

# VIPA SPEED7 Library

OPL\_SP7-LIB | SW90JS0MA V10.002 | Handbuch

HB00 | OPL\_SP7-LIB | SW90JS0MA V10.002 | de | 18-28

Baustein Bibliothek - Standard



VIPA GmbH  
Ohmstr. 4  
91074 Herzogenaurach  
Telefon: +49 9132 744-0  
Telefax: +49 9132 744-1864  
E-Mail: [info@vipa.com](mailto:info@vipa.com)  
Internet: [www.vipa.com](http://www.vipa.com)

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines.....</b>	<b>6</b>
1.1	Copyright © VIPA GmbH .....	6
1.2	Über dieses Handbuch.....	7
<b>2</b>	<b>Wichtige Hinweise.....</b>	<b>8</b>
2.1	Allgemein.....	8
2.2	Intern verwendete Bausteine.....	8
<b>3</b>	<b>Bibliothek einbinden.....</b>	<b>9</b>
3.1	Einbinden in Siemens SIMATIC Manager.....	9
3.2	Einbinden in Siemens TIA Portal.....	10
<b>4</b>	<b>Standard-Bausteine - "Standard" .....</b>	<b>11</b>
4.1	Konvertierung - "Converting" .....	11
4.1.1	FB 80 - LEAD_LAG - Lead/Lag Algorithmus.....	11
4.1.2	FC 93 - SEG - Bitmuster für 7-Segment-Anzeige erzeugen.....	12
4.1.3	FC 94 - ATH - ASCII-Zeichenkette in Hexadezimalzahl wandeln.....	13
4.1.4	FC 95 - HTA - Hexadezimalzahl in ASCII-Zeichenkette wandeln.....	14
4.1.5	FC 96 - ENCO - Bitnummer des niederwertigsten gesetzten Bits lesen.....	14
4.1.6	FC 97 - DECO - Vorgegebenes Bit im Wort setzen.....	15
4.1.7	FC 98 - BCDCPL - Zehnerkomplement erzeugen.....	15
4.1.8	FC 99 - BITSUM - Anzahl der gesetzten Bits zählen.....	16
4.1.9	FC 105 - SCALE - Werte skalieren.....	16
4.1.10	FC 106 - UNSCALE - Werte deskalieren.....	17
4.1.11	FC 108 - RLG_AA1 - Analogwert ausgeben.....	18
4.1.12	FC 109 - RLG_AA2 - Write Analog Value 2.....	19
4.1.13	FC 110 - PER_ET1 - Read/Write Ext. Per. 1.....	20
4.1.14	FC 111 - PER_ET2 - Read/Write Ext. Per. 2.....	20
4.2	IEC-Funktionen - "IEC" .....	21
4.2.1	Datum und Uhrzeit als zusammengesetzte Datentypen.....	21
4.2.2	FC 1 - AD_DT_TM - Zeitdauer auf einen Zeitpunkt addieren.....	22
4.2.3	FC 2 - CONCAT - Zusammenfassen zweier STRING-Variablen.....	22
4.2.4	FC 3 - D_TOD_DT - Zusammenfassen DATE und TIME_OF_DAY.....	22
4.2.5	FC 4 - DELETE - Löschen in einer STRING-Variable.....	23
4.2.6	FC 5 - DI_STRNG - Formatwandlung DINT nach STRING.....	23
4.2.7	FC 6 - DT_DATE - Extrahieren DATE aus DT.....	24
4.2.8	FC 7 - DT_DAY - Extrahieren des Wochentags aus DT.....	24
4.2.9	FC 8 - DT_TOD - Extrahieren TIME_OF_DAY aus DT.....	24
4.2.10	FC 9 - EQ_DT - Vergleich DT auf gleich.....	25
4.2.11	FC 10 - EQ_STRNG - Vergleich STRING auf gleich.....	25
4.2.12	FC 11 - FIND - Suchen in einer STRING-Variable.....	25
4.2.13	FC 12 - GE_DT - Vergleich DT auf größer oder gleich.....	26
4.2.14	FC 13 - GE_STRNG - Vergleich STRING auf größer oder gleich.....	26
4.2.15	FC 14 - GT_DT - Vergleich DT auf größer.....	27
4.2.16	FC 15 - GT_STRNG - Vergleich STRING auf größer.....	27
4.2.17	FC 16 - I_STRNG - Formatwandlung INT nach STRING.....	27
4.2.18	FC 17 - INSERT - Einfügen in eine STRING-Variable.....	28
4.2.19	FC 18 - LE_DT - Vergleich DT auf kleiner oder gleich.....	28
4.2.20	FC 19 - LE_STRNG - Vergleich STRING auf kleiner oder gleich.....	29
4.2.21	FC 20 - LEFT - Linker Teil einer STRING-Variable.....	29

4.2.22	FC 21 - LEN - Länge einer STRING-Variablen.....	29
4.2.23	FC 22 - LIMIT - Begrenzer.....	30
4.2.24	FC 23 - LT_DT - Vergleich DT auf kleiner.....	30
4.2.25	FC 24 - LT_STRNG - Vergleich STRING auf kleiner.....	31
4.2.26	FC 25 - MAX - Maximumauswahl.....	31
4.2.27	FC 26 - MID - Mittlerer Teil einer STRING-Variablen.....	32
4.2.28	FC 27 - MIN - Minimumauswahl.....	32
4.2.29	FC 28 - NE_DT - Vergleich DT auf ungleich.....	33
4.2.30	FC 29 - NE_STRNG - Vergleich STRING auf ungleich.....	34
4.2.31	FC 30 - R_STRNG - Formatwandlung REAL nach STRING.....	34
4.2.32	FC 31 - REPLACE - Ersetzen in einer STRING-Variable.....	34
4.2.33	FC 32 - RIGHT - Rechter Teil einer STRING-Variable.....	35
4.2.34	FC 33 - S5TI_TIM - Formatwandlung S5TIME nach TIME.....	35
4.2.35	FC 34 - SB_DT_DT - Zwei Zeitpunkte subtrahieren.....	36
4.2.36	FC 35 - SB_DT_TM - Zeitdauer von einem Zeitpunkt subtrahieren.....	36
4.2.37	FC 36 - SEL - Binärauswahl.....	37
4.2.38	FC 37 - STRNG_DI - Formatwandlung STRING nach DINT.....	37
4.2.39	FC 38 - STRNG_I - Formatwandlung STRING nach INT.....	38
4.2.40	FC 39 - STRNG_R - Formatwandlung STRING nach REAL.....	38
4.2.41	FC 40 - TIM_S5TI - Formatwandlung TIME nach S5TIME.....	38
4.3	Ein-/Ausgabe - <i>"IO"</i> .....	39
4.3.1	FB 20 - GETIO - PROFIBUS/PROFINET alle Eingänge lesen.....	39
4.3.2	FB 21 - SETIO - PROFIBUS/PROFINET alle Ausgänge schreiben.....	39
4.3.3	FB 22 - GETIO_PART - PROFIBUS/PROFINET Teil-Eingänge lesen.....	40
4.3.4	FB 23 - SETIO_PART - PROFIBUS/PROFINET Teil-Ausgänge schreiben.....	41
4.4	S5-Konvertierung - <i>"S5 Converting"</i> .....	43
4.4.1	FC 112 - Sine(x) - Sinus.....	43
4.4.2	FC 113 - Cosine(x) - Cosinus.....	44
4.4.3	FC 114 - Tangent(x) - Tangens.....	44
4.4.4	FC 115 - Cotangent(x) - Cotangens.....	45
4.4.5	FC 116 - Arc Sine(x) - Arcussinus.....	46
4.4.6	FC 117 - Arc Cosine(x) - Arcuscosinus.....	47
4.4.7	FC 118 - Arc Tangent(x) - Arcustangens.....	47
4.4.8	FC 119 - Arc Cotangent(x) - Arcuscotangens.....	48
4.4.9	FC 120 - Naperian Logarithm $\ln(x)$ - Natürlicher Logarithmus.....	49
4.4.10	FC 121 - Decimal Logarithm $\lg(x)$ - Dezimaler Logarithmus.....	50
4.4.11	FC 122 - Gen. Logarithm to Base b - Allgemeiner Logarithmus $\log(x)$ zur Basis b.....	50
4.4.12	FC 123 - E to Power n - E hoch n.....	51
4.4.13	FC 124 - 10 to Power n - 10 hoch n.....	52
4.4.14	FC 125 - ACCU 2 to Power ACCU 1 - AKKU 2 hoch AKKU 1.....	52
4.5	PID-Steuerung - <i>"PID Control"</i> .....	53
4.5.1	FB 41 - CONT_C - Kontinuierliches Regeln.....	53
4.5.2	FB 42 - CONT_S - Schrittregele.....	59
4.5.3	FB 43 - PULSGEN - Impulsformen.....	64
4.5.4	FB 58 - TCONT_CP - Kontinuierliches Temperaturregeln.....	73
4.5.5	FB 59 - TCONT_S - Temperatur-Schrittregele.....	91
4.6	Zeitfunktionen - <i>"Time Functions"</i> .....	98
4.6.1	UDT 60 - WS_RULES - Regel DB.....	98

4.6.2	FC 61 - BT_LT - Umrechnung Basiszeit in Lokalzeit.....	99
4.6.3	FC 62 - LT_BT - Umrechnung Lokalzeit in Basiszeit.....	100
4.6.4	FC 63 - S_LTINT - Einstellen Uhrzeitalarm in Lokalzeit .....	101

# 1 Allgemeines

## 1.1 Copyright © VIPA GmbH

### All Rights Reserved

Dieses Dokument enthält geschützte Informationen von VIPA und darf außer in Übereinstimmung mit anwendbaren Vereinbarungen weder offengelegt noch benutzt werden.

Dieses Material ist durch Urheberrechtsgesetze geschützt. Ohne schriftliches Einverständnis von VIPA und dem Besitzer dieses Materials darf dieses Material weder reproduziert, verteilt, noch in keiner Form von keiner Einheit (sowohl VIPA-intern als auch -extern) geändert werden, es sei denn in Übereinstimmung mit anwendbaren Vereinbarungen, Verträgen oder Lizenzen.

Zur Genehmigung von Vervielfältigung oder Verteilung wenden Sie sich bitte an: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

E-Mail: [info@vipa.de](mailto:info@vipa.de)

<http://www.vipa.com>



*Es wurden alle Anstrengungen unternommen, um sicherzustellen, dass die in diesem Dokument enthaltenen Informationen zum Zeitpunkt der Veröffentlichung vollständig und richtig sind. Das Recht auf Änderungen der Informationen bleibt jedoch vorbehalten.*

*Die vorliegende Kundendokumentation beschreibt alle heute bekannten Hardware-Einheiten und Funktionen. Es ist möglich, dass Einheiten beschrieben sind, die beim Kunden nicht vorhanden sind. Der genaue Lieferumfang ist im jeweiligen Kaufvertrag beschrieben.*

### EG-Konformitätserklärung

Hiermit erklärt VIPA GmbH, dass die Produkte und Systeme mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften übereinstimmen. Die Übereinstimmung ist durch CE-Zeichen gekennzeichnet.

### Informationen zur Konformitätserklärung

Für weitere Informationen zur CE-Kennzeichnung und Konformitätserklärung wenden Sie sich bitte an Ihre Landesvertretung der VIPA GmbH.

### Warenzeichen

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S und Commander Compact sind eingetragene Warenzeichen der VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300, S7-400 und S7-1500 sind eingetragene Warenzeichen der Siemens AG.

Microsoft und Windows sind eingetragene Warenzeichen von Microsoft Inc., USA.

Portable Document Format (PDF) und Postscript sind eingetragene Warenzeichen von Adobe Systems, Inc.

Alle anderen erwähnten Firmennamen und Logos sowie Marken- oder Produktnamen sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

**Dokument-Support**

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Fehler anzeigen oder inhaltliche Fragen zu diesem Dokument stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

E-Mail: [documentation@vipa.de](mailto:documentation@vipa.de)

**Technischer Support**

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Probleme mit dem Produkt haben oder Fragen zum Produkt stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefon: +49 9132 744-1150 (Hotline)

E-Mail: [support@vipa.de](mailto:support@vipa.de)

## 1.2 Über dieses Handbuch

**Zielsetzung und Inhalt**

Das Handbuch beschreibt die Baustein-Bibliothek *"Standard"* von VIPA:

- Beschrieben wird Aufbau, Projektierung und Anwendung in verschiedenen Programmersystemen.
- Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
- Das Handbuch ist in elektronischer Form als PDF-Datei verfügbar. Hierzu ist der Adobe Acrobat Reader erforderlich.
- Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
- Als Orientierungshilfe stehen im Handbuch zur Verfügung:
  - Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
  - Verweise mit Seitenangabe

**Piktogramme Signalwörter**

Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**GEFAHR!**

Unmittelbar drohende oder mögliche Gefahr. Personenschäden sind möglich.

**VORSICHT!**

Bei Nichtbefolgen sind Sachschäden möglich.



*Zusätzliche Informationen und nützliche Tipps.*

## 2 Wichtige Hinweise

### 2.1 Allgemein



*Nachfolgend finden Sie wichtige Hinweise, die grundsätzlich beim Einsatz der Bausteine zu beachten sind.*

### 2.2 Intern verwendete Bausteine



#### **VORSICHT!**

Folgende Bausteine werden intern verwendet und dürfen nicht überschrieben werden! Der direkte Aufruf eines internen Bausteins führt zu Fehler im entsprechenden Instanz-DB! Bitte verwenden Sie für den Aufruf immer die zugehörige Funktion.

FC/SFC	Bezeichnung	Beschreibung
FC/SFC 192	CP_S_R	wird intern für FB 7 und FB 8 verwendet
FC/SFC 196	AG_CNTRL	wird intern für FC 10 verwendet
FC/SFC 200	AG_GET	wird intern für FB/SFB 14 verwendet
FC/SFC 201	AG_PUT	wird intern für FB/SFB 15 verwendet
FC/SFC 202	AG_BSEND	wird intern für FB/SFB 12 verwendet
FC/SFC 203	AG_BRCV	wird intern für FB/SFB 13 verwendet
FC/SFC 204	IP_CONF	wird intern für FB 55 IP_CONF verwendet
FC/SFC 205	AG_SEND	wird intern für FC 5 AG_SEND verwendet
FC/SFC 206	AG_RECV	wird intern für FC 6 AG_RECV verwendet
FC/SFC 253	IBS_ACCESS	wird intern für SPEED-Bus-INTERBUS-Master verwendet
SFB 238	EC_RWOD	wird intern für EtherCAT-Kommunikation verwendet
SFB 239	FUNC	wird intern für FB 240, FB 241 verwendet

### 3 Bibliothek einbinden

#### Baustein-Bibliothek "Standard"

Die Baustein-Bibliothek finden Sie im *"Service/Support"*-Bereich auf [www.vipa.com](http://www.vipa.com) unter *"Downloads → VIPA Lib"* als *"Baustein-Bibliothek Standard - SW90JS0MA"* zum Download. Die Bibliothek liegt als gepackte zip-Dateien vor. Sobald Sie die Bausteine verwenden möchten, müssen Sie diese in Ihr Projekt importieren.




#### Folgende Bausteinbibliotheken stehen zur Verfügung

Datei	Beschreibung
Standard_S7_V0001.zip	<ul style="list-style-type: none"> <li>■ Bausteinbibliothek für Siemens SIMATIC Manager.</li> <li>■ Für den Einsatz in CPUs von VIPA bzw. S7-300 CPUs von Siemens.</li> </ul>
Standard_TIA_V0002.zip	<ul style="list-style-type: none"> <li>■ Bausteinbibliothek für Siemens TIA Portal V14.</li> <li>■ Für den Einsatz in CPUs von VIPA bzw. S7-300 CPUs von Siemens.</li> </ul>


### 3.1 Einbinden in Siemens SIMATIC Manager

#### Übersicht






Die Einbindung in den Siemens SIMATIC Manager erfolgt nach folgenden Schritten:

1.  ZIP-Datei laden
2.  Bibliothek "dearchivieren"
3.  Bibliothek öffnen und Bausteine in Projekt übertragen



#### ZIP-Datei laden

-  Navigieren Sie auf der Webseite zu der gewünschten ZIP-Datei, laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.

#### Bibliothek dearchivieren

1.  Starten Sie den Siemens SIMATIC Manager mit Ihrem Projekt.
2.  Öffnen Sie mit *"Datei → Dearchivieren"* das Dialogfenster zur Auswahl der ZIP-Datei.
3.  Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
4.  Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind.
5.  Starten Sie den Entpackvorgang mit [OK].

#### Bibliothek öffnen und Bausteine in Projekt übertragen

1.  Öffnen Sie die Bibliothek nach dem Entpackvorgang.
2.  Öffnen Sie Ihr Projekt und kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Bausteine" Ihres Projekts.
  - ⇒ Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die VIPA-spezifischen Bausteine.



Werden anstelle der SFCs FCs verwendet, so werden diese von den VIPA CPUs ab Firmware 3.6.0 unterstützt.

## 3.2 Einbinden in Siemens TIA Portal

### Übersicht

Die Einbindung in das Siemens TIA Portal erfolgt nach folgenden Schritten:

1. ZIP-Datei laden
2. ZIP-Datei entpacken
3. Bibliothek "dearchivieren"
4. Bibliothek öffnen und Bausteine in Projekt übertragen

### ZIP-Datei laden

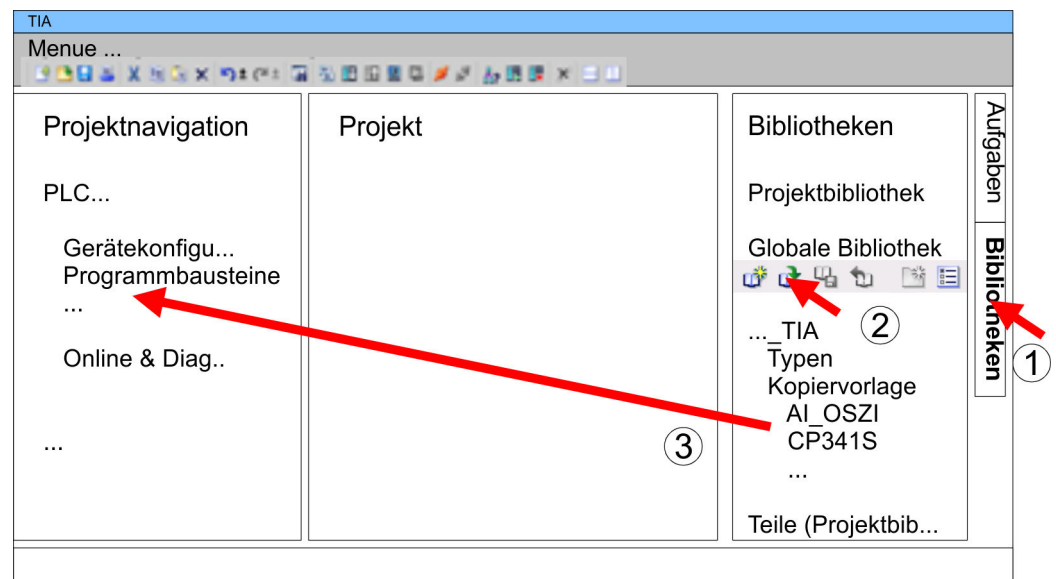
1. Navigieren Sie auf der Webseite zu der ZIP-Datei, welche zu Ihrer Programmversion passt.
2. Laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.

### ZIP-Datei entpacken

- Entpacken Sie die ZIP-Datei mit Ihrem Entpackprogramm in ein Arbeitsverzeichnis für das Siemens TIA Portal.

### Bibliothek öffnen und Bausteine in Projekt übertragen

1. Starten Sie das Siemens TIA Portal mit Ihrem Projekt.
2. Wechseln sie in die *Projektansicht*.
3. Wählen Sie auf der rechten Seite die Task-Card "Bibliotheken".
4. Klicken Sie auf "Globale Bibliothek".
5. Klicken Sie auf "Globale Bibliothek öffnen".
6. Navigieren Sie zu ihrem Arbeitsverzeichnis und laden Sie die Datei ...\_TIA.al1x.



7. Kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Programmbausteine" in der *Projektnavigation* Ihres Projekts. Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die VIPA-spezifischen Bausteine.

## 4 Standard-Bausteine - "Standard"

### 4.1 Konvertierung - "Converting"

#### 4.1.1 FB 80 - LEAD\_LAG - Lead/Lag Algorithmus

##### Beschreibung

Mit dem Funktionsbaustein Lead/Lag Algorithmus LEAD\_LAG können mit einer analogen Variable Signale bearbeitet werden. Ein Ausgang *OUT* wird aufgrund eines Eingangs *IN* und der angegebenen Werte für Verstärken *GAIN*, Voreilen *LD\_TIME* und Nacheilen *LG\_TIME* berechnet. Der Wert für die Verstärkung muss größer sein als Null. Der LEAD\_LAG Algorithmus arbeitet mit der folgenden Gleichung:

$$\text{und } OUT = \left[ \frac{LG\_TIME}{LG\_TIME + SAMPLE\_T} \right] PREV\_OUT + GAIN \left[ \frac{LD\_TIME + SAMPLE\_T}{LG\_TIME + SAMPLE\_T} \right] IN - GAIN \left[ \frac{LD\_TIME}{LG\_TIME + SAMPLE\_T} \right] PREV\_IN$$

Typischerweise wird der FB LEAD\_LAG zusammen mit Schleifen zur Kompensation bei der dynamischen Vorwärtsregelung eingesetzt. LEAD\_LAG besteht aus zwei Teilen. Die Phase LEAD verschiebt die Phase des Ausgangs des Funktionsbausteins, so dass der Ausgang dem Eingang vor eilt. Die Phase LAG hingegen verschiebt den Ausgang, so dass der Ausgang dem Eingang nacheilt. Da die Operation LG mit einer Integration gleichzusetzen ist, kann sie als Entstörelement oder als Tiefpassfilter eingesetzt werden. Die Operation LEAD entspricht einer Differentiation und entspricht deshalb einem Hochpassfilter. Beide Operationen zusammen (LEAD\_LAG) führen dazu, dass die Ausgangsphase dem Eingang bei niederen Frequenzen nacheilt und ihm bei hohen Frequenzen vor eilt. Deshalb kann LEAD\_LAG als Bandpassfilter eingesetzt werden.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Ein Signalzustand von 1 am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand 1, wenn der Funktionsbaustein fehlerfrei ausgeführt wird.
IN	Eingang	REAL	E, A, M, D, L, P, Konstante	Eingabewert des aktuellen Beispielzeitraums, der bearbeitet werden soll
SAMPLE_T	Eingang	INT	E, A, M, D, L, P, Konstante	Beispielzeit
OUT	Ausgang	REAL	E, A, M, D, L, P, Konstante	Ergebnis der Operation LEAD_LAG
ERR_CODE	Ausgang	WORD	E, A, M, D, L, P	Gibt den Wert W#16#0000 aus, wenn die Operation fehlerfrei ausgeführt wird. Wird ein anderer Wert ausgegeben, entnehmen Sie der Fehlerinformation nähere Angaben hierzu.
LD_TIME	Statisch	REAL	E, A, M, D, L, P, Konstante	Voreilzeit in Minuten
LG_TIME	Statisch	REAL	E, A, M, D, L, P, Konstante	Nacheilzeit in Minuten
GAIN	Statisch	REAL	E, A, M, D, L, P, Konstante	Verstärkung in % / % (Verhältnis von Ausgabeveränderung zu Eingabeveränderung als stetiger Zustand)

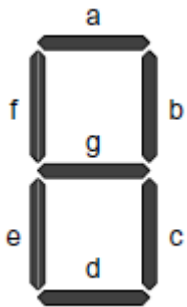
Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
PREV_IN	Statisch	REAL	E, A, M, D, L, P, Konstante	Vorheriger Eingang
PREV_OUT	Statisch	REAL	E, A, M, D, L, P, Konstante	Vorheriger Ausgang

**Fehlerinformation**

Ist *GAIN* kleiner als oder gleich 0, dann wird der Funktionsbaustein nicht ausgeführt. Der Signalzustand von *ENO* wird auf 0 und *ERR\_CODE* wird auf W#16#0009 gesetzt.

**4.1.2 FC 93 - SEG - Bitmuster für 7-Segment-Anzeige erzeugen****Beschreibung**

Die Funktion Bitmuster für 7-Segment-Anzeige erzeugen SEG wandelt jede der vier Hexadezimalziffern des angegebenen Quellworts *IN* in vier äquivalente Codes für eine 7-Segment-Anzeige um und schreibt diese in das Doppelwort des Ausgangs *OUT*. Das Bild unten zeigt die Beziehung zwischen den Hexadezimalziffern des Eingangs und den Bitmustern des Ausgangs.

**Parameter**

Ziffer	– g f e d c b a	Anzeige
0 0 0 0	0 0 1 1 1 1 1 1	0
0 0 0 1	0 0 0 0 0 1 1 0	1
0 0 1 0	0 1 0 1 1 0 1 1	2
0 0 1 1	0 1 0 0 1 1 1 1	3
0 1 0 0	0 1 1 0 0 1 1 0	4
0 1 0 1	0 1 1 0 1 1 0 1	5
0 1 1 0	0 1 1 1 1 1 0 1	6
0 1 1 1	0 0 0 0 0 1 1 1	7
1 0 0 0	0 1 1 1 1 1 1 1	8
1 0 0 1	0 1 1 0 0 1 1 1	9
1 0 1 0	0 1 1 1 0 1 1 1	A
1 0 1 1	0 1 1 1 1 1 0 0	b
1 1 0 0	0 0 1 1 1 0 0 1	C
1 1 0 1	0 1 0 1 1 1 1 0	d
1 1 1 0	0 1 1 1 1 0 0 1	E
1 1 1 1	0 1 1 1 0 0 0 1	F

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Ein Signalzustand von 1 am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand 1, wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	WORD	E, M, D, P, oder Konstante	Quellwort mit vier Hexadezimalziffern.
OUT	Ausgang	DWORD	A, M, D, L, P	Bitmuster des Ziels mit vier Bytes.

**Fehlerinformation**

Diese Funktion erkennt keine Fehlerbedingungen.

**4.1.3 FC 94 - ATH - ASCII-Zeichenkette in Hexadezimalzahl wandeln****Beschreibung**

Die Funktion ASCII-Zeichenkette in Hexadezimalzahl wandeln ATH wandelt die ASCII-Zeichenkette, auf die der Parameter *IN* zeigt, in Hexadezimalziffern um und speichert diese in der Zieltabelle, auf die der Parameter *OUT* zeigt. Da ein ASCII-Zeichen 8 Bits benötigt und eine Hexadezimalziffer nur 4 Bits, ist das Ausgabewort nur halb so lang wie das Eingabewort. Die ASCII-Zeichen werden nach dem Umwandeln in dem Ausgang in der gleichen Reihenfolge angeordnet, in der sie eingelesen wurden. Handelt es sich um eine ungerade Anzahl an ASCII-Zeichen, wird die Hexadezimalziffer in dem Halbbyte ganz rechts der zuletzt umgewandelten Hexadezimalziffer mit Nullen aufgefüllt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Ein Signalzustand von 1 am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand 1, wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	Pointer*	E, A, M, D, L	Zeigt auf die Anfangsadresse einer ASCII-Zeichenkette.
N	Eingang	INT	E, A, M, L, P	Anzahl der ASCII-Zeichen, die umgewandelt werden sollen.
RET_VAL	Ausgang	WORD	E, A, M, D, L, P	Gibt den Wert W#16#0000 aus, wenn die Operation fehlerfrei ausgeführt wird. Wird ein anderer Wert ausgegeben, entnehmen Sie der Fehlerinformation nähere Angaben hierzu.
OUT	Ausgang	Pointer*	A, M, D, L	Zeigt auf die Anfangsadresse der Tabelle.

\*) Pointer im Doppelwortformat bei der bereichsübergreifenden, registerindirekten Adressierung

**Fehlerinformation**

Wird ein ungültiges ASCII-Zeichen erkannt, dann wird es als "0" umgewandelt. Der Signalzustand von *ENO* wird auf 0 gesetzt und *RET\_VAL* ist gleich W#16#0007.

#### 4.1.4 FC 95 - HTA - Hexadezimalzahl in ASCII-Zeichenkette wandeln

##### Beschreibung

Die Funktion Hexadezimalzahl in ASCII-Zeichenkette wandeln HTA wandelt Hexadezimalziffern, auf die der Parameter *IN* zeigt, um und speichert sie in der Zeichenkette, auf die der Parameter *OUT* zeigt. Da ein ASCII-Zeichen 8 Bits benötigt und eine Hexadezimalziffer nur 4 Bits, ist das Ausgabewort doppelt so lang wie das Eingabewort. Jedes Halbbyte der Hexadezimalziffer wird in ein Zeichen umgewandelt, und zwar in der gleichen Reihenfolge, in der sie eingelesen werden (das Halbbyte ganz links der Hexadezimalziffer wird zuerst umgewandelt, danach das Halbbyte ganz rechts der gleichen Ziffer).

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Ein Signalzustand von 1 am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand 1, wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	Pointer*	E, A, M, D	Zeigt auf die Anfangsadresse der Hexadezimalziffern.
N	Eingang	WORD	E, A, M, L, P	Anzahl der Hexadezimalbytes, die umgewandelt werden sollen.
OUT	Ausgang	Pointer*	A, M, D, L	Zeigt auf die Anfangsadresse der Zieltafel.

\*) Pointer im Doppelwortformat bei der bereichsübergreifenden, registerindirekten Adressierung

##### Fehlerinformation

Diese Funktion erkennt keine Fehlerbedingungen.

#### 4.1.5 FC 96 - ENCO - Bitnummer des niederwertigsten gesetzten Bits lesen

##### Beschreibung

Die Funktion Bitnummer des niederwertigsten gesetzten Bits lesen ENCO wandelt den Inhalt des Parameters *IN* in eine Binärziffer mit 5 Bits um, die der Bitposition des Bits ganz rechts in dem Parameter *IN* entspricht, und gibt das Ergebnis als Funktionswert aus. Hat der Parameter *IN* den Wert 0000 0001 oder 0000 0000, dann wird der Wert 0 ausgegeben.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Ein Signalzustand von 1 am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand 1, wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	DWORD	E, M, D, L, P, Konstante	Wert, der codiert werden soll.
RET_VAL	Ausgang	INT	A, M, D, L, P	Wert, der ausgegeben wird (enthält eine Binärziffer mit 5 Bits).

**Fehlerinformation** Diese Funktion erkennt keine Fehlerbedingungen.

#### 4.1.6 FC 97 - DECO - Vorgegebenes Bit im Wort setzen

**Beschreibung** Die Funktion Vorgegebenes Bit im Wort setzen DECO wandelt eine Binärziffer mit 5 Bits (0 bis 31), die von Eingang *IN* angegeben wird, um, indem sie die entsprechende Bitposition in dem Rückgabewert der Funktion setzt. Ist der Parameter *IN* größer als 31, wird eine Modulo-32-Operation ausgeführt, damit eine Binärziffer mit 5 Bits ausgegeben werden kann.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Ein Signalzustand von 1 am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand 1, wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	DWORD	E, M, D, L, P, Konstante	Variable, die decodiert werden soll.
RET_VAL	Ausgang	INT	A, M, D, L, P	Ausgegebener Wert

**Fehlerinformation** Diese Funktion erkennt keine Fehlerbedingungen.

#### 4.1.7 FC 98 - BCDCPL - Zehnerkomplement erzeugen

**Beschreibung** Die Funktion Zehnerkomplement erzeugen BCDCPL gibt das Zehnerkomplement einer siebenstelligen BCD-Zahl aus, die von dem Parameter *IN* angegeben wird. Diese Operation rechnet mit folgender mathematischer Formel:

$$10000000 \text{ (als BCD)} - 7\text{stelliger BCD-Wert} = \text{Zehnerkomplement (als BCD)}$$

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Ein Signalzustand von 1 am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand 1, wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	DWORD	E, M, D, L, P, Konstante	7stellige BCD-Zahl
RET_VAL	Ausgang	DWORD	A, M, D, L, P	Ausgegebener Wert

**Fehlerinformation** Die Funktion erkennt keine Fehlerbedingungen.

#### 4.1.8 FC 99 - BITSUM - Anzahl der gesetzten Bits zählen

##### Beschreibung

Die Funktion Anzahl der gesetzten Bits zählen BITSUM zählt die Anzahl der Bits, die am Eingang *IN* auf 1 gesetzt werden, und gibt diesen Wert als Funktionswert aus.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Ein Signalzustand von 1 am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand 1, wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	DWORD	E, M, D, L, P, Konstante	Variable, in der die Bits gezählt werden sollen.
RET_VAL	Ausgang	INT	A, M, D, L, P	Ausgegebener Wert

##### Fehlerinformation

Diese Funktion erkennt keine Fehlerbedingungen.

#### 4.1.9 FC 105 - SCALE - Werte skalieren

##### Beschreibung

Die Funktion Werte skalieren SCALE wandelt einen ganzzahligen Wert *IN* in einen Realzahlenwert um, der in physikalischen Einheiten zwischen einem unteren und einem oberen Grenzwert *LO\_LIM* und *HI\_LIM* skaliert wird. Das Ergebnis wird in den Parameter *OUT* geschrieben. Die Funktion SCALE arbeitet mit der folgenden Gleichung:

$$OUT = [((FLOAT(IN) - K1) / (K2 - K1)) \cdot (HI\_LIM - LO\_LIM)] + LO\_LIM$$

Die Konstanten K1 und K2 werden unterschiedlich gesetzt, je nachdem, ob der Eingabewert *BIPOLAR* oder *UNIPOLAR* ist.

##### ■ BIPOLAR:

- Es wird angenommen, dass der ganzzahlige Eingabewert zwischen -27648 und 27648 liegt, deshalb sind  
K1 = -27648,0 und K2 = +27648,0.

##### ■ UNIPOLAR:

- Es wird angenommen, dass der ganzzahlige Eingabewert zwischen 0 und 27648 liegt, deshalb sind  
K1 = 0,0 und K2 = +27648,0.

Ist der ganzzahlige Eingabewert größer als K2, dann wird der Ausgang *OUT* an *HI\_LIM* gebunden und ein Fehler ausgegeben. Ist der ganzzahlige Eingabewert kleiner als K1, dann wird der Ausgang an *LO\_LIM* gebunden und ein Fehler ausgegeben. Zum umgekehrten Skalieren wird *LO\_LIM* > *HI\_LIM* programmiert. Beim umgekehrten Skalieren verringert sich der Ausgabewert, während der Eingabewert zunimmt.

## Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang <ul style="list-style-type: none"> <li>TRUE: aktiviert die Funktion</li> <li>FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status <ul style="list-style-type: none"> <li>TRUE: Funktion fehlerfrei ausgeführt</li> </ul> </li> </ul>
IN	INPUT	INT	E, M, D, L, P, Konstante	Eingabewert, der in einen Wert vom Datentyp REAL in physikalischen Einheiten skaliert werden soll.
HI_LIM	INPUT	REAL	E, M, D, L, P, Konstante	Oberer Grenzwert in physikalischen Einheiten
LO_LIM	INPUT	REAL	E, M, D, L, P, Konstante	Unterer Grenzwert in physikalischen Einheiten
BIPOLAR	INPUT	BOOL	E, A, M, D, L	Bei dem Signalzustand 1 handelt es sich um einen bipolaren Eingabewert. Bei dem Signalzustand 0 handelt es sich um einen unipolaren Eingabewert.
OUT	OUTPUT	REAL	E, A, M, D, L, P	Ergebnis der Skalierung
RET_VAL	INPUT	WORD	E, A, M, D, L, P	Gibt den Wert W#16#0000 aus, wenn die Operation fehlerfrei ausgeführt wird. Wird ein anderer Wert ausgegeben, entnehmen Sie der Fehlerinformation nähere Angaben hierzu.

## Fehlerinformation

- Ist der ganzzahlige Eingabewert größer als K2, dann wird der Ausgang *OUT* an *HI\_LIM* gebunden und ein Fehler ausgegeben.
- Ist der ganzzahlige Eingabewert kleiner als K1, dann wird der Ausgang an *LO\_LIM* gebunden und ein Fehler ausgegeben.
- Der Signalzustand von *ENO* wird auf FALSE und *RET\_VAL* wird auf W#16#0008 gesetzt.

## 4.1.10 FC 106 - UNSCALE - Werte deskalieren

## Beschreibung

Die Funktion Werte deskalieren UNSCALE wandelt einen Realzahlenwert *IN*, der in physikalischen Einheiten zwischen einem unteren und einem oberen Grenzwert *LO\_LIM* und *HI\_LIM* skaliert ist, in einen ganzzahligen Wert um. Das Ergebnis wird in den Parameter *OUT* geschrieben. Die Funktion UNSCALE arbeitet mit der folgenden Gleichung:

$$OUT = [((IN - LO\_LIM) / (HI\_LIM - LO\_LIM)) \cdot (K2 - K1)] + K1$$

Die Konstanten K1 und K2 werden unterschiedlich gesetzt, je nachdem, ob der Eingabewert *BIPOLAR* oder *UNIPOLAR* ist.

- BIPOLAR:**
  - Es wird angenommen, dass der ganzzahlige Eingabewert zwischen -27648 und 27648 liegt, deshalb sind  
K1 = -27648,0 und K2 = +27648,0.
- UNIPOLAR:**
  - Es wird angenommen, dass der ganzzahlige Eingabewert zwischen 0 und 27648 liegt, deshalb sind  
K1 = 0,0 und K2 = +27648,0.

Liegt der Eingabewert nicht in dem Bereich zwischen *LO\_LIM* und *HI\_LIM*, dann wird der Ausgang *OUT* an den nächsten Grenzwert (den oberen oder den unteren) für den angegebenen Bereich des jeweiligen Typs (*BIPOLAR* oder *UNIPOLAR*) gebunden und ein Fehler ausgegeben.

#### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>■ Freigabeeingang <ul style="list-style-type: none"> <li>– TRUE: aktiviert die Funktion</li> <li>– FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	Ausgang	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>■ Status <ul style="list-style-type: none"> <li>– TRUE: Funktion fehlerfrei ausgeführt</li> </ul> </li> </ul>
IN	Eingang	REAL	E, M, D, L, P, Konstante	Eingabewert, der in einen ganzzahligen Wert deskaliert werden soll.
HI_LIM	Eingang	REAL	E, M, D, L, P, Konstante	Oberer Grenzwert in physikalischen Einheiten
LO_LIM	Eingang	REAL	E, M, D, L, P, Konstante	Unterer Grenzwert in physikalischen Einheiten
BIPOLAR	Eingang	BOOL	E, A, M, D, L	Bei dem Signalzustand 1 handelt es sich um einen bipolaren Eingabewert, bei dem Signalzustand "0" handelt es sich um einen unipolaren Eingabewert.
OUT	Ausgang	INT	E, A, M, D, L, P	Ergebnis der Skalierung
RET_VAL	Ausgang	WORD	E, A, M, D, L, P	Gibt den Wert W#16#0000 aus, wenn die Operation fehlerfrei ausgeführt wird. Wird ein anderer Wert ausgegeben, entnehmen Sie der Fehlerinformation nähere Angaben hierzu.

#### Fehlerinformation

Liegt der Eingabewert nicht in dem Bereich zwischen *LO\_LIM* und *HI\_LIM*, dann wird der Ausgang *OUT* an den nächsten Grenzwert (den oberen oder den unteren) für den angegebenen Bereich des jeweiligen Typs (*BIPOLAR* oder *UNIPOLAR*) gebunden und ein Fehler ausgegeben. Der Signalzustand von *ENO* wird auf "0" und *RET\_VAL* wird auf W#16#0008 gesetzt.

### 4.1.11 FC 108 - RLG\_AA1 - Analogwert ausgeben

#### Beschreibung

Die Funktion RLG\_AA1 (Analogwert ausgeben) formt einen Eingangswert *XE* (Festpunktzahl) in einen Ausgabewert für eine Analogausgabebaugruppe entsprechend dem Nennbereich zwischen *OGR* und *UGR* um. Bei Überschreiten des Nennbereichs wird eine Fehlermeldung ausgegeben.

Parameter	Datentyp	Speicherbereich	Beschreibung
XE	INT	E, A, M, L, D, Konstante	Eingangswert <i>XE</i> als Festpunktzahl
BG	INT	E, A, M, L, D, Konstante	Angabe der Baugruppenadresse
KNKT	WORD	E, A, M, L, D, Konstante	Kanalnummer KN Kanaltyp KT

Parameter	Datentyp	Speicherbereich	Beschreibung
OGR	INT	E, A, M, L, D, Konstante	Obergrenze des Eingangswerts <i>XE</i>
UGR	INT	E, A, M, L, D, Konstante	Untergrenze des Eingangswerts <i>XE</i>
FEH	BOOL	E, A, M, L, D	Fehlerbit
BU	BOOL	E, A, M, L, D	Bereichsüberschreitung

**Abweichungen von S5**

- Parameter BG:
  - Eine Adressprüfung findet nicht statt. Der Wertebereich ist der gesamte P-Bereich.



*Diese FC wird nur dazu verwendet, den FB 251 eines bestehenden S5-Programms einer S5-CPU 941 bis 944 in eine FC eines S7-Programms für das Automatisierungssystem S7-400 umzuwandeln.*

**4.1.12 FC 109 - RLG\_AA2 - Write Analog Value 2****Beschreibung**

Die Funktion RLG\_AA2 (Analogwert ausgeben) formt einen Eingangswert *XE* (Gleitpunktzahl) in einen Ausgabewert für eine Analogausgabebaugruppe entsprechend dem Nennbereich zwischen *OGR* und *UGR* um. Bei Überschreiten des Nennbereichs wird eine Fehlermeldung ausgegeben.

Parameter	Datentyp	Speicherbereich	Beschreibung
XE	REAL	E, A, M, L, D, Konstante	Eingangswert <i>XE</i> als Gleitpunktzahl
BG	INT	E, A, M, L, D, Konstante	Angabe der Baugruppenadresse
P_Q	WORD	E, A, M, L, D, Konstante	Peripheriebereich normal/erweitert
KNKT	WORD	E, A, M, L, D, Konstante	Kanalnummer KN Kanaltyp KT
OGR	REAL	E, A, M, L, D, Konstante	Obergrenze des Eingangswerts <i>XE</i>
UGR	REAL	E, A, M, L, D, Konstante	Untergrenze des Eingangswerts <i>XE</i>
FEH	BOOL	E, A, M, L, D	Fehlerbit
BU	BOOL	E, A, M, L, D	Bereichsüberschreitung

**Abweichungen von S5**

- Parameter BG:
  - Eine Adressprüfung findet nicht statt. Der Wertebereich ist der gesamte P-Bereich.
- In S7 wird der Parameter *P\_Q* nicht ausgewertet.
- Die S5 Peripheriebereiche P/Q/IM3/IM4 werden auf den S7-Peripheriebereich abgebildet. Die Zuordnung des Peripheriebereichs müssen Sie in der Konfigurationstabelle durchführen.



*Diese FC wird nur dazu verwendet, den FB 41 eines bestehenden S5-Programms einer S5-CPU 928B, 945 oder 948 in eine FC eines S7-Programms für das Automatisierungssystem S7-400 umzuwandeln.*

#### 4.1.13 FC 110 - PER\_ET1 - Read/Write Ext. Per. 1

##### Beschreibung

Die Funktion PER\_ET1 (Lesen und Schreiben für Erweiterte Peripherie) überträgt wahlweise (je nach Parametrierung) einen Peripheriebereich in einen CPU-internen Bereich oder umgekehrt. Dadurch können Eingangsbytes von der erweiterten Peripherie gelesen und Ausgangsbytes zur erweiterten Peripherie geschrieben werden. Falls als interner Bereich ein Datenbaustein gewählt wird, so muss der Baustein vom Anwender vor dem Aufruf der Funktion mit der erforderlichen Länge angelegt worden sein.

Parameter	Datentyp	Speicherbereich	Beschreibung
PBIB	WORD	E, A, M, L, D, Konstante	Angabe der zu bearbeitenden Bereiche
ANF	INT	E, A, M, L, D, Konstante	Anfang des internen Bereichs
ANEN	WORD	E, A, M, L, D, Konstante	Anfang und Ende des Blocks auf der Anschaltung
E_A	BOOL	E, A, M, L, D, Konstante	Transferrichtung
PAFE	BOOL	E, A, M, L, D	Parametrierfehler

##### Abweichungen von S5

- Parameter **PBIB**:
  - In S7 wird der Peripheriebereich wie folgt ausgewertet:

	S5		S7
P-Bereich	0 bis 255	P-Bereich	0 bis 255
Q-Bereich	0 bis 255	P-Bereich	256 bis 511
IM3-Bereich	0 bis 255	P-Bereich	512 bis 767
IM4-Bereich	0 bis 255	P-Bereich	768 bis 1023
DB	0 bis 255	DB	0 bis 255
DX	0 bis 255	DB	256 bis 511
M	0 bis 199	M	0 bis 199
S		Fehlermeldung: "Ungültiger Wertebereich"	

- Die S5-Peripheriebereiche P/Q/IM3/IM4 werden auf den S7-Peripheriebereich abgebildet. Die Zuordnung des Peripheriebereichs müssen Sie in der Konfigurationstabelle durchführen.



*Diese FC wird nur dazu verwendet, den FB 196 eines bestehenden S5-Programms einer S5-CPU 95U, 103, 941 bis 944, 945, 928B, 948 in eine FC eines S7-Programms für das Automatisierungssystem S7-300/400 umzuwandeln.*

#### 4.1.14 FC 111 - PER\_ET2 - Read/Write Ext. Per. 2

##### Beschreibung

Die Funktion PER\_ET2 (Lesen und Schreiben für Erweiterte Peripherie) überträgt wahlweise (je nach Parametrierung) einen Peripheriebereich in einen CPU-internen Bereich oder umgekehrt. Dadurch können Eingangsbytes von der erweiterten Peripherie gelesen und Ausgangsbytes zur erweiterten Peripherie geschrieben werden. Falls als interner Bereich ein Datenbaustein gewählt wird, so muss der Baustein vom Anwender vor dem Aufruf der Funktion mit der erforderlichen Länge angelegt worden sein.

**Abweichungen von S5**

- Parameter *PBIB* (in DB definiert):
  - In S7 wird der Peripheriebereich wie folgt ausgewertet:

	S5		S7
P-Bereich	0 bis 255	P-Bereich	0 bis 255
Q-Bereich	0 bis 255	P-Bereich	256 bis 511
IM3-Bereich	0 bis 255	P-Bereich	512 bis 767
IM4-Bereich	0 bis 255	P-Bereich	768 bis 1023
DB	0 bis 255	DB	0 bis 255
DX	0 bis 255	DB	256 bis 511
M	0 bis 199	M	0 bis 199
S		Fehlermeldung: "Ungültiger Wertebereich"	

- Die S5-Peripheriebereiche P/Q/IM3/IM4 werden auf den S7-Peripheriebereich abgebildet. Die Zuordnung des Peripheriebereichs müssen Sie in der Konfigurationstabelle durchführen.



*Diese FC wird nur dazu verwendet, den FB 197 eines bestehenden S5-Programms einer S5-CPU 95U, 103, 941 bis 944, 945, 928B, 948 in eine FC eines S7-Programms für das Automatisierungssystem S7-300/400 umzuwandeln.*

**4.2 IEC-Funktionen - "IEC"****4.2.1 Datum und Uhrzeit als zusammengesetzte Datentypen****Aktualparameter für DATE\_AND\_TIME**

Der Datentyp DATE\_AND\_TIME ist ein zusammengesetzter Datentyp wie auch ARRAY, STRING und STRUCT. Die zulässigen Speicherbereiche für zusammengesetzte Datentypen sind der Datenbaustein (DB) und der Speicherbereich für Lokaldaten (L-Stack). Wenn Sie den Datentyp DATE\_AND\_TIME als Formalparameter in einer Anweisung verwenden, können Sie aufgrund des zusammengesetzten Datentyps die Aktualparameter nur in einem der folgenden Formate angeben:

- Als bausteinlokales Symbol aus der Variablendeklarationstabelle für einen bestimmten Baustein
- Als symbolischen Namen für einen Datenbaustein, z.B. "DB\_sys\_info.Zeit", der aus den beiden folgenden Teilen besteht:
  - Ein Name, der in der Symboltabelle für die Nummer des Datenbausteins definiert ist (z.B. "DB\_sys\_info" für DB 5)
  - Ein Name, der in dem Datenbaustein für das Element DATE\_AND\_TIME definiert ist (z.B. "Zeit" für eine Variable vom Datentyp DATE\_AND\_TIME, die in DB 5 enthalten ist)



*Sie können keine Konstanten als Aktualparameter für Formalparameter von zusammengesetzten Datentypen, einschließlich DATE\_AND\_TIME, verwenden. Auch können Sie keine absoluten Adressen als Aktualparameter an DATE\_AND\_TIME übergeben.*

### 4.2.2 FC 1 - AD\_DT\_TM - Zeitdauer auf einen Zeitpunkt addieren

#### Beschreibung

Die Funktion FC 1 addiert eine Zeitdauer *D* (Uhrzeit) einen Zeitpunkt *T* (Datum und Uhrzeit) und liefert als Ergebnis einen neuen Zeitpunkt (Datum und Uhrzeit). Der Zeitpunkt *T* muss im Bereich DT#1990-01-01-00:00:00.000 ... DT#2089-12-31-23:59:59.999 liegen. Die Funktion führt keine Eingangsprüfung durch. Liegt das Ergebnis der Addition nicht im oben angegebenen Bereich, wird das Ergebnis auf den entsprechenden Wert begrenzt und das Binärergebnis BIE auf "0" gesetzt.

#### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
T*	INPUT	DATE_AND_TIME	D, L	Zeitpunkt im Format DT
D	INPUT	TIME	E, A, M, D, L, Konstante	Zeitdauer im Format TIME
RET_VAL*	OUTPUT	DATE_AND_TIME	D, L	Summe im Format DT

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

### 4.2.3 FC 2 - CONCAT - Zusammenfassen zweier STRING-Variablen

#### Beschreibung

Die Funktion FC 2 fasst zwei STRING-Variablen zu einer Zeichenkette zusammen. Ist die Ergebniszeichenkette länger als die am Ausgangsparameter angelegte Variable, wird die Ergebniszeichenkette auf die maximal eingerichtete Länge begrenzt und das BIE-Bit auf "0" gesetzt.

#### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN1*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
IN2*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
RET_VAL*	OUTPUT	STRING	D, L	Zusammengefasste Zeichenkette

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

### 4.2.4 FC 3 - D\_TOD\_DT - Zusammenfassen DATE und TIME\_OF\_DAY

#### Beschreibung

Die Funktion FC 3 fasst die Datenformate DATE und TIME\_OF\_DAY (TOD) zusammen und wandelt diese Formate in das Datenformat DATE\_AND\_TIME (DT) um. Der Eingangswert *IN1* muss im Bereich DATE#1990-01-01 ... DATE#2089-12-31 liegen. Die Funktion führt keine Eingangsprüfung durch und meldet keine Fehler.

#### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN1	INPUT	DATE	E, A, M, D, L, Konstante	Eingangsvariable im Format DATE
IN2	INPUT	TIME_OF_DAY	E, A, M, D, L, Konstante	Eingangsvariable im Format TOD

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RET_VAL*	OUTPUT	DATE_AND_TIME	D, L	Rückgabewert im Format DT

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.5 FC 4 - DELETE - Löschen in einer STRING-Variable

##### Beschreibung

Die Funktion FC 4 löscht in einer Zeichenkette *L* Zeichen ab dem *P*. Zeichen (einschließlich). Die Funktion meldet keine Fehler.

- Ist *L* und/oder *P* gleich Null oder ist *P* größer als die aktuelle Länge der Eingangszeichenkette, wird die Eingangszeichenkette zurückgeliefert.
- Ist die Summe aus *L* und *P* größer als die Eingangszeichenkette, wird bis zum Ende der Zeichenkette gelöscht.
- Ist *L* und/oder *P* negativ wird ein Leerstring ausgegeben und das BIE-Bit auf "0" gesetzt.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN*	INPUT	STRING	D, L	STRING-Variable, in der gelöscht wird
L	INPUT	INT	E, A, M, D, L, Konstante	Anzahl der zu löschenden Zeichen
P	INPUT	INT	E, A, M, D, L, Konstante	Position des 1. zu löschenden Zeichens
RET_VAL*	OUTPUT	STRING	D, L	Ergebniszeichenkette

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.6 FC 5 - DI\_STRNG - Formatwandlung DINT nach STRING

##### Beschreibung

Die Funktion FC 5 wandelt eine Variable im DINT-Format in eine Zeichenkette um. Die Zeichenkette wird mit einem führenden Vorzeichen dargestellt. Ist die am Rückgabeparameter angegebene Variable zu kurz, findet keine Wandlung statt und das BIE-Bit wird auf "0" gesetzt.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
I	INPUT	DINT	E, A, M, D, L, Konstante	Eingangswert
RET_VAL*	OUTPUT	STRING	D, L	Ergebniszeichenkette

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.7 FC 6 - DT\_DATE - Extrahieren DATE aus DT

##### Beschreibung

Die Funktion FC 6 extrahiert das Datenformat DATE aus dem Format DATE\_AND\_TIME. DATE liegt zwischen den Grenzen DATE#1990-1-1 ... DATE#2089-12-31. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im DT-Format
RET_VAL	OUTPUT	DATE	E, A, M, D, L	Rückgabewert im Format DATE

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.8 FC 7 - DT\_DAY - Extrahieren des Wochentags aus DT

##### Beschreibung

Die Funktion FC 7 extrahiert den Wochentag aus dem Format DATE\_AND\_TIME. Die Funktion meldet keine Fehler. Der Wochentag wird als INTEGER-Wert zurückgeliefert:

- 1: Sonntag
- 2: Montag
- 3: Dienstag
- 4: Mittwoch
- 5: Donnerstag
- 6: Freitag
- 7: Samstag

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
RET_VAL	OUTPUT	INT	E, A, M, D, L	Rückgabewert im Format INT

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.9 FC 8 - DT\_TOD - Extrahieren TIME\_OF\_DAY aus DT

##### Beschreibung

Die Funktion FC 8 extrahiert das Datenformat TIME\_OF\_DAY aus dem Format DATE\_AND\_TIME. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
RET_VAL	OUTPUT	TIME_OF_DAY	E, A, M, D, L	Rückgabewert im Format TOD

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.10 FC 9 - EQ\_DT - Vergleich DT auf gleich

##### Beschreibung

Die Funktion FC 9 vergleicht die Inhalte zweier Variablen im Format DATE\_AND\_TIME auf gleich und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn der Zeitpunkt am Parameter *DT1* gleich dem Zeitpunkt am Parameter *DT2* ist. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DT1*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
DT2*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.11 FC 10 - EQ\_STRNG - Vergleich STRING auf gleich

##### Beschreibung

Die Funktion FC 10 vergleicht die Inhalte zweier Variablen im Format STRING auf gleich und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn die Zeichenkette am Parameter *S1* gleich der Zeichenkette am Parameter *S2* ist. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S1*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
S2*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.12 FC 11 - FIND - Suchen in einer STRING-Variable

##### Beschreibung

Die Funktion FC 11 liefert die Position der zweiten Zeichenkette *IN2* innerhalb der ersten Zeichenkette *IN1*. Die Suche beginnt links; es wird das erste Auftreten der Zeichenkette gemeldet. Ist die zweite Zeichenkette in der ersten nicht vorhanden, wird Null zurückgemeldet. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN1*	INPUT	STRING	D, L	STRING-Variable, in der gesucht wird
IN2*	INPUT	STRING	D, L	zu suchende STRING-Variable

IEC-Funktionen - IEC &gt; FC 13 - GE\_STRNG - Vergleich STRING auf größer oder gleich

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RET_VAL	OUTPUT	INT	E, A, M, D, L	Position der gefundenen Zeichenkette

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.13 FC 12 - GE\_DT - Vergleich DT auf größer oder gleich

##### Beschreibung

Die Funktion FC 12 vergleicht die Inhalte zweier Variablen im Format DATE\_AND\_TIME auf größer oder gleich und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn der Zeitpunkt am Parameter *DT1* größer (jünger) ist als der Zeitpunkt am Parameter *DT2*, oder wenn beide Zeitpunkte gleich sind. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DT1*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
DT2*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.14 FC 13 - GE\_STRNG - Vergleich STRING auf größer oder gleich

##### Beschreibung

Die Funktion FC 13 vergleicht die Inhalte zweier Variablen im Format STRING auf größer oder gleich und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn die Zeichenkette am Parameter *S1* größer oder gleich der Zeichenkette am Parameter *S2* ist. Die Zeichen werden beginnend von links über ihre ASCII-Codierung verglichen (z.B. ist 'a' größer als 'A'). Das erste unterschiedliche Zeichen entscheidet über das Vergleichsergebnis. Ist der linke Teil der längeren Zeichenkette identisch mit der kürzeren Zeichenkette, gilt die längere Zeichenkette als größer. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S1*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
S2*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.15 FC 14 - GT\_DT - Vergleich DT auf größer

##### Beschreibung

Die Funktion FC 14 vergleicht die Inhalte zweier Variablen im Format DATE\_AND\_TIME auf größer und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn der Zeitpunkt am Parameter *DT1* größer (jünger) ist als der Zeitpunkt am Parameter *DT2*. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DT1*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
DT2*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.16 FC 15 - GT\_STRNG - Vergleich STRING auf größer

##### Beschreibung

Die Funktion FC 15 vergleicht die Inhalte zweier Variablen im STRING-Format auf größer und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn die Zeichenkette am Parameter *S1* größer als die Zeichenkette am Parameter *S2* ist. Die Zeichen werden beginnend von links über ihre ASCII-Codierung verglichen (z.B. ist 'a' größer als 'A'). Das erste unterschiedliche Zeichen entscheidet über das Vergleichsergebnis. Ist der linke Teil der längeren Zeichenkette identisch mit der kürzeren Zeichenkette, gilt die längere Zeichenkette als größer. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S1*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
S2*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.17 FC 16 - I\_STRNG - Formatwandlung INT nach STRING

##### Beschreibung

Die Funktion FC 16 wandelt eine Variable im INT-Format in eine Zeichenkette um. Die Zeichenkette wird mit einem führenden Vorzeichen dargestellt. Ist die am Rückgabeparameter angegebene Variable zu kurz, findet keine Wandlung statt und das BIE-Bit wird auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
I	INPUT	INT	E, A, M, D, L, Konstante	Eingangswert
RET_VAL*	OUTPUT	STRING	D, L	Ergebniszeichenkette

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.18 FC 17 - INSERT - Einfügen in eine STRING-Variable****Beschreibung**

Die Funktion FC 17 fügt die Zeichenkette am Parameter *IN2* in die Zeichenkette am Parameter *IN1* nach dem *P*. Zeichen ein.

- Ist *P* gleich Null, wird die zweite Zeichenkette vor der ersten Zeichenkette eingefügt.
- Ist *P* größer als die aktuelle Länge der ersten Zeichenkette, wird die zweite Zeichenkette an die erste angehängt.
- Ist *P* negativ, wird ein Leerstring ausgegeben und das BIE-Bit auf "0" gesetzt. Das BIE-Bit wird auch auf "0" gesetzt, wenn die Ergebniszeichenkette länger ist als die am Ausgangsparameter angegebene Variable; in diesem Fall wird die Ergebniszeichenkette auf die maximal eingerichtete Länge begrenzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN1*	INPUT	STRING	D, L	STRING-Variable, in die eingefügt wird
IN2*	INPUT	STRING	D, L	einzufügende STRING-Variable
P	INPUT	INT	E, A, M, D, L, Konstante	Einfügeposition
RET_VAL*	OUTPUT	STRING	D, L	Ergebniszeichenkette

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.19 FC 18 - LE\_DT - Vergleich DT auf kleiner oder gleich****Beschreibung**

Die Funktion FC 18 vergleicht die Inhalte zweier Variablen im Format DATE\_AND\_TIME auf kleiner oder gleich und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn der Zeitpunkt am Parameter *DT1* kleiner (älter) ist als der Zeitpunkt am Parameter *DT2*, oder wenn beide Zeitpunkte gleich sind. Die Funktion meldet keine Fehler.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DT1*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
DT2*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
RET_VAL*	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.20 FC 19 - LE\_STRNG - Vergleich STRING auf kleiner oder gleich

##### Beschreibung

Die Funktion FC 19 vergleicht die Inhalte zweier Variablen im STRING-Format auf kleiner oder gleich und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn die Zeichenkette am Parameter S1 kleiner oder gleich der Zeichenkette am Parameter S2 ist. Die Zeichen werden beginnend von links über ihre ASCII-Codierung verglichen (z.B. ist 'A' kleiner als 'a'). Das erste unterschiedliche Zeichen entscheidet über das Vergleichsergebnis. Ist der linke Teil der längeren Zeichenkette identisch mit der kürzeren Zeichenkette, gilt die kürzere Zeichenkette als kleiner. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S1*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
S2*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.21 FC 20 - LEFT - Linker Teil einer STRING-Variable

##### Beschreibung

Die Funktion FC 20 liefert die ersten L Zeichen einer Zeichenkette.

- Ist L größer als die aktuelle Länge der STRING-Variable, wird der Eingangswert zurückgeliefert.
- Bei L = 0 und bei einem Leerstring als Eingangswert wird ein Leerstring zurückgeliefert.
- Ist L negativ wird ein Leerstring ausgegeben und das BIE-Bit auf "0" gesetzt.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
L	INPUT	INT	E, A, M, D, L, Konstante	Länge der linken Zeichenkette
RET_VAL*	OUTPUT	STRING	D, L	Ausgangsvariable im Format STRING

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.22 FC 21 - LEN - Länge einer STRING-Variablen

##### Beschreibung

Eine STRING-Variable enthält zwei Längen:

- Maximale Länge
  - Sie wird bei der Definition der Variablen in eckigen Klammern vorgegeben.
- Aktuelle Länge
  - Das ist die Anzahl der momentan gültigen Zeichen.

Die aktuelle Länge ist kleiner oder gleich der maximalen Länge. Die Anzahl der durch eine Zeichenkette belegten Bytes ist um 2 größer als die maximale Länge. Die Funktion FC 21 gibt die aktuelle Länge einer Zeichenkette (Anzahl der gültigen Zeichen) als Rückgabewert aus. Ein Leerstring ( ' ' ) hat die Länge Null. Die maximale Länge beträgt 254. Die Funktion meldet keine Fehler.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
RET_VAL	OUTPUT	INT	E, A, M, D, L	Anzahl der aktuellen Zeichen

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.23 FC 22 - LIMIT - Begrenzer****Beschreibung**

Die Funktion FC 22 begrenzt den Zahlenwert einer Variablen auf parametrierbare Grenzwerte.

- Als Eingangswerte sind Variablen vom Datentyp INT, DINT und REAL zugelassen.
- Alle parametrierten Variablen müssen vom gleichen Datentyp sein.
- Die Art der Variable wird über den ANY-Pointer erkannt.
- MN darf nicht größer sein als MX.
- Der Ausgangswert bleibt unverändert, und das BIE-Bit wird auf "0" gesetzt, wenn:
  - eine parametrierte Variable einen unzulässigen Datentyp hat.
  - alle parametrierten Variablen untereinander nicht den gleichen Datentyp haben.
  - der untere Grenzwert größer ist als der obere Grenzwert.
  - eine REAL-Variable keine gültige Gleitpunktzahl darstellt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
MN	INPUT	ANY	E, A, M, D, L	Untergrenze
IN	INPUT	ANY	E, A, M, D, L	Eingangsvariable
MX	INPUT	ANY	E, A, M, D, L	Obergrenze
RET_VAL	OUTPUT	ANY	E, A, M, D, L	Begrenzte Ausgangsvariable

**4.2.24 FC 23 - LT\_DT - Vergleich DT auf kleiner****Beschreibung**

Die Funktion FC 23 vergleicht die Inhalte zweier Variablen im Format DATE\_AND\_TIME auf kleiner und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn der Zeitpunkt am Parameter DT1 kleiner (älter) ist als der Zeitpunkt am Parameter DT2. Die Funktion meldet keine Fehler.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DT1*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
DT2*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.25 FC 24 - LT\_STRNG - Vergleich STRING auf kleiner****Beschreibung**

Die Funktion FC 24 vergleicht die Inhalte zweier Variablen im STRING-Format auf kleiner und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn die Zeichenkette am Parameter S1 kleiner als die Zeichenkette am Parameter S2 ist. Die Zeichen werden beginnend von links über ihre ASCII-Codierung verglichen (z.B. ist 'A' kleiner als 'a'). Das erste unterschiedliche Zeichen entscheidet über das Vergleichsergebnis. Ist der linke Teil der längeren Zeichenkette identisch mit der kürzeren Zeichenkette, gilt die kürzere Zeichenkette als kleiner. Die Funktion meldet keine Fehler.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S1*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
S2*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.26 FC 25 - MAX - Maximumauswahl****Beschreibung**

Die Funktion FC 25 wählt aus drei numerischen Variablenwerten den größten aus.

- Als Eingangswerte sind Variablen vom Datentyp INT, DINT und REAL zugelassen.
- Alle parametrisierten Variablen müssen vom gleichen Datentyp sein.
- Die Art der Variable wird über den ANY-Pointer erkannt.
- Der Ausgangswert bleibt unverändert, und das BIE-Bit wird auf "0" gesetzt, wenn:
  - eine parametrisierte Variable einen unzulässigen Datentyp hat.
  - alle parametrisierten Variablen untereinander nicht den gleichen Datentyp haben.
  - eine REAL-Variable keine gültige Gleitpunktzahl darstellt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN1	INPUT	ANY	E, A, M, D, L	1. Eingangswert
IN2	INPUT	ANY	E, A, M, D, L	2. Eingangswert

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN3	INPUT	ANY	E, A, M, D, L	3. Eingangswert
RET_VAL	OUTPUT	ANY	E, A, M, D, L	Größter der Eingangswerte



Die zugelassenen Datentypen INT, DINT und REAL müssen im ANY-Pointer angegeben werden. Parameter wie "MD20" sind ebenfalls zulässig; dazu muss "MD20" aber in "Symbol" mit dem entsprechenden Datentyp definiert werden.

**Beispiel in AWL:**

```
CALL FC 25
IN1 := P#M 10.0 DINT 1
IN2 := MD20
IN3 := P#DB1.DBX 0.0 DINT 1
RET_VAL := P#M 40.0 DINT 1
= M 0.0
```

**4.2.27 FC 26 - MID - Mittlerer Teil einer STRING-Variablen****Beschreibung**

Die Funktion FC 26 liefert den mittleren Teil einer Zeichenkette ( $L$  Zeichen ab dem  $P$ . Zeichen einschließlich).

- Geht die Summe aus  $L$  und  $(P-1)$  über die aktuelle Länge der STRING-Variablen hinaus, wird eine Zeichenkette ab dem  $P$ . Zeichen bis zum Ende des Eingangswerts geliefert.
- In allen anderen Fällen ( $P$  liegt außerhalb der aktuellen Länge,  $P$  und/oder  $L$  gleich Null oder negativ) wird ein Leerstring ausgegeben und das BIE-Bit auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
L	INPUT	INT	E, A, M, D, L, Konstante	Länge der mittleren Zeichenkette
P	INPUT	INT	E, A, M, D, L, Konstante	Position des ersten Zeichens
RET_VAL*	OUTPUT	STRING	D, L	Ausgangsvariable im Format STRING

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.28 FC 27 - MIN - Minimumauswahl****Beschreibung**

Die Funktion FC 27 wählt aus drei numerischen Variablenwerten den kleinsten aus.

- Als Eingangswerte sind Variablen vom Datentyp INT, DINT und REAL zugelassen.
- Alle parametrisierten Variablen müssen vom gleichen Datentyp sein.

- Die Art der Variable wird über den ANY-Pointer erkannt.
- Der Ausgangswert bleibt unverändert, und das BIE-Bit wird auf "0" gesetzt, wenn:
  - eine parametrisierte Variable einen unzulässigen Datentyp hat.
  - alle parametrisierten Variablen untereinander nicht den gleichen Datentyp haben.
  - eine REAL-Variable keine gültige Gleitpunktzahl darstellt.

#### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN1	INPUT	ANY	E, A, M, D, L	1. Eingangswert
IN2	INPUT	ANY	E, A, M, D, L	2. Eingangswert
IN3	INPUT	ANY	E, A, M, D, L	3. Eingangswert
RET_VAL	OUTPUT	ANY	E, A, M, D, L	Kleinster der Eingangswerte



Die zugelassenen Datentypen INT, DINT und REAL müssen im ANY-Pointer angegeben werden. Parameter wie "MD20" sind ebenfalls zulässig; dazu muss "MD20" aber in "Symbol" mit dem entsprechenden Datentyp definiert werden.

#### Beispiel in AWL:

```
CALL FC 27
IN1 := P#M 10.0 DINT 1
IN2 := MD20
IN3 := P#DB1.DBX 0.0 DINT 1
RET_VAL := P#M 40.0 DINT 1
= M 0.0
```

### 4.2.29 FC 28 - NE\_DT - Vergleich DT auf ungleich

#### Beschreibung

Die Funktion FC 28 vergleicht die Inhalte zweier Variablen im Format DATE\_AND\_TIME auf ungleich und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn der Zeitpunkt am Parameter DT1 ungleich dem Zeitpunkt am Parameter DT2 ist. Die Funktion meldet keine Fehler.

#### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DT1*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
DT2*	INPUT	DATE_AND_TIME	D, L	Eingangsvariable im Format DT
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.30 FC 29 - NE\_STRNG - Vergleich STRING auf ungleich

##### Beschreibung

Die Funktion FC 29 vergleicht die Inhalte zweier Variablen im STRING-Format auf ungleich und gibt das Vergleichsergebnis als Rückgabewert aus. Der Rückgabewert führt Signalzustand "1", wenn die Zeichenkette am Parameter S1 ungleich der Zeichenkette am Parameter S2 ist. Die Funktion meldet keine Fehler.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S1*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
S2*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
RET_VAL	OUTPUT	BOOL	E, A, M, D, L	Vergleichsergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.31 FC 30 - R\_STRNG - Formatwandlung REAL nach STRING

##### Beschreibung

Die Funktion FC 30 wandelt eine Variable im REAL-Format in eine Zeichenkette um.

- Die Zeichenkette wird mit 14 Stellen dargestellt:  
 $\pm v.nnnnnnnE\pm xx$ 
  - $\pm$ : Vorzeichen
  - v: 1 Vorkommastelle
  - n: 7 Nachkommastellen
  - x: 2 Exponentenstellen
- Ist die am Rückgabeparameter angegebene Variable zu kurz oder liegt am Parameter IN keine gültige Gleitpunktzahl an, findet keine Wandlung statt und das BIE-Bit wird auf "0" gesetzt.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN	INPUT	REAL	E, A, M, D, L, Konstante	Eingangswert
RET_VAL*	OUTPUT	STRING	D, L	Ergebniszeichenkette

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

#### 4.2.32 FC 31 - REPLACE - Ersetzen in einer STRING-Variable

##### Beschreibung

Die Funktion FC 31 ersetzt die Anzahl von L Zeichen der ersten Zeichenkette IN1 ab dem P. Zeichen (einschließlich) durch die vollständige zweite Zeichenkette IN2.

- Ist L gleich Null und P ungleich Null, wird die erste Zeichenkette zurückgeliefert.
- Ist L gleich Null und P gleich Null, wird die zweite Zeichenkette der ersten Zeichenkette vorangestellt.
- Ist L ungleich Null und P gleich Null oder Eins, wird ab dem 1. Zeichen (einschließlich) ersetzt.

- Liegt  $P$  außerhalb der ersten Zeichenkette, wird die zweite Zeichenkette an die erste Zeichenkette angehängt.
- Ist  $L$  und/oder  $P$  negativ wird ein Leerstring ausgegeben und das BIE-Bit auf "0" gesetzt. Das BIE-Bit wird auch auf "0" gesetzt, wenn die Ergebniszeichenkette länger ist als die am Ausgangsparameter angegebene Variable ist; in diesem Fall wird die Ergebniszeichenkette auf die maximal eingerichtete Länge begrenzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN1*	INPUT	STRING	D, L	STRING-Variable, in die eingesetzt wird
IN2*	INPUT	STRING	D, L	einzusetzende STRING-Variable
L	INPUT	INT	E, A, M, D, L, Konstante	Anzahl der zu ersetzenden Zeichen
P	INPUT	INT	E, A, M, D, L, Konstante	Position des 1. ersetzten Zeichens
RET_VAL*	OUTPUT	STRING	D, L	Ergebniszeichenkette

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.33 FC 32 - RIGHT - Rechter Teil einer STRING-Variable****Beschreibung**

Die Funktion FC 32 liefert die letzten  $L$  Zeichen einer Zeichenkette.

- Ist  $L$  größer als die aktuelle Länge der STRING-Variable, wird der Eingangswert zurückgeliefert.
- Bei  $L = 0$  und bei einem Leerstring als Eingangswert wird ein Leerstring zurückgeliefert.
- Ist  $L$  negativ, wird ein Leerstring ausgegeben und das Binärergebnis BIE auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN*	INPUT	STRING	D, L	Eingangsvariable im Format STRING
L	INPUT	INT	E, A, M, D, L, Konstante	Länge der rechten Zeichenkette
RET_VAL*	OUTPUT	STRING	D, L	Ausgangsvariable im Format STRING

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.34 FC 33 - S5TI\_TIM - Formatwandlung S5TIME nach TIME****Beschreibung**

Die Funktion FC 33 wandelt das Datenformat S5TIME in das Datenformat TIME um. Liegt das Ergebnis der Wandlung außerhalb des TIME-Zahlenbereichs, wird das Ergebnis auf den entsprechenden Wert begrenzt und das Binärergebnis BIE auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN	INPUT	S5TIME	E, A, M, D, L, Konstante	Eingangsvariable im Format S5TIME
RET_VAL	OUTPUT	TIME	E, A, M, D, L	Rückgabewert im Format TIME

**4.2.35 FC 34 - SB\_DT\_DT - Zwei Zeitpunkte subtrahieren****Beschreibung**

Die Funktion FC 34 subtrahiert zwei Zeitpunkte  $DTx$  (Datum und Uhrzeit) und liefert als Ergebnis eine Zeitdauer (Uhrzeit). Die Zeitpunkte  $DTx$  müssen in Bereich DT#1990-01-01-00:00:00.000 ... DT#2089-12-31-23:59:59.999 liegen. Die Funktion führt keine Eingangsprüfung durch. Es gilt:

- Ist  $DT1 > DT2$ , so ist das Ergebnis positiv.
- Ist  $DT1 < DT2$ , so ist das Ergebnis negativ.
- Liegt das Ergebnis der Subtraktion außerhalb des TIME-Zahlenbereichs, wird das Ergebnis auf den entsprechenden Wert begrenzt und das Binärergebnis BIE auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DT1*	INPUT	DATE_AND_TIME	D, L	1. Zeitpunkt im Format DT
DT2*	INPUT	DATE_AND_TIME	D, L	2. Zeitpunkt im Format DT
RET_VAL	OUTPUT	TIME	E, A, M, D, L	Differenz im Format TIME

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.36 FC 35 - SB\_DT\_TM - Zeitdauer von einem Zeitpunkt subtrahieren****Beschreibung**

Die Funktion FC 35 subtrahiert eine Zeitdauer  $D$  (Uhrzeit) von einem Zeitpunkt (Datum und Uhrzeit) und liefert als Ergebnis einen neuen Zeitpunkt (Datum und Uhrzeit). Der Zeitpunkt  $T$  muss im Bereich DT#1990-01-01-00:00:00.000 ... DT#2089-12-31-23:59:59.999 liegen. Die Funktion führt keine Eingangsprüfung durch. Liegt das Ergebnis der Subtraktion nicht im oben angegebenen Bereich, wird das Ergebnis auf den entsprechenden Wert begrenzt und das Binärergebnis BIE auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
T*	INPUT	DATE_AND_TIME	D, L	Zeitpunkt im Format DT
D	INPUT	TIME	E, A, M, D, L, Konstante	Zeitdauer im Format TIME
RET_VAL*	OUTPUT	DATE_AND_TIME	D, L	Differenz im Format DT

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

### 4.2.37 FC 36 - SEL - Binärauswahl

#### Beschreibung

Die Funktion FC 36 wählt abhängig von einem Schalter G einen aus zwei Variablenwerten aus.

- Als Eingangswerte an den Parametern *IN0* und *IN1* sind Variablen mit allen Datentypen zugelassen, die der Datenbreite Bit, Byte, Wort und Doppelwort entsprechen (nicht Datentyp DT und STRING).
- *IN0*, *IN1* und *RET\_VAL* müssen vom gleichen Datentyp sein.
- Der Ausgangswert bleibt unverändert, und das BIE-Bit wird auf "0" gesetzt, wenn:
  - eine parametrisierte Variable einen unzulässigen Datentyp hat.
  - alle parametrisierten Variablen untereinander nicht den gleichen Datentyp haben.
  - eine REAL-Variable keine gültige Gleitpunktzahl darstellt.

#### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
G	INPUT	BOOL	E, A, M, D, L, Konstante	Auswahlschalter
IN0	INPUT	ANY	E, A, M, D, L	1. Eingangswert
IN1	INPUT	ANY	E, A, M, D, L	2. Eingangswert
RET_VAL	OUTPUT	ANY	E, A, M, D, L	Ausgewählter Eingangswert

### 4.2.38 FC 37 - STRNG\_DI - Formatwandlung STRING nach DINT

#### Beschreibung

Die Funktion FC 37 wandelt eine Zeichenkette in eine Variable im Format DINT um

- Das erste Zeichen in der Zeichenkette darf ein Vorzeichen oder eine Ziffer sein, die dann folgenden Zeichen müssen aus Ziffern bestehen.
- Ist die Länge der Zeichenkette Null oder größer als 11 oder befinden sich unerlaubte Zeichen in der Zeichenkette, findet keine Wandlung statt und das BIE-Bit wird auf "0" gesetzt.
- Liegt das Ergebnis der Wandlung außerhalb des DINT-Zahlenbereichs, wird das Ergebnis auf den entsprechenden Wert begrenzt und das Binärergebnis BIE auf "0" gesetzt.

#### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S*	INPUT	STRING	D, L	Eingangszeichenkette
RET_VAL	OUTPUT	DINT	E, A, M, D, L	Ergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.39 FC 38 - STRNG\_I - Formatwandlung STRING nach INT****Beschreibung**

Die Funktion FC 38 wandelt eine Zeichenkette in eine Variable im Format INT um.

- Das erste Zeichen in der Zeichenkette darf ein Vorzeichen oder eine Ziffer sein, die dann folgenden Zeichen müssen aus Ziffern bestehen.
- Ist die Länge der Zeichenkette Null oder größer als 6 oder befinden sich unerlaubte Zeichen in der Zeichenkette, findet keine Wandlung statt und das BIE-Bit wird auf "0" gesetzt.
- Liegt das Ergebnis der Wandlung außerhalb des INT-Zahlenbereichs, wird das Ergebnis auf den entsprechenden Wert begrenzt und das Binärergebnis BIE auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S*	INPUT	STRING	D, L	Eingangszeichenkette
RET_VAL	OUTPUT	INT	E, A, M, D, L	Ergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.40 FC 39 - STRNG\_R - Formatwandlung STRING nach REAL****Beschreibung**

Die Funktion FC 39 wandelt eine Zeichenkette in eine Variable im Format REAL um.

- Die Zeichenkette muss in folgendem Format vorliegen:  
 $\pm v.nnnnnnnE\pm xx$ 
  - $\pm$ : Vorzeichen
  - v: 1 Vorkommastellen
  - n: 7 Nachkommastellen
  - x: 2 Exponentenstellen
- Ist die Länge der Zeichenkette kleiner als 14 oder ist sie nicht wie oben gezeigt aufgebaut, findet keine Wandlung statt und das BIE-Bit wird auf "0" gesetzt.
- Liegt das Ergebnis der Wandlung außerhalb des REAL-Zahlenbereichs, wird das Ergebnis auf den entsprechenden Wert begrenzt und das Binärergebnis BIE auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
S*	INPUT	STRING	D, L	Eingangszeichenkette
RET_VAL	OUTPUT	REAL	E, A, M, D, L	Ergebnis

\*) Parameter kann nur mit einer symbolisch definierten Variablen belegt werden.

**4.2.41 FC 40 - TIM\_S5TI - Formatwandlung TIME nach S5TIME****Beschreibung**

Die Funktion FC 40 wandelt das Datenformat TIME in das Format S5TIME um. Hierbei wird immer abgerundet. Ist der Eingangsparameter größer als das darstellbare S5TIME-Format (TIME#02:46:30.000), wird als Ergebnis S5TIME#999.3 ausgegeben und das Binärergebnis BIE auf "0" gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN	INPUT	TIME	E, A, M, D, L, Konstante	Eingangsvariable im Format TIME
RET_VAL	OUTPUT	S5TIME	E, A, M, D, L	Rückgabewert im Format S5TIME

**4.3 Ein-/Ausgabe - "/>****4.3.1 FB 20 - GETIO - PROFIBUS/PROFINET alle Eingänge lesen****Beschreibung**

Mit dem FB 20 GETIO lesen Sie alle Eingänge eines PROFIBUS-DP-Slave / PROFINET IO-Devices konsistent aus. Der FB 20 ruft dabei die SFC 14 DPRD\_DAT auf. Falls bei der Datenübertragung kein Fehler auftrat, werden die gelesenen Daten in den durch *INPUTS* aufgespannten Zielbereich eingetragen. Der Zielbereich muss dieselbe Länge aufweisen, die Sie für die selektierte Komponente projiziert haben. Bei einem PROFIBUS-DP-Slave mit modularem Aufbau bzw. mit mehreren DP-Kennungen können Sie mit einem FB 20-Aufruf jeweils nur auf die Daten einer Komponente / DP-Kennung unter der projizierten Anfangsadresse zugreifen.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
ID	INPUT	DWORD	E, A, M, D, L, Konstante	<ul style="list-style-type: none"> <li>low word: logische Adresse der DP-Slave- / PROFINET-IO-Komponente (Baugruppe bzw. Modul)</li> <li>high word: irrelevant</li> </ul>
STATUS	OUTPUT	DWORD	E, A, M, D, L	enthält die Fehlerinformation der SFC 14 DPRD_DAT in der Form DW#16#40xxxx00
LEN	OUTPUT	INT	E, A, M, D, L	Anzahl der gelesenen Daten in Bytes
INPUTS	IN_OUT	ANY	E, A, M, D	<p>Zielbereich für die gelesenen Daten.</p> <p>Er muss genauso lang sein, wie Sie für die selektierte DP-Slave- / PROFINET-IO-Komponente projiziert haben. Es ist nur der Datentyp BYTE zulässig.</p>

**Fehlerinformationen**

Siehe SFC 14 - DPRD\_DAT - Konsistente Nutzdaten lesen.

**4.3.2 FB 21 - SETIO - PROFIBUS/PROFINET alle Ausgänge schreiben****Beschreibung**

Mit dem FB 21 SETIO übertragen Sie die Daten aus dem durch *OUTPUTS* aufgespannten Quellbereich konsistent zum adressierten PROFIBUS-DP-Slave / PROFINET IO-Device und ggf. ins Prozessabbild (nämlich dann, wenn Sie den betroffenen Adressbereich des DP-Normslaves als Konsistenzbereich in einem Prozessabbild projiziert haben). Der FB 21 ruft dabei die SFC 15 DPWR\_DAT auf. Der Quellbereich muss dieselbe Länge aufweisen, die Sie für die selektierte Komponente projiziert haben. Bei einem DP-Normslave mit modularem Aufbau bzw. mit mehreren DP-Kennungen können Sie mit einem FB 20-Aufruf nur auf eine Komponente / DP-Kennung des DP-Slaves zugreifen.

Ein-/Ausgabe - IO &gt; FB 22 - GETIO\_PART - PROFIBUS/PROFINET Teil-Eingänge lesen

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
ID	INPUT	DWORD	E, A, M, D, L, Konstante	<ul style="list-style-type: none"> <li>Low word: logische Adresse der DPSlave- / PROFINET-IO-Komponente (Baugruppe bzw. Modul)</li> <li>High word: irrelevant</li> </ul>
LEN	INPUT	INT	E, A, M, D, L	Irrelevant
STATUS	OUTPUT	DWORD	E, A, M, D, L	enthält die Fehlerinformation der SFC 15 DPWR_DAT in der Form DW#16#40xxxx00
OUTPUTS	IN_OUT	ANY	E, A, M, D	Quellbereich für die zu schreibenden Daten. Er muss genauso lang sein, wie Sie für die selektierte DP-Slave- / PROFINET-IO-Komponente projektiert haben. Es ist nur der Datentyp BYTE zulässig.

**Fehlerinformationen**

Siehe SFC 15 - DPWR\_DAT - Konsistente Nutzdaten schreiben.

**4.3.3 FB 22 - GETIO\_PART - PROFIBUS/PROFINET Teil-Eingänge lesen****Beschreibung**

Mit dem FB 22 GETIO\_PART lesen Sie konsistent einen Teil des zu einem PROFIBUS-DP-Slave / PROFINET IO-Device gehörenden Prozessabbildbereichs. Der FB 22 ruft dabei die SFC 81 UBLKMOV auf.



Sie müssen dem OB, in dem der FB 22 GETIO\_PART aufgerufen wird, ein Teilprozessabbild der Eingänge zuordnen. Sie müssen weiterhin vor Aufruf des FB 22 den zugehörigen PROFIBUS-DP-Slave bzw. das zugehörige PROFINET IO-Device in dieses Teilprozessabbild der Eingänge aufnehmen. Falls Ihre CPU keine Teilprozessabbilder kennt oder Sie den FB 22 im OB 1 aufrufen wollen, müssen Sie vor Aufruf des FB 22 den zugehörigen PROFIBUS-DP-Slave bzw. das zugehörige PROFINET IO-Device in das Prozessabbild der Eingänge aufnehmen. Über die Parameter OFFSET und LEN legen Sie die auszulesende Teilmenge des Prozessabbildbereichs der über ID adressierten Komponente fest. Falls bei der Datenübertragung kein Fehler auftrat, erhält ERROR den Wert FALSE, und die gelesenen Daten werden in den durch INPUTS aufgespannten Zielbereich eingetragen. Falls bei der Datenübertragung ein Fehler auftrat, erhält ERROR den Wert TRUE, und STATUS erhält die Fehlerinformation der SFC 81 UBLKMOV. Falls der Zielbereich (Parameter INPUTS) kleiner ist als LEN, werden so viele Bytes übertragen, wie INPUTS aufnehmen kann. ERROR erhält den Wert FALSE. Falls der Zielbereich größer ist als LEN, werden die ersten LEN Bytes des Zielbereichs beschrieben. ERROR erhält den Wert FALSE.



Der FB 22 GETIO\_PART überprüft im Prozessabbild der Eingänge keine Grenzen zwischen Daten, die zu verschiedenen PROFIBUS-DP- bzw. PROFINET IO-Komponenten gehören. Sie müssen daher selbst darauf achten, dass der über OFFSET und LEN festgelegte Prozessabbildbereich zu einer Komponente gehört. Das Komponenten übergreifende Lesen kann für zukünftige Systeme nämlich nicht garantiert werden und gefährdet die Übertragbarkeit auf Systeme anderer Hersteller.

Ein-/Ausgabe - IO &gt; FB 23 - SETIO\_PART - PROFIBUS/PROFINET Teil-Ausgänge schreiben

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
ID	INPUT	DWORD	E, A, M, D, L, Konstante	<ul style="list-style-type: none"> <li>Low word: logische Adresse der DP-Slave- / PROFINET-IO-Komponente (Baugruppe bzw. Modul)</li> <li>High word: irrelevant</li> </ul>
OFFSET	INPUT	INT	E, A, M, D, L, Konstante	Nummer des ersten zu lesenden Bytes im Prozessabbild der Komponente (kleinstmöglicher Wert: 0)
LEN	INPUT	INT	E, A, M, D, L, Konstante	Anzahl der zu lesenden Bytes
STATUS	OUTPUT	DWORD	E, A, M, D, L	enthält die Fehlerinformation der SFC 81 UBLKMOV in der Form DW#16#40xxx00, falls <i>ERROR</i> = TRUE
ERROR	OUTPUT	BOOL	E, A, M, D, L	Fehleranzeige: <i>ERROR</i> = TRUE, falls beim Aufruf der SFC 81 UBLKMOV ein Fehler auftritt.
INPUTS	IN_OUT	ANY	E, A, M, D	Zielbereich für die gelesenen Daten: <ul style="list-style-type: none"> <li>Falls der Zielbereich kleiner ist als <i>LEN</i>, werden so viele Bytes übertragen, wie INPUTS aufnehmen kann. <i>ERROR</i> erhält den Wert FALSE.</li> <li>Falls der Zielbereich größer ist als <i>LEN</i>, werden die ersten <i>LEN</i> Bytes des Zielbereichs beschrieben. <i>ERROR</i> erhält den Wert FALSE.</li> </ul>

**Fehlerinformationen**

Siehe SFC 81 - UBLKMOV - Variable ununterbrechbar kopieren.

**4.3.4 FB 23 - SETIO\_PART - PROFIBUS/PROFINET Teil-Ausgänge schreiben****Beschreibung**

Mit dem FB 23 SETIO\_PART übertragen Sie die Daten aus dem durch *OUTPUTS* angespannten Quellbereich konsistent in einen Teil des zu einem PROFIBUS-DP-Slave / PROFINET IO-Device gehörenden Prozessabbildbereichs. Der FB 23 ruft dabei die SFC 81 UBLKMOV auf.



Sie müssen dem OB, in dem der FB 23 SETIO\_PART aufgerufen wird, ein Teilprozessabbild der Ausgänge zuordnen. Sie müssen weiterhin vor Aufruf des FB 23 den zugehörigen PROFIBUS-DP-Slave bzw. das zugehörige PROFINET IO-Device in dieses Teilprozessabbild der Ausgänge aufnehmen. Falls Ihre CPU keine Teilprozessabbilder kennt oder Sie den FB 23 im OB 1 aufrufen wollen, müssen Sie vor Aufruf des FB 23 den zugehörigen PROFIBUS-DP-Slave bzw. das zugehörige PROFINET IO-Device in das Prozessabbild der Ausgänge aufnehmen. Über die Parameter OFFSET und LEN legen Sie die zu beschreibende Teilmenge des Prozessabbildbereichs der über ID adressierten Komponente fest. Falls bei der Datenübertragung kein Fehler auftrat, erhält ERROR den Wert FALSE. Falls bei der Datenübertragung ein Fehler auftrat, erhält ERROR den Wert TRUE, und STATUS erhält die Fehlerinformation der SFC 81 UBLKMOV. Falls der Quellbereich (Parameter OUTPUTS) kleiner ist als LEN, werden so viele Bytes übertragen, wie OUTPUTS enthält. ERROR erhält den Wert FALSE. Falls der Quellbereich größer ist als LEN, werden die ersten LEN Bytes aus OUTPUTS übertragen. ERROR erhält den Wert FALSE.



Der FB 23 SETIO\_PART überprüft im Prozessabbild der Ausgänge keine Grenzen zwischen Daten, die zu verschiedenen PROFIBUS-DP- bzw. PROFINET IO- Komponenten gehören. Sie müssen daher selbst darauf achten, dass der über OFFSET und LEN festgelegte Prozessabbildbereich zu einer Komponente gehört. Das Komponenten übergreifende Schreiben kann für zukünftige Systeme nämlich nicht garantiert werden und gefährdet die Übertragbarkeit auf Systeme anderer Hersteller.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
ID	INPUT	DWORD	E, A, M, D, L, Konstante	<ul style="list-style-type: none"> <li>Low word: logische Adresse der DP-Slave- / PROFINET-IO-Komponente (Baugruppe bzw. Modul)</li> <li>High word: irrelevant</li> </ul>
OFFSET	INPUT	INT	E, A, M, D, L, Konstante	Nummer des ersten zu beschreibenden Bytes im Prozessabbild der Komponente (kleinstmöglicher Wert: 0)
LEN	INPUT	INT	E, A, M, D, L, Konstante	Anzahl der zu schreibenden Bytes
STATUS	OUTPUT	DWORD	E, A, M, D	enthält die Fehlerinformation der SFC 81 UBLKMOV in der Form DW#16#40xxxx00, falls ERROR = TRUE

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
ERROR	OUTPUT	BOOL	E, A, M, D	Fehleranzeige: <i>ERROR</i> = TRUE, falls beim Aufruf der SFC 81 UBLKMOV ein Fehler auftritt.
OUTPUTS	IN_OUT	ANY	E, A, M, D, L	Quellbereich für die zu schreibenden Daten: <ul style="list-style-type: none"> <li>Falls der Quellbereich kleiner ist als <i>LEN</i>, werden so viele Bytes übertragen, wie <i>OUTPUTS</i> enthält. <i>ERROR</i> erhält den Wert FALSE.</li> <li>Falls der Quellbereich größer ist als <i>LEN</i>, werden die ersten <i>LEN</i> Bytes aus <i>OUTPUTS</i> übertragen. <i>ERROR</i> erhält den Wert FALSE.</li> </ul>

**Fehlerinformationen**

Siehe SFC 81 - UBLKMOV - Variable ununterbrechbar kopieren.

**4.4 S5-Konvertierung - "S5 Converting"****4.4.1 FC 112 - Sine(x) - Sinus****Beschreibung**

Die Funktion FC 112 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. Der Eingangswert muss im Bereich von Null  
(REAL = +0.0000000e+00) ...  $2 \times \pi$  (REAL = +0.6283185e+01) liegen.
2. Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.  
 ⇒ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion gleich FALSE.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang               <ul style="list-style-type: none"> <li>TRUE: aktiviert die Funktion</li> <li>FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status               <ul style="list-style-type: none"> <li>TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

**Fehlerinformation**

Im Fehlerfall, wenn der Eingangswert nicht im Bereich von  $0 \dots 2 \times \pi$  liegt, setzt die Funktion das Verknüpfungsergebnis *ENO* auf Status gleich TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der restlichen Register und die der Hilfsmerker werden nicht verändert.



Diese FC wird nur dazu verwendet, den FB 101 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.

#### 4.4.2 FC 113 - Cosine(x) - Cosinus

##### Beschreibung

Die Funktion FC 113 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. Der Eingangswert muss im Bereich von Null  
(REAL = +0.0000000e+00) ...  $2 \times \pi$  (REAL = +0.6283185e+01) liegen.
2. Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null  
(REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.
  - ⇒ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis ENO nach dem Aufruf der Funktion gleich FALSE.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	■ Freigabeeingang <ul style="list-style-type: none"> <li>– TRUE: aktiviert die Funktion</li> <li>– FALSE: deaktiviert die Funktion</li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	■ Status <ul style="list-style-type: none"> <li>– TRUE: Ein Fehler ist aufgetreten</li> </ul>

##### Fehlerinformation

Im Fehlerfall, wenn der Eingangswert nicht im Bereich von  $0 \dots 2 \times \pi$  liegt, setzt die Funktion das Verknüpfungsergebnis ENO auf Status gleich TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der restlichen Register und die der Hilfsmerker werden nicht verändert.



Diese FC wird nur dazu verwendet, den FB 102 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.

#### 4.4.3 FC 114 - Tangent(x) - Tangens

##### Beschreibung

Die Funktion FC 114 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. Der Eingangswert muss im Bereich von Null  
(REAL = +0.0000000e+00) ...  $2 \times \pi$  (REAL = +0.6283185e+01) liegen.
2. Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.

3. ➔ Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.
- ⇒ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion gleich FALSE.

### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang <ul style="list-style-type: none"> <li>TRUE: aktiviert die Funktion</li> <li>FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status <ul style="list-style-type: none"> <li>TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

### Fehlerinformation

Im Fehlerfall setzt die Funktion das Verknüpfungsergebnis *ENO* auf Status TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Es liegt einer der folgenden Fehler vor:

- Der Eingangswert liegt nicht im Bereich  $0 \dots 2 \times \pi$ .
- Während der Berechnung der Funktion ist ein Zahlenbereichsüberlauf aufgetreten.
- Der Eingangswert beträgt  $\pi/2$  oder  $3 \times \pi/2$ . Dann ist der Funktionswert unendlich.

Die Belegung der restlichen Register und die der Hilfsmerker werden nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 103 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

## 4.4.4 FC 115 - Cotangent(x) - Cotangens

### Beschreibung

Die Funktion FC 115 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. ➔ Der Eingangswert muss im Bereich von  
REAL = +0.2938734e-34 ...  
(fast)  $2 \times \pi$  (REAL = +0.6283184e+01) liegen.
2. ➔ Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. ➔ Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.
- ⇒ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion gleich FALSE.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang <ul style="list-style-type: none"> <li>TRUE: aktiviert die Funktion</li> <li>FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status <ul style="list-style-type: none"> <li>TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

**Fehlerinformation**

Im Fehlerfall setzt die Funktion das Verknüpfungsergebnis *ENO* auf Status TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Es liegt einer der folgenden Fehler vor:

- Der Eingangswert liegt nicht im Bereich von  $\text{REAL} = +0.2938734\text{e-}34$  und  $\text{REAL} = +0.6283184\text{e}+01$
- Während der Berechnung der Funktion ist ein Zahlenbereichsüberlauf aufgetreten.
- Der Eingangswert beträgt 0 oder  $\pi$  oder  $2 \times \pi$ . Dann ist der Funktionswert unendlich.

Die Belegung der restlichen Register und die der Hilfsmerker werden nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 104 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

**4.4.5 FC 116 - Arc Sine(x) - Arcussinus****Beschreibung**

Die Funktion FC 116 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

- Der Eingangswert muss im Bereich von  $-1$  ( $\text{REAL} = -0.1000000\text{e}+01$ ) ...  $+1$  ( $\text{REAL} = +0.1000000\text{e}+01$ ) liegen.
- Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
- Der Eingangswert  $\text{DWORD} = \text{DW}\#16\#0000\ 0000$  wird wie der Gleitpunktwert Null ( $\text{REAL} = +0.0000000\text{e}+00$  entsprechend  $\text{DWORD} = \text{DW}\#16\#8000\ 0000$ ) behandelt.
  - ⇒ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion gleich FALSE.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang <ul style="list-style-type: none"> <li>TRUE: aktiviert die Funktion</li> <li>FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status <ul style="list-style-type: none"> <li>TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

**Fehlerinformation**

Im Fehlerfall, wenn der Eingangswert nicht im Bereich von -1 ... +1 liegt, setzt die Funktion das Verknüpfungsergebnis *ENO* auf Status gleich TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der restlichen Register und die der Hilfsmerker werden nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 105 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

**4.4.6 FC 117 - Arc Cosine(x) - Arcuscosinus****Beschreibung**

Die Funktion FC 117 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. Der Eingangswert muss im Bereich von -1 (REAL = -0.1000000e+01) ... +1 (REAL = +0.1000000e+01) liegen.
2. Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.
  - ⇒ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion gleich FALSE.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	■ Freigabeeingang <ul style="list-style-type: none"> <li>– TRUE: aktiviert die Funktion</li> <li>– FALSE: deaktiviert die Funktion</li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	■ Status <ul style="list-style-type: none"> <li>– TRUE: Ein Fehler ist aufgetreten</li> </ul>

**Fehlerinformation**

Im Fehlerfall, wenn der Eingangswert nicht im Bereich von -1 ... +1 liegt, setzt die Funktion das Verknüpfungsergebnis *ENO* auf Status gleich TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der restlichen Register und die der Hilfsmerker werden nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 106 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

**4.4.7 FC 118 - Arc Tangent(x) - Arcustangens****Beschreibung**

Die Funktion FC 118 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. Der Eingangswert muss im Bereich von -1 (REAL = -0.1000000e+01) ... +1 (REAL = +0.1000000e+01) liegen.

2. Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.
4. Bei einem Eingangswert größer als REAL = +0.1209486e+07 wird als Ergebnis  $+\pi/2$  ausgegeben.  
Bei einem Eingangswert kleiner als REAL = -0.5773456e+07 wird als Ergebnis  $-\pi/2$  ausgegeben.  
⇒ Das Verknüpfungsergebnis *ENO* wird auf Signalzustand FALSE gesetzt.

### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang</li> <li>– TRUE: aktiviert die Funktion</li> <li>– FALSE: deaktiviert die Funktion</li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status</li> <li>– TRUE: Ein Fehler ist aufgetreten</li> </ul>

### Fehlerinformation

Im Fehlerfall, wenn der Eingangswert nicht im Bereich von -1 ... +1 liegt, setzt die Funktion das Verknüpfungsergebnis *ENO* auf Status gleich TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der restlichen Register und die der Hilfsmerker werden nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 107 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

## 4.4.8 FC 119 - Arc Cotangent(x) - Arcuscotangens

### Beschreibung

Die Funktion FC 119 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. Der Eingangswert muss im Bereich von -1 (REAL = -0.1000000e+01) ... +1 (REAL = +0.1000000e+01) liegen.
2. Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.
4. Bei einem Eingangswert größer als REAL = +1.209486e+07 wird als Ergebnis  $+\pi/2$  ausgegeben.  
Bei einem Eingangswert kleiner als REAL = -0.5773456e+07 wird als Ergebnis  $-\pi/2$  ausgegeben.  
⇒ Das Verknüpfungsergebnis *ENO* wird auf Signalzustand FALSE gesetzt.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>■ Freigabeeingang <ul style="list-style-type: none"> <li>– TRUE: aktiviert die Funktion</li> <li>– FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>■ Status <ul style="list-style-type: none"> <li>– TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

**Fehlerinformation**

Im Fehlerfall, wenn der Eingangswert nicht im Bereich von -1 ... +1 liegt, setzt die Funktion das Verknüpfungsergebnis *ENO* auf Status gleich TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der restlichen Register und die der Hilfsmerker werden nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 108 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

**4.4.9 FC 120 - Naperian Logarithm  $\ln(x)$  - Natürlicher Logarithmus****Beschreibung**

Die Funktion FC 120 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. Der Eingangswert muss im Bereich von -1 (REAL = -0.1000000e+01) bis + 1 (REAL = +0.1000000e+01) liegen.
2. Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis nach dem Aufruf der Funktion FALSE.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>■ Freigabeeingang <ul style="list-style-type: none"> <li>– TRUE: aktiviert die Funktion</li> <li>– FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>■ Status <ul style="list-style-type: none"> <li>– TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

**Fehlerinformation**

Im Fehlerfall, wenn der Eingangswert kleiner oder gleich Null ist, setzt die Funktion das Verknüpfungsergebnis *ENO* von Status gleich TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der restlichen Register und die der Hilfsmerker wird nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 109 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

#### 4.4.10 FC 121 - Decimal Logarithm lg(x) - Dezimaler Logarithmus

##### Beschreibung

Die Funktion FC 121 erwartet den Eingangswert im AKKU 1 als Bit-Gleitpunktzahl.

1. ➔ Der Eingangswert muss im Bereich von  
-1 (REAL = -0.1000000e+01) bis + 1 (REAL = +0.1000000e+01) liegen.
2. ➔ Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. ➔ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis nach dem Aufruf der Funktion FALSE.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<div>■ Freigabeeingang</div> <ul style="list-style-type: none"> <li>– TRUE: aktiviert die Funktion</li> <li>– FALSE: deaktiviert die Funktion</li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<div>■ Status</div> <ul style="list-style-type: none"> <li>– TRUE: Ein Fehler ist aufgetreten</li> </ul>

##### Fehlerinformation

Im Fehlerfall, wenn der Eingangswert kleiner oder gleich Null ist, setzt die Funktion das Verknüpfungsergebnis *ENO* auf Status gleich TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der restlichen Register und die der Hilfsmerker wird nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 110 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

#### 4.4.11 FC 122 - Gen. Logarithm to Base b - Allgemeiner Logarithmus log (x) zur Basis b

##### Beschreibung

Die Funktion FC 122 erwartet den Eingangswert für die Basis (b) im AKKU 2 und den Eingangswert für den Numerus (x) im AKKU 1 als Gleitpunktzahlen.

