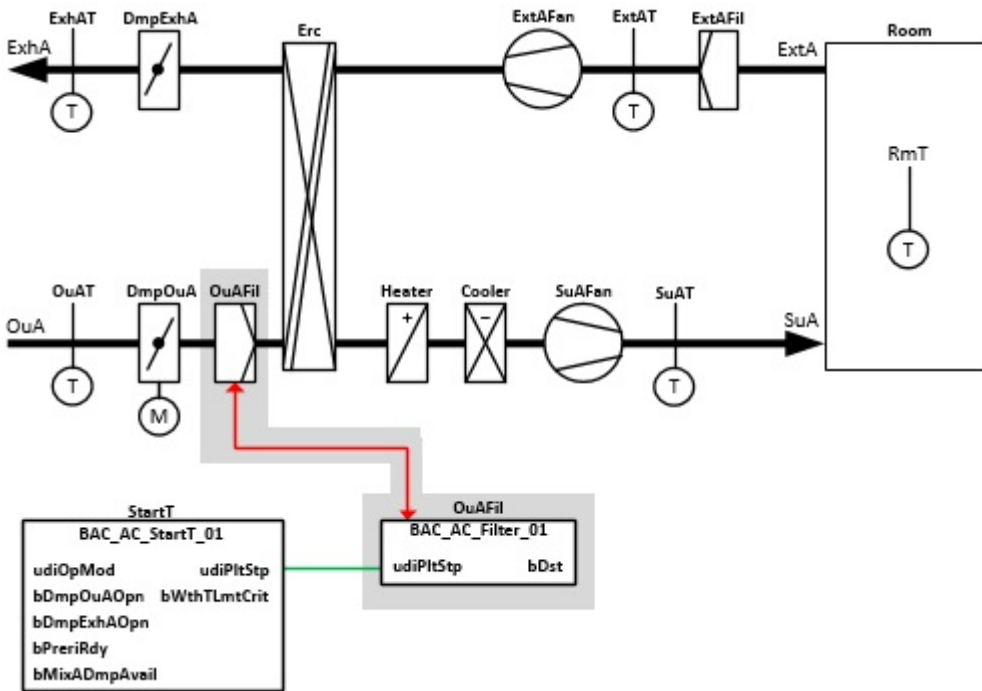
Eine steigende Flanke des Binäreingangs des Differenzdruckschalters setzt den RS-Flip-Flop in Selbsthaltung. Die Verschmutzung des Filters wird dadurch auch bei einem Anlagenstillstand weiterhin angezeigt. Ohne Ansprechen des Differenzdruckschalters wird nach einem erneuten Anlagenstart oder einer Anlagenquittierung das RS-Flip-Flop zurückgesetzt.

Zur Anzeige der Filterverschmutzung in der MBE dient ein BV-Objekt.

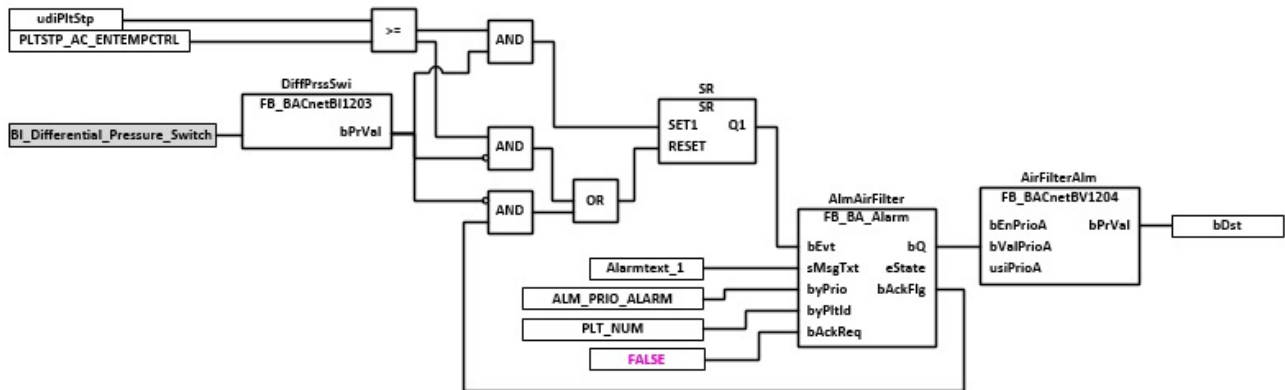
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

udiPltStp : UDINT;

udiPltStp: Über die Eingangsvariable werden dem Template die Anlagenschritte der RLT-Anlage zugeführt. [BAC AC StartT 01 |> 538](#)

VAR_OUTPUT

bDst : BOOL;

bDst: Die Ausgangsvariable zeigt an, dass die Filterüberwachung angesprochen hat und gibt einen Fehler zur Weiterverarbeitung aus.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [▶ 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [▶ 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
DiffPrssSw i	FB_BACnetBI1203 [▶ 72]	BI-Objekt Differenzdruckwächter für die Filterüberwachung
tonDiffPrss	SR	Störmeldespeicher für die Filterüberwachung
AlmAi rFilter	FB_BA_Alarm [▶ 182]	Der Alarmsammelbaustein nimmt die Störung der Filterüberwachung auf und gibt diese weiter über eine interne Variable an den Anlagen- Alarmsammler BAC_PltAlm_01 [▶ 372].
AirFilterAl m	FB_BACnetBV1204 [▶ 94]	BV-Objekt zur Anzeige des Alarms der Filterüberwachung

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Differential_P ressure_Switch	BOOL		Eingang	Digitaleingang - Meldung - Differenzdruckscha lter

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.31 BAC_AC_ExhADmp2P_01_xx**Funktionsbeschreibung**

Das Sub-Template [BAC_AC_ExhADmp2P_01_xx](#) ist für die Ansteuerung und Überwachung einer Fortluftklappe mit einem Federrücklaufantrieb und Endlagenkontrolle. Die Ansteuerung der Klappe erfolgt mit einem binären Ausgang.

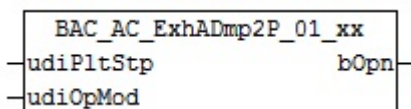


Die Ausgangsvariable bOpn gibt nur dann den tatsächlichen Zustand der Klappe aus, wenn in dem eingesetzten Template die Rückmeldung des Endlagenschalters SwiOpn vorhanden ist. Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang bOpn = TRUE. Dieses hat zur Folge, dass in dem Startprogramm einer Lüftungsanlage [BAC_AC_StartT_01 \[► 538\]](#) die Verzögerungszeit zum Starten der Ventilatoren (SuAFanDlyOn / ExtAFanDlyOn) an die Auffahrzeit der Klappe angepasst werden muss.

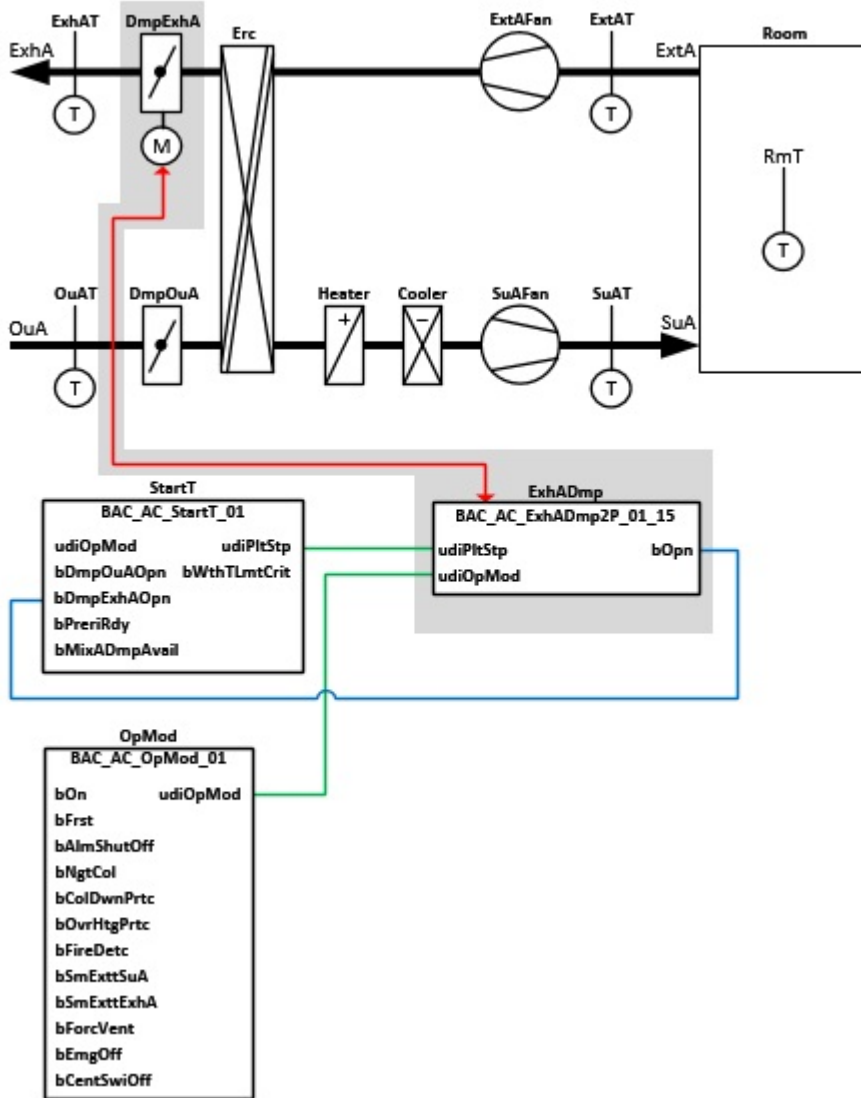
Das Template **BAC_AC_ExhADmp2P_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten der Klappen werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

Optionen	mechanische Vorrangbedie- nung Rückmeldung Handscharter	mechanische Vorrangbedie- nung Rückmeldung Relaisausgang	Endlage Auf	Endlage Zu
Instanz	LocSwi	FdbOut	SwiOpn	SwiClIs
Datenpunkt Typ	BI	BI	BI	BI
	8	4	2	1
BAC_AC_ExhADmp2P_01_00	0	0	0	0
BAC_AC_ExhADmp2P_01_02	0	0	1	0
BAC_AC_ExhADmp2P_01_03	0	0	1	1
BAC_AC_ExhADmp2P_01_08	1	0	0	0
BAC_AC_ExhADmp2P_01_10	1	0	1	0
BAC_AC_ExhADmp2P_01_11	1	0	1	1
BAC_AC_ExhADmp2P_01_12	1	1	0	0
BAC_AC_ExhADmp2P_01_14	1	1	1	0
BAC_AC_ExhADmp2P_01_15	1	1	1	1

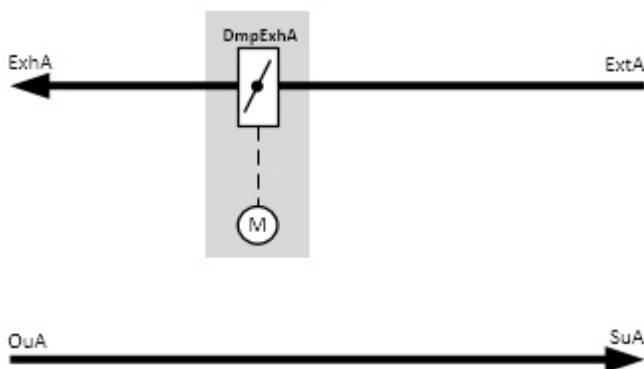
Schnittstelle



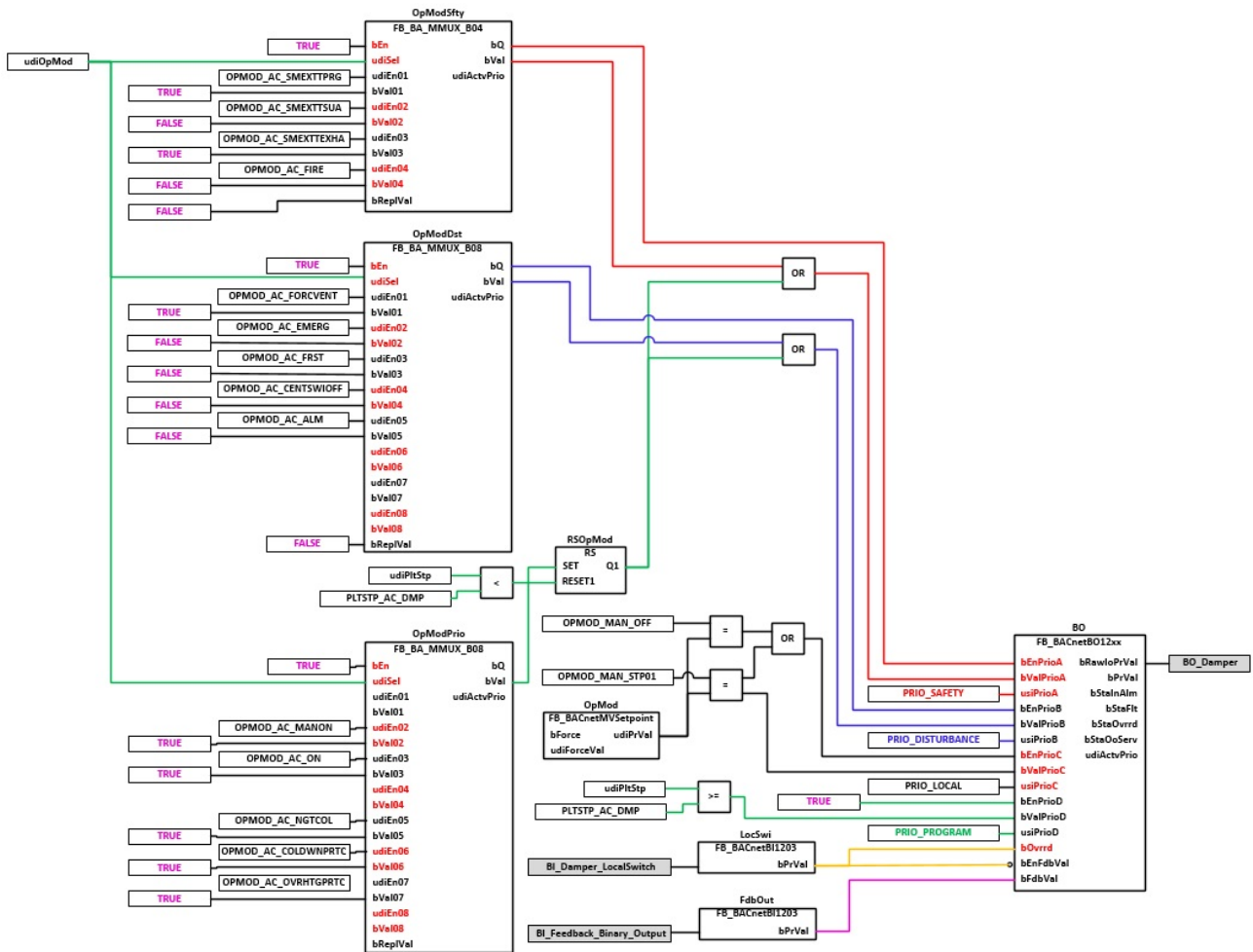
Anlagenschema 01



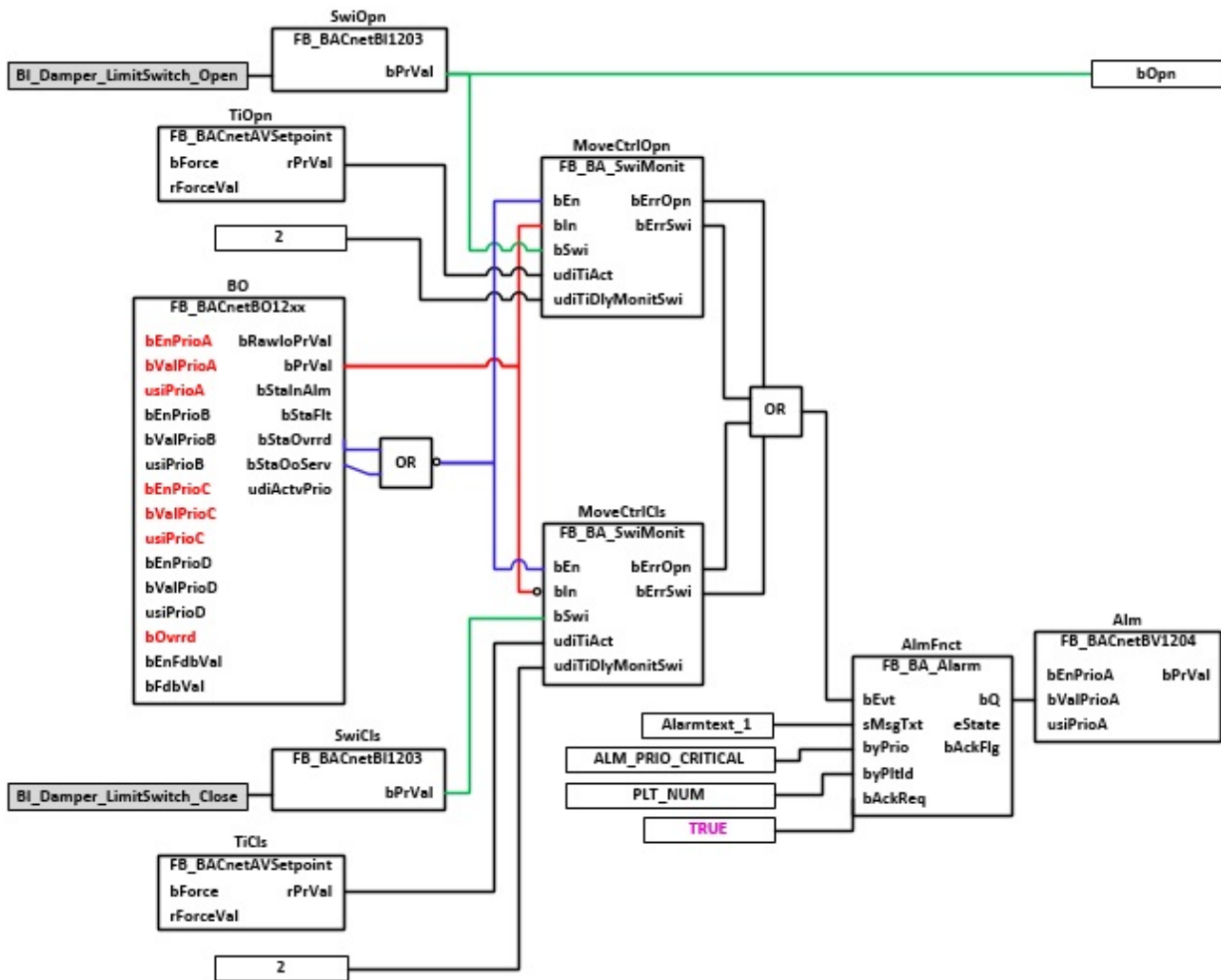
Anlagenschema 02



Blockschaltbild Ansteuerung der Klappe



Blockschaltbild Überwachung der Endlagenschalter



VAR_INPUT

```
udiPltStp : UDINT;
udiOpMod  : UDINT;
```

udiPltStp: Schritte beim Anfahren der RLT-Anlage. Siehe auch [BAC_AC_StartT_01](#) [▶ 538]

udiOpMode: Anlagenbetriebsart. Siehe auch [BAC_AC_OpMod_01](#) [▶ 524]

VAR_OUTPUT

```
bOpn      : BOOL;
```

bOpn: Endlage Offen der Klappe ist erreicht.

Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang **bOpn** = TRUE.

VAR CONSTANT

```
PLT_NUM   : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [▶ 182]

Die Auswertung der Alarmergebnisse einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [▶ 372]

mittels des Funktionsbausteins [FB_BA_AlarmPlt. \[► 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB_BA_ComnMsg \[► 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe		
SwiOpn	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss des Endlagenschalters Offen		
SwiCls	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss des Endlagenschalters Zu		
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung der Klappe von der MBE oder einem lokalen Bediendisplay		
LocSwi	FB_BACnetBI1203 [► 72]	X	BI-Objekt für für die Rückmeldung einer mechanischen Vorrangbedienung. (Hand/Not-Bedienebene)		
FdbOut	FB_BACnetBI1203 [► 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Relais		
TiOpn	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe des Wertes für die Auffahrzeit		
TiCls	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe des Wertes für die Zufahrzeit		
OpModSfty	FB_BA_MMUX_B04 [► 208]		Der Multiplexer definiert die Priorität Safty/Sicherheit der Fortluftklappe in Abhängigkeit der Anlagenbetriebsart udiOpMod		
			udiOpMod	Zustand Klappe	
			OPMOD_AC_SME XTTPRG	Entrauchung Programm	Öffnen
			OPMOD_AC_SME XTTSUA	Entrauchung Zuluft	Schliessen
			OPMOD_AC_SME XTTEXHA	Entrauchung Fortluft	Öffnen
			OPMOD_AC_FIRE	Feuer	Schliessen
OpModDst	FB_BA_MMUX_B08 [► 208]		Der Multiplexer definiert die Priorität Disturbance/Störung der Fortluftklappe in Abhängigkeit der Anlagenbetriebsart udiOpMod		
			udiOpMod	Zustand Klappe	
			OPMOD_AC_FOR CVENT	Zwangsbelüftung	Öffnen
			OPMOD_AC_EME RG	Notfall	Schliessen
			OPMOD_AC_FRST	Frost	Schliessen
			OPMOD_AC_CEN TSWIOFF	Zentralabschaltung	Schliessen
			OPMOD_AC_ALM	Störung	Schliessen
OpModPri o RSOpMod	FB_BA_MMUX_B08 [► 208] RS		Der Multiplexer definiert die Anlagenbetriebsart udiOpMod . Die Funktion RSOpMod soll verhindern, dass die Luftklappe im laufenden Prozess/Ventilatoren bei einem Wechsel der		

Instanz	Typ	optional	Aufgabe												
			Betriebsart sofort schließt. Vom Startprogramm her ist es so vorgesehen, dass bei einem Wechsel der Betriebsart die Anlage herunter fährt und dann neu startet.												
			<table border="1"> <thead> <tr> <th>udiOpMod</th> <th>Zustand Klappe</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_MAN ON</td> <td>Hand ein Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist</td> </tr> <tr> <td>OPMOD_AC_ON</td> <td>Ein Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist</td> </tr> <tr> <td>OPMOD_AC_NGT COL</td> <td>Nachtkühlung Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist</td> </tr> <tr> <td>OPMOD_AC_COL DWNPRTC</td> <td>Stützbetrieb, Auskühlschutz Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist</td> </tr> <tr> <td>OPMOD_AC_OVR HTGPRTC</td> <td>Überhitzungsschutz Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist</td> </tr> </tbody> </table>	udiOpMod	Zustand Klappe	OPMOD_AC_MAN ON	Hand ein Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist	OPMOD_AC_ON	Ein Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist	OPMOD_AC_NGT COL	Nachtkühlung Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist	OPMOD_AC_COL DWNPRTC	Stützbetrieb, Auskühlschutz Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist	OPMOD_AC_OVR HTGPRTC	Überhitzungsschutz Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist
udiOpMod	Zustand Klappe														
OPMOD_AC_MAN ON	Hand ein Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist														
OPMOD_AC_ON	Ein Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist														
OPMOD_AC_NGT COL	Nachtkühlung Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist														
OPMOD_AC_COL DWNPRTC	Stützbetrieb, Auskühlschutz Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist														
OPMOD_AC_OVR HTGPRTC	Überhitzungsschutz Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist														
BO	FB BACnetBO1203 ▶ 82]		<p>BO-Objekt für die Ansteuerung der Klappe</p> <table border="1"> <thead> <tr> <th>Priorität:</th> <th>Freigabe</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td>PRIO_SAFETY (1)</td> <td>siehe Multiplexer OpModSftyudiOpMod</td> <td>siehe Multiplexer OpModSfty Zustand Klappe</td> </tr> <tr> <td>PRIO_DISTURBANCE (3)</td> <td>siehe Multiplexer OpModDstudiOpMod</td> <td>siehe Multiplexer OpModDst Zustand Klappe</td> </tr> <tr> <td>PRIO_LOCAL (8)</td> <td>Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt</td> <td>Öffnen, wenn OpMod_udiPrVal = OPMOD_MAN_STP01 Ansonsten schließen</td> </tr> </tbody> </table>	Priorität:	Freigabe	Wert	PRIO_SAFETY (1)	siehe Multiplexer OpModSftyudiOpMod	siehe Multiplexer OpModSfty Zustand Klappe	PRIO_DISTURBANCE (3)	siehe Multiplexer OpModDstudiOpMod	siehe Multiplexer OpModDst Zustand Klappe	PRIO_LOCAL (8)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt	Öffnen, wenn OpMod_udiPrVal = OPMOD_MAN_STP01 Ansonsten schließen
Priorität:	Freigabe	Wert													
PRIO_SAFETY (1)	siehe Multiplexer OpModSftyudiOpMod	siehe Multiplexer OpModSfty Zustand Klappe													
PRIO_DISTURBANCE (3)	siehe Multiplexer OpModDstudiOpMod	siehe Multiplexer OpModDst Zustand Klappe													
PRIO_LOCAL (8)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt	Öffnen, wenn OpMod_udiPrVal = OPMOD_MAN_STP01 Ansonsten schließen													

Instanz	Typ	optional	Aufgabe
			<p>hat den Wert OPMOD_MAN_OFF (Hand Aus) 2. Das MV-Objekt hat den Wert OPMOD_MAN_STP01(Hand Ein)</p>
			<p>PRIO_PROGRAM (15) TRUE</p> <p>Wenn der Anlagenschritt udiPltStp >= PLTSTP_AC_DMP ist, dann soll die Klappe öffnen</p>
MoveCtrlOpen	FB_BA_SwiMonit [P_155]	X	Funktionsbaustein welcher die Endlage Auf der Klappe überwacht
MoveCtrlClose	FB_BA_SwiMonit [P_155]	X	Funktionsbaustein welcher die Endlage Zu der Klappe überwacht
AlmFunct	FB_BA_Alarm [P_182]	x	Der Funktionsbaustein AlmFunct erfasst das Ereignis der Überwachung der Endlagenschalter. Aktionen, die nach dem Eingang der Endlagenschalterstörung erfolgen soll, können im Template am Funktionsbaustein AlmFunct parametrieret werden.
Alm	FB_BACnetBV1204 [P_94]	x	BV-Objekt zur Anzeige der Klappenstörung in der MBE

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild
BI_Damper_LimitSwitch_Open	BOOL	X	Eingang Digitaleingang - Schalter Klappe Auf - Meldung - Betaetigt/Nicht betaetigt
BI_Damper_LimitSwitch_Close	BOOL	X	Eingang Digitaleingang - Schalter Klappe Zu - Meldung - Betaetigt/Nicht betaetigt
BI_Damper_LocalSwitch	BOOL	X	Eingang Digitaleingang - Schalter Hand Klappe - Meldung - Hand/Auto
BI_Feedback_Binary_Output	BOOL	X	Eingang Digitaleingang - Klappe Schaltbefehl - Rückmeldung - Ein/Aus
BO_Damper	BOOL		Ausgang Digitalausgang - Klappe - Schaltbefehl - Ein/Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.32 BAC_AC_OuADmp2P_01_xx

Funktionsbeschreibung

Das Sub-Template **BAC_AC_OuADmp2P_01_xx** ist für die Ansteuerung und Überwachung einer Außenluftklappe mit einem Federrücklaufantrieb und Endlagenkontrolle. Die Ansteuerung der Klappe erfolgt mit einem binären Ausgang.

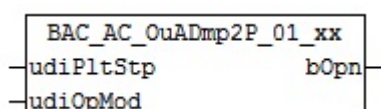


Die Ausgangsvariable bOpn gibt nur dann den tatsächlichen Zustand der Klappe aus, wenn in dem eingesetzten Template die Rückmeldung des Endlagenschalters SwiOpn vorhanden ist. Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang bOpn = TRUE. Dieses hat zur Folge, dass in dem Startprogramm einer Lüftungsanlage **BAC_AC_StartT_01** [► 538] die Verzögerungszeit zum Starten der Ventilatoren (SuAFanDlyOn / ExtAFanDlyOn) an die Auffahrzeit der Klappe angepasst werden muss.

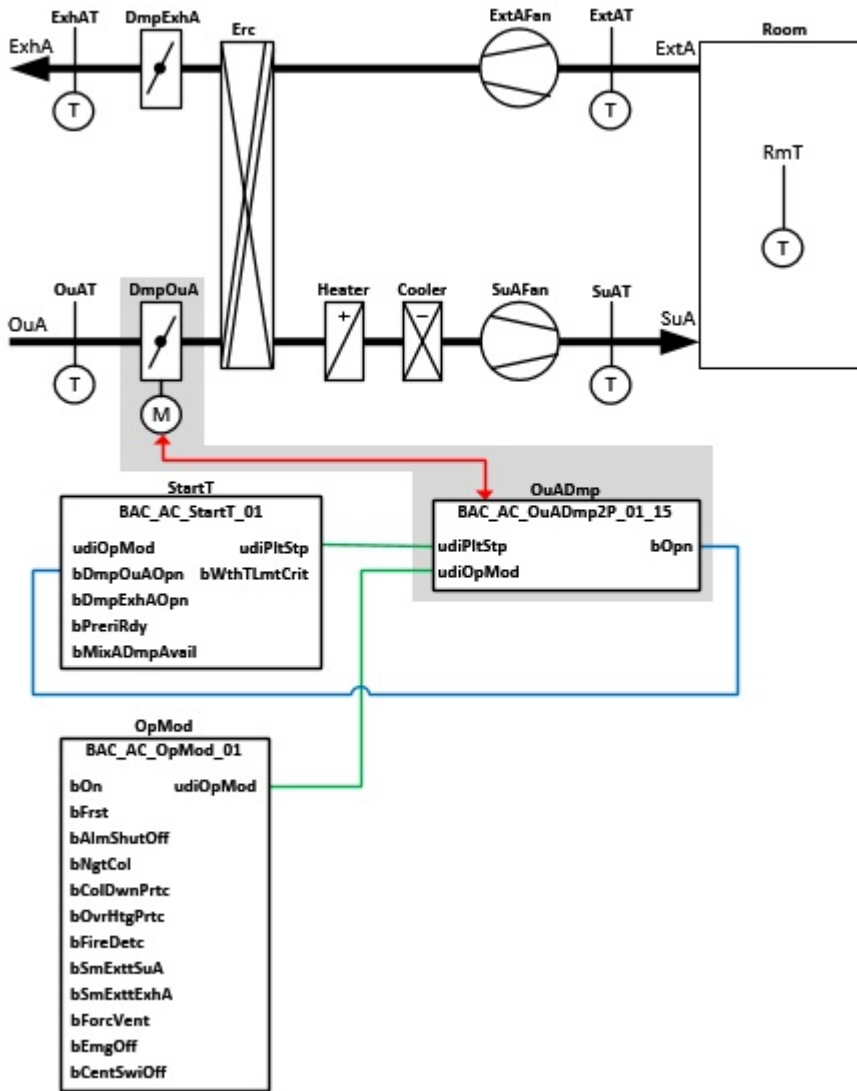
Das Template **BAC_AC_OuADmp2P_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten der Klappen werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

Optionen	mechanische Vorrangbedien- ung Rückmeldung Handscharter	mechanische Vorrangbedien- ung Rückmeldung Relaisausgang	Endlage Auf	Endlage Zu
Instanz	LocSwi	FdbOut	SwiOpn	SwiClIs
Datenpunkt Typ	BI	BI	BI	BI
	8	4	2	1
BAC_AC_OuADmp 2P_01_00	0	0	0	0
BAC_AC_OuADmp 2P_01_02	0	0	1	0
BAC_AC_OuADmp 2P_01_03	0	0	1	1
BAC_AC_OuADmp 2P_01_08	1	0	0	0
BAC_AC_OuADmp 2P_01_10	1	0	1	0
BAC_AC_OuADmp 2P_01_11	1	0	1	1
BAC_AC_OuADmp 2P_01_12	1	1	0	0
BAC_AC_OuADmp 2P_01_14	1	1	1	0
BAC_AC_OuADmp 2P_01_15	1	1	1	1

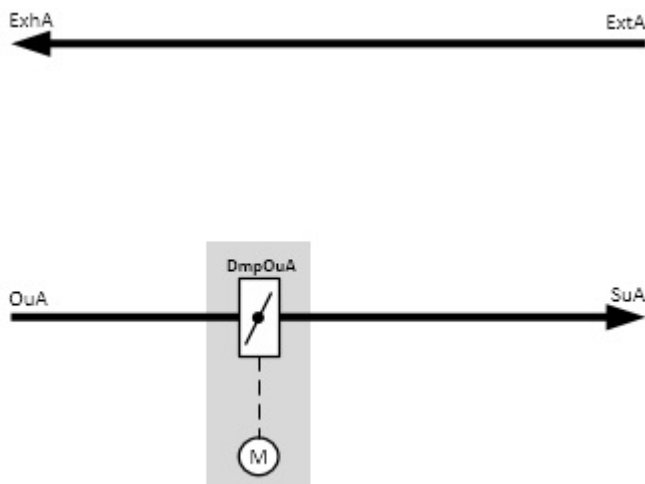
Schnittstelle



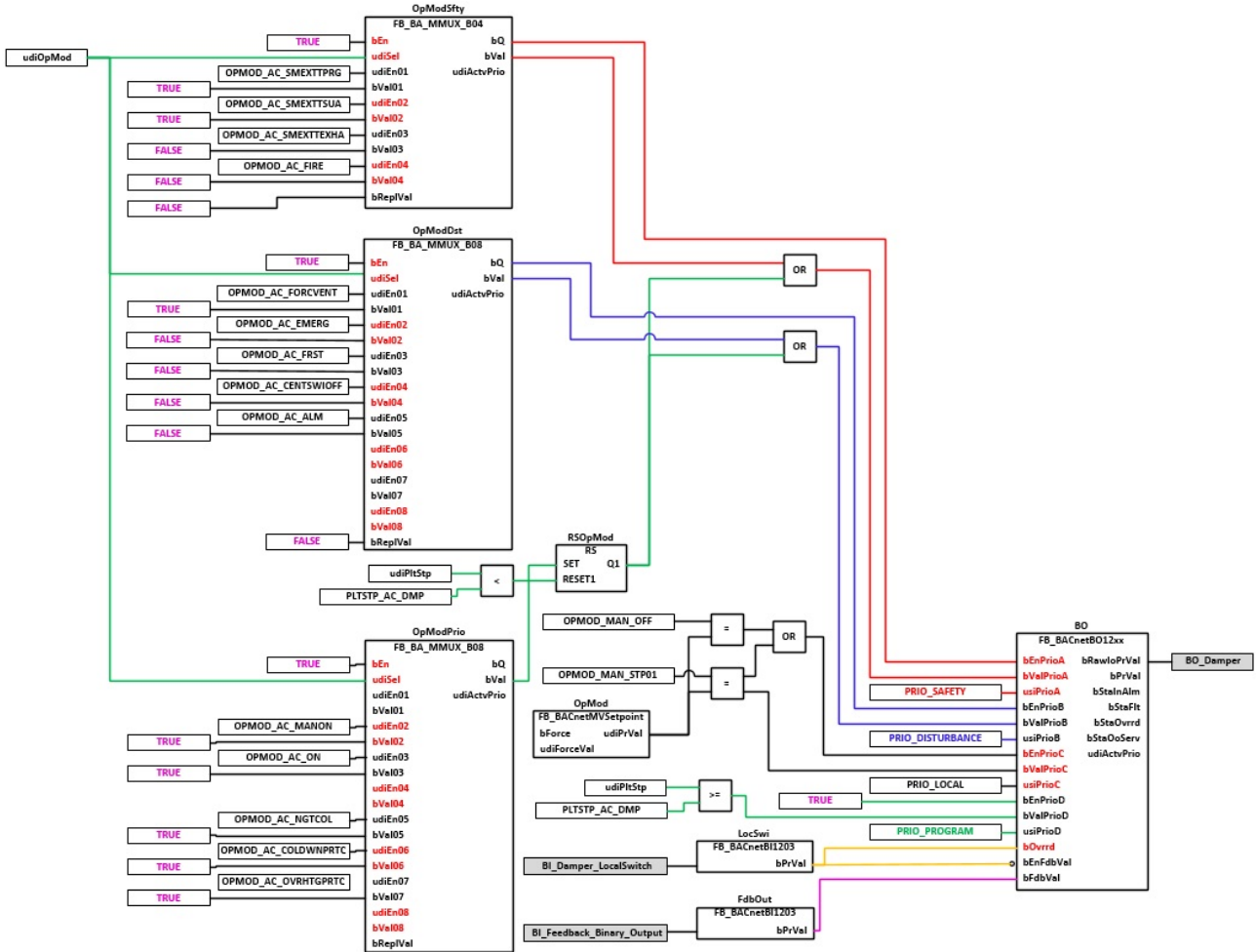
Anlagenschema 01



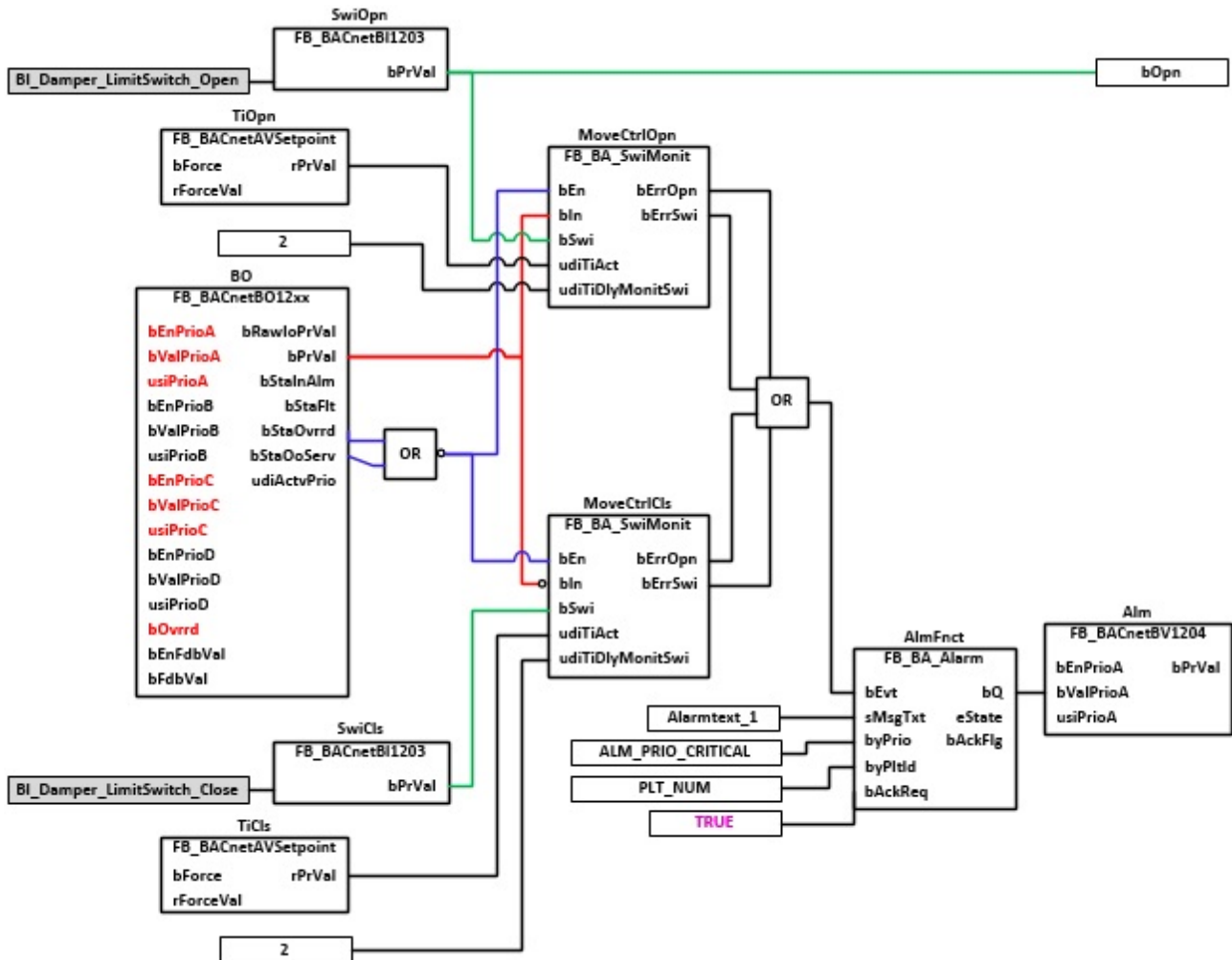
Anlagenschema 02



Blockschaltbild Ansteuerung der Klappe



Blockschaltbild Überwachung der Endlagenschalter



VAR_INPUT

```
udiPltStp : UDINT;
udiOpMod : UDINT;
```

udiPltStp: Schritte beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartT_01 \[▶ 538\]](#)

udiOpMode: Anlagenbetriebsart. Siehe auch [BAC AC OpMod_01 \[▶ 524\]](#)

VAR_OUTPUT

```
bOpn : BOOL;
```

bOpn: Endlage Offen der Klappe ist erreicht.

Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang **bOpn = TRUE**.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm. \[▶ 182\]](#)

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01 \[▶ 372\]](#)

mittels des Funktionsbausteins [FB_BA_AlarmPlt. \[► 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB_BA_CmnMsg \[► 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe																		
SwiOpn	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss des Endlagenschalters Offen																		
SwiCls	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss des Endlagenschalters Zu																		
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung der Klappe von der MBE oder einem lokalen Bediendisplay																		
LocSwi	FB_BACnetBI1203 [► 72]	X	BI-Objekt für für die Rückmeldung einer mechanischen Vorrangbedienung. (Hand/Not-Bedienebene)																		
FdbOut	FB_BACnetBI1203 [► 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Relais																		
TiOpn	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe des Wertes für die Auffahrzeit																		
TiCls	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe des Wertes für die Zufahrzeit																		
OpModSfty	FB_BA_MMUX_B04 [► 208]		Der Multiplexer definiert die Priorität Safty/Sicherheit der Außenluftklappe in Abhängigkeit der Anlagenbetriebsart udiOpMod																		
			<table border="1"> <thead> <tr> <th colspan="2">udiOpMod</th> <th>Zustand Klappe</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_SME XTTPRG</td> <td>Entrauchung Programm</td> <td>Öffnen</td> </tr> <tr> <td>OPMOD_AC_SME XTTSUA</td> <td>Entrauchung Zuluft</td> <td>Schließen</td> </tr> <tr> <td>OPMOD_AC_SME XTTEXHA</td> <td>Entrauchung Außenluft</td> <td>Öffnen</td> </tr> <tr> <td>OPMOD_AC_FIRE</td> <td>Feuer</td> <td>Schließen</td> </tr> </tbody> </table>	udiOpMod		Zustand Klappe	OPMOD_AC_SME XTTPRG	Entrauchung Programm	Öffnen	OPMOD_AC_SME XTTSUA	Entrauchung Zuluft	Schließen	OPMOD_AC_SME XTTEXHA	Entrauchung Außenluft	Öffnen	OPMOD_AC_FIRE	Feuer	Schließen			
udiOpMod		Zustand Klappe																			
OPMOD_AC_SME XTTPRG	Entrauchung Programm	Öffnen																			
OPMOD_AC_SME XTTSUA	Entrauchung Zuluft	Schließen																			
OPMOD_AC_SME XTTEXHA	Entrauchung Außenluft	Öffnen																			
OPMOD_AC_FIRE	Feuer	Schließen																			
OpModDst	FB_BA_MMUX_B08 [► 208]		Der Multiplexer definiert die Priorität Disturbance/Störung der Außenluftklappe in Abhängigkeit der Anlagenbetriebsart udiOpMod																		
			<table border="1"> <thead> <tr> <th colspan="2">udiOpMod</th> <th>Zustand Klappe</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_FOR CVENT</td> <td>Zwangsbelüftung</td> <td>Öffnen</td> </tr> <tr> <td>OPMOD_AC_EME RG</td> <td>Notfall</td> <td>Schließen</td> </tr> <tr> <td>OPMOD_AC_FRS T</td> <td>Frost</td> <td>Schließen</td> </tr> <tr> <td>OPMOD_AC_CEN TSWIOFF</td> <td>Zentralabschaltung</td> <td>Schließen</td> </tr> <tr> <td>OPMOD_AC_ALM</td> <td>Störung</td> <td>Schließen</td> </tr> </tbody> </table>	udiOpMod		Zustand Klappe	OPMOD_AC_FOR CVENT	Zwangsbelüftung	Öffnen	OPMOD_AC_EME RG	Notfall	Schließen	OPMOD_AC_FRS T	Frost	Schließen	OPMOD_AC_CEN TSWIOFF	Zentralabschaltung	Schließen	OPMOD_AC_ALM	Störung	Schließen
udiOpMod		Zustand Klappe																			
OPMOD_AC_FOR CVENT	Zwangsbelüftung	Öffnen																			
OPMOD_AC_EME RG	Notfall	Schließen																			
OPMOD_AC_FRS T	Frost	Schließen																			
OPMOD_AC_CEN TSWIOFF	Zentralabschaltung	Schließen																			
OPMOD_AC_ALM	Störung	Schließen																			
OpModPri o RSOpMod	FB_BA_MMUX_B08 [► 208] RS		Der Multiplexer definiert die Anlagenbetriebsart udiOpMod . Die Funktion RSOpMod soll verhindern, dass die Luftklappe im laufenden Prozess/Ventilatoren bei einem Wechsel der Betriebsart sofort schließt. Vom Startprogramm her ist es so vorgesehen, dass bei einem Wechsel der Betriebsart die Anlage herunterfährt und dann neu startet.																		

Instanz	Typ	optional	Aufgabe		
			udiOpMod		Zustand Klappe
			OPMOD_AC_MAN ON	Hand ein	Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist
			OPMOD_AC_ON	Ein	Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist
			OPMOD_AC_NGT COL	Nachtkühlung	Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist
			OPMOD_AC_COL DWNPRTC	Stützbetrieb,Auskühlschutz	Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist
			OPMOD_AC_OVR HTGPRTC	Überhitzungsschutz	Klappe schließt erst, wenn der Anlagenschritt udiPltStp < PLTSTP_AC_DMP nicht mehr gefordert ist
BO	FB_BACnetBO1203 [▶ 82]		BO-Objekt für die Ansteuerung der Klappe		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	siehe Multiplexer OpModSftyudiOpMod	siehe Multiplexer OpModSfty Zustand Klappe
			PRIO_DISTURBANCE (3)	siehe Multiplexer OpModDstudiOpMod	siehe Multiplexer OpModDst Zustand Klappe
			PRIO_LOCAL (8)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt hat den Wert OPMOD_MAN_OF	Öffnen, wenn OpMod_udiPrVal = OPMOD_MAN_ST P01 Ansonsten schließen

Instanz	Typ	optional	Aufgabe		
				F (Hand Aus) 2. Das MV-Objekt hat den Wert OPMOD_MAN_ST P01 (Hand Ein)	
			PRIO_PROGRAM (15)	TRUE	Wenn der Anlagenschritt udiPltStp >= PLTSTP_AC_DMP ist, dann soll die Klappe öffnen
MoveCtrlOpen	FB_BA_SwiMonit [► 155]	X	Funktionsbaustein welcher die Endlage Auf der Klappe überwacht		
MoveCtrlClose	FB_BA_SwiMonit [► 155]	X	Funktionsbaustein welcher die Endlage Zu der Klappe überwacht		
AlmFnc	FB_BA_Alarm [► 182]	x	Der Funktionsbaustein AlmFnc erfasst das Ereignis der Überwachung der Endlagenschalter. Aktionen, die nach dem Eingang der Endlagenschalterstörung erfolgen soll, können im Template am Funktionsbaustein AlmFnc parametrisiert werden.		
Alm	FB_BACnetBV1204 [► 94]	x	BV-Objekt zur Anzeige der Klappenstörung in der MBE		

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Damper_LimitSwitch_Open	BOOL	X	Eingang	Digitaleingang - Schalter Klappe Auf - Meldung - Betätigt/Nicht betätigt
BI_Damper_LimitSwitch_Close	BOOL	X	Eingang	Digitaleingang - Schalter Klappe Zu - Meldung - Betätigt/Nicht betätigt
BI_Damper_LocalSwitch	BOOL	X	Eingang	Digitaleingang - Schalter Hand Klappe - Meldung - Hand/Auto
BI_Feedback_Binary_Output	BOOL	X	Eingang	Digitaleingang - Klappe Schaltbefehl - Rückmeldung - Ein/Aus
BO_Damper	BOOL		Ausgang	Digitalausgang - Klappe - Schaltbefehl - Ein/Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.33 BAC_AC_CoIH_PID_01

Funktionsbeschreibung

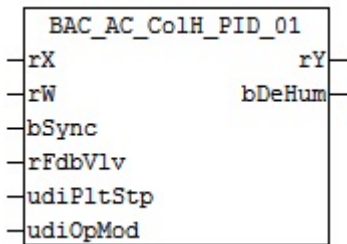
Das Sub-Template **BAC_AC_CoIH_PID_01** ist der Entfeuchterregler für einen Kühler mit Entfeuchtung. Die Referenzierung des Sollwertes, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte **X**, **W** und **Y**.

Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart **udiOpMod** und der globalen Feuchte-Kommunikationsstruktur **g_stSeqLinkH[PLT_NUM]**. Diese Daten- und Befehlsstruktur ist das Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein **FB_BA_SeqLink** [► 171] einer Anlage.

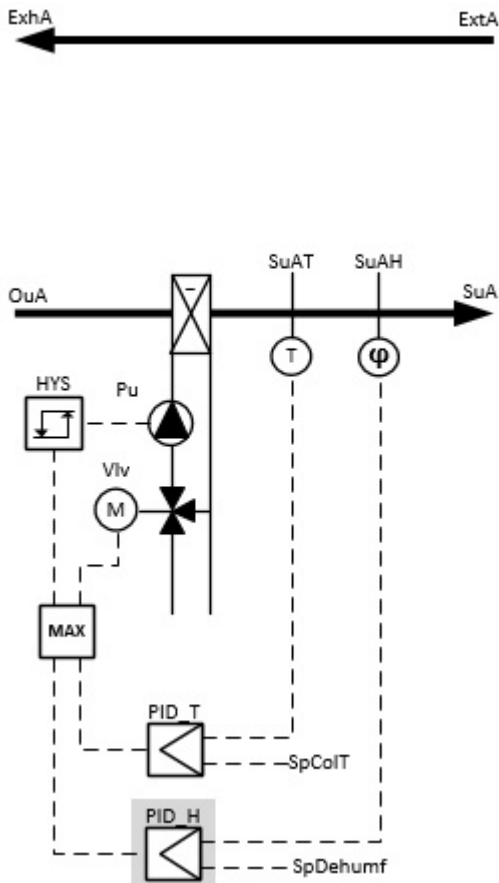
Zur Anzeige der Reglerfreigabe dient das BACnet-BV-Objekt **En**.

Die Grenzwertüberwachung des PID-Reglers wird in Abhängigkeit des Anlagenanfahrprozesses von dem Funktionsbaustein **EnEvtEn** gesteuert.

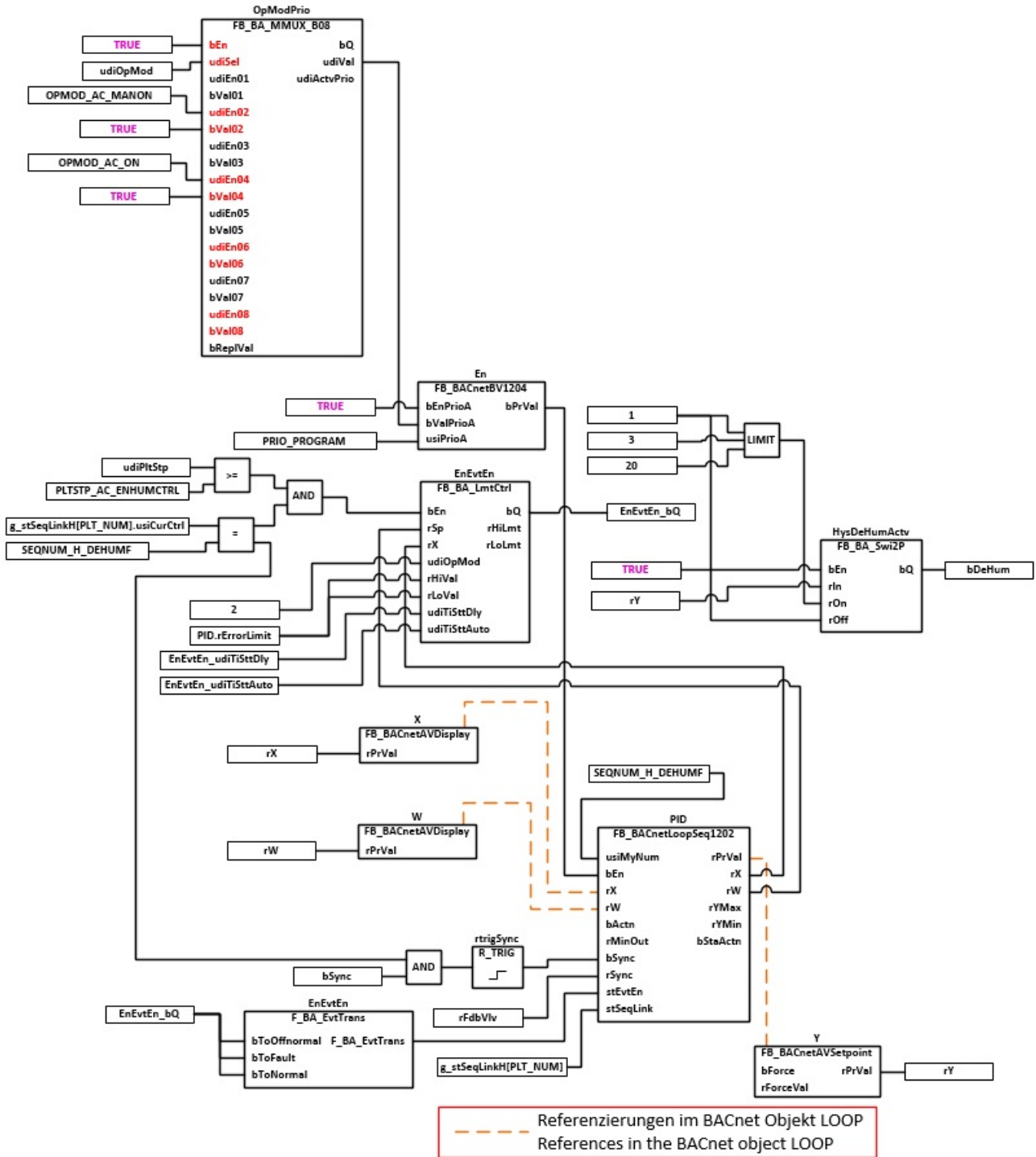
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

```

rX      : REAL;
rW      : REAL;
bSync   : BOOL;
rFdbVlv : REAL;
udiPltStp : UDINT;
udiOpMod : UDINT;
    
```

rX: Messwert Zulufttemperatur

rW: Sollwert der Zulufttemperatur

bSync: Eingang für die Synchronisation des Reglers

rFdbVlv: Stellungsrückmeldung Aktor

udiPltStp: Schritte beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartTH_01](#) [▶ 543].

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod_01](#) [▶ 524]

VAR_OUTPUT

```
rY : REAL;
```

rY: Ausgabe der Stellgröße für das Regelventil

bDeHum: Entfeuchtebetrieb aktiv.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm](#). [▶ 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01](#) [▶ 372] mittels des Funktionsbausteins [FB BA AlarmPlt](#). [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB BA ComnMsg](#) [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur **g_stSeqLinkH[PLT_NUM]** dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein **FB_BA_SeqLink** zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe		
X	FB BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts		
W	FB BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts		
OpModPri o	FB BA MMUX B08 [▶ 208]	Der Multiplexer definiert die Freigabebedingungen des Sequenzreglers in Abhängigkeit der Anlagenbetriebsart.		
		udiOpMod	Freigabe	Bemerkung
		OPMOD AC MANON [▶ 364]	Hand ein	TRUE Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet
		OPMOD AC ON [▶ 364]	Ein	TRUE Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm
En	FB BACnetBV1204 [▶ 94]	Das BV-Objekt dient zur Anzeige und Aktivierung der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay. Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart udiOpMod und der globalen Kommunikationsstruktur g_stSeqLinkH[PLT_NUM] .		

Instanz	Typ	Aufgabe
EnEvtEn	FB BA LmtCtrl [▶ 233]	<p>Das BACnet-Loop-Objekt PID überwacht die Funktion der Regelung in dem es den Sollwert W und den Istwert X miteinander vergleicht. Ist die Abweichung $W-X$ größer als das Property ErrorLimit, dann sendet das Loop-Objekt eine Meldung an die MBE.</p> <p>Beim Anlagenstillstand, im Moment des Starts und bis zum eingeregelter Zustand der Anlage wird das Melden des Loop-Objekts unterdrückt, damit keine falschen Meldungen an die MBE gesendet werden. Erst wenn die RLT-Anlage vollständig hochgefahren ist und die Regelung sich eingependelt hat, wird das Melden des Loop-Objekts freigegeben. Zusätzlich wird das Melden aktiviert, wenn die Regelung nach langer Zeit nicht in dem vom Property ErrorLimit definierten Bereich um den Sollwert gelangt ist.</p> <p>Die Freigabe des objektinternen Meldens erfolgt indem auf die BACnetEventTransitionBits des Loop-Objekts geschrieben wird.</p> <p>Für die Freigabe des Meldens vom Loop-Objekt müssen die folgenden Bedingungen erfüllt sein:</p> <p>1. Das Anlagenstartprogramm <u>BAC_AC_StartTH_01 [▶ 543]</u> hat die Überwachung der Regelung und die Grenzwertüberwachung der Fühler freigegeben udiPltStp >= PLTSTP_AC_ENHUMCTRL und Der Entfeuchterregler ist der aktive Regler in der Regelsequenz <u>g_stSeqLinkH[PLT_NUM] [▶ 364].usiCurCtrl = SEQNUM_H_DEHUMF</u> und Die absolute Zulufffeuchte hat sich dem Sollwert so weit angenähert, dass sie sich in einem Bereich zwischen rSp - ErrorLimit und rSp + ErrorLimit eingependelt hat. und Die absolute Zulufffeuchte muss mindestens für die Dauer von EnEvtEn_udiTiSttDly innerhalb des Bereiches von rSp - ErrorLimit und rSp + ErrorLimit verblieben sein.</p> <p>2. Der Timer EnEvtEn_udiTiSttAuto ist abgelaufen und die Regelung hat ihren Sollwertbereich nicht erreicht.</p>
	F_BA_EvtTrans	schreibt auf die BACnetEventTransitionBits to_offnormal, to_fault und to_normal des Loop-Objekts. Dazu muss der Eingang bEnEvtEn = TRUE sein
PID	FB BACnetLoopSeq1202 [▶ 103]	Sequenzregler absolute Zulufffeuchte Kühler.
rtrigSync	R_TRIG	Bei einer steigenden Flanke am Eingang bSync wird das Loop-Objekt auf den Wert von IrSync aufsynchronisiert. Wenn das Regelventil des Kühlers durch Schreiben einer höheren Priorität auf das zugehörige AO-Objekt von der MBE oder durch Betätigen der örtlichen Vorrangbedienung übersteuert wurde, weicht die aktuelle Position des Regelventils von dem Ausgang des Loop-Objekts ab. Mit den Variablen bSync und rFdbVlv kann die Synchronität zwischen der Position des Regelventils und dem Regler wieder hergestellt werden.
Y	FB BACnetAVSetpoint [▶ 70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts

Instanz	Typ	Aufgabe
HysDeHumActv	FB_BA_Swi2P [▶ 148] LIMIT	2 Punktschalter der anhand einer Hysterese (5) den Ausgang bDeHum in Abhängigkeit der Stellgröße rY ein- und ausschaltet. Ist die Stellgröße < 2, so wird bDeHum ausgeschaltet. Die Funktion LIMIT limitiert den oberen Grenzwert für den 2 Punktschalter HysDeHumActv .

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.34 BAC_AC_CoIT_01

Anwendung

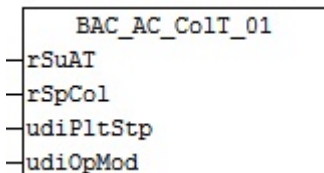
Das Template **BAC_AC_CoIT_01** dient der Steuerung und Regelung eines Kaltwasser-Luftkühlers ohne Entfeuchterregelung.

Das Template **BAC_AC_CoIT_01** ist ein Aufrufftemplate.

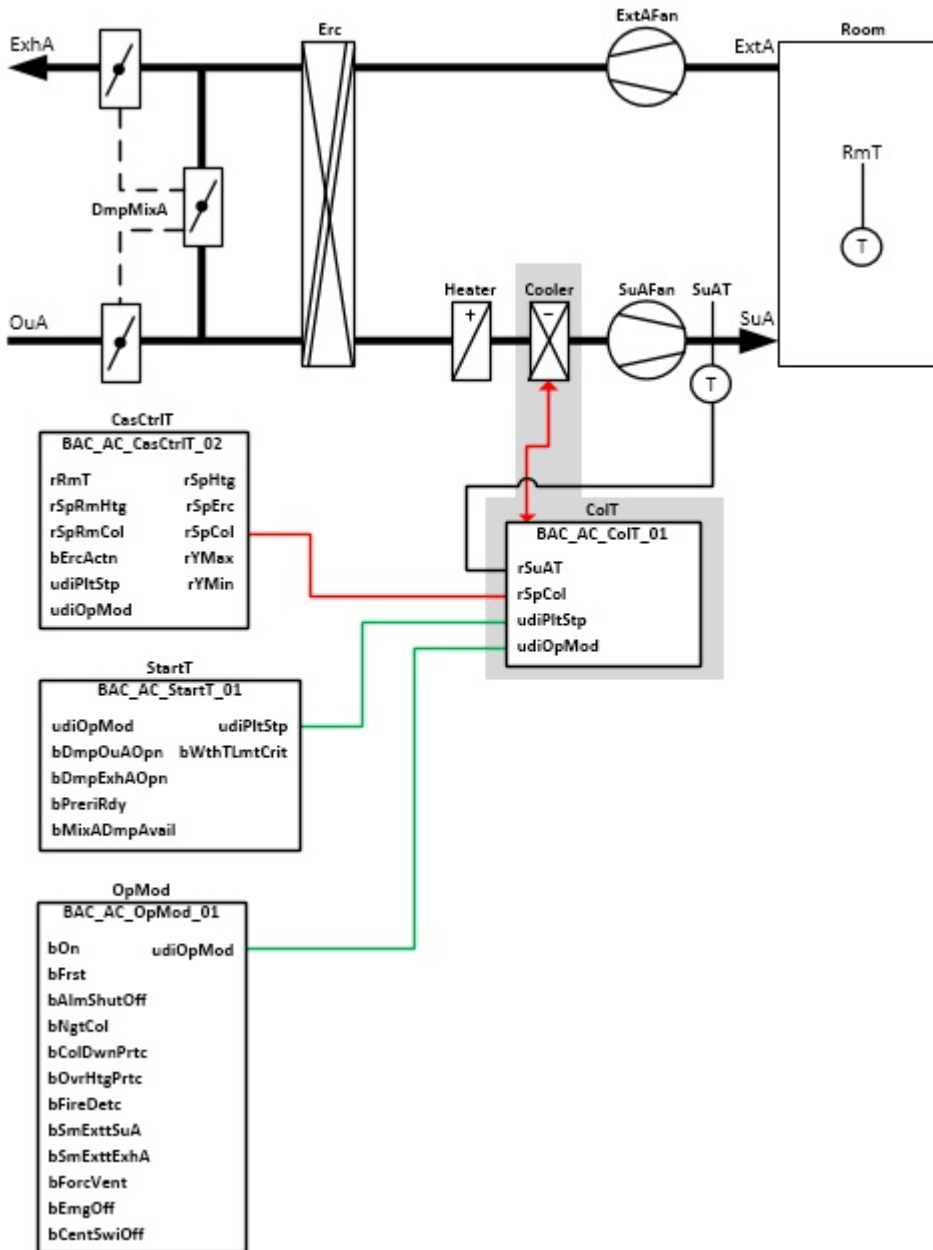
Innerhalb des Aufrufftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

- **CoITPID** Regelung der Zulufttemperatur
- **CoITPu** Ansteuerung der Kühlerpumpe
- **CoITViv** Ansteuerung eines stetigen Regelventils

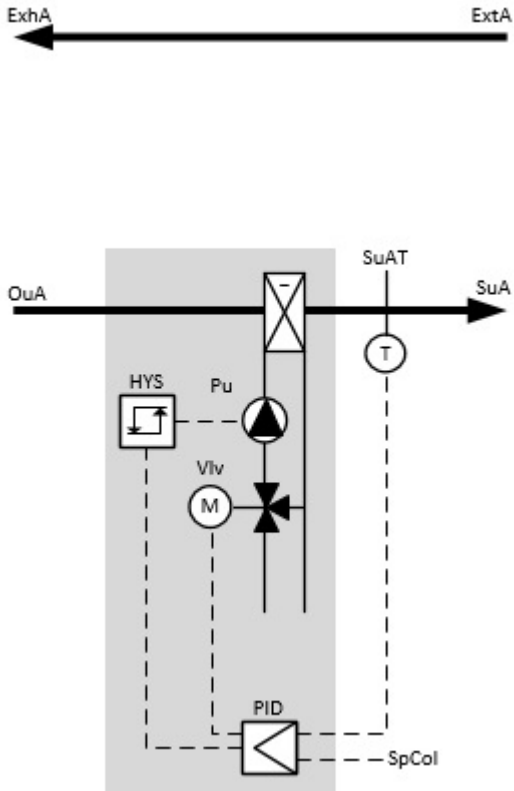
Schnittstelle



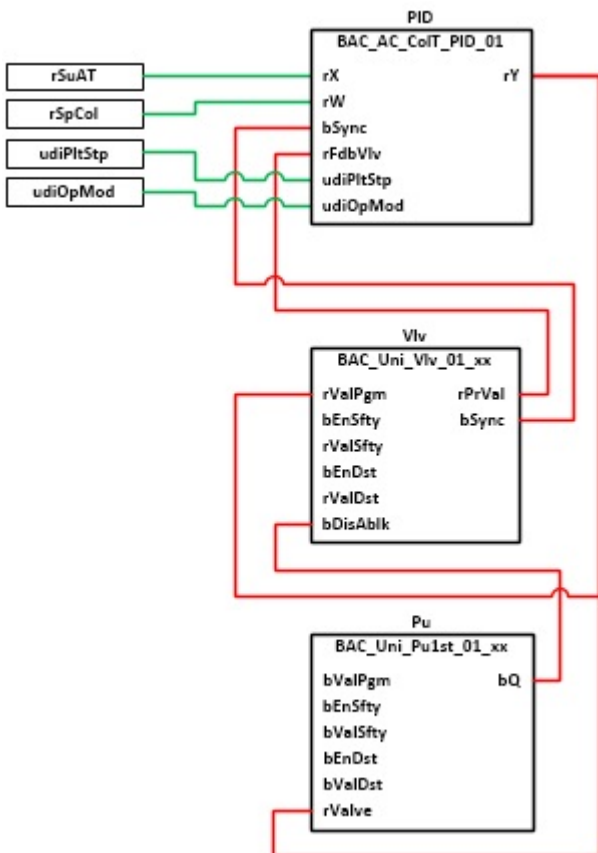
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```
rSuAT      : REAL;
rSpCol     : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
```

rSuAT: Messwert der Zulufttemperatur

rSpCol: Sollwert für die Zulufttemperatur

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC_AC StartT_01](#) [[▶ 538](#)].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC_AC OpMod_01](#) [[▶ 524](#)].

Programmbeschreibung

Instanz	Typ	Aufgabe
PID	BAC_AC_CoIT_PID_01 [▶ 471]	Sub-Template inklusive des PID-Sequenzreglers für die Zulufttemperaturregelung.
Vlv	BAC_Uni_Vlv_01_13 [▶ 608]	Sub-Template zur Ansteuerung eines stetigen Ventils
Pu	BAC_Uni_Pu1st_01_189 [▶ 579]	Sub-Template zur Ansteuerung einer einstufigen Pumpe

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

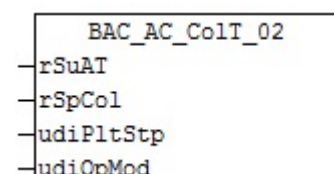
9.35 BAC_AC_CoIT_02**Anwendung**

Das Template **BAC_AC_CoIT_02** dient der Steuerung und Regelung eines Kaltwasser-Luftkühlers ohne Entfeuchterregelung.

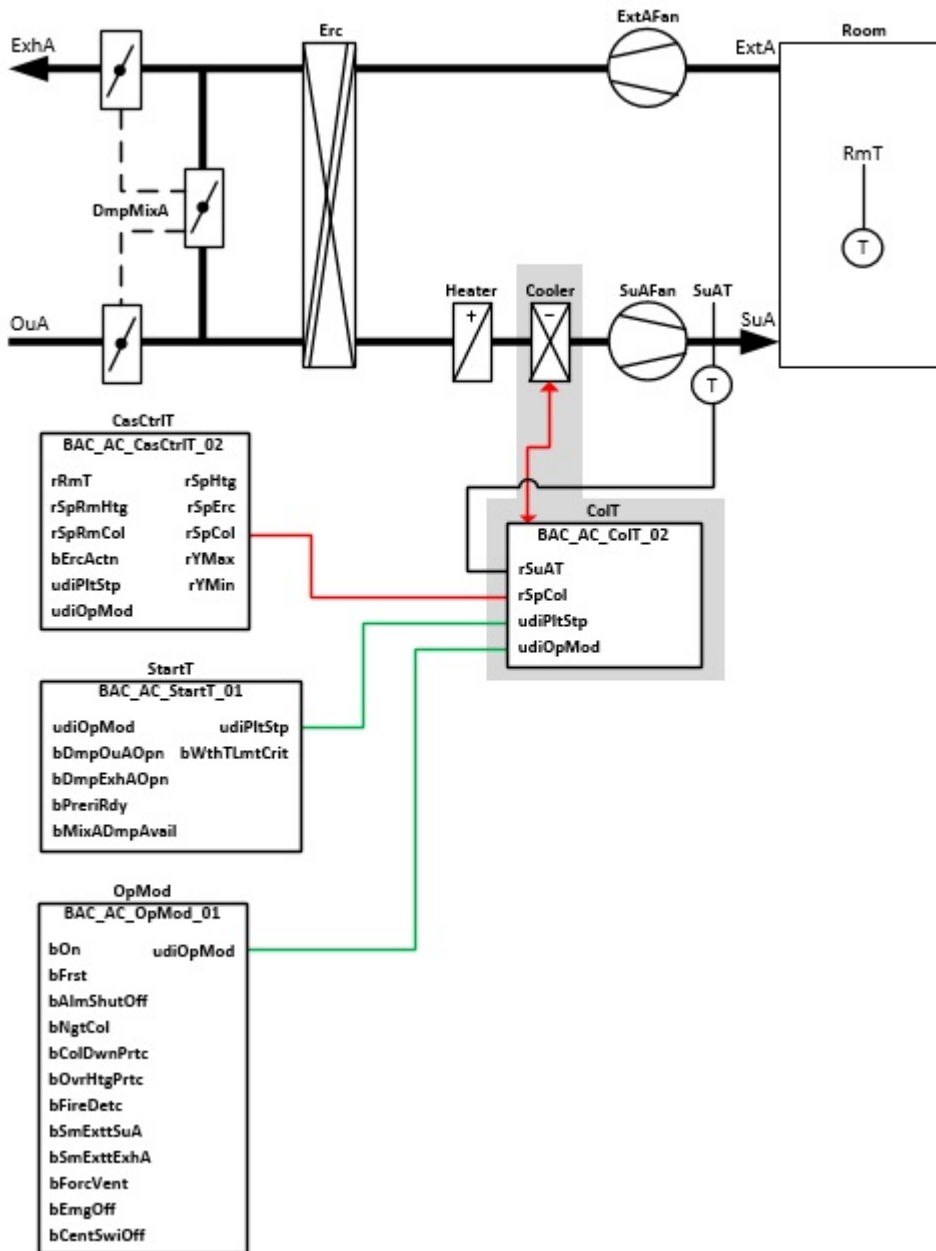
Das Template **BAC_AC_CoIT_02** ist ein Aufruftemplate.

Innerhalb des Aufruftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

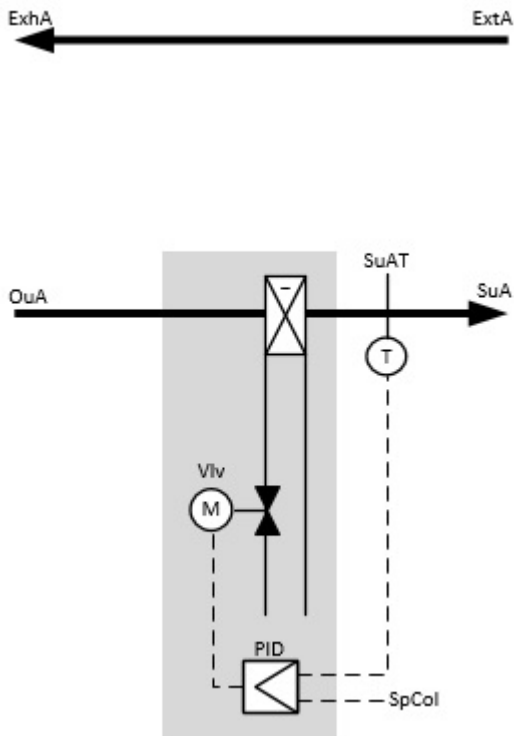
- **CoITPID:** Regelung der Zulufttemperatur
- **CoITVlv:** Ansteuerung eines stetigen Regelventils

Schnittstelle

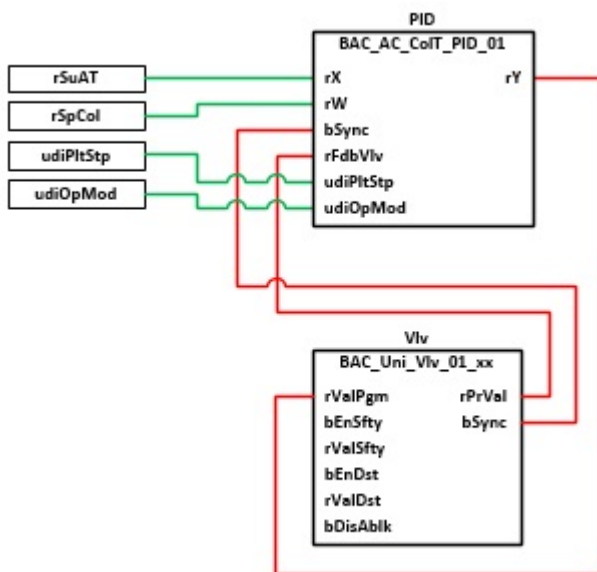
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

rSuAT      : REAL;
rSpCol     : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;

```

rSuAT: Messwert der Zulufttemperatur

rSpCol: Sollwert für die Zulufttemperatur

udiPltStp: Schrittte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC AC StartT 01](#) [► 538].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC AC OpMod 01](#) [► 524].

Programmbeschreibung

Instanz	Typ	Aufgabe
PID	BAC AC CoIT_PID_01 [▶ 471]	Sub-Template inclusive des PID-Sequenzreglers für die Zulufttemperaturregelung.
Vlv	BAC Uni Vlv_01_13 [▶ 608]	Sub-Template zur Ansteuerung eines stetigen Ventils

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.36 BAC_AC_CoIT_PID_01

Funktionsbeschreibung

Das Sub-Template **BAC_AC_CoIT_PID_01** ist der Sequenzregler für die Zulufttemperaturregelung eines Kühlers.

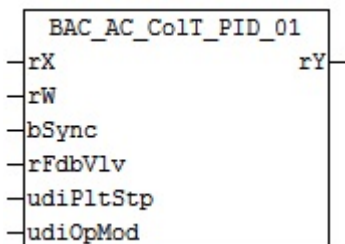
Die Referenzierung des Sollwertes, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte **X**, **W** und **Y**.

Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart **udiOpMod** und der globalen Temperatur-Kommunikationsstruktur **g_stSeqLinkT[PLT_NUM]**. Diese Daten- und Befehlsstruktur ist das Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein **FB_BA_SeqLink** [▶ 171] einer Anlage.

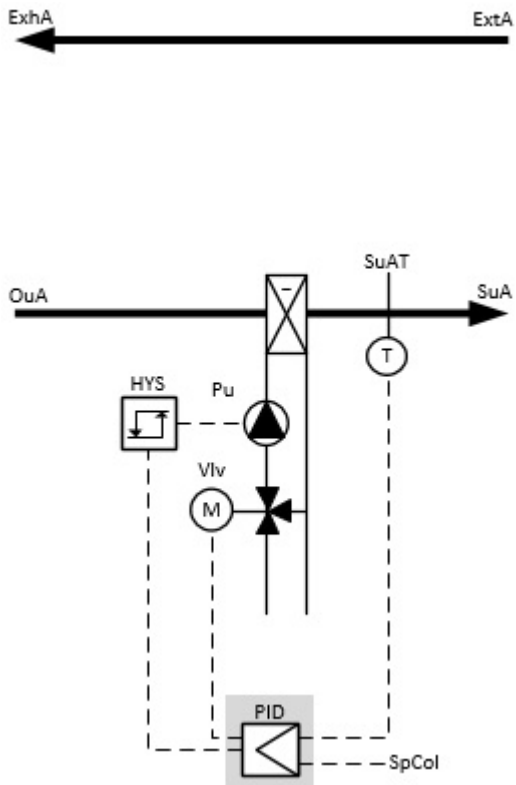
Zur Anzeige der Reglerfreigabe dient das BACnet-BV-Objekt **En**.

Die Grenzwertüberwachung des PID-Reglers wird in Abhängigkeit des Anlagenanfahrprozesses von dem Funktionsbaustein **EnEvtEn** gesteuert.

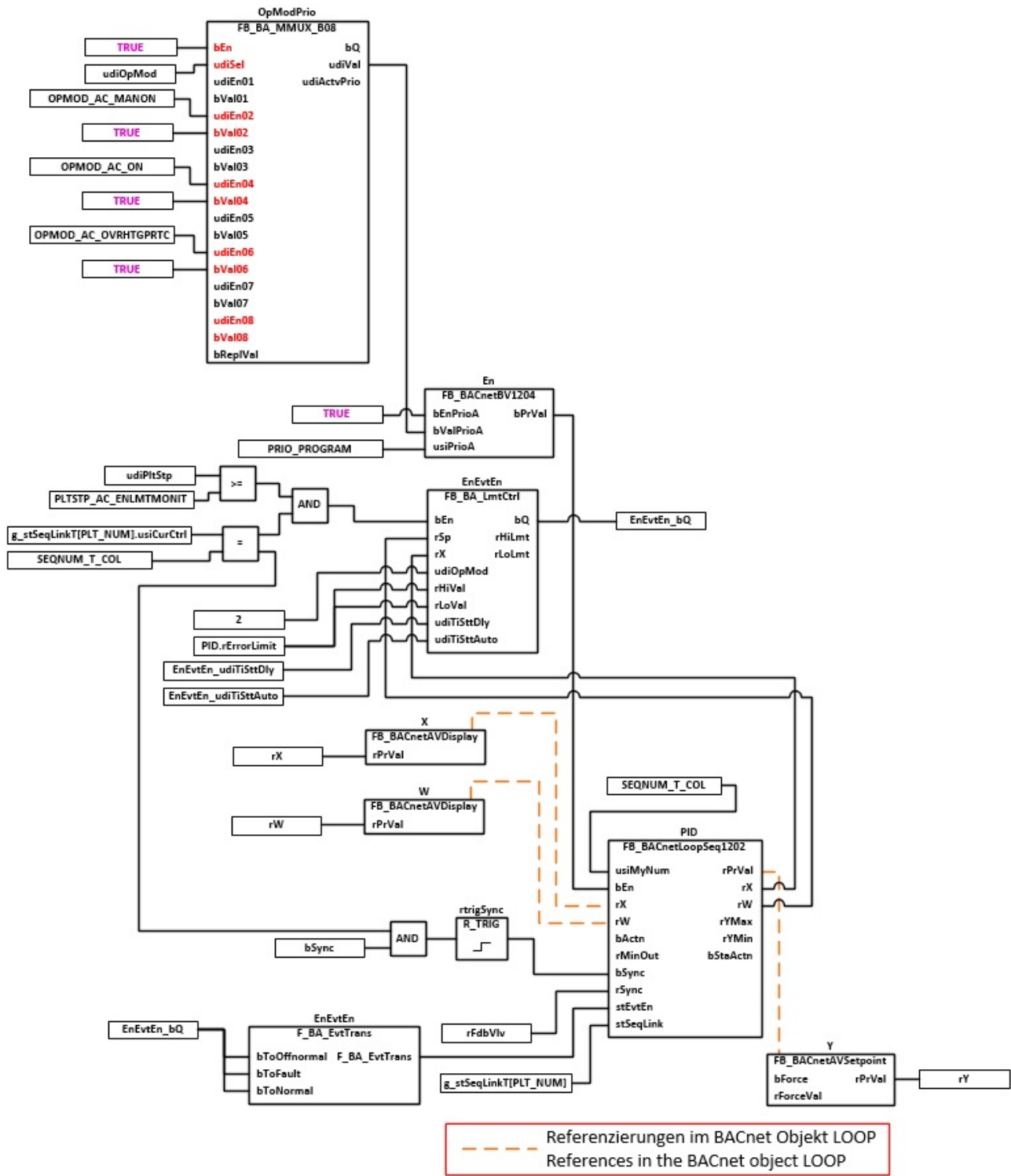
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

```

rX      : REAL;
rW      : REAL;
bSync   : BOOL;
rFdbVlv : REAL;
udiPltStp : UDINT;
udiOpMod : UDINT;
    
```

rX: Messwert Zulufttemperatur

rW: Sollwert der Zulufttemperatur

bSync: Eingang für die Synchronisation des Reglers

rFdbVlv: Stellungsrückmeldung Aktor

udiPltStp: Schritt beim Anfahren der RLT-Anlage. Siehe auch [BAC_AC_StartT_01 \[► 538\]](#)

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC_AC_OpMod_01 \[► 524\]](#)

VAR_OUTPUT

```
rY : REAL;
```

rY: Ausgabe der Stellgröße für das Regelventil

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm \[► 182\]](#)

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01 \[► 372\]](#) mittels des Funktionsbausteins [FB_BA_AlarmPlt \[► 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg \[► 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig, wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur `g_stSeqLinkT[PLT_NUM]` dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein [FB_BA_SeqLink](#) zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe												
X	FB_BACnetAVDisplay [► 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts												
W	FB_BACnetAVDisplay [► 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts												
OpModPri o	FB_BA_MMUX_B08 [► 208]	Der Multiplexer definiert die Freigabebedingungen des Sequenzreglers in Abhängigkeit der Anlagenbetriebsart.												
		<table border="1"> <thead> <tr> <th>udiOpMod</th> <th>Freigabe</th> <th>Bemerkung</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_MANO N [► 364]</td> <td>Hand ein</td> <td>Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet</td> </tr> <tr> <td>OPMOD_AC_ON [► 364]</td> <td>Ein</td> <td>Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm</td> </tr> <tr> <td>OPMOD_AC_OVRHT GPRTC [► 364]</td> <td>Überhitzungsschutz</td> <td>Die Anlage läuft in der Betriebsart Überhitzungsschutz</td> </tr> </tbody> </table>	udiOpMod	Freigabe	Bemerkung	OPMOD_AC_MANO N [► 364]	Hand ein	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet	OPMOD_AC_ON [► 364]	Ein	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm	OPMOD_AC_OVRHT GPRTC [► 364]	Überhitzungsschutz	Die Anlage läuft in der Betriebsart Überhitzungsschutz
udiOpMod	Freigabe	Bemerkung												
OPMOD_AC_MANO N [► 364]	Hand ein	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet												
OPMOD_AC_ON [► 364]	Ein	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm												
OPMOD_AC_OVRHT GPRTC [► 364]	Überhitzungsschutz	Die Anlage läuft in der Betriebsart Überhitzungsschutz												
En	FB_BACnetBV1204 [► 94]	Das BV-Objekt dient zur Anzeige und Aktivierung der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay. Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart udiOpMod und der globalen Kommunikationsstruktur g_stSeqLinkT .												

Instanz	Typ	Aufgabe
EnEvtEn	FB BA LmtCtrl [▶ 233]	<p>Das BACnet-Loop-Objekt PID überwacht die Funktion der Regelung in dem es den Sollwert W und den Istwert X miteinander vergleicht. Ist die Abweichung $W-X$ größer als das Property ErrorLimit, dann sendet das Loop-Objekt eine Meldung an die MBE.</p> <p>Beim Anlagenstillstand, im Moment des Starts und bis zum eingeregelter Zustand der Anlage wird das Melden des Loop-Objekts unterdrückt, damit keine falschen Meldungen an die MBE gesendet werden. Erst wenn die RLT-Anlage vollständig hochgefahren ist und die Regelung sich eingependelt hat, wird das Melden des Loop-Objekts freigegeben. Zusätzlich wird das Melden aktiviert, wenn die Regelung nach langer Zeit nicht in dem vom Property ErrorLimit definierten Bereich um den Sollwert gelangt ist.</p> <p>Die Freigabe des objektinternen Meldens erfolgt indem auf die BACnetEventTransitionBits des Loop-Objekts geschrieben wird.</p> <p>Für die Freigabe des Meldens vom Loop-Objekt müssen die folgenden Bedingungen erfüllt sein:</p> <ol style="list-style-type: none"> <p>Das Anlagenstartprogramm <u>BAC_AC_StartT_01 [▶ 538]</u> hat die Überwachung der Regelung und die Grenzwertüberwachung der Fühler freigegeben udiPltStp >= PLTSTP_AC_ENLMTMONIT und Der Kühlerregler ist der aktive Regler in der Regelsequenz <u>g_stSeqLinkT[PLT_NUM] [▶ 364].usiCurCtrl = SEQNUM_T_COL</u> und Die Zulufttemperatur hat sich dem Sollwert so weit angenähert, dass sie sich in einem Bereich zwischen rSp - ErrorLimit und rSp + ErrorLimit eingependelt hat. und Die Zulufttemperatur muss mindestens für die Dauer von EnEvtEn_udiTiSttDly innerhalb des Bereiches von rSp - ErrorLimit und rSp + ErrorLimit verblieben sein.</p> <p>Der Timer EnEvtEn_udiTiSttAuto ist abgelaufen und die Regelung hat Ihren Sollwertbereich nicht erreicht.</p>
	F_BA_EvtTrans	<p>schreibt auf die BACnetEventTransitionBits to_offnormal, to_fault und to_normal des Loop-Objekts. Dazu muss der Eingang bEnEvtEn = TRUE sein</p>
PID	FB BACnetLoopSeq1202 [▶ 112]	Sequenzregler Zulufttemperatur Kühler.
rtrigSync	R_TRIG	<p>Bei einer steigenden Flanke am Eingang bSync wird das Loop-Objekt auf den Wert von IrSync aufsynchronisiert.</p> <p>Wenn das Regelventil des Kühlers durch Schreiben einer höheren Priorität auf das zugehörige AO-Objekt von der MBE oder durch Betätigen der örtlichen Vorrangbedienung übersteuert wurde, weicht die aktuelle Position des Regelventils von dem Ausgang des Loop-Objekts ab. Mit den Variablen bSync und rFdbVlv kann die Synchronität zwischen der Position des Regelventils und dem Regler wieder hergestellt werden.</p>
Y	FB BACnetAVSetpoint [▶ 70]	Dass AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.37 BAC_AC_CoITH_01

Anwendung

Das Template **BAC_AC_CoITH_01** dient der Steuerung, Zulufttemperaturregelung und Zuluftfeuchteregeung eines Kaltwasser-Luftkühlers.

Das Template **BAC_AC_CoITH_01** ist ein Aufrufftemplate.

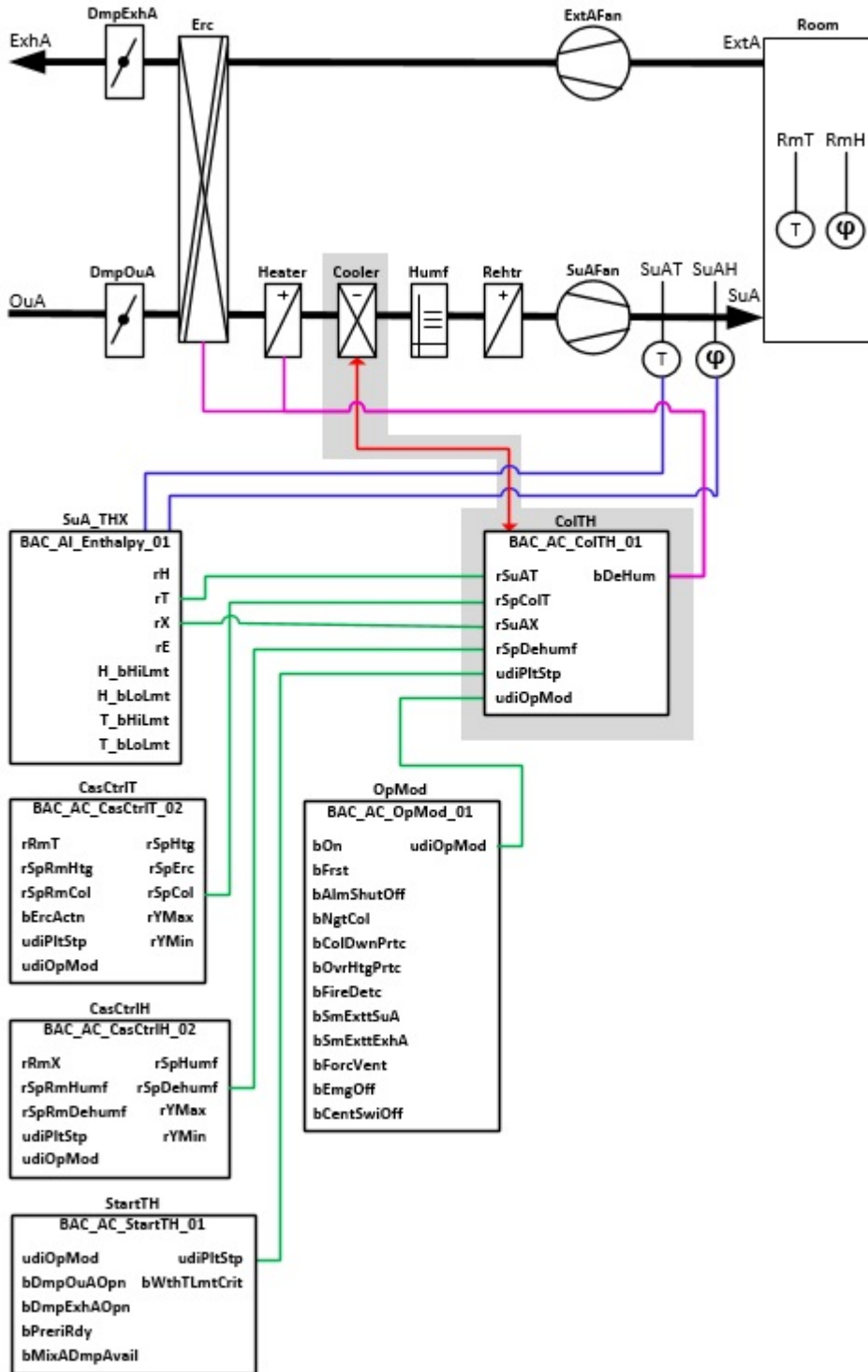
Innerhalb des Aufrufftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

- **CoITHPID_T**: Regelung der Zulufttemperatur
- **CoITHPID_H**: Regelung der absoluten Zuluftfeuchte
- **CoITHPu**: Ansteuerung der Kühlerpumpe
- **CoITHVlv**: Ansteuerung eines stetigen Regelventils

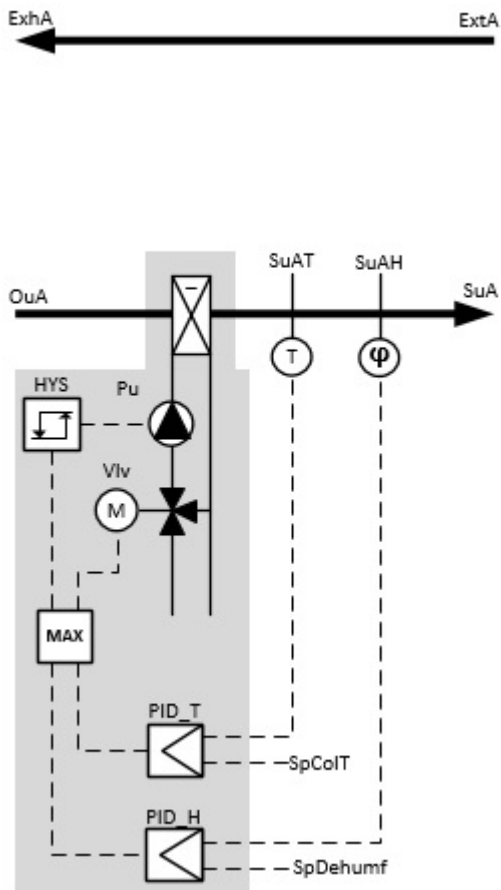
Schnittstelle



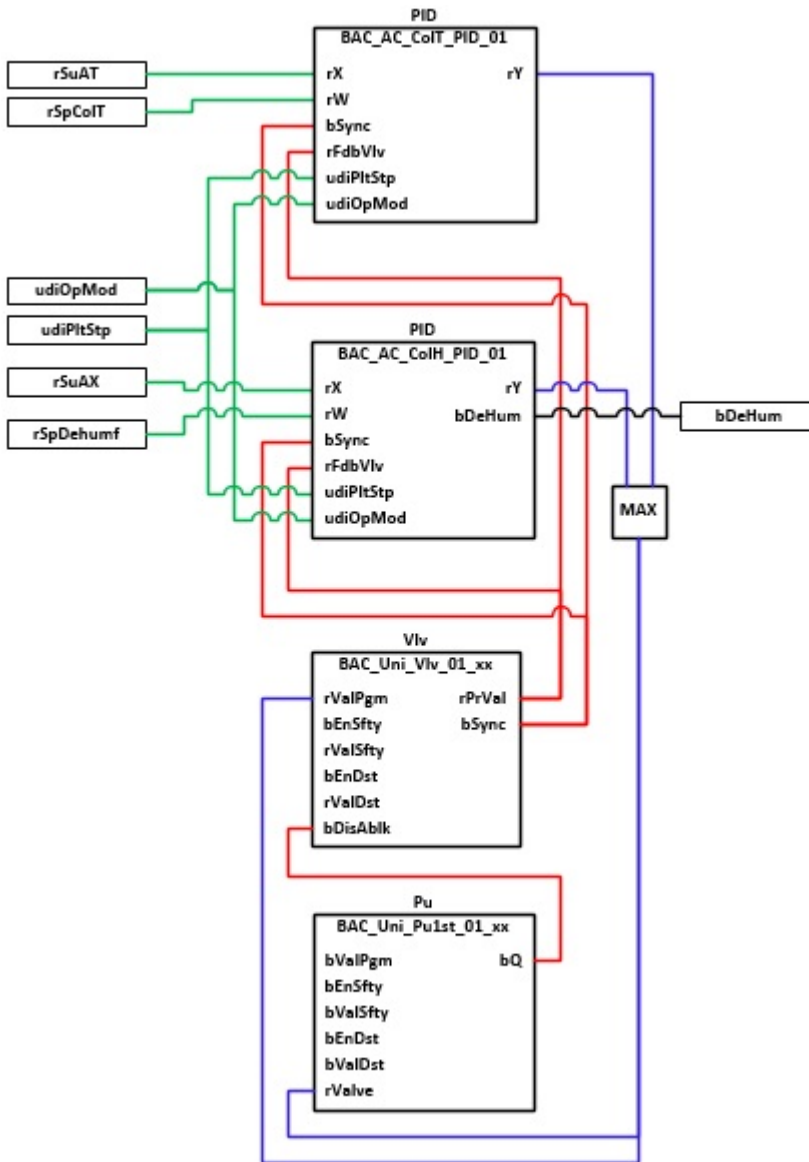
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```
rSuAT      : REAL;
rSpCoIT    : REAL;
rSuAX      : REAL;
rSpDehumf  : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
```

rSuAT: Messwert der Zulufttemperatur

rSpCoIT: Sollwert für die Zulufttemperatur

rSuAX: Errechneter Wert der absoluten Zuluftfeuchte. Dieser Wert entsteht im Template [BAC AI Enthalpy 01](#) [▶ 687].

rSpDehumf: Sollwert für die absolute Zuluftfeuchte.

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC AC StartTH 01](#) [▶ 543].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC AC OpMod 01](#) [▶ 524].

VAR_OUTPUT

```
bDeHum      : BOOL;
```

bDeHum: Entfeuchtebetrieb aktiv.

Bei aktivem Entfeuchtebetrieb wird die Energierückgewinnung im Heizbetrieb und der Vorerhitzer gesperrt.

Programmbeschreibung

Instanz	Typ	Aufgabe
CoITHPID_T	BAC_AC_CoIT_PID_01 [▶ 471]	Sub-Template inklusive des PID-Sequenzreglers für die Zulufttemperaturregelung.
CoITHPID_H	BAC_AC_CoIH_PID_01 [▶ 460]	Sub-Template inklusive des PID-Sequenzreglers für die absolute Zuluftfeuchteregeung.
Vlv	BAC_Uni_Vlv_01_13 [▶ 608]	Sub-Template zur Ansteuerung eines stetigen Ventils
Pu	BAC_Uni_Pu1st_01_189 [▶ 579]	Sub-Template zur Ansteuerung einer einstufigen Pumpe

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

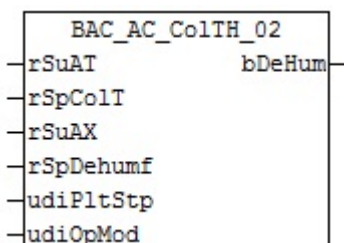
9.38 BAC_AC_CoITH_02**Anwendung**

Das Template **BAC_AC_CoITH_02** dient der Steuerung, Zulufttemperaturregelung und Zuluftfeuchteregeung eines Kaltwasser-Luftkühlers.

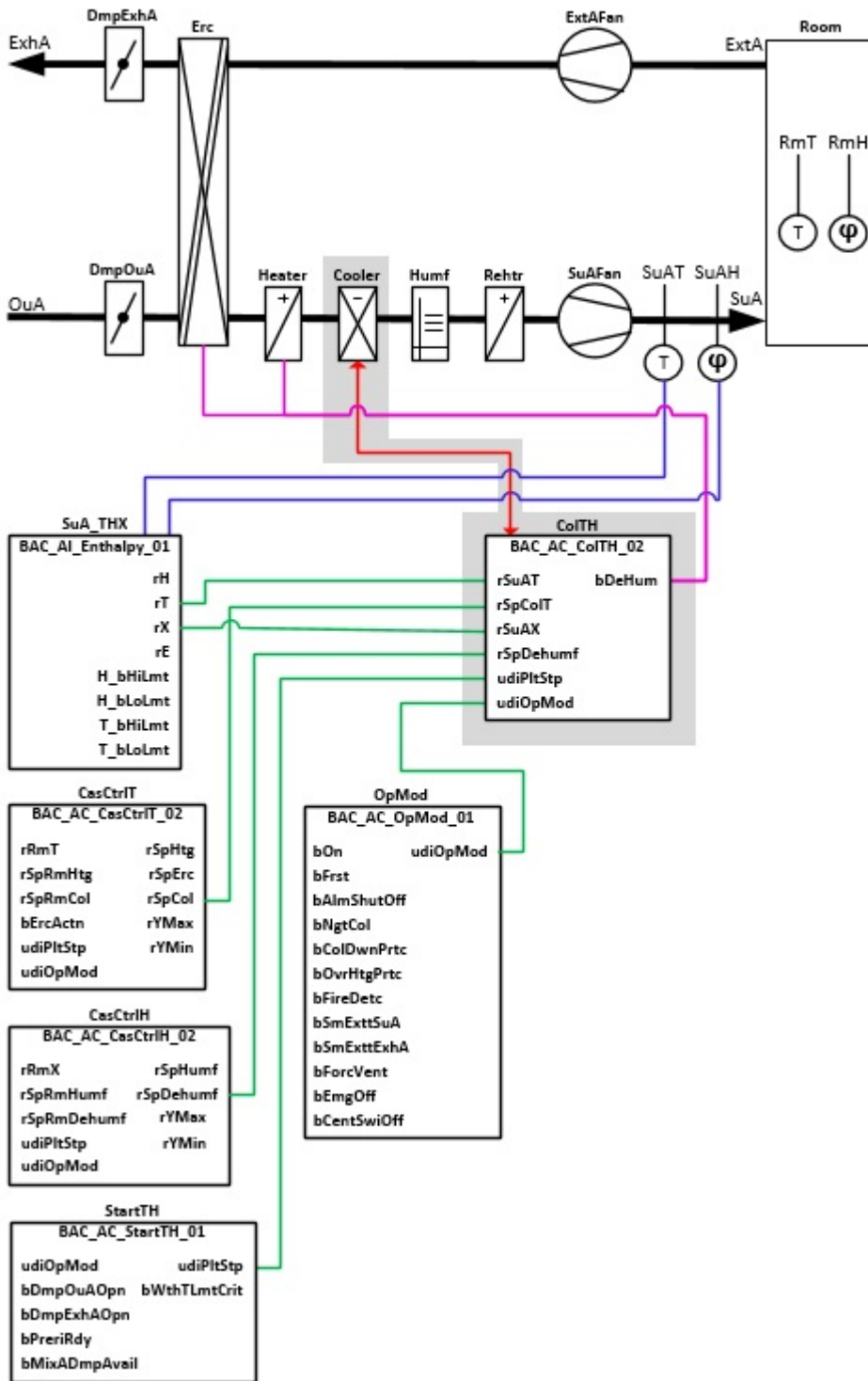
Das Template **BAC_AC_CoITH_02** ist ein Aufruftemplate.

