

SIEMENS

SIMATIC

Programmieranleitung Bausteine erstellen für PCS 7

Handbuch

Vorwort
Inhaltsverzeichnis

AS-Bausteine erstellen

Bildbausteine erstellen

Online-Hilfe erstellen

Bibliothek und Setup erstellen

Glossar, Index

1

2

3

4

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährdungsgrad folgendermaßen dargestellt:



Gefahr

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



Warnung

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



Vorsicht

bedeutet, dass eine leichte Körperverletzung oder ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

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

Achtung

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs 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

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind 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 können.

Copyright © Siemens AG 2000 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung

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, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D- 90327 Nürnberg

© Siemens AG 2000
Technische Änderungen bleiben vorbehalten

Siemens Aktiengesellschaft

A5E00092263



Vorwort

Zweck der Programmieranleitung

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

Die wesentlichen Punkte, die einen PCS 7-konformen AS-Baustein von einem reinen S7-Baustein unterscheiden, sind:

- Die Möglichkeit, Parameterwerte über einen Bildbaustein zu **beobachten**.
- Die Möglichkeit, Parameterwerte und damit das Verhalten des Bausteins über einen Bildbaustein zu **bedienen**
- Die Möglichkeit, asynchron auftretende Ereignisse und Bausteinzustände an die OS zu **melden** und dort über einen Bildbaustein oder eine WinCC-Meldeliste anzuzeigen.

Leserkreis

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

Voraussetzungen

Voraussetzungen sind daher 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 können Sie dem PCS7 Projektierungshandbuch entnehmen.

Vorgehensweise

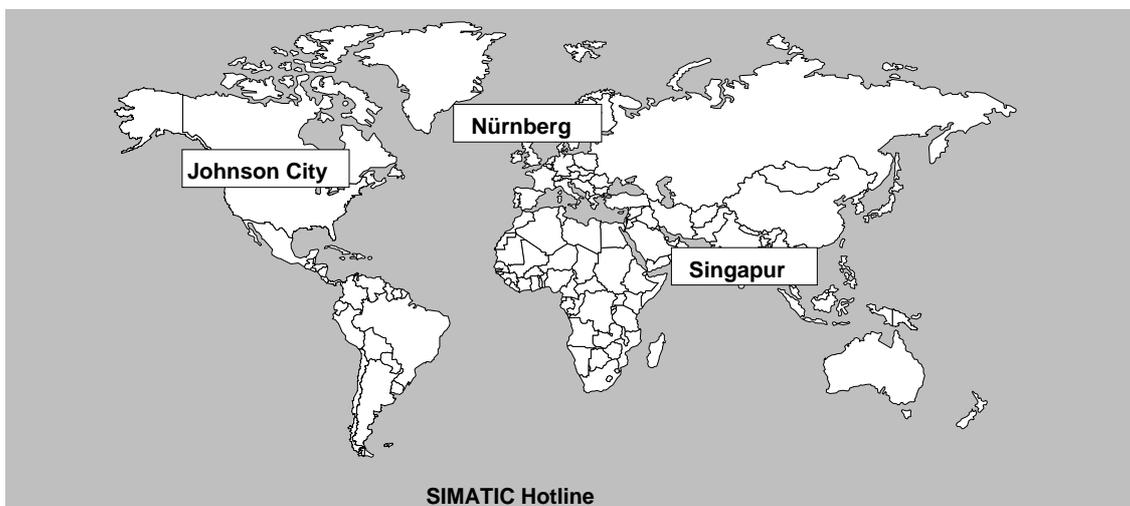
Die Programmieranleitung bietet Ihnen einen Überblick über die einzelnen Bestandteile eines PCS 7-konformen Bausteins. Sie orientiert sich dabei an der Reihenfolge, in der Sie vorgehen, wenn Sie Funktions- und Bildbausteine entwickeln.

- 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 Designers.
- 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 1.10 abgedruckt.

Customer Support, Technical Support

Weltweit erreichbar zu jeder Tageszeit:



Weltweit (Nürnberg) Technical Support	Weltweit (Nürnberg) Technical Support	
(FreeContact) Ortszeit: Mo.-Fr. 7:00 bis 17:00 Telefon: +49 (180) 5050 222 Fax: +49 (180) 5050 223 E-Mail: techsupport@ ad.siemens.de GMT: +1:00	(kostenpflichtig, nur mit SIMATIC Card) Ortszeit: Mo.-Fr. 0:00 bis 24:00 Telefon: +49 (911) 895-7777 Fax: +49 (911) 895-7001 GMT: +01:00	
Europa / Afrika (Nürnberg) Autorisierung	Amerika (Johnson City) Technical Support und Autorisierung	Asien / Australien (Singapur) Technical Support und Autorisierung
Ortszeit: Mo.-Fr. 7:00 bis 17:00 Telefon: +49 (911) 895-7200 Fax: +49 (911) 895-7201 E-Mail: authorization@ nbgm.siemens.de GMT: +1:00	Ortszeit: Mo.-Fr. 8:00 bis 19:00 Telefon: +1 423 461-2522 Fax: +1 423 461-2289 E-Mail: simatic.hotline@ sea.siemens.com GMT: -5:00	Ortszeit: Mo.-Fr. 8:30 bis 17:30 Telefon: +65 740-7000 Fax: +65 740-7001 E-Mail: simatic.hotline@ sae.siemens.com.sg GMT: +8:00
Die Sprachen an den SIMATIC Hotlines sind generell Deutsch und Englisch, bei der Autorisierungshotline wird zusätzlich Französisch, Italienisch und Spanisch gesprochen.		

SIMATIC Customer Support Online-Dienste

Das SIMATIC Customer Support bietet Ihnen über die Online-Dienste umfangreiche zusätzliche Informationen zu den SIMATIC-Produkten:

- Allgemeine aktuelle Informationen erhalten Sie
 - im **Internet** unter <http://www.ad.siemens.de/simatic>
- Aktuelle Produkt-Informationen und Downloads, die beim Einsatz nützlich sein können:
 - im **Internet** unter <http://www.ad.siemens.de/simatic-cs>
 - über das **Bulletin Board System** (BBS) in Nürnberg (*SIMATIC Customer Support Mailbox*) unter der Nummer +49 (911) 895-7100.

Verwenden Sie zur Anwahl der Mailbox ein Modem mit bis zu V.34 (28,8 kBaud), dessen Parameter Sie wie folgt einstellen: 8, N, 1, ANSI, oder wählen Sie sich per ISDN (x.75, 64 kBit) ein.

- Ihren Ansprechpartner für Automation & Drives vor Ort finden Sie über unsere Ansprechpartner-Datenbank:
 - im **Internet** unter <http://www3.ad.siemens.de/partner/search.asp>

Inhaltsverzeichnis

1	AS-Bausteine erstellen	
1.1	Voraussetzungen und Vorkenntnisse	1-1
1.1.1	Mitgelieferter Beispiel-Baustein.....	1-1
1.2	Aufbau eines AS-Bausteins	1-2
1.2.1	Voreinstellungen im SCL-Compiler	1-3
1.2.2	Voreinstellungen im SIMATIC Manager.....	1-5
1.2.3	Bausteinkopf.....	1-6
1.2.4	Deklarationsteil.....	1-11
1.2.5	Codeteil	1-19
1.3	Erstlauf	1-20
1.4	Zeitabhängigkeit.....	1-21
1.5	Behandlung von asynchronen Anlauf- und Fehler-OBs	1-23
1.6	Bedienen, Beobachten und Melden.....	1-26
1.6.1	Meldungsunterdrückung im Anlauf	1-31
1.6.2	Meldungsunterdrückung für bestimmte Meldungen.....	1-32
1.6.3	Quelle übersetzen	1-32
1.7	Meldungsprojektierung.....	1-33
1.8	Anbindung von Batch <i>flexible</i>	1-36
1.9	Erstellen von CFC-Bausteintypen	1-37
1.9.1	Beispiel: CONTROL2	1-37
1.10	Namenskonventionen und Nummernbereich.....	1-38
1.11	Quellcode des Beispiels.....	1-39
2	Bildbausteine erstellen	
2.1	Erstellungsweg	2-1
2.1.1	Entwurf des Bildbausteins.....	2-1
2.1.2	Projektierung des Bildbausteins.....	2-2
2.1.3	Test des Bildbausteins	2-3
2.2	Bildbausteinerstellung mit dem Faceplate Designer.....	2-4
2.2.1	Vorlagen des Faceplate Designer	2-4
2.2.2	Projektierungsschritte.....	2-6
2.2.3	Dynamisierung von Bildbausteinen.....	2-14
2.2.4	Sprachumschaltung	2-15
3	Online-Hilfe erstellen	
3.1	Aufbau der Hilfedatei.....	3-1
3.2	Aufbau der Registrierungsdatei.....	3-3
4	Bibliothek und Setup erstellen	
4.1	Bibliothek erstellen	4-1
4.2	Setup erstellen	4-2

Glossar

Index

1 AS-Bausteine erstellen

1.1 Voraussetzungen und Vorkenntnisse

Die hier beschriebenen Bausteine sind für die Verwendung mit PCS 7 ab V5.1 auf einer CPU S7-4xx gedacht. Für die Erstellung der Bausteine benötigen Sie die folgenden Softwarepakete:

- STEP7 Basis ab V5.1
- SCL-Compiler ab V5.1
- CFC ab V5.1

AS-Bausteine für PCS 7 werden mit der Programmiersprache SCL erstellt. Daher wird im Folgenden nur dieser Weg beschrieben. Weitere Informationen zu SCL entnehmen Sie bitte:

- der Online-Hilfe im SIMATIC Manager
(Aufruf von Hilfen zu Optionspaketen > Bausteine programmieren mit S7-SCL).
- dem Handbuch "S7-SCL für S7-300 und S7-400"

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

1.1.1 Mitgelieferter Beispiel-Baustein

Der hier beschriebene Baustein "CONTROL" wird als SCL-Quelle "**S7_CONTA.SCL**" (Deutsch) bzw. "**S7_CONTB.SCL**" (Englisch) bei PCS7TOOLS mitgeliefert und unter ...\\STEP7\\Examples\\ZDt25_01 bzw. ...\\ZEn25_01 installiert.

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

- Selektieren Sie in Ihrem Projekt den Quellordner und wählen Sie **Einfügen > Externe Quelle...**. Im Dialogfeld "Externe Quelle einfügen" gehen Sie in der Verzeichnisstruktur bis zum Ablageort der SCL-Quelle, selektieren sie und klicken auf "Öffnen".

Die SCL-Quelle befindet sich nun im Quellordner und muss mit dem SCL-Compiler übersetzt werden.

Stellen Sie vor dem Übersetzen sicher, dass sich der Baustein "OP_A_LIM" (FB 46) im Bausteinordner Ihres Projekts befindet. Ist dies nicht der Fall, kopieren Sie ihn aus der Bibliothek "PCS 7 Library\\Technological Blocks" in Ihr Projekt.

- Öffnen Sie mit Doppelklick die SCL-Quelle "S7_CONTA oder S7_CONTB", 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

Damit ein AS-Baustein im PCS 7-Umfeld lauffähig ist, muss er bestimmte formale und inhaltliche Kriterien erfüllen. Die folgenden Kapitel beschreiben, was Sie zur Erfüllung dieser Kriterien beachten bzw. unternehmen müssen.

Das untenstehende Blockbild zeigt den prinzipiellen Aufbau des Bausteins "CONTROL" (FB501). Die einzelnen Elemente des Bausteins werden in den angegebenen Kapiteln genauer erklärt.

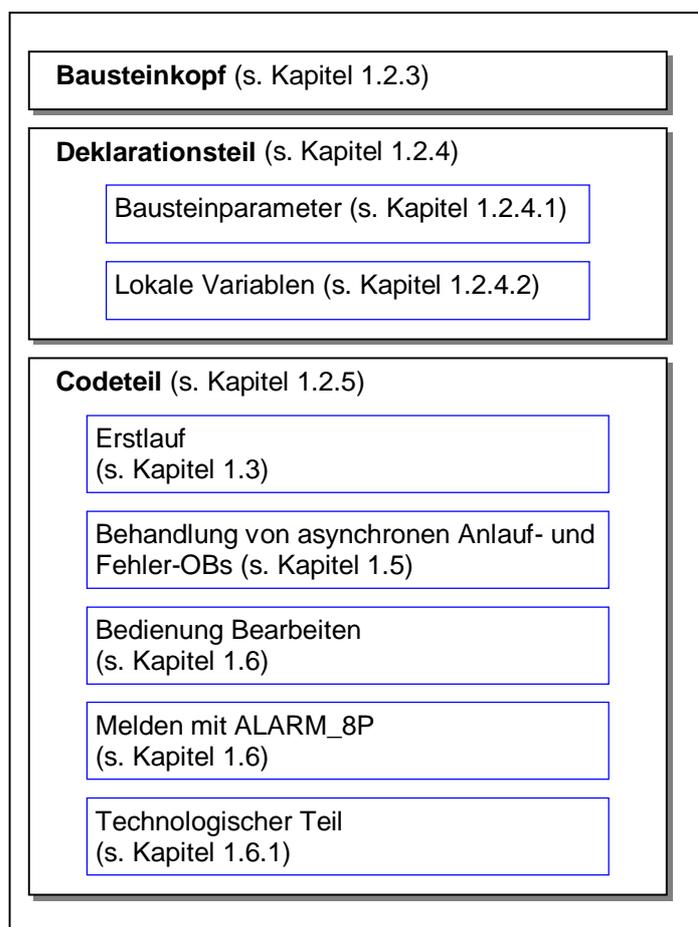


Bild 1-1: Aufbau des Beispiel-Bausteins "CONTROL" (FB501)

Funktionsbaustein oder Funktion

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

Falls Sie dies nicht benötigen, können Sie Ihren Baustein auch als Function (FC) realisieren.

1.2.1 Voreinstellungen im SCL-Compiler

Voreinstellungen für "Bausteine erzeugen"

Im SCL-Compiler sind unter **Options > Customize > Create Block** folgende Optionen einstellbar:

- **Overwrite blocks**

Überschreibt bereits existierende Bausteine im Ordner "Bausteine" eines S7-Programms, falls beim Übersetzungsvorgang Bausteine mit der gleichen Bezeichnung erzeugt werden.

