

SIEMENS

SIMATIC

Prozessleitsystem PCS 7 V7.0 Programmieranleitung Bausteine

Handbuch

Vorwort

Inhaltsverzeichnis

AS-Bausteine erstellen

1

Bildbausteine projektieren

2

Online-Hilfe erstellen

3

Bibliothek und Setup erstellen

4

Anhang

Beispiele: Quellcode der
Bausteine MEAS_MON,
MOTOR und VALVE

A

Glossar

Index

Sicherheitshinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.



Gefahr

bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Warnung

bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Vorsicht

mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Achtung

bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zugehörige Gerät/System darf nur in Verbindung mit dieser Dokumentation eingerichtet und betrieben werden. Inbetriebsetzung und Betrieb eines Gerätes/Systems dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieser Dokumentation sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie Folgendes:



Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden. Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Vorwort

Zweck der Programmieranleitung

Diese Programmieranleitung beschreibt, wie Sie PCS 7-konforme AS-Bausteine oder Bildbausteine erstellen.

Ein PCS 7-konformer AS-Baustein bietet im Unterschied zum reinen S7-Baustein im Wesentlichen folgende Möglichkeiten:

- Er kann Parameterwerte über einen Bildbaustein **beobachten**.
- Er kann Parameterwerte und damit das Verhalten des Bausteins über einen Bildbaustein **bedienen**.
- Er kann asynchron auftretende Ereignisse und Bausteinzustände an die OS **melden** und dort über einen Bildbaustein oder eine WinCC-Meldeliste anzeigen.

Leserkreis

Diese Programmieranleitung wendet sich an Entwickler von Automatisierungsbausteinen (AS-Bausteine) und/oder Bildbausteinen. Zusammen mit den von Siemens gelieferten leittechnischen PCS 7-Bausteinen sollen diese Bausteine in derselben Anlage durchgängig und aufeinander abgestimmt verwendet werden.

Voraussetzung

Vorausgesetzt werden Vorkenntnisse in der Entwicklung und der Anwendung von AS- und Bildbausteinen und der hierfür verwendeten Hard- und Software.

Im Weiteren werden nur die Punkte beschrieben, die für die Konformität eines Bausteins mit den PCS 7-Bausteinen notwendig sind.

Allgemeine Informationen über die Verwendung von PCS 7-Komponenten entnehmen Sie dem PCS 7-Projektierungshandbuch.

Vorgehensweise

Die Programmieranleitung bietet Ihnen einen Überblick über die einzelnen Bestandteile eines PCS 7-konformen Bausteins. Sie orientiert sich dabei an der Reihenfolge, in der Sie vorgehen, wenn Sie Funktions- und Bildbausteine entwickeln.

- Schritt für Schritt entwickeln Sie den AS-Baustein "CONTROL", einen einfachen Reglerbaustein. Dazu definieren Sie nacheinander den Bausteinkopf, die Parameter des Bausteins und seine lokalen Variablen. Dann erstellen Sie den Quellcode.
- Als Nächstes entwickeln Sie einen Bildbaustein. Diesen erstellen Sie mit dem WinCC Graphic Designer und den Elementen des Faceplate Designer.
- Abschließend entwerfen Sie für den Baustein eine Online-Hilfe und erstellen aus allen Komponenten eine lieferfähige Bibliothek MYLIB.

In jedem Kapitel werden nur die für das Verständnis der jeweils beschriebenen Punkte benötigten Teile des Beispiels angegeben. Das gesamte Beispiel des AS-Bausteins ist im Kapitel 1.11 abgedruckt.

Im Anhang sind als Beispiele für PCS 7-konforme Bausteine der Quellcode von den in der PCS 7 Library enthaltenen Bausteinen MEAS_MON, MOTOR und VALVE dargestellt. Diesen Quellcode - oder Teile daraus - können Sie als Vorlage für eigene Bausteine verwenden.

Hinweis

Die Verwendung des im Anhang enthaltenen Beispiel-Quellcodes liegt in Anwenderverantwortung. Eine Gewähr auf fehlerfreie Darstellung besteht nicht.

Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich an Ihren Siemens-Ansprechpartner in den für Sie zuständigen Vertretungen und Geschäftsstellen.

Ihren Ansprechpartner finden Sie unter:

<http://www.siemens.com/automation/partner>

Den Wegweiser zum Angebot an technischen Dokumentationen für die einzelnen SIMATIC Produkte und Systeme finden Sie unter:

<http://www.siemens.de/simatic-tech-doku-portal>

Den Online-Katalog und das Online-Bestellsystem finden Sie unter:

<http://mall.ad.siemens.com/>

Trainingscenter

Um Ihnen den Einstieg in das Prozessleitsystem SIMATIC PCS 7 und das Automatisierungssystem S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in D 90327 Nürnberg.

Telefon: +49 (911) 895-3200.

Internet: <http://www.sitrain.com>

Technical Support

Sie erreichen den Technical Support für alle A&D-Produkte wie folgt:

- Über das Web-Formular für den Support Request
<http://www.siemens.de/automation/support-request>
- Telefon: + 49 180 5050 222
- Fax: + 49 180 5050 223

Weitere Informationen zu unserem Technical Support finden Sie im Internet unter <http://www.siemens.com/automation/service>

Service & Support im Internet

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser komplettes Wissen online an.

<http://www.siemens.com/automation/service&support>

Dort finden Sie folgende Informationen:

- den Newsletter, der Sie ständig mit den aktuellsten Informationen zu Ihren Produkten versorgt.
- die für Sie richtigen Dokumente über unsere Suche in Service & Support.
- ein Forum, in dem Anwender und Spezialisten weltweit Erfahrungen austauschen.
- Ihren Ansprechpartner für Automation & Drives vor Ort.
- Informationen über Vor-Ort Service, Reparaturen, Ersatzteile. Vieles mehr steht für Sie unter dem Begriff "Leistungen" bereit.

Inhaltsverzeichnis

1	AS-Bausteine erstellen	1-1
1.1	Voraussetzungen und Vorkenntnisse.....	1-1
1.1.1	Einführung	1-1
1.1.2	So integrieren Sie den mitgelieferten Beispiel-Baustein in Ihr Projekt	1-3
1.2	Aufbau eines AS-Bausteins	1-4
1.2.1	Voreinstellungen im SCL-Compiler	1-5
1.2.2	Voreinstellungen im SIMATIC Manager	1-7
1.2.3	Bausteinkopf.....	1-9
1.2.4	Deklarationsteil	1-15
1.2.4.1	Bausteinparameter	1-15
1.2.4.2	Lokale Variablen	1-23
1.2.5	Codeteil.....	1-25
1.3	Erstlauf.....	1-26
1.4	Zeitabhängigkeit	1-27
1.5	Behandlung von asynchronen Anlauf- und Fehler-OBS	1-29
1.6	Bedienen, Beobachten und Melden	1-31
1.6.1	Meldungsunterdrückung im Anlauf	1-37
1.6.2	Meldungsunterdrückung für bestimmte Meldungen	1-39
1.6.3	Quelle übersetzen.....	1-40
1.7	Meldungsprojektierung	1-41
1.8	Anbindung von SIMATIC BATCH.....	1-45
1.9	Erstellen von CFC-Bausteintypen	1-46
1.9.1	CFC.....	1-46
1.9.2	Beispiel: CONTROL2.....	1-46
1.10	Namenskonventionen und Nummernbereich	1-48
1.11	Quellcode des Beispiels	1-49
2	Bildbausteine projektieren	2-1
2.1	Allgemeines zur Projektierung.....	2-1
2.1.1	Voraussetzungen und Vorkenntnisse	2-1
2.1.2	Phasen der Erstellung	2-2
2.1.2.1	Erstellungsweg	2-2
2.1.2.2	Entwurf des Bildbausteins	2-2
2.1.2.3	Projektierung des Bildbausteins	2-3
2.1.2.4	So testen Sie einen Bildbaustein.....	2-4
2.1.3	Bildbausteinerstellung mit dem Faceplate Designer	2-5
2.1.3.1	Faceplate Designer.....	2-5
2.1.3.2	Vorlagen des Faceplate Designer	2-5
2.1.3.3	Bausteinsymbol-Vorlagen.....	2-5
2.1.3.4	Vorlagenbilder.....	2-6
2.1.3.5	Projektierungsschritte	2-7
2.1.4	Bedienberechtigungen.....	2-8
2.1.4.1	Vergabe von Bedienberechtigungen	2-8
2.1.4.2	Bedienberechtigungen für Basiselemente projektieren.....	2-9

2.1.5	Übersicht ändern	2-10
2.1.6	Multiinstanz projektieren.....	2-11
2.1.7	Zahlenformate projektieren.....	2-15
2.1.8	Trendsicht projektieren	2-16
2.1.9	So projektieren Sie für einen AS-Bausteintyp verschiedene Bausteinsymbole und Bildbausteintypen.....	2-21
2.1.10	WebClient (Unterschiede zu WinCC)	2-22
2.1.10.1	Unterschiede bei der Projektierung von Bildbausteinen.....	2-22
2.1.10.2	Bildnamen	2-22
2.1.10.3	Groß- /Kleinschreibung bei Dateinamen	2-23
2.1.10.4	Laden von Bildern in Bildfenstern.....	2-23
2.1.10.5	Bildabwahl innerhalb von Skripten	2-24
2.1.10.6	Unterscheidung der Runtime-Umgebung WinCC <-> Web.....	2-24
2.1.10.7	Funktionsnamen in WinCC/Web	2-25
2.1.10.8	Global Script	2-26
2.1.10.9	VBS-Skript	2-26
2.1.10.10	Hinweise	2-26
2.1.11	Sprachumschaltung	2-27
2.1.12	Texte für Analogwert- und Binärwert-Bedienung aus ES.....	2-27
2.2	So arbeiten Sie mit dem Faceplate Designer	2-36
2.2.1	Beispiel: Erstellung eines neuen Bildbausteins für einen Regler	2-39
2.2.1.1	So erstellen Sie Vorlagen	2-39
2.2.1.2	Vorlagen bearbeiten	2-41
2.2.1.3	So bearbeiten Sie das Vorlagenbild @PG_REG_NEU_STANDARD.pdl	2-42
2.2.1.4	So bearbeiten Sie das Bild @PG_REG_NEU_NEUESICHT1.pdl	2-43
2.2.1.5	Dynamisierung von Bildbausteinen	2-43
2.2.1.6	So erstellen Sie eine Loop-Darstellung	2-44
2.2.1.7	So generieren Sie nachträglich eine neue Sicht	2-44
2.3	Basiselemente	2-45
2.3.1	Ablageort der Basiselemente	2-45
2.3.2	Analogwertdarstellung und Analogwertbedienung	2-45
2.3.3	Analogwertdarstellung mit "AdvancedAnalogDisplay"	2-48
2.3.4	Statischer Text.....	2-49
2.3.5	Einfache Balkendarstellung für Analogwerte.....	2-49
2.3.6	Zweifache Balkendarstellung für Analogwerte	2-50
2.3.7	Balkendarstellung horizontal	2-52
2.3.8	Balkendarstellung "Grenzwertanzeige"	2-53
2.3.9	Anzeige "Meldungsunterdrückung"	2-55
2.3.10	Anzeige "Batch Occupied"	2-56
2.3.11	Quittierung der Meldungen vom angewählten Bildbaustein	2-56
2.3.12	Anzeige-Baustein "Verriegelt" (Ventil, Motor).....	2-57
2.3.13	Sammelanzeige	2-57
2.3.14	Binärwertbedienung mit "Check Box_R"	2-58
2.3.15	Binärwertbedienung mit "Check Box_L"	2-60
2.3.16	Binärwertbedienung mit Kombinationsfeld (Combobox)	2-61
2.3.17	Binärwertbedienung mit Kombinationsfeld (3ComboBox).....	2-64
2.3.18	Binärwertbedienung mit Taste (Button) und Farbwechsel	2-66
2.3.19	Binärwertbedienung mit Taste (Button).....	2-67
2.3.20	Statusanzeige mit 2 Alternativen	2-68
2.3.21	Statusanzeige mit n Alternativen	2-69
2.3.22	Zustandsanzeige "Ventil".....	2-70
2.3.23	Zustandsanzeige "Motor".....	2-71
2.3.24	So projektieren Sie eine Permission.....	2-71
2.3.25	Taste "OpenNextFaceplate"	2-74
2.3.26	Taste "Meldungen Sperren/Freigeben"	2-76

2.3.27	Quality Code-Anzeigen.....	2-77
2.4	Skripte.....	2-78
2.5	Bitmaps.....	2-80
2.6	Bilder.....	2-83
2.7	Bildbausteine.....	2-84
2.7.1	Grunddaten der Vorlagenbilder.....	2-84
2.7.1.1	@PG_%Type%.pdl.....	2-84
2.7.1.2	@PG_%TYPE%.....	2-86
2.7.1.3	@PG_%Type%_%View%.pdl.....	2-87
2.7.2	Globale Sichten.....	2-88
2.7.2.1	Meldesicht.....	2-88
2.7.2.2	Batch-Sicht.....	2-88
2.7.2.3	Trendsicht.....	2-89
2.7.3	CTRL_PID.....	2-90
2.7.3.1	CTRL_PID: Sichten.....	2-90
2.7.3.2	CTRL_PID: Standardsicht.....	2-90
2.7.3.3	CTRL_PID: Wartungssicht.....	2-92
2.7.3.4	CTRL_PID: Parametersicht.....	2-93
2.7.3.5	CTRL_PID: Grenzsicht.....	2-95
2.8	Bausteinsymbole.....	2-97
2.8.1	Prozessbedienung.....	2-97
2.8.2	Vorlagenbilder @@PCS7Typicals.pdl und @Template.pdl.....	2-97
2.8.3	Bausteinsymbole im Bild @@PCS7_Typicals.....	2-99
2.8.4	Eigenschaften der Bausteinsymbole.....	2-100
2.8.4.1	Allgemeine Eigenschaften.....	2-100
2.8.4.2	CTRL_PID.....	2-101
2.8.4.3	CTRL_S.....	2-102
2.8.4.4	DOSE.....	2-104
2.8.4.5	FMCS_PID/FMT_PID.....	2-105
2.8.4.6	ELAP_CNT.....	2-106
2.8.4.7	MEAS_MON.....	2-107
2.8.4.8	SWIT_CNT.....	2-108
2.8.4.9	RATIO_P.....	2-108
2.8.4.10	OP_A.....	2-109
2.8.4.11	OP_A_LIM.....	2-109
2.8.4.12	OP_A_RJC.....	2-109
2.8.4.13	VALVE.....	2-110
2.8.4.14	VAL_MOT.....	2-110
2.8.4.15	MOTOR.....	2-111
2.8.4.16	MOT_SPED.....	2-112
2.8.4.17	MOT_REV.....	2-113
2.8.4.18	INTERLOK.....	2-113
2.8.4.19	OP_D3.....	2-114
2.8.4.20	OP_D.....	2-114
2.8.4.21	OP_TRIG.....	2-114
2.8.4.22	DIG_MON.....	2-115

3	Online-Hilfe erstellen	3-1
3.1	Voraussetzungen.....	3-1
3.2	Aufbau der Hilfedatei	3-1
3.3	Aufbau der Registrierungsdatei	3-3
3.4	Besonderheiten für die Hilfe-Erstellung von Muster-SFCs.....	3-5
4	Bibliothek und Setup erstellen	4-1
4.1	Voraussetzung.....	4-1
4.2	So erstellen Sie eine Bibliothek	4-1
4.3	So erstellen Sie ein Setup	4-2
A	Beispiele: Quellcode der Bausteine MEAS_MON, MOTOR und VALVE	A-1
A.1	MEAS_MON	A-1
A.2	MOTOR.....	A-7
A.3	VALVE	A-15
	Glossar	Glossar-1
	Index	Index-1

1 AS-Bausteine erstellen

1.1 Voraussetzungen und Vorkenntnisse

1.1.1 Einführung

Voraussetzungen

Für die Erstellung der AS-Bausteine sind folgende Voraussetzungen notwendig:

- PCS 7 ab V6.1 auf einer CPU S7-4xx
- Für die Erstellung der Bausteine benötigen Sie die folgenden Softwarepakete:
 - STEP 7 Basis ab V5.2
 - SCL-Compiler ab V5.1 SP4
 - CFC ab V6.0

Weitere Informationen zu SCL

AS-Bausteine für PCS 7 werden mit der Programmiersprache SCL erstellt.

Weitere Informationen zu SCL finden Sie in folgenden Dokumentationen:

- Online-Hilfe im SIMATIC Manager:
Wählen Sie hierzu den Menübefehl **Aufruf von Hilfen zu Optionspaketen > Bausteine programmieren mit S7-SCL**.
- Handbuch *S7-SCL Erste Schritte*
- Handbuch *S7-SCL für S7-300 und S7-400*

Die Handbücher finden Sie unter **Start > Simatic > Dokumentation**.

Hinweise zu Anwender-Datenbausteinen

Das Erstellen von Datenbausteinen (DB) ist nicht Bestandteil dieser Anleitung. Wenn Sie außer den Funktionsbausteinen (FB) auch eigene Datenbausteine (DB) erstellen, beachten Sie Folgendes:

- Verwenden Sie eine DB-Nummer im Bereich von 1 bis 60. Dieser Nummernbereich wird im CFC für andere Applikationen reserviert. Damit wird sichergestellt, dass die vom CFC beim Übersetzen erzeugten Instanz-DBs nicht mit den Nummern der Anwender-DBs kollidieren.
- Wenn Sie den voreingestellten Nummernbereich im CFC zu einem späteren Zeitpunkt verändern, kann nur noch ein Gesamtladen im Stoppzustand der CPU ausgeführt werden. Sie ändern den voreingestellten Nummernbereich im CFC über den Menübefehl **Extras > Einstellungen für Übersetzen/Laden**.

1.1.2 So integrieren Sie den mitgelieferten Beispiel-Baustein in Ihr Projekt

Pfad

Der hier beschriebene Baustein "CONTROL" wird als SCL-Quelle CONTx.SCL auf der PCS 7 Toolset-DVD mitgeliefert und je nach Sprache in einem der folgenden Pfade installiert:

- Deutsch: ...\\STEP7\\EXAMPLES\\zdt25_01\\S7_CONTA.SCL
- Englisch: ...\\STEP7\\EXAMPLES\\zen25_01\\S7_CONTB.SCL
- Englisch, bei französischer Installation:
...\\STEP7\\EXAMPLES\\zfr25_01\\S7_CONTC.SCL.

Vorgehen

Um den Beispiel-Baustein in Ihr Projekt zu übernehmen, gehen Sie folgendermaßen vor:

1. Markieren Sie in Ihrem Projekt den Quellordner und wählen Sie den Menübefehl **Einfügen > Externe Quelle...** .
Das Dialogfeld "Externe Quelle einfügen" wird geöffnet.
2. Markieren Sie in der Baumansicht den Ablageort der SCL-Quelle "S7_CONTA.SCL" oder "S7_CONTB.SCL" und klicken Sie auf die Schaltfläche "Öffnen".
Die SCL-Quelle befindet sich nun im Quellordner und muss mit dem SCL-Compiler übersetzt werden.
3. Stellen Sie vor dem Übersetzen sicher, dass sich der Baustein "OP_A_LIM" (FB 46) im Bausteinordner Ihres Projekts befindet. Wenn dies nicht der Fall ist, kopieren Sie ihn aus der PCS 7-Bibliothek in Ihr Projekt.
4. Überprüfen Sie die Voreinstellung des SCL-Compilers (siehe Kapitel 1.2.1).
5. Nehmen Sie in der Symboltabelle folgende Einstellungen vor:
 - Tragen Sie den symbolischen Namen "CONTROL" ein.
 - Tragen Sie die Adresse FB 501 ein.
 - Speichern Sie die Einträge.

Verwenden Sie die Nummern ab 500 aufwärts, um Kollisionen mit den Nummern der PCS 7-Standardbausteine zu vermeiden.

Siehe auch Abschnitt "Eintrag in die Symboltabelle" in Kapitel 1.2.2

6. Doppelklicken Sie auf die SCL-Quelle "S7_CONTA" oder "S7_CONTB".
Die gewünschte SCL-Quelle wird geöffnet.
7. Starten Sie die Übersetzung und beenden Sie den SCL-Compiler nach fehlerfreiem Durchlauf.

Der Beispiel-Baustein "FB 501" befindet sich nun im Bausteinordner Ihres Projekts.

1.2 Aufbau eines AS-Bausteins

Ein AS-Baustein ist im PCS 7-Umfeld nur dann lauffähig, wenn er bestimmte formale und inhaltliche Kriterien erfüllt. Die nachfolgenden Kapitel beschreiben, was Sie zur Erfüllung dieser Kriterien beachten und unternehmen müssen.

Das untenstehende Blockbild zeigt den prinzipiellen Aufbau des Bausteins "CONTROL" (FB 501). Die einzelnen Elemente des Bausteins werden in den entsprechenden Kapiteln genauer erklärt.

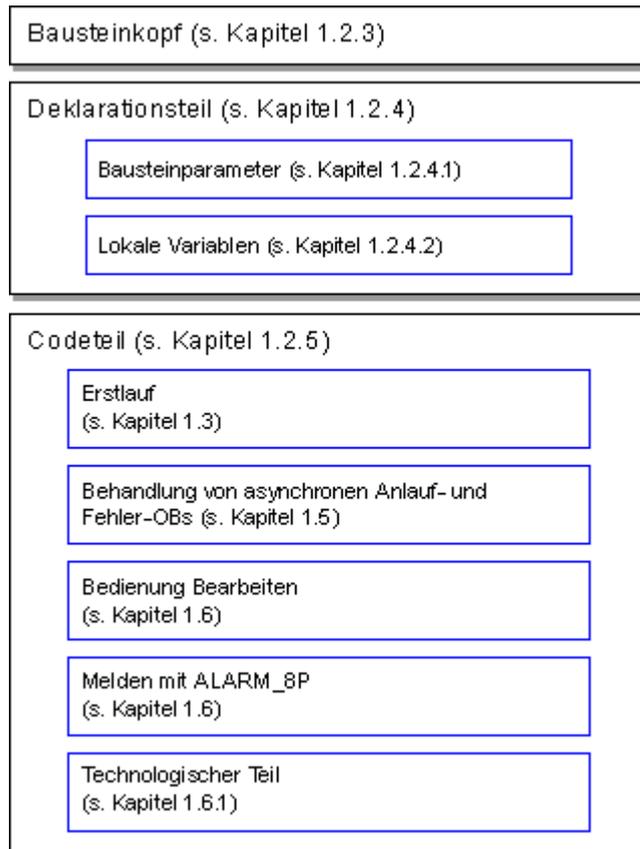


Bild 1-1: Aufbau des Beispiel-Bausteins "CONTROL" (FB 501)

Funktionsbaustein oder Funktion

Wenn Ihr Baustein Werte speichern, melden oder bedien- und beobachtbar sein soll, müssen Sie ihn als Funktionsbaustein (FB) realisieren. Ein FB hat ein Gedächtnis in Form eines Datenbausteins (DB), auch Instanz-Datenbaustein genannt.

Wenn Sie die angeführten Eigenschaften nicht benötigen, können Sie Ihren Baustein auch als Funktion (FC) realisieren.

1.2.1 Voreinstellungen im SCL-Compiler

Voreinstellungen für "Bausteine erzeugen"

Im SCL-Compiler sind unter **Extras > Einstellungen > Bausteine erzeugen** folgende Optionen einstellbar:

- **Bausteine überschreiben**

- Überschreibt bereits existierende Bausteine im Ordner "Bausteine" eines S7-Programms, wenn beim Übersetzungsvorgang Bausteine mit der gleichen Bezeichnung erzeugt werden.
- Überschreibt beim Laden Bausteine mit gleichem Namen, die bereits im Zielsystem vorhanden sind.

Wenn diese Option nicht aktiviert ist, müssen Sie eine Meldung bestätigen, um den Baustein zu überschreiben.

- **Warnungen anzeigen**

Legt fest, ob nach einem Übersetzungslauf zusätzlich zu den Fehlern auch Warnungen gemeldet werden.

- **Fehler vor Warnungen anzeigen**

Legt fest, dass in der Meldung die Fehler vor den Warnungen aufgelistet werden.

- **Referenzdaten erzeugen**

Erzeugt beim Erzeugen eines Bausteins automatisch Referenzdaten.

Um die Referenzdaten später zu erzeugen oder zu aktualisieren, wählen Sie den Menübefehl **Extras > Referenzdaten**.

- **Systemattribut "S7_server" berücksichtigen**

Berücksichtigt bei der Erzeugung eines Bausteins das Systemattribut für Parameter "S7_server". Sie vergeben dieses Attribut, wenn der Parameter für die Verbindungs- oder Meldungsprojektierung relevant ist. Der Parameter enthält die Verbindungs- und/oder die Meldungsnummern.

- **Bausteinnummer automatisch erzeugen**

Erzeugt automatisch für die in der Quelle enthaltenen Bausteinnamen die Bausteinnummer und trägt sie in der Symboltabelle ein.

Hinweis

(für dieses Beispiel)

Aktivieren Sie in jedem Fall das Optionskästchen "Systemattribut "S7_server" berücksichtigen", da dieser Baustein Meldungen enthält. Wenn Sie dieses Optionskästchen nicht aktivieren, wird beim Importieren und/oder beim Einfügen des Bausteins in einen CFC-Plan der Vorgang mit Fehlermeldung abgebrochen.

Voreinstellungen für "Compiler"

Im SCL-Compiler können Sie unter **Extras > Einstellungen > Compiler** die folgenden Optionskästchen aktivieren oder deaktivieren:

- "Feldgrenzen überwachen"
- "Debug Info erstellen"
- "OK Flag setzen"

Wir empfehlen die restlichen Optionskästchen immer zu aktivieren. Weitere Informationen zu den einzelnen Optionen finden Sie im SCL-Handbuch.

Beachten Sie bei der Entscheidung, ob Sie diese Optionen aktivieren oder deaktivieren, die folgenden Punkte:

- **Feldgrenzen überwachen**

Bei Einsatz von Arrays im Programm wird zur Laufzeit überprüft, ob Laufindex und Feldlängen-Vereinbarung im zulässigen Bereich liegen. Im Fehlerfall wird das OK-Flag beeinflusst und der ENO-Ausgang zurückgesetzt. Diese Überprüfung ist eine sehr laufzeitintensive Aktion.

Wenn Sie Arrays verwenden, lassen Sie die Option nur solange aktiviert, bis Ihr Baustein ausreichend getestet ist und es sicher ist, dass Index und Feldlänge zusammenpassen.

- **Debug Info erstellen**

Mit dieser Option können Sie einen Testlauf mit dem Debugger ausführen, nachdem das Programm übersetzt und in die CPU geladen worden ist. Der Speicherbedarf des Programms und die Ablaufzeiten im AS erhöhen sich jedoch durch diese Option. Aktivieren Sie diese Option daher nur während der Testphase des Bausteins und nicht in der Lieferversion.

- **OK Flag setzen**

Das OK-Flag ist eine systeminterne Variable. Wenn während der Ausführung einer Operation ein Fehler auftritt, z. B. Überlauf bei arithmetischen Operationen, wird das OK-Flag vom System beeinflusst und an den Ausgang ENO durchgereicht. Diese Überprüfung ist sehr laufzeitintensiv. Aus diesem Grund ist es sinnvoll, das automatische Setzen des OK-Flags abzuschalten und stattdessen unzulässige Operationen/Bereichsüberschreitungen im Bausteinalgorithmus selbst abzufangen. Im Fehlerfall können Sie das OK-Flag dann explizit setzen, wenn Sie den ENO-Ausgang zur Weiterverschaltung benutzen wollen. Dies wird vom System übernommen und kostet keine Performance, da der Zustand des OK-Flags vom System stets an den Ausgang durchgereicht wird.

1.2.2 Voreinstellungen im SIMATIC Manager

Sprache für die Anwender-Schnittstelle

Bei PCS 7-konformen AS-Bausteinen muss die Schnittstelle zum Anwender, z. B. Parameternamen und Kommentare, in englischer Sprache sein. Die Bausteine selbst können Sie in jeder beliebigen Landessprache entwickeln.

Auswahl der Landessprache

Wenn Sie Ihre Bausteine in einer Bibliothek zusammenfassen wollen (vgl. Kapitel 4.2), muss auch die "Landessprache" auf Englisch eingestellt sein. Dadurch erhalten die einzelnen Kataloge Ihrer Bibliothek PCS 7-konforme Namen, z. B. "Sources", "Symbols" und "Blocks". Wählen Sie hierzu im SIMATIC Manager über den Menübefehl **Extras > Einstellungen > Sprache** "English" als Landessprache und als Mnemonic.

Eintrag in der Symboltabelle

Der Name des Bausteins, der im Bausteinkopf eingetragen wird, muss in der Symboltabelle als symbolischer Name eingetragen sein.

Um den Namen des Bausteins in den Bausteinkopf einzutragen, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie im S7-Programm auf "Symbole".
Die Symboltabelle wird geöffnet.
2. Tragen Sie in der Spalte "Symbol" den symbolischen Namen ein, hier "CONTROL".
3. Tragen Sie in der Spalte "Adresse" eine FB-Nummer ein, hier "FB 501".
4. Tragen Sie in der Spalte "Kommentar" einen Text ein, der die Funktion des Bausteins näher bezeichnet (max. 80 Zeichen).

Der Kommentar ist eine Ergänzung zum symbolischen Bausteinnamen, da dieser allein in der Regel nicht genügend über den Verwendungszweck und über die Funktionalität des Bausteins aussagt.

Dieser Bausteinkommentar ist identisch mit dem Symbolkommentar, der im SIMATIC Manager in der Detailsicht oder den Objekteigenschaften des Bausteins angezeigt wird.

Beim Einfügen des Bausteins in einen CFC-Plan wird der Text als Bausteinkommentar im Kopf des Instanzbausteins dargestellt und kann, unabhängig vom Eintrag in der Symboltabelle, instanzspezifisch geändert werden.

Hinweis

Je nach Darstellungsbreite der CFC-Bausteine wird nur eine entsprechend begrenzte Anzahl Zeichen dargestellt. Der gesamte Kommentar wird aber temporär in der Kurzinformation angezeigt, die Sie erhalten, wenn Sie den Mauszeiger auf den Bausteinkopf positionieren.

5. Sichern und schließen Sie die Symboltabelle.

Weitere Informationen hierzu finden Sie in Abschnitt 1.10, Namenskonventionen und Nummernbereich.

1.2.3 Bausteinkopf

Baustein-Attribute

Der Bausteinkopf enthält die Verwaltungsinformationen des Bausteins (im Weiteren Baustein-Attribute genannt). Diese Attribute werden von den einzelnen PCS 7-Tools für verschiedene Zwecke genutzt. Sie werden im SIMATIC Manager in den Objekteigenschaften des Bausteins angezeigt und können dort auch geändert werden (siehe dazu Attribut KNOW_HOW_PROTECT).

Bausteinkopf des Beispiel-Baustein

Ausschnitt des Beispiel-Bausteins:

```
//Copyright (C) Siemens AG 1999. All Rights Reserved. Confidential
(*****
KURZBESCHREIBUNG:
Dieser Baustein gibt Ihnen ein Beispiel für die Entwicklung eines
PCS 7-konformen AS-Bausteins.

Er realisiert einen einfachen Regelalgorithmus nach der Formel:
Stellgröße = Verstärkungsfaktor * (Sollwert - Istwert)
Überschreitet der Prozesswert die obere Alarmgrenze, wird der Fehlerausgang
QH_ALM gesetzt. Zudem wird eine Meldung an die OS mit ALARM_8P generiert.
Mit der Variablen M_SUP_AH kann die Meldung unterdrückt werden.
Unterschreitet der Prozesswert die untere Alarmgrenze, wird der Fehlerausgang
QL_ALM gesetzt. Zudem wird eine Meldung an die OS mit ALARM_8P generiert.
Mit der Variablen M_SUP_AL kann die Meldung unterdrückt werden.

Der Baustein unterstützt SIMATIC BATCH und besitzt die dafür nötigen
Parameter BA_EN, BA_NA, BA_ID, OCCUPIED und STEP_NO.

Zur Verdeutlichung einer Zeitverzögerung enthält der Baustein zusätzliche
Eingänge:
Der Ausgang SUPP_OUT folgt dem Eingang SUPP_IN nach einer parametrierbaren
Wartezeit SUPPTIME.
..*****
//Ersteller: ABC      Datum: 13.08.00      Vers.:1.00
//Geändert:         Datum: 18.11.03      Vers.:
//Änderung:

//*****
// Bausteinkopf
//*****

FUNCTION_BLOCK "CONTROL"
TITLE =          'CONTROL'

{ // Liste der Systemattribute
S7_tasklist:= 'OB80,OB100';//Baustein wird bei Zeitfehler u. Neustart aufgerufen
S7_m_c:= 'true';          // Baustein ist bedien- und beobachtbar
S7_alarm_ui:= '1'        // Einstellung PCS7-Melddialog ('0'=Standard-Melddialog)
}
AUTHOR:  ABC
NAME:    CONTROL
VERSION: '0.02'
FAMILY:  XYZ
KNOW_HOW_PROTECT
```

Objekteigenschaften des übersetzten Beispiel-Bausteins

Die folgenden Bilder zeigen die Objekteigenschaften des übersetzten Beispiel-Bausteins mit Verweisen auf die jeweils betroffenen Attribute des Bausteinkopfs.

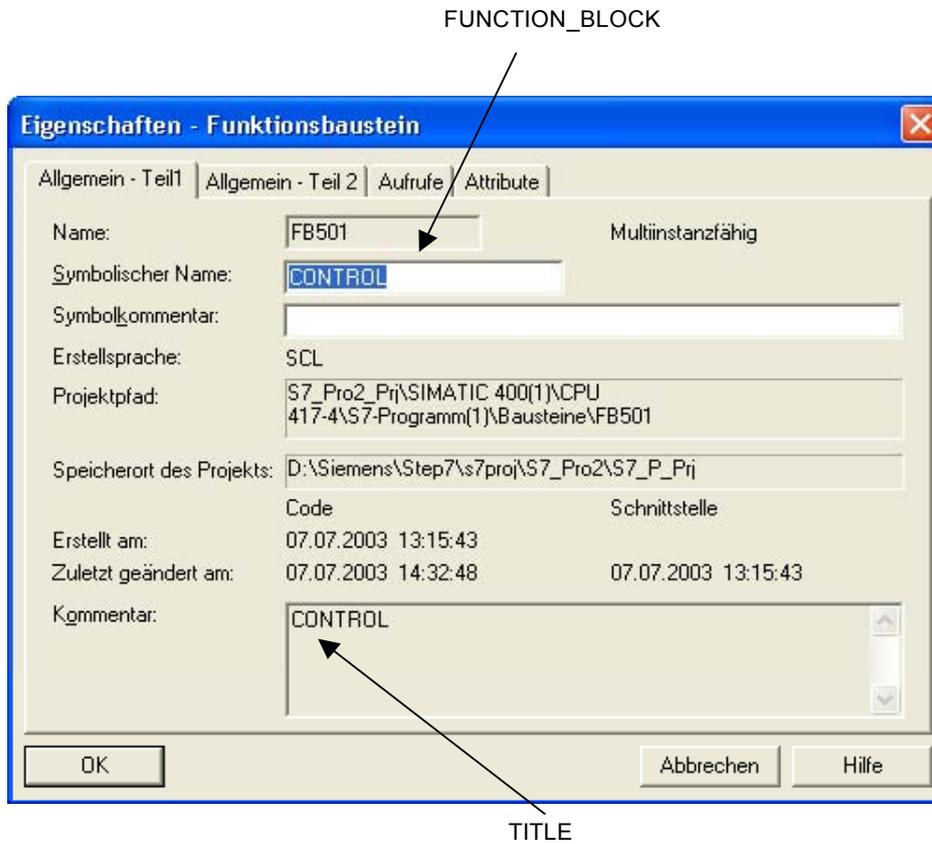


Bild 1-2: Objekteigenschaften des Bausteins (Allgemein - Teil 1)

- **FUNCTION_BLOCK**

Hier legen Sie den Namen des Bausteins mit maximal 8 Zeichen fest. Dieser Name wird in den Objekteigenschaften des Bausteins sowie in der Detaildarstellung des SIMATIC Managers und im Katalog des CFC angezeigt. Ordnen Sie diesem Namen vor dem Übersetzen des Bausteins in der Symboltabelle eine Bausteinnummer zu.

- **TITLE**

Diese Information wird unter PCS 7 nicht ausgewertet, sie wird jedoch im SIMATIC Manager in den Objekteigenschaften des Bausteins im Kommentarfeld angezeigt. Kommentare, die direkt unterhalb dieses Attributs angegeben werden, werden ebenfalls in den Objekteigenschaften des Bausteins im Kommentarfeld angezeigt. Alle anderen Kommentare im Bausteinkopf können nur mit dem SCL-Editor angezeigt werden.

Wir empfehlen, hier eine Kurzbeschreibung des Bausteins einzutragen.

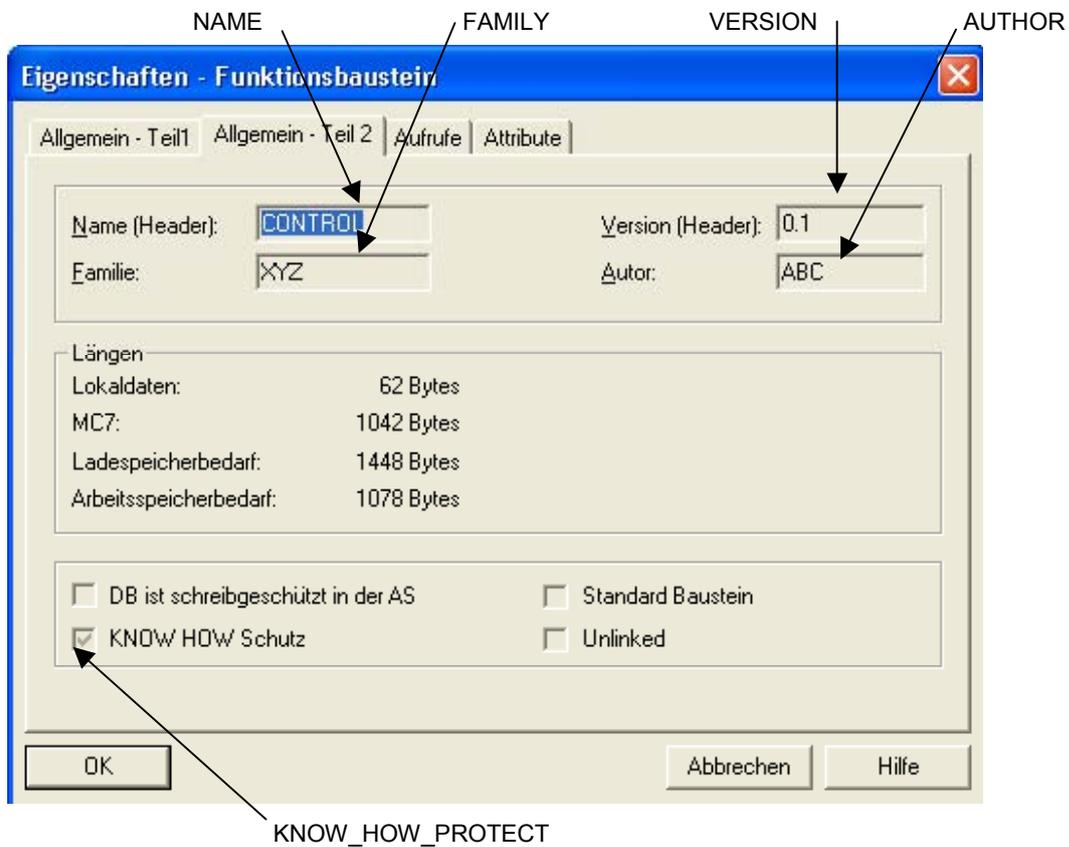


Bild 1-3: Objekteigenschaften des Bausteins (Allgemein - Teil 2)

- **NAME**

Tragen Sie hier denselben Namen für den Baustein ein, den Sie bereits unter "FUNCTION_BLOCK" festgelegt haben.

Bei Verwendung einer Online-Hilfe sind "NAME" und "FAMILY" Teilschlüssel zur Suche des Hilfetexts für diesen Baustein in der Hilfedatei.

- **VERSION**

Tragen Sie hier eine Versionskennung von 0.00 bis 15.15 ein.

- **FAMILY**

Wenn Sie Ihre Bausteine zu einer eigenen Bibliothek zusammenfassen und innerhalb dieser Bibliothek in verschiedene Gruppen einordnen wollen, geben Sie hier den Gruppennamen für diesen Baustein an. Verwenden Sie maximal 8 Zeichen für den Namen.

Bei Verwendung einer Online-Hilfe sind "FAMILY" und "NAME" Teilschlüssel zur Suche des Hilfetexts für diesen Baustein in der Hilfedatei (siehe auch Kap. 3.3).

- **AUTHOR**

Dieses Attribut enthält im Normalfall den Namen oder die Abteilung des Bausteinerstellers. Bei PCS 7-konformen Bausteinen wird es noch für zwei weitere Punkte genutzt:

- Wenn Sie Ihre Bausteine zu einer Bibliothek zusammenfassen wollen, geben Sie hier den gemeinsamen Namen für alle Bausteine dieser Bibliothek an. Verwenden Sie maximal 8 Zeichen für den Namen.
- Bei Verwendung einer Online-Hilfe wird über diesen Namen die passende Hilfedatei gesucht.

- **KNOW_HOW_PROTECT**

Mit diesem Attribut können Sie den Algorithmus und die Attribute des Bausteins gegen Einsichtnahme und Änderung schützen. Das Setzen dieses Attributs hat folgende Auswirkungen:

- Die Attribute des Bausteins sind im SIMATIC Manager in den Objekteigenschaften des Bausteins lesbar, sie können aber nicht geändert werden.
- Außerhalb Ihres Projekts lässt sich der Baustein selbst ohne die entsprechende Quelle nur noch mit dem AWL-Editor öffnen und nicht mehr mit SCL. Dabei werden nur die Bausteinparameter angezeigt. Innerhalb Ihres Projekts wird der SCL-Compiler gestartet.

- **Liste der Systemattribute für Bausteine**

Mit Hilfe der Systemattribute bereiten Sie einen Baustein für die Verbindung mit der OS vor. Z. B. definiert das Attribut "S7_m_c", ob der Baustein für eine OS-relevant ist. Wenn das der Fall ist, werden dafür in der OS notwendige interne Datenstrukturen angelegt. Zudem können Sie mit den Systemattributen den Einbau des Bausteins in einen CFC-Plan steuern. Z. B. wird mit dem Attribut "S7_tasklist" festgelegt, in welche OBs der Baustein automatisch eingebaut wird.

Tabelle 1-1: Systemattribute für PCS 7-konforme Bausteine

Systemattribut	Bedeutung	Default
S7_tasklist	Enthält eine Liste der OBs (z. B. Fehler- oder Anlauf-OBs), in die der Baustein vom CFC eingebaut werden soll.	wird nicht mehrfach eingebaut
S7_m_c	Definiert, ob der Baustein von einer OS aus bedient oder beobachtet werden kann.	false
S7_alarm_ui	Kennung für Meldeserver: S7_alarm_ui:='0' Standard-Meldedialog S7_alarm_ui:='1' PCS 7-Meldedialog	0
S7_tag	Wenn dieses Systemattribut den Wert 'false' hat, wird der Baustein nicht in die Messstellenliste der OS eingetragen und er erhält auf der OS keine Anwahl für "Loop in Alarm". Dies ist sinnvoll für Bausteine, die nur melden, aber keinen Bildbaustein besitzen. Wenn das Systemattribut nicht vorhanden ist und wenn der Baustein auch das Systemattribut "S7_m_c" hat, wird er in die Messstellenliste eingetragen.	true
S7_driver	Kennung für einen Signal vorverarbeitenden Treiberbaustein, der mit der CFC-Funktion im SIMATIC Manager "Baugruppentreiber erzeugen" automatisch mit dem zugehörigen Baustein verschaltet wird. Werte: 'chn', 'f'	
S7_hardware	Kennung für baugruppenspezifischen Treiberbaustein, der mit der CFC-Funktion im SIMATIC Manager "Baugruppentreiber erzeugen" automatisch in einen CFC-Plan eingefügt, parametrisiert und verschaltet wird. Werte: 'subnet', 'rack', 'sm', 'im', 'fm'	
S7_read_back	Definiert, ob die Bausteininstanz für die Funktion "Plan > Rücklesen" im CFC vorgesehen wird. Wenn dieses Systemattribut den Wert 'false' hat, können die Parameter des Instanzbausteins nicht rückgelesen werden.	true

Die Systemattribute werden im SIMATIC Manager in den Objekteigenschaften des Bausteins im Register "Attribute" angezeigt. Dort können Sie die Systemattribute ändern, wenn der Baustein nicht schreibgeschützt ist. Der Baustein ist nicht schreibgeschützt, wenn das Attribut "KNOW_HOW_PROTECT" im Bausteinkopf deaktiviert ist.

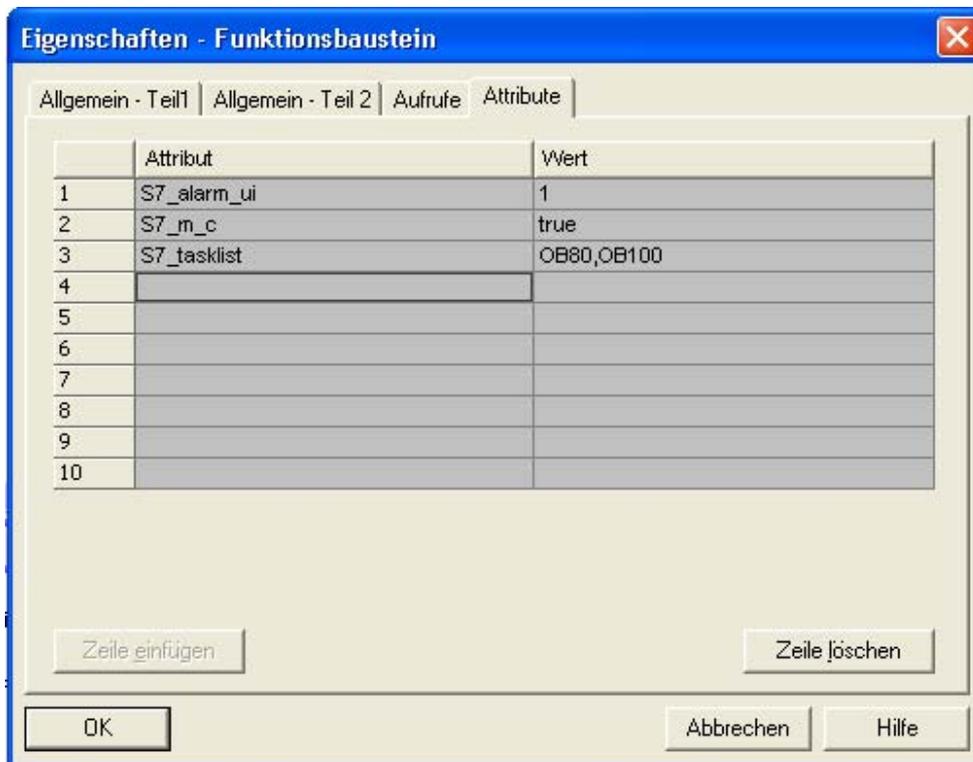


Bild 1-4: Systemattribute des Bausteins

Hinweis

Eine vollständige Liste der Systemattribute erhalten Sie über die Kontext-Hilfe und/oder über das Hilfethema "Attribute für Bausteine und Parameter".

1.2.4 Deklarationsteil

1.2.4.1 Bausteinparameter

Schnittstelle des Bausteins

Die Bausteinparameter definieren die Schnittstelle des Bausteins zu anderen Bausteinen und zu den Bedien- und Beobachtungswerkzeugen, wie CFC und WinCC .

Parametertypen

Es gibt die folgenden Parametertypen:

- **Eingangsparameter**

Die Festlegung als Eingangsparameter ist für PCS 7-konforme Bausteine in folgenden Fällen erforderlich:

- Sie wollen Parameterwerte von einem anderen Baustein holen.
- Sie wollen Parameter von der OS aus bedienen.
- Sie wollen die Darstellung eines Bildbausteins auf der OS mit Hilfe von Parametern beeinflussbar machen (z. B. Grenzen für Darstellbereiche).
- Sie wollen Parameter für Testzwecke vom CFC aus bedienbar machen.
- Sie benötigen Parameter zur Erzeugung von Meldungen, z. B. Message Event-ID des ALARM_8P-Bausteins.
- Wenn Sie im Baustein eine stoßfreie Umschaltung zwischen Eingangswerten vom Programm (AS) und Bedienwerten (OS) benötigen, können die Eingangsparameter vom Bausteinalgorithmus sowohl gelesen als auch zurückgeschrieben werden (siehe Durchgangsparameter). Im Gegensatz zu den Durchgangsparametern werden die Eingangsparameter nicht auf den verschalteten Ausgangsparameter zurückgeschrieben.

- **Ausgangsparameter**

Die Festlegung als Ausgangsparameter ist für PCS 7-konforme Bausteine in folgenden Fällen erforderlich:

- Sie wollen Parameterwerte an einen anderen Baustein weitergeben.
- Sie wollen Parameter von der OS aus beobachten.
- Sie wollen Parameter für Testzwecke vom CFC aus beobachten.

- **Durchgangsparameter**

Durchgangsparameter können vom Bausteinalgorithmus sowohl gelesen als auch zurückgeschrieben werden. Die Festlegung als Durchgangsparameter ist für PCS 7-konforme Bausteine erforderlich, wenn Sie im Baustein eine stoßfreie Umschaltung zwischen Eingangswerten vom Programm (AS) und Bedienwerten (OS) benötigen. Für die Umsetzung dieser Funktionalität benötigen Sie 3 Parameter:

- einen Eingangsparameter zum Umschalten
- einen Eingangsparameter für den verschalteten Wert
- einen Durchgangsparameter für den bedienten Wert.
Dieser Parameter muss ein Durchgangsparameter sein, da der verschaltete Wert immer auf den bedienten Wert zurückgeschrieben werden muss. Auf diese Weise ist gewährleistet, dass die Umschaltung vom verschalteten auf den bedienten Wert stoßfrei erfolgt.

Kommentare für Parameter

Wenn Sie die Bausteinparameter mit Kommentaren versehen wollen, tragen Sie diese durch "/" getrennt hinter der jeweiligen Parameterdefinition ein.

Die Kommentare werden im CFC in den Objekteigenschaften des jeweiligen Anschlusses sowie in den Objekteigenschaften des Bausteins im Register Anschlüsse angezeigt. Dort können Sie die Kommentare ändern, unabhängig vom Know-How-Schutz, der über das Attribut "KNOW_HOW_PROTECT" im Bausteinkopf gesetzt wird.

Systemattribute für Parameter

Bausteinparameter können ebenso wie der Baustein selbst mit Hilfe von Systemattributen weiter spezifiziert werden.

Sie können Folgendes spezifizieren:

- Wie der Parameter auf der OS dargestellt wird.
Beispiel: "S7_unit" definiert die Einheit des Parameters, z. B. Liter. Den an diesem Attribut angegebenen Text können Sie in Ihren Bildbaustein einblenden.
- Ob und wie der Parameter im CFC behandelt wird.
Beispiel: "S7_visible" definiert, ob der Parameter im CFC-Plan angezeigt wird oder nicht.

Tabelle 1-2: Systemattribute für Parameter für PCS 7-konforme Bausteine

Systemattribut	Betrifft	Bedeutung	Default
S7_sampletime	Zeitverhalten	Wenn ein Parameter dieses Systemattribut besitzt, wird er automatisch mit der Zykluszeit des aufrufenden Weck-OBs parametrierbar. Dazu muss bei der Übersetzung des CFC-Plans das Optionskästchen "Aktualisierung der Abtastzeit" aktiviert sein (vgl. Kapitel 1.4).	false
S7_dynamic	CFC	Wenn ein Parameter dieses Systemattribut besitzt, wird er im Testbetrieb des CFC automatisch zum Testen angemeldet.	false
S7_edit	CFC	Definiert, ob der Parameter für die spätere Bearbeitung im SIMATIC Manager in der Tabelle "Parameter/Signale bearbeiten" vorgesehen werden soll ohne den CFC-Plan zu öffnen.	false
S7_link	CFC	Definiert, ob der Parameter im CFC verschaltbar ist	true
S7_param	CFC	Definiert, ob der Parameter im CFC parametrierbar ist	true
S7_visible	CFC	Ein Parameter, bei dem dieses Systemattribut auf 'false' gesetzt ist, wird im CFC-Plan nicht dargestellt.	true
S7_qc	CFC, B&B	Dieses Attribut kennzeichnet Parameter, die neben dem Prozesswert über Quality Code verfügen. Der Quality Code muss dabei unmittelbar als nächster Parameter nach dem Prozesswert im Interface deklariert sein. Der Datentyp des Prozesswerts ist beliebig, der Datentyp des Quality Codes muss vom Typ "BYTE" sein.	false
S7_contact	SFC	Parameter gehört zu einer Anschlussgruppe	false
S7_m_c	B&B	Definiert, ob der Parameter von einer OS aus bedient oder beobachtet werden kann	false
S7_shortcut	B&B	Enthält eine maximal 16 Zeichen lange Bezeichnung des Parameters. Diese Bezeichnung, z. B. "Sollwert", kann auf der OS in einem Bildbaustein ausgegeben werden.	
S7_string_0	B&B	Dieses Systemattribut ist nur für Eingangsparameter oder Durchgangsparameter vom Datentyp BOOL sinnvoll. Es enthält einen maximal 16 Zeichen langen Text, der in einem Bildbaustein als Bedientext ausgegeben werden kann, z. B. 'Ventil öffnen'. Durch die Ausführung dieser Bedienung erhält der Parameter den Wert 0.	
S7_string_1	B&B	Dieses Systemattribut ist nur für Eingangsparameter oder Durchgangsparameter vom Datentyp BOOL sinnvoll. Es enthält einen max. 16 Zeichen langen Text, der in einem Bildbaustein als Bedientext ausgegeben werden kann, z. B. 'Ventil schließen'. Durch die Ausführung dieser Bedienung erhält der Parameter den Wert 1.	
S7_unit	B&B	Enthält den max. 16 Zeichen langen Namen der Maßeinheit des Parameters. Die Einheit, z. B. "mbar", kann im CFC am Bausteinanschluss angezeigt werden.	
S7_server	Server	Interface-Parameter ist einem Server zugeordnet. Meldeserver: S7_server:='alarm_archiv'	kein Serveraufruf
S7_a_type	Meldeserver	Interface-Parameter ist Meldenummerneingang von der Meldeart x oder Archivnummerneingang	

Verwendung und Änderung der Systemattribute

Bei Verwendung der Systemattribute "S7_string_0" und "S7_string_1" müssen Sie noch Folgendes beachten:

Der angegebene Wert kann im Bildbaustein als Bedientext ausgegeben werden. Wird die Bedienung ausgeführt, wird der Wert 0 oder 1 an das AS übertragen. Im CFC wird der aktuelle Wert des Parameters ausgegeben. Diese Wertausgabe können Sie ebenfalls mit Hilfe des Systemattributs anpassen.

Dazu teilen Sie den Wert des Systemattributs in zwei Teile und trennen diese beiden durch ein Gleichheitszeichen, z. B. S7_string_1 := 'Suppress HH =YES'. Der CFC erkennt das Gleichheitszeichen und ersetzt die Wertausgabe am Parameter durch den Teil, der hinter dem Gleichheitszeichen steht. In diesem Fall wird statt des Wertes 1 das Wort "YES" ausgegeben.

Im CFC werden maximal 8 Zeichen ausgegeben, auch wenn Sie mehr angeben. Im Bedienprotokoll wird immer der ganze Text ausgegeben, in diesem Fall "Suppress HH =YES".

Die Systemattribute werden im CFC in den Objekteigenschaften des jeweiligen Anschlusses angezeigt und können dort auch geändert werden. Die folgenden Bilder zeigen die Objekteigenschaften von Parametern vom Datentyp BOOL und von Parametern, die nicht vom Datentyp BOOL sind (Bild 1-6). Zudem enthalten die Bilder Verweise auf die betroffenen Systemattribute.