Innerhalb des Aufruftemplates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft:

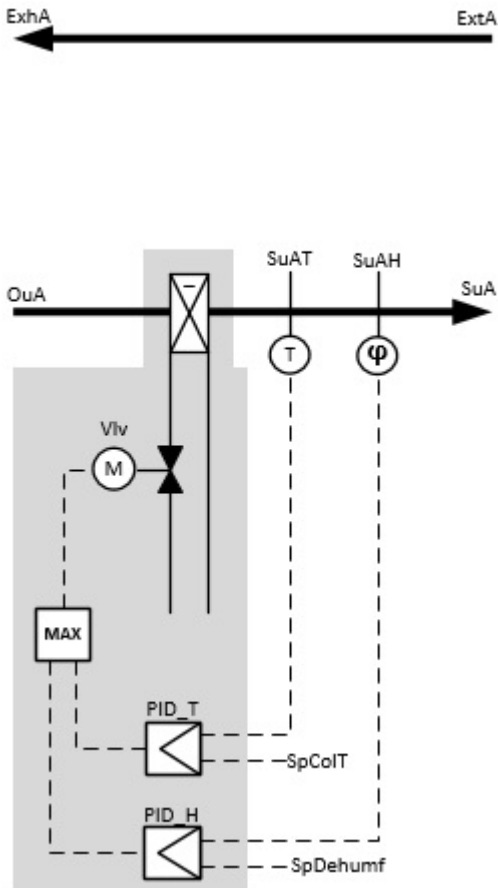
- **CoITHPID_T:** Regelung der Zulufttemperatur
- **CoITHPID_H:** Regelung der absoluten Zuluftfeuchte
- **CoITHVlv:** Ansteuerung eines stetigen Regelventils

Schnittstelle

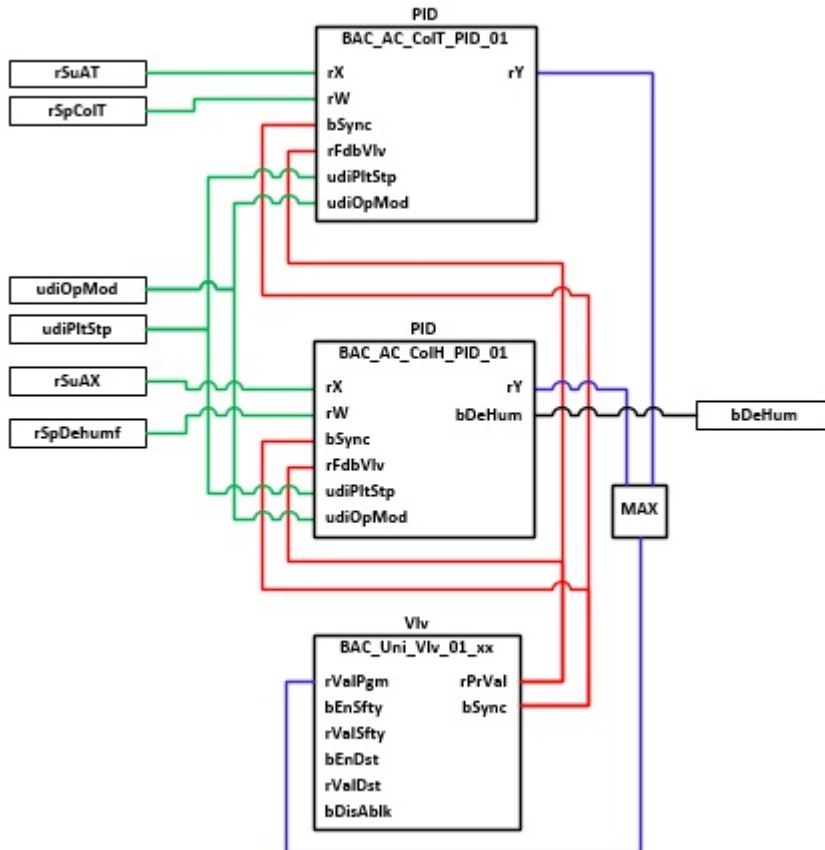
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```
rSuAT      : REAL;
rSpCoIT    : REAL;
rSuAX      : REAL;
rSpDehumf  : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
```

rSuAT: Messwert der Zulufttemperatur

rSpCoIT: Sollwert für die Zulufttemperatur

rSuAX: Errechneter Wert der absoluten Zuluftfeuchte. Dieser Wert entsteht im Template [BAC_AI_Enthalpy_01](#) [[▶ 687](#)].

rSpDehumf: Sollwert für die absolute Zuluftfeuchte.

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC_AC_StartTH_01](#) [[▶ 543](#)].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC_AC_OpMod_01](#) [[▶ 524](#)].

VAR_OUTPUT

```
bDeHum     : BOOL;
```

bDeHum: Entfeuchtebetrieb aktiv.

Bei aktiven Entfeuchtebetrieb wird die Energierückgewinnung im Heizbetrieb und der Vorerhitzer gesperrt.

Programmbeschreibung

Instanz	Typ	Aufgabe
CoITHPID_T	BAC_AC_CoIT_PID_01 [▶ 471]	Sub-Template inklusive des PID-Sequenzreglers für die Zulufttemperaturregelung.
CoITHPID_H	BAC_AC_CoIH_PID_01 [▶ 460]	Sub-Template inklusive des PID-Sequenzreglers für die absolute Zuluftfeuchteregeung.
Vlv	BAC_Uni_Vlv_01_13 [▶ 608]	Sub-Template zur Ansteuerung eines stetigen Ventils

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.39 BAC_AC_MixAT_01

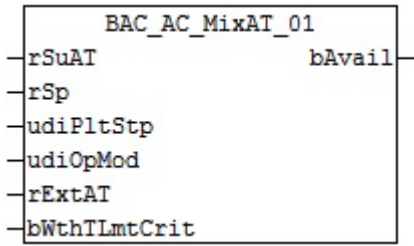
Anwendung

Das Aufruftemplate **BAC_AC_MixAT_01** dient zur Steuerung und Regelung von einem Mischluftsystem. Das Template **BAC_AC_MixAT_01** ist ein Aufruftemplate.

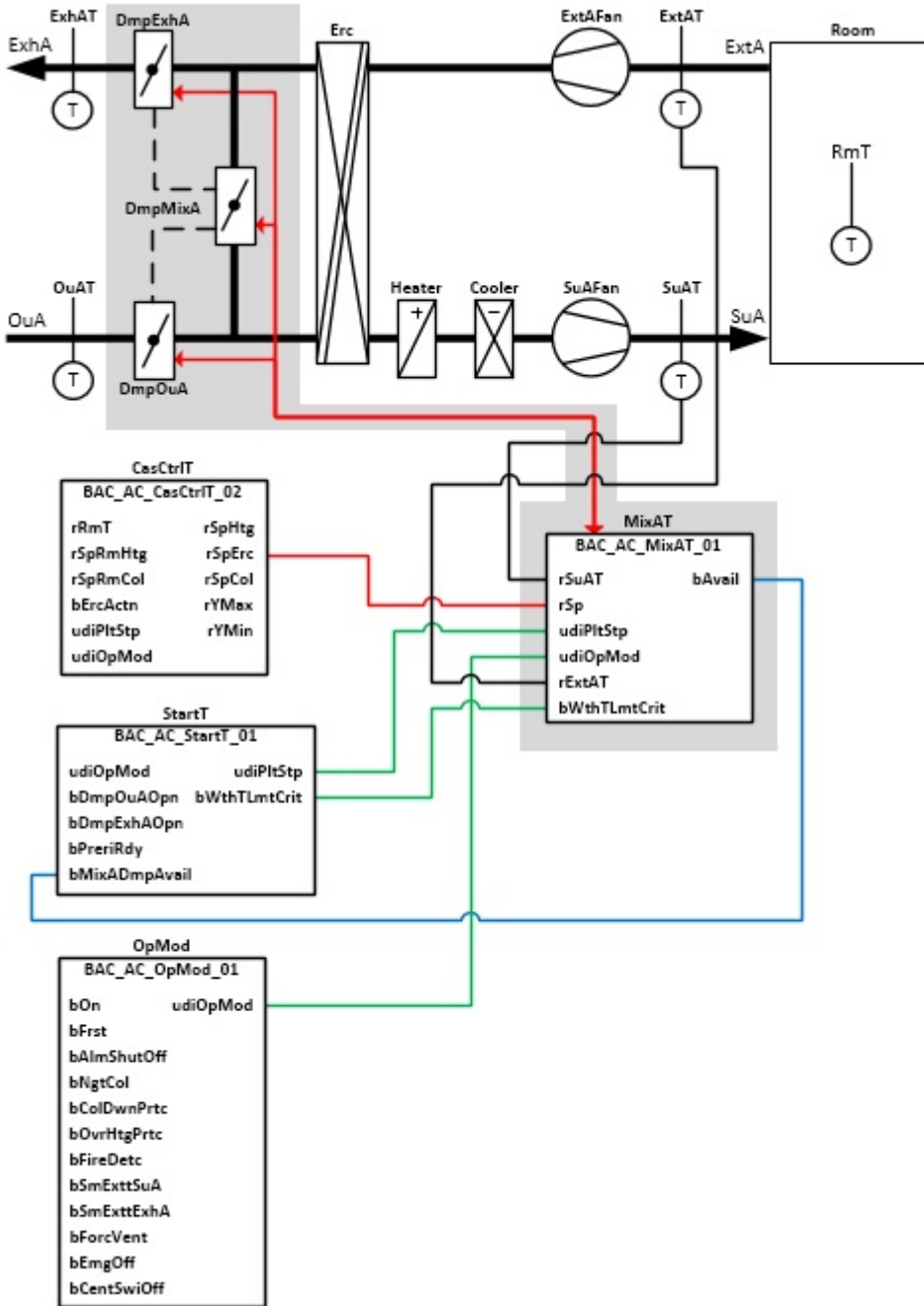
Innerhalb des Templates werden die folgenden Sub-Templates aufgerufen und miteinander verknüpft.

- **MixAT_PID:** Zulufttemperaturregler
- **DmpMixA:** Mischluftklappe
- **DmpOuA:** Außenluftklappe
- **DmpExhA:** Fortluftklappe

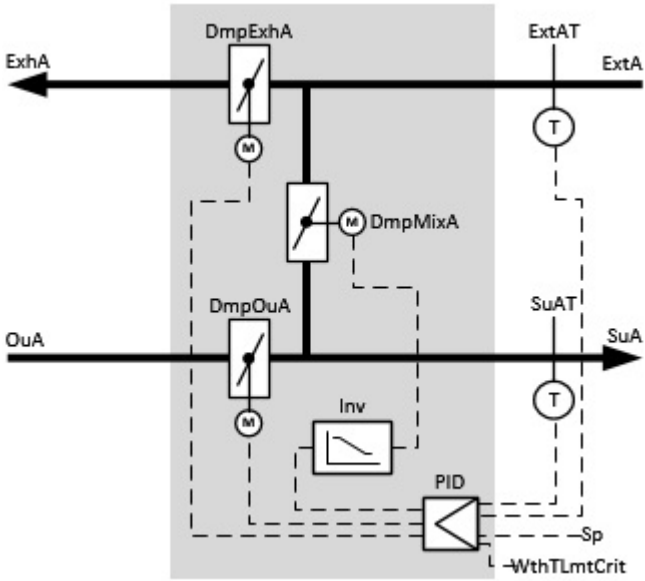
Schnittstelle



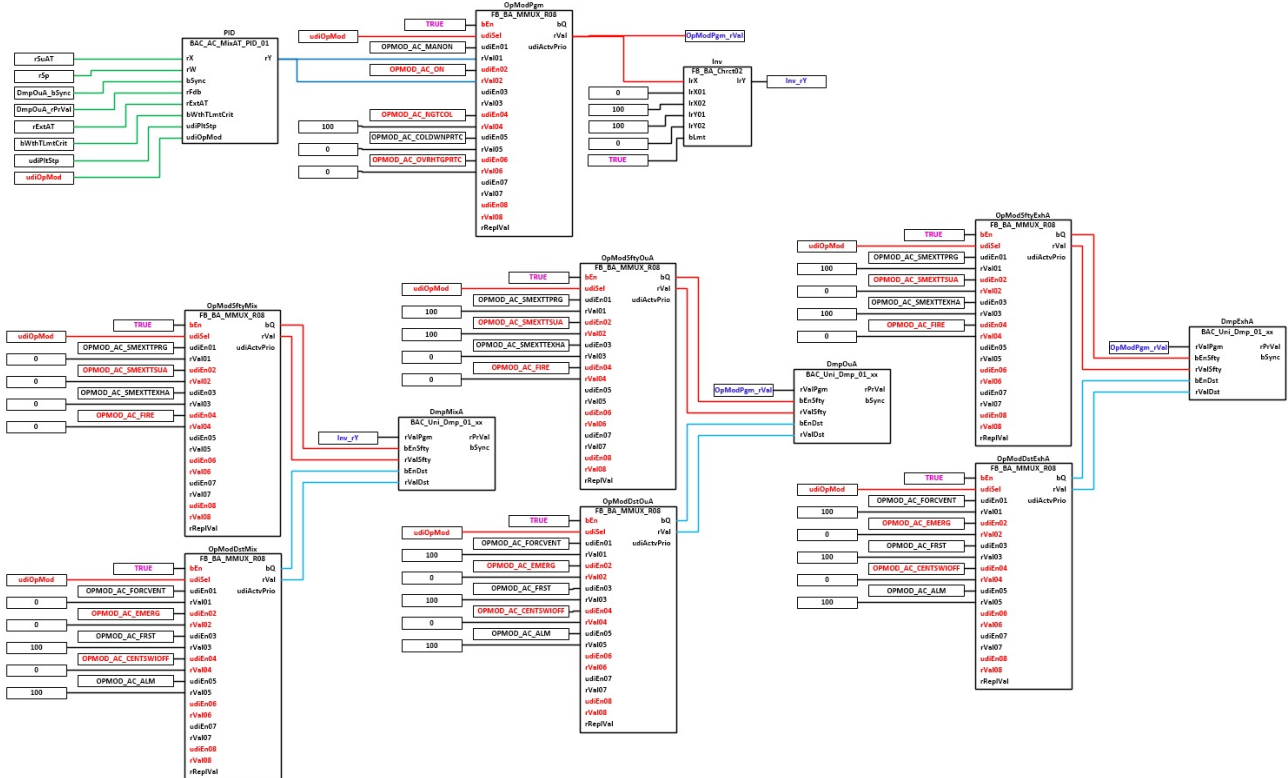
Anlagenschema 01



Anlagenschema 02



Blockdiagramm



VAR_INPUT

```

rSuAT      : REAL;
rSp        : REAL;
udiPltStp : UDINT;
udiOpMod   : UDINT;
rExtAT     : REAL;
bWthTLmtCrit : BOOL;
    
```

rSuAT: Messwert der Zulufttemperatur

rSp: Sollwert für die Zulufttemperatur

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm **BAC AC StartT_01** [▶ 538].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC AC OpMod 01](#) [[▶ 524](#)].

rExtAT: Messwert Ablufttemperatur.

bWthTLmtCrit: Signal vom Anlagenstartprogramm. Starten im Umluftbetrieb mittels Anfahrrampe.

VAR_OUTPUT

bAvail : BOOL;

bAvail: Dieser Ausgang zeigt an, dass ein Mischluftklappensystem vorhanden ist. Dem Anlagenstartprogramm wird durch diese Variable mitgeteilt, dass das Vorspülen des Vorerhitzers übersprungen wird, siehe [BAC AC StartT 01](#) [[▶ 538](#)].

Die Rampenfunktion in dem Template **MixAT_PID** ersetzt beim Starten der Lüftungsanlage das Vorspülen des Luftherhitzers.

Programmbeschreibung

Instanz	Typ	Aufgabe			
PID	BAC AC MixAT PID 01 [▶ 488]	Sub-Template inklusive des PID-Sequenzreglers für die Zulufttemperaturregelung.			
DmpMixA	BAC Uni Dmp 01 05 [▶ 568]	Sub-Template Umluftklappe			
DmpOuA	BAC Uni Dmp 01 05 [▶ 568]	Sub-Template Außenluftklappe			
DmpExhA	BAC Uni Dmp 01 05 [▶ 568]	Sub-Template Fortluftklappe			
Inv	FB BA Chrct02 [▶ 173]	Invertierung des Stellsignals der Außenluftklappe für Ansteuerung der Mischluftklappe.			
OpModPgm	FB BA MMUX R04 [▶ 208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Stellgröße für die Klappen in der Programm-Priorität (Automatikbetrieb)			
		udiOpMod	Wert		
		OPMOD_AC_M ANON	Hand Ein	Pgm_rY	Pgm_rY ist eine Variable die das Ergebnis der MIN-Auswahl des Mischluftklappenreglers und der Rampenfunktion an die Klappen weiter gibt.
		OPMOD_AC_A C_ON	Automatikbetrieb Ein	Pgm_rY	Pgm_rY ist eine Variable die das Ergebnis der MIN-Auswahl des Mischluftklappenreglers und der Rampenfunktion an die Klappen weiter gibt.
		OPMOD_AC_N GTCOL	Nachtauskühlung	100	
		OPMOD_AC_C OLDWNPRTC	Auskühlschutz	0	
OPMOD_AC_O VRHTGPRTC	Überhitzungsschutz	0			
OpModSftyMix	FB BA MMUX R04 [▶ 208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Sicherheitspriorität für die Umluftklappe			
		udiOpMod	Wert		

Instanz	Typ	Aufgabe		
		OPMOD_AC_SMEXT TPRG	Entrauchung Programm	0
		OPMOD_AC_SMEXT TSUA	Entrauchung Zuluft	0
		OPMOD_AC_SMEXT TEXHA	Entrauchung Fortluft	0
		OPMOD_AC_FIRE	Feuer	0
OpModDistMi x	FB_BA_MMUX_R04 [▶_208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Priorität Störung für die Umluftklappe		
		udiOpMod		Wert
		OPMOD_AC_FORCV ENT	Zwangsbelüftung	0
		OPMOD_AC_EMER G	Notfall	0
		OPMOD_AC_FRST	Frost	100
		OPMOD_AC_CENTS WIOFF	Zentralabschaltung	0
		OPMOD_AC_ALM	Störung/Alarm	100
OpModSftyOu A	FB_BA_MMUX_R04 [▶_208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Sicherheitspriorität für die Außenluftklappe		
		udiOpMod		Wert
		OPMOD_AC_SMEXT TPRG	Entrauchung Programm	100
		OPMOD_AC_SMEXT TSUA	Entrauchung Zuluft	100
		OPMOD_AC_SMEXT TEXHA	Entrauchung Fortluft	0
		OPMOD_AC_FIRE	Feuer	0
OpModDistOu A	FB_BA_MMUX_R04 [▶_208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Priorität Störung für die Außenluftklappe		
		udiOpMod		Wert
		OPMOD_AC_FORCV ENT	Zwangsbelüftung	100
		OPMOD_AC_EMER G	Notfall	0
		OPMOD_AC_FRST	Frost	0
		OPMOD_AC_CENTS WIOFF	Zentralabschaltung	0
OpModSftyEx hA	FB_BA_MMUX_R04 [▶_208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Sicherheitspriorität für die Fortluftklappe		
		udiOpMod		Wert
		OPMOD_AC_SMEXTTPRG	Entrauchung Programm	100
		OPMOD_AC_SMEXTTSUA	Entrauchung Zuluft	0
		OPMOD_AC_SMEXTTEXH A	Entrauchung Fortluft	100
OpModDistEx hA	FB_BA_MMUX_R04 [▶_208]	Der Multiplexer definiert anhand der Anlagenbetriebsart <i>udiOpMod</i> die Priorität Störung für die Fortluftklappe		
		udiOpMod		Wert
		OPMOD_AC_FORCV ENT	Zwangsbelüftung	100

Instanz	Typ	Aufgabe		
		OPMOD_AC_EMER G	Notfall	0
		OPMOD_AC_FRST	Frost	0
		OPMOD_AC_CENTS WIOFF	Zentralabschaltung	0
		OPMOD_AC_ALM	Störung/Alarm	0

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.40 BAC_AC_MixAT_PID_01

Funktionsbeschreibung

Das Sub-Template **BAC_AC_MixAT_PID_01** ist der Zulufttemperatur-Sequenzregler für ein Mischluftsystem.

Die Referenzierung des Sollwerters, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte **X**, **W** und **Y**.

Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart **udiOpMod** und der globalen Temperatur-Kommunikationsstruktur **g_stSeqLinkT[PLT_NUM]**. Diese Daten- und Befehlsstruktur ist das Bindeglied zwischen den einzelnen Sequenzreglern und

dem dazu gehörigen Steuerbaustein **FB_BA_SeqLink** [► 171] einer Anlage.

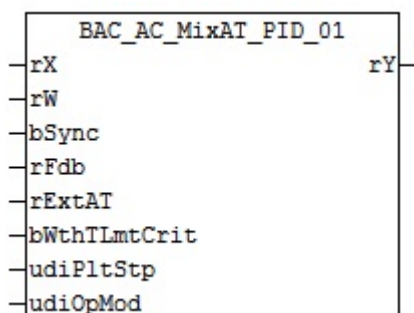
Zur Anzeige der Reglerfreigabe dient das BACnet-BV-Objekt **En**.

Die Grenzwertüberwachung des PID-Reglers wird in Abhängigkeit des Anlagenanfahrprozesses von dem Funktionsbaustein **EnEvtEn** gesteuert.

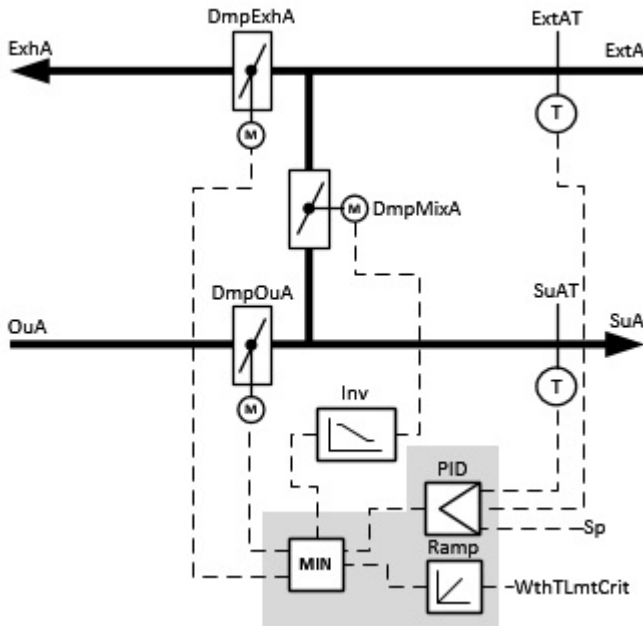
Die wesentlichen Aufgaben des Templates sind:

- Regelung der Zulufttemperatur
- Anfahren der Anlage durch langsames Öffnen der Außen- und Fortluftklappe.

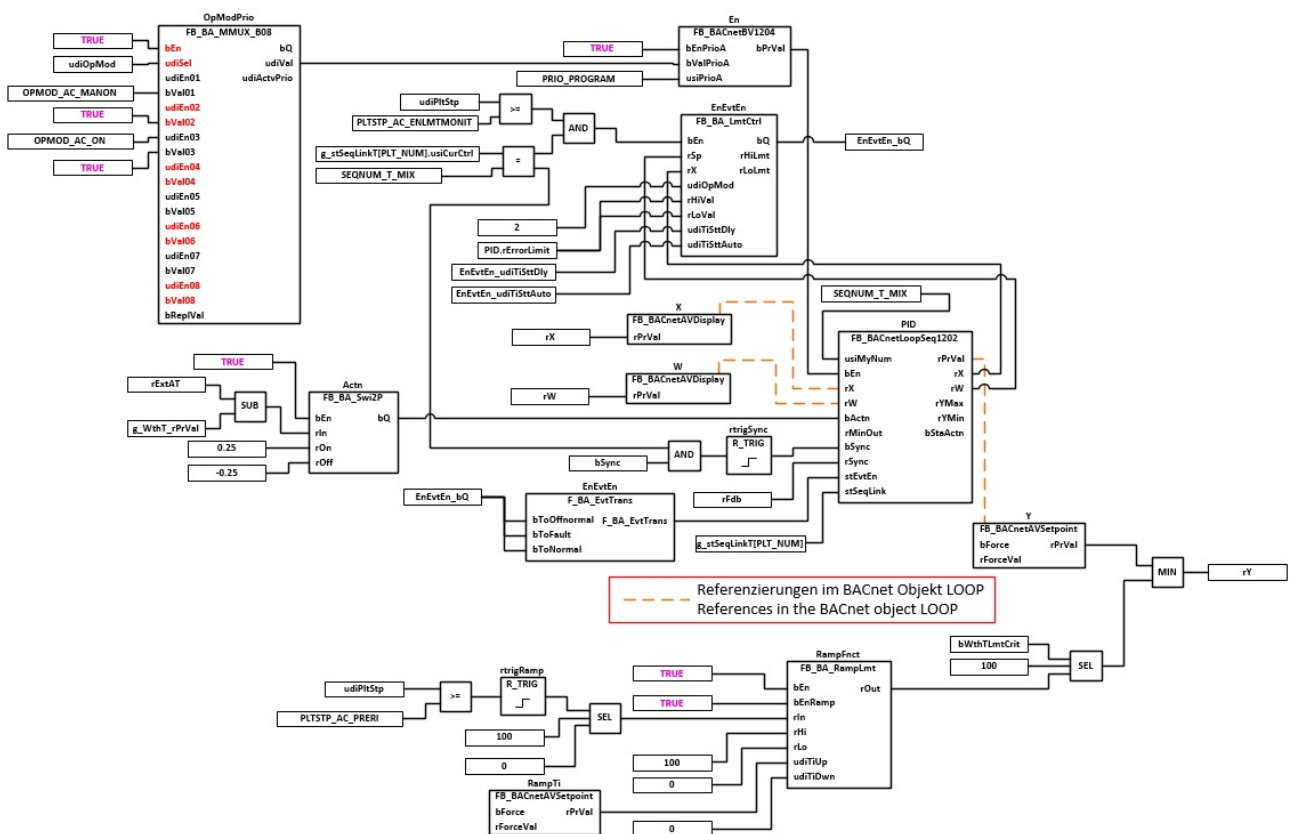
Schnittstelle



Anlagenschema



Blockschaltbild



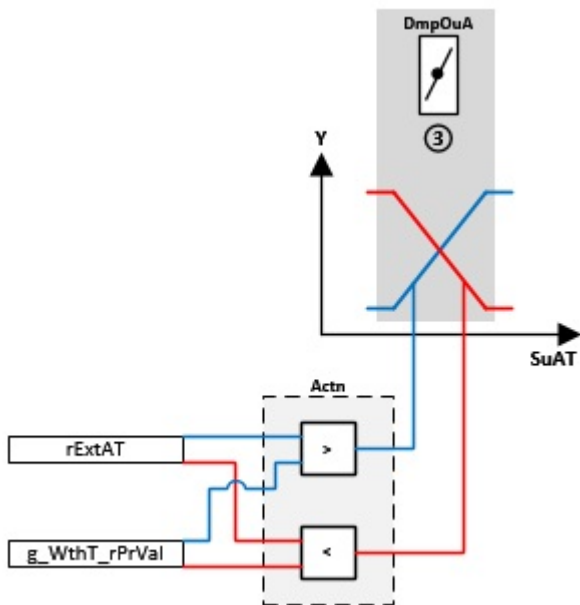
Wirksinn

Die Auswahl des Wirksinns des PID-Sequenzreglers erfolgt anhand eines Vergleichs von der Ablufttemperatur mit der Außentemperatur.

Die Logik des Wirksinns ist auf die **Aussenluftklappe DmpOuA** bezogen!

Ist die Ablufttemperatur größer als die Außentemperatur, dann ist der Wirksinn des PID-Reglers indirekt (Kühlbetrieb).

Ist die Ablufttemperatur kleiner als die Außentemperatur, dann ist der Wirksinn des PID-Reglers direkt (Heizbetrieb).



Information zur Regelung

Ein minimaler Aussenluftanteil muss am Property MinimumOutput des BACnet-Loop-Objekts eingestellt werden!

VAR_INPUT

```
rX      : REAL;
rW      : REAL;
bSync   : BOOL;
rFdb    : REAL;
rExtAT  : REAL;
udiPltStp : UDINT;
udiOpMod : UDINT;
```

rX: Messwert Zulufttemperatur

rW: Sollwert der Zulufttemperatur

bSync: Eingang für die Synchronisation des Reglers

rFdb: Stellungsrückmeldung Außenluftklappe

rExtAT: Messwert Ablufttemperatur

udiPltStp: Schritt beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartT_01 \[► 538\]](#)

udiOpMode: Anlagenbetriebsart. Siehe auch [BAC AC OpMod_01 \[► 524\]](#)

VAR_OUTPUT

```
rY      : REAL;
```

rY: Ausgabe der Stellgröße

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm. \[► 182\]](#)

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01 \[► 372\]](#)

mittels des Funktionsbausteins [FB_BA_AlarmPlt. \[▸ 186\]](#)

Die Auswertung verschiedener Anlageneignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB_BA_ComnMsg \[▸ 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur **g_stSeqLinkT[PLT_NUM]** dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein **FB_BA_SeqLink** zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe												
X	FB_BACnetAVDisplay [▸ 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts												
W	FB_BACnetAVDisplay [▸ 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts												
RampTi	FB_BACnetAVSetpoint [▸ 70]	Eingabe der Rampenzeit für eine Rampenfunktion, die während der Startphase der Lüftungsanlage die Aussen- und Fortluftklappe sehr langsam öffnet und die Umluftklappe schliesst.												
OpModPri o	FB_BA_MMUX_B08 [▸ 208]	Der Multiplexer definiert die Freigabebedingungen des Sequenzreglers in Abhängigkeit der Anlagenbetriebsart.												
		<table border="1"> <thead> <tr> <th colspan="2">udiOpMod</th> <th>Freigabe</th> <th>Bemerkung</th> </tr> </thead> <tbody> <tr> <td>OPMOD AC MANON [▸ 364]</td> <td>Hand ein</td> <td>TRUE</td> <td>Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet</td> </tr> <tr> <td>OPMOD AC ON [▸ 364]</td> <td>Ein</td> <td>TRUE</td> <td>Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm</td> </tr> </tbody> </table>	udiOpMod		Freigabe	Bemerkung	OPMOD AC MANON [▸ 364]	Hand ein	TRUE	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet	OPMOD AC ON [▸ 364]	Ein	TRUE	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm
		udiOpMod		Freigabe	Bemerkung									
OPMOD AC MANON [▸ 364]	Hand ein	TRUE	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet											
OPMOD AC ON [▸ 364]	Ein	TRUE	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm											
En	FB_BACnetBV1204 [▸ 94]	Mit dem BV-Objekt wird die Freigabe des Sequenzreglers in der MBE angezeigt.												
Actn	FB_BA_Swi2P [▸ 148] SUB	<p>Der Funktionsbaustein FB_BA_Swi2P [▸ 148] bestimmt den Wirksinn des Sequenzreglers.</p> <p>Ist die Subtraktion von der Ablufttemperatur und der Außentemperatur < - 0,25, dann ist der Ausgang des Funktionsbausteins Actn FALSE.</p> <p>Der Wirksinn des Loop-Objekts ist damit indirekt und die Mischluftklappensteuerung befindet sich im Heizbetrieb.</p> <p>Ist die Subtraktion von der Ablufttemperatur und der Außentemperatur > + 0,25, dann ist der Ausgang des Funktionsbausteins Actn TRUE.</p> <p>Der Wirksinn des Loop-Objekts ist damit direkt und die Mischluftklappensteuerung befindet sich im Kühlbetrieb.</p> <p>Die Hystere zwischen - 0.25 und + 0,25 des Funktionsbausteins Actn verhindert ein häufiges Umschalten des Reglerwirksinns bei annähernd gleicher Außen- und Ablufttemperatur.</p>												
EnEvtEn	FB_BA_LmtCtrl [▸ 233]	<p>Das BACnet-Loop-Objekt PID überwacht die Funktion der Regelung in dem es den Sollwert W und den Istwert X miteinander vergleicht. Ist die Abweichung W-X größer als das Property <i>ErrorLimit</i>, dann sendet das Loop-Objekt eine Meldung an die MBE.</p> <p>Beim Anlagenstillstand, im Moment des Starts und bis zum</p>												

Instanz	Typ	Aufgabe
		<p>eingeregelt Zustand der Anlage wird das Melden des Loop-Objekts unterdrückt, damit keine falschen Meldungen an die MBE gesendet werden. Erst wenn die RLT-Anlage vollständig hochgefahren ist und die Regelung sich eingependelt hat, wird das Melden des Loop-Objekts freigegeben. Zusätzlich wird das Melden aktiviert, wenn die Regelung nach langer Zeit nicht in dem vom Property <i>ErrorLimit</i> definierten Bereich um den Sollwert gelangt ist.</p> <p>Die Freigabe des objektinternen Meldens erfolgt indem auf die BACnetEventTransitionBits des Loop-Objekts geschrieben wird.</p> <p>Für die Freigabe des Meldens vom Loop-Objekt müssen die folgenden Bedingungen erfüllt sein:</p> <p>1. Das Anlagenstartprogramm <u>BAC AC StartT 01</u> [▶ 538] hat die Überwachung der Regelung und die Grenzwertüberwachung der Fühler freigegeben <i>udiPltStp</i> >= <u>PLTSTP AC ENLMTMONIT</u> [▶ 364] und Der Mischluftklappenregler ist der aktive Regler in der Regelsequenz. <u>g_stSeqLinkT[PLT_NUM]</u> [▶ 364].usiCurCtrl = SEQNUM_T_MIX und Die Zulufttemperatur hat sich dem Sollwert so weit angenähert, dass sie sich in einem Bereich zwischen rSp - ErrorLimit und rSp + ErrorLimit eingependelt hat. und Die Zulufttemperatur muss mindestens für die Dauer von <i>EnEvtEn_udiTiSttDly</i> innerhalb des Bereiches von rSp - ErrorLimit und rSp + ErrorLimit verblieben sein.</p> <p>2. Der Timer EnEvtEn_udiTiSttAuto ist abgelaufen und die Regelung hat Ihren Sollwertbereich nicht erreicht.</p>
rtrigSync	R_TRIG	<p>Bei einer steigenden Flanke am Eingang <i>bSync</i> wird das Loop-Objekt auf den Wert von <i>IrSync</i> aufsynchronisiert. Wenn die Außenluftklappe der Mischluftklappenregelung durch Schreiben einer höheren Priorität auf das zugehörige AO-Objekt von der MBE oder durch Betätigen der örtlichen Vorrangbedienung übersteuert wurde, weicht die aktuelle Position der Klappe von dem Ausgang des Loop-Objekts ab. Mit den Variablen <i>bSync</i> und <i>rFdb</i> kann die Synchronität zwischen der Position der Außenluftklappe und dem Regler wieder hergestellt werden.</p>
	F_BA_EvtTrans	<p>schreibt auf die BACnetEventTransitionBits <i>to_offnormal</i>, <i>to_fault</i> und <i>to_normal</i> des Loop-Objekts. Dazu muss der Eingang <i>bEnEvtEn</i> = TRUE sein</p>
PID	<u>FB BACnetLoopSeq1201</u> [▶ 107]	Sequenzregler Zulufttemperatur Mischluftklappenregelung
Y	<u>FB BACnetAVSetpoint</u> [▶ 70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts
rtrigRamp	R_TRIG >= SEL	Während der Startphase der Lüftungsanlage aktiviert dieses Netzwerk von Funktionen die Rampenfunktion.

Instanz	Typ	Aufgabe
RampFunct	FB BA RampLmt [► 143]	Mittels der Rampenfunktion werden die Außen- und Fortluftklappe beim Anfahren der Anlage sehr langsam geöffnet. Gestartet wird die Rampenfunktion nur bei kritischen Außentemperaturen unter 6°C . Diese Funktion ersetzt das Vorspülen des Warmwasserluftherhitzer.
	SEL, MIN	Das Ergebnis dieses Netzwerkes ist die Stellgröße für die Mischluftklappen.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.41 BAC_AC_ReHtr_01

Anwendung

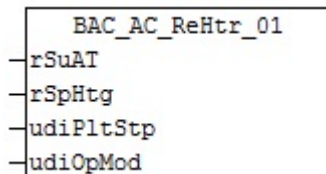
Das Aufruftemplate **BAC_AC_ReHtr_01** dient zur Steuerung und Regelung eines Nacherhitzers.

Die wesentlichen Aufgaben des Templates sind:

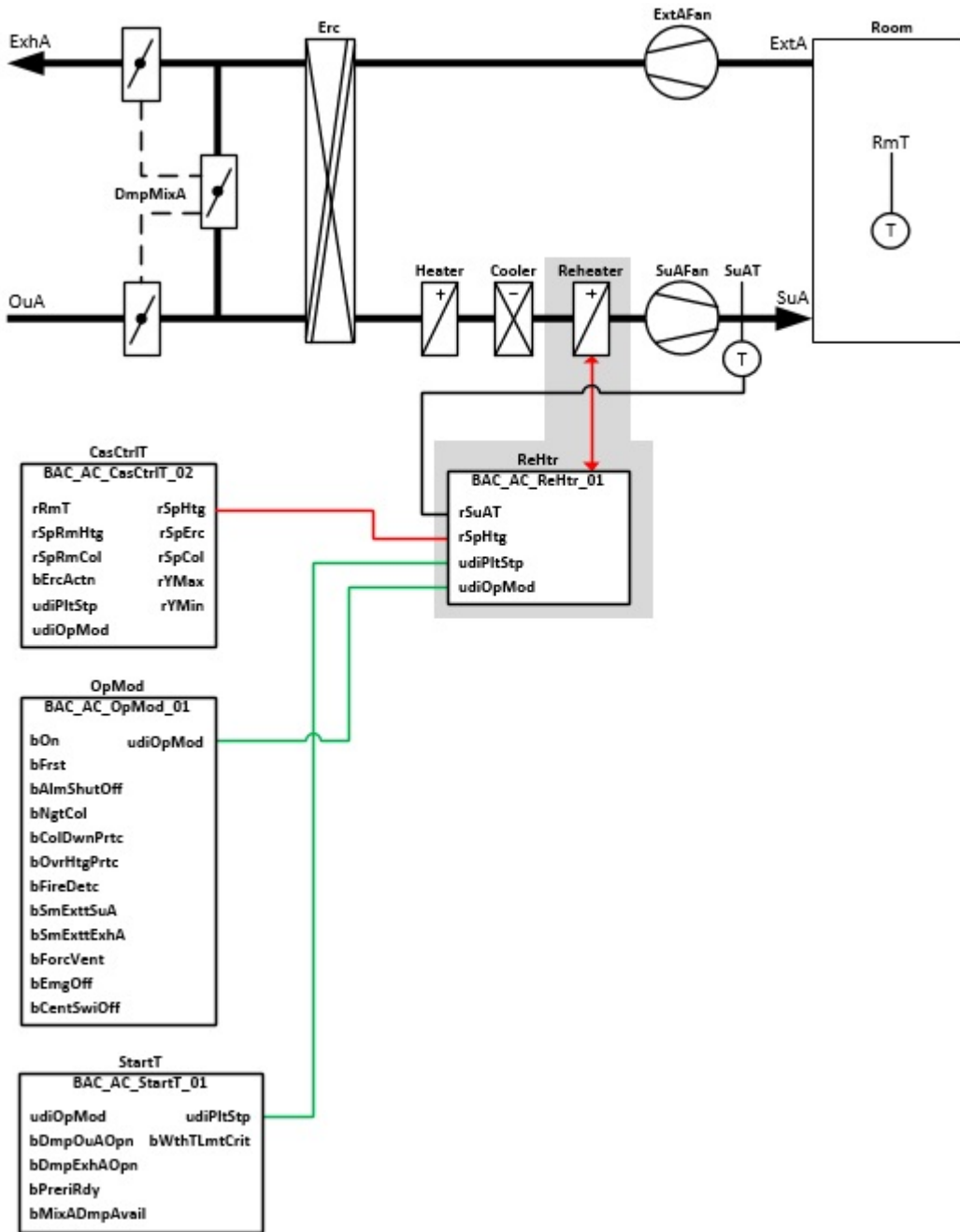
- Regelung der Zulufttemperatur
- Freigabe der Pumpe
- Ansteuerung des Regelventils

Die Freigabe des Zulufttemperaturreglers **SuA_PID** wird in dem Template über eine Auswertung der Anlagenbetriebsart **udiOpMod** gebildet.

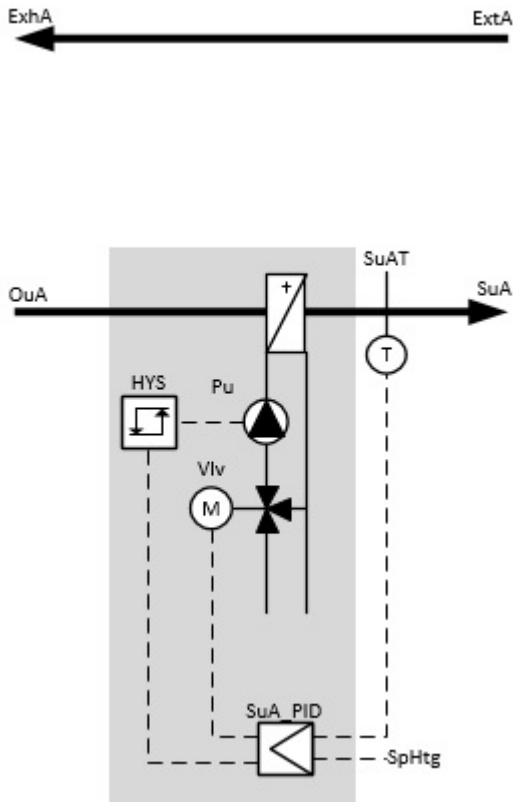
Schnittstelle



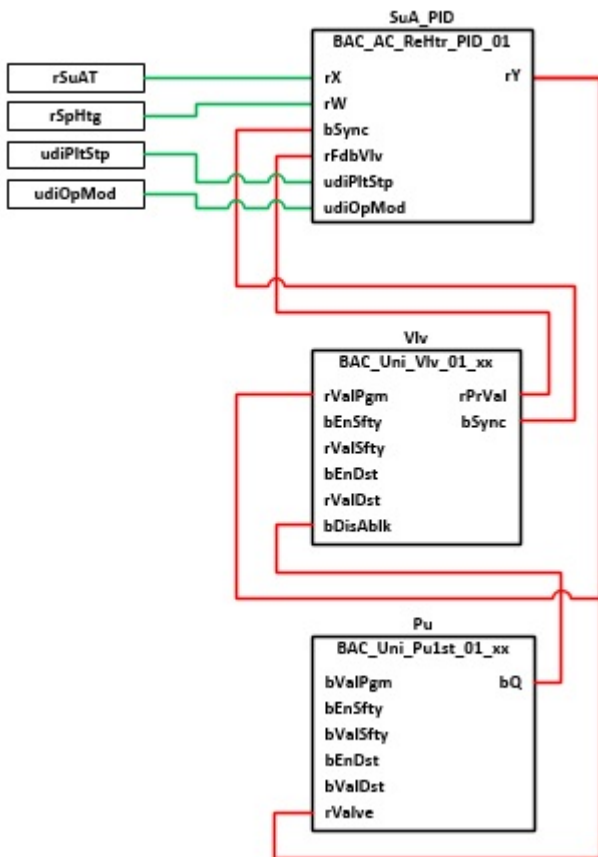
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

rSuAT      : REAL;
rSpHtg     : REAL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;

```

rSuAT: Messwert Zulufttemperatur

rSpHtg: Sollwert der Zulufttemperatur

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm [BAC_AC_StartT_01 \[► 538\]](#).

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl [BAC_AC_OpMod_01 \[► 524\]](#).

Programmbeschreibung

Instanz	Typ	Aufgabe
SuA_PID	BAC_AC_ReHtr_PID_01 [► 496]	Sub-Template Zulufttemperaturregelung. Die Temperaturregelung erfolgt stetig mittels eines PID-Sequenzreglers.
Vlv	BAC_Uni_Vlv_01_13 [► 608]	Sub-Template zur Ansteuerung eines stetigen Ventils
Pu	BAC_Uni_Pu1st_01_189 [► 579]	Sub-Template zur Ansteuerung einer einstufigen Pumpe

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.42 BAC_AC_ReHtr_PID_01**Funktionsbeschreibung**

Das Sub-Template **BAC_AC_ReHtr_PID_01** ist der Zulufttemperatur-Sequenzregler für einen Nacherhitzer.

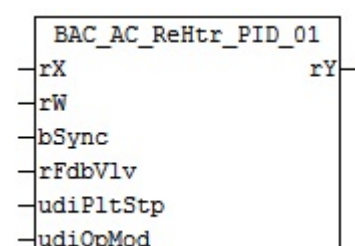
Die Referenzierung des Sollwertes, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte **X**, **W** und **Y**.

Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart **udiOpMod** und der globalen Temperatur-Kommunikationsstruktur **g_stSeqLinkT[PLT_NUM]**.

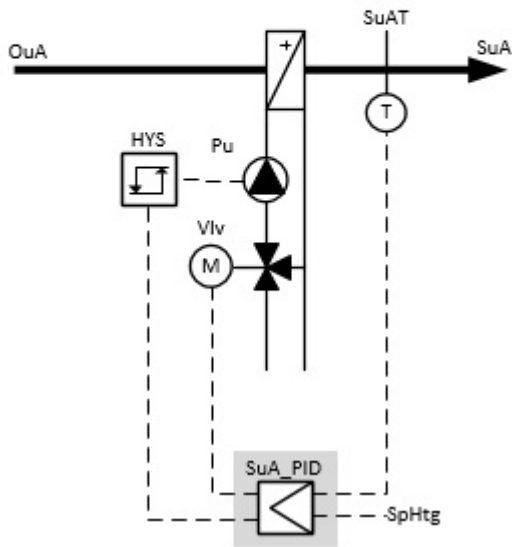
Diese Daten- und Befehlsstruktur ist das Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein [FB_BA_SeqLink \[► 171\]](#) einer Anlage.

Zur Anzeige der Reglerfreigabe dient das BACnet-BV-Objekt **En**.

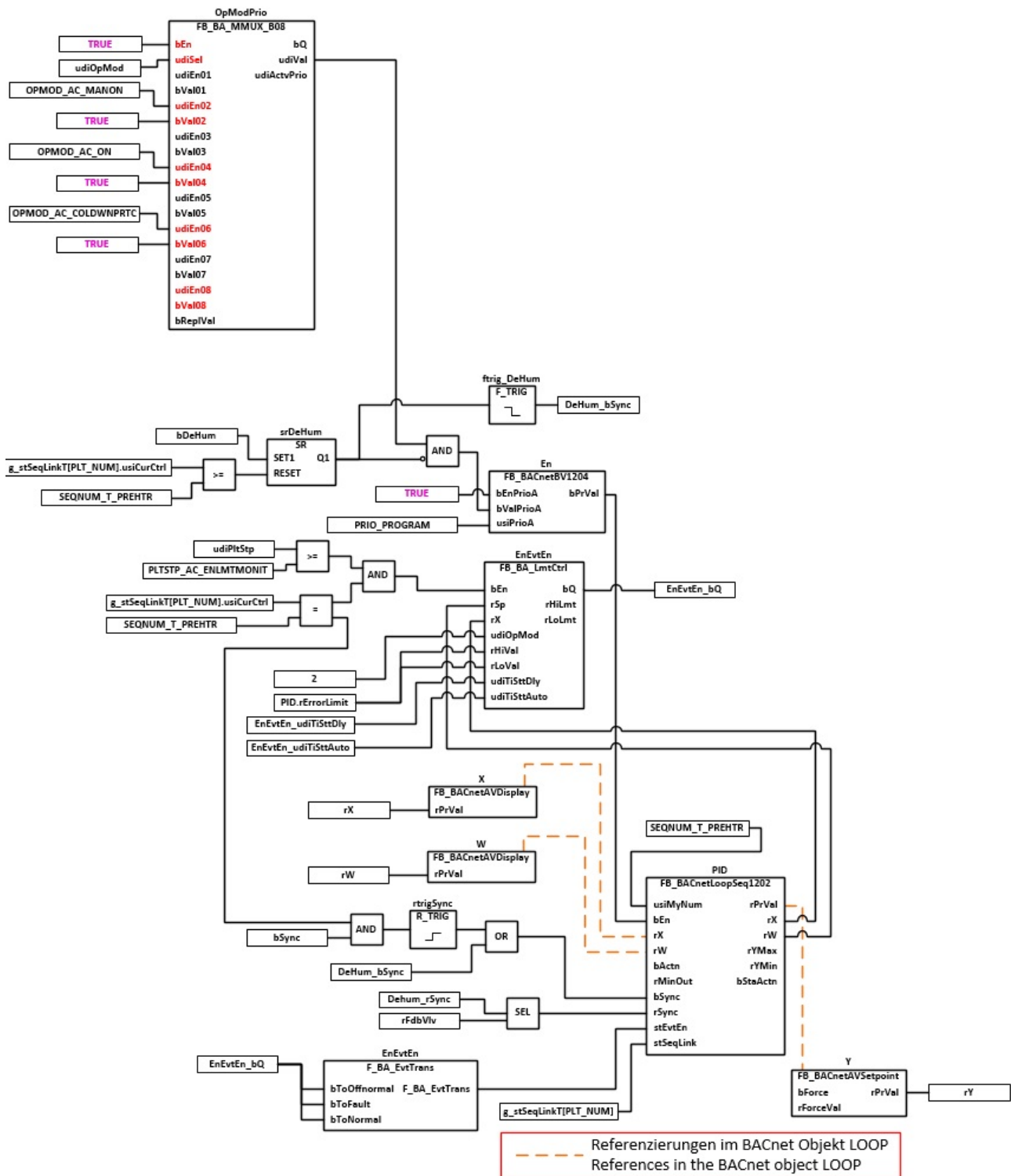
Die Grenzwertüberwachung des PID-Reglers wird in Abhängigkeit des Anlagenanfahrprozesses von dem Funktionsbaustein **EnEvtEn** gesteuert.

Schnittstelle

Anlagenschema



Blockschaltbild



VAR_INPUT

```

rX      : REAL;
rW      : REAL;
bSync   : BOOL;
rFdbVlv : REAL;
udiPltStp : UDINT;
udiOpMod : UDINT;
    
```

rX: Messwert Zulufttemperatur

rW: Sollwert der Zulufttemperatur

bSync: Eingang für die Synchronisation des Reglers

rFdbVlv: Stellungsrückmeldung Aktor

udiPltStp: Schritt beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartT 01 \[▶ 538\]](#)

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod 01 \[▶ 524\]](#)

VAR_OUTPUT

```
rY : REAL;
```

rY: Ausgabe der Stellgröße für das Regelventil

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm. \[▶ 182\]](#)

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm 01 \[▶ 372\]](#) mittels des Funktionsbausteins [FB BA AlarmPlt. \[▶ 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg \[▶ 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig, wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur [g_stSeqLinkT\[PLT_NUM\]](#) dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein [FB_BA_SeqLink](#) zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe																
X	FB BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts																
W	FB BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts																
OpModPri o	FB BA MMUX B08 [▶ 208]	Der Multiplexer definiert die Freigabebedingungen des Sequenzreglers in Abhängigkeit der Anlagenbetriebsart.																
		<table border="1"> <thead> <tr> <th>udiOpMod</th> <th></th> <th>Freigabe</th> <th>Bemerkung</th> </tr> </thead> <tbody> <tr> <td>OPMOD AC MANON [▶ 364]</td> <td>Hand ein</td> <td>TRUE</td> <td>Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet</td> </tr> <tr> <td>OPMOD AC ON [▶ 364]</td> <td>Ein</td> <td>TRUE</td> <td>Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm</td> </tr> <tr> <td>OPMOD AC COLDWNP RTC [▶ 364]</td> <td>Stützbetrieb, Auskühlschutz</td> <td>TRUE</td> <td>Die Anlage läuft in der Betriebsart Auskühlschutz</td> </tr> </tbody> </table>	udiOpMod		Freigabe	Bemerkung	OPMOD AC MANON [▶ 364]	Hand ein	TRUE	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet	OPMOD AC ON [▶ 364]	Ein	TRUE	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm	OPMOD AC COLDWNP RTC [▶ 364]	Stützbetrieb, Auskühlschutz	TRUE	Die Anlage läuft in der Betriebsart Auskühlschutz
		udiOpMod		Freigabe	Bemerkung													
		OPMOD AC MANON [▶ 364]	Hand ein	TRUE	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet													
OPMOD AC ON [▶ 364]	Ein	TRUE	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm															
OPMOD AC COLDWNP RTC [▶ 364]	Stützbetrieb, Auskühlschutz	TRUE	Die Anlage läuft in der Betriebsart Auskühlschutz															
En	FB BACnetBV1204 [▶ 94]	Das BV-Objekt dient zur Anzeige und Aktivierung der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay. Die Freigabe des Reglers erfolgt in Abhängigkeit der Anlagenbetriebsart.																

Instanz	Typ	Aufgabe
EnEvtEn	FB BA LmtCtrl [▶ 233]	<p>Das BACnet-Loop-Objekt PID überwacht die Funktion der Regelung in dem es den Sollwert W und den Istwert X miteinander vergleicht. Ist die Abweichung $W-X$ größer als das Property ErrorLimit, dann sendet das Loop-Objekt eine Meldung an die MBE.</p> <p>Beim Anlagenstillstand, im Moment des Starts und bis zum eingeregelter Zustand der Anlage wird das Melden des Loop-Objekts unterdrückt, damit keine falschen Meldungen an die MBE gesendet werden. Erst wenn die RLT-Anlage vollständig hochgefahren ist und die Regelung sich eingependelt hat, wird das Melden des Loop-Objekts freigegeben. Zusätzlich wird das Melden aktiviert, wenn die Regelung nach langer Zeit nicht in dem vom Property ErrorLimit definierten Bereich um den Sollwert gelangt ist.</p> <p>Die Freigabe des objektinternen Meldens erfolgt indem auf die BACnetEventTransitionBits des Loop-Objekts geschrieben wird.</p> <p>Für die Freigabe des Meldens vom Loop-Objekt müssen die folgenden Bedingungen erfüllt sein:</p> <p>1. Das Anlagenstartprogramm BAC_AC_StartT_01 [▶ 538] hat die Überwachung der Regelung und die Grenzwertüberwachung der Fühler freigegeben udiPitStp >= PLTSTP_AC_ENLMTMONIT und Der Nacherhitzerregler ist der aktive Regler in der Regelsequenz. g_stSeqLinkTIPLT_NUM [▶ 364].usiCurCtrl = SEQNUM_T_REHR und Die Zulufttemperatur hat sich dem Sollwert so weit angenähert, dass sie sich in einem Bereich zwischen rSp - ErrorLimit und rSp + ErrorLimit eingependelt hat. und Die Zulufttemperatur muss mindestens für die Dauer von EnEvtEn_udiTiSttDIy innerhalb des Bereiches von rSp - ErrorLimit und rSp + ErrorLimit verblieben sein.</p> <p>2. Der Timer EnEvtEn_udiTiSttAuto ist abgelaufen und die Regelung hat Ihren Sollwertbereich nicht erreicht.</p>
	F_BA_EvtTrans	schreibt auf die BACnetEventTransitionBits to_offnormal, to_fault und to_normal des Loop-Objekts. Dazu muss der Eingang bEnEvtEn = TRUE sein
PID	FB_BACnetLoopSeq12_02 [▶ 112]	Sequenzregler Zulufttemperatur Nacherhitzer.
rtrigSync	R_TRIG	Bei einer steigenden Flanke am Eingang bSync wird das Loop-Objekt auf den Wert von IrSync aufsynchronisiert. Wenn das Regelventil des Nacherhitzers durch Schreiben einer höheren Priorität auf das zugehörige AO-Objekt von der MBE oder durch Betätigen der örtlichen Vorrangbedienung übersteuert wurde, weicht die aktuelle Position des Regelventils von dem Ausgang des Loop-Objekts ab. Mit den Variablen bSync und rFdbVlv kann die Synchronität zwischen der Position des Regelventils und dem Regler wieder hergestellt werden.
Y	FB_BACnetAVSetpoint [▶ 70]	Dass AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.43 BAC_AC_PreHtr_01

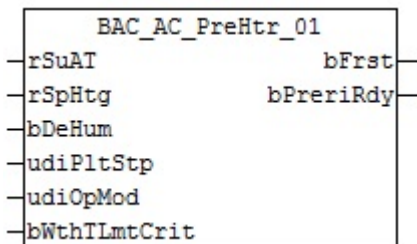
Anwendung

Das Aufruftemplate **BAC_AC_PreHtr_01** dient zur Steuerung, Regelung eines Warmwasserluftherhitzers.

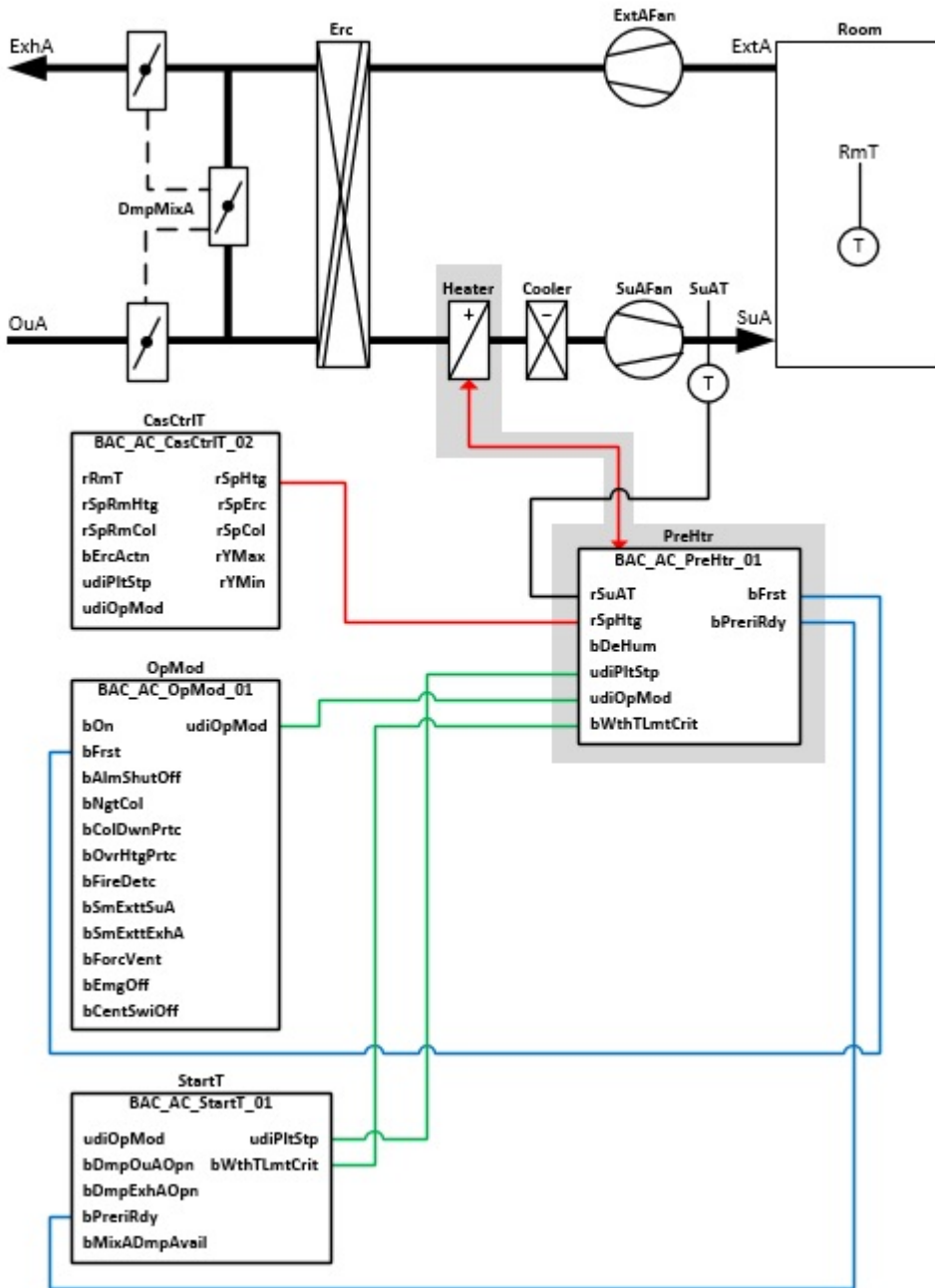
Die wesentlichen Aufgaben des Templates sind:

- Regelung der Zulufttemperatur
- Regelung der Rücklauftemperatur
- Frostüberwachung luftseitig mit Frostschutzthermostaten **FrstThermostat**
- Frostüberwachung luftseitig mit stetigen Frostschutzfühler **FrstT**
- Frostüberwachung wasserseitig mit Rücklauftemperaturfühler **RetWtrT**
- Freigabe der Erhitzerpumpe
- Ansteuerung und Überwachung des Erhitzerventils

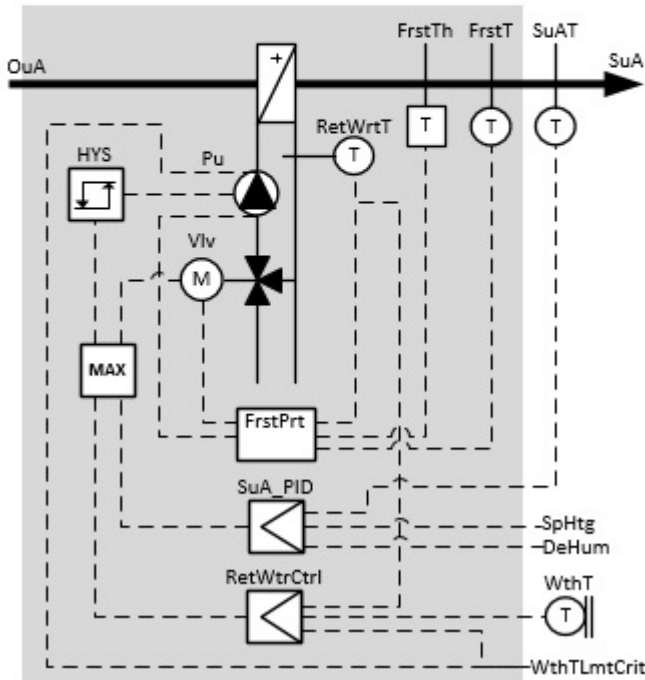
Schnittstelle



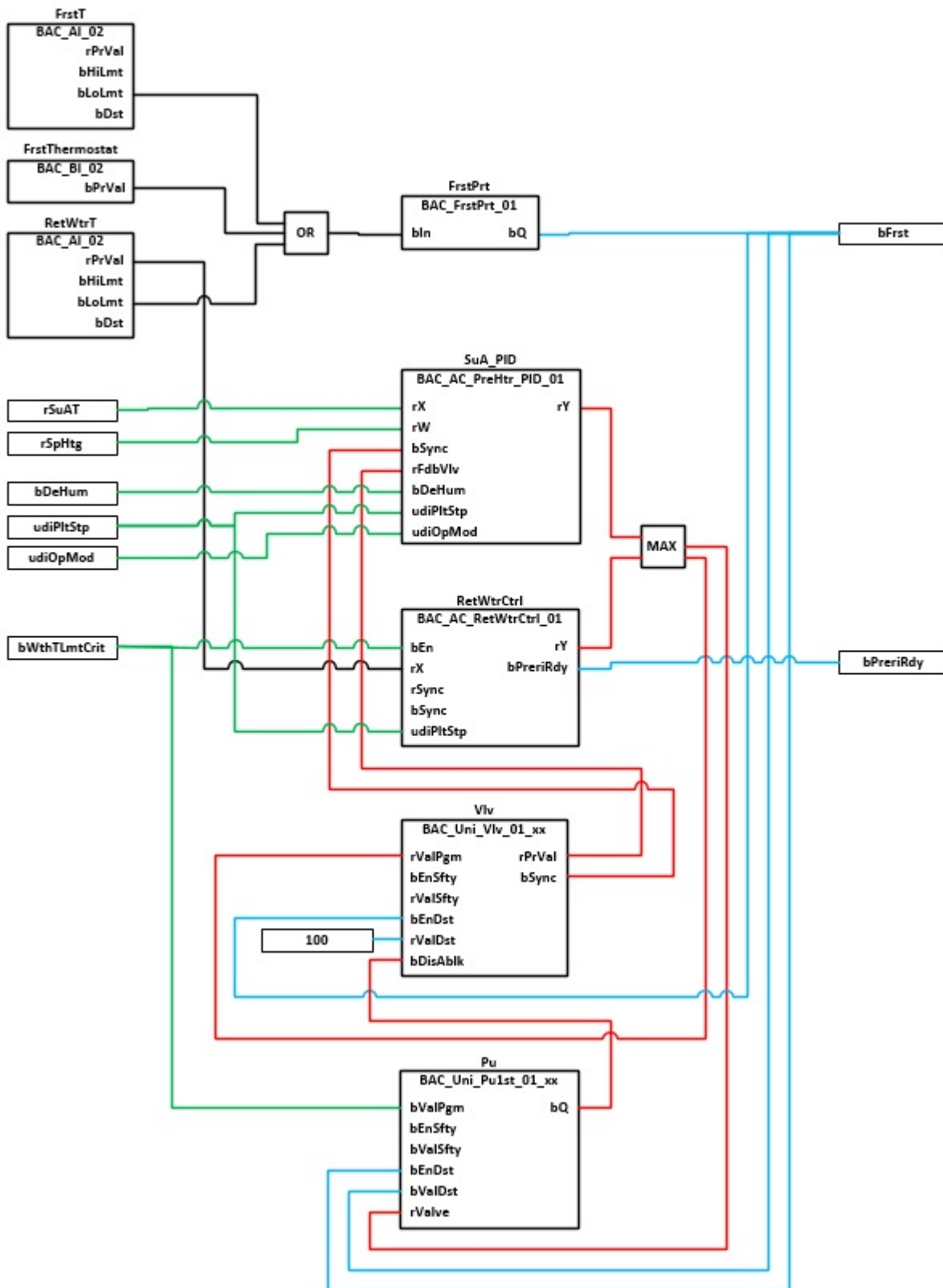
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

rSuAT      : REAL;
rSpHtg     : REAL;
bDeHum     : BOOL;
udiPltStp  : UDINT;
udiOpMod   : UDINT;
bWthTLmttCrit : BOOL;

```

rSuAT: Messwert der Zulufttemperatur

rSpHtg: Sollwert für die Zulufttemperatur

bDeHum: Betriebsmeldung Entfeuchtebetrieb

Bei aktiver Entfeuchtung wird der Zulufttemperaturregler **SuA_PID** gesperrt und damit aus der Sequenzregelung herausgenommen.

udiPltStp: Schritte Anlagenstartsequenz. Erzeugt werden die Anlagenschritte im Anlagenstartprogramm BAC_AC StartT_01 [[▶ 538](#)].

udiOpMod: Anlagenbetriebsart. Ermittelt wird die Anlagenbetriebsart in dem Programm für die Betriebsartenauswahl BAC_AC OpMod_01 [[▶ 524](#)].

bWthTLmtCrit: Zustandsmeldung Witterungstemperatur kritisch. Erzeugt wird diese Information im Anlagenstartprogramm BAC_AC StartT_01 [[▶ 538](#)].

Folgende Aktionen werden beim Unterschreiten eines kritischen Außentemperaturwertes ausgelöst:

- Einschaltung Erhitzerpumpe, Pumpenzwangslauf
- Freigabe Rücklauftemperaturegelung BAC_AC RetWtrCtrl_01 [[▶ 511](#)]
- Während des Anlagenstarts ist der Vorspülbetrieb aktiv, siehe BAC_AC RetWtrCtrl_01 [[▶ 511](#)]

VAR_OUTPUT

```
bFrst      : BOOL;
bPreriRdy : BOOL;
```

bFrst: Ausgabe Forstschutzprogramm aktive.

bPreriRdy: Ausgabe Vorspülprozess abgeschlossen.

Programmbeschreibung

Instanz	Typ	Aufgabe
RetWtrT	<u>BAC_AI_02</u> [▶ 685]	Sub-Template AI-Objekt Rücklauftemperaturfühler
FrstT	<u>BAC_AI_02</u> [▶ 685]	Sub-Template AI-Objekt Frostschutzfühler luftseitig
FrstThermostat	<u>BAC_BI_02</u> [▶ 704]	Sub-Template BI-Objekt Frostschutzthermostat
SuA_PID	<u>BAC_AC PreHtr_PID_01</u> [▶ 506]	Sub-Template Loop-Objekt Zulufttemperaturregelung
RetWtrCtrl	<u>BAC_AC RetWtrCtrl_01</u> [▶ 511]	Sub-Template Rücklauftemperaturregelung des Warmwasserluftherhitzer
FrstPrt	<u>BAC_FrstPrt_01</u> [▶ 515]	Sub-Template Frostschutzprogramm
Vlv	<u>BAC_Uni_Vlv_01_13</u> [▶ 608]	Sub-Template Regelventil
Pu	<u>BAC_Uni_Pu1st_01_189</u> [▶ 579]	Sub-Template Pumpe
	MAX	MAX-Auswahl der Stellsignale Zulufttemperaturregelung SuA_PID und Rücklauftemperaturregelung RetWtrCtrl
	OR	Sammelt Frostereignisse ein und gibt diese an das Frostschutzprogramm weiter. Der Frostfall tritt ein bei den folgenden Bedingungen <ul style="list-style-type: none"> • Ansprechen des Frostschutzthermostaten • Unterschreitung des LowLimit der Lufttemperatur hinter dem Heizregister • Unterschreitung des LowLimit der Rücklauftemperatur (RetWtr.bLoLmt = TRUE)

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.44 BAC_AC_PreHtr_PID_01

Funktionsbeschreibung

Das Sub-Template **BAC_AC_PreHtr_PID_01** ist der Sequenzregler für einen Vorerhitzer.

Die Referenzierung des Sollwertes, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte **X**, **W** und **Y**.

Die Freigabe des PID-Sequenzreglers erfolgt anhand der Anlagenbetriebsart **udiOpMod** und der globalen Temperatur-Kommunikationsstruktur **g_stSeqLinkT[PLT_NUM]**.

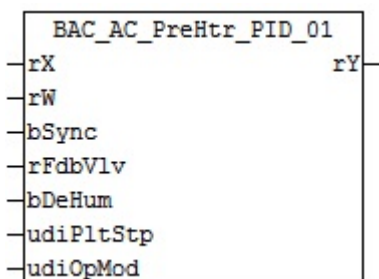
Diese Daten- und Befehlsstruktur ist das Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein **FB_BA_SeqLink** [► 171] einer Anlage.

Zur Anzeige der Reglerfreigabe dient das BACnet-BV-Objekt **En**.

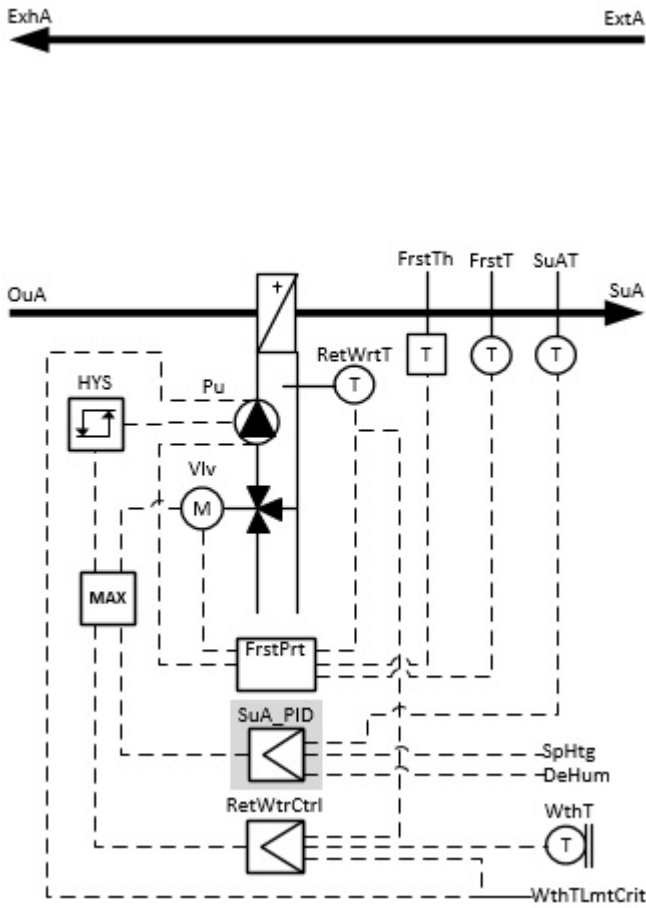
Die Grenzwertüberwachung des PID-Reglers wird in Abhängigkeit des Anlagenanfahrprozesses von dem Funktionsbaustein **EnEvtEn** gesteuert.

Im Entfeuchtebetrieb **bDeHum** = TRUE wird der PID-Sequenzregler gesperrt.

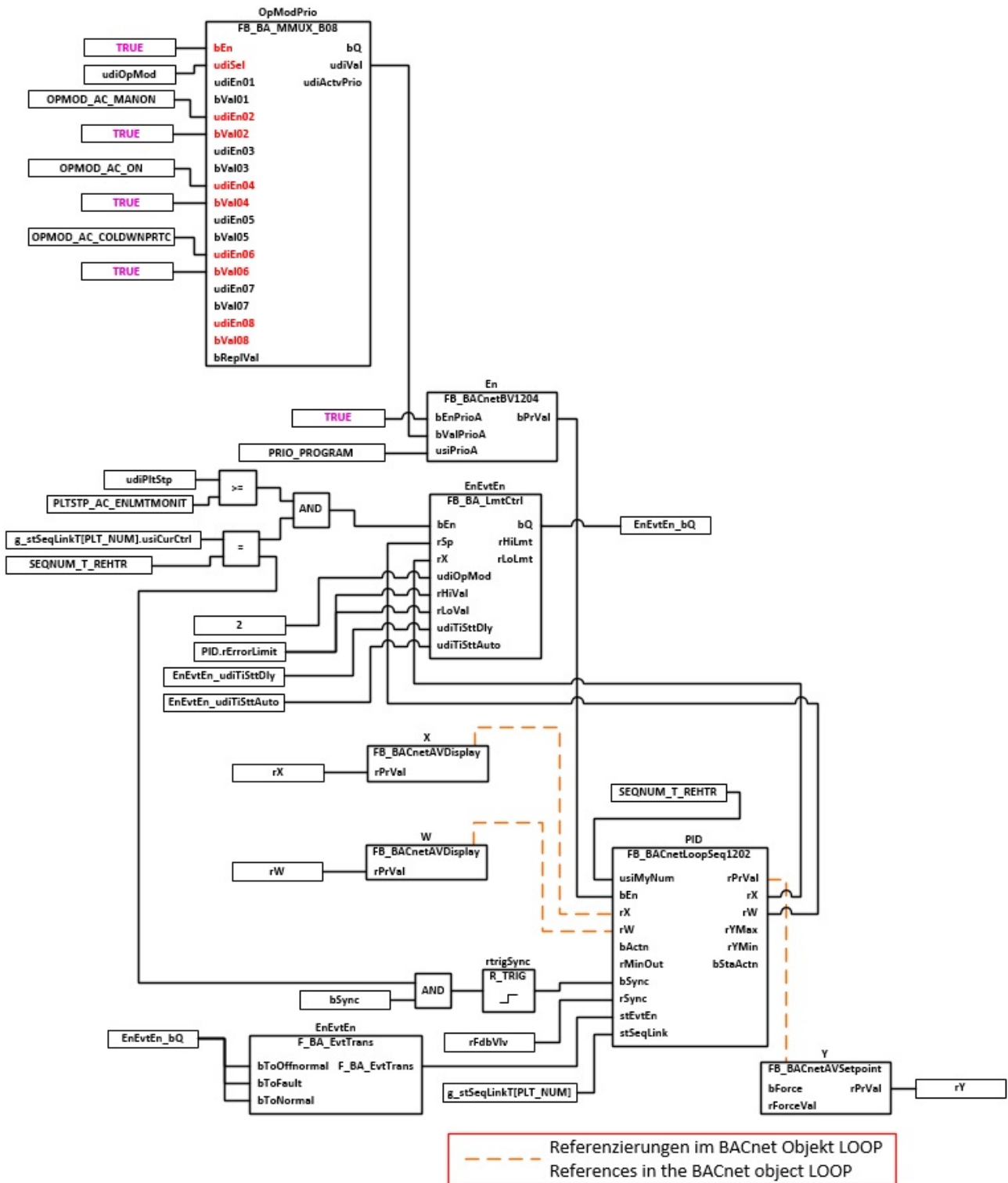
Schnittstelle



Anlagenschema



Blockschaltbild



--- Referenzierungen im BACnet Objekt LOOP
References in the BACnet object LOOP

VAR_INPUT

```

rX      : REAL;
rW      : REAL;
bSync   : BOOL;
rFdbVlv : REAL;
bDeHum  : BOOL;
udiPltStp : UDINT;
udiOpMod : UDINT;
    
```

rX: Messwert Zulufttemperatur

rW: Sollwert der Zulufttemperatur

bSync: Eingang für die Synchronisation des Reglers

rFdbVlv: Stellungsrückmeldung Aktor

bDeHum: Entfeuchtung aktiv, Sperrung der Freigabe des Zulufttemperaturreglers.

udiPltStp: Schritt beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartT 01 \[▶ 538\]](#)

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod 01 \[▶ 524\]](#)

VAR_OUTPUT

```
rY : REAL;
```

rY: Stellgrößenausgabe Regelventil

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm. \[▶ 182\]](#)

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm 01 \[▶ 372\]](#) mittels des Funktionsbausteins [FB BA AlarmPlt. \[▶ 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg \[▶ 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur [g_stSeqLinkT\[PLT_NUM\]](#) dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein [FB_BA_SeqLink](#) zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe												
X	FB BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts												
W	FB BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts												
OpModPri o	FB BA MMUX B08 [▶ 208]	Der Multiplexer definiert die Freigabebedingungen des Sequenzreglers in Abhängigkeit der Anlagenbetriebsart.												
		<table border="1"> <thead> <tr> <th colspan="2">udiOpMod</th> <th>Freigabe</th> <th>Bemerkung</th> </tr> </thead> <tbody> <tr> <td>OPMOD AC MANON [▶ 364]</td> <td>Hand ein</td> <td>TRUE</td> <td>Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet</td> </tr> <tr> <td>OPMOD AC ON [▶ 364]</td> <td>Ein</td> <td>TRUE</td> <td>Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm</td> </tr> </tbody> </table>	udiOpMod		Freigabe	Bemerkung	OPMOD AC MANON [▶ 364]	Hand ein	TRUE	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet	OPMOD AC ON [▶ 364]	Ein	TRUE	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm
		udiOpMod		Freigabe	Bemerkung									
OPMOD AC MANON [▶ 364]	Hand ein	TRUE	Die Anlage ist manuell über den Anlagenwahlschalter eingeschaltet											
OPMOD AC ON [▶ 364]	Ein	TRUE	Die Anlage läuft im Automatikbetrieb über das Zeitschaltprogramm											

Instanz	Typ	Aufgabe			
		OPMOD AC COLDWNPRTC ▶ 364 <table border="1" data-bbox="994 197 1230 320"> <tr> <td>Stützbetrieb, Auskühl schutz</td> <td>TRUE</td> <td>Die Anlage läuft in der Betriebsart Auskühlschutz</td> </tr> </table>	Stützbetrieb, Auskühl schutz	TRUE	Die Anlage läuft in der Betriebsart Auskühlschutz
Stützbetrieb, Auskühl schutz	TRUE	Die Anlage läuft in der Betriebsart Auskühlschutz			
En	FB BACnetBV1204 ▶ 94	Das BV-Objekt dient zur Anzeige und Aktivierung der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay. Die Freigabe des Reglers erfolgt in Abhängigkeit der Anlagenbetriebsart.			
	AND	Der AND-Baustein verriegelt die Freigabe des Sequenzreglers im Entfeuchtebetrieb. Die Umschaltung vom Vorerhitzer auf den Nacherhitzer erfolgt im Template BAC AC SeqT_01 ▶ 532 durch den Sequenzlinker.			
EnEvtEn	FB BA LmtCtrl ▶ 233	<p>Das BACnet-Loop-Objekt PID überwacht die Funktion der Regelung in dem es den Sollwert W und den Istwert X miteinander vergleicht. Ist die Abweichung $W-X$ größer als das Property ErrorLimit, dann sendet das Loop-Objekt eine Meldung an die MBE.</p> <p>Beim Anlagenstillstand, im Moment des Starts und bis zum eingeregelteten Zustand der Anlage wird das Melden des Loop-Objekts unterdrückt, damit keine falschen Meldungen an die MBE gesendet werden. Erst wenn die RLT-Anlage vollständig hochgefahren ist und die Regelung sich eingependelt hat, wird das Melden des Loop-Objekts freigegeben. Zusätzlich wird das Melden aktiviert, wenn die Regelung nach langer Zeit nicht in dem vom Property ErrorLimit definierten Bereich um den Sollwert gelangt ist.</p> <p>Die Freigabe des objektinternen Meldens erfolgt indem auf die BACnetEventTransitionBits des Loop-Objekts geschrieben wird.</p> <p>Für die Freigabe des Meldens vom Loop-Objekt müssen die folgenden Bedingungen erfüllt sein:</p> <ol style="list-style-type: none"> <p>Das Anlagenstartprogramm BAC AC StartT_01 ▶ 538 hat die Überwachung der Regelung und die Grenzwertüberwachung der Fühler freigegeben</p> <p>udiPltStp >= PLTSTP_AC_ENLMTMONIT und Der Erhitzerregler ist der aktive Regler in der Regelsequenz. g_stSeqLinkT[PLT_NUM] ▶ 364.usiCurCtrl = SEQNUM_T_PREHTR und Die Zulufttemperatur hat sich dem Sollwert so weit angenähert, dass sie sich in einem Bereich zwischen rSp - ErrorLimit und rSp + ErrorLimit eingependelt hat. und Die Zulufttemperatur muss mindestens für die Dauer von EnEvtEn_udiTiSttDly innerhalb des Bereiches von rSp - ErrorLimit und rSp + ErrorLimit verblieben sein.</p> <p>Der Timer EnEvtEn_udiTiSttAuto ist abgelaufen und die Regelung hat Ihren Sollwertbereich nicht erreicht.</p> 			
	F_BA_EvtTrans	schreibt auf die BACnetEventTransitionBits to_offnormal, to_fault und to_normal des Loop-Objekts. Dazu muss der Eingang bEnEvtEn = TRUE sein			

Instanz	Typ	Aufgabe
PID	FB_BACnetLoopSeq1202 [▶ 112]	Sequenzregler Zulufttemperatur Vorerhitzer.
rtrigSync	R_TRIG	Bei einer steigenden Flanke am Eingang bSync wird das Loop-Objekt auf den Wert von IrSync aufsynchronisiert. Wenn das Regelventil des Vorerhitzers durch Schreiben einer höheren Priorität auf das zugehörige AO-Objekt von der MBE oder durch Betätigen der örtlichen Vorrangbedienung übersteuert wurde, weicht die aktuelle Position des Regelventils von dem Ausgang des Loop-Objekts ab. Mit den Variablen bSync und rFdbVlv kann die Synchronität zwischen der Position des Regelventils und dem Regler wieder hergestellt werden.
Y	FB_BACnetAVSetpoint [▶ 70]	Dass AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.45 BAC_AC_RetWtrCtrl_01

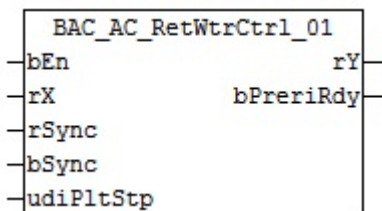
Anwendung

Das Aufruftemplate BAC_AC_RetWtrCtrl_01 dient zur Steuerung und Regelung der Rücklauftemperatur des Warmwasserluftherhitzers.

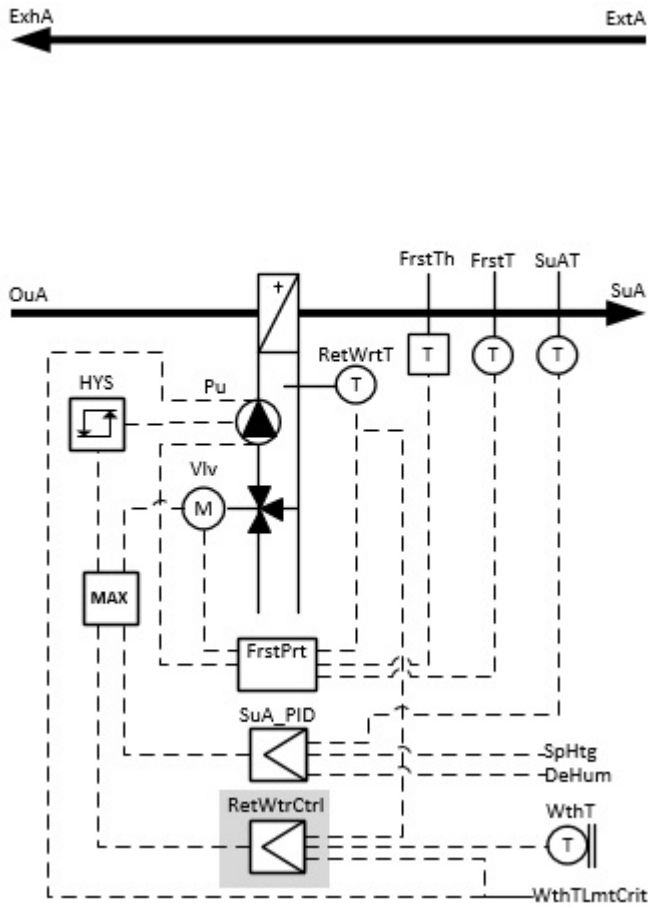
Die wesentlichen Aufgaben des Templates sind:

- Regelung der Rücklauftemperatur während des Vorspülprozesse beim Anlagenstart.
- Regelung der Rücklauftemperatur auf einen Frostschutzsollwert bei Anlagenbetrieb und Stillstand.

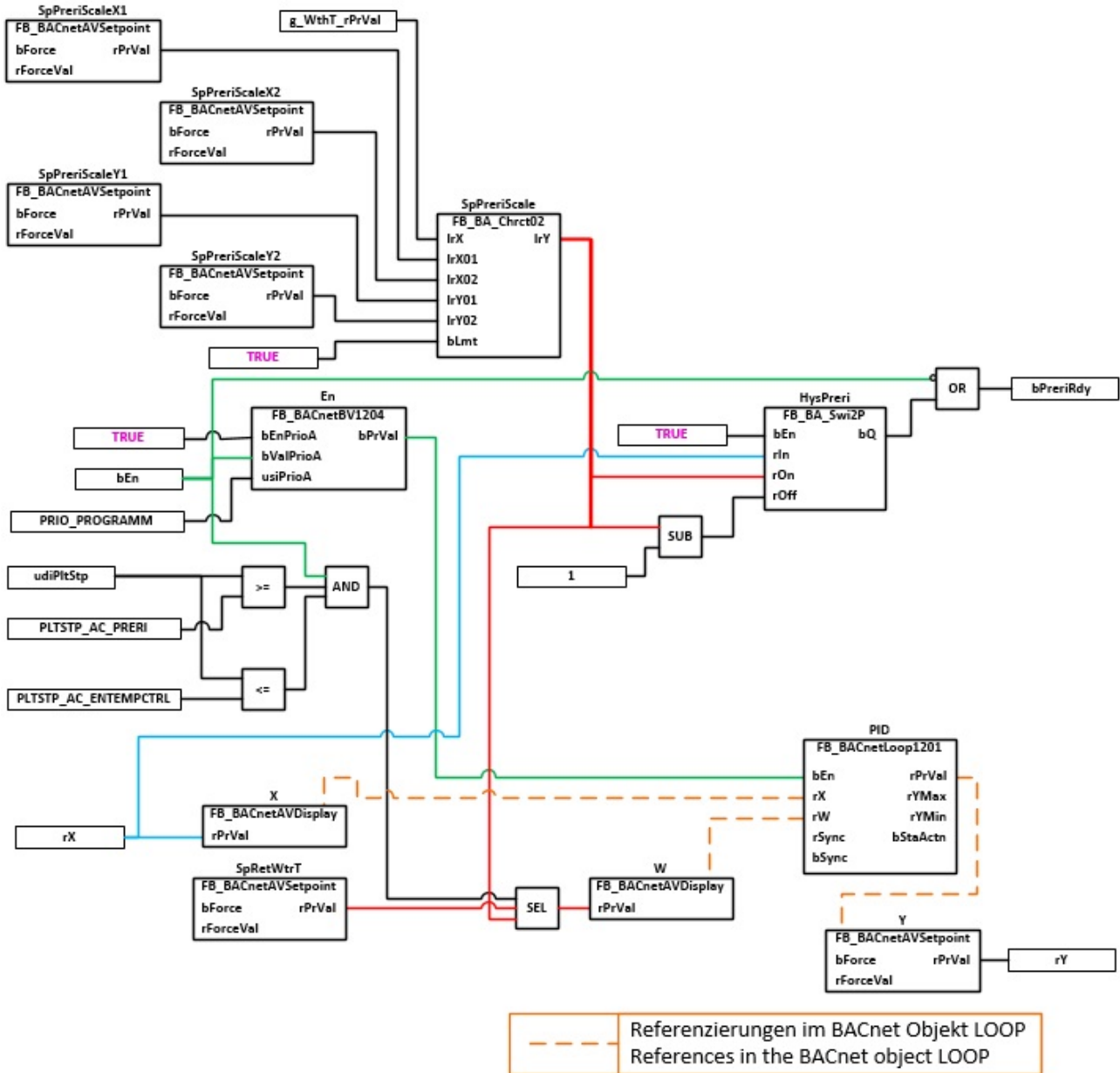
Schnittstelle



Diagramm



Blockschaltbild



VAR_INPUT

```

bEn      : BOOL;
rX       : REAL;
rSync   : BOOL;
bSync   : UDINT;
udiPltStp : UDINT;
    
```

bEn: Freigabe der Rücklauf temperaturregelung.

rX: Messwert der Rücklauf temperaturregelung

rSync: Synchronisationswert für den Regler **PID**

bSync: Eingang für die Synchronisation des Reglers **PID**

udiPltStp: Schritt beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartT_01](#) [▶ 538]

VAR_OUTPUT

```

rY       : REAL;
bPreriRdy : BOOL;
    
```

rY: Stellgrößen ausgabe

bPreriRdy: Meldung Vorspülprozess abgeschlossen

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmerbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_CmnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
SpPreriScaleX1	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Sollwertes Vorspülen Skalierungspunkt X1
SpPreriScaleX2	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Sollwertes Vorspülen Skalierungspunkt X2
SpPreriScaleY1	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Sollwertes Vorspülen Skalierungspunkt Y1
SpPreriScaleY2	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Sollwertes Vorspülen Skalierungspunkt Y2
SpRetWtrT	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Sollwert Rücklauftemperatur Wasser
SpPreriScale	FB_BA_Chrcct02 [► 173]	Funktionsbaustein zur Berechnung der Vorspültemperatur in Abhängigkeit der Außentemperatur für die Rücklauftemperaturregelung während des Anlagenstarts.
HysPreri	FB_BA_Swi2P [► 148]	Beim Anfahren der RLT-Anlage wird bei niedrigen Außentemperaturen zunächst das Heizregister mit warmen Wasser vorgespült. Erst danach werden die Klappen geöffnet und die Ventilatoren eingeschaltet. Der Funktionsbaustein HysPreri schaltet beim Erreichen der Vorspültemperatur SpPreriScale_IrY die Variable bPreriRdy auf TRUE. Damit wird dem Anlagenstartprogramm BAC_AC_StartT_01 [► 538] mitgeteilt, dass der Vorspülprozess abgeschlossen ist. Ist die Eingangsvariable bEn während des Anlagenstarts nicht auf TRUE gesetzt, so wird bPreriRdy sofort gesetzt ohne dass der Vorspülprozess abgeschlossen sein muss. Damit erfolgt bei höheren Außentemperaturen ein Anlagenstart der RLT-Anlage ohne Vorspülen des Heizregisters.
	SUB	Der Wert der Subtraktion ist der untere Schaltpunkt des Zweipunktschalters HysPreri .
	OR	Gibt den Status des Vorspülens durch HysPreri oder bEn aus.

Instanz	Typ	Aufgabe
X	FB_BACnetAVDisplay [▶ 69]	AV-Objekt zur Anzeige der Rücklauftemperatur des Warmwasserluftherhitzers. Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts PID .
W	FB_BACnetAVDisplay [▶ 69]	AV-Objekt zur Anzeige des Sollwertes des Rücklauftemperaturreglers. Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts PID .
	>=, <=, AND, SEL	Der Wert dieses Netzwerkes gibt den Sollwert für die Rücklauftemperaturregelung des Warmwasserluftherhitzers PID vor.
En	FB_BACnetBV1204 [▶ 94]	Das BV-Objekt dient zur Anzeige und Aktivierung der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay. Die Freigabe des Reglers erfolgt über die Eingangsvariable bEn .
PID	FB_BACnetLoop1201 [▶ 99]	Loop-Objekt Rücklauftemperaturregelung des Warmwasserluftherhitzers. Es ist zuständig für das Vorspülen des Heizregisters während des Anlagenstarts der Lüftungsanlage und der Regelung der Rücklauftemperatur auf den Sollwert SpRetWtrT . Die Rücklauftemperaturregelung wird bei niedrigen Außentemperaturen durch die Eingangsvariable bEn frei gegeben. Die Freigabe erfolgt in dem Aufruftemplate <u>BAC_AC_PreHtr_01</u> [▶ 501].
Y	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Anzeige der Stellgröße der Rücklauftemperaturregelung. Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts PID .

Versionshistorie

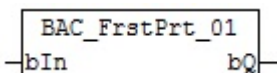
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.46 BAC_FrstPrt_01

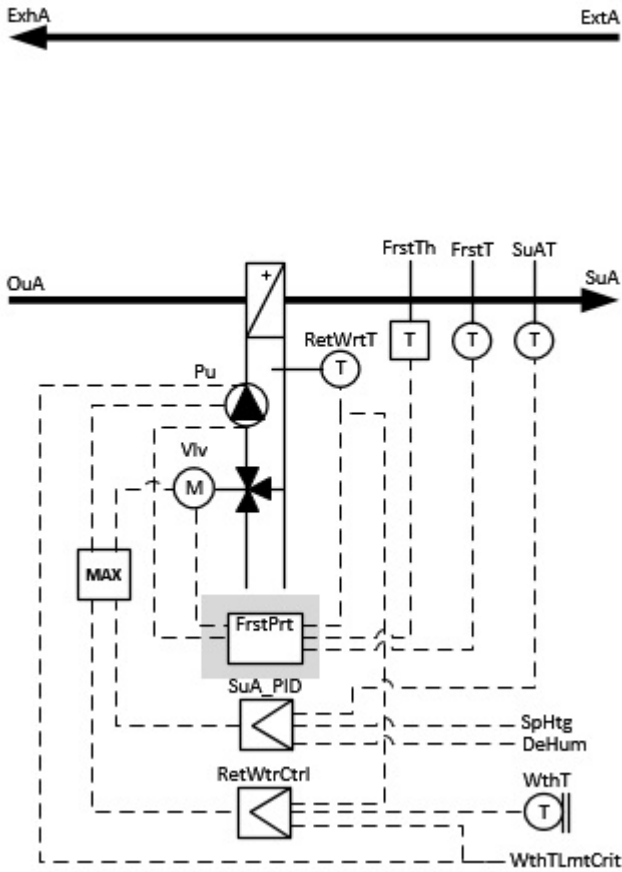
Funktionsbeschreibung

Das Template realisiert die Frostüberwachung eines Heizregister in einer raumluftechnischen Anlage.

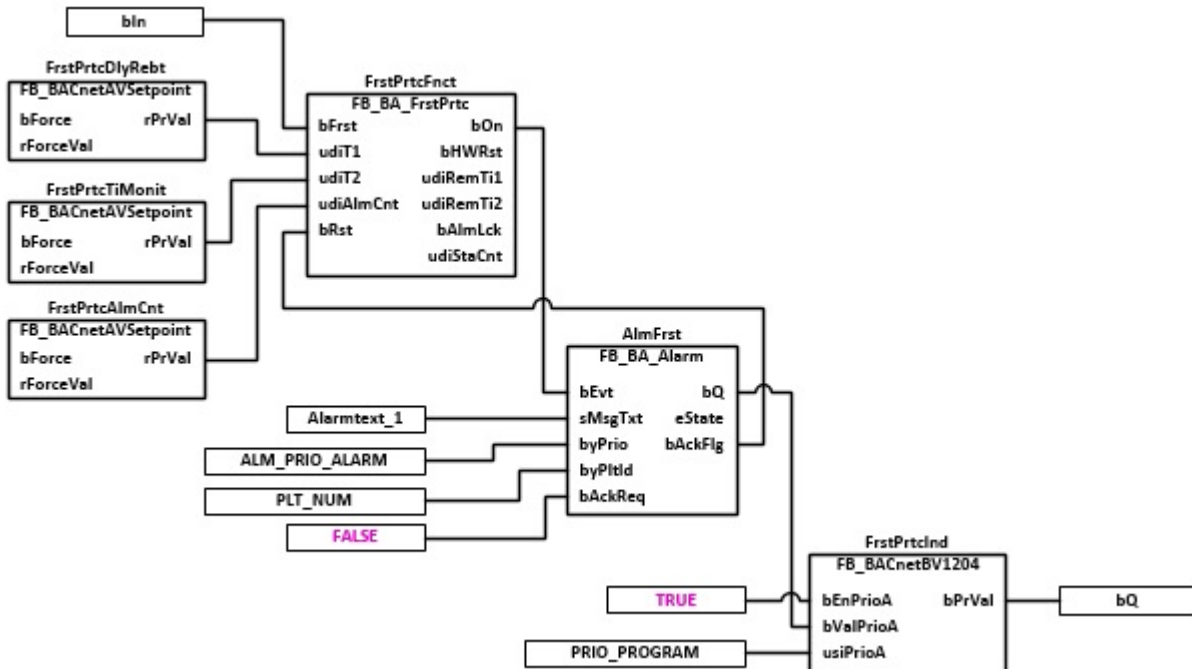
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

bIn : BOOL;

bIn: Ereignis Frost

VAR_OUTPUT

bQ : BOOL;

bQ: Meldung Frostschutz aktive

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [[▶ 182](#)]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [[▶ 372](#)] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
FrstPrtcDlyRebt	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Frostschutztimers T1 (Zeit innerhalb dessen der Frostschutzalarm wieder erloschen sein muss)
FrstPrtcTiMonit	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Frostschutztimers T2 (Zeitintervall innerhalb dessen kein weiterer Frostalarm erfolgen darf ohne das dieser manuell quittiert werden muss)
FrstPrtcAlmCnt	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe der maximalen Zahl der automatischen Anlagenneustarts nach eintreten des Frostfalls
FrstPrtcFncf	FB_BA_FrstPrtc [▶ 228]	Der Funktionsbaustein ist das Kernstück der Frostschutzüberwachung.
AlmFrst	FB_BA_Alarm	Der Funktionsbaustein AlmFrst erfasst das Ereignis Frostschutz. Ausserdem wird über den Ausgang bAckFlg ein Quittierungsimpuls an die Frostschutzüberwachung FrstPrtcFncf weiter geleitet. Aktionen, die nach dem Eingang der Frostschutz aktiv erfolgen sollen, können im Template am Funktionsbaustein AlmFrst parametrierf werden.
FrstPrtcInd	FB_BACnetBV1204 [▶ 94]	BV-Objekt zur Anzeige des Frostalarms

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.47 BAC_AC_VAV_01_xx

Funktionsbeschreibung

Das Template dient zur Ansteuerung eines Volumenstromreglers mittels analogen Ausgang.

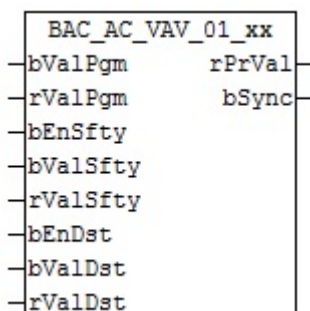
Ausstattungsvarianten

Das Template **BAC_AC_VAV_01_xx** existiert in verschiedenen Ausstattungsvarianten.

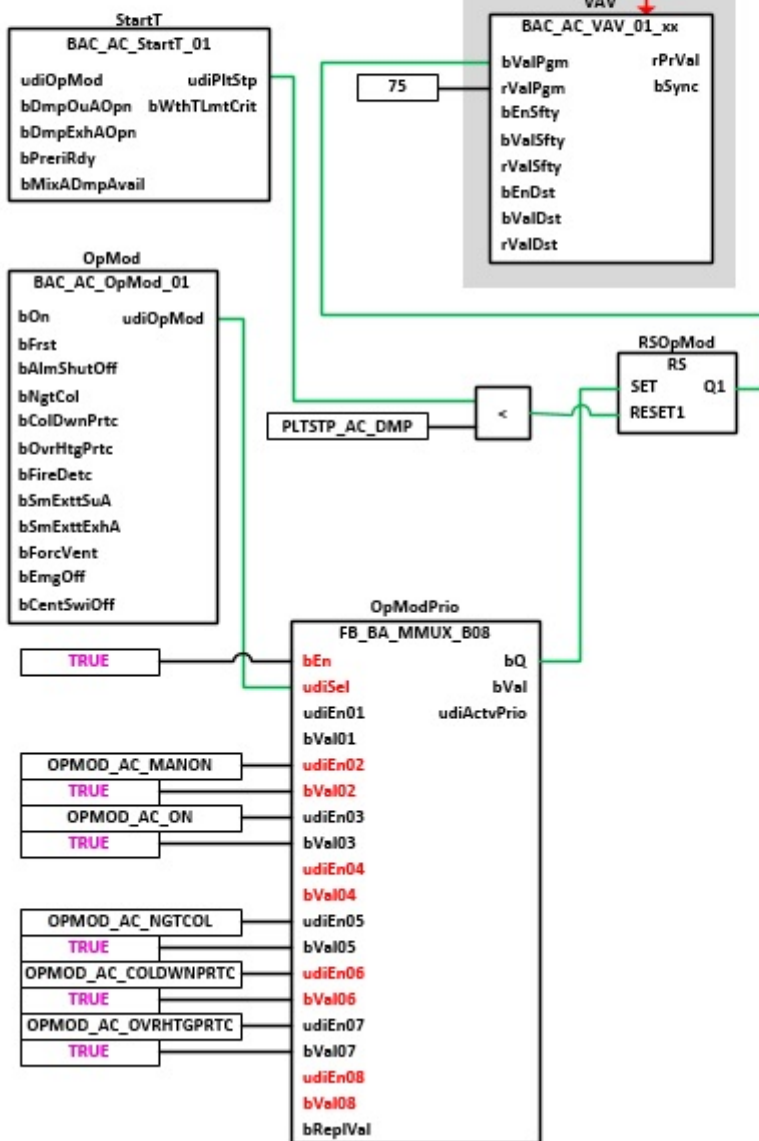
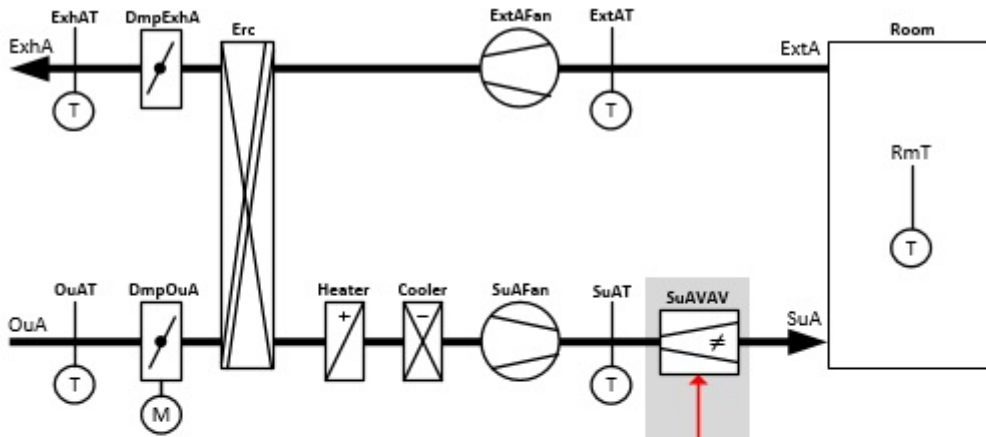
Die Ausstattungsvarianten des Volumenstromreglers werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

Optionen	Zwangsübersteuerung (Zwangssteuerung)	Rückmeldung des Volumenstroms bzw. der Klappensteuerung (Rm)	mechanische Vorrangbedienung Stellungsrückmeldung Poti (Rm-Ausg)	mechanische Vorrangbedienung Stellungsrückmeldung Handschalter (A-H)
Instanz	Force	Fdb	FdbOut	LocSwi
Datenpunkt Typ	BO	AI	AI	BI
	8	4	2	1
BAC_AC_VAV_01_000	0	0	0	0
BAC_AC_VAV_01_003	0	0	1	1
BAC_AC_VAV_01_004	0	1	0	0
BAC_AC_VAV_01_007	0	1	1	1
BAC_AC_VAV_01_008	1	0	0	0
BAC_AC_VAV_01_011	1	0	1	1
BAC_AC_VAV_01_012	1	1	0	0
BAC_AC_VAV_01_015	1	1	1	1

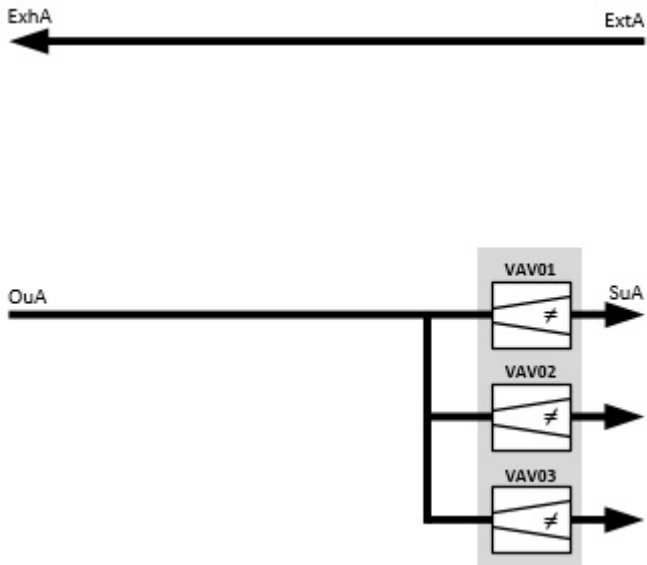
Schnittstelle



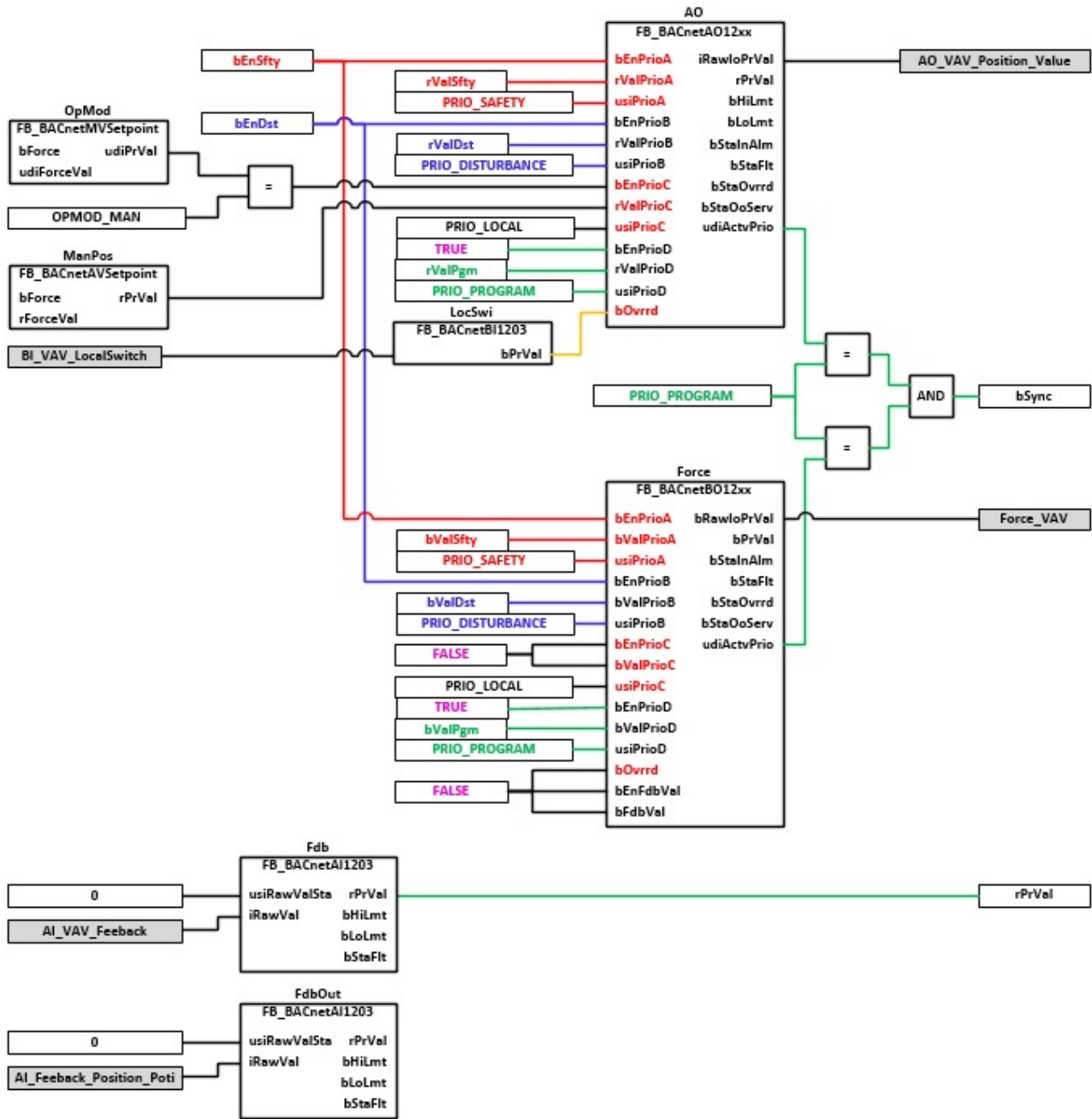
Anlagenschema 01



Anlagenschema 02



Blockschaltbild Variante BAC_AC_VAV_01_015



VAR_INPUT

bValPgm : BOOL;
rValPgm : REAL;
bEnSfty : BOOL;
bValSfty : BOOL;
rValSfty : REAL;
bEnDst : BOOL;
bValDst : BOOL;
rValDst : REAL;

bValPgm: Binärer Wert Programmpriorität

rValPgm: Analoger Wert Programmpriorität

bEnSfty: Freigabe Sicherheitspriorität

bValSfty: Binärer Wert Sicherheitspriorität

rValSfty: Analoger Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität. An diesen Eingang könnte z. B. eine Prozessrückmeldung angelegt werden.

bValDst: Binärer Wert Störungspriorität. An diesen Eingang könnte z. B. eine Prozessrückmeldung angelegt werden.

rValDst: Analoger Wert Störungspriorität

VAR_OUTPUT

```
bPrVal   : BOOL;
rPrVal   : REAL;
bSync    : BOOL;
```

bPrVal: Betriebsrückmeldung FU

rPrVal : aktuelle Drehzahl des FU.

bSync: Ausgabe eines Impulses um den zum FU zugehörigen Regler, beim Rücksetzen vom Hand- in den Automatikbetrieb auf die aktuelle Drehzahl des FU's zu synchronisieren.

VAR CONSTANT

```
PLT_NUM   : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optio- nal	Aufgabe			
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung des Volumenstromreglers von der MBE oder einem lokalen Bediendisplay			
ManPos	FB_BACnetAVSetpoint [► 70]		AV-Objekt zur Eingabe der Stellgröße des Volumenstromreglers bei manueller Übersteuerung			
Fdb	FB_BACnetAI1203 [► 49]	X	AI-Objekt zur Erfassung der Stellungsrückmeldung vom Volumenstromregler			
FdbOut	FB_BACnetAI1203 [► 49]	X	AI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Poti			
LocSwi	FB_BACnetBI1203 [► 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Rückmeldung Handschalter			
AO	FB_BACnetAO1203 [► 53]		AO-Objekt für die Ansteuerung des Volumenstromreglers			
			Priorität:	Freigabe	Wert	Bemerkung
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty	

Instanz	Typ	optional	Aufgabe			
			PRIO_DISTURBANANCE (3)	Eingang bEnDst	Eingang rValDst	
			PRIO_LOCAL (8)	OpMod_udiPrVal = OPMOD_MAN	ManPos_rPrVal	Bei Handbetrieb Wert von AV-Objekt ManPos
			PRIO_PROGRAM (15)	TRUE	Eingang rValPgm	Wert von Eingang rValPgm (z. B. von einem Regler)
BO	FB_BACnetBO1203 [▶ 82]	X	BO-Objekt für die Zwangssteuerung des Volumenstromreglers			
			Priorität:	Freigabe	Wert	
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty	
			PRIO_DISTURBANANCE (3)	Eingang bEnDst	Eingang bValDst	
			PRIO_LOCAL (8)			
			PRIO_PROGRAM (15)	TRUE	Eingang bValPgm	
	EQ, EQ, AND		Wert des Netzwerkes ist TRUE, wenn die Aktive Priorität PRIO_PROGRAM (15) ist. Kann zur Synchronisation des Reglers bei Rückkehr in den Automatikbetrieb benutzt werden			
TLogAO	FB_BACnetTLog1201 [▶ 137]		Zeichnet den Present Value des AO-Objekts auf			

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_VAV_LocalSwitch	BOOL	X	Eingang	Digitaleingang - Schalter Hand Volumenstromregler - Meldung - Hand/Auto
AI_Feedback_Position_Poti	INT	X	Eingang	Analogeingang - Hand Poti - Rueckmeldung - Stellwert
AI_VAV_Feedback	INT	X	Eingang	Analogeingang - Volumenstromregler - Messwert - Stellungsrueckmeldung
AO_VAV_Position_Value	INT		Ausgang	Analogausgang - Volumenstromregler - Stellwert Position
BO_Force_VAV	BOOL	X	Ausgang	Digitalausgang - Volumenstromregler Schaltbefehl - Zwangsuebersteuerung Ein/Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.48 BAC_AC_OpMod_01

Funktionsbeschreibung

Das Template **BAC_AC_OpMod_01** priorisiert verschiedene Ereignisse oder Befehle einer raumluftechnischen Anlage, wie zum Beispiel Brandalarm, Anforderung vom Zeitschaltplan oder die Anforderung vom Anlagenwahlschalter und schreibt eine resultierende Betriebsart bzw. einen resultierenden Anlagenstatus auf die Variable **udiOpMod**.

Innerhalb der Aggregate wird in Abhängigkeit von **udiOpMod** individuell reagiert, so dass sich die gesamte Anlage entsprechend der jeweils gültigen Betriebsart einstellt.

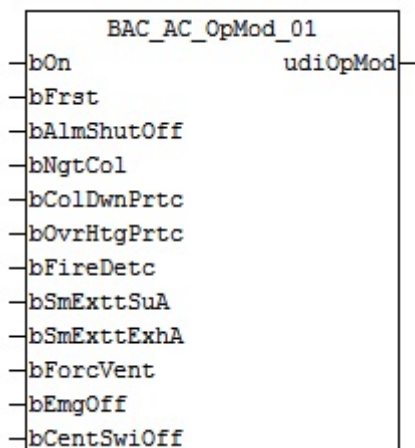
Beispiel: 1

Das Mischluftsystem **BAC_AC_MixAT_01** [► 483] fährt bei **udiOpMod** im Betriebsmodus Sommernachtkühlung (**udiOpMod** = 11) auf eine Außenluftfrate von 100%.

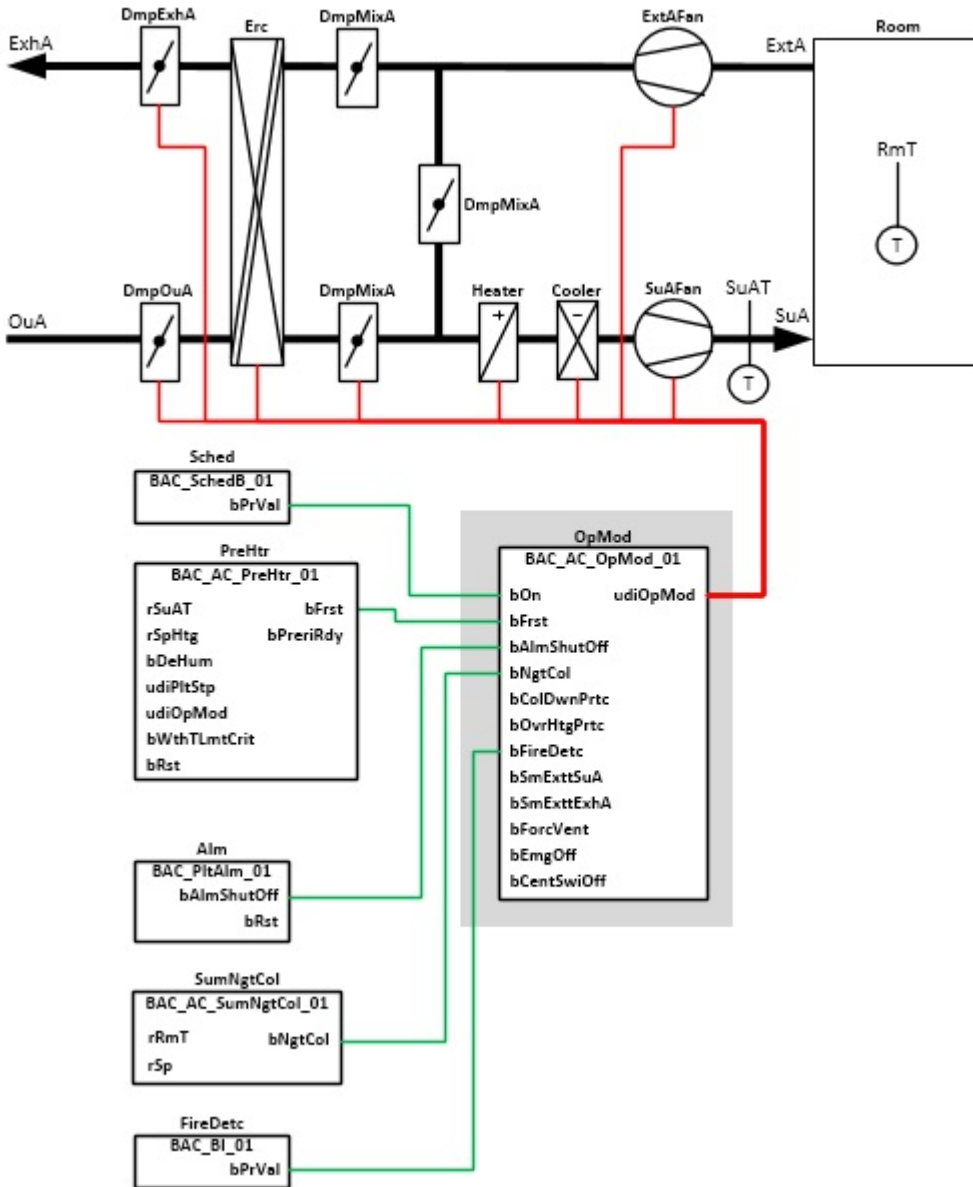
Beispiel: 2

Der Zuluft- und Abluftventilator schaltet bei einer Zwangsbelüftung ein. Geregelt Ventilatoren fahren auf eine Drehzahl von 100%.

