

**ABB Drives**

**Applikations-Handbuch  
Adaptive Programmierung**

**ABB**



Adaptive Programmierung

## **Applikations-Handbuch**

3AFE64527177 Rev C  
DE  
GÜLTIG AB: 08.04.2005



# Inhaltsverzeichnis

---

## ***Inhaltsverzeichnis***

### ***Einführung in dieses Handbuch***

Kapitel-Übersicht .....	7
Kompatibilität .....	7
Sicherheitsanweisungen .....	7
Zielgruppe .....	7
Verwendung .....	7
Ergänzende Handbücher .....	8

### ***Adaptive Programmierung***

Kapitel-Übersicht .....	9
Was ist die Adaptive Programmierung .....	9
Wie wird ein Programm erstellt .....	10
Wie wird das Programm an die Antriebs-Applikation angeschlossen .....	11
Wie wird die Ausführung des Programms gesteuert .....	11

### ***Funktionsbausteine***

Kapitel-Übersicht .....	13
Grundsätzliche Regeln .....	13
Baustein-Eingänge .....	13
Parameterwert als ein Integer-Eingang .....	14
Wie der Baustein den Eingang verwendet .....	14
Wie wird der Eingang ausgewählt und eingestellt .....	14
Konstante als ein Integer-Eingang .....	15
Einstellung und Anschluss des Eingangs .....	15
Parameterwert als ein boolescher Eingang .....	16
Wie verarbeitet der Baustein den Eingang .....	16
Wie wird der Eingang ausgewählt und eingestellt .....	16
Konstante als ein boolescher Eingang .....	17
Wie wird der Eingang eingestellt und aktiviert .....	17
Zeichenkette als Eingang .....	17
Wie wird der Eingang ausgewählt und eingestellt .....	17
Funktionsbausteine .....	18
ABS .....	18
ADD .....	18
UND .....	18
BITWEISE .....	19
VERGLEICH .....	19
ZÄHLER .....	20
DROT .....	21

EREIGNIS .....	21
FILTER .....	22
MASK SET .....	22
MAX .....	22
MIN .....	23
MULDIV .....	23
LEER .....	23
ODER .....	24
PI .....	24
PI-INIT .....	25
PI-BIPOLAR .....	25
RAMPE .....	25
SR .....	27
BIN SCHALTER .....	27
INT SCHALTER .....	28
AUS VERZ .....	28
EIN VERZ .....	29
TRIGGER .....	29
EXCL ODER .....	30

### ***Istwert-Signale und Parameter des ACS800 Standard-Anwendungsprogramms***

Kapitel-Übersicht .....	31
Istwert-Signale .....	31
Parameter .....	32

### ***Anwender-Dokumentation des Adaptiven Programms***

Kapitel-Übersicht .....	39
-------------------------	----

# Einführung in dieses Handbuch

---

## Kapitel-Übersicht

Dieses Kapitel gibt einen allgemeinen Überblick über das vorliegende Handbuch.

## Kompatibilität

Dieses Handbuch entspricht den Antriebs-Anwendungsprogrammen, in die die Adaptive Programmierung integriert ist.

## Sicherheitsanweisungen

Befolgen Sie alle Sicherheitsanweisungen, die Sie mit der Lieferung des Antriebs erhalten haben.

- Vor der Installation, Inbetriebnahme oder Verwendung des Antriebs lesen Sie die **Sicherheitsanweisungen vollständig** durch. Die vollständigen Sicherheitsanweisungen finden Sie auf den ersten Seiten des Hardware-Handbuches.
- Vor der Änderung der Standard-Einstellungen lesen Sie aufmerksam die **spezifischen Warnungen und Hinweise zu den Software-Funktionen**. Zu jeder Funktion sind im Programmierhandbuch in den Abschnitten, in denen die vom Anwender einstellbaren Parameter beschrieben sind, Warnungen und Hinweise enthalten, die Sie beachten müssen.

## Zielgruppe

Der Leser dieses Handbuches soll:

- über Erfahrungen und Kenntnisse der elektrischen Verkabelung verfügen und die elektronischen Komponenten sowie die in Elektro-Schaltplänen verwendeten Symbole kennen.
- Erfahrungen oder eine Schulung für Installation, Betrieb oder Wartung von ABB-Antrieben werden nicht benötigt.

## Verwendung

Dieses Handbuch ist eine Ergänzung des Programmierhandbuches des jeweiligen Antriebs-Anwendungsprogramms. Das Programmierhandbuch enthält die grundlegenden Informationen über die Antriebs-Parameter einschließlich der Parameter der Adaptiven Programmierung. Dieses Handbuch enthält weitere detaillierte Informationen über die Adaptive Programmierung:

- was leistet die Adaptive Programmierung,
- wie wird ein Programm erstellt,
- wie werden die Funktionsbausteine eingesetzt,

- wie wird das Programm dokumentiert,
- welche Parametereinstellungen und Istwert-Signale des ACS800 Standard-Anwendungsprogramms sind für die Adaptive Programmierung wichtig.

## **Ergänzende Handbücher**

Die Anwender-Dokumentation des Antriebs besteht aus:

- Programmierhandbuch (das entsprechende Handbuch wird mit der Antriebseinheit geliefert),
- Hardware-Handbuch (das entsprechende Handbuch wird mit der Antriebseinheit geliefert),
- Handbücher und Anleitungen für optionales Zubehör, Geräte und Programme (gehören jeweils zum Lieferumfang der Optionen).

# Adaptive Programmierung

---

## Kapitel-Übersicht

Dieses Kapitel beschreibt die Grundlagen der Adaptiven Programmierung und gibt eine Anleitung zur Erstellung von Programmen.

## Was ist die Adaptive Programmierung

Konventionell steuert der Anwender einen Antriebsprozess durch die Einstellung von Parametern. Jeder Parameter hat eine festgelegte Auswahl von Einstellungen oder einen bestimmten Einstellbereich. Mit Parametern ist eine Programmierung einfach, aber die Einstellmöglichkeiten sind begrenzt: eine weitergehende Anpassung an den Betriebsprozess ist nicht möglich. Mit der Adaptiven Programmierung kann jetzt ohne zusätzliche Programmier-Tools oder -Sprachen eine spezifische Anpassung des Antriebs an den Anwendungsprozess vorgenommen werden:

- Das Programm wird aus Funktionsbausteinen erstellt.
- Die Steuertafel ist gleichzeitig das Programmier-Tool.
- Der Anwender kann das Programm durch Einträge und Zeichnungen auf Blockdiagramm-Vorlagen dokumentieren.

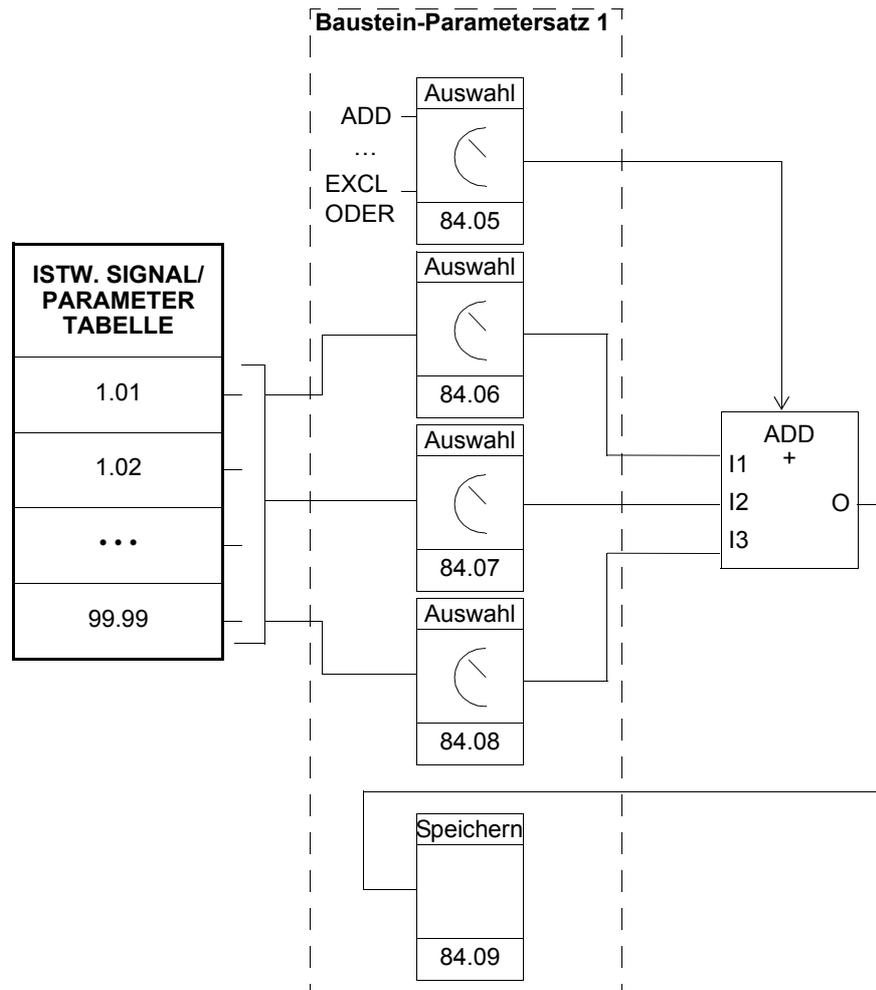
Die maximale Größe eines Adaptiven Programms beträgt 15 Funktionsbausteine. Das Programm kann aus mehreren separaten Funktionen erstellt werden.