Ebenso werden beim Laden Bausteine mit gleichem Namen, die bereits im Zielsystem vorhanden sind, überschrieben.

Wenn Sie diese Option nicht gewählt haben, müssen Sie eine Meldung bestätigen, bevor der Baustein überschrieben wird.

- **Display warnings**

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

- **Display errors before warnings**

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

- **Generate reference data**

Wählen Sie diese Option, wenn bei der Erzeugung eines Bausteins automatisch Referenzdaten erzeugt werden sollen.

Über den Menübefehl **Options > Reference Data** haben Sie jedoch die Möglichkeit, die Referenzdaten später zu erzeugen oder zu aktualisieren.

- **Include system attribute "S7_server"**

Wählen Sie diese Option, wenn bei der Erzeugung eines Bausteins das Systemattribut für Parameter "S7_server" berücksichtigt werden soll. Dieses Attribut vergeben Sie, wenn der Parameter für die Verbindungs- oder Meldungsprojektierung relevant ist. Der Parameter enthält die Verbindungs- bzw. die Meldungsnummern.

Hinweis für dieses Beispiel:

Das Optionskästchen "Include system attribute 'S7_server' " müssen Sie unbedingt setzen, da dieser Baustein Meldungen enthält. Beim Importieren bzw. beim Einfügen des Bausteins in einen CFC-Plan würde dieser Vorgang sonst mit Fehlermeldung abgebrochen.

Voreinstellungen für "Compiler"

Im SCL-Compiler können Sie unter **Options > Customize > Compiler** die drei Optionskästchen

- "Monitor array limits"
- "Create debug info"
- "Set OK flag"

an- oder abwählen. Die restlichen Optionskästchen sollten Sie immer angewählt lassen. Nähere Informationen zu den einzelnen Optionen können sie dem SCL-Handbuch entnehmen.

Bei der Entscheidung, ob Sie diese Optionen anwählen oder nicht, müssen Sie die folgenden Punkte beachten:

- **Monitor array limits**

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.

Falls Sie Arrays verwenden, sollten Sie die Option nur solange angewählt lassen, bis Ihr Baustein ausreichend getestet ist und sicher ist, dass Index und Feldlänge zusammenpassen.

- **Create debug info**

Diese Option ermöglicht es einen Testlauf mit dem Debugger durchzuführen, nachdem das Programm übersetzt und in die CPU geladen ist. Der Speicherbedarf des Programmes und die Ablaufzeiten im AS erhöhen sich jedoch durch diese Option. Sie sollten Sie daher nur während der Testphase des Bausteins anwählen, nicht aber in der Lieferversion.

- **Set OK flag**

Das OK Flag ist eine systeminterne Variable. Tritt während der Ausführung einer Operation ein Fehler auf, z.B. Überlauf bei arithmetischen Operationen, so 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 möchten. (Dies wird vom System übernommen und kostet keine Performance, da der Zustand des OK-Flags stets vom System an den Ausgang durchgereicht wird.)

1.2.2 Voreinstellungen im SIMATIC Manager

Bei PCS 7-konformen AS-Bausteinen sollte die Schnittstelle zum Anwender (Parameternamen, Kommentare usw.) in englischer Sprache sein. Die Bausteine selbst können Sie in jeder beliebigen Landessprache entwickeln. Die Beschreibung und die Beispiele beziehen sich auf die Landessprache "Englisch".

Auswahl der Landessprache

Falls Sie Ihre Bausteine in einer Bibliothek zusammenfassen wollen (vgl. Kapitel 4), muss auch die "Landessprache" auf Englisch eingestellt sein, damit die einzelnen Kataloge Ihrer Bibliothek PCS 7-konforme Namen erhalten (**Sources**, **Symbols** und **Blocks**). Hierzu müssen Sie im SIMATIC Manager über **Extras > Einstellungen > Sprache** als Landessprache und als Mnemonic "English" einstellen.

Eintrag in der Symboltabelle

Der Name des Bausteins, der im Bausteinkopf eingetragen wird (wie nachfolgend beschrieben), muss in der Symboltabelle als symbolischer Name eingetragen werden.

Zum Eintragen öffnen Sie im S7-Programm mit Doppelklick auf "Symbols" die Symboltabelle. Geben Sie in der Spalte "Symbol" den symbolischen Namen (hier: "CONTROL") ein, und ordnen Sie ihm in der Spalte "Address" eine FB-Nummer zu (hier: FB 501).

Siehe dazu auch Abschnitt 1.10, Namenskonventionen und Nummernbereich.

1.2.3 Bausteinkopf

Der Bausteinkopf enthält die Verwaltungsinformationen (im weiteren Baustein-Attribute genannt) des Bausteins. 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 (vgl. dazu Attribut KNOW_HOW_PROTECT).