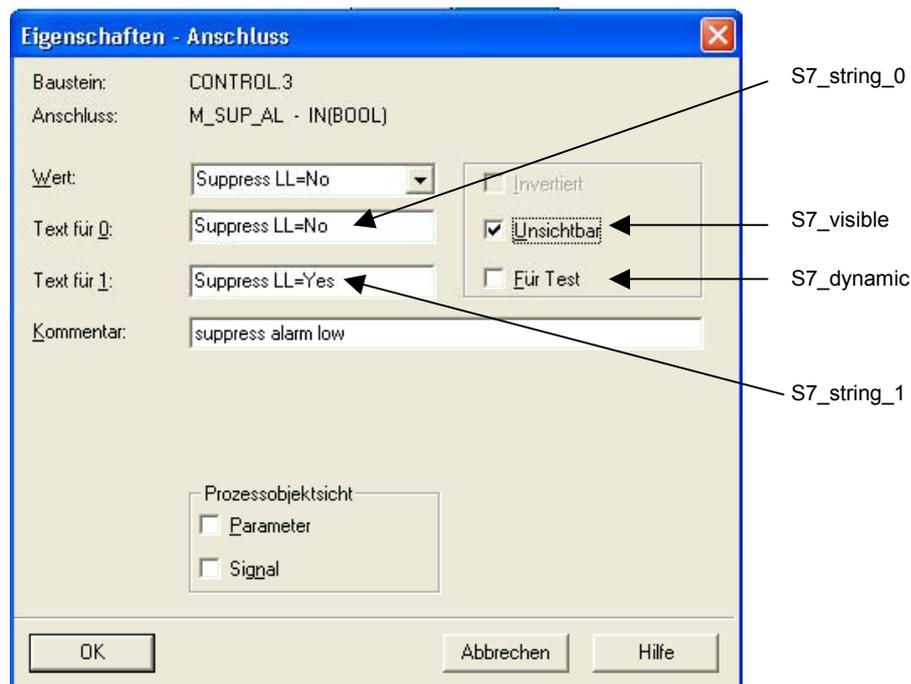


Bild 1-5: Objekteigenschaften von Parametern vom Datentyp BOOL

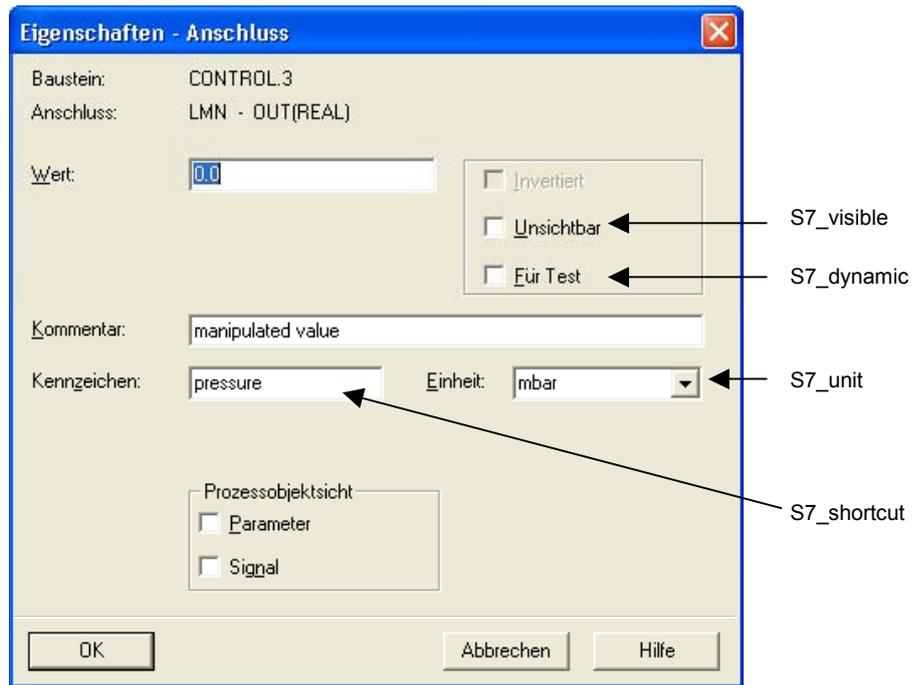


Bild 1-6: Objekteigenschaften von Parametern, die nicht vom Datentyp BOOL sind

Codierung der Bausteinparameter

Der folgende Auszug des Beispiel-Bausteins zeigt die Codierung der Bausteinparameter:

```

//*****
// Deklarationsteil: Bausteinparameter
//*****
VAR INPUT
SAMPLE_T {S7_sampletime:= 'true'; // Param. der Baustein-Abtastzeit
          // (Zyklus der Task)
          S7_visible:= 'false'; // Parameter ist unsichtbar
          S7_link:= 'false' // Parameter nicht verschaltbar
          } :REAL := 1; // Verzögerungszeit [s] (Vorbesetzung 1 Sek)

H_ALM {S7_m_c := 'true';
       S7_visible:= 'false';
       S7_link := 'false'} :REAL :=100; // oberer Grenzwert Alarm
                                     // (Vorbesetzung 100)

L_ALM {S7_m_c := 'true'; // Parameter ist B&B-fähig
       S7_visible:= 'false'; // Parameter ist unsichtbar
       S7_link := 'false' // und nicht verschaltbar
       } :REAL := 0; // unterer Grenzwert Alarm (Vorbesetzung 0)

M_SUP_AL {S7_visible:= 'false';
          S7_link:= 'false';
          S7_m_c:= 'true';
          S7_string_0:= 'Suppress LL=No'; // Bedientext für Wert (M_SUP_AL)= 0
          S7_string_1:= 'Suppress LL=Yes' // Bedientext für Wert (M_SUP_AL)= 1
          } :BOOL := 0; // Meldungsunterdrückung unterer Alarm

M_SUP_AH {S7_visible:= 'false';
          S7_link:= 'false';
          S7_m_c:= 'true';
          S7_string_0:= 'Suppress HH=No';
          S7_string_1:= 'Suppress HH=Yes'
          } :BOOL:= 0; // Meldungsunterdrückung oberer Alarm

SP_OP_ON {S7_visible:= 'false';
          S7_dynamic:= 'true' // CFC in Test/IBS: Anzeige aktueller Wert in AS)
          } :BOOL := 1; // 1 = Bedienfreigabe für Sollwert-Eingabe

SPBUMPON {S7_visible:= 'false';
          S7_link:= 'false';
          S7_m_c:= 'true';
          S7_string_0:= 'SP bumpless=Off';
          S7_string_1:= 'SP bumpless=On'
          }
          :BOOL := 1; // 1 = Stoßfreiheit für Sollwert

SP_EXTON {S7_visible:= 'false';
          S7_dynamic:= 'true' // CFC in Test/IBS: Anzeige aktueller Wert in AS)
          }
          :BOOL := 1; // 1: Externer Sollwert ein

SP_EXT {S7_dynamic:= 'true'}
        :REAL := 0; // Externer Sollwert

SP_HLM {S7_visible:= 'false';
        S7_link:= 'false';
        S7_m_c:= 'true';
        S7_shortcut:= 'SP high limit'; // Text(max 16 Zeichen) zur Anzeige auf OS
        S7_unit:= '' // Einheit (max 16 Zeichen)
        } :REAL := 100; // oberer Bediengrenzwert für Sollwert

```

```

SP_LLM    {S7_visible:='false';
           S7_link:='false';
           S7_m_c:='true';
           S7_shortcut:='SP low limit';
           S7_unit:='' }
           :REAL := 0;           // unterer Bediengrenzwert für Sollwert

PV_IN {S7_dynamic:='true';
       S7_m_c:='true';
       S7_unit:='%'} : REAL := 0; // Prozesswert (zu Begleitwert_PR04)

GAIN    {S7_link:='false';
         S7_edit:='para';           // Parametrierung im IEA
         S7_m_c:='true';
         S7_shortcut:='Gain';
         S7_unit:='' } :REAL := 1; // Proportionalbeiwert

EV_ID   {S7_visible:='false';
         S7_link:='false';
         S7_param:='false';        // Parameter im CFC nicht parametrierbar
         S7_server:='alarm_archiv'; // Vergabe der Meldenummer durch Server
         S7_a_type:='alarm_8p'     // Baustein meldet mit ALARM_8P
         } :DWORD := 0;          // Meldungsnummer

// Parameter für SIMATIC BATCH
STEP_NO {S7_visible := 'false';
         S7_m_c := 'true'} :DWORD; // Batch Schrittnummer
BA_ID {S7_visible := 'false';
       S7_m_c := 'true'} :DWORD; // Batch laufende Chargennummer
BA_EN {S7_visible := 'false';
       S7_m_c := 'true' } :BOOL := 0; // Parameter im CFC-Plan unsichtbar
// Parameter ist B&B-fähig
// Batch Belegt-Freigabe
BA_NA {S7_visible := 'false';
       S7_m_c := 'true'} :STRING[32] := ''; // Batch Chargenbezeichnung

OCCUPIED {S7_visible := 'false';
          S7_m_c := 'true'} :BOOL := 0; // Batch Belegt-Kennung

RUNUPCYC {S7_visible:='false';
           S7_link:='false'} :INT := 3; // Anzahl Erstlaufzyklen
SUPPTIME :REAL := 0; // Verzögerungszeit
SUPP_IN :REAL := 0; // Eingangswert für Verzögerungszeit
END_VAR

VAR_OUTPUT
LMN {S7_shortcut:='pressure'; // Bezeichnung des Parameters auf OS
     S7_unit := '%'; // Einheit des Parameters
     S7_m_c := 'true' // beobachtbar
     } :REAL; // Stellwertausgang

QH_ALM :BOOL := false; // 1 = Oberer Grenzwert Alarm hat angesprochen
QL_ALM :BOOL := false; // 1 = Unterer Grenzwert Alarm hat angesprochen

QSP_HLM {S7_visible:='false';
         S7_dynamic:='true'} : BOOL := 0; // 1= Sollwertausgang nach oben
         // begrenzt
QSP_LLM {S7_visible:='false';
         S7_dynamic:='true'} : BOOL := 0; // 1 = Sollwertausgang nach unten
         // begrenzt
Q_SP_OP {S7_visible:='false';
         S7_dynamic:='true';
         S7_m_c:='true'} : BOOL := 0; // Status: 1 = Bedienfreigabe für Sollwert

QOP_ERR {S7_visible:='false';
         S7_dynamic:='true'} : BOOL := 0; // 1 = Bedienfehler
QMSG_ERR {S7_visible:='false';
          S7_dynamic:='true'} : BOOL := 0; // ALARM_8P: Meldungsfehler

```

```
MSG_STAT {S7_visible:='false';
          S7_dynamic:='true'} : WORD := 0; // Meldungsfehler-Informationen

MSG_ACK  {S7_visible:='false';
          S7_dynamic:='true'} : WORD := 0; // Meldungen quittieren

SUPP_OUT :REAL := 0; // Ausgangswert für Verzögerungszeit
SP       {S7_dynamic:='true';
          S7_m_c:='true'} : REAL := 0; // Aktiver Sollwert

END_VAR
VAR_IN_OUT
SP_OP {S7_visible:='false';
       S7_link:='false';
       S7_m_c:='true';
       S7_shortcut:='Setpoint';
       S7_unit:='%'} : REAL := 0; // Bedieneingabe Sollwert

// freibelegbare Meldebegleitwerte des ALARM_8P

AUX_PR05 {S7_visible := 'false'} : ANY; // Begleitwert 5
AUX_PR06 {S7_visible := 'false'} : ANY; // Begleitwert 6
AUX_PR07 {S7_visible := 'false'} : ANY; // Begleitwert 7
AUX_PR08 {S7_visible := 'false'} : ANY; // Begleitwert 8
AUX_PR09 {S7_visible := 'false'} : ANY; // Begleitwert 9
AUX_PR10 {S7_visible := 'false'} : ANY; // Begleitwert 10

END_VAR
```

1.2.4.2 Lokale Variablen

Zusätzliche Variablen, die nicht als Bausteinparameter nach außen gegeben werden, müssen Sie als lokale Variablen definieren.

Es gibt zwei Sorten von lokalen Variablen:

- Statische Variablen
- Temporäre Variablen

Statische Variablen

Im Gegensatz zu temporären Variablen behalten statische Variablen ihren Wert über mehrere Aufrufe des Bausteins hinweg bei. Der Wert ändert sich erst dann, wenn Sie ihn im Bausteinalgorithmus ändern.

Für PCS 7-konforme Bausteine sind diese Variablen vor allem dann wichtig, wenn Sie in Ihrem Baustein bereits vorhandene eigene oder Standard-Bausteine aufrufen wollen. In diesem Fall müssen Sie einen **Multiinstanzbaustein** implementieren. Dazu definieren Sie innerhalb der statischen Variablen eine Instanz des aufgerufenen Bausteins.

Die aufgerufenen Bausteine müssen zum fehlerfreien Übersetzen des aufrufenden Bausteins im Bausteinordner des S7-Programms vorhanden sein.

Wenn Sie Parameter des aufgerufenen Bausteins nach außen sichtbar und verschaltbar machen wollen, müssen Sie diese im Bausteinalgorithmus von den Parametern oder in die Parameter Ihres Bausteins kopieren. Die Parameter des aufgerufenen Bausteins selbst sind nach außen nicht sichtbar.

Multiinstanzen

Beispiele zu Multiinstanzanwendungen können Sie dem Kapitel zu den CFC-Bausteintypen bzw. dem entsprechenden SCL-Code im Beispielprojekt entnehmen.

Hinweis

Aufgerufene SFBs und SFCs, z. B. SFC 6 (RD_SINFO) oder SFB 0 (CTU), werden beim Compilieren des aufrufenden Bausteins automatisch in der Standard Library gesucht und in Ihr S7-Programm eingefügt.

Wenn sich die aufgerufenen FBs und der aufrufende Baustein in derselben Bibliothek befinden, dann werden die aufgerufenen FBs beim Einfügen des aufrufenden Bausteins in einen CFC-Plan in den Bausteinordner kopiert. Sonst müssen Sie sie selbst kopieren.

Temporäre Variablen

Temporäre Variablen haben nur während **eines** Bausteinaufrufs Gültigkeit, d.h. sie müssen bei jedem Bausteinaufruf neu berechnet werden.

Für PCS 7-konforme Bausteine müssen Sie hierbei keine Besonderheiten beachten.

Ausschnitt des Beispiel-Bausteins:

```

//*****
// Deklarationsteil: temporäre Variablen
//*****

VAR TEMP
// Startinfo: Struktur mit Info für den OB, der den Baustein gerade
// aufgerufen hat
TOP_SI:   STRUCT
  EV_CLASS :BYTE;
  EV_NUM   :BYTE;
  PRIORITY :BYTE;
  NUM      :BYTE;
  TYP2_3   :BYTE;
  TYP1     :BYTE;
  ZI1      :WORD;
  ZI2_3    :DWORD;
END_STRUCT;

// Startinfo: Struktur mit Info für den letzten aufgerufenen Anlauf-OB
START_UP_SI: STRUCT
  EV_CLASS :BYTE;
  EV_NUM   :BYTE;
  PRIORITY :BYTE;
  NUM      :BYTE;
  TYP2_3   :BYTE;
  TYP1     :BYTE;
  ZI1      :WORD;
  ZI2_3    :DWORD;
END_STRUCT;

S7DT  :DATE_AND_TIME;      // Lokale Zeitvariable
DUMMY :INT;                // Hilfsvariable
END_VAR

```

1.2.5 Codeteil

Der Codeteil enthält den eigentlichen Algorithmus des Bausteins.

Für PCS 7-konforme Bausteine können Sie dadurch Folgendes realisieren:

- die reinen technologischen Funktionen des Bausteins
- Eigenschaften, um z. B. asynchron auftretende Ereignisse und Bausteinzustände an die OS zu melden und dort über einen Bildbaustein oder eine WinCC-Meldeliste anzuzeigen

1.3 Erstlauf

Implementierung eines Erstlaufteils

Beim ersten Aufruf Ihres Bausteins müssen Sie im Regelfall verschiedene Parameter initialisieren. Zudem kann es je nach technologischer Funktion Ihres Bausteins weitere Tätigkeiten geben, die Ihr Baustein nur einmalig ausführen muss. Wenn dies bei Ihrem Baustein so ist, müssen Sie einen Erstlaufteil implementieren.

Dazu definieren Sie eine Variable vom Datentyp BOOL, z. B. sbRESTART. Sie können die Variable als statische Variable realisieren.

Ihr Baustein wird nicht nur bei einem Neustart das erste Mal durchlaufen, sondern auch beim Neuladen des Bausteins im Betriebszustand RUN der CPU. Deshalb müssen Sie den Erstlaufteil typischerweise in den zyklischen Teil Ihres Bausteins integrieren. Wenn es notwendig ist, können Sie dadurch die Bearbeitung des Erstlaufs auch auf mehrere Aufrufzyklen des Bausteins ausdehnen.

```
//*****
// Abhängigkeiten vom aufrufenden OB
//*****

// Auslesen der Startinfo mittels SFC6 (RD_SINFO)
DUMMY := RD_SINFO (TOP_SI := TOP_SI, START_UP_SI := START_UP_SI);

IF sbRESTART THEN
// Erstlauf
TOP_SI.NUM := 100;      // Erstlauf als Neustart ausführen
sbRESTART := FALSE;   // Rücksetzen Erstlauf
END_IF;
// Aus welchem OB wurde der Baustein aufgerufen ?

CASE WORD_TO_INT(BYTE_TO_WORD(TOP_SI.NUM)) OF

//*****
// Behandlung von Fehler-OBs
//*****

// OB80: Zeitfehler
80:
QH_ALM := 0;          // Fehlerausgänge zurücksetzen
QL_ALM := 0;

//*****
// Anlauf
//*****

// OB100: Neustart
100:
QH_ALM := 0;          // Fehlerausgänge zurücksetzen
QL_ALM := 0;
siRUNUPCNT := RUNUPCYC; // RUNUPCYC-Wert abspeichern
ELSE
....
```

1.4 Zeitabhängigkeit

Programmierung einer Zeitabhängigkeit

Definieren Sie einen Eingangsparameter vom Datentyp REAL, z. B. SAMPLE_T, wenn für Ihren Baustein Folgendes gilt:

- Er wird in einer äquidistanten Zeitalarzebene bearbeitet.
- Er muss zur Ausführung zeitabhängiger Tätigkeiten die Länge des Zeitintervalls auswerten, z. B. bei Reglerbausteinen.

An diesem Eingangsparameter kann die Länge des Zeitintervalls angegeben werden.

Parametrieren Sie diesen Parameter je nach dem Weckalarm-OB um, in dem Ihr Baustein aufgerufen wird. Damit ist gewährleistet, dass Ihr Bausteinalgorithmus immer mit der richtigen Zeit arbeitet.

Wenn Sie diesen Parameter mit dem Systemattribut "S7_sampletime" versehen und dieses auf 'true' setzen, wird er vom CFC automatisch auf den Wert gesetzt, der für den aufrufenden OB passend ist. Eine eventuelle Untersetzung wird dabei ebenfalls berücksichtigt. Versehen Sie den Parameter auch mit den Systemattributen "S7_visible" und "S7_link" und setzen Sie diese auf 'false'. Damit wird der Parameter im CFC unsichtbar und nicht verschaltbar und sein Wert kann vom Anwender nicht versehentlich geändert werden.

Die automatische Belegung des Parameters durch den CFC funktioniert jedoch nur dann, wenn beim Übersetzen des Programms das Optionskästchen "Update sampling time" aktiviert ist.

Realisierung der Zeitabhängigkeit

Der folgende Ausschnitt des Beispiel-Bausteins zeigt die Realisierung einer solchen Zeitabhängigkeit: Mit Hilfe des Parameters SUPPTIME kann am Baustein eine Wartezeit parametrisiert werden. Änderungen am Eingang SUPP_IN werden nach Ablauf dieser Wartezeit auf den Ausgang SUPP_OUT durchgeschaltet.

```

/*****
// Deklarationsteil: Bausteinparameter
/*****
VAR_INPUT

    SAMPLE_T {S7_sampletime:= 'true' // Parameter Baustein-Abtastzeit
                // (=Zyklus der Task)
                S7_visible:= 'false'; // Parameter ist unsichtbar
                S7_link:= 'false' // Parameter nicht verschaltbar
            } :REAL := 1; // Verzögerungszeit [s] (Vorbesetzung 1 Sekunde)
    ....
END_VAR

/*****
// Deklarationsteil: statische Variablen
/*****
VAR
    ....
    sSUPP_IN :REAL := 0; // Altwert des Delay-Beispieleingangs
    ACT_TIME :REAL := 0; // Zeitzähler
    ....
END_VAR

VAR_OUTPUT
    ....
    SUPP_OUT :REAL := 0; // Ausgangswert für Verzögerungszeit
    ....
END_VAR

/*****
// Technologischer Teil
/*****

    IF (SUPP_IN <> sSUPP_IN) THEN
        ACT_TIME := SUPPTIME; // Zeitzähler initialisieren
        sSUPP_IN := SUPP_IN;
    END_IF;

    IF (ACT_TIME > 0) THEN // Wenn Wartezeit noch nicht abgelaufen ist
        ACT_TIME := ACT_TIME-SAMPLE_T; // Wartezeit herunterzählen
    ELSE
        SUPP_OUT := SUPP_IN; // Eingang durchschalten
    END_IF;
    ....

```

1.5 Behandlung von asynchronen Anlauf- und Fehler-OBs

Ein asynchrones Ereignis

Beim Auftreten eines asynchronen Ereignisses, z. B. Neustart, Ziehen/Stecken und Rack-Ausfall, wird vom AS ein asynchroner OB aufgerufen. Wenn Ihr Baustein auf ein solches Ereignis reagieren soll, müssen Sie Ihren Baustein in den betreffenden OB einbauen und im Bausteinalgorithmus feststellen, ob ein derartiges Ereignis aufgetreten ist.

Einbau in asynchrone OBs

Damit Ihr Baustein in bestimmte OBs eingebaut wird, verwenden Sie das Systemattribut "S7_tasklist". Als Wert tragen Sie alle OBs ein, die Sie benötigen, z. B. S7_tasklist := 'OB80,OB100'. Beim Einfügen des Bausteins in einen CFC-Plan wird der Baustein somit vom CFC automatisch neben dem aktuellen Weckalarm-OB auch in alle mit "S7_tasklist" angegebenen OBs eingebaut.

Prüfen des aufrufenden OBs

Um zu überprüfen, in welchem OB Ihr Baustein gerade läuft, müssen Sie im Bausteinalgorithmus den SFC 6 (RD_SINFO) aufrufen. Dieser liest die Startinfo Ihres Bausteins aus und liefert damit Informationen über den gerade aktiven OB (Parameter TOP_SI) und den zuletzt aufgerufenen Anlauf-OB (Parameter START_UP_SI).

Die beiden Parameter sind zwei identisch aufgebaute Strukturen, die Sie beide in Ihren temporären Variablen definieren müssen. Die einzelnen Strukturelemente haben dabei folgende Bedeutung:

Tabelle 1-3: Parameter TOP_SI und START_UP_SI

Strukturelement	Datentyp	Bedeutung
EV_CLASS	BYTE	Bit 0 bis 3: Ereigniskennung Bit 4 bis 7: Ereignisklasse
EV_NUM	BYTE	Ereignisnummer
PRIORITY	BYTE	Nummer der Ablaufebene
NUM	BYTE	Nummer des aufrufenden OBs
TYP2_3	BYTE	Datenkennung für ZI2_3
TYP1	BYTE	Datenkennung für ZI1
ZI1	WORD	Zusatzinfo 1
ZI2_3	DWORD	Zusatzinfo 2_3

Die Strukturelemente entsprechen inhaltlich den temporären Variablen des aufrufenden OBs. Diese können je nach OB andere Namen und Datentypen haben. Das bedeutet, dass Sie anhand der jeweiligen OB-Beschreibung (siehe Handbuch *STEP 7 - System- und Standardfunktionen*) die einzelnen Strukturelemente einander zuordnen und entsprechend auswerten müssen. Die folgende Tabelle bzw. der Ausschnitt des Beispiel-Bausteins zeigen dies am Beispiel des OB 80 (Zeitfehler).

Tabelle 1-4: Zuordnung der Elemente der Startinfo TOP_SI zu den temporären Variablen des OB 80

TOP_SI/STARTUP_SI		OB 80	
Strukturelement	Datentyp	Temporäre Variable	Datentyp
EV_CLASS	BYTE	OB80_EV_CLASS	BYTE
EV_NUM	BYTE	OB80_FLT_ID	BYTE
PRIORITY	BYTE	OB80_PRIORITY	BYTE
NUM	BYTE	OB80_OB_NUMBR	BYTE
TYP2_3	BYTE	OB80_RESERVED_1	BYTE
TYP1	BYTE	OB80_RESERVED_2	BYTE
ZI1	WORD	OB80_ERROR_INFO	WORD
ZI2_3	DWORD	OB80_ERR_EV_CLASS	BYTE
		OB80_ERR_EV_NUM	BYTE
		OB80_OB_PRIORITY	BYTE
		OB80_OB_NUM	BYTE

Hinweis

Jeder OB enthält in seinen temporären Variablen noch Datum und Uhrzeit des Aufrufs. Diese sind jedoch nicht in der mit dem SFC 6 gelesenen Startinfo enthalten.

PCS 7-konforme Bausteine werden nicht in den Wiederanlauf (OB 101) oder Kaltstart (OB 102) eingebaut.

Der folgende Ausschnitt des Beispiel-Bausteins zeigt die jeweilige Behandlung des OBs:

```

//*****
// Codeteil
//*****

CASE WORD_TO_INT(BYTE_TO_WORD(TOP_SI.NUM)) OF

//*****
// Behandlung von Fehler-OBs
//*****

    // OB 80: Zeitfehler
    80:
        QH_ALM := 0;           // Fehlerausgänge zurücksetzen
        QL_ALM := 0;

//*****
// Anlauf
//*****

    // OB100: Neustart
    100:
        QH_ALM := 0;           // Fehlerausgänge zurücksetzen
        QL_ALM := 0;
        siRUNUPCNT := RUNUPCYC; // RUNUPCYC-Wert abspeichern
ELSE

```

1.6 Bedienen, Beobachten und Melden

Bedienen und Beobachten

Damit die Parameter eines Bausteins von der OS aus **bedient** und **beobachtet** werden können, muss der Baustein für diese Anbindung an die OS entsprechend vorbereitet werden. Das betrifft sowohl die gewünschten Parameter als auch den Baustein selbst.

Bedienungen

Wenn Sie einen Parameterwert nur von der OS aus bedienen wollen, benötigen Sie dazu einen Durchgangs- oder Eingangsparameter für den bedienten Wert mit dem Systemattribut "S7_m_c".

Wenn Sie dagegen einen Parameterwert wahlweise von einem anderen Baustein holen oder von der OS bedienen wollen und der Umschaltvorgang vom verschalteten auf den bedienten Wert stoßfrei erfolgen soll, benötigen Sie dazu insgesamt drei Parameter:

- einen Eingangsparameter zum Umschalten zwischen Bedienung und Verschaltung
- einen Eingangsparameter für den verschalteten Wert
- einen Durchgangsparameter für den bedienten Wert mit dem Systemattribut "S7_m_c"
Dieser Parameter muss ein Durchgangsparameter sein, weil der verschaltete Wert für das stoßfreie Umschalten vom Bausteinalgorithmus auf den bedienten Wert zurückgeschrieben werden muss, solange der verschaltete Wert angewählt ist.

Verwenden Sie für alle Bedienfunktionen die Bedienbausteine der Bibliothek "PCS 7 Library V70" und deren korrespondierende Bedienmethode auf der OS. Damit sind alle notwendigen Verriegelungen und die wahlweise stoßfreie Umschaltung zwischen bedientem Wert und verschaltetem Wert vorhanden, z. B. für Hand-/Automatikumschaltung. Die Bedienbausteine können Sie mit der Multiinstanztechnik in Ihren Baustein einbauen.

Bausteine für begrenzendes Bedienverhalten

Für begrenzendes Bedienverhalten können Sie bei PCS 7 folgende Bausteine verwenden:

- OP_A_LIM (**operation analog limited**)
- OP_A_RJC: Weist bei Grenzwertverletzung die Bedieneingabe ab
- OP_A: Verwenden Sie diesen Baustein, wenn Sie keine Grenzwertprüfung benötigen.

Hinweis

AS-Baustein und Bildbaustein laufen asynchron zueinander, d.h. bei einer Bedienung durch den Bildbaustein wird der Bedienwert in den Instanz-DB des AS-Bausteins geschrieben und zu einem späteren Zeitpunkt vom AS-Baustein ausgewertet. Da sich zu diesem Zeitpunkt bereits die maßgeblichen Grenzen geändert haben können, sollten Sie den Bedienwert sowohl auf dem AS als auch auf der OS auf Fehler prüfen.

Bausteine für binäre Bedieneingaben

Für binäre Bedieneingaben stehen die Bausteine OP_D (FB 48), OP_D3 (FB 49) und OP_TRIG (FB 50) der PCS 7 Library V61 zur Verfügung. Weitere Informationen hierzu finden Sie in der Online-Hilfe.

Definition einer Bedieneingabe

Der folgende Ausschnitt zeigt die Definition einer Bedieneingabe:

```

//*****
// Bedieneingabe Sollwert SP_OP (Real-Wert) oder verschalteter Sollwert SP_EXT
//*****

// Multiinstanz-Aufruf OP_A_LIM
// (Bedeutung der Parameter siehe Online-Hilfe OP_A_LIM)

OP_A_LIM_1(U := SP_OP, U_HL:= SP_HLM, U_LL:= SP_LLM, OP_EN:= SP_OP_ON,
BTRACK:= SPBUMPON, LINK_ON:= SP_EXTON, LINK_U:= SP_EXT);

OK := OK AND ENO; //Enable Out des OP_A_LIM in OK-Flag des Bausteins
// übernehmen
Q_SP_OP := OP_A_LIM_1.QOP_EN; // 1: Freigabe Bedieneingabe SP

QOP_ERR := OP_A_LIM_1.QOP_ERR; // 1: Bedienfehler
QSP_HLM := OP_A_LIM_1.QVHL; // 1: Begrenzung Obergrenze
QSP_LLM := OP_A_LIM_1.QVLL; // 1: Begrenzung Untergrenze
SP := OP_A_LIM_1.V; // wirksamer Sollwert
    
```

Meldungen

Wenn Ihr Baustein Meldungen und/oder Ereignisse an die OS senden soll, können Sie in den statischen Variablen eine Multiinstanz eines Alarm-Bausteins definieren. Das Meldungs- und Quittierverhalten sowie die Übergabe von Begleitwerten wird durch die Eigenschaften der CPU (wählbar "quittiergetriggertes Melden") und des eingebauten Alarm-Bausteins bestimmt.

Fertige Alarm-Bausteine sind als SFBs in der "Standard Library" enthalten. Das sind z. B. folgende Bausteine:

ALARM	SFB 33	Überwachung eines Signals mit 1 bis 10 Begleitwerten mit Quittierungsanzeige
ALARM_8	SFB 34	Überwachung von bis zu 8 Signalen
ALARM_8P	SFB 35	Überwachung von bis zu 8 Signalen mit 1 bis 10 Begleitwerten
NOTIFY	SFB 36	Überwachung eines Signals mit 1 bis 10 Begleitwerten ohne Quittierungsanzeige
NOTIFY_8P	SFB 31	Überwachung von bis zu 8 Signalen mit 1 bis 10 Begleitwerten ohne Quittierungsanzeige

Einträge im Bausteinkopf

Damit der Baustein von der OS aus bedient und/oder beobachtet werden kann, setzen Sie zunächst in der Liste der Systemattribute im Bausteinkopf das Systemattribut "S7_m_c" auf 'true'.

Damit der PCS 7-Melddialog im SIMATIC Manager aufgerufen wird, tragen Sie im Bausteinkopf das Attribut S7_alarm_ui := '1' ein. Bei dem Wert '0' erhalten Sie den STEP 7-konformen Dialog.

Einträge im Deklarationsteil

Damit die Parameter Ihres Bausteins von der OS aus bedient und beobachtet werden können, setzen Sie für jeden einzelnen Parameter Ihres Bausteins, den Sie bedienen und beobachten wollen, das Systemattribut "S7_m_c" auf 'true'.

Wenn Ihr Baustein Meldungen und/oder Ereignisse an die OS senden soll, definieren Sie einen Eingang vom Datentyp DWORD (hier: EV_ID). Dieser Eingang nimmt beim Instanz-DB die Meldenummer auf, die automatisch vom System (Meldeserver) vergeben wird.

Die Meldenummer ist im gesamten S7-Projekt eindeutig, damit es in Projekten mit mehreren AS und OS nicht zu Kollisionen kommt. Aus dieser Meldenummer werden beim Datentransfer (OS-Übersetzen) die für WinCC notwendigen Nummern für die Einzelmeldungen abgeleitet.

Geben Sie an diesem Eingang das Systemattribut "S7_server" mit dem Wert 'alarm_archiv' und das Systemattribut "S7_a_type" mit dem Wert 'alarm_8p' oder gemäß dem eingebauten Meldungsbaustein ein.

Der Eingang soll im CFC nicht sichtbar, nicht verschaltbar und nicht parametrierbar sein, damit die vom System vergebenen Daten nicht versehentlich verändert werden.

Verwendung der Systemattribute im Bausteinkopf und für den Eingang EV_ID

Der folgende Ausschnitt des Beispiel-Bausteins zeigt die Verwendung der Systemattribute im **Bausteinkopf** und für den **Eingang EV_ID**, der die Meldenummer aufnehmen soll.

```
//*****
// Bausteinkopf
//*****

FUNCTION_BLOCK    "CONTROL"
TITLE=              'CONTROL'

{ // Liste der Systemattribute
S7_tasklist:=      'OB80,OB100'; // Baustein wird bei Zeitfehler und Neustart
                        // aufgerufen
S7_m_c:='true';    // Baustein ist bedien- und beobachtbar
S7_alarm_ui:=      '1'      // Einstellung PCS7-Melddialog ('0' = Standard-Dialg
}
AUTHOR:            ABC
NAME:              CONTROL
VERSION:           '0.02'
FAMILY:            XYZ
KNOW_HOW_PROTECT

//*****
// Deklarationsteil: Bausteinparameter
//*****

VAR_INPUT
....
                        // Parameter EVENT ID für Meldenummer
EV_ID  {S7_visible:='false'; // Parameter im CFC nicht sichtbar
        S7_link:='false';    // Parameter im CFC nicht verschaltbar
        S7_param :='false';  // Parameter im CFC nicht parametrierbar
        S7_server:='alarm_archiv'; // Vergabe der Meldenummer durch Server
        S7_a_type:='alarm_8p' // Baustein meldet mit ALARM_8P
        }      :DWORD := 0; // Meldungsnummer
...
END_VAR
```

Weitere Möglichkeiten für Meldungen

Sie können die nicht im Baustein benötigten Eingänge des ALARM-Bausteins auf das Baustein-Interface legen, um dem späteren Anwender weitere Möglichkeiten für Meldungen zu geben. Durch die Definition von zusätzlichen Eingängen und der Verknüpfung im Algorithmus Ihres Bausteins können Sie diese Meldungen sperren oder freigeben (siehe 1.6.2.). Sonst werden diese Meldungen ohne weitere Vorkehrungen behandelt und können nur vom Meldesystem der OS gesperrt werden. Das gilt auch für die nicht verwendeten Begleitwerte. Diese können Sie dann in den Meldungen verwenden wie in Abschnitt 1.7 beschrieben.

Definition des ALARM_8P

Im folgenden Beispiel sehen Sie die Definition des ALARM_8P:

```

//*****
// Deklarationsteil: Bausteinparameter
//*****
// freibelegbare Meldebegleitwerte des ALARM_8P
AUX_PR05 {S7_visible := 'false'} : ANY; // Begleitwert 5
AUX_PR06 {S7_visible := 'false'} : ANY; // Begleitwert 6
AUX_PR07 {S7_visible := 'false'} : ANY; // Begleitwert 7
AUX_PR08 {S7_visible := 'false'} : ANY; // Begleitwert 8
AUX_PR09 {S7_visible := 'false'} : ANY; // Begleitwert 9
AUX_PR10 {S7_visible := 'false'} : ANY; // Begleitwert 10
....
//*****
// Deklarationsteil: statische Variablen
//*****
....
//*****
// Deklarationsteil Multiinstanzen
//*****
OP_A_LIM_1: OP_A_LIM; // Bedienbaustein 1
ALARM_8P_1: ALARM_8P; // Erzeugung max. 8 Meldungen mit max. 10 Begleitwerten
...
//*****
// Melden mit ALARM_8P
//*****
// STRING-Variablen dürfen nicht als Begleitwert auf ALARM8_P verschaltet
// werden, deshalb in array of byte übertragen

FOR DUMMY := 1 TO 16
DO
    sbyBA_NA[DUMMY] := 0; //array löschen als Vorbesetzung
END_FOR;

DUMMY := BLKMOV (SRCBLK:= BA_NA,DSTBLK:=sbyBA_NA);
swSTEP_NO := STEP_NO; // Batch Schrittnr. (wegen I/O Begleitwert ALARM_8P)
sdBA_ID := BA_ID; // Batch ID (wegen I/O Begleitwert ALARM_8P)
srPV_IN := PV_IN; // Begleitwert darf kein Input sein

ALARM_8P_1(EN_R := TRUE, // Aktualisierung des Ausgangs ACK_STATE
ID := 16#EEEE, // Datenkanal für Meldungen (immer 16#EEEE)
EV_ID:= EV_ID, // Meldungsnummer > 0
SIG_1:= NOT M_SUP_AH AND QH_ALM, // zu überw. Signal 1 -> Meldung Alarm oben
SIG_2:= NOT M_SUP_AL AND QL_ALM, // zu überw. Signal 2 -> Meldung Alarm unten
SIG_3:= 0, // zu überwachendes Signal 3 -> keine Meldung
SIG_4:= 0, // zu überwachendes Signal 4
SIG_5:= 0, // zu überwachendes Signal 5
SIG_6:= 0, // zu überwachendes Signal 6
SIG_7:= 0, // zu überwachendes Signal 7
SIG_8:= 0, // zu überwachendes Signal 8
SD_1 := sbyBA_NA, // Begleitwert 1
SD_2 := swSTEP_NO, // Begleitwert 2
SD_3 := sdBA_ID, // Begleitwert 3
SD_4 := srPV_IN, // Begleitwert 4
SD_5 := AUX_PR05, // Begleitwert 5
SD_6 := AUX_PR06, // Begleitwert 6
SD_7 := AUX_PR07, // Begleitwert 7
SD_8 := AUX_PR08, // Begleitwert 8
SD_9 := AUX_PR09, // Begleitwert 9
SD_10:= AUX_PR10); // Begleitwert 10

QMSG_ERR := ALARM_8P_1.ERROR; // Zustandsparameter ERROR
MSG_STAT := ALARM_8P_1.STATUS; // Zustandsparameter STATUS
MSG_ACK := ALARM_8P_1.ACK_STATE; // aktueller OS Quittierzustand
....

```

1.6.1 Meldungsunterdrückung im Anlauf

Verfahren

Wenn Sie die Belastung des AS im Anlauf durch das gleichzeitige Generieren mehrerer Meldungen von verschiedenen Bausteinen verringern wollen, definieren Sie dafür einen Eingangsparameter RUNUPCYC vom Datentyp INT. An diesem Parameter können Sie die Anzahl der Anlaufzyklen angeben, während deren keine Meldung generiert werden soll. Sie müssen dann im Bausteinalgorithmus die Anzahl der Aufrufe zählen und erst nach Ablauf der parametrisierten Zyklen die Generierung der Meldungen freigeben. Der folgende Ausschnitt des Beispiel-Bausteins zeigt dieses Verfahren.

Beispiel

```

/*****
// Deklarationsteil: Bausteinparameter
/*****
VAR_INPUT
...
  H_ALM {S7_m_c := 'true';           // Parameter ist B&B-fähig
        S7_visible:='false';        // Parameter ist unsichtbar
        S7_link := 'false'          // und nicht verschaltbar
        } :REAL :=100;               // oberer Grenzwert Alarm (Vorbesetzung 100)

  L_ALM {S7_m_c := 'true';           // Parameter ist B&B-fähig
        S7_visible:='false';        // Parameter ist unsichtbar
        S7_link := 'false'          // und nicht verschaltbar
        } :REAL := 0;               // oberer Grenzwert Alarm (Vorbesetzung 0)
...
  RUNUPCYC {S7_visible:='false';
            S7_link:='false'} :INT := 3; // Anzahl Erstlaufzyklen
END_VAR
/*****
// Deklarationsteil: statische Variablen
/*****
VAR
...
sirUNUPCNT :INT := 0; // Zähler für RUNUPCYC-Bearbeitung
...
END_VAR
/*****
// Anlauf
/*****
// OB100: Neustart
100:
...
  sirUNUPCNT := RUNUPCYC; // RUNUPCYC-Wert abspeichern
...

```

```

/*****
// Technologischer Teil
/*****
IF sirUNUPCNT = 0          // RUNUPCYC-Zyklus bereits abgelaufen ?
THEN
  IF (PV_IN > H_ALM) THEN // Wenn der Prozesswert die obere
                        // Alarmgrenze verletzt
    QH_ALM := 1;        // Fehlerausgang setzen
    QL_ALM := 0;        // Fehlerausgang rücksetzen

  ELSIF (PV_IN < L_ALM) THEN // Wenn der Prozesswert die
                        // untere Alarmgrenze verletzt
    QL_ALM := 1;        // Fehlerausgang setzen,
    QH_ALM := 0;        // Fehlerausgang rücksetzen
  ELSE
    QH_ALM := 0;        // Fehlerausgänge rücksetzen
    QL_ALM := 0;

  END_IF;
ELSE
  sirUNUPCNT := sirUNUPCNT - 1;
END_IF;
END_CASE;

```

1.6.2 Meldungsunterdrückung für bestimmte Meldungen

Verfahren

Wenn Sie im Bedarfsfall, z. B. bei vorhersehbaren Meldungen, bestimmte Meldungen unterdrücken wollen, können Sie wie folgt vorgehen:

Sie definieren an Ihrem Baustein einen Eingangsparameter vom Datentyp BOOL, den Sie in Ihrem Bausteinalgorithmus auswerten, sodass bei unterdrückter Meldung dieses Ereignis nicht zum SIG-Eingang des ALARM-Bausteins weitergeleitet wird.

Im folgenden Beispiel werden die Eingänge M_SUP_AL und M_SUL_AH für die Unterdrückung eines einzelnen Alarms verwendet.

Beispiel

```

/*****
// Deklarationsteil: Bausteinparameter
*****/

VAR_INPUT
.....
                // Unterdrückung des ALARM LOW
M_SUP_AL {S7_visible:='false';
S7_link:='false';
S7_m_c:='true';
S7_string_0:= 'Suppress LL=No';      // Bedientext für Wert (M_SUP_AL)= 0
S7_string_1:= 'Suppress LL=Yes'      // Bedientext für Wert (M_SUP_AL)= 1
}
                :BOOL;      // Meldungsunterdrückung unterer Alarm Istwert

                // Unterdrückung des ALARM HIGH
M_SUP_AH {S7_visible:='false';
S7_link:='false';
S7_m_c:='true';
S7_string_0:= 'Suppress HH=No';      // Bedientext für Wert (M_SUP_AH)= 0
S7_string_1:= 'Suppress HH=Yes'      // Bedientext für Wert (M_SUP_AH)= 1
}
                :BOOL;      // Meldungsunterdrückung oberer Alarm Istwert

END_VAR

/*****
// Melden mit ALARM_8P
*****/
.....
ALARM_8P_1(EN_R := TRUE,                // Aktualisierung des Ausgangs ACK_STATE
ID := 16#EEEE,                          // Datenkanal für Meldungen (immer 16#EEEE)
EV_ID:= EV_ID,                          // Meldungsnummer > 0
SIG_1:= NOT M_SUP_AH AND QH_ALM,        // zu überw. Signal 1 -> Meldung Alarm oben
SIG_2:= NOT M_SUP_AL AND QL_ALM,        // zu überw. Signal 2 -> Meldung Alarm unten
SIG_3:= 0,                              // zu überw. Signal 3 -> keine Meldung
SIG_4:= 0,                              // zu überw. Signal 4
.....

```

1.6.3 Quelle übersetzen

Nach der Programmierung müssen Sie die Quelle mit dem SCL-Compiler übersetzen. Wählen Sie dazu den Menübefehl **Datei > Übersetzen**. Alternativ können Sie auch das entsprechende Symbol in der Funktionsleiste verwenden.

Nach fehlerfreier Übersetzung ist der Baustein FB 501 im Bausteinordner des S7-Programms enthalten.

Weitere Informationen finden Sie im Handbuch *S7-SCL für S7-300 und S7-400*.

1.7 Meldungsprojektierung

Allgemeines

Wenn ein Baustein Meldungen an die OS schicken soll, müssen Sie in den statischen Variablen die Multiinstanz eines Alarm-Bausteins definieren.

Sie können mit dem ALARM_8/ALARM_8P-Baustein bis zu 8 Signale überwachen, die Sie als Parameter am Alarm-Baustein angeben. Der Baustein merkt sich bei jedem Aufruf den aktuellen Zustand der Signale und sendet beim nächsten Aufruf eine Meldung an die OS, wenn sich eines der Signale geändert hat.

Meldungsprojektierung im SIMATIC Manager

Den EV_ID können Sie im SIMATIC Manager für den markierten Baustein über das Dialogfeld "PCS 7-Meldungsprojektierung" bearbeiten. Wählen Sie dazu den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung....**

Dabei können Sie einzelne Bestandteile der Meldungen, z. B. Meldetext und Meldeklasse, gegen Änderung an anderer Stelle sperren. Damit können Sie z. B. verhindern, dass bei einem Einbau Ihres Bausteins in einen CFC-Plan diese Meldung an der Bausteininstanz geändert wird.

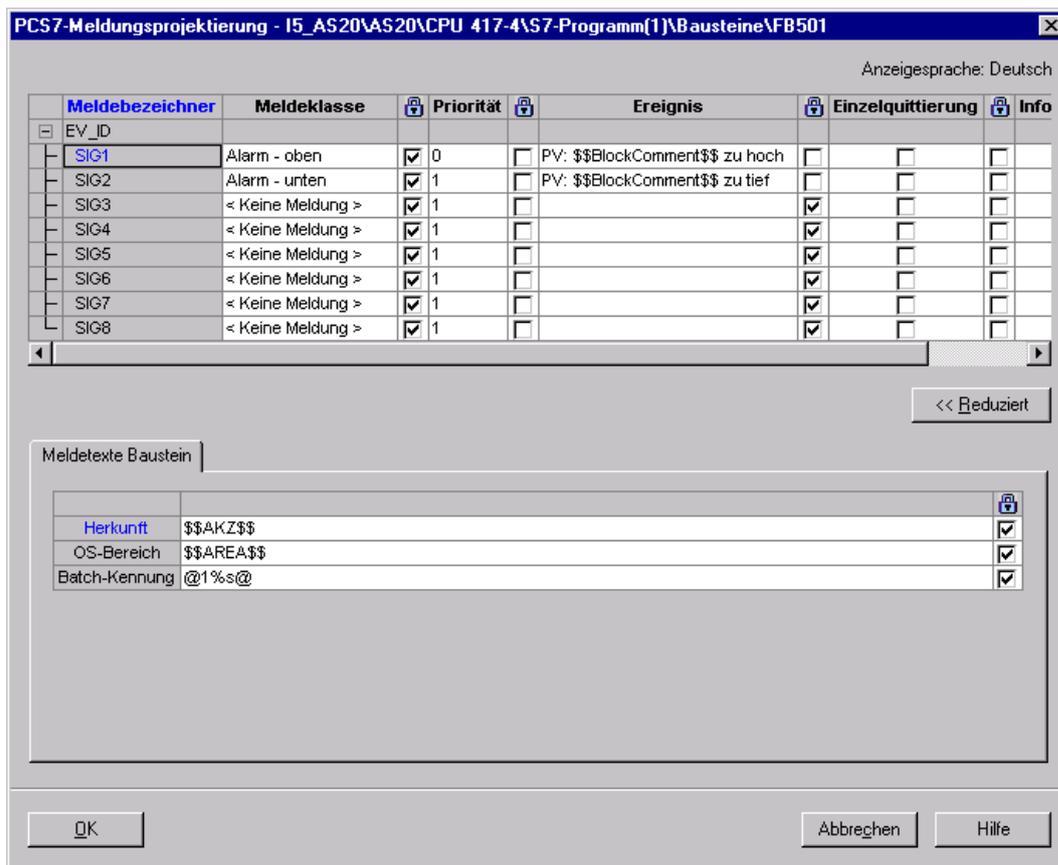


Bild 1-7::Meldungsprojektierung im SIMATIC Manager

Geben Sie zunächst die Texte ein, die für alle Meldungen dieses Bausteins gelten. Die einzelnen Texte entsprechen dabei den Anwendertextblöcken im AlarmLogging von WinCC.

Herkunft

Hier können Sie die Herkunft der Meldung angeben.

Wenn Sie die Herkunft als Schlüsselwort `$$AKZ$$` eintragen, geschieht beim Transferieren der Daten Folgendes:

Beim Übersetzen der OS werden der Pfad des Hierarchieordners, der Planname und der Bausteinname ermittelt und in den OS-Meldetexten abgelegt.

Hinweis

Der TH-Pfad wird nur eingetragen, wenn die entsprechenden Hierarchieordner namensbildend sind (Eigenschaften THO bzw. Einstellungen der TH).

OS-Bereich

Hier können Sie die Bereichszuordnung der Meldung angeben.

Wenn Sie den Bereich als Schlüsselwort `$$AREA$$` eintragen oder wenn Sie das Eingabefeld leer lassen, geschieht beim Transferieren der Daten Folgendes:

Beim Übersetzen der OS wird das entsprechende Attribut des Hierarchieordners ausgewertet und in den OS-Meldetexten abgelegt.

Batch-Kennung

Hier können Sie eine Batch-Kennung für die Meldung angeben.

Wenn Sie die Batch-Kennung eintragen, geschieht beim Transferieren der Daten Folgendes:

Beim Übersetzen der OS wird das entsprechende Attribut ausgewertet und in der Meldeliste von WinCC in der Spalte "Charge Name" abgelegt. Es handelt sich hierbei aber nicht um die Batch-ID, sondern um den Batch-Namen.

Wenn Ihr Baustein für das S7-Optionspaket SIMATIC BATCH geeignet sein soll, müssen Sie hier `@1%s@` eintragen. Dadurch erhält die Meldung als ersten Meldebegleitwert die BATCH-Chargenbezeichnung (vgl. Kapitel 1.8).

Wenn Sie SIMATIC BATCH nicht verwenden, müssen Sie hier nichts angeben.

Meldeklassen

Anschließend legen Sie für jede Meldung die Meldeklasse fest. Sobald Sie in der jeweiligen Meldungszeile die Spalte "Meldeklasse" anklicken, wird diese zum Kombinationsfeld und Sie können die Meldeklasse auswählen. Nicht belegte Meldungen müssen die Meldeklasse "< keine Meldung >" erhalten. Genaueres über die Behandlung von Meldungen können Sie der Dokumentation zu WinCC entnehmen.

Nehmen Sie folgende Einstellungen vor:

- Tragen Sie in der Spalte "Ereignis" eine Beschreibung der Fehlerursache ein. Verwenden Sie dafür maximal 40 Zeichen inklusive möglicher Begleitwerte.
- In der Spalte "Einzelquittierung" definieren Sie Folgendes:
 - Wenn die Meldung einzeln quittiert werden muss, aktivieren Sie das Auswahlkästchen.
 - Wenn die Meldung als Sammelquittierung quittiert werden kann, deaktivieren Sie das Auswahlkästchen.
- In der Spalte "Locked" (blaues Schloss-Symbol) definieren Sie Folgendes:
 - Wenn die Meldeklasse vom Anwender des Bausteins instanzspezifisch im CFC geändert werden darf, deaktivieren Sie das Auswahlkästchen.
 - Wenn Sie die Änderung der Meldeklasse sperren wollen, aktivieren Sie das Auswahlkästchen.

Priorität

Hier können Sie unterschiedliche Prioritäten einer Meldung vergeben.

In der Spalte "Locked" (blaues Schloss-Symbol) können Sie Folgendes definieren:

- Wenn die Priorität vom Anwender des Bausteins instanzspezifisch im CFC geändert werden darf, deaktivieren Sie das Auswahlkästchen.
- Wenn Sie die Änderung der Priorität sperren wollen, aktivieren Sie das Auswahlkästchen.

Ereignis

Geben Sie in diesem Feld den Meldetext ein.

Begleitwerte für Meldungen

Wenn eine Meldung zusätzliche Informationen, z. B. Messwerte, an die OS übertragen soll, müssen Sie zum Melden einen ALARM-Baustein verwenden, der die Angabe von Begleitwerten erlaubt (ALARM_8P = 10 Begleitwerte). Die an den Parametern SD_1 bis SD_10 des ALARM-Bausteins übergebenen Werte können Sie in der folgenden Form in die Meldungstexte einblenden:

@ Parameternummer Formatanweisung @

Im folgenden Beispiel wird der am Parameter SD_4 angegebene Wert in dezimaler Form ausgegeben. Die angebbaren Formatanweisungen entsprechen der C-Syntax.

@4%d@

Mit dem Schlüsselwort `$$BlockComment$$` können Sie den Kommentar der Bausteininstanz in den Meldetext einblenden.

In der Spalte "Locked" (blaues Schloss-Symbol) definieren Sie, ob der Meldetext vom Anwender des Bausteins instanzspezifisch im CFC geändert werden darf (Auswahlkästchen nicht aktiviert) oder gesperrt ist (Auswahlkästchen aktiviert).

Hinweis

Die Begleitwerte eines ALARM_8P sind vom Datentyp ANY, Anschlusstyp I/O. In der Online-Hilfe des ALARM_8P sind die zulässigen Datentypen der Begleitwerte aufgezählt. Wenn Sie im Baustein-Interface die Begleitwerte nicht als I/O, sondern als Input definieren wollen, müssen Sie diese auf eine Variable in der VAR-Sektion speichern und diese Variable als Begleitwert mit ALARM_8P übertragen.

Der Datentyp STRING darf nicht als Begleitwert übertragen werden. Sie müssen diesen, wie im Beispiel-Baustein implementiert, in ein ARRAY OF BYTE kopieren und dieses ARRAY als Begleitwert übertragen.

Einzelquittierung

Aktivieren Sie das Optionskästchen, wenn die Meldung als Einzelmeldung quittiert werden soll.

Infotext

Tragen Sie in diesem Feld den Infotext ein.

1.8 Anbindung von SIMATIC BATCH

Definition von Eingangs- und Durchgangsparemtern

Wenn Sie Ihre Bausteine mit dem S7-Optionspaket "SIMATIC BATCH" verwenden wollen, müssen Sie folgende Eingangs- oder Durchgangsparemter definieren:

Parametername	Bedeutung	Parametertyp	Datentyp
BA_EN	BATCH-Belegfreigabe	INPUT	BOOL
BA_NA	BATCH-Chargenbezeichnung	INPUT	STRING[32]
BA_ID	laufende Chargennummer	INPUT	DWORD
OCCUPIED	BATCH-Belegkennung	INPUT	BOOL
STEP_NO	BATCH-Schrittnummer	INPUT	DWORD

Ausschnitt des Beispiel-Bausteins:

```
//*****
// Deklarationsteil: Bausteinparameter
//*****
VAR_INPUT
....
    // Parameter für SIMATIC BATCH
STEP_NO {S7_visible := 'false';
    S7_m_c := 'true'} :DWORD;           // Batch Schrittnummer
BA_ID {S7_visible := 'false';
    S7_m_c := 'true'} :DWORD;           // Batch laufende Chargennummer
BA_EN {S7_visible := 'false';
    S7_m_c := 'true'} :BOOL := 0;       // Parameter im CFC-Plan unsichtbar
                                           // Parameter ist B&B-fähig
                                           // Batch Belegt-Freigabe
BA_NA {S7_visible := 'false';
    S7_m_c := 'true'} :STRING[32] := ''; // Batch Chargenbezeichnung

OCCUPIED {S7_visible := 'false';
    S7_m_c := 'true'} :BOOL := 0;       // Batch Belegt-Kennung
....
END_VAR
```

Meldungen generieren

Wenn Sie in einem solchen Baustein Meldungen generieren wollen, müssen Sie die Eingänge BA_NA, STEP_NO und BA_ID in der angegebenen Reihenfolge als Meldebegleitwerte verwenden.

Die Begleitwerte sind wie folgt belegt:

Begleitwert	Bedeutung
1	BATCH-Chargenbezeichnung BA_NA
2	BATCH-Schrittnummer STEP_NO
3	BATCH: laufende Chargennummer BA_ID
4 bis 7	Bausteinspezifisch belegt oder frei für Anwender.

1.9 Erstellen von CFC-Bausteintypen

1.9.1 CFC

Im Unterschied zur Programmierung mit SCL, bei der Variablen deklariert und Zuweisungen ausprogrammiert werden, basiert CFC auf der Verschaltung von grafischen Objekten. Mit CFC können Sie durch Platzieren und Verschalten von bereits vorhandenen Bausteinen neue Bausteine entwickeln.

Die folgende Beschreibung soll Ihnen einen Überblick und die prinzipielle Vorgehensweise erläutern. Eine ausführliche Beschreibung der Bausteinerstellung im CFC finden Sie im Handbuch *CFC für S7* und in der CFC-Online-Hilfe.

1.9.2 Beispiel: CONTROL2

Der fertiggestellte Beispiel-Baustein "CONTROL" soll um einen Multiplizierer erweitert werden. Mit diesem wird durch die Multiplikation der Eingangswerte "IN1" und "IN2" der Istwert gebildet. Der erweiterte Baustein soll als CONTROL2 (FB 601) erzeugt werden.