## Wie wird ein Programm erstellt

Der Programmierer verbindet einen Funktionsbaustein über einen Baustein-Parametersatz mit anderen Bausteinen. Die Parametersätze werden auch für das Lesen von Werten aus dem Antriebs-Applikationsprogramm und die Übertragung von Daten in das Antriebs-Applikationsprogramm verwendet. Jeder Baustein-Parametersatz besteht aus fünf Parametern.

Die Abbildung zeigt die Verwendung von Baustein-Parametersatz 1 des ACS800 Standard Anwendungsprogramms (Parameter 84.05 bis 84.09):

- Parameter 84.05 wählt den Typ des Funktionsbausteins.
- Parameter 84.06 wählt den Anschluss, der an den Eingang I1 des Funktionsbausteins angeschlossen wird.
- Parameter 84.07 wählt den Anschluss, der an den Eingang I2 des Funktionsbausteins angeschlossen wird.
- Parameter 84.08 wählt den Anschluss, der an den Eingang I3 des Funktionsbausteins angeschlossen wird.
- Parameter 84.09 speichert den Wert des Funktionsbaustein-Ausgangs. Der Anwender kann diesen Parameterwert nicht editieren/selbst einstellen, dies geschieht nur durch das Programm.







# Funktionsbausteine

---

## Kapitel-Übersicht

In diesem Kapitel werden die Funktionsbausteine beschrieben.

## Grundsätzliche Regeln

**Die Verwendung des Eingangs I1 ist zwingend notwendig (er darf nicht ohne Zuordnung eines Anschlusses bleiben).** Die Verwendung der Eingänge I2, I3, usw. ist bei den meisten Bausteinen freigestellt. Es gilt die Faustregel, dass ein Eingang ohne einen Anschluss keine Auswirkung auf den Ausgang/das Ausgangssignal des Bausteins hat.

## Baustein-Eingänge

Die Eingänge der Funktionsbausteine lesen drei Eingangsdaten-Formate:

- integer
- boolesch
- Text-Zeichenkette

Abhängig vom Baustein ist das verwendete Format unterschiedlich. Zum Beispiel verwendet der Baustein ADD als Eingang Integerwerte und der Baustein ODER als Eingang boolesche Werte. Das Format Text-Zeichenkette wird nur vom Baustein EREIGNIS verwendet.

---

**Hinweis:** Die Eingänge/Eingangssignale der Bausteine werden jeweils gelesen, wenn die Ausführung der einzelnen Funktionsbausteine beginnt und nicht gleichzeitig bei allen Funktionsbausteinen!

---

## Parameterwert als ein Integer-Eingang

### Wie der Baustein den Eingang verwendet

Der Baustein liest den eingestellten Wert als Integerwert.

---

**Hinweis:** Der Parameterwert der als Eingang gewählt wird, sollte ein reeller Wert oder Integerwert sein. Besitzt der Wert nicht das Integerformat als Standard, wird er vom Baustein umgewandelt. Die Integer- (Feldbus-) Skalierung für jeden Parameter wird im *Programmierhandbuch* angegeben.

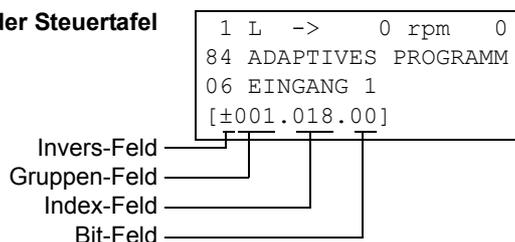
---

### Wie wird der Eingang ausgewählt und eingestellt

- Zum Auswahl-Parameter für den Eingang des Bausteins scrollen (AUF-/AB-Tasten der Steuertafel) und in den Edit-Modus umschalten (Enter-Taste).
- Entsprechend der Adresse, von der der Wert für den Eingang gelesen werden soll, die Werte in die Invers-, Gruppen-, Index- und Bit-Felder eingeben (Doppelpfeil- und Pfeil-Tasten der Steuertafel).

Die Abbildung unten zeigt das Anzeigefeld der Steuertafel mit dem Auswahl-Parameter für den Eingang I1 im Edit-Modus. Der Wert wird invertiert, wenn im Invers-Feld ein Minus ( - ) Zeichen eingegeben/angezeigt wird. Das Bit-Auswahlfeld ist bei den Eingangstypen Integerwert oder Text-Zeichenkette nicht wirksam.

Anzeigefeld der Steuertafel



**Beispiel:** Einstellung von Analogeingang AI1 auf 5,8 V bei einem Antrieb mit ACS800 Standard-Anwendungsprogramm. Wie wird das Signal an den MAX-Baustein im Adaptiven Programm angeschlossen? Welcher Wert ist als Baustein-Eingang einzugeben?

AI1 wird an den Baustein wie folgt angeschlossen:

- Zum Auswahl-Parameter für den Eingang I1 scrollen und in den Edit-Modus umschalten (Enter-Taste).
- Den Wert im Gruppenfeld auf 1 und den Wert im Index-Feld auf 18 einstellen. (Der Wert von AI1 wird intern als Istwert-Signal 1.18 gespeichert.)

Der Wert am Eingang des Baustein beträgt 5800, denn die Integer-Skalierung von Istwert-Signal 1.18 beträgt 0,001 V = 1 (gemäß Programmierhandbuch für das Standard-Anwendungsprogramm).

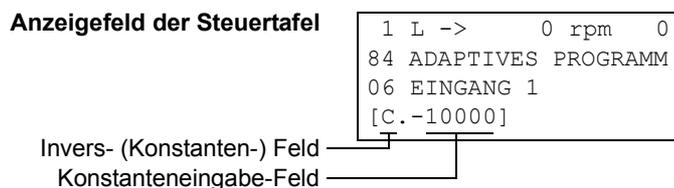
## Konstante als ein Integer-Eingang

### Einstellung und Anschluss des Eingangs

#### Option 1

- Zum Auswahl-Parameter für den Eingang des Baustein scrollen und in den Edit-Modus umschalten (Enter-Taste).
- Im Invers-Feld C einstellen (Doppelpfeil- und Pfeil-Tasten). Die Darstellung der Zeilenanzeige wechselt. Der Rest der Zeile ist jetzt ein Konstanten-Feld.
- Den Wert der Konstante in das Konstanten-Feld eingeben (Doppelpfeil- und Pfeil-Tasten).
- Mit der Enter-Taste bestätigen.

Die Abbildung unten zeigt das Anzeigefeld der Steuertafel mit dem Auswahl-Parameter für den Eingang I1 im Edit-Modus und angezeigtem Konstanten-Feld. Die Konstante hat einen zulässigen Wertebereich von -32768 bis 32767. Während das Adaptive Programm läuft, kann die Konstante nicht geändert werden.



#### Option 2

- Die Konstante auf einen Parameterwert einstellen.
- Den Konstanten-Wert wie vorher beschrieben mit dem Auswahl-Parameter an den Eingang des Baustein anschließen.

Die Konstante kann geändert werden, während das Adaptive Programm läuft. Sie hat einen zulässigen Wertebereich von -8388608 bis 8388607.

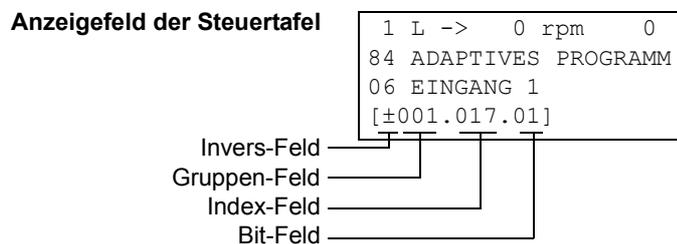
## Parameterwert als ein boolescher Eingang

Wie verarbeitet der Baustein den Eingang

- Der Baustein liest den eingestellten Wert als Integer-Wert.
- Der Baustein verwendet das Bit im Bit-Feld als den booleschen Eingang.

Mit Bitwert 1 ist der boolesche Wert wahr und mit 0 ist der boolesche Wert unwahr.

**Beispiel:** Die Abbildung unten zeigt das Anzeigefeld der Steuertafel mit dem Auswahl-Parameter für den Wert von Eingang I1 mit Anschluss an ein Bit, das den Status von Digitaleingang DI2 anzeigt. (Im ACS800 Standard-Anwendungsprogramm wird der Status der Digitaleingänge intern als Istwert-Signal 1.17 DI6-1 STATUS gespeichert. Bit 1 entspricht DI2, Bit 0 gehört zu DI1.)



Wie wird der Eingang ausgewählt und eingestellt

Siehe oben [Parameterwert als ein Integer-Eingang](#).