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 die Stellgröße 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 die Stellgröße 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 BATCH flexible und besitzt daher 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:          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';      // Bausteins ist bedien- und beobachtbar
S7_alarm_ui:= '1';        // Einstellung PCS7-Melddialog ('0'=Standard-Melddialog)
}
AUTHOR:      ABC
NAME:        CONTROL
VERSION:     '0.01'
FAMILY:      XYZ
KNOW_HOW_PROTECT
```

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

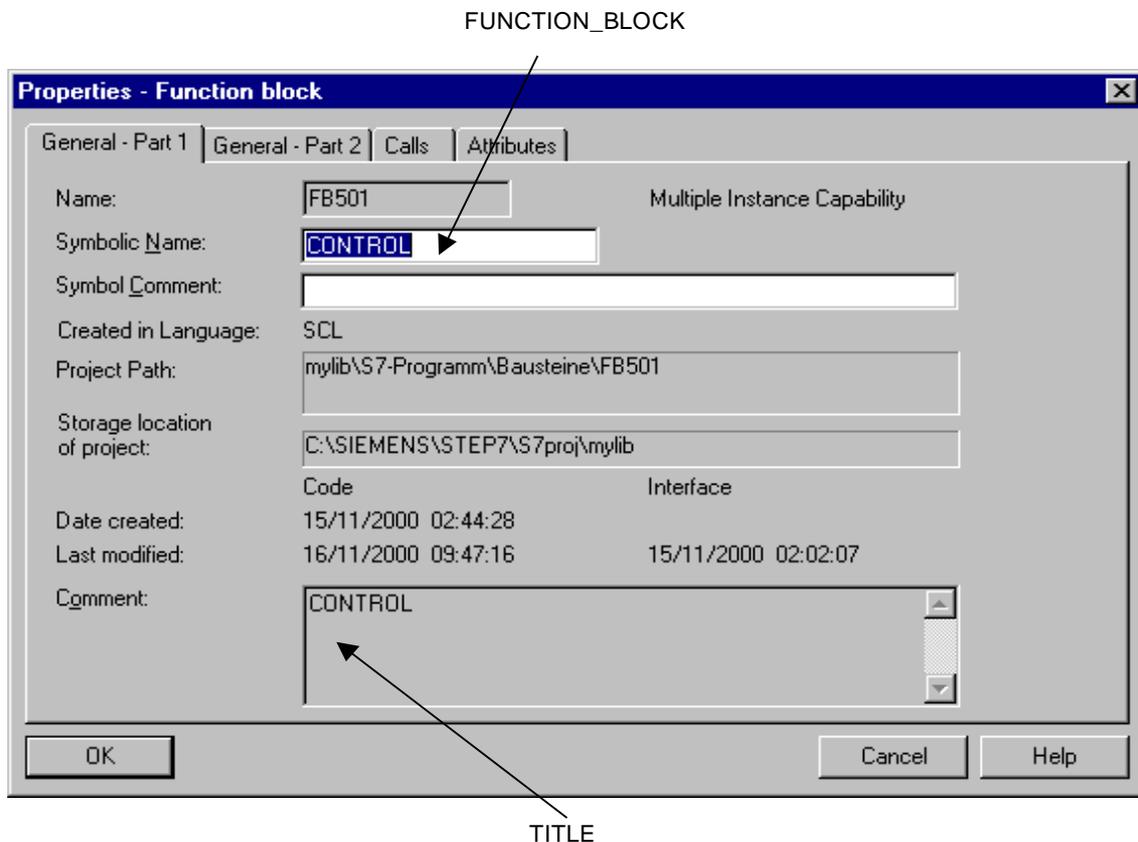


Bild 1-2: Objekteigenschaften des Bausteins (Teil 1)

- **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. Vor dem Übersetzen des Bausteins muss diesem Namen in der Symboltabelle eine Bausteinnummer zugeordnet werden.

- **TITLE**

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

Es wird empfohlen, hier den selben Namen einzutragen, wie bei FUNCTION_BLOCK.

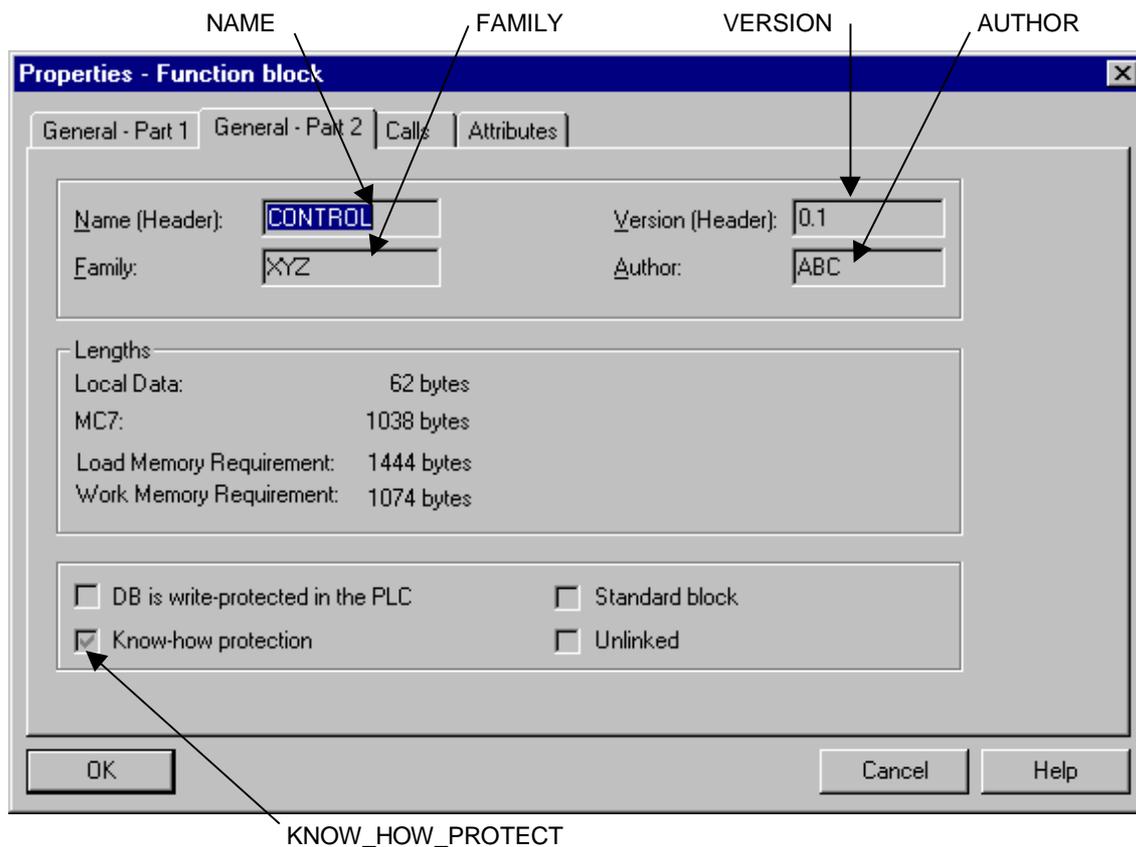


Bild 1-3: Objekteigenschaften des Bausteins (Teil 2)

- **NAME**
Tragen Sie hier denselben Namen wie bei FUNCTION_BLOCK ein. Bei Verwendung einer Online-Hilfe ist dieser Name (und FAMILY) ein Teilschlüssel zur Suche des Hilfetexts dieses Bausteins in der Hilfedatei.
- **VERSION**
Tragen Sie hier eine Versionskennung von 0.00 bis 15.15 ein.
- **FAMILY**
Falls Sie Ihre Bausteine zu einer eigenen Bibliothek zusammenfassen und in dieser Bibliothek in verschiedene Gruppen einordnen wollen, geben Sie hier einen maximal 8 Zeichen langen Gruppennamen für diesen Baustein an.
Bei Verwendung einer Online-Hilfe sind FAMILY und NAME Teilschlüssel zur Suche des Hilfetexts dieses Bausteins in der Hilfedatei.

- **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:

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

- **KNOW_HOW_PROTECT**

Mit Hilfe dieses Attributs können Sie den Algorithmus und die Attribute des Bausteins gegen Einsichtnahme und Änderung schützen. Ist es gesetzt, werden die Attribute des Bausteins im SIMATIC Manager in den Objekteigenschaften des Bausteins nur angezeigt, können aber nicht geändert werden. Der Baustein selbst lässt sich außerhalb Ihres Projekts ohne die entsprechende Quelle nur noch per AWL-Editor und nicht mehr per SCL öffnen. Es werden dabei jedoch nur die Bausteinparameter angezeigt. Im eigenen Projekt 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. So definiert z.B: **S7_m_c**, ob der Baustein für eine OS relevant ist, d.h. ob dafür in der OS notwendige interne Datenstrukturen angelegt werden. Zudem können Sie mit den Systemattributen den Einbau des Bausteins in einen CFC-Plan steuern. So wird z.B. mit **S7_tasklist** festgelegt, in welche OBs der Baustein automatisch eingebaut werden soll.

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' PCS7-Meldedialog	S7_alarm_ui := '0'
S7_tag	Hat dieses Systemattribut den Wert 'false', wird der Baustein nicht in die Messstellenliste der OS eingetragen, d.h. er erhält auf der OS keine Anwahl für "Loop in Alarm". Dies ist sinnvoll für Bausteine, die nur melden, aber keinen Bildbaustein besitzen.	false

Die Systemattribute werden im SIMATIC Manager in den Objekteigenschaften des Bausteins im Register "Attribute" angezeigt und können dort auch geändert werden, falls der Baustein nicht schreibgeschützt ist (Attribut **KNOW_HOW_PROTECT** im Bausteinkopf).

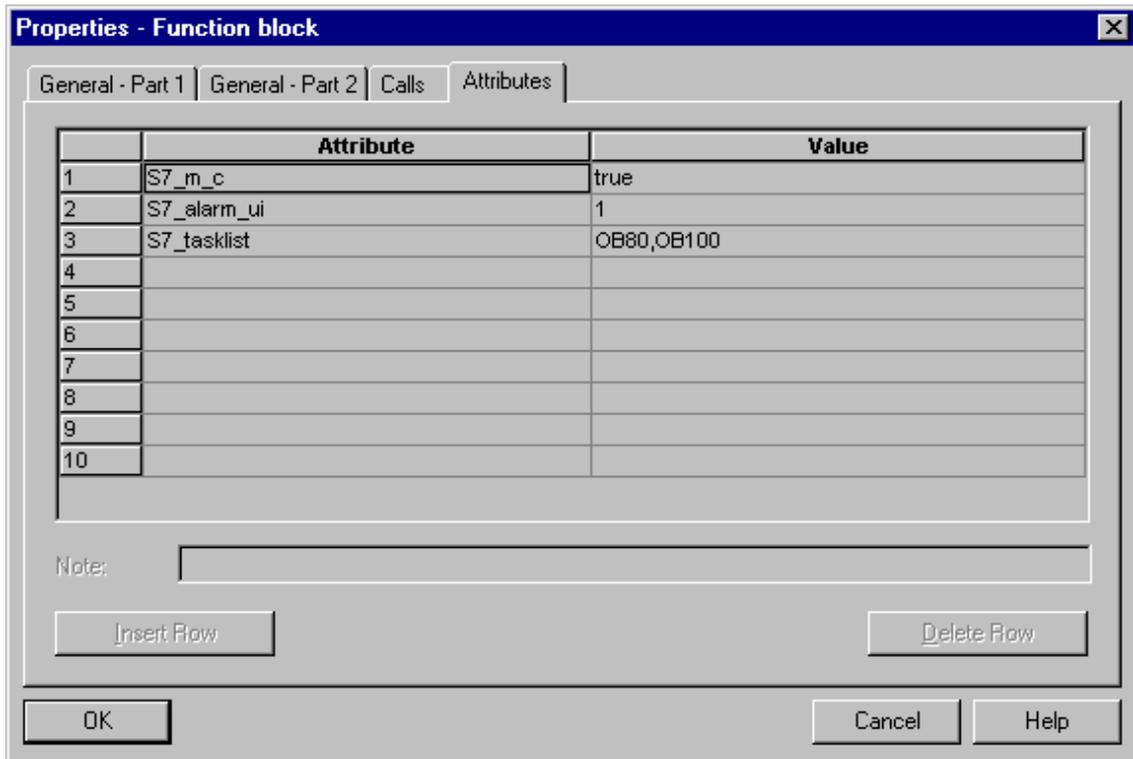


Bild 1-4: Systemattribute des Bausteins

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

Die Bausteinparameter definieren die Schnittstelle des Bausteins zu anderen Bausteinen sowie zu den Bedien- und Beobachtungs-Werkzeugen (CFC, WinCC ...).

Parametertypen

Es gibt die folgenden Parametertypen:

- **Eingangsparameter**

Die Festlegung als Eingangsparameter ist für PCS 7-konforme Bausteine erforderlich, wenn Sie

- Parameterwerte von einem anderen Baustein holen wollen oder
- Parameter von der OS aus bedienen wollen oder
- die Darstellung eines Bildbausteins auf der OS per Parameter beeinflussbar machen wollen (z.B. Grenzen für Darstellbereiche) oder
- Parameter für Testzwecke vom CFC aus bedienbar machen wollen oder
- Parameter zur Erzeugung von Meldungen (Message Event-ID des ALARM_8P-Bausteins)

- **Ausgangsparameter**

Die Festlegung als Ausgangsparameter ist für PCS 7-konforme Bausteine erforderlich, wenn Sie

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

- **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 bei der Umschaltung vom verschalteten auf den bedienten Wert diese Umschaltung stoßfrei erfolgt.

Kommentare für Parameter

Falls 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 auch geändert werden, unabhängig vom Know-How-Schutz (Attribut KNOW_HOW_PROTECT im Bausteinkopf).

Systemattribute für Parameter

Bausteinparameter können ebenfalls (wie der Baustein selbst) mit Hilfe von Systemattributen weiter spezifiziert werden.

Sie können damit definieren,

- wie der Parameter auf der OS dargestellt werden soll.
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 werden soll.
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

System-attribut	Betrifft	Bedeutung	Default
S7_sample-time	Zeitverhalten	Besitzt ein Parameter dieses Systemattribut, wird er automatisch mit der Zykluszeit des aufrufenden Weck-OBs parametrierbar. Dazu muss bei der Übersetzung des CFC-Plans das Optionskästchen "Aktualisierung der Abtastzeit" angewählt sein (vgl. Kapitel 1.4).	false
S7_dynamic	CFC	Besitzt ein Parameter dieses Systemattribut, 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_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.	- -

System-attribut	Betrifft	Bedeutung	Default
S7_string_0	B&B	Dieses Systemattribut ist nur für Eingangsparmeter (oder Durchgangsparmeter) 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 Eingangsparmeter (oder Durchgangsparmeter) 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 eine max. 16 Zeichen lange Einheit 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 bzw. 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 müssen Sie den Wert des Systemattributs in zwei Teile teilen und diese beiden durch ein Gleichheitszeichen trennen, z.B. S7_string_1 := 'Suppress HH =YES'. Der CFC erkennt das Gleichheitszeichen und ersetzt die Wertausgabe am Parameter durch den Teil hinter dem Gleichheitszeichen; d.h. in diesem Fall wird statt dem Wert 1 das Wort "YES" ausgegeben.

Es werden im CFC maximal 8 Zeichen ausgegeben, auch wenn Sie mehr angeben. Im Bildbaustein wird immer der ganze Text ausgegeben, d.h. hier "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.

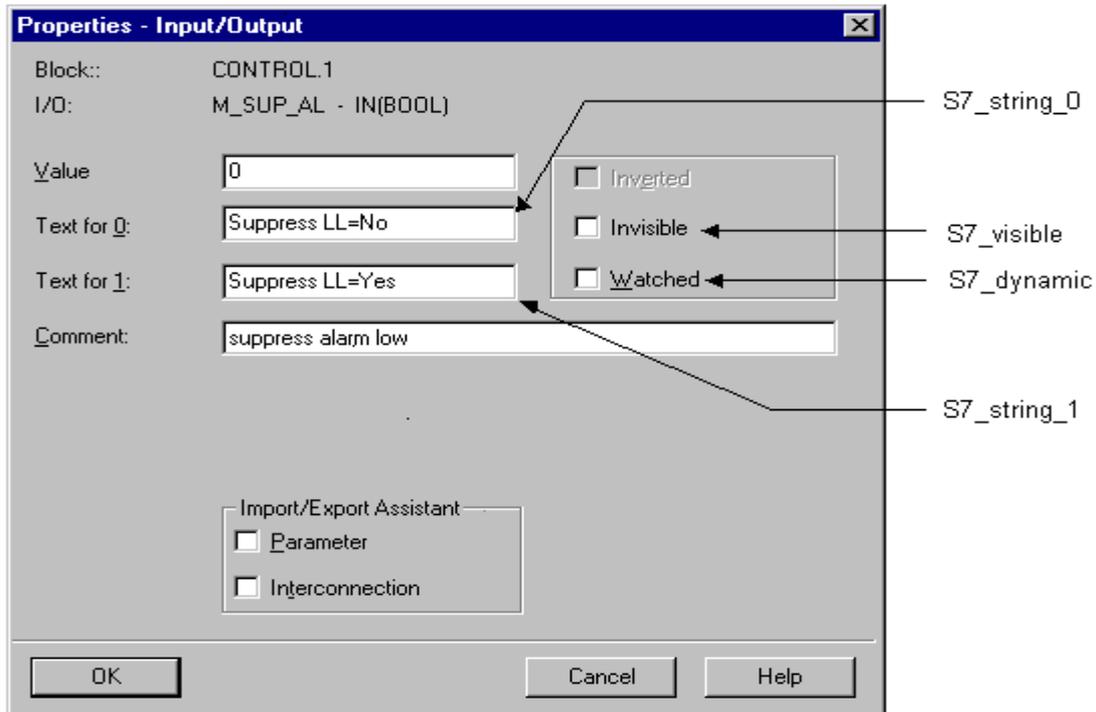


Bild 1-5: Objekteigenschaften von Parametern vom Datentyp BOOL

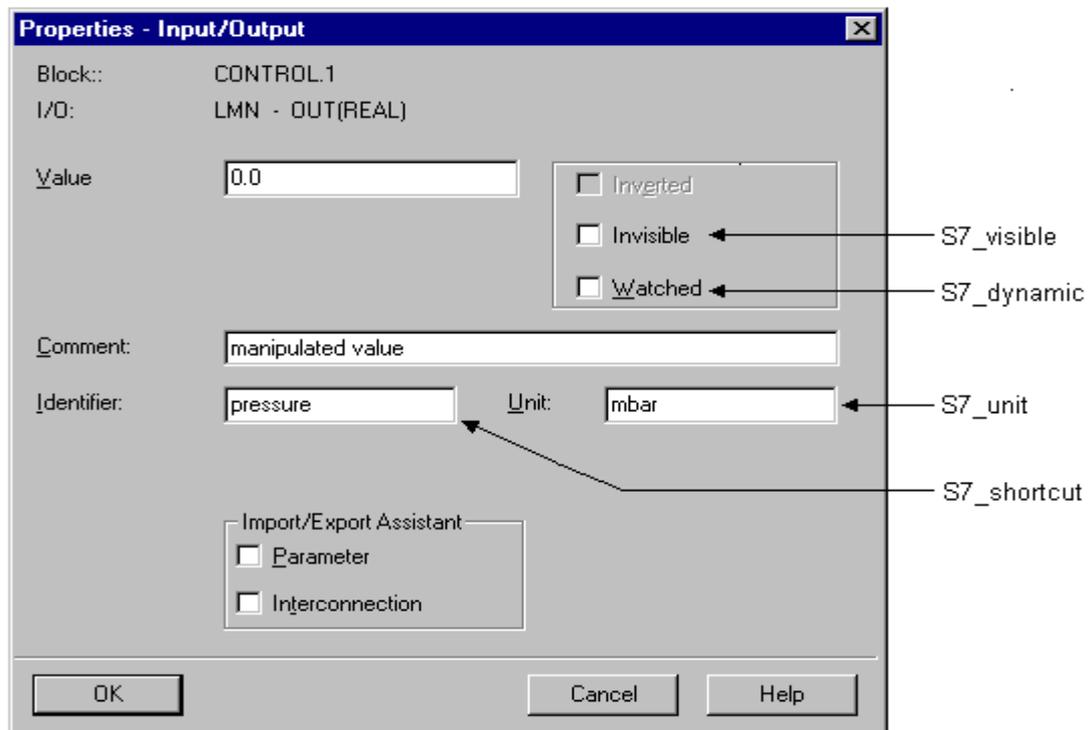


Bild 1-6: Objekteigenschaften von Parametern, die nicht vom Datentyp BOOL sind

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; // sample time [s] (Vorbesetzung 1 Sek)

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; // lower alarm limit (Vorbesetzung 0)

H_ALM {S7_m_c := 'true';
       S7_visible:= 'false';
       S7_link := 'false'} :REAL :=100; // upper alarm limit (Vorbesetzung 100)

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 // suppress alarm low

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; // suppress alarm high

SP_OP_ON {S7_visible:= 'false';
          S7_dynamic:= 'true' // CFC in Test/IBS: Anzeige aktueller Wert in AS)
        }
          :BOOL := 1; // Enable l=Operator for Setpoint Input

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; // Enable l=Bumpless for Setpoint On

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

SP_EXT {S7_dynamic:= 'true'}

          :REAL := 0; // External Setpoint
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; // Setpoint High Limit

SP_LLM {S7_visible:= 'false';
       S7_link:= 'false';
       S7_m_c:= 'true';
       S7_shortcut:= 'SP low limit';
       S7_unit:= ''
      }
       :REAL := 0; // Setpoint Low Limit

```

```

GAIN {S7_link:='false';
  S7_edit:='para'; // Parametrierung im Import-Export-Assistenten (IEA)
  S7_m_c:='true';
  S7_shortcut:='Gain';
  S7_unit:='' } :REAL := 1; // Proportional Gain

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; // Message ID

// Parameter für BATCH flexible
STEP_NO {S7_visible := 'false';
  S7_m_c := 'true'} :WORD; // Batch Step Number
BA_ID {S7_visible := 'false';
  S7_m_c := 'true'} :DWORD; // Batch ID
BA_EN {S7_visible := 'false'; // Parameter im CFC-Plan unsichtbar
  S7_m_c := 'true' // Parameter ist B&B-fähig
  } :BOOL := 0; // Batch Enable
BA_NA {S7_visible := 'false';
  S7_m_c := 'true'} :STRING[16] := ''; // Batch Name

OCCUPIED {S7_visible := 'false';
  S7_m_c := 'true'} :BOOL := 0; // Occupied by Batch

RUNUPCYC {S7_visible:='false';
  S7_link:='false'} :INT := 3; // Number of Run Up Cycles
SUPPTIME :REAL := 0; // sample delay
SUPP_IN :REAL := 0; // Input value for sample delay
END_VAR

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

QH_ALM :BOOL := false; // 1 = HH alarm active
QL_ALM :BOOL := false; // 1 = LL alarm active

QSP_HLM {S7_visible:='false';
  S7_dynamic:='true'} : BOOL := 0; // 1=Setpoint Output High Limit Active
QSP_LLM {S7_visible:='false';
  S7_dynamic:='true'} : BOOL := 0; // 1=Setpoint Output Low Limit Active
Q_SP_OP {S7_visible:='false';
  S7_dynamic:='true';
  S7_m_c:='true'} : BOOL := 0; // Status: 1=Operator May Enter Setpoint
QOP_ERR {S7_visible:='false';
  S7_dynamic:='true'} : BOOL := 0; // 1=Operator Error
QMSG_ERR {S7_visible:='false';
  S7_dynamic:='true'} : BOOL := 0; // ALARM_8P: Error Output
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
SUPP_OUT :REAL := 0; // Output value for sample delay
SP {S7_dynamic:='true';
  S7_m_c:='true'} : REAL := 0; // Active Setpoint
END_VAR

```

```
VAR_IN_OUT
PV_IN {S7_dynamic:='true';
      S7_m_c:='true';
      S7_unit:='%'} : REAL := 0;      // Process Value (to AUX_PR04 of Message)

SP_OP {S7_visible:='false';
      S7_link:='false';
      S7_m_c:='true';
      S7_shortcut:='Setpoint';
      S7_unit:''} : REAL := 0;      // Operator Input Setpoint

// freibelegbare Meldebegleitwerte des ALARM_8P

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
```

1.2.4.2 Lokale Variablen

Zusätzliche Variablen, d.h. solche, 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

Statische Variablen behalten im Gegensatz zu temporären Variablen ihren Wert über mehrere Aufrufe des Bausteins hinweg, solange, bis 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 **Multiinstanz**-Baustein 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.