Vorgehen

1. Öffnen Sie einen neuen CFC-Plan und platzieren Sie darin den Beispiel-Baustein "CONTROL".
2. Ziehen Sie aus der CFC Library\ELEMENTA per Drag&Drop einen Multiplizierer **MUL_R** (FC 63) in den Plan.
3. Verschalten Sie den Ausgang "OUT" des **MUL_R** mit dem Prozesswert (Parameter "PV_IN") des Beispiel-Bausteins.
4. Wählen Sie im Plan den Menübefehl **Ansicht > Plananschlüsse**. Der Interface-Editor wird geöffnet.

Alternativ können Sie in der Funktionsleiste auf das Symbol  klicken.

5. Markieren Sie im linken Fenster der "Plananschlüsse" das Symbol "IN".
6. Verschalten Sie die Eingänge "IN1" und "IN2" des **MUL_R** mit den Plananschlüssen: Mit Drag&Drop ziehen Sie den Bausteinanschluss auf den Plananschluss im rechten Fenster.
7. Schalten Sie über den Menübefehl **Objekteigenschaften > Register: Anschlüsse > Spalte: Unsichtbar > gesetzte Anschlüsse rücksetzen** alle unsichtbaren Anschlüsse - außer EN und ENO - des Bausteins CONTROL sichtbar.
8. Verschalten Sie alle sichtbaren Anschlüsse, außer dem bereits verschalteten Istwert, mit den Plananschlüssen des CFC-Plans.

9. Um den den CFC-Plan als Baustein zu übersetzen, nehmen Sie folgende Einstellungen vor:
- Wählen Sie den Menübefehl **Plan > Übersetzen > Plan als Bausteintyp**. Ein Dialogfeld wird geöffnet.
 - Tragen Sie im Register "Allgemein" die FB-Nummer ein, hier 601. Tragen Sie danach in den entsprechenden Feldern die weiteren Eigenschaften ein: Symbolischer Name, Familie, Autor, Version. Im Feld "Name" (Header) ist bereits der Name des CFC-Plans vorbesetzt.
 - Tragen Sie im Register "Attribute" die gewünschten Baustein-Attribute und Systemattribute ein. Das Systemattribut "S7_tasklist" müssen Sie hier nicht angeben (siehe nachfolgende Einbauregel).
 - Klicken Sie auf die Schaltfläche "OK". Die Übersetzung wird gestartet.

Einbauregel

Der CFC-Bausteintyp wird in folgende OBs eingebaut:

- in den als Default-Einstellung vorgesehenen zyklischen OB (z. B. OB 35)
- in jeden OB , der in einer Taskliste eines unterlagerten Bausteins enthalten ist. D.h. seine Taskliste ist die Vereinigungsmenge der Tasklisten der unterlagerten Bausteine. Die unterlagerten Bausteine selbst werden nur in den OBs aufgerufen, die in ihrer eigenen Taskliste enthalten sind. Im hier angegebenen Beispiel bedeutet das Folgendes:
 - Der Beispiel-Bausteins **CONTROL** hat die Taskliste "S7_tasklist = 'OB80,OB100' ".
 - Der Multiplizierer **MUL_R** hat keine Taskliste.
 - Der CFC-Bausteintyp hat daher die Taskliste "S7_tasklist = 'OB80,OB100' ". Im OB 80 und OB 100 wird aber nur der **CONTROL** aufgerufen, nicht der **MUL_R**.

Damit der CONTROL-Baustein nicht im OB 35 aufgerufen wird, müssen Sie ihn aus dem OB 35 entfernen oder in einen anderen OB verschieben. Das Gleiche gilt für den Baustein MUL_R.

1.10 Namenskonventionen und Nummernbereich

Nummernbereich

Um Nummernkonflikte mit den von Siemens gelieferten leittechnischen PCS 7-Bausteinen zu vermeiden, beginnen Sie die Nummerierung Ihrer Bausteine ab Nummer 501. Berücksichtigen Sie bei der Festlegung Ihrer Bausteinnummern auch die Leistungsdaten der von Ihrer Bibliothek unterstützten CPU-Typen.

Namen

Bei der Benennung Ihrer Bausteinparameter sollten Sie folgende Regel beachten:
Binäre Ausgänge beginnen mit **Q**, z. B. QH_ALM oder Q_L_ALM.

1.11 Quellcode des Beispiels

```
//Ersteller: ABC          Datum: 13.08.00          Vers.:1.00
//Geändert:             Datum: 18.11.03          Vers.:

//*****
// Bausteinkopf
//*****

FUNCTION_BLOCK "CONTROL"
TITLE = 'CONTROL'
{ // Liste der Systemattribute
S7_tasklist:= 'OB80,OB100'; // Baustein wird bei Zeitfehler und
                          // Neustart aufgerufen
S7_m_c:= 'true'; // Bausteins ist bedien- und beobachtbar
S7_alarm_ui:= '1' // Einstellung PCS7-Meldedialog ('0'=Standard-Meldedialog)
}
AUTHOR: ABC
NAME: CONTROL
VERSION: '0.02'
FAMILY: XYZ
KNOW_HOW_PROTECT

//*****
// Deklarationsteil: Bausteinparameter
//*****

VAR_INPUT
SAMPLE_T {S7_sampletime:= 'true'; // Parameter der Baustein-Abtastzeit
          // (Zyklus der Task)
          S7_visible:= 'false'; // Parameter ist unsichtbar
          S7_link:= 'false' // Parameter nicht verschaltbar
          } :REAL := 1; // Verzögerungszeit [s] (Vorbesetzung 1 Sek)

H_ALM {S7_m_c := 'true';
       S7_visible:= 'false';
       S7_link := 'false'} :REAL :=100; // oberer Grenzwert Alarm
                                           // (Vorbesetzung 100)
L_ALM {S7_m_c := 'true'; // Parameter ist B&B-fähig
       S7_visible:= 'false'; // Parameter ist unsichtbar
       S7_link := 'false' // und nicht verschaltbar
       } :REAL := 0; // unterer Grenzwert Alarm (Vorbesetzung 0)

M_SUP_AL {S7_visible:= 'false';
          S7_link:= 'false';
          S7_m_c:= 'true';
          S7_string_0:= 'Suppress LL=No'; // Bedientext für Wert (M_SUP_AL)= 0
          S7_string_1:= 'Suppress LL=Yes' // Bedientext für Wert (M_SUP_AL)= 1
          } :BOOL := 0; // Meldungsunterdrückung unterer Alarm

M_SUP_AH {S7_visible:= 'false';
          S7_link:= 'false';
          S7_m_c:= 'true';
          S7_string_0:= 'Suppress HH=No';
          S7_string_1:= 'Suppress HH=Yes'
          } :BOOL := 0; // Meldungsunterdrückung oberer Alarm

SP_OP_ON {S7_visible:= 'false';
          S7_dynamic:= 'true' // CFC in Test/IBS: Anzeige aktueller Wert in AS)
          } :BOOL := 1; // 1 = Bedienfreigabe für Sollwert-Eingabe

SPBUMPON {S7_visible:= 'false';
          S7_link:= 'false';
          S7_m_c:= 'true';
          S7_string_0:= 'SP bumpless=Off';
          S7_string_1:= 'SP bumpless=On'
          }
          :BOOL := 1; // 1 = Stoßfreiheit für Sollwert
SP_EXTON {S7_visible:= 'false';
          S7_dynamic:= 'true' // CFC in Test/IBS: Anzeige aktueller Wert in AS)
          }
          :BOOL := 1; // 1: Externer Sollwert ein

SP_EXT {S7_dynamic:= 'true'}

          :REAL := 0; // Externer Sollwert
SP_HLM {S7_visible:= 'false';
```

```

S7_link:='false';
S7_m_c:='true';
S7_shortcut:='SP high limit'; // Text(max 16 Zeichen) zur Anzeige auf OS
S7_unit:='' } // Einheit (max 16 Zeichen)
:REAL := 100; // Oberer Bediengrenzwert für Sollwert

SP_LLM {S7_visible:='false';
S7_link:='false';
S7_m_c:='true';
S7_shortcut:='SP low limit';
S7_unit:='' }
:REAL := 0; // Unterer Bediengrenzwert für Sollwert

PV_IN {S7_dynamic:='true';
S7_m_c:='true';
S7_unit:='%'} : REAL := 0; // Prozesswert (zu Begleitwert_PR04)

GAIN {S7_link:='false';
S7_edit:='para'; // Parametrierung im IEA
S7_m_c:='true';
S7_shortcut:='Gain';
S7_unit:='' } :REAL := 1; // Proportionalbeiwert

EV_ID {S7_visible:='false';
S7_link:='false';
S7_param:='false'; // Parameter im CFC nicht parametrierbar
S7_server:='alarm_archiv'; // Vergabe der Meldenummer durch Server
S7_a_type:='alarm_8p' // Baustein meldet mit ALARM_8P
} :DWORD := 0; // Meldungsnummer

// Parameter für SIMATIC BATCH
STEP_NO {S7_visible := 'false';
S7_m_c := 'true'} :DWORD; // Batch Schrittnummer
BA_ID {S7_visible := 'false';
S7_m_c := 'true'} :DWORD; // Batch laufende Chargennummer
BA_EN {S7_visible := 'false';
S7_m_c := 'true'} :BOOL := 0; // Parameter im CFC-Plan unsichtbar
// Parameter ist B&B-fähig
// Batch Belegt-Freigabe
BA_NA {S7_visible := 'false';
S7_m_c := 'true'} :STRING[32] := ''; // Batch Chargenbezeichnung

OCCUPIED {S7_visible := 'false';
S7_m_c := 'true'} :BOOL := 0; // Batch Belegt-Kennung

RUNUPCYC {S7_visible:='false';
S7_link:='false'} :INT := 3; // Anzahl Erstlaufzyklen
SUPPTIME :REAL := 0; // Verzögerungszeit
SUPP_IN :REAL := 0; // Eingangswert für Verzögerungszeit
END_VAR

```

```

VAR_OUTPUT
  LMN {S7_shortcut:='pressure';      // Bezeichnung des Parameters auf OS
      S7_unit := '%';                // Einheit des Parameters
      S7_m_c := 'true'               // beobachtbar
    } :REAL;                          // Stellwert

  QH_ALM :BOOL := false;             // 1 = Oberer Grenzwert Alarm hat angesprochen

  QL_ALM :BOOL := false;             // 1 = Unterer Grenzwert Alarm hat angesprochen

  QSP_HLM {S7_visible:='false';
          S7_dynamic:='true'} :BOOL := 0; // 1 = Sollwertausgang nach oben begrenzt

  QSP_LLM {S7_visible:='false';
          S7_dynamic:='true'} :BOOL := 0; // 1 = Sollwertausgang nach unten begrenzt

  Q_SP_OP {S7_visible:='false';
          S7_dynamic:='true';
          S7_m_c:='true'} :BOOL := 0; // Status: 1 = Bedienfreigabe für Sollwert

  QOP_ERR {S7_visible:='false';
          S7_dynamic:='true'} :BOOL := 0; // 1 = Bedienfehler

  QMSG_ERR {S7_visible:='false';
          S7_dynamic:='true'} :BOOL := 0; // ALARM_8P: Meldungsfehler

  MSG_STAT {S7_visible:='false';
          S7_dynamic:='true'} :WORD := 0; // Meldungsfehler-Informationen

  MSG_ACK {S7_visible:='false';
          S7_dynamic:='true'} :WORD := 0; // Meldungen quittieren

  SUPP_OUT :REAL := 0; // Ausgangswert für Verzögerungszeit
  SP {S7_dynamic:='true';
     S7_m_c:='true'} :REAL := 0; // Aktiver Sollwert

END_VAR

VAR_IN_OUT
  SP_OP {S7_visible:='false';
        S7_link:='false';
        S7_m_c:='true';
        S7_shortcut:='Setpoint';
        S7_unit:='%'} : REAL := 0; // Bedieneingabe Sollwert

  // freibelegbare Meldebegleitwerte des ALARM_8P

  AUX_PR05 {S7_visible := 'false'} : ANY; // Begleitwert 5
  AUX_PR06 {S7_visible := 'false'} : ANY; // Begleitwert 6
  AUX_PR07 {S7_visible := 'false'} : ANY; // Begleitwert 7
  AUX_PR08 {S7_visible := 'false'} : ANY; // Begleitwert 8
  AUX_PR09 {S7_visible := 'false'} : ANY; // Begleitwert 9
  AUX_PR10 {S7_visible := 'false'} : ANY; // Begleitwert 10

END_VAR

```

```

/*****
// Deklarationsteil: statische Variablen
/*****
VAR
  sbRESTART      :BOOL := TRUE;           // Erstlauf Merker
  siRUNUPCNT     :INT  := 0;              // Zähler für RUNUPCYC-Bearbeitung
  sSUPP_IN       :REAL := 0;              // Altwert des Delay-Beispieleingangs
  ACT_TIME       :REAL := 0;              // Zeitzähler

  srPV_IN        :REAL := 0;              // Begleitwert für PV_IN
  swSTEP_NO      :DWORD;                  // Batch Schrittnummer
  sdBA_ID        :DWORD;                  // Batch Chargennummer

  sbyBA_NA       :ARRAY[1..32] OF BYTE := 32(0);

/*****
// Deklarationsteil Multiinstanzen
/*****
  OP_A_LIM_1: OP_A_LIM; // Bedienbaustein 1
  ALARM_8P_1: ALARM_8P; // Erzeugung max. 8 Meldungen mit max. 10 Begleitwerten
END_VAR

/*****
// Deklarationsteil: temporäre Variablen
/*****
VAR TEMP
  // Startinfo: Struktur mit Info für OB, der den Baustein gerade aufgerufen hat
  TOP_SI: STRUCT
    EV_CLASS :BYTE;
    EV_NUM   :BYTE;
    PRIORITY :BYTE;
    NUM      :BYTE;
    TYP2_3   :BYTE;
    TYP1     :BYTE;
    ZI1      :WORD;
    ZI2_3    :DWORD;
  END_STRUCT;

  // Startinfo: Struktur mit Info für den letzten aufgerufenen Anlauf-OB
  START_UP_SI: STRUCT
    EV_CLASS :BYTE;
    EV_NUM   :BYTE;
    PRIORITY :BYTE;
    NUM      :BYTE;
    TYP2_3   :BYTE;
    TYP1     :BYTE;
    ZI1      :WORD;
    ZI2_3    :DWORD;
  END_STRUCT;

  S7DT      :DATE_AND_TIME; // Lokale Zeitvariable
  DUMMY     :INT;           // Hilfsvariable
END_VAR

```

```

/*****
// Codeteil
/*****
// Abhängigkeiten vom aufrufenden OB
/*****
// Auslesen der Startinfo mittels SFC6 (RD_SINFO)
DUMMY := RD_SINFO (TOP_SI := TOP_SI, START_UP_SI := START_UP_SI);

IF sbRESTART THEN
// Erstlauf
TOP_SI.NUM := 100;           // Erstlauf als Neustart ausführen
sbRESTART := FALSE;        // Rücksetzen Erstlauf
END_IF;

// Aus welchem OB wurde der Baustein aufgerufen ?
CASE WORD_TO_INT(BYTE_TO_WORD(TOP_SI.NUM)) OF

/*****
// Behandlung von Fehler-OBS
/*****
// OB80: Zeitfehler
80:
QH_ALM := 0;           // Fehlerausgänge zurücksetzen
QL_ALM := 0;
/*****
// Anlauf
/*****
// OB100: Neustart
100:
QH_ALM := 0;           // Fehlerausgänge zurücksetzen
QL_ALM := 0;
siRUNUPCNT := RUNUPCYC; // RUNUPCYC-Wert abspeichern
ELSE
/*****
// Bedieneingabe Sollwert SP_OP (Real-Wert) oder verschalteter Sollwert SP_EXT
/*****
// Multiinstanz Aufruf OP_A LIM
// (Bedeutung der Parameter siehe Online-Hilfe OP_A LIM)
OP_A_LIM_1(U := SP_OP, U_HL:= SP_HLM, U_LL:= SP_LLM, OP_EN:= SP_OP_ON,
BTRACK:= SPBUMPON, LINK_ON:= SP_EXTON, LINK_U:= SP_EXT);

OK := OK AND ENO; //Enable Out des OP_A LIM in OK-Flag des
// Bausteins übernehmen
Q_SP_OP := OP_A_LIM_1.QOP_EN; // 1: Freigabe Bedieneingabe SP
QOP_ERR := OP_A_LIM_1.QOP_ERR; // 1: Bedienfehler
QSP_HLM := OP_A_LIM_1.QVHL; // 1: Begrenzung Obergrenze
QSP_LLM := OP_A_LIM_1.QVLL; // 1: Begrenzung Untergrenze
SP := OP_A_LIM_1.V; // wirksamer Sollwert

/*****
// Technologischer Teil
/*****
IF (SUPP_IN <> sSUPP_IN) THEN
ACT_TIME := SUPPTIME; // Zeitzähler initialisieren
sSUPP_IN := SUPP_IN;
END_IF;

IF (ACT_TIME > 0) THEN // Wenn Wartezeit noch nicht abgelaufen ist
ACT_TIME := ACT_TIME-SAMPLE_T; // Wartezeit herunterzählen
ELSE
SUPP_OUT := SUPP_IN; // Eingang durchschalten
END_IF;

```

```

LMN := GAIN * (SP - PV_IN);          // Stellgröße berechnen
IF siRUNUPCNT = 0                    // RUNUPCYC-Zyklus bereits abgelaufen ?
THEN
  IF (PV_IN > H_ALM) THEN // Wenn Prozesswert die obere Alarmgrenze verletzt
    QH_ALM := 1;          // Fehlerausgang setzen
    QL_ALM := 0;          // Fehlerausgang rücksetzen

  ELSIF (PV_IN < L_ALM) THEN // Wenn Prozesswert die untere Alarmgr. verletzt
    QL_ALM := 1;          // Fehlerausgang setzen,
    QH_ALM := 0;          // Fehlerausgang rücksetzen
  ELSE
    QH_ALM := 0;          // Fehlerausgänge rücksetzen
    QL_ALM := 0;

  END_IF;
ELSE
  siRUNUPCNT := siRUNUPCNT - 1;
END_IF;
END_CASE;

//*****
// Melden mit ALARM_8P
//*****

// STRING-Variablen dürfen nicht als Begleitwert auf ALARMS_P verschaltet
// werden, deshalb in array of byte übertragen

FOR DUMMY := 1 TO 32
DO
  sbyBA_NA[DUMMY] := 0; // array löschen als Vorbesetzung
END_FOR;

DUMMY := BLKMOV (SRCBLK:= BA_NA,DSTBLK:=sbyBA_NA);
swSTEP_NO := STEP_NO; // Batch Schrittnummer
// (wegen I/O Begleitwert ALARM_8P)
sdBA_ID := BA_ID; // Batch Chargennummer
// (wegen I/O Begleitwert ALARM_8P)
srPV_IN := PV_IN; // Begleitwert darf kein INPUT sein

ALARM_8P_1(EN_R := TRUE, // Aktualisierung des Ausgangs ACK_STATE
ID := 16#EEEE, // Datenkanal für Meldungen (immer 16#EEEE)
EV_ID:= EV_ID, // Meldungsnummer > 0
SIG_1:= NOT M_SUP_AH AND QH_ALM, // zu überwachendes Signal
// 1 -> Meldung Alarm oben
SIG_2:= NOT M_SUP_AL AND QL_ALM, // zu überwachendes Signal
// 2 -> Meldung Alarm unten
SIG_3:= 0, // zu überwachendes Signal 3 -> keine Meldung
SIG_4:= 0, // zu überwachendes Signal 4
SIG_5:= 0, // zu überwachendes Signal 5
SIG_6:= 0, // zu überwachendes Signal 6
SIG_7:= 0, // zu überwachendes Signal 7
SIG_8:= 0, // zu überwachendes Signal 8
SD_1 := sbyBA_NA, // Begleitwert 1
SD_2 := swSTEP_NO, // Begleitwert 2
SD_3 := sdBA_ID, // Begleitwert 3
SD_4 := srPV_IN, // Begleitwert 4
SD_5 := AUX_PR05, // Begleitwert 5
SD_6 := AUX_PR06, // Begleitwert 6
SD_7 := AUX_PR07, // Begleitwert 7
SD_8 := AUX_PR08, // Begleitwert 8
SD_9 := AUX_PR09, // Begleitwert 9
SD_10:= AUX_PR10); // Begleitwert 10

QMSG_ERR := ALARM_8P_1.ERROR; // Zustandsparameter ERROR
MSG_STAT := ALARM_8P_1.STATUS; // Zustandsparameter STATUS
MSG_ACK := ALARM_8P_1.ACK_STATE; // aktueller OS Quittierzustand
END_FUNCTION_BLOCK

```

2 Bildbausteine projektieren

2.1 Allgemeines zur Projektierung

2.1.1 Voraussetzungen und Vorkenntnisse

In den nachfolgenden Kapiteln erhalten Sie alle Informationen, um einen Bildbaustein für PCS 7 zu erstellen. Das umfasst die Verwendung der für die Erstellung benötigten WinCC-Tools und im Besonderen das Arbeiten mit dem Faceplate Designer.

Hinweis

Die in dieser Beschreibung abgebildeten Bausteinsymbole und Bildbausteine sind exemplarisch und können in Details von der Darstellung der aktuellen Objekte abweichen.

Voraussetzungen und Vorkenntnisse

Die hier beschriebenen Bildbausteine sind für die Verwendung in WinCC gedacht. Für die Erstellung der Bausteine benötigen Sie das WinCC-Basispaket mit den leittechnischen Optionen "Basic Process Control" und "Advanced Process Control".

Folgende Kenntnisse werden vorausgesetzt:

- SIMATIC WinCC Systemkurs
- SIMATIC WinCC Offenheit N

Die Kurse werden vom A&D Trainingscenter angeboten.

2.1.2 Phasen der Erstellung

2.1.2.1 Erstellungsweg

Für das Erstellen eines Bildbausteins hat sich die folgende Vorgehensweise als vorteilhaft erwiesen:

- Entwurf des Bildbausteins
- Projektierung des Bildbausteins
- Test des Bildbausteins

2.1.2.2 Entwurf des Bildbausteins

Darstellung

Ein Bildbaustein ist die B&B-Schnittstelle zu einem AS-Baustein. Für die Anzeige eines Bildbausteins gibt es zwei Darstellungsarten:

- **Gruppendarstellung:** Darstellung der AS-Werte in unterschiedlichen Sichten mit Anwahlelement für die Kreisbilddarstellung
- **Kreisbilddarstellung:** Darstellung der Elemente aller Sichten der Gruppendarstellung

Systemattribute

Welche Ein-, Aus- und Durchgangparameter eines AS-Bausteins bedient und beobachtet werden können, legen Sie bei der Erstellung des AS-Bausteins durch die Systemattribute. Einzelheiten zu diesen Systemattributen finden Sie im Kapitel "Aufbau eines AS-Bausteins".

Parameter

Die Auswahl der Parameter erfolgt unter den folgenden Gesichtspunkten:

- Welche Daten benötigt das Bedienpersonal, um schnell und zweifelsfrei den aktuellen Zustand zu erfassen?
- Wie sollen diese Werte dargestellt werden?
- Welche Werte soll das Bedienpersonal verändern können?
- Welche Berechtigungsstufe ist für die Bedienung erforderlich?
- Sind prozessabhängige Bedienverriegelungen erforderlich?
- In welcher Sicht des Bildbausteins sollen die einzelnen Werte dargestellt werden?

Tipp: Gruppieren Sie dazu die einzelnen Parameter nach Funktionen. Platzieren Sie in der Sicht "Standard" die wichtigsten Elemente, vor allem jene, die sich fortlaufend ändern.

Gestaltung

Nach der Festlegung der Parameter und ihrer Darstellung erfolgt das Gestalten des Bildbausteins:

- die Auswahl der Bildelemente
- die Benennung der Bildelemente
- die Parametrierung der Bildelemente
- die Position der Bildelemente

Verwenden Sie Namen, die einen Bezug zum dargestellten Objekt haben und auch ausgesprochen werden können.

Beispiel

Wenn in einer Zustandsanzeige die Variable "OCCUPIED" dargestellt werden soll, nennen Sie diese Zustandsanzeige "OCCUPIED".

Diese Vorgehensweise erleichtert die Projektdokumentation und die Wartung.

2.1.2.3 Projektierung des Bildbausteins

WinCC Graphics Designer

Für die Projektierung des Bildbausteins steht Ihnen das Werkzeug "WinCC Graphics Designer" zur Verfügung. Ausgehend von den Vorlagen des Faceplate Designer, setzen Sie den Bildentwurf in WinCC-Bilder um. Eine ausführliche Beschreibung hierzu finden Sie im Kapitel 2.1.3, "Bildbausteinerstellung mit dem Faceplate Designer".

2.1.2.4 So testen Sie einen Bildbaustein

Vorgehen

Beim Test des Bildbausteins gehen Sie folgendermaßen vor:

1. Prüfen Sie im WinCC Explorer die Eigenschaften der erstellten Bilder anhand folgender Fragen:
 - Sind die Parameternamen richtig geschrieben?
 - Haben mehrfach dargestellte Parameter den identischen WinCC-Zyklus? Unterschiedliche Zyklen verwirren das Bedienpersonal, z. B. wenn die Balkenanzeige inkonsistent zur numerischer Anzeige ist, und erhöhen die Kommunikationslast des Systems.
 - Sind die Direktverbindungen korrekt?
 - Sind alle ereignisgesteuerten Skripte vorhanden?
2. Prüfen Sie in WinCC Runtime die Eigenschaften der erstellten Bilder anhand folgender Fragen:
 - Wird bei Mausklick auf das Bausteinsymbol der Bildbaustein in der Gruppendarstellung geöffnet?
 - Funktioniert in der Gruppendarstellung die Umschaltung der einzelnen Sichten?
 - Wird bei Mausklick auf die Taste "Kreisdarstellung" der Bildbaustein in der Kreisdarstellung geöffnet?
 - Werden die Werte des AS-Bausteins korrekt angezeigt?
 - Ist die Anzeige von Melde- und Trendsicht korrekt?
 - Funktioniert die Bedienfreigabe der bedienbaren Parameter?
 - Werden die Werte bei Bedienung in den AS-Baustein geschrieben?
 - Ist das Bedienprotokoll korrekt?

2.1.3 Bildbausteinerstellung mit dem Faceplate Designer

2.1.3.1 Faceplate Designer

Bestandteil des WinCC-Optionspakets "Advanced Process Control" ist das Faceplate Designer-Tool. Dieses Tool generiert die Vorlagen für die PCS 7-konforme Erstellung von Bildbausteinen.

2.1.3.2 Vorlagen des Faceplate Designer

Für die Erstellung von Bildbausteinen stehen Ihnen unter WinCC folgende Vorlagen zur Verfügung:

- Bausteinsymbole (fertige Symbole für Prozessbilder)
- Vorlagenbilder
- Objekt-Baukasten mit den Objekten für die Bildbaustein-Erstellung
- Globale Skripte

2.1.3.3 Bausteinsymbol-Vorlagen

WinCC-Bild "@@PCS7Typicals.pdl"



Die Bausteinsymbole sind im WinCC-Bild "@@PCS7Typicals.pdl" enthalten, z. B. Ventil, Antrieb, Messwert und Regler.

- Die Beispiel-Vorlagen sind beliebig änderbar, z. B. in Form, Farbe und Gestaltung, und schnell anpassbar, auch an projektspezifisch erstellte Bildbausteine.
- Die fertigen Aufruf-Skripte für die Bildbausteine sind schon enthalten und müssen nicht projiziert werden.
- Die Verschaltung erfolgt schnell und einfach mit dem Dynamic-Wizard "Bildbaustein mit Messstelle verbinden".

Weitere Informationen finden Sie in Kapitel 2.8, "Bausteinsymbole".

Hinweis

Speichern Sie das Bild nach Änderungen unter dem Namen "@@PCS7Typicals*.pd".

Nach diesen Namen sucht die Suchmaschine aus der TH zuerst. Nur wenn diese nicht gefunden werden, werden die Vorlagen aus "@@PCS7Typicals.pdl" verwendet.

2.1.3.4 Vorlagenbilder

Bilder und Bitmaps

Die Bilder und Bitmaps befinden sich im Verzeichnis "WinCC\options\pd\FaceplateDesigner_V6" und werden mit dem OS-Projekteditor in das Projekt kopiert. Siehe hierzu die Online-Hilfe "OS-Projekteditor Grunddaten".

Objekt-Baukasten

Eine Reihe fertig vorbereiteter Objekte (Anwenderobjekte) zur Erstellung eines Bildbausteins sind im WinCC-Bild "@PCS7Elements.pdl" enthalten, z. B. EA-Felder und Texte.

Das Bild ist im Pfad "Siemens\WinCC\options\pd\FaceplateDesigner_V6" abgelegt und wird beim Durchlauf des OS-Projekteditors in den Ordner "GraCS" des Projektverzeichnisses kopiert.

Weitere Informationen finden Sie in Kapitel 2.3, "Basiselemente".

Globale Skripte

Die Bildbaustein-Aufrufe bzw. Runtime-Funktionen befinden sich als globale Skripte im Verzeichnis "WinCC\aplib\FaceplateDesigner_V6".

Weitere Informationen finden Sie in Kapitel 2.4, "Skripte".

2.1.3.5 Projektierungsschritte

Hinweise zur Projektierung

Bei der Projektierung ist Folgendes zu beachten:

- Die mit dem Faceplate Designer erstellten Bildbausteine liegen zunächst im Verzeichnis "GraCS" des aktuell geöffneten Projekts. Für das projektübergeordnete Ablegen eigener Bildbausteine ist das Verzeichnis "\\Siemens\\WINCC\\options\\pd\\FaceplateDesigner" vorgesehen.

Mit dem OS-Projekteditor können Sie dateigranular auswählen, welche Standard-Bildbausteine aus dem Ordner

"\\Siemens\\WINCC\\options\\pd\\FaceplateDesigner_V6"

und welche selbst erstellten Bildbausteine aus dem Verzeichnis

"\\Siemens\\WINCC\\options\\pd\\FaceplateDesigner"

beim Erzeugen der Grunddaten in das Projekt kopiert werden sollen.

- Die Daten aus dem Verzeichnis "\\Siemens\\WINCC\\options\\pd\\FaceplateDesigner" müssen Sie auch auf das entsprechende Verzeichnis der WinCC-Clients kopieren.

Die an eigenen Bildbausteinen projektierten Funktionen können Sie bei Bedarf im Editor "Global Script" gegen Einsicht und Änderung schützen.

Informationen hierzu finden Sie in der Dokumentation zum Editor "Global Script".

- Die Dynamik der mit dem Faceplate Designer erstellten Bildbausteine ist vollständig über die Projektierung steuerbar. Die Performance eines Bildbausteins wird somit im besonderen Maße durch die Wahl einer geeigneten Dynamikprojektierung beeinflusst. Besonders wichtig ist in diesem Zusammenhang, auf eine optimale, schlanke Schnittstelle zwischen den AS- und OS-Funktionen zu achten. Dies trifft vor allem auf die Bausteinsymbole zu.

Hinweis

Hierbei empfiehlt es sich, das in V6 an den AS-Bausteinen neu eingeführte Statuswort "VSTATUS" insbesondere bei den Bausteinsymbolen zu verwenden, um Variablenanbindungen zu sparen.

Da die Standard-Bausteinsymbole auch AS-Bausteine aus der V5 visualisieren können müssen, konnte hier das neue Statuswort nicht verwendet werden.

- In der Dynamik der Bildbausteine darf kein C-Script mit fest kodiertem Instanznamen verwendet werden, da diese typspezifisch sind.

2.1.4 Bedienberechtigungen

2.1.4.1 Vergabe von Bedienberechtigungen

Instanzspezifische Vergabe von Bedienberechtigungen

Die Bedienberechtigungen können Sie bei den Bildbausteinen instanzspezifisch vergeben.

Die Projektierung und die instanzspezifische Information wird an den Bausteinsymbolen abgelegt.

Am Bausteinsymbol gibt es die zwei Eigenschaften "Processcontrolling_backup" und "HigherProcesscontrolling_backup", an denen Sie die Berechtigungsstufe für die Prozessbedienung und für die höherwertige Prozessbedienung einstellen können.

Die Informationen werden über ein Skript an den Bildbaustein übergeben.

Die Eigenschaften sind defaultmäßig auf die bisher gewohnten Berechtigungsstufen 5 und 6 eingestellt, können aber beliebig abgeändert werden. Hier ist jedoch zu beachten, dass die Stufen 1 bis 10 festen Bedeutungen bei WinCC zugeordnet sind.

Welche Parameter welcher der beiden Berechtigungsstufen unterliegen, wird von den bisherigen Bildbausteinen abgeleitet. Prozessbedienungen wie "Ein", "Aus", "Hand", "Automatik", "Sollwert stellen" sind der Berechtigungsstufe 5 und höhere Parametrierungen, wie "Grenzen", "Regelparameter" usw. der Berechtigungsstufe 6 zugeordnet.

Bereichsspezifische Bedienberechtigung

Die bereichsspezifische Bedienberechtigung bleibt nach wie vor erhalten.

Die bereichsspezifische Bedienberechtigung wird aufgrund der Zuordnung des Variablennamens (Tagname) geprüft. Hierzu ist es aber notwendig, dass in der Technologischen Hierarchie (TH) am Parameter "OS Bereich" der entsprechende Bereichsname parametrierung ist und mit dem tatsächlichen OS-Bereich im Picture Tree übereinstimmt.

2.1.4.2 Bedienberechtigungen für Basiselemente projektieren

Objekte "@Level5" und "@Level6"

Für die Projektierung von Bedienberechtigungen sind in den vom Faceplate Designer angelegten Sichten die Objekte "@Level5" und "@Level6" vorhanden.

Wenn neue Basiselemente in einer Sicht angelegt werden, können Sie diese über die Direktverbindung mit den Objekten "@Level5" und "@Level6" verbinden.

Diese Objekte dürfen Sie nicht löschen, da diese von Skripten beschrieben werden.

Die Eigenschaften "Hintergrundfarbe" und "Bedienfreigabe" werden verbunden, damit bei nicht vorhandener Bedienberechtigung das Feld mit "grau" hinterlegt wird.

Wenn die Bedienfreigabe von Objekten für Parameter aus dem AS benötigt wird, kann für die Bedienberechtigung auch die Passwort-Stufe über eine Direktverbindung weiter verschaltet werden. Bei nicht vorhandener Berechtigung werden die Objekte nicht grau geschaltet, sondern es wird ein Meldfenster mit der Meldung "Keine Berechtigung" geöffnet.

Als geeignetere Variante für das Zusammenführen von WinCC-Bedienberechtigungen und prozessabhängigen Freigaben auf ein Bedienelement gibt es bei den Basiselementen das Objekt "Permission". Die Projektierung dieses Objekts ist dort beschrieben.

Die Werte dieser Eigenschaften werden per Skript "PCS7_UpdatePermission_V6" geschaltet.

Bei "@Level5" ist die Bedienberechtigungsstufe 5 und bei "@Level6" die Bedienberechtigungsstufe 6 defaultmäßig eingestellt. Die Werte werden vom Bausteinsymbol über die Eigenschaften "Processcontrolling_backup" und "HigherProcesscontrolling_backup" versorgt und können somit instanzspezifisch geändert werden.

Vorgehen

1. Markieren Sie für die gewünschte Berechtigungsstufe das Objekt, z. B. @Level6, und wählen Sie "Objekteigenschaften".
2. Wählen Sie im Register "Ereignis → Propertythemen → Farben → Hintergrundfarbe" als Aktion eine Direktverbindung aus.
3. Wählen Sie unter "Ziel" im rechten Fenster und "Objekt im Bild" das neu eingebaute Objekt aus und wählen Sie dort als Eigenschaft die Hintergrundfarbe aus.
4. Wählen Sie im Register "Ereignis → Propertythemen → Sonstige → Bedienfreigabe" als Aktion eine Direktverbindung aus.
Bei diesem Ereignis ist bereits ein Skript hinterlegt, das gleichzeitig die Hintergrundfarbe steuert, wenn die Bedienfreigabe über eine AS-Variable gesteuert wird. Dieses Skript muss dann gelöscht werden.

5. Wählen Sie unter "Ziel" im rechten Fenster und "Objekt im Bild" das neu eingebaute Objekt aus und wählen Sie dort als Eigenschaft die Bedienfreigabe.

Wenn weitere Objekte in eine View eingebaut werden, wird das letzte Objekt, das über eine Direktverbindung verschaltet wurde, ausgewählt und der oben beschriebene Vorgang wiederholt.

Wichtig!

Sie müssen bei der Projektierung die Eigenschaften "Hintergrundfarbe" und "Bedienfreigabe" so einstellen, wie sie bei "@Level5" und "@Level6" projiziert wurden, z. B. bei der Hintergrundfarbe "grau" und bei der Bedienfreigabe "false". Damit ist die Funktionsfähigkeit der Skripte gewährleistet.

Hierzu besteht folgender Hintergrund:

Wenn mit dem Skript der gleiche Wert auf eine Eigenschaft geschrieben wird, den die Eigenschaft von der Projektierung her schon hatte, wird dieser Wert über die Direktverbindung nicht an andere Eigenschaften weiter gereicht. Deswegen müssen Sie darauf achten, dass die Ziel-Eigenschaften, die über die Direktverbindung beschrieben werden, die gleichen Projektierungswerte haben wie die Quell-Eigenschaften.

2.1.5 Übersicht ändern

Erstellen eines Bildbausteins ohne Meldungen und/oder ohne Sammelanzeige

Wenn Sie einen Bildbaustein ohne Meldungen und/oder ohne Sammelanzeige erstellen, müssen Sie in dem Bild "@PG_XXXXX_Overview.Pdl" einige Elemente und Skripte wieder entfernen, um eine bessere Performance zu erreichen. Folgende Elemente und Skripte müssen Sie entfernen:

- Button16, Meldungsquittierung
- Button17, Meldungen Sperren/Freigeben
- @Level5, Bedienberechtigungen
- MSG_LOCK, Symbol für Meldungsunterdrückung
- Bei dem Bildobjekt selbst das Skript am Ereignis "Bildanwahl"
- Wenn kein Batch-Parameter "Occupied" vorhanden, ist auch das Batch-Symbol "OCCUPIED" zu entfernen.

Beim Faceplate Designer können Sie durch die Optionskästchen "keine Sammelanzeige" und "keine Batch-Parameter" eine bessere Performance erreichen.

2.1.6 Multiinstanz projektieren

Multiinstanz

Multiinstanz bedeutet, dass in einem CFC-Plan mehrere Bausteine projiziert sind, die von einem Bildbaustein in der OS aus bedien- und beobachtbar sind.

Projektierung

Bei der Projektierung wird grundsätzlich das Symbol auf einen kompletten Variablennamen verschaltet.

Für den ausgewählten Variablennamen wird die Bedienberechtigungsprüfung für den gesamten Bildbaustein ausgeführt.

In den Bildern "@PG_XXXX.pdl" und "@PL_XXXX.pdl" müssen Sie am Objekt "@Faceplate" die Eigenschaft MULTI_INSTANCE auf 'true' setzen, damit der Typ "Multiinstanz" erkannt wird und vom Variablennamen nach der Berechtigungsprüfung der Bausteinname für die einzelnen Sichten abgeschnitten wird.

Alle Variablenanbindungen im Bildbaustein müssen Sie durch "/bausteinname" ergänzen.

z. B. /Regler.PV_IN.

Wichtig: Dies müssen Sie auch in den Skripten ausführen!

Im Bild OVERVIEW müssen auch alle Verschaltungen entsprechend ergänzt werden.

In den Bildern "@PG_XXXX.pdl" und "@PL_XXXX.pdl" müssen Sie am Objekt "@Faceplate" die Eigenschaften der Batch-Parameter für das Bedienprotokoll für einen Baustein festlegen. Hier ist die Multiinstanztechnik nicht berücksichtigt. Wenn hier Spezialfälle abgedeckt werden sollen, z. B. unterschiedliche BatchIDs oder StepNr. für die verschiedenen Bausteine, müssen Sie dies mit Skripten bei der Bildanwahl der einzelnen Sichten steuern.

Bei der Trendsicht ist bei "Modus = 2" die Multiinstanztechnik nicht berücksichtigt.

Online-Trends können nur zu dem Variablennamen angezeigt werden, der am Bausteinsymbol hinterlegt ist. Wenn Trends zu mehreren Bausteinen angezeigt werden sollen, erreichen Sie dies über die anderen Modi.

Einbringen von mehreren meldefähigen Bausteinen

Wenn Sie mehrere meldefähige Bausteine einbringen, müssen im Bild OVERVIEW folgende weiteren Punkte beachtet und realisiert werden:

- Sie müssen entsprechend der Anzahl meldefähiger Bausteine auch entsprechend viele Sammelanzeigen einbauen.
- Sie müssen das Skript für die Quittiertaste für Meldungen ergänzen. Skript "Button16/Mouseclick".
Bei der Quittierung muss für jede Instanz die Funktion "CSigAPIAcknowledgeTagAndCreateLTM" ausgeführt werden.

Beispiel:

```
TCHAR sztag1[_MAX_PATH + 1] = "";  
TCHAR sztag2[_MAX_PATH + 1] = "";  
  
GetComputerNameA(szStation, &dwSize);  
pszParentPicture = GetParentPicture(lpszPictureName);  
lpszCurrentUser = GetPropChar(pszParentPicture, "@Faceplate", "CurrentUser");  
pszTagname = GetPropChar(pszParentPicture, "OverviewWindow", "TagPrefix");  
  
strcpy(sztag1, pszTagname);  
strcpy(sztag2, pszTagname);  
strcat(sztag1, "Instanz1");  
strcat(sztag2, "Instanz2");  
printf ("sztag2 : %s\r\n", sztag2);  
  
if (!CSigAPIAcknowledgeTagAndCreateLTM(sztag1, lpszCurrentUser, szStation, &Err))  
    printf ("Fehler bei CSigAPIAcknowledgePicture : %s\r\n", Err.szErrorText);  
if (!CSigAPIAcknowledgeTagAndCreateLTM(sztag2, lpszCurrentUser, szStation, &Err))  
    printf ("Fehler bei CSigAPIAcknowledgePicture : %s\r\n", Err.szErrorText);
```

- Sie müssen das Skript für die Taste "Meldungen Sperren/ Freigeben" ergänzen.
Skript "Button17/Mouseclick".
Beim "Meldungen Sperren/ Freigeben" muss für jede Instanz die Funktion "CSigAPILockMessage" ausgeführt werden.

Beispiel:

```
DWORD dEventState1 = GetPropWord(lpszPictureName,"EventState","CollectValue");
DWORD dEventState2 = GetPropWord(lpszPictureName,"EventState2","CollectValue");
TCHAR sztag1[MAX_PATH + 1] = "";
TCHAR sztag2[MAX_PATH + 1] = "";
```

```
pszParentPicture = GetParentPicture(lpszPictureName);
pszTagName = GetPropChar(pszParentPicture,"OverviewWindow","TagPrefix");
lpszCurrentUser = GetPropChar(pszParentPicture,"@Faceplate","CurrentUser");
GetComputerNameA(szStation, &dwSize);
```

```
strcpy(sztag1,pszTagName);
strcpy(sztag2,pszTagName);
strcat(sztag1,"/R");
strcat(sztag2,"/M");
```

```
if ((dEventState & 65536) == 65536)
{
    bRet = CSigAPILockMessage(FALSE, sztag1, lpszCurrentUser, szStation, &Err);
    SetPictureUp(lpszPictureName,lpszObjectName,"@Lock.bmp");
}
else
{
    bRet = CSigAPILockMessage(TRUE, sztag1, lpszCurrentUser, szStation, &Err);
    SetPictureUp(lpszPictureName,lpszObjectName,"@Unlock.bmp");
}
```

```
if ((dEventState2 & 65536) == 65536)
{
    bRet = CSigAPILockMessage(FALSE, sztag2, lpszCurrentUser, szStation, &Err);
    SetPictureUp(lpszPictureName,lpszObjectName,"@Lock.bmp");
}
else
{
    bRet = CSigAPILockMessage(TRUE, sztag2, lpszCurrentUser, szStation, &Err);
    SetPictureUp(lpszPictureName,lpszObjectName,"@Unlock.bmp");
}
```

- Sie müssen für alle meldefähigen Bausteine ein Symbol zur Meldungsunterdrückung (MSG_LOCK) einbauen.
- Sie müssen für alle Batch-relevanten Bausteine ein OCCUPIED-Symbol einbringen.

2.1.7 Zahlenformate projektieren

Neue Eigenschaften für Zahlenformatierungen

Die relevanten Bausteinsymbole bekommen drei neue Eigenschaften für Zahlenformatierungen:

Format_InputValue	für Eingangswerte am Bausteinsymbol wie "Setpoint" und "ProcessValue"
Format_OutputValue	für die Stellgröße am Bausteinsymbol
Format_xx	für evtl. weitere Werte

Hinweis

Die Begriffe "Format_InputValue" und "Format_OutputValue" sind hier prozessorientiert gewählt worden und haben keinen direkten Bezug zu den Ein-/Ausgängen der AS-Bausteinparameter.

Beispiel: Beim FMT_PID ist der Parameter "PV" zwar am AS-Baustein ein Ausgangsparameter, aber aus Prozesssicht eine Eingangsgröße.

Mit **Format_InputValue** und **Format_OutputValue** werden die Analogwerte des Bausteinsymbols versorgt.

Die Default-Einstellung für **Format_InputValue** zeigt gleitende Vorkommastellen und zwei gleitende Nachkommastellen; mindestens eine Vorkommastelle wird immer gezeigt (0.##).

Die Default-Einstellung für **Format_OutputValue** zeigt gleitende Vorkommastellen und zwei gleitende Nachkommastellen; mindestens eine Vorkommastelle wird immer gezeigt (0.##).

Der Projektteur kann diese Default-Einstellung bei Bedarf instanzspezifisch ändern.

Alle drei Format-Eigenschaften des Bausteinsymbols werden per Skript in den Bildbaustein übertragen und können dort über eine Direktverbindung den Analogwerten zugewiesen werden.

Details:

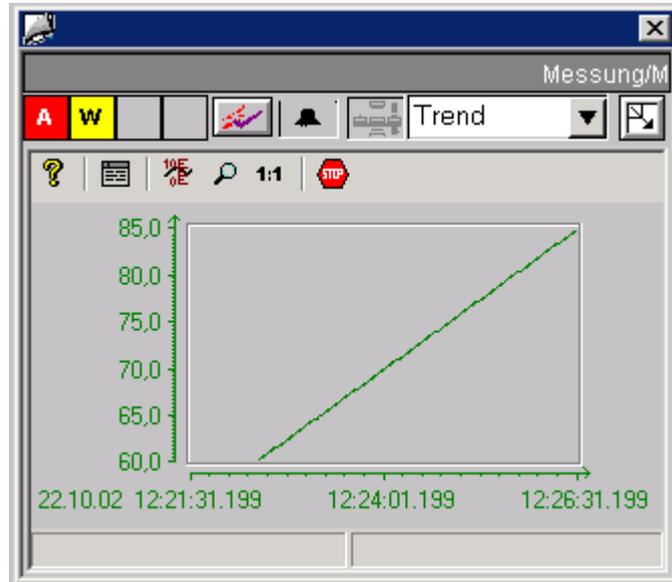
Die Format-Eigenschaften des Bausteinsymbols werden über das Skript "PCS7_OpenGroupDisplay_V6" in das übergeordnete Prototypbild "@PG_xxxxx.pdl" in das Objekt "Format" übertragen und abgelegt.

Wenn in einer Sicht ein Objekt "Format" mit den Eigenschaften "Format_InputValue, Format_OutputValue, Format_xx" existiert, werden bei Bildanwahl über das Skript "PCS7_Format_V6" die Formatierungen in dieses Objekt übertragen.

Von diesem Objekt aus können Sie nun über eine Direktverbindung die drei Formate beliebig den Analogwerten in dieser Sicht zuweisen.

2.1.8 Trendsicht projektieren

Realisierung von Trendbildern



Für die Realisierung von Trendbildern gibt es bei den Bausteinsymbolen neue Eigenschaften, z. B für Regler oder Messung:

- **ReturnPath:** Hier werden die Kurven-Daten für die entsprechende Messstelle übertragen.
- **StandardTrend:** Damit wird festgelegt, welche Trendfunktionalität in der Trendsicht dargestellt wird.

Standard-Trend:	Modus
0	<p>Trendfunktionalität wie in V5; für jede Instanz muss ein eigenes Trendbild projiziert werden. Die für alle PCS 7-Bildbausteine vorhandene Trendsicht "@PCS7_TREND.pdl" enthält das Bildfenster "TrendPicture". Damit an dieser Stelle für eine Messstelle ein Trendbild eingeblendet wird, müssen Sie folgendermaßen vorgehen:</p> <ol style="list-style-type: none"> 1. Öffnen Sie den WinCC-Editor "Tag Logging". 2. Legen Sie mit Hilfe des Archiv Wizards ein Variablen-Archiv für die Trendsicht an. 3. Legen Sie die Variablen für dieses Archiv an. <p>Eine ausführliche Anleitung zu den o.a. Schritten finden Sie in der WinCC-Online-Hilfe unter dem Index "Tag Logging".</p> <ol style="list-style-type: none"> 4. Öffnen Sie im Graphics Designer das Bild "@CONL1_Standard.pdl". 5. Konfigurieren Sie das TlgOnlineTrend-Objekt "Control1" entsprechend Ihren Anforderungen; für weitere Informationen nutzen Sie die Schaltfläche "Hilfe". 6. Speichern Sie das Bild unter dem Namen "@CONL1_<tagname>.pd". <p>Der Variablenname <tagname> muss identisch sein mit dem Messstellennamen des AS-Bausteins. Das Zeichen "/" im Namen müssen Sie durch das Zeichen "_" ersetzen, da Windows "/" in Dateinamen nicht zulässt.</p> <p>Beispiel für den Fensternamen einer Messstelle mit dem Variablennamen "Messung/M" @CONL1_Messung_M.pdl</p>
2	<p>Es wird ein Bild "@CONL1_Standard.pdl" aufgerufen, in dem ein oder mehrere Online-Variablen eingetragen werden.</p> <p>Es wird kein Archiv benötigt.</p> <p>x-Achse = 5 min</p> <p>Dies ist die Default-Einstellung, da hier von der Projektierung keine Aufwendungen notwendig sind und keine Namenskonventionen eingehalten werden müssen, um die Trends anzuzeigen.</p> <p>Über die Eigenschaft ReturnPath im Bausteinsymbol werden die Anzahl der Kurven, der Strukturelement-Name sowie die Farbe übergeben.</p>
>2	<p>Es wird ein Bild "@CONL1_Standard.pdl" aufgerufen, in dem ein oder mehrere Archivvariablennamen eingetragen werden.</p> <p>Damit diese Konfiguration ohne weiteren Projektierungsaufwand funktioniert, müssen folgende Konventionen erfüllt sein:</p> <p>Das Archiv muss "Prozesswertarchiv" heißen (unabhängig von der gewählten Spracheinstellung).</p> <p>Das Archiv muss auf dem gleichen Server liegen wie die Variablen selbst.</p> <p>Der Archivvariablenname muss so heißen, wie er standardmäßig beim Erstellen der Archivvariablen angelegt wird.</p> <p>Wenn in der Eigenschaft "tagname" des Bausteinsymbols im eingetragenen Variablenname das Serverprefix enthalten ist, muss im Archivvariablennamen das Serverprefix ohne "://" angegeben werden.</p> <p>Wenn in der Eigenschaft "tagname" der eingetragene Variablenname kein Serverprefix enthalten ist, muss der Archivvariablenname genau so heißen wie der Variablenname.</p> <p>Hinweis: Der Archivvariablenname ist in älteren Versionen der "tagname" ohne die Trennzeichen "/". In Versionen ab V6 ist der Archivvariablenname der "tagname". Die Archivvariable muss diesen Namen haben, damit der Trend im Bildbaustein funktioniert.</p> <p>Über die Eigenschaft ReturnPath im Bausteinsymbol werden die Anzahl der Kurven, der Variablenname und die Farbe übergeben.</p> <p>Die Dauer der X-Achse wird für diesen Modus aus dem Parameter "StandardTrend" gelesen (in min.)</p>

Standard-Trend:	Modus	
	Syntax für ReturnPath: .PV_IN : CO_GREEN ,	Strukturelement-Name mit Punkt für erste Kurve Trennzeichen zwischen Strukturelement-Name und Farbe Farbe für die erste Kurve Komma für weitere Kurve
	Beispiel: .PV_IN:CO_DKGREEN,.SP:CO_BLUE,.LMNR_IN:CO_DKRED (Default-Einstellung für CTRL_PID)	
	<p>Das Beispiel zeigt drei Kurven beim Regler (Istwert, Sollwert, Stellgröße Rückmeldung). Wenn die oben erwähnten 4 Konventionen nicht eingehalten werden können, können Sie die Eigenschaft "ReturnPath" mit weiteren Parametern ergänzen:</p> <p>*tagname: individueller Archivvariablenname *archivname: individueller Archivname (Eingabe ohne „\“) *archivserver: Kurve liegt auf einem Archiv-Server, (Eingabe des Serverprefix' vom Archiv-Server ohne „:“)</p> <p>Beispiel: .PV_IN:CO_DKGREEN,.SP:CO_BLUE,.LMNR_IN:CO_DKRED*tagname: MySpecialTag*archivname:MySpecialArchivname*archivserver: MySpecialArchivServer</p> <p>Hinweis: Es kann hier nur ein "tagname", "archivname", "archivserver" am Ende des Strings angegeben werden. Für den Fall von Multiinstanzen müssen die Archivvariablenamen entsprechend angepasst werden. Die Archivvariablen dürfen sich nur in der Membervariablen unterscheiden.</p> <p>Beispiel für Multiinstanz: Es gibt in einem CFC-Plan "MULTI" zwei MEAS_MON-Bausteine "M1" und "M2". Als Archivvariablenamen für den Messwert "U" werden per Default die Namen "MULTI/M1.U" und "MULTI/M2.U" erzeugt.</p>	
	*asia:	add Serverprefix in Archivvariable Wenn der Parameter *asia: nicht verwendet wird, wird bei vorhandenem Serverprefix in "tagname" dieser auch mit in die Ableitung des Archivvariablenamens eingebracht. Wenn kein Serverprefix in "tagname" vorhanden ist, wird auch bei der Ableitung des Archivvariablenamens kein Serverprefix berücksichtigt. *asia:MyServerPrefix Der Archivvariablenname enthält das serverprefix ohne ":", die Eigenschaft "tagname" des Bausteinsymbols aber nicht. In diesem Fall kann hier das Serverprefix ergänzt werden (Eingabe ohne ":",). *asia: Der Archivvariablenname enthält kein Serverprefix. In diesem Fall kann hier das Serverprefix ausgeblendet werden. Diese Variante ist auch zu verwenden, wenn in "tagname" auf dem Server kein Serverprefix eingetragen ist, da bei einem Aufruf des Bildes vom Client das Serverprefix immer vorhanden ist.

Standard-Trend:	Modus
	<p>Die Archivvariablen müssen Sie in "MULTI.M1_U" und "MULTI.M2_U" umbenennen.</p> <p>Textstring in der Eigenschaft ReturnPath im Bausteinsymbol: .M1_U:CO_DKGREEN,.M2_U:CO_BLUE*tagname:MULTI*archivname:SystemArchive</p> <p>Beispiel für *asia: Serverprefix einblenden: .U:CO_DKGREEN*asia:OS(1)_PC2</p> <p>Beispiel für *asia: Serverprefix ausblenden .U:CO_DKGREEN*asia:</p>

In der folgenden Tabelle werden die notwendigen zusätzlichen Parametrierungen abhängig von folgenden Gesichtspunkten dargestellt:

- Aufbau des Archivvariablennamens und Variablennamens im Bausteinsymbol
- Quelle, aus der der Bildbaustein aufgerufen wird
- Archivvariable auf dem eigenen Archiv-Server

	Server	Client mit Server-Bild	Client mit Client-Bild	Client mit Server-Bild und Archivvariable auf separatem Archiv-Server	Client mit Client-Bild und Archivvariable auf separatem Archiv-Server
Variablenname ohne Serverprefix Archivvariablenname ohne Serverprefix	*asia:	*asia:	nicht möglich, im "tagname" ist immer das Serverprefix	*asia: *archivname: MyArchivserver Prefix	nicht möglich, im "tagname" ist immer das Serverprefix
Variablenname mit Serverprefix Archivvariablenname ohne Serverprefix	*asia:	*asia:	*asia:	*asia: *archivname: MyArchivserver Prefix	*asia: *archivname: MyArchivserver Prefix
Variablenname ohne Serverprefix Archivvariablenname mit Serverprefix	*asia: MyServer Prefix	*asia: MyServer Prefix	nicht möglich, im "tagname" ist immer das Serverprefix	*archivname: MyArchivserver Prefix	nicht möglich, im "tagname" ist immer das Serverprefix
Variablenname mit Serverprefix Archivvariablenname mit Serverprefix	keine Einstellung nötig bei empfohlener Variablen- Projektierung	keine Einstellung nötig bei empfohlener Variablen- Projektierung	keine Einstellung nötig bei empfohlener Variablen- Projektierung	*archivname: MyArchivserver Prefix	*archivname: MyArchivserver Prefix

Bei Aufruf des Prototypbildes "@PG_xxx" wird vom Bausteinsymbol der Wert der Eigenschaft "StandardTrend" und "ReturnPath" an den Bildbaustein übertragen.