---

**Hinweis:** Der Parameter, der als Eingang gewählt ist, sollte einen gepackten booleschen Wert haben (binäres Datenwort). Siehe Programmierhandbuch.

---

## Konstante als ein boolescher Eingang

*Wie wird der Eingang eingestellt und aktiviert*

- Zum Auswahl-Parameter für den Eingang des Baustein blättern (scrollen) und in den Edit-Modus umschalten (Enter-Taste).
- Im Invers-Feld C einstellen (Doppelpfeil- und Pfeil-Tasten). Die Darstellung der Zeilenanzeige wechselt. Der Rest der Zeile ist jetzt ein Konstanten-Feld.
- Die Konstante eingeben. Ist der boolesche Wert wahr erforderlich, die Konstante auf -1 setzen. Ist der boolesche Wert unwahr erforderlich, eine 0 einstellen.
- Mit der Enter-Taste bestätigen.

## Zeichenkette als Eingang

*Wie wird der Eingang ausgewählt und eingestellt*

Eine Zeichenkette als Eingang ist nur beim Baustein EREIGNIS erforderlich.

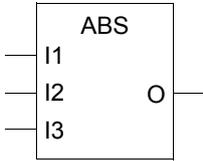
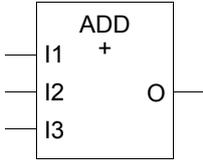
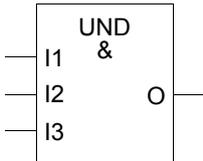
Für die Abfolge der Eingangsauswahl gelten die gleichen Angaben wie im vorstehenden Abschnitt [Parameterwert als ein Integer-Eingang](#). Das Bit-Einstell-Feld ist nicht aktiviert.

---

**Hinweis:** Der als Eingang eingestellte Parameter muss eine Zeichenkette als Wert haben. Das ACS800 Standard-Anwendungsprogramm enthält Parameter in Gruppe 85 NUTZER KONSTANTEN, die als Eingänge im Format Zeichenkette eingestellt werden können.

---

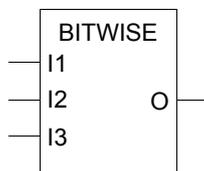
## Funktionsbausteine

<b>ABS</b>	<b>Typ</b>	Arithmetische Funktion																																															
	<b>Darstellung</b>																																																
	<b>Verwendung</b>	Ausgang ist der absolute Wert (Betrag) von Eingang I1 multipliziert mit I2 und dividiert durch I3. $O =  I1  \cdot I2 / I3$																																															
	<b>Anschlüsse</b>	Eingang I1, I2 und I3: 24 Bit Integerwerte (23 Bits + Vorzeichen) Ausgang (O): 24 Bit integer (23 Bits + Vorzeichen)																																															
<b>ADD</b>	<b>Typ</b>	Arithmetische Funktion																																															
	<b>Darstellung</b>																																																
	<b>Verwendung</b>	Ausgang ist die Summe der Eingänge. $O = I1 + I2 + I3$																																															
	<b>Anschlüsse</b>	Eingang I1, I2 und I3: 24 Bit Integerwerte (23 Bits + Vorzeichen) Ausgang (O): 24 Bit integer (23 Bits + Vorzeichen)																																															
<b>UND</b>	<b>Typ</b>	Logische Funktion																																															
	<b>Darstellung</b>																																																
	<b>Verwendung</b>	Der Ausgang ist wahr, wenn alle angeschlossenen Eingänge wahr sind. Andernfalls ist der Ausgang unwahr. Tabelle für Bedingung wahr:																																															
		<table border="1" data-bbox="497 1585 1385 1944"> <thead> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O (binär)</th> <th>O (angezeigter Wert)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>unwahr (alle Bits 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>unwahr (alle Bits 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>unwahr (alle Bits 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>unwahr (alle Bits 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>unwahr (alle Bits 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>unwahr (alle Bits 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>unwahr (alle Bits 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>wahr (alle Bits 1)</td> <td>-1</td> </tr> </tbody> </table>			I1	I2	I3	O (binär)	O (angezeigter Wert)	0	0	0	unwahr (alle Bits 0)	0	0	0	1	unwahr (alle Bits 0)	0	0	1	0	unwahr (alle Bits 0)	0	0	1	1	unwahr (alle Bits 0)	0	1	0	0	unwahr (alle Bits 0)	0	1	0	1	unwahr (alle Bits 0)	0	1	1	0	unwahr (alle Bits 0)	0	1	1	1	wahr (alle Bits 1)	-1
I1	I2	I3	O (binär)	O (angezeigter Wert)																																													
0	0	0	unwahr (alle Bits 0)	0																																													
0	0	1	unwahr (alle Bits 0)	0																																													
0	1	0	unwahr (alle Bits 0)	0																																													
0	1	1	unwahr (alle Bits 0)	0																																													
1	0	0	unwahr (alle Bits 0)	0																																													
1	0	1	unwahr (alle Bits 0)	0																																													
1	1	0	unwahr (alle Bits 0)	0																																													
1	1	1	wahr (alle Bits 1)	-1																																													

**Anschlüsse** Eingang I1, I2 und I3: boolesche Werte  
Ausgang (O): 24 Bit Integerwert (gepackt boolesch)

**BITWEISE** Typ Arithmetische Funktion

**Darstellung**



**Verwendung** Der Block vergleicht die Bits von drei 24 Bitwort-Eingängen und bildet die Ausgangsbits folgendermaßen:

$$O = (I1 \text{ ODER } I2) \text{ UND } I3$$

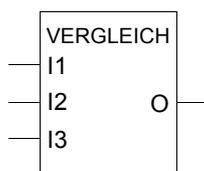
**Beispiel**

I1	I2	I3	O
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	0
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

**Anschlüsse** Eingang I1: 24 Bit Integerwert (gepackter boolescher Wert)  
Eingang I2: 24 Bit Integerwert (gepackter boolescher Wert)  
Eingang I3: 24 Bit Integerwert (gepackter boolescher Wert)  
Ausgang (O): 24 Bit Integerwert (gepackter boolescher Wert)

**VERGLEICH** Typ Komparative Funktion

**Darstellung**



**Verwendung** Ausgang Bits 0, 1 und 2:  
 - wenn  $I1 > I2$ ,  $O = \dots 001$  (Ausgang Bit 0 gesetzt.)  
 - wenn  $I1 = I2$ ,  $O = \dots 010$  (Ausgang Bit 1 gesetzt.)  
 - wenn  $I1 < I2$ ,  $O = \dots 100$  (Ausgang Bit 2 gesetzt.)  
 Ausgang Bit 3:  
 - wenn  $I1 > I2$ ,  $O = \dots 1xxx$  (Ausgang Bit 3 gesetzt und bleibt gesetzt bis  $I1 < I2 - I3$ , danach wird Bit 3 zurückgesetzt.)  
 Ausgangswert in der Anzeige:

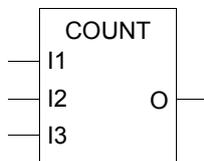
Bit 0	Bit 1	Bit 2	Bit 3	O (angezeigter Wert)
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
0	0	1	0	4
0	0	0	1	8
1	0	0	1	9
0	1	0	1	10
0	0	1	1	12

**Anschlüsse** Eingang I1, I2 und I3: 24 Bit Integerwerte (23 Bits + Vorzeichen)  
 Ausgang (O): 24 Bit integer (gepackt boolesch)

**ZÄHLER**

**Typ** Zähler-Funktion

**Darstellung**

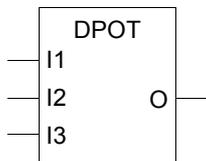


**Verwendung** Die Zähler-Funktion zählt die Anstiegsflanken des Eingangs I1.  
 Der Zähler wird durch die Anstiegsflanke von Eingang I2 zurückgesetzt und auf den mit Eingang I3 eingestellten Wert begrenzt.  
 I1: Trigger-Eingang  
 I2: Reset  
 I3: Max. Grenzwert des Zähler ( $B0 \dots B19 \rightarrow 0 \dots 1048575$ )  
 O: Wert des Zähler ( $B0 \dots B19 \rightarrow 0 \dots 1048575$ ) und Status des Zählers (B20). B20 = 1: Zähler am max. Grenzwert oder Eingang I3 ist negativ.

**Anschlüsse** Eingang I1 und I2: boolesche Werte  
 Eingang I3: 24 Bit Integerwert (20 Bits vom Zähler verwendet)  
 Ausgang (O): 24 Bit Integerwert (20 Bits für Zählerwert und 4 Anzeigebits)

**DPOT**      **Typ**      Zähler-Funktion

**Darstellung**



**Verwendung**

Die digital gesteuerte Rampenfunktion inkrementiert oder dekrementiert den Ausgang O (zählt aufwärts oder abwärts) entsprechend der Steuereingänge I1 und I2. Der Eingang I1 aktiviert die positive Richtung und I2 die negative Richtung. Sind beide Eingänge aktiviert, passiert nichts. Der Schritt wird von Eingang I3 definiert.