Falls Sie Parameter des aufgerufenen Bausteins nach außen sichtbar und verschaltbar machen wollen, müssen Sie diese im Bausteinalgorithmus von bzw. in 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, wie z.B. SFC6 (RD_SINFO) oder SFB0 (CTU) werden beim Compilieren des aufrufenden Bausteins automatisch in der Standard Library gesucht und in Ihr S7-Programm eingefügt.

Aufgerufene FBs werden beim Einfügen des aufrufenden Bausteins in einen CFC-Plan in den Bausteinordner kopiert, falls Sie sich in der selben Bibliothek befinden wie der aufrufende Baustein. Ansonsten 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 bedeutet dies, dass Sie hier neben den reinen technologischen Funktionen des Bausteins noch die Eigenschaften realisieren können, 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

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 durchführen muss. Falls dies bei Ihrem Baustein so ist, müssen Sie einen Erstlaufteil implementieren.

Dazu definieren Sie einen Variable vom Datentyp BOOL (z.B. sbRESTART). Sie können die Variable als statische Variable realisieren.

Da nicht gewährleistet werden kann, dass Ihr Baustein nur bei Neustart das erste Mal durchlaufen wird (z.B. beim Neuladen des Bausteins im Betriebszustand RUN der CPU), müssen Sie den Erstlaufteil typischerweise in den zyklischen Teil Ihres Bausteins integrieren. Dadurch können Sie die Bearbeitung des Erstlaufs, falls notwendig, 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

Falls Ihr Baustein in einer äquidistanten Zeitalarzebene bearbeitet wird, und er zur Durchführung zeitabhängiger Tätigkeiten die Länge des Zeitintervalls auswerten muss (z.B. Reglerbausteine), definieren sie einen Eingangsparameter (z.B. SAMPLE_T) vom Datentyp REAL, an dem die Länge des Zeitintervalls angegeben werden kann.

Diesen Parameter müssen Sie je nach Weckalarm-OB, in dem Ihr Baustein aufgerufen wird, umparametrieren. 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 für den aufrufenden OB passenden Wert gesetzt. Eine eventuelle Untersetzung wird dabei ebenfalls berücksichtigt. Sie sollten den Parameter dann auch mit den Systemattributen **S7_visible** und **S7_link** versehen und diese auf 'false' setzen. Damit wird der Parameter im CFC unsichtbar bzw. nicht verschaltbar und damit verhindert, dass sein Wert vom Anwender versehentlich geändert wird.

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

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; // sample time [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; // Output value for sample delay
    ....
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

Beim Auftreten eines asynchronen Ereignisses wie Neustart oder auch Ziehen/Stecken, Rackausfall und dergleichen, wird von der AS ein asynchroner OB aufgerufen. Falls 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 SFC6 (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 Ablauebene
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 aber 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 OB80 (Zeitfehler).

Tabelle 1-4: Zuordnung der Elemente der Startinfo TOP_SI zu den temporären Variablen des OB80

TOP_SI / STARTUP_SI		OB80	
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 SFC6 gelesenen Startinfo enthalten.
 - PCS7-konforme Bausteine werden nicht in den Wiederanlauf (OB 101) 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
//*****

    // 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

```

1.6 Bedienen, Beobachten und Melden

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

Bedienungen

Falls 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**).

Falls 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.

Alle Bedienfunktionen sollten über die Bedienbausteine der Bibliothek "PCS 7 Library Technological Blocks" und deren korrespondierende Bedienmethode auf der OS abgewickelt werden. 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 mittels der Multiinstanztechnik in Ihren Baustein einbauen.

Bei PCS 7 wird u.a. der **OP_A_LIM** (operation analog limited) eingesetzt.

Mit dem Baustein OP_A_LIM haben Sie ein begrenzendes Bedienverhalten gewählt. Alternativ können Sie den Baustein OP_A_RJC einsetzen, Dieser weist bei Grenzwertverletzung die Bedieneingabe ab. Wenn Sie keine Grenzwertprüfung benötigen, setzen Sie den Baustein OP_A ein.

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 der AS als auch auf der OS auf Fehler prüfen.

Für binäre Bedieneingaben stehen die Bausteine OP_D (FB 48), OP_D3 (FB 49) und OP_TRIG (FB 50) der PCS 7 Library "Technological Blocks" zur Verfügung (weitere Informationen: siehe Online-Hilfe).

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  
    SP_OP := OP_A_LIM_1.U;          // Sollwert zurückschreiben  
  
    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 des eingebauten Alarm-Bausteins bestimmt.

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

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

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'. Das gilt auch für das Melden. Damit der PCS 7-Meldedialog aufgerufen wird, tragen Sie im Bausteinkopf das Attribut S7_alarm_ui := '1' ein (bei 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 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' (bzw. gemäß dem eingebauten Meldungsbaustein) ein.

Der Eingang soll im CFC nicht sichtbar, nicht verschaltbar und nicht parametrierbar sein, um die vom System vergebenen Daten nicht versehentlich zu verändern.

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-Meldedialog ('0' = Standard-Dialg
}
AUTHOR:          ABC
NAME:             CONTROL
VERSION:         '0.01'
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; // Message ID
...
END_VAR

```

Die nicht im Baustein benötigten Eingänge des ALARM-Bausteins können auf das Baustein-Interface gelegt werden, um dem späteren Anwender weitere Möglichkeiten für Meldungen zu geben. Diese Meldungen werden ohne weitere Vorkehrungen wie Leittechnikmeldungen 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

Im folgenden Beispiel sehen Sie die Definition des ALARM_8P:

```

//*****
// Deklarationsteil: Bausteinparameter
//*****
....
// freibelegbare Meldebegleitwerte des ALARM_8P
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
....

//*****
// 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 Step Number (wegen I/O Begleitwert ALARM_8P)
sdBA_ID := BA_ID; // Batch ID (wegen I/O Begleitwert 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:= M_SUP_AH AND QH_ALM, // zu überw. Signal 1 -> Meldung Alarm oben
SIG_2:= 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 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 := PV_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

Falls 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 derer keine Meldung generiert werden soll. Sie müssen dann im Bausteinalgorithmus die Anzahl der Aufrufe zählen und erst nach Ablauf der parametrisierten Zyklen freigeben. Der folgende Ausschnitt des Beispiel-Bausteins zeigt dieses Verfahren.

```

//*****
// Deklarationsteil: Bausteinparameter
//*****
VAR_INPUT
...
  H_ALM {S7_m_c := 'true';
        S7_visible:= 'false';
        S7_link := 'false'
        } :REAL :=100;           // upper alarm limit (Vorbesetzung 100)
  L_ALM {S7_m_c := 'true';
        S7_visible:= 'false';
        S7_link := 'false'
        } :REAL := 0;           // lower alarm limit (Vorbesetzung 0)
...
  RUNUPCYC {S7_visible:= 'false';
            S7_link:= 'false'} :INT := 3;    // Number of Run Up Cycles
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 (LMN > H_ALM) THEN    // Wenn die Stellgröße die obere Alarmgrenze verletzt
    QH_ALM := 1;          // Fehlerausgang setzen
    QL_ALM := 0;          // Fehlerausgang rücksetzen

  ELSIF (LMN < L_ALM) THEN // Wenn die Stellgröße 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

Sollen im Bedarfsfall, z.B. bei vorhersehbaren Meldungen, bestimmte Meldungen unterdrückt werden, 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_SUP_AH für die Unterdrückung eines einzelnen Alarms verwendet:

```

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

VAR_INPUT
.....
    // Unterdrückung des ALARM HIGH
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      // suppress alarm low

    // 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;     // suppress alarm high
...
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:= M_SUP_AH AND QH_ALM,     // zu überw. Signal 1 -> Meldung Alarm oben
SIG_2:= 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 überwachendes Signal 4
.....
.....

```

1.6.3 Quelle übersetzen

Nach der Programmierung müssen Sie die Quelle mit dem SCL-Compiler übersetzen. Wählen Sie "Datei > Übersetzen" oder das Symbol für Übersetzen in der Funktionsleiste. Nach fehlerfreier Übersetzung ist der Baustein FB 501 im Bausteinordner des S7-Programms enthalten.

Weitere Informationen erhalten Sie im Handbuch "S7-SCL für S7-300 und S7-400".

1.7 Meldungsprojektion

Allgemeines

Für einen Baustein, der 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, falls sich eines der Signale geändert hat.

Meldungsprojektion im SIMATIC Manager

Den EV_ID können Sie im SIMATIC Manager über den Dialog **Edit > Special Object Properties > Message** bearbeiten.

Dabei können Sie einzelne Bestandteile der Meldungen (z.B. Meldetext, Meldeklasse usw.) gegen Änderung an anderer Stelle verriegeln, d.h. Sie können verhindern, dass bei einem Einbau Ihres Bausteins in einen CFC-Plan diese Meldung an der Bausteininstanz veränderbar ist.

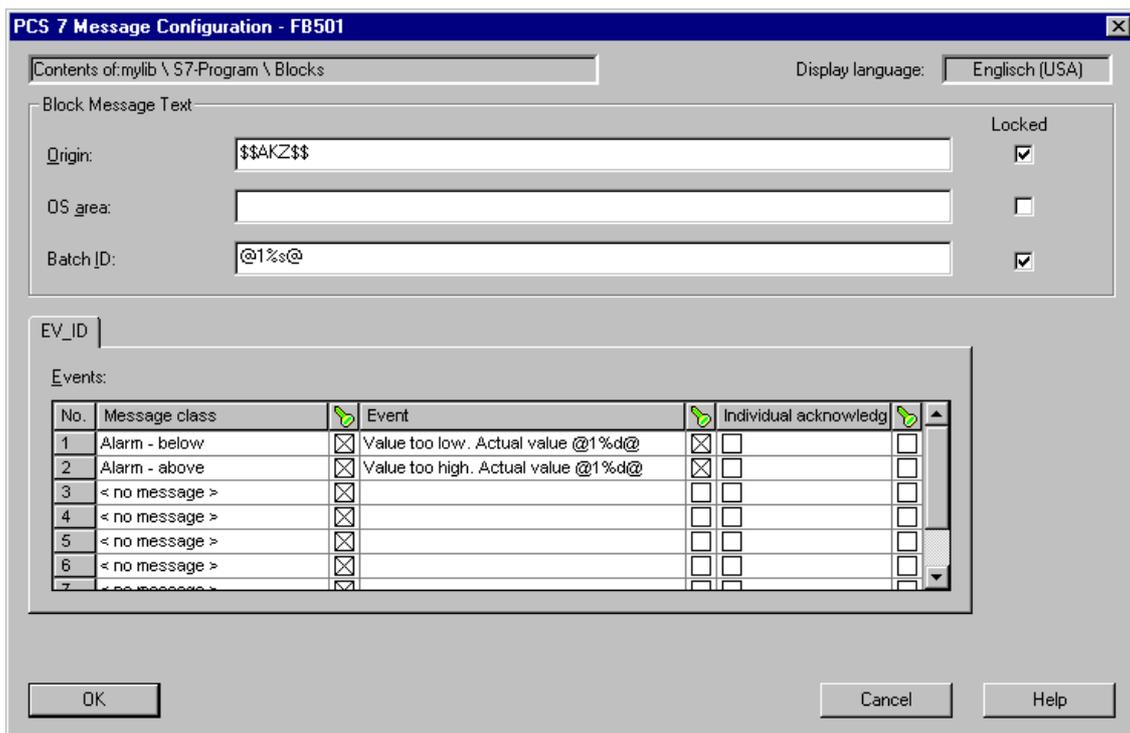


Bild 1-7 : Meldungsprojektion im SIMATIC Manager

Zuerst müssen Sie die Texte eingeben, die für alle Meldungen dieses Bausteins gelten. Die einzelnen Texte entsprechen dabei den Anwendertextblöcken im AlarmLogging von WinCC.

Origin:

Hier können Sie die Herkunft der Meldung angeben.

Geben Sie die Herkunft als Schlüsselwort `$$AKZ$$` ein, wird beim Transferieren der Daten durch die AS-OS-Verbindungsprojektierung der Pfad des Hierarchy-ordners, der Planname und der Bausteinname ermittelt und in den OS-Meldetexten abgelegt.

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

OS area:

Hier können Sie die Bereichszuordnung der Meldung angeben.

Geben Sie den Bereich als Schlüsselwort `$$AREA$$` ein oder tragen Sie nichts ein, wird beim Transferieren der Daten durch die AS-OS-Verbindungsprojektierung das entsprechende Attribut des Hierarchy-Ordners ausgewertet und in den OS-Meldetexten abgelegt.

Batch ID:

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

Geben Sie die Batch-Kennung ein, wird beim Transferieren der Daten durch die AS-OS-Verbindungsprojektierung 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. Falls Ihr Baustein für das S7-Optionspaket BATCH *flexible* geeignet sein soll, müssen Sie hier `@1%s@` eintragen. Dadurch wird die Meldung mit der BATCH-Chargenbezeichnung als ersten Meldebegleitwert versehen (vgl. Kapitel 1.8). Falls Sie BATCH *flexible* 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 "Message class" anklicken, wird diese zum Kombinationsfeld und Sie können die Meldeklasse auswählen. Nicht belegte Meldungen müssen die Meldeklasse "< no message >" erhalten. Genaueres über die Behandlung von Meldungen können Sie der Dokumentation zu WinCC entnehmen.