1. ➔ Beide Eingangswerte müssen größer als Null sein; zusätzlich darf die Basis nicht den Wert +1 haben.
2. ➔ Bei ordnungsgemäß ausgeführter Berechnung steht im AKKU 1 das Ergebnis als Gleitpunktzahl, im AKKU 2 der vorherige Inhalt von AKKU 3 und im AKKU 3 der vorherige Inhalt des AKKU 4. Der Inhalt von AKKU 4 wird nicht verändert. Die Belegung der Schmiermerker wird nicht verändert.
3. ➔ Bei einer Berechnung ohne Fehler ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion FALSE.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang <ul style="list-style-type: none"> <li>TRUE: aktiviert die Funktion</li> <li>FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status <ul style="list-style-type: none"> <li>TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

**Fehlerinformation**

Im Falle eines Fehlers, wenn einer der Eingangswerte kleiner oder gleich Null ist oder wenn die Basis den Wert +1 hat, setzt die Funktion das Verknüpfungsergebnis *ENO* auf den Signalzustand TRUE. Dann bleibt der Inhalt der AKKUs unverändert.



*Diese FC wird nur dazu verwendet, den FB 111 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

**4.4.12 FC 123 - E to Power n - E hoch n****Beschreibung**

Die Funktion FC 123 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.
2. Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion FALSE.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang <ul style="list-style-type: none"> <li>TRUE: aktiviert die Funktion</li> <li>FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status <ul style="list-style-type: none"> <li>TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

**Fehlerinformation**

Im Fehlerfall, wenn der Eingangswert nicht im Bereich von REAL = - 0.8802962e+02 ... REAL = +0.8802966e+02 liegt (dann würde der Funktionswert außerhalb des Zahlenbereichs liegen), setzt die Funktion das Verknüpfungsergebnis *ENO* auf den Signalzustand TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der Hilfsmerker wird nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 112 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

#### 4.4.13 FC 124 - 10 to Power n - 10 hoch n

##### Beschreibung

Die Funktion FC 124 erwartet den Eingangswert im AKKU 1 als Gleitpunktzahl.

1. ➔ Der Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.
2. ➔ Das Ergebnis legt die Funktion im AKKU 1 ebenfalls als Gleitpunktzahl ab.
3. ➔ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion FALSE.

##### Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<div>■ Freigabeeingang</div> <div>– TRUE: aktiviert die Funktion</div> <div>– FALSE: deaktiviert die Funktion</div>
ENO	OUTPUT	BOOL	E, A, M, D, L	<div>■ Status</div> <div>– TRUE: Ein Fehler ist aufgetreten</div>

##### Fehlerinformation

Im Fehlerfall, wenn der Eingangswert nicht im Bereich von REAL = -0.3823079e+02 ... REAL = + 0.3823080e+02 liegt (dann würde der Funktionswert außerhalb des Zahlenbereichs liegen), setzt die Funktion das Verknüpfungsergebnis auf den Signalzustand TRUE. Dann bleibt der Inhalt von AKKU 1 unverändert. Die Belegung der Hilfsmerker wird nicht verändert.



*Diese FC wird nur dazu verwendet, den FB 113 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

#### 4.4.14 FC 125 - ACCU 2 to Power ACCU 1 - AKKU 2 hoch AKKU 1

##### Beschreibung

Die Funktion FC 125 erwartet den Eingangswert für die Basis im AKKU 2 und den Eingangswert für den Exponenten im AKKU 1, beide als Gleitpunktzahlen.

1. ➔ Der Eingangswert für die Basis muss positiv sein.  
Ein Eingangswert DWORD = DW#16#0000 0000 wird wie der Gleitpunktwert Null (REAL = +0.0000000e+00 entsprechend DWORD = DW#16#8000 0000) behandelt.  
Für Null hoch Null wird als Ergebnis Null ausgegeben.
2. ➔ Das Ergebnis der Berechnung steht als Gleitpunktzahl im AKKU 1.
3. ➔ Bei ordnungsgemäß ausgeführter Berechnung ist das Verknüpfungsergebnis *ENO* nach dem Aufruf der Funktion FALSE.

**Parameter**

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Freigabeeingang <ul style="list-style-type: none"> <li>TRUE: aktiviert die Funktion</li> <li>FALSE: deaktiviert die Funktion</li> </ul> </li> </ul>
ENO	OUTPUT	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> <li>Status <ul style="list-style-type: none"> <li>TRUE: Ein Fehler ist aufgetreten</li> </ul> </li> </ul>

**Fehlerinformation**

Im Fehlerfall wird *ENO* TRUE wenn:

- der Eingangswert für die Basis kleiner als Null ist
- während der Berechnung der Funktion ein Zahlenbereichsüberlauf aufgetreten ist

Im Fehlerfall bleibt der Inhalt von AKKU 1 und 2 unverändert.



*Diese FC wird nur dazu verwendet, den FB 114 eines bestehenden S5-Programms in eine FC eines S7-Programms umzuwandeln.*

## 4.5 PID-Steuerung - "PID Control"

### 4.5.1 FB 41 - CONT\_C - Kontinuierliches Regeln

**Beschreibung**

Der FB 41 CONT\_C dient zum Regeln von technischen Prozessen mit kontinuierlichen Ein- und Ausgangsgrößen. Über die Parametrierung können Sie Teilfunktionen des PID Reglers zu- oder abschalten und damit diesen an die Regelstrecke anpassen.

**Parameter**

Parameter	Deklaration	Datentyp	Beschreibung
COM_RST	INPUT	BOOL	<p>COMPLETE RESTART / Neustart</p> <ul style="list-style-type: none"> <li>Der Baustein hat eine Neustartroutine, die bearbeitet wird, wenn der Eingang <i>COM_RST</i> gesetzt ist.</li> <li>Default: FALSE</li> </ul>
MAN_ON	INPUT	BOOL	<p>MANUAL VALUE ON / Handbetrieb einschalten</p> <ul style="list-style-type: none"> <li>Ist der Eingang <i>MAN_ON</i> gesetzt, ist der Regelkreis unterbrochen. Als Stellwert wird ein Handwert vorgegeben.</li> <li>Default: TRUE</li> </ul>
PVPER_ON	INPUT	BOOL	<p>PROCESS VARIABLE PERIPHERY ON / Istwert Peripherie einschalten</p> <ul style="list-style-type: none"> <li>Soll der Istwert von der Peripherie eingelesen werden, so muss der Eingang <i>PV_PER</i> mit der Peripherie verschaltet werden und der Eingang <i>PVPER_ON</i> gesetzt werden.</li> <li>Default: FALSE</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
P_SEL	INPUT	BOOL	PROPORTIONAL ACTION ON / P-Anteil einschalten <ul style="list-style-type: none"> <li>■ Im PID-Algorithmus lassen sich die PID-Anteile einzeln zu- und abschalten. Der P-Anteil ist eingeschaltet, wenn der Eingang <i>P_SEL</i> gesetzt ist.</li> <li>■ Default: TRUE</li> </ul>
I_SEL	INPUT	BOOL	INTEGRAL ACTION ON / I-Anteil einschalten <ul style="list-style-type: none"> <li>■ Im PID-Algorithmus lassen sich die PID-Anteile einzeln zu- und abschalten. Der I-Anteil ist eingeschaltet, wenn der Eingang <i>I_SEL</i> gesetzt ist.</li> <li>■ Default: TRUE</li> </ul>
INT_HOLD	INPUT	BOOL	INTEGRAL ACTION HOLD / I-Anteil einfrieren <ul style="list-style-type: none"> <li>■ Der Ausgang des Integrierers kann eingefroren werden. Hierzu muss der Eingang <i>INT_HOLD</i> gesetzt werden.</li> <li>■ Default: FALSE</li> </ul>
I_ITL_ON	INPUT	BOOL	INITIALIZATION OF THE INTEGRAL ACTION / I-Anteil setzen <ul style="list-style-type: none"> <li>■ Der Ausgang des Integrierers kann auf den Eingang <i>I_ITLVAL</i> gesetzt werden. Hierzu muss der Eingang <i>I_ITL_ON</i> gesetzt werden.</li> <li>■ Default: FALSE</li> </ul>
D_SEL	INPUT	BOOL	DERIVATIVE ACTION ON / D-Anteil einschalten <ul style="list-style-type: none"> <li>■ Im PID-Algorithmus lassen sich die PID-Anteile einzeln zu- und abschalten. Der D-Anteil ist eingeschaltet, wenn der Eingang <i>D_SEL</i> gesetzt ist.</li> <li>■ Default: FALSE</li> </ul>
CYCLE	INPUT	TIME	SAMPLE TIME / Abtastzeit <ul style="list-style-type: none"> <li>■ Die Zeit zwischen den Bausteinaufrufen muss konstant sein. Der Eingang <i>CYCLE</i> gibt die Zeit zwischen den Bausteinaufrufen an.</li> <li>■ Default: T#1s</li> <li>■ Wertebereich: <math>\geq 1\text{ms}</math></li> </ul>
SP_INT	INPUT	REAL	INTERNAL SETPOINT / Interner Sollwert <ul style="list-style-type: none"> <li>■ Der Eingang <i>SP_INT</i> dient zur Vorgabe eines Sollwertes.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: -100.0...100. 0 (%) oder phys. Größe<sup>1</sup></li> </ul>
PV_IN	INPUT	REAL	PROCESS VARIABLE IN / Istwert Eingang <ul style="list-style-type: none"> <li>■ Am Eingang <i>PV_IN</i> kann ein Inbetriebsetzungswert parametrierbar oder ein externer Istwert im Gleitpunktformat verschaltet werden.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: -100.0...100. 0 (%) oder phys. Größe<sup>1</sup></li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
PV_PER	INPUT	WORD	<p>PROCESS VARIABLE PERIPHERY / Istwert Peripherie</p> <ul style="list-style-type: none"> <li>Der Istwert in Peripherieformat wird am Eingang <i>PV_PER</i> mit dem Regler verschaltet.</li> <li>Default: W#16#0000</li> </ul>
MAN	INPUT	REAL	<p>MANUAL VALUE / Handwert</p> <ul style="list-style-type: none"> <li>Der Eingang <i>MAN</i> dient zur Vorgabe eines Handwertes mittels Bedien-Beobachterfunktion.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0...100.0 (%) oder phys. Größe<sup>2</sup></li> </ul>
GAIN	INPUT	REAL	<p>PROPORTIONAL GAIN / Proportionalbeiwert</p> <ul style="list-style-type: none"> <li>Der Eingang <i>GAIN</i> gibt die Reglerverstärkung an.</li> <li>Default: 2.0</li> <li>Wertebereich: <math>\geq</math> CYCLE</li> </ul>
TI	INPUT	TIME	<p>RESET TIME / Integrationszeit</p> <ul style="list-style-type: none"> <li>Der Eingang <i>TI</i> bestimmt das Zeitverhalten des Integrierers.</li> <li>Default: T#20s</li> <li>Wertebereich: <math>\geq</math> CYCLE</li> </ul>
TD	INPUT	TIME	<p>DERIVATIVE TIME / Differenzierzeit</p> <ul style="list-style-type: none"> <li>Der Eingang <i>TD</i> bestimmt das Zeitverhalten des Differenzierers.</li> <li>Default: T#10s</li> <li>Wertebereich: <math>\geq</math> CYCLE</li> </ul>
TM_LAG	INPUT	TIME	<p>TIME LAG OF THE DERIVATIVE ACTION / Verzögerungszeit des D-Anteils</p> <ul style="list-style-type: none"> <li>Der Algorithmus des D-Anteils beinhaltet eine Verzögerung, die am Eingang <i>TM_LAG</i> parametrisiert werden kann.</li> <li>Default: T#2s</li> <li>Wertebereich: <math>\geq</math> CYCLE/2</li> </ul>
DEADB_W	INPUT	REAL	<p>DEAD BAND WIDTH / Totzonenbreite</p> <ul style="list-style-type: none"> <li>Die Regeldifferenz wird über eine Totzone geführt. Der Eingang <i>DEADB_W</i> bestimmt die Größe der Totzone.</li> <li>Default: 0.0</li> <li>Wertebereich: <math>\geq</math> 0.0 (%) oder phys. Größe<sup>1</sup></li> </ul>
LMN_HLM	INPUT	REAL	<p>MANIPULATED VALUE HIGH LIMIT / Stellwert obere Begrenzung</p> <ul style="list-style-type: none"> <li>Der Stellwert wird immer auf eine obere und untere Grenze begrenzt. Der Eingang <i>LMN_HLM</i> gibt die obere Begrenzung an.</li> <li>Default: 100.0</li> <li>Wertebereich: <i>LMN_LLM</i> ...100.0 (%) oder phys. Größe<sup>2</sup></li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
LMN_LLM	INPUT	REAL	<p>MANIPULATED VALUE LOW LIMIT / Stellwert untere Begrenzung</p> <ul style="list-style-type: none"> <li>Der Stellwert wird immer auf eine obere und untere Grenze begrenzt. Der Eingang <i>LMN_LLM</i> gibt die untere Begrenzung an.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0... <i>LMN_HLM</i> (%) oder phys. Größe<sup>2</sup></li> </ul>
PV_FAC	INPUT	REAL	<p>PROCESS VARIABLE FACTOR / Istwertfaktor</p> <ul style="list-style-type: none"> <li>Der Eingang <i>PV_FAC</i> wird mit dem Istwert multipliziert. Der Eingang dient zur Anpassung des Istwertbereiches.</li> <li>Default: 1.0</li> </ul>
PV_OFF	INPUT	REAL	<p>PROCESS VARIABLE OFFSET / Istwertoffset</p> <ul style="list-style-type: none"> <li>Der Eingang <i>PV_OFF</i> wird mit dem Istwert addiert. Der Eingang dient zur Anpassung des Istwertbereiches.</li> <li>Default: 0.0</li> </ul>
LMN_FAC	INPUT	REAL	<p>MANIPULATED VALUE FACTOR / Stellwertfaktor</p> <ul style="list-style-type: none"> <li>Der Eingang <i>LMN_FAC</i> wird mit dem Stellwert multipliziert. Der Eingang dient zur Anpassung des Stellwertbereiches.</li> <li>Default: 1.0</li> </ul>
LMN_OFF	INPUT	REAL	<p>MANIPULATED VALUE OFFSET / Stellwertoffset</p> <ul style="list-style-type: none"> <li>Der Eingang <i>LMN_OFF</i> wird mit dem Stellwert addiert. Der Eingang dient zur Anpassung des Stellwertbereiches.</li> <li>Default: 0.0</li> </ul>
I_ITLVAL	INPUT	REAL	<p>INITIALIZATION VALUE OF THE INTEGRAL ACTION / Initialisierungswert für I-Anteil</p> <ul style="list-style-type: none"> <li>Der Ausgang des Integrierers kann am Eingang <i>I_ITL_ON</i> gesetzt werden. Am Eingang <i>I_ITLVAL</i> steht der Initialisierungswert.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0...100. 0 (%) oder phys. Größe<sup>2</sup></li> </ul>
DISV	INPUT	REAL	<p>DISTURBANCE VARIABLE / Störgröße</p> <ul style="list-style-type: none"> <li>Für eine Störgrößenaufschaltung wird die Störgröße am Eingang <i>DISV</i> verschaltet.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0...100. 0 (%) oder phys. Größe<sup>2</sup></li> </ul>
LMN	OUTPUT	REAL	<p>MANIPULATED VALUE / Stellwert</p> <ul style="list-style-type: none"> <li>Am Ausgang <i>LMN</i> wird der effektiv wirkende Stellwert in Gleitpunktformat ausgegeben.</li> <li>Default: 0.0</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
LMN_PER	OUTPUT	WORD	MANIPULATED VALUE PERIPHERY / Stellwert Peripherie <ul style="list-style-type: none"> <li>Der Stellwert in Peripherieformat wird am Ausgang <i>LMN_PER</i> mit dem Regler verschaltet.</li> <li>Default: W#16#0000</li> </ul>
QLMN_HLM	OUTPUT	BOOL	HIGH LIMIT OF MANIPULATED VALUE REACHED / Obere Begrenzung des Stellwertes angesprochen <ul style="list-style-type: none"> <li>Der Stellwert wird immer auf eine obere und untere Grenze begrenzt. Der Ausgang <i>QLMN_HLM</i> meldet die Überschreitung der oberen Begrenzung.</li> <li>Default: FALSE</li> </ul>
QLMN_LLM	OUTPUT	BOOL	LOW LIMIT OF MANIPULATED VALUE REACHED / Untere Begrenzung des Stellwertes angesprochen <ul style="list-style-type: none"> <li>Der Stellwert wird immer auf eine obere und untere Grenze begrenzt. Der Ausgang <i>QLMN_LLM</i> meldet die Überschreitung der unteren Begrenzung.</li> <li>Default: FALSE</li> </ul>
LMN_P	OUTPUT	REAL	PROPORTIONALITY COMPONENT / P-Anteil <ul style="list-style-type: none"> <li>Der Ausgang <i>LMN_P</i> beinhaltet den Proportionalanteil der Stellgröße.</li> <li>Default: 0.0</li> </ul>
LMN_I	OUTPUT	REAL	INTEGRAL COMPONENT / I-Anteil <ul style="list-style-type: none"> <li>Der Ausgang <i>LMN_I</i> beinhaltet den Integralanteil der Stellgröße.</li> <li>Default: 0.0</li> </ul>
LMN_D	OUTPUT	REAL	DERIVATIVE COMPONENT / D-Anteil <ul style="list-style-type: none"> <li>Der Ausgang <i>LMN_D</i> beinhaltet den Differentialanteil der Stellgröße.</li> <li>Default: 0.0</li> </ul>
PV	OUTPUT	REAL	PROCESS VARIABLE / Istwert <ul style="list-style-type: none"> <li>Am Ausgang <i>PV</i> wird der effektiv wirkende Istwert ausgegeben.</li> <li>Default: 0.0</li> </ul>
ER	OUTPUT	REAL	ERROR SIGNAL / Regeldifferenz <ul style="list-style-type: none"> <li>Am Ausgang <i>ER</i> wird die effektiv wirkende Regeldifferenz ausgegeben.</li> <li>Default: 0.0</li> </ul>

1) Parameter im Soll-, Istwertzweig mit gleicher Einheit

2) Parameter im Stellwertzweig mit gleicher Einheit

## Anwendung

Den Regler können Sie als PID-Festwertregler einzeln oder auch in mehrschleifigen Regelungen als Kaskaden-, Mischungs- oder Verhältnisregler einsetzen. Die Arbeitsweise basiert auf dem PID-Regelalgorithmus des Abtastreglers mit analogem Ausgangssignal, gegebenenfalls um eine Impulsformerstufe zur Bildung von pulsbreitenmodulierten Ausgangssignalen für Zwei- oder Dreipunktregelungen mit proportionalen Stellgliedern ergänzt.

Neben den Funktionen im Soll- und Istwertzweig realisiert der FB einen fertigen PID-Regler mit kontinuierlichem Stellgrößen-Ausgang und Beeinflussungsmöglichkeit des Stellwertes von Hand.

**Sollwertzweig**

Der Sollwert wird am Eingang *SP\_INT* im Gleitpunktformat eingegeben.

**Istwertzweig**

Der Istwert kann im Peripherie- und im Gleitpunktformat eingelesen werden. Die Funktion *CRP\_IN* wandelt den Peripheriewert *PV\_PER* in ein Gleitpunktformat von -100 .... +100 % nach folgender Vorschrift um:

$$\text{Ausgang von } \text{CPR\_IN} = \text{PV\_PER} * \frac{100}{27648}$$

Die Funktion *PV\_NORM* normiert den Ausgang von *CRP\_IN* nach folgender Vorschrift:

$$\text{Ausgang von } \text{PV\_NORM} = (\text{Ausgang von } \text{CPR\_IN}) * \text{PV\_FAC} + \text{PV\_OFF}$$

*PV\_FAC* ist mit 1 und *PV\_OFF* ist mit 0 vorbelegt.

**Regeldifferenzbildung**

Die Differenz von Soll- und Istwert ergibt die Regeldifferenz. Zur Unterdrückung einer kleinen Dauerschwingung aufgrund der Stellgrößen-Quantisierung (z.B. bei einer Puls-Breitenmodulation mit PULSEGEN) wird die Regeldifferenz über eine Totzone (DEAD-BAND) geleitet. Bei *DEADB\_W* = 0 ist die Totzone ausgeschaltet.

**PID-Algorithmus**

Der PID-Algorithmus arbeitet im Stellungsalgorithmus. Der Proportional-, Integral (*INT*) und Differentialanteil (*DIF*) sind parallel geschaltet und einzeln zu- und abschaltbar. Damit lassen sich P-, PI-, PD- und PID-Regler parametrieren. Aber auch reine I-Regler sind möglich.

**Handwertverarbeitung**

Es kann zwischen Hand- und Automatikbetrieb umgeschaltet werden. Bei Handbetrieb wird die Stellgröße einem Handwert nachgeführt. Der Integrierer (*INT*) wird intern auf *LMN - LMN\_P - DISV* und der Differenzierer (*DIF*) auf 0 gesetzt und intern abgeglichen. Das Umschalten in den Automatikbetrieb ist damit stoßfrei.

**Stellwertverarbeitung**

Der Stellwert wird mit der Funktion *LMNLIMIT* auf vorgebbare Werte begrenzt. Das Überschreiten der Grenzen durch die Eingangsgröße wird durch Meldebits angezeigt. Die Funktion *LMN\_NORM* normiert den Ausgang von *LMNLIMIT* nach folgender Vorschrift:

$$\text{LMN} = (\text{Ausgang von } \text{LMNLIMIT}) * \text{LMN\_FAC} + \text{LMN\_OFF}$$

*LMN\_FAC* ist mit 1 und *LMN\_OFF* mit 0 vorbelegt.

Der Stellwert steht auch im Peripherieformat zur Verfügung. Die Funktion *CRP\_OUT* wandelt den Gleitpunktwert *LMN* in einen Peripheriewert nach folgender Vorschrift um:

$$\text{LMN\_PER} = \text{LMN} * \frac{27648}{100}$$

**Störgrößenaufschaltung**

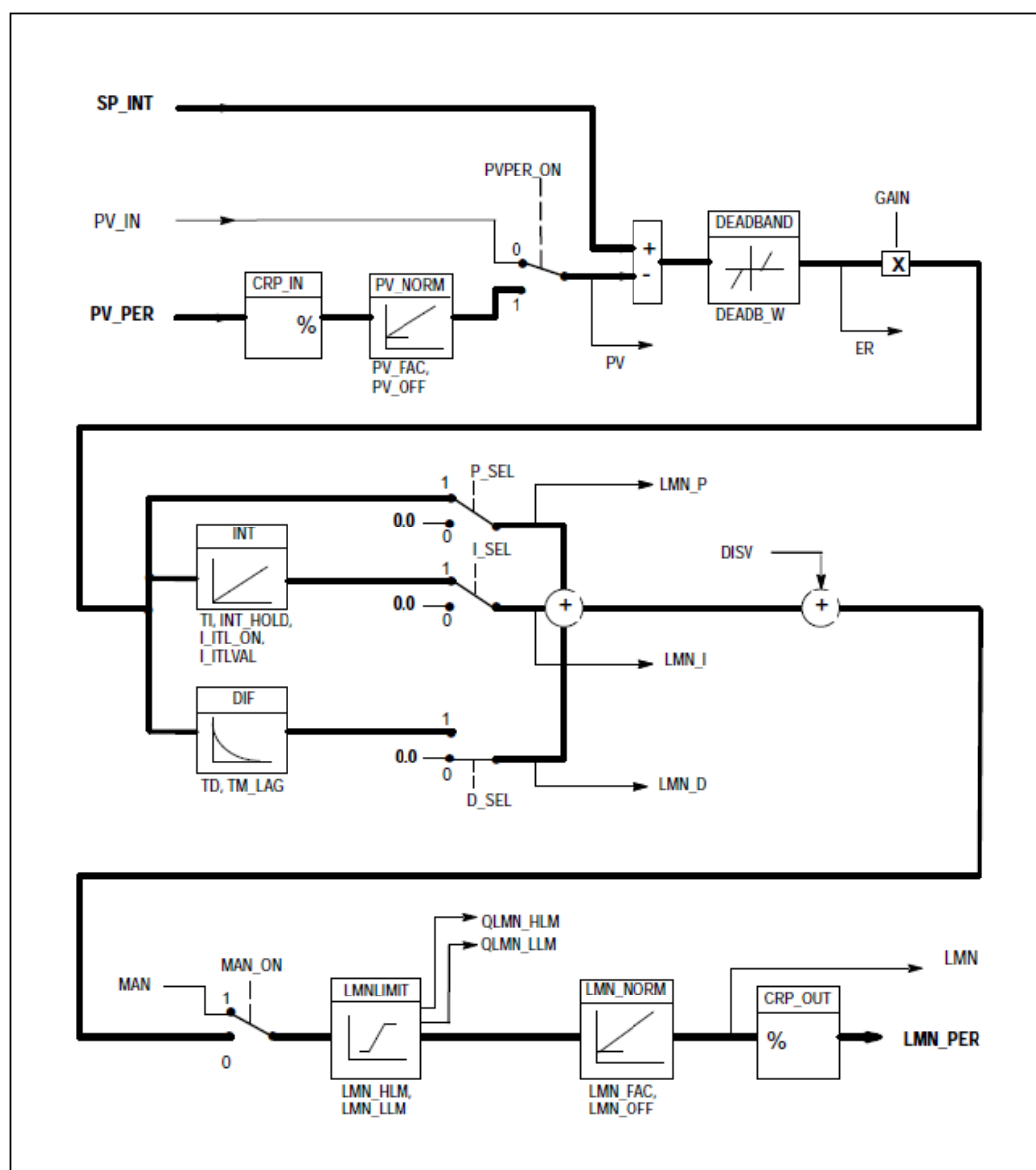
Am Eingang *DISV* kann eine Störgröße additiv aufgeschaltet werden.

**Betriebszustände***Neustart/Wiederanlauf*

- Der FB 41 CONT\_C verfügt über eine Neustartroutine, die durchlaufen wird, wenn der Eingangs-Parameter *COM\_RST* = TRUE gesetzt ist.
- Der Integrierer wird beim Anlauf intern auf den Initialisierungswert *I\_ITLVAL* gesetzt. Beim Aufruf in einer Weckalarmebene arbeitet er von diesem Wert aus weiter.
- Alle anderen Ausgänge werden auf ihre Vorbelegungswerte gesetzt.

**Fehlerinformationen**

Der Baustein enthält keine Fehlerprüfung, deshalb wird keine Fehlerinformation ausgegeben.

**Blockschaltbild****4.5.2 FB 42 - CONT\_S - Schrittregeln****Beschreibung**

Der FB 42 CONT\_S dient zum Regeln von technischen Prozessen mit binären Stellwertausgangssignalen für integrierende Stellglieder. Über die Parametrierung lassen sich Teilfunktionen des PI-Schrittreglers zu- oder abschalten und damit an die Regelstrecke anpassen.

## Parameter

Parameter	Deklaration	Datentyp	Beschreibung
COM_RST	INPUT	BOOL	<p>COMPLETE RESTART / Neustart</p> <ul style="list-style-type: none"> <li>Der Baustein hat eine Neustartroutine, die bearbeitet wird, wenn der Eingang <b>COM_RST</b> gesetzt ist.</li> <li>Default: FALSE</li> </ul>
LMNR_HS	INPUT	BOOL	<p>HIGH LIMIT SIGNAL OF REPEATED MANIPULATED VALUE / Oberes Anschlagssignal der Stellungsrückmeldung</p> <ul style="list-style-type: none"> <li>Das Signal "Stellventil am oberen Anschlag" wird am Eingang <b>LMNR_HS</b> verschaltet. <ul style="list-style-type: none"> <li><b>LMNR_HS</b> = TRUE heißt: Das Stellventil befindet sich am oberen Anschlag.</li> </ul> </li> <li>Default: FALSE</li> </ul>
LMNR_LS	INPUT	BOOL	<p>LOW LIMIT SIGNAL OF REPEATED MANIPULATED VALUE / Unteres Anschlagssignal der Stellungsrückmeldung</p> <ul style="list-style-type: none"> <li>Das Signal "Stellventil am unteren Anschlag" wird am Eingang <b>LMNR_LS</b> verschaltet. <ul style="list-style-type: none"> <li><b>LMNR_LS</b> = TRUE heißt: Das Stellventil befindet sich am unteren Anschlag.</li> </ul> </li> <li>Default: FALSE</li> </ul>
LMNS_ON	INPUT	BOOL	<p>MANIPULATED SIGNALS ON / Handbetrieb der Stellwertsignale einschalten</p> <ul style="list-style-type: none"> <li>Am Eingang <b>LMNS_ON</b> wird die Stellwertsignalverarbeitung auf Hand geschaltet.</li> <li>Default: FALSE</li> </ul>
LMNUP	INPUT	BOOL	<p>MANIPULATED SIGNALS UP / Stellwertsignal Hoch</p> <ul style="list-style-type: none"> <li>Bei Handbetrieb der Stellwertsignale wird am Eingang <b>LMNUP</b> das Ausgangssignal <b>QLMNUP</b> bedient.</li> <li>Default: FALSE</li> </ul>
LMNDN	INPUT	BOOL	<p>MANIPULATED SIGNALS DOWN / Stellwertsignal Tief</p> <ul style="list-style-type: none"> <li>Bei Handbetrieb der Stellwertsignale wird am Eingang <b>LMNDN</b> das Ausgangssignal <b>QLMNDN</b> bedient.</li> <li>Default: FALSE</li> </ul>
PVPER_ON	INPUT	BOOL	<p>PROCESS VARIABLE PERIPHERY ON / Istwert Peripherie einschalten</p> <ul style="list-style-type: none"> <li>Soll der Istwert von der Peripherie eingelesen werden, so muss der Eingang <b>PV_PER</b> mit der Peripherie verschaltet werden und der Eingang <b>PVPER_ON</b> gesetzt werden.</li> <li>Default: FALSE</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
CYCLE	INPUT	TIME	<p>SAMPLE TIME / Abtastzeit</p> <ul style="list-style-type: none"> <li>Die Zeit zwischen den Bausteinaufrufen muss konstant sein. Der Eingang <i>CYCLE</i> gibt die Zeit zwischen den Bausteinaufrufen an.</li> <li>Default: T#1s</li> <li>Wertebereich: <math>\geq 1\text{ms}</math></li> </ul>
SP_INT	INPUT	REAL	<p>INTERNAL SETPOINT / Interner Sollwert</p> <ul style="list-style-type: none"> <li>Der Eingang <i>SP_INT</i> dient zur Vorgabe eines Sollwertes.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0...100.0 (%) oder phys. Größe<sup>1</sup></li> </ul>
PV_IN	INPUT	REAL	<p>PROCESS VARIABLE IN / Istwert Eingang</p> <ul style="list-style-type: none"> <li>Am Eingang <i>PV_IN</i> kann ein Inbetriebsetzungswert parametrierbar oder ein externer Istwert im Gleitpunktformat verschaltet werden.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0...100.0 (%) oder phys. Größe<sup>1</sup></li> </ul>
PV_PER	INPUT	WORD	<p>PROCESS VARIABLE PERIPHERY / Istwert Peripherie</p> <ul style="list-style-type: none"> <li>Der Istwert in Peripherieformat wird am Eingang <i>PV_PER</i> mit dem Regler verschaltet.</li> <li>Default: W#16#0000</li> </ul>
GAIN	INPUT	REAL	<p>PROPORTIONAL GAIN / Proportionalbeiwert</p> <ul style="list-style-type: none"> <li>Der Eingang <i>GAIN</i> gibt die Reglerv Verstärkung an.</li> <li>Default: 2.0</li> <li>Wertebereich: <math>\geq \text{CYCLE}</math></li> </ul>
TI	INPUT	TIME	<p>RESET TIME / Integrationszeit</p> <ul style="list-style-type: none"> <li>Der Eingang <i>TI</i> bestimmt das Zeitverhalten des Integrierers.</li> <li>Default: T#20s</li> <li>Wertebereich: <math>\geq \text{CYCLE}</math></li> </ul>
DEADB_W	INPUT	REAL	<p>DEAD BAND WIDTH / Totzonenbreite</p> <ul style="list-style-type: none"> <li>Die Regeldifferenz wird über eine Totzone geführt. Der Eingang <i>DEADB_W</i> bestimmt die Größe der Totzone.</li> <li>Default: 1.0</li> <li>Wertebereich: 0.0...100.0 (%) oder phys. Größe<sup>1</sup></li> </ul>
PV_FAC	INPUT	REAL	<p>PROCESS VARIABLE FACTOR / Istwertfaktor</p> <ul style="list-style-type: none"> <li>Der Eingang <i>PV_FAC</i> wird mit dem Istwert multipliziert. Der Eingang dient zur Anpassung des Istwertbereiches.</li> <li>Default: 1.0</li> </ul>