Eingang I1: Zähler aufwärts

Eingang I2: Zähler abwärts

Eingang I3: Rampenschritt in positive/negative Richtung (Schritt/Sekunde).

**Hinweis:** Die interne Berechnung arbeitet mit 48 Bit Genauigkeit um Offset-Fehler zu vermeiden.

**Anschlüsse**

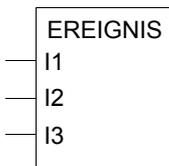
Eingang I1 und I2: Boolesche Werte

Eingang I3: 24 Bit Integerwert (23 Bits + Vorzeichen)

Ausgang (O): 24 Bit Integerwert (23 Bits + Vorzeichen)

**EREIGNIS**      **Typ**      Ereignis-Funktion

**Darstellung**



**Verwendung**

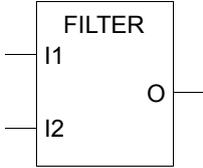
Eingang I1 löst die Ereignis-Meldung aus. I2 wählt den Parameter-Index von dem die Ereignis-Meldung (Text-Zeichenkette)gelesen wird. I3 wählt den Ereignis-Typ (Warnung oder Fehler).

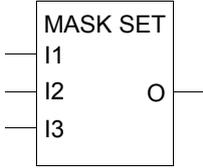
I1	I2	I3	Folge
0->1			Baustein aktiviert das Ereignis
0			Baustein deaktiviert das Ereignis
	I2		Inhalte der Ereignis-Meldung
		0	Ereignis-Typ: Warnung
		1	Ereignis-Typ: Fehler
		2	Ereignis-Typ: Ereignis

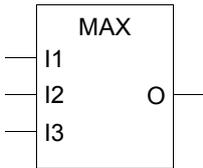
**Anschlüsse**

Eingang I1, I3: 24 Bit Integerwerte (23 Bits + Vorzeichen)

Eingang I2: Zeichenkette (zwingend)

<b>FILTER</b>	<b>Typ</b>	Filter-Funktion
	<b>Darstellung</b>	
	<b>Verwendung</b>	<p>Ausgang ist der gefilterte Wert von Eingang I1. Eingang I2 ist die Filter-Zeit.</p> $O = I1 \cdot (1 - e^{-t/I2})$ <p><b>Hinweis:</b> Die interne Berechnung erfolgt mit einer Genauigkeit von 48 Bits um Offset-Fehler zu vermeiden.</p>
	<b>Anschlüsse</b>	<p>Eingang I1: 24 Bit Integerwert (23 Bits + Vorzeichen)</p> <p>Eingang I2: 24 Bit Integerwert (23 Bits + Vorzeichen). Der Wert 1 entspricht 1 ms.</p> <p>Ausgang (O): 24 Bit integer (23 Bits + Vorzeichen)</p>

<b>MASK SET</b>	<b>Typ</b>	Logische Funktion																																																
	<b>Darstellung</b>																																																	
	<b>Verwendung</b>	<p>Diese Blockfunktion setzt die in I2 definierten Bits in I1 oder setzt sie zurück.</p> <p>Eingang I1: Wort-Eingang</p> <p>Eingang I2: Einstellung Worteingang</p> <p>Ausgang I3: Setzen/Rücksetzen I2 in I1.</p>																																																
		<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th colspan="4">Beispiel mit SETZEN</th> </tr> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Wahr</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>Wahr</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Wahr</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>Wahr</td> <td>1</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th colspan="4">Beispiel mit RÜCKSETZEN</th> </tr> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Unwahr</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>Unwahr</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Unwahr</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>Unwahr</td> <td>0</td> </tr> </tbody> </table>	Beispiel mit SETZEN				I1	I2	I3	O	0	0	Wahr	0	1	0	Wahr	1	1	1	Wahr	1	0	1	Wahr	1	Beispiel mit RÜCKSETZEN				I1	I2	I3	O	0	0	Unwahr	0	1	0	Unwahr	1	1	1	Unwahr	0	0	1	Unwahr	0
Beispiel mit SETZEN																																																		
I1	I2	I3	O																																															
0	0	Wahr	0																																															
1	0	Wahr	1																																															
1	1	Wahr	1																																															
0	1	Wahr	1																																															
Beispiel mit RÜCKSETZEN																																																		
I1	I2	I3	O																																															
0	0	Unwahr	0																																															
1	0	Unwahr	1																																															
1	1	Unwahr	0																																															
0	1	Unwahr	0																																															
	<b>Anschlüsse</b>	<p>Eingang I1: 24 Bit Integerwert (gepackter boolescher Wert)</p> <p>Eingang I2: 24 Bit Integerwert (gepackter boolescher Wert)</p> <p>Eingang I3: Boolescher Wert</p> <p>Ausgang (O): 24 Bit Integerwert (gepackter boolescher Wert)</p>																																																

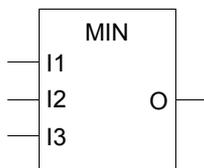
<b>MAX</b>	<b>Typ</b>	Komparative Funktion
	<b>Darstellung</b>	

**Verwendung** Ausgang ist der höchste Eingangswert.  
 $O = \text{MAX}(I1, I2, I3)$

**Anschlüsse** Eingang I1, I2 und I3: 24 Bit Integerwerte (23 Bits + Vorzeichen)  
 Ausgang (O): 24 Bit integer (23 Bits + Vorzeichen)

**MIN** **Typ** Komparative Funktion

**Darstellung**

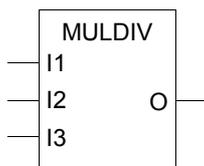


**Verwendung** Ausgang ist der niedrigste Eingangswert.  
 $O = \text{MIN}(I1, I2, I3)$

**Anschlüsse** Eingang I1, I2 und I3: 24 Bit Integerwerte (23 Bits + Vorzeichen)  
 Ausgang (O): 24 Bit integer (23 Bits + Vorzeichen)

**MULDIV** **Typ** Arithmetische Funktion

**Darstellung**

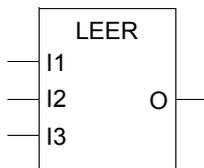


**Verwendung** Der Ausgang ist das Produkt von Eingang I1 und Eingang I2  
 dividiert durch Eingang I3.  
 $O = (I1 \cdot I2) / I3$

**Anschlüsse** Eingang I1, I2 und I3: 24 Bit Integerwerte (23 Bits + Vorzeichen)  
 Ausgang (O): 24 Bit integer (23 Bits + Vorzeichen)

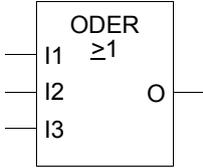
**LEER** **Typ** -

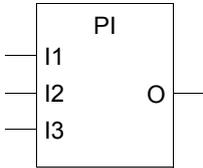
**Darstellung**

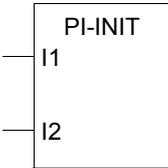
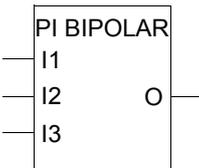
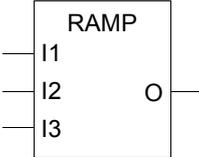


**Verwendung** Dieser Baustein ist ohne Funktion.

**Anschlüsse** -

<b>ODER</b>	<b>Typ</b>	Logische Funktion																																								
	<b>Darstellung</b>																																									
	<b>Verwendung</b>	<p>Der Ausgang ist wahr, wenn einer der Eingänge wahr ist. Tabelle für Bedingung wahr:</p> <table border="1"> <thead> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O (binär)</th> <th>O (angezeigter Wert)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>unwahr (alle Bits 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>wahr (alle Bits 1)</td> <td>-1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>wahr (alle Bits 1)</td> <td>-1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>wahr (alle Bits 1)</td> <td>-1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>wahr (alle Bits 1)</td> <td>-1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>wahr (alle Bits 1)</td> <td>-1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>wahr (alle Bits 1)</td> <td>-1</td> </tr> </tbody> </table>	I1	I2	I3	O (binär)	O (angezeigter Wert)	0	0	0	unwahr (alle Bits 0)	0	0	0	1	wahr (alle Bits 1)	-1	0	1	0	wahr (alle Bits 1)	-1	0	1	1	wahr (alle Bits 1)	-1	1	0	0	wahr (alle Bits 1)	-1	1	1	0	wahr (alle Bits 1)	-1	1	1	1	wahr (alle Bits 1)	-1
I1	I2	I3	O (binär)	O (angezeigter Wert)																																						
0	0	0	unwahr (alle Bits 0)	0																																						
0	0	1	wahr (alle Bits 1)	-1																																						
0	1	0	wahr (alle Bits 1)	-1																																						
0	1	1	wahr (alle Bits 1)	-1																																						
1	0	0	wahr (alle Bits 1)	-1																																						
1	1	0	wahr (alle Bits 1)	-1																																						
1	1	1	wahr (alle Bits 1)	-1																																						
	<b>Anschlüsse</b>	<p>Eingang I1, I2 und I3: Boolesche Werte Ausgang (O): 24 Bit Integerwert (gepackt boolesch)</p>																																								