Tragen Sie in der Spalte "Event text" eine Beschreibung der Fehlerursache ein (maximal 40 Zeichen inklusive möglicher Begleitwerte) und in der Spalte "Individual acknowledgement", ob die Meldung einzeln quittiert werden muss (falls das Auswahlkästchen angewählt ist) oder ob sie per Sammelquittierung quittiert werden kann.

In der Spalte "Locked" (grünes Schlüsselsymbol) definieren Sie, ob der Meldetext vom Anwender des Bausteins geändert werden darf (Auswahlkästchen nicht angewählt) oder gesperrt ist (Auswahlkästchen angewählt).

Begleitwerte für Meldungen

Soll eine Meldung zusätzliche Informationen (z.B. Messwerte) an die OS übertragen, müssen Sie zum Melden einen ALARM-Baustein verwenden, der Ihnen 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@

Sprachen

Falls Sie Ihre Meldungen in verschiedenen Sprachen (je nach Sprachauswahl im ALARM Logging von Win CC) ausgeben wollen, müssen Sie sie für jede gewünschte Sprache projektieren. Dazu wählen Sie im SIMATIC Manager den Dialog "**Options > Translating Text > User texts**" aus. Tragen Sie in der angebotenen Tabelle die Texte in den jeweiligen Fremdsprachenspalten ein.

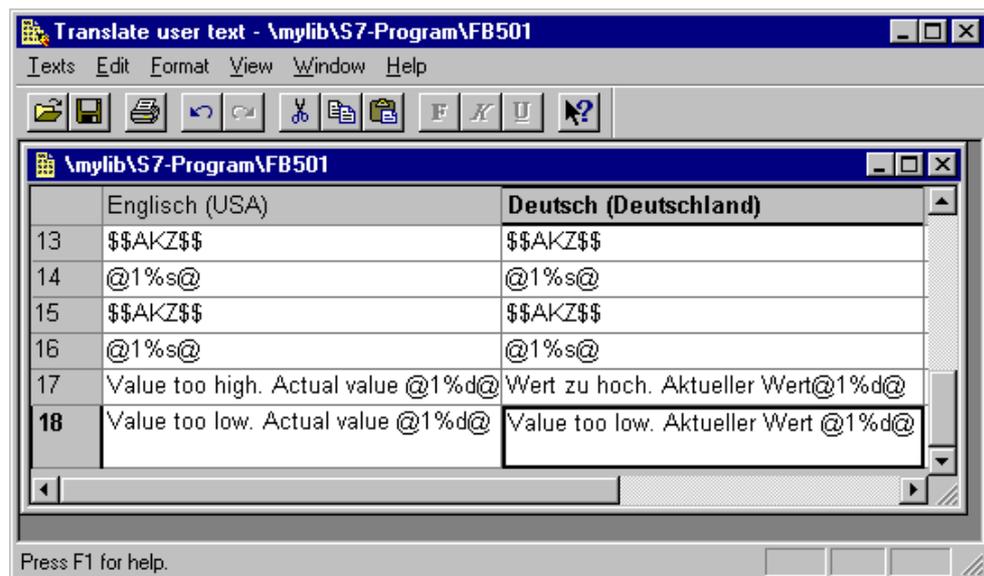


Bild 1-8: Übersetzung der Meldetexte

1.8 Anbindung von Batch *flexible*

Falls Sie Ihren Bausteine mit dem S7-Optionspaket "BATCH *flexible*" verwenden wollen, müssen Sie folgende Eingangs- bzw. Durchgangparameter definieren:

Parametername	Bedeutung	Parametertyp	Datentyp
BA_EN	BATCH-Belegfreigabe	INPUT	BOOL
BA_NA	BATCH-Chargenbezeichnung	INPUT	STRING[16]
BA_ID	laufende Chargennummer	INPUT	DWORD
OCCUPIED	BATCH-Belegkennung	INPUT	BOOL
STEP_NO	BATCH-Schrittnummer	INPUT	WORD

Ausschnitt des Beispiel-Bausteins:

```

//*****
// Deklarationsteil: Bausteinparameter
//*****
VAR_INPUT
....
        // Parameter für BATCH flexible
STEP_NO {S7_visible := 'false';
        S7_m_c := 'true'}
        :WORD; // Batch Step Number
BA_ID {S7_visible := 'false';
        S7_m_c := 'true'}
        :DWORD; // Batch ID
BA_EN {S7_visible := 'false'; // Parameter im CFC-Plan unsichtbar
        S7_m_c := 'true' // Parameter ist B&B-fähig, wird aber nur
        // auf binäre Änderung überwacht
        } :BOOL := 0; // Batch Enable

BA_NA {S7_visible := 'false';
        S7_m_c := 'true'}
        :STRING[16] := ''; // Batch Name

OCCUPIED {S7_visible := 'false';
        S7_m_c := 'true'}
        :BOOL := 0; // Occupied by Batch
....
END_VAR

```

Falls Sie in einem solchen Baustein Meldungen generieren wollen, müssen Sie dabei die Eingänge BA_NA, STEP_NO und BA_ID (in dieser 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

Im Unterschied zur Programmierung mit SCL, bei der Variablen deklariert und Zuweisungen ausprogrammiert werden, basiert CFC auf der Verschaltung von grafischen Objekten. D.h. Sie können damit durch Platzieren und Verschalten von bereits vorhandenen Bausteinen neue Bausteine entwickeln. Hierbei handelt es sich also um eine typische Anwendung des Multiinstanz-Verfahrens.

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

1.9.1 Beispiel: CONTROL2

Der fertiggestellte Beispiel-Baustein "CONTROL" soll um einen Multiplizierer erweitert werden. Mit diesem wird durch die Multiplikation zweier Eingangswerte (IN1 und IN2) der Istwert gebildet. Der erweiterte Baustein soll als CONTROL2 (FB601) erzeugt werden.

Vorgehensweise:

- Öffnen Sie einen neuen CFC-Plan und platzieren Sie darin den Beispiel-Baustein **CONTROL**.
- Aus der CFC Library\ELEMENTA ziehen Sie mit Drag&Drop einen Multiplizierer **MUL_R** (FC63) in den Plan.
- Verschalten Sie den Ausgang "OUT" des **MUL_R** mit dem Istwert (Parameter "PV_IN") des Beispiel-Bausteins.
- Öffnen Sie im Plan die Sicht der Plananschlüsse "Chart Inputs/Outputs" und selektieren Sie im Fenster "Interface" das Symbol "IN".
- Verschalten Sie die Eingänge "IN1" und "IN2" des **MUL_R** mit den Plananschlüssen: Mit Drag&Drop ziehen Sie den Bausteinanschluss auf den Plananschluss (rechtes Fenster).
- Verschalten Sie alle Ein- und Ausgänge des Beispiel-Bausteins (außer dem bereits verschalteten Istwert) mit den Plananschlüssen des CFC-Plans.
- Compilieren Sie über den Dialog "*Chart > Compile > Chart as Block Type*" den CFC-Plan als Baustein.
 - Im Register: "General" tragen Sie die FB-Nummer ein (hier 601). Tragen Sie danach in den entsprechenden Feldern die weiteren Eigenschaften ein: Family, Author, Version. Im Feld Name (header) ist bereits der Name des CFC-Plans vorbesetzt.
 - Im Register: "Attributes" geben Sie die gewünschten Baustein-Attribute und Systemattribute an. Das Systemattribut "S7_tasklist" müssen Sie hier nicht angeben (siehe nachfolgende Einbauregel).
 - Starten Sie die Übersetzung mit "OK".

Einbauregel:

Der CFC-Bausteintyp wird in jeden OB eingebaut, 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 jedoch nur in den OBs aufgerufen, die in ihrer eigenen Taskliste enthalten sind. Im hier angegebenen Beispiel heißt dies 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 OB80 und OB100 wird aber nur der **CONTROL** aufgerufen, nicht der **MUL_R**.

1.10 Namenskonventionen und Nummernbereich

Nummernbereich

Um Nummernkonflikte mit den von Siemens gelieferten leittechnischen PCS 7-Bausteinen zu vermeiden, sollten Sie die Nummerierung Ihrer Bausteine ab der Nummer 501 beginnen. Bitte 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:                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';      // Bausteins ist bedien- und beobachtbar
S7_alarm_ui:= '1';        // Einstellung PCS7-Meldedialog ('0'=Standard-Meldedialog)
}
AUTHOR:      ABC
NAME:        CONTROL
VERSION:     '0.01'
FAMILY:     XYZ
KNOW_HOW_PROTECT

//*****
// 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; // sample time [s] (Vorbesetzung 1 Sek)

  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; // lower alarm limit (Vorbesetzung 0)

  H_ALM {S7_m_c := 'true';
         S7_visible:= 'false';
         S7_link := 'false'} :REAL :=100; // upper alarm limit (Vorbesetzung 100)

  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 // suppress alarm low

  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; // suppress alarm high

  SP_OP_ON {S7_visible:= 'false';
            S7_dynamic:= 'true' // CFC in Test/IBS: Anzeige aktueller Wert in AS)
            } :BOOL := 1; // Enable l=Operator for Setpoint Input

```

```

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; // Enable 1=Bumpless for Setpoint On
SP_EXTON {S7_visible:='false';
  S7_dynamic:='true' // CFC in Test/IBS: Anzeige aktueller Wert in AS)
}
      :BOOL := 1; // 1: Select SP_EXT

SP_EXT {S7_dynamic:='true'}

      :REAL := 0; // External Setpoint
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; // Setpoint High Limit

SP_LLM {S7_visible:='false';
  S7_link:='false';
  S7_m_c:='true';
  S7_shortcut:='SP low limit';
  S7_unit:=''
      :REAL := 0; // Setpoint Low Limit

GAIN {S7_link:='false';
  S7_edit:='para'; // Parametrierung im Import-Export-Assistenten (IEA)
  S7_m_c:='true';
  S7_shortcut:='Gain';
  S7_unit:=''
      :REAL := 1; // Proportional Gain

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; // Message ID

// Parameter für BATCH flexible
STEP_NO {S7_visible := 'false';
  S7_m_c := 'true'} :WORD; // Batch Step Number
BA_ID {S7_visible := 'false';
  S7_m_c := 'true'} :DWORD; // Batch ID
BA_EN {S7_visible := 'false';
  S7_m_c := 'true' // Parameter im CFC-Plan unsichtbar
}
      :BOOL := 0; // Batch Enable

BA_NA {S7_visible := 'false';
  S7_m_c := 'true'} :STRING[16] := ''; // Batch Name

OCCUPIED {S7_visible := 'false';
  S7_m_c := 'true'} :BOOL := 0; // Occupied by Batch

RUNUPCYC {S7_visible:='false';
  S7_link:='false'} :INT := 3; // Number of Run Up Cycles
SUPPTIME :REAL := 0; // sample delay
SUPP_IN :REAL := 0; // Input value for sample delay
END_VAR

```

```

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

QH_ALM :BOOL := false;                 // 1 = HH alarm active

QL_ALM :BOOL := false;                 // 1 = LL alarm active

QSP_HLM {S7_visible:='false';
         S7_dynamic:='true'} : BOOL := 0; // 1=Setpoint Output High Limit Active

QSP_LLM {S7_visible:='false';
         S7_dynamic:='true'} : BOOL := 0; // 1=Setpoint Output Low Limit Active

Q_SP_OP {S7_visible:='false';
         S7_dynamic:='true';
         S7_m_c:='true'} : BOOL := 0;    // Status: 1=Operator May Enter Setpoint

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

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

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

SUPP_OUT :REAL := 0; // Output value for sample delay
SP {S7_dynamic:='true';
   S7_m_c:='true'} : REAL := 0; // Active Setpoint

END_VAR

VAR_IN_OUT
PV_IN {S7_dynamic:='true';
       S7_m_c:='true';
       S7_unit:='%'} : REAL := 0; // Process Value (to AUX_PR04 of Message)

SP_OP {S7_visible:='false';
       S7_link:='false';
       S7_m_c:='true';
       S7_shortcut:='Setpoint';
       S7_unit:''} : REAL := 0; // Operator Input Setpoint

// freibelegbare Meldebegleitwerte des ALARM_8P

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

```

```

/*****
// 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

swSTEP_NO :WORD;         // Batch Step Number
sdBA_ID   :DWORD;       // Batch ID

sbyBA_NA  :ARRAY[1..16] OF BYTE := 16(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 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

/*****
// 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;
  siRUNUPCYC := 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
  SP_OP := OP_A_LIM_1.U;    // Sollwert zurückschreiben
  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 (LMN > H_ALM) THEN // Wenn die Stellgröße die obere Alarmgrenze verletzt
    QH_ALM := 1; // Fehlerausgang setzen
    QL_ALM := 0; // Fehlerausgang rücksetzen

  ELSIF (LMN < L_ALM) THEN // Wenn die Stellgröße 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;

//*****
// 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 Step Number (wegen I/O Begleitwert ALARM_8P)
sdBA_ID := BA_ID; // Batch ID (wegen I/O Begleitwert 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:= M_SUP_AH AND QH_ALM, // zu überw. Signal 0 -> Meldung Alarm oben
  SIG_2:= M_SUP_AL AND QL_ALM, // zu überw. Signal 1 -> Meldung Alarm unten
  SIG_3:= 0, // zu überw. Signal 2 -> keine Meldung
  SIG_4:= 0, // zu überwachendes Signal 3
  SIG_5:= 0, // zu überwachendes Signal 4
  SIG_6:= 0, // zu überwachendes Signal 5
  SIG_7:= 0, // zu überwachendes Signal 6
  SIG_8:= 0, // zu überwachendes Signal 7
  SD_1 := sbyBA_NA, // Begleitwert 1
  SD_2 := swSTEP_NO, // Begleitwert 2
  SD_3 := sdBA_ID, // Begleitwert 3
  SD_4 := PV_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 erstellen

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
(angeboten vom A&D Trainingscenter unter ST-BWINCCS)
- SIMATIC WinCC Offenheit N
(angeboten vom A&D Trainingscenter unter ST-BWINCCN)

2.1 Erstellungsweg

Bei der Erstellung eines Bildbausteins hat sich die folgende Vorgehensweise als vorteilhaft erwiesen:

- Entwurf des Bildbausteins (Abschnitt 2.1.1)
- Projektierung des Bildbausteins (Abschnitt 2.1.2)
- Test des Bildbausteins (Abschnitt 2.1.3)

2.1.1 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- und Aus- und Durchgangparameter eines AS-Bausteins bedient und beobachtet werden können, wird bei der Erstellung des AS-Bausteins durch die Systemattribute festgelegt. Einzelheiten zu diesen Systemattributen können Sie im Kapitel "Aufbau eines AS-Bausteins" im Abschnitt "Deklarationsteil" nachlesen.