Im Prototypbild ist ein Objekt "Trendfunktionen" mit der Eigenschaft "StandardTrend" und "ReturnPath" hinterlegt. Darin werden die Informationen gespeichert und bei Bildanwahl des "@PCS7_Trend.pdl" über das Skript "PCS7_Trend.fct" ausgewertet.

Die gleiche Funktionalität wird in die Loop-Darstellung übertragen.

2.1.9 So projektieren Sie für einen AS-Bausteintyp verschiedene Bausteinsymbole und Bildbausteintypen

Vorgehen

Um eine Variation eines Bausteinsymbols zu einem AS-Baustein anzulegen, müssen Sie lediglich die Eigenschaft "type" am Bausteinsymbol ändern.

Wenn Sie auch eine Variation eines Bildbausteines zu einem AS-Baustein anlegen wollen, müssen Sie einen vom AS-Bausteintyp abweichenden Namen verwenden. Die Variation für "MEAS_MON" ist dann z. B. der Bildbausteinname "MEAS_NEU".

Nehmen Sie folgende Einstellungen am Bausteinsymbol vor:

1. Benennen Sie die Eigenschaft "Servername" in "PCS7 MEAS_NEU Control" um.
2. Legen Sie die neue Eigenschaft "StructureType" an dem Bausteinsymbol an, das den AS-Bausteintyp enthält ("MEAS_MON"). Dies ist notwendig, damit bei dem Wizard "Bildbaustein mit Messstelle verbinden" die Variablenselektion weiterhin funktioniert.

2.1.10 WebClient (Unterschiede zu WinCC)

2.1.10.1 Unterschiede bei der Projektierung von Bildbausteinen

Nachfolgend finden Sie die zu berücksichtigenden Unterschiede zwischen WinCC und WebClient.

2.1.10.2 Bildnamen

Adressierung

In WinCC-Skripten werden Bildnamen absolut oder relativ adressiert.

Beispiele:

- absolut

@screen.@win12:@1001.@top09:@pg_elap_cnt.OverviewWindow:@PG_ELAP_CNT_OverView.pdl

- relativ

@PG_ELAP_CNT_OverView.pdl

Beim WebClient funktioniert nur die relative Adressierung aus dem Kontext heraus, in dem das Skript läuft.

Weiterhin muss bei der WebClient-Skriptverarbeitung der Bildfensternamen und nicht der Bildname als Adresse angegeben werden.

Beispiel:

`SetPropChar(lpszPictureName,lpszObjectName,lpszPropertyName,szValue);`

WinCC: der lpszPictureName ist ein Zeiger auf den Bildnamen

WebClient: der lpszPictureName ist ein Zeiger auf den Bildfensternamen

Ausnahme sind das eigene Bild und das Vorgängerbild (ParentPicture), hier kann auch auf dem WebClient der Bildname als Zeiger verwendet werden. Wir empfehlen, immer den Bildfensternamen als Zeiger zu verwenden, da dieser eindeutig ist. Der Bildname kann mehrdeutig sein.

Beispiel für die relative Bildadressierung:

In einem Bild @PG_xxx.pdl gibt es ein Bildfenster "View", in dem z. B. die Standardsicht eines Bildbausteins angezeigt wird.

Wenn aus dem Bild in der Standardsicht, das in dem Bildfenster "View" geladen ist, ein anderes Bildfenster im Bild @PG_xxx.pdl adressiert wird, z. B.

"OperationWindow" zum Öffnen einer Bedienbox, ist die relative Bildfensteradresse folgendermaßen zu ermitteln:

sprintf(szPictureName,"../OperationWindow");

2.1.10.3 Groß- /Kleinschreibung bei Dateinamen

Dateinamen, z. B. bei Zeigern auf Bildnamen, werden bei WinCC generell in Großbuchstaben geliefert, z. B. @PG_MEAS_MON.PDL.

Auf dem WebClient werden die Dateinamen so geliefert, wie sie erstellt wurden, z. B. @Pg_Meas_Mon.pdl.

Bei String-Vergleichen müssen hier möglicherweise Anpassungen vorgenommen werden.

Beispiel:

```
( strcmp( GetPictureUp(IpszPictureName,IpszObjectName),
"@KEEPVISIBLEON.EMF") == 0 )
```

Diese Anpassung funktioniert in WinCC ohne Einschränkung. Auf dem Web funktioniert diese Anpassung nur, wenn der Dateiname ausschließlich aus Großbuchstaben besteht.

Erforderliche Anpassung:

```
( strcmp( _strlwr(GetPictureUp(IpszPictureName,IpszObjectName)),
"@keepvisibleon.emf") == 0 )
```

2.1.10.4 Laden von Bildern in Bildfenstern

Das Laden von Bildern in Bildfenstern unterscheidet sich bei WinCC und auf dem WebClient in folgenden Punkten:

- WinCC
In WinCC können Sie nach dem Laden eines Bildes in einem Bildfenster, z. B. mit SetVisible, die darin enthaltenen Objekte direkt über die Skriptverarbeitung adressieren und die Eigenschaften verändern.
- WebClient
Auf dem WebClient erfolgt das Laden von Bildern asynchron. Deswegen muss in Skripten nach dem Laden eines Bildes die weitere Skriptverarbeitung verzögert werden, wenn im gleichen Skript Objekte des Bildes adressiert werden.
Hierfür wird für den WebClient die Funktion **WaitForDocumentReady("Bildfenstername");** bereitgestellt.
Die Funktion verzögert den Skriptablauf, bis der als Parameter angegebene Bildfenstername ein geladenes Bild zurückmeldet.

2.1.10.5 Bildabwahl innerhalb von Skripten

Die Bildabwahl innerhalb von Skripten unterscheidet sich bei WinCC und auf dem WebClient in folgenden Punkten:

- WinCC
In WinCC kann das "Unsichtbar setzen" eines Bildes aus einem Skript, das im Kontext dieses Bildes läuft, an einer beliebigen Stelle des Skripts ausgeführt werden, da die Skriptverarbeitungs-Task unabhängig vom geschlossenen Bild das Skript weiter verarbeitet.
- WebClient
Auf dem WebClient muss das "Unsichtbar setzen" zwingend als letzte Aktion im Skript ausgeführt werden.
Skriptbefehle, die nach dem "Unsichtbar setzen" stehen, werden nicht mehr ausgeführt, da das Skript an dieser Stelle beendet wird.

2.1.10.6 Unterscheidung der Runtime-Umgebung WinCC <-> Web

In den Skripten von Grafikbildern oder Projektfunktionen ist es notwendig, unterscheiden zu können, ob die Runtime-Umgebung WinCC oder der WebClient ist.

Hierzu gibt es die folgenden Compiler-Anweisungen:

- `#ifdef RUN_ON_WEBNAVIGATOR`
- `#ifndef RUN_ON_WEBNAVIGATOR` (die Negation)

Damit können beispielsweise folgende Unterschiede zwischen WinCC und WebClient berücksichtigt werden:

- Skriptverzögerung mit `WaitForDocumentReady`
- unterschiedliche Bildadressierung
- unterschiedliche Funktionsnamen bei Leittechnikfunktionen
- Funktionen, die im Web nicht unterstützt werden

Beispiel aus dem Skript PCS7_ChangeView

```
#ifdef RUN_ON_WEBNAVIGATOR
    SetPropChar("../", "View", "PictureName", szViewName);
    WaitForDocumentReady("../View");
#else
    SetPropChar(IpszParent, "View", "PictureName", szViewName);
#endif
```

Hinweis

Die Syntax des Codeteils für Web wird nicht beim Compilieren des WinCC-Skripts überprüft, sondern nur beim Veröffentlichen der Bilder.

2.1.10.7 Funktionsnamen in WinCC/Web

Liste der unterstützten Funktionen

Funktionen, die eine Aktion auslösen, wurden mit anderen Funktionsnamen für das Web nachgebildet, damit die Aktion im Webnavigator-Client ausgelöst wird und nicht in WinCC-Runtime.

Beispiel:

SSMRTChangeWorkfield ändert das Arbeitsfenster in WinCC-Runtime und SSMChangeWorkfield ändert das Arbeitsfenster beim WebClient.

Eine Liste der unterstützten Funktionen und deren Namen finden Sie bei den entsprechenden Produktbeschreibungen. Zum Produkt "Faceplate Designer" werden folgende Funktionen unterstützt:

```
void PCS7_ChangeView(char* IpszPictureName, char* IpszObjectName);
BOOL PCS7_CheckPermission(char* IpszTagName, DWORD dwLevel);
void PCS7_UpdateGroupPermission(char* IpszPictureName);
void PCS7_UpdateGroupTagName(char* IpszPictureName, char* IpszObjectName, char*
value);
void PCS7_UpdateLoopTagName(char* IpszPictureName, char* IpszObjectName, char*
value);
void PCS7_UpdatePermission_V6(char* IpszParentPictureName, int CallFrom, char*
IpszPictureName);
void PCS7_OpenGroupDisplay_V6(char *IpszPictureName, char *IpszObjectName );
void PCS7_OpenLoopDisplay_V6(char* IpszPictureName);
void PCS7_OpenInputBoxBin_V6(char *IpszPictureName,char *IpszObjectName, int
CallFrom);
void PCS7_1vnStati_Variable_Changed_V6(char* IpszPictureName, char* IpszObjectName,
char* IpszPropertyName, double value);
void PCS7_OperationLog_V6(char* IpszPictureName, double dOldValue, double
dNewValue, char* IpszOperationText, char* IpszUnit);
void PCS7_Trend_V6(char* IpszPictureName, char* IpszObjectName, char*
IpszPropertyName, char * ArchivVar, char* OnlineVar, DWORD TrendColor, int CallFrom);
void PCS7_UpdateGroupTagName_V6(char* IpszPictureName, char* IpszObjectName,
char* value);
void PCS7_UpdateLoopTagName_V6(char* IpszPictureName, char* IpszObjectName, char*
IpszTagName);
void PCS7_UpdateBarLimits_V6(char* IpszPictureName, char* IpszObjectName, int
CallFrom);
void PCS7_UpdateBar_V6(char* IpszPictureName, char* IpszObjectName, int CallFrom);
void PCS7_OpenInputBoxAnalog_V6(char *IpszPictureName,char *IpszObjectName,int
CallFrom);
```

```
void PCS7_OpenGroupDisplay_I_V6(char *IpszPictureName, char *IpszObjectName,
char*IpszInterlokName);

void PCS7_OpenComboBox_V6(char *IpszPictureName,char *IpszObjectName,int
CallFrom);

void PCS7_OpenCheckBox_V6(char *IpszPictureName,char *IpszObjectName, int
CallFrom);

void PCS7_Open3ComboBox_V6(char *IpszPictureName,char *IpszObjectName,int
CallFrom);

void PCS7_Format_V6(char* IpszPictureName, char* IpszObjectName, int CallFrom,char*
IpszParent);

void PCS7_Combo_OK_V6(char* IpszPictureName, char* IpszObjectName, char*
IpszPropertyName);

void PCS7_Check_OK_V6(char* IpszPictureName, char* IpszObjectName, char*
IpszPropertyName);

void PCS7_Binary_OK_V6(char* IpszPictureName, char* IpszObjectName, char*
IpszPropertyName);

void PCS7_AnalogPercent_V6(char* IpszPictureName, char* IpszObjectName, char*
IpszPropertyName,double Prozent);

void PCS7_Analog_OK_V6(char* IpszPictureName, char* IpszObjectName, char*
IpszPropertyName);

void PCS7_3Combo_OK_V6(char* IpszPictureName, char* IpszObjectName, char*
IpszPropertyName);

void PCS7_2Stati_Variable_Changed_V6(char* IpszPictureName, char* IpszObjectName,
char* IpszPropertyName, double value);
```

2.1.10.8 Global Script

Für den WebClient wird "Global Script" nicht unterstützt, da es hier keine eigene Applikation "Script.exe" gibt.

Globale Funktionen sind als Projektfunktionen abzulegen. Sie werden beim Veröffentlichen ebenfalls übersetzt.

2.1.10.9 VBS-Skript

Wir empfehlen beim Erstellen von neuen Funktionen, diese in VBS-Skripten zu erstellen, da Sie bei VBS-Skripten keine Unterschiede zwischen WinCC und WebClient berücksichtigen müssen.

2.1.10.10 Hinweise

Im Unterschied zu WinCC gibt es auf dem WebClient keine globalen bildübergreifenden Variablen in C-Skripten.

Wir empfehlen auf zyklische, synchrone Funktionen zu verzichten, da die Laufzeit am Web Client größer ist als bei WinCC.

2.1.11 Sprachumschaltung

Die Vorlagen des Faceplate Designers sind dreisprachig angelegt: in Deutsch, Englisch und Französisch. Nach einer Sprachumschaltung in WinCC werden die Texte in der eingestellten Sprache angezeigt. Wenn Sie weitere Textelemente in ein Bild einbauen und eine Sprachumschaltung wünschen, müssen Sie diese Texte in allen gewünschten Sprachen im Graphics Designer eingeben. Die Sprachumschaltung erfolgt im Graphics Designer über den Menübefehl **Ansicht > Sprache... > Sprache wählen**.

2.1.12 Texte für Analogwert- und Binärwert-Bedienung aus ES

Einstellen der Sprache

Für die Beschriftung von Binäranzeigen, Analogwertanzeigen, Kombinationsfeldern, Optionskästchen und Bedienprotokollen werden die Texte aus den Bausteininstanzen von den Parameter-Attributen gelesen.

Beachten Sie dabei, dass die Default-Texte im ES für die Attribute "s7_shortcut", "s7_unit", "S7_string_0" und "S7_string_1" alle in Englisch projiziert sind.

In den Bildbausteinen wurden bisher die ES-Texte nur im Bedienprotokoll verwendet. In den Anzeigen des Bildbausteins wurden bislang die Texte in WinCC gebildet und waren dreisprachig.

Für die Migration müssen diese Texte im ES, bei Bedarf unter Verwendung des IEA, in die entsprechend gewünschte Sprache umgesetzt werden.

Mehrsprachigkeit der Texte

Wenn mehrsprachige Texte benötigt werden, können Sie die Texte mit dem Text-Library-Editor in WinCC in andere Sprachen übersetzen. Stellen Sie dazu im SIMATIC Manager genau die Sprache als Standardsprache für Anzeigegeräte ein, in der die Bedien- und Anzeigetexte in STEP 7 und im ES projiziert sind (im Normalfall "Englisch"). Dadurch ist gewährleistet, dass die übersetzten Texte beim nächsten Übersetzen der OS nicht überschrieben werden.

Wichtiger Hinweis

Da die Default-Texte im ES in Englisch existieren, ist es wichtig, dass Sie bei einem Projekt, das nicht in Englisch projiziert wird, eine Projektbibliothek für die AS-Bausteine anlegen und dort die englischen Default-Texte auf die gewünschte Sprache umstellen.

Beachten Sie hierbei, dass die neuen Texte nicht länger als die Default-Texte sein dürfen. Wenn Sie längere Texte nicht vermeiden können, muss am Bildbaustein geprüft werden, ob der Text noch korrekt angezeigt wird.

Hinweise zu Erstellung von Projektbibliotheken finden Sie im ES-Projektierungshandbuch.

Das Attribut "s7_unit" muss für die Übersetzung nicht in Betracht gezogen werden, da hier als Vorbesetzung entweder Leerzeichen oder international gebräuchliche Kurzbezeichnungen verwendet wurden.

Liste der anzupassenden Parameter-Attribute

Baustein	Parameter	S7_shortcut	S7_string_0	S7_string_1
CTRL_PID	AUT_ON_OP		Mode= Manual	Mode= Auto
	DEADB_W	Deadband		
	ER_HYS	ER hysteresis		
	ERH_ALM	ER: HH alarm		
	ERL_ALM	ER: LL alarm		
	GAIN	GAIN		
	HYS	Hysteresis		
	LMNR_IN	OUT		
	M_SUP_AH		Suppress HH=No	Suppress HH=Yes
	M_SUP_AL		Suppress LL=No	Suppress LL=Yes
	M_SUP_ER		Enable ER Alarm	Suppr. ER Alarm
	M_SUP_WH		Suppress H=No	Suppress H=Yes
	M_SUP_WL		Suppress L=No	Suppress L=Yes
	MAN_HLM	MAN high limit		
	MAN_LLM	MAN low limit		
	MAN_OP	MAN		
	MO_PVHR	Bar HL		
	MO_PVLR	Bar LL		
	OOS		In Service	Out of Service
	OPTI_EN		Disable Optimiz.	Enable Optimiz.
	PV_IN	PV		
	PVH_ALM	PV: HH alarm		
	PVH_WRN	PV: H alarm		
	PVL_ALM	PV: LL alarm		
	PVL_WRN	PV: L alarm		
	SP_HLM	SP high limit		
	SP_LLM	SP low limit		
	SP_OP	SP		
	SP_TRK_ON		No Tracking	Track SP:= PV
	SPBUMPON		SP may bump	SP bumpless
	SPDRLM	Neg. ramp		
	SPEXTSEL_OP		SP= Internal	SP= External
	SPRAMPOF		SP Ramp On	SP Ramp Off
	SPURLM	Pos. ramp		
	TM_LAG	Lag time		
	TN	TI		
	TV	TD		

Baustein	Parameter	S7_shortcut	S7_string_0	S7_string_1
CTRL_S	AUT_ON_OP		Mode= Manual	Mode= Auto
	BREAK_TM	Break time		
	DEADB_W	Deadband		
	ER_HYS	ER hysteresis		
	ERH_ALM	ER: HH alarm		
	ERL_ALM	ER: LL alarm		
	GAIN	GAIN		
	HYS	Hysteresis		
	LMNDN_OP		Stop	Close
	LMNR_IN	OUT		
	LMNUP_OP		Stop	Open
	M_SUP_AH		Suppress HH=No	Suppress HH=Yes
	M_SUP_AL		Suppress LL=No	Suppress LL=Yes
	M_SUP_ER		Enable ER Alarm	Suppr. ER Alarm
	M_SUP_WH		Suppress H=No	Suppress H=Yes
	M_SUP_WL		Suppress L=No	Suppress L=Yes
	MAN_HLM	MAN high limit		
	MAN_LLM	MAN low limit		
	MAN_OP	MAN		
	MO_PVHR	Bar HL		
	MO_PVLR	Bar LL		
	MTR_TM	Motor time		
	OOS		In Service	Out of Service
	OPTI_EN		Disable Optimiz.	Enable Optimiz.
	PULSE_TM	Pulse time		
	PV_IN	PV		
	PVH_ALM	PV: HH alarm		
	PVH_WRN	PV: H alarm		
	PVL_ALM	PV: LL alarm		
	PVL_WRN	PV: L alarm		
	RESET		Do Nothing	Reset QMSS_ST
	SP_HLM	SP high limit		
	SP_LLM	SP low limit		
	SP_OP	SP		
	SP_TRK_ON		No Tracking	Track SP:= PV
	SPBUMPON		SP may bump	SP bumpless
	SPDRLM	Neg. ramp		
	SPEXTSEL_OP		SP= Internal	SP= External
	SPRAMPOF		Ascent limit=On	Ascent limit=Off
	SPURLM	Pos. ramp		
	TM_LAG	Lag time		
	TN	TI		
	TV	TD		

Baustein	Parameter	S7_shortcut	S7_string_0	S7_string_1
DIG_MON	I		Off	On
	SUPPTIME	Suppress time		
	OOS		In Service	Out of Service
DOSE	ACK_TOL_OP		0	Acknowl.
	CANCEL_OP		0	Cancel
	COMP_CHG		Comp. change=Off	Comp. change=On
	DRIB_COR		Dribb. corr.=Off	Dribb. corr.=On
	DRIBB	Dribbling init.		
	DRIBBMAX	Max. dribbling		
	M_SUP_1		Suppr normal =No	Suppr normal=Yes
	M_SUP_2		Suppr over =No	Suppr over =Yes
	M_SUP_3		Suppr under =No	Suppr under =Yes
	MO_PVHR	Bar UL		
	MO_PVLR	Bar LL		
	OOS		In Service	Out of Service
	PAUSE_OP		Process=Continue	Process=Pause
	PDOS_TME	Postdose time		
	POSTDOSE		0	Postdose
	PV_IN	PV		
	RELAXTME	Relax time		
	REVERSE		Reverse=No	Reverse=Yes
	SP_HLM	SP high limit		
	SP_LLM	SP low limit		
	SP_OP	Setpoint		
	SPBUMPON		SP may bump	SP bumpless
	SPEXTSEL_OP		SP= Internal	SP= External
	START_OP		0	Dose=Start
	TOL_N	Lower tol. band		
TOL_P	Upper tol. band			
ELAP_CNT	HOURS	Hours		
	HOURS_AH	HH alarm		
	HOURS_OP	Preset value		
	HOURS_WH	H alarm		
	M_SUP_AH		Suppress HH=No	Suppress HH=Yes
	M_SUP_WH		Suppress H=No	Suppress H=Yes
	MO_HOUHR	Bar UL		
	MO_HOULR	Bar LL		
	OOS		In Service	Out of Service
	TRACK_OP		0	Preset

Baustein	Parameter	S7_shortcut	S7_string_0	S7_string_1
FMCS_PID	AUT_ON_OP		Mode= Manual	Mode= Auto
	BREAK_TM	Break time		
	DEADB_W	Deadband		
	GAIN	Gain		
	H_ALM	HH alarm		
	H_WRN	H alarm		
	HYS	Hysteresis		
	L_ALM	LL alarm		
	L_WRN	L alarm		
	LMN	OUT		
	LMN_HLM	LMN high limit		
	LMN_LLM	LMN low limit		
	LMN_OP	MAN		
	LMN_SAFE	LMN safety		
	LMNDN_OP		Stop	Close
	LMNUP_OP		Stop	Open
	M_SUP_AH		Suppress HH=No	Suppress HH=Yes
	M_SUP_AL		Suppress LL=No	Suppress LL=Yes
	M_SUP_WH		Suppress H=No	Suppress H=Yes
	M_SUP_WL		Suppress L=No	Suppress L=Yes
	MO_PVHR	Bar UL		
	MO_PVLR	Bar LL		
	MTR_TM	MTR time		
	OOS		In Service	Out of Service
	OP_SEL		OP operation=Off	OP operation=On
	OPTI_EN		Optim.=disable	Optim.=enable
	PULSE_TM	Pulse time		
	PV	PV		
	SDB_SEL		SDBParameter=On	SDBParameter=Off
	SP_HLM	SP high limit		
	SP_LLM	SP low limit		
	SP_OP	SP		
	SP_TRK_ON		SP track=Off	SP track=On
	SPBUMPON		SP may bump	SP bumpless
	SPEXTSEL_OP		SP= Internal	SP= External
	TD	TD		
	TI	TI		
	TM_LAG	Time lag		

Baustein	Parameter	S7_shortcut	S7_string_0	S7_string_1
FMT_PID	AUT_ON_OP		Mode= Manual	Mode= Auto
	BREAK_TM	Break time		
	D_F	deriv. fac.		
	DEADB_W	Deadband		
	GAIN	Gain		
	H_ALM	HH alarm		
	H_WRN	H alarm		
	HYS	Hysteresis		
	L_ALM	LL alarm		
	L_WRN	L alarm		
	LMN	OUT		
	LMN_HLM	LMN high limit		
	LMN_LLM	LMN low limit		
	LMN_OP	MAN		
	LMN_SAFE	LMN safety		
	LMNDN_OP		Stop	Close
	LMNUP_OP		Stop	Open
	M_SUP_AH		Suppress HH=No	Suppress HH=Yes
	M_SUP_AL		Suppress LL=No	Suppress LL=Yes
	M_SUP_WH		Suppress H=No	Suppress H=Yes
	M_SUP_WL		Suppress L=No	Suppress L=Yes
	MO_PVHR	Bar UL		
	MO_PVLR	Bar LL		
	MTR_TM	MTR time		
	OOS		In Service	Out of Service
	PFAC_SP	Gain		
	PULSE_TM	Pulse time		
	PV	PV		
	SDB_SEL		SDBParameter=On	SDBParameter=Off
	SP_HLM	SP high limit		
	SP_LLM	SP low limit		
	SP_OP	SP		
	SP_TRK_ON		SP track=Off	SP track=On
	SPBUMPON		SP may bump	SP bumpless
	SPEXTSEL_OP		SP= Internal	SP= External
	TD	TD		
	TI	TI		

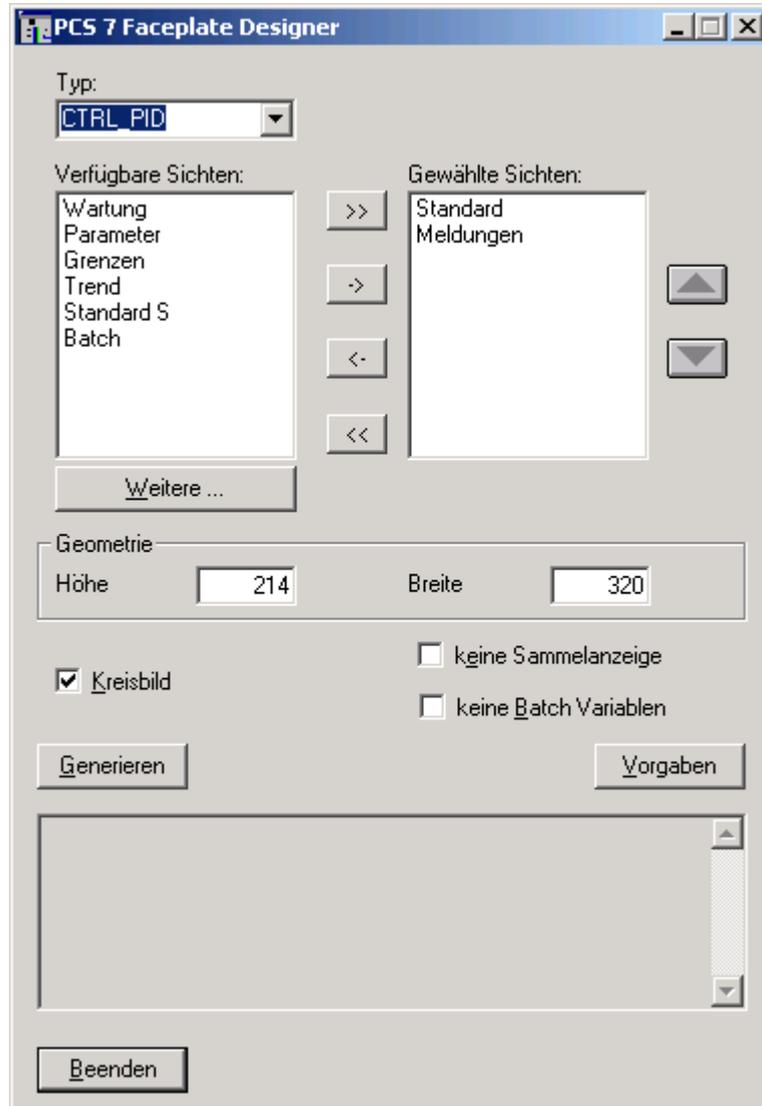
Baustein	Parameter	S7_shortcut	S7_string_0	S7_string_1
INTERLOK	I1_1		0	in_1
	I1_2		0	in_2
	I1_3		0	in_3
	I1_4		0	in_4
	I1_5		0	in_5
	I2_1		0	in_6
	I2_2		0	in_7
	I2_3		0	in_8
	I2_4		0	in_9
	I2_5		0	in_10
	OVERWRITE		Overwrite=Off	Overwrite=On
	MEAS_MON	HYS	Hysteresis	
M_SUP_AH			Suppress HH=No	Suppress HH=Yes
M_SUP_AL			Suppress LL=No	Suppress LL=Yes
M_SUP_WH			Suppress H=No	Suppress H=Yes
M_SUP_WL			Suppress L=No	Suppress L=Yes
MO_PVHR		Bar UL		
MO_PVLR		Bar LL		
OOS			In Service	Out of Service
U		PV		
U_AH		HH alarm		
U_AL		LL alarm		
U_WH		H alarm		
U_WL		L alarm		
MOTOR		AUT_ON_OP		Mode=Manual
	MAN_ON		Motor=Stop	Motor=Start
	MONITOR		Monitoring=Off	Monitoring=On
	OOS		In Service	Out of Service
	RESET		0	Error=Reset
	TIME_MON	Mon. time		
MOT_REV	AUT_ON_OP		Mode=Manual	Mode=Auto
	FORW_ON		0	Motor=Forward
	MONITOR		Monitoring=Off	Monitoring=On
	MOT_OFF		0	Motor=Off
	OOS		In Service	Out of Service
	RESET		0	Error=Reset
	REV_ON		0	Motor=Reverse
	TIME_OFF	Mon. time off		
	TIME_ON	Mon. time on		

Baustein	Parameter	S7_shortcut	S7_string_0	S7_string_1
MOT_SPED	AUT_ON_OP		Mode=Manual	Mode=Auto
	MONITOR		Monitoring=Off	Monitoring=On
	MOT_OFF		0	Motor=Off
	OOS		In Service	Out of Service
	RESET		0	Error=Reset
	SP1_ON		0	Motor=Speed 1
	SP2_ON		0	Motor=Speed 2
	TIME_MON	Mon. time		
OP_A	BTRACK		Bumpless=Off	Bumpless=On
	U	U		
OP_A_LIM	BTRACK		Bumpless=Off	Bumpless=On
	U	U		
OP_A_RJC	BTRACK		Bumpless=Off	Bumpless=On
	U	U		
OP_D	BTRACK		Bumpless=Off	Bumpless=On
	I0		Off	On
OP_D3	BTRACK		Bumpless=Off	Bumpless=On
	I1		0	Switch 1
	I2		0	Switch 2
	I3		0	Switch 3
OP_TRIG	I0		0	Reset
RATIO_P	IN_EX		Internal	External
	MO_U1HR	Bar UL		
	MO_U1LR	Bar LL		
	U1	U1		
	U2	U2		
	U2_HL	High limit U2		
	U2_LL	Low limit U2		
	V_HL	High limit V		
	V_LL	Low limit V		

Baustein	Parameter	S7_shortcut	S7_string_0	S7_string_1
SWIT_CNT	M_SUP_AH		Suppress HH=No	Suppress HH=Yes
	M_SUP_WH		Suppress H=No	Suppress H=Yes
	MO_VHR	Bar UL		
	MO_VLR	Bar LL		
	OOS		In Service	Out of Service
	TRACK_OP		0	Preset
	V	V		
	VAH	HH alarm		
	VTRACK_OP	Preset value		
	VWH	H alarm		
VALVE	AUT_ON_OP		Mode=Manual	Mode=Auto
	MAN_OC		Valve=Close	Valve=Open
	MONITOR		Monitoring=Off	Monitoring=On
	OOS		In Service	Out of Service
	RESET		0	Error=Reset
	TIME_MON	Mon. time		
VAL_MOT	AUT_ON_OP		Mode=Manual	Mode=Auto
	CLOS_VAL		0	Valve=Close
	MONITOR		Monitoring=Off	Monitoring=On
	OOS		In Service	Out of Service
	OPEN_VAL		0	Valve=Open
	RESET		0	Error=Reset
	SS_POS			
	STOP_VAL		0	Valve=Stop
	TIME_OFF	Mon. time off		
	TIME_ON	Mon. time on		

2.2 So arbeiten Sie mit dem Faceplate Designer

Die EXE und die DLLs zum Faceplate Designer V6.0 sind im Verzeichnis "..\WinCC\bin\FaceplateDesigner" abgelegt.



Vorgehen

1. Starten Sie im WinCC Explorer den Faceplate Designer.
2. Tragen Sie im Kombinationsfeld "Typ" den Namen für den neuen Bildbaustein ein.
 Sie können den Namen aus der Klappliste der Strukturtypen vom WinCC-Variablenhaushalt auswählen oder einen neuen Namen vergeben.
3. Im rechten Fenster "Gewählte Sichten" tragen Sie die einzelnen Sichten des Bildbausteins ein, die vom Faceplate Designer angelegt werden sollen. Wählen Sie diese aus dem linken Fenster "Verfügbare Sichten" aus oder legen Sie über die Schaltfläche "Weitere" neue Namen von Sichten an. Neue Sichten-Namen müssen Sie bei Bedarf auf Mehrsprachigkeit für die Anzeige der Sichten-Liste zusätzlich projektieren. Beim Generieren dieser neuen Sicht erhalten Sie hierzu einen Hinweis.



Die Vorgehensweise wird im Beispiel beschrieben (Kapitel 2.2.1).

4. Projektieren Sie die Bildhöhe und die Bildbreite der Sichten über die "Geometrie".
 Voreingestellt ist die Defaultgröße wie z. B. auch die Batch-Sicht oder die Meldesicht.

Beachten Sie beim Projektieren:

- Die Sichten der Bilder werden im Gruppenbild und im Kreisbild ausgegeben.
- Für Alarm, Batch und Trend werden universal gültige Bilder verwendet.
- Die Kreisbildgenerierung kann unterdrückt werden, gleichzeitig wird auch die Taste für die Anwahl des Kreisbildes unsichtbar geschaltet.
- Der Faceplate Designer wird in der Sprache angezeigt, die im WinCC Explorer eingestellt ist.
- Der Faceplate Designer ermittelt beim Öffnen die aktuelle WinCC-Oberflächensprache und präsentiert seinen Dialog in derselben Sprache.
- Nachfolgende Umschaltungen der WinCC-Oberflächensprache bei geöffnetem Dialogfeld werden ignoriert.
- Die Schaltfläche "Vorgaben" stellt die Grundeinstellung des Faceplate Designer wieder her.

- Mit der Schaltfläche "Generieren" starten Sie die Generierung der Bilder. So werden für einen Bildbausteinnamen "TEST" mit einer ausgewählten Sicht "Selected Views = Standard" folgende Bilder generiert:

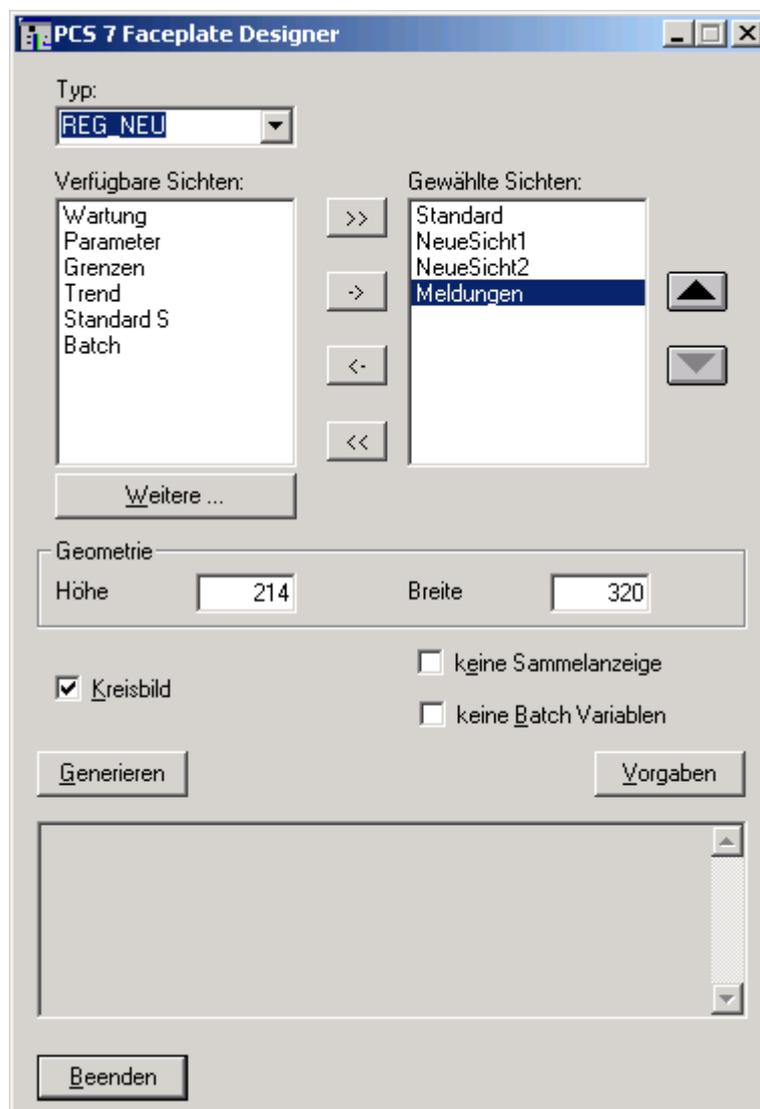
neues Bild	aus dem Vorlagenbild
@PG_TEST.pdl	@PG_%TYPE%.pdI
@PL_TEST.pdl	@PL_%TYPE%.pdI
@PG_TEST_OVERVIEW.pdl	@PG_%Type%_OVERVIEW.pdl
@PG_TEST_VIEWLIST.pdl	@PG_%Type%_VIEWLIST.pdl
@PG_TEST_STANDARD.pdl	@PG_%Type%_%View%.pdI

2.2.1 Beispiel: Erstellung eines neuen Bildbausteins für einen Regler

2.2.1.1 So erstellen Sie Vorlagen

Vorgehen

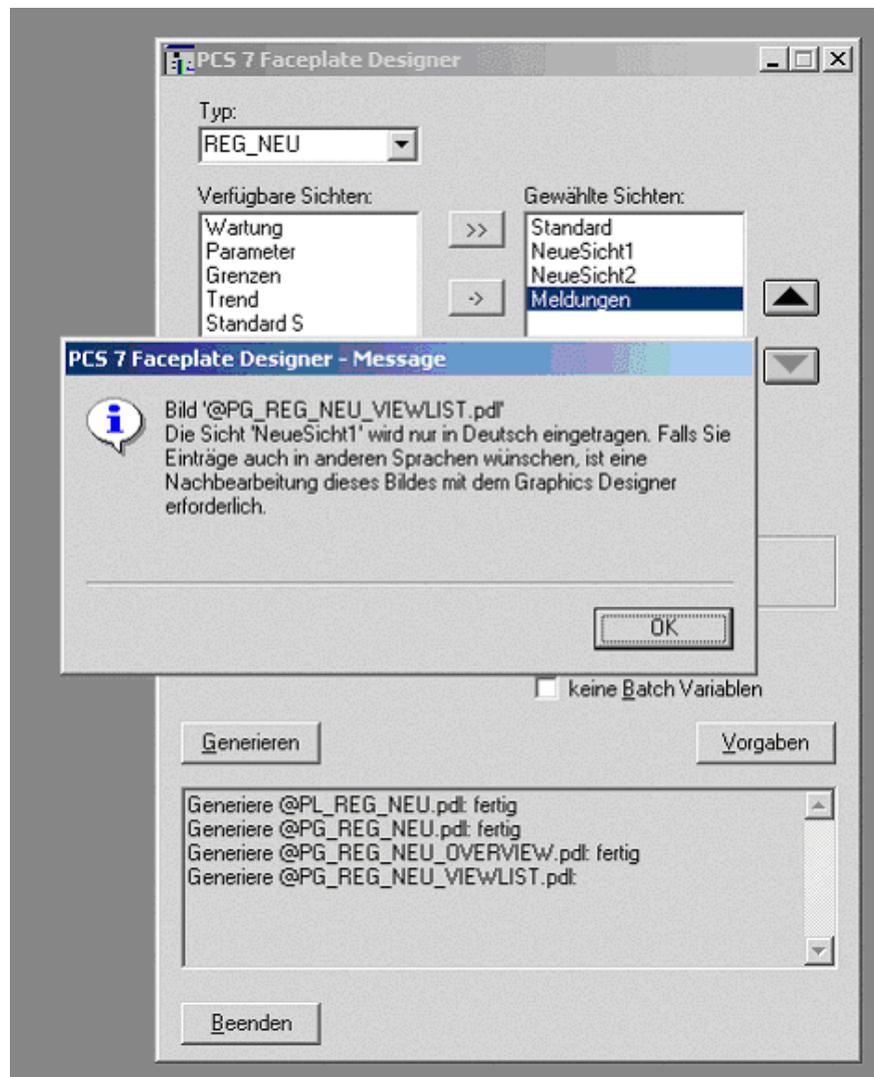
1. Starten Sie im WinCC Explorer den Faceplate Designer.
2. Tragen Sie im Eingabefeld "Typ" "REG_NEU" ein.
3. Klicken Sie in "Verfügbare Sichten" auf die Schaltfläche "Weitere...." und legen Sie zwei neue Sichten an: "NeueSicht1" und "NeueSicht2".
4. Markieren Sie die neuen Sichten und klicken Sie auf die Schaltfläche "->". Die neuen Sichten werden nach "Gewählte Sichten" übertragen.
5. Schieben Sie bei Bedarf die Sicht "Meldungen" mit der Schaltfläche "Pfeil nach unten" ans Ende der Liste.



6. Klicken Sie auf die Schaltfläche "Generieren".
 Folgende Vorlagenbilder werden erzeugt:

neues Bild	aus dem Vorlagenbild
@PG_REG_NEU.pdl	@PG_%TYPE%.pdl
@PL_REG_NEU.pdl	@PL_%TYPE%.pdl
@PG_REG_NEU_OVERVIEW.pdl	@PG_%Type%_OVERVIEW.pdl
@PG_REG_NEU_VIEWLIST.pdl	@PG_%Type%_VIEWLIST.pdl
@PG_REG_NEU_STANDARD.pdl	@PG_%Type%_%View%.pdl
@PG_REG_NEU_NEUESICHT1.pdl	@PG_%Type%_%View%.pdl
@PG_REG_NEU_NEUESICHT2.pdl	@PG_%Type%_%View%.pdl

Beim Generieren der neuen Sichten wird ein Hinweis zur Mehrsprachigkeit geöffnet.



Im unteren Ausgabefenster werden die generierten Dateien aufgelistet.

2.2.1.2 Vorlagen bearbeiten

Nach der Generierung der Vorlagenbilder können Sie mit der Bearbeitung der einzelnen Sichten beginnen:

```
@PG_REG_NEU_VIEWLIST.pdl
@PG_REG_NEU_STANDARD.pdl
@PG_REG_NEU_NEUESICHT1.pdl
@PG_REG_NEU_NEUESICHT2.pdl
```

Das Bild "@PG_REG_NEU_VIEWLIST.pdl" muss nur dann bearbeitet werden, wenn der Bildbaustein auch in anderen Sprachen als in der Erstersprache genutzt werden soll.

Das Beispiel wurde in deutscher Sprache generiert und zeigt im Graphics Designer bei der Spracheinstellung "Deutsch" folgendes Bild.



Wenn Sie im Graphics Designer auf die englische Sprache umschalten, wird folgendes Bild angezeigt:



Hier müssen Sie nun für die statischen Texte, deren Objektnamen "NeueSicht1" und "NeueSicht2" sind, die Eigenschaften "Text" anpassen, z. B. "NewView1" und "NewView2".

Das Gleiche ist auch für die französische Sprache vorzunehmen.

2.2.1.3 So bearbeiten Sie das Vorlagenbild @PG_REG_NEU_STANDARD.pdl

Vorgehen

Um den Standard-Bildbaustein des Reglers zu modifizieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie "@PG_REG_NEU_STANDARD.pdl".
2. Löschen Sie die Objekte "@Level6" und "@Level5" (siehe hierzu auch die Beschreibung "Bedienberechtigungen projektieren").
3. Öffnen Sie den Standard-Bildbaustein "@PG_CTRL_PID" für den Regler, markieren und kopieren Sie alles und fügen Sie es in das Bild "@PG_REG_NEU_STANDARD.pdl" ein.

Die Elemente "@Level6" und "@Level5", die für die Bedienberechtigung vorgesehen sind, und die entsprechenden Direktverbindungen auf die bedienbaren Elemente werden mitkopiert.

Hinweis

Beim Kopieren von Objekten werden Sonderzeichen beim Objektnamen der Kopie entfernt. Deswegen heißen die Objekte "@Level6" und "@Level5" im Bild "@PG_REG_NEU_STANDARD.pdl" zunächst "Level6" und "Level5" und müssen in die ursprünglichen Namen umbenannt werden, da nur diese per Skript mit Werten versorgt werden.

Anschließend können Sie die Objekte im Bild verändern, neue Objekte hinzufügen oder auch Objekte löschen.

Beachten Sie beim Löschen von Objekten, dass diese über eine Direktverbindung eine Information bekommen oder weiterleiten. Siehe hierzu "Dokumentation der Standardbildbausteine". Dort sind die Direktverbindungsketten aufgelistet, einschließlich der Objekte, die Informationen per Direktverbindung weiterleiten. In der Regel sind das folgende Objekte:

- "@Level6" und "@Level5" zum Übertragen der Bedienberechtigungen
- das Objekt "Format" zum Übertragen von instanzspezifischen Zahlenformaten (siehe hierzu die Beschreibung der Basiselemente ab Kapitel 2.3).

2.2.1.4 So bearbeiten Sie das Bild @PG_REG_NEU_NEUESICHT1.pdl

Aus dem Vorlagenbild "@PCS7Elements.pdl" können Sie Objekte einfügen und dynamisieren.

Vorgehen

1. Öffnen Sie im WinCC Graphics Designer die Bilder "@PG_REG_NEU_NEUESICHT1.pdl" und "@PCS7Elements.pdl".
2. Wählen Sie den Menübefehl **Fenster > Nebeneinander** und ordnen Sie die Bilder nebeneinander an.
3. Kopieren Sie die benötigten Bildelemente aus dem Bild "@PCS7Elements.pdl" in das Bild "@PG_REG_NEU_NEUESICHT1.pdl".
4. Vergeben Sie sinnvolle objektbezogene Objektnamen.
5. Stellen Sie die Positionen für die einzelnen Objekte ein.
6. Verschalten Sie die dynamischen Attribute der Bildelemente mit den Parametern des AS.
7. Bauen Sie die "Bedienberechtigungsketten" mit den Objekten "@Level5" und "@Level6" auf.
8. Speichern Sie das Bild.

2.2.1.5 Dynamisierung von Bildbausteinen

Es gibt unterschiedliche Wege, um eine Dynamisierung zu erreichen:

- Die Extension der benötigten Variablen ist bekannt.
Für das zu dynamisierende Objekt, z. B. Balken, rufen Sie die Objekteigenschaften auf. In der Spalte "Dynamik" des Eigenschaften-Fensters doppelklicken Sie das Symbol der Glühlampe für das gewünschte Attribut. Im Eingabefeld können Sie nun die Extension mit Punkt eintragen, z. B. ".PV_IN".
- Variable aus der Variablenliste
Der bei Prozessbildern übliche Weg führt über die Variablenliste. Hier sind allerdings alle Variablen aufgeführt. Mit der rechten Maustaste klicken Sie im Eigenschaften-Fenster auf das Glühlampen-Symbol und wählen Sie "Variable" aus.
Suchen Sie die gewünschte Variable und doppelklicken Sie darauf. Da in der Dynamisierung der vollständige Variablenname steht, löschen Sie den Text bis zum Punkt (".") vor der Extension.

2.2.1.6 So erstellen Sie eine Loop-Darstellung

Die Funktion "Bildanwahl über Messstelle" bietet Ihnen die Möglichkeit in die Loop-Darstellung zu springen. Daher empfiehlt es sich, immer eine Loop-Darstellung zu erzeugen, auch wenn nur eine Sicht am Bildbaustein benötigt wird.

Vorgehen

1. Generieren Sie mit dem Faceplate Designer die Bilder ohne die Loop-Darstellung.
Die Gruppenbild-Darstellung wird ohne die Loop-Bild-Anwahltaste generiert.
2. Generieren Sie nochmals mit Loop-Bild-Darstellung. Klicken Sie dabei bei allen Bildern auf die Schaltfläche "Nein".

2.2.1.7 So generieren Sie nachträglich eine neue Sicht

Vorgehen

Wenn Sie nachträglich eine neue Sicht generieren wollen und die bereits vorhandenen Sichten schon modifiziert haben, empfehlen wir die folgende Vorgehensweise:

1. Tragen Sie im Faceplate Designer den Typnamen und die gewählten Sichten genau so wieder ein wie in der ersten Generierung.
2. Tragen Sie zusätzlich die neue Sicht ein und starten Sie die Generierung.
Während der Generierung erhalten Sie die Abfrage für jede bereits vorhandene Datei, ob diese überschrieben werden soll.
3. Quittieren Sie diese Meldungen mit "Nein".

2.3 Basiselemente

2.3.1 Ablageort der Basiselemente

Die Basiselemente sind alle in dem Musterbild "@PCS7Elements.pdl" abgelegt. Das Bild wird im Verzeichnis "..\Wincc\options\pd\l\FaceplateDesigner_V6" abgelegt und vom OS-Projekteditor in das Projekt kopiert.

2.3.2 Analogwertdarstellung und Analogwertbedienung

Eigenschaften



Objekttyp	Objektname	Bildname
PCS7_AnalogValue	PCS7_AnalogValue1 PCS7_AnalogValue2 PCS7_AnalogValue3	@PCS7Elements.pdl

Die beiden Objekte "PCS7_AnalogValue1" und "PCS7_AnalogValue2" sind identisch und hier z. B. folgendermaßen parametrieren:

- "PCS7_AnalogValue1" für Analogwertbedienung
- "PCS7_AnalogValue2" für Analogwertanzeige

Bei diesen beiden Objekten wird die Gleitkommadarstellung des EA-Feldes verwendet. Die Objekte können Sie bei Zahlendarstellungen und Zahleneingaben verwenden, die nicht vom Prozess verändert werden, da hier das gleitende Komma nicht störend wirkt.

Wir empfehlen das Objekt "PCS7_AnalogValue3", das mit dem neuen AdvancedAnalogDisplay erstellt wurde. Hier gleiten nur die Vorkommastellen, während die Nachkommastellen instanzspezifisch projektierbar sind.

Bei Verwendung als reine Analogwertanzeige nehmen Sie folgende Einstellungen vor:

- Parametrieren Sie die Eigenschaft "Bedienfreigabe" auf "FALSE".
- Setzen Sie die Eigenschaft "Hintergrundfarbe" auf "grau".

Bei Verwendung als Analogwertbedienung wird bei einem Mausklick das Skript "PCS7_OpenInputBoxAnalog_V6" aufgerufen.

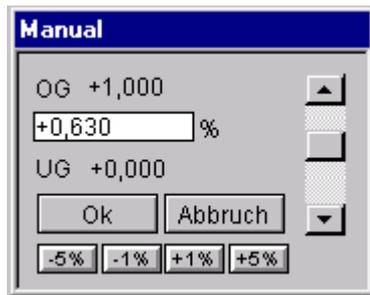
Bedienbild "@PCS7_BedAnalog.pdf"

Das Bedienbild "@PCS7_BedAnalog.pdf" wird bei folgendem Skript-Übergabeparameter aufgerufen:

CallFrom = 0

```
"PCS7_OpenInputBoxAnalog_V6(lpszPictureName,lpszObjectName,0);"
```

Dieses Bedienbild enthält einen Schieberegler mit prozentualer Verstellung und ist für Analogbedienungen mit Bediengrenzen vorgesehen.



Bedienbild "@PCS7_BedAnalog_NL.pdf"

Das Bedienbild "@PCS7_BedAnalog_NL.pdf" wird bei folgendem Skript-Übergabeparameter aufgerufen:

CallFrom = 1

```
"PCS7_OpenInputBoxAnalog_V6(lpszPictureName,lpszObjectName,1);"
```

Dieses Bedienbild enthält keinen Schieberegler und ist für Analogbedienungen ohne Bediengrenzen vorgesehen.



Um die Änderung einer prozessgesteuerten Bedienfreigabe direkt bei der Änderung in das Bedienbild zu übertragen, können Sie auch in der Eigenschaft "Bedienfreigabe" des Objektes "PCS7_AnalogValue" folgende Skripte aufrufen:

- "PCS7_OpenInputBoxAnalog_V6(lpszPictureName,lpszObjectName,10);"
für das Bedienbild mit Grenzen
- "PCS7_OpenInputBoxAnalog_V6(lpszPictureName,lpszObjectName,11);"
für das Bedienbild ohne Grenzen

Wenn Sie das Bedienbild angewählt haben und die Bedienfreigabe zum gleichen Zeitpunkt für diesen Wert entzogen wird, wird auch der Wert im Bedienbild unbedienbar.

Eigenschaften	Funktion
ProcessValue/OutputValue	Strukturelement des Analogwerts, der bedient wird
ProcessValue/VisibleValue	Strukturelement des Analogwerts, der dynamisiert wird
Limits/UpperLimit	Strukturelement für Bediengrenze "Oben" des Analogwerts
Limits/LowerLimit	Strukturelement für Bediengrenze "Unten" des Analogwerts
Sonstige/Bedienfreigabe	Bedienfreigabe von der AS-Seite, z. B. "Q_SP_OP" oder fest auf "Nein" parametrisiert, wenn nur eine Anzeige erfolgen soll
Sonstige/Anzeige_Analogwert	muss immer sichtbar geschaltet sein
Sonstige/Passwort	nicht benutzen, wird über die Bedienberechtigung abgewickelt
Sonstige/Anzeige_Linie	Anzeige der blauen Linie
Sonstige/Linienstärke	Linienstärke kann eingestellt werden.
Sonstige/Open_BedBox_from_Bar	Eigenschaften zu indirekten Aufrufen des Bedienbildes "@PCS7_BedAnalog.pdl", z. B. über Balken
Schrift/Text	Beschriftung des EA-Feldes (Sollwert)
Schrift/Unit	Einheit des Analogwerts dynamisch über .MEMBER#unit, z. B. .SP_OP#unit
Schrift/X-Ausrichtung_Text	linksbündig, wenn mehrere EA-Felder untereinander stehen rechtsbündig für einzeln stehende Analogwerte
Farben/Schriftfarbe	Auswahl der Schriftfarbe
Farben/Linienfarbe	Auswahl der Linienfarbe
Farben/Hintergrundfarbe_Value	Auswahl der Hintergrundfarbe "Value" wird mit Skript dynamisiert beim Ereignis "Änderung der Bedienfreigabe".
Farben/Schriftfarbe_Value	Auswahl Schriftfarbe "Value"
Geometrie/Breite_Analogwert	Breite des gesamten Analogfeldes
Geometrie/Breite_Linie_Text	Länge der Linie und des Textes kann eingestellt werden.
Geometrie/PositionX1_Linie_Text	muss bei Änderung der Text- und Linienlänge angepasst werden
Geometrie/PositionY1_Linie	Linienhöhe gegenüber der Schrift

Die Eigenschaft "Schrift/Text" ist mit dem Beispiel-Parameter ".SP_OP#shortcut" per Default eingestellt. Der Anzeigetext des Bausteinsymbols und die Überschrift des Analog-Bedienbildes werden aus dem AS gelesen (s7_shortcut). Bei Variablen, die keinen Anzeigetext besitzen, z. B. PV_IN, müssen Sie die Verschaltung löschen und den Text direkt in WinCC parametrieren, bei Bedarf dreisprachig.

Bei Änderung der Eigenschaft "Open_BedBox_from_Bar" wird ebenfalls das Skript "PCS7_OpenInputBoxAnalog_V6" aufgerufen. Hier wird ebenfalls zuvor überprüft, ob die Bedienfreigabe vorhanden ist.

Das Skript kann auch für die indirekte Bedienung verwendet werden. Z. B. wird beim Ereignis "Anklicken des Sollwert-Balkens" das entsprechende Bedienbild aufgerufen. (Siehe hierzu auch die Beschreibung bei der zweifachen Balkendarstellung für Sollwert und Istwert.) Dort wird die Änderung einer Eigenschaft über die Direktverbindung auf die Eigenschaft "Open_BedBox_from_Bar" umgeleitet.

2.3.3 Analogwertdarstellung mit "AdvancedAnalogDisplay"

Eigenschaften



Objekttyp	Objektname	Bildname
PCS7_AnalogValue	PCS7_AnalogValue3	@PCS7Elements.pdl

Für die Zahlendarstellungen, bei denen keine Gleitkommata, vor allem nicht für die Nachkommastellen, erwünscht sind, gibt es die Zahlendarstellung mit dem AdvancedAnalogDisplay. Hier kann das Zahlenformat instanzspezifisch über das Bausteinsymbol vorgegeben werden. Siehe Kapitel 2.1.7 "Zahlenformate projektieren".

Die Beschreibung der Objekte des Anwenderobjekts und die Funktion der Eigenschaften ist identisch mit der Beschreibung im Kapitel 2.3.2. Zusätzlich ist die Eigenschaft "Format" vorhanden. Das Zahlenformat wird über die Direktverbindung angesteuert, wenn instanzspezifisch Zahlenformate erwünscht sind.

2.3.4 Statischer Text

Eigenschaften

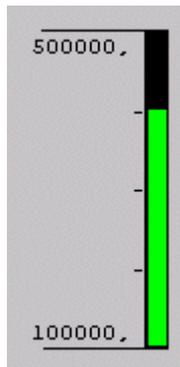
Der statische Text wird z. B. für die Beschriftungen von Optionskästchen und Ein-/Aus-Tasten verwendet.