<b>PI</b>	<b>Typ</b>	PI-Regler
	<b>Darstellung</b>	
	<b>Verwendung</b>	<p>Ausgang ist Eingang I1 multipliziert mit I2/100 plus Integral von I1 multipliziert mit I3/100.  <math display="block">O = I1 \cdot I2/100 + (I3/100) \cdot \int I1</math> <b>Hinweis:</b> Die interne Berechnung erfolgt mit einer Genauigkeit von 48 Bits um Offset-Fehler zu vermeiden.</p>
	<b>Anschlüsse</b>	<p>Eingang I1: 24 Bit Integerwert (23 Bit + Vorzeichen)  Eingang I2:  - 24 Bit Integerwert (23 Bit + Vorzeichen)  - Verstärkungsfaktor. 100 entspricht 1. 10 000 entspricht 100.  Eingang I3:  - Integrationskoeffizient. 100 entspricht 1. 10 000 entspricht 100.  Ausgang (O): 24 Bit Integerwert (23 Bits + Vorzeichen).  Der Bereich ist begrenzt auf 0...10000.</p>

<b>PI-INIT</b>	<b>Typ</b>	Initialisierungs-Baustein für den PI-Regler
	<b>Darstellung</b>	
	<b>Verwendung</b>	<p>Dieser Baustein initialisiert den PI-Baustein. Wenn Eingang I1 die Bedingung wahr erfüllt, schreibt der Baustein den Wert von I2 in den Ausgang des PI-Bausteins. Wenn I1 die Bedingung unwahr erfüllt, gibt der Baustein den Ausgang des PI-Regler Bausteins frei, womit der Normalbetrieb mit dem eingestellten Ausgang fortgesetzt wird. <b>Hinweis:</b> Der Baustein darf nur gemeinsam mit dem PI-Regler-Baustein verwendet werden. Der Baustein muss im Programm direkt nach dem PI-Baustein folgen.</p>
	<b>Anschlüsse</b>	<p>Eingang I1: boolescher Wert Eingang I2: 24 Bit Integerwert (23 Bits + Vorzeichen)</p>
<b>PI-BIPOLAR</b>	<b>Typ</b>	PI-Regler
	<b>Darstellung</b>	
	<b>Verwendung</b>	<p>Siehe Baustein PI. Ausnahme Ausgang (O) Bereich: -10000...10000.</p>
<b>RAMPE</b>	<b>Typ</b>	Rampen-Funktion
	<b>Darstellung</b>	

**Verwendung**

Der Baustein verwendet Eingang I1 als Sollwert. Die Werte (Eingänge I2 und I3) erhöhen oder vermindern in Schritten den Ausgang (O) solange, wie der Ausgang vom Grenzwert I1 verschieden ist. Wenn  $O = I1$ , bleibt der Ausgang unverändert.

Eingang I1: Sollwert

Eingang I2: Schritt in positive Richtung (Schritt/Sek.). Erhöht den Ausgang, wenn  $O < I1$ .

Eingang I3: Schritt in negative Richtung (Schritt/Sek.). Vermindert den Ausgang, wenn  $O > I1$ .

$$O_n = O_{n-1} + I2 \text{ wenn } I1 > O$$

$$O_n = O_{n-1} - I3 \text{ wenn } I1 < O$$

$$O_n = I1 \text{ wenn } I1 = O$$

**Beispiel:**

Eingang I1: 0 -> 150 -> -100 -> 0

Eingang I2: 100 Schritt/Sek.

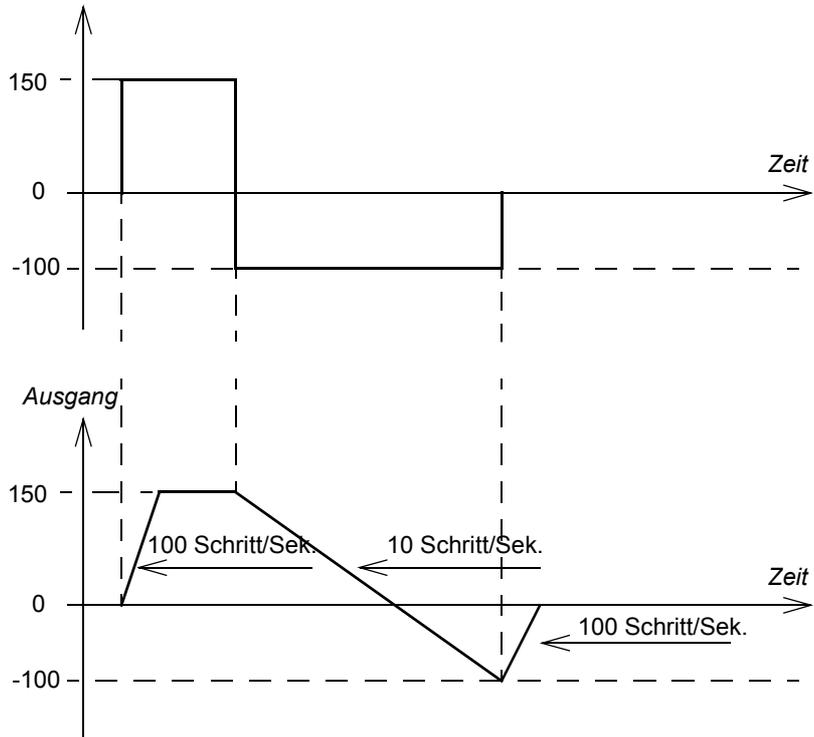
Eingang I3: 10 Schritt/Sek.

Ausgang:

Wird erhöht: Rampenschritt von Eingang I2

Wird vermindert: Rampenschritt von Eingang I3

*Eingang I1*



**Anschlüsse**

Eingang I1, I2 und I3: 24 Bit Integerwerte (23 Bits + Vorzeichen)

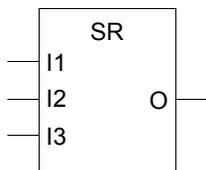
Ausgang (O): 24 Bit Integerwerte (23 Bits + Vorzeichen)

**Hinweis:** Die interne Berechnung erfolgt mit einer Genauigkeit von 48 Bits um Offset-Fehler zu vermeiden.

**SR**

**Typ** Logische Funktion

**Darstellung**



**Verwendung** Set/Reset Baustein. Eingang I1 aktiviert Set und I2 und I3 aktivieren Reset des Ausgangs.

- wenn I1, I2 und I3 unwahr sind, bleibt der aktuelle Wert am Ausgang.
- wenn I1 wahr ist und I2 and I3 unwahr sind, ist der Ausgang wahr.
- wenn I2 oder I3 wahr ist, ist der Ausgang unwahr.

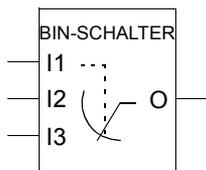
I1	I2	I3	O (binär)	O (angezeigter Wert)
0	0	0	Ausgang	Ausgang
0	0	1	unwahr (alle Bits 0)	0
0	1	0	unwahr (alle Bits 0)	0
0	1	1	unwahr (alle Bits 0)	0
1	0	0	wahr (alle Bits 1)	-1
1	0	1	unwahr (alle Bits 0)	0
1	1	0	unwahr (alle Bits 0)	0
1	1	1	unwahr (alle Bits 0)	0

**Anschlüsse** Eingang I1, I2 und I3: Boolesche Werte  
Ausgang (O): 24 Bit Integerwert (23 Bits + Vorzeichen)

**BIN SCHALTER**

**Typ** Logische Funktion

**Darstellung**



**Verwendung** Ausgang ist gleich Eingang I2, wenn Eingang I1 wahr ist oder der Ausgang ist gleich Eingang I3 wenn Eingang I1 unwahr ist.

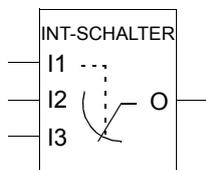
I1	I2	I3	O	O (angezeigter Wert)
0	I2	I3	I3	wahr = -1
1	I2	I3	I2	unwahr = 0

**Anschlüsse** Eingang I1, I2 und I3: Boolesche Werte  
Ausgang (O): 24 Bit Integerwert (gepackt boolesch)

**INT SCHALTER**

Typ Logische Funktion

Darstellung



Verwendung Ausgang ist gleich Eingang I2, wenn Eingang I1 wahr ist und er ist gleich Eingang I3 wenn Eingang I1 unwahr ist.