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? Gruppieren Sie dazu die einzelnen Parameter funktionsmäßig. 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 d.h. die Auswahl der Bildelemente, deren Benennung, die Parametrierung der Bildelemente und deren Position. Verwenden Sie immer Namen, die einen Bezug zum dargestellten Objekt haben und auch ausgesprochen werden können.

Beispiel: In einer Zustandsanzeige soll die Variable "OCCUPIED" dargestellt werden → nennen Sie diese Zustandsanzeige "OCCUPIED".

Diese Vorgehensweise erleichtert die Projektdokumentation und die Wartung.

2.1.2 Projektierung des Bildbausteins

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 Abschnitt 2.2 "Bildbausteinerstellung mit dem Faceplate Designer".

2.1.3 Test des Bildbausteins

Beim Test des Bildbausteins gehen Sie in 2 Schritten vor:

1. Prüfen Sie im WinCC Explorer die Eigenschaften der erstellten Bilder:

- 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 eventgetriggerten Skripte vorhanden?

2. Prüfen Sie in WinCC Runtime:

- 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?

2.2 Bildbausteinerstellung mit dem Faceplate Designer

Bestandteil des WinCC-Optionspakets "Advanced Process Control" ist der Faceplate Designer. Dieser enthält die Vorlagen zur PCS7-konformen Erstellung von Bildbausteinen.

Kompatibilität zu PCS7-Bildbausteinen

Die mit PCS 7 ausgelieferten Bildbausteine (.ocx Standard-Bildbausteine) können mit dem Faceplate Designer nicht verändert werden. Es besteht aber die Möglichkeit, neue Bildbausteine über den Faceplate Designer zu erstellen, die gleichnamig mit den Standard-Bildbausteinen sind (z.B.: MOTOR, VALVE, MEAS_MON, ...). Auf diese Weise werden die Standard-Bildbausteine überschrieben.

Die vorher im Prozessbild verwendeten "alten" .ocx-Bildbausteine (Symboldarstellung) müssen gegen die "neuen" Bausteinsymbole ausgetauscht werden, da sonst der richtige Bildbaustein nicht aufgerufen wird. Bereits eingebaute ocx-Bildbausteine werden nicht durch die "neuen" überschrieben.

Die neu erstellten Bildbausteine können bei der Projektierung und im Runtime genauso behandelt werden wie Standard-Bildbausteine.

Wurden irrtümlich Standard-Bildbausteine im Projekt überschrieben, so können Sie durch den Start des Split Screen Wizard die Standard-Bildbausteine aus der Bibliothek wieder ins Projekt kopieren.

Weitere Informationen

Weitere Informationen über die Komponenten und die Funktionalität des Faceplate Designers erhalten Sie über die WinCC-Online-Hilfe → WinCC Options → Faceplate Designer.

2.2.1 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 Objekten für die Bildbaustein-Erstellung
- Globale Skripte

Eine Auflistung aller Dateien finden Sie in der Online-Hilfe "WinCC Options > Faceplate Designer > Komponenten des Faceplate Designers > Dateiliste Faceplate Designer".

2.2.1.1 Bausteinsymbole



Die Bausteinsymbole sind im WinCC-Bild "@PCS7Typicals.PDL" enthalten, z.B.: Ventil, Antrieb, Messwert, Regler usw. und im Pfad "Siemens\WinCC\Options\PDL\Base_Data_Pool" abgelegt. Dieses befindet sich nach Durchlauf des Split Screen Wizards im Ordner "GraCS" des Projektverzeichnisses.

- Die Beispiel-Vorlagen sind beliebig änderbar in Form, Farbe, Gestaltung usw. und schnell anpassbar auch an projektspezifisch erstellte Bildbausteine.
- Die fertigen Aufrufskripte 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".

Hinweis

Für Änderungen ist das Bild unter dem Namen "@PCS7Typicals.PDL" zu speichern. Nach diesem Namen sucht die Suchmaschine aus der TH zuerst. Nur wenn dieses nicht gefunden wird, werden die Vorlagen aus "@PCS7Typicals.PDL" verwendet.

2.2.1.2 Vorlagenbilder

Die Bilder und Bitmaps befinden sich im Verzeichnis "WinCC\options\pd\FaceplateDesigner".

2.2.1.3 Objekt-Baukasten

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

Das Bild ist im Pfad "Siemens\WinCC\Options\PDL\FaceplateDesigner" abgelegt und wird beim Durchlauf des Split Screen Wizard in den Ordner "GraCS" des Projektverzeichnisses kopiert.

2.2.1.4 Globale Skripte

Die Bildbaustein-Aufrufe befinden sich als globale Skripte im Verzeichnis "WinCC\aplib\FaceplateDesigner".

2.2.2 Projektierungsschritte

Zur Erstellung des Bildbausteins:

- öffnen Sie im SIMATIC Manager die OS.
Der WinCC Explorer wird geöffnet.
- starten Sie im WinCC Explorer den Graphics Designer.
In den Projektdaten finden Sie die Vorlagen, um einen Bildbaustein zu erstellen. Die erforderlichen Vorlagenbilder sind im Pfad "Siemens/WinCC/options/pdl/FaceplateDesigner" abgelegt.
- Starten Sie vor der ersten Verwendung den Split Screen Wizard (in Base Data), damit die Dateien in den Ordner "GraCS" des Projektverzeichnisses kopiert werden.

Hinweise zur Projektierung

- Überschreiben Sie versehentlich eine Vorlagendatei (z.B.: @PG_%Type%_%View%.PDL), dann können Sie das Original aus dem Verzeichnis Siemens\WinCC\Options\Pdl\FaceplateDesigner kopieren.
- Es wird empfohlen, alle selbst erstellten Dateien für die Bildbausteine zunächst im GraCS-Verzeichnis des aktuellen Projekts abzulegen. Wenn die Dateien auch bei nachfolgendem Durchlauf des Split Screen Wizard nicht durch Originale überschrieben werden sollen, dann müssen sie in das Verzeichnis Siemens\WinCC\Options\Pdl\FaceplateDesigner kopiert werden.
- Soll ein Projekt auf einem anderen Rechner verwendet werden, dann kann im Projektverzeichnis\GraCS ein neuer Ordner mit dem Namen "FaceplateDesigner" erstellt werden. Die in diesem Ordner abgelegten Bildbausteine werden beim Start des Split Screen Wizard in das GraCS-Verzeichnis des Projektes kopiert. Bestehende, gleichnamige Dateien werden überschrieben.
- Die an eigenen Bildbausteinen projektierten Funktionen können bei Bedarf im Editor "Global Script" gegen Einsicht und Änderung geschützt werden. 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 geeigneter 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.
- In der Dynamik der Bildbausteine darf kein C-Skript mit fest kodiertem Instanznamen verwendet werden.
- In einem Bild, das zur Erstellung eines Bildbausteins dient, dürfen verschiedene Bildbausteintypen nicht gemischt verwendet werden, d.h. ein Bildbaustein, in dem z.B. eine Ventilsteuerung und eine Motorsteuerung enthalten sind, ist nicht erlaubt.

Weitere Informationen finden Sie in der WinCC Dokumentation unter "Projektierungshinweise, Tips und Tricks".

Vorgehensweise

Die folgenden Kapitel beschreiben am Beispiel des Bausteintyps MEAS_MON Schritt für Schritt den Weg zum fertigen Bildbaustein.

Die Bildbaustein-Erstellung geschieht in vier aufeinanderfolgenden Schritten:

1. Gruppenbildrahmen an Bildbausteintyp anpassen (Abschnitt 2.2.2.1)
2. Sichtenliste an Bildbausteintyp anpassen (Abschnitt 2.2.2.2)
3. Typ-/ und sichtspezifische Bilder bearbeiten (Abschnitt 2.2.2.3)
4. Kreisbildrahmen an Bildbausteintyp anpassen (Abschnitt 2.2.2.4)

2.2.2.1 Gruppenbildrahmen an den Bildbausteintyp anpassen

Das Bild "@PG_%Type%.PDL" stellt den Rahmen für die Gruppendarstellung eines Bildbausteins dar. Für die Anpassung an den Bausteintyp MEAS_MON gehen Sie wie folgt vor:

- Öffnen Sie im WinCC Graphic Designer das Vorlagebild "@PG_%Type%.PDL"
- Bildbausteintyp festlegen:
Selektieren Sie dazu das Objekt "Block Type" (%Type%) und öffnen Sie den Eigenschaftendialog (Register "Eigenschaften").
Objekteigenschaftenfenster → Eigenschaften → EA-Feld → Ausgabe/Eingabe → Ausgabewert → "%Type%".
Ersetzen Sie "%Type%" durch "MEAS_MON"
- Falls notwendig schalten Sie die Sammelanzeige "EventState" unsichtbar:
Objekteigenschaftenfenster → Eigenschaften → Sonstige → Anzeige → nein
Im Beispiel MEAS_MON ist dies nicht erforderlich.
- Speichern Sie das Bild unter dem Namen des Bausteintyps
"@PG_MEAS_MON.PDL"

2.2.2.2 Sichtenliste an den Bausteintyp anpassen

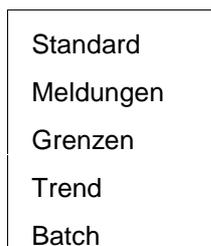
Das Bild "@PG_%Type%_VIEWLIST.PDL" enthält alle Sichten, die für die PCS 7 Standard-Bildbausteine V5.1 verwendet werden können.

Standard
Wartung
Meldungen
Parameter
Grenzen
Trend
Batch
StandardS

Für die Anpassung an den Bausteintyp MEAS_MON gehen Sie wie folgt vor:

- Öffnen Sie im WinCC Graphic Designer das Bild "@PG_%Type%_VIEWLIST.PDL".
- Entfernen Sie die nicht benötigten Sichten. Das sind beim MEAS_MON "Wartung", "Parameter" und "StandardS" (Markieren und löschen).
- Platzieren Sie die Einträge lückenlos in der gewünschten Reihenfolge: Markieren Sie dazu die Einträge in der Liste und verschieben Sie diese mit Hilfe der Pfeiltasten.
- Anpassen von Rahmen und Bildgröße:
 - Öffnen Sie für die Bildgröße den Eigenschaftendialog des Bild-Objekts "@PG_%Type%_VIEWLIST". Unter Eigenschaften/Bild-Objekt/Geometrie können Sie die erforderlichen Änderungen der Statik vornehmen (z.B.: Bildbreite 90, Bildhöhe 100).
 - Öffnen Sie nun den Eigenschaftendialog des Objekts "Comboframe". Unter Eigenschaften/Rechteck/Geometrie können Sie nun den Rahmen anpassen (wie Bildgröße).

Danach sieht das Bild wie folgt aus:



- Speichern Sie dieses Bild unter dem Bausteintypnamen, in unserem Beispiel unter "@PG_MEAS_MON_VIEWLIST.PDL".

2.2.2.3 Typspezifische und sichtspezifische Bilder bearbeiten

Das Bild "@PG_%Type%_%VIEW%.PDL" ist die Vorlage für eine Sicht des Bildbausteins.

Für die Melde-, Trend- und Batch-Sicht sind die Bilder "@PCS7_ALARM.PDL", "@PCS7_TREND.PDL" und "@PCS7_BATCH.PDL" vorhanden. Diese sind bei PCS7 für alle Bildbausteine identisch, eine Anpassung ist nur bei speziellen Anforderungen erforderlich. Für weitere Sichten ändern Sie den Platzhalter %VIEW% entsprechend. Die Sichten sind an das folgende Namensschema gebunden:

Verwendung	Platzhalter	Name
Grenzen	%VIEW%	= LIMITS
Parameter	%VIEW%	= PARAMETERS
Standard	%VIEW%	= STANDARD
Standard S	%VIEW%	= STANDARDS
Wartung	%VIEW%	= MAINTENANCE

Für die Anpassung an den Bausteintyp MEAS_MON gehen Sie wie folgt vor:

- Öffnen Sie im WinCC Graphics Designer das Bild "@PG_%Type%_%View%.PDL"
- Speichern Sie dieses Bild für die **Standardsicht** unter "@PG_MEAS_MON_STANDARD.PDL".
- Öffnen Sie im WinCC Graphic Designer das Bild "@PG_%Type%_%View%.PDL"
- Speichern Sie dieses Bild für die **Grenzsicht** unter "@PG_MEAS_MON_LIMITS.PDL".

Änderung der Bildgröße

Bei Änderungen der Größe, die Sie über den Eigenschaften-Dialog (Eigenschaften > Geometrie) vornehmen, müssen Sie folgendes beachten:

- Die Sichten der Bilder werden im Gruppenbild und im Kreisbild ausgegeben.
- Für Alarm, Batch und Trend werden derzeit universal gültige Bilder verwendet. Namen s.o.
- Ändern Sie die Größe der anderen Sichten, so müssen Sie ggf. auch diese Universal-Bilder ändern.