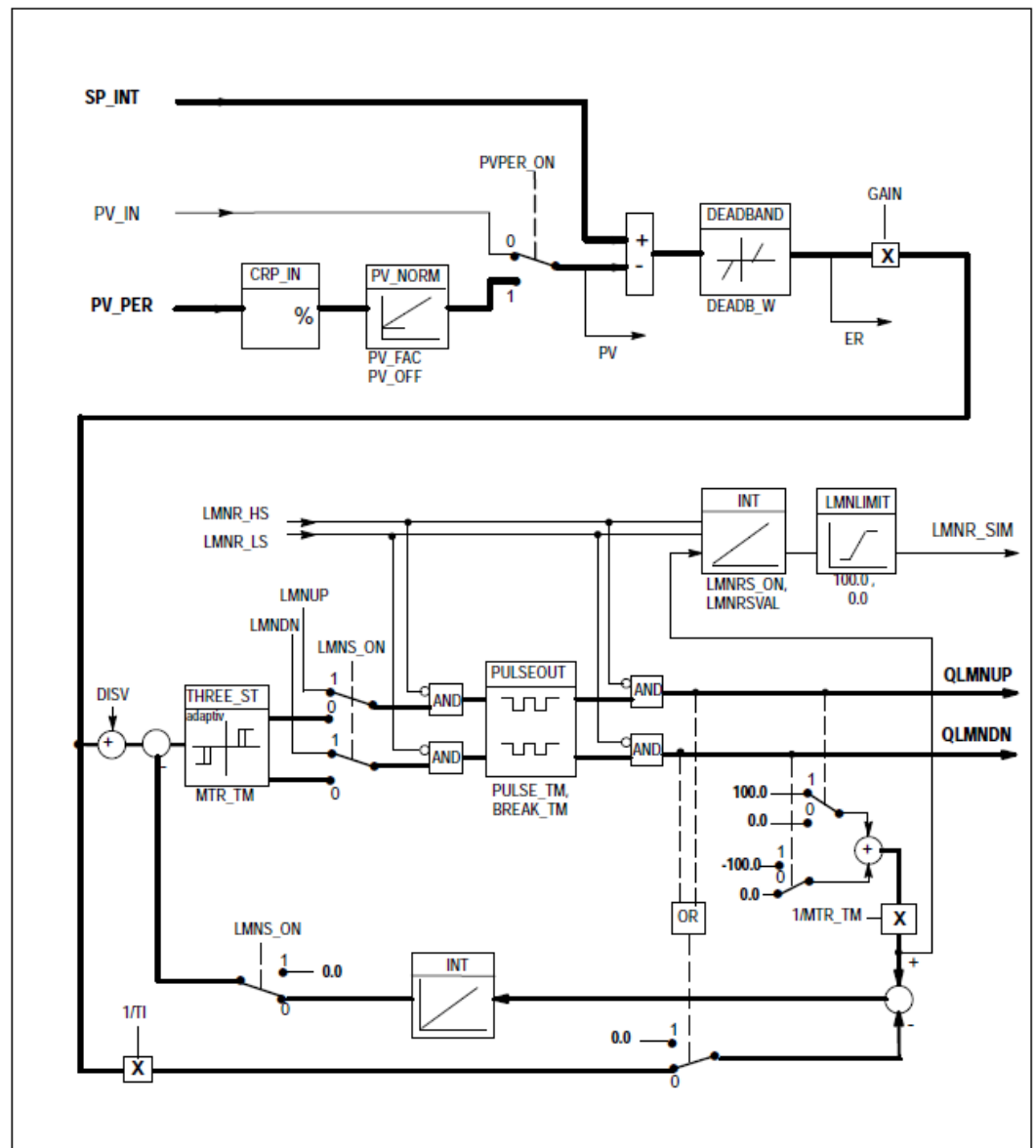
Parameter	Deklaration	Datentyp	Beschreibung
PV_OFF	INPUT	REAL	PROCESS VARIABLE OFFSET / Istwertoffset <ul style="list-style-type: none"> <li>Der Eingang <i>PV_OFF</i> wird mit dem Istwert addiert. Der Eingang dient zur Anpassung des Istwertbereiches.</li> <li>Default: 0.0</li> </ul>
PULSE_TM	INPUT	TIME	MINIMUM PULSE TIME / Mindestimpulsdauer <ul style="list-style-type: none"> <li>Am Parameter <i>PULSE_TM</i> kann eine minimale Impulslänge parametrisiert werden.</li> <li>Default: 1.0</li> <li>Wertebereich: <math>\geq</math> <i>CYCLE</i></li> </ul>
BREAK_TM	INPUT	TIME	MINIMUM BREAK TIME / Mindestpausendauer <ul style="list-style-type: none"> <li>Am Parameter <i>BREAK_TM</i> kann eine minimale Pausenlänge parametrisiert werden.</li> <li>Default: 0.0</li> <li>Wertebereich: <math>\geq</math> <i>CYCLE</i></li> </ul>
MTR_TM	INPUT	TIME	MOTOR MANIPULATED VALUE / Motorstellzeit <ul style="list-style-type: none"> <li>Am Parameter <i>MTR_TM</i> wird die Laufzeit des Stellventils vom Anschlag zu Anschlag eingetragen.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0...100. 0 (%) oder phys. Größe<sup>2</sup></li> <li>Wertebereich: <math>\geq</math> <i>CYCLE</i></li> </ul>
DISV	INPUT	REAL	DISTURBANCE VARIABLE / Störgröße <ul style="list-style-type: none"> <li>Für eine Störgrößenaufschaltung wird die Störgröße am Eingang <i>DISV</i> verschaltet.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0...100. 0 (%) oder phys. Größe<sup>2</sup></li> </ul>
QLMNUP	OUTPUT	BOOL	MANIPULATED SIGNAL UP / Stellwertsignal Hoch <ul style="list-style-type: none"> <li>Ist der Ausgang <i>QLMNUP</i> gesetzt, soll das Stellventil geöffnet werden.</li> <li>Default: FALSE</li> </ul>
QLMNDN	OUTPUT	BOOL	MANIPULATED SIGNAL DOWN / Stellwertsignal Tief <ul style="list-style-type: none"> <li>Ist der Ausgang <i>QLMNDN</i> gesetzt, soll das Stellventil geschlossen werden.</li> <li>Default: FALSE</li> </ul>
PV	OUTPUT	REAL	PROCESS VARIABLE / Istwert <ul style="list-style-type: none"> <li>Am Ausgang <i>PV</i> wird der effektiv wirkende Istwert ausgegeben.</li> <li>Default: 0.0</li> </ul>
ER	OUTPUT	REAL	ERROR SIGNAL / Regeldifferenz <ul style="list-style-type: none"> <li>Am Ausgang <i>ER</i> wird die effektiv wirkende Regeldifferenz ausgegeben.</li> <li>Default: 0.0</li> </ul>

1) Parameter im Soll-, Istwertzweig mit gleicher Einheit

2) Parameter im Stellwertzweig mit gleicher Einheit

<b>Anwendung</b>	<p>Der Regler kann als PI-Festwertregler einzeln oder in unterlagerten Regelkreisen bei Kaskaden-, Mischungs- oder Verhältnisregelungen eingesetzt werden, jedoch nicht als Führungsregler. Die Arbeitsweise basiert auf dem PI Regelalgorithmus des Abtastreglers und wird um die Funktionsglieder zur Erzeugung des binären Ausgangssignals aus dem analogen Stellsignal ergänzt.</p> <p>Neben den Funktionen im Istwertzweig realisiert der FB einen fertigen PI Regler mit binärem Stellwertausgang und Beeinflussungsmöglichkeit des Stellwertes von Hand. Der Schrittreger arbeitet ohne Stellungsrückmeldung.</p>
<b>Sollwertzweig</b>	Der Sollwert wird am Eingang <i>SP_INT</i> im Gleitpunktformat eingegeben.
<b>Istwertzweig</b>	<p>Der Istwert kann im Peripherie- und im Gleitpunktformat eingelesen werden. Die Funktion <i>CRP_IN</i> wandelt den Peripheriewert <i>PV_PER</i> in ein Gleitpunktformat von -100 .... +100 % nach folgender Vorschrift um:</p> $\text{Ausgang von } CPR\_IN = PV\_PER * \frac{100}{27648}$ <p>Die Funktion <i>PV_NORM</i> normiert den Ausgang von <i>CRP_IN</i> nach folgender Vorschrift:</p> $\text{Ausgang von } PV\_NORM = (\text{Ausgang von } CPR\_IN) * PV\_FAC + PV\_OFF$ <p><i>PV_FAC</i> ist mit 1 und <i>PV_OFF</i> ist mit 0 vorbelegt.</p>
<b>Regeldifferenzbildung</b>	Die Differenz von Soll- und Istwert ergibt die Regeldifferenz. Zur Unterdrückung einer kleinen Dauerschwingung aufgrund der Stellgrößen-Quantisierung (z.B. bei einer Puls-Breitenmodulation mit PULSEGEN) wird die Regeldifferenz über eine Totzone (DEAD-BAND) geleitet. Bei <i>DEADB_W</i> = 0 ist die Totzone ausgeschaltet.
<b>PI-Schrittalgorithmus</b>	Der FB arbeitet ohne Stellungsrückmeldung. Der I-Anteil des PI-Algorithmus und die gedachte Stellungsrückmeldung werden in einem Integrator (INT) berechnet und als Rückführungswert mit dem verbliebenen P-Anteil verglichen. Die Differenz geht auf ein Dreipunktglied (THREE_ST) und einen Impulsformer (PULSEOUT), der die Impulse für das Stellventil bildet. Über eine Adaption der Ansprechschwelle des Dreipunktgliedes wird die Schalthäufigkeit des Reglers reduziert.
<b>Störgrößenaufschaltung</b>	Am Eingang <i>DISV</i> kann eine Störgröße additiv aufgeschaltet werden.
<b>Betriebszustände</b>	<p><i>Neustart/Wiederanlauf</i></p> <ul style="list-style-type: none"> <li>■ Der FB 42 CONT_S verfügt über eine Neustartroutine, die durchlaufen wird, wenn der Eingangs-Parameter <i>COM_RST</i> = TRUE gesetzt ist.</li> <li>■ Alle anderen Ausgänge werden auf ihre Vorbelegungswerte gesetzt.</li> </ul>
<b>Fehlerinformationen</b>	Der Baustein enthält keine Fehlerprüfung, deshalb wird keine Fehlerinformation ausgegeben.

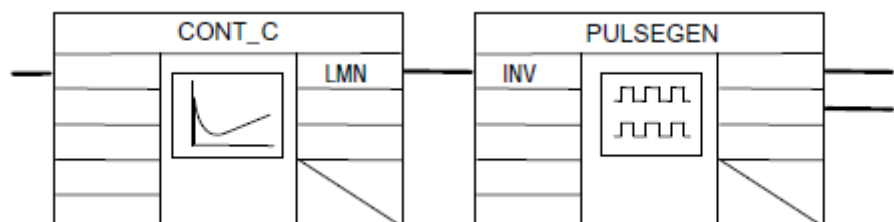
## Blockschaltbild



## 4.5.3 FB 43 - PULSGEN - Impulsformen

## Beschreibung

Der FB 43 PULSGEN dient zum Aufbau eines PID-Reglers mit Impulsausgang für proportionale Stellglieder. Mit dem FB 43 lassen sich PID-Zwei- oder Dreipunktregler mit Pulsbreitenmodulation aufbauen. Die Funktion wird meistens in Verbindung mit dem kontinuierlichen Regler CONT\_C angewendet.



## Parameter

Parameter	Deklaration	Datentyp	Beschreibung
INV	INPUT	REAL	<p>INPUT VARIABLE / Eingangsvariable</p> <ul style="list-style-type: none"> <li>Am Eingangsparameter <i>INV</i> wird eine analoge Stellwertgröße aufgeschaltet.</li> <li>Default: 0.0</li> <li>Wertebereich: -100.0...100.0 (%)</li> </ul>
PER_TM	INPUT	TIME	<p>PERIOD TIME / Periodendauer</p> <ul style="list-style-type: none"> <li>Am Parameter <i>PER_TM</i> wird die konstante Periodendauer der Pulsbreitenmodulation eingegeben. Sie entspricht der Abtastzeit des Reglers. Das Verhältnis Abtastzeit des Impulsformers zu Abtastzeit des Reglers bestimmt die Genauigkeit der Pulsbreitenmodulation.</li> <li>Default: T#1s</li> <li>Wertebereich: <math>\geq 20 \cdot CYCLE</math></li> </ul>
P_B_TM	INPUT	TIME	<p>MINIMUM PULSE/BREAK TIME / Mindestimpuls- bzw. Mindestpausendauer</p> <ul style="list-style-type: none"> <li>Am Parameter <i>P_B_TM</i> kann eine minimale Impuls- bzw. Pausenlänge parametrisiert werden.</li> <li>Default: T#50ms</li> <li>Wertebereich: <math>\geq CYCLE</math></li> </ul>
RATIOFAC	INPUT	REAL	<p>RATIO FACTOR / Verhältnisfaktor</p> <ul style="list-style-type: none"> <li>Durch den Eingangsparameter <i>RATIOFAC</i> kann das Verhältnis der Dauer von negativen zu positiven Impulsen verändert werden. Bei einem thermischen Prozess können damit unterschiedliche Zeitkonstanten für Heizen und Kühlen (z.B. Prozess mit elektrischer Heizung und Wasserkühlung) kompensiert werden.</li> <li>Default: 1.0</li> <li>Wertebereich: 0.1 ...10.0</li> </ul>
STEP3_ON	INPUT	BOOL	<p>THREE STEP CONTROL ON / Dreipunktregelung einschalten</p> <ul style="list-style-type: none"> <li>Am Eingangsparameter <i>STEP3_ON</i> wird die entsprechende Betriebsart aktiviert. Bei Dreipunktregelung arbeiten beide Ausgangssignale.</li> <li>Default: TRUE</li> </ul>
ST2BI_ON	INPUT	BOOL	<p>TWO STEP CONTROL FOR BIPOLAR MANIPULATED VALUE RANGE ON / Zweipunktregelung für bipolaren Stellwertbereich einschalten</p> <ul style="list-style-type: none"> <li>Am Eingangsparameter <i>ST2BI_ON</i> kann zwischen den Betriebsarten "Zweipunktregelung für bipolaren Stellwertbereich" und "Zweipunktregelung für unipolaren Stellwertbereich" gewählt werden. Hierbei muss <i>STEP3_ON</i> = FALSE sein.</li> <li>Default: FALSE</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
MAN_ON	INPUT	BOOL	<p>MANUAL MODE ON / Handbetrieb einschalten</p> <ul style="list-style-type: none"> <li>■ Durch Setzen des Eingangsparameter <i>MAN_ON</i> können die Ausgangssignale von Hand gesetzt werden.</li> <li>■ Default: FALSE</li> </ul>
POS_P_ON	INPUT	BOOL	<p>POSITIVE MODE ON / positiver Impuls ein</p> <ul style="list-style-type: none"> <li>■ Bei Handbetrieb Dreipunktregelung kann am Eingangsparameter <i>POS_P_ON</i> das Ausgangssignal <i>QPOS_P</i> bedient werden. Bei Handbetrieb Zweipunktregelung wird <i>QNEG_P</i> immer invertiert zu <i>QPOS_P</i> gesetzt.</li> <li>■ Default: FALSE</li> </ul>
NEG_P_ON	INPUT	BOOL	<p>NEGATIVE PULSE ON / negativer Impuls ein</p> <ul style="list-style-type: none"> <li>■ Bei Handbetrieb Dreipunktregelung kann am Eingangsparameter <i>NEG_P_ON</i> das Ausgangssignal <i>QNEG_P</i> bedient werden. Bei Handbetrieb Zweipunktregelung wird <i>QNEG_P</i> immer invertiert zu <i>QPOS_P</i> gesetzt.</li> <li>■ Default: FALSE</li> </ul>
SYN_ON	INPUT	BOOL	<p>SYNCHRONISATION ON / Synchronisation einschalten</p> <ul style="list-style-type: none"> <li>■ Es besteht die Möglichkeit durch Setzen des Eingangsparameters <i>SYN_ON</i> die Impulsausgabe mit dem Baustein, der die Eingangsgröße <i>INV</i> aktualisiert, automatisch zu synchronisieren. Damit ist gewährleistet, dass eine sich ändernde Eingangsgröße auch schnellstmöglich als Impuls ausgegeben wird.</li> <li>■ Default: TRUE</li> </ul>
COM_RST	INPUT	BOOL	<p>COMPLETE RESTART / Neustart</p> <ul style="list-style-type: none"> <li>■ Der Baustein hat eine Neustartroutine, die bearbeitet wird, wenn der Eingang <i>COM_RST</i> gesetzt ist.</li> <li>■ Default: FALSE</li> </ul>
CYCLE	INPUT	TIME	<p>SAMPLE TIME / Abtastzeit</p> <ul style="list-style-type: none"> <li>■ Die Zeit zwischen den Bausteinaufrufen muss konstant sein. Der Eingang <i>CYCLE</i> gibt die Zeit zwischen den Bausteinaufrufen an.</li> <li>■ Default: T#10ms</li> <li>■ Wertebereich: <math>\geq 1\text{ms}</math></li> </ul>

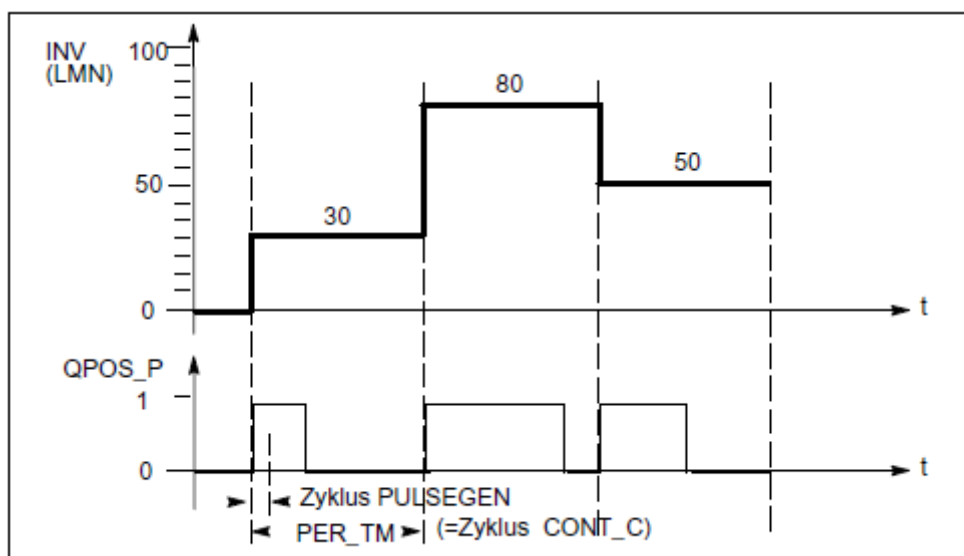
Parameter	Deklaration	Datentyp	Beschreibung
QPOS_P	OUTPUT	BOOL	OUTPUT POSITIVE PULSE / Ausgangssignal positiver Impuls <ul style="list-style-type: none"> <li>Der Ausgangsparameter <i>QPOS_P</i> ist gesetzt, wenn ein Impuls ausgegeben werden soll. Bei Dreipunktregelung ist es der positive Impuls. Bei Zweipunktregelung wird <i>QNEG_P</i> immer invertiert zu <i>QPOS_P</i> gesetzt.</li> <li>Default: FALSE</li> </ul>
QNEG_P	OUTPUT	BOOL	OUTPUT NEGATIVE PULSE / Ausgangssignal negativer Impuls <ul style="list-style-type: none"> <li>Der Ausgangsparameter <i>QNEG_P</i> ist gesetzt, wenn ein Impuls ausgegeben werden soll. Bei Dreipunktregelung ist es der negative Impuls. Bei Zweipunktregelung wird <i>QNEG_P</i> immer invertiert zu <i>QPOS_P</i> gesetzt.</li> <li>Default: FALSE</li> </ul>



Die Werte der Eingangsparameter werden im Baustein nicht begrenzt; eine Prüfung der Parameter findet nicht statt.

## Anwendung

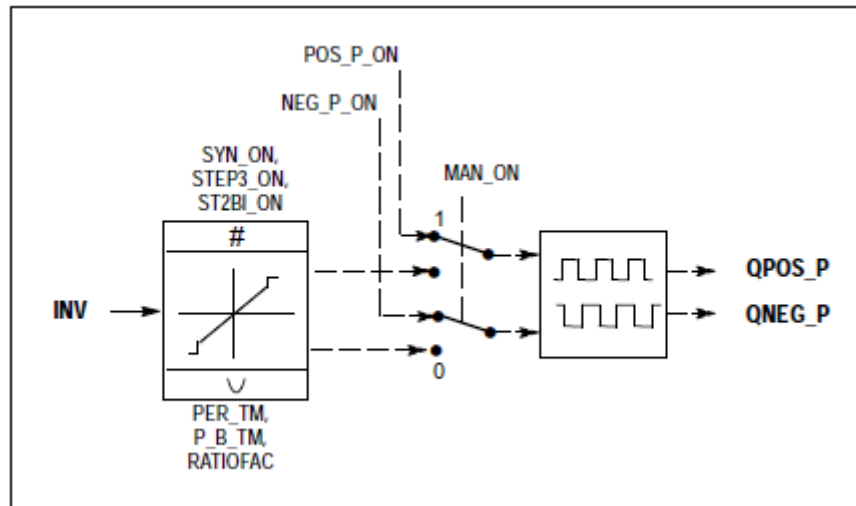
Die Funktion PULSGEN transformiert die Eingangsgröße *INV* (= LMN des PID-Reglers) durch Modulation der Impulsbreite in eine Impulsfolge mit konstanter Periodendauer, welche der Zykluszeit, mit der die Eingangsgröße aktualisiert wird, entspricht und in *PER\_TM* parametrisiert werden muss. Die Dauer eines Impulses pro Periodendauer ist proportional der Eingangsgröße. Dabei ist der durch *PER\_TM* parametrisierte Zyklus nicht identisch mit dem Bearbeitungszyklus des FB PULSGEN. Vielmehr setzt sich ein Zyklus *PER\_TM* aus mehreren Bearbeitungszyklen des FB PULSGEN zusammen, wobei die Anzahl der FB PULSGEN-Aufrufe pro *PER\_TM* Zyklus ein Maß für die Genauigkeit der Impulsbreite darstellt.



Eine Eingangsgröße 30% und 10 FB PULSGEN-Aufrufe pro  $PER\_TM$  bedeuten also:

- "1" am Ausgang  $QPOS$  für die ersten drei Aufrufe des FB PULSGEN (30% von 10 Aufrufen)
- "0" am Ausgang  $QPOS$  für sieben weitere Aufrufe des FB PULSGEN (70% von 10 Aufrufen)

### Blockschaltbild



### Stellwertgenauigkeit

Durch ein "Abtastverhältnis" von 1:10 (CONT\_C-Aufrufe zu PULSGEN-Aufrufe) ist die Stellwertgenauigkeit in diesem Beispiel auf 10% beschränkt, d.h. vorgegebene Eingangswerte  $INV$  können nur im Raster von 10% auf eine Impulslänge am Ausgang  $QPOS$  abgebildet werden. Entsprechend erhöht sich die Genauigkeit mit der Anzahl der FB PULSGEN-Aufrufe pro CONT\_C-Aufruf. Wird z.B. PULSGEN 100 mal häufiger aufgerufen als CONT\_C, so erreicht man eine Auflösung von 1% des Stellwertbereiches.

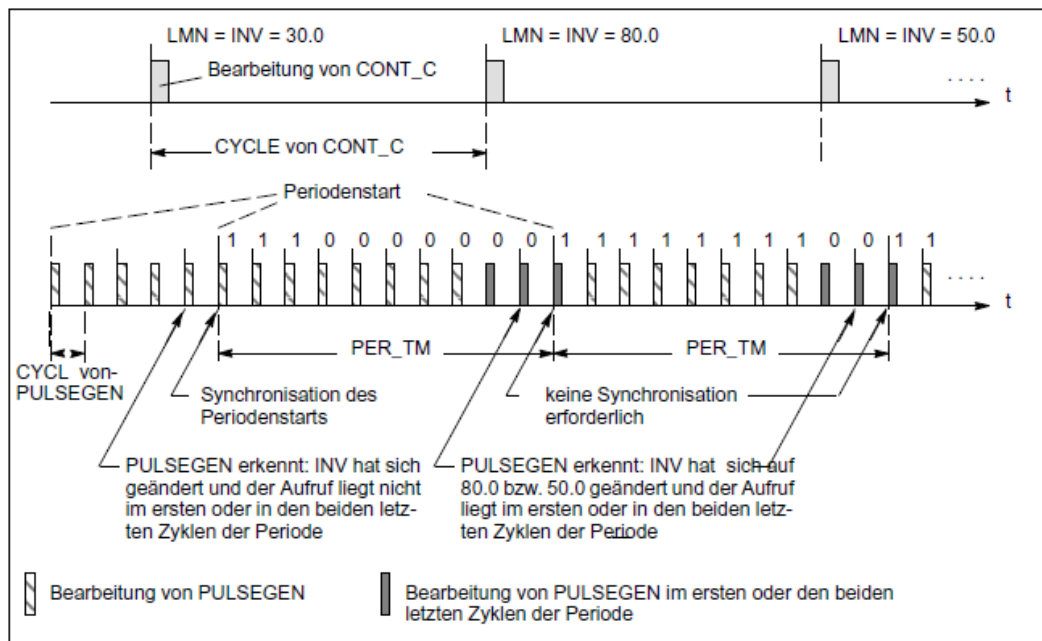


*Die Untersetzung der Aufrufhäufigkeit müssen Sie selbst programmieren.*

### Automatische Synchronisation

Es besteht die Möglichkeit die Impulsausgabe mit dem Baustein, der die Eingangsgröße  $INV$  (z.B. CONT\_C) aktualisiert, automatisch zu synchronisieren. Damit ist gewährleistet, dass eine ändernde Eingangsgröße auch schnellstmöglich als Impuls ausgegeben wird. Der Impulsformer wertet immer im Zeitabstand der Periodendauer  $PER\_TM$  die Eingangsgröße  $INV$  aus und wandelt den Wert in ein Impulssignal der entsprechenden Länge. Da aber  $INV$  meistens in einer langsameren Weckalarmebene berechnet wird, sollte der Impulsformer möglichst schnell nach der Aktualisierung von  $INV$  mit der Umwandlung des diskreten Wertes in ein Impulssignal beginnen. Dazu kann der Baustein den Start der Periode nach folgendem Verfahren selbst synchronisieren:

- Hat sich  $INV$  geändert und befindet sich der Baustein aufruf nicht im ersten oder in den letzten zwei Aufrufzyklen einer Periode, so wird eine Synchronisation durchgeführt. Die Impulsdauer wird neu berechnet und beim nächsten Zyklus mit einer neuen Periode wird mit der Ausgabe begonnen.



Die automatische Synchronisation lässt sich am Eingang SYN\_ON (= FALSE) abschalten.



Durch den Beginn der neuen Periode wird der Altwert von INV (d.h. von LMN) nach erfolgter Synchronisation mehr oder weniger ungenau auf das Impulssignal abgebildet.

## Betriebsarten

Je nach Parametrierung des Impulsformers können PID-Regler mit Dreipunktverhalten oder mit bipolarem bzw. unipolarem Zweipunktausgang konfiguriert werden. Nachstehende Tabelle zeigt die Einstellung der Schalterkombinationen für die möglichen Betriebsarten.

Betriebsart	Schalter		
	MAN_ON	STEP3_ON	ST2BI_ON
Dreipunktregelung	FALSE	TRUE	beliebig
Zweipunktregelung mit bipolarem Stellbereich (-100 % ... 100 %)	FALSE	FALSE	TRUE
Zweipunktregelung mit unipolarem Stellbereich (0 % ... 100 %)	FALSE	FALSE	FALSE
Handbetrieb	TRUE	beliebig	beliebig

## Dreipunktregelung

In der Betriebsart Dreipunktregelung können drei Zustände des Stellsignals erzeugt werden. Dazu werden die Zustandswerte der binären Ausgangssignale QPOS\_P und QNEG\_P den jeweiligen Betriebszuständen des Stellgliedes zugeordnet. Die Tabelle zeigt das Beispiel einer Temperaturregelung:

Ausgangssignale	Stellglied		
	heizen	aus	kühlen
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

Aus der Eingangsgröße wird über eine Kennlinie die Impulsdauer berechnet. Die Form dieser Kennlinie wird durch die Mindestimpuls- bzw. Mindestpausendauer und dem Verhältnissfaktor definiert. Der normale Wert für den Verhältnissfaktor ist 1. Die Knickpunkte an den Kennlinien werden durch die Mindestimpuls- bzw. Mindestpausendauer verursacht.

■ **Mindestimpuls- bzw. Mindestpausendauer**

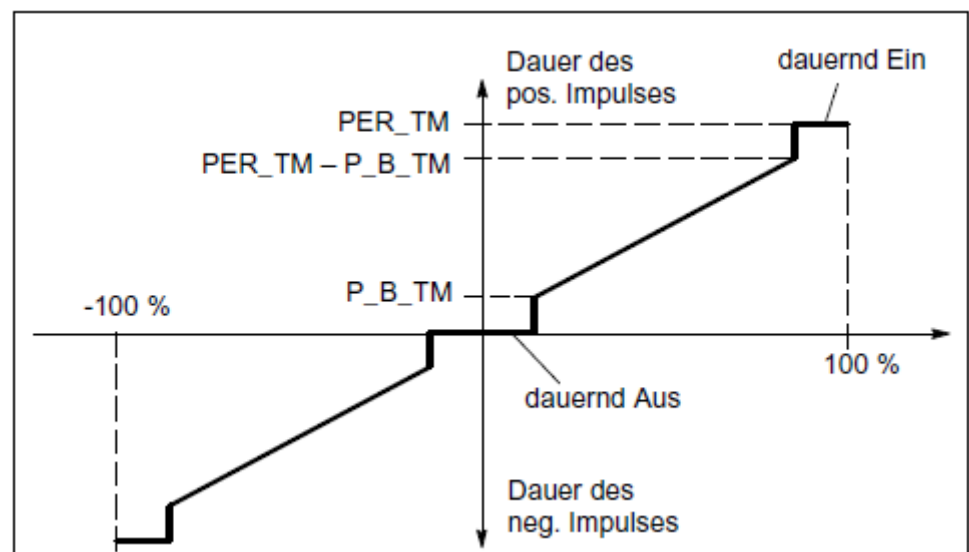
Eine richtig parametrisierte Mindestimpuls- bzw. Mindestpausendauer  $P\_B\_TM$  kann kurze Ein- oder Ausschaltzeiten, die die Lebensdauer von Schaltgliedern und Stell-einrichtungen beeinträchtigen, verhindern.



*Kleine Absolutwerte der Eingangsgröße LMN, die eine Impulsdauer kleiner als  $P\_B\_TM$  erzeugen würden, werden unterdrückt. Große Eingangswerte, die eine Impulsdauer größer als  $(PER\_TM - P\_B\_TM)$  erzeugen würden, werden auf 100 % bzw. -100 % gesetzt.*

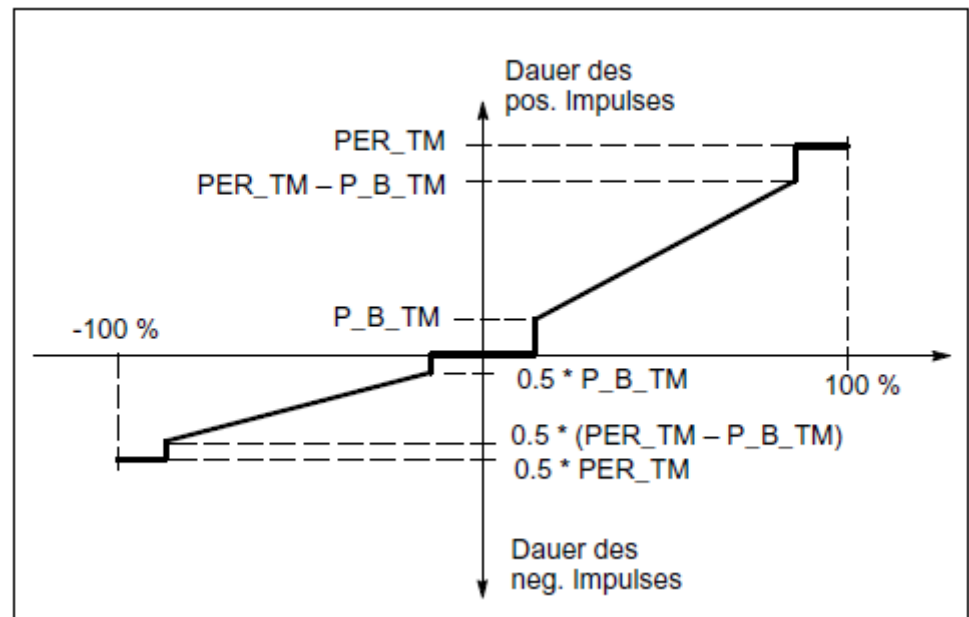
Die Dauer der positiven oder negativen Impulse errechnet sich aus Eingangsgröße (in %) mal Periodendauer:

$$\text{Impulsdauer} = \frac{INV}{100} * PER\_TM$$



### Dreipunktregelung unsymmetrisch

Über den Verhältnissfaktor  $RATIOFAC$  kann das Verhältnis der Dauer von positiven zu negativen Impulsen verändert werden. Bei einem thermischen Prozess lassen sich damit z.B. unterschiedliche Streckenzeitkonstanten für Heizen und Kühlen berücksichtigen. Der Verhältnissfaktor beeinflusst auch die Mindestimpuls- bzw. Mindestpausendauer. Verhältnissfaktor  $< 1$  bedeutet, der Ansprechwert für negative Impulse wird mit dem Verhältnissfaktor multipliziert.



■ **Verhältnissfaktor < 1**

Die aus Eingangsgröße mal Periodendauer berechnete Impulsdauer am negativen Impulsausgang wird um den Verhältnissfaktor verkürzt.

$$\text{positive Impulsdauer} = \frac{INV}{100} * PER\_TM$$

$$\text{negative Impulsdauer} = \frac{INV}{100} * PER\_TM * \text{RATIOFAC}$$

■ **Verhältnissfaktor > 1**

Die aus Eingangsgröße mal Periodendauer berechnete Impulsdauer am positiven Impulsausgang wird um den Verhältnissfaktor verkürzt.

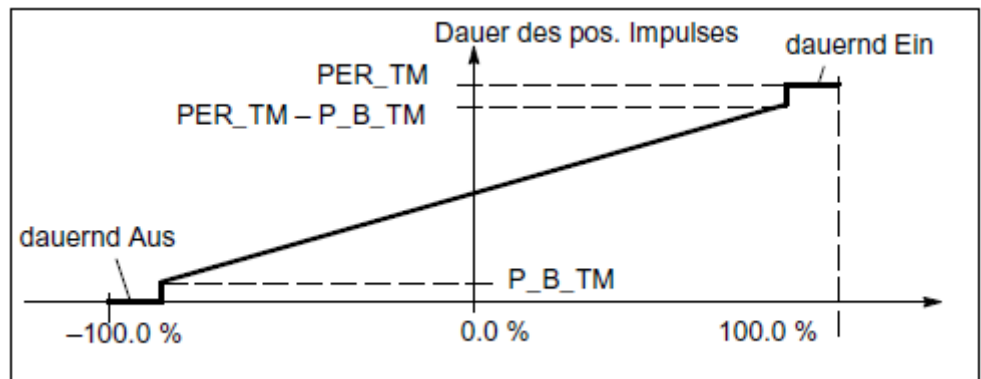
$$\text{negative Impulsdauer} = \frac{INV}{100} * PER\_TM$$

$$\text{positive Impulsdauer} = \frac{INV}{100} * \frac{PER\_TM}{\text{RATIOFAC}}$$

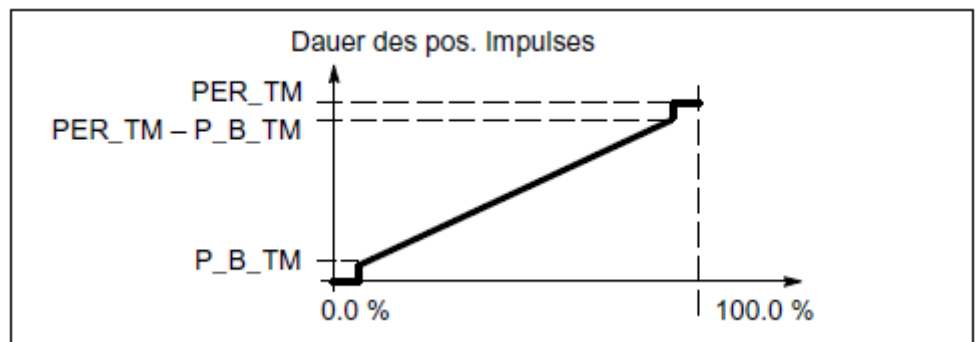
## Zweipunktregelung

Bei der Zweipunktregelung wird nur der positive Impulsausgang QPOS\_P von PUL-SEGEN mit dem betreffenden Ein/Aus-Stellglied verbunden. Je nach genutztem Stellwertbereich hat der Zweipunktregler einen bipolaren oder einen unipolaren Stellwertbereich.