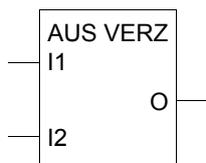
I1	I2	I3	O
0	I2	I3	I3
1	I2	I3	I2

Anschlüsse Eingang I1: boolescher Wert  
 Eingang I2 und I3: 24 Bit Integerwerte (23 Bits + Vorzeichen)  
 Ausgang (O): 24 Bit Integerwert (23 Bits + Vorzeichen)

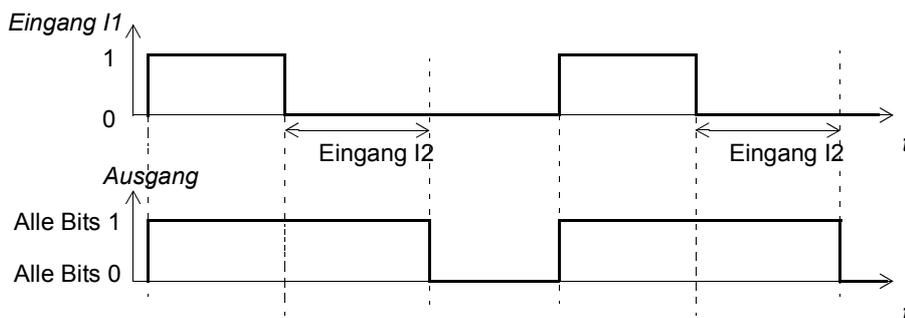
**AUS VERZ**

Typ Zeit-Funktion

Darstellung

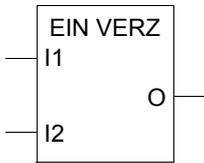
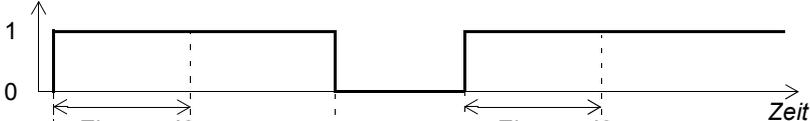
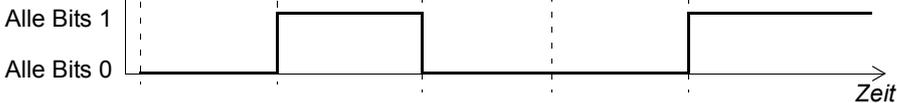


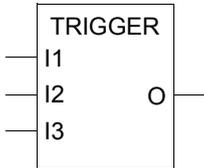
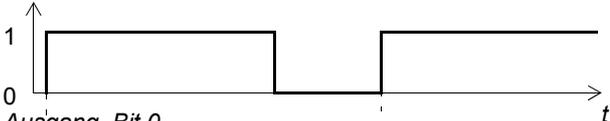
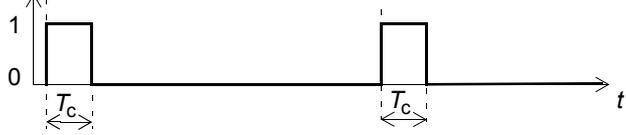
Verwendung Ausgang ist wahr, wenn Eingang I1 wahr ist. Der Ausgang ist unwahr, wenn Eingang I1 für eine Zeit gleich oder länger Eingang I2 die Bedingung unwahr erfüllt hat .



Angezeigte Werte: wahr = -1, unwahr = 0.

Anschlüsse Eingang I1: boolescher Wert  
 Eingang I2: 24 Bit Integerwert (23 Bits + Vorzeichen). Der Wert 1 entspricht 1 ms.  
 Ausgang (O):  
 - 24 Bit Integerwert (gepackt boolesch)

<b>EIN VERZ</b>	Type	Zeit-Funktion
	Darstellung	
	Verwendung	<p>Ausgang ist wahr, wenn Eingang I1 gleich oder länger Eingang I2 die Bedingung wahr erfüllt hat. Der Ausgang ist unwahr, wenn der Eingang unwahr ist.</p> <div style="display: flex; align-items: flex-start;"> <div style="margin-right: 20px;"> <p><i>Eingang I1</i></p>  </div> <div style="margin-right: 20px;"> <p><i>Eingang I2</i></p>  </div> <div style="margin-right: 20px;"> <p><i>Ausgang</i></p>  </div> </div> <p>angezeigte Werte: wahr = -1, unwahr = 0.</p>
	Anschlüsse	<p>Eingang I1: boolescher Wert                  Eingang I2: 24 Bit Integerwert (23 Bits + Vorzeichen). Der Wert 1 entspricht 1 ms.                  Ausgang (O): 24 Bit Integerwert (gepackt boolesch)</p>

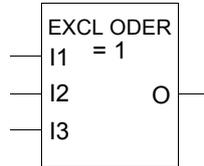
<b>TRIGGER</b>	Type	Zeit-Funktion
	Darstellung	
	Verwendung	<p>Die Anstiegsflanke von Eingang I1 setzt den Ausgang Bit 0 für einen Progr.-Zyklus.                  Die Anstiegsflanke von Eingang I2 setzt den Ausgang Bit 1 für einen Progr.-Zyklus.                  Die Anstiegsflanke von Eingang I3 setzt den Ausgang Bit 2 für einen Progr.-Zyklus.</p> <div style="display: flex; align-items: flex-start;"> <div style="margin-right: 20px;"> <p><i>Eingang I1</i></p>  </div> <div style="margin-right: 20px;"> <p><i>Ausgang, Bit 0</i></p>  </div> </div> <p style="text-align: center;"><b>Beispiel</b> <math>T_c = \text{Programm-Zykluszeit}</math></p>

**Anschlüsse** Eingang I1, I2 und I3: Boolesche Werte  
 Ausgang (O):  
 - 24 Bit Integerwert (23 Bits + Vorzeichen)

**EXCL  
 ODER**

**Typ** Logische Funktion

**Darstellung**



**Verwendung** Der Ausgang ist wahr wenn ein Eingang wahr ist, sonst ist der Ausgang unwahr.  
 Tabelle der Bedingung wahr:

I1	I2	I3	O (binär)	O (angezeigter Wert)
0	0	0	unwahr (alle Bits 0)	0
0	0	1	wahr (alle Bits 1)	-1
0	1	0	wahr (alle Bits 1)	-1
0	1	1	unwahr (alle Bits 0)	0
1	0	0	wahr (alle Bits 1)	-1
1	0	1	unwahr (alle Bits 0)	0
1	1	0	unwahr (alle Bits 0)	0
1	1	1	wahr (alle Bits 1)	-1

**Anschlüsse** Eingang I1, I2 und I3: Boolesche Werte  
 Ausgang (O):  
 - 24 Bit Integerwert (23 Bits + Vorzeichen)

# Istwert-Signale und Parameter des ACS800 Standard-Anwendungsprogramms

---

## Kapitel-Übersicht

In diesem Kapitel werden die Istwert-Signale, Parameter und einstellbaren Parameterwerte des ACS800 Anwendungsprogramms beschrieben, die für die Adaptive Programmierung von Bedeutung sind.

## Istwert-Signale

Die Tabelle unten enthält die Istwert-Signale, die für die Adaptiven Programme verwendet werden. Die Abkürzung FbEq steht für Feldbus-Äquivalent.

Index	Signal Name/Wert	Beschreibung	FbEq.
09	ISTWERTE	Signale für das Adaptive Programm	
09.01	AI1 SKALIERT	Wert von Analogeingang AI1 skaliert zu einem Integerwert.	20000 = 10 V
09.02	AI2 SKALIERT	Wert von Analogeingang AI2 skaliert zu einem Integerwert.	20000 = 20 mA
09.03	AI3 SKALIERT	Wert von Analogeingang AI3 skaliert zu einem Integerwert.	20000 = 20 mA
09.04	AI5 SKALIERT	Wert von Analogeingang AI5 skaliert zu einem Integerwert.	20000 = 20 mA
09.05	AI6 SKALIERT	Wert von Analogeingang AI6 skaliert zu einem Integerwert.	20000 = 20 mA
09.06	HAUPT DS STRW	Steuerwort (STRW) des Haupt-Sollwert Datensatzes der von der Master-Station über die Feldbus-Schnittstelle empfangen wird.	-32768 ... 32767
09.07	HAUPT DS SW 1	Sollwert 1 (SW 1) des Haupt-Sollwert Datensatzes der von der Master-Station über die Feldbus-Schnittstelle empfangen wird.	-32768 ... 32767
09.08	HAUPT DS SW 2	Sollwert 2 (SW2 ) des Haupt-Sollwert Datensatzes der von der Master-Station über die Feldbus-Schnittstelle empfangen wird.	-32768 ... 32767
09.09	HILFS DS SW 1	Sollwert 3 (SW 3) des Hilfs-Sollwert Datensatzes der von der Master-Station über die Feldbus-Schnittstelle empfangen wird.	-32768 ... 32767
09.10	HILFS DS SW 2	Sollwert 4 (SW 4) des Hilfs-Sollwert Datensatzes der von der Master-Station über die Feldbus-Schnittstelle empfangen wird.	-32768 ... 32767
09.11	HILFS DS SW 3	Sollwert 5 (SW 5) des Hilfs-Sollwert Datensatzes der von der Master-Station über die Feldbus-Schnittstelle empfangen wird.	-32768 ... 32767

## Parameter

Die Tabelle unten enthält die Parameter und einstellbaren Parameterwerte, die für die Adaptiven Programme verwendet werden. Die Abkürzung FbEq steht für Feldbus-Äquivalent.