2.2.2.4 Kreisbildrahmen an den Bausteintyp anpassen

Das Bild "@PL_%Type%.PDL" stellt den Rahmen für die Kreisbilddarstellung eines Bildbausteins dar. Für die Anpassung an den Bausteintyp MEAS_MON gehen Sie wie folgt vor:

- Öffnen Sie im WinCC Graphic Designer das Bild "@PL_%Type%.PDL"
- Ersetzen Sie im EA-Feld vom Objektnamen "BlockType" den Text "%Type%" durch den Bausteintyp-Namen:
Objekteigenschaftenfenster → Eigenschaften → EA-Feld → Ausgabe/Eingabe → Ausgabewert → "%Type%"
Ersetzen Sie "%Type%" durch "MEAS_MON"
- Falls notwendig schalten Sie die Sammelanzeige "EventState" unsichtbar:
Objekteigenschaftenfenster → Eigenschaften → Sonstige → Anzeige → nein
Im Beispiel MEAS_MON ist dies nicht erforderlich.
- Eintragen der gewünschten Sicht im Objekt "@VIEWS"

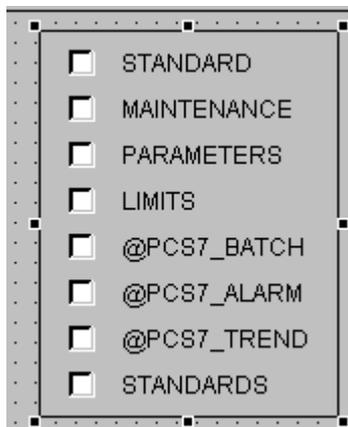


Bild 2-1 : Check-Box "@VIEWS"

- Objekteigenschaftenfenster → Eigenschaften → Check-Box → Ausgabe/Eingabe. Doppelklick auf "Selektierte Felder".
- Klicken Sie auf die erste gewünschte Sicht (z.B. "1" für STANDARD).
- Klicken Sie für die weiteren gewünschten Sichten mit gedrückter CTRL-Taste den Index der Sicht an.
- Bestätigen Sie die Auswahl mit der OK-Taste.
- Überprüfen Sie, ob die gewünschten Sichten als "ausgewählt" gekennzeichnet sind (Häkchen im entsprechenden Auswahlkästchen).
- Entfernen Sie die nicht benötigten Sichten, das sind beim MEAS_MON die Elemente "MAINTENANCE", "PARAMETERS" und "STANDARDS".
- Platzieren Sie die Elemente lückenlos in der gewünschten Reihenfolge im Bild und passen Sie die Bildgröße an die Größe der Bildfenster an.
- Speichern Sie das Bild unter dem Namen des Bausteintyps "@PL_MEAS_MON.PDL".

2.2.2.5 Trendsicht projektieren

In den bisherigen Schritten wurden ausschließlich typspezifische Bilder erstellt. Für die Trendsicht sind zusätzlich messstellenspezifische Projektierungsschritte erforderlich. Die für alle PCS7-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:

- Öffnen Sie den WinCC-Editor "Tag Logging"
- Legen Sie mit Hilfe des Archiv Wizards ein Variablen-Archiv für die Trendsicht an
- Legen Sie die Variablen für dieses Archiv an

Eine ausführliche Anleitung zu den o.a. Schritten finden Sie in der WinCC-Online-Hilfe unter dem Index "Tag Logging".

- Öffnen Sie im Graphics Designer das Bild "@CONL1_.PDL"
- Konfigurieren Sie das TlgOnlineTrend-Objekt "Control1" entsprechend ihren Anforderungen; für weitere Informationen nutzen Sie die Taste "Hilfe".
- Speichern Sie das Bild unter dem Namen "@CONL1_<tagname>.PDL". Der <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.

Beispiel für den Fensternamen des Bausteins MEAS_MON_1:
@CONL1_Messungen_Bedienungen_MEAS_MON_1.PDL

2.2.2.6 Weitere Sichten projektieren

Im Projektierungsschritt 3 (Abschnitt 2.2.2.3) wurden bereits die Sichten für einen Bildbausteintyp angelegt. In die vorgefertigten Sichten werden jetzt entsprechend dem Bildentwurf die Bildelemente eingefügt und parametrisiert. Die fertigen Beispiele "@PG_MEAS_MON_STANDARD.PDL" und "@PG_MEAS_MON_LIMITS.PDL" finden Sie auf der CD. Deshalb ist an dieser Stelle nur das grundsätzliche Vorgehen skizziert. Welche Objekte in die Bilder eingebaut sind, sowie die Eigenschaften der Objekte, können Sie über die "Objekteigenschaften" ermitteln.

Für die Sicht "Grenzen" des MEAS_MON gehen Sie so vor:

- Öffnen Sie im WinCC Graphics Designer das Bild "@PG_MEAS_MON_LIMITS.PDL".
- Öffnen Sie im WinCC Graphics Designer das Bild "@PCS7Elements.PDL".
- Ordnen Sie die Bilder nebeneinander an: Fenster > Nebeneinander.
- Kopieren Sie die benötigten Bildelemente vom Bild "@PCS7Elements.PDL" in das Bild "@PG_MEAS_MON_LIMITS.PDL".
- Bauen Sie weitere WinCC-Objekte wie Rechtecke usw. in Ihr Bild ein.
- Vergeben Sie sinnvolle (objektbezogene) Objektnamen.
- Stellen Sie die Position für die einzelnen Objekte ein.
- Verschalten Sie die dynamischen Attribute der Bildelemente mit den Parametern des AS-Bausteintyps. Bei den Objekten "PCS7_MSG_LOCK" und "PCS7_OCCUPIED" ist die Verschaltung bereits projektiert.
- Bauen Sie die "Bedienberechtigungsketten" auf. In den Vorlagenbildern befinden sich die Objekte "@Level5" und "@Level6". Beide Objekte werden von der Steuerung des überlagerten Bildes "@PG_MEAS_MON.PDL" bzw. "@PL_MEAS_MON.PDL" beeinflusst.

Ist ein Benutzer mit der Bedienberechtigungsstufe 5 eingeloggt, wird für das Objekt "@Level5" die Hintergrundfarbe "Weiß" eingestellt und die Bedienfreigabe auf "ja" gesetzt. Hat der Benutzer diese Berechtigung nicht, ist die Hintergrundfarbe "Grau" und die Bedienfreigabe "nein".

Das Objekt "@Level6" dient der Steuerung der Bedienberechtigungsstufe 6. Beide Elemente dienen als "Anker" für alle bedienbaren Objekte in Ihrem Bild. Von den Ereignissen "Hintergrundfarbe" und "Bedienberechtigung" werden Direktverbindungen von einem Element zum nächsten aufgebaut. Für Binärvariable ist eine Direktverbindung nicht erforderlich.

Im Bild "@PG_MEAS_MON_LIMITS.PDL" sehen die Ketten für die Bedienberechtigungsstufe 6 wie folgt aus:

"Bedienberechtigung": @Level6 > U_AH > U_WH > U_WL > U_AL > MO_PVHR > MO_PVLR > M_SUP_AH > M_SUP_WH > M_SUP_WL > M_SUP_AL

"Hintergrundfarbe": @Level6 > U_AH > U_WH > U_WL > U_AL > MO_PVHR > MO_PVLR

- Speichern Sie das Bild.

Für die Sicht "Standard" des MEAS_MON verfahren Sie entsprechend.

2.2.2.7 Eigene Sicht für Bausteintyp erstellen

Nachfolgend ist beschrieben, wie Sie aus einer Sicht, die bisher für alle PCS 7-Bausteine verwendet wurde, eine neue Sicht erstellen, die nur für einen Bausteintyp gilt. Bei diesem Beispiel handelt es sich um eine Alarm-Sicht.

- Speichern Sie das Originalbild "@PCS7_Alarm" unter "@PG_%Type%_Alarm" (Type = MEAS_MON).
- Kopieren Sie in Ihr Bild die beiden Objekte "@Level5" und "@Level6" aus einer bereits bestehenden Sicht, z.B. aus "@PG_MEAS_MON_Standard.PDL".

Sollten durch das Kopieren die "@" verlorengegangen sein, so sind diese wieder vor die Namen der beiden Objekte Level5 und Level6 einzufügen.

- Folgender Script ist hier für "@Level5" unter der C-Aktion einzubringen:

```
double dToolbarButtons;
if (value)
    dToolbarButtons = 62000;
else
    dToolbarButtons = 61952;
SetPropDouble(IpszPictureName, "AlarmList",
    "ToolbarButtons", dToolbarButtons);
```

Tip: Kopieren Sie dazu diesen Text aus dieser Dokumentation.

- Öffnen Sie das Bild "@PG_%Type%_VIEWLIST.PDL" und vergrößern Sie es so weit, dass eine neue Sicht als Text unten angefügt werden kann.
- Kopieren Sie ein Textfeld, z.B. "Standard", fügen Sie es ein und geben der Sicht einen neuen Namen.
Der Objektname muss mit dem neuen Sichtnamen übereinstimmen, hier "Alarm".
- Ändern Sie den zugehörigen Anzeigetext für die Viewlist; er kann vom Bildnamen abweichen und kann mehrsprachig eingegeben werden (siehe Graphics-Designer > Menu Ansicht > Sprache).
- Öffnen Sie das Kreisbild "@PL_%Type%.PDL" und vergrößern Sie die vorhandene Check-Box um ein zusätzliches Feld:
In den Eigenschaften "@Views" der Check-Box selektieren Sie "Geometrie" und erhöhen die "Anzahl Felder" um 1 (z.B. von 8 auf 9).
- Dieses neue Feld ist nach der neuen Sicht zu benennen:
Eigenschaften > Schrift > Index: die Nr. des hinzugefügten Feldes eingeben (hier: 9) und im Feld "Text" den Namen eintragen (hier: "Alarm").
- Damit das Feld später im Runtime sichtbar ist:
In den Eigenschaften "@Views" > Ausgabe/Eingabe und Doppelklick auf "Selektierte Felder". Wählen Sie in der Liste den Index "9" und schließen sie mit "OK". Hinweis: Mehrfachauswahl bei gedrückter CTRL-Taste.
- Für die neue Sicht im Kreisbildrahmen ein Bildfenster einbringen:
Kopieren Sie aus einer vorhandenen Sicht (z.B. STANDARD) das Objekt "Bildfenster" und geben Sie ihm einen neuen Namen (hier: "Alarm").

2.2.3 Dynamisierung von Bildbausteinen

Es gibt unterschiedliche Wege um eine Dynamisierung zu erreichen:

1. Die Extension der benötigten Variablen sind 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 eintragen.

2. Variable aus der Variablenliste

Der z.B. 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 Glühlampensymbol und wählen "Variable" aus.

Suchen Sie die gewünschte Variable und führen Sie darauf einen Doppelklick aus.

Jetzt steht in der Dynamisierung allerdings der vollständige Variablenname. Löschen Sie den Text bis auf die Extension, auch den Punkt (".").

Vorgang der Dynamisierung

- Öffnen Sie die Ansicht, die Sie dynamisieren wollen, z.B. "@PG_MEAS_MON_STANDARD.PDL".
- Meldezustandsanzeige unsichtbar schalten, wenn nicht benötigt:
Selektieren Sie dazu das Objekt "EventState" und öffnen Sie den Eigenschaftendialog.
Ändern Sie über Eigenschaften/Sammelanzeige/Sonstige das Attribut "Anzeige" auf "Nein".
- Objekte in Bild einbauen und parametrieren:
Für die Standard-Bildbausteine sind die benötigten Objekte vorbereitet und können direkt aus den Vorlagedateien in das Bild einkopiert werden, z.B. Balken aus dem Vorlagenbild "@PCS7Elements.PDL" kopieren und in das Bild einfügen.
Anschließend müssen Sie die Eigenschaften der Objekte verschalten. Sind die Zielbezeichnungen bekannt, dann können Sie diese direkt eingeben. Andernfalls können Sie die Verschaltung über den Variablenbrowser vornehmen. Beachten Sie, dass die Bezeichnung der AS-Instanz nicht im Variablennamen enthalten sein darf, sondern nur die Extension, z.B. im Variablennamen "Anlage1_Motor2.U_AH" muss die Bezeichnung der Instanz "Anlage1_Motor2." gelöscht werden (inkl. "."). Die Bezeichnung lautet dann "U_AH". Beachten Sie bei der Verwendung der Extension die Groß- und Kleinschreibung.
Die Aktualisierungszyklen sollten aufgrund besserer Performance möglichst identisch sein (Standard = 2 sec.).

- Ausgehend von den Objekten "@Level5" und "@Level6" eine Kette von Direktverbindungen für die einzelnen Berechtigungsstufen aufbauen:

Beispiel: Wenn Sie mehrere EA-Felder mit Berechtigungsstufe 6 schützen möchten, dann müssen Sie das Property "Bedienfreigabe" des Objekts "@Level6" per Direktverbindung mit dem Property "Bedienfreigabe" des ersten EA-Feldes verbinden:

- Markieren Sie das erste EA-Feld und öffnen Sie den Objekteigenschaften-dialog. Über die Registerkarte "Ereignis" können Sie nun die Direktverbindung zum Objekt "@Level6" projektieren.
- Das zweite EA-Feld müssen Sie nun per Direktverbindung auf das erste verbinden usw.

Bei der Projektierung ist zu beachten, dass neu hinzugefügte Bedienelemente grundsätzlich am Ende der Berechtigungskette eingefügt werden. Es wird empfohlen, die Reihenfolge der Berechtigungskette zu dokumentieren.

Der Wechsel der Hintergrundfarbe für die Bedienberechtigung kann auf die gleiche Weise verkettet werden.

- Speichern Sie das Bild unter dem entsprechenden Typnamen, hier: "@PG_MEAS_MON_STANDARD.PDL".

2.2.4 Sprachumschaltung

Die Vorlagen des Faceplate Designers sind dreisprachig angelegt (Deutsch/Englisch/Französisch) d.h. bei 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**.

3 Online-Hilfe erstellen

Voraussetzungen

Sie benötigen:

- den in WINDOWS-NT integrierten ASCII-Editor "Notepad" o.ä. zur Erstellung einer Registrierungsdatei.
- ein Erstellungswerkzeug für die Hilfethemen (z.B. "RoboHelp").

3.1 Aufbau der Hilfedatei

Falls 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, aus Übersichtlichkeitsgründen sollten Sie jedoch den Namen ihrer Bibliothek (bzw. einen gemeinsamen Namen Ihrer Bausteine) verwenden, z.B. "MYLIB__A.HLP".

Erstellen Sie für jeden Ihrer Bausteine ein eigenes Hilfethema (Topic). Anschließend müssen Sie für jedes Topic die Einsprungadresse (MAP-ID) zur Online-Hilfe definieren und auch in die Registrierungsdatei eintragen (vgl. Kapitel 3.2). Diese müssen innerhalb der jeweiligen Online-Hilfe eindeutig sein, sind aber sonst frei vererbbar.

Handelt es sich um eine relativ umfangreiche Bibliothek, können Sie auch eine hm-Datei erstellen, die alle verwendeten IDs enthält. Bei der Vergabe der MAP-IDs kann dann das Erstellungswerkzeug RoboHelp diese Datei verwenden.