■ **Zweipunktregelung mit bipolarem Stellwertbereich (-100 % ... 100 %)**



- Zweipunktregelung mit unipolarem Stellwertbereich (0 % ... 100 %)



An *QNEG\_P* steht das negierte Ausgangssignal zur Verfügung, falls die Verschaltung des Zweipunktreglers im Regelkreis ein logisch invertiertes Binärsignal für die Stellimpulse erfordert.

Impuls	Stellglied	
	ein	aus
QPOS_P	TRUE	FALSE
QNEG_P	FALSE	TRUE

### Handbetrieb bei Zwei- bzw. Dreipunkt- Regelung

Im Handbetrieb (*MAN\_ON* = TRUE) können die Binärausgänge des Dreipunkt- bzw. Zweipunktreglers über die Signale *POS\_P\_ON* und *NEG\_P\_ON* unabhängig von *INV* gesetzt werden.

	POS_P_ON	NEG_P_ON	QPOS_P	QNEG_P
Dreipunktregelung	FALSE	FALSE	FALSE	FALSE
	TRUE	FALSE	TRUE	FALSE
	FALSE	TRUE	FALSE	TRUE
	TRUE	TRUE	FALSE	FALSE
Zweipunktregelung	FALSE	beliebig	FALSE	TRUE
	TRUE	beliebig	TRUE	FALSE

**Betriebszustände***Neustart/Wiederanlauf*

- Bei Neustart werden alle Signalausgänge auf Null gesetzt.

**Fehlerinformationen**

Der Baustein enthält keine Fehlerprüfung, deshalb wird keine Fehlerinformation ausgegeben.

**4.5.4 FB 58 - TCONT\_CP - Kontinuierliches Temperaturregeln****Beschreibung**

Der FB 58 TCONT\_CP dient zum Regeln von Temperaturprozessen mit kontinuierlicher oder impulsförmiger Ansteuerung. Über die Parametrierung können Sie Teilfunktionen des PID-Reglers zu- oder abschalten und damit diesen an die Regelstrecke anpassen.

**Parameter**

Parameter	Deklaration	Datentyp	Beschreibung
PV_IN	INPUT	REAL	PROCESS VARIABLE IN / Istwert Eingang <ul style="list-style-type: none"> <li>■ Am Eingang <i>PV_IN</i> kann ein Inbetriebsetzungswert parametrierbar oder ein externer Istwert im Gleitpunktformat verschaltet werden.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: Abhängig von den eingesetzten Sensoren</li> </ul>
PV_PER	INPUT	WORD	PROCESS VARIABLE PERIPHERY / Istwert Peripherie <ul style="list-style-type: none"> <li>■ Der Istwert in Peripherieformat wird am Eingang <i>PV_PER</i> mit dem Regler verschaltet.</li> <li>■ Default: 0</li> </ul>
DISV	INPUT	REAL	DISTURBANCE VARIABLE / Störgröße <ul style="list-style-type: none"> <li>■ Für eine Störgrößenaufschaltung wird die Störgröße am Eingang <i>DISV</i> verschaltet.</li> <li>■ Default: 0.0</li> </ul>
INT_HPOS	INPUT	BOOL	INTEGRAL ACTION HOLD IN POSITIVE DIRECTION / I-Anteil blockieren in positive Richtung <ul style="list-style-type: none"> <li>■ Der Ausgang des Integrierers kann in positive Richtung blockiert werden. Hierzu muss der Eingang <i>INT_HPOS</i> auf TRUE gesetzt werden. Bei einer Kaskaderegulierung wird <i>INT_HPOS</i> des Führungsreglers mit <i>QLMN_HLM</i> des Folgereglers verschaltet.</li> <li>■ Default: FALSE</li> </ul>
INT_HNEG	INPUT	BOOL	INTEGRAL ACTION HOLD IN NEGATIVE DIRECTION / I-Anteil blockieren in negative Richtung <ul style="list-style-type: none"> <li>■ Der Ausgang des Integrierers kann in negative Richtung blockiert werden. Hierzu muss der Eingang <i>INT_HNEG</i> auf TRUE gesetzt werden. Bei einer Kaskaderegulierung wird <i>INT_HNEG</i> des Führungsreglers mit <i>QLMN_LLM</i> des Folgereglers verschaltet.</li> <li>■ Default: FALSE</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
SELECT	INPUT	BOOL	<p>SELECTION OF CALL PID AND PULSE GENERATOR / Auswahl des Aufrufverhaltens von PID und Impulsformer</p> <ul style="list-style-type: none"> <li>■ Ist der Impulsformer eingeschaltet, gibt es mehrere Möglichkeiten den PID-Algorithmus und Impulsformer aufzurufen: <ul style="list-style-type: none"> <li>– <i>SELECT</i> = 0: Der Regler wird in einer schnellen Weckalarmebene aufgerufen und es werden PID-Algorithmus und Impulsformer bearbeitet.</li> <li>– <i>SELECT</i> = 1: Der Regler wird im OB 1 aufgerufen und es wird nur der PID-Algorithmus bearbeitet.</li> <li>– <i>SELECT</i> = 2: Der Regler wird in einer schnellen Weckalarmebene aufgerufen und es wird nur der Impulsformer bearbeitet.</li> <li>– <i>SELECT</i> = 3: Der Regler wird in einer langsamen Weckalarmebene aufgerufen und es wird nur der PID-Algorithmus bearbeitet.</li> </ul> </li> <li>■ Default: 0</li> <li>■ Wertebereich: 0 ... 3</li> </ul>
PV	OUTPUT	REAL	<p>PROCESS VARIABLE / Istwert</p> <ul style="list-style-type: none"> <li>■ Am Ausgang <i>PV</i> wird der effektiv wirkende Istwert ausgegeben.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: abhängig von den eingesetzten Sensoren</li> </ul>
LMN	OUTPUT	REAL	<p>MANIPULATED VALUE / Stellwert</p> <ul style="list-style-type: none"> <li>■ Am Ausgang <i>LMN</i> wird der effektiv wirkende Stellwert in Gleitpunktformat ausgegeben.</li> <li>■ Default: 0.0</li> </ul>
LMN_PER	OUTPUT	WORD	<p>MANIPULATED VALUE PERIPHERY / Stellwert Peripherie</p> <ul style="list-style-type: none"> <li>■ Der Stellwert in Peripherieformat wird am Ausgang <i>LMN_PER</i> mit dem Regler verschaltet.</li> <li>■ Default: 0</li> </ul>
QPULSE	OUTPUT	BOOL	<p>QUTPUT PULSE SIGNAL / Pulsausgang</p> <ul style="list-style-type: none"> <li>■ Der Stellwert wird pulsweitenmoduliert am Ausgang <i>QPULSE</i> ausgegeben.</li> <li>■ Default: FALSE</li> </ul>
QLMN_HLM	OUTPUT	BOOL	<p>HIGH LIMIT OF MANIPULATED VALUE REACHED / Obere Begrenzung des Stellwertes angesprochen</p> <ul style="list-style-type: none"> <li>■ Der Stellwert wird immer auf eine obere und untere Grenze begrenzt. Der Ausgang <i>QLMN_HLM</i> meldet die Überschreitung der oberen Begrenzung.</li> <li>■ Default: FALSE</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
QLMN_LLM	OUTPUT	BOOL	<p>LOW LIMIT OF MANIPULATED VALUE REACHED / Untere Begrenzung des Stellwertes angesprochen</p> <ul style="list-style-type: none"> <li>Der Stellwert wird immer auf eine obere und untere Grenze begrenzt. Der Ausgang <i>QLMN_LLM</i> meldet die Überschreitung der unteren Begrenzung.</li> <li>Default: FALSE</li> </ul>
QC_ACT	OUTPUT	BOOL	<p>NEXT CYCLE, THE CONTINUOUS CONTROLLER IS WORKING / Kontinuierlicher Regleranteil wird beim nächsten Aufruf bearbeitet</p> <ul style="list-style-type: none"> <li>Der Parameter zeigt an, ob beim nächsten Baustei- naufruf der kontinuierliche Regleranteil bearbeitet wird (nur relevant, wenn <i>SELECT</i> den Wert 0 oder 1 hat).</li> <li>Default: TRUE</li> </ul>
CYCLE	INPUT/ OUTPUT	REAL	<p>SAMPLE TIME OF CONTINUOUS CONTROLLER [s] / Abtastzeit des kontinuierlichen Reglers [s]</p> <ul style="list-style-type: none"> <li>Vorgabe der Abtastzeit für den PID-Algorithmus. Der Optimierer berechnet in Phase 1 die Abtastzeit und trägt sie in <i>CYCLE</i> ein.</li> <li>Default: 0.1s</li> <li>Wertebereich: <math>\geq 1\text{ms}</math></li> </ul>
CYCLE_P	INPUT/ OUTPUT	REAL	<p>SAMPLE TIME OF PULSE GENERATOR [s] / Abtastzeit des Impulsformers [s]</p> <ul style="list-style-type: none"> <li>An diesem Eingang geben Sie die Abtastzeit für den Impulsformer-Anteil ein. Der FB 58 "TCONT_CP" berechnet in Phase 1 die Abtastzeit und trägt sie in <i>CYCLE_P</i> ein.</li> <li>Default: 0.2s</li> <li>Wertebereich: <math>\geq 1\text{ms}</math></li> </ul>
SP_INT	INPUT/ OUTPUT	REAL	<p>INTERNAL SETPOINT / Interner Sollwert</p> <ul style="list-style-type: none"> <li>Der Eingang <i>SP_INT</i> dient zur Vorgabe eines Sollwertes.</li> <li>Default: 0.0</li> <li>Wertebereich des Istwertes</li> </ul>
MAN	INPUT/ OUTPUT	REAL	<p>MANUAL VALUE / Handwert</p> <ul style="list-style-type: none"> <li>Der Eingang <i>MAN</i> dient zur Vorgabe eines Handwertes. Im Automatikbetrieb wird er dem Stellwert nachgeführt.</li> <li>Default: 0.0</li> </ul>
COM_RST	INPUT/ OUTPUT	REAL	<p>COMPLETE RESTART / Neustart</p> <ul style="list-style-type: none"> <li>Der Baustein hat eine Initialisierungsroutine, die bearbeitet wird, wenn der Eingang <i>COM_RST</i> gesetzt ist.</li> <li>Default: FALSE</li> </ul>
MAN_ON	INPUT/ OUTPUT	REAL	<p>MANUAL OPERATION ON / Handbetrieb einschalten</p> <ul style="list-style-type: none"> <li>Ist der Eingang <i>MAN_ON</i> gesetzt, ist der Regelkreis unterbrochen. Als Stellwert wird der Handwert <i>MAN</i> vorgegeben.</li> <li>Default: TRUE</li> </ul>

## Interne Parameter

Parameter	Deklaration	Datentyp	Beschreibung
DEADB_W	INPUT	REAL	<p>DEAD BAND WIDTH / Totzonenbreite</p> <ul style="list-style-type: none"> <li>Die Regeldifferenz wird über eine Totzone geführt. Der Eingang <i>DEADB_W</i> bestimmt die Größe der Totzone.</li> <li>Default: 0.0</li> <li>Wertebereich: Abhängig von den eingesetzten Sensoren</li> </ul>
I_ITLVAL	INPUT	REAL	<p>INITIALIZATION VALUE OF THE INTEGRAL ACTION / Initialisierungswert für I-Anteil</p> <ul style="list-style-type: none"> <li>Der Ausgang des Integrierers kann am Eingang <i>I_ITL_ON</i> gesetzt werden. Am Eingang <i>I_ITLVAL</i> steht der Initialisierungswert.</li> <li>Bei Neustart <i>COM_RST</i> = TRUE wird der I-Anteil auf den Initialisierungswert gesetzt.</li> <li>Default: 0.0</li> <li>Wertebereich: 0 bis 100 %</li> </ul>
LMN_HLM	INPUT	REAL	<p>MANIPULATED VARIABLE HIGH LIMIT / Stellwert obere Begrenzung</p> <ul style="list-style-type: none"> <li>Der Stellwert wird immer auf eine obere und untere Grenze begrenzt. Der Eingang <i>LMN_HLM</i> gibt die obere Begrenzung an.</li> <li>Default: 100.0</li> <li>Wertebereich: &gt; <i>LMN_LLM</i></li> </ul>
LMN_LLM	INPUT	REAL	<p>MANIPULATED VARIABLE LOW LIMIT / Stellwert untere Begrenzung</p> <ul style="list-style-type: none"> <li>Der Stellwert wird immer auf eine obere und untere Grenze begrenzt. Der Eingang <i>LMN_LLM</i> gibt die untere Begrenzung an.</li> <li>Default: 0.0</li> <li>Wertebereich: &lt; <i>LMN_HLM</i></li> </ul>
PV_FAC	INPUT	REAL	<p>PROCESS VARIABLE FACTOR / Istwertfaktor</p> <ul style="list-style-type: none"> <li>Der Eingang <i>PV_FAC</i> wird mit dem <i>PV_PER</i> multipliziert. Der Eingang dient zur Anpassung des Istwertbereiches.</li> <li>Default: 1.0</li> </ul>
PV_OFFS	INPUT	REAL	<p>PROCESS VARIABLE OFFSET / Istwertoffset</p> <ul style="list-style-type: none"> <li>Der Eingang <i>PV_OFFS</i> wird zum <i>PV_PER</i> addiert. Der Eingang dient zur Anpassung des Istwertbereiches.</li> <li>Default: 0.0</li> </ul>
LMN_FAC	INPUT	REAL	<p>MANIPULATED VARIABLE FACTOR / Stellwertfaktor</p> <ul style="list-style-type: none"> <li>Der Eingang <i>LMN_FAC</i> wird mit dem Stellwert multipliziert. Der Eingang dient zur Anpassung des Stellwertbereiches.</li> <li>Default: 1.0</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
LMN_OFFS	INPUT	REAL	<p>MANIPULATED VARIABLE OFFSET / Stellwertoffset</p> <ul style="list-style-type: none"> <li>Der Eingang <i>LMN_OFFS</i> wird zum Stellwert addiert. Der Eingang dient zur Anpassung des Stellwertbereiches.</li> <li>Default: 0.0</li> </ul>
PER_TM	INPUT	REAL	<p>PERIOD TIME [s] / Periodendauer [s]</p> <ul style="list-style-type: none"> <li>Am Parameter <i>PER_TM</i> wird die Periodendauer der Pulsbreitenmodulation eingegeben. Das Verhältnis Periodendauer zu Abtastzeit des Impulsformers bestimmt die Genauigkeit der Pulsbreitenmodulation.</li> <li>Default: 1.0 s</li> <li>Wertebereich: <math>\geq \text{CYCLE}</math></li> </ul>
P_B_TM	INPUT	REAL	<p>MINIMUM PULSE/BREAK TIME [s] / Mindestimpuls- bzw. Mindestpausendauer [s]</p> <ul style="list-style-type: none"> <li>Am Parameter <i>P_B_TM</i> kann eine minimale Impuls- bzw. Pausenlänge parametrisiert werden. <i>P_B_TM</i> wird intern auf <math>&gt; \text{CYCLE}_P</math> begrenzt.</li> <li>Default: 0.02 s</li> <li>Wertebereich: <math>\geq 0.0</math></li> </ul>
TUN_DLMN	INPUT	REAL	<p>DELTA MANIPULATED VARIABLE FOR PROCESS EXCITATION / Delta-Stellwert für Prozessanregung</p> <ul style="list-style-type: none"> <li>Die Prozessanregung für die Regleroptimierung erfolgt durch einen Stellwertsprung von <i>TUN_DLMN</i>.</li> <li>Default: 20.0</li> <li>Wertebereich: -100.0 ... 100.0 %</li> </ul>
PER_MODE	INPUT	INT	<p>PERIPHERY MODE / Peripherie Betriebsart</p> <ul style="list-style-type: none"> <li>An diesem Schalter können Sie den Typ der AE-Baugruppe eingeben. Der Istwert am Eingang <i>PV_PER</i> wird dadurch am Ausgang <i>PV</i> in °C normiert. <ul style="list-style-type: none"> <li><i>PER_MODE</i> = 0: Standard</li> <li><i>PER_MODE</i> = 1: Klima</li> <li><i>PER_MODE</i> = 2: Strom/Spannung</li> </ul> </li> <li>Default: 0</li> <li>Wertebereich: 0, 1, 2</li> </ul>
PVPER_ON	INPUT	BOOL	<p>PROCESS VARIABLE PERIPHERY ON / Istwert Peripherie einschalten</p> <ul style="list-style-type: none"> <li>Soll der Istwert von der Peripherie eingelesen werden, so muss der Eingang <i>PV_PER</i> mit der Peripherie verschaltet werden und der Eingang <i>PVPER_ON</i> gesetzt werden.</li> <li>Default: FALSE</li> </ul>
I_ITL_ON	INPUT	BOOL	<p>INITIALIZATION OF THE INTEGRAL ACTION ON / I-Anteil setzen</p> <ul style="list-style-type: none"> <li>Der Ausgang des Integrierers kann auf den Eingang <i>I_ITLVAL</i> gesetzt werden. Hierzu muss der Eingang <i>I_ITL_ON</i> gesetzt werden.</li> <li>Default: FALSE</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
PULSE_ON	INPUT	BOOL	PULSE GENERATOR ON / Impulsformer einschalten <ul style="list-style-type: none"> <li>■ Mit <i>PULSE_ON</i> = TRUE wird der Impulsformer eingeschaltet.</li> <li>■ Default: FALSE</li> </ul>
TUN_KEEP	INPUT	BOOL	KEEP TUNING ON / Optimierbetrieb halten <ul style="list-style-type: none"> <li>■ Ein Übergang in den Automatikbetrieb erfolgt erst, wenn <i>TUN_KEEP</i> = FALSE.</li> <li>■ Default: FALSE</li> </ul>
ER	OUTPUT	REAL	ERROR SIGNAL / Regeldifferenz <ul style="list-style-type: none"> <li>■ Am Ausgang <i>ER</i> wird die effektiv wirkende Regeldifferenz ausgegeben.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: Abhängig von den eingesetzten Sensoren</li> </ul>
LMN_P	OUTPUT	REAL	PROPORTIONALITY COMPONENT / P-Anteil <ul style="list-style-type: none"> <li>■ Der Ausgang <i>LMN_P</i> beinhaltet den Proportionalanteil der Stellgröße.</li> <li>■ Default: 0.0</li> </ul>
LMN_I	OUTPUT	REAL	INTEGRAL COMPONENT / I-Anteil <ul style="list-style-type: none"> <li>■ Der Ausgang <i>LMN_I</i> beinhaltet den Integralanteil der Stellgröße.</li> <li>■ Default: 0.0</li> </ul>
LMN_D	OUTPUT	REAL	DERIVATIVE COMPONENT / D-Anteil <ul style="list-style-type: none"> <li>■ Der Ausgang <i>LMN_D</i> beinhaltet den Differentialanteil der Stellgröße.</li> <li>■ Default: 0.0</li> </ul>
PHASE	OUTPUT	INT	PHASE OF SELF TUNING / Phasenanzeige der Regleroptimierung <ul style="list-style-type: none"> <li>■ Am Ausgang <i>PHASE</i> wird die aktuelle Ablaufphase der Regleroptimierung angezeigt (0...7).</li> <li>■ Default: 0</li> <li>■ Wertebereich: 0, 1, 2, 3, 4, 5, 7</li> </ul>
STATUS_H	OUTPUT	INT	STATUS HEATING OF SELF TUNING / Status Heizen der Regleroptimierung <ul style="list-style-type: none"> <li>■ <i>STATUS_H</i> zeigt einen Diagnosewert über die Suche des Wendepunktes beim Heizvorgang an.</li> <li>■ Default: 0</li> </ul>
STATUS_D	OUTPUT	INT	STATUS CONTROLLER DESIGN OF SELF TUNING / Status Reglerentwurf der Regleroptimierung <ul style="list-style-type: none"> <li>■ <i>STATUS_D</i> zeigt einen Diagnosewert über den Reglerentwurf beim Heizvorgang an.</li> <li>■ Default: 0</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
QTUN_RUN	OUTPUT	BOOL	TUNING IS ACTIVE (PHASE 2) / Optimierung läuft (Phase 2) <ul style="list-style-type: none"> <li>Die Optimierung wurde durch Aufschalten der Optimierungsstellgröße begonnen und befindet sich noch in Phase 2 (Wendepunktsuche).</li> <li>Default: 0</li> </ul>
PI_CON	OUTPUT	STRUCT	PI CONTROLLER PARAMETERS / PI Reglerparameter
GAIN	OUTPUT	REAL	PI PROPORTIONAL GAIN / PI Reglerverstärkung <ul style="list-style-type: none"> <li>Default: 0.0</li> <li>Wertebereich: % / phys. Einheit</li> </ul>
TI	OUTPUT	REAL	PI RESET TIME [s] / PI Integrationszeit [s] <ul style="list-style-type: none"> <li>Default: 0.0 s</li> <li>Wertebereich: <math>\geq 0.0</math> s</li> </ul>
PID_CON	OUTPUT	STRUCT	PID CONTROLLER PARAMETERS / PID Reglerparameter
GAIN	OUTPUT	REAL	PID PROPORTIONAL GAIN / PID Reglerverstärkung <ul style="list-style-type: none"> <li>Default: 0.0</li> </ul>
TI	OUTPUT	REAL	PID RESET TIME [s] / PID Integrationszeit [s] <ul style="list-style-type: none"> <li>Default: 0.0 s</li> <li>Wertebereich: <math>\geq 0.0</math> s</li> </ul>
TD	OUTPUT	REAL	PID DERIVATIVE TIME [s] / PID Differenzierzeit [s] <ul style="list-style-type: none"> <li>Default: 0.0 s</li> <li>Wertebereich: <math>\geq 0.0</math> s</li> </ul>
PAR_SAVE	OUTPUT	STRUCT	SAVED CONTROLLER PARAMETERS / Gespeicherte PID Reglerparameter <ul style="list-style-type: none"> <li>In dieser Struktur werden die PID-Parameter gespeichert.</li> </ul>
PFAC_SP	INPUT/ OUTPUT	REAL	PROPORTIONAL FACTOR FOR SETPOINT CHANGES / Proportionalfaktor bei Sollwertänderungen <ul style="list-style-type: none"> <li>Default: 1.0</li> <li>Wertebereich: 0.0 ... 1.0</li> </ul>
GAIN	OUTPUT	REAL	PROPORTIONAL GAIN / Reglerverstärkung <ul style="list-style-type: none"> <li>Default: 0.0</li> <li>Wertebereich: % / phys. Einheit</li> </ul>
TI	INPUT/ OUTPUT	REAL	RESET TIME [s] / Integrationszeit [s] <ul style="list-style-type: none"> <li>Default: 40.0 s</li> <li>Wertebereich: <math>\geq 0.0</math> s</li> </ul>
TD	INPUT/ OUTPUT	REAL	DERIVATIVE TIME [s] / Differenzierzeit [s] <ul style="list-style-type: none"> <li>Default: 10.0 s</li> <li>Wertebereich: <math>\geq 0.0</math> s</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
D_F	OUTPUT	REAL	DERIVATIVE FACTOR / Differenzierfaktor <ul style="list-style-type: none"> <li>■ Default: 5.0</li> <li>■ Wertebereich: 5.0 ... 10.0</li> </ul>
CON_ZONE	OUTPUT	REAL	CONTROL ZONE ON / Regelzone einschalten <ul style="list-style-type: none"> <li>■ Default: 100.0</li> <li>■ Wertebereich: <math>\geq 0.0</math></li> </ul>
CONZ_ON	OUTPUT	REAL	CONTROL ZONE / Regelzonenbreite <ul style="list-style-type: none"> <li>■ Default: FALSE</li> </ul>
PFAC_SP	INPUT/ OUTPUT	REAL	PROPORTIONAL FACTOR FOR SETPOINT CHANGES / Proportionalfaktor bei Sollwertänderungen <ul style="list-style-type: none"> <li>■ <i>PFAC_SP</i> gibt den wirksamen P-Anteil bei Sollwertänderung an. Er wird zwischen 0 und 1 eingestellt. <ul style="list-style-type: none"> <li>– 1: P-Anteil ist auch bei Sollwertänderungen voll wirksam.</li> <li>– 0: P-Anteil ist bei Sollwertänderungen nicht wirksam.</li> </ul> </li> <li>■ Default: 1.0</li> <li>■ Wertebereich: 0.0 ... 1.0</li> </ul>
GAIN	INPUT/ OUTPUT	REAL	PROPORTIONAL GAIN / Reglerverstärkung <ul style="list-style-type: none"> <li>■ Der Eingang <i>GAIN</i> gibt die Reglerverstärkung an. Eine Invertierung des Regelsinns wird durch das negative Vorzeichen von <i>GAIN</i> erreicht.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: % / phys. Einheit</li> </ul>
TI	INPUT/ OUTPUT	REAL	RESET TIME [s] / Integrationszeit [s] <ul style="list-style-type: none"> <li>■ Der Eingang <i>TI</i> (Nachstellzeit) bestimmt das Zeitverhalten des Integrierers.</li> <li>■ Default: 40.0 s</li> <li>■ Wertebereich: <math>\geq 0.0</math> s</li> </ul>
TD	INPUT/ OUTPUT	REAL	DERIVATIVE TIME [s] / Differenzierzeit [s] <ul style="list-style-type: none"> <li>■ Der Eingang <i>TD</i> (Vorhaltezeit) bestimmt das Zeitverhalten des Differenzierers.</li> <li>■ Default: 10.0 s</li> <li>■ Wertebereich: <math>\geq 0.0</math> s</li> </ul>
D_F	INPUT/ OUTPUT	REAL	DERIVATIVE FACTOR / Differenzierfaktor <ul style="list-style-type: none"> <li>■ Der Differenzierfaktor <i>D_F</i> bestimmt die Verzögerung des D-Anteils.</li> <li>■ <i>D_F</i> = Differenzierzeit / "Verzögerung des D-Anteils"</li> <li>■ Default: 5.0</li> <li>■ Wertebereich: 5.0 ... 10.0</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
CON_ZONE	INPUT/ OUTPUT	REAL	<p>CONTROL ZONE ON / Regelzonenbreite</p> <ul style="list-style-type: none"> <li>■ Ist die Regeldifferenz größer als die Regelzonenbreite <i>CON_ZONE</i>, so wird die obere Stellwertbegrenzung als Stellwert ausgegeben.</li> <li>■ Ist die Regeldifferenz kleiner als die negative Regelzonenbreite, so wird die untere Stellwertbegrenzung als Stellwert ausgegeben.</li> <li>■ Default: 100.0</li> <li>■ Abhängig von den eingesetzten Sensoren</li> </ul>
CONZ_ON	INPUT/ OUTPUT	BOOL	<p>CONTROL ZONE / Regelzone einschalten</p> <ul style="list-style-type: none"> <li>■ Mit <i>CONZ_ON</i> = TRUE können Sie die Regelzone einschalten.</li> <li>■ Default: FALSE</li> </ul>
TUN_ON	INPUT/ OUTPUT	BOOL	<p>SELF TUNING ON / Regleroptimierung einschalten</p> <ul style="list-style-type: none"> <li>■ Wenn <i>TUN_ON</i> = TRUE wird der Stellwert gemittelt bis entweder durch einen Sollwertsprung oder durch <i>TUN_ST</i> = TRUE die Stellwertanregung <i>TUN_DLMN</i> aufgeschaltet wird.</li> <li>■ Default: FALSE</li> </ul>
TUN_ST	INPUT/ OUTPUT	BOOL	<p>START SELF TUNING / Regleroptimierung starten</p> <ul style="list-style-type: none"> <li>■ Soll bei der Regleroptimierung am Arbeitspunkt der Sollwert konstant bleiben, wird durch <i>TUN_ST</i> = TRUE ein Stellwertsprung um <i>TUN_DLMN</i> aufgeschaltet.</li> <li>■ Default: FALSE</li> </ul>
UNDO_PAR	INPUT/ OUTPUT	BOOL	<p>UNDO CHANGE OF CONTROLLER PARAMETERS / Rückgängigmachen der Reglerparameteränderung</p> <ul style="list-style-type: none"> <li>■ Lädt die Reglerparameter <i>PFAC_SP</i>, <i>GAIN</i>, <i>TI</i>, <i>TD</i>, <i>D_F</i>, <i>CONZ_ON</i> und <i>CON_ZONE</i> aus der Datenstruktur <i>PAR_SAVE</i> (nur im Handbetrieb).</li> <li>■ Default: FALSE</li> </ul>
SAVE_PAR	INPUT/ OUTPUT	BOOL	<p>SAVE CURRENT CONTROLLER PARAMETERS / Aktuelle Reglerparameter sichern</p> <ul style="list-style-type: none"> <li>■ Sichert die Reglerparameter <i>PFAC_SP</i>, <i>GAIN</i>, <i>TI</i>, <i>TD</i>, <i>D_F</i>, <i>CONZ_ON</i> und <i>CON_ZONE</i> in die Datenstruktur <i>PAR_SAVE</i>.</li> <li>■ Default: FALSE</li> </ul>
LOAD_PID	INPUT/ OUTPUT	BOOL	<p>LOAD OPTIMIZED PI/PID PARAMETERS / Optimierte PID-Parameter laden</p> <ul style="list-style-type: none"> <li>■ Lädt die Reglerparameter <i>GAIN</i>, <i>TI</i>, <i>TD</i> in Abhängigkeit von <i>PID_ON</i> aus der Datenstruktur <i>PI_CON</i> bzw. <i>PID_CON</i> (nur im Handbetrieb)</li> <li>■ Default: FALSE</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
PID_ON	INPUT/ OUTPUT	BOOL	<p>PID MODE ON / PID Betriebsart einschalten</p> <ul style="list-style-type: none"> <li>Am Eingang <i>PID_ON</i> können Sie festlegen, ob der optimierte Regler als PI- oder als PID-Regler arbeiten soll. <ul style="list-style-type: none"> <li>PID-Regler: <i>PID_ON</i> = TRUE</li> <li>PI-Regler: <i>PID_ON</i> = FALSE</li> </ul> </li> <li>Es kann jedoch sein, dass bei manchen Streckentypen trotz <i>PID_ON</i> = TRUE nur ein PI-Regler entworfen wird.</li> <li>Default: TRUE</li> </ul>
GAIN_P	OUTPUT	REAL	<p>PROZESS PROPORTIONAL GAIN / Prozessverstärkung</p> <ul style="list-style-type: none"> <li>Identifizierte Prozessverstärkung. Beim Streckentyp I wird <i>GAIN_P</i> tendenziell zu klein geschätzt.</li> <li>Default: 0.0</li> </ul>
TU	OUTPUT	REAL	<p>DELAY TIME [s] / Verzugszeit [s]</p> <ul style="list-style-type: none"> <li>Identifizierte Verzugszeit vom Prozess.</li> <li>Default: 0.0</li> <li>Wertebereich: <math>\geq 3 \cdot \text{CYCLE}</math></li> </ul>
TA	OUTPUT	REAL	<p>RECOVERY TIME [s] / Ausgleichszeit [s]</p> <ul style="list-style-type: none"> <li>Identifizierte Ausgleichszeit vom Prozess. Beim Streckentyp I wird <i>TA</i> tendenziell zu klein geschätzt.</li> <li>Default: 0.0</li> </ul>
KIG	OUTPUT	REAL	<p>MAXIMAL ASCENT RATIO OF PV WITH 100 % LMN CHANGE / Maximaler Istwertanstieg bei einer Stellgrößenanregung von 0 nach 100 % [1/s]</p> <ul style="list-style-type: none"> <li><math>\text{GAIN}_P = 0.01 \cdot \text{KIG} \cdot \text{TA}</math></li> <li>Default: 0.0</li> </ul>
N_PTN	OUTPUT	REAL	<p>PROCESS ORDER / Prozessordnung</p> <ul style="list-style-type: none"> <li>Der Parameter gibt die Ordnung der Strecke an. Auch "nicht ganzzahlige Werte" sind möglich.</li> <li>Default: 0.0</li> <li>Wertebereich: 1.01 ... 10.0</li> </ul>
TM_LAG_P	OUTPUT	REAL	<p>TIME LAG OF PTN MODEL [s] / Zeitkonstante eines PTN-Modells [s]</p> <ul style="list-style-type: none"> <li>Zeitkonstante eines PTN-Modells (sinnvolle Werte nur für <math>N_{PTN} \geq 2</math>).</li> <li>Default: 0.0</li> </ul>
T_P_INF	OUTPUT	REAL	<p>TIME TO POINT OF INFLECTION [s] / Zeit bis zum Wendepunkt [s]</p> <ul style="list-style-type: none"> <li>Zeit von der Prozessanregung bis zum Wendepunkt.</li> <li>Default: 0.0</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
P_INF	OUTPUT	REAL	PV AT POINT OF INFLECTION - PV0 / Istwert am Wendepunkt-PV0 <ul style="list-style-type: none"> <li>Istwertänderung von der Prozessanregung bis zum Wendepunkt.</li> <li>Default: 0.0</li> <li>Wertebereich: Wertebereich des Istwertes</li> </ul>
LMN0	OUTPUT	REAL	MANIPULATED VAR. AT BEGIN OF TUNING / Stellwert zu Beginn der Optimierung <ul style="list-style-type: none"> <li>Wird in Phase 1 ermittelt (Mittelwert).</li> <li>Default: 0.0</li> <li>Wertebereich: 0 ... 100 %</li> </ul>
PV0	OUTPUT	REAL	PROCESS VALUE AT BEGIN OF TUNING / Istwert zu Beginn der Optimierung <ul style="list-style-type: none"> <li>Default: 0.0</li> <li>Wertebereich: Wertebereich des Istwertes</li> </ul>
PVDT0	OUTPUT	REAL	RATE OF CHANGE OF PV AT BEGIN OF TUNING [1/s] / Istwertsteigung zu Beginn der Optimierung [1/s] <ul style="list-style-type: none"> <li>Vorzeichen angepasst</li> <li>Default: 0.0</li> </ul>
PVDT	OUTPUT	REAL	CURRENT RATE OF CHANGE OF PV [1/s] / Momentane Istwertsteigung [1/s] <ul style="list-style-type: none"> <li>Vorzeichen angepasst</li> <li>Default: 0.0</li> </ul>
PVDT_MAX	OUTPUT	REAL	MAX. RATE OF CHANGE OF PV PER SECOND [1/s] / Max. Änderung des Istwertes pro Sekunde [1/s] <ul style="list-style-type: none"> <li>Maximale Ableitung des Istwertes am Wendepunkt (Vorzeichen angepasst, immer &gt; 0), wird verwendet zur Berechnung von <i>TU</i> und <i>KIG</i>.</li> <li>Default: 0.0</li> </ul>
NOI_PVDT	OUTPUT	REAL	RATIO OF NOISE IN PVDT_MAX IN % / Rauschanteil in PVDT_MAX in % <ul style="list-style-type: none"> <li>Je größer der Rauschanteil, desto ungenauer (sanfter) die Reglerparameter.</li> <li>Default: 0.0</li> </ul>
NOISE_PV	OUTPUT	REAL	ABSOLUTE NOISE IN PV / Absolutes Rauschen im Istwert <ul style="list-style-type: none"> <li>Differenz zwischen maximalem und minimalem Istwert in Phase 1.</li> <li>Default: 0.0</li> </ul>
FIL_CYC	OUTPUT	INT	NO OF CYCLES FOR MEAN-VALUE FILTER / Anzahl der Zyklen des Mittelwertfilters <ul style="list-style-type: none"> <li>Der Istwert wird über <i>FIL_CYC</i> Zyklen gemittelt. <i>FIL_CYC</i> wird bei Bedarf automatisch von 1 bis max. 1024 erhöht.</li> <li>Default: 1</li> <li>Wertebereich: 1 ... 1024</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
POI_CMAX	OUTPUT	INT	<p>MAX NO OF CYCLES AFTER POINT OF INFLECTION / Maximale Anz. Zyklen nach Wendepunkt</p> <ul style="list-style-type: none"> <li>■ Diese Zeit wird genutzt, um bei Messrauschen einen weiteren (d. h. besseren) Wendepunkt zu finden. Erst dann wird die Optimierung beendet.</li> <li>■ Default: 2</li> </ul>
POI_CYCL	OUTPUT	INT	<p>NUMBER OF CYCLES AFTER POINT OF INFLECTION / Anzahl Zyklen nach Wendepunkt</p> <ul style="list-style-type: none"> <li>■ Default: 0</li> </ul>

### Anwendung

- Die Arbeitsweise basiert auf dem PID-Regelalgorithmus, der mit zusätzlichen Funktionen für Temperaturprozesse ausgestattet ist. Der Regler liefert analoge Stellwerte und pulsbreitenmodulierte Stellsignale. Der Regler versorgt ein Stellglied, d. h. Sie können mit einem Regler entweder nur heizen oder nur kühlen.
- Den FB 58 TCONT\_CP können Sie sowohl für reine Heizstrecken als auch für reine Kühlstrecken einsetzen. Bei Einsatz für einen Kühlprozess müssen Sie *GAIN* mit einem negativen Wert parametrieren. Die so parametrierte Reglerinvertierung bewirkt, dass nun z.B. bei einem Anstieg der Temperatur sich die Stellgröße *LMN* und damit die Kühlleistung erhöht.
- Neben den Funktionen im Soll- und Istwertzweig realisiert der FB einen fertigen PID-Temperaturregler mit kontinuierlichem und binärem Stellgrößen-Ausgang. Zur Verbesserung des Regelverhaltens bei Temperaturstrecken hat der Baustein eine Regelzone und eine Reduzierung des P-Anteils bei Sollwertsprüngen. Die PI/PID-Parameter kann der Baustein mittels Regleroptimierung selbst einstellen.