Format: Zeichensatz Arial, Schriftgröße 12

Objektyp	Objektname	Bildname
PCS7_StaticText	StaticText	@PCS7Elements.pdl

2.3.5 Einfache Balkendarstellung für Analogwerte

Eigenschaften



Objektyp	Objektname	Bildname
PCS7_BarStandard_1	BarStandard_1	@PCS7Elements.pdl

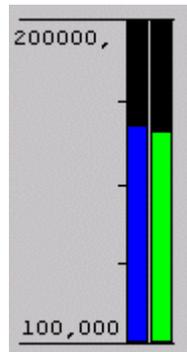
Eigenschaften	Funktion
Links/PV	Strukturelement des Analogwerts, der als Balken dynamisiert wird
Links/Range_LL	Strukturelement für "Balkengrenze unten" (Messanfangswert)
Links/Range_UL	Strukturelement für "Balkengrenze oben" (Messendwert)
Sonstige/PVunderLimit	Anzeige Messbereichsunterschreitung Der Istwert darf vom Anwender nicht verändert werden.
Sonstige/PVoverLimit	Anzeige Messbereichsüberschreitung Der Istwert darf vom Anwender nicht verändert werden.
Farben/Balkenfarbe_PV	Farbe des Balkens

An den verschalteten Eigenschaften PV, RANGE_LL, RANGE_UL sind jeweils Skripte hinterlegt (Aufruf des Skripts "PCS7_UpdateBar_V6.fct"), die bei Änderung der Werte eine evtl. Messbereichsüberschreitung mit Pfeilen an den Balkengrenzen anzeigen.

2.3.6 Zweifache Balkendarstellung für Analogwerte

Eigenschaften

Die zweifache Balkendarstellung wird z. B. für die gleichzeitige Darstellung von Sollwert und Istwert verwendet



Objekttyp	Objektname	Bildname
PCS7_BarStandard_2	BarStandard_2	@PCS7Elements.pdl

Eigenschaften	Funktion
Links/PV	Strukturelement des Istwerts, der als Balken dynamisiert wird
Links/SP	Strukturelement des Sollwerts, der als Balken dynamisiert wird
Links/Range_LL	Strukturelement für "Balkengrenze unten" (Messanfangswert)
Links/Range_UL	Strukturelement für "Balkengrenze oben" (Messendwert)
Sonstige/SPunderLimit	Anzeige Messbereichsunterschreitung Der Sollwert darf vom Anwender nicht verändert werden.
Sonstige/PVunderLimit	Anzeige Messbereichsunterschreitung Der Istwert darf vom Anwender nicht verändert werden.
Sonstige/SpoverLimit	Anzeige Messbereichsüberschreitung Der Sollwert darf vom Anwender nicht verändert werden.
Sonstige/PVoverLimit	Anzeige Messbereichsüberschreitung Der Istwert darf vom Anwender nicht verändert werden.
Farben/Balkenfarbe_SP	Farbe des Balkens für den Sollwert
Farben/Balkenfarbe_PV	Farbe des Balkens für den Istwert

Wenn Sie die Bedienfreigabe der Balkendarstellung auf "TRUE" setzen, wird beim Anklicken der Balkendarstellung die Bedienfreigabe über das Skript kurz auf "FALSE" und anschließend sofort wieder auf "TRUE" gesetzt.

Die Eigenschaft "Bedienfreigabe" kann anschließend über die Direktverbindung auf das zugehörige Basiselement "PCS7_AnalogValue", Eigenschaft "Open_BedBox_from_Bar", weitergeleitet werden und bewirkt dort den Aufruf des analogen Bedienbildes zum Sollwert (siehe auch Beschreibung der Analogwertbedienung, Kapitel 2.3.2).

Bei der Balkendarstellung wird die Pfeil-Darstellung für Werte unter- und oberhalb der Balkengrenzen ergänzt.

An den verschalteten Eigenschaften PV, SP, RANGE_LL, RANGE_UL sind jeweils Skripte hinterlegt (Aufruf des Skripts "PCS7_UpdateBar.fct"), die bei Änderung der Werte eine evtl. Messbereichsüberschreitung mit Pfeilen an den Balkengrenzen anzeigen.

2.3.7 Balkendarstellung horizontal

Eigenschaften

Die horizontale Darstellung des Balkens kann z. B. für Stellgrößen verwendet werden.



Objekttyp	Objektname	Bildname
PCS7_BarStandard_3	BarStandard_3	@PCS7Elements.pdl

Die Eigenschaft "Bedienfreigabe" kann über die Direktverbindung auf das zugehörige Basiselement "PCS7_AnalogValue", Eigenschaft "Open_BedBox_from_Bar", weitergeleitet werden und bewirkt dort den Aufruf des analogen Bedienbildes zum Sollwert (siehe auch Beschreibung der Analogwertbedienung, Kapitel 2.3.2).

Bei der Balkendarstellung wird die Pfeil-Darstellung für Werte unter- und oberhalb der Balkengrenzen ergänzt.

Eigenschaften	Funktion
Sonstige/Balkenfarbe	Farbe des Balkens
Sonstige/PVunderLimit	Anzeige der Messbereichsunterschreitung Der Istwert darf vom Anwender nicht verändert werden.
Sonstige/PVoverLimit	Anzeige der Messbereichsüberschreitung Der Istwert darf vom Anwender nicht verändert werden.
Sonstige/Unit	Einheit der Stellgröße
Links/RANGE_LL	Mess-Anfang der Balken-Anzeige
Links/50PWERT	50%-Anzeige wird bei Änderung von RANGE_LL oder RANGE_UL mit Skript errechnet.
Links/RANGE_UL	Mess-Ende der Balken-Anzeige
Links/PV	Strukturelement der Stellgröße, die als Balken dynamisiert wird

2.3.8 Balkendarstellung "Grenzwertanzeige"

Eigenschaften



Objektyp	Objektname	Bildname
PCS7_BarLimits	BarLimits	@PCS7Elements.pdl

Eigenschaften	Balken-Eigenschaft	Funktion
Grenzen/RANGE_UL	Maximalwert und Prozessanschluss	Messbereichsende des Balkens (.MO_PVHR)
Grenzen/RANGE_LL	Minimalwert, Nullpunktwert und Obergrenze RH5	Messbereichsanfang des Balkens (.MO_PVLR)
Grenzen/VALUE_AH	Obergrenze AH	Grenzanzeige "Alarm high" (.PVH_ALM)
Grenzen/VALUE_WH	Obergrenze WH	Grenzanzeige "Warning high" (.PVH_WRN)
Grenzen/VALUE_WL	Untergrenze WL	Grenzanzeige "Warning low" (.PVL_WRN)
Grenzen/VALUE_AL	Untergrenze AL	Grenzanzeige "Alarm low" (.PVL_ALM)
Grenzen/@ObergrenzeTH	Obergrenze TH	Steuerung der unteren Grenze des zweituntersten Segments Die Eigenschaft ist nur im Konfigurationsdialog sichtbar.
Grenzen/@ObergrenzeRH4	Obergrenze RH4	Steuerung der unteren Grenze des untersten Segments Die Eigenschaft ist nur im Konfigurationsdialog sichtbar.
Grenzen/@BalkenfarbeRH4	Balkenfarbe RH4	Farbbestimmung des zweituntersten Segments Die Eigenschaft ist nur im Konfigurationsdialog sichtbar.
Grenzen/@BalkenfarbeRH5	Balkenfarbe RH5	Farbbestimmung des untersten Segments Die Eigenschaft ist nur im Konfigurationsdialog sichtbar.

Eigenschaften	Balken-Eigenschaft	Funktion
Farben/FarbeAlarm	Balkenfarbe AH	
Farben/FarbeWarnung	Balkenfarbe WH	
Farben/FarbeHintergrund	Balkenfarbe TH	Farbe des mittleren Segments

Bei einer Änderung der unteren Grenzwerte "VALUE_WL", "VALUE_AL" und "RANGE_LL" wird das Skript "PCS7_UpdateBarLimits_V6" aufgerufen, da hier die Balkendarstellung für "Alarm" und "Warnung unten" nicht mit Mitteln des Standard-Balkens erzeugt werden kann. Für die oberen Grenzwerte ist kein Skript notwendig.

2.3.9 Anzeige "Meldungsunterdrückung"

Eigenschaften



Objekttyp	Objektname	Bildname
PCS7_MSG_LOCK	MSG_LOCK	@PCS7Elements.pdl

Eigenschaften	Funktion
Benutzerdefiniert2/ aktueller Zustand	Steuerung "Zustandsanzeige 2" Einmal durchgestrichene Meldungen sperren Anforderung vom AS (.MSG_LOCK) 
Benutzerdefiniert2/ aktueller Zustand1	Steuerung "Zustandsanzeige 3" Zweimal durchgestrichene Meldungen sind gesperrt. Rückmeldung von WinCC (.QMSG_SUP) 
Benutzerdefiniert2/ Anzeige1	"Zustandsanzeige 3" wird eingeblendet und überblendet "Zustandsanzeige 2" (.QMSG_LOCK)

2.3.10 Anzeige "Batch Occupied"

Eigenschaften



Objektyp	Objektname	Bildname
PCS7_OCCUPIED	OCCUPIED	@PCS7Elements.pdl

Eigenschaften	Funktion
Benutzerdefiniert2/ aktueller Zustand	Steuerung der Zustandsanzeige 1 / Anzeige, ob Baustein von Batch belegt ist. Strukturelement (.OCCUPIED)

2.3.11 Quittierung der Meldungen vom angewählten Bildbaustein

Eigenschaften

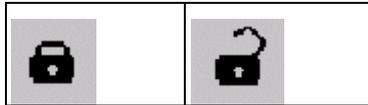


Objektyp	Objektname	Bildname
Button	ButtonQS	@PCS7Elements.pdl

Die Taste für die Sammelquittierung von Meldungen entspricht der Taste im Standard-Tastensatz "@Buttons11.pdl", jedoch mit einem anderem Skript für die instanzspezifische Meldequittierung. Die Sammelquittierung funktioniert nur im Zusammenhang mit einer Sammelanzeige.

2.3.12 Anzeige-Baustein "Verriegelt" (Ventil, Motor)

Eigenschaften



Objekttyp	Objektname	Bildname
Zustandsanzeige	Lock	@PCS7Elements.pdl

Eigenschaften	Funktion
Zustand/aktueller Zustand	Anzeige, ob Baustein verriegelt ist Strukturelement (.LOCK)

2.3.13 Sammelanzeige

Eigenschaften

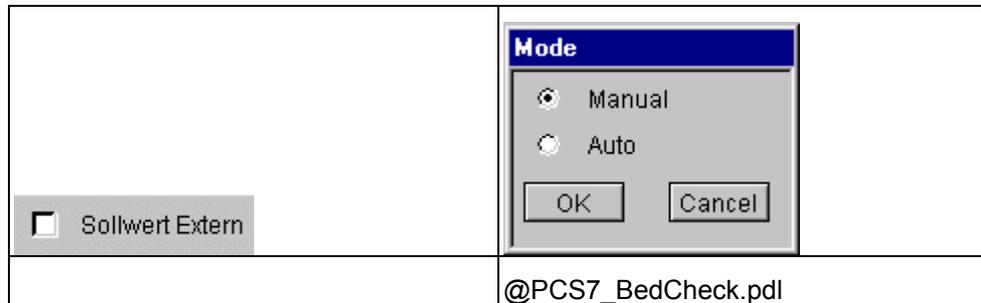


Objekttyp	Objektname	Bildname
Sammelanzeige	EventState	@PCS7Elements.pdl

Eigenschaften	Funktion
Sonstige/Sammelwert	Anzeige der Alarm- und Warnungszustände Strukturelement (.EventState)

2.3.14 Binärwertbedienung mit "Check Box_R"

Eigenschaften



Objektyp	Objektname	Bildname
CHECKBOX_R	Checkbox_R1	@PCS7Elements.pdl

Das Bild "@PCS7_BedCheck.pdl" wird über das Skript "PCS7_OpenCheckbox_V6" aufgerufen.

Eigenschaften	Funktion
Sonstige/DisplayActivWith	DisplayActivWith = 1 → Bei gesetztem Häkchen im Optionskästchen wird 'TRUE' auf die binäre Variable geschrieben. DisplayActivWith = 0 → Bei gesetztem Häkchen im Optionskästchen wird 'FALSE' auf die binäre Variable geschrieben.
Sonstige/Bedienfreigabe	Bedienfreigabe des Optionskästchens
Verschaltung/Input	Überschrift für Bedienbild zum Optionskästchen "@PCS7_BedCheck.pdl"
Verschaltung/Variable	Verschaltung auf das binäre Strukturelement, z. B. (.MSG_LOCK)
Verschaltung/NegatedVariable	Optionskästchen, das angeklickt wird; darf vom Anwender nicht verändert werden.
Verschaltung/Read_Text_From_AS	Text wird von "String_0" oder "String_1" gelesen und der Eigenschaft "Input" zugewiesen.
Verschaltung/CaptionCheckBoxOn	JA = Text von "Input" wird am Optionskästchen rechts angezeigt. NEIN = Bei wenig Platz wird kein Text zum Optionskästchen angezeigt.

Wenn Sie auf das Optionskästchen klicken, wird die interne Option "Inverted Variable" angewählt. Der Zustand der zu bedienenden binären Variable wird ermittelt und auf "Inverted Variable" zurück geschrieben (Eigenschaft "Verschaltung/NegatedVariable"). Weiterhin wird bei Änderung der Eigenschaft "Verschaltung/Variable" und bei Bildanwahl ein Skript durchlaufen, das die Anzeige von "NegatedVariable" aktualisiert (siehe auch Flussdiagramm Check_Box). Die Eigenschaft "Verschaltung/Variable" muss von der Projektierung auf "ungleich 1" und "ungleich 0" (0x8) eingestellt sein, damit in allen Fällen das Skript auch bei Bildanwahl durchlaufen wird.

Die Texte für die Beschriftung der Optionsfelder werden aus den Parameterattributen "String_0" und "String_1" der Bausteininstanz gelesen.

Beschriftung des Optionskästchens

Für die Beschriftung des Optionskästchens gilt Folgendes:

- Wenn die Eigenschaft "Read_Text_From_AS" auf 1 gesetzt wird, wird bei "DisplayActivWith = 1" der komplette Text von "String_1" rechts vom Optionskästchen angezeigt.
- Bei "DisplayActivWith = 0" wird der komplette Text von "String_0" rechts vom Optionskästchen angezeigt.
- Wenn die Eigenschaft "Read_Text_From_AS" auf 0 gesetzt wird, wird der an "Input" projizierte Text angezeigt. Bei der Anzeige von Ausgangsparametern müssen Sie dies einstellen, da hier kein "String_0" oder "String_1" existiert und somit Skriptfehler erzeugt würden. Außerdem dürfen Sie das Optionskästchen für diesen Fall nicht bedienbar schalten.

Beschriftung der Optionsfelder

Für die Beschriftung der Optionsfelder gilt Folgendes:

- Wenn der Text der Strings "String_0" und "String_1" ein "=" enthält, wird der rechte Teil der Strings (nach dem "=") für die Beschriftung der Optionsfelder verwendet.
- Wenn der Text der Strings kein "=" enthält, wird der komplette Text für die Beschriftung der Optionsfelder verwendet.

Überschrift des Bedienbildes

Für die Überschrift des Bedienbildes gilt Folgendes:

- Wenn der Text der Strings "String_0" und "String_1" ein "=" enthält, wird der linke Teil vor dem "=" für die Überschrift des Bedienbildes verwendet, bei "DisplayActivWith = 1" von "String_1" und bei "DisplayActivWith = 0" von "String_0".
- Wenn der Text der Strings kein "=" enthält, wird der Text von der Beschriftung des Optionskästchens genommen.

2.3.15 Binärwertbedienung mit "Check Box_L"

Eigenschaften

Suppress HH=No

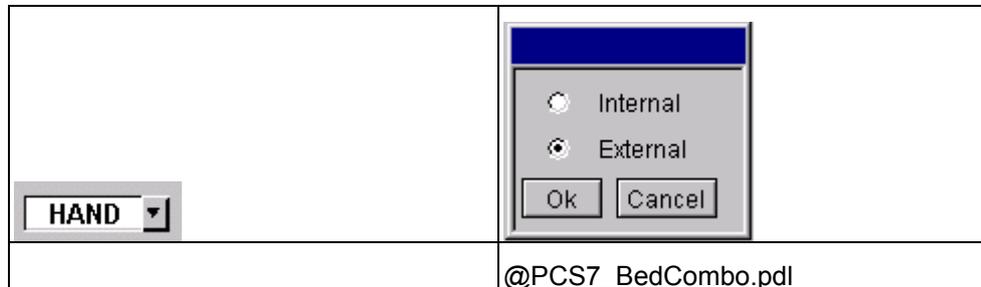
Die Funktionalität entspricht der Binärwertbedienung mit "CheckBox_R", nur wird der Anzeigetext auf der linken Seite dargestellt.

Auch wenn der Text nicht angezeigt wird, werden beide Varianten (Text rechts oder Text links) benötigt, abhängig davon, ob rechts oder links vom Optionskästchen ein weiteres bedienbares Element liegt, z. B. bei den Alarmgrenzen von MEAS_MON.

Eigenschaften und Funktion siehe 2.3.14.

2.3.16 Binärwertbedienung mit Kombinationsfeld (Combobox)

Eigenschaften



Objekttyp	Objektname	Bildname
PCS7_COMBOBOX	COMBOBOX1	@PCS7Elements.pdl

Das Bild "@PCS7_BedCombo.pdl" wird über das Skript "PCS7_OpenComboBox_V6.fct" aufgerufen.

Die Texte werden von "String_0" und "String_1" gelesen

Eigenschaften	Funktion
Farben/BackColor_Text1	Hintergrundfarbe für Text1
Farben/BackColor_Text2	Hintergrundfarbe für Text2
Schrift/Text1	Anzeigetext für Text1 (wird aus AS gelesen)
Schrift/Text2	Anzeigetext für Text2 (wird aus AS gelesen)
Displays/Display_Text1	Anzeige-Text1 (Steuerung der Anzeige, welcher Wert markiert ist)
Displays/Display_Text2	Anzeige-Text2 (Steuerung der Anzeige, welcher Wert markiert ist)
Sonstige/ OP_enabled_Text1	Bedienfreigabe für Text1
Sonstige/ OP_enabled_Text2	Bedienfreigabe für Text2
Sonstige/ Bedienfreigabe	Bedienfreigabe für komplettes Kombinationsfeld für Steuerung über "@Level5/6"
Sonstige/ Password_Text1	Bedienberechtigung für Text1 (wird nicht verwendet)
Sonstige/ Password_Text2	Bedienberechtigung für Text2 (wird nicht verwendet)
Links/Write_Variable1	Strukturelement, das mit Anwahl-Text1 beschrieben wird
Links/Write_Variable2	Strukturelement, das mit Anwahl-Text2 beschrieben wird
Links/Display_Variable1	Strukturelement, das den ersten Binärzustand (Text1) anzeigt.
Links/Display_Variable2	Strukturelement, das den zweiten Binärzustand (Text2) anzeigt

Eigenschaften	Funktion
Parameters/ Write_value_Text1	Gibt an, welcher Wert auf das Strukturelement "Write_Variable1" mit Anwahl-Text1 geschrieben wird ('TRUE' oder 'FALSE')
Parameters/Display_Text1_with	Gibt an, mit welchem Wert ('TRUE' oder 'FALSE') das Strukturelement "Display_Variable1" den Text1 anzeigt.
Parameters/ Write_value_Text2	Gibt an, welcher Wert auf das Strukturelement, "Write_Variable2" mit Anwahl-Text2 geschrieben wird ('TRUE' oder 'FALSE').
Parameters/Display_Text2_with	Gibt an, mit welchem Wert ('TRUE' oder 'FALSE') die "Display_Variable2" den Text2 anzeigt.

Die Texte für die Beschriftung der Optionsfelder werden aus den Parameterattributen "String_0" und "String_1" der Bausteininstanz wie folgt gelesen:

- Wenn der Text der Strings ein "=" enthält, wird der Teil nach dem "=" für die Beschriftung der Optionsfelder und der Teil vor dem "=" für die Überschrift des Bedienbildes verwendet.
- Wenn der Text der Strings kein "=" enthält, wird der komplette Text für die Beschriftung der Optionsfelder verwendet. Die Überschrift des Bedienbildes wird in diesem Fall mit Leerzeichen gefüllt.

Hinweise für die Bedienfreigabe

Für das Anzeigen und Hervorheben einer fehlenden Bedienfreigabe kann der Hintergrund des Textes im Kombinationsfeld grau geschaltet werden. Dies geschieht über eine Direktverbindung von "@Level5"- oder "@Level6"-Hintergrundfarbe auf "BackColor_Text1" in diesem Objekt und eine weitere Direktverbindung von "BackColor_Text1" auf "BackColor_Text2"-Hintergrundfarbe. Die Default-Einstellungen für die Hintergrundfarben müssen dann beide "grau" sein.

Diese Möglichkeit besteht jedoch nur, wenn die verschiedenen Schaltzustände des Kombinationsfeldes immer mit weißem Hintergrund angezeigt werden, damit die Bedienberechtigung zwischen grau und weiß schaltet.

Beispiel für Regler-Umschaltung "Hand/Automatik":

Beim Kombinationsfeld zur Betriebsartenumschaltung "Hand/Automatik" wird die Hintergrundfarbe zur Kennzeichnung der Betriebsart folgendermaßen verwendet:

- Weiß = Hand
- Grün = Automatik

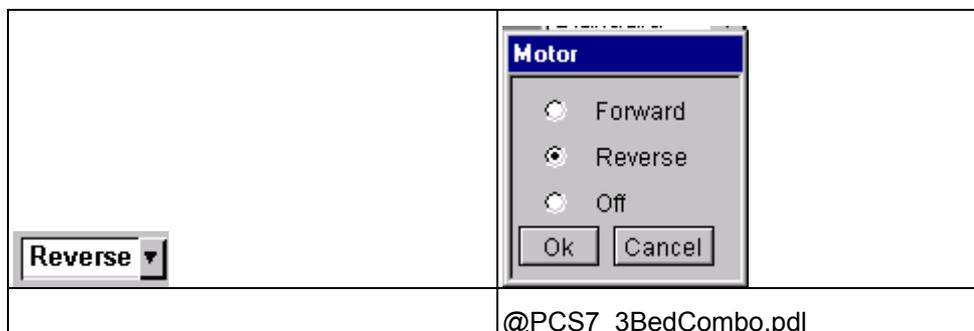
Aus diesem Grund kann hier bei fehlender Bedienberechtigung nicht auf den grauen Hintergrund geschaltet werden. Bei fehlender Bedienfreigabe wird deshalb bei den Kombinationsfeldern zusätzlich noch die Pfeil-Schaltfläche unsichtbar geschaltet.

Bedienfreigabe "Ja": 

Bedienfreigabe "Nein": 

2.3.17 Binärwertbedienung mit Kombinationsfeld (3ComboBox)

Eigenschaften



Objektyp	Objektname	Bildname
PCS7_3COMBOBOX	3COMBOBOX1	@PCS7Elements.pdl

Das Bild "@PCS7_3BedCombo.pdl" wird über das Skript "PCS7_Open3ComboBox_V6.fct" aufgerufen.

Die Texte werden von "String_0" und "String_1" gelesen.

Eigenschaften	Funktion
Farben/BackColor_Text1	Hintergrundfarbe für Text1
Farben/BackColor_Text2	Hintergrundfarbe für Text2
Farben/BackColor_Text3	Hintergrundfarbe für Text3
Schrift/Text1	Anzeigetext für Text1 (wird aus AS gelesen)
Schrift/Text2	Anzeigetext für Text2 (wird aus AS gelesen)
Schrift/Text3	Anzeigetext für Text3 (wird aus AS gelesen)
Displays/Display_Text1	Anzeige Text1 (Steuerung der Anzeige, welcher Wert markiert ist)
Displays/Display_Text2	Anzeige Text2 (Steuerung der Anzeige, welcher Wert markiert ist)
Displays/Display_Text3	Anzeige Text3 (Steuerung der Anzeige, welcher Wert markiert ist)
Sonstige/OP_enabled_Text1	Bedienfreigabe für Text1
Sonstige/OP_enabled_Text2	Bedienfreigabe für Text2
Sonstige/OP_enabled_Text3	Bedienfreigabe für Text3
Sonstige/Bedienfreigabe	Bedienfreigabe für komplettes Kombinationsfeld für Steuerung über "@Level5/6"
Sonstige/Password_Text1	Bedienberechtigung für Text1 (wird nicht verwendet)
Sonstige/Password_Text2	Bedienberechtigung für Text2 (wird nicht verwendet)
Sonstige/Password_Text3	Bedienberechtigung für Text3 (wird nicht verwendet)
Links/Write_Variable1	Strukturelement, das mit Anwahl-Text1 beschrieben wird

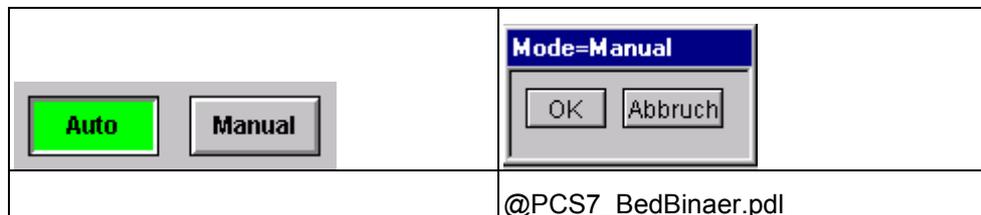
Eigenschaften	Funktion
Links/Write_Variable2	Strukturelement, das mit Anwahl-Text2 beschrieben wird
Links/Write_Variable3	Strukturelement, das mit Anwahl-Text3 beschrieben wird
Links/Display_Variable1	Strukturelement, das den ersten Binärzustand (Text1) anzeigt
Links/Display_Variable2	Strukturelement, das den zweiten Binärzustand (Text2) anzeigt
Links/Display_Variable3	Strukturelement, das den zweiten Binärzustand (Text3) anzeigt
Parameters/Write_value_Text1	Gibt an, welcher Wert auf das Strukturelement "Write_Variable1" mit Anwahl-Text1 geschrieben wird ('TRUE' oder 'FALSE').
Parameters/Display_Text1_with	Gibt an, mit welchem Wert die "Display_Variable" den Text1 anzeigt.
Parameters/Write_value_Text2	Gibt an, welcher Wert auf das Strukturelement "Write_Variable2" mit Anwahl-Text2 geschrieben wird ('TRUE' oder 'FALSE').
Parameters/Display_Text2_with	Gibt an, mit welchem Wert ('TRUE' oder 'FALSE') die "Display_Variable" den Text2 anzeigt.
Parameters/Write_value_Text3	Gibt an, welcher Wert auf das Strukturelement "Write_Variable3" mit Anwahl-Text3 geschrieben wird ('TRUE' oder 'FALSE').
Parameters/Display_Text3_with	Gibt an, mit welchem Wert ('TRUE' oder 'FALSE') die "Display_Variable" den Text3 anzeigt.

Die Texte für die Beschriftung der Optionsfelder werden aus den Parameterattributen "String_0" und "String_1" der Bausteininstanz wie folgt gelesen:

- Wenn der Text der Strings ein "=" enthält, wird der Teil der Strings nach dem "=" für die Beschriftung der Optionsfelder und der Teil vor dem "=" für die Überschrift des Bedienbildes verwendet.
- Wenn der Text der Strings kein "=" enthält, wird der komplette Text für die Beschriftung der Optionsfelder verwendet. Die Überschrift des Bedienbildes wird in diesem Fall mit Leerzeichen gefüllt.

2.3.18 Binärwertbedienung mit Taste (Button) und Farbwechsel

Eigenschaften



Objekttyp	Objektname	Bildname
PCS7_BinOp	BinOp0/BinOp1	@PCS7Elements.pdl

Mit "CMD2Steps" kann festgelegt werden, ob die Binärwertbedienung eine Ein- oder Zwei-Tastenbedienung ist.

- CMD2Steps = Ja
Bei Anwahl von "Auto" oder "Manual" wird über das Skript "PCS7_OpenInputBoxBin_V6.fct" das Bild "@PCS7_BedBinaer.pdl" aufgerufen. Mit "OK" in diesem Bild wird dann der über "Write_Value" festgelegte Wert an die an "Write_Variable" verbundene Variable geschrieben.
- CMD2Steps = Nein
Der über "Write_Value" festgelegte Wert wird direkt beim Anklicken von STOP oder LAUF an die mit "Write_Variable" verbundene Variable geschrieben.

Die Texte für die Beschriftung der Tasten werden aus den Parameterattributen "String_0" und "String_1" der Bausteininstanz wie folgt gelesen:

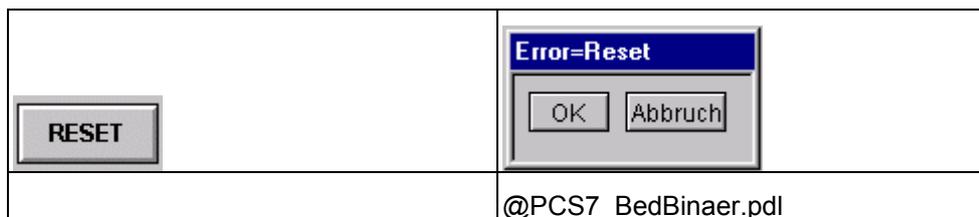
- Wenn der Text der Strings ein "=" enthält, wird der Teil der Strings nach dem "=" für die Beschriftung der Tasten verwendet und der komplette String für die Überschrift des Bedienbildes verwendet.
- Wenn der Text der Strings kein "=" enthält, wird der komplette Text für die Beschriftung der Tasten verwendet.

Eigenschaften	Funktion
Farben/Button_Colour	Hintergrundfarbe für Text, wenn aktiv und nicht bedienbar
Sonstige/Bedienfreigabe	Bedienfreigabe
Sonstige/DisplayActive	Eigenschaften-Anzeige von "Button On" Darf nicht verändert werden; wird über das Skript gesteuert.
Links/Write_Variable	Strukturelement, das mit Anwahl-Text beschrieben wird
Links/Display_Variable	Strukturelement, das den ersten Binärzustand (Text) anzeigt

Eigenschaften	Funktion
Parameters/Write_value	Gibt an, welcher Wert auf das Strukturelement "Write_Variable" mit Anwahl-Text geschrieben wird ('TRUE' oder 'FALSE')
Parameters/Display_is_active_with	Gibt an, mit welchem Wert ('TRUE' oder 'FALSE') die "Display_Variable" den Text mit der ausgewählten Farbe anzeigt (Darstellung aktiv und unbedienbar)
Parameters/ButtonText	Anzeigetext für Text darf nicht verändert werden; wird über das Skript gesteuert
Parameters/CMD2Steps	1- oder 2-Tastenbedienung

2.3.19 Binärwertbedienung mit Taste (Button)

Eigenschaften



Objekttyp	Objektname	Bildname
PCS7_ButtonBit	ButtonBit	@PCS7Elements.pdl

Die Funktion entspricht der Funktion wie in Kapitel 2.3.18 beschrieben mit folgenden Unterschieden:

- kein Farbumschlag
- Taste bleibt von der Darstellung her immer bedienbar (sie hat keine Darstellung "Button On").

Mit CMD2Steps kann festgelegt werden, ob die Binärwertbedienung eine Ein- oder Zwei-Tastenbedienung ist.

- CMD2Steps = Ja
Bei Anwahl von "Reset" wird über das Skript "PCS7_openinputboxbin_V6.fct" das Bild "@PCS7_BedBinär.pdl" aufgerufen. Wenn Sie in diesem Bild auf die Schaltfläche "OK" klicken, wird der über "Write_Value" festgelegte Wert an die mit "Write_Variable" verbundene Variable geschrieben.
- CMD2Steps = Nein
Der über "Write_Value" festgelegte Wert wird direkt beim Anklicken von "Reset" an die mit "Write_Variable" verbundene Variable geschrieben.

2.3.20 Statusanzeige mit 2 Alternativen

Eigenschaften



Objektyp	Objektname	Bildname
PCS7_Status_2_Alternative	Status_2_Alternativen	@PCS7Elements.pdl

Eigenschaften	Funktion
Links/Link	Verschaltung auf das binäre Strukturelement, z. B. (.MSG_LOCK)
Sonstige/Read_Text_From_AS	Beschriftung der Texte für die Statusanzeige aus dem AS ("String_0" und "String_1") lesen.
Sonstige/Anzeige	Darf nicht vom Anwender verändert werden, muss 'TRUE' bleiben
Sonstige/Bedienfreigabe	Bedienfreigabe
Schrift/Text_On	Beschreibung für Zustand 'True'
Schrift/Text_Off	Beschreibung für Zustand 'False'
Farben/BackColor_OFF	Hintergrundfarbe für Zustand 'Off'
Farben/BackColor_ON	Hintergrundfarbe für Zustand 'On'
Farben/TextColor_OFF	Textfarbe für Zustand 'Off'
Farben/TextColor_ON	Textfarbe für Zustand 'On'
Displays/Off	Anzeige des statischen Texts "T_off" Darf nicht vom Anwender verändert werden, wird von Skript gesteuert
Displays/On	Anzeige des statischen Texts "T_on" Darf nicht vom Anwender verändert werden, wird von Skript gesteuert

Wenn die Variable, die auf die Eigenschaft "Links/Link" verschaltet ist, geändert wird, wird das Skript "PCS7_2Stati_Variable_Changed_V6.fct" aufgerufen.

In diesem Skript wird aufgrund des Zustandes von "Links/Link" die Anzeige von "T_off" und "T_on" gesteuert.

Wenn Sie den Link auf einen Eingangsparameter legen, z. B. auf AUT_ON_OP, können die Statustexte aus dem AS gelesen werden (Read_Text_From_AS = TRUE). Bei Ausgangsparametern ist meistens weder "String_0" noch "String_1" vorhanden. In diesem Fall müssen Sie den Text direkt am Objekt (Text_On, Text_Off) hinterlegen (Read_Text_From_AS = FALSE).

2.3.21 Statusanzeige mit n Alternativen

Eigenschaften



Objekttyp	Objektname	Bildname
PCS7_Status_1_v_n	Status_1_v_n	@PCS7Elements.pdl

Eigenschaften	Funktion
Links/Link	Verschaltung auf das binäre Strukturelement (.MSG_LOCK)
Sonstige/Read_Text_From_AS	Beschriftung "Text_On" für die Statusanzeige aus dem AS (String_1) lesen
Sonstige/Anzeige	Darf nicht vom Anwender verändert werden; wird vom Skript gesteuert
Sonstige/Bedienfreigabe	Bedienfreigabe
Schrift/Text_On	Beschreibung für Zustand 'True'
Farben/BackColor_ON	Hintergrundfarbe für Zustand 'On'
Farben/TextColor_ON	Textfarbe für Zustand 'On'
Masks/Mask	Wert, mit dem das Textfeld sichtbar geschaltet wird

Bei der Statusanzeige mit n Alternativen werden jeweils so viele Objekte vom Typ **Status_1_v_n** übereinander gelegt, wie Alternativen benötigt werden.

Bei Änderung der Variablen, die auf die Eigenschaft "Links/Link" verschaltet ist, wird das Skript "PCS7_1vnStati_Variable_Changed_V6.fct" aufgerufen.

Über die Eigenschaft "Masks/Mask" können Sie jeweils für das Objekt einstellen, mit welchem Wert es angezeigt werden soll. Diese Möglichkeit eignet sich z. B. für eine Alternativensteuerung über einen INTEGER oder REAL.

Bei einer Alternativensteuerung über mehrere Binärwerte stellen Sie bei allen Objekten "Mask" auf den Wert 1. Die Priorität bei gleichzeitigem Anstehen von mehreren Binärwerten können Sie über die Eigenschaft "Ebene" einstellen. Bei WinCC gehen die Ebenen von 0 bis 15 und haben ebenfalls aufsteigende Priorität.

Beispiel

Wenn von zwei Objekten eines in Ebene 4 und eines in Ebene 5 liegt und beide übereinander liegen und sichtbar geschaltet sind, wird das Objekt in Ebene 5 über dem Objekt in Ebene 4 angezeigt.

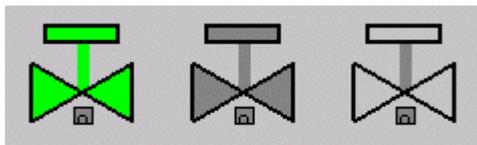
Wenn Sie den Link auf einen Eingangsparameter legen, z. B. auf REV_ON, kann der Statustext aus dem AS gelesen werden (Read_Text_From_AS = TRUE). Bei Ausgangsparametern ist meistens kein "String_1" vorhanden. In diesem Fall müssen Sie den Text direkt am Objekt (Text_On) hinterlegen (Read_Text_From_AS = FALSE).

2.3.22 Zustandsanzeige "Ventil"

Eigenschaften

Die folgende Ventil-Zustandsanzeige wurde aus PCS 7 V5 übernommen.

Für die Realisierung von Zustandsanzeigen ab PCS 7 V6 empfehlen wir die Verwendung der "Erweiterten Zustandsanzeige". Siehe Beschreibung WinCC Leittechnikoptionen.



Objektyp	Objektname	Bildname
PCS7_Valve_Stat	Valve_Stat1, Valve_Stat0, Valve_Stat2	@PCS7Elements.pdl

Eigenschaften	Funktion
Sonstige/Bedienfreigabe	Bedienfreigabe
Sonstige/Anzeige	Anzeige des Anwenderobjekts.
Link/VariableLink	Verschaltung auf das Strukturelement, wenn Steuerung der Zustände über REAL oder INTEGER erfolgen soll
State/Index	Manuelle Einstellung der Zustandsanzeige für binäre Steuerung über "State/Display"
State/Display	Steuerung der Zustände über binäre Variablen

Das Anwenderobjekt "Valve_Stat" kann auf die folgenden zwei Arten genutzt werden:

- Das Objekt wird als normale Zustandsanzeige genutzt. Die Zustände werden über die Eigenschaft "Link/VariableLink" gesteuert. "State/Display" muss auf "Ja" stehen. Es gibt die drei oben aufgeführten Zustände. Diese sind aber nur möglich, wenn die Ventilzustände in einer Variablen zur Verfügung stehen. Dies ist normalerweise nicht der Fall.
- Für jeden der drei Zustände wird ein Objekt "Valve_Stat" angelegt und diese Objekt werden übereinander gelegt. Mit "State/Index" wird der anzuzeigende Zustand eingestellt. Mit "State/Display" wird das entsprechende Strukturelement verbunden. Diese Variante wird beim Bildbaustein VALVE verwendet.

2.3.23 Zustandsanzeige "Motor"

Eigenschaften



Objekttyp	Objektname	Bildname
Zustandsanzeige	Motor	@PCS7Elements.pdl

Normale Zustandsanzeige mit zwei Alternativen und der Steuerung über die Eigenschaft "Zustand/aktuellerZustand".

2.3.24 So projektieren Sie eine Permission

Eigenschaften



Objekttyp	Objektname	Bildname
Permission	Permission_Setpoint	@PCS7Elements.pdl

Das Objekt "Permission" wird zur Erzeugung einer Gesamt-Bedienberechtigung für ein bedienbares Element "Setpoint" verwendet.

Es gibt bedienbare Elemente, die mehreren verschiedenen Berechtigungsprüfungen unterliegen.

Beispiel:

- Zum einen unterliegt "Setpoint" der Berechtigungsprüfung der User-Administration von WinCC, die beim Benutzerwechsel ablaufen muss (Steuerung über "@Level5" oder "@Level6").
- Des Weiteren unterliegt "Setpoint" einer Variablen des AS (Q_SP_OP, Bedienung nur erlaubt, wenn Sollwert intern eingestellt wird). Die Überprüfung der Berechtigung erfolgt bei Änderung der Variablen.

Beim FMCS_PID wird noch ein weiterer Parameter des AS (QPARF) zur Berechtigungsprüfung hinzugezogen.

Vorgehen

Um eine Permission zu projektieren, gehen Sie folgendermaßen vor:

1. Erstellen Sie die Direktverbindung von "@Level5" oder "@Level6", Eigenschaft "Bedienfreigabe" auf Objekt "Permission", Eigenschaft "Level_Source".
2. C-Skript bei Änderung der Eigenschaft "Level_Source":

```
BOOL bTag1 =!GetTagBitWait(".QPARF");
BOOL bTag2 =GetTagBitWait(".Q_SP_OP");
if (bTag1 && bTag2 && value)
{
    SetPropBOOL(IpszPictureName,IpszObjectName,"Target_Bedienfreigabe",TRUE);
    SetPropWord(IpszPictureName,IpszObjectName,"Target_Hintergrundfarbe",CO_WHITE);
}
else
{
    SetPropBOOL(IpszPictureName,IpszObjectName,"Target_Bedienfreigabe",FALSE);

SetPropWord(IpszPictureName,IpszObjectName,"Target_Hintergrundfarbe",CO_LTGRAY)
;
}
SetPropBOOL(IpszPictureName,IpszObjectName,"Level_Target",value);
```

Das Skript wird bei der Bildanwahl aufgerufen oder wenn sich ein neuer Benutzer anmeldet.

Gleichzeitig müssen auch die Variablen des AS überprüft werden, die mit in die Berechtigungsprüfung eingehen. In diesem Fall müssen die Variablen "QPARF" und "Q_SP_OP" geprüft werden.

Wenn alle drei Kriterien erfüllt sind, werden folgende Werte gesetzt:

- Die Eigenschaft "TARGET_BEDIENFREIGABE" wird auf 'TRUE' gesetzt.
- Die Eigenschaft "TARGET_Hintergrundfarbe" wird auf die Farbe "weiß" gesetzt.

Weiterhin wird der Wert der Eigenschaft "Level_Source" auf die Eigenschaft "Level_Target" übertragen.

"Level_Target" ist für das Verschalten der Direktverbindung auf weitere Objekte bestimmt.

3. C-Skript unter der Eigenschaft "Tags":

Dieses Skript wird von den Variablen aufgerufen, die mit in die Berechtigungsprüfung eingehen. Dies sind die Variablen QPARF und Q_SP_OP.

```

BOOL bTag1 =!GetTagBitWait(".QPARF");
BOOL bTag2 =GetTagBitWait(".Q_SP_OP");
** BOOL bLevel = GetPropBOOL(IpszPictureName,"@Level5","Operation");
if (bTag1 && bTag2 && bLevel)
{
  SetPropBOOL(IpszPictureName,IpszObjectName,"Target_Bedienfreigabe",TRUE);
  SetPropWord(IpszPictureName,IpszObjectName,"Target_Hintergrundfarbe",CO_WHITE);
}
else
{
  SetPropBOOL(IpszPictureName,IpszObjectName,"Target_Bedienfreigabe",FALSE);

  SetPropWord(IpszPictureName,IpszObjectName,"Target_Hintergrundfarbe",CO_LTGRAY)
;
}
return TRUE;

```

Bei Änderung einer dieser Variablen werden diese aus dem AS gelesen. Des Weiteren muss der Zustand von "@Level5" und "@Level6" überprüft werden.

Wenn alle Kriterien erfüllt sind, werden folgende Werte gesetzt:

- Die Eigenschaft "Target_Bedienfreigabe" wird auf 'TRUE' gesetzt.
- Die Eigenschaft "TARGET_Hintergrundfarbe" wird auf die Farbe "weiß" gesetzt.

Hinweis: Abhängig davon, von welchem "@Level" aus die Direktverbindung gesteuert wird, muss die entsprechende Direktverbindung im Skript abgefragt werden.

4. Erstellen Sie eine Direktverbindung von der Eigenschaft "Target_Bedienfreigabe" auf das eigentliche Objekt, das der Berechtigungsprüfung unterzogen wird, z. B. "Setpoint_AnalogValue", Eigenschaft "Bedienfreigabe".

An diesem Objekt darf die Bedienfreigabe nicht mehr verschaltet sein und muss den Wert 'FALSE' haben.

Die Hintergrundfarbe muss ebenfalls "grau" sein und wird entweder von der Bedienfreigabe aus über das Skript gesteuert oder von einer Direktverbindung mit der Eigenschaft "TARGET_Hintergrundfarbe" vom Objekt "Permission".

Die Eigenschaft "Passwort" wird für das "Permisson"-Objekt nicht mehr verwendet.

5. Erstellen Sie bei Bedarf eine weitere Direktverbindung von der Eigenschaft "Level_Target" auf ein weiteres "Permission"-Objekt.

2.3.25 Taste "OpenNextFaceplate"

Das Objekt "OpenNextFaceplate" ist dafür vorgesehen, innerhalb eines Plans aus einem geöffneten Bildbaustein einen weiteren Bildbaustein eines AS-Bausteins aufzurufen.

Aufruf des zugehörigen INTERLOK-Bausteins aus einem Motor- oder Ventil-Bildbaustein

Ein typischer Anwendungsfall ist der Aufruf des zugehörigen INTERLOK-Bausteins aus einem Motor- oder Ventil-Bildbaustein heraus.

- In der Eigenschaft "blockname" wird der Bausteinname des aufzurufenden Bausteins eingetragen.
- In der Eigenschaft "Servername" wird der Bausteintyp nach der üblichen Schreibweise eingetragen, z. B. "PCS7 INTERLOK Control" für den INTERLOK-Baustein.
- Wenn Sie auf die Taste "OpenNextFaceplate" klicken, wird von dem Variablennamen des aktuell geöffneten Bildbausteins der Bausteinname abgeschnitten und der unter "blockname" eingetragene Bausteinname angehängt.
- Des Weiteren wird das Strukturelement "#Comment", das in jedem Baustein vorkommt, angehängt und geprüft, ob es im Datenmanager vorhanden ist.

Wenn dies der Fall ist, wird der neu gebildete Variablenname in die Eigenschaft "tagname" geschrieben und der Bildbaustein mit dem Skript "PCS7_OpenGroupDisplay_V6" aufgerufen.

Über ein Skript, welches bei Änderung der Eigenschaft "check_tag" aufgerufen wird, wird ebenfalls geprüft, ob die Variable vorhanden ist. Die Taste wird nur bedienbar geschaltet, wenn die Variable im Datenmanager vorhanden ist.

Damit das Skript in der Eigenschaft "check_tag" auch bei Bildanwahl durchlaufen wird, muss ein gültiges Strukturelement darauf verschaltet sein, dessen Text sich vom Default-Text der Eigenschaft unterscheidet. Somit wird eine Änderung der Eigenschaft erkannt. Hierfür wird wieder das in allen Bausteinen vorhandene "#Comment"-Strukturelement verwendet.

Verwendung mehrerer INTERLOK-Bausteine für einen Motor oder ein Ventil

Ein weiterer häufiger Anwendungsfall ist, dass für einen Motor oder ein Ventil mehrere INTERLOK-Bausteine verwendet werden.

Für diesen Fall kann das gleiche Objekt in den INTERLOK-Baustein eingebaut werden. In der Eigenschaft "blockname" wird für den kaskadierbaren Aufruf eine "1" eingetragen.

Wenn die momentan aufgerufene Variable (tagname) des Bildbausteins als AS-Bausteinname einen Namen ohne angehängte Zahl hat, z. B. "L", wird als nächstes ein Baustein mit dem AS-Bausteinname "L1" aus dem gleichen Plan aufgerufen, wenn der Variablenname hierzu im Datenmanager vorhanden ist.

Wenn der momentan aufgerufene Variablenname des Bildbausteins als AS-Bausteinname "L1" hat, wird als nächstes "L2" erwartet usw.

Somit können im Prinzip beliebig viele INTERLOK-Bausteine aufgerufen werden.

Die Projektierungsvorschrift für dieses Szenario lautet folgendermaßen:

- In einem VALVE-Bildbaustein wird eine "OpenNextFaceplate"-Taste mit dem Blocknamen "LOCK" eingebaut.
- Im INTERLOK-Baustein wird eine "OpenNextFaceplate"-Taste mit dem Blocknamen "1" eingebaut.
- INTERLOK-Bausteine, die vom Bildbausteintyp "VALVE" mit dem oben genannten Mechanismus aufgerufen werden, müssen "LOCK", "LOCK1", "LOCK2" usw. heißen.

Bedienberechtigungen

Das Objekt "OpenNextFaceplate" hat ebenfalls die Eigenschaften "Processcontrolling_backup" und "HigherProcesscontrolling_backup" zum Einstellen der Bedienberechtigungen des aufzurufenden Bildbausteines. Die Default-Einstellung ist hier, wie bisher gewohnt, die Berechtigungsstufe 5 und 6. Wenn Sie die aus dem Quellsymbol eingetragenen Berechtigungen weiterleiten wollen, erreichen Sie dies über die Direktverbindungen von "@Level5" und "@Level6" nach "OpenNextFaceplate".

Direktverbindungen

@Level5/Berechtigung/Änderung → OpenNextFaceplate/Processcontrolling

@Level6/Berechtigung/Änderung → OpenNextFaceplate/HigherProcesscontrolling

2.3.26 Taste "Meldungen Sperren/Freigeben"

Objekte in der Übersicht

Die Funktion "Meldungen sperren/freigeben" wird über die Taste  in der Übersicht realisiert.

Die Taste ist nur sichtbar, wenn der angemeldete Operator die Berechtigungsstufe hat, die am Bausteinsymbol unter der Eigenschaft "Processcontrolling_backup" hinterlegt ist.

Eigenschaften

Objekttyp	Objektname	Bildname
Button	Button17	@PCS7Elements.pdl

Die Taste für das Sperren/Freigeben von Meldungen entspricht der Taste im Standard-Tastensatz "@Buttons11.pdl", jedoch mit einem anderem Skript für die instanzspezifische Melden und Sperren.

2.3.27 Quality Code-Anzeigen

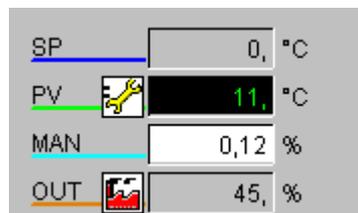
Darstellung der Quality Code-Anzeige

Die Quality Code-Anzeige ist eine erweiterte Zustandsanzeige mit 7 Alternativen.

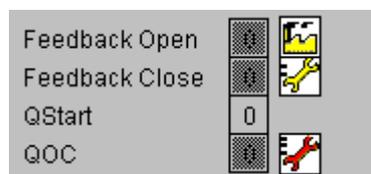
Quality Code	Klartext	Symbol
0x68	Qualität unbestimmt, gerätebezogen	
0x78 0x54	Qualität unbestimmt, prozessbezogen	
0x14 0x18 0x00	Qualität schlecht, gerätebezogen	
0x08 0x28 *)	Qualität schlecht, prozessbezogen	
0x44 0x48 0x60	Simulierter Wert	
0xA4	Wartungsbedarf	
alle anderen Werte	Qualität gut	(Kein Symbol)
*)	Der Code 0x08 wird in der OS generiert, wenn sie den Kontakt zum AS verloren hat.	

Der Klartext wird als Kurzinformation angezeigt, wenn der Mauszeiger über die Quality Code-Anzeige geführt wird.

Bei den **Analogwerten** wird die Anzeige für den Quality Code vor den Analogwerten eingeblendet.



Bei den **Binärwerten** wird in den Bildbausteinen für Ventile und Motoren in der Wartungssicht die Anzeige für den Quality Code hinter den Binärwertanzeigen eingeblendet.



Eigenschaften

Objekttyp	Objektname	Bildname
Anwender-Objekt	AnwenderObjekt1	@PCS7Elements.pdl

2.4 Skripte

Liste der Skripte

Die hier aufgelisteten Skripte werden im Verzeichnis "`\\Siemens\WinCC\aplib\FacePlateDesigner_V6`" bzw. "`\\Siemens\WinCC\aplib\FacePlateDesigner`" installiert und müssen **nicht** in den GraCS-Ordner des Projektverzeichnisses kopiert werden. Die Skripte sind nicht projektspezifisch, sondern rechner-spezifisch.

Skript-Name	Funktion
PCS7_OpenGroupDisplay_V6.Fct	Öffnen eines Bildbausteins "Gruppendarstellung"
PCS7_OpenGroupDisplay_I_V6.Fct	Öffnen eines zugehörigen INTERLOK-Bausteins bei Antrieben mit Mausklick "Rechts"
PCS7_UpdateGroupTagname_V6.fct	Das Skript wird bei Änderung der Eigenschaft "tagname" des Objekts "@Faceplate" aufgerufen. Die Eigenschaft "tagname" wird bei Aufruf des Bildbausteins beschrieben.
PCS7_OpenLoopDisplay_V6.Fct	Öffnen der Loop-Darstellung, Aufruf von Taste "Loop" in "@PG_%Type%.pd1"
PCS7_UpdateLoopTagname_V6.Fct	Das Skript wird aufgerufen bei Änderung der Eigenschaft "tagname" des Objekts "@Faceplate" im Loop-Bild. Die Eigenschaft "tagname" wird bei Aufruf des Bildbausteins beschrieben.
PCS7_CheckPermission.fct	Bedienberechtigung prüfen
PCS7_UpdatePermission_V6.Fct	Aufruf bei Änderung von "@CurrentUser" und Bildanwahl im Bild "@PG_%Type%.pd1" und im Bild "@PG_%Type%.pd1"
PCS7_ChangeView.fct	Aufruf einer anderen Sicht im Bildbaustein, Aufruf aus "@PG_%Type%_Viewlist.pd1"
PCS7_OperationLog_V6.fct	Bedienmeldung fuer WinCC
PCS7_OpenCheckbox_V6.Fct	Aufruf aus Basiselement "CHECKBOX_L1/R1". Öffnen des Bedienbildes "@PCS7_BedCheck.pd1"
PCS7_Check_OK_V6.fct	Skript wird aufgerufen von "OK"-Taste aus "@PCS7_BedCheck.pd1"
PCS7_OpenComboBox_V6.fct	Öffnen von Bedienbild "@PCS7_BedCombo.pd1" aus dem Basiselement "PCS7_COMBOBOX1"
PCS7_Combo_OK_V6.fct	Skript wird aufgerufen von "OK"-Taste aus "@PCS7_BedCombo.pd1"
PCS7_Open3ComboBox_V6.fct	Öffnen von Bedienbild "@PCS7_3BedCombo.pd1" aus dem Basiselement "PCS7_3COMBOBOX1"
PCS7_3Combo_OK_V6.fct	Skript wird aufgerufen von "OK"-Taste aus "@PCS7_3BedCombo.pd1"
PCS7_OpenInputBoxBin_V6.fct	Öffnen von Bedienbild "@PCS7_BedBinaer.pd1" aus dem Basiselement "PCS7_BinOp" und "PCS7_ButtonBit"
PCS7_Binary_OK_V6.fct	Skript wird aufgerufen von "OK"-Taste aus "@PCS7_BedBinaer.pd1"
PCS7_2Stati_Variable_Changed_V6.fct	Aufruf in Basiselement "PCS7_Status_2_Alternative"

Skript-Name	Funktion
PCS7_1vnStati_Variable_Changed_V6.fct	Aufruf in Basiselement "PCS7_Status_1_v_n"
PCS7_OpenInputBoxAnalog_V6.fct	Aufruf von Bedienbild "@PCS7_BedAnalog.pdl" oder "@PCS7_BedAnalog_NL.pdl" aus dem Basiselement "PCS7_AnalogValue"
PCS7_Analog_OK_V6.fct	Skript wird aufgerufen von "OK"-Taste aus "@PCS7_BedAnalog.pdl" oder "@PCS7_BedAnalog_NL.pdl"
PCS7_AnalogPercent_V6.fct	Skript wird aufgerufen bei inkrementeller Bedienung in "@PCS7_BedAnalog.pdl"
PCS7_UpdateBarLimits_V6.Fct	Skript wird von Basiselement "PCS7_BarLimits" aufgerufen
PCS7_UpdateBar_V6.Fct	Skript wird von Basiselement "PCS7_BarStandard1" und "PCS7_BarStandard2" aufgerufen
PCS7_Format_V6	Skript zur Zahlenformat-Übertragung
PCS7_Trend_V6.Fct	Skript für Trend-Darstellung im Bildbaustein

Die beiden Skripte "PCS7_ChangeView.fct" und "PCS7_CheckPermission.fct" sind unverändert aus der V5 übernommen und haben deswegen keine Endung _V6 im Namen.

2.5 Bitmaps

Liste der Bitmaps

Die Bitmaps werden im Verzeichnis

"...\Siemens\WinCC\options\pd\FaceplateDesigner_V6" installiert.

Bei Ablauf des OS-Projekteditors werden die Bitmaps in den "GraCS-Ordner" des Projektverzeichnisses kopiert.

Die Bitmaps werden in den entsprechenden Bildern bei Bildanwahl dynamisch nachgeladen.

Hinweis: Die hier in der Liste aufgeführten Symbole sind eine Auswahl der gebräuchlichsten Symbole.