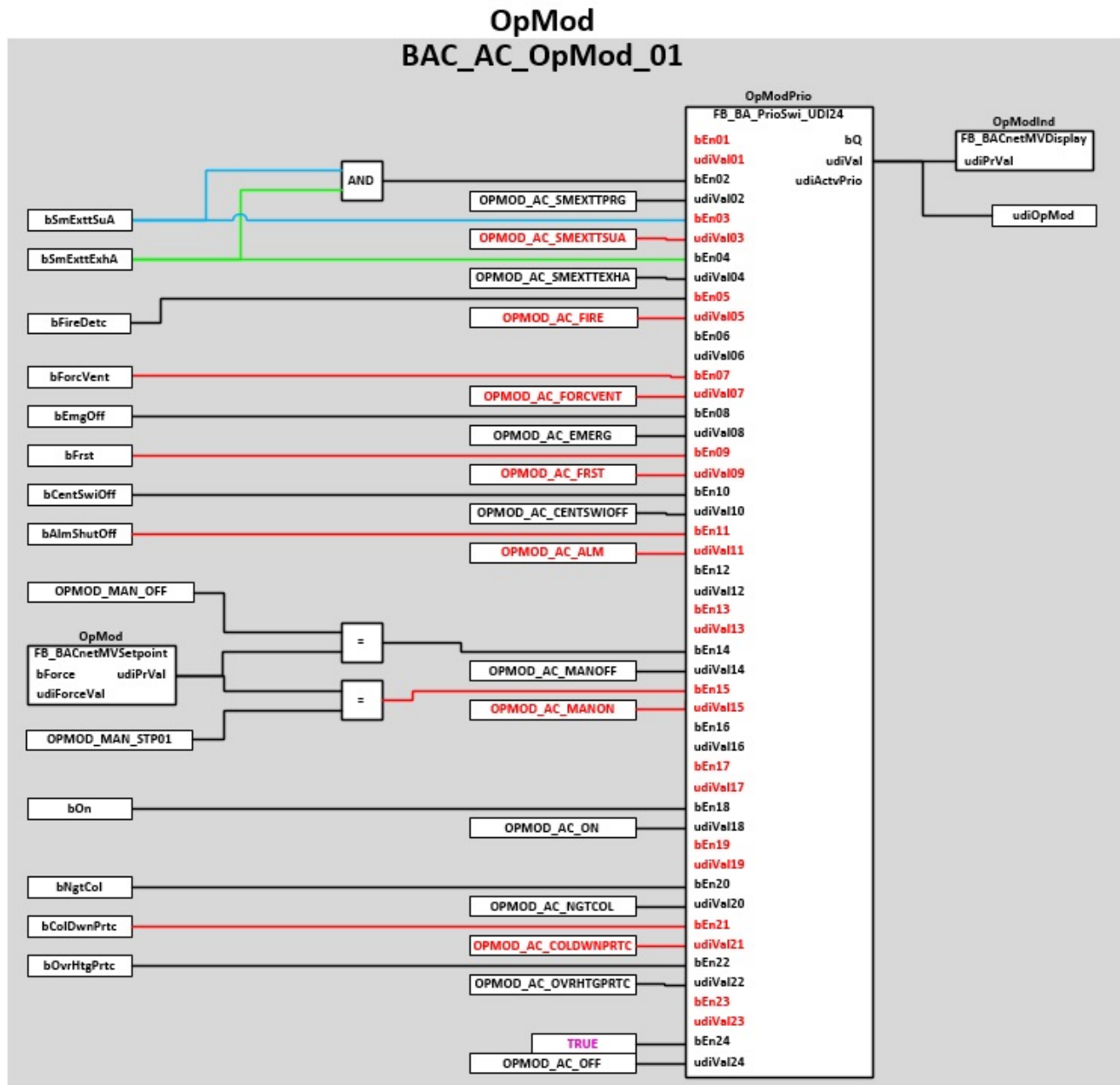
Schnittstelle



Anlagenschema



Blockdiagramm



Die folgenden Anlagenzustände bzw. Betriebsarten werden über die Variable beschrieben: *udiOpMod*

udiOpMod			
OPMOD AC OFF [▶ 364]	1	Aus//Off	Ausschaltung der Anlage
OPMOD AC ON [▶ 364]	2	Ein//On	Einschaltung der Anlage
OPMOD AC EMERG [▶ 364]	3	Notfall//Emergency	Ausschaltung der Anlage
OPMOD AC MANOFF [▶ 364]	4	Hand aus//Manual off	Ausschaltung der Anlage
OPMOD AC MANON [▶ 364]	5	Hand ein//Manual on	Einschaltung der Anlage
OPMOD AC FRST [▶ 364]	6	Frost//Frost	Ausschaltung der Anlage
OPMOD AC SMEXTTPRG [▶ 364]	7	Entrauchung Programm//Smoke extraction programm	Einschaltung der Anlage
OPMOD AC SMEXTTSUA [▶ 364]	8	Entrauchung Zuluft//Smoke extraction supply	Einschaltung Zuluft der Anlage
OPMOD AC SMEXTTEXHA [▶ 364]	9	Entrauchung Fortluft//Smoke extraction exhaust	Einschaltung Abluft der Anlage
OPMOD AC FIRE [▶ 364]	10	Feuer//Fire	Ausschaltung der Anlage

OPMOD_AC_NGTCOL [► 364]	11	Nachtkühlung//Night cooling	Einschaltung der Anlage
OPMOD_AC_COLDWNPRTC [► 364]	12	Stützbetrieb,Auskühlschutz//Cool down protection	Einschaltung der Anlage
OPMOD_AC_OVRHTGPRTC [► 364]	13	Überhitzungsschutz//Over heating protection	Einschaltung der Anlage
OPMOD_AC_ALM [► 364]	14	Störung//Alarm	Ausschaltung der Anlage
OPMOD_AC_FORCVENT [► 364]	15	Zwangsbelüftung//Forced ventilation	Einschaltung der Anlage
OPMOD_AC_CENTSWIOFF [► 364]	16	Zentralabschaltung//Central switch-off	Ausschaltung der Anlage

VAR_INPUT

```
bOn          : BOOL;
bFrst       : BOOL;
bAlmShutOff : BOOL;
bNgtCol     : BOOL;
bColDwnPrtc : BOOL;
bOvrHtgPrtc : BOOL;
bFireDetc   : BOOL;
bSmExttSuA  : BOOL;
bSmExttExuA : BOOL;
bForcVent   : BOOL;
bEmgOff     : BOOL;
bCentSwiOff : BOOL;
```

bOn: Anforderung vom Zeitschaltprogramm

bFrst: Frostschutzprogramm aktive

bAlmShutOff: Sammelstörmeldung - Anlage abschalten

bNgtCol: Anforderung vom Programm Sommernachtkühlung

bColDwnPrtc: Anforderung vom Programm Auskühlschutz

bOvrHtgPrtc: Anforderung vom Programm Überhitzungsschutz

bFireDetc: Meldung Feueralarm von Brandmeldezentrale

bSmExttSuA: Nachströmung vom Zuluftteil der Anlage bei Entrauchung angefordert.

bSmExttExhA: Entrauchung mit Abluftteil der Anlage Angefordert.

bForcVent: Anforderung Zwangsbelüftung

bEmgOff: Not-Aus

bCentSwiOff: Zentralabschaltung

VAR_OUTPUT

```
udiOpMod    : UDINT;
```

udiOpMod: Ausgabe der aktuellen Anlagenbetriebsart.

VAR CONSTANT

```
PLT_NUM     : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm. \[► 182\]](#)

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01 \[► 372\]](#)

mittels des Funktionsbausteins [FB_BA_AlarmPlt. \[► 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB_BA_ComnMsg \[► 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
OpMod	FB_BACnetMVSetpoint [► 131]	Das BACnet-MV-Objekt dient als Anlagenwahlschalter. Mit dem MV-Objekt kann die Anlage unabhängig von den Automatikprogrammen ein- und ausgeschaltet werden.
OpModPrio	FB_BA_PrioSwi_UDI24 [► 216]	Der Funktionsbaustein erfasst alle Befehle, priorisiert diese und gibt das Ergebnis am Ausgang udiVal aus.
OpModInd	FB_BACnetMVDisplay [► 130]	Das BACnet-MV-Objekt zeigt die aktuell gültige Anlagenbetriebsart an.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.49 BAC_AC_SeqH_01

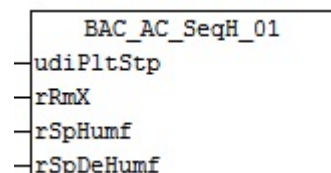
Funktionsbeschreibung

Das Template **BAC_AC_SeqH_01** ist zuständig für das Starten der Zuluftsequenzregelung Be- und Entfeuchtung einer RLT-Anlage.

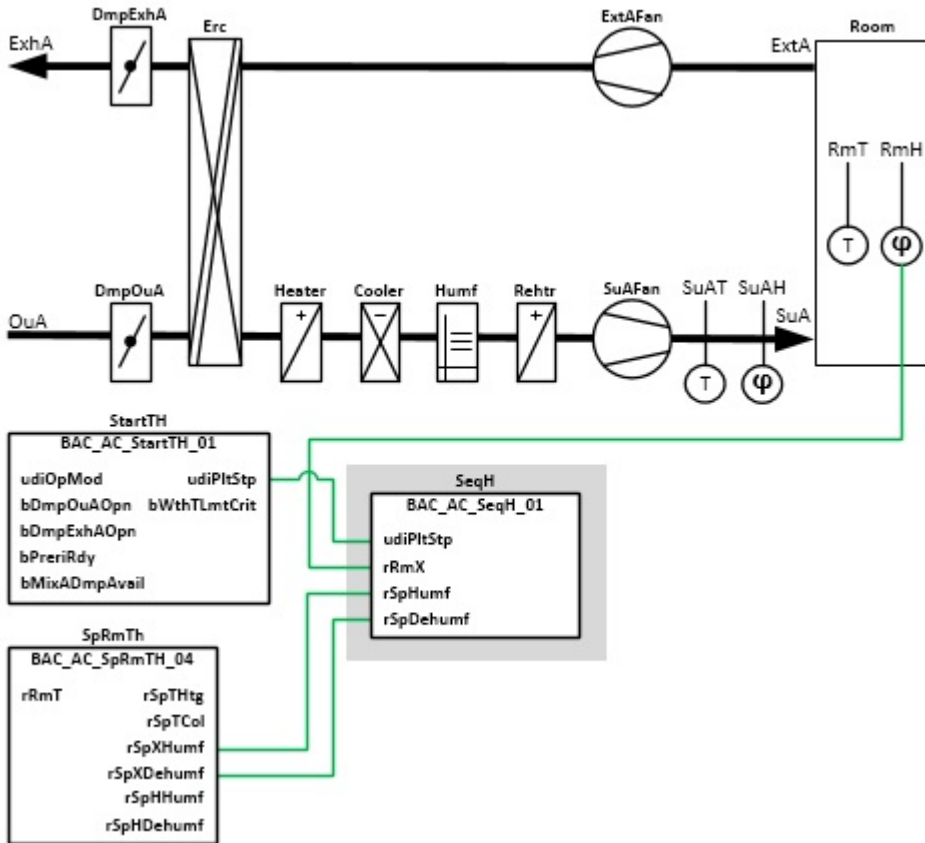
Beim Anfahren der RLT-Anlage wird bestimmt, ob sie mit der Be- oder Entfeuchte-Sequenz gestartet wird. Die Auswahl der Startsequenz erfolgt in Abhängigkeit der Anlagenschritte und der absoluten Raumfeuchte.

Nur ein Element der Sequenz kann regelnd sein. Wenn der Ausgang eines regelnden Sequenzelementes Y Min oder Y Max erreicht hat, wird die Regelung an den nächsten einschaltbereiten Sequenzregler übergeben.

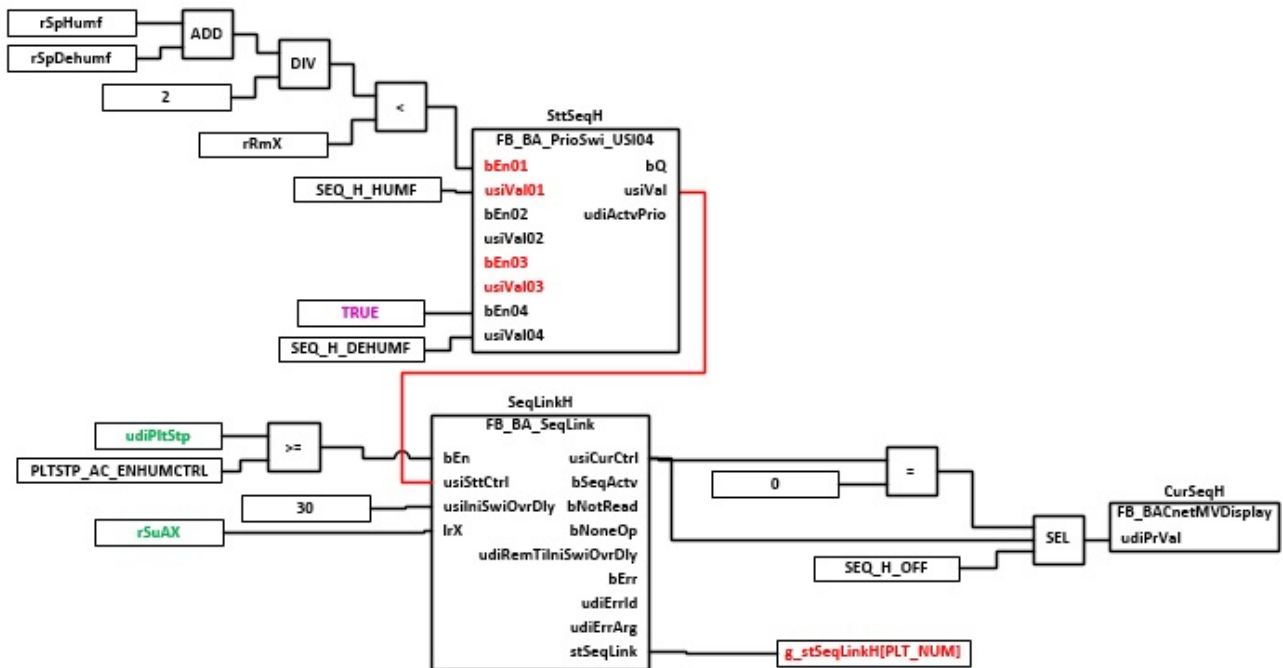
Schnittstelle



Anlagenschema

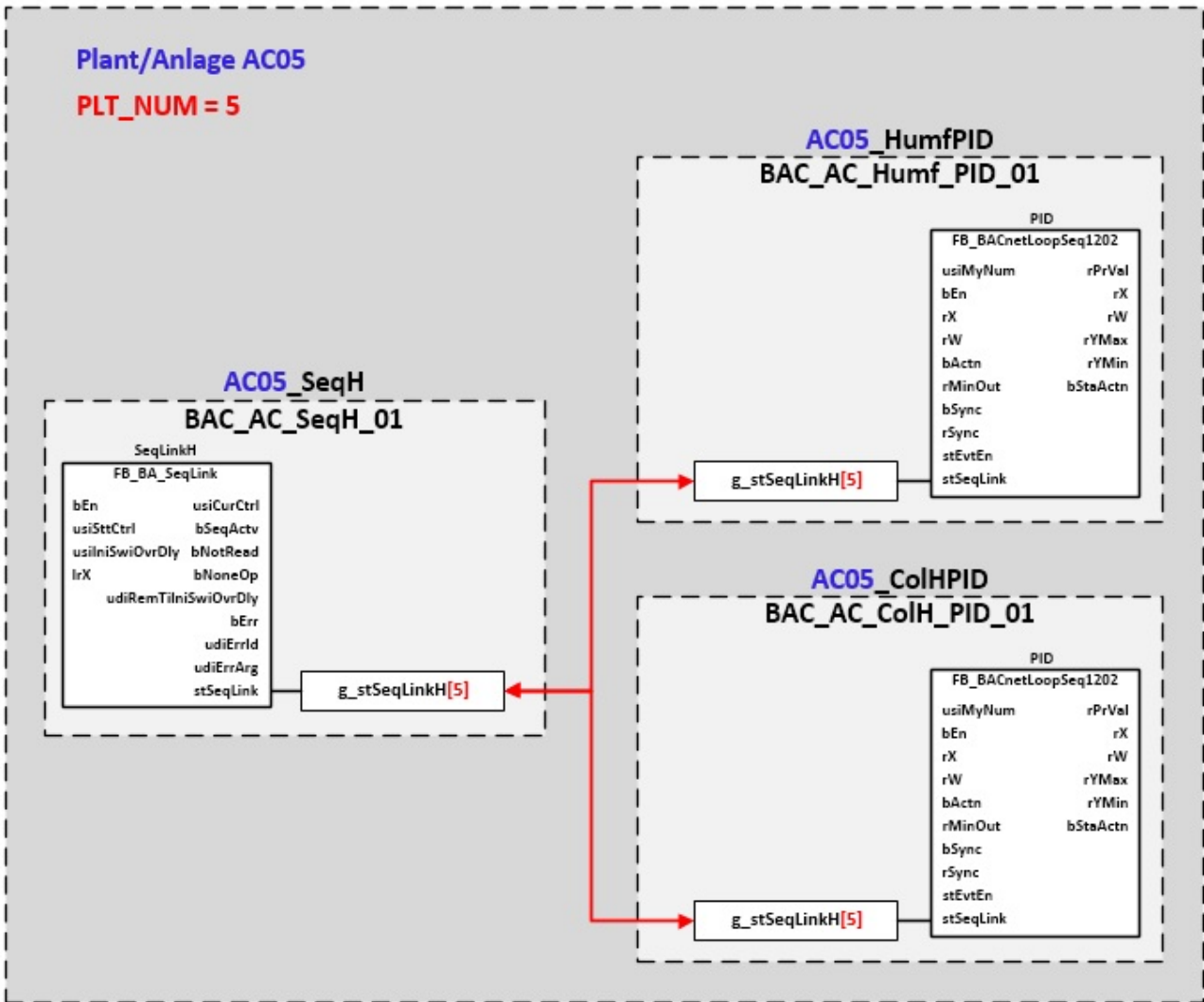


Blockschaltbild



Vernetzung der globalen Sequenz Link Struktur g_stSeqLinkH[PLT_NUM]

[g_stSeqLinkH\[PLT_NUM\] | 364](#)

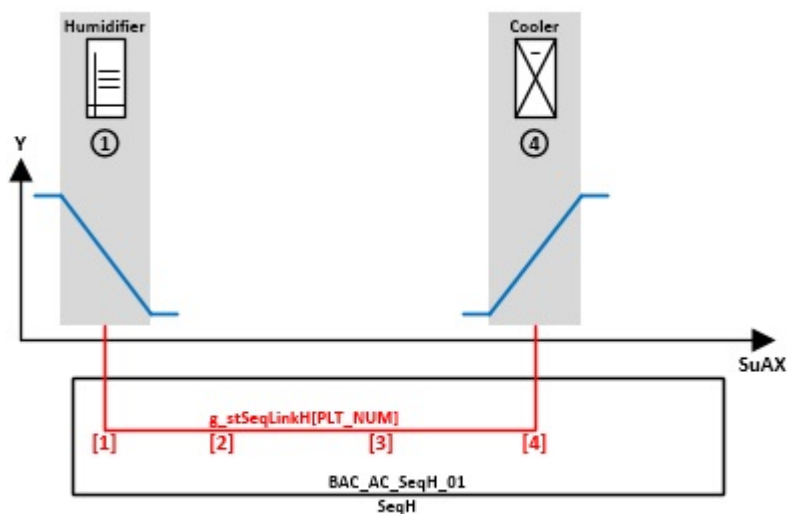


Reihenfolge der Sequenzen

Die Reihenfolge der Sequenzregler der einzelnen Aggregate muss mit der Sequenzreihenfolge übereinstimmen.

Die Belegung der Sequenznummern muss nicht durchgängig sein, es dürfen freie Anschlüsse dazwischen sein.

Z. B. 1 = Befeuchter, 4 = Kühler/Entfeuchter



Global definierte Sequenznummern der Sequenzregler:

SEQNUM_H_HUMF [▶ 364]	1	Sequenznummer Befeuchter//Sequence number humidifier
SEQNUM_H_MIX [▶ 364]	2	Sequenznummer Mischluft//Sequence number mixed air
SEQNUM_H_ERC [▶ 364]	3	Sequenznummer Energierückgewinnung//Sequence number energy recovery
SEQNUM_H_DEHUMF [▶ 364]	4	Sequenznummer Entfeuchter//Sequence number dehumidifier
SEQNUM_H_OFF [▶ 364]	5	kein Sequenzregler aktiv//no sequence controller active

VAR_INPUT

```

udiPltStp : UDINT;
rRmX      : REAL;
rSpHumf   : REAL;
rSpDehumf : REAL;
    
```

udiPltStp: Schritte beim Anfahren der RLT-Anlage. Siehe auch [BAC_AC_StartTH_01 \[▶ 543\]](#)

rRmX: Messwert Raumfeuchte (**absolute Feuchte**)

rSpHumf: Sollwert Befeuchten (**absolute Feuchte**)

rSpDehumf: Sollwert Entfeuchten (**absolute Feuchte**)

VAR CONSTANT

```

PLT_NUM : BYTE := 1;
    
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm \[▶ 182\]](#)

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01 \[▶ 372\]](#) mittels des Funktionsbausteins [FB_BA_AlarmPlt \[▶ 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg \[▶ 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur **g_stSeqLinkH[PLT_NUM]** dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein **FB_BA_SeqLink** zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
SttSeqH ADD, DIV, <	FB_BA_PrioSwi_USI0 4 [▶ 216]	Der Prioritätenschalter dient zur Auswahl der Startsequenz. Prio 1: bEN01 Ist der Mittelwert der Sollwerte rSpHumf / rSpDehumf kleiner als die absolute Raumfeuchte, so startet die Sequenz mit Befeuchten. (SEQNUM_H_HUMF [▶ 364]) gestartet. Prio 4: bEN04 Tritt der Fall Prio 1 (SEQNUM_H_HUMF [▶ 364]) nicht ein, so wird mit der Sequenz Entfeuchten (Kühler) gestartet (SEQNUM_H_Dehumf [▶ 364])
SeqLinkH	FB_BA_SeqLink [▶ 171]	Der Funktionsbaustein SeqLinkH ist das Kernstück des Templates BAC_SeqH_01 . Der Sequenzlinker ist über die globale Datenstruktur g_stSeqLinkH[PLT_NUM] mit allen Zuluftreglern der

Instanz	Typ	Aufgabe
		Sequenz verbunden. Er ist das zentrale Steuerorgan und für das Umschalten zwischen den Sequenzreglern und das Starten der Regelsequenz. verantwortlich. Die Freigabe des Sequenzlinkers am Eingang <i>bEn</i> erfolgt über die Schritte beim Anfahren der Lüftungsanlage, wenn udiPltStp >= der globalen Konstanten <code>PLTSTP_AC_ENHUMCTRL [▶ 364]</code> ist.
CurSeqH	FB_BACnetMVDisplay [▶ 130]	Das MV-Objekt zeigt den aktuell aktiven Sequenzregler an.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.50 BAC_AC_SeqT_01

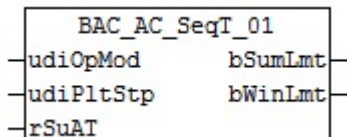
Funktionsbeschreibung

Das Template **BAC_AC_SeqT_01** ist zuständig für das Starten der Zulufttemperatur-Sequenzregelung einer RLT-Anlage.

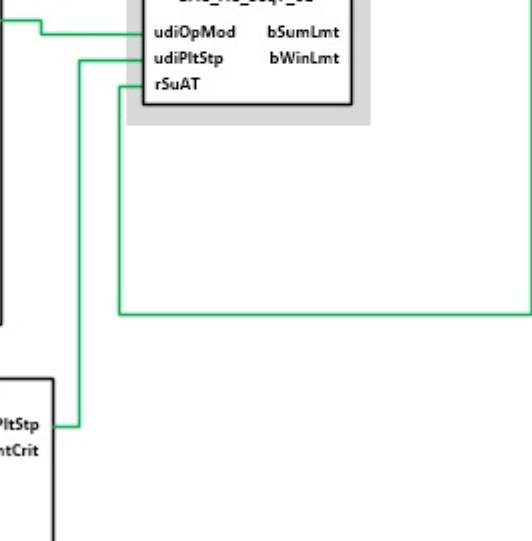
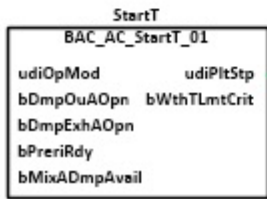
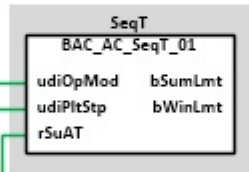
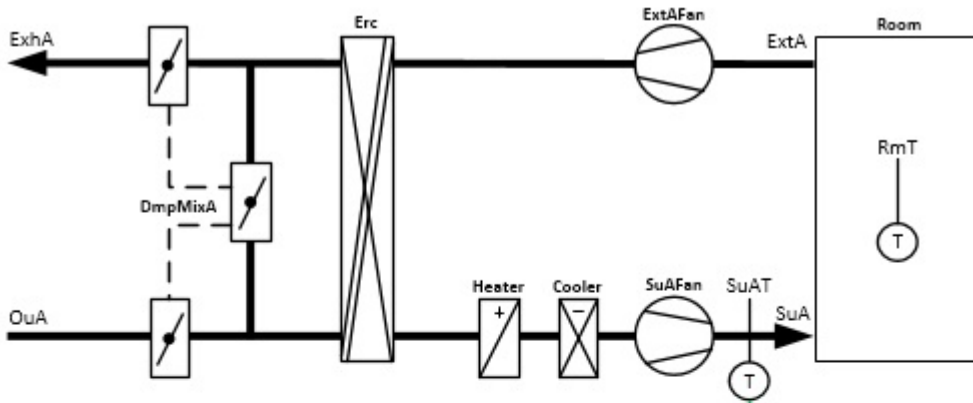
Beim Anfahren der RLT-Anlage wird bestimmt, ob sie mit der Heiz-, Kühl- oder WRG-Sequenz gestartet wird. Die Auswahl der Startsequenz erfolgt in Abhängigkeit der Anlagenbetriebsart und der Außentemperatur.

Nur ein Element der Sequenz kann regelnd sein. Wenn der Ausgang eines regelnden Sequenzelementes Y Min oder Y Max erreicht hat, wird die Regelung an den nächsten einschaltbereiten Sequenzregler übergeben.

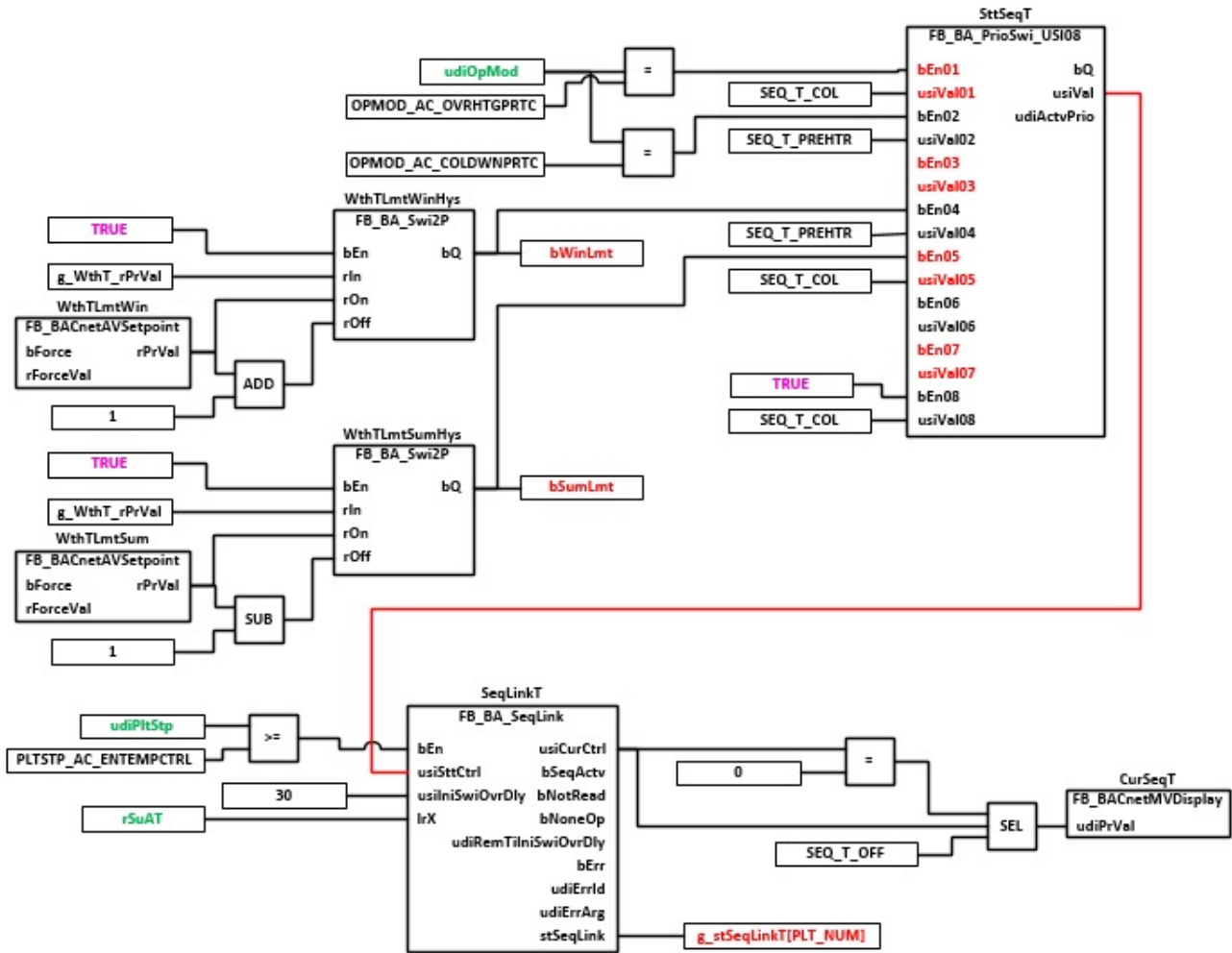
Schnittstelle



Anlagenschema

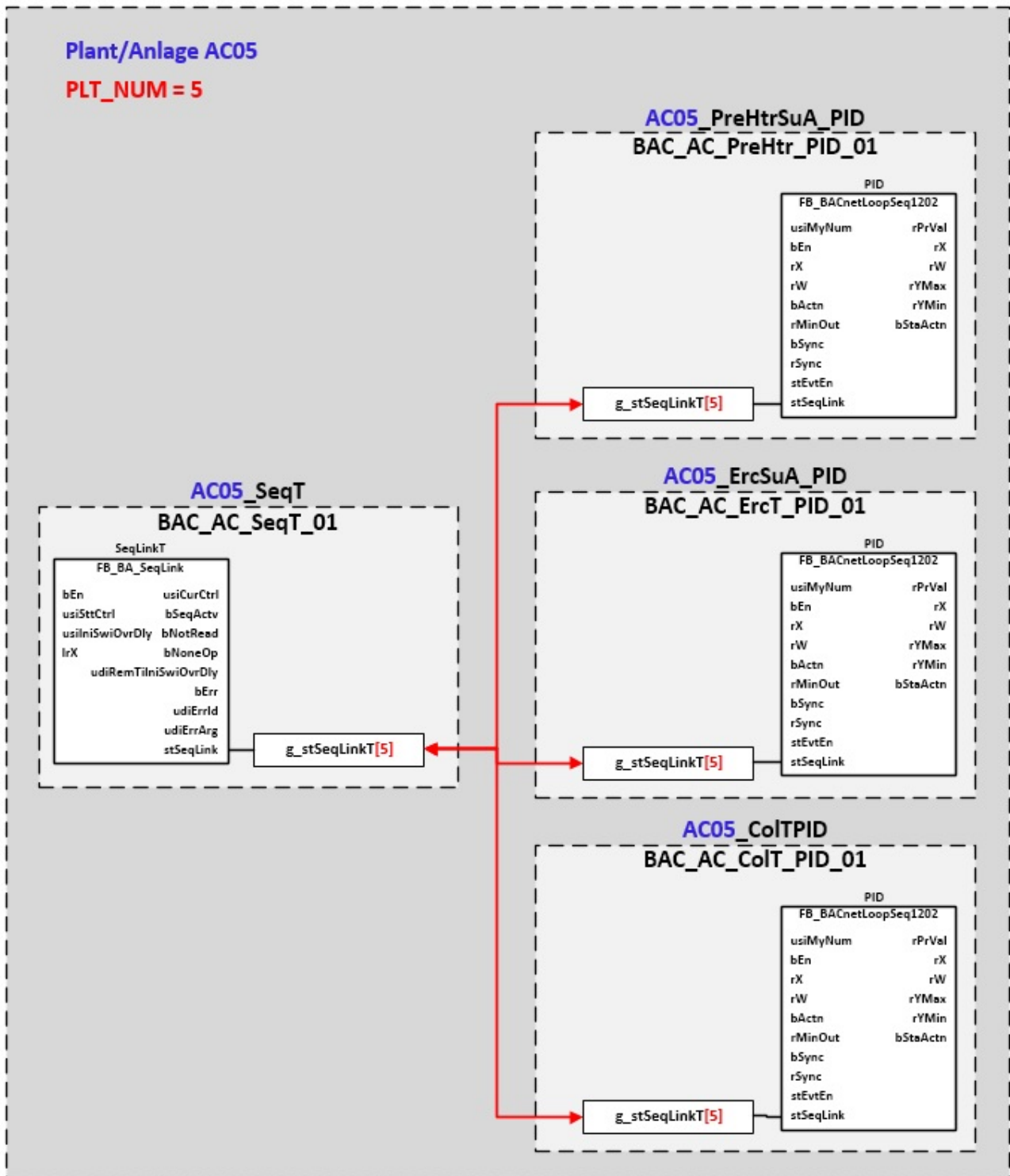


Blockschaltbild



Vernetzung der globalen Sequenz Link Struktur g_stSeqLinkT[PLT_NUM]

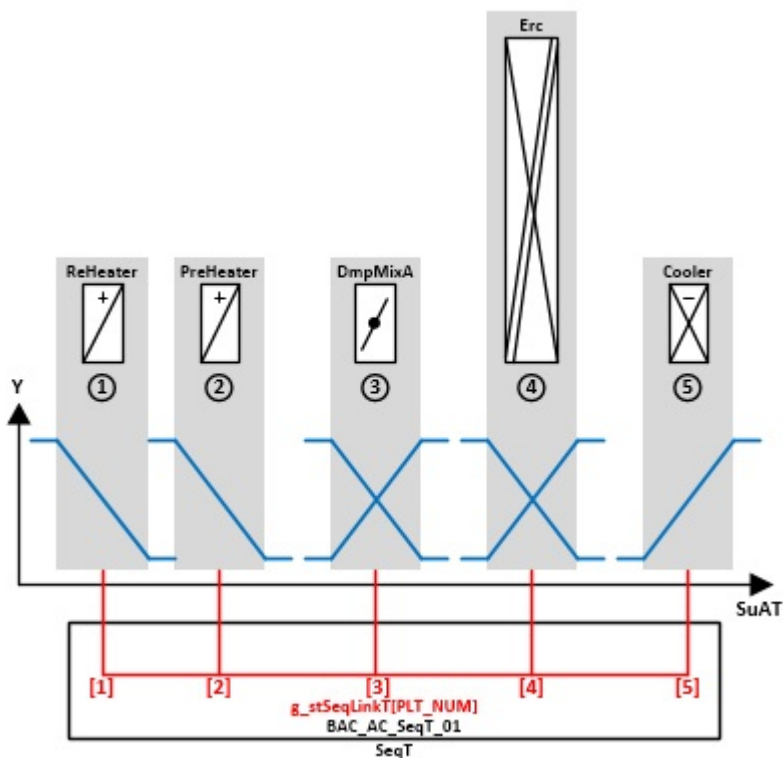
g_stSeqLinkT[PLT_NUM] [▶ 364]



Reihenfolge der Sequenzen

Die Reihenfolge der Regelsequenzen ist durch die global definierten Variablen vorgegeben. Im Bedarfsfall ist die Reihenfolge der Sequenzen leicht anzupassen.

Zum Beispiel dann wenn die Energierückgewinnung vor dem Mischluftsystem liegen sollte.



Global definierte Sequenznummern der Sequenzregler:

SEQNUM T REHTR [► 364]	1	Sequenznummer Nacherhitzer//Sequence number reheater
SEQNUM T PREHTR [► 364]	2	Sequenznummer Vorerhitzer//Sequence number preheater
SEQNUM T MIX [► 364]	3	Sequenznummer Mischluft//Sequence number mixed air
SEQNUM T ERC [► 364]	4	Sequenznummer Energierückgewinnung//Sequence number energy recovery
SEQNUM T COL [► 364]	5	Sequenznummer Kühler//Sequence number cooler
SEQNUM T OFF [► 364]	6	kein Sequenzregler aktiv//no sequence controller active

VAR_INPUT

```
udiOpMod   : UDINT;
udiPltStp  : UDINT;
rSuAT      : REAL;
```

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC_AC_OpMod_01 \[► 524\]](#)

udiPltStp: Schritte beim Anfahren der RLT-Anlage. Siehe auch [BAC_AC_StartT_01 \[► 538\]](#)

rSuAT: Messwert Zulufttemperatur

VAR_OUTPUT

```
bSumLmt    : BOOL;
bWinLmt    : BOOL;
```

bSumLmt: Sommerbetrieb

bWinLmt: Winterbetrieb

VAR CONSTANT

```
PLT_NUM    : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt. Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der

Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [[▶ 182](#)]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [[▶ 372](#)] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Innerhalb einer Lüftungsanlage mit Sequenzregler gibt die Anlagennummer vor welches Feld aus der globalen Datenstruktur [g_stSeqLinkT\[PLT_NUM\]](#) dazu dient, dass Bindeglied zwischen den einzelnen Sequenzreglern und dem dazu gehörigen Steuerbaustein [FB_BA_SeqLink](#) zu sein.

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
WthTLmtSum	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes ab welcher Außentemperatur die RLT-Anlage in der Kühlsequenz startet.
WthTLmtWin	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes ab welcher Außentemperatur die RLT-Anlage in der Heizsequenz startet.
SttSeqT	FB_BA_PrioSwi_USI08 [▶ 216]	<p>Der Prioritätenschalter dient zur Auswahl der Startsequenz.</p> <p>Prio 1: bEN01 Bei der Betriebsart Überhitzungsschutz (OPMOD_AC_OVRHTGPRTC) wird grundsätzlich mit der Sequenz Kühlen (SEQNUM_T_COL) gestartet.</p> <p>Prio 2: bEN02 Bei der Betriebsart Auskühlschutz (OPMOD_AC_COLDWNPRTC) wird grundsätzlich mit der Sequenz Heizen (SEQNUM_T_PREHTR) gestartet.</p> <p>Prio 4: bEN04 Der vorgelagerte Funktionsbaustein WthTLmtWinHys prüft ob die tatsächliche Außentemperatur unterhalb des kritischen Wertes von AV-Objekt <i>WthTLmtSum</i> liegt. Ist dieses der Fall dann startet die RLT-Anlage mit der Regelsequenz des Vorerhitzers. (SEQNUM_T_PREHTR)</p> <p>Prio 5: bEN05 Der vorgelagerte Funktionsbaustein WthTLmtSumHys prüft ob die tatsächliche Außentemperatur oberhalb des Wertes von AV-Objekt <i>WthTLmtSum</i> liegt. Ist dieses der Fall dann startet die RLT-Anlage mit der Regelsequenz des Kühlers. (SEQNUM_T_COL)</p> <p>Prio 8: bEN08 In den Übergangszeiten zwischen Winter und Sommer sind die Eingänge von bEn04 und bEn05 witterungsbedingt FALSE. In diesem Fall gilt am Prioritätenschalter die Priorität 8. Damit wird die RLT-Anlage in der Sequenz der Energierückgewinnung (SEQNUM_T_ERC) gestartet.</p>
SeqLinkT	FB_BA_SeqLink [▶ 171]	Der Funktionsbaustein SeqLinkT ist das Kernstück des Templates BAC_AC_SeqT_01 . Der Sequenzlinker ist über die globale Datenstruktur g_stSeqLinkT[PLT_NUM] mit allen Zuluftreglern der Sequenz verbunden. Er ist das zentrale Steuerorgan und für das Umschalten zwischen den Sequenzreglern und das Starten der Regelsequenz. verantwortlich.

Instanz	Typ	Aufgabe
		Die Freigabe des Sequenzlinkers am Eingang <i>bEn</i> erfolgt über die Schritte beim Anfahren der Lüftungsanlage, wenn udiPltStp \geq der globalen Konstanten PLTSTP_AC_ENTEMPCTRL ist.
CurSeqT	FB_BACnetMVDisplay [▶_130]	Das MV-Objekt zeigt den aktuell aktiven Sequenzregler an.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

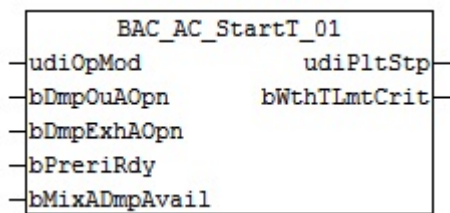
9.51 BAC_AC_StartT_01

Funktionsbeschreibung

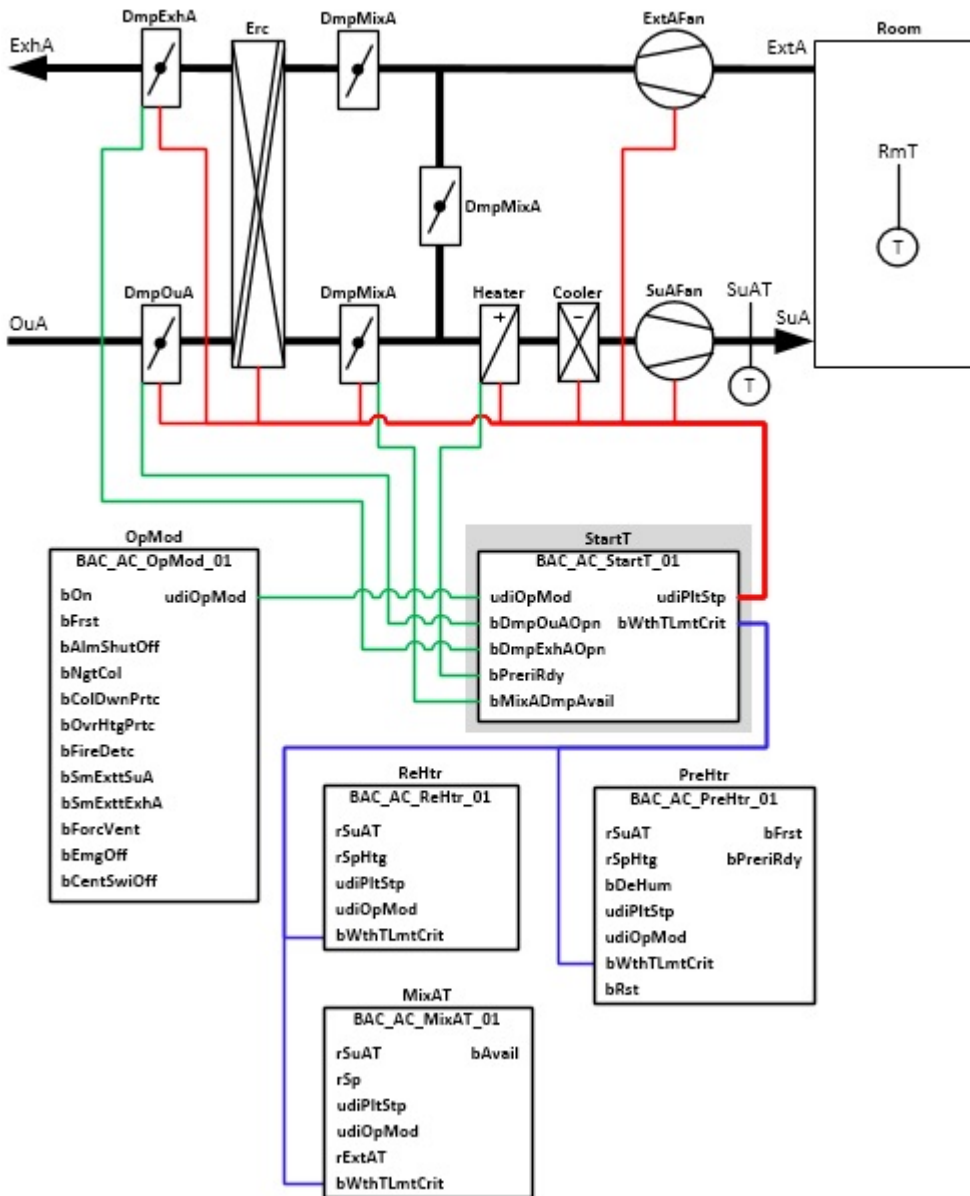
Das Template BAC_AC_StartT_01 ist zuständig für das schrittweise Anfahren eine RLT-Anlage ohne Be- und Entfeuchtung.

Die Schritte beim Anlagenstart werden durch die Variable **udiPltStp** beschrieben. Die Schritte werden an alle Aggregate der Anlage übertragen, so dass jedes Aggregat in den Anfahrbetrieb der RLT-Anlage eingebunden ist.

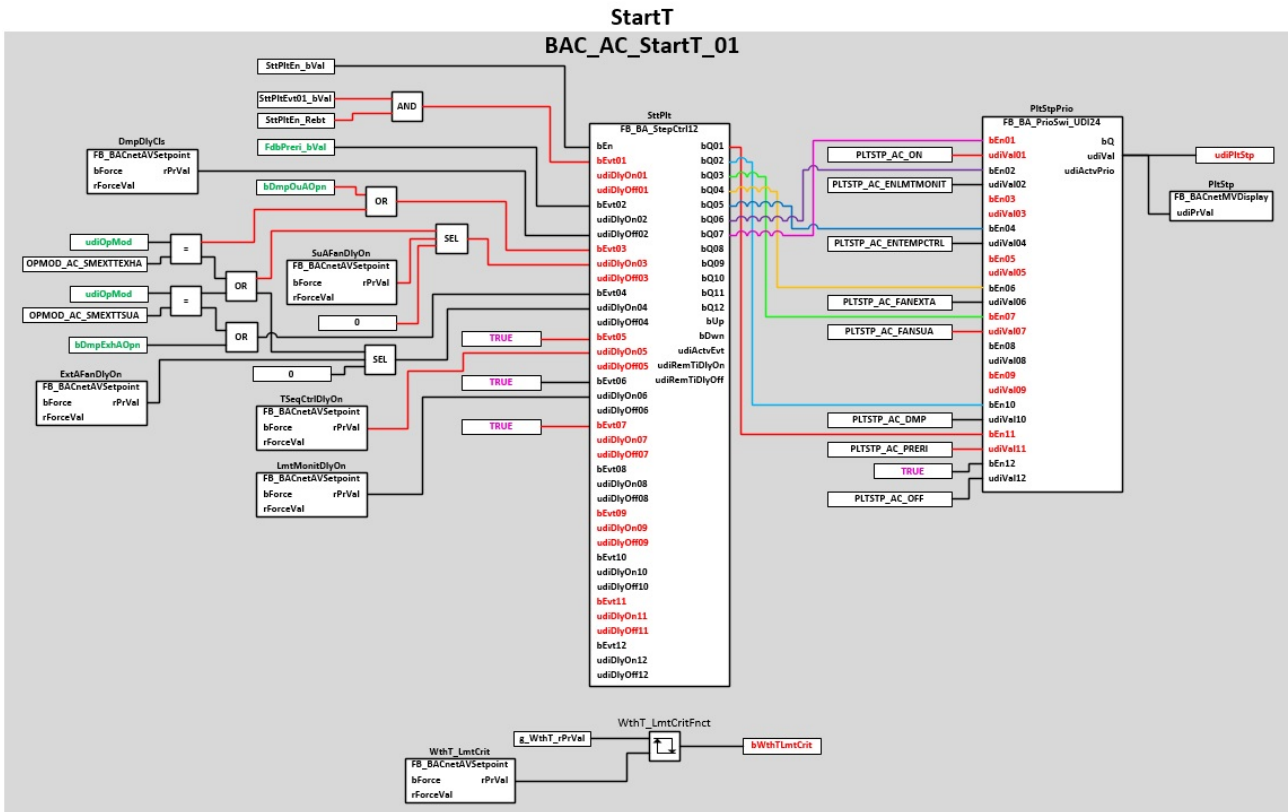
Schnittstelle



Anlagenschema



Blockdiagramm



Die folgenden Anlagenschritte werden über die Variable udiPltStp beschrieben:

udiPltStp	
PLTSTP_AC OFF [▶ 364]	Aus//Off
PLTSTP_AC PRERI [▶ 364]	Vorspülen//pre-rinse
PLTSTP_AC DMP [▶ 364]	Klappe//Damper
PLTSTP_AC FANSUA [▶ 364]	Zuluftventilator//Fan supply air
PLTSTP_AC FANEXTA [▶ 364]	Abluftventilator//Fan extract air
PLTSTP_AC ENTEMPCTRL [▶ 364]	Freigabe Temperaturregelung, Sequenz Link//Enable temperature control, Sequence link
PLTSTP_AC ENLMTMONIT [▶ 364]	Freigabe Grenzwertüberwachung//Enable limit monitoring
PLTSTP_AC ON [▶ 364]	Ein//On

VAR_INPUT

```

udiOpMod      : UDINT;
bExhADmpOpn  : BOOL;
bOuADmpOpn   : BOOL;
bPreriRdy    : BOOL;
bMixADmpAvail : BOOL;
    
```

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod 01 \[▶ 524\]](#)

bOuADmpOpn: Rückmeldung Außenluftklappe auf

bExhADmpOpn: Rückmeldung Fortluftklappe auf

bPreriRdy: Vorspülen am Heizregister abgeschlossen

bMixADmpAvail: Mischluftsystem für Anfahrprozess vorhanden

VAR_OUTPUT

```
udiPltStp      : UDINT;
bWhtTLmtCrit  : BOOL;
```

udiPltStp: Ausgabe der aktuellen Anlagen-Anfahrsschritte

bWhtTLmtCrit: Die Außentemperatur ist unterhalb eines kritischen Wertes, so dass beim Anfahren der RLT-Anlage mit dem Vorspülen des Heizregisters angefahren werden muss. Die Pumpen des Vor- und Nacherhitzer werden eingeschaltet. Bei den Mischluftklappen wird beim Anlagenstart die Rampenfunktion aktiviert.

VAR CONSTANT

```
PLT_NUM      : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm](#). [► 182]

Die Auswertung der Alarme einer Anlage, z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB BA AlarmPlt](#). [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
DmpDlyCls	FB BACnetAVSetpoint [► 70]	AV-Objekt verzögert das Schließen der Klappen beim Herunterfahren der RLT-Anlage. Damit wird die Nachlaufzeit der Ventilatoren beim Abschalten der Anlage berücksichtigt.
SuAFanDlyOn	FB BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Wertes für die Einschaltverzögerung des Zuluftventilators beim Einschalten der RLT-Anlage
ExtAFanDlyOn	FB BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Wertes für die Einschaltverzögerung des Abluftventilators beim Einschalten der RLT-Anlage
TSeqCtrlDlyOn	FB BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Werts für die zeitverzögerte Freigabe der Regelung
LmtMonitDlyOn	FB BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Werts für die zeitverzögerte Freigabe der Grenzwertüberwachung der Temperaturfühler
WthT_LmtCrit	FB BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Außentemperaturwertes ab wann die RLT-Anlage mit dem Vorspülen des Heizregisters starten soll.
SttPlt	FB BA StepCtrl12 [► 252]	<p>Der Funktionsbaustein SttPlt ist das Kernstück des Anlagenstartprogramms.</p> <p>Am Eingang bEn erfolgt eine generelle Freigabe des Funktionsbausteins. Ist bEn = FALSE, so sind Zwangsweise alle Ausgänge des Funktionsbausteins FALSE.</p> <p>Schritt 1: Vorspülen des Heizregisters PLTSTP_AC_PRERI Am Eingang bEvt01 wird das Anfahren der RLT zunächst mit dem Vorspülen des Heizregisters gestartet.</p> <p>Schritt 2: Öffnen der Außen- und Fortluftklappe PLTSTP_AC_DMP Wenn der Vorspülprozess abgeschlossen oder aufgrund hoher</p>

Instanz	Typ	Aufgabe
		<p>Außentemperaturen nicht erforderlich ist, wird der Eingang bEvt02 gesetzt. Dadurch wird der nächste Anfahrschritt, das Öffnen der Klappen angetriggert. Die Bedingung zum Weiterschalten in den Schritt Klappen Öffnen wird vom Funktionsbaustein FdbPreri gebildet.</p> <p>Schritt 3: Starten des Zuluftventilators PLTSTP_AC_FANSUA Am Eingang bEvt03 wird der Endlagenschalter der Außenluftklappe angeschlossen. Bei offener Klappe wird Schritt 3 das Starten des Zuluftventilators angetriggert. Das Starten des Zuluftventilators kann zusätzlich am Eingang udiDlyOn03 verzögert werden. In der Betriebsart Entrauchen wird der Wert für die Anfahrverzögerung der Ventilatoren durch den vorgelagerten Selector auf den Wert null gesetzt. Damit springt das Programm unverzögert in den nächsten Schritt.</p> <p>Schritt 4: Starten des Abluftventilators PLTSTP_AC_FANEXTA Am Eingang bEvt04 wird der Endlagenschalter der Fortluftklappe angeschlossen. Bei offener Klappe wird Schritt 4 das Starten des Fortluftventilators angetriggert. Das Starten des Abluftventilators kann zusätzlich am Eingang udiDlyOn04 verzögert werden. In der Betriebsart Entrauchen wird der Wert für die Anfahrverzögerung der Ventilatoren durch den vorgelagerten Selector auf den Wert null gesetzt. Damit springt das Programm unverzögert in den nächsten Schritt.</p> <p>Schritt 5: Freigabe der Zuluftsequenzregelung Temperatur PLTSTP_AC_ENTEMPCTRL Der Eingang bEvt05 ist konstant TRUE. Damit wird die Freigabe der Zuluftsequenzregelung nur durch den Wert von udiDlyOn05 verzögert.</p> <p>Schritt 6: Freigabe der Fühlergrenzwertüberwachung PLTSTP_AC_ENLMTMONIT Der Eingang bEvt06 ist konstant TRUE. Damit wird die Freigabe der Zuluftsequenzregelung nur durch den Wert von udiDlyOn06 verzögert.</p> <p>Schritt 7: Zeigt an, dass die Lüftungsanlage in vollem Betrieb ist PLTSTP_AC_ON Der Eingang bEvt07 ist konstant TRUE. Damit wird die Freigabe der Zuluftsequenzregelung nur durch den Wert von udiDlyOn06 verzögert.</p> <p>Beim Herunterfahren der RLT-Anlage läuft die Schrittkette unverzögert von Schritt 7 bis Schritt 3 herunter. Das Zufahren der Klappen nach dem Abschalten der Ventilatoren ist um den Timer DmpDlyCIs verzögert. Schritt 1 das Vorspülen des Heizregisters wird übersprungen.</p>
SttPltEn	FB_BA_MMUX_B04 ▶ 213	Der Multiplexer SttPltEn erzeugt die generelle Freigabe des Anfahrprogrammes in Abhängigkeit der Anlagenbetriebsarten. Ändert sich die Anlagenbetriebsart während der laufenden Betriebs dann wird die Variable SttPltEn_Rebt am Ausgang des Vergleiches = TRUE. Damit wird der Anlagenstartbaustein SttPlt kurzzeitig zurückgesetzt. Das Startprogramm wird damit in einen definierten Zustand versetzt und entsprechend der neuen Betriebsart neu gestartet.

Instanz	Typ	Aufgabe
SttPltEvt01	FB BA MMUX B24 [▶ 208]	Der Multiplexer SttPltEvt01 erzeugt die Freigabe zum Starten des Vorspülprozesses in Abhängigkeit der Anlagenbetriebsarten
FdbPreri	FB BA MMUX B08 [▶ 208]	Der Multiplexer FdbPreri erzeugt die Freigabe für das Öffnen der Klappen. In den Betriebsarten OPMOD_AC_SMEXTTPRG , OPMOD_AC_SMEXTTSUA , OPMOD_AC_SMEXTTEXHA , OPMOD_AC_NGTCOL und OPMOD_AC_OVRHTGPRTC wird am Ausgang bVal ein TRUE ausgegeben. Damit wird der Schritt 1 des Anlagenstartbausteins SttPlt bei diesen Betriebsarten übersprungen. Der Vorspülprozess wird als abgeschlossen gemeldet. Für den normalen Anlagenbetrieb wird das Öffnen der Klappen unter den folgenden Bedingungen angetriggert: 1. bMixADmpAvail = TRUE (Es ist ein Mischluftsystem vorhanden. Der Anfahrbetrieb wird mittels Mischluftsystem realisiert) 2. bPreriRdy = TRUE (Der Vorspülprozess ist abgeschlossen) 3. Die Außentemperatur ist unkritisch. Der Ausgang des Funktionsbaustein WthT_LmtCritFnc ist FALSE.
WthT_LmtCritFnc	FB BA Swi2P [▶ 148]	Der Funktionsbaustein WthT_LmtCritFnc ist ein Zweipunktschalter und setzt die Ausgangsvariable bWthTLmtCrit in Abhängigkeit der Außentemperatur auf TRUE oder FALSE. Der Schalterpunkt für den Zweipunktschalter wird über das AV-Objekt WthT_LmtCrit vorgegeben..
PltStpPrio	FB BA PrioSwi UDI12 [▶ 216]	Der Prioritätenschalter PltStpPrio wertet die aktuellen Schritte während des Anfahrprozesses aus und gibt sie mittels der Variable udiVal an das BACnet-MV-Objekt PltStp weiter zur Anzeige in der Managementebene.
PltStp	FB BACnetMVDisplay [▶ 130]	Das BACnet-MV-Objekt zeigt den aktuellen Schritt während des Anfahrprozesses an.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

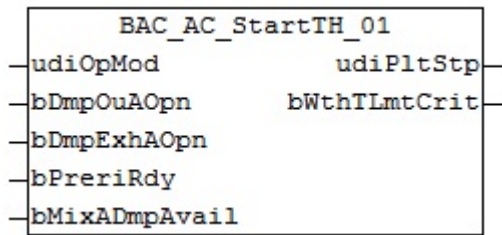
9.52 BAC_AC_StartTH_01

Funktionsbeschreibung

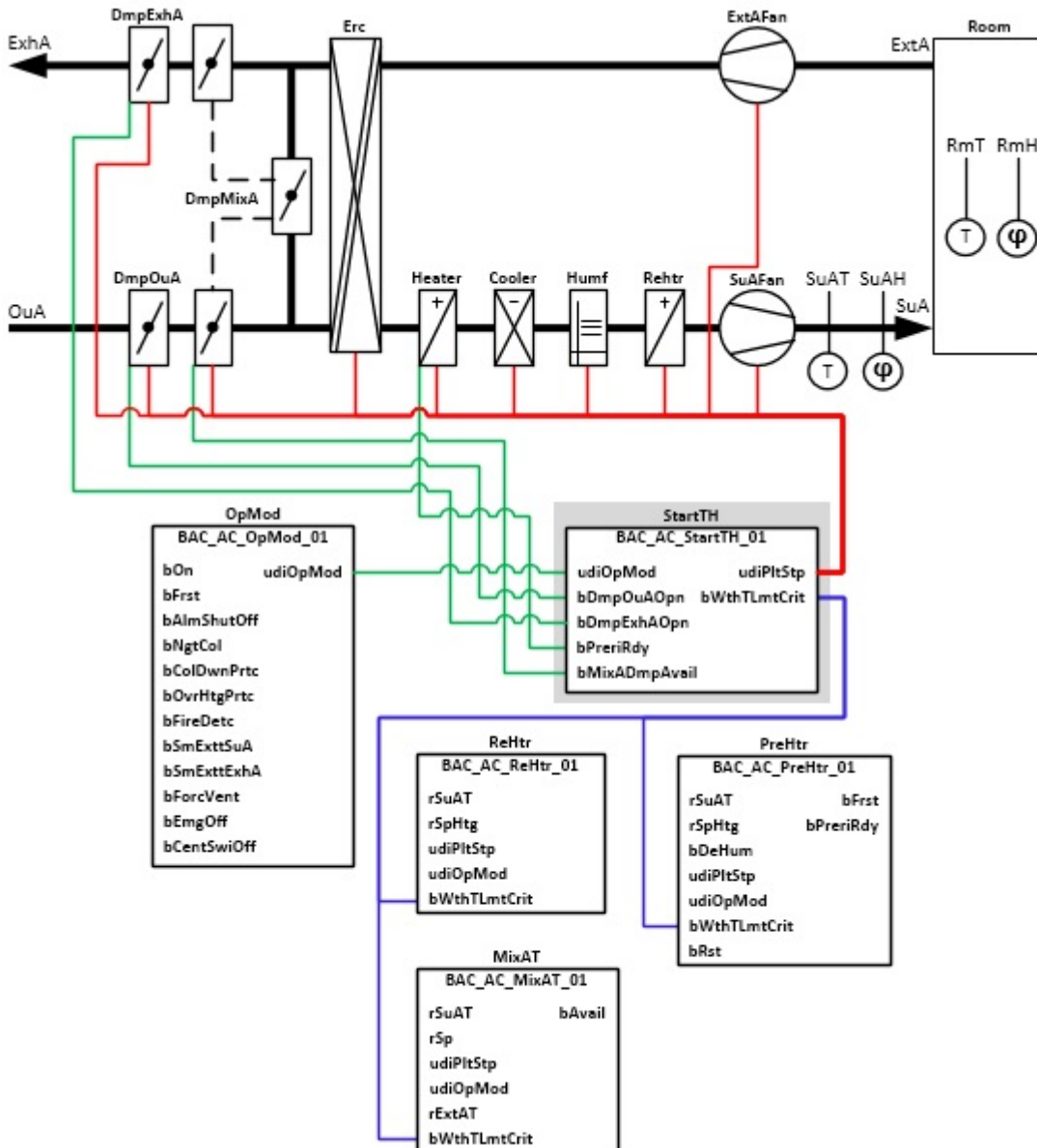
Das Template BAC_AC_StartTH_01 ist zuständig für das schrittweise Anfahren einer RLT-Anlage mit Be- und Entfeuchtung.

Die Schritte beim Anlagenstart werden durch die Variable **udiPltStp** beschrieben. Die Schritte werden an alle Aggregate der Anlage übertragen, so dass jedes Aggregat in den Anfahrbetrieb der RLT-Anlage eingebunden ist.

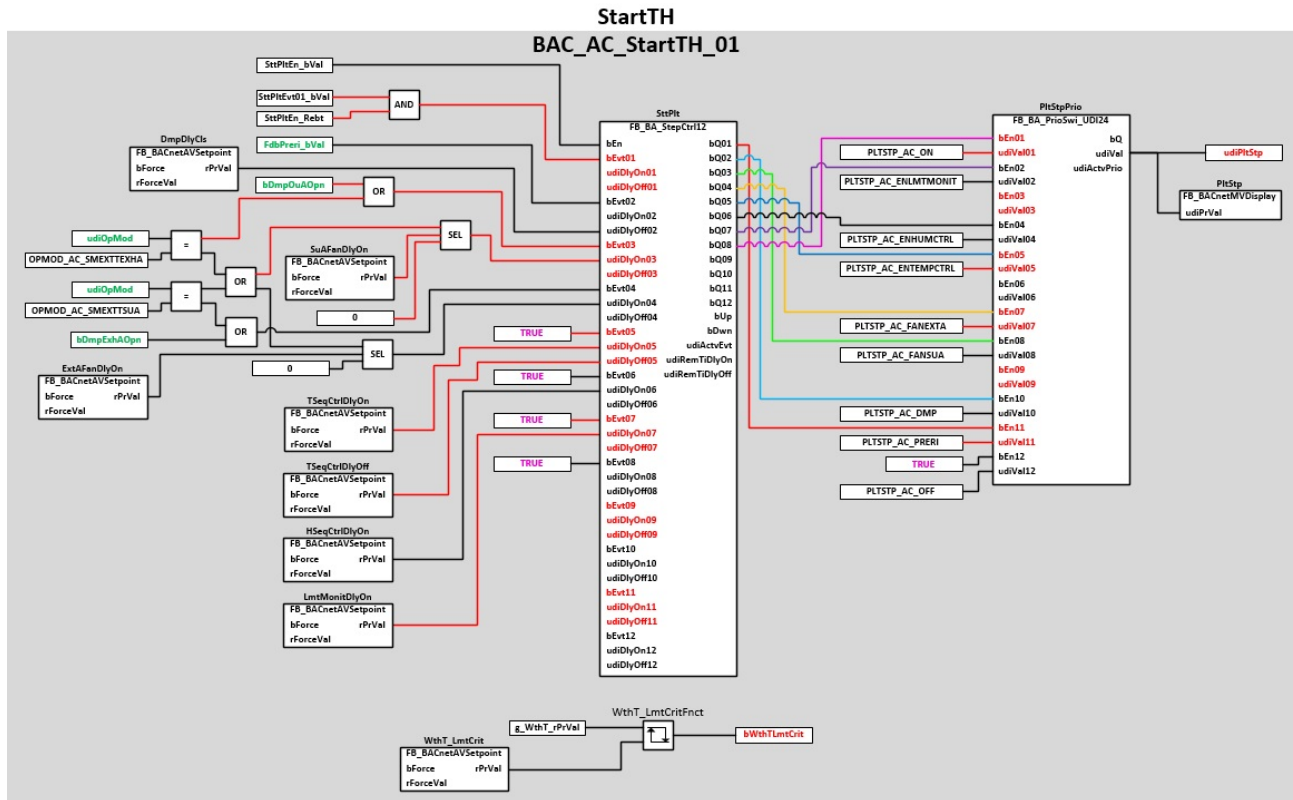
Schnittstelle



Anlagenschema



Blockdiagramm



Die folgenden Anlagenschritte werden über die Variable udiPltStp beschrieben:

udiPltStp	
PLTSTP_AC_OFF [▶ 364]	Aus//Off
PLTSTP_AC PRERI [▶ 364]	Vorspülen//pre-rinse
PLTSTP_AC DMP [▶ 364]	Klappe//Damper
PLTSTP_AC FANSUA [▶ 364]	Zuluftventilator//Fan supply air
PLTSTP_AC FANEXTA [▶ 364]	Abluftventilator//Fan extract air
PLTSTP_AC ENTEMPCTRL [▶ 364]	Freigabe Temperaturregelung,Sequenz Link//Enable temperature control, Sequence link
PLTSTP_AC ENHUMCTRL [▶ 364]	Freigabe Feuchteregelung,Sequenz Link//Enable temperature control, Sequence link
PLTSTP_AC ENLMTMONIT [▶ 364]	Freigabe Grenzwertüberwachung//Enable limit monitoring
PLTSTP_AC ON [▶ 364]	Ein//On

VAR_INPUT

```

udiOpMod      : UDINT;
bExhADmpOpn  : BOOL;
bOuADmpOpn   : BOOL;
bPreriRdy    : BOOL;
bMixADmpAvail : BOOL;
    
```

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod 01 \[▶ 524\]](#)

bOuADmpOpn: Rückmeldung Außenluftklappe auf

bExhADmpOpn: Rückmeldung Fortluftklappe auf

bPreriRdy: Vorspülen am Heizregister abgeschlossen

bMixADmpAvail: Mischluftsystem für Anfahrprozess vorhanden

VAR_OUTPUT

```

udiPltStp      : UDINT;
bWhtTLmtCrit  : BOOL;

```

udiPltStp: Ausgabe der aktuellen Anlagen-Anfahrsschritte

bWhtTLmtCrit: Die Außentemperatur ist unterhalb eines kritischen Wertes, so dass beim Anfahren der RLT-Anlage mit dem Vorspülen des Heizregisters angefahren werden muss. Die Pumpen des Vor- und Nacherhitzer werden eingeschaltet. Bei den Mischluftklappen wird beim Anlagenstart die Rampenfunktion aktiviert.

VAR CONSTANT

```

PLT_NUM      : BYTE := 1;

```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [► 182]

Die Auswertung der Alarmergebnisse einer Anlage, z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_CmnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
DmpDlyCls	FB_BACnetAVSetpoint [► 70]	AV-Objekt verzögert das Schließen der Klappen beim Herunterfahren der RLT-Anlage. Damit wird die Nachlaufzeit der Ventilatoren beim Abschalten der Anlage berücksichtigt.
SuAFanDlyOn	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Wertes für die Einschaltverzögerung des Zuluftventilators beim Einschalten der RLT-Anlage
ExtAFanDlyOn	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Wertes für die Einschaltverzögerung des Abluftventilators beim Einschalten der RLT-Anlage
TSeqCtrlDlyOn	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Wertes für die zeitverzögerte Freigabe der Temperaturregelung
TSeqCtrlDlyOff	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Wertes für die zeitverzögerte Ausschaltung der Temperaturregelung
HSeqCtrlDlyOn	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Wertes für die zeitverzögerte Freigabe der Feuchteregeleung
LmtMonitDlyOn	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Wertes für die zeitverzögerte Freigabe der Grenzwertüberwachung der Temperaturfühler
WthT_LmtCrit	FB_BACnetAVSetpoint [► 70]	AV-Objekt zur Eingabe des Außentemperaturwertes ab wann die RLT-Anlage mit dem Vorspülen des Heizregisters starten soll.
SttPlt	FB_BA_StepCtrl12 [► 252]	Der Funktionsbaustein SttPlt ist das Kernstück des Anlagenstartprogramms. Am Eingang bEn erfolgt eine generelle Freigabe des Funktionsbausteins. Ist bEn = FALSE, so sind Zwangsweise alle Ausgänge des Funktionsbausteins FALSE. Schritt 1: Vorspülen des Heizregisters PLTSTP_AC_PRERI

Instanz	Typ	Aufgabe
		<p>[▶ 364] Am Eingang bEvt01 wird das Anfahren der RLT zunächst mit dem Vorspülen des Heizregisters gestartet.</p> <p>Schritt 2: Öffnen der Außen- und Fortluftklappe <u>PLTSTP_AC_DMP [▶ 364]</u> Wenn der Vorspülprozess abgeschlossen oder aufgrund hoher Außentemperaturen nicht erforderlich ist, wird der Eingang bEvt02 gesetzt. Dadurch wird der nächste Anfahrsschritt, das Öffnen der Klappen angetriggert. Die Bedingung zum Weiterschalten in den Schritt Klappen Öffnen wird vom Funktionsbaustein FdbPreri gebildet.</p> <p>Schritt 3: Starten des Zuluftventilators <u>PLTSTP_AC_FANSUA [▶ 364]</u> Am Eingang bEvt03 wird der Endlagenschalter der Außenluftklappe angeschlossen. Bei offener Klappe wird Schritt 3 das Starten des Zuluftventilators angetriggert. Das Starten des Zuluftventilators kann zusätzlich am Eingang udiDlyOn03 verzögert werden. In der Betriebsart Entrauchen wird der Wert für die Anfahrverzögerung der Ventilatoren durch den vorgelagerten Selector auf den Wert null gesetzt. Damit springt das Programm unverzögert in den nächsten Schritt.</p> <p>Schritt 4: Starten des Abluftventilators <u>PLTSTP_AC_FANEXTA [▶ 364]</u> Am Eingang bEvt04 wird der Endlagenschalter der Fortluftklappe angeschlossen. Bei offener Klappe wird Schritt 4 das Starten des Fortluftventilators angetriggert. Das Starten des Abluftventilators kann zusätzlich am Eingang udiDlyOn04 verzögert werden. In der Betriebsart Entrauchen wird der Wert für die Anfahrverzögerung der Ventilatoren durch den vorgelagerten Selector auf den Wert null gesetzt. Damit springt das Programm unverzögert in den nächsten Schritt.</p> <p>Schritt 5: Freigabe der Zuluftsequenzregelung Temperatur <u>PLTSTP_AC_ENTEMPCTRL [▶ 364]</u> Der Eingang bEvt05 ist konstant TRUE. Damit wird die Freigabe der Zuluftsequenzregelung nur durch den Wert von udiDlyOn05 verzögert.</p> <p>Schritt 6: Freigabe der Zuluftsequenzregelung Feuchte <u>PLTSTP_AC_ENHUMCTRL [▶ 364]</u> Der Eingang bEvt06 ist konstant TRUE. Damit wird die Freigabe der Zuluftsequenzregelung nur durch den Wert von udiDlyOn06 verzögert.</p> <p>Schritt 7: Freigabe der Fühlergrenzwertüberwachung <u>PLTSTP_AC_ENLMTMONIT [▶ 364]</u> Der Eingang bEvt07 ist konstant TRUE. Damit wird die Freigabe der Zuluftsequenzregelung nur durch den Wert von udiDlyOn07 verzögert.</p> <p>Schritt 8: Zeigt an, dass die Lüftungsanlage in vollem Betrieb ist <u>PLTSTP_AC_ON [▶ 364]</u> Der Eingang bEvt08 ist konstant TRUE. Damit wird die Freigabe der Zuluftsequenzregelung nur durch den Wert von udiDlyOn07 verzögert.</p> <p>Beim Herunterfahren der RLT-Anlage wird zunächst die</p>

Instanz	Typ	Aufgabe
		<p>Feuchteregelung (Schritt 6) und danach zeitverzögert die Temperaturregelung (Schritt 5) abgeschaltet. Die Abschaltverzögerung der Temperaturregelung verhindert eine Kondensation von Restfeuchte in den Luftkanälen und daraus resultierend das Ansprechen eines Hygrostaten bei abgeschalteter Anlage.</p> <p>Um eine Nachlaufzeit der Ventilatoren zu überbrücken, ist das Schließen der Klappen beim Herunterfahren der Anlage durch den Timer DmpDlyCls verzögert.</p>
SttPltEn	FB BA MMUX B04 [▶ 213]	Der Multiplexer SttPltEn erzeugt die generelle Freigabe des Anfahrprogrammes in Abhängigkeit der Anlagenbetriebsarten. Ändert sich die Anlagenbetriebsart während der laufenden Betriebs dann wird die Variable SttPltEn_Rebt am Ausgang des Vergleiches = TRUE. Damit wird der Anlagenstartbaustein SttPlt kurzzeitig zurückgesetzt. Das Startprogramm wird damit in einen definierten Zustand versetzt und entsprechend der neuen Betriebsart neu gestartet.
SttPltEvt01	FB BA MMUX B24 [▶ 208]	Der Multiplexer SttPltEvt01 erzeugt die Freigabe zum Starten des Vorspülprozesses in Abhängigkeit der Anlagenbetriebsarten
FdbPreri	FB BA MMUX B08 [▶ 208]	<p>Der Multiplexer FdbPreri erzeugt die Freigabe für das Öffnen der Klappen.</p> <p>In den Betriebsarten OPMOD_AC_SMEXTTPRG, OPMOD_AC_SMEXTTSUA, OPMOD_AC_SMEXTTEXHA, OPMOD_AC_NGTCOL und OPMOD_AC_OVRHTGPRTC wird am Ausgang bVal ein TRUE ausgegeben. Damit wird der Schritt 1 des Anlagenstartbausteins SttPlt bei diesen Betriebsarten übersprungen. Der Vorspülprozess wird als abgeschlossen gemeldet.</p> <p>Für den normalen Anlagenbetrieb wird das Öffnen der Klappen unter den folgenden Bedingungen angetriggert:</p> <ol style="list-style-type: none"> 1. bMixADmpAvail = TRUE (Es ist ein Mischluftsystem vorhanden. Der Anfahrbetrieb wird mittels Mischluftsystem realisiert) 2. bPreriRdy = TRUE (Der Vorspülprozess ist abgeschlossen) 3. Die Außentemperatur ist unkritisch. Der Ausgang des Funktionsbausteins WthT_LmtCritFnct ist FALSE.
WthT_LmtCritFnct	FB BA Swi2P [▶ 148]	Der Funktionsbaustein WthT_LmtCritFnct ist ein Zweipunktschalter und setzt die Ausgangsvariable bWthTLmtCrit in Abhängigkeit der Außentemperatur auf TRUE oder FALSE. Der Schalterpunkt für den Zweipunktschalter wird über das AV-Objekt WthT_LmtCrit vorgegeben.
PltStpPrio	FB BA PrioSwi UDI1 2 [▶ 216]	Der Prioritätenschalter PltStpPrio wertet die aktuellen Schritte während des Anfahrprozesses aus und gibt sie mittels der Variable udiVal an das BACnet-MV-Objekt PltStp weiter zur Anzeige in der Managementebene.
PltStp	FB BACnetMVDisplay [▶ 130]	Das BACnet-MV-Objekt zeigt den aktuellen Schritt während des Anfahrprozesses an.

Versionshistorie

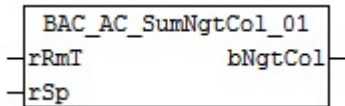
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.53 BAC_AC_SumNgtCol_01

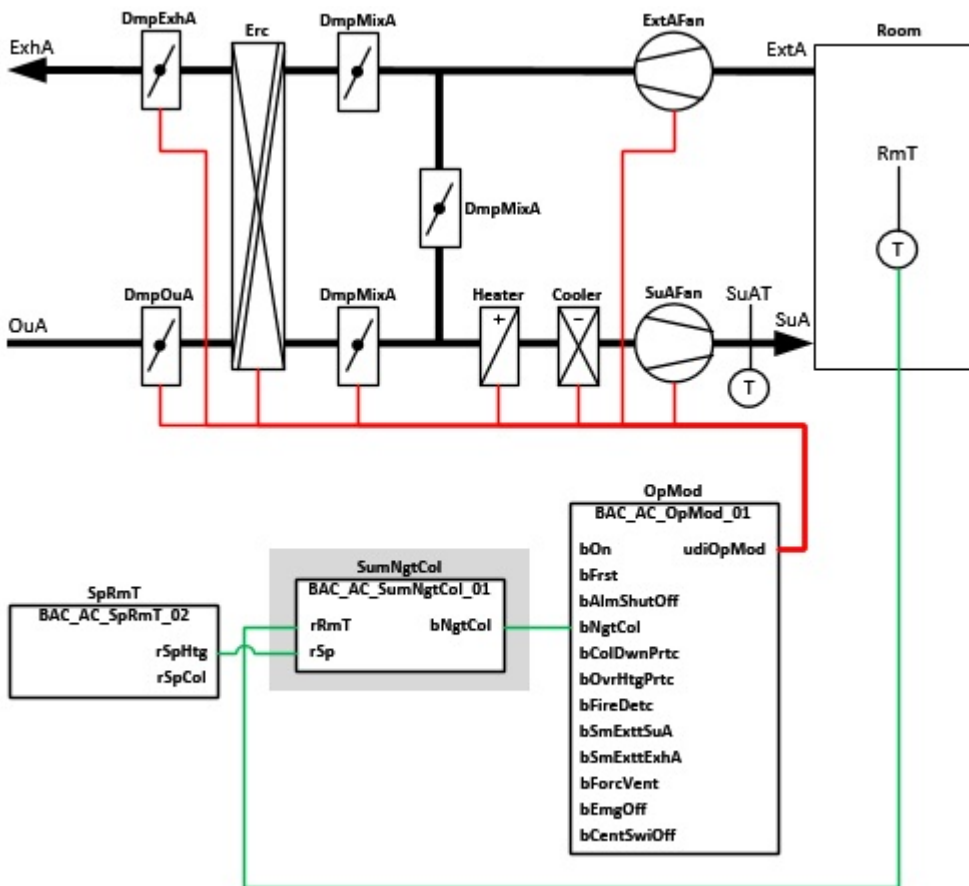
Funktionsbeschreibung

Das Template **BAC_AC_SumNgtCol_01** wird verwendet um nachts mit kühler Außenluft die tags zuvor aufgeheizten Räume herunter zu kühlen. Die Funktion der Sommernachtkühlung dient zur Verbesserung der Luftqualität und zur Einsparung elektrischer Energie. In den ersten Stunden des nächsten Sommertages wird elektrische Energie zur Kälteerzeugung eingespart.

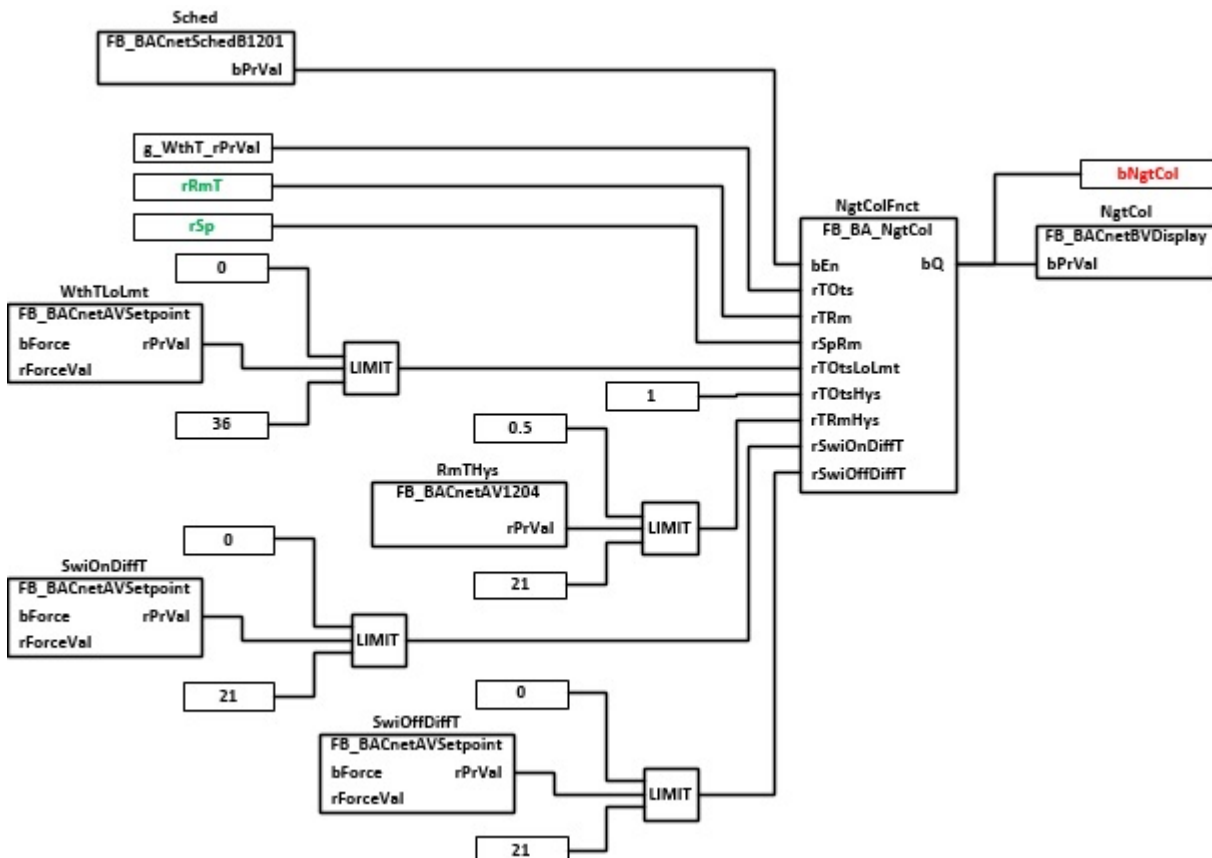
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

```
rRmT : REAL;
rSp : REAL;
```

rRmT : Messwert Raumtemperatur

rSp: Sollwert Raumtemperatur

VAR_OUTPUT

```
bNgtCol : BOOL;
```

bNgtCol: Sommernachtkühlung eingeschaltet

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins `FB_BA Alarm`. [[182](#)]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates `BAC_PltAlm_01` [[372](#)] mittels des Funktionsbausteins `FB_BA AlarmPlt`. [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates `BAC_PltComnMsg_01` durch den Funktionsbaustein `FB_BA ComnMsg` [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
RmTHys	FB BACnetAVSetpoint [▶ 70]	Eingabe eines Hysteresewertes um den Raumtemperatursollwert. Damit wird ein zu häufiges Ein- und Ausschalten bei schwankender Raumtemperatur vermieden. Der Einschaltpunkt ist überschritten wenn die Raumtemperatur $rRmT$ größer als $SpRm + RmTHys$ ist. Der Ausschaltpunkt ist erreicht wenn die Raumtemperatur $SpRm$ unterschritten hat.
Sched	FB BACnetSchedB120 1 [▶ 132]	Mit dem Schedule-Objekt werden die Zeiträume definiert in denen die Nachtkühlung freigegeben ist.
WthTLolmt	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für die untere Außentemperatur. Die untere Außentemperatur verhindert ein zu starkes Auskühlen.
SwiOffDifft	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für Differenz zwischen der Raumtemperatur und der Außentemperatur ab der die Sommernachtkühlung gesperrt wird.
SwiOnDifft	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für Differenz zwischen der Raumtemperatur und der Außentemperatur ab der die Sommernachtkühlung freigegeben wird.
NgtColFunct	FB BA NgtCol [▶ 236]	Der Funktionsbaustein NgtColFunct ist das Kernstück des Templates BAC_AC_SumNgtCol_01 und beinhaltet den eigentlichen Steuerungsprozess des Nachtkühlprogramms
NgtCol	FB BACnetBVDisplay [▶ 96], FB BACnetBV1204 [▶ 94]	Das BV-Objekt signalisiert, dass die Nachtkühlung eingeschaltet ist.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.54 BAC_H_HtgCir_01

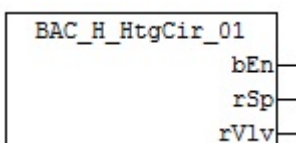
Anwendung

Das Aufruftemplate **BAC_H_HtgCir_01** ist ein Witterungstemperatur geführter Heizkreis. Es beinhaltet eine Pumpe 1-stufig, ein stetiges Ventil und eine Heizkennlinie mit 4 Stützpunkten.

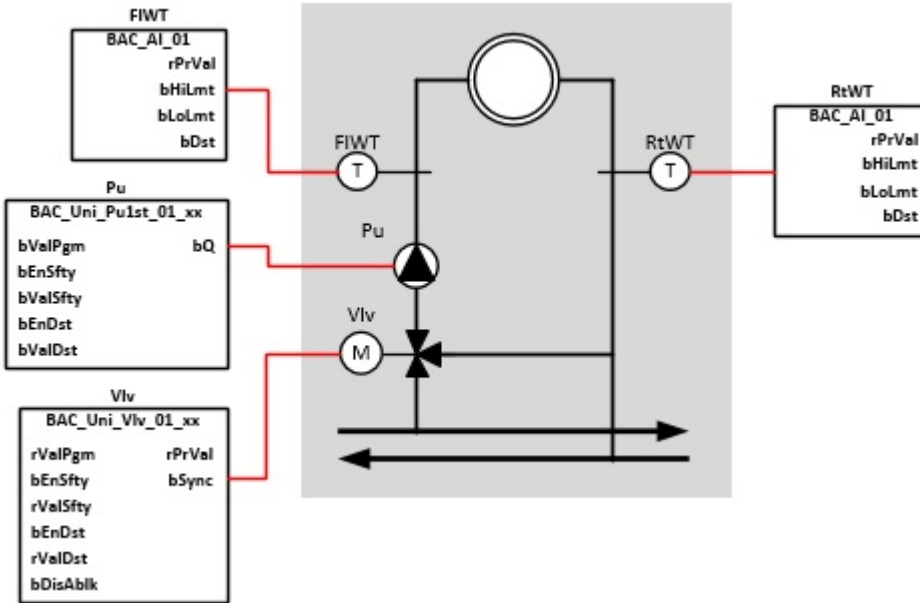
Die wesentlichen Aufgaben des Templates sind:

- Regelung der Vorlauftemperatur
- Ansteuerung der Pumpe
- Ansteuerung des Regelventils

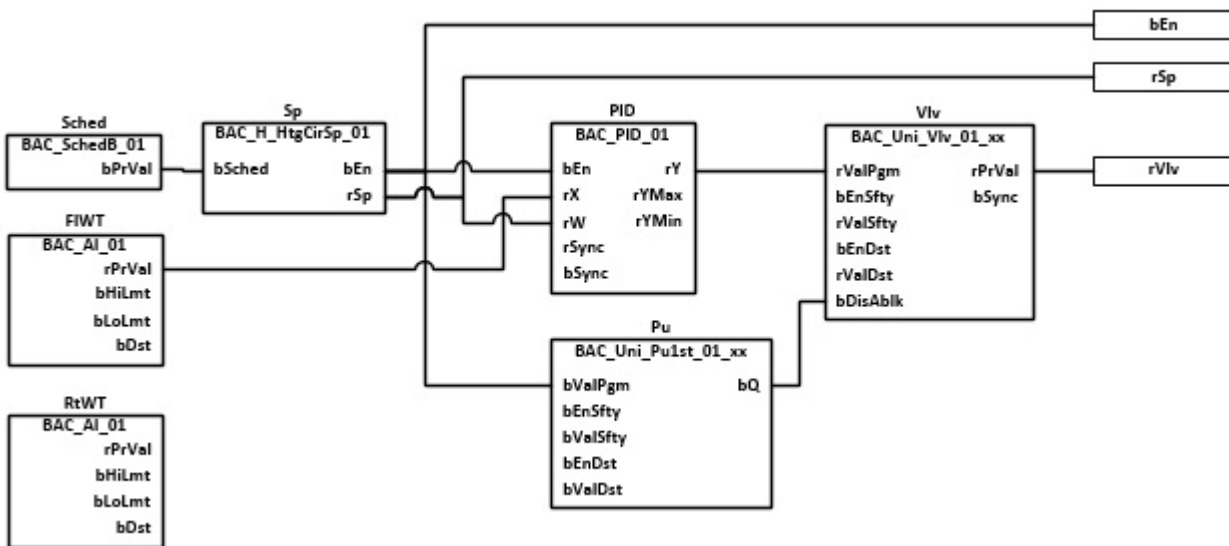
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_OUTPUT

```

bEn      : REAL;
rSp      : REAL;
rVlv     : REAL;
    
```

bEn: Ausgabe der Freigabe des Heizkreises

rSp: Ausgabe des Sollwertes des Heizkreises

rVlv: Ausgabe der Ventilstellung

Programmbeschreibung

Instanz	Typ	Aufgabe
Sched	BAC_SchedB_01 [▶ 718]	Sub-Template Zeitschaltprogramm für die Umschaltung zwischen Tag- und Nachtbetrieb
FIWT	BAC_AI_01 [▶ 683]	Sub-Template zur Anzeige der Vorlauftemperatur
RtWT	BAC_AI_01 [▶ 683]	Sub-Template zur Anzeige der Rücklauftemperatur

Instanz	Typ	Aufgabe
Sp	BAC H HtgCirSp_01 [▶_553]	Sub-Template Witterungstemperatur geführte Heizkreissteuerung. Es beinhaltet eine Heizkurve, einen Heizgrenzschalter und die Betriebsarten Frost / Nacht / Tag / Auto mit zugehörigem Sollwert.
PID	BAC_PID_01 [▶_597]	Sub-Template PID-Regler zur Regelung der Vorlauftemperatur
Vlv	BAC_Uni_Vlv_01_11 [▶_608]	Sub-Template zur Ansteuerung eines stetigen Ventils
Pu	BAC_Uni_Pu1st_01_61 [▶_579]	Sub-Template zur Ansteuerung einer einstufigen Pumpe

Versionshistorie

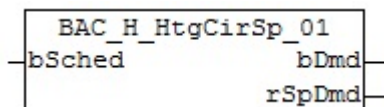
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.55 BAC_H_HtgCirSp_01

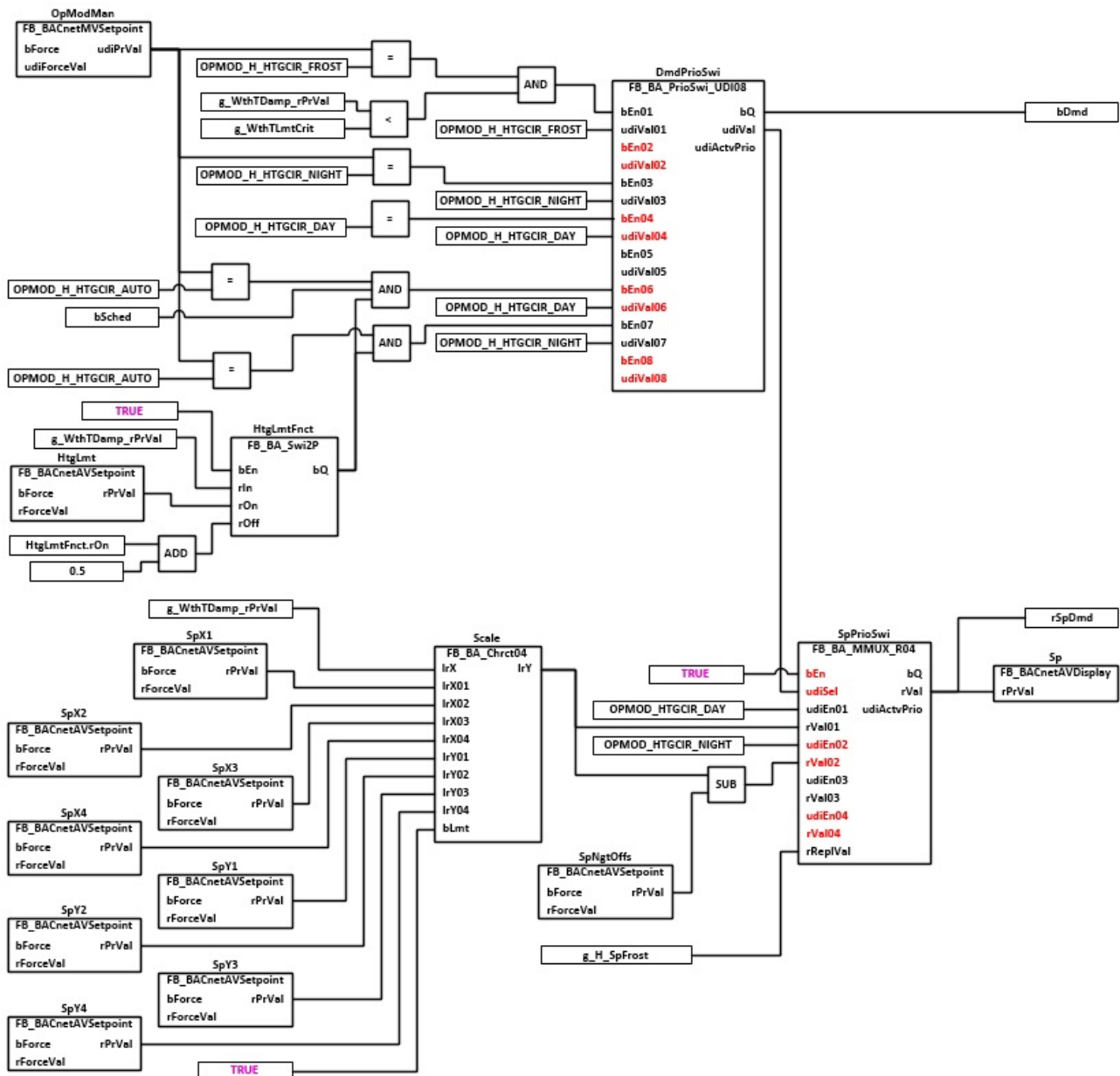
Anwendung

Das Template ist eine Witterungstemperatur geführte Heizkreissteuerung. Es beinhaltet eine Heizkurve, einen Heizgrenzschalter und die Betriebsarten Frost / Nacht / Tag / Auto mit zugehörigem Sollwert.

Schnittstelle



Blackschaltbild



VAR_INPUT

bSched : BOOL;

bSched: Eingang Zeitschaltplan für die Umschaltung zwischen Tag- und Nachtbetrieb

VAR_OUTPUT

bDmd : BOOL;

rSpDmd : REAL;

bDmd: Ausgabe Anforderung Heizbetrieb in Abhängigkeit der Betriebsart **OpModMan**.

rSpDmd: Ausgabe des Sollwertes für den Heizkreis.

VAR CONSTANT

PLT_NUM : BYTE := 1;

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt. Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der

Templates mittels des Alarmbausteins [FB BA Alarm](#). [[▶ 182](#)]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01](#) [[▶ 372](#)] mittels des Funktionsbausteins [FB BA AlarmPlt](#). [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg](#) [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
OpMod	FB BACnetMVSetpoint [▶ 131]	MV-Objekt zur manuellen Steuerung des Heizkreises von der MBE oder einem lokalen Bediendisplay. Folgende Betriebsarten können vorgegeben werden: OPMOD H HTGCIR FROST [▶ 364] / OPMOD H HTGCIR NIGHT [▶ 364] / OPMOD H HTGCIR DAY [▶ 364] / OPMOD H HTGCIR AUTO [▶ 364]
DmdPrioSwi	FB BA PrioSwi_UDI08 [▶ 216]	Der Prioritätenschalter priorisiert die Betriebsarten des Heizkreises.
HtgLmt	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Heizgrenzwertes.
HtgLmtFnc t	FB BA Swi2P [▶ 148]	Der Zweipunktschalter gibt in Abhängigkeit der gedämpften Außentemperatur g_WthTDamp_rPrVal [▶ 364] und des Heizgrenzwertes den Heizbetrieb in der Betriebsart OPMOD H HTGCIR AUTO [▶ 364] frei.
SpX1	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für den Stützpunkt X1
SpX2	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für den Stützpunkt X2
SpX3	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für den Stützpunkt X3
SpX4	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für den Stützpunkt X4
SpY1	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für den Stützpunkt Y1
SpY2	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für den Stützpunkt Y2
SpY3	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für den Stützpunkt Y3
SpY4	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für den Stützpunkt Y4
Scale	FB BA Chrct04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie für den Heizkreis in Abhängigkeit der Außentemperatur.
SpPrioSwi	FB BA MMUX_R04 [▶ 208]	Der Multiplexer-Funktionsbaustein gibt in Abhängigkeit der aktuellen Betriebsart OpModMan den dazu gehörigen Sollwert aus.
SpNgtOffs	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Wertes für die Nachtabenkung
Sp	FB BACnetAVDisplay [▶ 69]	Anzeige des Sollwertes für den Heizkreis. Dieser wird an dem Ausgang rSpDmd ausgegeben.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.56 BAC_DHW_01

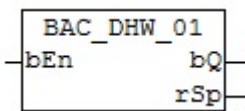
Anwendung

Das Aufruftemplate **BAC_DHW_01** dient zur Steuerung und Regelung eines Warmwasserspeichers.

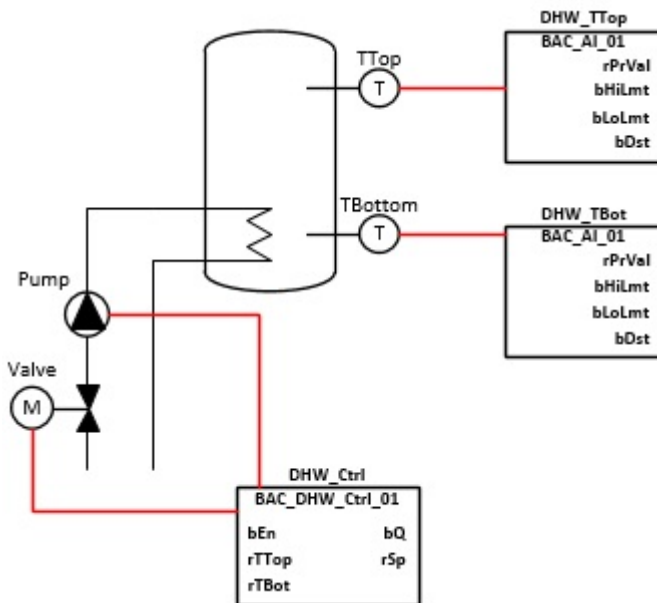
Die wesentlichen Aufgaben des Templates sind:

- Regelung der Speichertemperatur
- Steuerung Freigabe des Ladevorgangs

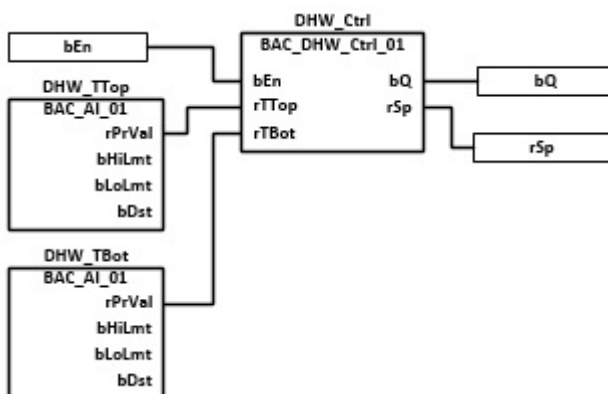
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

bEn : BOOL;

bEn: Eingang Freigabe des Templates. Ist dieser Eingang FALSE, so wird an **rSp** der Wert 0 ausgegeben und **bQ** wird FALSE.

VAR_OUTPUT

bQ : BOOL;
rSp : REAL;

bQ: Ausgang Ladung des Warmwasserspeichers.

rSp: Ausgabe des Sollwertes für die Warmwassertemperatur.

Wenn die Freigabe **En** des Templates FALSE ist, dann wird an **rSp** der Wert 0 ausgegeben.

Programmbeschreibung

Instanz	Typ	Aufgabe
DHW_TTop	BAC_AI_01 [▶ 683]	Sub-Template AI-Objekt Speichertemperatur oben
DHW_TBot	BAC_AI_01 [▶ 683]	Sub-Template AI-Objekt Speichertemperatur unten
DHW_Ctrl	BAC_DHW_Ctrl_01 [▶ 557]	Sub-Template Regelung und Steuerung der Speichertemperatur

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.57 BAC_DHW_Ctrl_01

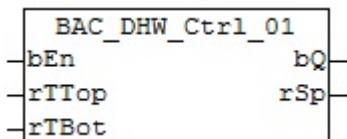
Anwendung

Das Template steuert die Ladung eines Warmwasserspeichers.

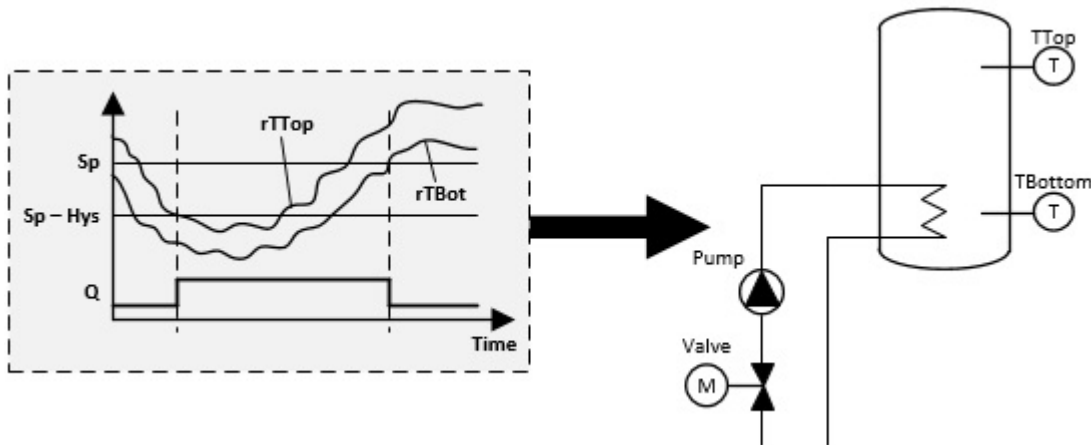
Die Freigabe erhält das Template mittels der Eingangsvariable **bEn**. Im Template wird nun die obere Speichertemperatur **rTTop** mit dem Wert der Differenz **Sp - Hys** verglichen. Unterschreitet die obere Speichertemperatur den Wert, so wird die Ladung des Warmwasserspeichers aktiviert.

Überschreitet die untere Speichertemperatur **rTBot** den Sollwert **Sp**, so wird die Ladung des Warmwasserspeichers deaktiviert.

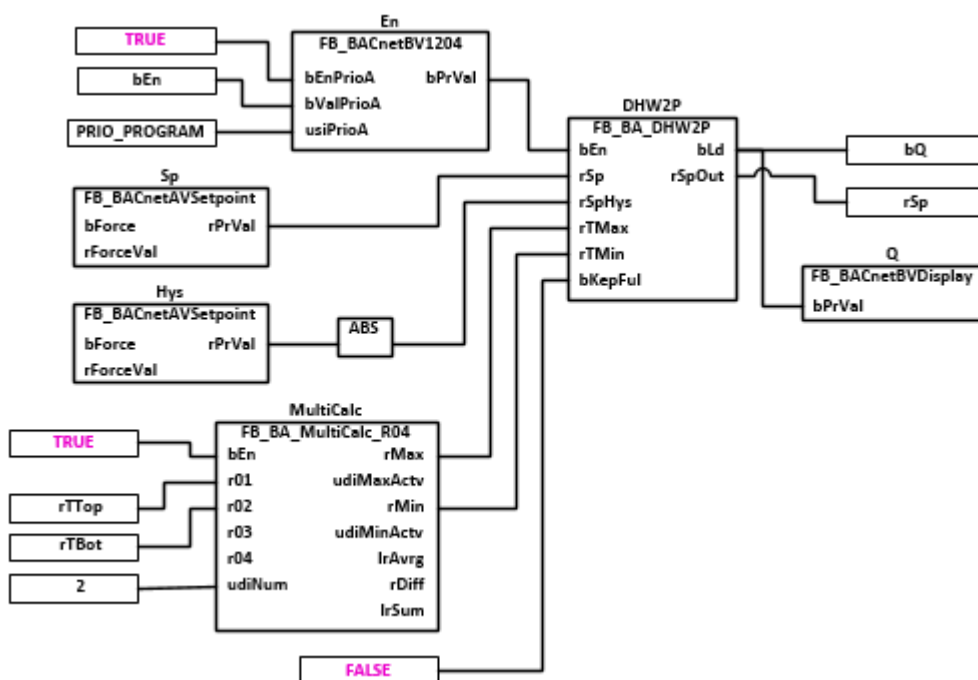
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

bEn : BOOL;
 rTTop : REAL;
 rTBot : REAL;

bEn: Eingang Freigabe des Templates. Ist dieser Eingang FALSE, so wird an **rSp** der Wert 0 ausgegeben und **bQ** wird FALSE.

rTTop: Eingang obere Speichertemperatur

rTBot: Eingang untere Speichertemperatur

VAR_OUTPUT

bQ : BOOL;
 rSp : REAL;

bQ: Ausgang Ladung des Warmwasserspeichers.

rSp: Ausgabe des Sollwertes für die Warmwassertemperatur.

Wenn die Freigabe **En** des Templates FALSE ist, dann wird an **rSp** der Wert 0 ausgegeben.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarmer von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmerbausteins [FB_BA_Alarm](#). [▶ 182]

Die Auswertung der Alarmer einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb der Templates [BAC_PltAlm_01](#) [▶ 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage erfolgt innerhalb der Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
En	FB_BACnetBV1204 [▶ 94]	BV-Objekt zur Anzeige der Freigabe des Templates Ladung Warmwasserspeicher.
Sp	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Sollwertes Warmwassertemperatur. Wird dieser Wert von der unteren Speichertemperatur rTBot überschritten, so wird die Ladung des Warmwasserspeichers deaktiviert. Der Warmwasserspeicher hat seine gewünschte Temperatur erreicht.
Hys	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe der Sollwert-Hystere. Wenn die obere Speichertemperatur rTTop den Wert der Differenz Sp - Hys unterschritten hat, so wird die Ladung des Warmwasserspeichers aktiviert. Anzeigt wird dieses mit dem BACnet Objekt Q .
MultiCalc	FB_BA_MultiCalc_R04 [▶ 211]	Der Funktionsbaustein ermittelt den Maximal- und Minimalwert der Eingänge r01-r04 und gibt diese an rMax und rMin aus.
DHW2P	FB_BA_DHW2P [▶ 219]	Der Funktionsbaustein ist das Kernstück um die Ladung eines Warmwasserspeichers zu steuern. Mittels eines Zweipunkt-Reglers wird die Ladung eines Warmwasserspeichers gesteuert. An dem Eingang bEn wird die Speicherladung frei gegeben. Bei aktiver Speicherladung ist der Ausgang bLd TRUE. Mit der Variablen rSp wird dem Funktionsbaustein der Sollwert für die Brauchwassertemperatur übergeben. An dem Eingang rTMin wird eine Minimalauswahl, am Eingang rTMax eine Maximalauswahl aller Temperaturfühler des Warmwasserspeichers angeschlossen. Bedingt durch die Temperaturschichtung im Warmwasserspeicher ist der oberste Fühler in allgemeinen der mit der höchsten Temperatur und der untere derjenige mit der niedrigsten.
Q	FB_BACnetBVDisplay [▶ 96]	BV-Objekt zur Anzeige ob die Ladung des Warmwasserspeichers aktiviert ist.