Index	Parametername / Wert	Beschreibung	FbEq
10	START/STOP/DREHR	Parameter, über die das Adaptive Programm Start, Stop und Drehrichtung des Antriebs steuern kann.	
10.01	EXT1 STRT/STP/ DREH		
	PARAM 10.04	Auswahl der Signalquelle durch <a href="#">10.04</a>	17
10.02	EXT2 STRT/STP/ DREH		
	PARAM 10.05	Auswahl der Signalquelle durch <a href="#">10.05</a> .	17
10.04	EXT 1 START ZEIGER	Einstellung der Signalquelle für Parameter <a href="#">10.01</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert: - Parameter-Zeiger: Inversion, Gruppe, Index und Bit-Felder. Die Bitzahl wirkt sich nur auf Bausteine aus, die boolesche Eingänge verarbeiten. - Konstanter Wert: Inversion und Konstant-Felder. Das Invers-Feld muss den Wert C haben, damit die Einstellung der Konstante wirksam ist.	
10.05	EXT 2 START ZEIGER	Einstellung der Signalquelle für Parameter <a href="#">10.02</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
11	SOLLWERTAUSWAHL	Parameter, über die das Adaptive Programm den Sollwert des Antriebs steuern kann.	
11.02	AUSWAHL EXT1/EXT2		
	PARAM 11.09	Auswahl der Signalquelle durch <a href="#">11.09</a> .	16
11.03	AUSW EXT SOLLW 1		
	PARAM 11.10	Auswahl der Signalquelle durch <a href="#">11.10</a> .	37
11.06	AUSW EXT SOLLW 2		
	PARAM 11.11	Auswahl der Signalquelle durch <a href="#">11.11</a> .	38
11.09	AUSWAHL EXT 1/ EXT2	Einstellung der Signalquelle für Parameter <a href="#">11.02</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
11.10	EXT 1 SW ZEIGER	Einstellung der Signalquelle für Parameter <a href="#">11.03</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
11.11	EXT 2 SW ZEIGER	Einstellung der Signalquelle für Parameter <a href="#">11.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14	RELAISAUSGÄNGE	Parameter, über die das Adaptive Programm die Relaisausgänge des Antriebs steuern kann.	
14.01	RELAIS RO1 AUSG.		

Index	Parametername / Wert	Beschreibung	FbEq
	PARAM 14.16	Auswahl der Signalquelle durch Parameter <a href="#">14.16</a> .	36
14.02	RELAIS RO2 AUSG.		
	PARAM 14.17	Auswahl der Signalquelle durch Parameter <a href="#">14.17</a> .	36
14.03	RELAIS RO3 AUSG.		
	PARAM 14.18	Auswahl der Signalquelle durch Parameter <a href="#">14.18</a> .	36
14.10	DIO MOD1 RO1		
	PARAM 14.19	Auswahl der Signalquelle durch Parameter <a href="#">14.19</a> .	7
14.11	DIO MOD1 RO2		
	PARAM 14.20	Auswahl der Signalquelle durch Parameter <a href="#">14.20</a> .	7
14.12	DIO MOD2 RO1		
	PARAM 14.21	Auswahl der Signalquelle durch Parameter <a href="#">14.21</a> .	7
14.13	DIO MOD2 RO2		
	PARAM 14.22	Auswahl der Signalquelle durch Parameter <a href="#">14.22</a> .	7
14.14	DIO MOD3 RO1		
	PARAM 14.23	Auswahl der Signalquelle durch Parameter <a href="#">14.23</a> .	7
14.15	DIO MOD3 RO2		
	PARAM 14.24	Auswahl der Signalquelle durch Parameter <a href="#">14.24</a> .	7
14.16	RO ZEIGER1	Einstellung der Signalquelle für Parameter <a href="#">14.01</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14.17	RO ZEIGER2	Einstellung der Signalquelle für Parameter <a href="#">14.02</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14.18	RO ZEIGER3	Einstellung der Signalquelle für Parameter <a href="#">14.03</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14.19	RO ZEIGER4	Einstellung der Signalquelle für Parameter <a href="#">14.10</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14.20	RO ZEIGER5	Einstellung der Signalquelle für Parameter <a href="#">14.11</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14.21	RO ZEIGER6	Einstellung der Signalquelle für Parameter <a href="#">14.12</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14.22	RO ZEIGER7	Einstellung der Signalquelle für Parameter <a href="#">14.13</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14.23	RO ZEIGER8	Einstellung der Signalquelle für Parameter <a href="#">14.14</a> .	

Index	Parametername / Wert	Beschreibung	FbEq
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
14.24	RO ZEIGER9	Einstellung der Signalquelle für Parameter <a href="#">14.15</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
15	ANALOGAUSGÄNGE	Parameter, über die das Adaptive Programm die Standard-Analogausgänge des Antriebs steuern kann.	
15.01	ANALOGAUSGANG 1		
	PARAM 15.11	Auswahl der Signalquelle durch <a href="#">15.11</a>	17
15.06	ANALOGAUSGANG 2		
	PARAM 15.12	Auswahl der Signalquelle durch <a href="#">15.12</a>	16
15.11	AO1 ZEIGER	Einstellung der Signalquelle für Parameter <a href="#">15.01</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
15.12	AO2 ZEIGER	Einstellung der Signalquelle für Parameter <a href="#">15.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
16	STEUEREINGÄNGE	Parameter, über die das Adaptive Programm die Steuereingänge des Antriebs steuern kann.	
16.01	FREIGABE		
	PARAM 16.08	Auswahl der Signalquelle durch Parameter <a href="#">16.08</a> .	15
16.08	FREIGABE ZEIGER	Einstellung der Signalquelle für Parameter <a href="#">16.01</a>	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
20	GRENZEN	Parameter, über die das Adaptive Programm die Grenzwert-Einstellungen für den Antriebsprozess steuern kann.	
20.13	MIN MOMENT AUSW	Auswahl des Grenzwerts für das Mindest-Drehmoment	
	PARAM 20.18	Auswahl der Grenzwert-Einstellung durch Param. 20.18	20
20.14	MAX MOMENT AUSW	Auswahl des Grenzwerts für das Maximal-Drehmoment	
	PARAM 20.19	Auswahl der Grenzwert-Einstellung durch Param. 20.19	19
20.18	MIN MOMENT ZEIGER	Einstellung des Grenzwertes für Param. 20.13	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
20.19	MAX MOMENT ZEIGER	Einstellung des Grenzwertes für Param. 20.14	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	

Index	Parametername / Wert	Beschreibung	FbEq
22	RAMPEN	Parameter, über die das Adaptive Programm die Beschleunigungs- und Verzögerungs-Rampen für den Antrieb steuern kann.	
22.01	AUSW. RAMPE		
	PAR 22.08 u. 09	Beschl.- und Verz.-Rampen-Einstellungen durch Parameter 22.08 und 22.09	15
22.08	ZEIGER BESCHL	Einstellung der Signalquelle für Parameter 22.01	100 = 1 s
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
22.09	ZEIGER VERZ	Einstellung der Signalquelle für für Parameter 22.01	100 = 1 s
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
26	MOTORSTEUERUNG	Parameter, über den das Adaptive Programm die Fluss-Werte für den Antrieb steuern kann.	
26.06	FLUSS SW ZEIGER	Auswahl der Signalquelle für den Fluss-Sollwert.	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
40	PID REGLER	Parameter, über die das Adaptive Programm die PID-Regelung des Prozesses steuern kann.	
40.07	AUSW. EING. ISTW1		
	PARAM 40.25	Auswahl der Signalquelle durch Parameter <a href="#">40.25</a> .	6
40.25	ISTWERT1 ZEIGER	Einstellung der Signalquelle für Parameter <a href="#">40.07</a>	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
83	ADAPT PROG STRG	Ausführungs-Steuerung des Adaptiven Programms.	
83.01	ADAPT PROG MODUS	Auswahl der Betriebsart des Adaptiven Programms.	
	STOP	Stop. Das Programm kann nicht editiert werden.	
	START	Läuft. Das Programm kann nicht editiert werden.	
	EDITIEREN	Stop und in den Editiermodus wechseln. Das Programm kann editiert werden.	
83.02	EDITIERBEFEHL	Wählt den Befehl für den Baustein der mit Parameter <a href="#">83.03</a> festgelegt wurde. Das Programm muss sich im Editiermodus befinden. (Siehe Parameter <a href="#">83.01</a> .)	
	KEIN	Ausgangswert. Der Wert wird auf KEIN zurückgesetzt, nachdem ein Editierbefehl ausgeführt worden ist.	