Dateiname des Bitmap	Symbol-darstellung	Verwendung in	Verzeichnis
@FP_PopUpIcon.bmp		Bildbaustein global	pd\FaceplateDesigner
@PCS7_AlarmCrossed.bmp		Bildbaustein global	pd\FaceplateDesigner
@PCS7_AlarmDisabled.bmp		Bildbaustein global	pd\FaceplateDesigner
@PCS7_AlarmEnabled.bmp		Bildbaustein global	pd\FaceplateDesigner
@PCS7_NotOccupied.bmp		Bildbaustein global	pd\FaceplateDesigner
@PCS7_Occupied.bmp		Bildbaustein global	pd\FaceplateDesigner
@PCS7_OpenLoop.bmp		Bildbaustein global	pd\FaceplateDesigner
@PCS7_Lock.bmp		Bildbaustein Motor/Ventil	pd\FaceplateDesigner
@PCS7_UnLock.bmp		Bildbaustein Motor/Ventil	pd\FaceplateDesigner
@CollectValue_S.emf		Bausteinsymbole global ASD	
@CollectValue_F.emf		Bausteinsymbole global ASD	
@CollectValue_transparent.		Bausteinsymbole global ASD	
@CollectValue_empty.emf		Bausteinsymbole global ASD	
@Ctrl_Manual.emf		Regler/Bausteinsymbol	
@Ctrl_intern.emf		Regler/Bausteinsymbol	
@Ctrl_extern.emf		Regler/Bausteinsymbol	
@Ctrl_Auto.emf		Regler/Bausteinsymbol	
@Ctrl_Track.emf		Regler/Bausteinsymbol	
@off.emf		Bausteinsymbol OPD	pd\Base_Data_Poo

Dateiname des Bitmap	Symbol-darstellung	Verwendung in	Verzeichnis
@on.emf		Bausteinsymbol OPD	pd\Base_Data_Poo
@Auto.emf		Bausteinsymbole allgemein	pd\Base_Data_Poo
@Manual.emf		Bausteinsymbole allgemein	pd\Base_Data_Poo
@motor_is_off.emf		Bildbaustein und Bausteinsymbol Motor	pd\FaceplateDesigner
@MOTOR_Error.emf		Bildbaustein und Bausteinsymbol Motor	pd\FaceplateDesigner
@motor_is_on.emf		Bildbaustein und Bausteinsymbol Motor	pd\FaceplateDesigner
@motor_off.emf		Bildbaustein und Bausteinsymbol Motor	pd\FaceplateDesigner
@motor_on.emf		Bildbaustein und Bausteinsymbol Motor	pd\FaceplateDesigner
@vaz_h.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@ vaz_h_close.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@ vaz_h_open.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@valve_nv.emf		Bausteinsymbol Ventil (Verriegelung)	pd\FaceplateDesigner
@valve_v.mf		Bausteinsymbol Ventil (Verriegelung)	pd\FaceplateDesigner
@vho_closed.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@vho_opened.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@VHO_Error.emf		Bildbaustein und Bausteinsymbol Ventil	
@vho_undef.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@vhz_closed.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@vhz_opened.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@vhz_.emf		Bildbaustein und Bausteinsymbol Ventil	pd\FaceplateDesigner
@vve_closed.emf		Bausteinsymbol Ventil	pd\FaceplateDesigner
@vve_opened.emf		Bausteinsymbol Ventil	pd\FaceplateDesigner

Dateiname des Bitmap	Symbol-darstellung	Verwendung in	Verzeichnis
@VVE_Error.emf		Bausteinsymbol Ventil	pd\FaceplateDesigner
@vve_undef.emf		Bausteinsymbol Ventil	pd\FaceplateDesigner
@vvt_closed.emf		Bausteinsymbol Ventil	pd\FaceplateDesigner
@vvt_opened.emf		Bausteinsymbol Ventil	pd\FaceplateDesigner
@vvt_undef.emf		Bausteinsymbol Ventil	pd\FaceplateDesigner

2.6 Bilder

Liste der Bilder

Die hier aufgelisteten Bilder werden im Verzeichnis
 "...\Siemens\WinCC\options\pdl\FaceplateDesigner_V6" installiert.

Bei Ablauf des OS-Projekteditors werden diese in den "GraCS-Ordner" des
 Projektverzeichnis kopiert.

Bilder	
@PCS7Elements.Pdl	Vorlagenbild für Basiselemente
@@PCS7Typicals.pdl	Vorlagenbild für Bausteinsymbole aus der Technologischen Hierarchie erzeugen
@Template.pdl	Vorlagenbild von Bausteinsymbole für Graphics Object Update. Der Unterschied zu "@@PCS7Typicals.pdl" ist nur die Eigenschaft "type".
@PCS7_BedAnalog.pdl	Bedienbild für Analogwerte
@PCS7_BedAnalog_NL.pdl	Bedienbild für Analogwerte ohne Grenzen, ohne Schiebepalken und ohne inkrementelle Bedienung.
@PCS7_BedBinaer.pdl	Bedienbild für Binär, 1 Punkt-Bedienung, nur Bestätigung (z. B. Auf/Zu/Hand/Auto bei Ventil/Motor).
@PCS7_BedCheck.pdl	Bedienbild binär mit 2-Punkt-Bedienung Button (z. B. Alarm/Warnung aktiv setzen)
@PCS7_BedKombo.pdl	Bedienbild binär für 2-Punkt-Bedienung Liste (z. B. Hand/Auto bei Regler)
@PCS7_3BedKombo.pdl	Bedienbild binär für 3-Punkt-Bedienung Liste (z. B. Hand/Auto bei Regler)
@PCS7_AnalogInputwithLimits.pdl	Bedienbild für Analogwerte (wird nicht verwendet, aber wegen Kompatibilität zu V5.1 weiterhin geliefert)
@PCS7_BinaryInput1of2.pdl	Bedienbild für Binärwerte (wird nicht verwendet, aber wegen Kompatibilität zu V5.1 weiterhin geliefert)
@PCS7_ALARM.pdl	Darstellung der Melde-Sicht in den Bildbausteinen mit Alarm
@PCS7_BATCH.pdl	Darstellung der Batch-Sicht in den Bildbausteinen
@PCS7_TREND.pdl	Darstellung der Trend-Sicht in den analogen Bildbausteinen
@PG_%Type%.pd	Vorlagenbild für Prototypbild-Bildbaustein
@PG_%Type%_%View%.pd	Vorlagenbild Bildfenster für die unterlagerten Sichten
@PG_%Type%_VIEWLIST.pdl	Vorlagenbild für Sichten-Auswahlmenü
@PG_%Type%_OverView.pdl	Vorlagenbild für Übersicht
@PL_%Type%.pd	Vorlagenbild für Loop-Darstellung

2.7 Bildbausteine

Allgemeines

Mit welchen Parametern der AS-Bausteininstanz die einzelnen Anzeigeobjekte im Bildbaustein visualisiert werden, können Sie im Graphics Designer Offline-Dialog nachsehen.

Bausteinkommentar

Der CFC-Bausteinkommentar wird im Bildbaustein als Kurzinformationstext (Tooltip) vom Variablennamen angezeigt.

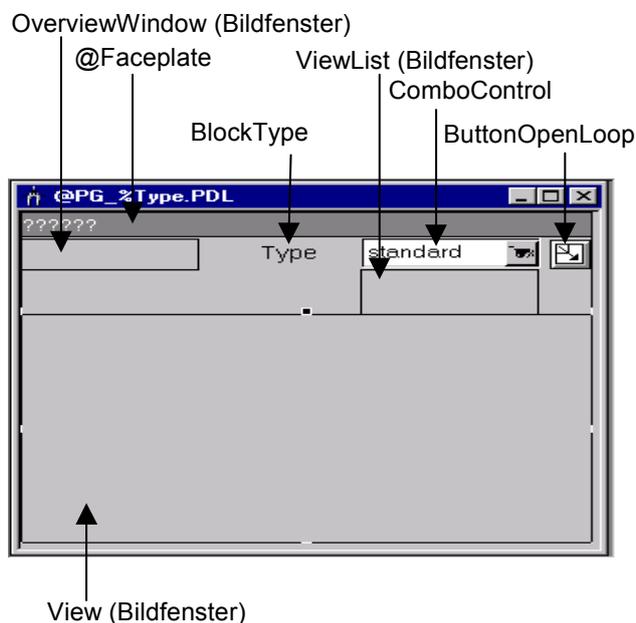
Damit wird erreicht, dass die Beschreibung der technologischen Funktion im Bedarfsfall auch im Bildbaustein verfügbar ist.

2.7.1 Grunddaten der Vorlagenbilder

2.7.1.1 @PG_%Type%.pdl

Übersicht

Für die Projektierung steht die Vorlage "@PG_%Type%.pdl" zur Verfügung.



Im Folgenden wird die Funktionalität der einzelnen Elemente beschrieben:

OverviewWindow

Bildfenster zur Darstellung des Alarmzustandes, Batch-Zustandes, instanzspezifischer Meldungsquittierung und Meldungsfreigabe für den Bildbaustein. Normalerweise wird hier eine Sammelanzeige eingeblendet. Wird der Bildbaustein für einen Multiinstanzbaustein verwendet, können hier auch mehrere Sammelanzeigen angezeigt werden. Die Anzeige wird über die Projektierung des Bildes "@PG_<Type>_Overview.pdf" festgelegt. Das gleiche Bild wird auch in der Loop-Sicht des Bildbausteins angezeigt.

@Faceplate

Dient zur Anzeige der Bausteininstanz bzw. der Variablen aus dem aufrufenden Objekt. Hier werden auch weitere Zusatzinfos für den Bildbaustein hinterlegt, wie FirstView, Name der Batch-Variablen, CurrentUser usw.

TrendFunktionen

Anwenderobjekt mit EA-Feld, für Speicherung von Daten der Trendfunktionen "Trendseite". Das Objekt ist online nicht sichtbar.

BlockType

Name des Bildbaustein-Typs. Dieser Typ-Name geht in die Namen aller zum Bildbaustein gehörigen Sichten ein. Das Objekt ist online nicht sichtbar.

ViewList

Bildfenster zur Darstellung und Anwahl der vorhandenen Sichten.

ComboControl

Anwahl- und Anzeigeelement für die verschiedenen Sichten. Dieses Element zeigt immer den Namen der aktuellen Sicht an.

ButtonOpenLoop

Objekt zur Anwahl des Loop-Bildes. Dieses Objekt wird vom Faceplate Designer automatisch unsichtbar geschaltet, wenn dort die Generierung bzw. das Update der Loop-Sicht nicht angewählt ist.

View

Bildfenster zur Darstellung der einzelnen Sichten des Bildbausteins.

OperationWindow

Bildfenster zum Aufblenden der Analogwertbedienungen.

ComboWindow

Bildfenster zum Aufblenden der Binärwertbedienungen.

2.7.1.2 @PG_%TYPE%

Bildobjekt @PG_%TYPE%

Geometrie/Pos	X=0, y = 0
Geometrie/Größe	Breite = 320, Höhe = 260

Bildfenster "View"

Geometrie/Pos	X=1, y = 47
Geometrie/Größe	Breite = 320, Höhe = 214

Anwenderobjekt @Faceplate

Das Anwenderobjekt "@Faceplate" besteht aus folgenden Elementen:

Eigenschaft im Anwenderobjekt	Element	Typ	Vorbesetzung
Geometrie/Pos	X=0, y = 0		
Geometrie/Größe	Breite = 320, Höhe = 20		
Tagname	Tagname	Stat. Text	Keine
Tag	Tag	Stat. Text	Text = MKZ
FirstView	FirstView	EA-Feld	Ausgabewert wird vom Faceplate Designer gesetzt
CurrentUser	CurrentUser	EA-Feld	Ausgabewert = Verschaltung auf interne Variable @Current User
BName	VarBatchname	EA-Feld	Ausgabewert = .BA_NA
BATCH_ID	VarBatchID	EA-Feld	Ausgabewert = .BA_ID
STEP_NO	VarBatch Schrittnummer	EA-Feld	Ausgabewert = .STEP_NO
STEP_N1	VarBatch Schrittnummer_N1	EA-Feld	Ausgabewert = .STEP_N1
Areaname	Areaname	EA-Feld	Ausgabewert = .#areaname
Processcontrolling_backup	POP	EA-Feld	Berechtigung = Processcontrolling
HigherProcesscontrolling_backup	HIPOP	EA-Feld	Berechtigung = Higher Processcontrolling
MULTI_INSTANCE	HIPOP.Verdeckte Eingabe	EA-Feld	True = MultiInstanceFaceplate

2.7.1.3 @PG_%Type%_%View%.pdl

Bildobjekt @PG_%Type%_%View%

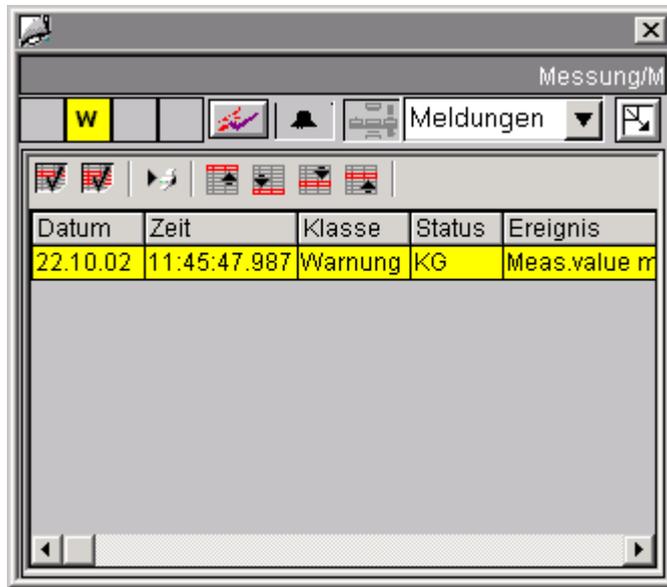
Geometrie/Pos	X=0, y = 0
Geometrie/Größe	Breite = 320, Höhe = 214

Rechteck @Frame

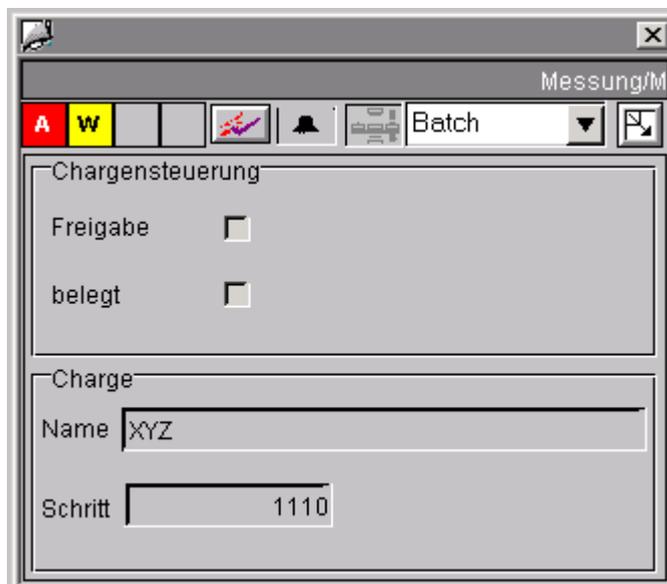
Geometrie/Pos	X=1, y = 50
Geometrie/Größe	Breite = 320, Höhe = 214

2.7.2 Globale Sichten

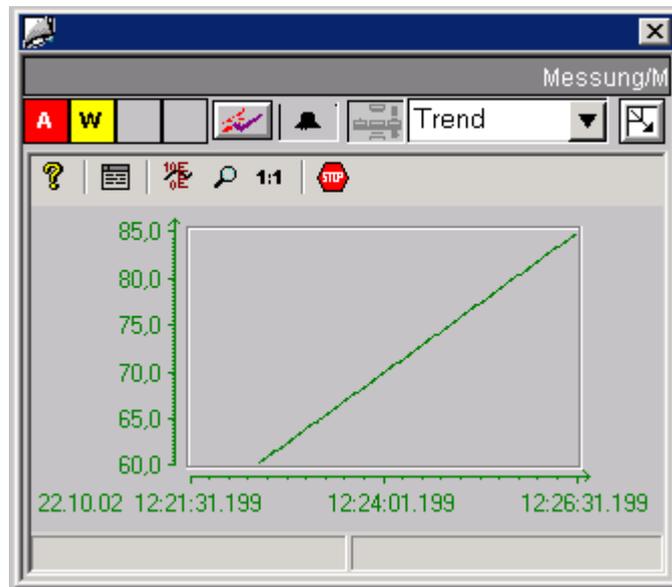
2.7.2.1 Meldesicht



2.7.2.2 Batch-Sicht



2.7.2.3 Trendsicht



Siehe dazu auch Kapitel 2.1.8, "Trendsicht projektieren".

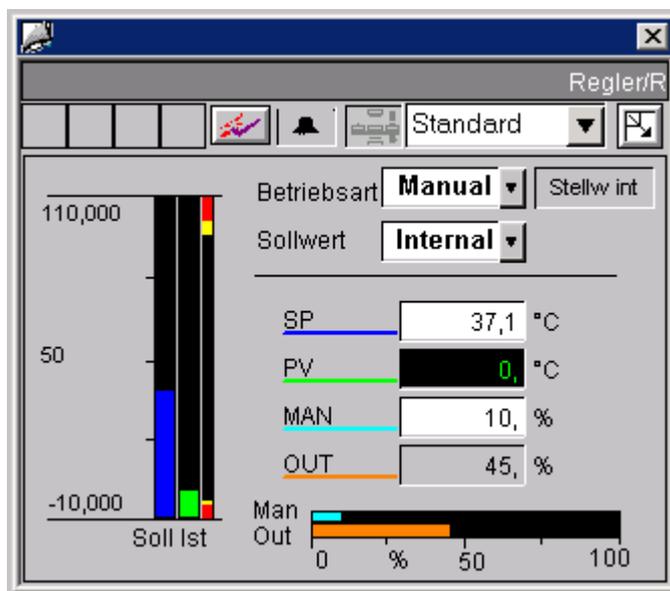
2.7.3 CTRL_PID

2.7.3.1 CTRL_PID: Sichten

Nachfolgend wird stellvertretend für alle PCS 7-Bildbausteine der Bildbaustein CTRL_PID mit seinen Sichten "Standardsicht", "Wartungssicht", "Parametersicht" und "Grenzansicht" beschrieben.

Informationen über alle Bildbausteine finden Sie in der Online-Hilfe der PCS 7-Bildbausteine.

2.7.3.2 CTRL_PID: Standardsicht



Bildbaustein-Standardsicht ab V6.0

Analoganzeigen und Zahlenformat

Alle Analoganzeigen sind mit dem "AdvancedAnalogDisplay" realisiert. Das Zahlenformat wird über die Eigenschaften "Format_InputValue" und "Format_OutputValue" des Bausteinsymbols versorgt. Siehe Kapitel 2.1.7, "Zahlenformate projektieren".

Bedienberechtigungen

Die Sicht hat die folgenden 2 Permission-Objekte für Sollwert und Stellgrößeneingabe, da für diese Größen die Bedienberechtigung von verschiedenen Faktoren abhängig ist:

- "Permission_Setpoint"
- "Permission_Manual"

Siehe auch Kapitel 2.3.24, Basiselemente, Permission-Objekt.

Außer den WinCC-Berechtigungsstufen werten die einzelnen Permission-Objekte auch folgende Parameter aus:

Permission-Objekt	Parameter
"Permission_Setpoint"	"Q_SP_OP = TRUE"
"Permission_Manual"	"QLMNOP = TRUE"

Bedienung PID-Tuner und Optimierung

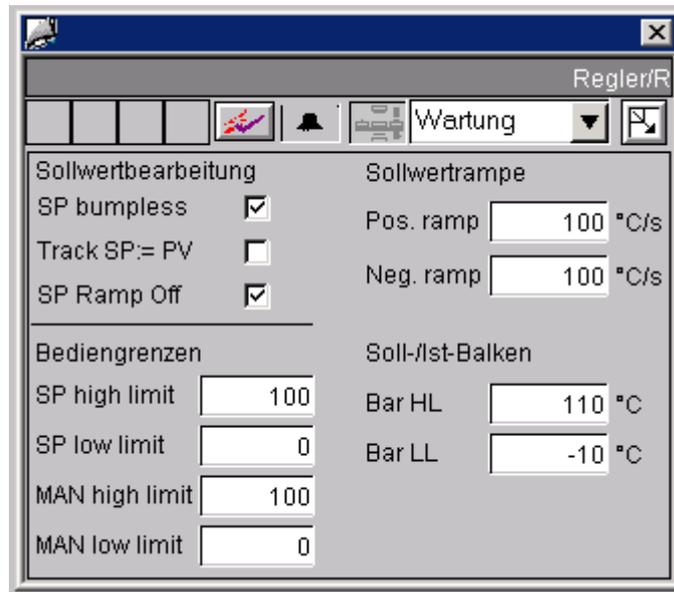
Den PID-Tuner bedienen Sie in der Parametersicht (Optimierung ein/aus).

Wenn Sie die Optimierung in der Parametersicht einschalten, erscheint in der Standardsicht ein Kombinationsfeld oberhalb des Betriebsarten-Kombinationsfeldes "Hand/Automatik". Über dieses Kombinationsfeld können Sie die Optimierung auch aus der Standardsicht wieder ausschalten. Bei "Optimierung Ein" sind alle anderen Bedienungen des Reglers gesperrt.

Reihenfolge und Rangierung von Direktverbindungen auf die bedienbaren Objekte

@Level5	->	Bedienfreigabe		
Manual_COMBOBOX	->	Bedienfreigabe		
External_COMBOBOX	->	Bedienfreigabe		
Permission_Setpoint	->	Level_Source	->	Level_Target
Permission_Manual	->	Level_Source		
Permission_Setpoint	->	Target_Bedienfreigabe		
Setpoint_AnalogValue	->	Bedienfreigabe		
Permission_Manual	->	Target_Bedienfreigabe		
Manual_AnalogValue	->	Bedienfreigabe		
Format	->	Format_InputValue		
Setpoint_AnalogValue	->	Format		
ProcessValue_AnalogValue	->	Format		
Format	->	Format_OutputValue		
Manual_AnalogValue	->	Format		
Output_AnalogValue	->	Format		

2.7.3.3 CTRL_PID: Wartungssicht



Bedienberechtigungen

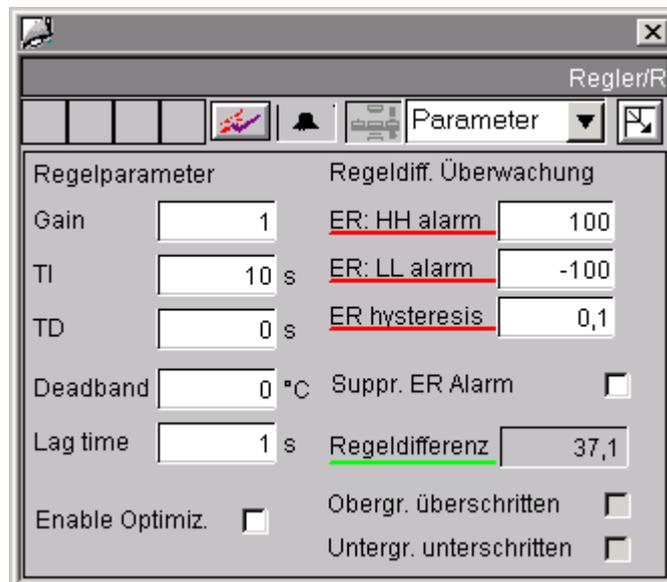
Außer den WinCC-Berechtigungsstufen wertet das Permission-Objekt "Permission_SP_Bumpless" auch den Parameter "OPTI_EN = FALSE" aus.

Reihenfolge und Rangierung von Direktverbindungen auf die bedienbaren Objekte

@Level6	->	Bedienfreigabe
Permission_SP_Bumpless	->	Level_Source
Permission_SP_Bumpless	->	Target_Bedienfreigabe
Bumpless_CHECKBOX_L	->	Bedienfreigabe
SP_TRK_ON_CHECKBOX_L	->	Bedienfreigabe
SPRAMP_OFF_CHECKBOX_L	->	Bedienfreigabe
SPHighLimit_AnalogValue	->	Bedienfreigabe
SPLowLimit_AnalogValue	->	Bedienfreigabe
ManHighLimit_AnalogValue	->	Bedienfreigabe
ManLowLimit_AnalogValue	->	Bedienfreigabe
SPURLM_AnalogValue	->	Bedienfreigabe
SPDRLM_AnalogValue	->	Bedienfreigabe
MO_PVHR_AnalogValue	->	Bedienfreigabe
MO_PVLR_AnalogValue	->	Bedienfreigabe

Permission_SP_Bumpless	->	Format
SPHighLimit_AnalogValue	->	Hintergrundfarbe_Value
SPLowLimit_AnalogValue	->	Hintergrundfarbe_Value
ManHighLimit_AnalogValue	->	Hintergrundfarbe_Value
ManLowLimit_AnalogValue	->	Hintergrundfarbe_Value
SPURLM_AnalogValue	->	Hintergrundfarbe_Value
SPDRLM_AnalogValue	->	Hintergrundfarbe_Value
MO_PVHR_AnalogValue	->	Hintergrundfarbe_Value
MO_PVLR_AnalogValue	->	Hintergrundfarbe_Value

2.7.3.4 CTRL_PID: Parametersicht



Analoganzeigen und Zahlenformat

Der Prozesswert "Regeldifferenz_AnalogValue" ist mit dem "AdvancedAnalogDisplay" realisiert. Das Zahlenformat wird über die Eigenschaft "Format_InputValue" des Bausteinsymbols versorgt.

Alle anderen Analoganzeigen sind mit dem herkömmlichen EA-Feld "Gleitpunktformat" realisiert.

Bedienberechtigungen

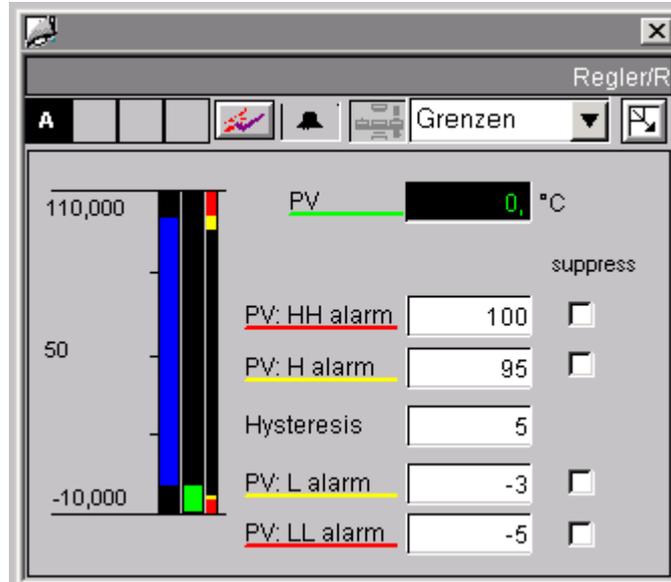
Außer den WinCC-Berechtigungsstufen wertet das Permission-Objekt "Permission_Gain" auch den Parameter "OPTI_EN = FALSE" aus.

Reihenfolge und Rangierung von Direktverbindungen auf die bedienbaren Objekte

@Level6	->	Bedienfreigabe
Permission_Gain	->	Level_Target
OPTI_EN_CHECKBOX_L	->	Level_Source
Permission_Gain	->	Target_Bedienfreigabe
Gain_AnalogValue	->	Bedienfreigabe
TN_AnalogValue	->	Bedienfreigabe
TV_AnalogValue	->	Bedienfreigabe
DEADB_W_AnalogValue	->	Bedienfreigabe
TM_LAG_AnalogValue	->	Bedienfreigabe
ERH_ALM_AnalogValue	->	Bedienfreigabe
ERL_ALM_AnalogValue	->	Bedienfreigabe
ER_HYS_AnalogValue3	->	Bedienfreigabe
M_SUP_ER_CHECKBOX_L	->	Bedienfreigabe
Permission_Gain	->	Target_Hintergrundfarbe
Gain_AnalogValue	->	Hintergrundfarbe_Value
TN_AnalogValue	->	Hintergrundfarbe_Value
TV_AnalogValue	->	Hintergrundfarbe_Value
DEADB_W_AnalogValue	->	Hintergrundfarbe_Value
TM_LAG_AnalogValue	->	Hintergrundfarbe_Value
ERH_ALM_AnalogValue	->	Hintergrundfarbe_Value
ERL_ALM_AnalogValue	->	Hintergrundfarbe_Value
ER_HYS_AnalogValue3	->	Hintergrundfarbe_Value
Format	->	Format_InputValue
Regeldifferenz_AnalogValue	->	Format

2.7.3.5 CTRL_PID: Grenzsicht

Bediengrenzen des Sollwerts



Der Sollwert-Balken zeigt hier die Bediengrenzen für den Sollwert an, bezogen auf die Balkengrenzen.

Die Bediengrenzen des Sollwerts werden in der Wartungssicht eingestellt.

Analoganzeigen und Zahlenformat

Der Prozesswert "ProcessValue_AnalogValue" ist mit dem "AdvancedAnalogDisplay" realisiert. Das Zahlenformat wird über die Eigenschaft "Format_InputValue" des Bausteinsymbols versorgt.

Alle anderen Analoganzeigen sind mit dem herkömmlichen EA-Feld "Gleitpunktformat" realisiert.

Bedienberechtigungen

Außer den WinCC-Berechtigungsstufen wertet das Permission-Objekt "Permission_AlarmHigh_AnalogValue" auch den Parameter "OPTI_EN = FALSE" aus.

Reihenfolge und Rangierung von Direktverbindungen auf die bedienbaren Objekte

@Level6		Bedienfreigabe
Permission_AlarmHigh_AnalogValue	->	Level_Source
Permission_AlarmHigh_AnalogValue	->	Target_Bedienfreigabe
AlarmHigh_AnalogValue	->	Bedienfreigabe
WarningHigh_AnalogValue	->	Bedienfreigabe
Hysterese_AnalogValue	->	Bedienfreigabe
WarningLow_AnalogValue	->	Bedienfreigabe
AlarmLow_AnalogValue	->	Bedienfreigabe
AlarmHigh_CHECKBOX_R	->	Bedienfreigabe
WarningHigh_CHECKBOX_R	->	Bedienfreigabe
WarningLow_CHECKBOX_R	->	Bedienfreigabe
AlarmLow_CHECKBOX_R	->	Bedienfreigabe
Permission_AlarmHigh_AnalogValue	->	Target_Hintergrundfarbe
AlarmHigh_AnalogValue	->	Hintergrundfarbe_Value
WarningHigh_AnalogValue	->	Hintergrundfarbe_Value
Hysterese_AnalogValue	->	Hintergrundfarbe_Value
WarningLow_AnalogValue	->	Hintergrundfarbe_Value
AlarmLow_AnalogValue	->	Hintergrundfarbe_Value
Format	->	Format_InputValue
ProcessValue_AnalogValue	->	Format

2.8 Bausteinsymbole

2.8.1 Prozessbedienung

Hinweis

Bei den Bausteinsymbolen sind keine Prozessbedienungen vorgesehen. Alle Prozessbedienungen erfolgen aus den Bildbausteinen.

2.8.2 Vorlagenbilder @@PCS7Typicals.pdl und @Template.pdl

Funktionen der Bausteinsymbole

Die bisher ausgelieferten Bausteinsymbole (V5.x) in den Bildern **@@PCS7Typicals.pdl** und **@Template.pdl** können alle Varianten von Bildbausteinen (OCX oder Bildbaustein V5.1/V5.2/V6.0) im entsprechenden Prototypbild öffnen.

Die neuen Funktionen stehen allerdings nur zur Verfügung, wenn die neuen Bausteinsymbole verwendet werden.

Die neuen Bausteinsymbole sind in den Bildern **@@PCS7Typicals.pdl** und **@Template.pdl** abgelegt.

@@PCS7Typicals.pdl

Das Bild **@@PCS7Typicals.pdl** wird für das automatische Anlegen von Bausteinsymbolen aus der TH verwendet.

Für alle OS-relevanten CFC-Bausteine können in den Plänen des Hierarchieordners und, je nach Einstellung für die unterlagerten Ordner, die Bausteinsymbole in einem Bild angelegt werden, wenn das Bild folgende Voraussetzungen erfüllt:

- Es ist in der Technologischen Hierarchie (TH) enthalten.
- Die Option "Bausteinsymbole aus der TH ableiten" ist gesetzt.

Um die Bausteinsymbole anzulegen, gibt es folgende Möglichkeiten:

- Wählen Sie in der TH den Menübefehl "Bausteinsymbole erstellen/ändern".
- Schalten Sie beim "OS übersetzen" die entsprechende Option im Assistenten ein.

Hierbei gilt:

Für eine CFC-Bausteininstanz mit dem symbolischen Typ-Namen CTRL_PID wird in diesem Bild eine Kopie von einem Bausteinsymbol aus dem Bild **@@PCS7Typicals.pdl** angelegt, an dessen Eigenschaft "type" der String "@CTRL_PID/1" eingetragen ist.

Wenn Sie das Bild "@@PCS7Typicals.pdl" ändern wollen, müssen Sie es unter dem Namen "@PCS7Typicals.pdl" speichern und dieses verändern. Beim Ableiten aus der TH wird automatisch das Bild "@PCS7Typicals.pdl" verwendet, wenn dieses im Projekt vorhanden ist.

Hinweis

Beim automatischen Erzeugen werden alle Bausteinsymbole in dem Bild gelöscht, die auch in "@@PCS7Typicals.pdl" vorkommen, aber nicht über die TH generiert wurden. Verwenden Sie deshalb bei manueller Projektierung und wenn Sie Bilder nachbearbeiten, als Vorlage für Bausteinsymbole das Bild "@Template.pdl", da hier die Eigenschaft "type" anders vorbelegt ist.

Neu ab V6 ist, dass an einer CFC-Bausteininstanz diese Referenz nun projektierbar ist und keine zwingende Namenskonvention an der Eigenschaft "type" besteht. Des Weiteren können im ES für einen Bausteintyp mehrere verschiedene Bausteinsymbole erzeugt werden.

Beispiel:

Wenn an einer CTRL_PID-Instanz als Symbolname "XXX" eingetragen wird, wird in dem Bild **@@PCS7Typicals.pdl** nach einem Bausteinsymbol referenziert, das in der Eigenschaft "type" den String "@CTRL_PID/XXX" eingetragen hat.

@Template.pdl

Das Bild **@Template.pdl** gilt vor allem als Vorlage für manuelle Projektierung von Bausteinsymbolen in WinCC-Bildern. Der Unterschied der Bausteinsymbole in diesen beiden Bildern liegt lediglich in der Eigenschaft "**type**". Diese Eigenschaft darf in dem Bild "@@PCS7Typicals.pdl" nicht verändert werden (Namenskonvention z. B. @MEAS_MON/1), da dies die Referenz zur Bestimmung der Objekte ist, die bei der Generierung über die TH erzeugt und gelöscht werden.

In **@Template.pdl** darf diese Eigenschaft verändert werden.

Geben Sie der Eigenschaft immer einen anderen Namen, als sie bei den Bausteinsymbolen in @@PCS7Typicals bereits besitzt. Sonst besteht die Gefahr, dass Bausteinsymbole, die aus dieser Vorlage kopiert wurden, in den Bildern gelöscht werden, die über die TH generiert werden.

Wenn der Bedarf besteht, die vorhandenen Symbole zu verändern, ist es sinnvoll, das Bild "@Template.pdl" unter einem anderen Namen zu speichern und dieses zu ändern. Das Bild wird sonst beim OS-Projekteditor (früher Split Screen Wizard) wieder zurückgesetzt.

Aktualisieren der Bildobjekte

Für den Wizard "**Aktualisieren der Bildobjekte**" kann sowohl das Bild **@@PCS7Typicals.pdl** als auch das Bild **@Template.pdl** verwendet werden.

Auch hier gilt die Eigenschaft "type" als Referenz zur Bestimmung der Objekte, die ausgetauscht werden sollen.

2.8.3 Bausteinsymbole im Bild @@PCS7_Typicals

Übersicht

REGLER	CTRL_PID	CTRL_S	DOSE	FMCS_PID	FMT_PID
tagname 999999,9 999999,9 Einheit 999999,9 M I T A	tagname 999999,9 999999,9 Einheit 999999,9 M I T A	tagname 999999,9 999999,9 Einheit 999999,9 I A	tagname 999999,9 999999,9 Einheit 999999,9 M E T A	tagname 999999,9 999999,9 Einheit 999999,9 M E T A	tagname 999999,9 999999,9 Einheit 999999,9 M I T A
ANALOGWERT	ELAP CNT	MEAS MON	SWIT CNT		
SAMMELANZEIGE	tagname 999999,9 A	tagname 999999,9 A	tagname 999999,9 A		
ANALOGWERT	RATIO P	OP_A	OP_A LIM	OP_A RJC	
	tagname 999999,9 I	tagname 999999,9	tagname 999999,9	tagname 999999,9	
VENTIL	VALVE	VAL MOT			
	tagname L S M S	tagname L S M L S	tagname L S M S	tagname L S M L S	
MOTOR	MOTOR	MOT SPED	MOT REV		
	tagname L M S	tagname L M S	tagname L M S		
BINÄRWERT	OP_D	OP_D3	OP_TRIG	BINÄRWERT	DIG_MON
	tagname ●	tagname ● 3	tagname ●	SAMMELANZEIGE	tagname ● A
Sonstige	INTERLOK	SFC_PLAN	SFC_TYP		
	tagname	SFC-Plan A	SFC-Type A		

2.8.4 Eigenschaften der Bausteinsymbole

2.8.4.1 Allgemeine Eigenschaften

Grundlegende Eigenschaften

Verändern Sie bei allen Bausteinsymbolen aus dem Bild "@@PCS7Typicals" folgende Eigenschaften grundsätzlich nicht:

- Geometrie/Breite
- Geometrie/Höhe
- Sonstige/Bedienfreigabe
- Sonstige/Passwort
- Sonstige/Anzeige
- General/Servername
- Styles/Sammelrelevant (nur bei Bausteinen mit Alarm_8P-Meldungen)

In allen Bausteinsymbolen vorhandene Eigenschaften

Folgende Eigenschaften sind bei allen Bausteinsymbolen vorhanden:

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Sonstige/Processcontrolling_backup	POP.Berechtigung	EA-Feld	Instanzspez. Bedienberechtigung, Default = 5
Sonstige/HigherProcesscontrolling_backup	HIPOP.Berechtigung	EA-Feld	Instanzspez. Bedienberechtigung, Default = 6
General/tag	NameOfTag.Ausgabewert	EA-Feld	Angezeigter Text im Symbol
General/type	Type.Ausgabewert	EA-Feld	Referenz für Symbolgenerierung aus der TH und für Wizards
General/tagname	Tagname.Ausgabewert	EA-Feld	Tatsächlicher Variablenname, der an die Variablen-Präfixe der Bildfenster weitergeleitet wird
General/Servername	Servername.Ausgabewert	EA-Feld	Baustein- bzw. Bildbaustein- Typ
General/Version	Version.Ausgabewert	EA-Feld	Versionsnummer
Styles/View_Tag	NameOfTag.Anzeige Rechteck17.Anzeige (wenn vorhanden)	Rechteck EA-Feld	Hier kann die Anzeige des Variablennamens ausgeblendet werden.
MouseClicked links	PCS7_OpenGroupDisplay_V6 (IpszPictureName, IpszObjectName)		Aufruf des Bildbausteins

2.8.4.2 CTRL_PID

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 110/Höhe = 77		
General/UnitPV	UnitPV.Text	Stat.Text	Anzeige: Einheit PV
General/ Unit_MAN_OP	Unit_MAN_OP.Text	Stat.Text	Anzeige: Einheit MAN_OP
Links/ CollectValue	Sammelanzeige.Sammelwert	Sammelanzeige.	.EventState
Links/ SetpointValue	SetpointValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Sollwert
Links/ ProcessValue	ProcessValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Istwert
Links/ OutputValue	OutputValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Stellgröße
Links / LMN_SEL	Tracking_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Stellwert nachführen
Links/ Mode_MAN_AUT	Manual_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Hand/ Automatik
Links/ Mode_INT_EXT	External_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Extern/ Intern
Styles/ ReturnPath	TrendFunktionen2 .Ausgabewert	EA-Feld	Siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/ StandardTrend	TrendFunktionen2 .Zeichensatzgröße	EA-Feld	Siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/ Format_InputValue	ProcessValue_Analoganzeige Erweitert.Format SetpointValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis. AdvancedAnalogDis.	Zahlenformatierung für Istwert und Sollwert
Styles/ Format_OutputValue	OutputValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis.	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	Weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

2.8.4.3 CTRL_S

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 110/Höhe = 77		
General/UnitPV	Unit.Text/.PV_IN#unit	Stat. Text	Anzeige: Einheit PV
General/Unit_MAN_OP	Unit.Text/.MAN_OP#unit	Stat. Text	Anzeige: Einheit Stellgröße
Links/CollectValue	Sammelanzeige.Sammelwert/	Sammelanzeige	.EventState
Links/SetpointValue	SetpointValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Sollwert
Links/ProcessValue	ProcessValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Istwert
Links/OutputValue	OutputValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Stellgröße
Links/Mode_MAN_AUT	Manual_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Hand/ Automatik
Links/Mode_INT_EXT	External_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Extern/ Intern
Links /LMN_SEL	Tracking_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Stellwert nachführen
Links /QLMNR_ON	OutputValue_Analoganzeige Erweitert.Anzeige Unit_MAN_OP.Anzeige	AdvancedAnalogDis. Stat. Text	Beschr. Siehe unten
Links /QLMNUP	LMNUP_StatusAnzeige	Stat. Text	Anzeige: QLMNUP
Links /QLMNDN	LMNDN_StatusAnzeige	Stat. Text	Anzeige: QLMNDN
Styles/ReturnPath	TrendFunktionen2 .Ausgabewert	EA-Feld	siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/StandardTrend	TrendFunktionen2 .Zeichensatzgröße	EA-Feld	siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/Format_InputValue	ProcessValue_Analoganzeige Erweitert.Format SetpointValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis. AdvancedAnalogDis.	Zahlenformatierung für Istwert und Sollwert
Styles/Format_OutputValue	OutputValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis.	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

Das Bausteinsymbol CTRL_S unterscheidet sich gegenüber dem Bausteinsymbol CTRL_PID in folgendem Punkt:

Bei nicht vorhandener Stellungsrückmeldung (LMNR_ON = 0) werden anstatt der Stellgröße die binären Ansteuersignale QLMNUP und QLMNDN angezeigt.

Die Sichtbarkeit dieser Texte wird auch über Skripte gesteuert. Die Skripte werden aufgerufen, wenn Sie die Eigenschaften QLMNUP und QLMNDN ändern.

Hinweis

Die Objekte "OutputValue_AnaloganzeigeErweitert" und "Unit_MAN_OP" müssen im Anwenderobjekt im Vordergrund liegen, damit die Sichtbarkeitssteuerung korrekt funktioniert.

2.8.4.4 DOSE

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 110/Höhe = 63		
General/UnitPV	UnitPV.Text	Stat. Text	Anzeige: Einheit PV
Links/CollectValue	Sammelanzeige.Sammelwert	Sammelanzeige	.EventState
Links/ProcessValue	ProcessValue_AnaloganzeigeErweitert.Wert	AdvancedAnalogDis.	Anzeige: Istwert
Links/SetpointValue	SetpointValue_AnaloganzeigeErweitert.Wert	AdvancedAnalogDis.	Anzeige: Sollwert
Links/SetpointExtern	External_ZustandsanzeigeErweitert.Status SetpointExternValue_AnaloganzeigeErweitert.Anzeige	AdvancedStatusDis AdvancedAnalogDis	.QSPEXTON Beschreibung siehe unten
Links/WertSetpointExtern	SetpointExternValue_AnaloganzeigeErweitert.Wert	AdvancedAnalogDis	wird bei .QSPEXTON über den Sollwert eingeblendet
Styles/ReturnPath	TrendFunktionen2.Ausgabewert	EA-Feld	siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/StandardTrend	TrendFunktionen2.Zeichensatzgröße	EA-Feld	siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/Format_InputValue	ProcessValue_AnaloganzeigeErweitert.Format SetpointValue_AnaloganzeigeErweitert.Format	AdvancedAnalogDis. AdvancedAnalogDis.	Zahlenformatierung für Istwert und Sollwert
Styles/Format_OutputValue	OutputValue_AnaloganzeigeErweitert.Format	AdvancedAnalogDis.	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

Beim DOSE-Baustein gibt es keinen Parameter, der den wirksamen Sollwert repräsentiert. Deshalb wird in Abhängigkeit von QSPEXTON die Sollwertanzeige eingeblendet.

QSPEXTON = 0 → "SetpointValue_AnaloganzeigeErweitert" wird eingeblendet

QSPEXTON = 1 → "SetpointExternValue_AnaloganzeigeErweitert" wird eingeblendet

2.8.4.5 FMCS_PID/FMT_PID

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 110/Höhe = 77		
General/UnitPV	Unit.Text/.PV#unit	Stat. Text	Anzeige: Einheit PV
General/Unit_MAN_OP	Unit.Text/.LMN#unit	Stat. Text	Anzeige: Einheit Stellgröße
Links/CollectValue	Sammelanzeige.Sammelwert/	Sammelanzeige	.EventState
Links/SetpointValue	SetpointValue_Analoganzeige Erweitert.Wert/.SP	AdvancedAnalogDis .	Anzeige: Sollwert
Links/ProcessValue	ProcessValue_Analoganzeige Erweitert.Wert/.PV	AdvancedAnalogDis .	Anzeige: Istwert
Links/OutputValue	OutputValue_Analoganzeige Erweitert.Wert/.LMN	AdvancedAnalogDis .	Anzeige: Stellgröße
Links/Tracking	Tracking_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Tracking LMN
Links/Mode_MAN_AUT	Manual_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Hand/ Automatik
Links/Mode_INT_EXT	External_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Extern/ Intern
Styles/ReturnPath	TrendFunktionen2 .Ausgabewert	EA-Feld	siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/StandardTrend	TrendFunktionen2 .Zeichensatzgröße	EA-Feld	siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/Format_InputValue	ProcessValue_Analoganzeige Erweitert.Format SetpointValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis . AdvancedAnalogDis .	Zahlenformatierung für Istwert und Sollwert
Styles/ Format_OutputValue	OutputValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis .	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

2.8.4.6 ELAP_CNT

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 97/Höhe = 45		
General/Unit	Unit.Text	Stat. Text	Anzeige: Einheit
Links/CollectValue	Sammelanzeige.Sammelwert	Sammelanzeige	.EventState
Links/Output_Value	ProcessValue_AnaloganzeigeErweitert.Wert	AdvancedAnalogDis.	.HOURS Anzeige max. 7 Stellen
Styles/Format_InputValue	ProcessValue_AnaloganzeigeErweitert.Format	AdvancedAnalogDis.	Zahlenformatierung für Istwert und Sollwert
Styles/Format_OutputValue	Format_OutputValue.Ausgabewert	EA-Feld	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

2.8.4.7 MEAS_MON

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 97/Höhe = 45		
General/Unit	Unit.Text	Stat. Text	Anzeige: Einheit
Links/CollectValue	Sammelanzeige.Sammelwert	Sammelanzeige	.EventState
Links/OutputValue	ProcessValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Istwert
Styles/ReturnPath	TrendFunktionen2 .Ausgabewert	EA-Feld	siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/StandardTrend	TrendFunktionen2 .Zeichensatzgröße	EA-Feld	siehe Kapitel 2.1.8, "Trendsicht projektieren"
Styles/Format_InputValue	ProcessValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis.	Zahlenformatierung für Istwert und Sollwert
Styles/ Format_OutputValue	Format_OutputValue. Ausgabewert	EA-Feld.	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

2.8.4.8 SWIT_CNT

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 97/Höhe = 45		
General/Unit	Unit.Text	Stat. Text	.V#UNIT
Links/CollectValue	Sammelanzeige.Sammelwert	Sammelanzeige	.EventState
Links/OutputValue	ProcessValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Istwert
Styles/Format_InputValue	ProcessValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis.	Zahlenformatierung für Istwert und Sollwert
Styles/ Format_OutputValue	Format_OutputValue .Ausgabewert	EA-Feld.	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

2.8.4.9 RATIO_P

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 97/Höhe = 32		
General/Unit	Unit.Text	Stat. Text	Anzeige: Einheit
Links/OutputValue	ProcessValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Istwert
Links/Mode_INT_EXT	External_Zustandsanzeige Erweitert.Status	AdvancedStatusDis.	Anzeige: Extern/ Intern
Styles/Format_InputValue	ProcessValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis.	Zahlenformatierung für Istwert und Sollwert
Styles/ Format_OutputValue	Format_OutputValue .Ausgabewert	EA-Feld.	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

2.8.4.10 OP_A

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 97/Höhe = 32		
General/Unit	Unit.Text	Stat. Text	Anzeige: Einheit
Links/OutputValue	ProcessValue_Analoganzeige Erweitert.Wert	AdvancedAnalogDis.	Anzeige: Istwert
Styles/Format_InputValue	ProcessValue_Analoganzeige Erweitert.Format	AdvancedAnalogDis.	Zahlenformatierung für Istwert und Sollwert
Styles/Format_OutputValue	Format_OutputValue. Ausgabewert	EA-Feld.	Zahlenformatierung für Stellwert
Styles/Format_xx	Format_xx.Ausgabewert	EA-Feld	weiteres Format, siehe Kap. 2.1.7, "Zahlenformate projektieren"

2.8.4.11 OP_A_LIM

Eigenschaften

Eigenschaften und Darstellung wie OP_A; siehe Kapitel 2.8.4.10, OP_A

2.8.4.12 OP__A_RJC

Eigenschaften

Eigenschaften und Darstellung wie OP_A; siehe Kapitel 2.8.4.10, OP_A

2.8.4.13 VALVE

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 90/Höhe = 67		
Links/CollectValue	Sammelanzeige_mitASD .Sammelwert	AdvancedStatusDis.	.EventState
Links/QMAN_AUT	Mode.Status1	AdvancedStatusDis.	Anzeige: Auto/Manual
Links/V_LOCK	Interlock.Status1	AdvancedStatusDis.	Anzeige: Lock
Links/QOPENED	Valve_Status.Status1	AdvancedStatusDis.	Anzeige: Ventil
Links/QCLOSED	Valve_Status.Status2	AdvancedStatusDis.	Anzeige: Ventil
Links/QOPENING	Valve_Status.Status3	AdvancedStatusDis.	Anzeige: Ventil
Links/QCLOSING	Valve_Status.Status4	AdvancedStatusDis.	Anzeige: Ventil

Wenn Sie auf die linke Maustaste klicken, wird der VALVE-Bildbaustein aufgerufen. Wenn Sie auf die rechte Maustaste klicken, wird der zugehörige INTERLOK-Bildbaustein aufgerufen.

Der Bausteinname des INTERLOK-Bausteins ist als Skript-Übergabeparameter hinterlegt; siehe Kapitel 2.4 "Skripte".

Die Default-Einstellung des Bausteinnamens ist "L". Der INTERLOK-Baustein muss im gleichen CFC-Plan wie der VALVE platziert sein.

2.8.4.14 VAL_MOT

Eigenschaften

Eigenschaften und Darstellung wie VALVE; siehe Kapitel 2.8.4.13, VALVE

2.8.4.15 MOTOR

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 90/Höhe = 54		
Links/CollectValue	Sammelanzeige_mitASD .Sammelwert	AdvancedStatusDis.	.EventState
Links/QMAN_AUT	Mode.Status1	AdvancedStatusDis.	Anzeige: Auto/Manual
Links/LOCK	Interlock.Status1	AdvancedStatusDis.	Anzeige: Lock
Links/QRUN	Motor_Status.Status1	AdvancedStatusDis.	Anzeige: Motor
Links/QSTOP	Motor_Status.Status2	AdvancedStatusDis.	Anzeige: Motor

Wenn Sie auf die linke Maustaste klicken, wird der MOTOR-Bildbaustein aufgerufen. Wenn Sie auf die rechte Maustaste klicken, wird der zugehörige INTERLOK-Bildbaustein aufgerufen.

Der Bausteinname des INTERLOK-Bausteins ist als Skript-Übergabeparameter hinterlegt; siehe Kapitel 2.4 "Skripte".

Die Default-Einstellung des Bausteinnamens ist "L". Der INTERLOK-Baustein muss im gleichen CFC-Plan wie der MOTOR platziert sein.

2.8.4.16 MOT_SPED

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 90/Höhe = 53		
Links/CollectValue	Sammelanzeige_mitASD1 .Sammelwert	AdvancedStatusDis.	.EventState
Links/QMAN_AUT	Mode.Status1	AdvancedStatusDis.	Anzeige: Auto/Manual
Links/LOCK	Interlock.Status1	AdvancedStatusDis.	Anzeige: Lock
Links/QRUN	Motor_Status.Status1	AdvancedStatusDis.	Anzeige: Motor
Links/QSTOP	Motor_Status.Status2	AdvancedStatusDis.	Anzeige: Motor
Links/QSPEED	Motor_Status.Status3	AdvancedStatusDis.	Anzeige: Motor
Links/QSTOPING	Motor_Status.Status4	AdvancedStatusDis.	Anzeige: Motor

Wenn Sie auf die linke Maustaste klicken, wird der MOT_SPED-Bildbaustein aufgerufen. Wenn Sie auf die rechte Maustaste klicken, wird der zugehörige INTERLOK-Bildbaustein aufgerufen.

Der Bausteinname des INTERLOK-Bausteins ist als Skript-Übergabeparameter hinterlegt; siehe Kapitel 2.4 "Skripte".

Die Default-Einstellung des Bausteinnamens ist "L". Der INTERLOK-Baustein muss im gleichen CFC-Plan wie der MOT_SPED platziert sein.

2.8.4.17 MOT_REV

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 90/Höhe = 53		
Links/CollectValue	Sammelanzeige_mitASD1 .Sammelwert	AdvancedStatusDis.	.EventState
Links/QMAN_AUT	Mode.Status1	AdvancedStatusDis.	Anzeige: Auto/Manual
Links/LOCK	Interlock.Status1	AdvancedStatusDis.	Anzeige: Lock
Links/QRUN	Motor_Status1.Status1	AdvancedStatusDis.	Anzeige: Motor
Links/QSTOP	Motor_Status1.Status2	AdvancedStatusDis.	Anzeige: Motor
Links/QDIR	Motor_Status1.Status3	AdvancedStatusDis.	Anzeige: Motor

Wenn Sie auf die linke Maustaste klicken, wird der MOT_REV-Bildbaustein aufgerufen. Wenn Sie auf die rechte Maustaste klicken, wird der zugehörige INTERLOK-Bildbaustein aufgerufen.

Der Bausteinname des INTERLOK-Bausteins ist als Skript-Übergabeparameter hinterlegt; siehe Kapitel 2.4 "Skripte".

Die Default-Einstellung für den Bausteinnamen ist "L". Der INTERLOK-Baustein muss im gleichen CFC-Plan wie der MOT_REV platziert sein.

2.8.4.18 INTERLOK

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 108/Höhe = 20		
Links/Link	Lock.AktuellerZustand	ZusAnz.	Lock Symbol

2.8.4.19 OP_D3

Eigenschaften

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 90/Höhe = 45		
Links/Output1	Zustandsanzeige1.Anzeige	Zustandsanzeige	.Q1
Links/Output2	Zustandsanzeige2.Anzeige	Zustandsanzeige.	.Q2
Links/Output3	Zustandsanzeige3.Anzeige	Zustandsanzeige.	.Q3

2.8.4.20 OP_D

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 90/Höhe = 45		
Links/Status	Zustandsanzeige.Aktueller Zustand	Zustandsanzeige	.Q0

2.8.4.21 OP_TRIG

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 90/Höhe = 40		
Links/Status	Zustandsanzeige.Aktueller Zustand	Zustandsanzeige	.SIGNAL

2.8.4.22 DIG_MON

Eigenschaften

Siehe auch Kapitel 2.8.4.1 "Allgemeine Eigenschaften"

Eigenschaften	Element und Eigenschaft im Anwenderobjekt	Objekt	Beschreibung
Geometrie	Breite = 90/Höhe = 40		
Links/Status	Zustandsanzeige.Aktueller Zustand	Zustandsanzeige	.Q
Links/CollectValue	Sammelanzeige3.Sammelwert	Sammelanzeige	.EventState

3 Online-Hilfe erstellen

3.1 Voraussetzungen

Folgende Voraussetzungen sind nötig:

- der in WINDOWS integrierte ASCII-Editor "Notepad" o.ä. zur Erstellung einer Registrierungsdatei
- ein Erstellungswerkzeug für die Hilfethemen (z. B. "RoboHelp")

3.2 Aufbau der Hilfedatei

Hilfedatei

Wenn Sie für Ihre Bausteine eine Online-Hilfe erstellen wollen, schreiben Sie mit dem Hilfe-Erstellungs-System eine Hilfedatei. Der Name dieser Datei ist frei wählbar. Verwenden Sie aus Übersichtlichkeitsgründen den Namen ihrer Bibliothek oder einen gemeinsamen Namen Ihrer Bausteine, z. B. "MYLIB__a.HLP".

Hilfethema

Erstellen Sie für jeden Ihrer Bausteine ein eigenes Hilfethema (Topic). Anschließend müssen Sie für jedes Topic die Einsprungsadresse zur Online-Hilfe definieren ("Topic-ID" + "Map #") und auch in die Registrierungsdatei eintragen (vgl. Kapitel 3.3). Diese Adressen sind frei wählbar, müssen aber innerhalb der jeweiligen Online-Hilfe eindeutig sein.

Wenn es sich um eine relativ umfangreiche Bibliothek handelt, können Sie auch eine hm-Datei erstellen, die alle verwendeten IDs enthält. Bei der Vergabe der MAP-IDs kann das Erstellungswerkzeug RoboHelp diese Datei verwenden.