Versionshistorie

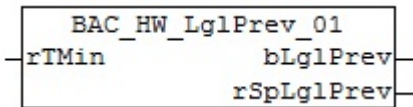
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.58 BAC_HW_LglPrev_01

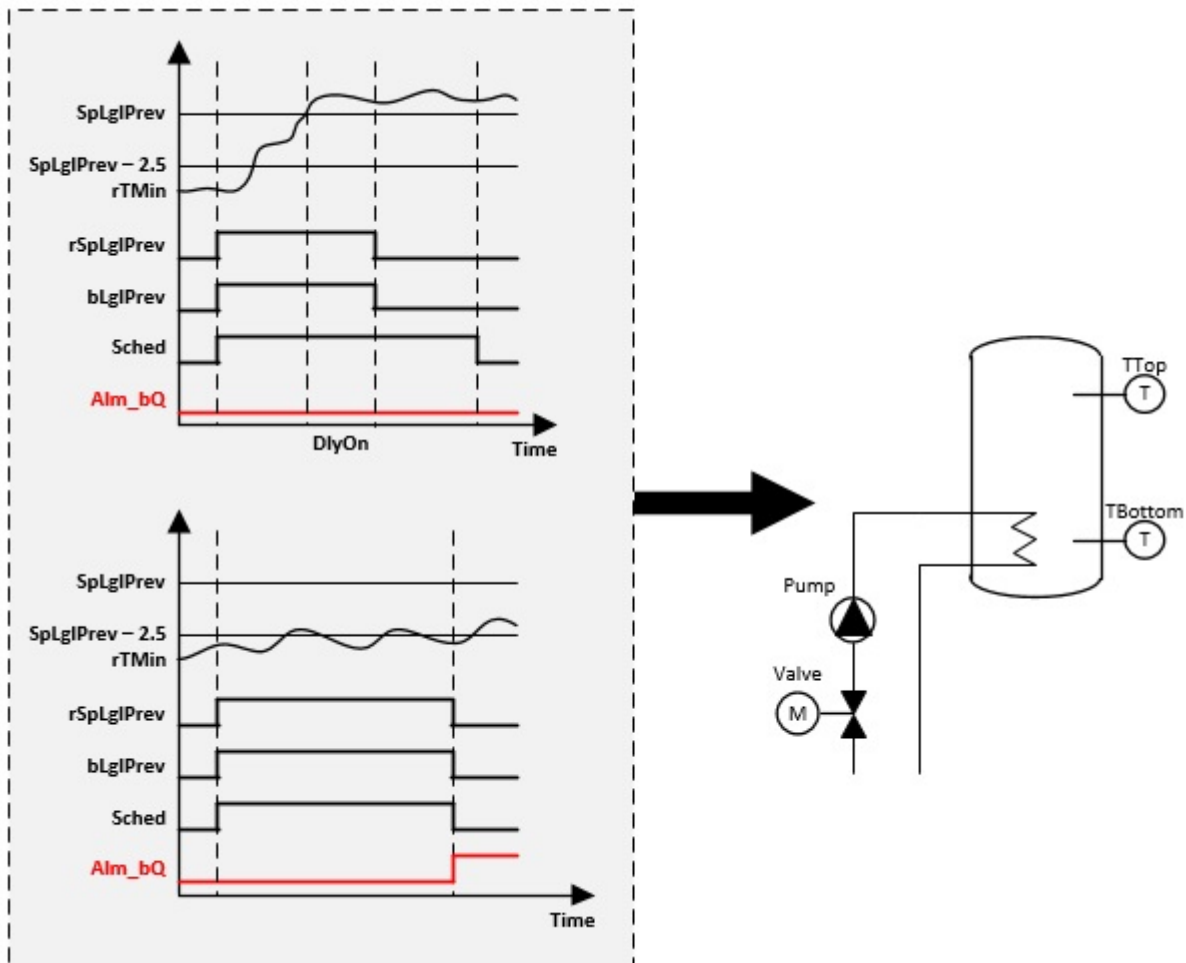
Anwendung

Das Template ist eine Schutzfunktion zur Verhinderung der Legionellen-Bildung im Warmwasser. Die Schutzfunktion wird durch den Zeitschaltplan **Sched** für ein bestimmtes Zeitfenster frei gegeben. Im Template wird nun die minimale Speichertemperatur **rTMin** mit dem Sollwert **SpLglPrev** verglichen und ein Ladebefehl **bLglPrev** und Sollwert **rSpLglPrev** ausgegeben. Dadurch soll die Speichertemperatur innerhalb des Zeitfensters **Sched** für eine bestimmte Zeit **TiLglPrev** anstehen. Wenn diese Funktion nicht erfüllt worden ist, so wird ein Alarm generiert **AlmLglPrev**.

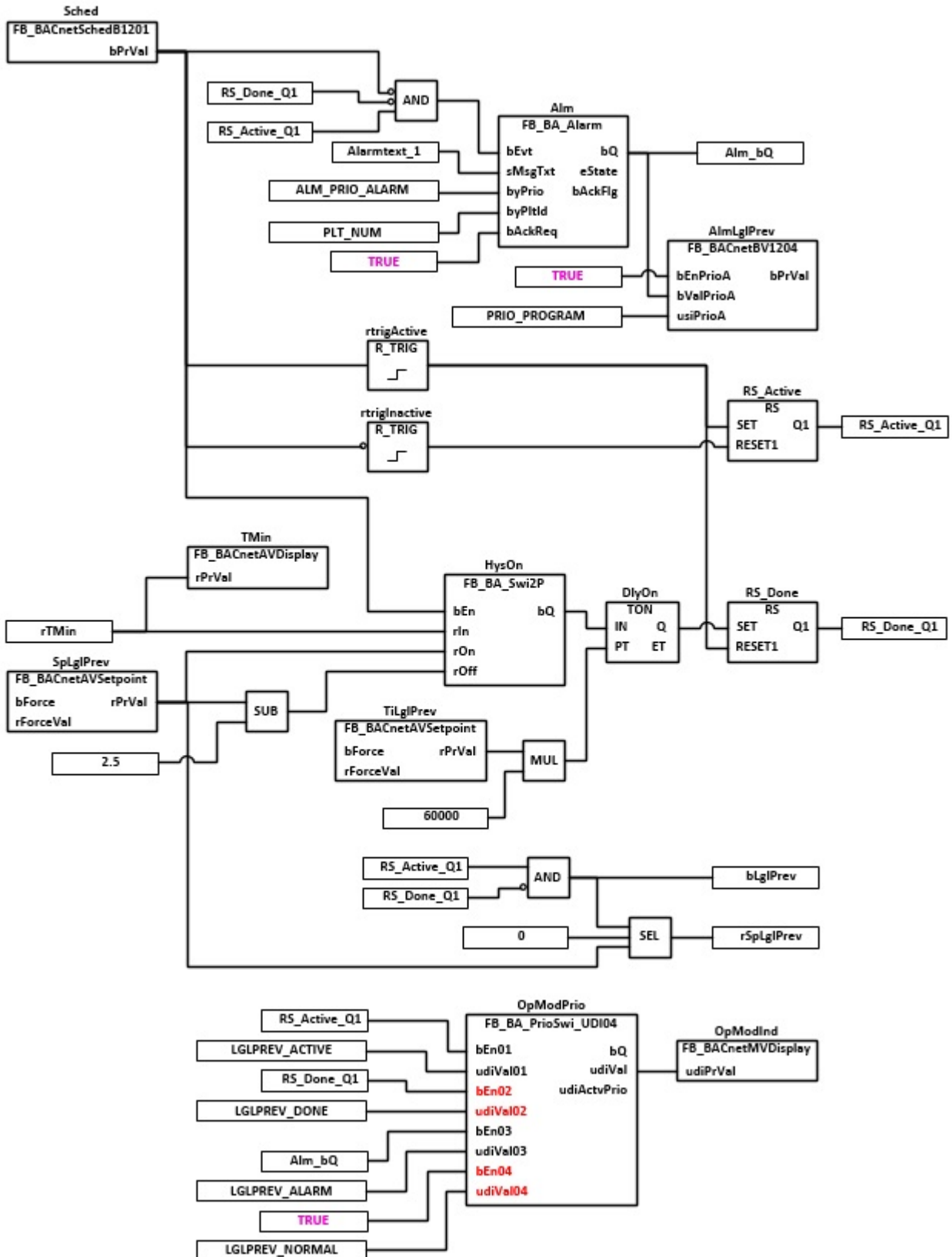
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_INPUT

rTMin : REAL;

rTMin: Eingang minimale Speichertemperatur

VAR_OUTPUT

```
bLglPrev      : BOOL;
rSpLglPrev    : REAL;
```

bLglPrev: Ausgabe Anforderung Legionellenschutz

rSpLglPrev: Ausgabe des Sollwertes Legionellenschutz

VAR CONSTANT

```
PLT_NUM      : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [[▶ 182](#)]

Die Auswertung der Alarmergebnisse einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [[▶ 372](#)] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
Sched	FB_BACnetSchedB1201 [▶ 132]	Eingabe Zeitschaltplan Freigabe Legionellenprävention
TMin	FB_BACnetAVDisplay [▶ 69]	Anzeige der minimalen Temperatur rTMin des Warmwasserspeichers.
SpLglPrev	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des Sollwertes Legionellenprävention.
HysOn	FB_BA_Swi2P [▶ 148]	Der Zweipunktschalter signalisiert in Abhängigkeit der minimalen Warmwasserspeichertemperatur rTMin und dem Sollwert Legionellenprävention SpLglPrev , dass die Legionellenprävention aktiv ist.
TiLglPrev	FB_BACnetAVSetpoint [▶ 70] MUL	AV-Objekt zur Eingabe der Zeitdauer Legionellenprävention. Multiplikation der Eingabe Zeitdauer Legionellenprävention in Minuten
DlyOn	TON	Zeitglied Verzögerung Legionellenprävention erfolgreich.
RS_Active	RS	Speicher Legionellenprävention aktiv
RS_Done	RS	Speicher Zeitfenster Legionellenprävention beendet.
Alm	FB_BA_Alarm [▶ 182]	Erfassung und Weiterverarbeitung des Alarms Legionellen-Sollwert nicht erreicht
AlmLglPrev	FB_BACnetBV1204 [▶ 94]	Meldet Alarm Legionellen-Sollwert nicht erreicht an die MBE.
OpModPrio	FB_BA_PrioSwi_UDI04 [▶ 216]	Der Prioritätenschalter priorisiert die Betriebsarten der Legionellenprävention.
OpModInd	FB_BACnetMVDisplay [▶ 130]	Das BACnet-MV-Objekt zeigt die aktuell gültige Betriebsart der Legionellen Schutzfunktion an. LGLPREV_NORMAL [▶ 364] Legionellenprävention normal, Alarm wurde zurückgesetzt

Instanz	Typ	Aufgabe
		LGLPREV_ACTIVE [▶ 364] Legionellenprävention aktiv
		LGLPREV_DONE [▶ 364] Legionellenprävention erledigt
		LGLPREV_ALARM [▶ 364] Legionellenprävention Alarm

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.59 BAC_Uni_FC_01_xx

Funktionsbeschreibung

Das Template **BAC_Uni_FC_01_xx** ist für die Ansteuerung eines Frequenzumrichters mit binären und analogen Ein- und Ausgängen.

i Die beiden Ausgangsvariablen rPrVal / bSync sind nur dann aktiv, wenn in dem eingesetzten Template die mechanische Vorrangbedienung FdbOutAO vorhanden ist. Ist dieses nicht der Fall, so geben die beiden Variablen den Wert Null aus.

Ausstattungsvarianten

Das Template **BAC_Uni_FC_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten des Templates werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

Optionen	Reparatur-schalter (Rep)	mechanische Vorrangbedienung Freigabe FU Rückmeldung Handschalter (Frg A-0-H)	mechanische Vorrangbedienung Freigabe FU Rückmeldung Relais (Rm-Ausg)	mechanische Vorrangbedienung Sollwert FU Rückmeldung Handschalter (Dreh A-H)	mechanische Vorrangbedienung Stellwert FU Rückmeldung Stellung Poti (Rm-Stellw)	Betriebsrückmeldung FU (Betr-FU)	Störmeldung FU (Stör-FU)	Störmeldung Motor-schutz-schalter (Therm)
Instanz	MntnSwi	LocSwi-BO	FdbOut-BO	LocSwiAO	FdbOutAO	FdbFC	DstFC	ThOvrlD
Datenpunkt Typ	BI	BI	BI	BI	AI	BI	BI	BI
	128	64	32	16	8	4	2	1
BAC_Uni_FC_01_002	0	0	0	0	0	0	1	0
BAC_Uni_FC_01_006	0	0	0	0	0	1	1	0
BAC_Uni_FC_01_007	0	0	0	0	0	1	1	1
BAC_Uni_FC_01_0086	0	1	0	1	0	1	1	0

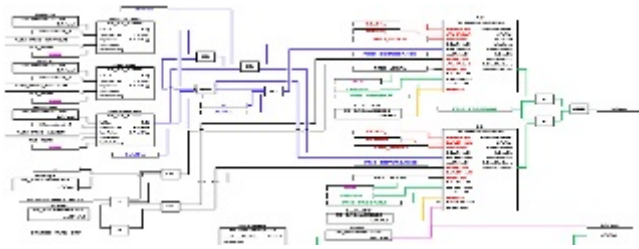
Optionen	Reparatur-schalter (Rep)	mechani-sche Vor-rangbe-dienung Freigabe FU Rück-meldung Hand-schalter (Frg A-0-H)	mechani-sche Vor-rangbe-dienung Freigabe FU Rück-meldung Relais (Rm-Ausg)	mechani-sche Vor-rangbe-dienung Sollwert FU Rück-meldung Hand-schalter (Dreh A-H)	mechani-sche Vor-rangbe-dienung Stellwert FU Rück-meldung Stellung Poti (Rm-Stellw)	Betriebs-rückmel-dung FU (Betr-FU)	Störmel-dung FU (Stör-FU)	Störmel-dung Mo-tor-schutz-schalter (Therm)
Instanz	MntnSwi	LocSwi-BO	FdbOut-BO	LocS-wiAO	FdbOu-tAO	FdbFC	DstFC	ThOvrlid
Daten-punkt Typ	BI	BI	BI	BI	AI	BI	BI	BI
	128	64	32	16	8	4	2	1
BAC_Uni_FC_01_087	0	1	0	1	0	1	1	1
BAC_Uni_FC_01_094	0	1	0	1	1	1	1	0
BAC_Uni_FC_01_130	1	0	0	0	0	0	1	0
BAC_Uni_FC_01_134	1	0	0	0	0	1	1	0
BAC_Uni_FC_01_135	1	0	0	0	0	1	1	1
BAC_Uni_FC_01_214	1	1	0	1	0	1	1	0
BAC_Uni_FC_01_215	1	1	0	1	0	1	1	1
BAC_Uni_FC_01_222	1	1	0	1	1	1	1	0
BAC_Uni_FC_01_223	1	1	0	1	1	1	1	1

Schnittstelle

BAC_Uni_FC_01_xx	
—bValPgm	bPrVal—
—rValPgm	rPrVal—
—bEnSfty	bSync—
—bValSfty	
—rValSfty	
—bEnDst	
—bValDst	
—rValDst	

Blockschaltbild Variante BAC_Uni_FC_01_223

<https://infosys.beckhoff.com/content/1031/tcba/Resources/12269787275.jpg>



VAR_INPUT

```
bValPgm    : BOOL;
rValPgm    : REAL;
bEnSfty    : BOOL;
bValSfty   : BOOL;
rValSfty   : REAL;
bEnDst     : BOOL;
bValDst    : BOOL;
rValDst    : REAL;
```

bValPgm: Binärer Wert Programmpriorität

rValPgm: Analoger Wert Programmpriorität

bEnSfty: Freigabe Sicherheitspriorität

bValSfty: Binärer Wert Sicherheitspriorität

rValSfty: Analoger Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität. An diesen Eingang könnte z. B. eine Prozessrückmeldung angelegt werden.

bValDst: Binärer Wert Störungspriorität. An diesen Eingang könnte z. B. eine Prozessrückmeldung angelegt werden.

rValDst: Analoger Wert Störungspriorität

VAR_OUTPUT

```
bPrVal     : BOOL;
rPrVal     : REAL;
bSync      : BOOL;
```

bPrVal: Betriebsrückmeldung FU

rPrVal : aktuelle Drehzahl des FU.

bSync: Ausgabe eines Impulses um den zum FU zugehörigen Regler, beim Rücksetzen vom Hand- in den Automatikbetrieb auf die aktuelle Drehzahl des FU's zu synchronisieren.

VAR CONSTANT

```
PLT_NUM    : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm](#). [[182](#)]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01](#) [[372](#)] mittels des Funktionsbausteins [FB BA AlarmPlt](#). [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg](#) [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe			
ThOvrd	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt Thermoschutz Motor (Kaltleiterauslösegerät, Motorschutzschalter etc.)			
DstFC	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt Störung Frequenzumrichter			
LocSwiBO	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt mechanische Vorrangbedienung Rückmeldung Handschalter binär			
LocSwiAO	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt mechanische Vorrangbedienung Rückmeldung Handschalter analog			
FdbFC	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt Betriebsrückmeldung FU			
MntnSwi	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt Wartungsschalter			
FdbOutAO	FB_BACnetAI1203 [▶ 49]	X	Rückgabewert Mechanische Vorrangbedienung Stellwert FU			
OpMod	FB_BACnetMVSetpoint [▶ 131]		MV-Objekt zur manuellen Steuerung des Frequenzumrichters von der MBE oder einem lokalen Bediendisplay			
ManSpd	FB_BACnetAVSetpoint [▶ 70]		AV-Objekt zur Eingabe der Drehzahl des Motors bei manueller Übersteuerung			
AlmThOvrd	FB_BA_Alarm [▶ 182]	x	Erfassung und Weiterverarbeitung eines Fehlers vom Thermoschutz des Motors (Kaltleiterauslösegerät, Motorschutzschalter etc.)			
AlmDstFC	FB_BA_Alarm [▶ 182]	x	Erfassung und Weiterverarbeitung eines Fehlers vom Frequenzumrichter			
AlmMntnSwi	FB_BA_Alarm [▶ 182]	x	Erfassung und Verarbeitung des Ereignisses Reparaturschalter ausgelöst.			
AO	FB_BACnetAO1203 [▶ 53]		AO-Objekt für die Vorgabe des Drehzahlsollwertes an den Frequenzumrichter			
			Priorität:	Freigabe	Wert	Bemerkung
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty	
			PRIO_DISTURBANCE (3)	Eingang bEnDst OR ThOvrd OR DstFC OR MntnSwi	keine Störung von bEnDst , ThOvrd , DstFC oder MntnSwi -> Wert von Eingang rValDst Störung von bEnDst , ThOvrd , DstFC oder MntnSwi = 0	
PRIO_LOCAL (8)	OpMod_u diPrVal = OPMOD_ MAN_OFF OR	Selector 0 OR ManSpd_rPrVal I	Bei Handbetrieb Wert von AV-Objekt ManSpd			

Instanz	Typ	optional	Aufgabe			
				OPMOD_MAN_STP01		
			PRIO_PROGRAM (15)	TRUE	rValPgm	Wert von Eingang rValPgm (z. B. Drehzahlorgabe vom Regler)
BO	<u>FB_BACnetBO1203</u> [▶ 82]		BO-Objekt für die Vorgabe der Freigabe an den Frequenzumrichter			
			Priorität:	Freigabe	Wert	Bemerkung
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty	
			PRIO_DISTURBANCE (3)	Eingang bEnDst OR ThOvrlid OR DstFC OR MntnSwi	Wert des Operators AND mit den Eingängen bValDst , ThOvrlid_bPrVal , MntnSwi_bPrVal , DstFC_bPrVal	
			PRIO_LOCAL (8)	OpMod_udiPrVal = OPMOD_MAN_OFF OR OPMOD_MAN_STP01	TRUE, wenn OpMod_udiPrVal = OPMOD_MAN_STP01	Bei Handbetrieb Wert von AV-Objekt ManSpd
			PRIO_PROGRAM (15)	TRUE	bValPgm	Wert von Eingang bValPgm
	EQ, EQ, AND		Wert des Netzwerkes ist TRUE, wenn die Aktive Priorität PRIO_PROGRAM (15) ist. Kann zur Synchronisation des Reglers bei Rückkehr in den Automatikbetrieb benutzt werden			
TLogAO	<u>FB_BACnetTLog1201</u> [▶ 137]		Trendaufzeichnung des AO-Objekts für die Vorgabe des Drehzahlollwertes an den Frequenzumrichter			

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Thermal_Overload	BOOL	X	Eingang	Digitaleingang - Motor thermische Überlast - Meldung - ausgelöst
BI_FC_Disturbance	BOOL	X	Eingang	Digitaleingang - FU Störung - Meldung - ausgelöst

Parameter	Typ	optional	Prozessabbild	
BI_FC_Feedback	BOOL	X	Eingang	Digitaleingang - FU - Meldung - Betrieb
BI_Maintenance_Switch	BOOL	X	Eingang	Digitaleingang – Wartungsschalter - Meldung - ausgelöst
BI_FC_Enable_LocalSwitch	BOOL	X	Eingang	Digitaleingang - Schalter Hand Freigabe FU - Meldung - Hand/ Auto
BI_Speed_LocalSwitch	BOOL	X	Eingang	Digitaleingang - Schalter Drehzahl - Meldung - Hand/ Auto
BI_Feedback_Binary_Output	BOOL	X	Eingang	Digitaleingang - Rückmeldung Schaltbefehl - Meldung - Ein/Aus
AI_Feedback_Speed_Poti	INT	X	Eingang	Analogeingang - Hand Potentiometer – Rückmeldung - Stellwert
BO_FC_Enable	BOOL		Ausgang	Digitalausgang - FU Schaltbefehl - Freigabe Ein/Aus
AO_FC_Speed_Value	INT		Ausgang	Analogausgang - FU Stellwert - Drehzahl

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.60 BAC_Uni_Dmp_01_xx

Funktionsbeschreibung

Das Template **BAC_Uni_Dmp_01_xx** ist für die Ansteuerung eines stetigen Klappenantriebs. Es besteht im Wesentlichen aus einem AO-Objekt inklusive Trend-Objekt, einem MV-Objekt zur manuellen Steuerung und dem dazu gehörigen AV-Objekt zur Eingabe der Position. Das Template wird ergänzt durch optionale BACnet Objekte, siehe Tabelle Ausstattungsvarianten.



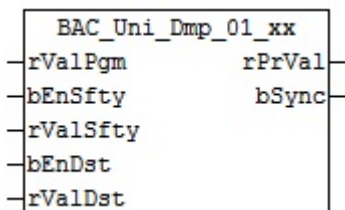
Die beiden Ausgangsvariablen rPrVal / bSync sind nur dann zur Synchronisation von Reglern verwendet werden, wenn in dem eingesetzten Template die Rückmeldung der Klappenstellung Fdb vorhanden ist.

Ausstattungsvarianten

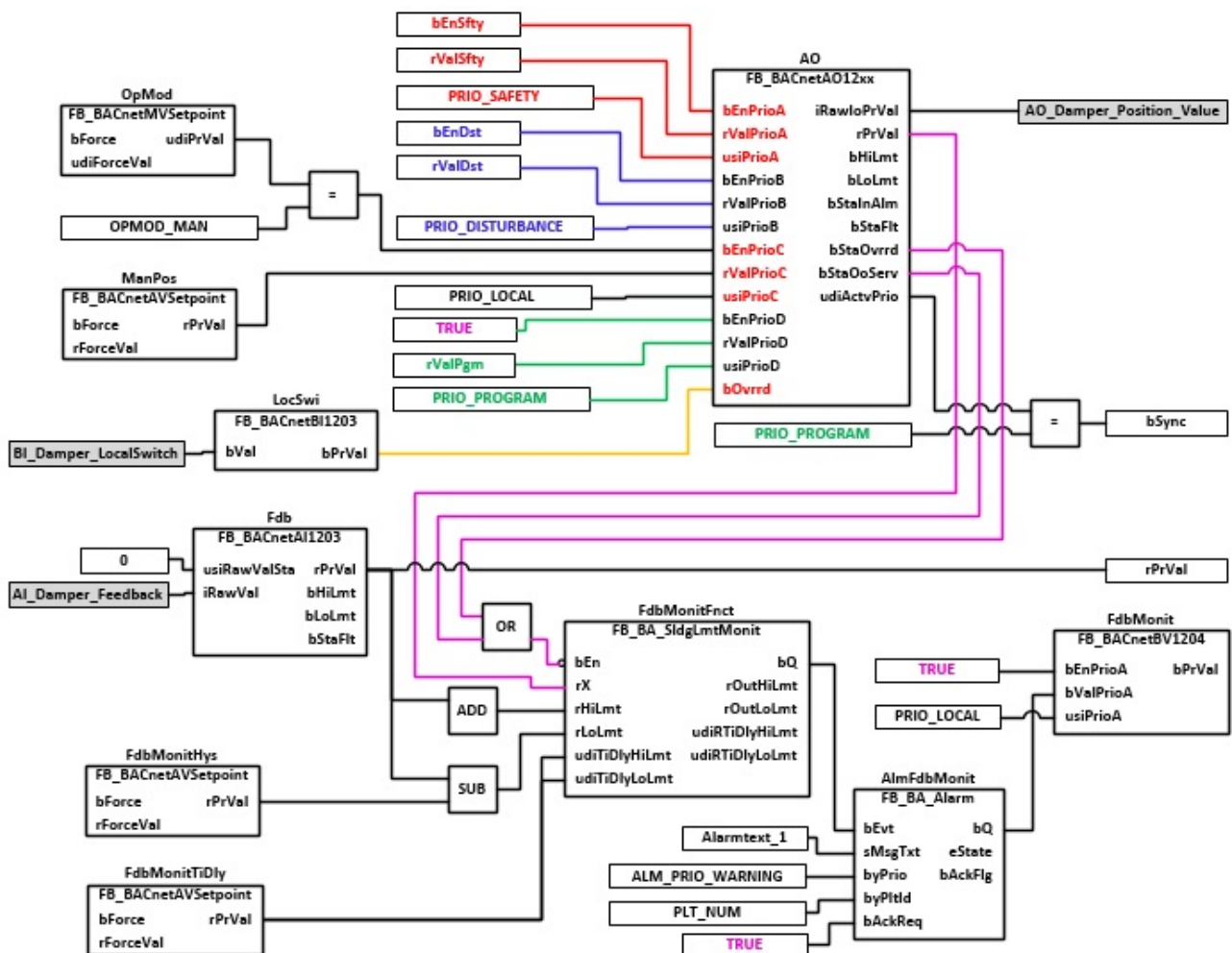
Das Template **BAC_Uni_Dmp_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten der Klappe werden nach einer Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

Optionen	Rückmeldung Klappenstellung mit Überwachung (Rm)	mechanische Vorrangbedienung Stellungs-rückmeldung Poti (Rm-Ausg)	mechanische Vorrangbedienung Stellungs-rückmeldung Hand-schalter (A-0-H)
Instanz	Fdb	FdbOut	LocSwi
Datenpunkt Typ	AI	AI	BI
	4	2	1
BAC_Uni_Dmp_01_00	0	0	0
BAC_Uni_Dmp_01_01	0	0	1
BAC_Uni_Dmp_01_03	0	1	1
BAC_Uni_Dmp_01_04	1	0	0
BAC_Uni_Dmp_01_05	1	0	1
BAC_Uni_Dmp_01_07	1	1	1

Schnittstelle



Blockschaltbild Variante BAC_Uni_Dmp_01_05



VAR_INPUT

```
rValPgm : REAL;
bEnSfty : BOOL;
bValSfty : BOOL;
bEnDst  : BOOL;
bValDst : BOOL;
```

rValPgm: Analoger Wert Programmpriorität

bEnSfty: Freigabe Sicherheitspriorität

rValSfty: Analoger Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität

rValDst: Analoger Wert Störungspriorität

VAR_OUTPUT

```
rPrVal : REAL;
bSync  : BOOL;
```

rPrVal : aktuelle Stellung des Klappenantriebs.

Die Ausgangsvariable **rPrVal** ist nur dann aktiv, wenn in dem eingesetzten Template die Rückmeldung der Klappenstellung **Fdb** vorhanden ist.

bSync: Ausgabe eines Impulses um der Klappe zugehörigen Regler beim Rücksetzen vom Hand- in den Automatikbetrieb auf die aktuelle Position der Klappe zu synchronisieren.

Die Ausgangsvariable **bSync** ist nur dann aktiv, wenn in dem eingesetzten Template die Rückmeldung der Klappenstellung **Fdb** vorhanden ist.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung des Klappenantriebs von der MBE oder einem lokalen Bediendisplay
ManPos	FB_BACnetAVSetpoint [► 70]		AV-Objekt zur Eingabe der Position des Klappe bei manueller Übersteuerung
FdbMonithys	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe der Hysteres für die Funktionsüberwachung des Regelventils mittels der Stellungsrückmeldung
FdbOut	FB_BACnetAI1203 [► 49]	X	AI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Poti

Instanz	Typ	optional	Aufgabe			
Fdb	FB_BACnetAI1203 [▶ 49]	X	AI-Objekt zur Erfassung der Stellungsrückmeldung Klappenstellung			
FdbMonitTidy	FB_BACnetAVSetpoint [▶ 70]	X	AV-Objekt zur Eingabe der Ansprechverzögerung der Funktionsüberwachung mittels Stellungsrückmeldung.			
LocSwi	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt meldet die mechanische Vorrangbedienung Stellungsrückmeldung Handschalter			
AO	FB_BACnetAO1203 [▶ 53]		AO-Objekt für die Ansteuerung des Regelventils.			
			Priorität:	Freigabe	Wert	Bemerkung
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty	
			PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst	
			PRIO_LOCAL (8)	OpMod_udiP rVal = OPMOD_MANN	ManPos_rPrVal	Bei Handbetrieb Wert von AV-Objekt ManPos
PRIO_PROGRAM (15)	TRUE	AbkFnct_bQ oder rValPgm	Wert von Eingang rValPgm oder 100% von Blockierschutz			
	EQ		Ist TRUE wenn die Aktive Priorität PRIO_PROGRAM (15) ist. Kann zur Synchronisation des Reglers bei Rückkehr in den Automatikbetrieb benutzt werden			
FdbMonitFunct	FB_BA_SldgLmtMonit [▶ 145]	X	Funktionsüberwachung des Klappenantriebs durch Vergleich des Stellausgangs und der Stellungsrückmeldung.			
AlmFdbMonit	FB_BA_Alarm [▶ 182]	X	Erfassung und Weiterverarbeitung eines Fehlers von der Stellungsrückmeldeüberwachung			
FdbMonit	FB_BACnetBV1204 [▶ 94]	X	Meldet Fehler Stellungsrückmeldung an die MBE			
TLogAO	FB_BACnetTLog1201 [▶ 137]		Zeichnet den Present Value des AO-Objekts auf			

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
AI_Damper_Feedback	INT	X	Eingang	Klappe - Messwert - Rückgabewert Stellung
AI_Feedback_Position_Poti	INT	X	Eingang	Klappe Potentiometer H-A - Messwert
BI_Damper_LocalSwitch	BOOL	X	Eingang	Klappe Schalter H-A - Meldung - Betriebsart

Parameter	Typ	optional	Prozessabbild	
AO_Damper_Position_Value	INT		Ausgang	Analogausgang - Klappe - Stellwert Position

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.61 BAC_Uni_Dmp2P_01_xx

Funktionsbeschreibung

Das Template **BAC_Uni_Dmp2P_01_xx** ist für die Ansteuerung und Überwachung einer Zweipunktklappe, z. B. eine Luftklappe mit einem Federrücklaufantrieb. Es besteht im Wesentlichen aus einem BO-Objekt für die Ansteuerung der Klappe und einem MV-Objekt zur manuellen Übersteuerung. Das Template wird ergänzt durch optionale BACnet Objekte, siehe Tabelle Ausstattungsvarianten.

Unter **IO-Verknüpfung** sind die Variablen zu finden, die mit dem Prozessabbild der Eingangs- und Ausgangsebene in der SPS verknüpft werden.



Die beiden Ausgangsvariablen bSwiOpn / bSwiCls geben nur dann den tatsächlichen Zustand der Klappe aus, wenn in dem eingesetzten Template die Rückmeldung der Endlagenschalter SwiOpn/ SwiCls vorhanden sind. Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang bSwiOpn = TRUE. Bei nicht angesteuerter Klappe wird der Ausgang bSwiCls = TRUE. Dieses kann zur Folge haben, dass in dem Startprogramm einer Lüftungsanlage eine Verzögerungszeit zum Auffahren der Klappe angegeben werden muss.

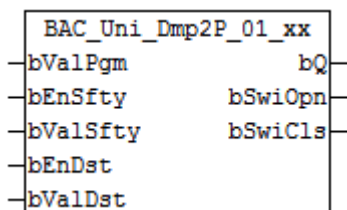
Ausstattungsvarianten

Das Template **BAC_Uni_Dmp2P_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten der Klappen werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

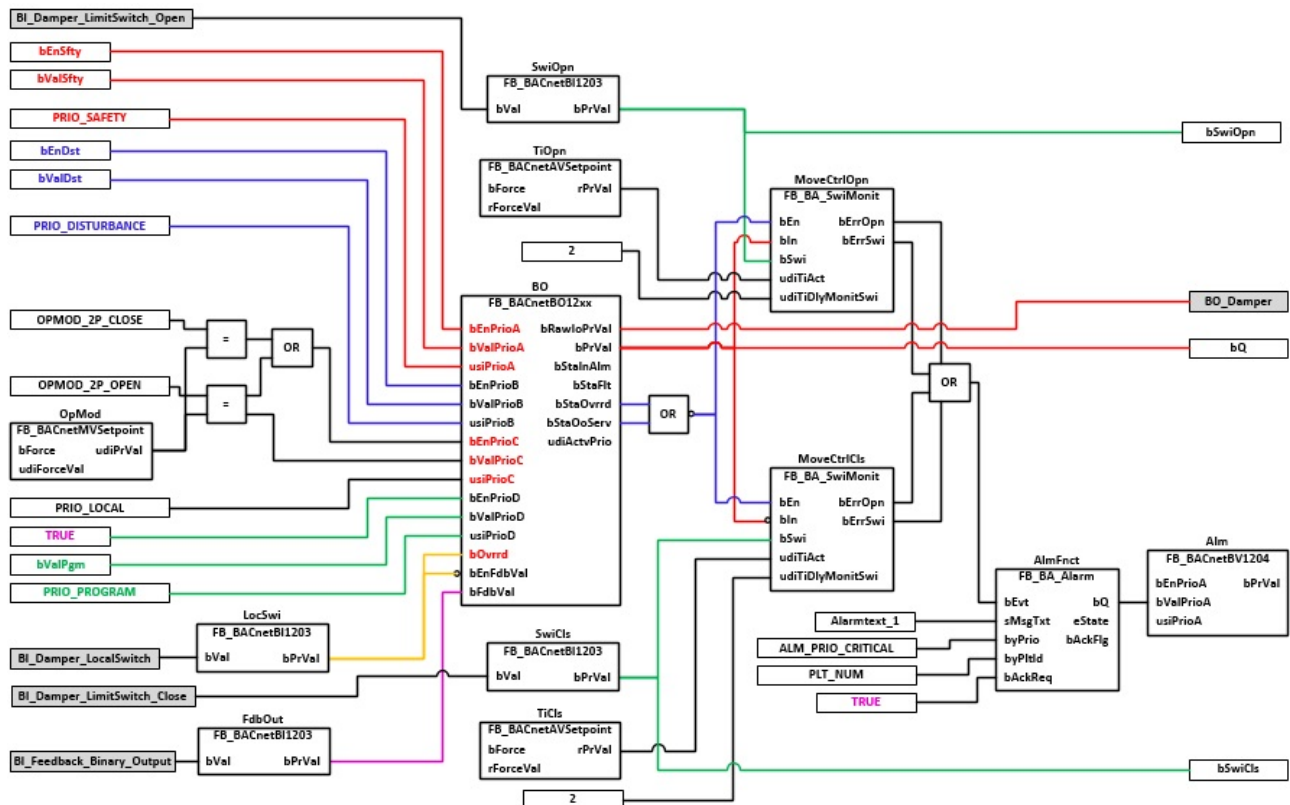
Optionen	mechanische Vorrangbedien- ung Rückmeldung Handscharter (A-0-H)	mechanische Vorrangbedien- ung Rückmeldung Relaisausgang (Rm-Ausg)	Endlage Auf (SchalterAuf)	Endlage Zu (SchalterZu)
Instanz	LocSwi	FdbOut	SwiOpn	SwiCls
Datenpunkt Typ	BI	BI	BI	BI
	8	4	2	1
BAC_Uni_Dmp2P_01_00	0	0	0	0
BAC_Uni_Dmp2P_01_02	0	0	1	0
BAC_Uni_Dmp2P_01_03	0	0	1	1
BAC_Uni_Dmp2P_01_08	1	0	0	0
BAC_Uni_Dmp2P_01_10	1	0	1	0
BAC_Uni_Dmp2P_01_11	1	0	1	1
BAC_Uni_Dmp2P_01_12	1	1	0	0

Optionen	mechanische Vorrangbedienung Rückmeldung Handschalter (A-0-H)	mechanische Vorrangbedienung Rückmeldung Relaisausgang (Rm-Ausg)	Endlage Auf (SchalterAuf)	Endlage Zu (SchalterZu)
Instanz	LocSwi	FdbOut	SwiOpn	SwiCls
Datenpunkt Typ	BI	BI	BI	BI
	8	4	2	1
BAC_Uni_Dmp2P_01_14	1	1	1	0
BAC_Uni_Dmp2P_01_15	1	1	1	1

Schnittstelle



Blockschaltbild Variante BAC_Uni_Dmp2P_01_15



VAR_INPUT

```

bValPgm      : BOOL;
bEnSfty      : BOOL;
bValSfty     : BOOL;
bEnDst      : BOOL;
bValDst     : BOOL;
  
```

bValPgm: Binärer Wert Programmpriorität

bEnfty: Freigabe Sicherheitspriorität

bValSfty: Binärer Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität

bValDst: Binärer Wert Störungspriorität

VAR_OUTPUT

```
bQ      : BOOL;
bSwiOpn : BOOL;
bSwiCls  : BOOL;
```

bQ: Status Ausgang Ansteuerung der Luftklappe

bSwiOpn: Endlage Offen der Klappe ist erreicht. Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang **bSwiOpn** = TRUE. Bei nicht angesteuerter Klappe wird der Ausgang **bSwiCls** = TRUE.

bSwiCls: Endlage Zu der Klappe ist erreicht. Ist keine Endlagenüberwachung vorhanden, so wird diese intern emuliert. Wird die Klappe angesteuert, so wird der Ausgang **bSwiOpn** = TRUE. Bei nicht angesteuerter Klappe wird der Ausgang **bSwiCls** = TRUE.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
SwiOpn	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss des Endlagenschalters Offen
SwiCls	FB_BACnetBI1203 [► 72]	X	BI-Objekt für den Anschluss des Endlagenschalters Zu
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung der Klappe von der MBE oder einem lokalen Bediendisplay
LocSwi	FB_BACnetBI1203 [► 72]	X	BI-Objekt für für die Rückmeldung einer mechanischen Vorrangbedienung. (Hand/Not-Bedienebene)
FdbOut	FB_BACnetBI1203 [► 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Relais
TiOpn	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe des Wertes für die Auffahrzeit

Instanz	Typ	optional	Aufgabe		
TiCls	FB_BACnetAVSetpoint [▶ 70]	X	AV-Objekt zur Eingabe des Wertes für die Zufahrzeit		
MoveCtrlOpen	FB_BA_SwiMonit [▶ 155]	X	Funktionsbaustein welcher die Endlage Auf der Klappe überwacht		
MoveCtrlClis	FB_BA_SwiMonit [▶ 155]	X	Funktionsbaustein welcher die Endlage Zu der Klappe überwacht		
BO	FB_BACnetBO1203 [▶ 82]		BO-Objekt für die Ansteuerung der Klappe		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty
			PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang bValDst
			PRIO_LOCAL (8)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt hat den Wert OPMOD_2P_CLOSE (Klappe schliessen) 2. Das MV-Objekt hat den Wert OPMOD_2P_OPEN (Klappe öffnen)	TRUE, wenn OpMod_udiPrVal = OPMOD_2P_OPEN
PRIO_PROGRAM (15)	TRUE	Eingang bValPgm			
AlmFnc	FB_BA_Alarm [▶ 182]	x	Der Funktionsbaustein AlmFnc erfasst das Ereignis der Überwachung der Endlagenschalter. Aktionen, die nach dem Eingang der Endlagenschalterstörung erfolgen soll, können im Template am Funktionsbaustein AlmFnc parametrisiert werden.		
Alm	FB_BACnetBV1204 [▶ 94]	x	BV-Objekt zur Anzeige der Klappenstörung in der MBE		

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Damper_LimitSwitch_Open	BOOL	X	Eingang	Klappe Endlagenschalter - Meldung - Auf
BI_Damper_LimitSwitch_Close	BOOL	X	Eingang	Klappe Endlagenschalter - Meldung - Zu
BI_Damper_LocalSwitch	BOOL	X	Eingang	Klappe Schalter H-0-A - Meldung - Betriebsart

Parameter	Typ	optional	Prozessabbild	
BI_Feedback_Binary_Output	BOOL	X	Eingang	Klappe Schalter H-0-A - Meldung - Betrieb
BO_Damper	BOOL		Ausgang	Klappe - Schaltbefehl - Auf

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.62 BAC_Uni_Mot1st_01_xx

Funktionsbeschreibung

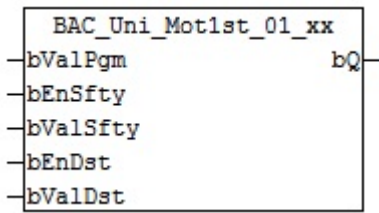
Das Template **BAC_Uni_Mot1st_01_xx** ist für die Ansteuerung eines einstufigen Motors, z. B. eines Ventilators mit binären Ein- und Ausgängen. Es besteht im Wesentlichen aus einem BO-Objekt für die Ansteuerung des Motors und einem MV-Objekt zur manuellen Übersteuerung. Das Template wird ergänzt durch optionale BACnet Objekte, siehe Tabelle Ausstattungsvarianten.

Ausstattungsvarianten

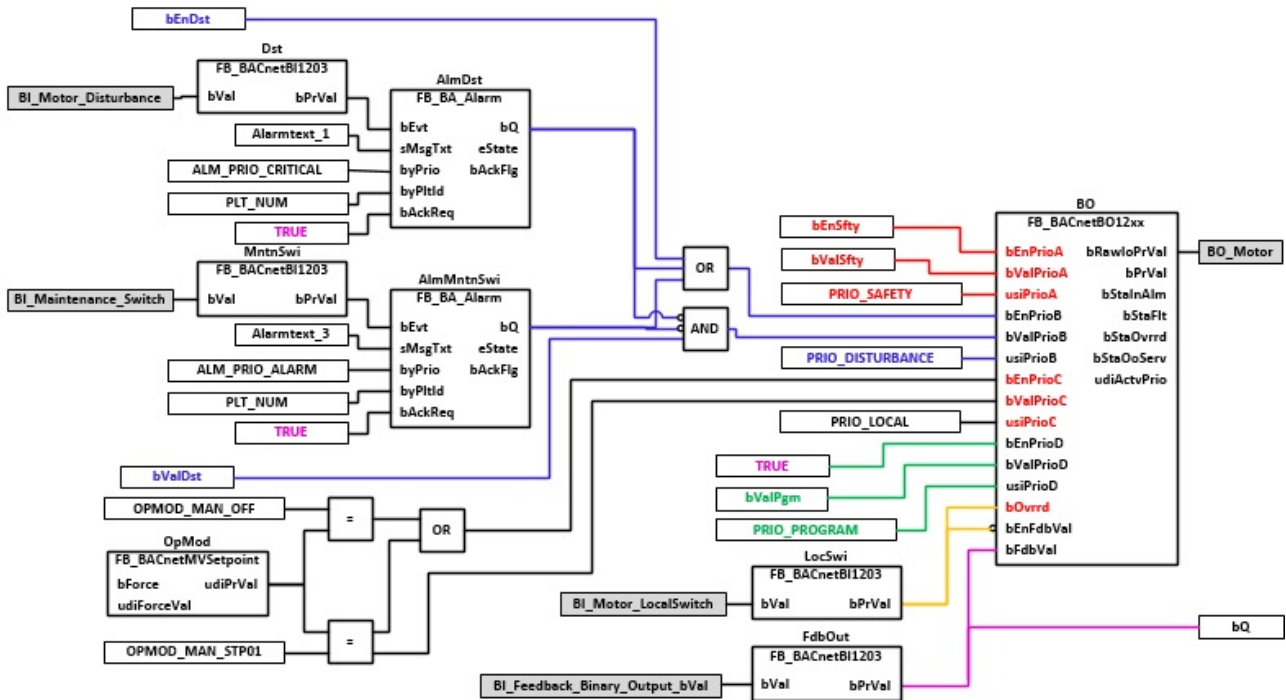
Das Template **BAC_Uni_Mot1st_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten des Templates werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

Optionen	Wartungsschalter (Rep)	mechanische Vorrangbedien- ung Rückmeldung Handscharter (A-0-H)	mechanische Vorrangbedien- ung Rückmeldung Relaisausgang (Rm-Ausg)	Betriebs-rück- meldung (Betr)	Störung (Stör)
Instanz	MntnSwi	LocSwi	FdbOut	Fdb	Dst
Datenpunkt Typ	BI	BI	BI	BI	BI
	16	8	4	2	1
BAC_Uni_Mot1st_01_01	0	0	0	0	1
BAC_Uni_Mot1st_01_03	0	0	0	1	1
BAC_Uni_Mot1st_01_11	0	1	0	1	1
BAC_Uni_Mot1st_01_13	0	1	1	0	1
BAC_Uni_Mot1st_01_17	1	0	0	0	1
BAC_Uni_Mot1st_01_19	1	0	0	1	1
BAC_Uni_Mot1st_01_27	1	1	0	1	1
BAC_Uni_Mot1st_01_29	1	1	1	0	1

Schnittstelle



Blockschaltbild Variante BAC_Uni_Mot1st_01_29



VAR_INPUT

```

bValPgm      : BOOL;
bEnSfty      : BOOL;
bValSfty     : BOOL;
bEnDst       : BOOL;
bValDst      : BOOL;
    
```

bValPgm: Binärer Wert Programmpriorität

bEnSfty: Freigabe Sicherheitspriorität

bValSfty: Binärer Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität. An diesen Eingang könnte z. B. eine Prozessrückmeldung angelegt werden.

bValDst: Binärer Wert Störungspriorität. An diesen Eingang könnte z. B. eine Prozessrückmeldung angelegt werden.

VAR_OUTPUT

```

bQ           : BOOL;
    
```

bQ: Betriebsrückmeldung

VAR CONSTANT

```

PLT_NUM      : BYTE := 1;
    
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmerbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarmer einer Anlage, z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_CmnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe			
Dst	FB_BACnetBI1203 [► 72]	x	BI-Objekt Pumpenstörung			
Fdb	FB_BACnetBI1203 [► 72]	X	BI-Objekt Betriebsrückmeldung			
FdbOut	FB_BACnetBI1203 [► 72]	X	BI-Objekt Rückmeldung mechanische Vorrangbedienung Rückmeldung Relaisausgang			
LocSwi	FB_BACnetBI1203 [► 72]	X	BI-Objekt Rückmeldung mechanische Vorrangbedienung Rückmeldung Handschalter			
MntnSwi	FB_BACnetBI1203 [► 72]	X	BI-Objekt Wartungsschalter			
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung der Pumpe von der MBE oder einem lokalen Bediendisplay			
AlmDst	FB_BA_Alarm [► 182]		Der Funktionsbaustein AlmDst erfasst das Ereignis Störung. Aktionen, die nach dem Eingang der Störung erfolgen sollen, können im Template am Funktionsbaustein AlmDst parametrier werden.			
AlmMntnSwi	FB_BA_Alarm [► 182]	X	Der Funktionsbaustein AlmMntnSwi erfasst das Ereignis Reparaturschalter ausgelöst. Aktionen, die nach dem Eingang Reparaturschalter ausgelöst erfolgen sollen, können im Template am Funktionsbaustein AlmMntnSwi parametrier werden.			
BO	FB_BACnetBO1203 [► 82]		BO-Objekt für die Vorgabe der Ansteuerung des einstufigen Motors			
			Priorität:	Freigabe	Wert	Bemerkung
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty	
PRIO_DISTURBANCE (3)	Eingang bEnDst OR Dst OR MntnSwi	Wert des Operators AND mit den Eingängen bValDst , MntnSwi , bPrVal , Dst_bPrVal				

Instanz	Typ	optional	Aufgabe			
			PRIO_LOCAL (8)	OpMod_udiPrVal = OPMOD_MAN_OFFOR OPMOD_MAN_STP01	TRUE, wenn OpMod_udiPrVal = OPMOD_MAN_STP01	Bei Handbetrieb Wert von AV-Objekt ManSpd
			PRIO_PROGRAM (15)	TRUE	bValPgm	Wert von Eingang bValPgm

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Motor_Disturbance	BOOL		Eingang	Digitaleingang - Motor Störung - Meldung - ausgelöst
BI_Feedback_Binary_Output	BOOL	X	Eingang	Digitaleingang - Motor Schaltbefehl - Rückmeldung - Ein/Aus
BI_Pump_Feedback	BOOL	X	Eingang	Digitaleingang - Motorbetrieb - Meldung - Ein/Aus
BI_Motor_LocalSwitch	BOOL	X	Eingang	Digitaleingang - Schalter Hand Motor - Meldung - Hand/Auto
BI_Maintenance_Switch	BOOL	X	Eingang	Digitaleingang - Wartungsschalter - Meldung - ausgelöst
BO_Motor	BOOL		Ausgang	Digitalausgang - Motor - Schaltbefehl - Ein/Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.63 BAC_Uni_Pu1st_01_xx

Funktionsbeschreibung

Das Template **BAC_Uni_Pu1st_01_xx** ist für die Ansteuerung einer einstufigen Pumpe mit binären Ein- und Ausgängen. Es besteht im Wesentlichen aus einem BO-Objekt für die Ansteuerung der Pumpe und einem MV-Objekt zur manuellen Steuerung.

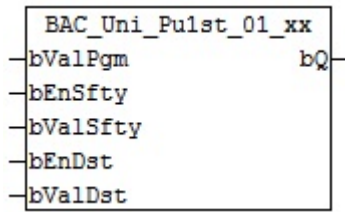
Das Template **BAC_Uni_Pu1st_01_xx** existiert in verschiedenen Ausstattungsvarianten. Innerhalb dieser Varianten gibt es zwei Grundvarianten. Eine Varianten ist mit, die andere ohne einen Eingang für die Positionsmeldung des zugehörigen Regelventils.

Der Eingang **rValve** dient der Einschaltung der Pumpe mittels eines Hystereseomoduls in Abhängigkeit der Ventilstellung. Benötigt wird dieses z. B. für Luftherhitzer in raumlufttechnischen Anlagen.

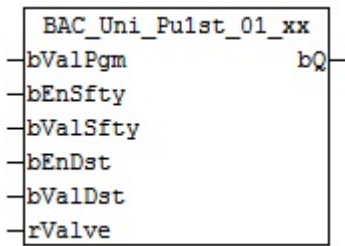
Optionen	Ventil- stellung (Valve)	Reserve	War- tungs- schalter (Rep)	Blockier- schutz (Block)	mechani- sche Vorrang- bedie- nung Rückmel- dung Hand- schalter (A-0-H)	mechani- sche Vorrang- bedie- nung Rückmel- dung Relais- ausgang (Rm- Ausg)	Betriebs- rückmel- dung (Betr)	Stör- meldung (Stör)
Instanz- name	rValve	-	MntnSwi	AblkFnct	LocSwi	FdbOut	Fdb	Dst
Daten- punkt Typ	-	BI	BI	-	BI	BI	BI	BI
	128	64	32	16	8	4	2	1
BAC_Uni_Pu1St_01_017	0	0	0	1	0	0	0	1
BAC_Uni_Pu1St_01_019	0	0	0	1	0	0	1	1
BAC_Uni_Pu1St_01_027	0	0	0	1	1	0	1	1
BAC_Uni_Pu1St_01_029	0	0	0	1	1	1	0	1
BAC_Uni_Pu1St_01_059	0	0	1	1	1	0	1	1
BAC_Uni_Pu1St_01_061	0	0	1	1	1	1	0	1
BAC_Uni_Pu1St_01_145	1	0	0	1	0	0	0	1
BAC_Uni_Pu1St_01_147	1	0	0	1	0	0	1	1
BAC_Uni_Pu1St_01_155	1	0	0	1	1	0	1	1
BAC_Uni_Pu1St_01_157	1	0	0	1	1	1	0	1
BAC_Uni_Pu1St_01_187	1	0	1	1	1	0	1	1
BAC_Uni_Pu1St_01_189	1	0	1	1	1	1	0	1

Schnittstelle1

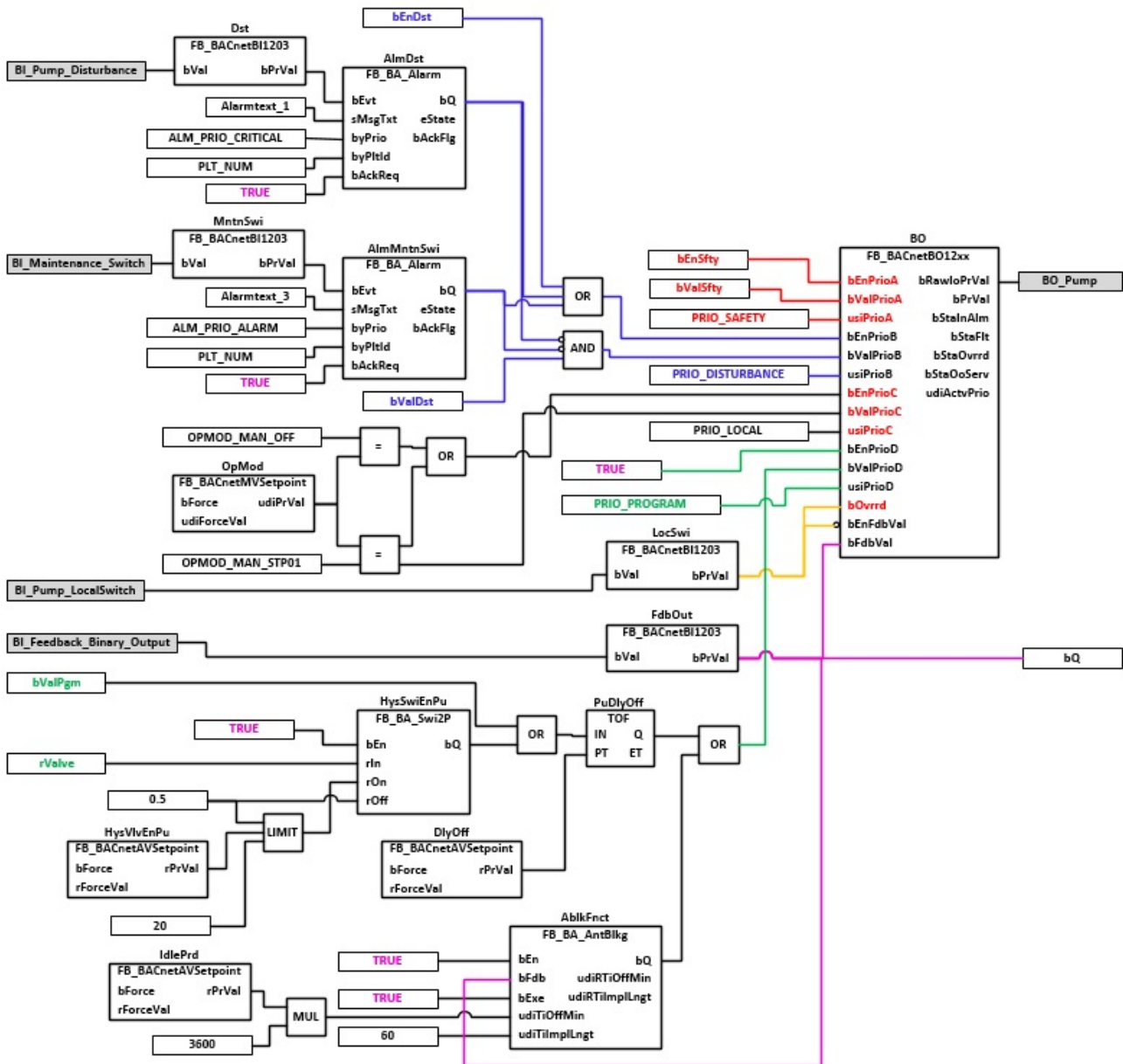
Ohne Eingang rValve



Mit Eingang rValve



Blockschaltbild Variante BAC_Uni_Pu1st_01_189



VAR_INPUT

```

bValPgm      : BOOL;
bEnSfty      : BOOL;
bValSfty     : BOOL;
bEnDst       : BOOL;
bValDst      : BOOL;
rValve       : REAL;

```

bValPgm: Binärer Wert Programmpriorität

bEnSfty: Freigabe Sicherheitspriorität

bValSfty: Binärer Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität. An diesen Eingang könnte z. B. eine Prozessrückmeldung angelegt werden.

bValDst: Binärer Wert Störungspriorität. An diesen Eingang könnte z. B. eine Prozessrückmeldung angelegt werden.

rValve: Eingang an dem die Ventilstellung angeschlossen wird.

VAR_OUTPUT

bQ : BOOL;

bQ: Pumpe eingeschaltet

VAR CONSTANT

PLT_NUM : BYTE := 1;

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm. \[▶ 182\]](#)

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01 \[▶ 372\]](#) mittels des Funktionsbausteins [FB BA AlarmPlt. \[▶ 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg \[▶ 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
Dst	FB BACnetBI1203 [▶ 72]	x	BI-Objekt Pumpenstörung
Fdb	FB BACnetBI1203 [▶ 72]	X	BI-Objekt Betriebsrückmeldung Pumpe
FdbOut	FB BACnetBI1203 [▶ 72]	X	BI-Objekt Rückmeldung mechanische Vorrangbedienung Rückmeldung Relaisausgang
LocSwi	FB BACnetBI1203 [▶ 72]	X	BI-Objekt Rückmeldung mechanische Vorrangbedienung Rückmeldung Handschalter
MntnSwi	FB BACnetBI1203 [▶ 72]	X	BI-Objekt Wartungsschalter
HysVlvEnPu	FB BACnetAVSetpoint [▶ 70]	x	AV-Objekt zur Eingabe des Hysteresewertes für den 2 Punktschalter HysSwiEnPu um die Pumpe über die Ventilstellung rValve zu schalten. Wird der Wert 0 eingegeben, so wird die Pumpe nicht über die Ventilstellung eingeschaltet.
DlyOff	FB BACnetAVSetpoint [▶ 70]		AV-Objekt zur Eingabe des Wertes Nachlaufzeit.
IdlePrd	FB BACnetAVSetpoint [▶ 70]		AV-Objekt zur Eingabe der maximalen Dauer einer Pumpenstillstandsperiode bis zur Ausgabe eines Blockierschutzimpulses.
OpMod	FB BACnetMVSetpoint [▶ 131]		MV-Objekt zur manuellen Steuerung der Pumpe von der MBE oder einem lokalen Bediendisplay
AlmDst	FB BA Alarm [▶ 182]		Der Funktionsbaustein AlmDst erfasst das Ereignis Pumpenstörung. Aktionen, die nach dem Eingang der Pumpenstörung erfolgen sollen, können im Template am Funktionsbaustein AlmDst parametrieret werden.

Instanz	Typ	optional	Aufgabe												
AlmMntnS wi	FB_BA_Alarm [▶ 182]	X	Der Funktionsbaustein AlmMntnSwi erfasst das Ereignis Reparaturschalter ausgelöst. Aktionen, die nach dem Eingang Reparaturschalter ausgelöst erfolgen sollen, können im Template am Funktionsbaustein AlmMntnSwi parametrieren werden.												
BO	FB_BACnetBO1203 [▶ 82]		BO-Objekt für die Ansteuerung der Pumpe												
			<table border="1"> <thead> <tr> <th>Priorität:</th> <th>Freigabe</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td>PRIO_SAFETY (1)</td> <td>Eingang bEnSfty</td> <td>Eingang bValSfty</td> </tr> <tr> <td>PRIO_DISTURBAN CE (3)</td> <td>Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Störung (Disturbance) des nachgeschalteten BO-Objekts aktivieren. Ereignisse: 1. Motorstörung vom Funktionsbaustein AlmDst 2. Eingang Template bEnDst 3. Reparaturschalter vom Funktionsbaustein AlmMntnSwi</td> <td>Am Eingang des Templates bEnDst lässt sich die Pumpe zwangsweise Ein oder Ausschalten, z. B. bei Frostgefahr am Luftherhitzer. Das zwangsweise Einschalten ist jedoch am AND-Modul verriegelt bei vorhandener Pumpenstörung oder ausgelöstem Reparaturschalter.</td> </tr> <tr> <td>PRIO_LOCAL (8)</td> <td>Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt hat den Wert OPMOD_MANN_OFF (Hand Aus) 2. Das MV-Objekt hat den</td> <td>TRUE, wenn OpMod_udiPrVal = OPMOD_MAN_STP01</td> </tr> </tbody> </table>	Priorität:	Freigabe	Wert	PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty	PRIO_DISTURBAN CE (3)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Störung (Disturbance) des nachgeschalteten BO-Objekts aktivieren. Ereignisse: 1. Motorstörung vom Funktionsbaustein AlmDst 2. Eingang Template bEnDst 3. Reparaturschalter vom Funktionsbaustein AlmMntnSwi	Am Eingang des Templates bEnDst lässt sich die Pumpe zwangsweise Ein oder Ausschalten, z. B. bei Frostgefahr am Luftherhitzer. Das zwangsweise Einschalten ist jedoch am AND-Modul verriegelt bei vorhandener Pumpenstörung oder ausgelöstem Reparaturschalter.	PRIO_LOCAL (8)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt hat den Wert OPMOD_MANN_OFF (Hand Aus) 2. Das MV-Objekt hat den	TRUE, wenn OpMod_udiPrVal = OPMOD_MAN_STP01
Priorität:	Freigabe	Wert													
PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty													
PRIO_DISTURBAN CE (3)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Störung (Disturbance) des nachgeschalteten BO-Objekts aktivieren. Ereignisse: 1. Motorstörung vom Funktionsbaustein AlmDst 2. Eingang Template bEnDst 3. Reparaturschalter vom Funktionsbaustein AlmMntnSwi	Am Eingang des Templates bEnDst lässt sich die Pumpe zwangsweise Ein oder Ausschalten, z. B. bei Frostgefahr am Luftherhitzer. Das zwangsweise Einschalten ist jedoch am AND-Modul verriegelt bei vorhandener Pumpenstörung oder ausgelöstem Reparaturschalter.													
PRIO_LOCAL (8)	Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt hat den Wert OPMOD_MANN_OFF (Hand Aus) 2. Das MV-Objekt hat den	TRUE, wenn OpMod_udiPrVal = OPMOD_MAN_STP01													

Instanz	Typ	optional	Aufgabe	
				Wert OPMOD_MANN_STP01 (Hand Ein)
			PRIO_PROGRAM (15)	TRUE Das OR-Modul bündelt Ereignisse welche das Schreiben auf die Priorität Pgm (Programm) des nachgeschalteten BO-Objekts aktivieren. Ereignisse: 1. Externe Anforderung der Pumpe von den Eingängen bValPgm oder rValve . Es kann aber auch die Pumpennachlaufzeit PuDlyOff noch aktiv sein. 2. Blockierschutzfunktion vom Funktionsbaustein AblkFnc aktiv
HysSwiEnPu	FB_BA_Swi2P [► 148]	x	2 Punktschalter der anhand einer Hysterese HysVivEnPu die Pumpe in Abhängigkeit der Ventilstellung rValve ein- und ausschaltet. Ist die Ventilstellung < 0.5, so wird die Pumpe ausgeschaltet.	
PuDlyOff	TOF		Zeitglied für Pumpennachlauf	
AblkFnc	FB_BA_AntBlkg [► 218]		Funktionsbaustein zur Ausgabe eines Blockierschutzimpulses	

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Pump_Disturbance	BOOL		Eingang	Pumpe - Anzeige - Störung
BI_Feedback_Binary_Output	BOOL	X	Eingang	Pumpe Schalter H-0-A - Meldung - Betrieb
BI_Pump_Feedback	BOOL	X	Eingang	Pumpe - Meldung - Betrieb
BI_Pump_LocalSwitch	BOOL	X	Ausgang	Pumpe Schalter H-0-A - Meldung - Betriebsart
BI_Maintenance_Switch	BOOL	X	Eingang	Pumpe - Anzeige - Wartungsschalter
BO_Pump	BOOL		Ausgang	Pumpe - Schaltbefehl - Ein

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.64 BAC_Uni_SmokeDetc_001

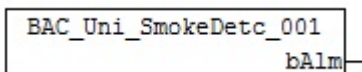
Funktionsbeschreibung

Das Template **BAC_Uni_SmokeDetc_001** dient der Ansteuerung und Überwachung eines Kanalrauchmelders.

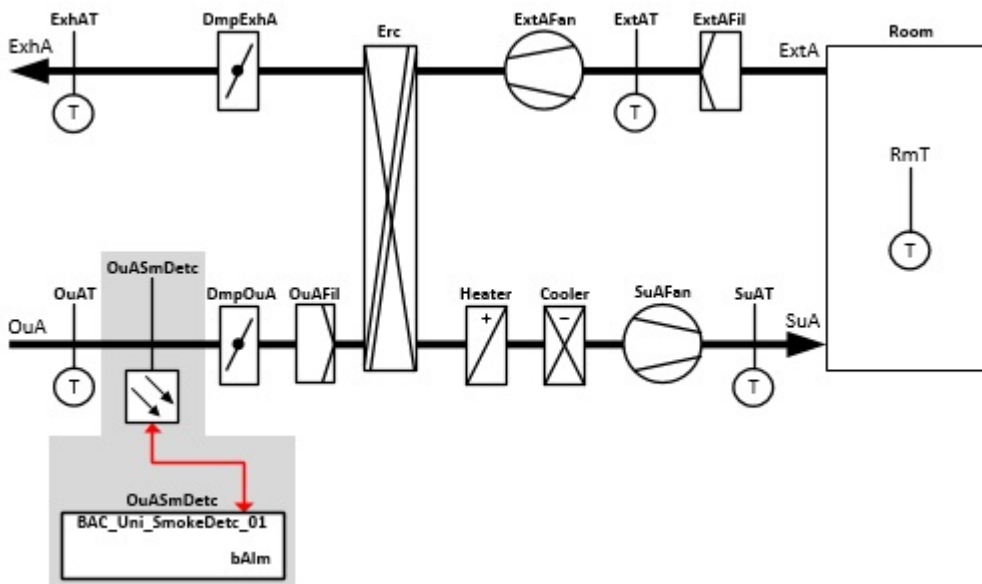
Es besteht aus zwei Binary Input Objekten für Verschmutzung und Rauchalarm und einem Binary Output Objekt für das Zurücksetzen des Kanalrauchmelders.

Für das Quittieren des Kanalrauchmelders ist das BACnet Property **MinimumOntime** des BO-Objekts auf einen Defaultwert eingestellt. Durch diese Zeiteinstellung wird der Quittierungsimpuls verlängert.

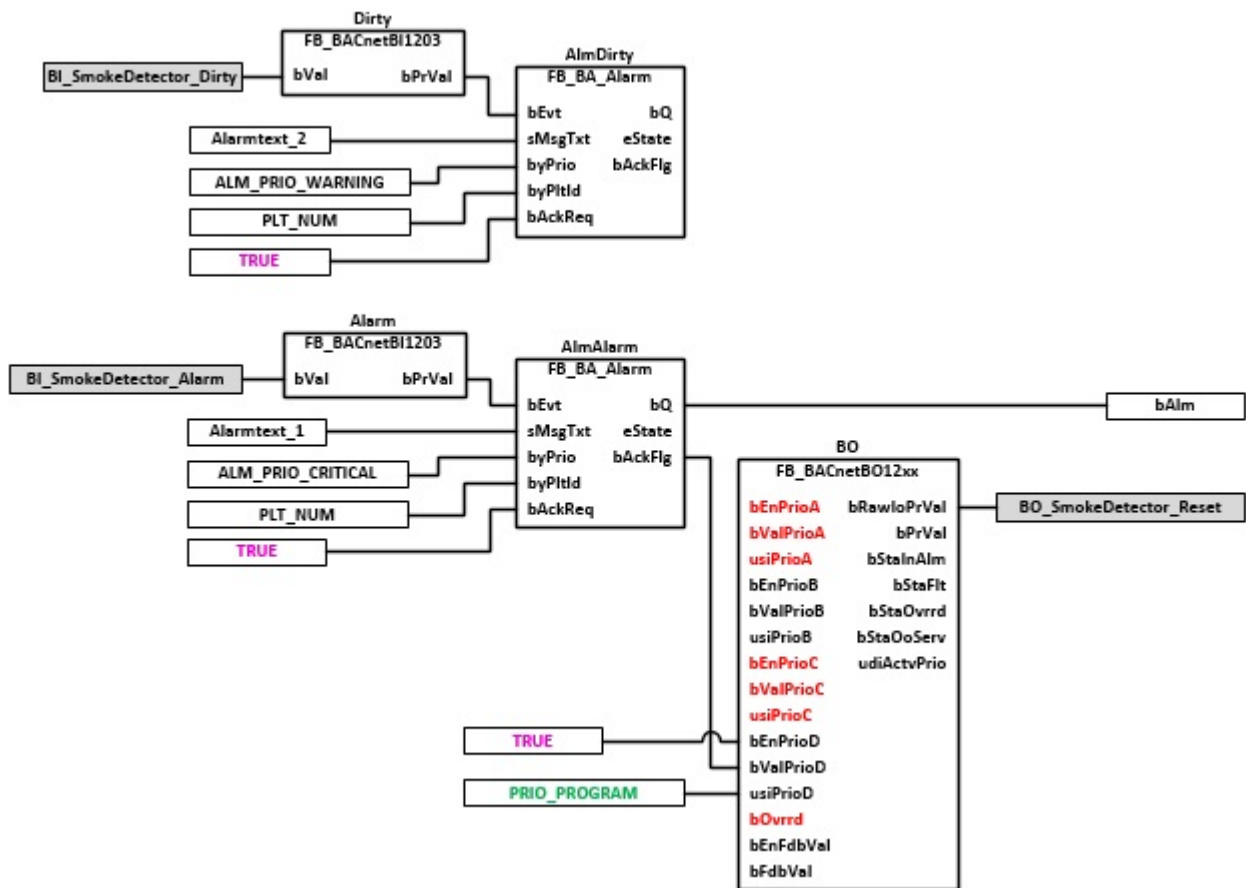
Schnittstelle



Anlagenbild



Blockschaltbild



VAR_OUTPUT

`bAlm` : BOOL;

bAlm: Der Ausgang zeigt an, dass der Rauchmelder ausgelöst hat.

VAR CONSTANT

`PLT_NUM` : BYTE := 1;

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt. Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins `FB_BA_Alarm`. [► 182]
 Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates `BAC_PltAlm_01` [► 372] mittels des Funktionsbausteins `FB_BA_AlarmPlt`. [► 186]
 Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates `BAC_PltComnMsg_01` durch den Funktionsbaustein `FB_BA_CmnMsg` [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe		
Dirty	FB BACnetBI1203 [▶ 72]		BI-Objekt Rauchmelder verschmutzt		
AlmDirty	FB BA Alarm [▶ 182]		Der Funktionsbaustein AlmDirty erfasst das Ereignis Rauchmelder verschmutzt. Aktionen, die nach dem Eingang Rauchmelder verschmutzt erfolgen sollen, können im Template am Funktionsbaustein AlmDirty parametrierbar werden.		
Alarm	FB BACnetBI1203 [▶ 72]		BI-Objekt Rauchmelder Meldung Alarm		
AlmAlarm	FB BA Alarm [▶ 182]		Der Funktionsbaustein AlmAlarm erfasst das Ereignis Rauchmelder Meldung Alarm. Aktionen, die nach dem Eingang Rauchmelder Meldung Alarm erfolgen sollen, können im Template am Funktionsbaustein AlmAlarm parametrierbar werden.		
BO	FB BACnetBO1203 [▶ 82]		BO-Objekt für den Quittierimpuls des Kanalrauchmelders.		
			Priorität:	Freigabe	Wert
			PRIO_PROGR AM (15)	TRUE	Über die Alarmaufnahme AlmAlarm wird der Anlagen Quittierimpuls an das BO-Objekt durchgereicht.

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_SmokeDetector_Dirty	BOOL		Eingang	Digitaleingang - Rauchmelder - Meldung - Verschmutzt
BI_SmokeDetector_Alarm	BOOL		Eingang	Digitaleingang - Rauchmelder - Meldung - Alarm
BO_SmokeDetector_Reset	BOOL		Ausgang	Digitalausgang - Rauchmelder - Schaltbefehl - Quittierung Ein/Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.65 BAC_Cont4Stp_01

Funktionsbeschreibung

Das Template ermittelt abhängig vom Eingangssignal die resultierenden Schaltstufen eines mehrstufigen Aggregats.

Es sind vier Einschaltsschwellen und eine Hysterese parametrierbar.

Schnittstelle

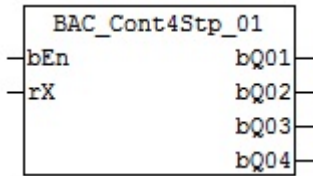
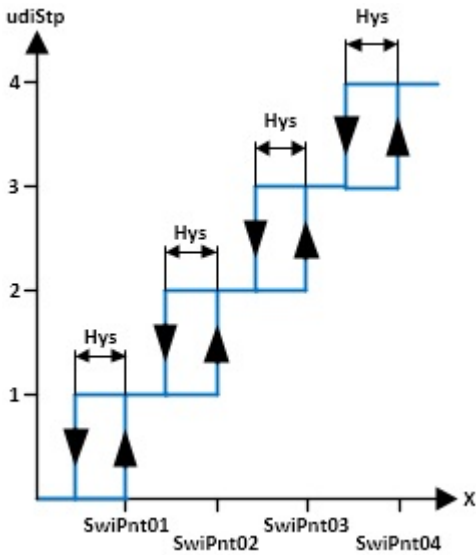


Diagramm 01

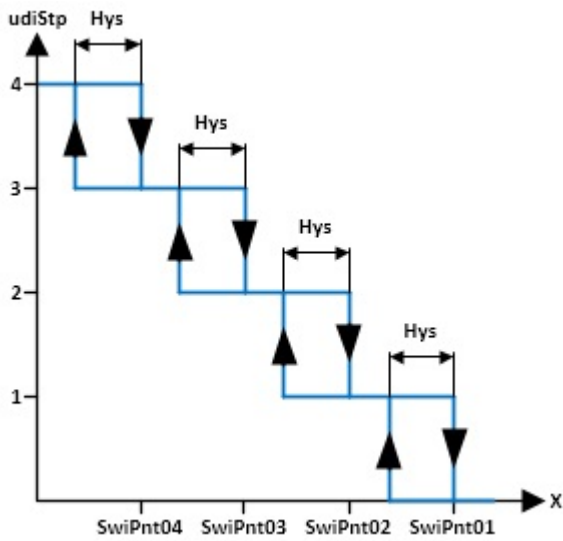
Wirksinn *Cont4Stp_bActn* = FALSE = Reverse = Heizen



DspStp	Cont4Stp_udiStp	SwiOn	SwiOff	Cont4Stp_udiRemTiDyOn	Cont4Stp_udiRemTiDyOff	bQ01	bQ02	bQ03	bQ04
0	0	SwiPnt01	SwiPnt01 - Hys	DlyOn	0	FALSE	FALSE	FALSE	FALSE
1	>= 1	SwiPnt02	SwiPnt01 - Hys	DlyOn	DlyOff	TRUE	FALSE	FALSE	FALSE
2	>= 2	SwiPnt03	SwiPnt02 - Hys	DlyOn	DlyOff	TRUE	TRUE	FALSE	FALSE
3	>= 3	SwiPnt04	SwiPnt03 - Hys	DlyOn	DlyOff	TRUE	TRUE	TRUE	FALSE
4	>= 4	SwiPnt04	SwiPnt04 - Hys	0	DlyOff	TRUE	TRUE	TRUE	TRUE

Diagramm 02

Wirksinn *Cont4Stp_bActn* = TRUE = Direct = Kühlen



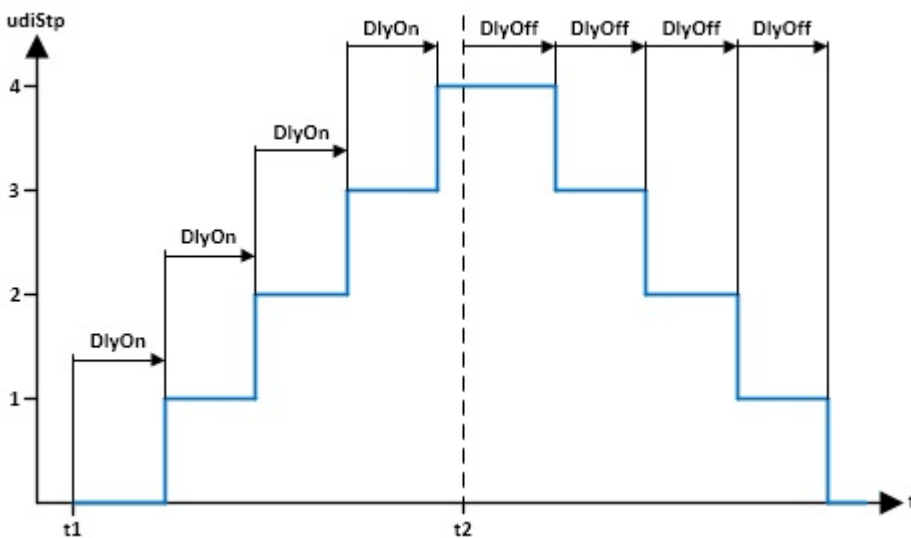
DspStp	Cont4St p_udiSt p	SwiOn	SwiOff	Cont4St p_udiRe mTiDyIO n	Cont4St p_udiRe mTiDyIO ff	bQ01	bQ02	bQ03	bQ04
0	0	SwiPnt01	SwiPnt01 + Hys	DlyOn	0	FALSE	FALSE	FALSE	FALSE
1	>= 1	SwiPnt02	SwiPnt01 +Hys	DlyOn	DlyOff	TRUE	FALSE	FALSE	FALSE
2	>= 2	SwiPnt03	SwiPnt02 + Hys	DlyOn	DlyOff	TRUE	TRUE	FALSE	FALSE
3	>= 3	SwiPnt04	SwiPnt03 + Hys	DlyOn	DlyOff	TRUE	TRUE	TRUE	FALSE
4	4	SwiPnt04	SwiPnt04 + Hys	0	DlyOff	TRUE	TRUE	TRUE	TRUE

Diagramm 03

Zeitverhalten der Ein- und Ausschaltverzögerungen

Zum Zeitpunkt t1 springt rX von £SwiPnt01 auf SwiPnt04

Zum Zeitpunkt t2 springt rX von SwiPnt04 auf £SwiPnt01 – Hys



Ein-/Ausgänge

VAR_INPUT

```
bEn      : BOOL;
rX       : REAL;
```

bEn: Generelle Freigabe des Funktionsbausteins. Ist *bEn* = FALSE, so sind sämtliche Ausgänge auf 0 gesetzt.

rX: Eingangswert von dem der Schaltzustand abgeleitet wird.

VAR_OUTPUT

```
bQ01     : BOOL;
bQ02     : BOOL;
bQ03     : BOOL;
bQ04     : BOOL;
```

bQ01: Anzeige des Status Stufe 01
TRUE = EIN; FALSE = AUS
udiStp >= 1

bQ02: Anzeige des Status Stufe 02
TRUE = EIN; FALSE = AUS
udiStp >= 2

bQ03: Anzeige des Status Stufe 03
TRUE = EIN; FALSE = AUS
udiStp >= 3

bQ04: Anzeige des Status Stufe 04
TRUE = EIN; FALSE = AUS
udiStp >= 4

VAR CONSTANT

```
PLT_NUM  : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer *PLT_NUM* festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [[182](#)]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [[372](#)] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
En	FB_BACnetBV1204 [94]	Generelle Freigabe des Funktionsbausteins. Ist <i>bEn</i> = FALSE, so sind sämtliche Ausgänge auf 0 gesetzt
X	FB_BACnetAVSetpoint [70]	Eingangswert von dem der Schaltzustand abgeleitet wird
Hys	FB_BACnetAVSetpoint [70]	Eingabe Absolutwert Hysterese
DlyOn	FB_BACnetAVSetpoint [70]	Einschaltverzögerung der Stufen in Sekunden
DlyOff	FB_BACnetAVSetpoint [70]	Ausschaltverzögerung der Stufen in Sekunden

Instanz	Typ	Aufgabe
SwiPnt01	FB_BACnetAVSetpoint [▶ 70]	Eingabe Einschaltpunkt Stufe 01, intern im Template limitiert zwischen 0 und 100
SwiPnt02	FB_BACnetAVSetpoint [▶ 70]	Eingabe Einschaltpunkt Stufe 02, intern im Template limitiert zwischen 0 und 100
SwiPnt03	FB_BACnetAVSetpoint [▶ 70]	Eingabe Einschaltpunkt Stufe 03, intern im Template limitiert zwischen 0 und 100
SwiPnt04	FB_BACnetAVSetpoint [▶ 70]	Eingabe Einschaltpunkt Stufe 04, intern im Template limitiert zwischen 0 und 100
Cont4Stp	FB_BA_Cont4Stp01 [▶ 139]	Der Funktionsbaustein Cont4Stp ist das Kernstück des Templates Stufenschalter.
DspStp	FB_BACnetMVDDisplay [▶ 130]	Anzeige aktive Stufe
SwiOn	FB_BACnetAVDisplay [▶ 69]	Anzeige aktiver Einschaltpunkt
SwiOff	FB_BACnetAVDisplay [▶ 69]	Anzeige aktiver Ausschaltpunkt
	LIMIT, LIMIT, LIMIT, LIMIT	Limitierung der Einschaltpunkte Stufe 1 - 4 auf 0 - 100

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

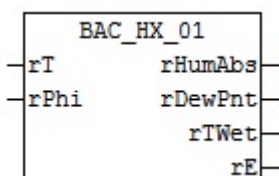
9.66 BAC_HX_01

Funktionsbeschreibung

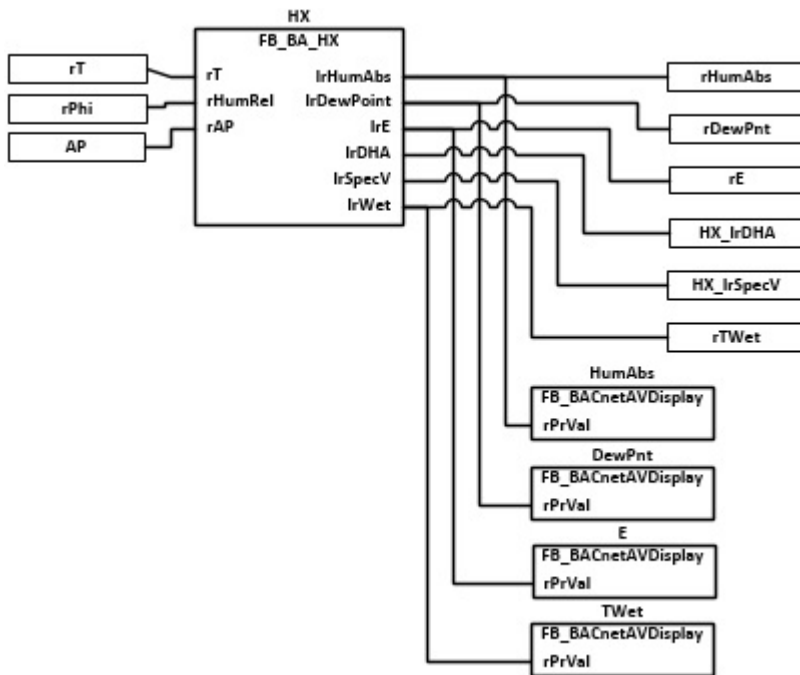
In diesem Template werden die **Taupunkttemperatur**, die **spezifische Enthalpie**, die **Feuchtkugeltemperatur** und die **absolute Feuchte** berechnet.

Für die Berechnung der Größen werden die Temperatur, die relative Feuchte und der barometrische Luftdruck benötigt.

Schnittstelle



Blockschaltbild



VAR_INPUT

```
rT      : REAL;
rPhi    : REAL;
```

rT: Aktueller Wert Temperatur [°C]

rPhi: Aktueller Wert der relativen Feuchte [%]

VAR_OUTPUT

```
rHumAbs : REAL;
rDewPnt : REAL;
rTWet   : REAL;
rE       : REAL;
```

rHumAbs: Berechneter Wert **Absoluten Feuchte g Wasser je kg trockener Luft [g/Kg]**

rDewPnt: Berechneter Wert **Taupunkttemperatur [°C]**

rTWet: Berechneter Wert **Feuchtkugeltemperatur [°C]**

rE: Berechneter Wert **Enthalpie [kJ/kg]**

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA Alarm**. [▶ 182]

Die Auswertung der Alarmergebnisse einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [▶ 372] mittels des Funktionsbausteins **FB_BA AlarmPlt**. [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA ComnMsg** [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
HX	FB BA HX [▶ 230]		Mit diesem Funktionsbaustein werden die Taupunkttemperatur, die spezifische Enthalpie, die Feuchtkugeltemperatur und die absolute Feuchte berechnet. Für die Berechnung der Größen werden die Temperatur, die relative Feuchte und der barometrische Luftdruck benötigt.
HumAbs	FB BACnetAVDisplay [▶ 69]		Anzeige des berechneten Wertes Absolute Feuchte g Wasser je kg trockener Luft [g/Kg]
DewPnt	FB BACnetAVDisplay [▶ 69]		Anzeige des berechneten Wertes Taupunkttemperatur [°C]
TWet	FB BACnetAVDisplay [▶ 69]		Anzeige des berechneten Wertes Feuchtkugeltemperatur [°C]
E	FB BACnetAVDisplay [▶ 69]		Anzeige des berechneten Wertes Enthalpie [kJ/kg]
TLogHum Abs	FB BACnetTLog1201 [▶ 137]		Trendaufzeichnung des AV-Objekts HumAbs Absolute Feuchte

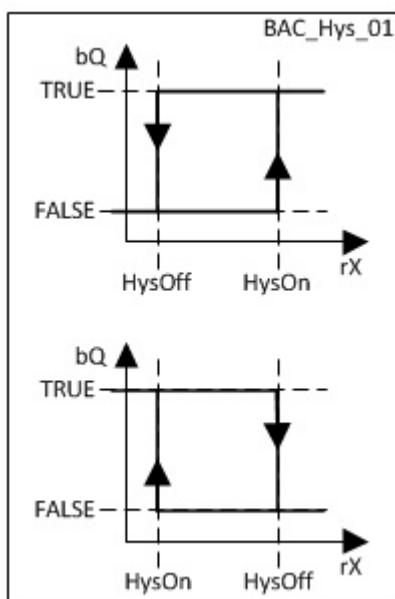
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

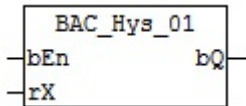
9.67 BAC_Hys_01

Funktionsbeschreibung

Das Template stellt eine Hysterese-Funktion mit festen Schaltepunkten dar.



Schnittstelle



VAR_INPUT

bEn : BOOL;
rX : REAL;

bEn: Freigabe

rX: Istwert

VAR_OUTPUT

bQ : BOOL;

bQ: Ausgabe aktueller Zustand der Hysterese-Funktion

Programmbeschreibung

Instanz	Typ	Aufgabe
En	FB BACnetBV1204 [▶ 94]	Das BV-Objekt dient zur Anzeige der Freigabe Hysterese Funktion
X	FB BACnetAVDisplay [▶ 69]	Das AV-Objekt zeigt den Istwert an, der über die Eingangsvariable rX an das Template angebunden wird
HysOn	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des oberen Grenzwertes der Hysterese-Funktion
HysOff	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des unteren Grenzwertes der Hysterese-Funktion
Hys	FB BA_Swi2P [▶ 148]	Der Funktionsbaustein Hys ist das Kernstück der Hysterese-Funktion
Q	FB BACnetBVDisplay [▶ 96]	Das BV-Objekt zeigt den aktuellen Zustand der Hysterese-Funktion

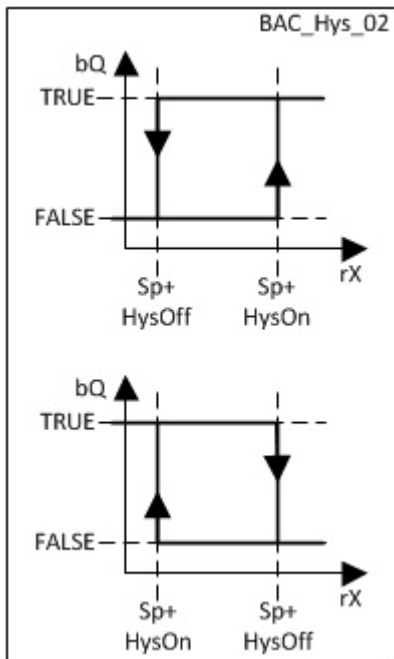
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

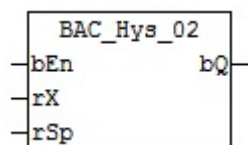
9.68 BAC_Hys_02

Funktionsbeschreibung

Das Template stellt eine Hysterese-Funktion mit dynamischen Schaltpunkten dar.



Schnittstelle



VAR_INPUT

bEn : BOOL;
 rX : REAL;

bEn: Freigabe

rX: Istwert

rSp: Sollwert

VAR_OUTPUT

bQ : BOOL;

bQ: Ausgabe aktueller Zustand der Hysterese-Funktion

Programmbeschreibung

Instanz	Typ	Aufgabe
En	FB BACnetBV1204 [▶ 94]	Das BV-Objekt dient zur Anzeige der Freigabe Hysterese Funktion
X	FB BACnetAVDisplay [▶ 69]	Das AV- Objekt zeigt den Istwert an, der über die Eingangsvariable rX an das Template angebunden wird
Sp	FB BACnetAVDisplay [▶ 69]	Das AV- Objekt zeigt den Sollwert an, der über die Eingangsvariable rSp an das Template angebunden wird
HysOn	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des oberen Grenzwertes der Hysterese-Funktion
HysOff	FB BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe des unteren Grenzwertes der Hysterese-Funktion

Instanz	Typ	Aufgabe
Hys	FB BA Swi2P [▶_148]	Der Funktionsbaustein Hys ist das Kernstück der Hysterese-Funktion
Q	FB_BACnetBVDisplay [▶_96]	Das BV-Objekt zeigt den aktuellen Zustand der Hysterese-Funktion an

Versionshistorie

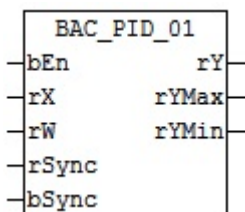
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.69 BAC_PID_01

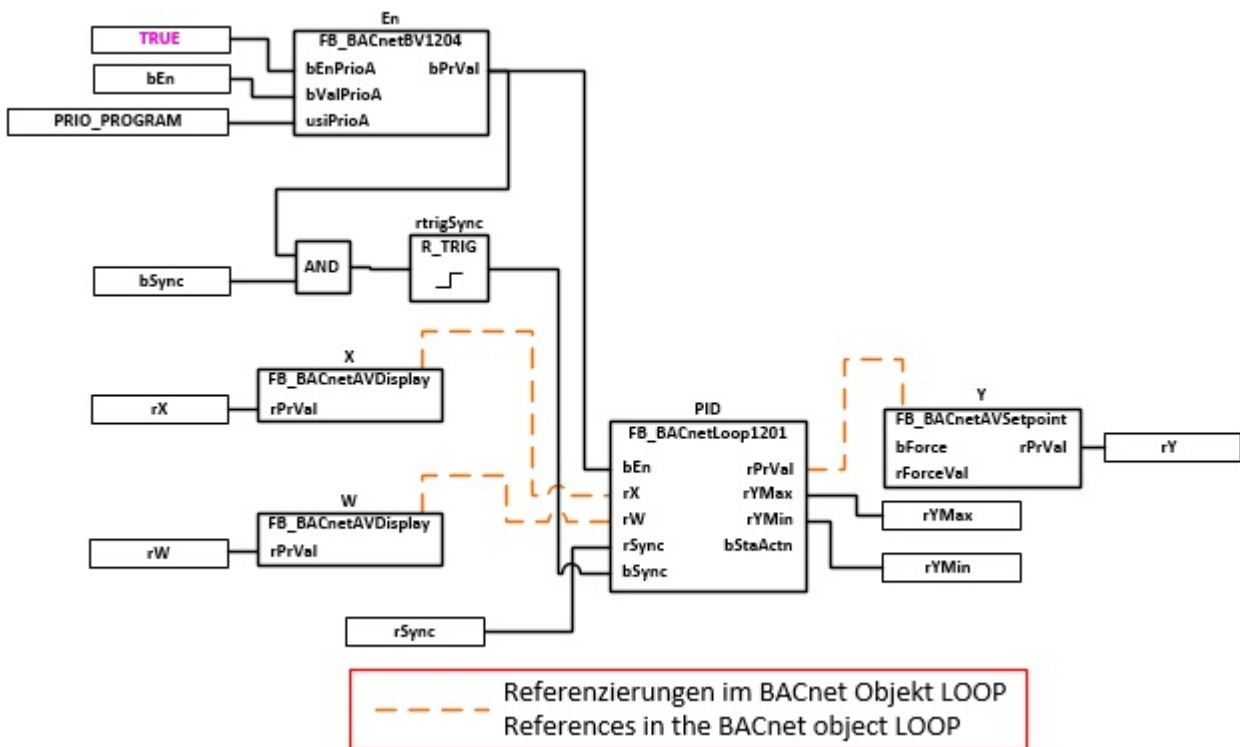
Funktionsbeschreibung

Das Template BAC_PID_01 ist ein universeller PID-Regler.
 Die Referenzierung des Sollwertes, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte X, W und Y.
 Die Freigabe des PID-Reglers erfolgt anhand der Eingangsvariable **bEn**.

Schnittstelle



Blockschaltbild



VAR_INPUT

```
bEn   : BOOL;
rX    : REAL;
rW    : REAL;
rSync : REAL;
bSync : BOOL;
```

bEn: Freigabe**rX:** Istwert**rW:** Sollwert**rSync:** Synchronisationswert**bSync:** Start Synchronisation**VAR_OUTPUT**

```
rY      : REAL;
rYMax   : REAL;
rYMin   : REAL;
```

rY: Ausgabe der Stellgröße**rYMax:** Maximalwertes Reglers**rYMin:** Minimalwert Reglers**VAR CONSTANT**

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [[▶ 182](#)]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [[▶ 372](#)] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
X	FB_BACnetAVDisplay [▶ 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts
W	FB_BACnetAVDisplay [▶ 69]	Das AV-Objekt zeigt den Sollwert des Eingangs rW an und ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts
En	FB_BACnetBV1204 [▶ 94]	Das BV-Objekt dient zur Anzeige der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay
PID	FB_BACnetLoop1201 [▶ 99]	PID-Regler
rtrigSync	R_TRIG	Bei einer steigenden Flanke am Eingang bSync wird das Loop-Objekt auf den Wert von rSync aufsynchronisiert.

Instanz	Typ	Aufgabe
Y	FB_BACnetAVSetpoint [▶_70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts

Versionshistorie

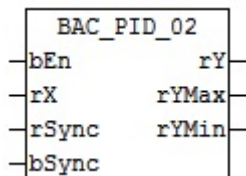
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.70 BAC_PID_02

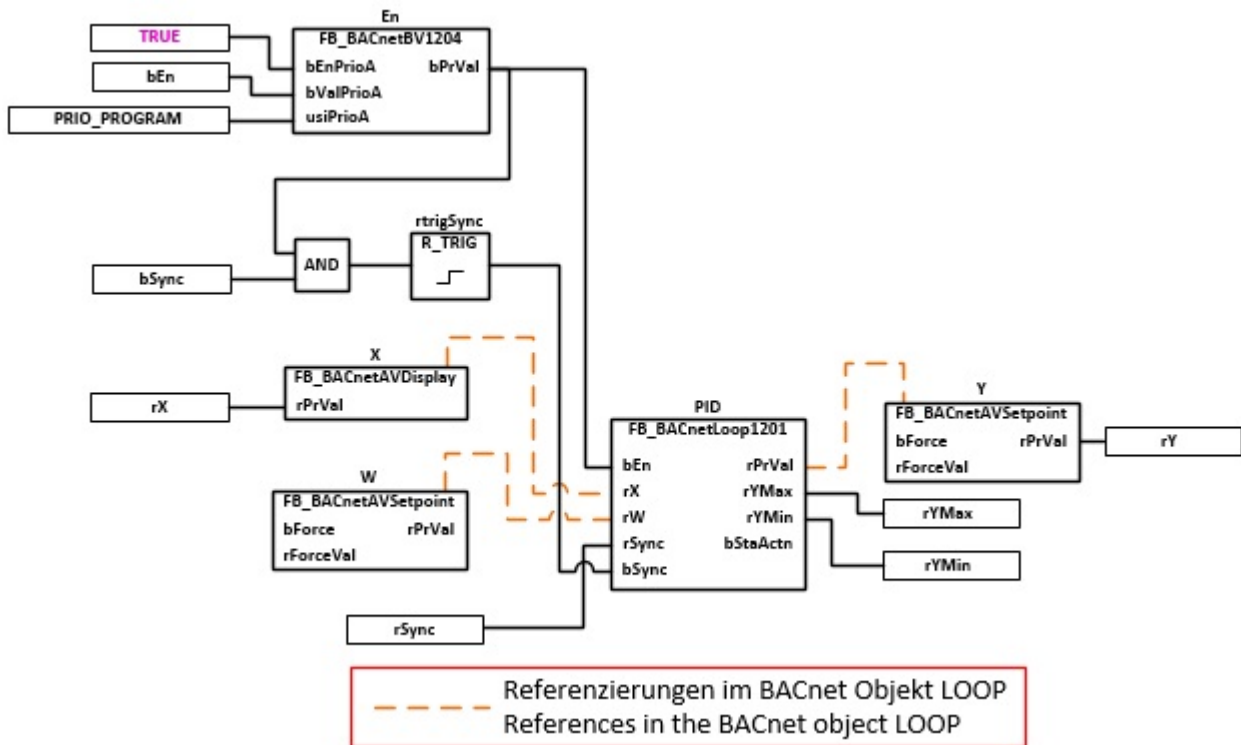
Funktionsbeschreibung

Das Template BAC_PID_02 ist ein universeller PID-Regler.
 Die Referenzierung des Sollwertes, des Istwertes und des Stellausgangs erfolgt mittels der BACnet-Value-Objekte X, W und Y.
 Die Freigabe des PID-Reglers erfolgt anhand der Eingangsvariable **bEn**.

Schnittstelle



Blockschaltbild



VAR_INPUT

```

bEn      : BOOL;
rX       : REAL;
rSync    : REAL;
bSync    : BOOL;
    
```

bEn: Freigabe

rX: Istwert

rSync: Synchronisationswert

bSync: Start Synchronisation

VAR_OUTPUT

```
rY      : REAL;
rYMax   : REAL;
rYMin   : REAL;
```

rY: Ausgabe der Stellgröße

rYMax: Maximalwertes Reglers

rYMin: Minimalwert Reglers

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [[182](#)]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb der Templates [BAC_PltAlm_01](#) [[372](#)] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb der Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
X	FB_BACnetAVDisplay [69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts
W	FB_BACnetAVSetpoint [70]	Über das AV-Objekt wird der Sollwert eingegeben. Es ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts
En	FB_BACnetBV1204 [94]	Das BV-Objekt dient zur Anzeige der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay
PID	FB_BACnetLoop1201 [99]	PID-Regler
rtrigSync	R_TRIG	Bei einer steigenden Flanke am Eingang bSync wird das Loop-Objekt auf den Wert von rSync aufsynchronisiert.
Y	FB_BACnetAVSetpoint [70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.71 BAC_PID_03

Funktionsbeschreibung

Das Template BAC_PID_03 ist ein universeller PID-Regler.
Die Freigabe des PID-Reglers erfolgt anhand der Eingangsvariable **bEn**.

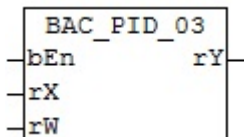
Das LOOP-Objekt benötigt keine Referenzierungsobjekte für den Sollwert, Istwert und Stellausgang. Dazu muss die SPS Kommentar Syntax wie folgt angegeben werden:

(BACnet_ManipulatedVariableReference : undefined :)
(BACnet_ControlledVariableReference : undefined :)
(BACnet_SetpointReference : undefined :)

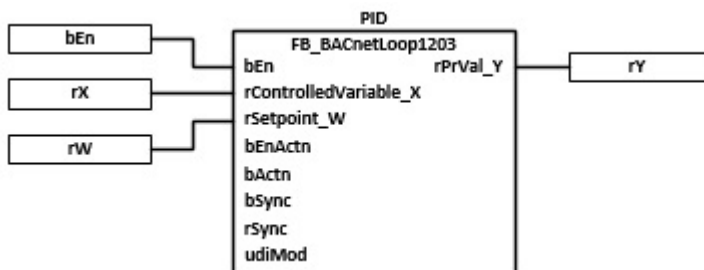
Im System Manager werden die undefined-Kommentare umgewandelt wie im Bild zu sehen ist:

Name	ID	Value	Type
+ ObjectIdentifier	75	Loop:0	BACnetObjectIdentifier
ObjectName	77	002_REG01LP01	ASCII_UTF8
Description	28	PID-Regler	ASCII_UTF8
Reliability	103	no_fault_detected	BACnetReliability
OutOfService	81	<input type="checkbox"/>	Bool
OutputUnits	82	Temperature_degrees_Celsius	BACnetEngineeringUnits
+ ManipulatedVariableReference	60	(AnalogInput:4194303;Present Value)	BACnetObjectPropertyReference
+ ControlledVariableReference	19	(AnalogInput:4194303;Present Value)	BACnetObjectPropertyReference
ControlledVariableUnits	20	Temperature_degrees_Celsius	BACnetEngineeringUnits
- SetpointReference	109	()	BACnetSetpointReference
+ setpointReference		(AnalogValue:0;Present Value)	BACnetObjectPropertyReference
Setpoint	108	20	Real

Schnittstelle



Blockschaltbild



VAR_INPUT

bEn : BOOL;
rX : REAL;
rW : REAL;

bEn: Freigabe

rX: Istwert

rW: Sollwert

VAR_OUTPUT

rY : REAL;

rY: Ausgabe der Stellgröße**Programmbeschreibung**

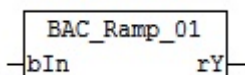
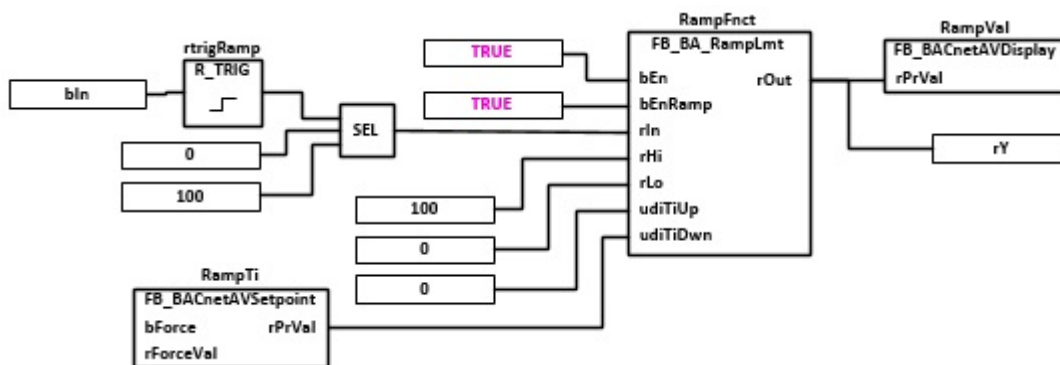
Instanz	Typ	Aufgabe
PID	FB_BACnetLoop1203	PID-Regler

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.72 BAC_Ramp_01**Funktionsbeschreibung**

Das Template stellt eine fallende Rampenbegrenzung dar. Eine steigende Flanke am Eingang **bln** löst die Rampenfunktion aus. Die Ausgangsvariable **rY** nimmt den Wert 100 an und dieser fällt linear in Abhängigkeit der Abfallzeit **RampTi** auf 0.

Schnittstelle**Blockschaltbild****VAR_INPUT**

bIn : BOOL;

bln: Eine steigende Flanke an diesem Eingang löst die Rampenfunktion aus.**VAR_OUTPUT**

rY : REAL;

rY: Ausgang der Rampenfunktion**VAR CONSTANT**

PLT_NUM : BYTE := 1;

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmerbausteins **FB_BA_Alarm**. [▶ 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [▶ 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
RampTi	FB_BACnetAVSetpoint [▶ 70]	AV-Objekt zur Eingabe der Abfallzeit
RampFnct	FB_BA_RampLmt [▶ 143]	Der Funktionsbaustein RampFnct ist das Kernstück der Rampenfunktion.
RampVal	FB_BACnetAVDisplay [▶ 69]	Das AV-Objekt zeigt den Ausgang der Rampenfunktion
rtrigRamp	R_TRIG SEL	Durch eine steigende Flanke am Eingang bln führt dieses Netzwerk dem Funktionsbaustein RampFnct für einen Zyklus den Wert 100 zu und löst somit die Rampenfunktion aus.

Versionshistorie

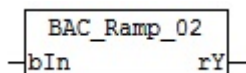
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.73 BAC_Ramp_02

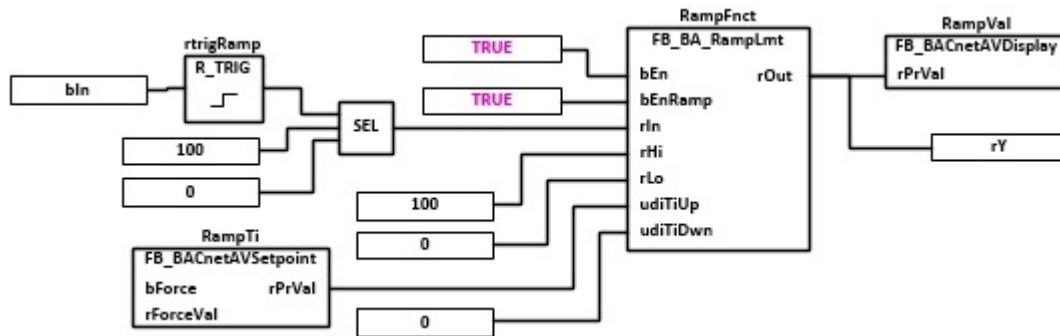
Funktionsbeschreibung

Das Template stellt eine steigende Rampenbegrenzung dar. Eine steigende Flanke am Eingang **bln** löst die Rampenfunktion aus. Die Ausgangsvariable **rY** nimmt den Wert 0 an und dieser steigt linear in Abhängigkeit der Anstiegszeit **RampTi** auf 100.

Schnittstelle



Blockschaltbild



VAR_INPUT

```
bIn      : BOOL;
```

bIn: Eine steigende Flanke an diesem Eingang löst die Rampenfunktion aus.

VAR_OUTPUT

```
rY      : REAL;
```

rY: Ausgang der Rampenfunktion

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarmergebnisses von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmergebnisbausteins `FB_BA Alarm`. [► 182]

Die Auswertung der Alarmergebnisse einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb der Templates `BAC_PltAlm_01` [► 372] mittels des Funktionsbausteins `FB_BA AlarmPlt`. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb der Templates `BAC_PltComnMsg_01` durch den Funktionsbaustein `FB_BA ComnMsg` [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
RampTi	<code>FB_BACnetAVSetpoint</code> [► 70]	AV-Objekt zur Eingabe der Anstiegszeit
RampFunct	<code>FB_BA_RampLmt</code> [► 143]	Der Funktionsbaustein <code>RampFunct</code> ist das Kernstück der Rampenfunktion.
RampVal	<code>FB_BACnetAVDisplay</code> [► 69]	Das AV-Objekt zeigt den Ausgang der Rampenfunktion
rtrigRamp	<code>R_TRIG</code> <code>SEL</code>	Durch eine steigende Flanke am Eingang <code>bIn</code> führt dieses Netzwerk dem Funktionsbaustein <code>RampFunct</code> für einen Zyklus den Wert 0 zu und löst somit die Rampenfunktion aus.

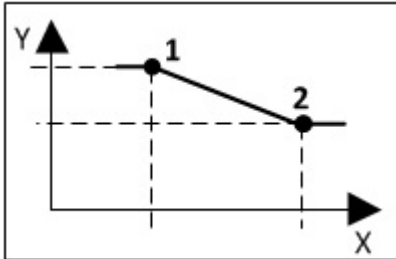
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

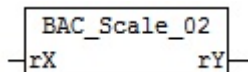
9.74 BAC_Scale_02

Funktionsbeschreibung

Das Template *BAC_Scale_02* stellt eine Linear-Interpolation mit 2 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen. Die Kennlinie wird durch die Stützpunkte [X1/Y1] bis [X2/Y2] bestimmt.



Schnittstelle



VAR_INPUT

rX : REAL;

rX: Eingangswert der Kennlinie

VAR_OUTPUT

rY : REAL;

rY: Errechneter Ausgangswert der Kennlinie

Programmbeschreibung

Instanz	Typ	Aufgabe
X1	FB_BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt X1 vorgegeben
X2	FB_BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt X2 vorgegeben
Y1	FB_BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y1 vorgegeben
Y2	FB_BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y2 vorgegeben
Scale	FB_BA_Chrct02 [▶ 173]	Der Baustein FB_BA_Chrct02 stellt eine Linear-Interpolation mit 2 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen
Y	FB_BACnetAVDisplay [▶ 69]	Ausgabe errechneter Ausgangswert der Kennlinie

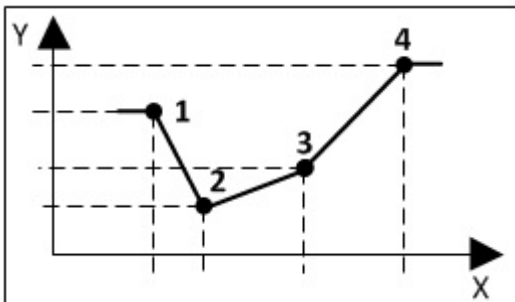
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

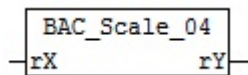
9.75 BAC_Scale_04

Funktionsbeschreibung

Das Template *BAC_Scale_04* stellt eine Linear-Interpolation mit 4 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen. Die Kennlinie wird durch die Stützpunkte [X1/Y1] bis [X4/Y4] bestimmt.



Schnittstelle



VAR_INPUT

rX : REAL;

rX: Eingangswert der Kennlinie

VAR_OUTPUT

rY : REAL;

rY: Errechneter Ausgangswert der Kennlinie

Programmbeschreibung

Instanz	Typ	Aufgabe
X1	FB BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt X1 vorgegeben
X2	FB BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt X2 vorgegeben
X3	FB BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt X3 vorgegeben
X4	FB BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt X4 vorgegeben
Y1	FB BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y1 vorgegeben
Y2	FB BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y2 vorgegeben
Y3	FB BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y3 vorgegeben
Y4	FB BACnetAVSetpoint [▶ 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y4 vorgegeben
Scale	FB BA_Chrct04 [▶ 175]	Der Baustein FB_BA_Chrct04 stellt eine Linear-Interpolation mit 4 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen
Y	FB BACnetAVDisplay [▶ 69]	Ausgabe errechneter Ausgangswert der Kennlinie

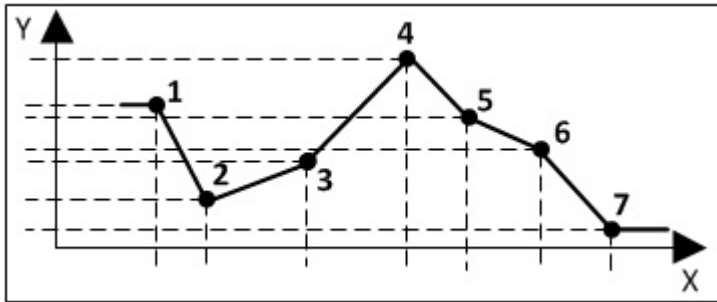
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

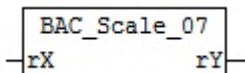
9.76 BAC_Scale_07

Funktionsbeschreibung

Das Template *BAC_Scale_07* stellt eine Linear-Interpolation mit 7 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen. Die Kennlinie wird durch die Stützpunkte [X1/Y1] bis [X7/Y7] bestimmt.