Index	Parametername / Wert	Beschreibung	FbEq																											
	SCHIEBEN	<p>Schiebt den Baustein in die mit Parameter <a href="#">83.03</a> festgelegte Position und setzt die folgende Bausteine eine Position vor/höher. An der freigewordenen Position kann ein neuer Baustein durch Programmieren der Baustein-Parameter eingefügt werden.</p> <p><b>Beispiel:</b> Ein neuer Baustein muss zwischen dem aktuellen Baustein Nummer vier (Parameter 84.20 ... 84.25) und fünf (Parameter 84.25 ... 84.29) eingefügt werden.</p> <p>Die geschieht wie folgt:</p> <ul style="list-style-type: none"> <li>- Im Programm mit Parameter <a href="#">83.01</a> in den Editier-Modus wechseln.</li> <li>- Die Position Nummer fünf als gewünschte Position für den neuen Baustein mit Parameter <a href="#">83.03</a> einstellen.</li> <li>- Den neuen Baustein in Position Nummer 5 schieben und alle folgenden Bausteine mit Parameter <a href="#">83.02</a> eine Position weiter schieben (Auswahl: SCHIEBEN).</li> <li>- Die freigemachte Position Nummer 5 mit den Parametern 84.25 bis 84.29 wie üblich programmieren.</li> </ul>																												
	LOESCHEN	Löscht den Baustein in der mit Parameter <a href="#">83.03</a> eingestellten Position und schiebt die folgenden Bausteine eine Stufe nach unten.																												
83.03	EDITIERTER BLOCK	Bestimmt die Positions-Nummer des Bausteins für den mit Parameter <a href="#">83.02</a> eingestellten Befehl.																												
	1 ... 15	Baustein-Positions-Nummer.																												
83.04	ZEITRASTER-AUSWAHL	Stellt die Ausführungs-Zykluszeit für das Adaptive Programm ein. Die Einstellung gilt für alle Bausteine.																												
	12 ms	12 Millisekunden																												
	100 ms	100 Millisekunden																												
	1000 ms	1000 Millisekunden																												
84	ADAPTIVES PROGRAM	Erstellung und Diagnose des Adaptiven Programms.																												
84.01	STATUS	<p>Anzeige des Statusworte des Adaptiven Programms. In der Tabelle unten sind die Bit-Status-Alternativen und die entsprechenden Werte auf der Steuertafelanzeige aufgeführt.</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Display</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>Gestoppt</td> </tr> <tr> <td>1</td> <td>2</td> <td>Läuft</td> </tr> <tr> <td>2</td> <td>4</td> <td>Gestört</td> </tr> <tr> <td>3</td> <td>8</td> <td>Editieren</td> </tr> <tr> <td>4</td> <td>10</td> <td>Prüfen</td> </tr> <tr> <td>5</td> <td>20</td> <td>Pushing</td> </tr> <tr> <td>6</td> <td>40</td> <td>Popping</td> </tr> <tr> <td>8</td> <td>100</td> <td>Initialisieren</td> </tr> </tbody> </table>	Bit	Display	Bedeutung	0	1	Gestoppt	1	2	Läuft	2	4	Gestört	3	8	Editieren	4	10	Prüfen	5	20	Pushing	6	40	Popping	8	100	Initialisieren	
Bit	Display	Bedeutung																												
0	1	Gestoppt																												
1	2	Läuft																												
2	4	Gestört																												
3	8	Editieren																												
4	10	Prüfen																												
5	20	Pushing																												
6	40	Popping																												
8	100	Initialisieren																												
84.02	FEHLERHAFTE PARAM	Zeigt die fehlerhaften Parameter-Einstellungen im Adaptiven Programm.																												
84.05	BLOCK1	Auswahl der Funktionsbausteine für Baustein-Parameter-Satz 1.																												
	ABS	Siehe Kapitel <a href="#">Funktionsbausteine</a> .																												
	ADD	Siehe Kapitel <a href="#">Funktionsbausteine</a> .																												
	UND	Siehe Kapitel <a href="#">Funktionsbausteine</a> .																												
	VERGLEICH	Siehe Kapitel <a href="#">Funktionsbausteine</a> .																												
	EREIGNIS	Siehe Kapitel <a href="#">Funktionsbausteine</a> .																												

Index	Parametername / Wert	Beschreibung	FbEq
	FILTER	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	MAX	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	MIN	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	MULDIV	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	LEER	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	ODER	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	PI	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	PI-INIT	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	SR	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	BIN SCHALTER	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	INT SCHALTER	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	AUS VERZ	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	EIN VERZ	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	TRIGGER	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
	EXCL ODER	Siehe Kapitel <a href="#">Funktionsbausteine</a> .	
84.06	EINGANG 1	Auswahl der Signalquelle für Eingang I1 von Baustein-Parameter-Satz 1 (BPS1).	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert: - Parameter-Zeiger: Inversion, Gruppe, Index und Bit-Felder. Die Bitzahl wirkt sich nur auf Bausteine aus, die boolesche Eingänge verarbeiten. - Konstanter Wert: Inversion und Konstant-Felder. Das Invers-Feld muss den Wert C haben, damit die Einstellung der Konstante wirksam ist. <b>Beispiel:</b> Der Status von Digitaleingang DI2 wird an Eingang 1 wie folgt angeschlossen: - Den Parameter (84.06) für die Auswahl der Signalquelle auf +.01.17.01 einstellen. (Das Anwendungsprogramm speichert den Status von Digitaleingang DI2 nach Bit 1 des Istwertsignals 01.17.) - Den Wert durch ändern des Vorzeichens des Zeiger-Wertes invertieren (-01.17.01.).	
84.07	EINGANG 2	Siehe Parameter <a href="#">84.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Siehe Parameter <a href="#">84.06</a> .	
84.08	EINGANG 3	Siehe Parameter <a href="#">84.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Siehe Parameter <a href="#">84.06</a> .	
84.09	AUSGANG	Speichert und zeigt den Ausgangswert von Baustein-Parameter-Satz 1 an.	
...	...	Speichert und zeigt den Ausgangswert von Baustein-Parameter-Satz 15 an.	
84.79	AUSGANG	Speichert den Ausgangswert von Baustein-Parameter-Satz 15. Siehe Parameter <a href="#">84.09</a> .	
85	NUTZER KONSTANTEN	Speichern der Konstanten und Meldungen des Adaptiven Programms.	
85.01	KONSTANTE1	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.02	KONSTANTE2	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	

Index	Parametername / Wert	Beschreibung	FbEq
85.03	KONSTANTE3	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.04	KONSTANTE4	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.05	KONSTANTE5	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.06	KONSTANTE6	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.07	KONSTANTE7	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.08	KONSTANTE8	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.09	KONSTANTE9	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.10	KONSTANTE10	Einstellung einer Konstante für das Adaptive Programm.	
	-8388608 bis 8388607	Integerwert.	
85.11	ZEICHENKETTE1	Speichert eine Meldung für die Verwendung im Adaptiven Programm (Baustein EREIGNIS).	
	NACHRICHT1	Nachricht/Meldung	
85.12	ZEICHENKETTE2	Speichert eine Meldung für die Verwendung im Adaptiven Programm (Baustein EREIGNIS).	
	NACHRICHT2	Nachricht/Meldung	
85.13	ZEICHENKETTE3	Speichert eine Meldung für die Verwendung im Adaptiven Programm (Baustein EREIGNIS).	
	NACHRICHT3	Nachricht/Meldung	
85.14	ZEICHENKETTE4	Speichert eine Meldung für die Verwendung im Adaptiven Programm (Baustein EREIGNIS).	
	NACHRICHT4	Message	
85.15	ZEICHENKETTE5	Speichert eine Meldung für die Verwendung im Adaptiven Programm (Baustein EREIGNIS).	
	NACHRICHT5	Nachricht/Meldung	
96	EXT AO	Parameter, durch deren Einstellung das Adaptive Programm die optionalen Analogausgänge des Antriebs steuern kann.	
96.01	EXT AO1		
	PARAM 96.11	Auswahl der Signalquelle durch Parameter <a href="#">96.11</a> .	16
96.06	EXT AO2		
	PARAM 96.12	Auswahl der Signalquelle durch Parameter <a href="#">96.12</a> .	16
96.11	EXT AO1 ZEIGER	Einstellung der Signalquelle für <a href="#">96.01</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	
96.12	EXT AO2 ZEIGER	Einstellung der Signalquelle für <a href="#">96.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameter-Zeiger oder konstanter Wert. Siehe Parameter <a href="#">10.04</a> .	

# Anwender-Dokumentation des Adaptiven Programms

---

## Kapitel-Übersicht

Dieses Kapitel enthält drei Blanko-Vorlagenblätter für Blockschaltbilder, auf denen die Adaptive Programmierung dokumentiert werden kann.











---

**ABB Automation Products GmbH**

Motors & Drives  
Edisonstraße 15  
D-68623 Lampertheim  
DEUTSCHLAND  
Telefon +49 (0)6206 503 503  
Telefax +49 (0)6206 503 600  
Internet [www.abb.com/motors&drives](http://www.abb.com/motors&drives)

**ABB Industrie & Gebäudesysteme**

**GmbH** Wienerbergstraße 11 B  
A-1810 Wien  
ÖSTERREICH  
Telefon +43-(0)1-60109-0  
Telefax +43-(0)1-60109-8305

**ABB Normelec AG**

Badener Straße 790  
CH-8048 Zürich  
SCHWEIZ  
Telefon +41-(0)1-4356666  
Telefax +41-(0)1-4356605

3AFE64527177 Rev C/ DE  
GÜLTIG AB: 08.04.2005