Einträge in der hm-Datei:

```
// Headerfile für Online-Hilfe Mylib-Funktionsbausteine
//
#define CONTROL    0x10                // dez. 16
#define CONTROL2  0x11                // dez. 17
#define CONTROL3  0x12                // dez. 18
....
```

Jedes Hilfethema enthält neben dem Hilfetext die folgenden Angaben:

- Topic-Titel** Überschrift des Hilfethemas für diesen Baustein
(üblicherweise gleich dem Bausteinnamen, evtl. mit
Kurzbezeichnung der Funktion).
- Topic-ID** Name und Einsprungsadresse des Topics
(Name = Header des Bausteins, siehe **Bild 1-3**)
- Index** Stichwörter, mit denen vom Indexverzeichnis zum Hilfethema
gesprungen werden kann

HLP-Datei und CNT-Datei

Die Hilfe kann aus zwei Dateien bestehen: einer HLP-Datei, die die Hilfethemen enthält, und einer CNT-Datei, die das Inhaltsverzeichnis enthält.

Die CNT-Datei ist dann sinnvoll, wenn die Baustein-Hilfe nicht ausschließlich als Kontext-Hilfe verwendet werden soll. Die Kontext-Hilfe wird mit F1 auf den markierten Baustein aufgerufen. Bei einer Bibliothek mit mehreren Bausteinen können die einzelnen Hilfethemen in einem Inhaltsverzeichnis aufgeführt werden. Damit können Sie zu den Hilfethemen der anderen Bausteine wechseln, ohne dass der betreffende Baustein vorhanden sein muss.

Diese CNT-Datei kann auch in der CNT-Datei eines anderen Hilfeprojektes mit einer INCLUDE-Anweisung aufgenommen werden, z. B. ":include Mylib__a.cnt". Die eingebundene CNT-Datei wird dann im Inhaltsverzeichnis des anderen Hilfeprojektes mit aufgeführt, wenn beide im gleichen Ordner der Installation vorhanden sind.

Online-Hilfe in mehreren Sprachen

Wenn Sie Ihre Online-Hilfe in mehreren Sprachen anbieten wollen, müssen Sie für jede gewünschte Sprache eine eigene Hilfedatei erstellen. Bei PCS 7 besteht der Name aus 8 Zeichen, von denen das letzte Zeichen für die Sprachkennung verwendet wird:

a	deutsch	
b	englisch	
c	französisch	
d	spanisch	wird z. Zt von PCS 7 nicht unterstützt
e	italienisch	wird z. Zt von PCS 7 nicht unterstützt
y	sprachunabhängig	z. B. für Reg-Datei

Über das Dialogfeld **Options > Customize > Language** stellen Sie die Landessprache ein. Mittels Registrierung (vgl. Kapitel 3.3) wird dann von PCS 7 die zur jeweiligen Landessprache passende Hilfedatei aufgerufen.

Zuletzt müssen Sie die Hilfedatei und, wenn vorhanden, die zugehörige CNT-Datei in das Unterverzeichnis des STEP 7-Verzeichnisses kopieren, in dem Ihre Bibliothek oder das Projekt mit Ihren Bausteinen installiert wird.

3.3 Aufbau der Registrierungsdatei

Registrierungsdatei

Schreiben Sie mit dem ASCII-Editor eine Registrierungsdatei, die die Informationen für Ihre Bausteine in die WINDOWS- Registrierung einträgt. Der Name der Registrierungsdatei ist frei wählbar. Verwenden Sie aus Übersichtlichkeitsgründen den Namen ihrer Bibliothek oder einen gemeinsamen Namen Ihrer Bausteine, z. B. "**Mylib_y.reg**".

Beispiel

Beispiel einer Reg-Datei für 3 Bausteine und 5 Sprachversionen (Spanisch und Italienisch mit englischem Hilfetext).

REGEDIT4

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\mylib\ABC]
"Version"="0.2"
"VersionDate"="23.11.2003"
"HelpFileGerman"="S7libs\mylib\MYLIB__a.hlp"
"HelpFileEnglish"="S7libs\mylib\MYLIB__b.hlp"
"HelpFileFrench"="S7libs\mylib\MYLIB__c.hlp"
"HelpFileSpanish"="S7libs\mylib\MYLIB__b.hlp"
"HelpFileItalian"="S7libs\mylib\MYLIB__b.hlp"

[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\mylib\ABC\XYZ]
"CONTROL"=dword:00000010
"CONTROL2"=dword:00000011
"CONTROL3"=dword:00000012
```

Hinweis

Beachten Sie bitte, dass es durch fehlerhafte Einträge in der Registry zu Störungen im Programmablauf kommen kann oder die gewünschte Funktion nicht ausgeführt wird.

Verwenden Sie daher die Schlüssel so wie in dem hier aufgeführten Beispiel.

Die folgenden Werte müssen in dem Schlüssel der Registrierung eingetragen sein:

[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\Name OfLibrary\Author]

Dabei steht der Bibliotheksname für einen von Ihnen frei gewählten Namen für Ihre Bibliothek (hier: mylib). Er entspricht dem Namen des STEP 7-Unterverzeichnisses, in dem Ihre Hilfedatei abgelegt ist. Unter diesem Namen wird Ihre Bibliothek im CFC-Editor angezeigt. **Author** steht für den von Ihnen beim Attribut AUTHOR im Bausteinkopf angegebenen Namen (hier: ABC).

Version

Enthält die Versionsnummer der gesamten Bibliothek. Dieser Eintrag ist optional.

VersionDate

Enthält das Erstellungsdatum der gesamten Bibliothek. Dieser Eintrag ist optional.

Pfad zur Hilfedatei

Enthält für alle gewünschten Sprachen den zum STEP 7-Verzeichnis relativen Pfad zur jeweiligen Hilfedatei, z. B.:

"HelpFileGerman"="S7libs\mylib\MYLIB__a.hlp".

Beachten Sie, dass die Trennzeichen doppelt angegeben werden müssen (\\). Über diesen Eintrag wird die zur im SIMATIC Manager eingestellten Landessprache passende Hilfedatei aufgerufen.

Hinweis

Die Sprachen Italienisch und Spanisch werden bisher nicht von PCS 7 unterstützt, deshalb werden bei diesen Sprachen die Hilfen in Englisch aufgerufen.

Danach müssen Sie den folgenden Schlüssel eintragen:

[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\Name OfLibrary \Author\Family]

Dabei steht **Family** für den von Ihnen beim Attribut FAMILY im Bausteinkopf angegebenen Namen (hier: XYZ).

Darunter müssen Sie für jeden Baustein den Bausteinnamen, wie im Bausteinkopf (Header) aufgeführt (siehe Bild 1-3), mit der Einsprungsadresse in die Hilfedatei angeben, z. B.: "CONTROL"=dword:00000010. Die Einsprungsadresse ist die Nummer der Topic-ID.

Wenn Sie Ihre Bausteine in mehrere Familien gruppiert haben, müssen Sie für jede Familie einen eigenen Schlüssel in die Registrierungsdatei einfügen.

Wenn Sie diese Registrierungsdatei ausgeführt haben (z. B. mit Doppelklick), wird nach dem Markieren eines Bausteins im CFC oder im SIMATIC Manager mit der F1-Taste über die eingestellte Landessprache und die Baustein-Attribute AUTHOR, FAMILY und FUNCTION_BLOCK in der WINDOWS-Registrierung die zugehörige Hilfedatei bestimmt und die betreffende Hilfe angezeigt.

3.4 Besonderheiten für die Hilfe-Erstellung von Muster-SFCs

Abweichend und ergänzend zu den Informationen in den Kapiteln 3.2 und 3.3 ist für die Erstellung von Online-Hilfen für Muster-SFCs Folgendes zu berücksichtigen.

Ablageort

Sie müssen die Hilfen für die Muster-SFCs in Ihr Installationsverzeichnis kopieren. Wir empfehlen für die Ablage den bereits existierenden Ordner "S7Hlp". Dieser Ordner enthält auch die SFC-Hilfen. Nur dadurch ist gewährleistet, dass Sie aus der selbst erstellten Hilfe auch zur SFC-Hilfe wechseln können, siehe unten. Das folgende Beispiel für die Registrierung bezieht sich auf diesen Ablageort.

Einträge für den Schlüssel

Author	Der Eintrag für "Autor" wird vom SFC vergeben, da dieser die Muster-SFCs in ablauffähige Bausteine übersetzt. Immer: ES_SFC .
Family	Den Familiennamen entnehmen Sie dem Eigenschaften-Dialog des Muster-SFC aus dem Feld "Familie" (kein Leerstring). Im Beispiel: SFC
Name	Den Muster-SFC-Namen entnehmen Sie dem Eigenschaften-Dialog des Muster-SFC aus dem Feld "Name" (kein Leerstring). Im Beispiel: "Dosi1" und "Heat1"

Beispiel für eine Registrierungsdatei für 2 Bausteintypen und 2 Sprachversionen

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\SFCTypes]

[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\SFCTypes\ES_SFC]
"HelpFileGerman"="S7Hlp\SFC_Typa.hlp"
"HelpFileENGLISH"="S7Hlp\SFC_Typb.hlp"
"Version"="1.1"
"VersionDate"="14.11.2003"

[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\SFCTypes\ES_SFC\SFC]
"Dosi1"=dword:00000888
"Heat1"=dword:00000889
```

Zur SFC-Hilfe wechseln

Damit Sie aus der selbst erstellten Hilfe zur SFC-Hilfe gelangen, können Sie in den einzelnen Topics einen Sprung auf das (interne) Inhaltsverzeichnis der SFC-Hilfe einfügen. Dieser Sprung sieht wie folgt aus:

[SFC-Hilfe](#)!JumpID('s7jsfcaa.hlp','IDH_CONTENTS')

Hinweis

Die Sprungadresse nach dem grünen doppelt unterstrichenen Text →
!JumpID('s7jsfcaa.hlp','IDH_CONTENTS') wird als verborgener Text formatiert.

4 Bibliothek und Setup erstellen

4.1 Voraussetzung

Zur Erstellung einer lieferfähigen Bibliothek inklusive des dazu gehörigen Setups benötigen Sie ein Programm zur Erstellung von Installationsprogrammen, z. B. "InstallShield".

4.2 So erstellen Sie eine Bibliothek

Vorgehen

Wenn Sie Ihre Bausteine und/oder Muster-SFCs in einer Bibliothek zusammenfassen wollen, gehen Sie folgendermaßen vor:

1. Legen Sie eine neue S7-Bibliothek an und erzeugen Sie darin ein S7-Programm.
2. Tragen Sie die Namen und Nummern Ihrer Bausteine sowie die dazugehörigen Kommentare in die Symboltabelle der Bibliothek ein (dies gilt nicht für Muster-SFCs).
3. Quellen:
Wenn Sie die Quellen der Bausteine mitliefern wollen, kopieren Sie die Quellen aus dem Quellordner Ihres Projektes in den Quellordner der Bibliothek.
4. Bausteine:
Kopieren Sie Ihre Bausteine aus dem Bausteinordner Ihres Projekts in den Bausteinordner der Bibliothek.

Hinweis:

- Falls Sie in Ihren Multiinstanzbausteinen nicht allgemein verfügbare Bausteine aufrufen (SFBs, SFCs), kopieren Sie diese ebenfalls in den Bausteinordner der Bibliothek.
- Die beim Übersetzen von Muster-SFCs erzeugten Bausteine dürfen Sie nicht kopieren.

5. Muster-SFCs:
Wenn noch nicht vorhanden, legen Sie einen Planordner an.
Kopieren Sie die Muster-SFCs aus dem Planordner Ihres Projekts in den Planordner der Bibliothek.

Hinweis:

Die zu den Muster-SFCs zugehörigen Bausteine werden dabei ebenfalls mitkopiert und im Bausteinordner abgelegt.

4.3 So erstellen Sie ein Setup

Vorgehen

1. Wenn Sie Ihre Bibliothek per Setup auf dem Zielrechner installieren wollen, entwerfen Sie mit dem Setup-Erstellungswerkzeug ein Installationskript, das die folgenden Aktionen ausführt:
 - Kopieren der Bausteinbibliothek in das Unterverzeichnis **S7LIBS** des STEP 7 Verzeichnisses
 - Aufruf des Programms **S7BIN\S7ALIBXX.EXE** im STEP 7-Verzeichnis, um die neue Bibliothek beim SIMATIC Manager bekannt zu machen
 - Kopieren der Hilfedatei (HLP- und CNT-Datei) in das Unterverzeichnis des STEP 7-Verzeichnisses, in das die Bausteinbibliothek kopiert worden ist (z. B. das Unterverzeichnis **S7LIBS\MYLIB**)
 - Aufruf der zur Hilfedatei gehörenden Registrierungsdatei
 - Kopieren der Prototypbilder in das Unterverzeichnis **options\pdl\FaceplateDesigner_V6** des WinCC-Verzeichnisses
 - Kopieren der Skripte in ein beliebiges Unterverzeichnis im Unterverzeichnis **aplib** des WinCC-Verzeichnisses
Sinnvollerweise sollte dieses Verzeichnis denselben Namen haben wie das, in das die Bausteinbibliothek kopiert worden ist (z. B. das Unterverzeichnis **options\pdl\mylib**).
 - Einrichten einer Deinstallationsmöglichkeit
2. Beachten Sie, dass die Bausteinbibliothek und die Online-Hilfe nur installiert werden können, wenn auf dem Zielrechner STEP 7 vorhanden ist. Die Prototypbilder können nur in ein Unterverzeichnis von WinCC installiert werden. Sehen Sie deshalb im Installationsdialog Abfragen auf das Vorhandensein von STEP 7 und WinCC vor.
3. Dazu können Sie in der Registrierung Folgendes abfragen:
 - Im Schlüssel HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7 können Sie den Namen STEP7_VERSION auf den Wert der gewünschten Version (z. B. "5.2") abfragen.
 - Im Schlüssel HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\WinCC\Setup können Sie den Namen der Version (z. B. V6.0 SP1) abfragen.

A Beispiele: Quellcode der Bausteine MEAS_MON, MOTOR und VALVE

A.1 MEAS_MON

```
// @(#) $Header:: L:/PCS7/PCS7LibV51.SW/TechBlocks/VCS/Algorithm/meas_mon.csv$
//Copyright (C) Siemens AG 1995. All Rights Reserved. Confidential

// PCS 7 Library Vx.x
// Function: Meas.value monitoring block
// Label Version 3.0
// Macro Version :V0.92
FUNCTION_BLOCK "MEAS_MON"
TITLE = 'Meas.value monitoring block'
//
{
    S7_tasklist:=      'OB100';
    S7_alarm_ui:=     '1';
    S7_m_c:=          'true'
}
AUTHOR:              TECHN61
NAME:                MEAS_MON
VERSION:             '3.0'
FAMILY:              CONTROL
KNOW_HOW_PROTECT

VAR_INPUT
OOS {S7_visible:='false';
    S7_link:='false';
    S7_m_c:='true';
    S7_string_0:='In Service';
    S7_string_1:='Out of Service'} :   BOOL := 0; // 1= Out of Service
M_SUP_AH {S7_visible:='false';
    S7_link:='false';
    S7_m_c:='true';
    S7_string_0:='Suppress HH=No';
    S7_string_1:='Suppress HH=Yes'} :   BOOL := 0; // 1=Suppress HH Alarm
M_SUP_AL {S7_visible:='false';
    S7_link:='false';
    S7_m_c:='true';
    S7_string_0:='Suppress LL=No';
    S7_string_1:='Suppress LL=Yes'} :   BOOL := 0; // 1=Suppress LL Alarm
M_SUP_WH {S7_visible:='false';
    S7_link:='false';
    S7_m_c:='true';
    S7_string_0:='Suppress H=No';
    S7_string_1:='Suppress H=Yes'} :   BOOL := 0; // 1=Suppress H Alarm (Warning)
M_SUP_WL {S7_visible:='false';
    S7_link:='false';
    S7_m_c:='true';
    S7_string_0:='Suppress L=No';
    S7_string_1:='Suppress L=Yes'} :   BOOL := 0; // 1=Suppress L Alarm (Warning)
CSF {S7_dynamic:='true'} :           BOOL := 0; // Control System Fault 1=External Error
MSG_LOCK {S7_visible:='false';
    S7_dynamic:='true';
    S7_m_c:='true'} :                 BOOL := 0; // Enable 1=Messages locked
MO_PVHR {S7_visible:='false';
    S7_m_c:='true';
    S7_shortcut:='Bar UL';
    S7_unit:='' } : REAL := 110; // High Limit Bar Range
MO_PVLR {S7_visible:='false';
    S7_m_c:='true';
    S7_shortcut:='Bar LL';
    S7_unit:='' } : REAL := -10; // Low Limit Bar Range
```

```

USTATUS {S7_visible:='false'} :   WORD := 0; // User Status Bits
U      {S7_gc:='true';
      S7_dynamic:='true';
      S7_m_c:='true';
      S7_shortcut:='PV';
      S7_unit:='' } : REAL := 0; // Analog Input (Measured Value)
QC_U : BYTE := 16#80; // Quality Code for Input U
U_AH   {S7_link:='false';
      S7_edit:='para';
      S7_m_c:='true';
      S7_shortcut:='HH alarm';
      S7_unit:='' } : REAL := 100; // HH Alarm Limit
U_WH   {S7_link:='false';
      S7_edit:='para';
      S7_m_c:='true';
      S7_shortcut:='H alarm';
      S7_unit:='' } : REAL := 95; // H Alarm Limit (Warning)
U_WL   {S7_link:='false';
      S7_edit:='para';
      S7_m_c:='true';
      S7_shortcut:='L alarm';
      S7_unit:='' } : REAL := -3; // L Alarm Limit (Warning)
U_AL   {S7_link:='false';
      S7_edit:='para';
      S7_m_c:='true';
      S7_shortcut:='LL alarm';
      S7_unit:='' } : REAL := -5; // LL Alarm Limit
HYS {S7_link:='false';
      S7_m_c:='true';
      S7_shortcut:='Hysteresis';
      S7_unit:='' } : REAL := 5; // Hysteresis of Analog Input
MSG_EVID {S7_visible:='false';
      S7_link:='false';
      S7_param :='false';
      S7_server:='alarm archiv';S7_a_type:='alarm_8p'} : DWORD := 0; // Message ID
BA_EN   {S7_visible:='false';
      S7_m_c:='true'} : BOOL := 0; // Batch Enable
OCCUPIED {S7_visible:='false';
      S7_m_c:='true'} : BOOL := 0; // Occupied by Batch
BA_ID   {S7_visible:='false';
      S7_m_c:='true'} : DWORD := 0; // Batch ID
BA_NA   {S7_visible:='false';
      S7_m_c:='true'} : STRING[32] := ''; // Batch Name
STEP_NO {S7_visible:='false';
      S7_m_c:='true'} : DWORD := 0; // Batch Step Number
RUNUPCYC {S7_visible:='false';
      S7_link:='false'} : INT := 3; // Number of Run Up Cycles
END_VAR

VAR_IN_OUT
AUX_PR05 {S7_visible:='false'} : ANY; // Auxiliary Value 5
AUX_PR06 {S7_visible:='false'} : ANY; // Auxiliary Value 6
AUX_PR07 {S7_visible:='false'} : ANY; // Auxiliary Value 7
AUX_PR08 {S7_visible:='false'} : ANY; // Auxiliary Value 8
AUX_PR09 {S7_visible:='false'} : ANY; // Auxiliary Value 9
AUX_PR10 {S7_visible:='false'} : ANY; // Auxiliary Value 10
END_VAR

VAR_OUTPUT
QERR {S7_visible:='false';
      S7_m_c:='true'} : BOOL := 1; // 1=Error
QH_ALM {S7_dynamic:='true'} : BOOL := 0; // 1=HH-Alarm active
QL_ALM {S7_dynamic:='true'} : BOOL := 0; // 1=LL Alarm active
QH_WRN {S7_dynamic:='true'} : BOOL := 0; // 1=H Alarm active (Warning)
QL_WRN {S7_dynamic:='true'} : BOOL := 0; // 1=L Alarm active (Warning)
QMSG_ERR {S7_visible:='false';
      S7_dynamic:='true'} : BOOL := 0; // 1=Message ERROR
QMSG_SUP {S7_visible:='false';
      S7_dynamic:='true';
      S7_m_c:='true'} : BOOL := 0; // 1=Message Suppression Active
MSG_STAT {S7_visible:='false';
      S7_dynamic:='true'} : WORD := 0; // Message: STATUS output
MSG_ACK {S7_visible:='false';
      S7_dynamic:='true'} : WORD := 0; // Message: ACK_STATE output
VSTATUS {S7_visible:='false';
      S7_m_c:='true'} : DWORD := 0; // Status word
END_VAR

```

```

CONST
C_OOS := 0; // 1= Out of Service
C_M_SUP_AH := 0; // 1=Suppress HH Alarm
C_M_SUP_AL := 0; // 1=Suppress LL Alarm
C_M_SUP_WH := 0; // 1=Suppress H Alarm (Warning)
C_M_SUP_WL := 0; // 1=Suppress L Alarm (Warning)
C_CSF := 0; // Control System Fault 1=External Error
C_MSG_LOCK := 0; // Enable 1=Messages locked
C_MO_PVHR := 110; // High Limit Bar Range
C_MO_PVLR := -10; // Low Limit Bar Range
C_USTATUS := 0; // User Status Bits
C_U := 0; // Analog Input (Measured Value)
C_QC_U := 16#80; // Quality Code for Input U
C_U_AH := 100; // HH Alarm Limit
C_U_WH := 95; // H Alarm Limit (Warning)
C_U_WL := -3; // L Alarm Limit (Warning)
C_U_AL := -5; // LL Alarm Limit
C_HYS := 5; // Hysteresis of Analog Input
C_AUX_PRO5 := 0; // Auxiliary Value 5
C_AUX_PRO6 := 0; // Auxiliary Value 6
C_AUX_PRO7 := 0; // Auxiliary Value 7
C_AUX_PRO8 := 0; // Auxiliary Value 8
C_AUX_PRO9 := 0; // Auxiliary Value 9
C_AUX_PRO10 := 0; // Auxiliary Value 10
C_MSG_EVID := 0; // Message ID
C_BA_EN := 0; // Batch Enable
C_OCCUPIED := 0; // Occupied by Batch
C_BA_ID := 0; // Batch ID
C_BA_NA := ''; // Batch Name
C_STEP_NO := 0; // Batch Step Number
C_RUNUPCYC := 3; // Lag: Number of Run Up Cycles
C_QERR := 1; // 1=Error
C_QH_ALM := 0; // 1=HH-alarm Active
C_QL_ALM := 0; // 1=LL Alarm Active
C_QH_WRN := 0; // 1=H Alarm active (Warning)
C_QL_WRN := 0; // 1=L Alarm Active (Warning)
C_QMSG_ERR := 0; // 1=Message ERROR
C_QMSG_SUP := 0; // 1=Message Suppression Active
C_MSG_STAT := 0; // Message: STATUS Output
C_MSG_ACK := 0; // Message: ACK_STATE output
C_VSTATUS := 0; // Status word
END_CONST

// Static Variables
VAR
siRUNUPCNT: int := 0; // Counter for RUNUPCYC editing
sb_SIG_1: bool := FALSE; //Merker ALARM_8P Signal 1
sb_SIG_2: bool := FALSE; //Merker ALARM_8P Signal 2
sb_SIG_3: bool := FALSE; //Merker ALARM_8P Signal 3
sb_SIG_4: bool := FALSE; //Merker ALARM_8P Signal 4
sb_SIG_5: bool := FALSE; //Merker ALARM_8P Signal 5
ALARM_8P_1: ALARM_8P; // Multiple instance ALARM_8P
siBA_ID: dword := 0; // Old value BA_ID
sbyBA_NA: array[1..32] of byte := 32(0);
VSTATUS_LOC : DWORD :=16#0; // Local static variable, in which the output VSTATUS
// is copied.
STEP_NO_LOC : DWORD; // Local variable, in which the input STEP_NO is
// saved.
PV_IN_LOC : REAL; // Local variable, in which the input PV_IN is saved.
dwDUMMY: DWORD := 0; // Stand-by
BA_ID_LOC : DWORD; // Local static variable, in which the input BA_ID
// is copied.
U_LOC : REAL; // Local static variable, in which the input U is copied, because of
// message in ALARM_8P

// Variables for the status word "VSTATUS" and "USTATUS"
VSTATUS_STR AT VSTATUS_LOC : STRUCT
    USTATUS : WORD;
    VSTATUS_LOW_BIT_8 : BOOL;
    VSTATUS_LOW_BIT_9 : BOOL;
    VSTATUS_LOW_BIT_10 : BOOL;
    VSTATUS_LOW_BIT_11 : BOOL;
    VSTATUS_LOW_BIT_12 : BOOL;
    VSTATUS_LOW_BIT_13 : BOOL;
    VSTATUS_LOW_BIT_14 : BOOL;
    VSTATUS_LOW_BIT_15 : BOOL;
    VSTATUS_LOW_BIT_0 : BOOL;
    VSTATUS_LOW_BIT_1 : BOOL;

```

```

                                VSTATUS_LOW_BIT_2 : BOOL;
                                VSTATUS_LOW_BIT_3 : BOOL;
                                VSTATUS_LOW_BIT_4 : BOOL;
                                VSTATUS_LOW_BIT_5 : BOOL;
                                VSTATUS_LOW_BIT_6 : BOOL;
                                VSTATUS_LOW_BIT_7 : BOOL;
                                END_STRUCT;
END_VAR

// Temporary Variables
VAR TEMP
pbALARM: BOOL;           // Call up ALARM_8P
pbM_SUP: BOOL;          // Message suppression
pb_SIG_1,pb_SIG_2,pb_SIG_3,pb_SIG_4:BOOL;
TOP_SI:   STRUCT
  EV_CLASS:  BYTE;
  EV_NUM:    BYTE;
  PRIORITY:  BYTE;
  NUM:       BYTE;
  TYP2_3:    BYTE;
  TYP1:      BYTE;
  ZI1:       WORD;
  ZI2_3:     DWORD;
END_STRUCT;
START_UP_SI: STRUCT
  EV_CLASS:  BYTE;
  EV_NUM:    BYTE;
  PRIORITY:  BYTE;
  NUM:       BYTE;
  TYP2_3:    BYTE;
  TYP1:      BYTE;
  ZI1:       WORD;
  ZI2_3:     DWORD;
END_STRUCT;
ERR : INT; // Error at startup
END_VAR

BEGIN
  // Write VSTATUS, STEP_NO resave as STEP_NO_LOC and
  // BA_ID resave as BA_ID_LOC.
  // Supply of VSTATUS output variables with the inputs.
  VSTATUS_STR.VSTATUS_LOW_BIT_0 := OCCUPIED;
  VSTATUS_STR.VSTATUS_LOW_BIT_1 := BA_EN;
  VSTATUS_STR.VSTATUS_LOW_BIT_2 := MSG_LOCK;
  // Supply of VSTATUS output variables (HIGH BYTE) with the values,
  // which come from the user via the input variable USTATUS (from external).
  VSTATUS_STR.USTATUS := USTATUS;
  STEP_NO_LOC := STEP_NO; // Resave, due to output of STEP_NO_LOC in ALARM_8P
  BA_ID_LOC := BA_ID;    // Resave, due to output of BA_ID_LOC in ALARM_8P
  U_LOC := U;           // Resave, due to output of U_LOC in ALARM_8P
  ERR := RD_SINFO (TOP_SI := TOP_SI, START_UP_SI := START_UP_SI);
  // Read out start info
  pbM_SUP := MSG_LOCK;
  IF TOP_SI.NUM = 100 THEN // When startup
    siRUNUPCNT := RUNUPCYC; // Saving the value of the RUNUPCYC input
    // Initialization outputs
    QMSG_ERR := C_QMSG_ERR;
    QMSG_SUP := C_QMSG_SUP;
    MSG_STAT := C_MSG_STAT;
    MSG_ACK := C_MSG_ACK;
    QH_WRN := C_QH_WRN;
    QL_WRN := C_QL_WRN;
    QH_ALM := C_QH_ALM;
    QL_ALM := C_QL_ALM;
    // pbALARM := NOT OOS; // Initialization first call ALARM_8P
    pb_SIG_1:= QH_ALM; // Alarm high
    pb_SIG_2:= QH_WRN; // Warning high
    pb_SIG_3:= QL_WRN; // Warning low
    pb_SIG_4:= QL_ALM; // Alarm low
    pbALARM :=TRUE; // Initialization ALARM
  ELSE;
    // LIMITS_P.1 für Alarmprüfung (optimiert)
    IF (U <= U_AL) THEN // Low limit responded
      QL_ALM := TRUE;
    ELSE;
      IF (U >= (U_AL+HYS)) THEN // Reset low limit responded
        QL_ALM := FALSE;
      ELSE; // QL_ALM remains unchanged

```

```

        END_IF;
    END_IF;
    IF (U >= U_AH) THEN // High limit responded
        QH_ALM := TRUE;
    ELSE;
        IF (U <= (U_AH-HYS)) THEN // Reset high limit responded
            QH_ALM := FALSE;
        ELSE; // QH_ALM remains unchanged
        END_IF;
    END_IF;
    // LIMITS P.2 for Warning check (optimized)
    IF (U <= U_WL) THEN // Low limit responded
        QL_WRN := TRUE;
    ELSE;
        IF (U >= (U_WL+HYS)) THEN // Reset low limit responded
            QL_WRN := FALSE;
        ELSE; // QL_WRN remains unchanged
        END_IF;
    END_IF;
    IF (U >= U_WH) THEN // High limit responded
        QH_WRN := TRUE;
    ELSE;
        IF (U <= (U_WH-HYS)) THEN // Reset high limit responded
            QH_WRN := FALSE;
        ELSE; // QH_WRN remains unchanged
        END_IF;
    END_IF;
    IF sirUNUPCNT = 0 THEN // Initialize alarms
        IF M_SUP_AH OR MSG_LOCK THEN
            pb_SIG_1:= 0; // Report possible outgoing
        ELSE;
            pb_SIG_1:= QH_ALM; // Alarm high
        END_IF;
        IF M_SUP_WH OR MSG_LOCK THEN
            pb_SIG_2:= 0; // Report possible outgoing
        ELSE;
            pb_SIG_2:= QH_WRN; // Warning high
        END_IF;
        IF M_SUP_WL OR MSG_LOCK THEN
            pb_SIG_3:= 0; // Report possible outgoing
        ELSE;
            pb_SIG_3:= QL_WRN; // Warning low
        END_IF;
        IF M_SUP_AL OR MSG_LOCK THEN
            pb_SIG_4:= 0; // Report possible outgoing
        ELSE;
            pb_SIG_4:= QL_ALM; // Alarm low
        END_IF;
        // pbALARM := (sb_SIG_1 <> pb_SIG_1) OR (sb_SIG_2 <> pb_SIG_2)
        // OR (sb_SIG_3 <> pb_SIG_3) OR (sb_SIG_4 <> pb_SIG_4)
        // OR (sb_SIG_5 <> CSF);
        pbALARM :=TRUE; // Initialization ALARM
    ELSE;
        pbALARM :=FALSE; // Initialization no ALARM
        pbM_SUP := TRUE;
        sirUNUPCNT := sirUNUPCNT - 1;
    END_IF;
END_IF;
IF pbALARM THEN // Call up ALARM_8P
    IF siBA_ID <> BA_ID_LOC THEN
        // STRING variables may not be interconnected to ALARM8_P as auxiliary
        //process values, therefore transferred in ARRAY OF BYTE.
        FOR ERR := 1 TO 32
            DO
                sbyBA_NA[ERR] := 0; // Delete array as default
            END FOR;
            ERR := BLKMOV (SRCBLK:= BA_NA,DSTBLK:=sbyBA_NA);
            siBA_ID := BA_ID_LOC; // Save modified BA_ID
        END_IF;
        // Call ALARM_8P with lock logic (MSG_LOCK).
        ALARM_8P_1( EN_R := TRUE, // Update the ACKL_STATE output
            ID := 16#EEEE, // PMC communication channel
            EV_ID:= MSG_EVID,
            SIG_1:= pb_SIG_1,
            SIG_2:= pb_SIG_2,
            SIG_3:= pb_SIG_3,
            SIG_4:= pb_SIG_4,
            SIG_5:= CSF,
            SIG_6:= 0,

```

```
SIG_7:= 0,
SIG_8:= 0,
SD_1 := sbyBA_NA,
SD_2 := STEP_NO_LOC,
SD_3 := BA_ID_LOC,
SD_4 := U_LOC,
SD_5 := AUX_PR05,
SD_6 := AUX_PR06,
SD_7 := AUX_PR07,
SD_8 := AUX_PR08,
SD_9 := AUX_PR09,
SD_10 := AUX_PR10);
QMSG_ERR := ALARM_8P_1.ERROR;
MSG_STAT := ALARM_8P_1.STATUS;
MSG_ACK := ALARM_8P_1.ACK_STATE;
IF (NOT QMSG_ERR) THEN // Note historical signals.
    sb_SIG_1:= pb_SIG_1;
    sb_SIG_2:= pb_SIG_2;
    sb_SIG_3:= pb_SIG_3;
    sb_SIG_4:= pb_SIG_4;
    sb_SIG_5:= CSF;
END_IF;
END_IF;
IF (MSG_STAT = 21) THEN // Block locked
    pbM_SUP := TRUE;
END_IF;
QMSG_SUP := pbM_SUP;
QERR := NOT OK; // Note negated OK-Flag result in the block.
// Power supply of the VSTATUS output variable with the outputs.
VSTATUS_STR.VSTATUS_LOW_BIT_14 := QMSG_SUP OR M_SUP_AH OR M_SUP_WH OR M_SUP_WL OR
M_SUP_AL ;
VSTATUS_STR.VSTATUS_LOW_BIT_15 := OOS;
VSTATUS := VSTATUS_LOC;
END_FUNCTION_BLOCK
```

A.2 MOTOR

```

// @(#) $Header:: L:/PCS7/PCS7LibV51.SW/TechBlocks/VCS/Algorithm/motor.csv $
//Copyright (C) Siemens AG 1995-1997. All Rights Reserved. Confidential

PCS 7 Library Vx.x
// Function: motor
// Label Version 3.0
// Macro Version :V0.92

FUNCTION_BLOCK "MOTOR"
TITLE = 'motor '
//
{
    S7_tasklist:=          'OB100';
    S7_alarm_ui:=         '1';
    S7_m_c:=              'true'
}
AUTHOR:                  TECHN61
NAME:                    MOTOR
VERSION:                 '4.0'
FAMILY:                  CONTROL
KNOW_HOW_PROTECT

VAR_INPUT
OOS {S7_visible:='false';
    S7_link:='false';
    S7_m_c:='true';
    S7_string_0:='In Service';
    S7_string_1:='Out of Service'} :   BOOL := 0; // 1= Out of Service
LOCK {S7_dynamic:='true';
    S7_m_c:='true'} :   BOOL := 0; // 1=Lock to OFF
LOCK_ON {S7_dynamic:='true';
    S7_m_c:='true'} :   BOOL := 0; // 1=Lock to ON
AUTO_ON {S7_dynamic:='true';
    S7_contact:='true'} :   BOOL := 0; // AUTO Mode:1=ON, 0=OFF
L_RESET {S7_dynamic:='true';
    S7_contact:='true'} :   BOOL := 0; // Linkable Input RESET
MSS {S7_dynamic:='true'} :   BOOL := 1; // Motor Protecting Switch: 0=Active
CSF {S7_dynamic:='true'} :   BOOL := 0; // Control System Fault 1=External Error
FB_ON {S7_qc:='true';
    S7_dynamic:='true';
    S7_m_c:='true'} :   BOOL := 0; // Feedback: 1=ON
QC_FB_ON : BYTE := 16#80; // Quality Code for FB_ON
QC_QSTART_I : BYTE := 16#80; // Quality Code for Input QSTART
ON_OP_EN {S7_visible:='false'} :   BOOL := 1; // Enable 1=Operator may input "ON"
OFFOP_EN {S7_visible:='false'} :   BOOL := 1; // Enable 1=Operator for "OFF"
MANOP_EN {S7_visible:='false'} :   BOOL := 1; // Enable: 1=Operator may input "MANUAL"
AUTOP_EN {S7_visible:='false'} :   BOOL := 1; // Enable: 1=Operator may input "AUTO"
LIOP_SEL {S7_dynamic:='true';
    S7_contact:='true'} :   BOOL := 0; // Select: 1=Linking, 0=Operator Active
AUT_L {S7_dynamic:='true';
    S7_contact:='true'} :   BOOL := 0; // Linkable Input for MANUAL/AUTO Mode
MONITOR {S7_link:='false';
    S7_m_c:='true';
    S7_string_0:='Monitoring=Off';
    S7_string_1:='Monitoring=On'} :   BOOL := 1; // Select: 1=Monitoring ON,
// 0=Monitoring OFF

TIME_MON {S7_link:='false';
    S7_edit:='para';
    S7_m_c:='true';
    S7_shortcut:='Mon. Time';
    S7_unit:='s'} :   REAL := 3; // Monitoring Time for ON [s]
SAMPLE_T {S7_visible:='false';
    S7_sampletime:='true'} :   REAL := 1; // Sample Time [s]
MSG_EVID {S7_visible:='false';
    S7_link:='false';
    S7_param:='false';
    S7_server:='alarm_archiv';S7_a_type:='alarm_8p'} :   DWORD := 0; // Message ID
BA_EN {S7_visible:='false';
    S7_m_c:='true'} :   BOOL := 0; // Batch Enable
OCCUPIED {S7_visible:='false';
    S7_m_c:='true'} :   BOOL := 0; // Occupied by Batch

```

```

BA_ID      {S7_visible:='false';
           S7_m_c:='true'} :   DWORD := 0;      // Batch ID
BA_NA      {S7_visible:='false';
           S7_m_c:='true'} :   STRING[32] := ''; // Batch Name
STEP_NO    {S7_visible:='false';
           S7_m_c:='true'} :   DWORD := 0;      // Batch Step Number
RUNUPCYC   {S7_visible:='false';
           S7_link:='false'} :   INT := 3;      // Lag: Number of Run Up Cycles
START_OFF  {S7_visible:='false'} :   BOOL := 1; // 1=Start up with Motor OFF
FAULT_OFF  {S7_visible:='false'} :   BOOL := 1; // 1=In case of Fault: Motor OFF
MSS_OFF    {S7_visible:='false'} :   BOOL := 1; // 1=In case of MSS-Fault: Motor OFF
USTATUS    {S7_visible:='false'} :   WORD := 0; // User STATUS Bits
END_VAR

VAR_IN_OUT
RESET      {S7_visible:='false';
           S7_link:='false';
           S7_m_c:='true';
           S7_string_0:='0';
           S7_string_1:='Error=Reset'} :   BOOL := 0; // Operator Input Error Reset
MAN_ON     {S7_visible:='false';
           S7_link:='false';
           S7_m_c:='true';
           S7_string_0:='Motor=Stop';
           S7_string_1:='Motor=Start'} :   BOOL := 0; // Operator Input: 1=ON, 0=OFF
AUT_ON_OP  {S7_visible:='false';
           S7_link:='false';
           S7_m_c:='true';
           S7_string_0:='Mode=Manual';
           S7_string_1:='Mode=Auto'} :   BOOL := 0; // Operator Input Mode 1=AUTO,
                                           // 0= MANUAL
AUX_PR04   {S7_visible:='false'} :   ANY; // Auxiliary Value 4
AUX_PR05   {S7_visible:='false'} :   ANY; // Auxiliary Value 5
AUX_PR06   {S7_visible:='false'} :   ANY; // Auxiliary Value 6
AUX_PR07   {S7_visible:='false'} :   ANY; // Auxiliary Value 7
AUX_PR08   {S7_visible:='false'} :   ANY; // Auxiliary Value 8
AUX_PR09   {S7_visible:='false'} :   ANY; // Auxiliary Value 9
AUX_PR10   {S7_visible:='false'} :   ANY; // Auxiliary Value 10
END_VAR

VAR_OUTPUT
QERR       {S7_visible:='false';
           S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 1; // 1=Error
QMSS_ST    {S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 0; // Unacknowledged Motor Protective Switch
QMON_ERR   {S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 0; // 1=Monitoring Error
QGR_ERR    {S7_dynamic:='true';
           S7_contact:='true'} :   BOOL := 0; // 1=Group Error
QOP_ERR    {S7_visible:='false';
           S7_dynamic:='true'} :   BOOL := 0; // 1=Operator Error
QRUN       {S7_dynamic:='true';
           S7_contact:='true';
           S7_m_c:='true'} :   BOOL := 0; // Status: 1=Motor running
QSTOP      {S7_dynamic:='true';
           S7_contact:='true';
           S7_m_c:='true'} :   BOOL := 0; // Status: 1=Motor STOP
QSTART     {S7_qc:='true';
           S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 0; // Control Output 1=START Active
QC_QSTART  : BYTE := 16#80; // Quality Code for Output QSTART
QON_OP     {S7_visible:='false';
           S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 0; // Status: 1=Operator enabled for "ON"
QOFF_OP    {S7_visible:='false';
           S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 0; // Status: 1=Operator enabled for "OFF"
QMAN_AUT   {S7_dynamic:='true';
           S7_contact:='true';
           S7_m_c:='true'} :   BOOL := 0; // 1=AUTO, 0=MANUAL Mode
QMANOP     {S7_visible:='false';
           S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 0; // Status: 1=Oper. ena. for "MANUAL" Mode

```

```

QAUTOP    {S7_visible:='false';
           S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 0; // Status: 1=Operator enabled for "AUTO"
QMSG_ERR  {S7_visible:='false';
           S7_dynamic:='true'} :   BOOL := 0; // 1=Message ERROR
QMSG_SUP  {S7_visible:='false';
           S7_dynamic:='true';
           S7_m_c:='true'} :   BOOL := 0; // 1=Message Suppression Active
MSG_STAT  {S7_visible:='false';
           S7_dynamic:='true'} :   WORD := 0; // Message: STATUS Output
MSG_ACK   {S7_visible:='false';
           S7_dynamic:='true'} :   WORD := 0; // Message: ACK_STATE output
VSTATUS  {S7_visible:='false';
           S7_m_c:='true'} :   DWORD := 0; // Status word
END_VAR

CONST
C_OOS := 0; // 1= Out of Service
C_LOCK := 0; // 1=Lock to OFF
C_LOCK_ON := 0; // 1=Lock to ON
C_AUTO_ON := 0; // AUTO Mode:1=ON, 0=Off
C_RESET := 0; // Operator Input Error Reset
C_L_RESET := 0; // Linkable Input RESET
C_MSS := 1; // Motor Protecting Switch: 0=Active
C_CSF := 0; // Control System Fault 1=External Error
C_FB_ON := 0; // Feedback: 1=ON
C_QC_FB_ON := 16#80; // Quality Code for FB_ON
C_QC_QSTART_I := 16#80; // Quality Code for Input QSTART
C_MAN_ON := 0; // Operator Input: 1=ON, 0=OFF
C_ON_OP_EN := 1; // Enable 1=Operator may input ON
C_OFFOP_EN := 1; // Enable 1=Operator for "OFF"
C_AUT_ON_OP := 0; // Operator Input Mode 1=AUTO, 0= MANUAL
C_MANOP_EN := 1; // Enable: 1=Operator may input MANUAL
C_AUTOP_EN := 1; // Enable: 1=Operator may input AUTO
C_LIOP_SEL := 0; // Select: 1=Linking, 0=Operator Active
C_AUT_L := 0; // Linkable Input for MANUAL/AUTO Mode
C_MONITOR := 1; // Select: 1=Monitoring ON, 0=Monitoring OFF
C_TIME_MON := 3; // Monitoring Time for ON [s]
C_SAMPLE_T := 1; // Sample Time [s]
C_AUX_PR04 := 0; // Auxiliary Value 4
C_AUX_PR05 := 0; // Auxiliary Value 5
C_AUX_PR06 := 0; // Auxiliary Value 6
C_AUX_PR07 := 0; // Auxiliary Value 7
C_AUX_PR08 := 0; // Auxiliary Value 8
C_AUX_PR09 := 0; // Auxiliary Value 9
C_AUX_PR10 := 0; // Auxiliary Value 10
C_MSG_EVID := 0; // Message ID
C_BA_EN := 0; // Batch Enable
C_OCCUPIED := 0; // Occupied by Batch
C_BA_ID := 0; // Batch ID
C_BA_NA := ''; // Batch Name
C_STEP_NO := 0; // Batch Step Number
C_RUNUPCYC := 3; // Lag: Number of Run Up Cycles
C_START_OFF := 1; // 1=Start up with Motor OFF
C_FAULT_OFF := 1; // 1=In case of fault: Motor OFF
C_MSS_OFF := 1; // 1=In case of MSS fault: Motor OFF
C_USTATUS := 0; // User STATUS Bits
C_QERR := 1; // 1=Error
C_QMSS_ST := 0; // Unacknowledged Motor Protective Switch
C_QMON_ERR := 0; // 1=Monitoring Error
C_QGR_ERR := 0; // 1=Group Error
C_QOP_ERR := 0; // 1=Operator Error
C_QRUN := 0; // Status: 1=Motor running
C_QSTOP := 0; // Status: 1=Motor STOP
C_QSTART := 0; // Control Output 1=START Active
C_QC_QSTART := 16#80; // Quality Code for Output QSTART
C_QON_OP := 0; // Status: 1=Operator enabled for "ON"
C_QOFF_OP := 0; // Status: 1=Operator enabled for "OFF"
C_QMAN_AUT := 0; // 1=AUTO, 0=MANUAL Mode
C_QMANOP := 0; // Status: 1=Oper. enabled for "MANUAL" Mode
C_QAUTOP := 0; // Status: 1=Operator enabled for "AUTO" Mode
C_QMSG_ERR := 0; // 1=Message ERROR
C_QMSG_SUP := 0; // 1=Message Suppression Active
C_MSG_STAT := 0; // Message: STATUS Output
C_MSG_ACK := 0; // Message: ACK_STATE output
C_VSTATUS := 0; // Status word
END_CONST

```

```

// Static Variables
VAR
sbI_OD1:      BOOL := 0;      // Flag of old operating value for OP_D.1
sbI_OD2:      BOOL := 0;      // Flag of old operating value for OP_D.2
sbQ_OD1:      BOOL := 0;      // Binary output for OP_D.1
sbALT_LINK_I_OT1:  BOOL := 0; // Old value of interconnectable input for OP_TRIG.1
sbALT_QSTART:  BOOL := 0;     // Historical process data for QSTART
sb_SIG_1:     BOOL := FALSE;  // ALARM_8P signal 1 flag
sb_SIG_2:     BOOL := FALSE;  // ALARM_8P signal 2 flag
sb_SIG_3:     BOOL := FALSE;  // ALARM_8P signal 3 flag
srAktZeit:    REAL := 0;      // Time passed
sirUNUPCNT:   INT  := 0;      // Counter for RUNUPCYC editing

//----- ALARM_8P.1 -----
ALARM_8P_1:   ALARM_8P;      // Multiple instanced ALARM_8P
dwDUMMY:     DWORD := 0;     // Stand-by
siBA_ID:     DWORD := 0;     // Old value BA_ID
sbyBA_NA:    ARRAY[1..32] OF BYTE := 32(0);

VSTATUS_LOC : DWORD :=16#0;  // Local variable, into which the VSTATUS output is copied
STEP_NO_LOC : DWORD;        // Local variable, into which the STEP_NO input is copied
BA_ID_LOC   : DWORD;        // Local variable, into which the BA_ID input is copied

// Variables for STATUSWORT "VSTATUS" and "USTATUS"
VSTATUS_STR AT VSTATUS_LOC : STRUCT
                                USTATUS : WORD;
                                VSTATUS_LOW_BIT_8 : BOOL;
                                VSTATUS_LOW_BIT_9 : BOOL;
                                VSTATUS_LOW_BIT_10 : BOOL;
                                VSTATUS_LOW_BIT_11 : BOOL;
                                VSTATUS_LOW_BIT_12 : BOOL;
                                VSTATUS_LOW_BIT_13 : BOOL;
                                VSTATUS_LOW_BIT_14 : BOOL;
                                VSTATUS_LOW_BIT_15 : BOOL;
                                VSTATUS_LOW_BIT_0 : BOOL;
                                VSTATUS_LOW_BIT_1 : BOOL;
                                VSTATUS_LOW_BIT_2 : BOOL;
                                VSTATUS_LOW_BIT_3 : BOOL;
                                VSTATUS_LOW_BIT_4 : BOOL;
                                VSTATUS_LOW_BIT_5 : BOOL;
                                VSTATUS_LOW_BIT_6 : BOOL;
                                VSTATUS_LOW_BIT_7 : BOOL;
                                END_STRUCT;
END_VAR

// Temporary variables
VAR_TEMP
TOP_SI:      STRUCT
EV_CLASS:   BYTE;
EV_NUM:     BYTE;
PRIORITY:   BYTE;
NUM:        BYTE;
TYP2_3:     BYTE;
TYP1:       BYTE;
ZI1:        WORD;
ZI2_3:      DWORD;
END_STRUCT;
START_UP_SI: STRUCT
EV_CLASS:   BYTE;
EV_NUM:     BYTE;
PRIORITY:   BYTE;
NUM:        BYTE;
TYP2_3:     BYTE;
TYP1:       BYTE;
ZI1:        WORD;
ZI2_3:      DWORD;
END_STRUCT;
ERR :      INT; // Startup error
pbLINK_ON_OD1:  BOOL; // Operating/Interconnection flag for OP_D.1.
pbLINK_I_OD1:  BOOL; // Interconnectable input for OP_D.1.
pbQOP_ERR:     BOOL; // Operating error pointer for OP_D.1.
pbLINK_I_OT1:  BOOL; // Interconnectable input for OP_TRIG.1.
pbQ_OT1:       BOOL; // Binary output for OP_TRIG.1.
pbALARM:       BOOL; // Call up ALARM_8P
pbM_SUP:       BOOL; // Message suppression
END_VAR

```

```

BEGIN
  // Supply of the VSTATUS, resave STEP_NO as STEP_NO_LOC,
  // Resave BA_ID as BA_ID_LOC
  // Supply of the VSTATUS output variable with the inputs.
  VSTATUS_STR.VSTATUS_LOW_BIT_0 := OCCUPIED;
  VSTATUS_STR.VSTATUS_LOW_BIT_1 := BA_EN;
  // Supply of the VSTATUS output variable (HIGH BYTE) with the values
  // that come from the user via the input variable USTATUS (from external).
  VSTATUS_STR.USTATUS := USTATUS;
  // Resave STEP_NO as STEP_NO_LOC and resave BA_ID as BA_ID_LOC
  STEP_NO_LOC := STEP_NO; // Resave, due to output of STEP_NO_LOC in ALARM_8P
  BA_ID_LOC := BA_ID; // Resave, due to output of BA_ID_LOC in ALARM_8P
  ERR := RD_SINFO (TOP_SI := TOP_SI, START_UP_SI := START_UP_SI); // Read out
  // start info

  // pbM_SUP := FALSE;
  IF (TOP_SI.NUM = 100) AND START_OFF THEN
    siRUNUPCNT := RUNUPCYC; // Save the value of the RUNUPCYC input
    // Initial states
    // Outputs to be set
    RESET := C_RESET;
    MAN_ON := C_MAN_ON;
    AUT_ON_OP := C_AUT_ON_OP;
    QERR := C_QERR;
    QMSS_ST := C_QMSS_ST;
    QMON_ERR := C_QMON_ERR;
    QGR_ERR := C_QGR_ERR;
    QOP_ERR := C_QOP_ERR;
    QRUN := C_QRUN;
    QSTOP := C_QSTOP;
    QSTART := C_QSTART;
    QC_QSTART := QC_QSTART_I; // Quality output described with value of input
    QON_OP := C_QON_OP;
    QOFF_OP := C_QOFF_OP;
    QMAN_AUT := C_QMAN_AUT;
    QMANOP := C_QMANOP;
    QAUTOP := C_QAUTOP;
    QMSG_ERR := C_QMSG_ERR;
    // QMSG_SUP := C_QMSG_SUP;

    MSG_STAT := C_MSG_STAT;
    MSG_ACK := C_MSG_ACK;
    // Static variables to be set
    sbI_OD1 := C_MAN_ON; // Reset flag of old operating value for OP_D.1.
    // to zero
    sbI_OD2 := C_AUT_ON_OP; // Reset flag of old operating value for OP_D.2.
    // to zero
    sbQ_OD1 := 0; // Has to be set for incorrect operator
    sbALT_LINK_I_OT1 := L_RESET; // Previous value set by LINK_I
    sbALT_QSTART := C_QSTART; // Set previous STOP-Modi
    srAktZeit := 0; // Set acttime to NULL (zero)
    pbALARM := TRUE; // Initialization ALARM_8P
  ELSE;
    // MOTOR Algorithm
    // OP_D.2 Select manual or automatic mode
    pbQOP_ERR := FALSE; // Is only set to 1 in fault scenarios
    IF LIOP_SEL THEN
      // Interconnection
      IF AUT_ON_OP = sbI_OD2 THEN
        // No operation occurs
      ELSE;
        pbQOP_ERR := TRUE; // Prohibited operator
      END_IF;
      AUT_ON_OP := AUT_L;
      // Write AUT_L back to operator input
      QMAN_AUT := AUT_L; // Interconnection
      QMANOP := FALSE; // No operator allowed
      QAUTOP := FALSE;
    ELSE;
      // Operator
      IF (sbI_OD2 <> AUT_ON_OP) AND
        NOT ((AUT_ON_OP AND AUTOP_EN) OR (NOT(AUT_ON_OP) AND MANOP_EN)) THEN
        // Prohibited operator
        pbQOP_ERR := TRUE;
        AUT_ON_OP := sbI_OD2;
      ELSE;
        // No or allowed operator
        QMAN_AUT := AUT_ON_OP;
      END_IF;
      QMANOP := MANOP_EN;
    END_IF;
  END_IF;

```

```

// Copy the inputs to the outputs
QAUTOP := AUTOP_EN;
END_IF;
sbI_OD2 := AUT_ON_OP; // Note old operating values
// OP TRIG.1.
pbLINK_I_OT1 := L_RESET AND MSS; // Link_I for OP_TRIG.1.( RESET )
// Activating RESET
pbQ_OT1 := FALSE; // Reset Q
IF ( pbLINK_I_OT1 AND ( NOT sbALT_LINK_I_OT1 ) ) THEN
    pbQ_OT1 := TRUE; // Set pbQ_OT1
ELSE;
    IF RESET THEN
        pbQ_OT1 := TRUE; // Set pbQ_OT1
    END_IF;
END_IF;
sbALT_LINK_I_OT1 := pbLINK_I_OT1; // Save previous values
RESET := FALSE;
// Motor protector circuit-breaker
// Establish motor protector circuit-breaker
QMSS_ST := QMSS_ST AND (NOT pbQ_OT1) OR (NOT MSS);
// Watchdog - 1 -
IF pbQ_OT1 THEN // If RESET came in this cycle,
// The monitor error has to be corrected
    QMON_ERR := 0;
END_IF;
// Inputs for OP_D.1.(manual mode)
pbLINK_ON_OD1 := LOCK OR LOCK_ON OR QMAN_AUT OR (QMSS_ST AND MSS_OFF)
OR (QMON_ERR AND FAULT_OFF);
pbLINK_I_OD1 := (LOCK_ON OR (AUTO_ON AND (NOT(QMON_ERR AND FAULT_OFF))))
AND (NOT LOCK) AND (NOT(QMSS_ST AND MSS_OFF));
// OP_D.1.
// Outputs of the manual mode for control
IF pbLINK_ON_OD1 THEN
    // Interconnection
    IF MAN_ON = sbI_OD1 THEN
        ; // No operation occurred
    ELSE;
        pbQOP_ERR := TRUE;
        // Prohibited operator
    END_IF;
    MAN_ON := pbLINK_I_OD1; // Write pbLINK_I_OD1 back to operator input
    sbQ_OD1 := pbLINK_I_OD1; // Interconnection
    QON_OP := FALSE; // No operator allowed
    QOFF_OP := FALSE;
ELSE; // Operator
    IF (sbI_OD1 <> MAN_ON) AND
    NOT ((MAN_ON AND ON_OP_EN) OR (NOT(MAN_ON) AND OFFOP_EN)) THEN
        // Prohibited operator
        pbQOP_ERR := TRUE;
        MAN_ON := sbI_OD1;
    ELSE; // No or allowed operator
        sbQ_OD1 := MAN_ON;
    END_IF;
    QON_OP := ON_OP_EN; // Copy the inputs to the outputs
    QOFF_OP := OFFOP_EN;
END_IF;
sbI_OD1 := MAN_ON; //Note old operating values
QOP_ERR := pbQOP_ERR; // Operator error
// Watchdog - 2 -
// Switch the motor on or off within a configured timespan.
IF (( sbALT_QSTART <> QSTART ) AND ( NOT QMON_ERR ))
// Change through control
THEN
    IF TIME_MON < SAMPLE_T THEN
        srAktZeit := SAMPLE_T;
    ELSE;
        srAktZeit := TIME_MON;
        // Set acttime
    END_IF;
END_IF;
IF srAktZeit >= SAMPLE_T THEN
    srAktZeit := srAktZeit - SAMPLE_T; // Bring down time
END_IF;