Schnittstelle



VAR_INPUT

rX : REAL;

rX: Eingangswert der Kennlinie

VAR_OUTPUT

rY : REAL;

rY: Errechneter Ausgangswert der Kennlinie

Programmbeschreibung

Instanz	Typ	Aufgabe
X1	FB_BACnetAVSetpoint > 70	Über das AV-Objekt wird der Wert für den Stützpunkt X1 vorgegeben
X2	FB_BACnetAVSetpoint > 70	Über das AV-Objekt wird der Wert für den Stützpunkt X2 vorgegeben
X3	FB_BACnetAVSetpoint > 70	Über das AV-Objekt wird der Wert für den Stützpunkt X3 vorgegeben
X4	FB_BACnetAVSetpoint > 70	Über das AV-Objekt wird der Wert für den Stützpunkt X4 vorgegeben
X5	FB_BACnetAVSetpoint > 70	Über das AV-Objekt wird der Wert für den Stützpunkt X5 vorgegeben
X6	FB_BACnetAVSetpoint > 70	Über das AV-Objekt wird der Wert für den Stützpunkt X6 vorgegeben
X7	FB_BACnetAVSetpoint > 70	Über das AV-Objekt wird der Wert für den Stützpunkt X7 vorgegeben
Y1	FB_BACnetAVSetpoint > 70	Über das AV-Objekt wird der Wert für den Stützpunkt Y1 vorgegeben

Instanz	Typ	Aufgabe
Y2	FB_BACnetAVSetpoint [► 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y2 vorgegeben
Y3	FB_BACnetAVSetpoint [► 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y3 vorgegeben
Y4	FB_BACnetAVSetpoint [► 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y4 vorgegeben
Y5	FB_BACnetAVSetpoint [► 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y5 vorgegeben
Y6	FB_BACnetAVSetpoint [► 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y6 vorgegeben
Y7	FB_BACnetAVSetpoint [► 70]	Über das AV-Objekt wird der Wert für den Stützpunkt Y7 vorgegeben
Scale	FB_BA_Chrc07 [► 176]	Der Baustein FB_BA_Chrc07 stellt eine Linear-Interpolation mit 7 Stützstellen dar und lässt sich zur Kennlinienerstellung nutzen
Y	FB_BACnetAVDisplay [► 69]	Ausgabe errechneter Ausgangswert der Kennlinie

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.77 BAC_Uni_Vlv_01_xx

Funktionsbeschreibung

Das Template **BAC_Uni_Vlv_01_xx** ist für die Ansteuerung eines stetigen Regelventils. Es besteht im Wesentlichen aus einem AO-Objekt inklusive Trend-Objekt, einem MV-Objekt zur manuellen Steuerung und dem dazu gehörigen AV-Objekt zur Eingabe der manuellen Position. Das Template wird ergänzt durch optionale BACnet Objekte, siehe Tabelle Ausstattungsvarianten.



Die beiden Ausgangsvariablen rPrVal / bSync sind nur dann aktiv, wenn in dem eingesetzten Template die Rückmeldung der Ventilstellung Fdb vorhanden ist. Ist dieses nicht der Fall, so geben die beiden Variablen den Wert Null aus. Das Interface der Templates BAC_Uni_Vlv3P_01_xx / BAC_Uni_Vlv_01_xx ist gleich. Aus diesem Grunde kann im Project Builder in einem Aufruf-Template ein 3Punkt-Ventil durch ein stetiges Ventil ausgetauscht werden, ohne das im SPS-Code nachgebessert werden muss.

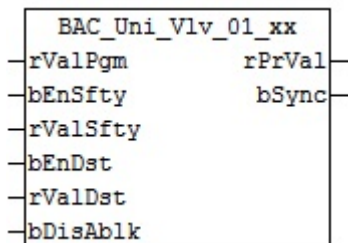
Ausstattungsvarianten

Das Template **BAC_Uni_Vlv_01_xx** existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten der Regelventils werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

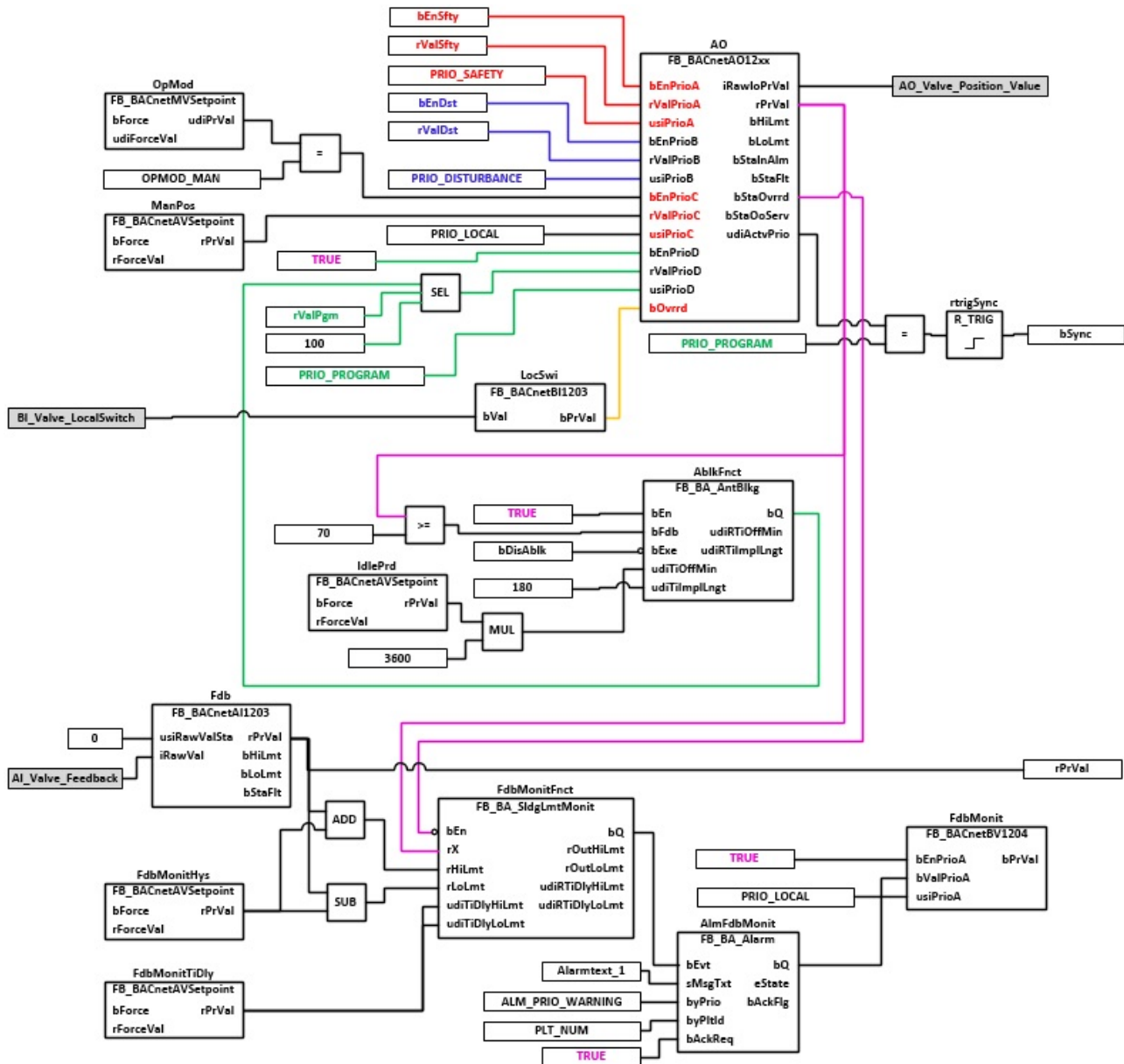
Optionen	Blockierschutz (Block)	Rückmeldung Ventilstellung mit Überwachung (Rm)	mechanische Vor- rangbedienung Stellungsrückmel- dung Poti (Rm-Ausg)	mechanische Vor- rangbedienung Stellungsrückmel- dung Handschal- ter (A-H)
Instanz	AblkFnct	Fdb	FdbOut	LocSwi
Datenpunkt Typ	-	AI	AI	BI
	8	4	2	1
BAC_Uni_Vlv_01_00	0	0	0	0
BAC_Uni_Vlv_01_01	0	0	0	1

Optionen	Blockierschutz (Block)	Rückmeldung Ventilstellung mit Überwachung (Rm)	mechanische Vor- rangbedienung Stellungsrückmel- dung Poti (Rm-Ausg)	mechanische Vor- rangbedienung Stellungsrückmel- dung Handschal- ter (A-H)
Instanz	AblkFunct	Fdb	FdbOut	LocSwi
Datenpunkt Typ	-	AI	AI	BI
	8	4	2	1
BAC_Uni_Vlv_01_03	0	0	1	1
BAC_Uni_Vlv_01_04	0	1	0	0
BAC_Uni_Vlv_01_05	0	1	0	1
BAC_Uni_Vlv_01_08	1	0	0	0
BAC_Uni_Vlv_01_09	1	0	0	1
BAC_Uni_Vlv_01_11	1	0	1	1
BAC_Uni_Vlv_01_12	1	1	0	0
BAC_Uni_Vlv_01_13	1	1	0	1

Schnittstelle



Blockschaltbild Variante BAC_Uni_Vlv_01_13



VAR_INPUT

```

rValPgm : REAL;
bEnSfty : BOOL;
bValSfty : BOOL;
bEnDst : BOOL;
rValDst : REAL;

```

rValPgm: Analoger Wert Programmpriorität

bEnSfty: Freigabe Sicherheitspriorität

rValSfty: Analoger Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität

rValDst: Analoger Wert Störungspriorität

bDisAbk: Verriegelung Blockierschutzfunktion. Verhindert die gleichzeitige Aktivierung des Blockierschutzes von der Pumpe und dem Regelventil, z. B. an einem Wärmetauscher

VAR_OUTPUT

```

rPrVal : REAL;
bSync : BOOL;

```

rPrVal : aktuelle Stellung des Regelventils.

bSync: Ausgabe eines Impulses um den Ventil zugehörigen Regler beim Rücksetzen vom Hand- in den Automatikbetrieb auf die aktuelle Position des Ventils zu synchronisieren.

Der Synchronisationsimpuls **bSync** sollte nur dann eingesetzt werden, wenn in dem eingesetzten Template die Rückmeldung der Ventilstellung **Fdb** vorhanden ist.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe			
OpMod	FB_BACnetMVSetpoint [► 131]		MV-Objekt zur manuellen Steuerung des Regelventils von der MBE oder einem lokalen Bediendisplay			
IdlePrd	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe der maximalen Dauer einer Ventilstillstandsperiode bis zur Ausgabe eines Blockierschutzimpulses.			
ManPos	FB_BACnetAVSetpoint [► 70]		AV-Objekt zur Eingabe der Position des Ventils bei manueller Übersteuerung			
FdbMonithys	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe der Hysterese für die Funktionsüberwachung des Regelventils mittels der Stellungsrückmeldung			
Fdb	FB_BACnetAI1203 [► 49]	X	AI-Objekt zur Erfassung der Stellungsrückmeldung vom Ventil.			
FdbOut	FB_BACnetAI1203 [► 49]	X	AI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Poti			
FdbMonitTidy	FB_BACnetAVSetpoint [► 70]	X	AV-Objekt zur Eingabe der Ansprechverzögerung der Funktionsüberwachung mittels Stellungsrückmeldung.			
LocSwi	FB_BACnetBI1203 [► 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Rückmeldung Handschalter			
AO	FB_BACnetAO1203 [► 53]		AO-Objekt für die Ansteuerung des Regelventils.			
			Priorität:	Freigabe	Wert	Bemerkung
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty	
			PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst	

Instanz	Typ	optional	Aufgabe			
			PRIO_LOCAL (8)	OpMod_udiPrVal = OPMOD_MAN	ManPos_rPrVal	Bei Handbetrieb Wert von AV-Objekt ManPos
			PRIO_PROGRAM (15)	TRUE	AbkFnc bQ oder rValPgm	Wert von Eingang rValPgm oder 100% von Blockierschutz
	EQ		Ist TRUE wenn die Aktive Priorität PRIO_PROGRAM (15) ist. Kann zur Synchronisation des Reglers bei Rückkehr in den Automatikbetrieb benutzt werden			
FdbMonitFunct	FB_BA_SldgLmtMonit [▶ 145]	X	Funktionsüberwachung des Regelventils durch Vergleich des Stellausgangs und der Stellungsrückmeldung.			
AlmFdbMonit	FB_BA_Alarm [▶ 182]	X	Erfassung und Weiterverarbeitung eines Fehlers von der Stellungsrückmeldeüberwachung			
FdbMonit	FB_BACnetBV1204 [▶ 94]	X	Meldet Fehler Stellungsrückmeldung an die MBE			
AbkFnc	FB_BA_AntBlkg [▶ 218]	X	Erzeugt Blockierschutzimpuls wenn in dem letzten Zeitintervall von IdlePrd keine Ventilfehrt mit mehr als 70% Fahrweg stattgefunden hat. Impulslänge 180 Sekunden.			
TLogAO	FB_BACnetTLog1201 [▶ 137]		Zeichnet den Present Value des AO-Objekts auf			

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_Valve_LocalSwitch	BOOL	X	Eingang	Digitaleingang - Schalter Hand Ventil - Meldung - Hand/Auto
AI_Valve_Feedback	INT	X	Eingang	Analogeingang - Ventil - Messwert - Stellungsrückmeldung
AI_Feedback_Position_Poti	INT	X	Eingang	Analogeingang - Hand Poti - Rückmeldung - Stellwert
AO_Valve_Position_Value	INT		Ausgang	Analogausgang - Ventil - Stellwert Position

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.78 BAC_Uni_Vlv3P_01_xx

Funktionsbeschreibung

Das Template **BAC_Uni_Vlv3P_01_xx** ist für die Ansteuerung eines Dreipunktventils. Es besteht im Wesentlichen aus zwei BO-Objekten für das Auf- und Zufahren und einem MV-Objekt zur manuellen Steuerung. Das Template wird ergänzt durch optionale BACnet Objekte, siehe Tabelle Ausstattungsvarianten.

Unter **IO-Verknüpfung** sind die Variablen zu finden, die mit dem Prozessabbild der Eingangs- und Ausgangsebene in der SPS verknüpft werden.



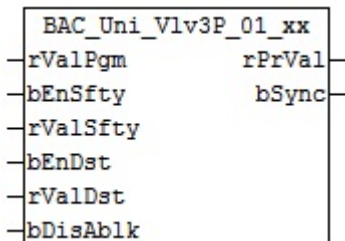
Die beiden Ausgangsvariablen rPrVal / bSync sind nur dann aktiv, wenn in dem eingesetzten Template die Rückmeldung der Ventilstellung Fdb vorhanden ist. Ist dieses nicht der Fall, so geben die beiden Variablen den Wert Null aus. Das Interface der Templates BAC_Uni_Vlv3P_01_xx / BAC_Uni_Vlv_01_xx ist gleich. Aus diesem Grunde kann im Project Builder in einem Aufruf-Template ein 3Punkt-Ventil durch ein stetiges Ventil ausgetauscht werden, ohne das im SPS-Code nachgebessert werden muss.

Ausstattungsvarianten

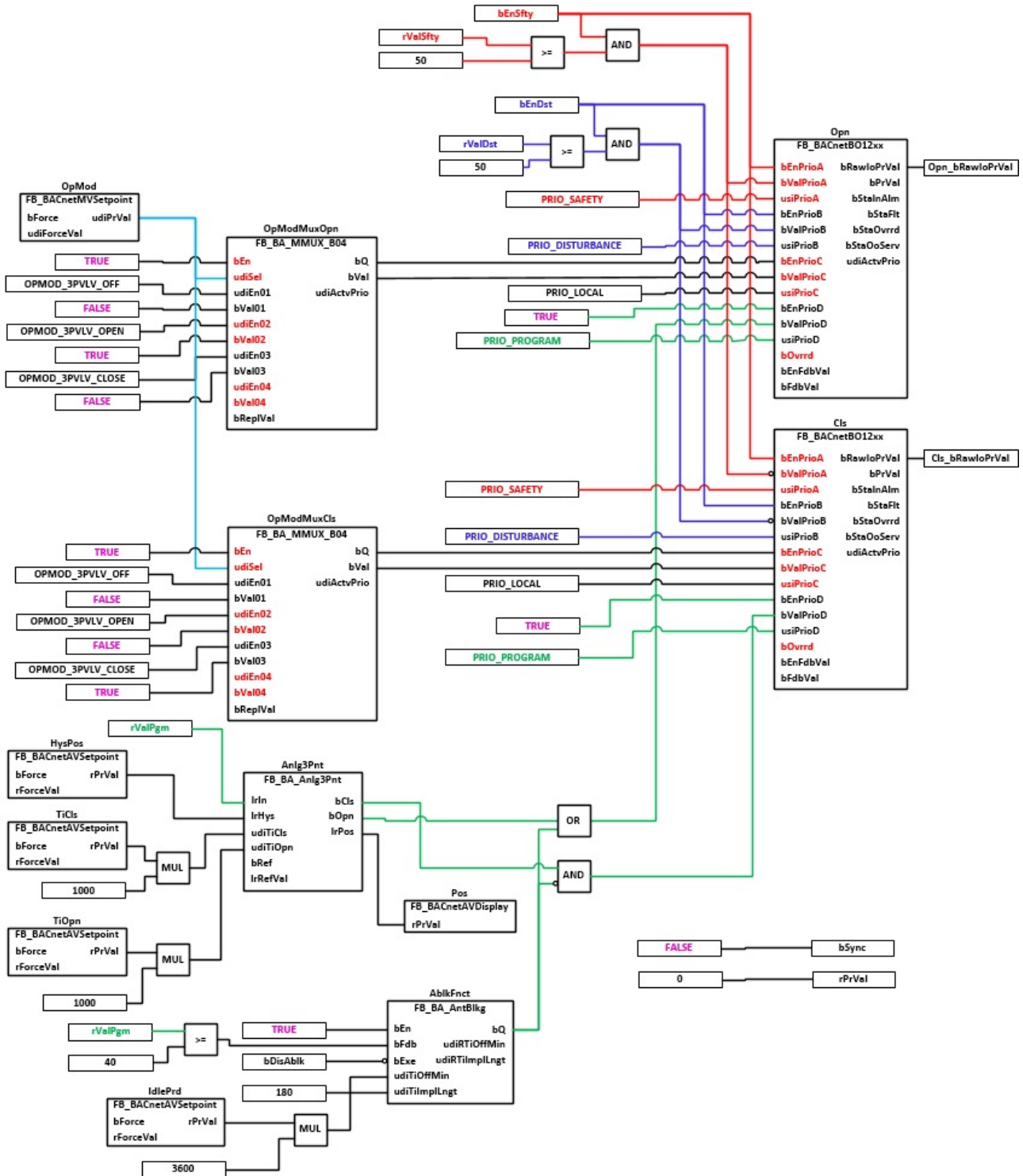
Das Template BAC_Uni_Vlv3P_xx existiert in verschiedenen Ausstattungsvarianten. Die Ausstattungsvarianten der Ventile werden nach einem Kennzeichnungsschlüssel benannt. Der Kennzeichnungsschlüssel leitet sich aus der nachstehenden Tabelle ab.

Optionen	Rückmeldung Ventilstellung	Blockierschutz	mechanische Vorrangbedien- ung Stellungsrück- meldung Zu	mechanische Vorrangbedie- nung Stellungsrück- meldung Auf	mechanische Vorrangbedie- nung Stellungsrück- meldung Hand- schalter	Endlage Zu	Endlage Auf
Instanz	Fdb	AblkFunct	FdbOutCls	FdbOutOpn	LocSwi	SwiCls	SwiOpn
Datenpunkt Typ	AI		BI	BI	BI	BI	BI
	64	32	16	8	4	2	1
BAC_Uni_Vlv3P_01_32	0	1	0	0	0	0	0

Schnittstelle



Blockschaltbild Variante BAC_Uni_Vlv3P_01_32



VAR_INPUT

```

bValPgm      : BOOL;
bEnSfty      : BOOL;
bValSfty     : BOOL;
bEnDst       : BOOL;
bValDst      : BOOL;
    
```

rValPgm: Analoger Wert Programmpriorität

bEnSfty: Freigabe Sicherheitspriorität

bEnDst: Freigabe Störungspriorität

bDisAbIk: Verriegelung Blockierschutzfunktion. Verhindert die gleichzeitige Aktivierung des Blockierschutzes von der Pumpe und dem Regelventil, z. B. an einem Wärmetauscher

VAR_OUTPUT

```
rPrVal      : REAL;
bSync       : BOOL;
```

rPrVal : aktuelle Stellung des Regelventils.

bSync: Ausgabe eines Impulses um den Ventil zugehörigen Regler beim Rücksetzen vom Hand- in den Automatikbetrieb auf die aktuelle Position des Ventils zu synchronisieren.

Der Synchronisationsimpuls **bSync** sollte nur dann eingesetzt werden, wenn in dem eingesetzten Template die Rückmeldung der Ventilstellung **Fdb** vorhanden ist.

VAR CONSTANT

```
PLT_NUM     : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [[▶ 182](#)]

Die Auswertung der Alarmergebnisse einer Anlage, z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [[▶ 372](#)] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
SwiOpn	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt zur Erfassung der Endlage Auf des Ventils
SwiCls	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt zur Erfassung der Endlage Zu des Ventils
LocSwi	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Handschalter
FdbOutOpn	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Auf
FdbOutCls	FB_BACnetBI1203 [▶ 72]	X	BI-Objekt zur Erfassung der mechanischen Vorrangbedienung Stellungsrückmeldung Zu
FdbVlv	FB_BACnetAI1203 [▶ 49]	X	AI-Objekt zur Erfassung der Stellungsrückmeldung vom Ventil.
HysPos	FB_BACnetAVSetpoint [▶ 70]		AV-Objekt zur Eingabe des Wertes Hysterese Starten der Positionsänderung
IdlePrd	FB_BACnetAVSetpoint [▶ 70]	X	AV-Objekt zur Eingabe des Wertes Ruhezeit Blockierschutz
OpMod	FB_BACnetMVSetpoint [▶ 131]		MV-Objekt zur manuellen Steuerung des Ventils von der MBE oder einem lokalen Bediendisplay

Instanz	Typ	optional	Aufgabe		
TiOpn	FB_BACnetMVSetpoint [▶ 131]		AV-Objekt zur Eingabe des Wertes für die Auffahrzeit		
TiCls	FB_BACnetMVSetpoint [▶ 131]		AV-Objekt zur Eingabe des Wertes für die Zufahrzeit		
OpModMuxOpn	FB_BA_MMUX_B04 [▶ 208]		Multiplexer Auswertung Betriebsart für das Öffnen		
OpModMuxCls	FB_BA_MMUX_B04 [▶ 208]		Multiplexer Auswertung Betriebsart für das Schließen		
Anlg3Pnt	FB_BA_Anlg3Pnt [▶ 138]		Der Funktionsbaustein Anlg3Pnt ist das Kernstück des Templates und für die Ansteuerung des 3Punkt Ventils vorgesehen. Er wandelt ein stetiges Signal für die Positionierung in die binären Befehle Auf/ Zu		
Pos	FB_BACnetAVDisplay [▶ 69]		Aktuelle errechnete Position des Stellantriebs		
Opn	FB_BACnetBO1203 [▶ 82]		BO-Objekt für das Auffahren des Stellantriebs		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	Eingang bEnSfty	FALSE
			PRIO_DISTURBAN CE (3)	Eingang bEnDst	FALSE
			PRIO_LOCAL (8)	Das OpModMuxOpn bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt hat den Wert OPMOD_3PVLV_OFF (Hand Aus) 2. Das MV-Objekt hat den Wert OPMOD_3PVLV_OPEN (Hand Auf) 3. Das MV-Objekt hat den Wert OPMOD_3PVLV_CLOSE (Hand Zu)	OpModMuxOpn_bVal = TRUE, wenn OpMod_udiPrVal = OPMOD_3PVLV_OPEN
PRIO_PROGRAM (15)	TRUE	Anlg3Pnt_bOpnORAbkFunct_bQ			
Cls	FB_BACnetBO1203 [▶ 82]		BO-Objekt für das Auffahren des Stellantriebs		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	Eingang bEnSfty	TRUE
			PRIO_DISTURBAN CE (3)	Eingang bEnDst	TRUE
			PRIO_LOCAL (8)	Das OpModMuxCls bündelt Ereignisse welche das Schreiben auf die Priorität Manuelle	OpModMuxCls_bVal = TRUE, wenn OpMod_udiPrVal = OPMOD_3PVLV_CLOSE

Instanz	Typ	optional	Aufgabe		
				Übersteuerung (Local) des nachgeschalteten BO-Objekts aktiviert. Ereignisse: 1. Das MV-Objekt hat den Wert OPMOD_3PVLV_OFF (Hand Aus) 2. Das MV-Objekt hat den Wert OPMOD_3PVLV_OPEN (Hand Auf) 3. Das MV-Objekt hat den Wert OPMOD_3PVLV_CLOSE (Hand Zu)	
			PRI0_PROGRAM (15)	TRUE	Anlg3Pnt_bClisAND NOTAbkFncT_bQ
AbkFncT	FB_BA_AntBlkg [▶ 218]	X	Erzeugt Blockierschutzimpuls wenn in dem letzten Zeitintervall von IdlePrd keine Ventifahrt mit mehr als 40% Fahrweg stattgefunden hat. Impulslänge 180 Sekunden.		
TLogPos	FB_BACnetTLog1 201 [▶ 137]		Zeichnet den Present Value des Pos-Objekts auf		

IO-Verknüpfung

In der zum Template gehörigen XML-Description sind in dem Bereich **Parameter** Variablen mit der Kennung **INPUT** oder **Output** deklariert. Diese Parameter können im Project Builder oder über die Excel-Import-Schnittstelle mit dem Prozessabbild der Eingangs- und Ausgangsebene in der SPS verknüft werden.

Parameter	Typ	Instanz	Typ	Prozessabbild	
SwiOpn_bVal	BOOL	SwiOpn	FB_BACnetBI120 3 [▶ 72]	Eingang	
SwiClis_bVal	BOOL	SwiClis	FB_BACnetBI120 3 [▶ 72]	Eingang	
LocSwi_bVal	BOOL	LocSwi	FB_BACnetBI120 3 [▶ 72]	Ausgang	
FdbOutOpn_bVal	BOOL	FdbOutOpn	FB_BACnetBI120 3 [▶ 72]	Eingang	
FdbOutClis_bVal	BOOL	FdbOutClis	FB_BACnetBI120 3 [▶ 72]	Eingang	
FdbVlv_usiRawValSta	USINT	FdbVlv	FB_BACnetAI12 03 [▶ 49]	Eingang	
FdbVlv_iRawVal	INT	FdbVlv	FB_BACnetAI12 03 [▶ 49]	Eingang	
Opn_bRawIoPrVal	BOOL	Opn	FB_BACnetBO12 03 [▶ 82]	Ausgang	
Clis_bRawIoPrVal	BOOL	Clis	FB_BACnetBO12 03 [▶ 82]	Ausgang	

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.79 Lüftungsanlagen

9.79.1 BAC_AC_KennzeichnungssystemAnlagenschlüssel

Kennzeichnungssystem Anlagentemplates Lüftungs- und Klimatechnik

Lüftungsanlagen mit Zu- und Abluftventilator und thermischer Luftbehandlung

Typ: BAC_AC_SE

Strategie		Ventilator		Vorerhitzer		Kühler		Energierückgewinnung		Mischluft		
Zulufr egelun g (einfac h)	1	1- stufig	1	Nein	0	Nein	0	Nein	0	Nein	0	
Zulufr egelun g	2	2- stufig	2	Ja	1	Ja	1	Platte	1	Ja	1	
Abluft/ Zuluftk askade (einfac h)	3	Drehz ahl	3					Rekup erativ	2			
Abluft/ Zuluftk askade	4	Druck	4					Rotatio n	3			
		Volum enstro m	5									

Lüftungsanlagen mit Zuluftventilator und thermischer Luftbehandlung

Typ: BAC_AC_S

Strategie		Ventilator		Vorerhitzer		Kühler		Energierückgewinnung		Mischluft		
Zulufr egelun g (einfac h)	1	1- stufig	1	Nein	0	Nein	0	Nein	0	Nein	0	
Zulufr egelun g	2	2- stufig	2	Ja	1	Ja	1	Rekup erativ	2	Ja	1	

Abluft/ Zuluftkaskade (einfach)	3	Drehzahl	3										
Abluft/ Zuluftkaskade	4	Druck	4										
		Volumenstrom	5										

Lüftungsanlagen mit Zu- und Abluftventilator mit thermischer Luftbehandlung und Be/Entfeuchtung

Typ: BAC_AC_SEH

Strategie	Ventilator		Vorerhitzer		Kühler		Energierückgewinnung		Mischluft		Befeuchter		Nacherhitzer		
Zuluft (einfach)	1	1-stufig	1	Nein	0	Nein	0	Nein	0	Nein	0	Nein	0	Nein	0
Zuluft	2	2-stufig	2	Ja	1	Ja	1	Platte	1	Ja	1	Dampf	1	Ja	1
Kaskade (einfach)	3	Drehzahl	3					Rekuperativ	2						
Kaskade	4	Druck	4					Rotation	3						
		Volumenstrom	5												

Abluftventilator ohne Luftbehandlung

Typ: BAC_AC_E

Steuerung	Ventilator
Anforderung von: Zeitschaltplan, Schalter o. externer Anforderung	1-stufig 1
	2-stufig 2
	Drehzahl 3
	Druck 4

9.79.2 BAC_AC_SE_3_4_1_1_1_0

Anwendung

Anlagenaufrufftemple einer temperaturgeregelten Lüftungsanlage.

Die wesentlichen Aufgaben des Templates sind:

- Anlagensteuerung mit den verschiedenen Betriebsarten und Sollwerten
- Temperaturregelung für die ausgewählten Sequenzelemente

Anlagenschlüssel

Lüftungsanlage mit Zu- und Abluftventilator und thermischer Luftbehandlung

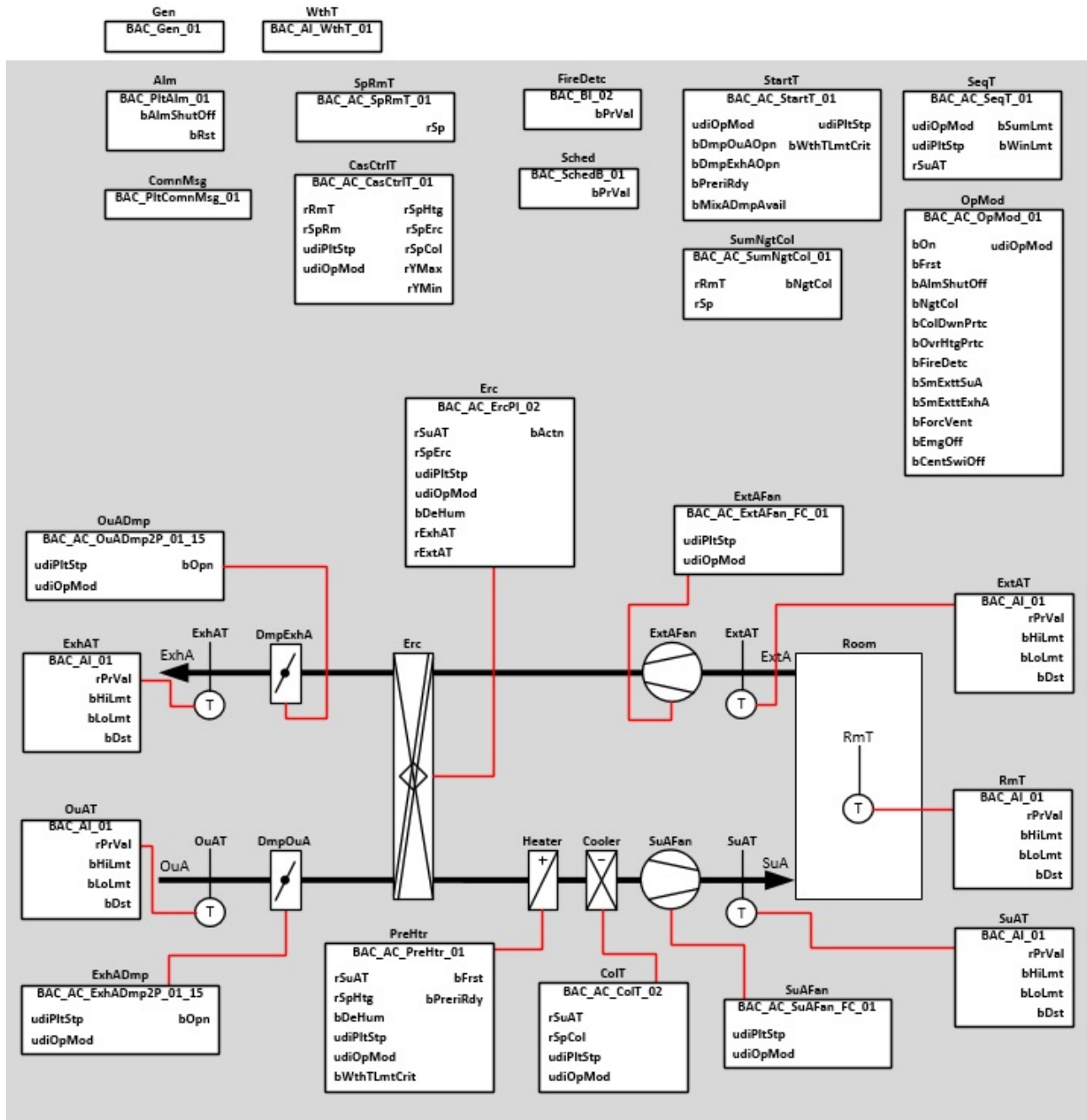
Typ: BAC_AC_SE

Strategie		Ventilator		Vorerhitzer		Kühler		Energierückgewinnung		Mischluft		
Zulufr egelun g (einfac h)	1	1- stufig	1	Nein	0	Nein	0	Nein	0	Nein	0	
Zulufr egelun g	2	2- stufig	2	Ja	1	Ja	1	Platte	1	Ja	1	
Abluft/ Zuluftk askade (einfac h)	3	Drehz ahl	3					Rekup erativ	2			
Abluft/ Zuluftk askade	4	Druck	4					Rotatio n	3			
		Volum enstro m	5									
Abluft/ Zuluftk askade (einfac h)	3	Druck	4	Vorerh itzer	1	Kühler	1	Energi erückg ewinnu ng	1	Mischl uft	0	

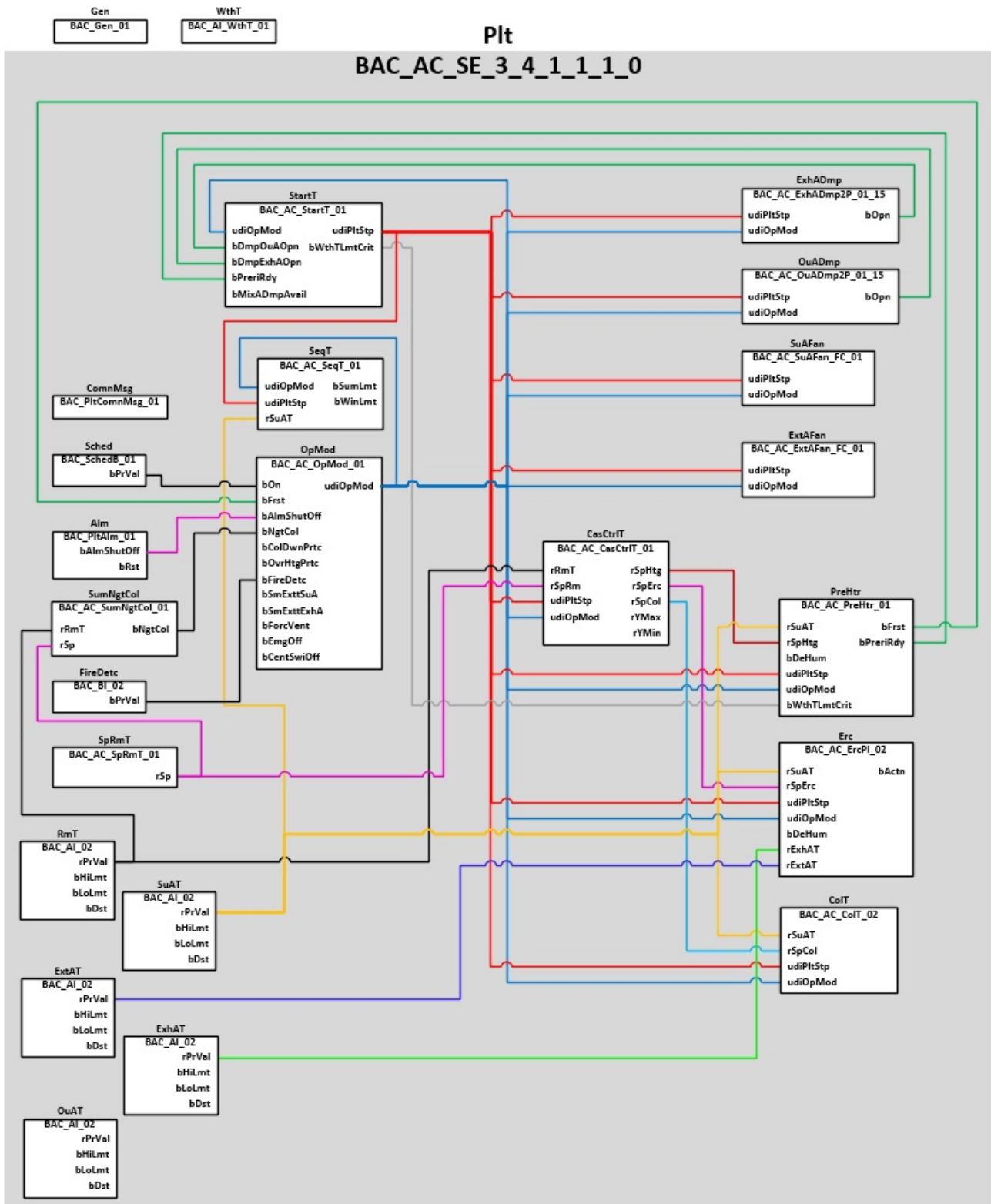
Schnittstelle

BAC_AC_SE_3_4_1_1_1_0

Anlagenschema



Blockschaltbild



Betriebsarten

Die Anlage wird eingeschaltet über ein Zeitschaltprogramm oder einen manuellen Anlagenschalter (Software).

Die Anlage kann die in der Tabelle aufgeführten Betriebsarten annehmen.

udiOpMod		
OPMOD_AC_OFF	1	Aus//Off

OPMOD_AC_ON	2	Ein//On
OPMOD_AC_EMERG	3	Notfall//Emergency
OPMOD_AC_MANOFF	4	Hand aus//Manual off
OPMOD_AC_MANON	5	Hand ein//Manual on
OPMOD_AC_FRST	6	Frost//Frost
OPMOD_AC_SMEXTTPRG	7	Entrauchung Programm//Smoke extraction programm
OPMOD_AC_SMEXTTSUA	8	Entrauchung Zuluft//Smoke extraction supply
OPMOD_AC_SMEXTTEXHA	9	Entrauchung Fortluft//Smoke extraction exhaust
OPMOD_AC_FIRE	10	Feuer//Fire
OPMOD_AC_NGTCOL	11	Nachtkühlung//Night cooling
OPMOD_AC_COLDWNPRTC	12	Stützbetrieb,Auskühlschutz//Cool down protection
OPMOD_AC_OVRHTGPRTC	13	Überhitzungsschutz//Over heating protection
OPMOD_AC_ALM	14	Störung//Alarm
OPMOD_AC_FORCVENT	15	Zwangsbelüftung//Forced ventilation
OPMOD_AC_CENTSWIOFF	16	Zentralabschaltung//Central switch-off

Feueralarm

Bei Feueralarm werden alle Einschaltbedingungen übersteuert und die Anlage ausgeschaltet. Es werden die Entrauchungsfunktionen frei gegeben. Mit den beiden binären Eingängen *SmExttSuA* / *SmExttExhA* können die 3 oben aufgeführten Entrauchungen vorgenommen werden.

Störabschaltungen

Die Anlage schaltet ab bei folgenden Störungen oder Betriebsmeldungen:

- Betriebsmeldung Feueralarm ([BAC AC StartT 01 \[▶ 538\]](#))
- Betriebsmeldung Frostalarm ([BAC FrstPrt 01 \[▶ 515\]](#))
- Störung der Ventilatoren ([BAC AC ExtAFan FC 01 \[▶ 430\]](#) / [BAC AC SuAFan FC 01 \[▶ 435\]](#))
- Störung der Außen- oder Fortluftklappe ([BAC AC OuADmp2P 01 15 \[▶ 454\]](#) / [BAC AC ExhADmp2P 01 15 \[▶ 446\]](#))

Anfahren der Anlage

Beim Einschalten der Anlage bei kalter Witterung wird im Anlagenschritt Vorspülen [PLTSTP AC PRERI \[▶ 364\]](#) die Regelung der Rücklauftemperatur des Warmwasserluftherhizers aktiviert. Dieser Vorgang soll das Einfrieren des Warmwasserluftherhizer während des Anlagenstarts verhindern.

Temperaturregelung

In diesem Anlagentemplate wird eine Raumluft-Zuluft-Kaskade als Regelstrategie realisiert.

In den Templates [BAC AC SpRmT 01 \[▶ 658\]](#) / [BAC AC CasCtrlT 01 \[▶ 649\]](#) werden die Sollwerte für die Anlage erzeugt und an die Sequenzregler weiter geleitet.

Programmbeschreibung

Instanz	Typ	Aufgabe
Alm	BAC PltAlm 01 [▶ 372]	Anlagenalarming
ComnMsg	BAC PltComnMsg 01 [▶ 376]	Anlagensammelmeldungen
ExhAT	BAC AI 01 [▶ 683]	Aufruftemplate Fortlufttemperatur
ExtAT	BAC AI 01 [▶ 683]	Aufruftemplate Ablufttemperatur
OuAT	BAC AI 01 [▶ 683]	Aufruftemplate Außenlufttemperatur
RmT	BAC AI 01 [▶ 683]	Aufruftemplate Raumtemperatur

Instanz	Typ	Aufgabe
SuAT	BAC AI 01 [▶ 683]	Aufruf-template Zulufttemperatur
Sched	BAC SchedB 01 [▶ 718]	Aufruf-template Anlagenzeitschaltprogramm
FireDetc	BAC BI 01 [▶ 703]	BI-Objekt zur Erfassung des Ereignisses Feueralarm
SuAFan	BAC AC SuAFan FC 01 [▶ 435]	Aufruf-template mit integriertem Druckregler für einen Drehzahl gesteuerten Zuluftventilator
ExtAFan	BAC AC ExtAFan FC 01 [▶ 430]	Aufruf-template mit integriertem Druckregler für einen Drehzahl gesteuerten Abluftventilator
ExhADmp	BAC AC ExhADmp2P 01 15 [▶ 446]	Aufruf-template Fortluftklappe mit einem Federrücklaufantrieb und Endlagenkontrolle
OuADmp	BAC AC OuADmp2P 01 15 [▶ 454]	Aufruf-template Außenluftklappe mit einem Federrücklaufantrieb und Endlagenkontrolle
SpRmT CasCtrlT	BAC AC SpRmT 01 [▶ 658] BAC AC CasCtrlT 01 [▶ 649]	Sollwert Raumluft-Zuluft-Kaskade
SeqT	BAC AC SeqT 01 [▶ 532]	Aufruf-template Sequenzsteuerung
OpMod	BAC AC OpMod 01 [▶ 524]	Anlagenbetriebsart
Start	BAC AC StartT 01 [▶ 538]	Aufruf-template des Anlagenstartprogramms einer temperaturgeregelten Lüftungsanlage
SumNgtCo l	BAC AC SumNgtCol 01 [▶ 549]	Nachtauskühlung
PreHtr	BAC AC PreHtr 01 [▶ 501]	Aufruf-template eines Warmwasserluftherhitzer
Erc	BAC AC ErcPI 01 [▶ 398]	Aufruf-template für die Ansteuerung eines Plattenwärmetauschers für die Energierückgewinnung einer Lüftungsanlage.
CoIT	BAC_AC_CoIT_02	Aufruf-template eines temperaturgesteuerten Kaltwasser-Luftkühlers (Ventil)

Anlagenalarmaufnahme

Die Alarmpriorität ALM_PRIO_CRITICAL ist eine anlagenabschaltende Störung.

Template	Subtemplate	Instanz	Typ	Alarmpriorität	
ExhAT		AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_AL ARM	
ExtAT		AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_AL ARM	
OuAT		AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_AL ARM	
RmT		AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_AL ARM	
SuAT		AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_AL ARM	

Template	Subtemplate	Instanz	Typ	Alarmpriorität	
SuAFan	SuAFanFU	AlmThOvrlid	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmDstFC	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	SuAFanDiffPrssMonit	AlmDiffPrss	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
ExtAFan	ExtAFanFU	AlmThOvrlid	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmDstFC	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	ExtAFanDiffPrssMonit	AlmDiffPrss	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
ExhADmp		AlmFnct	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
OuADmp		AlmFnct	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
PreHtr	PreHtrRetWtrT	AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstThermostat	AlmBI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstPrt	AlmFrst	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstPrtT	AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrPu	AlmDst	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrVlv	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	
Erc	ErcDmp	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	
ColT	ColTVlv	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	

Versionshistorie

Versionsnummer	Bemerkungen
1.0.1.0	erste Freigabe

9.79.3 BAC_AC_SE_4_4_1_1_0_1

Anwendung

Anlagenauffructemplate einer temperaturgeregelten Lüftungsanlage.

Die wesentlichen Aufgaben des Templates sind:

- Anlagensteuerung mit den verschiedenen Betriebsarten und Sollwerten
- Temperaturregelung für die ausgewählten Sequenzelemente

Anlagenschlüssel

Lüftungsanlage mit Zu- und Abluftventilator und thermischer Luftbehandlung

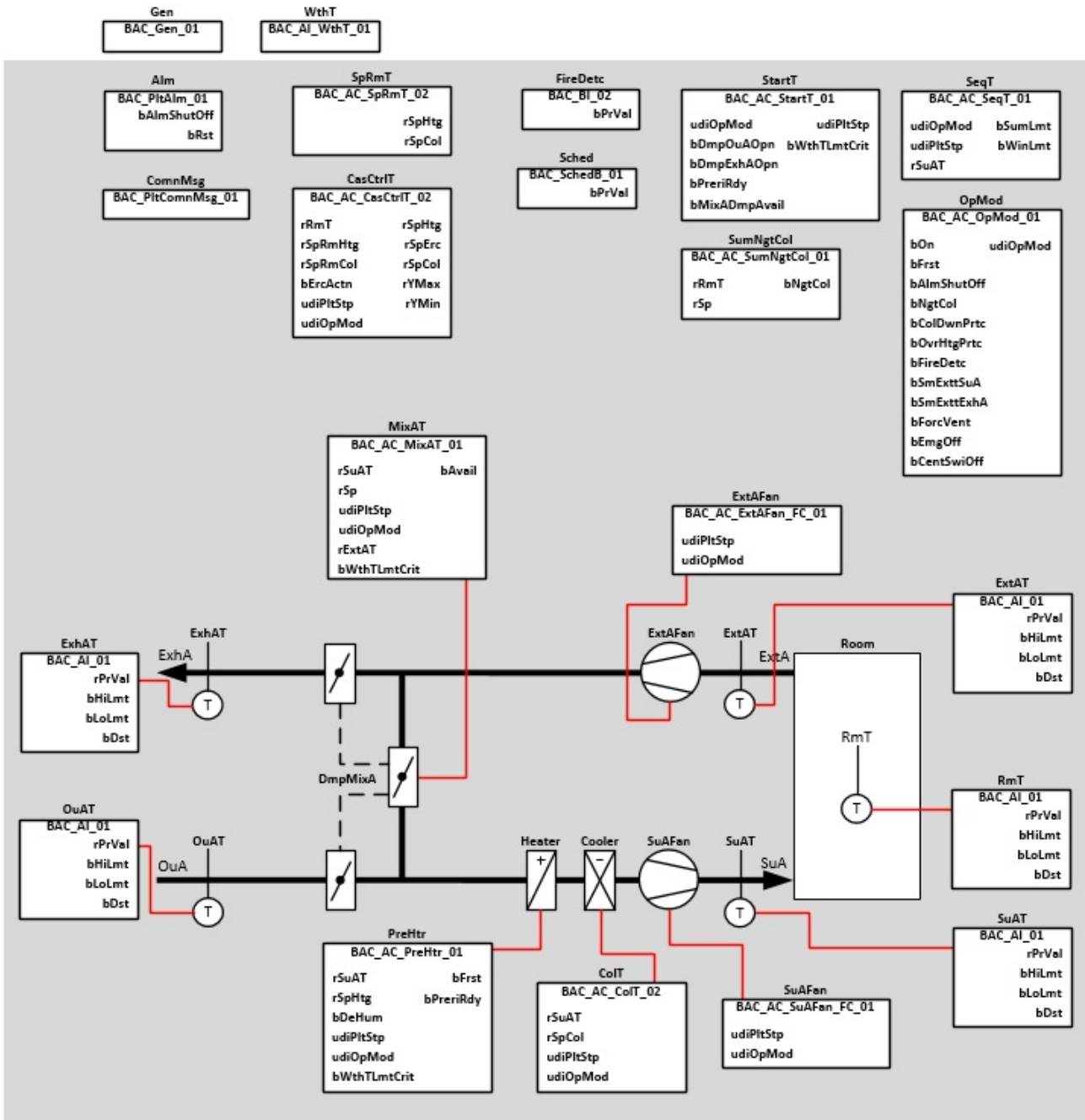
Typ: BAC_AC_SE

Strategie		Ventilator		Vorerhitzer		Kühler		Energierückgewinnung		Mischluft		
Zulufr egelun g (einfac h)	1	1- stufig	1	Nein	0	Nein	0	Nein	0	Nein	0	
Zulufr egelun g	2	2- stufig	2	Ja	1	Ja	1	Platte	1	Ja	1	
Abluft/ Zuluftk askade (einfac h)	3	Drehz ahl	3					Rekup erativ	2			
Abluft/ Zuluftk askade	4	Druck	4					Rotatio n	3			
		Volum enstro m	5									
Abluft/ Zuluftk askade (einfac h)	4	Druck	4	Vorerh itzer	1	Kühler	1	Energi erückg ewinnu ng	0	Mischl uft	1	

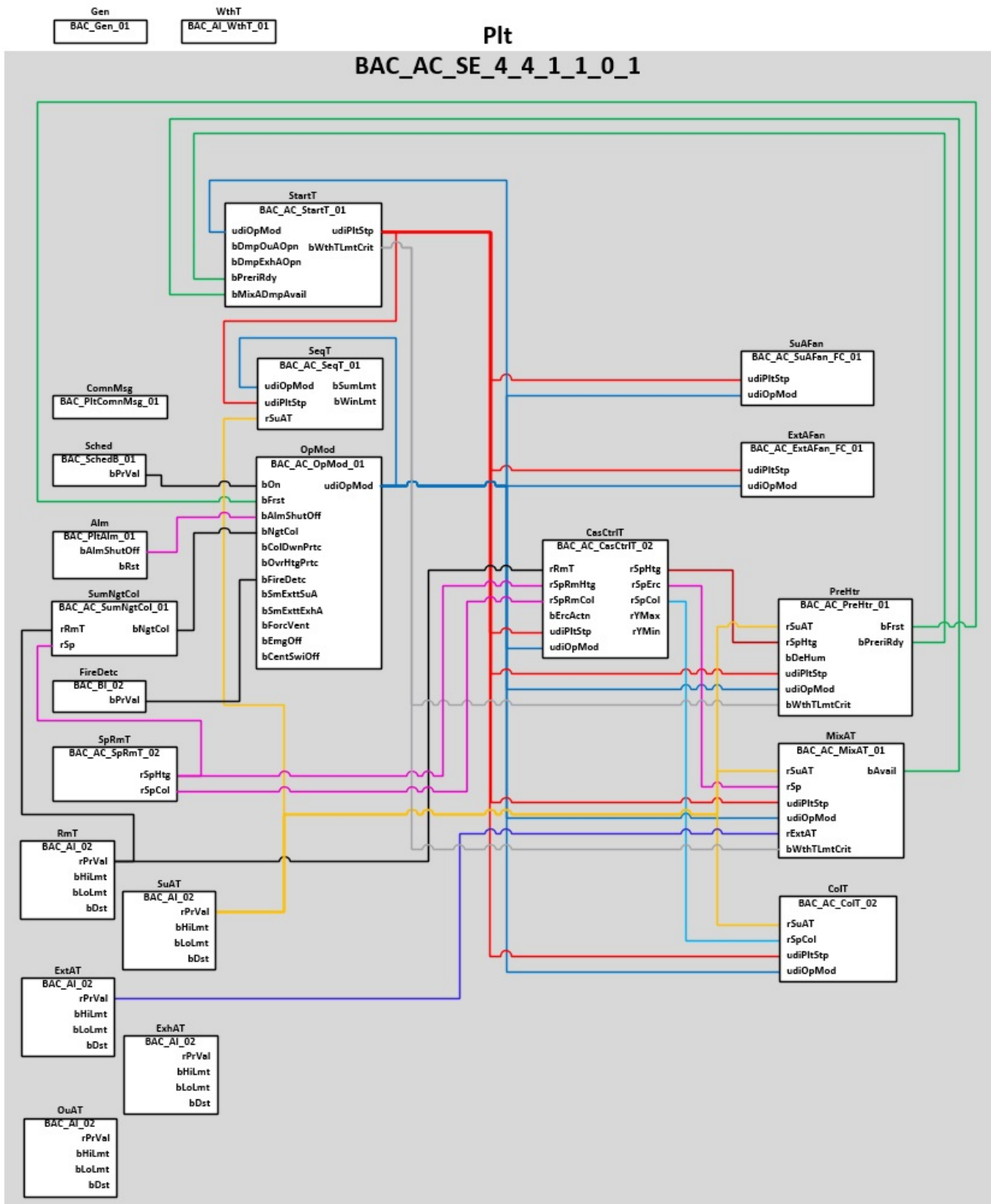
Schnittstelle

BAC_AC_SE_4_4_1_1_0_1

Anlagenschema



Blockschaltbild



Betriebsarten

Die Anlage wird eingeschaltet über ein Zeitschaltprogramm oder einen manuellen Anlagenschalter (Software).

Die Anlage kann die in der Tabelle aufgeführten Betriebsarten annehmen.

udiOpMod			
OPMOD_AC_OFF	1	Aus//Off	

OPMOD_AC_ON	2	Ein//On
OPMOD_AC_EMERG	3	Notfall//Emergency
OPMOD_AC_MANOFF	4	Hand aus//Manual off
OPMOD_AC_MANON	5	Hand ein//Manual on
OPMOD_AC_FRST	6	Frost//Frost
OPMOD_AC_SMEXTTPRG	7	Entrauchung Programm//Smoke extraction programm
OPMOD_AC_SMEXTTSUA	8	Entrauchung Zuluft//Smoke extraction supply
OPMOD_AC_SMEXTTEXHA	9	Entrauchung Fortluft//Smoke extraction exhaust
OPMOD_AC_FIRE	10	Feuer//Fire
OPMOD_AC_NGTCOL	11	Nachtkühlung//Night cooling
OPMOD_AC_COLDWNPRTC	12	Stützbetrieb,Auskühlschutz//Cool down protection
OPMOD_AC_OVRHTGPRTC	13	Überhitzungsschutz//Over heating protection
OPMOD_AC_ALM	14	Störung//Alarm
OPMOD_AC_FORCVENT	15	Zwangsbelüftung//Forced ventilation
OPMOD_AC_CENTSWIOFF	16	Zentralabschaltung//Central switch-off

Feueralarm

Bei Feueralarm werden alle Einschaltbedingungen übersteuert und die Anlage ausgeschaltet. Es werden die Entrauchungsfunktionen freigegeben. Mit den beiden binären Eingängen *SmExttSuA* / *SmExttExhA* können die 3 oben aufgeführten Entrauchungen vorgenommen werden.

Störabschaltungen

Die Anlage schaltet ab bei folgenden Störungen oder Betriebsmeldungen:

- Betriebsmeldung Feueralarm ([BAC AC StartT 01](#) [▶ 538])
- Betriebsmeldung Frostalarm ([BAC FrstPrt 01](#) [▶ 515])
- Störung der Ventilatoren ([BAC AC ExtAFan FC 01](#) [▶ 430] / [BAC AC SuAFan FC 01](#) [▶ 435])

Anfahren der Anlage

Dem Anlagenstartprogramm **StartT** wird durch die Variable **bAvail** des Mischluftsystems **MixAT** mitgeteilt, dass das Vorspülen des Vorerhitzers übersprungen wird.

Eine Rampenfunktion in dem Sub-Template **PID** ([BAC AC MixAT PID 01](#) [▶ 488]) des Mischluftsystems **MixAT** ([BAC AC MixAT 01](#) [▶ 483]) ersetzt beim Starten der Lüftungsanlage das Vorspülen des Luftherhitzers.

Temperaturregelung

In diesem Anlagentemplate wird eine Raumluft-Zuluft-Kaskade als Regelstrategie realisiert.

In den Templates [BAC AC SpRmT 02](#) [▶ 661] / [BAC AC CasCtrlT 02](#) [▶ 653] werden die Sollwerte für die Anlage erzeugt und an die Sequenzregler weiter geleitet.

Programmbeschreibung

Instanz	Typ	Aufgabe
Alm	BAC PltAlm 01 [▶ 372]	Anlagenalarming
ComnMsg	BAC PltComnMsg 01 [▶ 376]	Anlagensammelmeldungen
ExhAT	BAC AI 01 [▶ 683]	Aufruftemplate Fortlufttemperatur
ExtAT	BAC AI 01 [▶ 683]	Aufruftemplate Ablufttemperatur
OuAT	BAC AI 01 [▶ 683]	Aufruftemplate Außenlufttemperatur
RmT	BAC AI 01 [▶ 683]	Aufruftemplate Raumtemperatur
SuAT	BAC AI 01 [▶ 683]	Aufruftemplate Zulufttemperatur

Instanz	Typ	Aufgabe
Sched	BAC_SchedB_01 [▶ 718]	Aufruf-template Anlagenzeitschaltprogramm
FireDetc	BAC_BI_01 [▶ 703]	BI-Objekt zur Erfassung des Ereignisses Feueralarm
SuAFan	BAC_AC_SuAFan_FC_01 [▶ 435]	Aufruf-template mit integriertem Druckregler für einen Drehzahl gesteuerten Zuluftventilator
ExtAFan	BAC_AC_ExtAFan_FC_01 [▶ 430]	Aufruf-template mit integriertem Druckregler für einen Drehzahl gesteuerten Abluftventilator
SpRmT CasCtrlT	BAC_AC_SpRmT_02 [▶ 661] BAC_AC_CasCtrlT_02 [▶ 653]	Sollwert Raumluf-Zuluft-Kaskade
SeqT	BAC_AC_SeqT_01 [▶ 532]	Aufruf-template Sequenzsteuerung
OpMod	BAC_AC_OpMod_01 [▶ 524]	Anlagenbetriebsart
StartT	BAC_AC_StartT_01 [▶ 538]	Aufruf-template des Anlagenstartprogramms einer temperaturgeregelten Lüftungsanlage
SumNgtCol	BAC_AC_SumNgtCol_01 [▶ 549]	Nachtauskühlung
PreHtr	BAC_AC_PreHtr_01 [▶ 501]	Aufruf-template eines Warmwasserluftherhitzer
MixAT	BAC_AC_MixAT_01 [▶ 483]	Aufruf-template für die Steuerung und Regelung eines Mischluftsystems
ColT	BAC_AC_ColT_02	Aufruf-template eines temperaturgesteuerten Kaltwasser-Luftkühlers (Ventil)

Anlagenalarmaufnahme

Die Alarmpriorität ALM_PRIO_CRITICAL ist eine anlagenabschaltende Störung.

Template	Subtemplate	Instanz	Typ	Alarmpriorität
ExhAT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_AL ARM
ExtAT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_AL ARM
OuAT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_AL ARM
RmT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_AL ARM
SuAT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_AL ARM
SuAFan	SuAFanFU	AlmThOvrid	FB_BA_Alarm [▶ 182]	ALM_PRIO_CR ITICAL
		AlmDstFC	FB_BA_Alarm [▶ 182]	ALM_PRIO_CR ITICAL
		AlmMntnSwi	FB_BA_Alarm [▶ 182]	ALM_PRIO_AL ARM
	SuAFanDiffPrss Monit	AlmDiffPrss	FB_BA_Alarm [▶ 182]	ALM_PRIO_CR ITICAL

Template	Subtemplate	Instanz	Typ	Alarmpriorität	
ExtAFan	ExtAFanFU	AlmThOvrlD	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmDstFC	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	ExtAFanDiffPrssMonit	AlmDiffPrss	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
PreHtr	PreHtrRetWtrT	AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstThermostat	AlmBI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstPrt	AlmFrst	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstPrtT	AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrPu	AlmDst	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrVlv	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	
MixAT	DmpMixA	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	
	DmpOuA	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	
	DmpExhA	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	
CoIT	CoITVlv	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	

Versionshistorie

Versionsnummer	Bemerkungen
1.0.1.0	erste Freigabe

9.79.4 BAC_AC_SE_4_4_1_1_3_0

Anwendung

Anlagenaufrufftemplate einer temperaturgeregelten Lüftungsanlage.

Die wesentlichen Aufgaben des Templates sind:

- Anlagensteuerung mit den verschiedenen Betriebsarten und Sollwerten
- Temperaturregelung für die ausgewählten Sequenzelemente

Anlagenschlüssel

Lüftungsanlage mit Zu- und Abluftventilator und thermischer Luftbehandlung

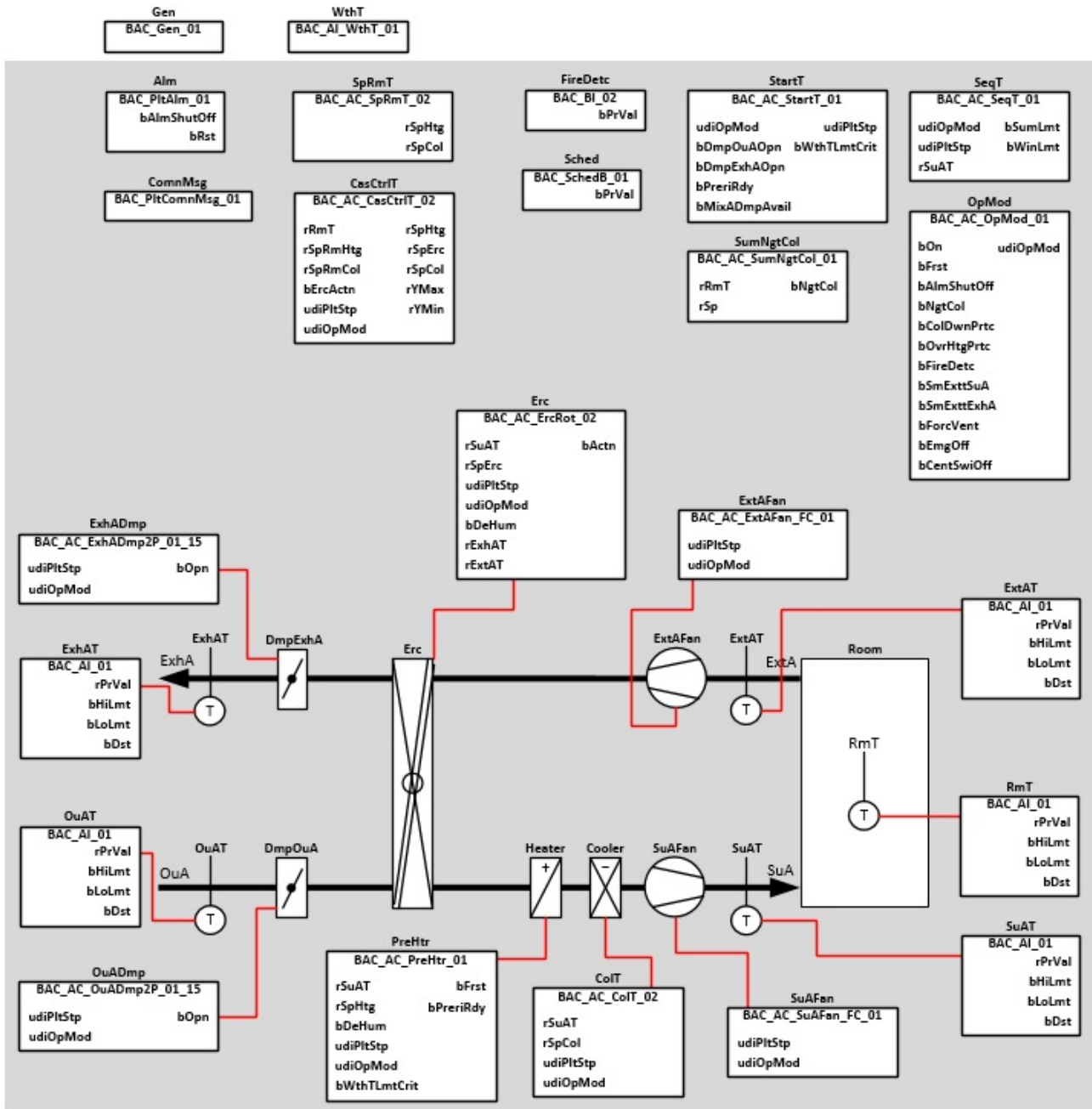
Typ: BAC_AC_SE

Strategie		Ventilator		Vorerhitzer		Kühler		Energierückgewinnung		Mischluft		
Zulufr egelun g (einfac h)	1	1- stufig	1	Nein	0	Nein	0	Nein	0	Nein	0	
Zulufr egelun g	2	2- stufig	2	Ja	1	Ja	1	Platte	1	Ja	1	
Abluft/ Zuluftk askade (einfac h)	3	Drehz ahl	3					Rekup erativ	2			
Abluft/ Zuluftk askade	4	Druck	4					Rotatio n	3			
		Volum enstro m	5									
Abluft/ Zuluftk askade (einfac h)	4	Druck	4	Vorerh itzer	1	Kühler	1	Energi erückg ewinnu ng	3	Mischl uft	0	

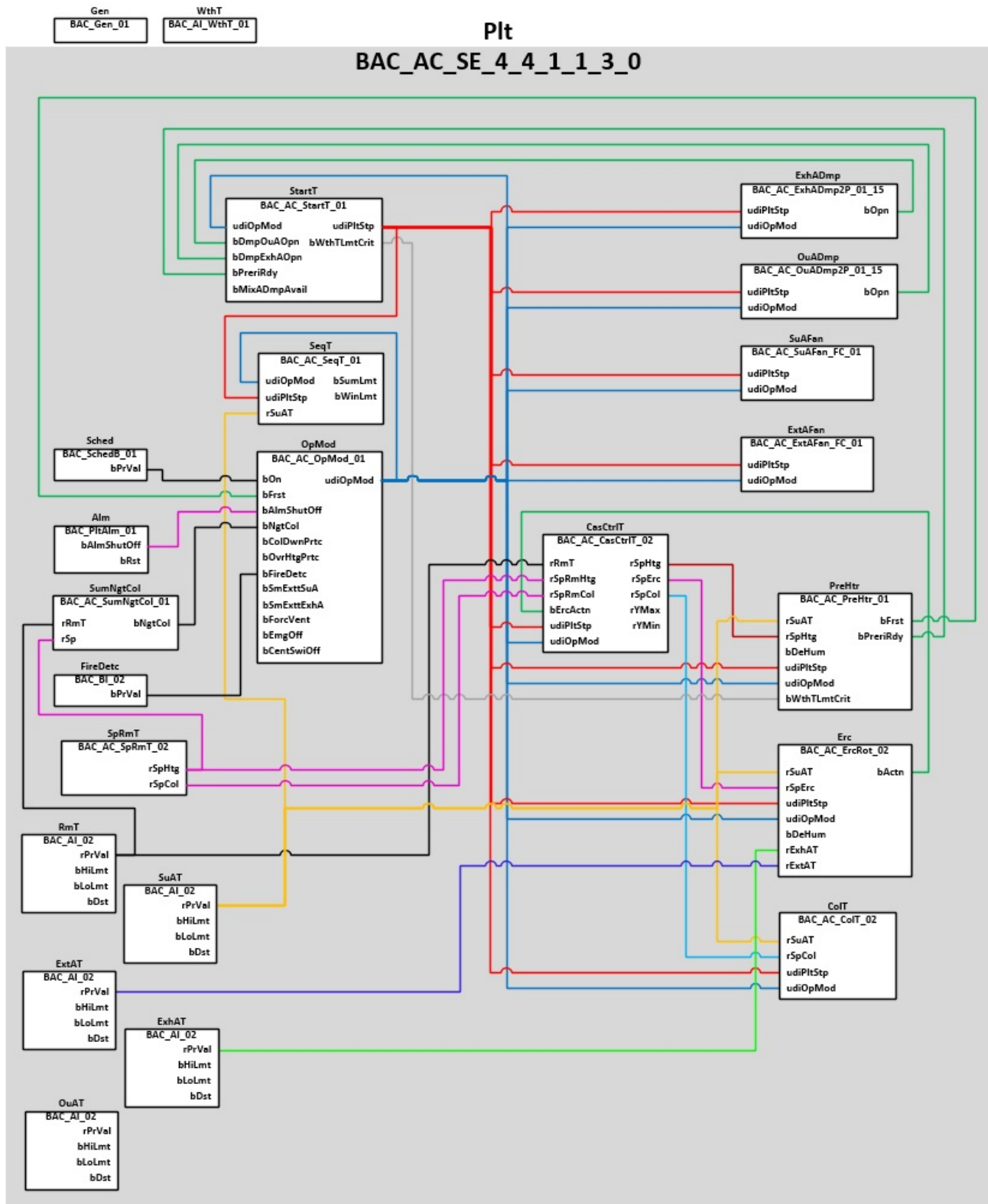
Schnittstelle

BAC_AC_SE_4_4_1_1_3_0

Anlagenschema



Blockschaltbild



Betriebsarten

Die Anlage wird eingeschaltet über ein Zeitschaltprogramm oder einen manuellen Anlagenschalter (Software).

Die Anlage kann die in der Tabelle aufgeführten Betriebsarten annehmen.

udiOpMod		
OPMOD_AC_OFF	1	Aus//Off

OPMOD_AC_ON	2	Ein//On
OPMOD_AC_EMERG	3	Notfall//Emergency
OPMOD_AC_MANOFF	4	Hand aus//Manual off
OPMOD_AC_MANON	5	Hand ein//Manual on
OPMOD_AC_FRST	6	Frost//Frost
OPMOD_AC_SMEXTTPRG	7	Entrauchung Programm//Smoke extraction programm
OPMOD_AC_SMEXTTSUA	8	Entrauchung Zuluft//Smoke extraction supply
OPMOD_AC_SMEXTTEXHA	9	Entrauchung Fortluft//Smoke extraction exhaust
OPMOD_AC_FIRE	10	Feuer//Fire
OPMOD_AC_NGTCOL	11	Nachtkühlung//Night cooling
OPMOD_AC_COLDWNPRTC	12	Stützbetrieb,Auskühlschutz//Cool down protection
OPMOD_AC_OVRHTGPRTC	13	Überhitzungsschutz//Over heating protection
OPMOD_AC_ALM	14	Störung//Alarm
OPMOD_AC_FORCVENT	15	Zwangselüftung//Forced ventilation
OPMOD_AC_CENTSWIOFF	16	Zentralabschaltung//Central switch-off

Feueralarm

Bei Feueralarm werden alle Einschaltbedingungen übersteuert und die Anlage ausgeschaltet. Es werden die Entrauchungsfunktionen freigegeben. Mit den beiden binären Eingängen *SmExttSuA* / *SmExttExhA* können die 3 oben aufgeführten Entrauchungen vorgenommen werden.

Störabschaltungen

Die Anlage schaltet ab bei folgenden Störungen oder Betriebsmeldungen:

- Betriebsmeldung Feueralarm ([BAC AC StartT 01 \[▶ 538\]](#))
- Betriebsmeldung Frostalarm ([BAC FrstPrt 01 \[▶ 515\]](#))
- Störung der Ventilatoren ([BAC AC ExtAFan FC 01 \[▶ 430\]](#) / [BAC AC SuAFan FC 01 \[▶ 435\]](#))
- Störung der Außen- oder Fortluftklappe ([BAC AC OuADmp2P 01 15 \[▶ 454\]](#) / [BAC AC ExhADmp2P 01 15 \[▶ 446\]](#))
- Störung des Rotationswärmetauschers ([BAC AC ErcRot 02 \[▶ 417\]](#))

Anfahren der Anlage

Beim Einschalten der Anlage bei kalter Witterung wird im Anlagenschritt Vorspülen [PLTSTP AC PRERI \[▶ 364\]](#) die Regelung der Rücklauftemperatur des Warmwasserluftherhizers aktiviert. Dieser Vorgang soll das Einfrieren des Warmwasserluftherhizer während des Anlagenstarts verhindern.

Temperaturregelung

In diesem Anlagentemplate wird eine Raumluft-Zuluft-Kaskade als Regelstrategie realisiert.

In den Templates [BAC AC SpRmT 02 \[▶ 661\]](#) / [BAC AC CasCtrlT 02 \[▶ 653\]](#) werden die Sollwerte für die Anlage erzeugt und an die Sequenzregler weiter geleitet.

Programmbeschreibung

Instanz	Typ	Aufgabe
Alm	BAC PltAlm 01 [▶ 372]	Anlagenalariming
ComnMsg	BAC PltComnMsg 01 [▶ 376]	Anlagensammelmeldungen
ExhAT	BAC AI 01 [▶ 683]	Aufruftemplate Fortlufttemperatur
ExtAT	BAC AI 01 [▶ 683]	Aufruftemplate Ablufttemperatur

Instanz	Typ	Aufgabe
OuAT	BAC_AI_01 [▶ 683]	Aufruf-template Außenlufttemperatur
RmT	BAC_AI_01 [▶ 683]	Aufruf-template Raumtemperatur
SuAT	BAC_AI_01 [▶ 683]	Aufruf-template Zulufttemperatur
Sched	BAC_SchedB_01 [▶ 718]	Aufruf-template Anlagenzeitschaltprogramm
FireDetc	BAC_BI_01 [▶ 703]	BI-Objekt zur Erfassung des Ereignisses Feueralarm
SuAFan	BAC_AC_SuAFan_FC_01 [▶ 435]	Aufruf-template mit integriertem Druckregler für einen Drehzahl gesteuerten Zuluftventilator
ExtAFan	BAC_AC_ExtAFan_FC_01 [▶ 430]	Aufruf-template mit integriertem Druckregler für einen Drehzahl gesteuerten Abluftventilator
ExhADmp	BAC_AC_ExhADmp2P_01_15 [▶ 446]	Aufruf-template Fortluftklappe mit einem Federrücklaufantrieb und Endlagenkontrolle
OuADmp	BAC_AC_OuADmp2P_01_15 [▶ 454]	Aufruf-template Außenluftklappe mit einem Federrücklaufantrieb und Endlagenkontrolle
SpRmT CasCtrlT	BAC_AC_SpRmT_02 [▶ 661] BAC_AC_CasCtrlT_02 [▶ 653]	Sollwert Raumluft-Zuluft-Kaskade
SeqT	BAC_AC_SeqT_01 [▶ 532]	Aufruf-template Sequenzsteuerung
OpMod	BAC_AC_OpMod_01 [▶ 524]	Anlagenbetriebsart
Start	BAC_AC_StartT_01 [▶ 538]	Aufruf-template des Anlagenstartprogramms einer temperaturgeregelten Lüftungsanlage
SumNgtCo l	BAC_AC_SumNgtCol_01 [▶ 549]	Nachtauskühlung
PreHtr	BAC_AC_PreHtr_01 [▶ 501]	Aufruf-template eines Warmwasserluftherhitzer
Erc	BAC_AC_ErcRot_02 [▶ 417]	Aufruf-template für die Ansteuerung eines Rotationswärmetauschers für die Energierückgewinnung einer Lüftungsanlage.
ColT	BAC_AC_ColT_02	Aufruf-template eines temperaturgesteuerten Kaltwasser-Luftkühlers (Ventil)

Anlagenalarmaufnahme

Die Alarmpriorität ALM_PRIO_CRITICAL ist eine anlagenabschaltende Störung.

Template	Subtemplate	Instanz	Typ	Alarmpriorität
ExhAT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_ALARM
ExtAT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_ALARM
OuAT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_ALARM
RmT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_ALARM
SuAT		AlmA1	FB_BA_Alarm [▶ 182]	ALM_PRIO_ALARM

Template	Subtemplate	Instanz	Typ	Alarmpriorität	
SuAFan	SuAFanFU	AlmThOvrlid	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmDstFC	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	SuAFanDiffPrssMonit	AlmDiffPrss	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
ExtAFan	ExtAFanFU	AlmThOvrlid	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmDstFC	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	ExtAFanDiffPrssMonit	AlmDiffPrss	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
ExhADmp		AlmFnct	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
OuADmp		AlmFnct	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
PreHtr	PreHtrRetWtrT	AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstThermostat	AlmBI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstPrt	AlmFrst	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrFrstPrtT	AlmAI	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrPu	AlmDst	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
	PreHtrVlv	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	
Erc	ErcMot	AlmDstFC	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmThOvrlid	FB BA Alarm [▶ 182]	ALM_PRIO_CRITICAL	
		AlmMntnSwi	FB BA Alarm [▶ 182]	ALM_PRIO_ALARM	
CoIT	CoITVlv	AlmFdbMonit	FB BA Alarm [▶ 182]	ALM_PRIO_WARNING	

Versionshistorie

Versionsnummer	Bemerkungen
1.0.1.0	erste Freigabe

9.79.5 BAC_AC_Sx_001

Anwendung

Das Aufruftemplate ist ein Anlagengrundprogramm für eine Lüftungsanlage. Es beinhaltet Basiselemente für folgende Lüftungsanlagentypen:

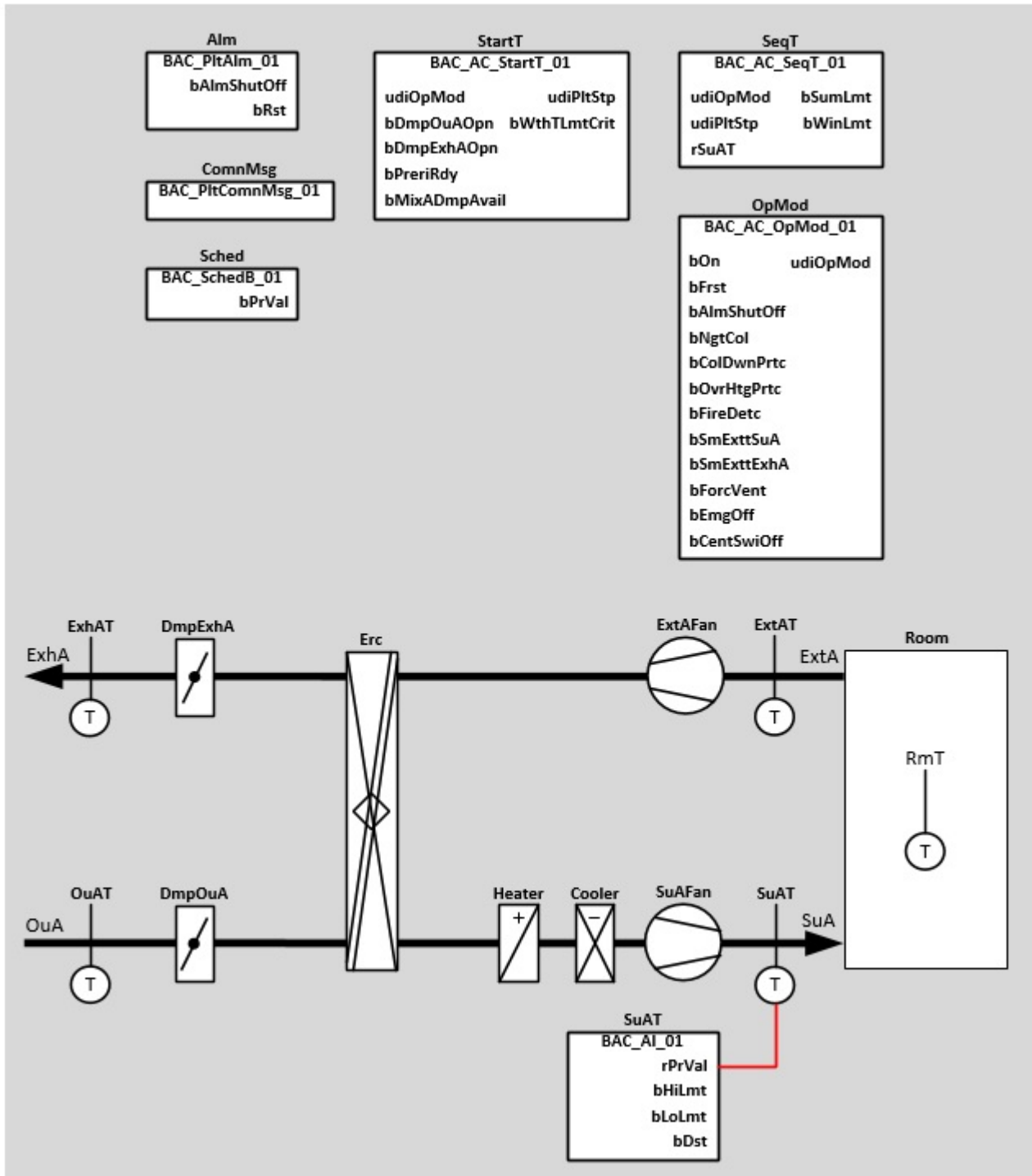
- **Lüftungsanlagen mit Zu- und Abluftventilator und thermischer Luftbehandlung**
- **Lüftungsanlagen mit Zuluftventilator und thermischer Luftbehandlung**

Die fehlenden Aggregate einer Lüftungsanlage müssen dem Anlagengrundprogramm hinzugefügt, aufgerufen und verknüpft werden, siehe Beispiel in dem Anlagenbeispiel [BAC AC SE 3 4 1 1 1 0 \[▶ 619\]](#).

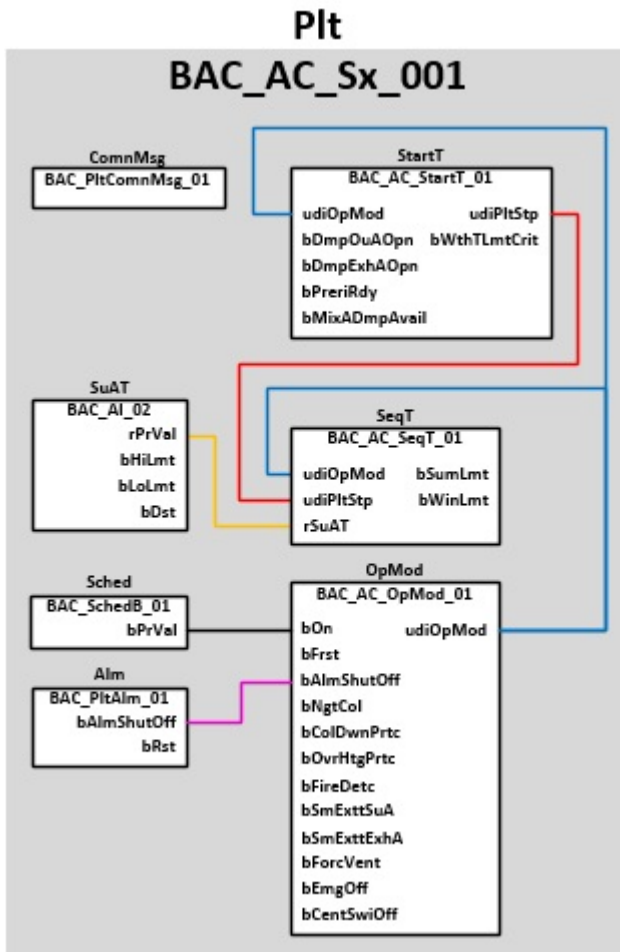
Schnittstelle

BAC_AC_Sx_001

Anlagenschema



Blackschaltbild



Betriebsarten

Die Anlage wird eingeschaltet über ein Zeitschaltprogramm oder einen manuellen Anlagenschalter (Software).

Die Anlage kann die in der Tabelle aufgeführten Betriebsarten annehmen.

udiOpMod		
OPMOD_AC_OFF	1	Aus//Off
OPMOD_AC_ON	2	Ein//On
OPMOD_AC_EMERG	3	Notfall//Emergency
OPMOD_AC_MANOFF	4	Hand aus//Manual off
OPMOD_AC_MANON	5	Hand ein//Manual on
OPMOD_AC_FRST	6	Frost//Frost
OPMOD_AC_SMEXTTPRG	7	Entrauchung Programm//Smoke extraction programm
OPMOD_AC_SMEXTTSUA	8	Entrauchung Zuluft//Smoke extraction supply
OPMOD_AC_SMEXTTEXHA	9	Entrauchung Fortluft//Smoke extraction exhaust
OPMOD_AC_FIRE	10	Feuer//Fire
OPMOD_AC_NGTCOL	11	Nachtkühlung//Night cooling
OPMOD_AC_COLDWNPRTC	12	Stützbetrieb,Auskühlschutz//Cool down protection
OPMOD_AC_OVRHTGPRTC	13	Überhitzungsschutz//Over heating protection
OPMOD_AC_ALM	14	Störung//Alarm
OPMOD_AC_FORCVENT	15	Zwangsbelüftung//Forced ventilation
OPMOD_AC_CENTSWIOFF	16	Zentralabschaltung//Central switch-off

Programmbeschreibung

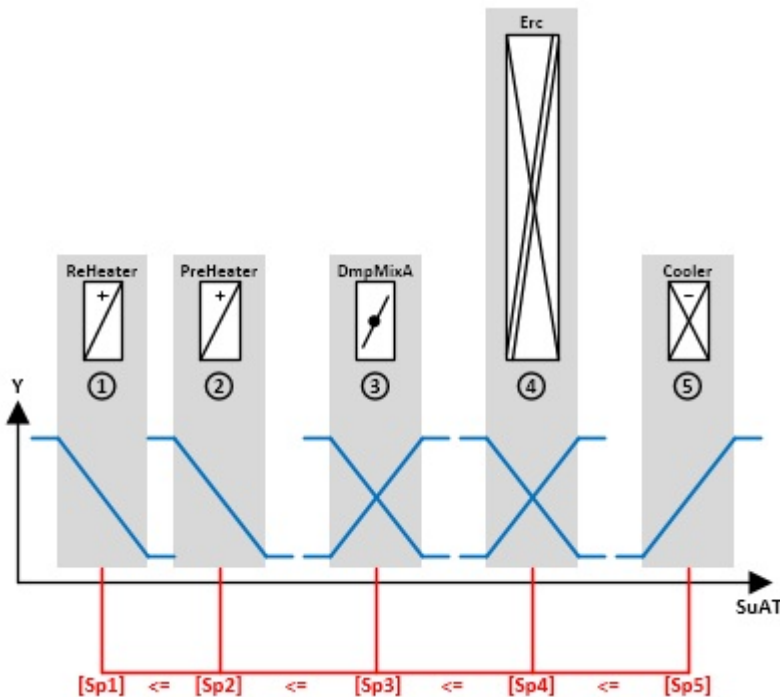
Instanz	Typ	Aufgabe
Alm	BAC_PltAlm_01 [▶ 372]	Anlagenalarming
ComnMsg	BAC_PltComnMsg_01 [▶ 376]	Anlagensammelmeldungen
SuAT	BAC_AI_01 [▶ 683]	Aufruftemplate Zulufttemperatur
Sched	BAC_SchedB_01 [▶ 718]	Aufruftemplate Anlagenzeitschaltprogramm
SeqT	BAC_AC_SeqT_01 [▶ 532]	Aufruftemplate Sequenzsteuerung
OpMod	BAC_AC_OpMod_01 [▶ 524]	Anlagenbetriebsart
Start	BAC_AC_StartT_01 [▶ 538]	Aufruftemplate des Anlagenstartprogramms einer temperaturgeregelten Lüftungsanlage

Versionshistorie

Versionsnummer	Bemerkungen
1.0.1.0	erste Freigabe

9.80 Sollwerte der Sequenz-Regler

In den Sequenzreglern müssen die Sollwerte monoton steigend sein: [Sp1] <= [Sp2] <= [Sp3] <= [Sp4] <= [Sp5] <= ... <= [Spn]



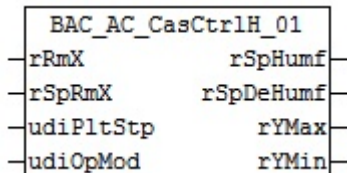
[Sp1]	rPrSpHtg	Sollwert Heizen
[Sp2]	rPrSpHtg	Sollwert Heizen
[Sp3]	rPrSpErc	Sollwert Mischluftklappen
[Sp4]	rPrSpErc	Sollwert Energierückgewinnung
[Sp5]	rPrSpCol	Sollwert Kühlen

9.80.1 BAC_AC_CasCtrlH_01

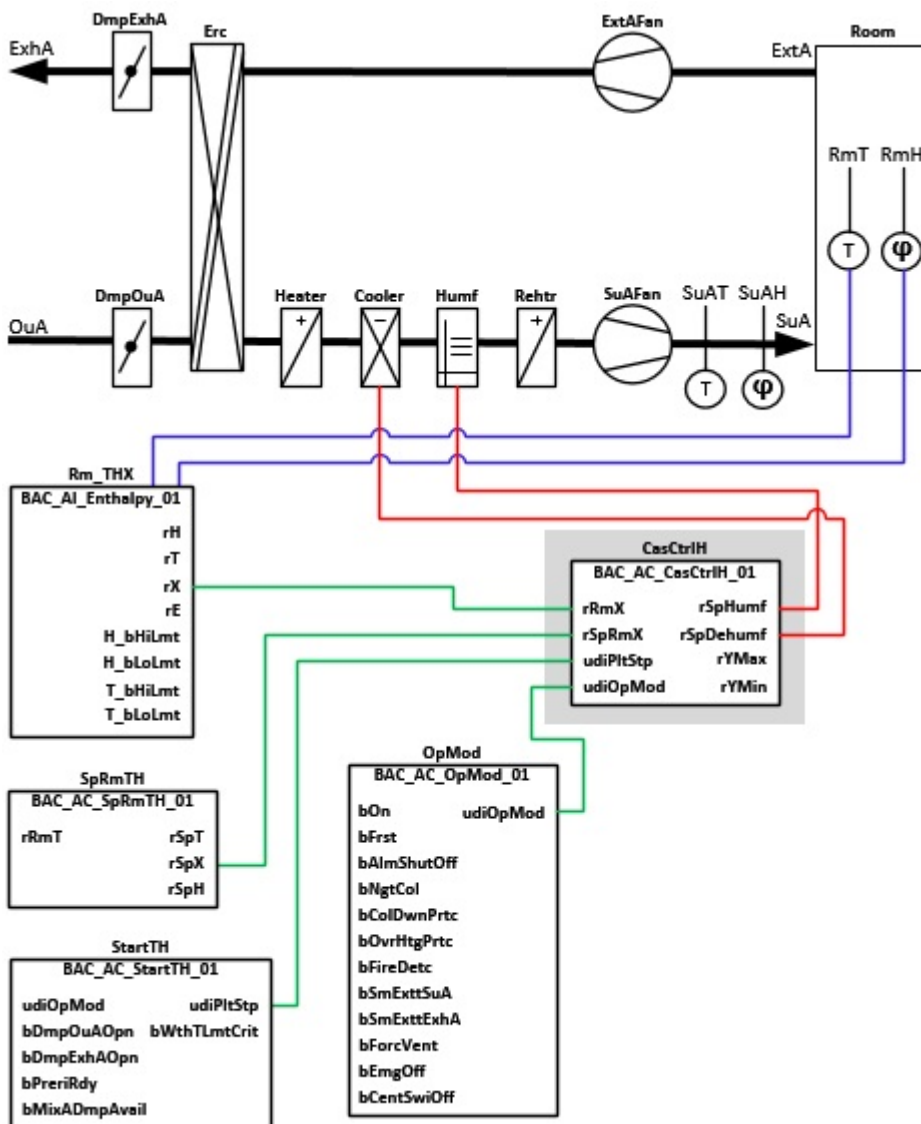
Anwendung

Das Template ist ein Kaskadenregler für die Zuluftfeuchte bestehend aus einem Führungsregler zur Sollwertberechnung für das Be- und Entfeuchten.

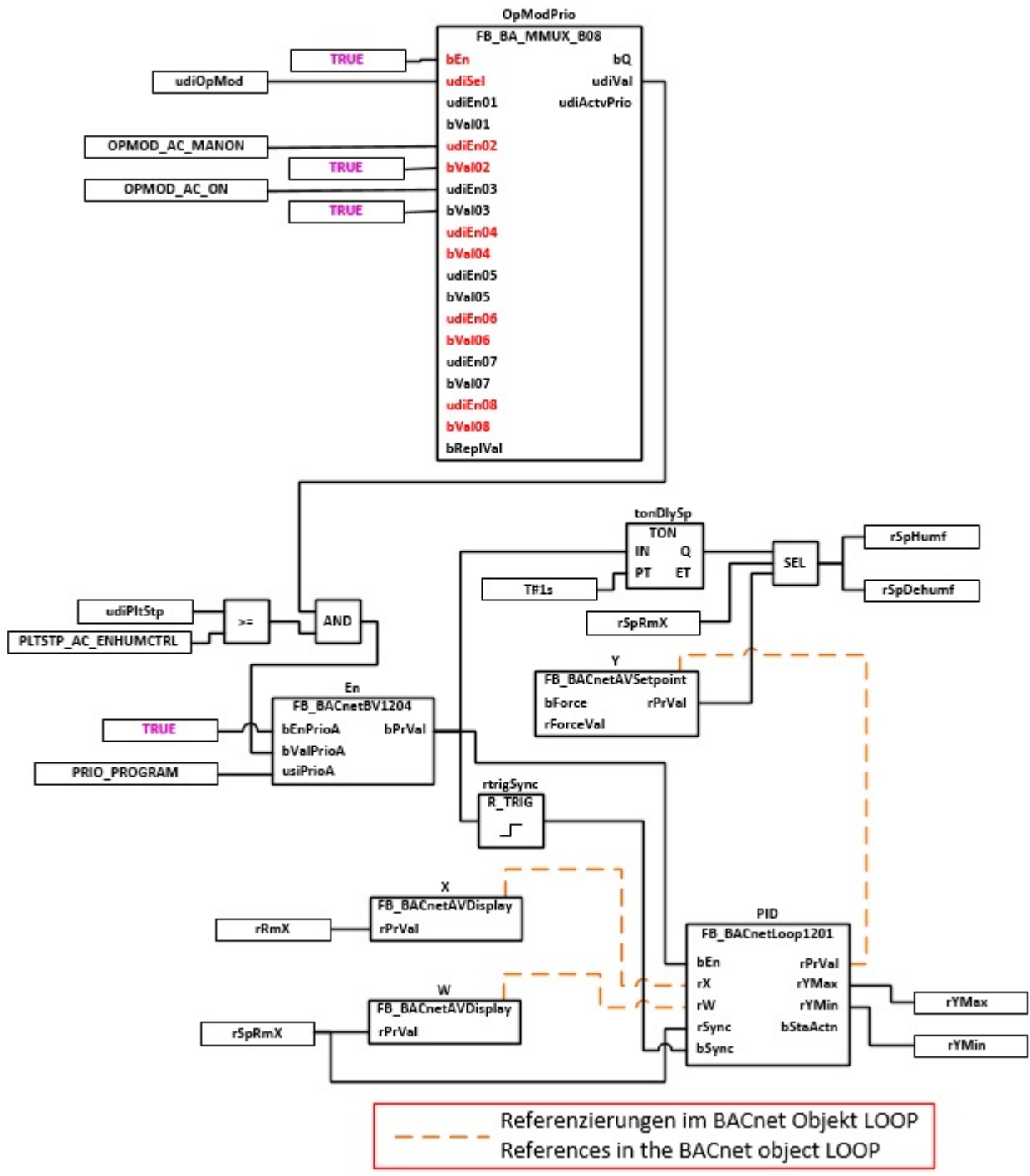
Schnittstelle



Anlagenschema 01



Blockschaltbild



--- Referenzierungen im BACnet Objekt LOOP
References in the BACnet object LOOP

VAR_INPUT

```
rRmX      : REAL;
rSpRmX    : REAL;
udiPltStp : UDINT;
udiOpMod  : UDINT;
```

rRmX: Istwert Raum absolute Feuchte in g/Kg

rSpRmX: Raumsollwert (Entfeuchten) in g/Kg

udiPltStp: Schritte beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartTH 01 \[▶ 543\]](#)

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod 01 \[▶ 524\]](#)

VAR_OUTPUT

```
rSpHumf      : REAL;
rSpDeHumf    : REAL;
rYMax        : REAL;
rYMin        : REAL;
```

rSpHumf: Zuluftsollwert Befeuchten g/Kg

rSpDehumf: Zuluftsollwert Entfeuchten g/Kg

rYMax: Oberer Wert der Regler-Ausgangsbegrenzung

rYMin: Unterer Wert der Regler-Ausgangsbegrenzung

VAR CONSTANT

```
PLT_NUM      : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [[182](#)]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [[372](#)] mittels des Funktionsbausteins **FB_BA_AlarmPit**. [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PitComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe									
X	FB_BACnetAVDisplay [69]	An das AV-Objekt ist die Eingangsvariable rRmX angeschlossen. Es ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts PID .									
W	FB_BACnetAVDisplay [69]	An das AV-Objekt ist die Eingangsvariable rSpRmX angeschlossen. Es ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts PID .									
OpModPri o	FB_BA_MMUX_B08 [208]	Der Multiplexer definiert die Freigabebedingungen des PID-Führungsreglers in Abhängigkeit der Anlagenbetriebsart udiOpMod . <table border="1" data-bbox="555 1496 1441 1615"> <thead> <tr> <th colspan="2">udiOpMod</th> <th>Freigabe Kaskadenregelung</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_MANON</td> <td>Hand ein</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_ON</td> <td>Ein</td> <td>TRUE</td> </tr> </tbody> </table>	udiOpMod		Freigabe Kaskadenregelung	OPMOD_AC_MANON	Hand ein	TRUE	OPMOD_AC_ON	Ein	TRUE
udiOpMod		Freigabe Kaskadenregelung									
OPMOD_AC_MANON	Hand ein	TRUE									
OPMOD_AC_ON	Ein	TRUE									
En	FB_BACnetBV1204 [94] >=, AND	Das BV-Objekt dient zur Anzeige der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay Das Ergebnis dieses Netzwerkes ist die Freigabe des PID-Führungsreglers. Die Freigabe ist abhängig von der Anlagenbetriebsart udiOpMod und der Anlagenschritte udiPitStp beim Anfahren der Lüftungsanlage.									
PID	FB_BACnetLoop1201 [99]	Führungsregler für eine Raumtemperaturregelung mittels Raum-Zuluft-Kaskade. Es liefert den einfachen Zulufttemperatur-Sollwert.									
rtrigSync	R_TRIG	Mit Erhalt der Reglerfreigabe En löst rtrigSync die Synchronisation des Führungsreglers PID auf dessen Eingangswert IrSync aus.									
Y	FB_BACnetAVSetpoint [70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts PID . Es zeigt den Zuluftfeuchte-Sollwert an.									

Instanz	Typ	Aufgabe
TLogY	FB_BACnetTLog1201 ▶ 137]	Trendaufzeichnung des Zuluftfeuchte-Sollwertes Y
tonDlySp	TON SEL	Nach Erhalt der Reglerfreigabe En gibt dieses Netzwerkes verzögert den Zuluftfeuchte-Sollwert aus.

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.2 BAC_AC_CasCtrlH_02

Funktionsbeschreibung

Das Template ist der Führungsregler für eine Raum- bzw. Abluft/Zuluft-Kaskadenfeuchtereugung.

Aufgabe der Führungsreglung ist die Raumfeuchte innerhalb des Behaglichkeitsfeldes zwischen einem oberen und unteren Sollwert zu halten.

Zur Begrenzung der Raumfeuchte nach unten dient das BACnet-Loop-Objekt **PID_Humf**. Es berechnet den Zuluftfeuchtesollwert für die Befeuchtung.

Zur Begrenzung der Raumfeuchte nach oben dient das BACnet-Loop- Objekt **PID_DeHumf**. Es berechnet den Zuluftfeuchtesollwert für die Entfeuchtung.

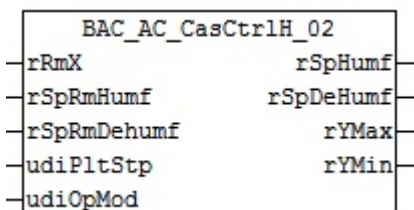
Beide Regler erhalten die Raumfeuchtesollwerte vom vorgelagerten Sollwert-Template.

Sollwerte für die Raumfeuchte als auch die durch die Loop-Objekte errechneten Sollwerte für die Zuluftfeuchte werden in g/Kg also in Absoluter Feuchte angegeben.

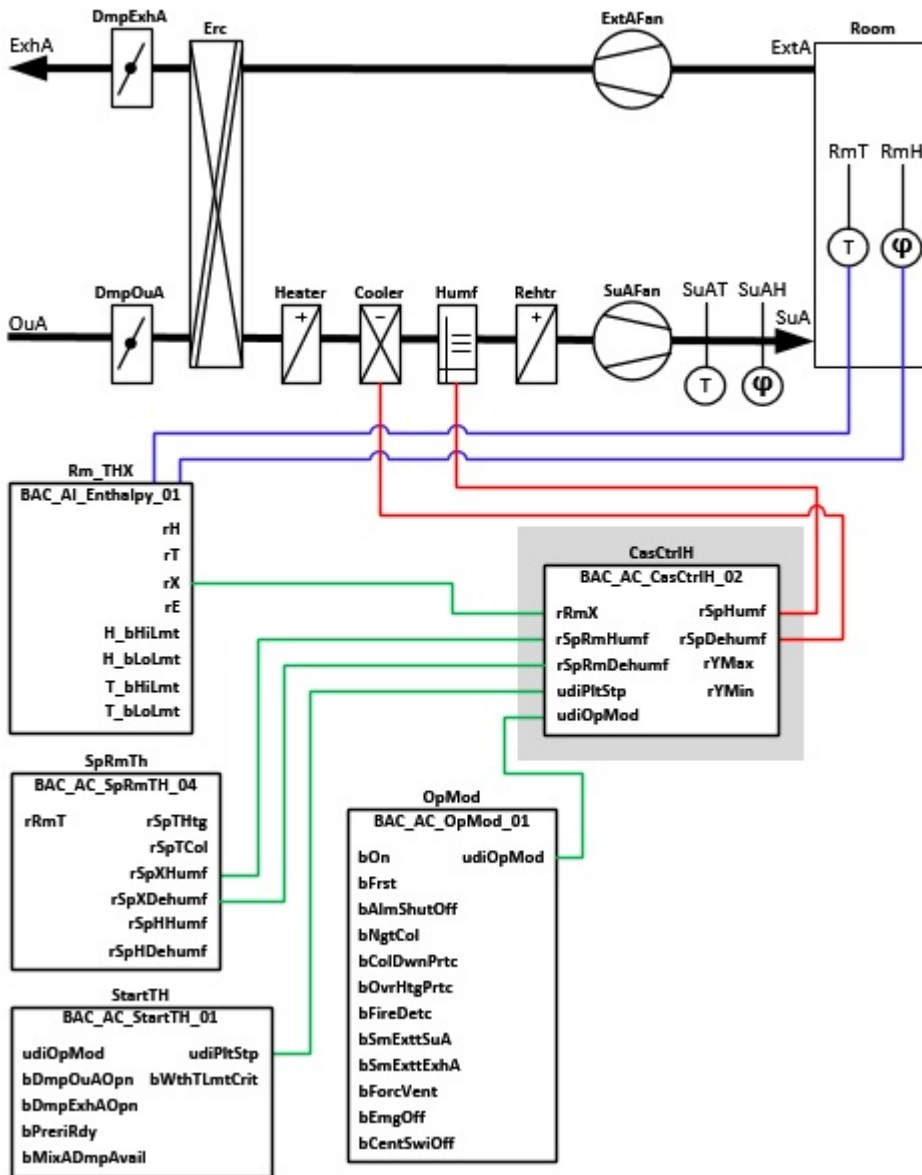
Die Parameter YMin, YMax, Verstärkung und Integralzeit der beiden Loop- Objekte werden durch den Baustein **PID_Sync** stets synchronisiert. Durch die Differenz des unteren und oberen Sollwertes sind die Ausgangskennlinien der beiden PI-Regler stets parallel verschoben. Eine Überschneidung der Zuluftsollwerte für Be- und Entfeuchten ist so ausgeschlossen.

Wichtig ist sicherzustellen, dass **rSpRmHumf** immer kleiner oder gleich **rSpRmDeHumf** ist. Dieses geschieht in dem vorgelagerten Sollwert-Template.

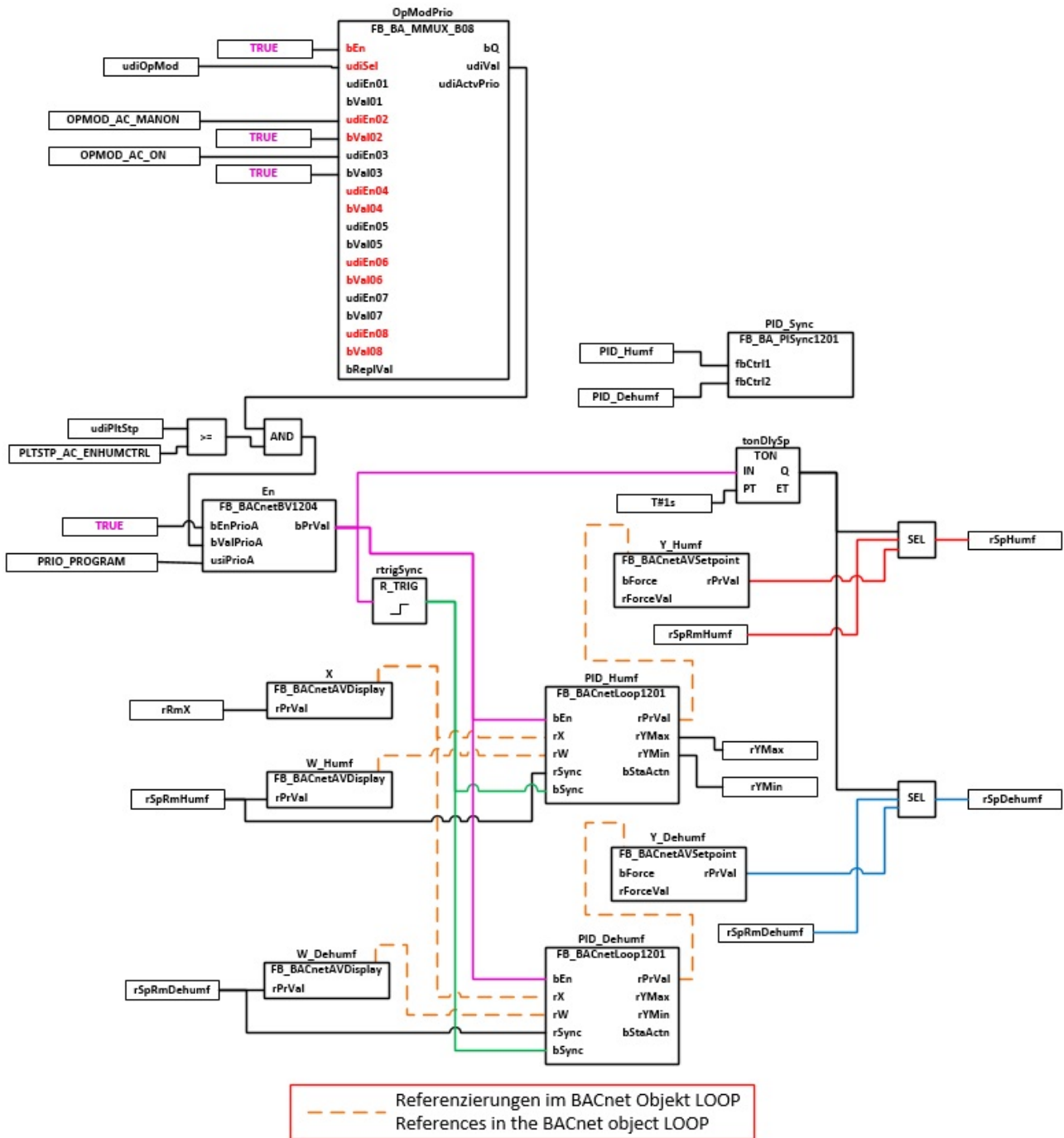
Schnittstelle



Anlagenbild



Blockschaltbild



VAR_INPUT

```

rRmX      : REAL;
rSpRmHumf : Real;
rSpRmDehumf : Real;
udiPltStp : REAL;
udiOpMod  : REAL;
    
```

rRmX: Istwert Raum absolute Feuchte in g/Kg

rSpRmHumf: Untere Raumsollwert (Befeuchten) in g/Kg

rSpRmDehumf: Oberer Raumsollwert (Entfeuchten) in g/Kg

udiPltStp: Stufen des Anlagenstarts beim Anfahren der RLT-Anlage. Siehe auch [BAC AC StartTH_01](#) [► 543].

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod_01](#) [► 524].