Einträge in der hm-Datei:

```
// Headerfile für Onlinehilfe 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)
Topic-ID	Name und Einsprungadresse des Topics (wie in der Registrierungsdatei festgelegt)
Index	Stichwörter, mit denen vom Indexverzeichnis zum Hilfethema gesprungen werden kann

Die Hilfe kann aus zwei Dateien bestehen, einer HLP-Datei (Hilfethemen) und einer CNT-Datei (Inhaltsverzeichnis).

Die CNT-Datei ist dann sinnvoll, wenn die Baustein-Hilfe nicht ausschließlich als Kontext-Hilfe (F1 auf den selektierten Baustein) verwendet werden soll. Bei einer Bibliothek mit mehreren Bausteinen können die einzelnen Hilfethemen in einem Inhaltsverzeichnis (Contents) aufgeführt werden. Damit besteht die Möglichkeit, auch zu den Hilfethemen der anderen Bausteine zu 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.

Falls 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 | werden z. Zt von PCS7 |
| e | italienisch | nicht unterstützt |
| y | sprachunabhängig | z.B. für Reg-Datei |

Mittels der Registrierung (vgl. Kapitel 3.2) wird von PCS 7 dann die zur jeweiligen über den Dialog "**Options > Customize > Language**" eingestellten Landessprache passende Hilfedatei aufgerufen.

Zuletzt müssen Sie die Hilfedatei (falls vorhanden auch die zugehörige CNT-Datei) in das Unterverzeichnis des STEP7-Verzeichnisses kopieren, in dem Ihre Bibliothek bzw. das Projekt mit Ihren Bausteinen installiert wird.

3.2 Aufbau der Registrierungsdatei

Schreiben Sie mit dem ASCII-Editor eine Registrierungsdatei, die die Informationen für Ihre Bausteine in die WINDOWS-NT-Registrierung einträgt. Der Name der Registrierungsdatei ist frei wählbar, aus Übersichtlichkeitsgründen sollten Sie jedoch den Namen ihrer Bibliothek (bzw. einen gemeinsamen Namen Ihrer Bausteine) verwenden, z.B. "**Mylib_y.reg**".

Beispiel einer Reg-Datei für 3 Bausteine und 5 Sprachversionen

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\mylib\ABC]
"Version"="0.1"
"VersionDate"="03.05.2000"
"HelpFileGerman"="S7libs\mylib\MYLIB__a.hlp"
"HelpFileEnglish"="S7libs\mylib\MYLIB__b.hlp"
"HelpFileFrench"="S7libs\mylib\MYLIB__c.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.

Es müssen die folgenden Werte in den Schlüssel der Registrierung eingetragen werden:

[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 STEP7-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 STEP7-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.

Danach ist der folgende Schlüssel einzutragen:

[HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7\2.0\Applications\s7libs\NameOfLibrary\Author\Family]

Darunter muss für jeden Baustein die Einsprungsadresse in die Hilfedatei angegeben werden, z.B.: "CONTROL"=dword:00000888. Dabei steht **Family** für den von Ihnen beim Attribut FAMILY im Bausteinkopf angegebenen Namen (hier: XYZ). Die Einsprungsadresse ist die Nummer der Topic-ID.

Falls Sie Ihre Bausteine in mehrere Familien gruppiert haben, müssen Sie für jede Familie einen eigenen Schlüssel in die Registrierungsdatei einfügen.

Wenn diese Registrierungsdatei ausgeführt worden ist (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-NT-Registrierung die zugehörige Hilfedatei bestimmt und die betreffende Hilfe angezeigt.

4 Bibliothek und Setup erstellen

Voraussetzungen

Zur Erstellung einer lieferfähigen Bibliothek inkl. dem dazu gehörigen Setup benötigen Sie ein Programm zur Erstellung von Installationsprogrammen, z.B. "InstallShield".

4.1 Bibliothek erstellen

Falls Sie Ihre Bausteine in einer Bibliothek zusammenfassen wollen, gehen Sie folgendermaßen vor:

1. Legen Sie eine neue S7-Bibliothek an und erzeugen darin ein S7-Programm
2. Tragen Sie die Namen und Nummern Ihrer Bausteine sowie die dazugehörigen Kommentare in die Symboltabelle der Bibliothek ein
3. Quellen:
Falls Sie die Quellen mitliefern wollen, kopieren Sie die Quellen aus dem Quellordner Ihres Projektes in den Quellordner der Bibliothek.
4. Kopieren Sie Ihre Bausteine aus dem Bausteinordner Ihres Projektes in den Bausteinordner der Bibliothek
5. Falls Sie in Ihren Multiinstanzbausteinen nicht allgemein verfügbare Bausteine aufrufen (SFBs, SFCs), kopieren Sie diese ebenfalls in den Bausteinordner der Bibliothek.

4.2 Setup erstellen

Falls Sie Ihre Bibliothek per Setup auf dem Zielrechner installieren wollen, entwerfen Sie mit dem Setup-Erstellungs-Werkzeug ein Installationskript, das die folgenden Aktionen durchführt:

- Kopieren der Bausteinbibliothek in das Unterverzeichnis **S7libs** des STEP7-Verzeichnisses
- Aufruf des Programms **S7bin\S7alibxx.exe** im STEP7-Verzeichnis, um die neue Bibliothek beim SIMATIC Manager bekannt zu machen
- Kopieren der Hilfedatei (HLP- und CNT-Datei) in das Unterverzeichnis des STEP7-Verzeichnisses, in den die Bausteinbibliothek kopiert worden ist (z.B. das Unterverzeichnis **S7libs\mylib**)
- Aufruf der zur Hilfedatei gehörenden Registrierungsdatei
- Kopieren der Prototypbilder in ein beliebiges Unterverzeichnis im Unterverzeichnis **Options\Pdl** des WinCC-Verzeichnisses. Sinnvollerweise sollte dieses Verzeichnis denselben Namen haben, wie das, in das die Bausteinbibliothek kopiert worden ist.
- Kopieren der Scripts in ein beliebiges Unterverzeichnis im Unterverzeichnis **aplib** des WinCC-Verzeichnisses. Sinnvollerweise sollte dieses Verzeichnis den selben Namen haben, wie das, in das die Bausteinbibliothek kopiert worden ist (z.B. das Unterverzeichnis **Options\Pdl\mylib**).
- Einrichten einer Deinstallationsmöglichkeit

Beachten Sie, dass die Bausteinbibliothek und die Online-Hilfe nur installiert werden können, wenn auf dem Zielrechner STEP7 vorhanden ist. Die Prototypbilder können nur in ein Unterverzeichnis von WinCC installiert werden. Sehen Sie deshalb im Installationsdialog Abfragen auf das Vorhandensein von STEP7 und WinCC vor. Dazu können Sie in der Registrierung im Schlüssel

HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\STEP7

den Namen **STEP7_VERSION** auf den Wert "5.1"

und im Schlüssel **HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\WinCC\Setup** den Namen **Version** auf den Wert "R 5.1" abfragen.

Glossar

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 Codebausteinen (FC, FB, 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 und 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 (Organisationsbausteine) werden vom Betriebssystem der CPU beim Auftreten asynchroner Ereignisse (z.B. Fehler) aufgerufen.

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.

BATCH flexible

Programmpaket für die Erstellung komplexer Rezeptsteuerungen für den gesamten Bereich von kleinen bis hin zu großen Anwendungen.

Baustein-Attribut

Mit Hilfe der Baustein-Attribute (→ *FUNCTION_BLOCK*, → *TITLE*, → *Liste der Systemattribute*, → *AUTHOR*, → *NAME*, → *VERSION*, → *FAMILY*, → *KNOW_HOW_PROTECT*) des Bausteins können Sie die Objekteigenschaften Ihres Bausteins beeinflussen

Bausteinkopf

1. Abschnitt des → *AS-Bausteins* mit Verwaltungsinformationen (→ *Baustein-Attribute*).
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. Dort erfolgt eine letzte Überprüfung vor der Zuweisung an den Baustein, weil sich in der Zeit zwischen dem OS-Senden und dem AS-Empfang die Prozessbedingungen verändern könnten.

Bedienberechtigung

Berechtigung des aktuellen Anlagenfahres 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

Softwarepaket 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.

CFC

Continuous Function Chart. Synonym für:

1. Funktionsplan, auf dem Bausteine platziert, verschaltet und parametrierbar werden können. Ein CFC-Plan besteht aus 1 bis 26 Teilplänen mit jeweils 6 Blättern.
2. 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.

Deklarationsteil

Bestandteil eines Bausteins, der die Schnittstelle des Bausteins sowie seine intern verwendeten Daten definiert.

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 bzw. Betriebsarten.

FAMILY

→ *Baustein-Attribut.*

Enthält bei Verwendung einer Bausteinbibliothek einen gemeinsamen Namen für eine Teilmenge der Bausteine. FAMILY und → NAME sind Teilschlüssel zur Suche des Hilfetextes eines Bausteins in der Online-Hilfe.

Funktion (FC)

In IEC 1131-3 festgelegt als Software-Einheit, die beim Ausführen ein einziges Ergebnis liefert (es kann auch ein strukturierter Datentyp sein) und keine speichernde Datenablage (Gedächtnis) hat. Der wesentliche Unterschied zu dem FB ist die fehlende Datenablage.

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 bietet können im Anwenderprogramm Parameter übergeben werden. Dadurch eignen sich Funktionsbausteine zur Programmierung von häufig wiederkehrenden komplexen Funktionen, z.B. Regelungen, Betriebsartenwahl. Da ein FB über ein Gedächtnis (Instanz-Datenbaustein) verfügt, kann auf dessen Parameter (z.B. Ausgänge) zu jeder Zeit an jeder beliebigen Stelle im Anwenderprogramm zugegriffen werden.

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.

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.

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 (Datenablage) beinhaltet die Instanzen (Datenablagen) der in ihm zusammengefassten (aufgerufenen) FBs.

NAME

→ *Baustein-Attribut*.

Enthält den symbolischen Namen des Bausteins; identisch mit → FUNCTION_BLOCK. NAME und → FAMILY sind Teilschlüssel zur Suche des Hilfetextes eines Bausteins in der Online-Hilfe.

OK-Flag

Das OK Flag ist eine systeminterne Variable. Tritt während der Ausführung einer Operation ein Fehler auf z.B. Überlauf bei arithmetischen Operationen, so wird das OK-Flag vom System beeinflusst und an den Bausteinausgang ENO durchgereicht.

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 Instanziierung. Es können mehrere Instanzen (d.h. echte Objekte) zu einer Schablone erstellt werden.

Registrierungsdatei

Eine ASCII-Datei (.reg) die alle Informationen enthält wie Pfad, Einsprungadressen für Online-Hilfe etc., um z.B. einen Baustein in der WINDOWS NT-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.

SCL

Höhere Programmiersprache zur Formulierung technologischer Problemlösungen in der SIMATIC S7 (PASCAL-ähnlich), entsprechend 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.).

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.

Systemattribute für Parameter

Spezielle Attribute, die die Darstellung des Parameters durch den → *Bildbaustein* oder seine Behandlung im CFC beeinflussen.

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.

UDT

→ *Anwenderdefinierte Datentypen*

VERSION

→ *Baustein-Attribut*.

Enthält die Versionsnummer des Bausteins

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.....	1-33
ALARM_8P	1-35
AS-Baustein	
Anwender-Schnittstelle	1-5
Aufbau	1-2
asynchronen Ereignis	1-23
Ausgangsparameter.....	1-11

B

Basic Process Control.....	2-1
BATCH flexible.....	1-36
Batch-Kennung	1-34
Bausteinkopf	1-6
Bausteinparameter.....	1-11
Bausteinsymbol.....	2-5
Bedienberechtigung	2-12
Bedienungen.....	1-26
Begleitwerte	1-35
Bereichszuordnung.....	1-34
Bibliothek	4-1
Bildbaustein	
Entwurf.....	2-1
Projektierung.....	2-2
Test.....	2-3
Bildgröße	
ändern.....	2-9
Blockbild CONTROL.....	1-2

C

CFC-Bausteintypen.....	1-37
CNT-Datei	3-2
Codeteil	1-19
CONTROL2	1-37

D

Deklarationsteil	1-11
Durchgangsparameter	1-11
Dynamisierung.....	2-14

E

Eingangsparameter.....	1-11
Erstlauf.....	1-20

F

Faceplate Designer	2-4
--------------------------	-----

G

Globale Skripte.....	2-5
Gruppenbildrahmen.....	2-7

H

Herkunft der Meldung.....	1-34
Hilfdatei	3-1
Hilfethema	3-1
HLP-Datei.....	3-2

I

INCLUDE-Anweisung.....	3-2
Installationsskript.....	4-2
InstallShield	4-1

K

Kommentare	1-12
Kompatibilität zu PCS7.....	2-4
Kreisbildrahmen	2-10

L

Landessprache.....	1-5
Lokale Variablen.....	1-18

M

MAP-ID.....	3-1
Meldebegleitwerte	1-36
Meldekasse	1-34
Meldungen.....	1-28
Meldungsprojektierung	1-33
Multiinstanz	1-18

N

Namenskonventionen.....	1-38
Nummernbereich.....	1-38

O

Objekt-Baukasten.....	2-5
-----------------------	-----

Objekteigenschaften des Bausteins..... 1-7, 1-8
 Online-Hilfe 3-1
 OP_A 1-26
 OP_A_LIM 1-26
 OP_A_RJC 1-26

P

Parametertypen 1-11
 Plananschlüsse..... 1-37
 Prozessbild 2-4

Q

Quellcode..... 1-39

R

Registrierungsdatei 3-3

S

S7-Bibliothek..... 4-1
 SAMPLE_T 1-21
 SCL-Compiler
 Voreinstellungen 1-3
 Setup 4-2
 Sichtenliste 2-7
 Split Screen Wizard 2-5
 Sprachen 1-35

Sprachkennung 3-2
 Sprachumschaltung..... 2-15
 Statische Variablen 1-18
 SUPPTIME 1-22
 Symboltabelle..... 1-5
 Systemattribute
 Bausteine 1-9
 Parameter..... 1-12

T

Tag Logging 2-11
 Temporäre Variablen..... 1-19
 Topic..... 3-1
 Trendsicht..... 2-11

V

Vorlagen 2-4
 Vorlagenbilder 2-5

W

Weckalarm-OB 1-23
 WinCC Graphics Designer 2-2

Z

Zeitabhängigkeit 1-21