Die Werte in den Regelbausteinen werden nur dann korrekt berechnet, wenn der Baustein in regelmäßigen Abständen aufgerufen wird. Deshalb sollten die Regelbausteine in einem Weckalarm-OB (OB 30 ... 38) aufgerufen werden. Die Abtastzeit wird am Parameter *CYCLE* vorgegeben.

### Sollwertzweig

Der Sollwert wird am Eingang *SP\_INT* im Gleitpunktformat physikalisch oder in Prozent eingegeben. Sollwert und Istwert müssen an der Regeldifferenzbildung die gleiche Einheit besitzen.

### Istwertauswahl (*PVPER\_ON*)

Der Istwert kann abhängig von *PVPER\_ON* im Peripherie- oder im Gleitpunktformat eingelesen werden.

<i>PVPER_ON</i>	Istwerteingabe
TRUE	Der Istwert wird über die Analogperipherie (PEW xxx) am Eingang <i>PV_PER</i> eingelesen.
FALSE	Der Istwert wird im Gleitpunktformat am Eingang <i>PV_IN</i> eingelesen.

### Istwertformatumwandlung *CRP\_IN* (*PER\_MODE*)

Die Funktion *CRP\_IN* wandelt den Peripheriewert *PV\_PER* abhängig vom Schalter *PER\_MODE* in ein Gleitpunktformat nach folgender Vorschrift um:

PER_MODE	Ausgang von CRP_IN	Analogeingabe-Typ	Einheit
0	$PV\_PER * 0.1$	Thermoelemente; PT100/NI100; Standard	°C; °F
1	$PV\_PER * 0.01$	PT100/NI100; Klima	°C; °F
2	$PV\_PER * 100/27648$	Spannung/Strom	%

#### Istwertnormierung $PV\_NORM$ ( $PV\_FAC$ , $PV\_OFFS$ )

Die Funktion  $PV\_NORM$  berechnet den Ausgang von  $CRP\_IN$  nach folgender Vorschrift:

$$\text{Ausgang von } PV\_NORM = \text{Ausgang von } CRP\_IN * PV\_FAC + PV\_OFFS$$

Sie kann für folgende Zwecke eingesetzt werden:

- Istwert-Anpassung mit  $PV\_FAC$  als Istwertfaktor und  $PV\_OFFS$  als Istwertoffset
- Normierung von Temperatur nach Prozent  
Sie wollen den Sollwert in Prozent eingeben und müssen nun den gemessenen Temperaturwert in Prozent umrechnen.
- Normierung von Prozent nach Temperatur  
Sie wollen den Sollwert in der physikalischen Größe Temperatur eingeben und müssen nun den gemessenen Spannungs/Strom-Wert in eine Temperatur umrechnen.

Berechnung der Parameter:

- $PV\_FAC = \text{Bereich von } PV\_NORM / \text{Bereich von } CRP\_IN$
- $PV\_OFFS = UG(PV\_NORM) - PV\_FAC * UG(CRP\_IN)$ ; mit  $UG$ : Untergrenze

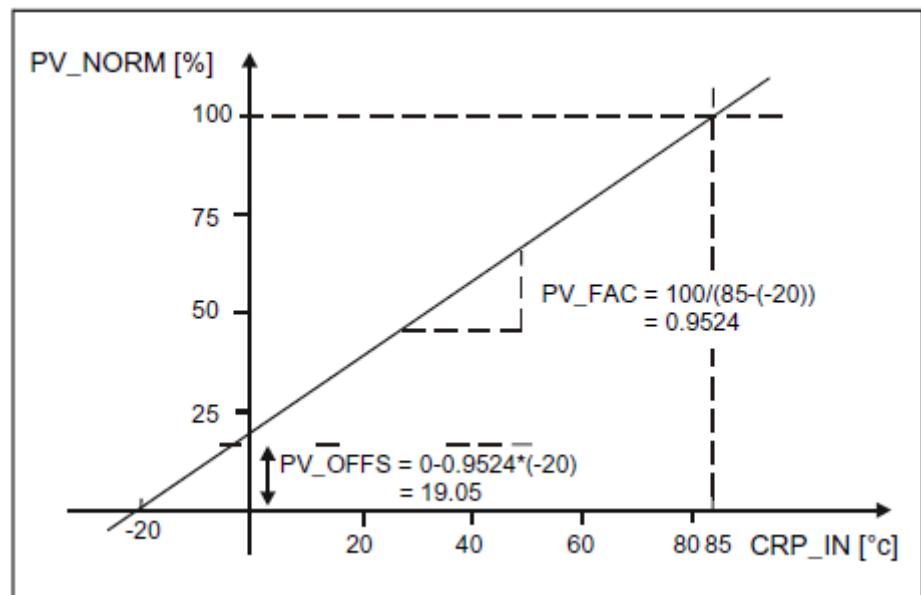
Mit den Defaultwerten ( $PV\_FAC = 1.0$  und  $PV\_OFFS = 0.0$ ) ist die Normierung abgeschaltet. Der effektiv wirksame Istwert wird am Ausgang  $PV$  ausgegeben.



Bei Impulsregelung muss der Istwert im schnellen Impulsaufzug dem Baustein übergeben werden (Grund: Mittelwertfilterung). Sonst kann sich die Regelqualität verschlechtern.

#### Beispiel zur Istwertnormierung

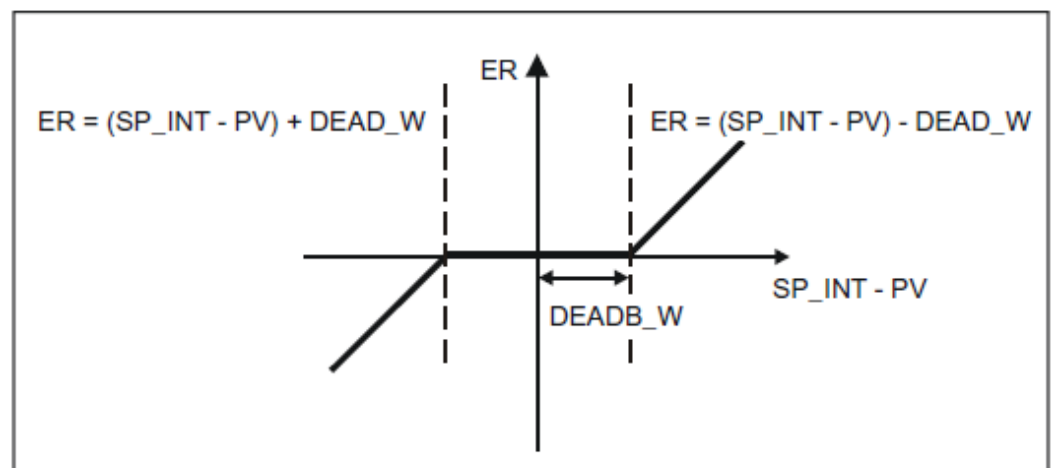
Wenn Sie den Sollwert in Prozent vorgeben wollen und Sie einen Temperaturbereich von -20 ... 85 °C an  $CRP\_IN$  anliegen haben, müssen Sie den Temperaturbereich in Prozent umnormieren. Das folgende Bild stellt ein Beispiel für die Anpassung des Temperaturbereichs -20 ... 85 °C auf intern 0 ... 100 % dar:

**Regeldifferenzbildung**

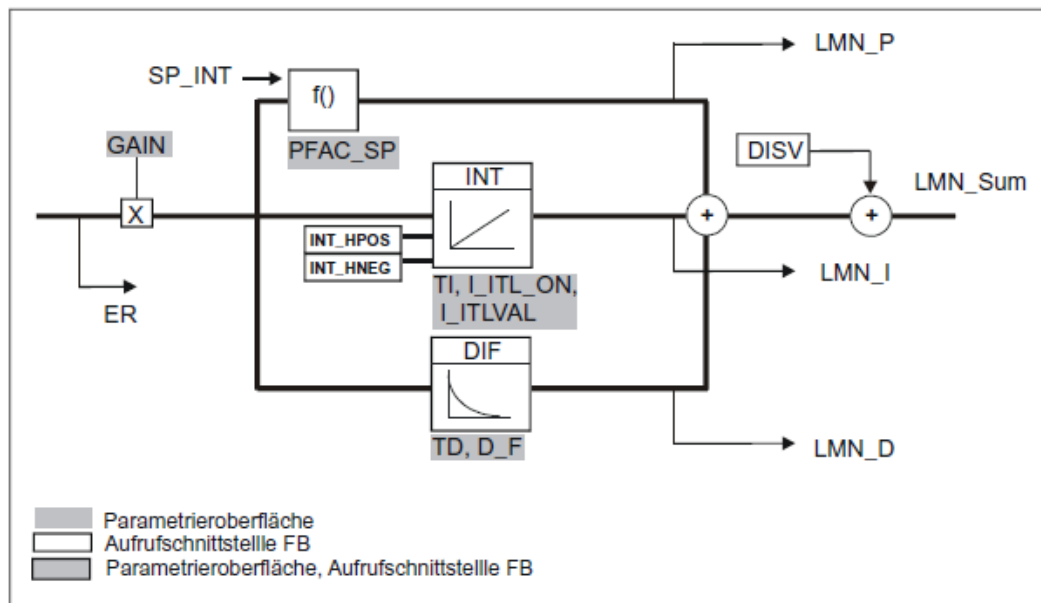
Die Differenz von Soll- und Istwert ergibt die Regeldifferenz vor der Totzone. Soll- und Istwert müssen in der gleichen Einheit vorliegen.

**Totzone (DEADB\_W)**

Zur Unterdrückung einer kleinen Dauerschwingung aufgrund der Stellgrößen-Quantisierung (z. B. bei einer Pulsweitenmodulation mit PULSEGEN) wird die Regeldifferenz über eine Totzone (DEADBAND) geleitet. Bei  $DEADB\_W = 0.0$  ist die Totzone ausgeschaltet. Die wirksame Regeldifferenz wird am Parameter *ER* angezeigt.

**PID-Algorithmus**

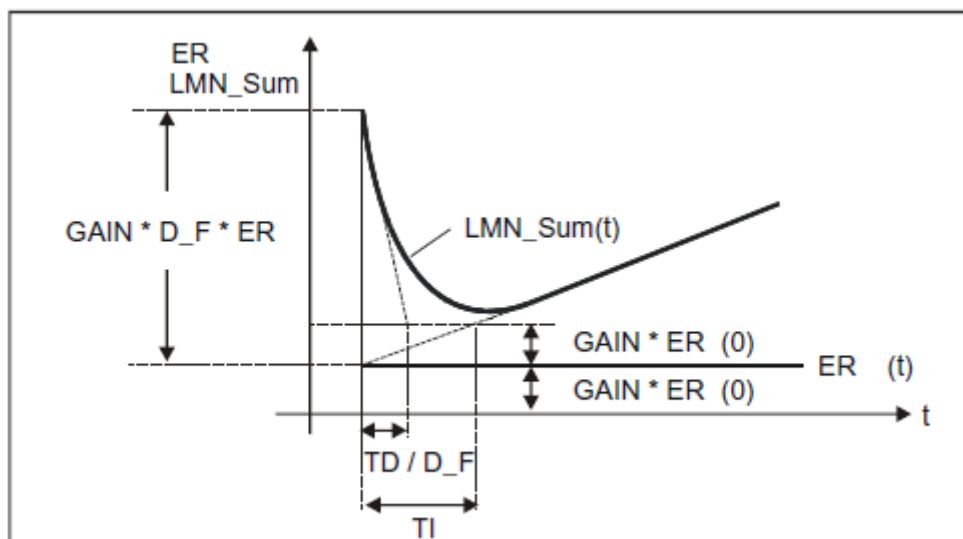
In folgendem Bild ist das Blockschaltbild des PID-Algorithmus dargestellt:



### PID-Algorithmus (*GAIN*, *TI*, *TD*, *D\_F*)

- Der PID-Algorithmus arbeitet im Stellungsalgorithmus. Der Proportional-, Integral (*INT*) und Differentialanteil (*DIF*) sind parallel geschaltet und lassen sich einzeln zu- und abschalten. Damit können Sie P-, PI-, PD- und PID-Regler parametrieren.
- Die Regleroptimierung unterstützt PI- und PID-Regler. Reglerinvertierung erfolgt über ein negatives *GAIN* (Kühlregler).
- Wenn Sie *TI* und *TD* auf 0.0 setzen erhalten Sie einen reinen P-Regler am Arbeitspunkt.

$$LMN\_Sum(t) = GAIN * ER(0) \left( 1 + \frac{I}{TI} * t + D\_F * e^{-\frac{t}{TD/D\_F}} \right)$$



LMN_Sum(t)	Stellgröße bei Automatikbetrieb des Reglers
ER (0)	Sprunghöhe der normierten Regeldifferenz
GAIN	Reglerverstärkung
TI	Integrationszeit
TD	Differenzierzeit
D_F	Differenzierfaktor

**Integrierer (TI, I\_ITL\_ON, I\_ITLVAL)**

Bei Handbetrieb wird er wie folgt nachgeführt:  $LMN\_I = LMN - LMN\_P - DISV$

Bei Begrenzung des Stellwerts wird der I-Anteil angehalten. Bei einer Regeldifferenz, die den I-Anteil in Richtung innerer Stellbereich bewegt, wird der I-Anteil wieder freigegeben.

Weitere Modifikationen des I-Anteils ergeben sich durch folgende Maßnahmen:

- Abschalten des I-Anteils des Reglers erfolgt mit  $TI = 0.0$
- Abschwächung des P-Anteils bei Sollwertänderungen
- Regelzone
- Die Stellwertgrenzen lassen sich online ändern

**Abschwächung des P-Anteils bei Sollwertänderungen (PFAC\_SP)**

Um Überspringen zu vermeiden können Sie den P-Anteil über den Parameter "Proportionalfaktor bei Sollwertänderungen" ( $PFAC\_SP$ ) abschwächen. Über  $PFAC\_SP$  können Sie zwischen 0.0 und 1.0 kontinuierlich wählen, wie stark der P-Anteil bei Sollwertänderungen wirken soll:

- $PFAC\_SP = 1.0$ : P-Anteil bei Sollwertänderung voll wirksam
- $PFAC\_SP = 0.0$ : P-Anteil bei Sollwertänderung nicht wirksam

Die Abschwächung des P-Anteils wird durch eine Kompensation am I-Anteil erreicht.

**Differenzierer (TD, D\_F)**

- Abschalten des D-Anteils des Reglers erfolgt mit  $TD = 0.0$ .
- Bei zugeschaltetem D-Anteil sollte folgende Gleichung eingehalten werden:  
 $TD = 0.5 * CYCLE * D\_F$

**Parametrierung eines P- oder PD-Reglers mit Arbeitspunkt**

Schalten Sie in der Parametrieroberfläche den I-Anteil ( $TI = 0.0$ ) und evtl. den D-Anteil ( $TD = 0.0$ ) ab. Führen Sie außerdem folgende Parametrierung durch:

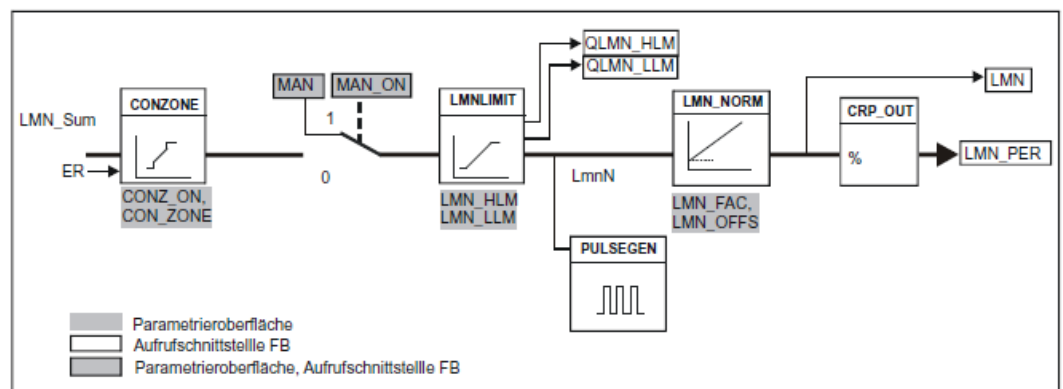
- $I\_ITL\_ON = TRUE$
- $I\_ITLVAL = \text{Arbeitspunkt}$

**Störgrößenaufschaltung (DISV)**

Am Eingang  $DISV$  können Sie eine Störgröße additiv aufschalten.

**Stellwertberechnung**

In folgendem Bild ist das Blockschaltbild der Stellwertberechnung dargestellt:



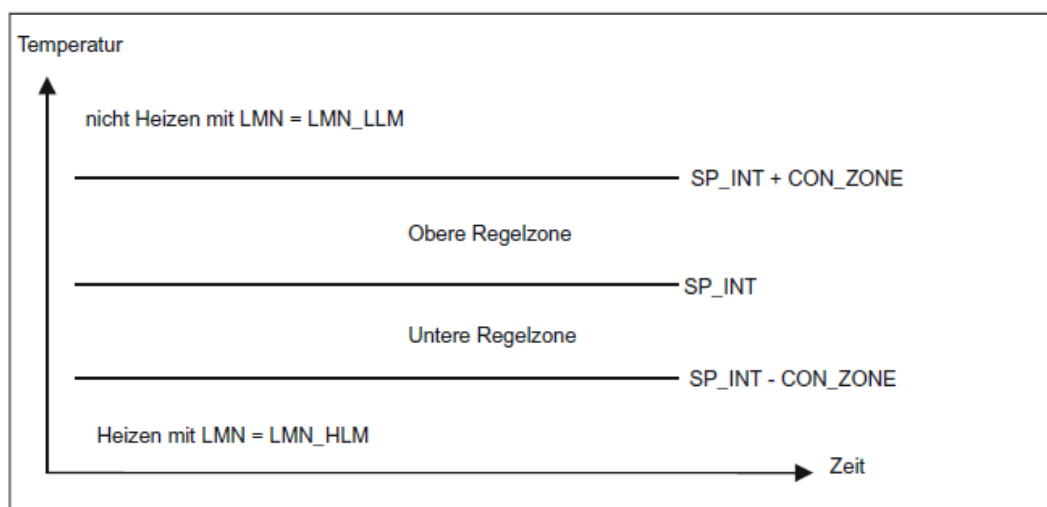
**Regelzone (CONZ\_ON, CON\_ZONE)**

Wenn **CONZ\_ON** = TRUE arbeitet der Regler mit einer Regelzone. Das bedeutet, dass der Regler nach folgendem Algorithmus angesteuert wird:

- Übersteigt der Istwert **PV** den Sollwert **SP\_INT** um mehr als **CON\_ZONE**, wird als Stellgröße der Wert **LMN\_LLM** ausgegeben (gesteuerter Regelbetrieb).
- Unterschreitet der Istwert **PV** den Sollwert **SP\_INT** um mehr als **CON\_ZONE**, wird **LMN\_HLM** ausgegeben (gesteuerter Regelbetrieb).
- Bewegt sich der Istwert **PV** innerhalb der Regelzone (**CON\_ZONE**), nimmt der Stellwert den Wert vom PID-Algorithmus **LMN\_Sum** an (automatischer Regelbetrieb).



*Der Übergang vom gesteuerten Regelbetrieb zum automatischen Regelbetrieb erfolgt unter Einhaltung einer Hysterese von 20% der Regelzone.*



*Bevor Sie die Regelzone von Hand einschalten müssen Sie sicherstellen, dass die Regelzonenbreite nicht zu klein eingestellt ist. Bei zu klein eingestellter Regelzonenbreite treten Schwingungen im Verlauf der Stellgröße und des Istwertverlaufs auf.*

**Vorteil der Regelzone**

Beim Eintritt in die Regelzone führt der zugeschaltete D-Anteil zu einem sehr schnellen Reduzieren der Stellgröße. Daher ist die Regelzone nur bei eingeschaltetem D-Anteil sinnvoll. Ohne Regelzone würde im wesentlichen nur der sich reduzierende P-Anteil die Stellgröße reduzieren. Die Regelzone führt zu einem schnelleren Einschwingen ohne Über-/Unterschwingen, wenn die ausgegebene minimale oder maximale Stellgröße weit von der für den neuen Arbeitspunkt stationär notwendigen Stellgröße entfernt ist.

**Handwertverarbeitung (MAN\_ON, MAN)**

Sie können zwischen Hand- und Automatikbetrieb umschalten. Bei Handbetrieb wird die Stellgröße einem Handwert nachgeführt. Der Integrierer (**INT**) wird intern auf **LMN - LMN\_P - DISV** und der Differenzierer (**DIF**) auf 0 gesetzt und intern abgeglichen. Das Umschalten in den Automatikbetrieb ist damit stoßfrei.



*Während einer Optimierung ist der Parameter **MAN\_ON** nicht wirksam.*

**Stellwertbegrenzung  
*LMNLIMIT* (*LMN\_HLM*,  
*LMN\_LLM*)**

Der Stellwert wird mit der Funktion *LMNLIMIT* auf die Stellwertgrenzen *LMN\_HLM* und *LMN\_LLM* begrenzt. Das Erreichen der Grenzen wird durch die Meldebits *QLMN\_HLM* und *QLMN\_LLM* angezeigt. Bei Begrenzung des Stellwerts wird der I-Anteil angehalten. Bei einer Regeldifferenz, die den I-Anteil in Richtung innerer Stellbereich bewegt, wird der I-Anteil wieder freigegeben.

**Stellwertgrenzen online  
ändern**

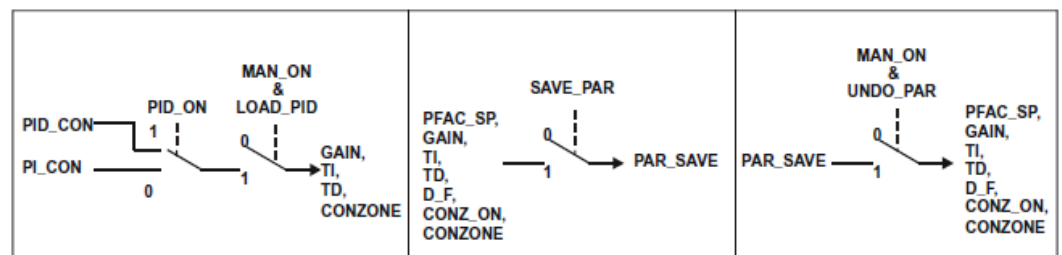
Wird der Stellwertbereich verringert und der neue unbegrenzte Stellwert liegt außerhalb der Grenzen, wird der I-Anteil und damit der Stellwert verschoben. Der Stellwert wird um die Änderung der Stellwertgrenze verringert. War der Stellwert vor der Änderung unbegrenzt, wird er genau auf die neue Grenze gesetzt (hier beschrieben für die obere Stellwertgrenze).

**Stellwertnormierung  
*LMN\_NORM* (*LMN\_FAC*,  
*LMN\_OFFS*)**

- Die Funktion *LMN\_NORM* normiert den Stellwert nach folgender Vorschrift:  
 $LMN = LmnN * LMN\_FAC + LMN\_OFFS$
- Sie kann für folgende Zwecke eingesetzt werden:  
Stellwert-Anpassung mit *LMN\_FAC* als Stellwertfaktor und *LMN\_OFFS* als Stellwertoffset
- Der Stellwert steht auch im Peripherieformat zur Verfügung. Die Funktion *CRP\_OUT* wandelt den Gleitpunktwert *LMN* in einen Peripheriewert nach folgender Vorschrift um:  
 $LMN\_PER = LMN * 27648/100$   
Mit den Defaultwerten (*LMN\_FAC* = 1.0 und *LMN\_OFFS* = 0.0) ist die Normierung abgeschaltet. Der effektiv wirksame Stellwert wird am Ausgang *LMN* ausgegeben.

**Reglerparameter spei-  
chern und zurückladen**

In folgendem Bild ist das Blockschaltbild dargestellt:

**Reglerparameter spei-  
chern *SAVE\_PAR***

Wenn Sie die aktuellen Reglerparameter als brauchbar einstufen, können Sie diese vor einer manuellen Änderung in eigens dafür vorgesehenen Strukturparametern im Instanz-DB des FB 58 speichern. Bei einer Regleroptimierung werden die gespeicherten Parameter durch die vor der Optimierung gültigen Werte überschrieben. *PFAC\_SP*, *GAIN*, *TI*, *TD*, *D\_F*, *CONZ\_ON* und *CON\_ZONE* werden in die Struktur *PAR\_SAVE* geschrieben.

**Gespeicherte Reglerpara-  
meter zurückladen  
*UNDO\_PAR***

Die zuletzt gespeicherten Reglerparameter können mit dieser Funktion wieder für den Regler aktiviert werden (nur im Handbetrieb).

**Wechsel zwischen PI- und  
PID-Parametern  
*LOAD\_PID* (*PID\_ON*)**

Nach einer Optimierung werden die PI- und PID-Parameter in den Strukturen *PI\_CON* und *PID\_CON* hinterlegt. Mit *LOAD\_PID* in Abhängigkeit von *PID\_ON* können Sie im Handbetrieb die PI- bzw. PID-Parameter auf die wirksamen Reglerparameter schreiben.

PID-Parameter <i>PID_ON</i> = TRUE		PI-Parameter <i>PID_ON</i> = FALSE	
GAIN	= PID_CON.GAIN	GAIN	= PI_CON.GAIN
TI	= PID_CON.TI	TI	= PI_CON.TI
TD	= PID_CON.TD		



- Die Reglerparameter werden mit *UNDO\_PAR* oder *LOAD\_PID* nur dann zurückgeschrieben, wenn die Reglerverstärkung ungleich Null ist:  
Bei *LOAD\_PID* werden die Parameter nur kopiert, falls das jeweiligen *GAIN*  $\neq 0$  ist (entweder vom PI- oder PID-Parametersatz). Damit ist der Fall berücksichtigt, dass noch keine Optimierung durchgeführt wurde bzw. PID-Parameter fehlen. War *PID\_ON* = TRUE und *PID.GAIN* = FALSE, wird *PID\_ON* auf FALSE gesetzt und die PI-Parameter kopiert.
- *D\_F*, *PFAC\_SP* werden durch die Optimierung voreingestellt. Sie können anschließend vom Anwender modifiziert werden. *LOAD\_PID* verändert diese Parameter nicht.
- Die Regelzone wird bei *LOAD\_PID* immer neu berechnet (*CON\_ZONE* = 250/*GAIN*), auch wenn *CONZ\_ON* = FALSE.

#### 4.5.5 FB 59 - TCONT\_S - Temperatur-Schrittregeln

##### Beschreibung

Der FB 59 TCONT\_S dient zum Regeln von technischen Temperatur-Prozessen mit binären Stellwertausgangssignalen für integrierende Stellglieder. Über die Parametrierung lassen sich Teilfunktionen des PI-Schrittreglers zu- oder abschalten und damit an die Regelstrecke anpassen.

##### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
CYCLE	INPUT	REAL	SAMPLE TIME OF STEP CONTROLLER [s] / Abtastzeit des Schrittreglers [s] <ul style="list-style-type: none"> <li>■ An diesem Eingang <i>CYCLE</i> geben Sie die Abtastzeit für den Regler ein.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: <math>\geq 0.001</math></li> </ul>
SP_INT	INPUT	REAL	INTERNAL SETPOINT / Interner Sollwert <ul style="list-style-type: none"> <li>■ Der Eingang <i>SP_INT</i> dient zur Vorgabe eines Sollwertes.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: Abhängig von den eingesetzten Sensoren</li> </ul>
PV_IN	INPUT	REAL	PROCESS VARIABLE IN / Istwert Eingang <ul style="list-style-type: none"> <li>■ Am Eingang <i>PV_IN</i> kann ein Inbetriebsetzungswert parametrierbar oder ein externer Istwert im Gleitpunktformat verschaltet werden.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: Abhängig von den eingesetzten Sensoren</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
PV_PER	INPUT	WORD	<p>PROCESS VARIABLE PERIPHERY / Istwert Peripherie</p> <ul style="list-style-type: none"> <li>Der Istwert in Peripherieformat wird am Eingang <i>PV_PER</i> mit dem Regler verschaltet.</li> <li>Default: 0</li> </ul>
DISV	INPUT	REAL	<p>DISTURBANCE VARIABLE / Störgröße</p> <ul style="list-style-type: none"> <li>Für eine Störgrößenaufschaltung wird die Störgröße am Eingang <i>DISV</i> verschaltet.</li> <li>Default: 0.0</li> </ul>
LMNR_HS	INPUT	BOOL	<p>HIGH LIMIT SIGNAL OF REPEATED MANIPULATED VALUE / Oberes Anschlagssignal der Stellungsrückmeldung</p> <ul style="list-style-type: none"> <li>Das Signal "Stellventil am oberen Anschlag" wird am Eingang <i>LMNR_HS</i> verschaltet.</li> <li><i>LMNR_HS</i> = TRUE: Das Stellventil befindet sich am oberen Anschlag.</li> <li>Default: FALSE</li> </ul>
LMNR_LS	INPUT	BOOL	<p>LOW LIMIT SIGNAL OF REPEATED MANIPULATED VALUE / Unteres Anschlagssignal der Stellungsrückmeldung</p> <ul style="list-style-type: none"> <li>Das Signal "Stellventil am unteren Anschlag" wird am Eingang <i>LMNR_LS</i> verschaltet.</li> <li><i>LMNR_LS</i> = TRUE: Das Stellventil befindet sich am unteren Anschlag.</li> <li>Default: FALSE</li> </ul>
LMNS_ON	INPUT	BOOL	<p>MANIPULATED SIGNALS ON / Handbetrieb der Stellwertsignale einschalten</p> <ul style="list-style-type: none"> <li>Am Eingang <i>LMNS_ON</i> wird die Stellwertsignalverarbeitung auf Hand geschaltet.</li> <li>Default: TRUE</li> </ul>
LMNUP	INPUT	BOOL	<p>MANIPULATED SIGNALS UP / Stellwertsignal Hoch</p> <ul style="list-style-type: none"> <li>Bei Handbetrieb der Stellwertsignale wird am Eingang <i>LMNUP</i> das Ausgangssignal <i>QLMNUP</i> bedient.</li> <li>Default: FALSE</li> </ul>
LMNDN	INPUT	BOOL	<p>MANIPULATED SIGNALS DOWN / Stellwertsignal Tief</p> <ul style="list-style-type: none"> <li>Bei Handbetrieb der Stellwertsignale wird am Eingang <i>LMNDN</i> das Ausgangssignal <i>QLMNDN</i> bedient.</li> <li>Default: FALSE</li> </ul>
QLMNUP	OUTPUT	BOOL	<p>MANIPULATED SIGNAL UP / Stellwertsignal Hoch</p> <ul style="list-style-type: none"> <li>Ist der Ausgang <i>QLMNUP</i> gesetzt, soll das Stellventil geöffnet werden.</li> <li>Default: FALSE</li> </ul>
QLMNDN	OUTPUT	BOOL	<p>MANIPULATED SIGNAL DOWN / Stellwertsignal Tief</p> <ul style="list-style-type: none"> <li>Ist der Ausgang <i>QLMNDN</i> gesetzt, soll das Stellventil geschlossen werden.</li> <li>Default: FALSE</li> </ul>

Parameter	Deklaration	Datentyp	Beschreibung
PV	OUTPUT	REAL	PROCESS VARIABLE / Istwert <ul style="list-style-type: none"> <li>■ Am Ausgang <i>PV</i> wird der effektiv wirkende Istwert ausgegeben.</li> <li>■ Default: 0.0</li> </ul>
PE	OUTPUT	REAL	ERROR SIGNAL / Regeldifferenz <ul style="list-style-type: none"> <li>■ Am Ausgang <i>PE</i> wird die effektiv wirkende Regeldifferenz ausgegeben.</li> <li>■ Default: 0.0</li> </ul>
COM_RST	INPUT/ OUTPUT	BOOL	COMPLETE RESTART / Neustart <ul style="list-style-type: none"> <li>■ Der Baustein hat eine Initialisierungsroutine, die bearbeitet wird, wenn der Eingang <i>COM_RST</i> gesetzt ist.</li> <li>■ Default: FALSE</li> </ul>

## Interne Parameter

Parameter	Deklaration	Datentyp	Beschreibung
PV_FAC	INPUT	REAL	PROCESS VARIABLE FACTOR / Istwertfaktor <ul style="list-style-type: none"> <li>■ Der Eingang <i>PV_FAC</i> wird mit dem Istwert multipliziert. Der Eingang dient zur Anpassung des Istwertbereiches.</li> <li>■ Default: 1.0</li> </ul>
PV_OFFS	INPUT	REAL	PROCESS VARIABLE OFFSET / Istwertoffset <ul style="list-style-type: none"> <li>■ Der Eingang <i>PV_OFFS</i> wird mit dem Istwert addiert. Der Eingang dient zur Anpassung des Istwertbereiches.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: Abhängig von den eingesetzten Sensoren</li> </ul>
DEADB_W	INPUT	REAL	DEAD BAND WIDTH / Totzonenbreite <ul style="list-style-type: none"> <li>■ Die Regeldifferenz wird über eine Totzone geführt. Der Eingang <i>DEADB_W</i> bestimmt die Größe der Totzone.</li> <li>■ Default: 0.0</li> <li>■ Wertebereich: Abhängig von den eingesetzten Sensoren</li> </ul>
PFAC_SP	INPUT	REAL	PROPORTIONAL FACTOR FOR SETPOINT CHANGES [0..1] / Proportionalfaktor bei Sollwertänderungen <ul style="list-style-type: none"> <li>■ <i>PFAC_SP</i> gibt den wirksamen P-Anteil bei Sollwertänderung an. Er wird zwischen 0 und 1 eingestellt. <ul style="list-style-type: none"> <li>– 1: P-Anteil ist bei Sollwertänderungen voll wirksam</li> <li>– 0: P-Anteil ist bei Sollwertänderungen nicht wirksam</li> </ul> </li> <li>■ Default: 1.0</li> <li>■ Wertebereich: 0.0 ... 1.0</li> </ul>

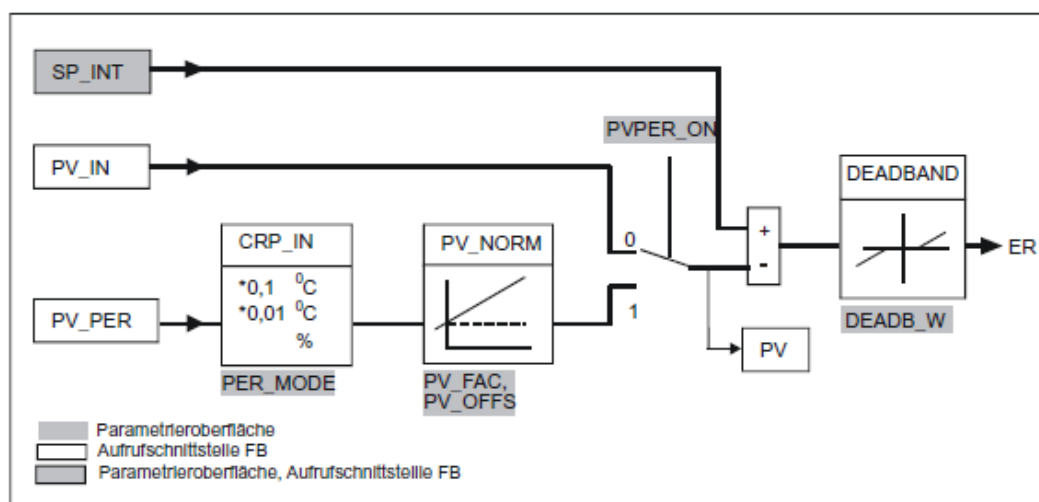
Parameter	Deklaration	Datentyp	Beschreibung
GAIN	INPUT	REAL	PROPORTIONAL GAIN / Reglerverstärkung <ul style="list-style-type: none"> <li>Der Eingang <i>GAIN</i> gibt die Reglerverstärkung an. Eine Invertierung des Regelsinns wird durch das negative Vorzeichen von <i>GAIN</i> erreicht.</li> <li>Default: 2.0</li> <li>Wertebereich: %/phys. Einheit</li> </ul>
TI	INPUT	REAL	RESET TIME [s] / Integrationszeit [s] <ul style="list-style-type: none"> <li>Der Eingang <i>TI</i> (Nachstellzeit) bestimmt das Zeitverhalten des Integrierers.</li> <li>Default: 40.0 s</li> <li>Wertebereich: <math>\geq 0.0</math> s</li> </ul>
MTR_TM	INPUT	REAL	MOTOR ACTUATING TIME / Motorstellzeit [s] <ul style="list-style-type: none"> <li>Am Parameter <i>MTR_TM</i> wird die Laufzeit des Stellventils vom Anschlag zu Anschlag eingetragen.</li> <li>Default: 30 s</li> <li>Wertebereich: <math>\geq CYCLE</math></li> </ul>
PULSE_TM	INPUT	REAL	MINIMUM PULSE TIME / Mindestimpulsdauer [s] <ul style="list-style-type: none"> <li>Am Parameter <i>PULSE_TM</i> kann eine minimale Impulslänge parametrisiert werden.</li> <li>Default: 0.1s</li> <li>Wertebereich: <math>\geq 0.0</math> s</li> </ul>
BREAK_TM	INPUT	REAL	MINIMUM BREAK TIME / Mindestpausendauer [s] <ul style="list-style-type: none"> <li>Am Parameter <i>BREAK_TM</i> kann eine minimale Pausenlänge parametrisiert werden.</li> <li>0.1s</li> <li>Wertebereich: <math>\geq 0.0</math> s</li> </ul>
PER_MODE	INPUT	INT	PERIPHERIE MODE / Peripherie Betriebsart <ul style="list-style-type: none"> <li>An diesem Schalter können Sie den Typ der AE-Baugruppe eingeben. Der Istwert am Eingang <i>PV_PER</i> wird dadurch am Ausgang <i>PV</i> in °C normiert.               <ul style="list-style-type: none"> <li><i>PER_MODE</i> = 0: Standard</li> <li><i>PER_MODE</i> = 1: Klima</li> <li><i>PER_MODE</i> = 2: Strom/Spannung</li> </ul> </li> <li>Default: 0</li> <li>Wertebereich: 0, 1, 2</li> </ul>
PVPER_ON	INPUT	BOOL	PROCESS VARIABLE PERIPHERY ON / Istwert Peripherie einschalten <ul style="list-style-type: none"> <li>Soll der Istwert von der Peripherie eingelesen werden, so muss der Eingang <i>PV_PER</i> mit der Peripherie verschaltet werden und der Eingang <i>PVPER_ON</i> gesetzt werden.</li> <li>Default: FALSE</li> </ul>

**Anwendung**

- Die Arbeitsweise basiert auf dem PI-Regelalgorithmus des Abtastreglers. Dieser wird um die Funktionsglieder zur Erzeugung des binären Ausgangssignals aus dem analogen Stellsignal ergänzt.
- Sie können den Regler auch in einer Reglerkaskade als unterlagerten Stellungsregler einsetzen. Über den Sollwerteingang *SP\_INT* geben Sie die Stellgliedposition vor. In diesem Fall müssen Sie den Istwerteingang und den Parameter *TI* (Integrationszeit) auf Null setzen. Anwendungsfall ist z. B. eine Temperaturregelung mit Heizleistungsregelung über Puls-Pause-Ansteuerung und Kühlleistungsregelung über eine Ventilklappe. Um die Klappe ganz zu schließen, sollte die Stellgröße (*ER \* GAIN*) negativ werden.
- Neben den Funktionen im Istwertzweig realisiert der FB 59 TCONT\_S einen fertigen PI-Regler mit binärem Stellwertausgang und Beeinflussungsmöglichkeit der Stellwert-signale von Hand. Der Schrittregler arbeitet ohne Stellungsrückmeldung.



Die Werte in den Regelbausteinen werden nur dann korrekt berechnet, wenn der Baustein in regelmäßigen Abständen aufgerufen wird. Deshalb sollten die Regelbausteine in einem Weckalarm-OB (OB 30 ... 38) aufgerufen werden. Die Abtastzeit wird am Parameter *CYCLE* vorgegeben.

**Regeldifferenzbildung****Blockschaltbild****Sollwertzweig**

Der Sollwert wird am Eingang *SP\_INT* im Gleitpunktformat physikalisch oder in Prozent eingegeben. Sollwert und Istwert müssen an der Regeldifferenzbildung die gleiche Einheit besitzen.

**Istwertauswahl  
(*PVPER\_ON*)**

Der Istwert kann abhängig von *PVPER\_ON* im Peripherie- oder im Gleitpunktformat eingelesen werden.

<i>PVPER_ON</i>	Istwerteingabe
TRUE	Der Istwert wird über die Analogperipherie (PEW xxx) am Eingang <i>PV_PER</i> eingelesen.
FALSE	Der Istwert wird im Gleitpunktformat am Eingang <i>PV_IN</i> eingelesen.

**Istwertformatumwandlung  
CRP\_IN (PER\_MODE)**

Die Funktion *CRP\_IN* wandelt den Peripheriewert *PV\_PER* abhängig vom Schalter *PER\_MODE* in ein Gleitpunktformat nach folgender Vorschrift um:

<i>PER_MODE</i>	Ausgang von <i>CRP_IN</i>	Analogeingabe-Typ	Einheit
0	$PV\_PER * 0.1$	Thermoelemente; PT100/ NI100; Standard	°C; °F
1	$PV\_PER * 0.01$	PT100/NI100; Klima	°C; °F
2	$PV\_PER * 100/27648$	Spannung/Strom	%

**Istwertnormierung  
PV\_NORM (PF\_FAC,  
PV\_OFFS)**

Die Funktion *PV\_NORM* berechnet den Ausgang von *CRP\_IN* nach folgender Vorschrift:

$$\text{Ausgang von } PV\_NORM = \text{Ausgang von } CRP\_IN * PV\_FAC + PV\_OFFS$$

Sie kann für folgende Zwecke eingesetzt werden:

- Istwert-Anpassung mit *PV\_FAC* als Istwertfaktor und *PV\_OFFS* als Istwertoffset
- Normierung von Temperatur nach Prozent  
Sie wollen den Sollwert in Prozent eingeben und müssen nun den gemessenen Temperaturwert in Prozent umrechnen.
- Normierung von Prozent nach Temperatur  
Sie wollen den Sollwert in der physikalischen Größe Temperatur eingeben und müssen nun den gemessenen Spannungs/Strom-Wert in eine Temperatur umrechnen.

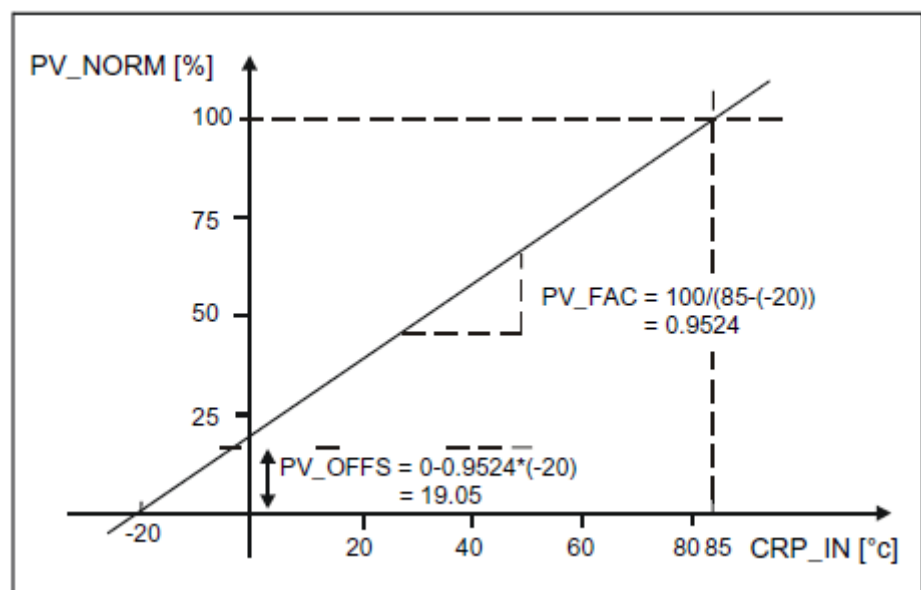
Berechnung der Parameter:

- $PV\_FAC = \text{Bereich von } PV\_NORM / \text{Bereich von } CRP\_IN$
- $PV\_OFFS = UG(PV\_NORM) - PV\_FAC * UG(CRP\_IN)$ ; mit UG: Untergrenze

Mit den Defaultwerten (*PV\_FAC* = 1.0 und *PV\_OFFS* = 0.0) ist die Normierung abgeschaltet. Der effektiv wirksame Istwert wird am Ausgang PV ausgegeben.

**Beispiel zur Istwertnormierung**

Wenn Sie den Sollwert in Prozent vorgeben wollen und Sie einen Temperaturbereich von -20 bis 85 °C an *CRP\_IN* anliegen haben, müssen Sie den Temperaturbereich in Prozent umnormieren. In folgendem Bild ist die Anpassung des Temperaturbereichs von -20 ... 85 °C auf intern 0 ... 100 % dargestellt:

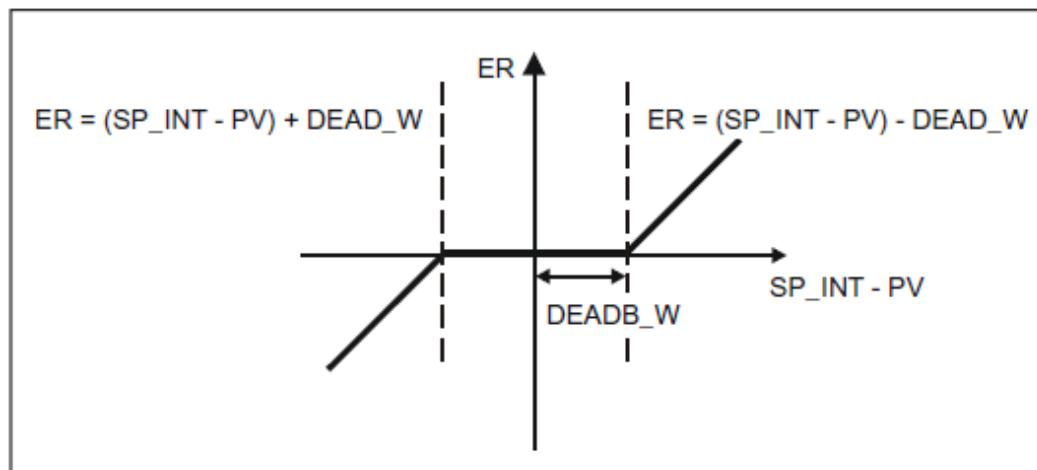


**Regeldifferenzbildung**

Die Differenz von Soll- und Istwert ergibt die Regeldifferenz vor der Totzone. Soll- und Istwert müssen in der gleichen Einheit vorliegen.

**Totzone (DEADB\_W)**

Zur Unterdrückung einer kleinen Dauerschwingung aufgrund der Stellgrößen-Quantisierung (z. B. bei einer Pulsweitenmodulation mit PULSEGEN) wird die Regeldifferenz über eine Totzone (DEADBAND) geleitet. Bei  $DEADB\_W = 0.0$  ist die Totzone ausgeschaltet. Die wirksame Regeldifferenz wird am Parameter  $ER$  angezeigt.

**PI-Schrittregler-Algorithmus**

Der FB 59 TCONT\_S arbeitet ohne Stellungsrückmeldung (siehe folgendes Blockschaltbild). Der I-Anteil des PI-Algorithmus und die gedachte Stellungsrückmeldung werden in einem Integrator (INT) berechnet und als Rückführungswert mit dem verbliebenen P-Anteil verglichen. Die Differenz geht auf ein Dreipunktglied (THREE\_ST) und einen Impulsformer (PULSEOUT), der die Impulse für das Stellventil bildet. Über eine Adaption der Ansprechschwelle des Dreipunktgliedes wird die Schalthäufigkeit des Reglers reduziert.

**Abschwächung des P-Anteils bei Sollwertänderungen**

Um Überspringen zu vermeiden können Sie den P-Anteil über den Parameter "Proportionalfaktor bei Sollwertänderungen" ( $PFAC\_SP$ ) abschwächen. Über  $PFAC\_SP$  können Sie nun zwischen 0.0 und 1.0 kontinuierlich wählen, wie stark der P-Anteil bei Sollwertänderungen wirken soll:

- $PFAC\_SP = 1.0$ : P-Anteil bei Sollwertänderung voll wirksam
- $PFAC\_SP = 0.0$ : Kein P-Anteil bei Sollwertänderung

Ein Wert von  $PFAC\_SP < 1.0$  kann wie beim kontinuierlichen Regler das Überspringen reduzieren, falls die Motorlaufzeit  $MTR\_TM$  klein gegenüber der Ausgleichszeit  $TA$  ist und das Verhältnis  $TU/TA < 0.2$  ist. Erreicht  $MTR\_TM$  20 % von  $TA$ , ist nur noch eine geringe Verbesserung zu erzielen.

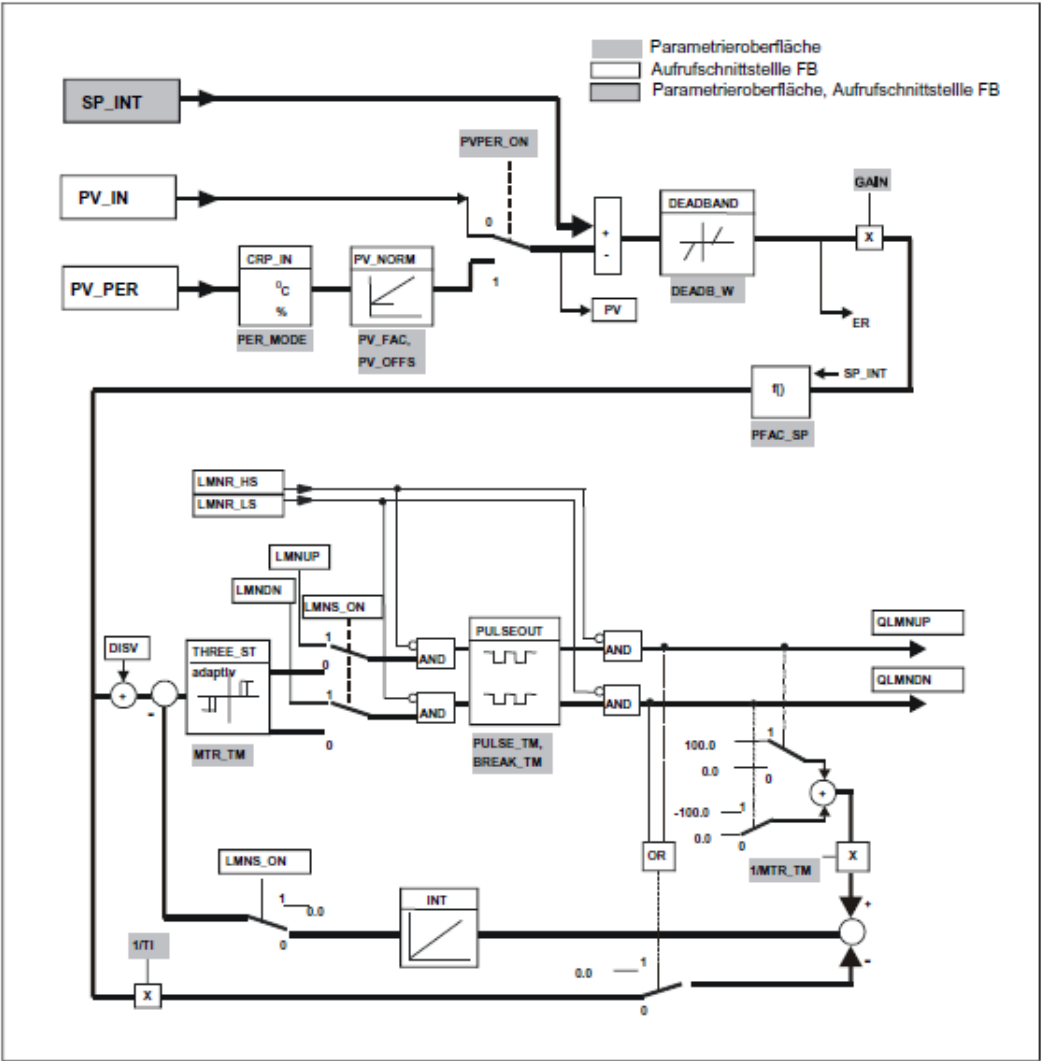
**Störgrößenaufschaltung**

Am Eingang  $DISV$  kann eine Störgröße additiv aufgeschaltet werden.

**Handwertverarbeitung (LMNS\_ON)**

Mit  $LMNS\_ON$  kann zwischen Hand- und Automatikbetrieb umgeschaltet werden. Bei Handbetrieb wird das Stellglied angehalten und der Integrierer (INT) wird intern auf 0 gesetzt. Über  $LMNUP$  und  $LMNDN$  kann das Stellglied AUF und ZU gefahren werden. Das Umschalten in den Automatikbetrieb erfolgt stoßbehaftet. Die anstehende Regeldifferenz führt über  $GAIN$  zu einer sprunghaftigen Änderung der internen Stellgröße. Durch das integral wirkende Stellglied wird jedoch nur eine rampenförmige Ansteuerung des Prozesses bewirkt.

Blockschaltbild



4.6 Zeitfunktionen - "Time Functions"

4.6.1 UDT 60 - WS\_RULES - Regel DB

**Beschreibung** Ihr System muss in einem DB einige Informationen bereitstellen, die von verschiedenen Bausteinen ausgewertet werden. Diesen Datenbaustein erstellen Sie als DB vom Typ der UDT 60 und tragen die für Ihren Ort geltenden Werte (in Lokalzeit!) ein.

Umrechnung Basiszeit < - > Lokalzeit und "Alarm stellen nach Lokalzeit"

Name	Typ	Anfangswert	Kommentar
B2L	STRUCT		Basiszeit < - > Lokalzeit
S	INT	2	Offset Basiszeit -> Lokalzeit [30 min] im Winter zulässig: -24 .. +24
T	INT	2	Differenz Winter- und Sommerzeit [30 min] zulässig: 2

**Regel für: Winterzeit -> Sommerzeit; Default: letzter Sonntag im März; 2:00 Uhr**

Name	Typ	Anfangswert	Kommentar
W2S	STRUCT		In WINTERZEIT angeben!
M	BYTE	B#16#3	Monat der Umstellung
W	BYTE	B#16#9	n-tes Auftreten des Wochentags (1 = erstes, 2 = zweites, .., 9 = letztes)
D	BYTE	B#16#1	Wochentag (Sonntag = 1)
H	BYTE	B#16#2	Stunde

**Regel für: Sommerzeit -> Winterzeit; Default: letzter Sonntag im Oktober, 3:00 Uhr**

Name	Typ	Anfangswert	Kommentar
S2W	STRUCT		In SOMMERZEIT angeben!
M	BYTE	B#16#10	Monat der Umstellung
W	BYTE	B#16#9	n-tes Auftreten des Wochentags (1 = erstes, 2 = zweites, .., 9 = letztes)
D	BYTE	B#16#1	Wochentag (Sonntag = 1)
H	BYTE	B#16#3	Stunde



*Alle Parameter die das Format BYTE haben, werden als BCD-Werte interpretiert!*



*Die Festlegung der Sommer-/Winterzeit-Umschaltunkte durch eine Regel ist in der EU ab dem Jahr 2002 vorgeschrieben.*

#### 4.6.2 FC 61 - BT\_LT - Umrechnung Basiszeit in Lokalzeit

##### Beschreibung

Der FC 61 errechnet die Lokalzeit zu der am Eingang vorgegebenen Basiszeit.

##### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
BT	INPUT	DATE_AND_TIME	Basiszeit
WS_DAT	INPUT	BLOCK_DB	Information zur Zeitzone sowie zur Sommer-/Winterzeit Umschaltung (Regel-DB)
RET_VAL	OUTPUT	INT	Fehlercode
LT	OUTPUT	DATE_AND_TIME	Lokalzeit

**Arbeitsweise** Die am Eingang *BT* eingegebene Basiszeit wird mittels der in einem DB hinterlegten Daten in die Lokalzeit umgerechnet und am Ausgang *LT* ausgegeben. Der DB enthält die Anzahl an 30 Minuten-Einheiten, durch die sich Basis- und Lokalzeit unterscheiden sowie die Differenz zwischen Sommer- und Winterzeit, ebenfalls in Einheiten von 30 Minuten. (Regel-DB) Ergibt sich bei der Berechnung ein Datumsüberlauf wird dies durch einen speziellen Rückgabewert gekennzeichnet.

**Aufrufende OBs** Die FC 61 BT\_LT kann in jeder Ablaufebeine aufgerufen werden.

**Aufrufumgebung** Der FC 61 benutzt intern folgende Funktionen. Diese Funktionen müssen Sie mit den hier angegebenen Nummern in ihr Projekt laden. FC 1 (AD\_DT\_TM), FC 7 (DT\_DAY), FC 35 (SB\_DT\_TM)

#### Ausgabewerte / Fehler

RET_VAL	LT	Beschreibung
0	Lokalzeit	Baustein fehlerfrei gelaufen
1	Lokalzeit	kein Fehler, aber Datumsprung
8082	DT#90-01-01-0:0:0	Ungültige Daten im Regel-Datenbaustein

### 4.6.3 FC 62 - LT\_BT - Umrechnung Lokalzeit in Basiszeit

**Beschreibung** Der FC 62 errechnet die Basiszeit zu der am Eingang vorgegebenen Lokalzeit.

#### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
LT	INPUT	DATE_AND_TIME	Lokalzeit
WS_DAT	INPUT	BLOCK_DB	Information zur Zeitzone sowie zur Sommer-/Winterzeit Umschaltung (Regel-DB)
RET_VAL	OUTPUT	INT	Fehlercode
BT	OUTPUT	DATE_AND_TIME	Basiszeit

**Arbeitsweise** Die am Eingang *LT* eingegebene Lokalzeit wird mittels der in einem DB hinterlegten Daten in die Basiszeit umgerechnet und am Ausgang *BT* ausgegeben. Der DB enthält die Anzahl an 30 Minuten-Einheiten, durch die sich Basis- und Lokalzeit unterscheiden sowie die Differenz zwischen Sommer- und Winterzeit, ebenfalls in Einheiten von 30 Minuten. (Regel-DB) Ergibt sich bei der Berechnung ein Datumsüberlauf wird dies durch einen speziellen Rückgabewert gekennzeichnet.

**"Verbotene Stunde"** Bei der Umschaltung von Winter- nach Sommerzeit wird die Lokalzeit um eine Stunde vorgestellt. Das bedeutet aber, dass die dazwischen liegende Stunde nicht durchlaufen wird. Für einen Zeitpunkt *LT* innerhalb dieser Stunde wird von der FC 62 LT\_BT in Sommerzeit "gedacht". Dies wird mit dem Returnwert 4 bzw. 5 gemeldet.

**"Doppelte Stunde"**

Bei der Umschaltung von Sommer- nach Winterzeit wird die Lokalzeit um eine Stunde zurückgestellt. Das bedeutet aber, dass diese eine Stunde zweimal durchlaufen wird. (Für ME(S)Z gilt dafür die Bezeichnung 2A und 2B). Für einen Zeitpunkt *LT* innerhalb dieser Stunde ist also eine eindeutige Abbildung auf eine Basiszeit nicht möglich. Die FC LT\_BT erhält als Eingangsparameter eine *LT* und muss vor der Umrechnung in *BT* entscheiden, ob der Wert im Sommer oder im Winter liegt. Für den Fall, dass die *LT* innerhalb der doppelten Stunde liegt, wird die *LT* als Winterzeit interpretiert. Dies wird mit dem Returnwert 2 bzw. 3 gemeldet.

**Aufrufende OBs**

Die FC 62 LT\_BT kann in jeder Ablafebene aufgerufen werden.

**Aufrufumgebung**

Der FC 62 benutzt intern folgende Funktionen. Diese Funktionen müssen Sie mit den hier angegebenen Nummern in ihr Projekt laden. FC 1 (AD\_DT\_TM), FC 7 (DT\_DAY), FC 35 (SB\_DT\_TM)

**Ausgabewerte / Fehler**

RET_VAL	LT	Beschreibung
0	Basiszeit	Baustein fehlerfrei gelaufen
1	Basiszeit	Kein Fehler, aber Datumsprung
2	Basiszeit	Die LT am Eingang liegt innerhalb der "doppelten" Stunde
3	Basiszeit	Wie 2, zusätzlich Datumsprung
4	Basiszeit	Die LT am Eingang liegt innerhalb der "verbotenen" Stunde
5	Basiszeit	Wie 4, zusätzlich Datumsprung
8082	DT#90-01-01-0:0:0	Ungültige Daten im Regel-Datenbaustein

**4.6.4 FC 63 - S\_LTINT - Einstellen Uhrzeitalarm in Lokalzeit****Beschreibung**

Der FC 63 stellt den gewünschten Uhrzeitalarm zum vorgegebenen Zeitpunkt. Dieser Zeitpunkt wird in Lokalzeit angegeben.

**Parameter**

Parameter	Deklaration	Datentyp	Beschreibung
OB_NR	INPUT	INT	Nr. des zu startenden OB (zul. 10 – 17)
SDT	INPUT	BLOCK_DB	Startdatum und Uhrzeit in Lokalzeit (siehe SFC 28)

Parameter	Deklaration	Datentyp	Beschreibung
PERIOD	INPUT	INT	Periode vom Ausgangspunkt SDT an: <ul style="list-style-type: none"> <li>■ W#16#0000 = Einmal</li> <li>■ W#16#0201 = Minütlich</li> <li>■ W#16#0401 = Stündlich</li> <li>■ W#16#1001 = Täglich</li> <li>■ W#16#1201 = Wöchentlich</li> <li>■ W#16#1401 = Monatlich</li> <li>■ W#16#1801 = Jährlich</li> <li>■ W#16#2001 = Monatsende</li> </ul>
WS_DAT	INPUT	DATE_AND_TIME	Information zur Zeitzone sowie zur Winter-/ Sommerzeit Umschaltung (s.o.)
RET_VAL	OUTPUT	INT	Fehlercode

**Arbeitsweise**

Die am Eingang *LT* eingegebene Lokalzeit wird mittels der in einem DB hinterlegten Regel in die Basiszeit umgerechnet. Der DB enthält die Anzahl an 30 Minuten-Einheiten, durch die sich Basis- und Lokalzeit unterscheiden sowie die Differenz zwischen Sommer- und Winterzeit, ebenfalls in Einheiten von 30 Minuten. (s.u.) Mit der errechneten Basiszeit wird der angegebene Uhrzeit-Alarm-OB parametrisiert und aktiviert. Ergibt sich bei der Berechnung ein Datumsüberlauf wird dies durch einen speziellen Rückgabewert gekennzeichnet.

**"Verbotene Stunde"**

Bei der Umschaltung von Winter- nach Sommerzeit wird die Lokalzeit um eine Stunde vorgestellt. Das bedeutet aber, dass die dazwischen liegende Stunde nicht durchlaufen wird. Für einen Zeitpunkt *LT* innerhalb dieser Stunde wird von der FC S\_LTINT in Sommerzeit "gedacht". Dies wird mit dem Returnwert (RET\_VAL) 4 bzw. 5 gemeldet.

**"Doppelte Stunde"**

Bei der Umschaltung von Sommer- nach Winterzeit wird die Lokalzeit um eine Stunde zurückgestellt. Das bedeutet aber, dass diese eine Stunde zweimal durchlaufen wird. (Für ME(S)Z gilt dafür die Bezeichnung 2A und 2B). Für einen Zeitpunkt *LT* innerhalb dieser Stunde ist also eine eindeutige Abbildung auf eine Basiszeit nicht möglich. Die FC S\_LTINT erhält als Eingangsparameter eine LT und muss vor der Umrechnung in *BT* entscheiden, ob der Wert im Sommer oder im Winter liegt. Für den Fall, dass die *LT* innerhalb der doppelten Stunde liegt, wird die *LT* als Winterzeit interpretiert. Dies wird mit dem Returnwert (RET\_VAL) 2 bzw. 3 gemeldet.

**Aufrufende OBs**

Die FC S\_LTINT kann in jeder Ablaufeinheit aufgerufen werden. Die FC S\_LTINT benutzt intern folgende Funktionen. Diese Funktionen müssen Sie mit den hier angegebenen Nummern in ihr Projekt laden. FC 7 (DT\_DAY), FC 35 (SB\_DT\_TM)

**Ausgabewerte / Fehler**

RET_VAL	Beschreibung
0	Baustein fehlerfrei gelaufen
1	Kein Fehler, aber Datumssprung
2	Die LT am Eingang war innerhalb der "doppelten" Stunde
3	Wie 2, zusätzlich Datumssprung
4	Die LT am Eingang liegt innerhalb der "verbotenen" Stunde
5	Wie 4, zusätzlich Datumssprung

RET_VAL	Beschreibung
8082	Ungültige Daten im Regel-Datenbaustein
8090	Fehlerhafter Parameter OB_NR
8091	Fehlerhafter Parameter SDT
8092	Fehlerhafter Parameter PERIOD
80A1	Der eingestellte Startzeitpunkt liegt in der Vergangenheit
80A2	OB ist nicht geladen
80A3	OB kann nicht gestartet werden