VAR_OUTPUT

```
rY_Hum      : REAL;
rY_DeHum    : REAL;
```

rY_Hum: Zuluftsollwert Befeuchten g/Kg

rY_DeHum: Zuluftsollwert Entfeuchten g/Kg

VAR CONSTANT

```
PLT_NUM     : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarmergebnisse einer Anlage, z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_CmnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe
X	FB_BACnetAVDisplay [► 69]	Das AV-Objekt ist referenziert auf den Istwert-Eingang der BACnet-Loop-Objekte
W_Humf	FB_BACnetAVDisplay [► 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts PID_Humf
W_Dehumf	FB_BACnetAVDisplay [► 69]	Das AV-Objekt ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts PID_Dehumf
En	FB_BACnetBV1204 [► 94]	Das BV-Objekt dient zur Anzeige und Aktivierung der Reglerfreigaben in der MBE oder in einen lokalen Bediendisplay.
	>=, AND	Das Ergebnis dieses Netzwerkes ist die Freigabe der beiden PID-Führungsregler PID_Humf/PID_Dehumf . Die Freigabe ist abhängig von der Anlagenbetriebsart udiOpMod und der Anlagenschritte udiPitStp beim Anfahren der Lüftungsanlage.
PID_Humf	FB_BACnetLoop1201 [► 99]	Führungsregler Befeuchten
PID_Dehumf	FB_BACnetLoop1201 [► 99]	Führungsregler Entfeuchten
rtrigSync	R_TRIG	Nach Erhalt der Reglerfreigabe bEn werden durch eine steigenden Flanke ausgelöst durch rtrigSync die beiden Führungsregler auf die Eingangssollwerte W_Humf_rPrVal / W_Dehumf_rPrVal aufsynchronisiert.
Y_Humf	FB_BACnetAVSetpoint [► 70]	Zuluftsollwert Befeuchten. Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts PID_Humf
Y_Dehumf	FB_BACnetAVSetpoint [► 70]	Zuluftsollwert Entfeuchten. Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts PID_Dehumf

Instanz	Typ	Aufgabe
TLogY_Humf	FB_BACnetTLog1201 [▶_137]	Trendaufzeichnung des Zuluftsollwertes heizen Y_Humf
TLogY_Dehumf	FB_BACnetTLog1201 [▶_137]	Trendaufzeichnung des Zuluftsollwertes heizen Y_Humf
PID_Sync	FB_BA_PISync1201 [▶_164]	Der Funktionsbaustein führt eine Parameter-Synchronisation der beiden Führungsregler PID_Humf/PID_Dehumf durch
tonDlySp	TON SEL SEL	Netzwerk zur Ausgabe der Sollwerte an die Ausgänge des Templates Nach Erhalt der Freigabe bEn werden um eine Sekunde verzögert tonDlySp die Sollwerte der Führungsregler an den Ausgängen des Templates ausgegeben rSpHumf / rSpDehumf . Im nicht freigegebenen Zustand der Anlage werden die Eingangssollwerte des Templates ausgegeben.

Versionshistorie

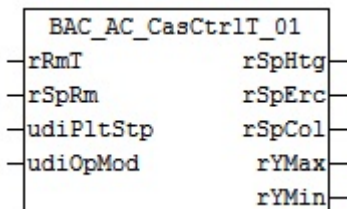
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.3 BAC_AC_CasCtrlT_01

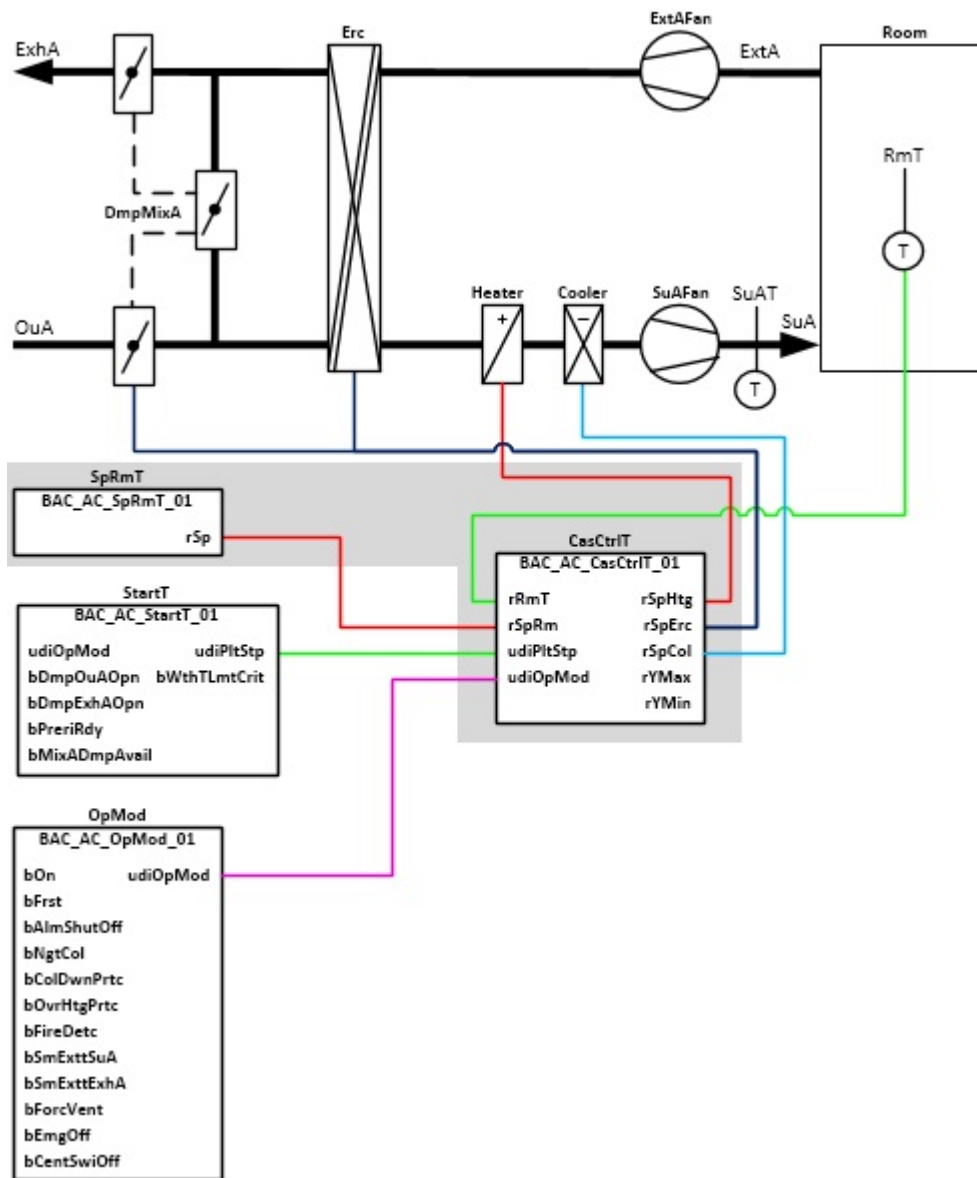
Anwendung

Das Template ist ein Kaskadenregler für die Zulufttemperatur bestehend aus einem Führungsregler zur Sollwertberechnung für das Heizen, Kühlen und der Energie-Rückgewinnung.

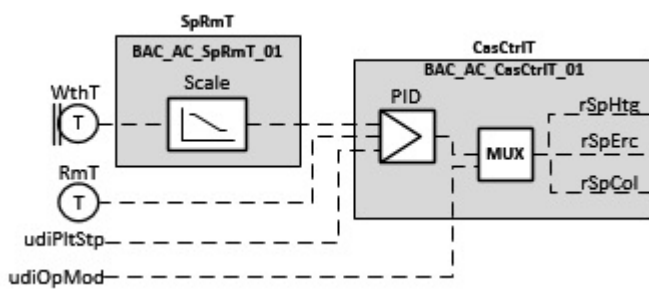
Schnittstelle



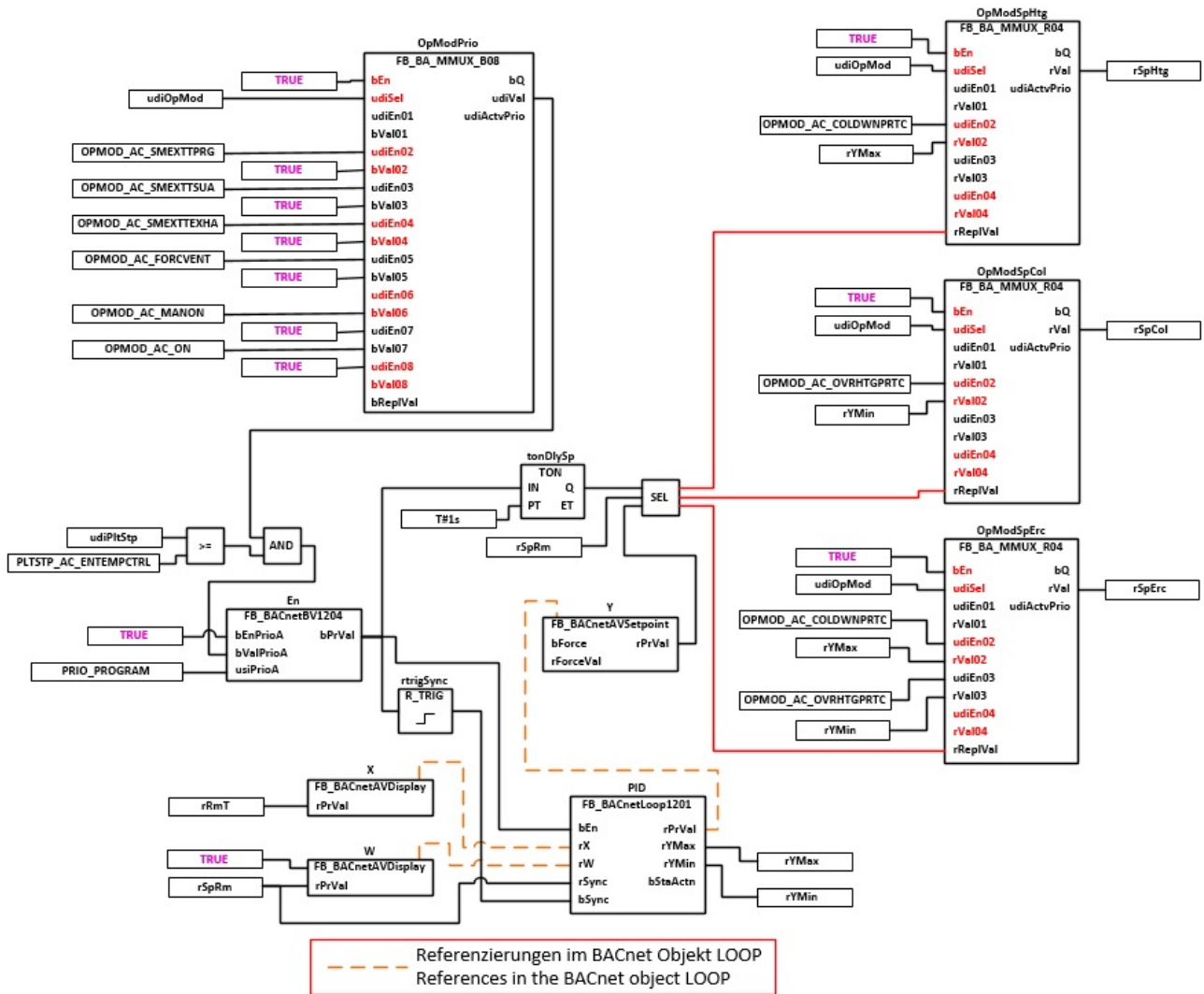
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```
rRmT      : REAL;
rSpRm     : REAL;
udiPltStp : UDINT;
udiOpMod  : UDINT;
```

rRmT: Eingangsvariable an der die Raumtemperatur angelegt wird. Die Raumtemperatur ist die Regelgröße des PID-Führungsreglers. Ist keine Raumtemperatur vorhanden, so kann die Ablufttemperatur einer Lüftungsanlage als Regelgröße genommen werden.

rSpRm: Eingangsvariable für den Sollwert Raumtemperatur, siehe Template [BAC AC SpRmT_01](#) [▶ 658].

udiPltStp: Schritt beim Anfahren der RLT-Anlage. Siehe auch [BAC AC Start_01](#) [▶ 538]

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod_01](#) [▶ 524]

VAR_OUTPUT

```
rSpHtg    : REAL;
rSpErc    : REAL;
rSpCol    : REAL;
rYMax     : REAL;
rYMin     : REAL;
```

rSpHtg: Errechneter Sollwert der Zulufttemperatur für das Heizen

rSpErc: Errechneter Sollwert der Zulufttemperatur für die Energierückgewinnung

rSpCol: Errechneter Sollwert der Zulufttemperatur für das Kühlen

rYMax: Oberer Wert der Regler-Ausgangsbegrenzung

rYMin: Unterer Wert der Regler-Ausgangsbegrenzung

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe																					
X	BACnetAVDisplay [► 69]	An das AV-Objekt ist die Eingangsvariable rRmT angeschlossen. Es ist referenziert auf den Istwert-Eingang des BACnet-Loop-Objekts PID .																					
W	BACnetAVDisplay [► 69]	An das AV-Objekt ist die Eingangsvariable rSpRm angeschlossen. Es ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts PID .																					
OpModPri o	FB_BA_MMUX_B08 [► 208]	Der Multiplexer definiert die Freigabebedingungen des PID-Führungsreglers in Abhängigkeit der Anlagenbetriebsart udiOpMod .																					
		<table border="1"> <thead> <tr> <th colspan="2">udiOpMod</th> <th>Freigabe Kaskadenregelung</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_SMEXTT PRG</td> <td>Entrauchung Programm</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_SMEXTT SUA</td> <td>Entrauchung Zuluft</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_SMEXTT EXHA</td> <td>Entrauchung Fortluft</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_FORCVENT</td> <td>Zwangsbelüftung</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_MANON</td> <td>Hand ein</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_ON</td> <td>Ein</td> <td>TRUE</td> </tr> </tbody> </table>	udiOpMod		Freigabe Kaskadenregelung	OPMOD_AC_SMEXTT PRG	Entrauchung Programm	TRUE	OPMOD_AC_SMEXTT SUA	Entrauchung Zuluft	TRUE	OPMOD_AC_SMEXTT EXHA	Entrauchung Fortluft	TRUE	OPMOD_AC_FORCVENT	Zwangsbelüftung	TRUE	OPMOD_AC_MANON	Hand ein	TRUE	OPMOD_AC_ON	Ein	TRUE
udiOpMod		Freigabe Kaskadenregelung																					
OPMOD_AC_SMEXTT PRG	Entrauchung Programm	TRUE																					
OPMOD_AC_SMEXTT SUA	Entrauchung Zuluft	TRUE																					
OPMOD_AC_SMEXTT EXHA	Entrauchung Fortluft	TRUE																					
OPMOD_AC_FORCVENT	Zwangsbelüftung	TRUE																					
OPMOD_AC_MANON	Hand ein	TRUE																					
OPMOD_AC_ON	Ein	TRUE																					
En	FB_BACnetBV1204 [► 94]	Das BV-Objekt dient zur Anzeige der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay																					
	>=, AND	Das Ergebnis dieses Netzwerkes ist die Freigabe des PID-Führungsreglers. Die Freigabe ist abhängig von der Anlagenbetriebsart und der Anlagenschritte udiPitStp beim Anfahren der Lüftungsanlage.																					
PID	FB_BACnetLoop1201 [► 99]	Führungsregler für eine Raumtemperaturregelung mittels Raum-Zuluft-Kaskade. Es liefert den einfachen Zulufttemperatur-Sollwert.																					
rtrigSync	R_TRIG	Mit Erhalt der Reglerfreigabe En löst rtrigSync die Synchronisation des Führungsreglers PID auf dessen Eingangswert IrSync aus.																					

Instanz	Typ	Aufgabe				
Y	FB BACnetAVSetpoint [▶ 70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts PID. Es zeigt den Zulufttemperatur-Sollwert.				
TLogY	FB BACnetTLog1201 [▶ 137]	Trendaufzeichnung des Zulufttemperatur-Sollwertes Y				
tonDlySp	TON SEL	Nach Erhalt der Reglerfreigabe En gibt dieses Netzwerkes verzögert den Zulufttemperatur-Sollwert an die nachfolgenden Multiplexer weiter.				
OpModSpHtg	FB BA MMUX R04 [▶ 208]	Der Multiplexer definiert den Sollwert Heizen rSpHtg in Abhängigkeit der Anlagenbetriebsart udiOpMod in bestimmten Fällen.				
		<table border="1"> <thead> <tr> <th>udiOpMod</th> <th>Sollwert</th> <th>udiOpMod</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_COLDWNPRT C</td> <td>Stützbetrieb, Auskühlschutz</td> <td>rYMax, Oberer Sollwert der Regler-Ausgangsbegrenzung</td> </tr> </tbody> </table>	udiOpMod	Sollwert	udiOpMod	OPMOD_AC_COLDWNPRT C
udiOpMod	Sollwert	udiOpMod				
OPMOD_AC_COLDWNPRT C	Stützbetrieb, Auskühlschutz	rYMax, Oberer Sollwert der Regler-Ausgangsbegrenzung				
OpModSpCol	FB BA MMUX R04 [▶ 208]	Der Multiplexer definiert den Sollwert Kühlen rSpCol in Abhängigkeit der Anlagenbetriebsart udiOpMod in bestimmten Fällen.				
		<table border="1"> <thead> <tr> <th>udiOpMod</th> <th>Sollwert</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_OVRHTG PRTC</td> <td>Überhitzungsschutz</td> </tr> </tbody> </table>	udiOpMod	Sollwert	OPMOD_AC_OVRHTG PRTC	Überhitzungsschutz
udiOpMod	Sollwert					
OPMOD_AC_OVRHTG PRTC	Überhitzungsschutz					
OpModSpErc	FB BA MMUX R04 [▶ 208]	Der Multiplexer definiert den Sollwert Energierückgewinnung rSpErc in Abhängigkeit der Anlagenbetriebsart udiOpMod in bestimmten Fällen.. see "Extracted nested table 121"				
		<table border="1"> <thead> <tr> <th>udiOpMod</th> <th>Sollwert</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_COLDWN PRTC</td> <td>Stützbetrieb, Auskühlschutz</td> </tr> <tr> <td>OPMOD_AC_OVRHTG PRTC</td> <td>Überhitzungsschutz</td> </tr> </tbody> </table>	udiOpMod	Sollwert	OPMOD_AC_COLDWN PRTC	Stützbetrieb, Auskühlschutz
udiOpMod	Sollwert					
OPMOD_AC_COLDWN PRTC	Stützbetrieb, Auskühlschutz					
OPMOD_AC_OVRHTG PRTC	Überhitzungsschutz					

Versionshistorie

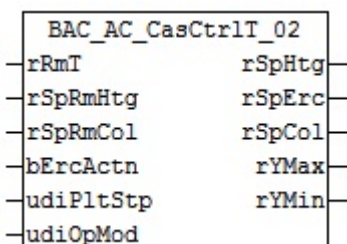
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.4 BAC_AC_CasCtrlT_02

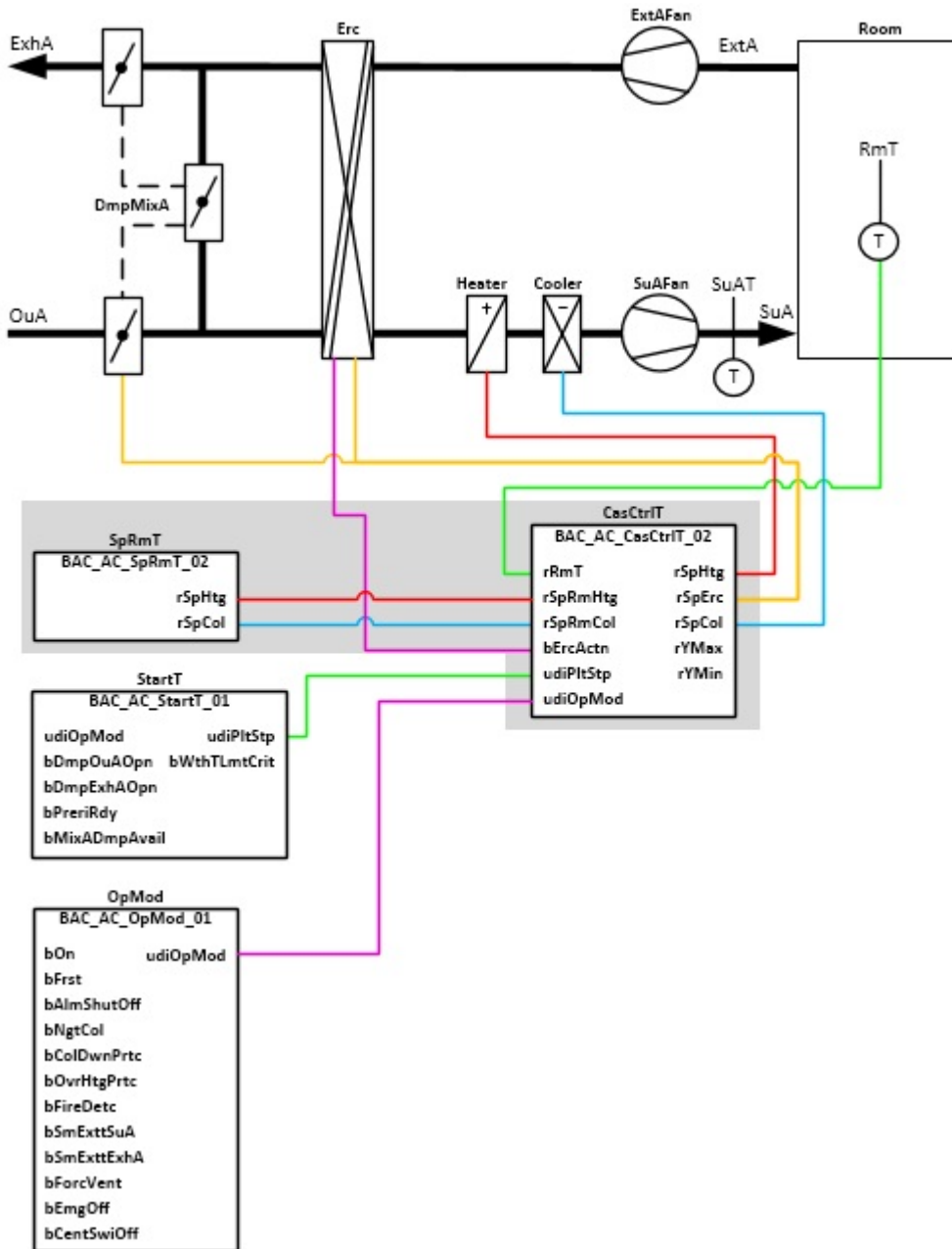
Anwendung

Das Template ist ein Kaskadenregler Zulufttemperatur bestehend aus zwei Führungsreglern zur Sollwertberechnung für das Heizen, Kühlen und der Energie-Rückgewinnung.

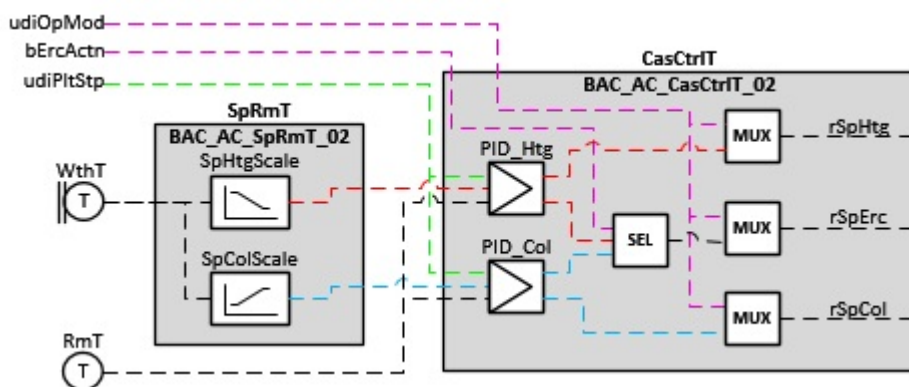
Schnittstelle



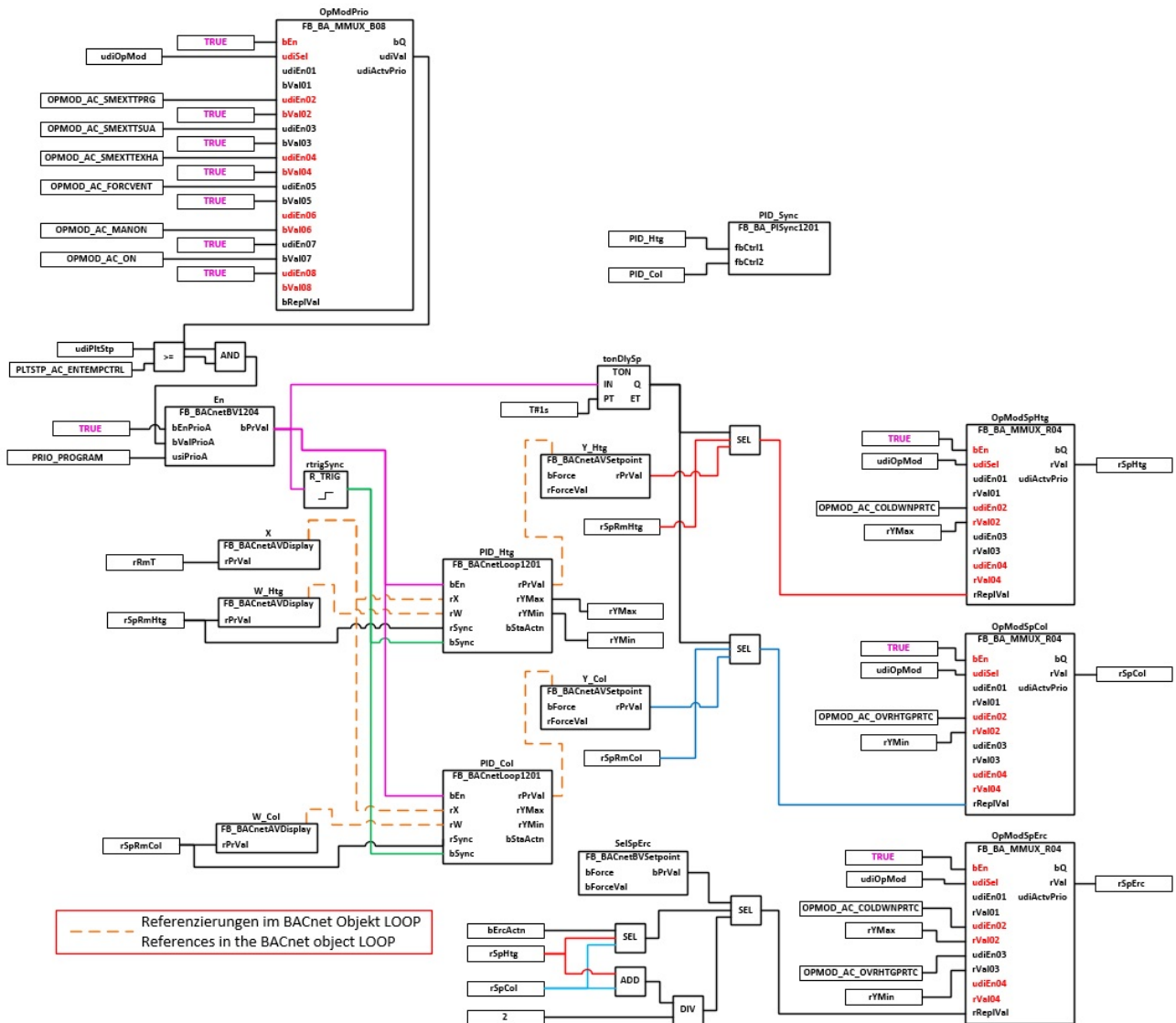
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_INPUT

```

rRmT          : REAL;
rSpRmHtg     : REAL;
rSpRmCol     : REAL;
udiPltStp    : UDINT;
udiOpMod     : UDINT;
    
```

rRmT: Eingangsvariable an der die Raumtemperatur angelegt wird. Die Raumtemperatur ist die Regelgröße der PID-Führungsregler. Ist keine Raumtemperatur vorhanden, so kann die Ablufttemperatur einer Lüftungsanlage als Regelgröße genommen werden.

rSpRmHtg: Eingangsvariable für den Sollwert Raumtemperatur Heizen, siehe Template [BAC AC SpRmT_02](#) [▶ 661].

rSpRmCol: Eingangsvariable für den Sollwert Raumtemperatur Kühlen, siehe Template [BAC AC SpRmT_02](#) [▶ 661].

bErcActn: Eingangsvariable an der der Wirksinn der Energierückgewinnung angelegt wird. In Abhängigkeit des Wirksinns wird der Sollwert für die Energierückgewinnung bestimmt. TRUE = Direct = Kühlen; FALSE = Reverse = Heizen

udiPltStp: Schritt beim Anfahren der RLT-Anlage. Siehe auch [BAC AC Start_01](#) [▶ 538]

udiOpMod: Anlagenbetriebsart. Siehe auch [BAC AC OpMod_01](#) [▶ 524]

VAR_OUTPUT

```
rSpHtg      : REAL;
rSpErc      : REAL;
rSpCol      : REAL;
rYMax       : REAL;
rYMin       : REAL;
```

rSpHtg: Errechneter Sollwert der Zulufttemperatur für das Heizen

rSpErc: Errechneter Sollwert der Zulufttemperatur für die Energierückgewinnung

rSpCol: Errechneter Sollwert der Zulufttemperatur für das Kühlen

rYMax: Oberer Wert der Regler-Ausgangsbegrenzung

rYMin: Unterer Wert der Regler-Ausgangsbegrenzung

VAR CONSTANT

```
PLT_NUM     : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [▶ 182]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [▶ 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	Aufgabe																		
X	FB_BACnetAVDisplay [▶ 69]	An das AV-Objekt ist die Eingangsvariable rRmT angeschlossen. Es ist referenziert auf die Istwert-Eingänge der BACnet-Loop-Objekte PID_Htg und PID_Col .																		
W_Htg	FB_BACnetAVDisplay [▶ 69]	An das AV-Objekt ist die Eingangsvariable rSpRmHtg angeschlossen. Es ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts PID_Htg .																		
W_Col	FB_BACnetAVDisplay [▶ 69]	An das AV-Objekt ist die Eingangsvariable rSpRmCol angeschlossen. Es ist referenziert auf den Sollwert-Eingang des BACnet-Loop-Objekts PID_Col .																		
OpModPri o	FB_BA_MMUX_B08 [▶ 208]	Der Multiplexer definiert die Freigabebedingungen der beiden PID-Führungsregler in Abhängigkeit der Anlagenbetriebsart udiOpMod .																		
		<table border="1"> <thead> <tr> <th>udiOpMod</th> <th>Freigabe</th> <th>Kaskadenregelung</th> </tr> </thead> <tbody> <tr> <td>OPMOD_AC_SMEXTT PRG</td> <td>Entrauchung Programm</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_SMEXTT SUA</td> <td>Entrauchung Zuluft</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_SMEXTT EXHA</td> <td>Entrauchung Fortluft</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_FORCVENT</td> <td>Zwangsbelüftung</td> <td>TRUE</td> </tr> <tr> <td>OPMOD_AC_MANON</td> <td>Hand ein</td> <td>TRUE</td> </tr> </tbody> </table>	udiOpMod	Freigabe	Kaskadenregelung	OPMOD_AC_SMEXTT PRG	Entrauchung Programm	TRUE	OPMOD_AC_SMEXTT SUA	Entrauchung Zuluft	TRUE	OPMOD_AC_SMEXTT EXHA	Entrauchung Fortluft	TRUE	OPMOD_AC_FORCVENT	Zwangsbelüftung	TRUE	OPMOD_AC_MANON	Hand ein	TRUE
udiOpMod	Freigabe	Kaskadenregelung																		
OPMOD_AC_SMEXTT PRG	Entrauchung Programm	TRUE																		
OPMOD_AC_SMEXTT SUA	Entrauchung Zuluft	TRUE																		
OPMOD_AC_SMEXTT EXHA	Entrauchung Fortluft	TRUE																		
OPMOD_AC_FORCVENT	Zwangsbelüftung	TRUE																		
OPMOD_AC_MANON	Hand ein	TRUE																		

Instanz	Typ	Aufgabe
		OPMOD_AC_ON Ein TRUE
En	FB BACnetBV1204 [▶ 94]	Das BV-Objekt dient zur Anzeige der Reglerfreigabe in der MBE oder in einen lokalen Bediendisplay.
	>=, AND	Das Ergebnis dieses Netzwerkes ist die Freigabe der beiden PID-Führungsregler. Die Freigabe ist abhängig von der Anlagenbetriebsart udiOpMod und der Anlagenschritte udiPitStp beim Anfahren der Lüftungsanlage.
PID_Htg	FB BACnetLoop1201 [▶ 99]	Führungsregler für eine Raumtemperaturregelung mittels Raum-Zuluft-Kaskade. Es liefert den Zulufttemperatur-Sollwert für das Heizen.
PID_Col	FB BACnetLoop1201 [▶ 99]	Führungsregler für eine Raumtemperaturregelung mittels Raum-Zuluft-Kaskade. Es liefert den Zulufttemperatur-Sollwert für das Kühlen.
rtrigSync	R_TRIG	Mit Erhalt der Reglerfreigabe En löst rtrigSync die Synchronisation der beiden Führungsregler PID_Htg / PID_Col auf deren Eingangswert IrSync aus.
Y_Htg	FB BACnetAVSetpoint [▶ 70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts PID_Htg . Es zeigt den Zulufttemperatur-Sollwert Heizen.
TLogY_Htg	FB BACnetTLog1201 [▶ 137]	Trendaufzeichnung des Zulufttemperatur-Sollwertes Heizen Y_Htg .
Y_Col	FB BACnetAVSetpoint [▶ 70]	Das AV-Objekt ist referenziert auf den Stellgrößenausgang des BACnet-Loop-Objekts PID_Col . Es zeigt den Zulufttemperatur-Sollwert Kühlen.
TLogY_Col	FB BACnetTLog1201 [▶ 137]	Trendaufzeichnung des Zulufttemperatur-Sollwertes Kühlen Y_Col .
PID_Sync	FB BA PISync1201 [▶ 164]	Der Funktionsbaustein führt eine Parameter-Synchronisation der beiden Führungsregler PID_HTG / PID_COL durch.
tonDlySp	TON SEL	Nach Erhalt der Reglerfreigabe En gibt dieses Netzwerkes verzögert die Zulufttemperatur-Sollwerte an die nachfolgenden Multiplexer weiter.
OpModSpHtg	FB BA MMUX_R04 [▶ 208]	Der Multiplexer definiert den Sollwert Heizen rSpHtg in Abhängigkeit der Anlagenbetriebsart udiOpMod in bestimmten Fällen.
	udiOpMod	Sollwert
	OPMOD_AC_COLDWNPRTC	Stützbetrieb,Auskühlschutz rYMax, Oberer Sollwert der Regler-Ausgangsbegrenzung
OpModSpCol	FB BA MMUX_R04 [▶ 208]	Der Multiplexer definiert den Sollwert Kühlen rSpCol in Abhängigkeit der Anlagenbetriebsart udiOpMod in bestimmten Fällen.
	udiOpMod	Sollwert
	OPMOD_AC_OVRHTGPRTC	Überhitzungsschutz rYMin, Unterer Sollwert der Regler-Ausgangsbegrenzung
SelSpErc	FB BACnetBV1204 [▶ 94]	Auswahl Strategie Sollwert Energierückgewinnung. TRUE : Mittelwert aus rSpHtg und rSpCol ; FALSE : Wirksinnabhängig, definiert durch den Eingang bErcActn
	SEL, ADD, DIV, SEL	Das Ergebnis dieses Netzwerkes liefert 2 Sollwerte für die Energierückgewinnung. Welcher genommen wird, ist abhängig von der Strategie Sollwert Energierückgewinnung, siehe SelSpErc
OpModSpErc	FB BA MMUX_R04 [▶ 208]	Der Multiplexer definiert den Sollwert Energierückgewinnung rSpErc in Abhängigkeit der Anlagenbetriebsart udiOpMod in bestimmten Fällen..
	udiOpMod	Sollwert
	OPMOD_AC_COLDWNPRTC	Stützbetrieb,Auskühlschutz rYMax, Oberer Sollwert der Regler-Ausgangsbegrenzung

Instanz	Typ	Aufgabe		
		OPMOD_AC_OVRHTGPRTC	Überhitzungsschutz	rYMin, Unterer Sollwert der Regler- Ausgangsbe- grenzung

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.5 BAC_AC_SpRmT_01

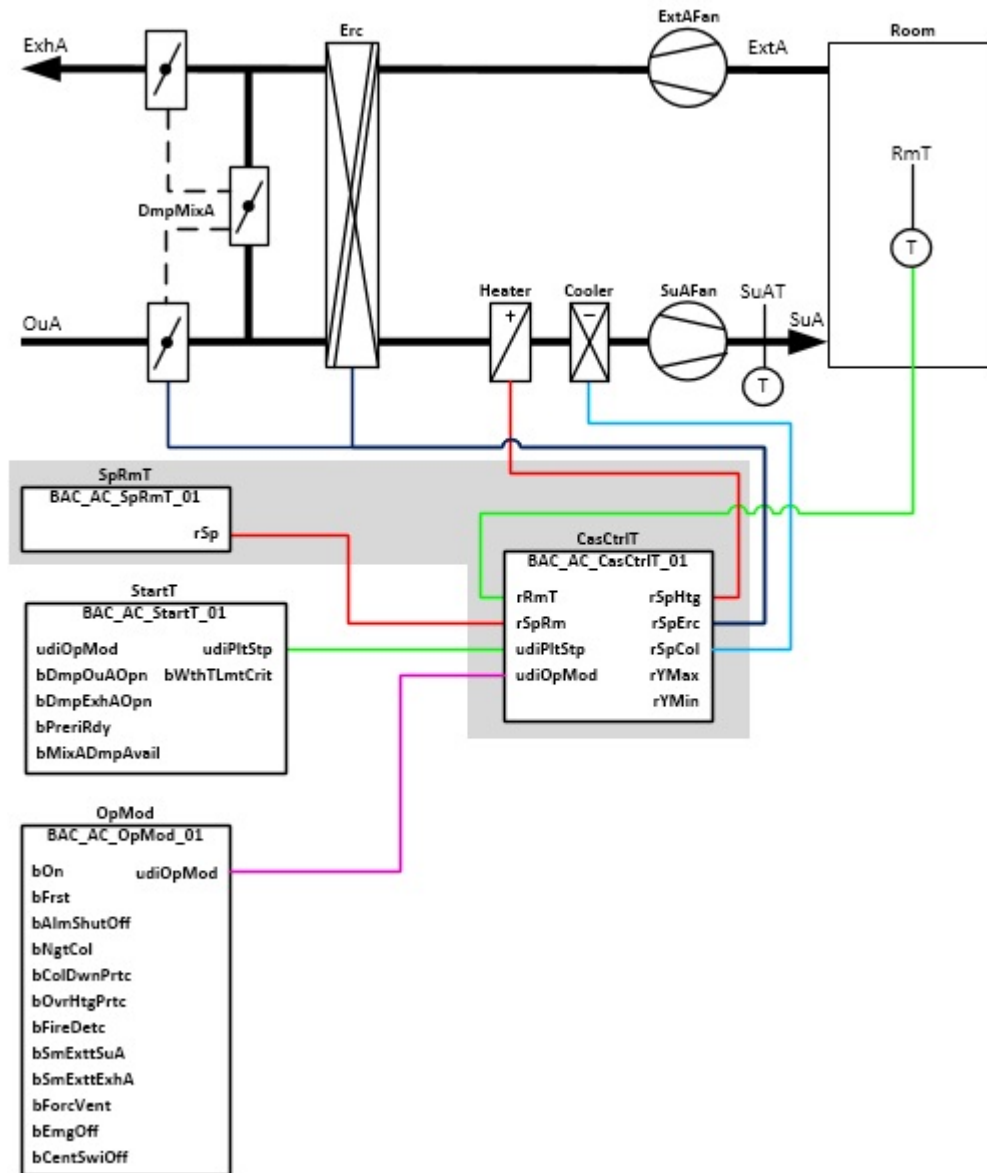
Anwendung

Das Template ist ein Sollwertprogramm für eine Abluft/Zuluftkaskade mit nur einem Raumtemperatursollwert inklusive Sommer- und Winterkompensation.

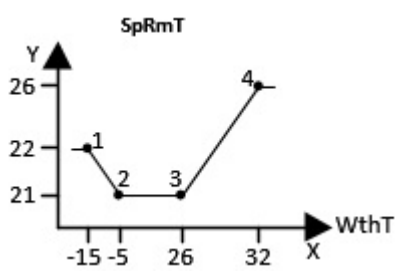
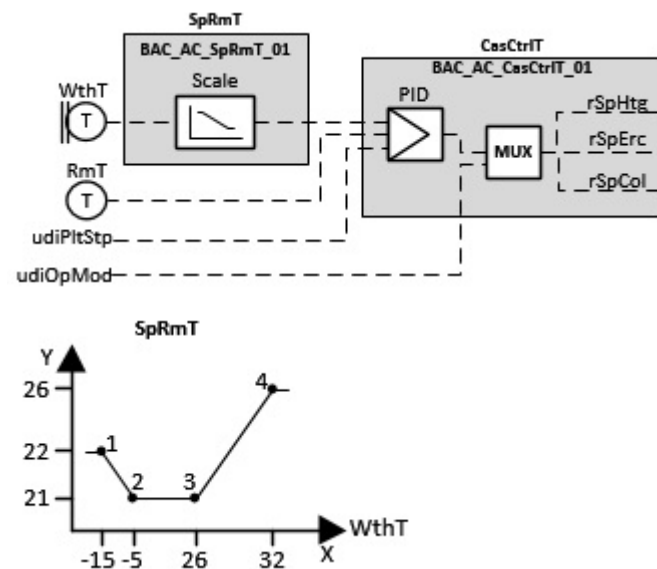
Schnittstelle

```
BAC_AC_SpRmT_01
rSp
```

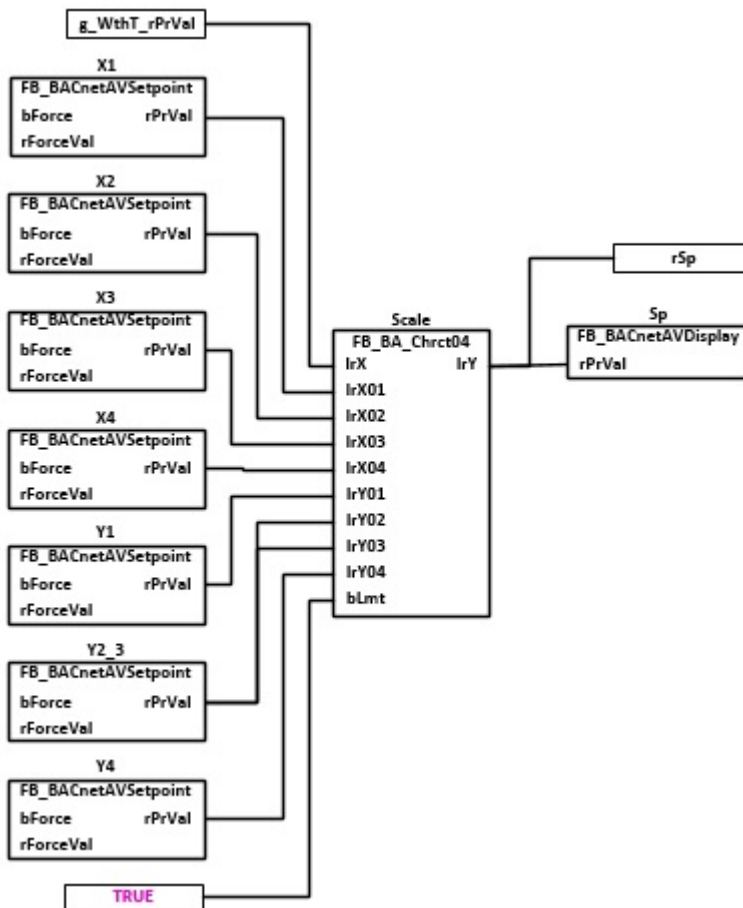
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_OUTPUT

```
rSp : REAL;
```

rSp: Errechneter Sollwert der Raumtemperatur

Programmbeschreibung

Instanz	Typ	Aufgabe
X1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X1
X2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X2
X3	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X3
X4	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X4
Y1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y1
Y2_3	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für die Stützpunkte Y2/Y3
Y4	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y4
Scale	FB_BA_Chrct04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie für die aktuelle Raumtemperatur in Abhängigkeit der Aussentemperatur.
Sp	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten, einfachen Raumtemperatur-Sollwertes. Dieser wird an dem Ausgang rSp ausgegeben.

Versionshistorie

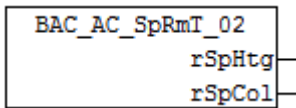
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.6 BAC_AC_SpRmT_02

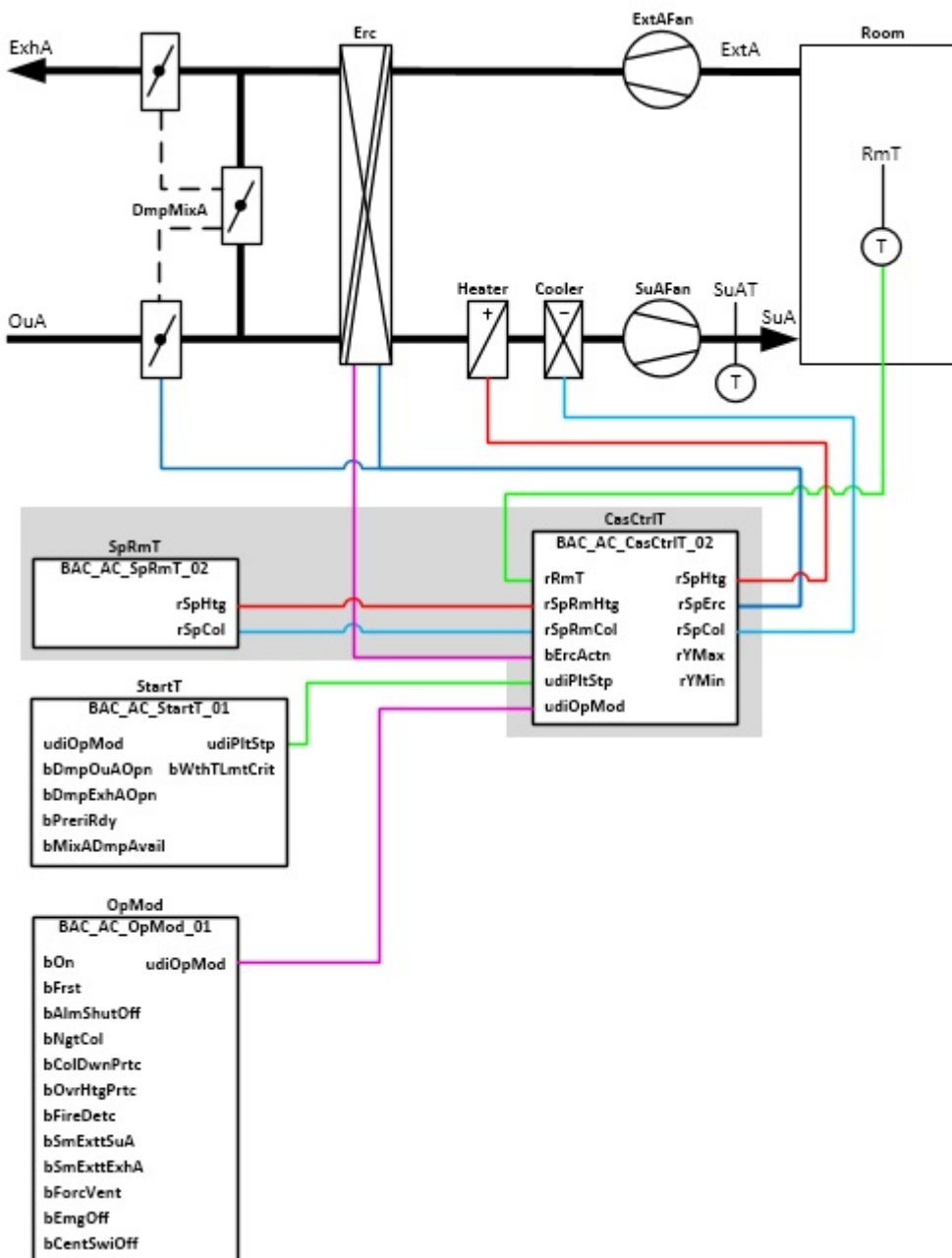
Anwendung

Sollwertprogramm für eine Abluft/Zuluftkaskade mit jeweils einem Raumtemperatursollwert für den Heiz- und Kühlbetrieb inklusive Sommer und Winterkompensation

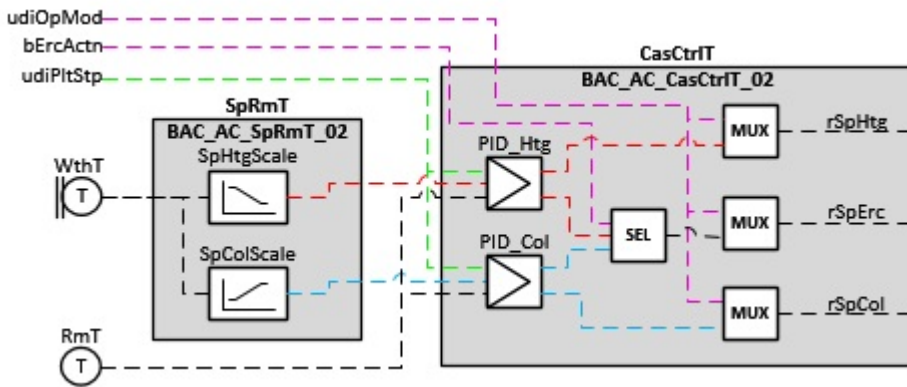
Schnittstelle



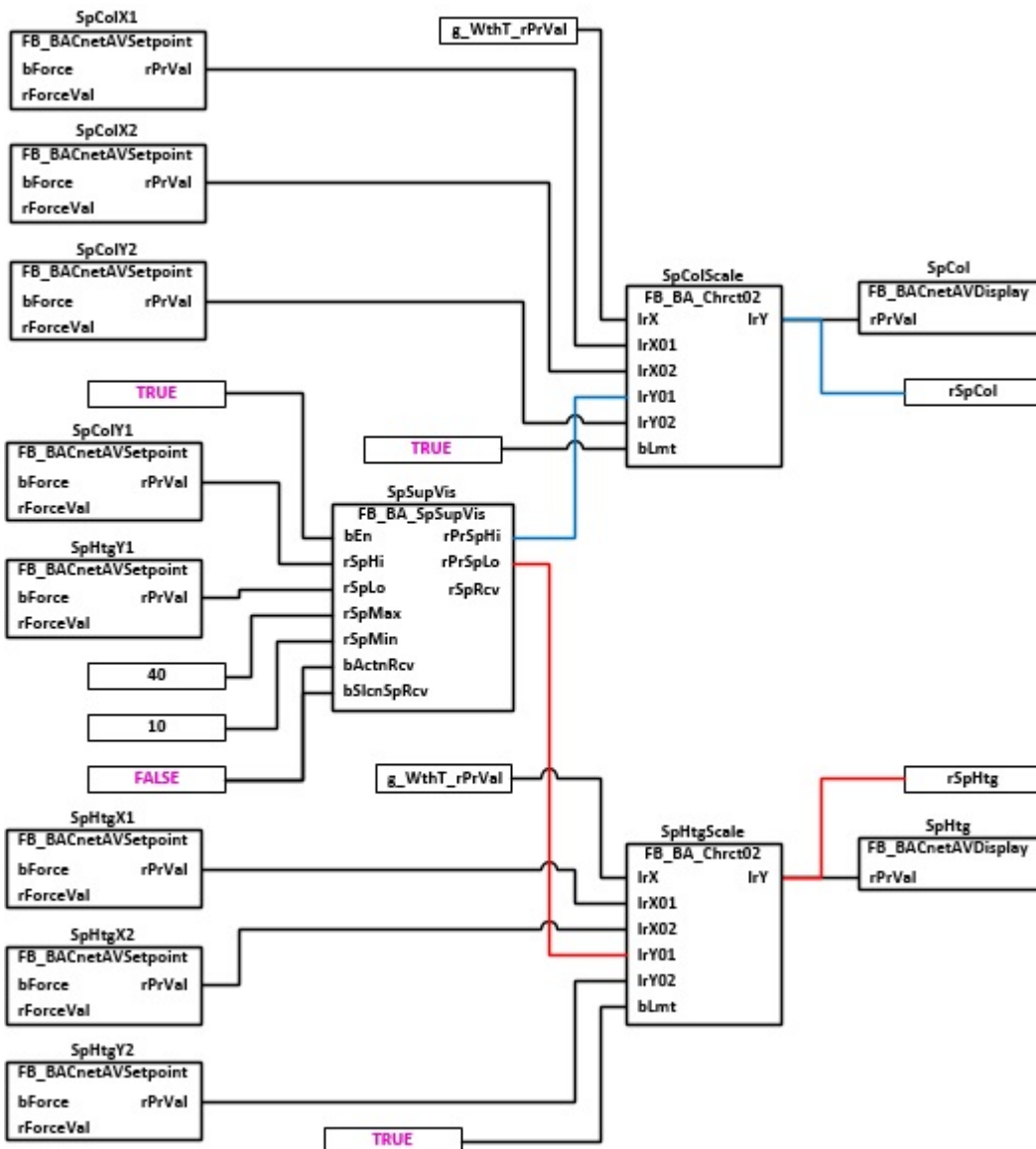
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_OUTPUT

rSp : REAL;

rSpHtg: Errechneter Sollwert der Raumtemperatur für das Heizen

rPrCol: Errechneter Sollwert der Raumtemperatur für das Kühlen

Programmbeschreibung

Instanz	Typ	Aufgabe
SpColX1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X1 der Sommerkompensation für den Sollwert Kühlen.
SpColX2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X2 der Sommerkompensation für den Sollwert Kühlen.
SpColY1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y1 der Sommerkompensation für den Sollwert Kühlen. SpColY1 ist der Basis-Sollwert für das Kühlen.
SpColY2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y2 der Sommerkompensation für den Sollwert Kühlen.
SpColScale	FB_BA_Chrc02 [▶ 173]	Der Funktionsbaustein ermittelt den Sommerkompensationswert für den Sollwert Kühlen.
SpHtgX1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X1 der Winterkompensation für den Sollwert Heizen.
SpHtgX2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X2 der Winterkompensation für den Sollwert Heizen.
SpHtgY1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y1 der Winterkompensation für den Sollwert Heizen. SpHtgY1 ist der Basis-Sollwert für das Heizen.
SpHtgY2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y2 der Winterkompensation für den Sollwert Heizen.
SpHtgScale	FB_BA_Chrc02 [▶ 173]	Der Funktionsbaustein ermittelt den Winterkompensationswert für den Sollwert Heizen.
SpSuVis	FB_BA_SpSupVis [▶ 246]	Der Funktionsbaustein dient zur Kontrolle der beiden Stützpunkte SpHtgY1 und SpColY1.
SpHtg	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumtemperatur-Sollwertes Heizen
SpCol	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumtemperatur-Sollwertes Kühlen

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.7 BAC_AC_SpRmTH_01

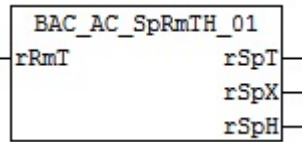
Anwendung

Das Template ist **BAC_AC_SpRmTH_01** ist ein einfaches Sollwertprogramm für eine raumlufttechnische Anlage mit einer Raum- bzw. Abluft-Kaskadenregelung für die Temperatur und Feuchte. Es gibt einen Raumtemperaturbasissollwert **SpT_Y2_3**. Eine energieneutrale Zone zwischen einen unteren (Heizbetrieb) und einem oberen Sollwert (Kühlbetrieb) existiert nicht. Eine außentemperaturabhängige Verschiebung des Raumtemperaturbasissollwertes ist durch eine Kurvenfunktion realisiert (Sommer-Winterkompensation).

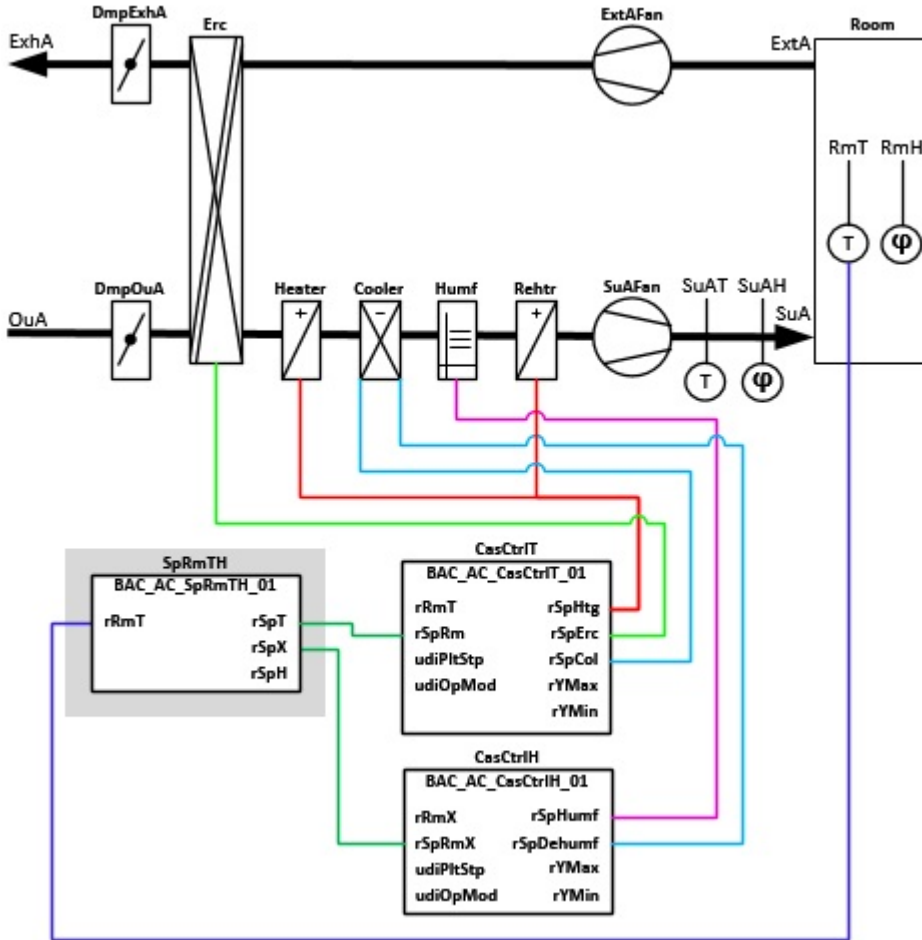
Die Eingabe des relativen Raumfeuchtesollwertes (%) erfolgt mit dem BACnet-AV-Objekt **SpH**. Eine energieneutrale Zone zwischen einem unteren (Befeuchtung) und einem oberen Sollwert (Entfeuchtung) der Feuchteregeung existiert nicht.

Der Sollwert für die relative Raumfeuchte wird innerhalb des Templates mittels der aktuellen Raumtemperatur in einen Sollwert für die absolute Raumfeuchte (g/Kg) umgerechnet **SpX**.

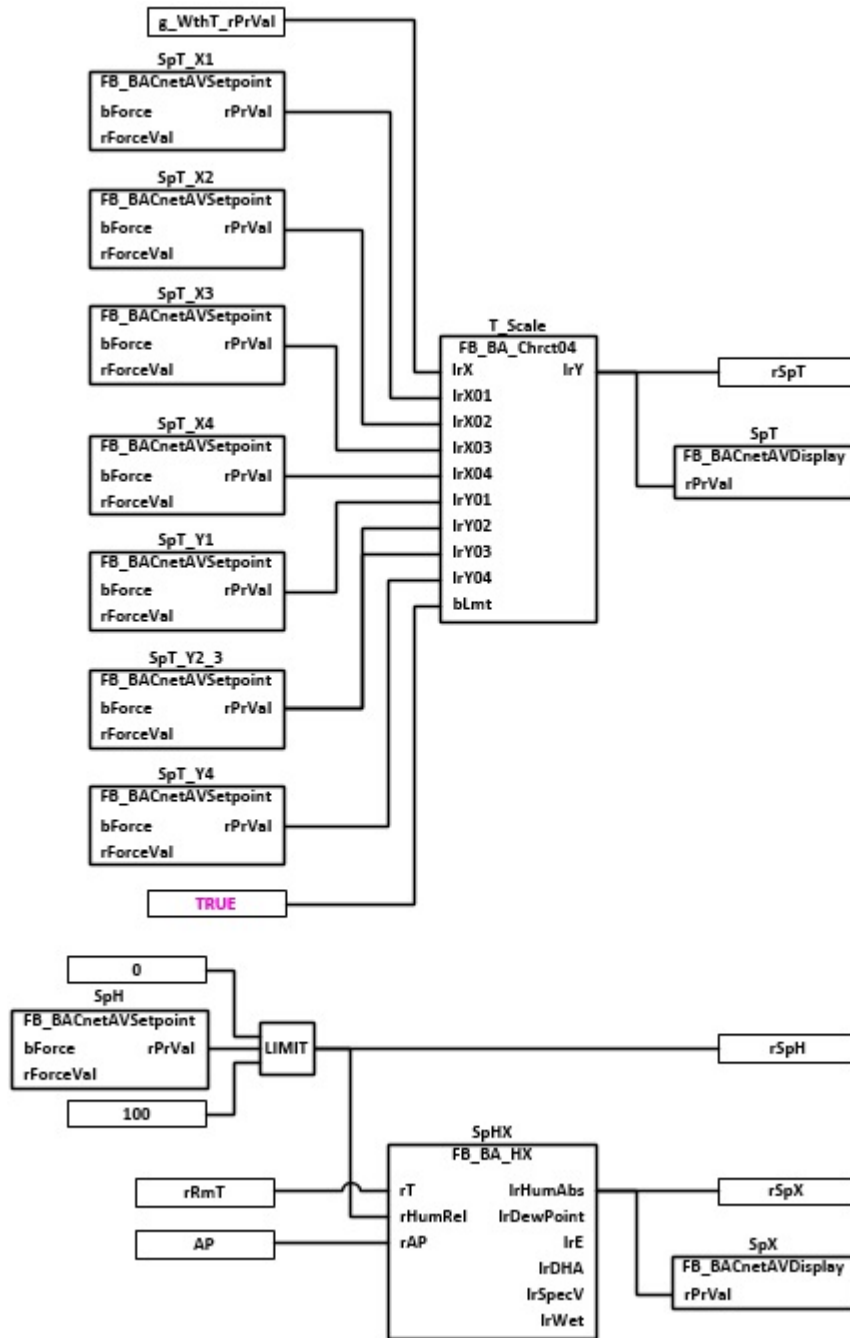
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_Input

rRmT : REAL;

rRmT: Messwert der Raumtemperatur

VAR_OUTPUT

rSpT : REAL;

rSpX : REAL;

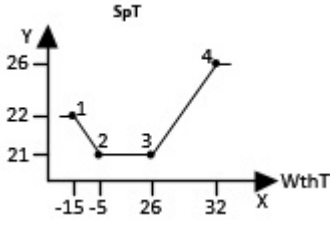
rSpH : Real;

rSpT: Errechneter Sollwert der Raumtemperatur

rSpX: Errechneter Sollwert der Raumfeuchte absolut

rSpH: Sollwert Raumfeuchte relativ

Programmbeschreibung

Instanz	Typ	Aufgabe
SpT_X1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X1
SpT_X2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X2
SpT_X3	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X3
SpT_X4	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X4
SpT_Y1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y1
SpT_Y2_3	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für die Stützpunkte Y2/Y3. Der Wert ist der Basissollwert
SpT_Y4	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y4
T_Scale	FB_BA Chrct04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie für die aktuelle Raumtemperatur in Abhängigkeit der Außentemperatur. 
SpT	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumtemperatur-Sollwertes.
SpH	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Raumfeuchtesollwertes relative (%).
SpHX	FB_BA HX [▶ 230]	Mit diesem Funktionsbaustein wird der Sollwert der absoluten Raumfeuchte berechnet. Für die Berechnung der Größe wird die Raumtemperatur, der Sollwert der relativen Feuchte SpH und der barometrische Luftdruck benötigt.
SpX	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumfeuchte-Sollwertes absolut (g/Kg).

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.8 BAC_AC_SpRmTH_02

Anwendung

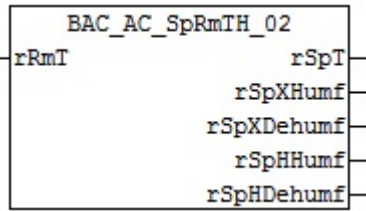
Das Template ist **BAC_AC_SpRmTH_02** ist ein einfaches Sollwertprogramm für eine raumluftechnische Anlage mit einer Raum- bzw. Abluft-Kaskadenregelung für die Temperatur und Feuchte.

Es gibt einen Raumtemperaturbasissollwert **SpT_Y2_3**. Eine energieneutrale Zone zwischen einem unteren (Heizbetrieb) und einem oberen Sollwert (Kühlbetrieb) existiert nicht. Eine außentemperaturabhängige Verschiebung des Raumtemperaturbasissollwertes ist durch eine Kurvenfunktion realisiert (Sommer-Winterkompensation).

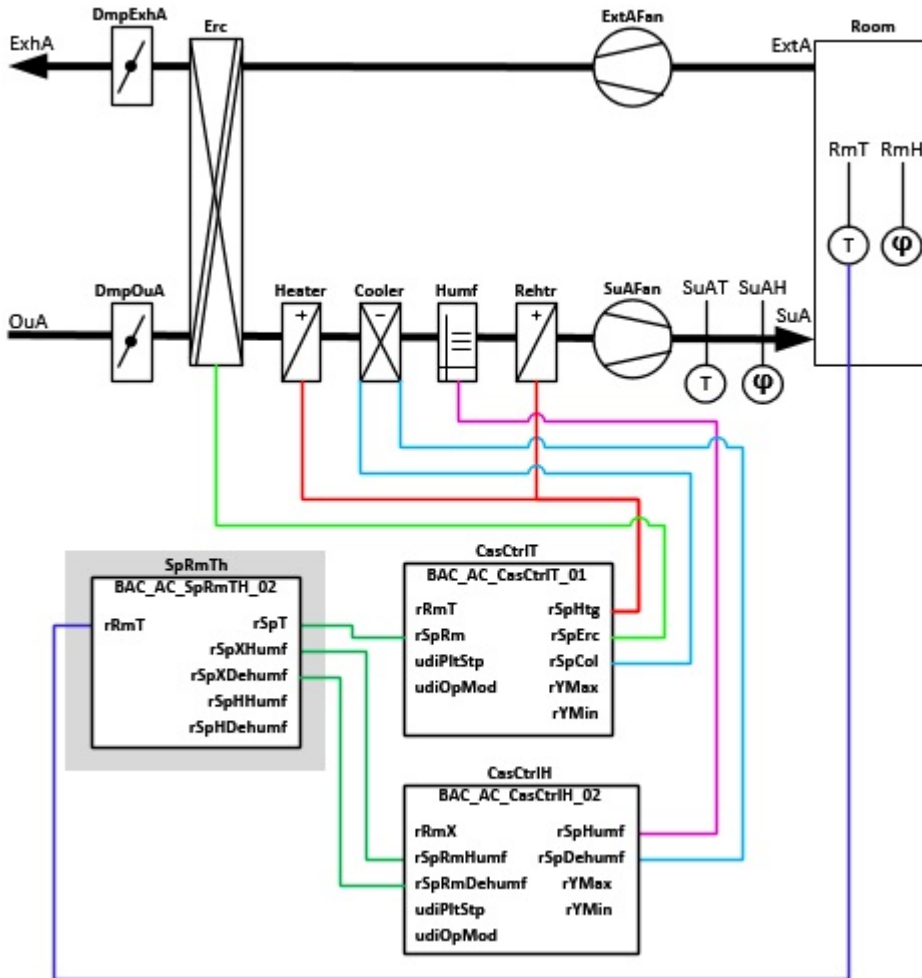
Die Eingabe der relativen Raumfeuchtesollwerte (%) erfolgt mit den BACnet-AV-Objekten **SpHHumf (Befeuchten)** und **SpHDehumf (Entfeuchten)**. Es gibt eine energieneutrale Zone zwischen dem unteren (**Befeuchten**) und oberen Sollwert (**Entfeuchten**), welche durch die Funktion **SpHSupVis** überwacht wird.

Die beiden Sollwerte für die relative Raumfeuchte werden innerhalb des Templates mittels der aktuellen Raumtemperatur in zwei Sollwerte für die absolute Raumfeuchte (g/Kg) umgerechnet, **SpXHumf (Befeuchten)** und **SpXDehumf (Entfeuchten)**.

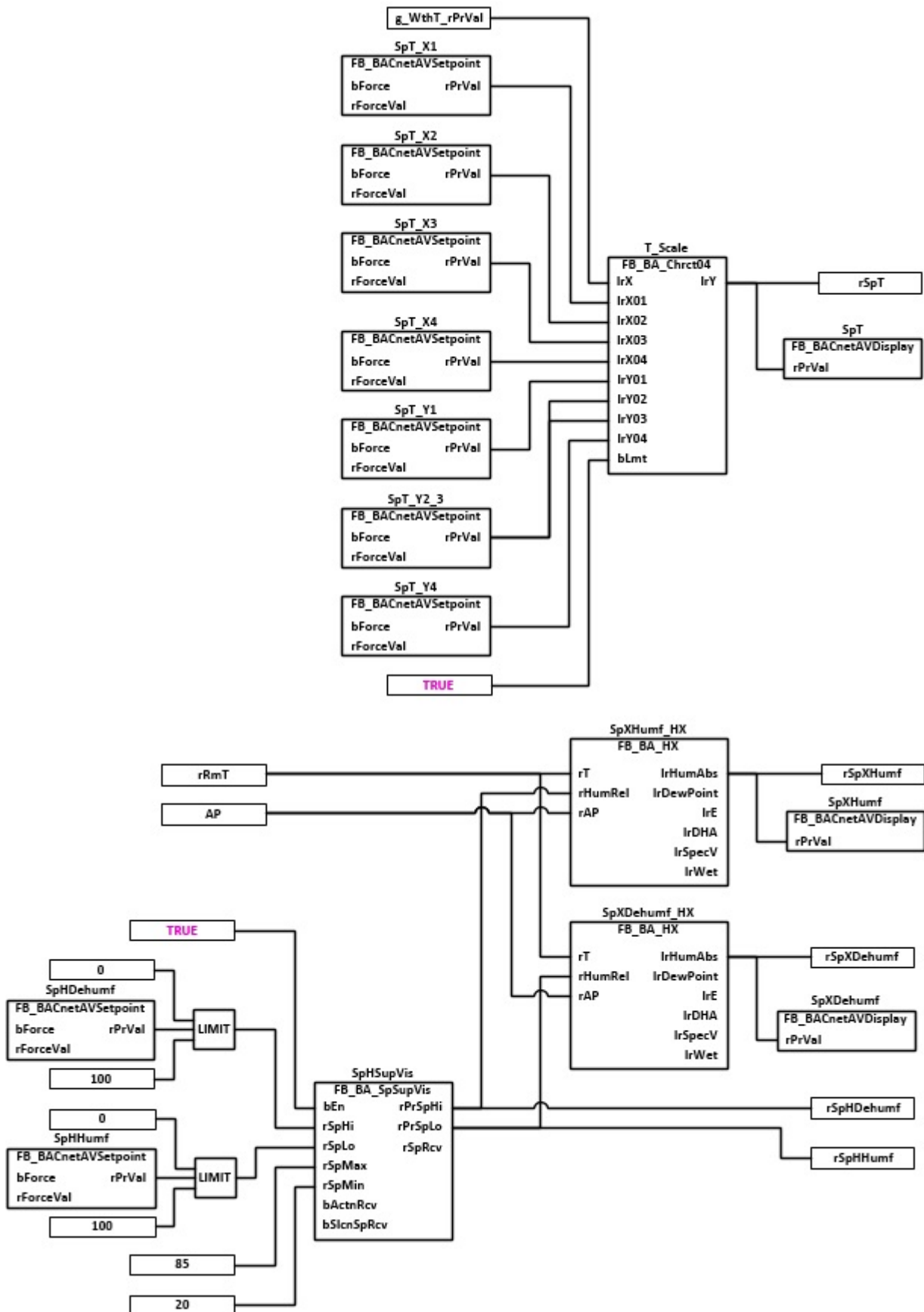
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_Input

`rRmT` : REAL;

rRmT: Messwert der Raumtemperatur

VAR_OUTPUT

```
rSpT      : REAL;
rSpXHumf  : REAL;
rSpXDehumf : REAL;
rSpHHumf  : REAL;
rSpHDehumf : REAL;
```

rSpT: Errechneter Sollwert der Raumtemperatur

rSpXHumf: Sollwert Absolute Feuchte; Befeuchten

rSpXDehumf: Sollwert Absolute Feuchte; Entfeuchten

rSpHHumf: Sollwert Relative Feuchte; Befeuchten

rSpHDehumf: Sollwert Relative Feuchte; Entfeuchten

Programmbeschreibung

Instanz	Typ	Aufgabe
SpT_X1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X1
SpT_X2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X2
SpT_X3	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X3
SpT_X4	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X4
SpT_Y1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y1
SpT_Y2_3	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für die Stützpunkte Y2/Y3. Der Wert ist der Basissollwert
SpT_Y4	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y4
T_Scale	FB_BA_Chrc04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie für die aktuelle Raumtemperatur in Abhängigkeit der Außentemperatur.
SpT	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumtemperatur-Sollwertes.
SpHHumf	FB_BACnetAVSetpoint [▶ 70]	Eingabe des relativen Raumfeuchtesollwertes Befeuchten(%).
SpHDehumf	FB_BACnetAVSetpoint [▶ 70]	Eingabe des relativen Raumfeuchtesollwertes Entfeuchten(%).
SpHSupVis	FB_BA_SpSupVis [▶ 246]	Der Funktionsbaustein dient zur Kontrolle der energieneutralen Zone zwischen den beiden Sollwerten SpHDehumf und SpHHumf
SpXHumf_HX	FB_BA_HX [▶ 230]	Mit diesem Funktionsbaustein wird der Sollwert der absoluten Raumfeuchte, Befeuchten, berechnet. Für die Berechnung der Größe wird die Raumtemperatur, der Sollwert der relativen Feuchte SpHHumf und der barometrische Luftdruck benötigt.
SpXHumf	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten absoluten Raumfeuchte-Sollwertes Befeuchten (g/Kg).

Instanz	Typ	Aufgabe
SpXDehumf_HX	FB BA HX [▶ 230]	Mit diesem Funktionsbaustein wird der Sollwert der absoluten Raumfeuchte, Entfeuchten, berechnet. Für die Berechnung der Größe wird die Raumtemperatur, der Sollwert der relativen Feuchte SpHDehumf und der barometrische Luftdruck benötigt.
SpXDehumf	FB BACnetAVDisplay [▶ 69]	Ausgabe des errechneten absoluten Raumfeuchte-Sollwertes Entfeuchten (g/Kg).

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

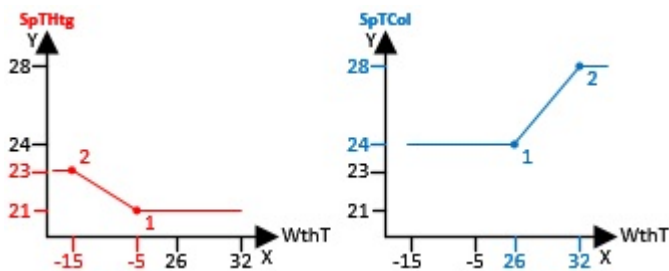
9.80.9 BAC_AC_SpRmTH_03

Anwendung

Das Template ist **BAC_AC_SpRmTH_03** ist ein Sollwertprogramm für eine raumluftechnische Anlage mit einer Raum- bzw. Abluft-Kaskadenregelung für die Temperatur und Feuchte.

Es gibt zwei Raumtemperaturbasissollwerte (**SpTHtgY1**, **SpTColY1**) mit einer energieneutralen Zone zwischen dem unteren (Heizbetrieb) und oberen Sollwert (Kühlbetrieb). Die energieneutralen Zone wird durch die Funktion **SpHSupVis** überwacht.

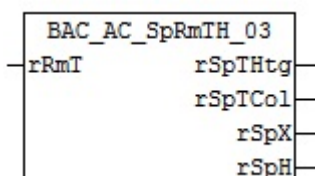
Eine außentemperaturabhängige Verschiebung der Raumtemperaturbasissollwerte ist durch zwei Kurvenfunktion realisiert (Sommer- Winterkompensation).



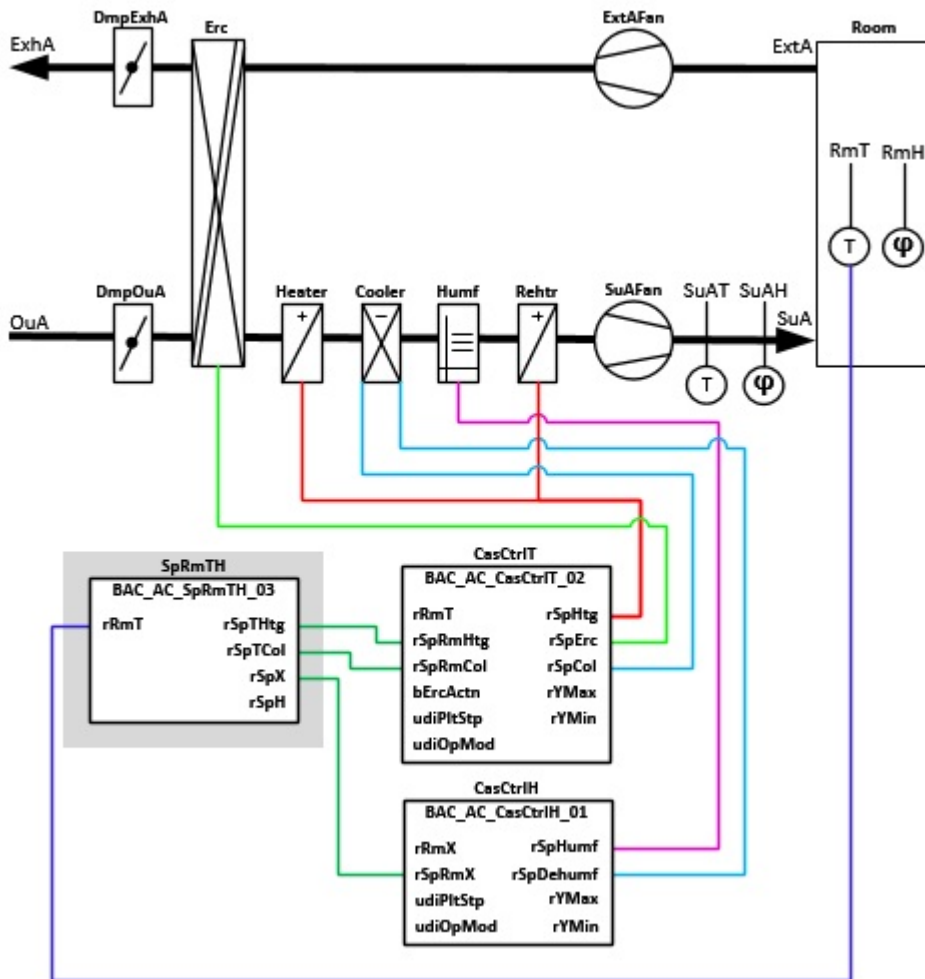
Die Eingabe des relativen Raumfeuchtesollwertes (%) erfolgt mit dem BACnet-AV-Objekt **SpH**. Eine energieneutrale Zone zwischen einem unteren Sollwert (Befeuchtung) und einem oberen Sollwert (Entfeuchtung) der Feuchteregeung existiert nicht.

Der Sollwert für die relative Raumfeuchte **SpH** wird innerhalb des Templates mittels der aktuellen Raumtemperatur in einen Sollwert für die absolute Raumfeuchte (g/Kg) umgerechnet **SpX**.

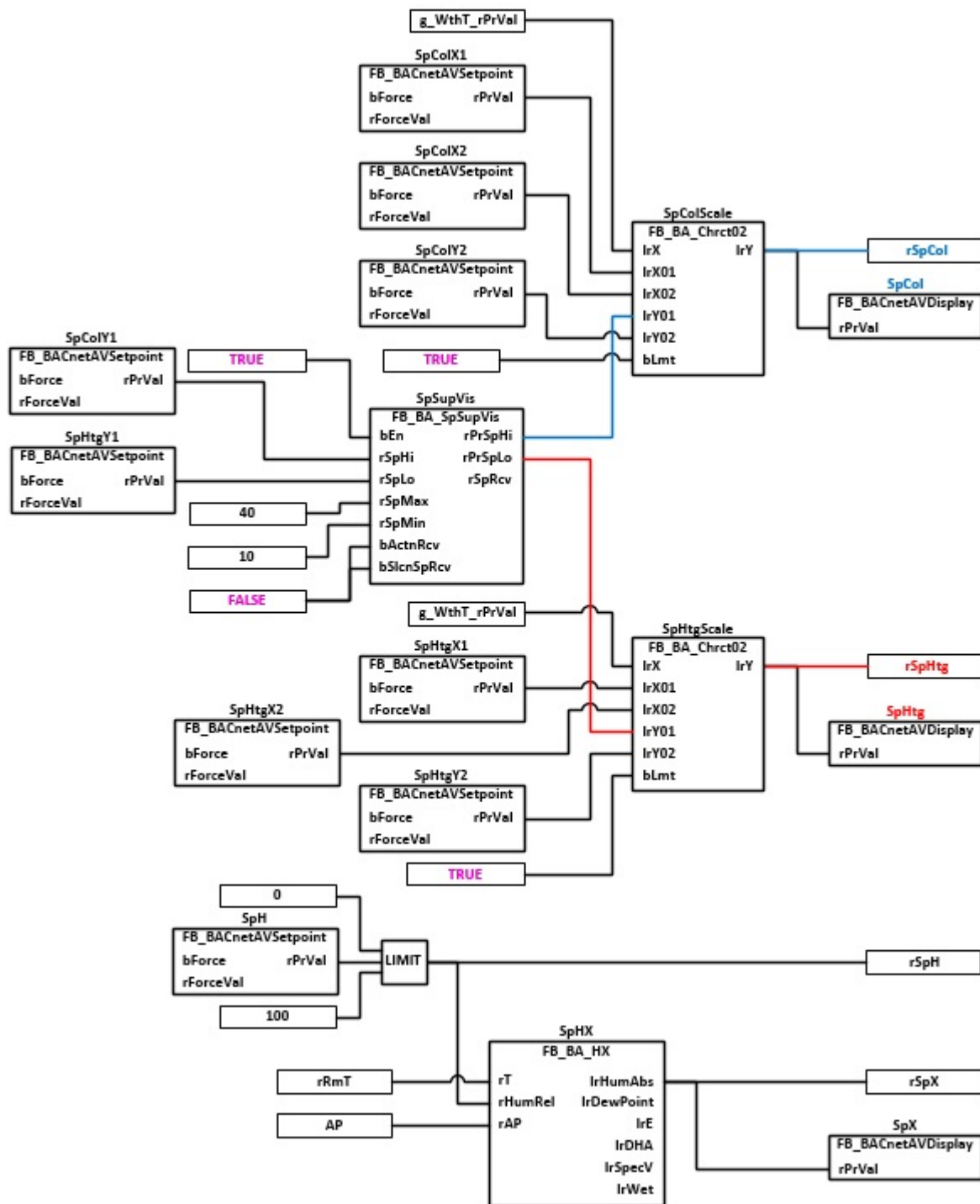
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_Input

```
rRmT : REAL;
```

rRmT: Messwert der Raumtemperatur

VAR_OUTPUT

```
rSpTHtg : REAL;
rSpTCol : REAL;
rSpX : REAL;
rSpH : Real;
```

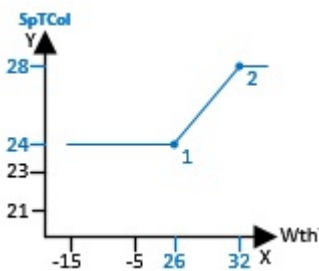
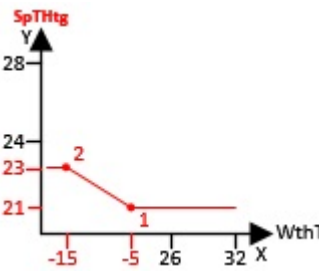
rSpTHtg: Errechneter Sollwert Heizen der Raumtemperatur

rSpTCol: Errechneter Sollwert Kühlen der Raumtemperatur

rSpX: Errechneter Sollwert der Raumfeuchte absolut

rSpH: Sollwert Raumfeuchte relativ

Programmbeschreibung

Instanz	Typ	Aufgabe
SpTColX1	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTColX1 Kühlen
SpTColX2	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTColX2 Kühlen
SpTColY1	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTColY1 Kühlen. Der Wert ist der Basissollwert für das Kühlen
SpTColY2	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTColY2 Kühlen
SpTColScale	FB BA Chrct04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie Kühlen für die aktuelle Raumtemperatur in Abhängigkeit der Aussentemperatur. 
SpTCol	FB BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumtemperatur-Sollwertes Kühlen
SpTHtgX1	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTHtgX1 Heizen
SpTHtgX2	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTHtgX2 Heizen
SpTHtgY1	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTHtgY1 Heizen. Der Wert ist der Basissollwert für das Heizen
SpTHtgY2	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTHtgY2 Heizen
SpTHtgScale	FB BA Chrct04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie Heizen für die aktuelle Raumtemperatur in Abhängigkeit der Aussentemperatur. 
SpTHtg	FB BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumtemperatur-Sollwertes Heizen

Instanz	Typ	Aufgabe
SpTSupVis	FB BA_SpSupVis [▶ 246]	Der Funktionsbaustein dient zur Kontrolle der energieneutralen Zone der beiden Sollwerte SpTCol und SpTHtg
SpH	FB BACnetAVSetpoint t [▶ 70]	Eingabe des Raumfeuchtesollwertes relativ (%).
SpHX	FB BA_HX [▶ 230]	Mit diesem Funktionsbaustein wird der Sollwert der absoluten Raumfeuchte berechnet. Für die Berechnung der Größe wird die Raumtemperatur, der Sollwert der relativen Feuchte SpH und der barometrische Luftdruck benötigt.
SpX	FB BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumfeuchte-Sollwertes absolut (g/Kg).

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

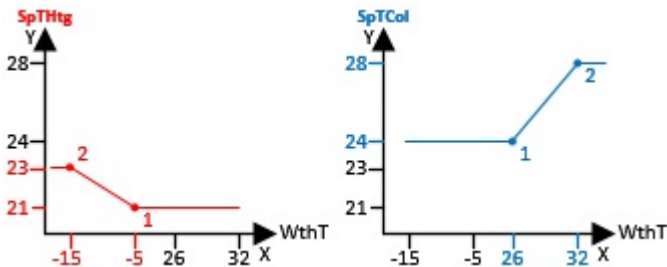
9.80.10 BAC_AC_SpRmTH_04

Anwendung

Das Template ist **BAC_AC_SpRmTH_04** ist ein Sollwertprogramm für eine raumluftechnische Anlage mit einer Raum- bzw. Abluft-Kaskadenregelung für die Temperatur und Feuchte.

Es gibt zwei Raumtemperaturbasissollwerte (**SpTHtgY1**, **SpTColY1**) mit einer energieneutralen Zone zwischen dem unteren (Heizbetrieb) und oberen Sollwert (Kühlbetrieb). Die energieneutralen Zone wird durch die Funktion **SpHSupVis** überwacht.

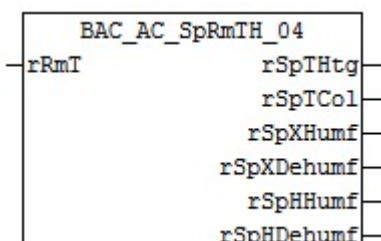
Eine außentemperaturabhängige Verschiebung der Raumtemperaturbasissollwerte ist durch zwei Kurvenfunktion realisiert (Sommer- Winterkompensation).



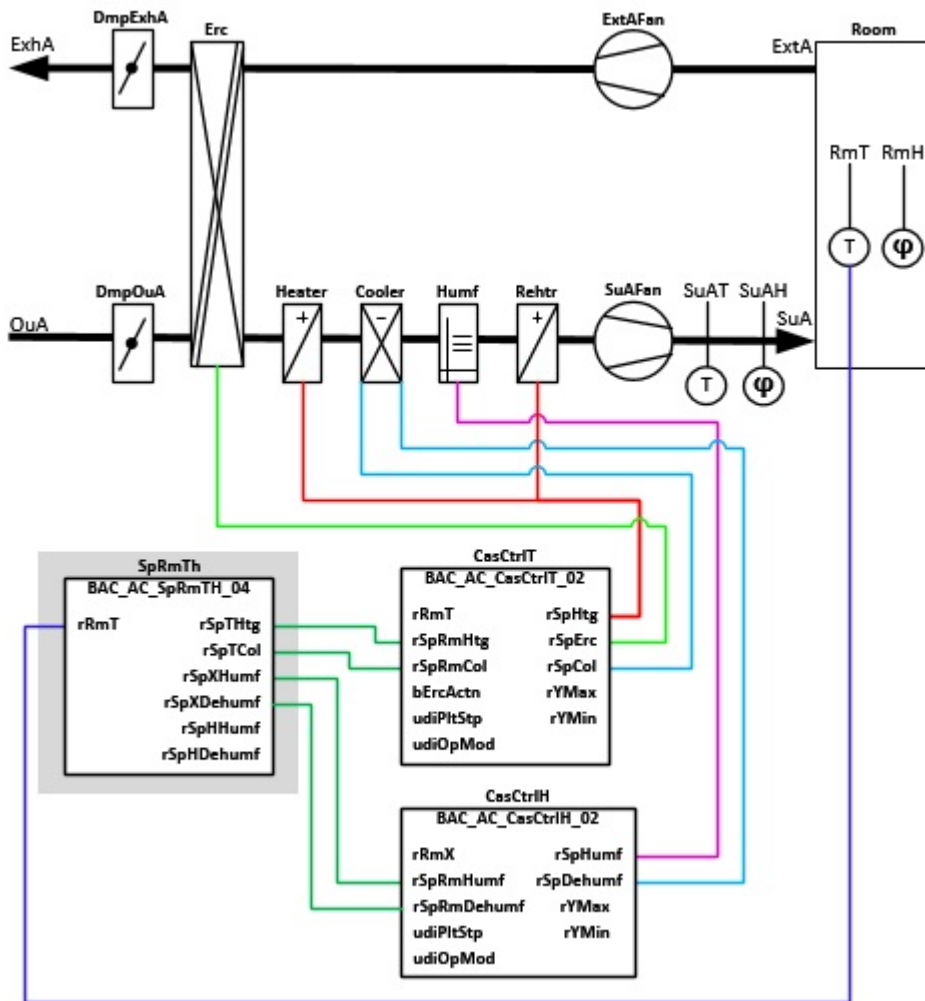
Die Eingabe der relativen Raumfeuchtesollwerte (%) erfolgt mit den BACnet-AV-Objekten **SpHHumf** (**Befeuchten**) und **SpHDehumf** (**Entfeuchten**). Es gibt eine energieneutrale Zone zwischen dem unteren (**Befeuchten**) und oberen Sollwert (**Entfeuchten**), welche durch die Funktion **SpHSupVis** überwacht wird.

Die beiden Sollwerte für die relative Raumfeuchte werden innerhalb des Templates mittels der aktuellen Raumtemperatur in zwei Sollwerte für die absolute Raumfeuchte (g/Kg) umgerechnet, **SpXHumf** (**Befeuchten**) und **SpXDehumf** (**Entfeuchten**).

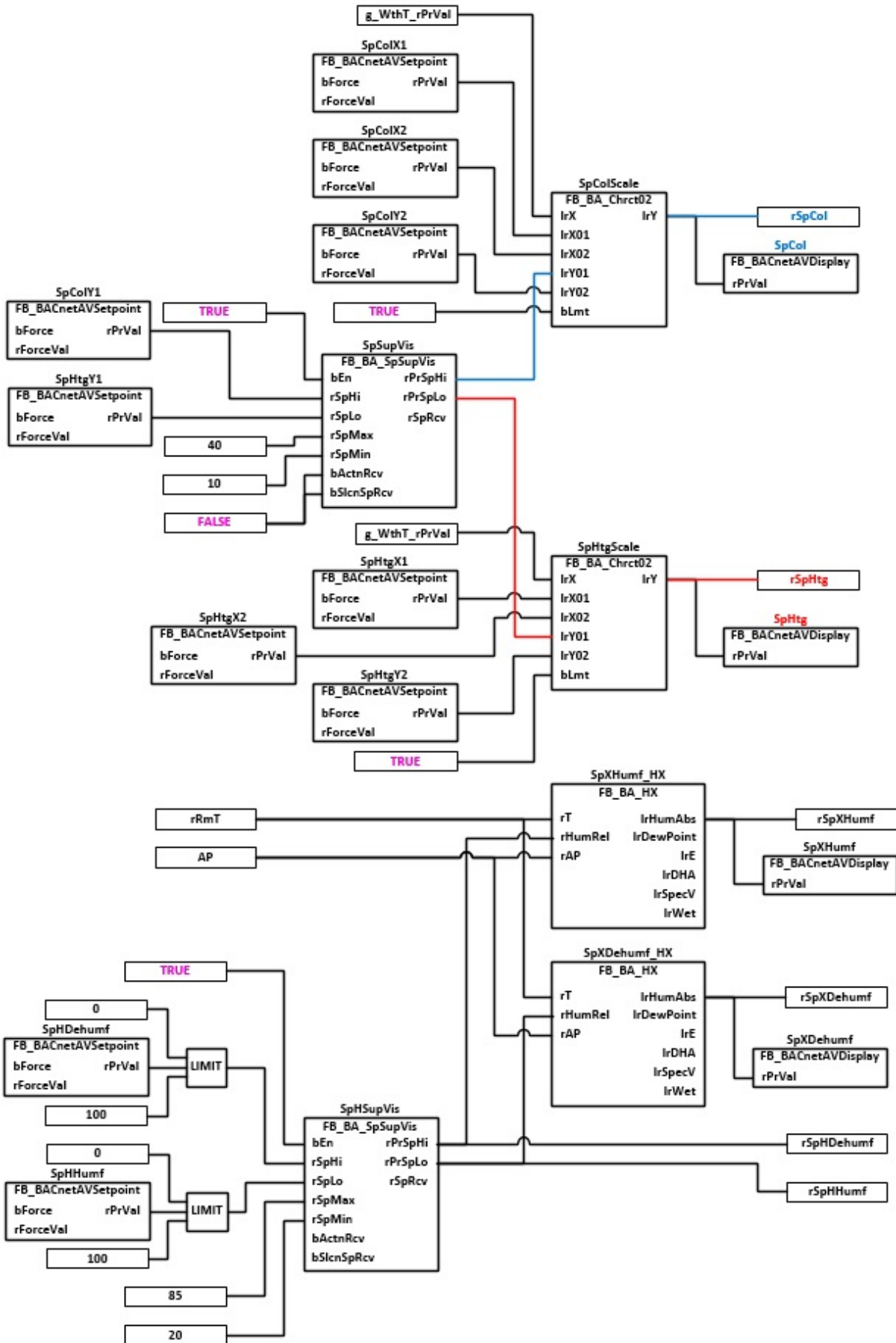
Schnittstelle



Anlagenschema



Blockschaltbild



VAR_Input

rRmT : REAL;

rRmT: Messwert der Raumtemperatur

VAR_OUTPUT

rSpTHtg : REAL;
 rSpTCol : REAL;
 rSpXHumf : REAL;
 rSpXDehumf : REAL;
 rSpHHumf : Real;
 rSpHDehumf : Real;

rSpTHtg: Errechneter Sollwert Heizen der Raumtemperatur

rSpTCol: Errechneter Sollwert Kühlen der Raumtemperatur

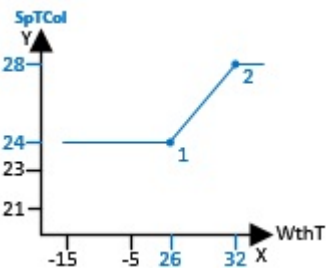
rSpXHumf: Errechneter Sollwert Absolute Feuchte; Befeuchten

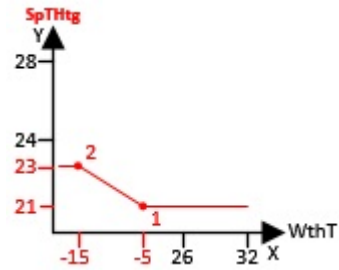
rSpXDehumf: Errechneter Sollwert Absolute Feuchte; Entfeuchten

rSpHHumf: Sollwert Relative Feuchte; Befeuchten

rSpHDehumf: Sollwert Relative Feuchte; Entfeuchten

Programmbeschreibung

Instanz	Typ	Aufgabe
SpTColX1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTColX1 Kühlen
SpTColX2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTColX2 Kühlen
SpTColY1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTColY1 Kühlen. Der Wert ist der Basissollwert für das Kühlen
SpTColY2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTColY2 Kühlen
SpTColScale	FB_BA_Chrc04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie Kühlen für die aktuelle Raumtemperatur in Abhängigkeit der Aussentemperatur. 
SpTCol	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumtemperatur-Sollwertes Kühlen
SpTHtgX1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTHtgX1 Heizen
SpTHtgX2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTHtgX2 Heizen
SpTHtgY1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTHtgY1 Heizen. Der Wert ist der Basissollwert für das Heizen
SpTHtgY2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt SpTHtgY2 Heizen

Instanz	Typ	Aufgabe
SpTHtgScale	FB_BA_Chrct04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie Heizen für die aktuelle Raumtemperatur in Abhängigkeit der Außentemperatur. 
SpTHtg	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Raumtemperatur-Sollwertes Heizen
SpTSupVis	FB_BA_SpSupVis [▶ 246]	Der Funktionsbaustein dient zur Kontrolle der energieneutralen Zone der beiden Sollwerte SpTCol und SpTHtg
SpHHumf	FB_BACnetAVSetpoint [▶ 70]	Eingabe des relativen Raumfeuchtesollwertes Befeuchten(%).
SpHDehumf	FB_BACnetAVSetpoint [▶ 70]	Eingabe des relativen Raumfeuchtesollwertes Entfeuchten(%).
SpHSupVis	FB_BA_SpSupVis [▶ 246]	Der Funktionsbaustein dient zur Kontrolle der beiden Sollwerte SpHDehumf und SpHHumf
SpXHumf_HX	FB_BA_HX [▶ 230]	Mit diesem Funktionsbaustein wird der Sollwert der absoluten Raumfeuchte, Befeuchten, berechnet. Für die Berechnung der Größe wird die Raumtemperatur, der Sollwert der relativen Feuchte SpHHumf und der barometrische Luftdruck benötigt.
SpXHumf	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten absoluten Raumfeuchte-Sollwertes Befeuchten (g/Kg).
SpXDehumf_HX	FB_BA_HX [▶ 230]	Mit diesem Funktionsbaustein wird der Sollwert der absoluten Raumfeuchte, Entfeuchten, berechnet. Für die Berechnung der Größe wird die Raumtemperatur, der Sollwert der relativen Feuchte SpHDehumf und der barometrische Luftdruck benötigt.
SpXDehumf	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten absoluten Raumfeuchte-Sollwertes Entfeuchten (g/Kg).

Versionshistorie

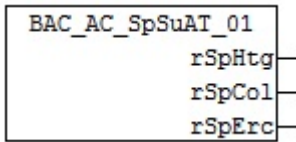
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.11 BAC_AC_SpSuAT_01

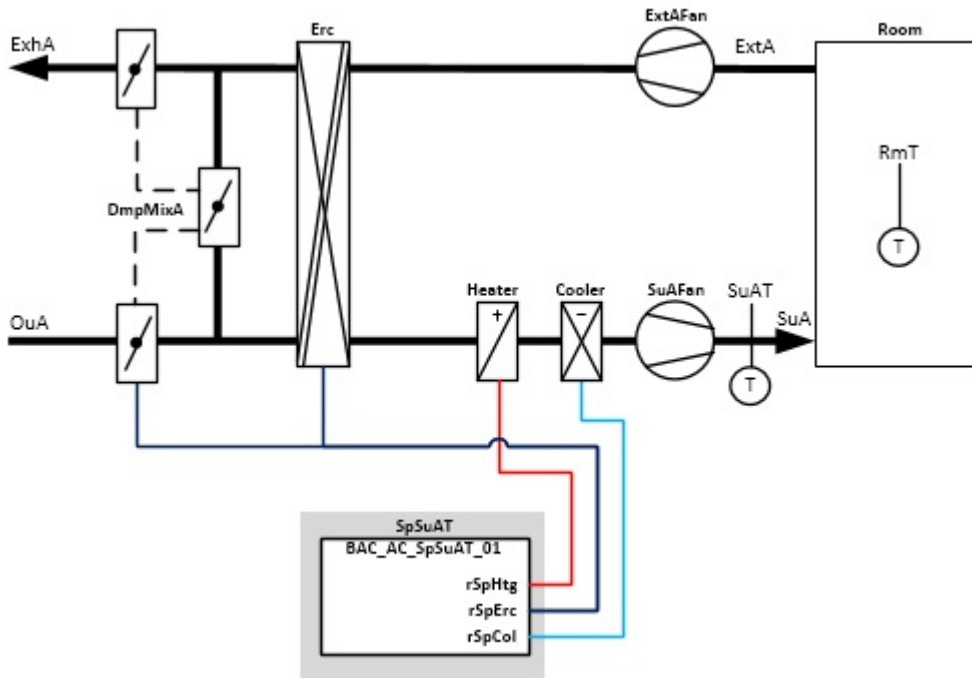
Anwendung

Sollwertprogramm für eine Zulufttemperatur Regelung mit einem Zulufttemperatur Sollwert inklusive Sommer- und Winterkompensation über eine Kennlinie.

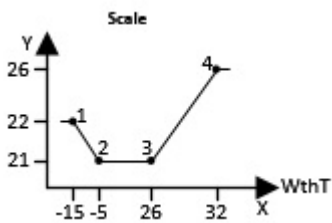
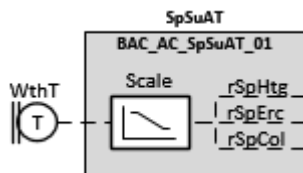
Schnittstelle



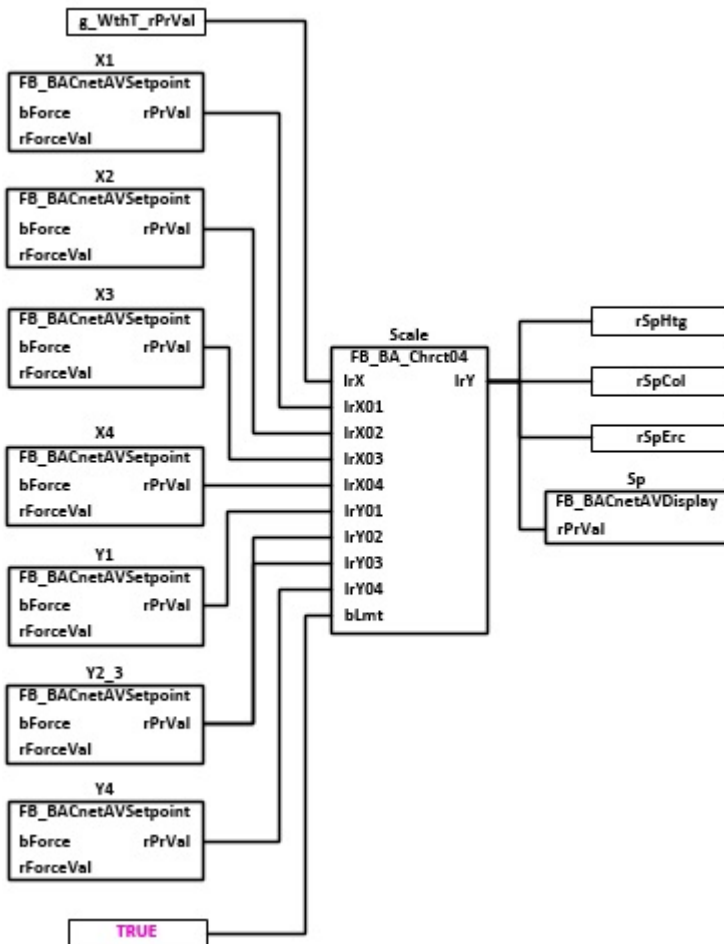
Anlagenschema 01



Anlagenschema 02



Blockschaltbild



VAR_OUTPUT

```
rSpHtg    : REAL;
rSpErc    : REAL;
rSpCol    : REAL;
```

rSpHtg: Errechneter Sollwert der Zulufttemperatur für das Heizen

rSpErc: Errechneter Sollwert der Zulufttemperatur für die Energierückgewinnung

rSpCol: Errechneter Sollwert der Zulufttemperatur für das Kühlen

Programmbeschreibung

Instanz	Typ	Aufgabe
X1	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X1
X2	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X2
X3	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X3
X4	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X4
Y1	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y1
Y2_3	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für die Stützpunkte Y2/Y3
Y4	FB BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y4
Scale	FB BA Chrct04 [▶ 175]	Der Funktionsbaustein berechnet die Sollwertkennlinie für die aktuelle Raumtemperatur in Abhängigkeit der Außentemperatur.
Sp	FB BACnetAVDisplay [▶ 69]	Ausgabe des errechneten, einfachen Raumtemperatur-Sollwertes. Dieser wird an die Ausgänge rSpHtg , rSpCol und rSpErc ausgegeben.

Versionshistorie

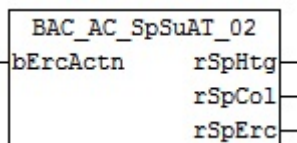
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.80.12 BAC_AC_SpSuAT_02

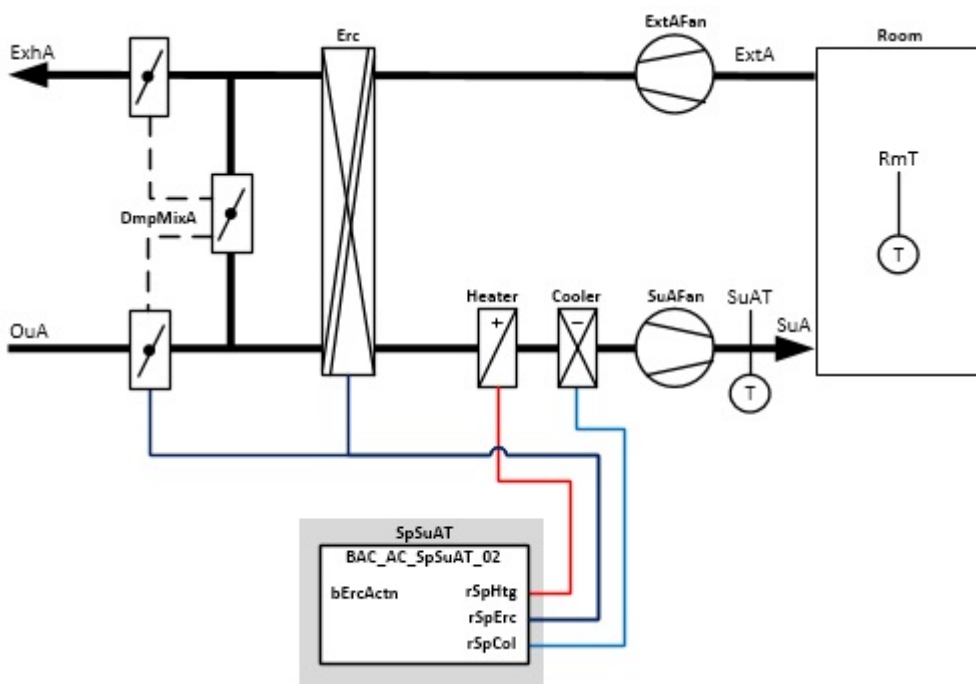
Anwendung

Sollwertprogramm für eine Zulufttemperatur Regelung mit separaten Zuluftsollwerten für das Heizen, Kühlen und der Energierückgewinnung inklusive zwei separaten Sollwertkennlinien für die Sommer- und Winterkompensation.

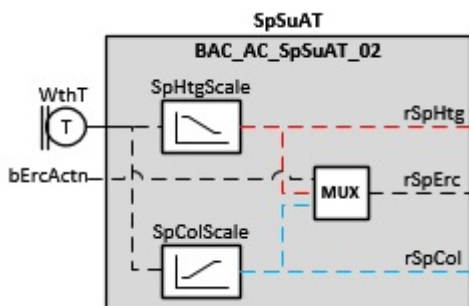
Schnittstelle



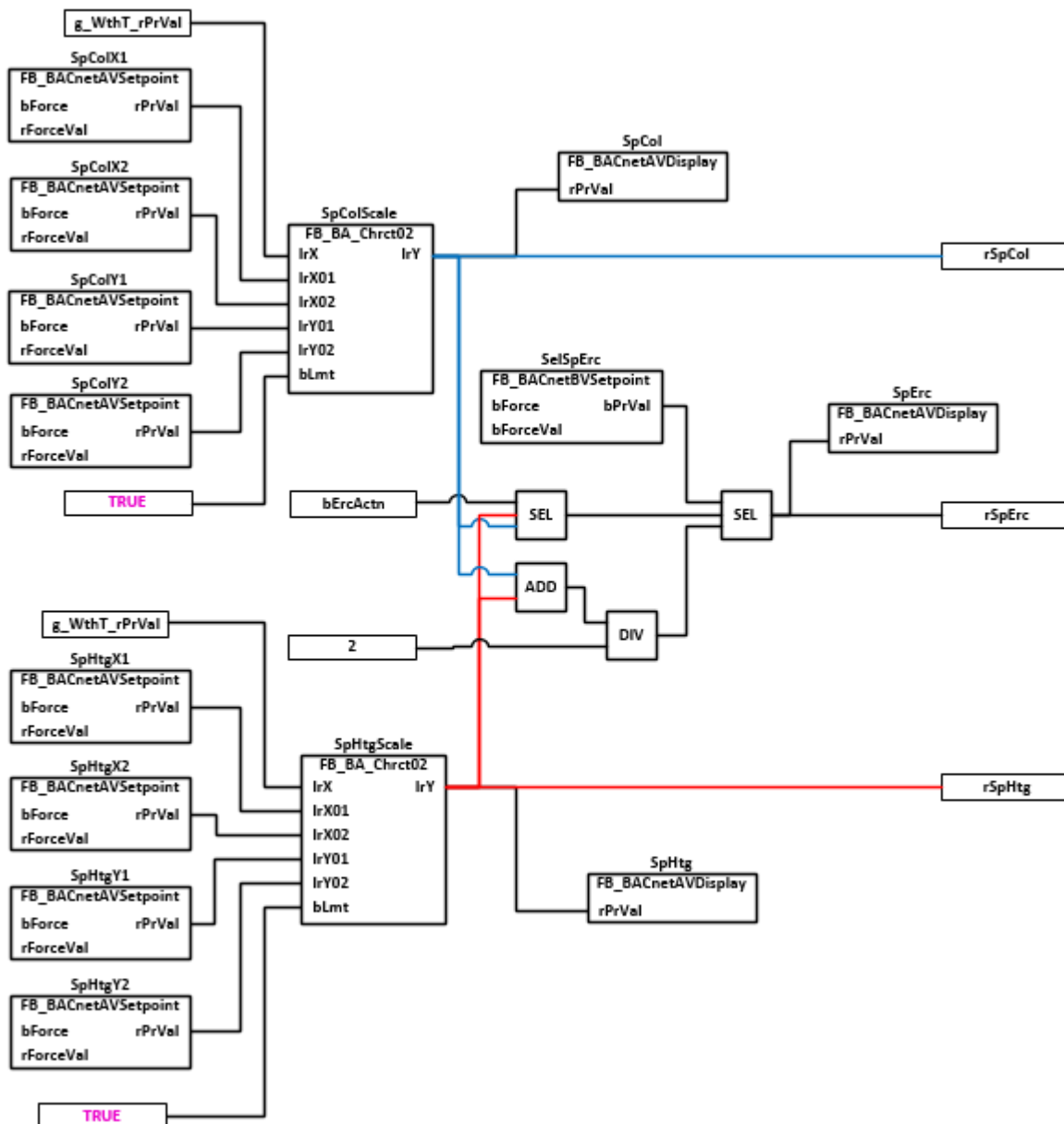
Anlagenschema 01



Anlagenschema 02



Blöckschaltbild



VAR_OUTPUT

bErcActn : BOOL;

bErcActn: Eingangsvariable an der der Wirksinn der Energierückgewinnung angelegt wird. In Abhängigkeit des Wirksinns wird der Sollwert für die Energierückgewinnung bestimmt.
TRUE = Direct = Kühlen; FALSE = Reverse = Heizen

VAR_OUTPUT

rSpHtg : REAL;

rSpErc : REAL;

rSpCol : REAL;

rSpHtg: Errechneter Sollwert der Zulufttemperatur für das Heizen

rSpErc: Errechneter Sollwert der Zulufttemperatur für die Energierückgewinnung

rSpCol: Errechneter Sollwert der Zulufttemperatur für das Kühlen

Programmbeschreibung

Instanz	Typ	Aufgabe
SpColX1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X1 der Sommerkompensation für den Sollwert Kühlen.
SpColX2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X2 der Sommerkompensation für den Sollwert Kühlen.
SpColY1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y1 der Sommerkompensation für den Sollwert Kühlen. SpColY1 ist der Basis-Sollwert für das Kühlen.
SpColY2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y2 der Sommerkompensation für den Sollwert Kühlen.
SpColScale	FB_BA_Chrc02 [▶ 173]	Der Funktionsbaustein ermittelt den Sommerkompensationswert für den Sollwert Kühlen.
SpHtgX1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X1 der Winterkompensation für den Sollwert Heizen.
SpHtgX2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt X2 der Winterkompensation für den Sollwert Heizen.
SpHtgY1	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y1 der Winterkompensation für den Sollwert Heizen. SpHtgY1 ist der Basis-Sollwert für das Heizen.
SpHtgY2	FB_BACnetAVSetpoint [▶ 70]	Eingabe des Wertes für den Stützpunkt Y2 der Winterkompensation für den Sollwert Heizen.
SpHtgScale	FB_BA_Chrc02 [▶ 173]	Der Funktionsbaustein ermittelt den Winterkompensationswert für den Sollwert Heizen.
SpSuVis	FB_BA_SpSupVis [▶ 246]	Der Funktionsbaustein dient zur Kontrolle der beiden Stützpunkte SpHtgY1 und SpColY1.
SelSpErc	FB_BACnetBVSetpoint [▶ 97]	Auswahl Strategie Sollwert Energierückgewinnung. TRUE: Mittelwert aus PrSpHtg und PrSpCol ; FALSE: Wirksinnabhängig, definiert durch den Eingang bErcActn
	SEL, ADD, DIV, SEL	Das Ergebnis dieses Netzwerkes liefert 2 Sollwert für die Energierückgewinnung. Welcher genommen wird, ist abhängig von der Strategie Sollwert Energierückgewinnung, siehe <i>SelSpErc</i>
SpHtg	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Zulufttemperatur-Sollwertes Heizen
SpCol	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Zulufttemperatur-Sollwertes Kühlen
SpErc	FB_BACnetAVDisplay [▶ 69]	Ausgabe des errechneten Zulufttemperatur-Sollwertes für die Energierückgewinnung

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81 BACnet Objekte

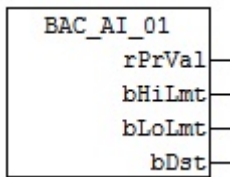
9.81.1 BAC_AI_01

Funktionsbeschreibung

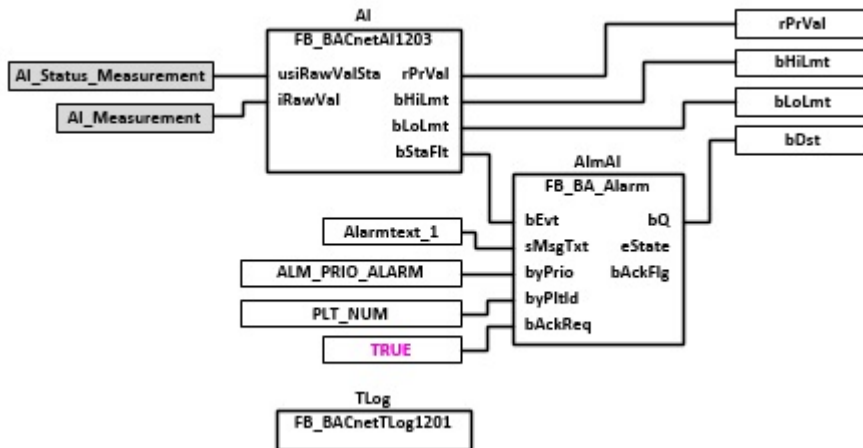
Das Template erfasst einen analogen Eingangswert von einer Busklemme (siehe **IO-Verknüpfung**) und rechnet ihn in einen Real-Prozesswert um. Außerdem wird das **Status-Flag Fault** des **AI**-Objekts ausgewertet und vom Alarmbaustein **AlmA1** erfasst und verarbeitet.

Das Template wird eingesetzt, damit ein analoges Eingangssignal (z. B. Messwert) im Programm als Real-Prozesswert verarbeitet werden kann.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```

rPrVal    : REAL;
bHiLmt    : BOOL;
bLoLmt    : BOOL;
bDst      : BOOL;
  
```

rPrVal: Aktueller Wert des Analog-Input-Objekts.

bHiLmt: Meldung oberer Grenzwert erreicht.

bLoLmt: Meldung unterer Grenzwert erreicht.

bDst: Zeigt den Zustand des Statusflags „Fault“ des Analog-Input-Objekts an. Nach Behebung des Fehlers kann die Meldung quittierungspflichtig sein. Dieses ist abhängig von der Parametrierung des Eingangs **bAckReq** des Funktionsbausteins **AlmAi**.