```

```

IF MONITOR THEN
  // If the monitor is active, the change from Q_START has to be written
  // immediately to the output.
  IF FB_ON = QSTART THEN // Before the monitor time expires,
                        // same Feedback and QSTART
    srAktZeit := 0;
    IF QSTART THEN // Report the corresponding feedback status
      QRUN := 1;
    ELSE;
      QSTOP := 1;
    END_IF;
  END_IF;
END_IF;
IF srAktZeit < SAMPLE_T THEN // If the monitor time expires, does the <=
  IF MONITOR THEN // watchdog then < instead function?
    IF ( FB_ON <> QSTART ) AND ( NOT pbQ_OT1 ) THEN
      // After the monitor time, feedback and QSTART are different
      QMON_ERR := 1;
    ELSE; END_IF;
  ELSE;
    IF QSTART THEN // Set QRUN and QSTOP
      QRUN := 1;
      QSTOP := 0;
    ELSE;
      QRUN := 0;
      QSTOP := 1;
    END_IF;
  END_IF;
END_IF;
// Control signal
sbALT_QSTART := QSTART;
QGR_ERR := QMON_ERR OR QMSS_ST OR CSF;
QSTART := sbQ_OD1 AND ((NOT(QMON_ERR AND FAULT_OFF)) OR LOCK_ON);
QC_QSTART := QC_QSTART_I; // Describe quality output with value of input.
// Watchdog - 3 -
// Set QRUN and QSTOP
IF QSTART THEN // Adapt QRUN and QSTOP to the previous value of QSTART.
  QSTOP := 0;
ELSE;
  QRUN := 0;
END_IF;
IF siRUNUPCNT = 0 THEN
  // Initialize alarm
  // pbALARM := (sb_SIG_1 <> QMSS_ST)
  // OR (sb_SIG_2 <> QMON_ERR) OR (sb_SIG_3 <> CSF);
  pbALARM := TRUE; // Initialization ALARM_8P
ELSE;
  siRUNUPCNT := siRUNUPCNT - 1;
  QMSG_SUP := TRUE;
  pbALARM := FALSE; // Initialization no ALARM
END_IF;
END_IF;
IF pbALARM THEN // Call up ALARM_8P
  IF siBA_ID <> BA_ID_LOC THEN
    // STRING variables may not be interconnected to ALARM8_P as auxiliary
    // process values, therefore transferred in ARRAY OF BYTE.
    FOR ERR := 1 TO 32
      DO
        sbyBA_NA[ERR] := 0; // Delete array as default
      END_FOR;
      ERR := BLKMOV (SRCBLK:= BA_NA,DSTBLK:=sbyBA_NA);
      siBA_ID := BA_ID_LOC; // Save modified BA_ID
    END_IF;
  END_IF;

```

```

// Call up ALARM_8P with lock logic (MSG_LOCK)
ALARM_8P_1( EN_R := TRUE, // Update the ACKL_STATE output
           ID := 16#EEEE, // PMC communication channel
           EV_ID:= MSG_EVID,
           SIG_1:= QMSS_ST,
           SIG_2:= QMON_ERR,
           SIG_3:= CSF,
           SIG_4:= 0,
           SIG_5:= 0,
           SIG_6:= 0,
           SIG_7:= 0,
           SIG_8:= 0,
           SD_1 := sbyBA_NA,
           SD_2 := STEP_NO_LOC,
           SD_3 := BA_ID_LOC,
           SD_4 := AUX_PR04,
           SD_5 := AUX_PR05,
           SD_6 := AUX_PR06,
           SD_7 := AUX_PR07,
           SD_8 := AUX_PR08,
           SD_9 := AUX_PR09,
           SD_10 := AUX_PR10);
QMSG_ERR := ALARM_8P_1.ERROR;
MSG_STAT := ALARM_8P_1.STATUS;
MSG_ACK := ALARM_8P_1.ACK_STATE;
sb_SIG_1:= QMSS_ST; // Note historical signals.
sb_SIG_2:= QMON_ERR;
sb_SIG_3:= CSF;
QMSG_SUP := (MSG_STAT = 21);
END_IF;
QERR := NOT OK;
// Power supply of the VSTATUS output variable with the outputs.
VSTATUS_STR.VSTATUS_LOW_BIT_3 := QMAN_AUT;
VSTATUS_STR.VSTATUS_LOW_BIT_5 := QMSS_ST;
VSTATUS_STR.VSTATUS_LOW_BIT_7 := QMON_ERR;
VSTATUS_STR.VSTATUS_LOW_BIT_8 := LOCK;
VSTATUS_STR.VSTATUS_LOW_BIT_9 := QRUN;
VSTATUS_STR.VSTATUS_LOW_BIT_10 := QSTOP;
VSTATUS_STR.VSTATUS_LOW_BIT_14 := QMSG_SUP;
VSTATUS_STR.VSTATUS_LOW_BIT_15 := OOS;
VSTATUS := VSTATUS_LOC;
END_FUNCTION_BLOCK

```

A.3 VALVE

```

// @(#) $Header:: L:/PCS7/PCS7LibV51.SW/TechBlocks/VCS/Algorithm/valv$
//Copyright (C) Siemens AG 1995. All Rights Reserved. Confidential

(*-----*)

Order of the part blocks:

1. OP_D.2
2. OP_TRIG.1
3. Watchdog RESET Logic
4. SEL_BOOL.1
5. OP_D.1
6. Watchdog (remaining)
7. XOR
8. MESSAGE

-----*)

//PCS 7 Library Vx.x
//Funktion: Single-Drive/Dual-Feedback Valve
//Etikett-Version 3.0
//Makro-Version :V0.92

FUNCTION_BLOCK "VALVE"
TITLE = 'Single-Drive/Dual-Feedback Valve'
//
{
    S7_tasklist:=      'OB100';
    S7_alarm_ui:=     '1';
    S7_m_c:=          'true'
}
AUTHOR:              TECHN61
NAME:                VALVE
VERSION:             '3.0'
FAMILY:              CONTROL
KNOW_HOW_PROTECT

VAR_INPUT
OOS {S7_visible:='false';
    S7_link:='false';
    S7_m_c:='true';
    S7_string_0:='In Service';
    S7_string_1:='Out of Service'} :   BOOL := 0; // 1= Out of Service
V_LOCK {S7_dynamic:='true';
    S7_m_c:='true'} :   BOOL := 0; // 1=Lock to SAVE position
VL_OPEN {S7_dynamic:='true';
    S7_m_c:='true'} :   BOOL := 0; // 1=Lock to OPEN
VL_CLOSE {S7_dynamic:='true';
    S7_m_c:='true'} :   BOOL := 0; // 1=Lock to CLOSE
AUTO_OC {S7_dynamic:='true';
    S7_contact:='true'} :   BOOL := 0; // AUTO Mode:1=Open, 0=Close
SS_POS {S7_dynamic:='true';
    S7_edit:='para';
    S7_m_c:='true'} :   BOOL := 0; // Safe Position. 1=Open, 0=Close
START_SS {S7_visible:='false'} :   BOOL := 1; // 1=Start with Safe State Position
// and Manual Mode.
FAULT_SS {S7_visible:='false'} :   BOOL := 1; // 1=In Case of fault: Safe State
// Position.
L_RESET {S7_dynamic:='true';
    S7_contact:='true'} :   BOOL := 0; // Linkable Input RESET
CSF {S7_dynamic:='true'} :   BOOL := 0; // Control System Fault 1=External Error
FB_OPEN {S7_qc:='true';
    S7_dynamic:='true';
    S7_m_c:='true'} :   BOOL := 0; // Feedback: 1=OPEN
QC_FB_OPEN : BYTE := 16#80; // Quality Code for FB_OPEN
FB_CLOSE {S7_qc:='true';
    S7_dynamic:='true';
    S7_m_c:='true'} :   BOOL := 0; // Feedback: 1=CLOSE
QC_FB_CLOSE : BYTE := 16#80; // Quality Code for FB_CLOSE
QC_QCONTROL_I : BYTE := 16#80; // Quality Code for Input QCONTROL
NO_FB_OP {S7_visible:='false'} :   BOOL := 0; // 1=No Feedback OPEN present
NO_FB_CL {S7_visible:='false'} :   BOOL := 0; // 1=No Feedback CLOSE present

```

```

MONITOR {S7_link:='false';
        S7_m_c:='true';
        S7_string_0:='Monitoring=Off';
        S7_string_1:='Monitoring=On'} :    BOOL := 1; // Select: 1=Monitoring ON,
                                           // 0=Monitoring OFF
NOMON_OP {S7_visible:='false'} :    BOOL := 0; // 1=No Monitoring OPEN
NOMON_CL {S7_visible:='false'} :    BOOL := 0; // 1=No Monitoring CLOSE
OP_OP_EN {S7_visible:='false'} :    BOOL := 1; // Enable 1=Operator may input OPEN
CL_OP_EN {S7_visible:='false'} :    BOOL := 1; // Enable: 1=Operator may input CLOSE
MANOP_EN {S7_visible:='false'} :    BOOL := 1; // Enable: 1=Operator may input
                                           // MANUAL
AUTOP_EN {S7_visible:='false'} :    BOOL := 1; // Enable: 1=Operator may input AUTO
LIOP_SEL {S7_contact:='true'} :    BOOL := 0; // Select: 1=Linking,
                                           // 0=Operator Active

AUT_L {S7_dynamic:='true';
       S7_contact:='true'} :    BOOL := 0; // Linkable Input for MANUAL/AUTO Mode
TIME_MON {S7_link:='false';
         S7_edit:='para';
         S7_m_c:='true';
         S7_shortcut:='Mon. Time';
         S7_unit:='s'} :    REAL := 3; // Monitoring Time [s]
SAMPLE_T {S7_visible:='false';
         S7_sampletime:='true'} :    REAL := 1; // Sample Time [s]
MSG_EVID {S7_visible:='false';
         S7_link:='false';
         S7_param :='false';
         S7_server:='alarm archiv';S7_a_type:='alarm_8p'} :    DWORD := 0; // Message ID
BA_EN {S7_visible:='false';
       S7_m_c:='true'} :    BOOL := 0; // Batch Enable
OCCUPIED {S7_visible:='false';
         S7_m_c:='true'} :    BOOL := 0; // Occupied by Batch
BA_ID {S7_visible:='false';
       S7_m_c:='true'} :    DWORD := 0; // Batch ID
BA_NA {S7_visible:='false';
       S7_m_c:='true'} :    STRING[32] := ''; // Batch Name
STEP_NO {S7_visible:='false';
        S7_m_c:='true'} :    DWORD := 0; // Batch Step Number
RUNUPCYC {S7_visible:='false';
         S7_link:='false'} :    INT := 3; // Lag: Number of Run Up Cycles
USTATUS {S7_visible:='false'} :    WORD := 0; // User STATUS Bits
END_VAR

VAR_IN_OUT
RESET {S7_visible:='false';
      S7_link:='false';
      S7_m_c:='true';
      S7_string_0:='0';
      S7_string_1:='Error=Reset'} :    BOOL := 0; // Operator Input Error Reset
MAN_OC {S7_visible:='false';
       S7_link:='false';
       S7_m_c:='true';
       S7_string_0:='Valve=Close';
       S7_string_1:='Valve=Open'} :    BOOL := 0; // Operator Input: 1=OPEN, 0=CLOSE
AUT_ON_OP {S7_visible:='false';
          S7_link:='false';
          S7_m_c:='true';
          S7_string_0:='Mode=Manual';
          S7_string_1:='Mode=Auto'} :    BOOL := 0; // Operator Input Mode 1=AUTO,
                                           // 0= MANUAL
AUX_PR04 {S7_visible:='false'} :    ANY; // Auxiliary Value 4
AUX_PR05 {S7_visible:='false'} :    ANY; // Auxiliary Value 5
AUX_PR06 {S7_visible:='false'} :    ANY; // Auxiliary Value 6
AUX_PR07 {S7_visible:='false'} :    ANY; // Auxiliary Value 7
AUX_PR08 {S7_visible:='false'} :    ANY; // Auxiliary Value 8
AUX_PR09 {S7_visible:='false'} :    ANY; // Auxiliary Value 9
AUX_PR10 {S7_visible:='false'} :    ANY; // Auxiliary Value 10
END_VAR

VAR_OUTPUT
QERR {S7_visible:='false';
     S7_m_c:='true'} :    BOOL := 1; // 1=Error
QCONTROL {S7_qc:='true';
         S7_dynamic:='true';
         S7_m_c:='true'} :    BOOL := 0; // Control Output: 0=Standstill
QC_QCONTROL : BYTE := 16#80; // Quality Code Output for QCONTROL
QMON_ERR {S7_dynamic:='true';
         S7_m_c:='true'} :    BOOL := 0; // 1=Monitoring Error
QGR_ERR {S7_dynamic:='true';
        S7_contact:='true'} :    BOOL := 0; // 1=Group Error
    
```

```

QMAN_AUT      {S7_dynamic:='true';
               S7_contact:='true';
               S7_m_c:='true'} :   BOOL := 0;      // 1=AUTO, 0=MANUAL Mode
QOP_ERR      {S7_visible:='false';
               S7_dynamic:='true'} :   BOOL := 0; // 1=Operator Error
QOP_OP       {S7_visible:='false';
               S7_dynamic:='true';
               S7_m_c:='true'} :   BOOL := 0;      // Status: 1=Operator enabled for "OPEN"
QCL_OP       {S7_visible:='false';
               S7_dynamic:='true';
               S7_m_c:='true'} :   BOOL := 0;      // Status: 1=Operator enabled for "CLOSE"
QAUTOP      {S7_visible:='false';
               S7_dynamic:='true';
               S7_m_c:='true'} :   BOOL := 0;      // Status: 1=Operator enabled for "AUTO"
QMANOP      {S7_visible:='false';
               S7_dynamic:='true';
               S7_m_c:='true'} :   BOOL := 0;      // Status: 1=Oper. ena. for "MANUAL" Mode
QOPENING     {S7_dynamic:='true';
               S7_m_c:='true'} :   BOOL := 0;      // 1=Valve is Opening
QOPENED     {S7_dynamic:='true';
               S7_contact:='true';
               S7_m_c:='true'} :   BOOL := 0;      // 1=Valve is OPEN
QCLOSING     {S7_dynamic:='true';
               S7_m_c:='true'} :   BOOL := 0;      // 1=Valve is Closing
QCLOSED     {S7_dynamic:='true';
               S7_contact:='true';
               S7_m_c:='true'} :   BOOL := 0;      // 1=Valve is CLOSED
QMSG_ERR     {S7_visible:='false';
               S7_dynamic:='true'} :   BOOL := 0; // 1=Message ERROR
QMSG_SUP     {S7_visible:='false';
               S7_dynamic:='true';
               S7_m_c:='true'} :   BOOL := 0;      // 1=Message Suppression Active
MSG_STAT     {S7_visible:='false';
               S7_dynamic:='true'} :   WORD := 0; // Message: STATUS Output
MSG_ACK     {S7_visible:='false';
              S7_dynamic:='true'} :   WORD := 0; // Message: ACK_STATE-output
VSTATUS     {S7_visible:='false';
              S7_m_c:='true'} :   DWORD := 0;     // Status-word
END_VAR

CONST
C_OOS := 0; // 1= Out of Service
C_V_LOCK := 0; // 1=Lock to SAVE position
C_VL_OPEN := 0; // 1=Lock to OPEN
C_VL_CLOSE := 0; // 1=Lock to CLOSE
C_AUTO_OC := 0; // AUTO Mode:1=Open, 0=Close
C_SS_POS := 0; // Safe Position. 1=Open, 0=Close
C_START_SS := 1; // 1=Start with Safe State Position and Manual Mode
C_FAULT_SS := 1; // 1=In Case of Fault: Safe State Position
C_RESET := 0; // Operator Input Error Reset
C_L_RESET := 0; // Linkable Input RESET
C_CSF := 0; // Control System Fault 1=External Error
C_FB_OPEN := 0; // Feedback: 1=OPEN
C_QC_FB_OPEN := 16#80; // Quality Code for FB_OPEN
C_FB_CLOSE := 0; // Feedback: 1=CLOSE
C_QC_FB_CLOSE := 16#80; // Quality Code for FB_CLOSE
C_QC_QCONTROL_I := 16#80; // Quality Code for Input QCONTROL
C_NO_FB_OP := 0; // 1=No Feedback OPEN present
C_NO_FB_CL := 0; // 1=No Feedback CLOSE present
C_MONITOR := 1; // Select: 1=Monitoring ON, 0=Monitoring OFF
C_NOMON_OP := 0; // 1=No Monitoring OPEN
C_NOMON_CL := 0; // 1=No Monitoring CLOSE
C_MAN_OC := 0; // Operator Input: 1=OPEN, 0=CLOSE
C_OP_OP_EN := 1; // Enable 1=Operator may input OPEN
C_CL_OP_EN := 1; // Enable: 1=Operator may input CLOSE
C_AUT_ON_OP := 0; // Operator Input Mode 1=AUTO, 0= MANUAL
C_MANOP_EN := 1; // Enable: 1=Operator may input MANUAL
C_AUTOP_EN := 1; // Enable: 1=Operator may input AUTO
C_LIOP_SEL := 0; // Select: 1=Linking, 0=Operator Active
C_AUT_L := 0; // Linkable Input for MANUAL/AUTO Mode
C_TIME_MON := 3; // Monitoring Time [s]
C_SAMPLE_T := 1; // Sample Time [s]
C_AUX_PR04 := 0; // Auxiliary Value 4
C_AUX_PR05 := 0; // Auxiliary Value 5
C_AUX_PR06 := 0; // Auxiliary Value 6
C_AUX_PR07 := 0; // Auxiliary Value 7
C_AUX_PR08 := 0; // Auxiliary Value 8
C_AUX_PR09 := 0; // Auxiliary Value 9

```

```

C_AUX_PR10 := 0; // Auxiliary Value 10
C_MSG_EVID := 0; // Message ID
C_BA_EN := 0; // Batch Enable
C_OCCUPIED := 0; // Occupied by Batch
C_BA_ID := 0; // Batch ID
C_BA_NA := ''; // Batch Name
C_STEP_NO := 0; // Batch Step Number
C_RUNUPCYC := 3; // Lag: Number of Run Up Cycles
C_USTATUS := 0; // User STATUS Bits
C_QERR := 1; // 1=Error
C_QCONTROL := 0; // Control Output: 0=Standstill
C_QC_QCONTROL := 16#80; // Quality Code Output for QCONTROL
C_QMON_ERR := 0; // 1=Monitoring Error
C_QGR_ERR := 0; // 1=Group Error
C_QMAN_AUT := 0; // 1=AUTO, 0=MANUAL Mode
C_QOP_ERR := 0; // 1=Operator Error
C_QOP_OP := 0; // Status: 1=Operator enabled for "OPEN"
C_QCL_OP := 0; // Status: 1=Operator enabled for "CLOSE"
C_QAUTOP := 0; // Status: 1=Operator enabled for "AUTO"
C_QMANOP := 0; // Status: 1=Oper. ena. for "MANUAL" Mode
C_QOPENING := 0; // 1=Valve is Opening
C_QOPENED := 0; // 1=Valve is OPEN
C_QCLOSING := 0; // 1=Valve is Closing
C_QCLOSED := 0; // 1=Valve is CLOSED
C_QMSG_ERR := 0; // 1=Message ERROR
C_QMSG_SUP := 0; // 1=Message Suppression Active
C_MSG_STAT := 0; // Message: STATUS Output
C_MSG_ACK := 0; // Message: ACK_STATE-output
C_VSTATUS := 0; // Status-word
END_CONST

// Static variables
VAR
sbRESTART: BOOL := 1; // First run(Default is1)
sb_SIG_1: BOOL := FALSE; // ALARM_8P signal 1 flag
sb_SIG_2: BOOL := FALSE; // ALARM_8P signal 2 flag

//----- OP_D.2 -----
sbAUT_ON_OP: BOOL := C_AUT_ON_OP; // Flag of old operating value for(I0) (0=MAN)

//----- OP_D.1 -----
sbMAN_OC: BOOL := C_MAN_OC; // Flag of old operating value for(I0) (0=CLOSED)
sbd1_Q_OC: BOOL := FALSE; // Q Ausgang

//----- OP_TRIG.1 -----
sbl_RESET: BOOL := C_L_RESET; // Flag for historical signal L_RESET (LINK_I)

//----- Watchdog-----
sbOC: BOOL := FALSE; // Old values direction
srAKTZEIT: REAL := 0; // Current time of the on-delay
sirUNUPCNT: INT := 0; // Counter for RUNUPCYC editing

//----- ALARM_8P.1 -----
ALARM_8P_1: ALARM_8P; // Multi-instanced ALARM_8P
dwDUMMY: DWORD := 0; // Stand-by
siBA_ID: DWORD := 0; // Old value BA_ID
sbyBA_NA: ARRAY[1..32] OF BYTE := 32(0);

VSTATUS_LOC : DWORD := 16#0; // Local variable, into which the VSTATUS output is copied.
STEP_NO_LOC : DWORD; // Local variable, into which the STEP_NO input is copied.
BA_ID_LOC : DWORD; // Local variable, into which the BA_ID input is copied.

// Variables for STATUSWORT "VSTATUS" and "USTATUS"
VSTATUS_STR AT VSTATUS_LOC : STRUCT
    USTATUS : WORD;
    VSTATUS_LOW_BIT_8 : BOOL;
    VSTATUS_LOW_BIT_9 : BOOL;
    VSTATUS_LOW_BIT_10 : BOOL;
    VSTATUS_LOW_BIT_11 : BOOL;
    VSTATUS_LOW_BIT_12 : BOOL;
    VSTATUS_LOW_BIT_13 : BOOL;
    VSTATUS_LOW_BIT_14 : BOOL;
    VSTATUS_LOW_BIT_15 : BOOL;
    VSTATUS_LOW_BIT_0 : BOOL;
    VSTATUS_LOW_BIT_1 : BOOL;
    VSTATUS_LOW_BIT_2 : BOOL;
    VSTATUS_LOW_BIT_3 : BOOL;
    VSTATUS_LOW_BIT_4 : BOOL;
    VSTATUS_LOW_BIT_5 : BOOL;

```

```

                                VSTATUS_LOW_BIT_6 : BOOL;
                                VSTATUS_LOW_BIT_7 : BOOL;
                                END_STRUCT;
END_VAR

// Temporary variables
VAR_TEMP
TOP_SI:    STRUCT
  EV_CLASS:  BYTE;
  EV_NUM:    BYTE;
  PRIORITY:  BYTE;
  NUM:       BYTE;
  TYP2_3:    BYTE;
  TYP1:      BYTE;
  ZI1:       WORD;
  ZI2_3:     DWORD;
END_STRUCT;
START_UP_SI:  STRUCT
  EV_CLASS:  BYTE;
  EV_NUM:    BYTE;
  PRIORITY:  BYTE;
  NUM:       BYTE;
  TYP2_3:    BYTE;
  TYP1:      BYTE;
  ZI1:       WORD;
  ZI2_3:     DWORD;
END_STRUCT;
ERR :        INT;           // Startup error
pbALARM:    BOOL;          // Call up ALARM_8P
pbM_SUP:    BOOL;          // Message suppression
//----- SEL_BOOL.1 -----
pbSEL_BOOL_Q:  BOOL;      // SEL_BOOL 1 output
//----- OP_D.1 and 2 -----
pbQOP_ERR:    BOOL;       // Temporary output value
//----- OP_TRIG.1 -----
pbRESET:     BOOL;        // RESET (Ausgang Q)
//----- Watchdog -----
pbQMON_ERR:  BOOL;        // Watchdog error
pbVerfahren:  BOOL;       // Valve travel
//----- Feedback -----
pbFB_OPEN:   BOOL;
pbFB_CLOSE:  BOOL;
END_VAR
BEGIN
  // Write VSTATUS, resave STEP_NO as STEP_NO_LOC, resave BA_ID as
  //BA_ID_LOC.
  // Supply of the VSTATUS output variable with the inputs.
  VSTATUS_STR.VSTATUS_LOW_BIT_0 := OCCUPIED;
  VSTATUS_STR.VSTATUS_LOW_BIT_1 := BA_EN;
  // Supply of the VSTATUS output variable (HIGH BYTE) with the values
  // that come from the user via the input variable USTATUS (from external).
  VSTATUS_STR.USTATUS := USTATUS;
  //Resave STEP_NO as STEP_NO_LOC und resave BA_ID as BA_ID_LOC
  STEP_NO_LOC := STEP_NO; // Resave, due to output of STEP_NO_LOC in ALARM_8P
  BA_ID_LOC := BA_ID;     // Resave, due to output of BA_ID_LOC in ALARM_8P
  ERR := RD_SINFO (TOP_SI := TOP_SI, START_UP_SI := START_UP_SI); // Read out
                                                                    // start info.

  pbM_SUP := FALSE;
  IF TOP_SI.NUM = 100 OR sbRESTART THEN // Startup or first run
    sbRESTART := FALSE;                // No first run anymore
    QMON_ERR := C_QMON_ERR;            // No error
    pbALARM := TRUE;                    // Initialization ALARM_8P
    IF START_SS THEN                   // At Start SafeStatePosition?
      AUT_ON_OP := false;              // Manual mode
      sbAUT_ON_OP := false;            // Manual mode
      sbMAN_OC := SS_POS;              // Idle position
      MAN_OC := SS_POS;               // Write back idle position
      sbDI_Q_OC := SS_POS;            // Idle position
      srAKTZEIT := 0;                 // Initialize watchdogs time
      QMAN_AUT := C_QMAN_AUT;          // Manual
      QOPENING := C_QOPENING;          // No movement
      QCLOSING := C_QCLOSING;
      sbL_RESET := C_L_RESET;          // No RESET
      RESET := C_RESET;                // No RESET
      QCONTROL := C_QCONTROL;          // Idle position
      QC_QCONTROL := QC_QCONTROL_I;    // Describe quality output with values of
                                                                    // input.
    
```

```

QMSG_ERR := C_QMSG_ERR;
QMSG_SUP := C_QMSG_SUP;
MSG_STAT := C_MSG_STAT;
MSG_ACK := C_MSG_ACK;
QGR_ERR := C_QGR_ERR;
QCL_OP := C_QCL_OP;
QOP_OP := C_QOP_OP;
QOP_ERR := C_QOP_ERR;
QMANOP := C_QMANOP;
QAUTOP := C_QAUTOP;
ELSE;
  IF (NOT AUT_ON_OP AND                               // Manual mode
      MONITOR) THEN                                  // Watchdog active
    IF NOT NO_FB_OP AND                               // Feedback OPEN available
        FB_OPEN THEN                                 // Feedback available
      MAN_OC := TRUE;                                // Track operator input
      QCONTROL := (MAN_OC XOR SS_POS);
      QC_QCONTROL := QC_QCONTROL_I; // Describe quality output with
                                     // values of input.

      QOPENED := TRUE;
      QOPENING := FALSE;
      QCLOSED := FALSE;
      QCLOSING := FALSE;
    END IF;
    IF NOT NO_FB_CL AND                               // Feedback CLOSE available
        FB_CLOSE THEN                                 // Feedback available
      MAN_OC := FALSE;                               // Track operator input
      QCONTROL := (MAN_OC XOR SS_POS);
      QC_QCONTROL := QC_QCONTROL_I; // Describe quality output with
                                     // values of input.

      QOPENED := FALSE;
      QOPENING := FALSE;
      QCLOSED := TRUE;
      QCLOSING := FALSE;
    END IF;
  END IF;
  QCL_OP := CL_OP_EN;
  QOP_OP := OP_OP_EN;
END IF;
IF TOP_SI.NUM = 100 THEN // Initialization at startup
  siRUNUPCNT := RUNUPCYC; // Save the value of the RUNUPCYC input
ELSE;
  // VALVE algorithm
  pbQOP_ERR := FALSE; // Optimistic initializations
  pbQMON_ERR := FALSE;
  pbVerfahren := FALSE; // No OPEN/CLOSE operation
  // OP D.2 (optimized) - 1 -
  IF LIOP_SEL THEN
    // Interconnection
    IF AUT_ON_OP = sbAUT_ON_OP THEN
      ;// No operation occurred
    ELSE; // Prohibited operator
      pbQOP_ERR := TRUE;
    END IF;
    AUT_ON_OP := AUT_L; // Write AUT_L back to operator input
                       // (BTRACK is always1).
    QMAN_AUT := AUT_L; // Interconnection
    QMANOP := FALSE; // No operation allowed
    QAUTOP := FALSE;
  ELSE; // Bedienung
    IF (sbAUT_ON_OP <> AUT_ON_OP) AND
        NOT ((AUT_ON_OP AND AUTOP_EN) OR (NOT(AUT_ON_OP) AND MANOP_EN)) THEN
      pbQOP_ERR := TRUE; // Prohibited operator
      AUT_ON_OP := sbAUT_ON_OP; // Write back old operating values
    ELSE; // No or allowed operation
      QMAN_AUT := AUT_ON_OP;
    END IF;
    QMANOP := MANOP_EN; // Copy the inputs to the outputs
    QAUTOP := AUTOP_EN;
  END IF;
  sbAUT_ON_OP := AUT_ON_OP; //Note old operating values
  //OP TRIG.1 (optimized) - 2 -
  IF ((L_RESET AND (NOT sbL_RESET)) OR RESET) THEN // Rising edge was for
                                                    // L_RESET or RESET.
    pbRESET := TRUE; // RESET
  ELSE;
    pbRESET := FALSE; // No RESET
  END IF;

```

```

sbl_RESET := L_RESET;           // Save previous values
RESET := FALSE;                 // Reset operator-controllable input
// Watchdog RESET Logic - 3 -
IF QMON_ERR THEN                // QMON_ERR Flip-Flop
  IF NOT pbRESET THEN
    pbQMON_ERR := TRUE;
  ELSE;
    // pbQMON_ERR bleibt 0
    // pbRESET bleibt wirksam
    IF TIME_MON < SAMPLE_T THEN // Reposition watchdog time
      srAKTZEIT := SAMPLE_T;
    ELSE;
      srAKTZEIT := TIME_MON;
    END_IF;
  END_IF;
ELSE;
  // RESET without error not effective
  pbRESET := FALSE;
END_IF;
IF V_LOCK OR VL_OPEN OR VL_CLOSE OR (pbQMON_ERR AND FAULT_SS) // SEL_BOOL.1 - 4 -
THEN
  pbSEL_BOOL_Q := SS_POS;       // Idle position
  IF (NOT V_LOCK) THEN
    IF VL_CLOSE THEN
      pbSEL_BOOL_Q := 0;       // Close valve
    ELSE;
      IF VL_OPEN THEN
        pbSEL_BOOL_Q := 1;    // Open valve
      END_IF;
    END_IF;
  END_IF;
ELSE;
  pbSEL_BOOL_Q := AUTO_OC;     // Automatic value
END_IF;
// OP_D.1 (optimized and expanded with LINK ON calculation) - 5 -
IF QMAN_AUT OR V_LOCK OR VL_OPEN OR VL_CLOSE OR (pbQMON_ERR AND FAULT_SS)
// LINK_ON
THEN // Interconnection
  IF MAN_OC = sbMAN_OC THEN
    ; // Keine Bedienung erfolgt
  ELSE;
    pbQOP_ERR := TRUE;        // Prohibited operator
  END_IF;
  IF sbD1_Q_OC <> pbSEL_BOOL_Q THEN // Valve travel?
    pbVerfahren := TRUE;
    sbD1_Q_OC := pbSEL_BOOL_Q; // LINK_I
  END_IF;
  MAN_OC := sbD1_Q_OC;       // Write output back to operator input
  // (BTRACK is always1).
ELSE; // Bedienung
  IF (sbMAN_OC <> MAN_OC) AND
  NOT ((MAN_OC AND OP_OP_EN) OR (NOT(MAN_OC) AND CL_OP_EN)) THEN
    pbQOP_ERR := TRUE;       // Prohibited operator
    MAN_OC := sbMAN_OC;
    // Write back old operating values
  ELSE;
    IF sbD1_Q_OC <> MAN_OC THEN // Valve travel?
      // Allowed operator
      pbVerfahren := TRUE;
      sbD1_Q_OC := MAN_OC;
    END_IF;
  END_IF;
  sbMAN_OC := MAN_OC;       //Note old operating values
  IF pbVerfahren           // Valve is traveled -> Reposition watchdog time
THEN
  IF TIME_MON < SAMPLE_T THEN
    srAKTZEIT := SAMPLE_T;
  ELSE;
    srAKTZEIT := TIME_MON;
  END_IF;
END_IF;

```

```

// Watchdog and position display - 6 -
IF NOT MONITOR THEN // No feedback watchdogs
  IF (srAKTZEIT >= SAMPLE_T) THEN // Timer is running
    srAKTZEIT := srAKTZEIT - SAMPLE_T; // Bring down time
    pbFB_OPEN := FALSE;
    pbFB_CLOSE := FALSE;
  ELSE;
    pbFB_OPEN := sbD1_Q_OC;
    pbFB_CLOSE := NOT sbD1_Q_OC;
  END_IF;
ELSE;
  IF NO_FB_OP THEN // No limit switch OPEN
    pbFB_OPEN := sbD1_Q_OC;
  ELSE;
    pbFB_OPEN := FB_OPEN;
  END_IF;
  IF NO_FB_CL THEN // No limit switch CLOSE
    pbFB_CLOSE := NOT sbD1_Q_OC;
  ELSE;
    pbFB_CLOSE := FB_CLOSE;
  END_IF;
END_IF;
IF MONITOR AND (((sbD1_Q_OC AND NOT pbFB_OPEN) AND NOT NOMON_OP) OR
((NOT sbD1_Q_OC AND NOT pbFB_CLOSE) AND NOT NOMON_CL) OR
(pbFB_OPEN AND pbFB_CLOSE) AND NOT (NOMON_OP AND NOMON_CL)) THEN
  IF (srAKTZEIT < SAMPLE_T) THEN
    pbQMON_ERR := TRUE;
  END_IF;
  IF (srAKTZEIT >= SAMPLE_T) THEN // Timer is running
    srAKTZEIT := srAKTZEIT - SAMPLE_T; // Bring down time
  END_IF;
ELSE;
  IF (sbD1_Q_OC AND pbFB_OPEN) OR
  (NOT sbD1_Q_OC AND pbFB_CLOSE) THEN
    srAKTZEIT := 0; // End position reached-> Reset watchdog time
  END_IF;
END_IF;
QMON_ERR := pbQMON_ERR;
IF QMAN_AUT OR V_LOCK OR VL_OPEN OR VL_CLOSE OR (QMON_ERR AND FAULT_SS) THEN
  // No OPEN/CLOSE operation allowed
  QOP_OP := FALSE;
  QCL_OP := FALSE;
ELSE;
  // Copy the inputs to the outputs
  QOP_OP := OP_OP_EN;
  QCL_OP := CL_OP_EN;
END_IF;
// Position displays
QOPENED := sbD1_Q_OC AND pbFB_OPEN AND NOT pbFB_CLOSE;
QCLOSED := NOT sbD1_Q_OC AND pbFB_CLOSE AND NOT pbFB_OPEN;
IF QOPENED THEN
  QOPENING := FALSE;
ELSE;
  QOPENING := sbD1_Q_OC AND NOT QMON_ERR;
END_IF;
IF QCLOSED THEN
  QCLOSING := FALSE;
ELSE;
  QCLOSING := NOT sbD1_Q_OC AND NOT QMON_ERR;
END_IF;
QOP_ERR := pbQOP_ERR; // Replace outputs
QGR_ERR := QMON_ERR OR CSF; // Calculate group monitor error
IF FAULT_SS AND (NOT(V_LOCK OR VL_OPEN OR VL_CLOSE))
THEN // Safe State Position at error?
  QCONTROL := (sbD1_Q_OC XOR SS_POS) AND NOT QMON_ERR;
  QC_QCONTROL := QC_QCONTROL_I; // Describe quality output with values of
  // input
ELSE;
  QCONTROL := (sbD1_Q_OC XOR SS_POS);
  QC_QCONTROL := QC_QCONTROL_I; // Describe quality output with values of
  // input
END_IF;
IF sirUNUPCNT = 0 THEN // Initialize alarms
  // pbALARM := (sb_SIG_1 <> QMON_ERR) OR (sb_SIG_2 <> CSF);
  pbALARM := TRUE; // Initialization ALARM_8P
ELSE;

```

```

        pbALARM :=FALSE;           // Initialization no ALARM
        siRUNUPCNT := siRUNUPCNT - 1;
        pbM_SUP := TRUE;
    END_IF;
END_IF;
IF pbALARM THEN // Call up ALARM 8P
    IF siBA_ID <> BA_ID_LOC THEN
        // STRING variables may not be interconnected to ALARM8_P as auxiliary
        // process values, therefore transferred in ARRAY OF BYTE.
        FOR ERR := 1 TO 32
            DO
                sbyBA_NA[ERR] := 0; // Delete array as default
            END_FOR;
            ERR := BLKMOV (SRCBLK:= BA_NA,DSTBLK:=sbyBA_NA);
            siBA_ID := BA_ID_LOC; // Save modified BA_ID
        END_IF;

        // Call up ALARM 8P with lock logic (MSG_LOCK)
        ALARM_8P_1( EN_R := TRUE, // Update the ACKL_STATE output
                   ID := 16#EEEE, // PMC communication channel
                   EV_ID:= MSG_EVID,
                   SIG_1:= QMON_ERR,
                   SIG_2:= CSF,
                   SIG_3:= 0,
                   SIG_4:= 0,
                   SIG_5:= 0,
                   SIG_6:= 0,
                   SIG_7:= 0,
                   SIG_8:= 0,
                   SD_1 := sbyBA_NA,
                   SD_2 := STEP_NO_LOC,
                   SD_3 := BA_ID_LOC,
                   SD_4 := AUX_PR04,
                   SD_5 := AUX_PR05,
                   SD_6 := AUX_PR06,
                   SD_7 := AUX_PR07,
                   SD_8 := AUX_PR08,
                   SD_9 := AUX_PR09,
                   SD_10 := AUX_PR10);
        QMSG_ERR := ALARM_8P_1.ERROR;
        MSG_STAT := ALARM_8P_1.STATUS;
        MSG_ACK := ALARM_8P_1.ACK_STATE;
        sb_SIG_1:= QMON_ERR; // Note historical signals
        sb_SIG_2:= CSF;
    END_IF;
    IF (MSG_STAT = 21) THEN // Block locked
        pbM_SUP := TRUE;
    END_IF;
    QMSG_SUP := pbM_SUP;
    QERR := NOT OK; // Note negated OK-Flag result in the block

    //Power supply of the VSTATUS output variable with the outputs.
    VSTATUS_STR.VSTATUS_LOW_BIT_3 := QMAN_AUT;
    VSTATUS_STR.VSTATUS_LOW_BIT_7 := QMON_ERR;
    VSTATUS_STR.VSTATUS_LOW_BIT_8 := V_LOCK;
    VSTATUS_STR.VSTATUS_LOW_BIT_9 := QOPENED;
    VSTATUS_STR.VSTATUS_LOW_BIT_10 := QCLOSED;
    VSTATUS_STR.VSTATUS_LOW_BIT_11 := QOPENING;
    VSTATUS_STR.VSTATUS_LOW_BIT_12 := QCLOSING;
    VSTATUS_STR.VSTATUS_LOW_BIT_14 := QMSG_SUP;
    VSTATUS_STR.VSTATUS_LOW_BIT_15 := OOS;
    VSTATUS := VSTATUS_LOC;
END_FUNCTION_BLOCK

```


Glossar

Einleitung

Das Glossar enthält die wichtigsten Begriffe zum Prozessleitsystem PCS 7.

A

Anwenderdefinierte Datentypen (UDT)

Anwenderdefinierte Datentypen sind vom Anwender erzeugte spezielle Datenstrukturen, die nach ihrer Definition im gesamten CPU-Programm verwendet werden können. Sie können wie elementare oder zusammengesetzte Datentypen in der Variablendeklaration von den Codebausteinen FC, FB und OB verwendet werden oder als Vorlage für die Erstellung von Datenbausteinen mit gleicher Datenstruktur dienen.

AKZ

Anlagenkennzeichen - Wird gebildet aus den Namen der THO, die die Eigenschaft "namensbildend" haben, sowie aus dem Namen des CFC-Plans und des Bausteins im CFC.

AS-Baustein

Objekt einer Bibliothek oder einer Bausteinstruktur, das einen Teil des S7-Anwenderprogramms enthält. Als AS-Bausteine werden die Bausteine bezeichnet, die in der CPU eines AS ablaufen.

Asynchrone OBs

Asynchrone OBs sind Organisationsbausteine, die vom Betriebssystem der CPU beim Auftreten asynchroner Ereignisse aufgerufen werden, z. B. bei einem Fehler.

AUTHOR

→ *Baustein-Attribut*

Enthält bei Verwendung einer Bausteinbibliothek den Bibliotheksnamen. Wird zur Identifikation der zur Bibliothek passenden Hilfedatei verwendet.

AWL

Anweisungsliste - Die Anweisungsliste ist eine maschinennahe, textuelle Programmiersprache gemäß IEC 1131-3.

B

BATCH *flexible*

ab V6.0 → SIMATIC BATCH

Baustein-Attribut

Mit Hilfe der Baustein-Attribute des Bausteins können Sie die Objekteigenschaften Ihres Bausteins beeinflussen. Es gibt die folgenden Baustein-Attribute:

→ *FUNCTION_BLOCK*, → *TITLE*, → *Liste der Systemattribute*, → *AUTHOR*,
→ *NAME*, → *VERSION*, → *FAMILY*, → *KNOW_HOW_PROTECT*

Bausteinkopf

1. Abschnitt des → *AS-Bausteins* mit Verwaltungsinformationen, den → *Baustein-Attributen*.
2. Oberer Teil des Bausteins in der grafischen Darstellung des CFC, der u.a. die Namen (Bausteintyp, Bausteinname), Kommentar und die Task-Zuordnung (Ablaufeigenschaft) enthält.

Bausteinsymbol

Symbolische Darstellung der wichtigsten Informationen eines → *AS-Bausteins*. Über das Bausteinsymbol kann der zugehörige → *Bildbaustein* aufgerufen werden.

Bedienen

Vorgang, bei dem der Anlagenfahrer Wert- bzw. Zustandsveränderungen bei einem Baustein veranlasst. In der Regel wird dies durch Eingaben an der OS eingeleitet, überprüft und an den Baustein im AS weitergeleitet. Da sich in der Zeit zwischen dem OS-Senden und dem AS-Empfang die Prozessbedingungen ändern können, erfolgt im AS eine letzte Überprüfung vor der Zuweisung an den Baustein.

Bedienberechtigung

Berechtigung des aktuellen Anlagenfahrers zum → *Bedienen* des → *AS-Bausteinparameters*

Beobachten

Teil der Aufgaben einer OS, die die Visualisierung der Prozessparameter und Zustände in verschiedenen Formen (numerisch, grafisch) ermöglicht.

Bibliothek

Software-Paket mit nach gemeinsamen Merkmalen zusammengefassten → AS- und/oder → *Bildbausteinen*.

Bildbaustein

Grafische Darstellung aller Elemente eines AS-Bausteins, die zum Bedienen und Beobachten auf einer OS vorgesehen sind.

C

CFC

Continuous Function Chart

1. Funktionsplan, auf dem Bausteine platziert, verschaltet und parametrierbar werden können. Ein CFC-Plan besteht aus 1 bis 26 Teilplänen mit jeweils 6 Blättern.
3. Editor zur technologieorientierten grafischen Projektierung der Automatisierungsaufgabe. Mit dem CFC-Editor wird aus vorgefertigten Bausteinen eine Gesamt-Softwarestruktur (CFC-Plan) erstellt.

CNT-Datei

Optionaler Bestandteil einer Online-Hilfe. Die CNT-Datei enthält das Inhaltsverzeichnis der Online-Hilfe.

Codeteil

Bestandteil eines Bausteins, der den Algorithmus des Bausteins enthält.

D

Deklarationsteil

Bestandteil eines Bausteins, der die Schnittstelle des Bausteins und seine intern verwendeten Daten definiert.

E

Erstlauf

Aus Bausteinsicht der Vorgang, bei dem der Baustein nach seiner Instanziierung zum ersten Mal bearbeitet wird. Anschließend befindet sich der Baustein in einem definierten Zustand bezüglich der Parameter und Betriebsarten.

F

FAMILY

→ *Baustein-Attribut*

Enthält bei Verwendung einer Bausteinbibliothek einen gemeinsamen Namen für eine Teilmenge der Bausteine. FAMILY und → NAME sind Teilschlüssel bei der Suche nach dem Hilfetext eines Bausteins in der Online-Hilfe.

Funktion (FC)

In IEC 1131–3 festgelegt als Software-Einheit, die beim Ausführen ein einziges Ergebnis liefert. Die Funktion kann auch ein strukturierter Datentyp sein. Im Unterschied zum FB hat die Funktion keine speichernde Datenablage (Gedächtnis).

FUNCTION_BLOCK

→ *Baustein-Attribut*

Enthält den symbolischen Namen des Bausteins. Wird für die Anzeige des Namens im SIMATIC Manager und im CFC-Plan verwendet.

Funktionsbaustein (FB)

Ein Funktionsbaustein ist gemäß IEC 1131-3 ein Codebaustein mit statischen Daten. Mit dem FB können im Anwenderprogramm Parameter übergeben werden. Dadurch eignen sich Funktionsbausteine zur Programmierung von häufig wiederkehrenden komplexen Funktionen, z. B. Regelungen und Betriebsartenwahl. Da ein FB über ein Gedächtnis (Instanz-Datenbaustein) verfügt, kann auf seine Parameter (z. B. Ausgänge) zu jeder Zeit an jeder beliebigen Stelle im Anwenderprogramm zugegriffen werden.

G

Global Script

Global Script ist innerhalb von → WinCC der Oberbegriff für vom Anwender erzeugte C-Funktionen, die projektweit oder auch projektübergreifend verwendet werden können.

Graphics Designer

Grafischer Editor in → WinCC zur Erstellung von Bildbausteinen.

K

KNOW_HOW_PROTECT

→ *Baustein-Attribut*

Das gesetzte Attribut schützt den Algorithmus des Bausteins gegen Einsichtnahme und Änderung, wenn die Quelle nicht im gleichen Programm vorhanden ist.

M

Meldeliste

Aus dem Runtimesystem von → *WinCC* heraus besteht die Möglichkeit, Listen mit Meldungen anzeigen zu lassen und zu bearbeiten. Die in den Listen angezeigten Meldungen beziehen sich ausschließlich auf das aktuelle Projekt.

Multiinstanzbaustein

Baustein, der sich aus mehreren Bausteinen zusammensetzt. Seine Instanz umfasst die Instanzen der FBs, die in diesem Baustein zusammengefasst sind und aufgerufen werden. Eine Instanz ist eine Datenablage.

N

NAME

→ *Baustein-Attribut*

Enthält den symbolischen Namen des Bausteins; identisch mit → *FUNCTION_BLOCK.NAME* und → *FAMILY* sind Teilschlüssel bei der Suche nach dem Hilfetext eines Bausteins in der Online-Hilfe.

O

OK-Flag

Das OK-Flag ist eine systeminterne Variable. Wenn während der Ausführung einer Operation ein Fehler auftritt, z. B. Überlauf bei arithmetischen Operationen, wird das OK-Flag vom System beeinflusst und an den Bausteinausgang ENO durchgereicht.

P

Prototypbild

Prototypbilder werden von → *WinCC* dazu verwendet, bereits projektierte Bildkomponenten wieder zu verwenden. Die Technik der Prototypbilder arbeitet mit sogenannten Templatebildern, die mehrfach in ein oder mehrere Vaterbilder eingebunden werden. Ein Templatebild ist nur eine Schablone, die erst in einem echten Objekt zum Leben erweckt wird. Ein Objekt auf Basis einer Schablone (=Prototypbild) entsteht durch eine sogenannte Instanzierung. Zu einer Schablone können mehrere Instanzen (d.h. echte Objekte) erstellt werden.

R

Registrierungsdatei

Eine ASCII-Datei (*.reg), die alle Informationen enthält, z. B. Pfad und Einsprungadressen für Online-Hilfe, um z. B. einen Baustein in der WINDOWS-Registrierung einzutragen. Über diese registrierten Informationen kann im CFC oder SIMATIC Manager für einen selektierten Baustein die Online-Hilfe in der gewünschten Landessprache aufgerufen werden.

S

SCL

Höhere Programmiersprache zur Formulierung technologischer Problemlösungen in der SIMATIC S7. SCL ist PASCAL ähnlich und entspricht der in IEC 1131-3 festgelegten Sprache ST (structured text).

Sicht

Darstellungsart eines Bildbausteins, in der bestimmte Werte eines AS-Bausteins angezeigt werden, z. B. Trendsicht, Meldesicht, Standardsicht usw.

SIMATIC BATCH

Programmpaket für die Erstellung komplexer Rezeptsteuerungen für den gesamten Bereich von kleinen bis hin zu großen Anwendungen.

Split Screen Wizard

Bestandteil von → *WinCC*: Initialisiert die Bildschirm- und Bildeinstellungen der OS.

Standardsicht

→ *Sicht* eines → *Bildbausteins*, bei der die wichtigsten Werte des zugehörigen → *AS-Bausteins* visualisiert werden.

Startinfo

Die Startinformation ist Bestandteil eines Organisationsbausteins (OB). Sie informiert den S7-Anwender detailliert über das Ereignis, das den Aufruf des OB ausgelöst hat.

Systemattribute für Bausteine

Spezielle Attribute, die den → AS-Baustein für die Verbindung mit der OS vorbereiten oder den Einbau des Bausteins in einen CFC-Plan beeinflussen. Beispiel: "S7_tasklist" enthält eine Liste der OBs, in die der Baustein vom CFC eingebaut werden soll.

Systemattribute für Parameter

Spezielle Attribute, die die Darstellung des Parameters durch den → *Bildbaustein* oder seine Behandlung im CFC beeinflussen. Beispiel: "S7_m_c" definiert, ob der Parameter von einer OS aus bedient oder beobachtet werden kann.

T

TITLE

→ *Baustein-Attribut*

Diese Information wird unter PCS 7 nicht ausgewertet, sie wird jedoch im SIMATIC Manager in den Objekteigenschaften des Bausteins im Kommentarfeld angezeigt.

Trendsicht

→ *Sicht* eines → *Bildbausteins*, bei der der zeitliche Verlauf der wichtigsten Werte des zugehörigen → *AS-Bausteins* visualisiert wird.

U

UDT

→ *Anwenderdefinierte Datentypen*

V

VERSION

→ *Baustein-Attribut*

Enthält die Versionsnummer des Bausteins.

W

WinCC

Windows Control Center - Ein Software-Paket zur technologieorientierten grafischen Entwicklung von → *Bildbausteinen* sowie zur Bedienung und Beobachtung des AS.

Index

A

Advanced Process Control.....	2-1
ALARM_8	1-41
ALARM_8P.....	1-44
Analogwertanzeige.....	2-45
Analogwertbedienung.....	2-45
Analogwertdarstellung.....	2-45, 2-48
AS-Baustein	
Anwender-Schnittstelle	1-7
Aufbau.....	1-4
asynchrones Ereignis	1-29
Ausgangsparameter	1-15

B

Balkendarstellung "Grenzwertanzeige"	2-53
Balkendarstellung für Analogwerte	2-49, 2-50
Balkendarstellung horizontal	2-52
Basic Process Control	2-1
Basiselemente	2-45
Batch Occupied	2-56
Batch-Kennung.....	1-42
Batch-Sicht	2-88
Bausteine	
Systemattribute	1-13
Bausteinkopf.....	1-9
Bausteinparameter	1-15
Bausteinsymbol	
CTRL_PID.....	2-101
RATIO_P.....	2-108
SWIT_CNT.....	2-108
Bausteinsymbol	
CTRL_S	2-102
DIG_MON	2-115
DOSE	2-104
ELAP_CNT	2-106
FMCS_PID.....	2-105
FMT_PID.....	2-105
INTERLOK.....	2-113
MEAS_MON	2-107
MOT_REV.....	2-113
MOT_SPED	2-112
MOTOR.....	2-111
OP_A_RJC	2-109
OP_A	2-109

OP_A_LIM.....	2-109
OP_D.....	2-114
OP_D3.....	2-114
OP_TRIG.....	2-114
VAL_MOT.....	2-110
VALVE.....	2-110
Bausteinsymbole.....	2-97
Bausteinsymbol-Vorlagen	2-5
Bedienberechtigungen	
bereichsspezifisch	2-8
instanzspezifisch	2-8
Bedienungen.....	1-31
Begleitwerte	1-44
Bereichszuordnung.....	1-42
Beschriftung	
Optionsfelder	2-59
Optionskästchen.....	2-59
Bibliothek	4-1
Bildbaustein	
Entwurf	2-2
Projektierung	2-3
Test	2-4
Bildbausteine	2-84
Bilder	2-83
Binärwertbedienung	
mit.....	2-58, 2-60
mit 3ComboBox.....	2-64
mit Combobox	2-61
mit Taste.....	2-67
mit Taste und Farbwechsel.....	2-66
Bitmaps	2-80
Blockbild CONTROL	1-4

C

CFC-Bausteintypen.....	1-46
CNT-Datei	3-2
Codeteil	1-25
CTRL_PID	
Bausteinsymbol	2-101
Grenzansicht	2-95
Parametersicht	2-93
Standardsicht	2-90
Wartungssicht.....	2-92
CTRL_S	
Bausteinsymbol	2-102

- D**
- DIG_MON
 - Bausteinsymbol 2-115
 - DOSE
 - Bausteinsymbol 2-104
 - Durchgangparameter 1-15
 - Dynamisierung 2-43
- E**
- Eingangparameter 1-15
 - ELAP_CNT
 - Bausteinsymbol 2-106
 - Erstlauf 1-26
- F**
- Faceplate Designer 2-5
 - FMCS_PID
 - Bausteinsymbol 2-105
 - FMT_PID
 - Bausteinsymbol 2-105
- G**
- Globale Sichten 2-88
 - Globale Skripte 2-6
 - Grunddaten
 - Vorlagenbilder 2-84
- H**
- Herkunft der Meldung 1-42
 - Hilfdatei 3-1
 - Hilfethema 3-1
 - HLP-Datei 3-2
- I**
- INCLUDE-Anweisung 3-2
 - Installationsskript 4-2
 - InstallShield 4-1
 - INTERLOK
 - Bausteinsymbol 2-113
- K**
- Kommentare 1-16
- L**
- Landessprache 1-7
 - Lokale Variablen 1-23
 - Loop-Darstellung
 - erstellen 2-44
- M**
- MEAS_MON A-1
 - Bausteinsymbol 2-107
 - Meldebegleitwerte 1-45
 - Meldekasse 1-43
 - Meldesicht 2-88
 - Meldungen 1-33
 - Meldungsprojektierung 1-41
 - Meldungsunterdrückung 2-55
 - MOT_REV
 - Bausteinsymbol 2-113
 - MOT_SPED
 - Bausteinsymbol 2-112
 - MOTOR A-7
 - Bausteinsymbol 2-111
 - Multiinstanz 1-23, 2-11
 - Multiinstanzbaustein 1-23
- N**
- Namenskonventionen 1-48
 - Nummernbereich 1-48
- O**
- Objekt-Baukasten 2-6
 - Objekteigenschaften des Bausteins 1-10, 1-11
 - Online-Hilfe 3-1
 - OP_A_RJC
 - Bausteinsymbol 2-109
 - OP_A 1-32
 - Bausteinsymbol 2-109
 - OP_A_LIM 1-32
 - Bausteinsymbol 2-109
 - OP_A_RJC 1-32
 - OP_D
 - Bausteinsymbol 2-114
 - OP_D3
 - Bausteinsymbol 2-114
 - OP_TRIG
 - Bausteinsymbol 2-114
- P**
- Parameter-Attribute 2-28
 - Parametertypen 1-15
 - Permission 2-71
 - Plananschlüsse 1-46

Q		T	
Quality Code-Anzeige	2-77	Tag Logging	2-17
Quellcode	1-49	Temporäre Variablen	1-24
Quittierung der Meldungen	2-56	Test	
R		Bildbaustein	2-4
RATIO_P		Topic	3-1
Bausteinsymbol	2-108	Trendsicht	2-16, 2-89
Registrierungsdatei	3-3	U	
S		Überschrift des Bedienbildes	2-59
S7-Bibliothek	4-1	Übersicht	2-10
Sammelanzeige	2-57	V	
SAMPLE_T	1-27	VAL_MOT	
SCL-Compiler		Bausteinsymbol	2-110
Voreinstellungen	1-5	VALVE	A-15
Setup	4-2	Bausteinsymbol	2-110
SIMATIC BATCH	1-45	Verriegelt	2-57
Skripte	2-78	Vorlagen	2-5
Sprachkennung	3-2	Vorlagenbilder	2-6
Sprachumschaltung	2-27	W	
Statische Variablen	1-23	Weckalarm-OB	1-29
Statischer Text	2-49	WinCC Graphics Designer	2-3
Statusanzeige		Z	
mit 2 Alternativen	2-68	Zahlenformatierung	2-15
mit n Alternativen	2-69	Zeitabhängigkeit	1-27
SUPPTIME	1-28	Zustandsanzeige:	2-70, 2-71
SWIT_CNT			
Bausteinsymbol	2-108		
Symboltabelle	1-7		
Systemattribute			
Bausteine	1-13		
Parameter	1-16		