VAR CONSTANT

```

PLT_NUM    : BYTE := 1;
  
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
AI	FB_BACnetAI1203 [▶ 49]		Analog Input Objekt zur Anzeige des Messwertes
AlmAi	FB_BA_Alarm [▶ 182]		Erfassung und Weiterverarbeitung des Status-Flags Fault des Messwertes Aktionen, die nach dem Eingang Status-Flags Fault ausgelöst erfolgen sollen, können im Template am Funktionsbaustein AlmAi parametrieren werden.
TLog	FB_BACnetTLog1201 [▶ 137]		Trendaufzeichnung des Messwertes des AI-Objekts

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
AI_Status_Measurement	USINT		Eingang	Analogeingang - Statusbyte Busklemme
AI_Measurement	INT		Eingang	Analogeingang - Messwert Busklemme

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

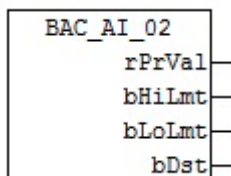
9.81.2 BAC_AI_02

Funktionsbeschreibung

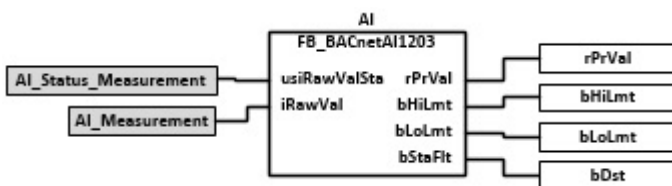
Das Template erfasst einen analogen Eingangswert von einer Busklemme (siehe **IO-Verknüpfung**) und rechnet ihn in einen Real-Prozesswert um.

Das Template wird eingesetzt, damit ein analoges Eingangssignal (z. B. Messwert) im Programm als Real-Prozesswert verarbeitet werden kann.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```
rPrVal   : REAL;
bHiLmt   : BOOL;
bLoLmt   : BOOL;
bDst     : BOOL;
```

rPrVal: Aktueller Wert des Analog-Input-Objekts.

bHiLmt: Meldung oberer Grenzwert erreicht.

bLoLmt: Meldung unterer Grenzwert erreicht.

bDst: Zeigt den Zustand des Statusflags „Fault“ des Analog-Input-Objekts an. Nach Behebung des Fehlers kann die Meldung quittierungspflichtig sein. Dieses ist abhängig von der Parametrierung des Eingangs

bAckReq des Funktionsbausteins **AlmAi**.

VAR CONSTANT

```
PLT_NUM   : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
AI	FB_BACnetAI1203 [► 49]		Analog Input Objekt zur Anzeige des Messwertes

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
AI_Status_Measurement	USINT		Eingang	Analogeingang - Statusbyte Busklemme
AI_Measurement	INT		Eingang	Analogeingang - Messwert Busklemme

Versionshistorie

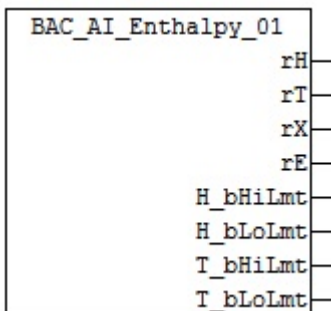
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.3 BAC_AI_Enthalpy_01

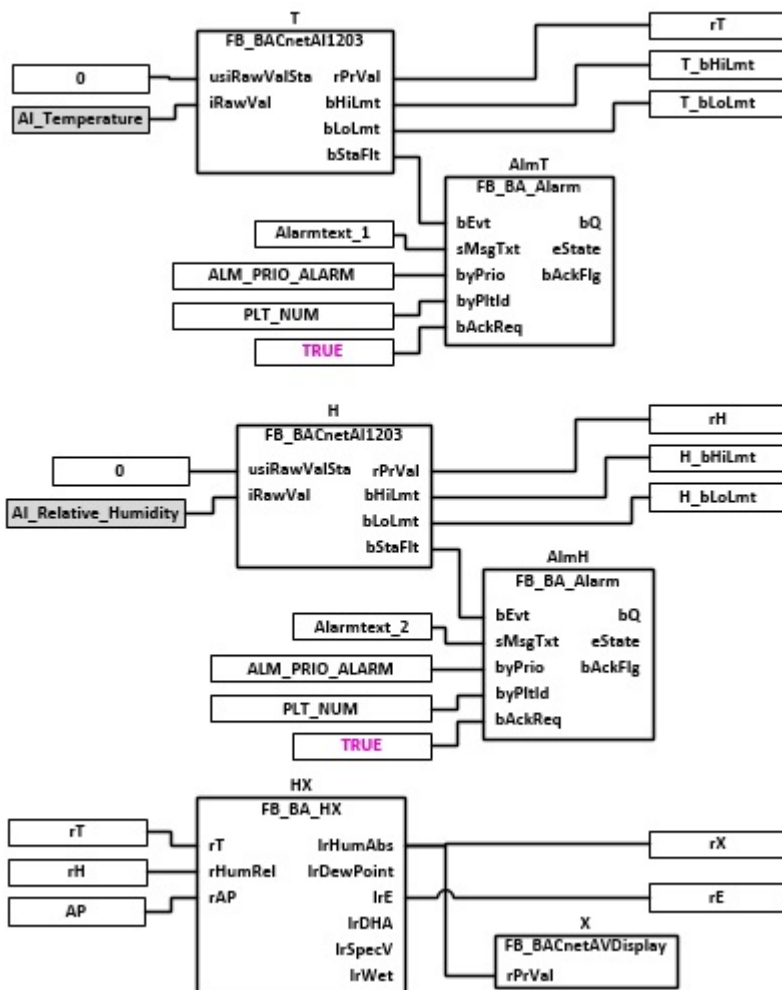
Funktionsbeschreibung

Das Template erfasst die beiden analogen Eingangswerte **Temperatur** und **Relative Feuchte** von einer Busklemme (siehe IO-Verknüpfung) und rechnet diese in Real-Prozesswerte um. Mit dem Funktionsbaustein **HX** werden die **Taupunkttemperatur**, die spezifische **Enthalpie**, **Feuchtkugelttemperatur** und die **Absolute Feuchte** berechnet.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```

rH      : REAL;
rT      : REAL;
rX      : REAL;
  
```

```
rE      : REAL;
H_bHiLmt : BOOL;
H_bLoLmt : BOOL;
T_bHiLmt : BOOL;
T_bLoLmt : BOOL;
```

rH: Aktueller Wert des Analog-Input-Objekts **H - Relative Feuchte [%]**

rT: Aktueller Wert des Analog-Input-Objekts **T - Temperatur [°C]**

rX: Aktueller Wert des Analog-Value-Objekts **X - Absolute Feuchte g Wasser je kg trockener Luft [g/Kg]**

rE: Berechnter Wert **Enthalpie [kJ/kg]**

H_bHiLmt: Relative Feuchte High Limit

H_bLoLmt: Relative Feuchte Low Limit

T_bHiLmt: Temperatur High Limit

T_bLoLmt: Temperatur Low Limit

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB BA Alarm**. [[▶ 182](#)]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [[▶ 372](#)] mittels des Funktionsbausteins **FB BA AlarmPlt**. [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB BA ComnMsg** [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
T	FB BACnetAI1203 [▶ 49]		Analog Input Objekt zur Anzeige der Temperatur
AlmT	FB BA Alarm [▶ 182]		Erfassung und Weiterverarbeitung des Status-Flags Fault der Temperatur Aktionen, die nach dem Eingang Status-Flags Fault ausgelöst erfolgen sollen, können im Template am Funktionsbaustein AlmT parametrisiert werden.
T_TLog	FB BACnetTLog1201 [▶ 137]		Trendaufzeichnung des Messwertes des AI-Objekts T
H	FB BACnetAI1203 [▶ 49]		Analog Input Objekt zur Anzeige der Relativen Feuchte
AlmH	FB BA Alarm [▶ 182]		Erfassung und Weiterverarbeitung des Status-Flags Fault der Relativen Feuchte Aktionen, die nach dem Eingang Status-Flags Fault ausgelöst erfolgen sollen, können im Template am Funktionsbaustein AlmH parametrisiert werden.

Instanz	Typ	optional	Aufgabe
H_TLog	FB_BACnetTLog1201 [▶ 137]		Trendaufzeichnung des Messwertes des AI-Objekts H
HX	FB_BA_HX [▶ 230]		Mit diesem Funktionsbaustein werden die Taupunkttemperatur, die spezifische Enthalpie und die absolute Feuchte berechnet. Für die Berechnung der Größen werden die Temperatur, die relative Feuchte und der barometrische Luftdruck benötigt. Die Enthalpie ist ein Maß für die Energie eines thermodynamischen Systems.
X	FB_BACnetAVDisplay [▶ 69]		Analog Value Objekt zur Anzeige der Absoluten Feuchte
X_TLog	FB_BACnetTLog1201 [▶ 137]		Trendaufzeichnung des Messwertes des AV-Objekts X

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessab- bild	
AI_Temperature	INT		Eingang	Analogeingang - Temperatur
AI_Relative_Humidity	INT		Eingang	Analogeingang - Relative Feuchte

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.4 BAC_AI_WthT_01

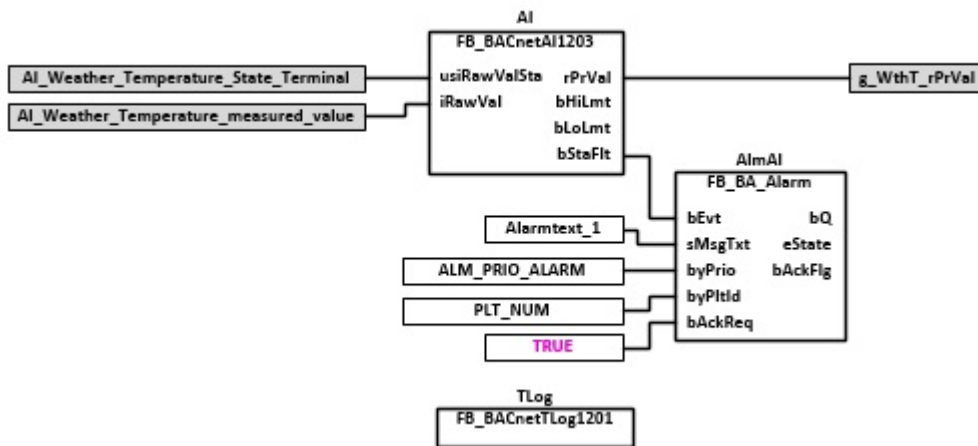
Funktionsbeschreibung

Das Template bildet die Witterungstemperatur ab. Es erfasst den analogen Eingangswert von einer Busklemme (siehe **IO-Verknüpfung**) und rechnet ihn in einen Real-Prozesswert um. Dieser Wert wird dann auf die globale Variable `g_WthT_rPrVal` [▶ 364] geschrieben. Außerdem wird das **Status-Flag Fault** des **AI**-Objekts ausgewertet und vom Alarmbaustein **AlmAI** erfasst und verarbeitet.

Schnittstelle

BAC_AI_WthT_01

Blockschaltbild



VAR CONSTANT

```
PLT_NUM : BYTE := 0;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins `FB_BA_Alarm`. [\[► 182\]](#)

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates `BAC_PltAlm_01` [\[► 372\]](#) mittels des Funktionsbausteins `FB_BA_AlarmPlt`. [\[► 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates `BAC_PltComnMsg_01` durch den Funktionsbaustein `FB_BA_CmnMsg` [\[► 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
AI	<code>FB_BACnetAI1203</code> [► 49]		Analog Input Objekt zur Anzeige des Messwertes Außentemperatur
AlmAI	<code>FB_BA_Alarm</code> [► 182]		Erfassung und Weiterverarbeitung des Status-Flags Fault des Messwertes Außentemperatur Aktionen, die nach dem Eingang Status-Flags Fault ausgelöst erfolgen sollen, können im Template am Funktionsbaustein AlmAI parametrisiert werden.
TLog	<code>FB_BACnetTLog1201</code> [► 137]		Trendaufzeichnung des Messwertes des AI-Objekts

IO-Verknüpfung

In der zum Template gehörigen XML-Description sind in dem Bereich **Parameter** Variablen mit der Kennung **INPUT** oder **Output** deklariert. Diese Parameter können im Project Builder oder über die Excel-Import-Schnittstelle mit dem Prozessabbild der Eingangs- und Ausgangsebene in der SPS verknüpft werden.

Parameter	Typ	Instanz	Typ	Prozessab- bild	
AI_Weather_Temperature_State_Terminal	USINT	AI	FB_BACnetAI1203 [▶ 49]	Eingang	Analogeingang - Witterungstemperatur - Status Busklemme
AI_Weather_Temperature_measured_value	INT	AI	FB_BACnetAI1203 [▶ 49]	Eingang	Analogeingang - Witterungstemperatur - Messwert

Versionshistorie

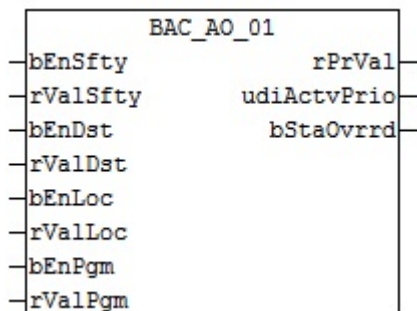
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.5 BAC_AO_01

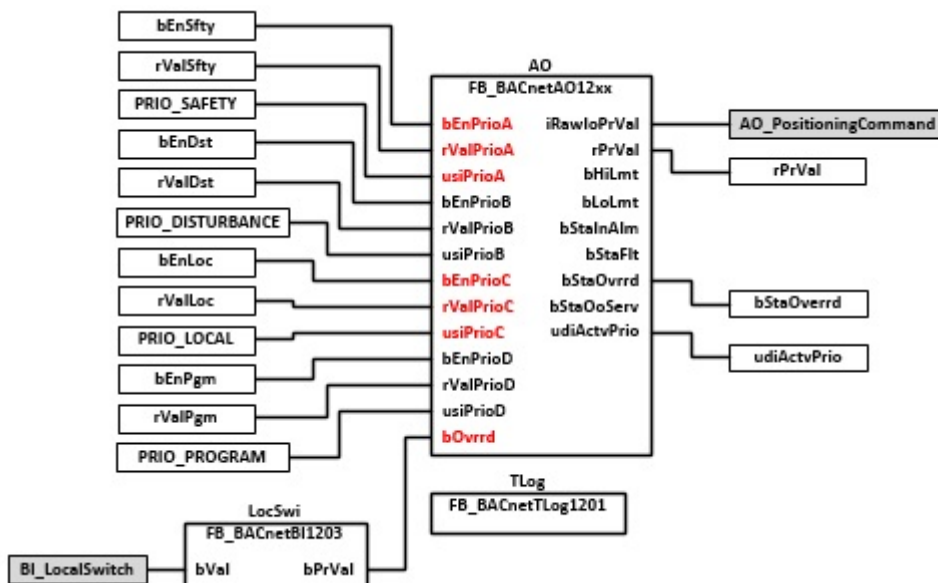
Funktionsbeschreibung

Das Template ermittelt den aktuellen Stellwert aus mehreren, priorisierten Sollwerten (Prioritätsmatrix, Eingänge des Templates) und übermittelt den Stellwert an eine Busklemme (siehe **IO-Verknüpfung**). Außerdem wird der manuelle Handeingriff mittels eines digitalen Eingangs **LocSwi** erfasst.

Schnittstelle



Blockschaltbild



VAR_INPUT

```

bEnSfty      : BOOL;
rValSfty     : REAL;
bEnDst       : BOOL;
rValDst      : REAL;
bEnLoc       : BOOL;
rValLoc      : REAL;
bEnPgm       : BOOL;
rValPgm      : REAL;

```

bEnSfty: Freigabe Sicherheitspriorität

rValSfty: Analoger Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität.

rValDst: Analoger Wert Störungspriorität.

bEnLoc: Freigabe Priorität manueller Eingriff

rValLoc: Analoger Wert Priorität manueller Eingriff

bEnPgm: Freigabe Programmpriorität

rValPgm: Analoger Wert Programmpriorität

VAR_OUTPUT

```

rPrVal       : REAL;
udiActvPrio  : UDINT;
bStaOverrd   : BOOL;

```

rPrVal: Aktueller Wert des Analog-Output-Objekts.

udiActvPrio: Anzeige der aktuellen Priorität des AO-Objekts

bStaOverrd: Anzeige, dass das AO-Objekt durch einen lokalen Mechanismus übersteuert wird, siehe **LocSwi**

VAR CONSTANT

```

PLT_NUM      : BYTE := 1;

```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [► 182]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [► 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe						
AO	FB_BACnetAO1203 [► 53]		AO-Objekt für die Ausgabe des Stellwertes						
			<table border="1"> <tr> <td>Priorität:</td> <td>Freigabe</td> <td>Wert</td> </tr> <tr> <td>PRIO_SAFETY (1)</td> <td>Eingang bEnSfty</td> <td>Eingang rValSfty</td> </tr> </table>	Priorität:	Freigabe	Wert	PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty
Priorität:	Freigabe	Wert							
PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty							

Instanz	Typ	optional	Aufgabe		
			PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst
			PRIO_LOCAL (8)	Eingang bEnLoc	Eingang rValLoc
			PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang rValPgm
LocSwi	FB_BACnetBI1203 [▶ 72]		Digitales Eingangs-Objekt zur Anzeige des Schalters für die lokale Übersteuerung des Stellwertes des AO-Objekts		
TLog	FB_BACnetTLog1201 [▶ 137]		Trendaufzeichnung des Stellwertes des AO-Objekts		

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_LocalSwitch	BOOL		Eingang	Digitaleingang - Schalter Hand - Meldung - Hand/Auto
AO_PositioningCommand	INT		Ausgang	Analog Ausgang - Stellbefehl

Versionshistorie

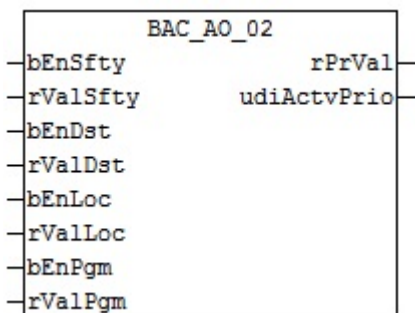
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.6 BAC_AO_02

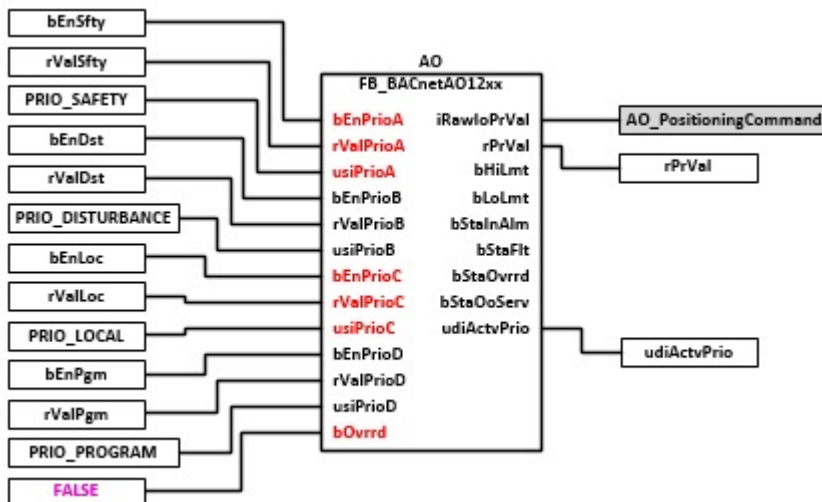
Funktionsbeschreibung

Das Template ermittelt den aktuellen Stellwert aus mehreren, priorisierten Sollwerten (Prioritätsmatrix, Eingänge des Templates) und übermittelt den Stellwert an eine Busklemme (siehe **IO-Verknüpfung**).

Schnittstelle



Blockschaltbild



VAR_INPUT

```

bEnSfty      : BOOL;
rValSfty     : REAL;
bEnDst       : BOOL;
rValDst      : REAL;
bEnLoc       : BOOL;
rValLoc      : REAL;
bEnPgm       : BOOL;
rValPgm      : REAL;

```

bEnSfty: Freigabe Sicherheitspriorität

rValSfty: Analoger Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität.

rValDst: Analoger Wert Störungspriorität.

bEnLoc: Freigabe Priorität manueller Eingriff

rValLoc: Analoger Wert Priorität manueller Eingriff

bEnPgm: Freigabe Programmpriorität

rValPgm: Analoger Wert Programmpriorität

VAR_OUTPUT

```

rPrVal       : REAL;
udiActvPrio  : UDINT;

```

rPrVal: Aktueller Wert des Analog-Output-Objekts.

udiActvPrio: Anzeige der aktuellen Priorität des AO-Objekts

VAR CONSTANT

```

PLT_NUM      : BYTE := 1;

```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins `FB_BA_Alarm`. [► 182]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates `BAC_PltAlm_01` [► 372] mittels des Funktionsbausteins `FB_BA_AlarmPlt`. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates `BAC_PltComnMsg_01` durch den Funktionsbaustein `FB_BA_ComnMsg` [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe															
AO	<u>FB_BACnetAO1203</u> [▶ 53]		AO-Objekt für die Ausgabe des Stellwertes															
			<table border="1"> <thead> <tr> <th>Priorität:</th> <th>Freigabe</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td>PRIO_SAFETY (1)</td> <td>Eingang bEnSfty</td> <td>Eingang rValSfty</td> </tr> <tr> <td>PRIO_DISTURBANCE (3)</td> <td>Eingang bEnDst</td> <td>Eingang rValDst</td> </tr> <tr> <td>PRIO_LOCAL (8)</td> <td>Eingang bEnLoc</td> <td>Eingang rValLoc</td> </tr> <tr> <td>PRIO_PROGRAM (15)</td> <td>Eingang bEnPgm</td> <td>Eingang rValPgm</td> </tr> </tbody> </table>	Priorität:	Freigabe	Wert	PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty	PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst	PRIO_LOCAL (8)	Eingang bEnLoc	Eingang rValLoc	PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang rValPgm
Priorität:	Freigabe	Wert																
PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty																
PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst																
PRIO_LOCAL (8)	Eingang bEnLoc	Eingang rValLoc																
PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang rValPgm																

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
AO_PositioningCommand	INT		Ausgang	Analog Ausgang - Stellbefehl

Versionshistorie

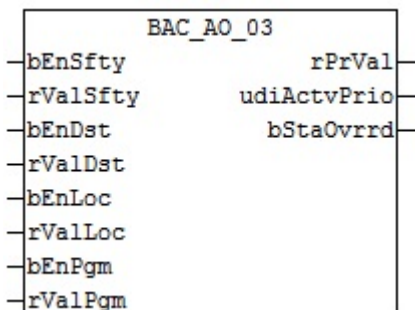
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.7 BAC_AO_03

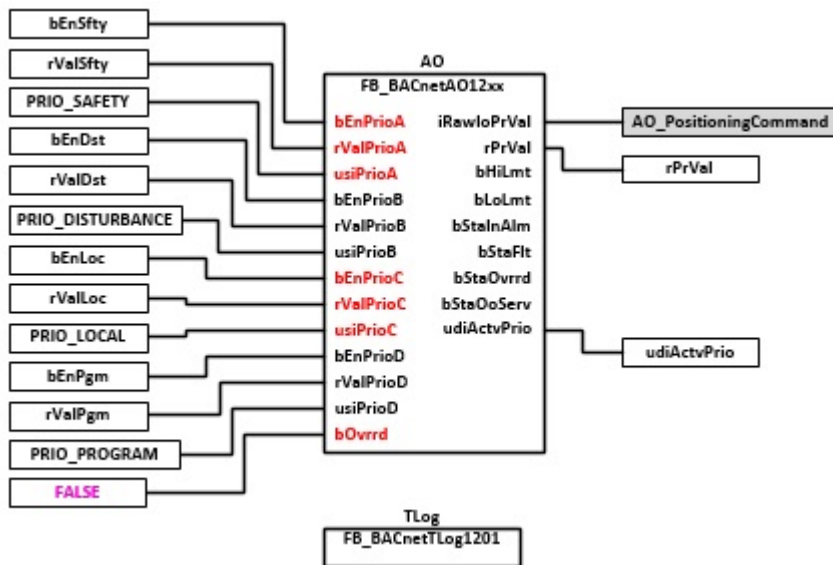
Funktionsbeschreibung

Das Template ermittelt den aktuellen Stellwert aus mehreren, priorisierten Sollwerten (Prioritätsmatrix, Eingänge des Templates) und übermittelt den Stellwert an eine Busklemme (siehe **IO-Verknüpfung**).

Schnittstelle



Blockschaltbild



VAR_INPUT

```

bEnSfty      : BOOL;
rValSfty     : REAL;
bEnDst      : BOOL;
rValDst     : REAL;
bEnLoc      : BOOL;
rValLoc     : REAL;
bEnPgm      : BOOL;
rValPgm     : REAL;

```

bEnSfty: Freigabe Sicherheitspriorität

rValSfty: Analoger Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität.

rValDst: Analoger Wert Störungspriorität.

bEnLoc: Freigabe Priorität manueller Eingriff

rValLoc: Analoger Wert Priorität manueller Eingriff

bEnPgm: Freigabe Programmpriorität

rValPgm: Analoger Wert Programmpriorität

VAR_OUTPUT

```

rPrVal      : REAL;
udiActvPrio : UDINT;

```

rPrVal: Aktueller Wert des Analog-Output-Objekts.

udiActvPrio: Anzeige der aktuellen Priorität des AO-Objekts

VAR CONSTANT

```

PLT_NUM     : BYTE := 1;

```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA Alarm](#). [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur

Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372]

mittels des Funktionsbausteins [FB_BA_AlarmPlt. \[► 186\]](#)

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB_BA_CmnMsg \[► 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe		
AO	FB_BACnetAO1203 [► 53]		AO-Objekt für die Ausgabe des Stellwertes		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty
			PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst
			PRIO_LOCAL (8)	Eingang bEnLoc	Eingang rValLoc
	PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang rValPgm		
TLog	FB_BACnetTLog1201 [► 137]		Trendaufzeichnung des Stellwertes des AO-Objekts		

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
AO_PositioningCommand	INT		Ausgang	Analog Ausgang - Stellbefehl

Versionshistorie

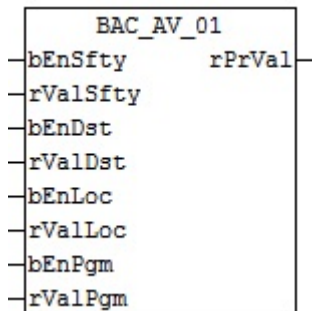
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.8 BAC_AV_01

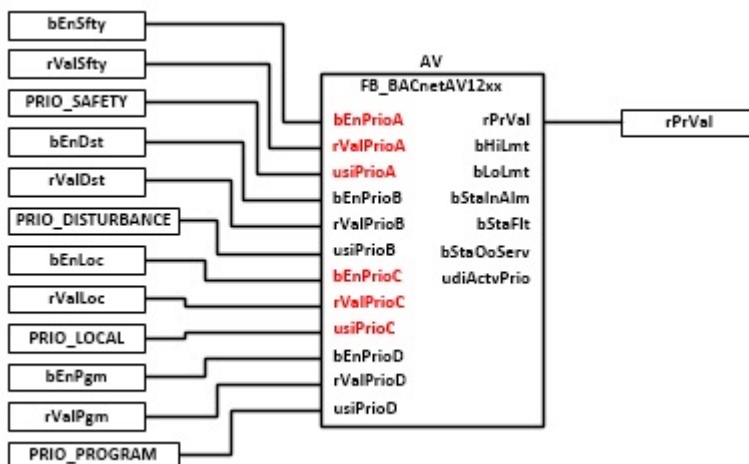
Funktionsbeschreibung

Das Template bildet einen REAL-Prozesswert aus mehreren, priorisierten Sollwerten (Prioritätsmatrix, Eingänge des Templates).

Schnittstelle



Blockschaltbild



VAR_INPUT

```

bEnSfty      : BOOL;
rValSfty     : REAL;
bEnDst       : BOOL;
rValDst      : REAL;
bEnLoc       : BOOL;
rValLoc      : REAL;
bEnPgm       : BOOL;
rValPgm      : REAL;
PRIO_PROGRAM : REAL;
  
```

bEnSfty: Freigabe Sicherheitspriorität

rValSfty: Analoger Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität.

rValDst: Analoger Wert Störungspriorität.

bEnLoc: Freigabe Priorität manueller Eingriff

rValLoc: Analoger Wert Priorität manueller Eingriff

bEnPgm: Freigabe Programmpriorität

rValPgm: Analoger Wert Programmpriorität

VAR_OUTPUT

```

rPrVal      : REAL;
  
```

rPrVal: Aktueller Wert des Analog-Value-Objekts.

VAR CONSTANT

```

PLT_NUM     : BYTE := 1;
  
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [▶ 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [▶ 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe															
AV	FB_BACnetAV1204 [▶ 67]		AV-Objekt für die Ausgabe des Stellwertes															
			<table border="1"> <thead> <tr> <th>Priorität:</th> <th>Freigabe</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td>PRIO_SAFETY (1)</td> <td>Eingang bEnSfty</td> <td>Eingang rValSfty</td> </tr> <tr> <td>PRIO_DISTURBANCE (3)</td> <td>Eingang bEnDst</td> <td>Eingang rValDst</td> </tr> <tr> <td>PRIO_LOCAL (8)</td> <td>Eingang bEnLoc</td> <td>Eingang rValLoc</td> </tr> <tr> <td>PRIO_PROGRAM (15)</td> <td>Eingang bEnPgm</td> <td>Eingang rValPgm</td> </tr> </tbody> </table>	Priorität:	Freigabe	Wert	PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty	PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst	PRIO_LOCAL (8)	Eingang bEnLoc	Eingang rValLoc	PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang rValPgm
Priorität:	Freigabe	Wert																
PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty																
PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst																
PRIO_LOCAL (8)	Eingang bEnLoc	Eingang rValLoc																
PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang rValPgm																

Versionshistorie

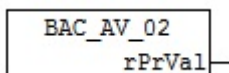
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.9 BAC_AV_02

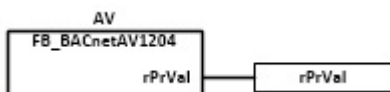
Funktionsbeschreibung

Das Template bildet einen REAL-Prozesswert im BACnet ab. Das Template wird eingesetzt um die Eingabe eines Sollwertes oder Parameters zu bedienen.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```
rPrVal : REAL;
```

rPrVal: Aktueller Wert des Analog-Value-Objekts.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
AV	FB_BACnetAV1204 [► 67]		AV-Objekt für die Ausgabe des Stellwertes

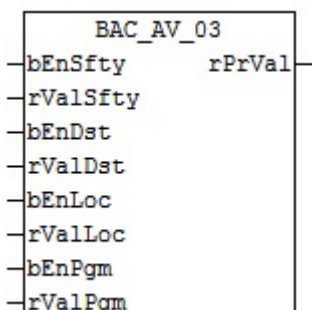
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

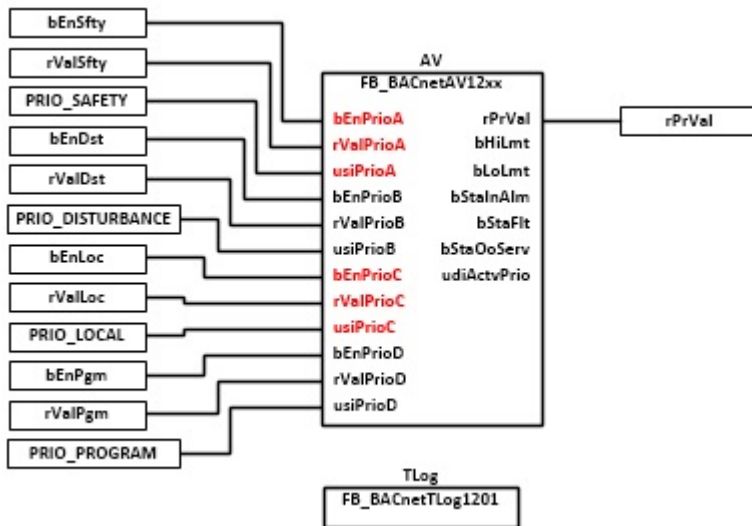
9.81.10 BAC_AV_03**Funktionsbeschreibung**

Das Template bildet einen REAL-Prozesswert aus mehreren, priorisierten Sollwerten (Prioritätsmatrix, Eingänge des Templates).

Der Prozesswert wird durch ein TrendLog-Objekt aufgezeichnet.

Schnittstelle

Blockschaltbild



VAR_INPUT

```
bEnSfty      : BOOL;
rValSfty     : REAL;
bEnDst      : BOOL;
rValDst     : REAL;
bEnLoc      : BOOL;
rValLoc     : REAL;
bEnPgm      : BOOL;
rValPgm     : REAL;
```

- bEnSfty:** Freigabe Sicherheitspriorität
- rValSfty:** Analoger Wert Sicherheitspriorität
- bEnDst:** Freigabe Störungspriorität.
- rValDst:** Analoger Wert Störungspriorität.
- bEnLoc:** Freigabe Priorität manueller Eingriff
- rValLoc:** Analoger Wert Priorität manueller Eingriff
- bEnPgm:** Freigabe Programmpriorität
- rValPgm:** Analoger Wert Programmpriorität

VAR_OUTPUT

```
rPrVal      : REAL;
```

rPrVal: Aktueller Wert des Analog-Value-Objekts.

VAR CONSTANT

```
PLT_NUM     : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt. Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA Alarm**. [[182](#)]
 Die Auswertung der Alarmergebnisse einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb der Templates **BAC_PltAlm_01** [[372](#)] mittels des Funktionsbausteins **FB_BA AlarmPlt**. [[186](#)]
 Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb der Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA ComnMsg** [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe		
AV	FB_BACnetAV1204 [▶ 67]		AV-Objekt für die Ausgabe des Stellwertes		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang rValSfty
			PRIO_DISTURBANCE (3)	Eingang bEnDst	Eingang rValDst
			PRIO_LOCAL (8)	Eingang bEnLoc	Eingang rValLoc
			PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang rValPgm
TLog	FB_BACnetTLog1201 1 [▶ 137]		Trendaufzeichnung des Prozesswertes des AV-Objekts		

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

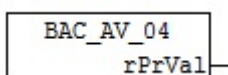
9.81.11 BAC_AV_04

Funktionsbeschreibung

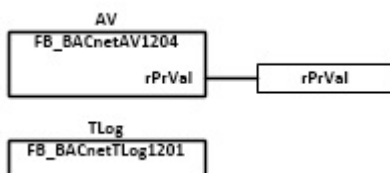
Das Template bildet einen REAL-Prozesswert im BACnet ab. Das Template wird eingesetzt um die Eingabe eines Sollwertes oder Parameters zu bedienen.

Der Prozesswert wird durch ein TrendLog-Objekt aufgezeichnet.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```
rPrVal : REAL;
```

rPrVal: Aktueller Wert des Analog-Value-Objekts.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmerbausteins **FB_BA_Alarm**. [▶ 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [▶ 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
AV	FB_BACnetAV1204 [▶ 67]		AV-Objekt für die Ausgabe des Stellwertes
TLog	FB_BACnetTLog1201 [▶ 137]		Trendaufzeichnung des Prozesswertes des AV-Objekts

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

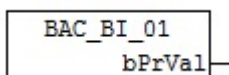
9.81.12 BAC_BI_01

Funktionsbeschreibung

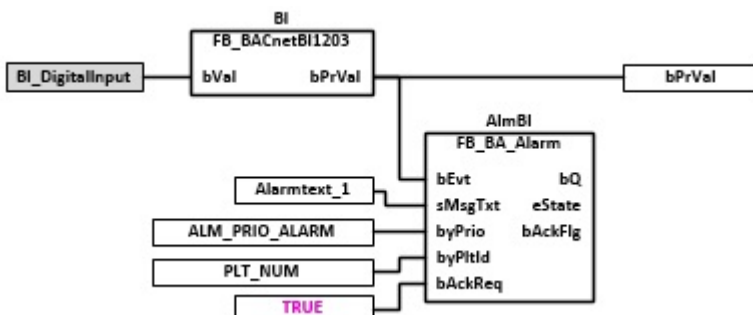
Das Template erfasst einen binären Eingangswert von einer Busklemme und gibt diesen als einen Boolean-Prozesswert aus. Außerdem wird der Prozesswert **bPrVal** des **BI**-Objekts ausgewertet und vom Alarmerbaustein **AlmBI** erfasst und verarbeitet.

Das Template wird eingesetzt, damit ein binäres Eingangssignal im Programm als Boolean-Prozesswert verarbeitet werden kann.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```
bPrVal : BOOL;
```

bPrVal: Aktueller Wert des Binary-Input-Objekts.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins `FB_BA_Alarm`. [► 182]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates `BAC_PltAlm_01` [► 372] mittels des Funktionsbausteins `FB_BA_AlarmPlt`. [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates `BAC_PltComnMsg_01` durch den Funktionsbaustein `FB_BA_ComnMsg` [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
BI	<code>FB_BACnetBI1203</code> [► 72]		Binary Input Objekt
AlmBI	<code>FB_BA_Alarm</code> [► 182]		Erfassung und Weiterverarbeitung des Prozesswertes bPrVal

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_DigitalInput	BOOL		Eingang	Digitaler Eingang Busklemme

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

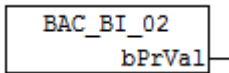
9.81.13 BAC_BI_02

Funktionsbeschreibung

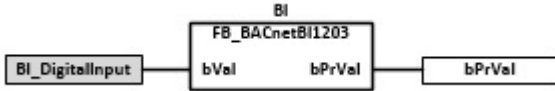
Das Template erfasst einen binären Eingangswert von einer Busklemme und gibt diesen als einen Boolean-Prozesswert aus.

Das Template wird eingesetzt, damit ein binäres Eingangssignal im Programm als Boolean-Prozesswert verarbeitet werden kann.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```
bPrVal : BOOL;
```

bPrVal: Aktueller Wert des Binary-Input-Objekts.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt. Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm. \[▶ 182\]](#) Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01 \[▶ 372\]](#) mittels des Funktionsbausteins [FB BA AlarmPlt. \[▶ 186\]](#) Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg \[▶ 200\]](#).

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
BI	FB_BACnetBI1203 [▶ 72]		Binary Input Objekt

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessab- bild	
BI_DigitalInput	BOOL		Eingang	Digitaler Eingang Busklemme

Versionshistorie

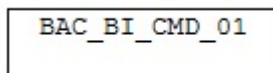
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.14 BAC_BI_CMD_01

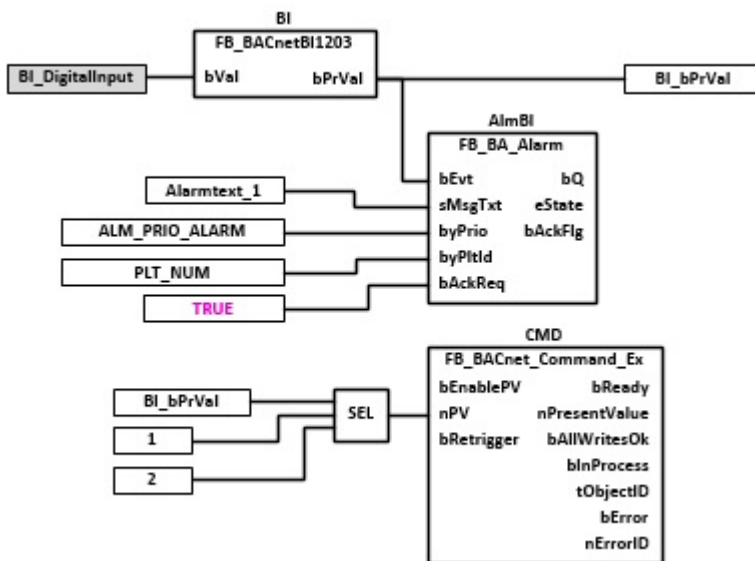
Funktionsbeschreibung

Das Template erfasst einen binären Eingangswert von einer Busklemme. Außerdem wird der Prozesswert **bPrVal** des **BI**-Objekts ausgewertet und vom Alarmbaustein **AlmBI** erfasst und verarbeitet. Aufgrund der Zustandsänderung an dem **BI**-Objekt kann über das Gruppenauftrags-Objekt **CMD** spezifizierte Properties anderer Objekte beschrieben werden.

Schnittstelle



Blockschaltbild



VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer **PLT_NUM** festgelegt.

Die Erfassung und Verarbeitung eines Alarmergebnisses von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [[182](#)]

Die Auswertung der Alarmergebnisse einer Anlage z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [[372](#)] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
BI	FB_BACnetBI1203 [72]		Binary Input Objekt

Instanz	Typ	optional	Aufgabe
AlmBI	FB_BA_Alarm [▶ 182]		Erfassung und Weiterverarbeitung des Prozesswertes BI_bPrVal
CMD	FB_BACnet_Command_Ex		Gruppenauftrag-Objekt

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_DigitalInput	BOOL		Eingang	Digitaler Eingang Busklemme

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.15 BAC_BO_01

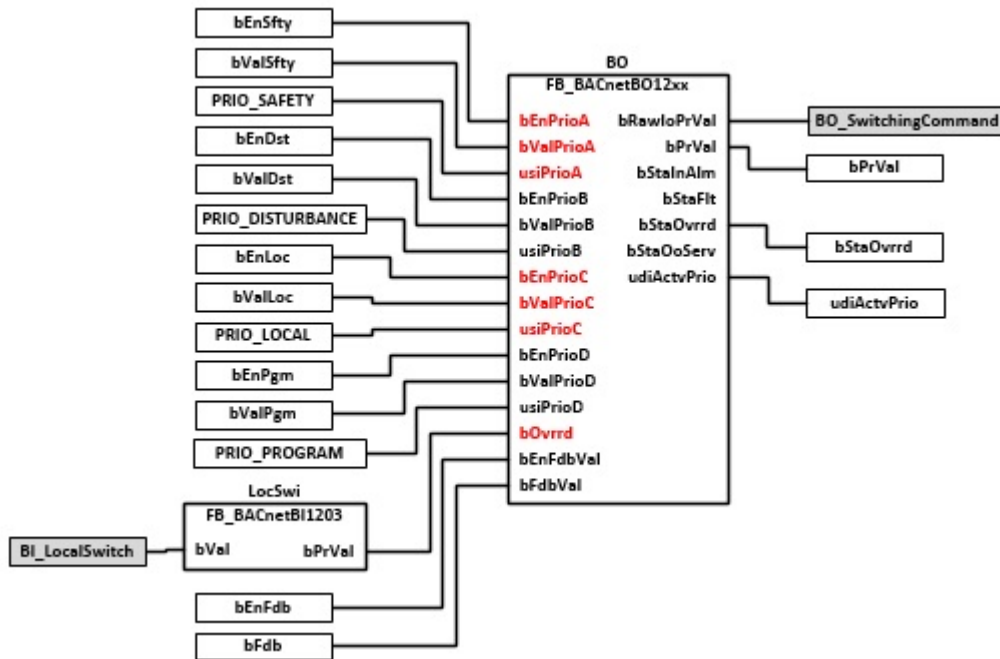
Funktionsbeschreibung

Das Template ermittelt den aktuellen Schaltwert aus mehreren, priorisierten Sollwerten (Prioritätsmatrix, Eingänge des Templates) und übermittelt den Schaltwert an eine Busklemme (siehe **IO-Verknüpfung**). Das Template erfasst zum einen den manuellen Handeingriff mittels eines digitalen Eingangs **LocSwi** und die Rückmeldung des Aktors über die beiden Eingänge **bEnFdb/bFdb** des Templates. Das Template wird eingesetzt um Schaltbefehle oder Impulse auszugeben und zu überwachen (z. B. 1-stufige Motoren/Pumpen, Klappen).

Schnittstelle



Blockschaltbild



VAR_INPUT

```

bEnSfty      : BOOL;
bValSfty     : BOOL;
bEnDst      : BOOL;
bValDst     : BOOL;
bEnLoc      : BOOL;
bValLoc     : BOOL;
bEnPgm      : BOOL;
bValPgm     : BOOL;

```

bEnSfty: Freigabe Sicherheitspriorität

bValSfty: Digitaler Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität.

bValDst: Digitaler Wert Störungspriorität.

bEnLoc: Freigabe Priorität manueller Eingriff

bValLoc: Digitaler Wert Priorität manueller Eingriff

bEnPgm: Freigabe Programmpriorität

bValPgm: Digitaler Wert Programmpriorität

VAR_OUTPUT

```

bPrVal       : BOOL;
udiActvPrio  : UDINT;
bStaOverrrd  : BOOL;

```

bPrVal: Aktueller Wert des Binären-Ausgangs-Objekts.

udiActvPrio: Anzeige der aktuellen Priorität des BO-Objekts

bStaOverrrd: Anzeige, dass das BO-Objekt durch einen lokalen Mechanismus übersteuert wird, siehe **LocSwi**

VAR CONSTANT

```

PLT_NUM      : BYTE := 1;

```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins **FB_BA_Alarm**. [▶ 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates **BAC_PltAlm_01** [▶ 372] mittels des Funktionsbausteins **FB_BA_AlarmPlt**. [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein **FB_BA_ComnMsg** [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe		
BO	FB_BACnetBO1203 [▶ 82]		BO-Objekt für die Ausgabe des Schaltwertes		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty
			PRIO_DISTURBAN CE (3)	Eingang bEnDst	Eingang bValDst
			PRIO_LOCAL (8)	Eingang bEnLoc	Eingang bValLoc
			PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang bValPgm
LocSwi	FB_BACnetBI1203 [▶ 72]		Digitales Eingangs-Objekt zur Anzeige des Schalters für die lokale Übersteuerung des Schaltwertes des BO-Objekts		

IO-Verknüpfung

In der zum Template gehörigen XML-Description sind in dem Bereich **Parameter** Variablen mit der Kennung **INPUT** oder **Output** deklariert. Diese Parameter können im Project Builder oder über die Excel-Import-Schnittstelle mit dem Prozessabbild der Eingangs- und Ausgangsebene in der SPS verknüpft werden.

Parameter	Typ	Instanz	Typ	Prozessabbild	
LocSwi_bVal	USINT	LocSwi	FB_BACnetBI1203 [▶ 72]	Eingang	
BO_bRawIoPrVal	BOOL	BO	FB_BACnetBO1203 [▶ 82]	Ausgang	

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessabbild	
BI_LocalSwitch	BOOL		Eingang	Digitaleingang - Schalter Hand - Meldung - Hand/Auto
BO_SwitchingCommand	BOOL		Ausgang	Digitalausgang - Schaltbefehl - Ein/Aus

Versionshistorie

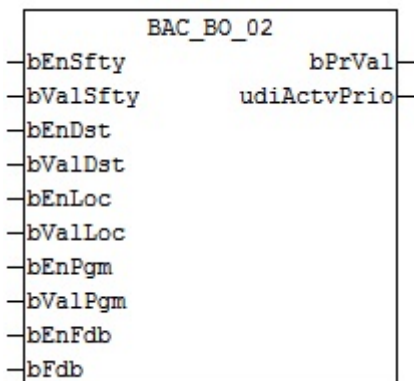
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.16 BAC_BO_02

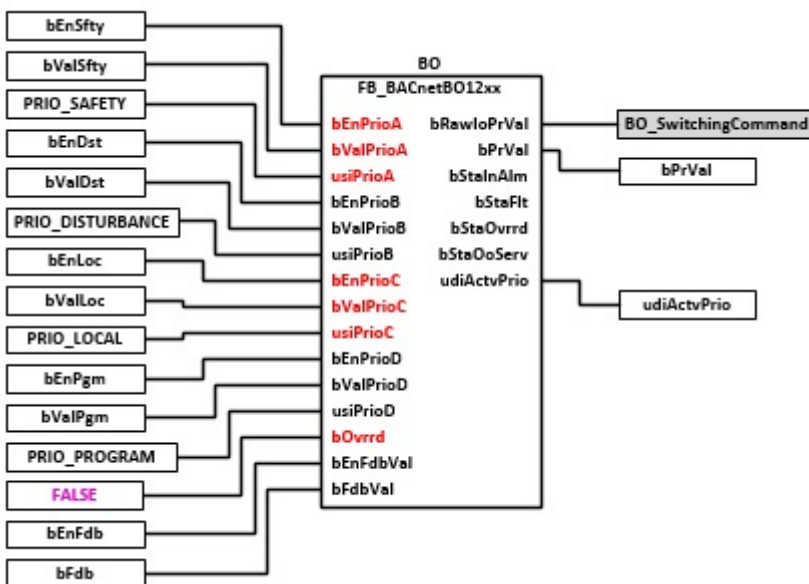
Funktionsbeschreibung

Das Template ermittelt den aktuellen Schaltwert aus mehreren, priorisierten Sollwerten (Prioritätsmatrix, Eingänge des Templates) und übermittelt den Schaltwert an eine Busklemme (siehe **IO-Verknüpfung**). Das Template erfasst die Rückmeldung des Aktors über die beiden Eingänge **bEnFdb/bFdb** des Templates. Das Template wird eingesetzt um Schaltbefehle oder Impulse auszugeben und zu überwachen (z. B. 1-stufige Motoren/Pumpen, Klappen).

Schnittstelle



Blockschaltbild



VAR_INPUT

```

bEnSfty      : BOOL;
bValSfty     : BOOL;
bEnDst       : BOOL;
bValDst      : BOOL;
bEnLoc       : BOOL;
bValLoc      : BOOL;
bEnPgm       : BOOL;
bValPgm      : BOOL;

```

bEnSfty: Freigabe Sicherheitspriorität

bValSfty: Digitaler Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität.

bValDst: Digitaler Wert Störungspriorität.

bEnLoc: Freigabe Priorität manueller Eingriff

bValLoc: Digitaler Wert Priorität manueller Eingriff

bEnPgm: Freigabe Programmpriorität

bValPgm: Digitaler Wert Programmpriorität

VAR_OUTPUT

```
bPrVal      : BOOL;
udiActvPrio : UDINT;
```

bPrVal: Aktueller Wert des Binären-Ausgangs-Objekts.

udiActvPrio: Anzeige der aktuellen Priorität des BO-Objekts

VAR CONSTANT

```
PLT_NUM      : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmergebnisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmergebnisse zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [[182](#)]

Die Auswertung der Alarmergebnisse einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [[372](#)] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmergebnisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe		
BO	FB_BACnetBO1203 [82]		BO-Objekt für die Ausgabe des Schaltwertes		
			Priorität:	Freigabe	Wert
			PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty
			PRIO_DISTURBANC E (3)	Eingang bEnDst	Eingang bValDst
			PRIO_LOCAL (8)	Eingang bEnLoc	Eingang bValLoc
PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang bValPgm			

IO-Verknüpfung

Variablen zur Verknüpfung mit den Klemmen

Parameter	Typ	optional	Prozessab- bild	
BO_SwitchingCommand	BOOL		Ausgang	Digitalausgang - Schaltbefehl - Ein/ Aus

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

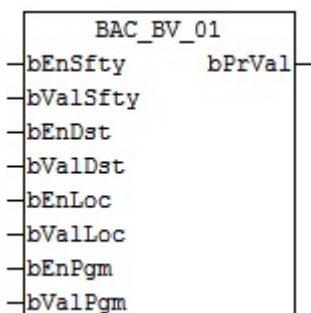
9.81.17 BAC_BV_01

Funktionsbeschreibung

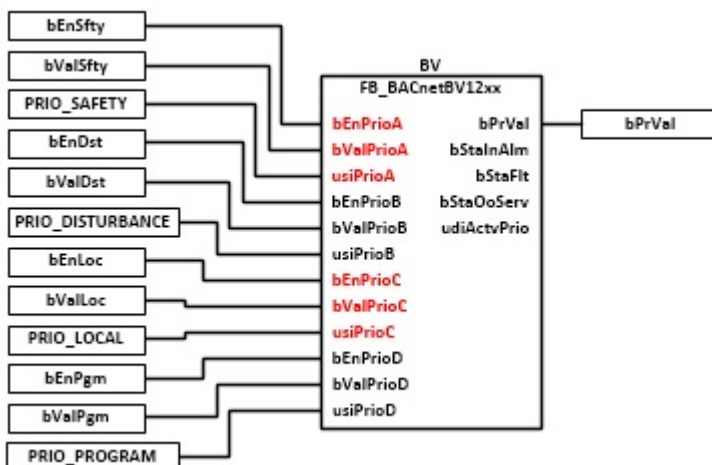
Das Template ermittelt einen Prozesswert aus mehreren, priorisierten Sollwerten (Prioritätsmatrix, Eingänge des Templates).

Das Template wird eingesetzt um einen binären Prozesswert auszugeben oder den Zustand eines Aggregates anzuzeigen.

Schnittstelle



Blockschaltbild



VAR_INPUT

```

bEnSfty      : BOOL;
bValSfty     : BOOL;
bEnDst       : BOOL;
bValDst      : BOOL;
bEnLoc       : BOOL;
bValLoc      : BOOL;
bEnPgm       : BOOL;
bValPgm      : BOOL;

```

bEnSfty: Freigabe Sicherheitspriorität

bValSfty: Digitaler Wert Sicherheitspriorität

bEnDst: Freigabe Störungspriorität.

bValDst: Digitaler Wert Störungspriorität.

bEnLoc: Freigabe Priorität manueller Eingriff

bValLoc: Digitaler Wert Priorität manueller Eingriff

bEnPgm: Freigabe Programmpriorität

bValPgm: Digitaler Wert Programmpriorität

VAR_OUTPUT

```
bPrVal : BOOL;
```

bPrVal: Aktueller Wert des Binären-Wert-Objekts.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm](#). [▶ 182]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [▶ 372] mittels des Funktionsbausteins [FB_BA_AlarmPlt](#). [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe															
BV	FB_BACnetBV1205 [▶ 91]		BV-Objekt für die Ausgabe oder Anzeige eines Prozesswertes															
			<table border="1"> <thead> <tr> <th>Priorität:</th> <th>Freigabe</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td>PRIO_SAFETY (1)</td> <td>Eingang bEnSfty</td> <td>Eingang bValSfty</td> </tr> <tr> <td>PRIO_DISTURBAN CE (3)</td> <td>Eingang bEnDst</td> <td>Eingang bValDst</td> </tr> <tr> <td>PRIO_LOCAL (8)</td> <td>Eingang bEnLoc</td> <td>Eingang bValLoc</td> </tr> <tr> <td>PRIO_PROGRAM (15)</td> <td>Eingang bEnPgm</td> <td>Eingang bValPgm</td> </tr> </tbody> </table>	Priorität:	Freigabe	Wert	PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty	PRIO_DISTURBAN CE (3)	Eingang bEnDst	Eingang bValDst	PRIO_LOCAL (8)	Eingang bEnLoc	Eingang bValLoc	PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang bValPgm
Priorität:	Freigabe	Wert																
PRIO_SAFETY (1)	Eingang bEnSfty	Eingang bValSfty																
PRIO_DISTURBAN CE (3)	Eingang bEnDst	Eingang bValDst																
PRIO_LOCAL (8)	Eingang bEnLoc	Eingang bValLoc																
PRIO_PROGRAM (15)	Eingang bEnPgm	Eingang bValPgm																

Versionshistorie

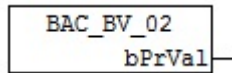
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.18 BAC_BV_02

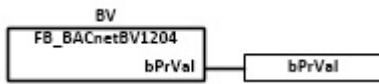
Funktionsbeschreibung

Das Template wird eingesetzt um einen binären Prozesswert auszugeben oder den Zustand eines Aggregates anzuzeigen.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```
bPrVal : BOOL;
```

bPrVal: Aktueller Wert des Binären-Wert-Objekts.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarmer und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarmer zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins `FB_BA_Alarm`. [[182](#)]

Die Auswertung der Alarmer einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates `BAC_PltAlm_01` [[372](#)]

mittels des Funktionsbausteins `FB_BA_AlarmPlt`. [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates `BAC_PltComnMsg_01` durch den Funktionsbaustein `FB_BA_ComnMsg` [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarmer und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
BV	<code>FB_BACnetBV1206</code> [94]		BV-Objekt für die Ausgabe oder Anzeige eines Prozesswertes

Versionshistorie

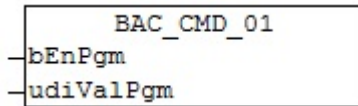
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.19 BAC_CMD_01

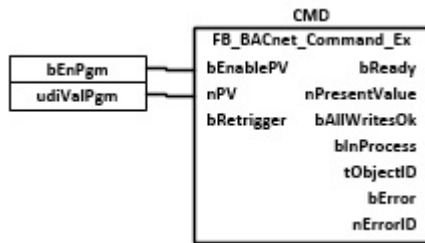
Funktionsbeschreibung

Das Template erfasst die beiden Eingangsvariablen `bEnPgm` / `udiValPgm`. Aufgrund der Zustandsänderung des Eingangswertes `udiValPgm` können über das Gruppenauftrags-Objekt `CMD` spezifizierte Properties anderer Objekte beschrieben werden.

Schnittstelle



Blockschaltbild



VAR_INPUT

bEnPgm : BOOL;
udiValPgm : UDINT;

bEnPgm: Die Eingangsvariable gibt den Eingangswert **udiValPgm** frei, so dass dieser an das Gruppenauftrags-Objekt geschrieben werden kann.

udiValPgm: Eingangswert der an das Gruppenauftrags-Objekt geschrieben wird.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
CMD	FB_BACnet_Command_Ex		Gruppenauftrag-Objekt

Versionshistorie

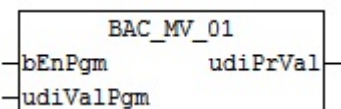
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.20 BAC_MV_01

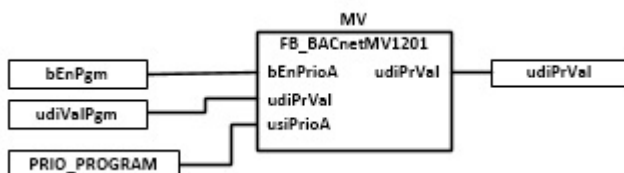
Funktionsbeschreibung

Das Template ermittelt einen Multistate-Prozesswert aus einer einfachen Priorisierung ab. Es wird eingesetzt, um einen Multistate-Wert zu bedienen und zu beobachten.

Schnittstelle



Blockschaltbild



VAR_INPUT

```
bEnPgm      : BOOL;
udiValPgm   : UDINT;
```

bEnPgm: Freigabe Programmpriorität

udiValPgm: Multistate-Wert Programmpriorität

VAR_OUTPUT

```
udiPrVal    : UDINT;
```

udiPrVal: Aktueller Wert des Multistate-Value-Objekts.

VAR CONSTANT

```
PLT_NUM     : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins `FB_BA_Alarm`. [[182](#)]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates `BAC_PltAlm_01` [[372](#)] mittels des Funktionsbausteins `FB_BA_AlarmPlt`. [[186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates `BAC_PltComnMsg_01` durch den Funktionsbaustein `FB_BA_ComnMsg` [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe						
MV	<code>FB_BACnetMV1201</code> [123]		MV-Objekt für die Ausgabe die Multistate-Wertes						
			<table border="1"> <thead> <tr> <th>Priorität:</th> <th>Freigabe</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td><code>PRIO_PROGRAM</code> (15)</td> <td>Eingang <code>bEnPgm</code></td> <td>Eingang <code>udiValPgm</code></td> </tr> </tbody> </table>	Priorität:	Freigabe	Wert	<code>PRIO_PROGRAM</code> (15)	Eingang <code>bEnPgm</code>	Eingang <code>udiValPgm</code>
Priorität:	Freigabe	Wert							
<code>PRIO_PROGRAM</code> (15)	Eingang <code>bEnPgm</code>	Eingang <code>udiValPgm</code>							

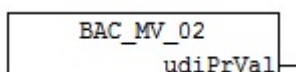
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

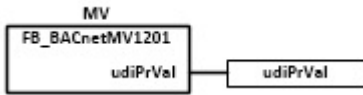
9.81.21 BAC_MV_02**Funktionsbeschreibung**

Das Template bildet einen Multistate-Prozesswert ab.

Es wird eingesetzt, um einen Multistate-Wert zu bedienen und zu beobachten.

Schnittstelle

Blockschaltbild



VAR_OUTPUT

```
udiPrVal : UDINT;
```

udiPrVal: Aktueller Wert des Multistate-Value-Objekts.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt. Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB BA Alarm](#). [[182](#)]
 Die Auswertung der Alarme einer Anlage, z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC PltAlm_01](#) [[372](#)] mittels des Funktionsbausteins [FB BA AlarmPlt](#). [[186](#)]
 Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB BA ComnMsg](#) [[200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
MV	FB_BACnetMV1201 [123]		MV-Objekt für die Ausgabe die Multistate-Wertes

Versionshistorie

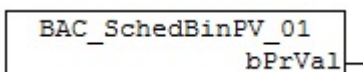
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.81.22 BAC_SchedBinPV_01

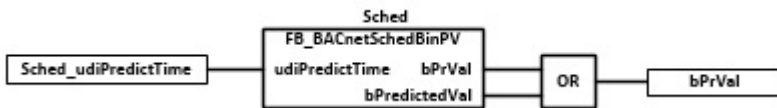
Funktionsbeschreibung

Das Template beinhaltet einen BACnet Scheduler vom Typ Binary Present Value. Die PLC-Ausgangsvariable ist vom Typ BOOL. Zudem enthält der Funktionsbaustein die Funktion "Vorausrechnende Ein- und Ausschaltzeit"
 Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

bPrVal : BOOL;

bPrVal: Ausgabewert des Zeitplans in Abhängigkeit des aktuell eingestellten Datums/Uhrzeit und der Zeiteinträge.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
Sched	FB_BACnetSchedBinPV [▶ 133]		BACnet Scheduler vom Ausgabotyp BOOL

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

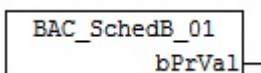
9.81.23 BAC_SchedB_01

Funktionsbeschreibung

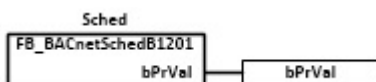
Das Template beinhaltet einen BACnet Scheduler vom Ausgabotyp BOOL.

Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

bPrVal : BOOL;

bPrVal: Ausgabewert des Zeitplans in Abhängigkeit des aktuell eingestellten Datums/Uhrzeit und der Zeiteinträge.

VAR CONSTANT

PLT_NUM : BYTE := 1;

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins FB_BA_Alarm. [▶ 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur

Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [▶ 372]

mittels des Funktionsbausteins [FB_BA_AlarmPlt.](#) [▶ 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates [BAC_PltComnMsg_01](#) durch den Funktionsbaustein [FB_BA_ComnMsg](#) [▶ 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
Sched	FB_BACnetSchedB1201 [▶ 132]		BACnet Scheduler vom Ausgabetyt BOOL

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

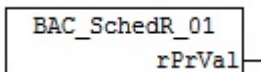
9.81.24 BAC_SchedR_01

Funktionsbeschreibung

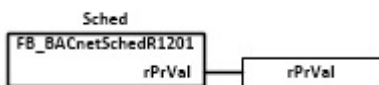
Das Template beinhaltet einen BACnet Scheduler vom Ausgabetyt REAL.

Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```
rPrVal : REAL;
```

rPrVal: Ausgabewert des Zeitplans in Abhängigkeit des aktuell eingestellten Datums/Uhrzeit und der Zeitplaneinträge.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm.](#) [▶ 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [▶ 372]

mittels des Funktionsbausteins [FB_BA_AlarmPlt.](#) [► 186]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB_BA_ComnMsg](#) [► 200].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
Sched	FB_BACnetSchedR1201 [► 134]		BACnet Scheduler vom Ausgabebetyp REAL

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

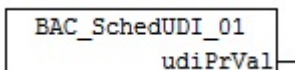
9.81.25 BAC_SchedUdi_01

Funktionsbeschreibung

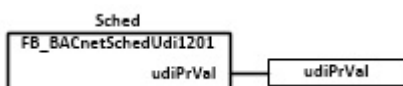
Das Template beinhaltet einen BACnet Scheduler vom Ausgabebetyp UDINT.

Die Parametrierung des Objekts erfolgt dabei entweder aus dem BACnet heraus oder per Kommentarzeilen im PLC-Deklarationsteil.

Schnittstelle



Blockschaltbild



VAR_OUTPUT

```
udiPrVal : UDINT;
```

udiPrVal: Ausgabewert des Zeitplans in Abhängigkeit des aktuell eingestellten Datums/Uhrzeit und der Zeitplaneinträge.

VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer `PLT_NUM` festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm.](#) [► 182]

Die Auswertung der Alarme einer Anlagen z. B. zur Erzeugung einer Sammelmeldung oder zur

Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [► 372]

mittels des Funktionsbausteins [FB_BA_AlarmPlt.](#) [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage, erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB_BA_ComnMsg](#) [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
Sched	FB_BACnetSchedUdi1201 [▶ 136]		BACnet Scheduler vom Ausgabebetyp UDINT

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

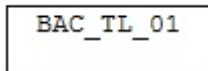
9.81.26 BAC_TL_01

Funktionsbeschreibung

Das Template wird eingesetzt um die Werte einer Datenquelle zu erfassen und im Trendlog-Speicher aufzuzeichnen.

Als Datenquellen gelten alle Integer, Real-, Boolean- und Multistate-Daten, die über eine BACnet Referenz Adresse adressiert werden können.

Schnittstelle



VAR CONSTANT

```
PLT_NUM : BYTE := 1;
```

PLT_NUM: Sämtliche Alarme und Ereignisse aller Anlagen innerhalb eines Controllers werden in einer globalen Alarm- und Ereignisliste erfasst. Die Zuordnung der Ereignisse und Alarme zu einer Anlage wird durch die Vergabe einer Anlagennummer PLT_NUM festgelegt.

Die Erfassung und Verarbeitung eines Alarms von einem Aggregat oder einem Gerät erfolgt innerhalb der Templates mittels des Alarmbausteins [FB_BA_Alarm.](#) [[▶ 182](#)]

Die Auswertung der Alarme einer Anlage, z. B. zur Erzeugung einer Sammelmeldung oder zur Anlagenabschaltung bei relevanten Störungen, erfolgt innerhalb des Templates [BAC_PltAlm_01](#) [[▶ 372](#)] mittels des Funktionsbausteins [FB_BA_AlarmPlt.](#) [[▶ 186](#)]

Die Auswertung verschiedener Anlagenereignisse innerhalb der Templates einer Anlage erfolgt innerhalb des Templates **BAC_PltComnMsg_01** durch den Funktionsbaustein [FB_BA_ComnMsg](#) [[▶ 200](#)].

Wichtig ! Die Zuordnung und Auswertung der Alarme und Ereignisse einer Anlage erfolgt nur dann richtig wenn alle Templates einer Anlage die gleiche Anlagennummer haben!

Die Anlagennummer kann im Projektbuilder im Parametermenü der Templates oder durch eine Spalte innerhalb des Excel-Imports erfolgen.

Programmbeschreibung

Instanz	Typ	optional	Aufgabe
TLog	FB_BACnetTLog1201 [▶ 137]		BACnet-Trend-Log-Objekt

Versionshistorie

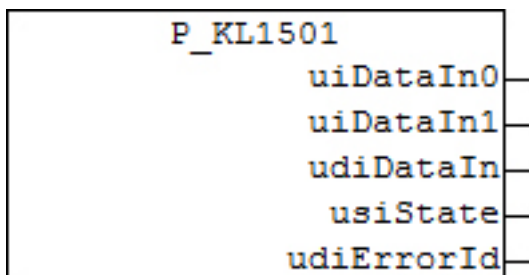
Versionsnummer	Bemerkungen
1.0.0.1	erste Freigabe

9.82 IO

9.82.1 P_KL1501

IO-Template zur Parametrierung einer KL1501: 1-Kanal Auf-Abwärtszähler. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein [FB_KL1501Config](#).

Schnittstelle



VAR_OUTPUT

```
uiDataIn0    : UINT;
uiDataIn1    : UINT;
udiDataIn    : UDINT;
usiState     : USINT;
udiErrorId   : UDINT;
```

uiDataIn0: Entspricht der Datenvariable 0 der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur direkten Verarbeitung im Prozessdatenbetrieb der Klemme.

uiDataIn1: Entspricht der Datenvariable 1 der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur direkten Verarbeitung im Prozessdatenbetrieb der Klemme.

udiDataIn: Diese Variable vom Typ UDINT dient der besseren Auswertung, falls ein 32-bit Zähler angewählt ist. Sie setzt sich aus den beiden o.a. Variablen *uiDataIn0* und *uiDataIn1* zusammen. *uiDataIn0* nimmt dabei den niederwertigen, *uiDataIn1* den höherwertigen Teil ein.

usiState: Entspricht der Statusvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur Statusbeurteilung im Prozessdatenbetrieb der Klemme.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe [Fehlercodes](#).

Parameter

```
iSetCounterType : INT;
```

iSetCounterType: An diesem Eingang ist der Zählertyp einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.

iSetCounterType	Zählertyp
0	32-Bit-Vorwärts/Rückwärts-Zähler
1	2 *16-Bit Vorwärts-Zähler
2	32-bit Gated-Counter, Gate-Eingang Low sperrt den Zähler
3	32-bit Gated-Counter, Gate-Eingang High sperrt den Zähler

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

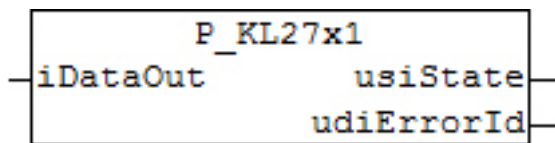
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.2 P_KL27x1

IO-Template zur Parametrierung einer KL2751 / KL2761: 1-Kanal Dimmerklemme. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein [FB_KL27x1Config](#).

Schnittstelle



VAR_INPUT

```
iDataOut : INT;
```

iDataOut: Ausgabedaten an die Klemme, d.h. Lichtwert. Werte kleiner oder gleich Null lassen das Leuchtmittel erlöschen.

VAR_OUTPUT

```
usiState : USINT;
udiErrorId : UDINT;
```

usiState: Entspricht der Statusvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur Statusbeurteilung im Prozessdatenbetrieb der Klemme.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe [Fehlercodes](#).

Parameter

```
bSetDimRampAbsolute : BOOL := FALSE;
iSetRampTime : INT := 3;
bSetWatchdogDisable : BOOL := FALSE;
uiSetWatchdogTimeout : UINT := 3000;
uiSetTimeoutOnValue : UINT := 16383;
uiSetTimeoutOffValue : UINT := 0;
iSetDimmerMode : INT := 0;
bSetOnAfterShortCircuit : BOOL := TRUE;
bSetLineFrequency60Hz : BOOL := FALSE;
```

bSetDimRampAbsolute: FALSE: Die eingestellte Rampenzeit *iSetRampTime* bezieht sich auf den kompletten Datenbereich (0 - 32767). Je kleiner der Sprung, desto kürzer die Rampenzeit. TRUE: Jeder Schaltschritt, egal wie groß, benötigt dieselbe Zeit, die unter *iSetRampTime* eingetragen ist.

iSetRampTime: Eingabe der Rampenzeit. Die Einstellung erfolgt nach unten aufgeführter Tabelle.

bSetWatchdogDisable: Der interne Watchdog wird deaktiviert.

uiSetWatchdogTimeout: Einstellung der Watchdog-Zeit als Vielfaches von 10ms.

uiSetTimeoutOnValue: Dieser Eingang legt den Lichtwert fest, der bei einem Feldbusfehler und aktuellen Prozessdaten > 0 ausgegeben wird. Der Eingabebereich ist auf 32767 begrenzt.

uiSetTimeoutOffValue: Dieser Eingang legt den Lichtwert fest, der bei einem Feldbusfehler und aktuellen Prozessdaten = 0 ausgegeben wird. Der Eingabebereich ist auf 32767 begrenzt.

iSetDimmerMode: An diesem Eingang ist der Dimmermodus einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.

bSetOnAfterShortCircuit: FALSE: Nach einem Kurzschluss bleibt das Licht ausgeschaltet. TRUE: Das Licht wird nach einem Kurzschluss wieder eingeschaltet.

bSetLineFrequency60Hz: FALSE: Netzfrequenz = 50 Hz. TRUE: Netzfrequenz = 60 Hz.

iSetRampTime	Element
0	50 ms
1	100 ms
2	200 ms
3	500 ms
4	1 s
5	2 s
6	5 s
7	10 s
iSetDimmerMode	Element
0	Automatische Erkennung
1	Phasenabschnitt
2	Phasenanschnitt
3	Gleichrichterbetrieb, positiv (positive Halbwelle mit Phasenanschnitt)
4	Gleichrichterbetrieb, negativ (negative Halbwelle mit Phasenanschnitt)

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

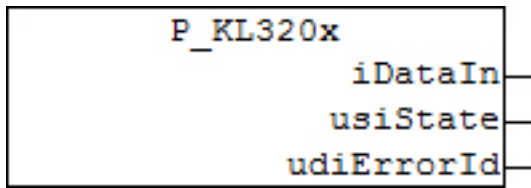
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.3 P_KL320x

IO-Template zur Parametrierung einer KL3201 oder KL3202: Eingangsklemme für Widerstandssensoren. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein [FB_KL320xConfig](#).

Schnittstelle



VAR_OUTPUT

```

iDataIn      : INT;
usiState     : USINT;
udiErrorId   : UDINT;
  
```

iDataIn: Entspricht der Datenvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur direkten Verarbeitung im Prozessdatenbetrieb der Klemme.

usiState: Entspricht der Statusvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur Statusbeurteilung im Prozessdatenbetrieb der Klemme.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe Fehlercodes.

Parameter

```

iSetSensorType : INT;
  
```

iSetSensorType: An diesem Eingang ist der verwendete Sensor einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.

iSetSensorType	Element
0	PT100
1	NI100
2	PT1000
3	PT500
4	PT500
5	NI1000
6	NI120
7	Ausgabe 10,0 Ω - 5000,0 Ω
8	Ausgabe 10,0 Ω - 1200,0 Ω
9	PT1000 - Zwei-Leiter-Anschluss

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

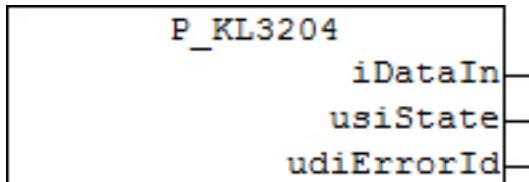
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.4 P_KL3204

IO-Template zur Parametrierung einer KL3204: 4-Kanal Eingangsklemme für Widerstandssensoren. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein [FB_KL320xConfig](#). Dieser Baustein konfiguriert nur einen Kanal der KL3204. Über den Project-Builder wird jedoch die entsprechende Anzahl Bausteine automatisch deklariert.

Schnittstelle



VAR_OUTPUT

```
iDataIn      : INT;
usiState     : USINT;
udiErrorId   : UDINT;
```

iDataIn: Entspricht der Datenvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur direkten Verarbeitung im Prozessdatenbetrieb der Klemme.

usiState: Entspricht der Statusvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur Statusbeurteilung im Prozessdatenbetrieb der Klemme.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe [Fehlercodes](#).

Parameter

```
iSetSensorType : INT;
```

iSetSensorType: An diesem Eingang ist der verwendete Sensor einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.

iSetSensorType	Element
0	PT100
1	NI100
2	PT1000
3	PT500
4	PT500
5	NI1000
6	NI120
7	Ausgabe 10,0 Ω - 5000,0 Ω
8	Ausgabe 10,0 Ω - 1200,0 Ω

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

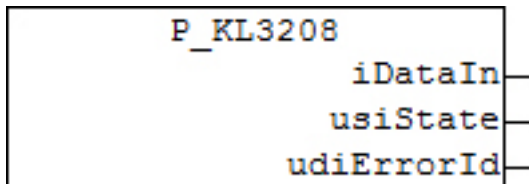
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.5 P_KL3208

IO-Template zur Parametrierung einer KL3208-0010: 8-Kanal Eingangsklemme für Widerstandssensoren. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein FB_KL3208Config. Dieser Baustein konfiguriert nur einen Kanal der KL3208. Über den Project-Builder wird jedoch die entsprechende Anzahl Bausteine automatisch deklariert.

Schnittstelle



VAR_OUTPUT

```

iDataIn      : INT;
usiState     : USINT;
udiErrorId   : UDINT;
  
```

iDataIn: Entspricht der Datenvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur direkten Verarbeitung im Prozessdatenbetrieb der Klemme.

usiState: Entspricht der Statusvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur Statusbeurteilung im Prozessdatenbetrieb der Klemme.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe Fehlercodes.

Parameter

```

iSetSensorType : INT;
  
```

iSetSensorType: An diesem Eingang ist der verwendete Sensor einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.

iSetSensorType	Element
0	PT1000
1	NI1000
2	NI1000 nach Landis&Staefa-Charakteristik: 1000Ω bei 0°C und 1500Ω bei 100°C.
3	NTC1K8
4	NTC1K8_TK
5	NTC2K2
6	NTC3K
7	NTC5K
8	NTC10K
9	NTC10KPRE
10	NTC10K_3204
11	NTC10KTYP2
12	NTC10KTYP3
13	NTC10KDALE

iSetSensorType	Element
14	NTC10K3A221
15	NTC20K
16	Poti, Auflösung 0,1 Ω
17	Poti, Auflösung 1 Ω
18	NTC100K

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

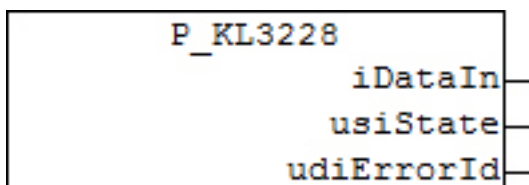
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.6 P_KL3228

IO-Template zur Parametrierung einer KL3228: 8-Kanal Eingangsklemme für Widerstandssensoren. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein [FB_KL3228Config](#). Dieser Baustein konfiguriert nur einen Kanal der KL3228. Über den Project-Builder wird jedoch die entsprechende Anzahl Bausteine automatisch deklariert.

Schnittstelle



VAR_OUTPUT

```
iDataIn      : INT;
usiState     : USINT;
udiErrorId   : UDINT;
```

iDataIn: Entspricht der Datenvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur direkten Verarbeitung im Prozessdatenbetrieb der Klemme.

usiState: Entspricht der Statusvariable der Eingangs-Prozessdaten. Dieser Ausgang eignet sich zur Statusbeurteilung im Prozessdatenbetrieb der Klemme.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe [Fehlercodes](#).

Parameter

```
iSetSensorType : INT;
```

iSetSensorType: An diesem Eingang ist der verwendete Sensor einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.

iSetSensorType	Element
0	PT1000

iSetSensorType	Element
1	NI1000
2	NI1000 nach Landis&Staefa-Charakteristik: 1000Ω bei 0°C und 1500Ω bei 100°C.

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

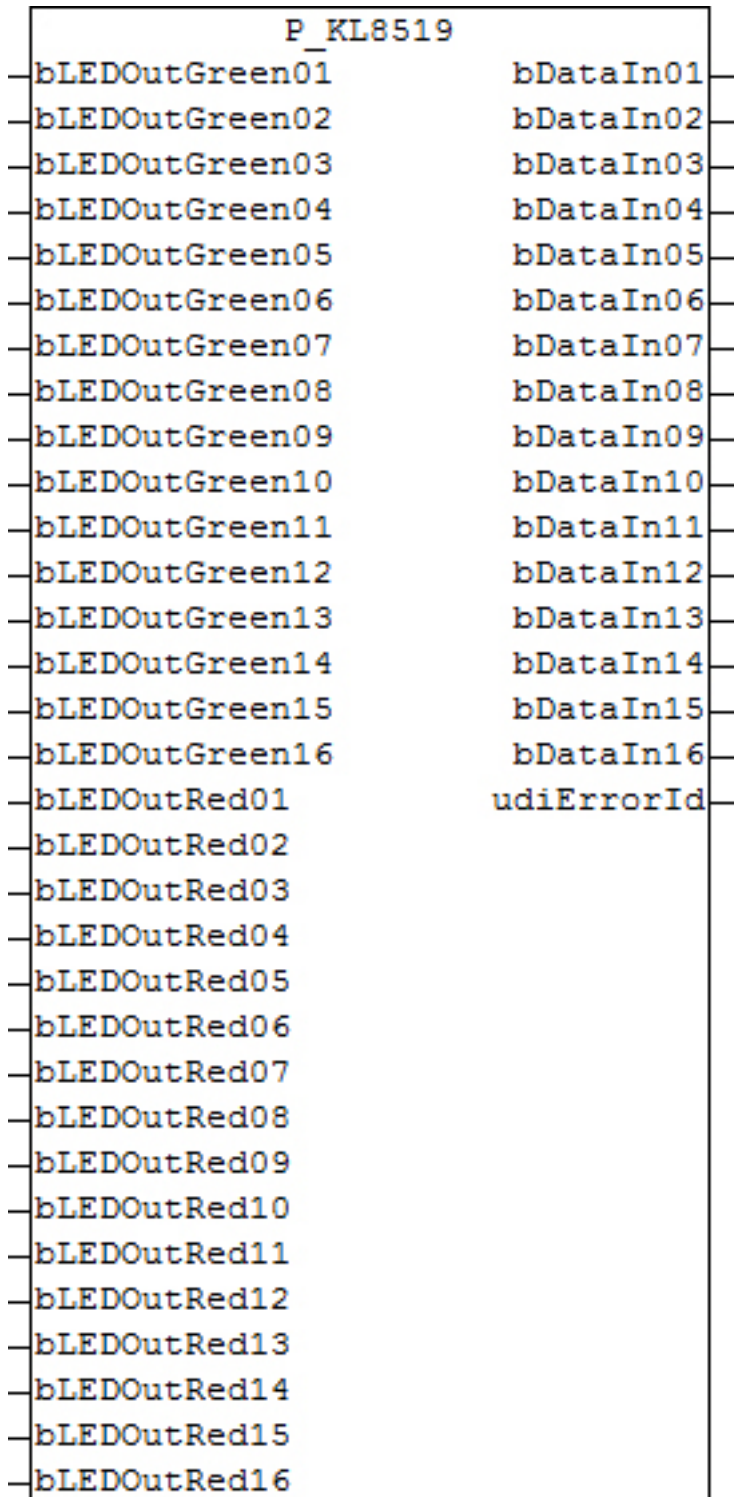
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.7 P_KL8519

IO-Template zur Parametrierung einer KL8519: 16-Kanal-Digital-Eingangs-Meldemodul. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein FB_KL8519.

Schnittstelle



VAR_INPUT

```
bLEDOutGreen01 .. bLEDOutGreen16 : BOOL;
bLEDOutRed01 .. bLEDOutRed16 : BOOL;
```

bLEDOutGreen01 .. bLEDOutGreen16: Schaltet die grüne LED des jeweiligen Kanals ein, wenn das Schalten der LED aus der SPS heraus über den jeweiligen Parameter *bLEDMoDePLC01* bis *bLEDMoDePLC16* freigegeben ist.

bLEDOutRed01 .. bLEDOutRed16: Schaltet die rote LED des jeweiligen Kanals ein, wenn das Schalten der LED aus der SPS heraus über den jeweiligen Parameter *bLEDMoDePLC01* bis *bLEDMoDePLC16* freigegeben ist.

VAR_OUTPUT

```
bDataIn01 .. bDataIn16 : BOOL;
udiErrorId : UDINT;
```

bDataIn01 .. bDataIn16: Dateneingang Kanal 1 bis 16.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe [Fehlercodes](#).

Parameter

```
bLEDMoDePLC01 .. bLEDMoDePLC16 : BOOL;
bLEDCoLOURAuto01 .. bLEDCoLOURAuto16 : BOOL;
```

bLEDMoDePLC01 .. bLEDMoDePLC16: Steht einer dieser Parameter auf TRUE, so ist die Standardfunktion der LEDs für diesen Kanal abgewählt. Das heißt, sie wird nicht mehr durch ein High-Signal des Kanals gesetzt, sondern direkt über die Eingänge *bLEDOuTGreen01* bis *bLEDOuTGreen16* für die grüne Farbgebung und *bLEDOuTRed01* bis *bLEDOuTRed16* für die rote Farbgebung. Ein Setzen beider Farben gleichzeitig ist auch möglich.

bLEDCoLOURAuto01 .. bLEDCoLOURAuto16: Ist der Standardmodus für einen Kanal aktiv (*bLEDMoDePLCxx* = FALSE), so lässt sich mit Hilfe dieser Parameter die Signalfarbe des Kanales anwählen:
bLEDCoLOURAutoxx = FALSE: die grüne LED ist für den Signalstatus "High" angewählt, *bLEDCoLOURAutoxx* = TRUE: die rote LED ist angewählt.

bDUALCoLOURAuto01 .. bDUALCoLOURAuto16: Ist dieser Parameter auf TRUE gesetzt, so befindet sich der jeweilige Kanal im Zweifarb-Betrieb. Ein High-Pegel am Eingang schaltet die unter *bLEDCoLOURAutoxx* angewählte Farbe, ein Low-Zustand die jeweils andere.

bLEDInvertAuto01 .. bLEDInvertAuto16: Ist einer dieser Parameter auf TRUE gesetzt, so kehrt sich das jeweilige Low- und High-Signalverhalten, wie es mit den Parametern *bLEDCoLOURAutoxx* und *bLEDDUALAutoxx* definiert ist, noch einmal um.

iKBusOffMoDe: Mit Hilfe dieses Parameters lässt sich das Verhalten der LEDs im Falle eines K-Bus-Fehlers definieren. Die Einstellung gilt jedoch nur für die LEDs, bei denen der Standardmodus durch *bLEDMoDePLCxx* = TRUE abgewählt wurde. 0: LED bleibt an, sofern sie über die SPS gesetzt wurde. 1: War die LED eingeschaltet, so blinkt sie mit 500ms Taktfrequenz. 2: War die LED eingeschaltet, so blinkt sie mit 1000ms Taktfrequenz. Alle anderen Eingaben werden als "0" interpretiert.

Parametrierfälle am Beispiel Kanal 1 für den Standardmodus *bLEDMoDePLC01* = FALSE:

Zum besseren Verständnis sind hier am Beispiel für den Kanal 1 die Parameterkombinationen für die drei Parameter *bLEDCoLOURAuto01*, *bLEDDUALAuto01* und *bLEDInvertAuto01* mit dem LED-Schaltverhalten aufgeführt. Alle anderen Kanäle können unterschiedlich parametrierbar werden.

Ist das Standardverhalten durch *bLEDMoDePLC01* = TRUE abgewählt, so haben die drei Auto-Parameter keinerlei Einfluss mehr, sondern nur noch die Eingänge *bLEDOuTGreen01* und *bLEDOuTRed01*.

bLEDCoLOURAuto01	bDUALCoLOURAuto01	bLEDInvertAuto01	Signalpegel am Eingang 1	LED-Farbe Kanal 1
FALSE	FALSE	FALSE	Low	AUS
FALSE	FALSE	FALSE	High	grün
FALSE	FALSE	TRUE	Low	grün
FALSE	FALSE	TRUE	High	AUS
FALSE	TRUE	FALSE	Low	rot
FALSE	TRUE	FALSE	High	grün
FALSE	TRUE	TRUE	Low	grün
FALSE	TRUE	TRUE	High	rot
TRUE	FALSE	FALSE	Low	AUS
TRUE	FALSE	FALSE	High	rot
TRUE	FALSE	TRUE	Low	rot
TRUE	FALSE	TRUE	High	AUS

bLEDColourAuto01	bDualColourAuto01	bLEDInvertAuto01	Signalpegel am Eingang 1	LED-Farbe Kanal 1
TRUE	TRUE	FALSE	Low	grün
TRUE	TRUE	FALSE	High	rot
TRUE	TRUE	TRUE	Low	rot
TRUE	TRUE	TRUE	High	grün

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.8 P_KL8524

IO-Template zur Parametrierung einer KL8524: 4 x 2-Kanal-Digital-Ausgangsmodul. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein [FB_KL8524](#).

Schnittstelle

P_KL8524	
usiSwitchModeOut01	bAutoManualChannel01
usiSwitchModeOut02	bAutoManualChannel02
usiSwitchModeOut03	bAutoManualChannel03
usiSwitchModeOut04	bAutoManualChannel04
bLEDOutGreen01_1	usiSwitchMode01
bLEDOutGreen01_2	usiSwitchMode02
bLEDOutGreen02_1	usiSwitchMode03
bLEDOutGreen02_2	usiSwitchMode04
bLEDOutGreen03_1	bStage1Channel01
bLEDOutGreen03_2	bStage2Channel01
bLEDOutGreen04_1	bStage2Channel02
bLEDOutGreen04_2	bStage1Channel02
bLEDOutYellow01_1	bStage1Channel03
bLEDOutYellow01_2	bStage2Channel03
bLEDOutYellow02_1	bStage1Channel04
bLEDOutYellow02_2	bStage2Channel04
bLEDOutYellow03_1	udiErrorId
bLEDOutYellow03_2	
bLEDOutYellow04_1	
bLEDOutYellow04_2	

VAR_INPUT

```
usiSwitchModeOut01 .. usiSwitchModeOut04 : USINT;
bLEDOutGreen01_1 .. bLEDOutGreen04_2 : BOOL;
bLEDOutYellow01_1 .. bLEDOutYellow04_2 : BOOL;
```

usiSwitchModeOut01 .. usiSwitchModeOut04: Ansteuerung Stufen Kanal 1 bis 4. Zulässige Werte: "0", "1" und "2". Alle anderen Eingaben werden als "0" interpretiert.

bLEDOutGreen01_1 .. bLEDOutGreen04_2: Ansteuerung der grünen LEDs Kanal 1 bis 4, Stufe 1 und 2, wenn das Schalten der LED aus der SPS heraus über den jeweiligen Parameter *bLEDMoDePLC01_1* bis *bLEDMoDePLC04_2* freigegeben ist.

bLEDOutYellow01_1 .. bLEDOutYellow04_2: Ansteuerung der gelben LEDs Kanal 1 bis 4, Stufe 1 und 2, wenn das Schalten der LED aus der SPS heraus über den jeweiligen Parameter *bLEDMoDePLC01_1* bis *bLEDMoDePLC04_2* freigegeben ist.

VAR_OUTPUT

```
bAutoManualChannel01 .. bAutoManualChannel04 : BOOL;
usiSwitchMode01 .. usiSwitchMode04 : USINT;
usiOnOffChannel01 .. usiOnOffChannel04 : USINT;
udiErrorId : UDINT;
```

bAutoManualChannel01 .. bAutoManualChannel04: Status Auto-Hand-Wahlschalter Kanal 1 bis 4: FALSE: Schalter steht auf "man", TRUE: Schalter steht auf "auto".

usiSwitchMode01 .. usiSwitchMode04: Numerischer Status (0, 1, 2) des Dreistufenschalters Kanal 1 bis 4.

usiOnOffChannel01 .. usiOnOffChannel04: Tatsächlich geschaltete Stufe Kanal 1 bis 4: Im Handbetrieb (Wahlschalter = man) die des Dreistufenschalters, im Automatikbetrieb (Wahlschalter = auto) die des jeweiligen Eingangs *usiSwitchmodeOut01* bis *usiSwitchmodeOut04*.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe Fehlercodes.

Parameter

```
bLEDMoDePLC01_1 .. bLEDMoDePLC04_2 : BOOL;
iKBusOffMoDeChannel01 .. iKBusOffMoDeChannel04 : INT;
bOutMoDeChannel01 .. bOutMoDeChannel04 : BOOL;
usiSwitchDelayChannel01 .. usiSwitchDelayChannel04 : USINT;
```

bLEDMoDePLC01_1 .. bLEDMoDePLC04_2: Steht einer dieser Parameter auf TRUE, so ist die Standardfunktion für die jeweilige LED abgewählt. Das heißt, sie wird nicht mehr durch die jeweilige Signalstufe beeinflusst, sondern direkt über die Eingänge *bLEDOutGreen01_1* bis *bLEDOutGreen4_2* für die grüne Farbgebung und *bLEDOutYellow01* bis *bLEDOutYellow04_2* für die gelbe Farbgebung. Ein Setzen beider Farben gleichzeitig ist auch möglich. Steht der Parameter hingegen auf FALSE, so gilt: Stufe nicht aktiv = gelb, Stufe aktiv = grün.

iKBusOffMoDeChannel01 .. iKBusOffMoDeChannel04: Anwahl des Verhaltens bei K-Bus-Fehler: 0: Kein Ausgang wird gesetzt, 1: Stufe 1 wird gesetzt, 2: Stufe 2 wird gesetzt. Alle anderen Eingaben werden als "0" interpretiert.

bOutMoDeChannel01 .. bOutMoDeChannel04: Ausgabemodus Kanal 1 bis Kanal 4: FALSE: Ausgabemodus 1, TRUE: Ausgabemodus 2, siehe Anleitung [FB_KL8524](#).

usiSwitchDelayChannel01 .. usiSwitchDelayChannel04: Eingabe der Schaltverzögerung für Stufe 2 für den jeweiligen Kanal als Vielfaches von 10ms.

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

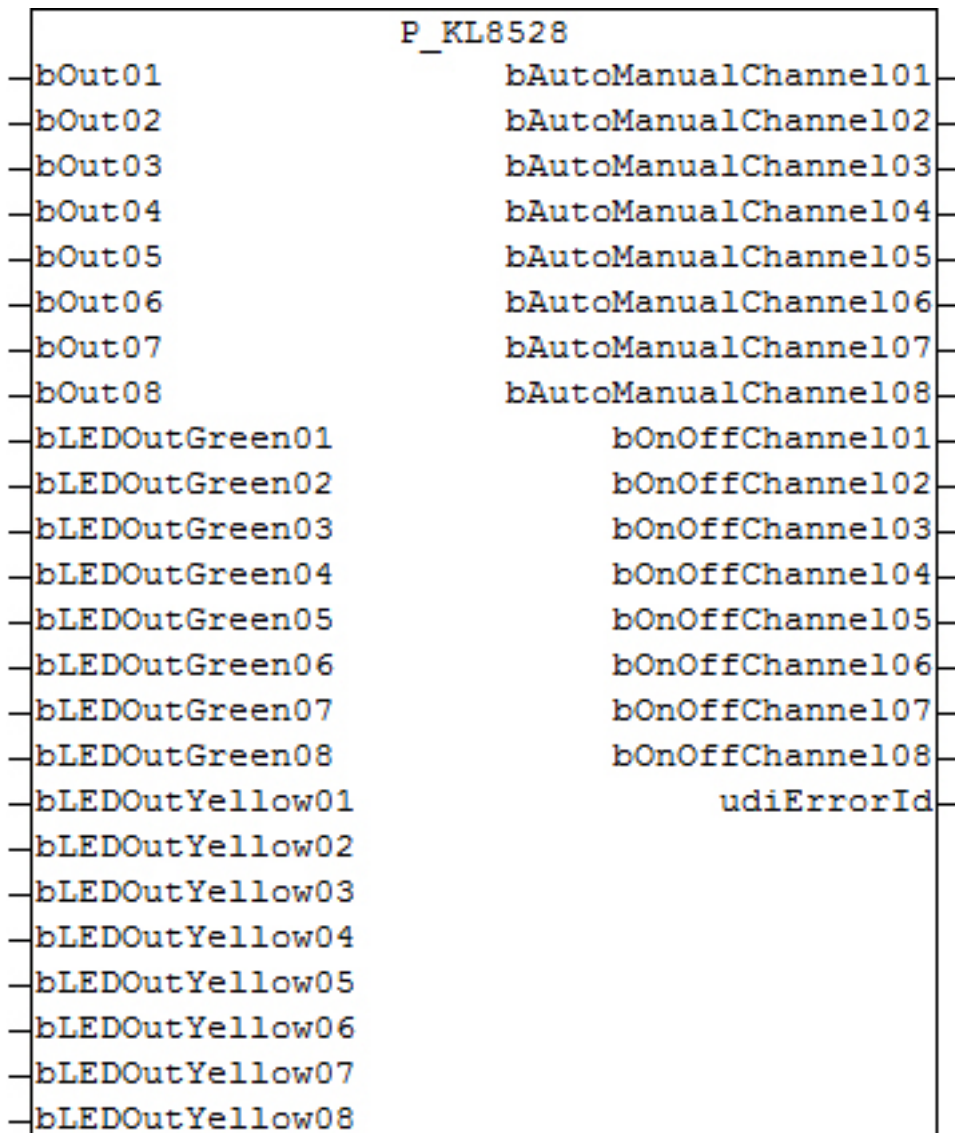
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.9 P_KL8528

IO-Template zur Parametrierung einer KL8528: 8-Kanal-Digital-Ausgangsmodul. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein [FB_KL8528](#).

Schnittstelle



VAR_INPUT

```
bOut01 .. bOut08           : BOOL;
bLEDOutGreen01 .. bLEDOutGreen08 : BOOL;
bLEDOutYellow01 .. bLEDOutYellow08 : BOOL;
```

bOut01 .. bOut04: Ansteuerung Ausgänge Kanal 1 bis 8.

bLEDOutGreen01 .. bLEDOutGreen08: Ansteuerung der grünen LEDs Kanal 1 bis 8, wenn das Schalten der LED aus der SPS heraus über den jeweiligen Parameter *bLEDMoDePLC01* bis *bLEDMoDePLC08* freigegeben ist.

bLEDOutYellow01 .. bLEDOutYellow08: Ansteuerung der gelben LEDs Kanal 1 bis 8, wenn das Schalten der LED aus der SPS heraus über den jeweiligen Parameter *bLEDMoDePLC01* bis *bLEDMoDePLC08* freigegeben ist.

VAR_OUTPUT

```
bAutoManualChannel01 .. bAutoManualChannel04 : BOOL;
bOnOffChannel01 .. bOnOffChannel04 : BOOL;
udiErrorId : UDINT;
```

bAutoManualChannel01 .. bAutoManualChannel08: Status Auto-Hand-Wahlschalter Kanal 1 bis 8: FALSE: Schalter steht nicht auf "auto", TRUE: Schalter steht auf "auto".

bOnOffChannel01 .. bOnOffChannel08: Status Auto-Hand-Wahlschalter Kanal 1 bis 8: FALSE: Schalter steht nicht auf "on", TRUE: Schalter steht auf "on".

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe [Fehlercodes](#).

Parameter

```
bLEDMoDePLC01 .. bLEDMoDePLC08 : BOOL;
bKBusOffMoDeChannel01 .. bKBusOffMoDeChannel08 : BOOL;
```

bLEDMoDePLC01 .. bLEDMoDePLC08: Steht einer dieser Parameter auf TRUE, so ist die Standardfunktion für die jeweilige LED abgewählt. Das heißt, sie wird nicht mehr durch die jeweilige Signalstufe beeinflusst, sondern direkt über die Eingänge *bLEDOutGreen01* bis *bLEDOutGreen08* für die grüne Farbgebung und *bLEDOutYellow01* bis *bLEDOutYellow08* für die gelbe Farbgebung. Ein Setzen beider Farben gleichzeitig ist auch möglich. Steht der Parameter hingegen auf FALSE, so gilt: Stufe nicht aktiv = gelb, Stufe aktiv = grün.

bKBusOffMoDeChannel01 .. bKBusOffMoDeChannel08: Ausgangsverhalten des Verhaltens bei K-Bus-Fehler. FALSE: Ausgang nicht gesetzt, TRUE: Ausgang gesetzt, falls Schalterstellung = "auto".

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

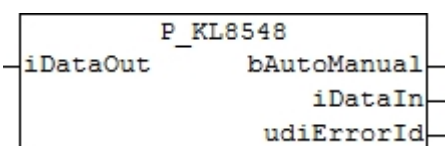
Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

9.82.10 P_KL8548

IO-Template zur Parametrierung einer KL8548: 8-Kanal-Analog-Ausgangsmodul 0...10 V. Dieses Template konfiguriert bei SPS-Neustart die Klemme mit den im Project-Builder eingegebenen Parametern und schaltet danach auf den Prozessdatenbetrieb um. Grundlage dieses Templates ist der Baustein [FB_KL8548](#). Dieser Baustein konfiguriert nur einen Kanal der KL8548. Über den Project-Builder wird jedoch die entsprechende Anzahl Bausteine automatisch deklariert.

Schnittstelle



VAR_INPUT

```
iDataOut      : INT;
```

iDataOut: Analoger Ausgabewert.

VAR_OUTPUT

```
bAutoManual   : BOOL;
iDataIn       : INT;
udiErrorId    : UDINT;
```

bAutoManual: Stellung des Schalters: FALSE = "man". TRUE = "auto".

iDataIn: Stellung des Bedienpotentiometers.

udiErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Siehe [Fehlercodes](#).

Parameter

```
bBarGraphEcoMode : BOOL;
iKBusOffMode     : INT;
iKBusOffValue    : INT;
```

bBarGraphEcoMode: Anzeigemodus des Bargraphs: FALSE: Fullscale-Mode (ausgefüllter Balken), TRUE: ECO-Mode (nur ein Punkt).

iKBusOffMode: Verhalten des Klemmausgangs bei K-Bus Fehler: 0: Ausgang wird 0, 1: Ausgang behält den letzten Wert bei, bei PLC-Stopp jedoch wird er 0, 2: Ausgang nimmt den Wert *iKBusOffValue* an, falls Schalterstellung = "auto". Alle anderen Eingaben werden als "0" interpretiert.

iKBusOffValue: Wert, den der Ausgang bei K-Bus Fehler (oder PLC gestoppt) im Modus 2 annehmen soll.

Entwicklungsinformationen

Entwicklungsumgebung	BACnet Revision	Zielsystem	erforderliches Supplement
TwinCAT 2.11 R3/x64 ab Build 2254	n/a	PC/CX	TS8040 TwinCAT Building Automation ab V1.1.0

Versionshistorie

Versionsnummer	Bemerkungen
1.0.0.0	erste Freigabe

10 TwinCAT BA Project Builder

Einleitung

Grundlage für eine wartbare Anlage ist die durchgängige Strukturierung der TwinCAT-Projektdateien. Zentraler Gedanke hierbei ist das Anlagenkennzeichnungssystem (AKS), das jedem Datenpunkt und jedem Programmbaustein einen Namen nach fest vorgegebenen Regeln zuordnet.

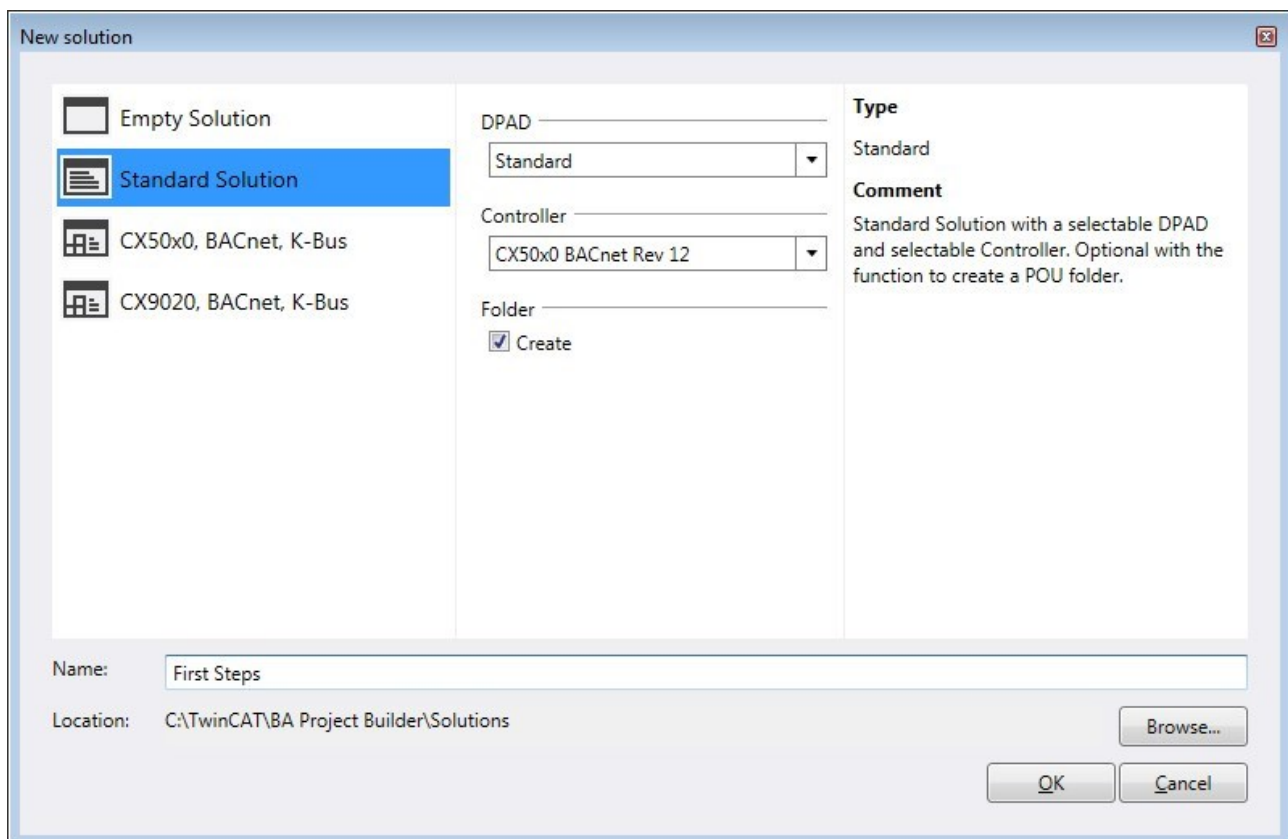
Der TwinCAT BA Project Builder generiert automatisch für alle Beckhoff-Controller die Projektdateien für TwinCAT PLC Control und TwinCAT System Manager. Er kann durch benutzerspezifische BA PLC Templates erweitert werden und bietet somit volle Flexibilität beim Engineering. AddIns bieten die Möglichkeit der Integration von Planungsunterlagen, wie z. B. Excel-Listen sowie die Anbindung an Visualisierungssysteme.

10.1 Erste Schritte

Im Folgenden werden die notwendigen Schritte aufgezeigt, die notwendig sind um eine Solution mit dem TwinCAT BA Project Builder zu erstellen.

Projekt erstellen

Rufen Sie über das Menü *Solution -> New* den Dialog zum Erstellen einer neuen Solution auf.

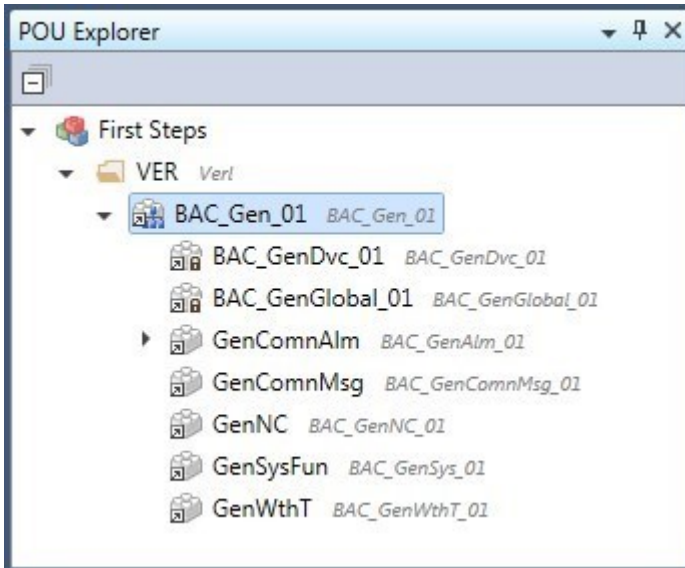


In diesem Dialog können Sie den Namen der Solution und den Pfad, in der diese abgelegt werden soll, angeben. Des Weiteren muss das Anlagenkennzeichnungssystem (engl. **D**ata **P**oint **A**ddressing **D**escription) ausgewählt werden, welches als Grundlage für die Projektierung der Solution genutzt wird. Der BA Project Builder bietet die Möglichkeit ein leeres Anlagenkennzeichnungssystem (AKS) auszuwählen und das AKS selber festzulegen.

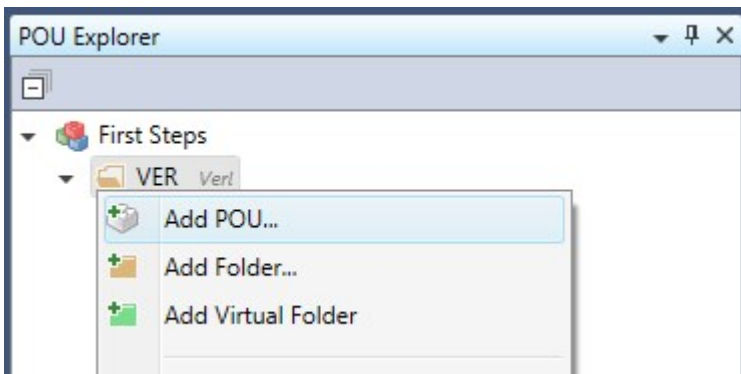
Templates hinzufügen

Der *POU Explorer* definiert die softwareseitige Anlagenstruktur. Ordner im *POU Explorer* legen die einzelnen Ebenen (DPAD Level) fest. Die möglichen Bezeichner (DPAD Key) einer Ebene werden durch das AKS vorgegeben. Über das Tool Window *Properties* kann der gewünschte DPAD Key ausgewählt werden.

Öffnen Sie das Tool Window *Templates* und ziehen Sie per Drag & Drop das Template *BAC_Gen_01* auf den Ordner im *POU Explorer*. Hierbei handelt es sich um ein Aufruftemplate, welches die grundlegenden SPS-Bausteine für den BACnet-Server in das SPS-Programm einfügt. Dieses Aufruftemplate ist mit weiteren Templates verknüpft, welche automatisch dem SPS-Programm hinzugefügt werden. Im *POU Explorer* werden alle POU's angezeigt, die dem SPS-Projekt hinzugefügt wurden.



Alternativ können sie auch über das Kontextmenü auf dem Ordner einen POU hinzufügen:

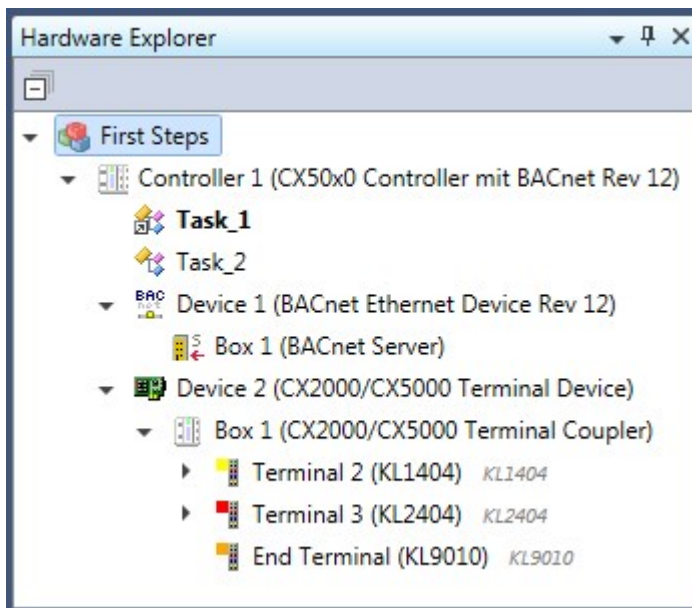


Durch einen Doppelklick auf einen POU öffnet sich ein Dialog, über den weitere Parameter oder BACnet-Properties editiert werden können.

Hardware hinzufügen

Sämtliche Controller, deren Geräte und Klemmen werden im *Hardware Explorer* zusammengestellt. Beim Anlegen der neuen Solution wurden automatisch ein CX50x0 mit den notwendigen SPS-Tasks, BACnet-Server und E/A-Geräten angelegt.

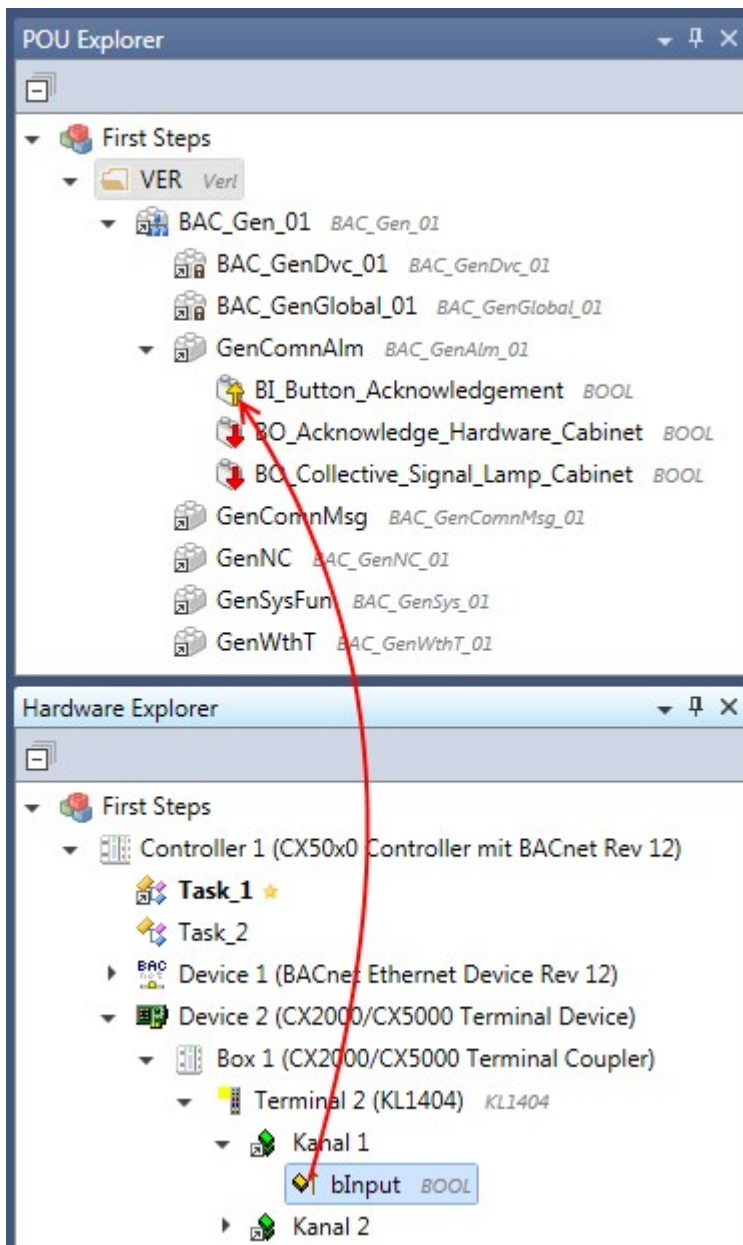
Unterhalb von *Device 2* befindet sich der Terminal Coupler *Box 1*. Jetzt ziehen sie per Drag & Drop aus dem Tool Window *Terminals* die Klemmen *KL1404* und *KL2404* zur *Box 1*. Der *Hardware Explorer* sollte anschließend wie folgt aussehen:



Auch hier können sie alternativ per Kontextmenü die Klemmen hinzufügen.

POUs mit Klemmen verknüpfen

Verfügen POUs über Ein- und Ausgänge, können diese mit Ein- und Ausgängen von Busklemmen verknüpft werden.

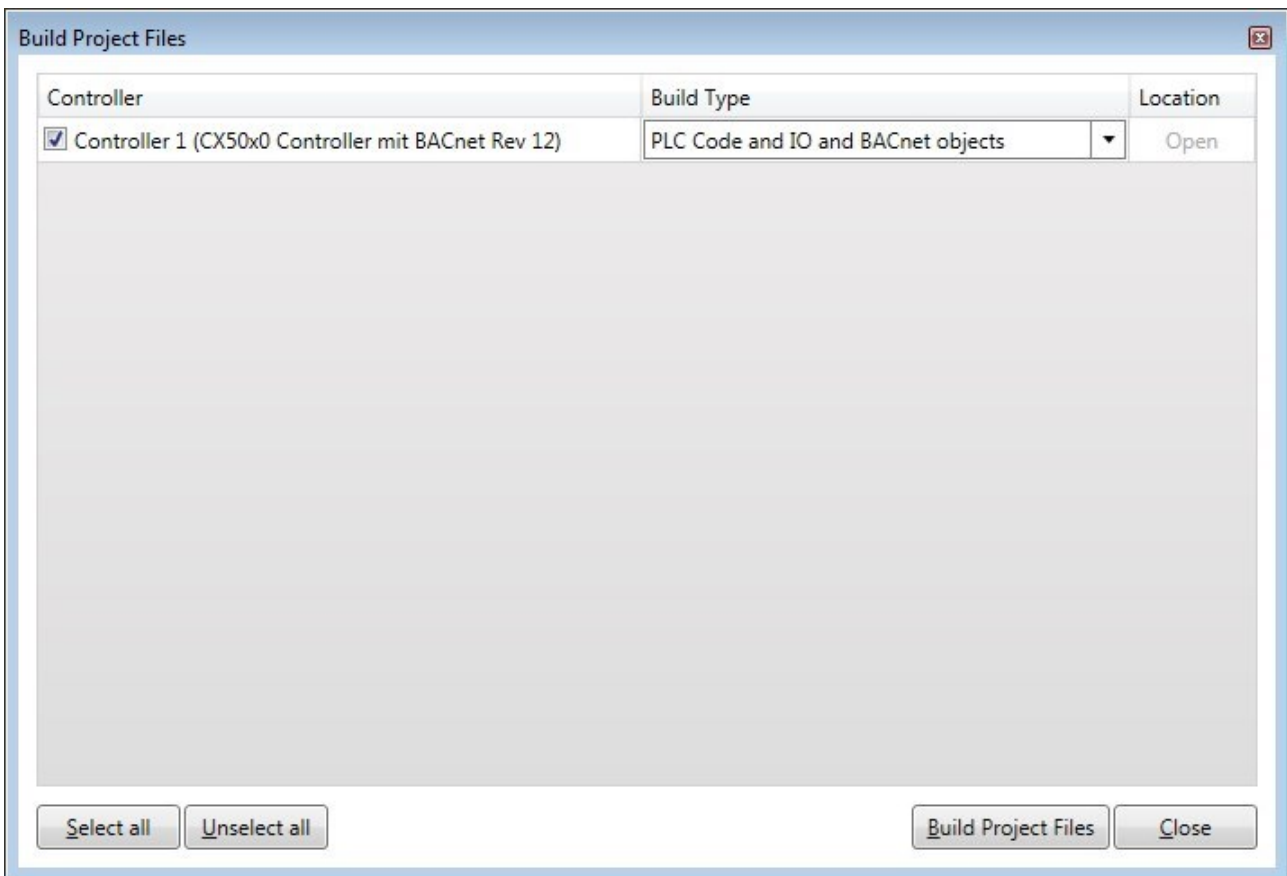


Die Verknüpfungen können per Drag & Drop oder über das Kontextmenü erstellt werden.

TwinCAT Projektdateien erzeugen

Wurden alle POUs angelegt, die Parameter und BACnet-Properties wie gewünscht eingestellt und mit der Hardware verknüpft, so können die TwinCAT Projektdateien generiert werden. Rufen Sie das Menü *Action* -> *Build Project Files* auf.

Wählen Sie die Controller aus, für die Sie die TwinCAT Projektdateien erzeugen wollen.



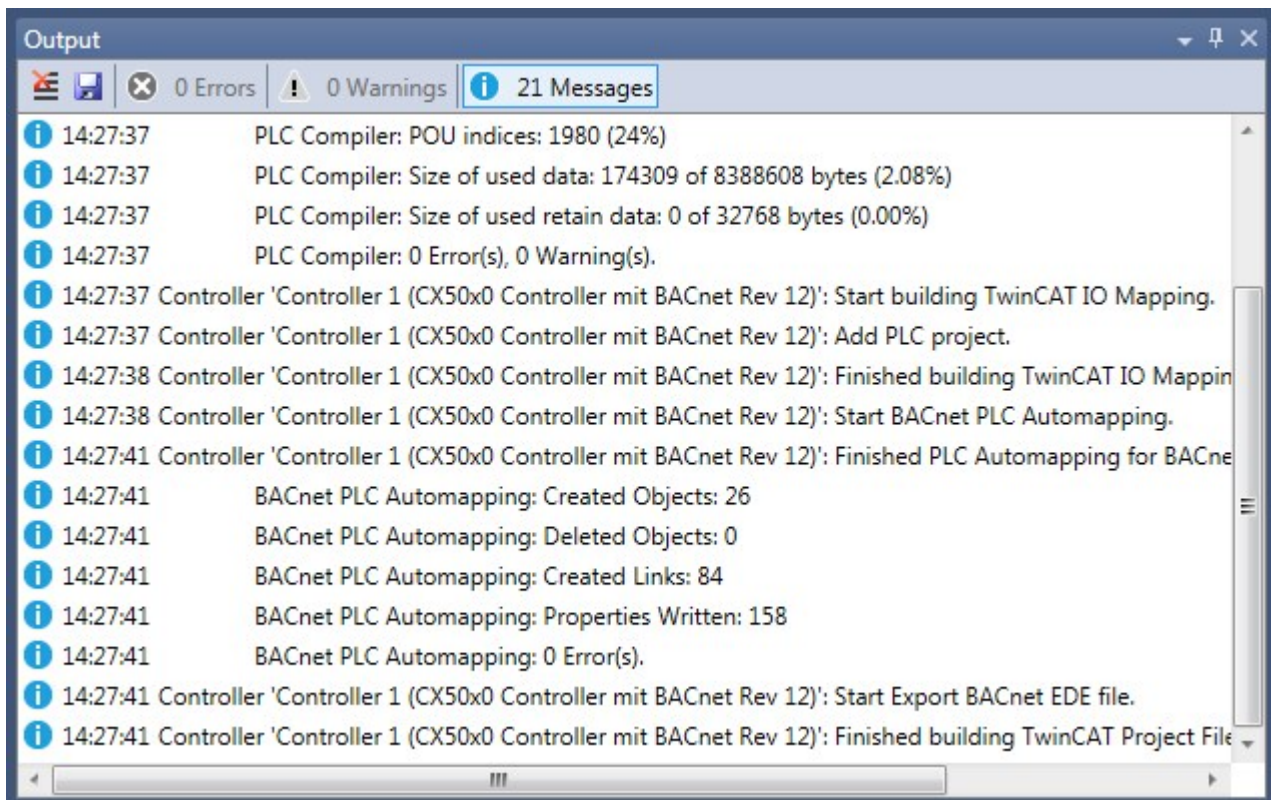
Die Spalte *Build Type* gibt an, welche Projektdateien angelegt werden. Dieses hat erheblichen Einfluss auf die Dauer der Generierung.

Voraussetzungen

PLC Code and IO and BACnet objects	Es wird die pro-Datei für das TwinCAT PLC Control erzeugt und compiliert. Des Weiteren wird die tsm-Datei für den TwinCAT System Manager mit allen E/A-Geräten, BACnet-Objekten und Verknüpfungen angelegt.
PLC Code and IO (without BACnet objects)	Es wird die pro-Datei für das TwinCAT PLC Control erzeugt und compiliert. Des Weiteren wird die tsm-Datei für den TwinCAT System Manager mit allen E/A-Geräten und Verknüpfungen angelegt. Es werden keine BACnet-Objekte hinzugefügt.
PLC Code (only pro file, no tpy file)	Es wird nur die pro-Datei für das TwinCAT PLC Control erzeugt. Das Program wird nicht compiliert.

Nach dem Starten der Generierung werden im Tool Window *Output* diverse Statusmeldungen ausgegeben.

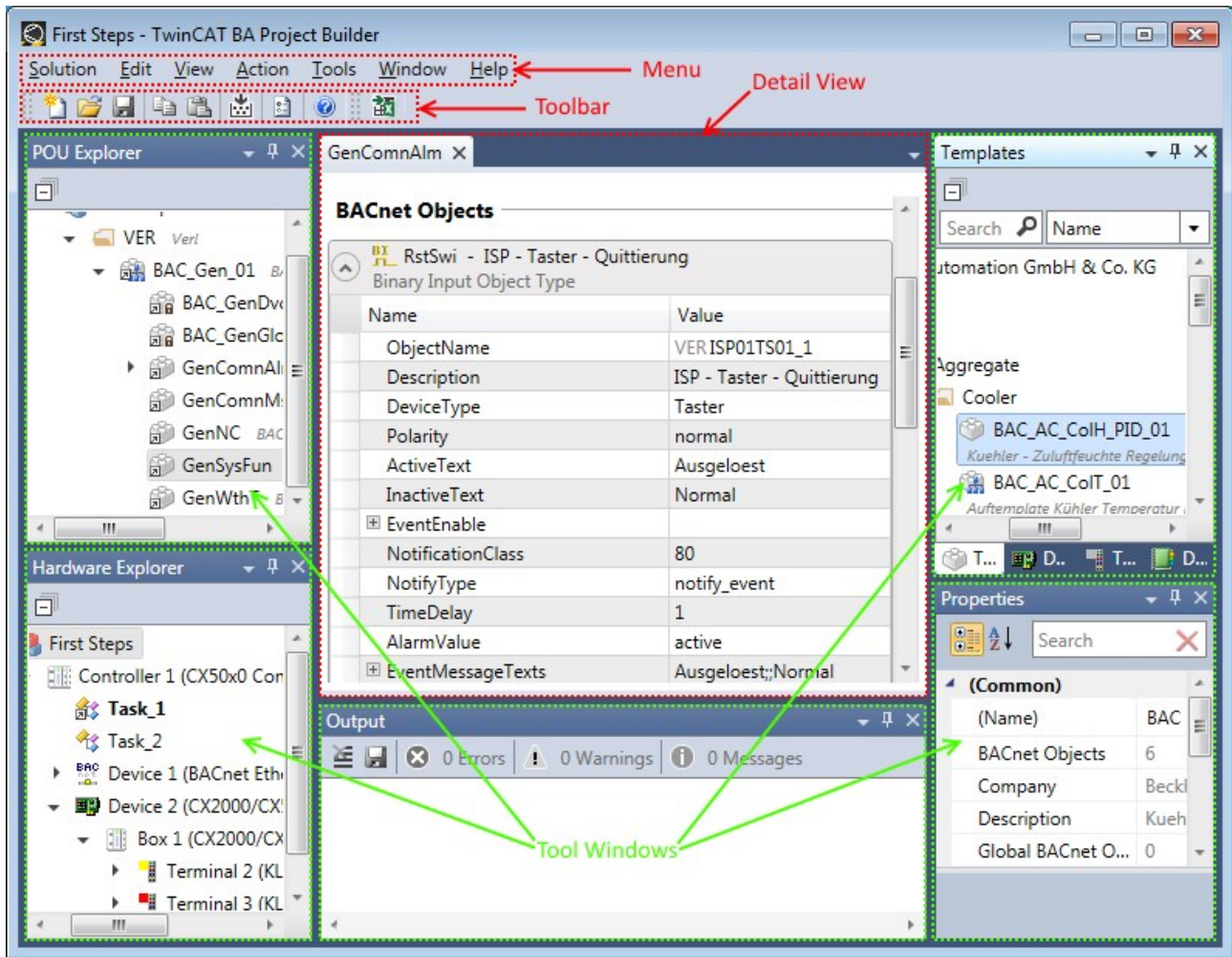
Diese Meldungen geben darüber Auskunft, ob alle Projektdateien erfolgreich erzeugt wurden oder ob Fehler aufgetreten sind.



In dem Ordner, in dem auch die Solution gespeichert wurde, wird für jeden Controller ein Verzeichnis angelegt. Dieses Verzeichnis hat den gleichen Namen wie der Controller im BA Project Builder und besitzt das Unterverzeichnis *~TwinCAT*. Dort befinden sich die generierten TwinCAT Projektdateien.

10.2 Hauptfenster

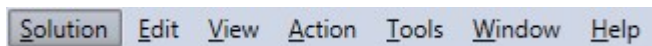
Der BA Project Builder teilt sich in mehrere Bereiche und Fenster auf.



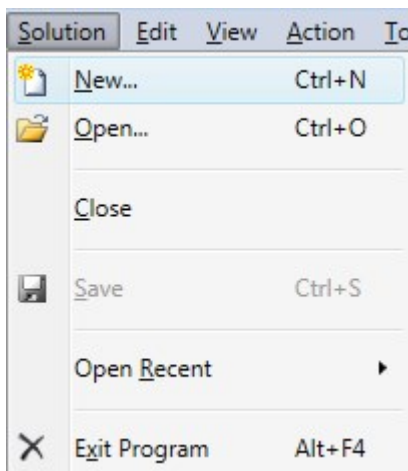
Menü

Über das Menü können die einzelnen Aktionen und Befehle ausgeführt werden. Jeder Menüpunkt kann auch über die Tastatur erreicht werden.

Nach dem Drücken der Alt-Taste ist bei jedem Menüpunkt ein Buchstabe unterstrichen.



Drücken Sie jetzt den unterstrichenden Buchstaben auf der Tastatur.



In dem Menü sind wiederum Buchstaben unterstrichen, mit dem Sie den Befehl ausführen können.

Um den Befehl *Close* im Menü *Solution* auszuführen, müssen Sie die Tasten <Alt> dann 's' und 'c' einzeln drücken.

Einige wichtige Menüpunkte sind auch über Shortcuts direkt erreichbar. Der entsprechende Shortcut, wenn verfügbar, steht im Menü hinter den jeweiligen Befehlen.

Um den Befehl *Open* per Shortcut aufzurufen, müssen Sie die Taste <Ctrl> gedrückt halten und dann die Taste 'O' gleichzeitig drücken.

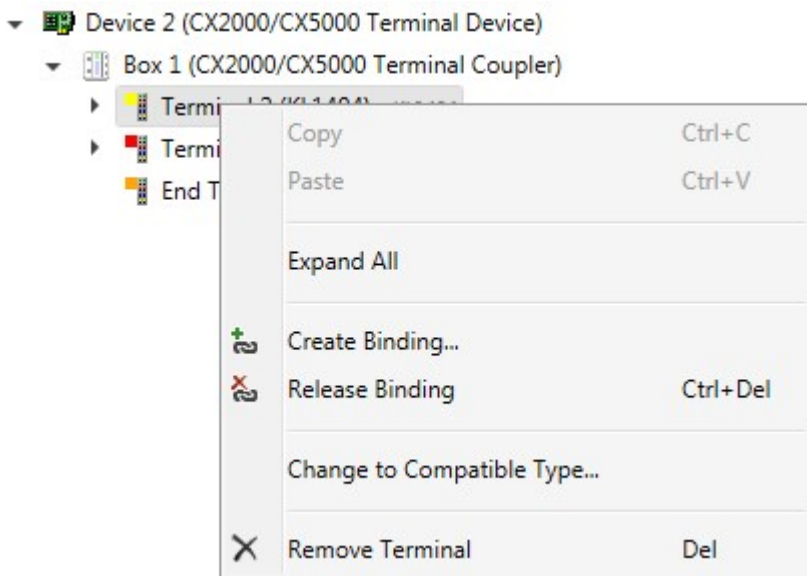
Toolbar

Wichtige Befehle sind über die Toolbar direkt erreichbar.



Kontextmenüs

Einige Elemente in den Tool Windows bieten Kontextmenüs an. Über diese lassen sich bestimmte Aktionen, wie z.B. Hinzufügen oder Löschen, direkt ausführen.



Detail View

In der Mitte des BA Project Builders befindet sich der *Detail View*. Durch einen Doppelklick auf bestimmte Elemente in den Tool Windows, werden in diesem Bereich weitere Dialoge geöffnet.

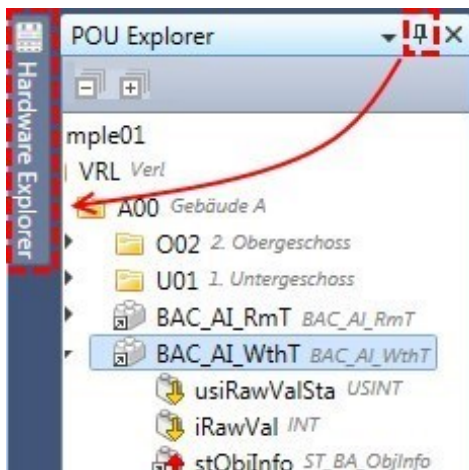
Tool Windows

Die Tool Windows gruppieren sich um den *Detail View*. Ein Tool Window ist ein eigenständiges Fenster, welches frei verschoben oder auch ganz ausgeblendet werden kann. Neben der Bedienung mit der Maus, bieten auch die Menüs *View* und *Window* entsprechende Befehle an.

Position anpassen

Die Tool Windows lassen sich frei in ihrer Größe und Position verschieben.

Klicken Sie hierzu mit der linken Maustaste auf den oberen Bereich eines Tool Windows. Falls mehrere Tool Windows in einem Bereich übereinander liegen, müssen Sie den entsprechenden Tab anklicken.



Ein Tool Window kann durch das X im oberen Bereich geschlossen werden. Über den Menüpunkt *View* können geschlossene Tool Windows wieder geöffnet werden. Alle Tool Windows lassen sich durch den Befehl *Window -> Close All* gleichzeitig schließen.

Der Befehl *Window -> Reset Window Layout* setzt alle Fenster wieder auf ihre Standard Positionen zurück.

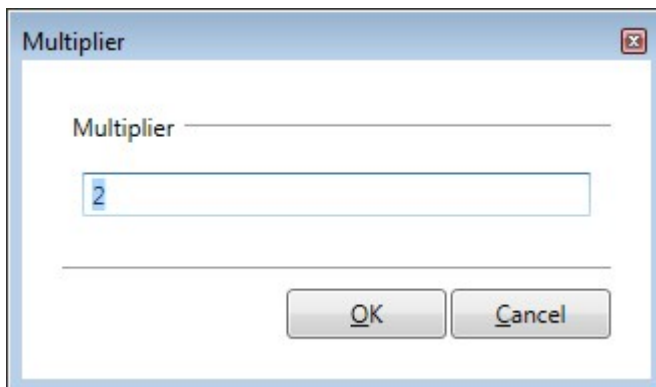
Drag & Drop

Bestimmte Elemente lassen sich per Drag & Drop von einem Tool Window in ein anderes kopieren. Auch innerhalb eines Tool Windows wird Drag & Drop z.B. für das Ändern der Position genutzt.

Gehen Sie mit der Maus auf das gewünschte Element und halten sie die linke Maustaste gedrückt. Ziehen Sie jetzt bei gedrückter Maustaste das Element auf die gewünschte Position. Das Icon unterhalb des Mauszeigers gibt darüber Auskunft, ob es an der aktuellen Stelle abgelegt werden kann.

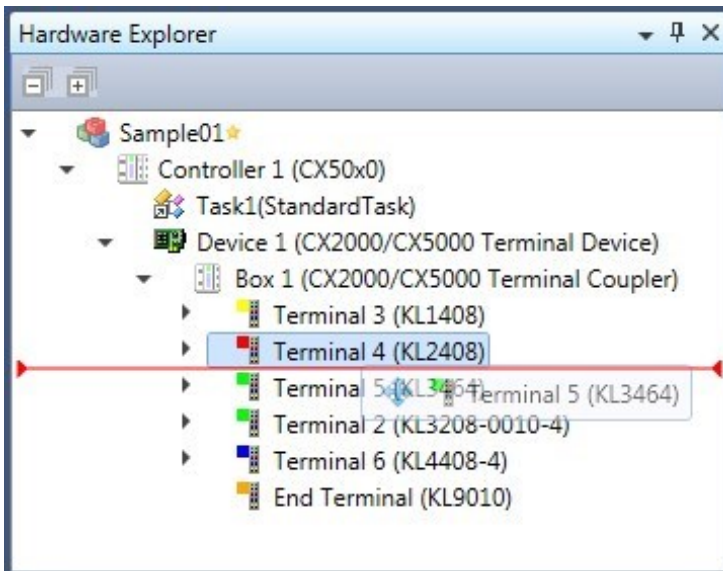
Wenn Sie vor dem Loslassen der linken Maustaste, die Shift-Taste gedrückt halten, können Sie mehrere Elemente gleichzeitig einfügen.

Geben Sie in dem Dialog die Anzahl der Elemente an, die sie hinzufügen wollen.



In einigen Tool Windows kann durch Drag & Drop die Position der Elemente verändert werden.

Ist die Position gültig, markiert eine rote Linie die Position, an die das Element verschoben wird.

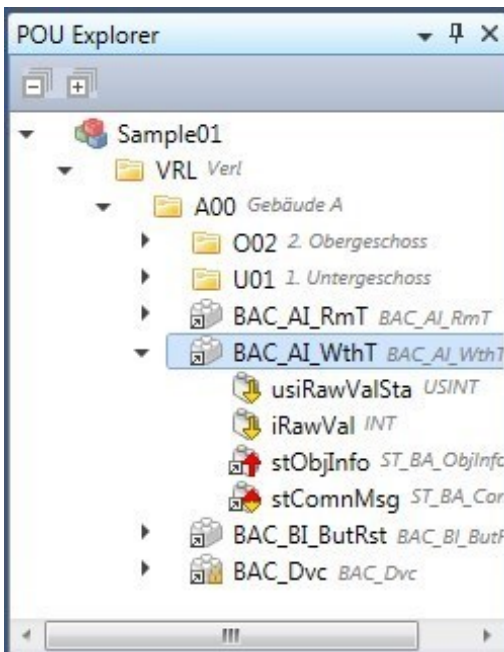


POU Explorer

Der *POU Explorer* strukturiert die einzelnen POU's gemäß dem Analgenkennzeichnungssystem (AKS). Jede Ordner-Ebene repräsentiert einen bestimmten Bereich des AKS.

Die POU's werden aus dem Tool Window *Templates* per Drag & Drop oder über das Kontextmenü auf den gewünschten Ordner gezogen.

Aus der Position innerhalb des *POU Explorers* wird beim Erzeugen der TwinCAT Projektdateien der vollständige Name des SPS-Bausteines abgeleitet.

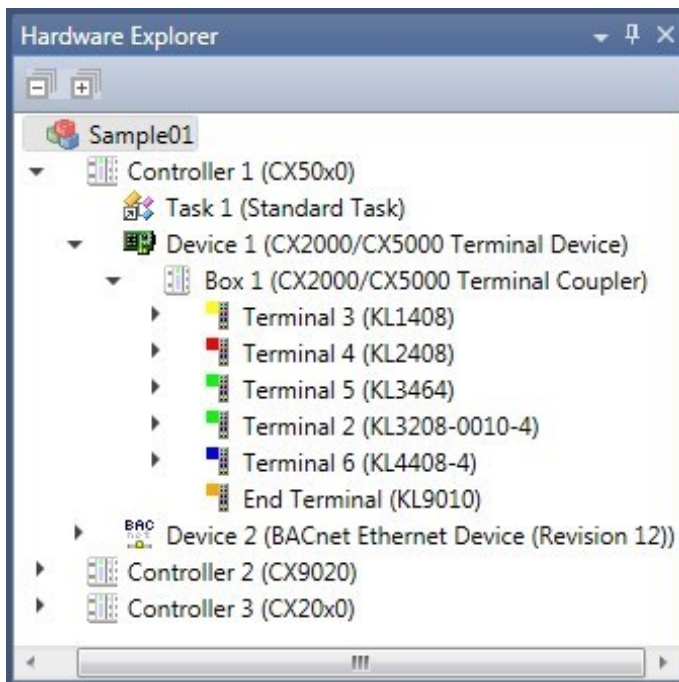


Ein Doppelklick auf ein POU öffnet im *Detail View* einen weiteren Dialog. Über diesen Dialog können zusätzliche Parameter, wie z.B. BACnet Properties, editiert werden.

Zu jedem markierten Element im *POU Explorer* werden im Tool Window *Properties* weitere Informationen angezeigt.

Hardware Explorer

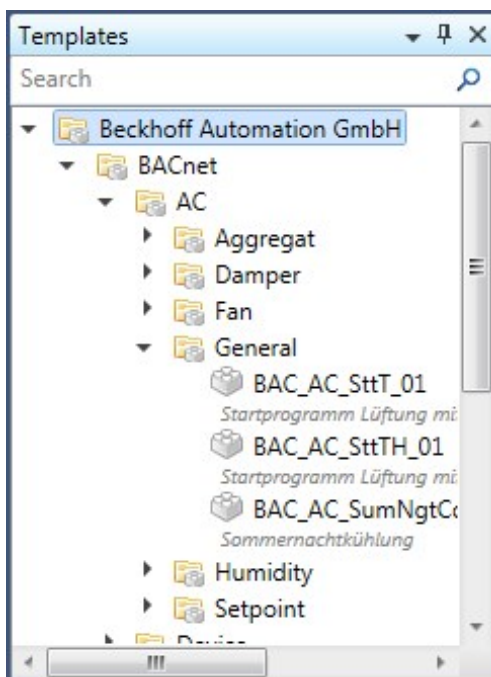
Im *Hardware Explorer* wird die verwendete Hardware eingetragen. Diese wird per Drag & Drop oder per Kontextmenü aus den Tool Windows *Devices* und *Terminals* hinzugefügt.



Auch hier öffnet ein Doppelklick im *Detail View* weitere Dialoge. Zu jedem selektierten Element werden im Tool Window *Properties* zusätzliche Informationen angezeigt.

Templates

Alle Templates, die dem *POU Explorer* hinzugefügt werden können, werden in diesem Tool Window angezeigt.



Über das Suchfeld im oberen Bereich kann durch Eingabe eines Namens die Anzeige im Tool Window gefiltert werden.

Das Tool Window *Properties* zeigt weitere Informationen zu jedem markierten Element an.

Devices

Die Elemente, die dem *Hardware Explorer* hinzugefügt werden können, werden im Tool Window *Devices* angezeigt.

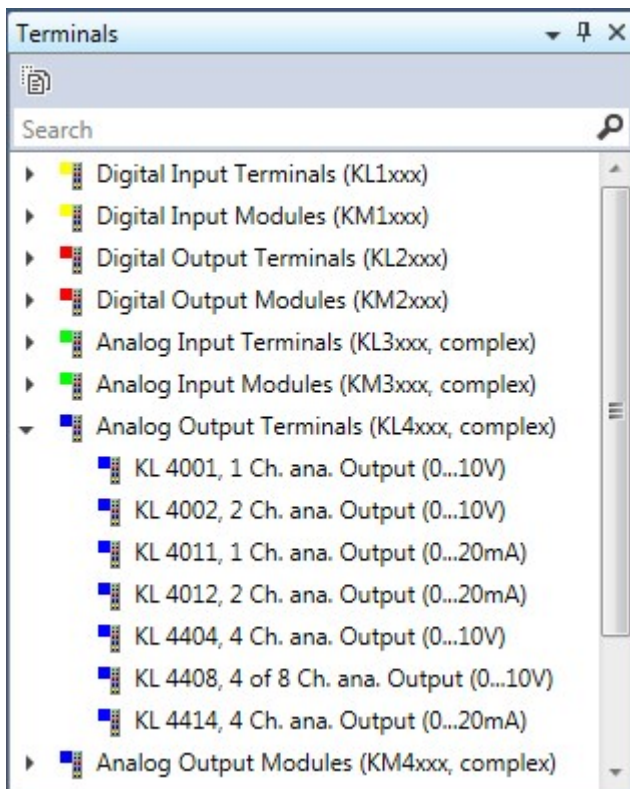


Über das Suchfeld im oberen Bereich kann durch die Eingabe eines Namens die Anzeige im Tool Window gefiltert werden.

Das Tool Window *Properties* zeigt weitere Informationen zu jedem markierten Element an.

Terminals

Alle Klemmen, die dem *Hardware Explorer* hinzugefügt werden können, befinden sich in dem Tool Window *Terminals*.



Um die Auswahl der Klemme zu beschleunigen, werden nur die meist benutzten Klemmen angezeigt. Durch die Schaltfläche im oberen Bereich wird der Anzeigefilter deaktiviert und alle verfügbaren Klemmen werden angezeigt.

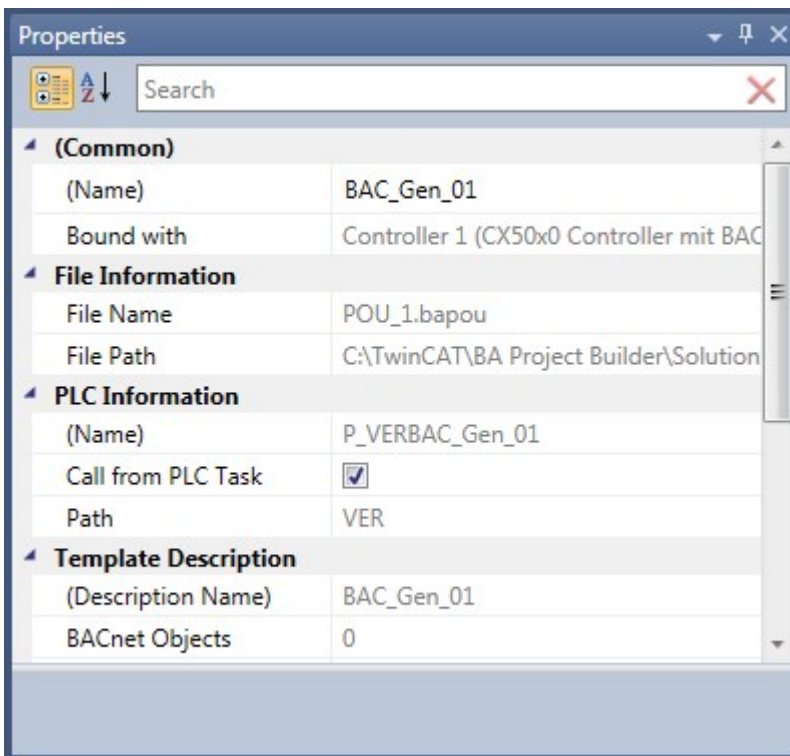
Über das Suchfeld im oberen Bereich kann durch die Eingabe eines Namens die Anzeige im Tool Window gefiltert werden.

Das Tool Window *Properties* zeigt weitere Informationen zu jedem markierten Element an.

Properties

Das Tool Window *Properties* zeigt von dem aktuell markierten Element eines anderen Fensters weitere Informationen an.

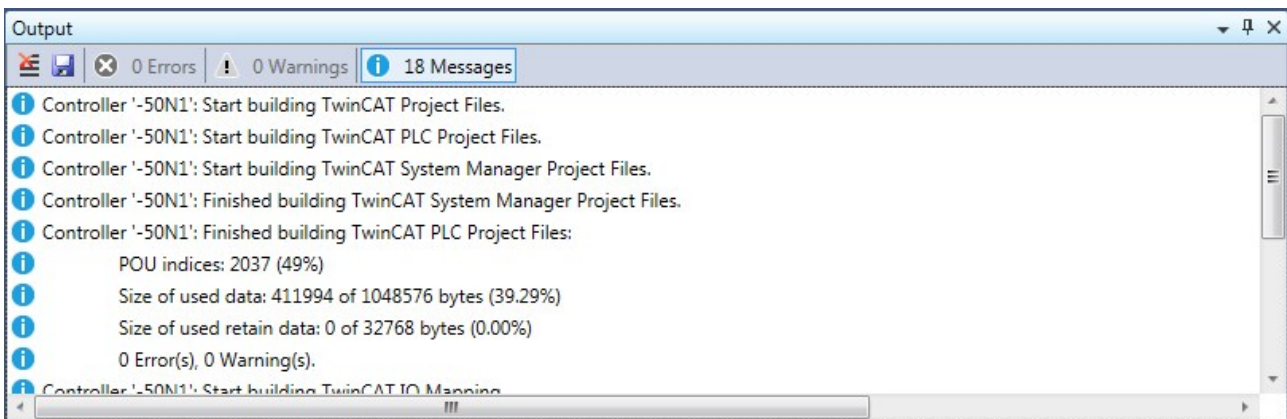
Mit den beiden Schaltflächen im oberen Bereich kann die Sortierung verändert werden (alphabetisch oder nach Kategorie).



Über das Suchfeld im oberen Bereich kann durch die Eingabe eines Namens die Anzeige im Tool Window gefiltert werden.

Output

Im Tool Window *Output* werden diverse Meldungen angezeigt. So werden Meldungen während der Erzeugung der TwinCAT Projektdateien ausgegeben oder wenn Fehler in der Projektierung erkannt werden.



Über die Schaltfläche im linken oberen Bereich kann das Tool Window *Output* gelöscht werden. Des Weiteren können die Meldungen in eine Textdatei abgespeichert werden.

Die Meldungen werden in drei Kategorien eingeteilt; *Errors*, *Warnings* und *Messages*. Jede Kategorie kann durch Betätigen der entsprechenden Schaltfläche oberen Bereich ausgeblendet werden.

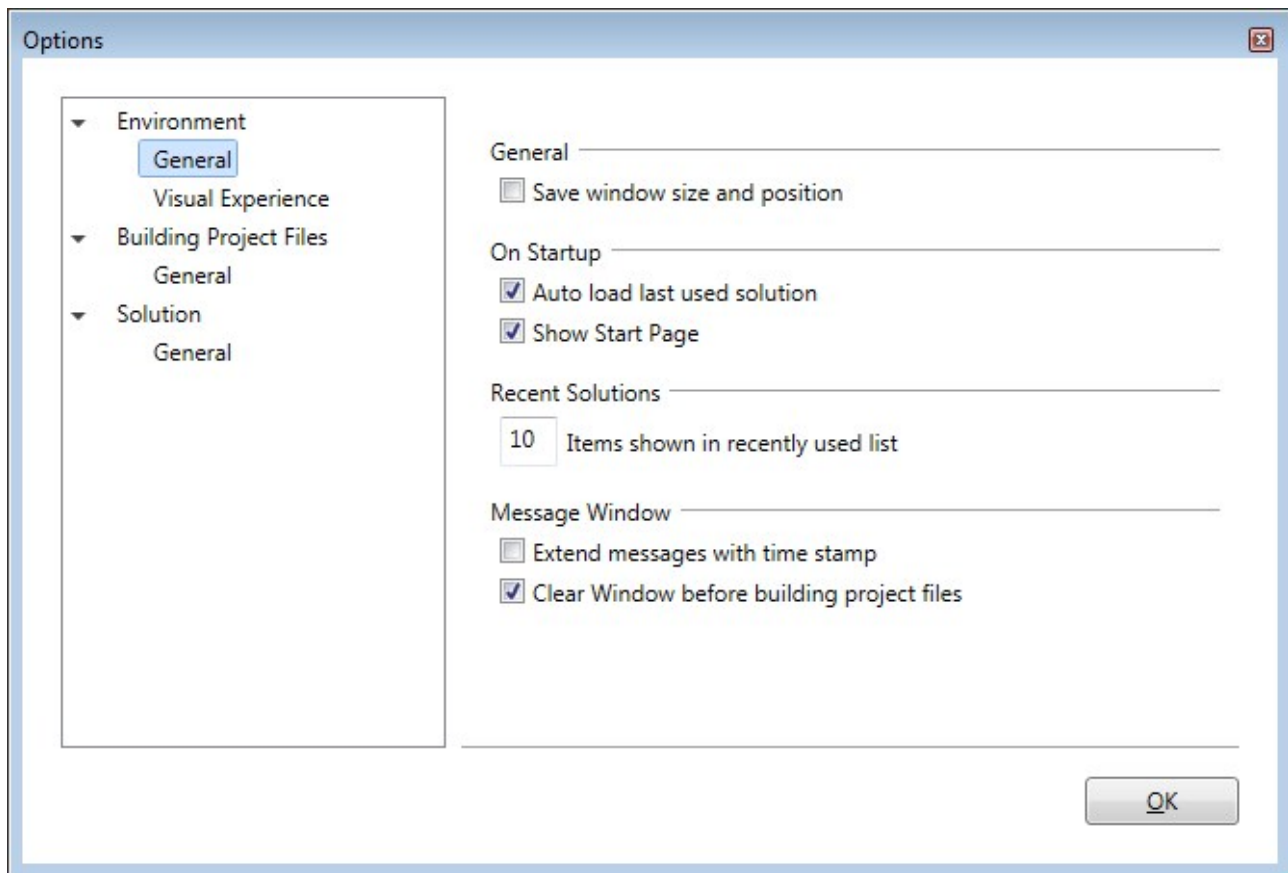
Durch Drücken der F4-Taste wird die nächste Fehlermeldung (Error) markiert.

10.3 Optionen

Über den Menüpunkt *Tools* --> *Options* kann der Dialog *Options* aufgerufen werden. Dieser Dialog bietet verschiedene Parameter an, welche das Aussehen als auch das Verhalten des BA Project Builder beeinflussen.

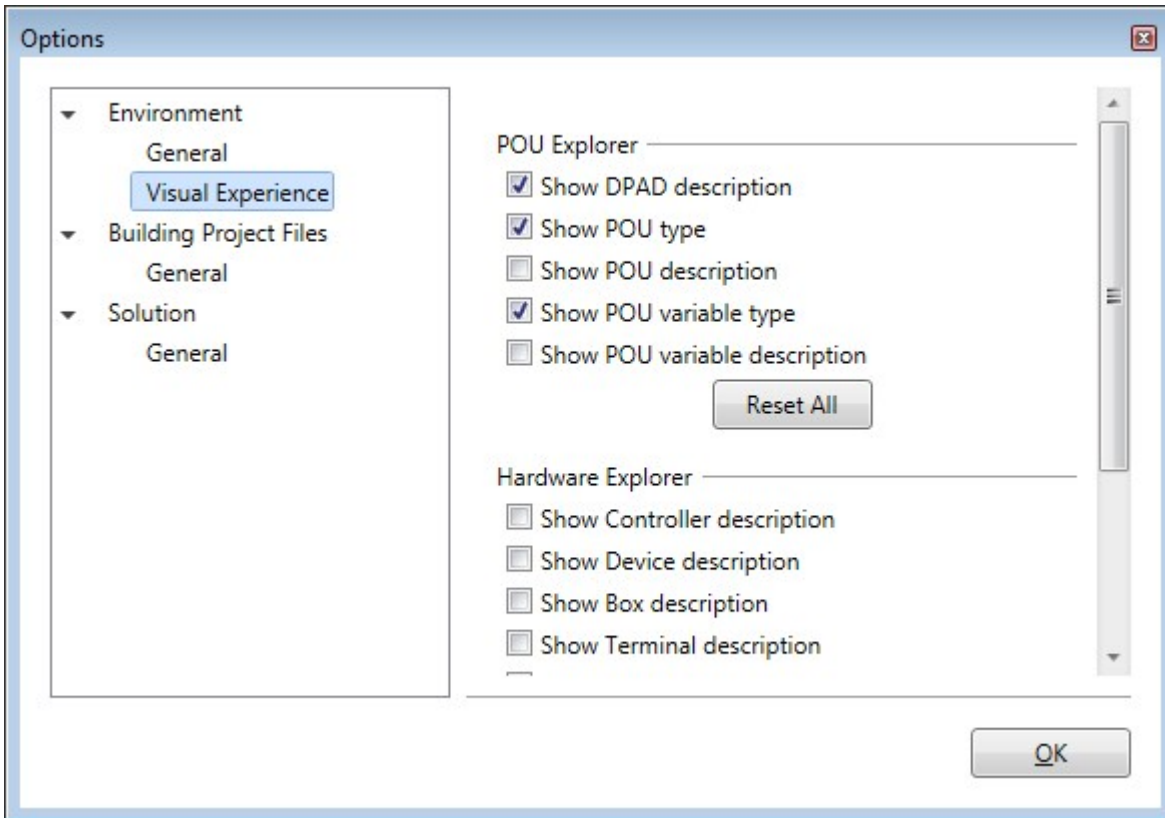
Environment

General



Optionsbereich	Funktion
General	Der BA Project Builder speichert beim Beenden die aktuellen Positionen und Größen der einzelnen Tool Windows. Diese werden beim Starten wieder in den vorherigen Zustand gebracht.
On Startup	Es kann festgelegt werden, ob beim Starten des BA Project Builders die zuletzt verwendete Solution automatisch geladen und ob die Start Page angezeigt wird.
Recent Solutions	Hier kann eingestellt werden, wie viele der zuletzt geladenen Solutions direkt über den Menüpunkt <i>Solution -> Open Recent</i> wieder geöffnet werden können.
Message Window	Das Verhalten des Tool Window <i>Output</i> kann durch diese Optionen angepasst werden.

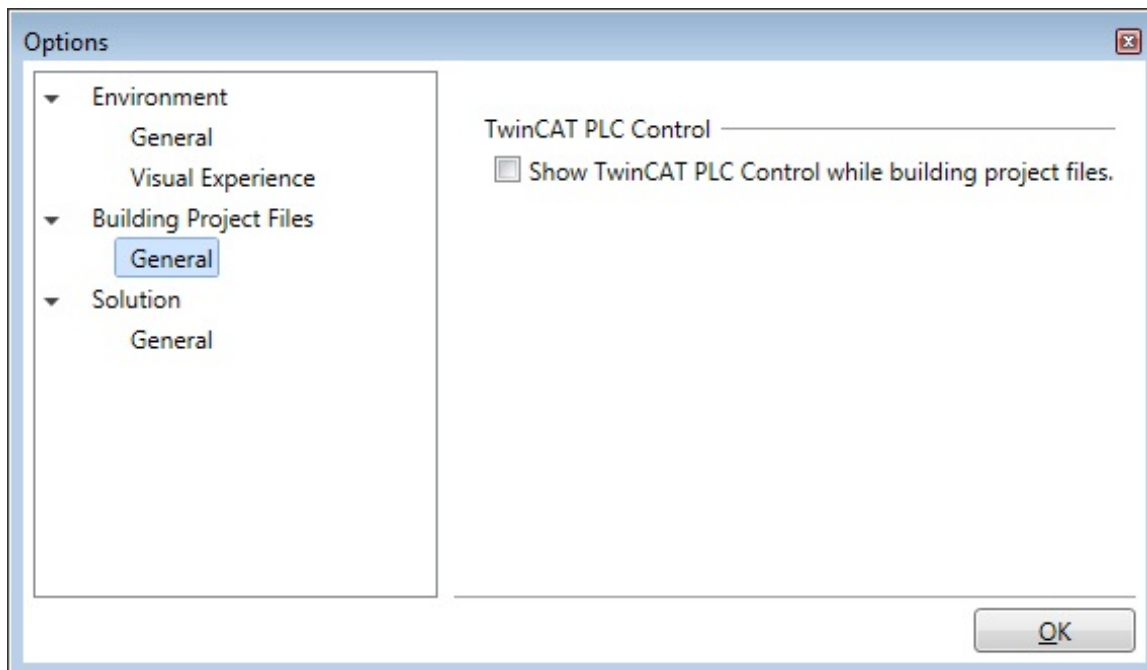
Visual Experience



Optionsbereich	Funktion
POU Explorer	Im Tool Window <i>POU Explorer</i> lassen sich zusätzliche Informationen an den einzelnen Elementen ein- und ausblenden.
Hardware Explorer	Im Tool Window <i>Hardware Explorer</i> lassen sich zusätzliche Informationen an den einzelnen Elementen ein- und ausblenden.
Templates Tool Window	Im Tool Window <i>Templates</i> lassen sich zusätzliche Informationen an den einzelnen Elementen ein- und ausblenden.

Building Project Files

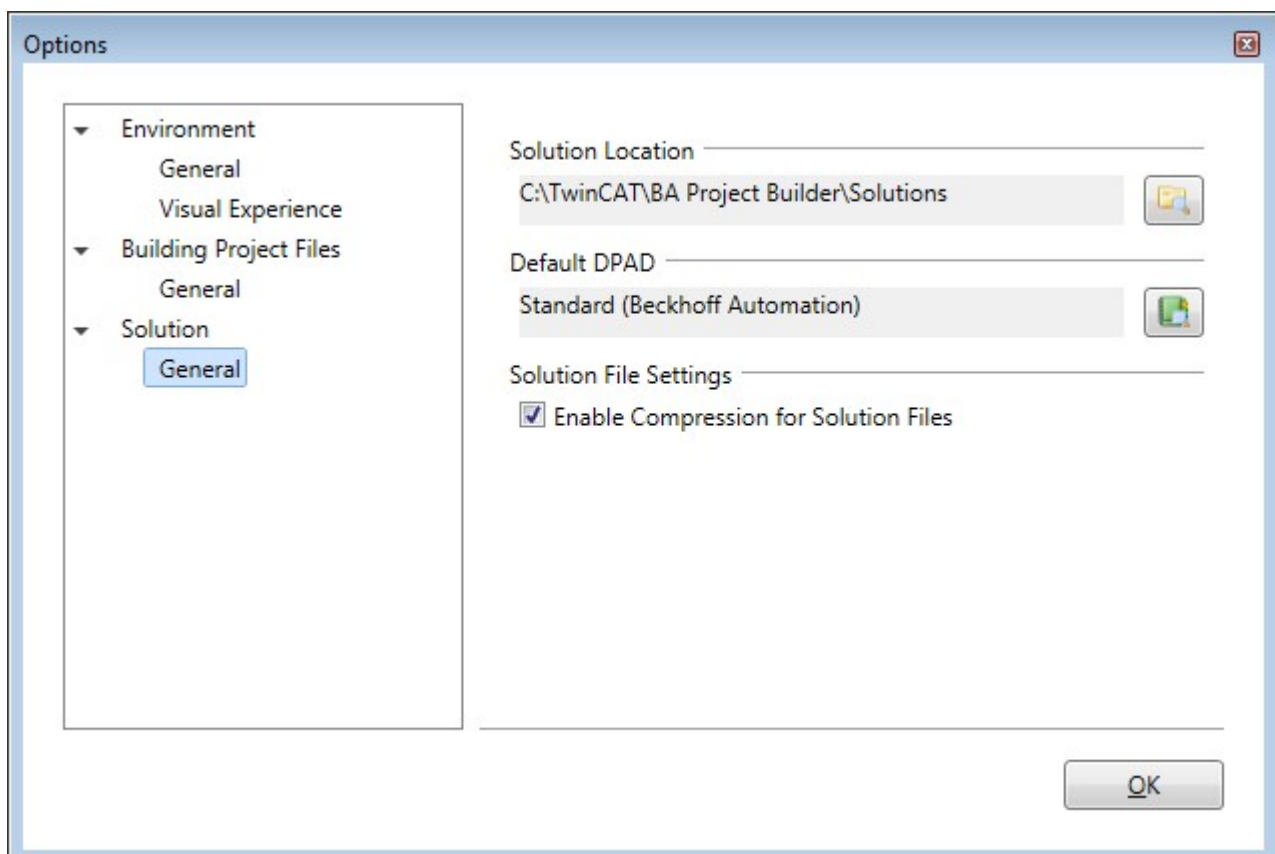
General



Optionsbereich	Funktion
TwinCAT PLC Control	Während der Erzeugung der TwinCAT Projektdateien wird das TwinCAT PLC Control angezeigt.

Solution

General



Optionsbereich	Funktion
Solution Location	Dieser Ordner wird beim Erzeugen einer neuen Solution vorgegeben.
Default DPAD	Dieses AKS wird beim Erzeugen einer neuen Solution vorgegeben.
Solution File Settings	Die Solution-Dateien werden möglichst platzsparend abgespeichert. Dadurch werden die Dateien deutlich kleiner, können allerdings nicht mehr direkt betrachtet (z.B. in einem Texteditor) werden.

10.4 AddIns erstellen

Der TwinCAT BA Project Builder bietet die Möglichkeit eigene AddIns zu erstellen. AddIns haben die Möglichkeit über ein Objektmodel auf die interne Datenhaltung des TwinCAT BA Project Builders zuzugreifen. Diese Zugriffe können lesend sowie schreibend erfolgen.

Eine ausführliche Dokumentation der AddIn API erhalten Sie hier: <https://infosys.beckhoff.com/content/1031/tcba/Resources/12269790091.zip>

10.4.1 Erste Schritte - WPF

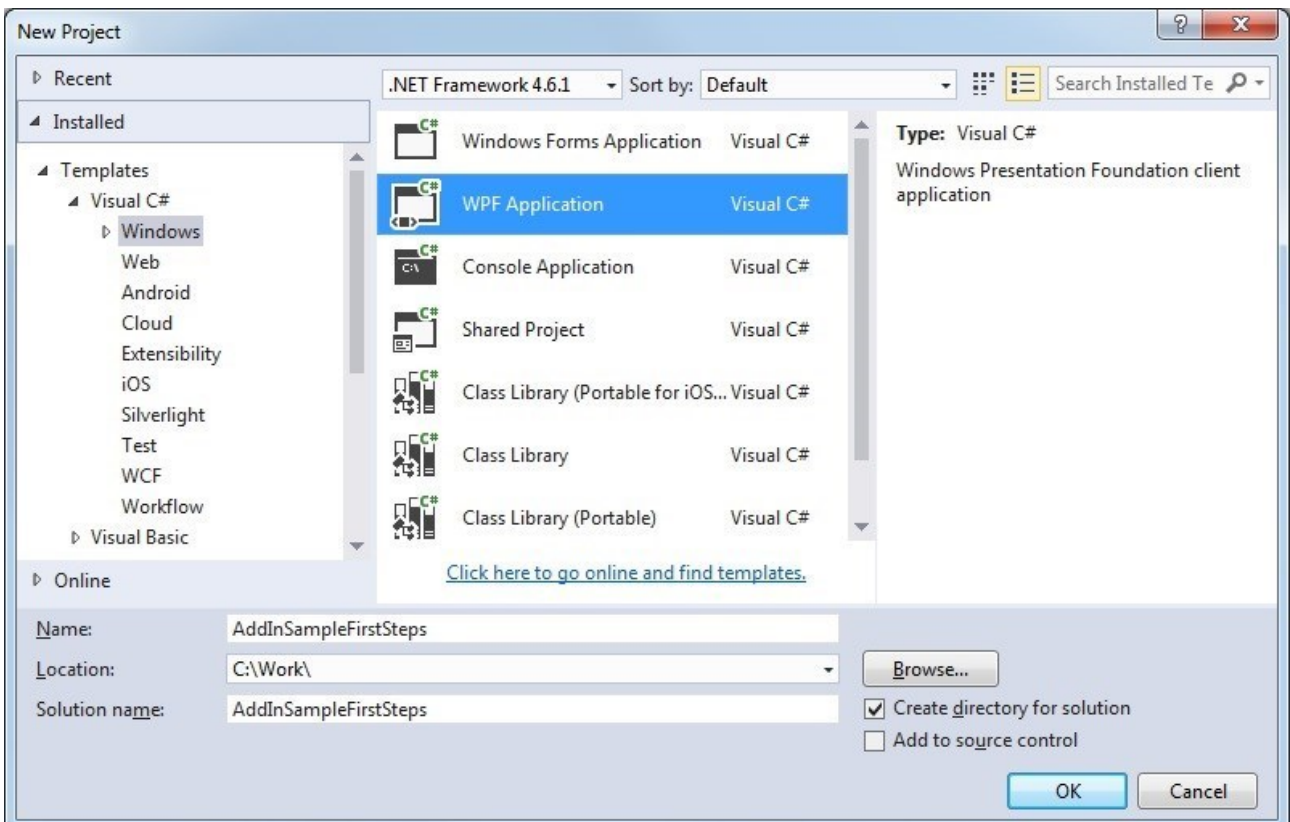
Die nachfolgenden Ausführungen zeigen alle notwendigen Schritte auf, um eine Grundlage für die eigentliche AddIn Entwicklung zu schaffen. Das vollständige Microsoft Visual Studio Projekt erhalten Sie hier: <https://infosys.beckhoff.com/content/1031/tcba/Resources/12269791499.zip>.

Voraussetzungen

- Microsoft Visual Studio 2013 oder höher
- Microsoft .NET Framework 4.6.1 oder höher
- Kenntnisse in der Programmiersprache C#

Projekt erstellen

Starten Sie das Microsoft Visual Studio und legen Sie ein neues WPF Projekt an. Wählen Sie zusätzlich *.NET Framework 4.6.1* aus.



Öffnen Sie jetzt die Eigenschaften des Projekts und ändern Sie im Bereich *Application* die Einstellung *Output type* auf *Class Library*. Prüfen Sie, ob *Target framework* auf *.NET Framework 4.6.1* gesetzt wurde.

Wechseln Sie zum Bereich *Build* und tragen Sie unter *Output path* den Pfad zum AddIn Ordner des TwinCAT BA Project Builders ein (standardmäßig *TwinCAT -> BA Project Builder -> AddIns*). Wählen Sie unter *Platform target* den Wert *x86* aus.

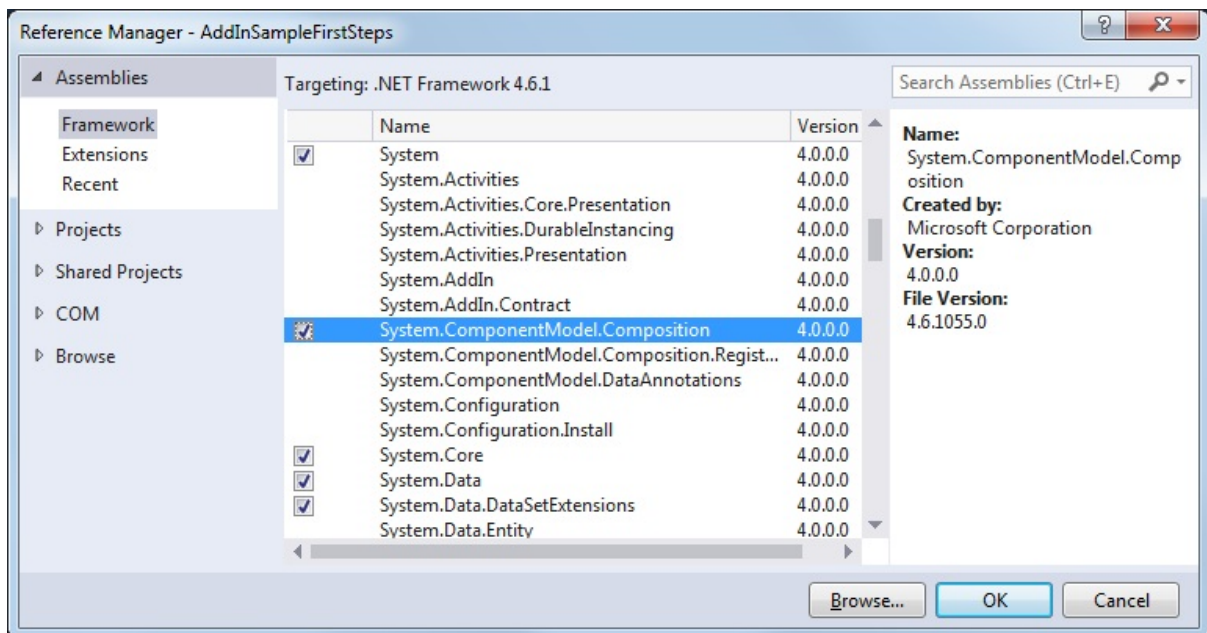
Da es sich bei diesem Projekt um eine Klassenbibliothek handelt, müssen Sie zum Starten und Debuggen ein externes Programm definieren mit dem das AddIn gestartet werden soll. Im Bereich *Debug* tragen Sie bei *Start external program* den Pfad zum TwinCAT BA Project Builder ein (standardmäßig *TwinCAT -> BA Project Builder -> TwinCAT.BA.ProjectBuilder.UI.exe*).

Entfernen Sie die standardmäßig erstellte Datei *App.xml* aus dem Projekt.

Referenzen hinzufügen

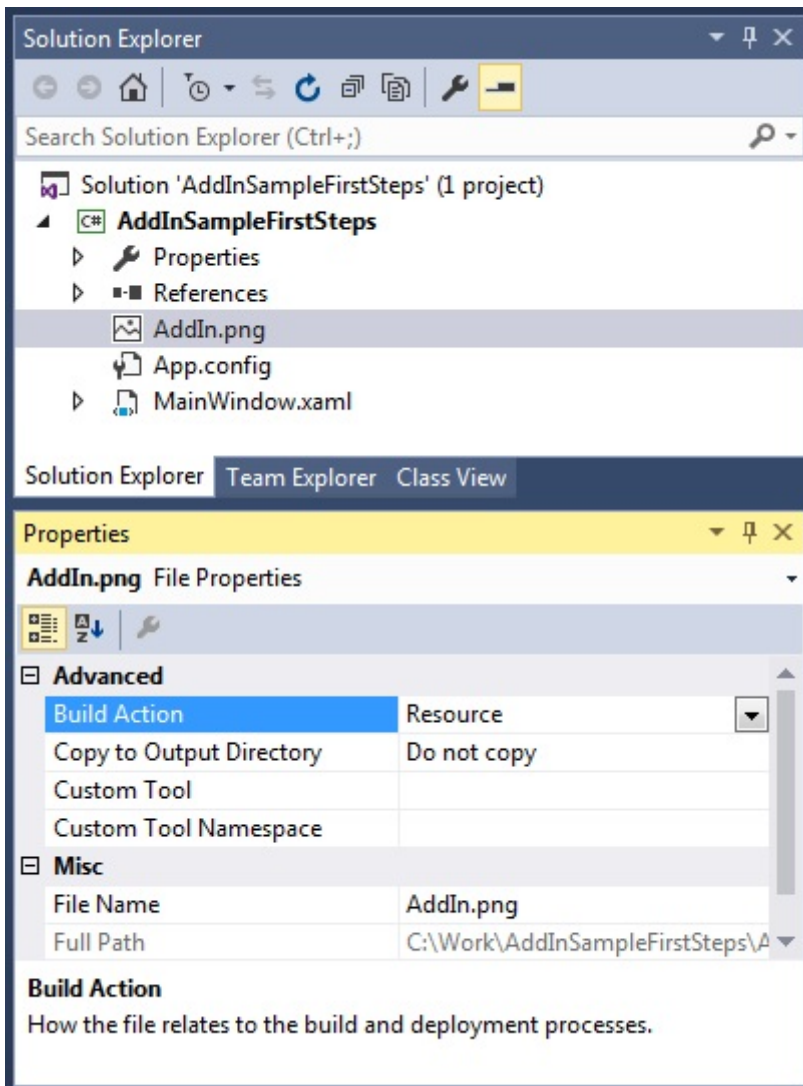
Der nächste Schritt umfasst das Hinzufügen von drei Referenzen.

- **System.ComponentModel.Composition:** Der TwinCAT BA Project Builder nutzt MEF (Managed Extensibility Framework) als Basistechnologie für die Erweiterbarkeit durch AddIns. Alle notwendigen Klassen für die Nutzung von MEF sind in dieser DLL enthalten.
- **TwinCAT.BA.ProjectBuilder.AddIn.Contract:** Über diese Referenz werden Interfaces zur Verfügung gestellt, die das AddIn implementieren muss, damit dieses vom TwinCAT BA Project Builder instanziiert werden kann. Enthalten ist diese in der Datei *TwinCAT -> BA Project Builder -> TwinCAT.BA.ProjectBuilder.AddIn.Contract.dll*. Setzen Sie für diese Referenz die Option *Copy Local* auf *False*.
- **TwinCAT.BA.ProjectBuilder.AddIn.IL:** Diese Referenz bietet Klassen für den Zugriff auf das interne Objektmodell des TwinCAT BA Project Builder. Enthalten ist diese in der Datei *TwinCAT -> BA Project Builder -> TwinCAT.BA.ProjectBuilder.AddIn.IL.dll*. Setzen Sie für diese Referenz die Option *Copy Local* auf *False*.



Icon hinzufügen

Fügen Sie, wenn gewünscht, dem Projekt ein Icon (16 x 16 Pixel) hinzu und setzen Sie die Eigenschaft *Build Action* auf *Resource*.



Bereitstellung

Damit das AddIn im TwinCAT BA Project Builder angezeigt und geladen werden kann, müssen zwei Klassen hinzugefügt werden. Zum einen eine Klasse die von *ContractAttribute* abgeleitet wird und einige Properties überschreibt. Diese dienen zur Anzeige des AddIns im TwinCAT BA Project Builder. Des Weiteren muss eine Klasse angelegt werden, die das Interface *IContractWpf* enthält. Durch die Methode *GetWindow* wird über den TwinCAT BA Project Builder das AddIn geladen.

Legen sie eine neue Klasse mit den Namen *AddInEntryMetadata* an und leiten diese von *ContractAttribute* ab. Fügen Sie die Eigenschaften *Description*, *ShortDescription*, *Name*, *Version* und *Icon* hinzu. Diese Informationen werden im TwinCAT BA Project Builder bei der Auswahl des AddIns angezeigt.

```
using TwinCAT.BA.ProjectBuilder.AddIn.Contract;

namespace AddInSampleFirstSteps
{
    public class AddInEntryMetadata : ContractAttribute
    {
        // Gets the description of the AddIn.
        public override string Description
        {
            get
            {
                return "AddIn Sample FirstSteps of the TwinCAT BA Project Builder for WPF";
            }
        }
    }
}
```

```

// Gets the short description of the AddIn.
public override string ShortDescription
{
    get
    {
        return "AddIn Sample FirstSteps for WPF";
    }
}

// Gets the name of the AddIn.
public override string Name
{
    get
    {
        return "Sample FirstSteps WPF";
    }
}

// Gets the version of the AddIn.
public override string Version
{
    get
    {
        return "1.0.0";
    }
}

// Gets the contract metadata icon path.
// "pack://application:,,,/<- Assembly name ->;component/<- Subfolder or image/icon file name"
public override string Icon
{
    get
    {
        return "pack://application:,,,/AddInSampleFirstSteps-WPF;component/AddIn.png";
    }
}
}

```

Legen Sie nun die Klasse *AddInEntryPoint* an und implementieren Sie das Interface *IContractWpf*. Ausserdem muss die Klasse mit den Attributen *Export* und *AddInEntryMetadata* dekoriert werden.

Das Interface enthält die Methode *GetWindow()*. Diese wird vom BA Project Builder aufgerufen, wenn das AddIn angezeigt werden soll. Des Weiteren müssen zwei Events implementiert werden, die für den Zugriff auf das Window *Output* vom BA Project Builder notwendig sind.

```

using System;
using System.ComponentModel.Composition;
using System.Windows;
using TwinCAT.BA.ProjectBuilder.AddIn.Contract;

namespace AddInSampleFirstSteps
{
    [Export(typeof(IContractWpf))] // Marks this class as an entry point for the TwinCAT BA Project
    Builder
    [AddInEntryMetadata] // Adds necessary information about the AddIn to this class
    public class AddInEntryPoint : IContractWpf
    {
        private AddInEntryMetadata addInEntryMetadata = null;
        public event EventHandler<ContractLogEventArgs> LogEvent;
        public event EventHandler<ContractClearLogEventArgs> ClearLogEvent;

        public Window GetWindow()
        {
            this.addInEntryMetadata = new AddInEntryMetadata();
            // Passing the main class of the AddIn
            return new MainWindow(this);
        }

        public void RaiseLogEvent(LogLevel level, string message)
        {
            OnLog(new ContractLogEventArgs(this.addInEntryMetadata, level, message));
        }

        // The event-invoking method that derived classes can override.
        private void OnLog(ContractLogEventArgs e)
        {
            // Make a temporary copy of the event to avoid possibility of

```

```

// a race condition if the last subscriber unsubscribes
// immediately after the null check and before the event is raised.
EventHandler<ContractLogEventArgs> handler = this.LogEvent;
if (handler != null)
{
    handler(this, e);
}
}

public void RaiseClearLogEvent()
{
    OnClearLog(new ContractClearLogEventArgs(this.addInEntryMetadata));
}

// The event-invoking method that derived classes can override.
private void OnClearLog(ContractClearLogEventArgs e)
{
    // Make a temporary copy of the event to avoid possibility of
    // a race condition if the last subscriber unsubscribes
    // immediately after the null check and before the event is raised.
    EventHandler<ContractClearLogEventArgs> handler = this.ClearLogEvent;
    if (handler != null)
    {
        handler(this, e);
    }
}
}
}

```

Programmierung

Die Hauptfunktionalitäten werden in der Klasse *MainWindow* implementiert.

```

using System.Windows;
using TwinCAT.BA.ProjectBuilder.AddIn.IL;
using TwinCAT.BA.ProjectBuilder.AddIn.IL.Hardware;

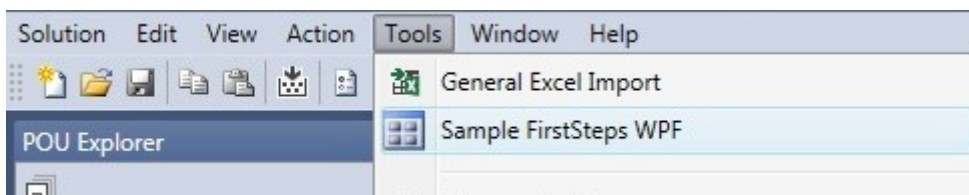
namespace AddInSampleFirstSteps
{
    public partial class MainWindow : Window
    {
        AddInEntryPoint addInEntryPoint = null;
        BAProjectBuilder projectBuilder = null;

        public MainWindow(AddInEntryPoint addInEntryPoint)
        {
            InitializeComponent();
            // Code here
            this.addInEntryPoint = addInEntryPoint;
            this.projectBuilder = new BAProjectBuilder();
        }
    }
}

```

Debuggen

Der TwinCAT BA Project Builder wird automatisch mit dem Projekt gestartet. Sie können das AddIn sowohl über das Menü *Tools* vom TwinCAT BA Project Builder ausführen ...



... als auch über die Toolbar.



In beiden Fällen wird durch den TwinCAT BA Project Builder eine Instanz vom AddIn angelegt und die Methode *GetWindow* aufgerufen.

Von der Klasse *AddInEntryMetadata* wird schon beim Starten des BA Project Builders eine Instanz angelegt und über die Properties alle Informationen ausgelesen, die für die Darstellung im TwinCAT BA Project Builder notwendig sind.

10.4.2 Erste Schritte - WinForms

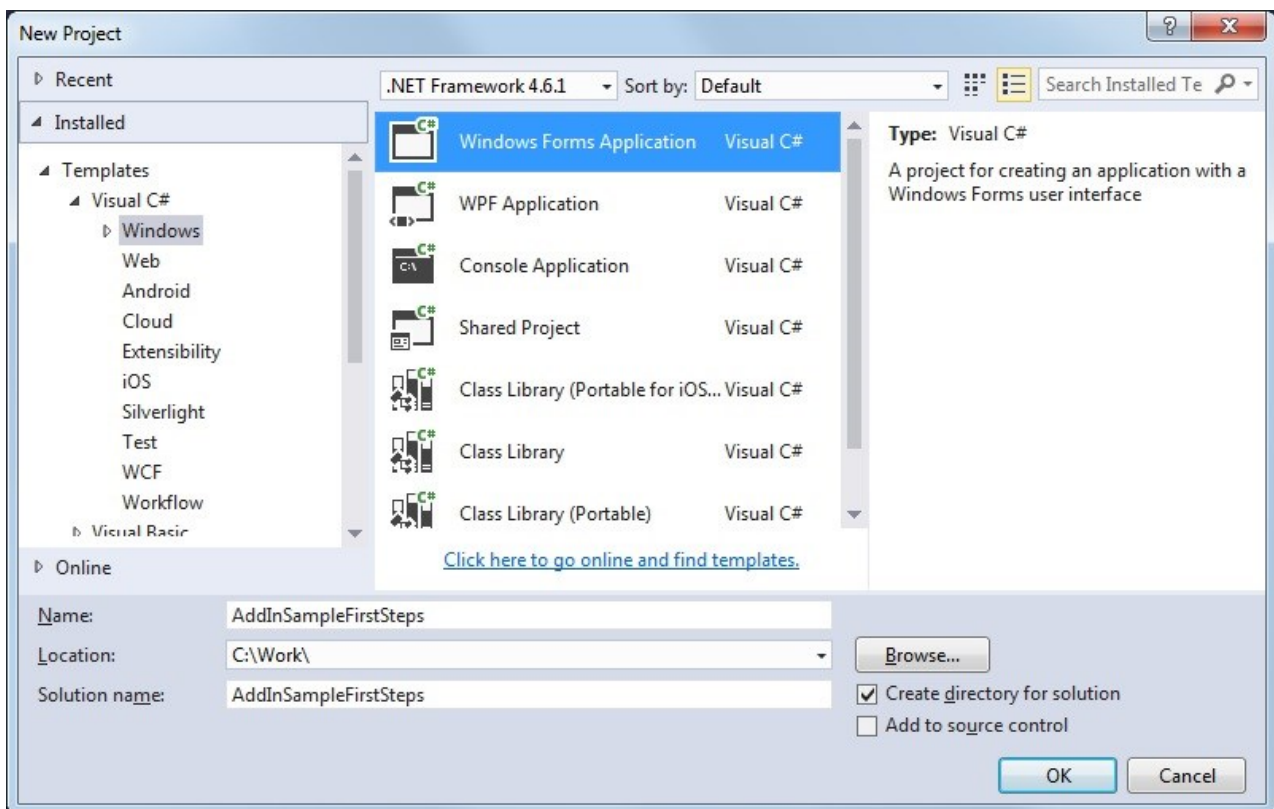
Die nachfolgenden Ausführungen zeigen alle notwendigen Schritte auf, um eine Grundlage für die eigentliche AddIn Entwicklung zu schaffen. Das vollständige Microsoft Visual Studio Projekt erhalten Sie hier: <https://infosys.beckhoff.com/content/1031/tcba/Resources/12269792907.zip>.

Voraussetzungen

- Microsoft Visual Studio 2013 oder höher
- Microsoft .NET Framework 4.6.1 oder höher
- Kenntnisse in der Programmiersprache C#

Projekt erstellen

Starten Sie das Microsoft Visual Studio und legen Sie ein neues Windows Forms Projekt an. Wählen Sie zusätzlich *.NET Framework 4.6.1* aus.



Öffnen Sie jetzt die Eigenschaften des Projekts und ändern Sie im Bereich *Application* die Einstellung *Output type* auf *Class Library*. Prüfen Sie, ob *Target framework* auf *.NET Framework 4.6.1* gesetzt wurde.

Wechseln Sie zum Bereich *Build* und tragen Sie unter *Output path* den Pfad zum AddIn Ordner des TwinCAT BA Project Builders ein (standardmäßig *TwinCAT -> BA Project Builder -> AddIns*). Wählen Sie unter *Platform target* den Wert *x86* aus.

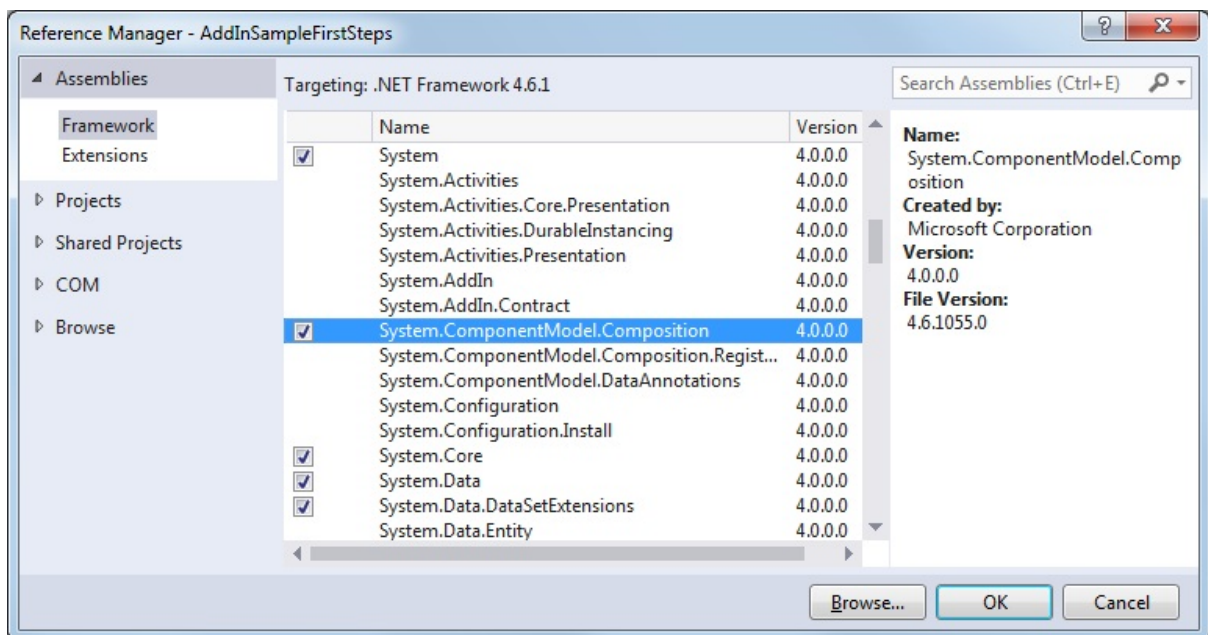
Da es sich bei diesem Projekt um eine Klassenbibliothek handelt, müssen Sie zum Starten und Debuggen ein externes Programm definieren mit dem das AddIn gestartet werden soll. Im Bereich *Debug* tragen Sie bei *Start external program* den Pfad zum TwinCAT BA Project Builder ein (standardmäßig *TwinCAT -> BA Project Builder -> TwinCAT.BA.ProjectBuilder.UI.exe*).

Entfernen Sie die standardmäßig erstellte Datei *Program.cs* aus dem Projekt.

Referenzen hinzufügen

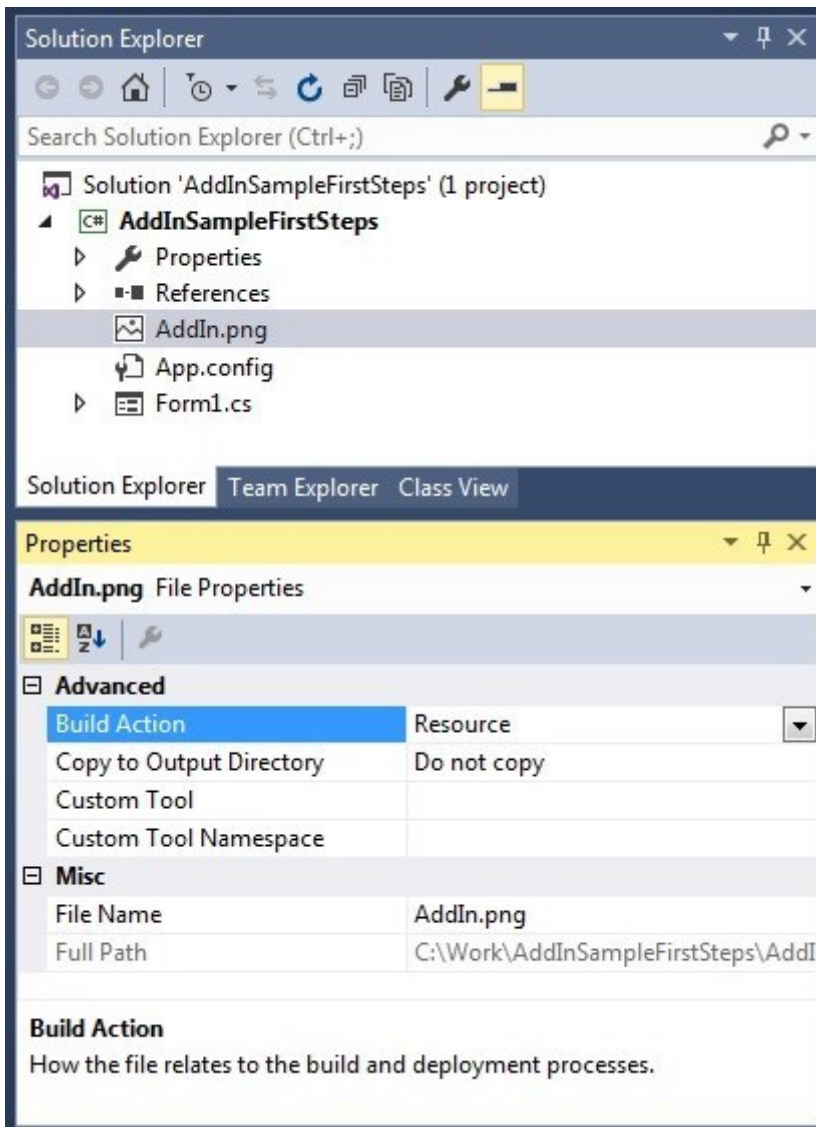
Der nächste Schritt umfasst das Hinzufügen von drei Referenzen.

- **System.ComponentModel.Composition:** Der TwinCAT BA Project Builder nutzt MEF (Managed Extensibility Framework) als Basistechnologie für die Erweiterbarkeit durch AddIns. Alle notwendigen Klassen für die Nutzung von MEF sind in dieser DLL enthalten.
- **TwinCAT.BA.ProjectBuilder.AddIn.Contract:** Über diese Referenz werden Interfaces zur Verfügung gestellt, die das AddIn implementieren muss, damit dieses vom TwinCAT BA Project Builder instanziiert werden kann. Enthalten ist diese in der Datei *TwinCAT -> BA Project Builder -> TwinCAT.BA.ProjectBuilder.AddIn.Contract.dll*. Setzen Sie für diese Referenz die Option *Copy Local* auf *False*.
- **TwinCAT.BA.ProjectBuilder.AddIn.IL:** Diese Referenz bietet Klassen für den Zugriff auf das interne Objektmodell des TwinCAT BA Project Builder. Enthalten ist diese in der Datei *TwinCAT -> BA Project Builder -> TwinCAT.BA.ProjectBuilder.AddIn.IL.dll*. Setzen Sie für diese Referenz die Option *Copy Local* auf *False*.



Icon hinzufügen

Fügen Sie, wenn gewünscht, dem Projekt ein Icon (16 x 16 Pixel) hinzu und setzen Sie die Eigenschaft *Build Action* auf *Resource*.



Bereitstellung

Damit das AddIn im TwinCAT BA Project Builder angezeigt und geladen werden kann, müssen zwei Klassen hinzugefügt werden. Zum einen eine Klasse die von *ContractAttribute* abgeleitet wird und einige Properties überschreibt. Diese dienen zur Anzeige des AddIns im TwinCAT BA Project Builder. Des Weiteren muss eine Klasse angelegt werden, die das Interface *IContractWinForms* enthält. Durch die Methode *GetForm* wird über den TwinCAT BA Project Builder das AddIn geladen.

Legen Sie eine neue Klasse mit den Namen *AddInEntryMetadata* an und leiten diese von *ContractAttribute* ab. Fügen Sie die Properties *Description*, *ShortDescription*, *Name*, *Version* und *Icon* hinzu. Diese Informationen werden im TwinCAT BA Project Builder bei der Auswahl des AddIns angezeigt.

```
using TwinCAT.BA.ProjectBuilder.AddIn.Contract;

namespace AddInSampleFirstSteps
{
    public class AddInEntryMetadata : ContractAttribute
    {
        // Gets the description of the AddIn.
        public override string Description
        {
            get
            {
                return "AddIn Sample FirstSteps of the TwinCAT BA Project Builder for WinForms";
            }
        }
    }
}
```

```

    }
}

// Gets the short description of the AddIn.
public override string ShortDescription
{
    get
    {
        return "AddIn Sample FirstSteps for WinForms";
    }
}

// Gets the name of the AddIn.
public override string Name
{
    get
    {
        return "Sample FirstSteps WinForms";
    }
}

// Gets the version of the AddIn.
public override string Version
{
    get
    {
        return "1.0.0";
    }
}

// Gets or sets the contract metadata icon path.
// "pack://application:,,,/<- Assembly name ->;component/<-Subfolder or image/icon file name"
public override string Icon
{
    get
    {
        return "pack://application:,,,/AddInSampleFirstSteps;component/AddIn.png";
    }
}
}
}

```

Legen Sie nun die Klasse *AddInEntryPoint* an und implementieren Sie das Interface *IContractWinForms*. Ausserdem muss die Klasse mit den Attributen *Export* und *AddInEntryMetadata* dekoriert werden.

Das Interface enthält die Methode *GetForm()*. Diese wird vom BA Project Builder aufgerufen, wenn das AddIn angezeigt werden soll. Des Weiteren müssen zwei Events implementiert werden, die für den Zugriff auf das Window *Output* vom BA Project Builder notwendig sind.

```

using System;
using System.ComponentModel.Composition;
using System.Windows.Forms;
using TwinCAT.BA.ProjectBuilder.AddIn.Contract;

namespace AddInSampleFirstSteps
{
    [Export(typeof(IContractWinForms))] // Marks this class as an entry point for the TwinCAT BA Project Builder
    [AddInEntryMetadata] // Adds necessary information about the AddIn to this class
    public class AddInEntryPoint : IContractWinForms
    {
        private AddInEntryMetadata addInEntryMetadata = null;
        public event EventHandler<ContractLogEventArgs> LogEvent;
        public event EventHandler<ContractClearLogEventArgs> ClearLogEvent;

        public Form GetForm()
        {
            this.addInEntryMetadata = new AddInEntryMetadata();
            // Passing the main class of the AddIn
            return new Form1(this);
        }

        public void RaiseLogEvent(LogLevel level, string message)
        {
            OnLog(new ContractLogEventArgs(this.addInEntryMetadata, level, message));
        }

        // The event-invoking method that derived classes can override.
        private void OnLog(ContractLogEventArgs e)
    }
}

```

```

    {
        // Make a temporary copy of the event to avoid possibility of
        // a race condition if the last subscriber unsubscribes
        // immediately after the null check and before the event is raised.
        EventHandler<ContractLogEventArgs> handler = this.LogEvent;
        if (handler != null)
        {
            handler(this, e);
        }
    }

    public void RaiseClearLogEvent()
    {
        OnClearLog(new ContractClearLogEventArgs(this.addInEntryMetadata));
    }

    // The event-invoking method that derived classes can override.
    private void OnClearLog(ContractClearLogEventArgs e)
    {
        // Make a temporary copy of the event to avoid possibility of
        // a race condition if the last subscriber unsubscribes
        // immediately after the null check and before the event is raised.
        EventHandler<ContractClearLogEventArgs> handler = this.ClearLogEvent;
        if (handler != null)
        {
            handler(this, e);
        }
    }
}

```

Programmierung

Die Hauptfunktionalitäten werden in der Klasse *Form1* implementiert.

```

using System.Windows.Forms;
using TwinCAT.BA.ProjectBuilder.AddIn.IL;

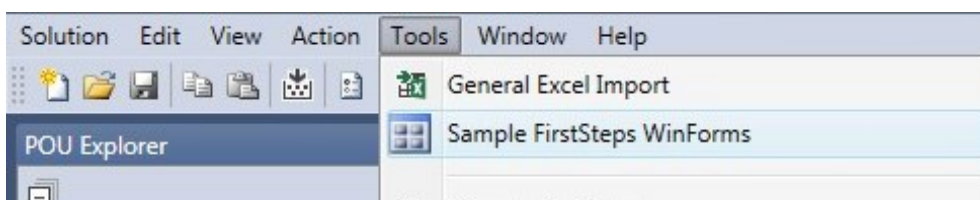
namespace AddInSampleFirstSteps
{
    public partial class Form1 : Form
    {
        AddInEntryPoint addInEntryPoint = null;
        BAProjectBuilder projectBuilder = null;

        public Form1(AddInEntryPoint addInEntryPoint)
        {
            InitializeComponent();
            // Code here
            this.addInEntryPoint = addInEntryPoint;
            this.projectBuilder = new BAProjectBuilder();
        }
    }
}

```

Debuggen

Der TwinCAT BA Project Builder wird automatisch mit dem Projekt gestartet. Sie können das AddIn sowohl über das Menü *Tools* vom TwinCAT BA Project Builder ausführen ...



... als auch über die Toolbar.



In beiden Fällen wird durch den TwinCAT BA Project Builder eine Instanz vom `AddIn` angelegt und die Methode `GetForm` aufgerufen.

Von der Klasse `AddInEntryMetadata` wird schon beim Starten des BA Project Builders eine Instanz angelegt und über die Properties alle Informationen ausgelesen, die für die Darstellung im TwinCAT BA Project Builder notwendig sind.

10.4.3 Beispiele

How To
Beispiel 01 - Zugriff auf das Output Window [► 765]
Beispiel 02 - Auslesen einer Template Description [► 765]
Beispiel 03 - Auflisten aller Terminal Descriptions [► 766]
Beispiel 04 - Hardware anlegen [► 767]

10.4.3.1 Beispiel 01: Zugriff auf das Output Window

Über das Interface `IContract` stehen Event Handler zur Verfügung, mit denen Meldungen in das Output Window geschrieben oder die Anzeige gelöscht werden kann. `IContract` wird an das Interface `IContractWpf` vererbt und von der Klasse `AddInEntryPoint` implementiert.

Beispiel C#

```
public partial class MainWindow : Window
{
    AddInEntryPoint addInEntryPoint = null;

    public MainWindow(AddInEntryPoint addInEntryPoint)
    {
        InitializeComponent();
        // Code here
        this.addInEntryPoint = addInEntryPoint;
    }

    private void buttonClear_Click(object sender, RoutedEventArgs e)
    {
        this.addInEntryPoint.RaiseClearLogEvent();
    }

    private void buttonLogEvent_Click(object sender, RoutedEventArgs e)
    {
        this.addInEntryPoint.RaiseLogEvent((LogLevel)comboBoxLogLevelCombo.SelectedIndex, textBoxMessage
        .Text);
    }
}
```

Download

<https://infosys.beckhoff.com/content/1031/tcba/Resources/12269794315.zip>

10.4.3.2 Beispiel 02: Auslesen einer Template Description

Allgemeine Informationen über ein Template werden durch die Klasse `TemplateDescription` zur Verfügung gestellt. Die Klasse `TemplateDescriptionProvider` und deren statische Methode `GetTemplateDecription` geben die Template Description des gewünschten Templates zurück.

Beispiel C#

```
public partial class MainWindow : Window
{
    BAProjectBuilder projectBuilder = null;

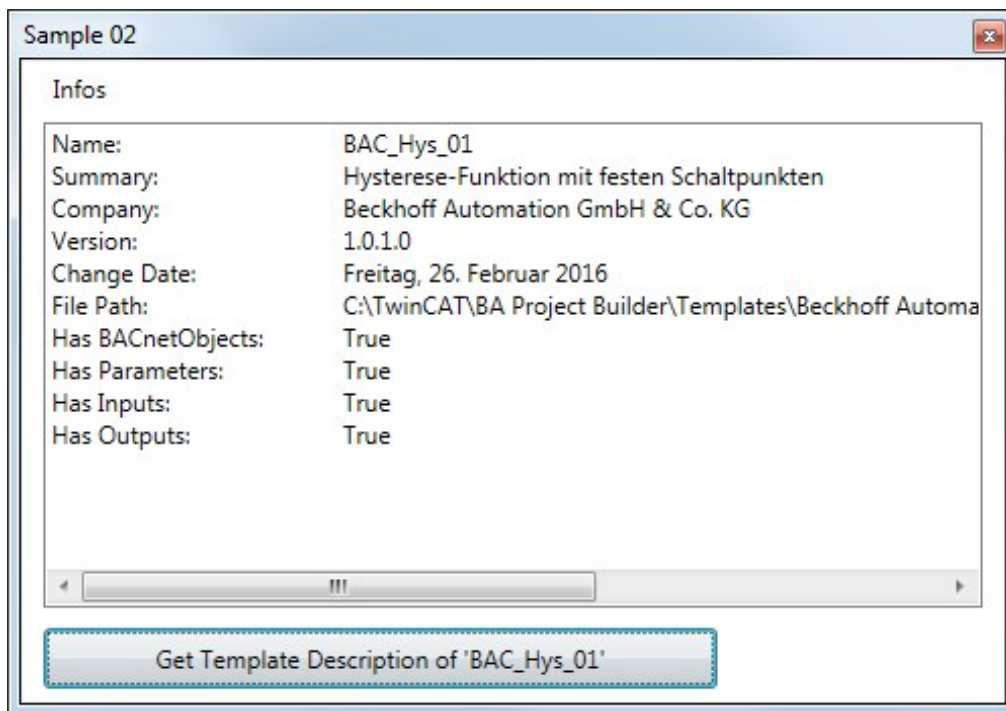
    public MainWindow(AddInEntryPoint addInEntryPoint)
    {
        InitializeComponent();
        // Code here
        this.projectBuilder = new BAProjectBuilder();
    }
}
```

```

}

private void buttonGetInfos_Click(object sender, RoutedEventArgs e)
{
    TemplateDescription td = this.projectBuilder.TemplateDescriptionProvider.GetTemplateDescription(
"BAC_Hys_01");
    List<string> items = new List<string>();
    items.Add("Name:\t\t\t" + td.Name);
    items.Add("Summary:\t\t" + td.Summary);
    items.Add("Company:\t\t" + td.Company);
    items.Add("Version:\t\t\t" + td.Version.ToString());
    items.Add("Change Date:\t\t" + td.ChangeDate.ToLongDateString());
    items.Add("File Path:\t\t\t" + td.FilePath);
    items.Add("Has BACnetObjects:\t" + td.HasBACnetObjects);
    items.Add("Has Parameters:\t\t" + td.HasParameters);
    items.Add("Has Inputs:\t\t\t" + td.HasInputs);
    items.Add("Has Outputs:\t\t\t" + td.HasOutputs);
    listBoxInfos.ItemsSource = items;
}
}

```



Download

<https://infosys.beckhoff.com/content/1031/tcba/Resources/12269795723.zip>

10.4.3.3 Beispiel 03: Auflisten aller Terminal Descriptions

Die Klasse HardwareDescriptionProvider stellt über die Read-Only Collection TerminalDescriptions alle Terminal Descriptions zur Verfügung. Mit der statischen Methode GetTerminalDecription kann die gewünschte Terminal Description ausgelesen werden.

Beispiel C#

```

public partial class MainWindow : Window
{
    BAProjectBuilder projectBuilder = null;

    public MainWindow(AddInEntryPoint addInEntryPoint)
    {
        InitializeComponent();
        // Code here
        this.projectBuilder = new BAProjectBuilder();
    }

    private void buttonGetDescriptions_Click(object sender, RoutedEventArgs e)
    {
        List<string> items = new List<string>();

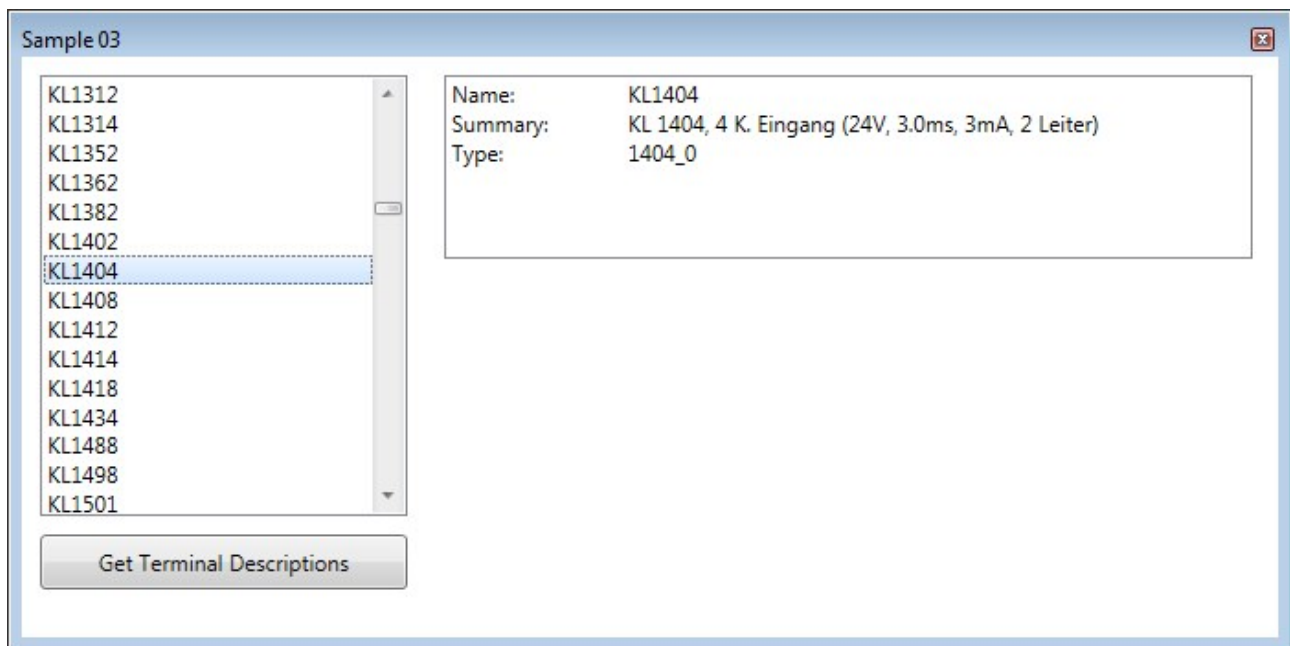
```

```

foreach (TerminalDescription td in this.projectBuilder.HardwareDescriptionProvider.TerminalDescriptions)
{
    items.Add(td.Abbreviation);
}
listBoxDescriptions.ItemsSource = items;
}

private void listBoxDescriptions_SelectionChanged(object sender, System.Windows.Controls.Selection
ChangedEventArgs e)
{
    TerminalDescription td = this.projectBuilder.HardwareDescriptionProvider.GetTerminalDescription(
(string)listBoxDescriptions.SelectedItem);
    List<string> items = new List<string>();
    items.Add("Name:\t" + td.Name);
    items.Add("Summary:\t" + td.Summary);
    items.Add("Type:\t\t" + td.Type);
    listBoxInfo.ItemsSource = items;
}
}

```



Download

<https://infosys.beckhoff.com/content/1031/tcba/Resources/12269797131.zip>

10.4.3.4 Beispiel 04: Hardware anlegen

Das folgende Beispiel legt eine Hardwarekonfiguration mit Controllern, SPS-Tasks, Buskoppler und Busklemmen an.

Beispiel C#

```

public partial class MainWindow : Window
{
    BAProjectBuilder projectBuilder = null;

    public MainWindow(AddInEntryPoint addInEntryPoint)
    {
        InitializeComponent();
        // Code here
        this.projectBuilder = new BAProjectBuilder();
    }

    private void buttonCreateHardware_Click(object sender, RoutedEventArgs e)
    {
        // add a CX2000 with the IO devices of the controller description to the solution
        //////////////////////////////////////
        ControllerDescription descriptionControllerCX2000 = this.projectBuilder.HardwareDescriptionProvider.GetControllerDescription("CX20x0");
    }
}

```

```

IController controllerCX2000 = this.projectBuilder.Solution.HardwareConfig.CreateController(desc
riptionControllerCX2000);
controllerCX2000.Name = "my CX2020";

// the CX20x0 comes with a terminal device, terminal coupler and two tasks
// get the task and change the name and the cycle time
ITask task01 = controllerCX2000.Tasks[0];
task01.Name = "Main";
task01.Interval = 45;

ITask task02 = controllerCX2000.Tasks[1];
task02.Name = "Background";
task02.Interval = 15;

// get the terminal device
IBoxDevice terminalDeviceCX2000 = controllerCX2000.Devices[0] as IBoxDevice;
terminalDeviceCX2000.Name = "my CX2000/CX5000 Terminal Device";

// get the terminal coupler
ITerminalCouplerBox terminalCouplerBoxCX2000 = terminalDeviceCX2000.Boxes[0] as ITerminalCoupler
Box;
terminalCouplerBoxCX2000.Name = "my CX2000/CX5000 Terminal Coupler";

// add terminals to the terminal coupler
ITerminal terminalKL1012 = terminalCouplerBoxCX2000.CreateTerminal(this.projectBuilder.HardwareD
escriptionProvider.GetTerminalDescription("KL1012"));
ITerminal terminalKL1404 = terminalCouplerBoxCX2000.CreateTerminal(this.projectBuilder.HardwareD
escriptionProvider.GetTerminalDescription("KL1404"));
ITerminal terminalKL2404 = terminalCouplerBoxCX2000.CreateTerminal(this.projectBuilder.HardwareD
escriptionProvider.GetTerminalDescription("KL2404"));

// link the channels of the terminals to the Main task
task01.CreateLink(terminalKL1012.Channels[0]);
task01.CreateLink(terminalKL1012.Channels[1]);

task01.CreateLink(terminalKL1404.Channels[0]);
task01.CreateLink(terminalKL1404.Channels[1]);
task01.CreateLink(terminalKL1404.Channels[2]);
task01.CreateLink(terminalKL1404.Channels[3]);

// link the channels of the KL2404 to the Background task
task02.CreateLink(terminalKL2404.Channels[0]);
task02.CreateLink(terminalKL2404.Channels[1]);
task02.CreateLink(terminalKL2404.Channels[2]);
task02.CreateLink(terminalKL2404.Channels[3]);

// add a BACnet Device
BACnetEthernetDevice bacnetDevice = controllerCX2000.CreateDevice(this.projectBuilder.HardwareDe
scriptionProvider.GetDeviceDescription("BACnet Ethernet Device Rev 12")) as BACnetEthernetDevice;
bacnetDevice.Name = "my BACnet Device";

// add a CP without(!) the IO devices of the controller description to the solution
////////////////////////////////////
ControllerDescription controllerDescriptionCP = this.projectBuilder.HardwareDescriptionProvider.
GetControllerDescription("IPC/CP ARM");
IController controllerCP = this.projectBuilder.Solution.HardwareConfig.CreateController(controll
erDescriptionCP, false);
controllerCP.Name = "my CP";

// add the task and change the name and the cycle time
ITask task03 = controllerCP.CreateTask(this.projectBuilder.HardwareDescriptionProvider.GetTaskDe
scription("Standard"));
task03.Name = "Main";
task03.Interval = 20;

// add the RT ethernet device
IBoxDevice ethernetRTDeviceCP = controllerCP.CreateDevice(this.projectBuilder.HardwareDescriptio
nProvider.GetDeviceDescription("Real Time Ethernet Protocol")) as IBoxDevice;
ethernetRTDeviceCP.Name = "my RT Ethernet Device";

// add the BK9000 coupler
ITerminalCouplerBox terminalCouplerBK9000 = ethernetRTDeviceCP.CreateBox(this.projectBuilder.Har
dwareDescriptionProvider.GetBoxDescription("BK9000 Ethernet Fieldbus Coupler")) as ITerminalCouplerB
ox;
terminalCouplerBK9000.Name = "my BK9000 Ethernet Coupler";

// add terminals to the terminal coupler
ITerminal terminalKL3204 = terminalCouplerBK9000.CreateTerminal(this.projectBuilder.HardwareDesc
riptionProvider.GetTerminalDescription("KL3204"));

```



```
// link the channels of the terminals to the Main task
task03.CreateLink(terminalKL3204.Channels[0]);
task03.CreateLink(terminalKL3204.Channels[1]);
task03.CreateLink(terminalKL3204.Channels[2]);
task03.CreateLink(terminalKL3204.Channels[3]);

this.Close();
}
}
```

Download

<https://infosys.beckhoff.com/content/1031/tcba/Resources/12269798539.zip>

11 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser [Downloadfinder](#) beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den [lokalen Support und Service](#) zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157

E-Mail: support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460

E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0

E-Mail: info@beckhoff.com

Internet: www.beckhoff.com

Mehr Informationen:
www.beckhoff.com/ts8040

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

